AD A117090

AGARD-AG-256

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

AGARDograph No. 256

Advances in the Techniques and Technology of the Application of Nonlinear Filters and Kalman Filters

DTIC
ELECTE
JUL 2 0 1982
D

NORTH ATLANTIC TREATY ORGANIZATION



DTIC FILE COPY

DISTRIBUTION AND AVAILABILITY
ON BACK COVER

82 07 19 198

NORTH ATLANTIC TREATY ORGANIZATION

ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT

(ORGANISATION DU TRAITE DE L'ATLANTIQUE NORD)

AGARDograph No.256

ADVANCES IN THE TECHNIQUES AND TECHNOLOGY OF THE APPLICATION OF NONLINEAR FILTERS AND KALMAN FILTERS

Edited by

Professor C.T.Leondes, Ph.D.,
School of Engineering and Applied Science
University of California, Los Angeles
7620 Boelter Hall
Los Angeles, California 90024
USA

DTIC
ELECTE
JUL 20 1982
D

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

This AGARDograph has been prepared and edited at the request of the Guidance and Control Panel of AGARD.

THE MISSION OF AGARD

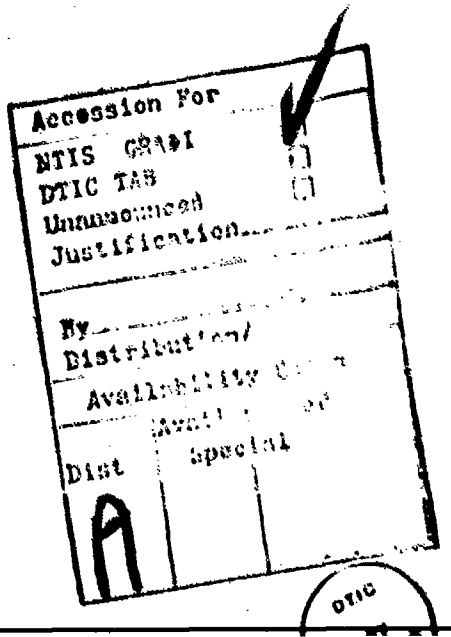
The mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Exchanging of scientific and technical information;
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Providing scientific and technical advice and assistance to the North Atlantic Military Committee in the field of aerospace research and development;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced
directly from material supplied by AGARD or the authors.



Published March 1982

Copyright © AGARD 1982
All Rights Reserved

ISBN 92-835-1418-1

Printed by Technical Editing and Reproduction Ltd
Harford House, 7–9 Charlotte St. London, W1P 1HD

PREFACE

The beginning of the theory of filtering is generally rather closely associated with the development of the Wiener filter, and, as so often happens, this development was motivated by applied issues, namely, the development of improved fire control techniques in World War II. In this application the (radar) sensors have stochastic (noise) inputs. Additionally, the objects (targets) being tracked were buffeted by stochastic (wind) disturbances. The criterion for optimum (Wiener) filter design was the minimization of the time average of the error squared, where the error was defined as the difference between the true target position as a function of time and the best or optimum estimate of the target position generated as a function of time by the Wiener filter. The development of the Wiener filter was based on the solution of the Wiener Hopf integral equation for the case of stationary stochastic processes whose descriptive statistical parameters were invariant with time. The Wiener Hopf integral equation was formulated as an integral equation for the weighting function of the linear time variant system which would minimize the ensemble average of the error squared in the case of non stationary stochastic processes, that is, stochastic processes whose descriptive statistical parameters were functions of time. By limiting the stochastic processes under consideration to stationary stochastic processes it was possible to take the Fourier transform of the Wiener Hopf integral equation, which was formulated in the time domain. Therefore, it was possible to go from the time domain to the frequency domain, in the case of stationary stochastic processes, and more readily solve the Fourier transformed Wiener Hopf integral equation for the transfer function of the (time invariant) Wiener filter.

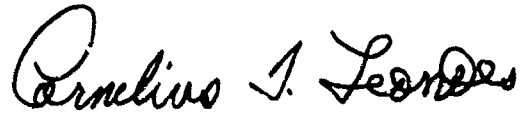
Now while the development of the optimum filter for stationary stochastic processes is useful in many applied instances there are many applied instances where nonstationary stochastic processes are involved. Thus the development of the optimum filter in this case was also essential.

In the period after World War II until the late 1950's a great deal of effort was expended on the international scene to developing the optimum filter in this case. While there was much in the way of interesting research results published in the technical literature, it remained for the development of the Kalman filter, presented in published papers and reports in and around 1960, for an effective solution to this problem in the continuing time case and a computationally efficient solution in the much more pervasive discrete time case. The discrete time case is much more pervasive today, of course, because of the implementation of virtually all Kalman filter applications by digital computers. The continuous time Kalman filter may be viewed as being generated by the differentation of the Wiener Hopf integral equation. What results in the process, very simply, is the structure of the Kalman filter, and that is what is really desired in any event, namely, the implementation of the Kalman filter in the continuous time case. In the case of the Kalman filter in the discrete time case this is very simply generated by noting that the minimum variance filter, that is the filter which minimizes the ensemble average of the error squared at discrete times, is the conditional expectation of the system state at any particular discrete time conditioned on the specific observations or measurements which have resulted at all discrete times up to and including the discrete time at which the minimum variance estimate is being determined by the Kalman filter. By utilizing certain rather straightforward and basic expressions this Kalman filter estimate can be generated rather readily in a convenient expression form as a linear function of the Kalman filter estimate of the system state at the just preceding discrete time, for instance, plus the noisy observation or measurement of the system state (vector) at the present time.

The next area for which the theoretical foundation had to be developed was that of filters for nonlinear dynamic systems. Regardless of whether a dynamic systems of interest is linear or nonlinear the minimum variance estimator is the conditional expectation of the dynamic system state (vector), conditioned on the previous observations or measurements of the noisy system state measurement vector. In the case of the Kalman filter this is rather easily generated as a convenient expression as noted above. In the case of nonlinear dynamic systems this is an extremely difficult task, in general, and, with rare exceptions, has to be dealt with through some approximate and effective linearitation technique. Specifically, in order to generate the conditional expectation, an expression for the conditional probability density function has to be generated. This is done through the use of the Ito calculus which, basically, correctly recognizes terms which appear to be second order terms as first order terms in going to limiting processes in the process of developing the equation which describes the evolution of the conditional probability density function, and thus generating this equation correctly. This equation, which is the equation for the evolution of the conditional probability density function, is known as the Fokker Planck equation. Since it is a nonlinear partial differential equation of high order if the system state vector is of high order its solution is a matter of very considerable computational complexity, and is very difficult to implement in any given system application, as a rule. This then forces the generation of effective approximate linearization techniques which are quite capable of providing adequate accurate filters. Such approximate filters have been generated and are known as extended Kalman filters, higher order extended Kalman filters, or are described by other terms, depending on the approximation technique.

Once this theoretical foundation for Kalman filtering techniques, nonlinear filtering techniques and effective approximation techniques therefore was put in place, the problem became that of developing system application techniques, dealing effectively with computational issues and techniques, and it is the intent of this NATO AGARDograph to treat effectively all of the above issues and techniques with particular emphasis on application to complex problems and systems of paramount interest to the NATO community. As such this NATO AGARDograph represents a unique document on the international scene.

A great debt of gratitude is owed by the editor to many individuals. First of all the co-authors of this rather prodigious undertaking all deserve an immeasureable vote of thanks and gratitude for their formidable contributions to this volume. Next the editor would like to express his gratitude for the support and encouragement of his fellow NATO AGARD Panel members in addition to the Panel Chairman, Geoffrey Howell, and Deputy Panel Chairman, Ronald Vaughn. Additionally, the continual first class support of Bernard Heliot, the Panel Executive, on this rather elaborately complex undertaking is greatly appreciated. Finally, the Editor's secretary, Bernice Roos, was of indispensable help and support on this undertaking.



CORNELIUS T.LEONDES
Editor

CONTENTS

	Page
PREFACE by C.T.Leondes	iii
	Reference
PART I – ADVANCED TOPICS IN THE THEORY OF NONLINEAR FILTERS AND KALMAN FILTERS	
NONLINEAR FILTERING THEORY by V.Krebs	1
EXACT AND APPROXIMATE NONLINEAR ESTIMATION TECHNIQUES by D.F.Liang	2
THE THEORY AND TECHNIQUES OF DISCRETE-TIME DECENTRALIZED FILTERS by T.H.Kerr and L.Chin	3
PART II – COMPUTATIONAL TECHNIQUES IN NONLINEAR AND LINEAR FILTERS	
ADVANCES IN COMPUTATIONAL EFFICIENCIES OF LINEAR FILTERING by L.Chin	4
DESIGN OF REAL-TIME ESTIMATION ALGORITHMS FOR IMPLEMENTATION IN MICROPROCESSOR AND DISTRIBUTED PROCESSOR SYSTEMS by V.B.Gylys	5
GLOBAL APPROXIMATION FOR NONLINEAR FILTERING WITH APPLICATION TO SPREAD SPECTRUM RANGING by W.M.Bowles and J.A.Cartelli	6
SYSTEM IDENTIFICATION OF NONLINEAR AERODYNAMIC MODELS by T.L.Trankle, J.H.Vincent and S.N.Franklin	7
TECHNIQUES AND METHODOLOGIES FOR THE ESTIMATION OF COVARIANCES, POWER SPECTRA AND FILTER-STATE AUGMENTATION by V.Held	8
PART III – ADVANCED NONLINEAR AND KALMAN FILTER APPLICATION AND METHODOLOGIES	
REDUCED ORDER KALMAN FILTER DESIGN AND PERFORMANCE ANALYSIS by P.S.Maybeck	9
DESIGN AND PERFORMANCE ANALYSIS OF AN ADAPTIVE EXTENDED KALMAN FILTER FOR TARGET IMAGE TRACKING by P.S.Maybeck	10
TECHNIQUES FOR THE DEVELOPMENT OF ERROR MODELS FOR AIDED STRAPDOWN NAVIGATION SYSTEMS by W.Lechner	11
USE OF FILTERING AND SMOOTHING ALGORITHMS IN THE ANALYSIS OF MISSILE SYSTEM TEST DATA by E.M.Duiven, C.L.Medler and J.F.Kasper, Jr.	12
INERTIAL NAVIGATION SYSTEM ERROR MODEL CONSIDERATIONS IN KALMAN FILTER APPLICATIONS by J.R.Huddle	13
OPTIMAL FILTERING AND CONTROL TECHNIQUES FOR TORPEDO-SHIP TRACKING SYSTEMS by R.Lunderstädt and R.Kern	14

Reference

SEPARATED-BIAS ESTIMATION AND SOME APPLICATIONS 15
by B.Friedland

COMPARISONS OF NONLINEAR FILTERS FOR SYSTEMS WITH NON-NEGLIGIBLE NONLINEARITIES 16
by D.F.Liang

KALMAN FILTER SATELLITE ORBIT IMPROVEMENT USING LASER RANGING MEASUREMENTS FROM A SINGLE TRACKING STATION 17
by K.F.Wakker and B.A.C.Ambrosius

STATE ESTIMATION OF BALLISTIC TRAJECTORIES WITH ANGLE ONLY MEASUREMENTS 18
by M.R.Salazar

NEW SMOOTHING ALGORITHMS FOR DYNAMIC SYSTEMS WITH OR WITHOUT INTERFERENCE 19
by K.Demirbaş

NONLINEAR FILTERING THEORY

by

Volker Krebs
Bodenseewerk Gerätetechnik GMBH
Postfach 11 20, D7770 Überlingen
Federal Republik of Germany

SUMMARY

After a historical review of the development of estimation theory, an introduction to nonlinear filtering theory is presented by means of a deductive approach.

First the so-called general estimation problem is defined which is concerned with the extraction of useful information from noisy measurements. Having then introduced nonlinear stochastic dynamical models for the signal process, the exact mathematical solution of the general estimation problem is outlined which yields the Kushner-Stratonovich equation respectively the Bayesian recursive estimator. From these equations the practical nonlinear filter approximations are derived in a deductive way. Most important are the local approximate filters of first order (type: extended Kalman Filter) and second order filters. In addition, there are global approximate filters (Bayes-law calculators) available. Several approximation methods for these estimators, such as orthogonal series expansion, Gaussian sums, point masses and splines are briefly discussed.

1. INTRODUCTION

Nonlinear filtering theory has been developed since the beginning of the sixties, nearly during the same time as the theory of optimal linear filters such as Wiener filters and Kalman-Bucy filters.

Nonlinear filtering is the theoretical as well as practical solution of the so-called general estimation problem which is concerned with the extraction of useful information from noisy measurements of certain signals.

This general estimation problem which will be discussed in more detail in the next paragraph, covers many problems in the field of communication- and control engineering applications such as

- suppression of noise by filtering
- estimation of states of linear or nonlinear stochastic dynamical systems
- parameter estimation for statical or dynamical systems.

Interest in particular modes of the general estimation problem, especially in celestial mechanics, dates back a long time; probably even to the work of Claudius Ptolemaeus /1/ in the second century. He assumed a geocentric model structure of the universe which required the assumption of epicycloids for the planets' orbital motion. Having used measured data from observations, Ptolemaeus evaluated the orbital parameters of the planet Mars, which allowed the prediction of the planet's position with reasonable accuracy.

In the astronomy of modern times it has been Legendre /2/ and independently Gauss /3/, who developed in 1795 the method of least squares for the minimization of the observation errors by determining the orbit of celestial bodies. The least squares or regression analysis technique may be regarded as the fundamentals of todays optimal estimation and filtering algorithms. Gauss already recognized the possibility of a dual approach to regression analysis:

It may be considered as

i) a deterministic optimization problem (minimizing the sum of the squared errors)

ii) a stochastic estimation problem (evaluation of the most probable parameter estimates which implies calculation of a probability density function and determination of its maximum).

As reported by Deutsch /4/ and Sorenson /5/, Gauss anticipated by that to some extent the maximum-likelihood estimation technique suggested by R.A.Fisher /6/ in 1912.

In the forties of our century the least squares principle was applied by Kolmogorov /7/ and Wiener /8/ to the problem of separating wide-band noise and signal processes, i. e. to the filtering problem. In the sequel the Wiener-Kolmogorov filter or Wiener filter which may be realized as an electrical network has found many applications in communication engineering.

The extension of the Wiener-Kolmogorov theory by Kalman /9/ 1960, and Kalman and Bucy /10/ 1961 to linear multivariable instationary processes by introducing the state space concept, was an important improvement and has opened a wide field of applications in the subsequent years /11/. This is because the Kalman filter is a recursive algorithm which is

easily to be implemented on a digital computer. A survey of the history of early aerospace Kalman filter applications is given in /12/.

As mentioned before, the theory of nonlinear filtering has been developed independently but nearly in parallel to the linear theory. This is probably due to the fact that the approach to either of the theories is quite different.

For the understanding of the (linear) Kalman filter on the one hand, a least squares approach may be used where no deeper insight into probability theory is required (see e. g. /13/, /14/). The nonlinear filtering theory on the other hand is usually based on a probabilistic approach; the solution of the nonlinear filtering problem requires then the calculation of the probability density of the state conditioned on all available observations and the initial probability density function. The mathematical tools for tackling this problem are partial differential equations and stochastic integrals.

Advances in the area of nonlinear filtering are due to Stratonovich /15/, Kushner /16/, Bucy /17/, Wonham /18/ and many others. However, it should be mentioned that the roots of this theory may be traced back to the beginning of the twentieth century and the study and mathematical description of diffusion processes (Einstein /19/). This is because the differential equation of a diffusion process and the mathematical model of a signal process affected by white noise (which is the basic model for all optimal nonlinear estimation algorithms) are equivalent.

After this historical review of the development of estimation theory we will give an outlook of the organization of this contribution.

In the following paragraph we will first introduce the general estimation problem and its mathematical solution which will lead us to the Fokker-Planck respectively Kushner-Stratonovich equation in the time-continuous case and the Bayesian recursive estimation equations in the time-discrete case.

Then we will discuss local approximations for nonlinear estimators i. e. easy implementable practical filters such as first order filters (type extended Kalman filter or iterated extended Kalman filters) and higher order filters.

Finally we will shortly outline some possibilities of global approximations for discrete-time nonlinear filters (Bayes-law calculators) such as orthogonal series expansion, Gaussian sums, point masses, and splines.

2. THE SOLUTION OF THE NONLINEAR FILTERING PROBLEM

2.1 The general estimation problem

Given a signal $s(t)$ which is disturbed by additive noise $n(t)$. The sum of both signals

$$y(t) = s(t) + n(t) \tag{1}$$

is observed (measured). In the most simple case, the general estimation problem consists of the evaluation of the signal $s(t_1)$ by processing of the information contained in all available measurements

$$Y_t := \left[y(t_0), \ldots, y(t)\right] := \left[y(\tau),\ t_0 \leq \tau \leq t\right] \tag{2}$$

through an estimator. Thus the desired ideal output of the estimator (Fig.1)

$$f_a(t) = s(t_1) \ , \ t_1 \gtrless t \tag{3}$$

will in reality be an estimate of this value, denoted by

$$f_a(t) = \hat{s}(t_1|t) := \hat{s}(t_1|Y_t). \tag{4}$$

This estimate has to be optimal in some sense which leads to the interpretation of the general estimation problem as an optimization problem. A cost functional $J(\varepsilon)$ which depends on the estimation error

$$\varepsilon(t_1) := s(t_1) - \hat{s}(t_1|t) \tag{5}$$

has to be defined and minimized with respect to this error.

As shown in Fig.1, we have three different types of estimation (prediction, filtering and smoothing) depending on the instant t_1 relative to the present time t.

Moreover, we distinguish three modes of estimation which are related to the kind of signal and observation processes:

- the discrete estimation problem where signal and observation are (discrete-time) random sequences

- the continuous-discrete estimation problem where the signal process is a (continuous) stochastic process and the observation is a (discrete) random sequence.

- the continuous estimation problem where signal and observation are continuous stochastic processes.

Finally, we speak of nonlinear or linear estimation when we process the observations in a nonlinear respectively linear manner in the estimator.

We will now follow the probabilistic approach to the general estimation problem.

Since $\{s(t)\}$, $\{n(t)\}$, and thus $\{y(t)\}$ are considered stochastic processes we surely want to know the probability of certain signal values $s(t_1)$ under the condition of a given realization

$$Y_t = \left[y(\tau),\ t_0 \leq \tau \leq t \right]$$

of the observation process $\{y(t)\}$. Generalized this means knowledge of the conditional probability density of $s(t_1)$ given Y_t, denoted by

$$p\left[s(t_1),\ t_1 \mid Y_t \right].$$

This function embodies all statistical information about $s(t_1)$ which is contained in the available observations.
The solution of the general estimation problem is consequently given by equations for the evolution of this conditional probability density function $p[s(t_1), t_1|Y_t]$, starting with the initial information $p[s(t_0), t_0]$.

If we know the conditional density we can obtain in a comparatively simple way special estimates of the signal (Fig.2), e. g.

- $\bar{s}_{MAP}$: the most probable estimate (Maximum A Posterori estimate) which indicates the maximum of the a posteriori density $p[s(t_1)|t]$

- $\bar{s}_{MV}$: the Minimum Variance estimate which indicates the center of gravity of the area under the density $p[s(t_1) \mid t]$.

The minimum variance estimate is of special importance since the underlying quadratic loss

$$J(\bar{s}) := E\{\bar{s}^2\} = E\left\{ \left[s(t_1) - \bar{s}(t_1|t) \right]^2 \right\} \tag{6}$$

weights larger errors in a stronger way than smaller ones, is independent of the sign of the error, and is last but not least mathematically well tractable.

It is easy to show (e. g. /23/) that the minimum variance estimate is given by the conditional mean (which we will indicate in the sequel by the symbol $\wedge$), and we have

$$\bar{s}_{MV}(t_1) = E\{ s(t_1) \mid Y_t \} := \hat{s}(t_1|t). \tag{7}$$

Moreover, we note that the conditional mean is an unbiased estimate, that is

$$E\{s(t_1) - s(t_1|t)\} = E\{s(t_1)\} - \underbrace{E\{E\{s(t_1)|Y_t\}\}}_{E\{s(t_1)\}} = 0 \tag{8}$$

2.2 Dynamical models for signal and noise

For a practical development of the equations of evolution for the conditional probability density function we have to specify mathematical models for the signal and noise processes. A model which is sufficiently close to the "real world" on the one hand, and analytically tractable on the other hand, is given by a Markov process in state space notation which may be described in continuous time by a nonlinear vector stochastic differential equation of diffusion type

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t), t] + \underline{G}[\underline{x}(t), t]\underline{w}(t), \quad t \geq t_0 \ ; \tag{9a}$$

$\underline{x}$ and $\underline{f}$ are n-vectors, $\underline{G}$ is of dimension n x q and $\{\underline{w}(t),\ t \geq t_0\}$ is a q-vector white Gaussian noise process with $E\{\underline{w}(t)\} = \underline{0}$ and $E\{\underline{w}(t) \cdot \underline{w}^T(\tau)\} = \underline{Q}\,\delta(t-\tau)$.

Equation (9a) can neither be integrated in the Riemann sense nor in the Riemann-Stieltjes sense; it requires for proper handling the introduction of a stochastic integral (which is due to the delta correlation of the white noise), thus we better write Eq. (9a) as the (Itô) stochastic differential equation

$$d\underline{x}(t) = \underline{f}[\underline{x}(t), t]dt + \underline{G}[\underline{x}(t), t]d\underline{\beta}(t), \quad t \geq t_0 \tag{9b}$$

where $\{\underline{\beta}(t)\}$ is a Wiener process with

$$E\{d\underline{\beta}(t)\} = \underline{0}\ ,\ E\{d\underline{\beta}(t)d\underline{\beta}^T(t)\} = \underline{Q}(t)\cdot dt. \tag{9c}$$

Eq.(9b) is formally equivalent to Eq.(9a) with $\underline{w}(t) := d\beta/dt$.

The initial state $\underline{x}(t_o)$ is often assumed a Gaussian random variable with known mean $\hat{\underline{x}}(t_o)$ and covariance matrix $\underline{P}(t_o)$ but it can be characterized as well by any other distribution $p[\underline{x}(t_o), t_o]$.

Moreover, we have a nonlinear m vector observation process including additive measurement noise,

$$\underline{y}(t) = \underline{h}[\underline{x}(t),t] + \underline{v}(t) \tag{10a}$$

where $\{\underline{v}(t),\ t \geq t_o\}$ is Gaussian white noise with $E\{\underline{v}(t)\} = \underline{0}$ and $E\{\underline{v}(t)\underline{v}^T(\tau)\} = \underline{R}\,\delta(t-\tau)$ which we write in the same way as before as an Itô differential equation

$$d\underline{z}(t) = \underline{h}[\underline{x}(t),t]dt + d\underline{\eta}(t)\ , \tag{10b}$$

this is formally equivalent to Eq.(10a) with the definitions

$$\underline{y}(t) := \frac{d\underline{z}(t)}{dt}\ ,\ \underline{v}(t) := \frac{d\underline{\eta}(t)}{dt}$$

and the Wiener process $\{\underline{\eta}(t)\}$ with

$$E\{d\underline{\eta}(t)\} = \underline{0}\ ,\ E\{d\underline{\eta}(t)d\underline{\eta}^T(t)\} = \underline{R}(t)\cdot dt. \tag{10c}$$

It is assumed that $\{\underline{\beta}(t)\}$, $\{\underline{\eta}(t)\}$, and $\underline{x}(t_o)$ are uncorrelated respectively independent.

The mathematical model Eq. (9), (10) is the basis for the solution of the nonlinear filtering problem in continuous time. This model is imbedded into the general estimation problem which appears obviously by comparing Fig.3 with Fig.1.

Discrete model

In the time-discrete case we have the stochastic vector difference equation

$$\underline{x}(t_{k+1}) = \underline{\Phi}[\underline{x}(t_k);t_{k+1},t_k] + \underline{\Gamma}[\underline{x}(t_k),t_k]\underline{w}(t_k)\ ,\ k = 0,1,2,\ldots \tag{11a}$$

as mathematical model for the signal sequence respectively the nonlinear plant under consideration. The initial state $\underline{x}(t_o)$ is normally distributed with known mean $\hat{\underline{x}}(t_o)$ and covariance matrix $\underline{P}(t_o)$ or has any other distribution.

The nonlinear observation sequence is given by

$$\underline{y}(t_k) = \underline{h}[\underline{x}(t_k),t_k] + \underline{v}(t_k)\ ,\ k = 0,1,2,\ldots \tag{11b}$$

The dimensions of the vectors correspond to those of the continuous case. The noise $\{\underline{v}(t_k)\}$ and $\{\underline{w}(t_k)\}$ are white Gaussian sequences with zero mean and covariance matrices $\underline{R}(t_k)$ resp. $\underline{Q}(t_k)$; $\underline{w}(t_k)$, $\underline{v}(t_k)$ and $\underline{x}(t_o)$ are assumed uncorrelated resp. independent.

Continuous-discrete model

Now we have a combination of continuous process dynamics and discrete measurements. Thus our system model is given by Eq.(9b) and Eq.(11b).

2.3 The Fokker-Planck equation

This equation, which is equally known as Kolmogorov's forward equation is a first but important step towards the solution of the nonlinear filtering problem in continuous time. It describes the evolution of the transition probability density $p[\underline{x}(t), t|\underline{x}(t_o), t_o]$ of the Markov process generated by the Itô differential equation (9b), assuming that the initial probability density is known. The measurements Eq. (10b) are not yet considered.

The Fokker-Planck equation in its most simple form traces back to the work of Einstein /19/ on Brownian motion in 1905. Further research on diffusion processes and the derivation of the corresponding partial differential equation was given by Fokker /20/ 1914, Planck /21/ 1917 and Kolmogorov /22/ in 1931.

This differential equation reads

$$\frac{\partial p}{\partial t} = - \sum_{i=1}^{n} \frac{\partial [p f_i]}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^{n} \frac{\partial^2 [p (\underline{G}\, \underline{Q}\, \underline{G}^T)_{ij}]}{\partial x_i \partial x_j} \tag{12}$$

with $p := p\,[\underline{x}(t),\ t\,|\underline{x}(t_o),\ t_o\,]$

and a delta impulse as initial condition for the density, which means that the initial value $\underline{x}(t_o) = \underline{\xi}$ is given.
With the introduction of the so called forward diffusion operator

$$\mathcal{L}(\cdot) := - \sum_{i=1}^{n} \frac{\partial [(\cdot) f_i]}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^{n} \frac{\partial^2 [(\cdot) (\underline{G}\, \underline{Q}\, \underline{G}^T)_{ij}]}{\partial x_i \partial x_j} \tag{13}$$

of the diffusion process $(\underline{x}(t),\ t \geq t_o)$ generated by the Itô differential equation (9b), the Fokker-Planck equation can be written as

$$\frac{\partial p}{\partial t} = \mathcal{L}(p) \ . \tag{14}$$

An analytical solution of this equation is possible only in a few simple cases where the process model is linear, i. e.

$$\underline{f}\,[\underline{x}(t),\ t\,] = \underline{F}(t) \cdot \underline{x}(t)$$

and the matrix $\underline{G}$ in Eq.(9b) is independent of the state vector $\underline{x}(t)$.

Example: To apply the Fokker-Planck equation, we demonstrate the transition of the probability density function for a simple example taken from /24/.

Given a first order linear process model as indicated in Fig.5. We are looking for the evolution of $p\,[x(t),\ t\,|x(t_o)=x_o]$ as solution of the Fokker-Planck equation.
Eq.(12) has the form

$$\frac{\partial p}{\partial t} = - \frac{\partial [p(-ax)]}{\partial x} + \frac{1}{2} Q \frac{\partial^2 p}{\partial x^2} \ . \tag{15}$$

Now, the Wiener process is Gaussian, and as we know normally distributed processes remain normally distributed while passing through a linear system; thus, the Gaussian assumption for the transition probability $p_{x(t)}(\xi,t|x_o)$ is straight forward. This yields from Eq.(15) two ordinary differential equations for the propagation of mean and variance with the solution

$$\bar{x} = x_o \exp[-a(t-t_o)] \ , \tag{16a}$$

$$\sigma_x^2(t) := \mathrm{var}(x) = \frac{Q}{2a} \left(1 - \exp[-2a(t-t_o)]\right) \ . \tag{16b}$$

Fig.5 illustrates the evolution of the corresponding transition probability density. The time history of the mean value $\bar{x}$ describes the characteristic motion of the system while the variance increases due to the diffusion from zero to $\sigma_x^2(t \to \infty) = Q/2a$.

2.4 The Kushner-Stratonovich equation

This equation represents the theoretical solution of the nonlinear filtering problem in continuous time.
Now the system model Eq.(9b) is considered with the observation Eq.(10b) included and the equation of evolution for the probability density $p\,[\underline{x},t\,|\underline{z}_t\,]$ of the state $\underline{x}$ conditioned on a realization $\underline{z}_t := [\underline{z}(\tau);\ t_o \leq \tau \leq t]$ of the observation process is looked for.
The first derivation of this equation was given by Stratonovich /15/ in 1960, who unfortunately was slightly errornous by omitting some terms. Kushner /25/, /16/ obtained the exact equation in 1964. Bucy /17/ developed the same equation in 1965 using the so called representation theorem.

The conditional probability density of the state given the observations satisfies the Kushner-Stratonovich equation

$$p[\underline{x},t+dt|\underline{Z}_t,d\underline{z}(t)] - p[\underline{x},t|\underline{Z}_t] = dp = \mathcal{L}(p)dt+[\underline{h}-\hat{\underline{h}}]^T\underline{R}^{-1}(d\underline{z}-\hat{\underline{h}}\,dt)p \tag{17}$$

This is a stochastic partial differential equation; it is stochastic because of the Wiener process $\{\underline{\eta}(t)\}$ contained in the differential observation $d\underline{z}$ and partial due to its Fokker-Planck part. Analytical solutions of this equation practically do not exist but it can be used for deriving the exact equations for the moments of the probability density; this is for instance important to obtain minimum variance estimates.

In view of Eq.(17) the principal behavior of the nonlinear estimator may be summarized as follows:

- The conditional probability density $p[\underline{x},t|\underline{Z}_t]$ changes as a result of the dynamics of the process model and due to the observations.
- The measurement information is used in form of the residual $(d\underline{z}-\hat{\underline{h}}dt)$. This residual is weighted with the matrix $[\underline{h}-\hat{\underline{h}}]^T\underline{R}^{-1}$. In case of stronger noise ($\|\underline{R}\|$ large) we have a weaker influence of the measurements. If the measurements are useless ($\underline{R}^{-1}\to\underline{0}$) we obtain prediction according to the Fokker-Planck equation.
- The estimator is nonlinear because the nonlinear vectors $\underline{h} = \underline{h}(\underline{x})$ and $\underline{f}(\underline{x})$ as well as $\underline{G}(\underline{x})$ require nonlinear processing of the measurements.

2.5 Minimum variance estimation

With regard to the realization of practical nonlinear filters we are interested in characteristic values of the conditional probability density function $p[\underline{x},t|\underline{Z}_t]$.

As mentioned in paragraph 2.1 the minimum variance estimate is the conditional mean which represents the first moment of the conditional density. Thus we are looking for the equation of evolution of

$$\hat{\underline{x}}(t_1|t) := E(\underline{x}(t_1)|\underline{Z}_t) \;;\; t_1 \geq t \;. \tag{18a}$$

In addition, we need the second moment, the conditional covariance matrix

$$\underline{P}(t_1|t) := E([\underline{x}(t_1)-\hat{\underline{x}}(t_1|t)][\underline{x}(t_1)-\hat{\underline{x}}(t_1|t)]^T|\underline{Z}_t) = E(\tilde{\underline{x}}^T(t_1|t)\tilde{\underline{x}}(t_1|t)) \tag{18b}$$

where

$$\tilde{\underline{x}}(t_1|t) := \underline{x}(t_1)-\hat{\underline{x}}(t_1|t) \tag{18c}$$

is the estimation error. This is because the matrix $\underline{P}$ is a measure of the accuracy of the estimates, since the mean of $\underline{P}$

$$E(\underline{P}(t_1|t)) = E(E(\tilde{\underline{x}}(t_1|t)\tilde{\underline{x}}^T(t_1|t)|\underline{Z}_t)) = E(\tilde{\underline{x}}(t_1|t)\tilde{\underline{x}}^T(t_1|t)) \tag{18d}$$

contains the variances of the components of the estimation error vector (18c) as diagonal elements.

Kushner /16/ proposed in 1964 to develop equations for the moments of the conditional density, which would yield a system of ordinary stochastic differential equations instead of the partial stochastic differential equation (17). Bucy /17/ obtained these equations for the first and second moment in the scalar case and Bass, Norum and Schwartz /26/ treated the general vector case in 1966.

The conditional mean (i. e. the minimum variance estimate) and the conditional covariance matrix satisfy the ordinary stochastic differential equations

$$d\hat{\underline{x}}(t) = \hat{\underline{f}}[\underline{x}(t),t]dt+[\widehat{\underline{x}(t)\underline{h}^T}-\hat{\underline{x}}(t)\hat{\underline{h}}^T]\underline{R}^{-1}(t)(d\underline{z}(t)-\hat{\underline{h}}dt) \tag{19a}$$

$$[d\underline{P}(t|t)]_{ij} = [(\widehat{f_i x_j}-\hat{f}_i\hat{x}_j)+(\widehat{x_i f_j}-\hat{x}_i\hat{f}_j)+(\widehat{\underline{G}\,\underline{Q}\,\underline{G}^T})_{ij}-(\widehat{x_i\underline{h}}-\hat{x}_i\hat{\underline{h}})^T\underline{R}^{-1}(t)(\widehat{\underline{h}x_j}-\hat{\underline{h}}\hat{x}_j)]dt + \tag{19b}$$

$$+[\widehat{x_i x_j\underline{h}}-\widehat{x_i x_j}\hat{\underline{h}}-\hat{x}_i\widehat{x_j\underline{h}}-\hat{x}_j\widehat{x_i\underline{h}}+2\hat{x}_i\hat{x}_j\hat{\underline{h}}]^T\underline{R}^{-1}(t)\cdot(d\underline{z}(t)-\hat{\underline{h}}dt)$$

with given initial conditions

$$\hat{\underline{x}}(t_o),\ \underline{P}(t_o|t_o) = \underline{P}(t_o);$$

and the more simple notation

$$\hat{\underline{x}}(t) := \hat{\underline{x}}(t|t).$$

For technical realization of a minimum variance estimator we need, however, further approximations (which will be discussed in para.3) since Eq.(19) requires knowledge of the whole conditional density; this is quite obvious because the expectation operations in (19) e. g. for the $\underline{f}$ vector are defined by

$$\hat{\underline{f}}(\underline{x}) := E\{\underline{f}(\underline{x}) \mid \underline{Z}_t\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \underline{f}(\underline{x}) p[\underline{x}(t), t|\underline{Z}_t]\, dx_1 \ldots dx_n \qquad (20)$$

2.6 The linear minimum variance estimator

We now consider as a special case a linear system model which we may write like Eq.(9a), (10a) using white noise instead of the Wiener process, in the form

$$\dot{\underline{x}}(t) = \underline{F}(t)\,\underline{x}(t) + \underline{G}(t)\underline{w}(t)\ ,\quad t \geq t_o \qquad (21a)$$

$$y(t) = \underline{H}(t)\,\underline{x}(t) + v(t)\ ,\quad t \geq t_o \qquad (21b)$$

The optimal minimum variance estimator for this system model which is easily obtained from Eq.(19), is given by

$$\frac{d\hat{\underline{x}}}{dt} = \underline{F}(t)\hat{\underline{x}}(t) + \underline{K}(t)\{\underline{y}(t) - \underline{H}(t)\hat{\underline{x}}(t)\} \qquad (22a)$$

$$\frac{d\underline{P}(t)}{dt} = \underline{F}(t)\underline{P}(t) + \underline{P}(t)\underline{F}^T(t) + \underline{G}(t)\underline{Q}(t)\underline{G}^T(t) - \underline{P}(t)\underline{H}^T(t)\underline{R}^{-1}(t)\underline{H}(t)\underline{P}(t)\ , \qquad (22b)$$

$$\underline{K}(t) := \underline{P}(t)\underline{H}^T(t)\underline{R}^{-1}(t)\ ;\quad t \geq t_o\ , \qquad (22c)$$

initial conditions $\underline{x}(t_o)$, $\underline{P}(t_o)$.

These are the equations of the well known Kalman-Bucy filter which goes back to Kalman and Bucy /10/, 1961 in the continuous case and Kalman / 9/, 1960 in the discrete case.

It is not our purpose to discuss the optimal linear estimator in this contribution in more detail because our main objective is nonlinear estimation. However, we will give an impression of the structure of this linear filter.

This will be used later as a reference for the understanding of the local nonlinear filter approximations.

Fig.6 shows the signal flow of the data processing in the Kalman-Bucy filter. It has the structure of a control loop where the "plant" equals the system model under consideration. The controller is of (matrix) proportioanl type ($\underline{K}(t)$) and updates the system model in the computer using the residuals ($\underline{y}-\hat{\underline{y}}$). The time variable Kalman gain $\underline{K}(t)$ is obtained as solution of the matrix Riccati differential equation (22b) for $\underline{P}$, which may be solved off-line since $\underline{P}$ does neither depend on the observations nor on the estimate $\hat{\underline{x}}$.

The stationary solution of this equation yields a constant gain for the filter which is then equivalent to the Wiener filter /7/, /8/.

Finally it should be emphasized that the linear minimum variance filter is the exact solution of the general estimation problem provided that

- the system model is linear

- $\underline{x}(t_o)$ and the white noise $\underline{w}(t)$, $\underline{v}(t)$ are normally distributed and uncorrelated.

In this case, the conditional density $p[\underline{x},t \mid \underline{Y}_t]$ is Gaussian which only requires first and second moments ($\hat{\underline{x}}$ and $\underline{P}$) for its representation. As we know Eq.(22) for $\hat{\underline{x}}$ and $\underline{P}$ contain no approximations, i. e. the evaluation of these moments is exact.

2.7 The Bayesian recursive estimation equations

The development of the recursion relations for the conditional probability density $p[\underline{x}(t_k) \mid \underline{Y}_{t_k}]$ in the time-discrete case is a much simpler task than in the continuous case.
This is due to the fact that we do not need stochastic integrals and stochastic differential equations but sums and difference equations, moreover, the discrete white noise is physically meaningful.

We consider the discrete system model, Eq.(11). The recursion equations for filtering (updating after measurements) and prediction (between measurements) can be obtained by using

Bayes' theorem of probability theory.

a) Filtering (calculation of the a posteriori density at $t=t_k$)

$$p[\underline{x}(t_k)|\underline{Y}_{t_k}] = \frac{1}{c(t_k)} \cdot p[\underline{y}(t_k)|\underline{x}(t_k)] \cdot p[\underline{x}(t_k)|\underline{Y}_{t_{k-1}}]$$

$$= \frac{1}{c(t_k)} \cdot p_{\underline{v}(t_k)}[\underline{y}(t_k)-\underline{h}[\underline{x}(t_k)]] \cdot p[\underline{x}(t_k)|\underline{Y}_{t_{k-1}}] \quad , \; k = 0,1,2,... \tag{23a}$$

normalization:

$$c(t_k) := p[\underline{y}(t_k)|\underline{Y}_{t_{k-1}}] = \int_{-\infty}^{\infty} p_{\underline{v}(t_k)}[\underline{y}(t_k)-\underline{h}[\underline{x}(t_k)]] \cdot p[\underline{x}(t_k)|\underline{Y}_{t_{k-1}}] \underline{dx}(t_k) \tag{23b}$$

Initial conditions: $p[\underline{x}(t_0)|\underline{Y}_{t_{0-1}}] = p[\underline{x}(t_0)]$

b) Prediction (calculation of the a priori density in the absence of measurements)

$$p[\underline{x}(t_{k+1})|\underline{Y}_{t_k}] = \int_{-\infty}^{\infty} p[\underline{x}(t_{k+1})|\underline{x}(t_k)] \cdot p[\underline{x}(t_k)|\underline{Y}_{t_k}] \underline{dx}(t_k) \tag{23c}$$

$$= \int_{-\infty}^{\infty} p_{\underline{\Gamma}\underline{w}(\tau_k)}[\underline{x}(t_{k+1})-\underline{\varphi}[\underline{x}(t_k)]] \cdot p[\underline{x}(t_k)|\underline{Y}_{t_k}] \underline{dx}(t_k) .$$

Analytical solutions of Eq.(23) are available only for linear systems with Gaussian initial conditions and disturbances which yield the Kalman filter.

For the general nonlinear case we have to look for numerical solutions of these equations which will lead us to global approximations of nonlinear filters (see para.4). There, the main problem is the amount of storage capacity needed for storing of the multidimensional conditional probability densities and the calculation of the nonlinear convolution integrals in Eq.(23). A good survey on the methods of density approximation is given by Sorenson /27/ in 1974.

Eq.(23) can, however, be used to develop the exact equations of the conditional mean $\hat{\underline{x}}$ and covariance matrix $\underline{P}$ for the nonlinear minimum variance estimator.

2.8 Continuous-discrete estimation

This mode of estimation is of special importance because in many applications the underlying mathematical model of the system has continuous dynamics while the observations are usually taken at discrete time instants.

The system model for continuous-discrete estimation is given by Eq.(9b) and Eq.(11b). Obviously, the filtering equation is Eq.(23a) of the discrete estimation problem and prediction over one time interval is accomplished by the Fokker-Planck equation (12).

3. LOCAL APPROXIMATIONS OF NONLINEAR FILTERS

After having available the exact solution of the general estimation problem, we are now looking for practical recursive algorithms which can be directly implemented on a computer to give us on-line estimates of the state vector $\underline{x}$.

For this purpose, we have to approximate the density functions as well as the nonlinear system functions.

One frequently used approach to these approximations is carried out in the following steps:

i) approximation of the conditional probability density function $p[\underline{x}(t),t \mid Z_t]$ by the moments of the distribution

ii) consideration of the first and second moment ($\hat{\underline{x}}$ and $\underline{P}$), i. e. minimum variance estimation, and neglecting or approximation of higher order moments

iii) approximation of the nonlinear system functions $\underline{f}[\underline{x}(t), t]$, $\underline{h}[\underline{x}(t), t]$, $\underline{G}[\underline{x}(t),t]$ by series expansion around appropriate reference points resp. trajectories.

This approximation is called local approximation since the resulting filtering algorithms are only applicable in the surrounding of the reference point.

3.1 First order filter approximations

Obviously first order approximations are the simplest ones and therefore most often used in practical applications. In the sequel we will present some of the more important filter algorithms which have been developed since the beginning of the sixties.

3.1.1 The continuous minimum variance filter (extended Kalman-Bucy filter)

The underlying system model is given by the (Itô) stochastic differential equations (9b) and (10b). We search for an approximation for the (exact) minimum variance filter equations (19) assuming

i) the conditional probability density $p[\underline{x}(t),t \mid \underline{Z}_t]$ is almost symmetric and concentrated near its mean. Thus odd central moments are negligible.

ii) The system functions $\underline{f}(\underline{x})$, and $\underline{G}(\underline{x})\cdot\underline{Q}\cdot\underline{G}^T(\underline{x})$ are expanded in Taylor series around the conditional mean $\hat{\underline{x}}$ up to terms of first order.

This expansion e. g. for $\underline{f}(\underline{x})$ yields

$$\underline{f}[\underline{x}(t),t] = \underline{f}[\hat{\underline{x}}(t),t] + \left.\frac{\partial \underline{f}[\underline{x}(t),t]}{\partial \underline{x}(t)}\right|_{\underline{x}=\hat{\underline{x}}} \cdot [\underline{x}(t)-\hat{\underline{x}}(t)] + \ldots \tag{24}$$

and, taking the conditional expectation we obtain

$$\hat{\underline{f}}[\underline{x}(t),t] := E(\underline{f}[\underline{x}(t),t] \mid \underline{Z}_t) \cong \underline{f}[\hat{\underline{x}}(t),t] \tag{25a}$$

since the first central moment $E\{\underline{x}-\hat{\underline{x}} \mid \underline{Z}_t\}$ is zero due to assumption i.

in a similar way we obtain

$$\hat{\underline{h}}[\underline{x}(t),t] \approx \underline{h}[\hat{\underline{x}}(t),t] \tag{25b}$$

and

$$\widehat{[\underline{G}\,\underline{Q}\,\underline{G}^T]} \cong \underline{G}[\hat{\underline{x}}(t),t]\underline{Q}(t)\underline{G}^T[\hat{\underline{x}}(t),t] \ . \tag{25c}$$

Bearing in mind that

$$\widehat{[\underline{x}\,\underline{h}^T-\hat{\underline{x}}\,\hat{\underline{h}}^T]} = \widehat{[(\underline{x}-\hat{\underline{x}})\underline{h}^T]} \approx \widehat{[\tilde{\underline{x}}\ \underline{h}^T(\hat{\underline{x}})]} + \left[\tilde{\underline{x}}\ \tilde{\underline{x}}^T\left(\frac{\partial \underline{h}}{\partial \hat{\underline{x}}}\right)^T\right]^{\wedge} = \underline{0} + \underline{P}(t|t)\left(\frac{\partial \underline{h}}{\partial \hat{\underline{x}}}\right)^T \tag{25d}$$

we immediately obtain with Eq.(25) the differential equation for $\hat{\underline{x}}$ from Eq.(19a); and similarly the equation for the covariance matrix $\underline{P}$, taking into account that $\underline{P}(t|t)$ does not depend immediately on the observations (the factor before $[d\underline{z}-\hat{\underline{h}}dt]$ in Eq.(19b) becomes zero). As a consequence, the conditional covariance matrix $\underline{P}(t|t)$ equals the unconditional matrix $\underline{P}(t)$.

Thus we have the equations for the first order minimum variance filter

$$d\hat{\underline{x}}(t) = \underline{f}[\hat{\underline{x}}(t),t]dt + \underline{K}(t)[d\underline{z}(t) - \underline{h}[\hat{\underline{x}}(t),t]dt] \tag{26a}$$

$$\frac{d\underline{P}(t)}{dt} = \frac{\partial \underline{f}[\hat{\underline{x}}(t),t]}{\partial \hat{\underline{x}}(t)}\underline{P}(t) + \underline{P}(t)\frac{\partial \underline{f}^T[\hat{\underline{x}}(t),t]}{\partial \hat{\underline{x}}(t)} - \underline{P}(t)\frac{\partial \underline{h}^T[\hat{\underline{x}}(t),t]}{\partial \hat{\underline{x}}(t)}\underline{R}^{-1}(t)\frac{\partial \underline{h}[\hat{\underline{x}}(t),t]}{\partial \hat{\underline{x}}(t)}\underline{P}(t) + \underline{G}[\hat{\underline{x}}(t),t]\underline{Q}(t)\underline{G}^T[\hat{\underline{x}}(t),t] \tag{26b}$$

$$\underline{K}(t) = \underline{P}(t)\frac{\partial \underline{h}^T[\hat{\underline{x}}(t),t]}{\partial \hat{\underline{x}}(t)}\underline{R}^{-1}(t) \ . \tag{26c}$$

initial cond.: $\hat{\underline{x}}(t_0)$, $\underline{P}(t_0)$; $t \geq t_0$.

For technical realization, both differential equations have to be solved on-line since the gain matrix $\underline{K}$ depends on the conditional mean $\hat{\underline{x}}$. This is different from the (linear) Kalman-Bucy filter where $\underline{K}$ can be calculated off-line without taking any observation.

Apart from this, the first order minimum variance filter is very similar to the Kalman-Bucy filter which is easy to see by comparing the block diagrams of both filters, Fig.7 and Fig.6.

Actually, the filter equations for the minimum variance filter (26) can be gained using the error model of the system equations (9b), (10b) and the linear Kalman-Bucy filter equations (22). The error $\delta\underline{x}(t)$ is defined as the derivation of the state vector $\underline{x}$ from the conditional mean $\hat{\underline{x}}$, i. e.

$$\delta\underline{x}(t) := \underline{x}(t) - \hat{\underline{x}}(t) \tag{27}$$

Now we have with Eq. (9b)

$$d[\delta\underline{x}(t)] = d[\underline{x}(t)-\hat{\underline{x}}(t)] = (\underline{f}[\underline{x}(t),t]-\underline{f}[\hat{\underline{x}}(t),t])dt + \underline{G}[\underline{x}(t),t]d\underline{\beta}(t)$$

and in view of Eq.(24) and (9a) with $\underline{G}(\underline{x}) \approx \underline{G}(\hat{\underline{x}})$ we may write

$$\dot{\delta\underline{x}}(t) = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{x}=\hat{\underline{x}}} \cdot \delta\underline{x}(t) + \underline{G}[\hat{\underline{x}}(t),t]\,\underline{w}(t) \tag{28a}$$

which in fact is a linear stochastic differential equation for the error $\delta\underline{x}$. In an analogous way we obtain the linearized measurement equation

$$\delta y(t) = \left.\frac{\partial \underline{h}}{\partial \underline{x}}\right|_{\underline{x}=\hat{\underline{x}}} \delta\underline{x}(t) + \underline{v}(t). \tag{28b}$$

Now the Kalman-Bucy filter equations (22) are directly applicable to the error model (28) and, taking into account that

$$\delta\hat{\underline{x}} = E\{\underline{x}-\hat{\underline{x}} \mid t\} = \underline{0}$$

We obtain equations (26), the first order minimum variance filter.

Because of this possibility of developing the minimum variance filter by an extension of the linear filter theory to a linearized nonlinear system, this filter is also well known as <u>extended</u> or <u>modified Kalman-Bucy filter.</u>

3.1.2 The linearized Kalman-Bucy filter

The assumptions i and ii for the approximation are the same as for the minimum variance filter with the exception that the linearization of $\underline{f}(\underline{x})$, $\underline{h}(\underline{x})$ and $\underline{G}(\underline{x})\,\underline{Q}\,\underline{G}^T(\underline{x})$ is now carried out about a known nominal trajectory $\bar{\underline{x}}(t)$. This trajectory can be specified by the differential equation.

$$d\bar{\underline{x}}(t) = \underline{f}[\bar{\underline{x}}(t),t]dt,\ \bar{\underline{x}}(t_o) = \hat{\underline{x}}(t_o);\ t \geq t_o\ . \tag{29}$$

If the first order approximation of the deviation from the nominal trajectory

$$\delta\underline{x}(t) = \underline{x}(t) - \bar{\underline{x}}(t) \tag{30}$$

is small enough we obtain an appropriate linear system model for this deviation in accordance with Eq.(28) and hence by application of the Kalman-Bucy filter (Eq.(22))the equations of <u>the linearized Kalman-Bucy filter</u>.

$$d[\delta\hat{\underline{x}}(t)] = \underline{F}(t)\delta\hat{\underline{x}}(t)\cdot dt + \underline{K}(t)\{\delta\underline{y}(t) - \underline{H}(t)\delta\hat{\underline{x}}(t)\}dt \tag{31a}$$

$$\frac{d\underline{P}(t)}{dt} = \underline{F}(t)\underline{P}(t) + \underline{P}(t)\underline{F}^T(t) - \underline{P}(t)\underline{H}^T(t)\underline{R}^{-1}(t)\underline{H}(t)\underline{P}(t) + \underline{G}(t)\underline{Q}(t)\underline{G}^T(t) \tag{31b}$$

$$\underline{K}(t) = \underline{P}(t)\underline{H}^T(t)\underline{R}^{-1}(t) \tag{31c}$$

$$\hat{\underline{x}}(t) = \bar{\underline{x}}(t) + \delta\hat{\underline{x}}(t) \tag{31d}$$

initial cond.: $\delta\hat{\underline{x}}(t_o) = 0,\ \underline{P}(t_o),\ t \geq t_o$

definitions:

$$\underline{F}(t) := \left.\frac{\partial \underline{f}[\underline{x}(t),t]}{\partial \underline{x}(t)}\right|_{\underline{x}(t)=\bar{\underline{x}}(t)}$$

$$\underline{H}(t) := \left.\frac{\partial \underline{h}[\underline{x}(t),t]}{\partial \underline{x}(t)}\right|_{\underline{x}(t)=\bar{\underline{x}}(t)}$$

$$\underline{G}(t) := \underline{G}[\bar{\underline{x}}(t),t]$$

Remarks:

i) Since the nominal trajectory $\bar{\underline{x}}(t)$ is usually not a constant, the linearized Kalman-Bucy filter is time variable

ii) The filter gain matrix $\underline{K}(t)$, however, can now be calculated beforehand (off-line) without taking observations. This is the difference to the extended Kalman-Bucy filter where we need as a reference for the linearization the conditional mean $\hat{\underline{x}}(t)$ which we only obtain by on-line filtering.

iii) The filter is useful especially in aerospace applications where the nominal trajectory $\bar{\underline{x}}(t)$ is often known. In general, it is not a simple task to define this trajectory properly. However, if the deviation $\delta\underline{x}$ becomes too large the linearization conditions are no longer valid and useless estimates result.

3.1.3 The continuous-discrete extended Kalman-Bucy filter

Because of the practical use of the continuous-discrete mode of estimation we will now present the estimation equations for this filter.

The system model consists of the (continuous) nonlinear stochastic differential equation (9b) and the (discrete) observation equation (11b).

The development of this estimator is straight-forward using the well-known discrete Kalman filter equations (which we did not give in this contribution) for updating (filtering) of the estimate after an observation on the one hand and the continuous extended Kalman-Bucy filter for prediction on the other hand.
The latter equations are obtained by setting $\underline{R}^{-1}$ in Eq.(26a), (26b) equal to zero. Thus summing up, we have the equations of the continuous-discrete extended Kalman-Bucy filter

i) Filtering (updating at the instant t_k of an observation)

$$\hat{\underline{x}}(t_k|t_k) = \hat{\underline{x}}(t_k|t_{k-1}) + \underline{K}(t_k)\left(\underline{y}(t_k) - \underline{h}[\hat{\underline{x}}(t_k|t_{k-1}), t_k]\right) \tag{32a}$$

$$\underline{P}(t_k|t_k) = [\underline{I} - \underline{K}(t_k)\underline{H}(t_k)]\underline{P}(t_k|t_{k-1}) \tag{32b}$$

$$\underline{K}(t_k) = \underline{P}(t_k|t_{k-1})\underline{H}^T(t_k)\cdot \left[\underline{H}(t_k)\underline{P}(t_k|t_{k-1})\underline{H}^T(t_k) + \underline{R}(t_k)\right]^{-1} = \underline{P}(t_k|t_k)\underline{H}^T(t_k)\underline{R}^{-1}(t_k) \qquad k = 0,1,2,\ldots \tag{32c}$$

initial cond.: $\hat{\underline{x}}(t_0|t_{0-1}) := \hat{\underline{x}}(t_0)$;

$\underline{P}(t_0|t_{0-1}) := \underline{P}(t_0)$

definitions: $\underline{H}(t_k) := \left.\frac{\partial \underline{h}[\underline{x}(t_k), t_k]}{\partial \underline{x}(t_k)}\right|_{\underline{x}=\hat{\underline{x}}(t_k|t_{k-1})}$;

$\underline{h}[\hat{\underline{x}}(t_k|t_{k-1}), t_k]$;

$\underline{K}(t_k) := \underline{K}[\hat{\underline{x}}(t_k|t_{k-1}), t_k]$

ii) Prediction (between obaservations $t_k \leq t < t_{k+1}$)

$$\frac{d\hat{\underline{x}}(t|t_k)}{dt} = \underline{f}[\hat{\underline{x}}(t|t_k), t] \tag{33a}$$

$$\hat{\underline{x}}(t_{k+1}|t_k) = \hat{\underline{x}}(t_k|t_k) + \int_{t_k}^{t_{k+1}} \underline{f}[\hat{\underline{x}}(t|t_k), t]\,dt$$

$$\frac{d\underline{P}(t|t_k)}{dt} = \underline{F}(t)\underline{P}(t|t_k) + \underline{P}(t|t_k)\underline{F}^T(t) + \underline{G}[\hat{\underline{x}}(t|t_k), \cdot]\cdot\underline{Q}(t)\underline{G}^T[\hat{\underline{x}}(t|t_k), t] \tag{33b}$$

definition: $\underline{F}(t) := \left.\frac{\partial \underline{f}[\underline{x}(t), t]}{\partial \underline{x}(t)}\right|_{\underline{x}=\hat{\underline{x}}(t|t_k)}$;

$\underline{F}[\hat{\underline{x}}(t|t_k), t]$

3.1.4 The iterated extended Kalman-Bucy filter

There are several possibilities of improving the estimates of the extended Kalman-Bucy filter by local iterations. The basic idea for these algorithms is the reduction of the estimation errors by an iterative re-linearization of the system nonlinearities $\underline{h}(\underline{x})$ and/or $\underline{f}(\underline{x})$ about improved reference values.

We will demonstrate this for the measurement nonlinearity $\underline{h}(\underline{x})$.

In the filtering equations (32) of the continuous-discrete extended Kalman-Bucy filter the reference $\hat{\underline{x}}$ for the evaluation of $\underline{h}$ $(\hat{\underline{x}})$ and the Jacobian matrix $\underline{H}(\underline{x})$ is the predicted state vector $\hat{\underline{x}}(t_k|t_{k-1})$ based on the information of the observation $\underline{y}(t_{k-1})$ at the instant t_{k-1}. In view of the filtering equation for the linearized Kalman-Bucy filter with reference value $\underline{\bar{x}}(t_k)$ (see e. g. /24/ p.183) we may write the corresponding equation (32a)

of the extended Kalman-Bucy filter in the form

$$\hat{\underline{x}}(t_k|t_k) = \hat{\underline{x}}(t_k|t_{k-1}) + \underline{K}[\hat{\underline{x}}(t_k|t_{k-1}),t_k]\cdot\{\underline{y}(t_k) - \underline{h}[\hat{\underline{x}}(t_k|t_{k-1}),t_k] - \underbrace{\underline{H}[\hat{\underline{x}}(t_k|t_{k-1}),t_k]\cdot[\hat{\underline{x}}(t_k|t_{k-1}) - \hat{\underline{x}}(t_k|t_{k-1})]}_{:=\bar{\underline{x}}(t_k)}\}}_{=\underline{0}} \tag{34}$$

Now, the updated estimate $\hat{\underline{x}}(t_k|t_k)$ on the average will be better (i. e. closer to the actual value $\underline{x}(t_k)$) than the predicted estimate $\hat{\underline{x}}(t_k|t_{k-1})$. Hence we relinearize about $\hat{\underline{x}}(t_k|t_k)$ and will obtain an improved updated estimate $\hat{\underline{x}}^{(2)}(t_k|t_k)$ by processing the observation $\underline{y}(t_k)$ once again through the filtering equation (34) with $\bar{\underline{x}}(t_k):=\hat{\underline{x}}(t_k|t_k)$. For this purpose the gain matrix $\underline{K}$ (Eq.(32c)) has to be recomputed as well using $\hat{\underline{x}}(t_k|t_k)$.

The iteration can then be repeated by linearization about $\hat{\underline{x}}^{(2)}(t_k|t_k)$ which yields $\hat{\underline{x}}^{(3)}(t_k|t_k)$. Generally the iteration procedure will be terminated when a further improvement is not possible, i. e.

$$\left|\hat{\underline{x}}^{(i)}(t_k|t_k) - \hat{\underline{x}}^{(i-1)}(t_k|t_k)\right| \leq \varepsilon_y \to \underline{0} \,. \tag{35}$$

Hence we sum up the equations of the iterated extended Kalman-Bucy filter with iteration of the measurement model.

i) Filtering (updating at the instant t_k of an observation $y(t_k)$)

$$\hat{\underline{x}}^{(i+1)}(t_k|t_k) = \hat{\underline{x}}(t_k|t_{k-1}) + \underline{K}[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k]\cdot \cdot[\underline{y}(t_k) - \underline{h}[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k] - \underline{H}[\hat{\underline{x}}^{(i)}(t_k|t_k)]\cdot \cdot[\hat{\underline{x}}(t_k|t_{k-1}) - \hat{\underline{x}}^{(i)}(t_k|t_k)]] \,, \tag{36a}$$

$$i = 1,\ldots,(i-1);\quad \hat{\underline{x}}^{(1)}(t_k|t_k) := \hat{\underline{x}}(t_k|t_{k-1})$$

$$k = 0,1,2,\ldots \qquad \hat{\underline{x}}^{(1)}(t_k|t_k) := \hat{\underline{x}}(t_k|t_k)$$

$$\underline{K}[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k] = \underline{P}(t_k|t_{k-1})\underline{H}^T[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k]\cdot \cdot[\underline{H}[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k]\underline{P}(t_k|t_{k-1})\cdot \cdot \underline{h}^T[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k] + \underline{R}(t_k)]^{-1} \tag{36b}$$

Covariance matrix: evaluation only once after having terminated the iterations, i. e. Eq.(32b) with $\hat{\underline{x}}^{(i)}(t_k|t_k)$

$$\underline{P}(t_k|t_k) = [\underline{I} - \underline{K}[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k]\underline{H}[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k]]\cdot \cdot \underline{P}(t_k|t_{k-1}) \tag{36c}$$

definition: $$\underline{H}[\hat{\underline{x}}^{(i)}(t_k|t_k),t_k] := \left.\frac{\partial \underline{h}[\underline{x}(t_k),t_k]}{\partial \underline{x}(t_k)}\right|_{\underline{x}=\hat{\underline{x}}^{(i)}(t_k|t_k)}$$

ii) Prediction (between observations $t_k \leq t < t_{k+1}$) Eq.(33) holds, starting the integration with the final iterated filter estimates $\hat{\underline{x}}^{(i)}(t_k|t_k)$ from Eq.(36a) and $\underline{P}(t_k|t_k)$ from Eq.(36c).

The iterated filter considerably reduces the influence of the measurement nonlinearity $\underline{h}(\underline{x})$ on the estimation quality. This was demonstrated by Denham and Pines /32/ in 1966.

If the system nonlinearity $\underline{f}(\underline{x})$ is, however, strongly nonlinear, further improvement may be achieved by iterating the prediction equations as well. The resulting estimator was first developed by Wishner, Tabaczynski, and Athans /29/ in 1968; it is known as "single stage iteration filter", while Jazwinski called it "iterated linear filter smoother" (/23/, p.280). The discrete-time version of this estimator is called discrete conditional-mean iteration-filter (Sage and Melsa /30/, p.470). Usually, these iterated filter algorithms converge very fast: the main improvement is already often reached after one or two iterations /29/. Thus, they represent a very useful tool for nonlinear filter applications.

3.2 Higher order filter approximations

As already pointed out, first order approximations have the advantage of a comparatively small amount of computational burden . Moreover, these filters can be developed by an extension of the linear filtering theory. Thus, deeper insight in the treatment of stochastic differential equations (e. g. the Itô stochastic calculus) is not necessarily required. As a consequence - with regard to nonlinear filter approximations - the extended Kalman filter type estimators have found wide spread applications.

On the other hand, smaller estimation errors may result if second partial derivatives of the nonlinear system functions $\underline{f}(\underline{x})$, $\underline{h}(\underline{x})$ are included in the estimation algorithms.

There exist a lot of different higher order approximation filters; as an example we will give in the sequel two important types, the continuous second-order minimum variance filter and the continuous-discrete modified Gaussian second-order filter.

3.2.1 The continuous second-order minimum variance filter

This filter was developed by Bass, Norum and Schwartz /26/ and independently by Jazwinski /31/ in 1966.

The system model is again Eq.(9b), (10b).

Assumptions:

i) The conditional probability density function $p\,[\underline{x}(t),\ t \mid \underline{z}_t]$ is almost symmetric and concentrated near its mean. Thus, odd central moments are negligible.

ii) The system functions $\underline{f}(\underline{x})$, $\underline{h}(\underline{x})$ and $\underline{G}(\underline{x})\,\underline{Q}\,\underline{G}^T(\underline{x})$ are expanded in Taylor series around the conditional mean up to terms of second order.

The assumption i is the same as for the first order minimum variance filter. However, even central moments of fourth order which will appear now (due to the second order series expansion) are neglected, i. e.

$$\left[(x_i-\hat{x}_i)(x_j-\hat{x}_j)(x_k-\hat{x}_k)(x_l-\hat{x}_l)\right]\hat{} \approx 0\ . \tag{37}$$

The expansion of the nonlinear functions and expectation operation e. g. for $\underline{f}(\underline{x})$ yields

$$\underline{f}\,[\underline{x}(t),t\,] \approx \underline{f}\,[\underline{x}(t),t] + \frac{1}{2}\,\frac{\partial^2 \underline{f}}{\partial \underline{x}^2} : \underline{P}(t|t) \tag{38}$$

with the definition

$$\frac{\partial^2 \underline{f}}{\partial \underline{x}^2} : \underline{P} = \begin{bmatrix} \mathrm{tr}\left[\frac{\partial^2 f_1}{\partial \underline{x}^2}\cdot \underline{P}\right] \\ \mathrm{tr}\left[\frac{\partial^2 f_2}{\partial \underline{x}^2}\cdot \underline{P}\right] \\ \vdots \\ \mathrm{tr}\left[\frac{\partial^2 f_n}{\partial \underline{x}^2}\cdot \underline{P}\right] \end{bmatrix} \tag{39a}$$

and the Hessian matrix

$$\frac{\partial^2 \varphi}{\partial \underline{x}^2} := \frac{\partial^2 \varphi}{\partial \underline{x}^2}\bigg|_{\underline{x}=\hat{\underline{x}}} = \left[\frac{\partial^2 \varphi}{\partial \hat{x}_i\,\partial \hat{x}_j}\right] \tag{39b}$$

we get after some calculation with Eq.(37) from Eq.(19a), (19b) the equations of the continuous second-order minimum variance filter.

$$d\underline{\hat{x}}(t) = \left[\underline{f}[\underline{\hat{x}}(t),t]+\frac{1}{2}\frac{\partial^2 \underline{f}[\underline{\hat{x}}(t),t]}{\partial \underline{\hat{x}}^2} : \underline{P}(t|t)\right]dt +$$

$$+\underline{K}(t)\left\{d\underline{z}(t)-\left[\underline{h}[\underline{\hat{x}}(t),t]+\frac{1}{2}\frac{\partial^2 \underline{h}[\underline{\hat{x}}(t),t]}{\partial \underline{\hat{x}}^2} : \underline{P}(t|t)\right]dt\right\} \qquad (40a)$$

$$d\underline{P}(t|t) = \left\{\underline{F}[\underline{\hat{x}}(t),t]\underline{P}(t|t)+\underline{P}(t|t)\underline{F}^T[\underline{\hat{x}}(t),t] -\right.$$

$$-\underline{P}(t|t)\underline{H}^T[\underline{\hat{x}}(t),t]\underline{R}^{-1}(t)\ \underline{H}[\underline{\hat{x}}(t),t]\underline{P}(t|t) +$$

$$+\underline{G}[\underline{\hat{x}}(t),t]\underline{Q}(t)\underline{G}^T[\underline{\hat{x}}(t),t] +$$

$$\left.+\frac{1}{2}\frac{\partial^2}{\partial \underline{\hat{x}}^2}\ (\underline{G}[\underline{\hat{x}}(t),t]\underline{Q}(t)\underline{G}^T[\underline{\hat{x}}(t),t]) : \underline{P}(t|t)\right\}dt - \qquad (40b)$$

$$-\frac{1}{2}\underline{P}(t|t)\left(\frac{\partial^2 \underline{h}[\underline{\hat{x}}(t),t]}{\partial \underline{\hat{x}}^2} : \underline{P}(t|t)\right)^T \underline{R}^{-1}(t)\cdot$$

$$\cdot\left\{d\underline{z}(t)-\left[\underline{h}[\underline{\hat{x}}(t),t] + \frac{1}{2}\frac{\partial^2 \underline{h}[\underline{\hat{x}}(t),t]}{\partial \underline{\hat{x}}^2} : \underline{P}(t|t)\right]dt\right\}$$

$$\underline{K}(t) = \underline{P}(t|t)\underline{H}^T[\underline{\hat{x}}(t),t]\underline{R}^{-1}(t) \qquad (40c)$$

initial cond.: $\underline{\hat{x}}(t_o)$, $\underline{P}(t_o|t_o) = \underline{P}(t_o)$, $t \geq t_o$

definition: $\underline{F} := \dfrac{\partial \underline{f}[\underline{\hat{x}}(t),t]}{\partial \underline{\hat{x}}}$, $\underline{H} := \dfrac{\partial \underline{h}[\underline{\hat{x}}(t),t]}{\partial \underline{\hat{x}}}$,

$\underline{K}(t) := \underline{K}[\underline{\hat{x}}(t),t]$.

Comparing this filter with the first-order filter (Eq.(26)) we see in which way the second order terms are now added for better approximation of the nonlinear functions $\underline{f}$ and $\underline{h}$. This is demonstrated also by Fig.8 in comparison with Fig.7. Moreover, the equation (40b) for the conditional covariance matrix $\underline{P}(t|t)$ is now a stochastic differential equation since it contains the observations.

3.2.2 The continuous-discrete modified Gaussian second-order filter

This filter was given by Jazwinski /32/,1966 and independently by Fisher /33/, 1967 and Athans, Wishner, Bertolini /34/ in 1968. It has the advantage that no stochastic differential equation is included in the estimation algorithm. Thus, it is easy to implement on a digital computer.

Now the continuous-discrete system model, Eq.(9b) and (11b) applies.

Assumptions:

i) The conditional probability density function $p\left[\underline{x}(t),t|\underline{Y}_{t_k}\right]$ is (approximately) Gaussian. Thus odd central moments are negligible and fourth central moments will be approximated by elements of the covariance matrix $\underline{P}$.

$$\left[(x_i-\hat{x}_i)(x_j-\hat{x}_j)(x_k-\hat{x}_k)(x_l-\hat{x}_l)|t_{n-1}\right]^{\wedge} =$$

$$= P_{jk}\cdot P_{il} + P_{jl}\cdot P_{ik} + P_{kl}\cdot P_{ij} \ . \qquad (41)$$

ii) The nonlinear system functions $\underline{f}(\underline{x})$, $\underline{h}(\underline{x})$ and $\underline{G}(\underline{x})$ resp. $\underline{G}(\underline{x})\ \underline{Q}\ \underline{G}^T(\underline{x})$ are expanded in Taylor series around the conditional mean $\underline{\hat{x}}(t|t_{k-1})$, $t_{k-1} \leq t \leq t_k$ up to terms of second order.

iii) The filter equations for $\hat{\underline{x}}(t_k|t_k)$ and $\underline{P}(t_k|t_k)$ at an observation will be developed by assuming a power series in $\hat{\underline{y}}(t_k|t_{k-1})$ which is carried to the first order only. The covariance equation will then be modified by omitting the immediate influence of the measurement $\underline{y}(t_k)$ on $\underline{P}(t_k|t_k)$, i. e. the second term in the power series for $\underline{P}(t_k|t_k)$ is dropped.

The result of assumption iii is an updating (filtering) equation of the form

$$\hat{\underline{x}}(t_k|t_k) = \underline{a}(t_k) + \underline{K}(t_k)\left[\underline{y}(t_k)-\hat{\underline{y}}(t_k|t_{k-1})\right]$$

$$\underline{P}(t_k|t_k) = \underline{C}(t_k). \tag{42}$$

These are in fact linear regression equations for the vector case, which yield the best minimum variance estimate $\hat{\underline{x}}(t_k|t_k)$.

The general formulation for $\underline{a}$, $\underline{K}$ and $\underline{C}$ can be found in Farison /35/. From Eq.(42) and /35/ we finally obtain after some calculation the filtering equations of the modified Gaussian second-order filter, making use of Eq.(38), (39) and (41).

The prediction equations can be derived from the continuos minimum variance filter (40) by omitting the terms which characterize the influence of the observations. Since we often have additional known input signals $\underline{u}(t)$ in the system (besides the stochastic inputs $\underline{w}(t)$, $\underline{v}(t)$) we will take them into account in the prediction equations, supposing, that now in Eq.(9b) we may write

$$\underline{f} := \underline{f}\left[\underline{x}(t),\underline{u}(t),t\right] = \underline{f}\left[\underline{x}(t),t\right]+\underline{B}\left[\underline{x}(t),t\right]\underline{u}(t) \tag{43}$$

Resulting, we obtain the continuous-discrete modified Gaussian second-order filter.

i) Filtering (updating at the instant t_k of an observation $\underline{y}(tk)$

$$\hat{\underline{x}}(t_k|t_k) = \hat{\underline{x}}(t_k|t_{k-1}) + \underline{K}(t_k)\Big\{\underline{y}(t_k) - \Big[\underline{h}\left[\hat{\underline{x}}(t_k|t_{k-1}),t_k\right] + \frac{1}{2}\frac{\partial^2\underline{h}\left[\hat{\underline{x}}(t_k|t_{k-1}),t_k\right]}{\partial\hat{\underline{x}}^2(t_k|t_{k-1})} : \underline{P}(t_k|t_{k-1})\Big]\Big\} \tag{44a}$$

$$\underline{P}(t_k|t_k) = \left[\underline{I} - \underline{K}(t_k)\underline{H}(t_k)\right]\underline{P}(t_k|t_{k-1}) \tag{44b}$$

$$\underline{K}(t_k) = \underline{P}(t_k|t_{k-1})\underline{H}^T(t_k)\left(\underline{H}(t_k)\underline{P}(t_k|t_{k-1})\underline{H}^T(t_k) + \frac{1}{2}\left(\partial^2_{\underline{h}}\,\underline{P}^2\partial^2_{\underline{h}}\right) + \underline{R}(t_k)\right)^{-1} \tag{44c}$$

$$k = 0,1,2,\ldots$$

initial cond.: $\hat{\underline{x}}(t_0|t_{0-1}) := \hat{\underline{x}}(t_0)$;

$\underline{P}(t_0|t_{0-1}) := \underline{P}(t_0)$

definitions: $\underline{H}(t_k) := \left.\frac{\partial\underline{h}\left[\underline{x}(t_k),t_k\right]}{\partial\underline{x}(t_k)}\right|_{\underline{x}=\hat{\underline{x}}(t_k|t_{k-1})}$;

$\underline{h}\left[\hat{\underline{x}}(t_k|t_{k-1}),t_k\right]$

$\underline{R}(t_k) := \underline{R}\left[\hat{\underline{x}}(t_k|t_{k-1}),t_k\right]$

$\left(\partial^2_{\underline{h}}\,\underline{P}^2\partial^2_{\underline{h}}\right)_{\kappa\lambda} := \sum_{i,j,k,l=1}^{n}\frac{\partial^2 h_\kappa}{\partial\hat{x}_i\partial\hat{x}_j}\cdot P_{jk}\cdot P_{kl}\frac{\partial^2 h_\lambda}{\partial\hat{x}_k\partial\hat{x}_l}$

$\kappa,\lambda = 1,\ldots,m$

$\hat{x}_i := \hat{x}_i(t_k|t_{k-1})$.

$P_{jk} := P_{jk}(t_k|t_{k-1})$

ii) Predicition (between observations $t_k \leq t < t_{k+1}$)

$$\frac{d\hat{\underline{x}}(t|t_k)}{dt} = \underline{f}[\hat{\underline{x}}(t|t_k),t] + \underline{B}[\hat{\underline{x}}(t|t_k),t]\underline{u}(t) + \\ + \frac{1}{2}\frac{\partial^2 \underline{f}[\hat{\underline{x}}(t|t_k),t_k]}{\partial \hat{\underline{x}}^2(t|t_k)} : \underline{P}(t_k|t_k) + \\ + \frac{1}{2}\frac{\partial^2 (\underline{B}[\hat{\underline{x}}(t|t_k),t]\underline{u}(t))}{\partial \hat{\underline{x}}^2(t|t_k)} : \underline{P}(t|t_k) \qquad (45a)$$

$$\frac{d\underline{P}(t|t_k)}{dt} = \underline{F}(t)\underline{P}(t|t_k) + \underline{P}(t|t_k)\underline{F}^T(t) + \\ + [\underline{G}[\underline{x}(t),t]\underline{Q}(t)\underline{G}^T[\underline{x}(t),t]|t_k] \qquad (45b)$$

$$\text{definition: } \underline{F}(t) = \left\{\frac{\partial \underline{f}[\underline{x}(t),t]}{\partial \underline{x}(t)} + \frac{\partial (\underline{B}[\underline{x}(t),t]\underline{u}(t))}{\partial \underline{x}(t)}\right\}\Bigg|_{\underline{x}=\hat{\underline{x}}(t|t_k)}$$

Remark: the approximation of $[\underline{G}\ \underline{Q}\ \underline{G}^T]^{\wedge}$ is possible by

i) expanding $\underline{G}\ \underline{Q}\ \underline{G}^T$ up to second order and taking then the expectation operation. This yields a form as in Eq. (40b).

ii) expanding $\underline{G}$ up to second order, generating the product $\underline{G}\ \underline{Q}\ \underline{G}^T$ and taking the expectation

iii) expanding $\underline{G}$ only to first order, generating the product $\underline{G}\ \underline{Q}\ \underline{G}^T$ and taking the expectation.

The procedure i is more convenient than ii while iii is of advantage with regard to the existence and uniqueness of the solution /36/.

The structure of this modified Gaussian second-order estimator is illustrated in Fig.9. If we omit the blocks containing the second order terms

$$\frac{\partial^2 \underline{f}}{\partial \hat{\underline{x}}^2} : \underline{P}, \quad \frac{\partial^2 (\underline{B}\ \underline{u})}{\partial \hat{\underline{x}}^2} : \underline{P} \quad , \quad \frac{\partial^2 \underline{h}}{\partial \hat{\underline{x}}^2} : \underline{P} \quad , \qquad (46)$$

we get the continuous-discrete extended Kalman-Bucy filter.

As mentioned, the extended Kalman-Bucy filter has often been applied due to its simple structure and low computational burden. A good survey of early nonlinear filter applications is given e.g. by Sorenson/37/ in 1973.
One popular field of applications of nonlinear filtering is the combined estimation of states and parameters in linear and nonlinear systems. In this case, a dynamical model for the unknown parameters is assumed, the parameter equations are added to the system model, and the "state variables" of this augmented model are estimated. However, the application of only a first order filter may cause biased estimates /38/ as a result of omitting higher order terms like (46) in the filter approximation. That is why modifications of the extended Kalman filter have been proposed /39/. On the other hand, second order filters such as the modified Gaussian second-order filter have been succesfully applied for parameter estimation problems /40/, /41/, /42/.

It should be noticed however, that second-order filters are generally less robust than first order ones and that they have a smaller region of convergency. Therefore the initial conditions should not be too bad and a special initialization procedure like a least squares parameter estimator will be of advantage /43/.

Apart from the filters derived in the preceding paragraph, there exist, of course, various other approximations which can not be discussed in this paper but are contained to a large extent in standard textbooks on estimation theory /23/, /24/, /30/, /44/). We just mention stochastic approximated filters (/45/, /46/) where stochastic approximated polynomials are substituted for the Taylor series expansion of the nonlinear systems functions $\underline{f}(\underline{x})$ and $\underline{h}(\underline{x})$.

4. GLOBAL APPROXIMATIONS OF NONLINEAR FILTERS (BAYES-LAW-CALCULATION)

The only consideration of the first and second moments of the density as we did in the local filter approximations of paragraph 4 is of questionable value if the a priori density $p[\underline{x}(t_0),t_0]$ and/or the noise processes $\{\underline{w}(t)\}$, $\{\underline{v}(t)\}$ are not Gaussian, i. e. asymmetrical or even multimodal.

In these cases, the implementation of the Bayesian recursive estimation equations (23) may be necessary. This requires the approximation of the complete conditional densities. As a result the validity of the estimator is not restricted to certain reference points in the state space, hence we speak of global approximations of nonlinear filters.

Assuming the time-discrete nonlinear system model (Eq.(11)) we obtained the corresponding Bayesian estimation equations (23) which have to be calculated in the following steps.

i) $\underline{t=t_0}$: evaluation of the filter density $p[\underline{x}(t_0) \mid \underline{y}_{t_0}]$ after the first observation $\underline{y}(t_0)$ according to Eq. (23a).

- calculation of the product
  $p_{\underline{v}(t_0)}[\underline{y}(t_0)-\underline{h}[\underline{x}(t_0)]] \cdot p[\underline{x}(t_0)]$
- normalization by solution of the nonlinear convolution integral (23b)

ii) evaluation of the a priori (prediction) density $p[\underline{x}(t_1)|\underline{y}(t_0)]$ by performing the nonlinear convolution in Eq.(23c) with
$p_{[\underline{w}(t_0)}[\underline{x}(t_1)-\underline{\varphi}[\underline{x}(t_0)]]$

iii) $\underline{t=t_1}$: calculation of the new filter density $p[\underline{x}(t_1) \mid \underline{y}_{t_1}]$ using the observation $\underline{y}(t_1)$ as under i) etc.

For the practical solution of the estimation problem, one has first to define appropriate finite collections of points for the approximation of the a priori density $p[\underline{x}(t_0)]$. In the two-dimensional case, we can illustrate this by a grid in the x_1-x_2-plane where the gridpoints are the references on which the approximation can be based. This grid can be maintained or redefined for every sampling interval (floating grid); the latter is often numerically more effective /47/. With known grid, the method of approximating the density has to be established as well as the method of numerical integration.

4.1 Orthogonal Series expansion approximation

The approximation of the density function by orthogonal functions or polynominals has frequently been proposed in the past. Sorenson and Stubberud /48/ used the Edgeworth-series which consists of Hermite polynomials. However, the coupling of a Hermite polynomial with a Gauss-Hermite quadrature /49/, /50/ seems to be more encouraging.

Moreover, a Fourier series expansion may be of use for applications with periodic probability density functions (e. g. in phase-modulation problems) /51/.

However, the inherent disadvantage of all these approximations by orthogonal series expansions is the fact, that the resulting density functions are not really density functions. They can, for instance, assume negative values which finally yields divergency of the estimator. In order to avoid this phenomenon, the number of terms in the series have to be increased which unfortunately augments the computational burden at the same time. Therefore, another approach has been suggested; it is the

4.2 Gaussian sum approximation

In this case the densities in the Bayes-law calculator are approximated by non-orthogonal functions i. e. a weighted sum of Gaussian probability densities. This has been proposed by Alspach and Sorenson /52/, /53/. For example a density $p(\underline{x})$ is approximated by

$$p_a(\underline{x}) = \sum_{i=1}^{q} \alpha_i \, N_{\underline{x}}(\hat{\underline{x}}_i, \underline{P}_i) \tag{47a}$$

with the definition

$$N_{\underline{x}}(\underline{a}, \underline{B}) := (2\pi)^{-n/2} (\det \underline{B})^{-1/2} \exp\left\{ -\frac{1}{2}(\underline{x}-\underline{a})^T \underline{B}^{-1} (\underline{x}-\underline{a}) \right\} \tag{47b}$$

and the nonnegative weighting coefficients α_i, characterized by

$$\sum_{i=1}^{q} \alpha_i = 1 \, . \tag{47c}$$

This approximation is motivated by the fact that p_a converges uniformly to p for a large class of densities.

The mean values $\hat{\underline{x}}_i$ represent the grid points for the approximation. These grid points are selected uniformly over the region where $p(\underline{x})$ is significantly different from zero. The covariances $\underline{P}_i$ are defined to be diagonal $(\sigma_i^2 \cdot \underline{I})$ and σ_i is selected to minimize the deviation between p and p_a. The coefficients α_i are chosen to be proportional to the values of the density function at the grid points $\hat{\underline{x}}_i$, and the number of terms q has to be increased until a suitable approximation is obtained.

The quality of the Gaussian sum approximation is illustrated in Fig.10 where a uniform distribution is approximated by this method /54/.

Now in view of the filter realization using Gaussian sums the proceeding can be the following:

i) the a priori density $p[\underline{x}(t_o)]$ is approximated by Gaussian sums.

ii) the evaluation of the filter density requires the calculation of the product

$$p_{v(t_o)}\left[\underline{y}(t_o)-\underline{h}\left[\underline{x}(t_o)\right]\right] \cdot p[\underline{x}(t_o)].$$

The result is unfortunately no longer a normal distribution. Therefore,

iii) the density $p_v\left[\underline{y}-\underline{h}(\underline{x})\right]$ will be linearized around each grid point; than an extended Kalman filter can be applied at each grid point to calculate $\hat{\underline{x}}_i(t_o \mid t_o)$ and $\underline{P}_i(t_o \mid t_o)$, the terms of the Gaussian sum approximation of the a posteriori density $p[\underline{x}(t_o) \mid t_o]$

iv) After that the prediction density $p[\underline{x}(t_1) \mid t_o]$ is evaluated in a similar way as under iii.

The result of the Gaussian sum approximation for realization of Bayes-law calculators essentially requires the parallel operation of so many extended Kalman-filters as we have terms in the Gaussian sum.

In view if Fig.10 it should be noticed that nearly 50 (!) extended Kalman filters are necessary for the implementation of a Gaussian sum approximate filter with uniform a priori probability density.

4.3 Point mass approximation

This approximation may be regarded as a special case of the Gaussian sum approximation where the different probability densities at the grid points are Dirac δ-impulses with given area (the point masses). Hence, the resulting density is always positive.

For example, the approximation of the a posteriori density can be written as

$$p[\underline{x}(t_k \mid \underline{Y}_{t_k}] = \sum_{i=1}^{q} \underline{m}_i^T(t_k \mid t_k) \cdot \delta\left[\underline{x}(t_k)-\hat{\underline{x}}_i(t_k \mid t_k)\right] \qquad (48)$$

with the q^n grid points $\hat{\underline{x}}_i$ $(i=1,\ldots,q)$ and the corresponding point masses $\underline{m}_i^T$.

This approximation has been given by Bucy /55/. For reduction of computational burden, a "floating grid" has been suggested in /47/. The grid is centered at the actual position of the conditional mean $\hat{\underline{x}}$ and the eigenvectors are used to define the principal axes of the grid. With the grid points given, the Bayesian recursion relations are readily evaluated, which is essentially equivalent to using a rectangular integration rule to accomplish the numerical quadratures.

4.4 Spline approximation

In this approximation the grid points are the mesh points for the interpolation of the probability densities with multidimensional cubic spline functions. This approximation has been suggested by De Figueiredo and Jan /56/ in 1971 for realization of discrete-time nonlinear filters.

The probability densities approximated by splines assume no negative values and the numerical treatment of the spline based filters is comparatively simple.

REFERENCES

/1/ PTOLEMAEUS, C. Opera quae exstant omnia, Vol.I - Syntaxis Mathematica; Editor: J.L.Heiberg. Leipzig: Teubner 1903

/2/ LEGENDRE, A.M. Nouvelles méthodes pour la détermination des orbites des comètes. Paris 1806

/3/ GAUSS, C.F. Theorie der Bewegung der Himmelskörper, welche in Kegelschnitten die Sonne umlaufen (German translation of the Latin original dated 1809). Hannover: Carl Meyer 1865

/4/ DEUTSCH, R. Estimation theory. Englewood Cliffs, N.J., Prentice Hall 1965

/5/ SORENSON, H.W. Least squares estimation: from Gauss to Kalman. IEEE Spectrum, Vol.7, pp. 63-68, 1970

/6/ FISHER, R.A. On an absolute criterion for fitting frequency curves. Messenger of Math., Vol.41, pp.155-160, 1912

/7/ KOLMOGOROFF, A. Interpolation und Extrapolation von stationären Zufallsfolgen. Bull.Acad.Sci.USSR, Ser.Math., Vol.5, pp 3-14, 1941

/8/ WIENER,N. Extrapolation, interpolation, and smoothing of stationary time series with engineering applications: New York: John Wiley 1949

/9/ KALMAN, R.E A new approach to linear filtering and prediction problems. Trans.ASME, Series D, Vol.82, pp.35-45, 1960

/10/ KALMAN, R.E. BUCY, R.S. New results in linear filtering and prediction theory. Trans. ASME, Series D, Vol.83, pp.95-108, 1961

/11/ SCHRICK, K.W.(ed.) Anwendung der Kalman-Filter-Technik .Oldenbourg, München 1977

/12/ SCHMIDT, S.F. The Kalman Filter :its recognition and development for aerospace applications. J.Guidance and Control, Vol.4 No.1 pp.4-7, 1981

/13/ BRYSON, A.E. HO, Y. Applied optimal control. Waltham/Mass.: Bleisdell 1969

/14/ KREBS, V. Lineare Optimalfilter in der Meß- und Regelungstechnik - eine Einführung. Aussprachetag des VDI/VDE: Filterverfahren und Beobachtersysteme in der Meß- und Regelungstechnik. Frankfurt/Main 1975

/15/ STRATONOVICH, R.L. Conditional Markov processes. Theory Prob. Appl. 5 pp. 156-178, 1960

/16/ KUSHNER, H.J. On the differential equations satisfied by conditional probability densities of Markov processes, with applications J.SIAM on Control, Series A, Vol.2, pp. 106-119, 1964

/17/ BUCY, R.S. Nonlinear filtering theory. IEEE Trans. on Autom. Control Vol.AC-10, p. 198, 1965

/18/ WONHAM, M. Some applications of stochastic differential equations to optimal nonlinear filtering. J.SIAM on Control, Series A Vol.2, pp. 347-369, 1965

/19/ EINSTEIN, A. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. Annalen der Physik 17 (1905), pp. 549-560

/20/ FOKKER, A.D. Die mittlerer Energie rotierender elektrischer Dipole im Strahlungsfeld. Annalen der Physik 43 (1914), pp. 810-820

/21/ PLANCK, M. Über einen Satz der statistischen Dynamik und seine Erweiterung in der Quantentheorie. Sitzungsberichte d.Königl.Preußischen Akad. d. Wiss. 1917, pp. 324-341

/22/ KOLMOGOROFF, A. Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung. Math. Ann. 104 (1931), pp. 415-458

/23/ JAZWINSKI, A.H. Stochastic processes and filtering theory. Academic Press, New York 1970

/24/ KREBS, V. Nichtlineare Filterung. Oldenbourg, München, 1980

/25/ KUSHNER, H.J. — On the dynamical equations of conditional probability density functions, with applications to optimal stochastic control theory. J.Math.Anal.Appl.8,(1964), pp.332-344.

/26/ BASS, P.W. NORUM, V.D. SCHWARTZ, L. — Optimal multichannel nonlinear filtering. J.Math.Analysis and Appl. 16 (1966), pp. 152-164.

/27/ SORENSON, H.W. — On the development of practical nonlinear filters. Contrib. in Laimiotis, G. (Ed.)Estimation theory. American Elsevier, New York 1974, pp. 63-80

/28/ DENHAM, W.F. PINES, S. — Sequential estimation when measurement function nonlinearity is comparable to measurement error. AIAA J.4 (1966), pp. 1071-1076

/29/ WISHNER, R.P. TABACZYNSKI, J.A. ATHANS, M. — On the estimation of the state of noisy nonlinear multivariable systems. IFAC Symposium Mehrgrößenregelung, Düsseldorf, 1968.

/30/ SAGE, A.P. MELSA J.L. — Estimation theory with applications to communications and control. McGraw-Hill, New York 1971

/31/ JAZWINSKI, A.H. — Filtering for nonlinear dynamical systems. IEEE Trans. on Autom. Control, Vol.AC-11 (1966), pp. 765-766

/32/ JAZWINSKI, A.H. — Stochastic processes with application to filtering theory. Analytical Mechanics Associates Inc., Seabrook, Maryland, Report No.66-6, 1966

/33/ FISHER, J.R. — Optimal nonlinear filtering. Advanc.Control Systems, Vol. (1967), pp. 197-300.

/34/ ATHANS, M. WISHNER, R.P. BERTOLINI, A. — Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. 1968 Joint Automatic Control Conference, Ann Arbor, Michigan, 1968, pp.364-382.

/35/ FARISON, J.B. — Parameter identification for a class of linear discrete systems. IEEE Trans. on Autom.Control, Vol.AC-12 (1967), p. 109

/36/ SCHWARTZ, L. STEAR, E.B. — A computational comparison of several nonlinear filters. IEEE Trans. on Autom.Control, Vol.AC-13(1968) pp.83-86

/37/ SORENSON, H.W. — Estimation for dynamic systems: a perspective. Fourth Symp. on Nonlinear Estimation Theory and its Applications. San Diego, Sept. 1973, pp. 291-318

/38/ LJUNG, L. — Asymptotic behaviour of the extended Kalman filter as a parameter estimator for linear systems. IEEE Trans.on Automatic Control AC-24 (1979), pp.36-50

/39/ YOSHIMURA, T. KONISHI, K SOEDA, T. — A modified extended Kalman filter for linear discrete-time systems with unknown parameters. AUTOMATICA 17(1981), pp.657-660

/40/ KREBS,V. — Combined state and parameter estimation for a class of nonlinear stochastic systems by modified nonlinear filtering. IFAC-Symposium on Stochastic Control, Budapest 1974 , Techn.Session B2, pp.121-129.

/41/ KREBS, V. — Zur Erkennung nichtlinearer stochastischer Systeme. Dissertation D 17, TH Darmstadt, 1976.

/42/ FISCHER, H.-P. FINNEMANN, H. — A nonlinear xenon estimator for on-line control of large PWRs. Atomenergie/Kerntechnik, Vol.37 (1981), pp.266-270.

/43/ BONSE, B. — Zustands- und Parameteridentifizierung bei linearen stochastischen Abtastsystemen. Dissertation D14-001 Universität-Gesamthochschule Paderborn, 1981

/44/ GELB, A.(ed.) — Applied optimal estimation. MIT Press, Cambridge, Massachusetts 1974

/45/ PHANEUF, R.J. — Approximate nonlinear estimation. Ph.D.-Dissertation, Massachusetts Institute of Technology, 1968

/46/ SUNAHARA, Y. — An approximate method of state estimation and control for nonlinear dynamical systems under noisy observations. IV IFAC-Congress, Warschau, 1969 , Techn.Session 56.3

/47/ BUCY, R.S. SENNE, K.D. — Digital synthesis of nonlinear filters. Automatica 7 (1971), pp. 287-298

/48/ SORENSON, H.W. STUBBERUD, A.R. — Nonlinear filtering by approximation of the a posteriori density. International J.Control 8 (1968), pp. 33-51

/49/ HECHT, C. — System identification using Bayesian Estimation. Proc. fourth Symp. on Nonlinear Estimation Theory and its Applications. San Diego, Sept. 1973, pp. 107-113

/50/ McREYNOLDS, S.R. — Multidimensional Hermite-Gaussian quadrature formulae and their application to nonlinear estimation. Proc.sixth Symp. on Nonlinear Estimation Theory and its Applications. San Diego, Sept. 1975, pp. 188-191

/51/ BUCY R.S. HECHT, C. SENNE K.D. — An engineer's guide to building nonlinear filters. Frank J. Seiler Research Laboratory, USAF Academy, Colorado. Report SRL-TR-72-0004, May 1972

/52/ ALSPACH, D.L. SORENSON H.W. — Nonlinear Bayesian estimation using Gaussian sum approximations. IEEE Trans. on Autom. Control, Bd. AC-17 (1972), pp.439-448

/53/ ALSPACH, D.L. — A Bayesian approximation technique for estimation and control of time-discrete stochastic systems. Ph.D.-Dissertation, University of California, San Diego, 1970

/54/ SORENSON,H.W. ALSPACH, D.L. — Recursive Bayesian estimation using Gaussian sums. Automatica 7 (1971), pp. 465-479

/55/ BUCY, R.S. — Bayes theorem and digital realizations for nonlinear filters. J. Am.Astronaut.Soc. 17 (1969), pp. 80-94

/56/ De FIGUEIREDO, R.J.P. JAN, Y.G. — Spline filters. Proc. second Symp. on Nonlinear Estimation Theory and its Applications. San Diego, Sept. 1971, pp. 127-138

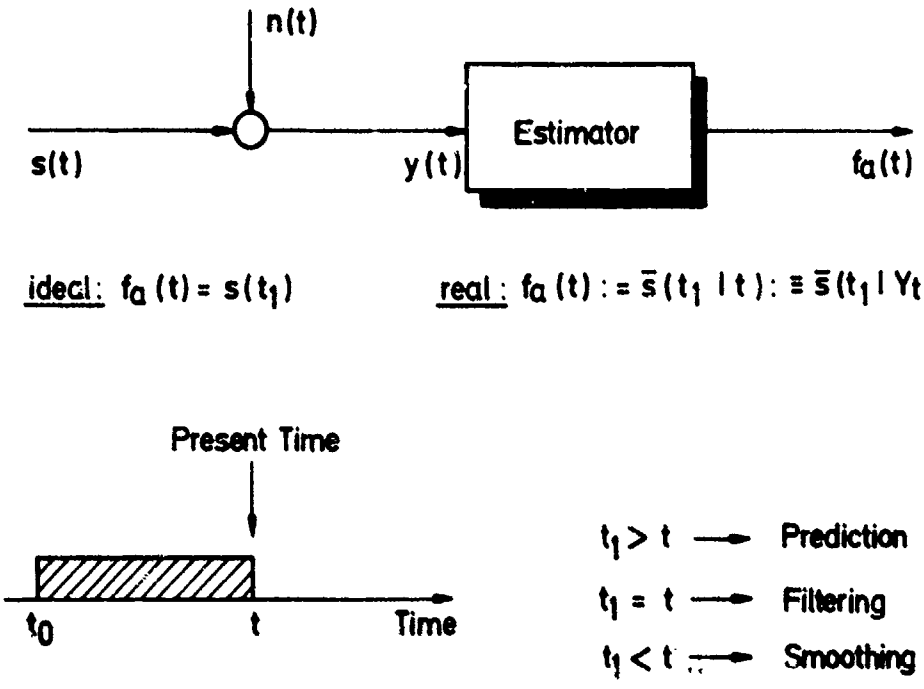


Fig.1: General Estimation Problem and definition of prediction, filtering and smoothing

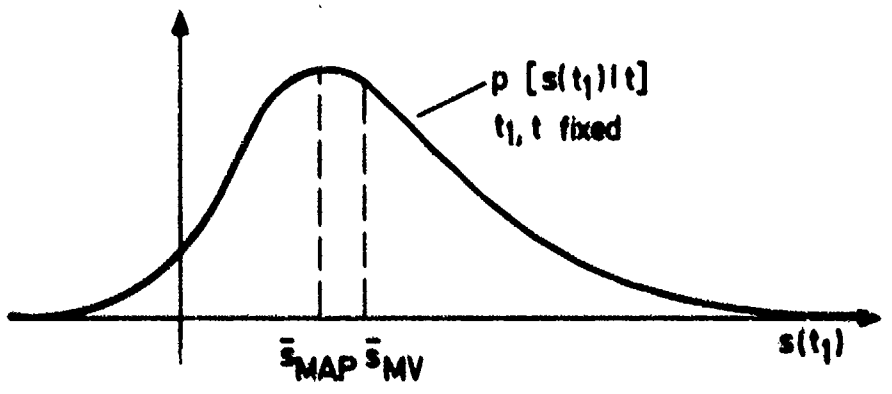


Fig.2: Different estimates based on $p[s(t_1)|t]$:
$\bar{s}_{MAP}$ - maximum a posteriori estimate
$\bar{s}_{KV}$ - minimum variance estimate

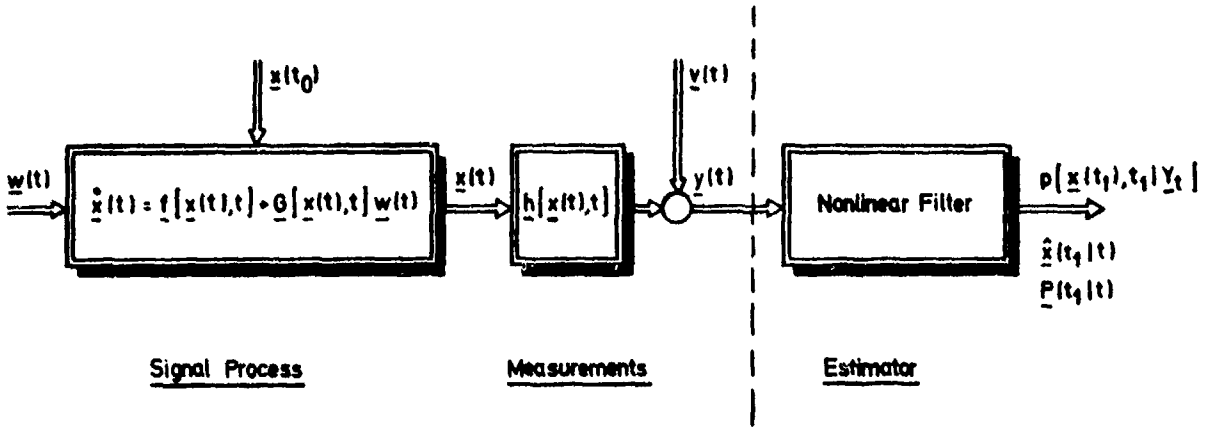


Fig.3: System model and nonlinear filter in continuous time

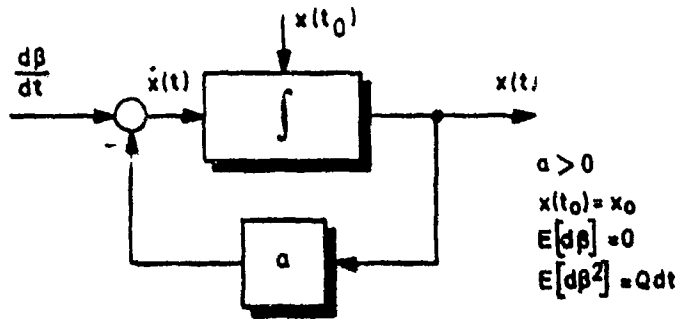


Fig.4: System model for demonstration of the Fokker-Planck equation

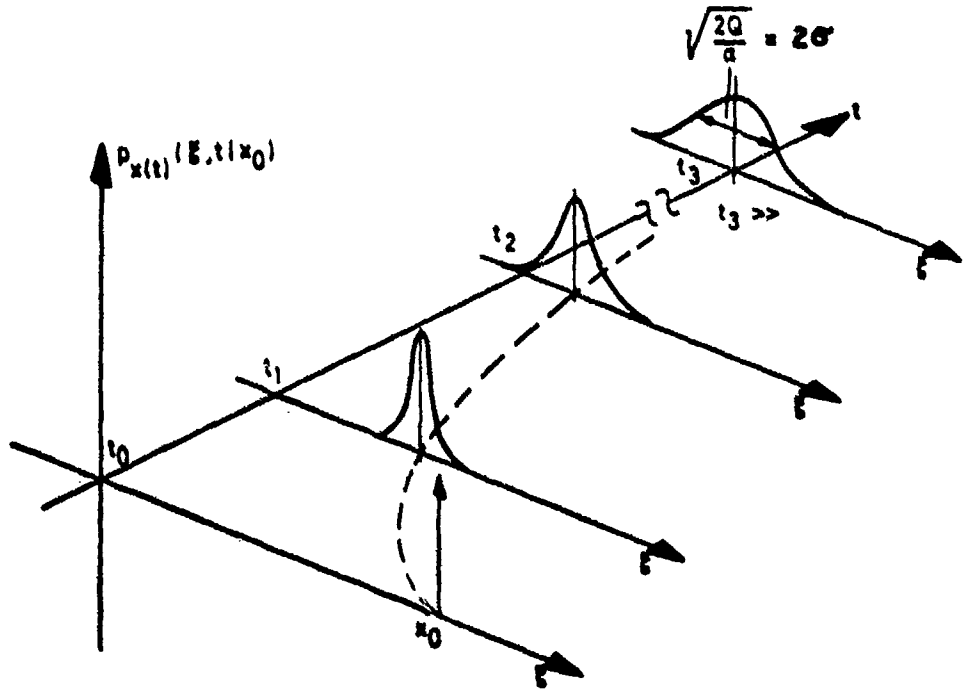


Fig.5: Evolution of the transition probability density for the first order system of Fig.4, given by the Fokker-Planck equation

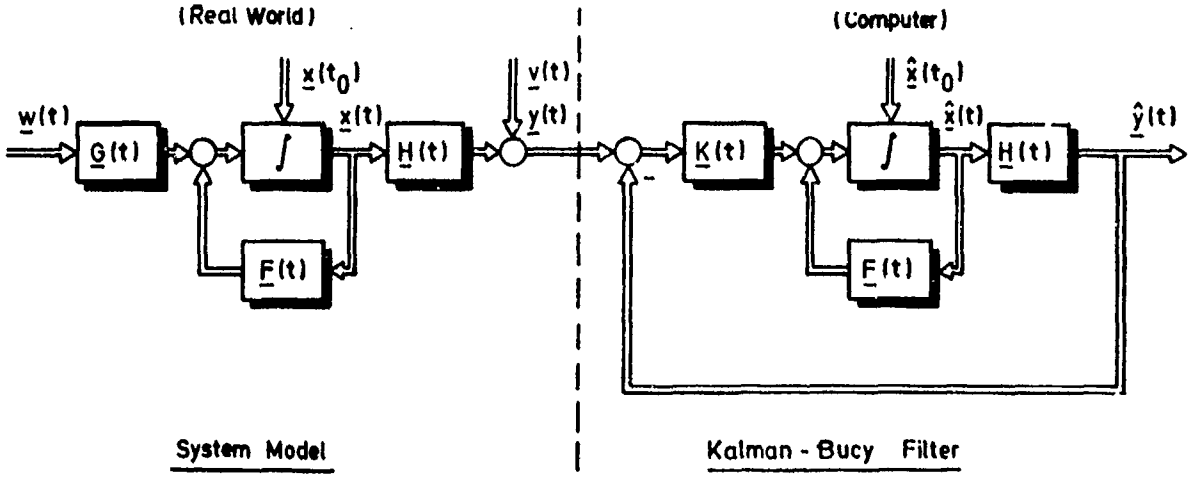


Fig.6: Linear system model and optimal minimum variance estimator (Kalman-Bucy filter) without covariance equation

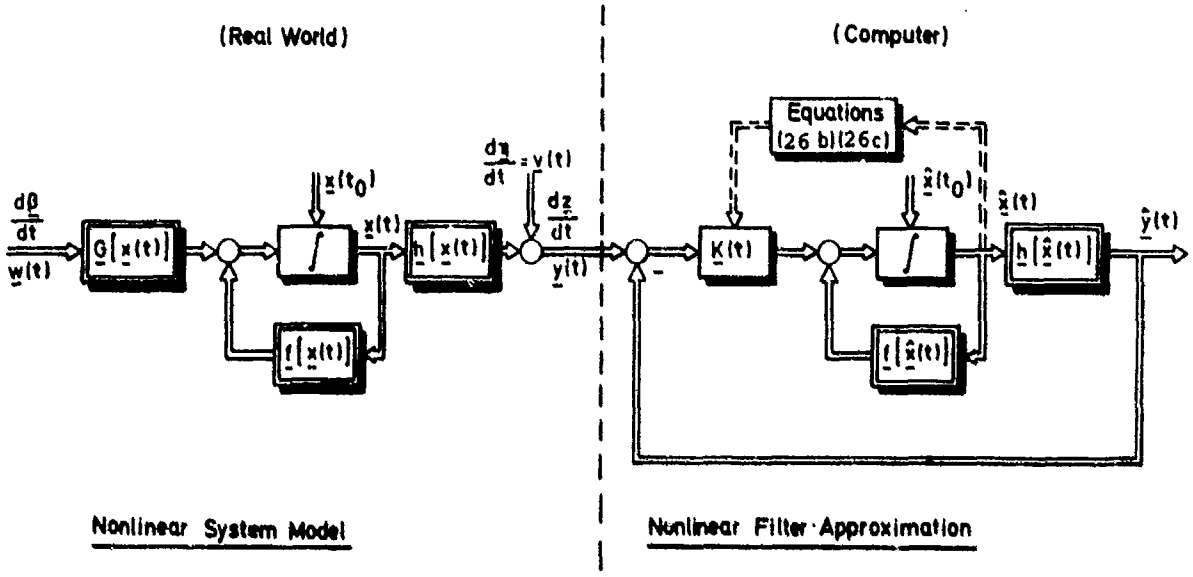


Fig.7: Nonlinear system model and continuous first order minimum variance filter (extended Kalman filter)

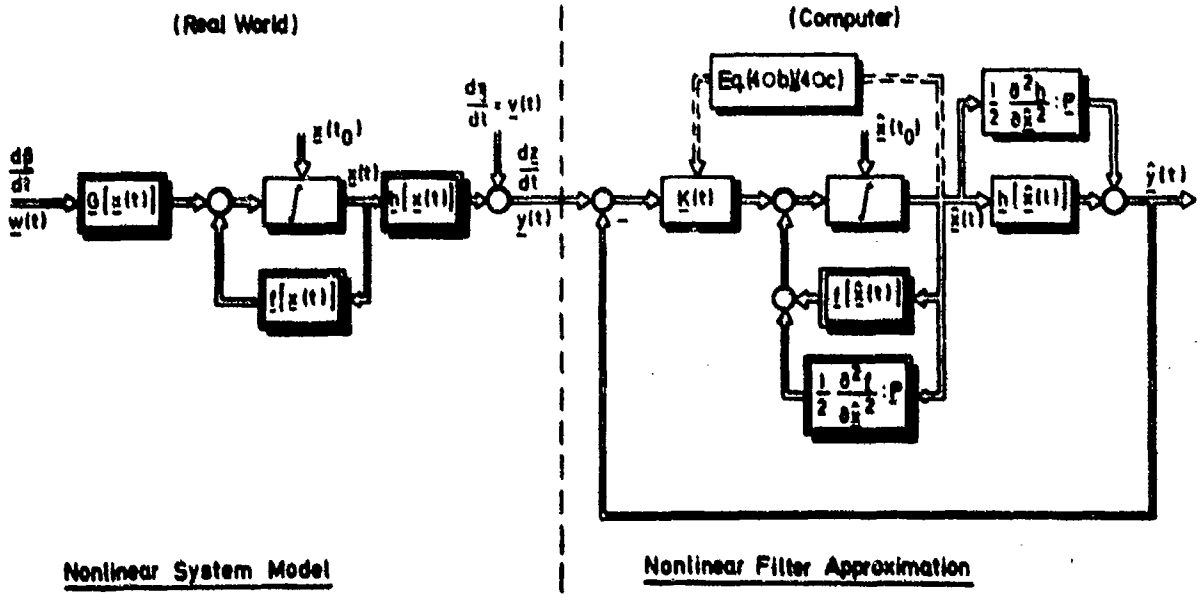


Fig.8: Nonlinear system model and continuous second-order minimum variance filter

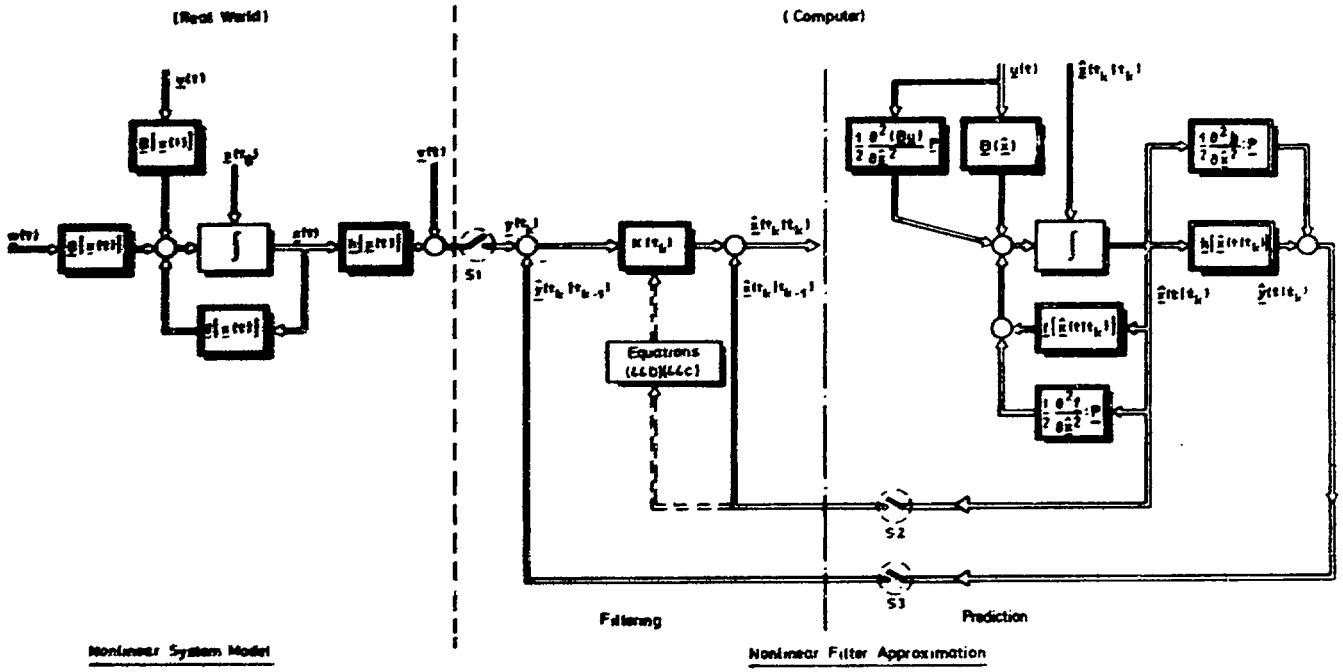


Fig.9: Nonlinear system model and continuous-discrete modified Gaussian second-order filter

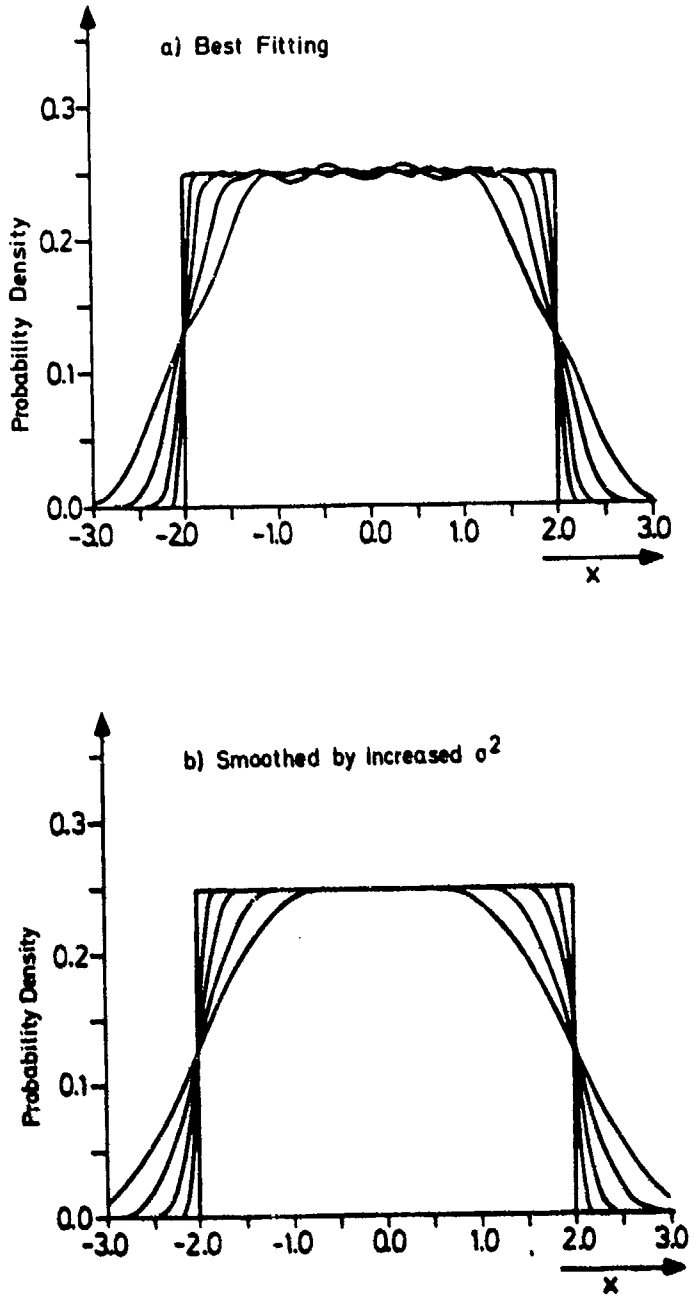


Fig.10: Gaussian sum approximation of a uniform distribution using 6, 10, 20 and 49 terms.

EXACT AND APPROXIMATE NONLINEAR ESTIMATION TECHNIQUES

BY
Dr. D.F. Liang
Defence Research Establishment Ottawa
Shirley's Bay, Ottawa
Canada K1A0Z4

SUMMARY

This chapter presents a unified approach to derive estimation algorithms for discrete and continuous nonlinear systems with and without delays, corrupted by white noise as well as non-white noise processes.

In the case of continuous systems, filtering algorithms are derived for nonlinear systems without delays, imbedded in white noise, correlated noise and noise free processes. The filtering equations obtained for nonlinear systems with white noise processes are exact, but for non-white noise processes the results obtained are approximate.

In the case of discrete-time systems, nonlinear estimation algorithms, that directly yield the fixed-lag, fixed-point and fixed-interval smoothing and the filtering algorithms, are derived for nonlinear delayed systems with measurements corrupted by white noise and correlated noise processes. The derivation is straightforward and clearly indicates the close links between three different classification of smoothers and the filtering estimator.

For systems with polynomial, product-type or state-dependent sinusoidal nonlinearities, the proposed algorithms can be practically realized without the need of approximation under the assumption that the estimator errors are Gaussian. Such an assumption is significantly different from the most commonly used assumption that the state is Gaussian.

CONTENTS

		PAGE
1.	GENERAL INTRODUCTION	2
1.1	Introduction	2
1.2	Scope of This Chapter	2
2.	EXACT AND APPROXIMATE MINIMUM VARIANCE FILTERING FOR NONLINEAR CONTINUOUS SYSTEMS	3
2.1	Introduction	3
2.2	Optimal Minimum Variance Continuous Nonlinear Filtering With White Noise Processes	3
2.3	Evaluation of Expectations	6
2.4	Minimum Variance General Continuous Nonlinear Filter	7
2.5	Minimum Variance Nonlinear Noise-Free Filtering	8
3.	MINIMUM VARIANCE FILTERING AND SMOOTHING FOR DISCRETE NONLINEAR DELAYED SYSTEMS WITH ADDITIVE WHITE NOISE	9
3.1	Introduction	9
3.2	The Problem Statement	9
3.3	The Derivation of Nonlinear Smoothing Algorithms	10
3.4	The Nonlinear Fixed-Lag Smoothing	12
3.5	The Nonlinear Fixed-Point Smoothing	13
3.6	The Nonlinear Fixed-Interval Smoothing	13
3.7	The Nonlinear Filtering	13
3.8	Estimation in Linear Discrete Systems	14
4.	MINIMUM VARIANCE FILTERING AND SMOOTHING FOR NONLINEAR SYSTEMS WITH CORRELATED NOISES	15
4.1	Introduction	15
4.2	The Problem Statement	15
4.3	The Derivation of the Smoother	15
4.4	The Nonlinear Filtering	18
4.5	Estimation in Linear Discrete Systems	18
5.	CONCLUSIONS	19
	REFERENCES	20

SECTION 1: GENERAL INTRODUCTION

1.1 INTRODUCTION

The importance of linear estimation theory as represented by Kalman-Bucy filter[1] is now well recognized. It has found extensive applications in aerospace systems, such as Apollo and space shuttle; guidance and navigation systems; econometrics, seismology and meteorology, biomedical, communications, and many other practical scientific and engineering problems.

However, dynamic system models and measurement models for the majority of realistic control problems are inherently nonlinear. Most of the work in nonlinear estimation is very theoretical, some no more than a philosophy of approach rather than a procedure leading to the derivation of practical estimations.

One of the main lines of attack to the nonlinear estimation problem is the probability approach pioneered by Stratonovich [2], and subsequently taken up by Kushner [3], Wonham [4] and Bucy [5]. The truly optimal nonlinear filters for systems corrupted with additive white noise, were given in Kushner [3,6], however, their exact solutions required infinite dimensional systems which are practically impossible to realize except in trivially simple cases. For practical realization, extensive work has been carried out to approximate the nonlinear filters. One group of papers [7-9] attempts to obtain the numerical solution using the so-called Bayesian point of view. They assume to have a completely valid probability description of the system, so that Bayes rule can be applied to obtain a recursive description of the *a posteriori* probability density function. However, in many realistic problems the mathematical model of the system contains uncertainty that cannot be priorly modeled by probability distribution functions. Furthermore, the Bayesian approach has the disadvantage of imposing a rather severe computational burden for even simple systems. Another group of methods [10-14] essentially approximates the mean and variance of the *a posteriori* density functions based on perturbation relative to a prescribed reference. The majority of these techniques employ the Taylor's series expansion of the dynamic system and measurement nonlinearities, neglecting second- and higher - order terms. Recently, Sunahara [15] proposed to replace the nonlinear functions by quasi-linear functions via stochastic linearization. In general, methods based on Taylor's series expansion suffer from the defect of replacing global distribution properties of a function by its local derivatives aggravated by corruption of noise processor. Thus it is questionable whether the more sophisticated approximation [16] provide useful improvements relative to the widely applied first-order approximation known as the extended Kalman filter.

In the areas of nonlinear smoothing estimation, Leondes et al. [17] derived the exact functional equations for the smoothing density functions and the smoothed estimates; but their solutions are prohibitive except in trivially simple cases. Other works on nonlinear smoothing were presented by Kailath and Frost [18], Lainiotis [19] and Lee [20].

As far as state estimation problems for systems with time delays are concerned, Kwakernaak [21] used the method of orthogonal projection to derive filtering equations for linear continuous systems with multiple time delays, which, when solved, also yield smoothing estimates. Priemer and Vacroux [22, 23] later considered the estimation problems in linear discrete systems containing multiple delays in the message model. Farooq and Mahalanabis [24] rederived the estimation algorithms of Priemer and Vacroux using the state augmentation technique, however, it was pointed out in [22] that the augmentation of state vectors has the effect of increasing the dimensions of the system, and thus lead to a filter that is computationally inefficient. Biswas and Mahalanabis [25] presented fixed-lag smoothing algorithms for the continuous systems with time delays by first discretizing the continuous systems and then employing the state augmentation technique. This was further extended to the fixed-interval smoothing problem by Faroog and Balasubramanian [26]. Approximate smoothing and filtering equations were also derived by Yu et al [27] for a general class of nonlinear functional differential systems.

1.2 SCOPE OF THIS CHAPTER

This chapter is devoted to the derivation of nonlinear estimation algorithms for discrete and continuous nonlinear dynamic systems with and without delays in the message models, corrupted by white Gaussian noise, correlated noise and colored noise processes.

Section 2 follows the presentation of [28, 30] and deals with continuous-time nonlinear systems without delays corrupted with additive white noise as well as non-white noise processes. The basic approach makes use of the matrix minimum principle together with the Kolmogorov [31] and Kushner [3,6] equations to minimize the error-variance, taken to be the estimation criterion. However, the exact algorithms derived in such a manner require infinite dimensional systems to realize, which is computationally impossible. In order that the estimation algorithms can be physically realized, it is assumed that the conditional probability density functions of the estimator errors are Gaussian. Techniques are presented to show how one can exactly evaluate expectations of polynomial, product-type or state-dependent sinusoidal nonlinearities under the above assumption.

For the purpose of assessing the performance of the proposed minimum variance filter and to compare it with various other approximate finite dimensional filters. Liang [32] selected various types of nonlinear systems, which were simulated on a digital computer. His results clearly indicate the superiority of the proposed minimum variance filter over those of other filters investigated, and theoretical explanations are also given for the apparent poor performance characteristics of the various other filters considered.

Section 3 follows the presentation of [28,33] to derive discrete-time filtering and smoothing algorithms for nonlinear time-delayed systems imbedded in white Gaussian noise processes. The main technique makes use of the matrix minimum principle to derive the optimal values of the coefficients in the estimation algorithms under the requirements that the estimates be unbiased. The resulting algorithms can be recursively evaluated under the assumption that the probability density functions of the estimator errors are Gaussian. Examples are included to illustrate the use of the proposed estimation algorithms, in particular, they provide better insight as to, how one can properly substitute for the

discrete-time indices, in order to arrive at the filtering, fixed-interval smoothing, fixed-point smoothing and the fixed-lag smoothing algorithms. Results pertaining to linear problems are directly deduced from the nonlinear estimation algorithms, they agree well with those derived in the literature, using other optimization techniques.

Section 4 deals with discrete nonlinear time-delayed systems imbedded in correlated noise processes [28,34]. The derivation as well as the presentation follow that of Section 3. Similar to that of Section 3, the linear estimation algorithms for these problems can be obtained from the nonlinear estimation algorithms.

Section 5 summarizes results presented in Sections 2 to 4.

SECTION 2

EXACT AND APPROXIMATE MINIMUM VARIENCE FILTERING FOR NONLINEAR CONTINUOUS SYSTEMS

2.1 INTRODUCTION

This section is devoted to the estimation problems of nonlinear continuous systems without delays corrupted by (1) additive white Gaussian noise, (2) correlated noise, and (3) noise-free processes.

In Section 2.2, the noise processes are assumed to be Gaussian white, the basic approach makes use of the matrix minimum principle to minimize the error-variance cost function, which is obtained from the exact conditional probability density function, as presented in Kushner [3,6] and Kolmogorov's equations [3]. Therefore, it is not unexpected that the exact nonlinear filtering equations derived for white noise problems closely resemble those of Bass et al. [13], except the argument of the expectations have been transformed from the state $x(t)$ into its estimator error $\tilde{x}(t)$. Section 2.3 shows that for filtering problems with polynomial, product-type or state-dependent sinusoidal nonlinearities, the filtering algorithms can be fully realized without any other approximations under the assumption that the conditional probability density functions of the estimator errors are Gaussian.

In many practical situations, nonlinear dynamic systems are imbedded in non-white noise processes. Therefore, Sections 2.4 and 2.5 deal with more general nonlinear estimation problems, they are respectively, imbedded in correlated noise and noise free processes, and the colored noise problem can be considered as a special case of the noise free estimation problem.

The estimation algorithms derived for non-white noise processes are suboptimal, since the change in probability density function due to the differential measurements δz is neglected. In the special case that the system and measurement models are linear, the resulting algorithms are optimal and agree well with those of the literature [35,36].

2.2 OPTIMAL MINIMUM VARIANCE CONTINUOUS NONLINEAR FILTERING WITH WHITE NOISE PROCESSES

Consider the class of nonlinear systems described by the stochastic differential equation [37]

$$\frac{dx(t)}{dt} = f[x(t),t] + G[x(t),t]w(t) \tag{2.1}$$

with the measurement given by

$$y(t) = h[x(t),t] + v(t) \tag{2.2}$$

Where $x(t)$ and $y(t)$ are the n-dimensional state and m-dimensional measurement vectors, f and h are, respectively, n-and m-dimensional nonlinear vector valued functions, and G is a vector valued matrix.

The random vectors $w(t)$ and $v(t)$ are, statistically independent zero-mean white Gaussian noise processes such that for all $t, \tau \geq t_o$

$$\begin{aligned} \mathrm{Cov}\{w(t), w(\tau)\} &= V_w(t)\ \delta(t-\tau) \\ \mathrm{Cov}\{v(t), v(\tau)\} &= V_v(t)\ \delta(t-\tau) \end{aligned} \tag{2.3}$$

and $$\mathrm{Cov}\{w(t), v(\tau)\} = 0$$

where $\delta(.)$ is the Dirac delta function, and the variances $V_w(t)$ and $V_v(t)$ are non-negative definite and positive definite, respectively.

The initial state vector $x(t_o) = x_o$ is a zero-mean Gaussian random process, independent of $w(t)$ and $v(t)$ for $t \geq t_o$, with a positive definite variance matrix

$$\mathrm{Var}\{x(t_o), x(t_o)\} = V_x(t_o)$$

In the typical filtering problem, it is required to compute $\hat{x}(t)$, the unbiased estimate of $x(t)$ conditioned on the set of measurements

$$Y(t) = [y(s)/t_o \leq s \leq t],$$

such that the cost function for $\tau > t_o$

$$J(\tau) = E\{[x(\tau) - \hat{x}(\tau)]^T M(\tau)\ [x(\tau) - \hat{x}(\tau)]/Y(\tau)\} \tag{2.4}$$

is minimized. Here $M(\tau)$ is an arbitrary symmetric positive definite matrix, T the matrix transpose and

E denotes the expectation operator conditioned upon the set of measurements $Y(\tau)$.

The filtering algorithm is assumed to satisfy the general nonlinear differential equation

$$\dot{\hat{x}}(t) = \ell[\hat{x}(t), t] + K(t)\, y(t) \tag{2.5}$$

where $\ell[\hat{x}(t), t]$ and $K(t)$ are yet unknown. Then the estimation problem is to determine the time varying nonlinear vector valued function $\ell[\hat{x}(t), t]$ and the gain algorithm $K(t)$ such that the cost function of Equation (2.4) is minimized.

In fact, the nonlinear filtering equation may be assumed to take various forms, however, once enough information concerning $\hat{x}(t)$ and $y(t)$ are included in the estimator model, the resulting filtering algorithms would be unique. For example, other dynamic filtering equations such as

$$\dot{\hat{x}}(t) = \ell[\hat{x}(t), t] + K(t)\,\{y(t) - E[y(t)]\} \tag{2.6}$$

can also be assumed. Comparing the structures of Equations (2.5) and (2.6), it is obvious that an extra term $-K(t)\,E[y(t)]$ is included in Equation (2.6), however, the resulting filtering algorithm using either one of the above estimator models, would result in exactly the same nonlinear filtering algorithm.

Now, let $\tilde{x}(t)$ denote the estimator error defined by

$$\tilde{x}(t) = x(t) - \hat{x}(t) \tag{2.7}$$

using Equations (2.1), (2.2), and (2.5), the derivatives of Equation (2.7) becomes

$$\dot{\tilde{x}}(t) = f[\tilde{x}(t) + \hat{x}(t), t] + G[\tilde{x}(t) + \hat{x}(t), t]\, w(t) - \ell[\hat{x}(t), t] - K(t)\,\{h[\tilde{x}(t) + \hat{x}(t), t] + v(t)\} \tag{2.8}$$

Since $\hat{x}(t)$ is required to be an unbiased estimate, it therefore requires the expectations of both $\tilde{x}(t)$ and $v(t)$ be zero. Hence, if the expectations of both sides of Equation (2.8) are taken, it is necessary that

$$\ell[\hat{x}(t), t] = \hat{f}[\tilde{x}(t) + \hat{x}(t), t] - K(t)\,\{\hat{h}[\tilde{x}(t) + \hat{x}(t), t]\} \tag{2.9}$$

where

$$\hat{f}[\tilde{x}(t) + \hat{x}(t), t] = E\{f[\tilde{x}(t) + \hat{x}(t), t]/Y(t)\}$$

$$\hat{h}[\tilde{x}(t) + \hat{x}(t), t] = E\{h[\tilde{x}(t) + \hat{x}(t), t]/Y(t)\}$$

Then the estimator error can be shown to satisfy the relation

$$\dot{\tilde{x}}(t) = f^*[\tilde{x}(t), \hat{x}(t), t] + G^*[\tilde{x}(t), \hat{x}(t), t]\, w^*(t) \tag{2.10}$$

where

$$f^*[\tilde{x}(t), \hat{x}(t), t] = f[\tilde{x}(t) + \hat{x}(t), t] - \hat{f}[\tilde{x}(t) + \hat{x}(t), t] + K(t)\,\{\hat{h}[\tilde{x}(t) + \hat{x}(t), t] - h[\tilde{x}(t) + \hat{x}(t), t]\}$$

and

$$G^*[\tilde{x}(t), \hat{x}(t), t] = [G[\tilde{x}(t) + \hat{x}(t), t] - K(t)] \begin{bmatrix} w(t) \\ v(t) \end{bmatrix}$$

Furthermore, Equation (2.2) can be rewritten as

$$y(t) = h[\tilde{x}(t) + \hat{x}(t), t] + v(t) \tag{2.11}$$

now the filtering problem of Equations (2.1) and(2.2) has been transformed into that of Equations (2.10) and(2.11).

Let $\phi[\tilde{x}(t)]$ be a twice continuously differentiable function of the vector $\tilde{x}(t)$; by definition of the conditional expectation operator

$$d\,E(\phi[\tilde{x}(t)]) = \phi[\tilde{x}(t)]\, dp[\tilde{x}(t), t/Y(t)]\, d\,\tilde{x}(t) \tag{2.12}$$

where $p[\tilde{x}(t), t/Y(t)]$ is the conditional probability density function.

Next, the change in $p[\tilde{x}(t), t/Y(t)]$ due to the dynamic equations of (2.10) and(2.11) must be computed. It can be shown that [37]

$$\delta p = p[\tilde{x}(t + \delta t), t + \delta t/Y(t), \delta y] - p[\tilde{x}(t), t/Y(t), \delta y] + p[\tilde{x}(t), t/Y(t), \delta y] - p[\tilde{x}(t), t/Y(tY] \tag{2.13}$$

where the first two terms are simply the change due to the dynamic equation of (2.10) and the last two terms are due to the differential measurements δy. These two changes are given by Kolmogorov and Kushner's equation [5,6]. Therefore,

$$\frac{dE(\phi[\tilde{x}(t)])}{dt} = E(L\,\phi[\tilde{x}(t)]) + E\{\phi[\tilde{x}(t)]\{h[\tilde{x}(t) + \hat{x}(t), t] - \hat{h}[\tilde{x}(t) + \hat{x}(t), t]\}\}$$

$$W_v(t)^{-1}(y(t) - \hat{h}[\tilde{x}(t) + \hat{x}(t), t]) \tag{2.14}$$

where

$$L\phi[\hat{x}(t)] = \sum_{i=1}^{n} f_i^*[\tilde{x}(t), \hat{x}(t), t] \frac{\partial \phi[\tilde{x}(t)]}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^{n} \{G^*[\tilde{x}(t), \hat{x}(t), t]$$

$$\Psi_{w*}(t)\ \ G*^{T}[\tilde{x}(t),\hat{x}(t),t]\}_{ij} \cdot \frac{\partial^{2}\phi[\tilde{x}(t)]}{\partial\tilde{x}_i\ \tilde{x}_j} \tag{2.15}$$

Then the error-variance equation is obtained from Equations (2.10), (2.11), (2.14) and 2.(15) by setting $\phi[\tilde{x}(t)] = \tilde{x}(t)\ \ \tilde{x}^{T}(t)$.

for which

$$\frac{dV_{\tilde{x}}(t)}{dt} = E\{\tilde{x}(t)\ f*^{T}[\tilde{x}(t),\hat{x}(t),t] + f*[\tilde{x}(t),\hat{x}(t),t]\ \tilde{x}^{T}(t)\} + G[\tilde{x}(t)+\hat{x}(t),t]$$

$$\Psi_{w}(t)\ G^{T}[\tilde{x}(t)+\hat{x}(t),t] + K(t)\ \Psi_{v}(t)\ \ K^{T}(t) + E\{\tilde{x}(t)\ \ \tilde{x}^{T}(t)\ h^{T}[\tilde{x}(t)+\hat{x}(t),t]$$

$$- V_{\tilde{x}}(t)\ h^{T}[\tilde{x}(t)+\hat{x}(t),t]\}\ \ \Psi_{v}^{-1}(t)\ \{z(t) - \hat{h}[\tilde{x}(t)+\hat{x}(t),t]\} \tag{2.16}$$

where it can be shown that

$$E\{\tilde{x}(t)\ f*^{T}[\tilde{x}(t),\hat{x}(t),t]\} = E\{\tilde{x}(t)\ f^{T}[\tilde{x}(t)+\hat{x}(t),t] - \tilde{x}(t)\ h^{T}[\tilde{x}(t)+\hat{x}(t),t]\ K^{T}(t)\}$$

Now the estimation problem takes the form of an optimal control problem in which K(t) is the only variable available for manipulation such that the cost function of Equation (2.4) is minimized.

To use the matrix minimum principle to solve this problem, a symmetric positive definite costate matrix P(t) is defined, and the Hamiltonian is given by

$$H = \text{trace}\ \{\dot{V}_{\tilde{x}}(t)\ P^{T}(t)\}$$

Using the concept of gradient matrices [38] the necessary condition which the optimal K(t) must satisfy is obtained as

$$K(t) = E\{\tilde{x}(t)\ h^{T}[\tilde{x}(t)+\hat{x}(t),t]\}\ \Psi_{v}^{-1} \tag{2.17}$$

note that K(t) is independent of the costate matrix P(t) and the weighting factor M(t).

Then the filtering estimate and the error-variance equation become

$$\dot{\hat{x}}(t) = \hat{f}[\tilde{x}(t)+\hat{x}(t),t] + E\{\tilde{x}(t)\ h^{T}[\tilde{x}(t)+\hat{x}(t),t]\}\ \Psi_{v}^{-1}(t)\ \{y(t) - \hat{h}[\tilde{x}(t)+\hat{x}(t),t]\} \tag{2.18}$$

and

$$\frac{dV_{\tilde{x}}(t)}{dt} = E\{\tilde{x}(t)\ f^{T}[\tilde{x}(t)+\hat{x}(t),t] + f[\tilde{x}(t)+\hat{x}(t),t]\ \tilde{x}^{T}(t)\}$$

$$+ E\{G[\tilde{x}(t)+\hat{x}(t),t]\ \Psi_{w}(t)\ G^{T}[\tilde{x}(t)+\hat{x}(t),t]\} - E\{\tilde{x}(t)\ h^{T}[\tilde{x}(t)+\hat{x}(t),t]\}$$

$$\Psi_{v}^{-1}(t)\ E\{h[\tilde{x}(t)+\hat{x}(t),t]\ \tilde{x}^{T}(t)\} + E\{\tilde{x}(t)\ \tilde{x}^{T}(t)\ h^{T}[\tilde{x}(t)+\hat{x}(t),t]$$

$$-V_{\tilde{x}}(t)\ h^{T}[\tilde{x}(t)+\hat{x}(t),t]\}\ \Psi_{v}^{-1}(t)\{y(t) - \hat{h}[\tilde{x}(t)+\hat{x}(t),t]\} \tag{2.19}$$

respectively.

It should be noted that before the above algorithms can be physically realized, a number of difficult expectations must be evaluated, they involve infinite dimensional systems except in trivially simple cases. Also notice that the algorithms derived here closely resemble the exact equations due to Bass et al [13]. The only difference is that the argument of the expectations have been transformed into $\tilde{x}(t)$. Such a transformation is particularly significant for filtering problems with polynomial, product-type or state-dependent sinusoidal nonlinearities. Since, in such cases, the filtering algorithms can be obtained without any further approximations, under the assumption that the probability density functions of the estimator errors are Gaussian.

Furthermore, the above derivation can be extended to filtering problems with non-white noise processes. However, in such cases, the change in $p[\tilde{x}(t), t/Y(t)]$ due to the differential measurements δy is neglected, and Equation (2.14) becomes simply

$$\frac{dE\{\phi[\tilde{x}(t)]\}}{dt} = E\{L\,\phi[\tilde{x}(t)]\} \tag{2.20}$$

It is noteworthy to mention that when such an approximation is made, the results obtained for the filtering problem corruped by white noise processes are equivalent to that of the stochastic linearization due to Sunahara [15]. In such a case the random forcing term in the error-variance equation of Equation (2.19) is neglected, whereas the filtering algorithm of Equation (2.18) remains unchanged.

When Equation (2.20) is used in place of Equation (2.14) to obtain the error-variance equation and the nonlinear functions are approximated by second-order Taylor series expansions, the resulting algorithms are commonly called modified minimum variance filter [16].

In order to test the performance of the proposed minimum variance filter and to compare it with various other approximate nonlinear filters [10-13, 15-16], Liang [32] selected various types of nonlinear systems, the stochastic filtering equations are transformed to Stratonovich's forms, and then simulated on a digital computer. The simulation results obtained from the proposed filtering algorithms are compared to various other approximate nonlinear filters. His results indicate the superiority of the proposed filter over those of other filters investigated.

2.3 EVALUATION OF EXPECTATIONS

The purpose of this section is to indicate how polynomial, product-type or state-dependent sinusoidal nonlinearities can be evaluated without the need of approximation under the assumption that the estimation errors are Gaussian.

Polynomial or Product-Type Nonlinearities

For the sake of completeness in presentation, it is worth mentioning that the expectations of higher order power terms can be easily evaluated using the following Lemma.

Lemma 1: Let $\{\tilde{x}(t), t\in T\}$ be a zero-mean Gaussian process. Then all odd order moments of $\tilde{x}$ vanish, and the even order moments can be expressed in terms of the second order moments using the following formula [39]

$$E\{\tilde{x}(t_1)\ldots\tilde{x}(t_n)\} = \Sigma E[\tilde{x}(t_{i_1})\tilde{x}(t_{i_2})]\ldots E[\tilde{x}(t_{i_{n-1}})\tilde{x}(t_{i_n})]$$

where the sum is taken over all possible ways of dividing the n points into n/2 combinations of pairs. The number of terms in the summation is equal to $1.3.5\ldots(n-3)\ (n-1)$.

It should also be noted that Lemma 1 can be rewritten as

$$E\{\tilde{x}(t_1)\ldots\tilde{x}(t_n)\} = E\{\tilde{x}(t_1)\tilde{x}(t_2)\}\,E\{\tilde{x}(t_3)\ldots\tilde{x}(t_n)\} + E\{\tilde{x}(t_1)\tilde{x}(t_3)\}$$
$$E\{\tilde{x}(t_2)\tilde{x}(t_4)\ldots\tilde{x}(t_n)\}\ldots + E\{\tilde{x}(t_1)\tilde{x}(t_n)\}E\{\tilde{x}(t_2)\tilde{x}(t_3)\ldots\tilde{x}(t_{n-1})\}$$

which implies that nth order moments can be obtained from the expectations of n-2th order moments. This is a rather useful formula in the evaluation of the expectatons of higher order power terms.

Nonlinearity Involving State-Dependent Sinusoids

In practical application of estimation techniques, one often encounters nonlinear terms involving state-dependent sinusoids. Fortunately, the expectations of such typeof nonlinear functions can be rigidly obtained.

Lemma 2: Let x_1 and x_2 be jointly normally distributed random variable. Then

$$E\{\cos(x_1 + x_2)\} = \mathrm{Re}\{\exp[j(x_1+x_2) - (V_{11} + V_{22} + 2V_{12})/2]\}$$

and
$$E\{\sin(x_1 + x_2)\} = \mathrm{Im}\{\exp[j(x_1+x_2) - V_{11} + V_{22} + 2V_{12})/2]\}$$

The above relationships are clearly seen from the definition of the characteristic function

$$\zeta_x(u) = E(e^{ju^T x}) = \exp(ju^T x - \tfrac{1}{2}u^T V_x u)$$

where

$$u^T = [u_1, u_2, \ldots u_k, \ldots u_n]$$
$$x^T = [x_1, x_2, \ldots, x_k, \ldots x_n]$$

and $V_x = [\mathrm{cov}(x_i, x_j)]$

Lemma 3: If an operator $I_k(u)$ is defined such that

$$I_k[\zeta_x(u)] = \frac{1}{j}\frac{\partial\zeta_x(u)}{\partial u_k}$$

then we could easily derive the following relationships:

(i) $I_k = \hat{x}_k + ju^T V_x e_k$ where e_k is the kth unit vector,

(ii) $I_k\,\zeta_x(u)\big/_{u=e_n} = (\hat{x}_k + jV_{nk})\,\zeta_x(e_n) = E(x_k e^{jx_n})\big/_{u=e_n}$

where $\zeta_x(e_n) = \exp(jx_n - \frac{V_{nn}}{2})$

(iii) $E(x_k x_\ell e^{jx_n}) = [(\sigma_k + I_k)(\sigma_\ell + I_\ell)]\,\zeta_x(u)\big/_{u=e_n}$

and

(iv) $E(x_1 x_2 \ldots x_n e^{jx_n}) = [(\sigma_1 + I_1)(\sigma_2 + I_2)\ldots(\sigma_n + I_n)]\,\zeta_x(u)\big/_{u=e_n}$

where we have

$$\prod_{i=1}^{\pi} \sigma_i = 0 \qquad \text{odd number}$$

Using these identity relationships, one could easily calculate the expectations of all Gaussian distributed state-dependent sinusoids. For example :

$$E\{x_1 x_2 \cos x_3\} = [V_{12} + (\hat{x}_1 + j V_{31})(\hat{x}_2 + j V_{23})] \, \mathrm{Re}\{\exp(j\hat{x}_3 - \frac{V_{33}}{2})\}$$

$$= \{[V_{12} + \hat{x}_1 \hat{x}_2 - V_{31} V_{23}]\cos \hat{x}_3 - (\hat{x}_1 V_{23} + \hat{x}_2 V_{31}) \sin \hat{x}_3\} e^{-V_{33}/2}$$

and

$$E\{x_1 x_2 \sin x_3\} = \{[V_{12} + (\hat{x}_1 \hat{x}_2 - V_{31} V_{23})] \sin \hat{x}_3 + (\hat{x}_1 V_{23} + \hat{x}_2 V_{31}) \cos \hat{x}_3\} e^{-V_{33}/2}$$

Nonlinearity Involving State-Dependent Relays

Section 2.2 noted that the implementation of nonlinear filtering algorithm requires the evaluation of the expectations of nonlinear vector-valued functions f and h, as well as products of these functions and estimation errors of states.

For a number of applications, f and h may simply be some forms of state-dependent relays. Expectations of some of these state-dependent relays are tabulated in [40].

Illustrative example: Consider a nonlinear system with message and measurement models described by

$$\dot{x}(t) = -\sin x(t) + u(t)$$

and $$y(t) = x^3(t) + v(t)$$

respectively. Where u(t) and v(t) are zero-mean white Gaussian noise processes, with variances $V_u(t)$ and $V_v(t)$, respectively.

Assuming that $\tilde{x}(t)$ is a Gaussian process, then the expectations of the nonlinear functions can be evaluated as follows

$$E\{x^3(t)\} = E\{[\tilde{x}(t) + \hat{x}(t)]^3\} = 3V_{\tilde{x}}(t)\,\hat{x}(t) + \hat{x}^3(t)$$

$$E\{\sin x(t)\} = E\{\sin[\tilde{x}(t) + \hat{x}(t)]\} = \sin \hat{x}(t)\; e^{-V_{\tilde{x}}(t)/2}$$

$$E\{\tilde{x}(t) \sin x(t)\} = -\cos \hat{x}(t)\, V_{\tilde{x}}(t)\, e^{-V_{\tilde{x}}(t)/2}$$

and

$$E\{\tilde{x}^2(t)\, x^3(t)\} = 9V_{\tilde{x}}^2(t)\,\hat{x}(t) + V_{\tilde{x}}(t)\,\hat{x}^3(t)$$

Hence, Equations (2.18) and (2.19) become

$$\dot{\hat{x}}(t) = -\sin \hat{x}(t)\; e^{-V_{\tilde{x}}(t)/2} + [3V_{\tilde{x}}^2(t) + 3V_{\tilde{x}}(t)\,\hat{x}^2(t)] V_v^{-1}[y(t) - 3V_{\tilde{x}}(t)\,\hat{x}(t) - \hat{x}^3(t)]$$

and

$$\dot{V}_{\tilde{x}}(t) = -2V_{\tilde{x}}(t)\, e^{-V_{\tilde{x}}(t)/2} \cos \hat{x}(t) + V_u(t) - [3V_{\tilde{x}}^2(t) + 3V_{\tilde{x}}(t)\,\hat{x}^2(t)]^2 V_v^{-1}(t)$$
$$+ [6V_{\tilde{x}}^2(t)\,\hat{x}(t)]\, V_v^{-1}(t)[y(t) - 3V_{\tilde{x}}(t)\,\hat{x}(t) - \hat{x}^3(t)]$$

respectively.

Note that in the evaluation of the expectations the only assumption needed is the Gaussian assumption of the estimator error, whereas all other finite dimensional algorithms delete some of the higher order terms of the error-variance. When the nonlinear functions in Equations (2.18) and (2.19) are approximated by Taylor series expansions, the results obtained can be identified with various other approximate nonlinear filtering algorithms in the literature [10, 12, 13, 16].

2.4 MINIMUM VARIANCE GENERAL CONTINUOUS NONLINEAR FILTER

Consider the general nonlinear message model

$$\dot{x}(t) = f[x(t),t] + G[x(t),t]\, w(t) + E[x(t),t]\, u(t) \qquad (2.21)$$

with measurement given by

$$y(t) = h[x(t),t] + v(t) + z(t) \qquad (2.22)$$

where u(t) and z(t) are known input time functions, w(t) and v(t) are correlated noise processes with mean $\mu_w(t)$ and $\mu_v(t)$, respectively, and also

$$\mathrm{Cov}\{w(t), v(\tau)\} = V_{wv}(t)\,\delta(t-\tau) \qquad (2.23)$$

and all other prior statistics follow that of Section 2.2

Following the development of Section 2.2 and neglecting the change of $p[\tilde{x}(t), t/Y(t)]$ due to the differential measurements $\delta y(t)$, then the filtering algorithm is given by

$$\dot{\hat{x}}(t) = \hat{f}[\tilde{x}(t) + \hat{x}(t), t] + \hat{G}[\tilde{x}(t) + \hat{x}(t), t]\, \mu_w(t) + \hat{E}[\tilde{x}(t) + \hat{x}(t), t]\, u(t)$$

$$+ K(t)\, \{y(t) - \mu_v(t) - z(t) - \hat{h}[\tilde{x}(t) + \hat{x}(t), t]\} \tag{2.24}$$

where the gain algorithm

$$K(t) = E\{\tilde{x}(t)\, h^T[\tilde{x}(t) + \hat{x}(t), t] + G[\tilde{x}(t) + \hat{x}(t), t]\Psi_{wv}(t)\}\Psi_v^{-1}(t) \tag{2.25}$$

and the error-variance equation is given by

$$\dot{V}_{\underset{x}{\sim}}(t) = E\{\tilde{x}(t)\, f^T[\tilde{x}(t) + \hat{x}(t), t] + f[\tilde{x}(t) + \hat{x}(t), t]\, \tilde{x}^T(t)\} + E\{G[\tilde{x}(t) + \hat{x}(t), t]$$

$$\Psi_w(t) G^T[\tilde{x}(t) + \hat{x}(t), t]\} - K(t)\, \Psi_v(t)\, K^T(t) \tag{2.26}$$

In the particular case that f and h are linear, G and E are independent of the state variable, namely

$$f[x(t), t] = F(t)\, x(t)$$

$$h[x(t), t] = H(t)\, x(t)$$

$$G[x(t), t] = G(t) \text{ and}$$

$$E[x(t), t] = E(t)$$

Then Equations (2.24)-(2.26) are, respectively,

$$\dot{\hat{x}}(t) = F(t)\, \hat{x}(t) + G(t)\mu_w(t) \quad + E(t)\, u(t) + K(t)\, \{y(t) - \mu_v(t) - z(t) - H(t)\, \hat{x}(t)\}$$

$$K(t) = [V_{\underset{x}{\sim}}(t)\, H^T(t) + G(t)\, \Psi_{wv}(t)]\Psi_v^{-1}(t)$$

and

$$\dot{V}_{\underset{x}{\sim}}(t) = F(t)\, V_{\underset{x}{\sim}}(t) + V_{\underset{x}{\sim}}(t)\, F^T(t) + G(t)\, \Psi_w(t)\, G^T(t) - K(t)\Psi_v(t)\, K^T(t)$$

These algorithms agree well with the general continuous Kalman filter [36].

2.5 MINIMUM VARIANCE CONTINUOUS NONLINEAR NOISE-FREE FILTERING

Consider the continuous nonlinear message model of Equation (2.1) with the noise-free measurement model given by

$$y(t) = h[x(t), t] \tag{2.27}$$

where w(t) is zero-mean, white noise with non-negative definite variance $\Psi_w(t)$. $h[x(t), t]$ is assumed to be continuously differentiable in t and has continuous second mixed partial derivatives with respect to the elements of x and also considered to be of full rank, otherwise an equivalent y(t) of lower dimension can be used.

Since y(t) is noise free $\Psi_w(t)$ is non-negative definite, when y(t) is differentiated, some of its elements may not contain any white noise. Therefore, each element of y(t) has to be differentiated as in

$$\frac{d}{dt} y_i = \frac{\partial}{\partial t} h_i + \frac{\partial}{\partial x_i} h \frac{d}{dt} x + \frac{1}{2} \text{trace}[G \Psi_w G^T (\frac{\partial^2}{\partial x_i^2} h_i)]$$

and Equation (2.1) is used to substitute for the derivative of the state variable, until white noise is obtained in the derivatives of each element in y(t).

The signals obtained can be arranged into two sets. In the first set

$$y_1(t) = h_1[x(t), \Psi_w(t), t] + N[x(t), t] w(t)$$

which comprises all derivatives of y(t) that contain linearly independent white noise.

It is assumed that the filtering estimate $\hat{x}(t)$ is given by

$$\dot{\hat{x}}(t) = \hat{f}[\tilde{x}(t) + \hat{x}(t), t] + K_1(t)\, y_1(t) + K_2(t)\, y_2(t)$$

since in place of Equation (2.27), there are two sets of measurements.

Here, $y_1(t)$ is considered as a known input, it does not contain any new information, and following the development of Section 2.2, the following filtering algorithm is obtained

$$\dot{\hat{x}}(t) = \hat{f}[\tilde{x}(t) + \hat{x}(t), t] + K_2(t)\{y_2(t) - \hat{h}_2[\tilde{x}(t) + \hat{x}(t), \Psi_w(t), t]\}$$

with the gain algorithm

$$K_2(t) = E\{\tilde{x}(t)\ h_2^T[\tilde{x}(t)+\hat{x}(t),\Psi_w(t),\ t] + G[\tilde{x}(t)+\hat{x}(t),t]\Psi_w(t)\quad N^T[\tilde{x}(t)+\hat{x}(t),t]\}$$

$$[E\{N[\tilde{x}(t)+\hat{x}(t),t]\Psi_w(t)\ N^T[\tilde{x}(t)+\hat{x}(t),t]\}]^{-1}$$

and the error-variance equation is given as

$$\dot{V}_{\tilde{x}}(t) = E\{\tilde{x}(t)\ f^T[\tilde{x}(t)+\hat{x}(t),t]+ f[\hat{x}(t)+\hat{x}(t),\ t]\ \tilde{x}^T(t)\}+E\{G[\tilde{x}(t)+\hat{x}(t),t]$$

$$\Psi_w(t)\ G^T[\tilde{x}(t)\ +\hat{x}(t),t]\} - K_2(t)E\{N[\tilde{x}(t)\ +\hat{x}(t),t]\Psi_w(t)N^T[\tilde{x}(t)+\hat{x}(t),t]\}K_2^T(t)$$

In the case that the vector valued functions are linear, the results presented here agree well with that of Bryson and Johansen [35].

SECTION 3

MINIMUM VARIANCE FILTERING AND SMOOTHING FOR DISCRETE NONLINEAR DELAYED SYSTEMS WITH ADDITIVE WHITE NOISE

3.1 INTRODUCTION

This section deals with discrete-time filtering and smoothing estimation of nonlinear systems with multiple delays imbedded in additive white noise processes. In general, the filtering algorithm enables one to estimate present values of the variables of interest using present data, whereas the smoother allows one to estimate past values. A typical smoothing problem is the post-flight estimation of the flight path of a missile based on tracking system measurements during the entire duration of the flight. If the estimates of the missile's position and velocity at one particular flight point are desired, the estimates can be based upon all the measurements recorded, including those made before and after that particular flight point.

In the estimation problems for linear systems without delays involving additive white noise processes, numerous papers have been written to deal with filtering and smoothing estimation, however, most of them are merely rederivation of earlier recursive algorithms presented by Carlton [41], Rauch [42] and Bryson and Frazier [43], or reformulation of problems using various estimation techniques.

However, Kelly and Anderson [44] pointed out that the algorithms for both discrete and continuous-time, linear, fixed-lag smoothing given in [43 , 18, 42, 45] may be unstable, and therefore impractical. To be more explicit, although the fixed-lag smoothing equations are bounded-input and bounded-output stable, realizations of these in [42,45] contain a subsystem which is unstable in the sense of Lyapunov. In [44], it is pointed out that the apparent culprit is an uncontrollable and unstable block in the smoother state equations which can be removed without affecting the input-output characteristics.

In[23], a computationally stable smoothing algorithm is derived for linear discrete systems containing time delays, using the method of orthogonal projection. The smoother for linear discrete systems without delays can be considered as a special case of the above problem, with time delay index setting to zero. The results of [23] are rederived in a simple manner in [24], using the state augmentation technique. However, the smoothers derived in such a manner are of nN dimensions, where n is the order of the message model and N is the amount of the fixed-lag.

In this section, the matrix minimum principle is applied to nonlinear discrete-time systems involving time delays, with measurement sequence imbedded in additive white noise processes. The resulting dynamic discrete estimation algorithms, as reported in [28, 46], are recursive in nature and directly yield the fixed-interval, fixed-lag, fixed-point smoothing and the filtering algorithms. The derivation is straight forward and shows the close links between the smoothing and filtering estimation algorithms.

In Section 3.2, the problem statement is presented. Section 3.3 presents the derivation of the nonlinear smoother. Sections 3.4 to 3.7, provide handy sets of reference equations respectively, for the fixed-lag, fixed-point, fixed-interval smoothing and the filtering algorithms. Section 3.8 shows the applicability of the presented algorithms to linear estimation problems.

3.2 THE PROBLEM STATEMENT

The message and measurement models for the discrete nonlinear time-delayed systems are given by

$$x(k+1) = \sum_{j=0}^{k} f_j[x(k-\alpha_j),\ k-\alpha_j] + G[x(k),k]w(k) \tag{3.1}$$

and

$$y(k) = h[x(k),k] + v(k) \tag{3.2}$$

Here x the state is an n-vector;y the measurement, an m-vector;w the random input, an r-vector; v the measurement noise, an m-vector; G, a nonlinear state dependent n x r matrix;k= 0,1, ... is the discrete time index. The nonlinear vector valued functions f_j and h are, respectively, n and m - dimensional.

The integer quantities α_j represent time delays which are ordered such that

$$\alpha_0 < \alpha_1 < \alpha_2 < \cdots < \alpha_L$$

The random vectors w(k) and v(k) are independent zero-mean white Gaussian sequences, for which

$$E\{w(k)\ w^T(j)\} = \Psi_w(k)\ \delta_{kj}$$

$$E\{v(k)\ v^T(j)\} = \Psi_v(k)\ \delta_{kj}$$

and

$$E\{w(k)\ v^T(j)\} = 0$$

for all integers k and j, where E{.} denotes the expectation operator, δ_{jk} the Kronecker delta, and Ψ_v and Ψ_w are m x m and r x r positive definite matrices, respectively.

The initial states x(0) and $x(-\alpha_j)$ for j = 1, ..., L are zero mean Gaussian random vectors, independent of v(k) and w(k), with a positive definite covariance matrix

$$E\{x(-\alpha_j)x^T(-\alpha_\ell)\} = V_x(\alpha_j, \alpha_\ell)$$

for $j,\ell = 0, \ldots, L$.

The smoothing problem is to obtain $\hat{x}(k-\ell+1/k+1)$, the unbiased smoothing estimate of $x(k-\ell+1)$, with $0 \leq \ell \leq k+1$, conditioned on the set of measurements

$$Y(k+1) = \{y(0), y(1), \ldots, y(k+1)\}$$

such that the cost function

$$J(k+1) = \text{Trace}\ [M(k)\ \underset{\sim}{V}_x(k-\ell+1/k+1)] \tag{3.3}$$

is minimized. Here M(k) is some symmetric non-negative definite weighting matrix, and $\underset{\sim}{V}_x(k-\ell+1/k+1)$ is

$$\underset{\sim}{V}_x(k-\ell+1/k+1) = E_{k+1}[\{x(k-\ell+1) - \hat{x}(k-\ell+1/k+1)\}$$
$$\{x(k-\ell+1) - \hat{x}(k-\ell+1/k+1)\}^T] \tag{3.4}$$

where $E_{k+1}\{.\}$ denotes the expectation operation conditioned on the set of measurements Y(k+1).

3.3 THE DERIVATION OF NONLINEAR SMOOTHING ALGORITHMS

The smoothing algorithm is assumed to be constrained by the nonlinear differential equation

$$\hat{x}(k-\ell+1/k+1) + \sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\alpha_j/k),\ k-\ell-\alpha_j] + K_{k+1}^{\ell} y(k+1) \tag{3.5}$$

Here, the assumption of linearity in innovations is made. The nonlinear functions $\sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\alpha_j/k),\ k-\ell-\alpha_j]$ and K_{k+1}^{ℓ} are yet to be determined.

In fact, the smoothing equation may take various forms;however, it is essential that enough information concerning $\hat{x}(k-\ell-\alpha_j/k)$ and y(k+1) are included in the estimator moded. For example, it can be shown that other dynamic equations such as

$$\hat{x}(k-\ell+1/k+1) = \sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\alpha_j/k),\ k-\ell-\alpha_j] + K_{k+1}^{\ell}\tilde{y}(k+1/k) \tag{3.6}$$

where $\tilde{y}(k+1/k) = y(k+1) - E_k\{y(k+1)\}$ would result in exactly the same smoothing algorithm as the one constrained by Equation (3.5).

Since the systems considered here are non-linear with multiple time delays, it is realistic to assume that the smoothing estimate is a linear combination of the sum of nonlinear functions of $\hat{x}(k-\ell-\alpha_j/k)$, to account for the time delay characteristics, and the present measurement y(k+1). Here $\hat{x}(k-\ell-\alpha_j/k)$ is assumed to have made optimum use of all the measurements up to y(k).

The problem formulated in such a manner may therefore lead to a smoother, optimal with respect to the imposed constraints, but not identical to the truly optimal one.

Now the estimation problem is to determine the time-varying nonlinear vector function $\sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\alpha_j/k),\ k-\ell-\alpha_j]$, and the algorithm K_{k+1}^{ℓ} such that the trace of $\underset{\sim}{V}_x(k-\ell+1/k+1)$ is minimized.

Let $\tilde{x}(k-\ell+1/k+1)$ denote the smoothing error defined by

$$\tilde{x}(k-\ell+1/k+1) = x(k-\ell+1) - \hat{x}(k-\ell+1/k+1) \tag{3.7}$$

Then we obtain

$$\tilde{x}(k-\ell+1/k+1) = \sum_{j=0}^{L} f_j[x(k-\ell-\alpha_j),\ k-\ell-\alpha_j] + G[x(k-\ell),k-\ell\ ]w(k-\ell)$$
$$- \sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\alpha_j/k),\ k-\ell-\alpha_j] - K_{k+1}^{\ell}\{h[x(k+1),k+1] + v(k+1)\} \tag{3.8}$$

In order that $\hat{x}(k-\ell+1/k+1)$ is an unbiased smoothing estimate, it is necessary that

$$\sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\alpha_j/k),\ k-\ell-\alpha_j] = \sum_{j=0}^{L} \hat{f}_j[x(k-\ell-\alpha_j),k-\ell-\alpha_j/k] - K_{k+1}^{\ell}\ \hat{h}[x(k+1),\ k+1/k] \tag{3.9}$$

where

$$\hat{f}_j[x(k-\ell-\alpha_j),\ k-\ell-\alpha_j/k] = E_k\{f_j[x(k-\ell-\alpha_j),\ k-\ell-\alpha_j]\}$$

and

$$\hat{h}[x(k+1),\ k+1/k] = E_k\{h[x(k+1),\ k+1]\}$$

Substituting Equations (3.9) into (3.5), the smoothing algorithm becomes

$$\hat{x}(k-\ell+1/k+1) = \hat{x}(k-\ell+1/k) + K^{\ell}_{k+1}\{y(k+1) - \hat{h}[x(k+1),\ k+1/k]\} \tag{3.10}$$

for $k = 0,1, \ldots,$ and $0 \le \ell \le k+1$.

Where

$$\hat{x}(k-\ell+1/k) = \sum_{j=0}^{L} \hat{f}_j[x(k-\ell-\alpha_j),\ k-\ell-\alpha_j/k] \tag{3.11}$$

Substitution of Equations (3.10) into(3.7), yields

$$\tilde{x}(k-\ell+1/k+1) = \tilde{x}(k-\ell+1/k) - K^{\ell}_{k+1}\{\tilde{h}[x(k+1)/k] + v(k+1)\} \tag{3.12}$$

where

$$\tilde{h}[x(k+1)/k] = h[x(k+1),\ k+1] - \hat{h}[x(k+1),\ k+1/k] \tag{3.13}$$

Therefore, the smoothing error-variance equation is given by

$$V_{\tilde{x}}(k-\ell+1/k+1) = V_{\tilde{x}}(k-\ell+1/k) - E_k\{\tilde{x}(k-\ell+1/k)\ \tilde{h}^T[x(k+1)/k]\}\ K^{\ell T}_{k+1} - K^{\ell}_{k+1}$$

$$E_k\{\tilde{h}[x(k+1)/k]\ \tilde{x}^T(k-\ell+1/k)\} + K^{\ell}_{k+1}\ \Psi_v(k+1)\ K^{\ell T}_{k+1}$$

$$+ K^{\ell}_{k+1} E_k\{\tilde{h}[x(k+1)/k]\ \tilde{h}^T[x(k+1)/k]\} K^{\ell T}_{k+1} \tag{3.14}$$

Here, K^{ℓ}_{k+1} is the only variable available for manipulation. The necessary condition for minimizing the trace of $V_{\tilde{x}}(k-\ell+1/k+1)$ and subject to the constraint of Equation (3.14) is provided by the matrix minimum principle [29], for which K^{ℓ}_{k+1} is considered as the control variable.

The necessary condition can now be otained from the condition

$$\frac{\partial}{\partial K^{\ell}_{k+1}} \text{Trace}[V_{\tilde{x}}(k-\ell+1/k+1)] = [0] \tag{3.15}$$

where [0] is the null matrix, and the result is

$$K^{\ell}_{k+1} = E_k\{\tilde{x}(k-\ell+1/k)\ \tilde{h}^T[x(k+1)/k]\}\ [\Psi_v(k+1) + E_k\{\tilde{h}[x(k+1)/k]\ \tilde{h}^T[x(k+1)/k]\}^{-1} \tag{3.16}$$

and Equation (3.14) is reduced to

$$V_{\tilde{x}}(k-\ell+1/k+1) = V_{\tilde{x}}(k-\ell+1/k) - K^{\ell}_{k+1} E_k\{\tilde{h}[x(k+1)/k]\tilde{x}^T(k-\ell+1/k)\} \tag{3.17}$$

Using Equations (3.1) and(3.11), there is the relation

$$\tilde{x}(k-\ell+1/k) = \sum_{j=0}^{L} \gamma_j[x(k-\ell-\alpha_j)/k] + G[x(k-\ell),k-\ell]w(k-\ell)$$

where

$$\gamma_j[x(k-\ell-\alpha_j)/k] = f_j[x(k-\ell-\alpha_j),\ k-\ell-\alpha_j] - \hat{f}_j[x(k-\ell-\alpha_j),\ k-\ell-\alpha_j/k]$$

and

$$E_k\{\tilde{x}(k-\ell+1/k)\tilde{h}^T(x(k+1)/k]\} = E_k\{\sum_{j=0}^{L} \gamma_j[x(k-\ell-\alpha_j)/k]\tilde{h}^T[x(k+1)/k]\}$$

Then the following recursive smoothing algorithms can also be derived:

$$V_{\tilde{x}}(k-\ell+1,k/k) = E_k\{\sum_{j=0}^{L} \gamma_j[x(k-\ell-\alpha_j)/k]\tilde{x}^T(k/k)\} \tag{3.18}$$

$$V_{\tilde{x}}(k-\ell+1,k-m+1/k) = \sum_{i=0}^{L}\sum_{j=0}^{L} E_k\{\gamma_i(k-\ell-\alpha_i)/k]\ \gamma^T_j[x(k-m-\alpha_j)/k]\} + G[x(k-\ell),\ k-\ell]\ \Psi_w(k-\ell)\ G^T[x(k-m),k-m]$$

$$\delta_{k-\ell,k-m} \tag{3.19}$$

and

$$V_{\tilde{x}}(k-\ell+1,k-m+1/k+1) = V_{\tilde{x}}(k-\ell+1,\ k-m+1/k) - K^{\ell}_{k+1} E_k\{\tilde{h}[x(k+1)/k]\tilde{x}^T(k-m+1/k)\} \tag{3.20}$$

for $0 \le \ell, m \le k+1$.

For $k<0$, there is no input to the smoother and therefore one can set $\hat{x}(-\alpha_j/-1)$ to zero for $j = 0,1, \ldots, L$, which in turn leads to

$$\tilde{x}(-\alpha_j/-1) = x(-\alpha_j)$$

and

$$V_{\tilde{x}}(-\alpha_j, -\alpha_\ell/-1) = V_x(\alpha_j, \alpha_\ell)$$

for $j,\ell = 0,1, \ldots, L$.

Since the smoothing estimator is unbiased, the expectations that are in Equations (3.16)-(3.20) can be replaced by the following

$$E_k\{\tilde{x}(k-\ell+1/k)\ \tilde{h}^T[x(k+1)/k]\} = E_k\{\tilde{x}(k-\ell+1/k)h^T[x(k+1), k+1]\}$$

$$E_k\{\tilde{h}[x(k+1)/k]\tilde{h}^T[x(k+1)/k]\} = E_k\{h[x(k+1), k+1]h^T[x(k+1), k+1]\} - \hat{h}[x(k+1),k+1/k]\hat{h}^T[x(k+1),k+1/k] \tag{3.21}$$

and

$$E_k\{\tilde{f}_i[x(k-\ell-\alpha_i)/k]\ \tilde{f}^T_j[x(k-\ell-\alpha_j)/k]\} = E_k\{f_i[x(k-\ell-\alpha_i), k-\ell-\alpha_i]$$

$$f_j^T[x(k-\ell-\alpha_j), k-\ell-\alpha_j]\} - \hat{f}_i[x(k-\ell-\alpha_i), k-\ell-\alpha_i/k]\ \hat{f}_j^T[x(k-\ell-\alpha_j), k-\ell-\alpha_j/k] \tag{3.22}$$

It should be noted that in the case of nonlinear systems the above expectations require infinite dimensional systems for their realization. The problem of obtaining a good approximation to the above expectations is, therefore, of practical importance.

For practical realization, it is assumed that the conditional probability density functions of the smoothing error $\tilde{x}$ are Gaussian, then the expectation can be obtained without any further approximation for systems with polynomial, product-type or state-dependent sinusoidal nonlinearities.

To evaluate the above expectation, one can make use of Equation (3.7) to replace the state variables by the sum of their respective estimators and the estimator errors. For example, in the scalar case of

$$h[x(k+1), k+1] = x^3(k+1)$$

it can be shown that

$$E_k\{\tilde{x}(k-\ell+1/k)\ \tilde{h}^T[x(k+1)/k]\} = E_k\{\tilde{x}(k-\ell+1/k)\quad [\tilde{x}^3(k+1/k) + 3\,\tilde{x}^2(k+1/k)$$

$$\hat{x}(k+1/k) + 3\,\tilde{x}(k+1/k)\quad \hat{x}^2(k+1/k) + \hat{x}^3(k+1/k)]\}$$

$$= V_{\tilde{x}}(k-\ell+1, k+1/k)\ \hat{x}^2(k+1/k) + 3\,V_{\tilde{x}}(k-\ell+1,k+1/k)\ V_{\tilde{x}}(k+1/k)$$

Notice that in the evaluation of the expectation, the only assumption needed is the Gaussian assumption of the estimator error. And it can be easily shown that if the nonlinearities were expanded in terms of first or second order Taylor,s series, the last term in the preceeding equation would have been dropped.

Similarly, it is obtained from Equations (3.21) and (3.7)

$$E_k\{\tilde{h}[x(k+1)/k]\ \tilde{h}^T[x(k+1)/k]\} = E_k\{x^6(k+1)\} - E_k\{x^3(k+1)\}^2 = [15\,V^3_{\tilde{x}}(k+1/k)$$

$$+ 45\,V^2_{\tilde{x}}(k+1/k)\ \hat{x}^2(k+1/k) + 15\,V_{\tilde{x}}(k+1/k)\quad \hat{x}^4(k+1/k) + \hat{x}^6(k+1/k)]$$

$$- [\hat{x}^3(k+1/k) + 3\,V_{\tilde{x}}(k+1/k)\quad \hat{x}(k+1/k)]^2 = 15\,V^3_{\tilde{x}}(k+1/k)$$

$$+ 36\,V^2_{\tilde{x}}(k+1/k)\ \hat{x}^2(k+1/k) + 9\,V_{\tilde{x}}(k+1/k)\ \hat{x}^4(k+1/k)$$

The preceeding evaluation is rather straightforward. In the particular case that higher-order terms do not improve system accuracy, one can delete the higher-order terms and approximate the expectations up to the arbitrary order, as desired.

3.4 THE NONLINEAR FIXED-LAG SMOOTHING

Replace k+1 and ℓ by k and N, respectively, where $N<k$, from Equations (3.10),(3.11) and (3.16)-(3.20) one would then obtain the following recursive nonlinear fixed-lag smoothing algorithms:

$$\hat{x}(k-N/k) = \hat{x}(k-N/k-1) + K^N_k\{y(k) - \hat{h}[x(k), k/k-1]\} \tag{3.23}$$

$$\hat{x}(k-N/k-1) = \sum_{j=0}^{L} \hat{f}_j[x(k-N-1-\alpha_j), k-N-1-\alpha_j/k-1] \tag{3.24}$$

$$K^N_k = E_{k-1}\{\tilde{x}(k-N/k-1)\ \tilde{h}^T[x(k)/k-1]\}[V_v(k) + E_{k-1}\{\tilde{h}[x(k)/k-1]\ \tilde{h}^T[x(k)/k-1]\}]^{-1} \tag{3.25}$$

$$V_{\underset{\sim}{x}}(k-N/k) = V_{\underset{\sim}{x}}(k-N/k-1) - K_k^N E_{k-1}\{\hat{\underset{\sim}{h}}[x(k)/k-1]\ \hat{\underset{\sim}{x}}^T(k-N/k-1)\} \tag{3.26}$$

$$V_{\underset{\sim}{x}}(k-\ell\ \ k-m/k) = V_{\underset{\sim}{x}}(k-\ell,\ k-m/k-1) - K_k^{\ell} E_{k-1}\{\hat{\underset{\sim}{h}}[x(k)/k-1]\hat{\underset{\sim}{x}}^T(k-m/k-1)\} \tag{3.27}$$

and

$$V_{\underset{\sim}{x}}(k-\ell,\ k-m/k-1) = \sum_{i=o}^{L}\sum_{j=o}^{L} E_k\{\hat{\underset{\sim}{f}}_i[x(k-1-\ell-\alpha_i)/k-1]\ \hat{\underset{\sim}{f}}_j^T[x(k-1-m-\alpha_j)/k-1]\} + G[x(k-1-\ell),k-1-\ell]$$
$$\Psi_w(k-1-\ell)G^T[x(k-1-m),k-1-m]\delta_{m,n}$$

Also we have

$$V_{\underset{\sim}{x}}(k-N,\ k+1/k) = E_k\{\hat{\underset{\sim}{x}}(k-N/k) \sum_{j=o}^{L} \hat{\underset{\sim}{f}}_j^T[x(k-\alpha_j)/k]\} \tag{3.28}$$

and

$$V_{\underset{\sim}{x}}(k+1/k) = \sum_{i=o}^{L}\sum_{j=o}^{L} E_k\{\hat{\underset{\sim}{f}}_i[x(k-\alpha_i)/k]\ \hat{\underset{\sim}{f}}_j^T[x(k-\alpha_j)/k]\} + G[x(k),\ k]\ \Psi_w(k)\ G^T[x(k),k] \tag{3.29}$$

3.5 THE NONLINEAR FIXED-POINT SMOOTHING

Setting $\ell = K - N+1$ where $k+1 > N$, from Equations (3.10),(3.11) and (3.16)-(3.20) one would then have the following recursive non-linear fixed-point smoothing algorithms:

$$\hat{x}(N/k+1) = \hat{x}(N/k) + K_{k+1}^{k-N+1}\ \{y(k+1) - \hat{h}[x(k+1),\ k+1/k]\}$$

$$K_{k+1}^{k-N+1} = E_k\{\hat{\underset{\sim}{x}}(N/k)\ \hat{\underset{\sim}{h}}^T[x(k+1)/k]\}\ [\Psi_v(k+1) + E_k\{\hat{\underset{\sim}{h}}[x(k+1)/k]\ \hat{\underset{\sim}{h}}^T[x(k+1)/k]\}]^{-1}$$

$$V_{\underset{\sim}{x}}(N/k+1) = V_{\underset{\sim}{x}}(N/k) - K_{k+1}^{k-N+1} E_k\{h[x(k+1)/k]\ \hat{\underset{\sim}{x}}^T(N/k)\}$$

$$V_{\underset{\sim}{x}}(N/k) = \sum_{i=o}^{L}\sum_{j=o}^{L} E_k\{\hat{\underset{\sim}{f}}_i[x(N-1-\alpha_i)/k]\ \hat{\underset{\sim}{f}}_j^T[x(N-1-\alpha_j)/k]\} + G[x(N-1),N-1]\Psi_w(N-1)\ G^T[x(N-1),N-1]$$

and

$$V_{\underset{\sim}{x}}(N,\ k+1/k) = E_k\{\hat{\underset{\sim}{x}}(N/k) \sum_{j=o}^{L} \hat{\underset{\sim}{f}}_j^T[x(k-\alpha_j)/k]\}$$

$V_{\underset{\sim}{x}}(k+1/k)$ and $V_{\underset{\sim}{x}}(k-\ell,\ k-m/k)$ are respectively, the same as Equations (3.28) and (3.27) and also from Equation (3.20), it can be shown that

$$V_{\underset{\sim}{x}}(N,\ k-m/k) = V_{\underset{\sim}{x}}(N,\ k-m/k-1) - K_k^{k-N} E_{k-1}\{\hat{\underset{\sim}{h}}[x(k)/k-1]\hat{\underset{\sim}{x}}^T(k-m/k-1)\}$$

3.6 THE NONLINEAR FIXED-INTERVAL SMOOTHING

Setting $k+1=N$, $\ell=N-k$, where $N\geq k$, Equations (3.10),(3.11) and (3.16)-(3.20) would yield the following recursive nonlinear fixed-interval smoothing algorithms:

$$\hat{x}(k/N) = \hat{x}(k/N-1) + K_N^{N-k}\ \{y(N) - \hat{h}[x(N),\ N/N-1]\}$$

$$\hat{x}(k/N-1) = \sum_{j=o}^{L} \hat{f}_j[x(k-1-\alpha_j),\ k-1-\alpha_j/N-1]$$

$$K_N^{N-k} = E_{N-1}\{\hat{\underset{\sim}{x}}(k/N-1)\ \hat{\underset{\sim}{h}}^T[x(N)/N-1]\}\ [\Psi_v(N) + E_{N-1}\{\hat{\underset{\sim}{h}}[x(N)/N-1]\ \hat{\underset{\sim}{h}}^T[x(N)/N-1]\}]^{-1}$$

$$V_{\underset{\sim}{x}}(k/N) = V_{\underset{\sim}{x}}(k/N-1) - K_N^{N-k}\ E_{N-1}\ \{\hat{\underset{\sim}{h}}[x(N)/N-1]\ \hat{\underset{\sim}{x}}^T\ (k/N-1)\}$$

$$V_{\underset{\sim}{x}}(k/N-1) = \sum_{i=o}^{L}\sum_{j=o}^{L} E_{N-1}[\hat{\underset{\sim}{f}}_i[x(k-1-\alpha_i)/N-1]\ \hat{\underset{\sim}{f}}^T{}_j[x(k-1-\alpha_j)/N-1]] + G[x(k-1),k-1]\Psi_w(k-1)G^T[x(k-1),\ k-1]$$

and

$$V_{\underset{\sim}{x}}(k,\ N/N-1) = E_{N-1}\{\hat{\underset{\sim}{x}}(k/N-1) \sum_{i=o}^{L} \hat{\underset{\sim}{f}}_i^T\ [x(N-1-\alpha_i)/N-1]\}$$

3.7 THE NONLINEAR FILTERING

The nonlinear filtering algorithm can be easily obtained from Equations (3.10),(3.11) and (3.16)-(3.20), by setting $\ell=0$.

$$\hat{x}(k+1/k+1) = \hat{x}(k+1/k) + K_{k+1}^{0}\{y(k+1) - \hat{h}[x(k+1),\ k+1/k]\}$$

$$\hat{x}(k+1/k) = \sum_{j=o}^{L} \hat{f}_j[x(k-\alpha_j),\ k-\alpha_j/k]$$

$$K^{0}_{k+1}=E_k\{\tilde{x}(k+1/k)\ \tilde{h}^{T}[x(k+1)/k]\}\ [\Psi_v(k+1)+ E_k\{\tilde{h}[x(k+1)/k]\ \tilde{h}^{T}[x(k+1)/k]\}]^{-1}$$

and

$$V_{\tilde{x}}(k+1/k+1) = V_{\tilde{x}}(k+1/k)-K^{o}_{k+1}E_k\{\tilde{h}[x(k+1)/k]\ \tilde{x}^{T}(k+1/k)\}$$

whereas

$V_{\tilde{x}}(k+1/k)$ is given by Equation (3.29)

3.8 ESTIMATION IN LINEAR DISCRETE SYSTEMS

In order to provide an insight into the structure of the smoothed estimate, consider the particular case of linear systems and measurements with

$$\sum_{j\ o}^{L} f_j[x(k-\alpha_j),k-\alpha_j]=\sum_{j=o}^{L} F_j(k)\ x(k-\alpha_j)$$

$$h[x(k),k] = H(k)\ x(k)$$

and

$$G[x(k),k] = G(k)$$

Then the linear fixed-lag smoothing algorithm can be directly obtained from Equations (3.23) to (3.28), respectively as the following

$$\hat{x}(k-N/k) = \hat{x}(k-N/k-1) + K^{N}_{k}\{y(k) - H(k)\ \hat{x}(k/k-1)\}$$

$$\hat{x}(k-N/k-1) = \sum_{j=o}^{L} F_j(k-N-1)\ \hat{x}(k-N-1-\alpha_j/k-1)$$

$$K^{N}_{k} = V_{\tilde{x}}(k-N,k/k-1)\ H^{T}(k)\{H(k)\ V_{\tilde{x}}(k/k-1)\ H^{T}(k)+ \Psi_v(k)\}^{-1}$$

$$V_{\tilde{x}}(k-N/k) = V_{\tilde{x}}(k-N/k-1) - K^{N}_{k}\ H(k)\ V_{\tilde{x}}(k,k-N/k-1)$$

and

$$V_{\tilde{x}}(k-\ell,k-m/k-1) = \sum_{i=o}^{L}\sum_{j=o}^{L} F_i(k-\ell-1)\ V_{\tilde{x}}(k-1-\ell-\alpha_i,k-1-m-\alpha_j/k-1)$$
$$F^{T}_{j}(k-m-1) + G(k-1-\ell)\ \Psi_w(k-1-\ell)\ G^{T}(k-1-\ell)\delta_{k-\ell,k-m}$$

Also $$V_{\tilde{x}}(k-N,k+1/k) = \sum_{j=o}^{L} V_{\tilde{x}}(k-N,k-\alpha_j/k)\ F^{T}_{j}(k)$$

and finally,

$$V_{\tilde{x}}(k+1/k) = \sum_{i=o}^{L}\sum_{j=o}^{L} F_i(k)\ V_{\tilde{x}}(k-\alpha_i,k-\alpha_j/k)F^{T}_{j}(k) + G(k)\ \Psi_w(k)\ G^{T}(k)$$

It can be easily identified that the structure of the above linear fixed-lag smoother is simply the stable fixed-lag smoother introduced by Premier and Vacroux [23]. In the same manner, one can identify the linear fixed-point smoother for linear systems without delays with that of Biswas and Mahalanabis [47]. Similarly, the results presented in Section 3.6 and 3.7 can be easily applied to yield the linear fixed-interval smoothing and the filtering algorithms, respectively.

Thus, a unified approach to obtain the filtering and smoothing algorithms for linear as well as nonlinear, delayed as well as nondelayed systems has been presented.

The results presented in this section can be extended to continuous time problems through a formal limiting procedure [48]. The presented approach can also be extended to nonlinear distributive systems with or without delays.

SECTION 4

MINIMUM VARIANCE FILTERING AND SMOOTHING FOR NONLINEAR SYSTEMS WITH CORRELATED NOISES

4.1 INTRODUCTION

In Section 3 the noise processes considered are assumed to be Gaussian white and mutually independent, however, in practical situations such assumptions are often invalid, since in physical systems, independent white noise processes simply do not exist. It is therefore natural to extend the estimation technique developed in Section 3 to more realistic problems, where message noise processes are correlated with measurement noise processes.

Recently, Raja Rao and Mahalanabis [49] derived estimation algorithms for linear systems with delay imbedded in correlated noise processes, however, their results appear to have a fundamental mistake in the procedure given, which leads to self-contradictory results; this is reported in [50].

In this section, estimation algorithms are derived for nonlinear discrete delayed systems with measurements imbedded in correlated noise processes. The derivation assumes that the smoothing estimator introduces new data in a linear additive fashion and makes use of the matrix minimum principle to minimize the error-variance cost functional. The resulting estimation algorithms as reported in [28,51], can be easily applied to yield the fixed-lag, fixed-point, fixed-interval smoothing and filtering algorithms, by properly substituting the time indices.

4.2 THE PROBLEM STATEMENT

Consider a discrete nonlinear time-delayed system modeled by

$$x(k+1) = \sum_{j=0}^{L} f_j[x(k-\alpha_j),\ k-\alpha_j] + G[x(k),\ k]w(k) \tag{4.1}$$

with measurement given by

$$y(k) = h[x(k),\ k] + v(k) \tag{4.2}$$

where the state x is an n-vector; the measurement y an m-vector; the state noise sequence w an r-vector; the measurement noise v an m-vector; G, a nonlinear state dependent nxr matrix. The nonlinear vector valued functions f_j and h are, respectively, n and m dimensional. α_j represents a time delay sequence ordered such that

$$\alpha_0 < \alpha_1 < \alpha_2 < \ldots < \alpha_L$$

The noise sequences w and v, are zero-mean white Gaussian with non-negative definite covariance V_w and positive definite V_v, respectively. Also

$$E\{v(k)\ w^T(j)\} = V_{vw}(k)\ \delta_{k,j}$$

for all integers k and j, where V_{vw} is non-negative definite.

The initial states $x(0)$ and $x(-\alpha_j)$, for $j = 1, \ldots, L$ are zero-mean Gaussian random vectors, which are independent of $v(k)$ and $w(k)$, with a positive definite covariance matrix

$$E\{x(-\alpha_j)\ x^T(-\alpha_\ell)\} = V_x(\alpha_j, \alpha_\ell)$$

for $j, \ell = 0, 1, \ldots, L$.

The smoothing problem is to obtain the unbiased smoothed estimate $\hat{x}(k-\ell+1/k+1)$ of the state $x(k-\ell+1)$, where $0 \leq \ell \leq k+1$, conditioned on the set of measurements

$$Y(k+1) = \{y(0), y(1), \ldots, y(k+1)\}$$

such that the following error-variance cost functional is minimized:

$$J(k+1) = \text{Trace}\ [M(k)\ V_{\tilde{x}}(k-\ell+1/k+1)] \tag{4.3}$$

where $M(k)$ is a symmetric positive definite weighting matrix, and $V_{\tilde{x}}(k-\ell+1/k+1)$ is defined by

$$V_{\tilde{x}}(k-\ell+1/k+1) = E_{k+1}[\{x(k-\ell+1) - \hat{x}(k-\ell+1/k+1)\}\{x(k-\ell+1) - \hat{x}(k-\ell+1/k+1)\}^T] \tag{4.4}$$

4.3 THE DERIVATION OF THE SMOOTHER

With reasoning similar to that of Section 3.2, the smoothed estimate is assumed to be constrained by the nonlinear dynamic equation

$$\hat{x}(k-\ell+1/k+1) = \sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\gamma_j/k),\ k-\ell-\gamma_j] + K_{k+1}^{\ell}\ y(k+1) \tag{4.5}$$

where $\sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\gamma_j/k), k-\ell-\gamma_j]$ and K_{k+1}^{ℓ} are yet to be determined.

Since it is required that the smoothed estimate be unbiased, it is necessary that

$$\sum_{j=0}^{L} b_j[\hat{x}(k-\ell-\gamma_j/k), k-\ell-\gamma_j] = \sum_{j=0}^{L} \hat{f}_j[x(k-\ell-\alpha_j), k-\ell-\alpha_j/k] - K_{k+1}^{\ell}\hat{h}[x(k+1), k+1/k] \quad (4.6)$$

where $$\hat{f}_j[x(k-\ell-\alpha_j), k-\ell-\alpha_j/k] = E_k\{f_j[x(k-\ell-\alpha_j), k-\ell-\alpha_j]\} \quad (4.7)$$

and $$\hat{h}[x(k+1), k+1/k] = E_k\{h[x(k+1), k+1]\}$$

With Equation (4.6) substituted into (4.5), the smoothed estimate becomes

$$\hat{x}(k-\ell+1/k+1) = \hat{x}(k-\ell+1/k) + K_{k+1}^{\ell}\{y(k+1) - \hat{h}[x(k+1), k+1/k]\} \quad (4.8)$$

for $k = 0,1,2, \ldots,$ and $0 \leq \ell \leq k+1$.

where $$\hat{x}(k-\ell+1/k) = \sum_{j=0}^{L} \hat{f}_j[x(k-\ell-\alpha_j), k-\ell-\alpha_j/k] \quad (4.9)$$

The error state equation is simply

$$\tilde{x}(k-\ell+1/k+1) = \tilde{x}(k-\ell+1/k) - K_{k+1}^{\ell}\{\tilde{h}[x(k+1)/k] + v(k+1)\} \quad (4.10)$$

where $$\tilde{h}[x(k+1)/k] = h[x(k+1), k+1] - \hat{h}[x(k+1), k+1/k]$$

Then, the error-variance equation is given as

$$V_{\tilde{x}}(k-\ell+1/k+1) = V_{\tilde{x}}(k-\ell+1/k) - K_{k+1}^{\ell}E_k\{\tilde{h}[x(k+1)/k]\,\tilde{x}^T(k-\ell+1/k)\} - E_k\{\tilde{x}(k-\ell+1/k)\,\tilde{h}^T[x(k+1)/k]\}$$

$$\cdot (K_{k+1}^{\ell})^T + K_{k+1}^{\ell}\,\Psi_v(k+1)(k^{\ell}{}_{k+1})^T + K_{k+1}^{\ell}E_k\{\tilde{h}[x(k+1)/k]\,\tilde{h}^T[x(k+1)/k]\}(k^{\ell}{}_{k+1})^T \quad (4.11)$$

The necessary condition for minimizing the trace of $V_{\tilde{x}}(k-\ell+1/k+1)$, can now be obtained from the condition

$$\frac{\partial}{\partial K_{k+1}^{\ell}} \text{Trace}\,[V_{\tilde{x}}(k-\ell+1/k)] = [0]$$

Hence, the gain algorithm K_{k+1}^{ℓ} is given by

$$K_{k+1}^{\ell} = E_k\{\tilde{x}(k-\ell+1/k)\,\tilde{h}^T[x(k+1)/k]\}\,[\Psi_v(k+1) + E_k\{\tilde{h}[x(k+1)/k]\,\tilde{h}^T[x(k+1)/k]\}]^{-1} \quad (4.12)$$

and Equation (4.11) becomes

$$V_{\tilde{x}}(k-\ell+1/k+1) = V_{\tilde{x}}(k-\ell+1/k) - K_{k+1}^{\ell}E_k\{\tilde{h}[x(k+1)/k]\,\tilde{x}^T(k-\ell+1/k)\} \quad (4.13)$$

and $$V_{\tilde{x}}(k-\ell+1, k-m+1/k+1) = V_{\tilde{x}}(k-\ell+1, k-m+1/k) - K_{k+1}^{\ell}E_k\{\tilde{h}[x(k+1)/k]\tilde{x}^T(k-m+1/k)\} \quad (4.14)$$

for $0 \leq \ell, m \leq k+1$.

Now, there remains the problem of evaluating $V_{\tilde{x}}(k-\ell+1/k)$ in Equation (4.13). Even though subtracting Equation (4.9) from Equation (4.1) would easily yield

$$\tilde{x}(k-\ell+1/k) = \sum_{j=0}^{L} \tilde{f}_j[x(k-\ell-\alpha_j)/k] + G[x(k-\ell), k-\ell]\,w(k-\ell) \quad (4.15)$$

where by definition

$$\tilde{f}_j[x(k-\ell-\alpha_j)/k] = f_j[x(k-\ell-\alpha_j), k-\ell-\alpha_j] - \hat{f}_j[x(k-\ell-\alpha_j), k-\ell-\alpha_j/k]$$

it is seen that the error-variance equation cannot be obtained by taking the expectation of Equation (4.15) multiplied by its own transpose, since the expectation

$$E_k\{\sum_{j=0}^{L} \tilde{f}_j[x(k-\ell-\alpha_j)/k]\,w^T(k-\ell)\}$$

can not be explicitly evaluated.

On the other hand, following the derivation presented above, the unbiased estimate of the state $x(k-\ell+1)$ can be obtained as

$$\hat{x}(k-\ell+1/k) = \sum_{j=0}^{L} \hat{f}_j[x(k-\ell-\alpha_j), k-\ell-\alpha_j/k-1] + K_k^{\ell}\{y(k) - \hat{h}[x(k), k/k-1]\} \quad (4.16)$$

which in turn leads to

$$\tilde{x}(k-\ell+1/k) = \sum_{j=0}^{L} \tilde{f}_j[x(k-\ell-\alpha_j),\ k-\ell-\alpha_j/k-1]\ \kappa_k^{\ell}\{\tilde{h}[x(k)/k-1] + v(k)\} + G[x(k-\ell),k-\ell]w(k-\ell) \quad (4.17)$$

The optimal value of the matrix κ_k^{ℓ} can be determined from the necessary condition that

$$\frac{\partial}{\partial \kappa_k^{\ell}} \text{trace } [M(k)\ V_{\tilde{x}}(k-\ell+1/k)] = [0] \quad (4.18)$$

Since $V_{\tilde{x}}(k-\ell+1/k)$ is the expectation of Equation(4.17) multiplied by its own transpose, then using Equation (4.18), the result is

$$\kappa_k^{\ell} = [E_{k-1}\{\sum_{j=0}^{L} \tilde{f}_j[x(k-\ell-\alpha_j)/k-1]\ \tilde{h}^T[x(k)/k-1]\} + G[x(k-\ell),\ k-\ell]\ \Psi_{wv}(k)\ \delta_{k,k-\ell}]$$

$$\cdot\ [\Psi_v(k) + E_{k-1}\{\tilde{h}[x(k)/k-1]\ \tilde{h}^T[x(k)/k-1]\}]^{-1} \quad (4.19)$$

and $V_{\tilde{x}}(k-\ell+1/k)$ is simply

$$V_{\tilde{x}}(k-\ell+1/k) = \sum_{i,j=0}^{L} E_{k-1}\{\tilde{f}_i[x(k-\ell-\alpha_i)/k-1]\ \tilde{f}_j^T[x(k-\ell-\alpha_j)/k-1]\}$$

$$+ G[x(k-\ell),k-\ell]\Psi_w(k-\ell]G^T[x(k-\ell),k-\ell]$$

$$- \kappa_k^{\ell}[\Psi_{vw}(k-\ell)G^T[x(k-\ell),k-\ell]\ \delta_{k,k-\ell}$$

$$+E_{k-1}\{\tilde{h}[x(k)/k-1]\ \sum_{j=0}^{L} \tilde{f}_j^T[x(k-\ell-\alpha_j)/k-1]\}] \quad (4.20)$$

also $V_{\tilde{x}}(k-\ell+1,\ k-m+1/k) = \sum_{i,j=0}^{L} E_{k-1}\{\tilde{f}_i[x(k-\ell-\alpha_i)/k-1]\tilde{f}_j^T[x(k-m-\alpha_j)/k-1]\} + G[x(k-\ell),k-\ell]\Psi_w(k-\ell)$

$$\cdot\ \delta_{k-\ell,k-m}G^T[x(k-m),k-m] - \kappa_k^{\ell}[\Psi_{vw}(k-m)\delta_{k,k-m}G^T[x(k-m),k-m]$$

$$+ E_{k-1}\{\tilde{h}[x(k)/k-1]\ \sum_{j=0}^{L} \tilde{f}_j^T[x(k-m-\alpha_j)/k-1]\}] \quad (4.21)$$

for $0 \le \ell,\ m \le k+1$.

When the discrete time index $k<0$, there is no input to the smoother and therefore $\hat{x}(-\alpha_j/-1)$ is set to zero for $j = 0, 1, \ldots, L$, which results in

$$\tilde{x}(-\alpha_j/-1) = x(-\alpha_j)$$

and

$$V_{\tilde{x}}(-\alpha_j,-\alpha_\ell/-1) = V_x(\alpha_j,\alpha_\ell)$$

for $j, \ell = 0, 1, \ldots, L$.

On the other hand, since the smoothing estimator is unbiased, the expectations that are in Equations (4.12) to (4.14), and (4.19) to (4.21) can be replaced by Equations (3.21) to (3.22) and also

$$E_{k-1}\{\tilde{f}_i[x(k-\ell-\alpha_i)/k-1]\ \tilde{f}_j[x(k-m-\alpha_j)/k-1]\} = E_{k-1}\{f_i[x(k-\ell-\alpha_i),k-\ell-\alpha_i]\ f_j^T[x(k-m-\alpha_j),\ k-m-\alpha_j]\}$$

$$- \hat{f}_i[x(k-\ell-\alpha_i),\ k-\ell-\alpha_i/k-1]\ \hat{f}_j^T[x(k-m-\alpha_j),k-m-\alpha_j/k-1] \quad (4.22)$$

It should be noted that in the case of nonlinear systems the above expectations require infinite dimensional systems to realize.

Therefore, as an approximation, it is assumed that the conditional probability density functions of the smoothing error $\tilde{x}$ are Gaussian. Again, it is important to note that under such an assumption, the algorithms presented in this section can be physically realized without any further approximation for systems with polynomial, product- type or state-dependent sinusoidal nonlinearities.

Also notice that three different types of smoothing and the filtering algorithms all follow immediately from Equations (4.8),(4.12)-(4.14),(4.16) and (4.19)-(4.21) with the following substitutions:

	The Replacement for k+1	The Replacement for ℓ
The filtering estimation	k+1	0
The fixed-lag smoothing	k	N
The fixed-point smoothing	k+1	k-N+1
The fixed-interval smoothing	N	N-k

4.4 THE NONLINEAR FILTERING

When ℓ is set to zero, Equations (4.8),(4.12)-(4.13),(4.16) and (4.19) to (4.20) become the filtering algorithms.

Namely:

$$\hat{x}(k+1/k+1) = \hat{x}(k+1/k) + K^0_{k+1}\{y(k+1) - \hat{h}[x(k+1),k+1/k]\}$$

$$\hat{x}(k+1/k) = \sum_{j=o}^{L} \hat{f}_j[x(k-a_j)),k-a_j/k-1] + K^0_k\{y(k)-\hat{h}[x(k),k/k-1]\}$$

$$K^0_{k+1} = E_k\{\tilde{x}(k+1/k)\ \tilde{h}^T[x(k+1)/k]\}\ [V_v(k+1) + E_k\{\tilde{h}[x(k+1)/k]\ \tilde{h}^T[x(k+1)/k]\}]^{-1}$$

$$V_{\tilde{x}}(k+1/k+1) = V_{\tilde{x}}(k+1/k) - K^0_{k+1}E_k\{\tilde{h}[x(k+1/k)]\tilde{x}^T(k+1)\}$$

$$V_{\tilde{x}}(k+1/k) = \sum_{i,j=o}^{L} E_{k-1}\{\tilde{f}_i[x(k-a_i)/k-1]\ \tilde{f}_j^T[x(k-a_j)/k-1]\} + G[x(k),k]\ V_w(k)\ G^T[x(k),k]$$

$$-K^0_k[V_{vw}(k)\ G^T[x(k),k] + E_{k-1}\{\tilde{h}[x(k)/k-1]\ \sum_{j=o}^{L} \tilde{f}_j^T\ [x(k-a_j)/k-1]\}]$$

and finally

$$K^0_k = [E_{k-1}\{\sum_{j=0}^{L} \tilde{f}_j[x(k-a_j)/k-1]\ \tilde{h}^T[x(k)/k-1]\}\ G[x(k),k]$$

$$\cdot\ V_{wv}(k)][V_v(k) + E_{k-1}\{\tilde{h}[x(k)/k-1]\ \tilde{h}^T[x(k)/k-1]\}]^{-1}$$

4.5 ESTIMATION IN LINEAR DISCRETE SYSTEMS

In this section, general linear estimation algorithms are obtained for discrete delayed systems corrupted by correlated noise processes. The algorithms can be easily converted to yield the fixed-lag, fixed point, fixed-interval smoothing and the filtering estimators together with their respective error-variance equations. This is done by making the proper choice of discrete indices.

For linear discrete systems, we have

$$\sum_{j=o} f_j[x(k-a_j),k-a_j] = \sum_{j=o}^{L} F_j(k)\ x\ (k-a_j)$$

$$h[x(k),k] = H(k)\ x(k)$$

and

$$G[x(k),k] = G(k)$$

Then the general linear estimation algorithms are as follows:

$$\hat{x}(k-\ell+1/k+1) = \hat{x}(k-\ell+1/k) + K^{\ell}_{k+1}\ \{y(k+1)$$

$$- H(k+1)\ \hat{x}(k+1)/k)\}$$

$$\hat{x}(k-\ell+1/k) = \sum_{j=o}^{L} F_j(k-\ell)\ \hat{x}(k-\ell-a_j/k-1) + K^{\ell}_k\{y(k) - H(k)\hat{x}(k/k-1)\}$$

$$K^{\ell}_{k+1} = V_{\tilde{x}}(k-\ell+1,k+1/k)\ H^T(k+1)\ [V_v(k+1) + H^T(k+1)\ V_{\tilde{x}}\ (k+1/k)\ H^T(k+1)]^{-1}$$

$$V_{\tilde{x}}(k-\ell+1/k+1) = V_{\tilde{x}}(k-\ell+1/k) - K^{\ell}_{k+1}\ H(k+1)\ V_{\tilde{x}}(k+1,k-\ell+1/k)$$

$$V_{\tilde{x}}(k-\ell+1/k) = \sum_{i,j=0}^{k} F_i(k-\ell)V_{\tilde{x}}(k-\ell-a_i,k-\ell-a_j/k-1)\ F_j^T(k-\ell) + G(k-\ell)\ V_w(k-\ell)\ G^T(k-\ell)$$

$$-\kappa_k^{\ell}[\Psi_{vw}(k-\ell)G^T(k-\ell) + H(k)\sum_{j=o}^{L} \underset{\sim}{V}_{\underset{\sim}{x}}(k,k-\ell-\alpha_j/k-1)\ F_j^T(k-\alpha_j)]$$

$$\kappa_k^{\ell} = [\sum_{j=o}^{L} F_j(k-\ell)\underset{\sim}{V}_{\underset{\sim}{x}}(k-\ell-\alpha_j,k/k-1)\ H^T(k)+G(k-\ell)\ \Psi_{wv}(k)\delta_{k,k-\ell}]$$

$$\cdot\ [\Psi_v(k) + H(k)\ \underset{\sim}{V}_{\underset{\sim}{x}}(k\ /k-1)H^T(k)]^{-1}$$

and
$$\underset{\sim}{V}_{\underset{\sim}{x}}(k-\ell+1,\ k-m+1/k) = \sum_{i,j=o}^{L} F_i(k-\ell)\underset{\sim}{V}_{\underset{\sim}{x}}(k-\ell-\alpha_i,k-m-\alpha_j/k-1)\ F_j^T(k-m) + G(k-\ell)\Psi_w(k-\ell)\delta_{k-\ell,k-m}G^T(k-m)$$

$$- \kappa_k^{\ell}[\Psi_{vw}(k-m)\delta_{k,k-m}G^T(k-m) + \sum_{j=o}^{L} H(k)\ \underset{\sim}{V}_{\underset{\sim}{x}}(k,k-m-\alpha_j/k-1)\ F_j^T(k-m)]$$

It can be easily shown that in the special case of discrete systems without delay, the results obtained for the filtering algorithm agree well with those in the literature [36]. The results presented in this section can also be extended to continuous time smoothing problems and nonlinear distributed parameter systems with or without delays [52]. The same basic approach can also be applied to derive estimation algorithms for nonlinear systems corrupted by colored noise [51].

SECTION 5

CONCLUSIONS

A unified approach is presented to derive estimation algorithms for discrete and continuous nonlinear systems with and without delays, corrupted by white Gaussian noise, correlated noise, colored noise as well as noise-free processes.

In Section 2, nonlinear filtering algorithms were derived for continuous nonlinear systems without delays corrupted by white noise, correlated noise and noise-free processes. The main technique involves the application of the matrix minimum principle together with the Kolmogorov and Kushner equations to minimize the error-variance, taken to be the estimation criterion. The filtering equations obtained for nonlinear systems with white noise processes are exact, but for non-white noise processes the results obtained are approximate.

For systems with polynomial, product-type or state-dependent sinusoidal nonlinearities, the proposed algorithms can be evaluated without the need for approximation under the assumption that the estimator errors are Gaussian. Such an assumption is significantly different from the most commonly used assumption that the state is Gaussian.

Nonlinear estimation algorithms for discrete nonlinear delayed systems with measurements corrupted by white noise and correlated noise processes are respectively, derived in Sections 3 and 4. The estimation algorithms directly yield the fixed-lag, fixed-point and fixed-interval smoothing and filtering algorithms, with proper substitution of the discrete time indices. The proposed technique makes use of the concept of the gradient matrix, the minimum principle to derive the optimal values of the coefficients in the estimation algorithms under the requirements that the estimates be unbiased and the error-variances minimized. The derivation is straight-forward and clearly indicates the close link between three different classifications of smoothers and the filtering estimator.

The results obtained are exact and optimal with respect to the imposed constraints on the dynamic equations of the estimators, minimizing the error-variances. The algorithms derived can be fully implemented in a digital computer under the assumption that the probability density functions of the estimator errors are Gaussian. For systems with polynomial, product-types and state-dependent sinusoidal nonlinearities, no further approximation is needed to evaluate the expectations involved in the algorithms. They are expected to be computationally efficient, since no augmentation of state variables is involved.

The results can also be applied to various special cases of nonlinear as well as linear estimation problems; for example: The estimation problems for linear or nonlinear systems without delays and the linear estimation problems of time-delayed systems corrupted by white noise as well as non-white noise processes, etc.

For linear estimation problems the results obtained are identified with published results in the literature. In the particular case of linear time-delayed systems corrupted by additive white noise, the results are stable, but the stability behaviour of the nonlinear estimators is yet to be investigated.

The results obtained in Sections 3 and 4 can be further extended to continuous time systems and nonlinear distributed parameter systems.

ACKNOWLEDGEMENTS

This work was carried out with the support of the Defence Research Establishment Ottawa (DREO), Canada. The author wishes to thank Mr. C.R. Iverson, Chief, DREO and Mr. K.A. Peebles, Head, Electromagnetics Section, for their encouragement. Thanks are also due to Mrs. D.M. Findlay and Mrs. F.L. Stiles for their amazing patience in typing the manuscript.

REFERENCES

1. R.E. Kalman, "A new approach to linear filtering and prediction problems, "Trans. ASME, J. Basic Engrg., vol. 82D, March 1960, pp. 34-45.

2. R.L. Stratonovich, "Conditional Markov process theory," Theory Prob. Appl. (U.S.S.R), vol. 5, 1960, pp. 156-178.

3. H.J. Kushner, "On differential equations satisfied by conditional probability densities of Markov processess," SIAM J. Control, vol. 2, 1964, pp. 106-119.

4. W.M. Wonham, "Some applications of stochastic differential equations to optimal nonlinear filtering," SIAM J. Control, vol. 2, 1965, pp. 347-369.

5. R.S. Bucy, "Nonlinear filtering," IEEE Trans. Aut. Control, vol. AC-10, April 1965, p. 198.

6. H.J. Kushner, "Dynamical equations for optimum nonlinear filtering," J. Differential Equations, vol. 3, 1967, pp. 179-190.

7. H.W. Sorenson and D.L. Alspach, "Recursive Bayesian estimation using Gaussian sums," IFAC J. Automatica, vol. 7, 1971, pp. 465-479.

8. R.S. Bucy, "Bayes theorem and digital realization for nonlinear filters," J. Astro. Sciences, vol. 17, 1969, pp. 80-94.

9. J.L. Center, Practical Nonlinear Filtering Based on Generalized Least Squares Approximation of the Conditional Probability Distribution, Ph. D. Dissertation, Washington University, St. Louis, 1972.

10. H. Cox, "On the estimation of state variables and parameters for noisy dynamic systems," IEEE Trans. Aut. Control, AC-9, Feb. 1964, pp. 5-12.

11. H.J. Kushner, "Approximations to optimal nonlinear filters", IEEE Trans. Aut. Control, vol. AC-12, Oct. 1967, pp. 546-556.

12. D.M. Detchmendy and R. Sridhar, "Sequential estimation of states and parameters in noisy nonlinear dynamical systems," Trans. ASME. J. Basic Engrg., vol. 88D, June 1966, pp. 362-368.

13. R.W. Bass, V.D. Norum and L. Schwartz, "Optimal multi-channel nonlinear filtering," J. Math. Anal. Appl., vol. 16, 1966, pp. 152-164.

14. A.P. Sage and W.S. Ewing, "On filtering and smoothing algorithms for nonlinear state estimation," Int. J. Control, vol. 11, no. 1, 1970, pp. 1-18.

15. Y. Sunahara, "An approximate method of state estimation for nonlinear dynamical systems," Trans. ASME, J. Basic Engrg., vol. 92D, June 1970, pp. 385-393.

16. L. Schwartz and E.B. Stear, "A computational comparison of several nonlinear filters," IEEE Trans. Aut. Control, vol. 13, Feb. 1968, pp. 83-86.

17. C.T. Leondes, J.B. Peller and E.B. Stear, "Nonlinear smoothing theory," IEEE Trans. System Sci. Cyb., vol. SSC-6, January 1970, pp. 63-71.

18. T. Kailath and P. Frost, "An innovations approach to least-squares estimation - Part II: Linear smoothing in additive white noise,"IEEE Trans. Aut. Control, vol. AC-13, Dec. 1968, pp. 655-660.

19. D.G. Lainiotis, "Optimal nonlinear estimation," Int. J. Control, vol. 14, no. 6, 1971, pp. 1137-1148.

20. C.N. Lee, "Nonlinear interpolation," IEEE Trans. Inf. Theory, vol. IT-17, January 1971, pp. 45-49.

21. H. Kuckernask, "Optimal filtering in linear systems with time delays," IEEE Trans. Aut. Control, vol. AC-12, April 1967, pp. 169-173.

22. R. Priemer and A.G. Vacroux, "Estimation in linear discrete systems with multiple time delays," IEEE Trans. Aut. Control, vol. AC-14, August 1969, pp. 384-387.

23. R. Priemer and A.G. Vacroux, "Smoothing in linear discrete systems with time delays," Int. J. Control, vol. 13, no. 2, 1971, pp. 299-303.

24. M. Farooq and A.K. Mahalanabis, "A note on the maximum likelihood state estimation of linear discrete systems with multiple time delays," IEEE Trans. Aut. Control, vol. AC-16, Feb. 1971, pp. 104-105.

25. K.K. Biswas and A.K. Mahalanabis, "Optimal smoothing for continuous-time systems with multiple time delays," IEEE Trans. Aut. Control, vol. AC-17, August 1972, pp. 572-574.

26. M. Farooq and R. Balasubramanian, "Fixed-interval smoothing of linear discrete systems with multiple delays," IEEE Trans. Aut. Control, vol. AC-21, April 1976, pp. 273-275.

27. T.K. Yu, J.H. Seinfeld, and W.H. Ray," Filtering in nonlinear time delay systems," IEEE Trans. Aut. Control, vol. AC-19, August 1974, pp. 324-333.

28. D.F. Liang, Nonlinear Estimation Systems with and without Delays, Ph.D. Dissertation, University of Alberta, Canada, 1974.

29. M. Athan, "The matrix minimum principle," J. Inf. Control, vol. 11, 1968, pp. 592-606.

30. D.F. Liang and G.S. Christensen," Exact and approximate state estimation for nonlinear dynamic systems," IFAC J. Automatica, vol. 11, 1975, pp. 603-612.

31. A.N. Kolmogorov, "Interpolation and extrapolation of stationary random sequence," Bull. Acad. Sci. U.S.S.R., Math. Ser., vol. 5, 1941.

32. D.F. Liang, "Comparisons of nonlinear filters for systems with non-negligible nonlinearities," NATO Agardograph 256, 1981.

33. D.F. Liang and G.S. Christensen," New estimation algorithms for discrete nonlinear systems and observations with multiple time delays," Int. J. Control, vol. 23, no. 5, 1976, pp. 613-625.

34. D.F. Liang and G.S. Christensen," Estimation for discrete nonlinear time-delayed systems and measurements with correlated and coloured noise processes," Int. J. Control, vol. 28, no.1, 1978, pp. 1-10.

35. A.E. Bryson and D.E. Johansen, "Linear filtering for time-varying systems using measurements containing colored noise," IEEE Trans. Aut. Control, vol. AC-10, Feb. 1965, pp. 4-10.

36. A.P. Sage and J.L. Melsa, Estimation Theory with Applications to Communications and Control. New York: McGraw-Hill, 1971.

37. K. Ito, "On Stochastic processes," Lecture Notes, Tata Inst. for Fundamental Research, Bombay, 1961.

38. F.C. Schweppe, Uncertain Dynamic Systems, New Jersey: Prentice Hall, 1973.

39. E. Parzen, Stochastic Processes, San Francisco: Holden-Day, Inc., 1962.

40. A. Gelb and W.E. Vander Velde, Multiple-Input Describing Functions and Nonlinear System Design, New York: McGraw Hill Co., 1968.

41. A.G. Carlton, "Linear estimation in stochastic processes," John Hopkins University, Appl. Phys. Lab Baltimore, Md., Internal Rept., 1962.

42. H.E. Rauch, "Solutions to the linear smoothing problem," IEEE Trans. Aut. Control, vol. AC-8, pp. 371-372, Oct. 1963.

43. A.E. Bryson and M. Frazier, "Smoothing for linear and nonlinear dynamic systems," Proc. Optimum System Synthesis Conf., U.S. Air Force Tech. Report ASD-TDR-063-119, Feb. 1963.

44. C.N. Kelly and B.D.O. Anderson, "On the stability of fixed lag smoothing algorithms," J. Franklin Inst., vol. 291, 1971, pp. 271-281.

45. J.S. Meditch, "On optimal linear smoothing theory," J. Inf. Control, vol. 10, 1967, pp. 598-615.

46. D.F. Liang and G.S. Christensen, "New filtering and smoothing algorithms for discrete nonlinear systems with time delays," Int. J. Control, vol. 23, no. 5, 1976, pp. 613-625.

47. K.K. Biswas and A.K. Mahalanabis, "An approach to fixed-point smoothing problems," IEEE Trans. Aerospace and Elect. Systems, vol. AES-8, Sept. 1972, pp. 676-680.

48. R.E. Kalman, "New methods in Wiener Filtering Theory," Proc. 1st Symp. on Engrg. Appl. of Random Function Theory and Probability, New York: Wiley, 1963.

49. B.V. Raja Rao and A.K. Mahalanabis, "Estimation in linear delayed discrete-time systems with correlated state and measurement noises," IEEE Trans. Aut. Control, vol. 16, June 1971, p. 267.

50. D.F. Liang and G.S. Christensen, "Comments on estimation in linear delayed discrete-time systems with correlated state and measurement noises," IEEE Trans. Aut. Control, vol. AC-20, 1975, pp. 176-177.

51. D.F. Liang and G.S. Christensen," Estimation for discrete nonlinear time-delayed systems and measurements with correlated and coloured noise processes," Int. J. Control, vol. 28, no. 1, 1978, pp. 1-10.

52. D.F. Liang, "State estimation for nonlinear distributed-parameter systems involving multiple delays," IEEE Trans. Aut. Control, vol. AC-23, 1978, pp. 502-504.

THE THEORY AND TECHNIQUES OF DISCRETE-TIME DECENTRALIZED FILTERS

THOMAS H. KERR
Intermetrics, Inc.
733 Concord Avenue
Cambridge, MA 02138

LEONARD CHIN
U.S. Naval Air
Development Center
Warminster, PA 18974

SUMMARY

The purpose of this chapter is multifold. One purpose is to provide an overview survey of the alternative decentralized filtering techniques that have evolved over the last decade and to indicate the current status of each approach. This aspect is important as a preliminary step in performing engineering by allowing the selection of the approach that best fits the constraints imposed by the specific application. Several contributions that are provided herein advance the state-of-the-art for two decentralized filtering approaches (viz., SLU and SPA) as formulated here in discrete-time by specifying and summarizing mechanization equations (with rationale), by analytically establishing stability of these estimation algorithms, and by providing tables that allow quantification of the computer burden upon implementation in terms of required memory allotment and algorithm cycle times. Thus a complete view of these two approaches to decentralized filtering is provided here. Current applications and likely future application areas for decentralized filtering are identified. A primary consideration was the proper pedagogical approach to simply explain somewhat obtuse prior material to make it easily accessible to many levels of readers (with a variety of backgrounds and primary interests) to demonstrate that decentralized filtering does in fact have a firm theoretical foundation.

1 PRELIMINARIES

1.1 Outline of the Chapter

The appropriate discrete-time models to be used for decentralized filtering are presented in Section 1.2 with certain drawbacks, that apparently were previously ignored, being identified and resolved. A brief overview survey of alternative decentralized filtering formulations, their current status, salient features and advantages/disadvantages is provided in Section 1.3. A particular drawback, encountered for several decentralized filtering approaches, of requiring the objective application to be reexpressed in an "output decentralized" form is discussed in Section 1.4. A recent approach for rigorously handling several filters (with nested state-variable system models) operating at differing measurement utilization rates for possible parallel processing implementation is described in Section 1.5.

Two particularly well-developed and appealing approaches to decentralized filtering are the Surely Locally Unbiased (SLU) filter and the Sequentially Partitioned Algorithm (SPA). The SLU approach is completely described in Section 2 and the SPA is described in Section 3. An analytic proof of the stability of both of these filters is offered in Section 4 as a significant technical contribution of this investigation. Present and anticipated future applications of decentralized filters are indicated in Section 5.

1.2 Appropriate Discrete-Time Models for Decentralized Filtering

Consider the following collection $\{S_i,\ i=1,2,\ldots,N\}$ of N interconnected dynamical subsystems (as in Refs. 8-14):

$$S_i:\ \underset{(n_i\times 1)}{\dot{\underline{x}}_i(t)} = F_i(t)\underline{x}_i(t)+L_{ii}(t)\underline{u}_i(t)+\underline{w}_i(t) \tag{1.2-1}$$

having discrete measurements available to the i^{th} subsystem S_i of the form:

$$\underset{(q_i\times 1)}{\underline{z}_i(t_k)} = H_i(t_k)\underline{x}(t_k)+\underline{v}_i(t_k) \tag{1.2-2a}$$

$$= [H_i(t_k)\ \vdots\ \hat{H}_i(t_k)]\begin{bmatrix} P_{\underline{x}\hat{x}_i} \\ \hline L_i(t_k)\end{bmatrix}\underline{x}(t_k)+\underline{v}_i(t_k) \tag{1.2-2b}$$

$$= [H_i(t_k)\underline{x}_i(t_k)+\hat{H}_i(t_k)\underline{u}_i(t_k)]+\underline{v}_i(t_k) \tag{1.2-2c}$$

$$= H_i(t_k)\underline{x}_i(t_k)+\hat{H}_i(t_k)L_i(t_k)\underline{x}(t_k)+\underline{v}_i(t_k) \tag{1.2-2d}$$

where $P_{x,i}$ is the projection operator from R^n

$$n = \sum_{i=1}^{N} n_i \tag{1.2-3}$$

to R^{n_i} and where the vector-valued interaction input is represented (by using the historically standard notation of weighting matrices L_{ij} popularized in Ref. 21 on p. 122) by

$$\underset{u_i(t)}{(p_i x 1)} \triangleq \underset{L_i(t)}{(p_i x n)} \underset{x(t)}{} = \left[\underset{L_{i1}(t)}{(p_i x n_1)} \;\middle|\; \underset{L_{i2}(t)}{(p_i x n_2)} \;\middle|\; \cdots \;\middle|\; \underset{L_{i,i-1}(t)}{(p_i x n_{i-1})} \;\middle|\; \underset{0}{} \;\middle|\; \underset{L_{i,i+1}(t)}{(p_i x n_{i+1})} \;\middle|\; \cdots \right] x(t) \tag{1.2-4a}$$

$$= \sum_{\substack{j=1 \\ j \neq i}}^{N} \underset{L_{ij}(t)}{(p_i x n_j)} \; \underset{x_j(t)}{(n_j x 1)} \tag{1.2-4b}$$

Notice that $u_i(t)$ has no direct component of $x_i(t)$.

The process and measurement noises $w_i(t)$ and $v_i(t)$ are assumed (as in Refs. 8-14) to be independent, zero mean, white Gaussian noises having associated covariance matrices $Q'_i(t)$ and $R_i(t)$, respectively, and uncorrelated with the Gaussian initial condition

$$x_i(0) \rightsquigarrow N(\underline{0}, \dot{P}_i) \tag{1.2-5}$$

Also $w_i(t)$ and $v_i(t)$ are assumed to be uncorrelated with the noises and initial conditions of other subsystems [viz., $w_j(t)$, $v_j(t)$, and $x_j(0)$ for $j \neq i$].

The behavior of an entire interconnected linear system is summarized at each time instant t by the n-dimensional full state vector

$$\underline{x}(t) \triangleq \left[x_1^T(t),\ x_2^T(t), \ldots, x_N^T(t) \right]^T \tag{1.2-6}$$

of which each user subsystem S_i has an n_i-vector valued subset $x_i(t)$. The projection operator introduced in Eq. 1.2-2b operates on $\underline{x}(t)$ to produce

$$P_{x,i}[\underline{x}(t)] = x_i(t) \tag{1.2-7}$$

and so can be represented as a matrix premultiplying $\underline{x}(t)$ in Eq. 1.2-7 of the following form

$$\underset{P_{x,i}}{(n_i x n)} = \left[\underset{0}{(n_i x n_1)} \;\middle|\; \underset{0}{(n_i x n_2)} \;\middle|\; \cdots \;\middle|\; \underset{0}{(n_i x n_{i-1})} \;\middle|\; I_{n_i} \;\middle|\; \underset{0}{(n_i x n_{i+1})} \;\middle|\; \cdots \;\middle|\; \underset{0}{(n_i x n_N)} \right] \tag{1.2-8}$$

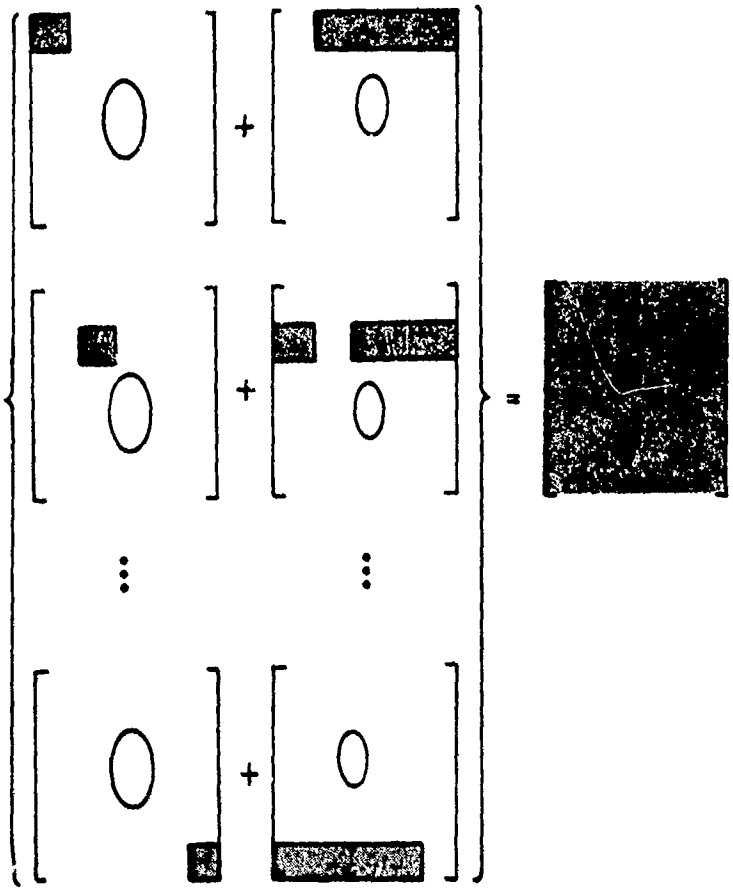
The differential equation that describes the time evolution of the aggregate state of Eq. 1.2-6 in continuous-time is

$$\dot{\underline{x}}(t) = \bar{F}(t)\underline{x}(t) + \underline{w}'(t) \tag{1.2-9}$$

where

$$\underline{w}'(t) = \left[w_1'^T(t),\ w_2'^T(t), \ldots, w_N'^T(t) \right]^T \tag{1.2-10}$$

and

$$F(t) \triangleq$$  (1.2-11a)

$$= \sum_{i=1}^{N} P_{x,i}^{T} \; F_i(t) \; P_{x,i} + \sum_{i=1}^{N} P_{x}^{T},i \; L_{ii}(t) \; L_i(t) \tag{1.2-11b}$$

$$= \sum_{i=1}^{N} P_{x,i}^{T} \left[F_i(t) \; P_{x,i} + L_{ii}(t) L_i(t) \right] \tag{1.2-11c}$$

The appropriate transition matrix for Eq. 1.2-9 is $\Phi(t,\tau)$ which satisfies

$$\frac{\partial}{\partial t} \Phi(t,\tau) = F(t) \; \Phi(t,\tau) \tag{1.2-12}$$

and

$$\Phi(\tau,\tau) = I_n \text{ for all } \tau \tag{1.2-13}$$

The exact solution of Eq. 1.2-9 is of the form

$$\underline{x}(t) = \Phi(t,\tau)\underline{x}(\tau) + \int_{\tau}^{t} \Phi(t,s) \; \underline{w}(s) ds \tag{1.2-14}$$

and the corresponding exact discrete-time representation is (p. 171 of Ref. 19):

$$\underline{x}(k+1) = \phi(k+1,k)\underline{x}(k) + \underline{w}(k) \tag{1.2-15}$$

where $t = (k+1)\Delta$, $\tau = k\Delta$ while the time step Δ has been suppressed and the exact equivalent to continuous white noise $\underline{w}'(t_k)$ is denoted by $\underline{w}(k)$ to be an n-vector-valued Gaussian white discrete-time process having the following statistics:

$$E\left[\underline{w}(k)\right] = \int_{k\Delta}^{(k+1)\Delta} \Phi((k+1)\Delta, s) \; E\left[\underline{w}'(s)\right] ds = \underline{0} \tag{1.2-16}$$

$$Q(k) \triangleq E\left[\underline{w}(k)\underline{w}^{T}(k)\right] = \int_{k\Delta}^{(k+1)\Delta} \Phi((k+1)\Delta, s) Q'(s) \Phi^{T}((k+1)\Delta, s) ds \tag{1.2-17}$$

$$E\left[\underline{w}(k)w^{T}(j)\right] = 0 \text{ for } k \neq j \tag{1.2-18}$$

The exact difference equation that describes the time evolution of the i^{th} subsystem

is

$$x_i(k+1) = P_{x,i}\,\underline{x}(k+1) \tag{1.2-19}$$

hence by substituting Eq. 1.2-15 into Eq. 1.2-19 yields the following result:

$$x_i(k+1) = P_{x,i}\,\bar{\Phi}(k+1,k)\underline{x}(k) + P_{x,i}\,\underline{w}(k) \tag{1.2-20a}$$

$$= \sum_{j=1}^{N} \bar{\phi}_{ij}(k+1,k)x_j(k) + w_i(k) \tag{1.2-20b}$$

$$= \bar{\phi}_{ii}(k+1,k)\,x_i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N} \bar{\phi}_{ij}(k+1,k)x_j(k) + w_i(k) \tag{1.2-20c}$$

Equivalent to Eq. 1.2-17 (p. 171 of Ref. 19), Q(k) evolves in time according to the following differential equation

$$\dot{\bar{Q}}(t,t_k) = \bar{F}(t)Q(t,t_k) + Q(t,t_k)\bar{F}^T(t) + \bar{Q}'(t) \tag{1.2-21}$$

where

$$\bar{Q}'(t) = \mathrm{diag}(Q'_1(t), Q'_2(t), \ldots, Q'_N(t)) \tag{1.2-22}$$

Taking a different tack, the exact discrete-time representation of the i^{th} local subsystem's solution of Eq. 1.2-1 is

$$x_i(k+1) = \phi_{ii}(k+1,k)x_i(k) + \int_{k\Delta}^{(k+1)\Delta} \phi_{ii}((k+1)\Delta,s)L_{ii}(s)u_i(s)ds + \int_{k\Delta}^{(k+1)\Delta} \phi_{ii}((k+1)\Delta,s)w'_i(s)ds \tag{1.2-23}$$

Applying the usual simplifying assumption (as in Eq. 4-124 of Ref. 19), that the inputs are constant over the interval of integration, yields the following subsystem evolution equation

$$x_i(k+1) = \phi_{ii}(k+1,k)x_i(k) + \dot{L}_{ii}(k)u_i(k) + \dot{w}_i(k) \tag{1.2-24}$$

where

$$\dot{L}_{ii}(k) \triangleq \int_{k\Delta}^{(k+1)\Delta} \phi_{ii}((k+1)\Delta,s)L_{ii}(s)ds \tag{1.2-25}$$

and

$$\dot{w}_i(k) \triangleq \int_{k\Delta}^{(k+1)\Delta} \phi_{ii}((k+1)\Delta,s)w'_i(s)\,ds \tag{1.2-26}$$

with $\dot{w}_i(k)$ having the following statistics:

$$E\left[\dot{w}_i(k)\right] = \underline{0} \tag{1.2-27}$$

$$\dot{Q}(k) \triangleq E\left[\dot{w}_i(k)\dot{w}_i^T(k)\right] = \int_{k\Delta}^{(k+1)\Delta} \phi_{ii}((k+1)\Delta,s)\,Q'_i(s)\,\phi_{ii}^T((k+1)\Delta,s)ds \tag{1.2-28}$$

$$E\left[\dot{w}_i(k)\dot{w}_i^T(j)\right] = \underline{0} \text{ for } j\neq k \tag{1.2-29}$$

$$E\left[\dot{w}_i(k)\dot{w}_i^T(j)\right] = \int_{k\Delta}^{(k+1)\Delta}\int_{j\Delta}^{(j+1)\Delta} \phi_{ii}((k+1)\Delta,s)\,E\left[w_i(s)w_i^T(\tau)\right]\phi_{ii}^T((j+1)\Delta,\tau)\,d\tau\,ds = \underline{0} \quad \text{for } i\neq i \tag{1.2-30}$$

Notice that, in general, the global solution of Eq. 1.2-15, as projected to the i^{th} subsystem as in Eq. 1.2-20c (although similar in form), can be quite different from the i^{th} local subsystem's solution of Eq. 1.2-24 unless both of the following conditions hold

$$\phi_{ii}(k+1,k) = \bar{\phi}_{ii}(k+1,k) \tag{1.2-31}$$

and

$$\int_{k\Delta}^{(k+1)\Delta} \Phi_{ii}((k+1)\Delta,s)L_{ii}(s)ds\ u_i(k) = \sum_{\substack{j=1\\ j\neq i}}^{N} \bar{\Phi}_{ij}(k+1,k)x_j(k) \qquad (1.2\text{-}32a)$$

A condition equivalent to Eq. 1.2-32a (via Eq. 1.2-4b) is

$$\left[\int_{k\Delta}^{(k+1)\Delta} \Phi_{ii}((k+1)\Delta,s)L_{ii}(s)ds\right]\sum_{\substack{j=1\\ j\neq i}}^{N} L_{ij}(k)x_j(k) = \sum_{\substack{j=1\\ j\neq i}}^{N} \bar{\Phi}_{ij}(k+1,k)x_j(k) \qquad (1.2\text{-}32b)$$

A <u>sufficient</u> condition for Eq. 1.2-32b to be satisfied is for the following to hold:

$$\left[\int_{k\Delta}^{(k+1)\Delta} \Phi_{ii}((k+1)\Delta,s)L_{ii}(s)ds\right]L_{ij}(k) = \hat{L}_{ii}(k)L_{ij}(k) = \hat{\Phi}_{ij}(k+1,k) \qquad (1.2\text{-}33)$$

for $j = 1,2,\ldots,N/\{i\}$.

In general, even the milder condition of Eq. 1.2-32 is <u>not</u> satisfied since $\Phi_{\ell\ell}(k+1,k)$ satisfies

$$\frac{\partial}{\partial t}\Phi_{\ell\ell}(t,\tau) = F_\ell(t)\,\Phi_{\ell\ell}(t,\tau) \qquad (1.2\text{-}34)$$

with boundary condition

$$\Phi_{\ell\ell}(\tau,\tau) = I_{n_\ell} \qquad (1.2\text{-}35)$$

while $\bar{\Phi}_{\ell\ell}(k+1,k)$ satisfies

$$\frac{\partial}{\partial t}\bar{\Phi}_{\ell\ell}(t,\tau) = P_{x,\ell}\frac{\partial}{\partial t}\bar{\Phi}(t,\tau)\,P_{x,\ell}^T \qquad (1.2\text{-}36a)$$

$$= P_{x,\ell}\,\bar{F}(t)\,\bar{\Phi}(t,\tau)\,P_{x,\ell}^T \qquad (1.2\text{-}36b)$$

$$= P_{x,\ell}\left[\sum_{j=1}^{N} P_{x,i}^T(F_i(t)P_{x,i} + L_{ii}(t)L_{ij}(t))\right]\bar{\Phi}(t,\tau)P_{x,\ell}^T \qquad (1.2\text{-}36c)$$

$$= F_\ell(t)\,\bar{\Phi}_{\ell\ell}(t,\tau) + \sum_{\substack{j=1\\ j\neq i}}^{N} L_{\ell\ell}(t)L_{\ell j}(t)\,\bar{\Phi}_{j\ell}(t,\tau) \qquad (1.2\text{-}36d)$$

with boundary conditions

$$\Phi_{\ell\ell}(\tau,\tau) = I_{n_\ell} \qquad (1.2\text{-}37)$$

$$\bar{\Phi}_{\ell j}(\tau,\tau) = \underline{0} \quad \text{for } j = 1,2,\ldots,N/(\ell) \qquad (1.2\text{-}38)$$

(Notation for the index j in Eq. 1.2-38 indicates all values from 1 to N are taken on except for the value currently held by ℓ.) Obviously, the solution of Eqs. 1.2-34 and -35 coincides with the solution of Eqs. 1.2-36d, -37, and -38 (as required for the condition of Eq. 1.2-31 to be satisfied) for the special case when

$$L_{\ell\ell}(t) = 0 \text{ for } i = 1,2,\ldots,N \qquad (1.2\text{-}39)$$

since then the aggregate of the local state-variable solutions provided by each subsystem is identical to what would be obtained by solving the aggregated n^{th} order system of essentially decoupled subsystems with the only allowable coupling occurring in the measurements. (In Refs. 15 and 35, it is demonstrated that for the application of decentralized estimation to the interconnected structure of the JTIDS RelNav net, the fairly stringent structural constraint of Eq. 1.2-39 is in fact satisfied.)

The issue of specifying conditions for exact correspondence for decentralized estimation apparently has not been considered in Refs. 8-14, 21, 26, 30, and 31, but were raised within this investigation for completeness. However, Ref. 54 does address the problem somewhat by advocating the use of "splitting methods" to obtain a solution to the problem of large-scale <u>linear</u> least squares estimation (in a Hilbert space) of a non-dynamic system. The approach of Ref. 54 utilizes two-level hierarchical coordination (by communication of a supremal coordinator's decisions) in an iterative fashion to several lower level decentralized local estimators. In this manner, the proper solution to the aggregate <u>static</u> problem is to be converged upon asymptotically. Ref. 54 also provides

suggested extensions for the use of splitting methods and a coordinating supremal controller in implementing the classical optimal regulation control of a time-varying linear dynamic system with a standard quadratic cost function to be minimized. Perhaps duality can be fruitfully exploited to obtain useful generalizations to optimal estimation of these techniques for optimal decentralized regulation.

Tentatively, the discrete-time model for decentralized estimation to be used here can be recapitulated from Eqs. 1.2-24 and 1.2-2d as

$$S_i: \; x_i(k+1) = \Phi_{ii}(k+1,k)\,x_i(k) + \overset{\bullet}{L}_{ii}(k)\,u_i(k) + \overset{\bullet}{w}_i(k) \tag{1.2-40}$$

with local measurements available to the i^{th} subsystem S_i of the form

$$z(t_k) = \left[\bar{H}_i(t_k)P_{x,i} + \hat{H}_i(t_k)L_i(t_k)\right]\underline{x}(t_k) + v(t_k) = H_i(t_k)\underline{x}(t_k) + v_i(t_k) \tag{1.2-41}$$

where $P_{x,i}$ and $L_i(t_k)$ are defined in Eqs. 1.2-8 and 1.2-4, respectively, and the statistics of the process noise $\overset{\circ}{w}_i(\cdot)$ are provided in Eqs. 1.2-27 to 1.2-30.

1.3 Brief Survey and Status of Alternate Approaches to Decentralized Filtering

The implicit motivation underlying all hierarchical approaches in systems theory is the pervading idea that it is generally easier to handle several lower order subsystems than one aggregate system of high order (Ref. 22). The fundamental idea is to decompose the large system into subsystems and then manipulate the smaller subsystems in such a way that the objectives of the overall system are met.

In the case of decentralized large-scale system applications that deal exclusively with the specification of adequate deterministic control inputs, the objective is to cause the aggregate of local subsystem control solutions to also be the global solution. This objective is frequently accomplished by coordination. That is, the globally optimum solution being homed-in upon asymptotically by the aggregate of locally optimum solutions is frequently accomplished via a central controller (viz., supremal controller). This general approach of utilizing a supremal coordinator involves "interconnection constraints" being routinely imposed as a further requirement to be satisfied by the assorted controls provided by the individual infimal subsystems.

For applications of decentralized control, a few of the more common approaches for implementing coordination (Ref. 22) are:

- The Prediction Principle (Ref. 23), where the supremal controller predicts a value for the interconnection variables, provides it to each local subsystem, and allows each local subsystem to proceed autonomously with its own local optimization calculations. Eventually, the predicted interconnection variable is checked, updated, and reissued.
- The Balance Principle (Refs. 23 and 24), where each infimal controller treats its interconnection variable as one of the controls to be specified, then the aggregate of calculated decentralized infimal controls is checked for conformity with the original interconnection variable.
- Use of Penalty Functions (Ref. 25), where the interconnection constraint is adjoined to the standard cost function via a penalty function.

Remark 1.3-1: While conceptually useful, Ref. 22 indicates on p. 62 that the Balance Principle has practical limitations since it may give rise to singular control problems that cannot be solved by standard iterative techniques.

Remark 1.3-2: An unfortunate disadvantage of penalty functions is that they are somewhat loose and do not force exact adherence to interconnection constraints.

In stark contrast to the decentralized control problem, the decentralized estimation problem has, in general, no mechanism for enforcing interconnection constraints, since no control is involved. Historically, the mathematical structure of both decentralized control and decentralized estimation was examined in exacting detail by Pearson in Ref. 21 where the following observations were made:

- Significant computational simplifications accrue in the control problem with quadratic cost function when all the subsystems are linear (pp. 152-3 of Ref. 21);
- Techniques exist for analytically proving proper coordination via iteration between aggregates of linear subsystems via a contraction mapping argument (pp. 182-8 of Ref. 21), but this approach is appropriate only over a very brief time interval

(t_0,t_1);

- An approach to decentralized filtering simplifies the computational coordination requirements when a suboptimal rule is used (p. 182 of Ref. 21).

During a critical examination on pp. 533-4 of Ref. 13 (as motivation for offering an alternate decentralized filtering technique) it is noted that to solve an implementation of Pearson's decentralized filter "in practice would require knowledge of the sequence of observations over the entire time interval k=0 to k=N_0 which is inconsistent with the tenents of sequential estimation since these are unavailable a priori." While possibly "useful for parameter estimation (i.e., identification), Pearson's decentralized estimation approach is not of much significance for state estimation."

The approach pioneered by Sanders in Ref. 8—of restricting the local subsystem's filter to be of the so-called Surely Locally Unbiased (SLU) class—appears to follow through on the predictions of Ref. 21, where a suboptimal rule was called for to simplify the computational burden. Similarly, Shah's Sequential Partioned Algorithm (SPA) approach to decentralized filtering reported in Refs. 13 and 14 also utilizes a simplifying suboptimal rule. A return to examine these two decentralized filtering approaches in more detail occurs in Sections 2 and 3. These two approaches along with the approach of Spyer (Refs. 32 and 33), as discussed below, are emphasized herein as being perhaps potentially more useful (from the viewpoint of offering sufficient supporting rigor and ease of engineering implementation) than the other approaches encountered to date.

For completeness, it is noted that another relatively recent approach (Ref. 26) exists for implementing decentralized estimation. However, this approach is only applicable to subsystems of the following restrictive form:

$$S_i: \underset{(n_i x 1)}{\dot{x}_i(t)} = F_i\, x_i(t) + w_i'(t) + \sum_{j=1}^{N} L_{ii}\, L_{ij}\, x_j(t) \qquad (1.3\text{-}1)$$

where the

$$L_{ii}\ L_{ij} \text{ are time-invariant for } i,j=1,2,\ldots,N \qquad (1.3\text{-}2)$$

and with all local measurement structures having no interconnection effects as modeled by

$$z_i(t_k) = \bar{H}_i x_i(t_k) + v_i(t_k) \qquad (1.3\text{-}3)$$

with contributions due to other subsystems $x_j(t_k)$ $(j\neq i)$ absent in the above. Consequently, the associated augmented system has the following form:

$$\underset{(nx1)}{\dot{\underline{x}}(t)} = (\mathrm{diag}(F_1,F_2,\ldots,F_N)+C)\underline{x}(t_k) + \underline{w}'(t) \qquad (1.3\text{-}4)$$

and

$$\underline{z}(t_k) = \mathrm{diag}(\bar{H}_1,\bar{H}_2,\ldots,\bar{H}_N)\underline{x}(t_k) + \underline{v}(t_k) \qquad (1.3\text{-}5)$$

where

$$\underline{z}(t_k) \triangleq \left[z_1^T(t_k), z_2^T(t_k), \ldots, z_N^T(t_k)\right]^T \qquad (1.3\text{-}6)$$

$$\underline{v}(t_k) \triangleq \left[v_1^T(t), v_2^T(t), \ldots, v_N^T(t)\right]^T \qquad (1.3\text{-}7)$$

and, as defined in Refs. 26 and 27 the composite interconnection matrix is:

$$C \triangleq \left[L_{ii}\ L_{ij}\right] \text{ for } i,j=1,2,\ldots,N \qquad (1.3\text{-}8)$$

It is asserted in Theorem 4 of Ref. 26 that the composite interconnection matrix being factorizable as

$$C=PS \qquad (1.3\text{-}9)$$

where

$$P \triangleq \mathrm{diag}(P_1, P_2, \ldots, P_N) \qquad (1.3\text{-}10)$$

and the P_i are the positive definite solutions of the algebraic Riccati equation

$$\underline{0} = P_i P_i + P_i F_i^T + Q_i - P_i \bar{H}_i^T R_i^{-1} \bar{H}_i P_i \qquad (1.3\text{-}11)$$

and S = any arbitrary skew-symmetric matrix is necessary and sufficient for the global

optimal estimate to consist of the aggregate of local optimal estimates of the following form

$$\dot{\hat{x}}_i(t) = (F_i - K_i\bar{H}_i)\hat{x}_i(t) + K_i z_i(t) - \sum_{j=1}^{N} K_i C_{ij}^{P} \hat{x}_j(t) \qquad (1.3\text{-}12)$$

where

$$K_i \triangleq P_i \bar{H}_i^T R_i^{-1} \qquad (1.3\text{-}13)$$

An unfortunate oversight is that C_{ij}^{P}, the "appropriate perturbation of C_{ij}," is not explicitly defined in Ref. 26. A similar discussion in Ref. 27 for decentralized estimators indicated that the matrices premultiplying x_j under the summation sign in Eq. 1.3-12 are obtained by calculations of full dimension n (as in Eq. 1.2-3). This would be an unacceptably large computer burden for many applications and could only be performed at a central computing facility or central node.

A decentralized (possibly parallel-processing) algorithm for implementing the n-dimensional global exact Kalman filter in a hierarchical manner has been studied in Ref. 28. However, the processing hierarchy is dictated by an internal system structure rather than by any external imposed protocol hierarchy (e.g., as exists in the JTIDS RelNav application) and it is assumed that every subsystem has access to all the measurement data.

Other approaches to decentralized filtering (such as Refs. 29 and 30) have been surveyed, but the common requirement of having to perform a transformation to achieve "output decentralization" is incompatible with many applications (see Section 1.4 for details and Ref. 30 for numerical examples). The approach of Ref. 31 requires a central processing node (that could be vulnerable in a tactical environment as a single target whose destruction would ruin all operations).

The approach of Ref. 32 strictly pertains to decentralized LQG estimation and control of a K-node system (where the K-nodes refer to K subsystems) where local filters share their information with all the other nodes. [LQG refers to Linear Quadratic Gaussian applications involving linear systems with Gaussian measurement and process noises with a single quadratic performance index (cost function) for control.] For many filtering applications, there is

- no strong interest in the feedback control aspect,
- no constant number K of subsystems, and
- no sharing of information between all the subsystems.

However, just the filtering portion of Ref. 32 can be extricated as done in Ref. 33 with some simplifications.

Given several redundant measurement sensors of the following form

$$z_j(k) = H_j(k)x(k) + v_j(k) \text{ for } j=1,2,\ldots,M \qquad (1.3\text{-}14)$$

it is reasonably well-known (Ref. 79) that the linear least-mean-square estimate of x(k) as in Eq. 1.2-6 has the form

$$\hat{x}(k|k) = \left[P^{-1}(k|k-1) + \sum_{j=1}^{M} H_j^T(k)R_j^{-1}(k)H_j(k)\right]^{-1}\left[\sum_{i=1}^{M} H_i^T(k)R_i^{-1}(k)z_i(k) + P^{-1}(k\ k-1)\hat{x}(k\ k-1)\right] \qquad (1.3\text{-}15)$$

with associated covariance of estimation error provided by

$$P(k|k) = \left[P^{-1}(k|k-1) + \sum_{j=1}^{M} H_j^T(k)R_j^{-1}(k)H_j(k)\right]^{-1} \qquad (1.3\text{-}16)$$

where $P(k|k-1) \triangleq$ covariance of error of estimating x at k as propagated from k-1 for full state aggregate. Speyer's filter (Refs. 32 and 33) is equivalent to the following form:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \sum_{j=1}^{M(k)} K_j(k)\left[z_j(k) - H_j(k)\hat{x}(k|k-1)\right] \qquad (1.3\text{-}17a)$$

$$= \hat{x}(k|k-1) + \sum_{j=1}^{M(k)} P(k|k-1)H_j^T R_j^{-1}\left[z_j - H_j\hat{x}(k|k-1)\right] \qquad (1.3\text{-}17b)$$

$$= \hat{x}(k|k-1) + P(k|k-1) \sum_{j=1}^{M(k)} H_j^T R_j^{-1} [z_j - H_j \hat{x}(k|k-1)] \qquad (1.3\text{-}17c)$$

but where instead several decentralized local estimators are used in the mechanization as

$$\hat{x}(k|k) = \sum_{j=1}^{M(k)} \left\{ P(k|k-1) \left[P_j^{-1}(k|k) \hat{x}_j(k|k) \right] + h_j(k) \right\} \qquad (1.3\text{-}18a)$$

$$= P(k|k-1) \sum_{j=1}^{M(k)} P_j^{-1}(k|k) \hat{x}_j(k|k) + \sum_{j=1}^{M(k)} h_j(k) \qquad (1.3\text{-}18b)$$

where

$$P_j(k|k) = E\left[\left(x(k) - \hat{x}_j(k) \right) \left(x(k) - \hat{x}_j(k) \right)^T \middle| Z_j(k) \right] \qquad (1.3\text{-}19)$$

and $h_j(k)$ satisfies a recursive equation of the form

$$h_j(k) = F(k) h_j(k-1) + G_j(k) \left(z_j(k) - H_j(k)\, x_j(k|k-1) \right) \qquad (1.3\text{-}20)$$

where $F(k)$ and $G_j(k)$ are precomputable matrices specified by

$$F(k) = P(k|k-1) \left[\Phi(k+1,k) P(k-1|k-2) \Phi^T(k+1,k) + Q(k-1) \right]^{-1} \Phi(k+1,k) \qquad (1.3\text{-}21)$$

$$G_j(k) = F(k) P(k-1|k-2) P_j^{-1}(k-1|k-1) \Phi^{-1}(k+1,k) - P(k|k-1) \left[\Phi P_j(k-1|k-1) \Phi^T + Q(k-1) \right]^{-1} \qquad (1.3\text{-}22)$$

Several ways for ordering the computations of Eqs. 1.3-17a to 1.3-22 for greater efficiency become evident but all approaches represent a fairly large computational burden. Ref. 33 only stresses the reduction in computational burden without tallying the burden encountered for implementation. Useful interpretations of Eqs. 1.3-18 and 1.3-20 are provided by Gobbini on p. 35 of Ref. 80. Generalizations of the Speyer filter are found in Ref. 81 and discussed in Ref. 80.

An overview of the results of this section are summarized in Table 1.3-1. Other approaches to decentralized filtering publicized too late to be considered here are Refs. 44 and 45.

1.4 Salient Features of a Canonical Transformation to Achieve Output Decentralization

Several approaches to decentralized filtering, such as Refs. 26, 29, and 30, require that the original system be of the "output decentralized" form as a condition for applicability. An approach is available that can be applied to some large-scale systems having a general measurement structure of the form:

$$z_i(k) = \bar{H}_i x_i(k) + \hat{H}_i \sum_{\substack{j=1 \\ j \neq i}}^{N} L_{ij} x_j(k) + v_i(k) \quad (\text{for } i=1,\ldots,N) \qquad (1.4\text{-}1)$$

so that so-called "output decentralization" of the form

$$z_j(k) = H_j \tilde{x}_j(k) + \tilde{v}_j(k) \qquad (1.4\text{-}2)$$

(for $j=1,\ldots,p$) is achieved, where each subsystem has access to (and responsibility for) measurements only for that subsystem. In output decentralization, the parameter p is not constrained to be identical to N and the subsystem state and noise groupings are different, in general, from those in Eq. 1.4-1 as denoted, respectively, by $\tilde{x}_j$ and $\tilde{v}_j$.

"Output decentralization," if achievable, is attained via a single transformation that must be applied to the entire system aggregate. Only after the output decoupling transformation has been applied are distinct individual subsystems revealed. Conditions for applicability of the output decentralization transformation, its mechanization, and its general inconvenience to apply are only briefly touched upon here.

The requisite decentralizing transformation was originally presented in Ref. 42 and discussed in more detail (as Chapter 7) in Ref. 43 but only for the case of "control input decoupling" or "control decentralization" without a consideration of measurement or process noise. "Output decoupling" results were left only as an implicit afterthought (via the concept of duality between control and measurement structures). For ease in accessibility and use of the same consistent notation throughout, the specific transformation for achieving "output measurement decoupling" is explicitly presented by Kerr in Appendix B of Ref. 15 following the approach of Ref. 30 but going further to show the effect of the transformation on the measurement and process noises that are naturally

TABLE 1.3-1

SUMMARY STATUS OF ALTERNATIVE DECENTRALIZED FILTERING APPROACHES

Primary Investigators	Exhaustive Open Literature Descriptions[1]	Overview Comments
J. D. Pearson	Ref. 21 (1970)	• Historically significant treatment of general structural framework and indication of problem areas or barriers to be overcome in the future.
M. Shah	Ref. 14[2] (1971) Ref. 13 (1978) Ref. 60 (1981)	• Accommodates time-varying models. • Allows analytic proof of stability (Section 4) for the linear case. • Intuitive. • "Biased" (but indications are that this aspect can be adequately handled). • Anticipated computer burden quantified (Section 3.2). • Accommodates EKF approach to nonlinear filtering (p. 245 of Ref. 9).
C. Sanders E. C. Tacker T. D. Linton R. Y.-S Ling	Ref. 8[3] (1973) Ref. 9 (1973) Ref. 10 (1976) Ref. 11 (1978) Ref. 12 (1979)	• Accommodates time-varying models. • Allows analytical proof of stability (Section 4) for the linear case. • Exploits measurement structure (when possible) for computational savings. • Unmodified SLU requires a special SVD matrix factorization (discussed in Section 2.2). • Unmodified SLU is "unbiased." • Anticipated computer burden quantified (Section 2.4). • Somewhat intuitive.
M. K. Sundareshan	Ref. 26 (1977)	• No subsystem interconnections § allowed in the measurement models. • Crucial C_{ij}^{P} not explicitly defined (see Section 1.3).
M. F. Hassan G. Salut M. G. Singh A. Titli	Ref. 29 (1976) Ref. 28 (1978) Ref. 13 (1978)	• No subsystem interconnections § allowed in the measurement model. • All measurement data available to each subsystem.
D. D. Siljak M. B. Vukcevic	Ref. 30 (1978)	• No subsystem interconnections § allowed in the measurement models.
E. Verriest B. Friedlander M. Morf	Ref. 31 (1979)	• Assumes a central computing node that assembles local estimates into global estimates.
J. L. Speyer T. S. Chang	Ref. 32 (1979) Ref. 33 (1980)	• Framework is originally for both estimation and LQG feedback regulation control (where control in the LQG sense and not in the sense of C^2) but exclusive filter framework can be extricated (Ref. 33).

[1]No known concrete applications or defense related prior considerations of these recent theoretical developments.

[2]Designated as the Sequentially Partitioned Algorithm (SPA) in Ref. 13.

[3]Designated as the Surely Locally Unbiased (SLU) filter (with 10 modified variations of the original formulation) including:

ROM filter—Reduced Order Model
ESLU filter—Expanded Surely Locally Unbiased
ESLU filter—Same as above, but without explicitly modeling communications noise (assumed negligible).

§A transformation (detailed in Appendix B.2 of Ref. 15 and Appendix A of Ref. 35 and summarized in Section 1.4) exists for converting general output measurement coupled subsystems into "output decoupled" subsystems, but technique only applicable to time-invariant structure and must be applied to aggregate system (with an associated harsh computer burden).

encountered in filtering applications. The essential observations of Kerr in Refs. 15 and 35 are now summarized here. Note that while the output is now represented in a completely decentralized manner in Eq. 1.4-2, the p scalar measurement noises of the different subsystems are correlated (a type of coupling), in general, unless the original system of Eq. 1.4-1 as an aggregate also has uncorrelated measurement noise components, as indicated by a block diagonal measurement noise covariance matrix of the following form

$$R = \begin{bmatrix} R_1 & & \bigcirc \\ & R_2 & \\ \bigcirc & & \ddots \\ & & & R_p \end{bmatrix} \qquad (1.4\text{-}3)$$

(which is trivially achievable if the original systems aggregate block R is strictly diagonal, but not very likely otherwise). Restrictions relating to the presence of measurement and process noise as encountered here while considering the salient features of output decentralizing transformations were not considered in Refs. 30, 42, and 43.

Drawbacks Limiting Applicability of Output Decentralization

While "output decentralization" could be acceptable for many applications, its usefulness for many potential applications appears to be limited since requirements for its use appear to exceed certain reasonable computational limits as now summarized. Apparent restrictions to applying "output decentralization":

- Decentralized transformation only strictly applicable to time-invariant linear systems (where the matrices F, and $\bar{H}_i, \hat{H}_i, L_{ij}$ of Eqs. 1.4-1 and 1.4-2, respectively, are constant);
- Original grouping of N subsystems must consist entirely of "observable pairs" (H_j, F_j) otherwise computational problems with corresponding transformation matrices T_j are encountered;
- Test of "observability" requirements being met as a requisite step is a difficult computational burden for real-time verification (alternative tests are provided in Section 5 of Ref. 46);
- Formation of transformation T_j (Eq. A.2-10 of Ref. 35) appears to be a formidable computational burden for each $j=1,\ldots,N$;
- While transformations using T_j can be applied at the subsystem level in a decentralized fashion, reshuffling of the aggregate of states using the permutation matrix P (Eq. A.2-14 of Ref. 35) constitutes a computational burden that is apparently only compatible with a central computing facility, a severe restriction for some applications;
- Reshuffled "output decentralized" representation of Eqs. A.2-18 and A.2-19 of Ref. 35 will not necessarily correspond to the state identities of the original subsystems (a consequence that can be inconvenient when a desirable physical association frequently provides insight for error overrides).

The restriction of item 1 above in requiring a time-invariant system is a hardship for nonlinear applications where repeated relinearizations are time-varying in general. An application could still be accommodated, in principle, by repeated application of the T and P transformations discussed in the above items 4 to 6 after every measurement update anywhere within the total system. However, the drawback to doing this is that it represents a multifold increase in a fairly substantial computational burden that may already be impracticable in many applications to be required to perform these tedious transformations even once.

1.5 Multirate/Multidimensional Two (or Three) Filter Coordination

In the restrictive situation (prevalent in inertial navigation system applications) that the observation matrix has the following special structure

$$\underset{mxn}{H_k} = \left[\underset{mxs}{C_k} \;\middle|\; \underset{mxq}{0}\right] \qquad (1.5\text{-}1)$$

where the dimensions are such that

$$s < q \quad \text{and} \quad n = s + q \qquad (1.5\text{-}2)$$

this feature can be exploited to a computational advantage by making use of the following

matrix and vector partitioning:

$$ {}^{n\times n}P_{k|k-1} = \left[\begin{array}{c|c} {}^{s\times s}P_{k|k-1} & {}^{s\times q}P_{k|k-1} \\ \hline {}^{q\times s}P^{T}_{k|k-1} & {}^{q\times q}P_{k|k-1} \end{array}\right] ; \qquad {}^{n\times 1}\hat{X}_{k|k-1} = \left[\begin{array}{c} {}^{s\times 1}\hat{X}_{k|k-1} \\ \hline {}^{q\times 1}\hat{X}_{k|k-1} \end{array}\right] \qquad (1.5\text{-}3) $$

where leading superscripts denote the dimensions of the submatrices and vectors for clarity. Partitioning, as done above, can be used to define the two separate but compatible filters depicted in Figure 1.5-1 that can perform their processing at different rates in order to:

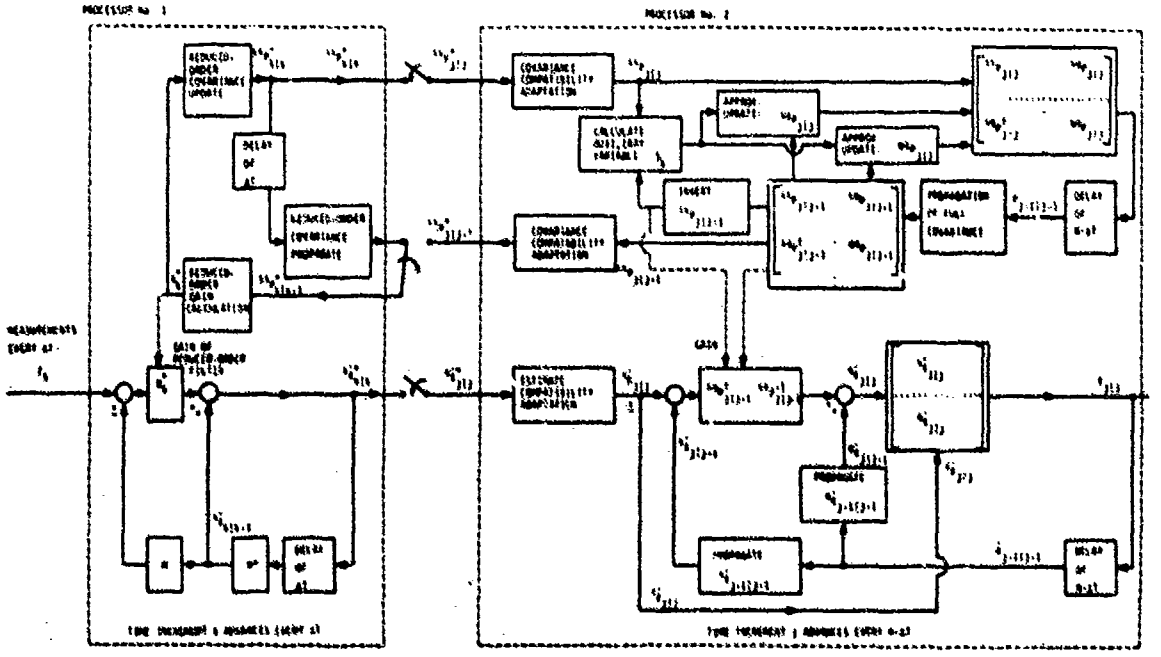
1. make use of the measurements at a fast rate (as they become available) and summarize the associated plethora of information (as a kind of rigorous data compression) in a reduced-order single estimate $s\hat{x}_{k|k}$, and

2. hand-over the distilled summary data $\hat{s}_{x_{k|k}}$ to be combined within a full-scale filter that processes at a slow rate (as required for the higher dimensional filter), but with consequently greater accuracy associated with a more detailed accounting of error contributors to be found in this higher fidelity model.

Standard Kalman filtering theory prescribes use of only a single unified filter. The remainder of this section describes the rationale that permits a theoretically rigorous implementation. The ideal structural and cycle-rate compatibility/conformability (match between what needs are anticipated for future integrated navigation system applications and what is offered by the multirate two filter configuration) follow as a consequence of structural degeneracies of Eq. 1.5-3 (due to the beneficial sparseness of the observation matrix in Eq. 1.5-1 as, respectively,

$$ P_{k|k} = P_{k|k-1} - P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T+R_k)^{-1}H_kP_{k|k-1} \qquad (1.5\text{-}4a) $$

$$ = \left[\begin{array}{c|c} {}^{s\times s}P_{k|k-1}-{}^{s\times s}P_{k|k-1}C_k^T(C_k{}^{s\times s}P_{k|k-1}C_k^T+R_k)^{-1}C_k{}^{s\times s}P_{k|k-1} & {}^{s\times q}P_{k|k-1}-{}^{s\times s}P_{k|k-1}C_k^T(C_k{}^{s\times s}P_{k|k-1}C_k^T+R_k)^{-1}C_k{}^{s\times q}P_{k|k-1} \\ \hline {}^{q\times s}P^T_{k|k-1}-{}^{q\times s}P^T_{k|k-1}C_k^T(C_k{}^{s\times s}P_{k|k-1}C_k^T+R_k)^{-1}C_k{}^{s\times s}P_{k|k-1} & {}^{q\times q}P_{k|k-1}-{}^{q\times s}P^T_{k|k-1}C_k^T(C_k{}^{s\times s}P_{k|k-1}C_k^T+R_k)^{-1}C_k{}^{s\times q}P_{k|k-1} \end{array}\right] \qquad (1.5\text{-}4b) $$



- All Switches Activate to Sample at every N·ΔT
- Compatibility Adaptation corresponds to a linear transformation to extract information pertinent to predesignated states (i.e., states of Reduced-Order filter may differ slightly from those of s_x but should be obtainable via a linear coordinate transformation)

Figure 1.5-1. MULTIRATE TWO FILTER APPROACH: Reduced-order filter processes measurements at fast rate (ΔT) and hands over estimates to more accurately detailed full-order filter for processing at the slower rate (N·ΔT).

and

$$\begin{bmatrix} {}^{s}\hat{x}_{k|k} \\ {}^{q}\hat{x}_{k|k} \end{bmatrix} = \begin{bmatrix} {}^{s}\hat{x}_{k|k-1} \\ {}^{q}\hat{x}_{k|k-1} \end{bmatrix} + \begin{bmatrix} {}^{ss}P_{k|k-1}C_k^T(C_k{}^{ss}P_{k|k-1}C_k^T+R_k)^{-1} \\ {}^{sq}P_{k|k-1}^T C_k^T(C_k{}^{ss}P_{k|k-1}C_k^T+R_k)^{-1} \end{bmatrix} (z_k - C_k{}^{s}\hat{x}_{k|k-1}) \tag{1.5-5a}$$

From Eq. 1.5-5a above, the following two component equations are extracted:

$${}^{s}\hat{x}_{k|k} = {}^{s}\hat{x}_{k|k-1} + {}^{ss}P_{k|k-1}\,[C_k^T(C_k{}^{ss}P_{k|k-1}C_k^T+R_k)^{-1}(z_k-C_k{}^{s}\hat{x}_{k|k-1})] \tag{1.5-5b}$$

$${}^{q}\hat{x}_{k|k} = {}^{q}\hat{x}_{k|k-1} + {}^{sq}P_{k|k-1}^T\,[\text{Same as within above brackets}] \tag{1.5-5c}$$

The common bracketed residual or "innovations" information provided by the measurements, occurring in both Eqs. 1.5-5a and 1.5-5c, can be solved for from Eq. 1.5-5b as:

$$C_k^T(C_k{}^{ss}P_{k|k-1}C_k^T+R_k)^{-1}(z_k-C_k{}^{s}\hat{x}_{k|k-1}) = {}^{ss}P_{k|k-1}^{-1}\,[{}^{s}\hat{x}_{k|k}-{}^{s}\hat{x}_{k|k-1}] \tag{1.5-6}$$

The explicit occurrence of this measurement residual in Eq. 1.5-5c is then eliminated to result in the following update form for the additional states of the full-state filter as

$${}^{q}\hat{x}_{k|k} = {}^{q}\hat{x}_{k|k-1} + {}^{sq}P_{k|k-1}^T\,{}^{ss}P_{k|k-1}^{-1}\,[{}^{s}\hat{x}_{k|k}-{}^{s}\hat{x}_{k|k-1}] \tag{1.5-7}$$

This equation is to be used as the basis of the update step of the full-state slow rate filter as further elaborated upon below.

Remark 1.5-1: Please notice that this derivation of the multirate filter avoids the severe (but unnecessary) restriction encountered in the original derivation offered in a slide presentation prior to Refs. 61, 62 that C_k in Eq. 1.5-1 must be invertible (or the equivalent restriction in Ref. 61 that C_k be equivalent to an identity matrix). A removal of the invertibility restriction was accomplished independently for the derivation presented here and by the later work of Ref. 61.

From Eq. 1.5-4b, the following three recursive relationships are obtained as submatrices within the partitioning:

$${}^{ss}P_{k|k} = [I-{}^{ss}P_{k|k-1}C_k^T(C_k{}^{ss}P_{k|k-1}C_k^T+R_k)^{-1}C_k]\,{}^{ss}P_{k|k-1} \tag{1.5-8}$$

$${}^{sq}P_{k|k} = [\text{Same as within above brackets}]\,{}^{sq}P_{k|k-1} \tag{1.5-9}$$

$${}^{qq}P_{k|k} = {}^{qq}P_{k|k-1} - {}^{sq}P_{k|k-1}^T\,{}^{ss}P_{k|k-1}^{-1}\,(I-[\text{Same as within above brackets}])\,{}^{sq}P_{k|k-1} \tag{1.5-10}$$

Eq. 1.5-8 can then be indirectly solved computationally for the common expression within brackets (as reoccurs throughout Eqs. 1.5-8 to 1.5-10) by only a single low dimensional (sxs) matrix inversion and matrix multiply as

$$[I-{}^{ss}P_{k|k-1}C_k^T(C_k{}^{ss}P_{k|k-1}C_k^T+R_k)^{-1}C_k] = {}^{ss}P_{k|k}\,{}^{ss}P_{k|k-1}^{-1} \tag{1.5-11}$$

Further computational simplifications accrue in evaluating Eq. 1.5-9 and Eq. 1.5-10 by utilizing a fundamental assumption that the following information is a good approximation at the instant of a fast/slow (i.e., reduced-order filter/full-state filter) transfer (every $N\cdot\delta T$ units of time where δT denotes the step-size of the reduced-order filter, while $N\cdot\delta T$ denotes the step-size of the associated full-state filter) as a consequence of the reduced-order filter's timely utilization of measurements z_k:

$${}^{s}\hat{x}_{k|k} = {}^{s}\hat{x}^{*}_{k|k} \tag{1.5-12}$$

$${}^{ss}P_{k|k} = {}^{ss}P^{*}_{k|k} \tag{1.5-13}$$

(where a superscript asterisk denotes calculations performed by the reduced-order filter). Exploitation of this reasonable assumption yields a computational approximation for the bracketed term of Eq. 1.5-11 as

$$S_k \triangleq {}^{ss}P^{*}_{k|k}\,{}^{ss}P_{k|k-1}^{-1} \tag{1.5-14}$$

from which a simplification of the update procedure for the full-state filter ensues from Eqs. 1.5-14, 1.5-8, 1.5-9, and 1.5-10 as:

$$ {}^{ss}P_{k|k} = {}^{ss}P^{*}_{k|k} \tag{1.5-15} $$

$$ {}^{sq}P_{k|k} = S_k \, {}^{sq}P_{k|k-1} \tag{1.5-16} $$

$$ {}^{qq}P_{k|k} = {}^{qq}P_{k|k-1} - {}^{sq}P^{T}_{k|k-1} \, {}^{ss}P^{-1}_{k|k-1} \, (I-S_k) \, {}^{sq}P_{k|k-1} \tag{1.5-17} $$

Similarly utilizing the assumption of Eq. 1.5-12, the update procedure for the remainder of the full-state filter provided by Eq. 1.5-7 simplifies to

$$ \hat{x}_{k|k} = \hat{x}_{k|k-1} + {}^{sq}P^{T}_{k|k-1} \, {}^{ss}P^{-1}_{k|k-1} \, [{}^{s}\hat{x}^{*}_{k|k} - {}^{s}\hat{x}_{k|k-1}] \tag{1.5-18} $$

$$ {}^{s}\hat{x}_{k|k} = {}^{s}\hat{x}^{*}_{k|k} \tag{1.5-19} $$

Observe that in both Eqs. 1.5-17 and 1.5-18, an additional computational consideration is the requirement for calculating the inverse of ${}^{ss}P_{k|k-1}$. This is easily accommodated since ${}^{ss}P_{k|k-1}$ is symmetric positive definite and of low dimension (sxs) and is to be calculated within the digital processor hosting the full state filter which, in turn, is allotted more time ($N \cdot \Delta T$) to complete its requisite calculations (notice the beneficial parallel processing associated with hosting the two filter algorithms on different machines). For both estimator and covariance calculations performed in the full-state filter, an exact implementation of the propagate step is recommended (and likewise easily accommodated due to the longer time of $N \cdot \Delta T$ allotted for computations associated with the full-state filter).

Coordination of the two reduced-order/full-state filters is achieved by passing the following covariance information back from the full-state to the reduced-state filter:

$$ {}^{ss}P^{*}_{k|k-1} = {}^{ss}P_{k|k-1} \tag{1.5-20} $$

at the $N \cdot \Delta T$ instants of data transfer. In this manner, the covariances of the reduced-order filter are not permitted to diverge away from the better overview assessments (due to a more detailed accounting of cross-correlations and considerations of additional significant error contributors) of the full-state filter. Thus the salient features and theoretical rationale underlying the operation of multirate reduced-order/full-state filters, as depicted in Figure 1.5-1, have been simply described.

A technique for performing a reduced-order suboptimal filter analysis for this multirate filtering approach to ascertain its appropriateness for a particular application is described in Ref. 61. Generalizations are immediate for extending this approach to three (or more) concatenated filters or one series/two parallel filters, if need be, for either the previously mentioned multirates of fast/slow or for three processing rates—fast/medium/slow. However, it is important to note the ordering hierarchy where the reduced-order filter model is a proper subset (perhaps by a similarity transformation) of the larger filter model.

2 THE SURELY LOCALLY UNBIASED (SLU) APPROACH FOR DECENTRALIZED FILTERING

2.1 Layout of Section 2

Justification for a detailed discrete-time treatment of the SLU filtering formulation is provided in Section 2.2.1. The special structure that must be exhibited by system observation matrices to enable use of SLU filtering to a computational advantage is described in Section 2.2.2 and matrix transformations to enable verification of this desirable structure are provided. The mechanization equations for implementing the SLU algorithm are provided in Section 2.3 and summarized as nine (9) entries in Table 2.3-1. A table of SLU operation counts to quantify the computer burden upon mechanization is provided in Section 2.4.

2.2 Derivation of the Discrete-Time SLU Filter

2.2.1 Motivation for Proceeding with a Discrete-Time Derivation

An application of the Matrix Minimum Principle (Refs 1 and 2), a well-known tool for the derivation of the continuous-time Kalman filter (Ref. 3), is presented here for an exclusively discrete-time derivation of the discrete-time SLU decentralized filtering equations. This demonstration is provided here for the following reasons:

- The SLU approach of Refs. 8-14 to decentralized filtering is via utilization of the continuous-time version of the Matrix Minimum Principle, where the decentralization requirement is essentially included in the routine analysis as an additional structural constraint, and the filter gains are then optimized to provide the minimum

variance solutions for the prescribed filter structures.

- The SLU results of Refs. 8-14 for decentralized filtering are stated and derived only within a continuous-time framework (discrepancies that can arise between discretization of continuous-time results and results derived exclusively via discrete-time techniques are noted and explained on pp. 136-9 of Ref. 4, on p. 297 of Ref. 5, and other difficulties discussed in Ref. 6); however, practical applications implemented with a digital computer appear to be better suited to the discrete-time formulation provided below. Indeed, realistic operations counts as provided in Section 2.4 to analytically quantify the required computer burden can be performed only for the discrete-time formulation (p. 93 of Ref. 12).

- The milestone application of the Matrix Minimum Principle to filtering problems (Ref. 3) unfortunately provides no clarification for why the particular cost function of just a terminal accuracy constraint was chosen. Use of the same terminal accuracy constraint persisted without clarification throughout Refs. 3, 8-12, and 7, but is questioned and appropriately modified both in similar circumstances in Refs. 47, 48, 49, and here.

First the special structure required for SLU filtering is identified in Section 2.2.2 and the descriptive transformation is described. The SLU filter gains are specified in Section 2.3 and the form of the local Riccati equation that must be recursively solved by each subsystem is also specified.

2.2.2 Special Structure Required for SLU Filtering

The discrete-time subsystem model that appears to be most appropriate for Surely Locally Unbiased (SLU) filtering is presented in Eqs. 1.2-40 and 1.2-41. The approach of Refs. 8-12 is to assume that each local estimator is of the following form:

$$\hat{x}_i(k+1|k) = \Gamma_i(k+1,k)\,\hat{x}_i(k|k-1) + K_i(k)z_i(k) \tag{2.2-1}$$

and the corresponding error in subsystem estimation is

$$e_i(k+1|k) \triangleq x_i(k+1) - \hat{x}_i(k+1|k) \tag{2.2-2}$$

The subsystem error of estimation evolves in time according to:

$$\begin{aligned} e_i(k+1|k) = {} & \Phi_{ii}(k+1,k)x_i(k) + \dot{L}_{ii}(k)u_i(k) + \dot{w}_i(k) - \Gamma_i(k+1,k)\hat{x}_i(k|k-1) \\ & - \Gamma_i(k+1,k)x_i(k) + \Gamma_i(k+1,k)x_i(k) \\ & - K_i(k)\left[H_i(k)\underline{x}(k) + v_i(k)\right] \end{aligned} \tag{2.2-3a}$$

$$\begin{aligned} = {} & \Gamma_i(k+1,k)e_i(k|k-1) \\ & + \left[\Phi_{ii}(k+1,k)P_{x,i}H_i^T(k)L_i(k) - K_i(k)H_i(k) - \Gamma_i(k+1,k)P_{x,i}\right]\underline{x}(k) \\ & + \left[\dot{\beta}_i(k) - K_i(k)v_i(k)\right] \end{aligned} \tag{2.2-3b}$$

The desirable property of the subsystem estimator providing conditionally unbiased estimates as

$$E\left[\hat{x}_i(k+1|k)\,|\,\underline{z}_i(k)\right] = E\left[x_i(k+1)\,|\,\underline{z}_i(k)\right] \tag{2.2-4}$$

imposes the following structural constraint (justified as on p. 291 of Ref. 7) on the middle coefficient matrix of Eq. 2.2-3b as

$$\underline{0} = \left[\Phi_{ii}(k+1,k)P_{x,i} + \dot{L}_{ii}(k)L_i(k) - K_i(k)H_i(k) - \Gamma_i(k+1,k)P_{x,i}\right] \tag{2.2-5}$$

By exploiting the structural decomposition of the observation matrix $H_i(k)$ as provided in Eq. 1.2-2d, the single condition of Eq. 2.2-5 (being satisfied for conditionally unbiased estimates) can be naturally decomposed for convenience into the following two conditions:

$$\left[\Phi_{ii}(k+1,k) - K_i(k)\bar{H}_i(k) - \Gamma_i(k+1,k)\right]P_{x,i} = \underline{0} \tag{2.2-6}$$

and

$$\left[\dot{L}_{ii}(k) - K_i(k)\hat{H}_i(k)\right]L_i(k) = \underline{0} \tag{2.2-7}$$

Via the definition of $L_i(k)$ in Eq. 1.2-4, the single condition of Eq. 2.2-7 can be further decomposed into an aggregate of several conditions as

$$\left[\mathring{L}_{ii}(k)-K_i(k)\hat{H}_i(k)\right] L_{ij}(k) = \underline{0} \quad \text{for } j = 1,2,\ldots,N/(i) \tag{2.2-8}$$

while the condition of Eq. 2.2-6 (as it applies to each of the component subsystems i=1,2,...,N) is equivalent to a familiar structural constraint of

$$\Gamma_i(k+1,k) = \Phi_{ii}(k+1,k)-K_i(k)\bar{H}_i(k) \tag{2.2-9}$$

to yield decentralized estimators of the form

$$\hat{x}_i(k+1|k)=\left[\Phi_{ii}(k+1,k)-F_i(k)\bar{H}_i(k)\right]\hat{x}_i(k|k-1)+K_i(k)\, z_i(k) \tag{2.2-10}$$

Before looking deeper into what is structurally involved in satisfying the condition of Eq. 2.2-8, observe that the weaker requirement to provide unconditionally unbiased estimates as explicitly represented by

$$E\left[\hat{x}_i(k+1|k)\right] = E\left[x_i(k+1)\right] \tag{2.2-11}$$

relies only upon use of the appropriate filter initial conditions:

$$\hat{x}_i(0|-1)=E\left[x_i(0)\right]=\underline{0} \tag{2.2-12}$$

as can be seen by taking total expectations throughout Eq. 2.2-3b and noting that only the first term makes a nonzero contribution unless $E[e_i(0|-1)] = 0$ (equivalent to Eq. 2.2-12).

Returning to parallel the continuous-time structural investigation of Ref. 8, the decentralized gain matrix $K_i(k)$ (appearing in Eq. 2.2-10) is defined to be of the Surely Locally Unbiased Class denoted by

$$\{SLU_i(k)\} \triangleq \overset{(n_i \times q_i)}{\{K_i(k)} \text{ such that } \mathring{L}_{ii}(k)-K_i(k)\hat{H}_i(k)=\underline{0}\} \tag{2.2-13}$$

if and only if

$$K_i(k) \text{ is a member of } \{SLU_i(k)\} \tag{2.2-14}$$

Notice that a $n_i \times q_i$ gain matrix being an element of $\{SLU_i(k)\}$ is a sufficient condition for satisfying the remaining conditionally unbiased constraint of Eqs. 2.2-7 or 2.2-8.

It is convenient to consider two other subsets of matrices corresponding to the i^{th} subsystem:

$$\Theta_i(k) \triangleq \left\{\overset{(n_i \times p_i)}{\Theta_i(k)} \text{ such that } \Theta_i(k)L_{ij}(k) = 0 \text{ for } j=1,2,\ldots,N/(i)\right\} \tag{2.2-15}$$

and

$$\Omega_i(k) \triangleq \left\{\overset{(n_i \times q_i)}{K_i(k)} \text{ such that there exists an element } \Theta_i'(k) \text{ a member of } \Theta_i(k) \text{ such that } L_{ii}(k) - K_i(k)H_i(k) + \Theta_i'(k) = \underline{0}\right\} \tag{2.2-16}$$

An interesting observation (as in Ref. 8) is that the condition of Eq. 2.2-8 is a subset of the class represented by $\Omega_i(k)$ in Eq. 2.2-16. This can be demonstrated by first noting that for

$$K'_i(k) \text{ is a member of } \Omega_i(k) \tag{2.2-17}$$

implies that there exists an $\Theta_i'(k)$ with

$$\Theta_i'(k)L_{ij}(k) = 0 \quad \text{for } j=1,2,\ldots,N/(i) \tag{2.2-18}$$

and

$$\mathring{L}_{ii}(k) - K_i(k)\,\hat{H}_i(k) + \Theta_i'(k) = 0 \tag{2.2-19}$$

Now postmultiplying Eq. 2.2-19 by $L_{ij}(k)$ yields

$$\left[\mathring{L}_{ii}(k)-K_i(k)\hat{H}_i(k)\right] L_{ij}(k)+\Theta_i'(k)\, L_{ij}(k) = \underline{0}+\underline{0}=\underline{0} \tag{2.2-20}$$

which simplifies via Eq. 2.2-18 to condition Eq. 2.2-8. It is immediate that the null

matrix $\underset{(n_i \times p_i)}{\underline{0}}$ is an element of $\theta_i(k)$ so

$$\{SLU_i(k)\} \subset \Omega_i(k) \subset \{\underset{(n_i \times q_i)}{K_i(k)} \text{ such that Eq. 2.2-8 is satisfied}\} \qquad (2.2\text{-}21)$$

A safe suboptimal decentralized estimation policy is to optimize the filter gains only over the class $\{SLU_i(k)\}$ of SLU filters to guarantee subsystem filters that are conditionally and unconditionally unbiased.

Remark 2.2.2-1: In Ref. 8, it is stated that the condition of Eq. 2.2-8 is equivalent to the set $\Omega_i(k)$ as defined in Eq. 2.2-16 (but a confirming proof eluded this author); however, the result of equivalence, while more satisfactory, was not utilized as being necessary here.

The only initial worry at the other extreme is that the SLU class may be empty; however, structural conditions that guarantee a realistic nonvacuous optimization will now be elucidated.

A nonvacuous optimization over the SLU class is guaranteed if the following condition holds.

Condition 2.2-1 (Ref. 8):

$$\text{rank } (\hat{H}_i(k)) = p_i < q_i \qquad (2.2\text{-}22)$$

where q_i is the dimension of the i^{th} local measurement, or, equivalently, satisfying a canonical structural assumption of

Condition 2.2-1' (Ref. 8):

$$\hat{H}_i(k) = \left[\begin{array}{c} I_{p_i} \\ \hline 0 \end{array}\right] \begin{array}{l} \} p_i \\ \} q_i - p_i \end{array} \qquad (2.2\text{-}23)$$

Adherence to the canonical structural assumption of Condition 2.2-1' can be forced for any arbitrary $\hat{H}_i(k)$ satisfying Condition 2.2-1 of Eq. 2.2-22 via the following result on p. 47 of Ref. 50.

Theorem 2.2-1:

For $\hat{H}_i(k)$, a $(q_i \times p_i)$-matrix of rank p_i, there exists a nonsingular $(p_i \times p_i)$-matrix $\Sigma_{i1}(k)$ and nonsingular $(q_i \times q_i)$-matrix $\Sigma_{i2}(k)$ such that

$$\left[\begin{array}{c} I_{p_i} \\ \hline 0 \end{array}\right] = \Sigma_{i2}^{-1}(k)\, \hat{H}_i(k)\, \Sigma_{i1}(k) \qquad (2.2\text{-}24a)$$

Proof that such a factorization is possible is deferred in Ref. 8 to Ref. 50 where only a standard existence proof is provided. A theoretical loose-end, previously unsolved, is how to computationally execute the factorization called for in Eq. 2.2-24a that is guaranteed to be theoretically possible from Theorem 2.2-1, but not yet specified in the literature on this subject. During the course of this investigation of Ref. 15, it was recognized that the so-called Singular Value Decomposition (SVD) algorithm (Ref. 38) will suffice to solve the factorization problem of Eq. 2.2-24 and that existing coded, standardized, and validated FORTRAN software implementations exist [e.g., EISPACK (pp. 265-9 of Ref. 39)].

In order to demonstrate this pleasant resolution, first consider the following general form of the answer that is returned from an SVD algorithmic operation as

$$\underset{H_i}{\overset{(q_i x p_i)}{\overset{\wedge}{}}} = \underset{U_i}{\overset{(q_i x q_i)}{}} \begin{bmatrix} s_1 & & & \\ & s_2 & & 0 \\ & & \ddots & \\ & 0 & & s_{p_i} \\ \hline & & 0 & \end{bmatrix} \underset{V_i^T}{\overset{(p_i x p_i)}{}} \qquad (2.2\text{-}24b)$$

for $q_i > p_i$, where additionally V_i and U_i are orthogonal:

$$U_i^{-1} = U_i^T; \; V_i^{-1} = V_i^T$$

Further, notice that upon rearranging Eq. 2.2-24b the following results

$$\sigma_i^{-1} \hat{H}_i (V_i^T)^{-1} \begin{bmatrix} s_1 & & & 0 \\ & s_2 & & \\ & & \ddots & \\ 0 & & & s_{p_i} \end{bmatrix}^{-1} = \begin{bmatrix} I_{p_i} \\ \hline 0 \end{bmatrix} \qquad (2.2\text{-}24c)$$

Now Eq. 2.2-24c is recognized to be exactly of the form of Eq. 2.2-24a that is sought (by making the following associations or assignments) with

$$z_{i1}^{-1} \triangleq U_i^T$$

and

$$z_{i2} \triangleq V_i \begin{bmatrix} \frac{1}{s_1} & & & \\ & \frac{1}{s_2} & & 0 \\ & & \ddots & \\ 0 & & & \frac{1}{s_{p_i}} \end{bmatrix}$$

Caveat: In Ref. 9, a comment occurs prior to Eq. 7 that the rank condition (of Theorem 2.2-1) being satisfied "roughly speaking" means "that at least one element of the state vector of each user is measured and that each of the interactions to each user has a direct effect on the local measurement obtained by the user." Similarly on p. 17 of Ref. 8, the above loose interpretation is reinforced in stating that the rank condition "means intuitively that the user can observe all the interactions to his unit." Both the above two intuitive structural requirements are adhered to in the JTIDS RelNav application. While the equivalence of Condition 1 and 1' (of Ref. 8 that correspond exactly to Condition 2.2-1 and Theorem 2.2-1 above) have been validated in Ref. 15, the verification of the "loose interpretations" implying compliance with Conditions 1 and 1' of Ref. 8 cannot be verified and are not satisfied by all applications, as demonstrated to be violated for JTIDS RelNav as shown in Eq. 4-13 of Ref. 35.

Earlier examples in Refs. 8-12 of how to apply the SLU filter to power system applications had degenerate "special" internal system structure that avoids the necessity of performing the explicit factorization that is usually required (pp. 23-4 of Ref. 8).

By the above defined transformation involving $z_{i1}(k)$ and $z_{i2}(k)$, the following transformation rule

$$\dot{L}_{ii} \longrightarrow \dot{L}_{ii}(k) z_{i1}(k) \triangleq \dot{L}'_{ii}(k) \qquad (2.2\text{-}25)$$

$$L_i \longrightarrow z_{i1}^{-1}(k) L_i(k) \triangleq L'_i(k) \qquad (2.2\text{-}26)$$

$$v_i(k) \longrightarrow z_{i2}^{-1}(k) v_i(k) \triangleq v'_i(k) \quad x \qquad (2.2\text{-}27)$$

$$\hat{H}_i(k) \longrightarrow z_{i2}^{-1}(k) \hat{H}_i(k) z_{i1}(k) = \begin{bmatrix} I_{p_i} \\ \hline 0 \end{bmatrix} \qquad (2.2\text{-}28)$$

$$\bar{H}_i(k) \longrightarrow z_{i2}^{-1}(k) \bar{H}_i(k) \triangleq \bar{H}'_i(k) \quad x \quad x \qquad (2.2\text{-}29)$$

$$z_i(k) \longrightarrow z_{i2}^{-1}(k) z_i(k) \triangleq z'_i(k) \qquad (2.2\text{-}30a)$$

$$R_i(k) \longrightarrow z_{i2}^{-1}(k) R_i(k) z_{i2}^{-T}(k) \triangleq R'_i(k) \qquad (2.2\text{-}30b)$$

$$K_i(k) \longrightarrow K_i(k) z_{i2}(k) \triangleq K'_i(k) \tag{2.2-31}$$

alters the original subsystem model of Eqs. 1.2-40 and 1.2-41, respectively, to be

$$x_i(k+1) = \phi_{ii}(k+1,k)\, x_i(k) + \left\{ \dot{L}_{ii}(k) z_{i1}^{-1}(k) \right\} z_{i1}(k) L_i(k) \underline{x}(k) + \dot{w}_i(k) \tag{2.2-32}$$

and

$$\begin{aligned} z'_i(k) &\triangleq z_{i2}^{-1}(k) z_i(k) \\ &= z_{i2}^{-1}(k) \bar{H}_i(k) x_i(k) + \begin{bmatrix} I_{p_i} \\ \hline O \end{bmatrix} z_{i1}^{-1}(k) L_i(k) \underline{x}(k) + z_{i2}^{-1} v_i(k) \end{aligned} \tag{2.2-33}$$

In conformity with this canonical transformation, the filter of Eq. 2.2-10 is also transformed into

$$\hat{x}_i(k+1|k) = \left[\phi_{ii}(k+1,k) - K'_i(k) \bar{H}'_i(k) \right] \hat{x}_i(k|k-1) + K'_i(k) z'_i(k) \tag{2.2-34}$$

Upon partitioning the following matrices and vectors to conform to the transformation of Eqs. 2.2-25 to 2.2-33, the result is

$$\underset{(q_i \times 1)}{z'_i(k)} = \begin{bmatrix} z'_{i1}(k) \\ \hline z'_{i2}(k) \end{bmatrix} \begin{matrix} \downarrow p_i \\ \uparrow q_i - p_i \end{matrix} = \begin{bmatrix} \bar{H}'_{i1}(k) \\ \hline \bar{H}'_{i2}(k) \end{bmatrix} x_i(k) + \begin{bmatrix} I_{p_i} \\ \hline O \end{bmatrix} L'_i(k) \underline{x}(k) + \begin{bmatrix} v'_{i1}(k) \\ \hline v'_{i2}(k) \end{bmatrix} \tag{2.2-35}$$

$$K'_i(k) = \left[K'_{i1}(k) \,\vdots\, K'_{i2}(k) \right] \tag{2.2-36}$$

$$R'_i(k) = \begin{bmatrix} \overbrace{R'_{i1}(k)}^{p_i} & \vdots & R'_{i3}(k) \\ \hline R'_{i3}(k) & \vdots & R'_{i2}(k) \end{bmatrix} \begin{matrix} \downarrow p_i \\ \uparrow q_i - p_i \end{matrix} \tag{2.2-37}$$

Substituting the partitioned matrices into Eq. 2.2-34, yields the following simplification

$$\hat{x}_i(k+1|k) = \left[\phi_{ii}(k+1,k) - K'_{i1}(k) \bar{H}'_{i1}(k) - K'_{i2}(k) \bar{H}'_{i2}(k) \right] \hat{x}_i(k|k-1) + K'_{i1}(k) z'_{i1}(k) + K'_{i2}(k) z'_{i2}(k) \tag{2.2-38a}$$

$$= \phi_{ii}(k+1,k) \hat{x}_i(k|k-1) + K'_{i1}(k) \left[z'_{i1}(k) - \bar{H}'_{i1}(k) \hat{x}_i(k|k-1) \right] + K'_{i2}(k) \left[z'_{i2}(k) - \bar{H}'_{i2}(k) \hat{x}_i(k|k-1) \right] \tag{2.2-38b}$$

It is at this point that attention is restricted to filter structures corresponding only to members of the SLU class (a class having guaranteed conditional unbiasedness by virtue of satisfying Eq. 2.2-13 as a condition for membership). The defining condition for SLU membership is also altered to reflect the transformations of Eqs. 2.2-33 and 2.2-34, embodied in the pertinent changes of Eqs. 2.2-25, 2.2-28, and 2.2-31, to yield

$$\underline{0} = \left[\dot{L}_{ii}(k) - K_i(k) \hat{H}_i(k) \right] \Phi_{ii}(k) \tag{2.2-39a}$$

$$= \dot{L}_{ii}(k) \Phi_{ii}(k) - K_i(k) \Phi_{i2}(k) \left[\Phi_{i2}^{-1}(k) \hat{H}_i(k) \Phi_{ii}(k) \right] \tag{2.2-39b}$$

$$= \dot{L}'_{ii}(k) - \left[K'_{i1}(k) \,\vdots\, K'_{i2}(k) \right] \begin{bmatrix} I_{p_i} \\ \hline O \end{bmatrix} \tag{2.2-39c}$$

$$= \dot{L}'_{ii}(k) - K'_{i1}(k) \tag{2.2-39d}$$

Therefore

$$K'_{i1}(k) = \dot{L}'_{ii}(k) \tag{2.2-40}$$

for filters of the SLU class, when expressed in terms of the canonical transformation of Eqs. 2.2-25 to 2.2-31. These filters, having the gain $K'_{i2}(k)$ as yet unspecified, have the following form:

$$\hat{x}_i(k+1|k)=\phi_{ii}(k+1,k)\hat{x}_i(k|k-1)+\hat{U}_{ii}(k)\left[z'_{i1}(k)-\bar{H}'_{i1}(k)\hat{x}_i(k|k-1)\right]+K'_{i2}(k)\left[z'_{i2}(k)-\bar{H}_{i2}(k)\hat{x}_i(k|k-1)\right] \quad (2.2\text{-}41)$$

While it is not strictly necessary to transform the original system to the canonical form of Eqs. 2.2-32 and 2.2-33 to utilize the optimal SLU decentralized filtering results, it is necessary to find the appropriate $Z_{i1}(k)$, $Z_{i2}(k)$, and calculate both $Z_{i1}^{-1}(k)$ and $Z_{i2}^{-1}(k)$ (since Z_{i1} and Z_{i2} are defined in terms of orthogonal and diagonal matrices following for Eq. 2.2-24c, no matrix inversion is needed here) for each measurement $z_i(k)$. The mechanization equations of SLU filtering are summarized in 9 simple steps in Table 2.3-1. The optimization to specify the remaining gains K_{i2}^* (i=1,2,...,N) is performed next in Section 2.3 using the Matrix Minimum Principle.

2.3 Specification of Optimal Gain and Ricatti Equations for the Decentralized SLU Class

In preparation for specifying the equation for the global estimation error, notice that Eq. 2.2-3b simplifies for the SLU class as

$$e_i(k+1|k) = \Gamma_i(k+1,k)e_i(k|k-1)+\left[w_i(k)-K_i(k)v_i(k)\right] \quad (2.3\text{-}1)$$

which, when augmented for all N subsystems, yields

$$\underline{e}(k+1|k)=\underline{\Gamma}(k+1,k)\underline{e}(k|k-1)+\underline{\tilde{W}}(k) \quad (2.3\text{-}2)$$

where

$$\underline{e}(k+1|k)\triangleq\sum_{i=1}^{N} P_{x,i}^T\, e_i(k+1|k) \quad (2.3\text{-}3)$$

$$\underline{\Gamma}(k+1,k)\triangleq\sum_{i=1}^{N} P_{x,i}^T\, \Gamma_i(k+1,k)P_{x,i} \quad (2.3\text{-}4)$$

$$\underline{\tilde{W}}(k)\triangleq\sum_{i=1}^{N} P_{x,i}^T\left[\dot{w}_i(k)-K_i(k)v_i(k)\right] \quad (2.3\text{-}5)$$

From $\dot{w}_i(k)$ and $v_i(k)$, the composite process $\underline{\tilde{W}}(k)$ inherits its properties of also being zero mean, white, Gaussian, and independent of the initial condition.

Step 1. Obtaining the difference equation for the time evolution of the covariance of the estimation error in a manner analogous to what is done in Refs. 3 and 7 for the "centralized" case, postmultiplying Eq. 2.3-2 by its own transpose and taking expectations throughout yields

$$\Sigma(k+1|k)\triangleq E\left[\underline{e}(k+1|k)\underline{e}^T(k+1|k)\right] \quad (2.3\text{-}6a)$$

$$=\underline{\Gamma}(k+1,k)\,\Sigma(k|k-1)\,\underline{\Gamma}^T(k+1,k)+\sum_{i=1}^{N} P_{x,i}^T\dot{Q}_i(k)P_{x,i}+P_{x,i}^T K_i(k)R_i(k)K_i^T(k)P_{x,i} \quad (2.3\text{-}6b)$$

A further association for the SLU case of unbiased filters being

$$\underline{\Gamma}(k+1,k)\triangleq\sum_{i=1}^{N} P_{x,i}^T\left[\phi_{ii}(k+1,k)-K_i(k)\bar{H}_i(k)\right]P_{x,i} \quad (2.3\text{-}7)$$

allows the observation of Ref. 8 that the solution to Eq. 2.3-6c can be shown by direct substitution to be

$$\Sigma(k+1|k)=\sum_{i=1}^{N} P_{x,i}^T\, P_i(k+1|k)\, P_{x,i} \quad (2.3\text{-}8)$$

where each $P_i(k+1|k)$ evolves in time according to the local discrete-time Riccati equations of the form

$$P_i(k+1|k)=\left[\phi_{ii}(k+1,k)-K_i(k)\bar{H}_i(k)\right]P_i(k|k-1)\left[\phi_{ii}(k+1,k)-K_i(k)\bar{H}_i(k)\right]^T+\dot{Q}_i(k)+K_i(k)R_i(k)K_i^T(k) \quad (2.3\text{-}9a)$$

$$=\left[\phi_{ii}(k+1,k)-\tilde{L}_{i1}(k)\bar{H}'_{i1}(k)-K'_{i2}(k)\bar{H}'_{i2}(k)\right]P_i(k|k-1)\left[\phi_{ii}(k+1,k)-\tilde{L}'_{i1}(k)\bar{H}'_{i1}(k)-K'_{i2}(k)\bar{H}'_{i2}(k)\right]^T$$

$$+\left[\tilde{C}_{i1}(k)-\tilde{L}_{i1}(k)R'_{i1}(k)\tilde{L}_{i1}^T(k)\right]+K'_{i2}(k)R'_{i2}(k)\tilde{L}_{i1}^T(k)+\tilde{L}_{i1}(k)R'_{i2}(k)K'^T_{i2}(k)+K'_{i2}(k)R'_{i2}(k)K'_{i2}(k) \quad (2\text{-}3\text{-}9b)$$

with natural initial condition

$$P_i(0|-1) = \overset{\circ}{P}_i \tag{2.3-10}$$

Step 2. Specifying the appropriate scalar performance measure or cost function to be the weighted-mean-squared error in estimation [to then be globally minimized in specifying the optimal gains K_{i2}^* (i=1,...,N) for the decentralized SLU filters] as

$$\bar{J} = E\{\sum_{k=0}^{N_o} \underline{e}^T(k|k-1)\bar{M}(k)\underline{e}^T(k|k-1)\} \tag{2.3-11a}$$

$$= \sum_{k=0}^{N_o}\sum_{i=1}^{N} E\left[e_i^T(k|k-1)M_i(k)e_i(k|k-1)\right] \tag{2.3-11b}$$

$$= \sum_{i=1}^{N} J_i\left[\{K'_{i2}(k)\}_{k=0}^{N_o}, \{M_i(k)\}_{k=0}^{N_o}, \{P_i(k|k-1)\}_{k=0}^{N_o}\right] \tag{2.3-11c}$$

where

$$\bar{M}(k) \triangleq \text{diag}\{M_1(k), M_2(k), \ldots, M_N(k)\} \tag{2.3-12}$$

the $M_i(k)$ are positive definite, and

$$J_i[\cdot] \triangleq \sum_{k=0}^{N_o} \text{tr}[P_i(k|k-1)M_i(k)] \tag{2.3-13}$$

A simplification now occurs in noting that minimizing 2.3-11a over the aggregate of SLU classes is identical to minimizing every local non-negative cost function of Eq. 2.3-13 over its i^{th} SLU class, so that

$$J_i\left[\{K_{i2}^*(k)\}_{k=0}^{N_0}, \{M_i(k)\}_{k=0}^{N_0}\right] \leq J_i\left[\{K'_{i2}(k)\}_{k=0}^{N_0}, \{M_i(k)\}_{k=0}^{N_0}\right] \tag{2.3-14}$$

While global optimality is sought, it has just been shown to degenerate into N local optimization problems, each of the same basic structure. Of course, the measurement data base over which optimization is performed is more restricted as an approximation. The Hamiltonian for the i^{th} local optimization is of the form

$$H_i(K'_{i2}(k), P_i(k|k-1), \Lambda_i(k+1), k)$$

$$\triangleq \text{tr}\{\left[\Phi_{ii,eq} - K'_{i2}\bar{H}'_{i2}\right]P_i\left[\Phi_{ii,eq} - K'_{i2}\bar{H}'_{i2}\right]^T Q_i + K'_{i2}R'^T_{i2}L'^T_{ii} + L'_{ii}R'_{i2}K'^T_{i2} + K'_{i2}R'_{i2}K'^T_{i2})\Lambda_i^T(k+1)\}$$

$$+ \boxed{\text{tr}[M(k)P(k|k-1)]} \tag{2.3-15a}$$

$$= \text{tr}\{(\Phi_{ii,eq}P_i\Phi^T_{ii,eq} - K'_{i2}\bar{H}'_{i2}P_i\Phi^T_{ii,eq} - \Phi_{ii,eq}P_i\bar{H}'^T_{i2}K'^T_{i2} + K'_{i2}\bar{H}'_{i2}P_i\bar{H}'^T_{i2}K'^T_{i2} + Q_i + K'_{i2}R'^T_{i2}L'^T_{ii} + L'_{ii}R'_{i2}K'^T_{i2} + K'_{i2}R'_{i2}K'^T_{i2})\Lambda_i^T(k+1)\}$$

$$+ \boxed{\text{tr}[M(k)P(k|k-1)]} \tag{2.3-15b}$$

where

$\Lambda_i(k+1)$ is the $n_i \times n_i$ costate matrix

and

$$Q_i \triangleq \overset{\circ}{Q}_i(k) + L'_{ii}(k)R'_{ii}(k)L'^T_{ii}(k) \tag{2.3-16}$$

$$\Phi_{ii,eq} \triangleq \Phi_{ii}(k+1,k) - L'_{ii}(k)\bar{H}'_{ii}(k) \tag{2.3-17}$$

Unlike the case for deriving the standard centralized Kalman filter using the Matrix Minimum Principle, there is no analog to the last three terms present in Eq. 2.3-15b, so final results will differ correspondingly due to the added complication.

Differentiating the Hamiltonian of Eq. 2.3-15b with respect to the appropriate variable yields the so-called "coupling equation" as

$$\underline{0} = \frac{\partial H_i}{\partial K_{i2}}(K'_{i2}, P_i(k|k-1), \Lambda_i(k+1), k)\Big|^* \tag{2.3-18a}$$

$$= \Lambda_i^*(k+1)\left[L'_{ii}R'_{i3} - \Phi_{ii,eq}P_i^*\bar{H}'^T_{i2} + K'^*_{i2}(\bar{H}'_{i2}P_i^*\bar{H}'^T_{i2} + R'_{i2})\right]$$

$$+ \Lambda_i^{*T}(k+1)\left[L'_{ii}R'_{i3} - \Phi_{ii,eq}P_i^*\bar{H}'^T_{i2} + K'^*_{i2}(\bar{H}'_{i2}P_i^*\bar{H}'^T_{i2} + R'_{i2})\right] \qquad (2.3\text{-}18b)$$

The question for the "backwards" or reverse time evolution of the costate (upon using the simplification allowed by Ref. 2) is:

$$\Lambda_i^*(k) = +\frac{\partial H_i}{\partial P}(K_i(k), P_i, \Lambda_i(k+1), k)\Big|_*$$

$$= \left[\Phi_{ii}(k+1,k) - L'^*_{ii}\bar{H}'_{i1} - K'^*_{i2}\bar{H}'_{i2}\right]^T \Lambda_i^*(k+1)\left[\Phi_{ii}(k+1,k) - L'^*_{ii}\bar{H}'_{i1} - K'^*_{i2}\bar{H}'_{i2}\right] + \boxed{M_i(k)} \qquad (2.3\text{-}19)$$

with transversality (i.e., boundary) condition

$$\Lambda_i^*(N_0) = \frac{\partial}{\partial P_i}\mathrm{tr}\left[M_i(N_0)P_i\right]\Big|_* = M_i(N_0) = M_i^T(N_0) > \underline{0} \qquad (2.3\text{-}20)$$

The following arguments proceed as in Refs. 3 and 12 (but with more justification as will be indicated parenthetically now that the boxed terms are included in Eqs. 2.3-15 and 2.3-19 according to the convention first utilized in Refs. 47, 48, 49, 15, and 35, and explained in more detail in Section 3 of Ref. 46). Note from the assumption of positive definiteness of $M_i(k)$ and the form of Eqs. 2.5-19 and 2.5-20 that the $\Lambda(k)$ that evolves backward in time from $k=N_0$ to $k=0$ is symmetric and positive definite. Without the boxed terms present in this analysis, it is more difficult (if not impossible) to rigorously establish that $\Lambda(k)$ is positive definite since it is not known for sure that the matrix quantity in brackets that pre- and postmultiplies the first term on the right side of Eq. 2.3-19 is guaranteed to be of full rank. Only because of the presence of the boxed term in Eq. 2.3-19 can it be concluded that the $\Lambda(k)$ evolving from Eq. 2.3-19 is indeed symmetric and positive definite (independent of the bracketed terms of Eq. 2.3-19 being of full rank). Thus

"$\Lambda^*_i{}^{-1}(k)$ exists for $0 \le k \le N_0$ and is symmetric."

Thus Eq. 2.3-18b can be premultiplied throughout by $\Lambda_i^{-1}(k)$ to result in

$$\underline{0} = 2\,L'_{ii}R'_{i3} - 2\Phi_{ii,eq}P_i^*\bar{H}'^T_{i2} + 2\,K'^*_{i2}(\bar{H}'_{i2}P_i^*\bar{H}'^T_{i2} + R'_{i2}) \qquad (2.3\text{-}21a)$$

or

$$K'^*_{i2}(k) = \left[\Phi_{ii}(k+1,k)P_i^*(k|k-1)\bar{H}'^T_{i2} - L'_{ii}(\bar{H}'_{i1}P_i^*(k|k-1)\bar{H}'^T_{i2} + R'_{i3}(k))\right](\bar{H}'_{i2}(k)P_i^*(k|k-1)\bar{H}'^T_{i2} + R'_{i2}(k))^{-1} \qquad (2.3\text{-}21b)$$

The above equation specifies the discrete-time optimum gain for minimum variance estimation within the class of SLU decentralized filters (having unbiased estimates). Despite a few similarities to what was obtained for continuous-time in Ref. 8, the result reported in Eq. 2.3-21b (as obtained in the original research of Refs. 15 and 35) is a new contribution. (Consistent with standard practice, superscript asterisks will be suppressed in what follows for notational convenience.)

Step 3. To obtain the appropriate Riccati equation to be solved in implementing the local estimator, differentiate the Hamiltonian of Eq. 2.3-15 with respect to $\Lambda_i(k+1)$ to return Eq. 2.3-9, the equation of evolution that $P_i(k|k-1)$ must satisfy as a dynamical constraint. Substituting the optimal gain of Eq. 2.3-21b back into Eq. 2.3-9 yields:

$$P_i(k+1|k) = \left[\Phi_{ii} - L'_{ii}\bar{H}'_{i1} - \Phi_{ii}P_i\bar{H}'^T_{i2}(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1} + L'_{ii}(\bar{H}'_{i1}P_i\bar{H}'^T_{i2} + R'_{i3})(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1}\bar{H}'_{i2}\right]P_i$$

$$\cdot\left[\Phi_{ii} - L'_{ii}\bar{H}'_{i1} - \Phi_{ii}P_i\bar{H}'^T_{i2}(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1} + L'_{ii}(\bar{H}'_{i1}P_i\bar{H}'^T_{i2} + R'_{i3})(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1}\bar{H}'_{i2}\right]^T$$

$$+ L'_{ii}R'_{i3}L'^T_{ii} - L'_{ii}R'^T_{i3}\bar{H}'^T_{i2}(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1}\bar{H}'_{i2}L'^T_{ii} - L'_{ii}(\bar{H}'_{i1}P_i\bar{H}'^T_{i2} + R'_{i3})(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1}\bar{H}'_{i2}L'^T_{ii}$$

$$+ L'_{ii}R'_{i3}(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1}\bar{H}'_{i2}P_i\Phi^T_{ii} - L'_{ii}R'_{i3}(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i3})^{-1}(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i3})L'^T_{ii}$$

$$+ \Phi_{ii}P_i\bar{H}'^T_{i2} - L'_{ii}(\bar{H}'_{i1}P_i\bar{H}'^T_{i2} + R'_{i3})\left](\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1}\bar{H}'_{i2}(\bar{H}'_{i2}P_i\bar{H}'^T_{i2} + R'_{i2})^{-1}\left[\Phi_{ii}P_i\bar{H}'^T_{i2} - L'_{ii}(\bar{H}'_{i1}P_i\bar{H}'^T_{i2} + R'_{i3})\right]^T \qquad (2.3\text{-}22a)$$

$$= \left[\Phi_{ii} - \dot{L}'_{ii}\bar{H}'_{i1}\right] P_i \left[\Phi_{ii} - \dot{L}'_{ii}\bar{H}'_{i1}\right]^T$$

$$- \left[(\Phi_{ii} - \dot{L}'_{ii}\bar{H}'_{i1}) P_i \bar{H}'^T_{i2} - \dot{L}'_{ii} R'_{i3}\right] (\bar{H}'_{i2} P_i \bar{H}'^T_{i2} + R'_{i2})^{-1} \left[(\Phi_{ii} - \dot{L}'_{ii}\bar{H}'_{i1}) P_i \bar{H}'^T_{i2} - \dot{L}'_{ii} R'_{i3}\right]^T$$

$$+ \left[\dot{Q}_i + \dot{L}'_{ii} R'_{i1} \dot{L}'^T_{ii}\right] \qquad (2.3\text{-}22b)$$

TABLE 2.3-1

SUMMARY OF DISCRETE-TIME MECHANIZATION EQUATIONS
OF SURELY LOCALLY UNBIASED (SLU) FILTERING

Order of Calculations	Mechanization Equations* for the General Case
Step 1	SVD algorithm performs the following factorization: $\underset{(q_i \times p_i)}{\Phi_i} = \underset{(q_i \times q_i)}{U_i} \begin{bmatrix} s_1 & & & 0 \\ & s_2 & & \\ & & \ddots & \\ 0 & & & s_{p_i} \\ & & 0 & \end{bmatrix}_{(q_i \times p_i)} \underset{(p_i \times p_i)}{V_i^T}$ Upon making the following assignments: $z_{i2} \triangleq V_i \begin{bmatrix} s_1 & & & 0 \\ & s_2 & & \\ & & \ddots & \\ 0 & & & s_{p_i} \end{bmatrix}$, $z_{i1}^{-1} \triangleq U_i^T$ Hence** $z_{i2}^{-1}(k)\hat{H}_i(k)z_{i1}(k) = \begin{bmatrix} I_{p_i} \\ 0 \end{bmatrix}_{q_i - p_i}^{p_i}$ for the condition $p_i < q_i$ (as confirmed/denied from the factorization).
Step 2	$\dot{L}_{ii} \longrightarrow \dot{L}_{ii}(k) z_{i1}(k) \triangleq \dot{L}'_{ii}(k)$
Step 3	$L_i \longrightarrow z_{i1}^{-1}(k) L_i(k) \triangleq L'_i(k)$
Not necessary in practice	$v_i(k) \longrightarrow z_{i2}^{-1}(k) v_i(k) \triangleq v'_i(k)$
Step 4	$\bar{H}_i(k) \longrightarrow z_{i2}^{-1}(k)\bar{H}_i(k) \triangleq \bar{H}'_i(k)$ $\qquad \hat{H}_i(k) \longrightarrow \begin{bmatrix} I_{p_i} \\ 0 \end{bmatrix}$
Step 5	$z_i(k) \longrightarrow z_{i2}^{-1}(k) z_i(k) \triangleq z'_i(k)$
Step 6	$R_i(k) \longrightarrow z_{i2}^{-1}(k) R_i(k) z_{i2}^{-T}(k) \triangleq R'_i(k)$ Positioning and assigning as: $R'_i(k) = \begin{bmatrix} \overbrace{R'_{i1}(k)}^{p_i} & \overbrace{R'_{i3}(k)}^{} \\ R'^T_{i3}(k) & R'_{i2}(k) \end{bmatrix}_{q_i - p_i}$ $\qquad R'^{(1)}_i(k) = \begin{bmatrix} R'_{i1}(k) & \\ & \end{bmatrix}$
Step 7	$P_i(k+1\|k) = \left[\Phi_{ii} - \dot{L}'_{ii}\bar{H}'_{i1}\right] P_i \left[\Phi_{ii} - \dot{L}'_{ii}\bar{H}'_{i1}\right]^T + \left[\dot{Q}_i + \dot{L}'_{ii} R'_{i1} \dot{L}'^T_{ii}\right]$ $- \left[(\Phi_{ii} - \dot{L}'_{ii}\bar{H}'_{i1}) P_i \bar{H}'^T_{i2} - \dot{L}'_{ii} R'_{i3}\right] (\bar{H}'_{i2} P_i \bar{H}'^T_{i2} + R'_{i2})^{-1} \left[(\Phi_{ii} - \dot{L}'_{ii}\bar{H}'_{i1}) P_i \bar{H}'^T_{i2} - \dot{L}'_{ii} R'_{i3}\right]^T$
Step 8	$L'_{i,m+1} = \left[\Phi_{i,m+1,m} P'_{i,m\|m-1} \bar{H}'^T_{i,m} - \dot{L}'_{i,m} R'_{i3,m}\right] \left[\bar{H}'_{i,m} P'_{i,m\|m-1} \bar{H}'^T_{i,m} + R'_{i2,m}\right]^{-1}$
Step 9	$\hat{x}_{i,m+1\|m} = \Phi_{i,m+1,m} \hat{x}_{i,m\|m-1} + \dot{L}_{i,m} \left[z_{i1,m} - \bar{H}_{i,m} \hat{x}_{i,m\|m-1}\right] + \dot{L}'_{i,m} \left[z'_{i,m} - \bar{H}'_{i,m} \hat{x}_{i,m\|m-1}\right]$

*This discrete-time formulation (derived in Ref. 15 and previously unavailable as acknowledged on p. 92 of Ref. 12) was necessary before computational burden of Table 2.4-1 could be quantified. **This factorization was theoretically guaranteed to be possible in Refs. 8-12, but accomplishment of this task in the general case by SVD first identified here and in Ref. 15. Validated SVD software is available for this task (Refs. 38 and 39).

For insight, it is mentioned in passing that six distinct steps constituting rearrangements (available upon request and provided in Ref. 15) were required in going from Eq. 2.3-22a to the concise expression of Eq. 2.3-22b. Bridging this gap was one of the most gruelling mathematical challenges encountered in this investigation even if it consisted of only algebraic manipulations of long unwieldy expressions. This may perhaps explain why from 1973 (Ref. 8) to 1979 (Ref. 12), the discrete-time case—useful for digital computer implementation—did not emerge even though the discrete-time case was acknowledged (p. 93 of Ref. 12) as necessary before operation counts can be compiled (as provided in Section 2.4). The 9 simple steps for mechanizing an SLU filter algorithm are summarized in Table 2.3-1.

2.4 Considerations of Computational Burden of SLU Filtering

The SLU computer memory allotment required may be obtained in two steps. By first reading from Table V, p. 750 of Ref. 36, the memory required for the standard Kalman filter is

$$5n_i^2 + 3n_i + 2n_iq_i + q_i^2 + q_i + 1 \tag{2.4-1}$$

Second, accounting for the additional occurrence of such characteristic SLU terms as L_{ii}, Z_{il}, Z_{i2}, Z_{il}^{-1}, and Z_{il}^{-1}, the corresponding appropriate dimensions are then added to Eq. 2.4-1 to yield

$$5n_i^2+3n_i+2n_iq_i+3q_i^2+q_i+n_ip_i+3p_i^2+1 \tag{2.4-2}$$

The number of adds, multiplies, and logic time requirements (as compiled in Ref. 15 using the techniques of Refs. 36 and 37) are summarized in Table 2.4-1 for an SLU filter. This information can be used to establish the filter cycle time required for processing a measurement once the add, multiply, and logic times are provided for the intended target machine.

TABLE 2.4-1

SUMMARY OF DISCRETE-TIME MECHANIZATION EQUATIONS
OF SURELY LOCALLY UNBIASED (SLU) FILTERING

Order of Calculating	Total Number of ADDS	Total Number of Multiplies	(Estimated) Logic Time
Step 1	See SVD in Refs. 39 and 84	See SVD in Refs. 39 and 84	See SVD in Refs. 39 and 84
Step 2	$n_ip_i^2-n_ip_i$	$n_ip_i^2$	$10+6n_ip_i^2+21n_ip_i+16n_i$
Step 3	$n_ip_i^2-n_ip_i$	$n_ip_i^2$	$10+6n_ip_i^2+21n_ip_i+13p_i$
Not necessary in practice	$q_i^2-q_i$	q_i^2	$10+6q_i^2+37q_i$
Step 4	$n_iq_i^2-n_iq_i$	$n_iq_i^2$	$10+6n_iq_i^2+21n_iq_i^2+16q_i$
Step 5	$q_i^2-q_i$	q_i^2	$10+6q_i^2+37q_i$
Step 6	$2q_i^3-2q_i^2$	$2q_i^3$	$20+13q_i^3+43q_i^2+22q_i$
Step 7	$2n_i^3-n_i^2(3q_i-p_i)-n_i^2+12n_i(q_i-p_i)^2+n_iq_ip_i-3n_i(q_i-p_i)-n_ip_i+(q_i-p_i)^3$	$2n_i^3+n_i^2(3q_i-p_i)+n_i\cdot(q_i-p_i)(3q_i-p_i)+n_ip_i^2+(q_i-p_i)^3$	$282+12n_i^3+129n_i^2+189n_i+6n_i^2(3q_i-p_i)+12n_i(q_i-p_i)^2+6n_i(q_i-p_i)p_i+6n_ip_i^2+69n_i(q_i-p_i)+21n_ip_i+7.5(q_i-p_i)^4+165.5(q_i-p_i)^3+189(q_i-p_i)+MAX(0.5(q_i-p_i)^2+2.5(q_i-p_i)+61+DIV(2(q_i-p_i)^2+(q_i-p_i)) + 41(q_i-p_i)^5$
Step 8	$2n_i^2(q_i-p_i)+2n_i\cdot(q_i-p_i)(q_i-3n_i\cdot(q_i-p_i)+(q_i-p_i)^3$	$2n_i^2(q_i-p_i)+2n_i(q_i-p_i)\cdot q_i+(q_i-p_i)^3$	$151+12n_i^2(q_i-p_i)+12n_i(q_i-p_i)q_i+69n_i(q_i-p_i)+64n_i+41(q_i-p_i)^3+139.5(q_i-p_i)^2+26(q_i-p_i)q_i+189(q_i-p_i)+16p_i+7.5(q_i-p_i)^4+\left[3+25(q_i-p_i)+0.5(q_i-p_i)^2\right]MUL+\left[(q_i-p_i)+2(q_i-p_i)^2\right]$DIV
Step 9	$n_i^2-n_i+2n_iq_i$	$n_i^2+2n_iq_i$	$150+6n_i^2+121n_i+12n_iq_i+42q_i+$MUL

*Compiled based on precedent conventions, and guidelines set forth in Refs. 36 and 37 for such calculations where in the above SLU specialization:

n_i dimension of the subsystems local SLU filter (defined in Eq. 1.2-11. p_i rank of component of observation matrix H_i (defined in Eq. 1.2-21.

q_i dimension of subsystems local measurements (defined in Eq. 1.2-2a). MUL unit logic time required for a multiply by an element retrieved from core memory (by indirect addressing) rather than by an element from the Arithmetic Logic Unit. (ALU). DIV unit logic time required for a division by an element retrieved from core memory (by indirect addressing) rather than by an element from the ALU.

• If $p_i = q_i$, no SVD is required and mechanization should revert to standard Kalman filter.

As observed in Refs. 8-12, the [illegible] filter treats the interaction input to each local subsystem as if it were just [illegible] mean, Gaussian, white noise disturbance, but of the appropriately tailored covariance level. Therefore, the unmodified SLU filter treats all interactions, deterministic or otherwise, as a purely stochastic uncorrelated interference effect. This apparent de-emphasis of the significant subsystem interactions may be initially unsettling for some anticipated applications since all information transfer between subsystems can only be via these interactions within the strict SLU filter structure. However, there are existing refinements of the SLU filtering formulation (Refs. 11 and 12) that appear to take better advantage of the correlated information intrinsically contained in the subsystem interactions by

- exchange of inter-subsystem interactions via a communication channel,
- exchange of subsystem state estimates via a communication channel,
- modeling possible noise in the communication channel if necessary.

An overview of the variety of identified SLU variations already developed in Refs. 11 and 12 are summarized here in Tables 2.4-2 and 2.4-3. In particular, use of a form of

TABLE 2.4-2

ALTERNATIVE VARIATIONS IN SURELY LOCALLY UNBIASED (SLU) FILTER MECHANIZATIONS: REF. 12

SLU Filter Variations	Information Exchange	Order of Local Filter	Communications Channel Noise Modeled in Design	Decentralized or Centralized Design
F_{CL}	None	n_i	--	Decentralized
F_{SLU}	None	n_i	--	Decentralized
F_{GD}	None	$\sum_{i=1}^{N} n_i$	--	Decentralized
F_{LGD}	None	$\sum_{i=1}^{N} n_i$	--	Decentralized
F_{ESLU}*	Interactions	$\bar{n}_i (>n_i)$	Yes	Decentralized
F^0_{ESLU}*	Interactions	$\bar{n}_i (>n_i)$	No	Decentralized
F_{RGD}	State Estimates	n_i	Yes	Centralized
F_{POSLU}	State Estimates	n_i	Yes	Centralized
F^0_{POSLU}	State Estimates	n_i	No	Centralized
F_{ROM}*	None	$n_i + n_{ROM,i}$	--	Decentralized

*Prime SLU candidates for the JTIDS RelNav application in the investigation of Ref. 15.

approximate "aggregation" in filter design is suggested in Chapter 5 of Ref. 12 as a compromise to obtain a so-called Reduced-Order Model (ROM) filter that has better performance than the SLU while also offering more realistically dimensioned computations than commonly associated with a centralized Kalman filter. Unlike what the specialized technical term "aggregation" may at first glance suggest, the "method of aggregation" is a well developed technique (Refs. 55, 56, and 57) for realistically representing the effects of interactions amongst the constituents of a large-scale system by using just a reduced-order smaller scale system. A pedagogically lucid and self-consistent defivation and explanation of the ROM filter is provided on pp. 5-2 to 5-13 of Ref. 15 and its potential for the JTIDS RelNav application is summarized in Ref. 35.

3 THE SEQUENTIALLY PARTITIONED ALGORITHM (SPA) FOR DECENTRALIZED FILTERING

3.1 Derivation of the Decentralized SPA Approach to Decentralized Filtering

An interesting approach to decentralized filtering first described in a 1971 British Ph.D. thesis (Ref. 14) by M. Shah has been relatively recently reexamined in Ref. 13 for

TABLE 2.4-3

TERMINOLOGY OF SLU ALTERNATIVE VARIATIONS

Symbolic Designation	Filter Name Designations (from Ref. 12)
F_{CL}	Completely Localized Mechanization (of otherwise Centralized)*
F_{SLU}	Surely Locally Unbiased
F_{GD}	Global Dynamics
F_{LGD}	Localized Global Dynamics
F_{ESLU}	Expanded Surely Locally Unbiased*
F^0_{ESLU}	Same as above, without modeling communications noise.
F_{RGD}	Reduced-Order Global Dynamics
F_{PDSLU}	Partially Decentralized Surely Locally Unbiased
F^0_{PDSLU}	Same as above, without modeling communications noise.
F_{ROM}	Reduced Order Model

*Treats local Model as if it were Global Model.

the discrete-time case of only two subsystems.

Remark 3.1-1: A few typos occur in the derivation of the SPA filter appearing in Ref. 13 (pp. 534-7). Notably, the measurement equation appearing in Eq. 11.3.1 involving $y_i(k+1)$ should have a time index of k+1 rather than k for x_i. Similarly, the time index of the measurement noise v_i appearing in Eq. 11.3.7 should be k+1 instead of k.

The mechanization equations for this decentralized filtering approach are now derived for an arbitrary number N of decentralized subsystems as expected to be encountered in the most general application. Returning to Eq. 1.2-20c

$$x_i(k+1)=\bar{\Phi}_{ii}(k+1,k)x_i(k)+\left[\sum_{\substack{j=1\\ j\neq i}}^{N}\bar{\Phi}_{ij}(k+1,k)x_j(k)\right]+w_i(k) \tag{3.1-1}$$

and Eq. 1.2-2, augmented by Eq. 1.2-4, as

$$z_i(k)=\bar{H}_i(k)x_i(k)+\hat{H}_i(k)L_i(k)\underline{x}(k)+v_i(k) \tag{3.1-2a}$$

$$=\bar{H}_i(k)x_i(k)+\hat{H}_i(k)\sum_{\substack{j=1\\ j\neq i}}^{N}L_{ij}(k)x_j(k)+v_i(k) \tag{3.1-2b}$$

(for i=1,2,...,N). Consistent with the definition of $e_i(k+1|k)$ in Eq. 2.2-2, the estimation error is

$$e_i(k|k)\triangleq x_i(k)-\hat{x}_i(k|k) \tag{3.1-3}$$

where

$$\hat{x}_i(k|k)=E(x_i(k)|s(k)) \tag{3.1-4}$$

While information patterns can be related to the associated underlying expanding sub-sigma algebras (of a probability space or triple) with respect to which conditional expectations are taken (as Radon-Nikodym derivatives), these concepts do not appear to be necessary here and so are avoided to expedite the presentation. By adding and subtracting the same terms simultaneously, Eq. 3.1-1 for the i^{th} subsystem may be rewritten as

$$x_i(k+1)=\Phi_{ii}(k+1,k)x_i(k)+\left[\sum_{\substack{j=1\\ j\neq i}}^{N}\bar{\Phi}_{ij}(k+1,k)\hat{x}_j(k)\right]+w_i^*(k) \tag{3.1-5}$$

while the measurement equation for the i^{th} subsystem is represented as

$$z_{ii}(k)=\bar{H}_i(k)x_i(k)+\hat{H}_i(k)\sum_{\substack{j=1\\ j\neq i}}^{N}\Phi_{ij}(k)\hat{x}_j(k|k-1)+v_i^*(k) \tag{3.1-6}$$

where in the above

$$w_i^*(k)\triangleq w_i(k)+\sum_{\substack{j=1\\ j\neq i}}^{N}\bar{\Phi}_{ij}(k+1,k)e_j(k|k) \tag{3.1-7}$$

$$v_i^*(k)\triangleq v_i(k)+\sum_{\substack{j=1\\ j\neq i}}^{N}L_{ij}(k)\,e_j(k|k-1) \tag{3.1-8}$$

Under the following assumptions at time-step k.

Assumption 3.1-1:

$$\hat{x}_j(k|k-1) \text{ is known (deterministically) for } j\neq i,\ j=1,2,\ldots,N \tag{3.1-9}$$

Assumption 3.1-2:

$$e_j(k|k),\ e_j(k|k-1) \text{ are Gaussian and white for } j\neq i,\ j=1,2,\ldots,N \tag{3.1-10}$$

(just as residuals should ideally be white when filter models and truth models are appropriately matched), therefore (as a consequence of Assumption 3.1-2):

Assumption 3.1-3:

$$w_i^*(k) \text{ and } v_i^*(k) \text{ can be treated as Gaussian white process and measurement noises} \tag{3.1-11}$$

and the standard Kalman filter algorithm can be applied to each subsystem (p. 536, Ref. 13) with the following appropriately modified covariances:

$$Q_{ii}^*(k)\triangleq E\left[w_i^*(k)w_i^{*T}(k)\right]=Q_i(k)+\sum_{\substack{j=1\\ j\neq i}}^{N}\bar{\Phi}_{ij}(k+1,k)P_j(k|k)\bar{\Phi}_{ij}^T(k+1,k) \tag{3.1-12}$$

$$R_{ii}^*(k)\triangleq E\left[v_i^*(k)v_i^{*T}(k)\right]=R_{ii}(k)+\hat{H}_i(k)\left[\sum_{\substack{j=1\\ j\neq i}}^{N}L_{ij}(k)P_j(k|k-1)L_{ij}^T(k)\right]\hat{H}_i^T(k) \tag{3.1-13}$$

The result of applying the standard filtering formulation to each subsystem and utilization of the above assumptions is

$$\hat{x}_i(k+1|k)=\bar{\Phi}_{ii}(k+1,k)\hat{x}_i(k|k)+\left[\sum_{\substack{j=1\\ j\neq i}}^{N}\bar{\Phi}_{ij}(k+1,k)\hat{x}_j(k|k)\right] \tag{3.1-14a}$$

$$=\bar{\Phi}_{ii}(k+1,k)\hat{x}_i(k|k)+\left[\sum_{\substack{j=1\\ j\neq i}}^{N}\hat{x}_j(k+1|k)\right] \tag{3.1-14b}$$

$$\hat{x}_i(k+1|k+1)=\hat{x}_i(k+1|k)+R_i(k+1)\left[z_{ii}(k+1)-\bar{H}_i(k+1)\hat{x}_i(k+1|k)-\hat{H}_i(k+1)\sum_{\substack{j=1\\ j\neq i}}^{N}L_{ij}(k+1)\hat{x}_j(k+1|k)\right] \tag{3.1-15}$$

The filter gain matrix appearing in Eq. 3.1-15 can be obtained from the following

$$R_i(k+1)=P_{ii}(k+1|k)\bar{H}_i^T(k+1)\left[\bar{H}_i(k+1)P_{ii}(k+1|k)\bar{H}_i^T(k+1)+R_{ii}^*(k+1)\right]^{-1} \tag{3.1-16}$$

where the extrapolate step consists of

$$P_{ii}(k+1|k)=\Phi_{ii}(k+1,k)P_{ii}^*(k|k)\Phi_{ii}^T(k+1,k)+Q_{ii}^*(k) \tag{3.1-17}$$

and the update step consists of the standard [illegible]

$$P_{ii}(k+1|k+1)=\left[I-\bar{K}_i(k+1)\bar{H}_i(k+1)\right]P_{ii}(k+1|k)\left[I-\bar{K}_i(k+1)\bar{H}_i(k+1)\right]^T$$

$$+\bar{K}_i(k+1)R_{ii}^*(k+1)\bar{K}_i^T(k+1) \qquad (3.1\text{-}18)$$

It is noted that this filter is suboptimal as perceived from a centralized point-of-view since it does not account for possible off diagonal terms in the covariance matrix. However, the above so-called SPA filter is computationally very attractive to implement, since it requires a substantially smaller number of elementary multiplication and addition operations and less computer storage than a globally optimum Kalman filter.

Further simplifications of the SPA filter were possible for the JTIDS RelNav application considered in Refs. 15 and 35 since there is no cross-coupling in the linearized system error models as represented by

$$\Phi_{ij}(k+1,k) = 0 \text{ for } i \neq j \qquad (3.1\text{-}19)$$

Hence, Eqs. 3.1-5, 3.1-7, 3.1-12, 3.1-14a, and 3.1-15, respectively, degenerate to

$$x_i(k+1)=\Phi_{ii}(k+1,k)x_i(k)+w_i^*(k) \qquad (3.1\text{-}20)$$

$$w_i^*(k)\triangleq w_i(k) \qquad (3.1\text{-}21)$$

$$Q_{ii}^*(k)\triangleq E\left[w_i^*(k)w_i^{*T}(k)\right]=E\left[w_i(k)w_i^T(k)\right]=Q_{ii}(k) \qquad (3.1\text{-}22)$$

$$\hat{x}_i(k+1|k)=\Phi_{ii}(k+1,k)\hat{x}_i(k|k) \qquad (3.1\text{-}23)$$

$$\hat{x}_i(k+1|k+1)=\hat{x}_i(k+1|k)+\bar{K}_i\left[z_{ii}(k+1)-\bar{H}_i(k+1)\hat{x}_i(k+1|k)-\hat{H}_i(k+1)\sum_{\substack{j=1\\ j\neq i}}^{N}L_{ij}(k+1)\Phi_{ij}(k+1,k)\hat{x}_j(k|k)\right] \qquad (3.1\text{-}24)$$

As seen from Eqs. 3.1-12 and 3.1-13, respectively, the SPA filter accounts for interconnection effects by utilising fictitious components of process and measurement noises with appropriately enlarged covariances (not unlike aspects of the SLU approach already noted). Thus the entire discrete-time SPA implementation for the condition of Eq. 3.1-19 is described by Eqs. 3.1-11, 3.1-22, 3.1-23, 3.1-24, 3.1-16, 3.1-17, and 3.1-18, as summarized in Table 3.1-1. Mechanization of these six equations constitutes the entire computational burden of SPA and, unlike SLU, involves no SVD factorization or additional transformations. The computational burden of SPA filtering is quantified in Section 3.2.

TABLE 3.1-1

SUMMARY OF DISCRETE-TIME MECHANIZATION EQUATIONS OF SEQUENTIALLY PARTITIONED ALGORITHM (SPA)

Order of Calculations	Mechanization Equations for the General Case*	Mechanization Equations Specialized for $L_{ij}(k+1,k)=0$ (as in Refs. 15 and 35)
Step 1	$\hat{x}_i(k+1\|k)=\Phi_{ii}(k+1,k)\hat{x}_i(k\|k)$	Same as general case and Standard Kalman Filter
Step 2	$w_i^*(k)=w_i(k)+\sum_{\substack{j=1\\ j\neq i}}^{N}L_{ij}(k+1,k)w_j(k)$ $Q_{ii}^*(k)=E\left[w_i^*(k)w_i^{*T}(k)\right]=Q_{ii}(k)+\sum_{\substack{j=1\\ j\neq i}}^{N}L_{ij}(k+1,k)Q_{jj}(k)L_{ij}^T(k+1,k)$	$w_i^*(k)\triangleq w_i(k)$ $Q_{ii}^*(k)\triangleq E\left[w_i^*(k)w_i^{*T}(k)\right]=E\left[w_i(k)w_i^T(k)\right]=Q_{ii}(k)$
Step 3	$P_{ii}(k+1\|k)=\Phi_{ii}(k+1,k)P_{ii}(k\|k)\Phi_{ii}^T(k+1,k)+Q_{ii}^*(k)$	Same as general case and Standard Kalman Filter
Step 4	$\bar{K}_i(k+1)=P_{ii}(k+1\|k)\bar{H}_i^T(k+1)\left[\bar{H}_i(k+1)P_{ii}(k+1\|k)\bar{H}_i^T(k+1)+R_{ii}^*(k+1)\right]^{-1}$	Same as general case therein lies the distinction between SPA and Standard Kalman Filter!
Step 5	$\hat{x}_i(k+1\|k+1)=\hat{x}_i(k+1\|k)+\bar{K}_i(k+1)\left[z_{ii}(k+1)-\bar{H}_i(k+1)\hat{x}_i(k+1\|k)\right]$	Same as general case and Standard Kalman Filter
Step 6	$P_{ii}(k+1\|k+1)=P_{ii}(k+1\|k)-\bar{K}_i(k+1)\bar{H}_i(k+1)P_{ii}(k+1\|k)$	$P_{ii}(k+1\|k+1)=\left[I-\bar{K}_i(k+1)\bar{H}_i(k+1)\right]P_{ii}(k+1\|k)$
Step 7	$P_{ii}(k+1\|k+1)=\left[I-\bar{K}_i(k+1)\bar{H}_i(k+1)\right]P_{ii}(k+1\|k)\left[I-\bar{K}_i(k+1)\bar{H}_i(k+1)\right]^T+\bar{K}_i(k+1)R_{ii}^*(k+1)\bar{K}_i^T(k+1)$	Same as general case and Joseph's form for Standard Kalman Filter

* Only the case of two subsystems is considered in Ref. 13, while generalization to N is presented here and in Ref. 14

In general, for the SPA filtering approach

$$E\left[x_i(k+1)\,|\,z_i(k)\right] \neq E\left[x_i(k+1)\,|\,Z(k)\right] \tag{3.1-25}$$

so that taking total expectations throughout Eq. 3.1-25 yields:

$$E\{\hat{x}_i(k+1|k)\}=E\{E\left[x_i(k+1)\,|\,z_i(k)\right]\}\neq E\{E\left[x_i(k+1)\,|\,Z(k)\right]\}=E\{x_i(k+1)\} \tag{3.1-26}$$

The above indicates that the SPA decentralized filter, along with most other practical filters, is a biased estimator (Ref. 34) (e.g., even in linear Gaussian applications, the optimum centralized Kalman filter has practicality constraints that frequently limit the order of the filter implementation that can be computationally accommodated to be much lower than the "truth" model, thus yielding a biased filter when implemented). As long as the bias is either evaluated, compensated, or shown to be tolerably small, its presence should not interfere with the intended purpose of any filter. Indeed, it is the SPA filter that appears to be most useful for the JTIDS RelNav application as discussed in Refs. 15 and 35. Results of subsequent simulations are reported in Ref. 60.

3.2 Computational Burden of SPA Filtering

The SPA computer memory allotment required may also be obtained by the same two step procedure discussed in Section 2.4. The SPA filter uses additional intermediate scratch calculations of dimension $(n_i \times n_i)$, $(q_i \times q_i)$, and $(q_i \times n_i)$ beyond those encountered in a conventional Kalman filter so the total memory requirement is

$$6n_i^2 + 3n_i + 3n_iq_i + 2q_i^2 + q_i + 1 \tag{3.2-1}$$

The number of adds, multiplies, and logic time requirements are summarized in Table 3.2-1 for an SPA filter. These results for SPA filtering were obtained by applying the standard methodology for operations counts from Refs. 36 and 37 to the algorithms summarized in Table 3.1-1 in a manner analogous to what was done in Section 2.4. Both the SPA (with the special structure of Eq. 3.1-19) and the general SLU filter implementations are analytically demonstrated to be stable in Section 4.

4. STABILITY OF DISCRETE-TIME DECENTRALIZED FILTERS OF THE SLU AND SPA CLASSES

4.1 Stability Overview

The proof of stability for the discrete-time formulations of SLU and SPA proceeds in a manner analogous to the continuous-time proof provided in Ref. 10 by also utilizing the stability framework of Ref. 16. The crux of the proof involves demonstrating that

$$V[\hat{x}(k|k-1),k] = \hat{x}^T(k|k-1)\,P^{-1}(k|k)\,\hat{x}(k|k-1) \tag{4.1-1}$$

is a valid Lyapunov function by demonstrating that it is positive definite and that the first variation along the trajectory of

$$\hat{x}(k+1|1) = [\Phi(k+1,k) - \Phi(k+1,k)K(k)H(k)]\,\hat{x}(k|k-1) \tag{4.1-2}$$

is negative definite. The inner product matrix in Eq. 4.1-1 being the inverse of the matrix that evolves from the matrix Riccati equations (encountered in both SPA and SLU filtering) which can be demonstrated to be uniformly bounded above and below under the condition of uniform complete observability of the subsystem (without also requiring uniform complete controllability or even controllability due to weakened hypothesis provided by Ref. 16).

However, recent results (Ref. 17, Appendix C of Ref. 18, and p. 244 of Ref. 19) indicated a minor error in the upper and lower bounds appearing in Refs. 40 and 41 (as utilized in Refs. 16 and 10). This error is corrected in Appendix A and shown to not adversely affect the stability conclusions for SPA and SLU as long as all observation matrices $H_i(k_i)$ are of full rank (a new restriction) as well as $R_i^{-1}(k)$ being bounded and $P_i(0)>0$. By this approach, only asymptotic stability is established for SLU and SPA in contradistinction to exponential asymptotic stability.

4.2 Precedent of an Analytic Framework Enabling a Demonstration of Stability for Several Decentralized Filtering Mechanizations

A discrete-time filter of the following form

$$\hat{x}(k+1|k)=\Phi(k+1,k)\hat{x}(k|k)=\Phi(k+1,k)\hat{x}(k|k-1)+\Phi(k+1,k)K(k)\left[z(k)-H(k)\hat{x}(k|k-1)\right] \tag{4.2-1a}$$

$$=\left[\Phi(k+1,k)-\Phi(k+1,k)K(k)H(k)\right]\hat{x}(k|k-1)+\Phi(k+1,k)K(k)z(k) \tag{4.2-1b}$$

TABLE 3.2-1

SUMMARY OF MECHANIZATION EQUATIONS AND QUANTIFICATION OF COMPUTATIONAL BURDEN* OF SEQUENTIALLY PARTITIONED ALGORITHM (SPA) FILTERING

Order of Calculations	Required SPA Equation Being Implemented	Total Number of Adds	Total Number of Multiplies	Logic Time (Estimated)
Step 1	Eq. 3.1-23 (same as standard Kalman filter)	$n_i^2 - n_i$	n_i^2	$10+6n_i^2+37n_i$
Step 2	Eq. 3.1-12	$(2n_i^3-n_i^2)N$	$(2n_i^3)N$	$(20+12n_i^3+42n_i^2+16n_i)N$ $+27+5n_i+MUL$
	Eq. 2.1-22 (same as standard KF for JTIDS RelNav Application)	none	none	none
Step 3	Eq. 3.1-17 (same as standard KF)	$4n_i^3-3n_i^2$	$4n_i^3$	$24n_i^3+89n_i^2+64n_i+67+MUL$
Step 4	Eq. 3.1-13	$q_i^3+(n_i^2q_i+n_iq_i^2-n_iq_i-q_i^2)N$	$q_i^3+(n_i^2q_i+n_iq_i^2)N$	$(20+6n_i^2q_i+6n_iq_i^2+21n_iq_i+21q_i^2$ $+16n_i+16q_i)N+37+6q_i^3+21q_i^2+21q_i$ $+MUL$
Step 5	Eq. 3.1-16 (same as standard KF)	$n_i^2+2n_iq_i^2-2n_iq_i+q_i^3$	$n_i^2q_i+2n_iq_i^2$ $+q_i^2$	$67+6n_i^2q_i+42n_iq_i+32n_i+12n_iq_i^2+$ $7.5q_i^3+41q_i^3+165n_i^2+108q_i+(q_i+$ $2q_i^2)DIV+(0.5q_i^2+2.5q_i+1)MUL$
Step 6	Eq. 3.1-24 (for JTIDS RelNav, N = 1, Refs. 35 and 15)	$2n_iq_i+q_i^2+(n_i^2q_i-n_iq_i+n_i^2-n_i)N$	$q_i+2n_iq_i+(n_i^2+n_i^2q_i)N$	$84+12q_i^2+12n_iq_i+42n_i+116q_i+3$ $MUL+(20+6n_i^2q_i+6n_i^2+21n_iq_i+53n_i)N$
Step 7	Eq. 3.1-18 (same as standard KF)	$2n_i^3-n_i^2$	$2n_i^3$	$47+12n_i^3+47n_i^2+32n_i+MUL$

*Compiled based on precedent, convention, and guidelines set forth in Refs. 36, 37, and 15 for such calculations, where in the above SPA specialization:

n_i ≙ dimension of the subsystem's local SPA filter (defined in Eq. 1.2-1)

q_i ≙ dimension of subsystem's local measurements (defined in Eq. 1.2-2a)

MUL ≙ unit logic time required for a multiply by an element retrieved from core memory (by indirect addressing) rather than by an element from the Arithmetic Logic Unit (ALU)

DIV ≙ unit logic time required for a division by an element retrieved from core memory (by indirect addressing) rather than by an element from the ALU.

N ≙ number of individual subsystems contributing covariance information.

---

with

$$K(k) = P(k|k-1)H^T(k)[H(k)P(k|k-1)H^T(k)+R(k)]^{-1} \quad (4.2-2)$$

$$P(k|k-1) = \Phi(k,k-1)P(k-1|k-1)\Phi^T(k,k-1)+Q(k) \quad (4.2-3)$$

$$P(k|k) = [I-K(k)H(k)]P(k|k-1) \quad (4.2-4a)$$

$$= [I-K(k)H(k)]P(k|k-1)[I-K(k)H(k)]^T+K(k)R(k)K^T(k) \quad (4.2-4b)$$

$$P(k_0) = P_0 \quad (4.2-5)$$

(for convenience, Eqs. 4.2-2, 4.2-3, and 4.2-4b may be rewritten as:

$$P(k+1|k)=\Phi P(k|k-1)\Phi^T-\Phi P(k|k-1)H^T\left[HP(k|k-1)H^T+R\right]^{-1}HP(k|k-1)\Phi^T+Q \tag{4.2-6a}$$

$$=\Phi\left[H^TR^{-1}H+P^{-1}(k|k-1)\right]^{-1}\Phi^T+Q \tag{4.2-6b}$$

as will later be utilized) is demonstrated to be exponentially asymptotically stable (p. 222 and pp. 228-9 of Ref. 58) if the following three conditions are each satisfied:

1. $\Phi(k+1,k)$, $H(k)$, $Q(k)$, $R(k)$, and $R^{-1}(k)$ are bounded; (4.2-7)

2. $[\Phi(k+1,k),H(k)]$ is uniformly completely observable (pp. 313-4 of Ref. 18), that is, there exist two positive scalars α_1 and α_2 such that

$$\alpha_1 I \le \sum_{j=\ell-\eta}^{\ell-1} \Phi^T(j,k)H^T(j)R^{-1}(j)H(j)\Phi(j,k) \le \alpha_2 I \tag{4.2-8}$$

for some

$$\ell \ge 0 \tag{4.2-9}$$

$$\eta \le \ell - 1 \tag{4.2-10}$$

3. $[\Phi(k+1,k),D(k)]$ is uniformly completely controllable for any $D(k)$ such that

$$D(k)D^T(k) = Q(k) \tag{4.2-11}$$

that is, there exist two positive scalars α_3 and α_4 such that

$$\alpha_3 I \le \sum_{j=\ell-\eta}^{\ell-1} \Phi(k,j+1)Q(j)\Phi^T(k,j+1) \le \alpha_4 I \tag{4.2-12}$$

for some ℓ and η as in Eqs. 4.2-8 and 4.2-9, respectively.

As mentioned on p. 222 of Ref. 58, sufficient conditions for asymptotic stability, but not necessarily exponential stability are known to be 1. and 2. above, and 3. relaxed to Eq. A.1-1 of Appendix A.

For a discrete-time system of the form

$$y(k+1) = f(y(k),k) \tag{4.2-13}$$

where $y(k)$ is the state at time-step k, if there exists a so-called scalar Lyapunov function $V[y(k),k]$ such that

a. $V(0,k) \equiv 0$ for all k (4.2-14)

b. $0<\gamma_1(||y(k)||) \le V[y(k),k] \le \gamma_2(||y(k)||)$ for all $k \ge k'$ (4.2-15)

and

$$\gamma_1(0) = \gamma_2(0) = 0 \tag{4.2-16}$$

with

$$\gamma_1(\,||y||\,) \to \infty \quad \text{as} \quad ||y|| \to \infty \tag{4.2-17}$$

c. $V[y,k]$ is continuous in y (4.2-18)

d. $V[y,k] \to \infty$ as $y \to \infty$ (4.2-19)

e. $\Delta V[y,k] \triangleq$ rate of increase of $V[\ldots]$ along the motion of a trajectory or path of Eq. 4.2-13, starting from y at time-step k

$$\triangleq \{V[y(k+1),k+1] - V[y(k),k]\}\frac{1}{\text{(step size)}} < 0 \tag{4.2-20}$$

or, equivalently,

$$V[y(k+1),k+1]-V[y(k),k] \le -\gamma_3(||y(k)||) < 0 \tag{4.2-21}$$

then the system described by the time evolution Eq. 4.2-13 is asymptotically stable (in the large) (p. 396, Corolary 1.2* of Ref. 59, p. 240 of Ref. 51). The proof of asymptotic stability (Appendix C of Ref. 18) (pp. 240-3 of Ref. 51) for the filter of Eq. 4.2-1 consists of utilizing a discrete-time Lyapunov function of the form of Eq. 4.1-1

(Eq. 32 of Ref. 16) as a standard way to demonstrate stability of the autonomous time-varying system of the form of Eq. 4.1-2 (Eq. 31 of Ref. 16) (where Eq. 4.1-2 is observed to be the homogeneous "undriven" portion of the filter of Eq. 4.2-1). While adherence of Eq. 4.1-1 to conditions a., c., and d. is immediate, it requires more work to establish that conditions b. and e. are satisfied as a consequence of satisfying the conditions 1, 2, and 3' of the hypotheses as Eqs. 4.2-7, 4.2-8, and A.1-1. The following bounds (obtained by an extremely tedious and careful analysis of McGarty as Eqs. C.88 and C.56 of Ref. 18) are correctly validated

$$0 < A \leq P(k|k) \leq B \tag{4.2-22}$$

where

$$A \triangleq \left[\sum_{j=k-\eta}^{k-1} \Phi(k,j+1)Q(j)\Phi^T(k,j+1)\right]^{-1} + \frac{\eta^2\alpha_2\alpha_4}{\alpha_1\alpha_3} \sum_{j=k-\eta}^{k} \Phi^T(j,k)H^T(j)R^{-1}(j)H(j)\Phi(j,k) \tag{4.2-23}$$

$$B \triangleq \left[\sum_{j=k-\eta}^{k} \Phi^T(j,k)H^T(j)R^{-1}(j)H(j)\Phi(j,k)\right]^{-1} + \frac{\eta^2\alpha_2\alpha_4}{\alpha_1\alpha_3} \sum_{j=k-\eta}^{k-1} \Phi(k,j+1)Q(j)\Phi^T(k,j+1) \tag{4.2-24}$$

and α_1, α_2, α_3, α_4, and η are defined in Eqs. 4.2-8, 4.2-9, and 4.2-12. From Eq. 4.2-22, inverses can be taken throughout to yield

$$0 < B^{-1} \leq P^{-1}(k|k) \leq A^{-1} \tag{4.2-25}$$

and finally pre- and postmultiplying the above by $\hat{x}^T(k|k-1)$ and $\hat{x}(k|k-1)$, respectively, yields

$$\begin{aligned} 0 < \hat{x}^T(k|k-1)B^{-1}\hat{x}(k|k-1) &\leq \hat{x}^T(k|k-1)P^{-1}(k|k-1)\hat{x}(k|k-1) \\ &\leq \hat{x}(k|k-1)A^{-1}\hat{x}(k|k-1) \end{aligned} \tag{4.2-26}$$

Thus by definition of the appropriate Lyapunov function to be used in this situation as Eq. 4.1-1, Eq. 4.2-26 can be alternatively recognized as

$$0 < \|\hat{x}(k|k-1)\|^2_{B^{-1}(k)} \leq V\left[\hat{x}(k|k-1),k\right] \leq \|\hat{x}(k|k-1)\|^2_{A^{-1}(k)} \tag{4.2-27}$$

which certifies all three of Eqs. 4.2-15, 4.2-16, and 4.2-17 comprising condition b. above.

By a procedure related to a dual optimal control problem (where the cost function is reminiscent of the exponent in an associated probability density function, the following condition holds (Eq. C.136 of Ref. 18):

$$V\left[\hat{x}(k|k-1),k\right] - V\left[\hat{x}(k-\eta|k-\eta-1),k-\eta\right] \leq -\beta_3^2\beta_5^{-1}\beta_6\|\hat{x}(k|k-1)\|^2 \tag{4.2-28}$$

where β_3, β_5, and β_6 are positive constants defined on pp. 374-7 of Ref. 18.

Now $\eta = 1$ in Eq. 4.2-28 demonstrates that the condition e. (Eq. 4.2-21) is satisfied for the Lyapunov function postulated in Eq. 4.1-1. All five conditions (a.-e.) being satisfied by the Lyapunov function of Eq. 4.1-1 therefore suffices to demonstrate asymptotic stability-in-the-large for the autonomous unforced system of Eq. 4.1-2 or, correspondingly, asymptotic stability for the randomly forced system of Eq. 4.2-1b that is the end objective.

4.3 Analytic Stability of the SLU Filters

In analogy to what is done on p. 200 of Ref. 10 for continuous-time, an existing stability theorem that serves as the basis of what is done in Sections 4.3 and 4.4 is as follows (same statement as Theorem 4.1 of Ref. 16, but proof is augmented and corrected as indicated in Appendix A):

Theorem 4.3-1 (Anderson):

If $\Phi(k,k-1)$, $\Phi^{-1}(k,k-1)$, $H(k)$, $Q(k)$, and $R^{-1}(k)$ are bounded, $H(k)$ is of full rank, $[\Phi, HR^{-1/2}]$ is uniformly completely observable, and condition 3' (as Eq. A.1-1) is satisfied, then a filter of the form of Eqs. 4.2-1b to 4.2-5 is asymptotically stable (in the large).

For the local SLU filters, as long as the pair $H_i(k)R_i^{-1/2}(k)$ (of Eq. 1.2-2a) and $\Phi_{ii}(k+1,k)$ (of Eq. 1.2-40) are uniformly completely observable, $H_i(k)$ is bounded and of full rank, and condition 3' is satisfied via an appropriate choice of

$P_o > 0$

then each local SLU filter is stable. This conclusion is reached by making the following associations in Eq. 2.2-41 and Eq. 2.3-22b

$$\Phi \longleftarrow (\Phi_{ii} - \bar{L}'_{ii}\bar{H}_{i1}) \tag{4.3-2}$$

$$K \longleftarrow K_{i2} \tag{4.3-3}$$

$$Q \longleftarrow Q_i + \bar{L}'_{ii} R_{i1} \bar{L}'^{T}_{ii} \tag{4.3-4}$$

$$H \longleftarrow \bar{H}_{i2} \tag{4.3-5}$$

$$R \longleftarrow R_{i2} \tag{4.3-6}$$

$$P_o \longleftarrow P_{oi} \tag{4.3-7}$$

to obtain equations corresponding to Eqs. 4.2-1b and 4.2-6a, respectively. The SPA filter has a structure that also allows stability to be demonstrated by invoking Theorem 4.3-1.

4.4 Analytic Stability of the SPA Filter

For the condition of Eq. 3.1-19 (as encountered for JTIDS RelNav in Refs. 15 and 35), the mechanization equations for the SPA filter are Eqs. 3.1-23, Eq. 3.1-12 (degenerating to 3.1-22), Eq. 3.1-17, Eq. 3.1-16, Eq. 3.1-24 (degenerating to Eq. 4.1-2 for condition of Eq. 3.1-19), and Eq. 3.1-18. Then by an association similar to that used in Section 4.3 and Ref. 10 with notable equivalences such as

$$R \longleftarrow R_{ii}^{*} \tag{4.4-1}$$

$$H \longleftarrow \bar{H}_i \tag{4.4-2}$$

$$K \longleftarrow \bar{K}_i \tag{4.4-3}$$

and the other associations being even more obvious, Theorem 4.3-1 can be invoked to conclude asymptotic stability of the SPA filter as specialized for the condition of Eq. 3.1-19 since it has the requisite structure [utilized in Ref. 10 as enunciated in Ref. 16 (with proof corrected herein in Appendix A)].

5 APPLICATIONS OF DECENTRALIZED FILTERING

Throughout Refs. 8-12, potential applications of decentralized filtering are indicated by examples in interconnected power systems for frequency monitoring (as a prelude to stabilized maintenance) and power load estimation. Refs. 15 and 35 investigate application of decentralized filtering to the Joint Tactical Information Distribution System (JTIDS) Relative Navigation (RelNav) feature currently being developed by the U.S. Joint Services and eventually intended for NATO. The SPA decentralized filtering formulation is recommended in Refs. 15 and 35, since it possesses a reasonably mild computational burden and an analytic guarantee of "filter stability" (viz., an ability to at least track the true state adequately whether or not the true states are stable) as a prerequisite for answering any other more probing questions concerning the stability of the JTIDS net or of the common grid defined by the "controller" for relative navigation/targeting. Further discussion and simulation studies of the SPA and other approaches to RelNav are provided in Ref. 60.

Sensor fusion is a concept that pervades several fields as addressed for Identification, Friend, Foe, or Neutral (IFFN) in Refs. 63, 64, and 65 and for Communication, Command, and Control (C^3) on p. 204 of Ref. 67. Ref. 63 indicates the breadth of approaches that are being pursued for the next generation of IFF beyond just the interegator/transponder beacon of the current L-band Mark 12. While Hughes/Fullerton has pursued IFFN using information theoretic techniques (i.e., Shannon theory, entropy arguments, and minimization of well-posed cost functions for achieving objectives) for aircraft identification, Hughes/Culver City has used the Bayesian approach to examine critical issues of intersensor correlation, and Ref. 63 considers theoretical and practical aspects of implementing Bayesian-based maximum likelihood decisions and majority rule decisions in this application. Indeed, Dynamics Research Corporation (DRC) has reportedly investigated use of "fuzzy set" techniques for reducing the indicated computational burden for this challenging problem, where even engine harmonics are being exploited for aircraft identification (Ref. 68) by fighter aircraft in the same vein as the aircraft identification from radar signatures performed by the E-3A AWACS, but with greater computational capacity than a fighter is availed with. A recently developed methodology for potentially reducing the computational burden in providing adequate data base handling for these life-or-death (friend or foe) classifications (by a novel conversion of the

problem of queries on an imprecise data base into a problem of statistical inference) is described by E. Wong in Ref. 73. Essential aspects of a Bayesian filtering approach and consideration of alternative (but necessary) approximations are treated in Ref. 66. This IFFN area, where several possibly correlated measurement sensors are utilized in making identification decisions under dynamic conditions, appears likely for fruitful utilization of decentralized filtering techniques such as Speyer's (Refs. 32 and 33) or the multirate multiple filtering approach of Section 1.5.

A fairly bleak picture of the U.S. and Allied C^3, as of 1980, as made evident from the unpleasant experiences of the "Nifty Nuget" war gaming exercise simulating an all-out conventional war against the attacking forces of the Warsaw Pact in Europe, is portrayed in Ref. 74 as a variety of solutions are sought for aspects of the total C^3 problem. An analytic framework for C^3 considerations in a form compatible with modern state-variable estimation techniques is laid out by the Technical Director of the Naval Electronic Systems Command in Ref. 70, with further vital descriptive elaborations in Refs. 75 and 76. Another perspective on current research issues persisting as problems in C^3 are examined in Ref. 67 and the analytical nature of compatible hierarchical decentralized structures for C^3 are considered in Refs. 67, 69, 70, and 71. On the other hand, it is important to take into account the moderating remarks of Ho (Ref. 77) to the effect that a pat solution to the C^3 problem is not readily at hand from estimation and control system theorist, but must be carefully tailored and developed in order to successfully resolve the C^3 problems. Leads toward this end are offered by Ref. 78.

Current specifications for the Phase 1 integration of the JTIDS RelNav and Global Positioning System (GPS) on the F-16A call for utilization of three separate filters, one for GPS, one for RelNav, and one dedicated to aided inertial navigation. This type of situation appears a likely candidate for the multirate filtering approach of Section 1.5 (as already applied to a navigation example in Ref. 61). The GPS filter could be used to incorporate position and velocity information at a fast rate in an unjammed environment, then feed it to a slower-rate higher fidelity navigation filter used for aiding the inertial navigation system in an integrated manner.

For two separate GPS and JTIDS filters of dimension 12 and 15, respectively, as considered in Ref. 82 (which, unfortunately, ignored filter throughput considerations) the advantage of two over one larger 19 state unified filter is obtained from the ratio of the total number of required operations (Ref. 36) as

$$\frac{(12)^3 + (15)^3}{(19)^3} = \frac{5103}{6859} = 0.74$$

or a 26% reduction in the total number of operations to be performed during each filter cycle even though the INS gyro drift-rate states are modeled twice. Unfortunately, a slight 2% increase in required computer memory allotment is indicated by

$$\frac{(12)^2 + (15)^2}{(19)^2} = \frac{369}{361} = 1.02;$$

however, the large benefit appears to be well worth the slight penalty.

The case favoring two separate filters is even more pronounced when considering an alternate state selection (Ref. 83) corresponding to two filters of state size 12 and 18 versus a single 22 state filter since calculations of the above form indicate savings to be achieved in both the number of operations (equivalent to algorithm cycle time of processing a filter measurement) and computer memory required as, respectively, 30% and 3%.

If two separate digital processors are used, parallel processing of each of the two filters on different machines provides the advantage that the system is only limited by the slower speed of the single larger filter (of = 15 or 18 states). In comparison, the smaller filter of 12 states can proceed through six (6) Kalman filter measurement processing cycles in the same time that a larger unified 22 state filter could complete only one cycle, as indicated by the following ratios:

$$\frac{(22)^3}{(12)^3} = \frac{10648}{1728} = 6.16$$

The conclusion is that a unified single filter will limit processing throughput and hinder full utilization of the GPS measurements available in an unjammed environment.

Another likely navigation application for use of decentralized filtering is in the navigation room of strategic submarines (as discussed in detail on p. 326 of Ref. 35) to avoid unnecessarily redundant state modeling in the three filters currently used for SINS/ESGN navigation in Trident SSBNs. An important SSBN application of another variation of standard Kalman filtering with optimized scheduling of alternative sensor measurements is reported in Ref. 49. The above enumerated list of candidate application areas for decentralized filtering is representative and nonexhaustive.

APPENDIX A: CORRECTING PAST STABILITY METHODS PRIOR TO GENERALIZATION FOR ESTABLISHING STABILITY OF DECENTRALIZED FILTERS

A.1 Status Review and a Counterexample

An interesting aspect of Kalman filtering is the so-called "robustness" or eventual correct time evolution of the solution of the Riccati equation despite an incorrectly specified initial value P_o (as long as it is positive definite). This aspect is particularly useful in applications where the situation of ignorance of initial uncertainty in the states of the filter model can (and does) occur, but without deleterious consequences as a result of precisely this robustness property. Kalman (Ref. 53) gave the first proof of this robustness result under the strong hypotheses of uniform complete observability and uniform complete controllability to provide the strong conclusion that the correct estimates are homed in upon at an exponential rate. B. D. O. Anderson (Ref. 16) provided a weaker conclusion of only asymptotic stability (rather than stability at an exponential rate) due to weaker but more frequently meet hypotheses that remove the explicit requirement of uniform complete controllability by requiring only that condition 3' (cf. Eqs. 4.2-11, -12):

$$\left[P_o + \sum_{k=k_o}^{k_1-1} \phi(k_o,k+1)\, Q(k+1)\phi^T(k_o,k+1)\right] \text{ is nonsingular for some } k_1 \geq k_o \qquad \text{(A.1-1)}$$

where P_o is the covariance of the Gaussian initial condition x_o, Q is the process noise covariance, and ϕ is the discrete-time transition matrix of the linear system model. It is emphasized on p. 223 of Ref. 58 that the condition of Eq. A.1-1 is satisfied if P_o is nonsingular (as can be selected) "irrespective of $\phi(k+1,k), Q(k)$," or even explicit controllability being achievable (an example being provided on pp. 137-8 of Ref. 16 of an estimator that is asymptotically stable despite the absence of process noise). Anderson's stability proofs for both continuous-time and discrete-time utilize slightly modified forms of standard arguments for Lyapunov functions V(x,t). In the case of continuous time, Anderson utilizes the following inequality (unnumbered equation following Eq. 23 of Ref. 16)

$$\dot{V}(x,t) \leq -x^T H^T R^{-1} H x \qquad \text{(A.1-2a)}$$

$$< 0 \qquad \text{(A.1-2b)}$$

where R is the measurement noise covariance, H is the observation matrix, and x is the state. Jazwinski's result (p. 241, unnumbered equations following Eq. 7.198 of Ref. 51) (as credited to Deyst and Price as Ref. 40, but now known to be in error as discussed in the preceding paragraph) of

$$V(x(k),k) - V(x(k-1),k-1) \leq -x^T(k) H^T(k) R^{-1}(k) H(k) x(k) - U^T(k) P^{-1}(k|k-1) U(k) \qquad \text{(A.1-3)}$$

with

$$U(k) \triangleq \left[P(k|k)P^{-1}(k|k-1) - I\right] \phi(k,k-1) x(k-1) \qquad \text{(A.1-4)}$$

is also utilized in Ref. 16 for the discrete-time case, but Anderson goes further since

$$-x^T(k) H^T(k) R^{-1}(k) H(k) x(k) - U^T(k) P^{-1}(k|k) U(k) \leq -x^T(k) H^T R^{-1}(k) H(k) x(k) \qquad \text{(A.1-5)}$$

to use only

$$V(x(k),k) - V(x(k-1),k-1) \leq -x^T(k) H^T(k) R^{-1}(k) H(k) x(k) \qquad \text{(A.1-6)}$$

$$< 0 \qquad \text{(A.1-7)}$$

under the conditions that

$$H(k) \text{ and } R^{-1}(k) \text{ are bounded for all } k \qquad \text{(A.1-8)}$$

[where the boundedness condition on H(k) is stated on p. 141 of Ref. 16 to be a new requirement]. Unfortunately, while the inequality of Eq. A.1-6 is true, the inequalities of both Eqs. A.1-2b and A.1-7 are questionable since for

$$R \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad H \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad x \triangleq \begin{bmatrix} 0 & 1 \end{bmatrix}^T \qquad \text{(A.1-9)}$$

it happens that

$$x^T H^T R^{-1} H x = 0 \not< 0 \qquad \text{(A.1-10)}$$

thus contradicting both Eqs. A.1-2b and A.1-7 that are utilized in Ref. 16. While an additional condition of requiring that H(k) also be of full rank will circumvent these

difficulties, the discrete-time case still relies on the [illegible]
40. Anderson's invalid stability proof has also been utilized by Sanders et al. (Refs. 10 and 11) in demonstrating asymptotic stability of decentralized filters of the Surely Locally Unbiased (SLU) class so Refs. 10 and 11 unknowingly inherited a flaw. However, the results can be patched up as done in Section A.2 by using the correction supplied by McGarty (Ref. 18) and Deyst (Ref. 41). Aasnaes and Kailath (Ref. 52) weakened the required assumptions to establish stability of the centralized filter even further and proved convergence at a t^{-k} rate (for some $k > 1$).

A.2 A Minor Correction to Update Anderson's Generalization

Following McGarty's rigorous derivation of Eq. C.127 of Ref. 18 (analogous to Eq. 57 in Ref. 17 and the unnumbered equation prior to Eq. 7.199 in Ref. 52), the following is obtained

$$V[\hat{x}(k|k-1),k] \leq \hat{x}^T(k-1|k-2)P^{-1}(k-1|k-1)\hat{x}(k-1|k-2)$$
$$- \hat{x}^T(k|k-1)H^T(k)R^{-1}(k)H(k)\hat{x}(k|k-1)$$
$$- u^T(k)[\Phi(k,k-1)P(k-1|k-1)\Phi^T(k,k-1)+Q(k)]^{-1}u(k) \qquad (A.2\text{-}1)$$

where $u(k)$ is appropriately defined in Eq. C.118 of Ref. 18 (analogous to Eq. 7.197 of Ref. 51 and Eq. 56 of Ref. 17)

$$u(k) \triangleq [P(k|k)P^{-1}(k|k-1) - I]\Phi(k,k-1)\hat{x}(k-1|k-2) \qquad (A.2\text{-}2)$$

Since the right-hand side of Eq. A.2-1 is upper bounded all the more if fewer nonnegative quantities are subtracted from it, Eq. A.2-1 is altered (as outlined for discrete-time on p. 143 of Ref. 16) to yield

$$V[\hat{x}(k|k-1),k] \leq \hat{x}^T(k-1|k-2)P^{-1}(k-1|k-1)\hat{x}(k-1|k-2)$$
$$- \hat{x}^T(k|k-1)H^T(k)R^{-1}(k)H(k)\hat{x}(k|k-1) \qquad (A.2\text{-}3)$$

or, equivalently,

$$V[\hat{x}(k|k-1),k] - \hat{x}^T(k-1|k-2)P^{-1}(k-1|k-1)\hat{x}(k-1|k-2)$$
$$= V[\hat{x}(k|k-1),k] - V[\hat{x}(k-1|k-2),k-1] \qquad (A.2\text{-}4a)$$
$$\leq -\hat{x}^T(k|k-1)H^T(k)R^{-1}(k)H(k)\hat{x}(k|k-1) \qquad (A.2\text{-}4b)$$
$$< 0 \qquad (A.2\text{-}4c)$$

where as offered as one correction following Eqs. A.1-9 and A.1-10

$$H(k) \text{ must be of full rank} \qquad (A.2\text{-}5)$$

to guarantee the requisite strict inequality needed in Eq. A.2-4c as a verification of condition (c) as Eq. 4.2-11 (without utilizing the strong condition of Eq. 4.2-11 but letting only the weaker more generally met condition 3' of Eq. A.1-1 suffice).

Throughout Eqs. A.2-1 to A.2-4 a tacit assumption is made that $P^{-1}(k|k-1)$ exists [i.e., $P(k|k-1)$ is nonsingular] otherwise it could not be routinely utilized in the above. The pertinence of Anderson's weaker condition 3' (Eq. A.2-1) first proposed in Ref. 16 now becomes evident since a discrete-time version (i.e., paralleling) his continuous-time proof of Lemma 3.1, p. 139 of Ref. 16 reveals that $P(k|k-1)$ that evolves in time from Eqs. 4.2-3, 4.2-4, and 4.2-5 and the matrix of Eq. A.1-1 have an identical nullspace. This means that if singularity occurs, it is simultaneous. Hence by a proper selection of P_o to be nonsingular, both Eq. A.1-1 and $P(k|k-1)$ are guaranteed to be nonsingular, and therefore invertible as needed to demonstrate asymptotic stability.

REFERENCES

1 Athans, M., "The Matrix Minimum Principle," Information and Control, Vol. 11, pp. 592-606, 1968.

2 Walsh, P. M., "On Symmetric Matrices and the Matrix Minimum Principle," IEEE Transactions on Automatic Control. Vol. AC-22, No. 6, Dec. 1977.

3 Athans, M., and Tse, E., "A Direct Derivation of the Optimal Linear Filter Using the Maximum Principle," IEEE Transactions on Automatic Control, Vol. AC-12, No. 6, pp. 690-698, 1967.

4 Sage, A. P., and White, C. C., Optimum Systems Control, Second Edition, Prentice-Hall, Englewood Cliffs, N.J., 1977.
5 Gupta, S. C., "Phase-Locked Loops," Proceedings of the IEEE, Vol. 68, No. 2, Feb. 1975.
6 Leondes, C. T., Phillis, I. A., and Chin, L., "Method of Optimizing the Update Intervals in Hybrid Navigation Systems," Journal of Guidance and Control, Vol. 2, No. 6, Nov.-Dec. 1979.
7 Sage, A. P., and Melsa, J. L., Estimation Theory with Applications to Communications and Control, McGraw-Hill Book Company, New York, 1971.
8 Sanders, C. W., Tacker, E. C., and Linton, T. D., "Decentralized Estimation Via Constrained Filters," Technical Report ECE-73-1, University of Wisconsin-Madison, 1973.
9 Sanders, C. W., Tacker, E. C., and Linton, T. D., "A Decentralized Filter for Interacting Systems," Proceedings of the Fourth Symposium on Non-Linear Estimation Theory and its Applications, San Diego, CA, 1973.
10 Sanders, C. W., Tacker, E. C., and Linton, T. D., "Stability and Performance of a Class of Decentralized Filters," International Journal of Control, Vol. 23, No. 2, 1976.
11 Sanders, C. W., Tacker, E. C., Linton, T. D., and Ling, R.Y-S., "Specific Structures for Large Scale State Estimation Algorithms Having Information Exchange," IEEE Transactions on Automatic Control, Vol. AC-23, No. 2, Apr. 1978.
12 Ling, R.Y-S., "Design and Evaluation of Decentralized Filters for Large-Scale Interconnected Systems," Ph.D. Thesis, University of Houston, Spring 1979.
13 Singh, M. G., and Titli, A., Systems: Decomposition, Optimization, and Control, Pergamon Press, New York, 1978.
14 Shah, M., "Suboptimal Filtering Theory for Interacting Control Systems," Ph.D. Thesis, Cambridge University, Cambridge, England, 1971.
15 Kerr, T. H., "Stability Conditions for the RelNav Community as a Decentralized Estimator—Final Report," Intermetrics, Inc., Report No. IR-480, 10 August 1980.
16 Anderson, B. D. O., "Stability Properties of Kalman-Bucy Filters," Journal of the Franklin Institute, Vol. 291, 1971.
17 Hitz, K. L., Fortmann, T. E., and Anderson, B. D. O., "A Note on Bounds on Solutions of the Riccati Equation," IEEE Transactions on Automatic Control, Vol. AC-17, No. 1, p. 178, Feb. 1972.
18 McGarty, T. P., Stochastic Systems and State Estimation, Wiley-Interscience, New York, 1974.
19 Maybeck, P. S., Stochastic Models, Estimation, and Control, Vol. 1, p. 244, Academic Press, New York.
20 Chin, L., "JTIDS Relative Navigation Overview," Proceedings of the International Conference on Information Sciences and Systems, Patras, Greece, 9-13 July 1979.
21 Pearson, J. D., "Dynamic Decomposition Techniques," in Optimization Methods for Large-Scale Systems, edited by D. A. Wismer, McGraw-Hill Book Company, 1971.
22 Smith, N. J., and Sage, A. P., "An Introduction to Hierarchical Systems Theory," Computers and Electrical Engineering, Vol. 1, pp. 55-71, Pergamon Press, New York, 1973.
23 Mesarovic, M. D., Macko, D., and Takahara, Y., Theory of Hierarchical Multilevel Systems, Academic Press, New York, 1970.
24 Mehra, R. K., and Davis, R. E., "A Generalized Gradient Method for Optimal Control Problems with Inequality Constraints and Singular Arcs," Proceedings of Joint Automatic Control Conference, pp. 144-151, 1971.
25 Gulnzy, N. J., and Sage, A. P., "System Identification in Large-Scale Systems with Hierarchical Structures," Computers and Electrical Engineering, Vol. 1, p. 23-42, 1973.
26 Sundareshan, M. K., "Generation of Multilevel Control and Estimation Schemes for Large-Scale Systems: A Perturbational Approach," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, No. 3, March 1977.
27 Sundareshan, M. K., "Decentralized Observation in Large-Scale Systems," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, No. 12, Dec. 1977.
28 Hassan, M. F., Salut, G., et al., "A Decentralized Computational Algorithm for the Global Kalman filter," IEEE Transactions on Automatic Control, Vol. AC-23, No. 2, Apr. 1978.
29 Hassan, M. F., "Optimum Kalman Filter for Large-Scale Systems Using the Partitioning Approach," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 5, pp. 714-720, Oct. 1976.
30 Siljak, D. D., and Vukcevic, M. B., "On Decentralized Estimation," International Journal of Control, Vol. 27, No. 1, pp. 113-131, 1978.
31 Verriest, E., Friedlander, B., and Morf, M., "Distributed Processing in Estimation and Detection," Proceedings of 18th IEEE Conference on Decision and Control, Fort Lauderdale, FL, pp. 153-158, Dec. 12-14, 1979.
32 Speyer, J. L., "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem," IEEE Transactions on Automatic Control, Vol. AC-24, No. 2, Apr. 1979.
33 Chang, T.-S., "Comments on 'Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control,'" IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, June 1980.
34 Asher, R. B., Herring, K. D., and Ryles, J. C., "Bias, Variance, and Estimation Error in Reduced Order Filters," Automatica, Vol. 12, pp. 589-600, Nov. 1976.
35 Kerr, T. H., and Chin, L., "A Stable Decentralized Filtering Implementation for JTIDS RelNav," Proceedings of Position, Location, and Navigation Symposium (PLANS), Atlantic City, N.J., Dec. 8-11, 1980.
36 Mendel, J. M., "Computational Requirements for a Discrete Kalman Filter," IEEE Transactions on Automatic Control, Vol. AC-16, pp. 748-758, Dec. 1971.
37 Gura, I. A., and Bierman, A. B., "On Computational Efficiency of Linear Filtering Algorithms," Automatica, Vol. 7, pp. 299-314, 1971.

38 Klema, V. C., and Laub, A. J., "The Singular Value Decomposition: Its Computation and Some Applications," IEEE Transactions on Automatic Control, Vol. AC-25, No. 2, pp. 164-176, Apr. 1980.

39 Garbow, B. S., et al., Matrix Eigensystem Routines—EISPACK Guide Extension, Lecture Notes in Computer Science, Vol. 51, Springer-Verlag, New York, 1977.

40 Deyst, J. J., and Price, C. F., "Conditions for Asymptotic Stability of the Discrete Minimum-Variance Linear Estimator," IEEE Transactions on Automatic Control, Vol. AC-13, No. 6, pp. 702-705, Dec. 1968.

41 Deyst, J. J., "Correction to Conditions for Asymptotic Stability of the Discrete Minimum-Variance Linear Estimator," IEEE Transactions on Automatic Control, pp. 562-563, Oct. 1973.

42 Luenberger, D., "Canonical Forms for Linear Multivariable Systems," IEEE Transactions on Automatic Control, Vol. AC-12, No. 3, pp. 290-293, June 1967.

43 Chen, C. T., Introduction to Linear System Theory, Holt, Rinehart, and Winston, New York, 1970.

44 Hsu, K., "On Discrete-Time Centralized and Decentralized Estimation and Stochastic Control," Ph.D. Thesis, University of Texas at Austin, Dec. 1979.

45 Yamada, T., "Comments on 'Two-Level Form of Kalman Filter,'" IEEE Transactions on Automatic Control, Vol. AC-25, No. 4, Aug. 1980.

46 Kerr, T., "Rectifying Several Pervasive Errors Appearing in Estimation and Control Theory" (submitted to IEEE Transactions on Automatic Control in 1981).

47 Sims, C. S., and Asher, R. B., "Optimal and Suboptimal Results in Full- and Reduced-Order Linear Filtering," IEEE Transactions on Automatic Control, Vol. AC-23, No. 3, June 1978.

48 Sims, C. S., and Stotts, L. C., "Linear Discrete Reduced Order Filtering," Proceedings of the 1978 IEEE Conference on Decision and Control, San Diego, CA, Jan. 1979.

49 Kerr, T. H., "Modeling and Evaluating an Empirical INS Difference Monitoring Procedure Used to Sequence SSBN Navaid Fixes," Proceedings of the Institute of Navigation Annual Meeting, Annapolis, MD, 19-20 June 1981 (also to appear in Navigation).

50 Nering, E. D., Linear Algebra and Matrix Theory, John Wiley and Sons, Inc., New York, 1963.

51 Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

52 Aasnaes, H. B., and Kailath, T., "Initial-Condition Robustness of Linear Least Squares Filtering Algorithms," IEEE Transactions on Automatic Control, Vol. AC-19, No. 4, pp. 393-398, Aug. 1974.

53 Kalman, R. E., "New Methods in Wiener Filtering Theory," in Proceedings of the Symposium on Engineering Applications of Random Function Theory and Probability, edited by J. L. Bogdanoff and F. Kozin, Wiley, New York, 1963.

54 Laub, A. J., and Bailey, F. N., "An Iterative Coordination Approach to Decentralized Decision Problems," IEEE Transactions on Automatic Control, Vol. AC-23, No. 6, pp. 1031-1036, Dec. 1978.

55 Aoki, M., "Control of Large-Scale Dynamic Systems by Aggregation," IEEE Transactions on Automatic Control, Vol. AC-13, No. 3, pp. 246-253, June 1968.

56 Aoki, M., "Aggregation," in Optimization Methods for Large-Scale Systems, edited by D. A. Wismer, McGraw-Hill, New York, 1971.

57 Aoki, M., "Some Approximation Methods for Estimation and Control of Large-Scale Systems," IEEE Transactions on Automatic Control, Vol. AC-23, No. 2, Apr. 1978.

58 Anderson, B. D. O., "Exponential Data Weighting in the Kalman Bucy Filter," Information Sciences, Vol. 5, pp. 217-230, 1973.

59 Kalman, R. E., and Bertram, J. E., "Control System Analysis and Design Via the Second Method of Lyapunov, Part II: Discrete-Time Systems," Journal of Basic Engineering, pp. 394-400, June 1960.

60 Gobbini, G. F., "Relative Nagivation by Means of Passive Rangings," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, M.I.T., Cambridge, MA, June 1981.

61 Medan, Y., and Bar-Itzhack, I. Y., "Error and Sensitivity Analysis Scheme of a New Data Compression Technique in Estimation," TAE Report No. 401, Technion Israel Institute of Technology, Department of Aeronautical Engineering, Haifa, Israel, May 1980 (to appear in AAIA Journal of Guidance and Control).

62 Bar-Itzhack, I. Y., "A Novel Method for Data Compression in Recursive INS Estimation," AAIA Journal of Guidance and Control, Vol. 3, No. 3, 1980.

63 Nahin, P. J., and Pokoski, J. L., "NCTR Plus Sensor Fusion Equals IFFN or Can Two Plus Two Equal Five," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-16, No. 3, pp. 320-337, May 1980.

64 Nahin, P. J., "IFFN: A Technological Challenge for the '80's," Air University Review, Vol. 27, pp. 2-16, Sept.-Oct. 1977.

65 Vighetta, B. P., and Murrow, D. J., "Comparison of Sea-Based Active and Passive Microwave Sensors for Detecting Low-Flying Targets," Proceedings of EASCON, 1975.

66 Makov, U. E., "Approximations to Unsupervised Filters," IEEE Transactions on Automatic Control, Vol. AC-25, No. 4, pp. 842-847, Aug. 1980.

67 Sandell, N. R., Laner, G. S., and Kramer, L. C., "Research Issues in Surveillance for C^3," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, pp. 201-210, 10-12 Dec. 1980.

68 Lindsley, W., "Simple/Complex Fighters," Aviation Week Space & Technology, p. 88, March 1981.

69 Drenick, R. F., "A Mathematical Approach to Organization Theory," Proceedings of the 19th IEEE Conference on Decision and Control," Albuquerque, NM, pp. 226-230, 10-12 Dec. 1980.

70 Lawson, J. S., "Command Control as a Process," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, pp. 1-6, 10-12 Dec. 1980.

71 Loparo, K. A., "Overlapping Control Structures and Security in Large Scale Systems," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, pp. 214-218, 10-12 Dec. 1980.

72 Conrad, E. R., "Preliminary Interface Specification F-16A to Joint Tactical Information Distribution System," General Dynamics, Fort Worth, TX, 30 March 1979.
73 Wong, E., "Incomplete Information in Database Systems," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, pp. 214-218, 10-12 Dec. 1980.
74 Broad, W. J., "Philosophers at the Pentagon," Science, Vol. 210, pp. 409-412, 24 Oct. 1980.
75 Schutzer, D., "Command and Control—Problems and Concerns," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, pp. 7-11, 10-12 Dec. 1980.
76 Athans, M., "System Theoretic Challenges and Research Opportunities in Military C^3 Systems," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, pp. 12-16, 10-12 Dec. 1980.
77 Ho, Y. C., "Games, Information, and Simulation in C^3," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, p. 219, 10-12 Dec. 1980.
78 Hajek, B., "Dynamic Decentralized Estimation and Control in a Multi-Access Broadcast Channel," Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, NM, pp. 618-623, 10-12 Dec. 1980.
79 Klinger, A., "Prior Information and Bias in Sequential Estimation," IEEE Transactions on Automatic Control, Vol. AC-13, No. 1, pp. 102-103, Feb. 1968.
80 Widnall, W. S., and Gobbini, G., "Decentralized Relative Navigation and JTIDS/GPS/INS Integrated Navigation Systems," First Interim Technical Report, Dept. of Aeronautics and Astronautics, M.I.T., Cambridge, MA (prepared for Avionics Laboratory of Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base), 10 June 1980.
81 Willsky, A., Bello, M., Castanon, B. C., Levy, G., and Verghese, G., "Combining and Updating of Local Estimates and Regional Maps Along Sets of One-Dimensional Tracks," Paper No. 955, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, Nov. 1979.
82 Kriegsman, B. A., and Stonestreet, W. M., "A Navigation Filter for an Integrated GPS/JTIDS/INS System for a Tactical Aircraft," Proceedings of IEEE Position, Location, and Navigation Symposium (PLANS), San Diego, CA, pp. 316-320, 6-9 Nov. 1978.
83 "MFBARS Radio Architecture Study: Final Draft," Technical Report by TRW, Redondo Beach, CA, March 1980.
84 Garbow, B. S., et al., Matrix Eigensystem Routines—EISPACK Guide Extension, Lecture Notes in Computer Science, Vol. 51, Springer-Verlag, New York, 1977.

ACKNOWLEDGEMENTS

This work was supported in part by the Naval Air Development Center under Intermetrics Contract No. N62269-79-D-0301. The authors would like to acknowledge the contribution of Dr. Jaw-Yih Wang in the preparation of the parametric summations preliminary to Table 2.4-1. Useful observations were made by Dr. Neal Carlson for improving the efficiency of implementing Speyer's filter that were more numerous than could be conveniently shown here. Thanks are also due for the useful assistance of Mr. Ben Polin of NADC. Special thanks are due to Dr. Alfred Gilman for discussions, encouragement, and creating an environment conducive to such investigations.

ADVANCES IN COMPUTATIONAL EFFICIENCIES OF LINEAR FILTERING

Leonard Chin
Naval Air Development Center
Warminster, Pennsylvania 18974
U.S.A.

A broad overview of various discrete-time linear filtering techniques including the Square-Root and variations of Square-Root, Factorized, Chandrasekhar, Partitioning and Decentralized algorithms as well as the basic Covariance and Information filters are presented. The purpose of this chapter is to examine and compare computer burdens of these well-known filtering algorithms from practical operation point of view.

1.0 INTRODUCTION

The entreating problem of estimation (prediction, filtering, smoothing) has attracted many mathematicians, scientists and engineers throughout history - beginning with the Babylonians who had applied a form of mathematics that was similar to the Fourier series to interpret observations and make decisions. Recent history has shown that Euler, Lagrange, Laplace, Bernoulli and others had contributed to the advancement of estimation theories in many ways. However, the solution to a class of estimation problems in finding the best value of an unknown parameter corrupted by noise (additive) was given by Gauss and Legendre, who had separately but concurrently formulated the method of Least-Squares, in which the fundamental concepts of redundant measurement data, observability, dynamic modelling, etc. were introduced. These concepts have provided foundations for many new developments in estimation theories known today. Readers who are interested in antiquity should consult Neugebauer [1] and Sorenson [2].

Modern history has recorded that Fisher [3] was one of the original investigators in the formulation of statistical estimation methods. However, the development of the linear, minimum, mean-square-error estimator, which is most familiar to engineers today, had not been completed until the 1940's when the Wiener-Hopf equation was first established by Wiener [4] and Kolmogorov [5]. Unfortunately the filter design using the Wiener-Hopf equation was proved to be impractical due to difficulties in obtaining explicit solutions. Swerling [6] as well as Carlton and Follin [7] circumvented these difficulties by obtaining the Least-Square estimates in a recursive manner. This new development is believed to be the starting point of the so-called "Kalman Filtering". The main contribution of Kalman and Bucy [8,9] was the transformation of the Wiener-Hopf integral equation into an equivalent nonlinear differential equation; the solution of which yields the covariance matrix, which contains all necessary information for the design of the optimal filter. Hence, by demanding a numerical rather than an analytical solution of the Wiener-Hopf equation, they have successfully developed a recursive filter that can be conveniently realized in real-time by a digital computer. In practical applications, the on-line computer used to implement the filter equations is usually limited in speed and memory. Frequently, it is not possible to program the ideal (theoretical filter) equations, since the truly optimal filter must model all error sources of the system. The solution to this problem (excessively large dimension) is usually to design suboptimal filters, in which certain states in the system have to be ignored or simplified. Another implementation problem associated with all digital computers is the numerical accuracy, which is seriously affected by the inherent finite nature of the computer. For example, in the case of the Kalman-Bucy algorithm, the covariance matrix contains all information needed for the filter synthesis (hence, it is called the Covariance Filter). For this reason, it is critical that the correctness of this matrix must be preserved at all times. However, because of matrix subtractions involved in each computing cycle, non-positive elements in the covariance matrix could result from truncation and round-off errors. Chin [10] has reported that this computational inability could lead to filter divergence.

In order to preserve the correctness of the covariance matrix (hence preventing filter divergence), a number of methods called "Square Root Covariance Algorithms" have been introduced [11-20]. In a parallel effort to preserve the correctness of the information matrix (inverse of covariance matrix) in the "Information Filter", the Square-Root Information Filter have also been developed [21-23]. While these algorithms generally yield improved numerical stability, the number of computational steps is usually greater than that of the standard Kalman filter. For this reason, engineers do have a certain degree of reluctance to employ the Square-Root algorithms. An attempt to improve this situation was made by Bierman [24] who has employed factorization techniques [25,26], which do not involve square root computations.

In the early 1970's, a different approach was explored for the design of recursive filters that avoids the utilization of the matrix Riccati equation. Kailath [27] has shown that for a certain class of special applications, the so-called X-Y function of Chandrasekhar could be used to gain two magnitudes of computational improvement over the usual Riccati-type algorithms. Independently, Linquist [28,29] and Lainiotis [30] have derived Chandrasekhar algorithms for discrete-time-invariant and continuous-time-varying systems, respectively. However, there is no guarantee that these algorithms will provide computational stability. Other intrinsic aspects of Chandrasekhar filters were discussed by Bremmer [31].

Recent advancement in the design of practical filter and smoother was to seek entirely new algorithms that not only provide computational stability but also minimize computer burdens. This new approach has been taken by Lainiotis [32-38] who introduced partitioning estimation concepts which are radically different from those of the Kalman footing. The partitioning approach constitutes essentially an adaptive framework, which yields fundamentally new estimation algorithms in naturally decoupled, parallel-processing realization, and most important of all, computationally attractive.

Another new attempt to economize computational steps as well as to ensure filter stability is the Surely Locally Unbiased (SLU) decentralized approach. The basic concept is to decompose a large system into small subsystems in such a manner that the filter for the overall system shall be optimized. This

efficient than the computation of one aggregated high order system. Also the decentralized filter has a special property that provides asymptotic stability.

The purpose of this paper is to examine, from computational aspects, advantages and disadvantages of various discrete-time linear filtering algorithms mentioned above. In order to facilitate the discussion, a linear system model as well as notations and conventions are defined in Section 2. Prior to the investigation of computer burdens, an overview of filtering algorithms selected for this study is presented in Section 3. The main contribution of this paper - a survey of filtering computational efficiencies - is given in Section 4, in which computer time and memory requirements for the Kalman (covariance and information filters), Square-Root, Factorized, Chandrasekhar, Partitioning and Decentralized algorithms are summarized. Finally, conclusions as well as recommendation are given in Section 5.

2.0 SYSTEM MODEL, NOTATIONS AND CONVENTIONS

The linear filtering problem treated in this paper is being stated using the following discrete-time model:

$$x(k+1) = \Phi(k+1,k)\, x(k) + u(k) \tag{1}$$

$$z(k) = H(k)\, x(k) + v(k) \tag{2}$$

For a given set of measurements:

$$\lambda_k = z(\ell),\ z(\ell+1),\ \ldots,\ z(k)$$

it is desired to find the optimal (in the sense of mean-square-error) filtered estimate $\hat{x}(k/k,\ell)$ of $x(k)$. It is assumed that $u(k)$ and $v(k)$ are uncorrelated zero-mean Gaussian noise with covariances $Q(k)$ and $R(k)$, respectively. Other notations and conventions are given as follows:

SYMBOL	DEFINITION
$x(k)$	system state vector at discrete time index k
$x(\ell)$	initial system state vector (Gaussian distributed)
$\bar{x}(\ell)$	the mean value of $x(\ell)$
$\hat{x}_n(\ell)$	nominal initial system state vector
$\hat{x}_r(\ell)$	remainder initial system state vector
$\hat{x}(k/k,\ell)$	optimal filtered estimate of $x(k)$
$\hat{x}_n(k/k,\ell)$	nominal optimal filtered estimate of $x(k)$
$\hat{x}_r(k/k,\ell)$	remainder optimal filtered estimate of $x(k)$
$x_i(k)$	i^{th} subsystem state vector
$\hat{x}_i(k+1/k)$	optimal state estimate of the i^{th} subsystem
$x(t)$	system state vector (continuous-time)
$\hat{x}(t)$	optimal filtered estimate of $x(t)$
$v(k)$	innovation sequence
$\Phi(k+1,k)$	system state transition matrix
$z(k)$	measurement vector at discrete time index k
$z_i(k)$	i^{th} subsystem measurement vector
$z_u(k)$	measurement vectors of the "data equation"
$z_x(k)$	measurement vectors of the "data equation"
$\Phi_{ii}(k+1,k)$	i^{th} subsystem state transition matrix
$h^T(k)$	linearized measurement operator (vector)
$H(k)$	measurement matrix
$\bar{H}_i(k)$	i^{th} subsystem measurement matrix
$\tilde{H}_i(k)$	coupling measurement matrix bit. i^{th} and other subsystems
$H_L(k)$	linearized measurement matrix
$F(t)$	fundamental matrix
$F[\hat{x}(t),t]$	linearized fundamental matrix
$u(k)$	system noise vector
$\bar{u}_i(k)$	i^{th} subsystem noise vector
$Q(k)$	system noise covariance matrix
$R_u(k)$	square root of $Q(k)$
$Q_i(k)$	i^{th} subsystem noise covariance matrix
$v(k)$	measurement noise vector
$\bar{v}_i(k)$	i^{th} subsystem measurement noise vector
$R(k)$	measurement noise covariance matrix

Updating

$$\hat{x}(k+1/k+1, \hat{\Delta}) = \hat{x}(k+1/k, \hat{\Delta}) + K(k+1, \hat{\Delta}) [z(k+1) - H(k+1)\hat{x}(k+1/k, \hat{\Delta})] \quad (5)$$

$$K(k+1, \hat{\Delta}) = P(k+1/k, \hat{\Delta}) H^T(k+1) [H(k+1)P(k+1/k, \hat{\Delta}) H^T(k+1) + R(k+1)]^{-1} \quad (6)$$

$$P(k+1/k+1, \hat{\Delta}) = [I - K(k+1, \hat{\Delta}) H(k+1)] P(k+1/k, \hat{\Delta}) \quad (7)$$

Note that Eq. (7) is correct only if the gain $K(k+1, \hat{\Delta})$ is optimum.

2. Stabilized Kalman Filter

The stabilized filter (sometimes called the Joseph algorithm [52]) is less sensitive to computer round-off errors. Another benefit is that it yields correct $P(k+1/k+1, \hat{\Delta})$ even if $K(k+1, \hat{\Delta})$ is non-optimum. The updating covariance matrix is given by

$$P(k+1/k+1, \hat{\Delta}) = [I - K(k+1, \hat{\Delta}) H(k+1)] P(k+1/k, \hat{\Delta}) [I - K(k+1, \hat{\Delta}) H(k+1)]^T + K(k+1, \hat{\Delta}) R(k+1) K^T(k+1, \hat{\Delta}) \quad (8)$$

Other updating and extrapolation equations are the same as Eqs. (3) - (6). Initial conditions for the standard as well as the stabilized filters are defined as follows:

$$E\{x(\hat{\Delta})\} = \hat{x}(\hat{\Delta}/\hat{\Delta}) \quad (9)$$

$$E\{[\hat{x}(\hat{\Delta}) - x(\hat{\Delta}/\hat{\Delta})] [\hat{x}(\hat{\Delta}) - \hat{x}(\hat{\Delta}/\hat{\Delta})]^T\} = P(\hat{\Delta}, \hat{\Delta}) \quad (10)$$

3. Extended Kalman Filter

The "extended" Kalman filtering is a popular technique for treating nonlinearities in the design of minimum variance estimators. Other methods of the same footing (Taylor series expansion) are Iterated Extended Kalman filtering, Gaussian Second-Order filtering, and Linearized Kalman filtering [46].

Since most physical nonlinear systems can be represented by differential equations and measurements are usually available at discrete time, therefore, it is proper as well as convenient (for series expansion) to describe system and measurement models as follows:

$$\dot{x}(t) = f[x(t), t] + u(t) \quad (11)$$

$$z(k) = h[x(t_k)] + v(k) \quad (12)$$

in which $u(t)$ and $v(k)$ are uncorrelated zero-mean Gaussian noise with

$$E[u(t)\ u^T(\tau)] = Q(t) \quad (13)$$

$$E[v(k)\ v^T(k)] = R(k) \quad ; \quad (14)$$

also the initial vector $x(\hat{\Delta})$ is Gaussian with mean and covariance given by Eq. (9) and Eq. (10).

Define:

$$F[\hat{x}(t), t] = \left. \frac{\partial f[x(t), t]}{\partial x(t)} \right|_{x(t) = \hat{x}(t)} \quad (15)$$

$$H_L(k) = \left. \frac{\partial h[x(t_k)]}{\partial x(t_k)} \right|_{x(t_k) = \hat{x}(t_k)} \quad (16)$$

Extrapolation equations are given by:

$$\frac{d\hat{x}(t)}{dt} = f[\hat{x}(t), t] \quad (17)$$

$$\frac{dP(t)}{dt} = F[\hat{x}(t), t]\ P(t) + P(t)\ F^T[\hat{x}(t), t] + Q(t) \quad (18)$$

Update equations are given by:

$$\hat{x}(k+1/k+1, \hat{\Delta}) = \hat{x}(k+1/k, \hat{\Delta}) + K(k+1, \hat{\Delta}) [z(k+1) - h[\hat{x}(k+1/k, \hat{\Delta})]] \quad (19)$$

$$P(k+1/k+1, \hat{\Delta}) = [I - K(k+1)\ H_L(k+1)]\ P(k+1/k, \hat{\Delta}) \quad (20)$$

$$K(k+1, \hat{\Delta}) = P(k+1/k, \hat{\Delta})\ H_L(k+1)\ [H_L(k+1)\ P(k+1/k, \hat{\Delta})\ H_L^T(k+1) + R(k+1)]^{-1} \quad (21)$$

Other forms of variation (Gaussian Second-Order, etc.) are given in reference [46].

B. Square Root Covariance Filters

1. Potter

The first square root algorithm was introduced by Potter [11] for a restricted application of zero system noise and measurements are scalar quantities, i.e., Eqs. (1) and (2) become

SYMBOL	DEFINITION
$R_i(k)$	i^{th} subsystem measurement noise covariance matrix
$r(k)$	scaler measurement noise variance
$P(\ell, \ell)$	a priori state error covariance matrix
$P_x(o)$	a priori state error covariance matrix
$P_n(\ell)$	nominal a priori state error covariance matrix
$P_r(\ell)$	remainder a priori state error covariance matrix
$P(k/k, \ell)$	state error covariance matrix
$P_n(k/k, \ell)$	nominal state error covariance matrix
$P_r(k/k, \ell)$	remainder state error covariance matrix
$P_i(k/k)$	i^{th} subsystem state error covariance matrix
$P^{-1}(k/k, \ell)$	information matrix
$S^{-1}(k/k, \ell)$	square root of information matrix
$R_x(k/k, \ell)$	square root of information matrix
$S(k/k)$	square root of covariance matrix
$R_{ux}(k+1)$	square root of cross-covariance between variables of "data equations"
$\hat{V}(k)$	lower-triangular square root of measurement noise covariance matrix
$\hat{U}(k)$	lower-triangular square root of systems noise covariance matrix
Z	$Z = S^T H$
$U(k/k)$	anti-symmetric matrix chosen to maintain $S(k,k)$ in lower triangular form
$T, \hat{T}(k)$ & $T(k+1)$	transformation matrix whose columns are composed of eigenvectors of the covariance matrix P
$D(k)$, $\bar{D}(k)$	diagonal matrix whose diagonal elements are eigenvalues of the state error covariance matrix
$U(k)$, $\bar{U}(k)$	unit upper triangular matrix
$d(k/k)$	transformed state vector, $d(k/k) = T^{-1}(k/k, \ell)\ x(k/k)$
$K(k)$	Kalman filter gain matrix
$K_n(k, \ell)$	nominal Kalman filter gain matrix
$K_i(k)$	i^{th} subsystem filter gain matrix
L_{ij}	coupling matrix between i^{th} and j^{th} subsystems
$O_n(k, \ell)$	observation matrix
$b(k/k)$	transformed optimum state vector
$e(k)$	residual error of least-squares fit
N_x, N_u	dimensions of x-data equation and u-data equation

Conventions:

a. Superscript T is used to denote transpose of vectors and matrices; $[\cdot]^{-T}$ is used to denote the transpose of the inverse of a matrix.

b. Matrices are denoted by upper case letters. Vectors are denoted by lower case letters except for k, ℓ, m and n which are integers and t denotes time.

c. Vectors are assumed to be columns unless otherwise denoted by superscript T.

d. Unless otherwise specified, the dimension of the state vector is n and the dimension of the measurement vector is m, $n \geq m$.

3.0 ALGORITHM DESCRIPTIONS

A. Covariance Filters

1. Standard Kalman Filter

The "standard" form of the Kalman filter refers to the estimator first given by Kalman [8], from which the discrete optimal filter was derived and subsequently documented in many books [45-51]. The filter algorithm is usually given in two sets of equations - one for extrapolation, the other one for updating:

Extrapolation

$$\hat{x}(k+1/k, \ell) = \Phi(k+1, k)\ \hat{x}(k/k, \ell) \tag{3}$$

$$P(k+1/k, \ell) = \Phi(k, k)\ P(k/k, \ell)\ \Phi^T(k, k) + Q(k) \tag{4}$$

Updating

$$\hat{x}(k+1/k+1, \ell) = \hat{x}(k+1/k, \ell) + K(k+1, \ell)\,[z(k+1) - H(k+1)\hat{x}(k+1/k, \ell)] \tag{5}$$

$$K(k+1, \ell) = P(k+1/k, \ell)\, H^T(k+1)\,[H(k+1)P(k+1/k, \ell)\, H^T(k+1) + R(k+1)]^{-1} \tag{6}$$

$$P(k+1/k+1, \ell) = [I - K(k+1, \ell)\, H(k+1)]\, P(k+1/k, \ell) \tag{7}$$

Note that Eq. (7) is correct only if the gain $K(k+1, \ell)$ is optimum.

2. Stabilized Kalman Filter

The stabilized filter (sometimes called the Joseph algorithm [52]) is less sensitive to computer round-off errors. Another benefit is that it yields correct $P(k+1/k+1, \ell)$ even if $K(k+1, \ell)$ is non-optimum. The updating covariance matrix is given by

$$P(k+1/k+1, \ell) = [I - K(k+1, \ell)\, H(k+1)]\, P(k+1/k, \ell)\, [I - K(k+1, \ell)\, H(k+1)]^T + K(k+1, \ell)\, R(k+1)\, K^T(k+1, \ell) \tag{8}$$

Other updating and extrapolation equations are the same as Eqs. (3) - (6). Initial conditions for the standard as well as the stabilized filters are defined as follows:

$$E\{x(\ell)\} = \hat{x}(\ell/\ell) \tag{9}$$

$$E\{[\hat{x}(\ell) - x(\ell/\ell)]\,[\hat{x}(\ell) - \hat{x}(\ell/\ell)]^T\} = P(\ell, \ell) \tag{10}$$

3. Extended Kalman Filter

The "extended" Kalman filtering is a popular technique for treating nonlinearities in the design of minimum variance estimators. Other methods of the same footing (Taylor series expansion) are Iterated Extended Kalman filtering, Gaussian Second-Order filtering, and Linearized Kalman filtering [46].

Since most physical non'linear systems can be represented by differential equations and measurements are usually available at discrete time, therefore, it is proper as well as convenient (for series expansion) to describe system and measurement models as follows:

$$\dot{x}(t) = f[x(t), t] + u(t) \tag{11}$$

$$z(k) = h[x(t_k)] + v(k) \tag{12}$$

in which $u(t)$ and $v(k)$ are uncorrelated zero-mean Gaussian noise with

$$E[u(t)\, u^T(t)] = Q(t) \tag{13}$$

$$E[v(k)\, v^T(k)] = R(k) \quad ; \tag{14}$$

also the initial vector $x(\ell)$ is Gaussian with mean and covariance given by Eq. (9) and Eq. (10).

Define:

$$F[\hat{x}(t), t] = \left.\frac{\partial f[x(t), t]}{\partial x(t)}\right|_{x(t) = \hat{x}(t)} \tag{15}$$

$$H_L(k) = \left.\frac{\partial h[x(t_k)]}{\partial x(t_k)}\right|_{x(t_k) = \hat{x}(t_k)} \tag{16}$$

Extrapolation equations are given by:

$$\frac{d\hat{x}(t)}{dt} = f[\hat{x}(t), t)] \tag{17}$$

$$\frac{dP(t)}{dt} = F[\hat{x}(t), t]\, P(t) + P(t)\, F^T[\hat{x}(t), t] + Q(t) \tag{18}$$

Update equations are given by:

$$\hat{x}(k+1/k+1, \ell) = \hat{x}(k+1/k, \ell) + K(k+1, \ell)\,[z(k+1) - h[\hat{x}(k+1/k, \ell)]] \tag{19}$$

$$P(k+1/k+1, \ell) = [I - K(k+1)\, H_L(k+1)]\, P(k+1/k, \ell) \tag{20}$$

$$K(k+1, \ell) = P(k+1/k, \ell)\, H_L(k+1)\,[H_L(k+1)\, P(k+1/k, \ell)\, H_L^T(k+1) + R(k+1)]^{-1} \tag{21}$$

Other forms of variation (Gaussian Second-Order, etc.) are given in reference [46].

B. Square Root Covariance Filters

1. Potter

The first square root algorithm was introduced by Potter [11] for a restricted application of zero system noise and measurements are scalar quantities, i.e., Eqs. (1) and (2) become

$$x(k+1) = \phi(k+1,k)\ x(k) \tag{22}$$

$$z(k) = h^T(k)\ x(k) + v(k) \tag{23}$$

where h(k) is a vector and the variance of the measurement noise r(k) is a scalar value. This method, as well as other methods discussed in the sequel, consists of defining a square root matrix S such that

$$P(k/k,\ell) \triangleq S(k/k,\ell)\ S^T(k/k,\ell). \tag{24}$$

The factorization of covariance square roots is generally not unique. However, this lack of uniqueness is not serious because a unique square root factorization can always be obtained by using the Cholesky decomposition technique [53] (sometimes referred to as the Banachiewicz and Dwyer algorithm) which factors any positive semi-definite symmetric matrix into the product of a lower triangular matrix and its transpose. Description of this algorithm can be found in reference [60] as well as in many other sources.

Define:

$$y(k+1,\ell) \triangleq S^T(k+1/k,\ell)\ h(k+1). \tag{25}$$

Consider the case in which Q(k) = 0, then the extrapolation equation for $S(k+1/k,\ell)$ is given by

$$S(k+1/k,\ell) = \phi(k+1,k)\ S(k/k,\ell) \tag{26}$$

The extrapolation as well as updating of the state vector is the same as Eq. (3) and Eq. (5), respectively. Other update equations are given below:

$$S(k+1/k+1,\ell) = S(k+1/k,\ell)\ [I + \alpha(k+1,\ell)\ y(k+1,\ell)\ y^T(k+1,\ell)] \tag{27}$$

$$K(k+1,\ell) = \frac{S(k+1/k,\ell)\ S^T(k+1/k,\ell)\ h(k+1)}{h^T(k+1)\ S(k+1/k,\ell)\ S^T(k+1/k,\ell)\ h(k+1) + r(k+1)} \tag{28}$$

in which $\alpha(k+1,\ell)$ is given by

$$\alpha = \frac{-1 \pm \left(1 - \dfrac{y^T y}{y^T y + r}\right)^{1/2}}{y^T y} \tag{29}$$

In Eq. (29), time index for α, y and r is the same, hence it is being suppressed for clarity. When there is no ambiguity, simplified notations such as Eq. (29) will be used in subsequent discussions.

2. Bellantoni and Dodge

This square root filter is an extension of the Potter algorithm by considering vector measurements (simultaneous) with correlated component errors. This method requires diagonalization of an n x n matrix, i.e.,

$$D \triangleq \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_n \end{bmatrix} = T^T(S^T H^T R H S)T \tag{30}$$

where $s_1, s_2, \ldots, s_n$ are eigenvalues of the covariance matrix and T is the transformation matrix consisting of eigenvectors in its columns. The extrapolation equations are the same as the Potter filter; the update equations were derived in reference [12]:

$$S(k+1/k+1,\ell) = S(k+1/k,\ell)\ \{I + T[(I + D)^{-1/2} - I]\ T^T\} \tag{31}$$

$$K(k+1,\ell) = S(k+1/k,\ell)\ B(I + B^T B)^{-1}\ (R^{1/2})^{-1} \tag{32}$$

where

$$B = S^T(k+1/k,\ell)\ H^T(k+1)\ (R^{1/2})^{-1}. \tag{33}$$

The extrapolation and updating of the system state vector are always the same as Eq. (3) and Eq. (5). As such, subsequent discussions will be concentrated on the covariance and the gain matrices.

3. Andrews; Tapley and Choe; Morf, Levy and Kailath (Continuous-Time Case)

(NOTE: Although this paper is mainly concerned with the discrete-time case, it is felt, however, that a brief discussion of the continuous-time case should be given here for completeness.)

Consider the system model given by

$$\dot{x}(t) = F(t)\ x(t) + G(t)\ u(t) \tag{34}$$

$$z(t) = H(t)\, x(t) + v(t) \tag{35}$$

where u(t) and v(t) are zero mean non-correlated white Gaussian noise with unity covariances. Let P(t) be the covariance of the error of the state estimate, P(t) obeys the following Riccati equations:

$$\dot{P}(t) = F(t)\, P(t) + P(t)\, F^T(t) + G(t)\, G^T(t) - P(t)\, H^T(t)\, H(t)\, P(t) \tag{36}$$

with initial condition:

$$P(o) = P_o \tag{37}$$

The Andrews algorithm (13) was the first Square Root filter that considers the existence of system noise. Similar to the Bellantoni and Dodge algorithm, it also processes vector measurements. However, it does not require any matrix diagonalization. The extrapolation of the square root covariance is given as:

$$\dot{S}(t) = F(k)\, S(t) + [W(t) + 1/2\, G(t)\, Z(t)\, G^T(t)]\, S^{-T}(t)\ , \tag{38}$$

in which Q(t) is the covariance of the system noise and W(t) is a skew symmetric matrix that maintains S(t) in the lower triangular form.

It should be pointed out that solution of Eq. (38) requires that the inverse of $S^T(t)$ be computed at each integration step which requires a large number of multiplications. This is undesirable. Therefore, Tapley and Choe [71] restructured the problem and let the skew symmetric matrix, W(t), be chosen in such a way that inversion of $S^T(t)$ is not needed.

$$[\dot{S}(t) - F(t)\, S(t)]\, S^T(t) = 1/2\, Q(t) + W(t) \tag{39}$$

Furthermore, in order to maintain S(t) in a lower triangular form, the following definitions

$$E(t) \triangleq F(t)\, S(t) \tag{40}$$

$$\bar{G}(t) \triangleq S(t) - E(t) \tag{41}$$

$$\bar{\bar{W}}(t) \triangleq W(t) + 1/2\, Q(t) \tag{42}$$

will be used to rewrite Eq. (39) as

$$\bar{G}(t)\, S^T(t) = \bar{\bar{W}}(t) \tag{43}$$

which is the desired result.

However, since the skew symmetric matrix, W(t), can be chosen arbitrarily, Morf, et al. [72] provided another method for computing the lower-triangular S(t) matrix that is simpler than the method just described above. Let S(t) be nonsingular and define

$$P(t) \triangleq S(t)\, S(t)^T \tag{44}$$

Equation (36) can be written as:

$$\dot{P}(t) = \dot{S}(t)\, S^T(t) + S(t)\, \dot{S}^T(t) \tag{45}$$

$$\dot{P}(t) = (F - 1/2\, SS^T H^T H)\, SS^T + 1/2\, GG^T S^{-T} S^T + SS^T (F^T - 1/2\, H^T H S S^T) + 1/2\, SS^{-1}\, GG^T \tag{46}$$

Multiplying Eq. (45) on the left by S^{-1} and on the right by S^{-T} yields

$$S^{-1}\, \dot{S} + \dot{S}^T\, S^{-T} = L \tag{47}$$

where

$$L \triangleq \bar{F}\, \bar{F}^T + \overline{GG^T} - \bar{H}^T \bar{H} \tag{48}$$

$$\bar{F} = S^{-1}\, F\, S \tag{49}$$

$$\bar{G} = S^{-1}\, G \tag{50}$$

$$\bar{H} = H\, S \tag{51}$$

Since S is lower triangular, $S^{-1}\, \dot{S}$ is the lower-triangular part of L, hence

$$\dot{S} = S\, \bar{L} \tag{52}$$

where $\bar{L}$ is the "lower-traingular part" operator.

Since Eq. (52) does not involve explicit skew symmetric matrix, it seems to be simpler to compute than the other two square root methods discussed above.

4. Schmidt

Instead of using Eq. (34) to extrapolate the square root covariance matrix, Schmidt [14] introduced a method which facilitates digital computations. This algorithm requires finding an orthogonal transformation matrix T, dimension (n+m) x (n+m), such that $T^T T = I$. (n is the dimension of the state vector and m is the dimension of the measurement vector.)

Consider the expression

$$\left[\Phi(k+1,k)\ S(k/k,\ell) \;\vdots\; [Q(k)]^{1/2}\right] T^T T \quad \begin{bmatrix} S^T(k/k,\ell)\,\Phi^T(k+1,k) \\ Q(k)^{1/2} \end{bmatrix} \tag{51}$$

which can be written as

$$\Phi(k+1,k)\ S(k/k,\ell)\ S^T(k/k,\ell)\ \Phi^T(k+1,k) + Q(k). \tag{52}$$

Expression (52) is the right side of Eq. (4). Hence, expression (51) must be the left side of Eq. (4). Therefore, the following relationship is established for the extrapolation of $S(k/k,\ell)$.

$$\begin{matrix} n \\ m \end{matrix} \left\{ \begin{bmatrix} S^T(k+1/k,\ell) \\ \underbrace{0}_{n} \end{bmatrix} \right. = T \left[\begin{matrix} S^T(k/k,\ell)\,\Phi^T(k+1,k) \\ \underbrace{(Q^{1/2}(k+1))^T}_{n} \end{matrix} \right] \begin{matrix} \} n \\ \} m \end{matrix} \tag{53}$$

In order to uniquely express $S^T(k+1/k,\ell)$ in terms of $S^T(k/k,\ell)$, $\Phi(k+i,k)$ and $Q^{1/2}(k+1)$, matrix T must be constructed such that Eq. (53) will be in triangular form. This can be done by using the Gram-Schmidt process or the Householder transformation.

Reference [16] provides descriptions of the Gram-Schmidt and Householder transformations. A more extensive treatment of this subject is found in chapter 5 of reference [60].

5. Carlson

The essence of Carlson's technique is to preserve the square root covariance matrix in triangular form during the extrapolation interval as well as the update time. In addition, Carlson recognized that the transition matrix is often block-triangular, the fact which can be exploited to further reduce computation steps. To preserve $S(k+1/k,\ell)$ in triangular form during extrapolation, two methods are suggested. One is basically the same as Eq. (53), the other is called the Root Sum Square (RSS), which computes the covariance matrix using Eq. (4), then $P(k+1/k,\ell)$ is factored (Cholesky decomposition) into triangular square root matrices $S(k+1/k,\ell)\ S^T(k+1/k,\ell)$. In order to make certain that $S(k+1/k+1,\ell)$ is in triangular form during update, the Potter algorithm is modified by demanding that

$$\bar{A} \triangleq \left[I - \frac{y\,y^T}{y^T y + r} \right]^{1/2} \tag{54}$$

be upper triangular, i.e., for scalar measurements

$$P(k+1/k+1,\ell) = P(k+1/k,\ell) - K(k+1)\ h^T(k+1)\ P(k+1/k,\ell) \tag{55}$$

which can be written as

$$P(k+1/k+1,\ell) = S(k+1/k,\ell)\ S^T(k+1/k,\ell) - \frac{S\,y\,y^T\,S^T}{y^T\,y + r} \tag{56}$$

and factored into

$$P(k+1/k+1,\ell) = S\left[I - \frac{y\,y^T}{y^T y + r} \right] S^T \quad . \tag{57}$$

Hence,

$$S(k+1/k+1,\ell) = S(k+1/k,\ell)\ \bar{A}(k+1) \ , \tag{58}$$

in which $\bar{A}(k+1)$ must be chosen such that $S(k+1/k+1,\ell)$ is also upper triangular. A method that can be used to select and compute the $\bar{A}(k+1)$ matrix is given in reference [20].

C. Information Filters

The Covariance Filter discussed in Section 3A is the Kalman-Bucy filter in its original form (the filter equations are derived from the covariance matrix). The Information Filter discussed in this section is basically of the same footing. However, the filter equations are derived from the inverse of the covariance matrix which is closely related to the information matrix (reference [47], p. 241). The motivation for taking this approach is to avoid computation difficulties in the case where the initial state error covariance $P(\ell,\ell)$ is unknown and assumed to be infinity.

The development of information filter equations is straightforward. This is done by applying the matrix inversion lemma

$$(\Gamma + \Pi^T \Sigma)^{-1} = \Gamma^{-1} - \Gamma^{-1}\ \Pi^T(I + \Sigma\,\Gamma^{-1}\,\Pi^T)^{-1}\ \Sigma\,\Gamma^{-1} \tag{59}$$

to the covariance matrix

$$P(k+1/k,\ell) = \Phi(k+1,k)\ P(k/k,\ell)\ \Phi^T(k+1,k) + Q(k) \tag{60}$$

by identifying

$$\Gamma = \Phi\,P\Phi^T \ ; \ \Pi^T = Q \text{ and } \Sigma = I \ . \tag{61}$$

The propagation of the information matrix was shown [54] to be:

$$P^{-1}(k+1/k,\ell) = F(k) - F(k)[F(k) + Q^{-1}(k)]^{-1} F(k) \tag{62}$$

where

$$F(k) \triangleq [\Phi^{T}(k+1,k)]^{-1} P^{-1}(k/k,\ell)\ \Phi^{-1}(k+1,k). \tag{63}$$

The update of the inverse covariance matrix from $P^{-1}(k/k-1,\ell)$ to $P^{-1}(k/k,\ell)$ is given by

$$P^{-1}(k/k,\ell) = P^{-1}(k/k-1,\ell) + H^{T}(k)\ R^{-1}(k)\ H(k). \tag{64}$$

By defining the state of the information filter as

$$\hat{d}(k/k) \triangleq P^{-1}(k/k,\ell)\ \hat{x}(k/k) \tag{65}$$

$$\hat{d}(k+1/k) \triangleq P^{-1}(k+1/k,\ell)\ \hat{x}(k+1/k). \tag{66}$$

It can be easily shown that the propagation of $\hat{d}(k+1/k)$ is

$$\hat{d}(k+1/k) = [I - P^{-1}(k+1/k,\ell)\ Q(k)]\ \Phi^{-T}(k+1/k)\ \hat{d}(k/k) \tag{67}$$

and the update of $\hat{d}(k/k)$ is

$$\hat{d}(k/k) = \hat{d}(k/k-1) + H^{T}(k)\ R^{-1}(k)\ z(k) \tag{68}$$

It will be shown in the next section that the information filter is more efficient than the covariance filter as far as update is concerned. However, in propagation, the covariance filter requires fewer computations.

Due to computational error, the use of Eqs. (62) and (64) may lead to nonnegative definiteness of $P^{-1}(k+1/k,\ell)$. Once again, this difficulty can be avoided by applying the square root concept, which will be discussed next.

D. Square Root Information Filters

1. Dyer and McReynolds

An efficient square root solution to the least square problem using the Householder algorithm was demonstrated by Golub [55], Businger and Golub [56], and Jordan [57]. Hanson and Lawson [58] extended the theory to include rank deficient systems, and adapted the Householder algorithm to solve sequential least squares problems. Dyer and McReynolds developed the square root information filter based on Householder's matrix triangularization procedure and Cox's [59] sequential estimation algorithm (dynamic programming formulation).

Recall in the Square Root Covariance Filter, it was defined (Eq. (28)):

$$P(k/k,\ell) \triangleq S(k/k,\ell)\ S^{T}(k/k,\ell).$$

For the development of the Square Root Information Filter (SRIF), it is consistent to define:

$$P^{-1}(k/k,\ell) \triangleq S^{-T}(k/k,\ell)\ S^{-1}(k/k,\ell) \tag{69}$$

$$b(k) \triangleq S^{-1}(k/k,\ell)\ \hat{x}(k/k) \tag{70}$$

The update of the inverse covariance square root is given by

$$n \left\{ \left[\begin{array}{c} S^{-1}(k/k,\ell) \\ \hline \underbrace{0}_{n} \end{array} \right] \right. \quad m \left\{ \right. = T \left[\begin{array}{c} S^{-1}(k/k-1,\ell) \\ \hline V^{-1}(k)\ H(k) \end{array} \right] \tag{71}$$

where T is the orthogonal transformation matrix defined previously. The update of b(k/k) is given by

$$n \left\{ \left[\begin{array}{c} b(k/k) \\ \hline e(k) \end{array} \right] \right. \quad m \left\{ \right. = T \left[\begin{array}{c} b(k/k-1) \\ \hline V^{-1}(k) z(k) \end{array} \right] \tag{72}$$

where e(k) is the residual error after processing the measurement. The propagation of the inverse covariance is given by

$$\left[\begin{array}{c|c} W(k+1/k) & G(k+1/k) \\ \hline 0 & S^{-1}(k+1/k,\ell) \end{array} \right] = T \left[\begin{array}{c|c} U^{-1}(k) & 0 \\ \hline S^{-1}(k/k,\ell)\ \Phi^{-1}(k/k) & S^{-1}(k/k,\ell)\ \Phi^{-1}(k/k) \end{array} \right] \tag{73}$$

where

$$G(k+1/k) \triangleq [R(k+1) + \Gamma^{T}(k+1/k)\ \Gamma(k+1/k)]^{1/2}. \tag{74}$$

The propagation of b(k+1/k) is given by

$$1 \left\{ \left[\begin{array}{c} \varepsilon(k+1) \\ \hline b(k+1/k) \end{array} \right] \right. \quad n \left\{ \right. = T \left[\begin{array}{c} 0 \\ \hline b(k/k) \end{array} \right] \tag{75}$$

where

$$a^{-1}(k) = C^T(k)\, C(k) + Q^{-1}(k) \tag{76}$$

$$C(k) = S^{-1}(k/k, \ell)\ \Phi^{-1}(k/k) \quad . \tag{77}$$

A different form of propagating $S^{-1}(k+1/k, \ell)$ and $b(k+1/k)$ is also available:

$$S^{-1}(k+1/k, \ell) = (I - [1 + \{a(k)\, Q^{-1}(k)\}^{1/2}]^{-1}\, C(k)\, a(k)\, C^T(k))\ S^{-1}(k/k, \ell)\ \Phi^{-1}(k/k) \tag{78}$$

$$b(k+1/k) = (I - a(k)\, [1 + \{a(k)\, Q^{-1}(k)\}^{1/2}]^{-1}\, C(k)\, C^T(k))\ b(k/k) \tag{79}$$

2. Bierman (G.J.)

It should be apparent from the previous section that while the Dyer-McReynolds SRIF is attractive, it relies heavily upon the Householder transformation as well as relying on the concept of dynamic programming, which seems to be a little too abstract and difficult to understand. For this reason, Bierman [22] introduced the recursive least-square approach intended to simplify the basic structure of SRIF. In essence, Bierman's square root data processing method utilized the so-called "data equation" and the sum-of-squares performance functional to develop equations that propagate the state estimate and its error covariance. Eqs. (1) and (2) are considered as "measurement equations" and Eqs. (80) and (81) below are considered to be a priori "data equations" associated with Eqs. (1) and (2), respectively:

$$z_u(k) = R_u(k)\, u(k) + w_u(k) \tag{80}$$

$$z_x(k) = R_x(k)\, X(k) + w_x(k) \tag{81}$$

where w_u and w_x are assumed to be zero mean, independent random processes with unity covariances.

Define:

$$Q(k) \triangleq R_u^T(k)\, R_u(k)$$

$$P_x(o) \triangleq R_x^{-T}(o)\, R_x^{-1}(o) \quad .$$

By selecting the performance functional to be

$$J(k+1) = \| R_x(o)\, x(o) - z_x(o) \|^2 + \sum_{i=o}^{k} \left(\| H(i)\, x(i) - z(i) \|^2 + \| R_u(i)\, u(i) - z_u(i) \|^2 \right) , \tag{82}$$

the problem is to minimize $J(k+1)$ with respect to $x(i)$ and $u(i)$ for $i = 0, 1, 2, \ldots, k$, such that the solution yields the optimal estimate of $x(k)$.

Bierman [60] has shown that the following "Information Arrays" contain all necessary information needed for state and covariance update as well as propagation. The actual data processing requires a transformation and update (mapping) given by Eq. (83) and Eq. (84), respectively:

$$\hat{T}(k) \begin{array}{c|c|c} R_x(k) & z_x(k) & \\ \hline H(k) & z(k) & \end{array} = \begin{array}{c|c|c} \hat{R}_x(k) & \hat{z}_x(k) & \} N_u \\ \hline 0 & e(k) & \} N_x \\ N_u & 1 & \end{array} \tag{83}$$

$$T(k+1) \begin{array}{c|c|c} R_u(k) & 0 & z_u(k) \\ \hline -R_x(k)\, \Phi^{-1}(k+1) & R_x(k)\, \Phi^{-1}(k+1) & \hat{z}_x(k) \end{array} = \begin{array}{c|c|c} R_u(k+1) & R_{ux}(k+1) & z_u(k+1) \\ \hline 0 & R_x(k+1) & z_x(k+1) \end{array} \tag{84}$$

$$\begin{matrix} N_u & N_x & 1 \end{matrix}$$

in which N_x and N_u are dimensions of the $x(k)$ and $u(k)$, respectively; $e(k)$ is the error in the least square fit, and $\hat{T}(k)$ and $T(k+1)$ are products of N_u elementary Householder transformations. Definitions of other symbols are given in Section 2.

The update estimate and covariance are

$$\hat{x}(k+1) = R_x^{-1}(k+1)\, z_x(k+1) \tag{85}$$

$$P_x(k+1) = R_x^{-1}(k+1)\, R_x^{-T}(k+1) \tag{86}$$

The propagation of state vector requires solution of $\hat{x}(k)$ and $\hat{x}(k+1)$ (i.e., $u(k)$ and $x(k+1)$ form an augmented data equation $\begin{bmatrix} u(k) \\ x(k+1) \end{bmatrix}$).

$$R_u(k+1)\, u(k) + R_{ux}(k+1)\, x(k+1) = z_u(k+1) - w_u(k) \tag{87}$$

$$R_x(k+1)\, x(k+1) = z_x(k+1) - w_x(k) \tag{88}$$

which can be solved using the Gaussian elimination method.

E. Factorized Filters

During the past decade, a number of authors [25, 26, 60-66] have contributed to improving the Kalman filtering computation efficiency by suggesting square-root-free triangular factorizations. Essentially, this approach is based on the rank one modification of the Cholesky method. For example, Agee and Turner [61] have proved that for a positive definite covariance matrix P such that $P = UDU^T$, in which U is a unit upper triangular matrix and D is a diagonal matrix with elements $d_1, d_2, \ldots, d_n$. Where n is the dimension of P, there exists an update P(k) matrix such that

$$P(k) = \bar{U}(k)\,\bar{D}(k)\,\bar{U}^T(k) = U(k)\,D(k)\,U^T(k) + cv(k)\,v^T(k)\ , \tag{89}$$

where c is a scalar. v(k) is a vector of n-dimension. If P(k) is positive definite, then $\bar{U}(k)$ and $\bar{D}(k)$ can be computed as follows:

For j = n, n-1, ..., 2, recursively compute $d_j(k)$ and $u_{ij}(k)$, which are elements of $\bar{D}(k)$ and $\bar{U}(k)$:

$$d_j(k+1) = d_j(k) + c_j\,v_j^2(k) \tag{90}$$

$$v_i(k) \leftarrow v_i(k) - v_j(k)\,u_{ij}(k) \qquad i = 1, \text{---} j-1 \tag{91}$$

$$u_{ij}(k+1) = u_{ij}(k) + c_j\,v_j(k)\,v_i(k)/d_j(k+1) \qquad i = 1, \text{---} j-1 \tag{92}$$

$$c_{j-1} = c_j\,d_j(k) \,/\, d_j(k+1) \tag{93}$$

The notation $\leftarrow$ is used for "replacement" in the FORTRAN implementation. Detailed proof is given in reference [60] (p. 45). This method of calculating D(k) and U(k) is generally valid for both covariance filters as well as information filter updates. To illustrate the usefulness of this factorized approach, consider the covariance update (Kalman filter):

$$P(k+1/k+1, \ell) = P(k+1\,/k, \ell) - P(k+1/k, \ell)\,H^T(k)\,H(k)\,P(k+1/k, \ell)\,H^T(k)\,H(k)\,P(k+1/k, \ell). \tag{94}$$

By assuming scalar measurement, (94) can be factored into:

$$U(k+1/k+1)\,D(k+1/k+1)\,U^T(k+1/k+1) = U(k+1/k)[D(k+1/k) - (\tfrac{1}{a})\,V(k+1/k)\,V^T(k+1/k)]\,U^T(k+1/k) \tag{95}$$

where

$$V(k+1/k) = D(k+1/k)\,U^T(k+1/k)\,H^T(k+1)\ , \tag{96}$$

and the scalar "a" is given by:

$$a = H(k+1)\,P(k+1/k)\,H^T(k+1) + R(k+1)\ . \tag{97}$$

Using Eq. (89), let

$$\bar{U}(k+1/k)\,\bar{D}(k+1/k)\,\bar{U}^T(k+1/k) = [D(k+1/k) - \tfrac{1}{a}\,V(k+1/k)\,V^T(k+1/k)]. \tag{98}$$

Then Eq. (95) can be written as

$$U(k+1/k+1)\,D(k+1/k+1)\,U^T(k+1/k+1) = [U(k+1/k)\,\bar{U}(k+1/k)]\,\bar{D}(k+1/k)[U(k+1/k)\,\bar{U}(k+1/k)]^T\ . \tag{99}$$

Since U(k+1/k) and $\bar{U}(k+1/k)$ are unit upper triangular, Eq. (99) yields:

$$U(k+1/k+1) = U(k+1/k)\,\bar{U}(k+1/k) \tag{100}$$

$$D(k+1/k+1) = \bar{D}(k+1/k)\ . \tag{101}$$

The above results show that the problem of factoring the filter update covariance has been reduced to the task of factoring a symmetric matrix $[D(k+1/k) - \frac{1}{a}\,V(k+1/k)\,V^T(k+1/k)]$ into $\bar{U}(k+1/k)$ and $\bar{D}(k+1/k)$.

Now, consider the covariance propagation

$$P(k+1/k) = \phi(k+1,k)\,P(k/k)\,\phi^T(k+1,k) + Q(k). \tag{102}$$

It is required to find $[U(k+1/k)\,D(k+1/k)\,U^T(k+1/k)]$ such that it is equal to the right-hand side of Eq. (102). Without loss of generality, let Q(k) be a diagonal matrix and let

$$\tilde{U}(k+1/k) \triangleq [\,\phi(k+1,k)\,U(k/k)\,\vdots\,I\,] \tag{103}$$

$$\tilde{D}(k+1/k) \triangleq \begin{bmatrix} D(k/k) & \vdots & 0 \\ \text{---} & \text{---} & \text{---} \\ 0 & \vdots & Q(k) \end{bmatrix}\ . \tag{104}$$

Then it can be shown that

$$G(k+1/k)\,\tilde{D}(k+1/k)\,G^T(k+1/k) = \phi(k+1,k)\,U(k/k)\,D(k/k)\,U^T(k/k)\,\phi^T(k+1,k) + Q(k)\ , \tag{105}$$

which is the desired result. Eq. (103) can be efficiently computed using the modified Gram-Schmidt method, the Householder transformation, or the Givens transformation.

F. Chandrasekhar Filters

An approach to minimize the computer burden was introduced by Kailath [27] who considered a special case of continuous-time stationary process and showed that differential equations of the Chandrasekhar type instead of the matrix Riccati differential equation can be used to compute the filter gain. This development was immediately followed by Linquist [28] who treated the same problem by means of the backward innovation process. Morf et al. [67, 69] and Linquist [29] have solved the discrete-time stationary process problem; Friedlander et al. [68] have treated the discrete-time nonstationary process problem by introducing a way of classifying stochastic processes in terms of an "index of nonstationarity". It was shown that Chandrasekhar equations can be derived from the extended Levinson-Whittle-Wiggins-Robinson algorithms for stationary time-series; Lainiotis [30] has provided generalized algorithms for the continuous-time nonstationary as well as stationary processes.

Approaching from the square-root algorithm viewpoint (i.e., propagation of the square root of $\frac{d}{dt} P(t)$ instead of the square root of P(t)), Morf et al. [70] derived the continuous-time Chandrasekhar filter equations which are identical to results given in reference [30], in which the "Partitioning Formulas" of Lainiotis were used.

For the purpose of comparing digital computational efficiency, only the discrete-time Chandrasekhar algorithm is described in the paper, since it was pointed out in reference [70] that the number of computing operations is approximately equal for various versions of Chandrasekhar filters.

By considering constant matrices Φ, H, P_o, Q and R associated with a system model given in (1) and (2), reference [29] presented the following results, from which the optimal filter gain matrix K(k) can be determined in the following manner:

$$K(k) = \Phi A(k)[HA(k) + R]^{-1} \tag{106}$$

where

$$A(k) = A(k-1) - A'(k-1)\, C^{-1}(k-1)\, A'^{T}(k-1)\, H^{T} \tag{107}$$

$$A'(k) = \Phi A'(k-1) - \Phi A(k-1)[HA(k-1) + R]^{-1} HA'(k-1) \tag{108}$$

$$C(k) = C(k-1) - A'^{T}(k-1)\, H^{T}(HA(k-1) + R)^{-1} HA'(k-1)\ , \tag{109}$$

with initial considerations:

$$A(o) = P_o H^{T} \tag{110}$$

$$A'(1) = \Phi P_o H^{T} \tag{111}$$

$$C(o) = H P_o H^{T} + R\ . \tag{112}$$

Note that A(k) and A'(k) are n x m matrices and C(k) is a symmetric m x m matrix. Thus, in order to solve for A(k) only 2 nm + [m(m+1)]/2 equations are needed, as in contrast to the conventional Kalman algorithm in which n^2 equations are required to compute the filter gain. If m << n, which is true in many practical situations, the number of equations to be solved in each step is of order n versus n^2. Since only the inverse of C(k) is needed in Eq. (107), Eq. (109) may be replaced by

$$C^{-1}(k) = C^{-1}(k-1) + C^{-1}(k-1)\, A'^{T}(k-1)\, H^{T}(HA(k) + R)^{-1} HA'(k-1)\, C^{-1}(k-1)\ , \tag{113}$$

which can be obtained by applying the matrix inversion lemma to Eq. (109).

The above results were also extended to the continuous-time case by Linquist [28] and they were shown to be exactly corresponding to the equations derived by Kailath [27].

G. Partitioning Filters

In a radically different approach to filtering and estimation in general, Lainiotis [32-38, 72-88] has developed the Partitioning Algorithms, which are fundamentally new techniques never explored before. The partitioning approach yields new results for linear as well as nonlinear estimation in naturally decoupled, computationally attractive and fast parallel-processing realizations. The partitioned filter contains the Kalman filter as a special case and it constitutes the natural framework for efficient change of initial conditions <u>without recourse to reprocessing the data</u>. This special property will lead to efficient computations. Several partitioning algorithms for discrete-time linear systems are given below.

1. General Partitioned Algorithm (GPA)

In the filtering problem stated in Section 2, solution provided by the Partitioning approach consists of decomposing the initial state vector x(0) into the sum of two independent vectors:

$$x(0) = x_n + x_r \tag{114}$$

where x_r is an unknown model parameter vector to be adapted (references [32, 33, 74]). Let $\hat{x}_n$ and P_n be the mean and covariance of x_n respectively (the choice of $\hat{x}_n$ and P_n is arbitrary). Since x_n and x_r are assumed to be independent, the following initial conditions relationship hold:

$$\hat{x}(1/0) = \hat{x}_n(0) + \hat{x}_r(0) \tag{115}$$

$$P(\ell,\ell) = P_n(\ell) + P_r(\ell) \tag{116}$$

The optimal filtered estimate and the corresponding error-covariance matrix $P(k,\ell)$ are given by

$$\hat{x}(k/k,\ell) = \hat{x}_n(k/k,\ell) + \hat{x}_r(k/k,\ell) \tag{117}$$

$$P(k,\ell) = P_n(k,\ell) + P_r(k,\ell) \qquad \text{for } k \geq \ell \tag{118}$$

where $\hat{x}_n(k/k,\ell)$ and $P_n(k,\ell)$ are computed using the standard Kalman filter equations with initial conditions $\hat{x}((\ell/\ell,\ell) = \hat{x}_n$ and $P(\ell,\ell) = P(\ell)$. The remainder estimate $\hat{x}_r(k/k,\ell)$ and the corresponding error covariance matrix are given by

$$\hat{x}_r(k/k,\ell) = \phi_n(k,\ell)\,\hat{x}_r(\ell/k,\ell) \tag{119}$$

$$P_r(k/\ell) = \phi_n(k,\ell)\,P_r(\ell/k)\,\phi_n^T(k,\ell) \tag{120}$$

where $\hat{x}_r(\ell/k,\ell)$ and $P_r(\ell/k)$ are the smoothed estimate of the partial initial state $x_r(\ell)$ and its covariance matrix respectively. They are given by

$$\hat{x}_r(\ell/k,\ell) = P_r(\ell/k)[M_n(k,\ell) + P_-^{-1}(\ell)\,\hat{x}_r(\ell)] \tag{121}$$

$$P_r(\ell/k) = [P_r(\ell)\,O_n(k,\ell) + I]^{-1}\,P_r(\ell) \tag{122}$$

where

$$M_n(k,\ell) = M_n(k-1,\ell) + \phi_n^T(k-1,\ell)\,\phi^T(k,k-1)\,H^T(k)\,P_{\tilde{z}_n}^{-1}(k/k-1,\ell) \tag{123}$$

$$\hat{z}_n(k,\ell)\,O_n(k,\ell) = O_n(k-1,\ell)\,\phi_m^T(k-1,\ell)\,\phi^T(k,k-1)\,H^T(k)\,P_{\tilde{z}_n}^{-1}(k/k,\ell)\,H(k)\,\phi(k,k-1)\,\phi_n(k-1,\ell) \tag{124}$$

$$\phi_m(k,k-1) = [I - K_n(k,\ell)\,H(k)]\,\phi(k,k-1) \tag{125}$$

$$\tilde{z}_c(k,\ell) = z(k) - H(k)\,\phi(k,k-1)\,\hat{x}_n(k-1/k-1,\ell) \tag{126}$$

$$P_{\tilde{z}_n}(k/k-1,\ell) = H(k)\,P_n(k/k-1,\ell)\,H^T(k) + R(k) \tag{127}$$

$$K_n(k,\ell) = P_n(k/k-1,\ell)\,H^T(k)\,P_{\tilde{z}_n}^{-1}(k/k-1,\ell)\,. \tag{128}$$

The GPA given above (Eqs. (117-128)) constitutes a family of realizations of the optimal linear filter, one for each initial-state-vector partitioning. For example, the Kalman filter is a member of this family for nominal initial conditions equal to actual initial conditions, namely $\hat{x}_r = 0$ and $P_r(\ell) = [0]$. Unlike the Kalman filter, GPA is applicable to all initial conditions including $P(\ell,\ell) = \infty$. With the freedom of choosing nominal initial conditions, GPA is closely related to the Chandrasekhar realization of the Kalman filter algorithm. Specifically, the computational advantages of the Chandrasekhar algorithm depends on the low-rank property of the actual initial conditions.

The basic approach of GPA is to decompose the initial state vector into the sum of <u>two statistically independent</u> Gaussian random vectors (Eq. (114)). The natural extension of this concept is to consider the decomposition of the initial state into the sum of an arbitrary number of jointly Gaussian random vectors which may be <u>statistically dependent</u>. Indeed this concept has been developed by Lainiotis and Andrisani II [88] into the so-called "multipartitioning" algorithm, which can be used for, among other applications, efficient parameter identifications and filtered state estimate of off-diagonal terms in the initial-state covariance matrix.

2. <u>Lambda Algorithm</u>

The Lambda algorithm has a decoupled structure which results from partitioning of the total data interval into nonoverlapping subintervals. Elemental filtering solutions are first computed in each subinterval with arbitrarily chosen nominal initial conditions. Then the overall solution is obtained by connecting the elemental piecewise solutions via GPA. Thus, the desired estimation results over the entire interval has been decomposed into a set of completely decoupled elemental solutions which can be computed in either a serial or parallel-processing mode.

Let the data interval consist of measurements $\lambda_n = \{z(0), z(1), \ldots, z(n)\}$, where $z(k) \triangleq z(t_k)$ and $t_o \leq t_k \leq t_n$. Given data interval $[t_o, t_n]$ to be divided into nonoverlapping subintervals $[t_i, t_j]$, the Lambda algorithm is given by

$$\hat{x}(j,o) = \hat{x}_n(j,i) + \hat{x}_r(j,o) \tag{129}$$

$$P(j,o) = P_n(j,i) + P_r(j,o) \tag{130}$$

where

$$\hat{x}(j,o) \triangleq \hat{x}(t_j/t_j,t_o) \tag{131}$$

$$P(j,o) \triangleq P(t_j,t_o) \tag{132}$$

are the optimal estimate and its covariance matrix at t_j with initial conditions $x(t_o/t_o)$ and $P(t_o,t_o)$ at t_o. The nominal quantities

$$\hat{x}_n(j,i) \triangleq \hat{x}_n(t_j/t_j,t_i) \tag{133}$$

$$P_n(j,i) \triangleq P_n(t_j, t_i) \tag{134}$$

are the nominal estimate of the corresponding nominal covariance at t_j, obtained using the standerd Kalman filtering equations with initial conditions at t_i given by

$$\hat{x}_n(i,i) \triangleq \hat{x}_n(t_i) \tag{135}$$

$$P_n(i,i) \triangleq P_n(t_i) \ . \tag{136}$$

The remainder initial conditions are given by

$$\hat{x}_r(i,o) \triangleq \hat{x}(i,o) - \hat{x}_n(i,i) \tag{137}$$

$$P_r(i,o) \triangleq P(i,o) - P_n(i,i) \ . \tag{138}$$

The remainder smoothed estimate $x_r(j,o)$ and its covariance matrix $P_r(j,o)$ are given by Eqs. (119-128) with the following proper identifications: $t_k = t_j$, $t_l = t_i$, $\hat{x}_r(l) = \hat{x}_r(i,o)$ and $P_r(l) = P_r(i,o)$.

To show the recursive nature of the Lambda algorithm, equations (129-130) and the present version of equations (121-122) may be combined to yield:

$$\hat{x}(j,o) = \hat{x}_n(j,i) + \phi_n(j,i)[P_r(i,o)\ O_n(j,i) + I]\ [P_r(i,o)\ H_n(j,i) + \hat{x}_r(i,o)] \tag{139}$$

$$P(j,o) = P_n(j,i) + \phi_n(j,i)[P_r(i,o)\ O_n(j,i) + I]\ P_r(i,o)\ \phi_n^T(j,i) \tag{140}$$

where $O_n(j,i) \triangleq O_n(t_j,t_i)$ and $H_n(j,i) \triangleq H_n(t_j,t_i)$ are obtained from equations (123-125) for the subinterval $[t_i,t_j]$. The recursive operations are repeated for each subinterval until $t_j = t_n$.

3. Delta Algorithm

The Delta algorithm is developed based on "doubling" the length of the partitioning interval of the Lambda algorithm. This development was motivated by applications of the Partitioned filter to time-invariant models, in which the number of iterations needed to reach steady state depends on the time constants of the model.

The "doubling" algorithm, known to be faster than the Chandrasekhar algorithm is given [79] by the following recursive equations:

$$P(2^{n+1}\Delta) = P_n(2^n\Delta) + \phi_n(2^n\Delta)[P(2^n\Delta)\ O_n(2^n\Delta)]^{-1}\ P_n(2^n\Delta)\ \phi_n^T(2^n\Delta) \tag{141}$$

where $P_n(\cdot)$, $\phi_n(\cdot)$, and $O_n(\cdot)$ are given by $P_n(2^{n+1})$,

$$P_n(2^{n+1}\Delta) = P_n(2^n\Delta) + \phi_n(2^n\Delta)[P_n(2^n\Delta)\ O_n(2^n\Delta) + I]^{-1}\ P_n(2^n\Delta)\ \phi_n^T(2^n\Delta) \tag{142}$$

$$\phi_n(2^{n+1}\Delta) = \phi_n(2^n\Delta)[P_n(2^n\Delta)\ O_n(2^n\Delta) + I]^{-1}\ \phi_n(2^n\Delta) \tag{143}$$

$$O_n(2^n\Delta) = O_n(2^n\Delta) + \phi_n^T(2^n\Delta)[P_n(2^n\Delta)\ O_n(2^n\Delta) + I]\ P_n(2^n\Delta)\ \phi_n(2^n\Delta) \tag{144}$$

for $n = 0, 1, 2, 3, \ldots$

It can be seen that matrix inversions which are required to obtain the Riccati solution at the end of a time interval which is twice as long as the interval in the previous iteration, i.e., doubling.

4. Per-Sample Partitioning

The Per-Sample partitioning is another extension of the basic Lambda algorithm, in which partitioning is done at every sample with zero nominal initial conditions, i.e., $\hat{x}_n(k) = 0$ and $P_n(k) = 0$ for $k = 1, 2, \ldots, N$. This ultimate partitioning of the data interval at every sampling instant yields completely decoupled linear estimation from sample to sample, thus resulting in a simple recursive algorithm, which was given [35, 37,75] by the following recursive equations:

$$\hat{x}(k+1,0) = \hat{x}_n(k+1,k) + \phi_n(k+1,k)[P(k,0)\ O_n(k+1,k) + I]^{-1}\ [P(k,0)\ H_n(k+1,k) + \hat{x}(k,0)] \tag{145}$$

$$P(k+1,0) = P_n(k+1,k) + \phi_n(k+1,k)[P(k,0)\ O_n(k+1,k) + I]^{-1}\ P(k,0)\ \phi_n^T(k+1,k) \tag{146}$$

where

$$H_n(k+1,k) = \phi^T(k+1,k)\ H^T(k+1)\ A(k+1)\ z(k+1) \tag{147}$$

$$O_n(k+1,k) = \phi^T(k+1,k)\, H^T(k+1)\, A(k+1)\, H(k+1)\, \phi(k+1,k) \tag{148}$$

and the Per-Sample nominal filter equations are as follows:

$$\hat{x}_n(k+1,k) = k_n(k+1,k)\, z(k+1) \tag{149}$$

$$P_n(k+1,k) = [I - k_n(k+1,k)\, H(k+1)]\, Q(k) \tag{150}$$

$$k_n(k+1,k) = Q(k)\, H^T(k+1)\, A(k+1) \tag{151}$$

where

$$A(k+1) = [H(k+1)\, Q(k)\, H^T(k+1) + R(k+1)]^{-1} . \tag{152}$$

The Per-Sample partitioning filter is memoryless as it can be seen from Eqs. (147-152), since all memory has been transferred to the basic partition filter equation (145). It is observed that computation of the nominal filter gain $k_n(k+1,k)$ is accomplished using Eq. (151) and Eq. (152), without repeated use of the Riccati equation as is required in the Kalman filter computation.

It is also observed that all other quantities, M_n, O_n, x_n, P_n and ϕ_n, are completely determined non-recursively by Eqs. (147-152), using only the model quantities and the data at the current time (t_k+1).

The remarkable nature of the Per-Sample partitioned filter and its computational advantages can be seen further by considering the case of time-invariant models. For time-invariant models, the recursive algorithm is exactly as given in Eqs. (145-152), except that now all relevant quantities are time-invariant. Specifically, the Per-Sample partitioned filter is now given [35, 37, 75] as follows:

Per-Sample Partitioned Filter

$$\hat{x}(k+1,0) = \hat{x}_n(k+1,k) + \phi_n[P(k,0)\, O_n + I]^{-1}\, [P(k,0)\, M_n(k+1,k) + \hat{x}(k,0)] \tag{153}$$

where

$$M_n(k+1,k) = \phi^T H^T A\, z(k+1) \tag{154}$$

$$O_n = \phi_n^T H^T A H \phi_n \tag{155}$$

$$\hat{x}_n(k+1,k) = k_n\, z(k+1) \tag{156}$$

$$P_n = [I - k_n H]\, Q \tag{157}$$

$$k_n = Q\, H^T A \tag{158}$$

and

$$A = [H\, Q\, H^T + R]^{-1} . \tag{159}$$

To fully appreciate how interesting the above version of the partitioned filter is, it must be noted that both the Kalman filter realization and the Chandrasekhar realization of the optimal linear estimate result in filter algorithms that are time-varying even for time-invariant models. Namely, both the Kalman and Chandrasekhar realizations are time-varying filters in the transient stage even for time-invariant models, yet it is seen that the above Per-Sample partitioning realization of the optimal filter for time-invariant models is a completely time-invariant one even from the first iteration. This transformation of a basically time-varying filter (at least at the transient stage) into an effectively time-invariant one is due to partitioning and the zero initialization at each sampling instant, which lead to completely decoupled and memoryless nominal filters.

It is noted further then, in view of the time invariance of all the relevant quantities, namely O_n, k_n, ϕ_n and P_n, that they only need to be computed once, and stored for subsequent use.

H. Decentralized Filters

During the last few years, a novel approach used for state estimation of large-scale systems has been decentralization. As a result, a number of decentralized filters have been developed. Two of which, addressing the general problem (interconnections exist in all subsystems as well as in the measurement), are the so-called Surely Locally Unbiased (SLU) [39-43], and Sequentially Partitioned Algorithm (SPA) filters. The attractive features of these algorithms are filter stability and computation efficiency. These properties for the continuous-time SLU formulation have been demonstrated by Sanders et al. [43]. The discussion of the discrete-time decentralized filters is given by Kerr and Chin in Part I of this volume. For self-contained purposes, a brief summary of these algorithms is given below.

1. The Surely Locally Unbiased Filter

Consider the following collection $\{S_i,\ i=1,2,\ldots,N\}$ of N interconnected subsystems:

$$x_i(k+1) = \phi_{ii}(k+1,k)\, x_i(k) + L_{ii} \sum_{\substack{j=1 \\ i \neq j}}^{N} L_{ij}\, x_j(k) + w_i(k) . \tag{160}$$

With measurement at S_i ,

$$z_i(k) = \bar{H}_i(k)\, x_i(k) + \hat{H}_i(k) \sum_{\substack{j=1 \\ i \neq j}}^{N} L_{ij}\, x_j(k) + v_i(k) \tag{161}$$

$\hat{H}_i(k)$ is a matrix of rank $p_i < q_i$, and has the physical interpretation that the local decision maker at S_i can observe all sybsystem interactions. $\bar{H}_i(k)$ is the local state observation matrix. The problem is to find N decentralized filter gains of the specific SLU class to minimize a global cost function. The approach is, first, to decouple into N local minimization of the constituent cost functions for the N local subsystems, then apply the discrete-time Matrix Minimum Principle to solve for the gain of each local subsystem. This procedure yields recursive computations.

The optimal state estimate is given by

$$\hat{x}_i(k+1/k) = \phi_{ii}(k+1,k)\, \hat{x}_i(k/k-1) + L'_{ii}(z'_{i1} - \bar{H}'_{i1}\, \hat{x}_i) + K'_{i2}(z'_{i2} - \bar{H}'_{i2}\, \hat{x}_i) \tag{162}$$

where

$$\begin{matrix} p_i \\ q_i \end{matrix} \left\{ \begin{bmatrix} z'_{i1} \\ \hline z_{i2} \end{bmatrix} \right. \triangleq u_2^{-1} z_i \tag{163}$$

$$p_i < q_i$$

$p_i \times q_i$ is the dimension of the i^{th} local measurement

$$u_2^{-1} \hat{H}_i\, u_1 \triangleq \left[\begin{array}{c} I_{p_i} \\ \hline 0 \end{array} \right] \Big\} \; q_i \tag{164}$$

$$L_{ii}\, u_1 \triangleq L'_{ii} \tag{165}$$

$$u_1^{-1} L_i \triangleq L'_i \tag{166}$$

$$u_1^{-1} v_i \triangleq v_i \triangleq \left[\begin{array}{c} v_{i1} \\ \hline v_{i2} \end{array} \right] \tag{167}$$

$$u_2^{-1} \bar{H}_i \triangleq \bar{H}'_i \triangleq \left[\begin{array}{c} \bar{H}'_{i1} \\ \hline \bar{H}_{i2} \end{array} \right] \Big\} \begin{matrix} p_i \\ q_i - p_i \end{matrix} \tag{168}$$

$$u_2^{-1} R_i\, u_2^{-T} \triangleq R'_i \triangleq \left[\begin{array}{c|c} R'_{i1} & R'_{i3} \\ \hline R_{i3} & R_{i2} \end{array} \right] \Big\} \; q_i - p_i \tag{169}$$

$$K_i\, u_2 \triangleq k'_i \triangleq \left[K'_{i1} \;\vdots\; K'_{i2} \right] \tag{170}$$

$$K'_{i2}(k) = \left[\phi_{ii}(k+1,k)\, P_i(k/k-1)\, \bar{H}'^{,T}_{i2}(k) - L'_{ii}(k) \left[\bar{H}'_{i1}(k)\, P_i(k/k-1)\, \bar{H}'^{,T}_{i2}(k) + R'_{i3}(k) \right] \right]$$

$$\cdot \left[\bar{H}'_{i2}(k)\, P_i(k/k-1)\, \bar{H}'^{,T}_{i2}(k) + R'_{i2}(k) \right]^{-1} \tag{171}$$

$$P_i(k+1/k) = \left[\phi_{ii}(k+1,k) - L'_{ii}(k)\, \bar{H}'_{i1}(k) \right] P_i(k/k-1) \left[\phi_{ii}(k+1,k) - L_{ii}(k)\, \bar{H}'_{i1} \right]^T$$

$$- \left[\left\{ \phi_{ii}(k+1,k) - L'_{ii}(k)\, \bar{H}'_{i1}(k) \right\} P_i(k/k-1)\, \bar{H}'^{,T}_{i2}(k) - L'_{ii}(k)\, R'_{i3}(k) \right]$$

$$\cdot \left[\bar{H}'_{i2}(k)\, P_i(k/k-1)\, \bar{H}'^{,T}_{i2}(k) + R_{i2}(k) \right]^{-1}$$

$$\cdot \left[\left\{ \phi_{ii}(k+1,k) - L'_{ii}(k)\, \bar{H}'_{i1}(k) \right\} P_i(k/k-1)\, \bar{H}'^{,T}_{i2}(k) - L'_{ii}(k)\, R'_{i3}(k) \right]^T$$

$$+ \left[\bar{Q}_i(k) + L'_{ii}(k)\, R'_{i1}(k)\, L^T_{ii}(k) \right] \tag{172}$$

$$\bar{Q}_i(k) \triangleq Q_i(k) - L'_{ii}(k)\, R'_{i1}(k)\, L'^{,T}_{ii}(k) \tag{173}$$

Equation (172) describes how the variance in estimation error evolves in discrete-time. The SLU filter treats the interaction input to each local subsystem as if it were just a zero-mean, Gaussian white noise, but of the appropriately adjusted covariance.

Although it is necessary to find $u_1(k)$ and $u_2(k)$ at each measurement, calculation of their inverses is not needed because they are imbedded in the decentralized filtering algorithm.

2. The Sequentially Partitioned Algorithm

Another formulation of the decentralized filter is given by Shah [90]. The so-called SPA filter can be summarized below. Consider the following subsystem and measurement equations:

$$x_i(k+1) = \bar{\Phi}_{ii}(k+1,k)\, x_i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N} \bar{\Phi}_{ij}(k+1,k)\, x_j(k) + w_i(k) \tag{174}$$

$$z_i(k) = \bar{H}_i(k)\, x_i(k) + \hat{H}_i(k)\, L_i(k)\, \underline{x}(k) + v_i(k) \tag{175}$$

$$= \bar{H}_i(k)\, x_i(k) + \hat{H}_i(k) \sum_{\substack{j=1 \\ j\neq i}}^{N} L_{ij}(k)\, x_j(k) + v_i(k) \tag{176}$$

for $i = 1, 2, \ldots, N$.

The estimation error is defined in the usual manner:

$$e_i(k/k) \triangleq x_i(k) - \hat{x}_i(k/k) \tag{177}$$

where

$$\hat{x}_i(k/k) = E\{x_i(k) / Z(k)\} \tag{178}$$

in which $Z(k)$ is the measurement set. By combining Eqs. (174) and (177), the i^{th} subsystem may be written as:

$$x_i(k+1) = \bar{\Phi}_{ii}(k+1,k)\, x_i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N} \bar{\Phi}_{ij}(k+1,k)\, \hat{x}_j(k) + w_i^*(k)\ , \tag{179}$$

and the measurement equation for the i^{th} subsystem is represented by

$$z_{ii}(k) = \bar{H}_i(k)\, x_i(k) + \hat{H}_i(k) \sum_{\substack{j=1 \\ j\neq i}}^{N} L_{ij}(k)\, \hat{x}_j(k/k-1) + v_i^*(k) \tag{180}$$

where

$$w_i^*(k) \triangleq w_i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N} \bar{\Phi}_{ij}(k+1,k)\, e_j(k/k) \tag{181}$$

$$v_i^*(k) \triangleq v_i(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N} L_{ij}(k)\, e_j(k/k-1)\ . \tag{182}$$

Assume e_j, w_i^* and v_i^* can be treated as Gaussian white noise, then the standard Kalman filter algorithm can be applied to each subsystem with the following appropriately modified covariances:

$$Q_{ii}^*(k) \triangleq E[w_i^*(k)\, w_i^{*T}(k)] \tag{183}$$

$$Q_{ii}^*(k) = Q_{ii}(k) + \sum_{\substack{j=1 \\ j\neq i}}^{N} \bar{\Phi}_{ij}(k+1,k)\, P_j(k/k)\, \bar{\Phi}_{ij}^T(k+1,k) \tag{184}$$

$$R_{ii}^*(k) \triangleq E[v_i^*(k)\, v_i^{*T}(k)] \tag{185}$$

$$R_{ii}^*(k) = R_{ii}(k) + \hat{H}_i(k) \left\{ \sum_{\substack{j=1 \\ j\neq i}}^{N} L_{ij}(k)\, P_j(k/k-1)\, L_{ij}^T(k) \right\} \hat{H}_i^T(k)\ . \tag{186}$$

The result of applying the Kalman filtering technique yields the following subsystem state propagation and state update:

$$\hat{x}_i(k+1/k) = \Phi_{ii}(k+1,k)\, \hat{x}_i(k/k) + \sum_{\substack{j=1 \\ j\neq i}}^{N} \bar{\Phi}_{ij}(k+1,k)\, \hat{x}_j(k/k) \tag{187}$$

or

$$\hat{x}_i(k+1/k) = \bar{\phi}_{ii}(k+1,k)\ \hat{x}_i(k/k) + \sum_{\substack{j=1 \\ j\neq i}}^{N} \hat{x}_j(k+1/k) \tag{188}$$

$$\hat{x}_i(k+1/k+1) = \hat{x}_i(k+1/k) + \bar{K}_i(k+1)[z_{ii}(k+1) - \bar{H}_i(k+1)\ \hat{x}_i(k+1/k) - \hat{H}_i(k+1) \sum_{\substack{j=1 \\ j\neq i}}^{N} L_{ij}(k+1)\ \hat{x}_j(k+1/k)] \tag{189}$$

in which the filter gain $\bar{K}_i(k+1)$ is computed in the usual (Kalman filtering) manner:

$$\bar{K}_i(k+1) = P_{ii}(k+1/k)\ \bar{H}_i^T(k+1)[\bar{H}_i(k+1)\ P_{ii}(k+1/k)\ \bar{H}_i^T(k+1) + R_{ii}^*(k+1)]^{-1}, \tag{190}$$

and the covariance propagation and update are given by

$$P_{ii}(k+1/k) = \phi_{ii}(k+1/k)\ P_{ii}(k/k)\ \phi_{ii}^T(k+1/k) + Q_{ii}^*(k) \tag{191}$$

$$P_{ii}(k+1/k+1) = [I - \bar{K}_i(k+1)\ \bar{H}_i(k+1)]\ P_{ii}(k+1/k)[I - \bar{K}_i(k+1)\ \bar{H}_i(k+1)]^T + \bar{K}_i(k+1)\ R_{ii}^*(k+1)\ \bar{K}_i(k+1). \tag{192}$$

4.0 COMPUTER BURDENS

Discussions of computer burdens of various algorithms described in the previous section can be found in open literature (e.g., [14-24]) in which considerable data were provided pertaining to the computation efficiency of Covariance and Information filters, and their Square-Root variations, as well as the Chandrasekhar and Facotrized filters. However, computer burdens of the recently developed Partitioning and Decentralized filters have been documented only in closed literature (e.g., [92,93]). The purpose of this section is to provide an assessment of computer time and memory requirements of these relatively new approaches as well as other conventional algorithms.

It is well known that a precise quantitative statement of computer central process unit (cpu) time and memory storage requirements are difficult to obtain, since the exact number of counts depends upon the manner in which the filter equations are programmed and the particular computer used to process the data. For these reasons, only an approximate assessment is given here. For example, the logic time [15] has been excluded. Also, the transition matrix and the measurement matrix are assumed to be given, since the number of operations required to compute these matrices is heavily dependent on the nature of the problem. Furthermore, in the process of assessing operation counts, no distinction is made between multiplication and division. Although the cpu time required to perform a division is longer than multiplication, this assumption impacts the results in a minimal manner because the number of divisions in a filtering cycle is very small compared to the number of multiplications. Since multiplication requires much more cpu time than addition and subtraction, hence, for first order-magnitude approximation, it is reasonable to regard computer time as directly proportional to the number of multiplications (including divisions and extracting square roots) needed to complete the filtering cycle. In general, computer time and memory requiremen s are given in terms of n and m, where n is the dimension of the state vector and m is the dimension of the measurement vector. In the case of decentralized filtering, n_i and q_i are used to represent dimensions of the state vector and measurement vector, respectively. In the case of sequential processing of vector measurements or scalar measurements, other symbols will be used. For example, Bierman's equations for SQIF and the factorized filters belong to this category. Naturally, cautions must be taken when comparison is made between sequential- or scalar-processing technique and vector-processing technique.

All matrix inversions are assumed to be performed via the Cholesky factorization routine, which requires only $[\frac{1}{2}n^3 + \frac{1}{2}n^2 + n\,q]$ operations (q is the number of multiplications required to extract the square root of a scalar) and $[\frac{1}{2}n^2 + \frac{1}{2}n]$ memory locations. The number of operations required for the calculation of eigenvalues and eigenvectors are difficult to assess because of the iterative process involved. Thus a variational parameter is allowed in the operation counts.

Following the simplified approach together with the above assumptions, the number of predominant operations (multiplication) and memories required for various algorithms are assessed, and results are presented in Tables 1 - 12. In addition, Table 13 is provided to show recent trends in computer operation speeds, so that cpu time for different machines can be derived for each algorithm.

A separate table for the Extended Kalman filter is not being made because the standard Kalman filter includes the Extended Kalman filter, in which a set of nonlinear differential equations must be integrated in order to propagate states between measurements. For this reason, up to 90% of cpu time required per filter cycle is spent in integrating differential equations. The remaining 10% of cpu time would be spent on performing the computation sequence of the standard Kalman filter. The square root covariance filter of Andrews, Tapley and Choe, Morf, Levy and Kailath are close enough (as far as computer operations are concerned) to be considered as one class; therefore, only one table is provided under the heading of "Andrews Square Root Filter".

Computation details of the three derivatives of the General Partitioned Algorithm - Lambda, Delta and Per-Sample Partitioning - as well as their square root formulations are documented in [92]. In general, computer burdens of these derivatives are considerably less than that required by the general formulation. Particularly attractive is the Per-Sample Partitioning algorithm, which is memoryless and without repeated use of the Riccati equation.

The entire class of zero-order systems with scalar sequential measurements has been treated in great detail by Bierman [60], which includes tables summarizing operation counts of SRIF and factorized filters as well as counts of the Householder transformation. Hence, these tables are not duplicated here.

TABLE 1

Computational Requirements of the Standard Kalman Filter

Step	Computation Sequence	Operations	Storage
1	$\hat{x}(k/k, \mathcal{A})$		n
2	$P(k/k, \mathcal{A})$		n^2
3	$\Phi(k+1,k)$		n^2
4	$\Phi(k+1,k)\ P(k/k, \mathcal{A})$	n^3	store in 2
5	$\Phi(k+1,k)\ P(k/k, \mathcal{A})\ \Phi^T(k+1,k)$	n^3	store in 4
6	$Q(k)$		n^2
7	$P(k+1/k, \mathcal{A}) = \Phi(k+1,k)\ P(k/k, \mathcal{A})\ \Phi^T(k+1,k) + Q(k)$		store in 5
8	$\hat{x}(k+1/k, \mathcal{A}) = \Phi(k+1,k)\ \hat{x}(k/k, \mathcal{A})$	n^2	store in 1
9	$H(k+1)$		$m\ n$
10	$P(k+1/k, \mathcal{A})\ H^T(k+1)$	$m\ n^2$	store in 7
11	$R(k+1)$		m^2
12	$H(k+1)\ P(k+1/k, \mathcal{A})\ H^T(k+1) + R(k+1)$	$m^2\ n$	m^2
13	$[H(k+1)\ P(k+1/k, \mathcal{A})\ H^T(k+1) + R(k+1)]^{-1}$	$\frac{1}{2}(m^3+3m^2)+mq$	
14	$K(k+1, \mathcal{A}) = P(k+1/k, \mathcal{A})\ H^T(k+1)[H(k+1)\ P(k+1/k, \mathcal{A})\ H^T(k+1) + R(k+1)]^{-1}$	$m^2\ n$	$m\ n$
15	$z(k+1)$		m
16	$\hat{x}(k+1/k+1, \mathcal{A}) = \hat{x}(k+1/k, \mathcal{A}) + K(k+1, \mathcal{A})[z(k+1) - H(k+1)\ \hat{x}(k+1/k, \mathcal{A})]$	$m\ n$	store in 8
17	$P(k+1/k+1, \mathcal{A}) = P(k+1/k, \mathcal{A}) - K(k+1, \mathcal{A})[P(k+1/k, \mathcal{A})\ H^T(k+1)]^T$	$m\ n^2$	store in 10
	Total	$2n^3+n^2+2mn^2$ $+ 2m^2n + m\ n$ $+ \frac{1}{2}m^3+\frac{3}{2}m^2+mq$	$3n^2+n+2mn$ $+ 2m^2 + m$

TABLE 2

Computational Requirements of the Stabilized Kalman Filter

Step	Computation Sequence	Operations	Storage
1	$\hat{x}(k/k, \mathcal{A})$		n
2	$P(k/k, \mathcal{A})$		n^2
3	$\Phi(k+1,k)$		n^2
4	$\Phi(k+1,k)\ P(k/k, \mathcal{A})$	n^3	store in 2
5	$\Phi(k+1,k)\ P(k/k, \mathcal{A})\ \Phi^T(k+1,k)$	n^3	store in 4
6	$Q(k)$		n^2
7	$P(k+1/k, \mathcal{A}) = \Phi(k+1,k)\ P(k/k, \mathcal{A})\ \Phi^T(k+1,k) + Q(k)$		store in 5
8	$\hat{x}(k+1/k, \mathcal{A}) = \Phi(k+1,k)\ \hat{x}(k/k, \mathcal{A})$	n^2	store in 1
9	$H(k+1)$		$m\ n$
10	$P(k+1/k, \mathcal{A})\ H^T(k+1)$	$m\ n^2$	store in 7

(Continued)

TABLE 2 (Continued)

Step	Computation Sequence	Operations	Storage
11	$R(k+1)$		m^2
12	$H(k+1)\ P(k+1/k, \ell)\ H^T(k+1) + R(k+1)$	$m^2\ n$	m^2
13	$[H(k+1)\ P(k+1/k, \ell)\ H^T(k+1) + R(k+1)]^{-1}$	$\frac{1}{2}(m^3+3m^2)+mq$	store in 12
14	$K(k+1, \ell) = P(k+1/k, \ell)\ H^T(k+1)[H(k+1)\ P(k+1/k, \ell)\ H^T(k+1) + R(k+1)]^{-1}$	$m^2\ n$	$m\ n$
15	$z(k+1)$		m
16	$\hat{x}(k+1/k+1, \ell) = \hat{x}(k+1/k, \ell) + K(k+1, \ell)[z(k+1) - H(k+1)\ \hat{x}(k+1/k, \ell)]$	$m\ n$	store in 8
17	$I - K(k+1, \ell)\ H(k+1)$	$n^2\ m$	store in 10
18	$[I - K(k+1, \ell)\ H(k+1)]\ P(k+1/k, \ell)$	n^3	store in 3
19	$[I - K(k+1, \ell)\ H(k+1)]\ P(k+1/k, \ell)[I - K(k+1, \ell)\ H(k+1)]^T$	n^3	store in 17
20	$K(k+1, \ell)\ R(k+1)$	$m^2\ n$	store in 18
21	$P(k+1/k+1, \ell) = [I - K(k+1, \ell)\ H(k+1)]\ P(k+1/k, \ell)[I - K(k+1, \ell)H(k+1)]^T + K(k+1, \ell)\ R(k+1)\ K^T(k+1, \ell)$	$n^2\ m$	store in 19
	Total	$4n^3+n^2+3mn^2 + 3m^2n + mn + \frac{1}{2}m^3+\frac{3}{2}m^2+mq$	$3n^2+n+2mn + 2m^2 + m$

TABLE 3

Computational Requirements of the Potter Square Root Filter

Step	Computation Sequence	Operations	Storage
1	$\hat{x}(k/k, \ell)$		n
2	$S(k/k, \ell)$		n^2
3	$\phi(k+1,k)$		n^2
4	$\hat{x}(k+1/k, \ell) = \phi(k+1,k)\ \hat{x}(k/k, \ell)$	n^2	store in 1
5	$S(k+1/k, \ell) = \phi(k+1,k)\ S(k/k, \ell)$	n^3	store in 2
6	$h(k+1)$		n
7	$y(k+1, \ell) = S^T(k+1/k, \ell)\ h(k+1)$	n^2	n
8	$S(k+1/k, \ell)\ y(k+1, \ell)$	n^2	n
9	$r(k+1)$		1
10	$y^T(k+1, \ell)\ y(k+1, \ell)$	n	1
11	$K(k+1, \ell) = S(k+1/k, \ell)\ y(k+1, \ell) / [y^T(k+1, \ell)\ y(k+1, \ell) + r(k+1)]$	n	store in 8
12	$z(k+1)$		1
13	$h^T(k+1)\ \hat{x}(k+1/k, \ell)$	n	1
14	$\hat{x}(k+1/k+1, \ell) = \hat{x}(k+1/k, \ell) + K(k+1, \ell)[z(k+1) - h^T(k+1)\ \hat{x}(k+1/k, \ell)]$	n	store in 4
15	$1 - y^T(k+1, \ell)\ y(k+1, \ell) / [y^T(k+1, \ell)\ y(k+1, \ell) + r(k+1)]$	1	store in 12
16	$\alpha = \pm [1 - y^T(k+1, \ell)\ y(k+1, \ell) / [y^T(k+1, \ell)\ y(k+1, \ell) + r(k+1)]\ ]^{1/2}$	q	store in 15
17	α = [step 16] / [step 10]	1	store in 13
18	$y(k+1, \ell)\ y^T(k+1, \ell)$	n	store in 6
19	$\alpha\ y(k+1, \ell)\ y^T(k+1, \ell)$	n	store in 18
20	$S(k+1/k+1, \ell) = S(k+1/k, \ell)[I - \alpha\ y(k+1, \ell)\ y^T(k+1, \ell)]$	n^2	store in 5
	Total	$n^3 + 4n^2 + 6n + q+2$	$2n^2 + 4n + 4$

TABLE 4

Computational Requirement of the Bellantoni and Dodge Square Root Filter

Step	Computation Sequence	Operations	Storage
1	$\hat{x}(k/k, \mathcal{D})$		n
2	$S(k/k, \mathcal{D})$		n^2
3	$\phi(k+1,k)$		n^2
4	$\hat{x}(k+1/k, \mathcal{D}) = \phi(k+1,k)\ \hat{x}(k/k, \mathcal{D})$	n^2	store in 1
5	$S(k+1/k, \mathcal{D}) = \phi(k+1,k)\ S(k/k, \mathcal{D})$	n^3	n^2
6	$R^{-\frac{1}{2}}(k+1)$		m^2
7	$H(k+1)$		$m\ n$
8	$R^{-\frac{1}{2}}(k+1)\ H(k+1)$	$m^2\ n$	$m\ n$
9	$z(k+1)$		m
10	$z(k+1) - H(k+1)\ \hat{x}(k/k, \mathcal{D})$	$m\ n$	store in 9
11	$B^T(k+1) = R^{-\frac{1}{2}}(k+1)\ H(k+1)\ S(k+1/k, \mathcal{D})$	$n^2\ m$	store in 7
12	$I + B^T(k+1)\ B(k+1)$	$m^2\ n$	store in 3
13	$[I + B^T(k+1)\ B(k+1)]^{-1}$	$\frac{1}{2}(n^3+3m^2)+ m\ q$	store in 12
14	$R^{-\frac{1}{2}}(k+1)$[step 10]	m^2	store in 4
15	[step 13] [step 14]	m^2	store in 10
16	[B(k+1)] [step 15]	$m\ n$	store in 8
17	$\hat{x}(k+1/k+1, \mathcal{D}) = \hat{x}(k+1/k, \mathcal{D}) + S(k+1/k, \mathcal{D})$[step 16]	n^2	store in 4
18	$B(k+1)\ B^T(k+1)$	$m\ n^2$	store in 12
19	D, diagonal matrix consisting eigenvalues	μ^*	store in 15
20	T, transformation matrix consisting eigenvectors		store in 16
21	$(I + D)^{-\frac{1}{2}} - I$	$m\ q$	store in 20
22	$[(I + D)^{-\frac{1}{2}} - I]\ T^T$	$m\ n$	store in 11
23	T[step 22] + I	$m\ n^2$	store in 18
24	$S(k+1/k+1, \mathcal{D}) = S(k+1/k, \mathcal{D})$[step 23]	n^3	store in 2
	* variable number of operations depending on computer being used as well as method being used to compute eigenvalues and eigenvectors.	Total $2n^3 + 2n^2 + 3n^2m + 2m^2n + 3\ m\ n + \frac{7}{2}m^2 + \frac{1}{2}m^3 + 2m\ q + \mu$	$3n^2 + n + 2\ n\ m + m + m^2$

TABLE 5

Computational Requirements of the Andrews Square Root Filter

Step	Computation Sequence	Operation	Storage
1	$\hat{x}(k/k, \mathcal{D})$		n
2	$S(k/k, \mathcal{D})$		n^2
3	$\phi(k+1,k)$		n^2
4	$\hat{x}(k+1/k, \mathcal{D}) = \phi(k+1,k)\ \hat{x}(k/k, \mathcal{D})$	n^2	store in 1
5	$S(k+1/k, \mathcal{D}) = \phi(k+1,k)\ S(k/k, \mathcal{D})$	n^3	n^2
6	$H(k+1)$		$n\ n$
7	$R(k+1)$		$\frac{m^2}{2} + \frac{m}{2}$
8	$R^{\frac{1}{2}}(k+1)$		$\frac{m^2}{2} + \frac{m}{2}$
9	$\bar{H}(k+1) = H(k+1)\ S(k+1/k, \mathcal{D})$	$n^2\ m$	store in 2
10	$UU^T = R + \bar{H}^T\ \bar{H}$	$\frac{m^2}{2} + \frac{m}{2}$	store in 3
11	U, U^{-1}	$\frac{m^3}{3} + m^2 - \frac{m}{3} + m\ q$	store in 10
12	$z(k+1)$		m
13	$z(k+1) - H(k+1)\ \hat{x}(k+1/k, \mathcal{D})$	$m\ n$	store in 12
14	U^{-1}[step 13]	$\frac{m^2}{2} + \frac{m}{2}$	store in 13

(Continued)

TABLE 5 (Continued)

Step	Computation Sequence	Operation	Storage
15	$\bar{H}\, U^{-T}$	$\frac{nm^2}{2} + \frac{nm}{2}$	store in 6
16	$[U + R^{\frac{1}{2}}]^{-1}$	$\frac{m^3}{6} + \frac{m^2}{2} + \frac{m}{3}$	store in 7
17	$S(k+1/k, \Delta)$ [step 15]	$n^2 m$	store in 11
18	[step 16] $\bar{H}^T$	$\frac{nm^2}{2} + \frac{nm}{2}$	store in 15
19	$\hat{x}(k+1/k+1, \Delta) = \hat{x}(k+1/k, \Delta)$ + [step 17] [step 14]	$n\, m$	store in 4
20	$S(k+1/k+1, \Delta) = S(k+1/k, \Delta)[I$ - (step 15)(step 16) $\bar{H}^T]$	$n^2 m$	store in 9
	Total	$n^3 + n^2 + 3n^2 m + \frac{3}{2} nm^2 + \frac{7}{2} nm + \frac{1}{2} m^3 + 2m^2 + \frac{1}{2} m + n\, q$	$3n^2 + n + m\, n + m^2 + 2m$

TABLE 6

Computational Requirements of the Schmidt Square Root Filter

Step	Computation Sequence	Operation	Storage
1	$\hat{x}(k/k, \Delta)$		n
2	$S(k/k, \Delta)$		n^2
3	$\phi(k+1,k)$		n^2
4	$\hat{x}(k+1/k, \Delta) = \phi(k+1,k)\, \hat{x}(k/k, \Delta)$	n^2	store in 1
5	$Q^{\frac{1}{2}}(k+1)$		n^2
6	$A = [\phi(k+1,k)\, S(k/k, \Delta) \mid Q^{\frac{1}{2}}(k+1)]$	n^3	$2n^2$
7	$A^T e$ (e is an arbitrary n-column vector)		$2n$
8	$A\, A^T e$		n
9	$A\, A^T e\, e^T$		store in 3
10	$T = I - A\, A^T e\, e^T$ / [first element of $A\, A^T e$ vector]		store in 9
11	$\bar{A} = T\, A$		store in 6
12	[first element of $A\, A^T e$ vector]$^{\frac{1}{2}}$		1
13	First column of $S(k+1/k, \Delta) = A\, A^T e$ / [step 12]		store in 8
14	To complete the $S(k+1/k, \Delta)$ matrix, steps 7-13 will be iterated (n-1) times. Since the dimension of A and T are effectively reduced by one at each iteration, the number of operations for:		
	7 & 8	$\sum_{i=1}^{n} (n+m)(n+1-i)$	
	9 & 10	$\sum_{i=1}^{n} (n-i)$	
	11	$\sum_{i=1}^{n-1} (n+m)(n-i)^2$	
	12 & 13	$\sum_{i=1}^{n} (n-i)$	
15	$h(k+1)$		n
16	$y(k+1, \Delta) = S^T(k+1/k, \Delta)\, h(k+1)$	n^2	store in 13
17	$S(k+1/k, \Delta)\, y(k+1, \Delta)$	n^2	n
18	$r(k+1)$		1
19	$y^T(k+1, \Delta)\, y(k+1, \Delta)$	n	1
20	$K(k+1, \Delta) = S(k+1/k, \Delta) y(k+1, \Delta) / y^T(k+1, \Delta) y(k+1, \Delta) + r(k+1)$	n	store in 17
21	$a(k+1)$		1
22	$h^T(k+1)\, \hat{x}(k+1/k, \Delta)$	n	1
23	$\hat{x}(k+1/k+1, \Delta) = \hat{x}(k+1/k, \Delta)$ + $K(k+1, \Delta)[a(k+1) - h^T(k+1) \hat{x}(k+1/k, \Delta)]$	n	store in 4

(Continued)

TABLE 6 (Continued)

Step	Computation Sequence	Operation	Storage
24	$1-y^T(k+1,\ell)y(k+1,\ell)\ /\ [y^T(k+1,\ell)y(k+1,\ell) + r(k+1)]$	1	store in 21
25	$-1 \pm \{1-y^T(k+1,\ell)y(k+1,\ell)/[y^T(k+1,\ell)y(k+1,\ell)+r(k+1)]\}^{\frac{1}{2}}$	q	store in 24
26	α = [step 25]/[step 19]	1	store in 22
27	$y(k+1,\ell)\ y^T(k+1,\ell)$	n	store in 15
28	$\alpha\ y(k+1,\ell)\ y^T(k+1,\ell)$	n	store in 27
29	$S(k+1/k+1,\ell) = S(k+1/k,\ell)[1-\alpha\ y(k+1,\ell)y^T(k+1,\ell)]$	n^2	store in 2
	Total	$n^3 + 3n^2 + 6n + q + 2$	$5n^2 + 6n + 5$
		$+ 2\sum_{i=1}^{n}(n-i) + \sum_{i=1}^{n}(n+m)(n+1-i) + \sum_{i=1}^{n}(n+m)(n-i)^2$	

TABLE 7

Computational Requirements of the Carlson Square Root Filter

Step	Computation Sequence	Operation	Storage
1	$\hat{x}(k/k,\ell)$		n
2	$S(k/k,\ell)$		n^2
3	$\phi(k+1,k)$		n^2
4	$\hat{x}(k+1/k,\ell) = \phi(k+1,k)\ \hat{x}(k/k,\ell)$	n^2	store in 1
5	$Q(k+1)$		n^2
6	$\phi(k+1,k)\ S(k/k,\ell)$	n^3	store in 2
7	$\phi(k+1,k)\ S(k/k,\ell)\ S^T(k/k,\ell)\ \phi^T(k+1,k)$	n^3	store in 6
8	$S(k+1/k,\ell) =$ $[\phi(k+1,k)S(k/k,\ell)S^T(k/k,\ell)\phi(k+1,k)+Q(k+1)]^{\frac{1}{2}}$	$\sum_{i=1}^{n-2}(n-1-i) + \sum_{i=1}^{n-1}(n-i)$ $+ (n-2) + (n-1)q$	
9	$h(k+1)$		n
10	$y(k+1,\ell) = S^T(k+1/k,\ell)\ h(k+1)$	n^2	n
11	$S(k+1/k,\ell)\ y(k+1,\ell)$	n^2	n
12	$r(k+1)$		1
13	$y^T(k+1,\ell)\ y(k+1,\ell)$	n	1
14	$K(k+1,\ell) = S(k+1/k,\ell)y(k+1,\ell)/[y^T(k+1,\ell)y(k+1,\ell)+r(k+1)]$	n	store in 11
15	$z(k+1)$		1
16	$h^T(k+1)\ \hat{x}(k+1/k,\ell)$	n	1
17	$\hat{x}(k+1/k+1,\ell) = \hat{x}(k+1/k,\ell) + K(k+1,\ell)[z(k+1)-h^T(k+1)\hat{x}(k+1/k,\ell)]$	n	store in 4
18	$y(k+1,\ell)\ y^T(k+1,\ell)$	n	store in 9
19	$I - [y(k+1,\ell)y^T(k+1,\ell)]/[y^T(k+1,\ell)y(k+1,\ell)+r(k+1)]$	n	store in 3
20	$[I - [y(k+1,\ell)y^T(k+1,\ell)]/[y^T(k+1,\ell)y(k+1,\ell)+r(k+1)]]^{\frac{1}{2}}$	same as step 8	store in 19
21	$S(k+1/k+1,\ell) = S(k+1/k,\ell)$[step 20]	$\sum_{i=1}^{n} i[n-(i-1)]$	store in 7
	Total	$2n^3 + 3n^2 + 6n$	$3n^2 + 4n + 4$
		$+ 2\sum_{i=1}^{n-2}(n-1-i) + 2(n-2)$	
		$+ 2(n-1)q + 2\sum_{i=1}^{n-1}(n-i)$	
		$+ \sum_{i=1}^{n} i[n-(i-1)]$	

TABLE 8

Computational Requirements of the Information Filter

Step	Computation Sequence	Operations	Storage
1	$\hat{d}(k/k,\ell)$		n
2	$P^{-1}(k/k,\ell)$		n^2
3	$\phi^{-1}(k+1,k)$		n^2
4	$Q^{-1}(k)$		n^2
5	$Q(k)$		n^2
6	$P^{-1}(k/k,\ell)\ \phi^{-1}(k+1,k)$	n^3	store in 2
7	$F(k) = \phi^{-T}(k+1,k)\ P^{-1}(k/k,\ell)\ \phi^{-1}(k+1,k)$	n^3	store in 6
8	$[F(k) + Q^{-1}(k)]^{-1}$	$\frac{1}{2}(n^3+3n^2)+nq$	store in 7
9	$F(k)[F(k) + Q^{-1}(k)]^{-1}$	n^3	store in 8
10	$P^{-1}(k+1/k,\ell) = F(k)-F(k)[F(k)+Q^{-1}(k)]^{-1}F(k)$	n^3	store in 9
11	$P^{-1}(k+1/k,\ell)\ Q(k)$	n^3	store in 5
12	$\phi^{-T}(k+1,k)\ \hat{d}(k/k,\ell)$	n^2	store in 1
13	$\hat{d}(k+1/k,\ell) = [I - P^{-1}(k+1/k,\ell)Q(k)]\ \phi^{-T}(k+1,k)\hat{d}(k/k,\ell)$	n^2	store in 12
14	$H(k+1)$		nm
15	$z(k+1)$		m
16	$R^{-1}(k+1)$		m^2
17	$H^T(k+1)\ R^{-1}(k+1)$	$n\ m^2$	store in 14
18	$\hat{d}(k+1/k+1,\ell) = \hat{d}(k+1/k,\ell) + H^T(k+1)\ R^{-1}(k+1)\ z(k+1)$	m^2	store in 13
19	$P^{-1}(k+1/k+1,\ell) = P^{-1}(k+1/k,\ell) + H^T(k+1)\ R^{-1}(k+1)\ H(k+1)$	$n\ m^2$	store in 10
	Total	$\frac{11}{2}n^3 + \frac{7}{2}n^2 + 2nm^2$ + m^2 + nq	$4n^2 + n + nm$ + $m + m^2$

TABLE 9

Computational Requirements of the Chandrasekhar Filter

Step	Computation Sequence	Operations	Storage
1	$\hat{x}(k/k)$		n
2	$\phi(k+1,k)$		n^2
3	$\phi(k+1,k)\ \hat{x}(k/k)$	n^2	store in 1
4	$A(k-1)$		$n\ m$
5	$A'(k-2)$		$n\ m$
6	$C^{-1}(k-2)$		m^2
7	M		$m\ n$
8	R		m^2
9	$(MA(k-1) + R)^{-1}$	$\frac{1}{2}(m^3+3m^2)+mq$	m^2
10	Eq. 113	$2m^3+3m^2n+nm^2$	store in 6
11	Eq. 108	$2m^2n + 4mn^2$	store in 5
12	Eq. 107	$m^2n + 2mn^2$	store in 4
13	Eq. 106	$n^2m + m^2n$	$n\ m$
14	$\hat{x}(k+1/k+1) = \hat{x}(k+1/k) + K(k+1)[z(k+1) - M\ \hat{x}(k+1/k)]$	$m\ n$	store in 3
	Total	$2m^3 + 7\ nm^2$ + $n^2(1+8m)$ + $m\ n$ + $\frac{1}{2}(m^3+3m^2)$ + mq	$2m^2 + n^2$ + $n(1+4m)$

TABLE 10

Computational Requirements of the General Partitioning Filter

(in addition to the "nominal" Kalman filter computation)

Step	Computation Sequence	Operations	Storage
1	$\hat{x}_n(k-1/k-1, \Delta)$		n
2	$\Phi(k,k-1)$		n^2
3	$H(k)$		$m\ n$
4	Eq. 126	$m\ n^2$	m
5	$P_n(k/k-1, \Delta)$		n^2
6	$R(k)$		m^2
7	Eq. 127	$m^2\ n$	n^2
8	$P_{g_n}^{-1}(k/k-1, \Delta)$		m^2
9	Eq. 128	$n^2\ m$	$n\ m$
10	Eq. 125	$n^2 m$	n^2
11	$O_n(k-1, \Delta)$		n^2
12	Eq. 124	$2n^3 + 3n^2 m + 2nm^2$	n^2
13	$P_r(\Delta)$		n^2
14	$P_r(\Delta)\ O_n(k, \Delta)$	n^3	n^2
15	$(P_r(\Delta)\ O_n(k, \Delta) + I)^{-1}$	$\frac{1}{2}(n^3 + 3n^2) + nq$	store in 14
16	Eq. 122	n^3	store in 15
17	$M(k-1, \Delta)$		$n\ m$
18	Eq. 123	$n^3 + n^2 m + n\ m^2$	store in 17
19	$P_r^{-1}(\Delta)$		n^2
20	$\hat{x}_r(\Delta)$		n
21	Eq. 121	$n^2 m + n^3$	store in 20
22	Eq. 119	n^2	store in 1
23	Eq. 120	$2n^3$	n^2
	Total	$8n^3 + n^2(6m+1) + 4m^2 n$ $+ \frac{1}{2}(n^3 + 3n^2) + nq$	$10n^2 + 2n + 2m^2$ $+ m + 3m\ n$

TABLE 11A

Computational Requirements of the SLU Filter

Step	Computation Sequence	Operations	Storage
1	$\hat{x}_i(k/k-1)$		n_i
2	$P_i(k/k-1)$		n_i^2
3	$\Phi_{ii}(k+1,k)$		n_i^2
4	$\bar{H}_i(k)$		$q_i \times p_i$
5	$U_2^{-1}(k)\ \bar{H}_i(k)\ U_i(k)$	see Table II B	store in 4
6	$\dot{L}_{ii}(k)$		$n_i \times p_i$
7	$\dot{L}_{ii}(k)\ U_i(k)$	$n_i\ p_i^2$	store in 6
8	$L_i(k)$		$p_i \times m$
9	$U_i^{-1}(k)\ L_i(k)$	$p_i^2\ n$	store in 8
10	$\bar{\bar{H}}_i(k)$		$q_i \times n_i$
11	$U_2^{-1}(k)\ \bar{\bar{H}}_i(k)$	$q_i^2\ n_i$	store in 10
12	$s_i(k)$		q_i
13	$U_2^{-1}(k)\ s_i(k)$	q_i^2	store in 12

(Continued)

TABLE 11A (Continued)

Step	Computation Sequence	Operations	Storage
14	$R_i(k)$		q_i^2
15	$U_2^{-1}(k)\ R_i(k)\ U_2^{-T}(k)$	q_i^3	store in 14
16	Eq. 172	$2n_i^3 + n_i^2(3q_i-p_i) + n_i(q_i-p_i)(2q_i-p_i) + n_ip_i^2 + (q_i - p_i)^3$	store in 2
17	Eq. 171	$2n_i^2(q_i-p_i) + 2n_i(q_i-p_i)+q_i+(q_i-p_i)^3$	$n_i \times (q_i - p_i)$
18	Eq. 162	$n_i^2 + 2n_i\ q_i$	store in 1
	Total	$2n_i^3 + q_i^3 + 2(q_i - p_i)^3 + n_i^2[1 + 5q_i-3p_i] + p_i^2(3n_i+n) + q_i^2(1 + 3n_i) + q_i - 3n_i\ p_i\ q_i + 2n_i(2q_i - p_i)$	$2n_i^2 + n_i + q_i\ p_i + p_in + 2q_i\ n_i$

$n \triangleq$ aggregate system dimension
$n_i \triangleq$ local system dimension
$p_i \triangleq$ interactive measurement dimension
$q_i \triangleq$ local measurement dimension

TABLE 11B

Step 5 of the SLU Filter Computation Sequence (Ref. [91])

Machine	Execution Time (sec.)
IBM 370/195 (Argonne National Laboratory)	1.0
IBM 360/75 (University of Illinois)	9.7
IBM 360/65 (AMES Laboratory)	17.0
IBM 370/165 (University of Toronto)	2.6
IBM 370/168 Mod 3 (Stanford University)	2.3
Burroughs 6700 (University of California, San Diego)	82.0
CDC 6600 (Kirtland Airforce Base)	4.4
CDC Cyber 175 (NASA Langley Research Center)	1.2
CDC 7600 (National Center for Atomospheric Research)	0.87
CDC 7600 (Lawrence Livermore Laboratory)	1.2
CDC 6400 (Northeastern University)	15.0
CDC 6400/6500 (Purdue University)	17.0
CDC 6600/6400 (University of Texas)	5.2
Honeywell 6070 (Bell Laboratories)	9.6
Univac 1110 (University of Wisconsin)	7.7
DEC KA - PDP - 10 (Yale University)	79.0
AMDAHL 470V/6 (University of Michigan)	2.1

TABLE 12

Computational Requirements of the SRA Filter

Step	Computation Sequence	Operations	Storage
1	$\hat{x}_i(k/k)$		n_i
2	$P_{ii}(k/k)$		n_i^2
3	$\bar{\Phi}_{ii}(k+1,k)$		n_i^2
4	$\bar{\Phi}_{ii}(k+1,k)\ \hat{x}_i(k/k)$	n_i^2	store in 1
5	Eq. 184	$N(2n_i^3)$	n_i^2
6	Eq. 191	$4n_i^3$	store in 2

(Continued)

TABLE 12 (Continued)

Step	Computation Sequence	Operations	Storage
7	Eq. 186	$q_i^3 + (n_i^2 q_i + n_i q_i^2)N$	q_i^2
8	Eq. 190	$n_i^2 q_i + 2n_i q_i^2 + q_i^2$	$n_i \times q_i$
9	Eq. 189	$q_i + 2n_i q_i + (n_i^2 + n_i^2 q_i)N$	store in 4
10	Eq. 192	$2 n_i^3$	store in 2
	Total	$2n_i^3(N+3) + n_i^2(1 + 2q_i + N + q_i N)$	$3n_i^2 + n_i$
		$+ n_i(q_i^2 N + 2q_i^2 + 2q_i) + q_i^3 + q_i^2 + q_i$	$+ q_i^2 + n_i q_i$

N $\triangleq$ number of subsystems $q_i \triangleq$ local measurement dimension

$n_i \triangleq$ local system dimension

TABLE 13

Trends in Speed of Computer Operations

Operation	1968 (μ sec)	1974 (μ sec)	1980 (μ sec)
Load	2	1	0.66
Multiply	10	6	2.86
Divide	15	7	5.61
Add	3	2	1.54
Store	2	2	0.44
Increment Index Register	2	1	0.33

5.0 CONCLUSION

The question of how to attain computation efficiency has puzzled many engineers despite the fact that many attempts have been made to present guidelines as to which algorithm is the best (most efficient). The answer is still imprecise, since it depends on factors such as operational computer parameters (instruction set, word length, cpu time, etc.), programming methods (single or double precision, linear or multi-dimension arrays, exploitation of symmetric and sparse matrices, etc.), the size and complexity (cross-coupling) of the transition matrix, and methods of processing measurement data (simultaneous, subgroup, sequential, decentralized, etc.).

The purpose of this paper is to provide an order-magnitude approximation on computational requirements of various filtering algorithms without making any specific recommendations as to which one is the "best". Results are given in tabulated form (Tables 1-12). In using these tables, caution must be exercised (especially when comparisons are made among algorithms) since they are not - and cannot be - compiled on a uniform basis. For example, Bierman's SRIF and Factorized filters are designed for the processing of sequential measurement data of a zero order dynamic system; the Partitioning filter is designed to deal with unknown parameters as well as state estimation, hence this algorithm is efficient in the sense that a separate adaptive routine is not needed. The Decentralized filter is most appropriate for large-scale but decomposed subsystems application; it is efficient in the sense that computer operations are less for a set of subsystems than that required for the aggregate system. Therefore, users of these algorithms are advised to perform cost-effectiveness trade-off studies according to given situations - before deciding which algorithm to be selected. It is hoped that this paper does provide sufficient information for such trade-off studies.

REFERENCES

1. O. Neugebauer, The Exact Sciences in Antiquity, Princeton Univ. Press., New Jersey, 1952.
2. H. W. Sorenson, "Estimation Theory: A Historical Perspective", Proc. S. U. 1972 IEEE Conference.
3. R. A. Fisher, Contributions to Mathematical Statistics, Wiley, New York, 1950.
4. A. N. Kolmogorov, "Interpolation und Extrapolation von Stationaren Zufalligen Folgen", Bulletin of the Academy of Sciences, USSR, Mathematical Series, Vol. 5, pp. 3-14, 1941.
5. N. Wiener, The Extrapolation, Interpolation and Smoothing of Stationary Time Series, Wiley, New York, 1949.

REFERENCES (Continued)

6. P. Swerling, "First-Order Error Propagation in a Stage-Wise Smoothing Procedure for Satellite Observations", J. Astronautical Sciences, Vol. 6, pp. 46-52, 1959.

7. A. G. Carlton and J. W. Follin, "Recent Development in Fixed and Adaptive Filtering", NATO AGARDograph 21, 1956.

8. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", J. Basic Eng. Trans. ASME, Vol. 82, pp. 35-45, 1960.

9. R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory", J. Basic Eng. Trans. ASME, Vol. 83D, pp. 95-108, 1961.

10. L. Chin, "Advances in Adaptive Filtering", Control and Dynamic Systems, C. T. Leondes, editor, Academic Press, 1979.

11. R. H. Battin, Astronautical Guidance, McGraw Hill, New York, 1964.

12. J. F. Bellantoni and K. W. Dodge, "A Square Root Formulation of the Kalman-Schmidt Filter", AIAA Journal, Vol. 5, No. 7, pp. 1309-1314, July 1967.

13. A. Andrews, "A Square Root Formulation of the Kalman Covariance Equations", AIAA Journal, Vol. 6, No. 6, pp. 1165-1166, June 1968.

14. S. F. Schmidt, "Computational Techniques in Kalman Filtering in 'Theory and Application of Kalman Filter'", NATO AGARDograph 139, pp. 65-86, Feb. 1970.

15. J. M. Mendel, "Computational Requirements for a Discrete Kalman Filter", IEEE Trans. Autom. Contr., Vol. AC-16, No. 6, pp. 748-758, Dec. 1971.

16. P. G. Kaminski, A. E. Bryson and S. F. Schmidt, "Discrete Square Root Filtering: A Survey of Current Techniques", IEEE Trans. Autom. Contr., Vol. AC-16, No. 6, pp. 727-736, Dec. 1971.

17. I. A. Gura and A. B. Bierman, "On Computational Efficiency of Linear Filtering Algorithms", Automatica, Vol. 7, pp. 299-313, 1971.

18. J. S. Meditch, "On the Data Storage Problem in Sequential Discrete-Time Smoothing", Proc. 10th Annual Allerton Conf. on Ckt. and Sys. Theory, Univ. of Ill., pp. 68-75, Oct. 1972.

19. G. J. Bierman, "A Comparison of Discrete Linear Filtering Algorithms", IEEE Trans. Aerospace and Elect. Sys., Vol. AES-9, No. 1, Jan. 1973.

20. N. A. Carlson, "Fast Triangular Formulation of the Square Root Filter", AIAA Journal, Vol. 11, No. 9, pp. 1259-1265, Sept. 1973.

21. G. J. Bierman, "On the Application of Discrete Square Root Information Filtering", Int. J. Control, Vol. 20, No. 3, pp. 465-477, 1974.

22. G. J. Bierman, "Sequential Square Root Filtering and Smoothing of Discrete Linear Systems", Automatica, Vol. 10, pp. 147-158, 1974.

23. G. J. Bierman, "The Treatment of Bias in the Square-Root Information Filter/Smoother", J. Optimization Theory and Applications, Vol. 16, No. 1 and No. 2, pp. 165-178, July 1975.

24. G. J. Bierman, "Measurement Updating Using the U-D Factorization", Automatica, Vol. 12, pp. 375-382, 1976.

25. R. Fletcher and M. J. D. Powell, "On the Modification of LDL^T Factorizations", Math. of Comp., Vol. 28, No. 128, pp. 1067-1087, Oct. 1974.

26. P. E. Gill, G. H. Golub, W. Murray and M. A. Saunders, "Methods for Modifying Matrix Factorizations", Math. of Comp., Vol. 28, No. 126, pp. 505-535, Apr. 1974.

27. T. Kailath, "Some New Algorithms for Recursive Estimation in Constant Linear Systems", IEEE Trans. Info. Theory, Vol. IT-19, pp. 750-760, Nov. 1973.

28. A. Lindquist, "Optimal Filtering of Continuous-Time Stationary Processes by Means of the Backward Innovation Process", SIAM J. Contr., Vol. 12, pp. 747-754, Nov. 1974.

29. A. Lindquist, "A New Algorithm for Optimal Filtering of Discrete-Time Stationary Processes", SIAM J. Contr., Vol. 12, pp. 736-746, Nov. 1974.

30. D. G. Lainiotis, "Generalized Chandrasekhar Algorithms: Time Varying Models", IEEE Trans. Autom. Contr., pp. 728-732, Oct. 1976.

31. R. F. Brammer, "A Note on the Use of Chandrasekhar Equations for the Calculation of the Kalman Filter Gain Matrix", IEEE Trans. Info. Theory, pp. 334-336, May 1975.

32. D. G. Lainiotis, "Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation", Proc. IEEE, 65, No. 8, 1976.

33. D. G. Lainiotis, "A Unifying Framework for Linear Estimation: Generalized Partitioned Algorithms", J. Info. Sci., 10, No. 4, June 1976.

34. D. G. Lainiotis, "Partitioned Estimation Algorithms, II: Linear Estimation", J. Info. Sci., 7, No. 3, Nov. 1974.

35. D. G. Lainiotis, "Partitioned Linear Estimation Algorithms: Discrete Case", IEEE Trans. Autom. Control, 20, No. 3, Apr. 1975.

36. D. G. Lainiotis, "Optimal Linear Smoothing: Continuous Data Case", Int. J. Contr., 17, No. 5, May 1973.

37. D. G. Lainiotis, "Discrete Riccati Equation Solutions: Partitioned Algorithms", IEEE Trans. Autom. Contr., 20, No. 4, Aug. 1975.

38. D. G. Lainiotis, "Partitioned Filters", Info. Sci., Vol. 17, Jan. 1979.

39. C. W. Sanders, E. C. Tacker and T. D. Linton, Decentralized Estimation via Constrained Filters, Tech. Rept. ECE-73-1, Univ. of Wisconsin, 1973.

REFERENCES (Continued)

40. C. W. Sanders, "A Decentralized Filter for Interacting Dynamical Systems", Proc. 4th Symp. on Nonlinear Estimation Theory and Applications, San Diego, CA, 1973.

41. C. W. Sanders, "Decentralized State Estimation in Interconnected Systems", Proc. 1974 IEEE Conf. on Decision and Control, Phoenix, AZ, 1974.

42. C. W. Sanders, "A New Class of Decentralized Filters for Interconnected Systems", IEEE Trans. Autom. Contrl., Vol. AC-19, No. 3, 1974.

43. C. W. Sanders, "Stability and Performance of a Class of Decentralized Filters", Int. J. Contr., Vol. 23, No. 2, 1976.

44. C. W. Sanders, E. C. Tacker, T. D. Linton and R. Y. S. Ling, "Specific Structures for Large Scale Estimation Algorithms Having Information Exchange", IEEE Trans. Autom. Contr., Vol. AC-23, No. 2, Apr. 1978.

45. A. H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

46. A. Gelb (Ed.), Applied Optimal Estimation, The M.I.T. Press, Massachusetts, 1974.

47. P. S. Maybeck, Stochastic Models, Estimation and Control, Academic Press, New York, 1979.

48. A. P. Sage and J. L. Melsa, Estimation Theory with Applications to Communications and Control, McGraw-Hill, New York, 1971.

49. J. L. Melsa and D. L. Cohn, Decision and Estimation Theory, McGraw-Hill, New York, 1978.

50. B. D. O. Anderson and J. B. Moore, Optimal Filtering, Prentice-Hall, New Jersey, 1979.

51. J. S. Meditch, Stochastic Optimal and Linear Estimation and Control, McGraw-Hill, New York, 1969.

52. R. S. Bucy and P. D. Joseph, Filtering for Stochastic Processes with Applications to Guidance, Interscience Publishers, New York, 1968.

53. V. N. Fadeeva, Computational Methods of Linear Algebra, Dover, New York, 1959.

54. A. S. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell, Waltham, Mass., 1964.

55. G. H. Golub, "Numerical Methods for Solving Linear Least Squares Problems", Numer. Math., Vol. 7, pp. 206-216, 1965.

56. P. Businger and G. H. Golub, "Linear Least Squares Solution by Householder Transformations", Numer. Math., Vol. 7, pp. 269-276, 1965.

57. T. Jorden, "Experiments on Error Growth Associated with Some Linear Least Square Procedures", Math. Comput., Vol. 20, pp. 325-328, 1966.

58. R. J. Hanson and C. L. Lawson, "Extensions and Applications of the Householder Algorithm for Solving Linear Least Squares Problems", Math. Comput., Vol. 23, pp. 787-812, 1969.

59. H. C. Cox, "Estimation of State Variables via Dynamic Programming", Proc. JACC, pp. 376-381, 1964.

60. G. J. Bierman, Factorization Methods for Discrete Sequential Estimation, Academic Press, 1977.

61. W. S. Agee and R. H. Turner, "Triangular Decomposition of a Positive Definite Matrix Plus a Symmetric Dyad with Application to Kalman Filtering", White Sands Missile Range Tech. Rept. No. 38, 1972.

62. G. J. Bierman and C. L. Thornton, "Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study", Automatica, 13, pp. 23-35, 1977.

63. W. M. Gentleman, "Least Squares Computations by Givens Transformation without Square Roots", J. Inst. Math. Appl., Vol. 12, pp. 329-336, 1973.

64. P. E. Gill, W. Murray and M. A. Saunders, "Methods for Computing and Modifying the LDV Factors of a Matrix", Math. of Comp., Vol. 29, No. 132, pp. 1051-1077, 1975.

65. C. L. Thornton and G. J. Bierman, "Gram Schmidt Algorithms for Covariance Propagation", Proc. IEEE Control and Decision Conf., Houston, Texas, pp. 489-498, 1975.

66. R. N. Wampler, "A Report on the Accuracy of Some Widely Used Least Squares Computer Programs", J. Amer. Statist. Assoc., 65 (330) pp. 549-565, 1970.

67. M. Morf, G. S. Sidhu and T. Kailath, "Some New Algorithms for Recursive Estimation in Constant, Linear, Discrete-Time Systems", IEEE Trans. Autom. Contr., Vol. AC-19, pp. 315-323, Aug. 1974.

68. B. Friedlander, T. Kailath, M. Morf and L. Ljung, "Extended Levinson and Chandrasekhar Equations for General Discrete-Time Linear Estimation Problems", IEEE Trans. Autom. Contr., Vol. AC-23, No. 4, pp. 653-659, Aug. 1978.

69. M. Morf and T. Kailath, "Square-Root Algorithms for Least-Squares Estimation", IEEE Trans. Autom. Contr., Vol. AC-20, No. 4, pp. 487-497, Aug. 1975.

70. M. Morf, B. Levy and T. Kailath, "Square-Root Algorithms for the Continuous-Time Linear Least-Square Estimation Problem", IEEE Trans. Autom. Contr., Vol. AC-23, No. 5, pp. 907-911, Oct. 1978.

71. B. D. Tapley and C. Y. Choe, "An Algorithm for Propagating the Square-Root Covariance Matrix in Triangular Form", IEEE Trans. Autom. Contr., Vol. AC-21, pp. 122-123, Feb. 1976.

72. E. A. Baptiste and E. Cosmidis, "Kalman-Bucy and Leiniotis Filters: Comparison and Interconnection", Proc. 1979 Conf. on Modeling and Simulation, ISA, Pittsburgh, PA.

73. D. G. Lainiotis, "Optimal Adaptive Estimation: Structure and Parameter Adaptation", IEEE Trans. Autom. Contr., Vol. AC-16, No. 2, Apr. 1971.

74. D. G. Lainiotis, "Partitioning: A Unifying Framework for Adaptive Systems, II: Control", Proc. IEEE, 64, No. 8, 1976.

REFERENCES (Continued)

75. D. G. Lainiotis, "Partitioned Estimation Algorithms, I: Nonlinear Estimation", J. Info. Sci., 7, No. 3, Nov. 1974.

76. D. G. Lainiotis, "Optimal Nonlinear Estimation", Int. J. Control, 14, No. 6, Dec. 1971.

77. D. G. Lainiotis (Ed.), Estimation Theory, American Elsevier, New York, 1974.

78. D. G. Lainiotis, "Partitioned Riccati Solutions and Integration - Free Doubling Algorithms", IEEE Trans. Autom. Contr., AC-21, No. 5, Oct. 1976.

79. D. G. Lainiotis, "Generalized Chandrasekhar Algorithms: Time Varying Models", IEEE Trans. Autom. Contr., AC-21, No. 5, Oct. 1976.

80. S. K. Park and D. G. Lainiotis, "Monte Carlo Study of the Optimal Nonlinear State Vector Estimator for Linear Model and Non-Gaussian Initial State", Int. J. Contr., 16, No. 6, Dec. 1972.

81. D. G. Lainiotis, S. K. Park and T. Revuluri, "Optimal State-Vector Estimation for Non-Gaussian Initial State-Vector", IEEE Trans. Autom. Contr., AC-16, No. 2, April 1971.

82. D. G. Lainiotis and K. S. Govindaraj, "A Unifying Approach to Linear Estimation via the Partitioned Algorithm I: Continuous Models", Proc. IEEE Conf. on Decision and Control, New York, Dec. 1975.

83. D. G. Lainiotis and K. S. Govindaraj, "A Unifying Approach to Linear Estimation via the Partitioned Algorithms, II: Discrete Models", Proc. IEEE Conf. on Decision and Control, New York, Dec. 1975.

84. D. G. Lainiotis, "Partitioned Riccati Algorithms", Proc. IEEE Conf. on Decision and Control, New York, Dec. 1975.

85. D. G. Lainiotis, "Generalized Chandrasekhar Algorithms: Time-Varying Models", Proc. IEEE Conf. on Decision and Control, New York, Dec. 1975.

86. D. G. Lainiotis, "Fast Riccati Equation Solution: Partitioned Algorithms", J. Computer and Elect. Eng., 2, No. 6, Nov. 1975.

87. D. G. Lainiotis and D. Andrisani, II, "Multipartitioning Linear Estimation Algorithms: Continuous Systems", IEEE Trans. Autom. Contr., Vol. AC-24, No. 6, Dec. 1979.

88. B. J. Eulrich, D. Andrisani and D. G. Lainiotis, "Partitioning Identification Algorithms", IEEE Trans. Autom. Contr., Vol. AC-25, No. 3, pp. 521-525, June 1980.

89. T. Kerr and L. Chin, "The Theory and Techniques of Discrete-Time Decentralized Filters" to appear.

90. M. Shah, "Suboptimal Filtering Theory for Interacting Control Systems", Ph.D. Dissertation, Cambridge University, Cambridge, England, 1971.

91. B. S. Garbow et al., Matrix Eigensystem Routines - EISPACK Guide Extension, Lecture Notes in Computer Science, Vol. 51, Springer-Verlag, New York, 1977.

92. E. Cornides, "Analysis of Estimation Algorithms", Ph.D. Dissertation, Department of Electrical Engineering, State University of New York at Buffalo, 1981.

93. T. H. Kerr, Stability Conditions for the RELNAV Community as a Decentralized Estimator, Intermetrics Report No. IR-480, March 1980.

DESIGN OF REAL-TIME ESTIMATION ALGORITHMS FOR IMPLEMENTATION IN MICROPROCESSOR AND DISTRIBUTED PROCESSOR SYSTEMS

VYTAS B. GYLYS
Senior Member of Technical Staff, Texas Instruments Incorporated

P.O. Box 405, Lewisville, Texas, 75067, USA

SUMMARY

Implemental design of real-time estimators is viewed as the mapping (mechanization) of estimation algorithms into a software structure designed to function in a real-time environment. This means meeting real-time constraints of the problem and overcoming the limitations of hardware while attaining the specified functional performance. Use of a microprocessor or a distributed system of small computers as hardware imposes additional constraints on mechanization of real-time estimators: the "smallness" of individual processing elements, arranged in a loose federation, must be overcome.

Distributed hardware architecture is suitable for a class of real-time control systems, built around an estimator, that do not require sophisticated and centralized management of resources. Such systems are designed to work under predetermined maximum loading and are characterized as being boundedly loadable. Small- to medium-scale estimation schemes used in navigation, guidance, and control applications typically belong to this class.

The article proposes the use of multiprogrammed processing, supporting execution of concurrent processes, as a suitable processing environment for real-time estimators which are to be implemented on the above described hardware. Concurrent processing is presented as the basic technique for overcoming the hardware limitations and for meeting the real-time constraints of the estimation problem.

Design of a decentralized real-time operating system for controlling multiprogrammed execution of processes in a distributed system is outlined next. This operating system consists of autonomous, local real-time executives, which operate under fixed allocation of resources. Process scheduling and communications are discussed.

The established structure of real-time software and process control can then be utilized to mechanize estimation algorithms as concurrent processes. Several known schemes for decomposing a Kalman filter into concurrent processes are illustrated.

I. INTRODUCTION

A. TECHNOLOGICAL PERSPECTIVE

During the last decade, adoption of microprocessors and distributed systems of micros as hardware for implementation of real-time estimation schemes has proliferated the use of Kalman filters in new and, until recently, undreamed applications. The new hardware technology and improved theoretical and implementational understanding of Kalman filters have made many of these applications technically feasible. Due to dramatic decrease in the cost of hardware, Kalman filters - even in applications for which a few years ago only a very crude estimator would have been considered - have become economically attractive. In many newly emerging real-time control systems built on the methodology of artificial intelligence, Kalman filters will continue to play a central role. The new artificial intelligence techniques will enable the system to "see" and perhaps, to some extent, interpret its own environment. However, the need for a mechanism which would estimate the state of the system will continue to persist. In fact, such future systems will place even higher technical demands on estimation, partly due to increasing availability of smart sensors (designs of which also exploit the microprocessor and special chip technologies) and of expanding variety of measurements from such sensors.

We complement the above introductory philosophical remarks by quoting R.K. Smyth [Smy], who states that "great changes, propelled by microprocessor technology, are sweeping avionics and control" and that they will continue in the decade ahead. He further identifies the disciplines and technologies which, according to him, will most likely influence the next generation of systems. They are "NAVSTAR/GPS, ring-laser gyros, air-traffic-control update, digital fly-by-wire, full-authority digital propulsion control, flat-panel displays and integrated data-control centers, modular and distributed avionics architecture, cost of airborne data processing, digital buses including potential of fiber optics, modern control theory including Kalman filters and optimal-state estimation, direct digital synthesis, high-level software languages, VLSI/VHSI circuits, and actuation and power-generation devices made possible by new rare-earth magnetic materials."

Successful integration of such diverse factors requires a multidisciplinary approach if the potential synergism is to be exploited. This requires that each specialist member of a multidisciplinary design team understand the other disciplines affecting his aspect of design. As an example, consider a control specialist who is a member of the team entrusted with the design of a real-time estimation scheme. Such a person should be capable of not only performing his traditional functions (such as selecting an appropriate system model and the right estimation algorithms) but also stating the functional requirements for sensors (whose measurements the estimation scheme is to use) or specifying the mechanization of estimation algorithms in the form of real-time software. Current digital technology makes designing smart sensors, capable of aiding the estimation process, a practically attainable goal. (These are the sensors capable of producing not only the basic estimation measurements but also some system identification data. Having such extra information continuously furnished to the estimator may preempt the

formidable problem of determining the parameters of noise statistics under severe timing constraints.) In the same vein, understanding the possibilities offered by recent ideas in operating system theory and software engineering may greatly facilitate the transformation of the mathematical model of an estimation scheme (i.e., of estimation algorithms) into robustly working real-time software. This latter aspect of real-time estimator development is the main topic of the present exposition.

B. IMPLEMENTAL DESIGN OF ESTIMATORS

Development of Kalman filters (or, more generally, of recursive estimators) for real-time applications roughly involves three aspects of design: system modeling, algorithm design, and implemental design. System modeling, although critically important, is very dependent on the problem at hand and so it is not discussed in the sequel. In a wide sense, algorithm design addresses not only the design of kernel estimation algorithms but also the design of complementary procedures, such as one whose function is to detect and then respond to detected nonwhite noise in measurements. Algorithm design is extensively covered in current literature. The same cannot be said, however, about the implemental design, i.e., about the process of mapping the algorithms into a system of software procedures which, when executed on some target equipment, will interact correctly with the environment and among themselves and also will satisfy the real-time constraints of the problem. One possible reason for this paucity of attention in literature to implemental design of real-time estimators is that it cannot be discussed in a mathematically concise way. The other reason, which is probably more fundamental, is that it has been viewed as an exclusive domain of programmers.

C. SCOPE AND OBJECTIVES

The present article addresses estimation and control system specialists who mainly are experts in system modeling and algorithm design but who also would like to learn more about transforming their designs into working real-time schemes. As it will become evident from the sequel, such a control specialist must depend on system programmers or on software engineers who also are members of the design team, for advice and contributions. Since the design team is often led by the control specialist, it is important that he know how to communicate with his software counterparts and state the requirements for implemental design. Thus, one of the main objectives of the present article is to help a control specialist acquire the technical background for performing these functions.

The level of the present article is introductory: we assume that the control specialist knows little about the design of real-time software, especially about its processing environment. Hence, in the sequel, we do not discuss a wide spectrum of design possibilities in generality but rather concentrate on a few approaches that have been experimentally proven to work, explaining them sometimes in detail. We feel that this will enhance the tutorial value of the exposition. One aspect of implemental design, which is stressed, is the real-time process control environment for estimation algorithms. However, this is not intended to be a general exposition of real-time operating systems, so here again we take a narrow path through the labyrinth of issues concerned with operating system and software engineering design.

We further narrow the scope of discussion to real-time estimator design for implementation on distributed microprocessor systems, although we use the term "microprocessor" generically: it refers to almost any small computer. However, this restrictive assumption about hardware has implications on the class of control systems considered.

D. BOUNDEDLY LOADABLE VS. NONSATURABLE SYSTEMS

Currently, distributed systems of small computers are used mainly for small to medium size control systems. Examples of such systems built around estimators are navigation, guidance, and flight control systems, as well as various types of artificial intelligence systems in which the dynamics of system state can be modeled stochastically and must be estimated. We refer to such systems as <u>boundedly loadable</u>; i.e., they are a priori designed to take a certain maximum loading, perhaps in the form of fixed maximal rates for measurement inputting and processing. We do this for two reasons: (1) typically, such control systems do not require sophisticated resource allocation algorithms and so they are easier to understand; and (2) they are the systems most likely to be implemented on the computers of the type considered here.

In contrast, we could also have considered estimator design for large-scale control systems which are smart enough to select intelligently any additional loading past a near-saturation point so as not to overload themselves and, at the same time, to perform their mission in some optimal way. For convenience, we shall call them <u>nonsaturable</u> systems. Various large-scale air traffic control and defense systems, discussed in literature during the sixties and seventies, are examples of the latter type of control systems. Although many such systems use estimators as kernel algorithms, they would not (even nowadays) be implemented on distributed systems of small computers. More likely, multiprocessor architectures, augmented with special (such as parallel) processors, would be applied to such a problem. They would also require the use of complex processing resource allocation algorithms and, thus, of a very different philosophy of real-time processing control from the one presented in the sequel. The use of sophisticated resource allocation probably would make centralized processing control preferable. Such processing control schemes are not only more difficult to understand but also much more difficult to validate. Instead, we shall describe what could be characterized as a fixed allocation, decentralized processing control scheme, i.e., decentralized over the processing elements of a distributed system.

The total state of a nonsaturable system is partly defined in terms of the states of N objects currently handled (processed) by the system. For example, if each object represents a tracked aircraft, the state of an object may consist of the description of its position, velocity, and some parameters. As the number of handled objects increases and as a nonsaturable system approaches its saturation point, the system must decide (typically on the basis of some risk function) which present objects could be deleted at a minimum risk in order to create processing capacity for newly incoming and

possibly more critical objects. Often the estimation procedure for all N objects is nearly identical. In such a situation, the use of large scale computers with parallel or vector pipeline arithmetic units is attractive. This leads to estimators which differ in structure and mechanization from those considered in the present exposition, for such large-scale computers can be best exploited if all N parallel estimation processes are centrally controlled.

In contrast, a boundedly loadable system is a priori designed to handle no more than B objects, where B is a fixed, small positive integer. Simple, fixed allocation, decentralized real-time process control schemes are effective for boundedly loadable estimation systems.

E. OUTLINE

Typical microprocessor system architectures under consideration consist of small generic building blocks, such as microcomputers, stand-alone direct access memory units, and digital interface units, the latter needed for communications among microcomputers or with the "outside" world. These building blocks are interconnected by means of bus systems and direct access global memory units. Smyth [Smy] mentions the DAIS and the Draper Laboratory Fault-Tolerant Multiprocessor as prototypes of such new architectures. We shall loosely refer to such architectures, formed from "small" building blocks, as "distributed microprocessor systems." Consequently, we begin (in Section II) by examining the implications of microprocessor use on algorithm design and implementation. Then (in Section III), we proceed to the problem of controlling in real-time the computational processes resulting from the software organization recommended in Section II. Next (in Section IV), we review current practices of Kalman filter design for real-time applications without paying too much attention to the complications imposed by the hardware under consideration. The next section (Section V) addresses the problems arising due to hardware limitations and timing constraints. It illustrates techniques for overcoming many of these problems by decomposition of filter algorithms into concurrently executable procedures. A summary follows in Section VII. Three appendices follow at the end: Appendix A summarizes the standard Kalman filtering equations; Appendix B reviews the so-called U-D factorization algorithms for a Kalman filter; and Appendix C states the estimation problem is GPS navigation, which is referred to in several examples throughout the text.

II. PROGRAMMING FOR REAL-TIME DISTRIBUTED SYSTEMS

A. HARDWARE ARCHITECTURE

For the development of our main theme, we need a generic model of hardware architecture. Thus, we view distributed systems considered here as built from constituent computers by interconnecting these computers either through shared, directly accessed global memory units or through data links (buses). These links may vary in speed, parallelism, and length. Obviously some distributed systems use both types of interconnections, i.e., direct access global memory units as well as buses.

For our purposes, a constituent processing element of any distributed system under consideration is assumed to be a small computer, not necessarily a "microcomputer" in the strict sense of the term, which consists of a CPU (an instruction/arithmetic processing unit) and interconnection ports/devices. It usually has some local (or private) memory and may also have special devices, such as a floating point arithmetic unit, connected to it. In the sequel, such a constituent computer of a distributed system, regardless of whether it is a microcomputer or a computer of some different type, is called a processing element (PE). As a special case considered, the entire distributed system may consist of a single PE. If a PE accesses both global and local memories, then its instruction address space must possess facilities to address both types of memories.

Another building block needed for the assumed model of distributed systems is a global data memory unit. "Data" refers here to the read-write and random access properties of such memory, while "global" emphasizes that such a memory unit is accessible from at least two processing elements. In constrast, a processing element may have two types of local memory: local data memory for storing the problem data which does not have to be communicated to other processing elements and local program memory of the read-only type for storing the instructions of programs residing in that processing element. In our model of a distributed system, executable instructions of a program are always assumed to be stored in the local memory of a processing element.

As an aside, many ideas presented in the sequel apply to hardware architectures more general than one just introduced. For instance, we could have considered hierarchical distributed systems in which some processing elements themselves are distributed systems; or distributed systems each processing element of which is a multiprocessor system by itself, consisting of several CPUs interconnected by common access memory modules. But the generic model introduced earlier, which is simpler and less centralized than the architectures mentioned in the present paragraph, can be effectively used with a relatively simple, fixed allocation operating system. Furthermore, it is sufficient for the applications considered here. Hence, it will be assumed in the sequel.

B. PROCESSES EXECUTED IN A DISTRIBUTED SYSTEM

The notion of a (computational) process is fundamental in modern theory of operating systems and is discussed in recent texts on operating systems and system programming (e.g., [Cof 73], [Fre], [Gra], and [Han]). It is also used extensively in the sequel to explain our models of real-time software and real-time process control. The term "process" will be used to describe the behavior or, say, the dynamics of a computer program that is stored (or is "residing") in a computer.

Freeman ([Fre], p. 108) further elaborates this concept by noting that "a program specifies a single sequence of actions. When we admit the possibility of a program stopping before it finishes execution (to wait for a signal or because the processor has been taken away from it, for example), we must associate with each program in execution some information that records its current state (what

the value in the instruction counter is, where the program is in memory, and so on). The concept of a program being "in execution," but not necessarily being executed at a particular moment, is what we shall now term a process. Operationally in a system, a process is a program specifying a sequence of actions and the associated information that represents its current state."

Thus, a single process is just a generalization, or an abstraction, of a processor moving through the text of a computer program. However, one additional idea is implied: there exists a controlling mechanism (an operating system) which is keeping track of the state of the processing resources (hardware, data, and programs) being used by or reserved for the executing programs. Keeping track of the state becomes important if the execution of a program can be temporarily interrupted by the operating system in favor of another waiting program.

Analogous to finite automata theory, it is convenient to characterize a process by describing the process states which such a process may attain and by defining the rules that govern transitions among these states. When a program is in execution or is considered by the operating system to be scheduled and executed, we say that the process created by that program is in an active state; otherwise, i.e., the process is said to be in an inactive state. Typically, there is only one inactive state.

The concept of a process is a powerful tool for understanding, modeling, and designing a computer system in which several programs reside in memory simultaneously so that the central processor of the computer (which in our case is a processing element of the distributed system) keeps switching among them according to some scheme or schedule. In such a situation, processes created by these programs are said to be executing concurrently, i.e., interleaved in time. The resulting processing environment is called multiprogrammed processing. This is the processing model which we assume for estimation schemes executed on a distributed system, or even on a single processing element.

When a distributed system is operating, several processes may be executed concurrently. Concurrent execution of processes can occur in several ways. On a single processing element of a distributed system, for example, processes may be executed interleaved in time by dividing the time line into segments and alternating the processes among segments such that the process which starts execution at the start of a segment is interrupted at its end, at which time the use of the processor is passed to another process. Thus, concurrent processes may occur under multiprogramming, even on a single processor. Another form of concurrent process execution in a distributed system arises when processes are executed simultaneously and possibly asynchronously in several processing elements of the system.

The hardware facilities needed for data communications among two communicating concurrent processes depend on where these processes are executed. If they are executed in interleaved time fashion within the same processing element, the local data memory unit of that processing element, provided that such a unit exists, may be used for communications among processes; otherwise, a global data memory unit, accessible from that processing element, is needed. On the other hand, if two communicating concurrent processes are executed on different processing elements, then two (not necessarily mutually exclusive) possibilities exist: either a mutually accessible global memory unit is available or data links (buses) interconnecting the system are needed.

Designer of a real-time estimation (or, more generally, of a realtime control) system is responsible for structuring his algorithms so that they result in computational processes which properly cooperate among themselves in real-time; i.e., they are appropriately synchronized and can correctly communicate (exchange) data. Selection of an appropriate programming language and access to convenient real-time process management utilities (the latter to be furnished by the real-time operating system) will make designer's work easier. Still, in order to be assured about the implementability of design, he must conceptually understand the logical consequences of process synchronization and communication requirements and must be capable of translating these requirements into his design. The purpose of Section III in the sequel is to introduce process management concepts and techniques that have proven themselves in implementation of real-time estimation schemes.

C. SOFTWARE MODEL

Workload partitioning, algorithm scheduling, and memory sizing are critical tasks in design of real-time control (in our case, estimation) systems for implementation under severe processing constraints. These tasks are begun early in the development cycle, usually during preproposal time investigations, and are reiterated many times thereafter until a design that satisfies requirement specifications emerges. The capability to transform algorithms into working real-time software is the key to success in this endeavor. This process of implemental design is greatly helped by having a suitable model of software architecture. As will become evident in Section III, such a model is also needed for design of a real-time operating system.

The software model introduced next in the sequel is intended for a boundedly loadable system which is to be implemented on a distributed system of small computers or even on a single microprocessor. This model assumes that some extended form of FORTRAN is used as the programming language. It is mentioned later in the sequel how a software model for an ALGOL-like language, such as ADA, would differ from the model to be introduced next.

MODEL:

a. The entire real-time applications software implementing an estimation scheme is partitioned into a set of programs and data sets.

b. Each program contains a main procedure and may also contain subprograms (subordinate or lower-level procedures).

c. Each program can create precisely one real-time process (which may remain inactive), so there

is one-to-one correspondence between programs and processes. (This assumption is not too restrictive for boundedly loadable systems implemented on small computers. Besides, the resulting implementation can be more easily understood and tested than that of a system with no restrictions on the extent of reentrancy. Here, a program unit is said to be reentrant if it may be shared, or executed concurrently by several processes at a time.)

d. Subprograms of the following two types are admissible:

- Private subprograms - such a subprogram belongs to a single program and, by Assumption c, to a single process, because it may be invoked only from that program;
- Shared subprograms - such a subprogram may be invoked from several programs and, thus, used by the processes generated by these programs. Shared subprograms must be implemented as reentrant procedures, but only as reentrant procedures of a priori known maximum concurrency.

e. Fixed allocation of programs to a processing element is used: at design time, each program is assigned to a single processing element and is never "split" between several processing elements. In the event that software redundancy can be tolerated, the present assumption does not forbid assigning replica of a program (or of a subprogram) to several processing elements, or even to a single processing element.

f. With Assumptions c and d, data sets of applications programs can be divided into three hierarchical globality levels: interprocess (interprogram) communication data sets; intraprocess (intraprogram) communication data sets; and local data sets. For example, when FORTRAN (possibly extended to handle limited reentrancy) is used as a programming language, interprocess and intraprocess communication data sets are implemented as ordinary labeled COMMON blocks; local data sets, as locally declared data with special provisions made for handling local variables in reentrant procedures. Furthermore, variables and constants are always placed into separate data sets. In order to distinguish between two types of constants -- (i) physical constants, such as the speed of light or the equatorial radius of a reference ellipsoid, and (ii) design parameters, such as the length of a state vector, which may change as the design progresses, but become constants by the time it is completed -- they are put into separate data sets.

Although the use of an ALGOL-like programming language, such as ADA [Weg], instead of FORTRAN would hardly perturb the model described above, it would affect the format and the structure of the source program. First, the nested block structure of ALGOL-like languages such as ADA would facilitate hierarchical nesting of procedures (private to a program) and the corresponding nesting of data sets according to the globality (scope) of data access. Furthermore, ADA provides basic language constructs for defining and implementing process control and synchronization mechanisms, such as the mechanism for synchronized communications among concurrent processes discussed in Section III. (For more information on this subject, refer to discussion of multitasking in [Weg]).

D. IMPLICATIONS ON ALGORITHM DESIGN

Adoption of distributed microprocessor systems as hardware has far-reaching implications on algorithm design. In order to meet the real-time response constraints while not exceeding the throughput capacity of individual processing elements, the entire estimation procedure must be partitioned into concurrently executable and interacting processes. The main implications of the resulting workload partitioning are that some algorithms will be decomposed into a set of concurrently executable smaller algorithms, the functional performance of which will not be as good as that of the original algorithm.

In the second part of the present exposition, we review several common decomposition schemes for a Kalman filter. It will suffice to mention at this point that for a recursive estimator, such as a Kalman filter, the covariance processing and the computation of Kalman gains are time-consuming procedures on a microprocessor and so may become prohibitively expensive if the microprocessor has no hardware-implemented floating arithmetic and if, consequently, all floating-point computing must be performed in interpretive form. On the other hand, there may exist a requirement, or just a need, to process the incoming measurements at a rate which would exceed the processing capacity of a processing element, i.e., which would "bust" its time line. As explained in the second part of the present exposition, this problem has several solutions, each of which results in suboptimal performance.

According to our earlier characterization of distributed microprocessor systems, it would seem that such a system could be incremented in small steps by adding to it, on the basis of need, processing elements, global data memory units, and other special boxes. Besides, hardware costs are relatively low. Then what are the reasons for having to struggle with an austere hardware budget? The answer to this question is that, in estimation applications considered here, severe constraints are typically imposed on the power consumption, volume, and weight of equipment. Besides, if the equipment under consideration is to be manufactured in large quantities, even small savings in cost per unit count. Another reason is the "smallness" of individual processing elements. For example, a single element may not have sufficient throughput capacity to accommodate a Kalman filter with a minimally acceptable rate of measurement processing. On the other hand, we may not want to split the filter algorithm among two processing elements for reasons such as the complexity or degraded performance of the modified algorithm.

E. PARTITIONING OF WORKLOAD

During a design cycle, the processing workload resulting from algorithms is several times repartitioned into concurrent processes until a satisfactory partitioning is obtained. We say that a workload partitioning is acceptable if it (i) satisfies the overall hardware budget; (ii) does not overload individual processing elements; (iii) yields minimally required or better execution rates to time-critical algorithms; (iv) does not appreciably degrade the functional performance of algorithms

through their decomposition; (v) does not excessively complicate the overall structure of real-time software; (vi) does not cause too much real-time processing overhead through introduced concurrencies.

It is difficult to satisfy this long list of requirements without some compromises. However, the first four are essential to the implementability and performance of design.

Workload partitioning is a critical design issue in distributed real-time processing. Thus, several years ago, when the interest in such processing began to emerge, attempts were made to formulate the principles and derive the algorithms for optimal workload partitioning. Samples of work from that period are [Gyl 76] and [Jen]. Later design experience showed that it is best to take a heuristic approach, roughly based on the following principles:

(1) Distribute the workload over the processing elements (PEs) so as to minimize the level of communications among the processes executed on different PEs, thus minimizing the inter-PE data traffic and the use of global data memory, while not overloading individual PEs.

(2) Decompose the workload assigned to a PE into processes so that the tasks that must run at about the same rate and can be executed at about the same time are assigned to the same process and thus are implemented as part of the same program.

(3) Guarantee that all time-critical algorithms (in their original or in a decomposed form) will execute at acceptable rates.

Ideally, hardware requirements should be stipulated as a byproduct of completed workload partitioning, but in practice this is seldom the case.

It should be evident from the foregoing that quantitative prediction of processor and memory loadings must be obtained for each candidate partitioning before it can be evaluated. Thus, the activity called "software timing and sizing," is very critical.

F. PREDICTION OF PROCESSOR AND MEMORY LOADINGS

Real-time system designers often refer to the task concerned with the prediction of processor and memory loadings as software timing and sizing (ST&S). This task produces input data for workload partitioning and also evaluates candidate partitioning schemes. So it is reiterated several times during the development cycle of a real-time system.

Load prediction requires much clerical effort (compilation and tabulation of input data, performance of arithmetic, and generation of reports). Hence, these tasks must be computerized. After this has been done, the effort is essentially reduced to derivation of execution timing and memory occupancy data for individual modules of the currently tentative software model. Once the ST&S data base has been established by entering in it the initial timing and sizing estimates for individual modules, from then on it needs only to be updated as better estimates become available.

Even the derivation of execution timing and memory occupancy for individual modules can be expedited by means of automated data processing techniques. As an example, the following methods have been found to be useful:

(1) Extending the compiler of the programming language to enable it, as a byproduct of compilation, (i) to segment the source code into blocks, each block ending with a branch operation, then to time each block while using an inputted timing model of the target machine; and (ii) to estimate the memory requirements of each module of executable code and of each data set defined in the source code.
(2) Using an instruction-level simulator to derive a timing model of each program, based on the a priori inputted probabilities or on the experimentally observed frequencies of various execution paths.

With the information provided by (i) and (ii) of Method 1, an experienced analyst can quickly time the most critical execution paths and compute the memory requirements of his software design. Method 2 produces a probabilistic model of processor loading and data memory occupancy, and so its outputs in a sense complement those of Method 1.

After either (1) or (2) has been accomplished the first time, the obtained processor timing and memory sizing data can be entered in the software model data base. Thereafter, as the design progresses toward maturity, this model and the processor and memory loading predictions for it need only to be iteratively refined.

The techniques illustrated above presuppose the availability of some source code. Often, the source code of key algorithms becomes available early during the development cycle and can be used to bootstrap the timing and sizing process by means of the above described techniques. Typically, these algorithms are programmed early in the high-level language of the ultimate real-time code for performance analysis simulations on a large computer in nonreal-time mode.

In the event that the source code of key algorithms is not available when the timing and sizing process must be bootstrapped, one may resort to mathematical timing and sizing models of key algorithms. For a Kalman filter or a similar estimator, such a model is formulated in terms of the state vector and measurement vector lengths: it predicts the timing in terms of basic arithmetic operation counts and the memory occupancy in terms of memory needed to store the principal vectors and matrices. References [May], [Tho], and [Bie] contain such loading prediction models for the estimation algorithms considered in the sequel.

G. DATA IDENTIFICATION

Experience in implemental design of real-time estimation systems has taught a bitter lesson about the identification of time- and source-dependent data and events. Implicit identification techniques, based on the order in which data arrives (or is generated) or dependent on the location where it is placed, are often favored by novice designers. But they are dangerous: some data may arrive late or never; some sources may intermittently fail to send data; the implicit identification of data by position in an array may be perturbed by the deletion or addition of sources. Also, implicit identification techniques lead to rigid and cumbersome implementations.

Explicit identification of data and events and recording of their reference times constitute a safer approach. It also leads to a more flexible implementation in the sense that the meaning of real-time data no longer depends on the time when it becomes available and on the memory location where it can be found.

Technically, explicit identification means that every data group (or record of an event) which is time-dependent is time-tagged by its reference time. Similarly, every source-dependent data group is tagged with the identification (ID) of its source. For example, suppose that a real-time Kalman filter, used for target tracking, operates on range and range-rate measurements of tracked targets. Then every simultaneous batch of range and range-rate measurements is time-tagged, say, with the estimated time at which the radar signal is reflected from the target, or with the observed time at which the signal is returned. Furthermore, as new targets are detected and go into tracking, they are assigned explicit and unique target IDs.

To facilitate the time-tagging of real-time data and events, a clock is needed. Such a clock, which will be called system control clock, is usually hardware-implemented and is driven by an oscillator. The time of this clock must be available to all processing elements of a distributed system. In Section III.C, we outline how the system control clock can be used to synchronize processes over the entire distributed system. Such a clock would be needed even in a uniprocessor (not necessarily multiprogrammed) real-time system. Computational processes retrieve the time of this clock by calling a special subprogram.

H. SOFTWARE DEVELOPMENT METHODOLOGY AND TOOLS

During the seventies, a great deal of progress was scored in the areas of software engineering and software development management. Some of the events contributing to this progress were the emergence of a robust programming style, known as structural programming, of software development management techniques such as the chief programmer's team, and of computer-aided software development and software management tools such as interactive program development terminals, data base management systems, or programming languages amenable to structured programming and concurrent processing.

These techniques and tools are useful in development of real-time software for the multiprogrammed processing environment, but they are well covered in the literature and thus are not addressed here. The only thing which we want to note is that software development, especially its testing and validation, for the applications and processing environment addressed here requires a hierarchical sequence of simulations.

One starts with high-level functional simulations of key algorithms in order to validate their performance and to determine the required processing rates. These high-level simulations are usually put together and performed by the control specialist responsible for algorithm design. They are performed off-line, (i.e., not in real-time) on a large-scale computer system. The insights obtained from such simulations facilitate the timing and sizing of software and the partitioning of workload for real time.

As the design process progresses and as the control specialist begins to think about the implementation of algorithms for the real-time processing environment, he starts (while guided by the feedback from the performance analysis, the tentative workload partitioning scheme, and the results of software timing and sizing) to modify and restructure them. In this effort, he continues to use the off-line simulator as a testbed. If this iterative process of design refinement is continued long enough, the modeled real-time software and algorithms begin to look more and more similar to the ultimate product. At the same time, the level of simulations progressively goes down as more details are modeled, simulated, and investigated.

Ultimately, the off-line simulation process eventually reaches a point of diminishing returns, primarily because of the difficulties in creating sufficiently realistic simulation scenarios, needed for complete validation of design, and in modeling with fidelity interactions among concurrent processes. Also at this time, partly tested real-time software for the target hardware usually becomes available. (This availability can be speeded up by copying pertinent portions of off-line simulation software and then embedding the copied software into the prepared control structure of real-time software.)

The next step is to switch to on-line (or real-time) simulations in which the actual real-time hardware and software are driven by special test equipment. In order to serve its purpose, such test equipment must

- Be capable of generating (or acquiring) at real-time rates the measurements and other system inputs that would be highly similar, if not identical, to the actual measurements (inputs) of the target operational system.
- Possess built-in facilities for collecting and reporting performance data.

For comparison, it is desirable that the performance analysis reports generated in simulations on the special test equipment be designed to look similar in form and contents to the performance reports

generated in off-line simulations. Finally, for testing various failure modes, the special test equipment must be capable of generating a wide range of extreme measurements and inputs. Examples of such special test equipment are briefly described in [Dam] and [Gyl 80].

III. CONTROL OF REAL-TIME PROCESSES

A. INTRODUCTION

Modern theory of operating systems, based on the concepts of process (task) and of process management, furnishes powerful tools for understanding and designing not only operating systems but also the applications software executed under the control of an operating system. (In the sequel, the term "applications software," in contrast to "system software," will refer to that portion of real-time software which implements control or estimation algorithms; analogous meanings will also be assigned to "applications program" and "applications process.") In real-time multiprogrammed computing, the applications processes are more intertwined with process state control than in non-real-time environment. Hence, the designer of such applications software needs to understand certain aspects of process management in order to be able to come up with a working implemental design. In contrast, a casual noureal-time scientific programmer, who programs in a high-level programming language, rarely needs to know much about the operating system beyond specifying his job to the operating system, getting it into the computer, and writing input and output statements. Thus, he can perform his functions nearly without any understanding of the actual environment in which his program is executed.

Understanding of the following aspects of process management, we think, is essential to a designer of a real-time multiprogrammed estimation (or control) system: resource allocation, process synchronization, process scheduling, and interprocess communications. Hence, the purpose of Section III is to review these as well as other related aspects of process management. Although it is not our intention to be digressed by a lengthy exposition of real-time operating systems, discussion in the sequel is essentially self-contained. (We shall henceforth use the term "real-time executive" or its abbreviated form "RT executive" to refer to a real-time operating system. We shall also use the terms "process management" and "process control" synonimously.)

B. FUNDAMENTAL ISSUES OF PROCESS MANAGEMENT

In the description of software model for a distributed system, we assumed multiprogramming, as the processing environment in the processing elements (PEs) of such a system. We also noted that concurrent processes executed within a processing element or in several different processing elements communicate among themselves via an exchange of problem and control data. The present section reviews fundamental issues of process management in the multiprogrammed multiprocessing environment in order to establish a perspective for the design approach described in the latter parts of Section III.

These fundamental issues, well known in operating system theory, are mutual exclusion, synchronization, deadlocks and their prevention, and interprocess communications. Critial regions and communication primitives are presented as technique for implementing mutual exclusion and synchronization. There are two additional issues of process management, allocation of memory resources and allocation of processor time (or scheduling), which will be addressed in Sections III.C through III.E. Due to the limitation of space, this exposition of fundamental issues is more concise than it ought to be. The issues are discussed only to the extent needed to make a control specialist aware of their existence and criticality. For a more detailed exposition, the reader is referred to recent texts on operating systems and system programming such as [Cof 73], [Fre], [Gra], [Han], or [Mad]. [Fre] and [Gra] are readable, elementary expositions of the topic, written mainly for aspiring system programmers; [Dam] and [Mad] are more detailed but still elementary texts, less advanced than [Cof 73].

1. Critical Regions

Consider a computer program, say $PROG_j$, and the process P_j created by execution of this program. A critical region (CR) of program $PROG_j$ is an executable segment of instructions in $PROG_j$, the executions of which may produce unpredictable and varying results if the values of some variables referenced from within this CR are changed by another process, say P_k, while P_j is executing the CR. Here, P_k is assumed to be a process concurrent with P_j. This may occur if (1) we do not know anything about the relative speeds of processes P_j and P_k and (2) we do not program $PROG_j$ and $PROG_k$ so as to prevent the unpredictable results.

If in the illustration of the preceding paragraph CR_j and CR_k are critical regions in programs $PROG_j$ and $PROG_k$, respectively, then two mutually exclusive possibilities exist: either CR_j and CR_k are critical with respect to each other (due to accessing of the same data set) or else CR_j and CR_k are mutually exclusive because each of them accesses a different data set. Hence, in order to be precise about a critical region, one must also specify the data set with respect to which the CR is critical. In the preceding paragraph, we could have done it by writing $CR_j(D)$ and $CR_k(D)$, where D would have referred to a mutually accessed data set.

The next example illustrates the use of critical regions in a Kalman filter.

EXAMPLE: Unpredictable results in a parallely mechanized Kalman filter.

Suppose that:

- Process P_e propagates the state vector $\underline{e}$, predicts the measurement vector $\underline{m}$, stores (perhaps occasionally) in a buffer the data needed for computation of linearized state-to-measurement transformations H, computes residuals, retrieves Kalman gains K computed by process P_c, and applies them to update $\underline{e}$.

- Process P_c propagates the state error covariance matrix P, retrieves the data for computation of H, computes Kalman gains K, and updates P.

Two segments in program $PROG_s$ are critical regions: one [say $CR_{s1}(H)$] contains the code for storing in a record (from which P_c reads) the data for computation of H; the other [say $CR_{s2}(K)$] copies (from the array into which P_c writes) for its own use the Kalman gains K computed by P_c. Similarly, there are two corresponding segments in program $PROG_c$ which also are critical regions: one [denoted by $CR_{c1}(H)$] retrieves the data for computation of H; the other [to be denoted by $CR_{c2}(K)$] stores the computed Kalman gains. If the relative speeds of proceses P_s and P_c are unpredictable (even if they are executed concurrently on the same processing element) and if no precautions are made to synchronize or otherwise regulate P_s and P_c in accessing mutually accessed data, then the results are unpredictable.

Section III.E reviews interprocess communication techniques for preventing such disasters. The preceding illustrations lead to the first fundamental issue of process management, mutual exclusion of communicating processes, which we discuss next.

2. Mutual Exclusion

Mutual exclusion of interdependent processes (of each process with respect to a mutually related critical region in its generating program) means that no more than one process can be in its critical region at a given time. We say that a process is in a critical region if it has already started the execution of the first executable instruction of this region but has not yet completed the execution of the last. Actually, the statement "the time when process P is in a critical region CR" refers to the entire time interval during which the above defined conditions hold, i.e., to the time interval spanned by the following two events: "P has entered CR" (P has started the execution of the first instruction of CR) and "P has left CR" (P has completed the execution of the last instruction).

We assumed in the above the principle of indivisibility of instruction execution. According to this principle, execution of an instruction, such as storing a quantity into a memory location or reading one from it, is an indivisible operation in the sense that the action performed by such an instruction cannot be interrupted after its execution has been started and before it is completed. By programming a short uninterruptable procedure, we can generalize this concept to an "indivisible macrooperation." In Section III.E, we discuss the use of such indivisible macrooperations (or procedures) in construction of communication primitives. These will be uninterruptable segments of code, sometimes implemented as uninterruptable subprograms, designed to protect entries to and handle exits from critical regions.

3. Synchronization

Synchronization of a process P_i with some other process P_j, where $i \neq j$, or with several other processes means ensuring that P_i will not proceed past some given point without an explicit signal, which P_i itself cannot generate due to lack of information about process P_j (or about several other processes). Hence, this information must be explicitly or implicitly provided to P_i from outside, i.e., by P_j, by other processes, or by the real-time executive. Note that strictly sequential processing on a single processor does not require any synchronization information.

A real-time executive passes synchronization information implicitly by scheduling processes for execution. Explicit exchange of synchronization information among concurrent processes generally requires the use of critical regions serviced by appropriate communication primitives.

4. Deadlocks

Two processes are said to be deadlocked if neither can continue until the other continues. A system deadlock occurs when all processes in the system become deadlocked.

Two concurrent processes P_1 and P_2, communicating through the execution of critical regions CR_1 and CR_2, respectively, may become deadlocked if the critical regions are improperly implemented: for example, if P_1 hangs up after entering CR_1 when it finds out that P_2 meanwhile has entered CR_2, and vice versa.

As pointed out in literature on operating system concepts (e.g., [Pre], p. 157), the occurrence of a deadlock is defined by the simultaneous coexistence of the following conditions:

(1) Processes claim exclusive control of the resources that they need for execution.

(2) Processes hold resources already allocated to them while awaiting additionally needed resources.

(3) Resources cannot be forcibly removed from the processes holding them until these processes no longer need them.

(4) There exists a circular chain of processes, such that each process in the chain holds some resources requested by the next process in the chain.

Although there is little probability of a deadlock in a typical multiprogrammed system designed even without any safeguards against deadlocks, it is imperative that any real-time system be designed so that deadlocks in such a system cannot occur, i.e., so the above four conditions can never be satisfied simultaneously. In applications considered here, we shall attain this objective by proper design of communication primitives and by requiring that no process by design be allowed to stay in a critical

region longer than some a priori fixed length of time.

C. DECENTRALIZED REAL-TIME EXECUTIVE WITH FIXED RESOURCE ALLOCATION

Next, we characterize a class of RT executives, several variations of which were successfully used in Phase I GPS (Global Positioning System) navigation user equipment sets designed by Texas Instruments (references [Dam], [Gyl 80], and [Upa] describe the navigation filters used in these sets). The main features of this class of RT executives are as follows:

(1) Decentralized control - each processing element (PE) has its own autonomously functioning RT executive which supports multiprogrammed execution of concurrent processes in the PE.

(2) Fixed allocation - there is fixed allocation of PEs to programs and of memory to global data sets, with the assumption that the software model defined in Section II.C is used (one implication of which is one-to-one correspondence between programs and processes).

(3) Synchronization of processes - processes executed in the distributed system and data communications among these processes are synchronized by means of periodic, systemwide interrupts. For this purpose, the time line is decomposed into consecutive intervals of fixed length Δt and an interrupt is broadcast systemwide at the end of each interval. Such intervals are often called fundamental time frames (FTFs). However, the jth processing element, PE_j, may be set up to respond only to every (n_j)th interrupt and to ignore the others. Typically, n_j is chosen so that $(n_j)\Delta t$ equals the period of the highest-rate, periodically executed process in PE_j.

(4) Restricted Reentrancy - since (according to the software model outlined in Section II.C) there is one-to-one correspondence between the programs and the processes created by these programs, subprograms concurrently shared by at least two processes are implemented either as uninterruptable procedures (if they execute fast) or else as reentrant procedures of a priori known maximum reentrancy.

(5) Scheduling - three types of processes are admissible and characterized according to the way they are scheduled:

- Cyclic (C-) processes - such a process is executed at a fixed rate, with the stipulated execution rate guaranteed to be met.
- Deadline (D-) processes - each time when such a process is scheduled, the deadline of its execution completion is specified; the RT executive tries to meet but does not guarantee this deadline.
- Background (B-) processes - such a process is allocated all the processing time of a PE that remains after (or between) execution of foreground and deadline processes; at any time, at most a single active background process is allowed in a PE.

Scheduling of processes is further discussed in Section III.D.

(6) Interprocess communications - depending on the nature of interaction between two processes, only the first or both of the following interprocess communication types are permitted without further reservations than those stated below:

- Critical regions (sections), protected/controlled by the WAIT and SIGNAL communication primitives, may be used to implement data communications among any two concurrent processes.
- Data buffers, whose access is controlled by a SET/RESET flag, may be used to implement one-way communications between two cyclic processes, scheduled at the same rate, whose executions are usually not interleaved in split fashion; i.e, if A and B are two such processes, then when process A starts an execution instance, it will complete this execution before B can start its next execution instance.

Interprocess communications are further discussed in greater detail in Section III.E.

D. PROCESS SCHEDULING AND PROCESS STATE CONTROL

1. Scheduling Philosophy and Requirements

Process scheduling allocates the processor time to processes and thus determines when processes will be executed. With our model of decentralized process control, all processes assigned to a processing element will be scheduled independently of processes executed on other processing elements.

Since scheduling also influences the structuring of algorithms into concurrent processes, scheduling is an important issue of real-time system design. Scheduling can be best explained through process states and process state control. This is the approach which we take in the present section. But we proceed with discussion of process state control only to the extent needed to define selected scheduling strategies for the decentralized, fixed allocation scheme introduced in the preceding section. Coffman [Cof 73] and [Cof 76] discuss process scheduling on an advanced (abstract) level; references [Fre], [Man], and [Mad] deal with it on a more elementary (less mathematical) level. Literature on scheduling for real-time processing is typically difficult for a nonspecialist to follow and is mainly confined

to journal articles: references [Bas], [Ber], [Jor], [Man], and [Liu] supplement the narrow viewpoint on real-time taken in the present article.

As noted above, process scheduling may be best introduced by defining the states which a process of a specified type may attain and then by defining the rules governing state transitions. We do this, but in an informal fashion. Detailed design of a scheduler for a real-time executive is a task which is usually delegated to system programmers; thus, a control specialist is seldom concerned with such details. However, to do his part of design, he needs to understand, in addition to the information conveyed by the state transition graphs, the scheduling priorities of processes, the facilities provided to him by the real-time executive for changing process states, and the attributes by which he can define or redefine a process or change its state.

Many strategies are possible for setting scheduling priorities. In literature (e.g., [Fre]), priority disciplines are often divided into two major classes:

- Static priorities - such a priority is set a priori in the sense that it cannot change while the process to which it applies is in an active state.
- Dynamic priorities - such a priority may change while the process to which it applies is in an active state.

In contrast to nonreal-time systems, scheduling of real-time processes requires some use of dynamic priorities or perhaps of a mixture of static and dynamic priorities. This becomes clearer if one recalls that in a general nonreal-time system, (1) very little is a priori known about the incoming jobs (processes); (2) incoming jobs are imprecisely characterized as they come into the system; and (3) the optimality criteria, such as maximizing the throughput without much regard for the turnaround time of individual jobs, make sense. Besides, the techniques for implementing schedulers operating on fixed priorities are better understood by programmers. These observations explain why static priorities are so widely used in general, nonreal-time processing.

On the other hand, the main objective of scheduling in the real-time applications considered here is to meet the response-time constraints required for specified performance while minimizing the cost of hardware or while staying within the allocated hardware budget. Designing a real-time scheduler operating on fixed priorities is not difficult if the available hardware resources are comfortably adequate. For example, Jordan [Jor] discusses a simple scheme for doing it by means of an a priori fixed multiharmonic scheduling pattern. We could proceed similarly, since nearly all processing load in the applications considered here is due to the algorithms which must be periodically reexecuted. We called them cyclic algorithms. Usually, it is not difficult to (1) identify a cyclic algorithm with the shortest period, say ΔT_0; (2) define a harmonic hierarchy of h periods $\Delta T_0, \Delta T_1, \Delta T_2, \ldots \Delta T_h$ such that $\Delta T_k = 2\Delta T_{k-1}$ for $k = 1,\ldots h$; and (3) assign every cyclic algorithm to a period class. Jordan then uses this technique as a basis for constructing a multiharmonic scheduling pattern. But such an approach, based on the estimates of algorithm maximum execution times, underutilizes the available processor resources.

A better approach is to classify all procedures and/or algorithms into three categories: (1) those with periodic rates that cannot be slipped because of the enormous penalty that would have to be paid otherwise (in estimation work, these typically are the procedures which logically control the estimation scheme but are not the estimation algorithms themselves); (2) those having period boundaries that represent desired but not absolutely required completion deadlines (in estimation work, these are the estimation algorithms); and (3) noncyclic procedures/algorithms which have to be executed only occasionally due to special conditions that may arise and which typically have no strict deadline (for example, a filter initialization procedure).

2. Process Types, Their States and State Transitions

Experience shows that nearly all real-time estimation schemes of the type considered here can be realized by means of the three types of processes (cyclic, deadline, and background) introduced in Section III.C. Next, we characterize these processes in greater detail than previously and, by means of the state graphs shown in Figure III.D-1, define their states and the rules governing state transitions:

(1) Cyclic (C-) processes - at the beginning of each new cycle (scheduling/execution period) of an active cyclic process, the RT executive automatically reschedules this process by putting it into the ready state so that the process is executed within each cycle and the events representing the starting time and the completion time of an execution instance are not separated by the boundary of a period. In other words, these two events are always located within the time interval spanning a single scheduling/execution period. As indicated in Figure III.D-1, a cyclic process, after it becomes activated by another process or by the RT executive, remains active until it becomes explictly deactivated (which is not shown in the state transition diagram) by a process or by the RT executive.

(2) Deadline (D-) process - such a process must be activated by another process or by the RT executive. With each activation, one needs to specify the completion deadline. The scheduler of RT executive does its best to meet the specified deadline or at least to minimize slipping the deadline. After the completion of each execution, a deadline process automatically returns to the inactive state.

(3) Background (B-) processes - at most one active background process at a time is allowed in a processing element of a distributed system. Such a process is then given all processor time that remains after all currently active cyclic and deadline processes have been serviced. After each complete execution pass, a background process automatically returns to the inactive state.

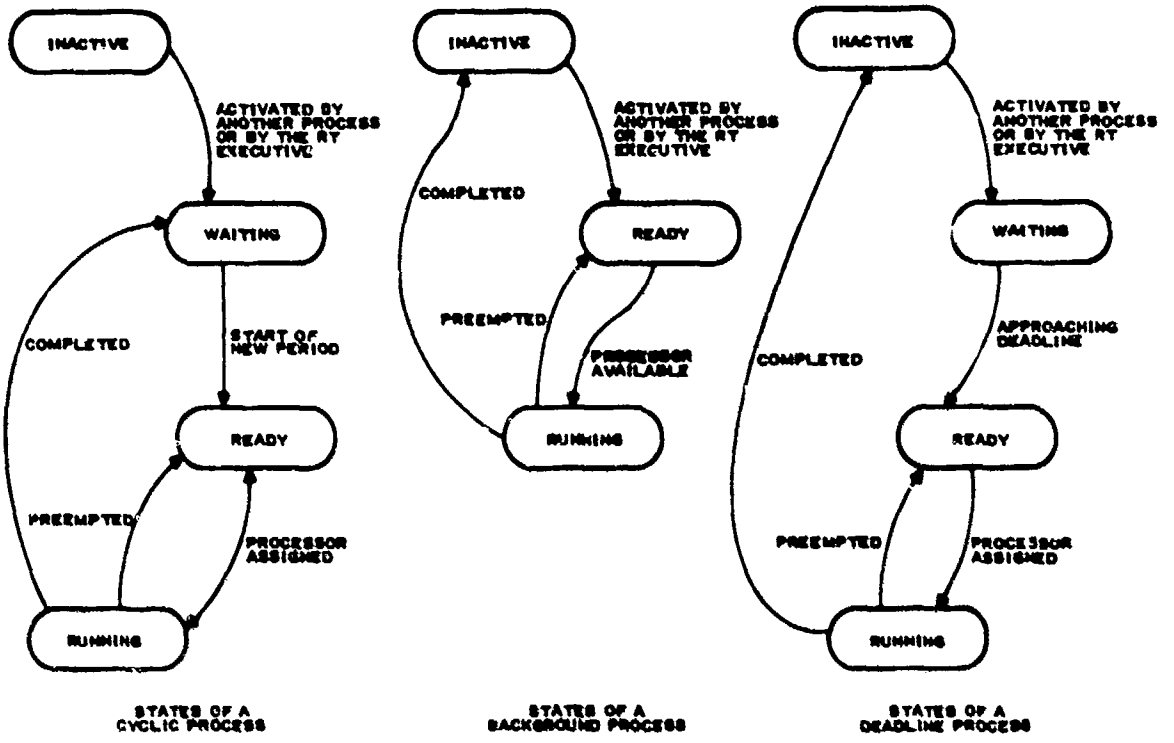


Figure III.D-1. State Transitions of Three Process Types. In order not to clutter the diagrams, it is not shown here (although it is assumed) that a process of any of the above shown three types may be deactivated from any of its active states.

It is further assumed that a process of any of the above three types, regardless of its current state, can be forcefully deactivated by invoking a special utility subprogram furnished by the RT executive. This provision facilitiates sudden reconfigurations of algorithms in real-time, which may be important in handling of emergency situations or of sudden changes in operational environment.

Complete specification of scheduling requires definition of priority rules for selecting a process (from the processes waiting in various ready queues) to be executed next on the processing element under consideration.

3. Priority Rules

Symbolically denote the priority of process x by $P_r(x)$. For processes x and y, write $P_r(x) > P_r(y)$ if x has a higher priority than y; $P_r(x) \geq P_r(y)$ if the priority of x is higher than or equal to that of y, etc. The following priority scheme seems to be reasonable for scheduling on a processing element processes of the three types introduced earlier:

- Suppose that C, D, and B represent active processes of the cyclic, deadline, and background types, respectively; then,

  $$P_r(C) > P_r(D) > P_r(B).$$

- For any two active cyclic processes, C_1 and C_2, with periods of lengths $\Delta t(C_1)$ and $\Delta t(C_2)$, respectively,

  $$P_r(C_1) \geq P_r(C_2) \text{ if } \Delta t(C_1) \leq \Delta t(C_2)$$

  with

  $$P_r(C_1) = P_r(C_2) \text{ only if } \Delta t(C_1) = \Delta t(C_2).$$

- For any two active deadline processes D_1 and D_2,

  $$P_r(D_1) \geq P_r(D_2) \text{ if}$$
  $$F[\Delta t_p(D_1), \Delta t_D(D_1)] \leq F[\Delta t_p(D_2), \Delta t_D(D_2)]$$

  with

  $$P_r(D_1) = P_r(D_2) \text{ only if}$$
  $$F[\Delta t_p(D_1), \Delta t_D(D_1)] = F[\Delta t_p(D_2), \Delta t_D(D_2)],$$

  where: (1) F is a real-valued priority function that is monotonically nondecreasing in each of its arguments (e.g., $F(r,x) = ar + bx$ where a and b are non-negative constants; if $a > 0$

and $b > 0$, then $F(r,x)$ is strictly increasing both in r and in x]; (ii) $\Delta t_P(x)$ is the processor time needed to complete the current execution of process x; (iii) $\Delta t_D(x)$ is the time remaining until the current deadline of process x.

Since at most one active B-process is allowed at a time in a processing element, no resolution of priorities among the B-processes assigned to the same processing element is needed. Priority ties among deadline processes can be resolved randomly, i.e., arbitrarily. However, it is best to specify a priori how to resolve the priority ties among a set of cyclic processes $C_1, C_2, ..., C_k$ with periods of identical lengths, for it affects the order in which these processes will be executed.

4. Overall Synchronization of Processes

Suppose that the distributed system under consideration contains n_{PE} processing elements (PEs). For reasons of simplicity, we have assumed decentralized executive control over these n_{PE} PEs. Unless appropriate measures are taken, processes executed not only in different PEs but even within the same PE may remain unsynchronized. Actually, two (although related) kinds of synchronization are implied here: synchronization of actions among processes, which can be attained by means of interrupts and interprocess data communications, and synchronization of processing with the outside world. The latter type of synchronization is usually accomplished through the monitoring of progression of time, of incoming measurements, or of other signals received from outside.

One technique for obtaining synchronization is to introduce systemwide timer interrupts, driven by an oscillator, for dividing the computational time line of each PE into time intervals of length Δt. In Section III.D, such intervals were called fundmental time frames (FTFs); the timer interrupts separating FTFs - FTF interrupts. The length of an FTF depends on the estimation problem and can be chosen to represent the fastest working rhythm in the system or in its measurement acquisition process. For example, in the estimation problem of GPS navigation (outlined in Appendix C), FTFs were chosen to 20 milliseconds long, because 20 millisecond cycles represent the basic transmission rhythm of GPS satelites; the length of an estimation cycle varies, but typically is about one second long.

To enable the applications processes in a processing element (PE) to read the time (actually, to count the FTF interrupts), the local real-time executive of such a PE must furnish a noninterruptable function subprogram, each call to which returns to the caller the current count of completed FTF interrupts. Applications processes, especially the measurement acquisition process, can use this time information to time-tag their outputs.

The synchronization method based on FTF interrupts, introduced above, can be extended to synchronize the cyclic processes of the entire distributed system. This can be done by extending the scheduling scheme credited to Jordan [Jor] in the next-to-last paragraph of Subsection 1 of the present section. With the notation introduced in that subsection, this extension is as follows:

(1) Let the lengths of fundamental time frames (FTFs) be expressible as $\Delta t = \Delta T_0/2^{k'}$ for some fixed nonnegative integer k', with ΔT_0 now representing the period (cycle) lengths of potentially highest rate cyclic process in the entire system.

(2) Synchronize all processes processing the periods of length ΔT_0 with FTFs by requiring that the cycle boundaries of such processes be always aligned with FTF boundaries.

(3) Let the period of any cyclic process in the system be expressible as $\Delta T_k = \Delta T_0 2^k$ for some nonnegative integer k.

(4) If $k>0$, then synchronize any cyclic process with the periods of length ΔT_k with all cyclic processes with the periods of length ΔT_j, where $j<k$, such that the cycle boundaries of former are always aligned with those of the latter.

5. Executive Service Routines and Program/Process Status Tables

Each time when the processing element returns the execution control to its RT executive, a subsystem of this executive, called a scheduler, decides which process presently posted on one of the ready queues will be executed next. (A ready queue can be thought of as a list of all processes of the same type which are in the ready state.) To make this decision, the scheduler follows the selection logic implied by the priority scheme adopted previously. Control specialist's parts of real-time software interface with these process management facilities via a set of subprograms, which sometimes are called executive service routines. Each executive service routine is an uninterruptable procedure, the execution of which requires a negligible amount of processor time. It is invocable from application processes and operates on data stored in program/process status tables. Functionally, executive service routines can be divided into two types: those which define/redefine a process for a program listed in the program/process status tables (PPS); and those which change the state of a process.

By an earlier assumption about the processing environment, each program of applications part of real-time software realizes at most one process at a time. Thus, the program/process status tables may be visualized as a two-dimensional array, each row of which represents a program and each column of which describes an attribute of programs or of processes defined for these programs. The entries in a row of PPS tables characterize a program and specify the state and characteristics of the process defined for this program. It is not always required that a process be defined for a program. If the latter is true at some time, such a program may be visualized as being inactive at that time, which is not equivalent to a possibly inactive but defined process.

The function of an executive service routine which defines/redefines a process for a specified program is to enter the characteristics of the process to be defined in an appropriate row of the PPS tables. Initially, a newly defined/redefined process is always declared to be inactive.

Process definition/redefinition subprograms enable applications processes to change the nature of the process associated with a program by changing its type, execution rate (if the latter is applicable to process type), priority, or other attributes. For example, a program which at some given time realizes a cyclic process may at some later time redefined to realize a deadline process (provided that the program logic allows doing it); or it may at some later time be redefined as a cyclic process with a changed period and a changed priority.

One could consider an alternative approach in which process types are defined and fixed prior to real-time operations, say, perhaps at process construction or program load time. Such an approach inconveniences a control system specialist because, at the start of design, he often is not sure himself under what scheduling rules various algorithms should be executed. At least, the capability to define the process at cold start offers a control system specialist a design convenience. Furthermore, the capability to redefine in real-time the process for a program enables the real-time system to reconfigure its mode of processing after a partial failure of equipment or after a drastic change in operational environment.

Executive service routines which change the state of a defined process will not be described here in detail. It is only important to note that, for each process type, a procedure must be furnished for every state transition defined in the process state transition graph.

E. INTERPROCESS COMMUNICATIONS AND SYNCHRONIZATION

1. Introduction

In order to perform common tasks, concurrent processes need to communicate through data. In the present section, we discuss the interprocess communication problem, as well as the related process synchronization problem, by taking a restrictive approach similar to our earlier handling of other aspects of real-time excutive design. For a more complete treatment of the subject, the interested reader should refer to a recent text on system programming or on operating systems, such as [Cof 73], [Fre], [Gra], [Hans], or [Mad].

A unit of data exchanged at a time is often called a message. In the applications considered here, a typical message is an array of homogeneous data (such as a Kalman gain vector/matrix), or a record of heterogeneous data (such as a Kalman gain vector, plus an identification tag of the measurement to which the gain vector corresponds), or just a flag indicating occurrence of an event.

It is easier to understand the interprocess communication problem if, with each message type, one can associate an area in global data memory reserved for storing a single or several instances of that message. One often uses the terms "buffer" or "communication buffer" when referring to such a dedicated memory area. Typically, a message contains two types of data: communicated (or applications) data and communication protocol data. The function of protocol data is to control the accessing of the buffer by the processes.

Messages of a certain type implement one-way communication between two or more processes if each unit message is entirely produced (written) by a single writer process and may be consumed (read) by one or several reader processes. In such a case, a buffer for storing messages may contain the communicated data produced only by a single writer process. On the other hand, several processes may be involved in generation and exchange of protocol data.

2. Assumptions and Design Principles

Next, we introduce the following restrictive assumptions as interprocess communication design principles:

(1) One-way communications - only one-way communications are admissible, which implies that any buffer may contain applications data produced only by a single writer process.

(2) Restriction on the length of stay in a critical region - no process remains in a critical region longer than for an a priori prescribed maximum length of time, such as a few milliseconds. (Recall that a process is assumed to be noninterruptable during the time interval spanned between entering and leaving such a region.)

(3) Limited waiting for the reading of data - if a reader process during its execution reaches a point where it tries to retrieve interprocess communication data from a buffer but cannot do it, because the buffer is locked out by a writer process which presently is writing into that buffer or because the buffer contains no new data that the reader process has not yet read (consumed), the reader process (perhaps after waiting at most for some predetermined length of time) proceeds to process other tasks without this time having retrieved the data.

(4) Limited waiting for the writing of data - if a writer process during its execution reaches a point where it wants to write interprocess communication data into a buffer but cannot do it because a reader process is presently accessing that buffer or because the buffer is full and its contents are not supposed to be overwritten, the writer process (perhaps after waiting at most for some predetermined length of time) proceeds to process other tasks without this time having written the data.

Principle (3) implies that on occasions (when no fresh data can be obtained) old data, such as old Kalman gains, will be repeatedly reused. Similarly, (4) implies that sometimes the passing of produced data will be skipped. The important idea here is that the processing logic must be designed not to fall apart if the situation described in (3) or in (4) rises. Only the overall system performance is allowed to degrade somewhat.

Adherence to the above stated principles for design of interprocess communications eliminates the possibility of deadlocks, for then the four necessary and sufficient conditions for a deadlock (e.g., [Fre], p. 157) cannot be met.

3. Communication Mechanisms and Their Implementations

Next we turn to specific designs of communication mechanisms, examining the capability of each to attain mutual exclusion of communication processes. In the sequel we examine the following interprocess communication techniques:

- Time-separated communications under the control of a single flag.
- Communications via a multiple buffer with or without a critical region.
- Noninterruptable communications via a critical region under the protection of Dijkstra's P and V semaphores or, perhaps, under more general communication control primitives.

It was noted in the discussion of mutual exclusion (Subsection 2 of Section III.B) that implementation of communication control mechanisms, which we called communication primitives, requires special indivisible operations. An operation was said to be indivisible if its execution, including the accessing of memory during its execution, cannot be interrupted. An indivisible operation may be implemented on several different levels: it may be a single machine language instruction such as a test-and-set instruction, an uninterruptable sequence of machine language instructions resulting from the compilation of a single or several high-level language statements, a single subroutine, or a pair of subroutines and a program segment between such a pair.

As an aside, the following two uninterruptable test-and-set instructions are useful in construction of communication primitives. The first tests whether its operand flag (the contents of a memory location) is nonzero; if it is, then this instruction sets the flag to zero and skips the next instruction; else, it proceeds to the next instruction without having changed the value of the operand flag. The second instruction type complements the first in the following sense: it tests whether the operand flag is zero; if it is, then the instruction changes the value of the flag to 1 and skips the next instruction; if not, it goes to the next instruction.

4. Time-Separated Communications under the Control of a Flag

Use of a single flag to control one-way communications between a writer process W and a reader process R is the simplest of all three communication techniques presented here. It attains mutual exclusion of processes because of the restrictions which it imposes on the participating processes.

To define one possible implementation of this communication method, let BUFFER be the name of the communication buffer and CFLAG be the name of the variable representing the control flag. The write procedure, executed by the writer process, is as follows:

```
write:  begin
        if CFLAG = 0 then goto alpha;
           else goto beta;
alpha:  write BUFFER;
        CFLAG := 1;
        end
beta:   (the next executable statement)
```

With the same variable names, the read procedure, executed by a reader process is as follows:

```
read:   begin
        if CFLAG = 1 then goto alpha;
           else goto beta;
alpha:  read from BUFFER;
        CFLAG := 0;
        end
beta:   (the next executable statement)
```

Despite their structural simplicity, the above write and read procedures should to be used only with caution, for they may prevent new data from being entered into the communication buffer until its previous contents have been read. On the other hand, this simple technique enforces mutual exclusion of communicating processes without assuming anything about their relative speeds. It is useful when the writer and reader processes are cyclically executed at the same average rate, because information will rarely be lost then. In such a case, it is convenient, although not absolutely necessary, to have the writer process precede the reader process in each cycle.

5. Communications Via Multiple Buffers

A multiple buffer of multiplicity M contains M data areas (each called a buffer) of identical structure and size. Such buffers are usually circularly arranged in the sense that the writer process,

after writing into the M_{th} buffer, next writes into the first buffer; similarly, the reader process, after reading from the M_{th} buffer, switches to the first. Hence, such a storage scheme is often called a circular buffer. In presence of fluctuations in execution rates of two communicating processes, communications via a multiple buffer, due to extra storage capacity, are less likely to loose information than communications based on the method described in the preceding subsection. Mutual exclusion is now attained by not allowing the reader process and the writer process to simultaneously access the same buffer. However, unless the communicating concurrent processes mutually exclude one another through scheduling control, safe implementation of multiple buffers requires the use of protected critical regions.

6. Uninterruptable Communications Via Protected Critical Regions

If one cannot or is not willing to make any assumptions about the relative speeds or execution times of communicating processes, then the following technique constitutes a general approach for controlling the access to a global data set by two or several concurrent processes, which always works (although we will later qualify "always"):

- With each global data set D, accessed by at least two processes, associate an integer valued access control flag S.
- Allow each process, P_I, communicating through D, to access D only from within critical regions, $CR_I(S)$, each protected (enclosed) by a pair of communication (synchronization) primitives operating on S. Let these two synchronization primitives, one at the entry point to a critical region and the other at its exit point, be integral parts of the critical region.
- Make the process which enters a critical region uninterruptable from the moment it starts to execute the first instruction of the entry point primitive until it completes the execution of the last instruction in the companion exit-point primitive.

Thus, a pair of corresponding communication primitives is a mechanism operating on S, which, after having been started, is executed to completion without interruption and which can be executed by only a single process at a time. Flag S is an integer that may be operated on only by a synchronization primitive or by a special procedure (the latter could be part of compilation or part of cold-start initialization) designed to initialize S.

It can be shown that the following primitives, derived from Dijkstra's P and V synchronization semaphores, handle all mutual exclusion and synchronization problems encountered in multiprogrammed multiprocessor systems of the type considered here (following the notation given on p. 129 of [Fre], all operations enclosed below within a pair of brackets are assumed to be uninterruptable):

```
WAIT(S) :  [S := S - 1; if S<0 then
           place the process which called WAIT on a wait queue,
           Q_S, and release the processor to another process; else
           enter the critical region.]

SIGNAL(S) : [S := S + 1; if S<0 then remove a process from Q_S and
           change its state to "ready" for processor allocation.]
```

For such a pair of synchronization primitives, one can define an initialization function, INIT(x,v), which initializes semaphore x to value v. Thus, if S is initialized to 1 by executing INIT(S,1), the write process W and the read process R can use the following procedures to communicate via the data written into (read from) the data set D. Each time that process W wants to write into BUFFER, it executes a code segment of the form:

```
begin
   WAIT(S);                 }
   write into D;            } A critical region
   SIGNAL(S);               } for process W
end
```

Similarly, each time when process R wants to read from data set D, it must execute a code segment, such as:

```
begin
   WAIT (S);                } A critical region
   read from D;             } for process R
   SIGNAL(S).
end
```

It is assumed that the global variable S is known as a semaphore to both of the above code segments. For a discussion of other (and more general) uses of the synchronization primitives WAIT and SIGNAL, as well as for definitions of other synchronization primitives, the interested reader is referred to Chapter 4 of [Fre].

In a foregoing paragraph, we asserted that the above described communications control mechanism is sufficient for the processing environment assumed here. The key to that is the assumptions (design principles) stated in subsection 2 of this section, especially the second assumption, according to which a process is not supposed to remain in a critical region longer than some a priori defined length of time. Adherence to this principle prevents deadlocks. This principle must be enforced at design time (at execution time a process is uninterruptable when it is inside a critical region and so cannot be forced out of it) by exercising care about the executable code which is placed inside critical

regions. In applications considered here, only the code segments requiring a limited amount of processor time and needed for writing data into a communication buffer (or for copying it from such a buffer) are allowed within a critical region.

Finally, we want to say a few words about the implementation of communication (or synchronization) primitives. First, the form in which they are implemented depends on the programming language used. If assembler language or a high-level language such as Fortran is used, these primitives should be implemented as an integral part of the real-time executive service facilities in the form of uninterruptable subprograms. If a programming language such as ADA [Weg], which is designed for multiprogrammed task execution, is used, then these or similar primitives are furnished as facilities built into the programming language.

IV. ALGORITHMIC AND PROCEDURAL ISSUES IN DESIGN OF REAL-TIME ESTIMATORS

A. INTRODUCTION

In the first part of present exposition, we examined computer implementation aspects of real-time control system design. In the course of doing it, we described software architecture for the real-time estimators to be implemented on a distributed system of small computers. The ideas on implemental design presented up to this point applied not only to realtime estimators but, more generally, to a variety of real-time control or communication systems.

In the second part, i.e., in Sections IV and V, we narrow down our focus to real-time estimators. We do it in two steps. The present section reviews selected issues pertaining to the design of computational algorithms and procedural logic for Kalman filters, although nearly all ideas will also apply to other types of recursive real-time estimators. Finally, Section V will illustrate filter mechanizations resulting from several known workload partitioning schemes. Typically, we end up with schemes requiring multiprogrammed processing environment. Concepts and techniques discussed in Sections II and III can then be applied to complete the implemental design of estimator software.

It is difficult to be objective and sufficiently broad in selection of algorithm and procedure design issues: what is important to one designer often is determined by his background and interests, and may appear to be insignificant to another. In our selection, we were guided by what we viewed as being critical to the real-time estimators of the type considered here. These factors are (i) modeling of the estimation problem, (ii) computational algorithms for implementing the covariance/gain filtering portion of the estimator, (iii) system identification in real-time, (iv) increasing the robustness of the estimation process against the perturbations such as bursts of high amplitude noise due to environmental disturbances, sensor failures, or sudden and drastic changes in the system model.

It is difficult to exaggerate the importance of modeling. But modeling depends on a particular problem. Hence, we shall not discuss it here. The purpose of the present section is to remind the reader about and to comment on the other issues identified above. However, since they are well covered in recent literature and really do not belong to our main theme, we shall not discuss any of them in detail. Rather we shall refer the interested reader to recent literature on topics related to these issues.

B. FILTER ALGORITHMS

In a Kalman filter, filter algorithms are the computational algorithms which perform covariance and gain processing and, after having been programmed, implement a critical kernel of real-time estimation software. They are critical mainly for two reasons: (i) they may potentially destabilize the estimation process or prevent it from converging and (ii) they may require an excessive amount of processing time and memory.

Covariance and gain (C/G) processing (filtering) algorithms perform the following functions: time propagation of state error covariance matrix P, computation of Kalman gains, and measurement updating of P. The current practice is to structure an estimation scheme so that the measurements in an estimation cycle are processed sequentially one by one and the Kalman gains are computed and state error covariances updated for each measurement separately. Such a scheme is called sequential processing of measurements.

Appendix A summarizes the original form of C/G processing algorithms for a linear Kalman filter with discrete measurements. Unfortunately, the measurement-update operation in the original form of Kalman filter algorithms [Equation (7) in Appendix A] is potentially unstable. Roundoff errors may eventually make the state error covariance matrix acquire negative characteristic roots and, thus, lose its positive definiteness. Hence, the criticality of numerical stability requirement in applications considered here motivates the use of square root filtering algorithms for covariance/gain processing.

Several variations of square root filtering are known. The version which has been defined and refined largely by Bierman is summarized in Appendix B. (References [Bie] and [Tho] describe to what we shall refer as Bierman's method and give timing and sizing models for it. The first of these references also discusses applications of square-root filtering techniques to information matrix estimation.) There is some controversy in literature about which particular form of square-root filtering should be used. Carlson [Car] describes what could be viewed as an alternative to Bierman's method, which is also attractive. Our selection of Bierman's method has been motivated mainly by years of satisfactory experience with it in applications to navigation problems. In any case, saving just a few percent of processing time should not be the decisive criterion for using one set of algorithms instead of another.

For a "neutral" overview of available options in square root filtering, the interested reader is referred to Chapter 7 of [May], which also summarizes comparative timing data for better known variants.

One benefit derived from the use of numerically stable covariance and gain processing algorithms is the feasibility to implement them and to make them perform in single (or reduced) precision floating-point arithmetic except for computations of some dot products. In a microprocessor, this often saves not only memory but also processing time because of the relative disparity in the speeds of single and double-precision floating-point operations. This disparity in processing speeds becomes especially large if the microprocessor does not have floating point arithmetic implemented in hardware form.

C. SYSTEM IDENTIFICATION

1. Identification Problems in Real-Time

We use the term "system identification" in a restricted sense to indicate acquisition of knowledge about the distributional properties of the stochastic processes representing the process and measurement noise sources in a Kalman filter.

In practice, one usually assumes that each noise source is represented by a stochastic process from some particular class of processes. In such a case, the identification problem reduces to determination of the parameters which define a particular process in the assumed class. Two distributional parameters that are usually of interest are the mean and the covariances of the stochastic process. These quantities may not be time invariant and so their values may have to be updated repeatedly. As the system model of a Kalman filter in Appendix A indicates, each noise source is typically modeled as a white, zero-mean gaussian process with unknown variances or covariances.

Should there be any suspicion that the process representing a noise source has a nonzero mean of unknown, but significant value, the unknown mean should be included in the system model as a state variable and estimated. Most often, the unknown parameter whose value is sought is a noise variance (for a scalar-valued process) or a noise covariance matrix (for a vector-valued process).

In cases of sequentially correlated noise, one also must effectively estimate autocorrelation coefficients in order to whiten the noise.

2. Identification Methods for a Kalman Filter

In the presence of colored noise (as is pointed out in Chapter 11 of [And]), retention of optimality properties of the filter is usually possible, although at the expense of increased complexity. This reference illustrates a few special cases (such as a situation in which the measurement noise process is Markov) and techniques for handling them which save the optimality properties of filter without increasing the dimensions of the state vector. Another approach is to replace the filter with one that is less complex by means of model order reduction. References [And] and [May] probably are the best introduction to the subject.

Methods for identifying the unknown covariances of noise processes can be roughly divided into: (1) adaptive (i.e., estimation time) methods, (2) heuristic on-line methods, and (3) a priori modeling methods. Occasionally, several methods are combined.

3. Adaptive Identification Methods

In Kalman filtering, the term "adaptive estimation" usually refers to on-line estimation techniques which include estimation of unknown distributional parameters in noise models. During the past decade, many adaptive estimation schemes were investigated and the results of this research reported in literature, e.g., refer to Brewer [Bre] or to Ohap and Stubberud [Oha].

Unfortunately, these on-line system identification techniques nontrivially increase the processing load in almost all cases and so may become prohibitively expensive with respect to processing time. For example, any measurement bias (i.e., nonzero mean of measurement noise) in principle can be estimated by modeling it as a component of the system state vector; but the computational load due to the processing Kalman gains and covariances in a filter is roughly proportional to Kn^3 operations, where n is the length of state vector and K is a scaling factor which depends on a particular algorithm used.

The adaptive techniques become even more computationally expensive when the noise random process under consideration is nonstationary or sequentially correlated. Then it is not enough to estimate the unknown distributional parameters once, say at the start of the estimation process, and then to continue using the obtained parameter estimates throughout the remaining part of the estimation process; but there is a need then to continue the estimation of changing distributional parameters throughout the estimation process. Furthermore, in some applications such as missile dynamics during powered flight, the noise characteristics may change so rapidly that even with almost unlimited processing resources it is impossible to input the measurements needed for system identification at a sufficiently high rate.

4. Heuristic On-Line and A Priori Modeling Methods

Current microprocessor and, in particular, special chip technologies, aided by modern methodology of software design and implementation, have made heuristic approach to system identification attractive. This approach utilizes the following two ideas.

First, with some planning, one can design the sensors and other measurement input ports (or at least the digital controllers of these devices) for a real-time estimation system so as to make them produce extra information in addition to the "regular" measurements specified in the system model. Usually, such extra information can be obtained at little additional cost as a byproduct of regular measurements. This extra information is often intended to help the estimation process (1) promptly

detect a change in the characteristics of a noise process or, more generally, in system state; (2) accurately approximate the values of process and measurement noise covariances.

Secondly, real-time test equipment (capable of creating a wide spectrum of possible operational environments and producing close-to-real-life measurements plus their extras) can be utilized to calibrate the noise parameters as a function of the received extra inputs for quick computation of covariances.

Next we illustrate applications of the heruristic modeling techniques outlined above to two estimation problems in GPS navigation. (Appendix C defines a simple version of the estimation problem for GPS navigation.) Several models of GPS navigation equipment have been or are being developed. A typical set of GPS user's equipment is built around a system of microprocessors and utilizes a specially designed receiver for obtaining pseudo-range and delta pseudo-range measurements at a high repetition rate. The receiver passes these measurements to a microprocessor based estimation system. The latter recursively produces a primary navigation solution (i.e., estimates the state vector) from which other navigation quantities of interest can be derived as byproducts.

EXAMPLE 1. One type of GPS navigation equipment was developed as part of test instrumentation for a long-range missile [Gyl 80]. Analysis of the process noise showed that all dynamics-related elements of process noise covariance matrix were expressible in terms of a single parameter, the acceleration variance. During a short powered flight, each of the three engines of the missile undergoes an acceleration peak and an acceleration valley; after the missile goes into the coasting flight, nearly all acceleration is due to gravitational attraction, which thereafter changes very slowly. In early design, several adaptive process noise covariance identification techniques were tried. They responded too slowly and were too expensive computationally. Thereafter, it was decided to instrument the system so as to provide the estimator with a discrete warning (completion) signal before (after) each event that drastically affected the acceleration (e.g., liftoff or a change in engine). Special real-time test equipment -- producing not only realistic GPS pseudo-range and delta pseudo-range measurements throughout a test mission but also the above described discrete event warning (completion) signals -- was used to select experimentally the best possible acceleration variance for each segment of a test mission.

EXAMPLE 2. This example deals with GPS navigation equipment for a medium-dynamics user [War]. The GPS receiver of this navigation equipment is designed to produce, in addition to GPS satellite pseudo-range and delta pseudo-range measurements, several parameters for computing the measurement noise variances; also the velocity variance from which all dynamics-related process noise covariances can be directly computed.

D. INCREASING THE ROBUSTNESS OF AN ESTIMATION PROCESS

Judging by a large number of current publications, much interest has been recently shown in robust statistical inference, including robust estimation (or, to be more specific, regression), the objectives of which are to handle the situations in which classical methods do poorly; for example, classical regression methods have difficulties with outlier and collinear data. What is nice about "off-line" statistical analysis is that, if one method of inference leads to suspicious results, the statistician can always try another one on the original data. In real-time estimation, however, we do not enjoy this luxury: data is processed at about its arrival rate; if the on-line analysis of data fails, it may be physically impossible or too expensive to repeat the experiment. This strongly motivates us to increase the robustness of an on-line estimation process. Proper preprocessing and screening of measurements contribute to it. Thus, while designing a real-time estimation scheme, one should always examine whether the considered application requires special procedures for (1) screening the measurements against isolated outliers, (2) detecting the leading and trailing edges of high-amplitude noise bursts, (3) detecting the onset of and then taking appropriate measures against nonwhiteness in measurement noise, and (4) censoring (imposing bounds on) measurements. One should also examine whether any special procedures are required for detecting the onset of a drastic change in the system model and for taking appropriate measures against detected changes.

One area which should be examined for each application is whether it is necessary to have procedures for monitoring the estimates and for altering them in case their values exceed a predetermined range. This simple heuristic technique, known to statisticians as censored estimation, has saved the situation in several known applications.

[Sch] is a highly readable reference which complements the discussion of algorithm and procedure design issues in the present section. Its discussion of the balancing of covariance matrices for filter convergence and stability is especially noteworthy.

V. DECOMPOSITION OF A KALMAN FILTER INTO CONCURRENT PROCESSES

A. MOTIVATION AND OVERVIEW

Sections II and III outlined software engineering techniques for decomposing a real-time control problem into concurrent processes. In applications considered here, such decomposition may enable an estimation scheme to satisfy the real-time constraints of the problem on a distributed system of small computers. Specified limitations on the weight, the volume, the power consumption, or cost of equipment often do not allow extending a distributed system through addition of extra processing elements. Thus, trying to satisfy the real-time constraints by incrementing the equipment is often unacceptable. In such a situation, decomposing the workload into concurrent processes is the only recourse. The present section illustrates this approach by means of several schemes for decomposing Kalman filters. Such decomposition of a Kalman filter (or of any recursive estimator) into concurrently executable procedures often constitutes part of what is known as *filter mechanization*, a term we mainly reserve for filter structuring.

In the sequel, we first introduce two basic structural formulations, direct and indirect, of a Kalman filter. Each can be used as a basis for decompositions presented subsequently. Next, in order to establish rationale and common reference for discussion of decompositions, we review the processing tasks comprising a single estimation cycle of a sequentially structured filter. Finally, we examine several schemes for decomposing the computations of such a filter into concurrent processes.

B. DIRECT/INDIRECT FORMULATIONS AND FEEDFORWARD/FEEDBACK MODES OF USE

1. Direct and Indirect Mechanizations

Two alternate approaches for formulating a Kalman filter are known. In the first, called direct (or total state) formulation, the state vector $\underline{s}$, which describes the total state of the system, is directly estimated; i.e., in each estimation cycle, $\underline{s}$ is first time-propagated and then measurement-updated. In the second approach, called indirect (or state error) formulation, a Kalman filter estimates not the system total state vector $\underline{s}$ but the error $\delta\underline{s}$ in $\underline{s}$. Thus, if an indirectly formulated filter is used, each estimation cycle involves three major steps: time-propagation of $\underline{s}$; estimation of error $\delta\underline{s}$ in $\underline{s}$; and updating of the propagated value of $\underline{s}$ by subtracting from it the estimate of $\delta\underline{s}$. For the estimation cycle with reference time t_k, the last step can be symbolically written as

$$\hat{\underline{s}}(k|k) = \hat{\underline{s}}(k|k-1) - \delta\hat{\underline{s}}(k|k).$$

Literature suggests that indirectly formulated Kalman filters were first introduced in navigation systems, although such a filter can be used in nearly any situation to which a directly formulated filter also applies.

2. Feedforward and Feedback Modes of Use

There are two basic modes, illustrated in Figure V.B-1, for using a Kalman filter (or any recursive estimator) in a control system: feedforward use and feedback use.

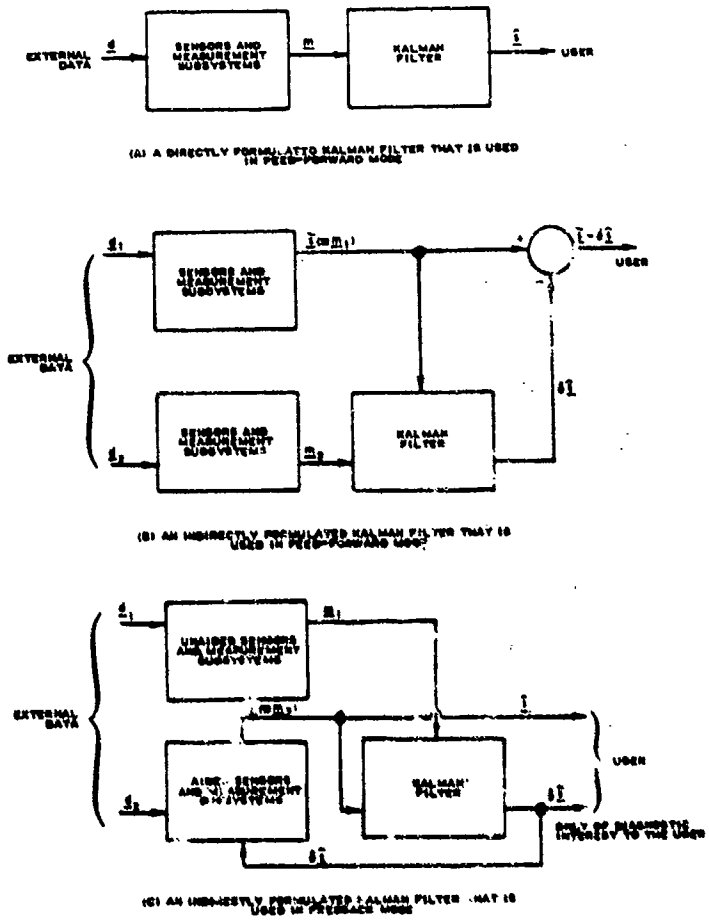


Figure V.B 1. Three Formulation and Use Mode Combinations of a Kalman Filter

A recursive estimator, used in feedback mode, feeds some of its estimates back into the control system from which it is receiving measurements. Feedback mechanization is widely used in integrated navigation systems, i.e., in navigation systems built around a recursive estimator which operates on measurements from several different types of sensors or measurement subsystems. Consider, for example, an integrated navigation system operating on two types of measurements: (1) position, velocity, and acceleration outputs of an inertial measurement subsystem; and (2) pseudo-range and delta pseudo-range measurements of GPS satellites. A loop which feeds back the estimation outputs into the inertial measurement subsystem could be used to recalibrate the inertial subsystem while the GPS measurements are being received and when they produce excellent navigation data. Another feedback loop could be

used for feeding aiding data to the GPS sensor (a special receiver) for quick reacquisition of GPS space vehicles (SVs). (Estimates of SV ranges and their time derivatives are the quantities which aid acquisition of a new SV or reacquisition of an SV whose track had been lost.) With feedback mechanization, it is often more convenient (although not mandatory) to use an indirectly mechanized Kalman filter. Part c of Figure V-B-1 graphically summarizes the above described filter mechanization, with the "AIDED SENSORS AND MEASUREMENT SUBSYSTEMS" box representing both the inertial measurement subsystem and the GPS receiver and with the "UNAIDED SENSORS AND MEASUREMENT SUBSYSTEMS" box standing for other possible (unidentified) measurement sources, say, such as an altimeter.

In the example of the preceding paragraph, if the user of equipment were interested only in accurate estimates of navigation quantities (such as position or velocity) and if he were indifferent to the calibration of the inertial measurement subsystem, he could have the filter mechanized in feedforward fashion, as illustrated in Part b of Figure V.B-1. In the latter case, the filter estimates errors in the navigational outputs of the inertial measurement subsystem (IMS), which, in Part b of Figure V.B-1, is the box outputting $\underline{m}_1$. Estimates of these errors are then subtracted from the navigational outputs of the IMS. Note that in the present case, (i) the outputs of the IMS are shown to be used by the Kalman filter for time-propagation of the navigation state vector $\underline{s}$ and (ii) the GPS receiver, the outputs of which are represented by $\underline{m}_2$, is not being aided (which is not a realistic assumption) by the Kalman filter. Similarly to the feedback mechanization, it is often more convenient (as indicated in Part b of Figure V.B-1) with the feedforward mechanization to use indirect formulation of a Kalman filter.

Care must be exercised in the design of a recursive estimator used in feedback fashion. A feedback loop may become unstable, which sometimes will manifest itself through sequentially correlated (nonwhite) measurement noise.

3. Pros and Cons of Indirect Formulation

In many applications of the type considered here (such as navigation systems), an indirectly formulated Kalman filter, designed to estimate low-frequency errors, can be executed at a considerably lower rate than the rate at which a direct filter would have to be run in order to perform comparably. This is because a linear model is often adequate for representation of low-frequency error dynamics. On the other hand, indirect formulation of a parallely structured filter usually costs more in processor time overhead than does direct formulation, because the error estimates in such a filter generally must be time-propagated in order to match their reference times with those of the state vector at each update of the latter.

C. REVIEW OF PROCESSING TASKS IN A SEQUENTIALLY STRUCTURED KALMAN FILTER

1. Sequentially Structured Filter

Before discussing parallely structured estimators, we briefly review the processing tasks comprising a single estimation cycle of a sequentially structured Kalman filter. We assume that such a filter processes measurements sequentially as scalars and that it is indirectly formulated. The standard Kalman filter equations summarized in Appendix A, then, suggest decomposition of work at time t_k into the following tasks:

(1) Time-propagation of state vector $\underline{s}$ from t_{k-1} to t_k; i.e., computation of $\hat{\underline{s}}(k)^- \triangleq \hat{\underline{s}}(k|k-1)$.

(2) Time-propagation of state error covariance matrix P; i.e., computation of $P(k)^- \triangleq P(k|k-1)$.

(3) Setting-up of the measurement processing loop, which (depending on the type of filter mechanization) may include activities such as the clearing of state error estimate vector $\delta\underline{s}$, the saving of state data, etc.

(4) Measurement processing loop, each pass through which processes scalar measurement $m_i(k)$ ($i = 1,2...n_m$) and requries execution of the following tasks:

  (4.1) Setting-up of indices and logic for the i^{th} pass through the loop.

  (4.2) Preprocessing of $m_i(k)$, which may include its conversion, transformations, and prefiltering.

  (4.3) Computation of predicted measurement, linear (or linearized) state-to-measurement transformation vector $\underline{h}(k)$, and measurement residual.

  (4.4) Screening of measurement $m_i(k)$ for acceptance/rejection by means of residual analysis.

  (4.5) Measurement-updating of state error covariance matrix P[such that the update $P(k)^+ = P(k|k)$ is completed in the last pass through the measurement loop] and computation of Kalman gain vector $K_i(k)$ for $m_i(k)$.

  (4.6) Updating of the estimate of state error $\delta\underline{s}(k)$ [such that the output of this step in the last pass through the measurement loop is $\delta\hat{\underline{s}}(k|k)$].

  (4.7) End-of-iteration processing.

(5) Measurement updating of state vector, i.e., computation of $\hat{\underline{s}}(k)^+ \triangleq \hat{\underline{s}}(k|k) = \hat{\underline{s}}(k|k-1) - \delta\hat{\underline{s}}(k,k)$.

In the sequel, the above tasks (4.1) through (4.4), (4.6), and (4.7) will be collectively referred to as measurement incorporation; the above tasks (2), (4.5), and modified forms of (4.1), (4.3), and (4.7), as covariance/gain filtering.

2. Departure from Sequentially Structured Filters

After decomposition of workload into the above identified or similar processing tasks and after obtaining timing and sizing estimates for each task, these tasks can be recombined into concurrently executable processes in many different ways. With timing and sizing data for each identified task available, it is easy to predict not only the processor time requirements for processes formed from these tasks but also the memory requirements for the programs and data sets implementing these processes.

This leads to candidate filter structures. The designer next faces the problem of selecting the most suitable structure (or set of structures) for the problem at hand. He may select a set of structures, each suitable to a particular mode or phase of his estimation problem, and then apply the process control facilities of the real-time executive outlined in Section III to have processes adaptively reconfigured (redefined) in real-time.

The price to be paid for solving the scheduling problem via structures of concurrent processes is poorer functional performance, because it leads to algorithms which usually are less "optimal" than their original, sequentially executable forms. Thus, one task which the designer now confronts is to determine by how much the performance actually suffers. This is normally done by means of simulations.

D. PARALLEL STRUCTURES

1. Introductory Remarks

Next, we illustrate several solutions to the real-time constraint problem through workload decomposition into concurrent processes. For this purpose, we outline several ways for parallel structuring of Kalman filters.

Figure V.D-1 illustrates how a slight restructuring of the procedure used in the sequential model of Section V.C changes a sequential scheme into a parallely structured estimator with two concurrent processes: one for propagation/update of state vector $\underline{s}$; the other for processing of measurements and covariance/gain computations. The outputs of the Kalman filter, $\hat{\underline{s}}(j|j)$, identified as user's estimate in Figure V.D-1, are not "strictly" Kalman in the sense that every second time they are computed while using a time-propagated value of $\delta\hat{\underline{s}}$. For an indirectly formulated filter, the second process estimates the error $\delta\underline{s}$ in state vector $\underline{s}$.

Further separation of measurement processing and covariance/gain computations by introduction of an additional concurrent process leads to a triply parallel filter structure consisting of three concurrently executable processes: one for state propagation/update; the second for measurement processing; the third for covariance/gain computations. The resulting scheme is illustrated in Figure V.D-2.

If, instead of decomposing the workload into two processes as in Figure V.D-1, we allowed one process to propagate the state vector, process the measurements, and update the state vector (with the Kalman gains computed by the other process), while assigning to the other the processing of covariances and the computation of Kalman gains, then we would obtain a doubly parallel estimation scheme with concurrent covariance/gain filtering, which is described in Subsection 4.

Next, we examine the above introduced three parallel filter structures in greater detail. We assume that only one processing element is available for filter functions. Hence, concurrent processes resulting from filter decompositions must be executed in interleaved time fashion on a single processor.

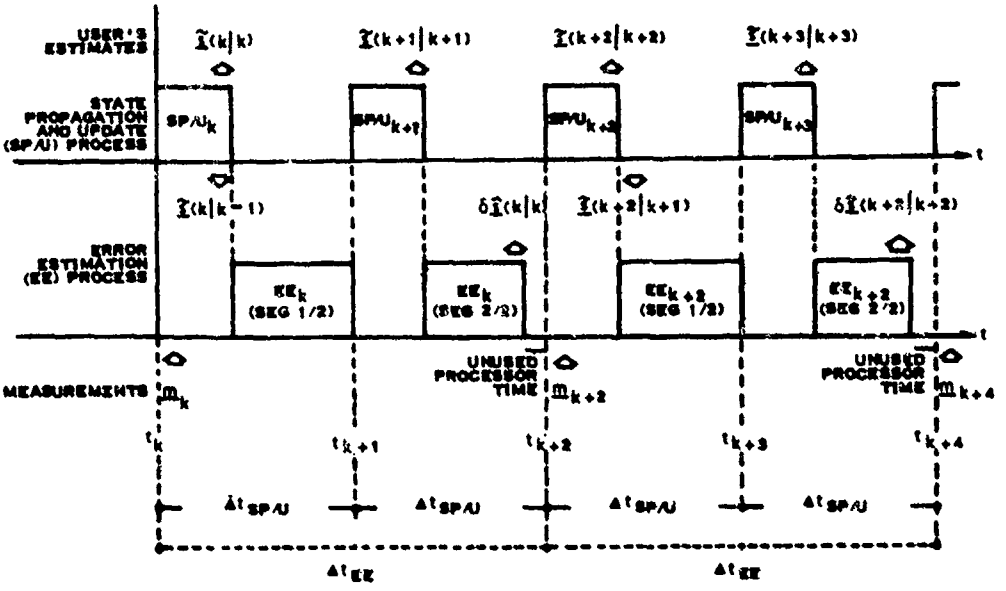
2. Parallel Estimation of State Error

This is the doubly parallel structure shown in Figure V.D-1. For this structure, we now assume indirect filter formulation. Hence, as mentioned previously, one process (called the error estimation process) estimates the error $\delta\underline{s}$ in state vector $\underline{s}$ (and actually performs all functions normally ascribed to a Kalman filter), while the other process (called the state propagation/updating process) propagates the state vector $\underline{s}$ and then updates $\underline{s}$ by subtracting from it the estimate of $\delta\underline{s}$ passed by the error estimation process.

Compared to the other two parallel filter structures discussed in the sequel, this scheme yields high-rate propagation/updating of the state vector. However, somewhat stale estimates of $\delta\underline{s}$ (although properly time-aligned by propagation) will in general be used for updating of $\underline{s}$.

This scheme may be the only recourse if it is required that the state vector $\underline{s}$ be propagated/updated at a rate much faster than that at which the whole filter can be executed. For example, this may be required in the navigation applications (where $\underline{s}$ represents the navigation solution) to computations such as aerial cargo drops or weapon deliveries. With this filter structure, it is helpful to aid the propagation of $\underline{s}$ with outputs from a measurement system (such as the velocity and acceleration inputs from an inertial subsystem), complementing the primary measurements on which the filter is operating.

This doubly parallel scheme reduces to a sequentially structured filter when the execution rates of the state P/U process and of the error estimation process are equal. In such a case (or when the rate of error estimation process is not significantly lower than that of state P/U process), this scheme, if properly handled, displays many good properties (such as robust initial convergence) of a Kalman filter.



REMARKS:

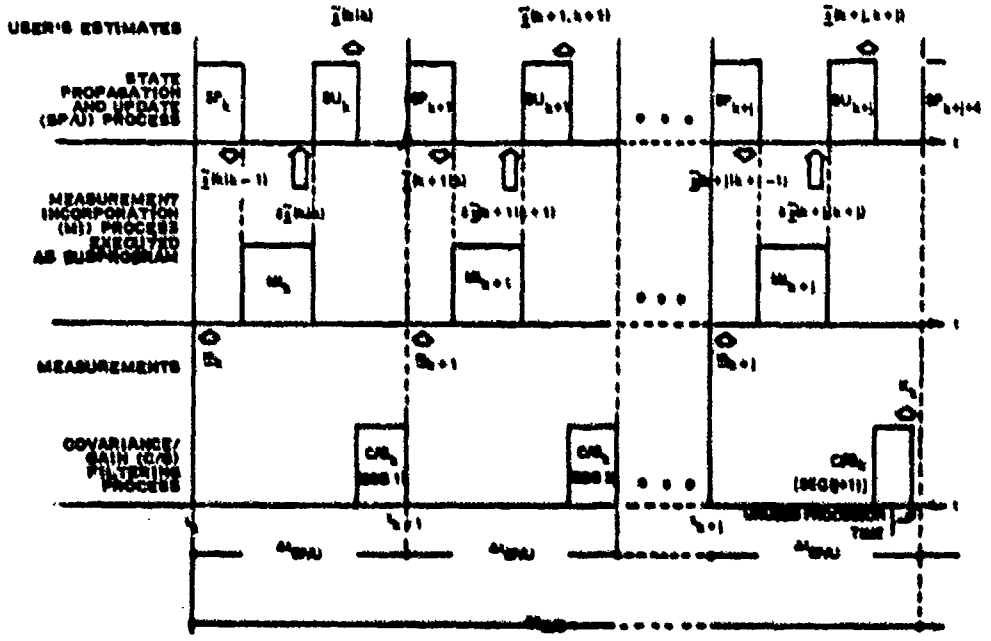
(1) THE ESTIMATES THAT ARE NOT STRICTLY "KALMAN" ARE CAPPED WITH AN "~" SIGN.

(2) SUBSCRIPTS k, $k+1$, ... REFER TO THE REFERENCE TIMES OF DATA OR OF PROCESS EXECUTIONS.

(3) $\Delta t_{SP/U}$ = CYCLE LENGTH OF STATE P/U PROCESS.

(4) Δt_{EE} = CYCLE LENGTH OF EE PROCESS.

Figure V.D-1. Scheduling of Doubly Parallel Filter with Concurrent Error Estimation for Special Case $\Delta t_{EE} = 2\,\Delta t_{SP/U}$



REMARKS:

(1) THE ESTIMATES THAT ARE NOT STRICTLY "KALMAN" ARE CAPPED WITH A "~" SIGN.

(2) SUBSCRIPTS k, $k+1$, ... REFER TO THE REFERENCE TIMES OF DATA.

(3) $\Delta t_{SP/U}$ = CYCLE LENGTH OF STATE P/U PROCESS

(4) $\Delta t_{C/G}$ = CYCLE LENGTH OF C/G FILTERING PROCESS

Figure V.D-2. Scheduling of Triply Parallel Filter with Concurrent Measurement Incorporation and Covariance/Gain Filtering

The parallel error estimation scheme easily lends itself to measurement screening and nicely responds to real-time changes in the system model. Its chief disadvantage is a relatively low rate of measurement incorporation, which may prohibit its use in dynamically lively applications. However, by introduction of a third concurrent process for covariance/gain filtering (and thus by separation of these functions from error estimation), the problem of low rate of measurement incorporation can be alleviated, but at the cost of an additional increase in dissimilarity from the structure of canonical Kalman filter algorithms. Such a triply parallel scheme is discussed in Subsection 3.

If the state propagation/update and error estimation processes are the only two processes assigned to a processing element, they can be implemented by defining the first process as a cyclic process and the second as a background process. The latter will be given all free processor time remaining after execution of the cyclic process. Should there be other background processes assigned to the same processing element, the following two options are available: (1) define the error estimation process as one of several background processes but allow only one such process to be active at a time; (2) define the error estimation process as a deadline process.

3. Triply Parallel Estimation: Covariance/Gain Filtering Performed Concurrently with Error Estimation

This estimation scheme (summarized in Figure V.D-2 and obtained as indicated in the preceding subsection) processes measurements at a higher rate than the doubly parallel scheme with concurrent error estimation, but does it at the cost of performing the covariance filtering and the gain computations at a much lower rate. It should be used when a high rate of measurement incoporation is more essential to performance than optimal gains.

It is not difficult to see that this filtering scheme is awkward in screening and rejecting measurements and, at best, only sluggishly responds to changes in system model and poorly converges after initialization. All these poor properties are due to low-rate, autonomous processing of gains and covariances.

To implement a triply parallel estimation scheme on a processing element, three processes need to be defined: (1) a cyclic state propagation/ update process, which can also be designed to serve as a logical controller of the entire Kalman filter; (2) a measurement incorporation process, which (depending on the measurement acquisition mechanism) may be implemented either as a cyclic process or an almost cyclically scheduled deadline process; and (3) a covariance/gain filtering process, defined as a background process, which will be given all free processor time remaining between repeated executions of the other two processes. However, should there be other background processes on the same processing element, then either one background process at most would be kept active at a time or the covariance/gain filtering process would have to be defined as a deadline process.

4. Doubly Parallel Estimation with Concurrent Covariance/Gain Filtering

This scheme is based on two processes. In each estimation cycle, the first (state estimation) process first propagates the state vector $\underline{x}$, then sequentially processes the current measurements while using the gains computed by the other process, and finally updates the state vector. Any of the two filter formulations (direct or indirect) can be used in state estimation. The other (covariance/gain filtering) process propagates the state error covariance matrix P, then enters a loop (structurally similar to the measurement processing loop in state error estimation process) to update P and compute gains.

This scheme is similar to the triply parallel estimation scheme introduced earlier except that the propagation/updating of state and the processing of measurements presently are done within the same process. Thus, (due to low-rate, autonomous computation of gains) this scheme has properties similar to those of triple estimation, the only main difference being that state propagation/updating and measurement incorporation are now performed within the same process. This limits the rate at which $\underline{x}$ can be propagated and updated.

This type of filter mechanization has been used successfully in GPS navigation applications described in [Dam] and [Gyl 80]. In these applications, the linearized state-to-measurement transformations change slowly after the initial convergence (which justified low rate of gain computation) and good performance depends on a high rate of measurement incorporation. The initial convergence and sudden changes in the system model or in measurements still require special care.

To implement this scheme, one can define the state estimation process as a cyclic process and the covariance/gain filtering process as a background process. The latter will be given all free processor time remaining between cyclic executions of the state estimation process. Should there be other background

VI. SUMMARY AND CONCLUSIONS

In this article, we addressed the control specialist who is facing the problem of having his estimation algorithms implemented as a working real-time system. We attached the name "implemental design" to the activities concerned with adaptation and restructuring of algorithms for computer implementation.

From the start, we assumed that distributed systems of microprocessors (or just of small computers) were the type of hardware on which implemental design was to be executed, but this did not rule out systems consisting of a single computer. With this assumption, we narrowed the discussion to a generic class of estimation/control real systems considered to be suited to implementation on distributed

microprocessors. We characterized them as small-to-medium-scale control systems designed to be boundedly loadable, i.e., to accept a processing load not exceeding the bounds established at design time. To this category belong practically all real-time control systems which are equipped with estimators and which are to be implemented on hardware of the type considered here.

Next, we introduced multiprogramming as a suitable processing environment and defined a structural model of software architecture for this environment. Such a model is needed for software timing and sizing, which was identified as an important task in implemental design of real-time software. It is also needed for design of a real-time operating system, called real-time executive.

We also introduced process as a fundamental concept from theory of operating systems and defined it to represent a program in execution, but not necessarily executing at the moment. To simplify the processing environment model, we also required preservation of one-to-one correspondence between programs and processes.

Our next topic was real-time executive. Selected issues of process management and resource allocation (such as deadlocks and their prevention, interprocess communications, and process synchronization) were reviewed. Thereafter, our attention turned to two issues in real-time executive design which very much affect a control specialist engaged in implemental design of estimation algorithms for real-time operations: process scheduling and interprocess communications.

At the start of the second part of the present exposition, we turned to the issues directly associated with estimators. In Section IV, we reviewed several algorithmic and procedural aspects of estimator design, which must be considered if the resulting real-time estimator is to be numerically stable, computationally efficient, and robust to disturbances in measurements. Discussed were stable algorithms for covariance/gain processing in a Kalman filter and real-time system identification techniques. In Section V, by means of illustrations, we looked into practical schemes for decomposing estimators of the Kalman filter type into structures of concurrent processes. Two basic filter mechanization schemes were compared, underlying such parallel structures, were compared. They are direct and indirect filter formulations. Also, two modes of filter use, feedforward and feedback, were introduced.

ACKNOWLEDGMENTS

The author wishes to express his gratitude to the staff of Texas Instruments Incorporated for the support received during the preparation of this article. He is particularly indebted to Phillip W. Ward for many years of opportunity to test in practice some of the ideas presented here, to Srini Raghavan for advice and criticism during preparation of this article, and to Beverly Littlejohn and Alice Dunbar for editing and typing expertise.

REFERENCES

[And] ANDERSON, B.D.O., and MOORE, J.B. (1979). Optimal Filtering. Prentice-Hall, Inc., Englewood Cliffs, N.J.

[Andr] ANDREWS, A. (1968). "A Square Root Formulation of the Kalman Covariance Equations." AIAAJ, Vol. 6, pp. 1165-1166.

[Bas] BASS, L.J. (1973). "On Optimal Processor Scheduling for Multiprogramming." SIAM J. Comput., Vol. 2, No. 4, pp. 273-280.

[Ber] BERG, R.O., and THURBER, K.J. (1972). "Real Time Task Scheduling for a Multicomputer System." Proc. of the National Electronics Conference, Vol. 27, pp. 275-280.

[Bie] BIERMAN, G.J. (1977). Factorization Methods for Discrete Sequential Estimation. Academic Press, New York, N.Y.

[Bre] BREWER, H.W. (1976). "Identification of the Noise Characteristics in a Kalman Filter." Control and Dynamic Systems, Vol. 12 (edited by C.T. Leondes). Academic Press, New York, N.Y.

[Car] CARLSON, N.A. (1973). "Fast Triangular Factorization of the Square Root Filter." AIAAJ, Vol. 11, No. 9, pp. 1259-1265.

[Cof 73] COFFMAN, Jr., E.G., and DENNING, P.J. (1973). Operating Systems Theory. Prentice-Hall, Inc., Englewood Cliffs, N.J.

[Cof 76] COFFMAN, Jr., E.G. (ed.) (1976). Computer and Job-Shop Scheduling Theory. John Wiley and Sons, Inc., New York, N.Y.

[Cox] COX, Jr., D.B. (1978). "Integration of GPS with Inertial Navigation Systems." Navigation, Vol. 25, No. 2, pp. 236-245.

[Dam] DAMOULAKIS, J.N., GYLYS, V., and UPADHYAY, T.N. (1978). "Recent Results in Navigation Systems Utilizing Signal Aiding from NAVSTAR Satellites." Record of IEEE 1978 Position Location and Navigation Symposium, pp. 388-395.

[Fre] FREEMAN, P. (1975). Software Systems Principles - A Survey. Science Research Associates, Inc., Chicago.

[Gra] GRAHAM, R.M. (1975). Principles of Systems Programming. John Wiley and Sons, Inc., New York, N.Y.

[Gel] GELB, A. (ed.) (1974). Applied Optimal Estimation. The MIT Press, Cambridge, MA.

[Gyl 76] GYLYS, V.B., and EDWARDS, J.A. (1976). "Optimal Partitioning of Workload for Distributed Systems." Digest of Papers, COMPCON 76, IEEE Computer Society, pp. 353-356.

[Gyl 80] GYLYS, V.B., and WARD, P.W. (1980). "Design and Performance of the Missile-Borne Receiver Set." Presented at NAECON 1979. Reprinted in Texas Instruments Equipment Group Engineering Journal, Vol. 3, No. 3.

[Han] HANSEN, P.B. (1973). Operating System Principles. Prentice-Hall, Inc., Englewood Cliffs, N.J.

[Jaz] JAZWINSKI, A.H. (1970). Stochastic Processes and Filtering Theory. Academic Press, N.Y.

[Jen] JENSEN, E.D., and BOEBERT, W.E. (1976). "Partitioning and Assignment of Distributed Processing Software." Digest of Papers, COMPCON 76,

IEEE Computer Society, pp. 348-352.

[Jor] JORDAN, J.W. (1970). "Task Scheduling for a Real Time Multiprocessor." NASA TN-D-5786.

[Liu] LIU, C.L., and LAYLAND, J.W. (1973). "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment." Journal of the Assoc. for Computing Machinery, Vol. 20, No. 1, pp. 46-61.

[Mad] MADNICK, S.E., and DONOVAN, J.J. (1974). Operating Systems. McGraw-Hill Book Company, New York, N.Y.

[Man] MANACHER, G.K. (1967). "Production and Stabilization of Real-Time Task Schedules." Journal of the Assoc. for Computing Machinery, Vol. 14, No. 3, pp. 439-465.

[May] MAYBECK, P.S. (1979). Stochastic Models, Estimation, and Control, Vol. 1. Academic Press, New York, N.Y.

[Oha] OHAP, R.F., and STUBBERUD, A.R. (1976). "Adaptive Minimum Variance Estimation in Discrete

Time Linear Systems." Control and Dynamic Systems, Vol. 12 (edited by C.T. Leondes). Academic Press, New York, N.Y.

[Sch] SCHMIDT, G.T. (1976). "Linear and Nonlinear Filtering Techniques." Control and Dynamic Systems, Vol. 12 (edited by C.T. Leondes). Academic Press, New York, N.Y.

[Smy] SMYTH, R.K. (1980). "Avionics and Controls in Review." Astronautics and Aeronautics, Vol. 18, No. 4, pp. 40-52.

[Tho] THORNTON, C.L., and BIERMAN, G.J. (1980). "UDU^T Covariance Factorization for Kalman Filtering." Control and Dynamic Systems, Vol. 16 (edited by C.T. Leondes). Academic Press, New York, N.Y.

[Upa] UPADHYAY, T.N., and DAMOULAKIS, J.N. (1980). "Sequential Piecewise Recursive Filter for GPS Low-Dynamics Navigation." IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-16, No. 4, pp. 481-491.

[Van] VAN DIERENDONCK, A.J., RUSSELL, S.S., KOPITZKE, E.R., and BRINBAUM, M. (1978). "The GPS Navigation Message." Navigation (J. of the Inst. of Navig.), Vol. 25, No. 2, pp. 147-165.

[War] WARD, P.W. (1982). "An Advanced NAVSTAR GPS Geodetic Receiver." Proc. of 3rd International Geodetic Symposium on Satellite Doppler Position.

[Weg] WEGNER, P.W. (1980). Programming with ADA - An Introduction by Means of Graduated Examples. Prentice-Hall, Inc., Englewood Cliffs, N.J.

APPENDIX A: KALMAN FILTER ALGORITHM FOR A DISCRETE LINEAR SYSTEM WITH SAMPLED MEASUREMENTS

In this appendix we summarize for reference purposes the standard form of Kalman filter algorithm, including the system model which this algorithm assumes, for a discrete linear system with sampled measurements.

A. SYSTEM MODEL

1. Propagation of the system state vector from $t = t_{k-1}$ to $t = t_k$:

$$\underline{x}(k) = \Phi(k, k-1)\underline{x}(k-1) + G(k-1)\underline{w}(k-1). \tag{1}$$

2. Measurements at $t = t_k$:

$$\underline{z}(k) = H(k)\underline{x}(k) + \underline{v}(k). \tag{2}$$

3. Initial conditions at $t = t_0$:

$$E[\underline{x}(0)] = \hat{\underline{x}}(0), \quad Cov[\underline{x}(0) - \hat{\underline{x}}(0)] = P(0). \tag{3}$$

4. Assumptions about system statistics:

   a. The processes $\{\underline{w}(k)\}$ and $\{\underline{u}(k)\}$ are zero mean, mutually independent Gaussian processes with covariances

$$E[\underline{w}(k)\underline{w}(j)^T] = Q(k)\,\delta_{kj}$$

   and

$$E[\underline{u}(k)\underline{u}(j)^T] = R(k)\,\delta_{kj}.$$

   b. Furthermore, $\underline{s}(0)$ is independent of $\underline{w}(k)$ and $\underline{u}(k)$ for any k.

B. ESTIMATION PROCEDURE

1. Propagation of estimates from $t = t_{k-1}$ to $t = t_k$:

$$\hat{\underline{s}}(k)^- = F(k,k-1)\hat{\underline{s}}(k-1)^+, \tag{4}$$

$$P(k)^- = F(k,k-1)P(k-1)^+F(k,k-1)^T + G(k-1)Q(k-1)G(k-1)^T. \tag{5}$$

2. Updating of estimates at $t = t_k$:

$$K(k) = P(k)^-H(k)^T[H(k)P(k)^-H(k)^T + R(k)]^{-1}, \tag{6}$$

$$P(k)^+ = [I - K(k)H(k)]P(k)^-, \tag{7}$$

$$\hat{\underline{s}}(k)^+ = \hat{\underline{s}}(k)^- + K(k)[\underline{m}(k) - H(k)\hat{\underline{s}}(k)^-]. \tag{8}$$

C. EXTENSIONS TO NONLINEAR MEASUREMENT EQUATIONS

The measurement equations (2) are replaced with $\underline{m}(k) = \underline{h}[\underline{s}(k), t_k] + \underline{u}(k)$. (9)

The linearized state-to-measurement transformation H appearing in filter equations (6) through (8) are now obtained by means of vector differentiation of $\underline{h}[\underline{s},t]$ with respect to $\underline{s}$:

$$H(k) = (\delta\underline{h}[\underline{s}(r),r]/\delta\underline{s})_{r=t_k}. \tag{10}$$

D. NOTATION USED ABOVE

- Upper-case letters represent matrices
- Lower-case letters represent scalars or (if marked with bars underneath) column vectors.
- $\underline{x}(k)$ represents the value of column vector $\underline{x}$ at $t = t_k$; similar notation is used to time-tag scalars and matrices.
- A^T represents the transpose of matrix A; if $\underline{x}$ is a column vector, then $\underline{x}^T$ represents the transpose of $\underline{x}$, which is a row vector.
- $\hat{\underline{s}}(k|i)$ denotes an estimate of $\underline{s}$ at time t_k that has been obtained by using a history of measurements up to and including the time $t_i (\le t_k)$.

APPENDIX B: U-D FACTOR COVARIANCE/GAIN PROCESSING ALGORITHMS FOR KALMAN FILTERS

A. INTRODUCTION

Covariance and gain processing algorithms, operating on U and D factors of state error covariance matrix P, are a technique for implementing "square root filtering" without requiring computation of square roots. These algorithms offer a recommended approach for overcoming the numerical instability inherent in the original formulation of Kalman filter algorithms. The latter is summarized in Appendix A. Numerical instability problems are caused by the repeated use of formula (7) in Appendix A for measurement updating of state error covariance matrix P. Examination of that formula shows that accumulation of roundoff errors may eventually cause matrix P to lose its positive definiteness. It should also be mentioned at this point that the U-D covariance factorization algorithms are just a numerically stable and computationally efficient method for implementing Kalman's estimation procedure (outlined in Appendix A), but not an estimator different from Kalman filter.

The term "U-D covariance factorization" comes from a property of nonnegative definite symmetric matrices, according to which positive semidefinite matrix P can be factored into $P = UDU^T$, where U is an upper triangular matrix with unit elements on its main diagonal and D is a diagonal matrix.

For convenience, we next summarize the basic U-D coverience factorization algorithms. For details, the reader is referred to [Bie] and [Tho], the second being an updated review of the topic, although narrower in scope than the first.

B. NOTATION

Suppose that the reference time of the current estimation cycle is t_k. To simplify the notation used in Appendix A (and to make it more compatible with the notation used in [Tho]), we drop explicit referencing to time and write $\bar{\underline{s}}$ for $\hat{\underline{s}}(k|k-1)$, $\hat{\underline{s}}$ for $\hat{\underline{s}}(k|k)$; Similarly, $\bar{P}$ for $P(k|k-1)$ and $\hat{P}$ for $P(k|k)$. Otherwise, we use the same letters as in Appendix A to denote the quantities in terms in which the system model and the Kalman filter algorithms are formulated.

In specification of computational algorithms, we use (in agreement with the current practice of algorithm definition) ":=" instead of "=" to connect the left- and right-hand sides of assignment statements. We do it to emphasize the idea of value replacement.

The following symbols are used to specify the size of system model: "n" to denote the length of state vector $\underline{s}$; "n_q", the length of process noise vector $\underline{w}$ ("q" here relates to process noise covariance matrix Q); and "n_m", the length of measurement vector $\underline{m}$, although, in the sequel, we consider only the sequential processing of scalar measurements.

C. U-D FACTOR MEASUREMENT-UPDATE ALGORITHM

Suppose that the following input quantities are given:

- H = 1xn matrix representing the linearized state-to-measurement transformation for the scalar measurement to be processed;
- R = noise variance ($\triangleq \alpha_0$) of the measurement to be processed;
- $\tilde{U},\tilde{D}$ = U- and D-factors of P (time-propagated to t_k or measurement-updated at t_k for all measurements so far processed at t_k).

Proceed as specified by the algorithm contained in Figure Ap.B-1 in order to perform covariance/gain update processing for a scalar measurement. To process a vector of n_m measurements, this algorithm would have to be embedded in a measurement processing loop (e.g., Section V.C) and then executed iteratively n_m times.

This algorithm uses a scalar λ and n-dimensional vector $\underline{f}$ and $\underline{g}$ as intermediate variables.

The outputs are:

- $\underline{k}$ = n-dimensional Kalman gain vector or (following the notation of Appendix A) nx1 matrix K;
- $\hat{U},\hat{D}$ = U- and D-factors of P (measurement-updated for all measurements so far processed at t_k.

Instead of outputting $\underline{k}$, it is often preferable to output separately

- $\underline{\nu}$ = n-dimensional normalized Kalman gain vector and
- α = the innovations variance, which, following the notation of Appendix A, can be expressed as $[H(k)P(k)^{-}H(k)^T + R(k)]$.

This is because α is used in the measurement processing loop for screening a processed measurement by testing whether its residual lies within an acceptance interval of the form $(-b\alpha_n, b\alpha_n)$, where b(>0) is a scaling parameter.

```
UDMUPD: begin
        f^T:=HU                          (where   f^T=[f_1,...,f_n])
        g:=D~f;                          (where   g^T=[g_1,...,g_n])
        for j=1,...,n do;
           α_j := α_{j-1} + f_j*g_j;     (where   α_0 = R and  α=α_n)
           if α_j = 0, then D^_j := D~_j;
              else D^_j := (α_{j-1}/α_j)D~_j;
           ν_j:=g_j;
           if j = 1, then go to ELj;
              if α_{j-1} = 0, then λ := 0;
                 else λ := -f_j/α_{j-1};

              for i = 1,...,j-1 do:
                 U^_ij := U~_ij + ν_i λ;
                 ν_i   := ν_i + U~_ij ν_j;
              end                        (Recycle if i<j-1)
ELj:    end                              (Recycle if j<n)
        k := ν(1/α_n);                   (k=[k_1,...,k_n])
        end UDMUPD
```

Figure Ap.B-1. Executable (computational) part of U-D Measurement-Update Algorithm, defined in [Tho], pp. 198-199. Inputs, outputs, and intermediate quantities are specified in Section C.

D. COMPARISON WITH THE ORIGINAL FORM OF KALMAN FILTER

Next, we restate the covariance/gain filtering part of the original form of Kalman Filter [defined in Appendix A by equations (5) through (7)] for processing a single scalar measurement in terms of the notation introduced in the present appendix. We do it in order to facilitate comparison between the original algorithms and the U-D factor covariance/gain processing algorithms summarized in the present appendix.

In terms of the simplified notation used in the present appendix, equations (5) through (7) of Appendix A yield the following procedure:

```
begin
P̄ := FP̂F^T+GQG^T;          (Covariance propagation to t_k)
ν := PH^T;                  (Normalized Kalman gain)
α := Hν+R;                  (Innovations covariance)
k := να^-1;                 (Kalman gains at t_k)
P̂ := P - kν^T;              (Covariance measurement-update at t_k)
end
```

The above form of covariance measurement update [or equivalently formula (7) in Appendix A) is known to be computationally unstable in the sense that it may make P acquire negative characteristic roots as a result of roundoff errors and overconvergence of P. A stabler, and although computationally more expensive, version of that formula for a scalar measurement (with H being 1xn matrix) is

$$P := (I - kH)P(I - kH)^T + kRk^T.$$

Its vector measurement version is

$$P := (I - KH)P(I - KH)^T + KRK^T$$

where K now is an nxn_m matrix and H is an $n_m xn$ matrix. Due to the amount of processing required, this stabler form is rarely used in real-time applications.

E. TIME-UPDATING PROPAGATION OF STATE ERROR COVARIANCES

When the U-D factorization algorithms are used for measurement updating of state error covariance matrix P and computation of Kalman gains, the following two alternative approaches for performing the time-propagation of P are available: the Conventional Propagation-of-Covariance Algorithm and the U-D Factor Time-Update Algorithm. In the sequel, these two approaches are outlined for propagation of P from time t_k to $t_{k+1} = t_k + \Delta t_k$, i.e., for computing $P(k+1)^- \triangleq P(k+1|k) \triangleq \bar{P}$ from the measurement-update of P from the preceding estimation cycle.

F. THE CONVENTIONAL PROPAGATION-OF-COVARIANCE ALGORITHM

This algorithm for computing $\bar{P}$, given $\hat{P}$ from the preceding cycle, in terms of factor matrices $\hat{U}$ and D is based on the following procedure:

a. Compute

$$\bar{P} := (F\hat{U})\hat{D}(F\hat{U})^T, \tag{1}$$

which yields the canonical product representation of matrix $\bar{P}$; here, $F=F(k+1|k)$ is the state transition matrix for propagation of the state vector $\underline{x}$ from t_k to t_{k+1}.

b. Compute the process covariance matrix $Q = Q(k,\Delta t_k)$, which may be a function of time and of the propagation step size Δt_k.

c. Compute

$$\bar{P} = \bar{P} + GQG^T, \tag{2}$$

where $G \triangleq G(k)$ is as defined by equation (1) in Appendix A.

d Factor $\bar{P}$ into $\bar{U}$ and $\bar{D}$ by means of the U-D Factorization Algorithm, specified in Figure Ap.B-2.

Although the computing implied by the above steps a through d is thought of as a stable process, it is noted on p. 188 of [Tho] that there exist important exceptions when F is large and/or P is ill-conditioned. In such situations, the resulting matrix P may have serious errors. Problems may also arise when, due to roundoff errors, some characteristic values become slightly negative.

On the positive side, the above covariance propagation algorithm yields (after Step c) $\bar{P}$ expressed in the canonical product form, various parts of which (especially its main diagonal elements) are often used for both on-line and postmission performance analysis. Also, the appearance of slightly negative characteristic roots in $\bar{P}$ can be avoided by keeping GQG^T sufficiently "large" compared to $\bar{P}$ and/or by monitoring and then boosting, on the basis of need, the elements of diagonal matrix $\bar{D}$. Another technique (in case G is not an identity matrix), borrowed from ridge regression, is to add to the right hand side of (2) a positive definite diagonal matrix D_1 on detection of the need to boost P.

G. U-D FACTOR PROPAGATION (TIME-UPDATE) ALGORITHM

This algorithm is based on modified Gram-Schmidt orthogonalization. It is described on pages 200-203 of [Tho].

In order to summarize it, we first need to define a weighted inner product of two n-component vectors $\underline{a}$ and $\underline{c}$, weighted (normalized) by the main-diagonal elements of an nxn matrix $B = \text{diag}\,[b_1,\ldots,b_n]$. We define this inner product as

$$(\underline{a}, \underline{c})_B = \underline{a}^T B \underline{c} = a_1 b_1 c_1 + \ldots + a_n b_n c_n. \tag{3}$$

Input: nxn symmetric matrix $\tilde{P}$, with main-diagonal and upper-triangular elements stored in an nxn array P.

Output: nxn unit-diagonal, upper-triangular matrix $\tilde{U}$, with its upper triangular portion stored in nxn array U (which optionally can be "equivalenced" with array P so that the original $\tilde{P}$ is destroyed).

Output: the main-diagonal elements of nxn diagonal matrix $\tilde{D}$ stored in vector D (which optionally can be stored in locations of the main-diagonal elements of array P).

Remark: the algorithm does not explicitly generate the main-diagonal unit elements of $\tilde{U}$.

```
UDPTCTR: begin

    for j = n, n-1,...,2 do:

      D̃j := Pj,j;

      α  := 1/D̃j;

      for k =1,...,j-1 do:

        β := Pk,j;

        Ũk,j := α*β;

        for i = 1,...,k do:

          Pi,k := Pi,k - β*Ũi,j;

        end

      end

    D̃1 = P1,1;

    end UDPCTR
```

Figure Ap. B-2. U-D Factorization Algorithm

Next, we use two matrices from the system model defined in Appendix A, the nxn state transition matrix $F = F(k+1, k)$ and the nxn_q process noise transformation matrix G, to define

$$W = [FU|G]. \tag{4}$$

[Here, $\hat{U}$ and $\hat{D}$ denote the measurements updates (from the preceding estimation cycle) of U and D, respectively.] Thus, W is an $nxN = nx(n + n_q)$ matrix, the jth row of which will be denoted by $\underline{w}_j^T$.

Finally, we combine in the indicated order the main-diagonal elements of nxn diagonal matrix $\tilde{D}$ with those of $n_q xn_q$ process noise matrix Q to define an NxN diagonal matrix $\tilde{D}$ (where again $N = n + n_q$) as

$$\begin{aligned}\tilde{D} &= \text{diag}\,[\tilde{D}_1,\ldots,\tilde{D}_N] \\ &= \text{diag}\,[D_1,\ldots,D_n, Q_{1,1},\ldots,Q_{n_q,n_q}].\end{aligned} \tag{5}$$

With the needed definitions completed, we are ready to summarize the U-D Factor Propagation Algorithm, which we do in Figure Ap. B-3.

H. CONCLUDING NOTES

Only the very basic forms (of "Bierman's method") of U-D factor covariance/gain processing algorithms have been summarized and compared here with the original form of Kalman's filter. For a more complete account of Bierman's approach refer to [Bie] or to [Tho]. For different approaches to "square root filtering", refer to Andrews [Andr] or Carlson [Car]. Chapter 6 of [And] and Chapter 7 of [May] contain textbook introductions to this topic. Comparative timing and sizing of filtering algorithms are discussed in [Bie], [Tho], and [May].

As noted in Section IV, in implemental design of real-time Kalman filters for microprocessor (or small computer) implementation, the value of "square root filtering" algorithms is mainly due to (1) their numerical stability, (2) their suitability for implementation in single-precision floating-point arithmetic (except for computation of some dot products), and (3) their reasonable computational efficiency compared to Kalman's original formulation. Criterion (2) is important in implementations of real-time estimators on the microprocessors which do not have floating-point hardware. The disparity between the processing speeds of single- and double-precision forms of floating-point arithmetic worsens as one goes from hardware to software (i.e., interpretive) implementationof this arithmetic.

Input: mxN matrix W (with rows $\underline{w}_1^T,\ldots,\underline{w}_n^T$).

Input: NxN diagonal matrix $\bar{D}$ defined by equation (5).

Output: the upper triangular part of nxn unit-diagonal, upper-triangular matrix $\tilde{U}$.

Output: the main-diagonal elements, stored as a vector, of nxn diagonal matrix D.

Define: $\underline{w}_j^{(0)} = \underline{w}_j$ for j = 1,...,n.

```
UDFCTRPR:  begin

           for j = n, n-1,...,2 do:

              D̃_j := (w_j^(n-j), w_j^(n-j))_D̄;

              for i = 1,..., j-1 do:

                 Ũ_ij := (w_j^(n-j), w_j^(n-j))_D̄ * (1/D̃_j);

                 w_j^(n-j+1) := w_j^(n-j) - (Ũ_i,j) * w_j^(n-j);

              end

           end

           D̃_1 = (w_1^(n-1), w_1^(n-1))_D̄;

           end UDFCTRPR
```

Figure Ap.B-3. U-D Factor Propagation (Time-Update) Algorithm

Constraints in the available processing resources and in the required real-time constraints usually motivate the exploitation of problem structure in order to reduce the processing load. There are three areas which should always be carefully examined and possibly exploited. The first is avoidance of floating-point operations on zero-valued operands. This can be attained via careful programming of algorithms. The second is structuring of vectors and matrices in the system model so as to introduce zero subvectors and submatrices, which would in turn yield an estimation problem of smaller size. This can often be accomplished through careful structuring of system model and mechanization of estimation algorithms. Finally, all kernel algorithms, operating on or producing matrices, should be designed to handle matrices as one dimensional arrays. The last feature allows efficient application of the same algorithm implementation (subprogram) to system models of varying dimensions. In some applications considered here, dimensions of the system model may change in real time.

APPENDIX C: ESTIMATION PROBLEM IN GPS USER'S NAVIGATION

A. INTRODUCTION

The present appendix summarizes the GPS estimation problem and defines a system model for it. This problem is cited as an illustration several times in the main body of the present exposition.

A user of GPS navigation equipment is assumed to be either moving or staying stationary close to the surface of the earth. Several different types of GPS user's navigation equipment for various classes of users (e.g., a stationary user, a land vehicle, an aircraft, or a ship) were recently developed or are still under development. [Dam], [Cyl 80], and [Upa] are samples of literature describing GPS user's navigation and/or its equipment. [Cox] discusses integration of GPS with inertial systems; [Van] describes GPS messages. The estimation problem of GPS navigation and its system model actually depend on the particular type of equipment under consideration. For pedagogical reasons, we overlook many technical details in the estimation problem and define an oversimplied system model for it.

B. COORDINATE SYSTEM

An earth-centered, earth-fixed (ECEF) coordinate system (with coordinate axes denoted by x, y, and z) is used in all GPS navigation processing described here. The z-axis of such a coordinate frame coincides with the polar axis of the reference ellipsoid; x and y lie in the equatorial plane. The particular version of ECEF frame assumed here has its x-axis pointing toward the Greenwich meridian; the y-axis 90° east of the x-axis.

C. NAVIGATION STATE VECTOR

The navigation state vector (which, in general, is a function of time) is defined by

$$\underline{s}^T = (b,f,x,y,z,v_x,v_y,v_z) \triangleq (b,f,\underline{p}^T,\underline{v}^T),$$

where:

b = the range bias in range measurements, due to a bias in the clock of user's navigation equipment set relative to the GPS time;

$f = db/dt$ = frequency drift rate of user's navigation equipment clock;

$\underline{p}^T = (x,y,z)$ = ECEF coordinates of the antenna phase center (PC) in user's navigation set;

$\underline{v}^T = (v_x,v_y,v_z)$ ECEF velocity components of the antenna PC.

In GPS navigation, even for moderate dynamics users, one usually models acceleration. For simplicity, acceleration is not modeled in the present case.

D. DISCRETE-TIME MODEL

The discrete time model of state vector dynamics is

$$\underline{s}(k) = F(k,k-1)\,\underline{s}(k-1) + \underline{w}(k-1).$$

Let

$$\Delta t_k = t_k - t_{k-1},$$

and ideally one would like to assume that $\underline{w}(k)$ is a zero mean white noise Gaussian process, with

$$E[\underline{w}(k)\underline{w}^T(k)] = Q(k,\Delta t_k),$$

$$E[\underline{s}(0)] = \hat{\underline{s}}(0), \text{ and}$$

$$E([\underline{s}(0) - \hat{\underline{s}}(0)][\underline{s}(0) - \hat{\underline{s}}(0)]^T) = P(0).$$

The state transition matrix F is defined by the following transformations:

$$b(k) = b(k-1) + \Delta t_k f(k-1),$$

$$f(k) = f(k-1)\exp[-\Delta t_k/\tau_b],$$

with the range bias correlation time τ_b assumed to be a constant or a slowly changing parameter; furthermore,

$$\underline{p}(k) = \underline{p}(k-1) + \Delta t_k \underline{v}(k-1);$$

$$\underline{v}(k) = \underline{v}(k-1).$$

NAVSTAR-GPS satellites (on the pseudo-range and delta pseudo-range measurements of which the navigation filter operates) will be referred to as space vehicles (SVs).

For each tracked SV, the navigation filter during a measurement processing cycle receives via the GPS receiver in the navigation set a pair of pseudo-range and delta pseudo-range (an observed change in pseudo-range over a count period of fixed length) measurements. For the j^{th} SV, SV_j, these two measurements will be denoted by $PR_j(t)$ and $DPR_j(t)$, respectively. A pseudo-range roughly is a range that has been synthesized from the readings of two distinct clocks (SV clock and user's navigation set clock) and that has not been corrected for the bias of user's navigation set clock with respect to the SV clock. In the sequel, it will be assumed that incoming pseudo-range measurements are already corrected (actually they are not) for other errors, such as the SV clock errors with respect to GPS system time and atmospheric signal delays. Thus, if b(t) denotes the true but unknown range bias at time t and if $PR_j(t)$ is the corrected pseudo-range from SV_j received at time t, then the range of the signal received from SV_j by user's navigation set at the same time t is represented by

$$R_j(t) = PR_j(t) - b(t).$$

Hence, the predicted pseudo-range measurement for SV_j at time t can be written as

$$\hat{PR}_j(t) = \hat{R}_j(t) + \hat{b}(t).$$

Similarly, the delta pseudo-range measurement $DPR_j(t)$ for SV_j at time t is defined by

$$DPR_j(t) = PR_j(t + \delta t) - PR_j(t - \delta t),$$

where $\delta t_{DPR} = 2\,\delta t$ is the duration of delta range count, which is characteristic of the receiver.

Hence, the measurement DPR_j is predicted at time t by means of

$$\begin{aligned}\hat{DPR}_j(t) &= \hat{PR}_j(t+\delta t) - \hat{PR}_j(t-\delta t)\\ &= \hat{R}_j(t+\delta t) - \hat{R}_j(t-\delta t) + \hat{b}(t+\delta t) - \hat{b}(t-\delta t)\\ &= \hat{R}_j(t+\delta t) - \hat{R}_j(t-\delta t) + \hat{f}(t)\delta t_{DPR}.\end{aligned}$$

In a measurement processing cycle, a (PR, DPR) measurement pair is received from each of four (or occasionally fewer) tracked SVs and subsequently processed by the navigation filter.

To complete the system model, the measurement equation at t_k is written as

$$\underline{m}(k) = h[\underline{s}(k),t_k] + \underline{u}(k),$$

where we assume that:

- The transpose of vector $\underline{m}(k)$ is of the form $[PR_1(k),\ldots,PR_{nk}(k), DPR_1(k),\ldots,DPR_{nk}(k)]$, with n_k (≤ 4) being the number of distinct SVs from which measurements are available at time t_k;
- $\underline{u}(k)$ is a zero mean, white noise Gaussian process with $E[\underline{u}(k)\underline{u}^T(k)] = \text{diag}[\delta^2_{PR1},\ldots, \delta^2_{PRnk}, \delta^2_{DPR1},\ldots, \delta^2_{DPRnk}]$;
- $\{\underline{w}(k)\}$ and $\{\underline{u}(k)\}$ are mutually independent stochastic processes, which are also independent of $\underline{s}(0)$.

It is assumed that the GPS receiver which acquires and preprocesses for the estimator pseudo-range and delta pseudo-range measurements is designed to furnish extra observables from which the estimator directly computes δ^2_{PRj} and δ^2_{DPRj}. Such a receiver is described in [War].

E. EQUATIONS FOR PREDICTION OF RANGES AND DELTA PSEUDO-RANGES

The signal range $R_j(t)$ from SV_j (received at time t) is computed by means of range equation

$$R_j(t) = ([x_j + \alpha y_j - x]^2 + [y_j - \alpha x_j - y]^2 + [z_j - z]^2)^{1/2} \quad (1)$$
$$= [(\Delta x_j)^2 + (\Delta y_j)^2 + (\Delta z_j)^2]^{1/2}$$

where:

- $\alpha = (\Omega)(\Delta t_{T(j)})$ is the angle by which the ECEF coordinate frame is rotated during the signal transit time from SV_j, with Ω representing earth's sidereal rate and $\Delta t_{T(j)}$ being the duration of signal transit from SV_j, a quantity which is unknown and must be estimated.
- (x_j, y_j, z_j) represents the ECEF position coordinates of V_j at time $t'_j = t - \Delta t_{T(j)}$; they must be computed by means of an extended form of Kepler's algorithm while using the transmitted orbital parameters (called ephemeris data) of SV_j. The best approach is to recompute periodically (at a relatively low rate) the least-squares polynomials for predicting, as a function of time, the position coordinates of SV_j, (x_j, y_j, z_j). This approach saves processor time and yields, by means of analytic differentiation of polynomials, approximations to the time derivatives $(\dot{x}_j, \dot{y}_j, \dot{z}_j)$ of (x_j, y_j, z_j). These time derivatives are useful in obtaining the time derivatives $(\Delta\dot{x}_j, \Delta\dot{y}_j, \Delta\dot{z}_j)$ of $(\Delta x_j, \Delta y_j, \Delta z_j)$.
- (x, y, z) are the ECEF coordinates of the position at time t of the antenna phase center in user's navigation set.

To predict $R_j(t)$ [i.e., to obtain $\hat{R}_j(t)$], the quantities (x,y,z) in equation (1) are replaced with their estimates at time t. The expression

$$R_j(t+\delta t) - R_j(t-\delta t), \quad (2)$$

needed in computing the predicted value of $DPR_j(t)$ at time t, can be (as experience has shown) efficiently and with sufficient accuracy approximated by means of

$$([dR(\tau)/d\tau]_{\tau=t})\delta t_{DPR} = [\Delta x_j \Delta\dot{x}_j + \Delta y_j \Delta\dot{y}_j + \Delta z_j \Delta\dot{z}_j]/R_j(t), \quad (3)$$

where, as earlier stated, δt_{DPR} is the duration of delta pseudo-range count. To obtain the predicted value of (2), one substitues in (3) the estimates of Δx_j, Δy_j, Δz_j, $\Delta\dot{x}_j$, $\Delta\dot{y}_j$, and $\Delta\dot{z}_j$ at time t.

F. MODELING OF PROCESS NOISE COVARIANCES

Suppose that the process noise manifests itself only through the unmodeled acceleration and through the range bias and the range bias rate (i.e., f = db/dt) components of state. Let Δt_k be the effective time step used for propagating the state error covariances P from t_{k-1} to t_k. Then, the process noise covariance at time t_k is of the form

$$Q(k, \Delta t_k) = \begin{bmatrix} Q_b(t_k) & \vdots & 0_{2x6} \\ \cdots & \cdots & \cdots \\ 0_{6x2} & \vdots & Q_d(k, t_k) \end{bmatrix}$$

where subscript "b" refers to range bias and its rate of change and subscript "d" to user's dynamics.

The 6x6 user's dynamics process noise submatrix Q_d is of the form

$$Q_d(k, t) = \begin{bmatrix} \bar{q}_{pp} I_3 & \vdots & \bar{q}_{pv} I_3 \\ \cdots & \cdots & \cdots \\ \bar{q}_{pv} I_3 & \vdots & \bar{q}_{vv} I_3 \end{bmatrix}$$

Hence, Q_d is constructed from three generating scalar parameters $\bar{q}_{pp}$, $\bar{q}_{pv}$, and $\bar{q}_{vv}$.

Define an auxiliary 2x2 matrix $\bar{Q}_d(k,\Delta t_k)$, constructed from these three parameters, as

$$\bar{Q}_d(k,\Delta t_k) = \begin{bmatrix} \bar{q}_{pp} & \bar{q}_{pv} \\ q_{pv} & q_{vv} \end{bmatrix} = \int_0^{\Delta t_k} [F_d(\tau)]\, N\, [F_d(\tau)]^T d\tau,$$

where $$F_d(\tau) = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 0 & 0 \\ 0 & n_{vv} \end{bmatrix},$$

and $n_{vv} = 2\sigma_v^2/\tau^2$ represents the power spectrum density of velocity noise due to unmodeled acceleration, σ_v^2 is the variance of velocity noise, and τ_v is the velocity correlation time constant. In general, σ_v^2 and τ_v are not time invariant.

Submatrix Q_b of Q is of the form

$$Q_b(\Delta t) = \begin{bmatrix} q_{bb} & q_{bf} \\ q_{bf} & q_{ff} \end{bmatrix},$$

where, assuming that the range bias rate correlation time τ_b is much greater than Δt,

$q_{bb} \cong (n_{bb})\Delta t + (n_{ff}/3)(\Delta t)^3$,

$q_{bf} \cong (n_{ff}/2)(\Delta t)^2$, and

$q_{ff} \cong (n_{ff})(\Delta t)$.

GLOBAL APPROXIMATION FOR NONLINEAR FILTERING WITH APPLICATION TO SPREAD SPECTRUM RANGING

W. Michael Bowles
Senior Scientist
Hughes Aircraft Company
El Segundo, CA

and

John A. Cartelli
Technical Staff
ESL
Sunnyvale, CA

This article discusses some global approximation procedures for nonlinear filtering. These procedures yield algorithms for recovering a signal from a measurement containing signal plus noise. The celebrated Kalman filter solves problems in which the signal evolves as the solution to a linear differential equation driven by white Gaussian noise with Gaussian initial condition and where the signal enters linearly into a measurement corrupted by white Gaussian noise. For problems where these linear Gaussian conditions are weakly violated, the Kalman filter serves as a good approximation. In many interesting problems, however, these conditions are not satisfied. For example, one of the random variables may have a multimodal density function or the signal may enter nonlinearly into the measurement. The techniques discussed here give the designer some tools to use when the Kalman filter is no longer a reasonable approximate solution to his problem.

1.0 ORGANIZATION OF ARTICLE

This article is organized as follows. The first section is devoted to a brief statement of the general problem addressed here, and a discussion is presented, by means of an example, of the difference between global and local approximations. The extended Kalman filter will be seen to be a local approximation, and the hope is that the example chosen will illustrate the inherent difference between the extended Kalman filter and approximations to be developed later. The example problem chosen to illustrate the difference between local and global approximation is that of analyzing a feedback loop with a nonlinearity in the forward path. The local approximation for this problem is a small signal analysis or linearization. The global approximation chosen is a describing function analysis. The difference between these two is that the global approach predicts fundamentally nonlinear phenomena like limit cycles, whereas the small signal gain approach cannot. The particular example chosen is also useful in that it will recur when filtering is discussed in the second section.

Section 2.0 develops some global nonlinear filtering approximations and illustrates their use with a radar ranging example. The first part of the section is devoted to the introduction of this radar example. A radio ranging problem, similar to the radar problem discussed here, provided the authors with motivation to explore global approximation. The problem is one that cannot be solved satisfactorily by an extended Kalman filter. The remainder of Section 2.0 is devoted to the introduction of particular approximation methods and their application to the radar problem. The applications are developed and compared and their performance demonstrated.

1.1 Problem Statement

The object of this article will be to give some approximate filtering algorithms for the problem of recovering the signal from a measurement containing signal plus noise. It will be assumed throughout this article that the signal to be estimated is a random process which satisfies

$$\dot{x} = f(x,t) + g(x,t)\, n(t) \tag{1}$$

where f and g are known functions and n(t) is an m-vector white noise with spectral matrix Q(t). The signal x(t) is an n-vector. It is further assumed that the measurement from which the signal is to be extracted is of the form

$$z(t) = h(x,t) + v(t) \tag{2}$$

where h is a known function and v(t) is a white noise with spectral matrix R(t) and is independent of n(t). Section 1.2 presents a radar tracking problem which has this form, and many other physical problems can be modeled by this set of equations. The vector x is called "the state vector" or "the message," depending on the situation. The vector z is called "the measurement." The object is to give some algorithms which process the measurements, z(t), to determine the state, x(t), with the smallest possible errors.

Some of the approaches to be used for extricating the signal from a noisy measurement

require the equations to be in discrete-time form. Discrete-time state and measurement equations have the form

$$x(i+1) = f(x(i), i) + g(x(i), i)\ n(i) \tag{3}$$

and

$$z(i+1) = h(x(i), i) + v(i) \tag{4}$$

where x and z are the discrete-time state and measurement, respectively. The functions f, g, h are assumed to be known. The noise sequences (h(i) and v(i) are white, independent and Gaussian.

1.2 The Difference Between Local and Global Approximation

In this section an example problem illustrates the differences between gobal and local approximations. The problem chosen is to analyze the behavior of a nonlinear feedback loop. The loop to be analyzed is diagrammed in Figure 1.

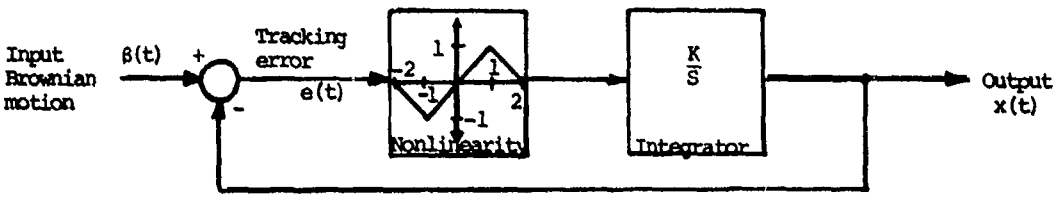


Figure 1. A Nonlinear Feedback Loop

This tracking loop is first order. It has a nonlinearity in the forward loop which is linear for errors smaller than one in magnitude and which goes to zero for errors larger in magnitude than two. The input, to be tracked by the loop, is a Brownian motion. A Brownian motion is a Gaussian random process, a typical time history of which might look like the trajectory in Figure 2.

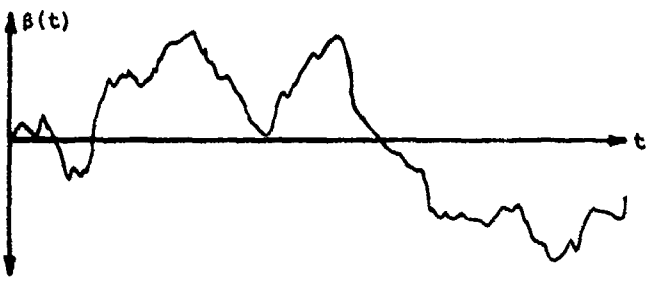


Figure 2. Possible Trajectory for a Brownian Motion

The Brownian motion can be thought of as the integral of a white noise. It shall be assumed that the spectral level of this white noise is q. The magnitude of q determines how difficult the input is to track.

The form of the forward loop nonlinearity is an obvious source of difficulty for this tracking loop. Since errors larger than two result in zero restorative force, it is expected that there is some level of q for which the loop will be unable to follow the input with a finite error. If the input were a ramp and q its slope, then it would be clear that above some magnitude of q the input would be untrackable by the loop in Figure 1. This would be clear because the tracking loop is first order and follows a ramp with a steady state error. For a ramp input to a first order loop, the magnitude of the steady state error is proportional to the ramp's slope. When the slope gets large enough that the steady state error is greater than one, the loop in Figure 1 will no longer follow it with a finite steady state error. The object is now to demonstrate the same kind of dependence of the loop following error on q for the given Brownian motion input.

Two approaches shall be taken. First, the loop will be linearized and analyzed statistically, and, secondly, a describing function approach shall be taken. The linearization approach is a local approximation and is directly analogous to an extended Kalman filter. The describing function approach is a global approach and is directly analogous to some of the approaches which shall be taken to the nonlinear filtering problem in Section 2.0.

To begin analyzing this loop consider the differential equation which the loop error satisfies

$$\frac{d}{dt}\, e(t) = -K\, f(e) + n(t) \tag{5}$$

In this equation f(e) is the forward loop nonlinearity shown in Figure 1, K is the forward loop gain, and n(t) is the white noise which is the derivative of the Brownian motion input. The first approach to analyzing this loop is to assume small error and linearize f(e) about zero. This yields the equation

$$\frac{d}{dt} e(t) = -Ke + n(t) \tag{6}$$

This linear equation is easy enough to analyze. The mean square tracking error is defined by

$$p(t) = E\{e^2(t)\} \tag{7}$$

and if it is assumed that $E\{e(o)\} = 0$ (that is, that the loop is initialized without any intentional error), then [Jazwinski (1)] p(t) satisfies the differential equation

$$\dot{p}(t) = -2K\, p(t) + q \tag{8}$$

The steady state mean squared error p_s satisfies $\dot{p}_s(t) = 0$ and is easily found to be $p_s = q/2K$. Large input dynamics (large q) result in large tracking errors and large forward tracking errors and large forward loop gain (large K means a fast tracking loop) results in small tracking error. If q is small enough and K large enough that the errors stay small, then this linearized analysis is a sufficient characterization of the loop's behavior. The problem with linearization is that it gives no indication of the loop's behavior as the errors get large. As the one sigma error predicted by linearized analysis gets large, only a small part of the error trajectories stay inside the range where the linearization is valid.

A second approach to analyze this loop is to use a describing function. The describing function procedure in this instance [Gelb and Vander Velde (2)] is to assume that the error is Gaussian and to replace the nonlinear block with the linear block whose output best matches in a mean square error sense that of the true nonlinearity. The gain of the linear block is called the describing function gain.

Assume that the error has a Gaussian density with variance p, then the describing function gain G(p) is

$$G(p) = \frac{1}{\sqrt{2\pi p}} \int_{-\infty}^{\infty} e f(e)\, \exp\left\{-\frac{e^2}{2p}\right\} de \tag{9}$$

Some manipulation yields the describing function gain to be

$$G(p) = \frac{2}{\sqrt{2\pi p}} \left[\int_0^1 \exp\left\{-\frac{e^2}{2p}\right\} de - \int_1^2 \exp\left\{-\frac{e^2}{2p}\right\} de \right] \tag{10}$$

This gain depends on p, the mean square tracking error and can be thought of as the gain an average error trajectory sees. As p, the error variance, gets small, G(p) approaches one, which is the linearized gain. As p gets large, the describing function gain G(p) approaches zero – a reflection of the fact that most of the error trajectories are outside the range to which the forward loop nonlinearity responds.

The differential equation for the tracking error when written using the describing function gain is

$$\frac{d}{dt} e(t) = -KG(p)\, e(t) + n(t) \tag{11}$$

The same approach as used for the linearized approach yields the variance equation

$$\dot{p} = -2KG(p)\, p + q \tag{12}$$

Dependence of the describing function gain on the variance complicates the variance equation. Finding the steady state variance requires solving the algebraic equation

$$G(p)\, p = \frac{q}{2K}$$

Closed form solution of this equation is not possible, but a graphical solution demonstrates its qualitative properties. The function G(p) p versus p is graphed in Figure 3. It is clear from the form of the graph that if q/2K is larger than about .42, then no steady state solution exists.

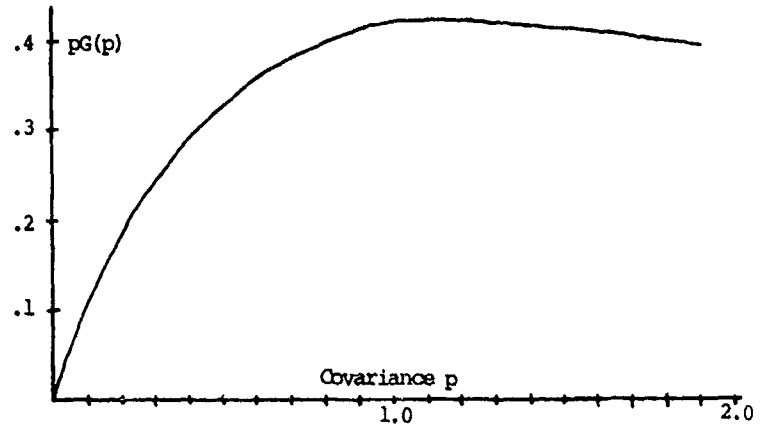


Figure 3. Left-Hand Side of Steady State Variance Equation

The describing function approach has provided two pieces of information which the linearization approach did not. First, it anticipates some gain reduction as the variance grows and correspondingly predicts worse tracking errors. It also predicts a combination of q and K for which the loop will be unable to track. The reason for the difference in the type of information these procedures give is that the describing function approximation depends on the global character of the nonlinearity. That is, it depends on the response of the nonlinearity to all different input magnitudes, not just one. The linearization approach, on the other hand, depends on the character (derivative) of the nonlinearity near zero. The describing function approach is therefore called a global approximation and the linearization approach is called a local approximation. In Section 2 these same ideas will be seen to apply to the nonlinear filtering problem. The extended Kalman filter will be shown to be a local approximation, and global approximations will be given. The difference in the type of information obtained from these two approaches will be of the same character as in the example just studied.

2.0 SOME GLOBAL APPROXIMATIONS FOR NONLINEAR FILTERING AND THEIR APPLICATION

This section gives some global approximations which have proven themselves useful in nonlinear filtering and applies some of these to some radio navigation and target tracking problems. The intent is to introduce some global approximations and to make clear how they are used by applying them.

The reader will observe that these global approximations are more burdensome to derive and implement than the extended Kalman filter. The global approximations will, as a reward, give filters that outperform the extended Kalman filter, particularly when the errors are large compared to the range for which linearization is valid. Additionally, these filters will offer more complete insight into the effect of the system nonlinearities on the filtering process. The describing function example in Section 1.2 illustrates the benefits of a global approximation. In that example the global approximation gave a more accurate description of the error covariance when the errors were larger than the linear range of the forward loop nonlinearity. As the error grew, the describing function analysis predicted gain reduction not predicted by the small signal approximation. In addition, the describing function approximation predicts a level of input that the tracking loop will no longer follow. Global filtering approximations will give improved fidelity and an analytical description of fundamentally nonlinear phenomena just as the describing function approach did.

The next things in this section are a brief discussion of nonlinear filtering and a description of the physical problem which serves as an example. A radar tracking problem is chosen as an example problem. Several different approaches are taken to this problem. Each approach leads to a different mathematical formulation of a nonlinear filtering problem. Some of the mathematical problems which stem from this radar problem are quite similar to those arising in other physical situations. Among these others are optical target tracking and laser communications.

The extended Kalman filter approach is demonstrated to be utterly unsuited for application to some of the formulations and inadequate in the others. The failure of the extended Kalman filter in this radar tracking problem makes it a good example problem for heralding the benefits of a global approach.

The last part of this section is devoted to some global approximations and to their application to the radar problem. The global approaches will prove to be quite effective. They will give filtering algorithms which are valid over a wide range of operating conditions, and they will provide an analytic description of the effect of the system nonlinearities. This description will be intuitively right and useful in understanding the filter's operation.

2.1 Background for Nonlinear Filtering Theory

As indicated earlier, the problem addressed here is that of extracting the signal from a noisy measurement. It is supposed that the signal evolves as the solution to a differential equation driven by noise. The signal will be denoted by the n-vector x(t) and is assumed to satisfy

$$\dot{x}(t) = f(x(t),t) + g(x(t),t)n(t) \tag{13}$$

The functions f and g are assumed known as is the initial probability density of x(o). The measurement z(t) is assumed to satisfy

$$z(t) = h(x(t),t) + v(t) \tag{14}$$

where h(x(t), t) is a known function. The time functions n(t) and v(t) are white noises. They are assumed to be independent and to have spectral density matrices

$$E\{n(t)\ n^T(t+\tau)\} = Q(t)\delta(\tau) \tag{15}$$

and

$$E\{v(t)\ v^T(t+\tau)\} = R(t)\delta(\tau) \tag{16}$$

The approach taken here to extract the signal from the noisy measurement of Eq. (14) is to compute at each point in time the conditional expectation of the signal, x, given all the measurements taken up to that time. Throughout what follows this conditional expectation shall be denoted by a hat. Thus, $\hat{x}(t)$ is the conditional expectation of x(t) given all measurements collected up to time t.

The measurements themselves are processed to yield the conditional mean, $\hat{x}(t)$. The measurements may be processed in other ways which might reasonably estimate the state vector. The conditional mean, however, is the estimator which minimizes the mean square error and is therefore a highly desirable one.

In this section the conditional mean is computed by propagating the solution to a differential equation. Kushner [Kushner(3)] first derived differential equations for the conditional mean. In fact he gave a differential equation for the conditional expectation of any twice continuously differentiable function of the state. Suppose $\phi(\cdot)$ is such a function, then

$$\frac{d}{dt}\hat{\phi}(x(t)) = \left[\widehat{\phi_x^T f} + \widehat{\tfrac{1}{2}\mathrm{tr}(gQg^T \phi_{xx})}\right] + (\hat{\phi}\hat{h} - \hat{\phi}\hat{h})^T R^{-1}(z(t) - \hat{h}) \tag{17}$$

where f, g, and h are the functions appearing in the problem description given in Eqs. (13) and (14). In Eq. (17) ϕ_x is the partial derivative of ϕ with respect to x and ϕ_{xx} is the second partial. The symbol tr stands for the trace of a matrix, that is, the sum of its diagonal entries. To obtain a differential equation for the conditional mean $\phi(x) = x$ is substituted into Eq. (17).

In general Eq. (17) cannot be solved in closed form. The reason is that on the right-hand side appear hats (^) denoting conditional expectation. Taking the conditional expectation requires having the conditional probability density function. Propagating the conditional mean using Eq. (17) is generally not enough to propagate the conditional probability density function.

In the case where the Kalman filter applies, the conditional density is Gaussian and can be characterized by its mean and variance. In this case propagating

$$\hat{x} \text{ and } \widehat{(x - \hat{x})(x - \hat{x})^T}$$

is all that is required to propagate the conditional probability density. Differential equations for the foregoing can be derived from Eq. (17). In the more general case, the mean and covariance are insufficient to characterize the conditional density. Generally, an infinite number of moments are required. It is not possible to propagate solutions to an infinite (or even large) number of differential equations, so some type of approximation is required. Such approximations are the subject of this article.

2.2 Description of a Radar Tracking Problem

A radar tracking problem serves, throughout this article, as an example. This section introduces the radar tracking problem to be used. The problem is described and posed mathematically. The basic task for a radar is to determine the distance, or range, from itself to some maneuvering target. A radar accomplishes this by transmitting a signal and measuring the time required for the signal to travel to the target, be reflected, and return to the radar transmitter. Often the same antenna that transmits the signal is used to receive the return. The return signal is then processed to estimate the two-way transit time. The range to the target is inferred from this transit time. In this article global approximations will be used to arrive at algorithms for processing the radar return.

Many different types of signal waveforms find use in radars. The algorithms to be developed here will apply directly, or with minor modification, to many of the different waveforms. To be specific, however, a particular waveform is chosen for this example. The basic waveform considered here is built from a pseudorandom number (PRN) code.

The pseudorandom number code is a piecewise constant waveform built from a sequence

of pseudorandom numbers. A pseudorandom number sequence is a sequence of numbers which can be generated systematically but which has some of the desired properties of a sequence of random numbers. A familiar example of such a sequence is computer generated noise. The method for building a PRN code from a PRN sequence is as follows. Suppose $\{a_i\}_{i\geq 0}$ is a sequence of pseudorandom numbers. Choose a length of time T and define a PRN code S(t) by

$$S(t) = a_i \qquad iT \leq t < (i+1)T \tag{18}$$

A possible PRN code S(t) is shown in Figure 4. The length of time T is usually called one chip. The inverse of T is usually called the chipping rate. For example, for a particular code with a chipping rate of 10 MHz, $T = 10^{-7}$ seconds.

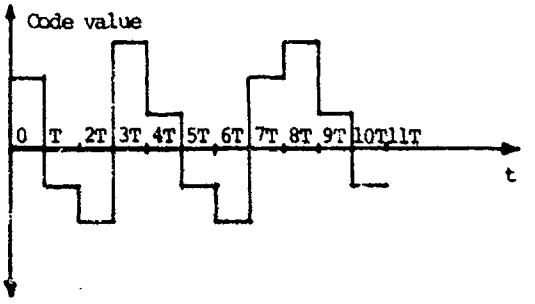


Figure 4. Graph of Possible PRN Code

A particularly useful class of the PRN codes is the class of linear feedback shift register codes. In practice such a code is generated by a pseudorandom number sequence in which the numbers are binary. These sequences look like coin toss sequences except that they repeat at some low frequency. Binary sequences are useful in practice because they can be generated by simple digital circuits.

A PRN code is used to build a signal waveform by multiplying a sinusoidal carrier wave by the PRN code. The result is then transmitted. If the PRN code is a ± 1 binary code, then multiplying this code by the carrier sinusoid simply introduces a 180° phase shift when the PRN code equals -1 or leaves the sinusoid unchanged when the code is +1.

After being reflected and received at the radar, the signal first has its carrier wave removed. What is left is a PRN code which is delayed by comparison to the code on the transmitted signal. The object is to determine how much this received code is delayed. There are several different approaches to the problem of determining the delay between the transmitted and received waveforms. Suppose that $S(t+\theta(t))$ is the received PRN code and $\theta(t)$ is the delay in that code. The delay $\theta(t)$ is a function of time because the target is moving and the two-way transit time changing. Besides the delayed code $S(t+\theta(t))$ the received signal contains some noise. In some cases this noise is thermal. In other cases the noise is predominantly due to other radio transmissions close in frequency to the radar's. A military target, for example, might transmit signals to confuse the radar. These signals would look like noise to the radar receiver.

The signal to be processed is a combination of the delayed PRN code, $S(t+\theta(t))$, and noise. If $z(t)$ is used to denote the signal to be processed, then mathematically

$$z(t) = S(t+\theta(t)) + v(t) \tag{19}$$

The sequence of events leading to this measurement is depicted in Figure 5.

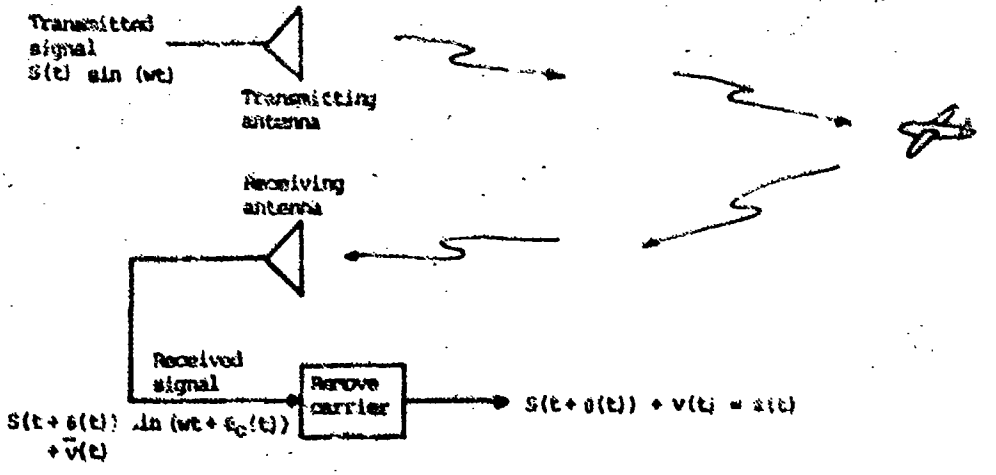


Figure 5. Sketch of Radar Tracking System

The measurement z(t) is scalar, and the measurement noise v(t) is white with spectral density r(t).

This radar tracking problem is now very nearly in the form of a nonlinear filtering problem. A state state variable model for $\theta(t)$ is all that is required to complete the mathematical formulation of the problem. Suppose that the radar antenna is at the origin of a three-dimensional Cartesian coordinate system. The target has coordinates $(x_1(t), x_2(t), x_3(t))$ in this system. The target coordinates evolve according to the equations of motion for the target. There is some uncertainty in the evolution of the target location, since the control inputs to the target (thrust level or changes in aerodynamic control surfaces) are unknown to the radar. The coordinates of the target can be aggregated into a vector x(t) which has n elements, the first three of which are the coordinates x_1, x_2, x_3. Other elements in the vector might be, for example, the target velocity or pitch angle. The target equation of motion can be put in the form

$$\dot{x}(t) = f(x(t)) + g(x(t))n(t)$$

The difficulty and importance of developing this state variable model is not to be minimized. The state model is the starting point for the filter designs which shall be presented. The effectiveness of the filter will depend on quality of the state variable model. A high dimension state vector leads to a computationally difficult design. It is necessary therefore to find a state model which captures the important features of the target motion in as short a state vector as possible. To be able to do this, the designer needs to understand both the physics of the target motion and the requirements of the filter not yet designed. Arriving at a state model is in itself a difficult problem.

For purposes of this article a simple target motion model will be used. It will be assumed that the delay $\theta(t)$ is a Brownian motion. That is,

$$\dot{\theta} = n(t) \tag{20}$$

where n(t) is a white noise with spectral density q. This target motion model is the simplest that still has some randomness. Even though it is a gross oversimplification of any real target's motion, it turns out to yield a filter that is useful for some problems. Part of the justification for using this model is that it keeps focus on the filter design problem instead of shifting it to the problem of modeling a particular target.

Mathematically, the radar signal processing problem can be posed as

$$\dot{\theta} = n(t) \tag{21}$$

$$z = S(t + \theta(t)) + v(t) \tag{22}$$

The extended Kalman filter (EKF) is inappropriate to apply to this problem. The EKF procedure requires differentiating $S(t + \theta(t))$ with respect to $\theta(t)$. Since $S(t + \theta(t))$ is piecewise constant, the derivative with respect to $\theta(t)$ is either zero or undefined (infinity). One could overcome this difficulty by using a finite difference instead of a derivative, but there is really no justification for a finite difference. The global approximations shown later apply directly to this problem. The fact that $S(t + \theta(t))$ is not continuous (or differentiable) does not cause them any difficulty.

There is another way to pose this radar signal processing problem. It is not quite so fundamental as the problem just posed. Some structure is imposed on the processor to arrive at a different formulation. There are two main reasons for considering this other formulation. First, this second formulation is mathematically the same as the problem of target tracking with a narrow field-of-view optical device (a telescope perhaps). This is an interesting and useful problem in its own right and a problem which appeals directly to most people's intuitions. Second, this second formulation is traditionally used to design radar signal processors. It is enlightening to see how traditional solutions compare to what global approximation yields and to compare the solutions obtained for these different formulations.

The second formulation of this radar tracking problem assumes that the received signal, after having the carrier signal removed, is multiplied by a code $S(t + \hat{\theta}(t))$. This code is the same as the received code except that it is delayed by $\hat{\theta}(t)$ instead of $\theta(t)$. The delay $\hat{\theta}(t)$ can be thought of as the delay that the radar expects for the received code to have.

The expected result of multiplying the received signal by $S(t + \hat{\theta}(t))$ is the autocorrelation of the PRN code at time shift $\theta(t) - \hat{\theta}(t)$ corrupted by some noise.

The code autocorrelation $R_{ss}(\tau)$ defined by

$$R_{ss}(\tau) = \lim_{u \to \infty} \frac{1}{2u} \int_{-u}^{u} S(t)S(t + \tau)dt \tag{23}$$

has the form

$$R_{ss}(\tau) = 1 - \left|\frac{\tau}{T}\right| \qquad \text{for } |\tau - NT_R| \leq T$$
$$\text{for N any integer} \qquad (24)$$
$$= \varepsilon(\tau) \qquad \text{otherwise}$$

where T is the code repeat time and $\varepsilon(\tau)$ is a function which is small compared to one. Figure 6 shows how $R_{ss}(\tau)$ might look. It will be assumed in what follows that the code autocorrelation $R_{ss}(\tau)$ has the form

$$R_{ss}(\tau) = 1 - \left|\frac{\tau}{T}\right| \qquad |\tau| < 1$$
$$= 0 \qquad \text{otherwise} \qquad (25)$$

This assumption is justified, as far as the consequent estimation procedure is concerned, if the *a priori* probability of the state (range) is concentrated on an interval T_R wide.



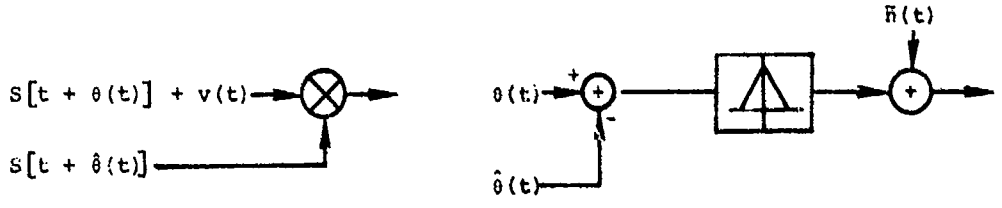
Figure 6. Picture of PRN Code Autocorrelation

In many practical systems the repeat time T_R may be very much larger than *a priori* timing uncertainties. For example, one ranging system in current operation has a repeat time T_R = 200 days, while initial timing uncertainties might be a few microseconds.

Mathematically the result of multiplying the measurement by a code with the expected delay is

$$z(t)S(t+\hat{\theta}(t)) = \left[S(t+\theta(t) + v(t)\right] S(t+\hat{\theta}(t))$$
$$= S(t+\theta(t))\, S(t+\hat{\theta}(t)) + \tilde{v}(t) \qquad (26)$$

In Eq. (26) $\tilde{v}(t) = S(t+\hat{\theta}(t))v(t)$ and is a white noise with spectral density r(t) just like v(t) was. The first term on the right-hand side of Eq. (26) displays a similarity to the integrand in the definition of the autocorrelation function. It seems reasonable to expect (and can be demonstrated) that the RHS of Eq. (26) and $R_{ss}\left[\theta(t) - \hat{\theta}(t)\right] + n(t)$ have equal time integrals, or, equivalently, that the outputs of the circuits shown in Figure 7(a) and (b) have equal time integrals. If these two outputs have equal time integrals, then a measurement processor which acts as a low pass filter will have the same response to one as to the other. The processors proposed later will act as low pass filters, so modeling the physical situation in Figure 7(a) by the block diagram of Figure 7(b) will be valid.



(a) Physical situation (b) Mathematical model

Figure 7. Model for Code Correlator

The essence of the arguments above is that, after some multiplications by known signals, the input signal may be taken to be a function $\tilde{z}(t)$ satisfying

$$\tilde{z}(\cdot) = R_{ss}\left[\theta(t) - \hat{\theta}(t)\right] + \tilde{v}(t) \qquad (27)$$

The input signal was correlated with a single known code to produce the measurement $\tilde{z}(t)$. Several such correlations of the input can be performed against several shifts of the known code (i.e., against codes $S[t + d_1(t)]$, $S[t + d_2(t)]$, ...). This produces several measurements $\tilde{z}_i(t)$

$$\tilde{z}_i(t) = R_{ss}\left|\theta(t) - d_i(t)\right| + \tilde{v}_i(t) \qquad (28)$$

where $d_i(t)$ is the time shift used to generate the i^{th} shift of the known code and the noises n_i are independent of one another.

This measurement model is quite similar in form to one which arises in optical target tracking. The measurement $\tilde{z}_i$ in Eq. (28) has two components. First, it contains noise. Second, it has a component due to signal. If the i^{th} delay, d_i, is equal to the delay on the received code, then the signal component is one. If the i^{th} delay is grossly different than the received delay, then the signal component is zero. A typical optical sensor has the image plane (television screen, for example) divided into a matrix of small squares. Each small square has a component of its output due to noise. It also has a component due to the target. The target component is one if the target is in the small square and zero if it is outside. The code tracking problem can then be visualized as a one-dimensional optical tracking problem.

It is worthwhile at this point to briefly describe how the delay of a PRN code has traditionally been tracked.

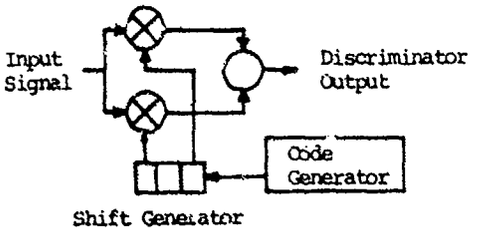


Figure 8. Code Loop Discriminator

Figure 8 shows a block diagram of the circuitry typically used to develop an error signal. The two multipliers shown in Figure 8 provide two correlations. The common input to these two multipliers is the received signal. The known code inputs to the correlators are time shifted by a fixed amount relative to one another. These time shifted codes are obtained by putting the code, out of the code generator, into a tapped delay line (shift register).

Suppose that the code in the center position of the shift register in Figure 8 is believed to be in synchronism with the input code. That is, the code $S[t + \hat{\theta}(t)]$ is in the center of the shift register. The code one shift to the right is then $S(t + \hat{\theta} - 1)$, and that one shift to the left is $S(t + \hat{\theta} + 1)$. The results of multiplying each of these with the code (disregarding input noise) on the input signal are modeled, according to earlier analysis, by $R_{ss}(\theta - \hat{\theta} + 1)$ and $R_{ss}(\theta - \hat{\theta} - 1)$, respectively. The difference between the two correlations with the input signal yields a measurement $z(t)$ which satisfies

$$z(t) = R(\theta - \hat{\theta} - 1) - R(\theta - \hat{\theta} + 1) + v(t) \tag{29}$$

The device shown in Figure 8 is called an early-late detector. Early-late refers to the fact that the correlations are with local codes which are earlier and later than the expected input. Figure 9 shows the mathematical model of the device in Figure 8.

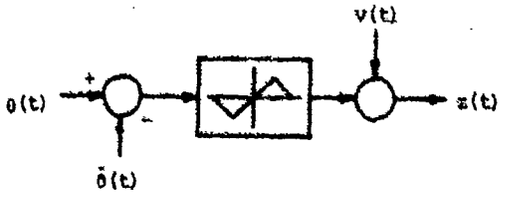


Figure 9. Mathematical Model of Code Loop Early-Late Detector

The early-late detector then gives an error measurement over a limited range. The circuit can be enclosed in a feedback loop and used to continuously track the input code delay. The resulting tracking loop is called a delay lock loop. In optical tracking problems there is an analogous error measuring circuit called a quadcell detector. This circuit has four optical sensors located in a square array - one sensor at each corner of the square. Imagine that the square is oriented so that its sides are either vertical or horizontal. The difference between the sum of the outputs of the two right cells and the sum of the outputs of the two left cells gives an azimuth error indication. The difference between the top and bottom gives an elevation error indication.

The problems of using a nonlinearity of this sort in a feedback loop are fairly obvious. Since the forward loop nonlinearity gives no output for errors greater than two, it cannot sustain errors of that size. This limits the input dynamics and noise level which the loop will track.

In constructing the early-late detector, only two multiplier outputs were used. If it appears to offer some advantage, many multiplier outputs can be provided. The object of the work which inspired the authors' interest in global approximation was to see if extra correlator outputs could be used to extend the range of delay dynamics and noise through which the delay can be tracked.

One approach to extending the range of operation is depicted in Figure 10. Figure 10(a) shows, on a single graph, the outputs of several early and several late multipliers. Adding all the early correlations and subtracting all the late correlations yields a broadened forward loop nonlinearity as shown in Figure 10(b). This method is suggested, for example, in [Spilker (4)] and in [Schiff (5)]. The difficulty with adding these extra multiplier outputs in this manner is that each additional correlation brings with it additional noise. The measurement resulting from adding more than two multiplier outputs is then noisier than the traditional early-late measurement. This means that in the cases where the traditional scheme is able to track with small errors, the extended range scheme will experience relatively larger errors. Because of this, the traditional wisdom is that extra multipliers are not useful for tracking. The fallacy with this is that the reasoning only applies in benign conditions where the traditional scheme can track with small errors. That is, the reasoning is based on a small signal analysis. Use of global approximation will indicate how this reasoning should be modified. If conditions are severe enough to cause the failure of the traditional scheme, then benefit can be derived by including extra multiplier outputs. This will be demonstrated in Section 2.3.



(a) Outputs of individual multipliers

(b) Nonlinearity obtained by subtracting all early multipliers from all late multipliers

Figure 10. One Method for Employing Multiple Correlator Outputs

Two starting points will be used to learn how to benefit from extra multiplier outputs. One approach starts by modeling the output of each multiplier as a separate measurement. The second approach starts by assuming the multiplier outputs are weighted and added together to yield a single nonlinear error measurement. This measurement nonlinearity, might, for example, look like the one graphed in Figure 10(b). In this second approach the shape of the nonlinearity is not specified. An optimum shape is determined by using global approximation techniques. This last approach is analogous to feedback communication wherein the moduation or measurement nonlinearity can be modified.

Altogether, three starting points have been suggested. The different approaches use the same model for the delay dynamics, but differ in that they use different models for the measurement. The first approach is to call the received code and noise in Eq. the measurement. The second and third approaches assume the received code has been multiplied by various shifts of a code generated in the receiver. The second approach considers the array of multiplier outputs to be the measurement. The third considers a single weighted sum of multiplier outputs to be the measurement.

None of these approaches is amenable to extended Kalman filtering. The extended Kalman filtering uses the partial derivative of the measurement nonlinearity with respect to the state variable evaluated at the expected state. As mentioned earlier, the first measurement model where the received code is the measurement is not amenable to this approach, because the code either has zero derivative or is not differentiable. The second approach where the multiplier outputs are considered measurements is not amenable, because the derivative of the code autocorrelation is zero further than one away from its peak. This causes the extended Kalman filter to ignore outputs of correlators further than two away from the expected delay. The third approach wherein the measurement is modeled as a single adjustable nonlinear function of the state is not amenable extended Kalman filtering either. Since the extended Kalman filter characterizes the nonlinearity by its slope evaluated at the expected value of the state, it considers the nonlinearities in Figures 9 and 10(b) to be equivalent. In terms of small signal behavior they are equivalent, but in terms of large error behavior they certainly are not. These approaches are all amenable to global approximation as will be seen in Section 2.3.

2.3 Gaussian Approximation

The radar tracking problem described in Section 2.2 has been mathematically described in several different ways. All of the mathematical problems which have arisen from the radar problem fit within the general framework of the nonlinear filtering problem introduced earlier. In this section an approximate solution for the nonlinear filtering problem will be used to arrive at solutions for the radar tracking problem. First, the approximation will be described generally. Next, the approximation will be applied to each of the mathematical problems which have arisen from the radar tracking problem. The inability of the extended Kalman filter to accomplish these design tasks will be pointed out, and the designs which result from the different mathematical formulations of the same radar problem will be compared.

In Section 2.1 Kushner's equations for propagating the mean and variance of the conditional density were presented. It was pointed out that the problem with these equations was that each equation required on its right-hand side the entire conditional density, while propagating the equation's left-hand side only yielded one moment of the conditional density. Since an infinite number of moments of the conditional density are required to

reconstruct it, an infinite number of equations must be solved to propagate the conditional density. To circumvent this difficulty an approximation will be used.

Suppose that the plant equation is

$$\dot{x} = f(x(t),t) + g(x(t),t)n(t) \tag{30}$$

and that an observation of the form

$$z(t) = h(x(t),t) + v(t) \tag{31}$$

is made. Differential equations for the conditional mean and covariance can be derived directly from Eq. (17) presented in Section 2.1. The differential equation for the mean is

$$\dot{\hat{x}} = \hat{f}(x,t) + \widehat{(x-\hat{x})h^T} R^{-1}(t)(z(t)-\hat{h}) \tag{32}$$

Denote the covariance matrix by P, that is

$$P = \widehat{(x-\hat{x})(x-\hat{x})^T} \tag{33}$$

The differential equation for the covariance is

$$\begin{aligned} P_{ij} = {} & \widehat{(x_i-\hat{x}_i)f_j} + \widehat{(x_j-\hat{x}_j)f_i} + \widehat{(gQg^T)_{ij}} - \widehat{(x_i-\hat{x}_i)h^T} R^{-1} \widehat{(x_j-\hat{x}_j)h} \\ & + \left[(x_i-\hat{x}_i)(x_j-\hat{x}_j)(h-\hat{h})\right]\hat{} \; R^{-1}(z(t)-\hat{h}) \end{aligned} \tag{34}$$

In this equation P_{ij} is the ij^{th} element of the covariance matrix P. The initial conditions for the conditional mean and covariance equations, Eqs. (33) and (34), come from the known initial density of the state. Specifically,

$$\hat{x}(o) = E\{x(o)\} \tag{35}$$

and

$$P(o) = E\left\{\left[x(o)-\hat{x}(o)\right]\left[x(o)-\hat{x}(o)\right]^T\right\} \tag{36}$$

The approximation to be used here is best explained by an intuitive inspection of the right-hand side of the mean and covariance equations. Notice that the conditional density is required on the right-hand side of these equations in order to carry out the expectation operations. These expectation operations are the integral of the indicated quantities against the conditional density. For example, the quantity $\hat{f}(x, t)$ appears on the right-hand side of the mean equation. This quantity can be expressed in terms of the conditional probability density. Suppose that the conditional density of the state at time t, given measurements up to time t, is $p_t(x|Z)$. The quantity $\hat{f}(x, t)$ is then given by

$$\hat{f}(x(t), t) = \int f(r, t)\, p_t(r|Z)\, dr \tag{37}$$

Since the conditional density appears in an integral like this, perhaps its precise shape is not critical to the accurate propagation of the conditional mean. The supposition is made that, in fact, as far as the quantities in the mean and variance equations are concerned, only the mean and variance of the conditional density are significant. If this is true, then the expectations on the right-hand side of the mean and variance equations can be carried out using any density which has the right mean and variance.

A density which is conveniently characterized by its mean and density is the Gaussian density and that is the one that will be used here. For example, using Gaussian approximation $\hat{f}(x, t)$ becomes

$$\begin{aligned} \hat{f}(x, t) &= \int f(r, t)\, p_t(r|Z)\, dt \\ &\approx \int f(r, t)\, \frac{1}{(2\pi \det P)^{n/2}} \exp\left\{\frac{1}{2}\left(r-\hat{x}\right)^T P^{-1}\left(r-\hat{x}\right)\right\} dr \end{aligned} \tag{38}$$

That is, all the conditional expectations on the right-hand side of the mean and variance equations are carried out by assuming that the conditional density is Gaussian in form with mean value $\hat{x}$ and covariance P. The derivatives of $\hat{x}$ and P appear on the left-hand side of the mean and covariance equations. What results is a coupled set of differential equations which can be solved by ordinary numerical methods. The effect of this

approximation is then to truncate the number of equations reuired to propagate the conditional density. Since the density has been supposed to only depend on its mean and variance, only the mean and variance equations need to be propagated. The procedure leads to equations which are much like the familiar Kalman filtering equations in form, but which, as will be seen, depend on the global character of the measurement and system nonlinearities.

2.4 Use of Gaussian Approximation to Determine Optimum Measurement Nonlinearity

The first mathematical problem on which Gaussian approximation shall be used is the deformable detector problem. Suppose that the input delay process $\theta(t)$ satisfies

$$\dot{\theta}(t) = n(t) \tag{39}$$

and that the nonlinear measurement, built from weighted correlator outputs, satisfies

$$z(t) = h(\theta(t) - \hat{\theta}(t)) + v(t) \tag{40}$$

The noise processes $n(t)$ and $v(t)$ have spectral densities $r(t)$ and $q(t)$, respectively. The shape of the measurement nonlinearity $h(e)$ depends on what weights the correlator outputs are multiplied by before being added together. Either one of the nonlinearities shown in Figures 9 and 10(b) could be achieved by some selection of weights. More generally, suppose that weight w_i is applied to the correlator shifted by i increments with respect to the expected on-time code. The result of this is a nonlinearity composed of straight line segments connecting the points $(-(n+1), 0)$, $(-n, w_{-n})$, $(-(n-1), w_{-(n-1)})$, ..., $(-1, w_{-1})$, $(0, w_0)$, ..., $(m, w_m)(m+1, 0)$ where the w_i are the arbitrary weights. This nonlinearity, shown in Figure 11, will be denoted by $h(e)$. Notice that for an integer i, $h(i) = w_i$. The detector drawn in Figure 11 would probably not be a useful one. The point is, however, that very general shapes are obtainable.

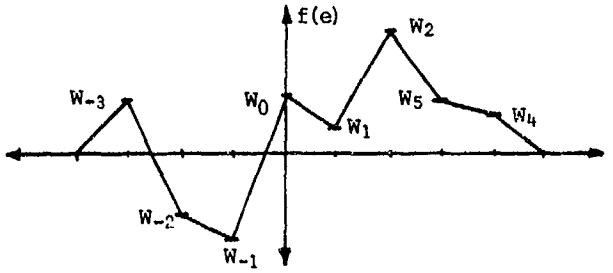


Figure 11. Hypothetical Detector Obtainable by Using Weights w_i

Noise is an essential consideration in determining the optimum detector shape. The noises present on the different correlator outputs are independent, zero mean, and white with equal covariances. Let $n_i(t)$ be the noise present on the output of the i^{th} correlator and let $E\{n_i(t)\ n_i(\tau)\} = r\delta(t-\tau)$. The spectral density of the noise on the weighted sum of the correlator outputs is then

$$E\left\{\sum_{i=-n} w_i n_i(t) \sum_{i=-n} w_i n_i(\tau)\right\} = r \sum_{i=-n} w_i^2 \delta(t-\tau) \tag{41}$$

The postdetection noise variance is then proportional to

$$\sum_{i=-n}^{m} w_i^2$$

The w_i must be chosen so that their signal detection assets outweigh their noise liabilities.

It will be assumed that the function $h(\cdot)$ is such that

$$\sum_{i=-n}^{m} w_i^2 \cong \int h^2(e)\ de \tag{42}$$

For the nonlinearities which will arise in this design, this condition will be satisfied.

The object is now to determine the measurement nonlinearity $h(\cdot)$ (weights w_i) which gives the best performance. The nonlinearity will be constrainod to have $\int h^2(e)de = 1$ and to be antisymmetric. Beyond that it is unconstrained. The procedure is to design a filter with $h(\cdot)$ unspecified and then to choose $h(\cdot)$ to optimize the filter's performance. The conditional moments [Fujisaki et al (6)] and [Clark (7)] satisfy

$$\dot{\hat{\theta}} = \widehat{(\theta - \hat{\theta})h}\,\frac{1}{r}(z(t) - \hat{h}) \tag{43}$$

and

$$\dot{p} = -\left[\widehat{(\theta - \hat{\theta})h}\right]^2 \frac{1}{r} + q + \widehat{(\theta - \hat{\theta})^2(h - \hat{h})}\,\frac{1}{r}\,(z(t) - \hat{h}) \tag{44}$$

These equations result from specializing Eqs. (32) and (34) to the present situation. A lower case p is used to represent the covariance to emphasize the fact that it is a scalar. Applying Gaussian approximation to this problem yields

$$d\hat{\theta} = \frac{G(p)}{r}\,dz \tag{45}$$

$$\frac{dp}{dt} = -\frac{G^2(p)}{r} + q \tag{46}$$

where

$$G(p) = \frac{1}{\sqrt{2\pi p}} \int eh(e)\,\exp\left\{-\frac{e^2}{2p}\right\} de \tag{47}$$

This can be changed into a more recognizable form by defining H(p) to be

$$H(p) = \frac{G(p)}{p} \tag{48}$$

$$= \frac{1}{p}\,\frac{1}{\sqrt{2\pi p}} \int_{-\infty}^{\infty} eh(e)\,\exp\left\{-\frac{e^2}{2p}\right\} de \tag{49}$$

The function H(p) is then the describing function gain for the nonlinearity h(·). If pH(p) is substituted for G(p), the filter equations become

$$d\hat{\theta} = \frac{pH(p)}{r}\,dz \tag{50}$$

$$\frac{dp}{dt} = -\frac{p^2H(p)^2}{r} + q \tag{51}$$

These equations can be recognized as the Kalman filter linearized with the describing function gains. Gaussian approximation can generally be interpreted as a Kalman filter linearized using a describing function. In general, there will usually be a data dependence in the covariance equation. Two elements of the problem under consideration combine to remove data dependence in the covariance equation. The measurement nonlinearity is a function of the estimation error, not of the state alone, and it is an antisymmetric function.

The mean and covariance equations represent an approximate solution to the filtering problem for an arbitrary nonlinearity h. The complete problem will be solved when the nonlinearity is selected to yield optimum filter performance. Inspection of the covariance reveals that only one term is affected by the choice of the nonlinearity. That term corresponds to the qudratic term in the usual Kalman filter. To minimize the covariance then, the best strategy is to maximize $h^2(p)$. Doing this makes the derivative of the covariance as small as possible. The optimum detector h* then satisfies

$$\max_{\|h\|_2=1} \left[\int_{-\infty}^{\infty} eh(e)\,\exp -\frac{e^2}{2p}\,de\right] = \left[\int_{-\infty}^{\infty} eh^*(e)\,\exp -\frac{e^2}{2p}\,de\right]^2 \tag{51}$$

This is equivalent to solving the unconstrained problem

$$\max_{\Gamma} \frac{\left[\int_{-\infty}^{\infty} e\Gamma(e)\,\exp -\frac{e^2}{2p}\,de\right]^2}{\int_{-\infty}^{\infty} \Gamma^2(e)\,de} \tag{52}$$

and setting

$$h^*(e) = \Gamma^*(e)\left[\int_{-\infty}^{\infty} \Gamma^{*2}(e)\,de\right]^{-1/2}$$

The Schwartz inequality may be used to solve for Γ^*.

$$\left[\int_{-\infty}^{\infty} e\Gamma(e)\ \exp\left\{-\frac{e^2}{2p}\right\}^2 de\right]^2 \leq \int_{-\infty}^{\infty}\Gamma^2(e)\ de \cdot \int_{-\infty}^{\infty} e^2\ \exp\left\{-\frac{e^2}{p}\right\} de \tag{53}$$

Then

$$\frac{\left[\int_{-\infty}^{\infty} e\Gamma(e)\ \exp\left\{-\frac{e^2}{2p}\right\} de\right]}{\int_{-\infty}^{\infty}\Gamma^2(e)\ de} \leq \int_{-\infty}^{\infty} e\ \exp\left\{-\frac{e^2}{p}\right\} de \tag{54}$$

and equality holds if

$$\Gamma^*(e) = e\ \exp\left\{-\frac{e^2}{2p}\right\} \tag{55}$$

Since

$$\int_{-\infty}^{\infty}\Gamma^{*2}(e)\ de = \int_{-\infty}^{\infty} e^2\left\{-\frac{e^2}{p}\right\} de \tag{56}$$

$$= \sqrt{\pi p}\ \frac{p}{2} \tag{57}$$

then it follows that

$$h^*(e) = \left(\frac{p}{2}\right)^{-1/2}\ \left(\pi p\right)^{-1/4} e\ \exp\left\{-\frac{e^2}{2p}\right\} \tag{58}$$

Some discussion of this nonlinearity is in order. The optimal forward loop nonlinearity is graphed in Figure 12.

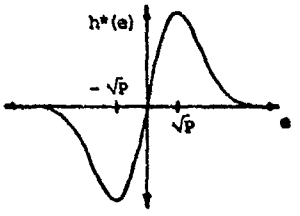


Figure 12. Optimum Measurement Nonlinearity

It is more or less linear over a range $(-\sqrt{p}, +\sqrt{p})$. An interesting feature of the design is that the nonlinearity is not fixed but changes in time. The nonlinearity depends directly on the tracking error variance p. Since p changes in time, then so does the nonlinearity. The variance, for example, might start out large when the radar is first turned on. As the signal is processed, the radar becomes more certain of the target whereabouts and the variance decreases. As the variance decreases, the linear range of the nonlinearity shrinks. The linear range of the nonlinearity is always between plus and minus one sigma of the predicted tracking error.

This general sequence of operations is more or less like ones used in many optical and radio signal acquisition systems. A narrow field of view is required for final tracking so that the final tracking accuracy is good, but a broad field is required during initialization because the initial uncertainties are usually large. The standard procedure then is to begin operation with a broad field device and, after the errors have been reduced, to switch over to a narrow field device. The procedure which has resulted from this nonlinear filtering approach agrees generally with standard practice and with intuition. This procedure offers the additional benefit that it provides an analytical framework for deciding when to switch between sensors.

Using the optimal detector gives mean and variance equations

$$\dot{\hat{\theta}}(t) = p\ \frac{H^*(p)}{r}\ z(t) \tag{59}$$

$$\frac{dp}{dt} = q - p^2\ \frac{H^{*2}(p)}{r} \tag{60}$$

where

$$H^* = \frac{1}{\sqrt{2\pi p}}\int_{-\infty}^{\infty} e h^*(e)\ \exp\left\{-\frac{e^2}{2p}\right\} de \tag{61}$$

and

$$h^*(e) = \left(\frac{p}{2}\right)^{-\frac{1}{2}} \left(\pi p\right)^{-\frac{1}{4}} e \exp\left\{-\frac{e^2}{2p}\right\}$$

Integrating yields

$$H^*(p) = \frac{\pi^{-1/4}}{2} p^{-3/4} \tag{62}$$

Thus

$$\dot{\hat{\theta}}(t) = \frac{\pi^{-1/4}}{2} \frac{p^{1/4}}{r} z(t) \tag{63}$$

$$\frac{dp}{dt} = q - \frac{\pi^{-1/2}}{4} \frac{p^{1/2}}{r} \tag{64}$$

The steady state covariance is then given by $\dot{p} = 0$ or

$$p^{1/2} = 4\pi^{1/2} qr \tag{65}$$

$$= 7.0898\ qr \tag{66}$$

Figure 13 shows results of Monte Carlo simulation of this tracking loop. One hundred runs were made, and the predicted and actual mean square tracking errors, as functions of time, are plotted in Figure 13. The conditions for these runs are as follows. The error at the start is Gaussian with expected value zero and variance 10.0. The spectral density of the noise in the state equation (called q) is 1.0. The spectral density of the measurement noise (called r) is 0.354. This spectral level is chosen so that the predicted steady state one sigma error is 0.25.

Figure 13 shows that for this problem Gaussian approximation does not work as well as one might hope. The covariance does not behave as it was predicted to behave. Figure 13 is a little misleading, however. The tendency is to believe that a particular trajectory will behave roughly like the one sigma trajectory. This is not the case, however. A sampling of the error trajectories indicates that for about 80 percent of the runs the errors behave as predicted. For the remaining 20 percent the errors are much worse than predicted. An average computed on the basis of these errors falls somewhere between the trajectories which are not captured and those that are. A given error trajectory, then, looks either somewhat worse or much better than the experimental one sigma plot which is shown.

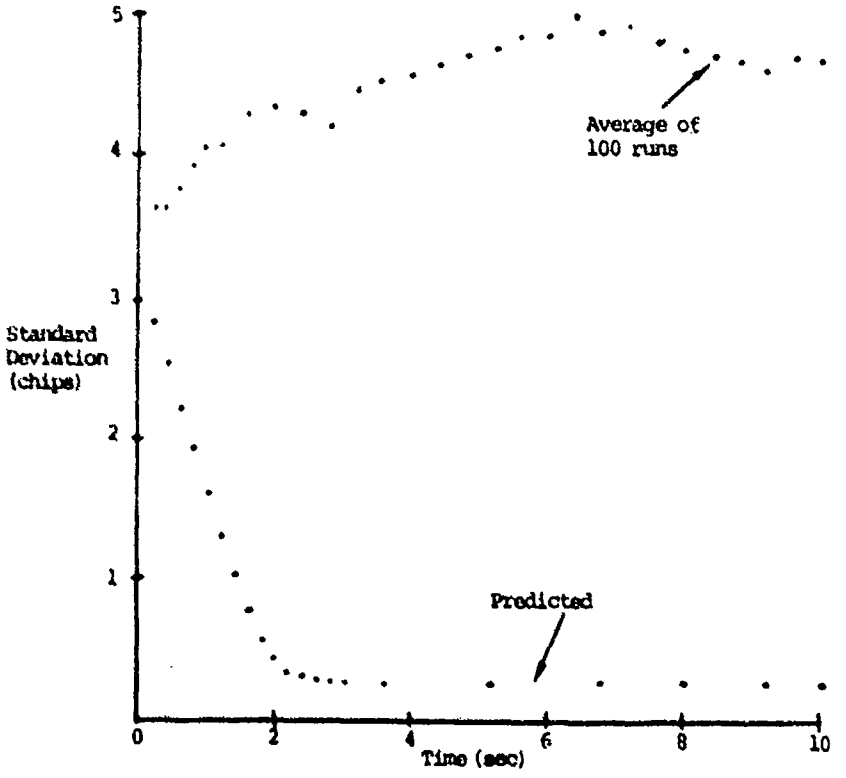


Figure 13. Acquisition Performance of Gaussian Approximation

The problem with the system, as it is now configured, is that the actual errors do not affect the covariance. Even when the actual errors are very large, the predicted covariance as plotted in Figure 13 is quite small. Somehow the actual errors must factor into the covariance computation. The reason that the measurement does not affect the covariance is that the measurement nonlinearity is antisymmetric. In the covariance equation, Eq. (44) for example, the coefficient on the measurement is

$$\frac{1}{r}\,\widehat{(\theta-\hat{\theta})^2\,(h-\hat{h})} = 0 \tag{67}$$

If a Gaussian density is assumed, then any antisymmetric function of $(\theta-\hat{\theta})$ has expected value zero. For this reason the data dependence drops out of the covariance equation and the covariance equation runs open loop. That is, the covariance is not responsive to actual errors.

This situation can perhaps be corrected by taking another measurement. The antisymmetric nonlinearity was constructed by adding weighted correlations, and a symmetric nonlinearity may be constructed by the same technique. In Section 2.5 it will be seen that this type of structure will follow naturally from posing the radar tracking problem differently. If multiplier outputs are treated as measurements, then using the Gaussian approximation procedure will result in a mean equation just like the one here but a significantly different covariance equation. The difference will be that the covariance equation will have an adaptive term. This term will be seen to drive the covariance, so that it agrees with the actual tracking error. Thus the covariance estimation will be closed loop instead of open loop.

2.5 Use of Gaussian Approximation for Processing Many Correlator Outputs

The next problem formulation to be considered is the one wherein the output of each multiplier is itself considered a measurement. This formulation imposes less structure on the problem that the last one did. In the last formulation it was assumed at the outset that multiplier outputs were to be weighted and summed together. The filter design was thus not allowed the option of processing each multiplier output individually and then combining them nonlinearly. The only degree of freedom the filter design was allowed to resolve was the shape of the weighting function. The approach taken there can be justified since it is parallel to the approach normally taken to design this type of signal processing hardware, and since the problem of determining an optimal measurement nonlinearity has application elsewhere. For radar signal processing, however, it will be seen that a more complete and better performing design can be achievdd by removing some of the constraints. In Sections 2.6 and 2.7 less constrained formulations shall be pursued.

Recall that when the outputs of the individual multipliers are considered as measurements, the system and measurement models become

$$\dot{\theta} = n(t) \tag{68}$$

and

$$z_i(t) = h_i(\theta-\hat{\theta}) + v(t) \tag{69}$$

where the measurement z_i is the output of the i^{th} multiplier. The function $h_i(x)$ is the code autocorrelation function at shift $x-i$ as graphed in Figure 14.

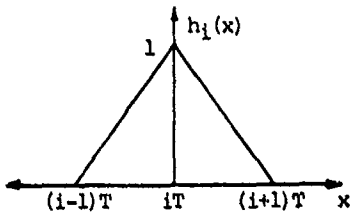


Figure 14. Output of i^{th} Measurement Nonlinearity Versus Tracking Error

Kushner's equation can be used to find the mean and covariance equation for this formulation just as it was for the last one. In this case the equations become

$$\dot{\hat{\theta}} = \sum_i \widehat{(\theta-\hat{\theta})h_i}\,\frac{1}{r}\,(z_i-\hat{h}_i) \tag{70}$$

and

$$\dot{p} = q - \left[\sum_i \widehat{(\theta-\hat{\theta})\,h_i}\right]^2 \frac{1}{r} + \sum_i \widehat{(\theta-\hat{\theta})^2(h_i-\hat{h}_i)}\,\frac{1}{r}\,(z_i-\hat{h}_i) \tag{71}$$

These are still unsolvable, but Gaussian approximation can be used to arrive at approximate mean and covariance equations just as in the previous section. Applying Gaussian approximation to this problem yields the mean equation

$$\dot{\hat{\theta}} = \frac{1}{r} \sum w_i z_i \tag{72}$$

and covariance equation

$$\dot{p} = q - \frac{1}{r}\left[\sum w_i\right]^2 + \left[\sum \alpha_i z_i - c\right] \tag{73}$$

In these equations the quantities w_i, α_i, c are given by the following:

$$h_i = \frac{1}{\sqrt{2\pi p}} \int_{-\infty}^{\infty} h_i(u) \exp\left\{-\frac{u^2}{2p}\right\} du \tag{74}$$

$$w_i = \frac{1}{\sqrt{2\pi p}} \int_{-\infty}^{\infty} u h_i(u) \exp\left\{-\frac{u^2}{20}\right\} du \tag{75}$$

$$\alpha_i = \frac{1}{\sqrt{2\pi p}} \int_{-\infty}^{\infty} u^2 h_i(u) \exp\left\{-\frac{u^2}{2p}\right\} du \tag{76}$$

$$c = \sum \alpha_i h_i \tag{77}$$

The mean equation in this case is somewhat different in appearance than the mean equation derived in Section 2.4. In reality, however, the two equations are strikingly similar. In the previous section the individual correlator outputs were weighted and summed to form a single measurement which entered the mean equation. The optimum weighting for the i^{th} correlator was determined to within a multiplicative function of p to be of the form $iT \exp(-(iT)^2/2p)$. The mean equation in this case is also driven by a weighted sum of correlator outputs. If p, the error covariance, is large compared to T, then the weight w_i is approximately

$$w_i \approx \frac{1}{\sqrt{2\pi p}} iT \exp\left\{-\frac{(iT)}{2p}\right\}$$

The filter starting with a less structured formulation has chosen to estimate the conditional mean with a structure quite similar to the one imposed during the formulation of the problem in the last section.

The covariance equation which has developed from the less structured formulation in this section is different than the covariance equation of the last section. The important difference is that measurements enter the covariance equation in this case. The effect of these measurements can be explained intuitively as follows.

The measurements are weighted and summed, and the result used to drive the covariance equation. Each measurement is some nonlinear function of the error (plus noise) and the weighted sum is also a nonlinear function of the error (plus noise). Thus, in this case the error affects the covariance directly. The weight applied to the i^{th} correlator output, when p is large, is

$$\alpha_i = \frac{1}{\sqrt{2\pi p}} \left((iT)^2 - p\right) \exp\left\{-\frac{(iT)^2}{2p}\right\}$$

This function is graphed below in Figure 16. These weights are applied to the correlator outputs and the weighted outputs summed. Thus some function of the tracking error, $h_2(e)$, enters the covariance equation. For integer argument i $h_2(i) = \alpha_i$. It is also true that when the error covariance, p, is large compared to one, the function $h_2(i)$ is the same as the function graphed in Figure 15. There is, then, a symmetric function of the tracking error which drives the covariance. The scalar c which is subtracted on the right-hand side of the covariance equation (Eq. 73) is equal to the expected value of $h_2(e)$. Expected value means integration against a Gaussian density with covariance p. The predicted covariance, p, then influences the value of c. If the predicted error covariance is on average equal to the actual error covariance, then $h_2(e)$ and -c cancel. When these terms cancel, the covariance equation reverts to the open loop equation derived in the last section. When these two terms do not agree, then a correction term is introduced to the covariance equation. This correction drives the covariance so as to bring the scalar c, the expected value of $h_2(e)$, and the average value of $h_2(e)$ into coincidence. The covariance equation is enclosed in a feedback loop and driven to match the actual errors.

Monte Carlo simulation of this tracking system was performed. The conditions for this simulation were identical to those leading to the results of the last section. Two changes have occurred due to the approach used in this section. The actual errors are smaller and the predicted errors are larger than those shown in Figure 13. There is a good match between the actual and predicted errors.

The approach taken in this section has been successful. It has yielded the qualitative information of the approach in the last section and has added a feedback term to the covariance equation. This addition has resulted in improved performance. The approach is still based on a formulation wherein structure is imposed. It is still the case that correlation of the received code with codes generated in the receiver is assumed. In the next section this structure will be removed, and what will result will be similar to the tracking loop derived here but will show improved performance.

2.6 Application of Gaussian Approximation to Estimation of PN-Code Delay

In the third formulation of the radar tracking problem the delayed PN code waveform is taken to be the measurement. This approach is the least structured of all those taken here. In addition a more general model for the delay process (target motion) will be assumed. It will be assumed that the delay is the first element in a vector whose elements are necessary to describe the target's motion.

Let $x_1(t)$ denote the varying PN-code delay due to target motion and $\underline{x}(t)$ equal the n-dimensional state vector whose first component equals $x_1(t)$. We interpret the above system as the two equations:

$$\begin{aligned} \dot{\underline{x}}(t) &= F\underline{x}(t) + Gn(t) \\ r(t) &= h(\underline{x}(t),t) + v(t) \end{aligned} \tag{78}$$

where

F = the state feedback matrix, possibly time varying
G = the input matrix, possibly time varying
n(t) = scalar white Gaussian noise (WGN) of spectral height Q which models the unknown target dynamics

$$h(\underline{x}(t),t) = S(t - x_1(t)) = \text{the received code} \tag{79}$$

v(t) = the received noise (WGN) of spectral height r(t), and
S(t) = the reference PN-code waveform (±1-valued)

The extended Kalman filter cannot be applied to this system because the measurement nonlinearity cannot be linearized about a reference trajectory. Attempting to linearize $h(\underline{x}(t),t)$ around a reference trajectory $\underline{x}_r(t)$ fails because

$$\left.\frac{\partial h(\underline{x}(t),t)}{\partial \underline{x}}\right|_{\underline{x}=\underline{x}_r} = \left[\frac{\partial h}{\partial x_1} \quad \frac{\partial h}{\partial x_2} \quad \cdots \quad \frac{\partial h}{\partial x_n}\right]\Bigg|_{\underline{x}=\underline{x}_r} = \left[\frac{\partial h}{\partial x_1} \quad 0 \quad \cdots \quad 0\right]\Bigg|_{\underline{x}=\underline{x}_r} \tag{80}$$

but

$$\frac{\partial h}{\partial x_1} = \frac{\partial}{\partial x_1} S(t - x_1) = -S'(t - x_1) \tag{81}$$

and $S'(t)$ is very badly behaved because of the switching discontinuities in the code. Extended Kalman filtering is hence ruled out at the start because of the sharpness of the nonlinearities in the observation equation. We shall have to resort to global approximation of the exact nonlinear filtering equations.

Let $\hat{\underline{x}}(t)$ denote the conditional mean of $\underline{x}(t)$ given the measurements, and we have by Jazwinski [1] formulas for the evolution of the first and second moments of the conditional probability density function of $\underline{x}(t)$:

$$\dot{\hat{\underline{x}}}(t) = F\hat{\underline{x}}(t) + \frac{1}{r}\left[\widehat{\underline{x}(t)h} - \hat{x}(t)\hat{h}\right]\left[z(t) - \hat{h}\right] \tag{81}$$

$$\begin{aligned} \frac{dP_{ij}(t)}{dt} &= \left[FP(t) + P(t)F^T + GQG^T\right]_{ij} - \frac{1}{r}\left[\widehat{x_i(t)h} - \hat{x}_i(t)\hat{h}\right]\left[\widehat{x_j(t)h} - \hat{x}_j(t)\hat{h}\right] \\ &\quad + \frac{1}{2r}\left[\widehat{(x_i - \hat{x}_i)(x_j - \hat{x}_j)h} - \widehat{(x_i - \hat{x}_i)(x_j - \hat{x}_j)}\,\hat{h}\right]\left[z(t) - \hat{h}\right] \end{aligned} \tag{82}$$

These equations are unsolvable, but a practical processing algorithm can be developed using Gaussian approximation. After applying Gaussian approximation, several simplifications can be made to the resulting equations.

The first simplification that can be performed on these equations is the elimination of the subtractive portion ($-\hat{h}$) of the innovations process ($z(t) - \hat{h}$) in the mean equation. This follows from two hypotheses: (1) the symmetry properties of the joint Gaussian distribution and (2) the assumption of high code frequency and lowpass processor. Explicitly, we assume "instantaneous averaging" of the product of two codes by the lowpass filter:

$$S(t - x)S(t) \approx R(x) \tag{83}$$

From Eq. (81) the term under examination is

$$-\frac{1}{r}\left[\widehat{\underline{x}(t)h} - \hat{\underline{x}}(t)\hat{h}\right]\hat{h} \tag{84}$$

which can be rewritten

$$-\frac{1}{r}\left[\widehat{(\underline{x}(t) - \hat{\underline{x}}(t))h(\underline{x}(t),t)}\right]\widehat{h(\underline{x}(t),t)}. \tag{85}$$

Using Gaussian approximation on Eq. (85) and merging the multiplied expectations yields

$$-\frac{1}{r}\int_{-\vec{\infty}}^{\vec{\infty}}\int_{-\vec{\infty}}^{\vec{\infty}}(\underline{X} - \hat{\underline{x}}(t))h(\underline{X},t)h(\underline{Y},t)N_{\underline{X}}(\hat{\underline{x}}(t),P(t))N_{\underline{Y}}(\hat{\underline{x}}(t),P(t))d\underline{X}d\underline{Y} \tag{86}$$

where $N_{\underline{X}}(\hat{\underline{x}},P)$ means $\underline{X}$ is normally distributed with mean $\hat{\underline{x}}$ and covariance P. Employing Eqs. (79) and (83) yields the useful identity

$$h(\underline{X},t)h(\underline{Y},t) = R(X_1 - Y_1) \tag{87}$$

Thus, Eq. (86) becomes

$$\frac{1}{r}\int_{-\vec{\infty}}^{\vec{\infty}}\int_{-\vec{\infty}}^{\vec{\infty}}(\underline{X} - \hat{\underline{x}}(t))R(X_1 - Y_1)N_{\underline{X}}(\hat{\underline{x}}(t),P(t))N_{\underline{Y}}(\hat{\underline{x}}(t),P(t))d\underline{X}d\underline{Y} \tag{88}$$

Define the error vector $\underline{e}(t)$ and associated dummy variables as such

$$\underline{e}(t) = \underline{x}(t) - \hat{\underline{x}}(t) \quad ; \quad \underline{V} = \underline{X} - \hat{\underline{x}}(t) \quad ; \quad \underline{W} = \underline{Y} - \hat{\underline{x}}(t) \tag{89}$$

and translate Eq. (88) from $(\underline{X}, \underline{Y})$ to $(\underline{V}, \underline{W})$ coordinates to get an expression for the i^{th} component of Eq. (88)

$$-\frac{1}{r}\int_{-\vec{\infty}}^{\vec{\infty}}\int_{-\vec{\infty}}^{\vec{\infty}}V_i R(V_1 - W_1)N_{\underline{V}}(\underline{0},P(t))N_{\underline{W}}(\underline{0},P(t))d\underline{V}d\underline{W} \tag{90}$$

This expression equals zero for all i because the function V_i is odd in the argument $(\underline{V}, \underline{W})$, while R, $N_{\underline{V}}$, and $N_{\underline{W}}$ are even in the same argument. Hence, the product is odd, and the integral is zero.

The second simplification occurs in the covariance equation. First, the subtractive portion of the third term of Eq. (82) is isolated.

$$-\frac{1}{r}\left[\widehat{(x_i(t) - \hat{x}_i(t))(\hat{x}_j(t) - x_j(t))} - \widehat{(x_i(t) - \hat{x}_i(t))(x_j(t) - \hat{x}_j(t))}\right]\widehat{h(\underline{x}(t),t)h(\underline{x}(t),t)} \tag{91}$$

Employing the Gaussian assumption and instantaneous averaging, we get that Eq. (91), after translation, equals

$$-\frac{1}{r}\int_{-\vec{\infty}}^{\vec{\infty}}\int_{-\vec{\infty}}^{\vec{\infty}}\left[V_i V_j - P_{ij}(t)\right]R(V_1 - W_1)N_{\underline{V}}(\underline{0},P(t))N_{\underline{W}}(\underline{0},P(t))d\underline{V}d\underline{W} \tag{92}$$

Secondly, the second term of Eq. (82) can be written

$$-\frac{1}{r}\left[\widehat{(x_i(t) - \hat{x}_i(t))h(\underline{x}(t),t)}\right]\left[\widehat{(x_j(t) - \hat{x}_j(t))h(\underline{x}(t),t)}\right] \tag{93}$$

Again, combining integrals under the Gaussian assumption, using instantaneous averaging, and translating the integration domain yields for Eq. (93)

$$-\frac{1}{r}\int_{-\vec{\infty}}^{\vec{\infty}}\int_{-\vec{\infty}}^{\vec{\infty}}V_i W_i R(V_1 - W_1)N_{\underline{V}}(\underline{0},P(t))N_{\underline{W}}(\underline{0},P(t))d\underline{V}d\underline{W} \tag{94}$$

The contention is that terms [Eqs. (92) and (94)] sum to zero. The calculation of these two integrals is detailed in Appendix I. They are

$$\text{Eq. (92)} = -\frac{1}{r}\left(P_{11}P_{j1}/P_{11}^2\right)\sqrt{P_{11}^3/\pi}\left[\exp\frac{-1}{4P_{11}} - 1\right] \tag{95}$$

$$\text{Eq. (94)} = +\frac{1}{r}\left(P_{11}P_{j1}/P_{11}^2\right)\sqrt{P_{11}^3/\pi}\left[\exp\left(\frac{-1}{4P_{11}}\right) - 1\right] \tag{96}$$

which negate each other, making receiver implementation much simpler.

Writing the remaining integral terms of Eqs. (81) and (82) in terms of the translated variable $\underline{V} = \underline{X} - \hat{\underline{X}}(t)$, we get the final result for the receiver

$$\dot{\hat{\underline{x}}}(t) = F\hat{\underline{x}}(t) + \frac{1}{r}\int_{-\vec{\infty}}^{\vec{\infty}} \underline{V} N_{\underline{V}}(\underline{0}, P(t)) S(t - \hat{x}_1(t) - V_1) dV_1 z(t) \tag{97}$$

$$\frac{dP_{ij}(t)}{dt} = \left[FP(t) + P(t)F^T + GQG^T\right]_{ij} + \frac{1}{r}\int_{-\vec{\infty}}^{\vec{\infty}} \left[V_i V_j - P_{ij}\right] N_{\underline{V}}(\underline{0}, P(t)) S(t - \hat{x}_1 - V_1) dV_1 z(t) \tag{98}$$

The receiver has the following notable features: (1) the first term of each equation is dynamics dependent and accounts for propagation of the estimated diffusion in accordance with the given dynamical model, (2) the second term updates the estimates continuously using the received signal. Both employ locally generated PN-code waveforms, $S(t - x_1(t) + V_1)$, (3) because of the use of the centered variable $\underline{e}(t)$ (or $\underline{V}$) in the two equations, there is feedback in the receiver from the present estimate ($x_1(t)$) to the phase of the local codes, and Eq. (98) drives itself and also drives Eq. (97) through adjusting the value of P(t).

The correspondence between this processor and those given earlier can be illuminated by assuming $\underline{x} = x_1$ and that F=0 & G=1 in [Eqs. (97) and (96)]. The receiver is specified by

$$\dot{\hat{x}}_1(t) = \frac{1}{r}\int_{-\infty}^{\infty} \frac{V_1}{\sqrt{2\pi P_{11}(t)}} \exp\left(\frac{-V_1^2}{2P_{11}(t)}\right) S\left(t - x_1(t) - V_1\right) dV_1 z(t) \tag{99}$$

$$\dot{P}_{11}(t) = Q + \frac{1}{r}\int_{-\infty}^{\infty} \frac{V_1^2 - P_{11}^2(t)}{\sqrt{2\pi P_{11}(t)}} \exp\left(\frac{-V_1^2}{2P_{11}(t)}\right) S\left(t - \hat{x}_1(t) - V_1\right) dV_1 z(t) \tag{100}$$

These equations can be interpreted in two ways. (1) The integrals in Eqs. (99) and (100) are convolutions, and thus prescribe linear filters through which the feedback corrected local code [$S(t - x(t))$] should be sent and subsequently multiplied by z(t). The covariance $P_{11}(t)$ is to be treated as a slowly varying filter parameter. (2) Moving z(t) inside the integral, the equations instruct us to form all correlations, $S(t - \hat{x}_1(t) + V_1)z(t)$, for all V_1 in $(-\infty, +\infty)$, and then use the specified weighting pattern to sum them. If we approximate the continuous domain integral by a grid with spacing of one chip, then we will have derived the structure discussed previously. Because of the difficulty of implementing interpretation (1), we shall use this second approach here.

Approximating Eqs. (98) and (100) by discretization of V , we get

$$\dot{\hat{x}}_1(t) = \frac{1}{r}\sum_{i=-\infty}^{\infty}\left[\frac{i}{\sqrt{2\pi P_{11}(t)}} \exp\left(\frac{-i^2}{2P_{11}(t)}\right)\right]\left[S(t - \hat{x}_1(t) - i) z(t)\right] \tag{101}$$

(weights; correlator outputs)

$$P_{11}(t) = Q + \frac{1}{r}\sum_{i=-\infty}^{\infty}\left[\frac{i^2 - P_{11}(t)}{\sqrt{2\pi P_{11}(t)}} \exp\left(\frac{-i^2}{2P_{11}(t)}\right)\right]\left[S(t - \hat{x}_1(t) - i) z(t)\right] \tag{102}$$

(weights; correlator outputs)

$S(t - \hat{x}_1(t) + i)$ is the local code which lags behind the on-time code, $S(t - \hat{x}_1(t))$, by i chips. The infinite sums in Eqs. (101) and (102) must be truncated for physical realizability. We shall assume that i goes from $-N_e$ to $+N_e$, and that the weights are small near those edge values. The weights in Eq. (101) are the same as those derived by [Boules (8, 9)] provided the values of $P_{11}(t)$ are the same. However, an auxiliary set of weights has been derived to drive the detector width, $2\sqrt{P_{11}(t)}$, making it a data dependent processor. The weighting patterns for the mean and variance equations are derived in Figures 15 and 16. Figure 15 is the extended detector characteristic, which provides the feedback signal to constantly null the error $e_1(t) = x_1(t) - \hat{x}_1(t)$. Figure 16 is interpretable as such: if z(t) should produce high correlation with codes $S(t - \hat{x}_1(t) - i)$ for $|i| < \sqrt{P_{11}(t)}$, then $P_{11}(t)$ is driven lower in order to produce a tighter mean detector weighting pattern for higher accuracy tracking; if z(t) correlates with $S(t - \hat{x}_1(t) - i)$ for $|i| > \sqrt{P_{11}(t)}$, then

$P_{11}(t)$ is increased to guard against losing lock. This data dependency helps guard against modeling errors and the inaccuracies of the Gaussian assumption.

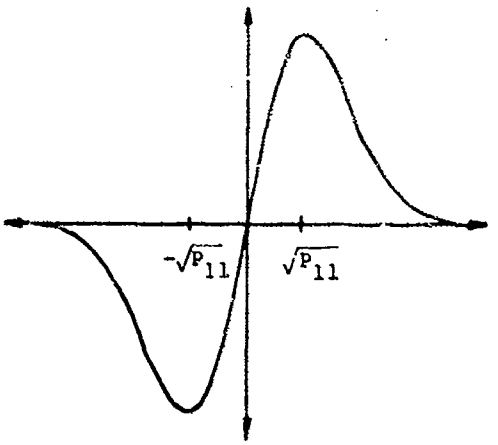


Figure 15. Weighting Pattern in Mean Equation

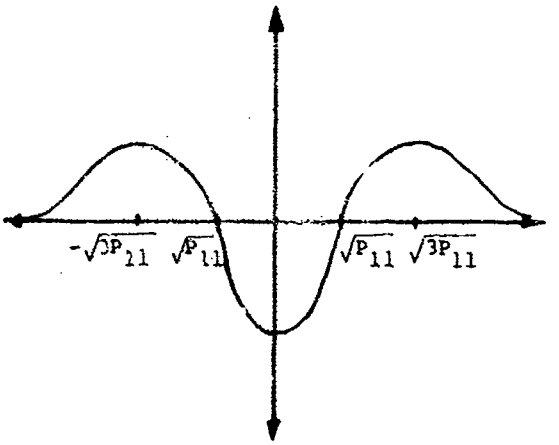


Figure 16. Weighting Pattern in Covariance Equation

A useful quality of this processor is that it is easily extensible to higher-dimension state variables. The two weighting-patterns derived are the only two necessary regardless of the dynamics order. This is seen from Eq. (97) and (98). The i^{th} component of the second term of Eq. (97) equals:

$$\frac{1}{r}\int_{-\infty}^{\infty} V_i\, N_{\underline{V}}\,(0,P(t))\,\delta(t-\hat{x}_1(t)-V_1)\,dV_1\,z(t) \tag{103}$$

Eliminating all integrations except those with respect to V_1 and V_i, and factoring the jointly-normal density of V_1 and V_i into a conditional times a marginal density, Eq. (103) becomes:

$$\frac{1}{r}\int_{-\infty}^{\infty} N_{V_1}\,(0,P_{11}(t))\,\delta\,(t-\hat{x}_1(t)-V_1)\left[\int_{-\infty}^{\infty} V_i\, N_{V_i}\left(\frac{P_{i1}}{P_{11}}V_1,\ P_{ii}-\frac{P_{i1}^2}{P_{11}}\right)dV_i\right]dV_1\,z(t) \tag{104}$$

The bracketed expression equals the conditional mean of V_i given V_1, which is $P_{i1}V_1/P_{11}$; hence, Eq. (104) becomes:

$$\frac{1}{r}\,(P_{i1}/P_{11})\int_{-\infty}^{\infty} V_1\, N_{V_1}\,(0,\ P_{11}(t))\,\delta\,(t-\hat{x}_1(t)-V_1)\,dV_1\,z(t) \tag{105}$$

This tells us that $\hat{x}_i(t)$ is to be driven by the output of the corresponding expression in the $\hat{x}_1(t)$ equation, scaled by $P_{i1}(t)/P_{11}(t)$.

Likewise, the second term of (98) is computable as:

$$\frac{1}{r}\left[\frac{P_{i1}\,P_{j1}}{P_{11}^2}\right]\int_{-\infty}^{\infty}\left[V_1^2-P_{11}(t)\right]\,N_{V_1}\,(0,\ P_{11}(t))\,\delta\,(t-\hat{x}_1(t)-V_1)\,dV_1\,z(t) \tag{106}$$

which specifies a scaling factor of $P_{i1}P_{j1}/P_{11}^2$ to be applied to the $P_{11}(t)$ equation's output to drive the corresponding term of the $P_{ij}(t)$ equation.

2.6 Discrete-Time Point-Mass Approximation Method for PDF Representation

a) The Point-Mass Approximation. We will now concern ourselves with an approximate method of solving for the conditional probability density function (PDF) of discrete-time systems. Here, a useful technique is to propagate the PDF only at a set of points

(grid) which support most of the PDF. To begin with, we specify the system and its exact recursive solution. The system is:

$$\underline{x}(t+1) = \underline{f}(\underline{x}(t),t) + \underline{g}(\underline{x}(t),t)\underline{n}(t) \tag{107}$$

$$\underline{z}(t) = \underline{h}(\underline{x}(t),t) + \underline{v}(t) \tag{108}$$

$$\underline{x}(0) = \underline{\xi} \tag{109}$$

where:

$\underline{x}(t)$ is an n-dimensional state-vector;

$\underline{z}(t)$ is a p-dimensional observation vector;

$\underline{n}(t)$ is a zero-mean m-dimensional WGN sequence of covariance matrix $Q(t)$;

$\underline{v}(t)$ is a zero-mean p-dimensional WGN sequence of covariance matrix $R(t)$;

$\underline{g}$ is a function which is matrix-valued (n x m)

$\underline{f}$ and $\underline{h}$ are nonlinear functions;

$\underline{\xi}$ is a zero-mean Gaussian random vector of covariance matrix $P(0|-1)$ and mean $\hat{\underline{x}}(0|-1)$ which provide the initial PDF for the filter

Letting $p_{t/\tau}(\underline{Y})$ denote the probability density of $\underline{x}(t)$ given the observations

$$Z_0^{\tau} \equiv \{\underline{z}(0), \underline{z}(1), \ldots, \underline{z}(\tau)\} , \tag{110}$$

it is necessary for the filter to produce two functions:

(1) The filtered PDF $p_{t/t}(\underline{Y})$ of the state $\underline{x}(t)$;

(2) The predicted PDF $p_{t+1/t}(\underline{Y})$ of the state $\underline{x}(t+1)$.

Once the PDF has been constructed, any type of estimate $\hat{\underline{x}}(t/t)$ or $\hat{\underline{x}}(t+1/t)$ can be generated, such as:

(1) $\hat{\underline{x}}(t/t) = \int_{-\infty}^{\infty} \underline{Y} p_{t/t}(\underline{Y}) d\underline{Y}$ = conditional mean, for a minimum-variance approach;

(2) $\hat{\underline{x}}(t/t) = \underline{Y}$ where $p_{t/t}(\underline{Y}) \geq p_{t/t}(\underline{\beta})$ for $\underline{Y} \neq \underline{\beta}$ for a maximum likelihood estimate

The filtering equation is simply an application of Bayes' law. Assuming that $p_{t/t-1}(\underline{Y})$ is available, and that $z(t)$ has just been observed, then

$$p_{t/t}(\underline{Y}) = \frac{\Pr\left[\underline{z}(t) | \underline{x}(t) = \underline{Y}\right] p_{t/t-1}(\underline{Y})}{\int_{-\infty}^{\infty} \Pr\left[\underline{z}(t) | \underline{x}(t) = \underline{Y}\right] p_{t/t-1}(\underline{Y}) d\underline{Y}} \tag{111}$$

But by the given normal distribution of $\underline{v}(t)$, which we will abbreviate as $N(\underline{0},R(t))$, we can deduce that

$$\Pr\left[\underline{z}(t) | \underline{x}(t) = \underline{Y}\right] = N(\underline{z}(t) - \underline{h}(\underline{Y},t), R(t)) , \tag{112}$$

thus the filtered update is recursively given by:

$$p_{t/t}(\underline{Y}) = \frac{N(\underline{z}(t) - \underline{h}(\underline{Y},t), R(t)) p_{t/t-1}(\underline{Y})}{\int_{-\infty}^{\infty} N(\underline{z}(t) - \underline{h}(\underline{Y},t), R(t)) p_{t/t-1}(\underline{Y}) d\underline{Y}} \tag{113}$$

To derive the predicted PDF, $p_{t+1/t}(\underline{Y})$, in terms of $p_{t/t}(\underline{Y})$, one must weigh and sum all the possible ways that $\underline{x}(t+1)$ can equal $\underline{Y}$. If we let $T_{t+1/t}(\underline{Y}|\underline{\beta})$ denote the transition probability, i.e., the probability that $\underline{x}(t+1) = \underline{Y}$ given that $\underline{x}(t) = \underline{\beta}$, then

$$p_{t+1/t}(\underline{Y}) = \int_{-\infty}^{\infty} T_{t+1/t}(\underline{Y}|\underline{\beta}) p_{t/t}(\underline{\beta}) d\underline{\beta} . \tag{114}$$

The function T is easily ascertained by knowledge of the distribution of $\underline{n}(t)$. Then we have:

$$T_{t+1/t}(\underline{\gamma}|\underline{\beta}) = N\left(\underline{\gamma} - \underline{f}(\underline{\beta},t) \ , \ \underline{g}(\underline{\beta},t)Q(t)\underline{g}^T(\underline{\beta},t)\right) \tag{115}$$

Hence the predicted density is recursively given by:

$$p_{t+1/t}(\underline{\gamma}) = \int_{-\infty}^{\infty} N\left(\underline{\gamma} - \underline{f}(\underline{\beta},t) \ , \ \underline{g}(\underline{\beta},t)Q(t)\underline{g}^T(\underline{\beta},t)\right) p_{t/t}(\underline{\beta})\,d\underline{\beta}. \tag{116}$$

Equations (113) and (116) allow recursive propagation of the conditional PDF. However, for general $\underline{f}$, $\underline{g}$, and $\underline{h}$, it is rare that closed-form solutions will be available. Hence, Bucy [16] and Bucy and Senne [11] developed the point-mass PDF representation.

The point-mass representation approximates $p_{t/t}$ and $p_{t+1/t}$ by a set of impulses:

$$p_{t/\cdot}(\underline{\gamma}) \approx \sum_{\ell_1,\ell_2,\ldots,\ell_n=1}^{2N+1} p_{t/\cdot}\left(\underline{b}(\ell_1,\ldots,\ell_n)\right) \ \delta\left(\underline{\gamma} - \underline{b}(\ell_1,\ldots,\ell_n)\right)$$

where $\underline{b}(\ell_1,\ldots,\ell_n)$ is a grid, taking values in R^n. The simpler scheme is to keep the grid fixed in time. A more advanced scheme, developed by Bucy [10], is to translate the grid to maintain its center on the conditional mean and rotate the grid to align its axes with the principle axes of the error ellipsoid. The simpler version turns the filtering and prediction equations of (113) and (116) into respectively:

$$p_{t/t}\left(\underline{b}(\ell_1,\ldots,\ell_n)\right) = \frac{N\left(\underline{z}(t) - \underline{h}\left(\underline{b}(\ell_1,\ldots,\ell_n),t\right),R(t)\right) p_{t/t-1}\left(\underline{b}(\ell_1,\ldots,\ell_n)\right)}{\sum_{m_1,\ldots,m_n=1}^{2N+1} N\left(\underline{z}(t)-\underline{h}\left(\underline{b}(m_1,\ldots,m_n),t\right),\ R(t)\right) p_{t/t-1}\left(\underline{b}(m_1,\ldots,m_n)\right)}, \tag{117}$$

and

$$p_{t+1/t}\left(\underline{b}(\ell_1,\ldots,\ell_n)\right) = \sum_{m_1,\ldots,m_n=1}^{2N+1} N\Big(\underline{b}(\ell_1,\ldots,\ell_n) - \underline{f}\left(\underline{b}(m_1,\ldots,m_n),t\right) \ , \tag{118}$$

$$\underline{g}\left(\underline{b}(m_1,\ldots,m_n),t\right)Q(t)\underline{g}^T\left(\underline{b}(m_1,\ldots,m_n),t\right)\Big) \ p_{t/t}\left(\underline{b}(m_1,\ldots,m_n)\right)$$

Equation (118) may, however, require a slight normalization to compensate for numerical innaccuracies which cause the total probability mass to deviate from 1. Alternatively, equations (117) and (118) may be combined and a single renormalization performed to account for the denominator of (117) and numerical innaccuracies.

This basic point-mass solution is not immediately useful to the problem of spread-spectrum ranging because that problem is naturally continuous-time. However, a close variant of the Bucy point-mass technique can be used to solve this system by approximating the exact continuous-time solution (Kushner's Equation) by point masses. Furthermore, the point-mass solution is very natural for PN-code tracking because of the discrete-time nature of the code.

Tracking the conditional mean $\underline{x}(t)$ requires computing the pdf of $\underline{x}(t)$, $p_{\underline{x}}$. This further requires an on-line computation of Kushner's Equation, from which Equations (81) and (82) stem. The full PDF-propagation equation is:

$$\frac{\partial p_{\underline{x}}(\underline{X},t|z_o^t)}{\partial t} = L[p_{\underline{x}}] + \frac{1}{r} p_{\underline{x}}(\underline{X},t|z_o^t)\left[h(\underline{X},t) - \widehat{h(\underline{x}(t),t)}\right]\left[\ z(t) - \hat{h}\right] \tag{119}$$

where L is the Fokker-Planck operator for our dynamics equation, (78), which is given by:

$$L[p_{\underline{x}}] = -\sum_{i=1}^{n}\left\{F_{ii}p_{\underline{x}} + F_i\underline{X}\frac{\partial p_{\underline{x}}}{\partial X_i}\right\} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} g_i Q g_j^T \frac{\partial^2 p_{\underline{x}}}{\partial X_i \partial X_j} \tag{120}$$

For:

F_i = i^{th} row of F;

F_{ij} = $(i,j)^{th}$ element of F; and

g_i = i^{th} element of $\underline{g}$.

Implementing this equation directly is impossible because of the continuous-valued domain over which p_X is defined, thus requiring infinite computer memory. This problem can be treated by Bucy's method of sampling p_X at a convenient spacing, and propagating p_X only at the chosen grid values of $\underline{X}$. Furthermore, this is natural for our system in particular because of the code waveforms, wherein a good spacing for the X_1-grid is 1 (chip).

Due to computational overhead of implementing higher-order dynamics-processes, I shall limit the discussion to the first-order process specified by $\dot{x}_1(t) = \xi(t)$. Then the discretized PDF is supported only by the natural unit-spaced grid selected for the x_1-axis. The following development is extensible to any n-dimensional process with its respective n-dimensional grid support. However, two important considerations must be borne in mind:

(1) For the derivative of $x_1(t)$, e.g., $x_2 = \dot{x}_1$ = velocity (chips/sec.), there is no natural grid-spacing as for x_1.

(2) Implementing an order-n process estimator will generally result in a different detector weighting-pattern for each component of the state-variable update. This results from the fact that the marginal and conditional densities of an arbitrary joint density are not necessarily of the same family, as was with the jointly Gaussian density.

Kushner's Equation exhibits the same dynamics/predictor, measurement/update type of structure, as evinced by the first and second terms of Equation (119). Equation (119) for our first-order case becomes:

$$\frac{\partial p_{\underline{X}_1}(X_1,t|\underline{Z}_o^t)}{\partial t} = \frac{1}{2} Q \frac{\partial^2 p_{\underline{X}_1}}{\partial X_1^2} + \frac{1}{r} p_{\underline{X}_1}\left[h(X_1,t) - \widehat{h(x_1(t),t)}\right]\left[z(t) - \hat{h} \right] \qquad (121)$$

Discretization of the dynamics term is best treated by approximating the second derivative by a three-point method. Letting Γ_1 denote the integer-separated values of X_1 that will support the sampled PDF, we may substitute

$$\frac{1}{2} Q \left[p_{\underline{X}_1}(\Gamma_1 + 1,\ t|z_o^t) - 2p_{\underline{X}_1}(\Gamma_1,\ t|z_o^t) + p_{\underline{X}_1}(\Gamma_1 - 1,\ t|z_o^t)\right] \qquad (122)$$

for the second derivative of p_{X_1} at Γ_1.

The observation term of Eq. (121) is best handled by deriving an equivalent term that is directly obtained from discrete-time and discrete PDF-space (Γ_1) probabilistic considerations. Handling this term directly in discrete time is justified by practical considerations. Given the continuous-time formulation of Eq. (121), a receiver would normally only approximate that calculation by discretizing all functions and derivatives in time. (This is simply because analog processing of the signals would be technically too difficult.) Hence, the associated approximations would be possible sources of inaccuracy. A rederivation of the observation term of Eq. (121) proved to be simpler and more robust to point-mass approximations than direct discretization of the same, under computer simulation.

Letting Δt denote the sampling period of the receiver, the correlations, $S(t-\Gamma_1)z(t)$, will be discretized in time and indexed by the integer n, for n=1, 2, 3,... Hence, our observed information is a sequence of $(2N_e + 1)$-dimensional random vectors parameterized by discrete-time index n:

$$\underline{v}(n) = \begin{bmatrix} v_{-N_e}(n) \\ v_i(n) \\ v_{+N_e}(n) \end{bmatrix} \quad \text{for} \quad n = 1,2,3,... \qquad (123)$$

where:

$$v_i(n) = \int_{(n-1)\Delta t}^{n\Delta t} z(\tau) S\left(\tau - \overbrace{(\hat{x}_1((n-1)\Delta t) + i)}^{\Gamma_i}\right) d\tau \ . \qquad (124)$$

N_e is the number of local-code phases ahead and behind the on-time code that are correlated with z(t). The estimate is also a discrete-time sequence; Δt is assumed to be small enough so that quantization error is acceptable. Notice that we will let the Γ_1-grid move rigidly as the estimate $\hat{x}_1(n\Delta t)$ moves, so that the center value of Γ_1 equals $\hat{x}_1(n\Delta t)$:

$$\Gamma_1 = \hat{x}_1(n\Delta t) + i \ , \ i = -N_e, -N_e+1, \ldots, +N_e \ . \tag{125}$$

This vector sequence of observations allows us to propagate the sampled PDF through Bayes' Law. Specifically, assume we have generated $p_{x_1}(\Gamma_1, (n-1)\Delta t | z_o^{(n-1)\Delta t})$ for $\Gamma_1 = \hat{x}_1((n-1)\Delta t) - N_e$ to $\hat{x}_1((n-1)\Delta t) + N_e$ in unit steps, and that:

$$\hat{x}_1((n-1)\Delta t) = \sum_{\Gamma_1} \Gamma_1 p_{x_1}(\Gamma_1, (n-1)\Delta t | z_o^{(n-1)\Delta t}) \ . \tag{126a}$$

Then we must generate $p_{x_1}(\Gamma_1, n\Delta t | z_o^{n\Delta t})$ for the same Γ_1, and compute the new estimate

$$\hat{x}_1(n\Delta t) = \sum_{\Gamma_1} \Gamma_1 p_{x_1}(\Gamma_1, n\Delta t | z_o^{n\Delta t}) \ . \tag{126b}$$

Lastly, the set of points, $p_{x_1}(\Gamma_1, n\Delta t, z_o^{n\Delta t})$, for the current Γ_1-grid (which is centered on $\hat{x}_1((n-1)\Delta t)$) must be interpolated to correspond to the values on the new Γ_1-grid centered on the new estimate $x_1(n\Delta t)$.

To begin this task, we must first apply the Fokker-Planck operator of Eq. (122) to get the predicted PDF values:

$$p_{x_1}(\Gamma_1, n\Delta t | z_o^{(n-1)\Delta t}) = p_{x_1}(\Gamma_1, (n-1)\Delta t | z_o^{(n-1)\Delta t}) + \frac{1}{2} Q\Delta t \Big[p_{x_1}(\Gamma_1+1, (n-1)\Delta t | z_o^{(n-1)\Delta t}) - 2p_{x_1}(\Gamma_1, (n-1)\Delta t | z_o^{(n-1)\Delta t}) + p_{x_1}(\Gamma_1-1, (n-1)\Delta t | z_o^{(n-1)\Delta t}) \Big] \tag{127}$$

for all Γ_1 in the current grid. The incoming vector $\underline{v}(n)$ then allows us to update these values to $p_{x_1}(\Gamma_1, n\Delta t | z_o^{n\Delta t})$ by Bayes Law:

$$p_{x_1}(\Gamma_1, n\Delta t | z_o^{n\Delta t}) = \frac{\Pr\left[\underline{v}(n) | x_1(n\Delta t) = \Gamma_1, z_o^{(n-1)\Delta t}\right] p_{x_1}(\Gamma_1, n\Delta t | z_o^{(n-1)\Delta t})}{\sum_{\Gamma_1} [\text{numerator}]} \tag{128}$$

Because $x_1(t)$ and n(t) are Markov processes and z(t) depends in a memoryless fashion upon them, we have:

$$\Pr\left[\underline{v}(n) | x_1(n\Delta t) = \Gamma_1, z_o^{(n-1)\Delta t}\right] = \Pr\left[\underline{v}(n) | x_1(n\Delta t) = \Gamma_1\right] \tag{129}$$

This likelihood function is easily computed by noting that the components of $\underline{v}(n)$ are distributed as such:

$$\begin{aligned} v_i(n) | x_1 = \Gamma_1 &\sim N(C\Delta t, r\Delta t) \text{ for } i = \Gamma_1 - \hat{x}_1((n-1)\Delta t) \ ; \\ v_i(n) | x_1 = \Gamma_1 &\sim N(0, r\Delta t) \text{ for all other } i \ . \end{aligned} \tag{130}$$

All $v_i(n)$ are independent since the correlations of white noise with the orthogonal local-code phases produce independent random variables. The single-chip grid-spacing is very important in this respect for computational simplicity. Thus, the likelihood function is given by:

$$\Pr\left[\underline{v}(n)\,|\,x_1(n\Delta t) = \Gamma_1\right] = \frac{1}{\sqrt{2\pi\Delta tr}} \exp\left[\frac{-\left(v_{\Gamma_1-\hat{x}_1}(n) - \Delta t\right)^2}{2\Delta tr}\right] \tag{131}$$

$$\times \prod_{\substack{i=-N_e \\ i\neq\Gamma_1-\hat{x}_1}}^{+N_e} \frac{1}{\sqrt{2\pi\Delta tr}} \exp\left[\frac{-(v_i(n))^2}{2\Delta tr}\right] ,$$

where $\hat{x}_1$ means $\hat{x}_1((n-1)\Delta t)$. Eq. (131) equals:

$$\left(\frac{1}{\sqrt{2\pi\Delta tr}}\right)^{2N_e+1} \exp\left[\sum_{i=-N_e}^{+N_e} \frac{-v_i^2(n)}{2\Delta tr}\right] \exp\left[\frac{2v_{\Gamma_1-\hat{x}_1}(n)\Delta t - (\Delta t)^2}{2\Delta tr}\right] \tag{132}$$

Using Eq. (132) in Eq. (128) yields the measurement-update formula:

$$p_{x_1}(\Gamma_1, n\Delta t|z_o^{n\Delta t}) = \frac{\exp\left[\frac{1}{r} v_{\Gamma_1-\hat{x}_1}(n)\right] p_{x_1}(\Gamma_1, n\Delta t|z_o^{(n-1)\Delta t})}{\sum_{i=-N_e}^{N_e} \exp\left[\frac{1}{r} v_i(n)\right] p_{x_1}(\hat{x}_1 + i, n\Delta t|z_o^{(n-1)\Delta t})} , \tag{133}$$

where $\hat{x}_1$ means $\hat{x}_1((n-1)\Delta t)$ and $\Gamma_1 = \hat{x}_1 + i$. A simple way to implement Eq. (133) is to scale the predicted probabilities $p_{x_1}(\Gamma_1, n\Delta t|z_o^{(n-1)\Delta t})$ by the factors $\exp(v_{\Gamma_i-\hat{x}_1}(n)/r)$ for each Γ_1. The scaled samples should then be normalized for unit sum, and Eq. (126b) employed to generate the new estimate. Lastly, the grid should be shifted to the new values,

$$i = \Gamma_1 - \hat{x}_1(n\Delta t),\ i = -N_e, -N_e+1, \ldots, +N_e , \tag{134}$$

An initial PDF assumption starts this recursive estimator/tracker.

Formulas Eq. (127) and (133), which specify the receiver for the Brownian-motion case, are comparable in computer overhead to the DDAG detector. These equations dictate the minimum-variance estimator for our state-space system with only the approximations of finite PDF grid and discrete time. Notice, however, that the grid-spacing was crucial to getting the measurement update formula for the Bayesian Detector, whereas in the DDAG detector, smaller spacing only implies sampling the weighting-patterns at more points.

CONCLUSION

The point-mass approximation is probably most useful where computing power is available and where the extended Kalman Filter is known to fail due to poor linearizability of f or h. Other global approximations are available which require less computation but which are not quite as general as point-mass and have their own associated problems. The most popular are:

(1) Gaussian Sums: derived by Alspach and Sorenson [12], this method is good for multimodal PDF's. It consists of representing the PDF as a sum of weighted Gaussian distributions (a non-orthogonal series) and propagating the weights essentially as a Kalman filter would.

(2) Other orthogonal series expansions: these include Edgeworth expansion [13], Gram Chalier series [14], Gauss-Hermite Polynomials [15], and Least-Squares Polynomial Approximations [16]. Generally two problems occur with these methods: (1) truncation of the series may result in some points of the PDF being negative, (2) truncation may result in an unnormalized PDF.

Due to the nature of the code-tracking problem (unimodality and the discrete-time nature of codes), the point-mass approximation was a natural choice for global nonlinear estimation.

References

1. Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

2. Gelb, Arthur and Wallace E. Vander Velde, Multiple-Input Describing Functions and Nonlinear System Design, McGraw-Hill, New York, 1968.

3. Kushner, H. J., "On the Differential Equations Satisfied by Conditional Probability Densities of Markov Processes, with Applications," J. SIAM on Control, Ser. A, Vol. 2, No. 1, pp. 1106-119, 1964.

4. Spilker, J. J., Digital Communications by Satellite, Prentice-Hall Inc., Englewood Cliffs, N. Y., 1977.

5. Schiff, M. L., "An Integrated Error Correcting/Pseudo Random Communication System," International Telemetry Conference, June, 1977.

6. Fujisaki, M., G. Kallianpur and H. Kunita, "Stochastic Differential Equations for the Nonlinear Filtering Problem," Osaka J. Math, Vol. 9, No. 1, 1972, pp. 19-40.

7. Clark, J. M. C., "Two Recent Results in Nonlinear Filtering Theory," in Recent Mathematical Developments in Control, D. J. Bell, ed., Academic Press, New York, 1973.

8. Bowles, W. M., Correlation Tracking, M.I.T. PhD. Thesis, June 1980.

9. Bowles, W. M., "Extended Range Delay Lock Loop," presented at 1979 NAECON conference.

10. Bucy, R. S., "Bayes Theorem and Digital Realizations for Nonlinear Filters," J. Astro. Sci., 17, 1969, pp. 80-94.

11. Bucy, R. S. and K. D. Senne, "Digital Synthesis of Nonlinear Filters," Automatica, 7, 1971, pp. 287-298.

12. Alspach, D. L. and H. W. Sorenson, "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," IEEE Trans. Auto. Ctrl., 17, 1972, pp. 439-448.

13. Sorenson, H. W. and A. R. Stubberud, "Nonlinear Filtering by Approximation of the A Posteriori Density," International Journal of Control, 8, 1968, pp. 33-51.

14. Srinivasan, K., "State Estimation by Orthogonal Expansion of Probability Distributions," IEEE Trans. Auto. Ctrl., 15, 1970, pp. 3-10.

15. Hecht, C., "Digital Realization of Nonlinear Filters," Proc. Second Symp. on Nonlinear Estimation Theory and its Appl., San Diego, September 1971, pp. 152-158.

16. Hildebrand, F. B., Introduction to Numerical Analysis, McGraw Hill, New York, 1956.

First, the computation of

$$- \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V_1 W_1 R(V_1 - W_1)\ N_{V_1}(0, P_{11}(t))\ N_{W_1}(0, P_{11}(t))\ dV_1\ dW_1 \qquad (I.1)$$

will be detailed. Inserting the multivariate Gaussian density function, this becomes:

$$\frac{-1}{2\pi P_{11}(t)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V_1 W_1 R(V_1 - W_1)\ \exp\left(\frac{-(V_1^2 + W_1^2)}{2P_{11}(t)}\right) dV_1 dW_1 \qquad (I.2)$$

Consider a rotation of the (V_1, W_1) axes by 45°, specified by:

$$\begin{bmatrix} g \\ f \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} V_1 \\ W_1 \end{bmatrix} \text{ or } \begin{bmatrix} V_1 \\ W_1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} g \\ f \end{bmatrix} \qquad (I.3)$$

Then, we can write:

$$V_1^2 + W_1^2 = f^2 + g^2 \text{ and } R(V_1 - W_1) = R(\sqrt{2}\, f) \qquad (I.4)$$

$$\text{and } V_1 W_1 = \tfrac{1}{2}(g + f)(g - f).$$

The function $R(\sqrt{2}f)$ is given by $1 - \sqrt{2}|f|$ for $|f| < 1/\sqrt{2}$, and 0 otherwise. Hence the full region of integration is a vertical strip $\sqrt{2}$ wide in the fg-plane (Fig. I.1). Since our three functions in Eqn. (I.4) are even, we may limit the integration to the shaded area indicated in Fig. (I.1), and multiply by 4. Thus, Eqn. (I.2) becomes:

$$\frac{-4}{2\pi P_{11}(t)} \int_0^{\frac{1}{\sqrt{2}}} \int_0^{\infty} \exp\left(\frac{-(f^2 + g^2)}{2P_{11}(t)}\right) \frac{(g + f)(g - f)}{2} (1 - \sqrt{2}\, f)\ dfdg, \qquad (I.5)$$

which becomes

$$(-2)\ \frac{1}{2\pi P_{11}(t)} \int_0^{\frac{1}{\sqrt{2}}} \int_0^{\infty} \exp\left(\frac{-(f^2 + g^2)}{2P_{11}(t)}\right)(g^2 - \sqrt{2}\, fg^2 - f^2 + \sqrt{2}\, f^3)\ dfdg \qquad (I.6)$$

a sum of four integrals over rectangular supports, and separable. Computation of the integrals and addition of the results yields the final value of:

$$+ \frac{P_{11}^3(t)}{\pi} \left[\exp\left(\frac{-1}{4P_{11}(t)}\right) - 1\right] \qquad (I.7)$$

as the value of (I.1).

Next, the value of

$$- \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(V_1^2 - P_{11}(t)\right) R(V_1 - W_1)\ N_{V_1}(0, P_{11}(t))\ N_{W_1}(0, P_{11}(t))\, dV_1 dW_1 \qquad (I.8)$$

will be shown to exactly cancel (I.1). This insures that a receiver with a one-dimensional state vector can safely drop the middle term in the variance equation. Expression (I.8) equals:

$$\frac{-1}{2\pi P_{11}(t)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(V_1^2 - P_{11}(t)\right) R(V_1 - W_1) \exp\left(\frac{-(V_1^2 + W_1^2)}{2P_{11}(t)}\right) dV_1 dW_1 \qquad (I.9)$$

Again, we shall align our new coordinate system with (I.3) and integrate over the region indicated in Fig. (I.1). Using

$$V_1^2 - P_{11}(t) = \frac{1}{2}(g + f)^2 - P_{11}(t) \; ; \; R(V_1 - W_1) = 1 - \sqrt{2}\, f \; ; \tag{I.10}$$

$$\text{and } V_1^2 + W_1^2 = f^2 + g^2$$

yields separable integrals, which all sum out to:

$$- \sqrt{\frac{P_{11}^3(t)}{\pi}} \exp\left[\left(\frac{-1}{4P_{11}(t)}\right) - 1\right], \tag{I.11}$$

which is the desired result.

To generalize this computation to the $(i, j)^{th}$ component of the covariance matrix requires first dealing with the term:

$$- \int_{-\vec{\infty}}^{\vec{\infty}} \int_{-\vec{\infty}}^{\vec{\infty}} V_i W_j R(V_1 - W_1)\, N_{\underline{V}}(\underline{O}, P(t))\, N_{\underline{W}}(\underline{O}, P(t))\, d\underline{V} d\underline{W} \tag{I.12}$$

Eliminating all variables of integration except for V_1, V_i, W_1, and W_j allows us to write this as:

$$- \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V_i W_j R(V_1 - W_1)\, N_{V_1, V_i}\left(\begin{bmatrix} o \\ o \end{bmatrix}, \begin{bmatrix} P_{11} & P_{1i} \\ P_{i1} & P_{ii} \end{bmatrix}\right) X \tag{I.13}$$

$$N_{W_1, W_j}\left(\begin{bmatrix} o \\ o \end{bmatrix}, \begin{bmatrix} P_{11} & P_{1j} \\ P_{j1} & P_{jj} \end{bmatrix}\right) dV_1 dV_i dW_1 dW_j$$

Re-arranging the remaining integrations:

$$- \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R(V_1 - W_1) \left[\int_{-\infty}^{\infty} V_i N_{V_1, V_i}\left(\begin{bmatrix} o \\ o \end{bmatrix}, \begin{bmatrix} P_{11} & P_{1i} \\ P_{11} & P_{ii} \end{bmatrix}\right) dV_i\right] X \tag{I.14}$$

$$\left[\int_{-\infty}^{\infty} W_j\, N_{W_1, W_j}\left(\begin{bmatrix} o \\ o \end{bmatrix}, \begin{bmatrix} P_{11} & P_{1j} \\ P_{j1} & P_{jj} \end{bmatrix}\right) dW_j\right] dV_1 dW_1$$

The bracketed integrals are computed using the properties of jointly Gaussian density function and recognizing conditional means and covariances. The two integrals are respectively computed to be:

$$\frac{P_{i1}}{P_{11}} V_1\, N_{V_1}(O, P_{11}) \; ; \; \frac{P_{j1}}{P_{11}} W_1\, N_{W_1}(O, P_{11}(t)), \tag{I.15}$$

which turns (I.14) into:

$$- \left(\frac{P_{i1}\, P_{j1}}{P_{11}^2}\right) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V_1 W_1\, R(V_1 - W_1) N_{V_1}(O, P_{11}(t)) N_{W_1}(O, P_{11}(t))\, dV_1 dW_1 \tag{I.16}$$

This is the same as expression (I.1) scaled by $P_{i1}P_{j1}/P_{11}^2$.

A similar computation shows that

$$- \int_{-\vec{\infty}}^{\vec{\infty}} \int_{-\vec{\infty}}^{\vec{\infty}} (V_i V_j - P_{ij})\, R(V_1 - W_1)\, N_{\underline{V}}(\underline{O}, P(t))\, N_{\underline{W}}(\underline{O}, P(t))\, d\underline{V} d\underline{W} \tag{I.17}$$

equals (I.8) scaled by the same factor. Thus, as (I.1) and (I.8) add to zero, so do

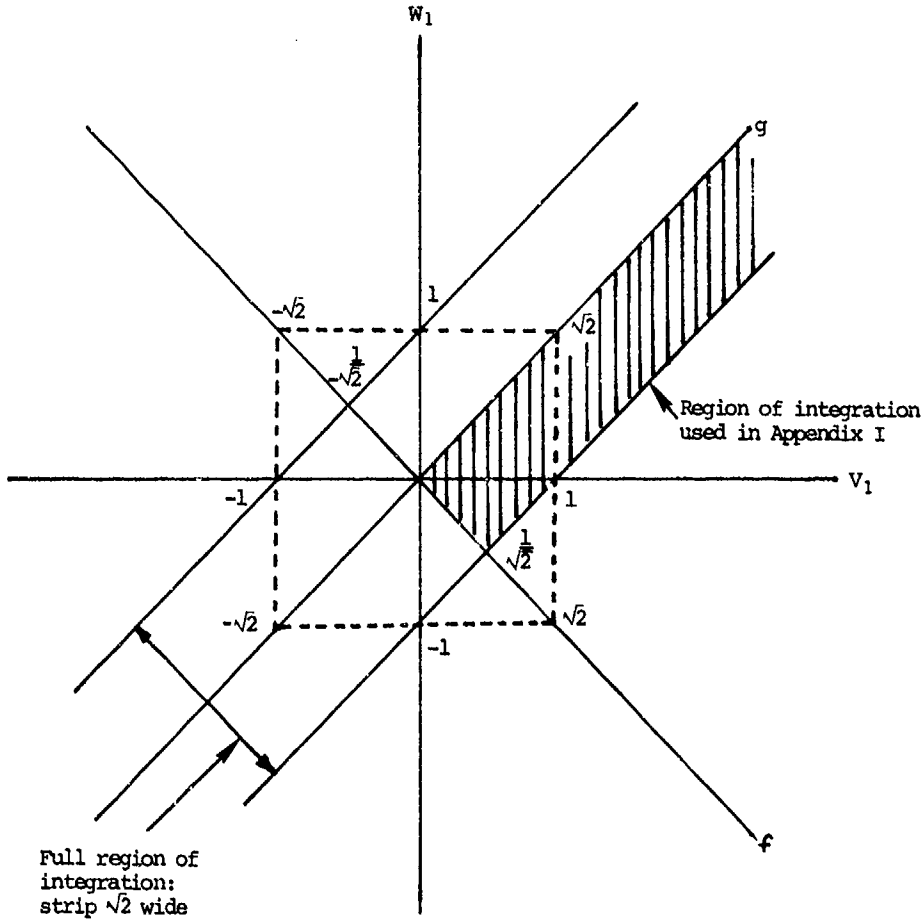


Figure A-I.1. Rotation Used in Appendix I.

SYSTEM IDENTIFICATION OF NONLINEAR AERODYNAMIC MODELS

T.L. Trankle, J.H. Vincent, S.N. Franklin
Systems Control Technology, Inc.
1801 Page Mill Road
Palo Alto, California 94304

SUMMARY

System identification is a technology for determining a mathematical model of a dynamic system from observations of its response to inputs. The effective application of system identification requires the integration of test planning (choice of sensors and of input test signals), monitoring of test execution, and data processing.

Identification technology is particularly useful for the determination of nonlinear aerodynamic models for aircraft maneuvering at high angles of attack. The methods outlined here (equation error, output error, and maximum likelihood algorithms) can directly estimate nonlinear aerodynamic coefficients in table look-up or multivariable spline formats. For application to nonlinear problems, the basic algorithms are enhanced by recent techniques for evaluation of partial derivatives of the likelihood function, calculation of parameter estimation uncertainties, and by the use of multidimensional splines as a generic model structure. An example application of these methods to the identification of F-4S fighter aircraft high angle of attack aerodynamics illustrates the technology.

NOMENCLATURE

Symbol	Symbol Definition	Units
A_x	Axial acceleration	ft/sec^2
A_y	Lateral acceleration	ft/sec^2
A_z	Vertical acceleration	ft/sec^2
b_w	Wing span	ft
$\bar{c}_w$	Wing mean aerodynamic	ft
C_D	Drag coefficient	--
C_L	Lift coefficient	--
C_ℓ	Rolling moment coeff.	--
$C_{\ell(\ )}$	$\partial C_\ell/\partial(\ )$	--
C_m	Pitching moment coeff.	--
$C_{m(\ )}$	$\partial C_m/\partial(\ )$	--
C_n	Yawing moment coeff.	--
$C_{n(\ )}$	$\partial C_n/\partial(\ )$	--
C_Y	Side force coeff.	--
$E(\ )$	Expected value of ( )	--
F_N	Net thrust	lb
f	State dynamics function	--
g	Gravity (32.2)	ft/sec^2
h_p	Pressure altitude	ft
J	Matrix of partial derivatives of measurements with respect to parameters	--
$\mathscr{L}$	Parameter estimation performance index	--
M	Fisher information matrix	--
M	Mach number	--
m	Number of data points	--
n_x	A_x/g	--
		--

Symbol	Symbol Definition	Units
n_z	$-A_z/g$	--
P	Roll rate	deg/sec (rad/sec)
p	Number of parameters to be identified	--
P	$(Pb_w)/(2V_T)$	--
Q	Pitch rate	deg/sec (rad/sec)
Q	$(Qc)/(2V_T)$	--
q	Dynamic pressure	lb/ft^2
R	Yaw rate	deg/sec (rad/sec)
R	$(Rb_w)/(2V_T)$	--
t	Time	--
u	System input	--
Var( )	Covariance of ( )	--
V_T	True airspeed	ft/sec
w_f	Fuel flow	lb/hr
x	System state	--
y	System output (measurement)	--
α	Angle of attack	deg
β	Sideslip	deg
δ_{AMB}	Ambient pressure ratio	--
δ_A	Aileron deflection	deg
δ_R	Rudder deflection	deg
δ_S	Horizontal tail deflection	deg
δ( )	Small perturbation in ( )	--
θ	Set of unknown parameters	--
θ_{AMB}	Ambient temperature ratio	--
ν_X	Measurement error of measurement X	--
ρ	Measurement autocorrelation	--
σ_X	Root mean square measurement error of measurement X	--
ω'	Angular rate nondimensionalized by length/speed	--

Super and Subscripts

$(\ )_{MEAS}$	Flight test measurement of ( )
$\hat{(\ )}$	Estimate of ( ) measurement
$(\ )_{B/B}$	Noseboom

1. MOTIVATION

System identification is a technology for determining a mathematical model of a dynamic system from observations of its response to its inputs. This technology has found application in a number of fields of engineering: aircraft aerodynamics [1], process control [2], electric power production and distribution [3], propulsion [4], medicine [5], econometrics [6], and structural dynamics [7].

The use of system identification methods can make unique contributions to the mathematical modeling of a dynamic system:

- Determination of parameters which model the dynamic, as opposed to the static, characteristics of the system. Examples of such parameters are those which model mechanical damping in a structure or the angular rate aerodynamic coefficients for an aircraft.

- Model validation. The use of input/output data from the operational system may be the only way to ensure validity of the mathematical system model.

Although system identification is often regarded as a set of techniques for data processing, the overall technology has a broader scope which includes test planning, instrumentation specification, and choice of mathematical model structure, as well as the numerical methods of parameter estimation and the statistical techniques of interpreting results. The overall scope of the problem can be illustrated with the aircraft aerodynamic modeling example. In planning the effort, the following questions should first be addressed.

- What unique contributions can system identification make to the particular modeling objectives?

- What is the purpose of the model? It might be used for active control design or for investigation of the physics underlying the aerodynamic phenomena. The ultimate purpose of the model will affect the choice of the stucture of the model to be identified and the estimation accuracy required.

- What test inputs are acceptable from the point of view of safety of operation while the data are being collected? Small control surface excursions may not excite the dynamic modes of interest while large excursions may cause spins or other dangerous maneuvers.

- What aspects of the input and output of the system must be measured? Possible measurements on the aircraft include

  - control surface position,

  - engine RPM and fuel flow,

  - attitude (roll, pitch, yaw),

  - angular rate,

  - angular acceleration,

  - translational acceleration,

  - velocity with respect to the air,

  - angle of attack and sideslip,

  - dynamic pressure, and

  - position.

- How accurately must these quantities be measured?

- What duration of experiment is most cost-effective? Long time records may be expensive to obtain, yet short time records may not yield accurate estimates. How high must the sampling rate be?

- How complex must the mathematical model be? How many parameters need to be estimated? For example, are aerodynamic forces and moments linear functions of vehicle states or do significant nonlinearities exist? The estimation of a much larger number of parameters may be required in the latter case.

- Can the quality of the data be tested as the experiment is being run? Can the system model be determined in real time or must it be determined by batch processing methods run off-line after the data have been collected?

- If the data from the experiment must be processed off-line following the experiment, what will be the expense of the processing? Given the nature of the chosen model, what type of processing method is required?

- How can we assess the validity of the model determined by processing the test data?

The technology of system identification is particularly well-suited to the processing of data from aircraft flight tests. Identification methods can extract information on mathematical aerodynamic models.

The development of an integrated flight testing procedure depends on the ability to identify nonlinear aerodynamic characteristics and propulsion system performance from flight test data. The identified models help to define performance, stability and control, and unaugmented airframe dynamic characteristics of the aircraft being evaluated. By identifying the nonlinear aerodynamic models in a multivariable, table-look-up format, direct correlations can be made with preflight aerodynamic predictions (e.g., wind tunnel data) and simulation models.

By using a data processing technique that can identify aerodynamic and installed propulsion models from many large-amplitude, dynamic test conditions, it is possible to enhance the test productivity

through a reduction in required test time. For the dynamic maneuvers, the test time is defined in terms of seconds. Other motivating factors that support the development of this technology include: improvements in safety of flight, and a general expansion of requirements for higher fidelity aerodynamic models of the aircraft.

Safety of flight can be enhanced during a flight envelope expansion test program by using system identification techniques to validate the aircraft simulation model for flight regimes already tested. The updated mathematical model can then be used to make preflight predictions for flight envelope expansion test conditions. In addition, when nonlinear identification models and identification techniques are used, the pilot's task is greatly simplified since he is not required to maintain small perturbation flight about a trimmed operating point.

The need for improved modeling of aircraft aerodynamic characteristics has been, and continues to be apparent in numerous areas of technical and operational importance. Four such areas are: 1) flying quality military specification compliance testing, 2) training simulations, 3) design methods for specification of aircraft characteristics, and 4) the development of mission profiles that make optimum use of the airplane's capabilities. In general, there is a need for an improved understanding of an airplanes's aerodynamic characteristics which supports design improvements for increased cost effectiveness, expanded mission flexibility and enhanced operational safety.

The objectives of this paper are to (1) outline the aspects of system identification which are relevant to the aircraft nonlinear aerodynamic identification problem, (2) introduce three new useful data processing techniques, and (3) demonstrate the application of the system identification technology to the analysis of flight test data from the Naval Air Test Center F-4S aircraft.

2. OVERVIEW OF SYSTEM IDENTIFICATION METHODOLOGY

2.1 The Integrated System Identification Process

2.1.1 Iterative Nature of the Process

Figure 1 indicates an integrated identification procedure which applies to a wide variety of dynamic system types (e.g., structural dynamics, vehicle dynamics, etc.). This system identification process consists of an iterative loop of test planning, actual testing, and data processing. Two feedback loops can be vital to the overall success of the program. An inner loop is closed during the data collection phase. This inner loop checks the quality of the data produced by the tests. The checking is done in real-time or nearly in real-time. If the quality of the data is determined to be poor, then corrective action can be taken while the test crew and facilities are still available. For example, the test can be repeated with modified inputs or sensor configurations. Poor data quality may be due to:

- failed sensor channels,
- excessively noisy sensor channels, or
- failure to excite dynamic modes of interest.

Data quality evaluation may be performed by a variety of real-time techniques including:

- visual inspection of data records as displayed by strip chart recorder or CRT,
- fault detection filters [8], or
- actual parameter estimation using algorithms configured for high computation speed [9].

An outer loop of test planning may also need to be closed about the entire identification process. The test planning is a "boot-strap" process. A model of the system is needed in order to choose test input signals and to specify sensor requirements. The characteristics of the model which result from the identification data processing may indicate that additional input signals or sensors are required for complete model identification. If this is the case, then another set of data collection tests will be required.

2.1.2 The Pretest Planning Phase

The pretest planning phase includes the specification of an instrument system and test inputs. The overall technology of system identification now extends to include analytic methods of specifying test input signals to minimize the uncertainty of estimates of parameters [10] and of specifying sensor accuracy requirements [11].

2.1.3 The Data Processing Phase

The actual processing of the data requires four major steps. These are:

- flight data processing and analysis,
- model structure determination,
- parameter estimation, and
- model validation.

2.1.3.1 Flight Data Preprocessing and Analysis

The preprocessing and analysis of flight test data is a major element of the integrated system identification procedure. The overall objectives of this task are to review measurement excitation, remove wildpoints, reconstruct unmeasured quantities (i.e., acceleration of the aircraft at the center of gravity), and develop a set of kinematically consistent measurements. Where measurement consistency cannot be established, requirements for instrumentation system error source modeling are defined. Measurement consistency has a significant impact on parameter identification accuracy since unaccounted-for errors will bias parameter estimates.

2.1.3.2 Model Structure Determination

The model structure determination phase [12] consists of processing the input/output data to determine the significant linear and nonlinear equations and associated parameters that are necessary to represent an observed system response. Questions addressed here include the determination of the order of the model (e.g., number of degrees of freedom) and a mathematical form (e.g., polynomial) to represent any nonlinear character in the dynamic equations. For linear dynamic systems, the determination of order is of primary importance. For nonlinear systems the determination of forms to represent nonlinearities has equal importance.

2.1.3.3 Parameter Estimation

The estimation of unknown parameter values follows the determination of a suitable model structure. Numerical values of unknown parameters are determined by choosing them to optimize some performance index which measures how well the mathematical model represents the observed data. Possible performance criteria include:

- minimization of sum of weighted squared fit errors, and
- minimization of the autocorrelation of the fit errors.

Fit error is the difference between the observed response of the dynamic system and the simulated response of the system model to the observed inputs.

The determination of model structures and the estimation of parameter values are often done in parallel. Model structures are determined by fitting several competing candidate models (specified by the user) to the observed system response. The model structure which gives the "best" fit of the data is the structure chosen.

Several criteria may be used to evaluate the closeness of the fit. Simple mean square fit error is not a suitable criterion for the comparison of all candidate models. The use of this criterion will always lead to the choice of highly complex models, since adding degrees of freedom to the model always leads to reduced mean square fit error. More useful criteria, when evaluating candidate models having different numbers of parameters are the mean square prediction error and the F statistic [13]. These criteria weight fit error against the number of free parameters in the model.

A two-stage process has been found to be effective for determining the structure of the model and estimates of the parameters. First, candidate models are evaluated by choosing their parameters to minimize fit error or mean square prediction error. This evaluation is done using a numerical scheme which accurately evaluates the performance index but which may not accurately evaluate the parameter estimates themselves. Once the model structure is established in this way, the parameter estimates are refined using a scheme which gives more accurate parameter estimates. For dynamic systems, a computationally efficient method which is effective for model structure determination is the equation error minimization method. Parameter estimates may subsequently be refined using output error minimization or combined state and parameter estimation methods. All three of these estimation methods are treated in greater detail in Section 2.3.

2.1.3.4 Model Validation

A good criterion for the validation of a model is the use of the model to predict new data. A typical procedure might be to use, say, 80% of the available data to determine the model structure and parameter values. Then the resulting model would be used to predict the remaining 20% of the data. The degree of validation achieved can then be interpreted from the accuracy of the prediction.

Statistical performance criteria include:

- mean square fit error,
- mean square prediction error, and
- whiteness (statistical independence) of residuals.

Whiteness can often be evaluated effectively by visual inspection of plots of the observed data superimposed on plots of the predicted data. Plots of the residuals themselves may also be used.

Finally, validation should include comparison of the model determined from system identification with models available a priori. Aerodynamic models, as determined by identification, should be compared to those determined from theoretical predictions or wind tunnel tests.

2.2 Test Planning

The first step in the integrated system identification procedure is the planning of the tests. A complete system identification test plan should include specifications for both the input signals [10]

2.1.3.1 Flight Data Preprocessing and Analysis

The preprocessing and analysis of flight test data is a major element of the integrated system identification procedure. The overall objectives of this task are to review measurement excitation, remove wildpoints, reconstruct unmeasured quantities (i.e., acceleration of the aircraft at the center of gravity), and develop a set of kinematically consistent measurements. Where measurement consistency cannot be established, requirements for instrumentation system error source modeling are defined. Measurement consistency has a significant impact on parameter identification accuracy since unaccounted-for errors will bias parameter estimates.

2.1.3.2 Model Structure Determination

The model structure determination phase [12] consists of processing the input/output data to determine the significant linear and nonlinear equations and associated parameters that are necessary to represent an observed system response. Questions addressed here include the determination of the order of the model (e.g., number of degrees of freedom) and a mathematical form (e.g., polynomial) to represent any nonlinear character in the dynamic equations. For linear dynamic systems, the determination of order is of primary importance. For nonlinear systems the determination of forms to represent nonlinearities has equal importance.

2.1.3.3 Parameter Estimation

The estimation of unknown parameter values follows the determination of a suitable model structure. Numerical values of unknown parameters are determined by choosing them to optimize some performance index which measures how well the mathematical model represents the observed data. Possible performance criteria include:

- minimization of sum of weighted squared fit errors, and
- minimization of the autocorrelation of the fit errors.

Fit error is the difference between the observed response of the dynamic system and the simulated response of the system model to the observed inputs.

The determination of model structures and the estimation of parameter values are often done in parallel. Model structures are determined by fitting several competing candidate models (specified by the user) to the observed system response. The model structure which gives the "best" fit of the data is the structure chosen.

Several criteria may be used to evalutate the closeness of the fit. Simple mean square fit error is not a suitable criterion for the comparison of all candidate models. The use of this criterion will always lead to the choice of highly complex models, since adding degrees of freedom to the model always leads to reduced mean square fit error. More useful criteria, when evaluating candidate models having different numbers of parameters are the mean square prediction error and the F statistic [13]. These criteria weight fit error against the number of free parameters in the model.

A two-stage process has been found to be effective for determining the structure of the model and estimates of the parameters. First, candidate models are evaluated by choosing their parameters to minimize fit error or mean square prediction error. This evaluation is done using a numerical scheme which accurately evaluates the performance index but which may not accurately evaluate the parameter estimates themselves. Once the model structure is established in this way, the parameter estimates are refined using a scheme which gives more accurate parameter estimates. For dynamic systems, a computationally efficient method which is effective for model structure determination is the equation error minimization method. Parameter estimates may subsequently be refined using output error minimization or combined state and parameter estimation methods. All three of these estimation methods are treated in greater detail in Section 2.3.

2.1.3.4 Model Validation

A good criterion for the validation of a model is the use of the model to predict new data. A typical procedure might be to use, say, 80% of the available data to determine the model structure and parameter values. Then the resulting model would be used to predict the remaining 20% of the data. The degree of validation achieved can then be interpreted from the accuracy of the prediction.

Statistical performance criteria include:

- mean square fit error,
- mean square prediction error, and
- whiteness (statistical independence) of residuals.

Whiteness can often be evaluated effectively by visual inspection of plots of the observed data superimposed on plots of the predicted data. Plots of the residuals themselves may also be used.

Finally, validation should include comparison of the model determined from system identification with models available a priori. Aerodynamic models, as determined by identification, should be compared to those determined from theoretical predictions or wind tunnel tests.

2.2 Test Planning

The first step in the integrated system identification procedure is the planning of the tests. A complete system identification test plan should include specifications for both the input signals [10]

and the instrument system [11]. Analytic methods exist for specifying these two critical items as functions of system identification accuracy requirements.

Design of the system excitation requires some a priori knowledge of the system characteristics as well as data collection constraints such as sampling rate and data length limits. It is also necessary to specify the objectives of the identification. For example, some parameters may be known accurately a priori. The objective, then, may be to identify only those parameters that are poorly known.

Figures 2 and 3 [1] illustrate the importance of test signal design in reducing the uncertainty in the estimates of parameters. Figure 2 shows an elevator input signal for identifying five parameters which model the longitudinal dynamics of the C-8 transport aircraft. The input time history is chosen to minimize the sum of the covariances of the five parameters. Figure 3 compares the standard deviation in parameter estimates for the optimal input and for a doublet input which has the same total energy as the optimal input. As can be seen, a doublet elevator input, commonly used for aircraft flight testing, is not as effective as the optimal input. In executing the test, the suboptimal input shown on Figure 2 results in performance nearly exactly that of the optimal, but is much easier to implement by the pilot.

The other important factor in the initial test plan is specification of the instrumentation. Unfortunately, the instrument systems employed to record data for system identification processing are often not designed with such ends in mind. For example, aircraft autopilot instruments, which are often used to record data during test flights, may not have sufficient accuracy to allow consistent aerodynamic parameter estimation. This is becoming less of a problem due to the use of navigation grade sensors in modern digital flight control system designs. The ring-laser gyro-based strapdown inertial reference unit is a good example of this change.

Sensors are subject to a variety of errors that degrade both state and parameter estimation accuracies. Reference 11 presents an analytical technique to determine the effect of sensor errors on estimation accuracies. Both random (e.g., additive uncorrelated noise in measurements) and systematic (e.g., instrument bias or scale factor errors) are treated. One important conclusion is that systematic errors of relatively small magnitude in comparison with random errors can cause significant parameter estimation bias. If such systematic errors are unavoidable, then parameters modeling them can be added to the set of total parameters to be estimated. This technique can reduce overall parameter estimation uncertainty.

2.3 Data Processing Algorithms

A large number of methods exist for performing system identification data processing. The best algorithm for any given application depends strongly on the type of model and on the nature of the available data. No one type of processing algorithm can handle all possible applications.

This section outlines three processing methods (Table 1) which have been found to be effective in a variety of applications. These methods are:

- equation error minimization methods,
- output error minimization methods, and
- simultaneous state and parameter estimation methods.

2.3.1 Equation Error

The equation error minimization methods estimate unknown parameters by choosing them to minimize a performance index. A continuous dynamic system must be represented as:

$$dx/dt = f(x, u, t, \theta) + w$$

where θ is a set of p unknown parameters and w is a time-varying unobservable disturbance. An analogous formulation exists for a discrete dynamic system. The performance index $\mathcal{J}_e(\theta)$ to be minimized is:

$$\mathcal{J}_e(\theta) = \left\{ \sum_{i=1}^{N} \frac{dx(t_i)}{dt} - f[x(t_i), u(t_i), t_i, \theta)] \right\}^2 \qquad (1)$$

The equation error minimization method is often called the least squares method because of the form of the performance index $\mathcal{J}_e(\theta)$.

The effective use of the equation error minimization requires the a priori determination of system states x, controls u, and state derivatives dx/dt over the time interval of the test. A priori here means that these quantities must be determined before the unknown system parameters are estimated. This determination may be done using direct measurements or using system characteristics which are independent of the parameters. For example, an unmeasured state derivative may be determined by (very carefully) numerically differentiating a measured state time history.

The term w is a stochastic quantity which represents unmeasurable process disturbances in the system. w includes wind gusts and unmodeled, high-order aerodynamic effects.

The special advantage of the equation error minimization method lies in the fact that many nonlinear dynamic system functions $f(x, u, t, \theta)$ are linear in the parameters θ. In other words,

$$f(x, u, t, \theta) = \sum_{j=1}^{p} \theta_j f_j(x, u, t) + f_{p+1}(x, u, t) \tag{2}$$

The functions f_j, $j=1,2,\ldots p+1$ are independent of the p unknown parameters θ_j. The parameter values which minimize $\mathscr{L}_e(\theta)$ can be found explicitly using linear algebraic operations [14]. The disadvantages of the equation error minimization method arise primarily from the requirement for very accurate measurements of states and controls. States will inevitably be measured with some error. No measurement at all may be available for other states.

2.3.2 Output Error

Output error minimization methods, like equation error minimization methods, estimate unknown parameters by choosing them to minimize a performance index. The dynamic system must be represented as

$$dx/dt = f(x, u, t, \theta), \quad x(t_0) = g(\theta) \tag{3}$$

$$y = h(x, u, t, \theta) + \nu \tag{4}$$

where θ is a set of p unknown parameters and ν is a time-varying, unobservable, additive measurement error. The performance index $\mathscr{L}_o(\theta)$ to be minimized is

$$\mathscr{L}_o(\theta) = \sum_{i=1}^{m} [y_i - \hat{y}(t_i, \theta)]^2 \tag{5}$$

Here y_i is the observed system output at time t_i. $\hat{y}(t_i, \theta)$ is the system output y predicted for time t_i by solving the system state equations and measurement equations using the measured system inputs $u(t_i)$ and the a priori parameter values θ.

The effective use of the output error minimization requires the very accurate measurement of system inputs u and the measurement of system outputs y. The method will tolerate errors in the measurement of y.

The term ν is a stochastic quantity which represents instrument measurement errors, e.g., analog-to-digital quantization noise.

The special advantage of the output error minimization method, with respect to the equation error method, is that the measurement requirements are greatly relaxed. The method does not require the accurate measurement of all state and state derivatives. Rather, it is effective using noisy measurements of the limited number of outputs that are available.

The actual determination of the parameter values θ which minimize the performance index $\mathscr{L}_o(\theta)$ is computationally more complex than the minimization $\mathscr{L}_e(\theta)$. This is because $\mathscr{L}_o(\theta)$ is a nonlinear function of the parameter set θ. Finding the minimizing parameter set requires an iterative numerical scheme [15, 16]. The application of such numerical methods is often not straightforward.

The principal disadvantage of the output error minimization scheme is that it does not explicitly allow for the presence of unmodeled disturbances in the state dynamics. Such disturbances are represented in the equation error method by the term w. "Process noise" is the term often used to describe these unmodeled effects.

It should be noted that the output error method can account for system dynamics disturbances of unknown magnitude if the form of these disturbances is accurately represented. The disturbance w must be explicitly represented as

$$w = w(t, \theta) \tag{6}$$

The unknown elements of the disturbance are represented using part of the unknown parameter vector θ. One might estimate the horizontal plane components of a steady wind present during a flight test, for example.

2.3.3 Combined State and Parameter Estimation

Methods which combine state and parameter estimation are required if significant levels of both unknown, unmodeled disturbances and measurement errors are present in the system under study. The performance index used here is very similar to the output error index $\mathscr{L}_o(\theta)$. However, the estimated outputs y are now direct functions of the observed outputs y. The performance index is

$$\mathscr{L}_s(\theta) = \sum_{i=1}^{M} [y_i - \hat{y}(t_i, \theta, y)]^2 \tag{7}$$

The estimated outputs are determined using both the system dynamic equations and the observed values of the outputs themselves.

Methods for the determination of $\hat{y}$ given measurements y and an assumed form of the system dynamics have been widely studied under the topics of state estimation [17] and linear system observation [18]. The use of the Kalman filter to estimate y in the modified output error performance index $\mathscr{L}_s$ leads to "maximum likelihood" parameter estimates [19]. The procedure requires that $\mathscr{L}_s$ be evaluated as a function of θ using the Kalman filter to estimate $\hat{y}$. The parameter values are

estimated by choosing them to minimize $\mathscr{P}_s(\theta)$ using a Gauss-Newton method [20]. The use of this maximum likelihood estimation procedure often allows the estimation of system noise levels as well as of parameters describing the physical plant.

2.4 Some Practical Considerations

This section briefly discusses a number of practical considerations which should be taken into account in an effective system identification data processing scheme.

2.4.1 Assumptions Regarding Measurement Noise Statistics

A common problem is to assume that the measurement error should be modeled as a Gaussian white process when in fact systematic errors such as bias and scale factor exist. Systematic measurement errors will usually cause larger parameter estimation errors than random noise errors of the same root-mean-square level. A very common scale factor found when dealing with any instrument using electronic pickoffs is -1.0. This is due to simple polarity errors made when installing the instrument. Reference 11 covers methods of assessing the relative significance of systematic measurement errors and random measurement errors.

2.4.2 Number of Independent Parameters In The Model

Problems can arise from an attempt to fit too complex a model to the available data. The chief symptom of this is that a large scatter of estimated parameter values will be seen if several data sets are used independently to estimate values for the same parameter set.

2.4.3 Extrapolation of Results

An identified model should not be used to predict system behavior for operating regimes far beyond those encountered during data collection. Operating regime predictions should be limited in both input bandwidth and amplitude to those tested.

2.4.4 Excitation of All System Modes

This problem can be avoided by careful choice of inputs during the test planning stage. A second solution is to process multiple maneuvers simultaneously which contain different control inputs. By doing this, the required modal information is extracted from a set of simpler maneuvers, rather than one complicated maneuver.

2.4.5 Effective Use of Sequential Data Processing Schemes

System identification data processing requires the computational steps of model structure determination, parameter estimation, and model validation. An additional preliminary step of prefiltering measurements may also be required for effective use of an equation error parameter estimation method. An effective overall computational scheme may require that the operations of prefiltering, model structure determination, and parameter estimation be carried out in a sequential rather than in a more nearly simultaneous manner. Care must be taken to ensure that the algorithms employed at any given stage do not remove critical information from the data. As a simple example, the bandpass of a noise prefilter should be higher than that of the modes of the system to be identified.

2.4.6 Process Noise

The term "process noise" refers to unmodeled factors in the state dynamics of the system being identified. Sources of process noise include:

(1) unmeasured environmental disturbances - wind gusts acting on an aircraft, for example,

(2) unmodeled nonlinearities or degrees of freedom in the state dynamics, and

(3) errors in measuring input signals.

The effect of process noise is usually, but not always, to degrade estimation accuracies. If measurements of system states are highly accurate, then the process noise becomes the major source of estimation error. Under some circumstances, process noise in the form of unmeasured environmental disturbances can improve estimation accuracy. The environmental disturbances might excite modes of the system which are not excited by the known input test signal.

The relative significance of process noise in an identification effort depends roughly upon the ratio

$$r = \text{RMS(process noise)}/\text{RMS(known inputs)} \tag{8}$$

where RMS( ) refers to the root-mean-square state excursion due to the indicated source of excitation. If r is large, then the process noise is significant. If r is small, then the process noise is not significant. It is difficult, however, to specify a value of r indicating the boundary between significant or insignificant process noise levels which will be valid for all systems.

Effective system identification methods exist for use when available data contain process noise. The equation error formulation is preferred if all system states can be measured or estimated accurately, otherwise the formulation combining state and parameter estimation will be required.

2.4.7 Initialization

Many parameter estimation formulations require the iterative, numerical solution of nonlinear equations. The output error and the combined state and parameter estimation formulations fall into this category. Iterative numerical algorithms require initial estimates of parameter values in order to begin execution of the first iteration. Inaccurate initial estimates may cause

(1) convergence of the estimation method (which usually employs some form of performance criterion minimization algorithm) to a local minimum, or

(2) divergence of the estimated parameter values as iterations proceed. Divergence may occur, for example, if the values of the initial parameter estimates cause an instability in the dynamic system model.

An effective way to obtain initial parameter estimates for starting iterative algorithms is often to employ the equation error estimation formulation. As noted in Section 2.3.1, the equation error formulation usually requires only the solution of a linear set of algebraic equations in order to obtain parameter estimates. Such equations may be solved without a priori parameter estimates. An effective two-step parameter estimation procedure is

(1) estimate initial parameter values using the equation error formulation, then

(2) refine these estimates using either the output error or the combined state and parameter estimation formulations.

The values of parameters estimated using the equation error formulation are sensitive to errors in measuring states (measurement noise). However, the parameter estimates calculated using an equation error criterion even with data corrupted by measurement noise are often sufficiently accurate for use as start-up values for iterative algorithms.

2.4.8 Numerical Methods

System identification algorithms engender a variety of numerical mathematical requirements. Table 2 lists four of these:

- solution of differential equations,
- solution of linear algebraic systems of equations,
- solution of least squares problems, and
- minimization of general nonlinear multivariable functions.

Effective methods to handle these problems range from the classical systematic elimination method of Gauss [25] for the solution of systems of linear algebraic equations to more recent developments in the solution of linear least squares problems [26].

An important consideration in using any of the methods of Table 2 is that of numerical conditioning. Numerical conditioning refers to the sensitivity of the output of a numerical algorithm to small changes in the input to the algorithm. For example, assume that in solving a system of n linear equations

$$Ax = b,$$

the matrix A is known exactly, but the vector b is subject to uncertainty δb. The norm of the resulting uncertainty in δx, x, is bounded by [25].

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\sqrt{\mu_1}}{\sqrt{\mu_n}} \quad \frac{\|\delta b\|}{\|b\|} \tag{9}$$

where μ_1 is the largest eigenvalue of AA^T and μ_n is the

smallest eigenvalue of AA^T. The quantity

$$\text{cond}(A) = \sqrt{\mu_1/\mu_n} \tag{10}$$

is called the condition number of A and is always greater than 1.0. Similar bounds for solution sensitivities exist for least squares parameter estimation problems [26].

The condition number of a numerical problem can give insight into the precision required to obtain acceptable accuracy of solution. The uncertainty δb, for example, might be due to rounding due to finite precision in computer word length. If the condition number of a problem is 10^6, then eight significant figures of accuracy would be required to maintain 1% accuracy in the solution.

3. RECENT TECHNIQUES FOR THE NONLINEAR AERODYNAMICS PROBLEM

Methods for the maximum likelihood identification of linear state dynamic models are well established [29]. Such methods have been applied to problems of linear modeling of aircraft aerodynamics using flight test [22] data. These methods are sometimes applied in a piecewise manner to fundamentally nonlinear systems.

The intrinsic nonlinear nature of aircraft aerodynamic models may inhibit the effective use of linear identification methods. For example, if a limited amount of data is available, it may not be possible to identify many linear perturbation models. A single nonlinear model may have fewer total free parameters. Also, excursions through the nonlinear portion of the model's dynamic range may be so rapid that no single linearized model can represent a significant portion of the trajectory.

The use of a nonlinear model may be required if the goal of the analysis is to determine which one of several competitive phenomenological mathematical models best fits available data. A "phenomenological" model is one that is constructed from fundamental physical principles. Such a model may have a very complex form mathematically but may have a minimum number of unknown coefficients.

There are certain computational difficulties associated with the use of a nonlinear dynamic model in a maximum likelihood parameter estimation algorithm.

(1) Calculation of Sensitivities. The estimation of parameters through the use of the maximum likelihood criterion requires the maximization of the likelihood of the data with respect to the unknown parameters. The determination of the maximizing parameter values requires numerical optimization techniques. The most efficient of these [27] are descendents of the Levenberg-Marquardt nonlinear least squares method [16,30]. These algorithms require the evaluation of the partial derivative of modeling residuals with respect to parameter values. This partial derivative is often called a "sensitivity". The calculation of these sensitivities is not difficult in principle. They satisfy differential equations which are closely related to the system dynamic equations, but which contain terms based on the algebraic partial derivatives of the dynamic equations (see Section 3.1). The difficulty is one of practice. Any time that the structure of the nonlinear model is changed, then the sensitivity differential equations must be changed also. This requires tedious algebraic differentiation of the modified dynamic equations.

(2) Evaluation of Covariance of Parameter Estimates. A parameter covariance matrix can be estimated using the Cramer-Rao bound [31]. The most common use of this bound assumes that the errors in predicting the response of the system are due to an additive, white (negligible autocorrelation) random process. If the analyst also desires confidence intervals for parameter estimates, then the additional assumption that the errors have a normal distribution must also be made. These assumptions are commonly violated when a nonlinear system is modeled. In particular, the whiteness assumption is typically violated.

(3) Generic Model Structure. There is a need to represent nonlinear functions of several variables in the model used in the identification algorithm. Ideally, a single generic form should represent multidimensional surfaces of arbitrary shape. These functions represent total aerodynamic force or moment coefficients as functions of angle of attack, angle of sideslip, and angular rates.

3.1 Problem Definition

The dynamic system is modeled as nx first order nonlinear differential equations.

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{w}, t, \underline{\theta}) \tag{11}$$

having an output measured at discrete times t_k

$$y(t_k) = h(\underline{x}, \underline{u}, t_k, \underline{\theta}) + v(t_k) \tag{12}$$

Here

$\underline{x}$ = nx component state

$\underline{u}$ = nu component inputs measured without error

$\underline{\theta}$ = nth component unknown parameter vector

$\underline{w}$ is a nw component random input (process noise source) having statistics

$$E(\underline{w}) = \underline{0} \tag{13}$$

$$E[\underline{w}(t_i)\,\underline{w}^T(t_j)] = Q(t_i)\,\delta_{ij} \tag{14}$$

The scalar $v(t_k) = v_k$ is a random measurement error having the statistics

$$E(v_k) = 0 \tag{15}$$

$$E(v_k^2) = r \tag{16}$$

Note that the assumption of scalar measurements does not cause a great loss of generality. This formulation can accomodate multiple sensors simply by assuming that the interval between measurements is

sometimes very small. The only loss of generality regards the representation of correlation of measurement error between sensors.

If we assume that the stochastic quantities have normal distributions then the joint density or likelihood function of a sample of nt measurements $y(t_k) = y_k$ is

$$\mathscr{P}(y_1, y_2, \ldots, y_{nt}; \underline{\theta})$$

$$\mathscr{L} = \left\{ \prod_{k=1}^{nt} \frac{\exp\left[-\frac{1}{2}\left(\hat{y}_k - y_k(\underline{\theta})\right)^2 / \sigma_k^2(\underline{\theta})\right]}{(2\Pi)^{1/2}\ \sigma_k(\underline{\theta})} \right\} \tag{17}$$

An extended Kalman filter [17] can generate both the measurement estimates y_K and the measurement uncertainties σ_K. The estimate of θ having the smallest variance is the one which maximizes $\mathscr{L}(y; \underline{\theta})$ with respect to θ. The maximization of $\mathscr{L}$ is equivalent to the minimization of the negative log likelihood function given by

$$-\log \mathscr{L}(\underline{y}; \underline{\theta}) = \tag{18}$$

$$\frac{1}{2} \sum_{K=1}^{nt} \left\{ \left[y_k - y(\underline{\theta})\right]^2 / \sigma_k^2(\underline{\theta}) + 2 \log \sigma_k(\underline{\theta}) \right\}$$

Note that if the σ_k are known, then the minimization of $-\log\mathscr{L}$ can be treated as a nonlinear least square problem.

Aspects of this problem addressed here are the following.

(1) The minimization of Eq. (18) with respect to $\underline{\theta}$ requires the use of an iterative numerical procedure similar to a Newton or quasi-Newton method. If the σ_k are known, then the most effective procedure is that of Levenberg and Marquardt [16,30]. A drawback to the Levenberg-Marquardt method is the requirement for the evaluation of $\partial\hat{y}/\partial\underline{\theta}$. Direct analog finite difference methods [32] avoid this problem by approximating the partial derivative by a finite difference. Section 3.2 extends the finite difference analog of the Levenberg-Marquardt procedure to the more general form of Eq. (18).

(2) Validation of an identified model should include the determination of confidence intervals or variances for estimated parameters. These may be estimated using the Cramer-Rao bound, which states that

$$E[(\hat{\theta} - \underline{\theta}^*)(\hat{\theta} - \underline{\theta}^*)^T] \geq M^{-1} \tag{19}$$

where M is the information matrix, given by

$$[M]_{ij} = -\left[\partial^2(\log \mathscr{L}(\underline{y}; \underline{\theta})/\partial\theta_i\ \partial\theta_j\right] \tag{20}$$

and $\underline{\theta}^*$ is the true parameter value.

Experienced analysts in the system identification field are aware that the Cramer-Rao bound is usually optimistic [33]. That is, it tends to predict parameter variances which are very small in comparison to variances observed among estimates derived from multiple data sets. The most easily implemented expressions for the information matrix assume that the measurement errors are not autocorrelated. If autocorrelation is accounted for, then the Cramer-Rao bound more realistically approximates the true parameter variance. Section 3.3 outlines methods based on principles of generalized least squares which automatically account for first order autocorrelation of measurements. This method produces parameter estimates which have a lower variance than those which do not account for autocorrelation.

(3) Modeling nonlinear aerodynamics requires the representation of nonlinear functions of several variables. For example, pitch moment of an aircraft is a nonlinear function of angle of attack α, angle of sideslip β, pitch rate Q, and elevator angle δ_e.

$$C_m = C_m(\alpha, \beta, Q, \delta_e) \tag{21}$$

Much work in the identification of these functions has been based on their representation as multidimensional polynomials [34]. This approach is effective for local models. A local model is one that is valid over a restricted region of the flight envelope, say for an α interval of 10^0. Such models often use expansions for functions like C_m in the states of degree no higher than two. The representation of a global model using polynomial expansions may require very high order polynomials. The representation of a pitch moment curve for one particular aircraft through a 40^0 angle of attack region requires a ninth degree polynomial [35]. Section 3.4 indicates how the use of local, low degree polynomial models leads very naturally to a spline formulation for a global model.

3.2 Derivative Free Minimization of the Negative Log Likelihood Function

Finding the parameter values $\hat{\underline{\theta}}$ which minimize Eq. (18) requires the use of a numerical optimization scheme. Typically, each iteration of the algorithm updates $\underline{\theta}$ as

$$\underline{\theta}_{n+1} = \underline{\theta}_n + \Delta\underline{\theta} \tag{22}$$

where $\Delta\underline{\theta}$ satisfies

$$M\Delta\underline{\theta} = -\underline{g} \tag{23}$$

$$g_i = \partial \log \mathcal{P}(\underline{y}, \underline{\theta}) / \partial\theta_i \tag{24}$$

and M is defined in Eq. (20).

Differentiating Eq. (18) gives [35]

$$g_i = -\sum_{k=1}^{nt} \left\{ [\nu_k(\partial\nu_k/\partial\theta_i) - \frac{1}{2}\nu_k^2(\partial\sigma_k^2/\partial\theta_i)/\sigma_k^2 + \frac{1}{2}\sigma_k^2/\partial\theta_i]/\sigma_k^2 \right\} \tag{25}$$

$$M_{ij} = -\sum_{k=1}^{nt} \left\{ (\partial\nu_k/\partial\theta_i)(\partial\nu_k/\partial\theta_j)/\sigma_k^2 + [-(\partial\nu_k/\partial\theta_i)(\partial\sigma_k^2/\partial\theta_j) \right. \tag{26}$$

$$-(\partial\nu_k/\partial\theta_j)(\partial\sigma_k^2/\partial\theta_i) + \nu_k(\partial\sigma_k^2/\partial\theta_i)(\partial\sigma_k^2/\partial\theta_j)$$

$$+\sigma_k(\partial^2\nu_k/\partial\theta_i\partial\theta_j) - \nu_k(\partial^2\sigma_k/\partial\theta_i\partial\theta_j)]\nu_k/\sigma_k^4$$

$$\left. -\frac{1}{2}(\partial\sigma_k^2/\partial\theta_i)(\partial\sigma_k^2/\partial\theta_j)/\sigma_k^4 + \frac{1}{2}(\partial^2\sigma_k^2/\partial\theta_i\partial\theta_j)/\sigma_k^2 \right\}$$

where $$\nu_k = y_k - \hat{y}_k(\underline{\theta}). \tag{27}$$

Following the spirit of the Levenberg-Marquardt method, we simplify the expression for the Hessian by dropping the terms proportional to ν_k. If the model fits the data well, ν_k should approach zero near convergence. With the exception of the last term of Eq. (26), it is now possible to evaluate both gradient and Hessian if only the first-order sensitivities ν_k and σ_k^2 to changes in θ are known.

It is possible to derive analytically ordinary differential equations for $\partial\nu_k/\partial\theta_i$ and $\partial\sigma_k^2/\partial\theta_i$. When the plant dynamic equations are linear, these differential sensitivity equations (also called "sensitivity equations") have a particularly simple form [36]. When the plant dynamics are nonlinear, however, a much more practical method is to approximate the partial derivatives with finite differences.

$$\partial\nu_k/\partial\theta_i \doteq [\nu_k(\underline{\theta} + \underline{\epsilon}_i) - \nu_k(\underline{\theta})] / |\underline{\epsilon}_i| \tag{28}$$

or

$$\partial\nu_k/\partial\theta_i \doteq [\nu_k(\underline{\theta} + \underline{\epsilon}_i) - \nu_k(\theta - \epsilon_i)] / 2|\underline{\epsilon}_i| \tag{29}$$

where $(\underline{\epsilon}_i)_j = \epsilon\delta_{ij}$.

Partial derivatives of $\sigma^2(\theta)$ are approximated similarly.

The "direct analog" [32] type of optimization algorithms use the finite difference approximations to the partial derivatives as direct substitutes for the partial derivatives. These algorithms have been studied carefully for the solution of the nonlinear least square problem and have been found to have convergence properties nearly identical to algorithms which use analytic expressions for the sensitivities.

Alternate methods for minimization of the negative log likelihood function without evaluating derivatives use algorithms which are not direct analogs of derivative methods. We have tested one of these methods [37] in a nonlinear system identification algorithm and found it to be not as effective as the direct analog method.

In practical implementations of the finite difference method for the solution of system identification problems, we typically simultaneously solve a set of nx nominal state equations together with nth sets of nx perturbed parameter state equations. This is required in order to implement the one-sided approximation to the sensitivity given in Eq. (28).

The last term of Eq. (26) cannot be eliminated by assuming that ν_k is small near the minimum. The term cannot be constructed from first order partial derivatives of ν and σ^2. It can be

estimated, however, using methods similar to those employed in solving large residual nonlinear least square problems [38]. This topic will not be treated in further detail here.

3.3 Autocorrelated Measurement Errors

It is desireable to determine the expected accuracy of the parameters estimated from flight data. The usual method for doing this is to compute a parameter covariance matrix as the inverse of the Fisher information matrix M.

Methods of generalized least squares [39] indicate expressions for parameter estimation covariance using the assumption that measurement errors are autocorrelated. This autocorrelated process is the output of a first order difference equation driven by white noise. Not only parameter covariances but also parameter estimates themselves are altered by the autocorrelation assumption. An estimation algorithm which does not explicitly account for the measurement error autocorrelation will still produce unbiased parameter estimates. However the actual variance of such estimates, as opposed to the Cramer-Rao predicted variance, will be higher than those produced by an algorithm which does explicitly account for autocorrelation.

We consider here only the output error case (no process noise) of the maximum likelihood estimator for dynamic systems. The estimated measurements $\hat{y}$ are functions of the unknown parameter set $\underline{\theta}$. The information matrix M is given by

$$M \triangleq \frac{1}{\sigma^2} J^T J \tag{30}$$

where J is the matrix of sensitivities

$$[J]_{k\ell} = -\partial \hat{y}(\underline{\theta}, t_k)/\partial \theta_\ell \tag{31}$$

for the case of white measurement noise. Iterations of the identification algorithm solve

$$M \Delta \underline{\theta} = -\underline{g} \tag{32}$$

where

$$\underline{g} = \frac{1}{\sigma^2} J^T \underline{\nu} \tag{33}$$

$$\nu_k = y(t_k) - \hat{y}(t_k)$$

The covariance of the measurement error is a diagonal matrix

$$E(\underline{\nu}\,\underline{\nu}^T) = \sigma^2 I \tag{34}$$

where $\nu_k = \nu(t_k)$, the measurement error at t_k. (35)

Now suppose that the measurement errors have a nondiagonal covariance matrix of

$$E[\underline{\nu}\,\underline{\nu}^T] = \sigma^2 V \tag{36}$$

For a purely linear estimation problem, (i.e. $y = J\underline{\theta} + \underline{\nu}$), the "generalized least squares" estimate of $\underline{\theta}$ satisfies [39]

$$[J^T V^{-1} J]\ \underline{\theta} = J^T V^{-1} \underline{y} \tag{37}$$

The covariance of the parameter estimates is

$$E\,[(\hat{\underline{\theta}} - \underline{\theta}^*)(\hat{\underline{\theta}} - \underline{\theta}^*)^T] = \sigma^2 (J^T V^{-1} J)^{-1} = \mathrm{Var}(\hat{\theta}) \tag{38}$$

Any other linear, unbiased estimator has a covariance matrix which exceeds that given in Eq. (38).

If the noise vector $\underline{\nu}$ is generated by a first order autoregressive process

$$\nu(t_k) = \rho \nu(t_{k-1}) + \epsilon_k \tag{39}$$

where ϵ_k is a zero mean, constant variance process, then V has the form

$$V = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{n-1} \\ \rho & 1 & \rho & & \rho^{n-2} \\ \rho^2 & \rho & 1 & & \rho^{n-3} \\ \vdots & & & & \vdots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \cdots & 1 \end{bmatrix} \tag{40}$$

The inverse of V is $V^{-1} = P^T P$ where

$$P = \begin{bmatrix} \sqrt{1-\rho^2} & 0 & \cdots & 0 \\ -\rho & 1 & & 0 \\ 0 & -\rho & & 0 \\ \vdots & & 1 & 0 \\ 0 & \cdots & -\rho & 1 \end{bmatrix} \frac{1}{\sqrt{1-\rho^2}} \tag{41}$$

The generalized least square estimator for $\underline{\theta}$ can be easily implemented by writing Eq. (37) as

$$[(P\,J)^T(P\,J)]\ \hat{\underline{\theta}} = (P\,J)^T P\,\underline{y} \tag{42}$$

The multiplications $P\,J$ and $P\,\underline{y}$ are simple because $\underline{P}$ is sparce.

For application to the output error system identification problem, a nonlinear least square problem, Eq. (42) is applied to parameter variations $\Delta\underline{\theta}$ on each iteration of a successive approximation algorithm.

$$(P\,J)^T(P\,J)\Delta\underline{\theta} = (P\,J)^T P\,[\underline{y} - \hat{\underline{y}}(\theta)] \tag{43}$$

If ρ is unknown, then it can be estimated using

$$\hat{\rho} \simeq \frac{\sum_{k=1}^{nt-1} (t_k)\ (t_{k+1})}{\sum_{k=1}^{nt} \nu^2(t_k)} \cdot \frac{nt-1}{nt-nth} \tag{44}$$

Each iteration calculates a $\Delta\underline{\theta}$ value using $\underline{P}$ evaluated by Eq. (44), with $\nu(t_k)$ calculated from $\underline{\theta}$ at the end of the previous iteration.

If the measurement noise, σ^2, is unknown, it can be estimated using

$$\hat{\sigma}^2 = \frac{1}{nt-nth}\,(P\nu)^T(P\nu)\ . \tag{45}$$

The covariance of the parameter estimates is

$$Var(\hat{\theta}) = \frac{1}{nt-nth}\,(\bar{P}\nu)^T(\bar{P}\nu)[(PJ)^T(\bar{P}J)]^{-1} \tag{46}$$

where

$$\bar{P} = P\sqrt{1-\rho^2}\ . \tag{47}$$

3.4 Spline Model Structure

The determination of a nonlinear, quasistatic aerodynamic (or hydrodynamic) model requires definition of a coefficient function having a general form

$$C = C(\alpha, \beta, \underline{\omega}', \underline{\delta}, R_e, F_r, M) \tag{48}$$

where α and β are relative flow angles, $\underline{\omega}'$ is a dimensionless angular rate vector, $\underline{\delta}$ is a control vector, and R_e, F_r, and M are the dimensionless numbers of Reynolds, Froude, and Mach. The model structure determination problem for identification of aerodynamic models usually refers to the problem of determining a mathematical form for this multivariable function.

Spline functions are effective ways to represent these coefficient functions. A one dimensional spline function is a piecewise polynomial function having certain continuity conditions between pieces. Figure 4 illustrates a one dimensional cubic spline. $C(\alpha)$ might represent pitch moment as a function of angle of attack. $C(\alpha)$ here is a cubic polynomial on each of the three regions indicated. The function is everywhere continuous and has continuous first and second derivatives. The points α_1, α_2, α_3, α_4 are called the knots of the spline.

The spline function has several properties which make it an effective interpolating function [40].

(1) The spline in Figure 4, for example, is uniquely determined once the values of the function at the four knots are known and certain end conditions are specified.

(2) The shape of the interpolating function is not overly sensitive to the function values at the knots. Small changes in these values do not cause overly large changes in interpolated function values between knots.

(3) The interpolating function $C(\alpha)$ has an optimal smoothness property. It is the unique function which interpolates the specific values at the knots, has the continuity conditions listed above, and has the minimum mean square curvature.

Spline function representation of nonlinear aerodynamic or hydrodynamic coefficient functions may be readily identified using either maximum likelihood, equation error or output error techniques. The analyst must specify the number of knots and their locations. The parameters to be identified are then the function values at the knots. The identification of the $C(\alpha)$ curve in Figure 4 would require the estimation of four parameters.

In Figure 4, the coefficients C_1 - C_4 are the function values at the knot locations α_1 - α_4. These coefficients are the parameters which will be estimated by the identification algorithm. The piecewise cubic polynomials K_{A1} - K_{A4} provide cubic interpolation of C_1 - C_4 for α in the range $[\alpha_1, \alpha_4]$ and linear extrapolation for α outside this range.

Each of the piecewise cubic polynomials is defined over the entire range of $[-\infty < \alpha < +\infty]$. The function $C(\alpha)$ is a linear combination of the K_{Ai} basis functions. The definitions of K_{Ai} are such that

$$K_{Ai}(\alpha_j) = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$$

This makes the coefficients in the linear combination equal to the C_i values. The array in Figure 4 defines the K_{Ai} functions over the five α regions.

The identification of spline functions is most effective when used with derivative free methods to minimize the negative log likelihood function. Such methods do not require the explicit calculation of the sensitivity of the spline function to changes in the parameters which define the spline. The only requirement is for the evaluation of the spline coefficients given function values at the knots, and for the evaluation of the function at intermediate points given the spline coefficients. Each iteration of the "direct analog" method requires the evaluation of the innovations (ε_k) for nominal and for perturbed parameter values.

Methods exist for the use of multidimensional spline functions to represent smooth surfaces [41]. Intermediate methods are also useful. An intermediate method represents variation of a function in one dimension with a spline function and representation in other dimensions with other types of functions, such as low order polynomials.

4. EXAMPLE APPLICATION

The results presented in this paper demonstrate the operational status of nonlinear system identification data processing techniques. Aerodynamic, installed thrust, and flight test instrumentation calibration models are identified for the F-4S from flight maneuvers which encompassed a large range in angle of attack, sideslip, airspeed, control inputs, and body rotation rates. The capability for identifying nonlinear aerodynamic models in a format compatible with preflight predictions is demonstrated. A methodology for determining the accuracy of the parameters estimates is presented.

4.1 F-4S Flight Test Program Overview

The identification results presented in this section are based on a flight test program planned specifically for the collection of data for identification. The sole objective of the flight test program was to generate test data which could be used to develop and validate nonlinear system identification analysis techniques. The Naval Air Test Center (NATC) F-4S aircraft (Buno 286), an F-4J which has been modified to include a maneuvering flap/slat system, was the test aircraft.

The airborne data acquisition systems used for the flight test program included a 3-axis rate gyro, a vertical gyro, a directional gyro, engine RPM and fuel flow measurements, a 3-axis linear accelerometer, control surface deflection measurements and a test airdata system. The airdata system uses a

noseboom which has a pitot-static head for the measurement of impact pressure, static pressure and temperature. The noseboom also has vanes for measurements of angle-of-attack and sideslip.

The flight test program was conducted in a clean configuration with the throttles fixed for each maneuver and with flight near and beyond stall. Test conditions were generally initiated from wing-level, constant altitude flight with $\alpha = 10^{\circ}$ and $M \approx .6$. For some test conditions, the pilot applied variable aft stick to maneuver the aircraft through the desired angle-of-attack test range. For some of the stall entry conditions lateral stick and/or pedal doublets were combined with the longitudinal stick command. Other maneuvers included single axis and multi-axis sequenced doublet inputs. These inputs were made by the pilot and were not intended to be repeatable nor precise with regard to their spectral characteristics, but were generally effective in creating large-amplitude motions. The overall test goal was to force the aircraft through a broad range of test conditions. The level of excitation of primary test variables achieved during the F-4S flight test program is summarized as follows:

- Angle of attack: $-1^{\circ} < \alpha < 40^{\circ}$
- Sideslip: $|\beta| < 18^{\circ}$
- Mach No: $M < .6$
- Rotational Rates: $|P| < 90^{\circ}/S$, $|Q| < 20^{\circ}/S$, $|R| < 25^{\circ}/S$
- Full amplitude control inputs

4.2 Modeling Approach

The approach selected for modeling nonlinear aerodynamic characteristics produces system identification results that can be used to validate preflight estimated aerodynamic models. The models are used for flight simulators and for making predictions of aircraft performance, stability and control characteristics. These aerodynamic models must account for the effect of a number of flight condition and aircraft configuration variables. The "art" in formulating the models is to represent the total aerodynamic coefficient by an incremental buildup, with each increment described by one or two independent variables. This process is illustrated by the following example for the rolling moment coefficient equation.

(1) Select the independent variables:

$$C_{\ell} = f(\alpha, \beta, P, R, \delta_R, \delta_A)$$

i.e., rolling moment coefficient is a function of angle of attack, sideslip, roll and yaw rate, rudder and aileron position.

(2) Partition independent variables into reasonable groups:

$$C_{\ell} = \Delta C_{\ell_{SIDESLIP}} + \Delta C_{\ell_{DYNAMIC}} + \Delta C_{\ell_{RUDDER}} + \Delta C_{\ell_{AILERON}}$$

(3) Select functional relationships for each group:

$$\Delta C_{\ell_{SIDESLIP}} = f(\alpha, \beta)$$

$$= C_{\ell_{\beta}}(\alpha) \cdot \beta$$

$$\Delta C_{\ell_{DYNAMIC}} = f(\alpha, P) + f(\alpha, R)$$

$$= C_{\ell_P}(\alpha)(Pb_w/2V_T) + C_{\ell_R}(\alpha)(Rb_w/2V_T)$$

$$\Delta C_{\ell_{RUDDER}} = f(\alpha, \delta_R)$$

$$= C_{\ell_{\delta_R}}(\alpha) \cdot \delta_R$$

$$\Delta C_{\ell_{AILERON}} = f(\alpha, \delta_A)$$

$$= C_{\ell_{\delta_A}}(\alpha) \cdot \delta_A$$

For this model formulation, each of the stability derivatives, (i.e., C_{ℓ_β}) is modeled as a nonlinear function of angle of attack. By using a cubic interpolation spline, as described in Section 3.4, the parameter identification algorithm solves for C_{ℓ_β} at specific values of angle of attack (i.e., the knots of the spline). This procedure is illustrated in Figure 5 which shows the identified variation of C_{ℓ_β} with angle of attack. For the F-4S parameter identification study, C_{ℓ_β} was identified for α = 5°, 15°, 25°, and 35°. The lower four parts of Figure 4 illustrate the variation of the interpolation splines with angle of attack.

Because these interpolation splines are scaled by the appropriate value of C_{ℓ_β} (i.e., the α = 5° spline is scaled by the value of C_{ℓ_β} for α = 5°), the summation of the four interpolation splines defines the value of C_{ℓ_β} for any value of angle of attack. It should also be noted that each interpolation spline has the value of C_{ℓ_β} when α equals its knot value, and it is zero for other knot values.

The spline formulation is suitable also for representing installed propulsion system performance models and test instrumentation calibration factors. As demonstrated in the next section, more complicated models can be represented by a bicubic spline formulation.

4.3 System Identification Results

The results presented in this section demonstrate the capability for identifying flight test instrumentation calibration factors, performance data including a model for net thrust, and stability and control characteristics from a common set of flight test data.

The aerodynamic models are generally represented by a cubic spline. The aerodynamic parameters are identified at four specific values of angle of attack (i.e., α = 5°, 15°, 25°, 35°). The aerodynamic estimates are expected to be most accurate in the $10° < \alpha < 30°$ range due to the density of test data in this range (see Figure 6 which crossplots α vs β for every quarter of a second for the six maneuvers analyzed).

A common format is used to present the identified models. The estimates are defined as a solid line. When cubic splines are used to represent the parameter, the solid line represents the cubic spline interpolation between each of the four knots. The identified parameters are noted with a solid circle symbol. The parameter estimation uncertainty is shown by dashed lines on either side of the estimate. The distance between the solid line and the dashed lines represents the 2σ uncertainty of the estimate. Preflight predictions are illustrated by a solid triangle symbol.

The validity of the parameter estimates can be established from three different considerations.

(1) Engineering judgement: are the estimates reasonable from the point of view of general agreement with preflight predictions?

(2) Estimation uncertainty: what is the magnitude of the + 2σ bands about the estimate?

(3) Prediction accuracy: How well does the identified model predict flight test measurements for many test maneuvers? Does the identified model predict these measurements better than a model based on preflight parameters?

4.3.1 Flight Test Sensor Calibration

Figure 7 summarizes the types of error terms considered for the accelerometers, rate gyros, vertical gyros, α/β vanes, and the pitot-static impact pressure measurements. These errors represent system level errors in that the contributions from separate sources (i.e., PCM calibration, sensor installation and sensor performance) generally cannot be identified. For example an identified scale factor error for an alpha vane could be due to upwash and/or PCM calibration errors. On the other hand, initial errors for the vertical gyro, which vary with test condition, are probably due to verticality errors resulting from the erection circuit. The identified instrumentation calibration models for the NATC F-4S are presented in Figure 8. All instruments required some correction and the roll rate gyro signal was found to be unusable. As a result, a roll rate signal had to be reconstructed from vertical and directional gyro measurements.

4.3.2 Aerodynamic and Thrust Models

Figures 9 through 12 present F-4S parameter identification results that illustrate the extraction of parameters for both thrust and aerodynamic models from a common set of flight test data.

Figure 9 illustrates the identified model for net thrust and the drag polar which has been reconstructed from models of C_D and C_L as a function of angle of attack. The model for net thrust was defined in terms of a linear relationship betweeen corrected net thrust and corrected fuel flow. The difference between the identified and preflight models for corrected net thrust is a bias in fuel flow of 58 lbs/hrs based on a pressure altitude of h_p = 30,000 ft. The accuracy of the flight test fuel flow sensor is 150 lbs/hr. The identified drag polar matches the preflight polar shape in terms of $C_{D_{MIN}}$, $(C_L/C_D)_{MAX}$ and $C_{L_{MAX}}$.

Figure 10 presents stabilizer, rudder and lateral control power estimates as functions of angle of attack. The flight derived control power estimates show good agreement with preflight data in terms of trends with angle of attack with slight differences in the actual value of the derivative. The uncertainty of the lateral control derivatives ($C_{\ell_{\delta_A}}$ and $C_{n_{\delta_A}}$) as shown by the fanning of the $\pm 2\sigma$ curves is due to the lack of lateral control excitation at low and high α. In general, when the uncertainty boundaries are large for some parts of the flight regime, this information can be used for planning additional flight test conditions.

Figure 11 presents the identified variation of yawing moment coefficient (C_n) with angle of attack and sideslip. Values for C_n and C_{n_β} are obtained from the C_n model at $\beta = 0^o$. C_n was modeled by a bicubic spline with specific values identified for $\alpha = 5^o$, 15^o, 25^o, 35^o and $\beta = 6^o$, 12^o and 18^o ($C_n = 0$ for $= 0$). A bicubic spline was selected for the C_n model due to the expected nonlinear variation with angle of attack and sideslip.

Identified values for roll rate and yaw rate stability derivatives for the F-4S are presented in Figure 12. Good agreement is evident between preflight prediction and identified values for C_{n_p} and C_{ℓ_p} (for $\alpha < 30$). The identified value for C_{n_r} is much greater than the preflight prediction, although both are independent of angle of attack. Significant differences exist between identified and predicted values of C_{ℓ_r} for $\alpha > 20^o$.

4.3.3 Flight Measurement Predictions

The ability of the identified aerodynamic and net thrust models to reproduce flight test measurements for the six test maneuvers is tabulated in Figure 13. Figure 13 summarizes the root mean square residuals between the flight test measurements and estimates of the measurements. The predicted measurements are generated from a simulation of the F-4S based on the identified aerodynamic, thrust, and instrumentation models. The simulation is run with actual flight test time histories of the control commands.

To gain some feeling for the relative worth of the identified model, the F-4S simulation was configured with the preflight aerodynamic data and run with the flight test control inputs. The resulting measurement residuals are also presented in Figure 13. In all cases, the identified model provides a better explanation of the flight test measurements. Figure 14 compares the time histories of the flight test measurements, predictions based on the identified model and predictions based on the preflight model for one of the flight conditions analyzed.

5. SUMMARY

(1) The integrated system identification technology encompasses test planning and on-line data consistency testing, as well as data processing.

(2) Effective system identification data processing often requires independent model structure determination, parameter estimation, and model validation steps.

(3) When working with nonlinear systems, the sensitivity functions needed in order to apply iterative estimation algorithms can most effectively be calculated using finite difference approximations.

(4) Autocorrelation of measurement errors significantly effects parameter estimation covariance.

(5) Spline functions are useful for the representation of nonlinear functions during the identification process.

(6) Nonlinear system identification data processing techniques can be used to identify aerodynamic, propulsion system and instrumentation calibration models from a common set of flight test conditions.

(7) A single aerodynamic/net thrust math model is identified for the F-4S from six different flight conditions. The identified mathematical model predicts the sensor measurements for the six flight conditions which encompass a large range in angle of attack, sideslip, airspeed, control inputs and body rotation rates. Aircraft response predictions are improved with the identified model relative to prediction based on a preflight model.

(8) Extraction of performance, stability and control, and high angle of attack characteristics from a single model has been illustrated.

(9) The capability for identifying nonlinear aerodynamic models in a format compatible with preflight predictions has been demonstrated.

(10) Dynamic test techniques, which require nonlinear system identification data processing techniques, can improve test productivity. The results presented in this paper are based on less than 300 seconds of flight test time.

REFERENCES

1. Hall,Jr., W.E., "System Identification-An Overview," Naval Research Reviews, Vol. 30, No. 4 April 1977, pp 1-20.

2. Gustavsson, I., "Survey of Applications of Identification in Chemical and Physical Processes," Proceedings of the 3rd IFAC Symposium, The Hague/Delft, The Netherlands, 12-15 June, 1973, pp 67-85.

3. Baeyens, R., Jacquet, B., "Applications of Identification Methods in Power Generation and Distribution," Proceedings of the 3rd IFAC Symposium, The Hague/Delft, The Netherlands, 12-15 June 1973, pp 1107-1121.

4. De Hoff, R.L., "Identification of a STOL Propulsion Plant Model," J. of Guidance and Control, Vol. 2, No. 3, May-June 1979.

5. Bekkay, G.A., Benkan, J.E.W., "Identification of Biological Systems: A Survey," Automatica, Vol. 14, No. 1, Jan. 1978, pp 41-47.

6. Chow, G.C., "Identification and Estimation in Econometric Systems: A Survey," IEEE Trans. on Automatic Control, Vol. AC-19, No. 6, Dec. 1974, pp 855-861.

7. Gersch, W., and Foutch, D.A., "Least Squares Estimates of Structural System Parameters Using Covariance Function Data," IEEE Trans. on Automatic Control, Vol. AC-19, No. 6, Dec. 1974, pp 898-903.

8. Willsky, A.A., "A Survey of Design Methods for Failure Detection in Dynamic Systems," Automatica, Vol. 12, 1976, pp 601-610.

9. Gupta, N.K., and Hall, Jr., W.E., "Methods for the Real Time Identification of Vehicle Parameters," Technical Report No. 4 under Office of Naval Research Contract N00014-72-C-0328, Feb. 1976.

10. Gupta, N.K., Hall, Jr., W.E., "Input Design for Identification of Aircraft Stability and Control Derivatives," NASA Contractor Report CR-2493, Feb. 1975.

11. Gupta, N.K., Hall, Jr., W.E., "Design and Evaluation of Sensor Systems for State and Parameter Estimation," J. Guidance and Control, Vol. 1, No. 6, Nov. - Dec., 1978, pp 397-403.

12. Gupta, N.K., Hall, Jr., W.E., Trankle, T.L., "Advanced Methods of Model Structure Determination from Test Data," J. Guidance and Control, Vol. 1, No. 3 May-June 1978, pp 197, 204.

13. Allen, D.M., "Mean Square Error of Prediction as a Criterion for Selecting Variables," Technometrics, Vol. 13, No. 3, Aug. 1971, pp 469, 475.

14. Lawson, C.L., and Hanson, R.J., Solving Least Square Problems, Prentice-Hall, 1974.

15. Gupta, N.K. and Mehra, R.K., "Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations," IEEE Trans. on Automatic Control, Vol. AC-10, No. 6, Dec. 1974, pp 774, 783.

16. Marquardt, D.W., "An Algorithm for Least Squares Estimation of Nonlinear Parameters," J. Soc. Indust. Appl. Math., Vol. 11, No. 2, 1963, pp 431-441.

17. Kalman, R.E., Bucy, R., "New Results in Linear Filtering and Prediction," Trans. ASME, Vol. 83D, 1961, p. 95.

18. Luenberger, D.G., "Observing the State of a Linear System," IEEE Trans. Military Electronics, Vol. MIL-8, 1964, pp 74-80.

19. Eyknoff, P., System Identification-Parameter and State Estimation, Wiley, 1974.

20. Rault, A., "Identification Applications to Aeronautics," Identification and System Parameter Estimation, American Elsevier Publishing Company, New York, 1973, pp 49-65.

21. Broersen, P.M.T., "Estimation of Multivariable Railway Vehicle Dynamics from Normal Operating Records," Identification and System Parameter Estimation, American Elsevier Publishing Company, New York, 1973, pp 425, 434.

22. Iliff, K.W., "Identification and Stochastic Control of an Aircraft Flying in Turbulence," J. Guidance and Control Vol. 1, No. 2, March-April 1978, pp 101-108.

23. Ward, R.C., "Numerical Computation of the Matrix Exponential with Accuracy Estimate," SIAM J. Numerical Analysis, Vol. 14, pp 600-610, 1977.

24. Henrici, P., Discrete Variable Methods in Ordinary Differential Equations, Wiley, 1962.

25. Forsythe, G., Moler, C.B., Computer Solution of Linear Algebraic Systems, Prentice-Hall, 1967.

26. Bierman, G.J., Factorization Methods for Discrete Sequential Estimation, Academic Press, New York, 1977.

27. Bard, Y., "Comparison of Gradient Methods for the Solution of Nonlinear Parameter Estimation Problems," SIAM J. Numerical Analysis, Vol. 7, No. 1, March 1970.

28. Dennis, Jr., J.E., More, J.J., "Quasi-Newton Methods, Motivation and Theory," SIAM Review, Vol. 19, No. 1, Jan. 1977, pp 46-87.

29. Iliff, K.W., and Taylor, L.W. Jr., "Determination of Stability Derivatives from Flight Data Using a Newton-Raphson Minimization Technique," NASA TN D-6579, 1972.

30. Levenberg, K., "A Method for the Solution of Certain Nonlinear Problems in Least Squares," Quart. Appl. Math., Vol. 2, 1944, pp 164-168.

31. Cramer, H., Mathematical Methods of Statistics, Princeton University Press, Princeton, NJ, 1946, pp 473-524.

32. Brown, K.M., Dennis, J.E. Jr., "Derivative Free Analogues of the Levenberg-Marquardt and Gauss Algorithms for Nonlinear Least Squares Approximation," Numer. Math. Vol. 18, pp 289-297.

33. Maine, R.E., and Iliff, K.W., "Use of Cramer-Rao Bounds on Flight Data with Colored Residuals," J. Guidance and Control, Vol. 4, No. 2, March - April 1981, pp 207 - 213.

34. Hall, W.E. Jr., and Gupta, N.K., "System Identification for Nonlinear Aerodynamic Flight Regimes," J. Spacecraft and Rockets, Vol. 14, No. 2, Feb. 1977, pp 73-80.

35. Hall, W.E. Jr., Gupta, N.K., Smith, R.G., "Identification of Aircraft Stability and Control Coefficients for the High Angle of Attack Regime," Engineering Technical Report under Contract NC0014-72-C-0328 to Office of Naval Research, March 1974.

36. Gupta, N.K. and Mehra, R.K., "Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations," IEEE Transactions on Automatic Control, Vol AC-19, No. 6, Dec 1974, pp 774-783.

37. Powell, M.J.D., "A Method for Minimizing A Sum of Squares of Nonlinear Functions without Calculating Derivatives," Computer Journal, Vol 7, 1965, pp 303-307.

38. Dennis, J.E. Jr., and Welsch, R.E., "Techniques for Nonlinear Least Squares and Robust Regression," Commun. Statist. - Simula. Computa., Vol B7, No. 4, 1978, pp 345-359.

39. Theil, H. "Generalized Least Squares and Linear Constraints: Correlated Disturbances and Autoregressive Transformations," Principles of Econometrics, John Wiley & Sons, New York, 1971.

40. de Boor, C., A Practical Guide to Splines, Springer-Verlag, New York, 1978.

41. Hayes, J.G., Halliday, J., "The Least Squares Fitting of Cubic Spline Surfaces to General Data Sets," J. Inst. Maths. Applics., Vol. 14, 1974, pp 89-103.

Table 1: System Identification Data Processing Methods

METHOD	ADVANTAGES	DISADVANTAGES	PERFORMANCE INDEX	MINIMIZATION ALGORITHM TYPICALLY USED	EXAMPLES OF APPLICATIONS
Equation Error	• Effective in presence of process noise • Computational simplicity	Sensitive to measurement errors	$\Sigma\left[\frac{dx}{dt} - f(x,u,\theta,t)\right]^2$	linear least squares [14]	aeronautics [20]
Output Error	• Effective in presence of measurement errors	Sensitive to process noise	$\Sigma[y - \hat{y}(t,\theta)]^2$	Gauss-Newton [16]	rail vehicles [21] gas turbines [4]
Combined State and Parameter Estimation	• Effective in presence of both measurement and process noise	Computational complexity	$\Sigma[y - \hat{y}(t,\theta,y)]^2$	Gauss-Newton [16] Kalman filter [17]	aeronautics [22]

NUMERICAL MATHEMATICAL REQUIREMENT		EFFECTIVE METHOD OF APPROACH
SOLUTION OF DIFFERENTIAL EQUATIONS	LINEAR	TRANSITION MATRIX [23]
	NONLINEAR	MULTISTEP METHODS (ADAMS-BASHFORTH) [24]
SOLUTION OF LINEAR ALGEBRAIC SYSTEMS OF EQUATIONS	POSITIVE DEFINITE, SYMMETRIC	CHOLESKY [25]
	GENERAL	GAUSSIAN ELIMINATION [25]
SOLUTION OF LEAST SQUARES PROBLEMS	LINEAR	FACTORIZATION, OR SQUARE ROOT METHODS [26]
	NONLINEAR	GAUSS-NEWTON [27]
MINIMIZATION OF GENERAL NONLINEAR MULTIVARIABLE FUNCTIONS		QUASI-NEWTON [28]

Table 2: Numerical Methods Used in System Identification

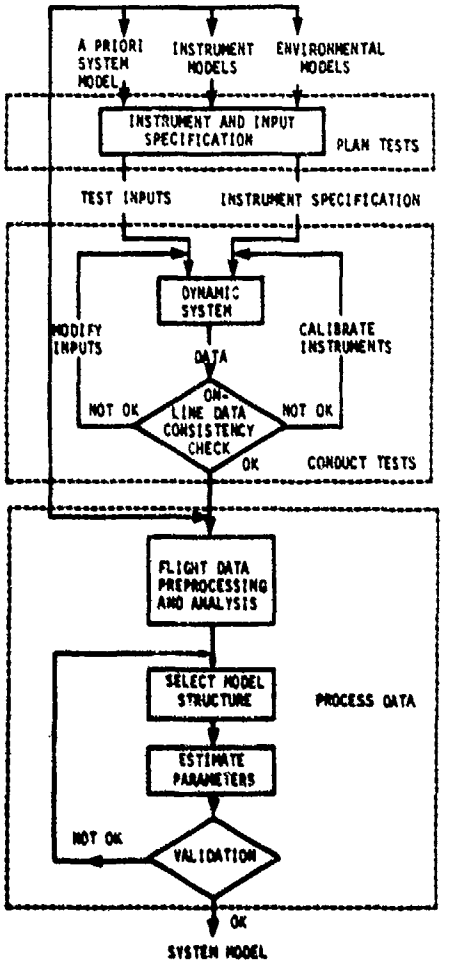


Figure 1: The Integrated System Identification Procedure

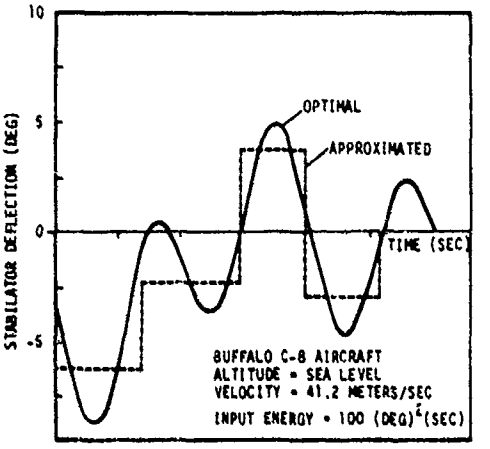


Figure 2: Optimal and Approximated Elevator Inputs

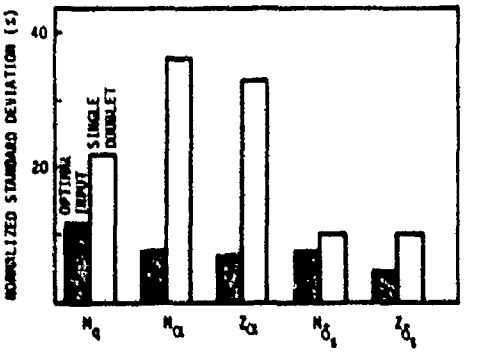
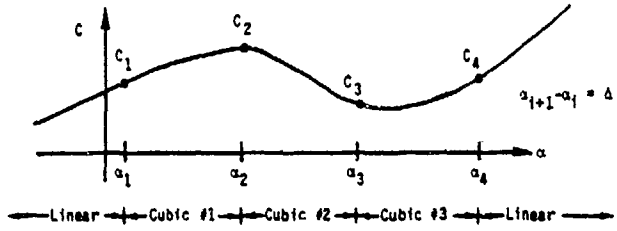


Figure 3: Comparison of Standard Deviation in Parameter Estimates for Optimal Input and Doublet (Same Input Energy)



$$C = \sum_{i=1}^{4} C_i * K_{Ai}(\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha)$$

C_i are coefficients to be identified.

	$\alpha \le \alpha_1$	$\alpha_1 \le \alpha \le \alpha_2$	$\alpha_2 \le \alpha \le \alpha_3$	$\alpha_3 \le \alpha \le \alpha_4$	$\alpha_4 \le \alpha$
K_{A1}	$1 - \frac{19}{15}\frac{(\alpha-\alpha_1)}{\Delta}$	$\frac{\alpha_2-\alpha}{\Delta} + \frac{4}{15}a$	$\frac{4}{15}b - \frac{1}{15}c$	$\frac{1}{15}d$	$-\frac{1}{15}\frac{(\alpha-\alpha_4)}{\Delta}$
K_{A2}	$\frac{8}{5}\frac{(\alpha-\alpha_1)}{\Delta}$	$\frac{\alpha-\alpha_1}{\Delta} - \frac{3}{5}a$	$\frac{\alpha_3-\alpha}{\Delta} - \frac{3}{5}b + \frac{2}{5}c$	$\frac{2}{5}d$	$\frac{2}{5}\frac{(\alpha-\alpha_4)}{\Delta}$
K_{A3}	$-\frac{2}{5}\frac{(\alpha-\alpha_1)}{\Delta}$	$\frac{2}{5}a$	$\frac{\alpha-\alpha_2}{\Delta} + \frac{2}{5}b - \frac{3}{5}c$	$\frac{\alpha_4-\alpha}{\Delta} - \frac{3}{5}d$	$-\frac{8}{5}\frac{(\alpha-\alpha_4)}{\Delta}$
K_{A4}	$\frac{1}{15}\frac{(\alpha-\alpha_1)}{\Delta}$	$-\frac{1}{15}a$	$-\frac{1}{15}b + \frac{4}{15}c$	$\frac{\alpha-\alpha_3}{\Delta} + \frac{4}{15}d$	$1 + \frac{19}{15}\frac{(\alpha-\alpha_4)}{\Delta}$

$$a = \frac{1}{\Delta^2}\left[\frac{(\alpha-\alpha_1)^3}{\Delta} - \Delta(\alpha-\alpha_1)\right] \qquad c = \frac{1}{\Delta^2}\left[\frac{(\alpha-\alpha_2)^3}{\Delta} - \Delta(\alpha-\alpha_2)\right]$$

$$b = \frac{1}{\Delta^2}\left[\frac{(\alpha_3-\alpha)^3}{\Delta} - \Delta(\alpha_3-\alpha)\right] \qquad d = \frac{1}{\Delta^2}\left[\frac{(\alpha_4-\alpha)^3}{\Delta} - \Delta(\alpha_4-\alpha)\right]$$

$$\Delta = \alpha_{i+1} - \alpha_i$$

Figure 4: Cubic Interpolating Polynomials

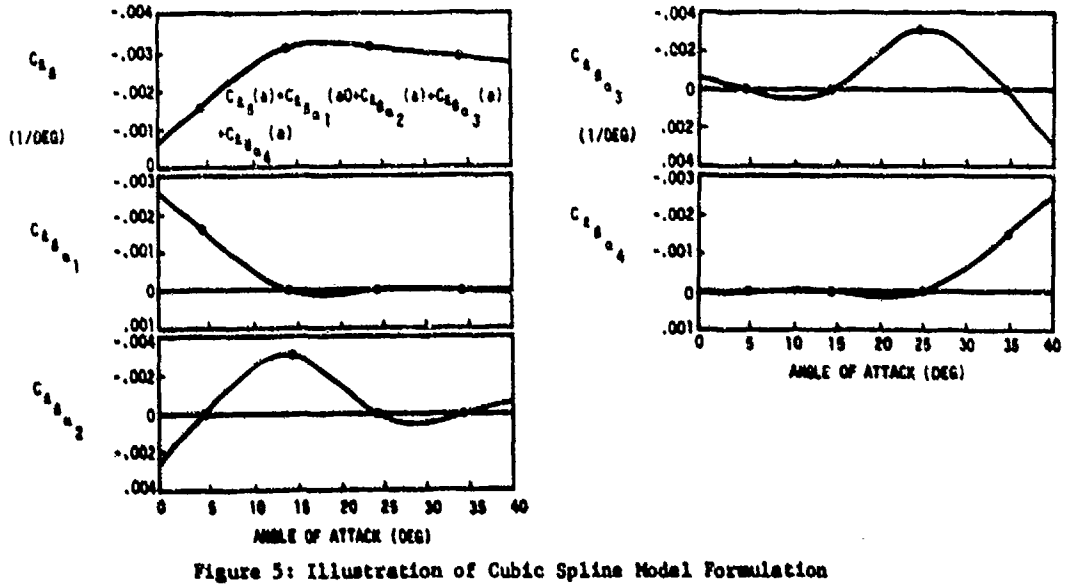


Figure 5: Illustration of Cubic Spline Modal Formulation

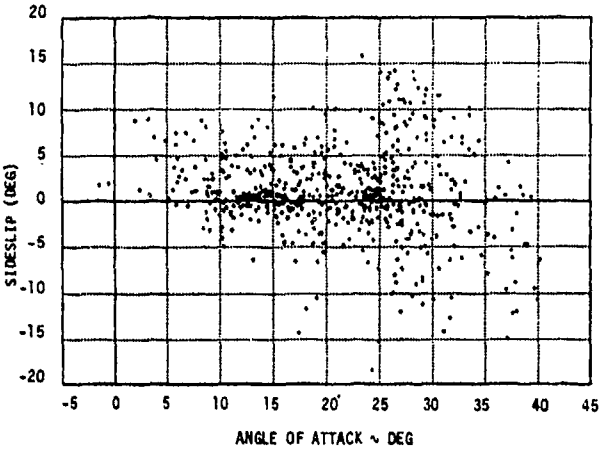


Figure 6: Angle of Attack/Sideslip Test Data Crossplot

SENSOR	BIAS	SCALE FACTOR	MISALIGNMENT	INITIAL ERROR
ACCELEROMETERS	✓	✓	✓	
RATE GYROS	✓	✓	✓	
VERTICAL GYRO VANE	✓	✓	✓	✓
IMPACT PRESSURE		✓		

Figure 7: Instrumentation System Error Models

- NOSEBCOM VANE

  $\alpha_{meas} = 1.102\hat{\alpha} + .06$ deg

- NOSEBOOM VANE

  $\beta_{meas} = 1.061\hat{\beta} - .63$ deg

- DYNAMIC PRESSURE

- ACCELEROMETER
  - AXIAL: BIAS = $.01\ f_{ps}^2$
  - LATERAL: BIAS = $.50\ f_{ps}^2$
  - VERTICAL: BIAS = $-2.10\ f_{ps}^2$

- VERTICAL GYRO
  - ROLL ALTITUDE: Vertical Alignment: $-.63^0 < \phi_c < 1.36^0$
  - PITCH ALTITUDE: Vertical Alignment: $.43^0 < \theta_c < .23^0$

- RATE GYRO
  - ROLL: Defective:
  - PITCH: Bias: $-.34^0/Sec < b_q < -.05^0/Sec$*
    Scale Factor: -.052
  - YAW: Bias: $.38^0/Sec < b_r < .58^0/Sec$*
    Misalignment from Vertical: 1.8^0
  - * Dependent on Maneuver

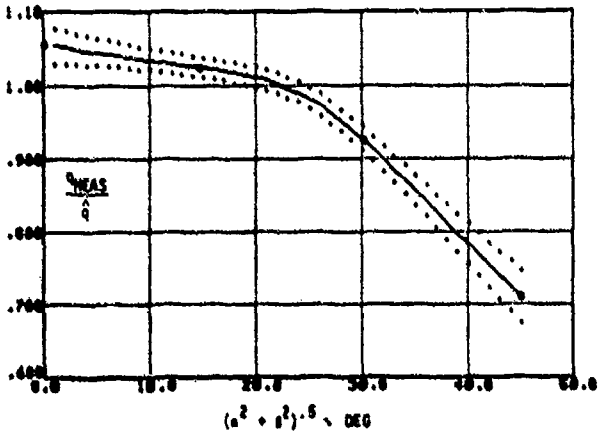
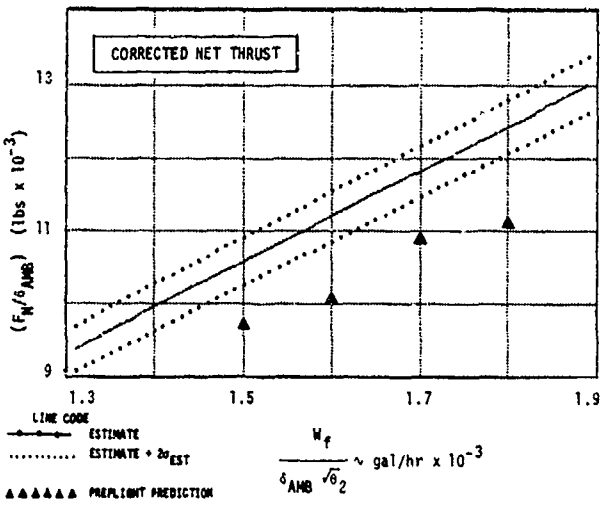


Figure 8: Identified Instrumental System Calibration Factors



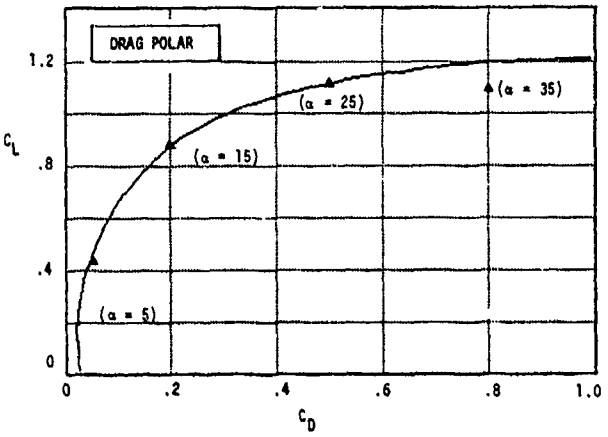
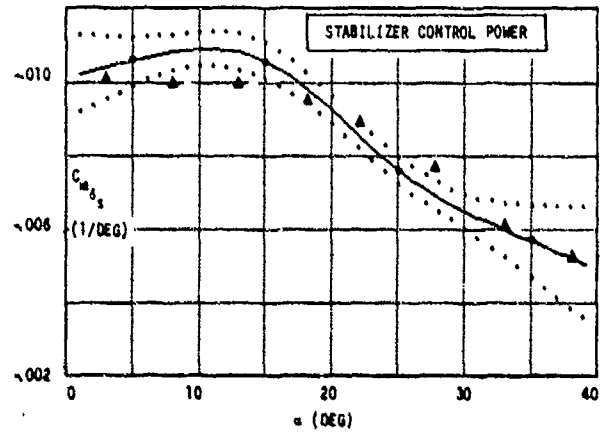


Figure 9: Identified Performance Characteristics



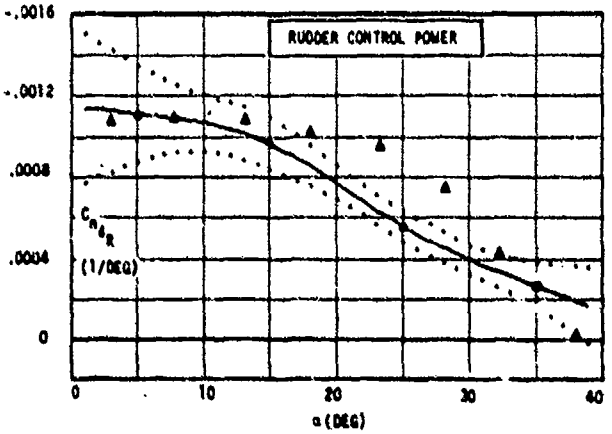


Figure 10: Identified Control Power Derivatives
(Continued on next page)

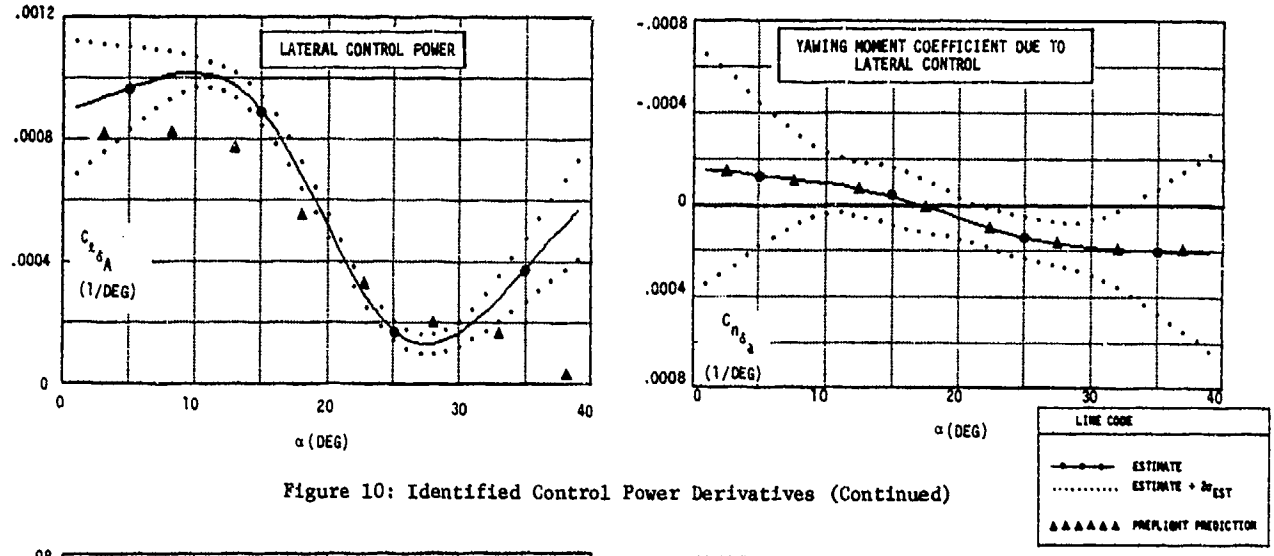


Figure 10: Identified Control Power Derivatives (Continued)

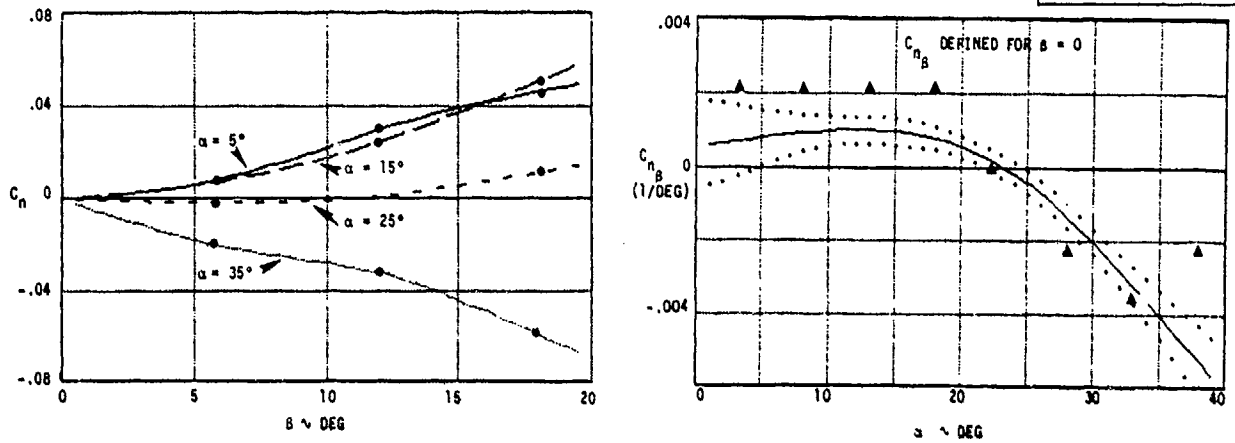


Figure 11: Identified Directional Static Stability

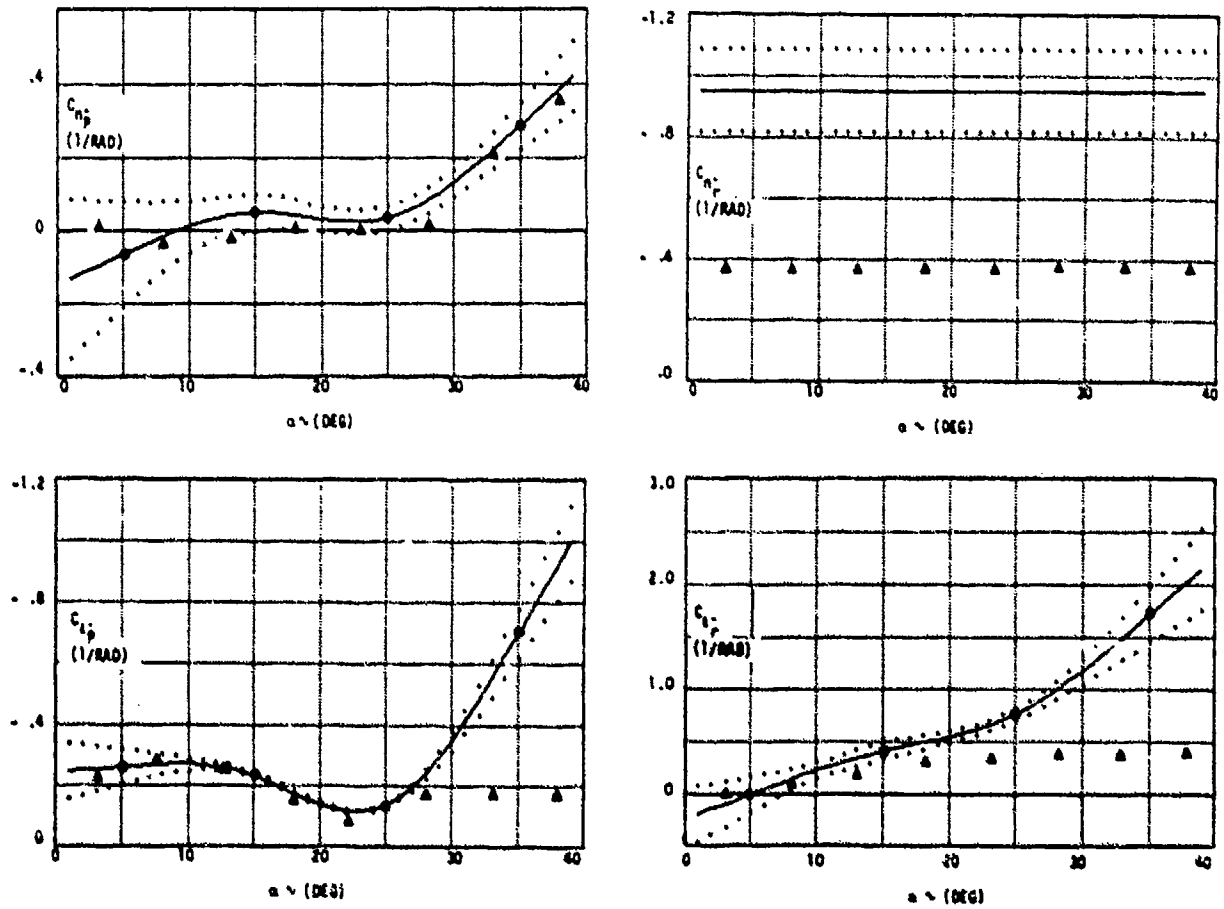


Figure 12: Identified Lateral-Directional Rotary Derivatives

MEASUREMENT	UNITS	RMS RESIDUAL FOR COMBINED SIX MANEUVERS IDENT. AERO/PROP CALIBRATED INS.	PREFLIGHT AERO/PROP CALIBRATED INS.
P	deg/sec	6.15	13.81
Q	deg/sec	2.85	11.61
R	deg/sec	1.78	11.04
αN/B	deg	1.07	2.82
βN/B	deg	.98	2.35
q	psf	2.22	6.56
b_p	ft	94.00	215.00
n_x	g's	.009	.028
n_y	g's	.012	.016
n_z	g's	.036	.073

Note: Measure Residual = Actual Measurement - Estimate of Measurement

Figure 13: Effect of Model Parameters on Measurement Residuals

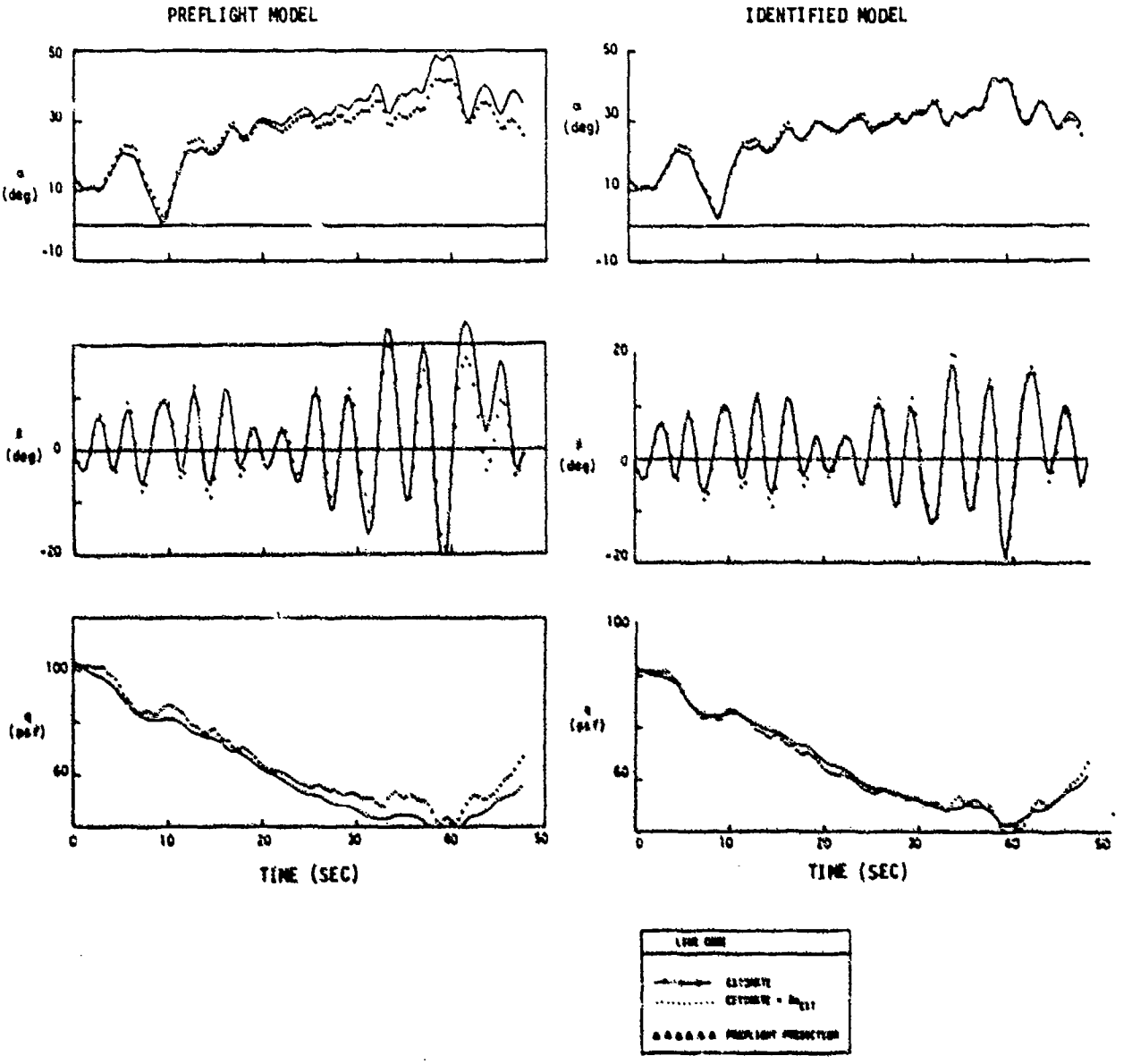


Figure 14: Measurement Prediction Improvement Due to Identified Parameters (continued on next page)

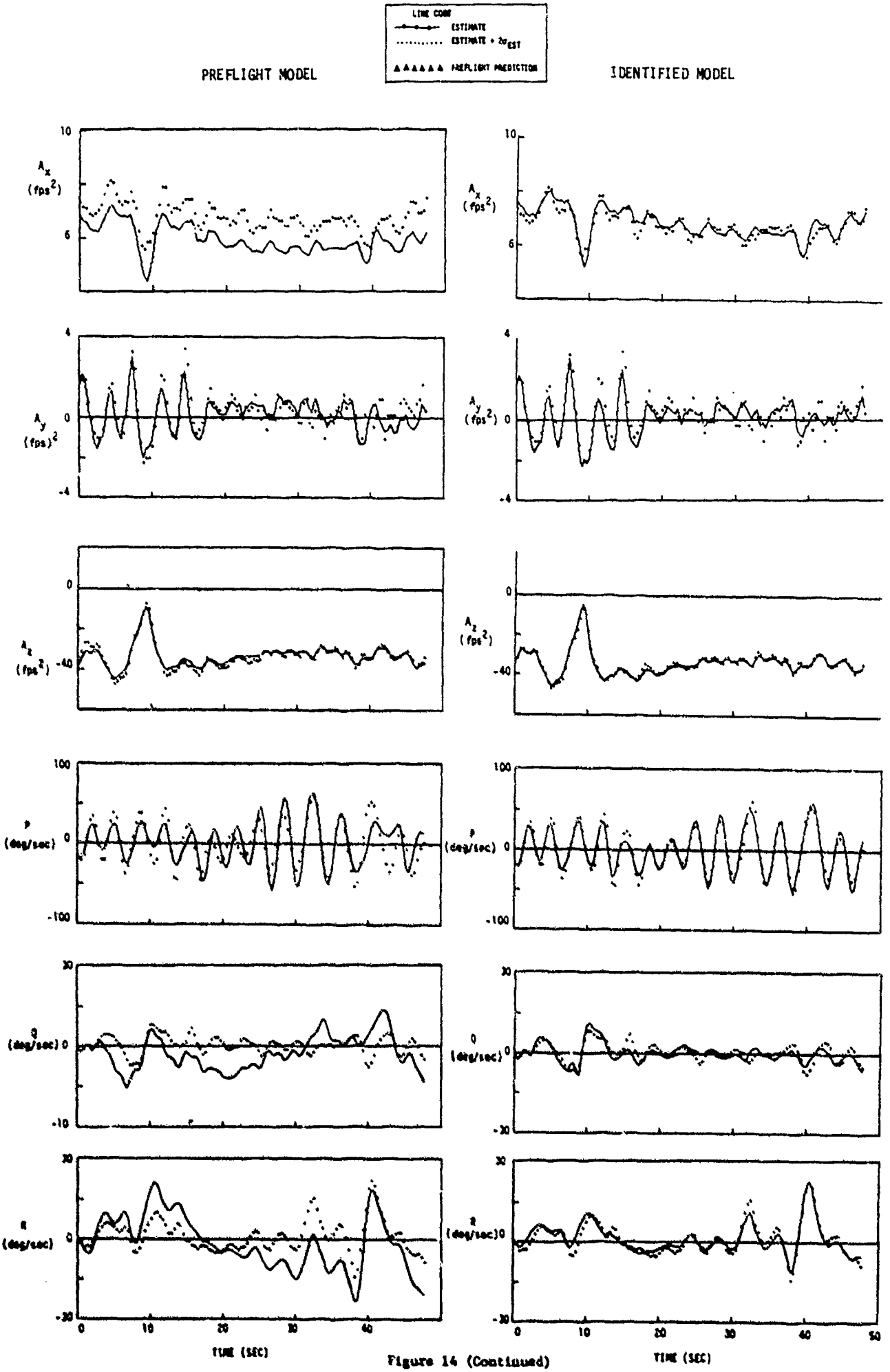


Figure 14 (Continued)

TECHNIQUES AND METHODOLOGIES FOR THE ESTIMATION OF COVARIANCES, POWER SPECTRA AND FILTER-STATE AUGMENTATION

by

Volkmar Held
Elektronik-System-Gesellschaft mbH
Vogelweideplatz 9
8000 München 80
West Germany

SUMMARY

For the realization of an optimal filter in practice quantitative knowledge of the system and measurement noise is necessary. The required stochastic parameters, covariances power spectra and in the case of coloured noise the dynamic model of the shaping filter usually has to be determined from measurements which precede the filter design.

This paper describes a technique for the estimation of stationary stochastic noise data from real measurements which contain the noise as well as non-observable deterministic values. The separation is accomplished by a specific smoothing procedure. The stochastic behaviour of the data is proven by a Gaussian probability distribution test. Then, based on conventional stochastic relations the covariarce and the spectral density are evaluated.

In the next step shaping filters are determined from the power spectral density to model coloured noise. The filter structure is selected and the filter parameters are optimally identified by a least squares procedure.

Finally the filter state is augmented by the degree of the shaping filter. As further evidence examples for the individual steps are given.

1. INTRODUCTION

For the design of optimal filters the dynamic model of the system and information about the expected stochastic system disturbances and measurement errors are required [1].

The dynamic system-model is generally derived without major problems from knowledge of the physical background of the system or from measurement of the system transfer-function.

Much more difficult is the determination of the system disturbances and measurement errors, here called system- and measurement noise. They have to be provided for the filter algorithms in the form of stochastic parameters: covariances and power spectral density. The estimation of these parameters requires theoretically an infinite number of measurements of a stationary stochastic noise-process. In reality only time-limited measurements exist which are not exactly stationary and contain additional deterministic variables.

The results of the parameter estimation are approximations which are usually sufficient if some limiting conditions (controllability and observability of the system and stability of the designed filter [2]) are guaranteed.

Another problem is the requirement of Gaussian distributed and uncorrelated (white) noise for the Kalman filter formulation. In reality white noise does not exist. Real noise (for example gyro drift or wind speed) is correlated, has a limited frequency band and is called coloured noise. A solution of this problem is possible by the so-called state vector augmentation [3] . Coloured Gaussian noise can be generated by a linear shaping-filter from Gaussian white noise (Fig. 1). The shaping filter, usually a first or second order linear filter, is added to the system model and the state vector of the



Fig. 1 Shaping Filter

system is augmented by the order of the shaping-filter. While this procedure is applicable to the system-noise without problems the treatment of the measurement-noise is, at least in theory, difficult. If the measurement-noise is integrated in the system by a shaping filter the measurements are error-free which would result in a loss of stochastic observability and filter stability.

Exact but relatively difficult solutions for the coloured noise are given in [4], [2].

In reality this is no problem. Coloured measurement-noise usually contains an additional part of high bandwidth which is approximately white and can be separated from the coloured part. While the coloured part is modelled by a shaping filter the approximately white part serves in the filter algorithm as the required measurement-noise. So the application of coloured system and measurement noise in a Kalman filter is possible.

In the following sections techniques for the estimation of the noise-parameters from real measurement for the application in a Kalman filter are described. In particular the evaluation of the covariance, power spectrum (for continuous filters) and of shaping filters are treated. These methods are based on the conventional mathematical relations for linear stochastic processes [2], [5].

2. DETERMINATION OF STATIONARY MEASUREMENTS

The large number of measurements which is required for the determination of stochastic parameters (e.g. variance or power spectral density) generally is not derived from an ensemble of measurements at a fixed time. Usually the time-variable measurements of one ensemble member is applied. This is correct, if the ergodic hypothesis is valid which means that this member is representative for the desired stationary stochastic noise process. Unfortunately, such representatives measurements are rarely available in practice. In most cases non-observable deterministic signals or measurement errors (trends) are superimposed to the stochastic noise. For example the Signal of a doppler-radar contains the deterministic aircraft velocity as well as the stochastic measurement error noise. If no redundant measurements are available a separation of the two variables is only possible by their different frequency behaviour. The resonance frequencies or the bandwidth of the observed system is approximately known from its physical background (e.g. phygoide of an aircraft). Usually this frequency lies below the frequency band of the measurement noise so that a separation by a filter or smoothing procedure is possible. Below a smoothing procedure is described which gives an excellent frequency separation.

Smoothing procedures have the property to provide off line a smoothed value at time t_i from measurements before and after t_i. If the procedure is repeated for varying t_i the result is a smoothed variable which consists of the lower frequency-parts of the measurements. The smoothing procedure is characterized by the following steps:

- From the measurements $y(t_i + \nu)$ the data within a data-window of the length $2\nu_{max}$ (Fig. 2) are selected.
- The measurements are approximated by a second order polynomial

$$\hat{y}(t_i+\nu) = a_0(t_i) + a_1(t_i)\nu + a_2(t_i)\,\nu^2/2 \qquad (1)$$

- The differences:

$$\hat{y}(t_i+\nu) - y(t_i+\nu) = \Delta y(t_i+\nu)$$

are weighted with a weighting-function $g(|\nu|)$ which is symmetrical to t_i (Fig. 2)

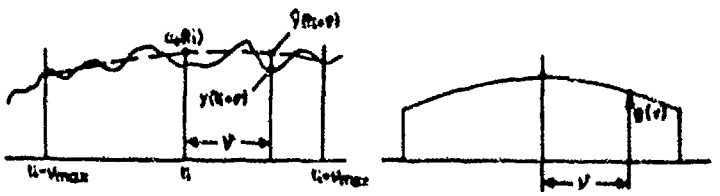


Fig. 2
Measurements y (~~)
Approximation ŷ (- - -)
Weighting function g(| ν |)

- The constants $a_0(t_i)$, $a_1(t_i)$ and $a_2(t_i)$ are estimated by the method of least squares.
- As shown later, the cut-off frequency of the smoothing procedure is defined by the length of the data window $2\nu_{max}$ and its transfer function by the weighting-function $g(\nu)$.
- If the data window is shifted along the time scale, the smoothed data are given by $a_0(t_i)$ with variable i.

In detail, the evaluation of the polynomial coefficients and of the transfer function runs as follows:

The differences, Eq. (1) are weighted, squared and integrated from $-\nu_{max}$ to $+\nu_{max}$:

$$J = \int_{-\nu_{max}}^{+\nu_{max}} g(\nu)^2 \left(\hat{y}(t_i+\nu) - y(t_i+\nu)\right)^2 d\nu$$

With Eq. (1), differentiation of J and setting the differential to 0 for the minimum of J results in:

$$\frac{\partial J}{\partial a_k} = 2 \int_{-\nu_{max}}^{+\nu_{max}} g(\nu)^2 \left(\hat{y}(t_i+\nu) - y(t_i+\nu)\right) \frac{\nu^k}{k!}\, d\nu = 0 \qquad (2)$$

k = 0, 1, 2.

With Eq. (1) this yields:

$$\int_{-v_{max}}^{+v_{max}} g(v)^2 v^k y(t_i+v)dv = \int_{-v_{max}}^{+v_{max}} g(v)^2 \left(a_0(t_i)v^k + a_1(t_i)v^{1+k} + a_2(t_i)\frac{v^{2+k}}{2}\right)dv \qquad (3)$$

Uneven powers of v disappear on the right side of Eq. (3) during the integration and the following relation remains:

$$\int_{-v_{max}}^{+v_{max}} g(v)^2 \begin{bmatrix} y(t_i+v) \\ v\ y(t_i+v) \\ v^2\ y(t_i+v) \end{bmatrix} dv = \int_{-v_{max}}^{+v_{max}} g(v)^2 \begin{bmatrix} 1 & 0 & \frac{v^2}{2} \\ 0 & v & 0 \\ v^2 & 0 & \frac{v^4}{2} \end{bmatrix} dv \cdot \begin{bmatrix} a_0(t_i) \\ a_1(t_i) \\ a_2(t_i) \end{bmatrix} \qquad (4)$$

The matrix on the right hand side is non-singular and can be inverted for the estimation of $a_k(t_i)$. Usually only $a_0(t_i)$ is of interest, sometimes the first (a_1) or second (a_2) derivative is required too.

For the determination of the cut-off frequency ω_g and transfer function $F(i\omega)$ of the smoothing procedure, Eq (4), is Fourier-transformed. The convolution of the left hand sind of Eq. (4) changes into the product of two Fourier-transformed integrals. The matrix of the right hand side consists of constants which remain unchanged during the Fourier-transformation. The result is equation (5):

$$\mathcal{F}\begin{bmatrix} g(v)^2 \\ g(v)^2 \cdot v \\ g(v)^2 \cdot v^2 \end{bmatrix} \cdot \mathcal{F}\, y(t_i) = \int_{-v_{max}}^{+v_{max}} g(v)^2 \begin{bmatrix} 1 & 0 & \frac{v^2}{2} \\ 0 & v & 0 \\ v^2 & 0 & \frac{v^4}{2} \end{bmatrix} dv \cdot \mathcal{F}\begin{bmatrix} a_0(t_i) \\ a_1(t_i) \\ a_2(t_i) \end{bmatrix} \qquad (5)$$

$\mathcal{F}$: Fourier transformation

with:

$$\mathcal{F}\begin{bmatrix} g(v)^2 \\ g(v)^2 v \\ g(v)^2 v^2 \end{bmatrix} = \int_{-v_{max}}^{+v_{max}} \begin{bmatrix} \cos\omega v \cdot g(v)^2 \\ -\sin\omega v\ g(v)^2 v \\ \cos\omega v\ g(v)^2 v^2 \end{bmatrix} \cdot dv \qquad (6)$$

The desired transfer functions:

$$F(i\omega)_{a_0} = \frac{\mathcal{F}\, a_0(t_i)}{\mathcal{F}\, y(t_i)}, \quad F(i\omega)_{a_1} = \frac{\mathcal{F}\, a_1(t_i)}{\mathcal{F}\, y(t_i)}, \quad F(i\omega)_{a_2} = \frac{\mathcal{F}\, a_2(t_i)}{\mathcal{F}\, y(t_i)} \qquad (7)$$

are determined by solution of Eq. (5).

The transfer functions depend exclusively on the weighting function. $g(v)^2$ has to be chosen such that the smoothing procedure approximates an ideal low pass as closely as possible. Therefore in the following section different weighting functions are examined.

3. WEIGHTING FUNCTIONS

The simplest weighting function is a rectangular function $g_0(v) = 1$. The measurements within a data-window are weighted equally (Fig. 3).

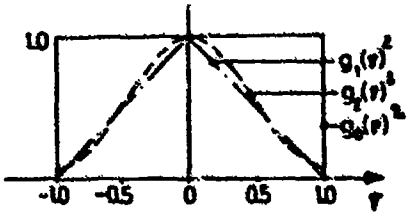


Fig. 3 Weighting functions $g_1(v)^2$ and $g_2(v)^2$

The amount of the corresponding transfer-function $\mathcal{F}_0(i\omega)a_0$ Eq. (7) is shown in Fig. (4). The function decreases down to a frequency of $0.5/\nu_{max}$ very rapidly which results in a sharp frequency cut-off. For increasing frequencies the function oscillates which means that the smoothed signal still contains parts of higher frequencies. To eliminate this disadvantage two different weighting-functions are tested. These functions are used in [6] for the smoothing of power spectral densities.

$$g_1(\nu)^2 = 1 - \frac{|\nu|}{\nu_{max}} \tag{8}$$

$$g_2(\nu)^2 = 0.54 + 0.46 \cos \frac{\pi \cdot \nu}{\nu_{max}} \tag{9}$$

The amount of the corresponding transfer-functions $\mathcal{F}_1(i\omega)a_0$, $\mathcal{F}_2(i\omega)a_0$

Eq. (8), (9)

is shown in Fig. (4):

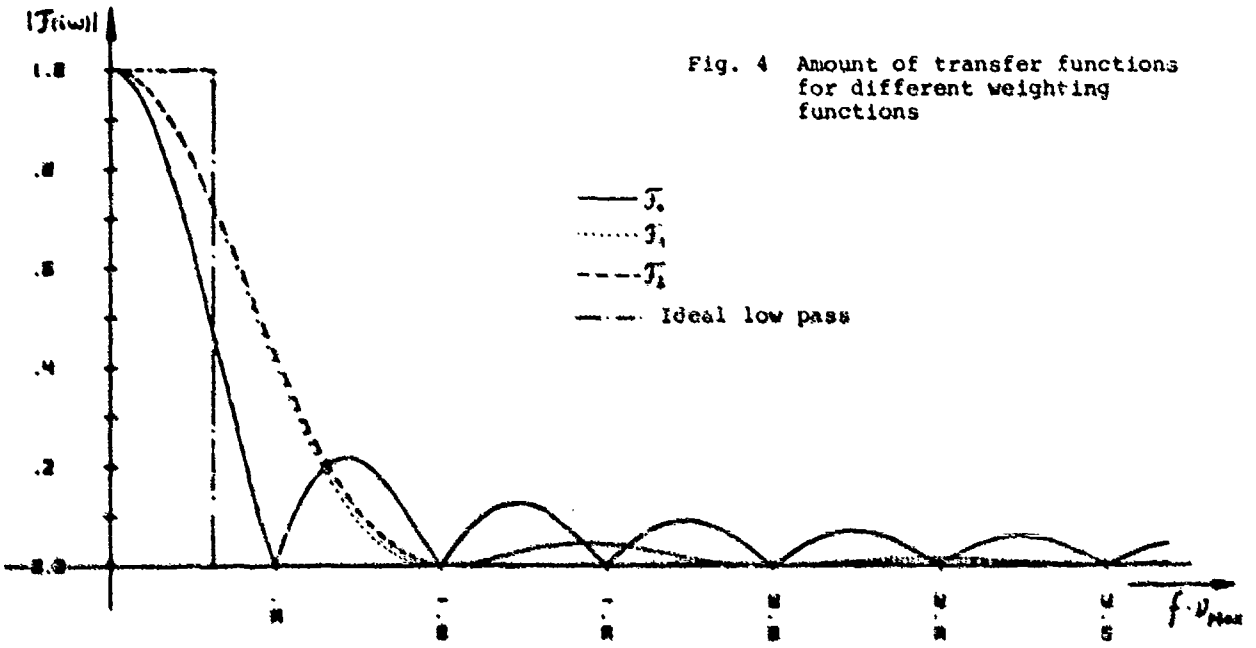


Fig. 4 Amount of transfer functions for different weighting functions

$\mathcal{F}$ is displayed dependent on the relative frequency f/ν_{max} where $2\ \nu_{max}$ is the width of the smoothing-function. The decrease of $\mathcal{F}_2$ and $\mathcal{F}_1$ is slighter than of $\mathcal{F}_0$ but the oscillations are much smaller, especially for the weighting-function $g_2(\nu)$.

In the logarithmic diagrams of Fig. (5) these properties are shown more distinctive.

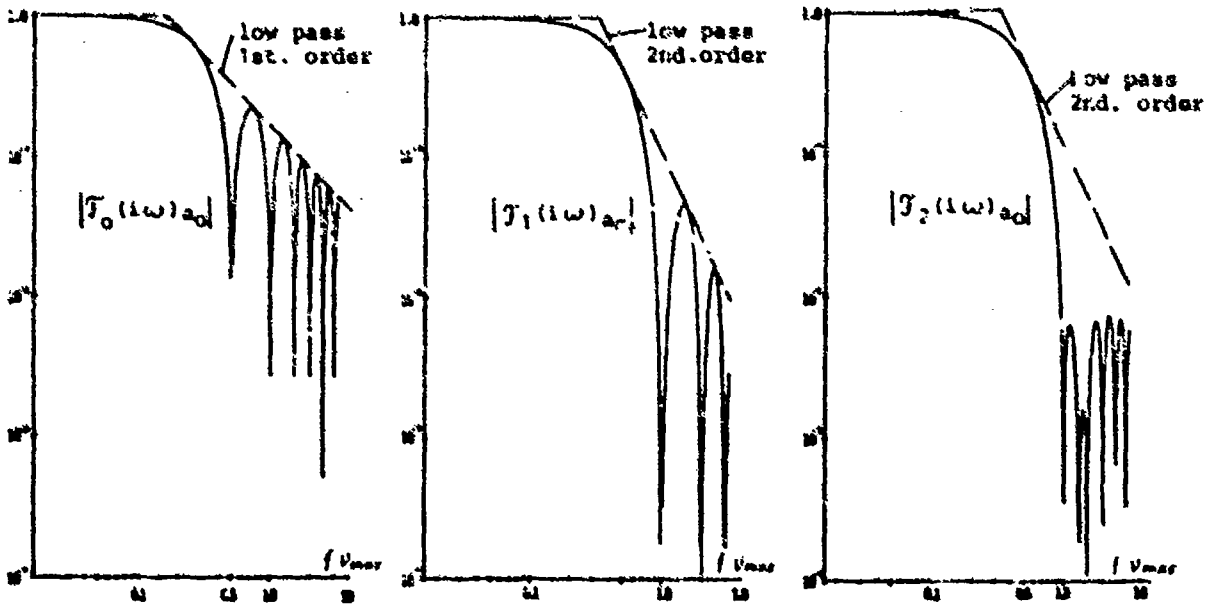


Fig. 5 Amount of transfer functions

For comparison, also low-pass filters of first and second order are displayed. Only with the weighting function $g_2(\nu)$ the smoothing procedure yields a much better frequency separation than a second order bandpass-filter. In this case it is a good approximation of an ideal low-pass. Therefore the weighting function $g_2(\nu)$ is recommended for application in the described smoothing procedure. The cut-off frequency is $f = 1/\nu_{max}$ (Fig. (5)). The smoothing procedure now separates deterministic or stochastic signals $\hat{y}(t_i) = a_o(t_i)$ from stochastic noise $(\Delta y\ (t_i))$ which is approximately stationary.

4. TEST OF GAUSSIAN DISTRIBUTION

Prior to the estimation of stochastic parameters from the stochastic noise a Gaussian probability distribution test is performed to prove the stochastic behaviour of the noise.

Fig. 6 shows an example of this test, (probability distribution of stochastic gyro-drift data which result from in-flight measurements). The criterion is a χ^2-test and a straight line in the Gaussian probability-distribution paper.

HAEUFIGKEITSVERTEILUNG VON 133 MESSWERTEN:

NR.	KLASSE VON	BIS	HAEUFIGKEIT EINZ.	SUMME	STRICHLISTE
0	-UNENDLICH	-1.1748673281.+1	0	0	|
1	-1.1748673282.+1	-8.8115049615.+0	9	9	|////X////
2	-8.8115049615.+0	-5.8743366409.+0	16	25	|////X////X////X/
3	-5.8743366409.+0	-2.9371683204.+0	17	42	|////X////X////X///
4	-2.9371683204.+0	1.1641532183.-10	26	70	|////X////X////X////X////X////
5	1.1641532183.-10	2.9371683206.+0	19	89	|////X////X////X////
6	2.9371683206.+0	5.8743366412.+0	19	108	|////X////X////X////
7	5.8743366412.+0	8.8115049617.+0	15	123	|////X////X////X
8	8.8115049617.+0	1.1748673281.+1	9	132	|////X////
9	1.1748673282.+1	+UNENDLICH	1	133	|/

DIFFERENZEN ZWISCHEN MESSUNG UND NORMAL VERTEILUNG

3.2
3.9
-2.8
2.7
-6.3
-0.8
2.9
3.2

CHIQUADRATTEST:
CHIQUADRAT 7.6816

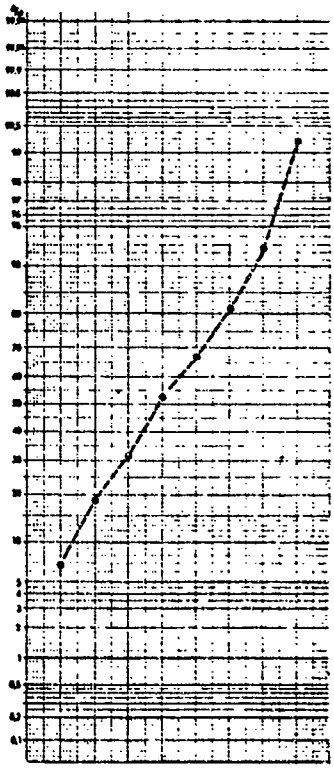


Fig. 6 Probability distribution of 133 gyro-drift measurements. χ^2-test and Gaussian probability distribution paper-test.

The result of the χ^2-test $(7.68 < 9.24)$ as well as the approximately straight line shows that the gyro drift is Gauss-distributed.

5. ESTIMATION OF COVARIANCES AND POWER SPECTRA

In the preceding three sections a technique has been derived to extract stationary stochastic system- or measurement noise data from measurements which contain system signals, deterministic errors or trends. Now the stochastic parameters, covariances and power spectral densitites, which are required for the Kalman filter can be estimated. As the smoothing procedure runs in a digital computer the noise output is a discrete time series. Therefore the following equations are given in discrete formulation too.

Correlation functions

For two discrete stochastic series $y_1(t_j)$ and $y_2(t_j+m\Delta T)$ with j = o, 1,.... 1-k, m = 0,1...K, mean value o and constant ΔT, the correlation function is:

$$\mathcal{R}^{*}_{y_1,y_2}(m\Delta T) = \frac{1}{I-k}\sum_{j=o}^{I-K} y_1(tj)\cdot y_2(tj+m\Delta T) \qquad (10)$$

$y_1 = y_2$: autocorrelation
$y_1 \neq y_2$: crosscorrelation

For negative m:

$$\mathcal{R}^{*}_{y_1,y_2}(-m\Delta T) = \mathcal{R}^{*}_{y_2,y_1}(m\Delta T)$$

The Covariance is given by $\mathcal{R}^{*}_{y_1,y_2}(o)$.

Power Spectral Density

In the frequency domain the stochastic functions y_1, y_2 are described by the power spectral density (signal power per Hertz at a frequency of $n\Delta f$. The power spectracl density is determined by the discrete Fourier-transformation of the correlation function.

$$\begin{aligned}\mathcal{S}^{*}_{y_1,y_2}(n\Delta f) = & \left\{\mathcal{R}_{y_1,y_2}(m\Delta T)\right\} = \\ & \Delta T\left[\mathcal{R}^{*}_{y_1,y_2}(0) + \sum_{m=1}^{K-1}\left(\mathcal{R}^{*}_{y_1,y_2}(m\Delta T) + \mathcal{R}^{*}_{y_2,y_1}(m\Delta T)\right)\cdot\cos\pi\frac{n\cdot m\Delta f}{fg}\right. \\ & \left. + \frac{1}{2}\left(\mathcal{R}^{*}_{y_1,y_2}(k\Delta T) + \mathcal{R}^{*}_{y_2,y_1}(k\Delta T)\right)\cos\pi\frac{n\,k\Delta f}{fg}\right] \\ & - i\Delta T\left[\sum_{m=1}^{K-1}\left(\mathcal{R}^{*}_{y_1,y_2}(m\Delta T) - \mathcal{R}^{*}_{y_2,y_1}(m\Delta T)\right)\cdot\sin\pi\frac{n\,m\Delta f}{fg}\right. \\ & \left. + \frac{1}{2}\left(\mathcal{R}^{*}_{y_1,y_2}(k\Delta T) - \mathcal{R}^{*}_{y_2,y_1}(k\Delta T)\right)\cdot\sin\pi\frac{n\,k\Delta f}{fg}\right]\end{aligned} \qquad (11)$$

With n = 0, 1, 2 ... K , $\Delta f = 1/2K\Delta T$ frequency resolution, $fg = 1/2\Delta T$ cut off frequency (Shannon), $i = \sqrt{-1}$.

For $y_1 \neq y_2$ $\mathcal{S}$ has an imaginary part which disappears for $y_1 = y_2$.

For the enhancement of the statistic certainty of the power spectral density, Eq. (11), the correlation function, Eq. (10) can be multiplied by a weighting-function, $g(m\Delta T)$:

$$\mathcal{R}_{y_1,y_2}(m\Delta T) = g(m\Delta T)\cdot\mathcal{R}^{*}_{y_1,y_2}(m\Delta T) \qquad (12)$$

The weighting function has a length of $2k\Delta T$. Multiplication in the time domain yields a convolution in the frequency domain:

$$\mathcal{F}\left\{\mathcal{R}_{y_1,y_2}(m\Delta T)\right\} = \mathcal{F}\left\{g(m\Delta T)\right\} * \mathcal{F}\left\{\mathcal{R}^{*}_{y_1,y_2}(m\Delta T)\right\}$$

It has been shown [6] that the weighting function $g_2(m\Delta T) = 0.54 + 0.46\cos(\pi m/K)$ smoothes the spectral density and enhances the statistic certainty by 2.3. The convolution is very simple because $\mathcal{F}\left\{g_2(m\Delta T)\right\}$ consists only of 3 values.

The result is given by:

$$\begin{aligned}\mathcal{S}_{y_1,y_2}(0) &= 0.54\cdot\mathcal{S}^{*}_{y_1,y_2}(0) + 0.46\,\mathcal{S}^{*}_{y_1,y_2}(\Delta f) \\ \mathcal{S}_{y_1,y_2}(n\Delta f) &= 0.23\cdot\mathcal{S}^{*}_{y_1,y_2}((n-1)\Delta f) + 0.54\,\mathcal{S}^{*}_{y_1,y_2}(n\Delta f) + 0.23\,\mathcal{S}_{y_1,y_2}((n+1)\Delta f) \\ \mathcal{S}_{y_1,y_2}(k\Delta f) &= 0.46\cdot\mathcal{S}^{*}_{y_1,y_2}((k-1)\Delta f) + 0.54\,\mathcal{S}^{*}_{y_1,y_2}(k\Delta f)\end{aligned} \qquad (13)$$

For the gyro drift data of Fig. 6 the correlation function, the power spectral density and the weighted spectral density are evaluated and displayed in Fig. 7 as an example. The frequency steps are 0.0159/sec and the validity of the relation $\int\int df = \mathcal{R}(0)$ is checked.

$\mathcal{R}^*$	I	=	0	34.248374	$\int^*$	I	=	0	894.8413346	$\int$	I	=	0	747.46569[illegible]
	I	=	1	31.[illegible]9493		I	=	1	574.4587946		I	=	1	510.3554[illegible]
	I	=	2	27.55[illegible]395		I	=	2	-24.6343795		I	=	2	127.627259
	I	=	3	23.671550		I	=	3	38.2[illegible]8748		I	=	3	16.777413
	I	=	4	20.383873		I	=	4	7.7[illegible]8056		I	=	4	1[illegible].9[illegible]32[illegible]
	I	=	5	15.333795		I	=	5	26.2095796		I	=	5	14.931[illegible]5[illegible]
	I	=	6	9.84[illegible]607		I	=	6	-4.3181245		I	=	6	5.548114
	I	=	7	4.485010		I	=	7	8.[illegible]598592		I	=	7	2.24[illegible]1
	I	=	8	0.130031		I	=	8	-4.8030209		I	=	8	0.892[illegible]9[illegible]
	I	=	9	-3.849921		I	=	9	7.0996667		I	=	9	2.017975
	I	=	10	-7.471387		I	=	10	-3.0919592		I	=	10	1.5[illegible]flag
										INTEGRAL SDF	=			34.2738532

Fig. 7 Evaluation of correlation function, power spectral density and convoluted power spectral density.

The table shows a much smoother weighted spectral density than the unweighted $\int^*$.

6. ESTIMATION OF LINEAR SHAPING FILTERS

One goal of the analysis of correlated stochastic noise data is the determination of linear shaping filters which generate coloured noise from white noise. If the spectral density, Eq. (13) is known, the following relation is valid [2] :

$$\int_{y_1,y_1}(\omega) = |\mathcal{F}(i\omega)|^2 \int_0$$

$\int_0$ Power spectral density of white noise (const.)
$\int_{y_1,y_1}(\omega)$ Power spectral density of coloured noise
$\mathcal{F}(i\omega)$ Transfer function of the shaping filter.

From this the amount of the shaping filter transfer-function follows:

$$|\mathcal{F}(i\omega)| = \sqrt{\int_{y_1 y_1}(\omega) / \int_0} \qquad (15)$$

From the empiric spectral density $\int_{y_1,y_1}(\omega)$ only the amount of the shaping filter transfer-function can be determined but not the phase. Therefore, all shaping filters theoretically can be used for the generation of the coloured noise which fulfil Eq. (15) approximately. In reality always the simplest of all possible shaping filters is used because the phase is of no interest for the noise.

A linear shaping filter is defined by its structure (differential equation) and its parameters (time constants, resonance frequencies and damping ratios). The selection of the structure and the estimation of the parameters is feasible by the following method: The amount of the transfer function, Eq. (15) is drawn in double-logarithmic scals. Two examples: gyro drift and velocity measurement error are shown in Fig. 8.

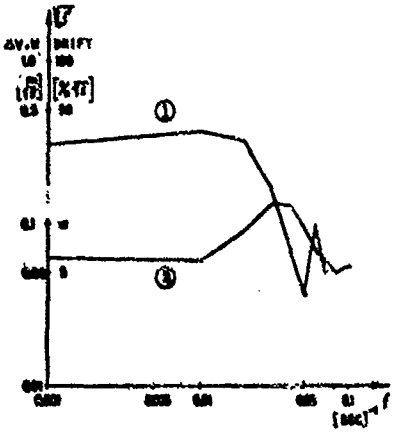


Fig. 8 Square-root of power spectral density of ① gyro drift, ② velocity measurement-error.

For a skilled person it is relatively easy to determine the structure of a linear filter which is a good approximation of the empiric value. For the examples of Fig. 8 the appropriate shaping-filter structures are given by the transfer functions (16) and (17) in the frequency domain (ω resonance frequencies, ξ damping constants, T time constants)

$$F(i\omega) = \frac{1}{-\frac{\omega^2}{\omega_{n1}^2} + i2\xi_1\frac{\omega}{\omega_{n1}} + 1} \tag{16}$$

$$F(i\omega) = \frac{iT_{v_1}\omega + 1}{-\frac{\omega^2}{\omega_4^2} + i2\xi_4\frac{\omega}{\omega_4} + 1} \tag{17}$$

Now the parameters α of the shaping filter (for the examples T_{vi}, ξ_i, ω_i and $\sqrt{S_0}$ have to be identified so that the empiric value $\sqrt{S_{y_1,y_1}(\omega)}$ is optimally approximated. This is obtained by minimizing the equation:

$$J(\alpha) = \sum_{\omega=0}^{\omega_g}\left[\sqrt{S_{y_1,y_1}(\omega)} - |F(i\omega,\alpha)|\cdot\sqrt{S_0}\right]^2 = \text{Min} \tag{18}$$

This equation is non-linear in reference to . One possible solution is the development of $J(\alpha)$ in a Taylor series [7] . The Taylor series is developed for $\alpha = \alpha^0$ (first approximation) and has three terms. For the scalar case (1 parameter) this yields:

$$J(\alpha) = J(\alpha)\Big|_{\alpha=\alpha^0} + \frac{\partial J}{\partial\alpha}\Big|_{\alpha=\alpha^0}\cdot(\alpha-\alpha^0) + \frac{\partial^2 J}{\partial\alpha^2}\Big|_{\alpha=\alpha^0}\cdot\frac{(\alpha-\alpha^0)^2}{2} = \text{Min} \tag{19}$$

The minimum of α is given by:

$$\frac{\partial J(\alpha)}{\partial\alpha} = 0 + \frac{\partial J}{\partial\alpha}\Big|_{\alpha=\alpha^0} + \frac{\partial^2 J}{\partial\alpha^2}\Big|_{\alpha=\alpha^0}(\alpha-\alpha^0) = 0 \tag{20}$$

For several parameters a vector $\underline{\alpha} = (\alpha_1, \alpha_2 \ldots\ldots)^T$ is introduced. Instead of Eq. (20) the following equation is valid:

$$\underbrace{\begin{bmatrix} \frac{\partial J}{\partial\alpha_1} \\ \frac{\partial J}{\partial\alpha_2} \\ \vdots \end{bmatrix}_{\underline{\alpha}=\underline{\alpha}^0}}_{\Delta} + \underbrace{\begin{bmatrix} \frac{\partial^2 J}{\partial\alpha_1^2} & \frac{\partial^2 J}{\partial\alpha_1\partial\alpha_2} & \cdots \\ \frac{\partial^2 J}{\partial\alpha_2\partial\alpha_1} & \frac{\partial^2 J}{\partial\alpha_2^2} & \cdots \\ \vdots & \vdots & \end{bmatrix}_{\underline{\alpha}=\underline{\alpha}^0}}_{H} \cdot \underbrace{\begin{bmatrix} \alpha_1-\alpha_1^0 \\ \alpha_2-\alpha_2^0 \\ \vdots \end{bmatrix}}_{\underline{\alpha}-\underline{\alpha}^0} = 0 \tag{21}$$

The first step is the determination of the differentials Δ and H. This is not difficult but tedious and is therefore omitted here. Then roughly estimated parameters α are introduced in Eq. (21) and, as H is usually non-singular, improved parameters $\underline{\alpha}$ can be determined by:

$$\underline{\alpha} - \underline{\alpha}^0 = -H^{-1}_{\underline{\alpha}=\underline{\alpha}^0}\cdot\Delta_{|\alpha=\alpha^0} \tag{22}$$

These parameters are inserted as $\underline{\alpha}^0$ in Eq. (22) in a second step and an iteration can be started, provided that the evaluated parameters converge towards the optimal parameters. In this case $J(\alpha)$ (Eq. (19)) is reduced from one iteration step to the next iteration step.

Unfortunately practical applications have shown that this method very often does not converge, because of too large steps of $\underline{\alpha}$ which overshoot the optimal parameters. This difficulty can be removed by a combination of Eq. (22) with a direct search method. After the determination of $\alpha - \underline{\alpha}^0$ (Eq. (22)) $J(\underline{\alpha}^0 + n/10(\alpha - \alpha^0))$, $n = 1,2,\ldots10$ is evaluated. The parameter increment $n/10(\underline{\alpha} - \alpha^0)$ which gives the smallest value of J is used for the next iteration step. With this combined technique fast convergence can be obtained in most cases.

The example, Table 1, shows the iteration results of the velocity measurement-errors, Fig. 8 and Eq. (17). The four parameters $\sqrt{S_0}$, T_{v_4}, ω_4, ξ_4 are identified in four iteration steps. Remarkable is the reduction of J down to 2 % of the initial value.

Parameter	δ_0	T_{V_4}	ω	ξ_4	γ
Ausgangswerte	0.060	12.0	0.35	0.5	0.0732
1. Iteration	0.054	10.516	0.317	0.518	0.0103
2. Iteration	0.051	9.839	0.301	0.555	0.0019
3. Iteration	0.052	9.767	0.295	0.596	0.0015
4 Iteration	0.052	9.767	0.295	0.596	0.0015

Table 1 Evaluation of the shaping filter-parameters for the velocity measuremcnt errors.

In Fig. 9 the transfer functions of the models with the optimal identified parameters are compared with the empiric values of Fig. 8. The fitting of the shaping

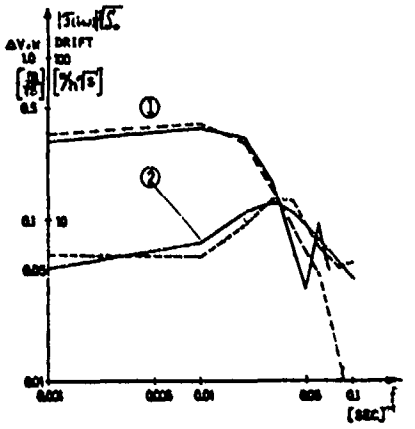


Fig. 9 Amount of transfer function of real coloured noise (empiric values) (——) and its model with optimally identified parameters (----)

① gyro drift ② velocity measurement errors.

filter is relative good in the lower frequency range. A refinement of the model for higher frequencies is not worthwhile because of the small amounts in that region.

If the curve fitting does not improve by an increasing number of iterations the structure of the shaping filter should be altered.

The determination of the shaping filter-structure and-parameters concludes the analysis of the stochastic system and measurement noise. Now the transformation of the shaping filter from the frequency- into the time-domain by Laplace or Fourier-tables is required. The resulting differential equation is included in the Kalman filter model and the filter state vector is augmented by the order of the shaping filter.

6. CONCLUSION

In the preceding sections methods and techniques for the estimation of stochastic parameters, which are necessary for the design of optimal filters were described. These methods have been derived from conventional stochastic operations and have proven remarkable effectiveness in a lot of applications [8] . The parameters are evaluated off line with a digital computer from measurements which should be stationary in respect to the system- and measurement noise. If considerable changes of the noise-parameters in dependance of deterministic parameters (e.g. aircraft velocity or weather conditions) exist, this effect can be modelled too by introduction of time-variable noise- and shaping-filter parameters.

REFERENCES

[1] Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME, Journal of Basic Engineering, Vol. 83, 1960, pp. 35-45.

[2] Gelb, A. ed.: Applied Optimal Estimation. Cambridge, Mass. and London, M.I.T. Press

[3] Sorensen, H.W. Stubbernd, A.R.: Linear Estimation Theory, Theory and Applications of Kalman-Filtering, Agardograph 139, 1970, pp. 1-42.

[4] Bryson, A.E., Johansen, D.E.: Linear Filtering for Time-Varying Systems Using Measurements Containing Coloured Noise. IEEE Transactions on Automatic Control, Vol. AC-10, 1965, pp. 4-10.

[5] Giloi, W.: Simulation und Analyse stochastischer Vorgänge. München-Wien, R. Oldenbourg, 1967.

[6] Blackman, R.B., Tukey, J.W.: Linear Data-Smoothing and Prediction in Theory and Practice. Adison & Wesley, 1965.

[7] Bekey, G.A.: System Identification - an Introduction and a Survey. Simulation, Oct. 1970, pp. 151-166.

[8] Held, V.: Die Bestimmung der wahren Lotrichtung im Flug. Dissertation, Universität Stuttgart, 1976.

REDUCED ORDER KALMAN FILTER DESIGN AND PERFORMANCE ANALYSIS

Peter S. Maybeck
Department of Electrical Engineering
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

SUMMARY

The design of an effective operational Kalman filter entails an iterative process of proposing alternative designs of different levels of complexity, "tuning" each for best attainable precision, and trading off performance capabilities and computer loading. The crux of such a design effort is establishing an adequate model upon which to base the filter - a model describing system dynamics, measurement device characteristics, and statistical properties of uncertainties associated with this structure, all of which becomes embodied in the filter algorithm. Using physical insights, order reduction and model simplification techniques, numerous prospective filters are proposed for a given application. Once this is accomplished, it is critical to be able to assess the true estimation performance of any given filter configuration when subjected to the real world environment. A systematic design procedure is described that exploits these stochastic modeling and performance analysis capabilities, and an example is used to emphasize some important aspects of the design approach.

1. INTRODUCTION

A Kalman filter is an optimal recursive data processing algorithm that accepts incomplete noise-corrupted measurements from sensors to provide an estimate of the state variables that describe the behavior of a dynamic system. It combines this real-time data with the results of stochastic modeling efforts, namely, (1) mathematical models of system dynamics and measurement device characteristics, (2) the statistical description of the system noises and disturbances, measurement errors, and uncertainties and/or inadequacies in the mathematical models themselves, and (3) any available a priori statistical information about the system states, to generate the desired state estimate. Under the assumptions that an adequate system model can be expressed in the form of a linear system driven by white Gaussian noise, its estimate can be shown to be optimal with respect to essentially any useful criterion of optimality: it is the minimum mean square error estimate, the generalized least squares estimate, the minimizer of any symmetric cost criterion, the maximum a posteriori estimate, the orthogonal projection of the true state onto the span of the measurements, the maximum likelihood estimate if there is no a priori state information (and superior to the maximum likelihood estimate if a priori statistics are available!), and, perhaps most importantly, its output totally defines the entire Gaussian conditional density function for the system state vector conditioned on the entire history of measurements that have been processed [1].

These optimality claims are impressive, but they are totally dependent upon the modeling assumptions. Mathematical models of both the system structure (state dynamics and measurement relations) and uncertainties are inherently embodied in the Kalman filter structure, and the fidelity of these models dictates the performance of the filter in actual application. Attaining an adequate mathematical model upon which to base the filter is the crux of the design problem. Thus, despite the mathematical formalism of the Kalman filter approach, a substantial amount of engineering insight, fundamental modeling capability, and experience is required to develop an effective operational filter algorithm.

Moreover, the designer typically does not have the luxury of implementing the filter based upon the best descriptive and most complete and complex model, often termed the "truth model." The final filter algorithm must meet the constraints of online computer time, memory, and wordlength, and these considerations dictate using as simple a filter as possible that meets performance specifications. Consequently, the designer must be able to exploit basic modeling alternatives to achieve a simple but adequate filter, adding or deleting model complexity as the performance needs and practical constraints require. The result is that he often generates not one, but several proposed filters of various degrees of sophistication and performance potential, and a tradeoff analysis is conducted.

Evaluation of the true performance capabilities of simplified, reduced order filters is thus of critical importance in the design procedure. Although a Kalman filter computes an error covariance matrix internally, this is a valid depiction of the true errors committed by the filter only to the extent that the filter's own system model adequately portrays true system behavior. It is very possible for the filter not to perform as well as it "thinks" it does. If the computed error covariance is inappropriately "small" (in norm, magnitude of individual eigenvalues, etc.), so is the computed filter gain: the filter discounts the data from the "real world" too much and weights its internal system model too heavily. Such a condition leads to filter state estimates not corresponding to true system behavior, with a simultaneous indication by the filter-computed covariance that the estimates are precise: filter divergence is exhibited. One significant task in the overall design process is the tuning of each proposed filter, iteratively choosing the design parameters (covariance matrix entries describing the statistics of uncertainties associated with the filter's dynamics and measurement models) that yield the best true estimation performance possible from that particular filter structure. This, in fact, is accomplished by choosing the design parameters so that the filter-computed error covariance is a good representation of the true error covariance.

The design of an effective operational Kalman filter entails an iterative process of proposing alternative designs through physical insights, tuning each, and trading off performance capabilities and computer loading. Section 2 discusses the development of numerous proposed filters for a given application. Section 3 then develops the ability to analyze the performance capability of any Kalman filter configuration operating in the real world environment. With such performance analysis available, Section 4 presents a systematic design procedure and Section 5 provides an example of exploiting these results.

2. PROSPECTIVE FILTER DESIGNS

Any prospective Kalman filter is based upon a design model of state dynamics and measurement characteristics. The design model dynamics equation is a linear stochastic differential equation for the n-dimensional state vector $\underline{x}(t)$:

$$d\underline{x}(t) = \underline{F}(t)\underline{x}(t)dt + \underline{B}(t)\underline{u}(t)dt + \underline{G}(t)d\underline{\beta}(t) \tag{1}$$

where $\underline{u}(t)$ is an r-vector of deterministic control inputs and $\underline{\beta}(t)$ is s-dimensional Brownian motion of diffusion $\underline{Q}(t)$ for all time t of interest, with statistical description given by

$$E\{\underline{\beta}(t)\} = \underline{0} \tag{2a}$$

$$E\{[\underline{\beta}(t) - \underline{\beta}(t')][\underline{\beta}(t) - \underline{\beta}(t')]^T\} = \int_{t'}^{t} \underline{Q}(t)d\tau \tag{2b}$$

where $E\{\ \}$ denotes expectation. The a priori information about the initial state $\underline{x}(t_0)$ is provided in the form of a Gaussian density specified by mean $\hat{\underline{x}}_0$ and covariance $\underline{P}_0$:

$$E\{\underline{x}(t_0)\} = \hat{\underline{x}}_0 \tag{3a}$$

$$E\{[\underline{x}(t_0) - \hat{\underline{x}}_0][\underline{x}(t_0) - \hat{\underline{x}}_0]^T\} = \underline{P}_0 \tag{3b}$$

Often Eq. (1) is written less rigorously as

$$\dot{\underline{x}}(t) = \underline{F}(t)\underline{x}(t) + \underline{B}(t)\underline{u}(t) + \underline{G}(t)\underline{w}(t) \tag{4}$$

by heuristically dividing through by dt, where $\underline{w}(t)$ is zero-mean white Gaussian noise (the hypothetical derivative of Brownian motion $\underline{\beta}(t)$) of strength $\underline{Q}(t)$:

$$E\{\underline{w}(t)\} = \underline{0} \tag{5a}$$

$$E\{\underline{w}(t)\underline{w}^T(t+\tau)\} = \underline{Q}(t)\delta(\tau) \tag{5b}$$

where $\delta(\tau)$ is the Dirac delta function.

At each sample time t_i, an m-dimensional vector of measurements $\underline{z}(t_i)$ becomes available, modeled as a linear combination of the states plus additive noise:

$$\underline{z}(t_i) = \underline{H}(t_i)\ \underline{x}(t_i) + \underline{v}(t_i) \tag{6}$$

where $\underline{v}(t_i)$ is zero-mean white Gaussian discrete-time noise with covariance $\underline{R}(t_i)$:

$$E\{\underline{v}(t_i)\} = \underline{0} \tag{7a}$$

$$E\{\underline{v}(t_i)\underline{v}^T(t_j)\} = \underline{R}(t_i)\delta_{ij} \tag{7b}$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if $i = j$; $\delta_{ij} = 0$ if $i \neq j$). The measurement corruption noise $\underline{v}(t_i)$ is usually assumed independent of the dynamics driving noise $\underline{w}(t)$ for all times t and t_i (although this is readily generalized [1]).

Once the system model has been defined by the structural parameters, i.e., the time histories of $(\underline{F},\underline{B},\underline{G},\underline{H})$, and the statistics of uncertainties, i.e., the $(\hat{\underline{x}}_0,\underline{P}_0)$ values and $(\underline{Q},\underline{R})$ time histories, the Kalman filter algorithm can be specified. Namely, the state estimate and error covariance are propagated from sample time t_{i-1} to the next sample time t_i by means of integrating

$$\dot{\hat{\underline{x}}}(t/t_{i-1}) = \underline{F}(t)\hat{\underline{x}}(t/t_{i-1}) + \underline{B}(t)\underline{u}(t) \tag{8}$$

$$\dot{\underline{P}}(t/t_{i-1}) = \underline{F}(t)\underline{P}(t/t_{i-1}) + \underline{P}(t/t_{i-1})\underline{F}^T(t) + \underline{G}(t)\underline{Q}(t)\underline{G}^T(t) \tag{9}$$

where the notation $\hat{\underline{x}}(t/t_{i-1})$ corresponds to the estimate (conditional mean) of $\underline{x}$ at time t, conditioned on measurements taken through sample time t_{i-1}, and $\underline{P}(t/t_{i-1})$ is the corresponding conditional state (and error) covariance. These are propagated from the initial conditions

$$\hat{\underline{x}}(t_{i-1}/t_{i-1}) = \hat{\underline{x}}(t_{i-1}^+)\ ,\quad \underline{P}(t_{i-1}/t_{i-1}) = \underline{P}(t_{i-1}^+) \tag{10}$$

using the results of the measurement update at time t_{i-1}, where the superscript + denotes "after measurement incorporation." Integration of (8) and (9) yields the best prediction of $\underline{x}(t_i)$ before the measurement at t_i is incorporated, denoted as $\hat{\underline{x}}(t_i^-)$, and the associated error covariance $\underline{P}(t_i^-)$:

$$\hat{\underline{x}}(t_i^-) = \hat{\underline{x}}(t_i/t_{i-1})\ ,\quad \underline{P}(t_i^-) = \underline{P}(t_i/t_{i-1}) \tag{11}$$

In fact, the solution to Eqs. (8) and (9) can be written explicitly as

$$\hat{\underline{x}}(t_i^-) = \underline{\phi}(t_i,t_{i-1})\hat{\underline{x}}(t_{i-1}^+) + \int_{t_{i-1}}^{t_i} \underline{\phi}(t_i,\tau)\underline{B}(\tau)\underline{u}(\tau)d\tau \tag{12}$$

$$\underline{P}(t_i^-) = \underline{\phi}(t_i,t_{i-1})\underline{P}(t_{i-1}^+)\underline{\phi}^T(t_i,t_{i-1}) + \int_{t_{i-1}}^{t_i} \underline{\phi}(t_i,\tau)\underline{G}(\tau)\underline{Q}(\tau)\underline{G}^T(\tau)\underline{\phi}^T(t_i,\tau)d\tau \tag{13}$$

in terms of the state transition matrix $\underline{\phi}$ associated with $\underline{F}(t)$ in Eq. (1) or (4), i.e., the solution to

$\dot{\underline{\Phi}}(t,t_0) = \underline{F}(t)\underline{\Phi}(t,t_0)$ and $\underline{\Phi}(t_0,t_0) = \underline{I}$. This form of propagation relation is especially useful for filter applications involving time-invariant system models, stationary noises, and fixed sampling period, as a replacement for direct numerical integration of (8) and (9). At sample time t_i, the measurement $\underline{z}(t_i)$ is incorporated into the estimate according to the update relations

$$\underline{K}(t_i) = \underline{P}(t_i^-)\underline{H}^T(t_i)[\underline{H}(t_i)\underline{P}(t_i^-)\underline{H}^T(t_i) + \underline{R}(t_i)]^{-1} \tag{14}$$

$$\hat{\underline{x}}(t_i^+) = \hat{\underline{x}}(t_i^-) + \underline{K}(t_i)[\underline{z}(t_i) - \underline{H}(t_i)\hat{\underline{x}}(t_i^-)] \tag{15}$$

$$\underline{P}(t_i^+) = \underline{P}(t_i^-) - \underline{K}(t_i)\underline{H}(t_i)\underline{P}(t_i^-) \tag{16}$$

Starting from the initial conditions of $\hat{\underline{x}}_0$ and $\underline{P}_0$ given by Eq. (3), this algorithm recursively processes time propagations and measurement updates. The propagations between measurement sample times inherently use the system dynamics model, Eqs (1) - (5), to provide the predicted state and error covariance. Through use of the measurement model, Eqs. (6) and (7), it also generates the best estimate of what the next measurement will be before it actually arrives. Then the measuring devices are sampled, the residual is computed as the difference between these measurements and their predicted values, and finally the filter gain (itself dependent on the structural and statistical models of (1) to (7)) optimally weights this residual to produce the updated state estimate. Because these models are embedded in the structure of the filter algorithm, the performance potential of any Kalman filter is directly a function of the adequacy of its assumed models.

A systematic design procedure will encompass the generation of alternative filter designs, each based on a particular set of models, and an evaluation of realistic performance capabilities and computer loading for each one. It is possible to devise filters based on very extensive models, and in fact it is useful to investigate the performance of the filter based on the most complete model, known as a "truth model", to establish a baseline of performance to which to compare others. However, such a filter is typically more sophisticated than required to meet performance specifications, and it is prohibitive computationally. The designer must seek simplified filter design models that retain the dominant features of the original system characteristics and provide adequate estimate precision. This is probably the most difficult aspect of designing a filter, and it requires substantial understanding of the real world system and of stochastic modeling, as well as competence in filtering theory.

Suppose a large-dimensioned, complex system model existed upon which a filter could be based that far exceeded performance requirements. Since the number of multiplications (time-consuming on a computer) and additions required by the filter algorithm is proportional to n^3 and the number of storage locations is proportional to n^2, where n is the state dimension, one significant means of decreasing the computer burden is to reduce the order by deleting and/or combining (or "aggregating") states [1-10]. There is often considerable physical insight into the relative significance of various states upon overall estimation precision, that suggests which states might be removed. States with consistently small rms value, such as those corresponding to higher order and/or higher frequency system modes which typically have lower energy associated with them, especially warrant inspection for possible removal. Other noncritical system modes might also be discarded, especially if they are only weakly observable or controllable [11,12]. An error budget performance analysis of the most complete filter, to be discussed in the next section, is an invaluable aid to this state dimension reduction.

In many applications, system models are in the form of the fundamental descriptive equations of some physical system, driven by time-correlated stochastic processes whose characteristics match those of physical phenomena such as noises, disturbances, and so forth. These, in turn, are modelled as the outputs of linear "shaping filters" [1] driven by white Gaussian noise, with dimension and defining parameters chosen so that these model outputs have statistical properties that replicate or closely approximate empirically observed means, autocorrelation functions, power spectral densities, etc. of the actual physical phenomena. Particularly in these shaping filters are substantial order reduction efforts usually made. Often a high dimensioned shaping filter in the overall "truth model" is replaced by a very low order shaping filter, such as a first order lag driven by zero-mean white Gaussian noise:

$$\dot{x}(t) = -[1/T]x(t) + w(t) \tag{17}$$

where the strength Q of the white noise w is $2\sigma^2/T$, so that the x process in steady state has mean of zero, mean squared value and variance of σ^2, autocorrelation function of $E\{x(t)x(t+\tau)\} = \sigma^2\exp(-|\tau|/T)$ so that T is the correlation time, and power spectral density function $[2\sigma^2/T]/[\omega^2 + (1/T)^2]$ so that the bandwidth of x is (1/T). The values of σ^2 and T are treated as design parameters to match the empirically observed mean squared value with σ^2 (or low-frequency power spectral density value with $2\sigma^2T$) and bandwidth with (1/T), ignoring less predominant characteristics. In fact, if the bandwidth is wide compared to the bandpass of the system driven by this noise, a zero-state trivial shaping filter might be proposed: white Gaussian noise of strength to match the low-frequency power spectral density characteristics.

It must be emphasized that deleting states and combining many states into fewer "equivalent" states must be evaluated in terms of resulting filter performance, as described in the next section. Experience has shown that reductions motivated even by the best of physical insight can sometimes degrade estimation accuracy unacceptably. Furthermore, an inappropriately reduced filter of state dimension n can often be outperformed by a filter involving fewer than n, differently chosen, states. The extreme case of this would be an unstable higher-dimensioned filter being based upon an unobservable system model in which, for instance, two states correspond to different physical variables but are indistinguishable in their effect on the model outputs, while the lower-dimensioned filter model combined states to achieve observability.

Simplification of system model matrices for a given dimension state model is also possible. Dominated terms in a single matrix element can be neglected, as in replacing the state transition matrix $\underline{\Phi}$ by $[\underline{I} + \underline{F}\,\Delta t]$ in a time-invariant or slowly varying model, ignoring higher order terms in each element. Moreover, entire weak coupling terms can be removed, producing matrix elements of zero and thus fewer required multiplications. Sometimes such removal allows decoupling the filter: a decentralized design achieved through nonsingular perturbations [13]. In numerous applications, the terms that can be ignored comprise

the time-varying nature of the model description, or at least the most rapid variations, so that a time-invariant model (or at least one that admits quasi-static coefficients) can be used for the basis of a computationally advantageous filter. Furthermore, a given problem can often be decomposed via singular perturbations [13-19] into a number of simpler nested problems, with "inner loops" operating at fast sample rates to estimate (and control) "high frequency" dynamics states (assuming slower states are random constants) and "outer loops" operating with longer sample periods to estimate (and control) slower dynamics (designed assuming that "faster" states have reached steady state conditions).

The number of multiplications and additions required by a filter algorithm can be minimized by transforming into a canonical state space representation, since the resulting system matrices embody a high density of zeroes. Obviously, one might also attempt to reduce the computational burden by increasing the sample period of the filter, if performance allows this option.

The methods discussed up to this point have involved the generation of a simplified model, with subsequent filter construction. It is also advantageous to consider approximating the filter structure itself. Because of the separability of the conditional mean and covariance equations in the filter, it is possible to precompute and store the filter gains rather than calculate them online. This precomputed filter gain history can often be approximated closely by curve-fitted simple functions, such as piecewise constant functions, piecewise linear functions, and weighted exponentials. Thus, the filter covariance and gain calculations, which comprise the majority of the computer burden, are replaced by a minimal amount of required computation and storage. Online practicality can be enhanced still further in the case of a filter based on a time-invariant system model driven by stationary noises. In many such applications, a short initial transient is followed by a long period of essentially steady state constant gain, and the approximation of using these constant gains for all time may be entirely adequate for desired performance. Of course, there are some drawbacks to using stored gain profiles. Future gains do not change appropriately when scheduled measurements are not made, due to data gaps or measurement rejection by reasonableness tests on the residuals [1]. Nor can the prestored gains adapt online to compensate for filter divergence or a system environment that is different than anticipated during the design phase [20]. Finally, lengthy simulations are usually required to determine a single gain history that will perform adequately under all possible conditions for an actual application.

3. PERFORMANCE ANALYSIS

Throughout the previous section, the critical significance of an "adequate" system model within the filter structure was stressed. To assess the capabilities of various filter designs relative to each other and to a set of performance specifications, one must have at his disposal a means of producing an accurate statistical portrayal of estimation errors committed by each filter in the "real world" environment, without actually building and testing each in the "real world". A performance analysis as depicted in Fig. 1

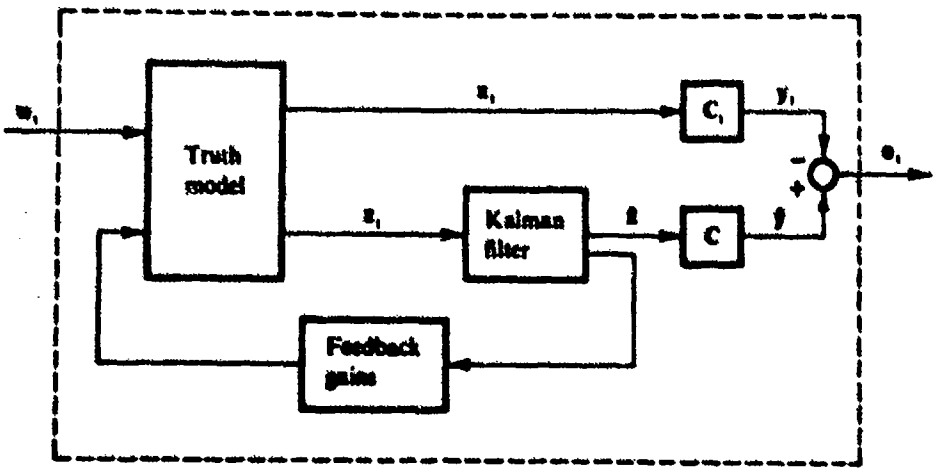


Fig. 1 Performance Evaluation of a Kalman Filter Design

fulfills this objective by replacing the "real world" measurement generation with the output of the best, most complete mathematical model that can be developed, called a "truth model" or reference model [1,21-24]. Extensive data analysis, and shaping filter design and validation are expended to ensure that this provides a very accurate representation of the "real world", since the ensuing performance evaluations and systematic design procedure are totally dependent upon this fidelity. For example, a very good generic model of the errors in an inertial navigation system (INS) has been constructed in the form of a linear model of about 70 states driven by white Gaussian noise [1,6,21,25-29]; thorough laboratory and flight testing of a particular INS allows complete specification of the model parameters to yield the "truth model" for that particular system. On the other hand, the design models used for the basis of operational aided-INS Kalman filters often represent these same characteristics with 15 or fewer states, with less but acceptable fidelity.

The truth model is composed of a stochastic differential equation for the n_t-dimensional state process $\underline{x}_t(t)$, where the subscript t denotes "truth model", and an associated output equation to generate the "true" sampled-data measurements $\underline{z}_t(t_i)$ in Fig. 1. If the truth model itself is a linear model, then it can be expressed by equations of the form of (1)-(7), but with subscript t added to all variables:

$$\dot{\underline{x}}_t(t) = \underline{F}_t(t)\underline{x}_t(t) + \underline{B}_t(t)\underline{u}(t) + \underline{G}_t(t)\underline{w}_t(t) \tag{18}$$

$$\underline{z}_t(t_i) = \underline{H}_t(t_i)\underline{x}_t(t_i) + \underline{v}_t(t_i) \tag{19}$$

where $\underline{w}_t(t)$ is zero-mean white Gaussian noise of strength $\underline{Q}_t(t)$ and $\underline{v}_t(t_i)$ is discrete-time zero-mean white Gaussian noise of covariance $\underline{R}_t(t_i)$ and independent of $\underline{w}_t(t)$; $\underline{x}_t(t_0)$ is described as Gaussian, of mean $\hat{\underline{x}}_{t0}$ and covariance $\underline{P}_{t0}$. These relations do not yet account for the feedback from the Kalman filter as depicted in Fig. 1; the required modifications will be described to simulate such feedback of filter outputs to the "real world" for control purposes. It is also possible to consider a nonlinear truth model, as

$$\dot{\underline{x}}_t(t) = \underline{f}_t[\underline{x}_t(t), \underline{u}(t), t] + \underline{G}_t(t)\underline{w}_t(t) \tag{20}$$

$$\underline{z}_t(t_i) = \underline{h}_t[\underline{x}_t(t_i), t_i] + \underline{v}_t(t_i) \tag{21}$$

with the statistical description of uncertainties given by $(\hat{\underline{x}}_{t0}, \underline{P}_{t0}, \underline{Q}_t, \underline{R}_t)$ as before. In fact, even more general Itô state stochastic differential equations can be used, as allowing $\underline{G}_t$ to be a function of $\underline{x}_t$ as well as t, to produce useful Markov state processes [1,20,30,31].

It is desired to achieve a meaningful comparison of filters that may differ substantially in state dimension and internal model specification, when each is subjected to the realistic measurement environment generated by the truth model. However, for any given application, there are certain variables of critical interest no matter which filter is under consideration, and every proposed filter must be able to provide estimates of these quantities or variables functionally related to them. For instance, in aided-INS applications, position, velocity, and often attitude variables are paramount, and any additional states in a particular design are of secondary importance. These p critical variables, denoted here as the stochastic process $\underline{y}(t)$, will serve as the basis of the performance analysis. They are assumed to be related to the filter states through a linear transformation, so that

$$\hat{\underline{y}}(t/t_{i-1}) = \underline{C}(t)\hat{\underline{x}}(t/t_{i-1}) \tag{22a}$$

is the estimate of $\underline{y}(t)$ for any $t\in(t_{i-1},t_i)$, using Eq. (8), and

$$\hat{\underline{y}}(t_i^-) = \underline{C}(t_i)\hat{\underline{x}}(t_i^-) \tag{22b}$$

$$\hat{\underline{y}}(t_i^+) = \underline{C}(t_i)\hat{\underline{x}}(t_i^+) \tag{22c}$$

are the estimates of $\underline{y}(t_i)$ before and after incorporation of the measurement $\underline{z}_t(t_i)$. Often $\underline{C}$ is time-invariant, and if its structure is $[\underline{I} \mid \underline{0}]$, then $\underline{y}$ is simply the first p of the n filter states.

As shown in Fig. 1, the truth model state process $\underline{x}_t(t)$ can also be used to generate something generally unavailable from the "real world": the true value of the quantities of interest at any time t as

$$\underline{y}_t(t) = \underline{C}_t(t)\underline{x}_t(t) \tag{23}$$

again in terms of a linear transformation represented by a p-by-n_t matrix $\underline{C}_t(t)$. Of course, Eqs. (22) and (23) can be extended to nonlinear functions instead of linear, but the linear function case is being emphasized here. Having access to the true values $\underline{y}_t(t)$, we can represent the true error committed by the particular Kalman filter in attempting to estimate the quantities of interest, when subjected to realistic measurements, as

$$\underline{e}_t(t) = \hat{\underline{y}}(t/t_{i-1}) - \underline{y}_t(t) \tag{24a}$$

$$\underline{e}_t(t_i^-) = \hat{\underline{y}}(t_i^-) - \underline{y}_t(t_i) \tag{24b}$$

$$\underline{e}_t(t_i^+) = \hat{\underline{y}}(t_i^+) - \underline{y}_t(t_i) \tag{24c}$$

over the sample period before sample time t_i, and just before and after measurement incorporation at time t_i, respectively. If impulsive feedback control from the filter into the "real world" system is admitted [1,6,21-23,27], then another error, corresponding to after both measurement updating and impulsive control application, is also of interest:

$$\underline{e}_t(t_i^{+c}) = \hat{\underline{y}}(t_i^{+c}) - \underline{y}_t(t_i^c) \tag{24d}$$

where the superscript c denotes after control is applied. The objective of a performance analysis is to characterize the true error process, Eq. (24), statistically. In a Monte Carlo analysis [1,24], many samples of the error stochastic process are produced by simulation, and then the sample statistics are

computed directly. If the truth model is itself totally linear as in Eqs. (18) and (19) and if strictly linear output relations (22) and (23) and only linear feedback are used, time histories of the statistics themselves can be computed directly in a mean and covariance analysis [1,21,23]. The relations for conducting either analysis will be developed after the "feedback gains" block of Fig. 1 is specified more precisely.

One type of feedback is impulsive control or discrete-time reset, in which quantities in the "real world" can be changed instantaneously based upon the estimate $\hat{\underline{x}}(t_i^+)$. For example, in aided-INS Kalman filters, estimates of the errors in position and velocity indications of the INS are fed back to the INS for correction by resetting contents of computer memory locations. Once $\hat{\underline{x}}(t_i^+)$ is computed, the reset control is calculated as a function (assumed linear) of it, $\underline{D}_t(t_i)\underline{x}(t_i^+)$, and after the control is applied, the truth model state becomes

$$\underline{x}_t(t_i^{+c}) = \underline{x}_t(t_i) - \underline{D}_t(t_i)\hat{\underline{x}}(t_i^+) \tag{25}$$

The filter should be "told" that this feedback to the system has occurred, so its state estimate is modified as

$$\hat{\underline{x}}(t_i^{+c}) = \hat{\underline{x}}(t_i^+) - \underline{D}(t_i)\hat{\underline{x}}(t_i^+) = [\underline{I}-\underline{D}(t_i)]\hat{\underline{x}}(t_i^+) \tag{26}$$

where the n-by-n $\underline{D}(t_i)$ models the effect of feedback through the actual n_t-by-n gains $\underline{D}_t(t_i)$ into the system. This $\hat{\underline{x}}(t_i^{+c})$ replaces $\hat{\underline{x}}(t_i^+)$ as the initial condition for the next time propagation. Some true system variables are controlled over the entire sample period rather than impulsively, which can be expressed by modifying Eq. (18) to

$$\dot{\underline{x}}_t(t) = \underline{F}_t(t)\underline{x}_t(t) - \underline{X}_t(t)\hat{\underline{x}}(t/t_{i-1}) + \underline{B}_t(t)\underline{u}(t) + \underline{G}_t(t)\underline{w}_t(t) \tag{27}$$

Again, the filter should be informed of such feedback, so (8) is changed to

$$\begin{aligned}\dot{\hat{\underline{x}}}(t/t_{i-1}) &= \underline{F}(t)\hat{\underline{x}}(t/t_{i-1}) - \underline{X}(t)\hat{\underline{x}}(t/t_{i-1}) + \underline{B}(t)\underline{u}(t)\\ &= [\underline{F}(t) - \underline{X}(t)]\hat{\underline{x}}(t/t_{i-1}) + \underline{B}(t)\underline{u}(t)\end{aligned} \tag{28}$$

Now consider Fig. 1 again: if the truth model is linear, then the entire system enclosed by the dashed lines is itself a linear system driven by white Gaussian noises. To characterize the output process $\underline{e}_t(t)$ from such a system model, one first characterizes the Gauss-Markov state process for the overall system - the augmented state process $\underline{x}_a(t)$ composed of both truth model states and filter states:

$$\underline{x}_a(t) \triangleq \begin{bmatrix}\underline{x}_t(t)\\ \hat{\underline{x}}(t/t_{i-1})\end{bmatrix} \tag{29}$$

From Eqs. (27) and (28), the augmented state process time propagation relation is

$$\dot{\underline{x}}_a(t) = \underline{F}_a(t)\underline{x}_a(t) + \underline{B}_a(t)\underline{u}(t) + \underline{G}_a(t)\underline{w}_t(t) \tag{30}$$

where

$$\underline{F}_a(t) = \begin{bmatrix}\underline{F}_t(t) & -\underline{X}_t(t)\\ \underline{0} & [\underline{F}(t)-\underline{X}(t)]\end{bmatrix}, \quad \underline{B}_a(t) = \begin{bmatrix}\underline{B}_t(t)\\ \underline{B}(t)\end{bmatrix}, \quad \underline{G}_a(t) = \begin{bmatrix}\underline{G}_t(t)\\ \underline{0}\end{bmatrix} \tag{31}$$

Solved forward from time t_{i-1} with the initial conditions

$$\underline{x}_a(t_{i-1}^{+c}) = \begin{bmatrix}\underline{x}_t(t_{i-1}^{c})\\ \hat{\underline{x}}(t_{i-1}^{+c})\end{bmatrix} \tag{32}$$

Measurement update relations are obtained by realizing that the truth model state is unaltered by a measurement,

$$\underline{x}_t(t_i^+) = \underline{x}_t(t_i^-) \tag{33}$$

and that the filter update can be written as

$$\begin{aligned}\hat{\underline{x}}(t_i^+) &= \hat{\underline{x}}(t_i^-) + \underline{K}(t_i)[\underline{z}_t(t_i) - \underline{H}(t_i)\hat{\underline{x}}(t_i^-)]\\ &= [\underline{I}-\underline{K}(t_i)\underline{H}(t_i)]\hat{\underline{x}}(t_i^-) + \underline{K}(t_i)\underline{H}_t(t_i)\underline{x}_t(t_i) + \underline{K}(t_i)\underline{v}_t(t_i)\end{aligned} \tag{34}$$

Putting these into augmented form yields

$$\underline{x}_a(t_i^+) = \underline{A}_a(t_i)\underline{x}_a(t_i^-) + \underline{K}_a(t_i)\underline{v}_t(t_i) \tag{35}$$

where

$$\underline{A}_a(t_i) = \begin{bmatrix}\underline{I} & \underline{0}\\ \underline{K}(t_i)\underline{H}_t(t_i) & [\underline{I}-\underline{K}(t_i)\underline{H}(t_i)]\end{bmatrix}, \quad \underline{K}_a(t_i) = \begin{bmatrix}\underline{0}\\ \underline{K}(t_i)\end{bmatrix} \tag{36}$$

Similarly, using Eqs. (25) and (26), the impulsive control update can be represented by

$$\underline{x}_a(t_i^{+c}) = \underline{D}_a(t_i)\underline{x}_a(t_i^{+}) \tag{37}$$

$$\underline{D}_a(t_i) = \begin{bmatrix} \underline{I} & -\underline{D}_t(t_i) \\ \underline{0} & [\underline{I}-\underline{D}(t_i)] \end{bmatrix} \tag{38}$$

If feedback is not employed, $\underline{D}_a(t_i)$ is simply an (n_t+n)-by-(n_t+n) identity matrix, so that $\underline{x}_a(t_i^{+c}) = \underline{x}_a(t_i^{+})$.

Finally, the true error committed by the filter can be expressed in terms of the augmented state vector as

$$\underline{e}_t(t) = \underline{C}_a(t)\underline{x}_a(t) \tag{39}$$

for any time t of interest, where, from Eqs. (22) - (24),

$$\underline{C}_a(t) = [-\underline{C}_t(t) \quad \underline{C}(t)] \tag{40}$$

Equations (29)-(32) and (35)-(40) are the basis of a Monte Carlo analysis, allowing simulation of many individual samples of the true error stochastic process. By taking appropriate expectations of these results, mean and covariance analysis relations can also be produced. For simplicity, assume all quantities are zero mean and concentrate attention on the covariance analysis results (these are readily generalized to the case of nonzero $\hat{\underline{x}}_0$ and $\underline{u}$). The time history of the error covariance

$$\underline{P}_e(t) = E(\underline{e}_t(t)\underline{e}_t^{T}(t)) \tag{41}$$

is then the desired output, and

$$\underline{P}_a(t) = E(\underline{x}_a(t)\ \underline{x}_a^{T}(t)) \tag{42}$$

is first computed as a means of obtaining this result. The appropriate initial conditions are

$$\underline{P}_a(t_0) = \begin{bmatrix} \underline{P}_{to} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix} \tag{43}$$

Propagating from sample time t_{i-1} to t_i is accomplished by integrating

$$\dot{\underline{P}}_a(t) = \underline{F}_a(t)\underline{P}_a(t) + \underline{P}_a(t)\underline{F}_a^{T}(t) + \underline{G}_a(t)\underline{Q}_t(t)\underline{G}_a^{T}(t) \tag{44}$$

forward from the initial condition $\underline{P}_a(t_{i-1}^{+c})$, as seen from Eqs. (30) - (32). The measurement update relation derived from Eqs. (35) and (36) is

$$\underline{P}_a(t_i^{+}) = \underline{A}_a(t_i)\underline{P}_a(t_i^{-})\underline{A}_a^{T}(t_i) + \underline{K}_a(t_i)\underline{R}_t(t_i)\underline{K}_a^{T}(t_i) \tag{45}$$

From (37) and (38), the impulsive control update is

$$\underline{P}_a(t_i^{+c}) = \underline{D}_a(t_i)\underline{P}_a(t_i^{+})\underline{D}_a^{T}(t_i) \tag{46}$$

As the time history of $\underline{P}_a$ is generated recursively using (44) - (46), the desired true error covariance can be obtained simultaneously via

$$\underline{P}_e(t) = \underline{C}_a(t)\underline{P}_a(t)\underline{C}_a^{T}(t) \tag{47}$$

as derived from (39) and (40). Because the augmented state system model is a linear system driven by white Gaussian noise, the covariance relations are not coupled to the actual measurement history realizations, so it is possible to perform this covariance analysis a priori, without resorting to explicit simulation of measurement process samples.

Although a covariance analysis is computationally more efficient than a Monte Carlo study and so should be exploited, especially in initial design phases, there are advantages to using the Monte Carlo approach in addition. First, a Monte Carlo study encompasses a system simulation in which the actual, entire filter algorithm is embedded. As such, portions of the simulation can be replaced by actual data or hardware as it becomes available in the system evolution. Moreover, sign errors in the filter algorithm that may not be readily apparent in a covariance analysis due to squaring effects become evident from Monte Carlo and covariance analyses based on the same models disagreeing with each other. Finally, effects of nonlinearities such as device saturation or neglected terms in attaining linear perturbation equations cannot be evaluated by a covariance analysis except in an approximate manner based on describing functions [22], and a full investigation requires a Monte Carlo analysis.

4. USE OF PERFORMANCE ANALYSIS IN DESIGN

Once a performance analysis capability is established, a systematic iterative design and tradeoff of proposed filters can be conducted. First of all, performance analysis allows proper filter tuning [1,6,22-24,27,30,32-35]. The basic objective of tuning is to achieve the best possible estimation accuracy from a proposed filter by selection of filter design parameters of $\underline{P}_0$ and the time histories of $\underline{Q}$ and $\underline{R}$ (see Eqs. (3), (5), and (7)). Basically, $\underline{P}_0$ is the determining factor in the initial transient performance of the filter, whereas the $\underline{Q}$ and $\underline{R}$ histories dictate the longer term or steady state performance and

time duration of transients. These covariances not only account for actual noises and disturbances in the physical system, but also are a means of declaring how adequately the filter-assumed model represents the "real world" system. Therefore, the simpler and less accurate the model, the stronger the noise strengths should be set (through addition of "pseudonoise" to the noises associated with true physical disturbance phenomena). However, it is difficult if not impossible to declare best parameter values a priori, and the specification is usually the result of an iterative search.

When tuning a filter, it is useful to compare the actual estimation error statistics, as provided by $P_e(t)$, to the filter's own representation of its error statistics through its internally computed covariance matrix given by Eqs. (9), (13), (14), and (16). Superimposed plots of "true" and filter-computed root mean square errors in estimating individual quantities of interest, as depicted in Fig. 2, are an

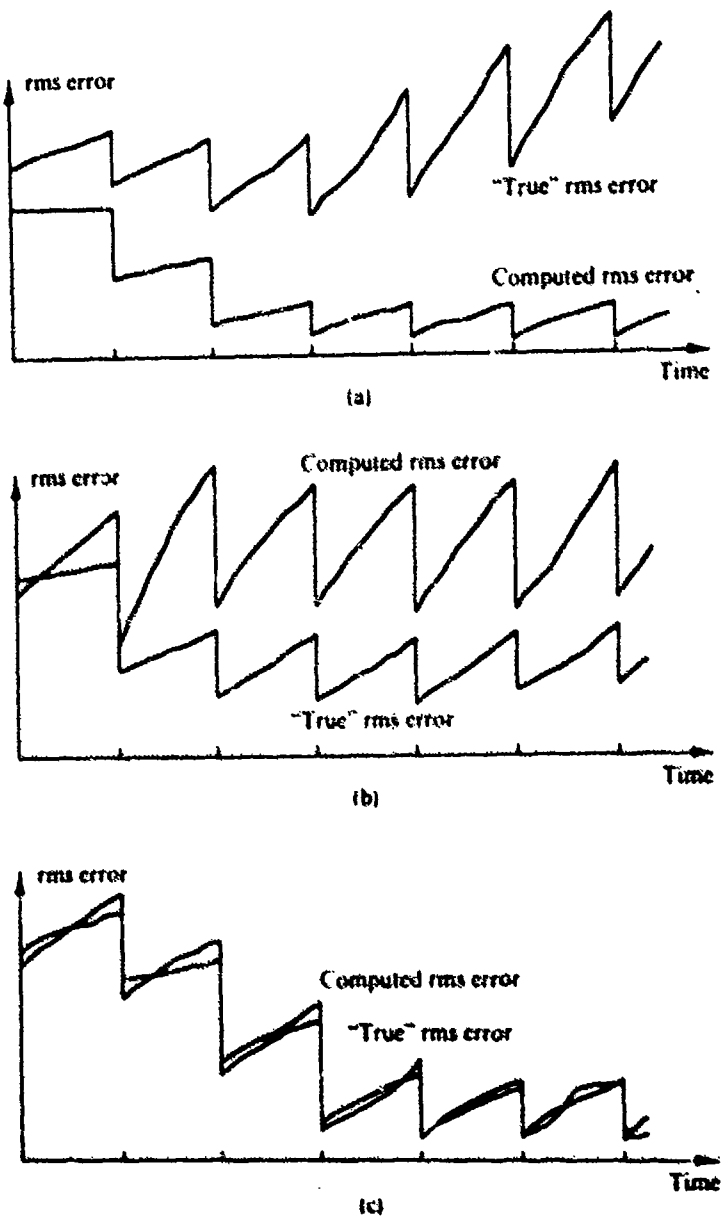


Fig. 2 Filter Tuning Through Performance Analysis. (a) Filter underestimates its own errors: divergence. (b) Filter overestimates its own errors: tracking of measurement noise. (c) Well-tuned filter.

invaluable aid to tuning and are often provided interactively by performance analysis software [23,24]. If, due to mistuned noise parameters such as attributing inappropriately small uncertainty to the internal dynamics model, the filter underestimates its own errors as in Fig. 2a, the filter "believes" its model output too much and does not weight the measurement information heavily enough. This can cause filter divergence [1,22,36-38] if the discrepancy is significant enough. On the other hand, if inappropriately small uncertainty and noise corruption is associated with the measurements, the filter overestimates its own errors and weights the measurements too heavily, expending too much effort tracking the noisy data and not exploiting the benefits of its internal model enough, as in Fig. 2b. By choosing the noise parameters so that the overall time histories of actual and filter-computed rms errors match well, the actual rms errors are effectively reduced at the same time, as seen in Fig. 2c. Allowing the filter to overestimate its own errors slightly, thereby guarding against divergence, is a commonly adopted means of generating a "conservative" or robust filter design, able to withstand modeling inaccuracies without becoming divergent [39]. Game theoretic minimax approaches [40] have also been used to design filters with acceptable performance over an entire range of uncertain parameter values. Filters are also purposely "robustified" in control applications in order to maintain closed loop system stability despite large variations in system parameters or operating conditions from those assumed during the design phase [41]. Another more formal result than iterative pseudonoise adjustment is provided by the concept of a minimum variance reduced order estimator [42,43], but again the performance analysis should be used to indicate the true capabilities of such a tuning approach. Furthermore, both offline tuning and online adaptation can be enhanced by exploiting the fact that the residuals of a well-tuned Kalman filter based on an adequate model should be a zero-mean, white Gaussian discrete-time process with covariance of $[\underline{H}(t_i)\underline{P}(t_i^-)\underline{H}^T(t_i) + \underline{R}(t_i)]$ as computed in the filter in Eq. (14)[1,20,22].

Once a particular filter has been tuned, an error budget [1,22,33-35] can be established. This is a depiction of the contributions of individual error sources to overall estimation errors, consisting of repeated performance analyses in which single or small groups of error sources in the truth model are "turned on" individually. If the filter under test were based upon the full-scale truth model, such an error budget would suggest potential choices of states to neglect in a reduced-order design (increasing the strength of appropriate noises to account for such deletion) that would yield the least performance degradation. On the other hand, if the filter under test were a proposed practical design and tuned properly, this error budget would indicate dominant sources of error, which may warrant either better models for error compensation within the filter or better system hardware (i.e., changes in the truth model) if better performance is sought. Sensitivity of estimation precision to parameter variations in the filter or system hardware can be obtained in a straightforward manner by repeated covariance analyses embodying the parameter changes, or adjoint methods [44] can be used to describe local sensitivity to small variations in many parameters simultaneously.

A systematic design procedure would first entail developing a "truth model" to portray actual system behavior very accurately, as validated with laboratory and operational test data; if it is nonlinear, it should be linearized about an appropriate nominal for later covariance analyses. The Kalman filter based upon the "truth model" is generated as a benchmark of performance; for this filter, there is no "tuning" to be accomplished and the filter-computed covariance is the desired true error covariance. Simplified, reduced order system models are then proposed by deleting and combining states associated with nondominant effects, removing weak coupling terms, employing approximations such as constant gains, and the like - this part of the design effort requires substantial physical insights into the problem at hand. Then a covariance performance analysis of each proposed Kalman filter is conducted; as an iteration within this step, each filter is tuned to provide best possible performance from each. A Monte Carlo analysis of the most promising designs is generated, as is a performance/computer loading tradeoff analysis to select a final design. This chosen design is then implemented on the online computer to be used in the actual system; for numerical stability and numerical precision of the online filter at modest wordlength, this implementation is best accomplished in square root or U-D covariance factorization form [1,43,45-47]. Finally, checkout, any required final tuning, and operational test of the filter is performed. Even in this last phase, performance analyses can be used to investigate and extend the results observed in online filter operation [48]. Through such a design procedure, a logical decision process based on sufficient empirical data is incorporated into the filter implementation.

5. EXAMPLE

Currently the Air Force is developing tactical weapon systems that will afford precision standoff delivery of ordnance. One such system is a glide vehicle with midcourse and terminal navigation and guidance accomplished through use of a strapdown inertial navigation system (INS) aided by a radiometric area correlator (RAC).

Two different low-cost strapdown inertial systems are competing for implementation. Although both use conventional accelerometers to measure specific force, one INS employs laser gyroscopes to measure angular rates, while the other uses conventional dry gyros. This difference will be seen to have a significant impact on the two systems' error characteristics and on Kalman filter performance capabilities.

As the glide vehicle flies a desired trajectory, the RAC provides a number of accurate position fixes by correlating a radiometric "picture" of the terrain immediately below the vehicle with a prestored reference map of that region. The number of such fixes is limited by the amount of computer memory allotted for the reference maps, five or six being a practical upper bound.

A Kalman filter was designed [49] to combine the information received from the INS and RAC, to estimate the errors being committed by the INS, and to feed back corrective signals to remove these estimated errors. Because of the restricted amount of computer memory allocated to the Kalman filter (less than 1000 words), the proposed design is very simple - two decoupled three-state filters. However, the adequacy of such a simple design to meet performance specifications was subject to significant question, so a covariance analysis has been conducted to determine estimation precision capabilities in a realistic environment [50].

Certain aspects of this study, such as RAC performance characteristics and a detailed portrayal of the glide vehicle trajectory (basically a nonmaneuvering glide with terminal pitchover and descent), are not

available for public release at this time. For this reason, the analysis results are presented in the form of percentages or unscaled graphs.

The proposed design is actually composed of two decoupled, three-state Kalman filters, each maintaining estimates of the INS position, velocity, and attitude angle error states along a single coordinate direction (east and north are the chosen axes). Thus, the six state variables being estimated are δx_e, δx_n - east and north components of the error in the INS-indicated position; δv_e, δv_n - errors in INS-indicated velocity; ϕ_e, ϕ_n - errors in INS-indicated attitude (tilts).

The dynamics model upon which the filters are based is

$$\begin{bmatrix} \dot{\delta x}_e \\ \dot{\delta v}_e \\ \dot{\phi}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -g \\ 0 & 1/R_0 & 0 \end{bmatrix} \begin{bmatrix} \delta x_e \\ \delta v_e \\ \phi_n \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_{e1} \\ w_{e2} \end{bmatrix} \tag{48}$$

and a similar model for δx_n, δv_n, and ϕ_e. In this equation, g is the magnitude of gravity and R_0 is the equatorial radius of the Earth. The driving term w_{e1} is a white Gaussian noise to model acceleration-level errors, and w_{e2} is an independent Gaussian noise to model errors associated with attitude error angular rates. This equation is in the form of Eq. (4) with constant $\underline{F}$ and $\underline{G}$ and $\underline{B} \equiv \underline{0}$, i.e., a linear, constant coefficient, stochastic differential equation driven by stationary zero-mean white Gaussian noise with constant strength $\underline{Q}$ in (5b).

The sampled-data measurements made available to the filter algorithms are generated by differencing INS-indicated position and RAC-indicated position. For instance, INS-indicated east position would be the true position plus the error state δx_e:

$$x_{e\text{-INS}} = x_{e\text{-TRUE}} + \delta x_e \tag{49}$$

whereas the RAC-indicated east position would be modeled as the true position corrupted by white Gaussian noise v:

$$x_{e\text{-RAC}} = x_{e\text{-TRUE}} - v \tag{50}$$

Differencing Eqs. (49) and (50) at sample time t_i yields

$$\begin{aligned} z(t_i) &= x_{e\text{-INS}}(t_i) - x_{e\text{-RAC}}(t_i) \\ &= \delta x_e(t_i) + v(t_i) \\ &= [1\ 0\ 0] \begin{bmatrix} \delta x_e(t_i) \\ \delta v_e(t_i) \\ \phi_n(t_i) \end{bmatrix} + v(t_i) \end{aligned} \tag{51}$$

This equation and the corresponding result for the difference of north position indications are of the form of Eq (6) with constant $\underline{H}$ and v being a scalar discrete-time zero-mean white Gaussian noise with time-varying autocorrelation as in (7).

Formulation of a viable system representation as a pair of independent three-state models depends upon insights and assumptions closely tied to the structure of both the full-scale INS error model and the measurement error model. First, the generally accepted nine-state model of INS error characteristics, in which position, velocity, and attitude error about three coordinate directions (east, north, up) are totally intercoupled [25-29], can be partitioned according to the state subsets (δx_e, δv_e, ϕ_n), (δx_n, δv_n, ϕ_e), ϕ_u, and (δx_u, δv_u). The last set is only weakly coupled to the first seven states, and in fact is totally decoupled from them if the vehicle were at rest and the Earth were nonrotating. Moreover, the errors in this vertical channel are kept suitably small with altimeter aiding external to the filter, so these two states are ignored.

Under the same assumption of the vehicle at rest on a nonrotating Earth, the first three sets of states also decouple, with the first two being characterized by Schuler oscillations and depicted as in Eq. (48). When the vehicle-centered east-north-up frame moves over a rotating Earth, response mode modification and intercoupling occur. But the error oscillations are still Schuler dominated since the Schuler angular rate $\sqrt{g/R_0}$ is significantly greater than coupling terms on the order of Earth rate Ω relative to inertial space and vehicle position angular rate relative to the Earth (i.e., velocity component divided by R_0); and Schular rate squared dominates vehicle position angular accelerations. Cross coupling occurs among the three attitude error states predominantly due to nonzero Earth rate and vehicle velocity, and the rate of change of velocity error states couple into those attitude errors through nonzero specific force. For instance, terms to be added to the right-hand side of Eq. (48) include

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -(f_u - g) & f_n \\ -\omega_u & 0 & \omega_e \end{bmatrix} \begin{bmatrix} \phi_e \\ \phi_n \\ \phi_u \end{bmatrix}$$

where f_u and f_n are specific force components, and ω_u and ω_e are components of angular rate of the east-north-up frame with respect to inertial space; additional cross-coupling effects also develop among the nine error states. The uncertainties w_{e1} and w_{e2} in Eq. (48) partially account for these neglected terms, though in a rather crude manner. Were it not for the severe computer memory restriction, more explicit incorporation of these effects and inclusion of the azimuth error state ϕ_u would warrant attention as a means of enhancing performance. Especially for the application envisioned here, in which flight is along a rather benign glide trajectory, the simplistic model given in Eq. (48) might well suffice.

Decoupling into two separate filters also requires that the measurements introduce no nonnegligible cross coupling. In fact, the measurement gradient matrix $\underline{H}$ associated with the full-scale INS error model does not intercouple the two horizontal channels with each other or the vertical channel. Moreover, if $v_1(t_i)$ and $v_2(t_i)$ are the measurement noises associated with east and north position differences, as in Eqs. (51) and (7), then the equiprobability ellipsoids for the vector $[v_1(t_i) \quad v_2(t_i)]^T$ must either have their principal axes aligned with the reference coordinate directions or be (nearly) circular, so that there is no cross correlation to intercouple the two horizontal channels. This poses no difficulty for this particular problem.

The Kalman filter based upon a model described by Eqs. (48)-(51) is given by Eqs. (8)-(16). To specify the filter algorithm completely, the dynamic noise strength $\underline{Q}$ in Eqs. (5), (9), and (13), the measurement noise strength time history $R(t_i)$ in Eqs. (7) and (14), and the initial state covariance $\underline{P}_0$ in Eq. (3) must be established for both filters. Finding the best such values iteratively through a process of filter tuning will be discussed subsequently.

In actual operation, the filter (the two 3-state filters can be viewed as a single decoupled 6-state filter) is driven by measurements $\underline{z}(t_i)$ from INS and RAC hardware, and provides estimates of the states, $\hat{\underline{x}}(t_i^+)$, which are used as corrective feedback to the INS. For analysis purposes, the "real world" environment is replaced by a "truth model," as in Fig. 1, and the quantities of interest $\underline{y}$ are the six states being estimated.

The truth model required in the performance analysis can be described by (18) and (19) with $\underline{B}_t = \underline{0}$, $\underline{G}_t = \underline{I}$, and $\underline{H}_t$ constant. Table 1 presents the 46 states of the truth model for the laser gyro system

Table 1 Truth Model State Description

State	Laser gyro INS system P_{t0} term	Laser gyro INS system Q_t term	Conventional gyro INS system P_{t0} term	Conventional gyro INS system Q_t term
Basic INS:				
Position errors (3)	$(1500\ \text{ft})^2$	0	$(1500\ \text{ft})^2$	0
Velocity errors (3)	$(2\ \text{ft/s})^2$	0	$(2\ \text{ft/s})^2$	0
Attitude errors (3)	$(0.5\ \text{millirad})^2$	$7.6\times10^{-11}\ \text{rad}^2/\text{s}$	$(0.5\ \text{millirad})^2$	0
Accelerometers:				
Accelerometer biases (3) (day-to-day nonrepeatability)	$(250\mu\text{g})^2$	0	$(200\mu\text{g})^2$	0
Accelerometer scale factor errors (3)	$(500\ \text{ppm})^2$	0	$(405.6\ \text{ppm})^2$	0
Accelerometer input axis misalignments (6)	$(10\ \text{arc-s})^2$	0	$(30\ \text{arc-s})^2$	0
Accelerometer biases (3) ($T_1 = 60$ min)	$(40\mu\text{g})^2$	$2P_{t0}/T_1$	$(60\mu\text{g})^2$	$2P_{t0}/T_1$
Accelerometer biases (3) ($T_2 = 15$ min)	$(20\mu\text{g})^2$	$2P_{t0}/T_2$	$(30\mu\text{g})^2$	$2P_{t0}/T_2$
Gravity knowledge:				
Gravity deflections (2) ($d_1 = 10$ n. mi.)	$(26\mu\text{g})^2$(east) $(17\mu\text{g})^2$(north)	$3P_{t0}v/d_1$	$(26\mu\text{g})^2$(east) $(17\mu\text{g})^2$(north)	$2P_{t0}v/d_1$
Gravity anomaly (1) ($d_2 = 60$ n. mi.)	$(35\mu\text{g})^2$	$2P_{t0}v/d_2$	$(35\mu\text{g})^2$	$2P_{t0}v/d_2$
Gyros:				
Gyro drift rate biases (3)	$(0.09\ \text{deg/h})^2$	$1.47\times10^{-18}\ \text{rad}^2/\text{s}^3$	$(2.0\ \text{deg/h})^2$(roll axis) $(1.33\ \text{deg/h})^2$(other two)	0
Gyro scale factor errors (3)	$(100\ \text{ppm})^2$	0	$(500\ \text{ppm})^2$	0
Gyro input axis misalignments (6)	$(6\ \text{arc-s})^2$	0	$(30\ \text{arc-s})^2$	0
Gyro drift rate (3) ($T_1 = 60$ min)	...	...	$(0.4\ \text{deg/h})^2$(osv) $(0.6\ \text{deg/h})^2$(isv)	$2P_{t0}/T_1$
Gyro drift rate (3) ($T_2 = 15$ min)	...	...	$(0.2\ \text{deg/h})^2$(osv) $(0.3\ \text{deg/h})^2$(isv)	$2P_{t0}/T_1$
g-sensitive drift coefficients (6)	...	...	$(2.0\ \text{deg/h/g})^2$	0
g^2-sensitive drift coefficients (3)	...	...	$(0.1\ \text{deg/h/g}^2)^2$	0
RAC:				
RAC biases (2)	P_{bx}(along-track) P_{by}(cross-track)	0	P_{bx}(along-track) P_{by}(cross-track)	0
Altimeter:				
Altimeter bias (1) ($d = 250$ n. mi.)	$(500\ \text{ft})^2$	$2P_{t0}v/d$	$(500\ \text{ft})^2$	$2P_{t0}v/d$
Altimeter scale factor error (1)	$(0.03)^2$	0	$(0.03)^2$	0

[29,50-52] and the 61 states corresponding to the conventional gyro system [29,50]. Associated with each state are its initial variance (appropriate $\underline{P}_{t0}$ diagonal term) and white driving noise strength ($\underline{Q}_t$ diagonal term); both $\underline{P}_{t0}$ and $\underline{Q}_t$ are assumed to be diagonal. For states that are modeled as random bias processes (the outputs of undriven integrator shaping filters), the appropriate $\underline{P}_{t0}$ term is given and the $\underline{Q}_t$ term is given as zero. For states that are modeled as first order Markov processes (outputs of first-order lags driven by white noise), the $\underline{Q}_t$ term is described in terms of the $\underline{P}_{t0}$ term and correlation time T in such a manner as to yield stationary processes.

The first nine states are the variables used to describe the error characteristics of an INS. Although each INS under consideration is a strapdown system, this error model can be expressed conveniently with respect to an east-north-up coordinate frame [25,29]. The $\underline{Q}_t$ terms associated with attitude errors are due to gyro drift and will be discussed subsequently.

Accelerometer errors are described by means of a day-to-day nonrepeatability bias, scale factor error, two input axis misalignments, and two first-order Markov process states for each accelerometer. Uncertainty in the knowledge of gravity also enters the truth model state equations at the acceleration level. The errors between the true geoid and the assumed ellipsoid for INS navigation computations have been described by means of first-order Markov process models [29], with mean square values and correlation distances as described in Table 1. If a correlation distance is denoted as d and the vehicle velocity magnitude as v, a corresponding correlation time is generated as T = d/v, thereby yielding the $\underline{Q}_t$ term expression in the table.

Gyro errors are depicted by a drift rate bias state (or Brownian motion state for the laser gyro, i.e., the output of an integrator driven by white Gaussian noise), scale factor error, two input axis misalignments, two first-order Markov process states, two g-sensitive drift coefficients (spin and input axes), and one g^2-sensitive drift coefficient (major spin-input coefficient) for each gyro. For the laser gyros, only the first four of these nine states are included, since the others are essentially nonexistent. Another marked difference from conventional gyros is embodied in the drift rate model. A typical gyro drift rate model is composed of the sum of first-order Gauss-Markov components with an additive white Gaussian noise. In conventional gyros, the time-correlated contributions dominate the very wideband (white) component, and the latter is often neglected. However, for laser gyros, the wideband (modeled as white) component predominates; its noise strength is given by the $\underline{Q}_t$ terms driving INS attitude errors in Table 1. A final difference of the two gyro types is the set of multiple table entries for certain conventional gyro states. On the Markov process states, oav denotes output axis vertical, while iav means input axis vertical. The roll axis gyro drift rate bias entry is higher than the others because a different gyro design is employed to withstand and indicate the larger range of rates that can occur about this axis. In the laser gyro INS, the gyro sensitive axes are canted off from the vehicle body axes to distribute high roll rates among three identical gyros.

Although Table 1 shows accelerometer errors to be very similar in the two inertial systems, the gyro characteristics are significantly worse in the conventional gyro INS. The low-frequency power spectral density value of the Gauss-Markov drift rate components in the conventional gyro is three orders of magnitude worse than the laser gyro white noise component. Moreover, drift rate biases, scale factor errors, and misalignments are considerably greater; and the g and g^2 errors have no counterpart in the laser gyro system.

The errors in the RAC data are modeled as a corruptive white Gaussian noise plus a bias. This is a necessarily unsophisticated model of RAC error characteristics, since only sparse and incomplete performance data were available at time of truth model development. Nevertheless, these data were sufficient to estimate appropriate noise strengths and to indicate that bias effects were not negligible. The strength of the two-dimensional white noise, $\underline{v}_t$ in Eq. (19), was found to be well modeled as

$$\underline{R}_t(t_i) = [\theta \cdot h(t_i)]^2 \underline{I} \tag{52}$$

where $h(t_i)$ is the vehicle altitude and θ is a parameter with classified numerical value. Each bias was modeled as a random constant with mean zero and variance as shown in Table 1, again the numerical values being classified. Although physical reasoning could lead to altitude-dependent variances on the bias states as well, the available data were not consistent or complete enough to warrant this formulation. Because high statistical confidence could not be placed in this model, a study of performance sensitivity to bias model parameter variations was deemed essential; this will be discussed further in the analysis presentation.

Finally, the altimeter errors are described in terms of a first-order Markov process noise plus a scale factor error. The altimeter is used to damp out the inherently unstable vertical errors in the INS, and so its errors drive certain INS error states in the truth model.

The covariance analysis technique was first used to tune the proposed filter for use in each of the two INS/RAC system configurations [50]. The filters' $\underline{P}_0$ and time histories of $\underline{Q}$ and R were iteratively modified to yield minimum rms values of the estimation error $\underline{e}_t$ components for all time of interest. For this application, terminal position errors are especially important, but the entire history of all errors must be considered to preclude being outside the bounds of a prestored RAC map at an update time and to insure sending proper corrective control commands during the terminal phase of flight.

Figure 3 plots the rms error (in log scale) in the east position estimate provided by the filter tuned to the laser gyro system. To aid the tuning process, these "actual" rms errors were compared with the filter's own representation of its errors - its own computed covariance $\underline{P}$. Despite the simple filter form and the fact that a constant $\underline{Q}$ is used for all time, the filter-computed rms error history essentially duplicates the results shown in Fig. 3. Moreover, this condition does effectively yield the best estimate precision. The results for the other five filter states, and those for the conventional gyro system, are very similar.

For computational simplicity, it was proposed to approximate the integral term in Eq. (13) by a diagonal matrix [49]. The original such design was found to be severely out of tune, and even the best tuning achievable with a diagonal matrix form yielded noticeably degraded performance. The degradation was naturally least in the channels for which direct measurements were available, i.e., position errors, and these are the estimates of primary interest for this application. However, the computation of three off-diagonal terms in a symmetric 3x3 matrix is not burdensome. Moreover, a followup study has indicated a substantial increase in importance of these off-diagonal terms for obtaining good performance along more highly dynamic trajectories with optimized measurement sample times. Therefore, weapon system development and testing was pursued with the design changed to incorporate these terms.

An error budget was generated to depict the contributions of individual error sources to the rms errors throughout the vehicle flight. Once the filter was tuned, repeated covariance analyses were conducted, each with a single error source removed. Table 2 presents the results for rms position errors at the terminal time. From this table, it is evident that the RAC errors have the greatest influence on estimate

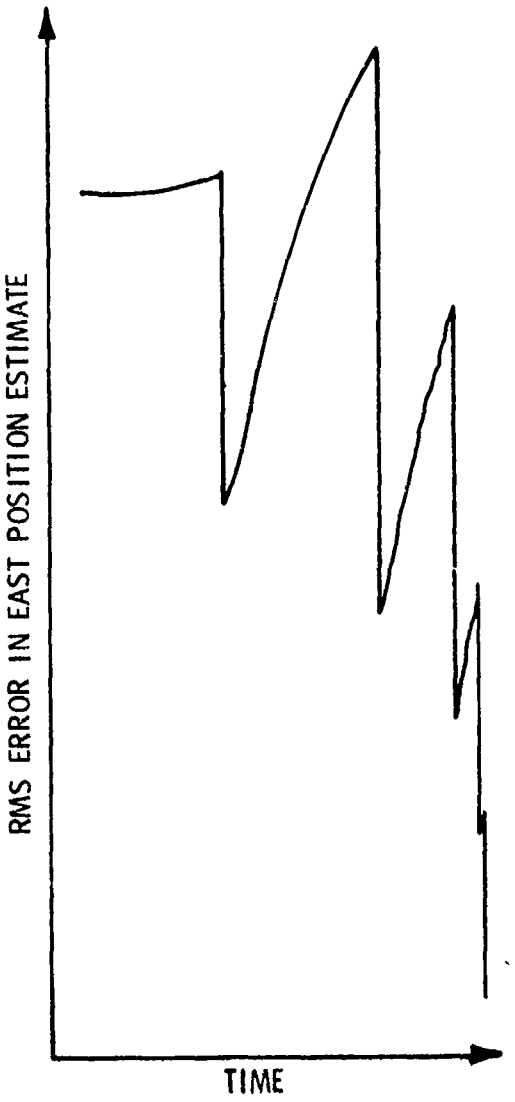


Fig. 3 RMS Error in East Position Estimate

precision at the terminal time. This is caused by the extreme accuracy of low-altitude RAC position fixes and the fact that the last two fixes are taken shortly before the end of flight to maximize the benefit of the limited number of updates. Error budgets for estimation errors earlier in the flight reveal an increased importance of INS sensor errors.

Table 2 also reveals that the laser gyro INS configuration outperforms the conventional gyro system, as would be predictable from relative precision of instruments as described in Table 1. Also, the white noise gyro drift rate model in the filters is appropriate for a laser gyro, whereas a first-order Markov process model, requiring an additional state per filter, would be a significantly better model for a conventional gyro. The table also shows that the gyro errors in the conventional INS system play a more dominant relative role in degrading performance than the same errors do in the laser gyro INS. These trends are accentuated at earlier times in the flight, especially in the case of dynamic trajectories.

Because of the significance of RAC errors and the sparse amount of test data concerning bias errors in this device, the sensitivity of estimation accuracy to varying bias levels was analyzed. Table 3 demonstrates the effect of varying the RAC bias variance from zero to four times the value listed in Table 1. These results and those depicted in Table 2 reveal that, if performance requirements are not met, seeking a better RAC system would be more beneficial than improving the INS precision. Similarly, if the filter complexity could be increased, it would be most advantageous to incorporate a better model for the errors in the RAC system position data.

Direct estimation of RAC biases by adding a fourth state to each filter is not feasible: adding the model $\dot{b} = w_b$ to Eq. (48), and modifying Eq. (51) to let z be $(\delta x_e + b + v)$, yields an unobservable system model. Basically, the filter would not be able to provide valid estimates of δx_e and b separately. Improved estimation performance can be achieved by replacing Eq. (52) with

$$\underline{R}(t_i) = ([\theta \cdot h(t_i)]^2 + R_b(t_i))\underline{I} \qquad (53)$$

Table 2 Error Budget

Error source removed	% of terminal rms nav errors: Laser gyro INS	Conventional INS
None (baseline)	100	100 (= 107.5% of laser error)
Accel errors	100	99.9
Gyro errors	100	98.1
Initial condition	100	100.0
RAC bias	95	96
All RAC errors	9	11

Table 3 Sensitivity to RAC Bias

RAC bias model standard deviation	% of terminal rms nav errors: Laser gyro INS	Conventional INS
0	95	96
standard	100	100
2 × standard	113	119

in the filter formulation, where R_k scales with the variance of the RAC bias in the truth model. Although bias errors are not directly compensated, the better model for rms measurement errors yields more proper weighting of update information, and thus enhanced performance. However, these conclusions are based upon the adequacy of the truth model depiction of RAC errors. Once enough performance data can be analyzed to have statistical confidence in the RAC error model, the preceding modification and other means of enhancement can be fully exploited in the operational filter.

Thus, because of severe restrictions on computer time and memory allotted, a very simple Kalman filter was designed to update a strapdown inertial system with position fixes from a radiometric area correlator. Nevertheless, its performance has been analyzed and found to meet specifications.

6. CONCLUSIONS

Attaining an adequate model upon which to base a Kalman filter is an essential aspect of designing an operational online filter algorithm. Numerous methods have been presented for generating models and filters of substantially different levels of complexity, state dimension, and performance potential. Another integral part of a systematic design approach is the realistic evaluation of any proposed filter's performance in estimating quantities of interest when subjected to the real world environment. Such performance analysis capability has also been described, and once again the adequacy of modeling efforts, here in the form of producing a "truth model" that accurately depicts the real world regardless of its required complexity, is shown to be an issue of primary importance. The design process itself is composed of iteratively proposing alternative filter designs, tuning each for best performance admitted by its structure, evaluating error budgets and sensitivities to parameter variations, and trading off performance capabilities and computer loading to yield the final algorithm for implementation.

ACKNOWLEDGMENTS

Figs. 1 and 2 are from Ref. [1] and are used with permission of Academic Press. Fig. 3 and the tables of Section 5 are from [50], with permission from the AIAA Journal of Guidance and Control.

REFERENCES

[1] Maybeck, P. S., Stochastic Models, Estimation and Control, Vol. 1, Academic Press, New York, 1979.

[2] Hirzinger, G., and G. Kreisselmeier, "On Optimal Approximation of High-Order Linear Systems by Low-Order Models," Int. J. Control, Vol. 22, No. 23, pp 399-408, 1975.

[3] Larson, V., and P. W. Likins, "Optimal Estimation and Control of Elastic Spacecraft," in Control and and Dynamic Systems, Advances in Theory and Applications (C. T. Leondes, ed.), Vol. 13, Academic Press, New York, 1977.

[4] Likins, P. W., Y. Ohkami and C. Wong, "Appendage Modal Coordinate Truncation in Hybrid Coordinate Dynamic Analysis," J. Spacecraft, Vol. 13, No. 10, pp 611-617, Oct. 1976.

[5] Schmidt, G. T., "Linear and Nonlinear Filtering Techniques," in Control and Dynamic Systems (C. T. Leondes, ed.), Vol. 12, Academic Press, New York, 1976.

[6] Schmidt, G. T. (ed.), Practical Aspects of Kalman Filtering Implementation, AGARD-LS-82, NATO Advisory Group for Aerospace Research and Development, London, May 1976.

[7] Simon, K. W., and A. R. Stubberud, "Reduced Order Kalman Filter," Int. J. Control, Vol. 10, pp 501-509, 1969.

[8] Skelton, R. E., P. C. Hughes and H. Hablani, "Order Reduction for Models of Space Structures Using Modal Cost Analysis," to appear in AIAA J. Guid. and Cont., Special Issue on Large Space Structure, 1981.

[9] Skelton, R. E., and P. W. Likins, "Techniques of Modeling and Model Error Compensation in Linear Regulator Problems," in Advances in Control and Dynamic Systems (C. T. Leondes, ed.), Vol. 14, Academic Press, New York, 1978.

[10] Tse, E. C. Y., J. V. Medanic and W. R. Perkins, "Generalized Hessenberg Transformations for Reduced Order Modeling of Large Scale Systems," Int. J. Control, Vol. 27, No. 4, pp 493-512, 1978.

[11] Klema, V. C., and A. J. Laub, "The Singular Value Decomposition: Its Computation and Some Applications," IEEE Trans. Automat. Control, Vol. AC-25, No. 2, pp 164-176, April 1980.

[12] Moore, B. C., "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," IEEE Trans. Automat. Control, Vol. AC-26, No. 1, pp 17-32, Feb 1981.

[13] Sandell, N. R., Jr., P. Varaiya, M. Athans and M. G. Safonov, "Survey of Decentralized Control Methods for Large Scale Systems," IEEE Trans. Automat. Control, Vol. AC-23, No. 2, pp 108-128, April 1978.

[14] Balas, M. J., "Observer Stabilization of Singularly Perturbed Systems," AIAA J. of Guid. and Cont., Vol. 1, No. 1, pp 93-95, Feb 1978.

[15] Calise, A. J., "A New Boundary Layer Matching Procedure for Singly Perturbed Systems," IEEE Trans. Automat. Control, Vol. AC-23, No. 3, pp 434-438, June 1978.

[16] Haddad, A. H., and P. V. Kokotovic, "Stochastic Control of Linear Singularly Perturbed Systems," IEEE Trans. Automat. Control, Vol. AC-22, No. 5, pp 815-821, Oct 1976.

[17] Kokotovic, P. V., R. E. O'Malley, Jr., and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory - An Overview," Automatica, Vol. 12, pp 123-132, 1976.

[18] Rauch, H. E., "Order Reduction in Estimation with Singular Perturbation," Fourth Symposium on Nonlinear Estimation and its Applications, pp 231-241, Sept 1973.

[19] Teneketzis, D., and N. R. Sandell, Jr., "Linear Regulator Design for Stochastic Systems by a Multiple Time Scales Method," IEEE Trans. Automat. Control, Vol. AC-22, No. 4, pp 615-621, Aug 1977.

[20] Maybeck, P. S., Stochastic Models, Estimation and Control, Vol. 2, Academic Press, New York, 1981 (to be published).

[21] D'Appolito, J. A., "The Evaluation of Kalman Filter Designs for Multisensor Integrated Navigation Systems," Tech. Rept. AFAL-TR-70-271, The Analytic Sciences Corp., Reading, Mass., Jan 1971.

[22] Gelb, A. (ed.), Applied Optimal Estimation, MIT Press, Cambridge, Mass., 1974.

[23] Hamilton, E. L., G. Chitwood and R. M. Reeves, "An Efficient Covariance Analysis Computer Program Implementation," Proc. IEEE Nat. Aerospace and Electronics Conf, Dayton, Ohio, pp 340-345, May 1976.

[24] Musick, S. H., "SOFE - A Computer Program for Kalman Filter Design," Proc. IEEE Nat. Aerospace and Electronics Conf., Dayton, Ohio, pp 742-749, May 1981; also "SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation, User's Manual," Tech. Rept. AFWAL-TR-80-1108, Air Force Wright Aeronautical Labs., Wright-Patterson AFB, Ohio, Oct 1980.

[25] Britting, K. R., Inertial Navigation System Analysis, John Wiley and Sons, New York, 1971.

[26] Heller, W. G., "Models for Aided Inertial Navigation System Sensor Errors, "Tech. Rept. TR-312-3, The Analytic Sciences Corp., Reading, Mass., Feb 1975.

[27] Leondes, C. T. (ed.), Theory and Applications of Kalman Filtering, AGARDograph No. 139, NATO Advisory Group for Aerospace Research and Development, London, Feb 1970.

[28] Pinson, J. C., "Inertial Guidance for Cruise Vehicles," in Guidance and Control of Aerospace Vehicles (C. T. Leondes, ed.), McGraw-Hill, New York, 1963.

[29] Widnall, W. S., and P. A. Grundy, "Inertial Navigation System Error Models," Tech. Rept. TR-03-73, Intermetrics, Inc., Cambridge, Mass., May 1973.

[30] Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

[31] McGarty, T. P., Stochastic Systems and State Estimation, John Wiley and Sons, New York, 1974.

[32] Friedland, B., "On the Effect of Incorrect Gain in the Kalman Filter," IEEE Trans. Automat. Control, Vol. AC-12, No. 4, p 610, Oct 1967.

[33] Griffin, R. E., and A. P. Sage, "Sensitivity Analysis of Discrete Filtering and Smoothing Algorithms," AIAA J., Vol. 7, No. 10, pp 1890-1897, 1969.

[34] Kwakernaak, H., "Sensitivity Analysis of Discrete Kalman Filters," Int. J. Control, Vol. 12, pp 657-669, 1970.

[35] Nash, R. A and F. B. Tuteur, "The Effect of Uncertainties in the Noise Covariance Matrices on the Maximum Likelihood Estimate of a Vector," IEEE Trans. Automat. Control, Vol. AC-13, No. 1, pp 86-88, Jan 1968.

[36] Fitzgerald, R. J., "Divergence of the Kalman Filter," IEEE Trans. Automat. Control, Vol. AC-16, No. 6, pp 736-747, Dec 1971.

[37] Price, C. F., "An Analysis of the Divergence Problem in the Kalman Filter," IEEE Trans. Automat. Control, Vol. AC-13, No. 6, pp 699-702, Dec 1968.

[38] Schlee, F. H., C. J. Standish and N. F. Toda, "Divergence in the Kalman Filter," AIAA J., Vol. 5, No. 6, pp 1114-1120, 1967.

[39] Safonov, M. G., and M. Athans, "Robustness and Computational Aspects of Nonlinear Stochastic Estimators and Regulators," IEEE Trans. Automat. Control, Vol. AC-23, No. 4, pp 717-725, Aug 1978.

[40] D'Appolito, J. A., and C. E. Hutchinson, "Low Sensitivity Filters for State Estimation in the Presence of Large Parameter Uncertainties," IEEE Trans. Automat. Control, Vol. AC-14, No. 3, pp 310-312, June 1969.

[41] Doyle, J. C., and G. Stein, "Robustness with Observers," IEEE Trans. Automat. Control, Vol. AC-24, No. 4, pp 607-611, Aug 1979.

[42] Hutchinson, C. E., J. A. D'Appolito, and K. J. Roy, "Applications of Minimum Variance Reduced - State Estimators," IEEE Trans. Aerospace and Electronic Sys., Vol. AES-11, No. 5, pp 785-794, Sept 1975.

[43] Vagners, J., "Design of Numerically Stable Flight Filters from Minimum Variance Reduced Order Estimators," Proc. of IEEE Nat. Aerospace and Electronics Conf., Dayton, Ohio, pp 577-581, May 1979.

[44] Clark, R. R., "Performance Sensitivity Analysis of a Kalman Filter Using Adjoint Functions," Tech. Rept. NAFI TR-1767, Naval Avionics Facility, Indianapolis, Ind., Feb 1972.

[45] Bierman, G. J., Factorization Methods for Discrete Sequential Estimation, Academic Press, New York, 1977.

[46] Carlson, N. A., "Fast Triangular Formulation of the Square Root Filter," AIAA J., Vol. 11, No. 9, pp 1259-1265, 1973.

[47] Kaminski, P. G., A. E. Bryson, Jr., and S. F. Schmidt, "Discrete Square Root Filtering: A Survey of Current Techniques, IEEE Trans. Automat. Control, Vol. AC-16, No. 6, pp 727-735, Dec 1971.

[48] Foote, A., C. Vellenga, J. Price, and W. Buchholz, "Applications of Covariance Analysis Simulation to Avionics Flight Testing," Proc. IEEE Nat. Aerospace and Electronics Conf., Dayton, Ohio, pp 750-756, May 1981.

[49] "Position Update Filter Function," Radiometric Area Correlation Guidance Captive Flight Test R&D Status Rept LMSC-D434550, Lockheed Missiles and Space Co., Inc., Sunnyvale, Cal., Nov. 1975.

[50] Maybeck, P. S., "Performance Analysis of a Particularly Simple Kalman Filter," AIAA J. of Guid. and Cont., Vol. 1, pp 391-396, Dec 1978.

[51] Morrison, R., H. Garret, and B. Walls, "A Strapdown Laser Gyro Navigator," Proc. IEEE Nat. Aerospace and Electronics Conf., Dayton, Ohio, May 1974.

[52] Pasik, D. J., M. I. Gneses and G. R. Taylor, "A Ring Laser Gyro Strapdown Inertial Navigation System: Performance Analysis and Test Results," AIAA Paper 75-1075, Boston, Mass, Aug 1975.

DESIGN AND PERFORMANCE ANALYSIS OF AN ADAPTIVE EXTENDED KALMAN FILTER FOR TARGET IMAGE TRACKING

Peter S. Maybeck
Department of Electrical Engineering
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

SUMMARY

A simple extended Kalman filter is designed to track targets using outputs from a forward-looking infrared (FLIR) sensor as measurements. It exploits knowledge unused by current correlation trackers - size, shape, and motion characteristics of the target, atmospheric jitter spectral description, and background and sensor noise characteristics - to yield enhanced performance. Monte Carlo performance analyses indicate that the ability of a nonadaptive four-state filter to track a realistic distant point source target with an error standard deviation of 0.2 picture elements under expected tracking conditions surpasses the correlation trackers' abilities by an order of magnitude.

Although very accurate tracking performance is achieved under nominally assumed conditions, robustness studies portray a significant degradation when the filter's internal model does not depict the target's intensity profile or motion characteristics well. Background noise properties are shown to be of secondary importance at expected signal-to-noise ratios. These studies emphasize the need for good models and adaptivity within the filter structure.

In order to track air-to-air missiles at close range, an eight-state filter incorporates a modified target dynamics model as well as online adaptation to target shape effects, changing target motion characteristics, and maximum signal intensity. It is shown to possess considerable performance potential for highly maneuverable targets despite background clutter.

1. INTRODUCTION

Current research and development efforts are examining several methods of tracking a target for the purpose of depositing high power laser energy on that target in the presence of several disturbances. These disturbances include any effect that can cause relative motion between the beam and target, such as target motion, atmospheric jitter, vibration in optics mirrors, and sensor measurement errors.

One tracking method under investigation employs a forward-looking infrared (FLIR) sensor together with a correlation algorithm to provide relative target position information to the laser pointing system. The FLIR sensor generates averaged outputs of an array of infrared detectors as they are mechanically scanned through a limited field of view. The digitized outputs can either be stored or displayed on a cathode ray tube (CRT) in real time, each output corresponding to the average intensity over one picture element (pixel). The horizontal and vertical scanning of the detectors through the FLIR field of view results in an array of pixels called a frame of data, with normal frame rates on the order of 30Hz. Because of this rapid measurement rate, attention can be confined to a pixel array smaller than the entire frame for tracking purposes: an 8-by-8 array is a typical tracking window, yielding a tolerable amount of computer storage and loading.

The correlation algorithm [1 - 4] first stores a complete set of intensity data from the FLIR outputs, and then correlates those data with the new information at a later time. In this manner, it estimates the two-dimensional position offset from one set of data to the next, which can be used to generate commands to keep the system centered on the target. This type of tracker needs no prior information about the type of target to perform the tracking function, and so it is well suited to many general applications.

In many practical tracking problems, however, the type of target being tracked will be known, even if in a very general sense. This implies that certain target parameters such as shape, size, and motion characteristics will either be known or could be estimated. Moreover, the statistical effects of atmospheric disturbances on radiated wavefronts are known and could supply information to a tracker that would aid in separating the true target motion from the apparent motion (jitter) due to these disturbances. This separation is important since the wavefront of the high energy laser will not undergo the same distortion in the atmosphere as the infrared wavefronts emanating from the target.

The purpose of this effort is to exploit the knowledge about potential target characteristics and atmospheric jitter in the design of an extended Kalman filter [5-7] to replace the current correlation algorithm in the tracking loop. Initially, a simple extended Kalman filter algorithm is designed to track a point source (distant) target with rather benign dynamics, based on FLIR measurements assumed to be corrupted by temporally and spatially uncorrelated noises, and assuming all system defining parameters have known nominal values [8,9]. Section 2 establishes the required mathematical models and basic filter, and Section 3 analyzes its performance capabilities in a realistic environment via Monte Carlo analysis [8,9]. It consistently outperforms the correlation tracker under nominally assumed conditions, employing knowledge unused by that tracker to yield the enhanced capability. Robustness studies are conducted in Section 4 to indicate how much filter performance degrades when an accurate portrayal of the tracking problem differs from that assumed in the filter design [10,11]. Of specific interest are variations in (1) the height, spread, shape, and orientation of the target intensity pattern in the FLIR image plane, (2) target motion characteristics, and (3) background noise rms value and both spatial and temporal correlations. This investigation provides insights into a prioritized list of design modifications and online adaptation capabilities required to allow this type of filter to track highly maneuverable targets, with spatially distributed and changing image intensity profiles, against background clutter. The subsequent sections delineate specific modeling and adaptation methods to yield a filter capable of accurate

tracking of an air-to-air missile at close range, with performance evaluations of proposed designs achieved by Monte Carlo simulations [11,12].

2. MODELS AND FILTER FOR BENIGN TRACKING TASK

As originally conceived, the problem of interest was accurate tracking of a point source target based on FLIR measurements, to provide appropriate inputs to a pointing controller [8,9]. In essence, this involves determining the pointing errors in two dimensions from the center of the FLIR field of view, given measurements of average intensity level over each of 64 pixels in an 8-by-8 "tracking window" array provided by the FLIR at a 30 Hz rate. Many applications for which this system is being considered do in fact require the acquisition and tracking of targets at long ranges. Because of these distances, even very large targets appear as point sources of infrared radiation and can be accurately modeled as such. Due to the physics of wave propagation and optics, the resulting intensity pattern on the FLIR image plane can be modeled as a bivariate Gaussian function with circular equal intensity contours. This is a special case of elliptical contours as depicted in Fig. 1, in which $\sigma_{g1} = \sigma_{g2} = \sigma_g$. Letting $(x_{peak}(t), y_{peak}(t))$ locate the center of the pattern relative to the center of the 8-by-8 pixel array, the apparent target intensity model for circular equal intensity contours is

$$I_{target}(\xi_x,\xi_y,t) = I_{max} \exp\left[-\frac{1}{2\sigma_g^2} \{[\xi_x - x_{peak}(t)]^2 + [\xi_y - y_{peak}(t)]^2\} \right] \tag{1}$$

where the peak intensity value I_{max} and glint dispersion σ_g were originally assumed to be known. The apparent location of the target is actually the sum of effects due to true target dynamics, atmospheric disturbances, and vibration (denoted, respectively, by subscripts d, a, and v):

$$x_{peak}(t) = x_d(t) + x_a(t) + x_v(t) \tag{2a}$$

$$y_{peak}(t) = y_d(t) + y_a(t) + y_v(t) \tag{2b}$$

The objective of the tracker is to estimate $x_d(t)$ and $y_d(t)$ so that they can be regulated by closed-loop control.

A generally applicable model for benign target dynamics is desired, one which is simple and yet accounts for the time-correlated behavior of realistic targets. To fulfill these objectives, an independent first-order Gauss-Markov model in each direction was chosen, as produced by

$$\dot{x}_d(t) = -[1/T_d]\, x_d(t) + w_{dx}(t) \tag{3}$$

where T_d is the characteristic correlation time of the target and w_{dx} is a zero-mean white Gaussian noise with autocorrelation function

$$E\,(w_{dx}(t)w_{dx}(t+\tau)) = 2\sigma_d^2/T_d\, \delta(\tau) \tag{4}$$

so that σ_d is the rms value of $x_d(t)$, and similarly for $y_d(t)$. By suitable choice of σ_d and T_d, samples from these processes can be made to exhibit amplitude and rate-of-change characteristics appropriate to a variety of long range targets as seen in the image plane. Again, these two parameters were originally assumed to be known.

Atmospheric disturbance causes an apparent offset of the location of the intensity pattern known as jitter. Through spectral analysis of this phenomenon, it has been shown [13,14] that $x_a(t)$ and $y_a(t)$ can each be adequately modeled as the output of a third-order linear shaping filter [6] with transfer function $[K\omega_1\omega_2(s+\omega_1)^{-1}(s+\omega_2)^{-2}]$, driven by unit-strength white Gaussian noise. Here $\omega_1 \simeq 14$ rad/sec, $\omega_2 \simeq 660$ rad/sec, and K can be adjusted to obtain the desired rms jitter characteristic on the output and was assumed to be known.

Vibrations of the FLIR system can also cause relative pointing errors. However, for this study, it is assumed to be on a ground-based stable platform, so vibration effects are neglected. For airborne applications and other scenarios, vibration-induced effects may warrant considerably more attention.

Thus, the target intensity pattern given by (1) has been fully described. However, this pattern is not directly available for observation: it is corrupted by background noise and inherent FLIR errors first. There are various forms of background noise, ranging from night sky background to clutter, i.e., from zero to high time and spatial correlations. FLIR errors, such as thermal noise and dark current effects, can be modeled as temporally and spatially uncorrelated noise. Letting $z_{jk}(t_i)$ denote the measurement available at time t_i of the average intensity over the pixel in the jth row and kth column of the 8-by-8 array, then

$$z_{jk}(t_i) = \frac{1}{A_p} \iint_{\substack{\text{region of} \\ \text{jkth pixel}}} I_{target}(\xi_x,\xi_y,t_i)\,d\xi_x d\xi_y + n_{jk}(t_i) + b_{jk}(t_i) \tag{5}$$

where A_p is the area of one pixel, $n_{jk}(t_i)$ models the FLIR noise effects, and $b_{jk}(t_i)$ models the background effects on the jkth pixel. Arraying the 64 scalar equations (5) in a single measurement vector yields a measurement model of the form

$$\underline{z}(t_i) = \underline{h}[\underline{x}(t_i),t_i] + \underline{n}(t_i) + \underline{b}(t_i) \tag{6}$$

where $\underline{x}$ is the output of an eight-state linear dynamics system model (one state equation as given by (3) and three coupled linear equations to generate x_a, and similarly for the y axis), and the other vectors are of dimension 64. From (5) and (6) it can be seen that $\underline{h}$ represents the effect of the point-spread function given in (1). Note that, in such a formulation, the spatial correlation of background noise is readily represented by the off-diagonal elements of the 64-by-64 matrix $E(\underline{b}(t_i)\underline{b}^T(t_i))$.

The model just developed accounts for time-correlated dynamics, bandwidth effects of jitter, and other pertinent characteristics. An extended Kalman filter could be based on this model to perform the desired tracking task.

First, to enhance computational feasibility, the model can be simplified to some degree. In view of the discrepancy between the two break frequencies of the atmospheric disturbance shaping filter and the greater importance of the lower frequency characteristics, atmospheric disturbance effects x_a and y_a were approximated as the outputs of first-order systems with break frequency ω_1 (thus preserving proper spectral shape at the significant frequencies below ω_2). This yielded a four-state linear time-invariant model of the form

$$\dot{\underline{x}}(t) = \underline{F}\underline{x}(t) + \underline{w}(t) \tag{7}$$

with diagonal $\underline{F}$ and stationary white Gaussian input $\underline{w}(t)$. Thus, the filter equations for propagating the state estimate $\hat{\underline{x}}$ and error covariance $\underline{P}$ from sample time t_{i-1}^+ (after measurement incorporation at that time) to t_i^- (before measurement update) become [6]:

$$\hat{\underline{x}}(t_i^-) = \underline{\Phi}\hat{\underline{x}}(t_{i-1}^+) \tag{8}$$

$$\underline{P}(t_i^-) = \underline{\Phi}\underline{P}(t_{i-1}^+)\underline{\Phi}^T + \underline{Q}_d \tag{9}$$

where the state transition matrix $\underline{\Phi}$ (also diagonal) and input covariance

$$\underline{Q}_d = \int_{t_{i-1}}^{t_i} \underline{\Phi}(t_i-\tau)\,\underline{Q}\,\underline{\Phi}^T(t_i-\tau)d\tau \tag{10}$$

are constant and readily calculated once offline. Because of the low state dimensionality, linearity, and time invariance of the dynamics model, these filter time propagations are especially simple.

Furthermore, each measurement equation, (5), was simplified as well. First, the two-dimensional integral term is replaced by $I_{target}(x_{ck}, y_{cj}, t_i)$ where (x_{ck}, y_{cj}) is the location of the center of the jkth pixel. Second, the combined effects of $\underline{n}$ and $\underline{b}$ in (5) are represented by a single vector $\underline{v}$, assumed to have spatially and temporally uncorrelated components of constant and equal variance:

$$E(\underline{v}(t_i)\underline{v}^T(t_j)) = \begin{cases} R\underline{I} & t_i = t_j \\ \underline{0} & t_i \neq t_j \end{cases} . \tag{11}$$

These simplifications are made to reduce complexity substantially, but are subject to reevaluation if performance capabilities are inadequate.

The large number of measurements cause a computational loading problem in the normal extended Kalman filter formulation of the measurement update:

$$\underline{K}(t_i) = \underline{P}(t_i^-)\underline{H}^T(t_i)[\underline{H}(t_i)\underline{P}(t_i^-)\underline{H}^T(t_i) + \underline{R}(t_i)]^{-1} \tag{12}$$

$$\underline{P}(t_i^+) = \underline{P}(t_i^-) - \underline{K}(t_i)\underline{H}(t_i)\underline{P}(t_i^-) \tag{13}$$

$$\hat{\underline{x}}(t_i^+) = \hat{\underline{x}}(t_i^-) + \underline{K}(t_i)(\underline{z}(t_i) - \underline{h}[\hat{\underline{x}}(t_i^-), t_i]) \tag{14}$$

where $\underline{h}$ is defined in (6) and approximated componentwise by I_{target} evaluated at the center of each pixel, rather than its spatial average over the entire pixel, and $\underline{H}(t_i)$ is the partial of $\underline{h}$ with respect to $\underline{x}$, evaluated at $\hat{\underline{x}}(t_i^-)$. The gain calculation in (12) requires inversion of a 64-by-64 matrix. To circumvent this burden, (12) and (13) are replaced by the equivalent inverse covariance form [6,7]:

$$\underline{P}^{-1}(t_i^+) = \underline{P}^{-1}(t_i^-) + \underline{H}^T(t_i)\underline{R}^{-1}(t_i)\underline{H}(t_i) \tag{15}$$

$$\underline{P}(t_i^+) = [\underline{P}^{-1}(t_i^+)]^{-1} \tag{16}$$

$$\underline{K}(t_i) = \underline{P}(t_i^+)\underline{H}^T(t_i)\underline{R}^{-1}(t_i). \tag{17}$$

This form only requires two 4-by-4 matrix inversions online; $\underline{R}^{-1}(t_i)$ is constant and is generated once offline (it is also diagonal if (11) is used).

3. PERFORMANCE ANALYSIS UNDER NOMINAL CONDITIONS

The performance capabilities of the extended Kalman filter were evaluated and compared to those of the correlator algorithm by means of a Monte Carlo analysis [8,9]. In this analysis, the full-scale model developed in Section 2 was used to generate sample-by-sample simulations, differing in particular realizations drawn from random noise sources. Sample statistics of the tracking errors committed by each algorithm were computed on the basis of 20 simulation runs (chosen by observing convergence of computed statistics to consistent values as the number of runs were increased).

This analysis was directed at four areas of primary interest:

1) Performance as a function of signal-to-noise ratio S/N defined here as

$$\frac{S}{N} = \frac{I_{max}}{\text{(rms value of background noise)}} \tag{18}$$

Ratios of 20, 10, and 1 were investigated.

2) Performance as a function of intensity pattern size (spot size on image plane) relative to pixel size: Gaussian beam dispersion σ_g was set to both 3 and 1 pixel.

3) Performance as a function of the ratio of rms target motion to rms atmospheric jitter: (σ_d/σ_a) values of 5, 1, and 0.2 were considered.

4) Performance as a function of target correlation time: targets with T_d of both 1 and 5 sec. were simulated.

All of these studies were conducted under nominally assumed conditions: as the parameters defining the real-world environment were changed, the filter was (artificially) provided knowledge of their value. Thus, there was no purposeful model mismatch between the filter and the actual tracking environment; such important robustness studies are described in the next section. Also, the filter was implemented in open loop for this initial analysis: computed offsets were not fed to a pointing control system to be nulled out.

In order to optimize its performance, the filter must be tuned by adjusting the strengths of both the dynamic driving noises and measurement corruption noises. For the simulations conducted, the FLIR and background noises in (5) and (6) were assumed independent, spatially and temporally uncorrelated, and Gaussian, so R in (11) was set equal to the sum of the variances for n_{jk} and b_{jk}. Adequate tuning results when the strength of the white noise terms and correlation times for both the simulation and filter target dynamics models are set equal, and when the rms values for atmospheric jitter for both models (of different order) are equated. Fig. 2 depicts the actual versus filter-computed error standard deviation committed in estimating atmospheric jitter in the horizontal direction for a typical case ($S/N = 10$, $\sigma_g = 3$, $\sigma_d = \sigma_a = 1$, $T_d = 1$); in this and all other cases, the adequacy of both the proposed filter tuning and the order reduction of the filter dynamics model is demonstrated by the good agreement between the two curves. When the signal-to-noise ratio is decreased to one, the filter tends to underestimate its own errors (by about the same margin it overestimates them in Fig. 2) using the tuning philosophy described, due to the mismatch between true and filter models becoming more apparent; this can readily be remedied by increasing the filter Q_d entries if desired. In fact, for a "conservative" or robust filter that is able to withstand modeling errors yet still provide good estimation performance (beyond the bare minimum of nondivergent characteristics [15]), one might want to tune the filter purposely so that it overestimates its own errors somewhat [6,7]. Had biased estimates been a problem for this application, it could have been combatted by tuning so as to match filter-computed error variances and actual mean square errors [7]; incorporating the "bias correction term" from second order filtering [7,16] or implementing an entire second order filter [5,7,16] would be prohibitive computationally.

For the typical case and tuning philosophy depicted in Fig. 2, Fig. 3 portrays the sample mean error $\pm 1\sigma$ (standard deviation) committed by the filter in estimating the target true horizontal location. By comparison, Fig. 4 depicts the performance of the correlator algorithm under the same conditions. In this case, both algorithms yield rather unbiased estimates, but the filter error standard deviation is only one fourth that of the correlator after the 5 sec. simulation (and the correlator performance is steadily worsening).

The filter tuning is independent of the Gaussian glint dispersion (spot size). Furthermore, if S/N is adjusted by changing only I_{max} in (18), it is also independent of the signal-to-noise ratio. This allows portrayal of performance of a singly tuned filter in different tracking environments. Tables I and II present a comparison of the two algorithms in mean and 1σ tracking error for the three signal-to-noise ratios examined, Table I pertaining to the case of $\sigma_g = 3$ pixels and Table II to $\sigma_g = 1$ pixel.

TABLE I

MEAN ERROR AND 1σ ERROR COMPARISONS WITH $\sigma_g = 3$ PIXELS

	Correlation Tracker		Extended Kalman Filter	
S/N	mean error (pixels)	1σ error (pixels)	mean error (pixels)	1σ error (pixels)
20	0.5	1.5	0.0	0.2
10	3.0	3.0	0.0	0.2
1	15.0	30.0	0.0	0.8

TABLE II

MEAN ERROR AND 1σ ERROR COMPARISONS WITH $\sigma_g = 1$ PIXEL

	Correlation Tracker		Extended Kalman Filter	
S/N	mean error (pixels)	1σ error (pixels)	mean error (pixels)	1σ error (pixels)
20	7.0	8.0	0.0	0.2
10	8.0	10.0	0.0	0.2
1	15.0	30.0	0.0	0.8

These are results at the end of the 5 sec. simulations and each represents an average of values generated from Monte Carlo simulations using the three different values of σ_d/σ_a. The extended Kalman filter performs well as the signal-to-noise ratio is lowered, with no noticeable change between the ratios 20 and 10, and exhibiting only a slight degradation in performance at S/N = 1. It consistently outperforms the correlation tracker, especially at lower signal-to-noise ratios. In fact, the correlation algorithm repeatedly exhibited a divergent characteristic in the more difficult tracking environments. As shown in Table II, decreasing the dispersion of the Gaussian intensity function seriously affected the correlator tracking, whereas the filter is essentialy unaffected.

When the ratio of rms target motion to rms atmospheric jitter was decreased from 5 to 1, the mean filter error remained close to zero, but the 1σ value increased from 0.2 to 0.5 pixels (for the case of S/N = 20, σ_g = 3, T_d = 1). Decreasing it further to σ_d/σ_a = 0.2 resulted in a return to a level of about 0.2 pixels.[9] This seems to imply that the filter is able to distinguish between true and apparent target motion more easily when there is a significant amplitude difference between the two effects. In all three corresponding cases, the correlation algorithm had a 1σ error of about one pixel, so that even in the worst case, the filter surpassed the correlator's tracking ability by a wide margin.

Increasing the target correlation time from 1 to 5 sec. had no discernible effect on the performance of either tracker. However, a larger variation in T_d might well demonstrate that, as the characteristics of the true target dynamics and atmospheric jitter become more distinctly different, the filter is better able to separate the effects and enhance tracking, whereas the correlator cannot perform this function.

4. ROBUSTNESS OF FILTER

The marked performance improvement over that attained by a correlation algorithm was achieved by a filter based on appropriate modeling assumptions, parameter values and tuning. However, this raises the robustness issue of how much filter performance degrades when an accurate portrayal of the tracking problem differs from that assumed in the filter design. In this section, first the sensitivity of the estimation performance to large variations in parameter values within assumed model forms is depicted. Then sensitivity to variations in the basic structure of the appropriate models is presented. Finally, insights into required design modifications and online adaptation capability are summarized [10,11].

4.1 Sensitivity to Model Parameter Mismatches

The extended Kalman filter was based upon nominal parameter values of

(1) maximum intensity level, I_{max} = 10 units (arbitrary scale)
(2) glint dispersion (spread), σ_g = 3 pixels
(3) target dynamics rms value, $\sigma_d = \sigma_a$ (rms jitter) = 1 pixel
(4) target dynamics correlation time, T_d = 1 sec
(5) signal-to-noise ratio, S/N = 10 (i.e., rms background noise = 1 = 0.1 I_{max})

When the actual environment was well modeled by these parameter values, the standard deviation of the errors in estimating target states x_d and y_d were each 0.56 pixel, and 0.55 pixel in estimating atmospheric jitter states x_a and y_a. Since all errors in these studies were essentially zero-mean, these are also rms error values.

Table III summarizes the effect of varying the true value of these parameters in the Monte Carlo simulation (20 runs per evaluation) without altering the values in the filter. The resulting actual error

TABLE III

ACTUAL ESTIMATION ERROR AVERAGE STANDARD DEVIATIONS WITH MODEL PARAMETER MISMATCHES

True Parameter		Target σ (in pixels)	Jitter σ (in pixels)
I_{max} =	1	3.7	1.7
	10*	.56	.55
	20	.70	1.5
σ_g =	1	3.0	1.5
	3*	.56	.55
	5	.65	.63
σ_d/σ_a =	.2	.38	.47
	1*	.56	.55
	5	2.6	1.6
T_d =	.2	.66	.65
	1*	.56	.55
	5	.43	.46
S/N =	1	3.8	8.0
	10*	.56	.55
	20	.55	.57

* = design conditions, values assumed by filter

standard deviations (averaged over the 5-sec simulations) in estimating target position and atmospheric jitter are presented as each parameter is separately varied from the design conditions.

For the first two robustness studies, the real world I_{max} and σ_g descriptors of the target intensity profile were allowed to vary. When I_{max} is 1, the real FLIR images are of a target much more highly masked by background noise than the filter assumes (S/N is actually 1 rather than 10), with resulting severe degradation. Such a low S/N in fact produces poor performance even without a parameter mismatch [8,9], so this result is expected. However, when I_{max} is increased to 20, the filter again has difficulty, apparently due to searching for the wrong shape of intensity profile due to mismodeled I_{max}. When the true target intensity is less spread out than the filter assumes (σ_g = 1 pixel), the real image can move substantially within the large envelope being sought by the filter, with significant deterioration in performance. On the other hand, when the real image is larger than assumed, estimation accuracy is acceptable: the intensity peak can be located rather precisely.

The filter assumed a rather benign target trajectory, as is appropriate for distant targets. In the next set of robustness studies, the rms values and correlation times of the first order Gauss-Markov process were allowed to vary from design values. When σ_d/σ_a is set to 0.2, the real target motion amplitudes are less than assumed by the filter, and the estimation accuracy is acceptable (the filter overestimates its own errors, and smaller errors could be achieved with correctly assumed σ_d). Unacceptably large errors are produced when the filter underestimates the dynamics amplitudes (σ_d/σ_a = 5). Correlation time variations by a factor of five have insignificant effect, yielding somewhat greater errors when the true target exhibits higher frequency motion than assumed, and smaller errors when the trajectories are more time-correlated than anticipated.

Finally, mismatches in the background noise model were investigated. As in the case of varying I_{max}, changing the rms value for background noise affects S/N, but the trends in performance differ for the two cases. Again, low S/N results in poor estimation (even without mismatches). However, when S/N is high and I_{max} is properly modeled, the assumed target intensity shape is correct while the corrupting noise is actually less than assumed, and the filter tracks the target well.

4.2 Sensitivity to Variations in Model Structure

The filter under investigation was designed to track distant point targets with low angular rate and acceleration capabilities, against a temporally and spatially uncorrelated background. Now it is desired to establish the robustness of the filter to the structure of the assumed models, considering shorter range targets such that (1) shape effects become significant and (2) target dynamics can become more violent, and also considering scenarios in which background noise can be highly correlated, both spatially and/or temporally. One does not necessarily expect accurate tracking by the filter under these very different conditions; rather, the goal is a prioritized list of the characteristics of the new scenario that cause the severest performance degradation.

In fact, rms errors double when true equal intensity contours are ellipses with major axis dimension ten times that of the minor axis, instead of circular as assumed: such an intensity pattern would be representative of some air-to-air missile targets. Unmodeled or mismodeled target motion (especially involving persistent nonzero mean velocities and accelerations, or varying degrees of maneuvering in a single scenario) also have a very serious effect on tracking ability. Extensive performance analyses showed that loss of track occurs consistently whenever unmodeled motion allows the target to move out of the field of view in one or two sample periods, even with a closed-loop system assumed able to null out any estimated errors in a single sample period: the lack of a viable target velocity estimate is a critical shortcoming in this environment. Mismodeled background noise, misrepresented in spatial and temporal correlation as well as in rms value, does not have significant effect at moderate expected values of S/N. For example, at the nominal S/N of 10, introducing exponential spatial correlation symmetrically in all directions with a correlation distance of 1.5 pixels increases the rms tracking error by .03 pixels, while introducing both this spatial correlation and a long temporal correlation (such that the correlation coefficient for a given pixel from one sample time to the next is 0.95) increases the rms error by only 0.1 pixel. When S/N is reduced to 1, such correlations cause consistent loss of track, but, as already seen, very low S/N degrades performance greatly even in the absence of mismodeling in the filter.

4.3 Insights from Robustness Analysis

Thus, to generate a filter capable of tracking air-to-air missiles in background clutter, one must include the following aspects in the design:

1) ability to estimate size, shape, and orientation of the target image;
2) online estimation of target intensity height I_{max} since it is uncertain, varying, and important to filter residual generation and tracking performance;
3) ability to predict future position by maintaining at least a velocity estimate in addition to a position estimate (acceleration estimates may well be required also);
4) adaptation to maneuvers (detecting a maneuver not predicted by the filter, via residual monitoring, and responding appropriately as through gain changing).

Moreover, spatial and temporal correlation of background noise need not be modeled in the filter for expected S/N values. The next sections establish the design and capabilities of a filter with these features.

5. ELLIPTICAL EQUAL INTENSITY CONTOURS

5.1 Basic Model

Analysis of real FLIR data indicated that air-to-air missile images could be well approximated by a bivariate Gaussian intensity pattern, but with elliptical, rather than circular, equal intensity contours.

The ratio of major and minor axis magnitudes, $(\sigma_{g1}/\sigma_{g2})$ as in Fig. 1, typically ranges from 1 to about 10, depending on the aspect angle of the missile. Moreover, the magnitude of σ_{g2} varies with range from target to the tracker.

For development of the filter, it was assumed that the semimajor axis of the ellipse could be aligned with the missile velocity vector (ignoring small angle of attack and sideslip angle). Since target velocity in the FLIR image plane is to be estimated, $\hat{v}_x$ and $\hat{v}_y$ are used to establish the angular orientation of the ellipse major axis. Letting Δx_1 and Δx_2 be measured from $(x_{peak}(t), y_{peak}(t))$ along the principal axes, the target intensity model (1) becomes

$$I_{target}(\Delta x_1, \Delta x_2, t) = I_{max} \exp\{-\tfrac{1}{2}[(\Delta x_1/\sigma_{g1})^2 + (\Delta x_2/\sigma_{g2})^2]\} \tag{19}$$

where I_{max}, σ_{g1}, and σ_{g2} are treated as uncertain (slowly changing) parameters to be estimated simultaneously with the states. Various methods of estimating these uncertain parameters where considered [7,17-24], including treating them as additional states, multiple model Bayesian estimation for discretized parameters, full-scale and approximated maximum likelihood methods, and least-squares techniques.

5.2 Estimation of σ_{g1} and σ_{g2}

Very good performance and small computational burden were achieved by generating the estimates $\hat{\sigma}_{g1}(t_i)$ and $\hat{\sigma}_{g2}(t_i)$ that minimized the quadratic cost

$$C[\underline{Z}(t_i), \underline{a}] = \{\underline{z}(t_i) - \underline{h}[\hat{\underline{x}}(t_i^-), t_i; \underline{a}]\}^T \{\underline{z}(t_i) - \underline{h}[\hat{\underline{x}}(t_i^-), t_i; \underline{a}]\} \tag{20}$$

as a function of $\underline{a}$, where $\underline{Z}(t_i)$ is the measurement history $\{\underline{z}(t_1), \ldots, \underline{z}(t_i)\}$, $\underline{a}$ is the vector of uncertain parameters to be estimated, and $\underline{h}$ is as defined in (6) but using I_{target} as defined in (19). This can be viewed as a least-squares approximation to an estimate based on maximizing the likelihood function $\ell n\, f[\underline{x}(t_i), \underline{z}(t_i)|\underline{Z}(t_{i-1}), \underline{a}]$ with respect to both $\underline{x}(t_i)$ and $\underline{a}$. Usually, one might seek a weighted sum of quadratics of the most recent N residuals instead of a cost involving only the single current residual as in (20) for better performance, but the 64-dimensional measurement in this problem provides significant spatial averaging to supplant temporal averaging. In fact, use of (20) yields very acceptable results. A recursive gradient solution to minimizing (20) was implemented as

$$\underline{\hat{a}}(t_i) = \underline{\hat{a}}(t_{i-1}) + k(\partial C^T/\partial \underline{a})\Big|_{\hat{\underline{x}}(t_i^-), \underline{\hat{a}}(t_{i-1})} \tag{21}$$

i.e., a single gradient step is taken each sample period, with k a scalar step-size control value (established empirically as 0.001) and with the partial derivative evaluated using the currently available state and parameter estimates $\hat{\underline{x}}[t_i^-; \underline{\hat{a}}(t_{i-1})]$ and $\underline{\hat{a}}(t_{i-1})$, respectively. Many terms required in the evaluation of this partial derivative are already available from the filter gain computations. Fig. 5 is indicative of the performance of this simple algorithm; it displays the first half second of a representative single sample time history of estimates of σ_{g1} and σ_{g2}, when true values were $\sigma_{g1} = 5$ pixels, $\sigma_{g2} = 1$ pixel, while the filter was initialized with $\hat{\sigma}_{g1} = \hat{\sigma}_{g2} = 3$ pixels. With these parameter values, the mean and standard deviation of errors in estimating σ_{g1} and σ_{g2} all assumed average values (time averaged over the 4.8 sec following the 0.2 sec transient period obvious in Fig. 5) of approximately 0.15 pixel. These results were obtained for constant true σ_{g1} and σ_{g2}, for a trajectory at constant radius from the tracker; similarly good results were obtained when the aspect angle of the missile varied so that "true" σ_{g1} and σ_{g2} in fact varied.

5.3 Estimation of I_{max}

Although I_{max} could have been treated in like manner, what was eventually implemented was a more direct use of measurement information that provided excellent performance with very small computational burden. Simply selecting the highest observed pixel intensity at time t_i as an estimate of I_{max} was explored, but it suffered due to both background noise corruption effects and a bias even in the absence of noise. If there were no noise, the maximum pixel intensity is the average intensity over the pixel closest to the centroid of the Gaussian intensity profile, which is less than the value I_{max} at the centroid. This bias is a function of the centroid location, σ_{g1}, and σ_{g2}, and can be substantial for small σ_{g1} and σ_{g2}. Assuming the centroid is located at the center of the pixel, on the average, a bias function $b(\sigma_{g1}, \sigma_{g2})$ can be developed; for this feasibility study, a second-order polynomial fit approximation was used. Thus an estimate based on a single time sample of measurement data is

$$\hat{I}_1(t_i) = \max_k [z_k(t_i);\ 1 \le k \le 64] - b[\hat{\sigma}_{g1}(t_{i-1}), \hat{\sigma}_{g2}(t_{i-1})] \tag{22}$$

and this is time averaged with previous estimates to reduce the variance due to background noise:

$$\hat{I}(t_i) = c\,\hat{I}(t_{i-1}) + [1 - c]\hat{I}_1(t_i). \tag{23}$$

Performance capabilities are indicated by a set of simulations in which the missile was flown on an inertially straight trajectory such that at t = 3 sec, it was at a minimum range of 10 km from the tracker, with $v_x = 500$ m/s and $v_y = 300$ m/s as seen in the FLIR image plane (each pixel is a 20 urad square). At that minimum distance, the "true" values were $\sigma_{g1} = 3$ pixels, $\sigma_{g2} = 1$ pixel, and for the whole simulation "true" $I_{max} = 25$. Selecting the highest pixel intensity yielded an I_{max} with mean of 24.47 and standard deviation of 1.12; the latter statistic is comparable to the rms background noise of $\sqrt{2}$. Just time averaging via (23) with c = 0.8 but with no bias correction yielded mean and σ of 24.53 and 0.33, respectively. Using bias correction only via (22) yielded 24.99 and 1.16 respectively, while using (22) and (23) together resulted in a mean of 24.99 and σ of 0.30. These results were achieved with simultaneous estimation of σ_{g1} and σ_{g2}, with precision comparable to that discussed earlier.

6. TARGET MOTION COMPENSATION

6.1 Simple Six-State Filter

As indicated previously, at least the target's velocity must be estimated in addition to its position, to predict its position one sample period ahead for appropriate tracking controller command generation. A velocity estimate was also required in the previous section to orient the elliptical intensity contours. The simplest possible dynamics model for FLIR plane motion that includes velocity states would be

$$\dot{\underline{x}}(t) = \underline{v}(t) \tag{24a}$$

$$\dot{\underline{v}}(t) = \underline{w}(t) \tag{24b}$$

with $\underline{w}$ white Gaussian noise with autocorrelation $E\{\underline{w}(t)\underline{w}^T(t+\tau)\} = \underline{Q}(t)\delta(\tau)$, and $\underline{Q}(t)$ chosen (adaptively) to provide an adequate representation of target maneuverability. Such a model only increases the filter state dimension from four to six, and the dynamics model remains linear. Though computationally simple, the filter based on such a model does not yield very good performance for this application. For instance, Fig. 6 presents the mean error ± 1 standard deviation in estimating horizontal position for a 20-run Monte Carlo simulation of the inertially straight trajectory described previously, with true $I_{max} = 25$, $\sigma_{q1} = 5$ pixels, $\sigma_{q2} = 1$ pixel, and background noise rms value of 2. In fact, this plot corresponds to a case of estimating $\underline{Q}$ online after an acquisition phase, as discussed subsequently, but is representative of results in which $\underline{Q}$ is artifically tuned offline at a constant value (after acquisition) for good performance on this type of trajectory. The projection of the inertially constant velocity into the FLIR image plane changes with time as the tracker rotates to maintain the target in the center of its field of view: an unmodeled noninertial acceleration is thus created, manifesting itself in the positive slope of the mean error depicted in the figure. Moreover, velocity estimates diverge significantly over the last second, yielding eventual loss of track if the simulations were over longer periods. That the trend in the figure is due to noninertial acceleration was corroborated by tracking a missile at a constant range, yielding essentially zero-mean error and $\sigma \simeq 0.2$ pixels for all time with no divergence. The tracker control signals which cause the noninertial acceleration could be made available for filter compensation and improved performance. However, less benign target trajectories further justified the need for a better dynamics model.

6.2 Preferable Eight-State Filter

As a result, an eight-state filter was generated that estimated acceleration in the FLIR plane as well, as

$$\dot{\underline{x}}(t) = \underline{v}(t) \tag{25a}$$

$$\dot{\underline{v}}(t) = \underline{a}(t) \tag{25b}$$

$$\dot{\underline{a}}(t) = \underline{w}(t). \tag{25c}$$

Alternative models of acceleration, such as an exponentially time-correlated process model (which introduces an additional uncertain parameter, the correlation time) and a constant turn-rate model [25,26] of

$$\dot{\underline{a}}(t) = -\omega^2\underline{v}(t) + \underline{w}(t) \tag{26a}$$

$$\omega = |\underline{v}(t) \times \underline{a}(t)|/|\underline{v}(t)|^2 \tag{26b}$$

(which yields nonlinear dynamics) were considered, but (25) was explored most fully because of its simplicity and performance potential. In a duplicate tracking environment as used to generate Fig. 6, the eight-state filter produced much improved tracking performance as indicated in Fig. 7 (without being provided control signals for compensation).

6.3 Acquisition

The preceding results reflect a filter provided with perfect initial state knowledge, so recovery from realistic initial condition errors was investigated. To provide acquisition capability, the filter initial covariance $\underline{P}_0$ was assumed diagonal with large entries corresponding to target states: 25 pixel^2, 2000 pixels^2/sec^2, and 100 pixels^2/sec^4, respectively. Further, the $\underline{Q}$ values were maintained at a high value (600 pixels^2/sec^5) for 0.5 sec after initialization. With 8 m/sec true initial velocity error in each direction, performance is as depicted in Fig. 8: acquisition is accomplished in about half a second, followed by tracking capability as portrayed previously.

6.4 Adaptive Tuning

Adaptive estimation of the dynamics noise covariance matrix was investigated to allow self-tuning to an uncertain and dynamically changing environment. The ability to adjust filter bandwidth online was considered necessary because an air-to-air missile can exhibit a wide range of dynamic characteristics. Various methods of covariance estimation were considered, including maximum likelihood, multiple model Bayesian adaptation, and correlation and covariance matching techniques [7,17-24,27-29]. Due to performance and computational considerations, the approximation to maximum likelihood estimation first described in [27] was employed: if the filter covariance propagation and update equations are as given in Eqs. (9) and (13), then an estimate of $\underline{Q}_d(t_i)$ is provided by

$$\hat{\underline{Q}}_d(t_i) = (1/N) \sum_{j=i-N+1}^{i} [\delta\underline{x}(t_j)\delta\underline{x}^T(t_j) + \underline{P}(t_j^+) - \underline{\Phi}\, P(t_{j-1}^+)\underline{\Phi}^T] \tag{27}$$

where

$$\delta\underline{x}(t_j) = \hat{\underline{x}}(t_j^+) - \hat{\underline{x}}(t_j^-). \tag{28}$$

Equation (27) can also be derived heuristically by noting that

$$E\{\delta\underline{x}(t_j)\delta\underline{x}^T(t_j)\} = \underline{K}(t_j)\underline{H}(t_j)\underline{P}(t_j^-) \tag{29}$$

and substituting this and (9) into (13), and approximating the ensemble average in (29) by a temporal average over the most recent N sample periods. To reduce storage requirements, a fading memory approximation to the finite memory result (27) was implemented as

$$\hat{\underline{Q}}_d(t_i) = k\,\hat{\underline{Q}}_d(t_{i-1}) + [1 - k]\hat{\underline{Q}}_{d1}(t_i) \tag{30}$$

where $\hat{\underline{Q}}_{d1}(t_i)$ is a single term from the summation in (27). The parameter k was empirically set to 0.8 as a tradeoff: $k\epsilon(0.8, 1.0)$ provides smoother estimates while $k\epsilon(0, 0.8)$ provides more rapid response to true changes in dynamics. On the inertially straight trajectory and similar benign trajectories, the filter with adaptive $\underline{Q}_d$ estimation performed approximately the same as a filter with artifical offline tuning to a given trajectory.

However, this adaptation was insufficient for abrupt significant maneuvers. In one set of simulations, the missile initiated a 20 g pullup at t = 4 sec of the previously defined trajectory. With a field of view of 160 μrad by 160 μrad, such an acceleration can move the image 3 to 4 pixels laterally in two sample periods. Moreover, virtually all of the position information for the lateral direction (i.e., region of high intensity profile gradients) is within 1 or 2 times σ_{q2} (= 1 pixel here). The $\underline{Q}_d$, appropriately low for the benign portion of the trajectory, did not respond fast enough to preclude loss of track. Although the filter residuals were of large magnitude when the maneuver was conducted, the gains were too low to weight them enough; by the time the gains grew, there was little overlap of the actual and predicted intensity profiles, so the residuals were insignificant.

6.5 Significant Maneuver Indication and Appropriate Filter Response

To maintain track required several matrix coefficients (filter gain and/or covariance matrix $\underline{P}$, as well as $\underline{Q}_d$) and some state estimates to change significantly in a single sample period: for instance, true elevation acceleration changes from 0 to 2400 pixels/sec^2 when the lateral maneuver begins. One possible variable for reliable and rapid indication and quantification of a maneuver is the scalar

$$\delta\underline{x}(t_i)^T\delta\underline{x}(t_i) = \mathrm{tr}[\delta\underline{x}(t_i)\delta\underline{x}^T(t_i)] \tag{31}$$

where

$$\delta\underline{x}(t_i) = \underline{P}(t_i^+)\underline{H}^T(t_i)\underline{R}^{-1}(t_i)(\underline{z}(t_i) - \underline{h}[\hat{\underline{x}}(t_i^-), t_i]) \tag{32}$$

and the filter gain shown in (32) is used because of dimensionality considerations, as discussed in Section 2. Since inappropriately small $\underline{P}(t_i^+)$ values is part of the maneuver compensation problem, and since $\underline{R}(t_i)$ is assumed diagonal, an alternative indicator is the magnitude or separate components of the available vector

$$\delta\underline{y}(t_i) = \underline{H}^T(t_i)(\underline{z}(t_i) - \underline{h}[\hat{\underline{x}}(t_i^-), t_i]) \tag{33}$$

where $\delta\underline{x}(t_i) = \underline{P}(t_i^+)\delta\underline{y}(t_i)/R(t_i)$. At the sample instant after initiation of the 20 g maneuver, at t = 4.03 sec., both (31) and the elevation position component of (33) increased by an order of magnitude, then returned to their normal magnitudes on the following sample instant, t = 4.07 sec (because the true and projected images had already diverged far enough laterally, about 5 pixels, to generate essentially no overlap and thus very small residuals). Therefore, a maneuver was detected by one of these indicators surpassing a magnitude threshold, with the components of $\delta\underline{y}(t_i)$ providing direction information about the unmodeled maneuver as well.

Upon maneuver detection the appropriate response is (1) to increase the filter gain directly, rather than to allow slow increase via $\underline{Q}_d$ estimation, (2) to incorporate updated state estimates for reprocessing the previous sample period's state estimate time propagation and for future propagation, and (3) to expand the field of view (e.g., treating averaged intensities of 2-by-2 arrays of pixels, instead of individual pixels, as filter measurements). Increased gain was introduced by reverting to an acquisition cycle of very high reinitialized $\underline{P}$ and high $\underline{Q}_d$ for 0.5 sec thereafter. To reprocess the state propagation, a curve-fit function was established for unmodeled position displacements as a function of σ_{q1}, σ_{q2}, and the first two components of $\delta\underline{y}(t_i)$; these displacements were assumed to be the result of an unmodeled acceleration acting over the previous sample period. If a maneuver is detected at time t_i, the acceleration state estimates of $\hat{\underline{x}}(t_i^+)$ are modified by these calculated increments, and $\hat{\underline{x}}(t_i^-)$ recomputed. Incorporating these changes postponed divergence for about 15 sample periods. A further ad hoc adaptation used the six target state components of $[\hat{\underline{x}}(t_i^+) - \underline{x}(t_i^-)]$ throughout the acquisition period as an indicator of unmodeled target dynamics, which was then added during the next sample period to the standard propagation equations. This allowed nondivergent results for the entire simulation period. Expansion of the field of view was not evaluated, but it is a useful means of maintaining track at the expense of some resolution.

Although these modifications allowed track to be maintained, the required artificial introduction of nonzero acceleration estimates points out an important shortcoming of zero-mean acceleration models as in (25) or exponentially time-correlated process descriptions. The constant turn-rate model (26) has been shown to be more representative of many airborne targets at close range [25,26], so current efforts are considering its incorporation into the filter structure.

7. CONCLUSION

A simple four-state extended Kalman filter has been developed to track a distant point-source target with benign dynamics, using outputs from a forward-looking infrared (FLIR) sensor as measurements. As shown in part by Figs. 3 and 4, it consistently outperforms the currently used correlation tracker, with the most substantial improvement being gained in scenarios with low signal-to-noise ratio and/or small

target spot size relative to detector (pixel) size. The filter exploits knowledge unused by the correlation tracker - size, shape and motion characteristics of the target, atmospheric jitter spectral description, and background and sensor noise characteristics - to yield the enhanced performance. However, robustness studies have revealed a serious degradation in tracking performance when the filter-assumed model does not represent the actual tracking environment well, indicating appropriate design modifications to generate a filter capable of tracking less benign targets in a realistic and more uncertain environment.

Figs. 7 and 8 are indicative of performance capabilities of the resulting eight-state adaptive tracker for realistic but not overly harsh trajectories of a missile viewed at short range. Good tracking performance is achieved on the basis of estimating target velocity and acceleration as well as position, assuming elliptical intensity profile contours with major axis aligned with the estimated velocity vector, and adaptively estimating I_{max}, σ_{g1}, σ_{g2}, and Q_d. To address more dynamic environments, detection and appropriate response to maneuver initiation have been investigated with some success, but rather tenuous ad hoc modifications were required due to assumed zero-mean acceleration models. Current research is concentrated on better target acceleration models and on different filter forms to exploit these models, such as the multiple model adaptive filtering algorithm that probabilistically weights the outputs of a bank of filters, each based on one of the alternative models. Also, work is being accomplished on a generalization in which target intensity patterns are uncertain or irregular enough to discount a bivariate Gaussian model and instead to preprocess the measurements to provide the entire $\underline{h}$ function adaptively to the filter, as through spatial modal decomposition.

ACKNOWLEDGMENTS

Figs. 1 and 5-8 are from Ref. [12] and Figs. 2-4 are from Ref. [8], with permission of the IEEE.

REFERENCES

[1] Richards, C. L., "Correlation Tracking Algorithm," SAMRT-76-0076 Eng. Data Release, Aeronutronic Ford Corp., Newport Beach, Cal., July 2, 1976.

[2] Richards, C. L., "Results of HAWK Image Tracking Experiment," SAMRT-76-0087 Eng. Data Release, Aeronutronic Ford Corp., Newport Beach, Cal., Aug 17, 1976.

[3] Richards, C. L., "Correlation Tracking Software," SAMRT-76-0088 Eng. Data Release, Aeronutronic Ford Corp., Newport Beach, Cal., Aug 17, 1976.

[4] Richards, C. L., "Precision Line and Point Reticle Location," SAMRT-77-0006 Tech. Data Release, Ford Aerospace and Commun. Corp., Newport Beach, Cal., Mar 8, 1977.

[5] Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

[6] Maybeck, P. S., Stochastic Models, Estimation and Control, Vol. 1, Academic Press, New York, 1979.

[7] Maybeck, P. S., Stochastic Models, Estimation and Control, Vols. 2 and 3, Academic Press, New York, 1981.

[8] Maybeck, P. S., and D. E. Mercier, "A Target Tracker Using Spatially Distributed Infrared Measurements," IEEE Trans. Automat. Control, Vol. AC-25, No. 2, pp 222-225, April 1980.

[9] Mercier, D. E., "An Extended Kalman Filter for Use in a Shared Aperture Medium Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Dec 1978.

[10] Maybeck, P. S., D. A. Harnly and R. L. Jensen, "Robustness of a New Infrared Target Tracker," Proc. IEEE Nat. Aerospace and Electronics Conf., Dayton, Ohio, pp 639-644, May 1980.

[11] Jensen, R. L., and D. A. Harnly, "An Adaptive Distributed-Measurement Extended Kalman Filter for a Short Range Tracker," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Dec. 1979.

[12] Maybeck, P. S., R. L. Jensen and D. A. Harnly, "An Adaptive Extended Kalman Filter for Target Image Tracking," IEEE Trans. Aerospace and Electron. Sys., Vol. AES-17, No. 2, pp 173-180, March 1981.

[13] Hogge, C. B., and R. R. Butts, "Frequency Spectra for the Geometric Representation of Wavefront Distortions due to Atomospheric Turbulence," IEEE Trans. Antenna and Propagat., Vol. AP-24, pp 144-154, Mar 1976.

[14] "Advanced Adaptive Optics Control Techniques," Tech. Rept. TR-996-1, The Analytic Sciences Corp., Reading, Mass., Jan 1978.

[15] Safonov, M. G., and M. Athans, "Robustness and Computational Aspects of Nonlinear Stochastic Estimators and Regulators," IEEE Trans. Automat. Control, Vol. AC-23, No. 4, pp 717-725, Aug 1978.

[16] Athans, M., R. P. Wishner and A. Bertolini, "Suboptimal State Estimators for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements," IEEE Trans. Automat. Control, Vol. AC-13, No. 5, pp 504-518, Oct 1968.

[17] Åström, K. J., and P. Eykhoff, "System Identification - A Survey," Automatica, Vol. 7, pp 123-162, 1971.

[18] Athans, M., and C. B. Chang, "Adaptive Estimation and Parameter Identification Using Multiple Model Estimation Algorithm," Tech. Note 1976-28, ESD-TR-76-184, Lincoln Lab., Lexington, Mass., June 1976.

[19] Isermann, R., U. Baur, W. Bamberger, P. Kneppo, and H. Siebert, "Comparison of Six On-Line Identification and Parameter Estimation Methods," Automatica, Vol. 10, pp 81-103, 1974.

[20] Kailath, T. (ed.), Special Issue on System Identification and Time-Series Analysis, IEEE Trans. Automat. Control, Vol. AC-19, No. 6, Dec. 1974.

[21] Maybeck, P. S., "Combined Estimation of States and Parameters for On-Line Applications," Ph.D. Disseration, Massachusetts Inst. Technology, and Tech. Rept. T-557, C. S. Draper Lab., Cambridge, Mass., Feb. 1972.

[22] Maybeck, P. S., "Parameter Uncertainties and Adaptive Estimation," unpublished, Air Force Inst. of Tech., Wright-Patterson AFB, Ohio, 1978.

[23] Saridis, G. N., "Comparison of Six On-Line Identification Algorithms," Automatica, Vol. 10, pp 69-79, 1974.

[24] Yared, K. I., "On Maximum Likelihood Identification of Linear State Space Models," Ph.D. Dissertation, LIDS-TH-920, Massachusetts Inst. Technology, Cambridge, Mass., July 1979.

[25] "Firefly III IFFC Fire Control System," Tech. Rept. 19008 ACS 12004, General Electric Co., Aircraft Equipment Div., Binghamton, N. Y., Dec. 1979 (revised, Jan. 1981).

[26] Worsley, W. H., "Comparison of Three Extended Kalman Filters for Air-to-Air Tracking," M.S. Thesis, Air Force Inst. of Tech., Wright-Patterson AFB, Ohio, Dec. 1980.

[27] Abramson, P. D., Jr., "Simultaneous Estimation of the State and Noise Statistics in Linear Dynamic Systems," Ph.D. Dissertation, Massachusetts Inst. Tech., Rept. TE-25, Cambridge, Mass., May 1968.

[28] Belanger, P. R., "Estimation of Noise Covariance Matrices for a Linear Time-Varying Stochastic Process," Automatica, Vol. 10, pp 267-275, 1974.

[29] Mehra, R. K., "Approaches to Adaptive Filtering," IEEE Trans. Automat. Control, Vol. AC-17, pp 693-698, Oct. 1972.

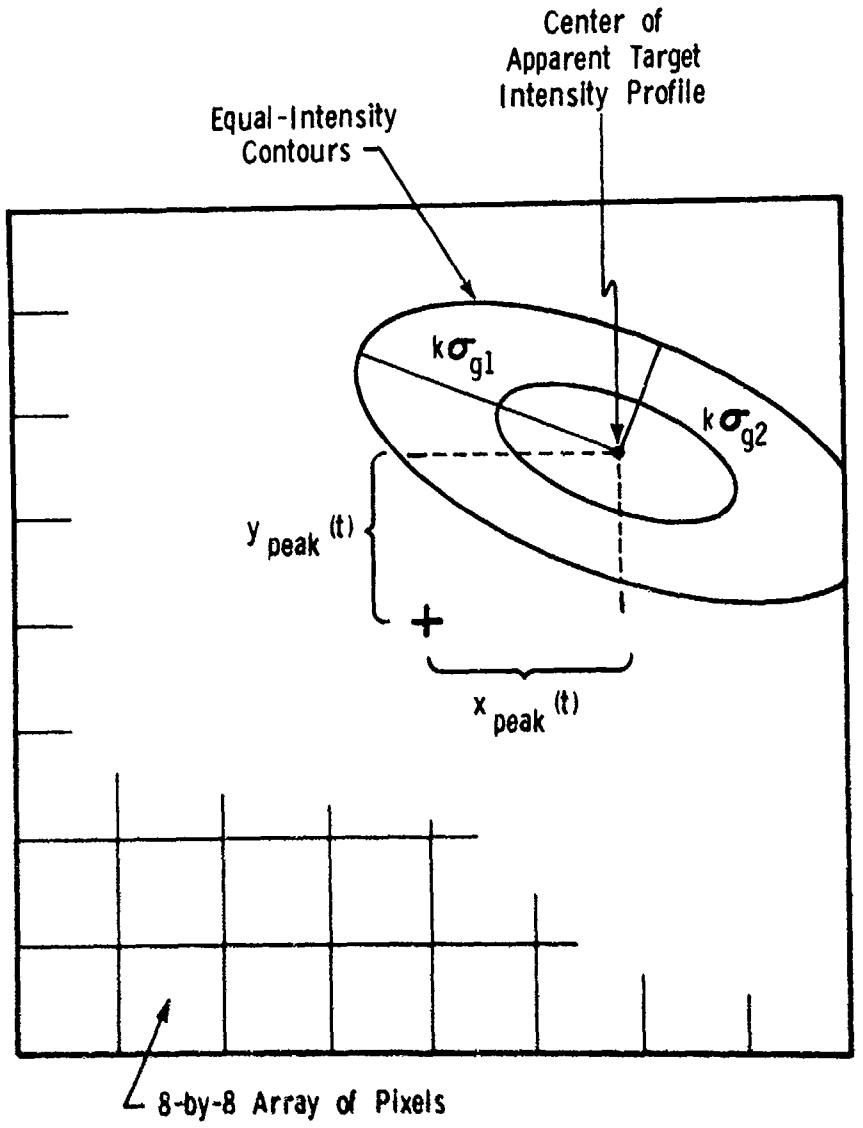


Fig. 1. Apparent Target Intensity Pattern on Image Plane.

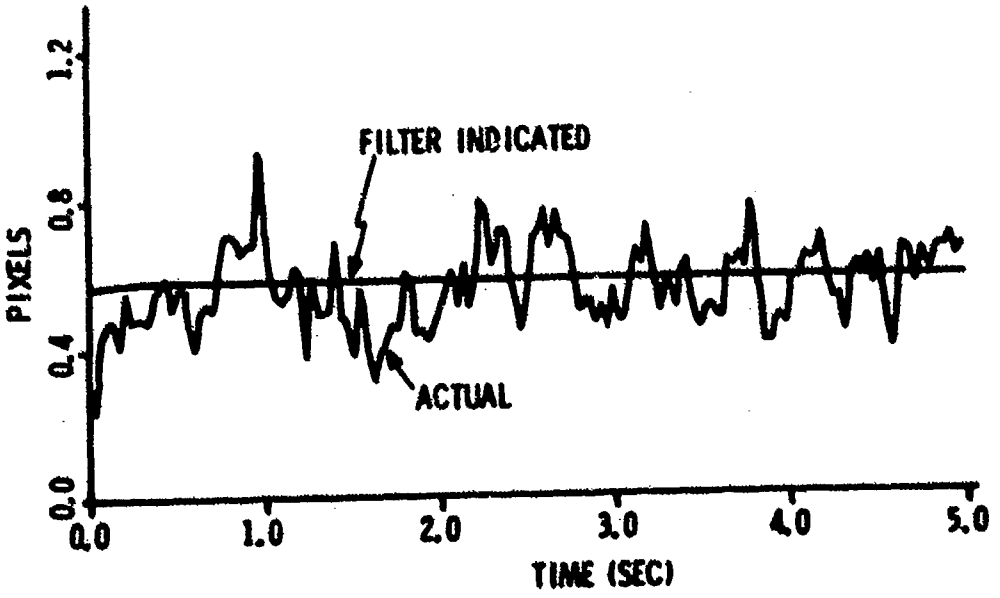


Fig. 2. $\dot{R}_a$ Error Standard Deviation: Actual Versus Filter-Computed ($S/N = 10$, $\sigma_g = 3$, $\sigma_d = \sigma_a = 1$, $T_d = 1$).

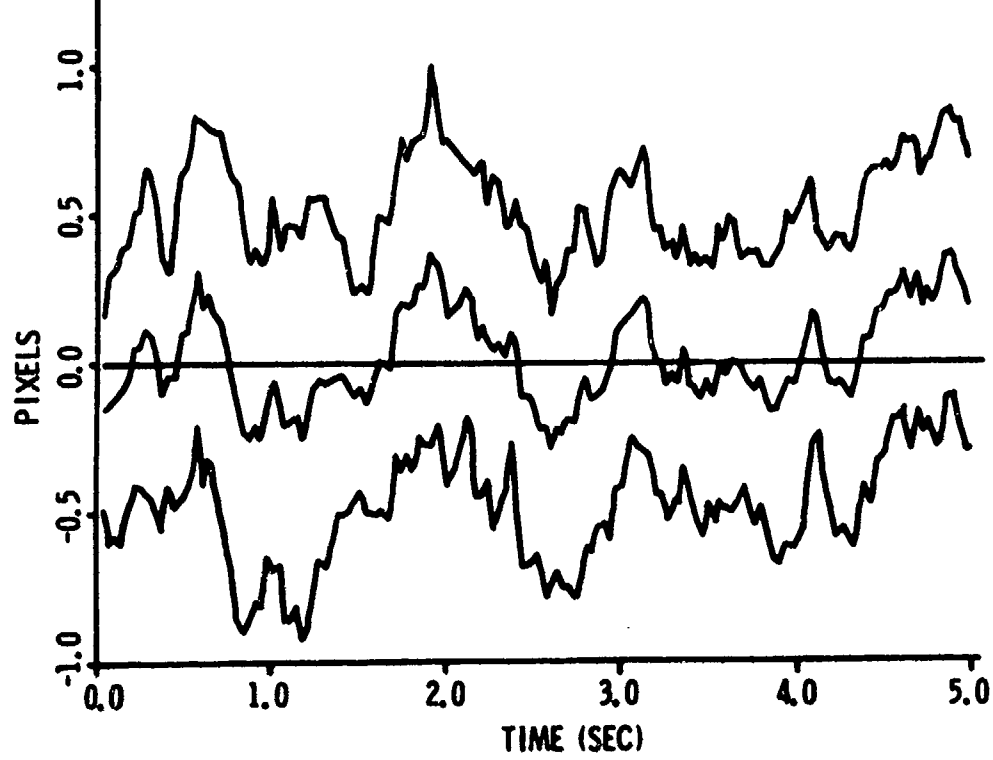


Fig. 3. $\hat{x}_d$ Mean Error $\pm 1\sigma$ (S/N = 10, $\sigma_g = 3$, $\sigma_d = \sigma_a = 1$, $T_d = 1$).

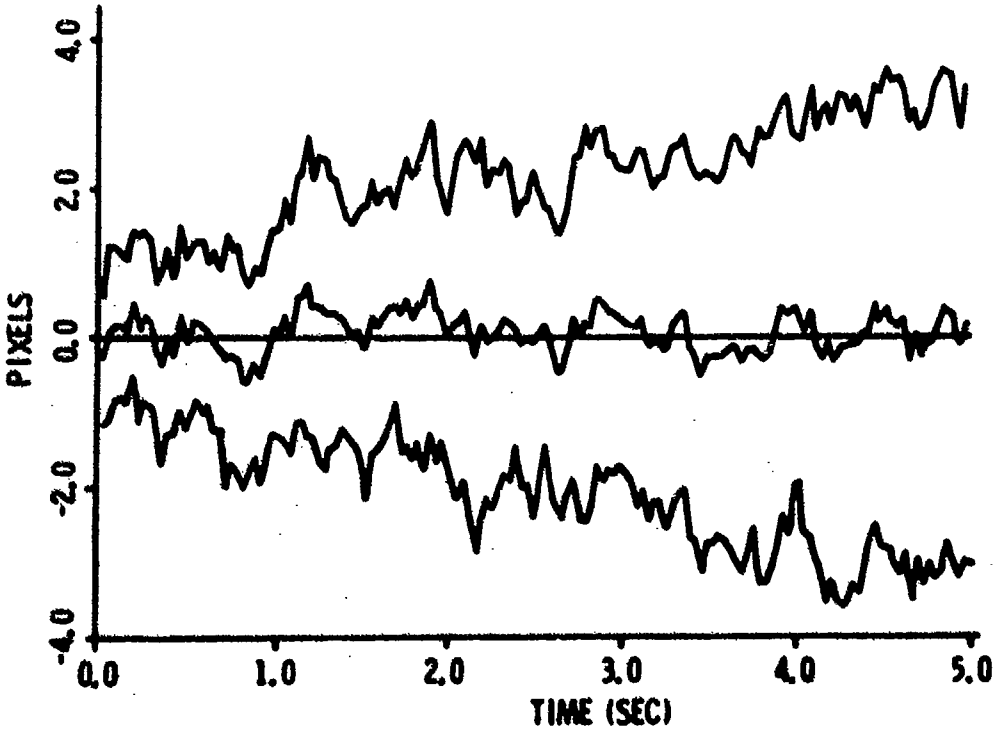


Fig. 4. Correlator Target Horizontal Position Mean Error $\pm 1\sigma$ (S/N = 10, $\sigma_g = 3$, $\sigma_d = \sigma_a = 1$, $T_d = 1$).

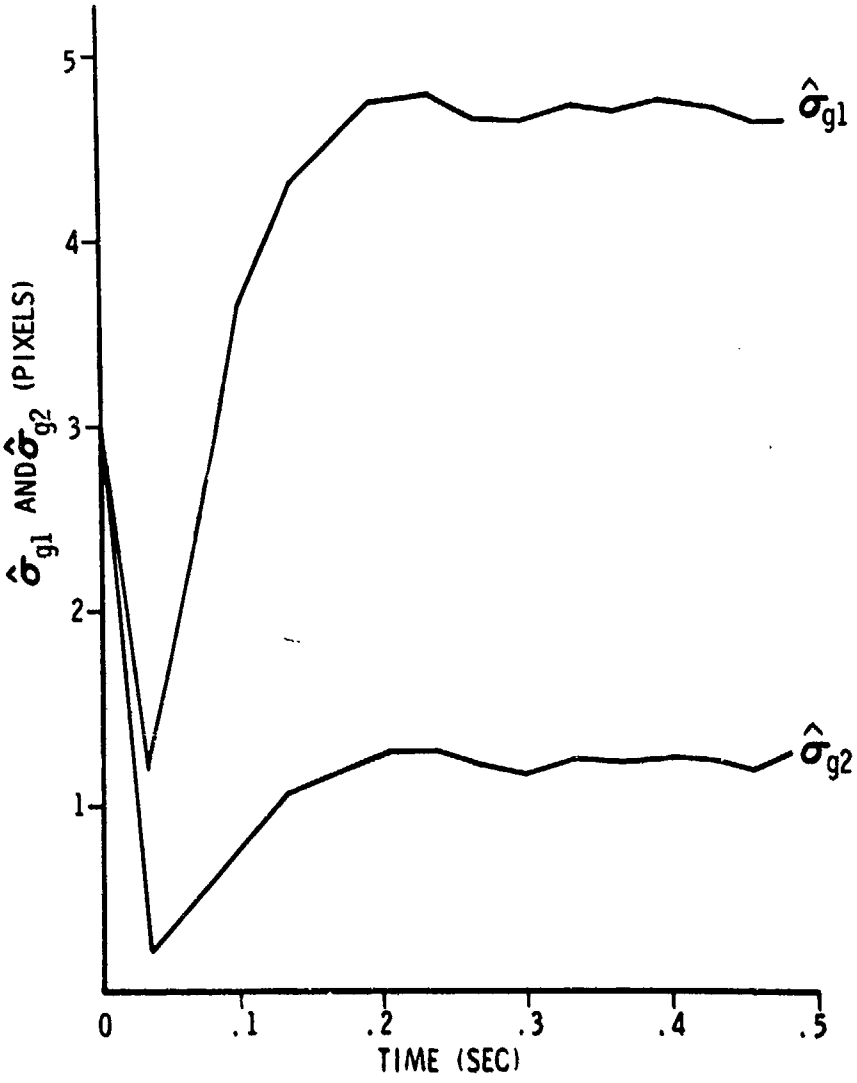


Fig. 5. Sample of Estimates of σ_{g1} and σ_{g2}.

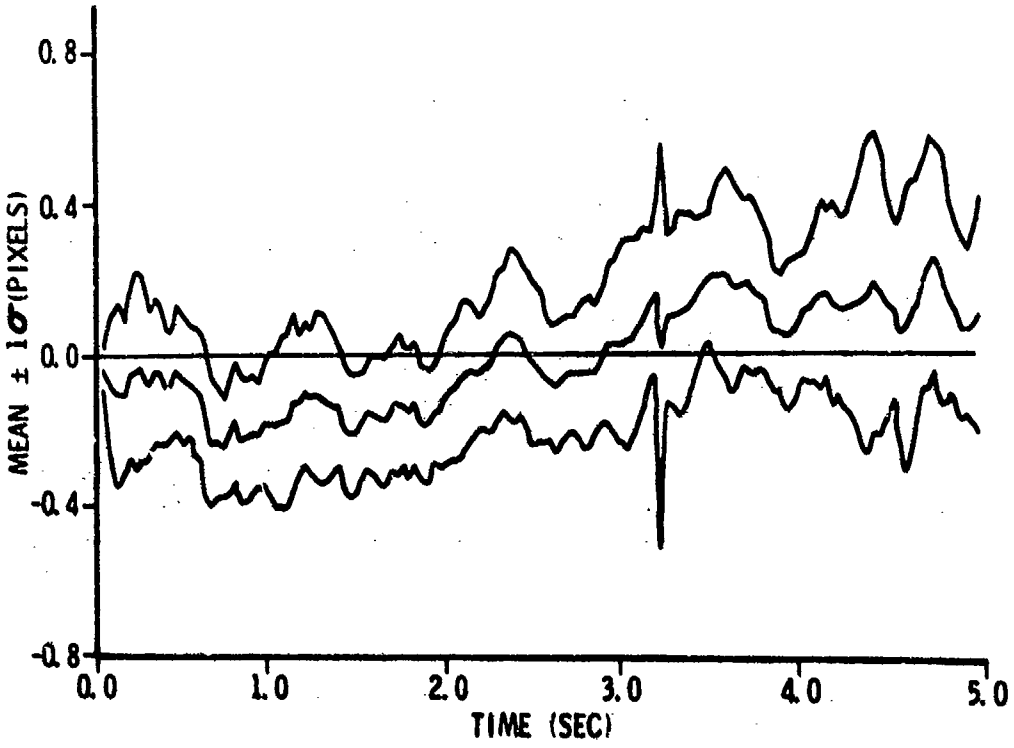


Fig. 6. Target Horizontal Position Error Mean ±1σ: Six-State Filter.

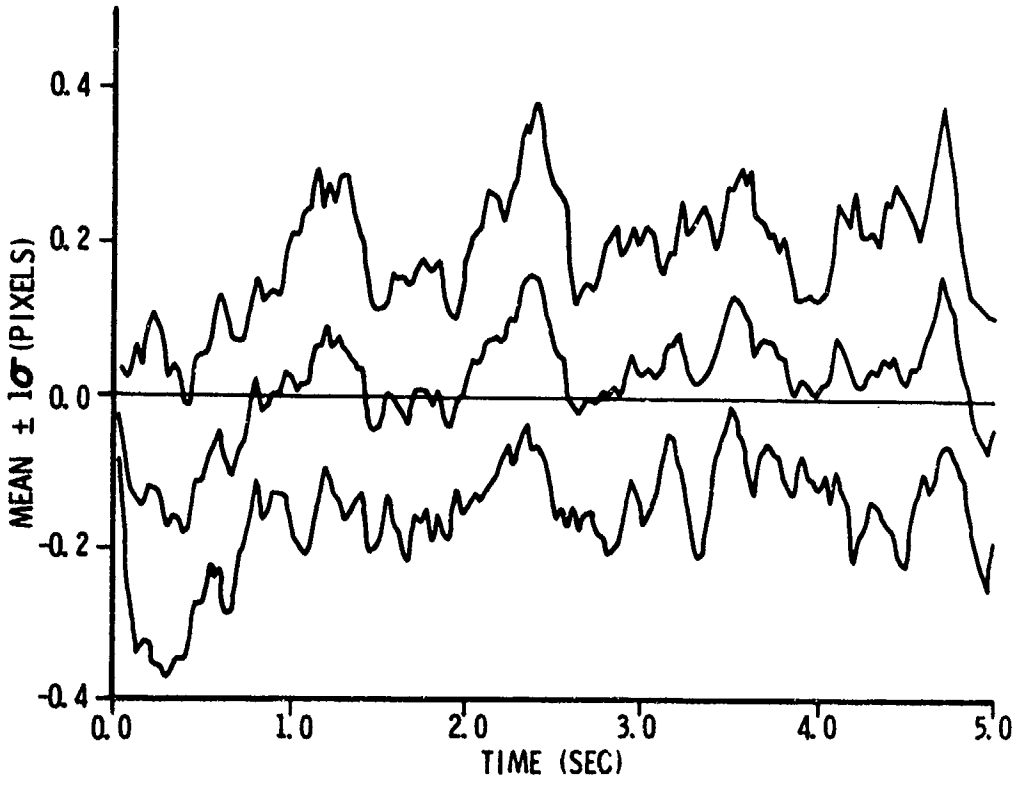


Fig. 7. Target Horizontal Position Error Mean ±1σ: Eight-State Filter.

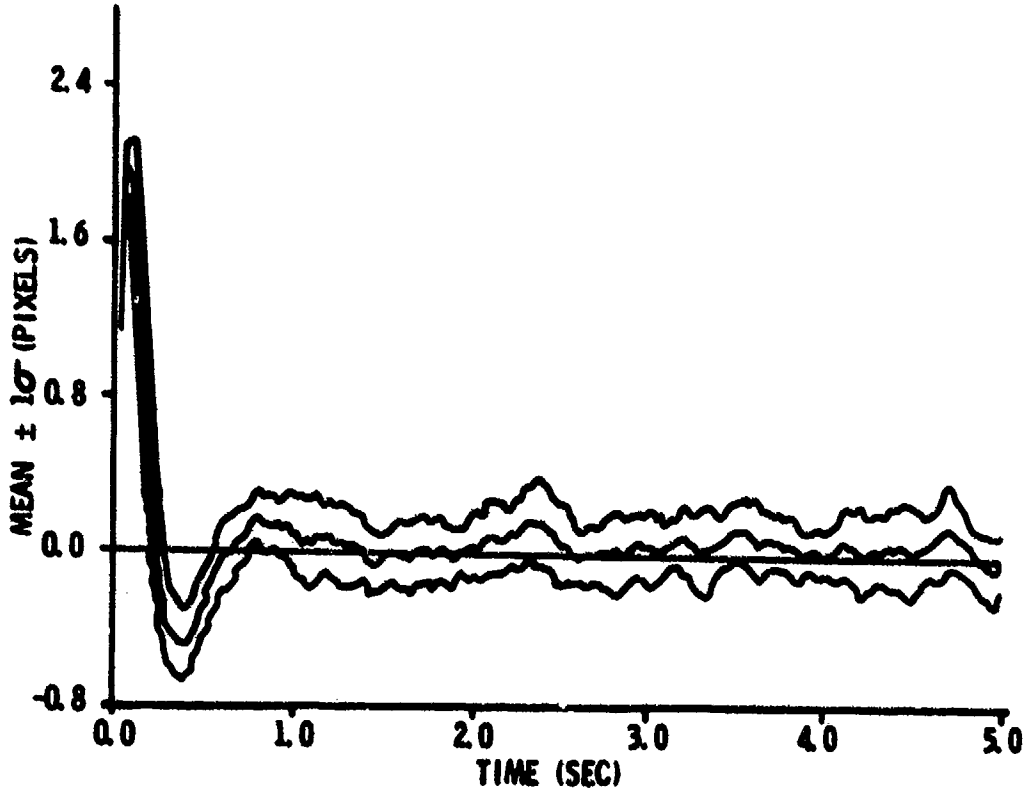


Fig. 8. Target Horizontal Position Error Mean ±1σ: Eight-State Filter; Recovery from Initialization Errors.

TECHNIQUES FOR THE DEVELOPMENT OF ERROR MODELS FOR AIDED STRAPDOWN NAVIGATION SYSTEMS

by
Dr. W. Lechner
D F V L R
Institut für Flugführung
3300 Braunschweig
West-Germany

SUMMARY

In order to increase the accuracy levels of the inertial navigation systems (INS) additional external measurements were used, such as those provided by radar equipment, TACAN facilities, MLS and so on. The combination of the different data is often carried out by using the methods of Kalman filtering, which need sufficiently exact error models especially for the INS. As far as strapdown systems (SDS) are concerned the development of mathematical descriptions of the error behaviour leads to high-order models, because the dynamic environment of the SDS has to be taken into account. However, for real-time navigational computations it is necessary to provide low-order error models. In the paper account is given of how adaptive low order error models were developed and adaptive filtering applied. The results were checked by using measured and simulated SDS data.

LIST OF SYMBOLS

SUBSCRIPTS/SUPERSUBSCRIPTS

b	body-fixed axes
n	navigational axes
i	inertial reference coordinate frame
e	earth-fixed axes
x, y, z	long, cross, vertical directions
N, E, D	north, east, down directions
ac	accelerometer
gy	gyro
$\underline{symbol}$	vector expression
$\underline{\underline{symbol}}$	matrix
$\overline{symbol}^{*}$	true value + error

NAVIGATIONAL SYMBOLS

$\underline{\underline{C}}_{nb}$	transformation matrix
ϕ, θ, Ψ	bank, pitch and azimuth angles
$\underline{\omega}$	angular rates
$\underline{a}$	accelerations
$\underline{v}$	velocities
φ	geographical latitude
λ	geographical longitude
h	height
R_N, R_E	earth radii
g	gravity
$\Delta\underline{\phi}_b$, $\Delta\underline{v}_b$	angular or velocity increments
$\underline{\omega}^{ie}$	earth angular rate
χ	track angle
γ	flight path angle
T_a	sampling period
t	time

SENSOR ERROR COEFFICIENTS

c^{sy}	axes misalignment
D^{0}	fixed drifts
D^{g}	mass unbalance drifts
D^{gg}	anisoelasticity drifts
D^{v}	fixed scale factor errors
D^{vv}	quadratic nonlinearity scale factor errors
D^{sq}	asymmetric scale factor errors
$D^{\dot{\omega}}$	angular acceleration drifts
D^{Jw}	anisoinertia drifts
c^{ac}	axes misalignment
B^{0}	bias
B^{v}	fixed scale factor error
B^{sq}	asymmetric scale factor error
B^{vv}	quadratic nonlinearity scale factor error
B^{vvv}	cubic nonlinearity scale factor error
B^{xy}	cross coupling

H^{gy} angular momentum of the rotor
A^{gy} rotor transverse moment of inertia
C^{gy} rotor polar moment of inertia

$\delta hbar$ bias of the barometric altimeter

SYMBOLS FOR THE FILTER ALGORITHMS

$\underline{x}$ state vector
$\hat{\underline{x}}$ estimated state vector
$\tilde{\underline{x}}$ estimation error
$\underline{A}$ state space notation of the error model (transitions matrix)
$\underline{\Phi}$ time-discrete transition matrix
$\underline{R}$, $\underline{P}$, $\underline{Q}$ covariance matrices
$\underline{K}$ gain matrix
$\underline{x}'$, P' predicted state vector or covariance matrix
$\underline{H}$ measurement matrix
q_{ee}, q_{vv} square root of the diagonal elements of the matrix Q

MATHEMATICAL OPERATIONS

E expectation operator
$\underline{c} = \underline{\omega} \times \underline{I}$ cross-product
$\underline{I}$ identity matrix
δ error term
Δ difference term
$f(x)$ function of x
$\dot{x}$ dx/dt
$\underline{x}^T$, $\underline{P}^T$ transponse of a vector or matrix
$\bar{x}$ mean value

1. INTRODUCTION

Inertial navigation systems (INS) are used in civil and military aviation and are needed in all cases where autonomous navigation is essential (space flights, missiles, navigation on land and at sea). Up to now most of the INS have had a platform mechanisation: gyros and accelerometers are mounted on a platform that is isolated cardanically from the angular motions of the carrier. A double integration of the accelerometer signals gives the ground speed and the position of the carrier. The heading and attitude angles are contained in the directions of the gimbals.

If the sensors are "strapped down" on the carriers directly no gimbals and servomotors are necessary. This type of INS mechanisation is called a strapdown system (SDS). The accelerometer signals measured in a body-fixed coordinate frame are transformed to a navigational reference frame by means of the gyro signals. This results in the following advantages in comparison with the so-called platform systems /1/:
- simple mechanical construction
- the provision of accelerations and angular rates in body-fixed axes
- easy maintenance due to the modular construction
- the economical provision of redundancy by means of skewed sensitivity axes.

However, against these advantages must be weighed certain drawbacks:
- increased demands on the efficiency of the navigation computer
- extreme demands on the accuracy of the sensors, which have to measure the full dynamic environment of the SDS. In contrast to the platform mechanisation, the gyros do not operate as sensors for zero signals.

The accuracy levels of the inertial systems are frequently insufficient on their own, and for this reason additional external measurements are used, such as those provided by radar equipment, TACAN facilities, MLS and so on. The strapdown signals are exact in a short time period but the navigation errors increase with the time. The combination of this INS error behaviour with the complementary error statistics of the radio signals is often carried out using the methods of optimal filtering. Here it is important that the dynamic error behaviour of the SDS should be carefully modelled, i.e. described mathematically. The conformity of the error model with the real error behaviour of the SDS is of major importance for the accuracy level of an aided SDS. In the development of suitable error models it is necessary to find a compromise between, on the one hand, a mathematically simple description of the error behaviour that satisfies real-time computation requirements and as far as possible creates no numerical problems and, on the other hand, a determination of the actual error behaviour, including all important sources of error, which is as accurate as possible. In comparison with the platform systems it is far more difficult to find such a compromise in the case of SDS, since the sensor errors, and therefore the system errors as well, depend to a very large extent on the dynamic environment of the SDS. According to the structure of a particular flight path it is possible for a few isolated sensor errors or a large number of different sensor errors to have an effect at system level. The structure of the flight path can change several times during a flight, e.g. if there are sections with only a few manoeuvres or even with a straight flight, or sections with extreme manoeuvres like the terrain following flights. An error model that conforms to the real SDS error behaviour for all possible cases will thus generally lead to an unacceptably sophisticated error model with regard to the real-time computations or numerical problems involved. In this paper the attempt is made to

solve the problem by using adaptive error models and adaptive filtering, thereby avoiding some of the drawbacks of SDS. As an example measured and simulated strapdown data were used to demonstrate and to check the methods suggested here.

2. TECHNIQUES FOR THE DEVELOPMENT AND CHECKING OF SDS ERROR MODELS

The error model that will be developed can only be checked properly if adequate criteria are available to check the conformity between the model and real world. A high-order error model often serves as a reference for discussing a simplified low-order error model that has been derived from it. The differences between the two error models can then be interpreted statistically in terms of covariance matrices.

A disadvantage of these techniques lies in the assumption of a linearized high-order model as a reference model, which leads to problems in the case of SDS. It is scarcely possible to express nonlinear algorithm errors of the strapdown navigation equations, for instance, in a linear state space notation. In these conditions it proved to be better to include nonlinear effects in a SDS simulation and to interpret the results of the simulation as a reference system corresponding to the real error behaviour. It is then possible to compare the sensor and system errors estimated by a Kalman filter that is to operate on the basis of simplified error models with the known errors derived from the simulation results. It is also possible to check the covariance matrix $\underline{P}$ from the self-diagnosis of the Kalman filter by determining the expectation values with reference to the differences from the simulated and estimated errors.

In the subsequent chapters both methods are applied in a complementary manner. The construction of the simulation of a SDS is described in detail in /2/.

2.1 GENERAL ERROR EQUATIONS OF STRAPDOWN NAVIGATION SYSTEMS

The derivation of the strapdown error model is based on the navigation equations, a knowledge of which is assumed here /3, 4/. The equations can be given by

$$\dot{\underline{C}}_{nb} = \underline{C}_{nb} \left[\underline{\Omega}_b^{ib} - \underline{\Omega}_b^{in}\right] \tag{1}$$

$$\underline{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2}$$

$$\dot{\underline{v}}_n = \underline{C}_{nb}\,\underline{a}_b - (2\,\underline{\omega}_n^{ie} + \underline{\omega}_n^{en}) \times \underline{v}_n + \underline{g} \tag{3}$$

$$\underline{\omega}_n^{ie} = (\omega_o \cos\varphi\ ,\ 0\ ,\ -\omega_o \sin\varphi)^T \tag{4}$$

$$\underline{\omega}_n^{en} = \left(\frac{V_E}{R_E}\ ,\ -\frac{V_N}{R_N}\ ,\ -\frac{V_E}{R_E}\tan\varphi\right)^T \tag{5}$$

$$\varphi = x_N/R_N \tag{6}$$

$$\lambda = x_E/R_E/\cos(\varphi) \tag{7}$$

The individual navigation equations are differentiated with respect to the errors

$$\delta\underline{C}_{nb}\ ,\ \delta\underline{v}_n\ ,\ \delta\varphi\ ,\ \delta\lambda\ ,\ \delta h\ ,\ \delta\underline{a}_b\ ,\ \delta\underline{\omega}_b \tag{8}$$

and the following relation is used

$$\delta\underline{C}_{bn} = -\left[\underline{\varepsilon}_n \times \underline{I}\right] \cdot \underline{C}_{nb} \tag{9}$$

Then, after carrying out numerous transformations, approximations and rearrangings, one obtains the linearized system error model in a state space notation. The vertical speed and the vertical position must be aided by the signals of a barometric altimeter /5/. The causes of sensor errors that can be identified are listed in Tab. 1 below. The values given here are based on the data from the TELEDYNE SDG5 gyro and the SYSTRON DONNER 4833A-1PX accelerometer, and are interpreted as incapable of further compensation.

type of gyro error	ε^{gy}	D^{0}	D^{g}	D^{gg}	D^{ω}	$D^{\omega\omega}$	$D^{\lvert\omega\rvert}$	$D^{\Delta\omega}$	$D^{2\omega}$
magnitude	6arcs	0,01°/h	0,02°/h/g	0,03°/h/g^2	$3\cdot10^{-5}$	$3\cdot10^{-5}$	$3\cdot10^{-6}$	$4\cdot10^{-4}$s	$4\cdot10^{-4}$s
dependent on	ω	-	a	a^2	ω	ω^2	$\lvert\omega\rvert$	$\dot{\omega}$	ω^2

type of acclerometer error	ε^{ac}	B^{0}	B^{g}	$B^{\lvert g\rvert}$	B^{gg}	B^{3g}	B^{2g}
magnitude	6arcs	10^{-4}g	10^{-5}	10^{-5}	10^{-5}/g	10^{-6}/g^2	10^{-5}/g
dependent on	a	-	a	$\lvert a\rvert$	a^2	a^3	a^2

Tab. 1: Sensor error coefficients

The sensor errors listed in Tab. 1 cause misalignments or velocity errors at the system error level corresponding to the integrals

$$\underline{\varepsilon}_n = \int_0^t \underline{\underline{C}}_{nb}(t) \cdot \delta\underline{\omega}_b(t) \cdot dt \tag{10}$$

$$\delta\underline{v}_n = \int_0^t \underline{\underline{C}}_{nb}(t) \cdot \delta\underline{a}_b(t) \cdot dt \tag{11}$$

Some of the sensor errors can be ignored as regards their effects at system level in the case of realistic flight paths in which high angular rates or high accelerations only occur in periods that are substantially shorter than the total flying time. As regards the unbalance drift of the gyros

$$\delta\underline{\omega}_b = \begin{bmatrix} D_x^{g} & 0 & 0 \\ 0 & D_y^{g} & 0 \\ 0 & 0 & D_z^{g} \end{bmatrix} \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{12}$$

a misalignment corresponding to the fixed drift of 0.01 °/h only occurs if the horizontal accelerations a_x, a_y are 0.5 g for the whole duration of the flight. In realistic flight paths this is only possible by means of constant manoeuvring. However, this requires high angular rates which, in combination with the misalignment of the sensitivity axes of the gyro

$$\delta\underline{\omega}_b = \begin{bmatrix} \varepsilon_x^{gy} & 0 & 0 \\ 0 & \varepsilon_y^{gy} & 0 \\ 0 & 0 & \varepsilon_z^{gy} \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{13}$$

contribute so much to the misalignment at system level that the unbalance drifts can be ignored. Thus an angular rate of only 1 °/sec produces, with a misalignment of 6 arcs, a drift of approx. 0.1 °/h according to Eq.(13). The drift caused by the vertical component is a constant 0.02 °/h in flights that contain fewer manoeuvres and can be considered as a fixed drift of the gyros. A corresponding consideration holds good for the anisoelasticity drifts. This requires constant horizontal accelerations of 0.3 g to produce a drift of 0.01 °/h, and can therefore also be ignored at system level.

The error model for the accelerometers can also be simplified considerably for realistic flight path on the basis of Eq.(11). In Fig. 1 the individual errors are represented as functions of the maximum accelerations that occur. For accelerations up to a maximum of 1 g it is sufficient to model the bias and the axes misalignment of the accelerometers, since even with a constant acceleration of 1 g the errors are still ten times smaller.

Similar considerations for all important sensor errors leave 17 gyro and 6 accelerometer errors, i.e. a total of 23 sensor errors. The error model shown in Fig. 2 contains a total of 33 state variables which can be subdivided as follows:

3 angular errors
3 velocity errors
3 position errors
} 9 system errors

1 bias of the barometric altimeter
} 1 altimeter error

3 axes misalignments
3 fixed drifts
3 fixed scale factor errors
3 asymmetry scale factor errors
3 quadratic nonlinearity scale factor errors
1 angular acceleration drift
1 anisoinertia drift
} 17 gyro errors

3 axes misalignments
3 bias errors
} 6 acceleration errors

In contrast to a platform mechanisation, the sensor errors now produce the errors at system level via the transformation matrix $\underline{C}_{nb}$. $\underline{C}_{nb}$ itself is a trigonometric function of the actual attitude and heading angles. In addition the sensor errors depend on the dynamic environment of the SDS. All together this leads to a total error behaviour of a SDS that depends to a very large extent on the actual manoeuvres flown. The number of sensor errors having an effect at system level varies considerably according to the particular flight path.

2.2 PROCEDURES FOR THE EXAMINATION OF ERROR MODELS FOR STRAPDOWN SYSTEMS

Fig. 3 shows the construction of the software for the examination of error models for SDS. It consists essentially of 4 parts:
- the simulation of an unaided SDS, including the calculation of the reference flight path,
- the design of simplified low-order models by means of the covariance analysis methods and on the basis of simulated as well as measured sensor signals,
- the Kalman filter that contains simplified low-order error models for the SDS and at least
- the comparison of assumed sensor errors, or of system errors determined from the simulation, with errors estimated by Kalman filtering.

The simulation of the SDS shown in the upper part of Fig.(3) begins with the selection of tracks, flight path angles and velocities. Based on this input the sensor signals are calculated for the gyros or the accelerometers in terms of angle or velocity increments and are then fed into the sensor error model. The corresponding errors of position, velocity and attitude and heading angles are then obtained by comparing the results of the subsequent navigation calculation with the data of the reference flight path. All these values are stored on a magnetic disc for further use. The software for the simulation of a SDS has been developed at DFVLR and a detailed description is contained in /2/.

For the design of simplified low-order models $\underline{\phi}_1$, the high-order reference error models $\underline{\phi}$ were used and for both the discrete error and covariance propagation equations were calculated.

$$\underline{x}_1'(k+1) = \underline{\phi}_1(k) \cdot \underline{x}_1(k) \tag{14}$$

$$\underline{P}_1'(k+1) = \underline{\phi}_1(k) \cdot \underline{P}_1(k) \cdot \underline{\phi}_1^T(k) + \underline{Q}_1(k) \tag{15}$$

$$\underline{x}'(k+1) = \underline{\phi}(k) \cdot \underline{x}(k) \tag{16}$$

$$\underline{P}'(k+1) = \underline{\phi}(k) \cdot \underline{P}(k) \cdot \underline{\phi}^T(k) + \underline{Q}(k) \tag{17}$$

The statistical interpretation of the difference between the corresponding components of the state vectors $\underline{x}_1$ and $\underline{x}$ leads to the design of the system noise matrix $\underline{Q}$.

The Kalman filter algorithms that now follow use the data from the simulated SDS and are based on the simplified low-order models developed by covariance analysis calculations. The state vector covers 3 position errors, 3 velocity errors and 3 angular errors, as well as a certain number of sensor errors, such as gyro drifts, scale factor errors, accelerometer bias etc. The measurements required for the Kalman filter can be obtained from the position signals of the simulated reference flight path, which is known exactly. In that case the reference flight path is considered as an external measurement carried out by a tracking radar system. Together with the initial covariance matrix $\underline{P}$, the system noise matrix $\underline{Q}$, and the covariance matrix assumed for the simulated measurements $\underline{R}$, the Kalman filter algorithms have all the necessary output data at their disposal. The principles of Kalman filtering for aided inertial navigation systems have been described in detail in /6, 7, 8/. The result of the filtering is the estimation vector $\hat{\underline{x}}$ and the covariance matrix $\underline{P}$.

At the end of the programme system the comparison is made between the assumed sensor errors, or system errors determined from the simulated SDS, and the estimated errors based on simplified low-order models. The differences determined in this way correspond to the so-called "true estimation errors" and, according to the theory of Kalman filtering, must correspond with the predicted covariance matrix $\underline{P}$. This relationship is expressed by Eq.(18)

$$\underline{P} = E\left[\tilde{\underline{x}}\tilde{\underline{x}}^T\right] \tag{18}$$

Roughly speaking the Eq.(18) means that the estimation errors must lie within the 1σ-limits defined by the diagonal elements of the covariance matrix $\underline{P}$. The aim when deriving simplified low-order models is to use as few state variables as possible, while still satisfying Eq.(18), and in so doing to obtain acceptable values for the 1σ-limits.

2.3 THE SELECTION OF 3 TYPICAL FLIGHT PATHS

The decisive factor in the error behaviour of a SDS is the dynamic environment of the system, and for this reason 3 examples of typical flight paths will be given to illustrate this error behaviour. Fig. 4 contains the flight paths and the corresponding flight levels, and Fig. 5 shows the simulated and measured bank-rates and cross-accelerations.

FLIGHT PATH 1

Flight path 1 is a simulated straight flight of approx. 100 min duration, containing 4 full turns. The flight path is at a constant altitude and the ground speed is approx. 300 kts. This flight path is selected to simulate the structure of the flight path of a civil airliner flying from a point A to a point B as directly as possible. Angular rates and accelerations only occur during the initiation and conclusion of the turns and during take-off and landing, and their values then are approx. $\pm$ 15 °/s or $\pm$ 0.3 g.

FLIGHT PATH 2

Flight path 2 was flown by a F104 combat aircraft. A SDS made by the TELEDYNE company was part of the instrumentation system /9/. The path contains several manoeuvres carried out at a speed of approx. 700 kts. The profile of the altitude contains climb and descent phases, with descent speeds up to approx. 100 kts. These manoeuvres produce bank-rates of up to 80 °/s and cross-accelerations of up to 0.3 g. Noise terms whose standard deviations can be estimated at approx. 7 °/s or 0.1 g are superimposed on the sensor signals /10/.

FLIGHT PATH 3

Flight path 3 was the result of a sophisticated simulation /2/ and is selected to represent a remotely piloted vehicle (RPV) flight path. This is shown clearly by the altitude profile, which contains a simulation of "terrain following". The flight path is based on the following parameters:

- catapult take-off with a flight path angle of 20° and acceleration to a speed of 275 m/s,
- total flying distance approx. 280 km, flying time approx. 40 min,
- altitude profile during the mission: high-low-low-high, with a total of 2 low-level stages (schematized terrain following) having a total length of approx. 31 km,
- vertical manoeuvres of up to approx. + 2 g,
- horizontal turns of up to approx. + 3 g.

These extreme manoeuvres produce bank-rates of up to $\pm$ 50 °/s, and accelerations of up to 5 g.

3. THE DERIVATION OF ADAPTIVE ERROR MODELS

It is only possible to model a few sensor error coefficients in a low-order model that is suited to real-time applications within the Kalman filter algorithms. For this reason, if one only models the fixed gyro drift, for instance, the actual error behaviour would only be adequately described during a section without manoeuvres. During a full turn, for example, a misalignment to the gyro sensitivity axes of 6 arcs at an angular rate of only 360 °/3 min produces a system misalignment after one turn of

$$\delta d_b = \varepsilon^{gy} \cdot \omega_z = 6 \text{ arcs} \cdot 360\ ^\circ/3 \text{ min} = 0.2\ ^\circ/\text{h} \rightarrow 36 \text{ arcs/turn} \tag{19}$$

This value is 20 times larger than the assumed fixed gyro drift of 0.01 °/h. However, if only a few sensor error coefficients in a low-order error model are available for the whole flight, a possible solution is to switch between various simple and predefined models. The criterion for switching depends on the dynamic environment of the SDS at a given moment, i.e. the error model has adaptive qualities. The remaining, non-modelled part of the error budget of the SDS can then be interpreted as system noise.

The angular rate ω_z, for example, can serve as a simple criterion for switching between 2 different low-order sensor error models.

$$\text{rms of } \omega_z < 1\ ^\circ/\text{s} \quad \rightarrow \quad \text{sensor error model 1} \tag{20}$$

$$\text{rms of } \omega_z \geq 1\ ^\circ/\text{s} \quad \rightarrow \quad \text{sensor error model 2} \tag{21}$$

The state vector $\underline{x}$, the error model $\underline{\Phi}$ and the covariance matrix $\underline{P}$ consist of one part which is non-switchable (corresponding to the system errors) and another part which can be switched (corresponding to the sensor errors). If, for example, one proceeds on the basis of 9 SDS errors and models the sensors by means of 3 coefficients, the result is an order of 12 for the Kalman filter. If the vertical components can be calculated separately /11/, a total of only 10 state variables remain.

$$\underline{x} = [\underbrace{\varepsilon_N,\ \varepsilon_E,\ \varepsilon_D,\ \delta v_N,\ \delta v_E,\ \delta\varphi,\ \delta\lambda,}_{\text{non-switchable system errors}}\ \underbrace{d_1,\ d_2,\ d_3}_{\text{switchable sensor errors}}]^T; \tag{22}$$

According to the nature of the dynamic environment, the coefficients d_1 to d_3 describe various sensor errors for the duration of a flight. Fig. 6 shows the principle of the switching procedure. For example, the procedure could start with the state vector $\underline{x}_1$, the error model $\underline{\Phi}_1$ and the covariance matrix $\underline{P}_1$. At this moment, $\underline{x}_1$, $\underline{\Phi}_1$ and $\underline{P}_1$ take into account the fixed gyro drift. 3 minutes later the angular rate ω_z reaches the rms value of 2 °/s, i.e. the criterion for switching has been fulfilled. The elements of $\underline{x}_1$, $\underline{\Phi}_1$ and $\underline{P}_1$ corresponding to the sensor errors at this moment are then stored on the computer buffer and the vector $\underline{x}_2$ and the matrices $\underline{\Phi}_2$, $\underline{P}_2$ are retrieved from the buffer. $\underline{x}_2$, $\underline{\Phi}_2$ and $\underline{P}_2$ might now contain, for example, in addition to the system errors, 3 fixed scale factor errors of the gyros. The system errors, of course, are non-switchable but transferred if a switching procedure occurs. After a further 3 minutes, for example, the rms

value of the angular rate ω_z again decreases to zero and the storage procedure is then repeated in reverse, i.e. $\underline{x}_2$, $\underline{\underline{\Phi}}_2$ and $\underline{\underline{P}}_2$ are stored in a buffer and $\underline{x}_1$, $\underline{\underline{\Phi}}_1$ and $\underline{\underline{P}}_1$ are used again in the Kalman filter algorithms. The sensor errors estimated in the period when $\underline{x}_1$, $\underline{\underline{\Phi}}_1$ and $\underline{\underline{P}}_1$ were valid, are now used again as initial conditions.

The purpose of this switching procedure is constantly to model those sensor errors in the error model that predominate in the error budget. Moreover, since the Kalman filter algorithms adopt the results of the last estimates as initial conditions when certain sensor error coefficients are used again, it is sufficient to have relatively short, though frequently recurring, periods in order to improve the covariance matrices $\underline{\underline{P}}$. The procedure of switching is especially efficient in the case of flight paths that can be divided up into sections where only a few but different sensor errors predominate. In this case a number of predefined low-order sensor error models are sufficient to describe the overall system error behaviour for the whole duration of a flight. In order to establish low-order sensor error models that are to describe the real system error behaviour with sufficient precision, it is necessary first to determine the predominant sensor error coefficients.

3.1 LOW-ORDER SENSOR ERROR MODELS FOR FLIGHT PATH 1

The flight paths of commercial airliners are characterized by their small number of manoeuvres, which include relatively low angular rates and correspondingly low values for the accelerations. This means for the error budget of a SDS that the fixed gyro drift produces the main contribution. The errors proportional to $\underline{\omega}$ are effective only for the duration of the manoeuvres (e.g. changes of course, holdings). Assuming, as above, that only a few manoeuvres occur - i.e. that the duration of the manoeuvres is considerably smaller than the total flying time - then, in principle, on account of the integral Eq.(10) only the sensor errors dependent on ω_z will produce a system error which can no longer be ignored in comparison with the contribution of the fixed gyro drifts. For the given flight path 1, which contains a total of 4 full turns with a duration of 3 min each, a ratio of 12 min/100 min for the angular rate ω_z is obtained. The other components of the angular rates ω_x, ω_y only occur for a short time during the beginning and the end of the turns, and all the sensor errors dependent on ω_x, ω_y can therefore be ignored in the system error budget. The following 6th-order gyro error model is left:

$$\delta\omega_x = -\varepsilon_y^{gy}\omega_z + D_x^o \tag{23}$$

$$\delta\omega_y = \varepsilon_x^{gy}\omega_z + D_y^o \tag{24}$$

$$\delta\omega_z = D_z^{\omega}\omega_z + D_z^o \tag{25}$$

As regards the accelerometer errors for flight path 1 it is sufficient to consider only the bias values, because a commercial airliner usually flies standard turns characterized by negligible accelerations.

$$\delta\underline{a}_b = (B_x^o \quad B_y^o \quad B_z^o)^T \tag{26}$$

Thus for flight path 1 the following sensor error models are compared:

model 1/1: fixed gyro drift
model 1/2: accelerometer bias
model 1/3: fixed scale factor error D_z^{ω} and axes misalignment of the gyros ε_x^{gy}, ε_y^{gy}
model 1/4: switch between the models 1/1 and 1/3
model 1/5: this model contains all 23 sensor errors and is considered as a reference.

Fig. 7 contains the time histories of the angular error ε_0 for the 5 different error models mentioned above. It can be seen how the overall system error is produced by the individual sensor errors: the approximate linear increase of ε_0 is caused by the fixed gyro drift (model 1/1) and the system error contributions of the fixed scale factor error and the axes misalignment (model 1/3) corresponds clearly to the manoeuvres simulated. During the turns the fixed scale factor error leads to a system misalignment that could be calculated as follows:

$$\varepsilon_D = D_z^{\omega} \cdot \omega_z \cdot \Delta T = 3 \cdot 10^{-5} \cdot \frac{360^o}{3\ \text{min}} \cdot 3\ \text{min} = 0.01^o \tag{27}$$

This value can be read off directly from the height of the steps in the time history of the system misalignment caused by model 1/3. During the straight lines of flight path 1 model 1/3 does not produce any additional error contribution. Because the contribution of the accelerometer bias (model 1/2) is rather small model 1/4, which is able to switch between models 1/1 and 1/3, fits the reference error behaviour of model 1/5 best.

3.2 LOW-ORDER SENSOR ERROR MODELS FOR FLIGHT PATH 2

Flight path 2 flown by a F104 combat aircraft contains a large number of extreme manoeuvres. The discussion of the causes of errors by using plausibility methods, as in flight path 1, is no longer possible in this case. For this reason, all the sensor errors were analysed individually with reference to their effect at system level: this was done by calculating the corresponding error propagation Eq.(14,16). Especially striking in this connection was the large error contribution of the anisoertia term

$$\delta\underline{\omega}_b = \frac{A^{gy}}{H^{gy}} \cdot \begin{bmatrix} -\dot{\omega}_y \\ \dot{\omega}_x \\ \dot{\omega}_y \end{bmatrix} \qquad (28)$$

A further error contribution is provided by the sensor errors proportional to the angular rates. As has already been shown even small angular rates lead to large system errors. For this reason the following sensor error models were compared:

model 2/1: fixed gyro drift
model 2/2: scale factor error D_x^{ω}, $D_x^{|\omega|}$
model 2/3: anisoertia term $D^{\Delta\omega}$
model 2/4: switch between model 2/1 ard a model consisting of D_x^{ω}, $D_x^{|\omega|}$ and $D^{\Delta\omega}$
model 2/5: reference model containing 23 sensor error coefficients.

Fig. 8 shows the time histories of the east position errors. The model 2/3 provides a position error that is too large in comparison with the reference values (model 2/5). The fixed gyro drift (model 2/1) and the scale factor errors (model 2/2) produce a position error with a negative sign. The time history of the error behaviour caused by the model 2/4 is nearest the reference. At the end of this real flight of approx. 40 min, the following error budget is obtained:

model	2/1	2/2	2/3	2/4	2/5
east position error after 40 min	-500 m	-500 m	6000 m	5000 m	5000 m

The effect whereby some sensor errors are cancelled out depends on the sign of the sensor error coefficients and the type of manoeuvres flown. If the aircraft executes contrary manoeuvres such as the following bank angle commands $0/30^{o}/0^{o}/-30^{o}/0^{o}$, then some errors will always cancel each other out to some extent. In this case a position error during flight can be partly reduced by selecting appropriate manoeuvres.

3.3 LOW-ORDER SENSOR ERROR MODELS FOR FLIGHT PATH 3

Similar considerations obtain for flight path 3, which is meant to simulate a RPV path, as for flight path 2. High angular rates and accelerations were assumed in order to simulate the extreme dynamic environment of a SDS installed on a RPV. After the calculation of all the individual sensor errors the following 5 sensor error models are left for comparison:

model 3/1: fixed gyro drift
model 3/2: fixed scale factor error D_x^{ω}, D_z^{ω}
model 3/3: anisoertia term $D^{\Delta\omega}$
model 3/4: switch between model 3/1 and a model consisting of D_x^{ω}, D_z^{ω} and $D^{\Delta\omega}$
model 3/5: reference model identical to the models 1/5 and 2/5.

The effect of error contributions on the system level with different signs is especially noticeable in the time histories of the east position errors of flight path 3 shown in Fig. 9. While the contribution of the anisoertia term (model 3/3) increases up to approx. -4 km, the fixed scale factor errors (model 3/2) produce a position error of up to approx. 2 km. The effect of the fixed gyro drift is negligible in the simulated RPV flight path 3 (model 3/1). Error model 3/4, which switches between the error models defined above, describes with sufficient accuracy the position error caused by all the different sensor errors. The rank of the sensor error model 3/4 is only 3.

3.4 THE QUALITIES OF THE SELECTED ADAPTIVE SENSOR ERROR MODELS

For flight path 1, 2 and 3 sensor error models 1/4, 3/4 and 3/4 are selected as being suitable for use for Kalman filtering and real-time applications. These sensor error models exhibit the following qualities:

- The sensor error models contain only 3 state variables, i.e. the rank of the system error model is 12, or only 10 in the case of separate vertical velocity and vertical position errors.
- Switching between these error models in accordance with different flight sections with many or few manoeuvres guarantees adequate adaptive qualities.
- The non-modelled sensor errors do not produce large system errors and can thus be interpreted as system noise modelled in the system noise matrix $\underline{Q}$.

4. THE USE OF SENSOR ERROR MODELS FOR KALMAN FILTERING

In the preceeding Section the attempt was made to determine those sensor errors that produce the largest contributions to the error budget at system level. If only a few sensor errors define the system error, it is possible to model the corresponding sensor error coefficients in a low-order sensor model. However, it may happen that all the sensor errors explained in Section 2 cause large system errors and that none of the sensor error coefficients can be ignored in comparison with any other. In this case a Kalman filter would have a rank up to 33 and this is bound to lead to numerical problems or to

violate real-time computational conditions.

4.1 NUMERICAL PROBLEMS IN HIGH-ORDER ERROR MODELS

Numerical problems, for example round-off errors, often result in negative elements within the trace of the covariance matrix $\underline{\underline{P}}$. These negative elements are not defined at all and have fatal consequences for the filtering. In order to avoid such problems the methods of formulating the filter equations in terms of triangular matrices can be applied. As the comparison between various procedures has shown /12/, these methods generally involve an increase in the number of calculations in comparison with conventional filter equations. The advantage of avoiding numerical problems, e.g. negative elements within the trace of the covariance matrix $\underline{\underline{P}}$, must therefore usually be paid for with an increase in computational operations.

In order to save the number of computational operations, it is possible to take into account the structures of the discrete transition matrix $\underline{\underline{\Phi}}$ and the covariance matrix $\underline{\underline{P}}$. Most of the elements of these matrices are always zero. Using the system error model without any sensor error coefficient $\underline{\underline{\Phi}}_{11}$ and the matrix $\underline{\underline{\Phi}}_{12}$ for the cross-coupling of the sensor errors via the transformation matrix $\underline{\underline{C}}_{nb}$ to the system error level, the covariance propagation equation of the Kalman filter can be stated as follows:

$$\underline{\underline{P}}\ (k+1) = \begin{bmatrix} \underline{\underline{\Phi}}_{11} & \underline{\underline{\Phi}}_{12} \\ \underline{\underline{0}} & \underline{\underline{I}} \end{bmatrix} \cdot \underline{\underline{P}}(k) \cdot \begin{bmatrix} \underline{\underline{\Phi}}_{11} & \underline{\underline{0}} \\ \underline{\underline{\Phi}}_{12}^{T} & \underline{\underline{I}} \end{bmatrix} + \underline{\underline{Q}}(k) \tag{29}$$

With the symmetrical matrices P and Q

$$\underline{\underline{P}} = \begin{bmatrix} \underline{\underline{P}}_{11} & \underline{\underline{P}}_{12} \\ \underline{\underline{P}}_{12}^{T} & \underline{\underline{P}}_{22} \end{bmatrix} \quad ; \quad \underline{\underline{Q}} = \begin{bmatrix} \underline{\underline{Q}}_{11} & \underline{\underline{Q}}_{12} \\ \underline{\underline{Q}}_{12}^{T} & \underline{\underline{Q}}_{22} \end{bmatrix} \tag{30}$$

is follows by multiplication

$$\underline{\underline{P}}_{11}(k+1) = [(\underline{\underline{\Phi}}_{11}\,\underline{\underline{P}}_{11} + \underline{\underline{\Phi}}_{11}\,\underline{\underline{P}}_{12}^{T})\,\underline{\underline{\Phi}}_{11}^{T}](k) + \underline{\underline{P}}_{12}(k+1) \cdot \underline{\underline{\Phi}}_{12}^{T}(k) \tag{31}$$

$$\underline{\underline{P}}_{12}(k+1) = [(\underline{\underline{\Phi}}_{11}\,\underline{\underline{P}}_{12} + \underline{\underline{\Phi}}_{12}\,\underline{\underline{P}}_{22})\,\underline{\underline{\Phi}}_{12}^{T}](k) \tag{32}$$

$$\underline{\underline{P}}_{22}(k+1) = \underline{\underline{P}}_{22}(k) \tag{33}$$

Using $\underline{x}_1$ for the state variables without sensor error coefficients and $\underline{x}_2$ for the sensor error the propagation equation for the state vector

$$\begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix}(k+1) = \begin{bmatrix} \begin{bmatrix} \underline{\underline{\Phi}}_{11} & \underline{\underline{\Phi}}_{12} \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{bmatrix}(k) \tag{34}$$

can be stated in the simplified form

$$\underline{x}_1\ (k+1) = \underline{\underline{\Phi}}_{11}(k)\ \underline{x}_1(k) + \underline{\underline{\Phi}}_{12}(k)\ \underline{x}_2(k) \tag{35}$$

$$\underline{x}_2\ (k+1) = \underline{x}_2(k) \tag{36}$$

The saving in the number of multiplications can be calculated by using n_{max} for the maximum number of state variables, which is the sum of n_1 system errors and n_2 sensor errors. In the case of the usual matrix-vector-operations a maximum number m_{max} of multiplications is necessary:

$$m_{max} = \underbrace{2n_{max}^{3}}_{\text{propagation of } P} + \underbrace{n_{max}^{2}}_{\text{propagation of } \underline{x}} \tag{37}$$

In accordance with the simplified Eq.(31, 32, 35) it follows for the minimum number m_{min} of multiplications with P_{22} as a diagonal matrix:

$$m_{min} = \underbrace{2n_1^3 + 2n_1^2n_2 + n_1^2n_2}_{\text{propagation of } P_{11}} + \underbrace{2n_1^2n_2 + n_1n_2 + n_1^2n_2}_{\text{propagation of } P_{12}} + \underbrace{n_1^2 + n_1n_2}_{\text{propagation of } \underline{x}} \tag{38}$$

or, by summing the terms, the factor n of the savings in the number of multiplications can be given thus:

$$n = \frac{n_{max}}{n_{min}} - 1 = \frac{2(n_1+n_2)^3 + (n_1+n_2)^2}{2n_1^3 + 6n_1^2n_2 + 2n_1n_2 + n_1^2} - 1 \tag{39}$$

Fig. 10 shows the function $n=f(n_2)$ on the basis of a fixed number of system errors: 3 angular misalignments, 3 velocity errors, 3 position errors and the altimeter bias.

Although there is a saving in the number of multiplications by taking into account the structure of the matrices $\underline{\Phi}$ and $\underline{P}$, the calculations still need too much time for a high-order model. If one tries to model 10 error coefficients, this would lead to a number m of multiplications of

$$m = 2n^3/n(n_2) = 2n^3/2 = 8000 \tag{40}$$

A digital computer with a floating point processor and an assumed time comsumption of approx. 60 µs for one multiplication will thus need approx. 0.5 s merely for the covariance and error propagation equations. It can be seen from this very rough estimate that it is essential to use low-order models in real-time applications. If the order of the Kalman filter is limited to a maximum number of 13 to 15 state variables, numerical problems do not arise and real-time applications remain possible.

4.2 THE INTERPRETATION OF THE NON-MODELLED SENSOR ERROR COEFFICIENTS AS A SYSTEM NOISE

There is another basic possiblitiy of describing the error behaviour of SDS for flight paths in which a large number of sensor errors are effective at system level and the numerical problems of using high-order error models cannot be solved satisfactorily. This is the interpretation of the non-modelled sensor error coefficients as system noise in terms of the matrix $\underline{Q}$. Roughly speaking, the covariance propagation equation supplies over-optimistic values in the case of non-modelled sensor error coefficients. This effect can be compensated for by a corresponding increase in the elements of the system noise matrix $\underline{Q}$. However, this procedure leads to problems if essential error contributions dependent on the sign of the sensor signals are interpreted as unbiased random signals. This applied, for example, in the case of a fixed scale factor error of the gyro: if the aircraft performs a left-hand turn and then a right-hand turn, the system misalignments caused by these sensor error coefficients cancel each other out to some extent. The interpretation of these sensor error coefficients as corresponding system noise would make the covariance matrix worse - irrespective of the particular direction of the turns. Therefore, this method leads to covariance matrices that assume over-pessimistic values in the case of contrary manoeuvres. Thus, of the sensor error coefficients defined in Section 2, those that are basically suitable for description by means of noise matrices are the fixed gyro drifts, the accelerometer bias and the quadratic non-linear scale factor error of the gyros.

Of course, the sensor error coefficients which are taken into account in the system noise matrix $\underline{Q}$ do not occur in the form of state variables and therefore estimation by means of Kalman filtering is no longer possible. For this reason non-modelled sensor error coefficients should only be interpreted as system noise if their contribution to the overall system error behaviour is relatively small. When the corresponding elements of the noise matrix $\underline{Q}$ are being determined, the gyro errors are to be written in angular terms and the accelerometer errors in velocity terms.

If the system matrix $\underline{\Phi}$ is approximated by means of the identity matrix $\underline{I}$ for a sufficiently short interval T_a, then the following applies:

$$\underline{\varepsilon}_n(k+1) = \underline{\varepsilon}_n(k) + T_a \cdot \underline{C}_{nb}(k) \cdot \delta\underline{\omega}_b(k) \tag{41}$$

$$\delta\underline{v}_n(k+1) = \delta\underline{v}_n(k) + T_a \cdot \underline{C}_{nb}(k) \cdot \delta\underline{a}_b(k) \tag{42}$$

When $\underline{C}_{nb}\underline{C}_{nb}^T = \underline{I}$, it follows for the variances σ that

$$\underline{\sigma}_\varepsilon^2(k+1) = \underline{\sigma}_\varepsilon^2(k) + \underbrace{T_a^2 \cdot E[\delta\underline{\omega}_b(k) \cdot \delta\underline{\omega}_b^T(k)]}_{\underline{q}_{\varepsilon\varepsilon}^{*2}} \tag{43}$$

With the integration of the terms $q_{\varepsilon\varepsilon}^{*2}$ up to the time $k \cdot T_a$, variances for the misalignments ε are obtained in accordance with

$$\underline{\sigma}_\varepsilon^2(k \cdot T_a) = k\underline{q}_{\varepsilon\varepsilon}^{*2} \tag{44}$$

In order to obtain a standard deviation σ that corresponds to the basic relation misalignment = drift multiplied by time at the end of the observed interval kT_a, the following equation has to be fulfilled:

$$\sqrt{E[\delta\underline{\omega}_b(k) \cdot \delta\underline{\omega}_b^T(k)]} \cdot (kT_a) = \sqrt{k\underline{q}_{\varepsilon\varepsilon}^2} \tag{45}$$

By rearrangement one obtains a term in the form of a spectral density for determining the elements $q_{\varepsilon\varepsilon}$ of the system noise matrix $\underline{Q}$

$$\underline{q}_{\varepsilon\varepsilon}^2 = k \cdot T_a \cdot E[\delta\underline{\omega}_b \cdot \delta\underline{\omega}_b^T] \quad [\text{rad}^2/\text{s}] \tag{46}$$

or, by a corresponding calculation for the accelerometer errors:

$$\underline{q}^2_{vv} = kT_a \cdot E[\delta\underline{a}_b \cdot \delta\underline{a}_b^T] \quad [(m/s)^2/s] \tag{47}$$

For the fixed gyro drift value of 0.01 °/h or the accelerometer bias of 10^{-4} g the following numerical values are obtained:

$$q^2_{\varepsilon\varepsilon} = 2.37 \cdot 10^{-15} \cdot kT_a \quad [1/s] \tag{48}$$

$$q^2_{vv} = 10^{-6} \cdot kT_a \quad [(m/s)^2/s] \tag{49}$$

For the remaining sensor error coefficients the calculation of the expectation values is necessary. The sensor signals consist generally of a low-frequency part that describes the flight path, and superimposed high-frequency vibrations. Statistically speaking, they represent non-stationary time series. The calculation of the expectation values consequently poses considerable problems: for example, the variances that have been determined depend on the selection of the observed flight path section. The equations in Section 4 thus only permit a rough estimate of the order of magnitude of the elements of the system noise matrix that correspond to the non-modelled sensor error coefficients.

4.3 THE ADJUSTMENT OF THE SYSTEM NOISE MATRIX $\underline{Q}$ FOR THE 3 DIFFERENT FLIGHT PATHS SELECTED

The adjustment of the system noise matrix $\underline{Q}$ for the fixed gyro drift not included in the error model was carried out in accordance with Eq.(46). Despite the fact that they were non-stationary the expectation values of the sensor signals were calculated so that sensor errors depending on the dynamic environment of the SDS can also be included in the system noise matrix. The sensor signals and the angular accelerations varied between the listed values. Mean values were assumed for the expectation values in order to make a rough estimate for the elements of the system noise matrix $\underline{Q}$ possible. The results are listed in Tab. 2:

flight path 1	$0.4\ °/s < \sigma_{\omega_b} < 0.6\ °/s$ $\sigma_{\dot{\omega}_b} = 0$	$\bar{\sigma}_{\omega_b} = 0.5\ °/s$ $\bar{\sigma}_{\dot{\omega}_b} = 0.$
flight path 2	$0.9\ °/s < \sigma_{\omega_b} < 4.84\ °/s$ $2.3\ °/s^2 < \sigma_{\dot{\omega}_b} < 8.2\ °/s^2$	$\bar{\sigma}_{\omega_b} = 3\ °/s$ $\bar{\sigma}_{\dot{\omega}_b} = 5.3\ °/s^2$
flight path 3	$0.5\ °/s < \sigma_{\omega_b} < 5.2\ °/s$ $0.2\ °/s^2 < \sigma_{\dot{\omega}_b} < 22\ °/s^2$	$\bar{\sigma}_{\omega_b} = 2.8\ °/s$ $\bar{\sigma}_{\dot{\omega}_b} = 11\ °/s^2$

Tab. 2: Standard deviation for the sensor signals

For the fixed scale factor error $D^{\omega}=1\cdot10^{-5}$ and the anisoertia term $D^{\Delta\omega}=4\cdot10^{-4}$s the corresponding elements of the system noise matrix $\underline{Q}$ are listed in Tab. 3:

	$q^2_{\varepsilon\varepsilon}/D^{\omega}[1/s]^2$	$q^2_{\varepsilon\varepsilon}/D^{\Delta\omega}[1/s]^2$
flight path 1	$5 \cdot 10^{-11}$	-
flight path 2	$6 \cdot 10^{-9}$	$3.3 \cdot 10^{-6}$
flight path 3	$5.2 \cdot 10^{-9}$	$1.4 \cdot 10^{-5}$

Tab. 3: Elements of the matrix $\underline{Q}$ corresponding to the sensor error coefficients D^{ω}, $D^{\Delta\omega}$

4.3.1 THE CALCULATION OF ELEMENTS OF THE SYSTEM NOISE MATRIX $\underline{Q}$ FOR FLIGHT PATH 1

5 different error models will again be considered. The non-modelled sensor error coefficients in each case are interpreted as system noise.

Error model 1/6
Δ: no sensor error coefficients taken into account
$\underline{Q}$ = f (all sensor error coefficients)
Thus, in this error model 1/6 all the main sensor error coefficients are interpreted as system noise. It follows from Eq.(46, 47) and Tab. 2 that

$$q_{\varepsilon\varepsilon}^2 = 10^{-11}/s \text{ and } q_{vv}^2 = 2.5 \cdot 10^{-7}(\frac{m}{s})^2/s.$$

Error model 1/7
Here, in contrast to error model 1/6, the fixed gyro drift was modelled. This results in

$$q_{\varepsilon\varepsilon}^2 = 10^{-12}/s \text{ and } q_{vv}^2 = 2.5 \cdot 10^{-7}(\frac{m}{s})^2.$$

Error model 1/8
Here, switching takes place between the fixed gyro drift and the gyro errors D_z^{ω}, c_x^{gy}, ε_y^{gy}. For this error model 1/8 the elements $q_{\varepsilon\varepsilon}$ were taken as zero. For $q_{vv}=2.5\cdot10^{-7}(\frac{m}{s})^2/s$ holds.

Error model 1/9
The error model 1/9 is basically similar to 1/8. However, here the non-modelled sensor error coefficients are considered as system noise. It thus follows for the elements of the noise matrix $\underline{Q}$ that

$$q_{\varepsilon\varepsilon} = 7 \cdot 10^{-13}/s \text{ and } q_{vv} = 2.5 \cdot 10^{-7}(\frac{m}{s})^2/s.$$

Error model 1/10
This error model serves as a reference, i.e. all the sensor error coefficients are modelled.

Fig. 11 contains the time histories of the 1σ-values for the component of the east position error. According to this, the models 1/7 to 1/10 exhibit a largely identical result, i.e. the axes misalignments and the fixed scale factor error of the gyros as well as the accelerometer bias can be interpreted with sufficient accuracy as system noise for this particular flight path 1. However, the situation is different in the case of model 1/6, which does not model any sensor error coefficients. Although on the whole sufficient agreement with the reference model 1/10 can be obtained, deviations up to approx. ± 500 m do occur in the middle of flight path 1.

4.3.2 THE CALCULATION OF ELEMENTS OF THE SYSTEM NOISE MATRIX $\underline{Q}$ FOR FLIGHT PATH 2

Once again 5 different error models are considered.

Error model 2/6
All the sensor error coefficients were interpreted as system noise. This leads to

$$q_{\varepsilon\varepsilon}^2 = 2.4 \cdot 10^{-10}/s \text{ and } q_{vv}^2 = 10^{-7}(\frac{m}{s})^2/s.$$

Error model 2/7
The fixed gyro drift only is modelled. The remaining sensor error coefficients result in

$$q_{\varepsilon\varepsilon}^2 = 2.35 \cdot 10^{-10}/s \text{ and } q_{vv}^2 = 10^{-7}(\frac{m}{s})^2/s.$$

Error model 2/8
In error model 2/8 switching takes place between the fixed gyro drifts $\underline{D}^{\omega}$ and the gyro errors D_z^{ω}, D_x^{ω}, $D^{\Delta\omega}$. The elements $q_{\varepsilon\varepsilon}^2$ are set to zero and

$$q_{vv}^2 = 10^{-7}(\frac{m}{s})^2/s.$$

Error model 2/9
This is basically the same as error model 2/8, however, all the non-modelled error contributions are included in the system noise matrix. This leads to

$$q_{\varepsilon\varepsilon}^2 = 10^{-12} \text{ 1/s and } q_{vv}^2 = 10^{-7}(\frac{m}{s})^2/s.$$

Error model 2/10
This is the reference model again including all sensor error coefficients as state variables.

Fig. 12, which contains the time histories for the 1σ-values of the angular misalignment ε_D, shows clearly the difficulties involved in the interpretation of dominant sensor error coefficients as an unbiased noise. The anisoertia term not contained in error models 2/6 and 2/7 leads to results which deviate considerably from the reference values. The elements of the system noise matrix $\underline{Q}$ can be adjusted according to whether agreement with the reference values is to be achieved at the beginning, in the middle or at the end of flight path 2. The results for the error models 2/8 and 2/9 represent the actual error limits sufficiently well because no dominant senor error coefficient is interpretated as system noise.

4.3.3 THE CALCULATION OF ELEMENTS OF THE SYSTEM NOISE MATRIX $\underline{Q}$ FOR FLIGHT PATH 3

Again a total of 5 different error models are considered.

Error model 3/6
All the sensor error coefficients are taken into account in the system noise. This leads to

$$q_{\varepsilon\varepsilon}^2 = 1.2 \cdot 10^{-10}/s \text{ and } q_{vv}^2 = 2.5 \cdot 10^{-7}(\frac{m}{s})^2/s.$$

Error model 3/7
Only the fixed gyro drift is modelled. Therefore

$$q^2_{\varepsilon\varepsilon} = 10^{-10}/s \text{ and } q^2_{vv} = 2.5 \cdot 10^{-7}(\frac{m}{s})^2/s.$$

Error model 3/8
Now sensor error models are switched between D^o_x, D^o_y, D^o_z and D^ω_x, D^ω_z, $D^{\Delta\omega}$. The elements $q^2_{\varepsilon\varepsilon}$ are set to zero and

$$q^2_{vv} = 2.5 \cdot 10^{-7}(\frac{m}{s})^2/s.$$

Error model 3/9
This error model corresponds to model 3/8, however, all the non-modelled sensor error coefficients are included in the system noise matrix. This results in

$$q^2_{\varepsilon\varepsilon} = 10^{-12}/s \text{ and } q^2_{vv} = 2.5 \cdot 10^{-7}(\frac{m}{s})^2/s.$$

Error model 3/10
This is the reference model again. It is identical to models 1/10 and 2/10.

Fig. 13 contains the time histories of the 1σ-values for the north position error. Error models 3/6 and 3/7, which do not include the dominant sensor error coefficient $D^{\Delta\omega}$, give only an inaccurate representation of the reference values. The modelling of $D^{\Delta\omega}$ alone leads to deviations of approx. ± 800 m. The 1σ-values of the reference model 3/10 can be approximated with a deviation of approx. ± 400 m by using error 3/9, which takes account of additional sensor error coefficients in the system noise matrix $\underline{Q}$.

4.4 THE ADAPTIVE KALMAN FILTERING ALGORITH

It was shown in Section 4.3 how some of the sensor errors can be modelled or taken into account in the system noise matrix $\underline{Q}$. However, there remain a number of sensor error coefficients that can be effective at system level when certain manoeuvres are performed. These sensor error coefficients can lead to a divergence of the Kalman filter algorithms. The cause of this divergence lies in an over-optimistic covariance matrix $\underline{P}$, which reduces the elements of the gain matrix $\underline{K}$ and thus - roughly speaking - ignores the measurements being received. In order to avoid this divergence, the system noise matrix $\underline{Q}$ can frequently be increased to a suitable extent, though this lowers the level of system accuracy drastically. In the case of SDS, the calculation of constant system noise matrices $\underline{Q}$ must allow for the worst dynamic environment of the SDS. For example, a large number of sensor error coefficients effective during a short manoeuvre are able to initiate the divergence of the Kalman filter algorithms.

It is far easier to solve this problem by using variable system noise matrices $\underline{Q}$. This leads to matrices $\underline{Q}$ dependent on the actual manoeuvres flown. The calculation of the elements of such a variable system noise matrix $\underline{Q}$ can be performed by means of the so-called adaptive Kalman filtering taking into account the statistics of the filter residuals. The relationship between the expectation values of the measurement $\underline{z}$, the covariance matrix $\underline{R}$ for the measurement errors of, for example, a radar unit, the measurement matrix $\underline{H}$ and the unknown noise matrix $\underline{Q}$ can be formulated as follows:

$$E[(\underline{z}-\underline{H}\underline{x}')\cdot(\underline{z}-\underline{H}\underline{x}')^T](k) = \underline{R}(k) + \underline{H}[\underline{P}'(k) + \underline{Q}(k)]\underline{H}^T. \tag{50}$$

The estimates $\underline{\hat{x}}$ and the covariance matrix $\underline{P}$ correspond to the predicted expressions $\underline{x}'$, $\underline{P}'$ via

$$\underline{x}'(k) = \underline{\phi}(k)\ \underline{\hat{x}}(k-1) \tag{51}$$

$$\underline{P}'(k) = \underline{\phi}(k)\ \underline{P}(k)\ \underline{\phi}^T(k) + \underline{Q}(k) \tag{52}$$

The filter residuals $(\underline{z}-\underline{H}\underline{x}')$ in many adaptive filtering procedures are used as the basis of a statistical analysis /13, 14, 15, 16, 17, 18/. In visual terms Eq.(50) means that the differences between measured state variables and predicted state variables are described statistically by the corresponding covariance matrices $\underline{R}$, $\underline{P}$ and $\underline{Q}$. The result is a variable system noise matrix $\underline{Q}$ or a corresponding effect on the gain matrix $\underline{K}$.

From the large number of possible adaptive filtering algorithms the one based on the work of JAZWINSKI /13, 14/ was selected because it requires a particularly small number of calculations and is thus especially suitable for real-time applications. In this algorithm the calculation of the expectation values of the filter residuals takes place on the basis of n measurements backward in time history. If this expectation values do not correspond to the sum of the covariance matrices $\underline{P}$ and $\underline{R}$, this can be compensated for via the noise matrix $\underline{Q}$.

According to /14/, for a matrix $\underline{B}$

$$\underline{B} = E\ [(\underline{z}-\underline{H}\underline{x}')\ (\underline{z}-\underline{H}\underline{x}')^T] - \underline{R} - \underline{H}\underline{P}'\underline{H}^T \tag{53}$$

the following criterion applies:

$$B_{ll} > 0 \rightarrow \underline{Q} = \underline{B} \text{ or } B_{ll} < 0 \rightarrow \underline{Q} = 0 \tag{54}$$

By use of this relatively simple algorithm the system noise matrix $\underline{Q}$ continuously adapts itself to the predicted statistics of the Kalman filter. However, only at the level of the measurements can the elements of the matrix $\underline{Q}$ be calculated by this simple algorithm. An extension of the adaptive Kalman filtering in order to calculate all the elements of the matrix $\underline{Q}$ increases the computational burden drastically.

5. DISCUSSION OF RESULTS

5.1 THE RESULTS OBTAINED FROM AN ADAPTIVE FILTER WITH SWITCHABLE SENSOR ERROR MODELS

The discussion of the results based on low-order error models within an adaptive filter assumes external measurements and a knowledge of the estimation errors. Both conditions can be fulfilled in the case of simulated data. The known reference flight paths 1 and 3 are interpreted as external measurements and the system and sensor errors whose values are known from the simulation are suitable for checking the estimations of the adaptive filtering.

5.1.1 THE RESULTS FOR FLIGHT PATH 1

Fig. 14 contains for flight path 1 the time histories of the parameter L defined as follows

$$\text{rms of } |\underline{\omega}| < 1\ ^{\circ}/s \rightarrow L = 1 \text{ or rms of } |\underline{\omega}| > 1\ ^{\circ}/s \rightarrow L = 2 \qquad (55)$$

In the event of manoeuvres the sensor error models were exchanged in accordance with the criterion for switching. If the flight path corresponds to a straight line, the error behaviour is described by the error model which only contains the fixed gyro drifts.

Fig. 15 is based on error model 1/9 and represents the time histories of the estimation errors and the corresponding 1σ-bands for the fixed gyro drift D_y^0. The accuracy can can be improved to about 0.005 °/h. However, these extreme levels of accuracy require a period of observation of approx. 60 min. The estimation errors reach an accuracy level of 0.002 °/h. If one compares the time histories of the estimation error and the corresponding 1σ-band, it becomes clear that the self-diagnosis of the filter produces values that are over-pessimistic by about the factor 2. The reason for this effect lies in the interpretation of the non-modelled sensor error coefficients as system noise.

Fig. 16 indicates especially well the existence of sensor errors that are effective at system level but non-modelled. It contains the elements of the system noise matrix calculated by the adaptive filter. An increase in the elements to values up to 1.3 m/s can be seen in the section where manoeuvres were performed. This is a result of large deviations between the reference and the modelled error behaviour of a SDS in the case of manoeuvres. The take-off and the first turn are not visible in Fig. 16 because the covariance matrix $\underline{P}$ is still decreasing from its initial value and therefore the increase in the filter residuals during these manoeuvres does not affect the calculation of the adaptive system noise matrix.

5.1.2 THE RESULTS FOR FLIGHT PATH 3

Fig. 17 shows how the frequent and extreme manoeuvres of flight path 3 cause a large number of switching procedures between the sensor error models 2/9.

Fig. 18 contains the time histories of the estimation error and the 1σ-band for the anisoertia term of the gyros. The assumed accuracy at the beginning of $\pm\ 2 \cdot 10^{-3}$ s can be improved to approx. $\pm\ 2 \cdot 10^{-4}$ s. The procedure of switching can clearly be recognized in the shape of the 1σ-band since of course it is only possible to obtain an estimate, and thus an improvement of the covariance matrx $\underline{P}$, if the error model L=2 is used for filtering.

Fig. 19 shows for flight path 3 that it is only at the end of the time histories of the system noise matrix $\underline{Q}$ that the filter residuals exceed the limits given by the covariance matrices $\underline{R}$ and $\underline{P}$.

5.2 THE RESULTS OBTAINED FROM AN ADAPTIVE KALMAN FILTER WITHOUT THE MODELLING OF SENSOR ERRORS

The aim of switching between various error models was to give a sufficiently accurate mathematical description of the error behaviour of a SDS, although low-order sensor error models were used. The success of these techniques depends on the following conditions, however:

- It must be possible to divide up a flight path into sections of different structure on the basis of suitable criteria and
- only a few sensor errors are effective at system level in each of these sections.

The validity of these conditions was examined in Section 3 on the basis of covariance and error analysis for flight paths 1, 2 and 3. In the case of arbitrary flight paths which are subject only to the limits imposed by the aircraft's specifications it may happen that the conditions referred to above can no longer be adhered to with sufficient accuracy. A possible technique in this case is to take account of all the sensor errors in the system noise matrix $\underline{Q}$.

5.2.1 THE RESULTS FOR FLIGHT PATH 1

Fig. 20 contains the time histories of the estimation error and the corresponding 1σ-band for the angular misalignment ε_D with respect to flight path 1. The results are based on the switchable sensor error model 1/9, which has already been discussed. The estimation error and the 1σ-values show a sufficient correspondence.

Fig. 21 contains the estimation errors and 1σ-band for the angular misalignment ε_D once more. However, all the sensor error coefficients are included in the system noise matrix $\underline{Q}$. It can be seen that the filtering leads to far too optimistic values for the 1σ-band, and Eq.(18) is violated.

5.2.2 THE RESULTS FOR FLIGHT PATH 3

The problems that have just been discussed are shown for flight path 3 in Fig. 22. Whereas the various estimates for the angular misalignment ε_D might still be acceptable for flight path 1, the situation with flight path 3, with its frequent and considerable manoeuvres, is different: here extreme deviations from the reference error behaviour arise if the sensor error coefficients are only taken into account in the system noise.

In Fig. 23 the corresponding results are shown for sensor error model 3/9 which can be switched between the dominant error sources. The comparison between the results shown in Fig. 22 and Fig. 23 demonstrates clearly the advantage of the sensor error model 3/9 for flight path 3.

5.3 THE RESULTS IN THE CASE OF INTERRUPTED EXTERNAL MEASUREMENTS

The results discussed so far are based on external measurements that were available all the time. However, if a radar unit or a MLS station fails or the aircraft is too far away from such a station, for exemple, it becomes especially important to have a modelling of the real SDS error behaviour which is as accurate as possible, since it is necessary to predict the errors according to the error equations all the while the external measurements are interrupted.

5.3.1 THE RESULTS FOR FLIGHT PATH 1

Figs. 24 and 25 show the time histories of the estimation errors and the corresponding 1σ-band for the angular misalignment ε_D in the event of external measurements being interrupted after 40 min; two error models are compared. Fig. 24 is based on the switchable error model 1/9, which has already been discussed. In Fig. 25 the corresponding results are given for an error model which contains no sensor error coefficients. Here, the faulty self-diagnosis of the filter can be seen. If sensor error model 1/9 is used, a very small estimation error is obtained.

The errors at the angular level lead to position errors according to the error model used. Figs. 26 and 27 represent the results for these components. The following results can be achieved:

δx_N = -300 m ± 600 m: error model 1/9

δx_N = -2000 m ± 500 m: error model 1/6

Thus, the use of error models without sensor errors leads to far lower levels of accuracy even in the case of flight path 1 which contains fewer manoeuvres.

5.3.2 THE RESULTS FOR FLIGHT PATH 3

Fig. 28 shows the estimation error and the 1σ-band for the angular misalignment ε_D for flight path 3. After 20 min the external measurements are lost. Based on the error model 3/9 an accuracy of about 0.1 ° is reached at the end of the flight path 3, the estimation error is less than 0.01 °.

Fig. 29 is based on a sensor error model which interprets all the sensor errors as system noise. Although the self-diagnosis of the filter gives a value of approx. 0.02°, the actual estimation error is approx. 0.2 °, i.e. a serious violation of Eq.(18) occurs.

Figs. 30 and 31 compare the results of the two error models for the position error components. Whereas the switchable error model 3/9 procudes 1σ-values of approx.±1300 m after the loss of the external measurements (Fig. 30), the error model without sensor errors gives 1σ-values of approx. ±1500 m, although the estimation errors are as much as -5000 m (Fig. 31).

6. SUMMARY AND CONCLUSIONS

The basic problems in the design of error models for aided SDS is the need to provide, in a low-order error model, a sufficiently realistic mathematical description of a large number of sensor error coefficients which, depending on the manoeuvres flown, can be effective at system error level. Three different flight paths were used to discuss the typical sensor and system error behaviour of a SDS and to develop corresponding error models. If the flight path can be divided up into sections with different structures where

only a few sensor errors are effective at system level, a sufficiently good description of the real system error behaviour by switching between various low-order sensor error models.

In the case of sensor errors that cannot be modelled although they are effective at system error level, the system noise matrix $\underline{Q}$ was correspondingly enlarged. This was done on the one hand by giving rough estimates of the sensor error effects not included in the error model, and on the other hand by using adaptive Kalman filtering, which makes it possible to calculate the elements of the system noise matrix $\underline{Q}$ on the basis of a statistical analysis of the filter residuals.

Finally the results of the calculations obtained from simulated flight paths were discussed for various error models with or without the modelling of sensor error coefficients, and also when external measurements were interrupted.

The results in the case of the loss of external measurements are of particular interest since here increased demands are made on an error model for a SDS.

If the consideration of real-time conditions means it is only possible to model a few sensor errors, then it is necessary to find a suitable method for including in the system noise matrix those sensor and algorithm errors which are not modelled but are effective at system error level. The calculation of fixed matrices $\underline{Q}$ proved to be inadequate because, with the exception of constant bias, all the sensor errors have to be regarded as being dependent on the particular dynamic environment of the SDS.

If low-order error models are used for flight paths with extreme manoeuvres, in which a large number of sensor errors can be effective at system level, it is necessary to adapt continuously the system noise matrix to the predected statistics of the Kalman filter. The problem of adaptive Kalman filtering lies in the need to develop efficient algorithms for the statistical analysis of the filter residuals. The analysis of the filter residuals should provide all the elements of the system noise matrix $\underline{Q}$.

7. REFERENCES

/1/ Stieler B.
Einführung in die Strapdown Systeme
DGON I 1977

/2/ Wetzig V.
Simulationen zum Fehlerverhalten eines Strapdown-Navigationssystems auf einer RPV-Flugbahn ohne überlagerte Vibrationen
DFVLR Braunschweig

/3/ Britting K.
Inertial Navigation System Analysis
ISBN 0-471-10485-X, John Wiley & Sons, 1971

/4/ Stieler B.
Die Navigationsgleichungen und das Fehlerverhalten von Trägheitsnavigationssystemen
DFVLR Braunschweig, CCG 1980

/5/ Winter H.
Stützung des Vertikalkanals einer Trägheitsplattform mit einem barometrischen Höhenmesser
DGON Symposium Heidelberg 1974

/6/ Gelb A.
Applied Optimal Estimation
MIT PRESS, 1974

/7/ Stieler B., Winter H.
Advanced Instrumentation and Data Evaluation Techniques for Flight Tests
XI ICAS CONGRESS, Lisboa 1978

/8/ Stieler B., Lechner W.
Calibration of an INS based on flight data
AGARD Conference Proceedings No. 220, 1976

/9/ Kubbat W.J.
Application of Strapdown Inertial Navigation to high Performance Pligher Aircraft
AGARD LECTURE SERIES No. 95, 1978

/10/ Lohl, N.
Vermessung und Modellierung der Flugzeugbewegungen sowie deren Auswirkung auf die Navigationsgenauigkeit eines Strapdown-Systems.
DFVLR Braunschweig

/11/ Stieler B., Winter H.
Gyroscopic Instruments and their Application to Flight Testing
AGARDograph, April 1981

/12/ Neal A.C.
Fast Triangular Formulation of the Square Root Filter
AIAA Journal, Vol. 11, No. 9, Sept. 1973

/13/ Jazwinski A.H.
Stochastic Processes and Filtering Theory
Mathematics in Science and Engineering, Vol. 64
ACADEMIC PRESS, 1970

/14/ Jazwinski A.H.
Adaptive Filtering
Automatica, Vol. 5, 1969

/15/ Lee T.
A Direct Approach to Identify the Noise Covariances of the Kalman Filtering
IEEE Transactions on Automatic Control, Aug. 1980

/16/ Godbole S.
Kalman Filtering with No A Priori Information About Noise
IEEE Transactions on Automatic Control, Oct. 1972

/17/ Mehra R.K.
Approaches to Adaptive Filtering
IEEE Transactions on Automatic Control, Oct. 1972

/18/ Chang C.B., Whiting R.H., Athans M.
On the State and Parameter Estimation for Manoeuvering Reentry Vehicles
IEEE Transactions on Automatic Control, Febr. 1977

/19/ Stambaugh J.S.
Propagation and System Accuracy Impact of Major Sensor Errors on a Strapdown Aircraft Navigator
IEEE Aerospace and Electronics, Nov. 1973

/20/ Joos D.K., Krogmann U.K.
Estimation of Strapdown Senaor Parameter for Inertial System Error Compensation
31st Symposium of the AGRAD, London, Oct. 1980

8. FIGURES

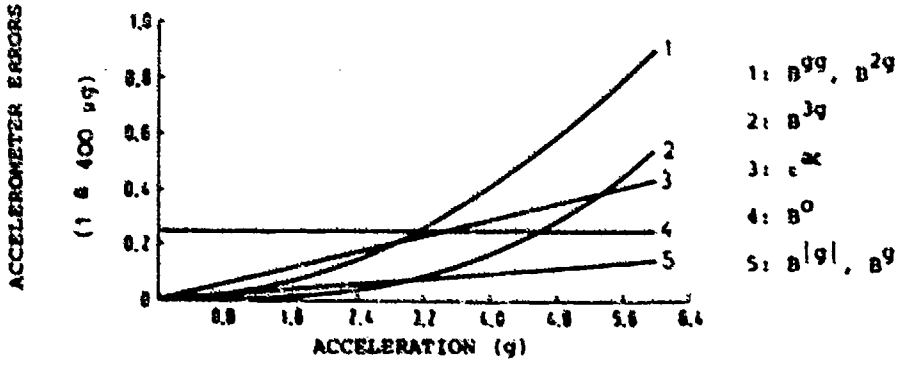


Fig. 1: Errors of the accelerometers

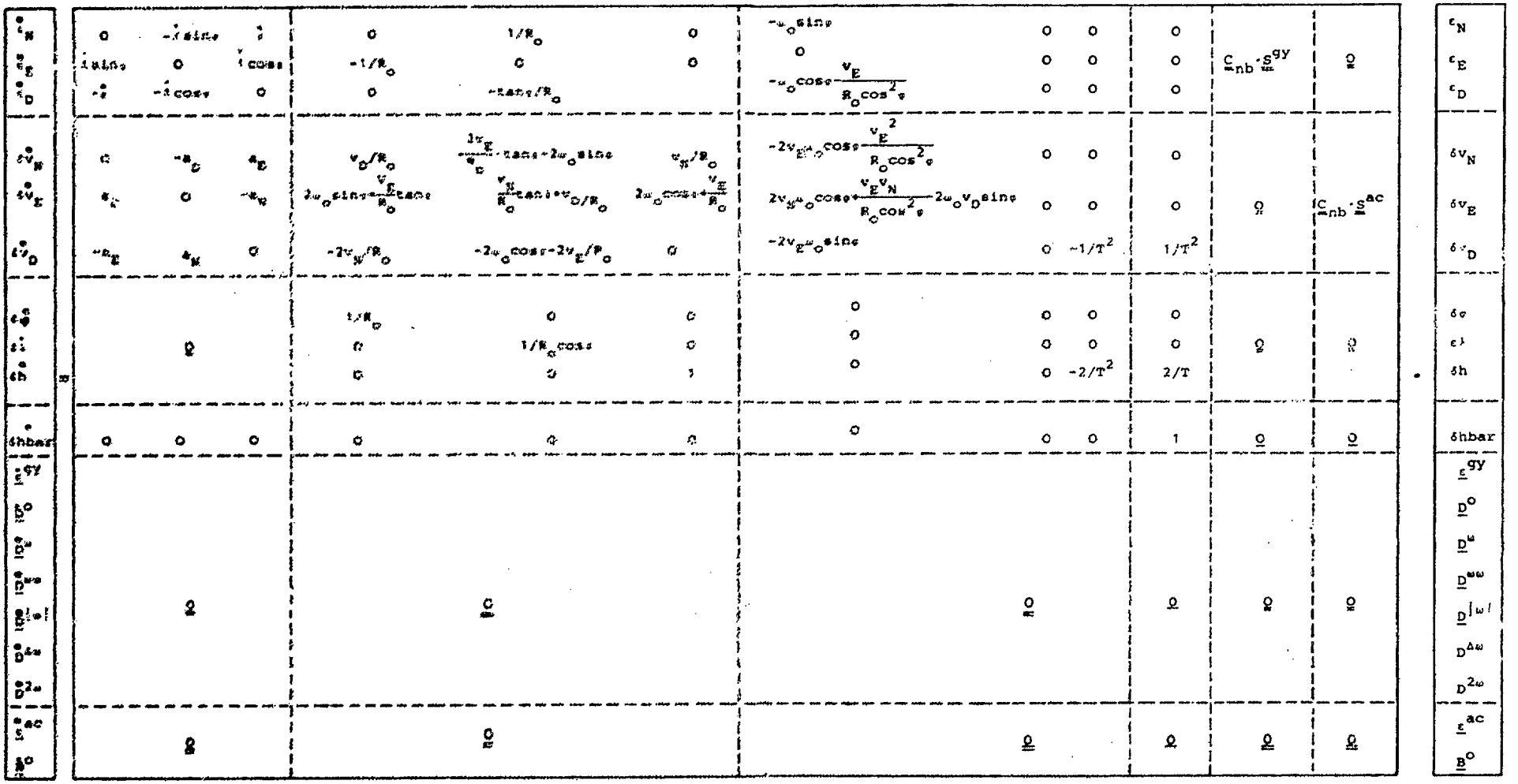


Fig. 2: Reference error model for a strapdown navigation system

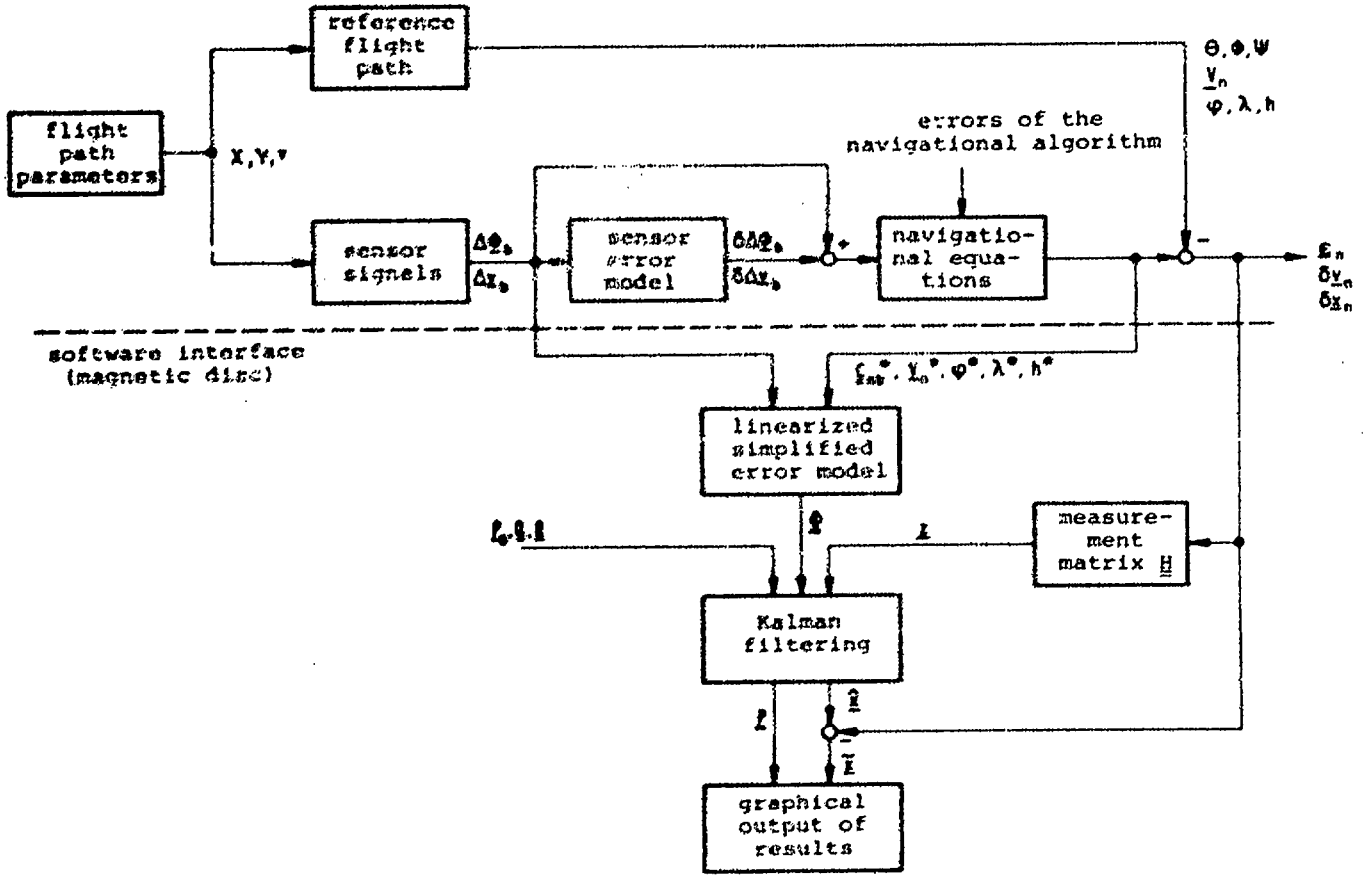


Fig. 3: Construction of the software for the examination of error models

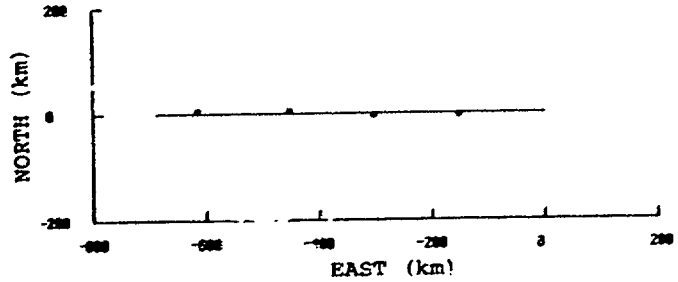


Fig. 4a: Lateral path of flight path 1

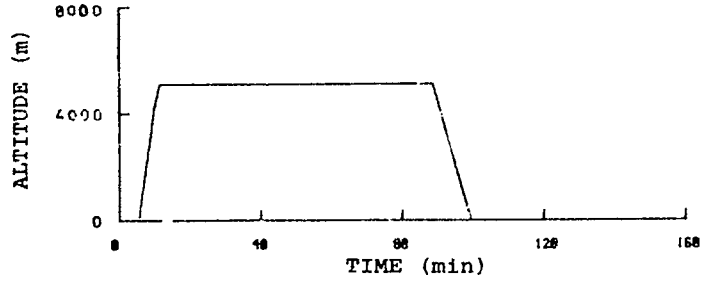


Fig. 4b: Altitude profile of flight path 1

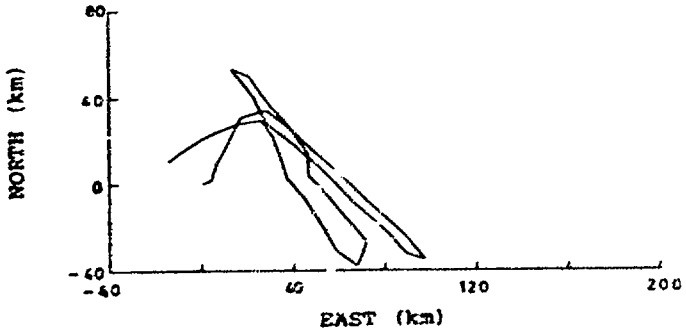


Fig. 4c: Lateral path of flight path 2

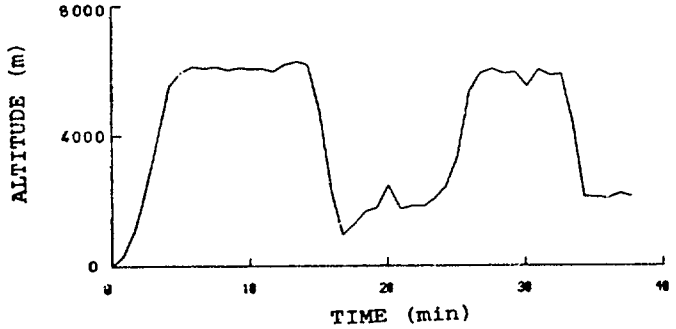


Fig. 4d: Altitude profile of flight path 2

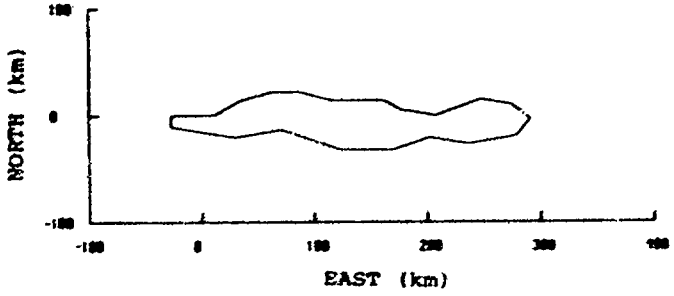


Fig. 4e: Lateral path of flight path 3

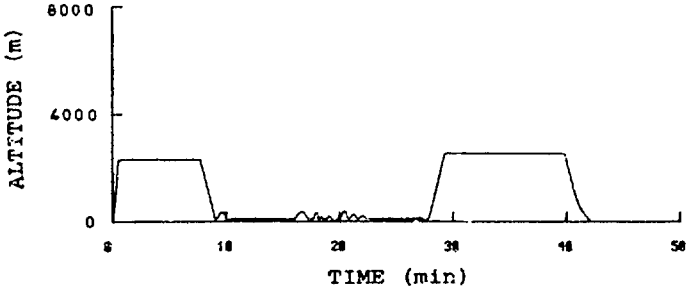


Fig. 4f: Altitude profile of flight path 3

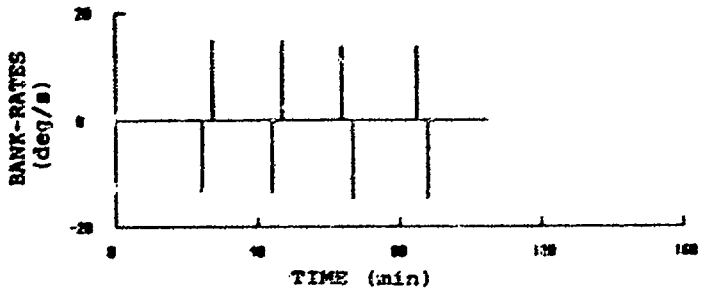


Fig. 5a: Bank-rates of flight path 1

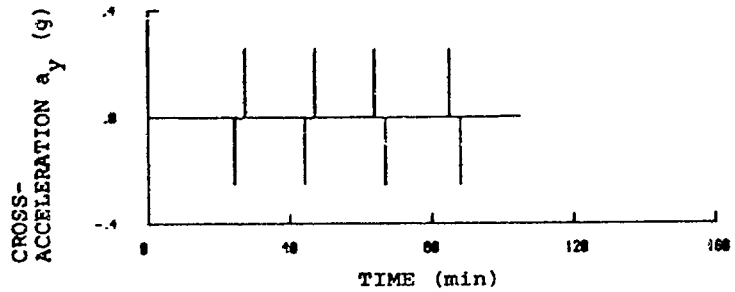


Fig. 5b: Cross-acceleration of flight path 1

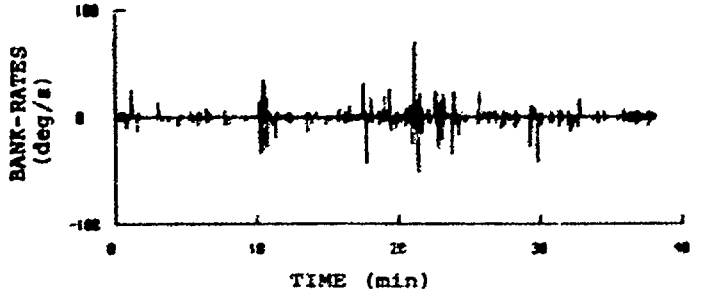


Fig. 5c: Bank-rates of flight path 2

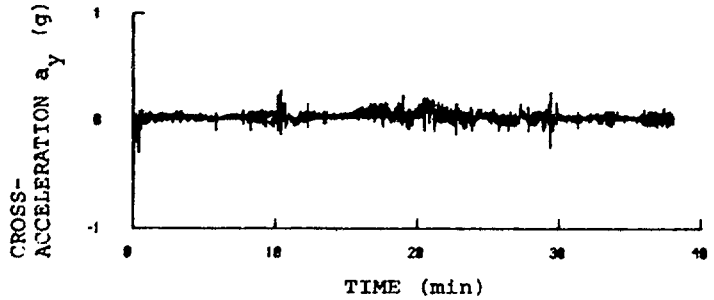


Fig. 5d: Cross-acceleration of flight path 2

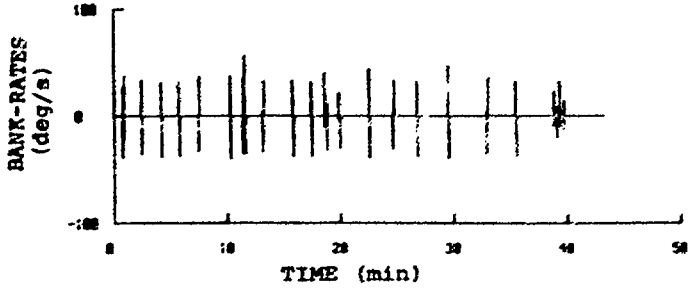


Fig. 5e: Bank-rates of flight path 3

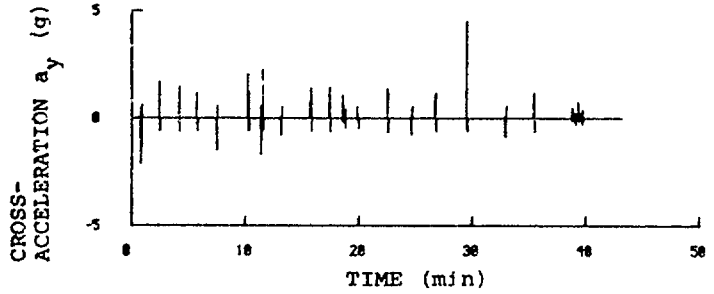


Fig. 5f: Cross-acceleration of flight path 3

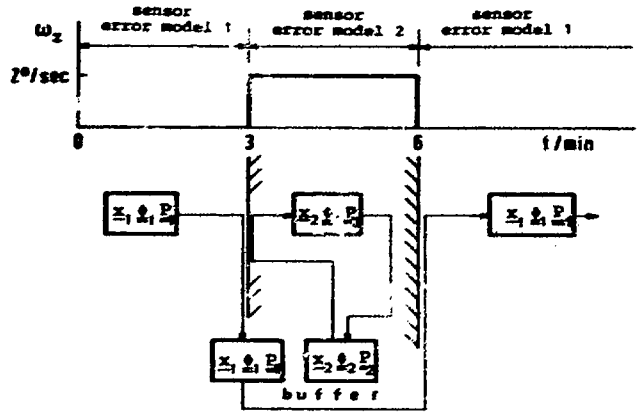


Fig. 6: Principle of the switching procedure

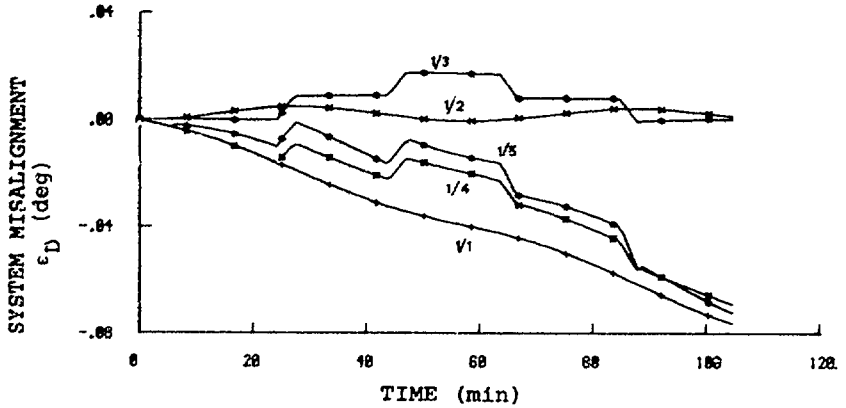


Fig. 7: System misalignment ε_D for flight path 1 with respect to sensor error models 1/1 to 1/5

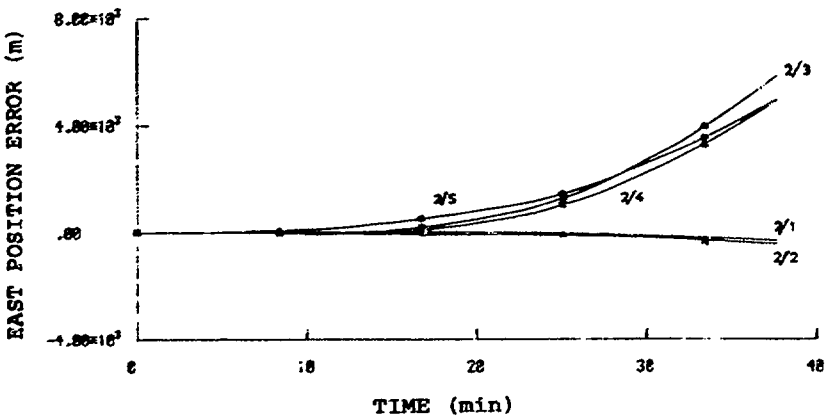


Fig. 8: East position error for flight path 2 with respect to sensor error models 2/1 to 2/5

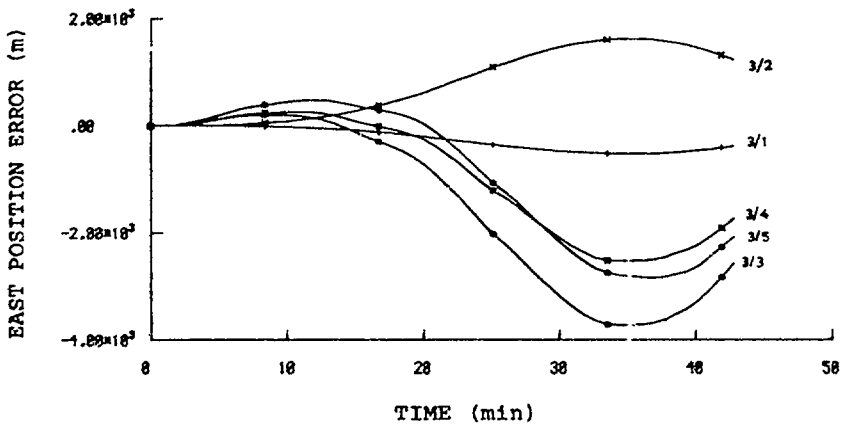


Fig. 9: East position error for flight path 3 with respect to sensor error models 3/1 to 3/5

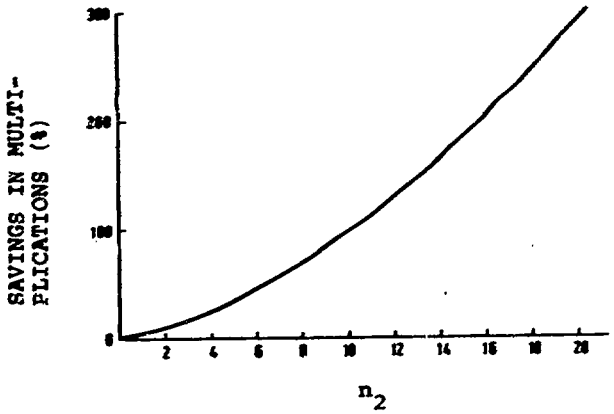


Fig. 10: Savings in the number of multiplications

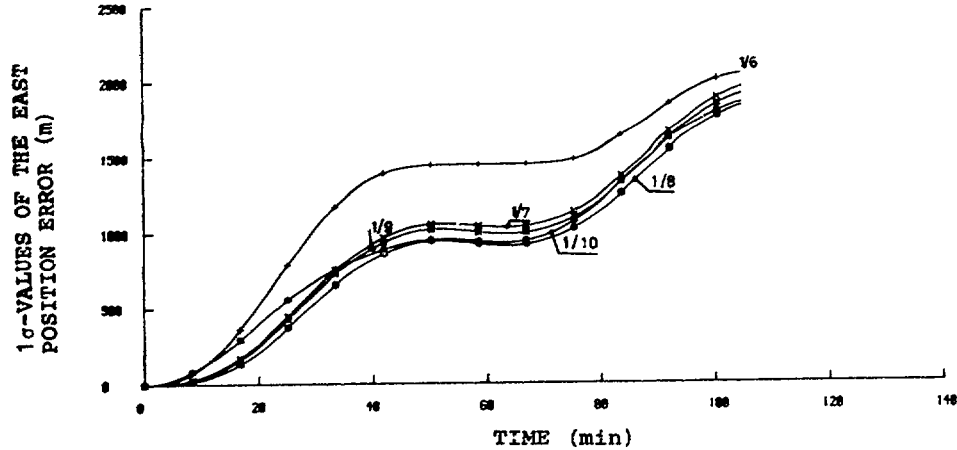


Fig. 11: 1σ-values of east position error for flight path 1 with respect to error models 1/6 to 1/10

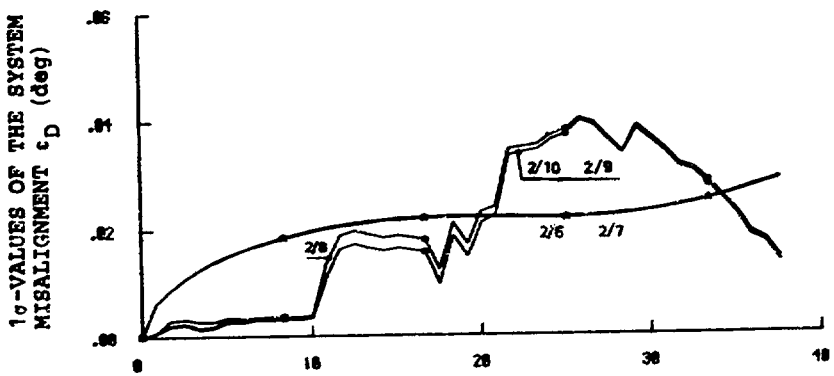


Fig. 12: 1σ-values of the system misalignment ϵ_D for flight path 2 with respect to sensor error models 2/6 to 2/10

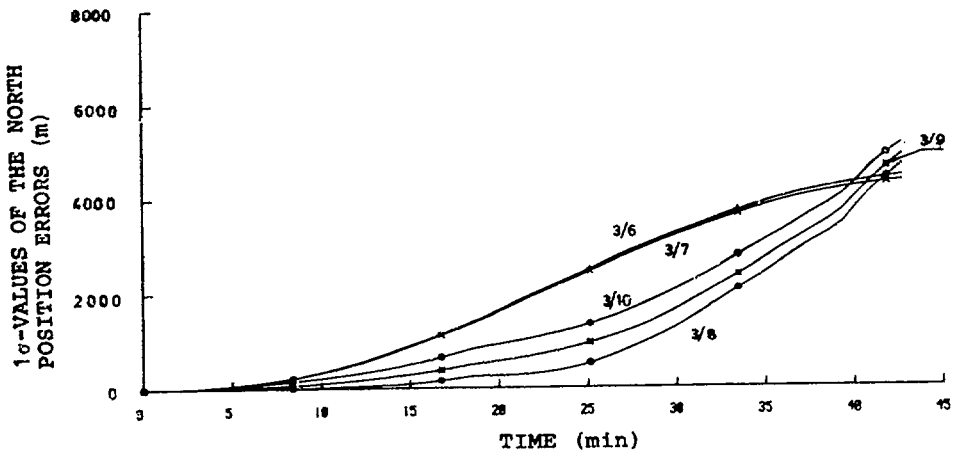


Fig. 13: 1σ-values of east position error for flight path 3 with respect to sensor error models 3/6 to 3/10

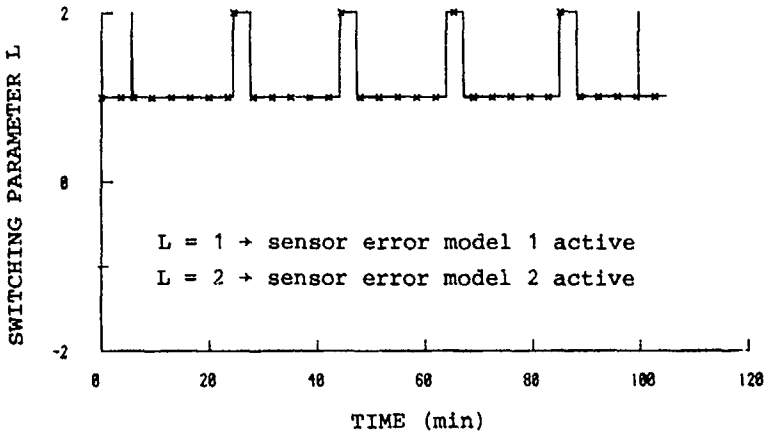


Fig. 14: Switching parameter L for flight path 1

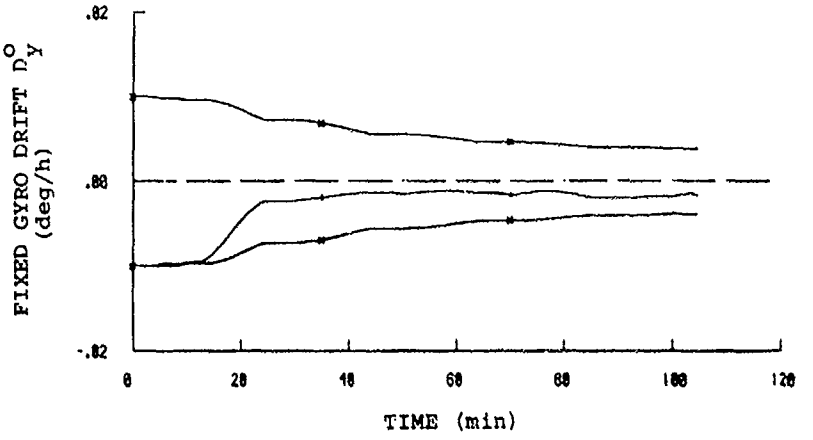


Fig. 15: Estimation error and 1σ-limits for the fixed gyro drift D_y^o for flight path 1

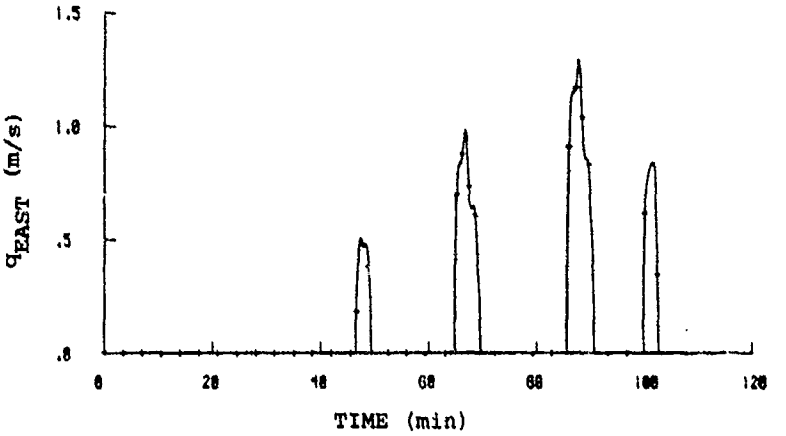


Fig. 16: Element q_{EAST} of the noise matrix Q for flight path 1

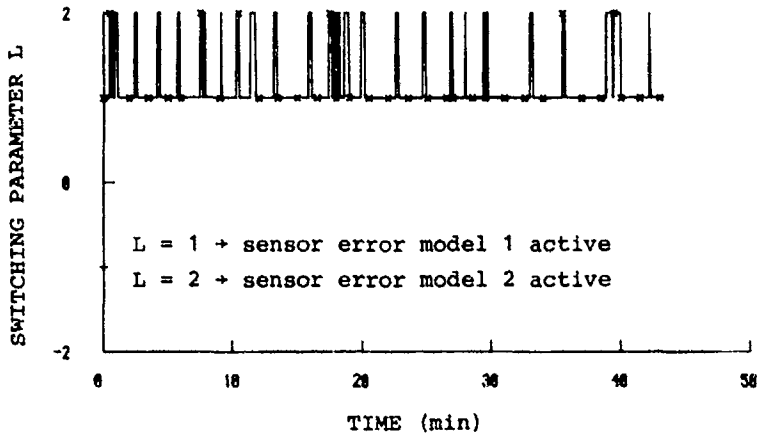


Fig. 17: Switching parameter L for flight path 3

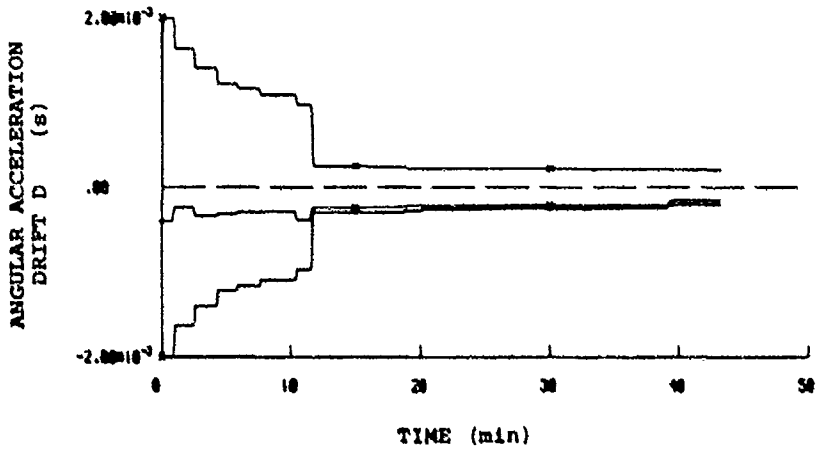


Fig. 18: Estimation error and 1σ-values for the angular acceleration drift $D^{\Delta\omega}$ for flight path 3

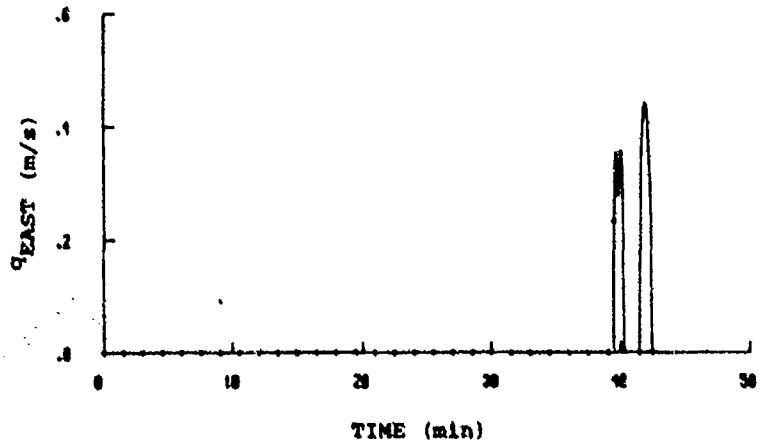


Fig. 19: Element q_{EAST} of the noise matrix Q for flight path 3

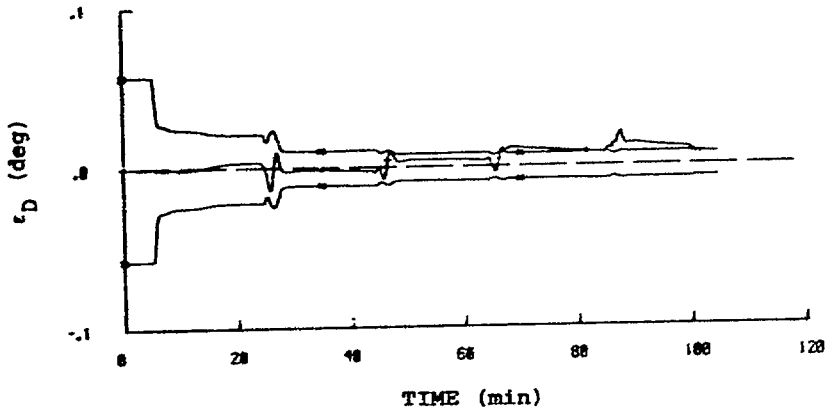


Fig. 20: Estimation errors and 1σ-values for the system misalignment ε_D for flight path 1 with respect to error model 1/9

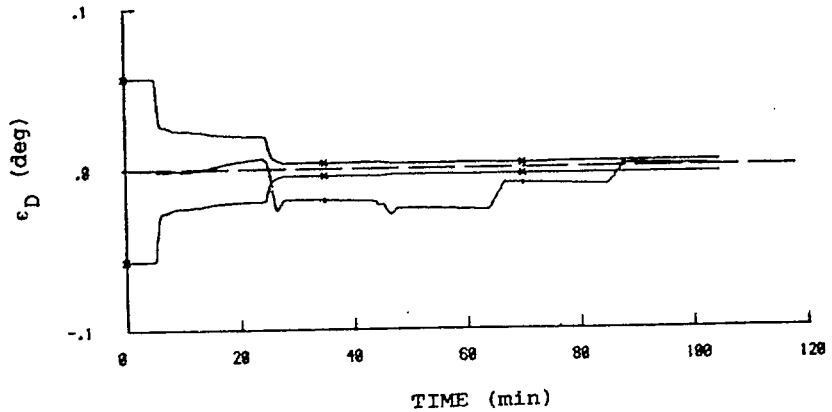


Fig. 21: Estimation errors and 1σ-values for the system misalignment ε_D for flight path 1 with respect to error model 1/6

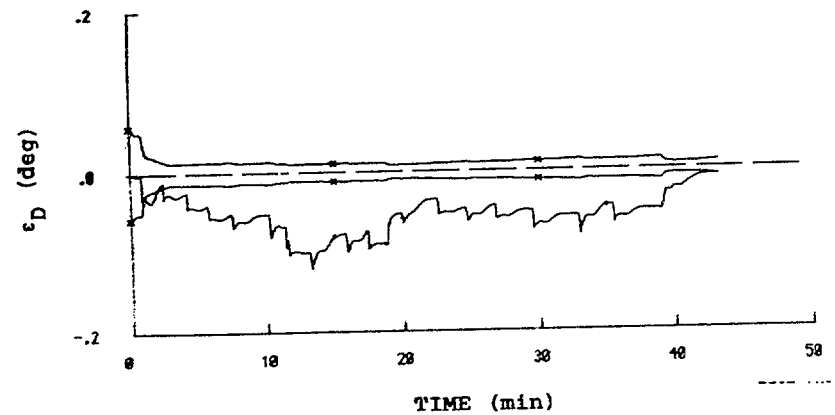


Fig. 22: Estimation errors and 1σ-values for the system misalignment ε_D for flight path 3 with respect to error model 3/6

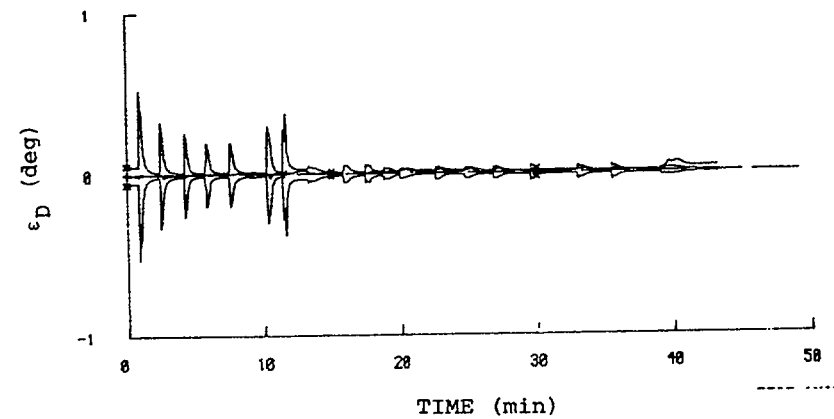


Fig. 23: Estimation errors and 1σ-values for the system misalignment ε_D for flight path 3 with respect to sensor error model 3/9

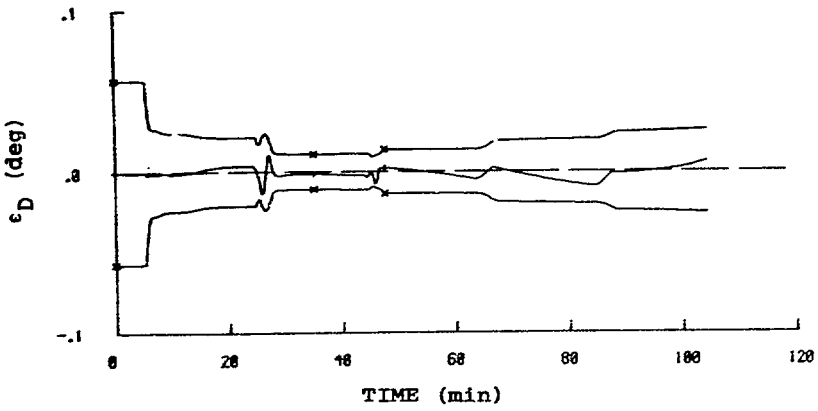


Fig. 24: Estimation errors and 1σ-values of the system misalignment ε_D for flight path 1 with respect to error model 1/9 and a 40 min measurement period

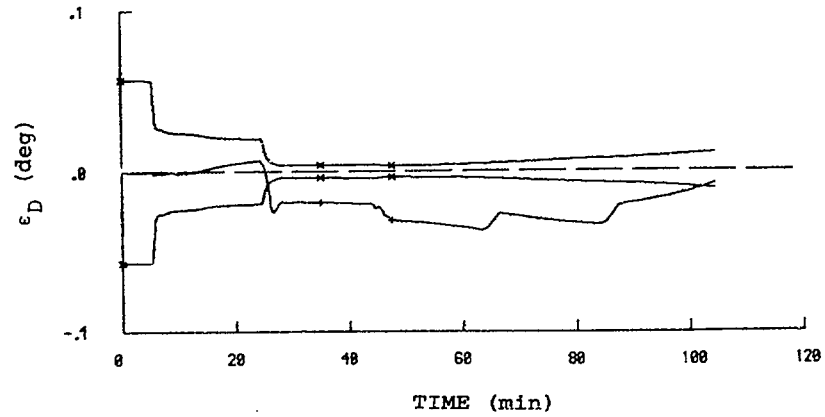


Fig. 25: Estimation errors and 1σ-values of the system misalignment ε_D for flight path 1 with respect to error model 1/6 and a 40 min measurement period

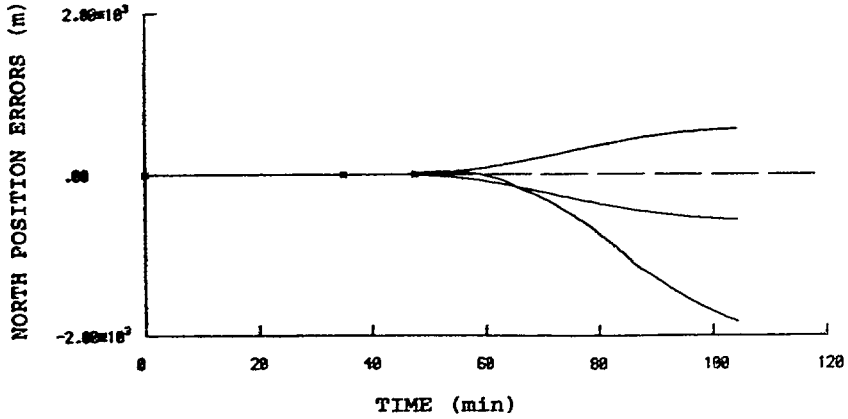


Fig. 26: Estimation errors and 1σ-values of the north position errors for flight path 1 with respect to error model 1/6 for a 40 min measurement period

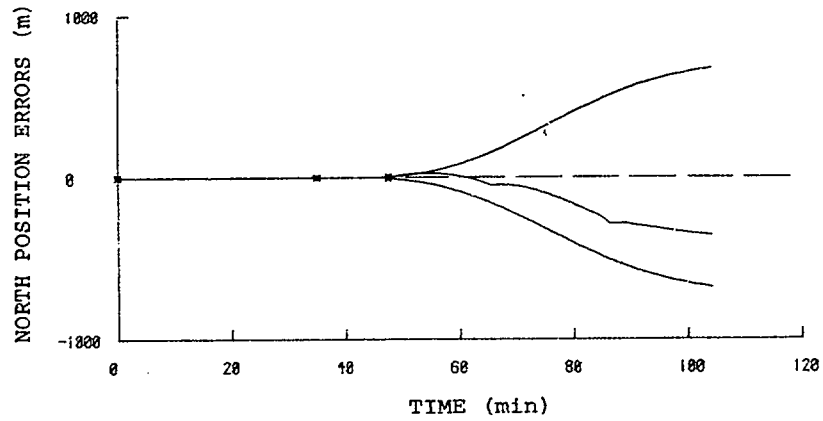


Fig. 27: Estimation errors and 1σ-values of the north position errors for flight path 1 with respect to error model 1/9 for a 40 min measurement period

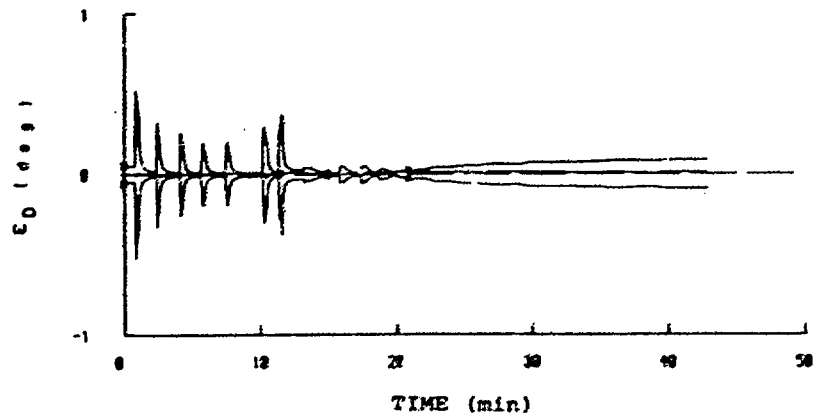


Fig. 28: Estimation errors and 1σ-values of the system misalignment ε_D for flight path 3 with respect to error model 3/9 and a 20 min measurement period

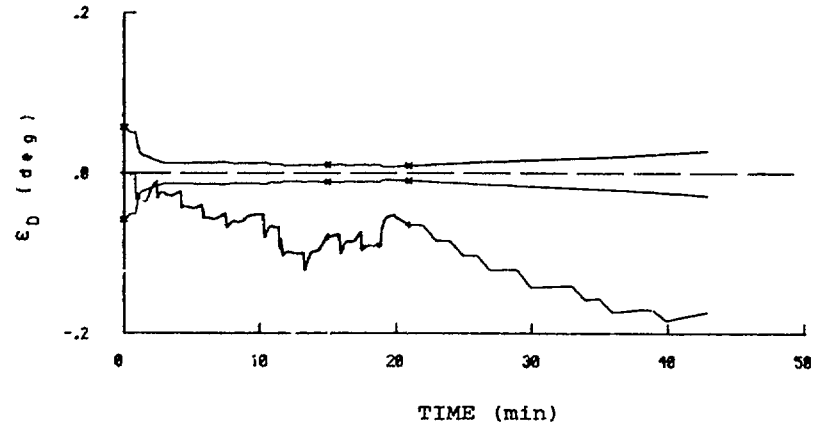


Fig. 29: Estimation errors and 1σ-values of the system misalignment ε_D for flight path 3 with respect to error model 3/6 and a 20 min measurement period

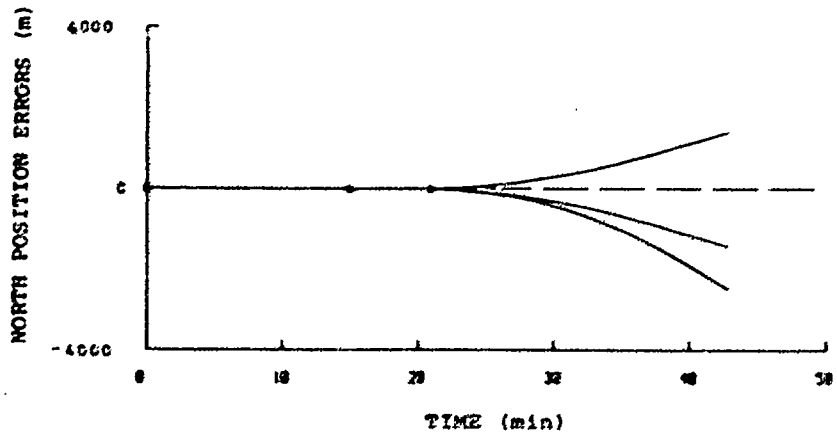


Fig. 30: Estimation errors and 1σ-values of the north position errors for flight path 3 with respect to error model 3/9 and a 20 min measurement period

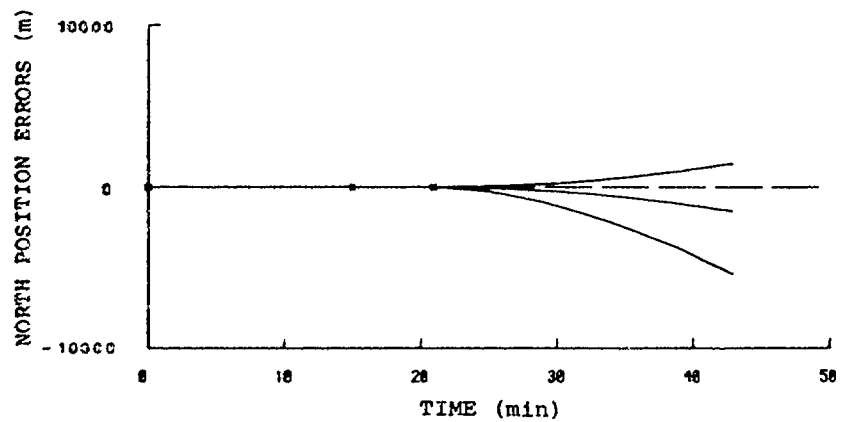


Fig. 31: Estimation errors and 1σ-values of the north position errors for flight path 3 with respect to error model 3/9 and a 20 min measurement period

USE OF FILTERING AND SMOOTHING ALGORITHMS IN THE ANALYSIS OF MISSILE SYSTEM TEST DATA

Edward M. Duiven; Charles L. Medler; Joseph F. Kasper, Jr.

THE ANALYTIC SCIENCES CORPORATION
One Jacob Way
Reading, Massachusetts 01867
USA

The increasing complexity of modern weapon systems demands corresponding increases in the sophistication of the approaches used to test these systems. The following two chapters discuss advanced techniques for the processing of missile system test data. In the first chapter, data from multiple references are used in a post-flight analyzer that is based in large part on a smoothing algorithm. The objectives of the processing are to obtain the best estimate of overall system accuracy and to recover the maximum information about individual guidance system error contributors. In the second chapter, a procedure for validating the models used in filtering and smoothing algorithms is presented. The precedure checks model validity using data from multiple system tests. It employs well-known statistical hypothesis testing methods in an innovative manner.

INTRODUCTION

Modern weapon systems -- particularly ballistic missiles -- have grown in complexity by a significant amount over the past 20 to 30 years. Designers and developers now concern themselves with the total system aspect of missile development. Obtaining a broad system-level understanding of the missile and its environment has become vital.

Increasingly, system-level understanding is supported by modern analytic methods including applications of filtering and smoothing theory. Advances in the theory have taken place in concert with weapon system development. Two inter-related elements of the modern analytic approach are system modeling and system testing. Models, which are mathematical representations of the physical characteristics of a system, have a number of uses.

Models for the errors associated with various system components and subsystems are formulated and then combined to create an overall weapon system error model. Such a model can be used to generate performance projections (e.g., weapon system accuracy) even before the weapon system has been built and deployed. Parameters of the model can also be varied about the nominal values to illustrate where the overall system is most sensitive to variations in subsystem performance. In this way, critical elements in the system design can be identified and given extra attention in the development effort.

Prior to testing, the weapon system error model can be exercised to determine how well a proposed test procedure supports understanding of weapon system behavior. Knowing the ability of a given test to isolate key system performance characteristics is a valuable aid in test program management. Once testing has been performed, data are available to support validation of the various models. Quite often, conditions existing in the test environment must necessarily be different than those which would exist in the operational environment. Models provide the mathematical bridge which enables accurate extrapolation from performance under test conditions to performance under operational conditions.

Finally, models for system behavior are the vehicle which supports development of next-generation, advanced systems. By efficiently characterizing system behavior, models serve as the building blocks for future system design activities. In many ways, they represent the "legacy" of a system development program.

The keys to overall weapon system modeling are proper formulation of component and subsystem models, sufficient testing to obtain data which are representative of all system characteristics and a reliable model validation procedure to ensure that the mathematical representation corresponds faithfully to the actual system.

The two chapters which follow are closely related. They are both involved with the application of model-based modern analytic methods to ballistic missile development.

The first is concerned with the Minuteman III flight test program. In particular, the chapter discusses the use of external reference information to enable determination of specific error characteristics which make up the model for the missile guidance system. It is shown that the ability to identify elements of the model is strongly influenced by the nature of the available reference systems and the chosen test plan. An algorithm based on a Generalized Likelihood Ratio (GLR) test is seen to be effective in isolating certain nonlinear error phenomena, provided that adequate reference data are available.

The second chapter is concerned with validation of the models for multiple phases in the operation of a ballistic missile system. A procedure based on statistical hypothesis testing methods is presented. The procedure can be used to determine whether a proposed test program is capable of isolating bias error phenomena. Alternatively, when data from multiple tests have been collected, the procedure provides a statistical assessment of the presence of bias error phenomena in the system being tested.

BALLISTIC MISSILE GUIDANCE SYSTEM EVALUATION USING MULTIPLE REFERENCES

I. INTRODUCTION

A. OVERVIEW

Historically, ballistic missile system performance evaluation has been accomplished by post-flight processing of guidance system telemetry and radar measurements. This was adequate when the goal of system testing was weapon system accuracy estimation. However, as guidance systems became more accurate, the goal of system testing shifted to the characterization of guidance system errors in the "operational" environment, and it consequently became necessary to upgrade range instrumentation quality.

As a consequence, instrumentation and post-flight data processor development activities aimed at upgrading guidance system test methodologies were instituted [1-3]. This chapter focuses on the test programs initiated by the U.S. Air Force in 1970 culminating in launch of the last Minuteman III Production Verification Missile (PVM) in March 1980. Over the 10 years, a series of programs directed at improved guidance system characterization and advances in the quality and availability of external reference systems were initiated.

Section II is an overview of the USAF test programs. In Sections III and IV, a discussion of the methodology and performance evaluation studies undertaken to "optimize" the recovery of guidance system error characteristics is presented. Section V discusses a data processor -- the Post Flight Analyzer (PFA) -- developed to evaluate data from a series of test programs. The processing of the flight test data, and the associated results, are described in Section VI. Finally, Section VII presents a summary and suggests enhancements that may be desirable for future ballistic missile testing.

B. ACKNOWLEDGMENT

The data processing methodology described in this chapter, as well as the performance evaluation and data analysis software, is the result of a dedicated effort by a number of TASC employees. The original GLR test methodology was formulated by H.L. Jones and refined by K.S. Tait. The performance evaluation software was developed by J.A. D'Appolito, C.M. Ermer, L.M. Hawthorne, and D.J. Meyer. C.J. Vahlberg and D.J. Meyer developed the data processing software.

II. BACKGROUND

In 1971, with the Minuteman III test program approximately two-thirds complete, a significant decision was made by the USAF; it resulted in the termination of system testing at the Eastern Test Range (ETR) for cost reduction. Prior to this, approximately one-half of the test missiles had been launched at the ETR. Subsequently, flight tests were conducted only at the Western Test Range (WTR). The high accuracy tracking capability and the good geometry characteristics available at the ETR were sacrificed.

The guidance analysis community began to recognize the limitations of the test methodology in operation. The level of accuracy desired from the test program could not be achieved with the external reference systems available at the WTR. Also, the quantization levels associated with guidance telemetry data were not consistent with the levels required for guidance system error characterization. The USAF began to look at alternative reference systems that could overcome test limitations.

At approximately the same time the Charles Stark Draper Laboratory (CSDL) was actively developing a floated inertial reference platform, known as AIRS, the Advanced Inertial Reference Sphere [4]. The instruments developed for AIRS were designed to provide system quality an order of magnitude better than the primary Minuteman III guidance system (the NS-20).

The AIRS development schedule and the need for an improved reference system were nearly coincident. As a result, the USAF instituted the Missile Performance Measurement System (MPMS) Program. An AIRS platform was incorporated in a separate wafer* along with its associated electronics, power supplies, telemetry unit, cooling system, etc. The primary guidance system computer was modified to allow for the time-tagging of guidance system outputs (integrated specific force). Time-tagging was a means of eliminating the impact of large quantization levels on the recovery of guidance system errors; it reduced quantization-induced errors by approximately an order of magnitude.

Although originally planned as a multi-missile test program, only one MPMS missile was flown. The test was conducted on special Test Missile No. 11 (STM-11) on 15 July 1976. AIRS functioned well and much was learned about the performance of the AIRS hardware as well as NS-20 instrument error characteristics. However, cost considerations dictated that that program be limited. MPMS led to the FLY-2 Program in which the AIRS platform was replaced by a second NS-20.

*Minuteman III is capable of carrying one or more insertable missile body sections (wafers) between the fourth stage motor and the payload section.

In principal, the FLY-2 concept would appear less than optimal. In most test programs it is desirable that the measuring device be an order of magnitude more accurate that the system being tested. However, by taking advantage of the NS-20 instrument orientations (Figs. 1 and 2) and the fact that the platforms can be aligned, in azimuth, to any desired orientation, within gimbal constraints (Fig. 3), the reliance on "identical" systems can be minimized. Thus, for FLY-2, it became a question of how to orient the platforms to achieve "optimal" recovery of a set of "primary error sources." Optimization study results are discussed in Sections III and IV.

A series of three FLY-2 missiles were flown between November 1976 and June 1977. Data from the three flights provided significant insight into a number of error contributors not previously included as part of the system error model. However, FLY-2 was limited in its performance assessment capabilities due to the lack of an independent reference with an accuracy superior to the WTR radars.

This shortcoming was resolved with the introduction of a GPS* receiver within the FLY-2 wafer. The original intent of the GPS Receiver Test Program (GPS/RTP or FLY-2/GPS) was to demonstrate that the receiver could provide an accurate post-boost update in the ballistic missile environment. Since it was projected that the receiver would maintain lock† on the signals transmitted by the satellites, it was determined that the range and range-rate measurements should be used for post-mission evaluation.

Two test flights were performed as part of the Minuteman III test program. Test missiles PVM-18 and PVM-19 were launched on 31 January 1980 and 27 March 1980. The presence of an accurate external reference, in addition to the FLY-2 configuration, provided the best data collected during the Minuteman III test program. In summary, the USAF had pursued a course that led to ever-increasing test capability without having to make major changes to an overall test philosophy.

III. EVALUATION METHODOLOGY

The methodology used to "optimize" FLY-2 (and subsequently FLY-2/GPS) performance is very strongly tied to the objectives set forth for the test programs. Thus, it is important that these objectives be presented and the test program goals be put into perspective. The goals of FLY-2 were to:

- Validate the effect on system accuracy of preflight software (Ground Program) modifications introduced as a consequence of the Guidance Improvement Program†
- Increase the understanding of a number of "priority" error sources included in the guidance system error model
- Detect and identify "unmodeled" error sources
- Identify sources of anomalous performance using the unique data characteristics available from the FLY-2 instrumentation system.

The same goals were established for FLY-2/GPS with the additional goal of demonstrating GPS receiver performance in the "operational" environment.

The priority error sources are: 1) initial azimuth misalignment, 2) accelerometer cross-axis compliance, 3) gyro g^2 and g^4 coefficients, and 4) platform compliance. The error mechanisms for these quantities are included in the Minuteman III guidance error model. However, the coefficients typically could not be separated during static testing;§ sled testing does not provide the appropriate dynamics for coefficient observability. Thus, these quantities may be "observed" only during powered flight. The goal of FLY-2 and FLY-2/GPS was to provide data to assist in characterizing these priority error sources.

The unmodeled errors of interest can actually be called mismodeled errors. There has been speculation that certain of the "bias" error coefficients have time-varying characteristics - specifically shifts and/or ramps. This type of error, if present in the guidance system, could have a significant impact on weapon system accuracy. In addition to the bias shift/ramp type of error, additional unmodeled errors include gyro g^2- and g^4-sensitive error coefficients. These "unmodeled" errors could be addressed using FLY-2 and/or FLY-2/GPS data -- once suitable models had been developed for them.

*Global Positioning System - A satellite navigation system being developed by a tri-service Joint Program Office [5].

†Except during the staging events when high acceleration rates are present due to motor shutdown and startup.

†One in a series of accuracy upgrades made to the Minuteman III guidance system between 1970 and 1976.

§Static testing implies all testing in a 1-g field, including tumble and vibration testing.

Finally, sources of anomalous performance are those error characteristics that were not anticipated but had been discovered as a result of data analysis. A variety of these surfaced during the test programs. However, no evaluation studies had been undertaken, ahead of time, to assess the ability to detect and isolate sources of anomalous performance.

Based on the objectives discussed above, there are two criteria by which the test program may be assessed (optimized): 1) recovery of the priority error sources in a post-mission data evaluation environment, and 2) detection and identification of unmodeled (or improperly modeled) error characteristics.

The USAF test program objectives called for post-mission processing of test data to extract information about the priority error sources and the "unmodeled" errors. It was determined that the processor would be based on a Kalman filter [6,7]. The filter estimates the priority error sources and the "unmodeled" errors incorporated in the filter, to some level of confidence. To address the question of the presence of instrument error coefficient shifts and/or ramps, a new methodology was developed. The technique referred to as the Generalized Likelihood Ratio (GLR) test is a direct extension of the Kalman filter. The GLR tests use filter residuals to determine whether there are any unmodeled errors (bias shifts and/or ramps) that would cause the residuals to be other than a zero-mean, white-noise sequence.

Figure 4 depicts the flow of data through the post-flight evaluation software. The filter processes the radar, dual NS-20 guidance telemetry data, and, if available, GPS measurements to estimate the errors in the filter model. The filter minimizes, in a mean-squared error sense, the error between the actual measurements and those predicted by the model. The NS-20 error model assumes that the principal instrument errors are biases over the period of powered flight. It is well known [6] that the sequence of measurement residuals will be a zero-mean white-noise sequence if the filter models are correct.

However, if certain of the error coefficients display sudden shifts or ramping characteristics, the model is not correct and the measurement residuals will not be white and zero-mean. The GLR algorithm tests the mean and whiteness of the filter residuals [8]. The test is a two-step process. The first step determines whether a shift in one (or more) of the coefficients has taken place. This is referred to as the detection process. Detection is performed by forming a weighted sum of the last M measurement residuals* and using this quantity as a test statistic in a binary hypothesis test. If the test statistic, λ, is greater than a specified threshold, ε, a shift is detected. If it is smaller than ε, no shift is assumed to be present.

The second step is identification. The GLR formulation results in an algorithm that generates an estimate of the state that shifted, the time of the shift, and the shift magnitude. Under the assumption of no a priori knowledge about the jump characteristic the GLR estimate is "optimal." The capability to estimate the jump characteristics makes the GLR test more attractive than other residual-based detection processes [9].

The critical parameters of the GLR test design are the Probability of False Alarm (P_F) and the Probability of Detection (P_D). The Probability of False Alarm is defined as the probability that a shift will be detected when no shift occurs. It is shown in [8] that the higher the value of ε selected, the lower the Probability of False Alarm. However, that is not the only trade-off.

The Probability of Detection, defined as the probability that a jump (if present) will be detected, is a function of the shift detection threshold, ε, and the window length, M, as well as the magnitude of the jump itself, v [8]. As seen in Fig. 5, the longer the detection window, the higher the Probability of Detection for a given P_F. However, the length of the window is limited by the missile flight time, computational capabilities of the hardware used for post-flight processing, and the fact that multiple jumps may occur during the flight.

The methodology used for evaluation of the FLY-2 and FLY-2/GPS flight test programs is depicted in Fig. 6. The same error covariance analysis procedures were used for both the FLY-2 and FLY-2/GPS studies. Only the FLY-2/GPS simulation is discussed here.

There are three steps involved in the generation of projected FLY-2/GS performance estimates. The first step is simulation of the GPS segment. It is necessary for two reasons:

- To develop a time history of GPS satellite orbital positions and velocities so that proper accounting is made for GPS/missile geometry
- To generate the GPS satellite position, velocity, and clock calibration error covariances.

In Fig. 6, the FILTER module represents the second step, recursive solution of the filter error covariance propagation and update equations. These equations are solved once

*The quantity M is referred to as the GLR detection window length.

for a specific GPS satellite measurement schedule. The outputs of the FILTER module are time histories of filter-indicated performance and the Kalman gain matrices.

The third step in the evaluation process, the SYSTEM module in Fig. 6, involves recursive solution of the linear system error covariance equations. These equations are solved repeatedly to produce an error budget, using the same Kalman gain file each time. When all error contributors have been evaluated, the overall measurement system performance projection can be calculated from the detailed error-source-by-error-source breakdown. This analysis produces the following benefits:

- Determination of key error contributors - indicating where to focus attention for subsequent performance improvements
- Identification of insignificant error contributors - indicating where a less costly (i.e., poorer quality) subsystem might be substituted with minimal performance degradation.

The methodology presented in this section was used for "optimization" of FLY-2/GPS performance. A more detailed discussion of the error covariance methodology can be found elsewhere [5,10].

The FLY-2/GPS error covariance simulations determine the Kalman filter estimation error covariance matrix based on a sequence of measurements. There are three sets of error measurements associated with the GPS-RTP. The first is the difference between the two IMU velocity measurements, the second set of error measurements are those associated with the range radars, and the third set results from processing of the GPS measurements.

The error sources for FLY-2/GPS are those associated with the two IMUs, the radar and the GPS satellites. Table I lists the errors modeled for each of the IMUs and selected for use in the Filter Model and Truth (i.e., system) Model formulations.

Table II lists the error sources associated with the WTR measurements. Error sources associated with the GPS satellite, propagation delays, the missile receiver are given in Table III. The uncertainties in satellite position, velocity, etc. are provided by a program which simulates the GPS satellite ground tracking process, and determines the estimation error covariance for the GPS satellites. The propagation errors and carrier and code-loop errors listed in Table III are modeled as white measurement noise sequences in the simulation and are not estimated.

IV. PERFORMANCE PROJECTIONS

FLY-1 (or single IMU vs. radar) performance was evaluated using several trajectories that emulate nominal missions flown from Vandenberg Air Force Base (VAFB) to the Kwajalein Atoll. FLY-1 performance was developed as a baseline against which FLY-2 and FLY-2/GPS performance can be compared. The nominal ground track and specific force profiles for these trajectories are shown in Figs. 7 and 8, respectively. For these analyses, the azimuth offset angle, $\Delta\alpha$, (Fig. 3) is assumed to be zero. Range measurements from the South Vandenberg Air Force Base, Point Mugu, and Pillar Point radars were assumed available every half-second beginning 15, 40, and 50 seconds into the mission, respectively.

FLY-1 results are presented in terms of normalized (unitless) quantities called recovery ratios. Two types of recovery ratios are of interest: 1) guidance error recovery ratio, and 2) error coefficient recovery ratio. The former is defined as

$$R_G = \frac{\text{RMS Error In Estimate of Guidance Quantity}}{\text{RMS Guidance Error In Absence of Tracking}} \qquad (1)$$

These are obtained for:

- Down-Range, Cross-Track, Vertical Position and Velocity Errors at Reentry Vehicle Deployment
- Down-Range and Cross-Track Miss Distances
- CEP [11]
- Initial Azimuth Misalignment.

Recovery ratios for these quantities are always less than or equal to 1.00; the smaller the value, the better the recovery of the error of interest.

The error coefficient recovery ratio (R_C) is the ratio of the final rms uncertainty in the estimate of the error coefficient, σ_F, to the initial rms or a priori uncertainty, σ_0. That is:

$$R_C = \frac{\sigma_F}{\sigma_0} \qquad (2)$$

Down-range and cross-track guidance error recovery ratios for FLY-1 are given in Table IV. Vertical position and velocity ratios (not shown) are essentially identical to the down-range numbers. The processing of combined radar and NS-20 data yields cross-track error reductions of 40 to 50 percent. Crosstrack miss distance recovery is essentially equal to cross-track velocity error recovery because cross-track velocity error at boost-burnout is the major source of cross-track miss. Unfortunately, processing of the combined radar and single NS-20 data does not produce any significant improvement in down-range (or vertical) guidance error estimation as a consequence of radar accuracy and geometry relative to the missile trajectory. The Vandenberg and Point Mugu stations essentially provide only down-range information. However, the NS-20 IMU is more accurate in the determination of down-range position than the WTR radars. Thus, the NS-20 "calibrates the down-range radars." Point Pillar provides good cross-track information and is the source of recovery for these errors. The reduction in error of the predicted impact point (i.e., CEP recovery) results solely from the reduction in cross-track miss prediction error.

Priority error source recovery ratios for FLY-1 are summarized in Table V. This table lists the smallest recovery ratio attained for a given coefficient over all simulated flights. Since the radar data basically yields cross-track information only, a 20% reduction in initial azimuth error is attained. However, processing of FLY-1 data produces no significant recovery of any of the remaining priority error sources.

The guidance error and error coefficient recovery ratios are excellent measures of flight test performance; however, considered individually they are too numerous to use in a meaningful optimization criterion. Furthermore, there is no single flight test configuration which simultaneously minimizes all recovery ratios of interest. Instead, two simple measures of performance, one for guidance error recovery, and a second for error coefficient recovery were developed.

In the course of the optimization studies, approximately 150 FLY-2 flights were simulated. Table VI lists, for each priority error source the best (i.e., smallest) coefficient recovery ratio attained over all flights. It must be emphasized that no one flight simulation yielded all these results.

If a particular error source coefficient strongly influenced the error behavior of an IMU, simply averaging the outputs of two systems (under the assumption that the error sources in both systems are equal in rms value and uncorrelated) would reduce the effect of that error source on system error by a factor of $1/\sqrt{2}$ or 0.71. It could be argued that coefficient recovery ratios greater than 0.71 are not significant. Table VII shows that FLY-2 produces no significant recovery of accelerometer or platform compliance coefficients. The same is true for gyro bias and g-dependent drift rates. In fact, of all the priority error source coefficients only gyro g^2- and g^4-dependent drift rates are recovered at a significant level. For these coefficients it is convenient to define a composite coefficient recovery ratio:

$$R_{COMP} = \frac{1}{4} [\min R_{\delta E} + \min R_{\delta B} + \min R_J + \min R_P] \qquad (3)$$

where

min R = smallest recovery ratio attained for a given coefficient in a given run

δE, δB signify gyro g^2-dependent drift coefficients

J,P signify gyro g^4-dependent drift coefficients

R_{COMP} and R_C are used as measures of FLY-2 performance for optimization purposes.

A major concern was selection of primary and secondary NS-20 IMU azimuth offsets and trajectory reentry angle to optimize (i.e., minimize) the guidance error CEP and composite coefficient recovery ratios. Three-axis velocity difference data and radar tracking data were processed every 4.5 seconds throughout the boost phase using a Kalman filter algorithm to estimate guidance errors, instrument and platform error coefficients, and initial alignment errors.

With regard to azimuth offset angle ($\Delta\sigma$) optimization, one might assume that simultaneous offset of both IMUs is desirable. However, for the optimization criteria selected, this is not the case. To illustrate, results of two test cases from the series of medium reentry angle studies are summarized in Table VII. The flights had a fixed azimuthal difference of either 30 or 45 deg between the two systems. However, the orientation of the primary guidance system (System 1) was varied about the $\Delta\sigma$=0 deg orientation. The composite coefficient recovery ratio and the CEP recovery ratio for these flights are also presented.

For a fixed azimuthal difference, superior recovery always occurs when one system is launched with zero offset. This was observed to be the case at all reentry angles. In this orientation the gyro and accelerometer errors contribute the least to guidance errors. When both IMUs are offset so that their individual contributions to guidance errors are comparable, the optimal post-flight data processor cannot distinguish between the two systems. Thus, recovery ratios are poor. Conversely, when one system is placed

on-axis, its contribution to guidance errors is greatly reduced and the errors of the off-axis system become more observable. The on-axis IMU becomes the reference through which errors in the off-axis system are recovered.

Azimuth angle offset optimization studies were performed, with one platform always at zero offset, for low, medium, and high reentry angles with results shown in figs. 9, 10, and 11. All three sets of results are quite similar and demonstrate that initial azimuth error recovery shows little variation with offset angle. This is because much of the initial azimuth error recovery comes from radar tracking data and not IMU velocity difference data.

The guidance CEP recovery ratio shows some variation with azimuth offset angle. However, the variation is not pronounced. Guidance CEP recovery is a minimum when $\Delta\alpha$ = 180 deg, i.e., when the level platform axes for the two systems are antiparallel. However, actual implementation of this configuration is not possible due to guidance system gimbal constraints.

Referring again to Figs. 9 through 11, the composite coefficient recovery ratio shows the greatest variation with azimuth offset angle of any of the recovery ratios considered. Furthermore, for all reentry angles considered, this ratio reaches a minimum with a 30 deg offset of the secondary IMU. This minimum is fairly broad, however, providing low composite coefficient ratios in the range of 22.5 to 45 deg. Also, the composite coefficient recovery ratio plot is symmetric about zero degree offset so that both positive and negative offsets are useful.

As a consequence of these optimization studies, the following conclusions were drawn:

- The primary guidance system should be aligned to the target azimuth ($\Delta\alpha$ = 0 deg)
- The secondary guidance platform should be offset 22.5 to 45 deg from the primary guidance system
- The reentry angles for FLY-2 should be in the middle of the systems capability range.

The first two recommendations were followed on all three FLY-2 flights. However, a range of reentry angles was selected so that specific error coefficient recovery could be emphasized rather than minimization of the composite performance index.

Having addressed the "optimization" issues associated with FLY-2, it is possible to assess GLR test performance. The GLR test was specifically designed to detect and identify shifts or ramps in certain guidance system instrument error coefficients. Attention is directed here to the detection and identification of shifts in the accelerometer bias and/or gyro bias drift coefficients. The results are based on the same model used in the optimization studies. In addition, two forms of the GLR test mechanization are considered - Fixed-Lag and Fixed-Interval.*

Fixed-Lag GLR is based on a data window (M) of fixed length. The relationship between window length and detectable jump magnitude is shown in Figs. 12 and 13. For shifts in both accelerometer bias and gyro bias drift, there is an asymptotic relationship between detectable jump magnitude and window length. The minimum acceptable window length is approximately 20 (100 sec for the 5 sec sampling interval). However, maximum detectability for all possible jump times would require window lengths on the order of 45 (225 sec). Figure 12 indicates that accelerometer shifts on the order of 10σ (σ is the initial rms uncertainty) are detectable, with a false alarm probability (P_F) of 0.05 and a detection probability (P_D) of 0.50. For gyro bias drift, shifts on the order of 50σ are detectable if they occur prior to third-stage thrust termination.

The regions of superior shift detectability are shown in Figs. 14 and 15.† Accelerometer bias shifts are most detectable if they occur after burnout. The poor performance prior to burnout is caused by the large specific force components exciting the higher-order accelerometer and gyro error terms. Thus, bias shifts must be large relative to the specific force effects on the g-dependent errors if they are to be observable. After thrust termination, the detection of smaller shifts is possible (for the same P_D) because of the lower specific force component magnitudes.

It is well known [11-14] that gyro bias drift errors enter guidance velocity error through the term $\psi \times \underline{f}$, where ψ is the platform misalignment vector resulting from gyro drift errors and $\underline{f}$ is the specific force vector. It follows that gyro bias drift coefficient shift detection should be the best during the period of powered flight when $\underline{f}$ is maximum. The data plotted in Fig. 15 substantiates this premise.

*The terms Fixed-Interval and Fixed-Lag were selected because of the close association of the formulations to the Fixed-Interval and Fixed-Lag Smoothers [6].

†Fixed-Lag and Fixed-Interval results are shown in Figs. 14 and 15 for comparison; only the latter are discussed.

Fixed-Interval GLR uses a variable length window which runs from the candidate jump time to the end of the data interval. Figures 12 and 13 show the effect of the increased window length on jump detection. The projected detection performance of Fixed-Interval GLR is presented in Figs. 14 and 15 for comparison with the Fixed-Lag algorithm. These figures show the jump magnitude required to produce a detection probability of 0.50 when the threshold is set for a false alarm rate of 0.05. Naturally, the larger windows result in improved performance for all cases; however, the improvement is most dramatic for accelerometer jumps prior to burnout (200 sec).

Accelerometer bias jumps of about 2σ are uniformly detectable throughout the flight. The significant difference between the two GLR mechanizations in detecting jumps before burnout is explained as follows. The Fixed-Interval algorithm always has available the filter residuals after burnout where the effect of an earlier accelerometer shift is highly observable. The Fixed-Lag version lacks this information and is unable to identify a small shift in the presence of large g-sensitive error coefficients.

Significant information concerning a variety of the priority errors may be obtained using FLY-2 flight test data. However, a number of shortcomings were identified based on insights gained during the performance evaluation and optimization studies. These shortcomings were borne out during subsequent data processing activities.

Principal among the shortcomings is the inability to distinguish between certain error sources whose signatures, in measurement space, are nearly identical. It is impossible, for example, to separate between initial primary and secondary IMU azimuth misalignments; consequently, the need for an accurate, independent position/velocity reference is apparent. The advent of the GPS-Receiver Test Program (GPS-RTP) was most timely since it provided the potential for uniquely accurate reference system measurements. Figure 16 shows the GPS satellite geometry anticipated for the missile test dates.

The incorporation of the GPS measurements significantly improves the capability to estimate guidance-system-induced deployment errors as well as initial azimuth misalignment. Table VIII summarizes the results of the FLY-2/GPS performance evaluation study. These results represent those associated with a medium reentry angle trajectory. The secondary system was offset 45 deg from the primary. The table also contains the projected performance for FLY-1/Radar and FLY-2/Radar. It is apparent that overall weapon system test program performance could be greatly enhanced via the use of GPS data.

Recovery of the priority error sources is also improved. Table IX presents the secondary system guidance error coefficient recovery capability, for a particular FLY-2/GPS mission. These are the error sources that demonstrate the significant recovery capability. FLY-1/Radar and FLY-2 performance projections are also included.

Certain of the recovery ratios tend to remain large (poor recovery), irrespective of the measurement type or quality. This is a consequence of the processor's inability to separate the various error sources. Consequently, for evaluation of future systems, a new methodology to define filter models, filter dimensions, etc. that recognizes the limitation of "optimal" data processors should be developed. In addition, prelaunch and flight test data must be processed in a complementary manner to provide maximum system understanding relative to each type of data.

The value of GPS measurements in the detection of instrument coefficient shifts is demonstrated in Figs. 17 and 18. A factor of two to three improvement in detection capability can be achieved with the incorporation of the GPS information. Other than this improvement, however, the characteristics of the detection process are unchanged. The GLR methodology must be modified to account for the inability of the GLR test to identify, with high confidence, a number of instrument error characteristics.

V. DATA PROCESSOR STRUCTURE

The top-level structure of the Post Flight Analyzer (PFA) is depicted in Fig. 19. The PFA is structured such that guidance system initial condition errors and instrument error coefficient shifts are determined using a five-step process:

- Data Preprocessing
- Filter Analysis
- Model Analysis
- Jump Detection and Identification
- Decision Making.

In data preprocessing, telemetry data from two MS-20s, the GPS measurements, and data from several radars are sorted, time synchronized, compensated for deterministic errors, rotated to appropriate coordinate frames for comparison, and combined such that all relevant high-rate data is reduced to a rate suitable for the advanced analysis tools. Since this step involves substantial computation, the PFA allows parallel processing of each data type, thereby decreasing the preprocessor timeline substantially.

The filter analysis programs take the sequence of range and range-rate difference measurements (GPS and/or radar) and velocity difference measurements (between the

two NS-20s) provided by the preprocessor and calculates estimates of NS-20 error coefficients (primary and secondary systems), GPS errors, and radar errors. The estimation process is carried out with a suboptimal Kalman filter augmented with smoothing capabilities for key epoch times of the missile flight (e.g., deployment). The residual differences between the measurements and the estimates form the basis for the model and shift analysis tests which follow.

The model analysis programs characterize the residual differences provided by the filter. In particular, tests are performed to determine if the residuals are a zero-mean white noise process with a variance predicted by the filter model. If these tests are passed, the mathematical model imbedded in the filter is consistent with the true system dynamics. If the "whiteness" tests are failed, further analysis into the nature of the failure are initiated.

Jump detection and identification analysis is used to seek one or more instrument parameter shifts consistent with the filter residual characteristics. The GLR test provides the primary means of instrument coefficient jump detection and identification.

A. DATA PREPROCESSING

A number of important steps must be taken to prepare raw recorded data so that it may be efficiently and accurately analyzed. The software that performs these steps is depicted in Fig. 20.

The PIGA* Prefilter calculates the specific velocity sensed by the primary NS-20 and the indicated velocity difference between NS-20s. Seven categories of deterministic errors are compensated:

- The six PICA pulse sums are adjusted (using the telemetered time-of-last-pulse) to reduce the effects of quantization and sample time differences. Compensations are also made to account for timing differences due to the asynchronous sampling of the PIGAs and due to guidance computer clock drift rate.
- The six PIGA pulses are compensated for errors due to "coning," the result of a misalignment between the PIG float and PIAG input axes. The misalignment angle and phase are calibrated using prelaunch telemetry data.
- The PIGA pulses are subsequently passed through a digital low-pass filter to reduce the residual random errors resulting from quantization and timing uncertainties. The result is the "best estimate" of all six PIGA pulse sums.
- The PIGA pulses are compensated with pre-flight estimates of bias and nonlinearity errors and then transformed into a velocity vector in NS-20 computational coordinates.
- The velocity vector is then corrected for platform-to-computer misalignments. The misalignment is based on the prelaunch values of gyro g^2 and g^4 error coefficients. Initial misalignment, due to gyro torquer limit cycling, is also taken into account.
- Platform compliance errors are compensated next using a 27-term platform bending model.
- Finally, the velocity of the secondary NS-20 relative to the primary NS-20 (lever arm effect) is subtracted from the sensed velocity of the secondary system. This compensation is based upon the telemetered NS-20 gimbal angles which have been interpreted, smoothed, and differentiated.

The results at this point in the NS-20 processing are two measurements of the integrated specific force† (specific velocity), as sensed by the two sets of instruments, compensated for all known deterministic errors. The specific velocity vectors are provided to the trajectory integrator.

The Trajectory Integrator calculates the position and velocity of the missile based on the best available gravity model and the sensed specific velocity. To allow for refinements in the trajectory (as bad data is removed and errors are estimated) without repeating the long integration process required by this program, the total gravity gradient matrix is also calculated.

Radar data is preprocessed using two programs. The Radar Synchronizer performs the function of merging data: range measurement data from up to 10 tracking radars are

*The accelerometers used on Minuteman III are PIGAs or Pendulous Integrating Gyroscopic Accelerometers [11].

†Specific force is the sum of all forces acting on the vehicle except for gravity [11].

extracted from the raw data tapes, put in common engineering units and time synchronized. The result is a single sequential file containing all available radar data.

The Radar Prefilter determines the difference between measured radar range and range-rate and computed range and range-rate based on the NS-20-indicated position and velocity time history generated by the Trajectory Integrator. High-frequency measurement noise is reduced by averaging all range differences over a 4.5 sec time interval. The geometry of each measurement is also determined in order to properly weight the one-dimensional range and range-rate difference measurements in the estimation of three-dimensional position and velocity vectors.

Two types of GPS measurement data are available: preflight data and inflight data. The primary purpose of the preflight processing is to calibrate GPS-related errors, most notably receiver clock errors. These estimates are then applied to the inflight data.

The data tape records contain a mixture of parameter values from receiver-generated "high-rate" and "low-rate" data tables; two separate programs extract the required components from each table. The first reads selected values from the low-rate table and prepares the data for input to the calculation of pseudo-range.* These data items, such as master time delays, user and satellite epoch count differences, etc., are either constant or slowly time-varying quantitites. The required high-rate data table items are accessed by a second program. The items include the replica code counter states, vernier range corrections, range-rates, and status and identification tags for each channel. The program uses this data, along with that supplied from the low-rate table, to form the receiver-to-satellite pseudo-range measurement and also scales the range-rates and corrects for the range-rate computational delay prior to outputting time-tagged, satellite-indexed, corrected pseudo-range and range-rate measurements.

The GPS measurements are compensated for the following calibratable errors:

- Satellite clock errors
- Receiver clock gravity-sensitive trending
- Tropospheric propagation delays
- Relative position and velocity offset between GPS antenna phase center and the primary IMU
- Relativistic effects, both special and general, between GPS receiver and ground-based user.

A final preprocessing program determines the differences between the measured and computed range and range-rate using the receiver- and satellite-indicated position and velocity. The satellite state vector is computed at each time of signal transmission using the best estimate of the "Block II" ephemeris data provided by the GPS Joint Program Office [15]. The receiver position and velocity are determined from the best estimate trajectory interpolated to the time of signal reception. The resulting difference measurements are then compressed to suppress measurement noise effects. These range and range-rate differences, along with the computed measurement geometry, are the final preprocessor outputs.

B. FILTER ANALYSIS

The filter calculations use the sequence of range and range-rate differences (GPS and/or radar) and velocity differences (between NS-20s) to calculate the best estimate of NS-20 guidance coefficients, GPS errors, and radar errors based on *a priori* error statistics. The residual differences between the measurements and estimates are the basic inputs to the GLR tests. Data flow is shown in Fig. 21. The residual calculations are divided into two separate sets of programs for computational efficiency. The first, Gain Calculation, requires the straight-forward, although lengthy, computation of Kalman gains based on the nominal trajectory, the system error model and the measurement sequence.

Two forms of smoother may be used. The Fixed-Point Smoother [16] allows the computationally efficient estimation of a limited number of smoothed states at selected times. This is particularly attractive if only certain candidate states are suspected to be time-varying. The Fixed-Interval Smoother [6] allows the smoothed estimation of all parameters but requires more computations. Both have a place in the search for unmodeled parameter changes.

The second program set, Residual Calculator, uses the gains to interpret the measurements from the data preprocessor. During preliminary data editing, the relatively simple residual calculations can be performed many times, using the same set of gains, without significant loss of accuracy. The more lengthy gain calculations need be repeated only after "bad" data has been removed or whenever the filter model is changed.

*Pseudo-range measurements contain "true" slant range plus the receiver clock phase offset.

C. MODEL ANALYSIS AND JUMP DETECTION/IDENTIFICATION

The primary jump detection is performed using the Generalized Likelihood Ratio (GLR) test. The Kalman filter residuals are used in the GLR test. The GLR test computations are divided into two routines: the first calculates the GLR gain matrices, the second performs the GLR test on the data. (The GLR gain matrices need only to be recomputed when the filter gain matrices are recomputed.) The GLR Test routines use the gain matrices to determine whether the filter residuals are consistent with the model. In the event of a jump detection, the time and identity of the parameter(s) which changed (jumped) are estimated. These jump estimates may then be used to change the model and the residual calculations may be repeated.

Residual Tests are performed on the filter and smoother outputs. If the models are correct, the residuals will be a zero-mean, uncorrelated random sequence and with variance as predicted by the model. Thus, the sample mean and variance calculations provide some clues as to the nature of the modeling errors. The residual tests provide a "quick-look" capability to identify missions with possible parameter jumps.

VI. DATA PROCESSING EVALUATION RESULTS

Data processing included evaluation of data from the three FLY-2 flights and the first FLY-2/GPS mission. Here, the processing results from one FLY-2 flight (STM-13W) and the FLY-2/GPS flight (PVM-18) are highlighted. Table X summarizes the principal characteristics of each flight.

The STM-13W data analysis focused on jump detection and identification. A quick-look technique for determining the possibility of PIGA and/or gyro error coefficient shifts was developed based on the velocity difference data generated by the dual IMUs. Figure 22 is a plot of sensed velocity differences over the first 1500 sec of powered flight. Three distinct phases are evident. During the first or powered flight phase (0 to 180 sec), uncompensated acceleration-dependent errors cause parabolic error growth. Over the second phase (180 sec to 500 sec) no error growth is evident since the vehicle is experiencing a nearly zero specific force. The third phase (after 500 sec) provides a clear indication of a PIGA bias shift.

The PIGA bias shift is easily detected by the GLR algorithm. However, in order to identify a gyro error (e.g., bias drift) coefficient shift during the powered flight phase, a closer look at the data over the first interval is required. Rather than velocity differences, it is more enlightening to examine platform-to-computer misalignment angles, $\underline{\psi}$. These angles can be obtained by recalling that, over a very short time interval Δt,

$$\int_{t_o}^{t_o + \Delta t} (\underline{\psi} \times \underline{f})\, dt = \underline{\Delta v} \tag{4}$$

The $\underline{\Delta v}$ are the differences in measured velocity. If $\underline{\psi}$ is assumed constant over the interval Δt,

$$\underline{\psi} \times \int_{t_o}^{t_o + \Delta t} (\underline{f})dt = \underline{\Delta v} = - [A]\, \underline{\psi} \tag{5}$$

where [A] is the skew-symmetric matrix of integrated specific force components. As a consequence

$$\underline{\psi} = - [A^{-1}]\, \underline{\Delta v} \tag{6}$$

Figure 23 presents the estimates of the three components of $\underline{\psi}$ in computer axes (down-range, cross-track, and up). Straight-line approximations to the curves correspond to the assumption that the gyro drift is caused by a bias only possibly with a jump change in magnitude. The curves in Fig. 23 indicate the possibility of gyro bias error breaks occurring at approximately 60 and 135 sec. However, the shifts are not large and may be accounted for by the NS-20 error model.

It was anticipated that a PIGA bias shift would be evident in the GLR output and perhaps one or more gyro bias shifts would be detected. It should be recalled that the GLR test answers the question: what is the relative likelihood that a given guidance coefficient experienced a shift at a given time, compared to the null hypothesis (no shift)? The likelihood ratio (ℓ) is the quantity used to quantify the alternative hypotheses. Large ℓ implies a significant jump in the parameter of interest. Values of $\ell < 10$ are not significant as indicators of a jump at the $P_F = 0.05$ level. Results were obtained using both the Fixed-Lag GLR algorithm with a 100 sec data window, and from the Fixed-Interval GLR algorithm incorporating data up to 600 sec into the flight. For brevity, only Fixed-Lag results are presented herein.

The Fixed-Lag GLR results (Fig. 24) display a weak PIGA bias anomaly at 140 sec which falls well below the $\ell = 10$ threshold. Consequently, this weak anomaly was not deemed to be significant. A strong anomaly is evident in the post-boost phase. While this anomaly has minimal impact on system accuracy (because of its time of occurrence), it was evaluated in detail. The conclusion of the analysis is that all three PIGAs in the primary system experience shifts in their bias level near 450 sec.

Fixed-Lag gyro bias drift likelihood ratios demonstrate a false response due to the large PIGA bias shifts (Fig. 25). Only two of the three gyro bias drift states are presented; the third drift state is nearly unobservable and consequently of no interest. Indepth analysis of the telemetry and radar data indicates that no detectable gyro bias drift anomalies occurred during the boost phase.

Results of the STM-13W data analysis are typical of those for each FLY-2 flight. The presence of a large PIGA bias shift near the 450 sec time print was evident on all the flights. In addition, there was no indication of a detectable PIGA or gyro anomaly during boost.

It was concluded that anomalous guidance system performance, in the form of shifts in instrument parameters, is not significant statistically or in terms of weapon system accuracy. Thus, for the FLY-2/GPS data processing activity, the focus shifted to guidance coefficient estimation and GPS receiver performance assessment.

For the PVM-18 mission, four GPS satellites provided range and range-rate measurements. The subsatellite points for each satellite are indicated in Fig. 26. The signals from the satellites were acquired approximately five sec after launch and, except during staging events, provided accurate information out to approximately 1000 sec. Satellites No. 1 and No. 2 yield very good cross-range information, while Satellites No. 3 and No. 4 contribute primarily to down-range information. Overall, GDOP (Geometric Dilution of Precision [5]) was approximately 4-5 over the flight, with the vertical channel having the poorest GDOP. The cross-track axis had a single-channel GDOP less than 1.0. Values of azimuth, as measured from north, and elevation above the local-horizontal plane of the satellite relative to the receiver are summarized in Table XI.

The primary-minus-secondary IMU velocity differences for PVM-18 are presented in Fig. 27. The three phases of error growth (i.e., powered flight, free flight, and post PIGA bias shift) are evident.

The velocity differences are quite small through the first 125 sec of flight (the first and second stages of missile thrusting). A rapid divergence occurs during third-stage thrusting, leveling off in all three axes at thrust termination. The nature of the third-stage divergence, occurring most noticeably in the cross-track direction and with an obvious thrust dependence, is indicative of large residual gyro errors (or possibly PIGA misalignments) in one or both systems. Cross-track sensitivity to gyro errors is extremely high in Minuteman due to the exclusively in-plane (x-z plane) thrusting pattern utilized. The sensitivity to errors about azimuth is further accentuated by the particular trajectory flown.

Range and range-rate difference measurements along the Satellite No. 1 line-of-sight are presented in Figs. 28 and 29, respectively. The dotted line represents the high frequency measurement data and the solid line represents the "smooth" difference measurements used in the subsequent data analysis. These are typical of the range and range-rate measurements provided by all four satellites. (The large spikes in the range-rate measurements occur during periods when the tracking algorithm is in frequency track only.)

The continuous GPS measurement availability and excellent measurement quality made it possible to improve the post-mission analysis results. The S_{A1} and F_{S1} gyro errors were identified as the primary sources of the observed impact error for PVM-18. This conclusion is based on the fact that the post-flight analysis is in excellent agreement with preflight predictions of the miss-distance contributions of the S_{A1} and F_{S1} error sources. However, even with the excellent GPS measurement quality, the inherent capability for accurately separating the individual error contributions of the S_{A1} and F_{S1} instrument errors is still in question. This separability problem was demonstrated based on results of a sensitivity study performed as part of the PVM-18 analysis. This study examined the sensitivity of the estimated impact error contribution of critical gyro error sources to their initial rms uncertainties. Results of this study show significant variations in the estimated miss distance resulting from the individual gyro error sources, while total impact error displays minimal net variation. This behavior is indicative of a basic inability to isolate individual gyro error source contributors.

Post-flight evaluation of Minuteman performance based on GPS test measurements indicates that GPS is an excellent absolute reference for both position and velocity. Estimation of initial platform misalignment is also improved. In addition, qualitative insight into PIGA performance can also be obtained through examination of IMU/GPS velocity differences. A performance issue that impacts future utilization of GPS must still be addressed, however. This issue is the proper use and interpretation of data that is extremely accurate at the position and velocity levels but, due to the complexity of the underlying guidance error model, does not provide unique insight into the magnitudes of specific instrument-related error mechanisms. In other words, the model observability problem must be addressed as it becomes necessary to work to finer levels of detail in generating system understanding. The capabilities of filtering and smoothing analysis offer a great deal in evaluation of complex weapon systems but only to a certain threshold which must be identified.

VII. SUMMARY

Over the past 10 years the U.S. Air Force has upgraded the accuracy of the Minuteman III. To assess and further these accuracy upgrades, a number of flight test program instrumentation enhancements were incorporated. These include post-flight processing of multiple IMU test data (the MPMS and FLY-2 Programs), and use of GPS data (the FLY-2/GPS Program). This series of flight test programs was planned and executed in a logical manner to minimize cost/schedule impacts.

The Minuteman III flight test program enhancements have been successful. A number of significant error mechanisms were identified and isolated using data obtained during the MPMS, FLY-2, and FLY-2/GPS Programs. With the improved accuracy objectives associated with the next generation ICBM system, problems of flight test optimization and post-mission processing will continue to provide challenging opportunities over the years ahead. There will almost assuredly be further advances in filtering and smoothing theory to support the needed growth in system understanding.

REFERENCES

1. Duiven, E.M., et al, "Detection of Anomalous Guidance System Performance Using FLY-2 and GPS Measurements," Proc. of the 8th Biennial Guidance Test Symposium (Holloman Air Force Base), October 1979.

2. Fussell, R., et al, "A Method for Determining the Performance of a Precision Inertial Guidance System," Proc. of the 1979 AIAA Guidance and Control Conference (Boulder, Colorado), August 1979.

3. Thompson, T., "Performance of the SATRACK/GPS Trident I Missile Tracking System," Proc. of the 1980 Position Location and Navigation Symposium, (Atlantic City, New Jersey) December 1980.

4. "AIRS Description Document," Honeywell, Inc., Report No. 0972-11167-A, 13 February 1973.

5. "Global Positioning System," Journal of Navigation, Vol. 25, No. 2, Summer 1978.

6. Gelb, A., ed., Applied Optimal Estimation, M.I.T. Press, Cambridge, Massachusetts, 1974.

7. Leondes, C.T., ed., Theory and Application of Kalman Filtering, NATO AGARDograph No. 139, Bradford House, London, England, 1970.

8. Willsky, A.S. and Jones, H.L., "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems," IEEE Trans. Automa. Contr., Vol. AC-21, No. 2, pp. 108-112, February 1976.

9. Willsky, A.S., Deyst, J.J., and Crawford, B.S., "Adaptive Filtering and Self-Test Methods for Failure Detection and Compensation," Joint Automatic Control Conference (Austin, Texas), June 1970.

10 [illegible] E.M. "Suboptimal Linear Filtering," AIAA Journal, Vol. 11, No. 3, March [illegible] pp. 196-198.

11. Pitman, G.R., ed., Inertial Guidance, John Wiley & Sons, Inc., New York, New York, 1962.

12. Hutchinson, C.E., and Nash, R.A., "Comparison of Error Propagation in Local Level and Space Stable Systems," IEEE Trans. Aero. Elect. Systems, Vol. AES-7, No. 6, pp. 1138-1142, November 1971.

13. Leondes, C.T., ed., Guidance and Control of Aerospace Vehicles, McGraw-Hill, New York, 1963.

14. Britting, K.R., Inertial Navigation Systems Analysis, John Wiley & Sons, Inc., New York, New York, 1971.

15. "Space Vehicle Navigation Subsystem and NTS PRN Navigation Assembly/User System Segment and Moniter Station," GPS Joint Project Office, Report No. MH08-00002-400, Revision 6, October 1979.

16. Biswas, K.K., and Mahalanabis, A.K., "An Approach to Fixed-Point Smoothing Problems," IEEE Trans. Aero. Elect. Systems, Vol. AES-8, No. 5, pp. 676-682, September 1972.

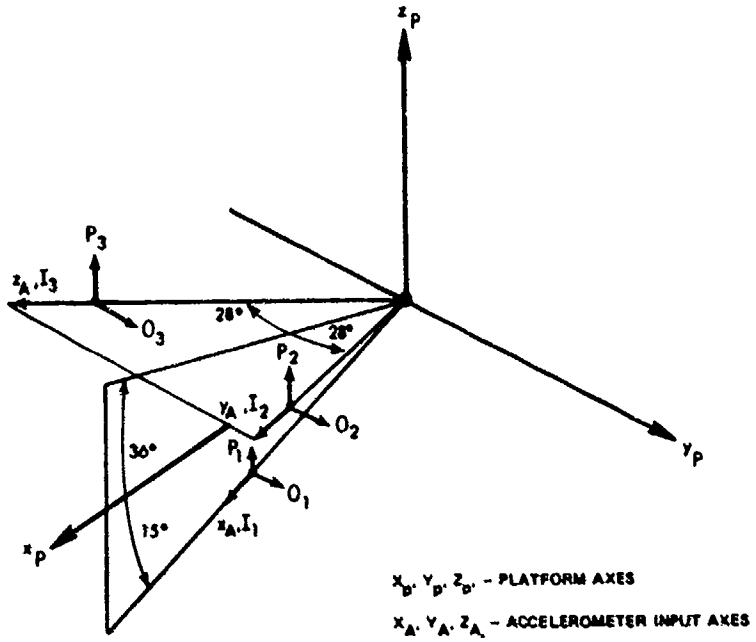


Figure 1 Accelerometer Input Axes

R-20116

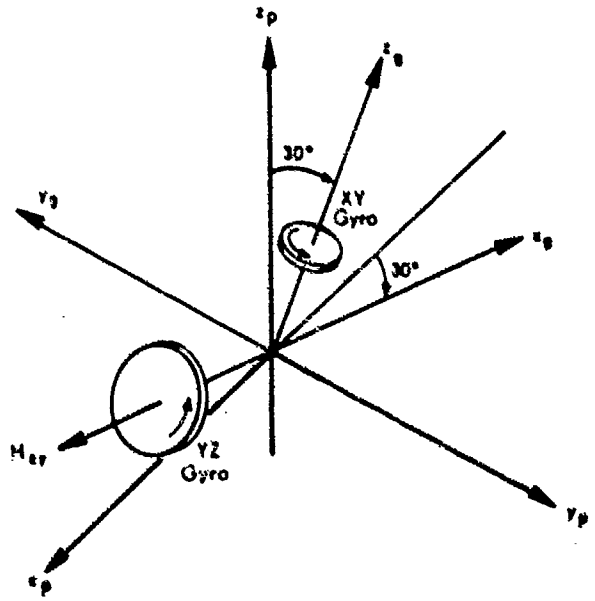
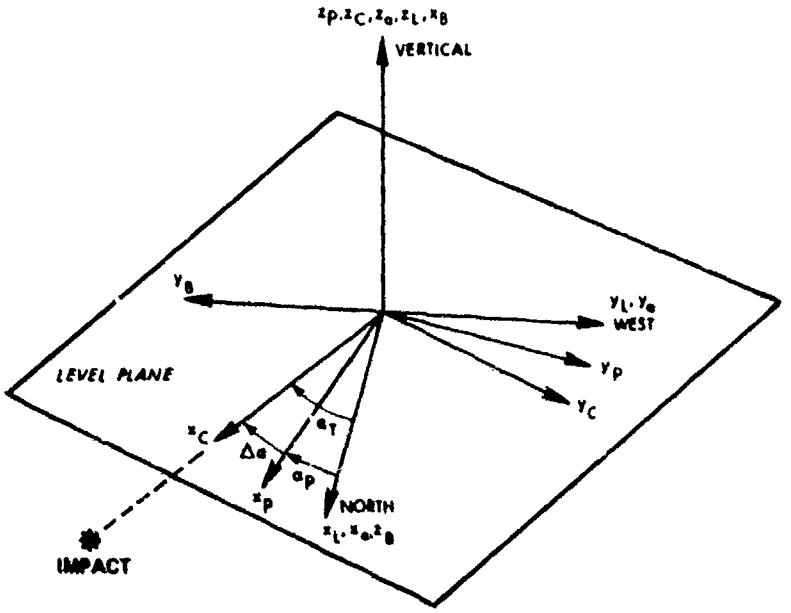


Figure 2 Gyro Input Axes



$x_L\ (x_L, y_L, z_L)$ = LAUNCH-SITE FRAME
$x_c\ (x_c, y_c, z_c)$ = COMPUTER FRAME
$x_p\ (x_p, y_p, z_p)$ = PLATFORM FRAME
$x_a\ (x_a, y_a, z_a)$ = GYRO COMPASS FRAME
$x_B\ (x_B, y_B, z_B)$ = MISSILE BODY FRAME

Figure 3 Inertial Measurement Unit Frames

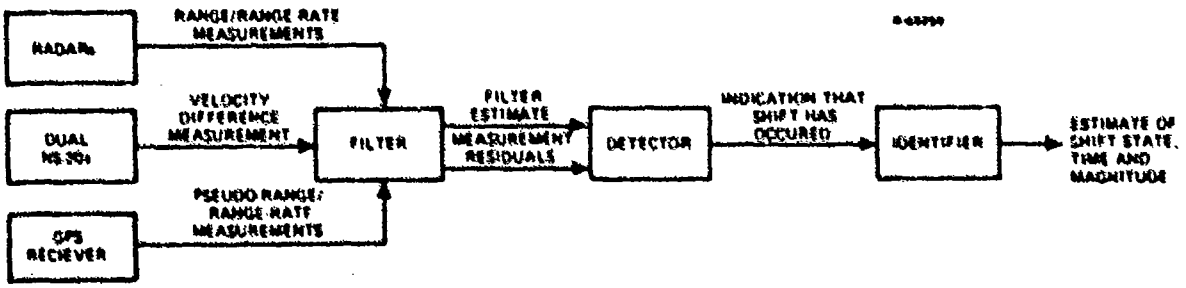


Figure 4 Post-Flight Evaluation Software Data Flow

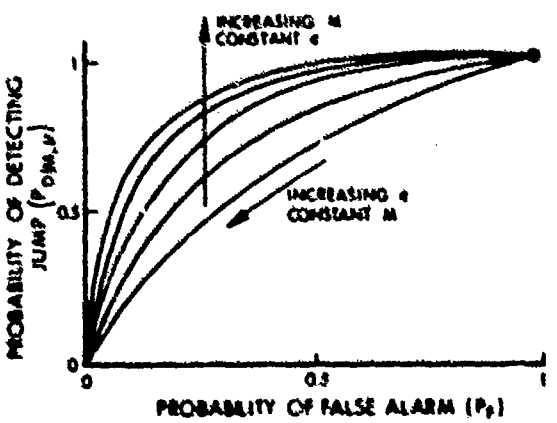


Figure 5 General Relationship Between P_D and P_F as a Function of M and ε

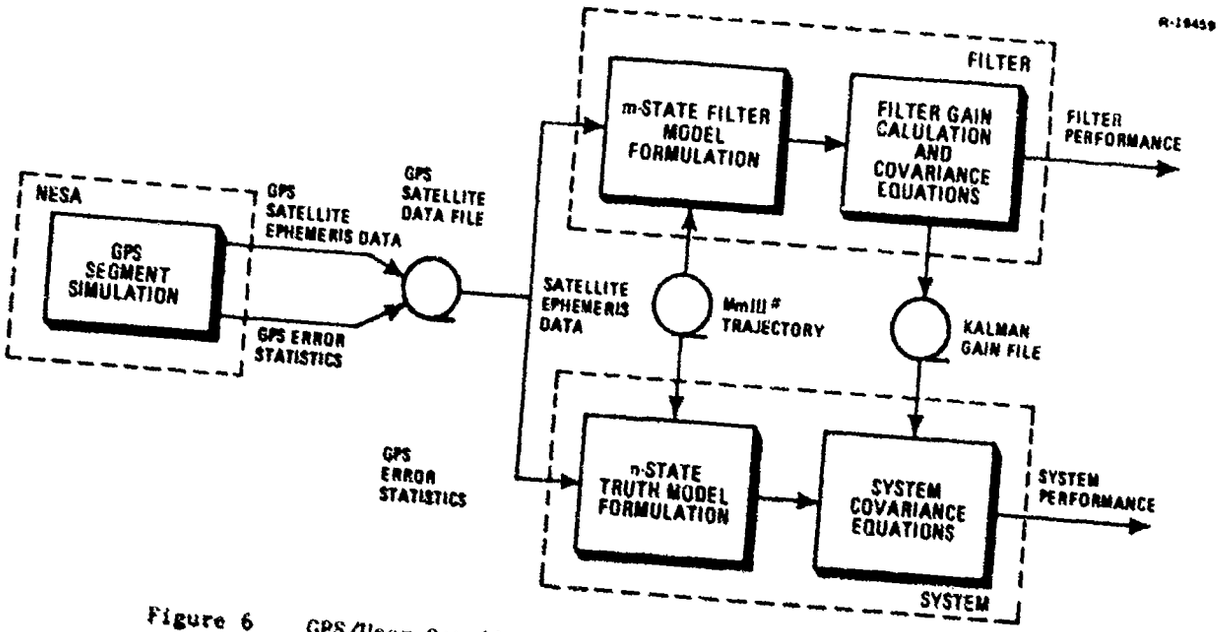


Figure 6 GPS/User Satellite Performance Projection Methodology

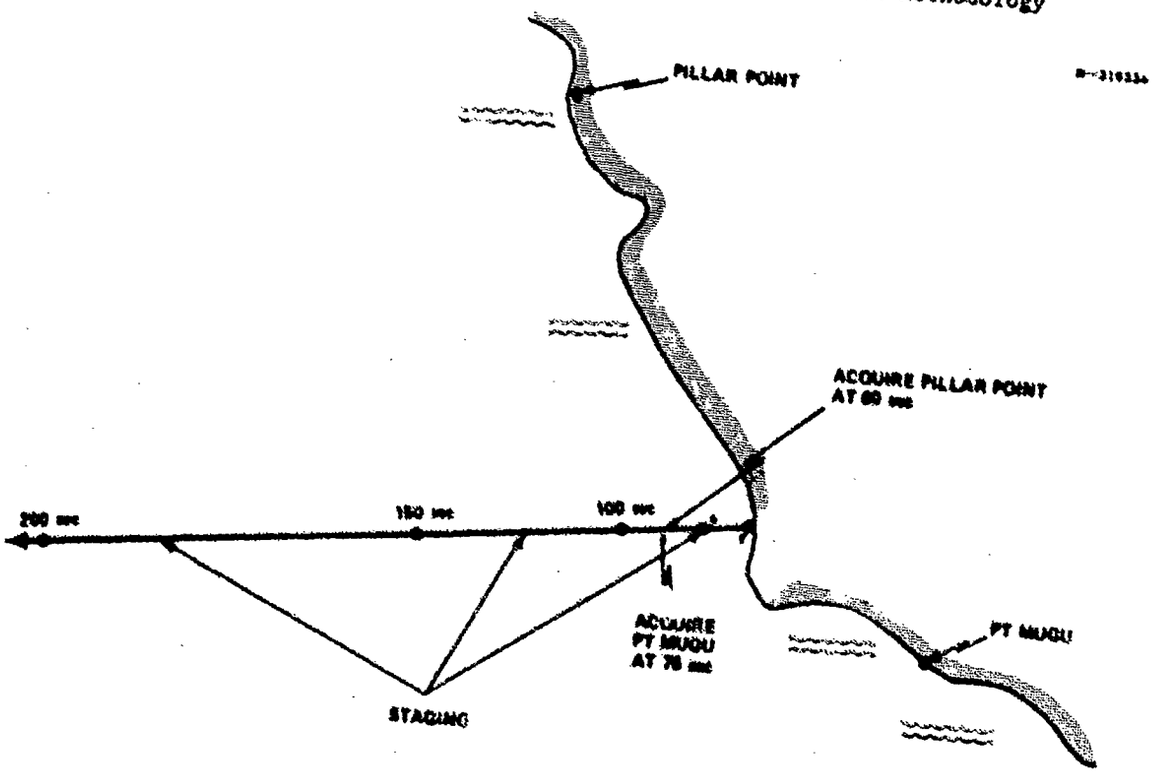


Figure 7 Typical Minuteman III Test Trajectory

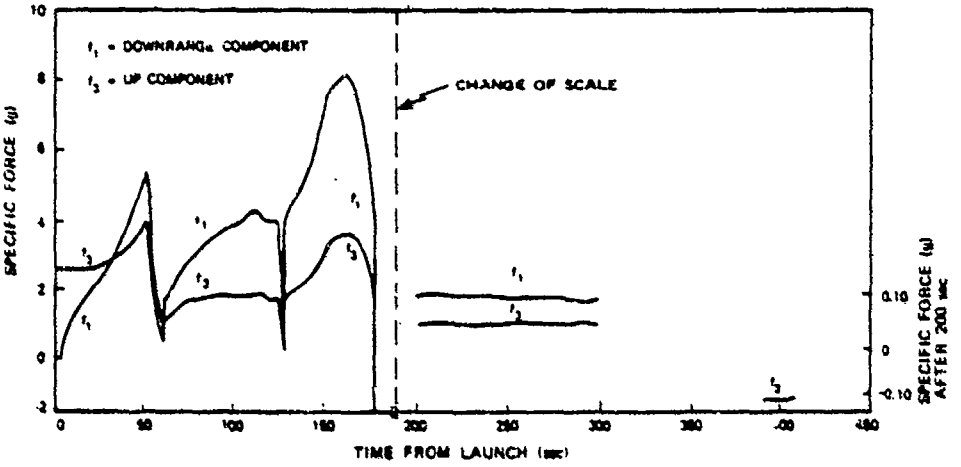


Figure 8 Typical Specific Force Time History

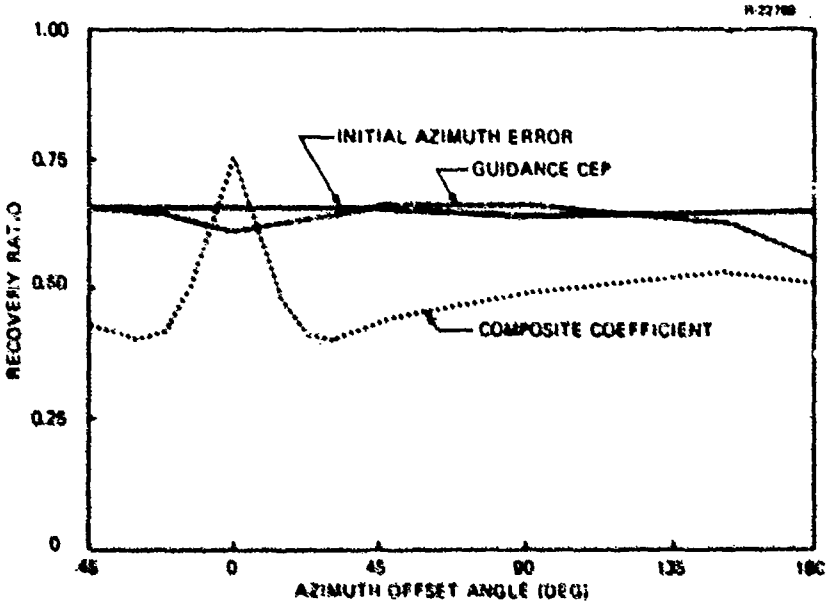


Figure 9 Recovery Ratios as a Function of Secondary INU Azimuth Offset (Low Reentry Angle)

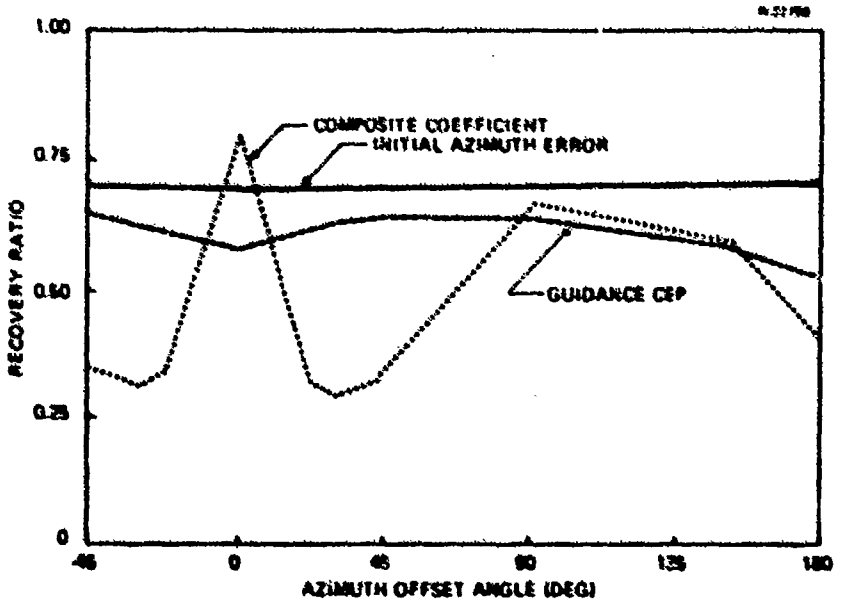


Figure 10 Recovery Ratios as a Function of Secondary INU Azimuth Offset Angles (Medium Reentry Angle)

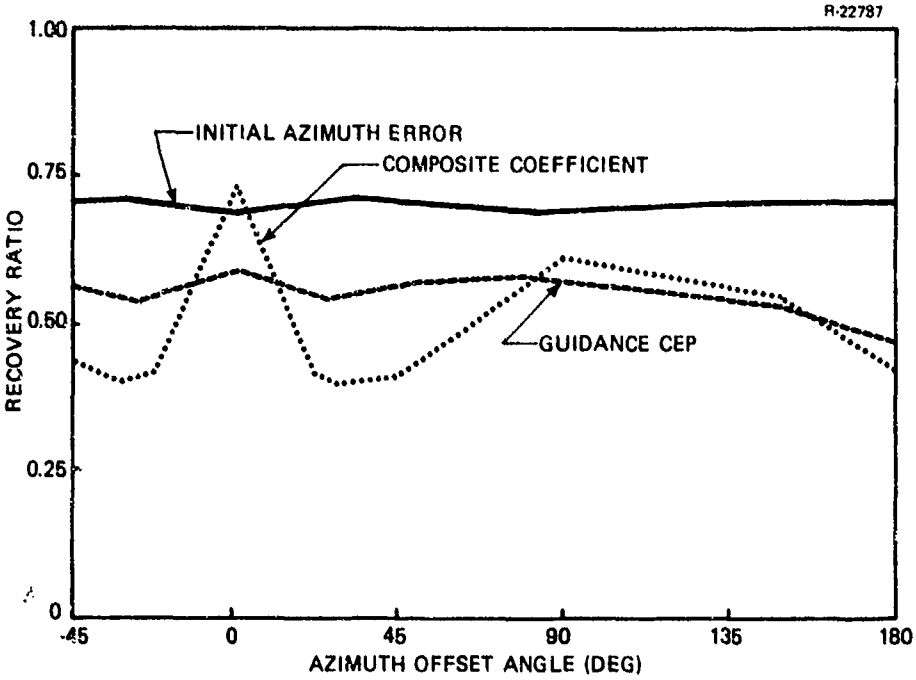


Figure 11 Recovery Ratios as a Function of Secondary IMU Azimuth Offset Angle (High Reentry Angle)

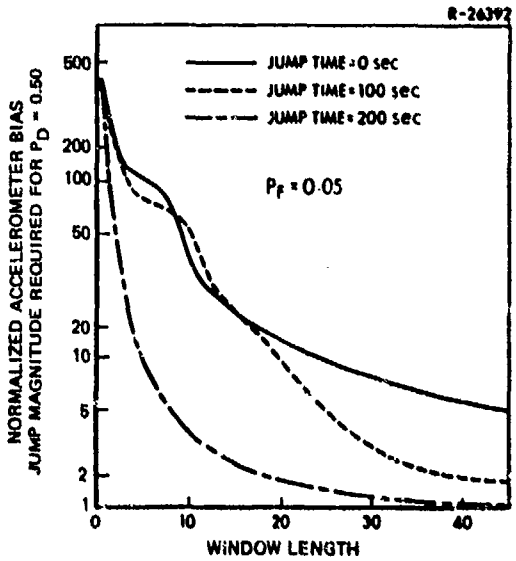


Figure 12 Accelerometer Bias Jump Detection vs Window Length Using Fixed-Lag GLR

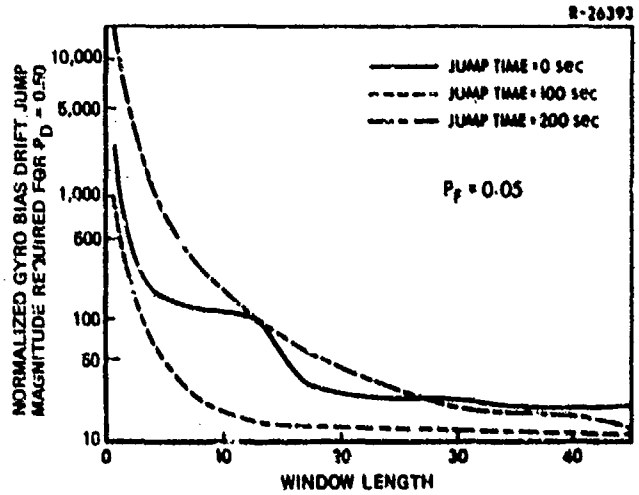


Figure 13 Gyro Bias Drift Jump Detection vs Window Length Using Fixed-Lag GLR

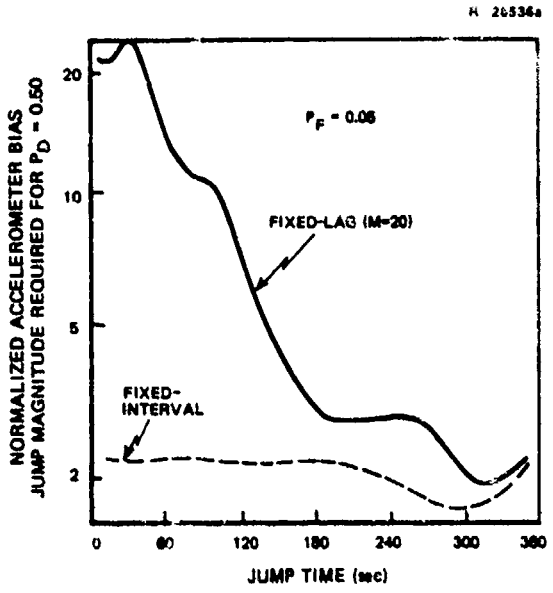


Figure 14 Accelerometer Bias Jump Detection vs Jump Time

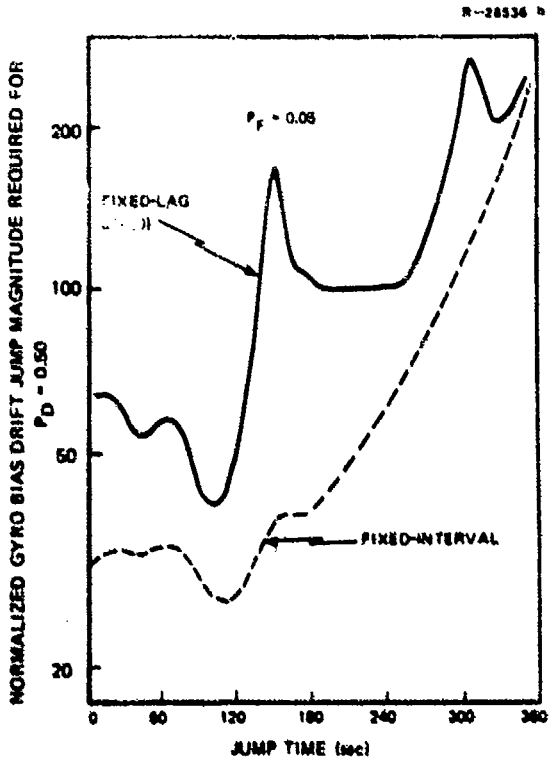


Figure 15 Gyro Bias Drift Jump Detection vs Jump Time

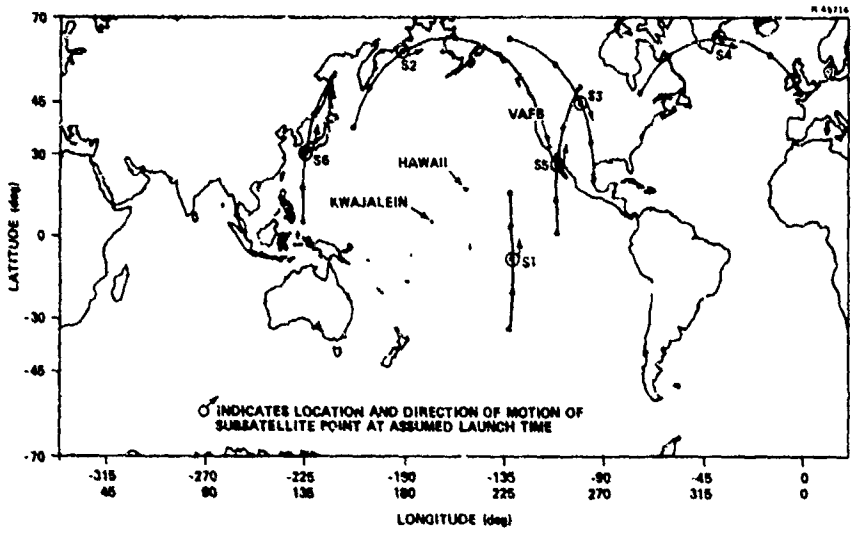


Figure 16 Phase I GPS Satellite Configuration

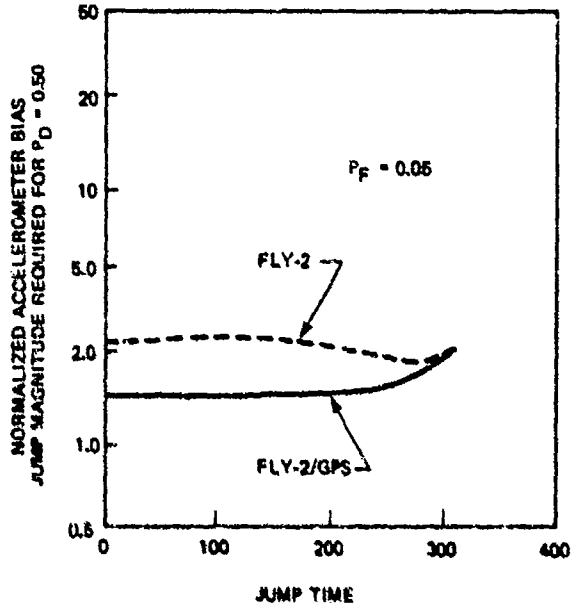


Figure 17 Normalized Accelerometer Bias Jump Detection vs Jump Time

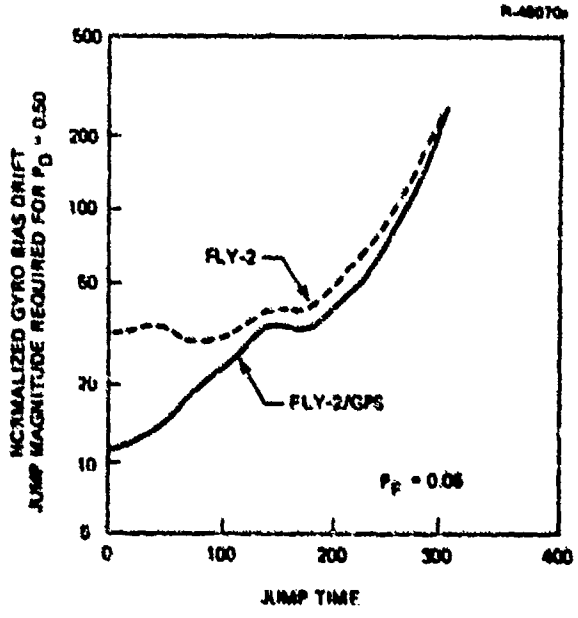


Figure 18 Normalized Gyro Bias Drift Jump Detection vs Jump Time

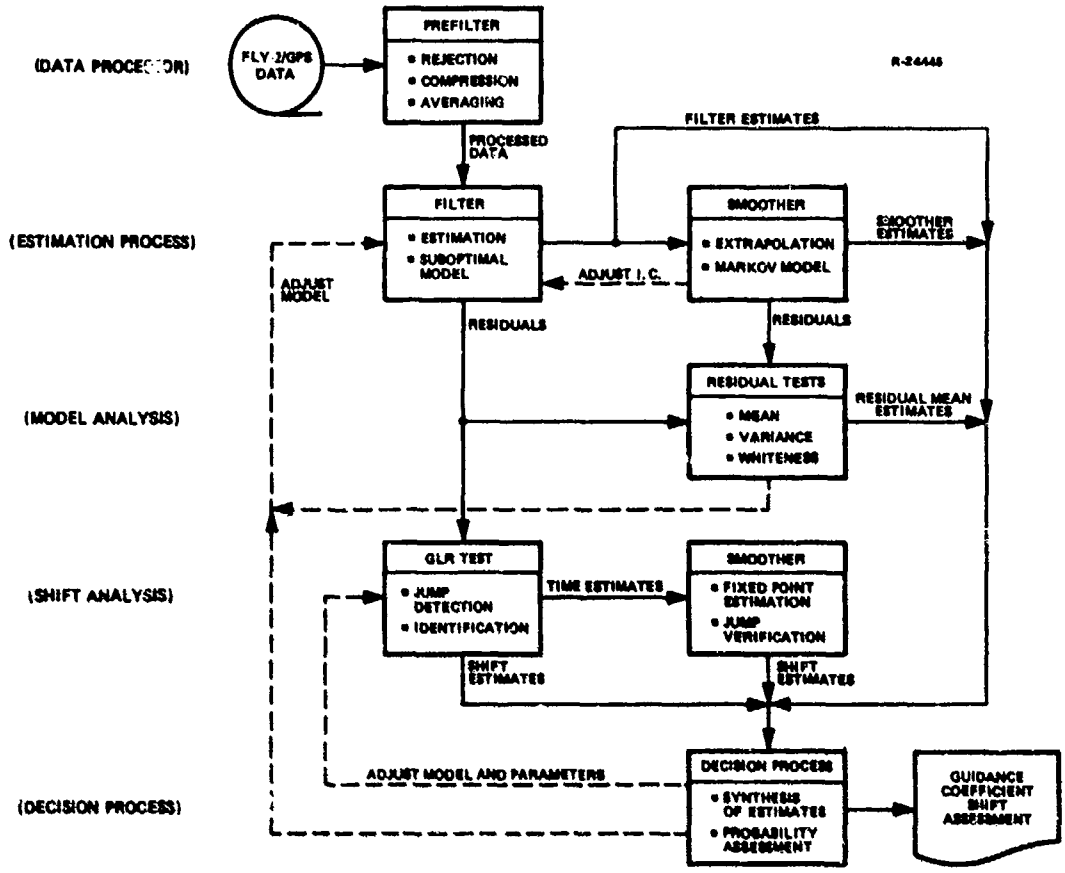


Figure 19 Post Flight Analyzer Data Flow

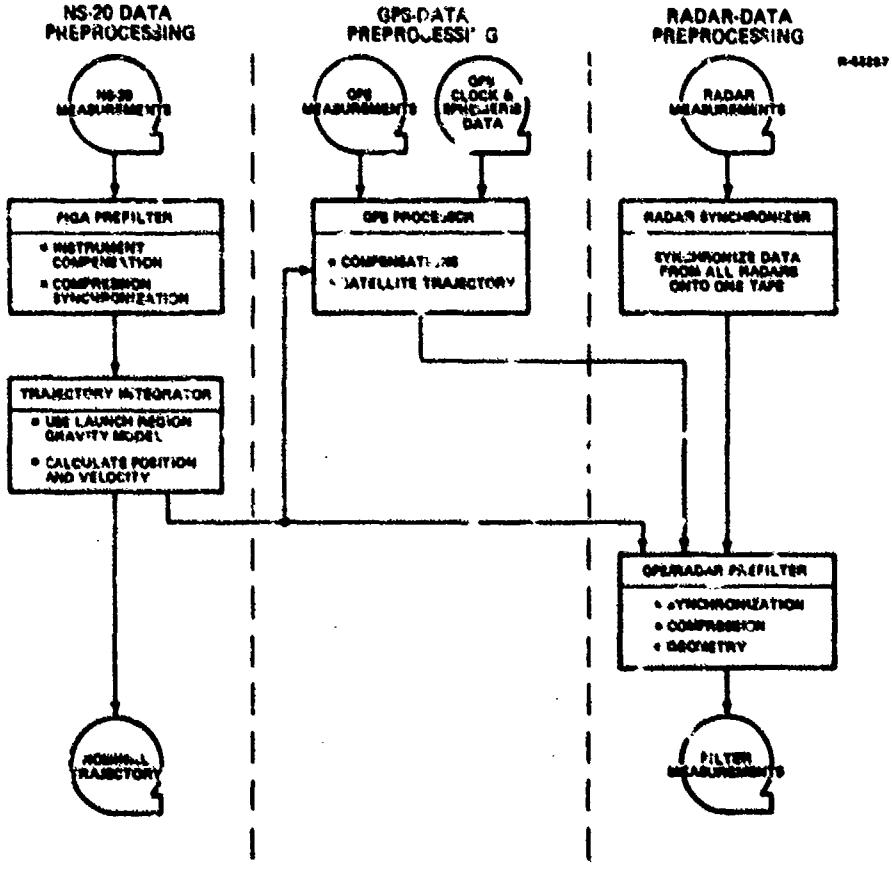


Figure 20 Data Flow for Preprocessing Steps

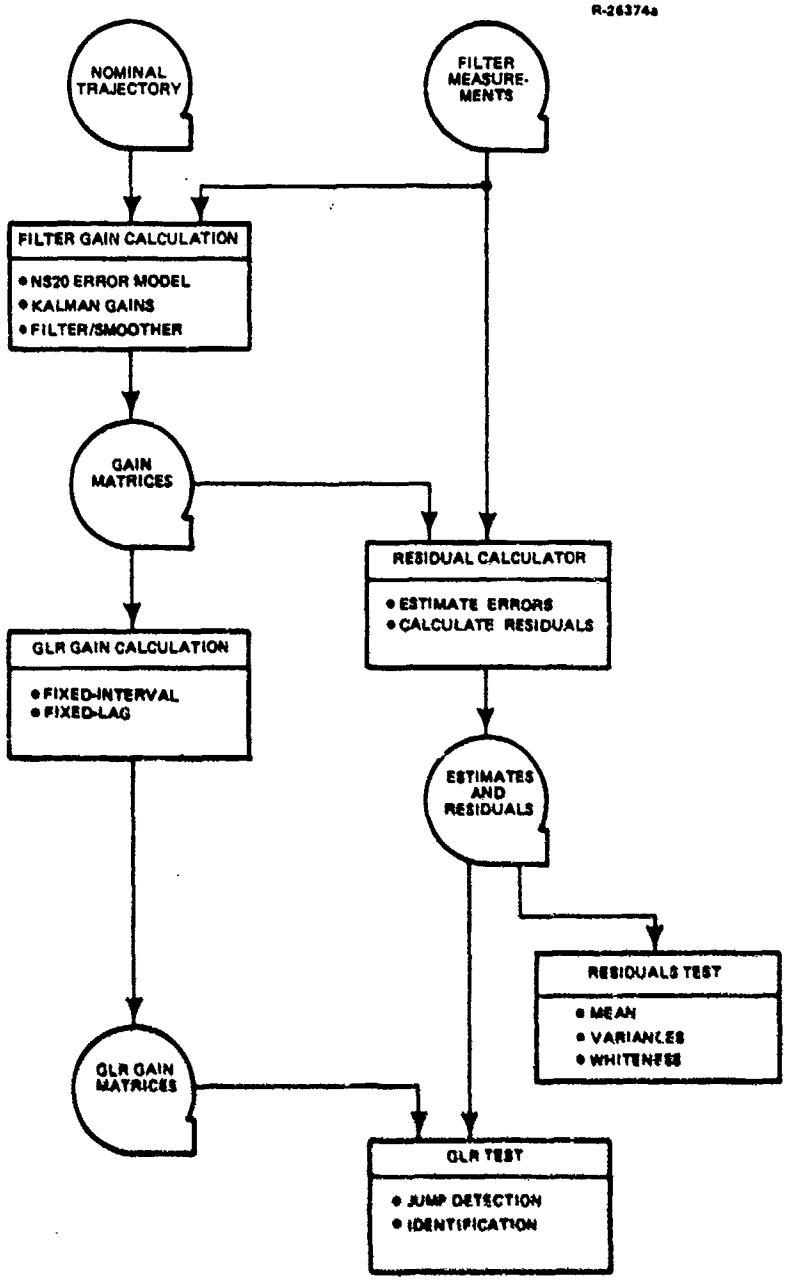


Figure 21 Data Flow for Residual Calculation and Evaluation

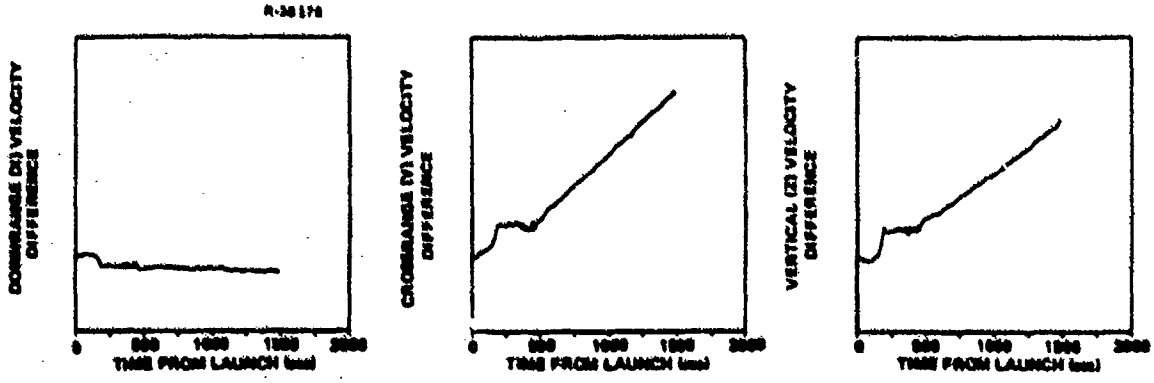


Figure 22 Sensed Velocity Comparison (Primary Minus Secondary in Computer Coordinates)

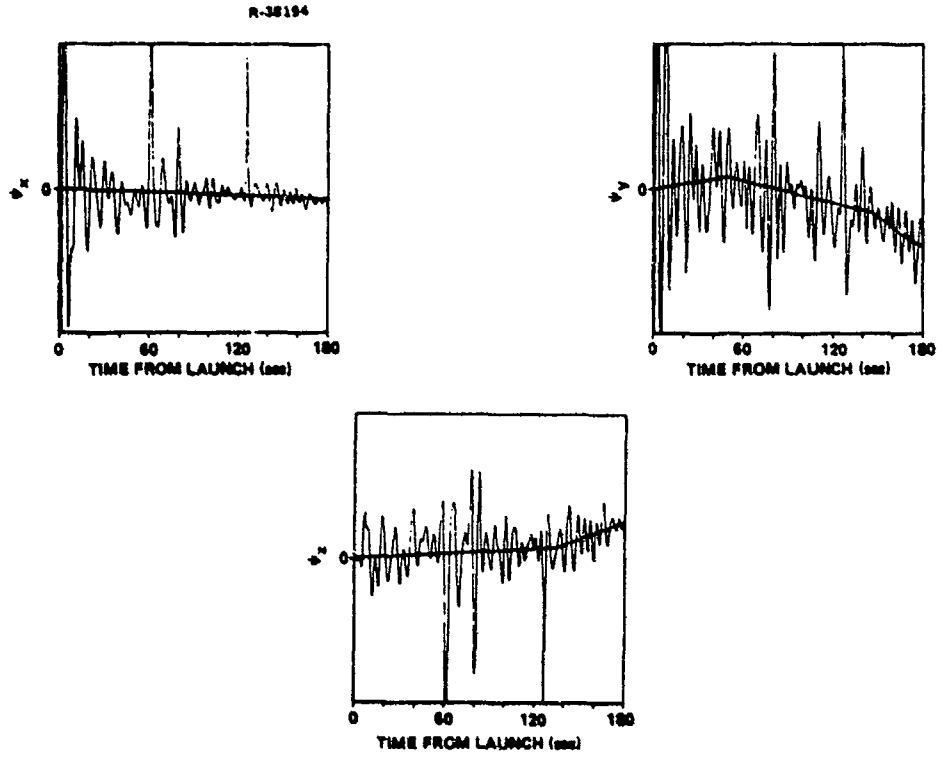


Figure 23 ψ-Angle Estimates

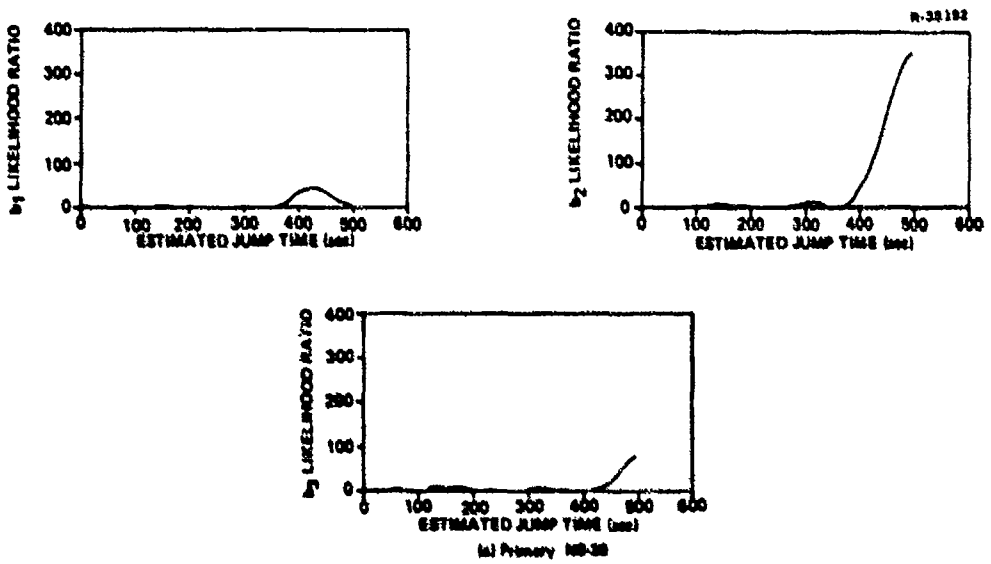


Figure 24 Fixed-Lag Likelihood Ratios for Primary System PIGA Bias Shifts

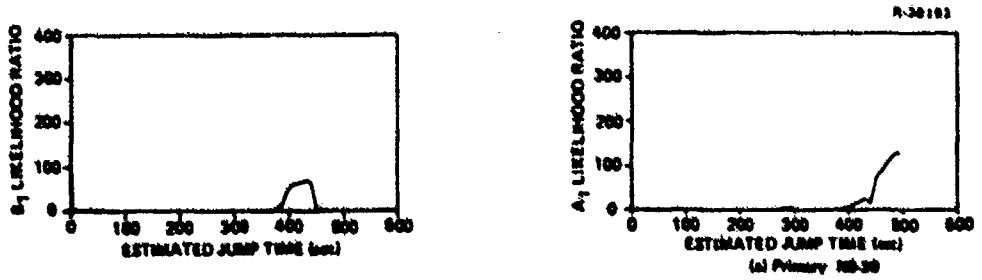


Figure 25 Fixed-Lag Likelihood Ratios for Primary Gyro Bias Drift (False Indication Resulting From Large PIGA Bias Shift at 450 sec)

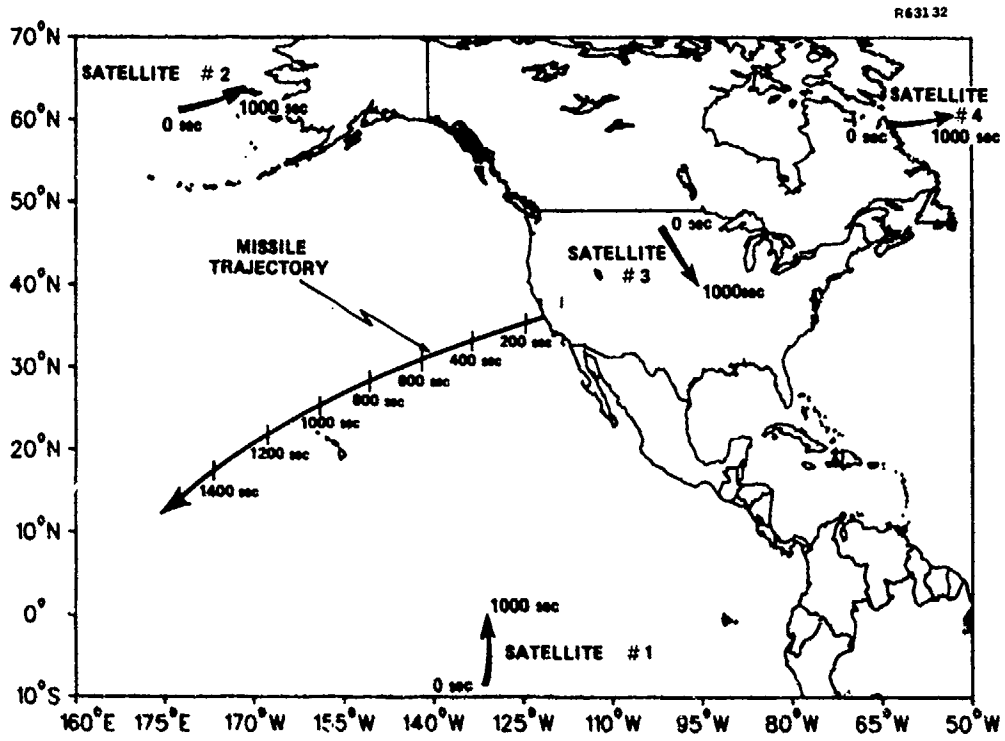


Figure 26 GPS Satellite Orbits During PVM-18 Flight Test

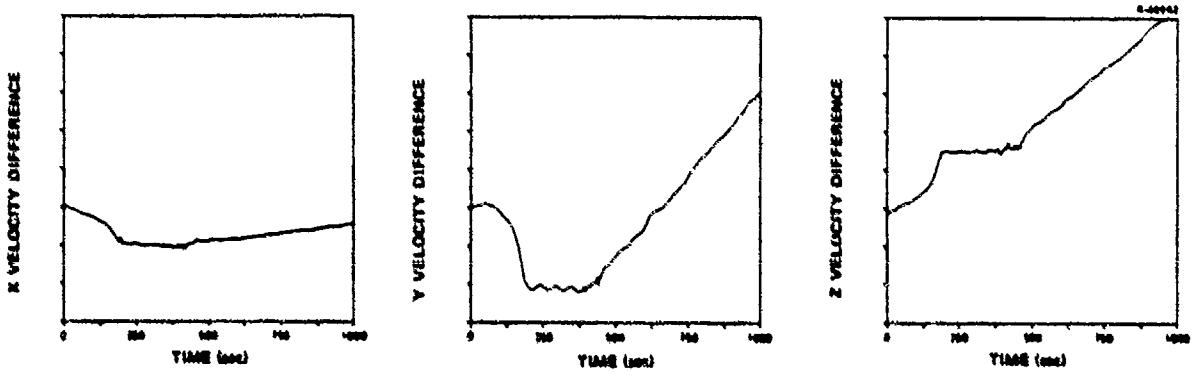


Figure 27 Primary Minus Secondary IMU Velocity Differences (Computer Coordinates)

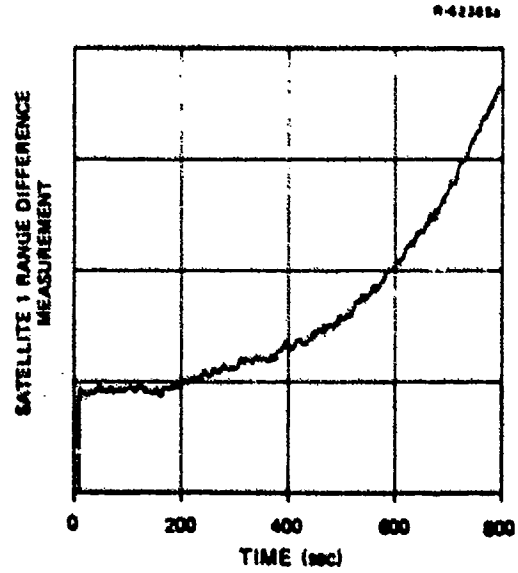


Figure 28 GPS-Primary Satellite 1 Range Difference Measurements

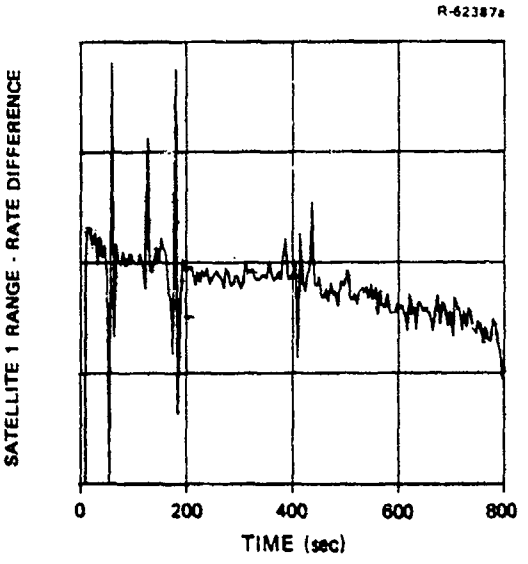


Figure 29 GPS-Primary Satellite 1 Range-Rate Difference Measurements

TABLE I

IMU ERROR MODEL SUMMARY

T-3541

ERROR OR ERROR SOURCE NAME	NUMBER OF STATES		
	FULL MODEL	TRUTH MODEL	FILTER MODEL
PRIMARY			
Position Errors	3	3	3
Velocity Errors	3	3	3
Alignment Errors	3	3	3
Initial Alignment Errors	3	3	3
PRIMARY - NEW			
Differential Position Errors	3	3	3
Differential Velocity Errors	3	3	3
Differential Alignment Errors	3	3	3
Differential Initial Alignment Errors	3	3	3
PRIMARY IMU INSTRUMENT ERRORS SOURCES			
Accelerometers			
Uncorrelated Bias	3	3	3
Scale Factor	3	-	-
Input g^2 Nonlinearity	3	-	-
Input g^3 Nonlinearity	3	-	-
Input Axis Misalignments	6	-	-
Cross-Axis Nonlinearity	3	-	-
Q-Matrix Calibration Errors	9	9	9
Platform Compliance Errors	27	-	-
Gyros			
Bias	3	3	3
Mass Unbalance	6	2	2
Anisoelasticity	6	3	3
Gyro g^4 Coefficients	8	3	3
Temperature Dependent Drift	3	-	-
NEW IMU INSTRUMENT ERROR SOURCES			
Accelerometers			
Uncorrelated Bias	3	3	3
Scale Factor	3	-	-
Input g^2 Nonlinearity	3	-	-
Input g^3 Nonlinearity	3	-	-
Input Axis Misalignments	6	-	-
Cross-Axis Nonlinearity	3	-	-
Q-Matrix Calibration Errors	9	9	9
Platform Compliance Errors	27	-	-
Gyros			
Bias	3	3	3
Mass Unbalance	6	2	2
Anisoelasticity	6	3	3
Gyro g^4 Coefficients	8	3	3
Temperature Dependent Drift	3	-	-
TOTAL NUMBER OF STATES	186	70	70

TABLE II

RADAR ERROR MODEL SUMMARY

ERROR SOURCE NAME	NUMBER OF STATES		
	FULL MODEL	TRUTH MODEL	FILTER MODEL
RANGE MEASUREMENTS			
Bias Error	1	1	1
Scale Factor Error	1	1	1
Random Error	1	1	1
Measurement Noise	1	1	1
Survey Errors	0	0	0
RANGE-RATE MEASUREMENTS			
Bias Error	1	1	1
Scale Factor	1	1	1
Random Error	1	1	1
Measurement Noise	1	1	1
Survey Errors	0	0	0

TABLE III

GPS ERROR MODEL SUMMARY

ERROR SOURCE NAME	NUMBER OF STATES		
	FULL MODEL	TRUTH MODEL	FILTER MODEL
SATELLITE ERRORS			
Position	3	3	0
Velocity	3	3	0
Solar Radiation Force	1	1	0
Gravitation Constant	1	1	0
Satellite Clock	3	3	0
PROPAGATION ERRORS	2	0	0
RECEIVER ERRORS			
Missile Clock	5	3	3
Carrier and Code Loop	2	0	0

TABLE IV

FLY-1 GUIDANCE ERROR RECOVERY RATIOS AT BOOST BURN-OUT

REENTRY ANGLE	DOWN-RANGE POSITION ERROR	CROSS-TRACK POSITION ERROR	DOWN-RANGE VELOCITY ERROR	CROSS-TRACK VELOCITY ERROR	PREDICTED CROSS-TRACK MISS	PREDICTED CEP
Low	0.98	0.56	0.99	0.63	0.63	0.82
Medium	0.99	0.53	0.99	0.56	0.56	0.76
High	0.99	0.55	0.99	0.62	0.63	0.79

TABLE V

BEST FLY-1 PRIORITY ERROR SOURCE COEFFICIENT RECOVERY RATIOS

ACCELEROMETER COEFFICIENTS			GYRO COEFFICIENTS				INITIAL AZIMUTH ERROR
1ST ORDER NONLINEARITY (δF_{il})	CROSS-TRACK g^2 (Bc2)	PLATFORM COMPLIANCE	BIAS	g-DRIFT	g^2-DRIFT	g^4-DRIFT	
0.99	0.99	0.99	0.99	0.99	0.96	0.98	0.80

TABLE VI

BEST FLY-2 PRIORITY ERROR SOURCE COEFFICIENT RECOVERY RATIOS

ACCELEROMETER COEFFICIENTS			GYRO COEFFICIENTS						
1ST ORDER NONLINEARITY (δF_{il})	CROSS-TRACK g^2 (Bc2)	PLATFORM COMPLIANCE	BIAS	g-DRIFT		g^2-DRIFT		g^4-DRIFT	
				δC	δD	δB	δE	P	J
0.93	0.96	0.93	0.99	0.76	0.81	0.48	0.18	0.07	0.22

TABLE VII

EFFECT OF AZIMUTH OFFSET VARIATIONS ON ERROR RECOVERY - FLY-2/GPS

DIFFERENTIAL AZIMUTH OFFSET $\lvert\Delta\alpha_1-\Delta\alpha_2\rvert$ (deg)	INDIVIDUAL AZIMUTH OFFSETS (deg)		COMPOSITE COEFFICIENT RECOVERY	GUIDANCE CEP RECOVERY
	$\Delta\alpha_1$	$\Delta\alpha_2$		
30	0	30	0.29	0.62
	-15	15	0.47	0.65
45	0	45	0.32	0.63
	-15	30	0.44	0.72
	-22.5	22.5	0.49	0.76

TABLE VIII

SUMMARY OF GUIDANCE ERROR RECOVERY RATIOS AT BOOST BURN-OUT

TRACKING SYSTEM	GUIDANCE RECOVERY RATIOS				
	DOWN-RANGE POSITION ERROR	CROSS-TRACK POSITION ERROR	DOWN-RANGE VELOCITY ERROR	CROSS-TRACK VELOCITY ERROR	INITIAL AZIMUTH ERROR
FLY-1/Radar	0.98	0.56	0.99	0.63	0.80
FLY-2/Radar	0.71	0.50	0.71	0.60	0.60
FLY-2/GPS	0.01	0.02	0.02	0.03	0.36

TABLE IX

SECONDARY GUIDANCE SYSTEM PRIORITY GYRO ERROR SOURCE COEFFICIENT RECOVERY RATIOS

INSTRUMENTATION CONFIGURATION	RECOVERY RATIOS			
	δB	δE	P	J
FLY-1/Radar	0.98	0.96	0.98	0.99
FLY-2/Radar	0.54	0.33	0.22	0.56
FLY-2/GPS	0.51	0.25	0.15	0.37

TABLE X

FLIGHT TEST PARAMETERS

PARAMETER	STM-13W	PVM-18
Launch Date	31 January 1977	31 January 1980
Launch Time	05:05:00 PST	05:40:00 PST
Reentry Angle	Low	Medium
Primary Azimuth Offset	0 deg	0 deg
Secondary Azimuth Offset	45 deg	0 deg

TABLE XI

GPS SATELLITE/RECEIVER GEOMETRY

SATELLITE NUMBER	GEOMETRY*	
	AZIMUTH (deg)	ELEVATION (deg)
1	198.0	34.8
2	324.0	32.0
3	42.8	66.0
4	36.3	36.5

*At time of launch.

VALIDATION OF FILTER/SMOOTHER MODELS

I. INTRODUCTION

A. MOTIVATION FOR MODEL VALIDATION

The importance of the model used in design of a Kalman filter/smoother is well-recognized. In practical applications, tradeoffs inevitably exist between complex models which represent the real world with great fidelity and simplified approximate models which lead to less costly implementations. To the system analyst, the question of model validity is of great importance for two reasons. First, if Kalman filtering results are to be interpreted with confidence, it is essential that the model be a valid representation of the physical system. Second, and perhaps even more important, the model often represents a baseline design of the system with associated baseline system performance characteristics. If that model is not a valid representation of the actual system being tested, there is an implication that the system does not match its baseline design, and therefore may not meet its baseline performance characteristics.

In this chapter, a procedure for model validation is described. The procedure is based on statistical hypothesis tests focused on the question: "Are the estimates generated by a Kalman smoother from system test data consistent with the system model?" The procedure is a multiple-test approach; filter/smoother estimates from several system tests are combined in a common data base on which the statistical hypothesis tests are performed.

The model validation procedure discussed is based on well-known statistical hypothesis testing methods [1,2]. The contributions presented are first, a problem formulation and second, analysis and data reduction procedures which lead to efficient application of the statistical hypothesis tests. Another approach to the model validation problem which is potentially applicable to the same class of systems is based on maximum-likelihood parameter identification procedures [3,4]. In each of those methods, Expectation-Maximization (E-M) algorithms [5] are used to obtain maximum likelihood estimates of statistical parameters. Those estimates are then available for use in the calculation of test statistics for hypothesis testing similar to that discussed here.

Another approach to model validation is based on direct examination of residual sequences resulting from an ensemble of test results [6-8]. Procedures based on this approach are designed only to detect the presence of data/model inconsistency; unlike the procedures described in this chapter, they do not attempt to isolate the source of the inconsistency to specific system parameters. Kashyap and Rao [6] discuss the model validation problem from this viewpoint with an emphasis on much smaller systems than considered here and under the restrictive assumption that all tests are based on identical scenarios. Goodrich and Caines [7] consider the statistical parameter identification problem based on likelihood functions formed directly from the innovations (i.e., filter residuals) of each test. Baram [8] approaches the model validation problem by generating test statistics from normalized residual sequences from each test. Although these latter approaches have considerable potential because of their generality, they are computationally very costly, and do not lead to the development of an expanding data base available for use by the test designer/analyst.

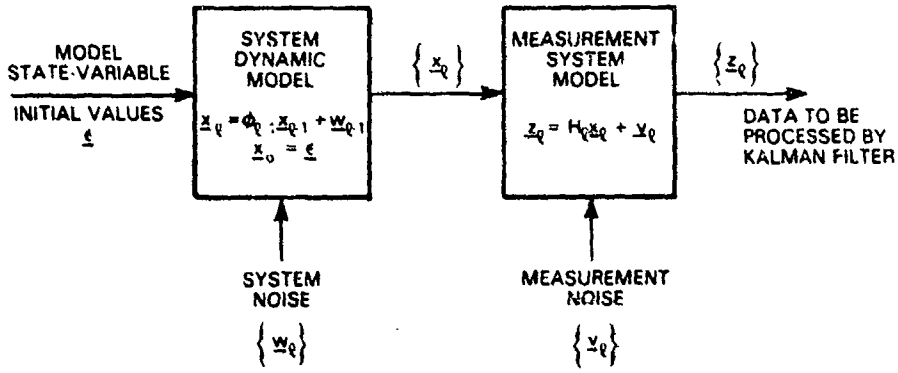
B. THE MODEL VALIDATION PROBLEM FOR LINEAR DISCRETE-TIME SYSTEMS

The models used in the design of post-test data processors are typically developed via a two-step procedure. First, models using algebraic and differential (or difference) equations are developed for each component and error source of the complete system based on physical understanding. Second, specific numerical values for the various parameters of these models are determined through laboratory tests, analysis of previous field tests, and engineering judgment. This procedure leads to a formulation of the model validation problem based on the model structure and parameters shown in Fig. 1.

In this discrete-time formulation, the structure of the model, including the order of the difference equations describing the dynamic model, is assumed to be correct. The various model parameters shown in Fig. 1 are less certain and are to be validated. The vector $\underline{\varepsilon}$ is referred to as the initialization vector for the model. Use of a separate symbol for $\underline{\varepsilon}$ (which is exactly $\underline{x}_0$ in Fig. 1) is motiviated by the multiple-phase models discussed in Section II D. Note that a separate block has been shown representing the measurement system because the measurement system model is an essential part of the overall model validation question. If dynamic states are used in the measurement system model, they must be included in the $\underline{x}_k$ vector representing system dynamics in Fig. 1.

The model parameters fall into two categories: Structural parameters which define the elements of matrices Φ_k and H_k and statistical parameters which define the normal density functions which model $\underline{\varepsilon}$, $\underline{w}_k$ and $\underline{v}_k$. The approach described here is directed only at $\underline{b}$ and $\underline{\Sigma}$, the statistical parameters associated with $\underline{\varepsilon}$. For typical systems, these parameters are among the most crucial in determining overall performance. In a ballistic missile system, for example, non-zero values of $\underline{b}$ lead directly to a bias in the impact distribution; similarly, off-nominal values of $\underline{\Sigma}(\neq \underline{\Sigma}_0)$ lead directly to an off-nominal Circular-Error-Probable (CEP) for the system. The other statistical parameters and the structural parameters in Fig. 1 should not be dismissed as unimportant, but they are

R-74149



R-74150

SYMBOL	DEFINITION	FILTER BASELINE MODEL
$\underline{b}, \Sigma$	MEAN-VALUE AND COVARIANCE OF $\underline{\xi}$	$\underline{b} = 0, \Sigma = \Sigma_0$
Φ_ℓ	SYSTEM STATE-TRANSITION MATRICES	Φ_ℓ^0
$\underline{m}_w, Q_\ell$	MEAN-VALUE AND COVARIANCE OF $\underline{w}_\ell$	$\underline{m}_w = 0, Q_\ell = Q_\ell^0$
$\underline{m}_v, R_{v\ell}$	MEAN-VALUE AND COVARIANCE OF $\underline{v}_\ell$	$\underline{m}_v = 0, R_{v\ell} = R_{v\ell}^0$

Figure 1 Model Structure and Parameters

generally somewhat better known than $\underline{b}$ and Σ. Exceptions to this, and an area of current research, are the parameters associated with Markov error models often used to represent various system error sources. Approaches to validating these parameters have been presented in the literature [7,8] but algorithms which are practical for use in large system test data analysis have not been developed as yet.

The model validation problem addressed in this chapter is as follows:

Given Models as defined by the matrices:

$$\Sigma_0,\ \Phi_\ell^k,\ H_\ell^k,\ Q_\ell^k,\ R_{v\ell}^k$$

$k = 1, \ldots, N$ = Number of tests

$\ell = 1, \ldots, \ell_f(k)$ = Number of Kalman updates for test k

and optimal smoothed estimates $\hat{\underline{x}}_0^{*k}$, of the initial state vector of each test, $\underline{x}_0^k$

Evaluate

Consistency of the model assumptions:

$\underline{b} = 0,\ \Sigma = \Sigma_0$

with the estimates, $\hat{\underline{x}}_0^{*k}$, assuming that the remainder of the model is correct

Only the model initial-mean validation procedures are described in this chapter. Validation of the model initial covariance matrix is based on the same input data and involves similar but not identical algorithms. A summary of covariance matrix validation procedures is given in Section IV.

C. ACKNOWLEDGMENT

Design and development of the procedures described in this chapter and of a software system which implements them has been the work of several individuals at The Analytic Sciences Corporation. The original concept was developed J.L. Center and P.J. Olinski.

Design of the bias capability procedures was primarily by J.E. Sacks and covariance matrix validation procedures were developed by F.K. Sun. The software system, including the data base design, was developed by S.L. Rubin.

II. VALIDATION OF THE MODEL INITIAL MEAN

In this section, details of the model-mean validation procedures are presented. The first subsection provides a description of a "data equation" representation of smoothed estimates from a Kalman filter/smoother. The other two subsections describe algorithms for data processing and capability analysis which are based on the data equation.

A. THE DATA EQUATION

Efficient computation of all test statistics and probabilities described in this section is based on a single equation, called the data equation, for each test result. The data equation defines the explicit relationship between smoothed estimates calculated by a Kalman filter/smoother algorithm and $\underline{\epsilon}$, the initialization vector of the dynamic model state. Figure 2 illustrates the relationship between the data equation and the model structure of Fig. 1. The smoothed estimate of $\underline{x}_o$ which appears on the left-hand side of the data equation can be interpreted as a noisy measurement of $\underline{\epsilon}$. The response matrix, D, describes the cross-coupling which exists between initialization errors and estimates, and the noise term, $\underline{\epsilon}'$, represents the accumulated effect of the assumed zero-mean dynamic system and measurement system noise processes, $\{\underline{w}_\ell\}$ and $\{\underline{v}_\ell\}$.

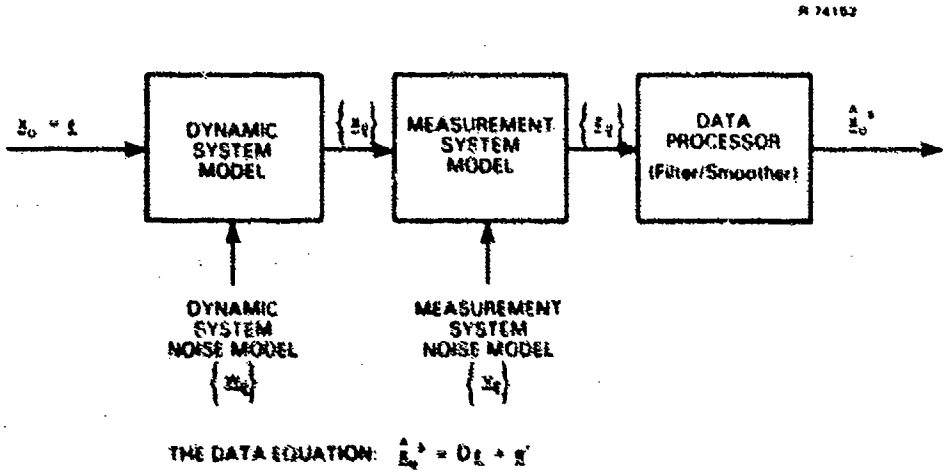


Figure 2 Data Equation Description of Complete System Structure

All of the data processing procedures which are used for validating statistical parameters (Σ_o and $\underline{b}$) of $\underline{\epsilon}$ are based on using only $\hat{\underline{x}}_o^s$, D, and R as input data. It can be shown that, in fact, $\hat{\underline{x}}_o^s$ is a sufficient statistic for Σ_o and $\underline{b}$. That is, one could not infer any more about the probability density function of $\underline{\epsilon}$ by any type of processing on the original data sequence, $\{\underline{z}_\ell\}$, than by correctly processing $\hat{\underline{x}}_o^s$ alone. This compression of the thousands of individual measurements contained in typical data sequences into a single vector estimate makes the validation procedures described in this chapter attractive.

In the following paragraphs, the data equation for each test k ($k = 1, \ldots, N$) is described. The superscript k is suppressed to avoid unnecessary notational complexity, but it should be remembered that, in general, different data equation matrices (D^k, R^k) result for each test.

1. Data Equation for Initial Mean Validation

Let the initialization vector in Fig. 2 be rewritten

$$\underline{\epsilon} = \underline{b} + \underline{\epsilon}^r \tag{1}$$

where

$\underline{b} = E\{\underline{\epsilon}\}$ the actual initial mean, n×1

$\underline{\epsilon}^r \sim N(0, \Sigma_o)$

where Σ_o is assumed known. The data equation in Fig. 2 can be rewritten

$$\hat{\underline{x}}_o^s = D\underline{b} + \underline{\epsilon} \tag{2}$$

where

$$\text{Cov}\ (\underline{e}) = D\ \Sigma_o\ D^T + \text{Cov}\ (e') \tag{3}$$

and we define

$$R = \text{Cov}\ (\underline{e}) = \text{Cov}\ (\hat{\underline{x}}_o^s) \tag{4}$$

For an optimal filter/smoother,

$$\text{Cov}\ (\underline{x}_o) = \text{Cov}\ (\hat{\underline{x}}_o^s + \tilde{\underline{x}}_o^s)$$

or

$$\Sigma_o = R + P^s \tag{5}$$

where P^s is the error covariance of the fixed-point smoothed estimate of $\underline{x}_o$. Furthermore, for an optimal filter/smoother, and for $\underline{\varepsilon} = \underline{x}_o$,

$$E\left[\hat{\underline{x}}_o^s\ \underline{\varepsilon}^T\right] = E\left[(D\ \underline{\varepsilon} + \underline{e}')\ \underline{\varepsilon}^T\right] = D\ \Sigma_o$$

$$= E\left[\hat{\underline{x}}_o^s\ (\hat{\underline{x}}_o^s + \tilde{\underline{x}}_o^s)^T\right] = R$$

Therefore, provided the inverse exists,

$$D = R\ \Sigma_o^{-1} \tag{6}$$

So, given the *a priori* covariance, Σ_o, and the Kalman smoother error covariance, P^s, one can ideally use Eqs. (5, 6) to find matrices R and D necessary to define the data equation, Eq. (2).

Unfortunately, for many system models of interest, observability of the various components of $\underline{x}_o$ varies widely. The effect is that R is a very poorly-conditioned matrix and computation of R via

$$R = \Sigma_o - P^s \tag{7}$$

is very inaccurate. The inaccuracy is due both to the finite word length effect on the subtraction and to round-off error during computation of P^s in typical filter/smoother algorithms. This drawback, combined with the requirement for inverting Σ_o in Eq. (6), has motivated the use of recursive equations for the direct computation of the D and R matrices. These equations are presented in the appendix along with an outline of their derivation.

2. The Normalized Data Equation

Although data processing procedures based on statistical hypothesis tests could be developed directly from Eq. (2), there are two practical reasons for not doing this. First, test statistics based on $\hat{\underline{x}}_o^s$ in Eq. (2) would not be distributed according to standard probability distributions for which efficient computational procedures already exist. Second, as mentioned earlier, limited model observability results in a poorly-conditioned R-matrix, i.e., a high degree of linear dependency among the components of $\hat{\underline{x}}_o^s$. Both of these difficulties can be overcome by a normalization procedure which will result in a reduced number of normalized estimates, z_i, which will be pairwise uncorrelated. Test statistics based on these normalized estimates will be central- or non-central χ^2 random variables under the various hypotheses considered in the next section.

To obtain the normalized data equation, Eq. (2) is premultiplied by a matrix, M, chosen so that the resulting noise term, $M\underline{e}$, will have identity covariance. If we let

$$R = R^{\frac{1}{2}}\ (R^{\frac{1}{2}})^T \tag{8}$$

where $R^{\frac{1}{2}}$ is any positive semi-definite square-root of R, then M can be computed from

$$M\ R^{\frac{1}{2}} = I \tag{9}$$

Because of the ill-conditioned nature of typical R matrices, M will usually be a pseudo-inverse of $R^{\frac{1}{2}}$. One procedure which has proved satisfactory for computing M is based on use of a singular-value-decomposition algorithm to obtain matrices U and S such that

$$R = U\ S\ U^T \tag{10}$$

where

$$S = \text{Diag}\,\{s_1, s_2, \ldots, s_n\} \tag{11}$$

and U is an n×n matrix of eigenvectors of $R^T R$. The transformation, M, can therefore be computed using the formula

$$M = (S^{\frac{1}{2}})^{-} U^T \tag{12}$$

where $(\ )^{-}$ denotes a pseudo-inverse matrix and

$$(S^{\frac{1}{2}})^{-} = \left[\begin{array}{cccc:c} \frac{1}{\sqrt{s_1}} & & & 0 & \\ & \frac{1}{\sqrt{s_2}} & & & \\ & & \ddots & & 0_{m\times(n-m)} \\ 0 & & & \frac{1}{\sqrt{s_m}} & \end{array}\right] \tag{13}$$

is a rank-m pseudo-inverse of $S^{\frac{1}{2}}$. Multiplying Eq. (2) on the left by M yields the following normalized data equation for the bias problem,

$$z = H\,\underline{b} + \underline{v} \tag{14}$$

where

$$\underline{z} = M\,\underline{\hat{x}}_o^s \tag{15}$$

is an m × 1 vector of normalized estimates and

$$H = M\,D \tag{16}$$

is an m × n normalized response matrix. The covariance of the noise term in Eq. (14) is

$$\text{Cov}(\underline{v}) = \text{Cov}\,(M\underline{e}) = M\,R\,M^T = I_{m\times m} \tag{17}$$

as desired.

B. DATA PROCESSING PROCEDURES

Data processing for model-mean validation consists of a four-step procedure designed to "evaluate" consistency. The four steps are: model acceptance, error detection, error isolation, and parameter estimation. Each of the steps is based on quantities derived from the cumulative, normalized data equation

$$\underline{z} = H\,\underline{b} + \underline{v} \tag{18}$$

where $\underline{z}$, H, and $\underline{v}$ now represent collections of elements from single-test data equations,

$$\underline{z} = \begin{bmatrix} \underline{z}^1 \\ \vdots \\ \underline{z}^N \end{bmatrix}, \quad H = \begin{bmatrix} H^1 \\ \vdots \\ H^N \end{bmatrix}, \quad \underline{v} = \begin{bmatrix} \underline{v}^1 \\ \vdots \\ \underline{v}^N \end{bmatrix} \tag{19}$$

1. Hypothesis Testing Procedures

Model acceptance is designed to test the validity of the normalization process. If some of the model matrices (ϕ_k, H_k, Q_k, R_{vk}) do not accurately model the system which generated the data processed by the Kalman filter, then each $\underline{v}^k$ in Eq. (19) may not have identity covariance. This would cause subsequent statistical hypothesis tests to be unreliable because of deviations from the assumed χ^2 (central or non-central) distributions.

The hypothesis to be tested is

$$H_A: \quad \underline{v} \sim N\,(0, I) \tag{20}$$

The test statistic to be used is

$$\lambda_A = ||\underline{z} - H\,\hat{\underline{b}}||^2 \tag{21}$$

where $\hat{\underline{b}}$ is the least-squares (also maximum-likelihood under H_A) estimate of $\underline{b}$ based on Eq. (18). The hypothesis test is

$$\lambda_A \begin{array}{c} \text{Reject} \\ > \\ < \\ \text{Accept} \end{array} \lambda_A \tag{22}$$

where the threshold, λ_A, is determined from a specified level-of-significance, α, such that

$$\Pr \{ \Lambda_A > \lambda_A \mid H_A \text{ is true} \} = \alpha \tag{23}$$

The threshold, λ_A, is determined from Eq. (23) using the fact that if H_A is true, then

$$\Lambda_A \sim \chi^2_p \tag{24}$$

where degrees of freedom p = m - Rank H and m is the dimension of $\underline{z}$. Rejection of H_A means that, with high confidence, there exists a modeling error <u>other than</u> $\underline{b} \neq 0$, and the succeeding analysis procedures may yield misleading results. Non-rejection of H_A, of course, does not preclude the possibility of other modeling errors. It does imply that if such errors exist, they have not caused $\underline{z}$ to deviate significantly from its baseline statistical distribution.

<u>Error detection</u> is designed to detect the presence of an error of the type considered, a non-zero mean in this case. The hypothesis to be tested is simply:

$$H_D : \underline{b} \neq 0 \tag{25}$$

and the test statistic used is derived from the same least-squares solution used in the model acceptance test,

$$\Lambda_D = || H \hat{\underline{b}} ||^2 \tag{26}$$

The bias detection hypothesis test becomes

$$\Lambda_D \underset{\text{Accept}}{\overset{\text{Reject}}{\gtrless}} \lambda_D \tag{27}$$

where the threshold, λ_D, is again based on a specified level-of-significance, α,

$$\Pr \{ \Lambda_D > \lambda_D \mid H_D \text{ is true} \} = \alpha \tag{28}$$

Under hypothesis H_D,

$$\Lambda_D \sim \chi^2_p \quad , \quad p = \text{Rank } H \tag{29}$$

enabling calculation of the threshold, λ_D, from the central χ^2 distribution. This test, like the model acceptance test, is a significance test; rejection of H_D means that, with high confidence, a non-zero bias exists in the system.

Figure 3 is helpful in interpreting detection and isolation test results. The triangle shows the relationship between the data, $\underline{z}$, its projection on the linear space spanned by possible bias vectors which has length Λ_D (called "explained sum-of-squares" or ESS), and the residual vector which has length Λ_A (called "residual sum-of-squares" or RSS). For the isolation tests discussed below, the projection of $\underline{z}$ on a <u>subspace</u> of that spanned by all possible biases is considered.

Another interpretation of Λ_D is as a Generalized Likelihood Ratio (GLR) which could be used as a GLR test statistic to select one of the two alternative hypotheses:

$$H_D : \underline{b} = 0 \quad \underline{vs} \quad H_D : \underline{b} \neq 0.$$

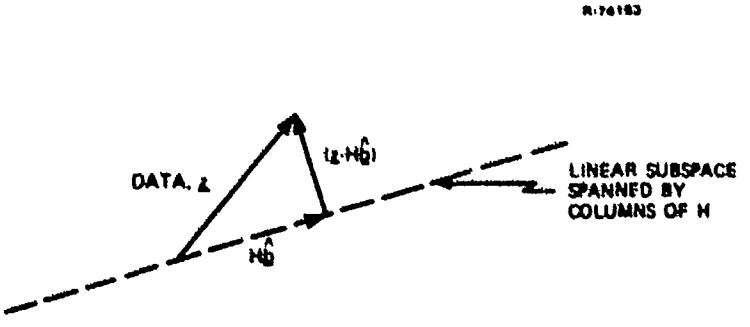


Figure 3 Decomposition of Data Into Explained and Residual Portions

The third and fourth steps of the consistency evaluation are done simultaneously following detection of a modeling error. In the error isolation step, the analyst attempts to isolate the error to a subset of the components of $\underline{b}$. The following notation is used to describe the isolation procedure:

$$B(J) = \{ \underline{b} : n\times 1 \text{ vector, } b_j \text{ arbitrary if } j \in J,\ b_j = 0 \text{ if } j \notin J \} \tag{30}$$

where

$$J = \{1 \le j_1 < j_2 \ldots < j_q \le n\}$$

is an index set with $q \le n$ elements. Selection of the index sets, J, could be based on engineering judgment or on automatic set selection procedures.

The hypothesis to be tested for each J can be written

$$H_J : \underline{b} \in B(J) \tag{31}$$

and an appropriate test statistic is

$$\Lambda_J = \Lambda_D - \| H \hat{\underline{b}}_J \|^2 \tag{32}$$

where $\hat{\underline{b}}_J$ is the least-squares (and maximum likelihood) solution of the over-determined set of equations

$$H_J \underline{b}_J \cong \underline{z} \tag{33}$$

In H_J, columns of H whose indices are not in J are replaced by zeros.

The form of the isolation test is

$$\Lambda_J \underset{\text{Accept}}{\overset{\text{Reject}}{\gtrless}} \lambda_J \tag{34}$$

where the threshold, λ_J, is determined from

$$\Pr \{ \Lambda_J > \lambda_J \mid H_J \text{ is true} \} = \alpha \tag{35}$$

Under hypothesis H_J, it can be shown that

$$\Lambda_J \sim \chi^2_p \qquad p = \text{Rank } H - \text{Rank } H_J \tag{36}$$

so that the threshold computation is again based on a central χ^2 distrubtion.

Each isolation test statistic, Λ_J, can be interpreted as a GLR. The two alternative hypotheses for each J are:

H_U : $\underline{b}$ unconstrained

H_J : $\underline{b} \in B(J)$

The fourth component of the evaluation is the estimate, $\hat{\underline{b}}_J$, generated for each of the sets tested. Engineering judgment is of utmost importance in the interpretation of the $(\Lambda_J, \hat{\underline{b}}_J)$ pairs. Λ_J is a measure of how much of the ESS, Λ_D, is explained by biases in B(J). A perfect score, $\Lambda_J = 0$, would indicate "certain" isolation of the bias. In actual data processing, random effects are aliased by various bias sources; different bias sources may have similar signatures in the data space. As a result, the analyst may "accept" (i.e., fail to reject) two or more sets, B(J), as explaining the data. Selection of the most likely bias candidates from among those accepted is based on statistical scoring techniques and, most importantly, good engineering judgment.

2. Output Information from Hypothesis Tests

Output information from a computer program which implements the hypothesis tests must include test statistics and thresholds for each of the tests along with the least-squares estimates, $\hat{\underline{b}}_J$, for each alternative index set, J. A sophisticated isolation algorithm should also include a ranking system for determining which of the alternatives are the best candidates for explaining the data. A discussion of criteria for ranking the various alternatives is beyond the scope of this chapter [2].

C. CAPABILITY ANALYSIS

In this subsection, procedures for calculating probabilities associated with the detection and isolation tests described above are presented. These probabilities are of great importance to the system analyst. First, during the planning stage of system tests and test instrumentation, it is important to learn as much as possible about the observability of various system errors, if they in fact exist, under the proposed test program. Second, capability analysis based on these probabilities provides a guide for planning and interpreting the data processing procedures and results.

1. Scaling the Bias Errors

Capability analysis results are based on the probability density functions of the test statistics. These densities are completely determined by the original model and specfication of model errors which are assumed to exist. The procedure to be used to specify error sources considered in capability analysis is to select errors which are "significant" according to a well-defined system performance criterion. This is called "scaling the error sources."

For model mean validation, system errors are non-zero components in the initialization error bias vector, $\underline{b}$. The scaling criterion has the form:

$$||\underline{b}||_E^2 = \underline{b}^T E \underline{b} \tag{37}$$

where

$$E = \frac{1}{N} \sum_{k=1}^{N} J_k^T J_k \tag{38}$$

N = number of tests from which data are to be processed,

and

$$J_k = \frac{\partial \underline{r}_k}{\partial \underline{\varepsilon}} \quad s \times n \tag{39}$$

is a gradient matrix defining the sensitivity of a sxl output vector, $\underline{r}$, to the initialization error vector, $\underline{\varepsilon}$.

2. Evaluating Detection and Isolation Probabilities

Probabilities associated with the detection and isolation tests are calculated from the known probability density functions for test statistics Λ_D and Λ_J. Let $\underline{b}_T$ represent a hypothesized true bias in the a priori distribution of $\underline{\varepsilon}$. Then the cumulative data vector, $\underline{z}$, in Eq. (18) would be

$$\underline{z} = H\underline{b}_T + \underline{v} \tag{40}$$

Therefore $\underline{z}$ would be distributed normally with mean $H\underline{b}_T$ and covariance I, denoted $\underline{z} \sim N(H\underline{b}_T, I)$. Let the least-squares (and maximum likelihood) estimate of $\underline{b}_T$ be written

$$\hat{\underline{b}} = H^- \underline{z} \tag{41}$$

where H^- represents a pseudo-inverse of H. Therefore, the detection test statistic defined in Eq. (26) can be written

$$\Lambda_D = \underline{z}^T \left[H^{-T} (H^T H) H^- \right] \underline{z} \tag{42}$$

Since the bracketed matrix is idempotent, and $\underline{z}$ is normal with identity covariance, Λ_D is a non-central χ^2 random variable, denoted

$$\Lambda_D \sim \chi^2 (p, \delta^2)$$

where

p = degrees of freedom = Rank H

δ^2 = non-centrality parameter = $||H\underline{b}_T||^2$

The probability of detecting the hypothesized bias, $\underline{b}_T$, is given by

$$P_D(\underline{b}_T) = \Pr \{\chi^2(p, \delta^2) > \lambda_D\} \tag{43}$$

where the threshold, λ_D, satisfies Eq. (28). This probability can be evaluated using well-known procedures for non-central χ^2 densities.

For the error probabilities associated with isolation tests, the situation is slightly more complex, but again results in evaluating probabilities from a non-central χ^2 density function. The constrained estimate, $\hat{\underline{b}}_J$, can be denoted

$$\hat{\underline{b}}_J = H_J^- \underline{z} \tag{44}$$

Combining Eq. (32) with Eqs. (42), (44) we have

$$\Lambda_J = \underline{z}^T \left[H^{-T} (H^T H) H^- - H_J^{-T} (H^T H) H_J^-\right] \underline{z} \tag{45}$$

The bracketed matrix is again idempotent so that

$$\Lambda_J \sim \chi^2 (p', \delta_J^2) \tag{46}$$

where

$$p' = \text{rank } H - \text{rank } H_J \tag{47}$$

and

$$\delta_J^2 = ||(I-H_J H_J^-) H\underline{b}_T||^2$$

For a specified index set and hypothesized true bias $\underline{b}_T$, the probability of not rejecting (i.e., accepting) set J as possibly containing the bias is

$$P_J(\underline{b}_T) = \Pr\{\chi^2(p',\delta_J^2) < \lambda_J\} \tag{48}$$

where the threshold, λ_J, satisfies Eq. (35). If $\underline{b}_T$ is, in fact, contained in set B(J), then $\delta_J^2 = 0$, and

$$P_J(\underline{b}_T) = 1 - \alpha \tag{49}$$

where α is the level of significance used to select the threshold, λ_J.

Although Λ_D and Λ_J can be interpreted as generalized likelihood ratios, they were developed from a geometrical viewpoint.* Λ_J can be interpreted as the square of the distance from the point $(H\underline{b}_T + \underline{v})$ in $\underline{z}$-space to the set HB(J) defined by

$$HB(J) = \{\zeta : \zeta = H\underline{b}_J, \ \underline{b}_J \in B(J)\} \tag{50}$$

Then, $P_J(\underline{b}_T)$ can be interpreted as the probability that this distance is less than a threshold distance, λ_J, which defines an "acceptance region" around HB(J).

3. Output Information from Capability Analysis

For capability analysis, biases $\underline{b}_T$ which satisfy

$$||\underline{b}_T||_E = d \tag{51}$$

for a specified error magnitude, d, are considered. There are many ways in which probabilities associated with specified alternative sets, B(J), can be calculated and presented. One which has been successfully implemented is to consider true biases, $\underline{b}_T$, which are in the set

$$B(I,d) = \{\underline{b}_T : \underline{b}_T \in B(I) \text{ and } ||\underline{b}_T||_E = d\} \tag{52}$$

where I is a specified index set. Then maximum and minimum probabilities

$$P_{max,min}(J,I) = \max_{\underline{b}_T \in B(I,d)}, \min P_J(\underline{b}_T) \tag{53}$$

can be computed; they define the range of probabilities of accepting set B(J) as explaining a bias caused by $\underline{b}_T \in B(I,d)$. Examples of results of this type are presented in Section III.

*Developed by J.E. Sacks

D. SOFTWARE SYSTEM IMPLEMENTATION

This section describes an efficient software system implementing the procedures presented here. Extension of the procedures to multiple-phase models is discussed first since this extension has a significant impact on the software design. The remaining subsections summarize major design considerations and organization of the complete software system.

1. Multiple-Phase Models

The system model considered in the previous section is specified by the system and measurement model equations in Fig. 1. For some systems, however, a complete model of a complex system can be most efficiently represented by two or more sets of system and model equations, each set representing one "phase" of system operation. The relationships between variables describing system operation during two adjacent phases is defined at a specific value of the independent variable, called the interface time, by the following equation:

$$\underline{x}_{jo} = \Lambda_{j,j-1}\, \underline{x}_{j-1}\,(T_{j-1}) + \Lambda_{jo}\, \underline{\varepsilon}_j \tag{54}$$

where

$\underline{x}_{jo}$ = initial condition of phase j

$\underline{x}_{j-1}\,(T_{j-1})$ = state of phase j-1 at interface time, T_{j-1}

$\Lambda_{j,j-1}$, Λ_{jo} are known transformation matrices

$\underline{\varepsilon}_j \sim N\,(0, \Sigma_o^j)$ = the initialization error for phase j.

Equation (54) provides the motivation for distinguishing between $\underline{\varepsilon}$ and $\underline{x}_o$ in the previous section; the statistical parameters of $\underline{x}_{jo}$ vary from one test to another because of variations in $\Lambda_{j,j-1}$ and/or the distribution of $\underline{x}_{j-1}\,(T_{j-1})$ but $\underline{\varepsilon}_j$ is modeled by a probability distribution which can reasonably be assumed to be the same for all tests.

Distinct measurement systems will generally be used during distinct phases and the resulting measurements processed in distinct filter/smoother runs resulting in a collection of smoothed estimates, $\underline{\hat{x}}_o^{sj}$, $j = 1, \ldots, N_p$, where N_p is the number of phases for which estimates are available. Therefore, N_p data equations must be combined to form a single, normalized data equation for the entire test. Another important difference in the multiple-phase case is that in order to account correctly for correlations which must exist between estimates of initial conditions in different phases, smoothed estimates (along with corresponding D and R matrix elements) of a subset of the state vector of the earlier phase at interface time must be included in the data equation.

For a two-phase model, the composite data equation (before normalization) is

$$\begin{bmatrix} \underline{\hat{x}}^{s1} \\ \cdots \\ \underline{\hat{x}}_o^{s2} \end{bmatrix} = \left[\begin{array}{c|c} D_1 & 0 \\ \hline D_2\Lambda_{2,1}\Phi_1^{\star} & D_2\Lambda_{20} \end{array}\right] \begin{bmatrix} \underline{b}_1 \\ \cdots \\ \underline{b}_2 \end{bmatrix} + \begin{bmatrix} \underline{e}_1 \\ \cdots \\ \underline{e}_2 \end{bmatrix} \tag{55}$$

where

$$\underline{\hat{x}}^{s1} = \begin{bmatrix} \underline{\hat{x}}_o^{s1} \\ \cdots \\ \underline{\hat{x}}_i^{s1} \end{bmatrix}, \quad \Phi_1^{\star} = \Phi\,(T_1, t_o) \tag{56}$$

and $\underline{x}_i^1$ is the subset of the phase-one state vector at the interface time corresponding to non-zero columns of $\Lambda_{2,1}$. The composite covariance matrix is

$$R^c = \mathrm{Cov} \begin{bmatrix} \underline{\hat{x}}^{s1} \\ \underline{\hat{x}}_o^{s2} \end{bmatrix} = \left[\begin{array}{c|c} R_1 & \text{(symmetric)} \\ \hline D_2\Lambda_{2,1}R_i^1 & R_2 \end{array}\right] \tag{57}$$

where

$$R_i^1 = \mathrm{Cov}\,(\underline{\hat{x}}_i^{s1}, \underline{\hat{x}}^{s1}) \quad . \tag{58}$$

Since

$$R_1 = \mathrm{Cov}\begin{pmatrix} \underline{\hat{x}}_o^{s1} \\ \underline{\hat{x}}_i^{s1} \end{pmatrix} = \left[\begin{array}{cc} R_{10} & R_{10i} \\ \hline & R_i^i \end{array}\right] , \tag{59}$$

R_i^i is a sub-matrix of the R_1-matrix for phase 1 based on the augmented estimate defined in Eq. (56).

The fact that the smoothed estimate of $\underline{x}_{1i}$ provides critical information is not surprising. In fact, it can be shown [9] that in order to obtain sufficient statistics for estimating statistical parameters in a two-phase system, a third estimate (in addition to $\underline{\hat{x}}_o^{sj}$, j = 1,2) is necessary. In [9], it is shown that ($\underline{\hat{x}}_o^{s1}$, $\underline{\hat{x}}_i^{s1}$, $\underline{\hat{x}}_o^{s2}$) are sufficient statistics for the parameter estimation and validation problem of interest in this chapter.

2. Important Design Considerations

The procedures described in Section II are based on the normalized data equation elements ($\underline{z}_k$, H_k) for each test result. The extent to which calculations of ($\underline{z}_k$, H_k) should be formalized depends on several factors. Some of the most important are:

i) Size (i.e., number of states) and observability of the models

ii) Number of phases in the model

iii) Number of tests available for analysis

For highly observable (i.e., well-conditioned R) single-phase models and a small number of tests (say up to 10), little formal data organization is necessary. Data equation matrices (D,R) could be computed from closed-form formulas (Eqs. (5) and (6)) or from the recursive formulas (given in the appendix) implemented concurrently with the filter/smoother equations. One additional program would be required to compute and store normalized data equation elements, ($\underline{z}_k$, H_k). Subsequent analysis (data processing or capability analysis) could then be performed by specially-developed programs, possibly using subroutines from standard regression analysis packages. For more complex cases however, especially when multiple-phase models are involved, a more formal approach is essential to obtain the maximum benefit from the analysis techniques.

The organization of the software system presented in Fig. 4 includes four distinct program segments and two data base segments. The first of the program segments, The Interface Program, is optional depending on the choice of method for computing D and R matrices. If closed-form formulas are used, or if the recursive equations are integrated with the filter/smoother, then resulting (D,R) matrices can be written directly to the Input Data Base. Alternatively, the Interface Program, which implements the recursive equations for D and R, can read a file stored by the filter/smoother which completely describes the model used in the data processing. The minimum set of matrices which must be stored in that file is shown in Fig. 4. The interface program must also compute and store the cumulative state transition matrix for phase j of test k.

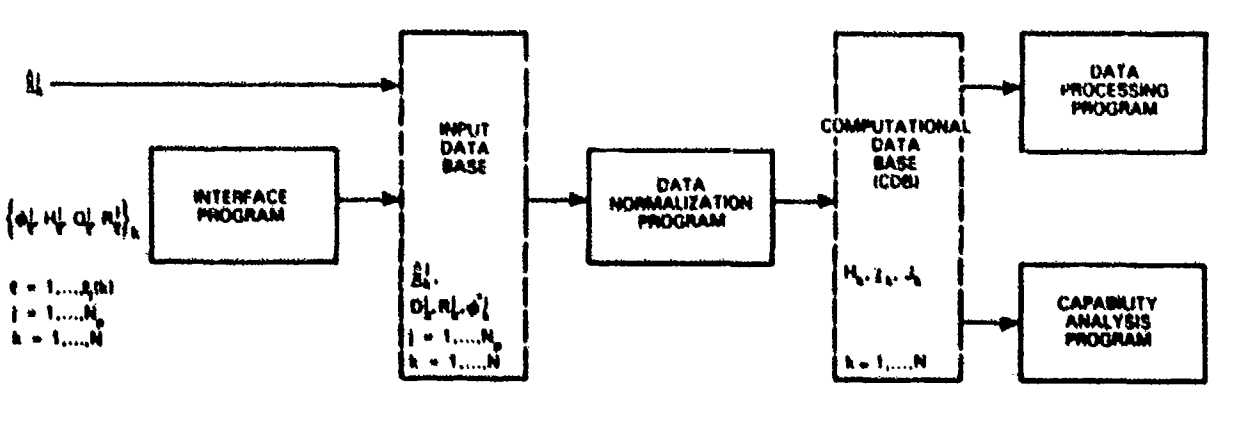


Figure 4 Software System Organization

The second program segment, Data Normalization, performs two functions. First, for multiple-phase models, it assembles the composite data equation matrices and the error response matrix, J_k, for each test, k = 1, . . . , N. Second, for all models, it normalizes the composite data equation elements and stores the resulting ($\underline{z}_k$, H_k) along with J_k on the Computational Data Base (CDB). All subsequent analyses deal only with data on

the CDB. Calculations performed by the Interface and Phase Combination Programs are relatively expensive but are performed only once for each phase of each test. Therefore, repeated calculations using the last two program segments can be performed at relatively low cost. The CDB contains sufficient statistics for the model validation problem.

The Data Processing and Capability Analysis Programs contain implementations of the statistical hypothesis tests and probability computations discussed in Section II. These two segments share many subroutines and are complementary in that one, Capability Analysis, assesses the capability of the other, Data Processing, to solve the problem. User inputs and possible outputs for these two segments are summarized in Fig. 5. Required inputs for the Capability Analysis Program do not include normalized data, $\underline{z}_k$; that is, all of the computations necessary to perform capability analysis can be done before data are actually collected. This program can therefore be used as a powerful observability analysis tool in the evaluation of proposed test programs. In the Capability Analysis Program, details of how the hypothesized true biases, $\underline{b}_T$, are to be selected are not indicated. One method was discussed in Section II B; example results are included in the next section.

R-74156

	DATA PROCESSING PROGRAM	CAPABILITY ANALYSIS PROGRAM
USER INPUTS	• $(H_k, \underline{z}_k)$ FOR TESTS TO BE ANALYZED • LEVEL-OF-SIGNIFICANCE, α • ALTERNATIVE INDEX SETS $\{J_i, i = 1, 2, \ldots\}$	• H_k FOR TESTS TO BE ANALYZED • LEVEL-OF-SIGNIFICANCE, α • ALTERNATIVE INDEX SETS $\{J_i, i = 1, 2, \ldots\}$ • HYPOTHESIZED TRUE BIASES, $\{\underline{b}_{Ti}, i = 1, 2, \ldots\}$
TYPICAL OUTPUT DATA	• HYPOTHESIS TEST RESULTS (acceptance, detection, isolation) • RANKED LIST OF ALTERNATIVE SETS $\{J_i, i = 1, 2, \ldots\}$	• DETECTION PROBABILITIES $P_D(\underline{b}_T)$ • ISOLATION PROBABILITIES $P_J(\underline{b}_T)$

Figure 5 Inputs and Outputs for Analysis Programs

III. EXAMPLE: INITIAL MEAN VALIDATION IN A TWO-PHASE SYSTEM

This example is based on a simple two-phase model of a ballistic missile system. The dynamic equations represent error models of Inertial Navigation Systems (INSs) which might be used in a mobile missile launcher (prelaunch) and onboard a missile (postlaunch). Numerical values used are for illustration only and are not based on any specific weapon system. The parameters to be validated are the components of the assumed zero mean of the state initialization vector for each phase

$$\underline{b}_j = E\{\underline{\epsilon}_j\} = \underline{0}_j, \quad j = 1,2. \tag{60}$$

A. BASELINE MODELS

1. Prelaunch (Phase 1) Model

The 14 states in this model are summarized in Table I along with the one-sigma uncertainty of the initialization vector and time constants associated with the Markov error models. This model represents errors in a local-level, velocity-damped (damping constant = 1.0), inertial navigation system moving in a path tangential to the earth's surface at velocities small relative to earth rotation [10,11]. Three representative error sources are modeled: gyro drift rates, accelerometer sensing errors due to deflection of the local vertical, and velocity reference errors.

Measurements processed by a Kalman filter/smoother for state estimation in phase 1 are to be generated by subtracting INS-indicated position from an externally-derived position measurement. Measurement noise is assumed to be 100 ft (one-sigma) in each horizontal direction (north, east).

TABLE I

PHASE 1 MODEL STATE VECTOR AND INITIALIZATION ERROR MODEL

STATE NUMBER	VARIABLE NAME	σ_{io}=INITIAL UNCERTAINTY	MARKOV TIME CONSTANT
1	Latitude Error	0.5 nmi	
2	Longitude Error	0.5 nmi	
3	North Velocity Error	0.2 kts	
4	East Velocity Error	0.2 kts	
5	North Platform Tilt	0.3 $\widehat{\text{min}}$	
6	East Platform Tilt	0.3 $\widehat{\text{min}}$	
7	Azimuth Misalignment	2.0 $\widehat{\text{min}}$	
8	North Gyro Drift-Rate	0.1 $\widehat{\text{min}}$/hr	18 hrs
9	East Gyro Drift-Rate	0.1 $\widehat{\text{min}}$/hr	18 hrs
10	Vertical Gyro Drift-Rate	0.1 $\widehat{\text{min}}$/hr	18 hrs
11	North Vertical Deflection	0.25 $\widehat{\text{min}}$	50 hrs
12	East Vertical Deflection	0.25 $\widehat{\text{min}}$	50 hrs
13	North Velocity Reference Error	0.5 kts	(Bias Error)
14	East Velocity Reference Error	0.5 kts	(Bias Error)

2. Postlaunch (Phase 2) Model

Table II describes the 17 states used to represent a space-stable inertial navigation system with orthogonal axes parallel to an earth-centered coordinate system. The table also includes one-sigma uncertainties of the components of $\underline{\varepsilon}_2$. Two representative error sources are modeled: accelerometer bias errors due to miscalibration and uncompensated gyro drift-rate errors. The latter include biases in each coordinate axis (three gyros assumed) and two terms representing thrust-dependent bias errors in the x and z directions.

TABLE II

PHASE 2 MODEL STATE VECTOR AND INITIALIZATION ERROR MODEL

STATE NUMBER	VARIABLE NAME			σ_{oi}=INITIAL UNCERTAINTY
1	r_x	Position Error	INS Errors	50 ft
2	r_y			100 ft
3	r_z			100 ft
4	v_x	Velocity Error		0.5 ft/sec
5	v_y			0.5 ft/sec
6	v_z			0.5 ft/sec
7	ψ_x	Misalignment		0.5 $\widehat{\text{min}}$
8	ψ_y			0.2 $\widehat{\text{min}}$
9	ψ_z			0.2 $\widehat{\text{min}}$
10	a_x	Accelerometer Errors	Bias Error Source Models	0.02 ft/sec^2
11	a_y			0.02 ft/sec^2
12	a_z			0.02 ft/sec^2
13	ε_x	Gyro Drift-Rate		8 $\widehat{\text{min}}$/hr
14	ε_y			8 $\widehat{\text{min}}$/hr
15	ε_z			8 $\widehat{\text{min}}$/hr
16	ε_{fx}	g-Sensitive Gyro		10 $\widehat{\text{min}}$/hr/g
17	ε_{fz}	Drift-Rate		10 $\widehat{\text{min}}$/hr/g

Position measurements for phase 2 are assumed to be similar to those in Phase 1 so that $\underline{z}$ consists of measurements of INS position error along each coordinate axis with additive noise which has a one-sigma value of 30 ft.

3. Phase Interface Model

The following transformation matrices define a simple model of the process by which the Phase 2 INS is initialized by the Phase 1 INS at interface time just prior to launch. The complete initial state vector for Phase 2 can be written

$$\underline{x}_{20} = \Lambda_2 \Lambda_1 \underline{x}_1(t_i) + \underline{\varepsilon}_2 \tag{61}$$

where the Λ matrices are defined as follows.

Λ_1 defines a 7×1 output vector from Phase 1,

$$\underline{y}_{1i} = \Lambda_1 \underline{x}_1(t_i) \tag{62}$$

where

$$\Lambda_1 = (\Lambda_{11} \quad 0_{7\times 7}) \tag{63}$$
$$\Lambda_{11} = \text{Diag}\ (1,\ \cos L,\ 1,\ 1,\ 1,\ 1,\ -1)$$

and L = latitude at launch.

Next, Λ_2 contains the necessary scale factors and coordinate transformations to map $\underline{y}_{1i}$ into the Phase 2 INS errors:

$$\underset{17\times 7}{\Lambda_2} = \begin{bmatrix} \Lambda_{21} & & \\ & \Lambda_{22} & \\ & & \Lambda_{23} \\ 0_{8\times 7} & & \end{bmatrix} \tag{64}$$

where

$$\Lambda_{21} = C_1 \begin{pmatrix} 0 & 0 \\ -\sin\beta & \cos\beta \\ \cos\beta & \sin\beta \end{pmatrix},\ \Lambda_{22} = C_2 \begin{pmatrix} 0 & 0 \\ -\sin\beta & \cos\beta \\ \cos\beta & \sin\beta \end{pmatrix},\ \Lambda_{23} = \begin{pmatrix} 0 & 0 & -1 \\ -\sin\beta & \cos\beta & 0 \\ \cos\beta & \sin\beta & 0 \end{pmatrix}$$

In these matrices, β represents the launch direction (azimuth) and C_1, C_2 are position and velocity unit conversion factors. The 17 × 1 vector, $\underline{\varepsilon}_2$, in Eq. (61) represents initialization errors in the Phase 2 INS introduced during the transfer alignment process and Phase 2 INS instrument errors (biases). These errors are assumed to be uncorrelated with $\underline{x}_1(t_i)$. The mean value of $\underline{\varepsilon}_2$ is to be validated.

B. DESCRIPTION OF THE TESTS

For this example, 10 test scenarios were simulated using standard covariance analysis procedures. Parameters which summarize the test scenarios are tabulated in Table III. The 10 tests describe a mix found in a typical test program, some yielding much better observability of system error sources than others. No one mission would provide a clear test of the correctness of the zero-mean assumption for the model's initialization vectors. Only by processing data from an ensemble of missions can a determination be made with a high degree of confidence.

TABLE III

PARAMETERS OF THE TEST SCENARIOS

TEST NUMBERS	MODEL PARAMETERS				BASELINE MODEL PREDICTED PERFORMANCE CEP (feet)
	Phase 1	Phase 2			
	LATITUDE L	LAUNCH AZIMUTH β	LAUNCH ELEVATION θ	RANGE R(nmi)	
1,4	30°	135°	45°	850	8400
2	30°	135°	30°	2400	15000
3	30°	135°	65°	1500	14200
5,8	0°	135°	45°	850	8800
6	0°	135°	30°	2400	15200
7	0°	135°	65°	1500	14200
9,10	30°	180°	45°	1900	8900

C. CAPABILITY ANALYSIS RESULTS FOR THE EXAMPLE SYSTEM

1. Bias Detection Capability

Table IV contains bias detection capability results for several cases. Five different hypothesized "true" bias sources are considered. Each bias is identified by the state variables in the left hand column which indicate that the corresponding element of $\underline{\varepsilon}_1$ or $\underline{\varepsilon}_2$ is biased by the indicated magnitude. The a priori sigmas are from Tables I and II. For the two cases which contain more than one biased component, two probabilities are printed corresponding to the worst ($P_{D(min)}$) and best ($P_{D(max)}$) cases among all those biases satisfying the error constraint, $||\underline{b}_T||_E = d$. This error is the rms miss-distance across the ensemble of 10 tests which would be caused by each hypothesized bias. The two values of d for which results are given in the tables (1000 and 2000 ft) are relatively small for the example system which has an average CEP across the 10 tests of 11,000 ft (see Table III). Thus, the probabilities presented correspond to a system bias which would be difficult to detect in the impact domain defined by down-range and cross-range error based on the 10-test ensemble.

TABLE IV

BIAS DETECTION CAPABILITY RESULTS

TRUTH SET CONTAINING BIAS AND A PRIORI SIGMA	ERROR MAGNITUDE d (ft)	$P_{D(MIN)}$ (a) NUMBER OF TESTS			BIAS AT $P_{D(MIN)}$	$P_{D(MAX)}$ (a) NUMBER OF TESTS			BIAS AT $P_{D(MAX)}$
		5	10	20		5	10	20	
$\{\delta V_N\}$, 0.2 kts	1000	1.0	1.0	1.0	3.29 (b)				
	2000	1.0	1.0	1.0	6.58 (b)				
$\{\delta H\}$, 2.0 min	1000	α	α	0.11	0.209 (c)	(ONLY ONE VALUE OF SINGLE-COMPONENT BIAS YIELDS REQUIRED d)			
	2000	0.11	0.11	0.12	0.419 (c)				
$\{\varepsilon_N\}$, 0.1 min/hr	1000	0.18	0.29	0.53	0.086				
	2000	0.52	0.86 (d)	1.0	0.172				
$\{\delta A_{VD(N,E)}\}$ 0.25 min	1000	0.14	0.19	0.29	-0.124 0.124	0.26	0.44	0.77	0.235 0.259
	2000	0.29	0.52 (d)	0.86	-0.245 0.250	0.78	0.99	1.0	0.471 0.517
$\{a_x, a_y, a_z\}$ 0.02 ft/sec^2	1000	α	α	0.11	-0.0020 (c) 0 -0.0019	0.14	0.18	0.27	-0.0072 0 0.0075
	2000	0.11	0.12	0.14	-0.0040 (c) 0 -0.0038	0.27	0.48	0.82	-0.014 0 0.015

NOTES:

(a) Based on thresholds for level-of-significance α = 0.10; Table entries = α indicate no detectability of the hypothesized bias

(b) $b_i \gg \sigma_i$ implies low system sensitivity to ε_i

(c) $b_i < \sigma_i$ implies high system sensitivity to ε_i

(d) Simulated data processing results for these cases are discussed in text.

For each case (i.e., each bias source and each d value) three probabilities are given corresponding to the number of test results assumed to be available for processing. The five-test case assumes that only tests 1-4 and 9 are available and the 20-test case assumes that each of the 10 tests described in Table III is conducted twice.

The various cases in Table IV can be grouped into three broad categories. For some cases, indicated by note (b) in the figure, the bias magnitude required to produce a significant d is far greater than the modeled prior uncertainty. Such a bias, if it existed in the system, would represent a severe system anomaly and would be easily detected by the data processing algorithm if the associated state variable is observable

in the classical control/estimation theory sense. For the north-velocity bias case shown, the smoothed estimates of the system initial velocity for the 10-test ensemble would clearly indicate the presence of the bias; the model validation data processing software would not be necessary.

A second category is typified by the cases indicated by note (c) in Table IV. Here the system performance measure is extremely sensitive to the hypothesized bias so that, for the {δH} case for example, a bias of only one-tenth of the a priori uncertainty would cause a 1000 ft rms error. For these cases, even perfect measurement of the states of interest would have great difficulty in distinguishing between the unbiased a priori model and the actual, slightly biased density function.

The third category includes the rest of the cases in the table and the majority of cases in practical applications. For these cases, system sensitivity is moderate, relative to the d of interest, and the unique capability of the data processing algorithm based on data from multiple tests is of maximum advantage. The steadily increasing probabilities as the numbers of tests increases is clearly evident.

2. Bias Isolation Capability

Table V contains bias isolation capability results. Three of the hypothesized bias sets for which detection capability results were presented in Table IV are used here. The probability of accepting each of six alternative sets is computed for each case. As with the detection capability results, for multiple-component hypothesized biases, worst-case ($P_{J(MAX)}$) and best-case ($P_{J(MIN)}$) probabilities are computed for each case. The effect of increasing d (from 2000 to 4000 ft) or the number of tests (from 10 to 20) is shown for each truth/alternative pair.

TABLE V

BIAS ISOLATION CAPABILITY RESULTS

Pr{ACCEPT THE ALTERNATIVE SET}
(FOR MULTIPLE COMPONENT TRUE BIASES, $P_J(MAX)/P_J(MIN)$ ARE SHOWN)

TRUTH SET CONTAINING BIAS	ALTERNATIVE SETS							
	d	N	$\{\varepsilon_N, \varepsilon_E, \varepsilon_z\}$	$\{\delta A_{VD}\}$	$\{\delta V_{REF}\}$	$\{\psi\}$	$\{a\}$	$\{\varepsilon, \varepsilon_f\}$
$\{\varepsilon_N\}$	2000	10	1-α	0.13	0.13	0.13	0.12	0.11
	2000	20	1-α	0.	0.	0.	0.	0.
	4000	10	1-α	0.	0.	0.	0.	0.
$\{\delta A_{VD(N,E)}\}$	2000	10	0.48/0.03	1-α	0.90/0.89 (b)	0.57/0.02	0.46/0.01	0.48/0.01
	2000	20	0.14/0.01	1-α	0.90/0.88	0.25/0.	0.13/0.	0.15/0.
	4000	10	0.03/0.	1-α	0.89/0.86	0.02/0.	0./0.	0./0.
$\{a\}$	2000	10	0.88/0.50	0.88/0.50	0.88/0.51	0.88/0.50	1-α	0.88/0.49
	2000	20	0.86/0.17	0.86/0.17	0.86/0.17	0.86/0.17	1-α	0.85/0.16
	4000	10	0.84/0.01	0.81/0.01	0.81/0.01	0.81/0.01	1-α	0.80/0.01

NOTES:
(a) Based on thresholds for level-of-significance α = 0.10; Table entries = 1-α = the designed probability that an alternative set containing the truth be accepted.

(b) Probability 0.90 = 1-α indicates extreme difficulty of distinguishing between this truth/alternative pair.

A general feature of the results is that for alternative sets which do not contain the hypothesized bias, the probabilities of acceptance (i.e., the probabilities of mis-isolation) do not vary much from one alternative set to another. This result is typical of mis-isolation probabilities which have been computed for many weapon system models. Exceptions to this general feature do occur however; one such case is indicated by note (b) in the figure. That result (mis-isolation probability = 1-α) means that biases in the vertical deflection errors and the velocity reference errors in a local-level inertial navigation system have almost identical signatures in the data space. It can be shown that by using engineering judgment in interpreting isolation test results, incorrect acceptance of δV_{REF} as a bias source can almost certainly be avoided.

D. DATA PROCESSING RESULTS

In order to illustrate applications of data processing algorithms to the example two-phase system, two sequences of simulated, normalized, biased data, $\{z^k\}$, were

generated. Biases simulated in the two sequences corresponded to two cases considered in the previous bias capability analysis results, each of which caused d = 2000 ft of rms error:

Case I $\underline{b}_T$: $\{b_8 = E\{\varepsilon_N(0)\} = 0.172$ $\widehat{\text{min}}$/hr$\}$

Case II $\underline{b}_T$: $\{b_{11} = E\{\delta A_{VD(N)}(0) = -0.245$ $\widehat{\text{min}}$,

$b_{12} = E\{\delta A_{VD(E)}(0) = 0.250$ $\widehat{\text{min}}\}$

For each case, the sequence $\{\underline{z}^k\}$ was made up of 12 sets of 10 $\underline{z}^k$ (representing results of each of the 10 test scenarios).

1. Case I: Single-Component Bias

This case is based on a single-component bias which is highly detectable (P_D = 0.86) and easily isolated ($P_J \tilde{=} 0.13$ for J not including ε_N). In 10 of the 12 trials, the hypothesis, $\underline{b}$ = 0, was rejected at the α = 0.1 level, an experimental detection rate of 0.83. Isolation results for the six alternaive sets considered earlier (Table V) are summarized in Table VI. Acceptance rates of incorrect alternative sets for this small number of trials are higher than predicted, but for each set, two of the false acceptances occurred in the trials in which the detection test failed to reject the $\underline{b}$ = 0 hypothesis. Thus, use of engineering judgment in these cases would help to avoid an incorrect conclusion; after failure of a detection test, the analyst should not conclude that a bias exists in any of the (falsely) isolated sets without a careful examination of the bias estimates, $\hat{\underline{b}}_J$, produced for each set.

TABLE VI

BIAS ISOLATION RESULTS FOR CASE 1

ALTERNATIVE SET (J)	NUMBER OF TRIALS FOR WHICH SET J WAS ACCEPTED (N_J)	ACCEPTANCE RATE (N_J/12)	PREDICTED PROBABILITY (P_J)
$\{\varepsilon_N, \varepsilon_E, \varepsilon_Z\}$	12	1.0	(1-α)
$\{\delta A_{VD(N,E)}\}$	3	0.24	0.13
$\{\delta V_{REF(N,E)}\}$	3	0.25	0.13
$\{\underline{\psi}\}$	4	0.33	0.13
$\{\underline{a}\}$	3	0.25	0.12
$\{\underline{\varepsilon}, \underline{\varepsilon}_f\}$	3	0.25	0.11

2. Case II: Two-Component Bias

This case is based on a two-component bias which is only marginally detectable (P_D = 0.52) and very easily mis-isolated ($P_J \tilde{=} 0.50$ for several sets which do not include the biased states). The hypothesis "$\underline{b}$ = 0" was rejected at the α = 0.1 level for only three of the 12 trials for this case; this 0.25 detection rate was poorer than the predicted value 0.52 but within reason because of the small sample size.

IV. SUMMARY AND EXTENSIONS

A formal, organized approach to validation of the statistical parameters of initialization errors in linear dynamic system models used by Kalman filter/smoothers has been presented. The approach for the mean-value (bias) parameters has been discussed in detail; a similar approach which has been developed for initial covariance matrix validation is discussed briefly in the following paragraph. The bias validation procedures include statistical hypothesis tests for detecting and isolating bias errors based on data processing results and capability analysis formulas for calculating probabilities of detecting and isolating (or mis-isolating) hypothesized model bias errors. All of the analysis procedures are based on a data equation representation of the overall system which transforms an initialization vector, $\underline{\varepsilon}$, into a smoothed estimate vector, $\hat{\underline{x}}^s_o$ (Fig. 2).

Analysis procedures for the covariance problem are also based on the data equation of Fig. 2, but normalization for this problem is done with respect to the covariance of $\underline{\varepsilon}'$ (process and measurement noise effects) rather than with respect to the covariance of $\hat{\underline{x}}^s_o$. Hypothesis tests for covariance matrix validation are based on quadratic forms in normal random variables. Another difference in the covariance problem is the way in which errors are scaled during capability analysis. Since a model error for this problem is of the form ΔI_o, the effect of this error on the system output vector is a change,

$\Delta\Sigma_f$, in the covariance matrix of the system's output. A variety of scalar measures might be used as norms on $\Delta\Sigma_f$; one which would be appropriate for weapon systems such as the example in Section III is the change in CEP due to $\Delta\Sigma_o$.

The procedures have potential application beyond the basic model validation problem. For example, if the model is viewed as describing an ideal system design, the procedures can be used as system parameter estimation tools. Since parameter estimates are generated from sufficient statistics for all the data, the algorithms provide an efficient method for compressing and storing the data generated by an ensemble of tests. For situations in which both bias and covariance errors exist in the system model, the algorithm can be used recursively to successively adjust estimates in the data equations as improved estimates of model parameters are generated.

APPENDIX A: RECURSIVE CALCULATION OF DATA EQUATION MATRICES

In this appendix, recursive equations are presented which can be used to compute matrices (D,R) associated with the smoothed estimate data equation for a single-phase model as discussed in Section II C. An outline of the derivation is also presented.

D,R FOR A SINGLE-PHASE MODEL

Let the linear discrete-time system model be represented for each $\ell = 0, 1, .., \ell_f$ by

$$\underline{x}_{\ell+1} = \phi_\ell \underline{x}_\ell + \underline{w}_\ell \ , \ \mathrm{Cov}\ (\underline{w}_\ell) = Q_\ell \tag{A-1}$$
$$\underline{x}_o = \underline{\varepsilon} \sim N\ (0, \Sigma_o)$$

$$\underline{z}_{\ell+1} = H_{\ell+1}\ \underline{x}_{\ell+1} + \underline{v}_{\ell+1} \ , \ \mathrm{Cov}\ (\underline{v}_{\ell+1}) = R_{v\ell+1} \tag{A-2}$$

Then matrices D_ℓ, R_ℓ such that

$$\hat{\underline{x}}_\ell = D_\ell\ \underline{x}_o + \underline{e}_\ell \qquad \ell = 0, 1, \ldots \tag{A-3}$$
$$R_\ell = \mathrm{Cov}\ (\hat{\underline{x}}_\ell) = \phi_\ell\ \Sigma_o\ D_\ell^T + \mathrm{Cov}\ (\underline{e}_\ell) \tag{A-4}$$

where $\hat{\underline{x}}_\ell$ is the optimal filtered estimate at time ℓ, can be computed from the formulas

$$D_{\ell+1} = (I - K_{\ell+1}\ H_{\ell+1})\ \phi_\ell\ D_\ell + K_{\ell+1}\ H_{\ell+1}\ \phi_{\ell+1}^* \tag{A-5}$$
$$D_o = 0$$

and

$$R_{\ell+1} = \phi_\ell\ R_\ell\ \phi_\ell^T + K_{\ell+1}\ \Psi_{\ell+1}\ K_{\ell+1}^T \tag{A-6}$$
$$R_o = 0$$

where

$$\phi_{\ell+1}^* = \phi(\ell+1,\ 0) = \phi_\ell\ \phi_{\ell-1} \cdots \phi_o \tag{A-7}$$

$$\Psi_{\ell+1} = H_{\ell+1}\ P_{\ell+1/\ell}\ H_{\ell+1}^T + R_{v\ell+1} \tag{A-8}$$

($K_{\ell+1}$ is the Kalman gain and $P_{\ell+1/\ell}$ is the one-step prediction error covariance.)

Equations (A-5), (A-6) can be derived by an induction procedure. The zero initial conditions are a result of the filter initialization

$$\hat{\underline{x}}_o = E\ [\underline{x}_o] = 0 \tag{A-9}$$

The induction procedure is based on the Kalman filter update formulas and uses the orthogonality property of an optimal filter

$$\mathrm{Cov}\ (\hat{\underline{x}}_{\ell+1},\ \underline{v}_{\ell+1}) = 0 \tag{A-10}$$

where

$$\underline{v}_{\ell+1} = \underline{z}_{\ell+1} - H_{\ell+1}\ \phi_\ell\ \hat{\underline{x}}_\ell \tag{A-11}$$

is the innovation at time $\ell+1$. The $\phi_{\ell+1}^*$ factor in Eq. (A-5) results from the following representation of the state at $\ell+1$,

$$\underline{x}_{\ell+1} = \phi_{\ell+1}^*\ \underline{x}_o + \underline{w}_{\ell+1}^* \tag{A-12}$$

where $\underline{w}^{*}_{\ell+1}$ is the cumulative effect of all process noise.

Now let Eqs. (A-1), (A-2) be replaced by an augmented state model:

$$\underline{x}^{a}_{\ell+1} = \phi^{a}_{\ell}\, \underline{x}^{a}_{\ell} + \underline{w}^{a}_{\ell} \tag{A-13}$$

$$\underline{x}^{a}_{o} = \begin{pmatrix} \underline{x}_o \\ \underline{x}_o \end{pmatrix}$$

$$\underline{z}^{a}_{\ell+1} = H^{a}_{\ell+1}\, \underline{x}^{a}_{\ell+1} + \underline{v}_{\ell+1} \tag{A-14}$$

where

$$\underline{x}^{a}_{\ell} = \begin{pmatrix} \underline{x}_\ell \\ \underline{x}_o \end{pmatrix} \quad , \quad \phi^{a}_{\ell} = \begin{pmatrix} \phi_\ell & 0 \\ 0 & I \end{pmatrix} \quad , \quad \underline{w}^{a}_{\ell} = \begin{pmatrix} \underline{w}_\ell \\ 0 \end{pmatrix}$$

and (A-15)

$$H^{a}_{\ell+1} = (H_{\ell+1} \quad 0)$$

A filtered estimate of this augmented-state system is therefore

$$\underline{\hat{x}}^{a}_{\ell} = \begin{pmatrix} \underline{\hat{x}}_\ell \\ \underline{\hat{x}}^{s}_{\ell} \end{pmatrix} \tag{A-16}$$

where $\underline{\hat{x}}^{s}_{\ell}$ is the optimal smoothed estimate of $\underline{x}_o$.

The data equation for the smoothed estimate $\underline{\hat{x}}^{s}_{\ell}$ is

$$\underline{\hat{x}}^{s}_{\ell} = D^{s}_{\ell}\, \underline{x}_o + \underline{e}'_{\ell} \quad , \quad \ell = 0,1,\ldots \tag{A-17}$$

where

$$D^{s}_{\ell+1} = D^{s}_{\ell} - K^{s}_{\ell+1}\, H_{\ell+1}\, (\phi_\ell\, D_\ell - \phi^{*}_{\ell+1}) \tag{A-18}$$
$$D^{s}_{o} = 0$$

Corresponding covariance matrices are

$$R^{s}_{\ell} = \mathrm{Cov}\,(\underline{\hat{x}}^{s}_{\ell}) \quad , \quad R^{c}_{\ell} = \mathrm{Cov}\,(\underline{\hat{x}}_\ell, \underline{\hat{x}}^{s}_{\ell}) \tag{A-19}$$

which satisfy recursions

$$R^{c}_{\ell+1} = \phi_\ell\, R^{c}_{\ell} + K_{\ell+1}\, \psi_{\ell+1}\, K^{s}_{\ell+1}{}^{T} \tag{A-20}$$
$$R^{c}_{o} = 0$$

$$R^{s}_{\ell+1} = R^{s}_{\ell} + K^{s}_{\ell+1}\, \psi_{\ell+1}\, K^{s}_{\ell+1}{}^{T} \tag{A-21}$$
$$R^{s}_{o} = 0$$

Equations (A-18) - (A-21) are derived directly from Eqs. (A-5), (A-6) by replacing each matrix by an augmented form, including

$$R^{a} = \begin{pmatrix} R & R^{c} \\ R^{cT} & R^{s} \end{pmatrix} \text{ and } D^{a} = \begin{pmatrix} D \\ D^{s} \end{pmatrix}, \tag{A-22}$$

and then equating corresponding partitions of the two equations.

Summarizing: for a single-phase system in which the smoothed data equation matrices (D^{s}, R^{s}) are required, the necessary recursive equations are (A-5), (A-18) and (A-21). Matrices R and R^{c} are not required for this case. The matrices denoted (D,R) in Section II are the smoothed data equation matrices (D^{s}_{ℓ}, R^{s}_{ℓ}) in this appendix, evaluated after the last update at time t_ℓ.

REFERENCES

1. Anderson, T.W., An Introduction to Multivariate Statistical Analysis, John Wiley & Sons, New York, 1958.

2. Rao, C.R., Linear Statistical Inference and its Application, John Wiley & Sons, New York, 1973.

3. Levy, L.J., Shumway, R.H., Olsen, D.E., and Deal, F.C. Jr., "Model Validation from an Ensemble of Kalman Filter Tests," Proc. of the 21st Midwest Symposium on Circuits and Systems, August 1978.

4. Sun, F.K., "An Alternative Approach for Maximum Likelihood Identification," Proc. of the 18th Conference on Decision and Control, Ft. Lauderdale, Florida, 1979.

5. Dempster, A.P., Laird, N.M., and Rubin, D.B., "Maximum Likelihood from Incomplete Data via the EM Algorithm," J. of the Royal Statistical Society, Vol. 39, November 1977.

6. Kashyap, R.L., and Rao, A.R., Dynamic Stochastic Models from Empirical Data, Mathematics in Science and Engineering, Vol. 122, Academic Press, New York 1976.

7. Goodrich, R.L., and Cains, P.E., "Linear System Identification from Nonstationary Cross-Sectional Data," IEEE Transactions on Automatic Control, Vol. AC-24, No. 3, June 1979.

8. Baram, Y., "Nonstationary Model Validation from Finite Data Records," IEEE Transactions on Automatic Control, Vol. AC-25, No. 1, February 1980.

9. Sun, F.K., "Statistical Inference Regarding Unknown Random Changes in Linear Dynamic Systems Using Cross-Sectional Data," to be published.

10. Pinson, J.C., "Inertial Guidance for Cruise Vehicles," in Guidance and Control of Aerospace Vehicles, C.T. Leondes, Ed. New York, McGraw-Hill, 1963, pp. 113-187.

11. Gelb, A., Ed., Applied Optimal Estimation, MIT Press, Cambridge, Massachusetts, 1974.

INERTIAL NAVIGATION SYSTEM ERROR MODEL CONSIDERATIONS IN KALMAN FILTER APPLICATIONS

by
James R. Huddle, Ph.D
Chief Scientist
Litton Guidance and Control Systems Division
Woodland Hills, California USA 91364

SUMMARY

This chapter develops the full linear model describing the propagation of error for inertial navigation systems employing the local-level, wander-azimuth navigation mechanization equations. The model applies to Schuler-tuned, space-stable or strapdown inertial system instrumentations. For this model, alternative approximate linear models are developed which in different operational applications have proved adequate as "design models" for the application of Kalman estimation theory.

INTRODUCTION

The application of modern linear estimation techniques requires an "adequate" model of the system whose states are to be estimated. The adequacy of a particular model of a physical system is usually determined by digital computer simulation. In these simulations as accurate a model as is obtainable is employed to represent the physical system and the subset of its states which are of interest in the application are estimated using a Kalman filter based on a "design model"[1] of this system. The adequacy of the design model is ascertained by observing how well the state estimates track the actual states of the simulated physical system using the root mean square (rms) difference between them as a criterion. Some designers also compare the standard deviations from the estimator covariance matrix to these rms values as an indicator of design adequacy. A more absolute judgement of estimator performance is obtainable by comparing the rms estimate errors with the standard deviations from the covariance matrix for the estimator based on the exact model of the simulated system, the so-called "real-world" model covariance analysis approach. To economize on mechanization requirements it is usually desirable to simplify the latter complete estimator design model, motivating the examination of various simplified candidate design models.

Regardless of the estimator design criterion employed, the work of estimator design involves a "tuning" process in which states are added or deleted, dynamic intercouplings are changed and white noise components are altered. The subject of this chapter deals with an aspect of this latter process for the case of inertial navigation systems. Herein it is shown that the analytical model for propagation of error in the navigation variables of position, velocity and attitude have alternative forms with various degrees of approximation. The model differences are important in that they imply different real-time digital computer mechanization requirements in terms of memory and duty cycle. These models have been employed in operational systems and have been proved effective for differing application requirements.

LOCAL-LEVEL COORDINATE SYSTEM NAVIGATION EQUATIONS

The process of terrestrial navigation using inertial equipment involves the measurement of force by a usually orthogonal triad of accelerometers whose orientation relative to the earth is established and maintained by three usually orthogonal gyroscopic axes. Since the orientation of the accelerometers relative to the earth is known, the force due to gravity can be removed from these measurements using an analytical model, to obtain the acceleration of the inertial system center of mass relative to inertial space. Correction of these measurements obtained or expressed in some reference navigation coordinate frame for Coriolis accelerations due to the effects of earth and reference frame rotation rate relative to inertial space, yields the rate of change of system velocity relative to the earth with respect to the reference navigation coordinate frame. Integration of these variables with proper initialization, then yields the velocity of the inertial system relative to the earth. Transformation of the velocity components to an earth fixed frame and subsequent integration yields system position change relative to the earth thus accomplishing the navigation objective. To make these statements more specific, the navigation equations that are mechanized for most aircraft inertial navigation systems assume as the navigation reference coordinate frame $\{x,y,z\}$, a local-level system with "wander azimuth angle" α, as illustrated in Figure 1. This coordinate frame resides at the center of mass of the inertial system and is maintained in the local-level orientation as the system is moved relative to the earth. Alternative mechanizations for the behavior of the wander angle are possible. For example if $\alpha(t) \equiv 0$ for all t, then the navigation equations correspond to the north-slaved mechanization since the y-axis is always directed northward. In this case the x and y axes are coincident with the local east and north axes while the z axis still remains coincident with the local vertical.

COORDINATE FRAMES FOR ERROR MODEL DEVELOPMENT

The local-level coordinate navigation equations that are implemented in the real-time digital computer have been derived in detail elsewhere[2] and are summarized in extended form in the appendix to be readily employed below. The development of the describing error equations presented here is more conceptual than mathematically rigorous to simplify presentation of the material. The development is facilitated by the introduction of two additional orthogonal coordinate systems to that depicted in Figure 1. The frame shown there is hereafter referred to as the reference frame and is local-level at the true position of the inertial system with the azimuth angle, α.

PLATFORM FRAME

The first additional frame is called the platform frame and is slightly misaligned from the reference frame via small attitude error angles as defined by the skew-symmetric transformation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p = [I + \phi] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1}$$

where:*

$$\phi = \begin{bmatrix} 0 & \phi_z & -\phi_y \\ -\phi_z & 0 & \phi_x \\ \phi_y & -\phi_x & 0 \end{bmatrix}$$

where the angles are positive counter-clockwise about their respective axes, the variables $\phi_{x,y}$ representing tilt of the platform coordinates and the azimuth variable ϕ_z, representing the difference:

$$\phi_z \triangleq \alpha_p - \alpha \tag{2}$$

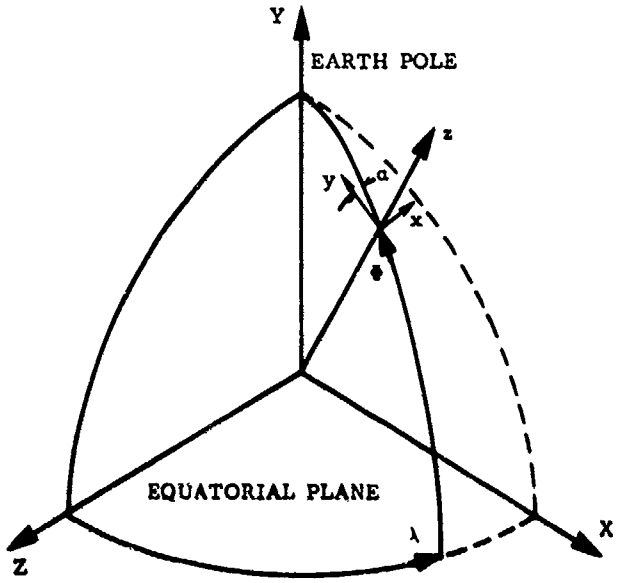


Figure 1. Illustration of the Local-Level Navigation Reference Coordinate Frame [x,y,z] and an Earth-Fixed [X,Y,Z] Reference Coordinate Frame.

SYSTEM COMPUTED LOCAL GEODETIC FRAME

The second orthogonal coordinate frame used in the error description is called the system computed local geodetic frame and differs in angular orientation from the reference coordinate frame by three small, independent non-orthogonal rotations due to errors in system computed geodetic position and the wander angle:

$\delta\phi \triangleq \phi_c - \phi = -\delta\theta_E$ is the error in system computed geodetic latitude positive clockwise about the local east axis. Note $\delta\theta_E$ is positive counter-clockwise about the east axis

$\delta\lambda \triangleq \lambda_c - \lambda$ is the error in system computed longitude positive counterclockwise about the earth's polar axis. Note this error can be projected onto the local north and vertical axes knowing the system latitude as:

$$\delta\theta_N \triangleq \delta\lambda \cos\phi$$
$$\delta\theta_V \triangleq \delta\lambda \sin\phi \tag{3}$$

$\delta\alpha \triangleq \alpha_c - \alpha$ is the difference between the system computed wander angle and that of the reference coordinate system.

The three sources of angular rotation can be expressed about the reference coordinate axes knowing the wander angle as:

*The notation convention in this chapter normally identifies scalar quantities with subscripts except where the text defines vector quantities. Brackets [ ], are employed to identify matrices which are all defined explicitly in the text. Variables with no subscripts are normally vectors which are defined explicitly in the text and where relevant the text indicates the coordinate system in which the vector is assumed to be expressed.

$$\delta\theta_x \triangleq \delta\theta_E \cos\alpha + \delta\theta_N \sin\alpha$$
$$\delta\theta_y \triangleq \delta\theta_N \cos\alpha - \delta\theta_E \sin\alpha \quad (4)$$
$$\delta\theta_z \triangleq \delta\lambda \sin\phi + \delta\alpha$$

These three rotations describe completely the difference in orientation of a coordinate system described by the direction cosine matrix $[D]_c$ defined in the appendix, based upon the computed latitude, longitude and wander angle $[\phi, \lambda, \alpha]_c$ of a local-level coordinate frame relative to the reference local-level coordinate frame:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_c = [I + \delta\theta] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5)$$

where:

$$[\delta\theta] \triangleq \begin{bmatrix} 0 & \delta\theta_z & -\delta\theta_y \\ -\delta\theta_z & 0 & \delta\theta_x \\ \delta\theta_y & -\delta\theta_x & 0 \end{bmatrix} \quad (6)$$

ALTERNATE INSTRUMENT MECHANIZATIONS OF INERTIAL SYSTEMS

Several mechanizations can be employed for obtaining the inertial instrument measurements. Regardless of the instrumentation approach employed, the local-level coordinate system formulation of the navigation mechanization equations can be employed to accomplish the navigation function as long as the accelerometer measurements can be expressed along the local-level coordinate axes. In all mechanizations the attitude error ϕ, between the platform frame and the local-level frame is minimized prior to use of the inertial system for navigation by a process called alignment.

SCHULER-TUNED PLATFORM MECHANIZATION

The Schuler-tuned platform mechanization of the inertial instrumentation has been employed for several decades. In this approach it is attempted to maintain the accelerometer sensing axes coincident with the local-level frame by appropriate precessing of the system gyros. The commonly mounted gyros and accelerometers comprise what is called the stable element which is isolated from the angular motion of the carrying vehicle by a gimbal set. Once initial alignment of the accelerometer sensing axes with the local-level frame coordinates has been achieved, this orientation is maintained by precessing the gyros, relative to inertial space, by the system computed spatial rate ω_c, of the local-level frame. This term is the sum of the system computed angular rate of the local-level frame relative to the earth ρ_c, and the system computed rotation rate of the earth relative to inertial space, Ω_c:

$$\omega_c = \rho_c + \Omega_c \quad (7)$$

The platform frame in this mechanization is the orientation defined by the accelerometer sensing axes which are ideally maintained coincident with the local-level reference frame. Consequently the angular rate of the platform coordinate frame relative to the instrument sensing axes expressed in the local-level frame ν_c, is mechanized as zero.

STRAPDOWN AND SPACE-STABLE INERTIAL INSTRUMENT MECHANIZATIONS

More generally as in the case of an alternate gimbaled inertial system or a strapdown inertial system mechanization, the platform coordinate frame is a computed orientation relative to that where the accelerometer axes actually exist, that is determined using the inertial instrument measurements. In the error-less case the platform frame is again of course coincident with the local-level reference frame. In a strapdown system where there is no gimbal set and the inertial instruments are "strapped" to the carrying vehicle frame, the transformation matrix between the platform frame and the instrument frame is computed using the system computed spatial rate of the local-level frame ω_c, minus the gyro measurements of angular rate of the vehicle frame relative to inertial space ω_g, both expressed in the local-level frame:

$$\nu_c = \omega_c - [P]_c\, \omega_g \quad (8)$$

For an alternate gimbaled inertial system such as space-stable, where the instruments remain fixed relative to inertial space, the platform to instrument frame transformation matrix is obtained as the system computed spatial angular rate, expressed in the local-level frame ω_c:

$$\nu_c = \omega_c \quad (9)$$

since the accelerometer axes are presumed fixed relative to inertial space.

In all the above mechanizations the linear transformation or direction cosine matrix from the accelerometer coordinate frame $[x,y,z]_a$ to the platform frame can be computed once initialization by alignment has been achieved, by integration of the matrix differential equation:

$$[\dot{P}]_c = [\nu]_c\, [P]_c \quad (10)$$

where the anti-symmetric matrix of relative angular rates of the platform frame relative to the instrument frame in platform axes is defined as:

$$[\nu]_c \triangleq \begin{bmatrix} 0 & \nu_z & -\nu_y \\ -\nu_z & 0 & \nu_x \\ \nu_y & \nu_x & 0 \end{bmatrix}_c = \begin{cases} 0 & \text{Schuler-Tuned Platform} \\ [\omega_c - [P]_c\ \omega_g] & \text{Strapdown System} \\ [\omega]_c & \text{Space-Stable Platform} \end{cases}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p = [P]_c \begin{bmatrix} x \\ y \\ z \end{bmatrix}_a \tag{11}$$

$$[P]_c \triangleq \begin{bmatrix} \langle x_p, x_a \rangle & \langle x_p, y_a \rangle & \langle x_p, z_a \rangle \\ \langle y_p, x_a \rangle & \langle y_p, y_a \rangle & \langle y_p, z_a \rangle \\ \langle z_p, x_a \rangle & \langle z_p, y_a \rangle & \langle z_p, z_a \rangle \end{bmatrix}_c$$

Using the three coordinate systems defined above, we may now develop a set of ten linear differential equations describing the propagation of error for the local-level coordinate system navigation equations and the three different inertial instrument implementations just discussed. These ten equations describe the time rate of change of velocity difference δV, attitude error ϕ, position error $\delta\theta$, and elevation error, δh.

ERROR IN THE SYSTEM COMPUTED VELOCITY

The rate of change of velocity difference is obtained by first differencing the representation of the mechanized version of this equation employing actual values in the system computer, (11A) of the appendix, with the representation of the acceleration equation for the local-level reference navigation frame, (1A) of the appendix. Since the accelerometer measurements are expressed in the platform coordinate frame, their difference from the accelerometer measurements expressed in the reference frame due to the effects of attitude error is:

$$\delta a \triangleq A_p - A = [\phi]\ A \tag{12}$$

The system computed Coriolis acceleration components are all determined from variables which reside in registers in the system computer. These system computed values differ from the values of the same variables that correspond to the reference navigation coordinate frame as defined below:

$$\delta V \triangleq V_c - V = \begin{bmatrix} \delta V_x \\ \delta V_y \\ \delta V_z \end{bmatrix} \tag{13}$$

$$\delta \rho \triangleq \rho_c - \rho = \begin{bmatrix} \delta \rho_x \\ \delta \rho_y \\ \delta \rho_z \end{bmatrix} \tag{14}$$

$$\delta \Omega \triangleq \Omega_c - \Omega = \begin{bmatrix} \delta \Omega_x \\ \delta \Omega_y \\ \delta \Omega_z \end{bmatrix} \tag{15}$$

which upon examination of Equations (7A, 12A) ignoring the small effects of error in the computed radii of curvature, yields:

$$\begin{aligned} \delta \rho_x &= -\delta V_y \cdot R^{-1} \\ \delta \rho_y &= \delta V_x \cdot R^{-1} \\ \delta \rho_z &= \delta \dot{\alpha} + \delta \rho_V \\ \delta \rho_V &\triangleq \rho_{N_c} \tan \phi_c - \rho_N \tan \phi = \delta \rho_N \tan \phi + \rho_N \cdot \delta\phi \cdot \sec^2 \phi \\ \delta \rho_N &= \delta \rho_x \sin \alpha + \delta \rho_y \cos \alpha + \rho_E \cdot \delta \alpha \end{aligned} \tag{16}$$

where to first order R can be assumed to be a nominal radius for the earth.

Further to first-order:

$$\begin{aligned}\delta\Omega_x &= -\,\Omega \sin\phi \sin\alpha\delta\phi + \Omega\cos\phi\cos\alpha\delta\alpha\\ \delta\Omega_y &= -\,\Omega \sin\phi \cos\alpha\delta\phi - \Omega\cos\phi\sin\alpha\delta\alpha\\ \delta\Omega_z &= \Omega\cos\phi\,\delta\phi\end{aligned} \tag{17}$$

which with (4) can be written as:

$$\begin{aligned}\delta\Omega_x &= \Omega_y \cdot \delta\theta_z - \Omega_z \cdot \delta\theta_y\\ \delta\Omega_y &= \Omega_z \cdot \delta\theta_x - \Omega_x \cdot \delta\theta_z\\ \delta\Omega_z &= \Omega_x \cdot \delta\theta_y - \Omega_y \cdot \delta\theta_x\end{aligned} \tag{18}$$

or:

$$\delta\Omega = [\delta\theta]\,\Omega = \Omega \times \delta\theta \tag{19}$$

Combining the above equations, the difference between (11A) and (1A) to first-order is:

$$\delta\dot{V} \triangleq \dot{V}_c - \dot{V} = \begin{bmatrix}\delta\dot{V}_x\\ \delta\dot{V}_y\\ \delta\dot{V}_z\end{bmatrix} \tag{20}$$

$$\approx [\phi]\,A + [\delta\rho + 2\delta\Omega]\,V - [\delta V]\,(\rho + 2\Omega) + \delta\gamma + \nabla$$

where the following skew-symmetric matrix definitions have been employed:

$$[\delta\rho + 2\delta\Omega] \triangleq \begin{bmatrix} 0 & [\delta\rho + 2\delta\Omega]_z & -[\delta\rho + 2\delta\Omega]_y\\ -[\delta\rho + 2\delta\Omega]_z & 0 & [\delta\rho + 2\delta\Omega]_x\\ [\delta\rho + 2\delta\Omega]_y & -[\delta\rho + 2\delta\Omega]_x & 0\end{bmatrix}$$

$$[\delta V] \triangleq \begin{bmatrix} 0 & \delta V_z & -\delta V_y\\ -\delta V_z & 0 & \delta V_x\\ \delta V_y & -\delta V_x & 0\end{bmatrix} \tag{21}$$

and:

$$\delta\gamma = \begin{bmatrix}\delta\gamma_x\\ \delta\gamma_y\\ \delta\gamma_z\end{bmatrix} = \gamma - \gamma_c$$

represents the error in the computation of the gravity vector from the analytical model due solely to error in system computed geodetic position and elevation. If an ellipsoidal equipotential surface is assumed for the gravity model $\delta\gamma_x = \delta\gamma_y = 0$, and $\delta\gamma_z$ is in error due only to error in system computed latitude and elevation. (22)

$$\nabla = \begin{bmatrix}\nabla_x\\ \nabla_y\\ \nabla_z\end{bmatrix}$$

represents generalised accelerometer measurement error due to all instrument related error sources transformed onto the reference local-level coordinate frame plus the difference between actual gravity and that represented by the gravity model. Note if an ellipsoidal equipotential surface is assumed for the gravity model, error exists along the level axes due to the deflection of the vertical to the geoid relative to that of the ellipsoid[3]. The error exists along the vertical due to model error for the intensity of the gravity.

At this point some remarks should be made about the meaning of the various quantities defined above. Generally the definition of "error" that has been selected stands for the simple difference between the value of the variable as it physically exists in the system computer and the value of the variable as interpreted in the reference local-level navigation coordinate system when determined without error. For the case of earth rate, the error (18) corresponds to the difference between the earth rate determined in the computer coordinates and that for the reference coordinates. For velocity error (13, 20), this is not the case as by construction the velocity error which arises from the integration of Equation (20) results from the transformation error ϕ, between the platform and reference coordinate systems, the errors in the computed Coriolis acceleration and gravity vector components and of course any measurement errors associated with the accelerometers.

If as occurs in some applications, it is desired to express the acceleration measurement in an earth-fixed coordinate frame prior to integration, as [X,Y,Z] shown in Figure 1, then the total transformation error must include the effects of the position error $\delta\theta$, of (4) since the measurements A_p, must be transformed through the system computed direction cosine matrix $[D]_c$ of the appendix, to the earth-fixed frame. In this case however, the mechanization equations which are integrated to compute the system velocity are different from (1A) as the earth-fixed frame in which the integration is performed is obviously not rotating with respect to the earth as is the local-level navigation frame. The correct mechanization equation for the rate of change of inertial system velocity relative to the earth with respect to earth-fixed coordinates is, in

error-free vector form:

$$\dot{V}_e = A_e - 2\Omega_e \times V_e - \gamma_e \tag{23}$$

where:

$\dot{V}_e \triangleq \begin{bmatrix} \dot{V}_X \\ \dot{V}_Y \\ \dot{V}_Z \end{bmatrix}$ denotes the rate of change of velocity relative to the earth with respect to the earth-fixed frame which when integrated in the earth-fixed frame yields system velocity with respect to the earth in the earth-fixed frame

$A_e \triangleq \begin{bmatrix} A_X \\ A_Y \\ A_Z \end{bmatrix}$ is the specific force measured by the accelerometers due to the true system acceleration and the modeled gravity at the true system position, expressed in the earth-fixed frame

$\Omega_e \triangleq \begin{bmatrix} 0 \\ \Omega \\ 0 \end{bmatrix}$ is the earth rotation rate expressed in the earth-fixed frame of Figure 1

$\gamma_e \triangleq \begin{bmatrix} \gamma_X \\ \gamma_Y \\ \gamma_Z \end{bmatrix}$ is the projection of the modeled gravity vector at the true system position onto the earth-fixed frame:

One mechanized version of (23) that would be integrated in the system computer is:

$$\dot{V}_{e_c} = [D]_c^T A_p - 2\Omega_e \times V_{e_c} - \gamma_{e_c} \tag{24}$$

where all components are expressed in the earth-fixed coordinate system prior to integration. Note the earth rate components are known exactly in the earth-fixed frame, but the direction and magnitude of the gravity vector components relative to these coordinates are a function of the system computed latitude, longitude and elevation and hence subject to the errors in these variables. The equation for the difference between (23) and (24) is then to first order:

$$\delta\dot{V}_e \triangleq \dot{V}_{e_c} - \dot{V}_e \approx [D]^T [\phi] A - 2[\delta V_e] \Omega_e + [D]^T ([\delta\theta]\gamma + \delta\gamma) + \nabla_e \tag{25}$$

where relative to the local-level reference frame:

$$[\phi] \triangleq \begin{bmatrix} 0 & (\phi - \delta\theta)_z & -(\phi - \delta\theta)_y \\ -(\phi - \delta\theta)_z & 0 & (\phi - \delta\theta)_x \\ (\phi - \delta\theta)_y & -(\phi - \delta\theta)_x & 0 \end{bmatrix} \tag{26}$$

represents angular error incurred in the transformation of the accelerometer measurements from the platform frame to an earth-fixed frame. Here, as $\delta\theta$ is small, we have:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \approx [I - \delta\theta] \begin{bmatrix} x \\ y \\ z \end{bmatrix}_c \tag{27}$$

Note that the computer frame is not an earth-fixed frame but related to an earth-fixed frame through the computed position and wander angle $[\phi, \lambda, \alpha]_c$. Further:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p \approx [I + \phi][I - \delta\theta] \begin{bmatrix} x \\ y \\ z \end{bmatrix}_c \approx [I + \phi] \begin{bmatrix} x \\ y \\ z \end{bmatrix}_c \tag{28}$$

ERROR IN THE SYSTEM COMPUTED POSITION AND ELEVATION

In that the direction cosine matrix $[D]_c$ defines the system computed latitude, longitude and wander angle, the perturbation of this matrix due to error in these three independent variables is of interest. Directly from the definition (5A), the following perturbations relative to $[\delta\lambda, \delta\phi, \delta\alpha]$ can be obtained which upon employing the definition (5A,4) yields not surprisingly the second equivalence:

$$
\begin{aligned}
\delta\langle X,x\rangle &= -\delta\lambda\,(c\alpha\, s\lambda + s\alpha\, s\phi\, c\lambda) \\
&\quad -\delta\alpha\,(s\alpha\, c\lambda + c\alpha\, s\phi\, s\lambda) \quad = \delta\theta_z \cdot \langle X,y\rangle - \delta\theta_y \cdot \langle X,z\rangle \\
&\quad -\delta\phi\,(s\alpha\, c\phi\, s\lambda) \\
\delta\langle X,y\rangle &= \delta\lambda\,(s\alpha\, s\lambda - c\alpha\, s\phi\, c\lambda) \\
&\quad +\delta\alpha\,(s\alpha\, s\phi\, s\lambda - c\alpha\, c\lambda) \quad = \delta\theta_x \cdot \langle X,z\rangle - \delta\theta_z \cdot \langle X,x\rangle \\
&\quad -\delta\phi\,(c\alpha\, c\phi\, s\lambda) \\
\delta\langle X,z\rangle &= -\delta\phi\,(c\phi\, s\lambda) + \delta\lambda\,(c\phi\, c\lambda) = \delta\theta_y \cdot \langle X,x\rangle - \delta\theta_x \cdot \langle X,y\rangle \\
\delta\langle Y,x\rangle &= -\delta\alpha\,(c\alpha\, c\phi) - \delta\phi\,(s\alpha\, s\phi) = \delta\theta_z \cdot \langle Y,y\rangle - \delta\theta_y \cdot \langle Y,z\rangle \\
\delta\langle Y,y\rangle &= -\delta\alpha\,(s\alpha\, c\phi) - \delta\phi\,(c\alpha\, s\phi) = \delta\theta_x \cdot \langle Y,z\rangle - \delta\theta_z \cdot \langle Y,z\rangle \\
\delta\langle Y,z\rangle &= \delta\phi\, c\phi \qquad\qquad\qquad = \delta\theta_y \cdot \langle Y,x\rangle - \delta\theta_x \cdot \langle Y,y\rangle \\
\delta\langle Z,x\rangle &= \delta\lambda\,(s\alpha\, s\phi\, s\lambda - c\alpha\, s\lambda) \\
&\quad +\delta\alpha\,(s\alpha\, s\lambda - c\alpha\, s\phi\, c\lambda) \quad = \delta\theta_z \cdot \langle Z,y\rangle - \delta\theta_y \cdot \langle Z,z\rangle \\
&\quad -\delta\phi\,(s\alpha\, c\phi\, c\lambda) \\
\delta\langle Z,y\rangle &= \delta\lambda\,(s\alpha\, c\lambda + c\alpha\, s\phi\, s\lambda) \\
&\quad +\delta\alpha\,(c\alpha\, s\lambda + s\alpha\, s\phi\, c\lambda) \quad = \delta\theta_x \cdot \langle Z,z\rangle - \delta\theta_z \cdot \langle Z,x\rangle \\
&\quad -\delta\lambda\,(c\alpha\, c\phi\, c\lambda) \\
\delta\langle Z,z\rangle &= -\delta\lambda\,(c\phi\, s\lambda) - \delta\phi\,(s\phi\, c\lambda) = \delta\theta_y \cdot \langle Z,x\rangle - \delta\theta_x \cdot \langle Z,y\rangle
\end{aligned}
\tag{29}
$$

where (s,c) denote (sine, cosine) respectively. Clearly:

$$[\delta D] \triangleq [D]_c - [D] \approx [\delta\theta]\,[D] \tag{30}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [D_c - \delta D] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Thus as stated above, we have confirmed that the perturbation vector angle $\delta\theta$, represents a rotation from the local-level reference frame to the computed geodetic frame. Proceeding formally we may differentiate the first-order approximation in (30) to obtain:

$$[\dot{\delta D}] = [\dot{\delta\theta}]\,[D] + [\delta\theta]\,[\dot{D}] \tag{31}$$

which from (4A):

$$[\dot{D}] = [\rho]\,[D] \tag{32}$$

yields for (31):

$$[\dot{\delta D}] = ([\dot{\delta\theta}] + [\delta\theta]\,[\rho])\,[D] \tag{33}$$

and further upon differentiation of (30) and using definitions we obtain to first-order:

$$[\dot{\delta D}] = [\delta\rho]\,[D] + [\rho]\,[\delta D] \triangleq [\dot{D}]_c - [\dot{D}] \tag{34}$$

where:

$$[\delta\rho] \triangleq [\rho]_c - [\rho] \tag{35}$$

which using (30) yields:

$$[\dot{\delta D}] = ([\delta\rho] + [\rho]\,[\delta\theta])\,[D] \tag{36}$$

Note $\delta\rho$ denotes the difference in the angular rate of the system computed geodetic frame and the local-level frame both taken with respect to the earth frame, whereas $\dot{\delta\theta}$ denotes the angular rate of change of the system computed geodetic frame taken with respect to the local-level geodetic frame. From Equations (31, 36) since [D] is invertible, we obtain:

$$[\dot{\delta\theta}] = [\delta\rho] + [\rho]\,[\delta\theta] - [\delta\theta]\,[\rho] \tag{37}$$

or in scalar form upon expansion:

$$\delta\dot{\theta}_x = \delta\rho_x + \rho_z\delta\theta_y - \rho_y\delta\theta_z$$
$$\delta\dot{\theta}_y = \delta\rho_y + \rho_x\delta\theta_z - \rho_z\delta\theta_x \tag{38}$$
$$\delta\dot{\theta}_z = \delta\rho_z + \rho_y\delta\theta_x - \rho_x\delta\theta_y$$

or in vector form:

$$\delta\dot{\theta} = \delta\rho - \rho \times \delta\theta \tag{39}$$

The form (39) merely indicates that the rate of change of angular error of the computed geodetic frame relative to the local-level reference frame $\delta\dot{\theta}$, is simply the sum of the rate of change of this angular error relative to the earth $\delta\rho$ or $\delta\dot{\theta}_e$, and the Coriolis effect due to the rotation rate between the computed geodetic frame and the earth. The second term in (39) can also be viewed as the error in resolving the correct relative angular rate vector ρ, from the earth-fixed frame to the local-level reference frame due to the error $\delta\theta$, in a knowledge of its orientation when the system computed latitude, longitude and wander-angle are employed.

The final position error equation is that for the elevation of the system relative to the earth. Elevation error is obtained by integrating the difference between the system computed and actual system velocity relative to the earth as projected onto the local vertical axis:

$$\dot{\delta h}_c = V_{z_c} - V_z \triangleq \delta V_z \tag{40}$$

ERROR IN THE PLATFORM FRAME ORIENTATION

To derive the dynamics of the attitude error ϕ, of the platform frame relative to the local-level reference frame we can proceed in a manner analogous to that for angular position error above. First the difference between the system computed platform frame to instrument frame transformation $[P]_c$, and the local-level to instrument frame transformation $[P]$, is derived:

$$[\delta P] \triangleq [P]_c - [P] = [\phi]\,[P] \tag{41}$$

where:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [P] \begin{bmatrix} x \\ y \\ z \end{bmatrix}_a$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p = [I + \phi] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{42}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p \triangleq [P]_c \begin{bmatrix} x \\ y \\ z \end{bmatrix}_a$$

have been used. Differentiation of (41) yields:

$$[\delta\dot{P}] = ([\dot{\phi}] + [\phi]\,[v])\,[P] \tag{43}$$

where $[\dot{\phi}]$ is expressed and differentiated with respect to the local-level frame. Further

$$[\delta\dot{P}] = [\dot{P}]_c - [\dot{P}] = [v]_c\,[P]_c - [v]\,[P] \tag{44}$$
$$= ([\delta v] + [v]\,[\phi])\,[P]$$

where:

$$[\delta v] \triangleq [v]_c - [v] \tag{45}$$

is the difference between the angular rate of the system computed platform frame with respect to the instrument frame expressed in the platform frame $[v]_c$, and the angular rate of the local level frame with respect to the instrument frame $[v]$ expressed in the local-level frame. Equating (43, 44) using the invertibility of $[P]$ yields:

$$[\dot{\phi}] = [\delta v] + [v]\,[\phi] - [\phi]\,[v] \tag{46}$$

which yields the vector form:

$$\dot{\phi} = \delta v + \phi \times v \tag{47}$$

It is of interest to specify (47) for the three instrumentation mechanizations discussed above: Schuler-tuned, space-stable and strapdown.

In the case of a Schuler-tuned mechanization, the platform frame is the instrument frame, hence v_c is zero. However the instrument frame rotates relative to the local-level frame as the gyros are processed at

$$\delta\dot{\theta}_x = \delta\rho_x + \rho_z\delta\theta_y - \rho_y\delta\theta_z$$
$$\delta\dot{\theta}_y = \delta\rho_y + \rho_x\delta\theta_z - \rho_z\delta\theta_x \qquad (38)$$
$$\delta\dot{\theta}_z = \delta\rho_z + \rho_y\delta\theta_x - \rho_x\delta\theta_y$$

or in vector form:

$$\delta\dot{\theta} = \delta\rho - \rho \times \delta\theta \qquad (39)$$

The form (39) merely indicates that the rate of change of angular error of the computed geodetic frame relative to the local-level reference frame $\delta\dot{\theta}$, is simply the sum of the rate of change of this angular error relative to the earth $\delta\rho$ or $\delta\dot{\theta}_e$, and the Coriolis effect due to the rotation rate between the computed geodetic frame and the earth. The second term in (39) can also be viewed as the error in resolving the correct relative angular rate vector ρ, from the earth-fixed frame to the local-level reference frame due to the error $\delta\theta$, in a knowledge of its orientation when the system computed latitude, longitude and wander-angle are employed.

The final position error equation is that for the elevation of the system relative to the earth. Elevation error is obtained by integrating the difference between the system computed and actual system velocity relative to the earth as projected onto the local vertical axis:

$$\dot{\delta h}_c = V_{z_c} - V_z \triangleq \delta V_z \qquad (40)$$

ERROR IN THE PLATFORM FRAME ORIENTATION

To derive the dynamics of the attitude error ϕ, of the platform frame relative to the local-level reference frame we can proceed in a manner analogous to that for angular position error above. First the difference between the system computed platform frame to instrument frame transformation $[P]_c$, and the local-level to instrument frame transformation $[P]$, is derived:

$$[\delta P] \triangleq [P]_c - [P] = [\phi]\,[P] \qquad (41)$$

where:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [P] \begin{bmatrix} x \\ y \\ z \end{bmatrix}_a$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p = [I + \phi] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad (42)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p \triangleq [P]_c \begin{bmatrix} x \\ y \\ z \end{bmatrix}_a$$

have been used. Differentiation of (41) yields:

$$[\delta\dot{P}] = ([\dot{\phi}] + [\phi]\,[v])\,[P] \qquad (43)$$

where $[\dot{\phi}]$ is expressed and differentiated with respect to the local-level frame. Further

$$[\delta\dot{P}] = [\dot{P}]_c - [\dot{P}] = [v]_c\,[P]_c - [v]\,[P] \qquad (44)$$
$$= ([\delta v] + [v]\,[\phi])\,[P]$$

where:

$$[\delta v] \triangleq [v]_c - [v] \qquad (45)$$

is the difference between the angular rate of the system computed platform frame with respect to the instrument frame expressed in the platform frame $[v]_c$, and the angular rate of the local level frame with respect to the instrument frame $[v]$ expressed in the local-level frame. Equating (43, 44) using the invertibility of $[P]$ yields:

$$[\dot{\phi}] = [\delta v] + [v]\,[\phi] - [\phi]\,[v] \qquad (46)$$

which yields the vector form:

$$\dot{\phi} = \delta v + \phi \times v \qquad (47)$$

It is of interest to specify (47) for the three instrumentation mechanizations discussed above: Schuler-tuned, space-stable and strapdown.

In the case of a Schuler-tuned mechanization, the platform frame is the instrument frame, hence v_c is zero. However the instrument frame rotates relative to the local-level frame as the gyros are precessed at

the computed spatial rate ω_c, applied about the platform frame coordinates as opposed to the local-level frame coordinates. The instrument frame also rotates relative to the local-level frame due to the generalized drift rate of the gyros ε, hence:

$$\delta v = \delta\omega + \phi \times \omega + \varepsilon = \dot{\phi} \tag{48}$$

where $\delta\omega$ denotes the scalar difference of the components of the vectors ω_c and ω:

$$\delta\omega \triangleq \begin{bmatrix} \omega_{x_c} - \omega_x \\ \omega_{y_c} - \omega_y \\ \omega_{z_c} - \omega_z \end{bmatrix} = \delta\rho + \delta\Omega \tag{49}$$

and $\phi \times \omega$ expresses the rotation rate of the platform frame relative to the local-level frame when the local-level spatial angular rate components are applied in the misaligned platform frame even if they are perfectly known.

For the space-stable mechanization:

$$v_c \triangleq \omega_c \tag{50}$$

whereas the angular rate of the local-level frame relative to the instrument frame is:

$$v = \omega - \varepsilon \tag{51}$$

where the components of v are about the local-level frame coordinates. Hence for the space-stable mechanization using (47, 50, 51), we have

$$\dot{\phi} = \delta\omega + \phi \times \omega + \varepsilon \tag{52}$$

For the strapdown mechanization:

$$v_c \triangleq \omega_c - [P]_c\, \dot{\theta}_g \tag{53}$$

whereas the angular rate of the local-level frame relative to the instrument frame is:

$$v = \omega - [P]\, \dot{\theta} \tag{54}$$

where we define generalized gyro drift rate about the instrument axes ε_g, as the difference between the actual frame rotation rate relative to inertial space, $\dot{\theta}$, and the gyro output measurements, $\dot{\theta}_g$:

$$\varepsilon_g \triangleq \dot{\theta} - \dot{\theta}_g \tag{55}$$

yielding the generalized drift rate about local-level coordinates as:

$$\varepsilon = [P]\, \varepsilon_g \tag{56}$$

Hence differencing (53, 54) using (41) and (55, 56), we have to first order:

$$\delta v = \delta\omega - [\phi]\, [P]\, \dot{\theta} + \varepsilon \tag{57}$$

Consequently using (47, 51, 57) we have

$$\dot{\phi} = \delta\omega - [\phi]\, [P]\, \dot{\theta} + \varepsilon + \phi \times (\omega - [P]\, \dot{\theta})$$

which as:

$$[\phi]\, [P]\, \dot{\theta} = -\phi \times [P]\, \dot{\theta} \tag{58}$$

yields:

$$\dot{\phi} = \delta\omega + \phi \times \omega + \varepsilon \tag{59}$$

to describe the dynamics of the attitude error for the strapdown instrument mechanization. We note in summary that regardless of the three mechanizations employed, the dynamics of the attitude error propagation (48, 52, 59) are identical in form. Note however the generalized drift rate vector ε, employed in these equations is obtained by transforming the drift rate about instrument axes to the local-level frame coordinate axes.

ANGULAR ROTATION BETWEEN THE PLATFORM AND SYSTEM COMPUTED GEODETIC FRAMES

The dynamics of the angular rotation from the system computed geodetic frame to the platform frame:

$$\psi \triangleq \phi - \delta\theta \tag{60}$$

relative to the local-level frame, can be obtained by differencing the corresponding individual rates as derived above (39, 59):

$$\dot{\psi} \triangleq \dot{\phi} - \delta\dot{\theta} = \delta\rho - \delta\theta \times \Omega + \phi \times \omega + \varepsilon - \delta\rho - \delta\theta \times \rho \tag{61}$$

or:

$$\dot{\psi} = \psi \times \omega + \varepsilon \tag{62}$$

The advantage of (62) is that it is simpler to propagate than the attitude error expression (59). Knowledge of ψ and $\delta\theta$ allows determination of ϕ at any instant of time by (60). Noting that since in (62) the time rate of change of ψ is relative to the local-level frame which rotates with angular rate ω with respect to inertial space we can conclude:

$$\dot{\psi}_I \triangleq \dot{\psi} + \omega \times \psi = \varepsilon \tag{63}$$

where $\dot{\psi}_I$ denotes the time rate of change of ψ relative to inertial space. In words, the rate of change of ψ viewed from inertial space is simply the generalized drift rate of the system gyros projected onto the inertial frame coordinate axes[4].

RELATIONSHIP BETWEEN COORDINATE FRAMES

The correspondence between the coordinate frames discussed above can be summarized conveniently in Figure 2 below. Herein the arrows indicate in which direction the denoted transformation is used in obtaining the ensuing coordinate frame. Since all transformations are orthogonal however, the inverse transformation is realized by the transpose.

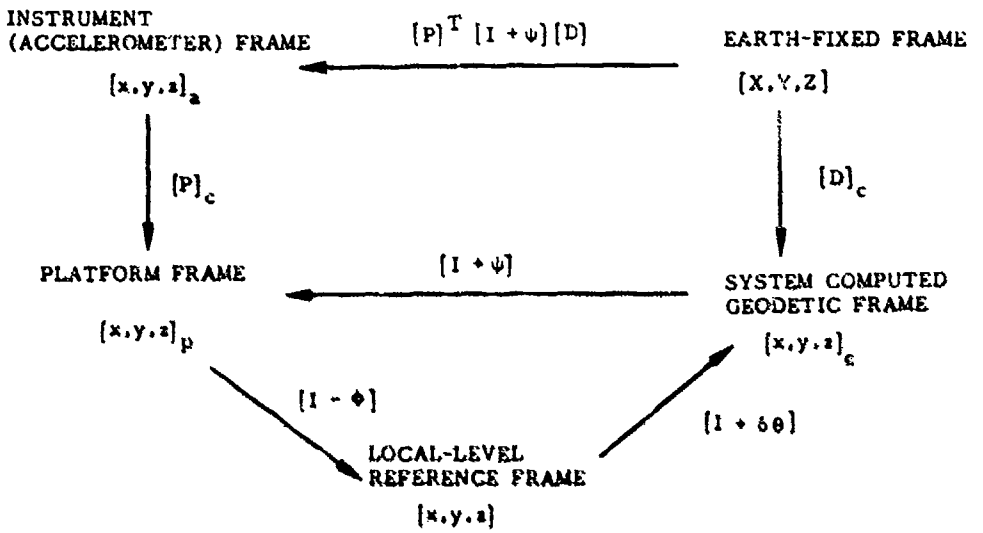


Figure 2. Relationship Between Coordinate Frames

KALMAN FILTER MODELING CONSIDERATIONS

After having developed the fundamental linear differential equations describing the error propagation for inertial navigation systems, we are now in a position to discuss various simplifications which have been made to implement operational Kalman filter designs. The objective of such simplifications in a specific application is to reduce the computer memory and duty cycle requirements without incurring unacceptable degradation in the performance of the operational filter.

Simplifications that are investigated by Kalman filter design engineers generally fall into two categories. The first of these categories deals with the question as to how many of the modelable instrument error states present in the error projection vectors for accelerometer measurement error ∇, and generalized gyro drift rate ε, need be included as states in the Kalman filter. The elemental states to be considered include instrument bias, scale factor errors, sensitive axis mechanical misalignments, sensitivity to products of acceleration and angular rates, correlated noise, etc. The decision as to which of these states are incorporated in the Kalman design model is made after several design iterations in which performance with and without various states present in the design model is determined, "filter tuning" to accommodate absence of states in the design model is performed and trade-offs in the specification of error parameters values for the actual system have been investigated. Such a design process is usually of lengthy duration and requires the use of highly-sophisticated simulation software for its accomplishment. These issues will not concern us further here.

The second category of simplifications addresses the error dynamics of the navigation system errors (20, 39, 59) and is what is of interest here. Two types of simplification are of interest, these being reduction of the modeled error states as above, and secondly the reduction of the dynamic coupling between the error states that are retained. Simplifications that have proved to be especially useful in operational filter design are:

- Vertical axis model elimination
- Level axes Coriolis acceleration elimination
- Use of the "ψ equation", (62)
- Alternative definitions of azimuth error

and are discussed in detail below.

VERTICAL AXIS MODEL ELIMINATION

The subset of the above error model equations which describe the propagation of error for the vertical axis of the local-level coordinate frame navigation equations is:

$$\begin{aligned} \dot{\delta h} &= \delta V_z \\ \dot{\delta V}_z &= \phi_y A_x - \phi_x A_y + \delta C_z + \delta \gamma_z + \nabla_z \end{aligned} \tag{64}$$

where the error in the system computed Coriolis acceleration component along the vertical axis is:

$$\delta C_z \triangleq (\delta\rho + 2\delta\Omega)_y \cdot V_x - (\delta\rho + 2\delta\Omega)_x \cdot V_y + (\rho + 2\Omega)_y \cdot \delta V_x - (\rho + 2\Omega)_x \cdot \delta V_y \tag{65}$$

Due to the dependence of the normal gravity γ, on the elevation[3]

$$\frac{\delta\gamma}{\delta h} = -2\gamma J - 2\Omega^2 \tag{66}$$

where $J = 0.5\,[M^{-1} + N^{-1}]$, is the mean curvature of the ellipsoid and M and N are the meridional and normal radii of curvature, the error in system computed elevation leads to divergence of the vertical axis errors ($\delta h, \delta V_z$). Because of this effect the system elevation computation error is normally bounded in aircraft applications by the use of a reference source of elevation as a barometric altimeter which is also used in a servo loop arrangement to bound the vertical velocity error and bias the error in the vertical acceleration measurements. This is done even when the level axes computations are uncorrected with other reference sensor data and are operating in the "free-inertial" mode.

Since conventional fixed-gain error control mechanizations using reference elevation measurements have obtained adequate vertical axis performance in most applications, this channel is often ignored in the application of Kalman filtering to inertial systems.* Further since there is little cross-coupling from the vertical axis to the level axes, there is little need to model the vertical axis error states in obtaining good control over the level axis error states. In this regard the elevation error does not affect at all the propagation of error for the level axes whereas the vertical velocity error affects level axis error propagation only through error in computation of the level axes Coriolis acceleration components as discussed below.

LEVEL AXES CORIOLIS ACCELERATION ELIMINATION

The errors in the system computed level axes Coriolis acceleration components are:

$$\begin{aligned} \delta C_x \triangleq\ & (\delta\rho + 2\delta\Omega)_z \cdot V_y - (\delta\rho + 2\delta\Omega)_y \cdot V_z \\ & + (\rho + 2\Omega)_z \cdot \delta V_y - (\rho + 2\Omega)_y \cdot \delta V_z \\ \delta C_y \triangleq\ & (\delta\rho + 2\delta\Omega)_x \cdot V_z - (\delta\rho + 2\delta\Omega)_z \cdot V_x \\ & + (\rho + 2\Omega)_x \cdot \delta V_z - (\rho + 2\Omega)_z \cdot \delta V_x \end{aligned} \tag{67}$$

In most aircraft applications the vertical velocity V_z is nominally zero except at a few times during flight. Consequently, modeling of such an effect would only be considered if small transient effects were important. For example even if the vertical velocity were 100 feet per second (fps), system computed latitude error were a large 10 nautical miles (nm) thereby inducing an error in computed earth rate of 0.03 $\widehat{\text{sec}}$/sec at mid-latitudes (45°) **, and system velocity error were a large 10 fps (0.1 $\widehat{\text{sec}}$/sec), the acceleration error for the Coriolis components associated with V_z is less than 3 milligals.

Due to bounding of the vertical velocity error to a few feet per second via the vertical axis mechanization as discussed above, the magnitude of the acceleration error term due to δV_z is substantially less than the uncertainty in the gravity model terms $\gamma_{x,y,z}$ present in the full velocity error equations (20). For example for a 1 fps error and mid-latitude operation of a vehicle traveling at 1000 fps in a direction reinforcing the level axis earth rate terms, the Coriolis acceleration error is less than 5 milligals which is small compared to the gravity model uncertainty of 40 to 50 milligals on a world-wide basis. These effects are a fortunate coincidence for the Kalman filter designer as he can normally eliminate these dynamical vertical axis dependencies from his design model along with the vertical axis model as noted above.

This type of magnitude of effect analysis can be extended to the Coriolis error components involving the vehicle level velocity components $V_{x,y}$, and the error in their system computed values. One finds again for mid-latitude operation with vehicle velocity components of 1000 fps, errors of 10 fps in the system computed velocity components and 10 nm latitude error,*** the magnitude of the error in these system computed Coriolis acceleration terms is less than 10 milligals. Consequently for most applications the error in the system computed Coriolis acceleration components can be ignored on a magnitude basis relative to a more dominating source of "noise" that arises from the uncertainty in the gravity model. There are some

---

*Exceptions to this rule can occur when very precise measurements of elevation are available which if their use is to be optimal, requires full modeling of the vertical axis error propagation. In some cases corrections can be obtained for level axis tilts due to their acceleration dependent effect on vertical velocity error in (64).

**Assumed as an average condition.

***Here we have presumed use of non-singular forms of the navigation equations where ρ_z is comparable to the earth rate component about the vertical axis.

instances however where the error in these latter system computed Coriolis terms may be important on a dynamical basis. Inspection of free-inertial error propagation curves reveals that the cross-coupling of the level axis velocity errors between axes induces modulation of the Schuler error oscillations with the long-term Foucault period obtained as:

$$\tau_f = \frac{2\pi}{\Omega_z} \tag{68}$$

which is 24 hours at the pole, 33.85 hours at latitude of 45° and infinite at the equator. Normally however when a Kalman filter is being employed, the corrections to the inertial system errors using other reference sensors are obtained at substantially higher frequencies than involved here which "quench" such long-term oscillations and make then irrelevant. All in all then in most cases, the Kalman filter designer discovers that the dynamical coupling of errors associated with the error in the system computed Coriolis acceleration error components may be deleted from his filter design model.

USE OF THE "ψ EQUATION"

As noted previously, the "ψ equation" (62), provides an alternative way of representing attitude error that is simpler than the attitude error propagation equation (59), provided position error is propagated via (39) such that attitude can be recovered at any time instant. Hence in most Kalman filter design models for inertial systems the two following equations are employed, prior to further simplification which may be possible in some applications, to describe position and attitude error propagation:

$$\delta\dot{\theta} = \delta\rho + \delta\theta \times \rho \tag{69}$$

$$\dot{\psi} = \psi \times \omega + \epsilon \tag{70}$$

where attitude error can be computed at any time via:

$$\phi = \psi + \delta\theta \tag{71}$$

ALTERNATIVE DEFINITIONS OF AZIMUTH ERROR

The two attitude errors $\phi_{x,y}$ define the tilt of the platform frame relative to the reference frame with ϕ_z defining the azimuth misalignment. Since error in system computed position is defined by the error in latitude $\delta\phi$, and longitude $\delta\lambda$, it is clear by (4) that the two level position error variables $\delta\theta_{x,y}$ are sufficient to define position error with $\delta\theta_z$ as an error variable to define the azimuth misalignment of the system computed geodetic frame due to error in the computed wander angle $\delta\alpha$ and the error in knowledge of the north direction due to an error in knowledge of longitude. Consequently there appears to be a form of redundance in the system azimuth misalignment definition which if properly exploited might lead to a simplification of the describing error model equations by reducing by one the number of azimuth error states to be considered. A review of the material presented above reveals that the behavior of the wander-angle of the reference frame α, has not specifically been defined. The discussion to this point has only assumed that whatever this behavior is, the resulting error variables:

$$\begin{aligned} \phi_z &\triangleq \alpha_p - \alpha \\ \delta\theta_z &\triangleq \lambda_c - \alpha + \delta\lambda \sin\phi \end{aligned} \tag{72}$$

remain small such that second order effects can be ignored preserving the linearity of the error model.

In the following discussion we confine ourselves to non-singular mechanizations of the local-level, wander-azimuth navigation mechanizations as noted at (8A, 9A) of the appendix. Without loss of generality and for ease of exposition, we consider the most common wander-azimuth mechanization wherein:

$$\rho_{z_c} \triangleq 0 \tag{73}$$

which yields via (13A)

$$\dot{\lambda}_c = -\rho_{N_c} \tan\phi_c \tag{74}$$

and via the definition

$$\rho_z = -\delta\rho_z \tag{75}$$

Note a complete definition of all terms in the error equations (20, 39, 62) now only requires specific definition of the terms ρ_z, $\delta\rho_z$, such that overall linearity is preserved. This specification is obtained by imposing two constraints. The first constraint is (16), which by (73, 74) yields:

$$\dot{\alpha} = -\rho_N \tan\phi - \delta\rho_z \tag{76}$$

The second constraint results from the selection of azimuth behavior for the error model in accordance with one of the alternatives discussed below which then obtains the specification of $\delta\rho_z$.

THE CASE OF THE EIGHT STATE LEVEL AXES ERROR MODEL

In what might be termed the "normal" error model we apply the constraint:

$$\rho_z = 0 \tag{77}$$

which yields the result via (75) that:

$$\delta\rho_z = 0 \tag{78}$$

Clearly (77, 78) are small so that an eight state linear model for the error propagation is obtained using (20, 39, 62, 77, 78). The full set of eight differential equations are summarized in Table 1 where the additional simplifying assumption of nominally constant altitude of flight:

$$V_z = 0$$

$$A_z = \gamma \tag{79}$$

has also been made. An additional property of this error model not fundamental to the error propagation itself is seen to be via (76):

$$\dot{\alpha} = -\rho_N \tan \phi \tag{80}$$

THE CASE $\delta\theta_z \equiv 0$

In many applications it turns out as a practical matter that the "position" azimuth error term $\delta\theta_z$ of the "normal" error model is small relative to the platform azimuth misalignment ϕ_z, and can simply be ignored in the Kalman filter design. This amounts to the presumption that the system computed local-level geodetic frame obtains the same azimuth as the local-level reference frame even though it is angularly-displaced from it.

Thus we have:

$$\delta\theta_z \equiv 0 \tag{81}$$

yielding via (38), since:

$$\delta\dot{\theta}_z = 0 \tag{82}$$

that:

$$\delta\rho_z = \rho_x \cdot \delta\theta_y - \rho_y \cdot \delta\theta_x \tag{83}$$

Hence from (75):

$$\rho_z = \rho_y \cdot \delta\theta_x - \rho_x \cdot \delta\theta_y \tag{84}$$

Clearly again since (83, 84) are small, a linear model for the error propagation is obtained from (20, 39,62, 83, 84) except that in this case by (81) it has only seven instead of eight states. The seven differential equations for this model, ignoring second-order effects, are summarized in Table 1. Additional properties of this error model by (76), (60), (4) are:

$$\dot{\alpha} = -\rho_N \tan \phi + \rho_y \cdot \delta\theta_x - \rho_x \cdot \delta\theta_y \tag{85}$$

$$\psi_z \equiv \phi_z$$

$$\delta\alpha = -\delta\lambda \sin \phi$$

THE CASE $\phi_z \equiv 0$

The natural alternative to the prior case is that where the azimuth of the platform frame coincides with that of the reference frame such that:

$$\phi_z \equiv 0 \tag{86}$$

yielding via (59), since:

$$\dot{\phi}_z = 0 \tag{87}$$

that:

$$\delta\rho_z = -\delta\Omega_z + \phi_y \cdot \omega_x - \phi_x \cdot \omega_y - \epsilon_z \tag{88}$$

which by (7, 17, 60, 62) can be expressed as:

$$\delta\rho_z = \rho_x \cdot \delta\theta_y - \rho_y \cdot \delta\theta_x - \dot{\phi}_z \tag{89}$$

Hence from (75):

$$\rho_z = \dot{\phi}_z + \rho_y \cdot \delta\theta_x - \rho_x \cdot \delta\theta_y \tag{90}$$

Clearly again since (89, 90) are small, a linear error model for the error propagation is obtained from (20, 39, 62, 89, 90) where in this case only seven states are present since the substitution:

$$\delta\phi_z \equiv -\psi_z \tag{91}$$

is made. The seven differential equations for this model ignoring second-order effects, are summarized in Table 1. Additional properties of this model by (76), (4) are:

TABLE I

A COMPARISON OF THREE LEVEL AXES LINEAR ERROR MODELS FOR THE LOCAL-LEVEL, WANDER-AZIMUTH ($\rho_{z_c} = 0, \omega_{z_c} = \Omega_{z_c}$)

NAVIGATION EQUATIONS FOR NOMINALLY CONSTANT ALTITUDE ($V_z = 0$, $A_z = g$)

(ALL THREE MODELS EMPLOY THE "ψ" EQUATION FOR THREE OF THEIR STATES)

"Normal" Eight State Error Model	Seven State, $\delta\theta_z \equiv 0$ Error Model	Seven State, $\phi_z \equiv 0$ Error Model
$\delta\dot{\theta}_x = \delta\rho_x - \rho_y \cdot \delta\theta_z$	$\delta\dot{\theta}_x = \delta\rho_x$	$\delta\dot{\theta}_x = \delta\rho_x + \rho_y \cdot \psi_z$
$\delta\dot{\theta}_y = \delta\rho_y + \rho_x \cdot \delta\theta_z$	$\delta\dot{\theta}_y = \delta\rho_y$	$\delta\dot{\theta}_y = \delta\rho_y - \rho_x \cdot \psi_z$
$\delta\dot{\theta}_z = \rho_y \cdot \delta\theta_x - \rho_x \cdot \delta\theta_y$	$\delta\dot{V}_x = \phi_z \cdot A_y - (\psi + \delta\theta)_y \cdot \gamma + 2\Omega_z \cdot \delta V_y + \delta\gamma_x + V_y \cdot \{(\rho + 2\Omega)_x \cdot \delta\theta_y - (\rho + 2\Omega)_y \cdot \delta\theta_x\} + \nabla_x$	$\delta\dot{V}_x = -(\psi + \delta\theta)_y \cdot \gamma + 2\Omega_z \cdot \delta V_y + \delta\gamma_x + \nabla_x + V_y \cdot \{(\rho + 2\Omega)_x \cdot \delta\theta_y - (\rho + 2\Omega)_y \cdot \delta\theta_x - \dot{\psi}_z\}$
$\delta\dot{V}_x = (\psi + \delta\theta)_z \cdot A_y - (\psi + \delta\theta)_y \cdot \gamma + 2\Omega_z \cdot \delta V_y + 2V_y \cdot \delta\Omega_z + \delta\gamma_x + \nabla_x$	$\delta\dot{V}_y = -\phi_z \cdot A_x + (\psi + \delta\theta)_x \cdot \gamma - 2\Omega_z \cdot \delta V_x + \delta\gamma_y - V_x \cdot \{(\rho + 2\Omega)_x \cdot \delta\theta_y - (\rho + 2\Omega)_y \cdot \delta\theta_x\} + \nabla_y$	$\delta\dot{V}_y = (\psi + \delta\theta)_x \cdot \gamma - 2\Omega_z \cdot \delta V_x + \delta\gamma_y + \nabla_y - V_x \cdot \{(\rho + 2\Omega)_x \cdot \delta\theta_y - (\rho + 2\Omega)_y \cdot \delta\theta_x - \dot{\psi}_z\}$
$\delta\dot{V}_y = -(\psi + \delta\theta)_z \cdot A_x + (\psi + \delta\theta)_x \cdot \gamma - 2\Omega_z \cdot \delta V_x - 2V_x \cdot \delta\Omega_z + \delta\gamma_y + \nabla_y$		
$\delta\rho_z = -\rho_z = 0$	$\delta\rho_z = -\rho_z = \rho_x \cdot \delta\theta_y - \rho_y \cdot \delta\theta_x \qquad \psi_z \equiv \phi_z$	$\delta\rho_z = -\rho_z = \rho_x \cdot \delta\theta_y - \rho_y \cdot \delta\theta_x - \dot{\psi}_z \qquad \psi_z \equiv -\delta\theta_z$
$\dot{\alpha} = -\rho_N \tan\phi$	$\dot{\alpha} = -\rho_N \tan\phi + \rho_z \qquad \delta\alpha = -\delta\lambda \sin\phi$	$\dot{\alpha} = -\rho_N \tan\phi + \rho_z \qquad \delta\alpha = -\psi_z - \delta\lambda \sin\phi$

The Following Equations Apply to All Three of the Above Error Models

Vector Equations	Text Equation
$\delta\dot{\theta} = \delta\rho + \delta\theta \times \rho$	(39)
$\delta\dot{V} = A \times (\psi + \delta\theta) - \delta\{(\rho + 2\Omega) \times V\} + \delta\gamma + \nabla$	(20)
$\dot{\phi} = \delta\omega + \phi \times \omega + \epsilon$	(59)
$\dot{\phi} = \phi \times \omega + \epsilon$	(62)
$\delta\omega = \delta\rho + \delta\Omega$	(49)
$\delta\Omega = \Omega \times \delta\theta$	(19)
$\phi = \phi - S\theta$	(60)

Scalar Equations	Text Equation
$\phi_z \triangleq \alpha_p - \alpha$	(2)
$\delta\alpha \triangleq \alpha_c - \alpha$	(3)
$\delta\theta_x = \delta\theta_E \cos\alpha + \delta\theta_N \sin\alpha$ $\delta\theta_y = \delta\theta_N \cos\alpha - \delta\theta_E \sin\alpha$ $\delta\theta_z = \delta\lambda \sin\phi + \delta\alpha$	(4)
$\delta\rho_x = -\delta V_y/R$ $\delta\rho_y = \delta V_x/R$	(16)

$$\dot{\alpha} = -\rho_N \tan\Phi + \dot{\psi}_z + \rho_y \cdot \delta\theta_x - \rho_x \cdot \delta\theta_y \qquad (92)$$

$$\delta\alpha = -\psi_z - \delta\lambda \sin\Phi$$

One advantage of this design model relative to the prior approximation can be appreciated by comparing the velocity and position error equations in the second and third columns of Table 1. This comparison indicates that the effect of azimuth misalignment can be modeled either at the acceleration level ($\delta\theta_z \equiv 0$) or the velocity level ($\phi_z \equiv 0$). The latter model can often be exploited to reduce computer duty-cycle requirements in that the coefficient terms ($\rho_{x,y}$) being in effect integrals of the acceleration terms ($A_{x,y}$), are smoother and easier to deal with in implementing the model in the digital computer.

A PARTICULAR KALMAN FILTER DESIGN MODEL

If we combine a number of the error model simplifications discussed above, e.g., elimination of the vertical axis model and Coriolis acceleration, use of the "ψ equation" and the azimuth error definition $\phi_z \equiv 0$, the following relatively simple seven state design model for a Kalman filter is obtained:

$$\begin{aligned}
\delta\dot{\theta}_x &= -\delta V_y \cdot R^{-1} + \rho_y \cdot \psi_z \\
\delta\dot{\theta}_y &= \delta V_x \cdot R^{-1} - \rho_x \cdot \psi_z \\
\delta\dot{V}_x &= -(\psi + \delta\theta)_y \cdot \gamma + (\delta\gamma_x + \nabla_x) \\
\delta\dot{V}_y &= (\psi + \delta\theta)_x \cdot \gamma + (\delta\gamma_y + \nabla_y) \\
\dot{\psi}_x &= \Omega_z \cdot \psi_y - \omega_y \cdot \psi_z + \epsilon_x \\
\dot{\psi}_y &= -\Omega_z \cdot \psi_x + \omega_x \cdot \psi_z + \epsilon_y \\
\dot{\psi}_z &= \omega_y \cdot \psi_x - \omega_x \cdot \psi_y + \epsilon_z
\end{aligned}$$

This model has proved useful in practice.

REFERENCES

1. Huddle, J.R., "Application of Kalman Filtering Theory to Augmented Inertial Navigation Systems", Chapter 11, NATO-AGARDograph 139, Edited by C.T. Leondes, Feb., 1970.

2. Brockstein, A.J., and J.T. Kouba, Derivation of Free-Inertial, General Wander-Azimuth Mechanization Equations, Litton Systems Inc., Publication 15960, June, 1969, Revised June, 1981.

3. Heiskanen, W.A., and H. Moritz, Physical Geodesy, W.H. Freeman and Company, San Francisco, 1967.

4. Pinson, J.C., "Inertial Guidance for Cruise Vehicles", Chapter 4 of Guidance and Control of Aerospace Vehicles, Edited by C.T. Leondes, McGraw Hill Book Co., 1963.

APPENDIX - SUMMARY OF LOCAL-LEVEL NAVIGATION MECHANIZATION EQUATIONS

The coordinate frame [x,y,z] referred to herein has two axes [x,y] in the level plane with the z axis normal to the earth's surface at the position of the inertial system. The rates of change of the inertial system velocity relative to the earth taken relative to the rotating local-level coordinate frame and expressed along these axes are:

$$\dot{V} \triangleq \begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{V}_z \end{bmatrix} = A + C - \gamma \qquad (1A)$$

where:

$$A \triangleq \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix}$$

is the specific force measured by the accelerometers due to the true system acceleration and modeled gravity components expressed along the local-level coordinate axes. These measurements can be obtained by direct instrumentation wherein the accelerometers are maintained coincident with the local-level coordinate system as in a Schuler-tuned mechanization or transformed onto these coordinates as occurs with strapdown or space-stable inertial systems.

$$C \triangleq \begin{bmatrix} [(\rho + 2\Omega)_z \cdot V_y - (\rho + 2\Omega)_y \cdot V_z] \\ [(\rho + 2\Omega)_x \cdot V_z - (\rho + 2\Omega)_z \cdot V_x] \\ [(\rho + 2\Omega)_y \cdot V_x - (\rho + 2\Omega)_x \cdot V_y] \end{bmatrix} = [\rho + 2\Omega]\, V$$

are the Coriolis acceleration components which account for earth rotation Ω, and local-level navigation frame rotation $\omega = \rho + \Omega$, relative to inertial space.

$$\rho \triangleq \begin{bmatrix} \rho_x \\ \rho_y \\ \rho_z \end{bmatrix}$$

are the angular rates of rotation of the local-level coordinates expressed about these axes which result as the system moves relative to the earth.

$$\Omega \triangleq \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}$$

are the angular rates of rotation of the earth relative to inertial space expressed about the local-level coordinate axes.

$$\gamma \triangleq \begin{bmatrix} \gamma_x \\ \gamma_y \\ \gamma_z \end{bmatrix}$$

are components of the modeled gravity vector of the earth. Usually the normal gravity field is assumed which corresponds to an ellipsoidal equipotential surface as the reference figure of the earth where $\gamma_x = \gamma_y = 0$ and the effect of earth rotation rate is included in the gravity determination.

$$[\rho + 2\Omega] \triangleq \begin{bmatrix} 0 & (\rho + 2\Omega)_z & -(\rho + 2\Omega)_y \\ -(\rho + 2\Omega)_z & 0 & (\rho + 2\Omega)_x \\ (\rho + 2\Omega)_y & -(\rho + 2\Omega)_x & 0 \end{bmatrix}$$

is a skew-symmetric matrix expressing the sum of earth rate and local-level frame spatial rate with respect to inertial space.

The relative angular rates of rotation are computed after the accelerometer measurements are corrected for Coriolis accelerations and the normal gravity and integrated to obtain the computed level velocity components $V_{x,y}$, as:

$$\rho \triangleq \begin{bmatrix} \rho_x \\ \rho_y \\ \rho_z \end{bmatrix} = \begin{bmatrix} -V_y \cdot R_y^{-1} \\ V_x \cdot R_x^{-1} \\ \dot{\alpha} + \rho_N \tan \phi \end{bmatrix} \qquad (2A)$$

where:

$R_{x,y}$ are the radii of curvature of an ellipsoidal equipotential surface assumed as the figure of the earth. Note different datums (different ellipsoids of reference) are often used which better fit the geoid in the different local areas of system use.

ϕ, λ are the geodetic latitude and longitude of system position

α is the azimuth of the y axis in the level plane

$\rho_N = [\rho_x \sin \alpha + \rho_y \cos \alpha]$ is the relative angular rate about the north axis.
Note $\dot{\alpha} = 0$ in the expression for ρ_z above yields the north-slaved local-level mechanization if $\alpha(0) = 0$.

The earth rate components are computed as:

$$\Omega \triangleq \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} \Omega \cos \phi \sin \alpha \\ \Omega \cos \phi \cos \alpha \\ \Omega \sin \phi \end{bmatrix} \tag{3A}$$

The relative angular rates of change can be used to compute the change in inertial system geodetic position and the azimuth angle α, of the local-level coordinate system. Normally this is achieved using a direction cosine mechanization of the form

$$[\dot{D}] = [\rho]\,[D] \tag{4A}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [D] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The matrix [D] is the orthogonal direction cosine transformation between a set of earth-fixed axes [X,Y,Z], usually with Y coincident with the terrestrial pole and X, Z in the equatorial plane, and the local-level coordinate system as illustrated in Figure 1. In this case the local-level coordinate axes are realized from the earth-fixed coordinates by performing a counter-clockwise rotation about the earth's polar axis λ, a clockwise rotation ϕ, about the displaced X-axis in the equatorial plane and a final counter-clockwise rotation α, about the then vertical axis z. More explicitly:

$$[D] \triangleq \begin{bmatrix} \langle X,x\rangle & \langle Y,x\rangle & \langle Z,x\rangle \\ \langle X,y\rangle & \langle Y,y\rangle & \langle Z,y\rangle \\ \langle X,z\rangle & \langle Y,z\rangle & \langle Z,z\rangle \end{bmatrix} \tag{5A}$$

$$\langle X,x\rangle = \cos\alpha \cos\lambda - \sin\alpha \sin\phi \sin\lambda$$

$$\langle X,y\rangle = -\sin\alpha \cos\lambda - \cos\alpha \sin\phi \sin\lambda$$

$$\langle X,z\rangle = \cos\phi \sin\lambda$$

$$\langle Y,x\rangle = \sin\alpha \cos\phi$$

$$\langle Y,y\rangle = \cos\alpha \cos\phi$$

$$\langle Y,z\rangle = \sin\phi$$

$$\langle Z,x\rangle = -\cos\alpha \sin\lambda - \sin\alpha \sin\phi \cos\lambda$$

$$\langle Z,y\rangle = \sin\alpha \sin\lambda - \cos\alpha \sin\phi \cos\lambda$$

$$\langle Z,z\rangle = \cos\phi \cos\lambda$$

Further:

$$[\rho] \triangleq \begin{bmatrix} 0 & \rho_z & -\rho_y \\ -\rho_z & 0 & \rho_x \\ \rho_y & -\rho_x & 0 \end{bmatrix} \tag{6A}$$

is the anti-symmetric matrix of relative angular rates of change of the navigation coordinate axes relative to the earth-fixed frame expressed in the navigation coordinate axes. These rates result as the inertial system position changes relative to the earth. More explicitly the geodetic position and wander angle rates of change may be expressed as:

$$\begin{aligned} \dot{\phi} &= -\rho_E = [\rho_y \sin\alpha - \rho_x \cos\alpha] \\ \dot{\lambda} &= \rho_N \sec[\phi] \\ \dot{\alpha} &= \rho_z - \rho_N \tan[\phi] \end{aligned} \tag{7A}$$

As a note of interest only six elements of the direction cosine matrix need to be propagated in the mechanization of the navigation equations. Inspection of the propagation equation (4A) reveals that the elements of any column of the direction cosine matrix is propagated using only the other two elements of the column and the appropriate relative angular rates. Any column (row) specifies an earth-fixed (local-level) coordinate axis relative to the local-level (earth-fixed) frame. Two such columns (rows) are sufficient to completely define the transformation since the missing axis is simply the vector cross-product of the other two axes, e.g., Z = XxY. Further inspection of the direction cosines (5A) reveals $[\phi,\alpha,\lambda]$ can be determined fully from five of the direction cosine elements.

where ρ_z varies dependent on the type of azimuth mechanization selected. The direction cosine mechanization avoids the apparent signularities above if the relative angular rates $\rho_{x,y,z}$, remain non-singular. The level components are non-singular functions of the system velocity relative to the earth via (2A). The azimuth relative rotation rate ρ_z is usually selected to be a non-singular function μ, which can but need not be a function of the computed navigation variables:

$$\rho_z = \mu \tag{8A}$$

Note for non-singular behavior of ρ_z, the rate of azimuth wander angle change is singular as:

$$\dot{\alpha} = -\rho_N \tan \phi + \mu \tag{9A}$$

The final navigation equation is that used to compute the system position change along the local vertical relative to the reference ellipsoid. The equation that is integrated to obtain elevation change is:

$$\dot{h} = V_z \tag{10A}$$

In words, the time rate of change of elevation is the system velocity relative to the earth projected along the vertical axis of the local-level reference frame.

ACTUAL REPRESENTATION OF THE LOCAL-LEVEL NAVIGATION MECHANIZATION EQUATIONS

The correct representation of the rate of change of velocity relative to the earth with respect to the local-level coordinate frame, that is actually mechanized in the system computer differs from (1A) and is written:

$$\dot{V}_c \triangleq \begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{V}_z \end{bmatrix} = A_p + [\rho + 2\Omega]_c V_c - \gamma_c \tag{11A}$$

where:

$$A_p \triangleq \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix}_p = [I + \phi]A$$

are the accelerometer measurements as they are made along the platform coordinate axes as introduced in the text of the chapter. Note since the accelerometers measure the sum of system acceleration and actual gravity (as opposed to modeled gravity) and further make these measurements with error due to instrument imperfections, the vector A_p should implicitly include such effects. We choose here however to represent the difference between actual and modeled gravity and instrument measurement error both in the vector ϕ of the text and let A_p represent the sum of system acceleration amd modeled gravity as viewed from the platform coordinate axes.

$$\gamma_c \triangleq \begin{bmatrix} \gamma_x \\ \gamma_y \\ \gamma_z \end{bmatrix}_c$$

are the components of the modeled gravity that are computed using system computed values of position and wander angle $[\phi, \lambda, \alpha]_c$.

$$\Omega_c \triangleq \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}_c$$

are components of the earth rotation rate that are computed using the system computed values of position and wander angle, $[\phi, \alpha]_c$.

and:

$$\rho_c \triangleq \begin{bmatrix} \rho_x \\ \rho_y \\ \rho_z \end{bmatrix}_c \triangleq \begin{bmatrix} -V_y \cdot R_y^{-1} \\ V_x \cdot R_x^{-1} \\ \dot{\alpha} + \rho_N \tan \phi \end{bmatrix}_c \tag{12A}$$

are the system computed values of the relative angular rate using the system computed values of velocity V_c, that result from the integration of (11A), the radii of curvature that are computed using system computed values of position and wander angle $[\phi, \alpha]_c$ and ρ_{z_c} is the azimuth relative rotation rate that is not explicitly specified at this point. Note however the terms in this equation:

$$\rho_{z_c} = \dot{\alpha}_c + \rho_{N_c} \tan \phi_c$$

$$\rho_{N_c} = \rho_{x_c} \sin \alpha_c + \rho_{y_c} \cos \alpha_c \tag{13A}$$

are all obtained from values which are computed in the system computer. The reader is cautioned not to associate the velocity triplet V_c, with the computer coordinate system introduced in the text of the chapter but simply consider it as the set of numbers that result from the integration of Equation (11A).

The actual direction cosine matrix between the earth-fixed axes and system computed geodetic frame is obtained using the system computed relative angular rates between these two coordinate frames as:

$$[\dot{D}]_c = [\rho]_c [D]_c \tag{14A}$$

where:

$$[\rho]_c \triangleq \begin{bmatrix} 0 & \rho_{z_c} & -\rho_{y_c} \\ -\rho_{z_c} & 0 & \rho_{x_c} \\ \rho_{y_c} & -\rho_{x_c} & 0 \end{bmatrix}$$

and $[D]_c$ is the computation of $[D]$ in (5A) above using $[\phi, \lambda, \alpha]_c$.

Finally, the actual representation of the equation which is integrated in the system computer to obtain elevation is:

$$\dot{h}_c = V_{z_c} \tag{15A}$$

OPTIMAL FILTERING AND CONTROL TECHNIQUES FOR TORPEDO-SHIP TRACKING SYSTEMS

by

Prof. Dr.-Ing. R. Lunderstädt
and
Dipl.-Ing. R. Kern

University of the German Armed Forces
Holstenhofweg 85
D-2000 Hamburg 70
FRG

SUMMARY

For destinating interception trajectories for torpedo-ship systems it is evident to use on one hand optimal control theory and on the other hand optimal filtering techniques. At first a simple mathematical model is given for the plane torpedo motion, which is sufficient to investigate principal effects of controlling and filtering. For known target trajectories, then in a first main part of the paper an optimal control law is derived which implies the minimization of a linear combination of interception time and engine energy. The second main part is dedicated to the filtering problem of the measurement data. Thereby it is shown that for the torpedo a continuously working Kalman filter can be used. For generating the target trajectory it is more advantageous to take discrete equations in relative coordinates. For this an appropriate algorithm is derived which bases on an extended Kalman filter. This algorithm is used for smoothing and filtering the target (ship) data and furthermore for generating the target trajectory. As the central point of the paper is not the mathematical theory but the engineering application for all theoretical derivations, simulation results are given, which are obtained by calculations on a hybrid computer.

1. INTRODUCTION

Besides applications in aeronautics and astronautics especially in naval engineering interception problems often occur for the guidance of naval vehicles by which one understands in general the "hard" interaction of two moving objects, e. g. a target and a pursuer. This means mathematically that at a prescribed or by an algorithm (guidance law) fixed point of time the position coordinates of the two moving vehicles have to be equal.

A well-known interception trajectory is for example the dog fight curve, where a pursuer is controlled in such a way that his forward axis is always directed on the target [1]. This classical trajectory has the advantage of a generally good accuracy. The disadvantage is the long interception time especially in such cases, where the velocities of the target and the pursuer are nearly equal [2]. Referring to this the different so-called interception course procedures [3] including the proportional navigation [4] are more favorable. They use some information about the motion of the target in advance in order to hit the target at a prescribed collision point. Especially at the end of the motion, these procedures have the disadvantage that stability problems occur in the guidance law which deteriorate the accuracy or cut off the interception completely [5]. Modern time domain procedures of the control theory, as for instance the quadratic optimal control, are used in only a few cases by now for the conception of guidance laws for interception trajectories [6], [7]. The reasons for this lie in the relatively high mathematical expenditure which is troublesome especially for the solution of problems in real-time.

Depending on the set problem time-and fuel- or energy-optimal algorithms are offering for modern solutions of interception trajectories. By this, approximately the spectrum of the most important applications for the problem torpedo-ship is covered. Because of the complexity of the equations of motion, especially of the pursuer, the guidance laws cannot given analytically in this case. If one wants to exclude numerical procedures, which is useful for basic declarations and necessary for real-time tasks then one obviously has to simplify the mathematical model of the pursuer. A reasonable restriction is the consideration of only plane motions, such as they occur for example for an interception torpedo-surface ship. If one emphasizes furthermore energetic points of view then one gets linear equations of motion, which are derived from the kinetics of mass points with in the mean only little alterations of the velocity of the pursuer. On this model it is easy to apply the optimal control theory.

The realization of the optimal control laws assumes the measurement of the position and the velocity coordinates of the target (ship) and the pursuer (torpedo). In both cases stochastic disturbances occur. Because of the linear mathematical model of the pursuer the disturbances for this one can be eliminated in a simple way by using a Kalman filter. In the case of the target the task is not so easy, as the target trajectory is generally not known and has to be generated from the measurement data. As the measurement equation is nonlinear and a linearization about a nominal trajectory is not possible, one gets at first a nonlinear filter problem [8] for the target trajectory. However, this problem can be attributed to an extended Kalman filter so that altogether this measurement task can be handled by the linear theory, too.

2. MATHEMATICAL MODEL

Starting from a plane motion for the target and the pursuer and assuming the validity of the kinetics of mass points for the pursuer then the normalized equations of motion for the torpedo can be described in accordance with Fig. 1 in the form

$$\dot{x}_1 = x_3 + v_1(x_1, x_2, t) \quad , \quad x_1(t_o) = x_{10}$$
$$\dot{x}_2 = x_4 + v_2(x_1, x_2, t) \quad , \quad x_2(t_o) = x_{20} \qquad (2.1a)$$

$$
\begin{aligned}
\dot{x}_3 &= -\varepsilon \cdot x_3 \cdot \sqrt{x_3^2 + x_4^2} - x_5 \quad , \quad x_3(t_o) = x_{30} \\
\dot{x}_4 &= -\varepsilon \cdot x_4 \cdot \sqrt{x_3^2 + x_4^2} - x_6 \quad , \quad x_4(t_o) = x_{40} \\
\dot{x}_5 &= -\frac{1}{\beta} \cdot x_5 + \frac{k}{\beta} \cdot u_1 \cdot \cos(x_7) \quad , \quad x_5(t_o) = x_{50} \\
\dot{x}_6 &= -\frac{1}{\beta} \cdot x_6 + \frac{k}{\beta} \cdot u_1 \cdot \sin(x_7) \quad , \quad x_6(t_o) = x_{60} \\
\dot{x}_7 &= x_8 \quad , \quad x_7(t_o) = x_{70} \\
\dot{x}_8 &= \frac{1}{\gamma} \cdot u_2 \quad , \quad x_8(t_o) = x_{80} \quad .
\end{aligned}
\tag{2.1b}
$$

Thereby x_1 and x_2 are the position coordinates of the torpedo, x_3 and x_4 its relative velocities in relation to its flow field, x_5 und x_6 the components T_{x_1} and T_{x_2} of the thrust T, x_7 the direction of the thrust in the coordinate system (x_1,x_2) and x_8 the angular velocity of the torpedo. The components of the flow field are v_1 and v_2. Further the coefficient in the quadratic drag law is denoted with ε, and the moment of inertia of the torpedo about its yaw axis is denoted with γ. The time constant of the engine is β, the amplification of the engine is k. As control variables are introduced u_1, as a measure for the amount of the thrust T, and u_2 as the rudder angle. The time is denoted with t, accordingly to that d/dt = (˙).

The system of the equations (2.1) is nonlinear and additionally time varying as far as instationary flow fields are permitted. Altogether it is not accessible to an analytic treatment within the scope of an optimization problem. Therefore, in the following the simplifications

1) $\gamma \to 0$

2) $\beta \to 0$

3) $v_f = \text{const.}$

4) $x_3^2 + x_4^2 \approx 1$

are taken, i. e. the dynamics of the angular motion of the torpedo and that of the engine are neglected and a constant flow field is assumed. Furthermore the system (2.1) is considered to be normalized in such a way that in addition to v_f also the relative velocity v_r of the torpedo can be regarded approximately as constant in its absolute value. Under these assumptions the system (2.1) turns in

$$
\begin{aligned}
\dot{x}_1 &= x_3 + v_1 \quad , \quad x_1(t_o) = x_{10} \\
\dot{x}_2 &= x_4 + v_2 \quad , \quad x_2(t_o) = x_{20} \\
\dot{x}_3 &= -\varepsilon \cdot x_3 - u_1 \cdot \cos(u_2) \quad , \quad x_3(t_o) = x_{30} \\
\dot{x}_4 &= -\varepsilon \cdot x_4 - u_1 \cdot \sin(u_2) \quad , \quad x_4(t_o) = x_{40} \quad .
\end{aligned}
\tag{2.2}
$$

As new control variables now u_1 as thrust and u_2 as its direction in the coordinate system (x_1,x_2) are introduced. Taking finally

$$
\begin{aligned}
\tilde{u}_1 &= u_1 \cdot \cos(u_2) \\
\tilde{u}_2 &= u_1 \cdot \sin(u_2)
\end{aligned}
\tag{2.3}
$$

the linear pursuer dynamic follows

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\varepsilon & 0 \\ 0 & 0 & 0 & -\varepsilon \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ 0 \\ 0 \end{bmatrix} \quad , \tag{2.4}
$$

which means that the motion of the torpedo is described by a linear system of the form

$$
\dot{\underline{x}} = \underline{A} \cdot \underline{x} + \underline{B} \cdot \underline{\tilde{u}} + \underline{v} \quad , \quad \underline{x}(t_o) = \underline{x}_o \quad . \tag{2.5}
$$

At first it is assumed for the target that its dynamics is completely known for $t \geq t_o$. It will be described by the position coordinates

$$\xi_1 = \xi_1(t)$$
$$\xi_2 = \xi_2(t) \tag{2.6}$$

and the belonging velocities

$$\xi_3(t) = \dot{\xi}_1$$
$$\xi_4(t) = \dot{\xi}_2 \quad , \tag{2.7}$$

so that the vector of the relative motion of target (ship) and pursuer (torpedo) is given by

$$\underline{e}(t) = \underline{\xi}(t) - \underline{x}(t) \quad . \tag{2.8}$$

3. OPTIMIZATION PROBLEM

In the following, first the optimization problem is formulated, then it is solved and discussed in detail.

3.1 Formulation

As mentioned preliminary, for an interception torpedo-ship time- and energy-optimal solutions are of interest. Therefore the optimization problem - this means the destination of the control variables $\tilde{u}_1$ and $\tilde{u}_2$ in Eq. (2.4) and the determination of the interception time T - is formulated in such a way that the quadratic performance index

$$J(\underline{u},T) = \frac{1}{2}\cdot\underline{e}^T(T)\cdot\underline{P}\cdot\underline{e}(T) + \frac{1}{2}\cdot\int_{t_o=0}^{T}(\delta\cdot(1-\delta)\cdot[\underline{e}^T(t)\cdot\underline{Q}\cdot\underline{e}(t) + u_1^2(t)])\cdot dt \tag{3.1}$$

is minimized. Thereby the weighting factor δ considers the interception time T. It is defined for

$$0 \leq \delta < 1 \quad . \tag{3.2}$$

Consequently one gets for $\delta = 0$ pure energy-optimal and for $\delta \to 1$ pure time-optimal solutions. Decidedly, the latter case should be excluded because of the linearity of the optimization problem. The weighting matrices $\underline{P}$ and $\underline{Q}$ in Eq. (3.1) are given by

$$\underline{P} = \begin{bmatrix} p_{11} & 0 & 0 & 0 \\ 0 & p_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad \underline{Q} = \begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.3}$$

whereby with $\underline{P}$ the interception condition $x_1(T) = \xi_1(T)$, $x_2(T) = \xi_2(T)$ and with $\underline{Q}$ a weighting of the control deviation $\underline{e}(t)$ is realized. Consequently it is $p_{11} > 0$, $p_{22} > 0$ and $q_{11} \geq 0$, $q_{22} \geq 0$.

3.2 Solution

For the solution of the optimization problem in the following the absolute value of the thrust $u_1(t)$ is considered to be unconstraint. By this the control variables $\tilde{u}_1(t)$ and $\tilde{u}_2(t)$ are unconstraint, too. Therefore one can use the methods of the calculus of variations, an application of the maximum principle is superfluous. Moreover, by this the linearity of the optimization problem is guaranteed.

From the calculus of variations [9] one gets for the optimal control variables

$$\overset{*}{\tilde{u}}_1 = -\frac{1}{(1-\delta)}\cdot\phi_3(t,T)$$
$$\overset{*}{\tilde{u}}_2 = -\frac{1}{(1-\delta)}\cdot\phi_4(t,T) \quad , \tag{3.4}$$

with ϕ_3 and ϕ_4 as components of the adjoint vector $\underline{\phi}$ belonging to $\underline{x}$. With Eq. (3.4) it follows from Eq. (2.3)

$$\overset{*}{u}_1 = -\frac{1}{(1-\delta)}\cdot\sqrt{\phi_3^2(t,T) + \phi_4^2(t,T)}$$
$$\tan(\overset{*}{u}_2) = \phi_4(t,T)/\phi_3(t,T) \quad . \tag{3.5}$$

The optimal interception time $\overset{*}{T}$ can be computed from

$$H(T) - \frac{\partial}{\partial T}\left[\frac{1}{2}\,\underline{e}^{T}(T)\cdot\underline{P}\cdot\underline{e}(T)\right] = 0 \quad , \tag{3.6}$$

with H as HAMILTONIAN. The condition (3.6) is equivalent to the demand

$$\frac{\partial J(T)}{\partial T} = 0 \; . \tag{3.7}$$

The explicit solution of the optimization problem now requires the destination of the adjoint vector $\underline{\psi}(t)$. If this is done then, the optimal control variables are explicitly known from Eq. (3.4), the optimal interception time $\overset{*}{T}$ can be computed from Eq. (3.6) and the open-loop control system (2.4), (3.4) can be fed-back. For this at first a special case.

3.2.1 Special Case $\underline{Q} = \underline{0}$

In the special case $\underline{Q} = \underline{0}$, where no weighting of the control deviation $\underline{e}(t)$ in the performance index (3.1) happens, the destination of $\underline{\psi}(t)$ and by this the solution of the optimization problem is very simple. One gets for the optimal thrust and for the optimal direction of the thrust in the coordinate system (x_1, x_2)

$$\begin{aligned} \overset{*}{u}_1(t,T) &= -\frac{K(T)}{\epsilon\cdot(1-\delta)}\cdot\left[1-e^{-\epsilon\cdot(T-t)}\right] \\ \tan[\overset{*}{u}_2(T)] &= \tan[\alpha(T)] \quad . \end{aligned} \tag{3.8}$$

Thereby K and α are constants of integration, which follow from the transversality condition

$$\psi_i(T) = p_{ii}\cdot[\xi_i(T)-x_i(T)], \quad i = 1,2 \tag{3.9}$$

and are given by

$$\begin{aligned} K(T) &= -\frac{p}{1+\frac{p}{\epsilon^3\cdot(1-\delta)}\cdot k_3(T,T)}\cdot\sqrt{k_1^2(T)+k_2^2(T)} \\ \tan[\alpha(T)] &= \frac{k_2(T)}{k_1(T)} \quad . \end{aligned} \tag{3.10}$$

For an interception equal for both position coordinates, $p_{11} = p_{22} = p$ is set. The auxiliary functions $k_1(T),\ldots,k_3(t,T)$ in Eq. (3.10), which are introduced as abbreviations, are defined by

$$\begin{aligned} k_1(T) &= x_{10} + v_1\cdot T + \frac{x_{10}}{\epsilon}\cdot(1-e^{-\epsilon\cdot T}) - \xi_1(T) \\ k_2(T) &= x_{20} + v_2\cdot T + \frac{x_{20}}{\epsilon}\cdot(1-e^{-\epsilon\cdot T}) - \xi_2(T) \\ k_3(t,T) &= -\frac{1}{\epsilon} + \epsilon\cdot t + e^{-\epsilon\cdot t} + e^{-\epsilon\cdot T}\cdot[1-\cosh(\epsilon\cdot t)] \quad . \end{aligned} \tag{3.11}$$

As it can be seen from Eq. (3.8) the optimal thrust is an exponential function with $\overset{*}{u}_1(T,T) = 0$ and its direction is constant, which means that the control procedure works with an optimal deflection angle, as the target coordinates $\xi_1(T)$ and $\xi_2(T)$ are considered as completely known before. With Eq. (3.8) and Eq. (3.10) for the trajectory of the pursuer follows

$$\begin{aligned} x_1(t,T) &= x_{10} + v_1\cdot t + \frac{x_{10}}{\epsilon}\cdot(1-e^{-\epsilon\cdot t}) + \frac{K\cdot\cos(\alpha)}{\epsilon^3\cdot(1-\delta)}\cdot k_3(t,T) \\ x_2(t,T) &= x_{20} + v_2\cdot t + \frac{x_{20}}{\epsilon}\cdot(1-e^{-\epsilon\cdot t}) + \frac{K\cdot\sin(\alpha)}{\epsilon^3\cdot(1-\delta)}\cdot k_3(t,T) \quad , \end{aligned} \tag{3.12}$$

which results in the special case of vanishing initial velocities $x_{10} = x_{20} = 0$ and vanishing of the flow field $v_1 = v_2 = 0$ in

$$\frac{x_2(t,T)-x_{20}}{x_1(t,T)-x_{10}} = \tan[\alpha(T)] \quad . \tag{3.13}$$

The optimal trajectory is therefore a straight line. Finally, the optimal interception time $\overset{*}{T}$ is computed from

$$\frac{1}{2}\dot{\delta} + \frac{p}{2}\cdot\frac{\partial}{\partial T}\left\{\frac{k_1^2(T)+k_2^2(T)}{1+\frac{p}{\epsilon^3\cdot(1-\delta)}\,k_3(T,T)}\right\} = 0 \tag{3.14}$$

which has to be done numerically.

An example for the solution derived shows Fig. 2 and Fig. 3. In this example the target trajectory

$$\xi_1(t) = (1+v_1)\cdot t$$

$$\xi_2(t) = 0$$

for $t \leq 0$ and

$$\xi_1(t) = \sin(t) + v_1\cdot t$$

$$\xi_2(t) = \cos(t) - 1 + v_2\cdot t$$

for $t > 0$ is assumed. Thus the trajectory is a circle which is displaced by the velocities v_1 and v_2 of the flow field. For the pursuer the initial conditions $x_{10} = 0$, $x_{20} = 1$, $x_{30} = x_{40} = 0$ are valid. The drag coefficient is $\varepsilon = 1$. Furthermore it is $v_1 = 0{,}1$, $v_2 = 0$ and $p = 10^3$. Aim of the investigation is to look for the influence of the weighting factor δ which is responsible for the optimal interception time $\tilde{T}$. In Fig. 2 the optimal interception time $\tilde{T}$ and the constants of integration K and a are drawn over the parameter δ. In Fig. 3 the target trajectory is pointed out and for different values of δ the interception trajectories are given. The figuration outlines that by an appropriate choice of δ any combinations of time- and energy-optimal pursuer trajectories can be realized.

The solution so far includes only the open-loop control problem. If a closed-loop control is needed, then the theory given in [9] yields with $p_{11} = p_{22} = p$

$$\begin{bmatrix} \tilde{u}_1(t,T) \\ \tilde{u}_2(t,T) \end{bmatrix} = \frac{1}{(1-\delta)} \cdot \begin{bmatrix} r_{1/2}(T-t) & 0 & r_{3/4}(T-t) & 0 \\ 0 & r_{1/2}(T-t) & 0 & r_{3/4}(T-t) \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} - \begin{bmatrix} g_1(t,T) \\ g_2(t,T) \end{bmatrix} \quad (3.15)$$

which means that the optimal closed-loop control law is of the form

$$\underline{\tilde{u}}(t,T) = \underline{R}(T-t)\cdot\underline{x}(t) - \underline{g}(t,T) \quad . \quad (3.16)$$

So it consists of a linear feed-back and additionally of a feed-forward control. The time-varying coefficients of the control law in Eq. (3.15) are given by

$$r_{1/2}(T-t) = \frac{1}{\varepsilon}\cdot\left[1-e^{-\varepsilon\cdot(T-t)}\right] \cdot r(T-t)$$

$$r_{3/4}(T-t) = \frac{1}{\varepsilon^2}\cdot\left[1-e^{-\varepsilon\cdot(T-t)}\right]^2 \cdot r(T-t) \quad (3.17)$$

$$r(T-t) = \frac{p}{1+\frac{p}{\varepsilon^3\cdot(1-\delta)}\cdot k_3(T-t)}$$

$$k_3(T-t) = -1+\varepsilon\cdot(T-t)+e^{-\varepsilon\cdot(T-t)}\cdot\left[2-\mathrm{chl}\varepsilon\cdot(T-t)\right] \quad .$$

and for the feed-forward control one gets

$$g_1(t,T) = r_{1/2}(T-t)\cdot\xi_1(T)-v_1\cdot r_{3/4}(T-t)$$

$$g_2(t,T) = r_{1/2}(T-t)\cdot\xi_2(T)-v_2\cdot r_{3/4}(T-t) \quad , \quad (3.18)$$

which follows both from the solution of a RICCATImatrix differential equation [9]. The control law depends on the target final coordinates $\xi_1(T)$ and $\xi_2(T)$ because of the feed-forward control $g(t,T)$. Therefore these are to be known. The coefficients of the controller are pure time functions, which can be calculated off-line. For $p \gg 1$ they are nearly independent of δ, which means that the weighting of the interception time is needed only for destinating itself from (3.1). Exemplary for $\varepsilon = 1$ and $p = 10^3$ the coefficients of the controller are drawn in Fig. 4. In order to make them independent of the interception time T and the parameter δ they are normalized with $(1-\delta)$ and as new variable the time $\tau = T-t$ is introduced. The example of the open-loop control in Fig. 5 and Fig. 6 is again taken as a basis for the closed-loop control as well. In Fig. 5 the optimal thrust $\tilde{u}_1(t)$ is drawn for $\delta = 0{,}7$ both for the open- and closed-loop control. One sees the formal conformity of both curves. Only at the end of the trajectory little deviations occur, which base on the interception condition which is with $p = 10^3$ only finite weighted. Additionally, in Fig. 5 the optimal thrust for one with

$$\Delta v_1(t) = 0{,}5\cdot[\sigma(t-0{,}5)-\sigma(t-1{,}5)+\sigma(t-2{,}5)-\sigma(t-3{,}5)]$$

disturbed motion is drawn. By this disturbance, which includes two rectangular pulses, the efficiency of defensives measures against the torpedo can be simulated. The trajectories of the problem mentioned are given in Fig. 6. Thereby it is obvious that the closed-loop control works satisfying also in the case of the disturbed motion.

3.2.2 General Case $\underline{Q} \neq \underline{0}$

In the general case $\underline{Q} \neq \underline{0}$ the conditions (3.4) are the same as before. Of course for the destination of the adjoint functions $\psi_3(t)$ and $\psi_4(t)$ now distinctions are necessary, as the optimization problem contains for the eigenvalues λ_i of the system the characteristic equations

$$\lambda_i^4 - \varepsilon^2 \cdot \lambda_i^2 + q_{ii} = 0 \ , \quad i = 1,2 \tag{3.19}$$

Therefore the three cases

$$\begin{aligned} &1)\ q_{ii} < \frac{\varepsilon^4}{4} \\ &2)\ q_{ii} = \frac{\varepsilon^4}{4} \\ &3)\ q_{ii} > \frac{\varepsilon^4}{4} \end{aligned} \tag{3.20}$$

have to be treated separately. As the procedure is identical in all three cases here case 3) is taken, being of special interest in practice. All further considerations are referred to this case. With the abbreviations

$$\begin{aligned} \gamma_i &= \sqrt{\frac{1}{2} \cdot \left[\sqrt{q_{ii}} + \frac{\varepsilon^2}{2} \right]} \\ \omega_i &= \sqrt{\frac{1}{2} \cdot \left[\sqrt{q_{ii}} - \frac{\varepsilon^2}{2} \right]} \ , \end{aligned} \tag{3.21}$$

which are the absolute values of the real and imaginary parts of the eigenvalues λ_i, $i = 1,2$, and the auxiliary functions given in Appendix A one gets

$$\psi_{i+2}(t,T) = - \left[m_{1i}(t) - \frac{m_{1i}(T)}{m_{oi}(T)} \cdot m_{oi}(t) \right] \cdot \psi_{io} - (1-\delta) \cdot q_{ii} \cdot f_{i+2}(t,T) \ . \tag{3.22}$$

The constants of integration ψ_{io}, $i = 1,2$ are related to

$$\psi_{io} = - \frac{p \cdot k_{1i}(T) + (1-\delta) \cdot q_{ii} \cdot k_{2i}(T)}{1 + \frac{p}{(1-\delta)} \cdot l_{1i}(T) + q_{ii} \cdot l_{2i}(T)} \ , \tag{3.23}$$

with the connection

$$\begin{aligned} \psi_{10} &= K \cdot \cos(\alpha) \\ \psi_{20} &= K \cdot \sin(\alpha) \end{aligned} \tag{3.24}$$

to the constants K and α used before. In Eq. (3.23) it is assumed again $p_{11} = p_{22} = p$, the abbreviations $k_{1i}(T)$, $k_{2i}(T)$, $l_{1i}(T)$ and $l_{2i}(T)$ are assigned in Appendix A. In Eq. (3.22) it is now important that because of Eq. (A.4) and Eq. (A.8) the solution of the optimization problem depends not only on the final state $(\xi_1(T), \xi_2(T))$ of the target but also on the whole target trajectory for $0 \leq t \leq T$.

With Eq. (3.22) and Eq. (3.4) it is now possible to integrate the equations of motion (2.4). By this the trajectory of the torpedo is known. The analytic expressions for $x_1(t)$ and $x_2(t)$ will not be given here, in this connection it is referred to [10]. In order to show the influence of the weighting matrix $\underline{Q}$ it is referred to the example treated before. It is shown in Fig. 7 with a parameter variation of $q_{11} = q_{22} = q$, the time of interception T is thereby fixed. It is evident from the graphs that with the parameter q completely different interception trajectories can be realized. This is important in order to make an attack to the ship with the torpedo for instance from behind. The consideration till now are only valid for the open-loop control problem. If one wants to use a closed-loop control, which is generally necessary, one is led again to Eq. (3.15). The coefficients of the controller are now given by

$$\begin{aligned} r_i(T-t) &= - \frac{d_{1i}(T-t)}{d_{oi}(T-t)} \cdot r_{oi}(T-t) - (1-\delta) \cdot q_{ii} \cdot \frac{d_{2i}(T-t)}{d_{oi}(T-t)} \\ r_{i+2}(T-t) &= \left[\frac{d_{1i}(T-t)}{d_{oi}(T-t)} \right]^2 \cdot r_{oi}(T-t) + \\ &\quad (1-\delta) \cdot q_{ii} \cdot \left[\frac{c_{3i}(T-t) \cdot d_{oi}(T-t) + d_{1i}(T-t) \cdot d_{2i}(T-t)}{d_{oi}^2(T-t)} \right] \ , \quad i = 1,2 \ . \end{aligned} \tag{3.25}$$

The abbreviation $r_{oi}(T-t)$ is assigned in Appendix B. The functions $c_{3i}, \ldots, d_{2i}$ in Eq. (3.25) follow from Appendix A, if one replaces there the argument t by t-T. In contrast to Eq. (3.17) the coefficients of the controller have now stationary values not equal to zero, which can be seen from

$$\bar{r}_i = \lim_{T\to\infty} r_i(T-t) = (1-\delta)\cdot\sqrt{q_{ii}}$$

$$\bar{r}_{i+2} = \lim_{T\to\infty} r_{i+2}(T-t) = (1-\delta)\cdot\left[-\varepsilon+\sqrt{\varepsilon^2+2\cdot\sqrt{q_{ii}}}\right] \quad , \quad i = 1,2 \quad . \tag{3.26}$$

The stationary value of the coefficient r_i is not at all influenced by the drag, the stationary value of the coefficient r_{i+2} only little, as $0 \leq \varepsilon \leq 1$ can be assumed from physical reasons and furthermore (3.20) deals with case 3). Moreover the little influence of ε on r_i and on r_{i+2} is not only valid for the stationary values but in general also for $0 \leq t \leq T \leq \infty$, as it can be seen from a numerical exploitation of Eq. (3.25). For the feed-forward control $g_i(t,T)$, $i = 1,2$ in (3.15) one gets now

$$g_i(t,T) = \frac{h_i(T-t)}{n_i(T-t)}\cdot\xi_i(T)-[r_{i+2}(T-t)-\Omega_i(T-t)]\cdot v_i - \\ - q_{ii}\cdot[r_i(T-t)\cdot i_{1i}(t,T)+r_{i+2}(T-t)\cdot i_{2i}(t,T)+(1-\delta)\cdot i_{4i}(t,T)] \quad , \tag{3.27}$$

with the auxiliary functions $h_i(T-t)$ and $\Omega_i(T-t)$ given in Appendix B. The integrals i_{ji} ($j=1,2,4$; $i=1,2$) are thereby explained as in Appendix A but now with t as lower and T as upper integration limit. From Eq. (3.27) it is evident that for the feed-forward control in analogy to the open-loop control the whole target trajectory has to be known for $0 \leq t \leq T$. It is not sufficient to know only the final values ($\xi_1(T)$, $\xi_2(T)$).

The control law (3.15) with the coefficients (3.25) and the feed-forward control (3.27) produces altogether good results. It has, however the disadvantage that it cannot be used for unknown target trajectories. Therefore one has to try with a suboptimal solution to eliminate this disadvantage. For this purpose one develops the integrals (A.4) and (A.8) given in Appendix A by partial integration up to the accelerations $\ddot{\xi}_i(T)$, $i=1,2$. The feed-forward control (3.27) then goes over in

$$g_i(t,T) = r_i(T-t)\cdot\xi_i(t)+r_{i+2}(T-t)\cdot[\xi_{i+2}(t)-v_i]+R_i(t,T) \quad , \tag{3.28}$$

with the residual function

$$R_i(t,T) = (1-\delta)\cdot\varepsilon\cdot[\xi_{i+2}(t)-\xi_{i+2}(T)] - \Omega_i(T-t)\cdot[\xi_{i+2}(T)-v_i] + \\ + r_i(T-t)\cdot I_{1i}(t,T)+r_{i+2}(T-t)\cdot I_{2i}(t,T)-(1-\delta)\cdot q_{ii}\cdot I_{4i}(t,T) \quad . \tag{3.29}$$

This residual function depends on the integrals given in Appendix B and on an additional expression resulting from the hydrodynamic drag. For ships as targets it is sure that their accelerations are small. So far it is allowed to neglect these integrals the influence of which for $t\to T$ in any way vanishes. Furthermore it can be proven that also the additive expressions of higher order depending on ε are small. By this it is possible to put

$$R_i(t,T) = 0 \tag{3.30}$$

for $0 \leq t \leq T$ and the optimal control law can be replaced by the suboptimal expression

$$\tilde{u}_i(T-t) = -\frac{1}{(1-\delta)}\cdot\left\{r_i(T-t)\cdot\left[\xi_i(t)-x_i(t)\right] + \right. \\ \left. + r_{i+2}(T-t)\cdot\left[\xi_{i+2}(t)-x_{i+2}(t)-v_i\right]\right\} \quad , \quad i = 1,2 \quad . \tag{3.31}$$

In the suboptimal control law no preliminary knowledge of the target trajectory is necessary. Only the actual target coordinates for the specific time t are needed.

For the example treated before, but now with $v_i = 0$, $i = 1,2$ and $\varepsilon = 0,5$, Fig. 8 shows the difference between the optimal and the suboptimal solution. One sees that the optimal solution is comparable in a certain scope with the proportional navigation, the suboptimal solution with a dog-fight curve.

An additional example is given in Fig. 9, where the target with the coordinates

$$\xi_1(t) = (1+v_1)\cdot t$$

$$\xi_2(t) = 0$$

for $t \geq 0$ is pursued. Variation parameter is again $q_{11} = q_{22} = q$ and q_{11}, q_{22} respectively. The figure points out that also in the suboptimal case different pursuer trajectories are realizable by an appropriate choice of q, which is extremely important for tactical considerations.

4. OPTIMAL FILTERING

In the further considerations the general suboptimal control law (3.31) is used. In this control law occur the state variables $x_1,\ldots,x_4$ of the pursuer (torpedo) and the state variables $\xi_1,\ldots,\xi_4$ of the target (ship) If one uses in accordance to Eq. (2.8) relative coordinates then the number of state variables is reduced from eight to four but in this case no inertial declarations for the coordinates are possible. It is all the same, which consideration is used. In any case the state variables contained in the control law have to be measured. These measurements are subject to stochastic errors so that one

has to develop appropriate filter algorithms.

4.1 Kalman Filter for Torpedo

As shown in Eq. (2.4) the torpedo can be described by linear state equations. One can assume that in these state equations no stochastic parts occur if for example no influences of wave motions of the flow field are considered. With a proportional linear working measurement device one gets the measurement equation

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \underline{w}(t) \quad , \tag{4.1}$$

if only the position coordinates and

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \underline{w}(t) \quad , \tag{4.2}$$

if also the velocity coordinates are measured, i. e. in general is

$$\underline{y} = \underline{C} \cdot \underline{x} + \underline{w}(t) \ . \tag{4.3}$$

Assuming that $\underline{w}$ in Eq. (4.3) is a normal distributed white noise process with $E[\underline{w}(t)] = \underline{0}$, then for (2.4) and (4.1) a linear Kalman filter can be outlined. In accordance with the theoretical background given in [11] for this case

$$\dot{\hat{\underline{x}}} = \underline{A} \cdot \hat{\underline{x}} + \underline{K}(t) \cdot [\underline{y} - \underline{C} \cdot \hat{\underline{x}}] + \underline{d} \ , \ \hat{\underline{x}}(0) = \hat{\underline{x}}_o = E[\underline{x}(0)] \tag{4.4}$$

is valid.

Here is for abbreviation $\underline{d} = \underline{B} \cdot \tilde{\underline{u}} + \underline{v}$ the deterministic part in Eq. (2.11). The Kalman matrix $\underline{K}(t)$ now follows from

$$\underline{K}(t) = \underline{P}(t) \cdot \underline{C}^T \cdot \underline{R}^{-1} \quad , \tag{4.5}$$

where the covariance matrix $\underline{P}(t)$ obeys the RICCATI matrix differential equation

$$\dot{\underline{P}} = \underline{A} \cdot \underline{P} + \underline{P} \cdot \underline{A}^T - \underline{P} \cdot \underline{C}^T \cdot \underline{R}^{-1} \cdot \underline{C} \cdot \underline{P} \ , \ \underline{P}(0) = E[(\hat{\underline{x}}_o - \underline{x}_o) \cdot (\hat{\underline{x}}_o - \underline{x}_o)^T] \ . \tag{4.6}$$

Furthermore it is

$$E[\underline{w}(t) \cdot \underline{w}(t+\tau)] = \underline{R} \cdot \delta(\tau) \tag{4.7}$$

for the noise process in (4.3). If one assumes for $\underline{P}(0)$ and $\underline{R}$ diagonal matrices of the form

$$\begin{aligned} \underline{P}(0) &= \mathrm{diag}(\sigma_{ii}^2) \quad , \quad i = 1,\ldots,4 \\ \underline{R} \quad &= \mathrm{diag}(r_{ii}^2) \quad , \quad i = 1,\ldots,4 \end{aligned} \tag{4.8}$$

and postulates additionally that there are equal stochastic qualities for the position coordinates on one hand and for the velocity coordinates on the other hand, which means

$$\begin{aligned} \sigma_{11} &= \sigma_{22} \\ \sigma_{33} &= \sigma_{44} \end{aligned} \tag{4.9}$$

and

$$\begin{aligned} r_{11} &= r_{22} \\ r_{33} &= r_{33} \end{aligned} \tag{4.10}$$

then the filter can be given analytically in a simple way. One gets

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \\ \dot{\hat{x}}_4 \end{bmatrix} = \begin{bmatrix} -k_{11} & 0 & 1-k_{13} & 0 \\ 0 & -k_{22} & 0 & 1-k_{24} \\ -k_{31} & 0 & -(c+k_{33}) & 0 \\ 0 & -k_{42} & 0 & -(c+k_{44}) \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} + \begin{bmatrix} k_{11} \cdot y_1 + k_{13} \cdot y_3 \\ k_{22} \cdot y_2 + k_{24} \cdot y_4 \\ k_{31} \cdot y_1 + k_{33} \cdot y_3 \\ k_{42} \cdot y_2 + k_{44} \cdot y_4 \end{bmatrix} + \underline{d} \ . \tag{4.11}$$

Thereby is - because of Eq. (4.9) and (4.10) -

$$k_{11} = k_{22}, \quad k_{13} = k_{24}$$
$$k_{33} = k_{44}, \quad k_{31} = k_{42} \quad . \tag{4.12}$$

The explicit expressions of the elements $k_{ij}(i=1,\ldots,4;\ j=1,\ldots,4)$ are assigned in Appendix C.

If the velocities x_3 and x_4 are not measured then

$$y_3 = y_4 \equiv 0$$

is valid. From this follows

$$k_{13} = k_{33} \equiv 0 \ .$$

In order to consider this case the auxiliary variable Δ is introduced in the elements k_{ij} in Appendix C. It is

$$\Delta = 1$$

for measuring of x_3 and x_4 and

$$\Delta = 0$$

for nonmeasuring of x_3 and x_4. By this the stated expressions are valid for both cases. In the elements of the Kalman matrix is evident that these are not explicitly dependent on the relations (4.8) but depend only on the ratios

$$\rho^2 = \left(\frac{r_{11}}{\sigma_{11}}\right)^2 \ , \ \lambda^2 = \left(\frac{r_{11}}{\sigma_{33}}\right)^2 \ , \ \gamma^2 = \left(\frac{r_{11}}{r_{33}}\right)^2 \quad . \tag{4.13}$$

This simplifies considerably the choice of appropriate values especially for $\underline{P}(0)$.

In order to have an idea of the time dependance of the $k_{ij}(i=1,\ldots,4;\ j=1,\ldots,4)$, in Fig. 10 and Fig. 11 the amplification $k_{11}(t)$ is drawn exemplarily above the time. Parameter is the ratio ρ^2. In Fig. 10 there exists no measurement of x_3 and x_4, i. e. $\Delta = 0$, in Fig. 11 with measurement of x_3 and x_4 is $\Delta = 1$. At first it is evident that for small values of ρ^2, which means a very good measurement with little variance, the initial amplification is very high. This is obvious. The filter namely considers the measurement more than the initial condition $\underline{x}_0$, which is only known with a certain probability. For great values of ρ^2 the opposite consideration is valid. Furthermore it is evident in comparison of Fig. 10 with Fig. 11 between each other that in the case $\Delta = 0$ the amplifications are greater than in the case $\Delta = 1$. This is obvious, too, as in the first case, because of the missing measurement of velocity, the whole information is included in the measurement data of the position coordinates (x_1,x_2), with the help of which the velocity coordinates $\hat{x}_3$ and $\hat{x}_4$ have to be generated. The efficiency of the filter can be seen in Fig. 12. In this figure it is referred to the example treated in chapter 3. It is an interception considered with $\delta = 0$ for a known target trajectory. The measurement data $y_1,\ldots,y_4$ are disturbed with a known measurement noise of the navigation instruments. These disturbed state variables yield the optimal control $\hat{u}_1(t)$ in Fig. 12a. Following to this the measurement data $y_1,\ldots,y_4$ are fed to the Kalman filter, this generates $\hat{x}_1,\ldots,\hat{x}_4$ and by these the optimal control is realized. This shows Fig. 12b. One sees clearly that the filter adapts to the mean value $E[\underline{w}(t)] = \underline{0}$ of the disturbances and gives altogether a very good result.

4.2 Passive Tracking

For the destination of the target data needed in the control law, i. e. the state variables $\xi_1(t),\ldots,\xi_4(t)$ one can go two different ways. On one hand the variables $\xi_1(t),\ldots,\xi_4(t)$ can be determinated directly by active bearing measurements (active tracking). On the other hand it is also possible to use only passive bearing measurements (passive tracking). From this one gets the relative coordinates $e_1(t),\ldots,e_4(t)$. As the state vector $\underline{x}$ is known from chapter 4.1 also in this case the inertial coordinates $\xi_1(t),\ldots,\xi_4(t)$ can be stated by Eq. (2.8). In the further considerations this way is gone.

For the passive target tracking it is useful to change from a continuous consideration to a discrete one. This has advantages for the algorithms to be developed. For this it is necessary to write the equations of motion (2.4) of the torpedo in a discrete form. With the sampling time T_0 that means

$$\begin{bmatrix} x_1[(k+1)\cdot T_0] \\ x_2[(k+1)\cdot T_0] \\ x_3[(k+1)\cdot T_0] \\ x_4[(k+1)\cdot T_0] \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{1}{\varepsilon}\cdot(1-e^{-\varepsilon\cdot T_0}) & 0 \\ 0 & 1 & 0 & \frac{1}{\varepsilon}\cdot(1-e^{-\varepsilon\cdot T_0}) \\ 0 & 0 & e^{-\varepsilon\cdot T_0} & 0 \\ 0 & 0 & 0 & e^{-\varepsilon\cdot T_0} \end{bmatrix} \cdot \begin{bmatrix} x_1(k\cdot T_0) \\ x_2(k\cdot T_0) \\ x_3(k\cdot T_0) \\ x_4(k\cdot T_0) \end{bmatrix} + \tilde{\underline{c}}(k\cdot T_0) \tag{4.14a}$$

$$\tilde{\underline{c}}(k\cdot T_o) = \begin{bmatrix} \frac{1}{\varepsilon^2}\cdot(1-e^{-\varepsilon\cdot T_o}-\varepsilon\cdot T_o) & 0 \\ 0 & \frac{1}{\varepsilon^2}\cdot(1-e^{-\varepsilon\cdot T_o}-\varepsilon\cdot T_o) \\ \frac{1}{\varepsilon}\cdot(e^{-\varepsilon\cdot T_o}-1) & 0 \\ 0 & \frac{1}{\varepsilon}\cdot(e^{-\varepsilon\cdot T_o}-1) \end{bmatrix} \cdot \begin{bmatrix} \tilde{u}_1(k\cdot T_o) \\ \tilde{u}_2(k\cdot T_o) \end{bmatrix} + \begin{bmatrix} v_1\cdot T_o \\ v_2\cdot T_o \\ 0 \\ 0 \end{bmatrix}, \quad k = 0,1,2,\ldots \qquad (4.14b)$$

The dynamics of the torpedo is consequently described by the general difference equation

$$\underline{x}[(k+1)\cdot T_o] = \underline{\phi}(T_o)\cdot\underline{x}(k\cdot T_o) + \tilde{\underline{B}}(T_o)\cdot\tilde{\underline{u}}(k\cdot T_o) + \underline{v}(T_o) \quad . \qquad (4.15)$$

For the target (ship) the considerations have to be improved in some details.

Case 1: Assuming the ship is moving with constant velocity, the discrete equations of motion for the target are given by

$$\begin{bmatrix} \xi_1[(k+1)\cdot T_o] \\ \xi_2[(k+1)\cdot T_o] \\ \xi_3[(k+1)\cdot T_o] \\ \xi_4[(k+1)\cdot T_o] \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_o & 0 \\ 0 & 1 & 0 & T_o \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \xi_1(k\cdot T_o) \\ \xi_2(k\cdot T_o) \\ \xi_3(k\cdot T_o) \\ \xi_4(k\cdot T_o) \end{bmatrix} + \begin{bmatrix} v_1\cdot T_o \\ v_2\cdot T_o \\ 0 \\ 0 \end{bmatrix}, \quad k = 0,1,2,\ldots, \qquad (4.16)$$

where now $\xi_3(t)$ and $\xi_4(t)$ are relative velocities in accordance with $x_3(t)$ and $x_4(t)$. Inserting Eq. (4.14) and (4.16) in Eq. (2.8) one gets as relative motion between the torpedo and the ship

$$\begin{bmatrix} e_1[(k+1)\cdot T_o] \\ e_2[(k+1)\cdot T_o] \\ e_3[(k+1)\cdot T_o] \\ e_4[(k+1)\cdot T_o] \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_o & 0 \\ 0 & 1 & 0 & T_o \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e_1(k\cdot T_o) \\ e_2(k\cdot T_o) \\ e_3(k\cdot T_o) \\ e_4(k\cdot T_o) \end{bmatrix} + \underline{z}(k\cdot T_o), \quad k = 0,1,2,\ldots, \qquad (4.17)$$

which is identical with the control deviation $\underline{e}(t)$ in Eq. (2.8). In Eq. (4.17) $\underline{z}(k\cdot T_o)$ is a known control vector given explicitly by

$$\underline{z}(k\cdot T_o) = \begin{bmatrix} \frac{1}{\varepsilon}\cdot(-1+e^{-\varepsilon\cdot T_o}+\varepsilon\cdot T_o)\cdot[x_3(k\cdot T_o)+\frac{1}{\varepsilon}\cdot\tilde{u}_1(k\cdot T_o)] \\ \frac{1}{\varepsilon}\cdot(-1+e^{-\varepsilon\cdot T_o}+\varepsilon\cdot T_o)\cdot[x_4(k\cdot T_o)+\frac{1}{\varepsilon}\cdot\tilde{u}_2(k\cdot T_o)] \\ (1-e^{-\varepsilon\cdot T_o})\cdot[x_3(k\cdot T_o)+\frac{1}{\varepsilon}\cdot\tilde{u}_1(k\cdot T_o)] \\ (1-e^{-\varepsilon\cdot T_o})\cdot[x_4(k\cdot T_o)+\frac{1}{\varepsilon}\cdot\tilde{u}_2(k\cdot T_o)] \end{bmatrix} . \qquad (4.18)$$

Consequently, the relative motion is described by the general linear difference equation

$$\underline{e}[(k+1)\cdot T_o] = \underline{\Gamma}(T_o)\cdot\underline{e}(k\cdot T_o) + \underline{z}(k\cdot T_o) \quad . \qquad (4.19)$$

Case 2: The assumption of constant ship velocity is in some cases not fulfilled. Indeed it is sufficient to restrict for variable velocity on linear changes. With this assumption Eq. (4.16) turnes in

$$\begin{bmatrix} \xi_1[(k+1)\cdot T_o] \\ \xi_2[(k+1)\cdot T_o] \\ \xi_3[(k+1)\cdot T_o] \\ \xi_4[(k+1)\cdot T_o] \\ \xi_5[(k+1)\cdot T_o] \\ \xi_6[(k+1)\cdot T_o] \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_o & 0 & \frac{T_o^2}{2} & 0 \\ 0 & 1 & 0 & T_o & 0 & \frac{T_o^2}{2} \\ 0 & 0 & 1 & 0 & T_o & 0 \\ 0 & 0 & 0 & 1 & 0 & T_o \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \xi_1(k\cdot T_o) \\ \xi_2(k\cdot T_o) \\ \xi_3(k\cdot T_o) \\ \xi_4(k\cdot T_o) \\ \xi_5(k\cdot T_o) \\ \xi_6(k\cdot T_o) \end{bmatrix} + \begin{bmatrix} v_1\cdot T_o \\ v_2\cdot T_o \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} . \qquad (4.20)$$

In Eq. (4.20) the additional state variables ξ_5 and ξ_6 are constant parameters to be identified. The difference equation for the relative vector $\underline{e}(\tilde{t})$, which is now of the dimension (6x1) follows from Eq. (4.20) in the same way as in the case before; $\underline{F}(T_o)$ is the system matrix from (4.20) and the control vector $\underline{z}(k \cdot T_o)$ derives from (4.18).

As measurement equation for the generation of the target trajectory, which means the destination of the vector $\underline{e}(k \cdot T_o)$ only the bearing angle

$$\beta(k \cdot T_o) = \tan^{-1}\left[\frac{e_1(k \cdot T_o)}{e_2(k \cdot T_o)}\right] + \eta(k \cdot T_o) \tag{4.21}$$

is available [8]. Thereby $\eta(k \cdot T_o)$ is a white noise process with normal distribution

$$\begin{aligned} &E[\eta(k \cdot T_o)] = 0 \\ &E[\eta(j \cdot T_o) \cdot \eta(k \cdot T_o)] = \begin{cases} \sigma^2(k \cdot T_o) & ; \; j = k \\ 0 & ; \; j \neq k \end{cases}, \quad k = 0,1,2,\ldots \end{aligned} \tag{4.22}$$

The measurement equation (4.21) is nonlinear. From this equation in connection with Eq. (4.19) the vector $\underline{e}(k \cdot T_o)$ has to be determined. If one excludes pathological cases [8] this can be done in a deterministic way, i. e. $\eta(k \cdot T_o) \equiv 0$ for Eq. (4.17) with four, and for Eq. (4.20) with six measurements. Necessary condition for the observability is thereby a sufficient course changing of the pursuer in order to have enough information on the target for solving Eq. (4.19). In the stochastic case as well at least four measurements are needed to solve (4.17) and six to solve (4.20), respectively, but in general it is better to have more in order to carry out a data preprocessing. The stochastic measurements of Eq. (4.21) have to be prepared by a dynamic filter. Following to the considerations in [8] and [12] one then obtains an extended Kalman filter belonging to (4.19) and given in Appendix D. This filter has been derived for the measurement Eq. (4.17), for Eq. (4.20) it has to be extended correspondingly. The filter algorithm itself follows from a linearization of the measurement equation about the momentary estimation vector $\hat{\underline{e}}(k+1|k)$. Now, simulations show that the derived algorithm has an inefficient convergence behaviour, especially in respect of the covariance matrix $\underline{P}(k|k)$. This bases on the dependence of the linearized measurement vector $\underline{c}(k)$ on the estimated state vector $\hat{\underline{e}}(k|k-1)$, by which a feed-back occurs in the calculation of the amplification vector $\underline{k}(k)$, which represents here the Kalman matrix. A decoupling of state estimation and covariance matrix calculation can be reached in a simple way, if one submits the measurement equation to a pseudo-linearization [8], [12]. Then one gets the following algorithm with better convergence behaviour:

Prediction Phase:

$$\begin{aligned} &\hat{\underline{e}}(k+1|k) = \underline{F}(k+1,k) \cdot \hat{\underline{e}}(k,k) + \underline{z}(k) \\ &\hat{\underline{P}}(k+1|k) = \underline{F}(k+1,k) \cdot \hat{\underline{P}}(k|k) \cdot \underline{F}^T(k+1,k) \quad , \quad k = 0,1,2,\ldots \end{aligned} \tag{4.23}$$

Measurement Phase:

$$\hat{\underline{c}}^T(k+1) = [\cos\beta(k+1) \; , \; -\sin\beta(k+1), \; 0, \; 0] \tag{4.24}$$

Correction Phase:

$$\begin{aligned} \hat{\underline{k}}(k+1) &= \hat{\underline{P}}(k+1|k) \cdot \hat{\underline{c}}(k+1) \cdot \left\{\hat{\underline{c}}^T(k+1) \cdot \hat{\underline{P}}(k+1|k) \cdot \hat{\underline{c}}(k+1) + \sigma^2(k+1)\right\}^{-1} \\ \hat{\underline{e}}(k+1|k+1) &= \hat{\underline{e}}(k+1|k) - \hat{\underline{k}}(k+1) \cdot \hat{\underline{c}}^T(k+1) \cdot \hat{\underline{e}}(k+1|k) \\ \hat{\underline{P}}(k+1|k+1) &= \hat{\underline{P}}(k+1|k) - \hat{\underline{k}}(k+1) \cdot \hat{\underline{c}}^T(k+1) \cdot \hat{\underline{P}}(k+1|k) \end{aligned} \tag{4.25}$$

Initial Condition:

$$\begin{aligned} &\hat{\underline{e}}(0|0) = \underline{0} \\ &\hat{\underline{P}}(0|0) = \mathrm{diag}(\hat{\sigma}_o^2) \; . \end{aligned} \tag{4.26}$$

In this algorithm, as well as in the extended Kalman filter in Appendix D, the argument T_o is omitted. The variance $\hat{\sigma}_o^2$ in Eq. (4.26) can be set in general equal to one.

The algorithm given by (4.23) to (4.26) is now used to generate the target trajectory

$$\xi_1(t) = 1 \cdot t$$

$$\xi_2(t) = 0$$

for $t \geq 0$. The pursuer (torpedo) thereby drives the following identification trajectory with the initial conditions $x_{10} = 1{,}12$; $x_{20} = 4{,}50$:

$$0 \leq t \leq 1\text{:} \quad x_1(t) = x_{10} = \text{const.}$$

$$x_2(t) = -1 \cdot t + x_{20} \quad ,$$

$$1 \leq t \leq 2: \quad x_1(t) = 1 \cdot (t-1) + x_{10}$$

$$x_2(t) = x_2(1) = \text{const.}$$

$$2 \leq t \leq 3: \quad x_1(t) = x_1(2) = \text{const.}$$

$$x_2(t) = -1 \cdot (t-2) + x_2(1)$$

$$3 \leq t \leq 4: \quad x_1(t) = 1 \cdot (t-3) + x_1(2)$$

$$x_2(t) = x_2(3) = \text{const.}$$

$$4 \leq t \leq 5: \quad x_1(t) = x_1(4) = \text{const.}$$

$$x_2(t) = -1 \cdot (t-4) + x_3(3) \quad .$$

Consequently the pursuer dynamics is assumed as an ideal rectangular trajectory leading to certain simplifications in Eq. (4.18). As sampling time $T_0 = 1/12$ is chosen, i. e. there are 12 scanning steps within each course correction of the pursuer. By this we get the possibility of a preprocessing in the algorithm (4.23) to (4.26). The results of the simulation are given in Fig. 13 to Fig. 15 for the standard deviations of the measurement noise of $\sigma = (0,5^0; 1^0; 2^0)$. In Fig. 13 the relative distance error $\Delta R = (\hat{R}-R)/R$ is drawn in per cent with $R = (e_1^2+e_2^2)^{1/2}$. Fig. 14 shows the course error $\Delta\theta = \hat{\theta}-\theta$, and in Fig. 15 the velocity error $\Delta v_s = \hat{v}_s - v_s$ is assigned with $v_s = (\xi_3^2+\xi_4^2)^{1/2}$. The geometric configuration of the parameters R, θ and v_s is explained in Fig. 13 again.

From all three figures it is evident that the filter is not working until the first course correction of the torpedo. This bases on the fact - as mentioned above - that the problem is not observable before [8]. After the first course correction the filter gives usable results although it cannot be overlooked that the filter would diverge without a further course correction. This bases especially on the parallel course of torpedo and ship during the second motion interval $1 \leq t \leq 2$. This divergence can be removed by the second course correction of the torpedo. After this the stationary accuracy is reached in the relative distance and in the velocity. The accuracy can be raised in the course by another course correction.

5. CONTROL AND FILTERING

It is obious to combine the results of chapter 3 and chapter 4 in order to carry out a closed loop control-filter procedure. This is done by a hybrid simulation on the basis of the simulation plan of Fig. 16.

On the top of the figure the dynamics of the torpedo is given as derived in chapter 2 inclusive the Kalman filter of chapter 4.1. Below the target filter is outlined and the nonlinear measurement equation for the relative target data and the controllers are given. The two systems - torpedo (pursuer) and ship (target) - are connected by a sample and hold circuit, which carries over the continuous system of the torpedo in a discrete one, as it is needed for the target filter.

In a first phase the target trajectory has to be generated. In this identification phase the switch between the two controllers is turned to the left (1) and the system is working in an open loop. If the torpedo has enough information about the target, the identification phase is finished, i. e. the switch between the two controllers is turned to the right (2) and the closed loop control begins. During this phase the target filter is working in parallel. If an additional time-optimization is needed as outlined in chapter 3, the controller which in hardware is a microprocessor or a minicomputer, respectively, carries out a target trajectory prediction and destinates $\hat{T}$ from Eq. (3.6). In this paper, however, this is not a central point.

As an example of the simulation procedure the results of Fig. 17 are given. Referring to the target trajectory of chapter 3.2.2, where the target is moving on the x_1-ξ_1-axis with the normalized constant velocity "one" the pursuer is started with the initial conditions $x_1(0) = 1,12$; $x_2(0) = 4,50$, the variance of the measurement device is $\sigma^2 = 1^0$. In the identification phase $0 \leq t \leq 3$ the target trajectory is generated. In order to look for geometric influences two different identification trajectories are chosen. At the time t=3 the control phase begins, it is finished at t=9. The controlled trajectories are comparable with the results in Fig. 9. The obtained accuracy can be considered as satisfied. In the x_1-ξ_1-axis one gets an absolute error of $\delta e_1 = -0,15$ for the upper, and of $\delta e_1 = -0,275$ for the lower pursuer trajectory, in the x_2-ξ_2-axis the corresponding values are $\delta e_2 = -0,025$ and $\delta e_2 = 0,03$, respectively.

6. EXTENSIONS

In chapter 2 a control law is derived which is well appropriate to any target trajectories. In chapter 4 and chapter 5 the considerations are focussed on targets with constant velocities. An extension for linear changes in the target velocity i. e. constant acceleration, is given by Eq. (4.20). In order to prove the efficiency of the algorithms derived including this equation the example of chapter 4.2 is taken but now for

$$\xi_1(t) = 0,5 \cdot t + 0,125 \cdot t^2$$

$$\xi_2(t) \equiv 0 \ .$$

Consequently, the ship now has an acceleration of $\ddot{\xi}_1 = 0{,}25$ in the direction of ξ_1. The identification course of the torpedo is very similar to that given in chapter 4.2. The results of this example are assigned in Fig. 18. Without any detailed discussion it is obvious that the target filter works in this case efficiently, too.

Future efforts of the authors will be concentrated especially on targets with any velocity of time. Furthermore, the theoretical considerations will be concentrated in general on the problem of the best identification trajectory. This may be done by an intensive investigation of the observability matrix.

7. CONCLUSION

The paper presented deals with control and filter problems for the interception of torpedo-ship situations. Central points are engineering applications and not primary theoretical aspects. The restriction on plane motions is not decisive, it is only done for formulas limitations. All analytic results are prepared such, that they can be realized in a simple manner numerically by a microprocessor or a minicomputer, respectively. Some of the calculations thereby can be done off-line which is appropriate for real time situations.

ACKNOWLEDGMENT

The authors are grateful to Lt.j.g. Dipl.-Ing. M. Weingart for his encouragement in working for the theory and the numerical results of chapter 4.2. Furthermore they acknowledge the support given by Mrs. U. Barkmann and Mrs. I. Czechatz, who typed and otherwise prepared the manuscript.

REFERENCES

[1] -: Naval Operations Analysis. Naval Institute Press, Annapolis (Maryland) 1977.

[2] Lunderstädt, R.: Steuerung und Regelung von Unterwasserfahrzeugen. Preprints of the workshop "Regelungssynthese im Zustandsraum" of VDI/VDE-Gesellschaft Meß- und Regelungstechnik, Frankfurt 1977.

[3] Garnell, P.; East, D.J.: Guided Weapon Control Systems. Pergamon Press, London 1977.

[4] Stockum, L.A.: A Study of Guidance Controllers for Homing Missiles. Ph. D. Thesis, The Ohio State University 1974.

[5] Hofmann, W.: Ein vereinheitlichter Zugang zur systematischen Auslegung von Flugkörper-Lenksystemen am Beispiel der Zweipunktlenkung. Preprints of 105. Wehrtechnisches Symposium "Regelungstechnik-Automatisierungstechnik", Mannheim 1979.

[6] Rees, N.W.: An Application of Optimal Control Theory to the Guided Torpedo Problem. Proceedings of JACC (1968).

[7] Hofmann, W.: Ein Ansatz zur Verbesserung des Ablageverhaltens von mit Proportionalnavigation gelenkten Flugkörpern durch Einsatz moderner Regelungsverfahren. Preprints of 105. Wehrtechnisches Symposium "Regelungstechnik-Automatisierungstechnik", Mannheim 1979.

[8] Aidala, V.J.: Kalman Filter Behavior in Bearings-Only Tracking Applications. IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-15, Nr. 1, p. 29-39.

[9] Athans, M.; Falb, P.L.: Optimal Control. McGraw-Hill Book Com., New York 1966.

[10] Lunderstädt, R.: Zur Optimierung ebener Interzeptionsbewegungen. Regelungstechnik (29) 1981, Heft 8 (to appear).

[11] Bryson, A.E.; Ho, Y.C.: Applied Optimal Control. Ginn and Company, Waltham, Mass. 1969.

[12] Weingart, M.: Implementierung und Erprobung eines Kalman-Filters für die Zielbahngenerierung aus Peilmessungen. Diploma Thesis, University of the German Armed Forces, Hamburg 1981.

APPENDIX A

Auxiliary Functions for the Open-loop Control Problem in the Case $\underline{Q} \neq \underline{0}$.

$$
\begin{aligned}
c_{oi}(t) &= ch(\gamma_i \cdot t)\cdot\cos(\omega_i \cdot t) - \frac{(\gamma_i^2-\omega_i^2)}{2\gamma_i\cdot\omega_i}\cdot sh(\gamma_i\cdot t)\cdot\sin(\omega_i\cdot t)\\
c_{1i}(t) &= \frac{\gamma_i\cdot(3\omega_i^2-\gamma_i^2)}{2\gamma_i\cdot\omega_i\cdot(\gamma_i^2+\omega_i^2)}\cdot ch(\gamma_i\cdot t)\cdot\sin(\omega_i\cdot t) + \frac{\omega_i\cdot(3\gamma_i^2-\omega_i^2)}{2\gamma_i\cdot\omega_i\cdot(\gamma_i^2+\omega_i^2)}\cdot sh(\gamma_i\cdot t)\cdot\cos(\omega_i\cdot t)\\
c_{2i}(t) &= \frac{1}{2\gamma_i\cdot\omega_i}\cdot sh(\gamma_i\cdot t)\cdot\sin(\omega_i\cdot t)\\
c_{3i}(t) &= \frac{\gamma_i}{2\gamma_i\cdot\omega_i\cdot(\gamma_i^2+\omega_i^2)}\cdot ch(\gamma_i\cdot t)\cdot\sin(\omega_i\cdot t) - \frac{\omega_i}{2\gamma_i\cdot\omega_i\cdot(\gamma_i^2+\omega_i^2)}\cdot sh(\gamma_i\cdot t)\cdot\cos(\omega_i\cdot t)
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
m_{2i}(t) &= c_{2i}(t) + \varepsilon\cdot c_{3i}(t)\\
m_{1i}(t) &= c_{1i}(t) + \varepsilon\cdot m_{2i}(t)\\
m_{oi}(t) &= c_{o}(t) + \varepsilon\cdot m_{1i}(t)
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
f_{i+2}(t,T) = &\left[m_{2i}(t) - \frac{m_{2i}(T)}{m_{oi}(T)}\cdot m_{oi}(t)\right]\cdot x_{io} + \left[c_{3i}(t) - \frac{c_{3i}(T)}{m_{oi}(T)}\cdot m_{oi}(t)\right]\cdot(x_{i+2,0}+v_i) +\\
&+\left[m_{oi}(t)\cdot(c_{oi}(T)-1) - m_{oi}(T)\cdot(c_{oi}(t)-1)\right]\cdot\frac{\varepsilon}{q_{ii}\cdot m_{oi}(T)}\cdot v_i -\\
&-\left[i_{ui}(t) - \frac{i_{ui}(T)}{m_{oi}(T)}\cdot m_{oi}(t)\right]
\end{aligned}
\tag{A.3}
$$

$$
i_{ui}(t) = \int_0^t m_{1i}(t-\tau)\cdot\xi_i(\tau)\cdot d\tau \quad , \quad i = 1,2 \quad . \tag{A.4}
$$

$$
\begin{aligned}
d_{2i}(t) &= c_{2i}(t) - \varepsilon\cdot c_{3i}(t)\\
d_{1i}(t) &= c_{1i}(t) - \varepsilon\cdot d_{2i}(t)\\
d_{oi}(t) &= c_{oi}(t) - \varepsilon\cdot d_{1i}(t)
\end{aligned}
\tag{A.5}
$$

$$
\begin{aligned}
k_{1i}(T) = &\left[1+q_{ii}\frac{c_{2i}(T)\cdot m_{2i}(T)}{c_{oi}(T)\cdot m_{oi}(T)}\right]\cdot x_{io} + \left[\frac{d_{1i}(T)}{c_{oi}(T)} + q_{ii}\frac{c_{2i}(T)\cdot c_{3i}(T)}{c_{oi}(T)\cdot m_{oi}(T)}\right]\cdot(x_{i+2,0}+v_i) +\\
&+\left[\frac{d_{2i}(T)}{c_{oi}(T)} - (c_{oi}(T)-1)\cdot\frac{c_{2i}(T)}{c_{oi}(T)\cdot m_{oi}(T)}\right]\cdot\varepsilon\cdot v_i + \left[\frac{i_{1i}(T)}{c_{oi}(T)} - \frac{c_{2i}(T)\cdot i_{ui}(T)}{c_{oi}(T)\cdot m_{oi}(T)}\right]\cdot q_{ii} - \frac{1}{c_{oi}(T)}\cdot\xi_i(T)\\
k_{2i}(T) = &\left[\frac{c_{1i}(T)}{c_{oi}(T)} + q_{ii}\cdot\frac{c_{3i}(T)\cdot m_{2i}(T)}{c_{oi}(T)\cdot m_{oi}(T)}\right]\cdot x_{io} + \left[\frac{d_{2i}(T)}{c_{oi}(T)} + q_{ii}\cdot\frac{c_{3i}^2(T)}{c_{oi}(T)\cdot m_{oi}(T)}\right]\cdot(x_{i+2,0}+v_i) +\\
&+\left[\frac{c_{3i}(T)}{c_{oi}(T)} - (c_{oi}(T)-1)\cdot\frac{(c_{3i}(T)-\frac{\varepsilon}{q_{ii}}\cdot m_{oi}(T))}{c_{oi}(T)\cdot m_{oi}(T)}\right]\cdot\varepsilon\cdot v_i - \left[\frac{i_{3i}(T)}{c_{oi}(T)} + q_{ii}\cdot\frac{c_{3i}(T)\cdot i_{ui}(T)}{c_{oi}(T)\cdot m_{oi}(T)}\right]
\end{aligned}
\tag{A.6}
$$

$$
\begin{aligned}
l_{1i}(T) &= \frac{c_{2i}(T)\cdot m_{1i}(T)}{c_{oi}(T)\cdot m_{oi}(T)} - \frac{c_{3i}(T)}{c_{oi}(T)}\\
l_{2i}(T) &= \frac{c_{3i}(T)\cdot m_{1i}(T)}{c_{oi}(T)\cdot m_{oi}(T)}
\end{aligned}
\tag{A.7}
$$

$$i_{1i}(t) = \int_0^t c_{3i}(t-\tau)\cdot\xi_i(\tau)\cdot d\tau$$

$$i_{3i}(t) = \int_0^t c_{oi}(t-\tau)\cdot\xi_i(\tau)\cdot d\tau \quad , \quad i = 1,2 \quad . \tag{A.8}$$

APPENDIX B

Auxiliary Functions for the Closed-loop Control Problem in the Case $\underline{Q} \neq \underline{0}$.

$$r_{oi}(T-t) = \frac{1}{n_i(T-t)}\cdot\Big\{p\cdot\Big[c_{oi}(T-t)\cdot d_{oi}(T-t) + q_{ii}\cdot c_{2i}(T-t)\cdot d_{2i}(T-t)\Big] -$$

$$-(1-\delta)\cdot q_{ii}\cdot\Big[c_{1i}(T-t)\cdot d_{oi}(T-t) + q_{ii}\cdot c_{3i}(T-t)\cdot d_{2i}(T-t)\Big]\Big\}$$

$$n_i(T-t) = c_{oi}(T-t)\cdot d_{oi}(T-t) + \frac{p}{(1-\delta)}\cdot\Big[c_{3i}(T-t)\cdot d_{oi}(T-t) - c_{2i}(T-t)\cdot d_{1i}(T-t)\Big] + \tag{B.1}$$

$$+ q_{ii}\cdot c_{3i}(T-t)\cdot d_{1i}(T-t) \quad , \quad i = 1,2 \quad .$$

$$h_i(T-t) = -\,p\cdot\Big\{c_{oi}(T-t)\cdot d_{oi}(T-t)\cdot m_{1i}(T-t) + q_{ii}\cdot[c_{3i}(T-t)\cdot(c_{oi}(T-t)\cdot c_{2i}(T-t) + d_{oi}(T-t)\cdot m_{2i}(T-t)) +$$

$$+ c_{3i}(T-t)\cdot d_{1i}(T-t)\cdot m_{1i}(T-t) - c_{2i}(T-t)\cdot d_{1i}(T-t)\cdot m_{2i}(T-t) + q_{ii}\cdot c_{3i}^3(T-t)]\Big\}$$

$$g_i(T-t) = \frac{p}{n_i(T-t)}\cdot\Big\{m_{1i}(T-t)\cdot[c_{oi}(T-t)\cdot d_{1i}(T-t) - c_{1i}(T-t)\cdot d_{oi}(T-t)] + \tag{B.2}$$

$$+ q_{ii}\cdot[c_{2i}(T-t)\cdot c_{3i}(T-t)\cdot(d_{1i}(T-t) - c_{1i}(T-t)) + c_{3i}^2(T-t)\cdot(c_{oi}(T-t) - d_{oi}(T-t))]\Big\} +$$

$$+ \frac{(1-\delta)\cdot q_{ii}}{n_i(T-t)}\cdot\Big\{m_{2i}(T-t)\cdot[c_{1i}(T-t)\cdot d_{oi}(T-t) - c_{oi}(T-t)\cdot d_{1i}(T-t)] +$$

$$+ c_{oi}(T-t)\cdot c_{3i}(T-t)\cdot[c_{oi}(T-t) - d_{oi}(T-t)] +$$

$$+ q_{ii}\cdot c_{3i}^2(T-t)\cdot[c_{1i}(T-t) - d_{1i}(T-t)]\Big\} + (1-\delta)\cdot\varepsilon\cdot[c_{oi}(T-t) - 1] \quad , \quad i = 1,2 \quad .$$

$$I_{1i}(t,T) = \int_t^T c_{1i}(t-\tau)\cdot\ddot{\xi}_i(\tau)\cdot d\tau$$

$$I_{2i}(t,T) = \int_t^T c_{oi}(t-\tau)\cdot\ddot{\xi}_i(\tau)\cdot d\tau \tag{B.3}$$

$$I_{3i}(t,T) = \int_t^T \Big[c_{3i}(t-\tau) - \frac{\varepsilon}{q_{ii}}\cdot c_{oi}(t-\tau)\Big]\cdot\ddot{\xi}_i(\tau)\cdot d\tau \quad , \quad i = 1,2 \quad .$$

APPENDIX C

Elements of the Kalman Matrix $\underline{K}(t)$ for the Torpedo Inertial Filter.

$$K(t) = \frac{1}{\varepsilon^4}[2 - 2\cdot ch(\varepsilon\cdot t) + \varepsilon\cdot t\cdot sh(\varepsilon\cdot t)] + \frac{\rho^2}{\varepsilon^3}[2 - ch(\varepsilon\cdot t) + (\varepsilon\cdot t - 1)\cdot e^{\varepsilon\cdot t}] +$$

$$+ \lambda^2\cdot t\cdot e^{\varepsilon\cdot t} + (\rho\cdot\lambda)^2\cdot e^{\varepsilon\cdot t} + \delta\cdot\frac{\lambda^2}{\varepsilon}\cdot(t+\varepsilon^2)\cdot sh(\varepsilon\cdot t) \tag{C.1}$$

$$k_{11}(t) = \frac{1}{N(t)}\cdot\left\{\frac{1}{\varepsilon^3}\cdot\left[-2+ch(\varepsilon\cdot t)+(1+\varepsilon\cdot t)\cdot e^{-\varepsilon\cdot t}\right] + 2\cdot\left(\frac{\rho}{\varepsilon}\right)^2\cdot\left[-1+ch(\varepsilon\cdot t)\right] + \right.$$

$$\left. + \lambda^2\cdot e^{\varepsilon\cdot t} + \Delta\cdot\frac{\gamma^2}{\varepsilon}\cdot sh(\varepsilon\cdot t)\right\}$$

$$k_{13}(t) = \Delta\cdot\gamma^2\cdot k_{31}(t) \qquad (C.2)$$

$$k_{31}(t) = \frac{1}{N(t)}\cdot\left\{\frac{1}{\varepsilon^2}\cdot\left[1-(1+\varepsilon\cdot t)\cdot e^{-\varepsilon\cdot t}\right] + \frac{\rho^2}{\varepsilon}\cdot\left[1-e^{-\varepsilon\cdot t}\right]\right\}$$

$$k_{33}(t) = \Delta\cdot\frac{\gamma^2}{N(t)}\cdot(\rho^2+t)\cdot e^{-\varepsilon\cdot t} \quad .$$

$$k_{22}(t) = k_{11}(t)$$

$$k_{24}(t) = k_{13}(t) \qquad (C.3)$$

$$k_{42}(t) = k_{31}(t)$$

$$k_{44}(t) = k_{33}(t) \quad .$$

APPENDIX D

Extended Kalman Filter for the Relative Motion of Torpedo-Ship.

Prediction Phase:

$$\hat{\underline{e}}(k+1|k) = \underline{F}(k+1,k)\hat{\underline{e}}(k|k) + \underline{z}(k)$$

$$\underline{P}(k+1|k) = \underline{F}(k+1,k)\cdot\underline{P}(k|k)\cdot\underline{F}^T(k+1,k) \quad , \quad k = 0,1,2,... \qquad (D.1)$$

Measurement Phase:

$$\underline{c}^T(k+1) = \left.\frac{\partial h}{\partial \underline{e}}\right|_{\underline{e}=\hat{\underline{e}}(k+1|k)} = \left[\frac{\cos\beta(k+1|k)}{\hat{R}(k+1|k)} \ , \ -\frac{\sin\beta(k+1|k)}{\hat{R}(k+1|k)} \ , \ 0,0\right]$$

$$h = \tan^{-1}\left[\frac{e_1}{e_2}\right] \qquad (D.2)$$

$$\hat{R}^2 = \hat{e}_1^2 + \hat{e}_2^2$$

Correction Phase:

$$\underline{k}(k+1) = \underline{P}(k+1|k)\cdot\underline{c}(k+1)\cdot\left\{\underline{c}^T(k+1)\cdot\underline{P}(k+1|k)\cdot\underline{c}(k+1)+\sigma^2(k+1)\right\}^{-1}$$

$$\hat{\underline{e}}(k+1|k+1) = \hat{\underline{e}}(k+1|k) + \underline{k}(k+1)\cdot\left\{\beta(k+1)-h[\hat{\underline{e}}(k+1|k)]\right\} \qquad (D.3)$$

$$\underline{P}(k+1|k+1) = \underline{P}(k+1|k) - \underline{k}(k+1)\cdot\underline{c}^T(k+1)\cdot\underline{P}(k+1|k)$$

Initial Condition:

$$\underline{P}(0|0) = \mathrm{diag}(\sigma_o^2) \quad ; \quad \sigma_o^2 >> 1$$

$$\hat{\underline{e}}(0|0) = \underline{0} \quad . \qquad (D.4)$$

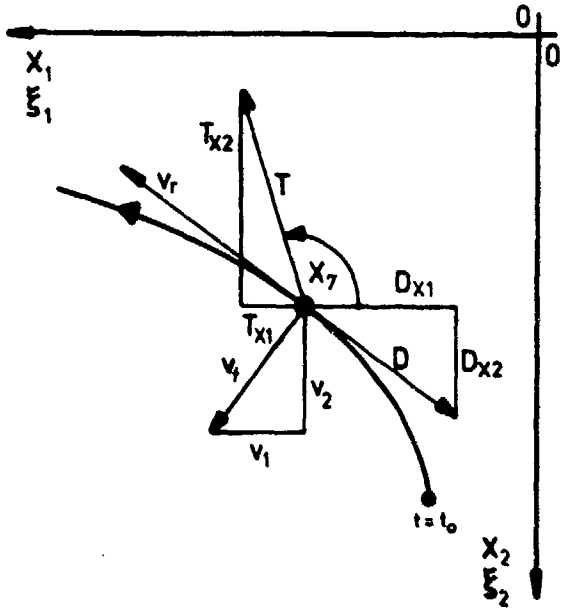


Fig. 1 Kinetics of torpedo

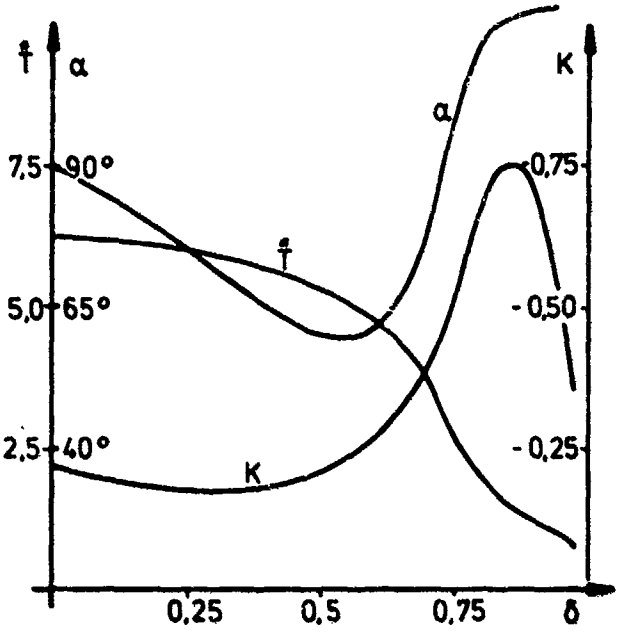


Fig. 2 Optimal constants of open-loop control problem

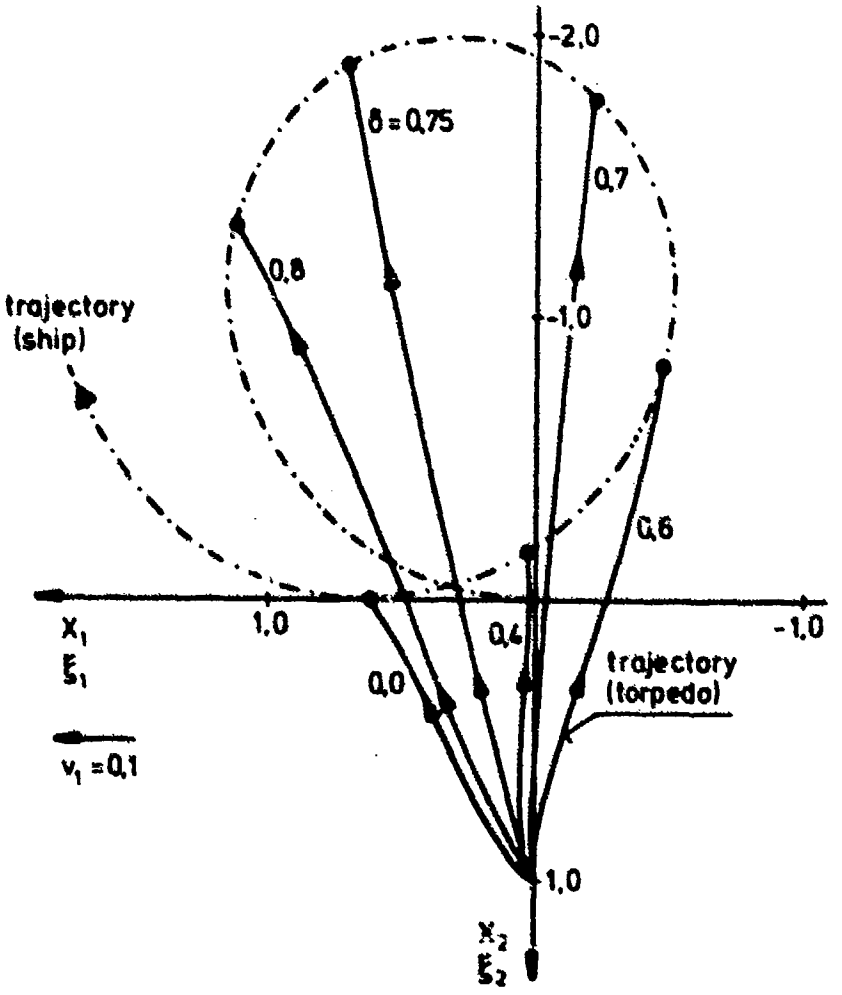


Fig. 3 Optimal interception for torpedo ship (open loop)

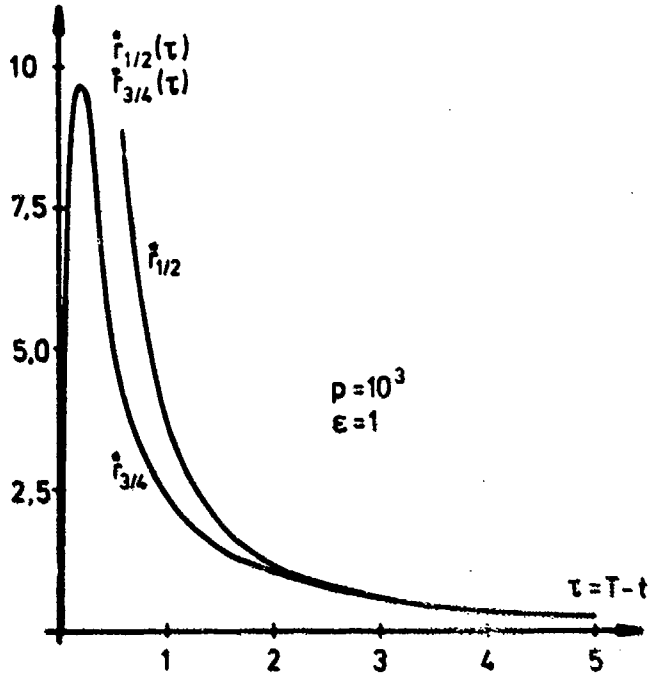


Fig. 4 Coefficients of the control law

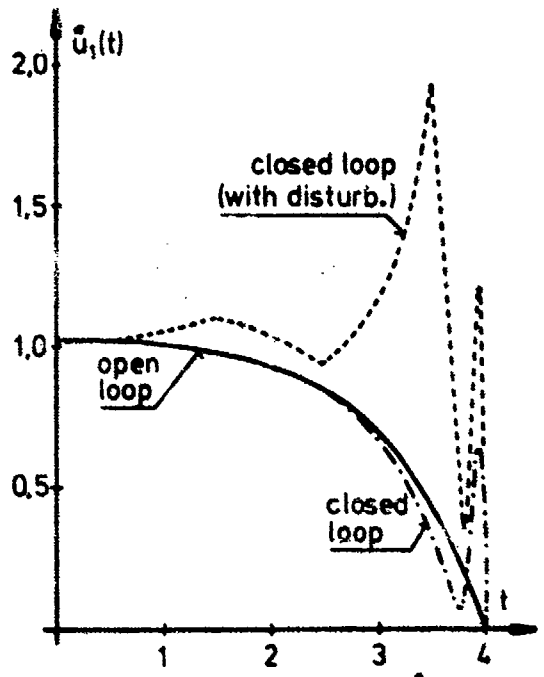
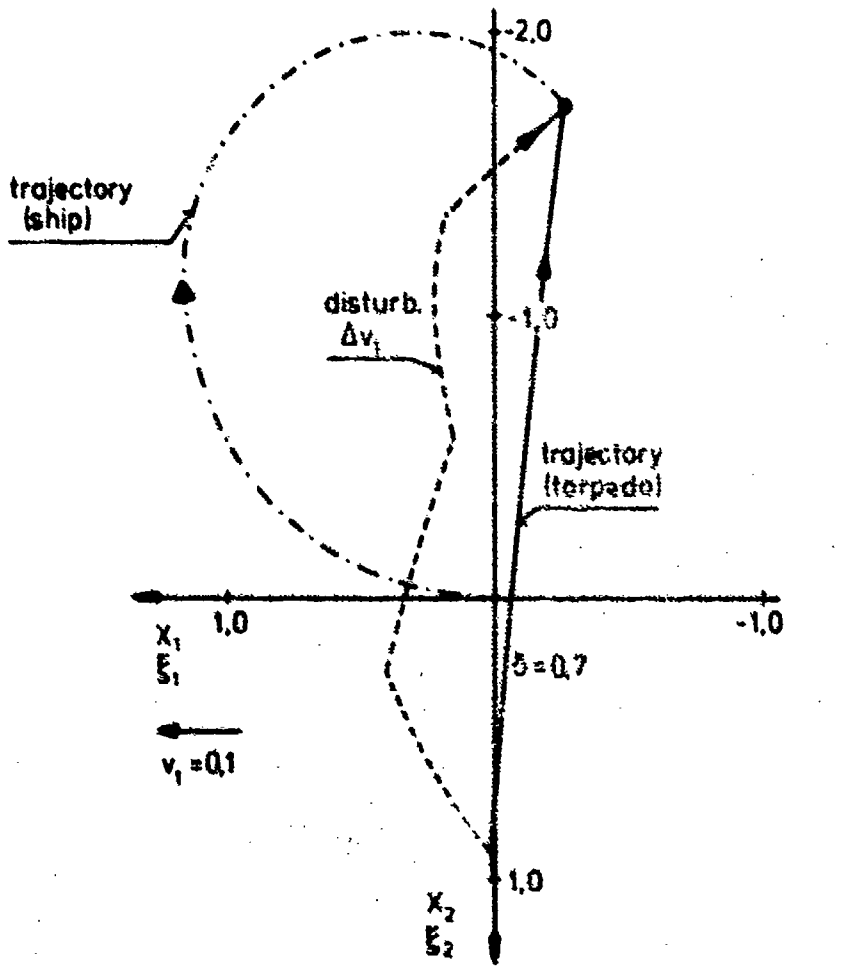


Fig. 5 Optimal control $\hat{u}_1(t)$



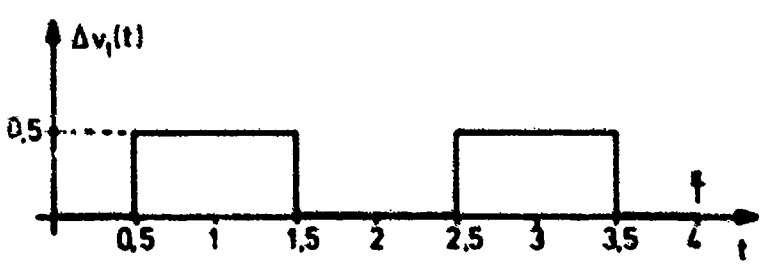


Fig. 6 Optimal interception for torpedo-ship (closed loop)

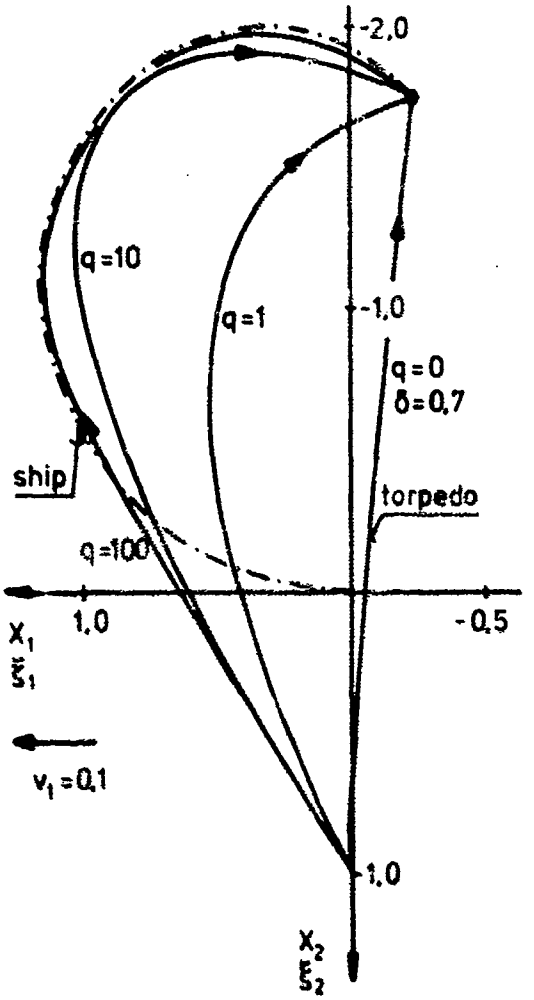


Fig. 7 Optimal interception for $\underline{Q} = \underline{0}$

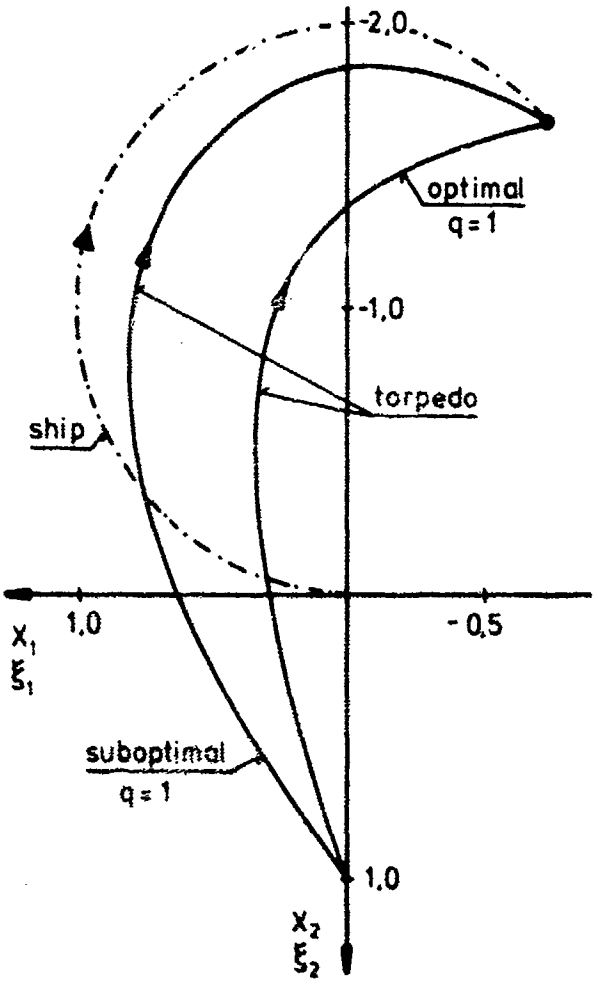


Fig. 8 Optimal and suboptimal interception

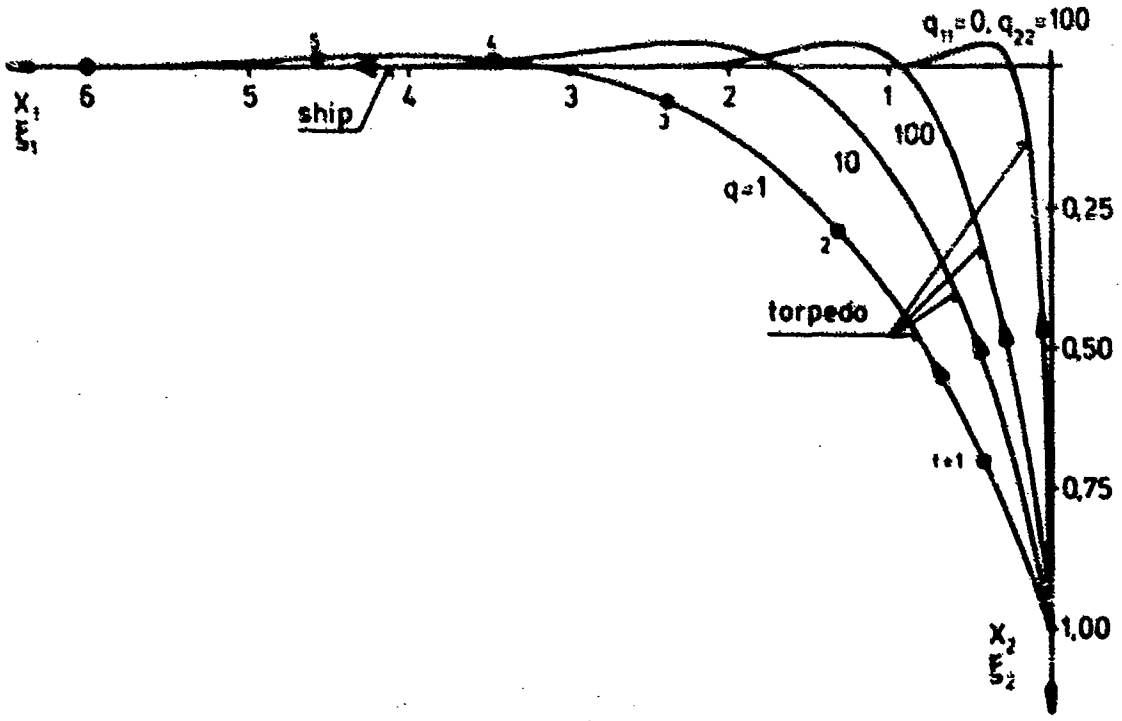


Fig. 9 Optimal interception trajectories

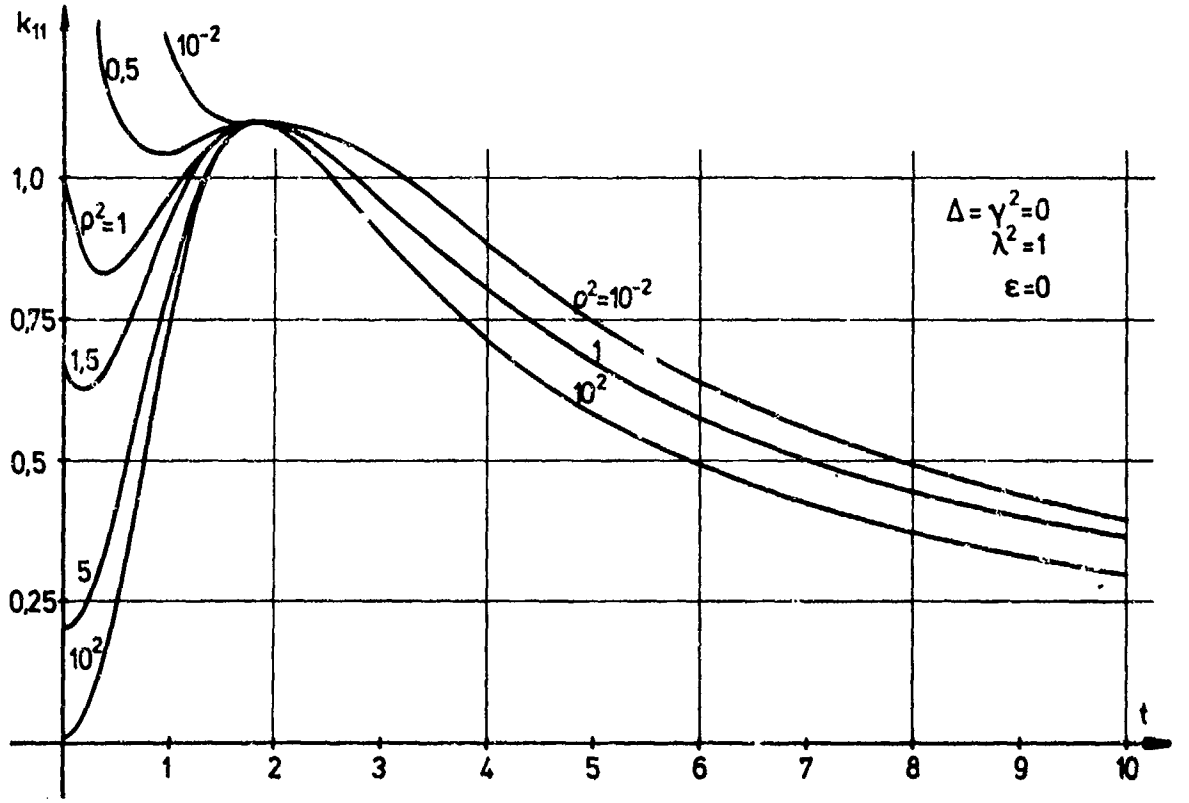


Fig. 10 Amplification $k_{11}(t)$ of Kalman matrix (without velocity measurements)

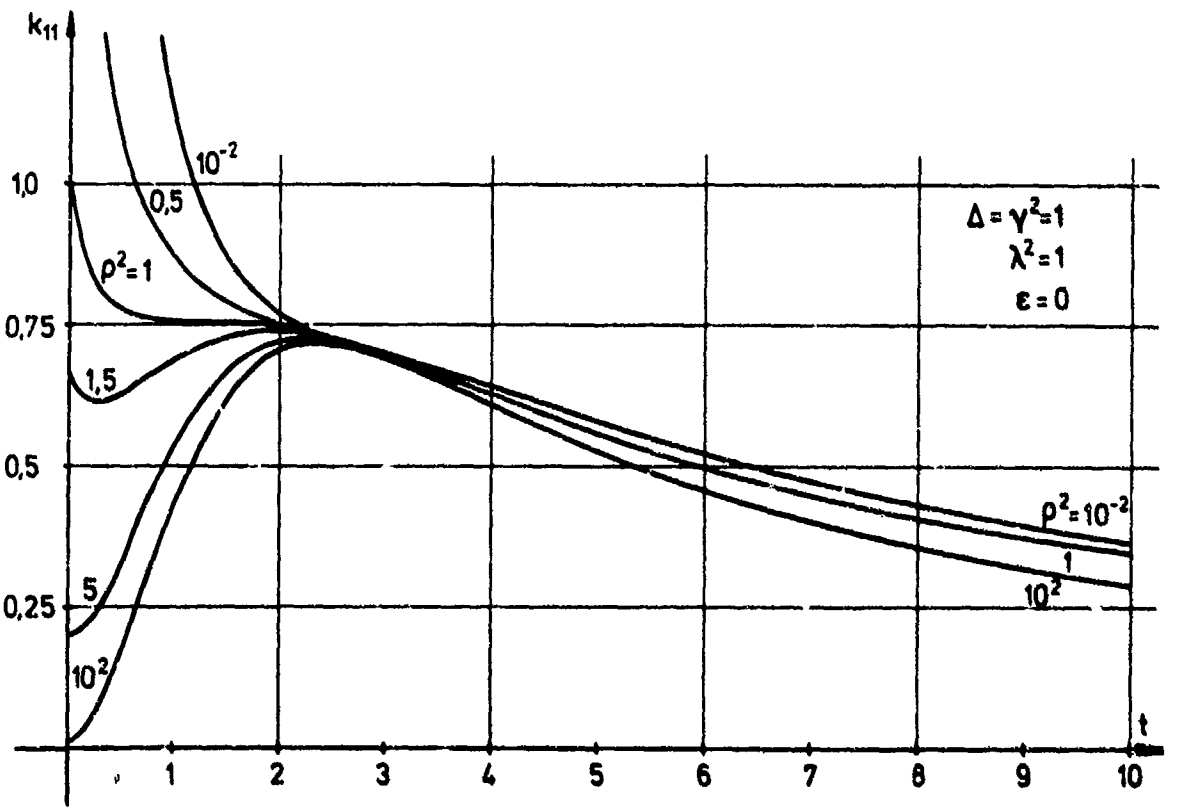


Fig. 11 Amplification $k_{11}(t)$ of Kalman matrix (with velocity measurements)

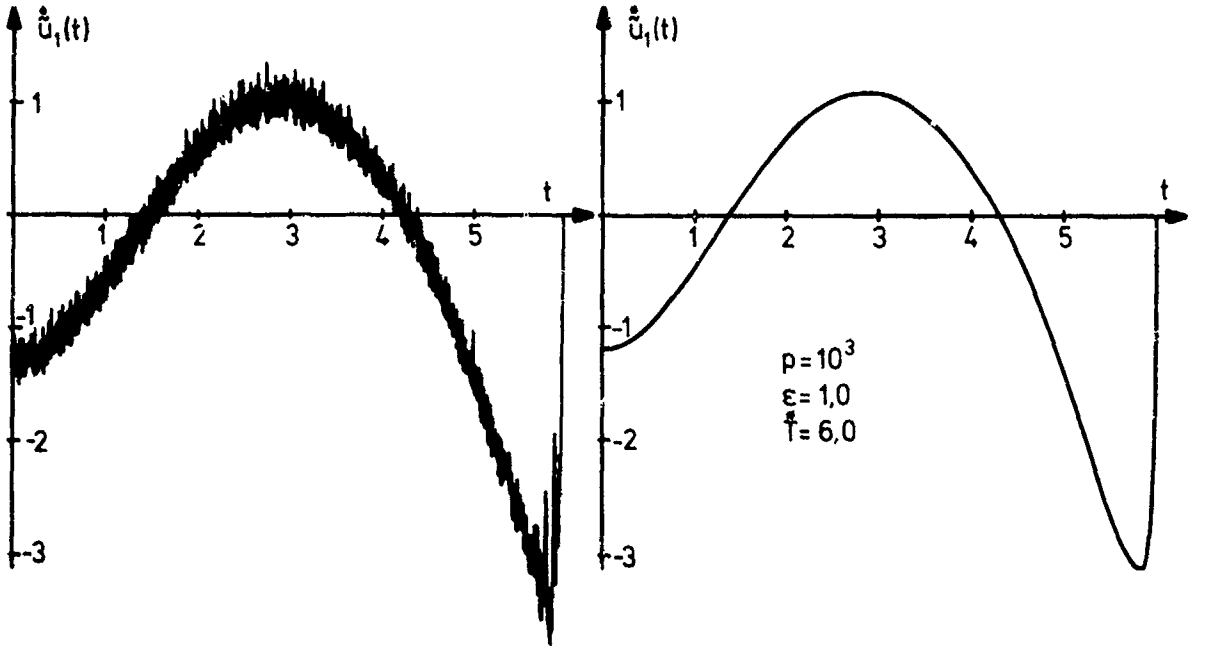


Fig. 12 Optimal control $\tilde{u}_1^*(t)$
a) without Kalman filter b) with Kalman filter

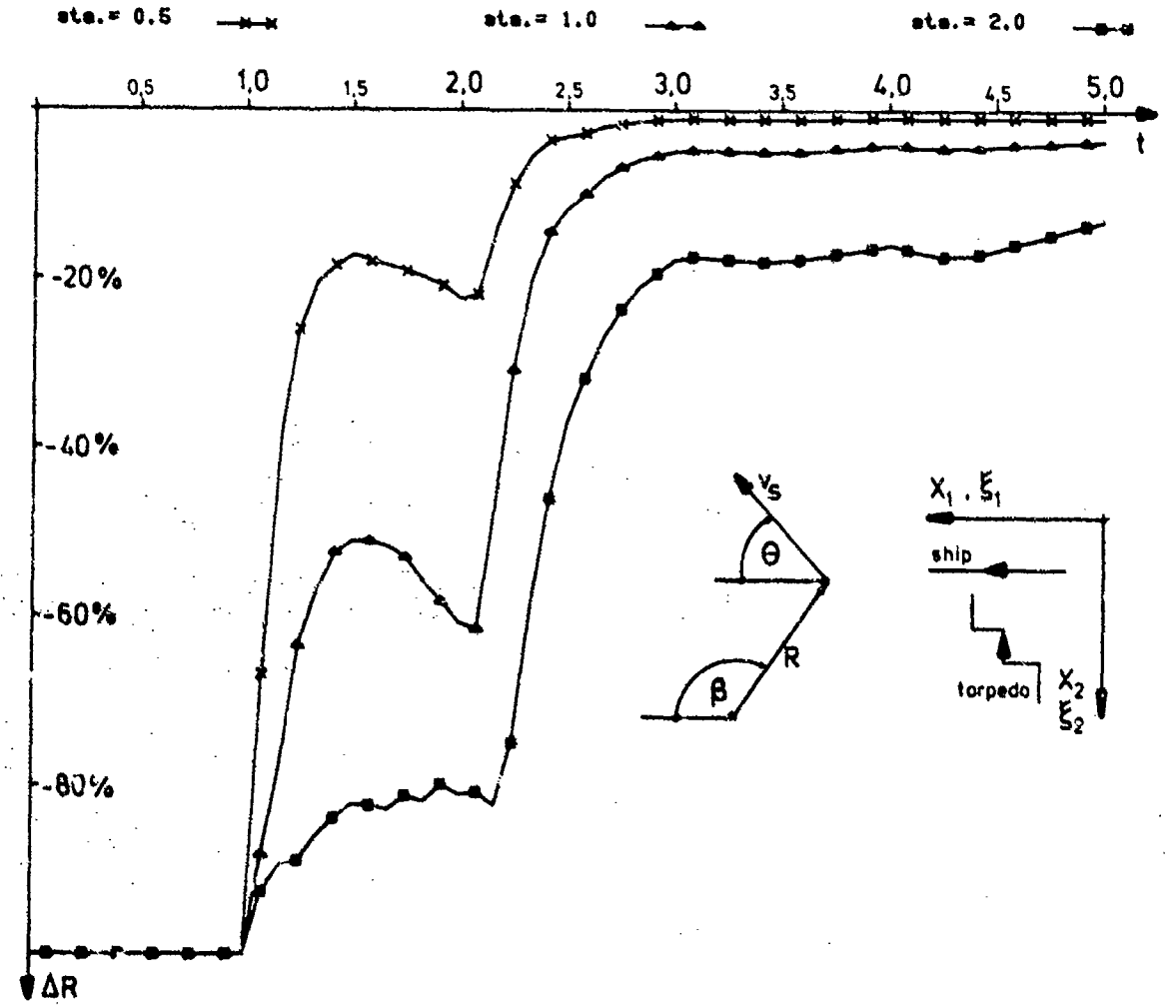


Fig. 13 Target filter: relative distance error

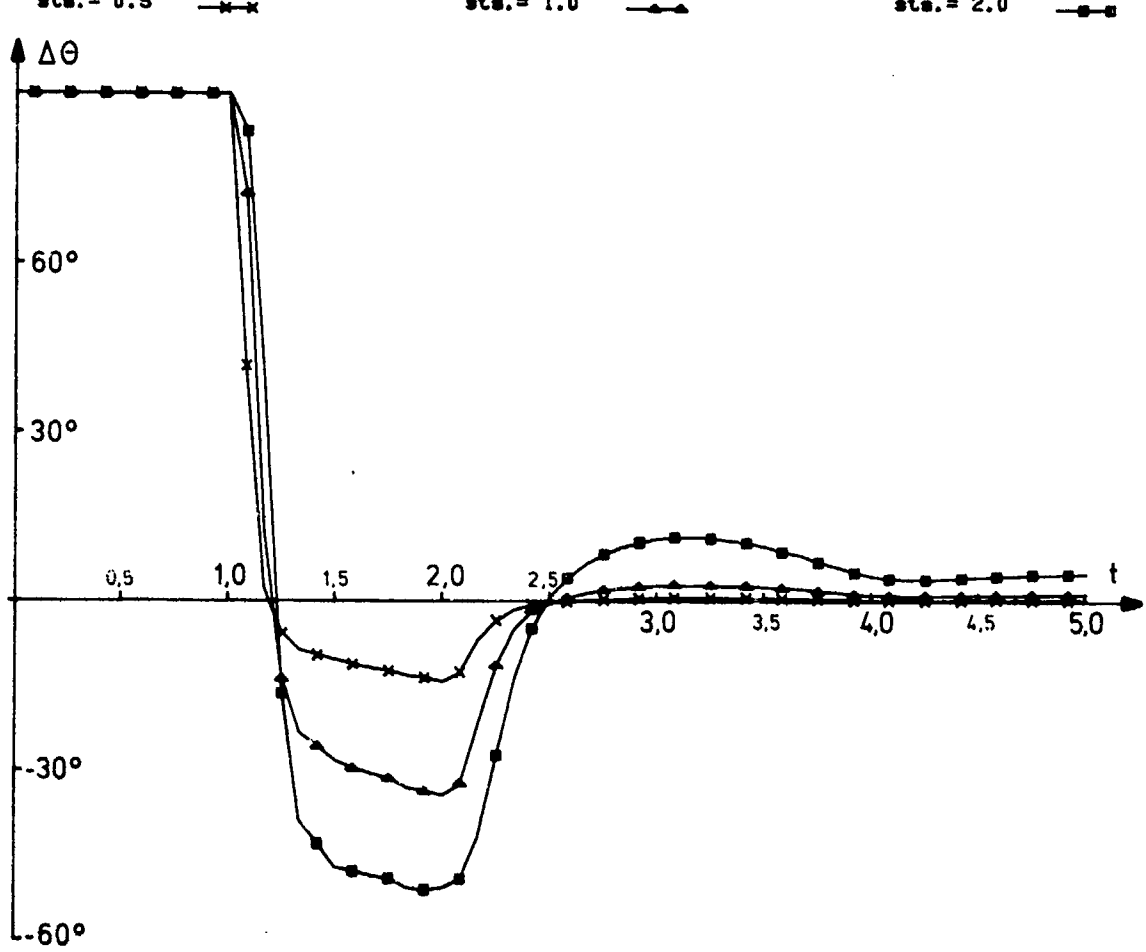


Fig. 14 Target filter: course angle error

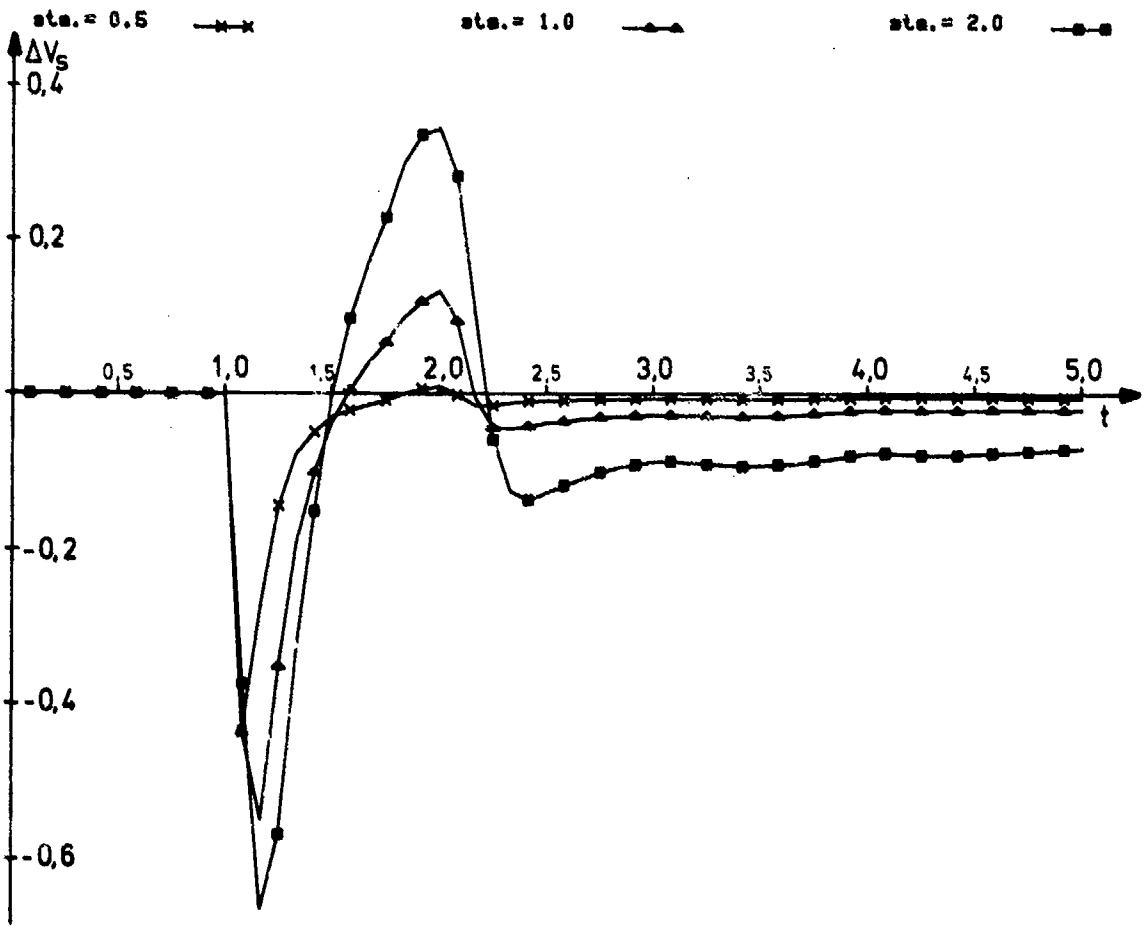


Fig. 15 Target filter: velocity error

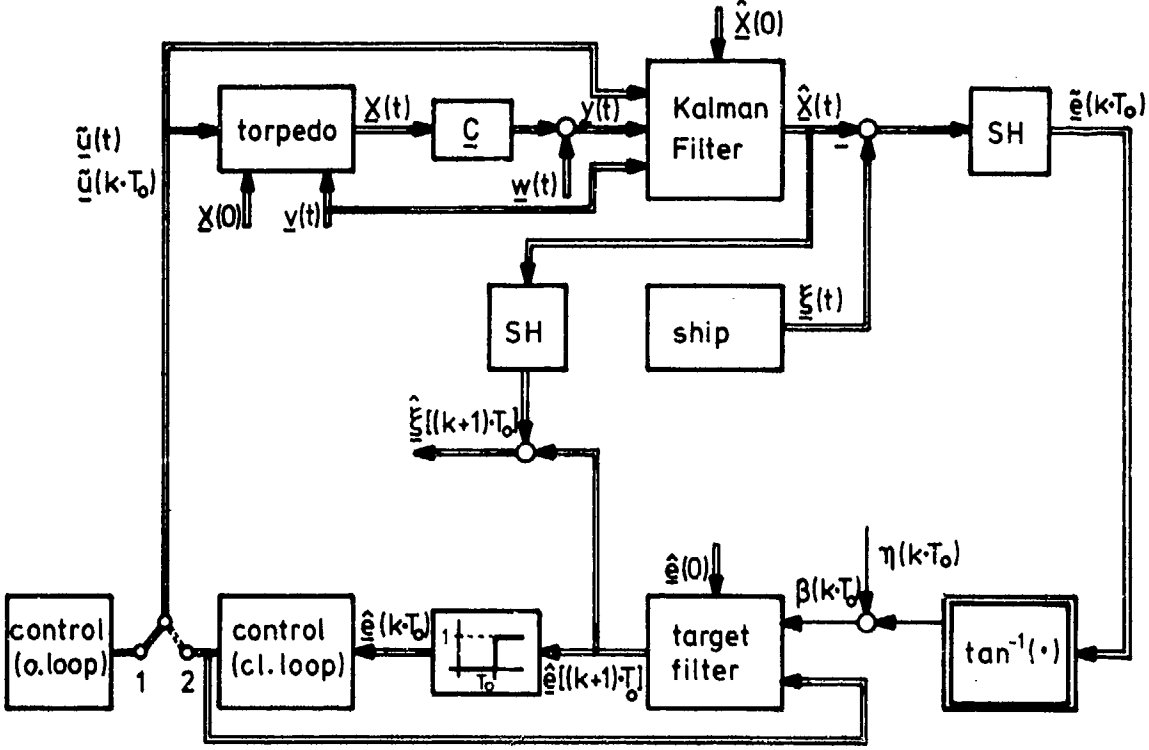


Fig. 16 Simulation plan

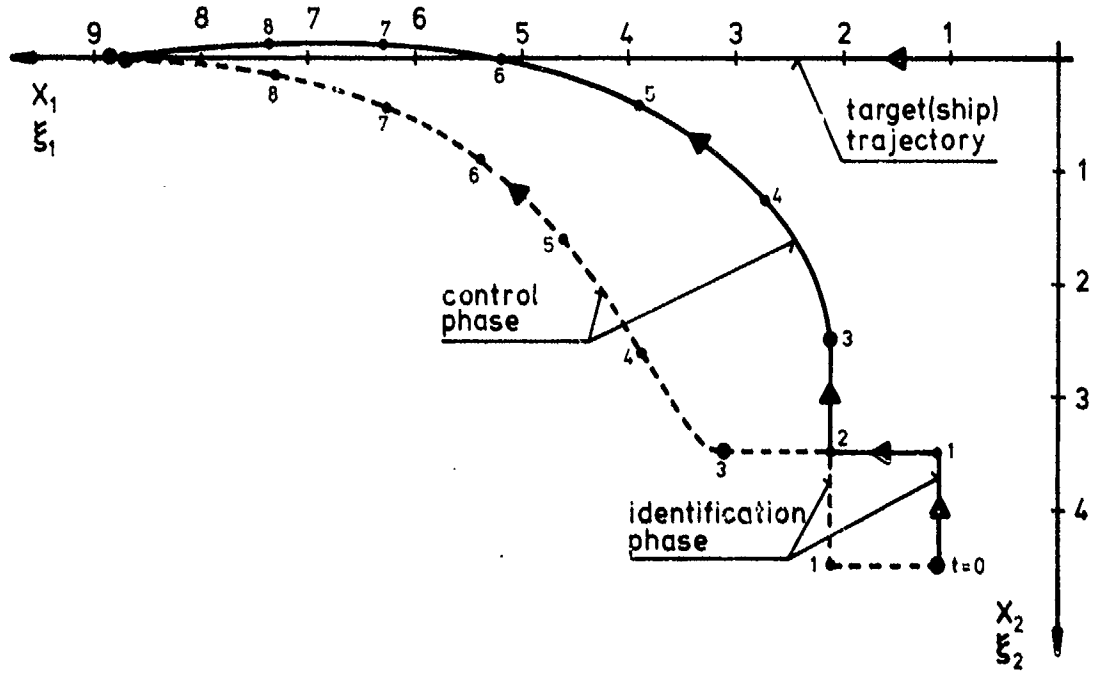


Fig. 17 Interception trajectories with identification and control

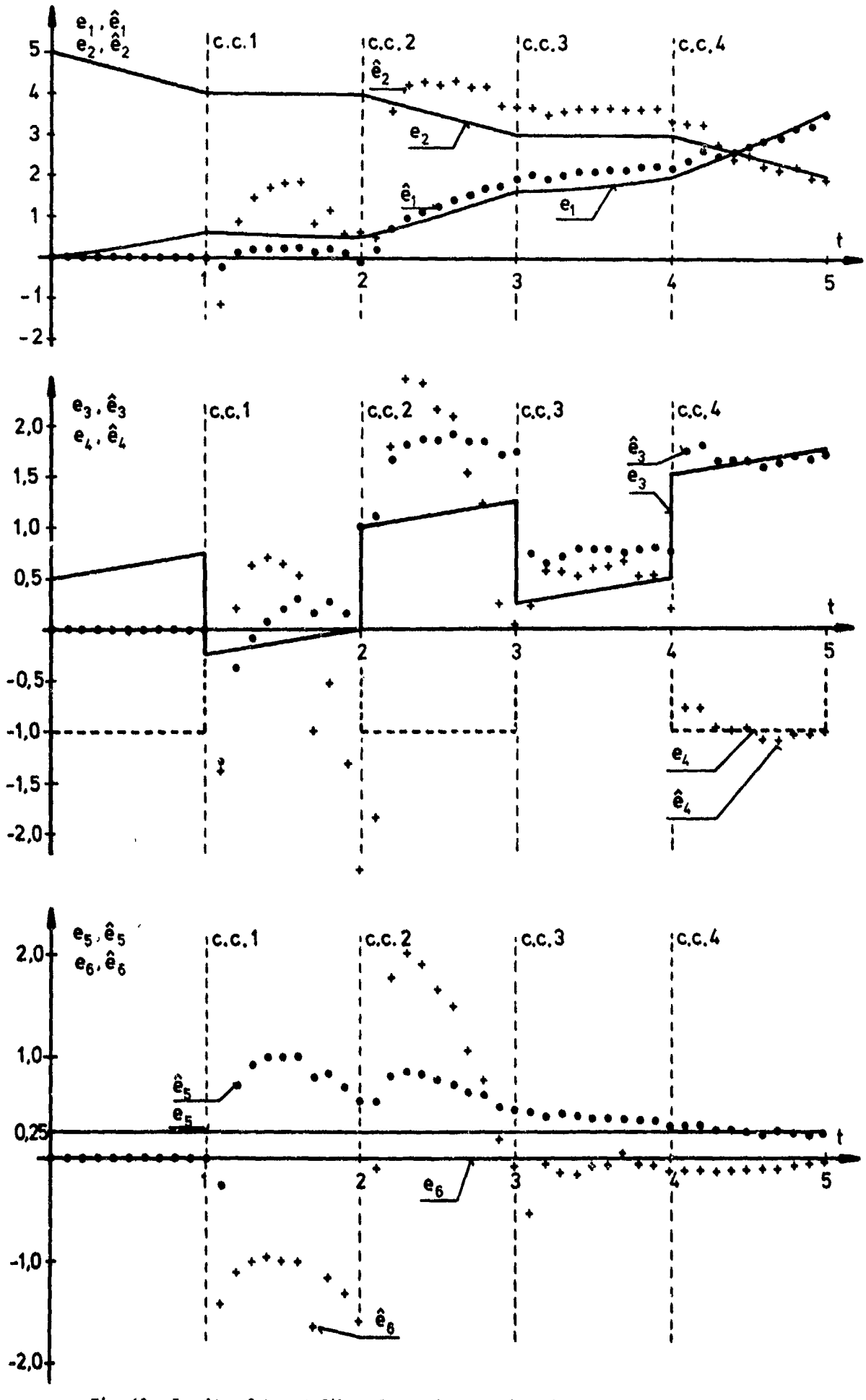


Fig. 18 Results of target filter for an interception with variable target velocity

SEPARATED-BIAS ESTIMATION AND SOME APPLICATIONS

by
Bernard Friedland
The Singer Company, Kearfott Division
1150 McBride Avenue
Little Falls, New Jersey
07424
U.S.A.

SUMMARY

A number of applications of Kalman-Bucy filtering require the estimation of unknown constants (biases) as well as dynamic state variables. In 1969, a method of separating the estimation of the bias vector from the estimation of the dynamic state vector was presented as an alternative to state vector augmentation which often had numerical conditioning difficulties. The optimum estimate $\hat{x}$ of the dynamic state was shown to be the sum of a bias-free estimate $\bar{x}$ (i.e., the estimate that would be obtained if the constant bias vector b were zero) and a correction term $V\hat{b}$, where $\hat{b}$ is the optimum estimate of the bias and V is a matrix determined from the other matrices of the problem. Moreover, the bias estimate $\hat{b}$ can be obtained from the residuals (innovations) of the bias-free estimator. This paper reviews the general theory, using a new derivation, and summarizes some of the extensions that various investigators have contributed during the past decade. Several applications, including calibration and failure detection and identification, are discussed.

1. INTRODUCTION

No contribution since World War II has influenced system science more than the recursive filtering theory of Kalman and Bucy [1,2]. The theoretical significance and practical utility of this work became widely recognized within a few short years of its advent in the early 1960's. Dozens of papers soon appeared which presented alternate derivations and interpretations and demonstrated potential applications in various fields--most notably aerospace but also in industrial process control and even in the field of econometrics.

All this activity made the benefits of Kalman-Bucy filtering quickly evident to a wide audience. And it exposed some of the limitations that were not obvious at first. One of these limitations was the tendency of the calculated quantities (particularly the "covariance matrix") to become ill-conditioned, with the elapse of time, in processes of high dimension. (Considerable research has been devoted to general methods of improving numerical conditioning of the required calculations--and continues to the present time--but this general subject is beyond the scope of this paper.)

Even when ill-conditioning did not cause serious problems, the implementation of the Kalman filter in many instances created a severe burden for the typical airborne computer of the early sixties and motivated a quest for ways to reduce the computational requirements, even at the expense of a sacrifice in the theoretically attainable performance.

The problems of computer loading and prospective ill-conditioning had to be faced in one of the early proposed applications of Kalman filtering: mixing of navigation aid data with inertial data in aided-inertial navigation systems. In this application [3], most of the variables to be estimated are constants (biases, drift rates, scale factor errors, misalignment angles, etc.). The customary treatment of these unknown constants as state variables results in state vectors of high dimension. Around 1969 we reasoned that it should be possible to exploit the fact that many, if not most, of the state variables are constants to reduce the complexity of the filter, and thereby to alleviate the computational burden and to minimize the possibility of ill-conditioning. We initiated an analysis which culminated in our paper [4] in which the estimation of the constant or "bias" parameters was separated from the estimation of the dynamic state variables.

We showed that it is possible to obtain an optimum estimate $\hat{x}$ of the dynamic state using a filter having the structure shown in Fig. 1, and consisting of a bias-free-state estimator, a bias-estimator, and a bias-correction matrix V. Mathematically the optimum state estimate $\hat{x}$ is the sum of the bias-free-state estimate $\bar{x}$ and a correction term $V\hat{b}$, where $\hat{b}$ is the optimum estimate of the bias, i.e.,

$$\hat{x} = \bar{x} + V\hat{b} \tag{1}$$

The bias-free state estimate $\bar{x}$ is obtained by processing the observations in a Kalman filter designed under the assumption that the bias vector b is identically zero. In the standard implementation of the bias-free filter, the difference

$$\bar{r} = y - \bar{y} \tag{2}$$

between the actual observation and the estimate thereof is produced. This difference signal known as the "residual" or nowadays by the more popular term "innovation" is the input to a second filter which can be called the "bias-estimator" because its output is the optimum estimate $\hat{b}$ of the unknown bias vector b. The bias estimate $\hat{b}$, multiplied by the correction matrix V is finally added to $\bar{x}$, in accordance with Eq.(1) to yield the desired optimum state estimate.

We expected to find that the bias-separated filter implementation would require fewer numerical operations than the augmented-state implementation. But we found, to our disappointment, that the number of operations for each implementation were comparable. (The bias-separated implementation did open some new possibilities for approximations that may not have been evident in the augmented-state implementation.)

But the advantages of the bias-separated implementation for avoiding numerical ill-conditioning are

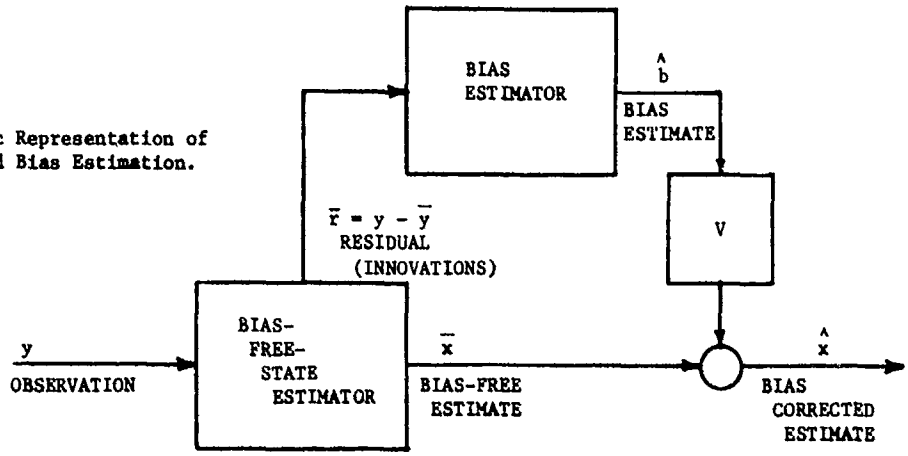


Figure 1. Schematic Representation of Separated Bias Estimation.

obvious. In the augmented-state implementation the overall process is of order $N + K$, where N is the number of dynamic state variables (i.e., the dimension of x) and K is the number of biases (the dimension of b) and the $N + K$ variables are all coupled, in the filter and in the covariance matrix propagation. In the bias-separated implementation the maximum dimension one needs to be concerned with the larger of N or K, and errors in the estimation of the bias do not contaminate the estimation of the bias-free estimate $\bar{x}$ of the dynamic state.

The strategy we employed in 1969 for deriving the bias-separated filter was motivated by a similar strategy we had then been using in our study of quasi-optimum control. Investigators unfamiliar with that technique found the results to be lacking in motivation. Several authors (Agee and Turner [5], Mendel and Washburn [6,7], and Bierman [8]) have contributed to correcting this deficiency by providing alternate derivations and interpretations.

The bias-free residuals (or innovations) occurred naturally in the derivation of [4] but their significance was not fully appreciated at the time: the interpretations resulting from the work of Kailath et al [9,10] had only just begun to emerge. In retrospect it is evident that the bias separation method introduced in [4] exemplifies one of the applications that can be made of the residuals of a Kalman filter. Failure detection and diagnosis [11] exemplifies another application that can be made of the residuals. We shall return to this application subsequently.

2. REVIEW OF THEORY

It has been remarked above that the bias-separated structure of Fig. 1 can be derived in a number of different ways. Since new methods of derivation can afford new insights, we offer here still another derivation based on the theory of linear observers. Since the latter theory does not depend on properties of stochastic processes, the derivation shows that the structural properties of the separated-bias estimation algorithm transcends the stochastic process underpinnings.

For simplicity we consider only a continuous-time process

$$\dot{x} = Ax + Bb + u \tag{3}$$

with observations given by

$$y = Cx + Db + v \tag{4}$$

where b is a constant (but unknown) vector (called the "bias") and u and v white noise processes having known spectral density matrices, Q and R respectively.

In accordance with well-known theory [12], an observer for the process Eqs.(3) - (4) is defined by

$$\dot{\hat{x}} = A\hat{x} + B\hat{b} + K_x(y - C\hat{x} - D\hat{b}) \tag{5}$$

$$\dot{\hat{b}} = K_b(y - C\hat{x} - D\hat{b}) \tag{6}$$

These relations are depicted in Fig. 2a. The gain matrix

$$K = \begin{bmatrix} K_x \\ K_b \end{bmatrix} \tag{7}$$

is chosen to make the observer asymptotically stable. If the gain matrix is chosen optimally, the observer Eqs.(5) - (6) is the Kalman filter; otherwise the observer has only the property that the error

$$e = \begin{bmatrix} x - \hat{x} \\ b - \hat{b} \end{bmatrix}$$

tends asymptotically to zero.

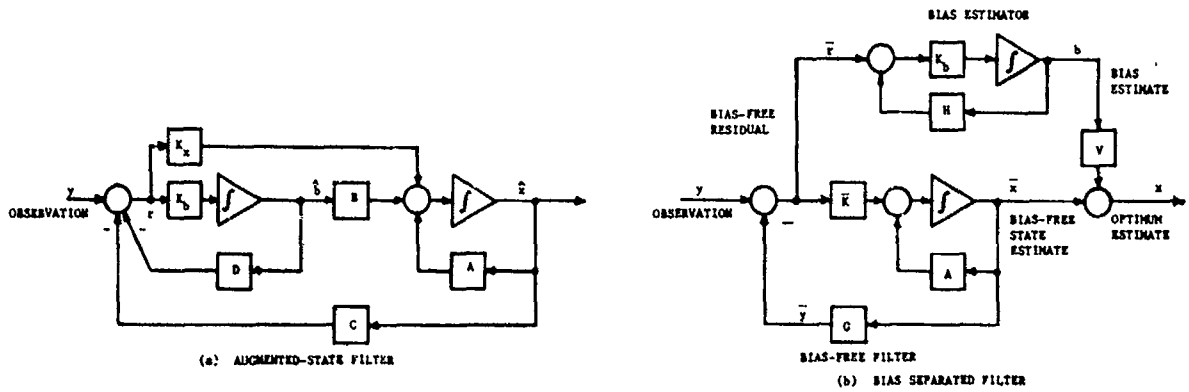


Figure 2. Two Forms of Observers for Linear Systems with Bias.

Now consider the possibility of expressing the observer Eq.(5) and Eq.(6) in the bias-separated form shown in Fig.2b, i.e.,

$$\hat{x} = \bar{x} + V\hat{b} \tag{8}$$

where $\bar{x}$ is the state of the bias-free observer, given by

$$\dot{\bar{x}} = A\bar{x} + K\bar{r} \tag{9}$$

and $\hat{b}$ is the bias estimate given by

$$\dot{\hat{b}} = K_b(\bar{r} - H\hat{b}) \tag{10}$$

where $\bar{r}$ is the bias-free residual, given by

$$\bar{r} = y - C\bar{x} \tag{11}$$

Note the following:

- The bias-free observer Eq.(9) retains the general form of an observer and is hence asymptotically stable if $\bar{K}$, as yet unspecified, is appropriately chosen.

- The bias estimator Eq.(10) is also in the form of an observer except that its input is the bias free residual $\bar{r}$ (not y). Moreover, the bias filter gain K_b is the same matrix that appears in Eq.(6). In principle, the results to be obtained could be generalized somewhat by permitting K_b in Eq.(6) and K_b in Eq.(10) to be different matrices. We will not do this, however

Our objective in the analysis that follows is to find relationships between the matrices V, $\bar{K}$ and H that must hold in order Eqs.(8) - (11) be equivalent to Eqs.(5) - (6). To this end, substitute Eq.(8) into Eq.(6) to obtain

$$\begin{aligned}\dot{\hat{b}} &= K_b[y - C(\bar{x} + V\hat{b}) - D\hat{b}] \\ &= K_b[\bar{r} - (CV + D)\hat{b}]\end{aligned}$$

which is of the form of Eq.(10) provided that

$$H = CV + D \tag{12}$$

This is one of the relations we are seeking.

Also, substitute Eq.(8) into Eq.(5) to obtain

$$\begin{aligned}\dot{\bar{x}} + V\dot{\hat{b}} + \dot{V}\hat{b} &= A(\bar{x} + V\hat{b}) + B\hat{b} + K_x(\bar{r} - H\hat{b}) \\ &= A\bar{x} + K_x\bar{r} + (B + AV - K_xH)\hat{b}\end{aligned} \tag{13}$$

But, from Eq.(9) and Eq.(10), the left-hand side of Eq.(13) is

$$\dot{\bar{x}} + V\dot{\hat{b}} + \dot{V}\hat{b} = A\bar{x} + \bar{K}\bar{r} + VK_b(\bar{r} - H\hat{b}) + \dot{V}\hat{b}$$

The $A\bar{x}$ term cancels on both sides of Eq.(13) leaving

$$(\bar{K} + VK_b)\bar{r} + (\dot{V} - VK_bH)\hat{b} = K_x\bar{r} + (B + AV - K_xH)\hat{b}$$

Thus Eq.(5) is satisfied for all $\bar{r}$ and $\hat{b}$ provided that

$$\bar{K} = K_x - VK_b \tag{14}$$

and

$$\begin{aligned}\dot{V} &= AV + VK_bH + B - K_xH \\ &= (A - \bar{K}C)V + B - \bar{K}D\end{aligned} \tag{15}$$

Thus the augmented-state observer of Fig.2a, with gain matrices K_x and K_b obtained by any method whatsoever, can be transformed into the bias-separated form of Fig.2b, provided the matrices H, V, and $\bar{K}$ satisfy the two algebraic equations (12) and (14) and the matrix differential equation (15), which becomes the matrix Riccati equation

$$\dot{V} = (A - K_x C)V + VK_b D + VK_b CV + B - K_x D \tag{16}$$

upon substitution of Eq.(14).

This derivation is strictly algebraic and does not require that any significance be attached to the matrices that appear in the respective relations, but only that a matrix V which satisfies Eq.(16) can be found. (The general conditions on A, B, C, D, K_x and K_b that guarantee the existence of a solution to Eq.(16) have, to our knowledge, not been explored.) In particular, there is no requirement that K_x and K_b be optimum for the noise u and v. And, irrespective, of the optimality of these gains, the steady state errors in the estimation of x and b will tend to zero if these gains result in a stable observer. An alternate demonstration of this property was given in [13].

If the observer gains K_x and K_b are optimum for the noises u and v in Eq.(3) and Eq.(4), however, then the bias-separated filter is also optimum. And it is then possible to provide interpretations of the matrices $\bar{K}$ and V. In particular, as shown in [4], $\bar{K}$ is the optimum gain for the bias-free filter, i.e., for estimating the state x when b is known to be identically zero, i.e.,

$$\bar{K} = \bar{P}C'R^{-1} \tag{17}$$

where

$$\dot{\bar{P}} = A\bar{P} + \bar{P}A' - \bar{P}C'R^{-1}C\bar{P} + Q \tag{18}$$

with Q and R being the spectral density matrices of u and v, respectively. Moreover, the matrix V can be interpreted as the ratio of the cross-covariance matrix of x and b to the covariance matrix of b. Specifically, if

$$\begin{aligned} P_{xb} &= E[(x - \hat{x})(b - \hat{b})'] \\ P_b &= E[(b - \hat{b})(b - \hat{b})'] \end{aligned} \tag{19}$$

then, as shown in [4],

$$V = P_{xb}P_b^{-1} \tag{20}$$

This helps to provide an intuitive interpretation of the bias correction equation (8), in which V is seen to be the gain matrix for correcting the bias. By Eq.(20) this matrix is proportional to the cross-correlation between the error in estimating the state and the error in estimating the bias. If the influence of the latter on the former is relatively weak, as measured by a small cross-correlation matrix P_{xb}, it is only reasonable to expect that the correction to the bias-free estimate $\bar{x}$, when knowledge of b is obtained, would likewise be small. Likewise, if the cross-correlation between the bias estimate and the state estimate is strong, we should expect a large bias correction. In addition, we would expect the magnitude of the bias correction to be inversely proportional to the uncertainty in the estimate, and this explains the presence of P_b^{-1} in Eq.(20).

Another useful relationship involving V, as given in [4], is

$$\hat{P} = \bar{P} + VP_bV' \tag{21}$$

where $\hat{P}$ is the covariance matrix of the estimate of x in the presence of bias, and $\bar{P}$ is the covariance in the absence of bias. Since VMV' is a postive-(semi) definite matrix, it is clear that P is larger than $\bar{P}$, which is of course to be expected. But Eq.(21) quantifies the difference between P and $\bar{P}$. In particular if VP_bV' is small relative to $\bar{P}$, then the increase in error due to b is correspondingly small and b is not significant in the estimation of x. Since it is possible to include all the bias variables except one, say b_i, in the state x and apply the result of Eq.(21) to b_i alone, this provides a way of assessing the effect of each component b_i of the bias vector b on the estimate of x. Those components which do not contribute significantly to the increase in P over $\bar{P}$ are candidates for omission in a suboptimal implementation.

A useful interpretation of the bias estimation equation (10) can be had by considering the problem of estimating an unknown constant b observed through noise, i.e.,

$$\dot{b} = 0 \tag{22}$$

with observation $\bar{r}$ given by

$$\bar{r} = Hb + \zeta \tag{23}$$

where ζ is white noise, having a spectral density matrix R. Direct application of basic Kalman filter theory shows that the optimum estimator is exactly in the form of Eq.(10) with the gain matrix K_b given by

$$K_b = P_bH'R^{-1} \tag{24}$$

with P_b being the solution of the variance equation

$$\dot{P}_b = -P_bH'R^{-1}HP_b \tag{25}$$

It is shown in [4] that these are precisely the relations that are satisfied by K_b and P_b. Hence the operation of the bias estimator Eq.(10) can be interpreted as the extraction of a constant observed in white noise, with the residual vector $\bar{r}$ being the observation. If the bias b is known to be zero, then by

Eq.(23) $\bar{r}$ is zero-mean white noise with the same spectral density as the original observation noise. This confirms a well-known property of the residuals in a bias-free filter. But Eq.(23) also explains the effect of a nonzero bias b in the original dynamic equations on the residual of a Kalman filter designed for zero bias. In particular, the residual $\bar{r}$ is not a zero mean process but rather has a mean given by Hb with the "equivalent observation" matrix H given by Eq.(12). As one might have expected the observation bias matrix D appears directly in H. But the dynamic bias matrix B appears in H only through its influence on V. Moreover, since D also appears in the equation for V it is not entirely accounted for by the D term in Eq.(12).

The interpretation of Eq.(10) as the extraction of a constant observed in white noise, was first advanced by Mendel and Washburn [6,7] (see also [13]). This interpretation is particularly useful in failure detection applications as discussed below.

For simplicity, the above review was given in terms of a continuous-time process. There are exactly analogous results for discrete-time systems which are presented in summary form in Appendix A.

3. EXTENSIONS OF THEORY

Alternate Derivations--As often happens with theoretical results, not everyone was pleased with the method used to derive the bias separated filter, and several investigators contributed alternate derivations which may provide added insight or suggest further extensions.

In 1971, Lin and Sage [14] reported on their approach to bias estimation using maximum likelihood methods and which entailed solution of a two-point boundary-value problem. They obtained results which were subsequently shown by Godbole [15] to be identical to the original results in [4]. As Godbole pointed out, this fact was hardly surprising, since it had been known for several years that the Kalman filter is a recursive implementation of the solution of the two-point boundary value problem.

In 1972, Agee and Turner [5] derived equations for the (discrete-time) bias-separated filter by starting with the correction equation of the form

$$\hat{x} = \bar{x} + V\hat{b}$$

and, by a method somewhat similar to the method used in Section 2, determined the conditions under which the decoupling is possible. One of their conclusions is that the partitioning is only possible when the bias is not a random process. In other words, although the bias need not be a constant, but rather may be given by

$$\dot{b} = Zb \tag{26}$$

it would not be permissible to include a noise input on the right-hand side of Eq.(26). Hence any attempt at extending the result to the case in which the bias is a random process must of necessity lead to a sub-optimum filter. It is noted, however, that the derivation in Section 2, is independent of the manner in which the gains K_x and K_b are obtained for the augmented-state filter. The augmented state filter (Fig.2a) and the corresponding gains K_x and K_b, generally can be found even if the bias b is a random process. Thus it would seem that the restriction that b not be a random process is somehow superfluous. The Agee and Turner result of [5] thus suggests either that the bias-free filter of Fig.2b that produces $\bar{x}$ is not the optimum filter for the process with $b \neq 0$, or that a solution to Eq.(16) for the correction matrix V cannot be found. It is interesting, but of no real importance, that $\bar{x}$ is the estimate of x in the absence of bias. If this were not the case, but if we could solve for V, $\bar{K}$ and H, then the bias-separated structure of Fig.2b could still be used.

Also in 1972, Tacker et al, in studying control of interconnected energy systems [16] (apparently independently) discovered the bias-separation result within the framework of linear optimum control theory.

In the early 1970's the square-root method of implementing the optimum recursive filter had been gaining in popularity as another method of overcoming problems of numerical conditioning. In 1975, Bierman, an active investigator in this field suggested [8] that "the [square-root information filter] SRIF is a natural method of dealing with biases," and developed a derivation using this methodology. In the course of this development, several additional results and interpretations emerged. He showed the relationship between the correction matrix V and the "estimation sensitivity" and "consider covariance" matrices of importance in orbit determination. He also pointed out how the bias-separation method can be used to compute smoothing solutions.

A very rigorous development of the results of [4] for both continuous-time and discrete time systems was published in 1978 by Mendel and Washburn [6,7] based on Washburn's 1977 doctoral dissertation. The development assumed the bias separated form of Eq.(8) and, like Agee and Turner, found the conditions under which Eq.(8) is valid. In the course of this development they demonstrated that the estimation of the bias using the residuals (innovations) of the bias-free filter is equivalent to extraction of a constant observed in white noise, and used this property to apply well-known algorithms in which the bias vector changes dimension with time.

Extension to Time-Varying Bias--The original development of the bias-separated algorithm, as given in [4], was confined to a constant bias, i.e., $\dot{b} = 0$, but it was remarked at that time that the extension to a time-varying bias would be fairly simple. The explicit extension was first presented by Tacker and Lee [17] in 1972. Bierman subsequently pointed out [18] that the results of Tacker and Lee could be obtained more directly by noting that if $\dot{b} = Zb$, then $b(t) = \Phi(t,0)b_0$ where $\Phi(t,0)$ is the state transition matrix corresponding to Z and hence the time-varying bias estimation problem can be replaced by the problem of estimating the initial state b_0 of $b(t)$.

Extension to Nonlinear Systems--Few dynamic systems of practical interest are linear; Kalman-Bucy filtering is often used to estimate the state of such nonlinear systems, nevertheless. The standard technique used for nonlinear systems is the "extended Kalman filter," (EKF) in which the actual nonlinear equations are

used in computing the residuals and for the dynamic model, but in which the correction due to the residual is linear. Specifically, for a continuous-time process

$$\dot{z} = f(z) + u \tag{27}$$

with observations given by

$$y = g(z) + v \tag{28}$$

in which u and v are white noise processes, the EKF equations are

$$\dot{\hat{z}} = f(\hat{z}) + Kr \tag{29}$$

with

$$r = y - g(\hat{z}) \tag{30}$$

The gain matrix K is computed from the covariance matrix $\hat{P}$, i.e.,

$$K = \hat{P}G'R^{-1} \tag{31}$$

with

$$\dot{\hat{P}} = F\hat{P} + \hat{P}F' - \hat{P}G'R^{-1}G\hat{P} + Q \tag{32}$$

in which F and G are Jacobian matrices of f and g, respectively, evaluated along the estimated trajectory, i.e.,

$$F = \left[\frac{\partial f}{\partial z}\right]_{z=\hat{z}} \qquad G = \left[\frac{\partial g}{\partial z}\right]_{z=\hat{z}} \tag{33}$$

The covariance matrix $\hat{P}$ is computed, along with $\hat{z}$, as part of the EKF algorithm.

In many practical applications the EKF algorithm outlined above works quite well. In these applications it would be worthwhile to develop a bias-separated form of the EKF for the case in which the state z includes dynamic variables and biases, i.e., in systems in which the state vector z can be partitioned into a dynamic state x and a bias b:

$$z = \left[\begin{array}{c} x \\ \hline b \end{array}\right]$$

and hence

$$f(z) = f(x,b) \qquad , \qquad g(z) = g(x,b)$$

The direct extension of the separated-bias form to the EKF is not as obvious as it might appear at first glance, owing to the nonlinear nature of $f(\ )$ and $g(\ )$. In particular, the Jacobian matrices that enter into the covariance matrix P of Eq.(32) are evaluated at the optimum estimate of x and b, so that the representation of the variance equation in the manner of Eq.(21) which leads to the separated-bias form may not be valid, and this leads to some difficulty in determining the gains $\bar{K}$ and K_b to be used in the bias-separated form. It may be argued, with Agee and Turner [5], that the EKF is not optimum anyway, and hence any reasonable choice of gains might be acceptable. Adopting this viewpoint, however, does not end the matter. In the bias separated structure, for example, the bias free residual is

$$\bar{r} = y - C\bar{x}$$

The counterpart of this in the nonlinear equation is

$$\bar{r} = y - g(\bar{x},0)$$

In other words the state used to compute the expected observation is $\bar{x}$ not $\hat{x}$. This is counter to the spirit of the EKF in which the linearization is always made about the complete state estimate. If the effect of the bias on the estimated state is small, it may not matter too much whether the linearization is about $\hat{x}$ or about $\bar{x}$. When the bias is significant, however, the difference between $\hat{x}$ and $\bar{x}$ may be enough to affect the results significantly.

The generalization of the bias-separation algorithm to nonlinear dynamics was considered by Sinha and Mahalanabis [19] in 1972. They applied the results of [4] to partitioning the dynamic state and bias estimation equations. They considered both the discrete-time and the continuous-time case, but did not elucidate the problem of where to evaluate the requisite partial derivatives.

The special case in which the bias b enters linearly into the dynamics and observations was studied carefully by Mendel [20] in 1976. Considering only the discrete-time problem, he showed that the separated-bias algorithm fits into the EKF algorithm, except that the matrix V in the correction equation must be recomputed after the bias-free state update in order to implement the EKF algorithm exactly. He does not address the issue of whether the additional computer time needed to compute V twice per time-update step is justifiable in view of the likelihood that the EKF is not optimal.

A method of possibly overcoming any problems that might arise because of nonlinear effects on the difference between $\hat{x}$ and $\bar{x}$ is to reset the bias computation from time to time, i.e., by setting $\bar{x}$ to $\hat{x}$ and, simultaneously b to zero. This operation merely resets the means. It would be improper to reset the covariance matrices, since the uncertainty in the bias is not changed by resetting the mean. The resetting operation can be performed whenever a suitable (ad hoc) test reveals that the difference between $\hat{x}$ and $\bar{x}$ has a significant nonlinear effect.

There are certain situations in which the EKF can be improved upon by using second-order terms in the

filter. In the case of a continuous-time process with discrete-time observations, the state estimate between observations is determined by

$$\dot{\hat{z}} = f(\hat{z}) + \frac{1}{2}\left(\frac{\partial f}{\partial \hat{z}}\right)' P \tag{34}$$

where $$\left(\frac{\partial f}{\partial \hat{z}}\right)' P = \left[\text{trace}\left(\frac{\partial f_1}{\partial \hat{z}}\right)' P \; . \; . \; . \; \text{trace}\left(\frac{\partial f_n}{\partial \hat{z}}\right)' P\right]'$$

where P is the covariance matrix. The state estimate is updated at instants t_i of observation using

$$\hat{z}(t_i) = \tilde{z}(t_i) + K[y(t_i) - g(\tilde{z}(t_i))] \tag{35}$$

where $\tilde{z}(t_i)$ is the optimum estimate of the state just prior to the observation, obtained by integrating Eq.(34) over the interval $[t_i, t_{i-1}]$ starting with $\hat{z}(t_{i-1})$. The presence of the term $1/2(\partial f/\partial \hat{z})'P$ in Eq.(34) distinguishes the second-order filter from the EKF.

In 1974, Shreve and Hedrick [21] showed that the covariance matrix can be propagated in the separated form, as given in Appendix A, but that the fundamental state separation equation, Eq.(1), does not generally apply unless the observation equations are linear.

Extension to Noise on Bias--By the manner in which the basic theory was developed in [4] it was apparent that it would be difficult to extend the bias separation result to the case in which b is a random process, i.e.,

$$\dot{b} = \xi$$

where ξ is white noise. The difficulty was confirmed by the aforementioned 1972 analysis of Agee and Turner [5].

Since an absolutely constant bias is a mathematical idealization--no physical bias is perfectly constant for all time--an extension of the bias-separation algorithm, even as an approximation, would be highly desirable. The need for such an extension was recognized by Tanaka [22] who, in 1975, developed an algorithm for discrete-time systems which retains some of the features of the original algorithm but does not completely decouple the bias calculations from the calculation of the dynamic state. The possibility of dropping terms in Tanaka's algorithm when the noise on the bias is small might merit attention, but does not seem to have been explored.

More recently, Washburn and Mendel [23] have generalized the results of [4] in several directions. They considered the general process

$$\begin{aligned} \dot{x} &= Ax + \epsilon Bz + u \\ \dot{z} &= Cz + \epsilon Dx + v \\ y &= Hx + \epsilon Nz + w \end{aligned} \tag{34}$$

with

which, as $\epsilon \to 0$ reduces to two uncoupled systems in which the substate z is not observed. They treated this problem by assuming the desired optimum estimate $\hat{x}$ to be of the form

$$\hat{x} = G\tilde{x} + V\tilde{z} + \hat{\zeta} \tag{35}$$

This is a generalization of the basic separated estimation formula, Eq.(1), owing to the appearance of the matrix G and the correction term $\hat{\zeta}$. For the general case of Eq.(34), the generalized separated estimation formula has scarcely any advantage over the augmented-state form that would arise directly from Eq.(34). When ϵ is small, however, they have shown, using perturbation methods, that a suboptimal estimator of the form of Eq.(1) (i.e., with $G=I$ and $\hat{\zeta}=0$) can be found. These results were illustrated in [23] by a simple example which, however, is not very convincing; perhaps it does not show the results to their best advantage. Another shortcoming of the Washburn-Mendel analysis is that too many terms are deleted when $\epsilon \to 0$. There is no need to include ϵ in front of B and N in Eq.(34) since existing theory already permits the treatment of these terms without the need for approximation. Nevertheless, this analysis suggests the possibility of using perturbation methods as a general method of extending the results of the separated-bias estimation algorithm.

4. APPLICATION TO FAILURE DETECTION AND ESTIMATION

The difference between the actual observation and its optimum estimate, known as the residual r or innovation process, has known statistical properties: namely

(i) $\quad E[r] = 0 \qquad$ (zero mean)

(ii) $\quad E[r(\tau)r'(\tau)] = W\delta(t-\tau)$ (white noise of spectral density W)

Deviations of the empirical statistics of the process as obtained from operating data, may serve as an indicator that the actual process is not the same as the process for which the optimum filter was designed. If such deviations start small but suddenly become large, this could be evidence that a change (i.e., a failure) has taken place in the system. The general use of residuals for fault detection and isolation was suggested in 1971 by Mehra and Peschon [11]. A number of investigators subsequently took this suggestion and developed techniques for estimating changes in bias, and hence for detecting and correcting of system failures that may be attributed to such changes.

In the context of failure detection, it is useful to draw a distinction between determining whether or not a bias is present and estimating its size (i.e., magnitude and direction) given (or assuming) that it is present. Since an estimated bias b of zero is equivalent to no bias, it is reasonable to believe that the separated-bias estimation algorithm can be used to advantage for failure detection and estimation. In 1977 Bellingham and Lees have reported using essentially this procedure for detection of malfunctions in chemical process control systems. The basic technique and some simulation results are given in [24] and and some experimental results are presented in [25].

A significant limitation of the separated bias algorithm, as developed in [4], is that it rests on the assumption that the bias is constant from the start of estimation forever after. The consequence of this assumption is that the bias estimator gain matrix tends to zero (as $1/t$) and hence the capability of the estimator to track a bias change that occurs after the estimator is turned on diminishes with time. Since the theory was developed on the assumption that the bias does not change there is no reason to expect that the bias estimator would have such a capability, but it does impose a practical limitation upon using the basic algorithm without modification. One obvious ad-hoc modification would be simply to prevent the bias estimator gain from going to zero, by assuming, for example, that the uncertainty in the bias cannot be reduced beyond an arbitrarily specified level. This assumption prevents the bias estimator gain from being reduced to zero. As has been shown in [13] and [26] constant bias is estimated with zero steady state error. This means that even if the bias changes after the estimator is turned on, if it remains constant after the change, it will be estimated without error. In fact, if the bias is piecewise-constant, but the intervals between transitions are widely spaced in time, the bias filter will track the piecewise-constant bias moderately well.

But preventing the bias estimator gain from going to zero is an ad-hoc remedy and it ought to be possible to achieve better performance by using more sophisticated statistical procedures.

One procedure for failure detection which has recently been receiving a great deal of attention is the "generalized likelihood ratio (GLR) technique" developed over the past two decades on the basis of the pioneering work of Abraham Wald on sequential estimation. The direct application of the GLR technique for estimating jumps (including, but in principle not limited to, changes in bias) in dynamic systems was proposed by Willsky and Jones [27] in 1976. They showed that the residual in case of bias change can be expressed as

$$r = Gv + w \tag{36}$$

where v is the bias change to be estimated, w is zero mean white noise, and G is a matrix calculated in the GLR algorithm. And, reasoning from (36) they established a correction equation of the form

$$\hat{x} = \bar{x} + (\Phi - F)\hat{v} \tag{37}$$

where $\bar{x}$ is the estimate of x under the hypothesis of no failure

$\hat{v}$ is the GLR estimate of v

Φ , F are matrices defined by the GLR algorithm.

The GLR estimate $\hat{v}$ is obtained by processing the sequence of residuals, as explained in [27].

These results apparently were obtained independent of the prior results on bias estimation. It became clear subsequently, however, that the matrix G in Eq.(36) is equivalent to H in Eq.(23) above and $\Phi - F$ is equivalent to V in Eq.(1). Hence a close relationship between the GLR method and the earlier bias estimation method can be seen: both methods use the residuals of the bias-free filter to obtain an estimate of the bias (b or v) and both use the resulting estimate to correct the bias-free estimate. The procedures used to obtain the estimate b or v are different, of course. Owing to the assumption that b is a constant for all time, the estimation equation for b is linear, and fairly simple, whereas the GLR algorithm for obtaining v is nonlinear and quite complex. In circles in which the GLR algorithm is popular, it is generally felt that the added complexity of the algorithm gives it advantages in terms of superior performance.

Later in 1976, Chien and Adams [28] published another failure detection method which also uses the residuals of the bias-free estimator. This technique employs the sequential probability ratio test (SPRT), which also derives from the original work of Wald but differs in some of its details from the GLR method of Willsky and Jones. Chien and Adams observed several deficiencies of the SPRT, namely that the latter did not account for the likelihood of no failure when the estimator is first turned on, and consequently that a failure, if it were to occur, would occur later. They proposed an ad hoc correction to the SPRT to overcome these deficiencies, and demonstrated the applicability of the method to inertial navigation system.

In 1979 Chang and Dunn [29] returned to the GLR approach of Willsky and Jones. They presented a recursive algorithm for implementing the GLR calculations and showed that this algorithm can be interpreted as being those of a Kalman filter for estimating a constant bias, starting at the time of occurrence of the jump. The relationship between the bias separation method and the GLR approach was thus made quite explicit. (This relationship was recently considered further by Caglayan [30]).

Another approach to the problem of detecting and estimating the occurrence and magnitudes of transitions in a piecewise constant signal was introduced by Friedland in 1979 [31,32]. The underlying idea is to model changes in the bias as the result of a highly nongaussian noise input, i.e.,

$$b_{n+1} = b_n + v_n$$

where v_n has a probability density function that includes a delta function at the origin to account for the finite probability that no transition in b takes place at any given instant. Using available theory for maximum likelihood estimation with nongaussian noise, Friedland developed an approximate recursive algorithm for obtaining a maximum likelihood estimate of b_n. Then, assuming that the interpretation of Eq.(23) (i.e., that bias estimation is equivalent to extraction of a constant buried in white noise) remains

valid when the bias is not constant but only piecewise constant, Friedland and Grabousky [33] developed a recursive algorithm for detection of failures in dynamic systems. This algorithm seems computationally comparable to the Chang-Dunn recursive implementation of the GLR algorithm, but a detailed comparison of the two, which would reveal any significant differences, remains to be performed.

5. ADDITIONAL APPLICATIONS

Trajectory Estimation--Our work on separated bias estimation was motivated by our observation that there are many problems in which there are relatively few dynamic state variables but a large number of parameters to be estimated and that separation of the parameter estimation from the estimation of the dynamic state variables would be computationally advantageous. This was borne out by the experience of Agee and Turner [5] who used the method in the program they developed for their BET (Best Estimate of Trajectory) computer program. They found, for example, that although there might be only nine dynamic state variables, there might be as many as sixty-six constant parameters in an accurate model of the sensors used on the White Sands Missile Range where the program would be employed. They examined several other techniques but discarded them because of numerical difficulties or because they did not produce satisfactory estimates of the bias. They reported using the algorithm, with the modifications discussed earlier, with both simulated and actual trajectory data.

We used the bias separation technique in our own work on orbit and satellite mass determination [34] which was reported in 1970. Although there were not a large number of parameters to be estimated, the method was beneficial because our program was to be developed by modifying an existing program in which no provision was made for estimating these parameters, and which was written in a manner that did not readily lend itself to increasing the dimension of the state vector to accommodate the additional state variables. But it was fairly easy to add the bias estimation capability to the existing program.

Aided-Inertial Navigation--When the Kalman filtering technique is applied to mixing inertial data with other navigation data, it is found that many of the variables to be estimated are actually biases. These include gyro and accelerometer bias ("drift rates") scale factor uncertainties, misalignment angles, and uncertain parameters in the navigation aid such as doppler scale factor and boresight errors. The bias separation technique described herein would be appropriate for this application. Its use in this context is suggested by Farrell [35] and alluded to by Nash et al [36] in connection with testing of inertial navigation systems and components.

Calibration--Also in connection with inertial navigation, as well as in other applications, it is necessary to determine the bias vector b in a system defined by Eqs.(3) and (4). An optimum estimate of the state x is often not required in such cases. It is clear that the structure of Fig. 1 can be used to obtain an optimum estimate $\hat{b}$, and the correction term Vb can be omitted if x is not required. This application was described by Friedland [37] in 1977 in which the method was illustrated by an example of the calibration (i.e., determination of drift rates and g-dependent error coefficients) of a two-axis gyro.

A by-product of the analysis of [37] was an alternate representation of the bias estimation equation, Eq.(10). It was shown that the bias b can be expressed as the product of the bias covariance matrix P_b as obtained in Eq.(25) with a vector q which is obtained by integrating the weighted residuals. In mathematical terms, it was shown that

$$\hat{b} = P_b q \tag{38}$$

where

$$\dot{q} = H' R^{-1} r \tag{39}$$

This form of the bias estimation equation is readily verified by substituting the expression for $\hat{b}$ and its derivative into Eq.(10), taking into account that P_b is given by Eq.(25).

This form of the calibration equation is advantageous when an estimate of b is needed only at the end of a fixed calibration interval. In this case it is necessary only to determine the vector q, by numerical integration of the weighted residual $H'R^{-1}r$. Then at the end of the predetermined calibration time T_c the estimate is obtained by multiplying $q(T_c)$ by the covariance matrix $P_b(T_c)$. It is thus not necessary to store P_b for other instants of time. This results in a considerable reduction in computer storage requirements over what might be required to compute $\hat{b}$ using Eq.(10).

From Eq.(25) it is readily determined that

$$P_b(t) = [P_b^{-1}(0) + L(t)]^{-1}$$

where

$$L(t) = \int_0^t H'(\tau) R^{-1} H(\tau) d\tau$$

In the absence of a priori information on b $P_b^{-1}(0) = 0$ and $P_b(t) = L^{-1}(t)$. Hence L must have an inverse for some value of t in order that b can be determined. In most cases it is necessary to introduce motion into the dynamic system, i.e., to make $C(t)$ and $D(t)$ time-varying so as to produce a matrix $H(t)$, defined by Eq.(12) which results in a nonsingular L matrix for some value of time. It is argued in [37] that an optimal choice of H would be such that $L(t)$ is a diagonal matrix which is a generalized orthogonality requirement on the elements of the matrix H.

Closed-Loop Control--It is often necessary to deal with biases in closed-loop systems. Fortunately the bias-separation technique is applicable to closed loop systems as well as open-loop systems. This was demonstrated by Tacker, Lee, et al [16,17] in 1972, in connection with their studies of interconnected electric power systems. Another use of the separated-bias estimation technique in connection with decentralized control of power systems was reported by Venkatesvarlu and Mahalanabis [38] in 1977. The use of this technique by Bellingham and Lees [24,25] in connection with closed-loop control of chemical processes was discussed above.

CONCLUSIONS

The pervasiveness of Kalman-Bucy filtering theory in so many aspects of modern control and estimation has motivated a great deal of research directed toward making the theory easier to use. The bias separation method reviewed in this paper is a result of this type of research. One would hope that this, and others in the same spirit, enhance the practical utility of the basic theory.

The bias-separation technique, originally presented in 1969, and subsequently extended in various directions, has a variety of practical applications some of which were described here. But a number of issues regarding this technique remain and merit further investigation. We have already remarked that the question of how the theory may be extended to cover the case in which the bias is not deterministic is not completely settled; the negative result of Agee and Turner counters the formal separation property used in the derivation presented in Section 2 of this paper. Another issue that is not fully resolved is the proper extension to nonlinear systems.

Another area of investigation that might merit further attention is the relationship between the bias separation method discussed in this paper to other methods of simplifying the Kalman filter calculations. Some work in this direction has already been done. In 1974, Samant and Sorenson [39] compared the bias separation method of this paper with an order-reduction method in which only a portion of the state-vector is optimally estimated. Both the Samant and Sorenson algorithm and the bias separation algorithm are optimum under the same set of assumptions and hence ought to give the same results. But there are differences in computational efficiency as measured in storage and number of operations. Samant and Sorenson conclude that their algorithm requires a larger number of operations, and is thus less efficient in terms of speed, but that it may be more efficient in terms of storage. In 1977, Chang and Dunn [40] studied the errors caused by omitting some state variables from the model used in the design of the estimator. If the states omitted are biases then, by virtue of Eq.(1) the error is given by $e = \hat{x} - x = V\hat{b}$. Since V and the statistics of $\hat{b}$ are known, the statistical properties of the error due to omission of b can readily be determined. Chang and Dunn, however, consider a more general case than is represented by our model as given by Eq.(3). Additional studies of these approximation methods might be worthwhile.

Another investigation that might be pursued is to determine the implications of the dual of the bias-separation algorithm in regard to deterministic optimum control. The mathematical duality between deterministic (linear, quadratic) optimum control and Kalman-Bucy filtering is well known. Hence the bias-separation method ought to have a dual in optimum control and this dual might have interesting properties with practical application.

With regard to applications, we see no reason why any problem in which biases are treated by augmenting the state vector cannot be treated by the method of this paper. The algorithm is entirely straightforward. In preparing this paper, we have endeavored to review all the applications that have been described in archival journals, but there was no feasible method of covering applications described in other literature such as technical reports issued under government contracts, etc., except those that were brought to our attention, such as the report of Agee and Turner [5], or a recent report by W. E. Hall, et al [41] describing an application to rotorcraft parameter identification. We would be grateful to receive descriptions of additional applications that may be known to readers of this paper.

REFERENCES

[1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," J. Basic Engr. Trans. ASME, Series D, Vol. 82, March 1960, pp. 35-45.

[2] Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," J. Basic Engr. Trans. ASME, Series D, Vol. 83, March 1961, pp. 95-108.

[3] Richman, J. and Friedland, B., "Design of Optimum Mixer-Filter for Aircraft Navigation Systems," Proc. National Aerospace Electronics Conf., Dayton, OH, May 1967, pp. 429-438.

[4] Friedland, B., "Treatment of Bias in Recursive Filtering," IEEE Trans. Automatic Control, Vol. AC-14, No. 4, August 1969, pp. 359-367.

[5] Agee, W. S. and Turner, R. H., "Optimal Estimation of Measurement Bias," Technical Report No. 41 (AD 753961), Mathematical Services Branch, Analysis and Computation Division, National Range Operations Directorate, White Sands Missile Range, NM, December 1972.

[6] Mendel, J. M. and Washburn, H. D., "Multistage Estimate of Bias States," Proc. 1976 IEEE Conf. Decision and Control, Clearwater, FL, December 1976, pp. 629-630.

[7] Mendel, J. M. and Washburn, H. D., "Multistage Estimation of Bias States in Linear Systems," Int. J. Control, Vol. 28, No. 4, 1978, pp. 511-524.

[8] Bierman, G. J., "The Treatment of Bias in the Square-Root Information Filter/Smoother," J. Optimization Theory and Applications, Vol. 16, Nos. 1/2, 1975, pp. 165-178.

[9] Kailath, T., "An Innovations Approach to Least-Squares Estimation--I: Linear Filtering in Additive White Noise," IEEE Trans. Automatic Control, Vol. AC-13, No. 6, December 1968, pp. 646-655.

[10] Kailath, T. and Frost, P., "Innovations Approach to Least Squares Estimation--II: Linear Smoothing in Additive White Noise," IEEE Trans. Automatic Control, Vol. AC-13, No. 6, December 1968, pp. 655-660.

[11] Mehra, R. K. and Peschon, J., "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems," Automatica, Vol. 7, 1971, pp. 637-640.

valid when the bias is not constant but only piecewise constant, Friedland and Grabousky [33] developed a recursive algorithm for detection of failures in dynamic systems. This algorithm seems computationally comparable to the Chang-Dunn recursive implementation of the GLR algorithm, but a detailed comparison of the two, which would reveal any significant differences, remains to be performed.

5. ADDITIONAL APPLICATIONS

Trajectory Estimation--Our work on separated bias estimation was motivated by our observation that there are many problems in which there are relatively few dynamic state variables but a large number of parameters to be estimated and that separation of the parameter estimation from the estimation of the dynamic state variables would be computationally advantageous. This was borne out by the experience of Agee and Turner [5] who used the method in the program they developed for their BET (Best Estimate of Trajectory) computer program. They found, for example, that although there might be only nine dynamic state variables, there might be as many as sixty-six constant parameters in an accurate model of the sensors used on the White Sands Missile Range where the program would be employed. They examined several other techniques but discarded them because of numerical difficulties or because they did not produce satisfactory estimates of the bias. They reported using the algorithm, with the modifications discussed earlier, with both simulated and actual trajectory data.

We used the bias separation technique in our own work on orbit and satellite mass determination [34] which was reported in 1970. Although there were not a large number of parameters to be estimated, the method was beneficial because our program was to be developed by modifying an existing program in which no provision was made for estimating these parameters, and which was written in a manner that did not readily lend itself to increasing the dimension of the state vector to accommodate the additional state variables. But it was fairly easy to add the bias estimation capability to the existing program.

Aided-Inertial Navigation--When the Kalman filtering technique is applied to mixing inertial data with other navigation data, it is found that many of the variables to be estimated are actually biases. These include gyro and accelerometer bias ("drift rates") scale factor uncertainties, misalignment angles, and uncertain parameters in the navigation aid such as doppler scale factor and boresight errors. The bias separation technique described herein would be appropriate for this application. Its use in this context is suggested by Farrell [35] and alluded to by Nash et al [36] in connection with testing of inertial navigation systems and components.

Calibration--Also in connection with inertial navigation, as well as in other applications, it is necessary to determine the bias vector b in a system defined by Eqs.(3) and (4). An optimum estimate of the state x is often not required in such cases. It is clear that the structure of Fig. 1 can be used to obtain an optimum estimate $\hat{b}$, and the correction term Vb can be omitted if $\hat{x}$ is not required. This application was described by Friedland [37] in 1977 in which the method was illustrated by an example of the calibration (i.e., determination of drift rates and g-dependent error coefficients) of a two-axis gyro.

A by-product of the analysis of [37] was an alternate representation of the bias estimation equation, Eq.(10). It was shown that the bias $\hat{b}$ can be expressed as the product of the bias covariance matrix P_b as obtained in Eq.(25) with a vector q which is obtained by integrating the weighted residuals. In mathematical terms, it was shown that

$$\hat{b} = P_b q \tag{38}$$

where

$$\dot{q} = H' R^{-1} r \tag{39}$$

This form of the bias estimation equation is readily verified by substituting the expression for $\hat{b}$ and its derivative into Eq.(10), taking into account that P_b is given by Eq.(25).

This form of the calibration equation is advantageous when an estimate of b is needed only at the end of a fixed calibration interval. In this case it is necessary only to determine the vector q, by numerical integration of the weighted residual $H'R^{-1}r$. Then at the end of the predetermined calibration time T_c the estimate is obtained by multiplying $q(T_c)$ by the covariance matrix $P_b(T_c)$. It is thus not necessary to store P_b for other instants of time. This results in a considerable reduction in computer storage requirements over what might be required to compute $\hat{b}$ using Eq.(10).

From Eq.(25) it is readily determined that

$$P_b(t) \left[P_b^{-1}(0) + L(t)\right]^{-1}$$

where

$$L(t) = \int_0^t H'(\tau) R^{-1} H(\tau) d\tau$$

In the absence of a priori information on b $P_b^{-1}(0) = 0$ and $P_b(t) = L^{-1}(t)$. Hence L must have an inverse for some value of t in order that b can be determined. In most cases it is necessary to introduce motion into the dynamic system, i.e., to make $C(t)$ and $D(t)$ time-varying so as to produce a matrix $H(t)$, defined by Eq.(12) which results in a nonsingular L matrix for some value of time. It is argued in [37] that an optimal choice of H would be such that $L(t)$ is a diagonal matrix which is a generalized orthogonality requirement on the elements of the matrix H.

Closed-Loop Control--It is often necessary to deal with biases in closed-loop systems. Fortunately the bias-separation technique is applicable to closed loop systems as well as open-loop systems. This was demonstrated by Tacker, Lee, et al [16,17] in 1972, in connection with their studies of interconnected electric power systems. Another use of the separated-bias estimation technique in connection with decentralized control of power systems was reported by Venkatesvarlu and Mahalanabis [38] in 1977. The use of this technique by Bellingham and Lees [24,25] in connection with closed-loop control of chemical processes was discussed above.

[12] Luenberger, D.G., "An Introduction to Observers," IEEE Trans. Automatic Control, Vol. AC-16, No. 6, December 1971, pp. 596-602.

[13] Friedland, B., "Notes on Separate-Bias Estimation," IEEE Trans. Automatic Control, Vol. AC-23, No. 4, August 1978, pp. 735-738.

[14] Lin, J. L. and Sage, A. P., "Algorithms for Discrete Sequential Maximum Likelihood Bias Estimation and Associated Error Analysis," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-1, No. 4, October 1971, pp. 314-324.

[15] Godbole, S. S., "Comparison of Friedland's and Lin-Sage's Bias Estimation Algorithms," IEEE Trans. Automatic Control, Vol. AC-19, No. 2, April 1974, pp. 143-145.

[16] Tacker, E. C.; Lee, C. C.; Reddoch, T. W., Tan, T. O.; and Julich, P. M., "Optimal Control of Interconnected Electric Energy Systems--A New Formulation," Proc. IEEE, Vol. 60, October 1972, pp. 1239-1241.

[17] Tacker, E. C. and Lee, C. C., "Linear Filtering in the Presence of Time-Varying Bias," IEEE Trans. Automatic Control, Vol. AC-17, No. 6, December 1972, pp. 828-829.

[18] Bierman, G. J., "Comments on 'Linear Filtering in the Presence of Time-Varying Bias'," IEEE Trans. Automatic Control, Vol. AC-18, No. 4, August 1973, p. 412.

[19] Sinha, A. K. and Mahalanabis, A. K., "Modeling Error Compensation in Nonlinear Estimation Problems," IEEE Trans. Systems, Man, and Cybernetics, Vol. AC-18, No. 6, November 1973, pp. 632-636.

[20] Mendel, J. M., "Extension of Friedland's Bias Filtering Technique to a Class of Nonlinear Systems," IEEE Trans. Automatic Control, Vol. AC-21, No. 2, April 1976, pp. 296-298.

[21] Shreve, R. L. and Hedrick, W. R., "Separating Bias and State Estimates in a Recursive Second-Order Filter," (Tech. Corresp.) IEEE Trans. Automatic Control, Vol. AC-19, No. 5, October 1974, pp. 585-586.

[22] Tanaka, A., "Parallel Computation in Linear Discrete Filtering," IEEE Trans. Automatic Control, Vol. AC-20, No. 4, August 1975, pp. 573-575.

[23] Washburn, H. D. and Mendel, J. M., "Multistage Estimation of Dynamical and Weakly Coupled States in Continuous-Time Linear Systems," IEEE Trans. Automatic Control, Vol. AC-25, No. 1, February 1980, pp. 71-76.

[24] Bellingham, B. and Lees, F. P., "Practical State and Bias Estimation of Process Systems with Initial Information Uncertainty," Int. J. Systems Sci., Vol. 8, No. 7, 1977, pp. 813-840.

[25] Bellingham, B. and Lees, F. P., "The Detection of Malfunction Using a Process Control Computer: A Kalman Filtering Technique for General Control Loops," Trans. I Chem E., Vol. 55, 1977, pp. 253-265.

[26] Friedland, B., "Recursive Filtering in the Presence of Biases with Irreducible Uncertainty," IEEE Trans. Automatic Control, Vol. AC-21, No. 5, October 1976, pp. 789-790.

[27] Willsky, A. S. and Jones, H. L., "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems," IEEE Trans. Automatic Control, Vol. AC-21, No. 1, February 1976, pp. 108-112.

[28] Chien, T-T, and Adams, M. B., "A Sequential Failure Detection Technique and its Application," IEEE Trans. Automatic Control, Vol. AC-21, No. 5, October 1976, pp. 750-757.

[29] Chang, C. B. and Dunn, K. P., "On GLR Detection and Estimation of Unexpected Inputs in Linear Discrete Systems," IEEE Trans. Automatic Control, Vol. AC-24, No. 3, June 1979, pp. 499-501.

[30] Caglayan, A. K., "Simultaneous Failure Detection and Estimation in Linear Systems," Proc. 19th IEEE Conf. Decision and Control, Vol. 2, Albuquerque, NM, December 1980, pp. 1038-1041.

[31] Friedland, B., "Maximum-Likelihood Estimation of a Process with Random Transitions (Failures)," IEEE Trans. Automatic Control, Vol. AC-24, No. 6, December 1979, pp. 932-937.

[32] Friedland, B., "Multidimensional Maximum Likelihood Failure Detection and Estimation," IEEE Trans. Automatic Control, Vol. AC-26, No. 2, April 1981, pp. 567-570.

[33] Friedland, B. and Grabousky, S. M., "Estimating Sudden Changes of Biases in Linear Dynamic Systems," IEEE Trans. Automatic Control, Vol. AC-26, No. 6, December 1981.

[34] Friedland, B.; Hutton, M.; and Richman, J., "Satellite Mass Determination Using Live Data," RADC-TR-69-375, Rome Air Development Center, Griffiss Air Force Base, NY, 13440, January 1970.

[35] Farrell, J. L., Integrated Aircraft Navigation, NY: Academic Press, 1976, p. 183.

[36] Nash, R. A.,Jr.; Kasper, J. F., Jr.; Crawford, B. S.; and Levine, S. A., "Application of Optimal Smoothing to Testing and Evaluation of Inertial Navigation Systems and Components," IEEE Trans. Automatic Control, Vol. AC-16, No. 6, December 1971, pp. 806-816.

[37] Friedland, B., "On the Calibration Problem," IEEE Trans. Automatic Control, Vol. AC-22, No. 6, December 1977, pp. 889-905.

[38] Venkateswarlu, B. E. and Mahalanabis, A. K., "Design of Decentralised Load-Frequency Regulators," Proc. IEE, Vol. 124, No. 9, September 1977, pp. 817-820.

[39] Samant, V. S. and Sorenson, H. W., "On Reducing Computational Burden in the Kalman Filter," Automatica, Vol. 10, 1974, pp. 61-68.

[40] Chang, C.B. and Dunn, K-P, "Kalman Filter Compensation for a Special Class of Systems," IEEE Trans. Aerospace and Electronic Systems, Vol. AES-13, No. 6, November 1977, pp. 700-706.

[41] Hall, W. E.; Bohn, J.; and Vincent, J., "Development of Advanced Techniques for Rotorcraft State Estimation and Parameter Identification," Systems Control, Inc., NASA Contractor Report CR 159297, August 1980.

APPENDIX A
BIAS SEPARATION THEORY FOR DISCRETE-TIME SYSTEMS

Process and Observation Models

$$x(n+1) = \Phi(n)x(n) + B(n)b + u(n)$$

$$y(n) = C(n)x(n) + D(n)b + v(n)$$

where b = constant (to be determined)

$$E[u(n)u'(k)] = Q(n)\delta_{nk} \quad , \quad E[v(n)v'(k)] = R_n\delta_{nk} \quad , \quad E[u(n)v'(k)] = 0$$

Bias-Separated Filter

$$\hat{x}(n) = x(n) + V(n)\hat{b}(n)$$

where
- $\hat{x}(n)$ = optimum estimate corrected for bias
- $\bar{x}(n)$ = bias-free estimate
- $\hat{b}(n)$ = optimum estimate of bias

} after processing observation $y(n)$

Bias-Free Filter

$$\bar{x}(n) = \tilde{x}(n) + \bar{K}(n)\bar{r}(n)$$

where
- $\tilde{x}(n) = \Phi(n-1)\bar{x}(n-1)$ = predicted state of bias-free filter
- $\bar{r}(n) = y(n) - C(n)\tilde{x}(n)$ = bias-free residual
- $\bar{K}(n)$ = bias-free filter gain matrix

Bias Estimator

$$\hat{b}(n) = \hat{b}(n-1) + K_b(n)[\bar{r}(n) - H(n)\hat{b}(n-1)]$$

where
- $H(n) = C(n)U(n) + D(n)$
- $K_b(n)$ = bias estimator gain matrix

Matrix Propagation Equations

Bias Free Gain: $$\bar{K}(n) = \tilde{P}(n)C(n)[C(n)\bar{P}(n)C(n) + R(n)]^{-1}$$

Prior Covariance: $$\tilde{P}(n+1) = \Phi(n)\tilde{P}(n)\Phi'(n) + Q(n)$$

Posterior Covariance: $$\bar{P}(n) = [I - \bar{K}(n)C(n)]\tilde{P}(n)$$

Bias Gain: $$K_b(n) = M(n+1)H(n)R^{-1}(n)$$

Bias Covariance: $$M^{-1}(n+1) = M^{-1}(n) + H'(n)[C(n)\tilde{P}(n)C'(n) + R(n)]^{-1}$$

$$U(n+1) = \Phi(n)V(n) + B(n)$$

$$V(n) = U(n) - \bar{K}(n)H(n)$$

COMPARISONS OF NONLINEAR FILTERS FOR SYSTEMS WITH NON-NEGLIGIBLE NONLINEARITIES

by

Dr. D. F. Liang
Defence Research Establishment Ottawa
Shirley's Bay, Ottawa
Canada K1A 0Z4

SUMMARY

This paper examines the structural differences and performance characteristics of several distinct estimation algorithms, as applied to some practical continuous and discrete-time state estimation problems. The extensive simulation results presented, indicate that when noise inputs are not "too small" and appropriate *a priori* estimates are available, the extended Kalman filter can be expected to perform satisfactorily. When nonlinear effects are significant, the realizable minimum variance filter is remarkably superior to any other filter investigated. When the level of noise inputs are large enough to effectively cover the effects of nonlinearities, no particular filter can be said to be consistently superior to any other filter.

CONTENTS

		PAGE
1.	INTRODUCTION	1
2.	COMPARISONS OF CONTINUOUS - TIME NONLINEAR FILTERS	2
2.1	Introduction	2
2.2	Nonlinear Filters for Phase-Lock Loop	4
2.3	Nonlinear Filters for Van der Pol's Oscillator	5
3.	COMPARISONS OF DISCRETE - TIME NONLINEAR FILTERS	6
3.1	Introduction	6
3.2	Nonlinear Systems with Quadratic Measurement Nonlinearities	7
3.3	Simulation Results and Discussions	8
3.3.1	State and Parameter Estimation with Quadratic Measurement	8
3.3.2	State and Parameter Estimation with Linear Measurement	9
4.	CONCLUSIONS	9
	REFERENCES	10

SECTION 1

INTRODUCTION

One practical problem of great importance in control theory is the estimation of the state of a physical system, on the basis of noisy measurements. For linear systems with additive white noise the procedure for obtaining optimal unbiased minimum variance estimates was first formulated by Kalman and Bucy [1], and it has been successfully applied to numerous engineering and scientific problems.

In contrast to this, truly optimal nonlinear estimation algorithms have not been practically implementable, since it requires an infinite dimensional system to realize. Therefore, considerable attention has been devoted to methods of approximating the *a posteriori* density functions based on perturbations relative to a prescribed reference. The majority of these techniques [2-4] employ the Taylor's series expansions of the dynamic and measurement nonlinearities, neglecting second or higher-order terms.

As long as the second-order (and higher-order) terms in the perturbation equations are negligible, the application of first-order extended Kalman filters (EKF) has been found to yield valid and satisfactory results. If nonlinearities are significant, however, first-order approximations of the system equations are inadequate, and the EKFs tend to be unstable and exhibit divergent behaviour. In some of these cases, Kushner [5] and Athans et al [6] reported that filter performance can be substantially improved by local iterations or the inclusion of second-order effects. On the other hand, Schwartz and Stear [7] on the basis of their simulation study, concluded that the added computational complexity of several second-order filters may not provide useful improvements relative to the EKF. In Kushner's simulation [5] of the Van der Pol's equation, he reported that the implementation of a modified Gaussian second-order filter cannot prevent the

filter from diverging. Jazwinski [8] in his textbook, stated that it is questionable whether higher-order approximations would improve performance in cases where the extended Kalman filter does not work at all (diverges).

Attempting to alleviate some of the difficulties of the Taylor series expansion approach, Sunahara [9] proposed to replace nonlinear dynamic functions by quasi-linear functions via statistically optimized approximation. His results and also those of Austin and Leondes [10] indicate that this approach may be more accurate than those of Taylor series expansions.

Recently, Liang and Christensen [11] derived a realizable minimum variance estimation algorithm for nonlinear continuous systems using the matrix minimum principle together with the Kolmogorov and Kushner's equations. They also applied the matrix minimum principal to derive nonlinear estimation algorithms [12] for discrete-time nonlinear time-delayed systems with measurements corrupted by white noise and non-white noise processes. They noted that for systems with polynomial, product-type or state-dependent sinusoidal nonlinearities [13], their proposed minimum variance algorithms can be practically realized without the need of approximation under the assumption that the estimation errors are Gaussian.

Since it is difficult to theoretically assess the virtues of any one nonlinear filter vis-a-vis the others, it is necessary to conduct extensive numerical simulations and tests to provide meaningful comparisons between the performance characteristics of the filters. Simulations of various nonlinear filters not only could provide considerable insight into the stability behaviour of some of these filters, but also provide ad hoc guidelines to establish situations in which specific nonlinear algorithms would have demonstrable advantages.

In Section 2, we deal with state estimation problems of continuous-time nonlinear dynamic systems. Various structures of dynamic continuous-time finite dimensional nonlinear filter are tabulated. Two types of dynamic systems with non-negligible nonlinearities were selected and simulated on a digital computer. Extensive simulation results accompanied with discussion, are presented to compare the performance behavior of these filters.

In Section 3, we deal with state estimation problems of discrete-time nonlinear systems. Section 3.1 presents a brief summary of Liang and Christensen's nonlinear discrete-time filtering algorithm [12]. Section 3.2 applies their minimum variance filtering algorithm (MVF) to a general class of discrete-time state estimation problems, where the measurement model and/or system model contains second-order nonlinearities. Simulation results accompanied by discussion are presented in Section 3.3 to compare the performance behavior of the MVF and EKF.

SECTION 2

COMPARISONS OF CONTINUOUS-TIME NONLINEAR FILTERS

2.1 INTRODUCTION

Consider a class of nonlinear systems described by the stochastic differential equation [8]

$$\frac{dx(t)}{dt} = f[x(t), t] + G[x(t), t]\, w(t) \tag{2.1}$$

with measurement given by

$$z(t) = h[x(t), t] + v(t) \tag{2.2}$$

Where $x(t)$ and $z(t)$ are the n-dimensional state and m-dimensional measurement vectors, f and h are, respectively, n- and m-dimensional nonlinear vector valued functions, and G is a vector valued matrix.

The random vectors $w(t)$ and $v(t)$ are statistically independent zero-mean white Gaussian noise processes such that for all t, $\tau \geq t_0$

$$\begin{aligned} \mathrm{Cov}[w(t), w(\tau)] &= V_w(t)\, \delta(t-\tau) \\ \mathrm{Cov}[v(t), v(\tau)] &= V_v(t)\, \delta(t-\tau) \end{aligned} \tag{2.3}$$

and

$$\mathrm{Cov}[w(t), v(\tau)] = 0$$

Where $\delta(.)$ is the Dirac delta function, and the variances $V_w(t)$ and $V_v(t)$ are non-negative definite and positive definite, respectively.

The initial state vector $x(t_0) = x_0$ is a zero-mean Gaussian random process, independent of $w(t)$ and $v(t)$ for $t \geq t_0$, with a positive definite variance matrix

$$\mathrm{Var}[x(t_0), x(t_0)] = V_x(t_0)$$

In the design of nonlinear filters, a number of different exact and approximate nonlinear state and error-variance equations have been proposed in the literature [3,7,8,9,11]. For comparitive purposes, various structures of these nonlinear filters are tabulated in Table 2.1.

TABLE 2.1 VARIOUS EXACT AND APPROXIMATE NONLINEAR FILTERS

Message Model: $\dot{x}(t) = f[x(t),t] + G[x(t),t]w(t)$

Measurement Model: $z(t) = h[x(t),t] + v(t)$

FILTER NOMENCLATURE	FILTER DYNAMICS	ERROR-VARIANCE EQUATIONS
Extended Kalman Filter	$\dot{\hat{x}} = f(\hat{x},t) + V_{\tilde{x}} \frac{\partial h^T(\hat{x},t)}{\partial \hat{x}} \Psi_v^{-1} \left\{ z - h(\hat{x},t) \right\}$ --(A)	$\dot{V}_{\tilde{x}} = \frac{\partial f(\hat{x},t)}{\partial \hat{x}} V_{\tilde{x}} + V_{\tilde{x}} \frac{\partial f^T(\hat{x},t)}{\partial \hat{x}} + G[\hat{x},t]\Psi_w G^T[\hat{x},t] - V_{\tilde{x}} \frac{\partial h^T(\hat{x},t)}{\partial \hat{x}} \Psi_v^{-1} \frac{\partial h(\hat{x},t)}{\partial \hat{x}} V_{\tilde{x}}$ --(B)
Modified Minimum Variance Filter	$\dot{\hat{x}} = f(\hat{x},t) + \frac{1}{2} \frac{\partial^2 f(\hat{x},t)}{\partial \hat{x}^2} : V_{\tilde{x}} + V_{\tilde{x}} \frac{\partial h^T(\hat{x},t)}{\partial \hat{x}} \Psi_v^{-1} \left\{ z - h(\hat{x},t) - \frac{1}{2} \frac{\partial^2 h(\hat{x},t)}{\partial \hat{x}^2} : V_{\tilde{x}} \right\}$ --(C)	$\dot{V}_{\tilde{x}}$ = Right-hand-side of equation (B)
Truncated Minimum Variance Filter	Same as equation (C)	$\dot{V}_{\tilde{x}}$ = Right-hand-side of equation (B) $- \frac{1}{2} V_{\tilde{x}}^2 : \frac{\partial^2 h(\hat{x},t)}{\partial \hat{x}^2} : \Psi_v^{-1} \left\{ z - h(\hat{x},t) - \frac{1}{2} \frac{\partial^2 h(\hat{x},t)}{\partial \hat{x}^2} : V_{\tilde{x}} \right\}$
Quasi-Moment Minimum Variance Filter	Same as equation (C)	$\dot{V}_{\tilde{x}}$ = Right-hand-side of equation (B) $+ V_{\tilde{x}}^2 : \frac{\partial^2 h(\hat{x},t)}{\partial \hat{x}^2} : \Psi_v^{-1} \left\{ z - h(\hat{x},t) - \frac{1}{2} \frac{\partial^2 h(\hat{x},t)}{\partial \hat{x}^2} : V_{\tilde{x}} \right\}$
Stochastic Linearization Filter	$\dot{\hat{x}} = \hat{f}(\bar{x} + \hat{x},t) + E\{\tilde{x} h^T(\bar{x} + \hat{x},t)\} \Psi_v^{-1} \left\{ z - \hat{h}(\bar{x} + \hat{x},t) \right\}$ --(D)	$\dot{V}_{\tilde{x}} = E\left\{\tilde{x} f^T(\bar{x} + \hat{x},t)\right\} + E\left\{f(\bar{x} + \hat{x},t)\tilde{x}^T\right\} + G[\hat{x},t]\Psi_w G^T[\hat{x},t] - E\left\{\tilde{x} h^T(\bar{x} + \hat{x},t)\right\}\Psi_v^{-1} E\left\{h(\bar{x} + \hat{x},t)\tilde{x}^T\right\}$ --(E)
Minimum Variance	$\dot{\hat{x}}$ = Same as equation (D)	$\dot{V}_{\tilde{x}}$ = Right-hand-side of equation (E) $+ E\left\{\tilde{x}\tilde{x}^T h^T(\bar{x} + \hat{x},t) - V_{\tilde{x}} h^T(\bar{x} + \hat{x},t)\right\} \cdot \Psi_v^{-1} \left\{ z - \hat{h}(\bar{x} + \hat{x},t) \right\}$

Tensor Computation:

$$\left(\frac{\partial^2 f(\hat{x},t)}{\partial \hat{x}^2} : V_{\tilde{x}}\right)_i = \sum_{j,k=1}^{n} V_{jk} \frac{\partial^2 f_i(\hat{x},t)}{\partial \hat{x}_j \partial \hat{x}_k}$$

$$\left(V_{\tilde{x}}^2 : \frac{\partial^2 h_k(\hat{x},t)}{\partial \hat{x}^2}\right)_{ij} = \sum_{q,r=1}^{n} \frac{\partial^2 h_k(\hat{x},t)}{\partial \hat{x}_q \partial \hat{x}_r} (V_{iq} V_{jr} + V_{ir} V_{jq})$$

$$\left(V_{\tilde{x}}^2 : \frac{\partial^2 h}{\partial \hat{x}^2}\right) : \xi = \left[\sum_{k=1}^{m} \left(V_{\tilde{x}}^2 : \frac{\partial^2 h_k(\hat{x},t)}{\partial \hat{x}^2}\right)_{ij} \xi_k\right]$$

filters, the only difference is the error-forcing term that appears in the error-variance equations. In the quasi-moment filter it enters with a plus sign, in the truncated filter that term enters with a minus sign and a factor of one-half, and the modified minimum variance filter is a compromise between the truncated and the quasi-moment filters. In comparing the stochastic linearization filter with the minimum variance nonlinear filter, the later has obviously preserved the error forcing term. On the other hand, comparing the first four linearized filters with the last two filters, the essential differences are due to approximations made of expectations of $f(x)$, $\tilde{x}h(x)$ and $h(x)$. It is also noteworthy to mention that when $h(x)$ is linear, the modified, the truncated and the quasi-moment minimum variance filters are identical, and the minimum variance filter is identical to the stochastic linearization filter.

In order to compare the performance characteristics of these nonlinear filters, two different types of dynamic systems with non-negligible nonlinearities were selected, the stochastic equations were transformed to Straconovich's forms [8] and then simulated on a digital computer. The integration scheme was performed by the minimum-error-bound fourth-order Runge-Kutta method [14].

2.2 NONLINEAR FILTERS FOR PHASE-LOCK LOOP

In this section, we deal with the design of nonlinear filters for a two dimensional phase-lock loop. This phase detection problem is of major technological importance and widely known in satellite communication. The dynamic model selected consists of phase and phase rate with all the plant noise additive to the phase rate only. Namely:

$$\dot{x}_1 = x_2$$

and

$$\dot{x}_2 = w$$

the measurement model is represented as:

$$z_1 = \cos x_1 + v_1$$

$$z_2 = \sin x_1 + v_2$$

where w, v_1 and v_2 are zero mean white Gaussian noise processes, with variances V_w, V_{v_1} and V_{v_2} respectively.

Filtering equations for the extended Kalman filter (EKF), stochastic linearization filter (SLF) and minimum variance filter (MVF) are presented as follows:

Extended Kalman Filter

$$\dot{\hat{x}}_1 = \hat{x}_2 - V_{v_1}^{-1} V_{11} \sin \hat{x}_1 (z_1 - \cos \hat{x}_1) + V_{v_2}^{-1} V_{11} \cos \hat{x}_1 (z_2 - \sin \hat{x}_1)$$

$$\dot{\hat{x}}_2 = - V_{v_1}^{-1} V_{12} \sin \hat{x}_1 (z_1 - \cos \hat{x}_1) + V_{v_2}^{-1} V_{12} \cos \hat{x}_1 (z_2 - \sin \hat{x}_1)$$

$$\dot{V}_{11} = 2 V_{12} - V_{11}^2 V_{v_1}^{-1} \sin^2 \hat{x}_1 - V_{11}^2 V_{v_2}^{-1} \cos^2 \hat{x}_1$$

$$\dot{V}_{12} = V_{22} - V_{11} V_{12} V_{v_1}^{-1} \sin^2 \hat{x}_1 - V_{11} V_{12} V_{v_2}^{-1} \cos^2 \hat{x}_1$$

and

$$\dot{V}_{22} = V_w - V_{12}^2 V_{v_1}^{-1} \sin^2 \hat{x}_1 - V_{12}^2 V_{v_2}^{-1} \cos^2 \hat{x}_1$$

Stochastic Linearization Filter

Setting $E = \exp(- V_{11}/2)$

$$\dot{\hat{x}}_1 = \hat{x}_2 - V_{v_1}^{-1} V_{11} E \sin \hat{x}_1 (z_1 - E \cos \hat{x}_1) + V_{v_2}^{-1} V_{11} E \cos \hat{x}_1 (z_2 - E \sin \hat{x}_1)$$

$$\dot{\hat{x}}_2 = - V_{v_1}^{-1} V_{12} E \sin \hat{x}_1 (z_1 - E \cos \hat{x}_1) + V_{v_2}^{-1} V_{12} E \cos \hat{x}_1 (z_2 - E \sin \hat{x}_1)$$

$$\dot{V}_{11} = 2 V_{12} - V_{11}^2 E^2 V_{v_1}^{-1} \sin^2 \hat{x}_1 - V_{11}^2 E^2 V_{v_2}^{-1} \cos^2 \hat{x}_1$$

$$\dot{V}_{12} = V_{22} - V_{11} V_{12} E^2 V_{v_1}^{-1} \sin^2 \hat{x}_1 - V_{11} V_{12} E^2 V_{v_2}^{-1} \cos^2 \hat{x}_1$$

and

$$\dot{V}_{22} = V_w - V_{12}^2 E^2 V_{v_1}^{-1} \sin^2 \hat{x}_1 - V_{12}^2 E^2 V_{v_2}^{-1} \cos^2 \hat{x}_1$$

Minimum Variance Filter

The equations for $\dot{\hat{x}}_1$ and $\dot{\hat{x}}_2$ are identical with those of the S.L.F., whereas the error variance equations have the following terms in addition to those of the S.L.F:

For $\dot{V}_{11}$ add $-V_{11}^2$ E F

For $\dot{V}_{12}$ add $-V_{11}\ V_{12}$ E F where $F = \psi_{v_1}^{-1} \cos \hat{x}_1 (z_1 - E \cos \hat{x}_1) + \psi_{v_2}^{-1} \sin \hat{x}_1 (z_2 - E \sin \hat{x}_1)$

For $\dot{V}_{22}$ add $-V_{12}^2$ E F

A careful examination of these equations indicates that the major difference between the EKF and SLF is due to the approximation made of $E = \exp(-V_{11}/2)$. The effect of this term is accentuated when the measurement residual is significant, and the error variances are large. The major difference between the SLF and MVF is due to the error forcing terms which appear in the error variance equations. The effect of this term in proportion to all the other terms is accentuated, when V_{11} is small and the measurement residual is significant. The measurement residual is significant when the *a priori* state estimates are significantly different from the true values. It is also affected by the level of noise inputs.

To compare the performance characteristics of these three filters, their filtering equations were simulated on a digital computer. Each AVG output response presented in Figures 2.1 to 2.8 represents the average results of 5 simulation runs, that are representative of many other simulation runs not presented in this paper.

Figures 2.1 and 2.2 show the effects of initial state estimates for the following two sets of prior statistics:

Figure 1: $x_1(0) = x_2(0) = 2.0$, $\Psi_v = \Psi_w = 0.01$, $V_{11}(0) = V_{22}(0) = 0.01$ and $V_{12}(0) = 0$

Figure 2: $x_1(0) = x_2(0) = 2.0$, $\Psi_v = \Psi_w = 0.01$, and $V_{11}(0) = V_{22}(0) = V_{12}(0) = 1.0$

When Ψ_v and Ψ_w are increased to 0.1, simulation results for similar conditions were obtained and presented in Figures 2.3 and 2.4.

From Figures 2.1 and 2.4, it is apparent that the performance of the EKF and SLF are almost identical to that of the MVF for small noise variances with large error-variance and when *a priori* state estimates are close to the true values. However, when *a priori* state estimates are significantly different from the true values, and initial error-variances are overly-optimistic, the performance characteristics of both the EKF and SLF are significantly inferior to those of the MVF. These experimental results agree well with what was theoretically expected. It is also clear that the MVF is significantly more insensitive to the selection of *a priori* state estimates and initial error-variances.

For noise variances of one unit, AVG outputs of these three filters are presented in Figures 2.5 and 2.6. It is interesting to observe that when the noise input levels are relatively high compared to the effects of nonlinearities, the EKF is as good as any other nonlinear filter investigated, and no particular filter can be said to be consistently superior to any other nonlinear filter. This appears to be intuitively obvious, because large noise inputs can effectively "cover" neglected nonlinearities.

On the other hand, when the noise variances are reduced to 0.001, both the EKF and SLF diverged, while the MVF can track the true values of the phase and phase rate amazingly well. Typical simulation results for some of these runs are presented in Figures 2.7 and 2.8. However, it should be noted that if the initial error-variances are 1.0, the noise variances are further reduced to 0.0001, and when the initial state estimates are far from the true values of the system states, even the MVF would diverge as others had done so much sooner. But if one were to drastically reduce the initial variances, occasionally the MVF and also the EKF and SLF would all be stable again, with the MVF still performing better than the others. This indicates that in some applications of nonlinear filters, the system designer must be cautious not to select the initial variances to be too large.

2.3 Nonlinear Filters for Van Der Pol's Oscillator

Consider a state estimation problem of the Van der Pol's oscillator described by

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_1 + 3x_2(1 - x_1^2)$$

with measurement given by

$$z = x_1 - 0.1x_1^3 + v$$

This example was selected not only because of its third-order nonlinearities, but also that Kushner had previously shown that for this state estimation problem even with a linear measurement model, the linearized filter was extremely unstable and was completely useless within a fraction of a time unit. Even the implementation of a Gaussian second-order filter proved to be unstable. Therefore, in this study, we are only concerned with the comparison of a minimum variance filter with a stochastic linearization filter.

The structural difference between these two filters is mainly due to the preservation of error forcing terms in the MVF. Namely:

For $\dot{V}_{11}$ the additional term is $-0.6\ V_{11}^2\ \hat{x}_1\ \Psi_v^{-1}\ [z - \hat{z}]$

For $\dot{V}_{22}$ the additional term is $-0.6\ V_{12}^2\ \hat{x}_1\ \Psi_v^{-1}\ [z - \hat{z}]$

For $\dot{V}_{12}$ the additional term is $-0.6\ V_{11}\ V_{12}\ \hat{x}_1\ \Psi_v^{-1}\ [z - \hat{z}]$

where $\hat{z} = \hat{x}_1 - 0.1(\hat{x}_1^3 + 3\hat{x}_1\ V_{11})$

Therefore, it can be expected that the difference between the SLF and MVF will be accentuated whenever the values of initial error-variances and $\hat{x}_1$ are increased and also when the measurement residual is significant.

To verify this, some typical results of extensive simulations are presented in Figures 2.9 to 2.13. Here, we have

$$\hat{x}_1(0) = x_1(0) = 2.0,\ \hat{x}_2(0) = x_2(0) = 0.0 \text{ and } V_{12}(0) = 0.0$$

Figure 2.9 illustrates the effects of noise corruption on filter measurement input. Figures 2.10 to 2.12 illustrate the effects of noise variances and initial error-variances on the performance of the SLF and MVF. From these, it is evident that when initial error-variances are very small, and the *a priori* estimates are reasonably accurate, the performance characteristics of the SLF and MVF are almost identical, with the SLF having a slightly smaller phase-shift. However, as expected from theoretical considerations, for increasing values of initial error-variances and noise variances, the difference between the performance of the SLF and MVF becomes more significant. Here, quite consistently, the MVF is far more capable than the SLF in tracking the true values of the states and the former is also much less sensitive to these selections of initial error-variances.

To further illustrate the difference in filter responses, Figure 2.13 presents the filter state estimates, the errors of state estimates and their one sigma error bounds for the following statistics:

$$\Psi_v = 0.1,\ V_{11}(0) = V_{22} = 0.5 \text{ and } V_{12}(0) = 0$$

The poorer performance of the SLF is evidently due to its optimistic estimation of error bounds.

Figure 2.14 is presented to demonstrate the severe effects of initial state estimates and error-variances on the performance of the SLF and MVF. It is quite evident that the system designer must be careful in his selections of appropriate initial error-variances. The extensive simulation experience of the author suggests that in the design of the SLF and MVF for dynamic systems with non-negligible nonlinearities, it is important not to use overly-pessimistic initial error-variances, since error-variances that were too large could excessively damp the system dynamics and Kalman gain matrix to reject some of the valuable measurement data inputs.

SECTION 3

COMPARISON OF DISCRETE-TIME NONLINEAR FILTERS

3.1 Introduction

Consider a general class of discrete-time nonlinear systems described by

$$x(k + 1) = f[x(k), k] + G[x(k), k]w(k) \tag{3.1}$$

with measurements represented by

$$z(k) = h[x(k), k] + v(k) \tag{3.2}$$

where the state x is an n-vector; the measurement z an m-vector; the state noise sequence w an r-vector; the measurement noise v an s-vector; G is a non-linear state-dependent $n \times r$ matrix; f and h are, respectively, n- and m-dimensional.

The random vectors $w(k)$ and $v(k)$ are independent zero-mean white Gaussian sequences, for which

$$E_k[w(k)w^T(j)] = \Psi_w(k)\ \delta_{kj} \tag{3.3}$$

$$E_k[v(k)v^T(j)] = \Psi_v(k)\ \delta_{kj} \tag{3.4}$$

and

$$E_k[w(k)v^T(j)] = 0 \tag{3.5}$$

for all integers k and j, where $E_k[.]$ denotes the expectation operation conditioned on $Z(k) = [z(0), z(1), \ldots, z(k)]$. Ψ_w and Ψ_v are $r \times r$ and $m \times m$ positive definite matrices, respectively.

The discrete-time realizable minimum variance unbiased estimate $\hat{x}(k + 1/k + 1)$ of the state vector $x(k + 1)$ has been derived by Liang and Christensen [12]. Since their results will be repeatedly used throughout this paper, they are stated here for quick reference.

Lemma 1: The minimum variance estimate of $x(k+1)$ is given by the following set of equations:

$$\hat{x}(k+1/k+1) = \hat{x}(k+1/k) + K_{k+1}\{z(k+1) - \hat{h}[x(k+1),k+1/k]\} \tag{3.6}$$

where we have:

$$\hat{x}(k+1/k) = \hat{f}[x(k),k/k] = E_k\{f[x(k),k]/k\} \tag{3.7}$$

$$K_{k+1} = E_k\{\tilde{x}(k+1/k)\tilde{h}^T[x(k+1)/k]\}\{V_v(k+1) + E_k\{\tilde{h}[x(k+1)/k]\tilde{h}^T[x(k+1)/k]\}\}^{-1} \tag{3.8}$$

$$V_{\tilde{x}}(k+1/k) = E_k\{\tilde{f}[x(k),k/k]\tilde{f}^T[x'k),k/k]\} + G[x(k),k]V_w(k)G^T[x(k),k] \tag{3.9}$$

and

$$V_{\tilde{x}}(k+1/k+1) = V_{\tilde{x}}(k+1/k) - K_{k+1}\,E_k\{\tilde{h}[x(k+1)/k]\tilde{x}^T(k+1/k)\} \tag{3.10}$$

In order to provide an insight into the structure of Liang and Christensen's realizable nonlinear filtering algorithm, in the next section their filtering algorithms will be applied to estimate the states of a specific class of nonlinear discrete-time systems.

3.2 Nonlinear Systems with Quadratic Measurement Nonlinearities

In the special case of the state estimation problem for discrete-time dynamic systems described by equation (3.1) with measurements represented by

$$z(k) = H_k x_k + \sum_{j=1}^{m} \phi_j x_k^T E_k^j x_k + v_k \tag{3.11}$$

where E_k^j represents a set of m symmetric n x n matrices. $\phi_1, \phi_2, \ldots, \phi_m$ denote the natural basis vectors:

$$\phi_1 = \begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}, \quad \phi_2 = \begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix}, \quad \ldots, \quad \phi_n = \begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix} \tag{3.12}$$

Then equations (3.6) to (3.8) can easily be shown to be:

$$\hat{x}(k+1/k+1) = \hat{x}(k+1/k) + K_{k+1}\{z(k+1) - (H_{k+1} + \sum_{j=1}^{m}\phi_j \hat{x}^T(k+1/k)E_{k+1}^j)\hat{x}(k+1/k) - \sum_{j=1}^{m}\phi_j \operatorname{trace}(E_{k+1}^j V_{\tilde{x}}(k+1/k))\} \tag{3.13}$$

$$\hat{x}(k+1/k) = \hat{f}[x(k),k/k] \tag{3.14}$$

$$K_{k+1} = V_{\tilde{x}}(k+1/k)\{H_{k+1} + 2\sum_{j=1}^{m}\phi_j \hat{x}^T(k+1/k)E_{k+1}^j\}^T \Big\{V_v(k+1) + (H_{k+1} + 2\sum_{j=1}^{m}\phi_j \hat{x}^T(k+1/k)E_{k+1}^j) \cdot V_{\tilde{x}}(k+1/k)\{H_{k+1} + 2\sum_{\ell=1}^{m}\phi_\ell \hat{x}^T(k+1/k)E_{k+1}^\ell\}^T + \sum_{j,\ell=1}^{m} 2\phi_j \operatorname{trace}(E_{k+1}^j V_{\tilde{x}}(k+1/k)E_{k+1}^\ell V_{\tilde{x}}(k+1/k))\phi_\ell^T\Big\}^{-1} \tag{3.15}$$

A numerically stable form of the error variance equation is represented by:

$$V_{\tilde{x}}(k+1) = \{I - K_{k+1}(H_{k+1} + 2\sum_{j=1}^{m}\phi_j \hat{x}^T(k+1/k)E_{k+1}^j)\} V_{\tilde{x}}(k+1/k)\Big\{I - K_{k+1}(H_{k+1} + 2\sum_{j=1}^{m}\phi_j \hat{x}(k+1/k)E_{k+1}^j)\Big\}^T + K_{k+1}\{V_v(k+1) + 2\sum_{j,\ell=1}^{m}\phi_j \operatorname{trace}(E_{k+1}^j V_{\tilde{x}}(k+1/k) E_{k+1}^\ell V_{\tilde{x}}(k+1/k))\}K_{k+1}^T \tag{3.16}$$

The equation for $V_{\tilde{x}}(k+1/k)$ remained the same as that of equation (3.9).

In the special case that the message model of (3.1) is represented by:

$$x(k+1) = \Phi_k x_k + \sum_{i=1}^{n}\phi_i x_k^T C_k^i x_k + w_k$$

where C_k^i represents a set of n symmetric n x n matrices.

Then equations (3.14) and (3.9) are respectively, further reduced to:

$$\hat{x}(k+1/k) = \Phi_k \hat{x}(k/k) + \sum_{i=1}^{n}\phi_i(\hat{x}^T(k/k)C_k^i \hat{x}(k/k) + \operatorname{trace}\{C_k^i V_{\tilde{x}}(k/k)\})$$

and

$$V_{\tilde{x}}(k+1/k) = [\Phi_k + 2\sum_{i=1}^{n} \phi_i \hat{x}^T(k/k)G_k^i]V_{\tilde{x}}(k/k)$$

$$[\Phi_k + 2\sum_{q=1}^{n} \phi_q \hat{x}^T(k/k)G_k^q]^T + \Psi_w(k)$$

$$+ \sum_{i,q=1}^{n} 2\phi_i \text{ trace } [G_k^i V_{\tilde{x}}(k/k)G_k^q V_{\tilde{x}}(k/k)]\phi_q^T$$

It should be noted that for state estimation problems with measurements containing quadratic functions of the states, the realizable nonlinear estimation algorithms presented here are numerically stable. In the special case that both the message and measurement models consist of some quadratic functions of the states, the results presented here are identical to those derived by H.W. Sorenson [15].

3.3 Simulation Results and Discussions

3.3.1 State and Parameter Estimation with Quadratic Measurement

In order to test and compare the performance characteristics of the minimum variance estimator with those of the extended Kalman filter, two simple state and parameter estimation problems were selected. The first one considered here is represented by the following equations:

$$x_1(k+1) = x_1(k) - x_1(k)\, x_2(k) + w_1$$
$$x_2(k+1) = x_2(k) + w_2$$
$$z(k) = x_1(k)\, x_2(k) + v$$

where noise variances of v, w_1 and w_2 are respectively, Ψ_v, Ψ_{w_1} and Ψ_{w_2}.

The dynamic structures of the minimum variance filter and the extended Kalman filter are represented in the following equations:

Extended Kalman Filter	Additional Terms for MVF
1. $\hat{x}_1(k+1/k) = \hat{x}_1(k) - \hat{x}_1(k)\,\hat{x}_2(k)$	$- V_{12}(k)$
2. $\hat{x}_2(k+1/k) = \hat{x}_2(k)$	None
3. $V(k+1/k) = \Phi(k+1/k)\, V(k/k)\, \Phi^T(k+1/k) + Q$ where $\Phi(k+1/k) = \begin{bmatrix} 1-\hat{x}_2(k+1/k) & -\hat{x}_1(k+1/k) \\ 0 & 1 \end{bmatrix}$ and $Q = \begin{bmatrix} \Psi_{w_1} & 0 \\ 0 & \Psi_{w_2} \end{bmatrix}$	$+ JG_M$ where $G_M = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $J = V_{12}^2(k) + V_{11}(k)V_{22}(k)$
4. $K = V(k+1/k)H^T W^{-1}$ where $H = [\hat{x}_2(k+1/k) \quad \hat{x}_1(k+1/k)]$	None
5. $W = H\, V(k+1/k)H^T + \Psi_v$	$+ V_{12}^2(k+1/k) + V_{11}(k+1/k)V_{22}(k+1/k)$
6. $V(k+1) = (I - KH)V(k+1/k)(I - KH)^T + K\,\Psi_v\,K^T$	$+ K[V_{12}^2(k+1/k) + V_{11}(k+1/k)V_{22}(k+1/k)]K^T$
7. $\hat{z} = \hat{x}_1(k+1/k)\hat{x}_2(k+1/k)$	$+ V_{12}(k+1/k)$
8. $\hat{x}(k+1) = \hat{x}(k+1/k) + K(z - \hat{z})$	None

It is apparent that the major difference between the EKF and MVF is due to the terms $V_{12}^2 + V_{11}\,V_{22}$ and V_{12}. Therefore it can be expected that the performance characteristics of these two filters will be greatly affected by the selections of initial variances, in particular, the choice of $V_{12}(0)$. In order to compare the performance of these two filters, some typical results of extensive simulations are presented in Figures 3.1 to 3.6. Here, each of the graphs presented represents the average results of 150 simulation runs.

Figure 1 shows the MVF filter output responses for perfect *a priori* initial state estimates, with small noise variances and initial error-variances. For this particular case it is obvious that the MVF is only marginally better than the EKF.

Figure 3.2 shows the effects of initial state estimates on filter performance for

$$x_1(0) = x_2(0) = 2.0,\ V_{11}(0) = V_{12}(0) = V_{22}(0) = 0.01,\ \text{and}\ \Psi_v = \Psi_w = 0.01$$

When the initial error-variances are increased to the value of 1.0, simulation results for similar conditions are shown in Figure 3.3, while Figure 3.4 shows their estimator errors and one sigma error bounds. When simulation results of Figures 3.2 and 3.3 are compared with the outputs of the true states presented in

Figure 3.1(a), it is evident that the MVF is much more insensitive to the choice of initial state estimates. Moreover, it is observed that when the appropriate values of state estimates are made available, the EKF is slightly better than the MVF. But when the *a priori* state estimates are far different from the true values of system states, the MVF is much superior than the EKF for both large and small initial variances. However, when the noise variances are increased, the gap between the MVF and EKF gradually diminishes. For these cases, the EKF performs almost as well as the MVF.

Figures 3.5 and 3.6 further demonstrates the effects of initial variances V_{11}, V_{22} and V_{12} on the performance of the two filters. It is apparent that the selection of initial V_{12} is as important as the selection of V_{11} and V_{22}. it seems that the selection of an inappropriate $V_{12}(0)$ may have more significant effects on the stability and performance of both filters. From the author's simulation experience it is noted that in the design of the EKF for systems with non-negligible nonlinearities, the designer should be careful in not using overly-optimistic initial error-variances. For the MVF, the designer should be careful not to use overly-pessimistic initial error-variances, since error-variances that were too large could excessively enlarge the added measurement variances. As a result, valuable measurement data could end up being rejected.

3.3.2 State and Parameter Estimation With Linear Measurement

Another simple numerical example that was investigated assumed the system model of Section 3.3.1 with measurement given by

$$z(k + 1) = x_1(k) + v(k)$$

Numerical results presented in Figures 3.7 to 3.10 were all obtained from 150-run Monte Carlo simulations.

Figures 3.7 and 3.8 show the effects of initial state estimates on the filter performance for the following set of prior statistics

$$\Psi_v = \Psi_w = 0.01,\ V_{11}(0) = V_{22}(0) = V_{12}(0) = 1.0 \text{ and } x_1(0) = x_2(0) = 2.0$$

The results presented here are quite similar to those presented in Figures 3.3 and 3.4. Therefore, the author's comments on Figures 3.3 and 3.4 are also equally applicable here.

To further demonstrate the effects of initial variances V_{11}, V_{22} and V_{12} on the performance of the MVF and EKF, Figures 3.9 and 3.10 are presented assuming

$$\hat{x}_1(0) = \hat{x}_2(0) = -2.0 \text{ and } x_1(0) = x_2(0) = 2.0$$

Where Figures 3.9 and 3.10 assumed noise variances of 0.01 and 1.0, respectively. It is apparent that the performance characteristics of the MVF and EKF are quite similar to those of Figures 3.5 and 3.6, where the MVF is significantly better and more insensitive to the selections of error-variances than the EKF. Various comments that were made in Section 3.3.1, are equally applicable for this subsection.

SECTION 4: CONCLUSIONS

This paper has presented a brief summary on the comparisons of dynamic structures for various finite dimensional filters. Extensive simulation results accompanied with discussions, were presented to compare the performance behaviour of some of these filters.

From the extensive numerical results obtained one can derive several conclusions, some of the most important are stated here.

1. When the level of noise inputs is large enough to effectively cover the effects of nonlinearities, no particular filter can be said to be consistently superior to any other filter.

2. When the noise inputs are not "too small" (relative to the effects of nonlinearities), and as long as the *a priori* estimates are available, the extended Kalman filter can be expected to perform as well as any other nonlinear filter.

3. When nonlinear effects are non-negligible, the performance of the realizable minimum variance filter is far superior to any other filter investigated, it is also much more insensitive to the choice of *a priori* estimates.

4. In general, in the design of the EKF for dynamic systems with non-negligible nonlinearities, the designer should be cautious not to select overly optimistic initial error-variances. But in the design of the MVF, the designer should be cautious not to select overly pessimistic initial error-variances.

It should also be noted that for nonlinear systems with polynomial, product-type and sinusoidal nonlinearities, the derivation and implementation of the MVF would only be slightly more difficult than the EKF or the SLF, etc. But the MVF could sometimes be much more accurate and stable than the other estimators investigated.

REFERENCES

1. R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Trans. ASME, J. Basic Engineering, vol. 82D, 1960, pp. 35-45.

2. D.M. Detchmendy and R. Sridhar. "Sequential Estimation of States and Parameters in Noisy Nonlinear Dynamical Systems," Trans. ASME, J. Basic Engineering, vol. 88D, 1966, pp. 362-368.

3. R.W. Bass and V.D. Norum and L. Schwartz, "Optimal Multi-Channel Nonlinear Filtering," J. Math. Anal. Applications, vol. 16, 1966, pp. 152-164.

4. H. Cox, "On the Estimation of State Variables and Parameters for Noisy Dynamic Systems," IEEE Trans. Aut. Control, vol. AC-9, 1964, pp. 5-12.

5. H.J. Kushner, "Approximations to Optimal Nonlinear Filters," IEEE Trans. Aut. Control, vol. AC-5, 1967, pp. 546-556.

6. M. Athans, R.P. Wishner, and A. Bertolini, "Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements," IEEE Trans. Aut. Control, vol AC-13, 1968, pp. 504-514.

7. L. Schwartz and E.B. Stear," A Computational Comparison of Several Nonlinear Filters," IEEE Trans. Aut. Control, vol. AC-13, pp. 83-86.

8. A.H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

9. Y. Sunahara, "An Approximate Method of State Estimation for Nonlinear Dynamical Systems," Trans. ASME, J. Basic Engineering, vol. 92D, 1970, pp. 385-393.

10. J.W. Austin and C.T. Leondes, "Statistically Linearized Estimation of Reentry Trajectories," IEEE Trans. Aerospace and Electronic Systems, vol. AES-17, no. 1, pp. 54-61, 1981.

11. D.F. Liang and G.S. Christensen, "Exact and Approximate State Estimation for Nonlinear Dynamic Systems," IFAC, Automatica, vol. 11, 1975, pp. 603-613.

12. D.F Liang and G.S. Christensen, "Estimation for Discrete Nonlinear Time-Delayed Systems and Measurements with Correlated and Coloured Noise Processes," Int. J. Control, vol. 28, 1978, pp. 1-10.

13. D.F. Liang, "Exact and Approximate Nonlinear Estimation Techniques," NATO Agardograph 256, 1981.

14. A. Ralston, A First Course in Numerical Analysis, McGraw-Hill, New York.

15. H.W. Sorenson, "Approximate Solutions of the Nonlinear Filtering Problem," 1977, IEEE Decision and Control Conference, pp. 620-625.

ACKNOWLEDGEMENTS

This work was carried out with the generous support of the Defence Research Establishment Ottawa (DREO), Canada. The author wishes to thank Mr.C.R. Iverson, Chief, DREO and Mr. K.A. Peebles, Head Electromagnetics Section, for their encouragement. Thanks are also due to Mr. W. Royds for his contribution, especially in his superb software programming effort, and Miss B.L. Pershaw for her patience in typing the manuscript.

FIGURE 2.1 PHASE-LOCK LOOP SIMULATION; AVG OUTPUTS OF X_1 AND X_2 (5-RUN AVERAGES)

$V_{11}(0) = V_{22}(0) = 0.01$

$V_{12}(0) = 0.0$

$X_1(0) = X_2(0) = 2.0$

$\Psi_V = \Psi_W = 0.01$

T TRUE STATE

V MINIMUM VARIANCE FILTER

L STOCHASTIC LINEARIZATION FILTER

E EXTENDED KALMAN FILTER

FILTER ESTIMATES AND TRUE X_1 VS TIME

FILTER ESTIMATES AND TRUE X_2 VS TIME

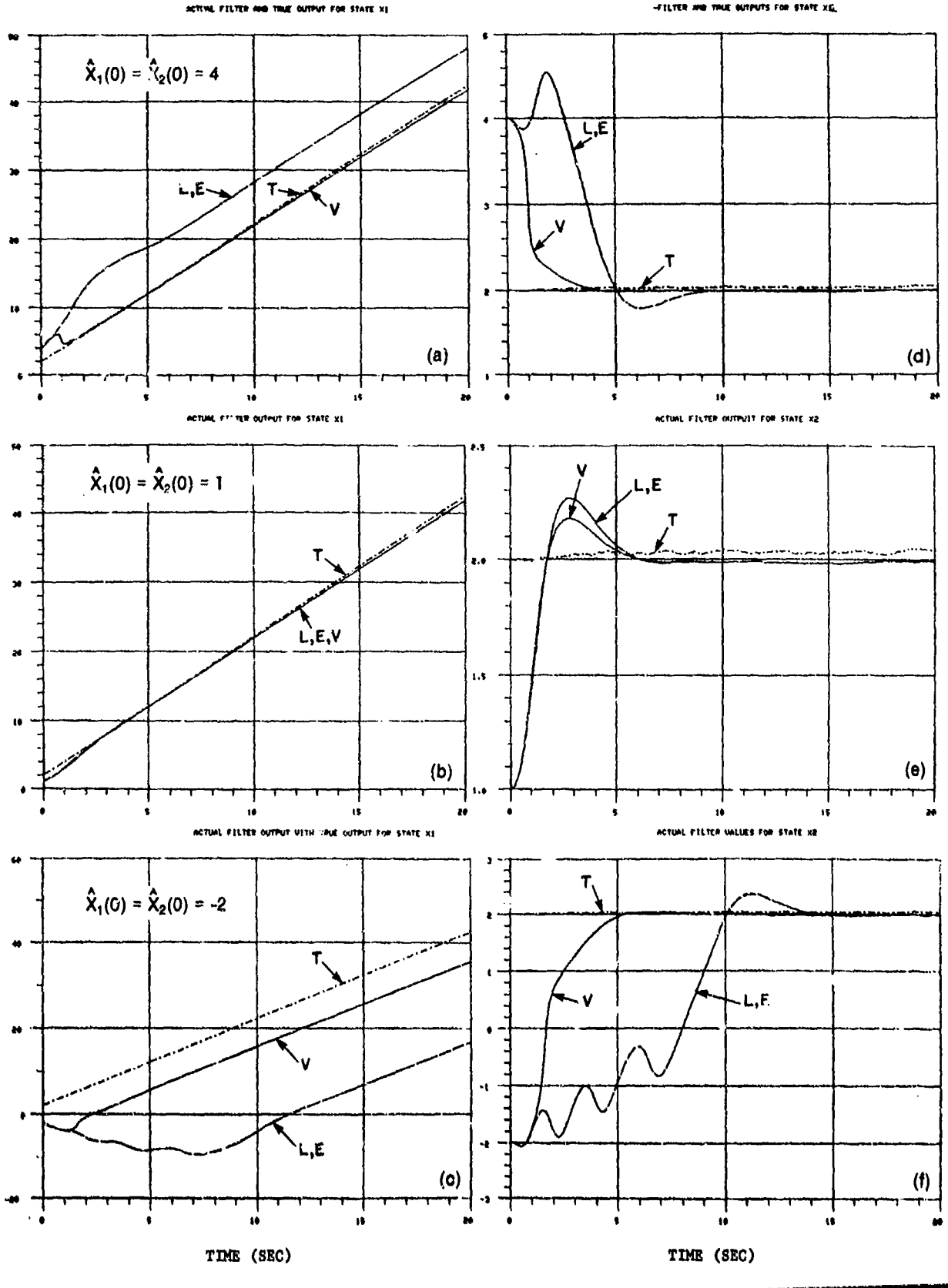


FIGURE 2.2: PHASE-LOCK LOOP; EFFECTS OF $\hat{X}(0)$ WITH SMALL Ψ AND LARGER V

$V_{11}(0) = V_{22}(0) = 1.0$

$V_{12}(0) = 1.0$

$X_1(0) = X_2(0) = 2.0$

$\Psi_V = \Psi_W = 0.01$

T TRUE STATE

V MINIMUM VARIANCE FILTER

L STOCHASTIC LINEARIZATION FILTER

E EXTENDED KALMAN FILTER

FILTER ESTIMATES AND TRUE X_1 VS TIME

FILTER ESTIMATES AND TRUE X_2 VS TIME

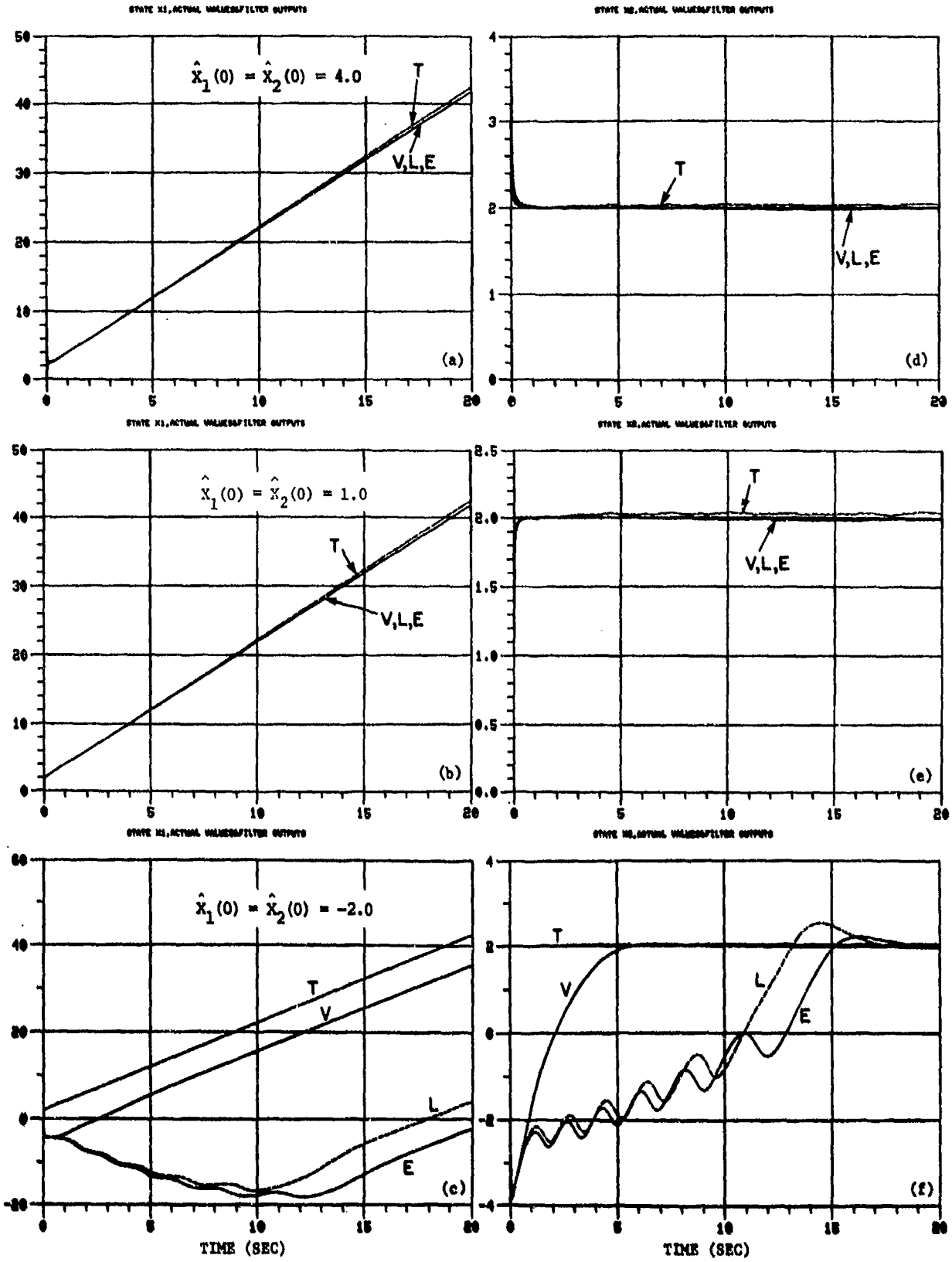


FIGURE 2.3 PHASE-LOCK LOOP SIMULATION; AVG OUTPUTS OF X_1 AND X_2 (5-RUN AVERAGES)

$V_{11}(0) = V_{22}(0) = 0.01$

$V_{12}(0) = 0.0$

$X_1(0) = X_2(0) = \hat{X}_1(0) = 2.0$

$\Psi_V = \Psi_W = 0.1$

T TRUE STATE

V MINIMUM VARIANCE FILTER

L STOCHASTIC LINEARIZATION FILTER

E EXTENDED KALMAN FILTER

FILTER ESTIMATES AND TRUE X_1 VS TIME

FILTER ESTIMATES AND X_2 VS TIME

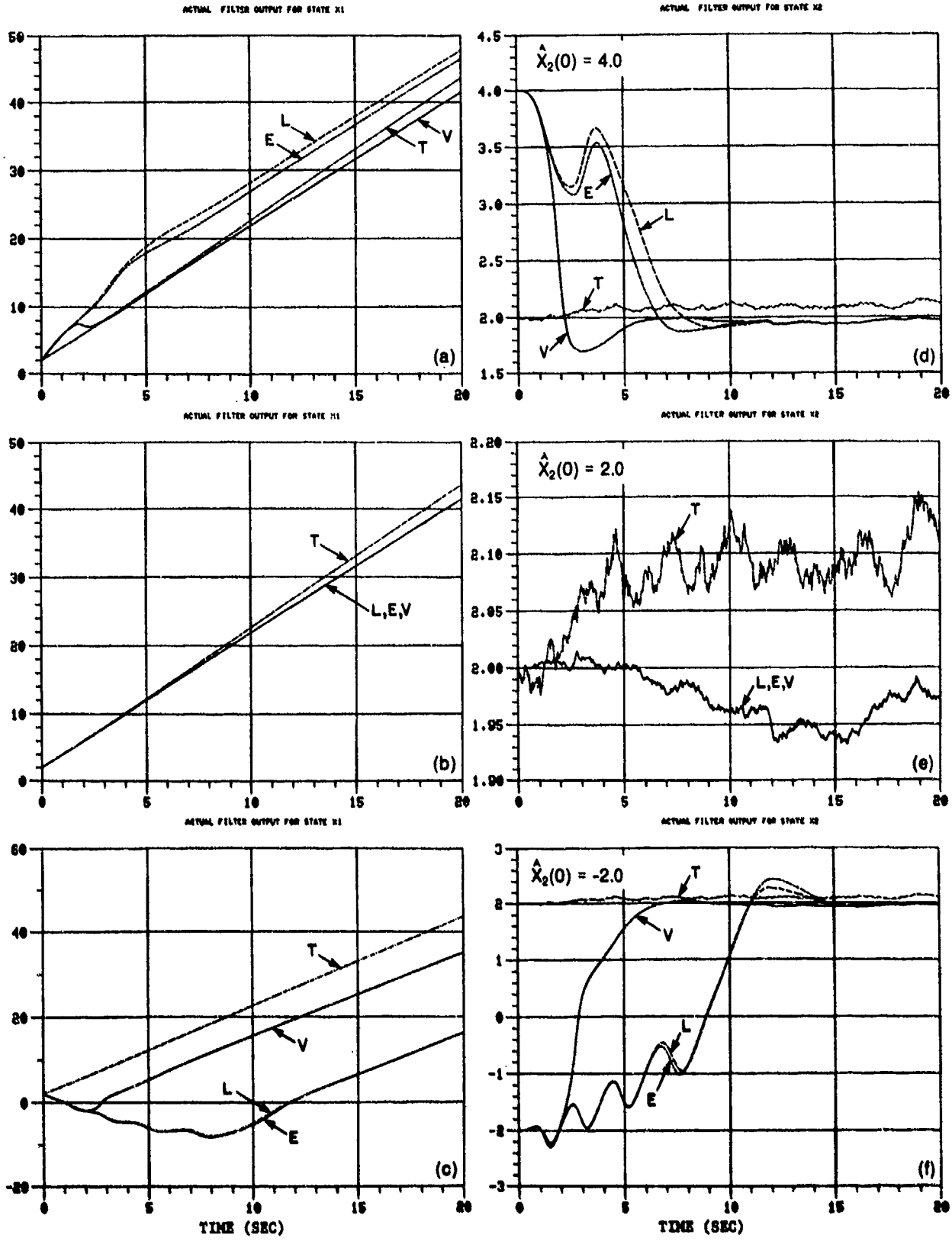


FIGURE 2.4 PHASE-LOCK LOOP SIMULATION; AVG OUTPUTS OF X_1 AND X_2 (5-RUN AVERAGES)

$V_{11}(0) = V_{22}(0) = 1.0$

$V_{12}(0) = 1.0$

$X_1(0) = X_2(0) = \hat{X}_1(0) = 2.0$

$\Psi_V = \Psi_W = 0.1$

T TRUE STATE

V MINIMUM VARIANCE FILTER

L STOCHASTIC LINEARIZATION FILTER

E EXTENDED KALMAN FILTER

FILTER ESTIMATES AND TRUE X_1 VS TIME

FILTER ESTIMATES AND TRUE X_2 VS TIME

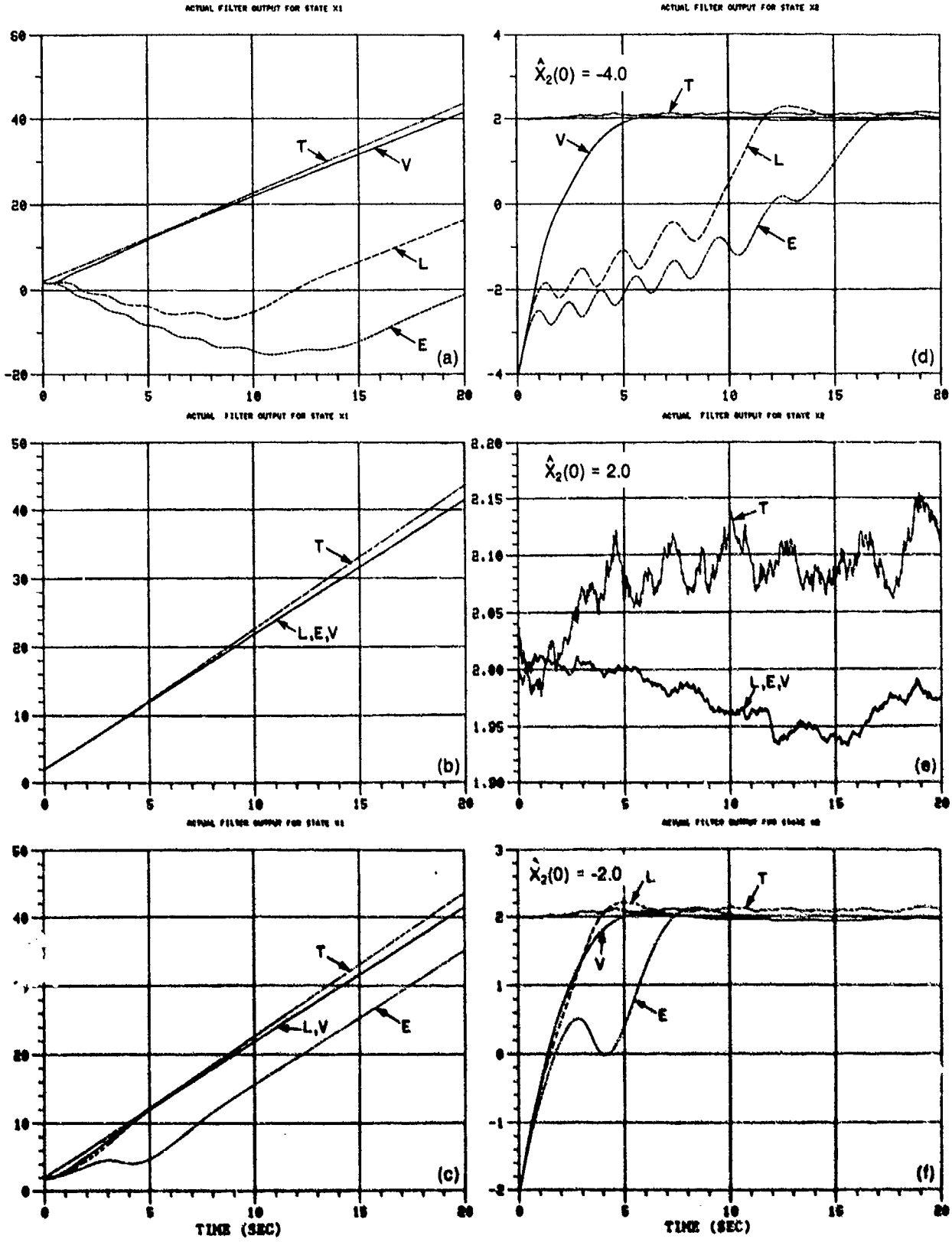


FIGURE 2.5:RMS OUTPUTS OF PHASE-LOCK LOOP WITH LARGE Ψ

$V_{11}(0) = V_{22}(0) = 0.01$ $\quad \hat{X}_1(0) = \hat{X}_2(0) = 1.0$

$V_{12}(0) = 0.0$ $\quad \Psi_V = \Psi_W = 1.0$

$X_1(0) = X_2(0) = 2$

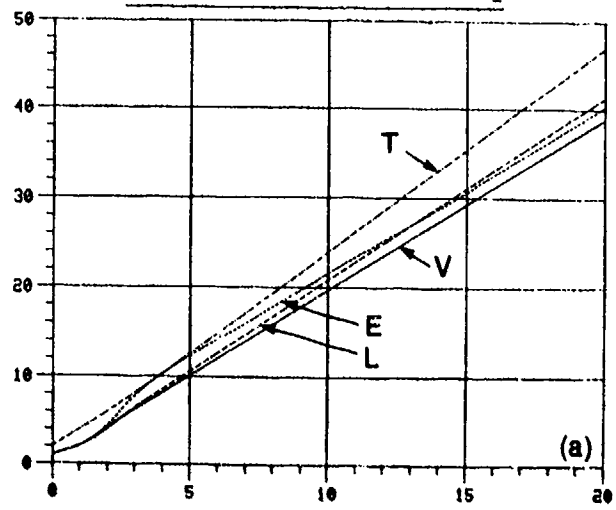
T TRUE STATE

V MIN. VAR. FILTER

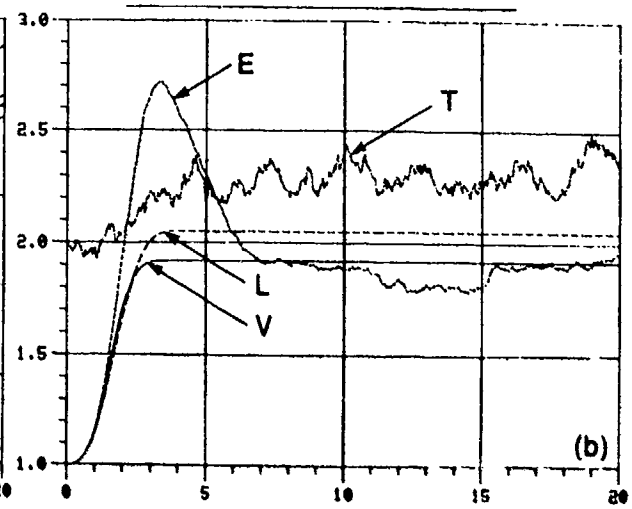
L STOCH. LIN. FILTER

E EXT. KALMAN FILTER

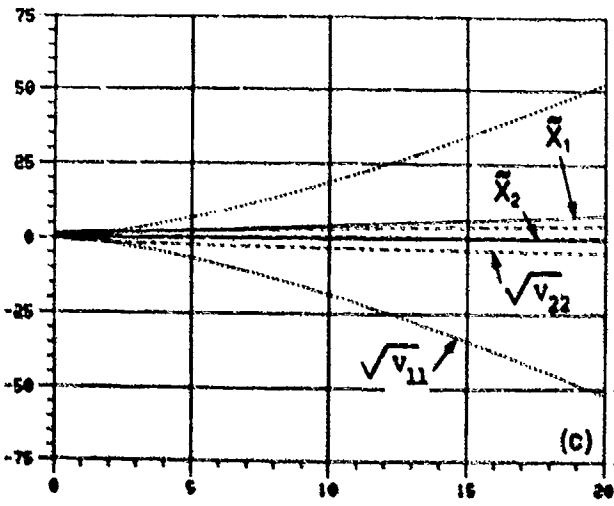
ACTUAL FILTER OUTPUT FOR STATE X_1



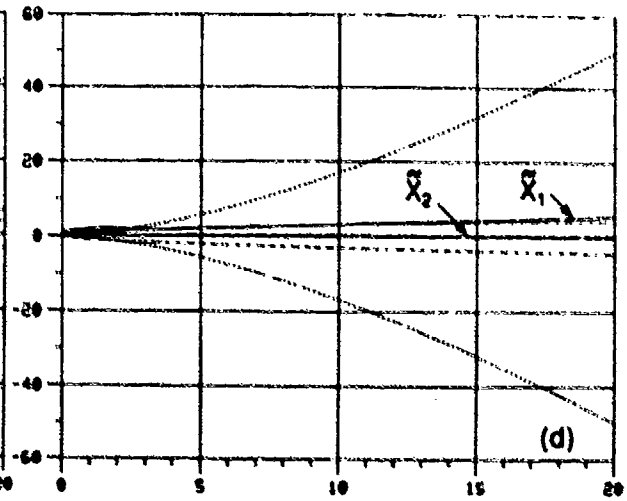
ACTUAL FILTER OUTPUT FOR STATE X_2



ERROR BOUNDS FOR MINIMUM VARIANCE FILTER



ERROR BOUNDS FOR STOCH. LIN. FILTER



ERROR BOUNDS FOR EKF

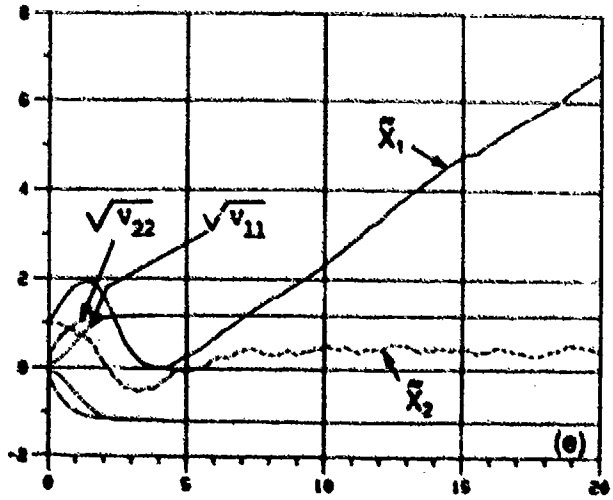


FIGURE 2.6 MVG OUTPUTS OF PHASE-LOCK LOOP SIMULATION

$V_{11}(0) = V_{22}(0) = 0.01$ $\quad \hat{X}_1(0) = \hat{X}_2(0) = 2.0$

$V_{12}(0) = 0.0$ $\quad \Psi_v = \Psi_w = 1.0$

$X_1(0) = X_2(0) = 2.0$

T TRUE STATE

V MINIMUM VARIANCE FILTER

L STOCHASTIC LINEARIZATION FILTER

E EXTENDED KALMAN FILTER

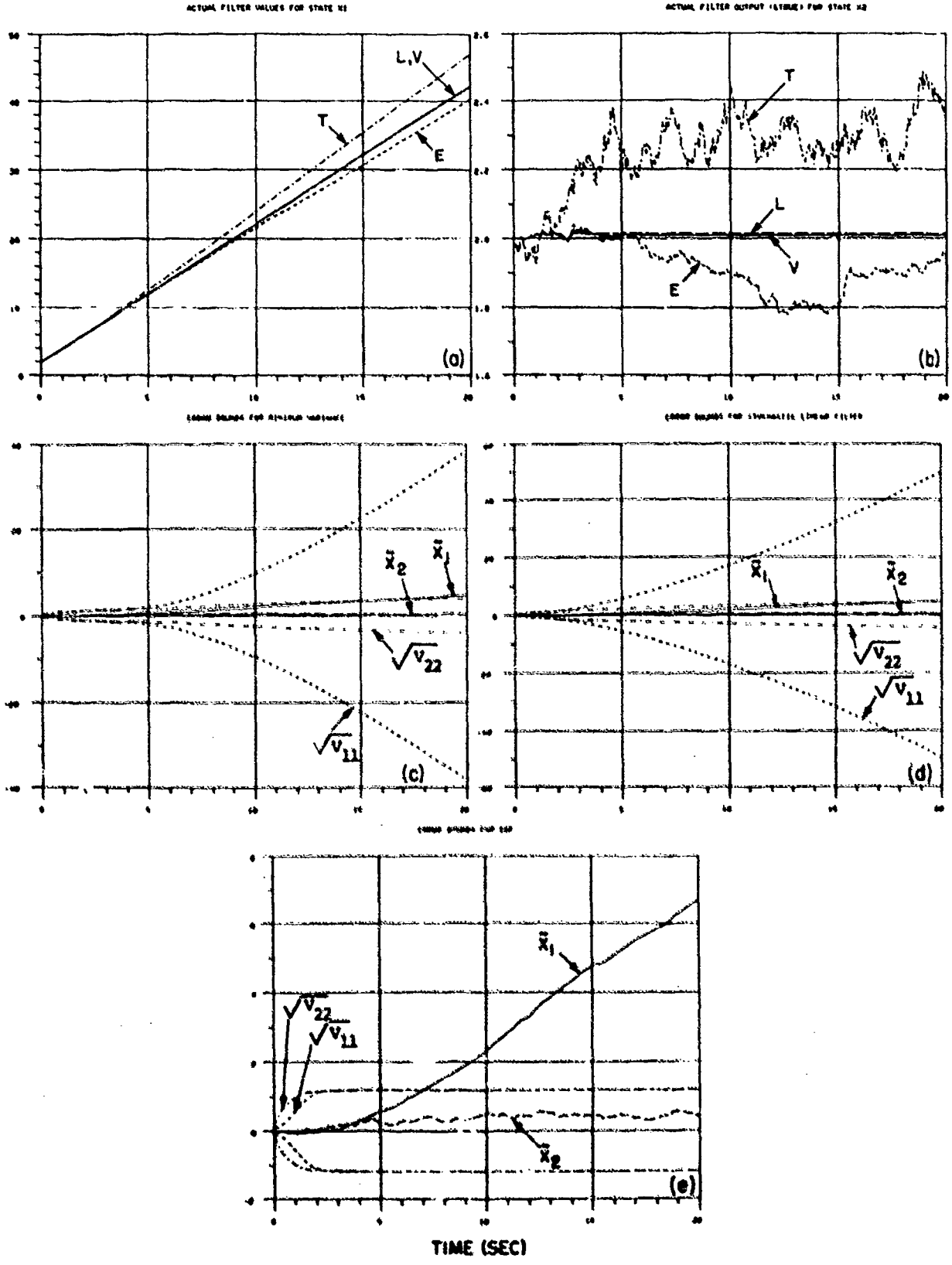


FIGURE 2.7 PHASE-LOOK LOOP SIMULATION; OUTPUTS OF MIN. VAR. FILTER (5-RUN AVERAGES)

$V_{11}(0) = V_{22}(0) = V_{12}(0) = 1.0$

$X_1(0) = X_2(0) = \hat{X}_1(0) = 2.0$

$\Psi_V = \Psi_W = 0.001$

T TRUE STATE

V MINIMUM VARAINCE FILTER

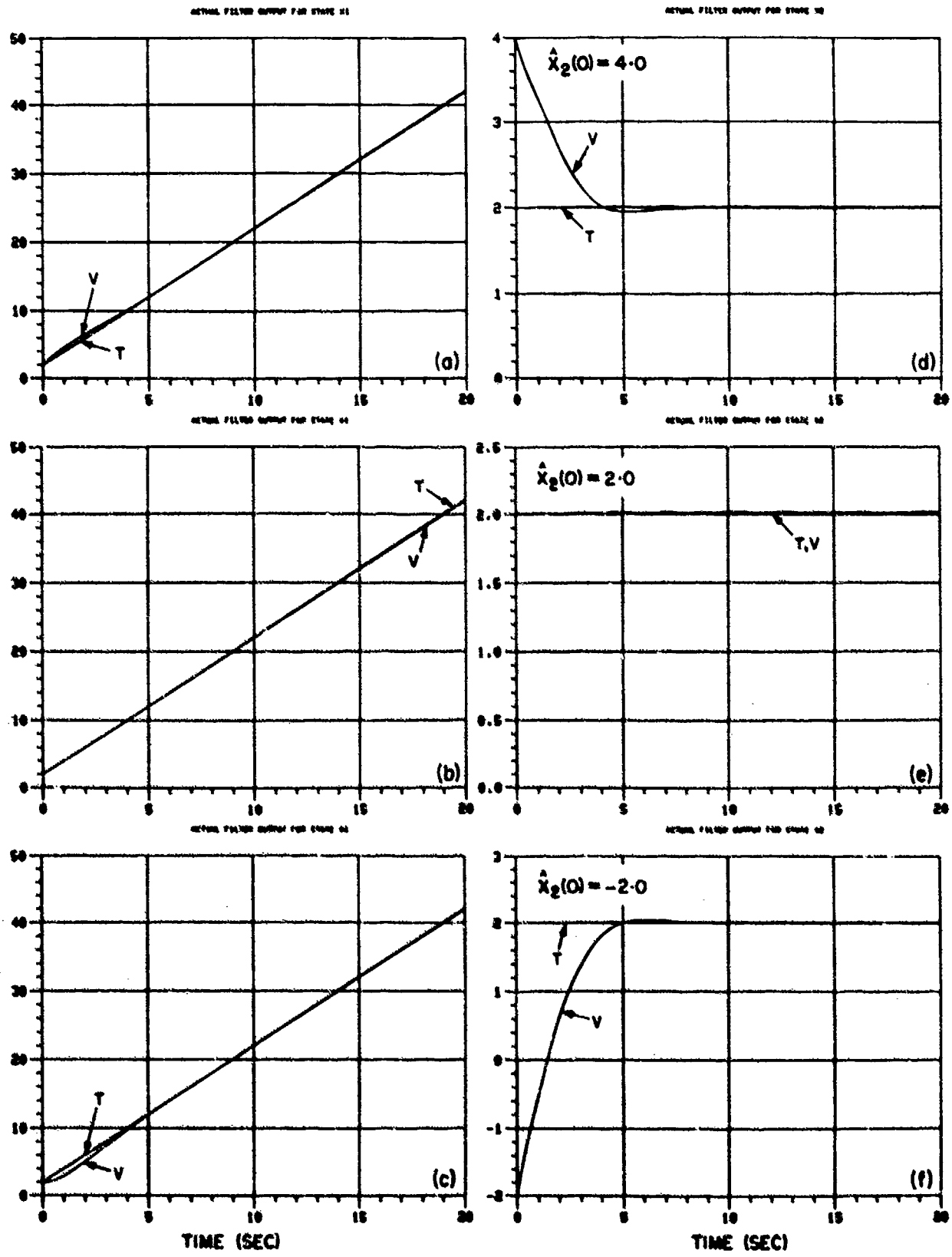
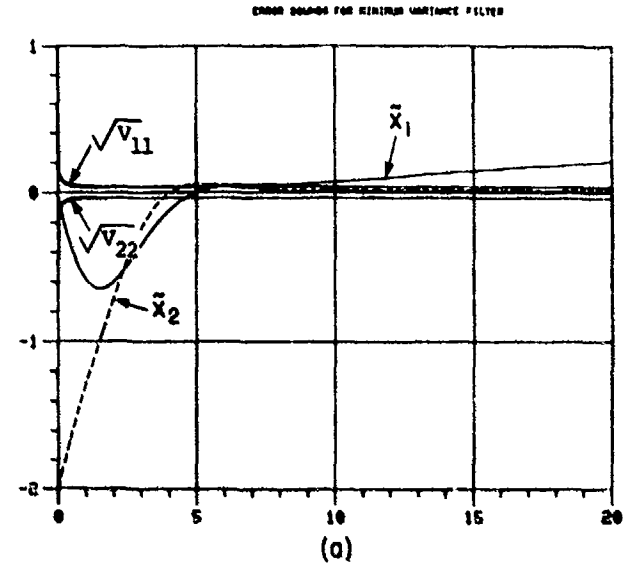


FIGURE 2.8 ERROR BOUNDS FOR MINIMUM VARIANCE FILTER

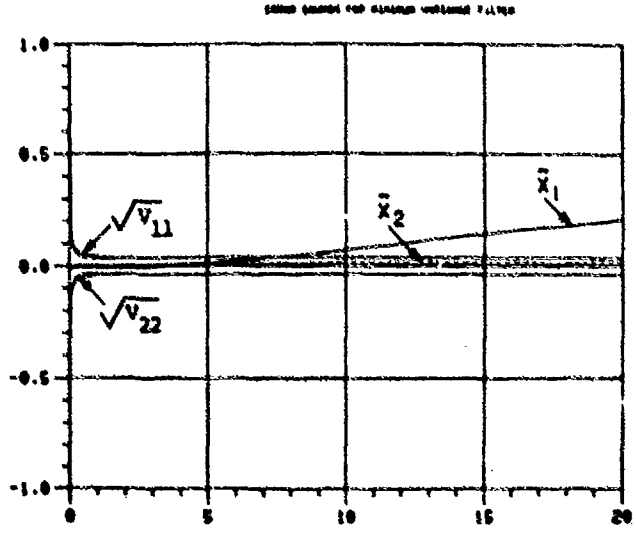
$V_{11}(0) = V_{22}(0) = V_{12}(0) = 1.0$

$X_1(0) = X_2(0) = \hat{X}_1(0) = 2.0$

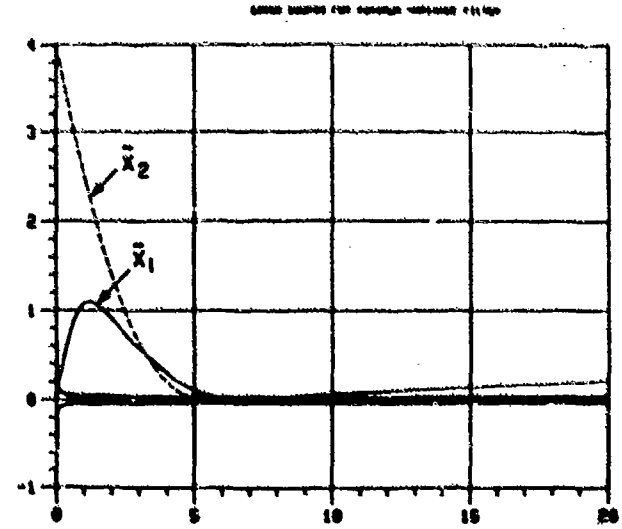
$\Psi_V = \Psi_W = 0.001$



(a) $\hat{X}_2(0) = 4.0$



(b) $\hat{X}_2(0) = 2.0$



TIME (SEC)

(c) $\hat{X}_2(0) = -2.0$

FIGURE 2.9: VAN DER POL'S OSCILLATOR; EFFECTS OF Ψ ON FILTER MEASUREMENT INPUT

$X_1(0) = 2.0$

$X_2(0) = 0.0$

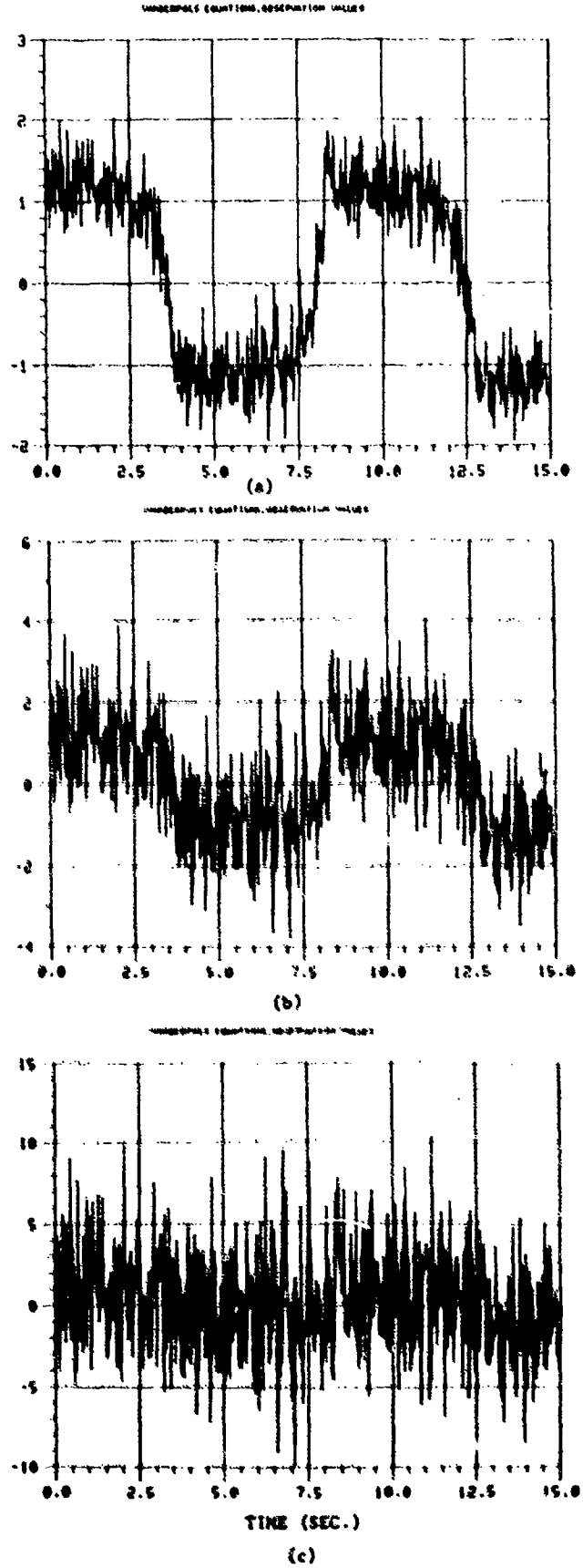


FIGURE 2.10: VAN DER POL'S OSCILLATOR; EFFECTS OF Ψ WITH VERY SMALL V

$X_1(0) = \hat{X}_1(0) = 2.0$

$X_2(0) = \hat{X}_2(0) = 0.0$

$V_{11}(0) = V_{22}(0) = 0.01$

$V_{12}(0) = 0$

— — — — TRUE STATE

———— MINIMUM VARIANCE FILTER

---------- STOCHASTIC LINEARIZATION FILTER

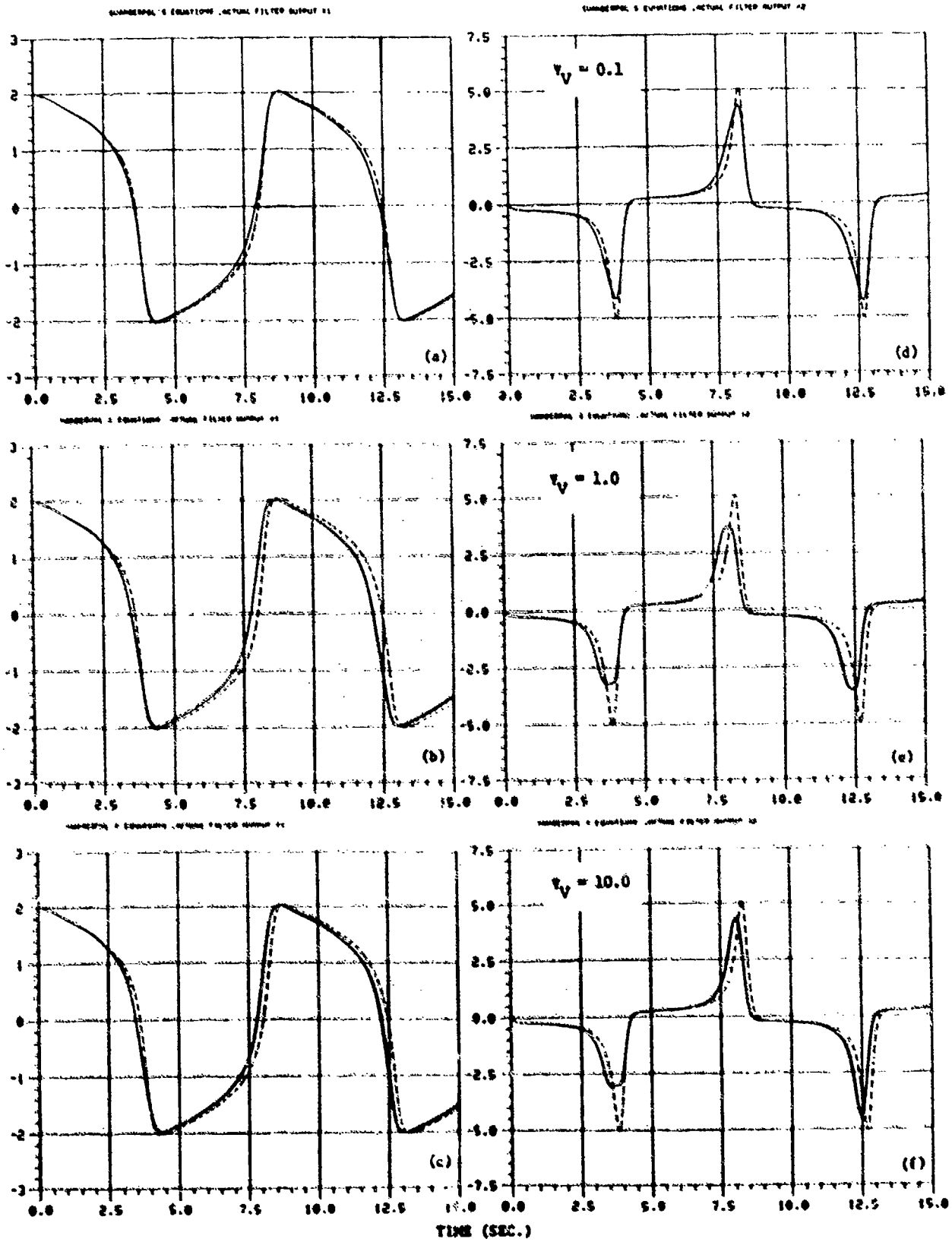


FIGURE 2.11: VAN DER POL'S OSCILLATOR: EFFECTS OF Ψ WITH SMALL V

$\hat{X}_1(0) = X_1(0) = 2.0$

$\hat{X}_2(0) = X_2(0) = 0.0$

$V_{11}(0) = V_{22}(0) = 0.5$

$V_{12}(0) = 0.0$

— — — — TRUE STATE

———— MINIMUM VAR. FILTER

---------- STOCHASTIC LINEARIZATION FILTER

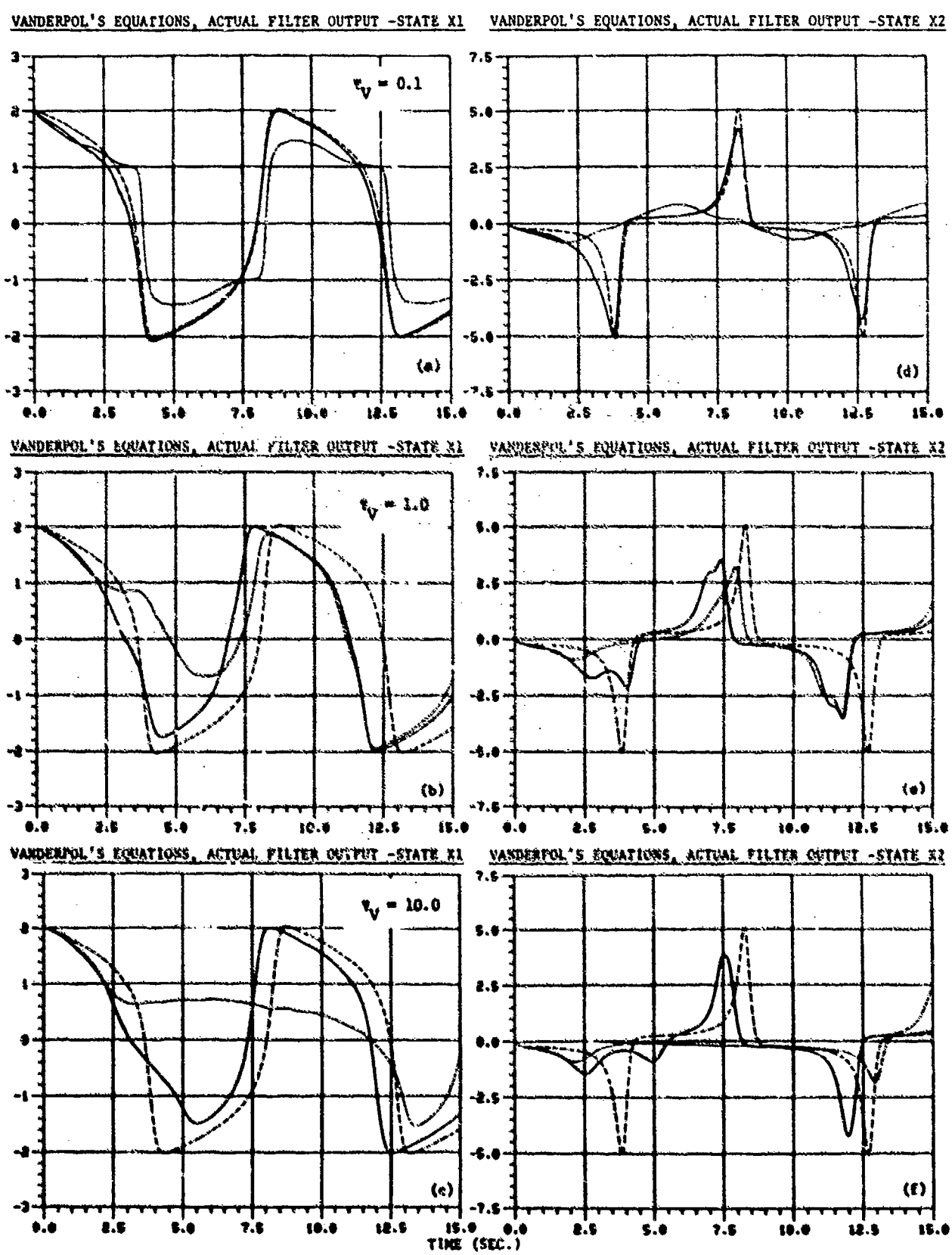


FIGURE 2.12: VAN DER POL'S OSCILLATOR; EFFECTS OF V WITH LARGER V

$\hat{X}_1(0) = X_1(0) = 2.0$

$\hat{X}_2(0) = X_2(0) = 0.0$

$V_{11}(0) = V_{22}(0) = 5.0$

$V_{12}(0) = 0.0$

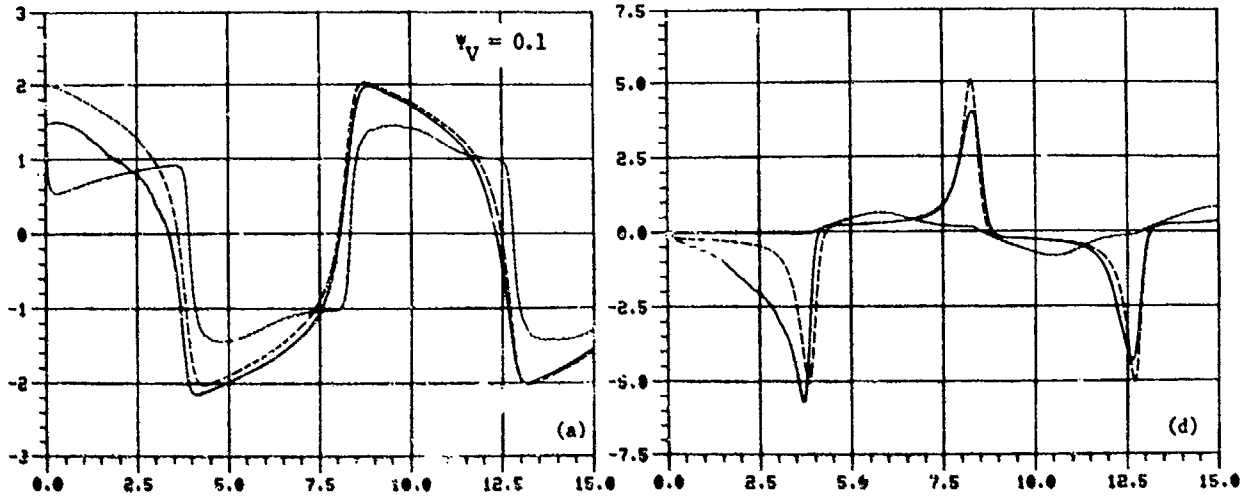
— — — — TRUE STATE

—— —— MINIMUM VAR. FILTER

----- ----- STOCHASTIC LINEARIZATION FILTER

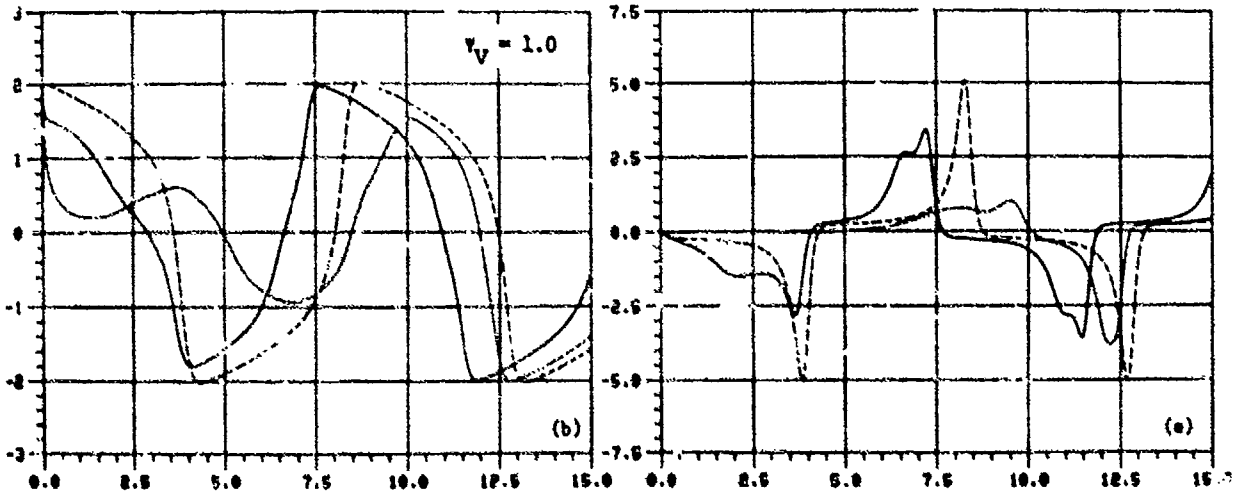
VANDERPOL'S EQUATIONS, ACTUAL FILTER OUTPUT FOR STATE X1

VANDERPOL'S EQUATIONS, ACTUAL FILTER OUTPUT FOR STATE X2



VANDERPOL'S EQUATIONS, ACTUAL FILTER OUTPUT -STATE X1

VANDERPOL'S EQUATIONS, ACTUAL FILTER OUTPUT -STATE X2



VANDERPOL'S EQUATIONS, ACTUAL FILTER OUTPUT -STATE X1

VANDERPOL'S EQUATIONS, ACTUAL FILTER OUTPUT -STATE X2

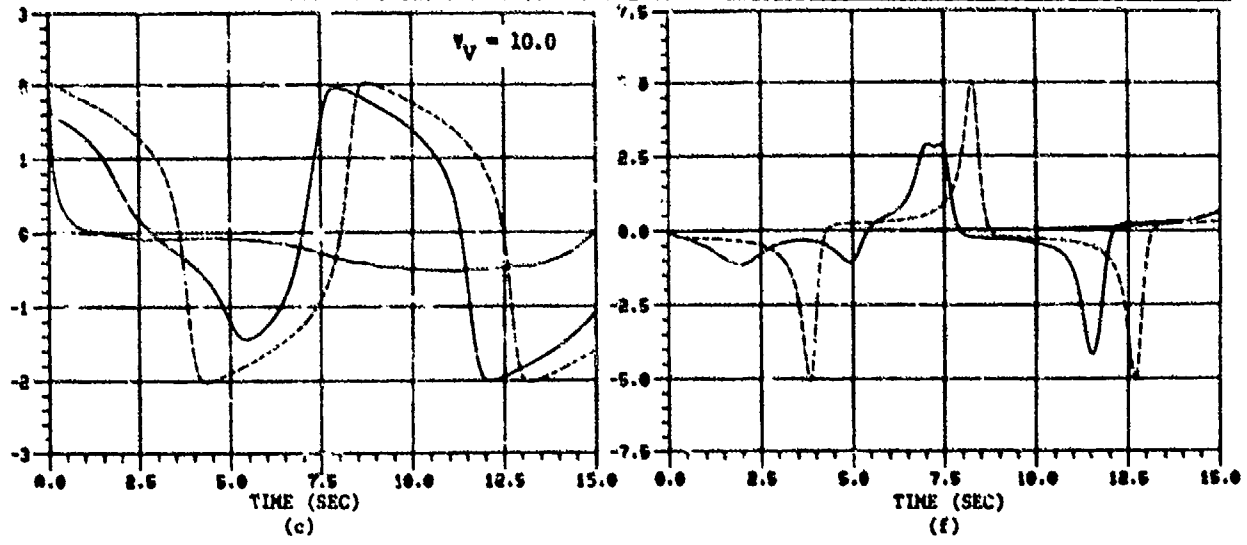


FIGURE 2.13: OUTPUTS OF VAN DER POL'S OSCILLATOR WITH "REASONABLE" INITIAL ESTIMATES AND Ψ

$\hat{X}_1(0) = X_1(0) = 2.0$

$\hat{X}_2(0) = X_2(0) = 0.0$

$V_{11}(0) = V_{22}(0) = 0.5$

$V_{12}(0) = 0.0$

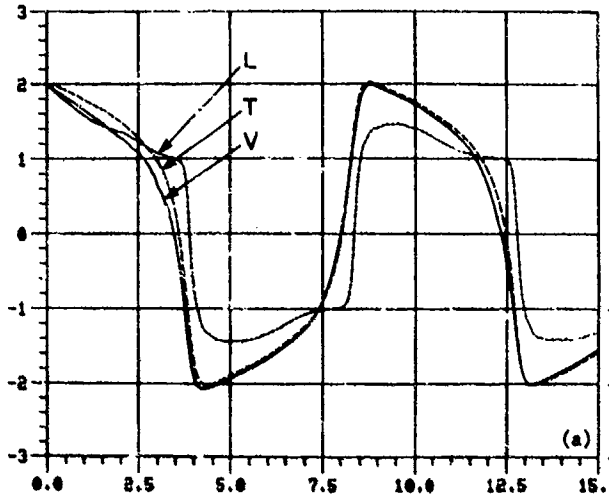
T TRUE STATE

V MINIMUM VARIANCE FILTER

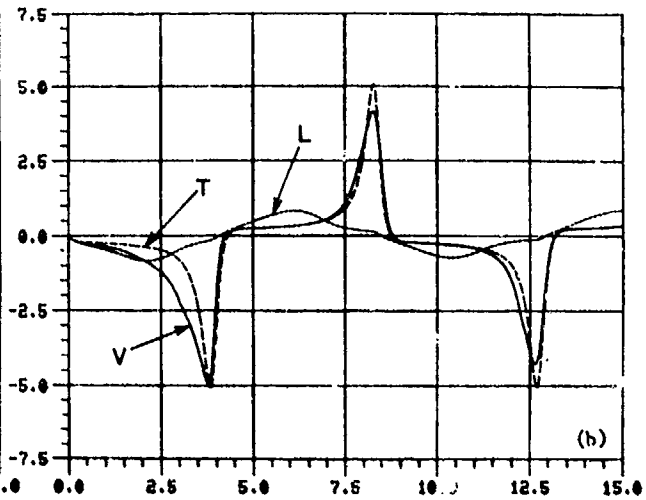
L STOCHASTIC LINEARIZATION FILTER

$\Psi_v = 0.1$

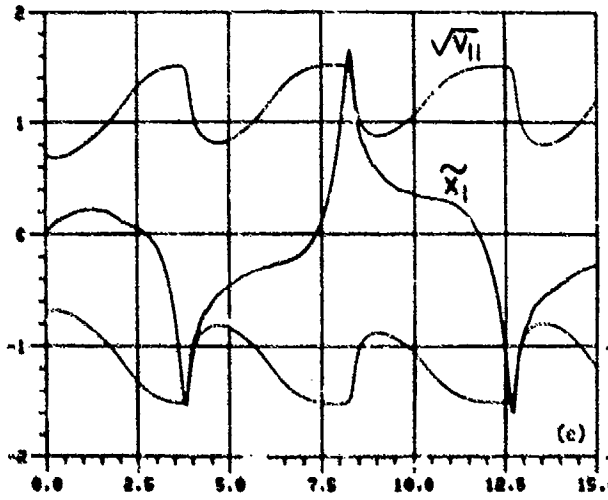
VANDERPOL'S EQUATIONS, ACTUAL FILTER OUTPUT -STATE X1



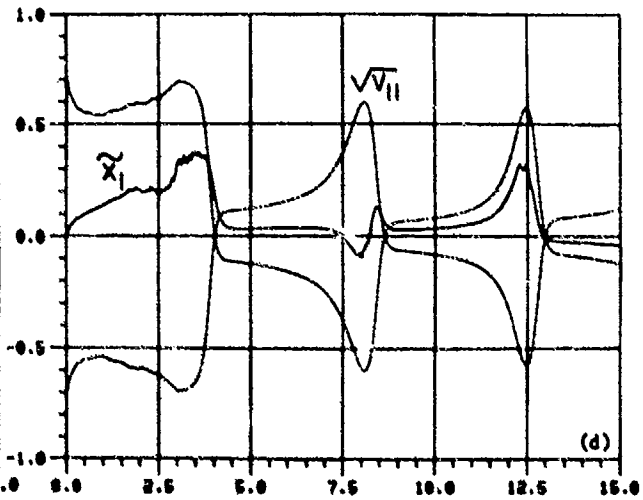
VANDERPOL'S EQUATIONS, ACTUAL FILTER -STATE X2



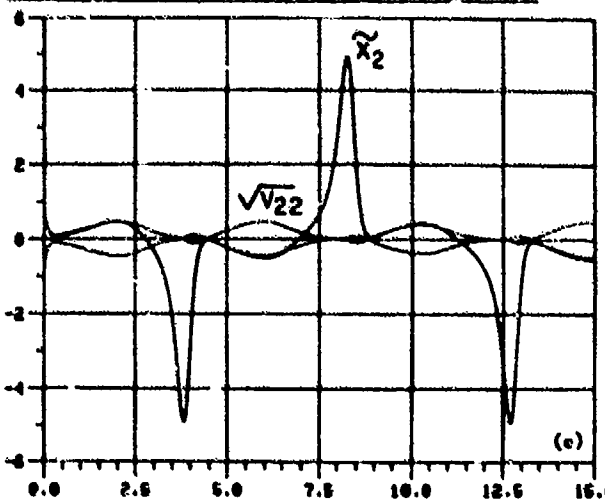
VANDERPOL'S EQUATIONS, STOCHASTIC LIN. FILTER, ERROR BOUNDS -STATE X1



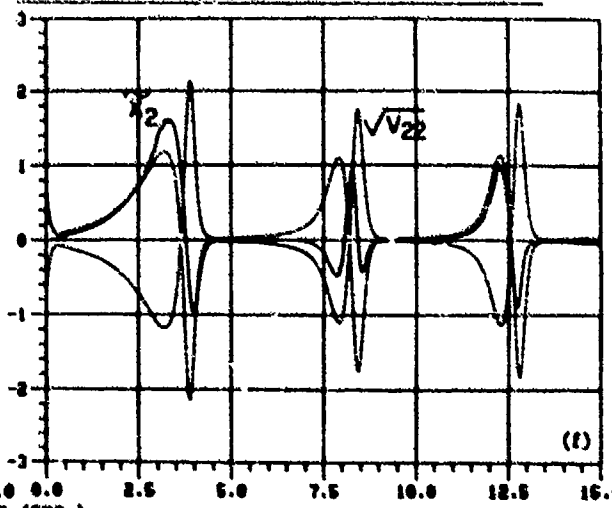
VANDERPOL'S EQUATIONS, MINIMUM VARIANCE FILTER ERROR BOUNDS -STATE X1



VANDERPOL'S EQUATIONS, STOCHASTIC LINEARIZATION FILTER, ERROR BOUNDS -STATE X2



VANDERPOL'S EQUATIONS, MINIMUM VARIANCE FILTER ERROR BOUNDS -STATE X2



TIME (SEC.)

FIGURE 2.14: VAN DER POL'S OSCILLATOR; EFFECTS OF INITIAL STATE AND ERROR-VARIANCES

$X_1(0) = 2$, $X_2(0) = 0$

$V_{12}(0) = 0$, $\Psi_v = 0.1$

— — — — — — TRUE STATE

———————— MINIMUM VARIANCE FILTER

---------- STOCHASTIC LIN. FILTER

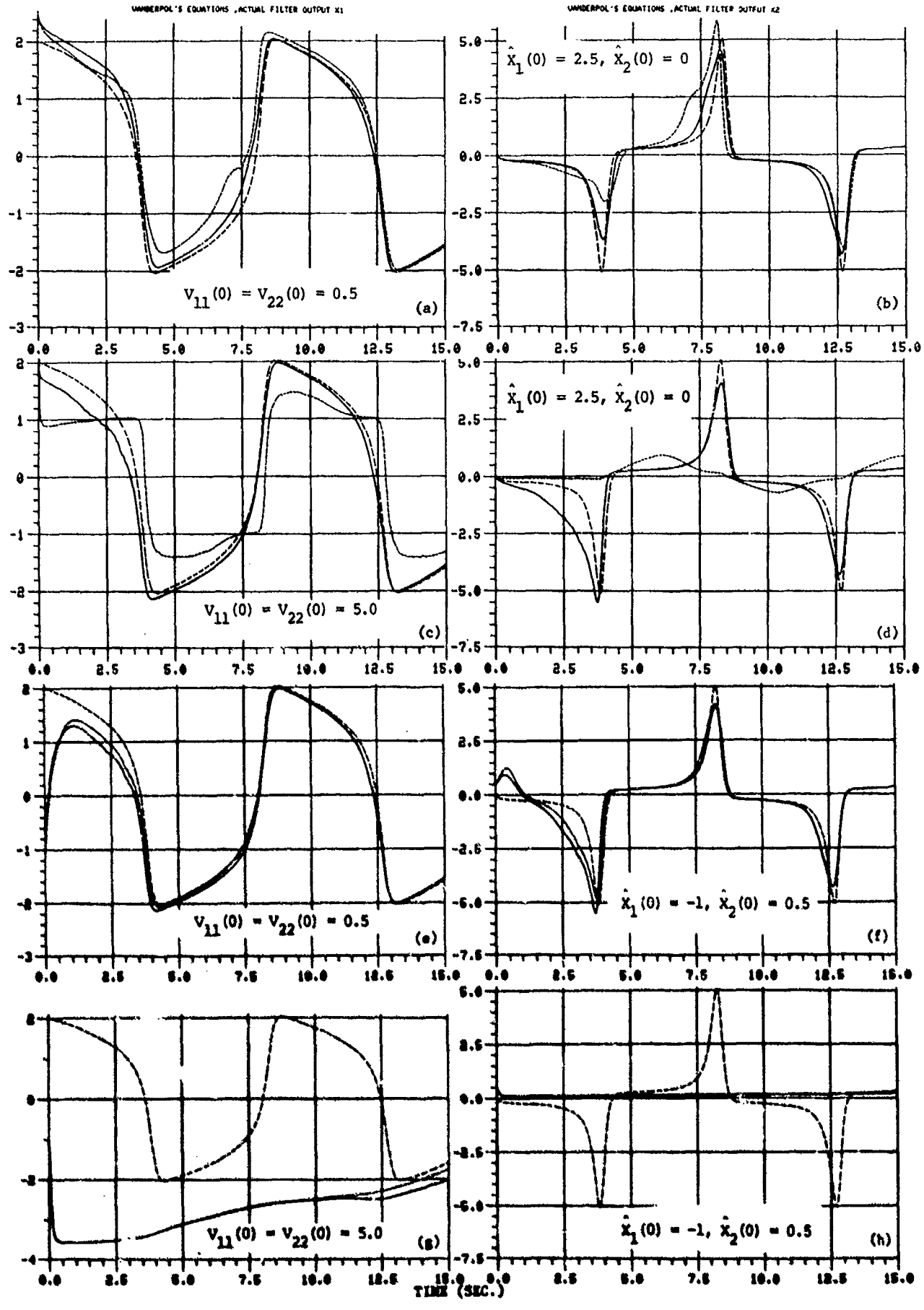


FIGURE 3.1A VG OUTPUTS OF DISCRETE CASE WITH NONLINEAR MEASUREMENT; SMALL Ψ AND V

$\hat{X}_1(0) = \hat{X}_2(0) = 2.0$

$X_1(0) = X_2(0) = 2.0$

$V_{11}(0) = V_{22}(0) = V_{12}(0) = 0.01$

$\Psi_V = \Psi_W = 0.01$

MEASUREMENT MODEL:

$Z = X_1 X_2 + V$

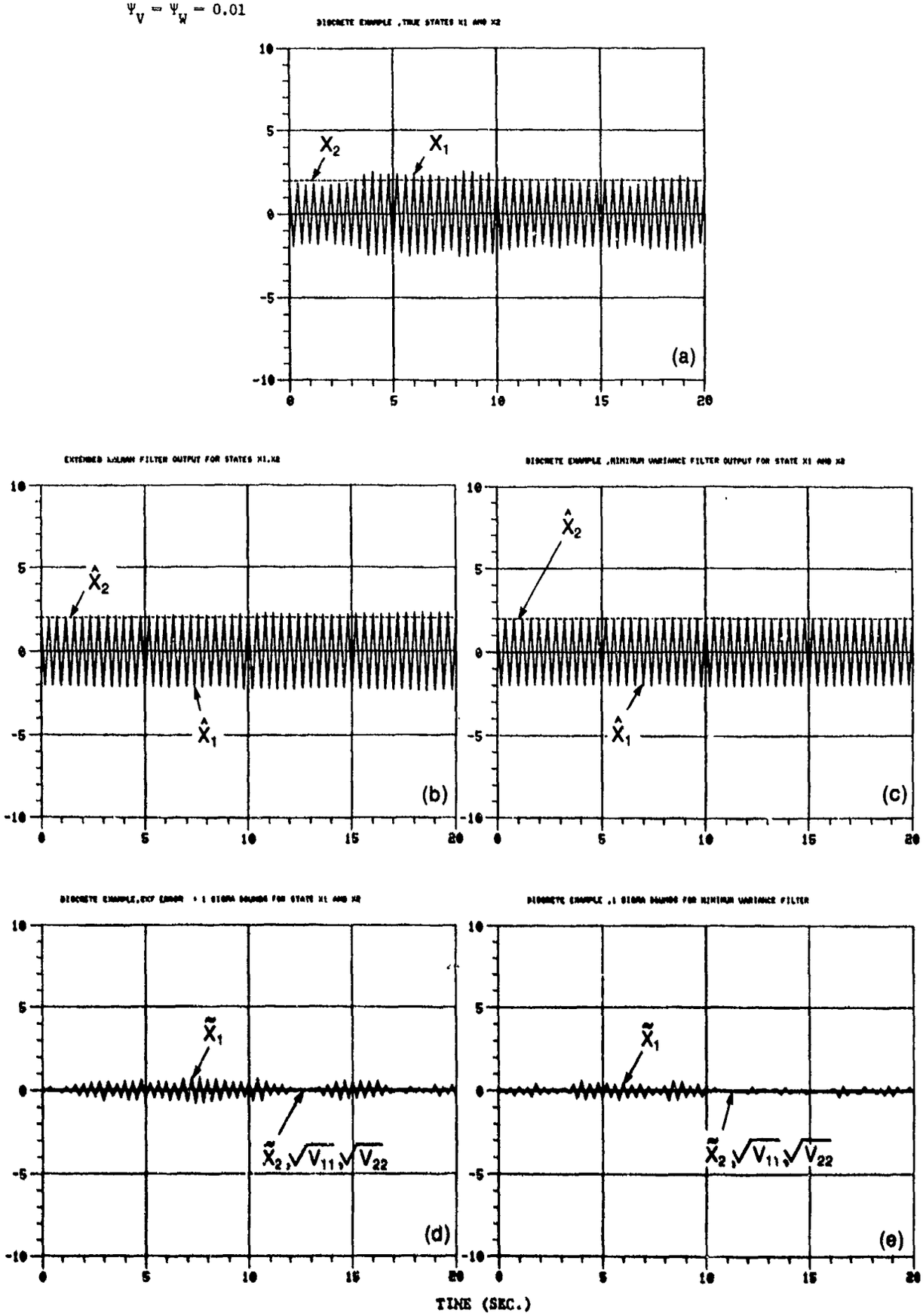


FIGURE 3.2: DISCRETE CASE WITH NONLINEAR MEASUREMENT; EFFECTS OF $\hat{X}(0)$ WITH SMALL Ψ AND V

$\Psi_V = \Psi_W = 0.01$

$X_1(0) = X_2(0) = 2.0$

$V_{11}(0) = V_{22}(0) = V_{12}(0) = 0.01$

MEASUREMENT MODEL: $Z = X_1 X_2 + V$

_____ FILTER ESTIMATE $\hat{X}_1$

---------- FILTER ESTIMATE $\hat{X}_2$

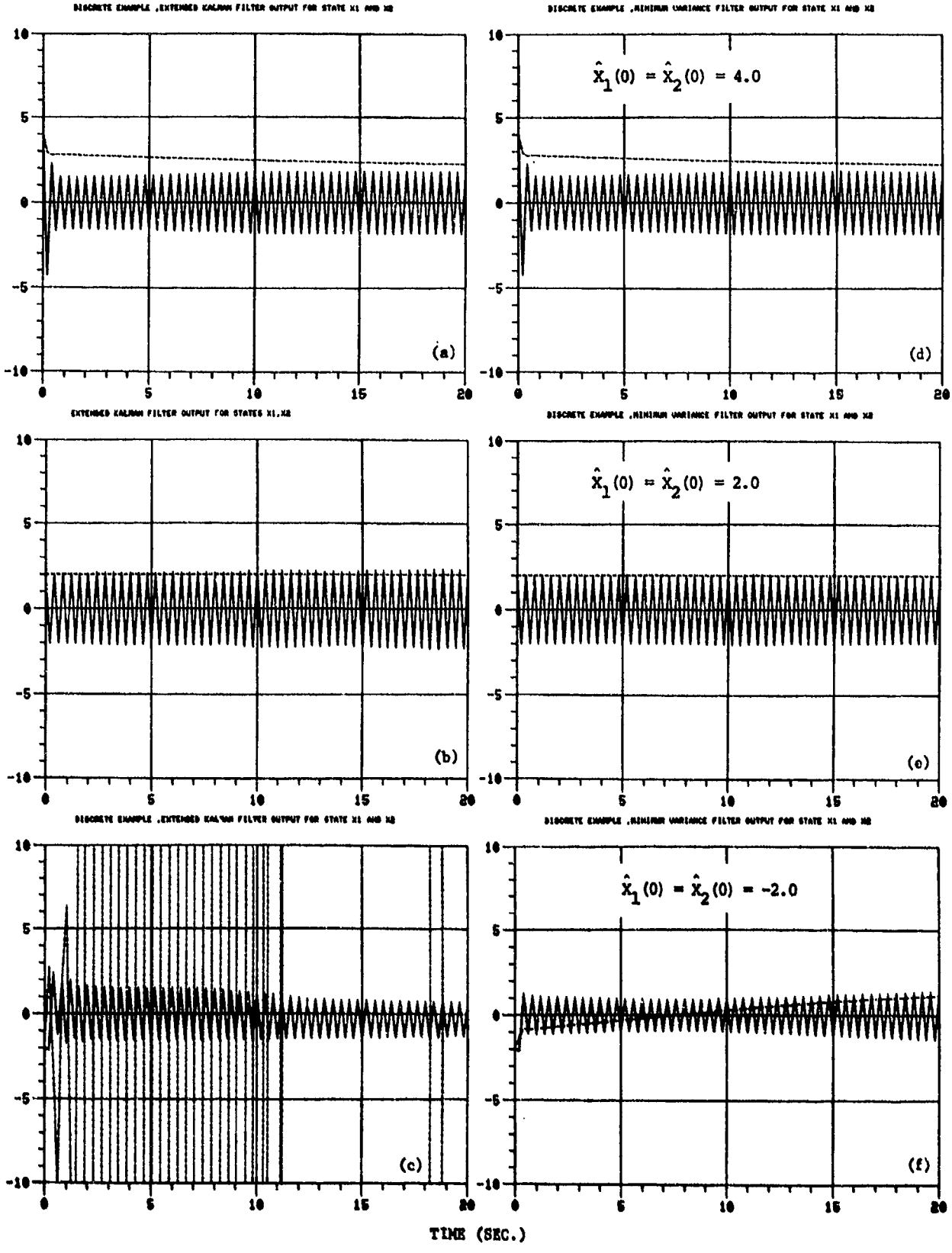


FIGURE 3.3 COMPARISON OF AVG OUTPUTS; EFFECTS OF $\hat{X}(0)$ WITH SMALL Ψ AND LARGER V

$X_1(0) = X_2(0) = 2.0$

$V_{11}(0) = V_{22}(0) = V_{12}(0) = 1.0$

$\Psi_V = \Psi_W = 0.01$

MEASUREMENT MODEL: $Z = X_1 X_2 + V$

———— FILTER ESTIMATE $\hat{X}_1$

- - - - - - - - - - - - FILTER ESTIMATE $\hat{X}_2$

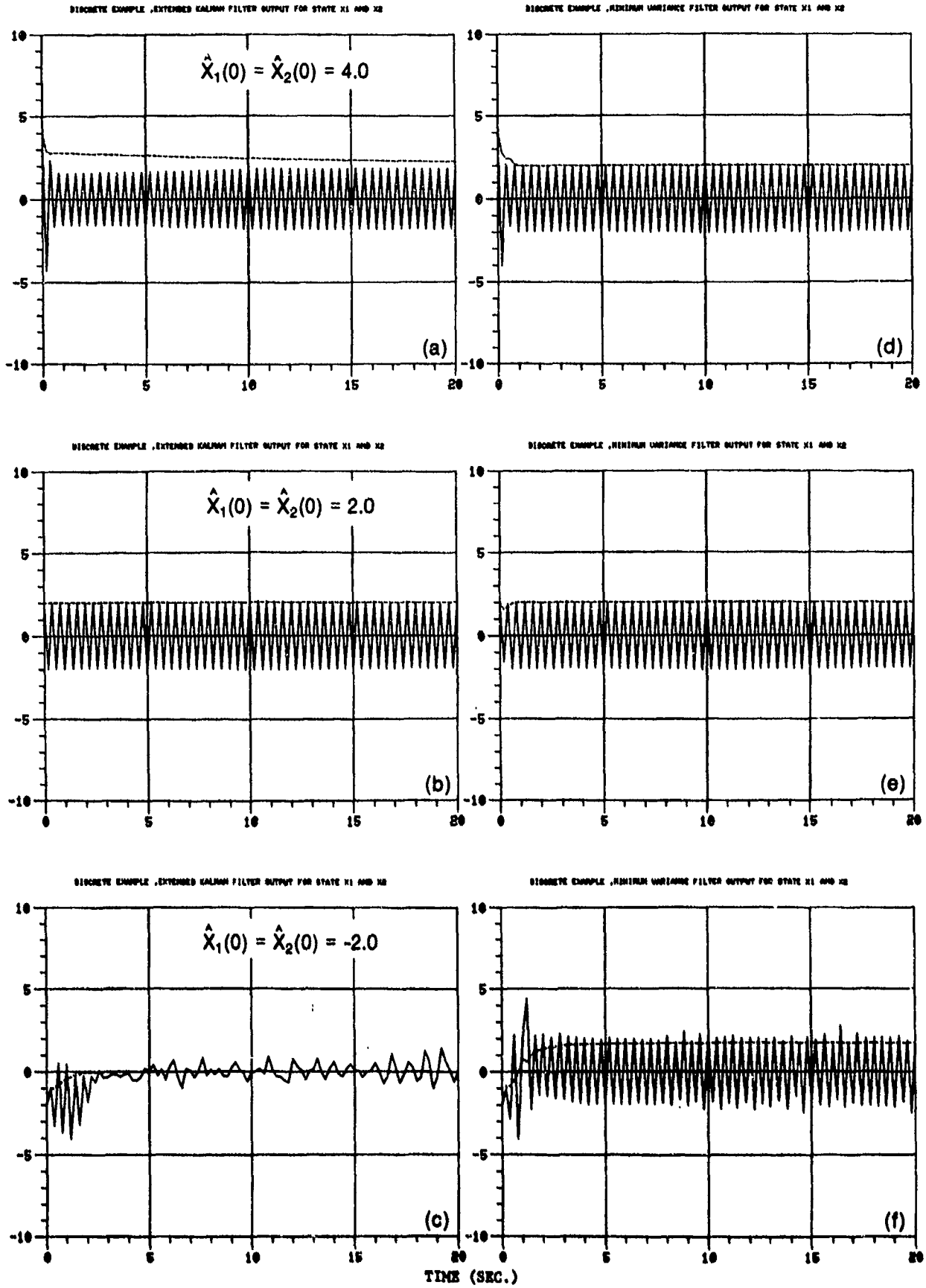


FIGURE 3.4: COMPARISON OF ERROR BOUNDS; EFFECTS OF SMALL Ψ AND LARGER V

$X_1(0) = X_2(0) = 2.0$

$V_{11}(0) = V_{22}(0) = V_{12}(0) = 1.0$

$\Psi_V = \Psi_W = 0.01$

MEASUREMENT MODEL: $Z = X_1 X_2 + V$

——— $\tilde{X}_1$ ---------- $\sqrt{V_{11}}$

— — — — $\tilde{X}_2$ — · — · — $\sqrt{V_{22}}$

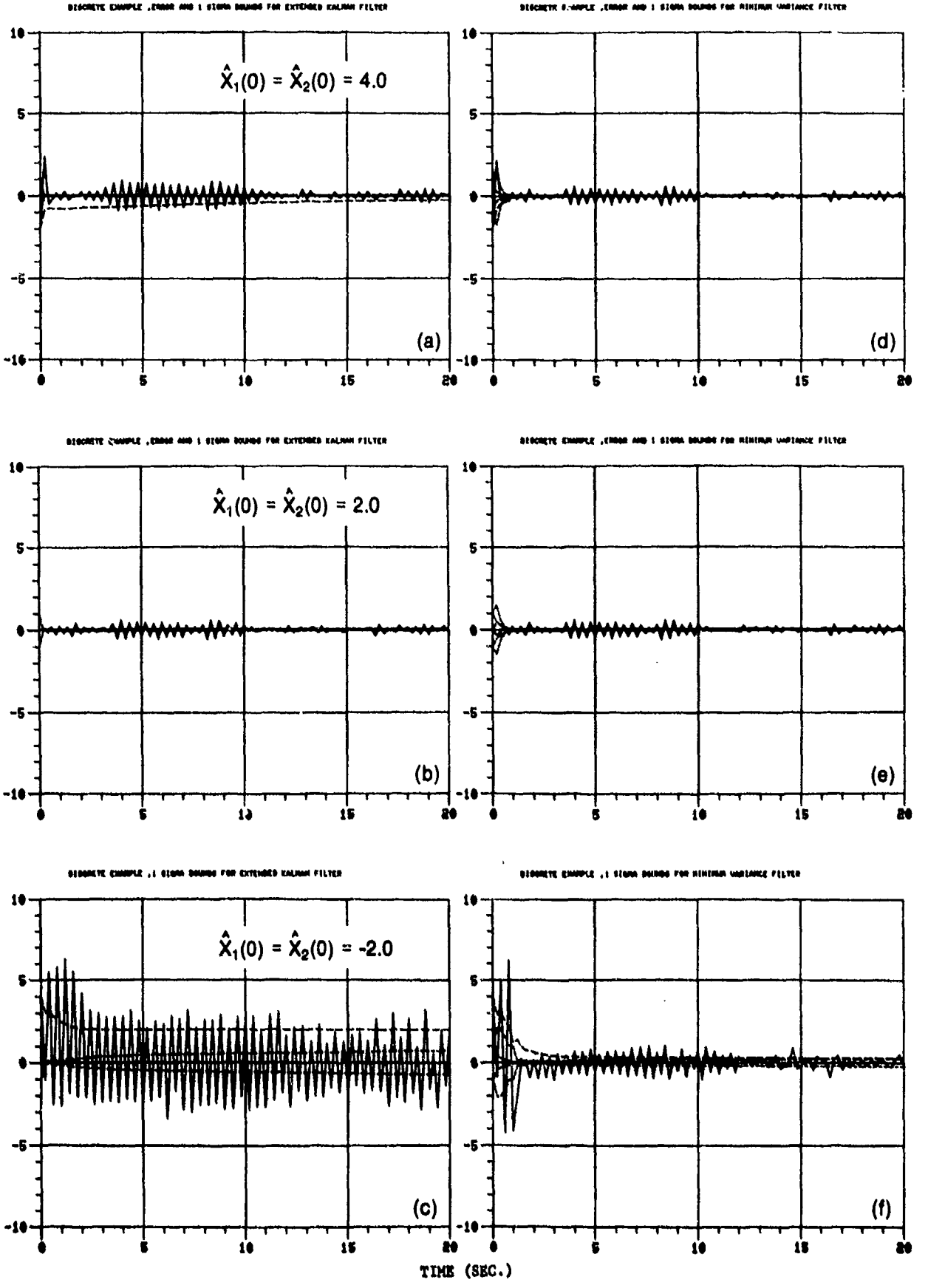


FIGURE 3.5: COMPARISON OF AVG OUTPUTS; EFFECTS OF INITIAL V WITH SMALL Ψ

$\Psi_V = \Psi_W = 0.01$

$X_1(0) = X_2(0) = 2.0$

$\hat{X}_1(0) = \hat{X}_2(0) = 4.0$

MEASUREMENT MODEL: $Z = X_1 X_2 + V$

——— FILTER ESTIMATE $\hat{X}_1$

- - - - - - - - - - FILTER ESTIMATE $\hat{X}_2$

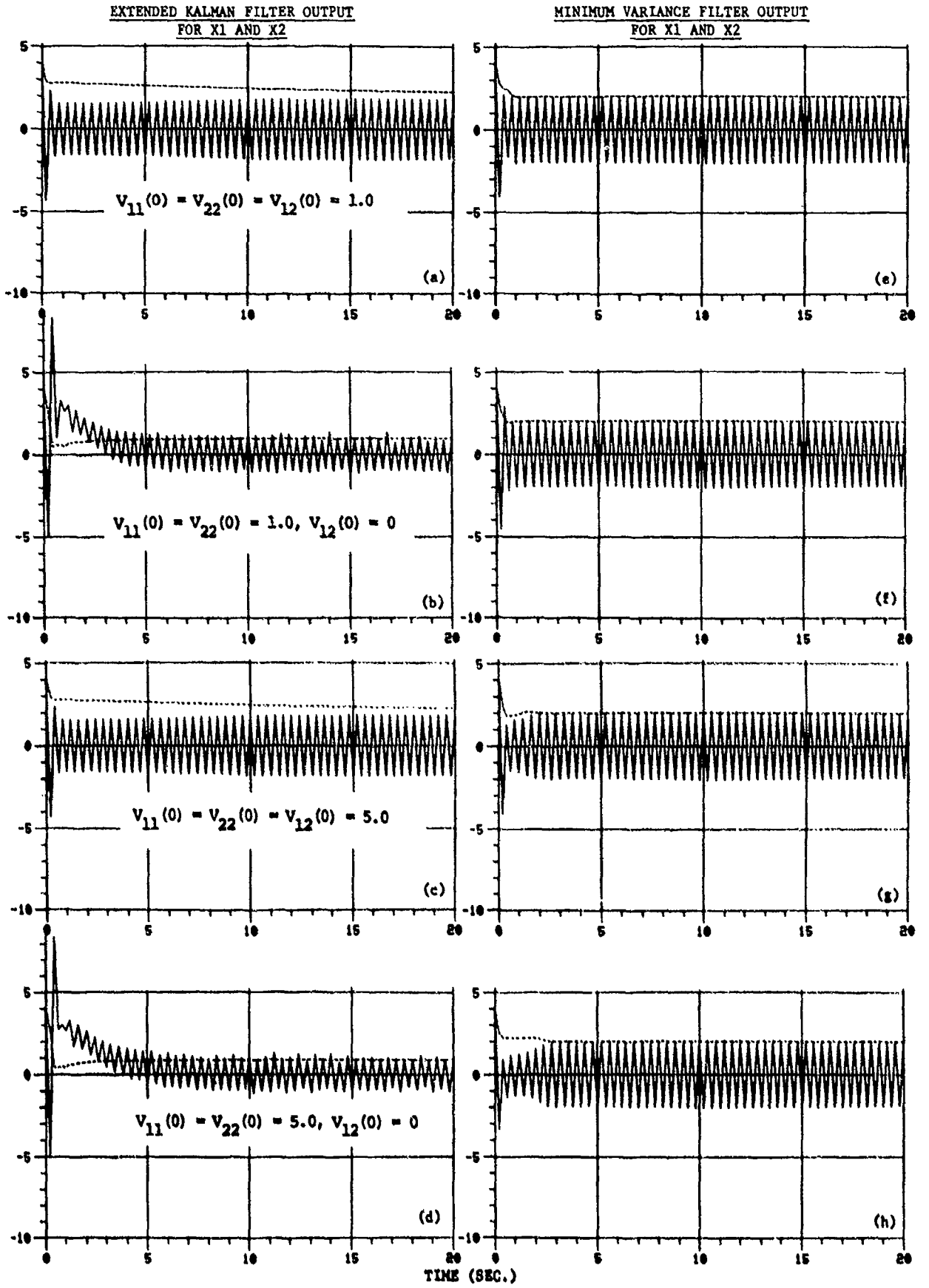


FIGURE 3.6: COMPARISON OF EKF OUTPUTS; EFFECTS OF INITIAL V WITH LARGER Ψ

$\Psi_V = \Psi_W = 1.0$

$X_1(0) = X_2(0) = 2.0$

$\hat{X}_1(0) = \hat{X}_2(0) = 4.0$

MEASUREMENT MODEL: $Z = X_1 X_2 + V$

———— FILTER ESTIMATE $\hat{X}_1$

---------- FILTER ESTIMATE $\hat{X}_2$

EXTENDED KALMAN FILTER OUTPUT FOR X1 AND X2

MINIMUM VARIANCE FILTER OUTPUT FOR X1 AND X2

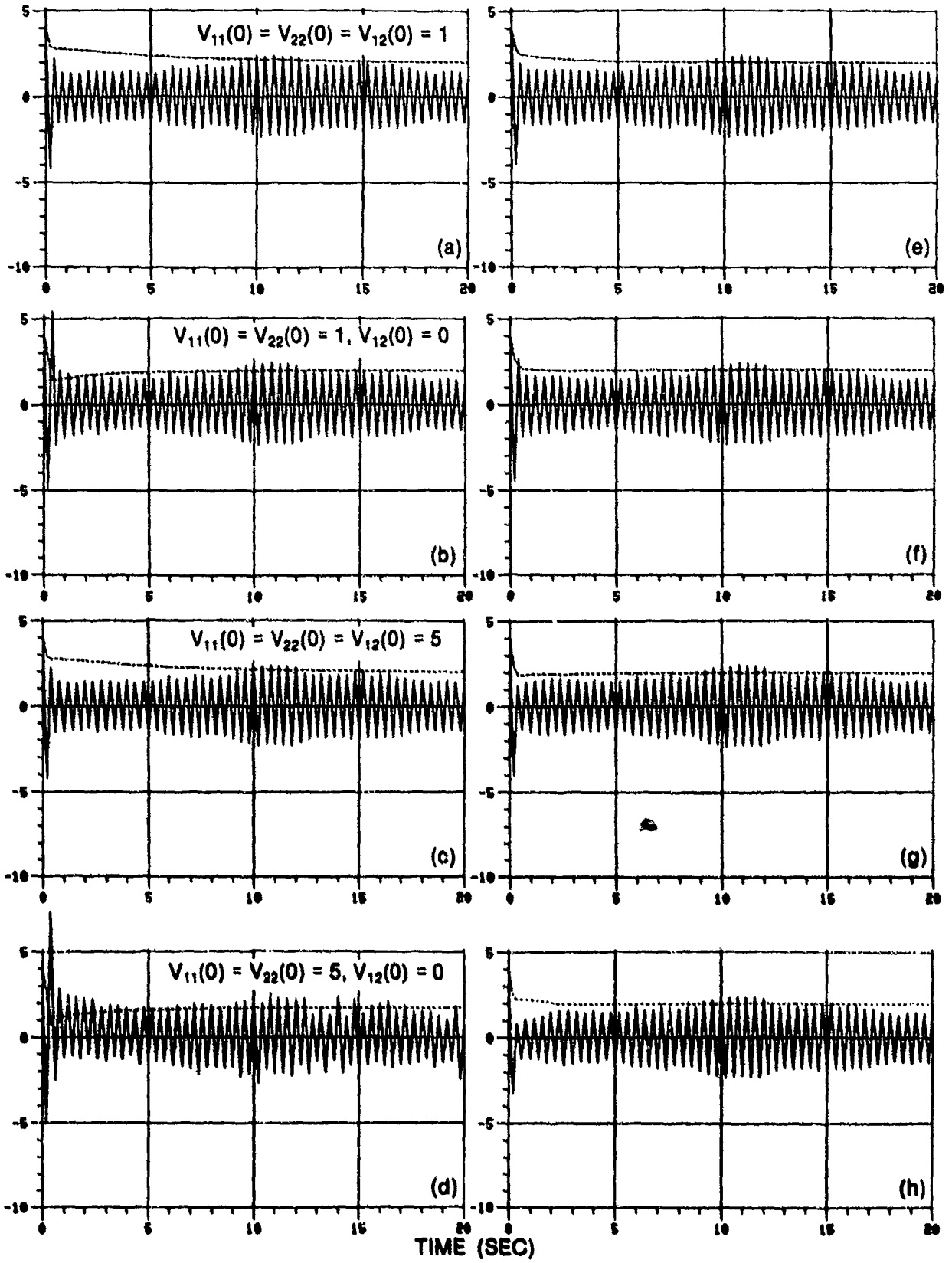


FIGURE 3.7: DISCRETE CASE WITH LINEAR MEASUREMENT; EFFECTS OF $\hat{X}(0)$ WITH SMALL Ψ AND LARGE V

$V_{11}(0) = V_{22}(0) = V_{12}(0) = 1.0$

$X_1(0) = X_2(0) = 2.0$

$\Psi_V = \Psi_W = 0.01$

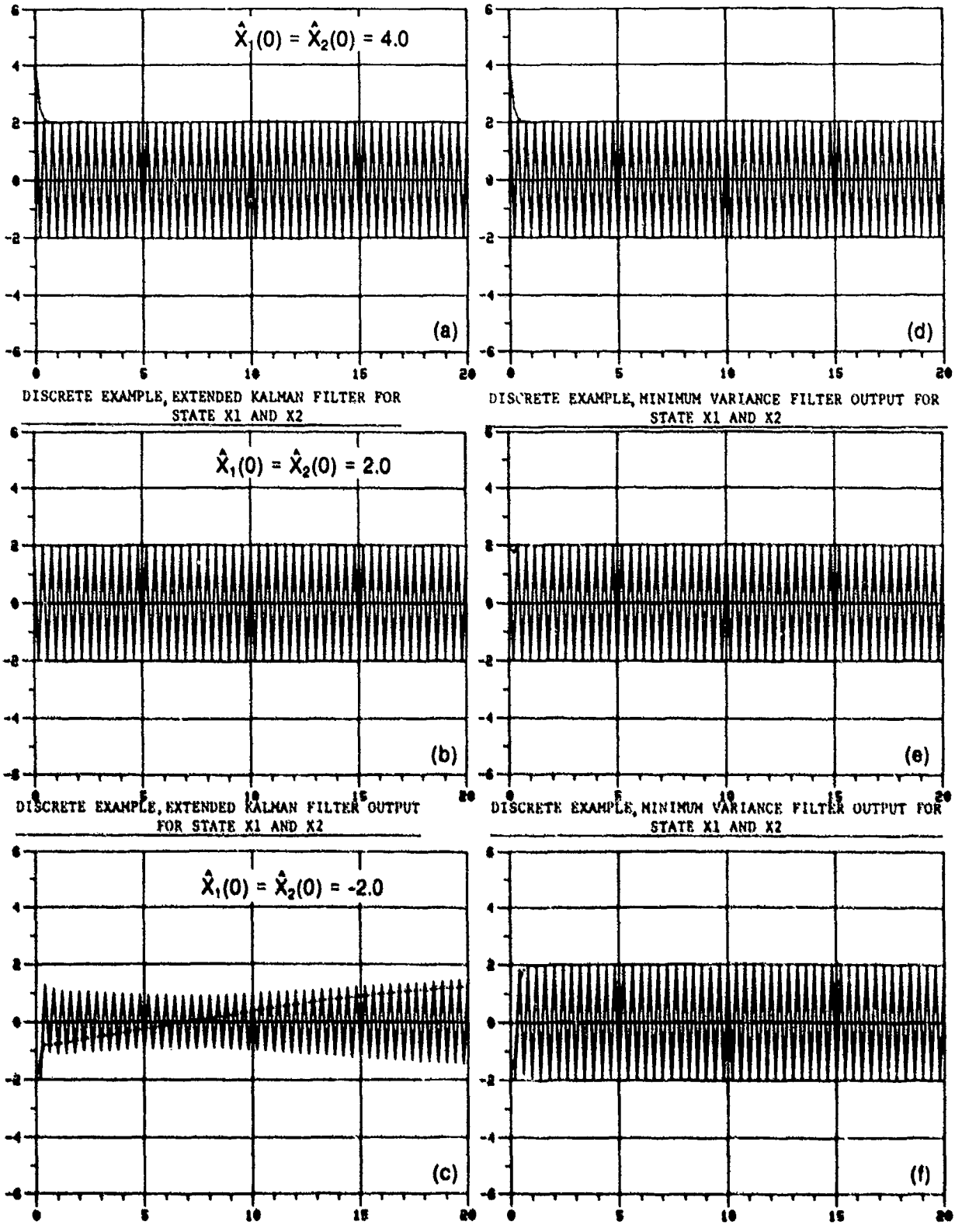
MEASUREMENT MODEL: $Z = X_1 + V$

———— FILTER ESTIMATE $\hat{X}_1$

---------- FILTER ESTIMATE $\hat{X}_2$

DISCRETE EXAMPLE, EXTENDED KALMAN FILTER OUTPUT FOR STATE X1 AND X2

DISCRETE EXAMPLE, MINIMUM VARIANCE FILTER OUTPUT FOR STATE X1 AND X2



DISCRETE EXAMPLE, EXTENDED KALMAN FILTER FOR STATE X1 AND X2

DISCRETE EXAMPLE, MINIMUM VARIANCE FILTER OUTPUT FOR STATE X1 AND X2

DISCRETE EXAMPLE, EXTENDED KALMAN FILTER OUTPUT FOR STATE X1 AND X2

DISCRETE EXAMPLE, MINIMUM VARIANCE FILTER OUTPUT FOR STATE X1 AND X2

FIGURE 3.8: DISCRETE CASE WITH LINEAR MEASUREMENT; EFFECTS OF $\hat{X}(0)$ WITH SMALL Ψ AND LARGE V

$V_{11}(0) = V_{22}(0) = V_{12}(0) = 1.0$

$X_1(0) = X_2(0) = 2.0$

$\Psi_V = \Psi_W = 0.01$

MEASUREMENT MODEL: $Z = X_1 + V$

———— $\tilde{X}_1$ ---------- $\sqrt{V_{11}}$

— — — — $\tilde{X}_2$ — · — · — $\sqrt{V_{22}}$

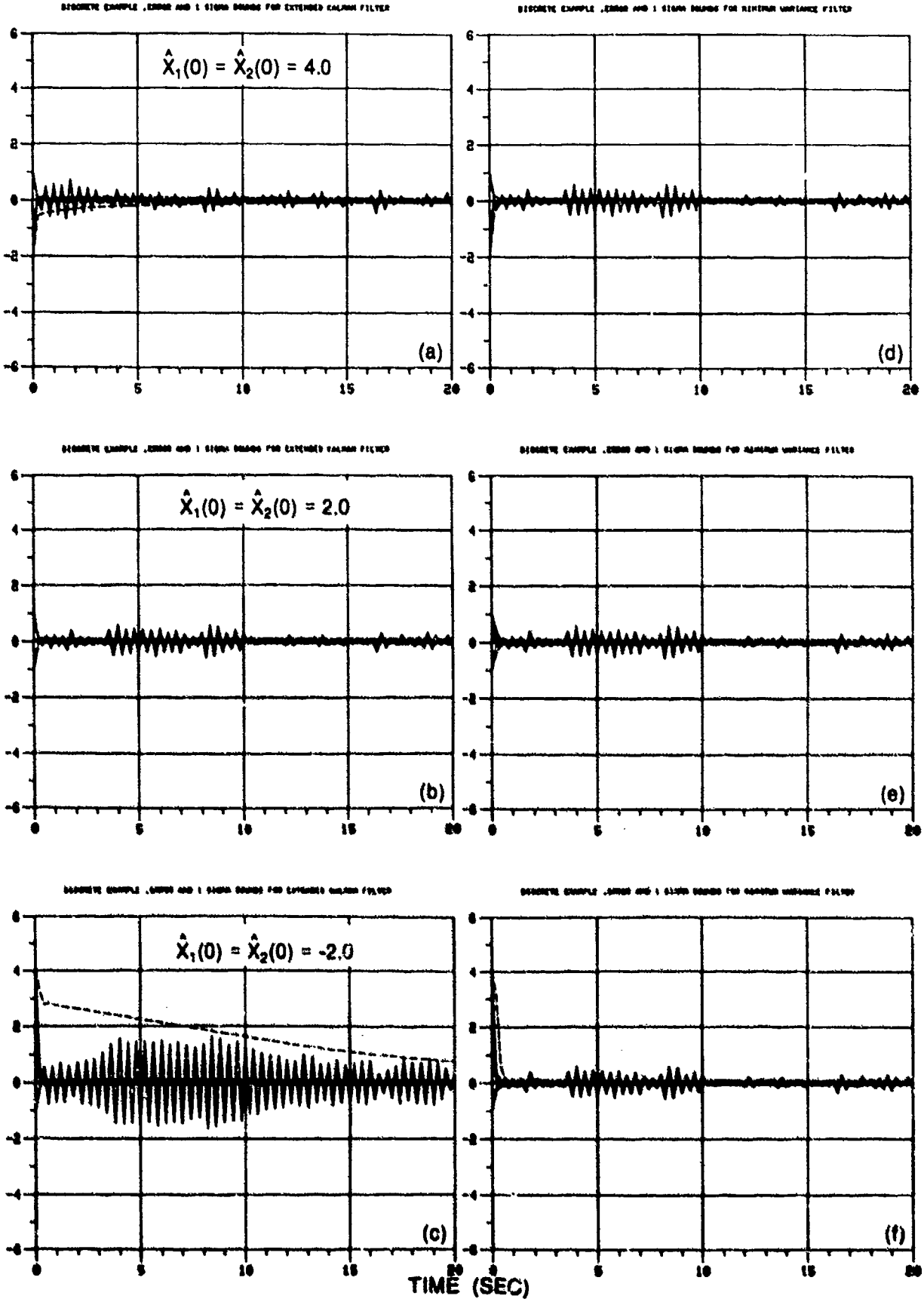


FIGURE 3.9: COMPARISON OF EKF OUTPUTS; EFFECTS OF INITIAL V WITH SMALL Ψ

$\Psi_V = \Psi_W = 0.01$

$X_1(0) = X_2(0) = 2.0$

$\hat{X}_1(0) = \hat{X}_2(0) = -2.0$

MEASUREMENT MODEL: $Z = X_1 + V$

——— FILTER ESTIMATE $\hat{X}_1$

---------- FILTER ESTIMATE $\hat{X}_2$

EXTENDED KALMAN FILTER OUTPUT FOR STATE X1 AND X2

MINIMUM VARIANCE FILTER OUTPUT FOR STATE X1 AND X2

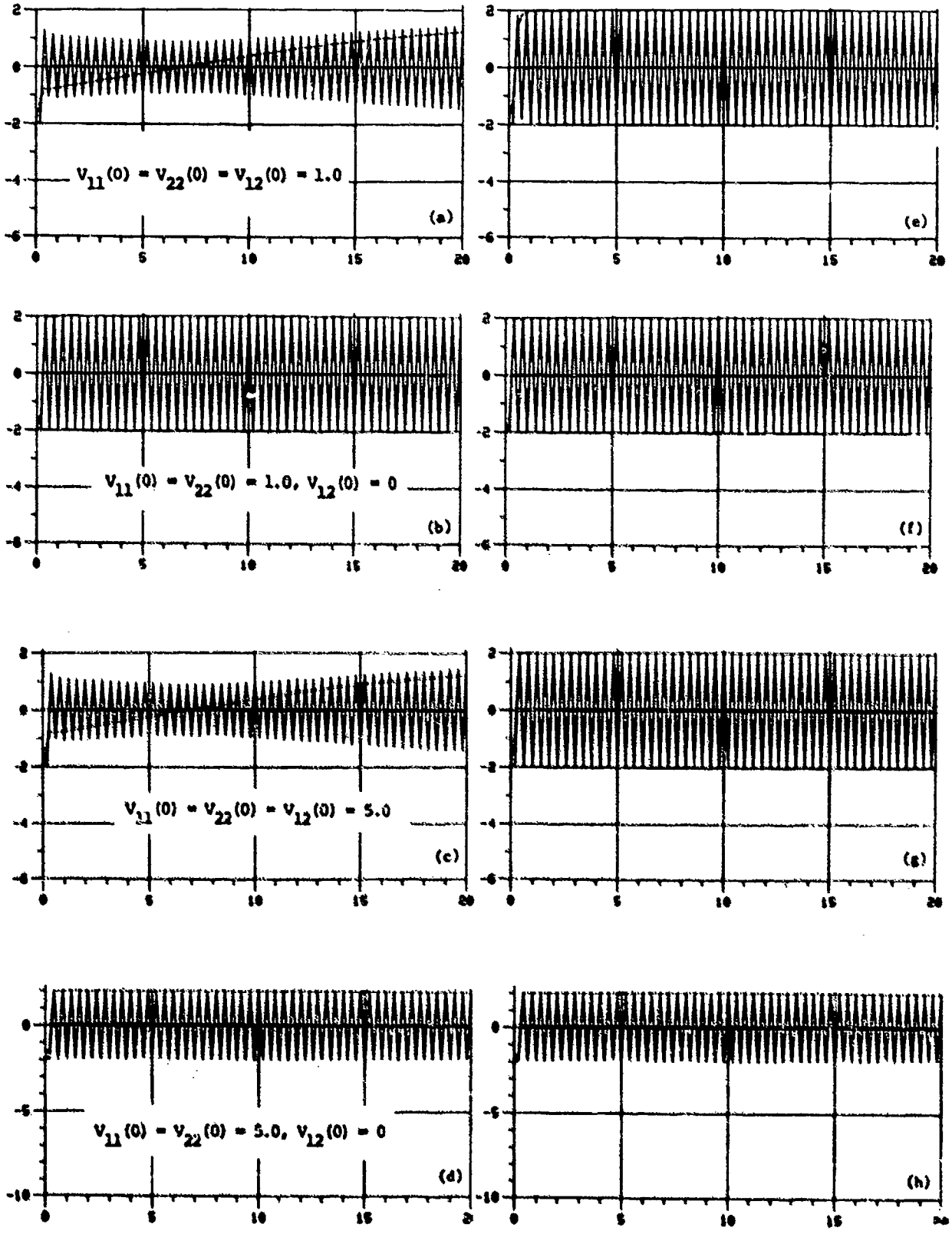


FIGURE 3.10: COMPARISON OF OUTPUTS; EFFECTS OF INITIAL V WITH LARGER Ψ

$\Psi_V - \Psi_W = 1.0$

$X_1(0) = X_2(0) = 2.0$

$\hat{X}_1(0) = \hat{X}_2(0) = -2.0$

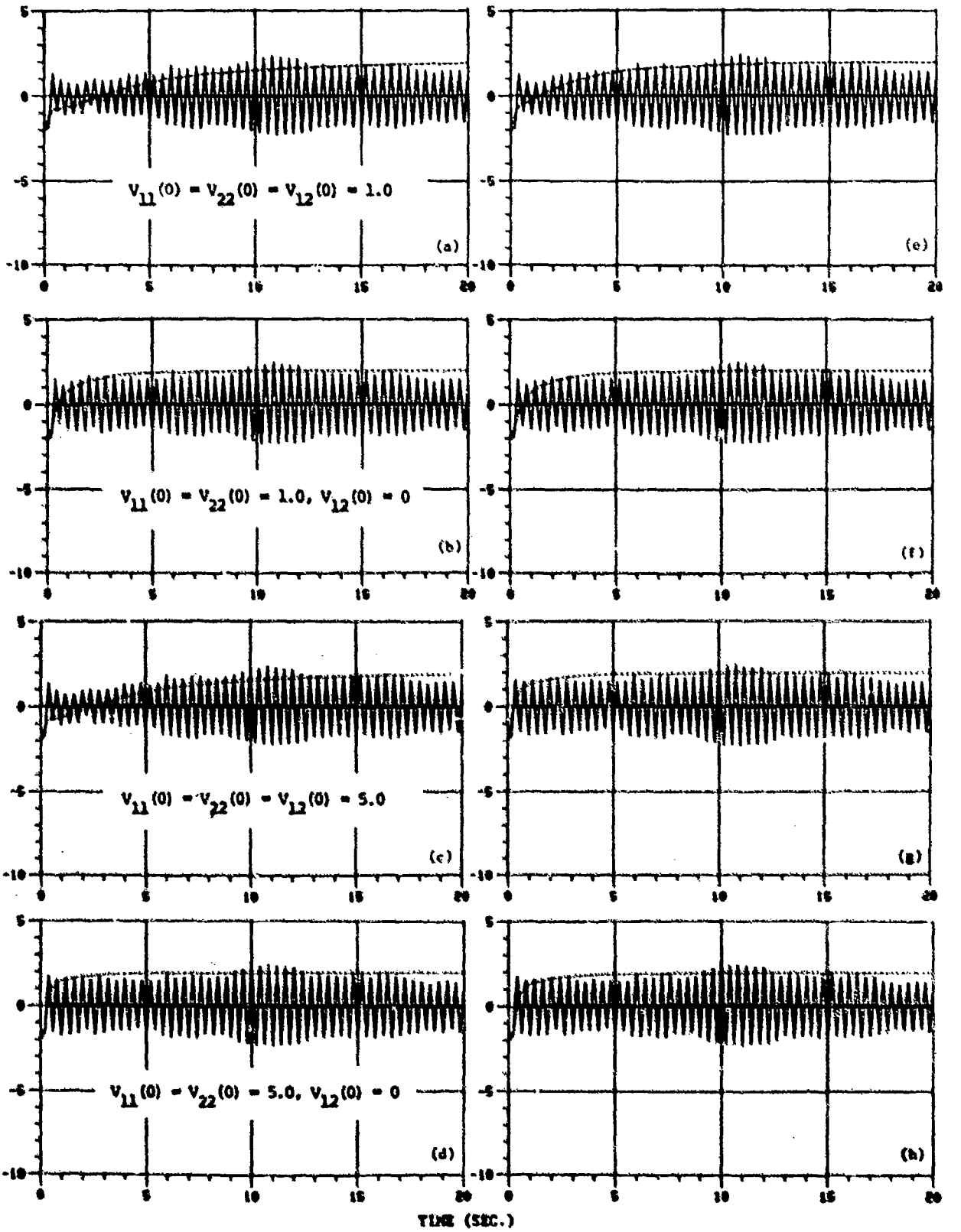
MEASUREMENT MODEL: $Z = X_1 + V$

———— FILTER ESTIMATE $\hat{X}_1$

---------- FILTER ESTIMATE $\hat{X}_2$

EXTENDED KALMAN FILTER OUTPUT FOR X1 AND X2

MINIMUM VARIANCE FILTER OUTPUT FOR X1 AND X2



KALMAN FILTER SATELLITE ORBIT IMPROVEMENT USING LASER RANGING MEASUREMENTS FROM A SINGLE TRACKING STATION

K.F. Wakker and B.A.C. Ambrosius
Delft University of Technology
Department of Aerospace Engineering
Kluyverweg 1, Delft, The Netherlands

SUMMARY

Modern satellite ranging lasers emit short pulses at a low beam divergence and therefore require accurate satellite position predictions. This paper deals with a study to investigate the possibilities to use the laser range observations acquired at only one tracking station to provide real-time position prediction updates during a pass, and also better predictions for subsequent passes over that station. A computer program, called SORKA, has been developed, which is based on an extended Kalman filter scheme. The computational approach adopted in SORKA is described in some detail. In particular, the methods to compute the state-transition matrix and the state-noise covariance matrix are emphasized. Typical divergence phenomena arising from processing accurate range-only measurements are discussed and the effectiveness of techniques to reduce these instabilities is demonstrated. Laser range measurements acquired during 8 successive passes of GEOS-1 over the Kootwijk groundstation have been processed and some results are presented.

1. INTRODUCTION

Since August 1976 the Department of Geodesy of Delft University of Technology operates a satellite laser ranging system at Kootwijk in the Netherlands. From there, the Working Group for Satellite Geodesy (WSG) acquires on a routine basis day and night ranging data of geodetic satellites. Within the period 1976 to 1979 more than 49,500 observations have been obtained at Kootwijk during 1108 passes of BEACON-C, GEOS-1, GEOS-2, GEOS-3, SEASAT, STARLETTE and LAGEOS (Ref. 1). All these satellites encircle the earth in near-circular orbits between 800 km and 2300 km altitude, except for LAGEOS whose altitude is about 5900 km. In 1980, about 13,500 observations were acquired during 370 passes of GEOS-3, STARLETTE and LAGEOS. The ruby pulse-laser system consists of a multi-mode Q-switched oscillator, a spark-gap activated pulse chopper and two amplifier stages (Refs. 1, 2). The output energy in routine operation is 1 to 2 J, with a maximum of 3 J. The transmitted laser beam has a diameter of 19 cm and the divergence is adjustable from 1 to 20 arcminutes. Untill autumn 1980, the system generated 4 ns wide pulses at a maximum rate of 15 pulses per minute, producing measurements with an accuracy of about 25 cm root-mean-square. Recently, the accuracy level has been improved to about 15 cm, by reducing the pulse width to 2 ns. In addition, a new range-gate generator has been installed, having a manually adjustable time window with a minimum half-width of 0.1 µs. This will possibly give the capability of ranging also to LAGEOS in daytime.

For many years there exists a close cooperation between WSG and the Section Orbital Mechanics (SOM) of the Department of Aerospace Engineering of Delft University of Technology. This Section supports WSG in the field of orbit computations for the geodetic satellites used in the laser ranging activities. The support ranges from satellite position predictions, needed for the automatic pointing of the laser, to orbit determination and geophysical parameter estimation from laser observations acquired at Kootwijk and other laser ranging stations. To increase the accuracy of the laser pointings, it was decided in 1978 to investigate the possibilities to use laser observations from Kootwijk in a (semi-) real-time mode. This paper describes the preliminary results of that study.

2. LASER POINTING PREDICTIONS

For the routine operations of pointing the laser at a satellite, at present use is made of the AIMLASER computer program, developed at the Smithsonian Astrophysical Observatory (SAO), Cambridge, Massachusetts. This program has been modified by SOM to satisfy the specific needs of WSG and is regularly improved and updated. This Delft-version of AIMLASER (Ref. 3) is also in use at a number of other European laser stations. The input for the orbit prediction program consists of a set of mean orbital elements, distributed weekly by SAO. These parameters are determined by SAO from the so-called quick-look laser ranging data as returned to SAO by many groundstations distributed all over the world. Experience has shown that quite often the satellite position predictions on basis of these SAO elements and using the AIMLASER program reach a level of inaccuracy which is incompatible with lasers emitting a low-divergence beam. To give an indication of the accuracy needed, consider a satellite which passes over Kootwijk at a distance of 1000 km. When the beam divergence is 2 arcminutes, the diameter of the beam at the altitude of the satellite is 580 m. So, in this case a position prediction accuracy of about 300 m is needed in order to guarantee that the satellite is hit by the laser pulse.

To correct for position prediction errors, the Kootwijk laser system has been equipped with a manually controlable firing-time adjustment switch. During nighttime passes, if the satellite is sunlit, the operator may look through the laser telescope in order to estimate by what amount the firing time has to be delayed or advanced. This is possible for satellites up to a visual magnitude of +13. Usually, however, if at the beginning of

a pass no return signal is received, a systematic variation of the time adjustment is applied until returns are being registered. Sometimes, the adjustment can remain constant during a pass, but on other occasions re-adjustments are necessary. By this technique, the operator in fact corrects for position errors in the direction of the satellite's path across the sky. It has been found (Ref. 4) that a considerable improvement of the score of laser returns is obtained by this manual control of the laser firing time.

One of the main disadvantages of this technique is that it corrects for in-track position errors only, while errors perpendicular to the satellite's path cannot be corrected for. It actually has occurred that the observer saw through the telescope that the satellite was too far from the predicted track without being able to apply corrections. Therefore, a need exists for more-precise position predictions. This need becomes even more stringent with the development of more-advanced laser systems operating with a smaller beam divergence. But there is still another need for more-accurate predictions. To minimize the chances of false triggerings by noise pulses, a laser system uses a range-gate generator, which determines the time window in which the return signal is expected. For daytime ranging to distant satellites, like LAGEOS, very short windows of up to 0.1 μs are required. But this implies that the radial distance to the satellite has to be predicted with an accuracy of better than 30 m.

To increase the accuracy of the predicted satellite positions a number of possibilities exists. For instance, it would be possible to replace the SAO elements by more-accurate orbital parameters and to use a satellite position prediction program that is more accurate than AIMLASER. The primary reason for the SAO elements being sometimes inaccurate is that they are based on the quick-look ranging data, which may be rather sparse during some periods and which may contain systematic or gross errors. Therefore, an approach may be adopted as applied by the University of Texas at Austin for the prediction of LAGEOS. In that case, several weeks of laser observations from a number of groundstations are used to determine the orbit of the satellite very accurately. From this orbit determination an extremely accurate orbit is extrapolated for a long time in advance to yield one or more state vectors for each day. From these state vectors, the station generates its own laser pointing angles and computes the satellite's distance by numerical integration of the equations of motion. At Kootwijk, research is going on along this line to improve the predictions for LAGEOS. It is doubtful, however, if such a technique will yield prediction accuracies for satellites below 1500 km altitude that satisfy the needs of the narrow-beam laser stations.

Therefore, it is felt that it would be attractive to process in real time the laser range measurements registered during a pass over a station in order to increase the accuracy of subsequent laser pointings and window settings during that pass. This means that all computations have to be performed on an on-line computer within the period between two successive laser firings. To start up the process it is required that at the beginning of a pass observations are actually acquired. Fortunately, at low elevation angles the slant range to the satellite is large so that there is a good chance that the satellite is within the laser beam, even if the position predictions are relatively bad. After the first observations have been processed and the accuracy of the position prediction has increased, the beam divergence and the time window setting may be narrowed. In case the predictions are so bad that no returns are received, a systematic search procedure may be executed at the beginning of a pass. This type of closed-loop tracking is primarily applicable to stations with medium- to high-power lasers, since only then sufficiently wide beams can be used to guarantee laser returns.

Recent observations from a station can also be used in an off-line mode to improve the orbit and by that the laser pointing angles for subsequent passes. This also makes the station less dependent on a distant large computing center. This aspect could be of great importance for the mobile laser ranging systems. These systems, however, are designed around low-power lasers (about 10 mJ per shot) operating with a beam divergence of about 0.5 arcminute and a high repetition rate of about 10 pulses per second. Signal levels are on the average below one photoelectron per shot for LAGEOS and in the order of ten photoelectrons per shot for satellites in lower orbits. As a result, large numbers of false returns are recorded which need to be identified in order not to upset the orbit improvement scheme. Whether this identification will be possible during a satellite pass over the station still has to be investigated.

The main objective against the real-time laser pointing update approach is the fact that from the orbital mechanics point of view, it is very precarious to use range measurements from only one groundstation to improve the satellite's orbit. It is evident that from such observations during one pass very little information is gained on the orientation of the orbital plane. Therefore, the basic scheme presently envisaged is that previous range measurements obtained during several passes over the tracking station yield the orientation of the orbit to such a level of accuracy that during the next pass the range measurements can be used in real time mainly to improve the prediction of the satellite's position in the orbital plane. During the very first part of a pass the laser mount's angular position read-outs can possibly also be used as observation quantities. Though these angles will, in general, contain systematic errors, the inclusion of these quantities for the whole tracking period as quasi-observations with a relatively low weight might also be attractive to stabilize the orbit improvement process. Nevertheless, it is anticipated that, periodically, after long observation gaps, additional orbital information will be required which is derived from tracking data acquired at other groundstations.

3. THE EXTENDED KALMAN FILTER

It is well known that, basically, there exist two methods for processing satellite tracking data to estimate the orbit of a satellite. The batch processor, which yields the estimate at some reference epoch, requires that the entire sequence of observations be processed before the estimate can be made. On the other hand, the sequential orbit determination procedure processes one observation at a time and produces an estimate of the state vector at the observation time. So, this sequential procedure suits best the requirements for real-time orbit improvement, although it can also be used to predict the next satellite pass over the station. While the formal mathematical equivalence between the sequential estimation algorithm and the batch estimation algorithm can be shown, it is known from practical experience that the sequential process is much more sensitive to errors introduced by linearizations, which may result in estimate divergence problems.

The technique selected in this study is the extended Kalman filter, which is described in many textbooks (e.g. Ref. 5). An interesting geometric derivation of the Kalman filter equations is given in Ref. 6. Applications of the extended Kalman filter to orbit dynamics problems are discussed extensively in Refs. 7-14. For this study a computer program, called SORKA (Satellite Orbit Refinement using a KAlman filter) has been developed, which at the moment functions primarily as a test program. Therefore, it contains at present different options and alternatives to investigate in detail the Kalman filter characteristics for this application. It is emphasized here that the aim of SORKA is not to reach the high accuracy level and the extensive capabilities of computer programs like GEODYN (NASA) or UTOPIA (Univ. of Texas), but merely to satisfy the accuracy level needed for laser pointings and to be compatible with small local computers. Until now, SORKA runs on an IBM 370/158 computer, but in the design of the program precautions have been taken to make the implementation of a stripped-down version on a small local computer possible. To elucidate the description of the computational approach adopted in SORKA, a brief outline of the filter scheme will be given.

The satellite motion can be described by the set of equations

$$\dot{\underline{X}} = F(\underline{X}, t); \quad \underline{X}(t_0) = \underline{X}_0 \tag{1}$$

where $\underline{X}$ denotes the state vector. When $\underline{X}_0$ is specified and F is known, the orbit of the satellite, and thus its state at a later time, can be obtained from integration of Eq. (1). In orbit determinations, $\underline{X}_0$ is not known perfectly and therefore observations of the motion must be processed to obtain a more-accurate estimate of the state vector. Assume that a best estimate $\hat{\underline{X}}_0$ of the state at t_0 is known with an accuracy represented by the covariance matrix $\hat{P}_0$. In the time-update step, Eq. (1) is integrated to yield an initial state estimate at the time of the first observation t_1 : $\underline{X}_1$. Its covariance matrix, P_1, is computed from

$$P_1 = \Phi_{1,0}\, \hat{P}_0\, \Phi_{1,0}^T + Q_{1,0}$$

where $\Phi_{1,0}$ is the state-transition matrix, used to map the state from t_0 to t_1 and defined by

$$\Phi_{1,0} = \left(\frac{\delta \underline{X}_1}{\delta \underline{X}_0}\right)_{\hat{X}_0}$$

and $Q_{1,0}$ is the state-noise covariance matrix, representing errors occurring during the state vector integration. These errors, which are assumed in this study to be Gaussian distributed with a zero mean, include such effects as deterministic model errors, numerical integration errors and random process noise.

In the subsequent observation-update step the Kalman filter adds the information of the observations $\underline{Z}_1$ at t_1 to the initial estimates, yielding more-accurate estimates $\hat{\underline{X}}_1$ and $\hat{P}_1$. For this, in SORKA the general non-linear relations between the measured quantities and the state vector are used:

$$\underline{Z} = O(\underline{X}, t) + \underline{V} \tag{2}$$

where the stochastic vector $\underline{V}$ represents Gaussian distributed measurement errors with a zero mean and a covariance matrix R. The observation residuals at t_1 : $\underline{\varepsilon}_1$, are defined by

$$\underline{\varepsilon}_1 = \underline{Z}_1 - \underline{Z}_1^*$$

where $\underline{Z}_1^*$ denotes the predicted observations as computed from Eq. (2) and the initial estimate $\underline{X}_1$:

$$\underline{Z}_1^* = O(\underline{X}_1, t_1)$$

The new estimate $\hat{\underline{X}}_1$ is given by

$$\hat{\underline{X}}_1 = \underline{X}_1 + K_1\, \underline{\varepsilon}_1 \tag{3}$$

where K_1 is the Kalman gain matrix, satisfying

$$K_1 = P_1\, H_1^T (H_1\, P_1\, H_1^T + R_1)^{-1} \tag{4}$$

and H_1 is the observation matrix, defined by

$$H_1 = \left(\frac{\partial G}{\partial \underline{X}}\right)_{\underline{X}_1}$$

So, the matrix H is evaluated for the most accurate state estimate available at this stage: the reference state $\underline{X}_1$. The state covariance matrix at t_1 corresponding to the optimal state estimate can be found from

$$\hat{P}_1 = (I - K_1 H_1) P_1$$

where I is a unit matrix of appropriate dimensions. Once $\hat{\underline{X}}_1$ and $\hat{P}_1$ are known, the same process is repeated to compute the state and the state covariance matrix at the time of the next observation. In this way, the best estimate at any time contains the information of the last and all previous observations.

The scheme given above holds for all types of observations. In this study, the observations are ranges from the laser to the satellite, but SORKA has been designed such that in the future also azimuth and elevation observations can be dealt with. As a reasonable assumption, range, azimuth and elevation can be considered independent observations. Then, a computational simplification is possible, permitting the individual observations at each observation time to enter the algorithm one after another as scalars. In that case, the term in brackets in Eq. (4) reduces to a scalar. So, the inversion of this term reduces to a division by a scalar, which avoids numerical problems which may occur in matrix inversions.

For each time-update step, the function F, the state-transition matrix, ϕ, and the state-noise covariance matrix, Q, have to be evaluated. For the integration of the state equations, Eq. (1), a relatively simple force field has been adopted. During a pass only the first five zonal harmonics (J_2 to J_6) and the first tesseral harmonic ($J_{2,2}$) of the Legendre series expansion for the earth's gravity field (geopotential) are accounted for. For the state integration between subsequent passes a more-extended gravity model is used, including terms up to $J_{6,6}$. The dynamical equations are integrated with a fourth-order Runge-Kutta method. The stepsize depends on the time between subsequent measurements and has an upper limit of 40 s. The computation of the state-transition matrix and the state-noise covariance matrix is described in the following Sections. In each observation-update step, the function G and the matrices R and H are required. The measurement-noise covariance matrix is assigned the values of the known measurement noise variances. The observation matrix is evaluated at a reference state, usually taken as $\underline{X}_1$. In Section 6 an alternative for this reference state will be introduced.

4. THE STATE-TRANSITION MATRIX

From Eq. (1) a differential equation can be derived for the state-transition matrix:

$$\dot{\phi}_{t,0} = \left(\frac{\partial F}{\partial \underline{X}}\right)_{\underline{X}_t} \phi_{t,0} \; ; \quad \phi_{0,0} = I \tag{5}$$

These so-called variational equations can be integrated numerically, together with the state equations. An alternative approach is to avoid the use of ϕ and to directly integrate a matrix Riccati differential equation in P (Refs. 9, 10). Such a direct numerical integration provides a mathematically rigorous solution and is readily applicable in the presence of all types of disturbing forces. The method, however, necessitates the simultaneous integration of a large set of differential equations, thus making the procedure time consuming. As SORKA is designed for real-time operations, it was therefore decided to compute the state-transition matrix in some approximating analytical way. It is realized that this approach may entail subtle but important effects which impair the accuracy of the results, in particular when the analytical expressions for the computation of ϕ require a simplification of the force field, while for the state integration a more-extensive force field is used. However, studies described in Refs. 15, 16 hint that, in general, a truncation of the force field is permissible, provided at least J_2 is explicitly included in the variational equations. For SORKA, two alternative analytical techniques have been explored.

The first method can only be applied if the period between successive measurements is relatively short. It is based on the assumption that the expressions for the gradient of the geopotential can be linearized over the time interval between successive measurements. Then, from Eq. (5) the approximative relation

$$\dot{\phi}_{1,0} = N_0 \, \phi_{1,0} \tag{6}$$

can be derived (Ref. 17), where the matrix N_0 can be partitioned into four sub-matrices: two being a 3x3 null matrix, one a 3x3 unit matrix and one a 3x3 matrix containing the second-order partial derivatives of the geopotential with respect to the X, Y and Z components of the state vector. Integration of Eq. (6) leads to an exponential function which can be approximated by the series expansion

$$\phi_{1,0} = I + N_0 \Delta t + \frac{1}{2} (N_0 \Delta t)^2 + \frac{1}{6} (N_0 \Delta t)^3 + \ldots$$

where $\Delta t = t_1 - t_0$. Analytical expressions have been derived (Ref. 17) for the second-order derivatives of the geopotential where the effects of the zonal harmonics J_2 upto J_6

and the first tesseral harmonic $J_{2,2}$ are accounted for. These expressions, as well as the expressions for the first-order partial derivatives, which are required for the state integration, were composed by applying the REDUCE formula-manipulation computer program, which is available on the DEC-10 computer of Twente University of Technology. Though this analytical technique of computing ϕ has been implemented in SORKA and has been tested extensively, for the computations described in this paper a second analytical technique has been used exclusively, since it consumes less computer time.

This second technique was developed primarily for the computation of the state-transition matrix for longer periods between two successive measurements, such as between two different passes. The analytical expressions for the elements of the state-transition matrix are obtained by applying the chain-rule to the relations that exist between the state vector and the orbital elements at any time, and to the relations between the orbital elements at a time t_1 and those at t_0:

$$\frac{\partial \underline{X}_1}{\partial \underline{X}_0} = \left(\frac{\partial \underline{X}}{\partial \underline{p}^{osc}}\right)_1 \cdot \frac{\partial \underline{p}_1^{osc}}{\partial \underline{p}_0^{osc}} \cdot \left(\frac{\partial \underline{p}^{osc}}{\partial \underline{X}}\right)_0 \qquad (7)$$

where $\underline{p}^{osc}$ denotes the vector of osculating orbital elements. The matrices in brackets can be derived easily from the geometrical expressions describing a Keplerian orbit relative to the non-rotating geocentric reference frame (e.g. Ref. 18). To avoid the classical problems for orbits with very low eccentricity or inclination, in principle the use of non-singular orbital elements is preferable. It was demonstrated in Ref. 19, however, that in practice, if the computations are performed on a computer with a reasonable word-length, the problems occur only at extremely low values of eccentricity or inclination. For a word-length of 64 bits, for example, the classical elements can be used if e and sin(i) are larger than 10^{-10}. For simplicity, therefore the classical elements were adopted in SORKA to compute the state-transition matrix. To guarantee that singularity problems will never occur, a simple engineering measure has been introduced which precludes that e and sin(i) can take values smaller than 10^{-10}.

It has been shown in Ref. 19 that, for the application described in this paper, the matrix $\partial \underline{p}_1^{osc}/\partial \underline{p}_0^{osc}$ can be computed with sufficient accuracy by neglecting short-periodic and long-periodic perturbations in the osculating elements, and by taking into account only the secular perturbations due to the oblateness of the earth (J_2). If the orbital altitude is less than 400 km, which is very unlikely for geodetic satellites, also the secular perturbations due to atmospheric drag should be accounted for. Analytical expressions were derived to compute the matrix when only these two types of perturbations are considered. In SORKA only the J_2 secular perturbations are taken into account when computing the middle matrix in Eq. (7). A further simplification was introduced by substituting mean instead of osculating elements in the expressions for the matrices. These mean elements are obtained by subtracting the short-period J_2 contribution from the osculating elements, according to the non-singular elements conversion technique described in Ref. 20. So, the computation sequence is as follows. From the state estimate at the start of the filter process, t_0, the osculating elements are computed. These are converted into mean elements. The values of the orbital elements at the start of the first pass, t_1, are computed by adding only the secular perturbations (e.g. Ref. 21) to the mean elements at t_0. Next the matrix $\partial \underline{p}_0/\partial \underline{X}_0$ is computed, applying the usual transformation relations between (osculating) orbital elements and position and velocity (e.g. Ref. 18). Subsequently, the matrices $\partial \underline{p}_1/\partial \underline{p}_0$ and $\partial \underline{X}_1/\partial \underline{p}_1$ are computed. Matrix multiplication finally yields $\partial \underline{X}_1/\partial \underline{X}_0$. For subsequent gaps between passes, the same process is repeated, but then t_0 is the time of the last observation of the previous pass and t_1 becomes the time of the first observation of the next pass. If this technique is applied for the computation of the state-transition matrix during a pass, t_1 simply becomes the time of the next observation. Although there are numerous approximations in this approach, it was found (Ref. 22) that they hardly affect the results.

5. THE STATE-NOISE COVARIANCE MATRIX

It is well known that Kalman filter applications often suffer from state estimate divergence problems. In principal, these are a result of non-linearities, errors due to an incomplete mathematical model and to a lesser extent also of computational truncation and round-off errors. Physically, the state divergence can be explained as follows. When during the observations processing the state estimates become more accurate, and hence the covariance matrix becomes smaller, new observations, which reflect the true state, will yield progressively smaller corrections. If there are too many gross measurement errors or if there is any error in the dynamical model that is not accounted for properly by the assumed model errors, represented by the state-noise covariance matrix Q, then successive estimates of the state may tend to follow an erroneous course and to diverge from the true state. Consequently, the estimated state covariance matrix fails to represent the true estimation error.

In the time-update step the state vector and its covariance matrix are integrated. During this integration, errors will be introduced due to dynamic modeling errors and integration errors. Generally, the errors will be non-random. Methods have been developed (e.g. Refs. 8, 9, 11, 12, 23) to account for the model errors in some way. For computational simplicity, in SORKA the crude assumption has been made that the errors are random and can be handled by a proper choice of the covariance matrix Q. A suitable choice for Q that prevents filter divergence has to come from experience gathered during

tests on the filter performance. In SORKA, two methods are used to compute the state-noise covariance matrix.

For the short time intervals between successive laser observations during a pass, the computation is based on the assumption that the unmodeled forces acting on the satellite yield accelerations that have the same root-mean-square value in all three coordinate directions. At present, a value of 2.5 m/s/day is used. From these accelerations, the standard deviations of the velocity errors after a time-update step are found by multiplying the acceleration by the length of the time interval. These three standard deviations of the velocity are the only components of Q that are used. The standard deviations of the position error after the next time-update step evolve from these components through the state-transition matrix. It can be argued physically, and it was demonstrated by numerical experiments (Ref. 22), that model errors are of minor importance during passes. The filter process was shown to be rather insensitive to relatively large variations of the values selected for the unmodeled accelerations. This is due to the fact that force model errors cannot build up large effects during a pass and that non-linearity errors mostly dominate the state estimation (Section 6).

For the integration interval between successive passes, a different approach for computing Q has been selected. Since the state errors are mainly due to the truncation of the gravity field model used for the integration of the state vector, an upper bound for the magnitude of these errors can be estimated. In the current study, where observations of only one tracking station are processed, these errors will be considerably smaller than indicated by that upper bound. This is because in that case the satellite traverses during the passes always (nearly) the same spatial region of the gravity field, and only a fraction of the total, mainly periodic, variation of the orbital parameters due to unmodeled gravitational forces has to be accounted for in Q. After a number of tests for satellites at altitudes of 1000 km to 2000 km, fixed values were selected for the along-track, cross-track and radial position and velocity errors equal to 20 m and 2 cm/s, respectively. In addition, a few dominant correlations have been introduced in the state-noise covariance matrix in cross-track, radial and along-track components. Finally, an orthogonal transformation is applied to obtain the corresponding matrix in X, Y and Z components. The Q-matrix is then added to the state covariance matrix, P, which has been integrated in one step over the complete time interval between the passes, using the analytically computed state-transition matrix.

6. EFFECTS OF NON-LINEARITY ERRORS

The Kalman filter technique has been developed for linear systems. When the filter is applied to the highly non-linear equations encountered in orbit dynamics, approximations linearized about a reference state are used. Because in SORKA the full equations of motion are integrated numerically and the general non-linear relations are used for the computation of the measured quantities from the state vector, linearizations only occur in the integration of the state covariance matrix, P, and in the Kalman algorithm for the observation-update step. In Section 4, it has been pointed out that the filter process is not very sensitive to the method used for the integration of P. So, linearization errors will mainly be introduced in the observation-update step.

During testruns with simulated laser range observations of GEOS-3 (Section 8), it was found (Ref. 24) that after the first few range observations had been processed, excessively large state corrections occurred, although the initial residuals were found to be relatively small, while the data were known to contain no gross measurement errors. Normally, one would expect the state corrections to decrease gradually as more observations are processed, because the state estimates become more accurate. The testruns were started with an input state vector corrupted with noise of 500 m standard deviation for the position components and 0.5 m/s for the velocity components.

To investigate why the state corrections did not decrease gradually, the behavior of the Kalman gain matrix, K, was studied in more detail. Since the initial residuals were relatively small, only this matrix could cause the large state corrections. As the linearizations introduced through the observation matrix, H, were suspected to be responsible for this filter behavior, a test was performed in which the effects of changes in the reference state on the gain matrix were studied. As only range measurements are processed, the observation matrix and the gain matrix reduce to a row matrix and a column matrix, respectively. This simplifies the interpretation of the results. The predicted state vector and its covariance matrix were extracted out of a SORKA simulation run at the time of the fourth observation, when problems first occurred. Then, the position elements of the state vector were varied systematically by applying changes which were proportional to the state correction vector $K\underline{z}$, as computed by SORKA at that observation time. This was done to insure that the applied changes of the reference position vector were in the same direction as the nominal position correction computed by SORKA. The velocity components were not considered in the evaluation because they do not appear in the matrix H. Each new state vector thus obtained was used as an alternative reference state for which a new gain matrix was computed. Plotting the individual elements of these column matrices against the total change of the reference state, finally yielded an indication of the dependence of the Kalman gain matrix on the reference state. In Fig. 1 the results are presented for the two elements of the gain matrix that affect the X and Y components of the state vector. It is clear that K is strongly dependent on the reference state and even shows a near-singular behavior. These high gain values lead to large state corrections, which may result in considerable linearization errors and may cause divergence. In this test the noise level of the

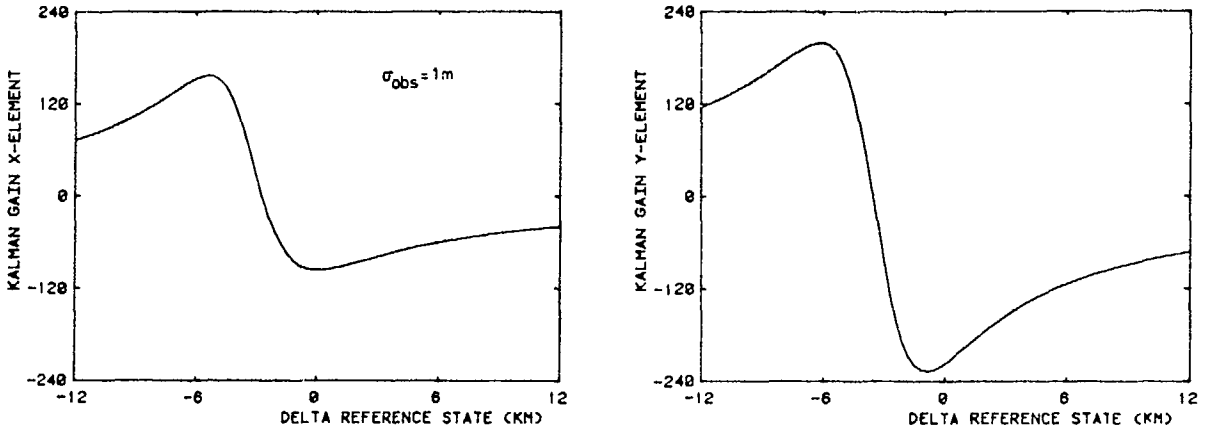


Fig. 1: The dependence of the gain matrix on the reference state.

simulated observations was 1 m. Additional tests proved that smaller measurement standard deviations further amplify the near-singular behavior and that a larger measurement noise lessens the dependence of the gain on the reference state.

Physically, this peculiar behavior of the gain can be explained as follows. At the start of a pass, the accuracy of the measurement is much higher than the accuracy of the state vector. In Fig. 2a this is visualized by a position variance ellipsoid and a thin slice representing the first range measurement and its variances. After processing the first few highly-accurate range measurements, which are all taken in nearly the same direction, the position variance ellipsoid will be flattened considerably in about the direction of the observations (Fig. 2b). The information content of only a few observations is generally not sufficient to yield a good approximation of the real satellite orbit. So, the next observation may lie relatively far outside the error ellipsoid. In combination with the extreme flattening of the error ellipsoid this causes the filter to try to cure the situation by producing a large position shift vector. This vector points in a direction that is governed mainly by the orientation of the major axis of the ellipsoid; its magnitude can be much larger than the state standard deviation in that direction.

For visualizing the filter process for the case illustrated in Fig. 2b, further simplifications can be introduced. The strongly flattened position variance ellipsoid is approximated by a plane, which is about perpendicular to the observation vector (Fig. 3a). The range measurement variances can be approximated by a plane perpendicular to the range vector as computed from the reference state. Using this representation it will be clear that the observation-update step will yield an optimal state which lies on the intersection of the two planes. The state correction is dependent on the range residual,

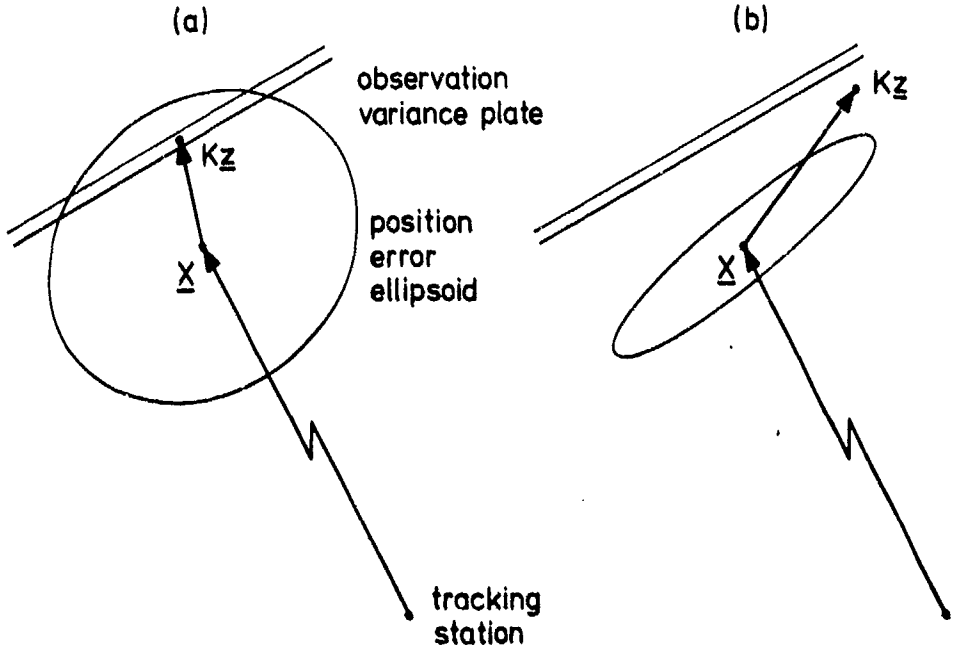


Fig. 2: Sketch of the state correction at the start of a pass (a) or after the first few highly-accurate range observations have been processed (b).

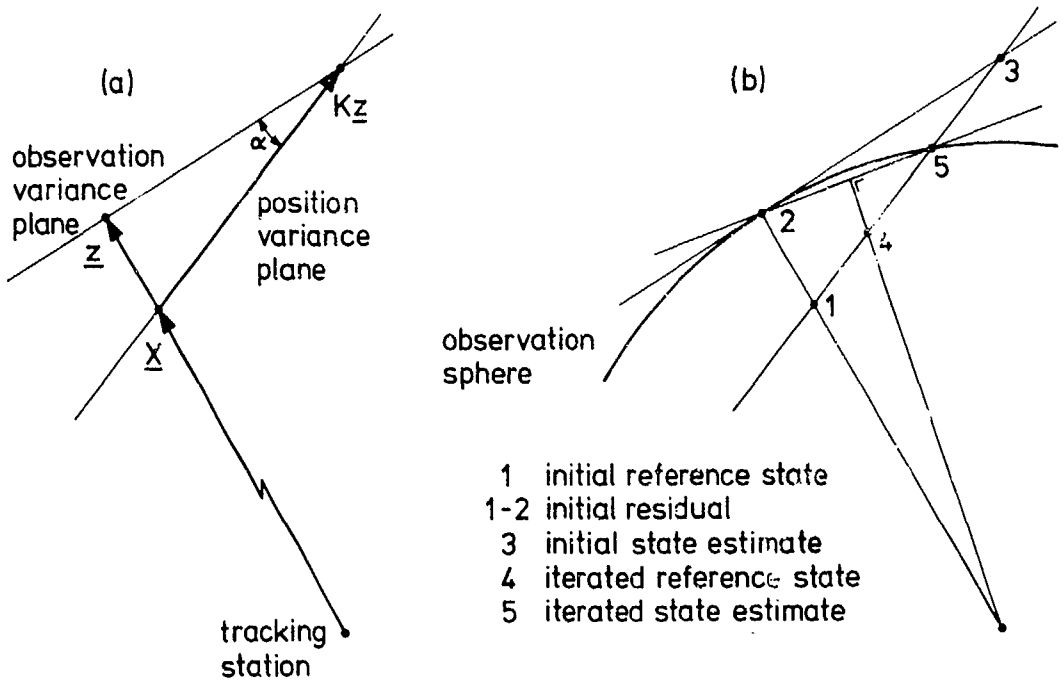


Fig. 3: Simplified diagram of the state correction after the first few highly-accurate range observations have been processed, as provided by the pure Kalman filter (a) or by the filter with the reference state iteration modification (b).

$\underline{z}$, and the angle α between the two planes. Comparison with Eq. (3) shows that in this simpified diagram the magnitude of the Kalman gain corresponds in a first approximation to $1/\sin\alpha$. This explains the behavior of K as shown in Fig. 1. When both planes are nearly parallel, small changes in the reference state along the position variance plane will cause large variations of α and thus of the gain matrix. However, if α becomes zero, the approximation of the state and measurement variances by two (parallel) planes fails and the gain matrix becomes very small.

Large state corrections will certainly introduce appreciable non-linearity errors in the state estimates. This is illustrated in Fig. 3b, from which it may be concluded that the initial (linear optimal) state estimate (3) does not coincide with the observation sphere described by the range observation. In order to force the state estimate towards the intersection of the sphere with the position variance plane, representing a more-realistic approximation of the true optimal estimate, an iteration scheme has been derived (Ref. 24) that also reduces the magnitude of the state corrections, and that yields improved filter performance. During the iterations the values of P_1, R and $\underline{z}_1$ are held fixed. In the first iteration step the initial state estimate, $\underline{X}_1$, is used as the reference state. In the next iteration step, the mean of the initial reference state and the first updated state estimate, $\hat{X}_1$, is used as the reference state. For this new reference state, new values of H and K are computed. Using this new K value, $\hat{P}_1$ is recomputed and a new state correction $K\underline{z}_1$ is computed, which is added to the initial reference state, $\underline{X}_1$. The iterative process stops if the computed values for the reference state have converged (Fig. 3b). Sometimes, however, the process does not converge at all. This may happen, for instance, if the state variance plane lies outside the observation sphere. To cope with this problem the measurement standard deviation may be increased, but from a theoretical point of view this is a precarious measure, since it will also allow erroneous observations to enter the solution.

Although the iteration scheme improves the stability of the filter, errors will inevitably build up during the processing of the first few observations when the iterative process is not yet effective and may still cause filter divergence. Eventually, a simple engineering solution has been adopted (Ref. 24) that proved to be highly effective. Before each observation-update step the diagonal elements of the state covariance matrix, P, are multiplied by a number slightly greater than one. By this, the relative magnitude of the correlation coefficients, which, in general, describe the flattening of the error ellipsoid, is implicitly reduced and a less flattened ellipsoid will result. Therefore, the multiplicative factor has been called the correlation correction factor, η_{corr}. Only in the rare cases that the minor axis of the error ellipsoid happens to lie along one of the coordinate axes, no appreciable reduction of the flattening results and the method breaks down. Figure 4 clearly shows the reduction of the sensitivity of the Kalman gain matrix to the reference state for increasing values of the correlation correction factor. In the near-singularity region, a 0.01 percent increase of the variances has a tremendous effect on the gain matrix; outside this region, the gain remains almost unchanged. It proved that the application of a correlation correction factor is a most efficient way to stabilize the filter, and is more effective than the iteration process.

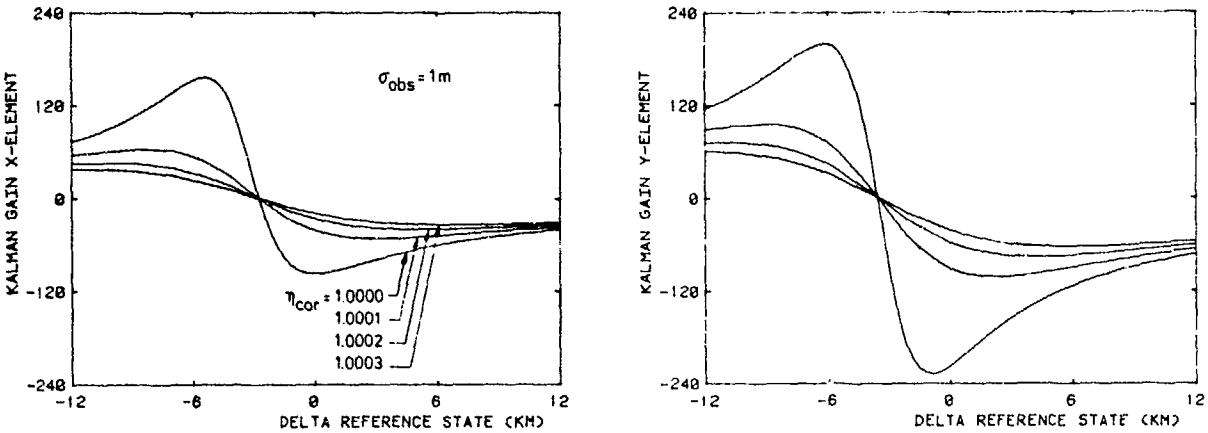


Fig. 4: The dependence of the gain matrix on the reference state for different values of the correlation correction factor.

In SORKA, both the reference state iteration scheme and the correlation correction method are applied, while the possibility to increase the measurement standard deviation has also been implemented. Before each observation-update step, the diagonal elements of the state covariance matrix are multiplied by the correlation correction factor (default 1.00001). The numerical tests have shown that the iterative process then usually converges immediately. As a matter of fact one should be suspicious if this does not occur, because it may indicate that an erroneous observation is being processed, which can upset the filtering process.

The presence of these bad observations is an additional problem, which may complicate the above mentioned divergence suppression methods. Gross measurement errors will corrupt the state estimates, especially when they occur at the beginning of a pass where the state variances are large. To reduce the effects of bad data, a statistical technique has been developed in Ref. 25. It is based on the introduction of a reliability criterion for bounding the propagation of non-detected measurement gross errors into the state vector. The state covariance matrix after the observation-update step is computed such that it is not only a measure for the effects of random errors but also for the effects of a non-detected gross measurement error. So, the filter is forced to make a balance between the precision and the reliability of the updated state vector. This goal is reached by computing the updated state and state covariance matrix according to

$$\hat{\underline{X}} = \underline{X}_1 + \frac{1}{1+\alpha} K_1 \underline{z}_1$$

$$\hat{P}_1 = (I - K_1 H_1) P_1 + \frac{\alpha^2}{(1+\alpha)^2} K_1 H_1 P_1 \qquad (8)$$

The factor α is a function of some statistical parameters and the ratio R/HPH^T. At the start of a pass, the state variances are large and α takes a relatively large value. Consequently, the addition of the α-terms on the right-hand sides of Eqs. (8) will yield a smaller state correction and will blow up the position error ellipsoid in the direction of its minor axis. This is also favorable from the viewpoint of limiting the linearization errors and preventing divergence. As more observations are processed and P becomes smaller, α decreases and the process more closely approximates the optimal Kalman filter scheme. This alternative technique has also been included in SORKA as an option. Tests where only screened data were processed yielded similar results as obtained with the application of the correlation correction factor.

7. DIVERGENCE MONITORING

A proper use of the techniques described in the previous Sections will, in general, lead to a stable Kalman filter behavior when processing laser range observations. Nevertheless, sometimes divergence of the filter may still occur, which results in incorrect and useless state vector estimates. In simulations, divergence can be detected very easily. In those cases, divergence is recognized if the estimated state vectors deviate considerably more from the simulated state vectors, from which the simulated observations are computed, than the estimated state vector standard deviations given by the state covariance matrix. If real observations are processed, such a comparison, of course, is not possible. The only way to detect divergence in real observations processing is to study the history of the observation residuals, $\underline{z}$. The observation residuals relative to the predicted state vector at the time of an observation are used in the observation-update step. The covariance matrix of the residual vector is used in the computation of the Kalman gain matrix. Thus, for divergence detection there is available a sample of a stochastic variable (the residual vector), which, under the assumptions made, has a Gaussian distribution with zero mean and known covariance matrix, and which is not correlated in time.

The two divergence monitoring methods implemented in SORKA are based on testing the

validity of the residual covariance matrix. Therefore, the squared residual is weighted with the covariance matrix. This squared weighted residual is a stochastic variable which should have a chi-squared distribution, with, in case of range-only measurements, one degree of freedom per observation. A method can be developed to test, with a given degree of confidence, if the sum of n squared weighted residuals corresponds indeed to a chi-squared distribution with n degrees of freedom. However, as after a number of processed observations this method will become very slow, in SORKA two faster divergence detection techniques have been included. One method is based on a fading-memory filter, in which the most recent residuals have a greater weight in the sum. The other is based on the digital low-pass filter technique described in Ref. 26, which also results in a test which is more sensitive to the last measurements. Both methods are handicapped if only a few observations during a pass are available, but were found to work satisfactory if many observations are processed. Presently, the methods are only used to monitor if divergence occurs. In the final version of SORKA one of the methods will be linked in a closed-loop mode to correct for divergence as soon as such a tendency is detected.

8. RESULTS

To investigate the capabilities of SORKA in processing range observations from only one tracking station, many simulations have been performed. In these tests simulated observations of GEOS-3 (Table 1) were used. The results of the simulations are treated in detail in Refs. 24, 27, 28. After it had been demonstrated that SORKA is capable of yielding accurate state estimates, numerical experiments were done (Refs. 22, 24, 25, 29) with actual observations of GEOS-1 (Table 1). These laser range observations had been acquired at the Kootwijk tracking station. The data arc, with a length of 54 hour, comprises a total of 611 measurements, distributed over 8 passes during the period July 11 to July 13, 1978, in which GEOS-1 completed 27 revolutions about the earth. In Fig. 5 the sub-satellite points at the times of the observations are plotted. The general direction of the satellite's groundtracks is from west to east. The same observations have also been used in studies described in Refs. 30, 31 to estimate from laser range data acquired at Kootwijk and Wettzell (Fed. Rep. Germany) the orbit of the satellite and the coordinates of the Wettzell tracking station. In those studies the least-squares batch-processing orbit determination and parameter estimation GEODYN computer program (Ref. 32) was used. To be able to judge the quality of the Kalman filter estimates, an orbit solution was generated with GEODYN from the 611 Kootwijk observations. The computed orbital ephemeris, containing state vectors at the observation times and at regular 300 s intervals throughout the complete data span, was stored. Because of the very high accuracies obtainable with GEODYN, that orbit solution could be considered the real-world orbit of the satellite to which the SORKA results are compared. The state vector differences, which are interpreted as the SORKA estimate errors, were subsequently transformed into errors in the classical orbital elements as well as into errors in radial, cross-track and along-track direction. Orbital differences expressed in these parameters suit better the physical interpretation of the results rather than the fast varying rectangular components of the state vector differences.

Table 1: General satellite data.

	GEOS-1	GEOS-3
Satellite number	6508901	7502701
Launch date	November 6	April 9
Shape	octagonal prism with hemispherical cap on down end and octagonal pyramid on top	octagonal prism with radar altimeter dish on down end and octagonal pyramid on top
Dimensions (cm)	132 wide 81 high	122 wide 131 high
Mass (kg)	172.5	345.9
Stabilization	gravity-gradient	gravity-gradient
Transmitters	TRANSIT and MINI-TRACK beacon, SECOR and GRARR transponders.	doppler beacon, C-band and S-band transponders, radar altimeter.
Laser reflectors	322 reflectors on 0.18 m^2 bottom-mounted flat array	264 reflectors in conical ring around the periphery of bottom side
Orbit (mid 1978)		
semi-major axis (km)	8073	7221
eccentricity	0.0717	0.0014
inclination (deg)	59.4	115.0
period (min)	120	102

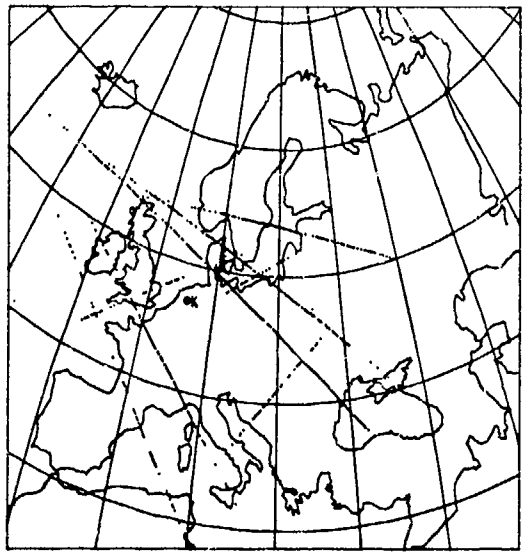


Fig. 5: The GEOS-1 sub-satellite points at the observation times. The general direction of the satellite's groundtracks is from west to east.

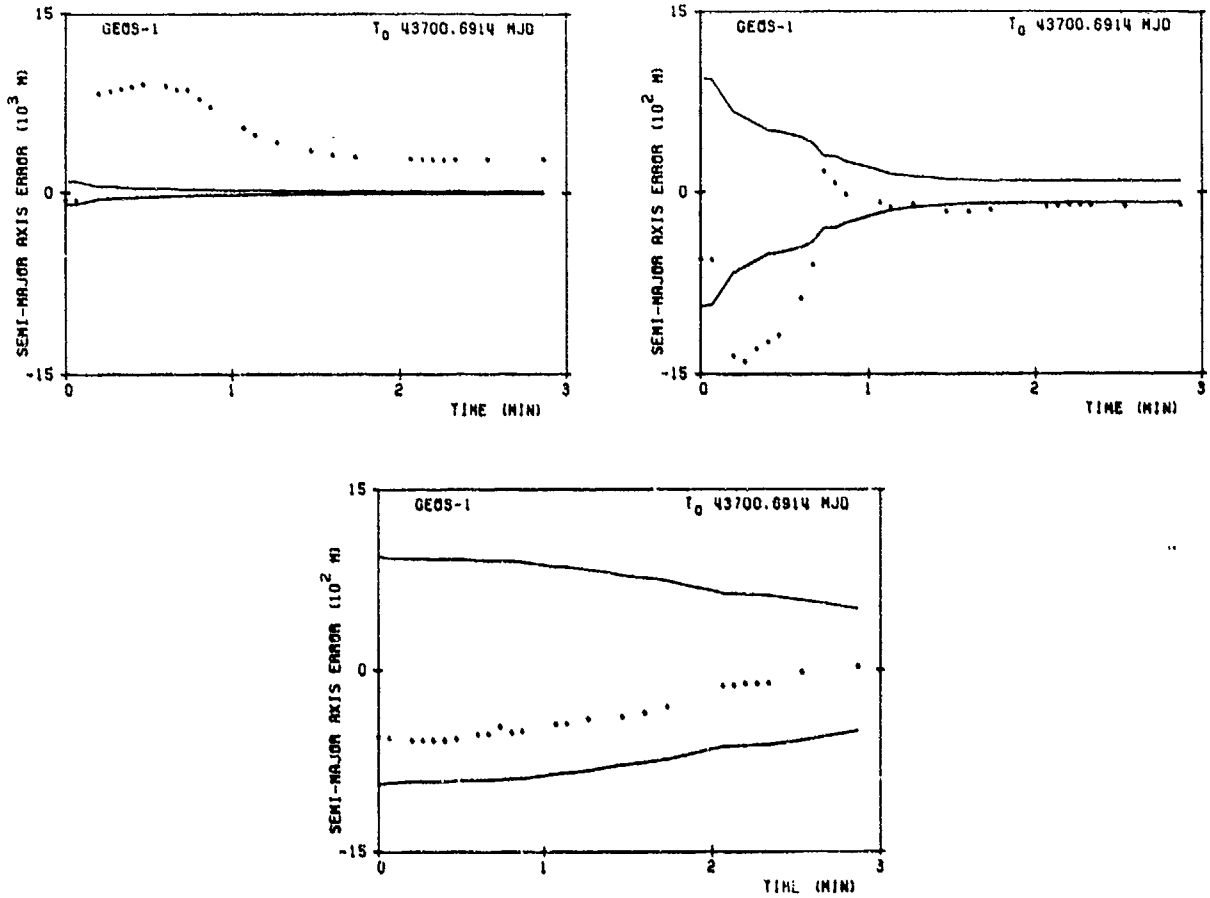


Fig. 6: The semi-major axis errors and standard deviations during the first pass. The state and its covariance matrix are computed by the pure Kalman filter (top left), with the reference state iteration scheme (top right) or with both the iteration scheme and the correlation correction factor (bottom).

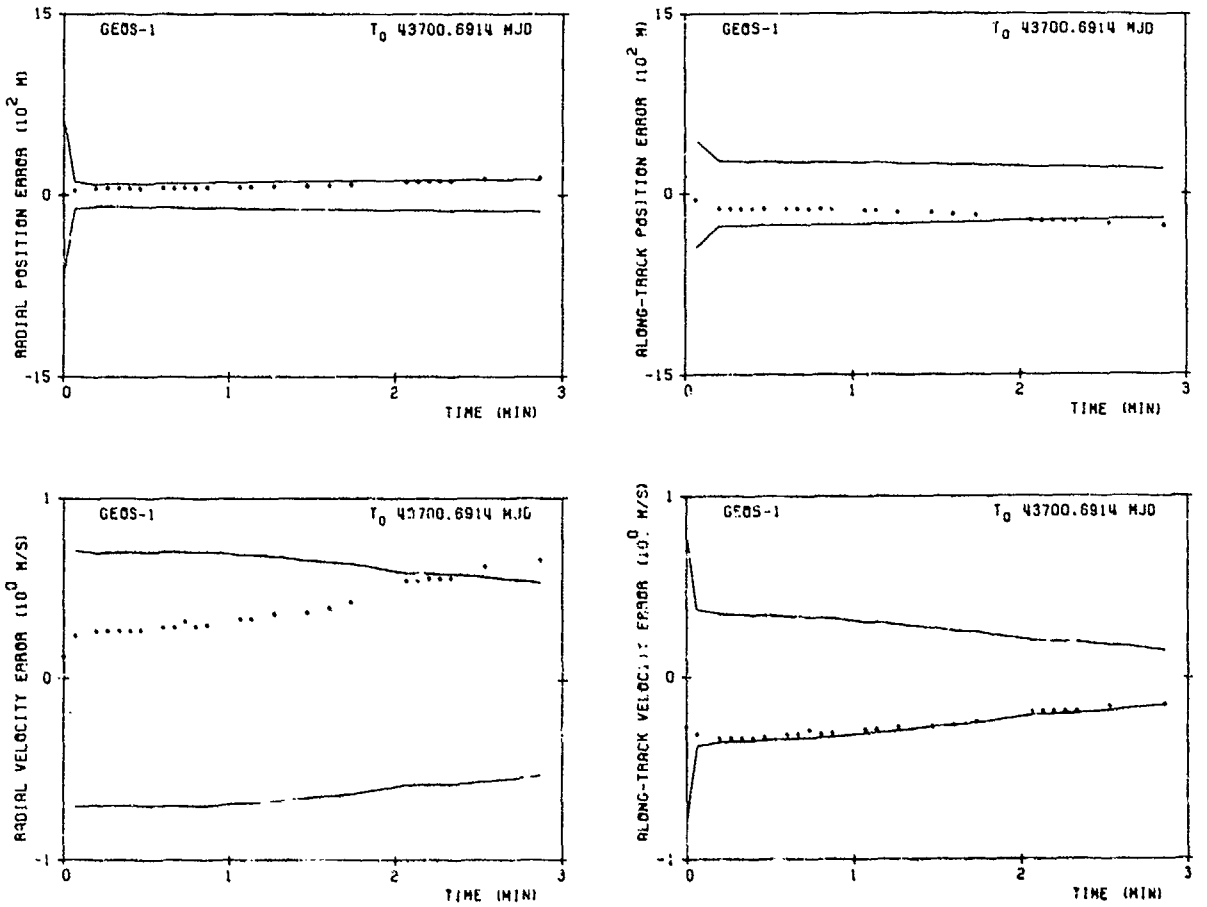


Fig. 7: The radial and along-track state errors and standard deviations during the first pass.

The results presented in this paper refer only to the first three passes, which are separated by two and one revolutions, respectively. In this initial phase of the data are all problems are encountered that are characteristic for processing range observations from a single station, in particular if the initial state is not known accurately. To simulate this last condition, the initial state as computed by GEODYN in rectangular coordinates was corrupted with noise of 1 km standard deviation in the position components and 1 m/s in the velocity components. For the filtering process this corresponds more or less to a worst case analysis since, in general, some state components will be known to a higher accuracy, in particular those which are related to the along-track velocity. The formal standard deviation of the observations, which were known to contain no gross errors, was taken to be 25 cm, corresponding to the accuracy level of the Kootwijk laser system in 1978. The results can be divided into two categories, which are governed by different parameters. The first category pertains only to the first pass. These results are primarily affected by the accuracy of the initial state estimate and by errors due to non-linearities. The two-revolutions data gap between the first and the second pass is a transient region. The second category starts at the beginning of the second pass; these results are primarily influenced by errors in the orbit perturbations modeling.

Figure 6 shows the effects of the reference state iterations and the application of the correlation correction factor, η_{cor}, on the semi-major axis errors during the first pass. The solid lines indicate the standard deviations of the semi-major axis estimates, as provided by the Kalman filter. The value of T_0, indicated in the plots, refers to the origin of the time scale; i.e. the time expressed in Modified Julian Days (MJD) of the first measurement of that pass. The plots show that the basic filter without modifications leads to divergence where semi-major axis errors of 9 km occur, which are much larger than the filter estimates for the standard deviation. The introduction of the reference state iteration scheme yields a much better filter behavior. After reaching a peak of about 1.5 km, the errors decrease rapidly to a level of about 100 m. Applying also a correction factor $\eta_{cor} = 1.00001$ leads to a more stable process where the errors decrease smoother and no sign of divergence is present. Of course, this is achieved at the expense of a slower converging process and larger estimated standard deviations. The evolution of the radial and along-track position and velocity component errors is shown in Fig. 7. These results were obtained from a computation where both the reference state iterations and the correction factor were applied. The radial position error is always less than 140 m; the along-track velocity error decreases to an end-of-pass value of 15 cm/s. The radial velocity error increases to an end-of-pass value of about 65 cm/s, which can be understood when it is realized that observations from a single pass do not yield

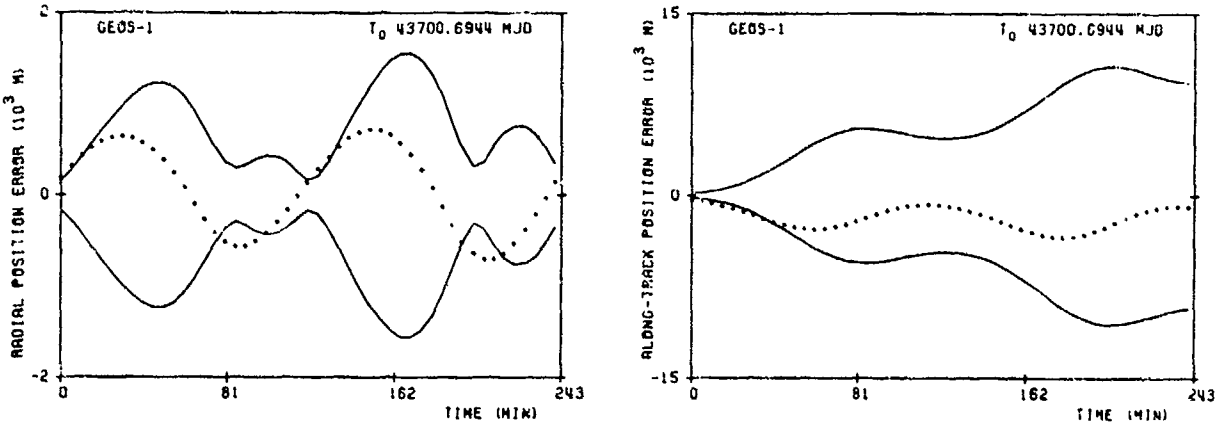


Fig. 8: The radial and along-track position errors and standard deviations during the two-revolutions data gap between the first and the second pass.

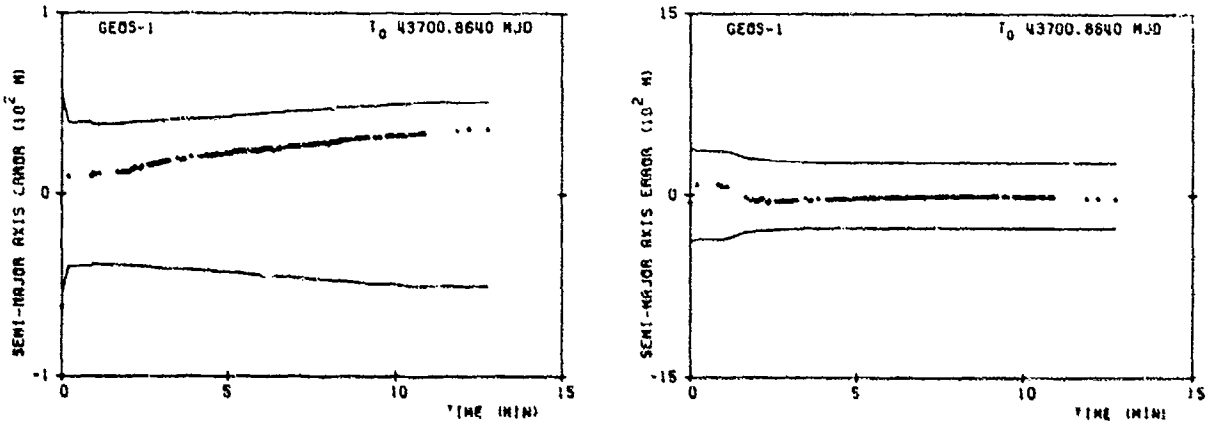
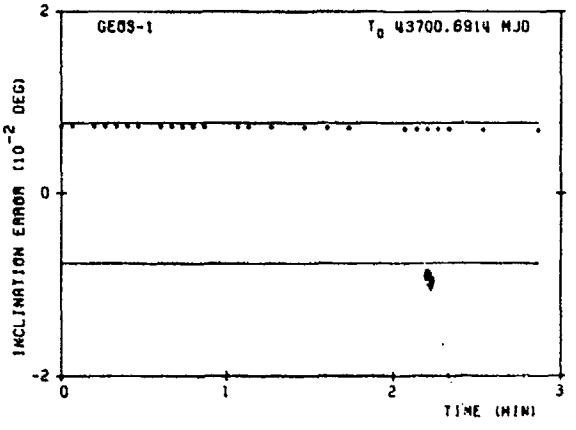


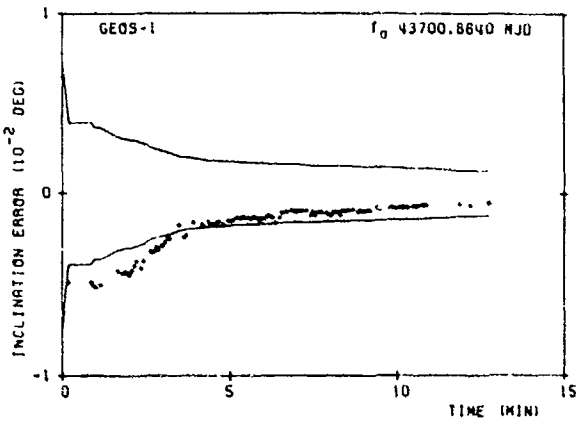
Fig. 9: The effects of the assumed state noise at the end of the two-revolutions data gap on the semi-major axis errors and standard deviations during the second pass. The plot on the left refers to assumed radial, cross-track and along-track position and velocity errors of 20 m and 2 cm/s, respectively; the plot on the right to errors twenty times as large.

sufficient information on the precise shape of the orbit. A positive radial velocity error leads to increasing radial position errors and, according to the equations for the satellite motion, to increasing negative along-track position errors.

The radial and along-track position errors during the two revolutions between the first and the second pass are plotted in Fig. 8. The oscillating along-track position error remains less than 3.5 km during this period and is about 0.9 km at the start of the second pass. The radial position error is about 0.1 km at the start of the second pass and never exceeds 0.8 km during the two revolutions. To compute the standard deviations, the state covariance matrix has been propagated by applying the series expansion technique for Φ and the constant value of 2.5 m/s/day for the computation of Q. From Fig. 8 the along-track position accuracy estimate shows up to be rather conservative. This is partly due to a too pessimistic modeling of the effects of unmodeled perturbations, but for a larger part due to the rather large state variances at the end of the first pass, which in turn are a result of the use of the correlation correction factor during that pass.

Figure 9 depicts for the second pass the effects of the assumptions made for the computation of the state-noise covariance matrix after the two-revolutions data gap. While for all results presented in this paper the state noise after an interval between two passes is computed from the assumption of radial, cross-track and along-track position and velocity errors of 20 m and 2 cm/s, respectively, in Fig. 9 also the results are plotted for state noise values which are twenty times larger. It is clear that the semi-major axis errors during the second pass do not differ considerably for the two assumptions; in both cases the errors are less than 30 m. The estimated state variances, however, are more realistic for the smaller state noise values. That the state component estimates do not differ much in the two cases and that the accuracy estimates hardly vary during the pass is characteristic for the semi-major axis. For the other orbital elements it was found that, generally, the smaller state noise values yield more-accurate estimates and decreasing estimated standard deviations.





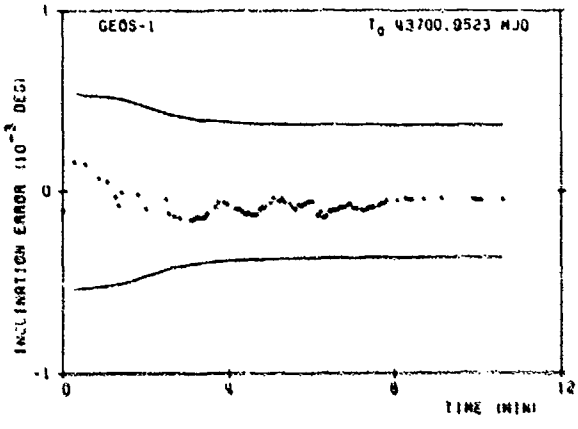
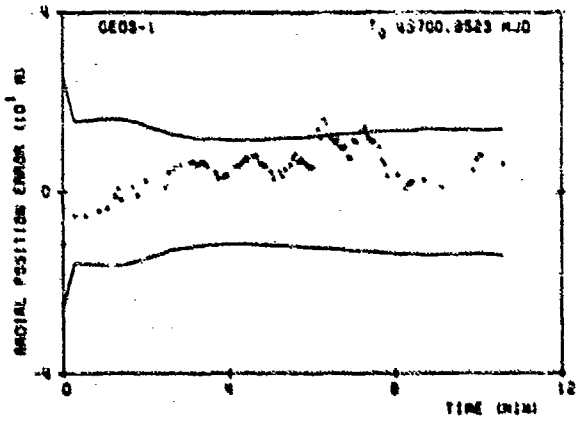
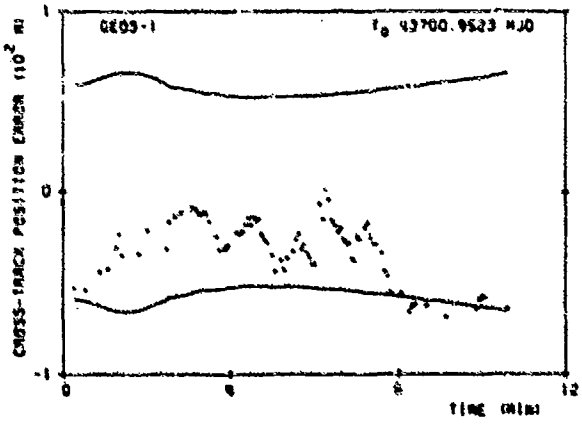


Fig. 10: The inclination errors and standard deviations during the first (top left), second (top right) and third (bottom) pass.





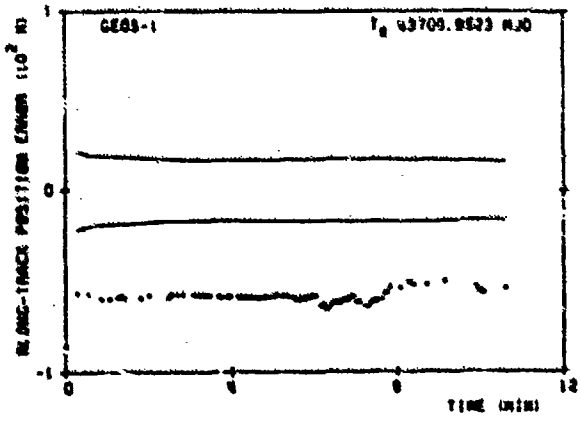


Fig. 11: The radial, cross-track and along-track position errors and standard deviations during the third pass.

The inclination errors during the first three passes are shown in Fig. 10. During the first pass the inclination errors take a nearly constant value of about 0.007° and show no decrease with time. This could be expected because range measurements from a single pass contain almost no information on the orientation of the orbital plane. Consequently, inclination errors are not filtered out easily. At the beginning of the second pass the inclination error still has the same value but after the first observations of the second pass have been processed the error decreases and converges to an end-of-pass value of about 0.0007°. At the end of the third pass the inclination error is even less than 0.00006°.

The total position errors just before the processing of the first observation of the second and third pass, which are primarily errors in cross-track and along-track direction, are 1.4 km and 0.4 km, respectively. Assuming that the distance of GEOS-1 at the start of a pass is 2500 km, a position error of 1.5 km corresponds, depending on the pass geometry, to a topocentric angular error of 1 to 2 arcminutes. For ranging to satellites at altitudes below 2500 km the Kootwijk laser station usually applies a beam divergence of 3 to 10 arcminutes. So, in this example, the Kalman filter position prediction is so accurate that at the first laser firings during the second and third pass the satellite is actually within the laser beam and no satellite search process is required.

The radial, cross-track and along-track position errors during the third pass are shown in Fig. 11. The errors in radial direction are generally less than 15 m, in cross-track direction generally less than 60 m and in along-track direction about 60 m. These accuracies fully satisfy the Kootwijk laser system requirements mentioned in Section 2 for the application of a small beam divergence and a short time window. Interesting is the behavior of the along-track position error. The filter computes standard deviations of about 17 m, while the estimated along-track position contains in reality a nearly constant error of about -60 m. This indicates that, just as shown in Fig. 7 for the first pass, only the first few observations contribute to the improvement of the along-track position component. The computation of the along-track state variances is obviously too optimistic, indicating that a further tuning of filter parameters still has to be performed.

9. CONCLUSIONS

For investigating the possibilities to use laser range observations acquired at the Kootwijk satellite observatory for (semi-) real-time improvement of the predicted satellite positions, a computer program called SORKA has been developed. The main requirements were that SORKA could satisfy both the accuracy level needed for operational laser ranging and the capabilities of a small local computer. Until now, SORKA runs on an IBM 370/158 computer and the implementation on a local computer still has to be studied. In the design of SORKA precautions have been taken to make such an implementation possible.

All tests in applying SORKA to simulated and real observations of GEOS-1 and GEOS-3 look promising and justify the continuation of the efforts to improve the computations scheme such that it best suits the laser ranging system characteristics. It has been demonstrated that a stable Kalman filter process can be obtained when using laser range measurements from one tracking station. It was found that divergence, which very easily may occur when processing only laser range measurements, can effectively be suppressed by applying a correlation correction factor and an iteration scheme in the observation-update step.

10. REFERENCES

1. Aardoom, L., Delft Univ. Technology, Dept. Geodesy, Kootwijk SLR station status, 1980, paper presented at the third LAGEOS Working Group meeting, NASA Goddard Space Flight Center.
2. Aardoom, L. and Zeeman, P.W., The satellite ranging system at Kootwijk, in: Proceedings of laser workshop, National Technical Univ., Athens, 1978, pp. 89-96.
3. Ambrosius, B.A.C., Piersma, H.J.D. and Wakker, K.F., Delft Univ. Technology, Dept. Aerospace Engin., Description of the AIMLASER satellite orbit prediction program and its implementation on the Delft University IBM computer, 1976, Report LR-218.
4. de Groot, W.A., Delft Univ. Technology, Dept. Aerospace Engin., Investigation on the accuracy of the SAO orbital elements (in Dutch), 1978.
5. Jazwinski, A.H., Stochastic processes and filtering theory, New York, Academic Press, 1970, pp. 272-281.
6. Williams, D.E., Royal Aircraft Establ., Farnborough, A geometric derivation of the Kalman filter equations, 1978, Technical Report 78122.
7. Rogers, C.E. and Christodoulou, C., Wright-Patterson AFB, The extended Kalman filter applied to the determination of the orbital parameters of a passive earth satellite, Thesis Air Force Inst. Technology, 1971.
8. Tapley, B.D. and Schutz, B.E., A comparison of orbit determination methods for geodetic satellites, in: Proc. of first international symposium on the use of artificial satellites for geodesy and geodynamics, Athens, 1973, pp. 523-562.
9. Tapley, B.D., Statistical orbit determination theory, in:Recent advances in dynamical astronomy, Dordrecht, Reidel, 1973, pp. 396-425.
10. Schutz, B.E., McMillan, J.D. and Tapley, B.D., A comparison of estimation methods for the reduction of laser observations of a near-earth satellite, 1973, paper presented at AAS/AIAA astrodynamics conference, Vail.

11. Torroglosa, V., ESA, Paris, Filtering theory applied to orbit determination, 1973, ESRO CR(P)-532.
12. Winn, C.B., Optimal estimation of unmodelled accelerations on the ONERA navigational satellite, J. Spacecraft, 12, 2, 1975, pp. 79-82.
13. Thornton, C.L. and Bierman, G.J., A numerical comparison of discrete Kalman filtering algorithms: An orbit determination case study, 1976, IAF-76-015, 27th IAF congress, Anaheim.
14. Dunham, J.B., Autonomous satellite orbit determination during development phases of the global positioning system, in: Flight mechanics/estimation theory symposium, 1979, NASA Conf. Publ. 2123, pp. 137-156.
15. Rice, D.R., An investigation into the effects of using simplified force models in the computation of state transition matrices, 1967, AIAA Paper No. 67-123, New York.
16. May, J.A., A study of the effects of state transition matrix approximations, in: Flight mechanics/estimation theory symposium, 1979, NASA Conf. Publ. 2123, pp. 29-48.
17. van Hulzen, J.J.P., Delft Univ. Technology, Dept. Aerospace Engin., Preliminary study on the application of a Kalman filter to the orbit determination of satellites (in Dutch), 1978.
18. Cornelisse, J.W., Schöyer, H.F.R. and Wakker, K.F., Rocket propulsion and spaceflight dynamics, London, Pitman, 1979, pp. 377-388.
19. Kamp, A., Delft Univ. Technology, Dept. Aerospace Engin., Analytical partial derivatives for a perturbed Keplerian orbit (in Dutch), Technical thesis, 1979.
20. Merson, R.H., Royal Aircraft Establ., Farnborough, The dynamical model of PROP, a computer program for the refinement of the orbital parameters of an earth satellite, 1966, Technical Report 66255.
21. Kozai, Y., The motion of a close earth satellite, Astron. J., 64, 1274, 1959, pp. 367-377.
22. Ekkebus, E.J., Delft Univ. Technology, Dept. Aerospace Engin., Problems in attaining accurate state estimates from an extended Kalman filter, processing laser range-only observations from a single tracking station, Technical thesis, 1981.
23. Wright, J.R., Sequential orbit determination with auto-correlated gravity modeling errors, 1980, Paper AIAA-80-0239, Pasadena.
24. van Hulzen, J.J.P., Delft Univ. Technology, Dept. Aerospace Engin., Kalman filter satellite orbit determination using laser range observations by a single station, Thesis advanced study, 1980.
25. Vermeer, M., Delft Univ. Technology, Dept. Geodesy, Kalman filter orbit determination for geodetic satellite laser ranging; a theoretical inquiry, Technical thesis, 1981.
26. Traas, C.R., National Aerospace Laboratory, Amsterdam, Digital filtering methods, with applications to satellite attitude determination in the presence of modelling errors, Part 1: Theory, 1976, NLR-TR-76048C, Part 2: Extension of theory and applications, 1979, NLR-TR-79039L.
27. van Hulzen, J.J.P., Delft Univ. Technology, Dept. Aerospace Engin., Kalman filter satellite orbit determination from simulated laser range observations from one groundstation (in Dutch), Technical thesis, 1979.
28. Kamp, A. and van Hulzen, J.J.P., Delft Univ. Technology, Dept. Aerospace Engin., State transition matrix computations for Kalman filter orbit determination from laser range observations, 1979, paper presented at the 30th IAF congress, Munich; also: Memorandum M-363, 1980.
29. Wakker, K.F., Ambrosius, B.A.C. and van Hulzen, J.J.P., Kalman filter orbit improvement from Kootwijk laser range observations, Adv. Space Res., 1, 1981, pp. 79-82.
30. Wakker, K.F. and Ambrosius, B.A.C., Delft Univ. Technology, Dept. Aerospace Engin., A study on the determination of the Kootwijk-Wettzell baseline from satellite laser ranging at these stations, 1980, paper submitted to the 2nd meeting of the LAGEOS Working Group, NASA Goddard Space Flight Center; also: Report LR-295.
31. Wakker, K.F. and Ambrosius, B.A.C., Delft Univ. Technology, Dept. Aerospace Engin., Estimation of the Wettzell coordinates from satellite laser ranging at Kootwijk, San Fernando and Wettzell, 1980, paper submitted to the EROS plenary meeting, Grasse; also: Report LR-296.
32. Martin, T.V., Washington Analytical Services Center, Riverdale, GEODYN descriptive summary, contract no. NAS 5-22849, 1978.

ACKNOWLEDGEMENTS

This study would not have been possible without the participation of our orbit mechanics graduate and post-graduate students E.J. Ekkebus, J.J.P. van Hulzen and A. Kamp. Their contributions are gratefully acknowledged. The graduate student in satellite geodesy M. Vermeer contributed by developing a technique to bound the propagation of gross measurement errors into the orbit solution. We also wish to express our thanks to H.L. Jonkers and H. Leenman of our Department for the many valuable discussions. Support was provided by student grants from the Ministry of Education and Science and by the Delfts Hogeschoolfonds.

STATE ESTIMATION OF BALLISTIC TRAJECTORIES WITH ANGLE ONLY MEASUREMENTS

by
Michael R. Salazar
Scientist
Nichols Research Corporation
4040 So. Memorial Parkway
Huntsville, Alabama
35802

SUMMARY

This report presents two methods for determining the optimum state estimate for the angle only ballistic trajectory problem.

The first method utilizes the techniques of Marquardt matrix conditioning and explicit Jacobian in determining the weighted least squares state estimate of a ballistic trajectory. This Marquardt least squares (MLS) algorithm is a batch or nonrecursive process. A description of a similar least squares batch technique is included in this report for a better understanding and a means of evaluation by comparison.

The second method applies the explicit Jacobian technique to the recursive Kalman filter equations. This improved Jacobian-Kalman filter formulation when combined with the MLS batch process for initialization forms the desired total angle only tracking algorithm.

NOTATION

A	Measurement matrix
Δt	Time between measurements
ΔX	Deviations of state from nominal set
$\hat{\Delta X}$	Estimate of Δx
ΔY	Deviations of measurements from nominal or calculated set ($\hat{Y}$)
E, E_T	Total energy
ECI	Earth centered inertial coordinate system
ϵ	Measurement errors
f, g	The f and g series values
f_1, f_2, f_3	State vector velocity components
f_4, f_5, f_6	Time derivative of state vector velocity components
G	Gravitational constant
g_1, g_2	Azimuth, elevation measurements
H, H'	Kalman gain matrix
I	Identity matrix
J	Jacobian matrix
KE	Kinetic energy
λ	Marquardt matrix conditioning factor
M	Mass of earth
μ	$G \times M$
PE	Potential energy
Φ	State transition matrix
R	Range
$\dot{R}$	Range rate
$\underline{R}$	Range vector
R_e	Radius of earth

R_S	Center of earth to sensor radius
R_T	Center of earth to target radius
R_x, R_y, R_z	The x,y,z components of $\underline{R}$
$\underline{S}$	Sensor position vector
S, S_1^{-1}	State covariance matrix
S_x, S_y, S_z	The x,y,z components of $\underline{S}$
SCT	Sensor centered topographic coordinate system
σ_E	One sigma uncertainty in E
$\sigma_\theta, \sigma_\phi$	One sigma measurement uncertainty in θ, ϕ
$\underline{T}$	Target position vector
$\underline{T}_0$	Initial target position vector
T_x, T_y, T_z	The x,y,z components of $\underline{T}$
$t_n, t_{n-1} \cdots t_{n-L}$	Time for each measurement
τ	Time step for f and g series
θ, ϕ	Azimuth, elevation measurements
$\dot{\theta}, \dot{\phi}$	First time derivative of θ, ϕ
$\ddot{\theta}, \ddot{\phi}$	Second time derivative of θ, ϕ
u_0, p_0, q_0	The f and g series terms
$\underline{V}_0$	Initial target velocity vector
$\underline{V}_R$	Relative velocity vector (target-sensor)
V_S	Sensor velocity magnitude
$\underline{V}_S$	Sensor velocity vector
V_T	Target velocity magnitude
$\underline{V}_T$	Target velocity vector
W	Measurement covariance matrix
$\dot{X}$	Time derivative of X
$\hat{X}$	Estimate of X
$\underline{XS}$	Sensor state vector
$X, \underline{X}$	State vector
x, y, z	State vector position components
$\dot{x}, \dot{y}, \dot{z}$	State vector velocity components
x_1, x_2, x_3	State vector position components
x_4, x_5, x_6	State vector velocity components
$\dot{x}_1, \dot{x}_2, \dot{x}_3$	Time derivative of state vector position components
$\dot{x}_4, \dot{x}_5, \dot{x}_6$	Time derivative of state vector velocity components
$\dot{x}_R, \dot{y}_R, \dot{z}_R$	The x,y,z components of $\underline{V}_R$
$\dot{x}_S, \dot{y}_S, \dot{z}_S$	The x,y,z components of $\underline{V}_S$
χ^2	Chi-squared
$\hat{Y}$	Estimate of Y
$Y, \underline{Y}$	Measurement vector
*	Normalized

1. INTRODUCTION

This report deals with the problem of state estimation of ballistic trajectories with angle only measurements. This type of problem becomes difficult when the observer is free-falling and more difficult if the observer is then located in the plane of the observed trajectory. The methods described in this report are very effective against this most difficult case, and superior to existing angle only tracking filters in terms of stability, computational requirements, and tracking performance.

The first method described herein utilizes the methods of Marquardt matrix conditioning[1] and explicit Jacobian in determining the weighted least squares state estimate for the nonlinear, time varying, dynamic system of a ballistic trajectory with nonlinear noisy measurements. In this case both the equations of motion and the angle observations are nonlinear functions of the state. This Marquardt least squares (MLS) technique is a nonrecursive or batch process in that all the observations must be processed each time a state estimate is made. The method of incorporating a priori knowledge of the total energy into the MLS algorithm to assist in poor observability problems is also discussed.

The second method takes the explicit Jacobian technique developed for the MLS algorithm and applies it to the recursive Kalman filter equations. This improved Jacobian-Kalman filter formulation together with the MLS for initialization form the complete angle only tracking algorithm.

This report is organized as follows. Section 2 provides the background material and the step by step development of the equations for the standard weighted least squares batch filter and MLS algorithms. A discussion of the energy constraint concept and the application of the explicit Jacobian method to the Kalman filter is included in this section. Subsection 2.1 presents the weighted least squares solution for the general nonlinear system problem. This serves as a common basis for both the standard weighted least squares batch filter and the MLS algorithms which are developed in subsections 2.2 and 2.3, respectively. No attempt is made to make these derivations mathematically rigorous. The energy constraint concept is discussed in subsection 2.4. The application of the explicit Jacobian technique to the Kalman filter equations is presented in subsection 2.5. Section 3 gives the performance results for several ballistic trajectory problem test cases. Section 4 contains the conclusions.

2. ALGORITHM DESCRIPTION

2.1 Weighted Least Squares Concept

In general, the fundamental problem of concern can be stated as follows. A nonlinear system can be represented by the linearized model matrix equation:

$$\Delta Y = A\Delta X + \varepsilon \tag{2-1}$$

where ΔY is an $n \times 1$ matrix of the deviations of the observations (Y) from the nominal or calculated set ($\hat{Y}$), ΔX is a $k \times 1$ matrix of deviations of the unknown parameters from a nominal (known) set, A is an $n \times k$ ($n \geq k$) known matrix, and ε is an $n \times 1$ matrix of observation errors (unknown). The problem is: given ΔY and A and the linearized model of Eq. (2-1), find the "best" estimate of ΔX called $\hat{\Delta X}$. Once the "best" estimate $\hat{\Delta X}$ has been determined, the "best" estimate of the unknown parameters $\hat{X}$ may be determined by

$$\hat{X} = X + \hat{\Delta X} \tag{2-2}$$

where X is the known or nominal set of parameters. In this case the "best" estimate is acheived when the weighted least squares criterion is satisfied; that is, when the sum of the squares of the components of the weighted residual vector is minimized. This quantity can be represented by:

$$\Delta Y^T W^{-1} \Delta Y \tag{2-3}$$

where W is the known $n \times n$ measurement covariance matrix. This weighting matrix accounts for the difference in confidence between various observations and the possible correlation between them.

The well known weighted least squares solution (formulated by Gauss in 1794*) to this problem is

$$\hat{\Delta X} = (A^T W^{-1} A)^{-1} A^T W^{-1} \Delta Y \tag{2-4}$$

which combined with Eq. (2-2) yields the total expression

$$\hat{X} = X + (A^T W^{-1} A)^{-1} A^T W^{-1} \Delta Y \quad . \tag{2-5}$$

Given some initial guess of X, this expression is normally iterated as follows until the current parameter estimates do not vary appreciably from the previous iteration.

---

*Formulation included only the diagonal terms of W.

Evaluate A with X
Calculate $\hat{Y}$ for X
$\Delta Y = Y - \hat{Y}$
$\hat{X} = X + (A^T W^{-1} A)^{-1} A^T W^{-1} \Delta Y$
$X = \hat{X}$

Eq. (2-5) establishes the basis for both the standard weighted least squares batch filter and the Marquardt least squares algorithms. The following sections will show how these two different formulations are derived from this common expression thus demonstrating both their similarities and differences.

2.2 Standard Weighted Least Squares Batch Filter

For a given set of observation times $t = t_n, t_{n-1} \cdots t_{n-L}$ the linearized model represented by Eq. (2-1) can be expanded to give

$$\begin{pmatrix} \Delta Y_n \\ \Delta Y_{n-1} \\ \vdots \\ \Delta Y_{n-L} \end{pmatrix} = \begin{pmatrix} A_n \Delta X_n \\ A_{n-1} \Delta X_{n-1} \\ \vdots \\ A_{n-L} \Delta X_{n-L} \end{pmatrix} + \begin{pmatrix} \varepsilon_n \\ \varepsilon_{n-1} \\ \vdots \\ \varepsilon_{n-L} \end{pmatrix} . \tag{2-6}$$

For a time-varying "linear" differential equation model, that is,

$$\frac{d}{dt} \Delta X(t) = F\big(X(t)\big) \; \Delta X(t) \tag{2-7}$$

the "approximate" solution is given by

$$\Delta X(t + \Delta t) = \Delta X(t) + \Delta t \; F\big(X(t)\big) \; \Delta X(t) \quad . \tag{2-8}$$

Factoring out $\Delta X(t)$

$$\Delta X(t + \Delta t) = \Big\{ I + \Delta t \; F\big(X(t)\big) \Big\} \; \Delta X(t) \tag{2-9}$$

which yields the following transition relation

$$\Delta X_{n-1} = \Phi_{n-1,n} \; \Delta X_n \quad . \tag{2-10}$$

Note that two approximations were required in arriving at this last expression: (a) the linear differential equation model and (b) its approximate solution. The significance of this will become apparent in the development of the Marquardt weighted least squares algorithm in subsection 2.3. Using Eq. (2-10), Eq. (2-6) can be written as

$$\begin{pmatrix} \Delta Y_n \\ \Delta Y_{n-1} \\ \vdots \\ \Delta Y_{n-L} \end{pmatrix} = \begin{pmatrix} A_n & \Delta X_n \\ A_{n-1} & \Phi_{n-1,n} \quad \Delta X_n \\ \vdots & \vdots \\ A_{n-L} & \Phi_{n-L,n} \quad \Delta X_n \end{pmatrix} + \begin{pmatrix} \varepsilon_n \\ \varepsilon_{n-1} \\ \vdots \\ \varepsilon_{n-L} \end{pmatrix} . \tag{2-11}$$

Now factor out ΔX_n to yield

$$\begin{pmatrix} \Delta Y_n \\ \Delta Y_{n-1} \\ \vdots \\ \Delta Y_{n-L} \end{pmatrix} = \begin{pmatrix} A_n \\ A_{n-1} \Phi_{n-1,n} \\ \vdots \\ A_{n-L} \Phi_{n-L,n} \end{pmatrix} \Delta X_n + \begin{pmatrix} \varepsilon_n \\ \varepsilon_{n-1} \\ \vdots \\ \varepsilon_{n-L} \end{pmatrix} . \tag{2-12}$$

From Eq. (2-12) the following matrix is defined

$$J_n = \begin{pmatrix} A_n \\ A_{n-1} \Phi_{n-1,n} \\ \vdots \\ A_{n-L} \Phi_{n-L,n} \end{pmatrix} . \tag{2-13}$$

Eq. (2-11) can now be written in the compact form

$$\Delta Y_{(n)} = J_n \; \Delta X_n + \varepsilon_{(n)} \quad . \tag{2-14}$$

The weighted least squares solution for this linearized model can be written as an extension of Eq. (2-4) to give

$$\hat{\Delta X}_n = \left(J_n^T W_{(n)}^{-1} J_n\right)^{-1} J_n^T W_{(n)}^{-1} \Delta Y_{(n)} \quad . \tag{2-15}$$

Similarly, the final expression becomes

$$\hat{X}_n = X_n + \left(J_n^T W_{(n)}^{-1} J_n\right)^{-1} J_n^T W_{(n)}^{-1} \Delta Y_{(n)} \tag{2-16}$$

which can be iterated by the procedure described in subsection 2.1. While the iterative solution for this expression essentially represents the standard weighted least squares batch filter, a few more definitions are required before the final algorithm can be presented.

For the ballistic trajectory problem, the state (parameter) vector may be expressed in terms of an earth-centered inertial Cartesian coordinate system

$$\underline{X} = \begin{pmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} \quad . \tag{2-17}$$

For a spherical earth the exoatmospheric trajectory equations of motion are

$$\dot{\underline{X}} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{pmatrix} = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ -\dfrac{GMx_1}{R^3} \\ -\dfrac{GMx_2}{R^3} \\ -\dfrac{GMx_3}{R^3} \end{pmatrix} \tag{2-18}$$

where G is the gravitational constant, M is the earth's mass, and R is the magnitude of the position vector.

The first order Taylor series approximation for the transition matrix Φ defined in Eqs. (2-8), (2-9), and (2-10) is

$$\begin{aligned} \Phi_{n-1,n} &= I + F\left(X(t_n)\right)\ (t_{n-1} - t_n) \\ \Phi_{n-2,n} &= I + F\left(X(t_n)\right)\ (t_{n-2} - t_n) \\ &\vdots \\ \Phi_{n-L,n} &= I + F\left(X(t_n)\right)\ (t_{n-L} - t_n) \end{aligned} \tag{2-19}$$

where the $F\left(X(t_n)\right)$ matrix is defined by

$$\left[F\left(X(t_n)\right)\right]_{i,j} = \left.\frac{\partial f_i}{\partial x_j}\right|_{X=X(t_n)} = \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_6} \\ \vdots & & \vdots \\ \dfrac{\partial f_6}{\partial x_1} & \cdots & \dfrac{\partial f_6}{\partial x_6} \end{pmatrix}_{X=X(t_n)} \tag{2-20}$$

where f_i are the derivative functions of the state vector ($\dot{\underline{X}} = \underline{f}(X)$ defined in Eq. (2-18). The partial derivatives for Eq. (2-20) are given in Appendix A. A more accurate method for determining the transition matrix is to use the transition matrix determined for the previous observation time in determining the transition matrix for the current observation time. This reduces the time interval over which the transition matrix must be valid and allows for evaluation of the F matrix with the updated nominal state. This method produces the following set

$$\Phi_{n-1,n} = I + F\left(X(t_n)\right)\ (t_{n-1} - t_n)$$
$$\Phi_{n-2,n} = \left\{I + F\left(X(t_{n-1})\right)\ (t_{n-2} - t_{n-1})\right\}\ \Phi_{n-1,n}$$
$$\vdots$$
$$\Phi_{n-L,n} = \left\{I + F\left(X(t_{n+1-L})\right)\ (t_{n-L} - t_{n+1-L})\right\}\ \Phi_{n+1-L,n+2-L} \qquad (2\text{-}21)$$

By this method the magnitude of the time interval is restricted to the time between observations. If this time interval is still too large, this propagation technique can be further applied by subdividing the time step between observations. For example, if there are m subdivided time steps of h then the first expression in Eq. (2-21) would be

$$\Phi_{n-1,n} = \left\{I + hF\left(X(t_n)\right)\right\}\ \left\{I + hF\left(X(t_n - h)\right)\right\}..\left\{I + hF\left(X(t_n - (m-1)h)\right)\right\} \qquad (2\text{-}22)$$

The angle-only observation set for the ballistic trajectory problem can be defined as

$$\underline{Y} = \begin{pmatrix}\theta\\ \phi\end{pmatrix} = \begin{pmatrix}g_1\\ g_2\end{pmatrix} = \begin{pmatrix}\tan^{-1}\dfrac{Rx}{Ry}\\ \tan^{-1}\dfrac{Rz}{\sqrt{R_x^2 + R_y^2}}\end{pmatrix} \qquad (2\text{-}23)$$

where Rx, Ry, Rz is the relative position vector (target-sensor) in a topographic coordinate system which is defined in Appendix B. The first order Taylor series approximation for the measurement matrix A_n defined in Eq. (2-6) is

$$\left[A_n\right]_{i,j} = \left[A\left(X(t_n)\right)\right]_{i,j} = \frac{\partial g_i}{\partial x_j}\bigg|_{X=X(t_n)} = \begin{bmatrix}\dfrac{\partial g_1}{\partial x_1} & \cdots & \dfrac{\partial g_1}{\partial x_6}\\ \dfrac{\partial g_2}{\partial x_1} & \cdots & \dfrac{\partial g_2}{\partial x_6}\end{bmatrix}_{X=X(t_n)} \qquad (2\text{-}24)$$

where g_i are the functions relating observations to states $\underline{Y} = \underline{g}(X)$ defined in Eq. (2-23). The remaining set $A_{n-1}\ldots A_{n-L}$ are obtained in similar fashion using $X(t_{n-1})\ldots X(t_{n-L})$. The partial derivatives in Eq. (2-24) are given in Appendix B.

The weighting or measurement covariance matrix W, assumed in this study, is

$$[W]_{i,j} = \begin{bmatrix}\sigma_\phi^2 & 0\\ 0 & \sigma_\phi^2\end{bmatrix} \qquad (2\text{-}25)$$

where σ_θ and σ_ϕ represent the one sigma uncertainties in the uncorrelated measurement set (θ,ϕ).

One final step is required before the final algorithm can be presented. Instead of building the entire matrix J_n defined in Eq. (2-13), Eq. (2-16) may be rewritten in the following form

$$\hat{X}_n = X_n + \left\{\sum_{i=1}^{L+1} J_n^T(i) W_{n+1-i}^{-1} J_n(i)\right\}^{-1} \left\{\sum_{i=1}^{L+1} J_n^T(i)\ W_{n+1-i}^{-1}\ \Delta Y_{n+1-i}\right\} \qquad (2\text{-}26)$$

where $J_n(1) = A_n$; $J_n(2) = A_{n-1}\Phi_{n-1,n}$; ...; $J_n(L+1) = A_{n-L}\Phi_{n-L,n}$.

All definitions have now been given for the standard weighted least squares batch filter algorithm which is presented in Figure 1 using the notation defined in this section. This is essentially the algorithm described by Lincoln Laboratory[2] and is basically the Gauss weighted least squares solution defined for the ballistic trajectory problem. The formulation presented in Figure 1 requires at least three pairs of angle observations or three cycles of the measurement loop. For this formulation the nominal state vector is defined at the final observation point.

In this algorithm some initial state guess is successively corrected in an attempt to satisfy the weighted least squares criterion. For observations at times $t_n, t_{n-1}, \ldots t_{n-L}$ this quantity is

$$\sum_{i=1}^{L+1} \left(Y_{n+1-i} - \hat{Y}_{n+1-i}\right)^T W_{n+1-i}^{-1} \left(Y_{n+1-i} - \hat{Y}_{n+1-i}\right) \qquad (2\text{-}27)$$

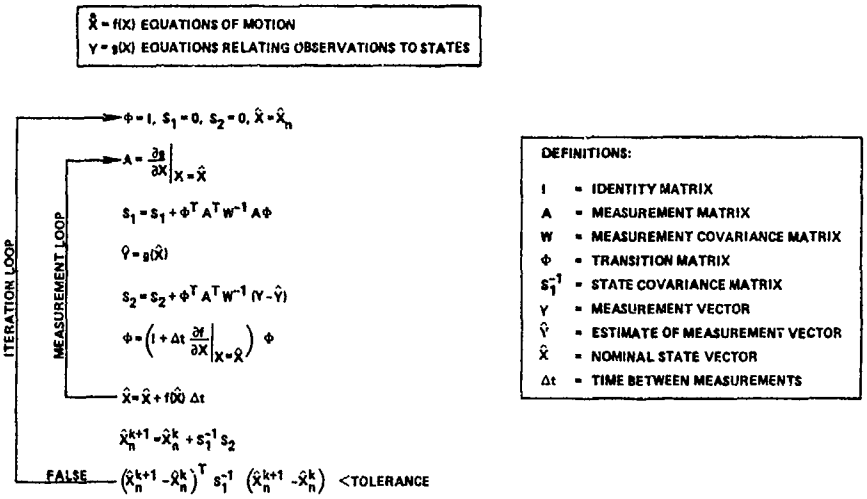


Fig. 1 Standard weighted least squares batch filter algorithm

If the process is converging, this quantity becomes increasingly smaller at a decreasing rate as it asymptotically approaches its minimum value. The process could thus be terminated when this quantity does not decrease significantly from the previous iteration. Another criterion for terminating the iterative process is the one shown in Figure 1

$$\left(\hat{X}_n^{k+1} - \hat{X}_n^k\right)^T S_1^k \left(\hat{X}_n^{k+1} - \hat{X}_n^k\right) \quad . \tag{2-28}$$

Note that in this quantity the measurement residual has been replaced by the difference in the nominal state vectors for successive iterations and the measurement covariance by the state covariance. If the process is converging, this quantity approaches zero. Therefore, termination of the iterative process would be based on the magnitude of this quantity rather than a change in magnitude as in the previous method. In either case the cutoff point is established at the level at which tracking performance is unaffected.

Because the weighted least squares batch filter is a batch or nonrecursive process, all of the observations must be processed each time a new measurement pair is added to the observation set. In addition to this requirement, all the measurements of the observation set must be processed for each iteration of the least squares process as indicated in Figure 1. Therefore, termination of the iterative process at the earliest possible point is essential. The nonrecursive characteristic of the batch filter and resulting processing requirements tend to restrict this algorithm to the initialization of track function. For the initialization of track function, the resulting weighted least squares state estimate and state covariance are projected forward to the next observation point where they serve as the initial state and covariance for a Kalman filter algorithm. The weighted least squares algorithm can also be applied again at this point using these same projected quantities as initial conditions.

In Section 3 it will be shown that this algorithm does not always converge. Therefore, a series of tests are made after each iteration to determine if the process is converging. The algorithm is determined to be nonconvergent if any of the following tests are true:

1) Too many iterations (15)
2) Position magnitude too large (1.5×10^7 m)
3) Velocity magnitude too large (1.0×10^4 m/sec)
4) Altitude too low (100 m).

The values listed are possible limits for these tests. If the process is nonconvergent, then the pair of angle measurements for the next observation point are added to the observation set and the weighted least squares algorithm is performed again.

Thus far, the problem of obtaining an initial state guess to begin the iterative least squares process has not been addressed. One possible method for obtaining this initial state guess is the one described by Lincoln Laboratory[1] and presented in Figure 2. For briefness, the derivation of this method is given in Appendix C. In this method the azimuth and elevation are fit with a second order polynomial for a batch of angle measurement data (at least three pairs). The azimuth and elevation along with their first and second derivatives are solved for with the polynomial fit at the time of the last observation. These quantities are then used in an iterative set of energy and geometry equations to solve for the range and range rate, which are in turn used to estimate the initial guess of the state vector for the weighted least squares process.

The iterative energy/geometry equations used to establish the range and range rate require an initial guess of the range and an a priori estimate of the target energy. This set of equations is cycled until the new estimate of the range does not vary from the previous estimate by more than some given tolerance. The selection of energy as a constraint for the initial state guess is made because of its relative constancy over the whole trajectory for a given set of ICBM threats with the same ground range. While the initial state

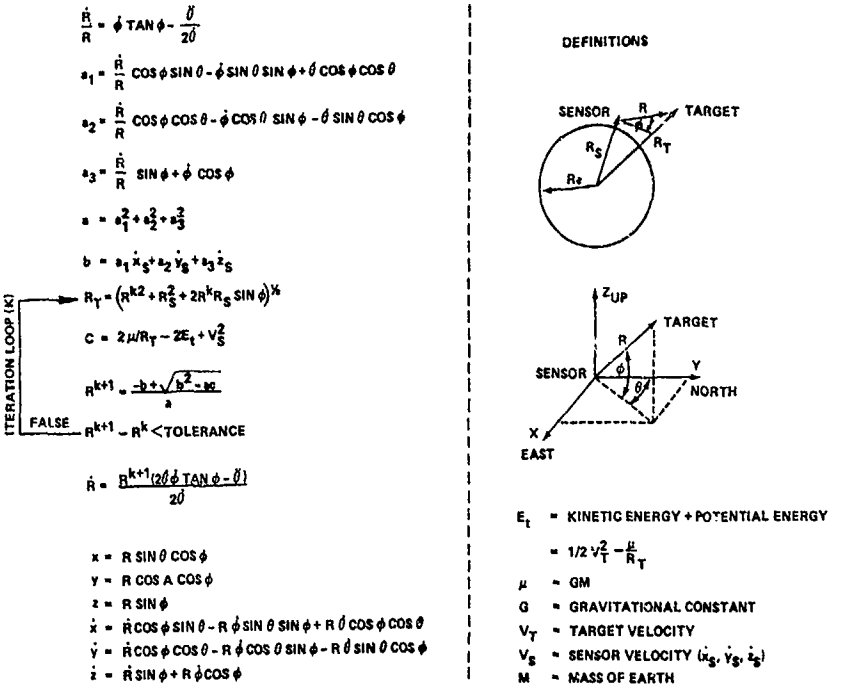


Fig. 2 Initial state guess algorithm

guess from this process is constrained in an energy sense, the final state estimate from the batch filter algorithm (Figure 1) satisfies the weighted least squares criterion and is therfore independent from the initial state guess and the a priori energy estimate.

If this initial state guess algorithm fails to converge, then the pair of angle measurements for the next observation point are added to the observation set and the process is repeated.

2.3 Marquardt Weighted Least Squares

The Marquardt weighted least squares (MLS) technique is derived from the same basic least squares solution as the batch filter algorithm described in subsection 2.2. For clarity this expression is repeated here

$$\hat{X}_n = X_n + \left(J_n^T W_{(n)}^{-1} J_n\right)^{-1} J_n^T W_{(n)}^{-1}\ \Delta Y_{(n)} \tag{2-29}$$

where

$$J_n = \begin{pmatrix} A_n \\ A_{n-1}\ \Phi_{n-1,n} \\ \vdots \\ A_{n-L}\ \Phi_{n-L,n} \end{pmatrix} . \tag{2-30}$$

At this point one may deviate from the derivation of the previous section by solving for the Jacobian matrix [Eq. (2-30)] explicitly, thus eliminating the requirement for a transition matrix Φ which is based on an approximate solution [Eq. (2-8)] to an approximate model [Eq. (2-7)]. This is accomplished by a different interpretation of the measurement matrix A as defined in Eq. (2-24) and Appendix B. Instead of taking the partial derivatives of the functions relating observations to states with respect to each state for some instant in time $\left(\frac{\partial g(X)}{\partial X}\Big|_{X=X(t_n),X(t_{n-1})\ldots X(t_{n-L})}\right)$, proceed to take the partials of the measurement equations at times $t_n, t_{n-1} \ldots t_{n-L}$ with respect to each state at time t_n

$$\left(\frac{\partial g(X)}{\partial X_n}\Big|\, X=X(t_n), X(t_{n-1})\ldots X(t_{n-L})\right) .$$

This requires a closed form solution of the equations of motion which is supplied through f and g series equations for a free falling body. While this approach requires a somewhat cumbersome development of equations, the final working algorithm will be shown to be a very precise and rapid method for calculating the Jacobian matrix explicitly.

For simplicity in this section the observation set is defined somewhat differently from the previous section

$$\underline{Y} = \begin{pmatrix} \theta \\ \\ \phi \end{pmatrix} = \begin{pmatrix} g_1 \\ \\ g_2 \end{pmatrix} = \begin{pmatrix} \tan^{-1}\left(\frac{R_y}{R_x}\right) \\ \\ \sin^{-1}\left(\frac{R_z}{|\underline{R}|}\right) \end{pmatrix} \tag{2-31}$$

where

$$\underline{R} = \underline{T} - \underline{S} \text{ or } \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} - \begin{pmatrix} S_x \\ S_y \\ S_z \end{pmatrix} \tag{2-32}$$

where $\underline{T}$ and $\underline{S}$ are the ECI position vectors of the target and sensor respectively. Now proceed to take the partial derivatives of the measurement equations with respect to each state $(x_1, x_2 \ldots x_6)$ as specified in Eq. (2-24) keeping in mind the new interpretation of the measurement matrix A.

$$\frac{\partial \theta}{\partial x_i} = \frac{R_x}{R_x^2 + R_y^2}\left(\frac{\partial R_y}{\partial x_i} - \frac{R_y}{R_x}\frac{\partial R_x}{\partial x_i}\right) \tag{2-33}$$

$$\frac{\partial \phi}{\partial x_i} = \left(|\underline{R}|^2 - R_z^2\right)^{-\frac{1}{2}}\left(\frac{\partial R_z}{\partial x_i} - \frac{R_z}{|\underline{R}|^2}\left(R_x \frac{\partial R_x}{\partial x_i} + R_y \frac{\partial R_y}{\partial x_i} + R_z \frac{\partial R_z}{\partial x_i}\right)\right) . \tag{2-34}$$

From Eq. (2-32) continue with

$$\frac{\partial R_x}{\partial x_i} = \frac{\partial T_x}{\partial x_i}, \frac{\partial R_y}{\partial x_i} = \frac{\partial T_y}{\partial x_i}, \frac{\partial R_z}{\partial x_i} = \frac{\partial T_z}{\partial x_i} . \tag{2-35}$$

For some instant in time the solution at this point would be trivial: $\frac{\partial T_x}{\partial x_1} = 1$; $\frac{\partial T_y}{\partial x_2} = 1$; $\frac{\partial T_z}{\partial x_3} = 1$; all remaining terms are zero. This would be the desired result for the least squares batch filter described in subsection 2.2. For the general case of interest continue with the f and g series equations for a free falling body

$$\underline{T} = f\underline{T}_0 + g\underline{V}_0, \ \underline{T}_0 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_{t=t_n}, \ \underline{V}_0 = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \end{pmatrix}_{t=t_n} . \tag{2-36}$$

Continuing with Eq. (2-36)

$$\begin{aligned}
&\frac{\partial T_x}{\partial x_i} = x_1 \frac{\partial f}{\partial x_i} + x_4 \frac{\partial g}{\partial x_i}, \ i = 2, 3, 5, 6 \\
&\frac{\partial T_x}{\partial x_1} = x_1 \frac{\partial f}{\partial x_1} + f + x_4 \frac{\partial g}{\partial x_1} \\
&\frac{\partial T_x}{\partial x_4} = x_1 \frac{\partial f}{\partial x_4} + x_4 \frac{\partial g}{\partial x_4} + g \\
&\frac{\partial T_y}{\partial x_i} = x_2 \frac{\partial f}{\partial x_i} + x_5 \frac{\partial g}{\partial x_i}, \ i = 1, 3, 4, 6 \\
&\frac{\partial T_y}{\partial x_2} = x_2 \frac{\partial f}{\partial x_2} + f + x_5 \frac{\partial g}{\partial x_2} \\
&\frac{\partial T_y}{\partial x_5} = x_2 \frac{\partial f}{\partial x_5} + x_5 \frac{\partial g}{\partial x_5} + g \\
&\frac{\partial T_z}{\partial x_i} = x_3 \frac{\partial f}{\partial x_i} + x_6 \frac{\partial g}{\partial x_i}, \ i = 1, 2, 4, 5 \\
&\frac{\partial T_z}{\partial x_3} = x_3 \frac{\partial f}{\partial x_3} + f + x_6 \frac{\partial g}{\partial x_3} \\
&\frac{\partial T_z}{\partial x_6} = x_3 \frac{\partial f}{\partial x_6} + x_6 \frac{\partial g}{\partial x_6} + g
\end{aligned} \tag{2-37}$$

In order to continue, the eighth order terms for f and g are introduced. But first, define the terms u_8, p_8, and q_8 for simplicity.

$$u_0 = \frac{\mu}{|\underline{T}_0|^3} = \frac{\mu}{(x_1^2 + x_2^2 + x_3^2)^{3/2}}$$

$$p_0 = \frac{\underline{V}_0 \cdot \underline{T}_0}{|\underline{T}_0|^2} = \frac{x_4 x_1 + x_5 x_2 + x_6 x_3}{x_1^2 + x_2^2 + x_3^2} \qquad (2\text{-}38)$$

$$q_0 = \frac{|\underline{V}_0|^2 - |\underline{P}_0|^2 u_0}{|\underline{P}_0|^2} = \frac{x_4^2 + x_5^2 + x_6^2}{x_1^2 + x_2^2 + x_3^2} - \frac{\mu}{(x_1^2 + x_2^2 + x_3^2)^{3/2}}$$

where μ is the product of the gravitational constant and mass of the earth. With these terms f and g can be defined

$$\begin{aligned} f = 1 &- \tfrac{1}{2} u_0 \tau^2 + \tfrac{1}{2} u_0 p_0 \tau^3 + \tfrac{1}{24} (3 u_0 q_0 - 15 u_0 p_0^2 + u_0^2) \tau^4 \\ &+ \tfrac{1}{8} (7 u_0 p_0^2 - 3 u_0 p_0 q_0 - u_0^2 p_0) \tau^5 \\ &+ \tfrac{1}{720} (630 u_0 p_0^2 q_0 - 24 u_0^2 q_0 - u_0^2 - 45 u_0 q_0^2 - 945 u_0 p_0^4 + 210 u_0^2 p_0^2) \tau^6 \\ &+ \tfrac{1}{3040} (882 u_0^2 p_0 q_0 - 3150 u_0^2 p_0^3 - 9450 u_0 p_0^3 q_0 \\ &+ 1575 u_0 p_0 q_0^2 + 63 u_0^3 p_0 + 10395 u_0 p_0^5) \tau^7 \\ &+ \tfrac{1}{40320} (1107 u_0^2 q_0^2 - 24570 u_0^2 p_0^2 q_0 - 2205 u_0^3 p_0^2 + 51975 u_0^2 p_0^4 \\ &- 42525 u_0 p_0^2 q_0^2 + 155925 u_0 p_0^4 q_0 + 1575 u_0 q_0^3 + 117 u_0^3 q_0 \\ &- 135135 u_0 p_0^6 + u_0^4) \tau^8 \quad , \end{aligned} \qquad (2\text{-}39)$$

$$\begin{aligned} g = \tau &- \tfrac{1}{6} u_0 \tau^3 + \tfrac{1}{4} u_0 p_0 \tau^4 + \tfrac{1}{120} (9 u_0 q_0 - 45 u_0 p_0^2 + u_0^2) \tau^5 \\ &+ \tfrac{1}{360} (210 u_0 p_0^3 - 90 u_0 p_0 q_0 - 15 u_0^2 p_0) \tau^6 \\ &+ \tfrac{1}{5040} (3150 u_0 p_0^2 q_0 - 54 u_0^2 q_0 - 225 u_0 q_0^2 - 4725 u_0 p_0^4 \\ &+ 630 u_0^2 p_0^2 - u_0^3) \tau^7 + \tfrac{1}{40320} (3024 u_0^2 p_0 q_0 - 12600 u_0^2 p_0^3 \\ &- 56700 u_0 p_0^2 q_0 + 9450 u_0 p_0 q_0^2 + 62370 u_0 p_0^4 + 126 u_0^3 p_0) \tau^8 \quad . \end{aligned}$$

Using Eqs. (2-38) and (2-39) continue with

$$\begin{aligned} \frac{\partial f}{\partial x_i} &= \frac{\partial f}{\partial u_0} \frac{\partial u_0}{\partial x_i} + \frac{\partial f}{\partial p_0} \frac{\partial p_0}{\partial x_i} + \frac{\partial f}{\partial q_0} \frac{\partial q_0}{\partial x_i} \\ \frac{\partial g}{\partial x_i} &= \frac{\partial g}{\partial u_0} \frac{\partial u_0}{\partial x_i} + \frac{\partial g}{\partial p_0} \frac{\partial p_0}{\partial x_i} + \frac{\partial g}{\partial q_0} \frac{\partial q_0}{\partial x_i} \quad . \end{aligned} \qquad (2\text{-}40)$$

Solving first for the partials of u_0, p_0, q_0 with respect to the states

$$\frac{\partial u_0}{\partial x_i} = \frac{-3\mu x_i}{(x_1^2 + x_2^2 + x_3^2)^{5/2}} \quad , \; i = 1, 2, 3$$

$$\frac{\partial u_0}{\partial x_i} = 0, \; i = 4, 5, 6$$

$$\frac{\partial p_0}{\partial x_i} = \frac{x_{i+3}}{x_1^2 + x_2^2 + x_3^2} - \frac{2x_i (x_4 x_1 + x_5 x_2 + x_6 x_3)}{(x_1^2 + x_2^2 + x_3^2)^2} \quad , \; i = 1, 2, 3$$

$$\frac{\partial p_0}{\partial x_i} = \frac{x_{i-3}}{x_1^2 + x_2^2 + x_3^2} \quad , \; i = 4, 5, 6 \qquad (2\text{-}41)$$

$$\frac{\partial q_0}{\partial x_i} = \frac{-(x_4^2 + x_5^2 + x_6^2) 2x_i}{(x_1^2 + x_2^2 + x_3^2)} + \frac{3\mu x_i}{(x_1^2 + x_2^2 + x_3^2)^{5/2}} \quad , \; i = 1, 2, 3$$

$$\frac{\partial q_0}{\partial x_i} = \frac{2x_i}{x_1^2 + x_2^2 + x_3^2} \quad , \; i = 4, 5, 6 \quad .$$

And finally solving for the partials of f and g with respect to u_0, p_0, q_0

$$\frac{\partial f}{\partial u_0} = -\frac{1}{2}\tau^2 + \frac{1}{2}p_0\tau^3 + \frac{1}{24}(3q_0 - 15p_0^2 + 2u_0)\tau^4$$
$$+ \frac{1}{8}(7p_0^2 - 3p_0q_0 - 2u_0p_0)\tau^5$$
$$+ \frac{1}{720}(630p_0^2q_0 - 48u_0q_0 - 3u_0^2 - 45q_0^2 - 945p_0^3 + 420u_0p_0^2)\tau^6$$
$$+ \frac{1}{5040}(1764u_0p_0q_0 - 6300u_0p_0^3 - 9450p_0^2q_0 + 1575p_0q_0^2 + 189u_0^2p_0$$
$$+ 10395p_0^3)\tau^7 + \frac{1}{40320}(2214u_0q_0^2 - 4914u_0p_0^2q_0 - 6615u_0^2p_0^2$$
$$+ 10395u_0p_0^4 - 42525p_0^2q_0 + 155925p_0^4q_0 + 1575q_0^3 + 351u_0^2q_0$$
$$- 135135p_0^6 + 4u_0^3)\tau^8$$

$$\frac{\partial q}{\partial u_0} = -\frac{1}{6}\tau^3 + \frac{1}{4}p_0\tau^4 + \frac{1}{120}(9q_0 - 45p_0^2 + 2u_0)\tau^5$$
$$+ \frac{1}{360}(210p_0^3 - 90p_0q_0 - 30u_0p_0)\tau^6$$
$$+ \frac{1}{5040}(3150p_0^4q_0 - 108u_0q_0 - 225q_0^2 - 4725p_0^4 + 1260u_0p_0^2 - 3u_0^2)\tau^7$$
$$+ \frac{1}{40320}(6048u_0p_0q_0 - 25200u_0p_0^3 - 56700p_0^4q_0 + 9450p_0q_0^2$$
$$+ 62370p_0^6 + 378u_0^2p_0)\tau^8$$

$$\frac{\partial f}{\partial p_0} = \frac{1}{2}u_0\tau^3 + \frac{1}{24}(-30u_0p_0)\tau^4 + \frac{1}{8}(21u_0p_0^2 - 3u_0q_0 - u_0^2)\tau^5$$
$$+ \frac{1}{720}(1260u_0p_0q_0 - 3780u_0p_0^3 + 420u_0^2p_0)\tau^6 \qquad (2\text{-}42)$$
$$+ \frac{1}{5040}(882u_0^2q_0 - 9450u_0^2p_0^2 - 28350u_0p_0^2q_0 + 1575u_0q_0^2$$
$$+ 63u_0^3 + 51975u_0p_0^4)\tau^7$$
$$+ \frac{1}{40320}(-49140u_0^2p_0q_0 - 4410u_0^3p_0 + 207900u_0^2p_0^2 - 85050u_0p_0q_0^2$$
$$+ 623700u_0p_0^4q_0 - 810810u_0p_0^6)\tau^8$$

$$\frac{\partial q}{\partial p_0} = \frac{1}{4}u_0\tau^4 + \frac{1}{120}(-90u_0p_0)\tau^5 + \frac{1}{360}(630u_0p_0^2 - 90u_0q_0 - 15u_0^2)\tau^6$$
$$+ \frac{1}{5040}(6300u_0p_0q_0 - 18900u_0p_0^3 + 1260u_0^2p_0)\tau^7$$
$$+ \frac{1}{40320}(3024u_0^2q_0 - 37800u_0^2p_0^2 - 170100u_0p_0^2q_0 + 9450u_0q_0^2$$
$$+ 311850u_0p_0^4 + 126u_0^3)\tau^8$$

$$\frac{\partial f}{\partial q_0} = \frac{1}{24}(3u_0)\tau^4 + \frac{1}{8}(-3u_0p_0)\tau^5 + \frac{1}{720}(630u_0p_0^2 - 24u_0^2 - 90u_0q_0)\tau^6$$
$$+ \frac{1}{5040}(882u_0^2p_0 - 945u_0p_0^3 + 3150u_0p_0q_0)\tau^7$$
$$+ \frac{1}{40320}(2214u_0^2q_0 - 2457u_0^2p_0^2 - 85050u_0p_0^2 + 155925u_0p_0^4$$
$$+ 4725u_0q_0^2 + 117u_0^3)\tau^8$$

$$\frac{\partial q}{\partial q_0} = \frac{1}{120}(9u_0)\tau^5 + \frac{1}{360}(-90u_0p_0)\tau^6 + \frac{1}{5040}(3150u_0p_0^2 - 54u_0^2)$$
$$- 450u_0q_0)\tau^7 + \frac{1}{40320}(3024u_0^2p_0 - 56700u_0p_0^3 + 18900u_0p_0q_0)\tau^8 .$$

This completes the concept of explicit Jacobian which is a fundamentally different approach from the algorithm described in subsection 2.2 where the transition matrix is required. In the next section the extension of this explicit Jacobian concept to the Kalman filter formulation is proposed.

The iterative solution method of the weighted least squares formulation [Eq. (2-29)] can be enhanced by the method of Marquardt matrix conditioning[1]. Basically this method involves the addition of a variable factor λ to the diagonal terms of the state covariance $(J_n^T W_{(n)}^{-1} J_n)$ in order to improve the convergence properties of the iterative solution method

(Gauss-Newton). This method results in the following modification of Eq. (2-29)

$$\hat{X}_n = X_n + \underbrace{\overbrace{\left(J_n^T W_{(n)}^{-1} J_n + \lambda I\right)^{-1}}^{\text{normalized } S_1^*} \overbrace{J_n^T W_{(n)}^{-1} \Delta Y_{(n)}}^{\text{normalized } S_2^*}}_{\text{denormalized } \Delta X_n} \quad . \tag{2-43}$$

Note that in order to include the factor λ, a normalizing/denormalizing procedure is required. Using the symbol * to indicate normalized, Eq. (2-43) can be represented by the following set of equations which define this normalizing procedure.

$$\begin{aligned}
\hat{X}_n &= X_n + \Delta X_n \\
\Delta X_n^* &= (S_1^* + \lambda I)^{-1} S_2^* \\
\left[S_1^*\right]_{i,j} &= S_1(i,j) / \sqrt{S_1(i,i)}\ \sqrt{S_1(j,j)} \\
\left[S_2^*\right]_i &= S_2(i) / \sqrt{S_1(i,i)} \\
\Delta X_n &= \Delta X_n^* / \sqrt{S_1(i,i)} \quad .
\end{aligned} \tag{2-44}$$

The Marquardt method approaches the Gauss-Newton method (used in the algorithm described in subsection 2.2) as $\lambda \to 0$ and the method of steepest descent as $\lambda \to \infty$. In addition, the step size increases as $\lambda \to 0$ and decreases as $\lambda \to \infty$. The strategy is to decrease λ if the solution is converging and to increase λ if it is diverging. This method thus has the ability to converge from a distant initial guess, and also the ability to converge rapidly once the vicinity of the solution is reached.

As discussed in subsection 2.2, one can avoid building the entire matrix J_n by formulating Eq. (2-43) with its algebraic equivalent

$$\begin{aligned}
\hat{X}_n = X_n + &\left\{\left(\sum_{i=1}^{L+1} J_n^T(i)\ W_{n+1-i}^{-1}\ J_n(i)\right) + \lambda I\right\}^{-1} \\
&\cdot \left\{\sum_{i=1}^{L+1} J_n^T(i) W_{n+1-i}^{-1}\ \Delta Y_{n+1-i}\right\}
\end{aligned} \tag{2-45}$$

where $J_n(1) = A_n$; $J_n(2) = A_{n-1}$;...$J_n(L+1) = A_{n-L}$.

With the techniques of explicit Jacobian and Marquardt matrix conditioning established, the final working algorithm is now presented in Figure 3. The algorithm is presented in the same format as the standard weighted least squares batch filter of subsection 2.2 (Figure 1) so that the two methods can be compared directly. The absence of a transition matrix and the use of Marquardt matrix conditioning are the distinguishing features for this formulation. The use of the closed form solution for the equations of motion (f and g series) required by the explicit Jacobian technique is also a key element. Also note that the partials for the measurement matrix are the partials of the measurement equations at times t_n, t_{n-1},...t_{n-L} with respect to each state at time t_n. This means that the R_x, R_y, R_z terms in Eqs. (2-33) and (2-34) are evaluated with the states at time t_n; t_{n-1},...t_{n-L} while Eqs. (2-37), (2-38) and (2-41) are evaluated with the state at time $t_n(X_n)$.

The method for varying the Marquardt factor λ is included in Figure 2. The chi-squared (χ^2) quantity is the criterion used to determine if the process is converging. If this quantity increases from the previous iteration, the process is determined to be diverging and the factor λ is increased by a factor of 50. If this quantity decreases from the previous iteration, the process is determined to be converging and the factor λ is decreased by a factor of 10. An initial value for λ of 0.0001 is used in this algorithm. Note that in the divergent case the process is looped back only to the point where the factor λ is added thus using the same quantities for S_1^* and S_2^*.

Like the batch filter of the previous section the Marquardt least squares algorithm requires at least three pairs of angle observations (three cycles of measurement loop) and defines the nominal state vector at the final observation point. The Marquardt least squares is also a batch or nonrecursive process so that all of the observations must be processed each time a new measurement pair is added to the observation set. Both algorithms produce the same weighted least squares state estimate by successively correcting some initial state guess. A comparison of the two methods would therefore be concerned with the stability and the computation requirements of each.

The Marquardt least squares could use the energy constrained initial state guess algorithm described in Figure 2 to supply the inital state guess to begin the iterative process.

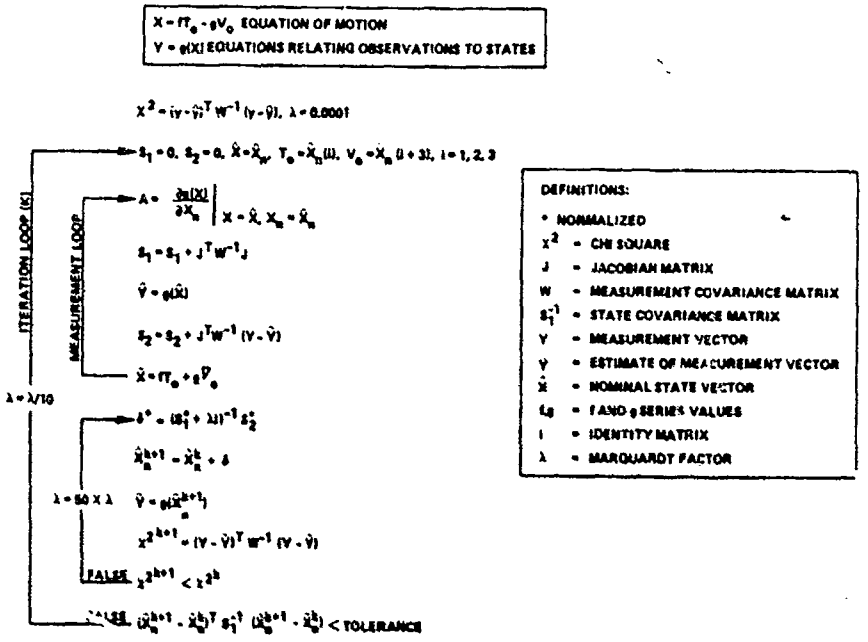


Fig. 3 Marquardt weighted least squares with explicit Jacobian

A less expensive and simplified method can also be used since the Marquardt least squares process is very stable and converges for a very wide range of state estimates. This simplified initial state guess technique is described as follows.

For observation times of t_n, t_{n-1},...t_{n-L} the following observation set may be defined

$$\underline{Y}_n = \begin{pmatrix} \theta_n \\ \phi_n \end{pmatrix}$$
$$\underline{Y}_{n-1} = \begin{pmatrix} \theta_{n-1} \\ \phi_{n-1} \end{pmatrix} \quad . \tag{2-46}$$
$$\vdots$$
$$\underline{Y}_{n-L} = \begin{pmatrix} \theta_{n-L} \\ \phi_{n-L} \end{pmatrix}$$

Using the first (n-L) and last (n) observation points the approximate angle rates are

$$\dot{\theta} = (\theta_n - \theta_{n-L}) / (t_n - t_{n-L})$$
$$\dot{\phi} = (\phi_n - \phi_{n-L}) / (t_n - t_{n-L}) \quad . \tag{2-47}$$

The velocity magnitude for the target can be approximated by the square root of the sum of its components squared

$$V = (\dot{R}^2 + (R\dot{\theta})^2 + (R\dot{\phi})^2)^{1/2} \quad . \tag{2-48}$$

Solving for the range rate yields

$$\dot{R} = (V^2 - (R\dot{\theta})^2 - (R\dot{\phi})^2)^{1/2} \quad . \tag{2-49}$$

Given some initial guess for the velocity magnitude V and range (acquisition range R_0), the range rate $\dot{R}$ may be determined from Eq. (2-49). The range at the final observation point may now be determined by

$$R_n = R_0 + \dot{R}\,(t_n - t_{n-L}) \quad . \tag{2-50}$$

At this point the working coordinate system of Figure 4 (X', Y', Z') is introduced so that the state vector in this new system can be computed directly from the quantities R_n, $\dot{R}$, $\dot{\theta}$, $\dot{\phi}$.

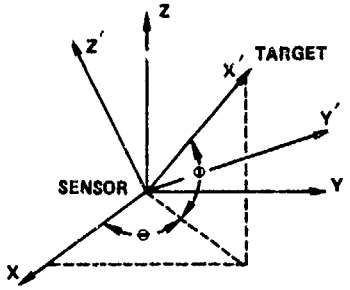


Fig. 4 Working coordinate system (X', Y', Z')

The X, Y, Z coordinate system in Figure 4 is the observer centered Cartesian coordinate system produced from the difference in the target and sensor ECI position vectors ($\underline{X} - \underline{XS}$). The state vector for the target at the final observation point in the working coordinate system can now be computed directly as

$$\underline{X}' = \begin{pmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ x_5' \\ x_6' \end{pmatrix} = \begin{pmatrix} R_n \\ 0 \\ 0 \\ \dot{R} \\ R_n\dot{\theta} \\ R_n\dot{\phi} \end{pmatrix} . \qquad (2\text{-}51)$$

The transformation matrix to convert this state vector from the X', Y', Z' system to the X, Y, Z system is

$$C = \begin{bmatrix} \cos\phi\cos\theta & -\sin\theta & -\sin\phi\cos\theta \\ \cos\phi\sin\theta & \cos\theta & -\sin\phi\sin\theta \\ \sin\phi & 0 & \cos\phi \end{bmatrix}_{\substack{\theta=\theta_n \\ \phi=\phi_n}} . \qquad (2\text{-}52)$$

The desired state vector at the final observation point in an ECI coordinate system can be written now

$$\underline{X} = \underline{CX}' + \underline{XS} . \qquad (2\text{-}53)$$

This simplified technique could be used as the primary method for establishing the initial state guess for the Marquardt least squares algorithm, or it could be used as a backup for the energy constrained iterative algorithm (Figure 2) when it fails to converge.

2.4 Energy Constraint

The problem of state estimation of ballistic trajectories with angle only measurements becomes difficult when the observer is free-falling and more difficult if the observer is then located in the plane of the observed trajectory. This section presents the idea of incorporating an energy constraint into the angle-only tracking algorithms to assist in these poor observability problems or to enhance the solution of any type of tracking problem in general. The selection of energy as a constraint is made because of its relative constancy over the whole trajectory for a given set of ICBM threats with the same ground range. The following energy constraint method can be incorporated into either of the weighted least squares algorithms discussed in subsection 2.2 or subsection 2.3.

The total energy (per unit mass) of a free-falling body can be expressed as the sum of the kinetic and potential energy

$$E = KE + PE = \frac{1}{2}V^2 - \frac{\mu}{R} \qquad (2\text{-}54)$$

where V is the velocity magnitude, R is the magnitude of the position vector (earth centered system), and μ is the product of the gravitational constant and the earth's mass. The total energy defined in Eq. (2-54) is incorporated into the angle only tracking algorithm by assuming some a priori knowledge of the expected magnitude of this quantity and its associated uncertainty. This expected or mean energy magnitude can be considered as a pseudo measurement thus expanding the measurement set of the Marquardt least squares for example to

$$\underline{Y} = \begin{pmatrix} \theta \\ \phi \\ E \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} = \begin{pmatrix} \tan^{-1} \dfrac{R_y}{R_x} \\ \sin^{-1} \dfrac{R_z}{|\underline{R}|} \\ 1/2\ V^2 - \dfrac{\mu}{R} \end{pmatrix} \quad . \tag{2-55}$$

In other words the a priori mean energy magnitude serves as the measured energy while the estimated or calculated energy is determined from Eq. (2-54) using the estimated state vector. Because the total energy is constant over the whole trajectory its contribution as a pseudo measurement is utilized only once at a single measurement point.

Reformulating the total energy in terms of the target ECI state vector

$$E = \frac{\dot{x}_1^2 + \dot{x}_2^2 + \dot{x}_3^2}{2} - \frac{\mu}{(x_1^2 + x_2^2 + x_3^2)^{1/2}} \tag{2-56}$$

the required partials for the Jacobian matrix (subsection 2.3) or measurement matrix (subsection 2.2) are obtained

$$\begin{aligned} \frac{\partial E}{\partial x_i} &= \frac{\mu x_i}{(x_1^2 + x_2^2 + x_3^2)^{1/2}} \qquad i = 1, 2, 3 \\ \frac{\partial E}{\partial x_i} &= \dot{x}_i \qquad i = 4, 5, 6 \quad . \end{aligned} \tag{2-57}$$

The measurement covariance matrix is also expanded to accommodate the expanded measurement set

$$[W]_{i,j} = \begin{bmatrix} \sigma_\theta^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_E^2 \end{bmatrix} \quad . \tag{2-58}$$

The uncertainty in the energy could be input directly or computed from a uniform distribution given some maximum and minimum energy values

$$\sigma_E^2 = \frac{(E_{max} - E_{min})^2}{12} \quad . \tag{2-59}$$

Because the energy constraint concept is based on some a priori energy estimate, the resulting least squares state estimate will be biased if this assumed a priori energy is different from the actual energy. Furthermore, the bias will increase as this difference increases and as the uncertainty in this pseudo measurement is decreased. An analysis and possible cure for the energy constraint bias problem is given in subsection 3.2.

2.5 Kalman Filter Application of Explicit Jacobian

Unlike the batch least squares process of subsections 2.2 and 2.3 the Kalman filter is a recursive minimum variance filter. Using two angle measurements at each observation point, this algorithm determines the current target state estimate such that the state covariance is minimized. Thus, the state estimate is conditioned on all measurements made up to that time. The Kalman filter is recursive in the sense that only the current measurement need be processed.

The minimum variance estimate equations can be arrived at through the weighted least squares concept, so the minimum variance estimate is also the weighted least squares estimate, and the Kalman filter can be said to be a recursive form of the batch filter. The Kalman filter uses the same linearization approximations as the batch filter of subsection 2.2 for the measurement and transition matrices. However, due to its recursive nature, errors introduced through these approximations can build up resulting in a tracking performance which tends to deviate from the least squares batch filter results. The improved Kalman filter formulation presented in this section will be shown to match the batch least squares results.

Using the notation developed in the previous sections the linearized model matrix equation for the recursive problem is

$$\Delta Y_n = A_n \Phi_{n,n-1} \Delta X_{n-1} + \epsilon_n \tag{2-60}$$

where

ΔY_n = deviations of the observations (y) from the nominal or calculated set ($\hat{y}$) at time t_n

A_n = matrix which transforms the state at time t_n to equivalent observation parameters

$\phi_{n,n-1}$ = transition matrix which transforms the state at time t_{n-1} to time t_n

ΔX_{n-1} = deviations from the estimate of the state at time t_{n-1} which correspond to ΔY_n

ϵ_n = observation errors.

The well known Kalman filter solution to this problem is presented in Figure 5. As discussed in subsection 2.2 and repeated here for clarity the transition matrix Φ in the usual approach is determined by solving the "linear" differential equation model

$$\frac{d}{dt}\,\Delta X(t) = F\,(X(t))\quad \Delta X(t) \quad . \tag{2-61}$$

An "approximate" solution is given by

$$\Delta X(t + \Delta t) = \Delta X(t) + \Delta t\, F\,(X(t))\quad \Delta X(t) \tag{2-62}$$

so that

$$\Phi(t + \Delta t,\, t) = I + \Delta t\, F\,(X(t)) \tag{2-63}$$

or

$$\Phi_{n,n-1} = I + \Delta t\, F\,(X(t_n)) \tag{2-64}$$

where

$$F\left[(X(t_n))\right]_{i,j} = \left.\frac{\partial f_i}{\partial X_j}\right|_{X=X(t_n)} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_6} \\ \vdots & & \vdots \\ \frac{\partial f_6}{\partial x_1} & \cdots & \frac{\partial f_6}{\partial x_6} \end{pmatrix}_{X=X(t_n)} \tag{2-65}$$

and

$$f(X) = \frac{dX}{dt} \quad . \tag{2-66}$$

As discussed in subsection 2.2 the accuracy of Φ can be improved by breaking Δt up into equal fractions $h = \frac{\Delta t}{m}$ and using the equation

$$\Phi(t + kh,t) = \left[I + hF\left(t + (k-1)h\right)\right]\ \Phi\left((t + (k-1)h,t\right),\ k = 1,2,3\ldots m \quad . \tag{2-67}$$

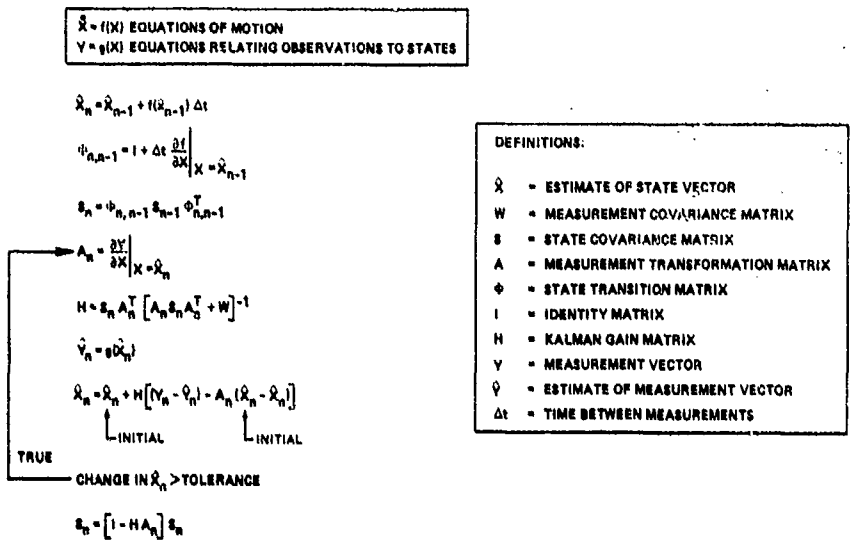


Fig. 5 Kalman filter algorithm

Additional accuracy can be achieved by adding another term to the approximate solution of Eq. (2-62)

$$\Delta X(t + \Delta t) = \Delta X(t) + \Delta t\, F(t)\, \Delta X(t) + \frac{\Delta t^2}{2}\left(\dot{F}(t) + F^2(t)\right)\, \Delta X(t) \quad . \qquad (2\text{-}68)$$

Another method for attacking the accuracy problem is to solve for the Jacobian matrix ($A_n\, \Phi_{n,n-1}$) explicitly as was done for the MLS algorithm in subsection 2.3. The partial derivatives of the measurement equations with respect to each state for the explicit Jacobian are identical to those derived in subsection 2.3 for the MLS algorithm. In order to implement the explicit Jacobian into the Kalman filter formulation of Figure 5 one need only to substitute the Jacobian where ever the pair $A_n\, \Phi_{n,n-1}$ occurs. Using the notation of Figure 5 the Kalman gain matrix equation can be written as

$$H = \Phi_{n,n-1}\, S_{n-1}\, \Phi^T_{n,n-1}\, A^T_n \left[A_n\, \Phi_{n,n-1}\, S_{n-1}\, \Phi^T_{n,n-1}\, A^T_n + W\right]^{-1} \qquad (2\text{-}69)$$

which would yield the following Jacobian formulation

$$H = \Phi_{n,n-1}\, S_{n-1}\, J^T \left[J\, S_{n-1}\, J^T + W\right]^{-1} \quad . \qquad (2\text{-}70)$$

In similar fashion the updated state covariance matrix equation can be written as

$$S_n = \Phi_{n,n-1}\, S_{n-1}\, \Phi^T_{n,n-1} - H\, A_n\, \Phi_{n,n-1}\, S_{n-1}\, \Phi^T_{n,n-1} \qquad (2\text{-}71)$$

which would yield the following Jacobian formulation

$$S_n = \Phi_{n,n-1} S_{n-1} \Phi^T_{n,n-1} - HJS_{n-1}\Phi^T_{n,n-1} \quad . \qquad (2\text{-}72)$$

The resulting Jacobian-Kalman formulation is presented in Figure 6.

This is only an intermediate formulation presented at this point for clarity. From Figure 6 the equation

$$\hat{X}_n = \hat{X}_n + H(Y_n - \hat{Y}_n) \qquad (2\text{-}73)$$

can be written as

$$\Phi_{n,n-1}\, \hat{X}_{n-1} = \Phi_{n,n-1}\, \hat{X}_{n-1} + H(Y_n - \hat{Y}_n) \qquad (2\text{-}74)$$

or

$$\hat{X}_{n-1} = \hat{X}_{n-1} + \Phi^{-1}_{n,n-1}\, H(Y_n - \hat{Y}_n) \quad . \qquad (2\text{-}75)$$

From Figure 6

$$\Phi^{-1}_{n,n-1}\, H = S_{n-1}\, J^T\, (J\, S_{n-1}\, J^T + W)^{-1} \quad . \qquad (2\text{-}76)$$

Letting $H' = \Phi^{-1}_{n,n-1}\, H$ and using Eqs. (2-75) and (2-76) the final Jacobian-Kalman algorithm is presented in Figure 7. This final algorithm is very similar to the intermediate result of Figure 6 but has the following advantages. In the final algorithm the transition matrix has been eliminated entirely from the equations used to determine the final state estimate. When the transition matrix is calculated for the state covariance update, it is evaluated with the final state estimate.

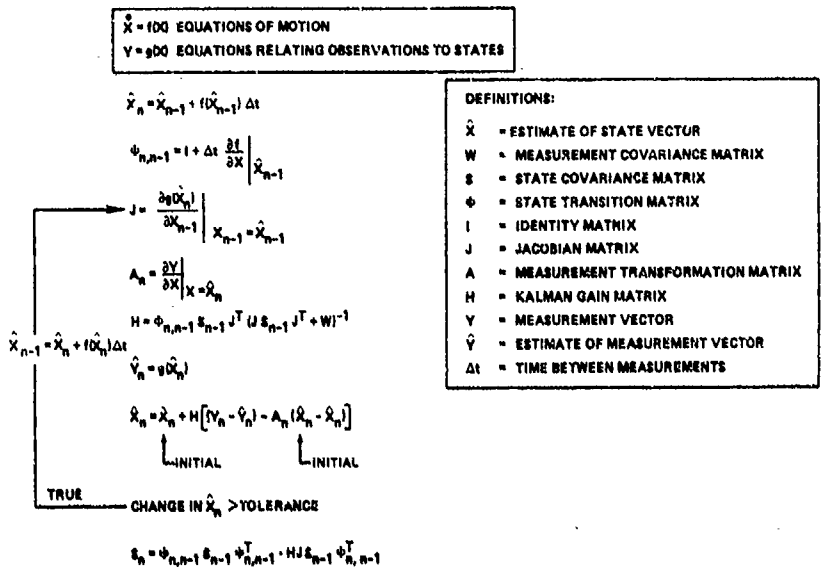


Fig. 6 Jacobian-Kalman algorithm (intermediate)

For the Kalman filter formulation Eqs. (2-37), (2-38), and (2-41) are evaluated with $\hat{X}_{n-1}$ while R_x, R_y, R_z terms of Eqs. (2-33) and (2-34) are evaluated with $\hat{X}_n$. Note that for this formulation the state estimate $\hat{X}_{n-1}$ must be projected forward to t_n each time an

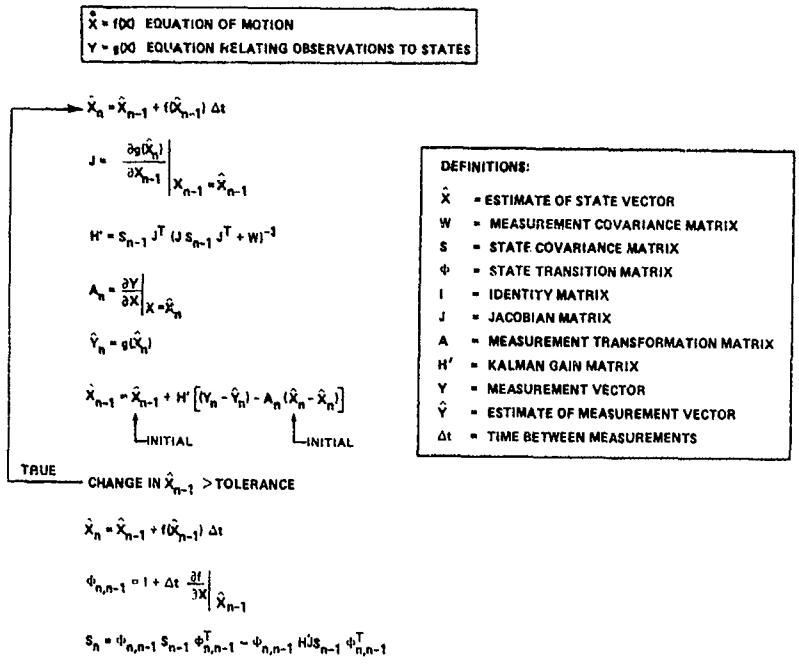


Fig. 7 Jacobian-Kalman algorithm (final)

iteration is performed. This is accomplished with the f and g series [Eqs. (2-36) and (2-39)].

3. PERFORMANCE EVALUATION

3.1 Test Case Description

Five test cases were used to demonstrate the performance of the tracking algorithms presented in this report. These cases represent different degrees of difficulty in terms of relative engagement geometry. In each case both the sensor and target are on ballistic trajectories. For cases 3, 4, and 5 the sensor states are identical. The initial ECI state vectors for these test cases are given in Table I. Some of the initial characteristic trajectory parameters which further describe these five test cases are given in Table II.

3.2 Case 1

In terms of tracking performance this case provides the best results of the five test cases. The aspect angle in Table II confirms the favorable tracking geometry provided with this case. The Marquardt least squares (MLS) results for this case are shown in Figure 8 for both the constrained and unconstrained mode. The results are for 50 Monte Carlo trials and assume an angle measurement accuracy of 20 μrad (1σ). A frame time of 10 sec was used so the initial results shown at 20 sec track time are for three measurements. For the energy constrained mode, three sets of results are shown for assumed one sigma energy uncertainties of 10, 20, and 40 percent $\left(\frac{\sigma_{energy}}{energy} \times 100\right)$. Note that all constrained cases approach the unconstrained results.

The standard least squares batch filter results for this same case are shown in Figure 9. As expected, these agree with the MLS results of Figure 8 since both algorithms provide the same least squares solution to the problem. The somewhat erratic behavior of the energy constrained performance appears to be case dependent since almost identical results were obtained for different sets of measurements simulated with different random number sequences. Note that the standard least squares batch filter experienced convergence problems for the unconstrained mode when less than five measurements were utilized while the MLS algorithm did not.

In Figure 10 an attempt is made to give some indication of the computational speed of the two least square algorithms. The algorithms were made equal in terms of state projection methods and criterion used for termination of the iterative process. The performance is given in terms of CDC 7600 CPU run time per Monte Carlo trial. No attempt was made to enumerate the number of computational operations for the two algorithms. While this is a rough computational comparison, the results shown in Figure 10 indicate promise and warrant further investigation for what appears to be a faster MLS algorithm.

The bias error introduced through the a priori estimate of the target energy in the energy constrained mode was also investigated for this case. In this analysis the effect of the error in the a priori estimated energy on the resulting bias was examined. The results of

Table I
Test Cases - ECI Initial State Vectors

	X (m)	Y (m)	Z (m)	$\dot{X}$ (m/sec)	$\dot{Y}$ (m/sec)	$\dot{Z}$ (m/sec)
CASE 1						
TARGET	2055397.0	-2486767.2	6353203.9	-1337.40	-5320.04	-1810.57
SENSOR	-369437.1	-3841530.9	5376142.4	251.65	2616.77	-2004.02
CASE 2						
TARGET	55986.6	-1977835.8	8149027.5	171.90	-3871.66	-2046.71
SENSOR	-409131.8	-4254290.7	6641171.5	190.14	1977.13	-974.70
CASE 3						
TARGET	-216183.4	-2011908.4	7467052.0	-594.01	-4874.89	-2107.44
SENSOR	-339809.8	-4448893.3	5091404.1	-493.60	-1039.40	2544.72
CASE 4						
TARGET	-255729.4	-1857730.1	7591108.4	-566.59	-4828.72	-1896.57
SENSOR	-339809.8	-4448893.3	5091404.1	-493.60	-1039.40	2544.72
CASE 5						
TARGET	-198123.0	-1731424.1	7661489.0	-576.84	-4841.18	-1718.58
SENSOR	-339809.8	-4448893.3	5091404.1	-493.60	-1039.40	2544.72

Table II
Test Cases - Initial Characteristic Parameters

CASE	SENSOR ALTITUDE		TARGET ALTITUDE		RANGE		ASPECT ANGLE
	KFT	KM	KFT	KM	NMI	KM	(deg)
1	3573	1089	2506	764	1500	2778	44.8
2	5032	1534	6650	2027	1496	2770	10.7
3	1322	403	4520	1378	1839	3406	3.3
4	1322	403	4793	1461	1944	3601	3.8
5	1322	403	4917	1499	2021	3743	3.2

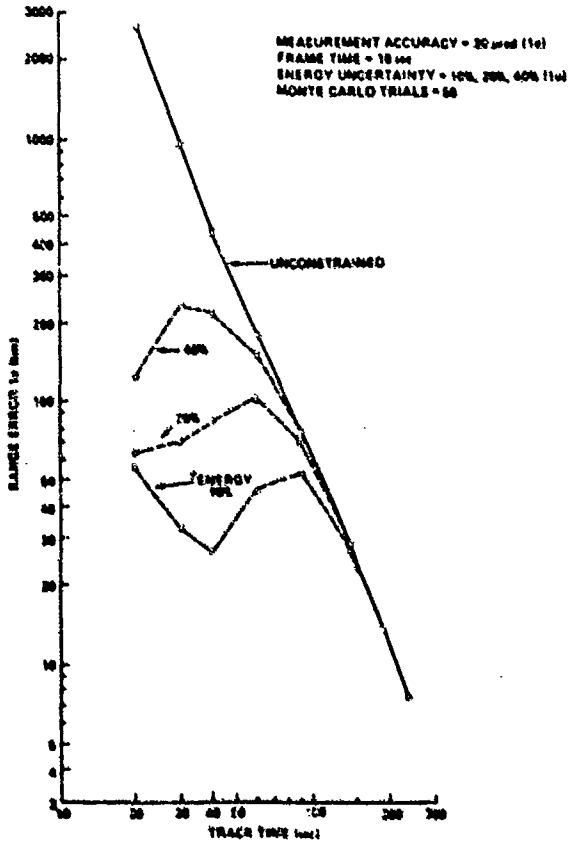


Fig. 8 Marquardt least squares - Case 1

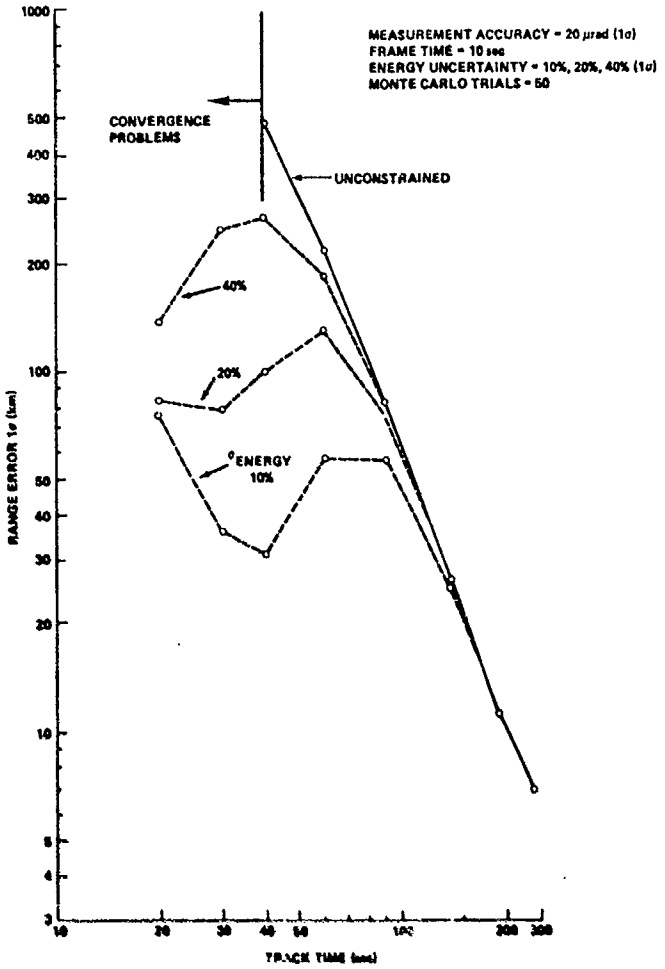


Fig. 9 Standard least squares batch filter - Case 1

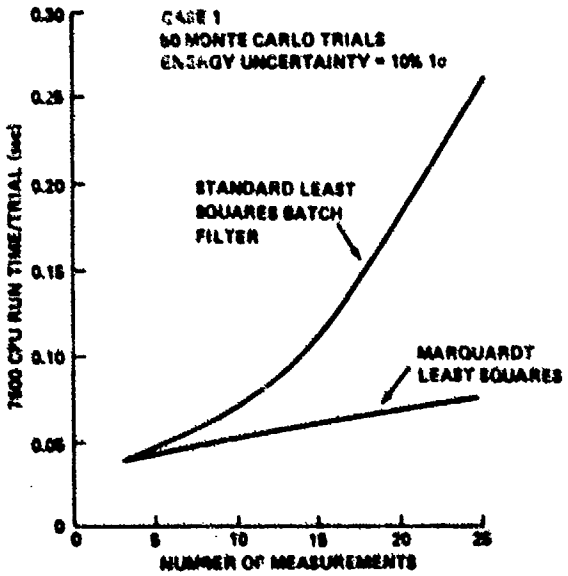


Fig. 10 Algorithm run time

this analysis are shown in Figure 11 for a track time of 150 sec and for assumed one sigma energy uncertainties of 10 and 20 percent. As an example, if the a priori energy measurement was 20 percent in error and a one sigma energy uncertainty of 10 percent was used then the bias on the range estimate would be 30 km. One must therefore balance the benefits gained through the use of smaller energy uncertainties against the resulting increase in bias for incorrect a priori energy measurements.

In order to eliminate the bias problem and the high processing requirements of the constrained batch process, the ideal approach is to initialize track with the constrained least squares batch algorithm and then to handover to the unconstrained Kalman filter for the continuous track function. The least squares batch algorithm would operate in the energy constrained mode to improve performance and guarantee convergence of the unconstrained Kalman

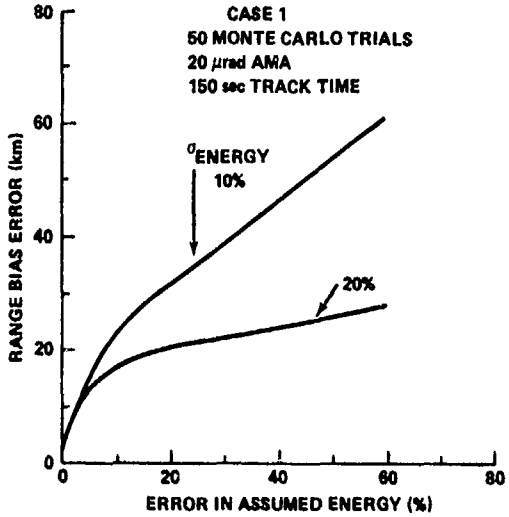


Fig. 11 Bias error due to incorrect assumed energy

filter. Using this method, any bias error introduced through the energy constrained least squares batch process would be removed by the unconstrained Kalman filter. This method is especially attractive if the Kalman filter performance matches the least squares batch results which is the case for Jacobian-Kalman filter formulation.

The Jacobian-Kalman filter results for Case 1 are presented in Figure 12. A five measurement constrained MLS was used to initialize the Jacobian-Kalman filter. Three sets of results are shown for a 5, 10, and 20 percent one sigma energy uncertainty for initialization. Also indicated are the MLS results from Figure 8 to show how the Jacobian-Kalman matches the least squares batch performance when initialized with an energy constrained MLS algorithm.

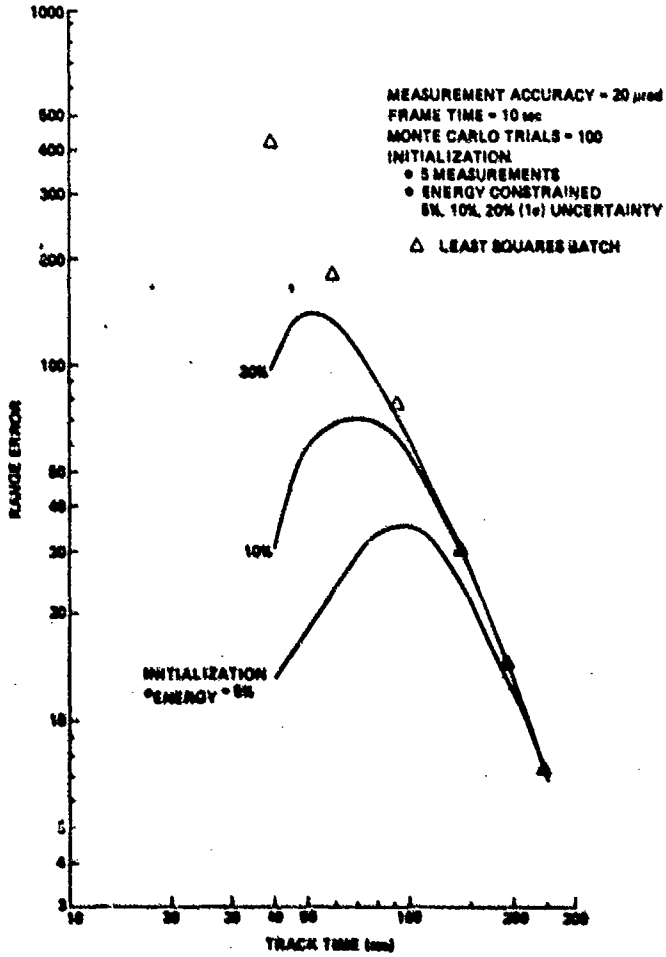


Fig. 12 Jacobian-Kalman - Case 1

In order to demonstrate how the unconstrained Kalman filter removes the bias error introduced through an incorrect a priori energy estimate in the constrained least squares batch process the following test was made with Case 1. A bias error was introduced by assuming an a priori energy estimate for the five measurement least squares initialization algorithm which was 10 percent in error. The resulting bias error after initialization (Time = 40

sec) and after each subsequent measurement processed by the Jacobian-Kalman filter is presented in Figure 13.

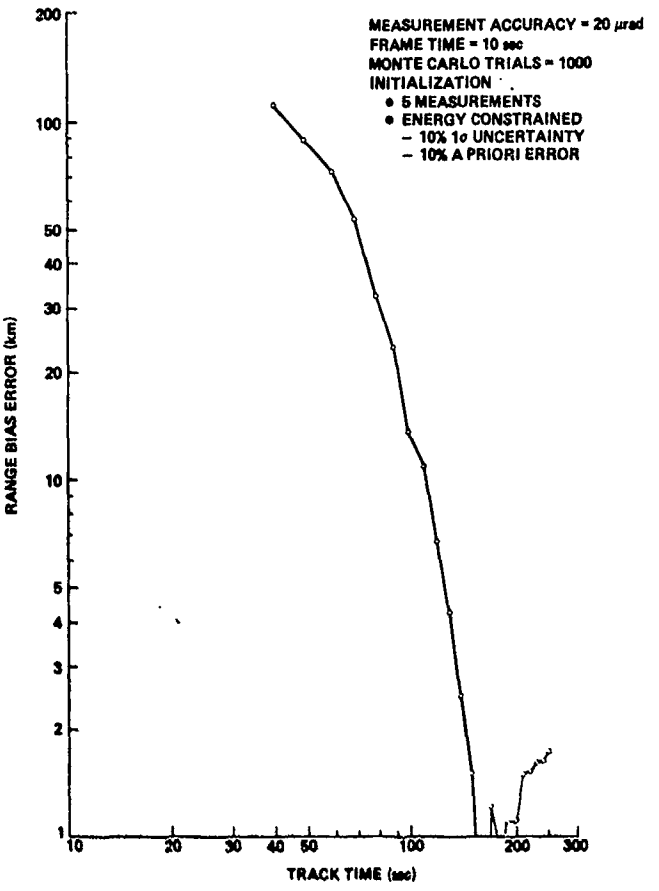


Fig. 13 Jacobian-Kalman bias - Case 1

3.3 Case 2

This is a more difficult case than the previous one due to its near in-plane geometry. The MLS results for this case are shown in Figure 14 for both the constrained and unconstrained mode. The standard least squares batch filter results for this same case are given in Figure 15 for comparison. Notice that the unconstrained and 40 percent energy uncertainty constrained cases were omitted for the standard least squares batch filter due to convergence problems. Also note that in order to ensure convergence for this problem beyond 100 sec of track, this algorithm must begin track at 10 measurements or less. These problems were not experienced with the MLS algorithm.

The Jacobian-Kalman filter results for Case 2 are presented in Figure 16. A five measurement MLS algorithm with energy constraints was used for initialization. Also indicated are the MLS results from Figure 14 for comparison.

3.4 Case 3, Case 4, and Case 5

These three cases which are all very similar in geometry proved to be the most difficult. The aspect angles in Table 3-2 indicate a relative geometry very near to in-plane. The unconstrained MLS results for these three cases are given in Figure 17. Although no results are presented for the standard least squares batch filter, convergence problems were again experienced for both the constrained and unconstrained mode using this algorithm.

The Jacobian-Kalman filter results for these three cases are presented in Figure 18. A five measurement MLS algorithm with energy constraints was used for initialization. These results match the MLS performance shown in Figure 17.

4. CONCLUSIONS

Based on the performance results of Section 3, the MLS algorithm is a very stable and computationally fast algorithm. The MLS algorithm has been shown to be a superior algorithm in terms of stability and computational requirements when compared to existing least squares batch algorithms which use both a measurement and transition matrix. This algorithm has never experienced convergence difficulties against even the most difficult of relative engagement geometries. In addition this algorithm has been shown to converge for a very wide range of initial state guesses (±50 percent range guess). The use of the explicit Jacobian method provides for higher accuracy and speed by eliminating the need for the

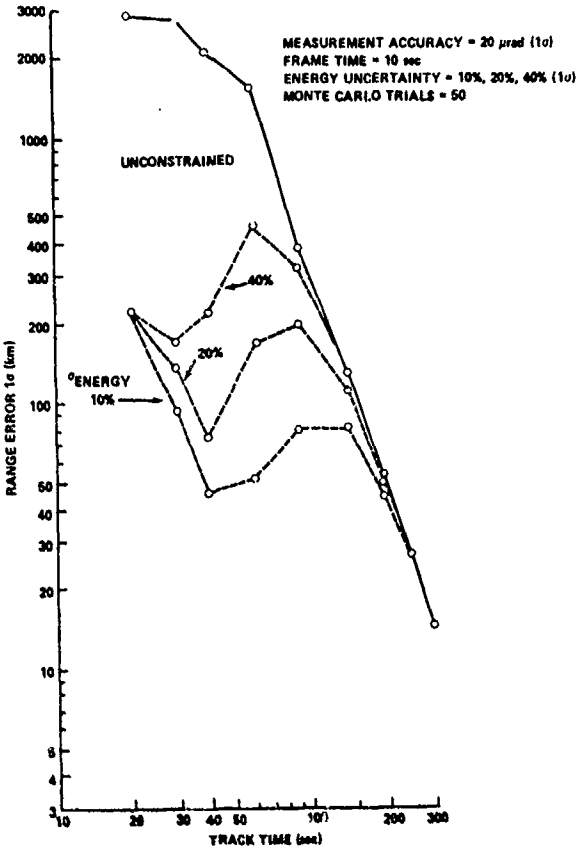


Fig. 14 Marquardt least squares - Case 2

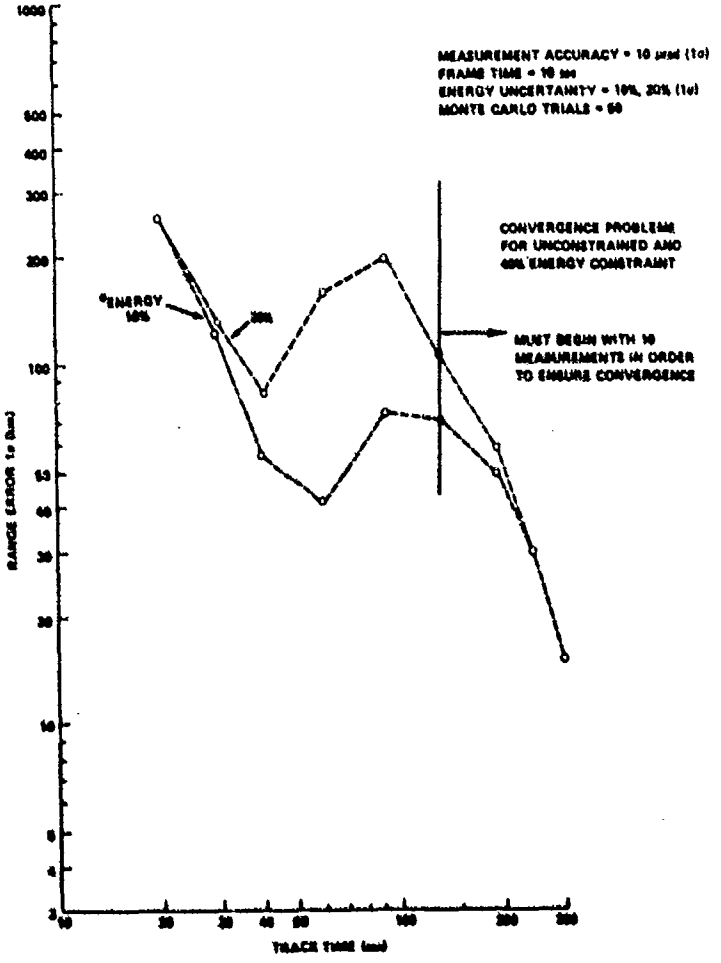


Fig. 15 Standard least squares batch filter - Case 2

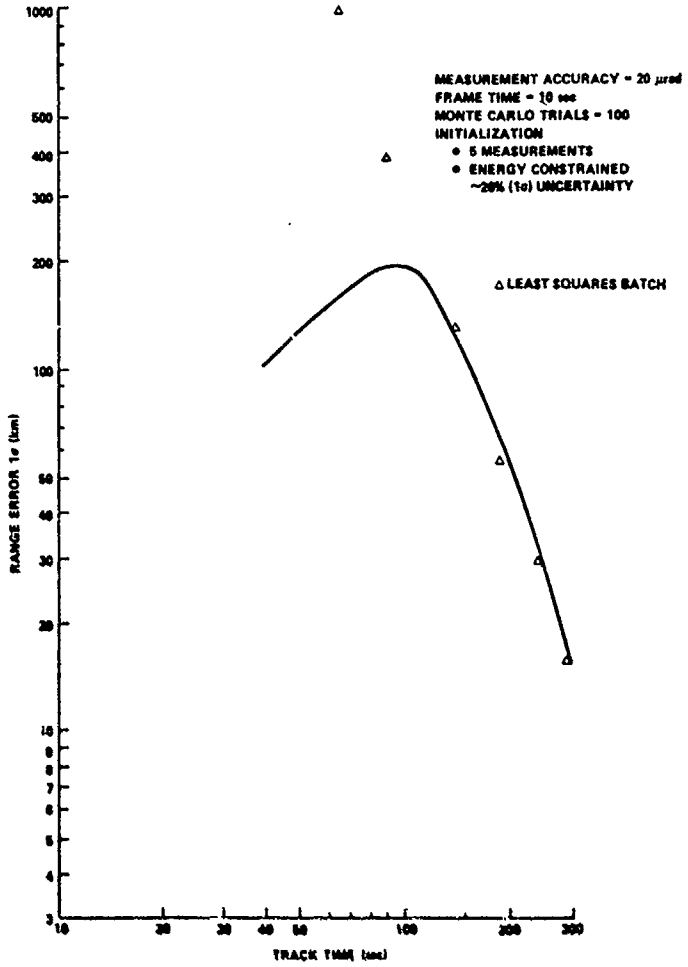


Fig. 16 Jacobian-Kalman - Case 2

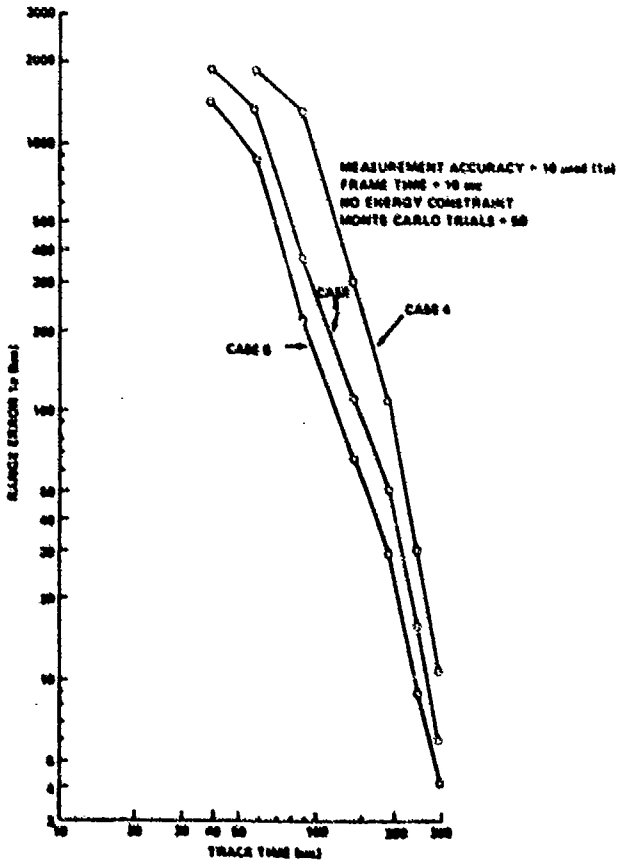


Fig. 17 Marquardt least squares - Cases 3, 4, and 5

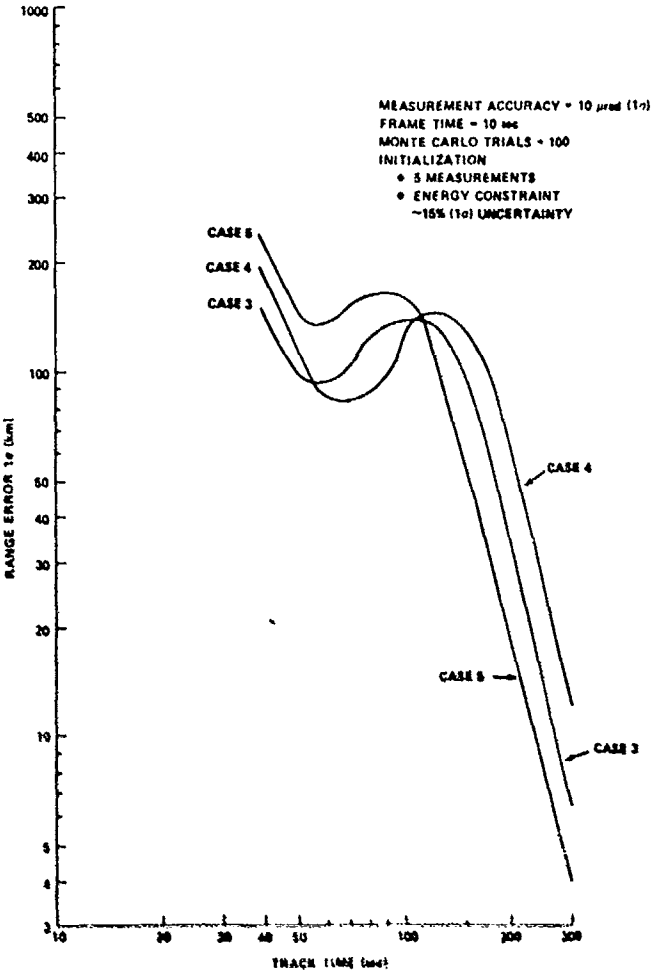


Fig. 18 Jacobian-Kalman - Cases 3, 4, and 5

transition matrix used in the usual approach. This improved accuracy along with the technique of Marquardt matrix conditioning contribute to the enhanced stability of the algorithm. When used in an energy constrained mode this algorithm serves as an ideal initialization technique for the Jacobian-Kalman filter algorithm.

Bias errors are introduced in the energy constrained mode of the MLS algorithm when the a priori estimated energy measurement is in error. This bias becomes greater as the uncertainty in the energy measurement is decreased. This problem can be eliminated by initializing track in the constrained mode and then handing over to the unconstrained Jacobian-Kalman filter algorithm for the continuous track function. By this method, any bias error introduced in the energy constrained initialization process is removed in the unconstrained Kalman filter. This approach is especially attractive since the Jacobian-Kalman filter performance matches the least squares batch results.

The application of the explicit Jacobian technique to the Kalman filter formulation has produced a recursive tracking algorithm whose performance has been shown to match that of the least squares batch process. This is an improvement over existing Kalman filter algorithms which require both a measurement and transition matrix which results in a tracking performance that tends to deviate from the least squares batch performance bound. For the test cases described in Section 3, the tracking performance (1σ range error) for the Jacobian-Kalman filter algorithm was consistently 30 to 50 percent better than the performance of a tracking algorithm which used this latter type of Kalman filter formulation.

REFERENCES

1. Marquardt, Donald W., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," Journal of the Society for Industrial and Applied Mathematics, Vol. 5, No. 1, March 1957, pp. 32-38.

2. Chang, C. B., and Dunn, K. P., Angle-Only Tracking Algorithms and Their Performance, Project Report No. RMF-183, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts, 26 October 1979.

APPENDIX A. TRANSITION EQUATION

State vector

$$\underline{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

Equations of motion

$$\dot{\underline{X}} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \underline{f}(X) = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{pmatrix} = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ -\dfrac{GMX_1}{R^3} \\ -\dfrac{GMX_2}{R^3} \\ -\dfrac{GMX_3}{R^3} \end{pmatrix}$$

$R = \sqrt{x_1^2 + x_2^2 + x_3^2}$

G = gravitational constant

M = earth's mass

Transition equation

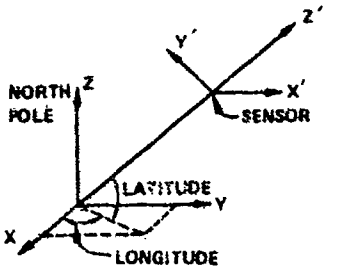
$$\phi_{n-1,n} = I + F\left(X(t_n)\right) \quad (t_{n-1} - t_n)$$

F matrix

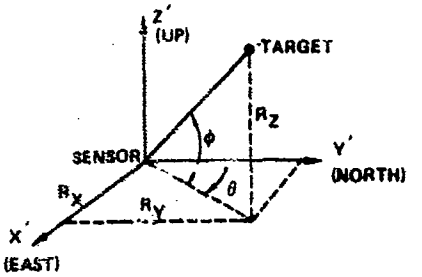
$$\left[F\left(X(t_n)\right)\right]_{i,j} = \left.\frac{\partial f_i}{\partial x_j}\right|_{X=X(t_n)}$$

$$\left(\begin{array}{c|c|c|ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \dfrac{GM(2x_1^2 - x_2^2 - x_3^2)}{R^5} & \dfrac{3GMx_1x_2}{R^5} & \dfrac{3GMx_1x_3}{R^5} & 0 & 0 & 0 \\ \hline \dfrac{3GMx_1x_2}{R^5} & \dfrac{GM(2x_2^2 - x_1^2 - x_3^2)}{R^5} & \dfrac{3GMx_2x_3}{R^5} & 0 & 0 & 0 \\ \hline \dfrac{3GMx_1x_3}{R^5} & \dfrac{3GMx_2x_3}{R^5} & \dfrac{GM(2x_3^2 - x_1^2 - x_2^2)}{R^5} & 0 & 0 & 0 \end{array}\right)_{X=X(t_n)}$$

APPENDIX B. MEASUREMENT EQUATION



EARTH CENTERED INERTIAL (ECI)



SENSOR CENTERED TOPOGRAPHIC (SCT)

Range vector

$$\underline{R} = \underline{T} - \underline{S} \text{ or } \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix} = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} - \begin{pmatrix} S_x \\ S_y \\ S_z \end{pmatrix}$$

Transformation equation

$$\underline{R}(SCT) = T\,\underline{R}(ECI)$$

$$T = \begin{pmatrix} -\sin(long) & \cos(long) & 0 \\ -\cos(long)\sin(lat) & -\sin(long)\sin(lat) & \cos(lat) \\ \cos(long)\cos(lat) & \sin(long)\cos(lat) & \sin(lat) \end{pmatrix}$$

Measurement vector

$$\underline{Y} = \begin{pmatrix} \theta \\ \\ \phi \end{pmatrix} = g(X) = \begin{pmatrix} g_1 \\ \\ g_2 \end{pmatrix} = \begin{pmatrix} \tan^{-1}\dfrac{R_x}{R_y} \\ \\ \tan^{-1}\dfrac{R_z}{\sqrt{R_x^2 + R_y^2}} \end{pmatrix}$$

Measurement equation

$$\Delta Y_n = A_n \Delta X_n + \epsilon_n$$

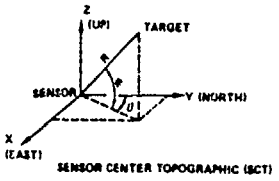
A matrix

$$[A_r]_{i,j} = [A(X(t_n))]_{i,j} = \left.\frac{\partial g_i}{\partial x_j}\right|_{X=X(t_n)}$$

$$\left(\begin{array}{c|c|c|c|c|c} \dfrac{x_2}{x_1^2 + x_2^2} & \dfrac{-x_1}{x_1^2 + x_2^2} & 0 & 0 & 0 & 0 \\ \hline \dfrac{-x_1 x_3 (x_1^2 + x_2^2)^{-\frac{1}{2}}}{x_1^2 + x_2^2 + x_3^2} & \dfrac{-x_2 x_3 (x_1^2 + x_2^2)^{-\frac{1}{2}}}{x_1^2 + x_2^2 + x_3^2} & \dfrac{(x_1^2 + x_2^2)^{\frac{1}{2}}}{x_1^2 + x_2^2 + x_3^2} & 0 & 0 & 0 \end{array}\right) \underline{X} = \underline{R}(SCT)$$

APPENDIX C. DERIVATION OF INITIAL STATE GUESS ALGORITHM

Relative State Equations



SENSOR CENTER TOPOGRAPHIC (SCT)

$$
\begin{aligned}
x &= R \sin\theta \cos\phi \\
y &= R \cos\theta \cos\phi \\
z &= R \sin\phi \\
\dot{x} &= \dot{R} \sin\theta \cos\phi - R\dot{\phi} \sin\theta \sin\phi + R\dot{\theta} \cos\theta \cos\phi \\
\dot{y} &= \dot{R} \cos\theta \cos\phi - R\dot{\phi} \cos\theta \sin\phi - R\dot{\theta} \cos\theta \sin\phi \\
\dot{z} &= \dot{R} \sin\phi + R\dot{\phi} \cos\phi
\end{aligned}
\tag{C-1}
$$

Dividing $\dot{x}$, $\dot{y}$, $\dot{z}$ by R

$$
\begin{aligned}
a_1 &= \frac{\dot{x}}{R} = \frac{\dot{R}}{R} \sin\theta \cos\phi - \dot{\phi} \sin\theta \sin\phi + \dot{\theta} \cos\theta \cos\phi \\
a_2 &= \frac{\dot{y}}{R} = \frac{\dot{R}}{R} \cos\theta \cos\phi - \dot{\phi} \cos\theta \sin\phi - \dot{\theta} \cos\theta \sin\phi \\
a_3 &= \frac{\dot{z}}{R} = \frac{\dot{R}}{R} \sin\phi + \dot{\phi} \cos\phi
\end{aligned}
\tag{C-2}
$$

Velocity Vector Definitions

$\underline{V}_T$ = Target velocity vector in SCT system

$\underline{V}_S$ = Sensor velocity vector in SCT system

$\underline{V}_R$ = Relative velocity vector (TARGET-SENSOR)

$$\underline{V}_T = \underline{V}_S + \underline{V}_R$$

$$
\underline{V}_T = \begin{bmatrix} \dot{x}_S \\ \dot{y}_S \\ \dot{z}_S \end{bmatrix} + \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{z}_R \end{bmatrix} = \begin{bmatrix} \dot{x}_S + a_1 R \\ \dot{y}_S + a_2 R \\ \dot{z}_S + a_3 R \end{bmatrix}
$$

Target velocity magnitude squared

$$
\begin{aligned}
|\underline{V}_T|^2 &= (\dot{x}_S + a_1 R)^2 + (\dot{y}_S + a_2 R)^2 + (\dot{z}_S + a_3 R)^2 \\
|\underline{V}_T|^2 &= R^2(a_1^2 + a_2^2 + a_3^2) + R(2\dot{x}_S a_1 + 2\dot{y}_S a_2 + 2\dot{z}_S a_3) + (\dot{x}_S^2 + \dot{y}_S^2 + \dot{z}_S^2)
\end{aligned}
\tag{C-3}
$$

Energy equation per unit mass

$$E_T = E_{kinetic} + E_{potential} = \frac{1}{2}|\underline{V}_T|^2 = \frac{\mu}{R_T} \tag{C-4}$$

where R_T = distance from earth center to target and μ = gravitational constant × earth's mass.

Combining Eqs. (C-3) and (C-4)

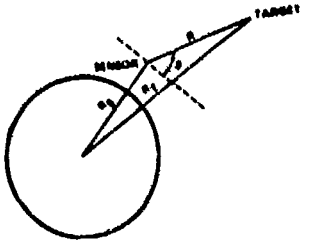
$$
\underbrace{R^2(a_1^2 + a_2^2 + a_3^2)}_{a} + \underbrace{R(2\dot{x}_S a_1 + 2\dot{y}_S a_2 + 2\dot{z}_S a_3)}_{b} + \underbrace{(\dot{x}_S^2 + \dot{y}_S^2 + \dot{z}_S^2) - 2E_T - \frac{2\mu}{R_T}}_{c} = 0 \tag{C-5}
$$

Quadratic solution of R

$$R = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{C-6}$$

From geometry

$$R_T = (R^2 + R_S^2 + 2RR_S \sin\phi)^{\frac{1}{2}} \tag{C-7}$$



For a free-falling body

$$\frac{\dot{R}}{R} = \dot{\phi} \tan \phi - \frac{\ddot{\theta}}{2\dot{\theta}} \qquad \text{(C-8)}$$

1) Calculate $\frac{\dot{R}}{R}$ [Eq. (C-8)]

2) Calculate a_1, a_2, a_3 [Eq. (C-2)]

3) Calculate a, b [Eq. (C-5)]

4) Calculate R_T [Eq. (C-7)]

5) Calculate c [Eq. (C-5)]

6) Calculate R [Eq. (C-6)]

7) Repeat 4, 5, 6 until R converges

8) Calculate $\dot{R}$ [Eq. (C-8)]

9) Calculate x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ [Eq. (C-1)]

NEW SMOOTHING ALGORITHMS FOR DYNAMIC SYSTEMS WITH OR WITHOUT INTERFERENCE

Kerim Demirbaş

School of Engineering and Applied Science
University of California, Los Angeles
California, USA

1. INTRODUCTION

In target tracking: First, a motion model which describes the motion of the target tracked as accurately as possible, and an observation model are obtained. In a clear environment (i.e., there does not exist any interference such as jamming or clutter); these models are in general, discrete and linear with respect to the disturbance and observation noises. Moreover the observation noise is an additive Gaussian noise where the Gaussian assumption is due to the Central Limit Theorem. Next, using one of the estimation algorithms already developed in the literature[9-13] (e.g., the (extended) Kalman filter algorithm), the target states are estimated.

If either the motion or observation model is nonlinear (in general, this is the case), then the optimum solution to this estimation problem can not be given due to nonlinear functions in the models. However, a suboptimal solution is given by using a nonlinear estimation algorithm. By a nonlinear estimation algorithm (e.g., the extended Kalman filter algorithm), the states are estimated as follows: First, using a Taylor series expansion, the nonlinear functions are linearized around some points so that the nonlinear estimation problem is reduced to a linear one. Then using a known (usually) solution to this linear estimation problem, the states are estimated. Hence, if the nonlinear functions are not smooth enough for a Taylor series expansion, then the nonlinear estimation algorithm may produce estimates which are much different from the actual values of the states. Therefore, some estimation algorithms for an estimation problem with nonlinear functions which are not smooth enough for a Taylor series expansion are needed.

If both the motion and observation models are linear, we have a problem of state estimation in a linear discrete time system. The solution to this problem has been treated extensively in the literature.[9-13]

In the tracking of a target in the presence of interference (such as jamming or clutter which is, in general, not Gaussian noise), an observation model with only additive observation noise may not be used since it does not account for the possibility of utilizing measurements originating simultaneously from the interference source and the target. Still, if a classical estimation algorithm were used to track the target by using an observation model with only observation noise, the estimates of the target states may diverge from the actual values. Hence, some estimation algorithms are needed for discrete models with arbitrary random interference as well as an observation noise.

This chapter treats the problem of state estimation for discrete models with or without interference. As a result, three new smoothing algorithms are presented so that the need mentioned above is fulfilled. The main idea for these smoothing algorithms is that of quantizing the states of the models to a finite set of states. This (approach) reduces the smoothing problem to a multiple (composite) hypothesis testing problem. Further, using three decoding techniques of information theory, the smoothing algorithms are developed. The first smoothing algorithm is referred to as Optimum Decoding Based Smoothing Algorithm, which uses the Viterbi decoding algorithm. The second smoothing algorithm is referred to as Stack Sequential Decoding Based Smoothing Algorithm, which uses a stack sequential decoding algorithm. The third one is referred to as Suboptimum Decoding Based Smoothing Algorithm, which uses a suboptimum decoding algorithm.

2. SMOOTHING ALGORITHMS

2.1 Models and Assumptions

Through this section, we deal with the following discrete models

$$\begin{aligned} x(k+1) &= f(k, x(k), u(k), w(k)) \quad \text{(Motion Model)} \\ z(k) &= g(k, x(k), v(k)) \quad \text{(Observation Model)} \end{aligned} \tag{2.1.1}$$

for target tracking in a clear environment. In the presence of interference we deal with the following models.

$$\begin{aligned} x(k+1) &= f(k, x(k), u(k), w(k)) \quad \text{(Motion Model)} \\ z(k) &= g(k, x(k), I(k), v(k)) \quad \text{(Observation Model)} \end{aligned} \tag{2.1.2}$$

where

$x(0)$ is an $n \times 1$ initial (target) state random vector

$x(k)$ is an $n \times 1$ (target) state vector at time k

$u(k)$ is a $q \times 1$ known pilot-command vector at time k

$w(k)$ is a $p \times 1$ disturbance-noise vector at time k with zero mean and known statistics

$v(k)$ is an $\ell \times 1$ observation-noise vector at time k with zero mean and known statistics

$z(k)$ is an $r \times 1$ observation vector at time k

$I(k)$ is an $m \times 1$ interference vector with known statistics.

Time k is time $t_o + kT_o$ where t_o and T_o are the initial time and the observation interval, respectively.

$f(k,x(k),u(k),w(k))$, $g(k,x(k),I(k),v(k))$, and $g(k,x(k),v(k))$ are (linear or nonlinear) vectors with appropriate dimensions.

Furthermore, the random vectors $x(0)$, $w(j)$, $w(k)$, $v(\ell)$, $v(m)$, $I(n)$, and $I(p)$ are assumed to be independent for all j, k, ℓ, m, n, p.

2.2 Quantization of States and Transition Probabilities

This section describes a kind of quantization for target states and some difficulties in calculating transition probabilities between quantization levels.

Let us consider the state $x(k)$. It is a random vector whose range is in the Space R^n (n-dimensional Euclidean space). Let us divide R^n into nonoverlapping subspaces, R_i^n's, and assign a unique value x_{qi} to each subspace R_i^n where subscript q stands for quantization.

Definition 2.2.1 A function $x_q(\cdot) \triangleq Q\{x(\cdot)\}$ is a quantizer for the state $x(\cdot)$ if the following hold

(a) $x_q(\cdot) \triangleq Q\{x(\cdot)\} = x_{qi}$ whenever $x(\cdot) \in R_i^n$.

(b) x_{qi} is unique for each R_i^n.

Definition 2.2.2 $x_q(\cdot)$ is the quantized state (vector) at time $\cdot$, and its possible values are called the quantization levels of the state $x(\cdot)$.

Definition 2.2.3 Subspace R_i^n is sometimes called Gate (or Cell) R_i^n.

Definition 2.2.4 x_{qi} is the quantization level for Gate (Cell) R_i^n.

Quantization means that whenever a random state vector $x(\cdot)$ falls within a given subspace, say R_i^n, the state $x(\cdot)$ is quantized to the unique value x_{qi} (see Figure 2.2.1). Let us now define the transition probabilities, which govern the target motion within the gates.

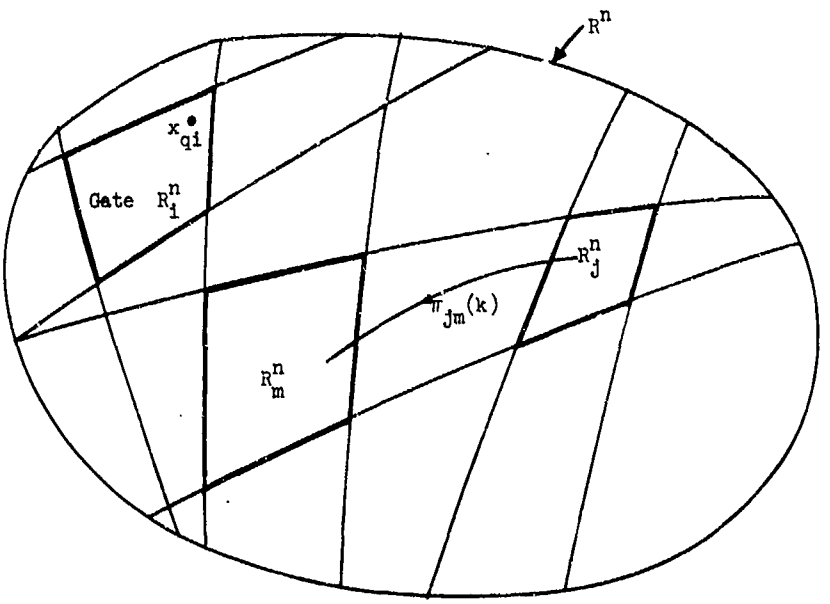


Figure 2.2.1. Quantization and Transition Probabilities

Definition 2.2.5 The transition probability $\pi_{jm}(k)$ is the probability that the state $x(k+1)$ will lie in the gate R_m^n when the state $x(k)$ is in the gate R_j^n, i.e.,

$$\pi_{jm}(k) \triangleq \text{Prob}\{x(k+1) \in R_m^n | x(k) \in R_j^n\} . \qquad (2.2.1)$$

By definition, the conditional probability $\pi_{jm}(k)$ can be rewritten as

$$\pi_{jm}(k) = \frac{\text{Prob}\{x(k+1) \in R_m^n,\ x(k) \in R_j^n\}}{\text{Prob}\{x(k) \in R_j^n\}}$$

$$= \frac{1}{\int_{R_j^n} p(x(k))\, dx(k)} \left[\int_{R_j^n} \int_{R_m^n} p(x(k+1),\ x(k))\, dx(k+1)\, dx(k)\right]$$

$$= \frac{1}{\int_{R_j^n} p(x(k))\, dx(k)} \left\{\int_{R_j^n} \left[\int_{R_m^n} p(x(k+1)\,|\,x(k))\, dx(k+1)\right] p(x(k))\, dx(k)\right\} \qquad (2.2.2)$$

where

$p(x(k+1),\ x(k))$ is the joint probability density function of $x(k+1)$ and $x(k)$

$p(x(k))$ is the probability density function of $x(k)$

$p(x(k+1)\,|\,x(k))$ is the conditional probability density function of $x(k+1)$ given $x(k)$.

It is not usually easy to evaluate the transition probability $\pi_{jm}(k)$ analytically. The difficulties are due to the shapes of the gates (R_j^n and R_m^n), the statistics of the disturbance noise vectors ($w(\cdot)$'s), and the initial state vector $x(0)$. In order to see this, consider the following linear motion example

$$x(k+1) = Ax(k) + w(k) \qquad (2.2.3)$$

where

$x(0)$ is an n x 1 Gaussian initial state vector

$x(k)$ is an n x 1 state vector at time k

$w(k)$ is an n x 1 Gaussian disturbance vector at time k

A is a constant transition matrix with appropriate dimension.

Moreover, the random vectors $x(0)$, $w(k)$, $w(\ell)$ are assumed to be statistically independent for all k, ℓ. Hence $x(k+1)$ and $x(k)$ are linear transformations of the Gaussian random vectors $x(0)$, $w(0)$, $w(1)$, ..., and $w(k)$. Thus, $p(x(k))$ and $p(x(k+1)\,|\,x(k))$ are normal density functions. Therefore the evaluation of the probability

$$p\{x(k+1) \in R_m^n \,|\, x(k) \in R_j^n\}$$

is not analytically possible. The problem is more difficult if the motion model is not linear. If the transition probability $\pi_{jm}(k)$ needs to be calculated, it should be performed numerically. Even this may be difficult. In other words, the evaluation of the exact transition probabilities between gates is not practical. Therefore, the next section discusses an approximate target motion model obtained by approximating the disturbance noise vector $w(k)$ and the initial state vector $x(0)$ by discrete random vectors (see Appendix B), and by quantizing the state $x(k)$, as described above, for all $k = 1, 2, \ldots$. For this finite state model, the transition probabilities can be calculated easily.

2.3 A Finite State Model for the Target Motion

Throughout this section, gates are assumed to be generalized rectangles such that the zero vector 0 (origin) is located in the center of a generalized rectangle, say R_0^n (see Figure 2.3.1).

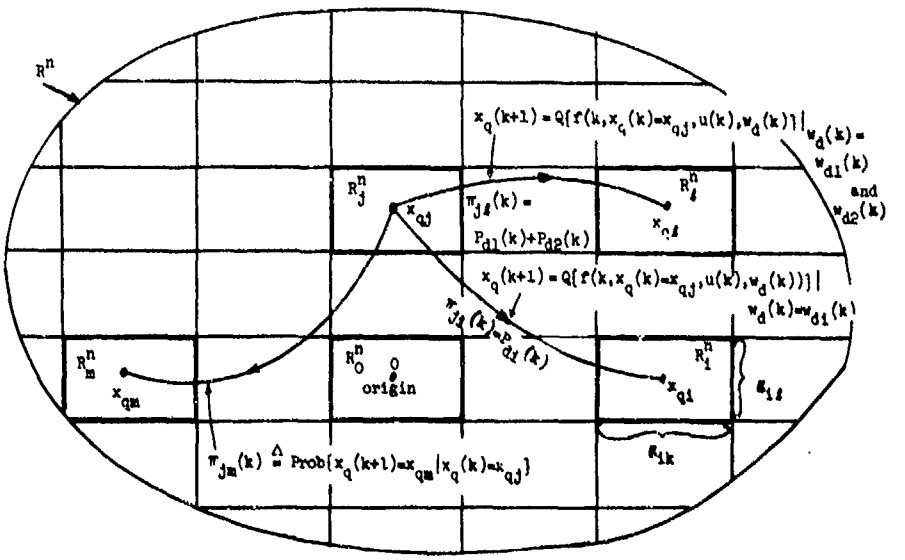


Figure 2.3.1. Quantization with Generalized Rectangles

Let the lengths of the sides of a generalized rectangle, say R_i^n, be $g_{i1}, g_{i2}, \ldots, g_{in}$. These lengths are said to be the sizes of Gate R_i^n. Moreover, the quantization levels for gates are assumed to be the center points of the gates, namely

$$x_q(\cdot) \triangleq Q\{x(\cdot)\} = x_{qi} \qquad \text{if} \qquad x(\cdot) \in R_i^n \tag{2.3.1}$$

where x_{qi} is the center of the generalized rectangle (gate) R_i^n.

Let us now define the finite state model which approximates the target motion model. The flow chart of this finite state model is in Figure 2.3.2. For each k, the disturbance noise vector w(k)

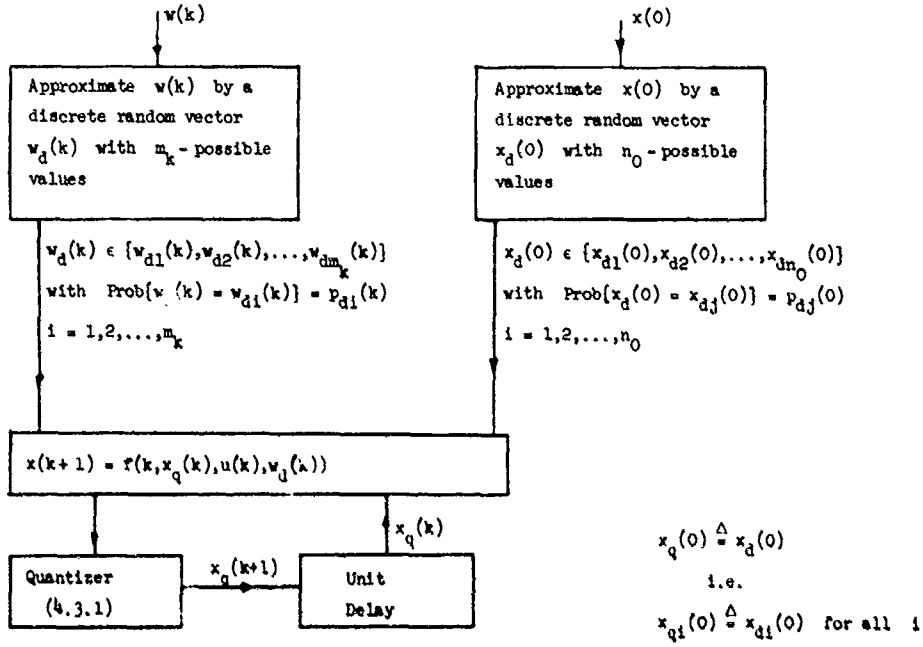


Figure 2.3.2. The Flow Chart of the Finite State Model

is approximated by a discrete random vector $w_d(k)$ whose possible values are $w_{d1}(k)$, $w_{d2}(k)$, ..., $w_{dm_k}(k)$; the corresponding probabilities are $p_{d1}(k)$, $p_{d2}(k)$, ..., $p_{dm_k}(k)$, i.e.,

$$\text{Prob}(w_d(k) = w_{di}(k)) = p_{di}(k) \qquad i = 1, 2, \ldots, m_k$$

where m_k is a positive integer, and the subscript d stands for discrete (see Appendix B). Also the initial state vector x(0) is approximated by a discrete random vector $x_d(0)$ whose possible values are $x_{d1}(0)$, $x_{d2}(0)$, ..., $x_{dn_0}(0)$; the corresponding probabilities are $p_{d1}(0)$, $p_{d2}(0)$, ..., $p_{dn_0}(0)$, i.e.,

$$\text{Prob}(x_d(0) = x_{di}(0)) = p_{di}(0) \qquad i = 1, 2, \ldots, n_0$$

where n_0 is a positive integer. Further, in the (target) motion model, replacing the disturbance noise vector w(k) and the initial state vector x(0) by the discrete random vectors $w_d(k)$ and $x_d(0)$ respectively, and then quantizing the states by Quantizer (2.3.1), the target motion model is reduced to the finite state model

$$x_q(k+1) = Q\{f(k, x_q(k), u(k), w_d(k))\} \tag{2.3.2}$$

where

$Q(\cdot)$ is Quantizer (2.3.1)

$x_q(k)$ is the quantized state vector at time k, and its possible values (i.e., the quantization levels of the state vector x(k)) are $x_{q1}(k)$, $x_{q2}(k)$, ..., $x_{qn_k}(k)$ where n_k is the number of possible quantization levels of the state vector x(k)

$x_q(0) \triangleq x_d(0)$, (by definition, $x_{qi}(0) \triangleq x_{di}(0)$, i = 1, 2, ..., n_0; in other words, the quantization levels of x(0) are assumed to equal the possible values of the discrete random vector x(0))

u(k) is a known pilot-command vector at time k.

Throughout this chapter whenever the target motion model (or the state equations, or the target motion) is mentioned we refer to Model (2.3.2); i.e., it is assumed that the target motion is described by (Finite State) Model (2.3.2).

The transition probability $\pi_{jl}(k)$, which is defined by the conditional probability that the

quantized state vector $x_q(k+1)$ will be equal to the quantization level $x_{q\ell}$ for Gate R_ℓ^m, given that the quantized state vector $x_q(k)$ is equal to the quantization level x_{qj} for Gate R_j^m, namely,

$$\pi_{j\ell}(k) = \text{Prob}\{x_q(k+1) = x_{q\ell} | x_q(k) = x_{qj}\} \tag{2.3.3}$$

is determined as follows (see Figure 2.3.1).

Let us assume that the quantized state vector $x_q(k)$ is equal to the quantization level x_{qj} for Gate R_j^n (i.e., the target is in R_j^n at time k). The transitions from this quantization level to the others are determined by the discrete random vector $w_d(k)$ and the function $Q\{f(k, x_q(k) = x_{qj}, u(k), w_d(k))\}$. The discrete random vector $w_d(k)$ can take any value in the set $\{w_{d1}(k), w_{d2}(k), \ldots, w_{dm_k}(k)\}$ with corresponding probabilities $p_{d1}(k), p_{d2}(k), \ldots, p_{dm_k}(k)$. Thus, the quantized state vector $x_q(k+1)$ may be equal to at most m_k different quantization levels. If the function $f(k, x_q(k) = x_{qj}, u(k), w_d(k))$ maps x_{qj} into another gate, say R_i^n for only one possible value, say $w_{di}(k)$, of the discrete random vector $w_d(k)$, then the transition probability $\pi_{ji}(k)$ (from Gate R_j^n to Gate R_i^n) is the probability that the possible value $w_{di}(k)$ of $w_d(k)$ occurs, i.e.,

$$\pi_{ji}(k) = p_{di}(k)$$

However, if the function $f(k, x_q(k) = x_{qj}, u(k), w_d(k))$ maps x_{qj} into another gate, say R_ℓ^n, for more than one possible value (say $w_{d1}(k)$ and $w_{d2}(k)$) of $w_d(k)$, the transition probability $\pi_{j\ell}(k)$ (from Gate R_j^n to Gate R_ℓ^n) is the probability that the discrete random vector $w_d(k)$ is equal to one of these possible values ($w_{d1}(k)$ or $w_{d2}(k)$), i.e.,

$$\pi_{j\ell}(k) = \sum_n p_{dn}(k) = p_{d1}(k) + p_{d2}(k)$$

where the summation is over all n such that

$$Q\{f(k, x_q(k) = x_{qj}, u(k), w_{dn}(k))\} = x_{q\ell} \quad .$$

Having determined the finite state model, we can represent the target motion by a diagram called a Trellis diagram for the target motion.

2.4 A Trellis Diagram for the Target Motion

Let us assume that the quantized state vector $x_q(k)$ has n_k possible values, say, $x_{q1}(k), x_{q2}(k), \ldots, x_{qn_k}(k)$ where n_k is a positive integer. To represent the target motion by a graph, we adopt the following conventions

(1) Each possible value of $x_q(k)$ is represented on the k^{th} column by a point (sometimes called node) with the corresponding quantization level so that the k^{th} column contains the possible quantization levels of $x_q(k)$ (in other words, the possible gates in which the target can lie at time k) where $k = 0, 1, 2, \ldots$

(2) The transition from one quantization level to another is represented by a line having a direction indicating the direction of the target motion.

Hence, the target motion from time zero to time L can be represented by a directed graph shown in Figure 2.4.1, which is called the trellis diagram for the target motion from time zero to time L.

Definition 2.4.1 A path in the trellis diagram is any sequence of directed lines where the final vertex of one is the initial vertex of the next one.

2.5 Approximate Observation Models

So far the target motion model has been reduced to a finite state model which uses the quantized state vectors ($x_q(\cdot)$'s). However, the observation models in (2.1.1) and (2.1.2) use the target state vectors ($x(\cdot)$'s). Thus, in the observation models in (2.1.1) and (2.1.2), replacing the state vector $x(k)$ by the quantized state vector $x_q(k)$, the following approximate observation models are obtained

$$z(k) = \begin{cases} g(k, x_q(k), v(k)) & \text{in clear environments} \\ g(k, x_q(k), I(k), v(k)) & \text{in the presence of interference} \end{cases} \tag{2.5.1}$$

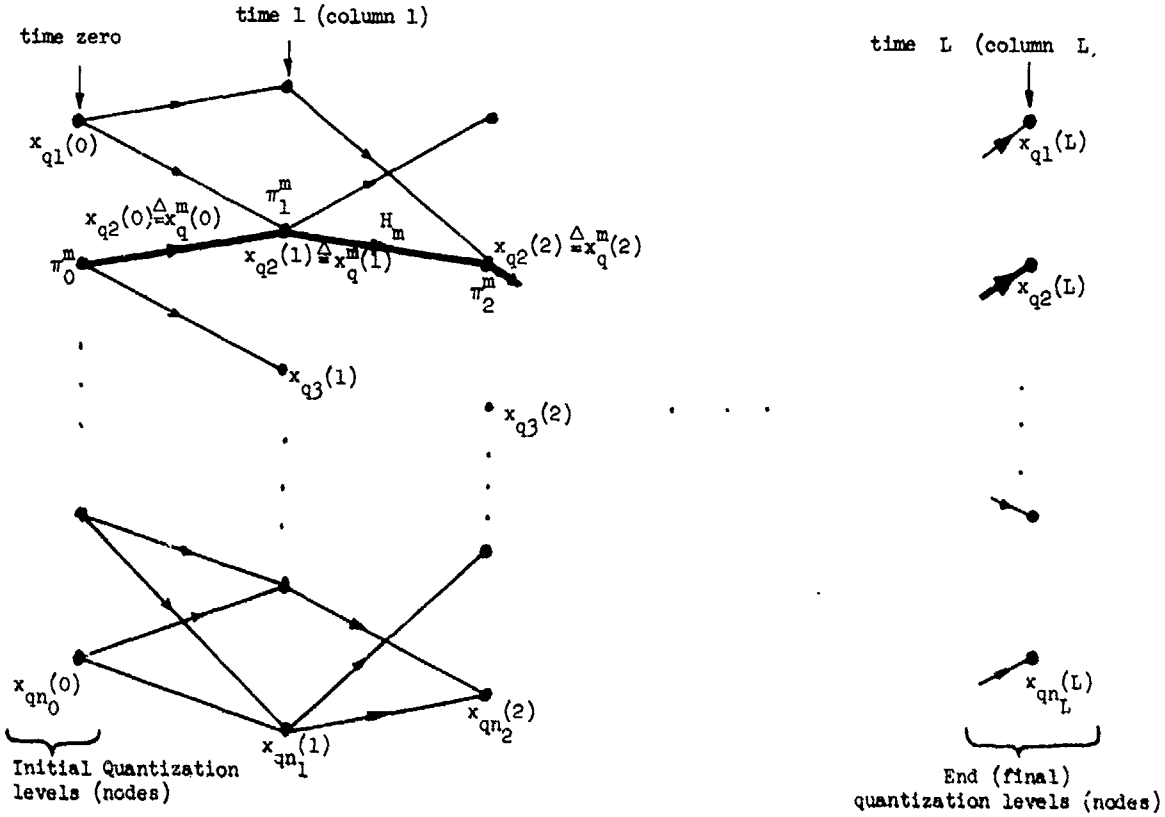


Figure 2.4.1. The Trellis Diagram for the Target Motion

From now on, whenever the observation model (or the measurement model (equations)) is mentioned, we refer to the models in (2.5.1). The observation models in (2.5.1) are used in the following analyses.

Let us consider the trellis diagram in Figure 2.4.1 where it is assumed that, without loss of generality, the target will be tracked from time zero up to and including time L. Therefore, the trellis diagram is drawn from time zero to time L. Time zero refers to the initial state. Let us now define the following which will be used throughout our further analyses.

n_i is the number of quantization levels for the gates in which the target may lie at time i, in other words, the number of possible values of the quantized state vector $x_q(i)$ where $i = 0, 1, 2, \ldots, L$

$\underline{x}(i)$ is the set of all the quantization levels for the gates in which the target may lie at time i, namely,

$$\underline{x}(i) \triangleq (x_{q1}(i), x_{q2}(i), \ldots, x_{qn_i}(i))$$

where

$$i = 0, 1, 2, \ldots, L$$

M is the number of possible paths through the trellis diagram; this number is less than or equal to

$$\prod_{j=0}^{L} n_j$$

H_m is the m^{th} path through the trellis diagram, which is indicated by a thick line

$x_q^m(i)$ is the quantization level for the gate in which the target lies at time i when it follows path H_m. In other words, the possible value of the quantized state vector $x_q(i)$ which the m^{th} path passes through. For example, in Trellis diagram 2.4.1,

$$x_q^m(0) = x_{q2}(0),\ x_q^m(1) = x_{q2}(1),\ x_q^2(2) = x_q^m(2),\ \ldots$$

π_0^m is the probability that the possible value of the initial state vector $x_q(0)$ from which the m^{th} path starts occurs, namely, $\pi_0^m = \text{Prob}\{x_q(0) = x_q^m(0)\}$. For example, in Trellis diagram 2.4.1

$$\pi_0^m = \text{Prob}\{x_q(0) = x_{q2}(0)\}$$

π_i^m is the transition probability from the $(i-1)^{th}$ gate for the m^{th} path (i.e., the gate from which the target passes at time i-1 when it follows path H_m) to the i^{th} gate for the m^{th} path. In other words, it is the transition probability that the target will be at the i^{th} quantization level (node) of path H_m at time i when it is at the i-1 quantization level (node) of H_m at time i-1, that is, $\pi_I^m \triangleq \text{Prob}\{x_q(i) = x_q^m(i) | x_q(i-1) = x_q^m(i-1)\}$. For example, in Trellis diagram 2.4.1

$$\pi_1^m = \text{Prob}\{x_q(1) = x_q^m(1) | x_q(0) - x_q^m(0)\}$$

$$\triangleq \text{Prob}\{x_q(1) = x_{q2}(1) | x_q(0) = x_{q2}(0)\}$$

$$\pi_2^m = \text{Prob}\{x_q(2) = x_q^m(2) | \, x_q(1) = x_q^m(1)\}$$

$$\triangleq \text{Prob}\{x_q(2) = x_{q2}(2) | x_q(1) = x_{q2}(1)\}$$

π_0^{max} is the maximum of the probabilities that the quantization levels at time zero occur, i.e.,

$$\pi_0^{max} \triangleq \max_{a \in \underset{\sim}{x}(0)} \text{Prob}\{x_q(0) = a\}$$

π_i^{max} is the maximum of the transition probabilities from the quantization levels at time i-1 to the quantization levels at time i (where i = 1, 2, ..., L), that is,

$$\pi_i^{max} = \max_{\substack{a \in \underset{\sim}{x}(i) \\ b \in \underset{\sim}{x}(i-1)}} \text{Prob}\{x_q(i) = a | x_q(i-1) = b\}$$

π_0^{min} is the minimum of the probabilities that the quantization levels at time zero occur, i.e.,

$$\pi_0^{min} = \max_{a \in \underset{\sim}{x}(0)} \text{Prob}\{x_q(0) = a\}$$

π_i^{min} is the minimum of the transition probabilities from the quantization levels at time i-1 to the quantization levels at time i (where i = 1, 2, ..., L), namely,

$$\pi_i^{min} = \min_{\substack{a \in \underset{\sim}{x}(i) \\ b \in \underset{\sim}{x}(i-1)}} \text{Prob}\{x_q(i) = a | x_q(i-1) = b\}$$

$\underset{\sim}{x}_L^m \triangleq (x_q^m(0), x_q^m(1), \ldots, x_q^m(L))$ which is the sequence of the quantization levels (nodes) which the m^{th} path passes through, obviously,

$$x_q^m(i) \in \underset{\sim}{x}(i) \qquad i = 0, 1, 2, \ldots, L$$

$z^L = (z(1), z(2), \ldots, z(L))$ is the observation sequence from time 1 to time L

$I^L \triangleq (I(1), I(2), \ldots, I(L))$ is the interference sequence from time 1 to time L.

Obviously, the target motion occurs along one of the possible paths in the trellis diagram. Hence our aim is to decide a path in the trellis diagram, which is most probably followed by the target, by using the observation sequence z^L. Because of randomness in the models, our approach must be statistical, i.e., a statistical optimization problem. Based on the observations, we shall guess which path was followed by the target. Hence, a criterion is needed. For a tracking problem, a suitable criterion may be the minimum error probability criterion, which is a special case of Bayes criterion in Detection Theory. Using this criterion reduces the problem of finding the path most probably followed by the target to a (Composite) Multiple Hypothesis Testing problem.

2.6 Minimum Error Probability Criterion

In the previous section, we labeled the M possible paths through the trellis diagram H_1, H_2, ..., H_M. Sometimes these paths are referred to as Hypotheses. Hence, using the minimum error probability criterion and the observation sequence, we would like to decide which hypotheses is true (in other words, we would like to find the path most porobably followed by the target). To accomplish this we develop a decision rule that assigns each point in the observation space D to one of the hypotheses. Therefore, we can view the decision rule as dividing the whole observation space D into M subspaces D_1, D_2, ..., and D_M (see Figure 2.6.1). If the observations fall in the subspace D_i, we decide that the target followed path H_i (i.e., H_i is true). Subspace D_i is called the decision region for Hypothesis H_i. Therefore, we must choose the decision regions D_1, D_2, ..., and D_M in such a way that the overall error probability is minimized.

The overall error probability, sometimes called the Bayes risk (R), is defined by

$$R \triangleq \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \left(\int_{z^L \in D_i} p(H_j) \, p'(z^L | H_j) \, dz^L \right) \tag{2.6.1}$$

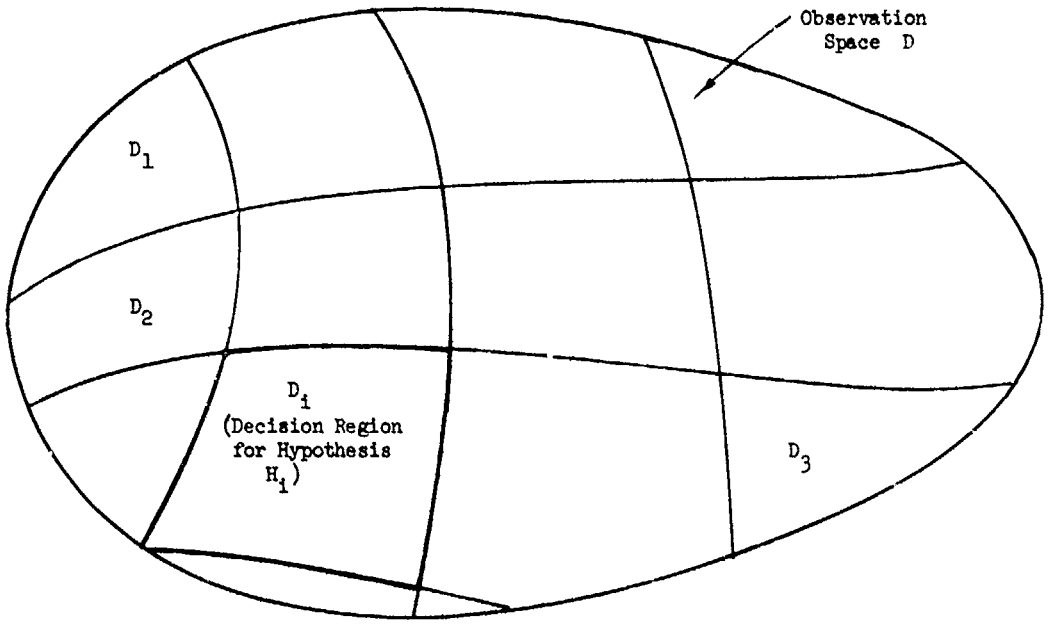


Figure 2.6.1. Observation Space and Decision Regions

where

$$p'(z^L|H_j) = \begin{cases} p(z^L|H_j) & \text{in clear environments} \\ \int_{I^L} p(z^L|H_j, I^L)\, p(I^L)\, dI^L & \text{in the presence of interference} \end{cases} \tag{2.6.2}$$

$p(H_j)$ is the probability that Hypothesis H_j (the path H_j) is true, and it is called the a priori probability of Hypothesis H_j

$p(z^L|H_j)$ is the conditional probability of the observation sequence z^L in clear environments, given that Hypothesis H_j is true (i.e., the target followed the path H_j)

$p(z^L|H_j, I^L)$ is the conditional probability of the observation sequence z^L in the presence of interference, given Hypothesis H_j and the interference sequence I^L

$p(I^L)$ is the joint density function of the interference sequence I^L.

In order to find the optimal decision rule, we vary the decision regions $D_1, D_2, \ldots,$ and D_M so that the risk R is minimized. It is well known that the optimum decision rule[14] is

$$\text{choose } H_i \text{ if } p(H_i)\, p'(z^L|H_i) > p(H_j)\, p'(z^L|H_j) \quad \text{for all } j \neq i \tag{2.6.3}$$

For a given observation sequence z^L, if the inequality in (2.6.3) becomes an equality for one or more Hypotheses, H_j's, any one of these H_j's and H_i can be chosen as the decision. This does not change the average error probability. Throughout this chapter for all observation sequences, z^L's, for which the inequality in (2.6.3) becomes an equality, the decision is made at random (e.g., by the flip of a fair dime) among the hypotheses satisfying the equality. Hence the optimum decision region D_i for Hypothesis H_i becomes

$$D_i \triangleq \{z^L:\ p(H_i)\, p'(z^L|H_i) > p(H_j)\, p'(z^L|H_j) \text{ for all } j \neq i, \text{ and all the observation sequences, } z^L\text{'s, for which the inequality in (2.6.3) becomes an equality and then Hypothesis } H_i \text{ has been chosen}\}. \tag{2.6.4}$$

It should be noted that the decision regions are nonoverlapping, namely,

$$D_i \cap D_j = \phi \quad \text{for } i \neq j$$

where $\cap$ and ϕ stand for intersection and the empty set respectively. However, the union of all the decision regions cover the whole observation space D, that is,

$$D = \bigcup_{i=1}^{M} D_i$$

Hence, the optimum decision rule may be interpreted as follows: If the observation sequence z^L falls within the optimum decision region D_i, then choose Hypothesis (path) H_i as the decision, i.e.,

$$\text{choose} \quad H_i \quad \text{if} \quad z^L \in D_i \tag{2.6.5}$$

Having determined the optimum decision rule (2.6.3) (or (2.6.5)) with respect to the minimum error probability criterion, we apply it to tracking problems in the next section.

2.7 Optimum Decision Rule for the Target Paths

Let us consider the motion models in both (2.1.1) and (2.1.2) and the observation models in (2.5.1). The a priori probability of Hypothesis H_i can be rewritten as

$$p(H_i) = \prod_{k=0}^{L} \pi_k^i \tag{2.7.1}$$

since the disturbance noise vector $w(k)$ is assumed to be independent of $w(j)$ and $x(0)$ for all $j \neq k$, where π_k^i is as defined in Section 2.5. Further, using the assumption that the interference vector $I(k)$ is independent of $I(j)$ for all $k \neq j$, we can rewrite the joint density function of the interference sequence I^L as

$$p(I^L) = \prod_{k=1}^{L} p(I(k)) \tag{2.7.2}$$

where $p(I(k))$ is the probability density function of the interference vector $I(k)$. Moreover, recognizing that the sequence $\underline{x}_L^m$, defined in Section 2.5, describes Hypothesis H_i completely, and using (2.7.2) and the assumption that the observation noise is independent from sample to sample, the function $p'(z^L|H_i)$ in (2.6.3) can be rewritten as

$$\begin{aligned} p'(z^L|H_i) &= p'(z^L|\underline{x}_L^i) \\ &= \prod_{k=1}^{L} p'(z(k)|x_q^i(k)) \end{aligned} \tag{2.7.3}$$

where

$$p'(z(k)|x_q^i(k)) = \begin{cases} p(z(k)|x_q^i(k)) & \text{in clear environments} \\ \int_{I(k)} p(z(k)|x_q^i(k), I(k))\, p(I(k))\, dI(k) & \text{in the presence of interference} \end{cases} \tag{2.7.4}$$

$p(z(k)|x_q^i(k))$ is the conditional probability of the observation $z(k)$ in clear environments in (2.5.1), given that $x_q(k) = x_q^i(k)$, i.e.,

$$p(z(k)|x_q^i(k)) \triangleq p(z(k)|x_q(k) = x_q^i(k))$$

$p(z(k)|x_q^i(k), I(k))$ is the conditional probability of the observation $z(k)$ in the presence of interference in (2.5.1), given that $x_q(k) = x_q^i(k)$ and $I(k)$, that is

$$p(z(k)|x_q^i(k), I(k)) = p(z(k)|x_q(k) = x_q^i(k), I(k))$$

Let us now consider the function $p'(z(k)|x_q^i(k))$ in the presence of interference, that is

$$p'(z(k)|x_q^i(k)) = \int_{I(k)} p(z(k)|x_q^i(k), I(k))\, p(I(k))\, dI(k) \tag{2.7.5}$$

whether or not this integral can be evaluated in a closed form depends on the function $g(k, x_q(k) = x_q^i(k), I(k), v(k))$, and the statistics of the interference $I(k)$ and the observation noise $v(k)$. In many cases, a numerical integration might be used to evaluate it. However, throughout this chapter, approximating the interference vector $I(k)$ by a discrete random vector $I_d(k)$ whose possible values are $I_{d1}(k)$, $I_{d2}(k)$, ..., $I_{dr_k}(k)$, with corresponding probabilities $p(I_{d1}(k))$, $p(I_{d2}(k))$, ..., and $p(I_{dr_k}(k))$, i.e.,

$$\text{Prob}\{I_d(k) = I_{di}(k)\} = p(I_{di}(k))$$

the integral in (2.7.5) is reduced to a summation

$$\int_{I(k)} p(z(k)|x_q^i(k),\ I(k))\ p(I(k))\ dI(k) \simeq \sum_{\ell=1}^{r_k} p(z(k)|x_q^i(k),\ I_{d\ell}(k))\ p(I_{d\ell}(k)) \qquad (2.7.6)$$

where r_k is the number of possible values of the approximating discrete vector $I_d(k)$. In other words, by chainging the interference I(k) to $I_d(k)$ we make another approximation for the observation model in the presence of interference in (2.5.1). The observation model becomes

$$\begin{aligned} z(k) &= g(k,\ x_q(k),\ I(k) = I_d(k),\ v(k)) \\ &\triangleq g(k,\ x_q(k),\ I_d(k),\ v(k)). \end{aligned} \qquad (2.7.7)$$

From now on, throughout the chapter, if it is not easy to calculate the integral in (2.7.5), the integral will be approximated by (2.7.6). In other words, when the integral in (2.7.5) can not be easily evaluated, Observation model (2.7.7), (instead of the observation model in (2.5.1)), will be used for further analyses.

Substituting (2.7.1) and (2.7.3) into Optimum Decision Rule (2.6.3), we obtain

$$\text{Choose}\quad H_i\quad \text{if}\quad \pi_0^i \prod_{k=1}^{L} \pi_k^i\ p'(z(k)|x_q^i(k)) > \pi_0^j \prod_{k=1}^{L} \pi_k^j\ p'(z(k)|x_q^j(k))\quad \text{for all } j\neq i \qquad (2.7.8)$$

However, it is frequently more convenient to perform summations than multiplications. Since the ln function is a monotone increasing function, taking the natural logarithms of both sides of the inequalities in (2.7.8), we get

$$\text{Choose } H_i \text{ if } \ln \pi_0^i + \sum_{k=1}^{L} \{\ln \pi_k^i + \ln p'(z(k)|x_q^i(k))\} > \ln \pi_0^j + \sum_{k=1}^{L} \{\ln \pi_k^j + \ln p'(z(k)|x_q^j(k))\} \quad \text{for all } j \neq i \qquad (2.7.9)$$

where

$$p'(z(k)|x_q^i(k)) = \begin{cases} p(z(k)|x_q^i(k)) & \text{in clear environments} \\ \left.\begin{array}{ll} \int p(z(k)|x_q^i(k), I(k))\ p(I(k))\ dI(k) & \text{using (2.5.1)} \\ \sum_{\ell=1}^{r_k} p(z(k)|x_q^i(k), I_{d\ell}(k))\ p(I_{d\ell}(k)) & \text{using (2.7.7)} \end{array}\right\} & \text{in the presence of interference} \end{cases} \qquad (2.7.10)$$

$p(z(k)|x_q^i(k), I_{d\ell}(k)) = p(z(k)|x_q(k) = x_q^i(k), I(k) = I_{d\ell}(k))$ which is the conditional probability of z(k) in (2.7.7), given that $x_q(k) = x_q^i(k)$ and $I_d(k) = I_{d\ell}(k)$.

Either one of the expressions in (2.7.8) and (2.7.9) with the convention in Section 2.6 is the optimum decision rule for deciding the path most probably followed by the target.

Now we are going to verify the following equalities for the obeservatico models in the presence of interference in (2.5.1) and (2.7.7).

$$\begin{aligned} p'(z(k)|x_q^i(k)) &= p(z(k)|x_q^i(k)) \\ p(z^L|H_i) &= \prod_{k=1}^{L} p(z(k)|x_q^i(k)) \\ &= p'(z^L|H_i) \end{aligned} \qquad (2.7.11)$$

where

$p(z(k)|x_q^i(k))$ is the conditional probability of the observation z(k) in the presence of interference, given that $x_q(k) = x_q^i(k)$

$p(z^L|H_i)$ is the conditional probability of the observation sequence z^L in the presence of interference, given that Hypothesis H_i is true (i.e., the target followed the path H_i).

Let us see this fact for the observation model in the presence of interference in (2.5.1). From (2.7.10), we have

$$p'(z(k)|x_q^i(k)) = \int_{I(k)} p(z(k)|x_q^i(k),\ I(k))\ p(I(k))\ dI(k) \tag{2.7.12}$$

Using Bayes' rule and the assumption that the interference I(k) is independent of the initial state vector x(0) and of the disturbance noise vector w(j) for all j, k. We obtain

$$p(z(k)|x_q^i(k),\ I(k)) = \frac{p(z(k),\ x_q^i(k),\ I(k))}{p(x_q^i(k))\ p(I(k))} \tag{2.7.13}$$

Hence, substituting (2.7.13) into (2.7.12) and recalling that the integration is taken over all the sample space of I(k), yields the first equality in (2.7.11). The second equality in (2.7.11) follows from the assumption that the interference and the observation noise are independent from sample to sample. The same equalities in (2.7.11) can be verified for Observation Model (2.7.7). Hence for all observation models already considered, the function $p'(\cdot|\cdot)$ is a conditional probability density function.

Let us give some definitions which will be used later on

Definition 2.7.1 The metric, denoted by $MN(x_{qi}(0))$, of the initial node $x_{qi}(0)$ is defined by

$$MN(x_{qi}(0)) = \ln[\text{Prob}(x_q(0) = x_{qi}(0))] \tag{2.7.14}$$

Consequently

$$MN(x_q^m(0)) = \ln \pi_0^m$$

Definition 2.7.2 The metric, denoted by $M(x_{qj}(k-1) \to x_{qi}(k))$, of the branch which connects the quantization level (node) $x_{qj}(k-1)$ to the quantization level $x_{qi}(k)$ is defined by

$$M(x_{qj}(k-1) \to x_{qi}(k)) \triangleq \ln[\text{Prob}(x_q(k) = x_{qi}(k)|x_q(k-1) = x_{qj}(k-1))] + \ln p'(z(k)|x_{qi}(k)) \tag{2.7.15}$$

Definition 2.7.3 The metric of a path from time zero to time i is the summation of the metric of the initial node from which the path starts and of the metrics of the branches which the path consists of. For example, the metric, denoted by $M(x_q^m(i))$, of the portion between the nodes $x_q^m(0)$ and $x_q^m(i)$ of the path (hypothesis) H_m is

$$M(x_q^m(i)) = \ln \pi_0^m + \sum_{k=1}^{i} [\ln \pi_k^m + \ln p'(z(k)|x_q^m(k))] \tag{2.7.16}$$

Consequently, the metric, sometimes denoted by $M(H_m)$, of the path H_m (through the trellis) is

$$M(H_m) \triangleq M(x_q^m(L)) = \ln[p(H_m)\ p'(z^L|H_m)] \tag{2.7.17}$$

where $x_q^m(L)$ is the end node of the path H_m, $p(H_m)$ and $p'(z^L|H_m)$ are given by (2.7.1), (2.7.3) and (2.7.10).

Definition 2.7.4 The error probability of a path, say H_m, in a trellis diagram T with N possible paths $H_1, H_2, \ldots, H_N$ is the probability of deciding a path which is different than H_m as the one most probably followed by the target when the target actually followed the path H_m. This error probability is denoted by either $Pg_m(H_1, H_2, \ldots, H_N)$ or $Pg_m(T)$ where subscripts E and m stand for error and the m^{th} path. Hence

$$P_{E_m}(H_1, \ldots, H_M) \triangleq P_{E_m}(T)$$

$$\triangleq \text{Prob}\{z^L \in \bar{D}_m | H_m\}$$

$$= \int_{z^L \in \bar{D}_m} p(z^L|H_m)\, dz^L \qquad (2.7.18)$$

where $\bar{D}_m$ is the complement of the decision region, D_m, for the path H_m, and $p(z^L|H_m)$ is the probability density function of the observation sequence z^L when the target actually followed the path H_m. Hence from (2.6.1), the overall error probability for the detection of the path most probably followed by the target (denoted by R, P_E, $P_E(H_1, \ldots, H_M)$, or $P_E(T)$) can be expressed in terms of the path error probabilities as follows

$$P_E = \sum_{m=1}^{M} p(H_m)\, P_{E_m}(H_1, H_2, \ldots, H_M) \qquad (2.7.19)$$

where $p(H_m)$ is given by (2.7.1).

Definition 2.7.5 The density function of the observation sequence z^L when the target actually followed the path H_m (i.e., $p(z^L|H_m)$) is referred to as the likelihood function for the path (hypothesis) H_m.

Therefore, the optimum decision rule for deciding the path most probably followed by the target from time zero to time L lead us to choosing the path (from time zero to time L) with largest metric (in the trellis diagram). This can be handled by using the Viterbi Decoding Algorithm (VDA)[15,16], which is the optimum decoding algorithm. The algorithm which obtains the trellis diagram for the target motion (model), as described before, and which finds the path most likely followed by the target by using VDA is referred to as Optimum Decoding Based Smoothing Algorithm (ODSA).

2.8 OPTIMUM DECODING BASED SMOOTHING ALGORITHM

Initial Step - reducing the target motion model to a finite state model, as described before, obtain a trellis diagram for the target motion (model) from time zero to the time, say L, until which the target will be tracked. Then assign to each node its metric.

First Step - for each node at time one: Using the observation z(1), evaluate the metrics of the branches connecting the initial nodes to the node at time one; adding these metrics to the metrics of the initial nodes from which the branches start, find the metrics of the paths merging at the node at time one, and label the path with largest metric (which is called the best path for the node at time one), then discard the other paths. Finally assign the largest metric to the node at time one (which is called the metric of the node at time one).

k^{th} Step - for each node at time k: Using the observation at time k, calculate the metrics of the branches connecting the nodes at time k-1 to the node at time k; adding these metrics to the metrics of the nodes at time k-1 from which the branches start, find the metrics of the paths merging at the node at time k and label the path with the largest metric (which is called the best path for the node at time k), then discard the other paths. Finally assign the largest metric to the node at time k (which is called the metric of the node at time k).

If k = L, stop, and choose among the nodes at time L the one with the largest metric, then decide the best path for this node as the path followed by the target.

The following section illustrates the optimum decoding based smoothing algorithm by an example.

2.8.1 An example

Let us consider a target whose motion from time zero to time 2 is described by Figure 2.8.1. Using ODSA, we would like to find the path in the trellis diagram which was most probably followed by the target from time zero to time 2.

Initial Step - To each node at time zero, its metric is assigned, i.e.,

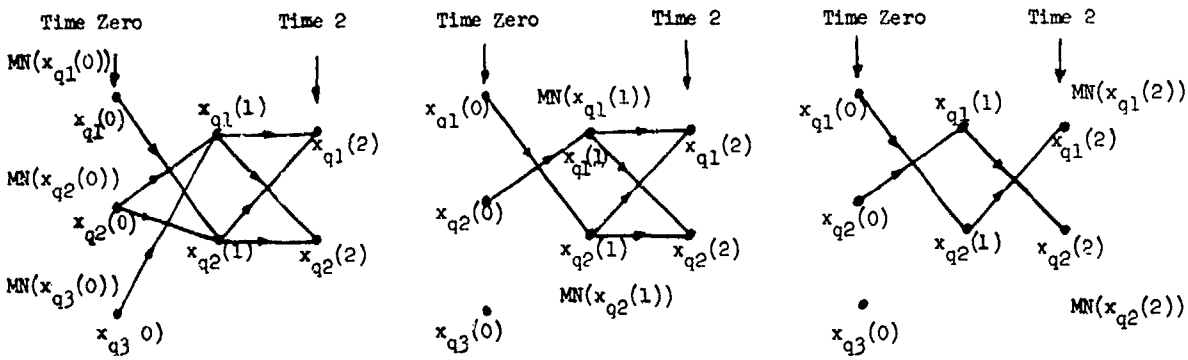
$$MN(x_{q1}(0)) = \text{Prob}[x_q(0) = x_{q1}(0)]\,, \qquad i = 1, 2, 3$$

From now on, the metric of the node $x_{q1}(k)$ is represented by $MN(x_{q1}(k))$.

First Step - Consider the node $x_{q1}(1)$. The branches $x_{q2}(0)\, x_{q1}(1)$ and $x_{q3}(0)\, x_{q1}(1)$ are the only ones connecting the nodes at time zero to $x_{q1}(1)$. Hence calculating the metrics of these branches and then adding these metrics to the metrics of the nodes $x_{q2}(0)$ and $x_{q3}(0)$, the following are obtained

$$A_{11} \triangleq M(x_{q2}(0) \to x_{q1}(1)) + MN(x_{q2}(0))$$

$$A_{12} \triangleq M(x_{q3}(0) \to x_{q1}(1)) + MN(x_{q3}(0))$$



a. Trellis Diagram for the Target Motion from time zero to time 2.

b. Diagram at the end of the first step.

c. Diagram at the end of the second step.

Figure 2.8.1. Diagrams for the Example for the Optimum Decoding based Smoothing Algorithm

Further, assuming that $A_{11} \geq A_{12}$, the path $x_{q2}(0)\ x_{q1}(1)$ is chosen as the best path for the node $x_{q1}(1)$, and A_{11} is assigned to the node $x_{q1}(1)$ as its metric, i.e.,

$$MN(x_{q1}(1)) = A_{11}$$

then the path $x_{q3}(0)\ x_{q1}(1)$ is discarded. Let us now assume that the following are similarly found for the node $x_{q2}(1)$.

$x_{q1}(0)\ x_{q2}(1)$ is the best path for $x_{q2}(1)$

$$MN(x_{q2}(1)) = M(x_{q1}(0) \rightarrow x_{q2}(1)) + MN(x_{q1}(0))$$

Hence, we have Figure 2.8.1.b at the end of the first step.

Second Step - Consider the node $x_{q1}(2)$. The branches $x_{q1}(1)\ x_{q1}(2)$ and $x_{q2}\ x_{q1}(2)$ are the ones connecting the nodes at time one to the node $x_{q1}(2)$. Hence calculating the metrics of these branches and adding these metrics to the metrics of the nodes $x_{q1}(1)$ and $x_{q2}(1)$, the following are obtained

$$A_{21} \triangleq M(x_{q1}(1) \rightarrow x_{q1}(2)) + MN(x_{q1}(1))$$

$$A_{22} \triangleq M(x_{q2}(1) \rightarrow x_{q1}(2)) + MN(x_{q2}(1))$$

Further, assuming that $A_{22} \geq A_{21}$, the path $x_{q1}(0)\ x_{q2}(1)\ x_{q1}(2)$ is chosen as the best path for the node $x_{q1}(2)$, and A_{22} is assigned to the node $x_{q1}(2)$ as its metric, that is,

$$MN(x_{q1}(2)) = A_{22}$$

then the path $x_{q2}(0)\ x_{q1}(1)\ x_{q1}(2)$ is discarded. Let us now assume that the following are similarly found for the node $x_{q2}(2)$.

$x_{q2}(0)\ x_{q1}(1)\ x_{q2}(2)$ is the best path for $x_{q2}(2)$

$$MN(x_{q2}(2)) = M(x_{q1}(1) \rightarrow x_{q2}(2)) + MN(x_{q1}(1))$$

Hence, we have Figure 2.8.1.c at the end of the second step. In addition, assuming that

$$MN(x_{q2}(2)) \geq MN(x_{q1}(2))$$

the path $x_{q2}(0)\ x_{q1}(1)\ x_{q2}(2)$ is chosen as the path followed by the target from time zero to time 2.

Having defined the optimum decoding based smoothing algorithm, its performance should be determined. This is discussed in the next two sections.

2.8.2 An Upper Bound for the Overall Error Probability

Let us consider a target whose motion from time zero to time L is described by a trellis diagram with M possible paths $H_1, H_2, \ldots, H_m$ from time zero to time L. The evaluation of the overall error probability P_E for the detection of the path followed by the target from time 0 to time L is conceptually easy; however, it is in general computationally impractical since it contains multidimensional integrals. On the other hand, upper bounds on P_E are available which in some cases approximate P_E quite well. One of these bounds is presented below.

Let Γ_i be a subset of the observation space D such that

$$\Gamma_i \triangleq \{z^L:\ M(H_j) \geq M(H_i) \text{ for some } j \neq i\} \tag{2.8.2.1}$$

where $M(H_i)$ is the metric of the path (hypothesis) H_i and it is given by (2.7.17). Then Γ_i contains the complement, $\bar{D}_i$, of the optimum decision region D_i (since the observation sequences (z^L's) for which the inequality in (2.6.4) becomes an equality are resolved at random into the decision regions satisfying the equality). It then follows from (2.7.18) the error probability of the path H_i can be upper bounded by

$$P_{E_i}(H_1, H_2, \ldots, H_M) \leq \int_{z^L \in \Gamma_i} p(z^L|H_i)\, dz^L$$

$$= \int_{z^L \in \Gamma_i} p(z^L|H_i)\, \phi(z^L)\, dz^L \tag{2.8.2.2}$$

where the function $\phi(z^L)$ is defined by

$$\phi(z^L) \triangleq \begin{cases} 1 & \text{if } z^L \in \Gamma_i \\ 0 & \text{elsewhere} \end{cases} \tag{2.8.2.3}$$

Furthermore, $\phi(z^L)$ can be upper bounded as follows:

If $z^L \in \Gamma_i$, it then follows from (2.8.2.1) that for some $j \neq i$, $M(H_j) - M(H_i) \geq 0$. Hence

$$\exp \alpha[M(H_j) - M(H_i)] \geq 1$$

for some $j \neq i$ and any non-negative number α. Thus, for any non-negative numbers α and ρ we have

$$\Big(\sum_{j \neq i} \exp \alpha[M(H_j) - M(H_i)]\Big)^{\rho} \geq 1 \qquad \text{for any } \alpha,\ \rho \geq 0 \tag{2.8.2.4}$$

On the other hand, if $z^L \notin \Gamma_i$, then the expression on the left hand side of the inequality in (2.8.2.4) is at least a non-negative number. Therefore, for all z^L, we obtain

$$\phi(z^L) \leq \Big(\sum_{j \neq i} \exp \alpha[M(H_j) - M(H_i)]\Big)^{\rho} \qquad \text{for any } \alpha,\ \rho \geq 0 \tag{2.8.2.5}$$

Further, substituting (2.8.2.5) into (2.8.2.2) yields

$$P_{E_i}(H_1, \ldots, H_M) \leq \int_{z^L \in \Gamma_i} p(z^L|H_i) \Big(\sum_{j \neq i} \exp \alpha[M(H_j) - M(H_i)]\Big)^{\rho} dz^L \tag{2.8.2.6}$$

The integrand in (2.8.2.6) is obviously non-negative. Hence enlarging the domain of the integration in (2.8.2.6) makes the value of the integral larger. Therefore, the error probability of H_i can be further upper bounded by taking the integration over the whole observation space

$$P_{E_i}(H_1, \ldots, H_M) \leq \int_{z^L} p(z^L|H_i) \exp - \alpha\rho\, M(H_i)[\sum_{j\neq i} \exp \alpha\, M(H_j)]^{\rho}\, dz^L \quad \text{for any } \alpha,\ \rho \geq 0 \qquad (2.8.2.7)$$

Since α and ρ are any arbitrary non-negative numbers, α can be set equal to $1/(1+\rho)$. Hence taking $\alpha = 1/(1+\rho)$ and further using (2.7.17), (2.7.11), and the following equality

$$a^b = e^{b \ln a} \qquad \text{for any } a,\ b \in R,\ a > 0$$

the error probability of the path H_i can be bounded by

$$P_{E_i}(H_1,H_2,\ldots,H_M) \leq \int_{z^L} \Big(\prod_{k=0}^{L} \frac{1}{\pi_k^i}\Big)^{\frac{\rho}{1+\rho}} \Big[\prod_{k=1}^{L} p'(z(k)|x_q^i(k))\Big]^{\frac{1}{1+\rho}} \Big\{\sum_{j\neq i} \Big(\prod_{k=0}^{L} \pi_k^j\Big)^{\frac{1}{1+\rho}} \Big[\prod_{k=1}^{L} p'(z(k)|x_q^j(k))\Big]^{\frac{1}{1+\rho}}\Big\}^{\rho} dz^L$$

$$\text{for any } \rho \geq 0 \qquad (2.8.2.8)$$

where $p'(z(k)|x_q^i(k))$ is given by (2.7.10) and π_k^i is defined in Section 2.5. The bound in (2.8.2.8) is that of Gallager's type. Substituting this bound for the error probability of the path H_i in (2.7.19) yields an upper bound on the overall error probability for the detection of the path followed by the target.

2.8.3 An Ensemble Upper Bound for the Overall Error Probability

Let us consider a target whose motion is described by a trellis diagram T with M possible paths $H_1, H_2, \ldots, H_M$ from time zero to time L and let H_i pass through the quantization levels $x_q^i(0), x_q^i(1), \ldots, x_q^i(L)$ (see Figure 2.4.1). In order to derive an ensemble bound, let us start defining the following symbols, which will be used in the following analyses.

X^e is the set of all possible quantization levels from time one to time L, namely

$$X^e \triangleq \{\text{all possible values of } x_q(k) \text{ for } k = 1, 2, \ldots, L\}$$

N^e is the number of elements in X^e

$\underset{\sim}{H}^e$ is the set of all L-tuples of X^e

$\mathcal{E}$ is the ensemble (or set) of all M-tuples of $\underset{\sim}{H}^e$. Hence $\mathcal{E}$ contains $(N^e)^{LM}$ elements in it

H_i^e is the set of all quantization levels which the path H_i passes through from time one to time L, i.e.,

$$H_i^e \triangleq \{x_q^i(1), x_q^i(2), \ldots, x_q^i(L)\} \in \underset{\sim}{H}^e$$

which is an L-tuple of X^e

T^e is the set of all quantization levels from time one to time L in T. In other words,

$$T^e \triangleq (H_1^e, H_2^e, \ldots, H_M^e) \in \mathcal{E}$$

$\mathcal{E}^M$ is the ensemble of all possible trellis diagrams with M possible paths from time zero to time L, which are obtained from the trellis diagram T by replacing only T^e by elements of $\mathcal{E}$. Hence this ensemble contains $(N^e)^{LM}$ elements in it. $\mathcal{E}^M$ is referred to as the ensemble of each motion (or trellis diagram) in itself. Obviously $\mathcal{E}^M$ is the ensemble of T too (since $T \in \mathcal{E}^M$).

The exact expressions for both the error probability and the upper bound, given in the previous section, contain multidimensional integrals which are generally very complex to evaluate. Therefore, instead of evaluating these, we consider an average error probability over the ensemble $\mathcal{E}$. Averaging an error probability over this ensemble is referred to as "Random Coding" which is the central technique of Information Theory. An upper bound averaged over the ensemble $\mathcal{E}$ is called an ensemble bound, which turns out to be quite simple to evaluate. Obviously, at least one trellis diagram in $\mathcal{E}^M$ must have an error probability as small as this ensemble bound (by convention, both error probabilities and error probability bounds are assigned to the related trellis diagrams). In other words an ensemble upper bound will give us an upper bound on the error probability for the best trellis diagram in $\mathcal{E}^M$ (i.e., the trellis diagram with minimum error probability in $\mathcal{E}^M$).

In order to derive an ensemble error probability or an ensemble bound, first a probability

density function $Q^e(\cdot)$ is defined on the ensemble $\mathcal{E}$ such that

$$Q^e(T^e) \triangleq \prod_{i=1}^{M} Q(H_i^e)$$

$$Q(H_i^e) \triangleq \prod_{k=1}^{L} q(x_q^i(k)) \qquad (2.8.3.1)$$

where $q(\cdot)$ is an arbitrary probability density function on X^e. Hence $Q(\cdot)$ is a probability density function on the set $\underline{H}^e$. Then an error probability or an error probability bound is averaged with respect to $Q^e(\cdot)$ over the ensemble.

The ensemble error probability, denoted by either $\bar{P}_E$ or $\overline{P_E(T)}$, for the detection of the path (through a trellis diagram T in $\mathcal{E}_M$) most likely followed by the target (associated with T) is defined by

$$\bar{P}_E = \sum_{T^e \in \mathcal{E}} Q^e(T^e)\, P_E(T)$$

$$= \sum_{T \in \mathcal{E}_M} Q^e(T^e)\, P_E(T) \qquad (2.8.3.2)$$

where $P_E(T)$ is the overall error probability for the detection of the path (through T) most likely followed by the target (associated with T) and T^e is the set of all quantization levels from time one to time L in T. Substituting (2.7.19) into (2.8.3.2) and changing the order of summations, the ensemble error proability can be rewritten in terms of the path ensemble error probabilities as

$$\bar{P}_E = \sum_{i=1}^{M} p(H_i)\, \overline{P_{E_i}(T)} \qquad (2.8.3.3)$$

where

$$\overline{P_{E_i}(T)} = \sum_{T^e \in \mathcal{E}} Q^e(T^e)\, P_{E_i}(T) \qquad (2.8.3.4)$$

where $P_{E_i}(T)$ is the error probability of the path H_i, and $\overline{P_{E_i}(T)}$ is referred to as the ensemble error probability of the path H_i. As being noticed, a bar at the top of a quantity (symbol) denotes the ensemble average of that quantity. Ensemble upper bounds for the detection of the path most probably followed by the target can be obtained by averaging upper bounds for the path error probabilities over the ensemble. Let $P_{EB_i}(T)$ be an upper bound for the error probability of the path H_i, i.e.,

$$P_{E_i}(T) \leq P_{EB_i}(T) \qquad (2.8.3.5)$$

Substituting this bound for the error probability of H_i in (2.8.3.4) yields the following bound for the ensemble error probability of the path H_i

$$\overline{P_{E_i}(T)} \leq \sum_{T^e \in \mathcal{E}} Q^e(T^e)\, P_{EB_i}(T)$$

$$\triangleq \overline{P_{EB_i}(T)} \qquad (2.8.3.6)$$

where $\overline{P_{EB_i}(T)}$ is referred to as an ensemble upper bound for the error probability of the path H_i. Further, substituting the bound in (2.8.3.6) for the ensemble error probability of the path H_i in (2.8.3.3) yields the following bound for the ensemble error probability

$$\bar{P}_E \leq \sum_{i=1}^{M} p(H_i)\, \overline{P_{EB_i}(T)} \qquad (2.8.3.7)$$

Let us now derive an ensemble bound for the overall error probability by using the bound in (2.8.2.8). Substituting the bound in (2.8.2.8) and (2.8.3.1) into (2.8.3.6), we can upper bound the ensemble error probability of the path H_i as

$$\overline{P_{E_i}(T)} \le \sum_{H_1^e \in \underset{\sim}{H}^e} \cdots \sum_{H_M^e \in \underset{\sim}{H}^e} Q(H_1^e)\, Q(H_2^e) \cdots Q(H_M^e) \int_{z^L} b_i^{-\frac{\rho}{1+\rho}} [p'(z^L|H_i)]^{\frac{1}{1+\rho}} \{ \sum_{j \neq i} b_j^{\frac{1}{1+\rho}} [p'(z^L|H_j)]^{\frac{1}{1+\rho}} \}^{\rho} \, dz^L$$

$$\text{for any } \rho \ge 0 \qquad (2.8.3.8)$$

where $p'(z^L|H_i)$ is defined by (2.7.3) and

$$b_i \triangleq \prod_{k=0}^{L} \pi_k^i \qquad (2.8.3.9)$$

Changing the order of summations and integration, (2.8.3.8) can be rewritten as

$$\overline{P_{E_i}(T)} \le \int_{z^L} b_i^{-\frac{\rho}{1+\rho}} \{ \sum_{H_i^e \in \underset{\sim}{H}^e} Q(H_i^e) [p'(z^L|H_i)]^{\frac{1}{1+\rho}} \} \{ \sum_{H_j^e \in \underset{\sim}{H}^e} \cdots \sum_{j \neq i} (\prod_{j \neq i} Q(H_j^e)) [\sum_{j \neq i} b_j^{\frac{1}{1+\rho}} (p'(z^L|H_j)^{\frac{1}{1+\rho}}]^{\rho} \} \qquad (2.8.3.10)$$

where ρ is an arbitrary non-negative number. If we restrict the parameter ρ to lie in [0,1], then the term in the last braces can be further upper bounded by using Jensen's inequality. Let f(R) be the term in the last brackets, namely,

$$f(R) \triangleq R^{\rho} \triangleq [\sum_{j \neq i} b_j^{\frac{1}{1+\rho}} (p'(z^L|H_j))^{\frac{1}{1+\rho}}]^{\rho} \qquad (2.8.3.11)$$

where

$$R \triangleq \sum_{j \neq i} b_j^{\frac{1}{1+\rho}} (p'(z^L|H_j))^{\frac{1}{1+\rho}} \qquad (2.8.3.12)$$

Then, f(R) is a convex $\cap$ function for any $\rho \in [0,1]$ since R^{ρ} is a convex $\cap$ function for any $\rho \in [0,1]$. Furthermore, the term in the last braces in (2.8.3.10) is the expectation of f(R) with respect to the following probability density function

$$\sum_{H_j^e \in \underset{\sim}{H}^e} \cdots \sum_{j \neq i} (\prod_{j \neq i} Q(H_j^e)) \qquad (2.8.3.12)$$

Therefore, using Jensen's inequality[19] (If R is a random variable, f(R) is a convex $\cap$ function of R, and E(R) is finite, then

$$E(f(R)) \le f(E(R)) \qquad (2.8.3.13)$$

where E stands for the expectation) and recognizing that H_j^e is summed over the same space $\underset{\sim}{H}^e$ for all j; and

$$\sum_{H_j^e \in \underset{\sim}{H}^e} Q(H_j^e) = 1 \qquad (2.8.3.14)$$

we obtain the following bound for the term in the last braces in (2.8.3.10).

$$\sum_{H_j^e \in \underset{\sim}{H}^e} \cdots \sum_{j \neq i} (\prod_{j \neq i} Q(H_j^e)) (\sum_{j \neq i} b_j^{\frac{1}{1+\rho}} (p'(z^L|H_j))^{\frac{1}{1+\rho}}]^{\rho} \le [\sum_{j \neq i} b_j^{\frac{1}{1+\rho}}]^{\rho} [\sum_{H_i^e \in \underset{\sim}{H}^e} Q(H_i^e) (p'(z^L|H_i))^{\frac{1}{1+\rho}}]^{\rho} \qquad (2.8.3.15)$$

$$\text{for any } \rho \in [0,1]$$

Substituting this bound into (2.8.3.10), we get

$$\overline{P_{E_i}(T)} \le b_i^{-\frac{1}{1+\rho}} \Big[\sum_{j\neq i} b_j^{\frac{1}{1+\rho}}\Big]^{\rho} \int_{z^L} \Big[\sum_{H_i^e \in \underline{H}^e} Q(H_i^e)\ (p'(z^L|H_i))^{\frac{1}{1+\rho}}\Big]^{\rho+1}\, dz^L \qquad (2.8.3.16)$$

Further, using the following inequalities

$$\prod_{k=0}^{L} \pi_k^{\min} \le b_i \le \prod_{k=0}^{L} \pi_k^{\max} \quad \text{for all } i \qquad (2.8.3.17)$$

the term outside the integral in (2.8.3.16) can be further upper bounded by

$$b_i^{-\frac{\rho}{1+\rho}} \Big[\sum_{j\neq i} b_j^{\frac{1}{1+\rho}}\Big]^{\rho} \le (M-1)^{\rho} \prod_{k=0}^{L} \Big(\frac{\pi_k^{\max}}{\pi_k^{\min}}\Big)^{\frac{\rho}{1+\rho}} \quad \text{for all } i \qquad (2.8.3.18)$$

Let us now consider the term in the last brackets in (2.8.3.16). Substituting (2.8.3.1) and (2.7.3) into this term, and changing the order of summations and multiplications yield

$$\begin{aligned}
\sum_{H_i^e \in \underline{H}^e} Q(H_i^e)\ (p'(z^L|H_i))^{\frac{1}{1+\rho}} &= \sum_{H_i^e \in H^e} \prod_{k=1}^{L} q(x_q^i(k))\ (p'(z(k)|x_q^i(k)))^{\frac{1}{1+\rho}} \\
&= \prod_{k=1}^{L} \Big\{ \sum_{x_q^i(k)\in X^e} q(x_q^i(k))(p'(z(k)|x_q^i(k)))^{\frac{1}{1+\rho}} \Big\} \\
&= \prod_{k=1}^{L} \Big\{ \sum_{x\in X^e} q(x)\ (p'(z(k)|x))^{\frac{1}{1+\rho}} \Big\} \qquad (2.8.3.19)
\end{aligned}$$

the last equality follows from the fact that for all i and k, $x_q^i(k)$ is summed over the same space X^e. Substituting (2.8.3.18) and (2.8.3.19) into (2.8.3.16) yields the following ensemble upper bound for the error probability of H_i.

$$\overline{P_{E_i}(T)} \le (M-1)^{\rho} \Big(\prod_{k=0}^{L} \frac{\pi_k^{\max}}{\pi_k^{\min}}\Big)^{\frac{\rho}{1+\rho}} \prod_{k=1}^{L} \Big\{\int_{z(k)} \Big\{\sum_{x\in X^e} q(x)\ (p'(z(k)|x))^{\frac{1}{1+\rho}}\Big\}^{\rho+1}\Big\}$$

$$\triangleq \overline{B(T)} \quad \text{for all } i \text{ and any } \rho \in [0,1] \qquad (2.8.3.20)$$

where $p'(z(k)|x)$ is given by (2.7.10), $\pi_k^{\max}$ and $\pi_k^{\min}$ are defined in Section 7.5, and $q(\cdot)$ is an arbitrary density function on X^e. Substituting the bound in (2.8.3.20) into (2.8.3.7), and recognizing that this bound does not depend on i (that is, the paths) yield

$$\begin{aligned}
\bar{P}_E &\le \sum_{i=1}^{M} p(H_i)\ \overline{B(T)} \\
&= \overline{B(T)} \sum_{i=1}^{M} p(H_i) = \overline{B(T)} \qquad (2.8.3.21)
\end{aligned}$$

Hence, the bound in (2.8.3.20) is also an upper bound for the ensemble error probability for the detection of the path most probably followed by the target.

If the function g(k, ., ., .) in the observation model being considered, and the statistics of the observation noise v(k) and the interference I(k) (in the presence of interference) are time-invariant, then the term in braces in (2.8.3.20) is time-invariant. Hence, in this case, the ensemble upper bound in (2.8.3.20) becomes

$$\bar{P}_E \le (M-1)^{\rho} \Big(\prod_{k=0}^{L} \frac{\pi_k^{\max}}{\pi_k^{\min}}\Big)^{\frac{\rho}{1+\rho}} \Big\{\int_{z(k)} \Big\{\sum_{x\in X^e} q(x)\ (p'(z(k)|x))^{\frac{1}{1+\rho}}\Big\}^{\rho+1}\Big\}^{L} \quad \text{for any } \rho \in [0,1] \qquad (2.8.3.22)$$

Using the relation that $\exp[\ln a] = a$ for any $a > 0$, the bound in (2.8.3.22) can be rewritten as

$$\bar{P}_E \le \exp\text{-}L\Big(E_0(\rho,q) - \rho\,\frac{\ln(M-1)}{L} - \frac{\rho}{1+\rho}\,\frac{\ln C}{L}\Big) \quad \text{for any } \rho \in [0,1] \qquad (2.8.3.23)$$

where

$$E_0(\rho,\underline{q}) \triangleq -\ln\Big\{\int_{z(k)} \Big[\sum_{x\in X^e} q(x)\ (p'(z(k)|x))^{\frac{1}{1+\rho}}\Big]^{\rho+1}\Big\} \tag{2.8.3.24}$$

$$G \triangleq \prod_{k=0}^{L} \frac{\pi_k^{max}}{\pi_k^{min}}$$

$E_0(\rho,\underline{q})$ is referred to as the Gallager function since it was first defined by Gallager[17]. Recalling that ρ is any arbitrary number in $[0,1]$ and $\underline{q}(\cdot)$ is an arbitrary probability density function on X^e leads us to obtaining the tightest bound on $\bar{P}_E$ by minimizing the right hand side in (2.8.3.23) over ρ and $\underline{q}$. This gives us the following bound

$$\bar{P}_E \leq \exp -LE(M,G,L) \tag{2.8.3.24}$$

where

$$E(M,G,L) \triangleq \max_{\underline{q}} \max_{\rho\in[0,1]} \Big[E_0(\rho,\underline{q}) - \rho\,\frac{\ln(M-1)}{L} - \frac{\rho}{1+\rho}\,\frac{\ln G}{L}\Big]$$

$$= \max_{\rho\in[0,1]} \Big[\max_{\underline{q}} E_0(\rho,\underline{q}) - \rho\,\frac{\ln(M-1)}{L} - \frac{\rho}{1+\rho}\,\frac{\ln G}{L}\Big] \tag{2.8.3.25}$$

As it has been noticed, the maximization is taken over all $\rho \in [0,1]$ and the set of all possible probability density functions on X^e. In order to evaluate $E(M,G,L)$, it is necessary to analyze $E_0(\rho,\underline{q})$ as a function of ρ. The important properties of this function are stated in the following theorem. The proof of this theorem is presented in Reference [17].

Theorem 2.8.3.1

Assume that the average mutual information, denoted by $I(\underline{q})$, which is defined by

$$I(\underline{q}) \triangleq \int_{z(k)} \sum_{x\in X^e} q(x)\ p'(z(k)|x)\ \ln\Big\{\frac{p'(z(k)|x)}{\sum_{x\in X^e} q(x)\ p'(z(k)|x)}\Big\}\ dz(k) \tag{2.8.3.26}$$

is nonzero (in fact, $I(\underline{q})$ is always non-negative). Then, $E_0(\rho,\underline{q})$ has the following properties

$$E_0(\rho,\underline{q}) = 0 \qquad \text{for } \rho = 0 \tag{2.8.3.27}$$

$$E_0(\rho,\underline{q}) > 0 \qquad \text{for } \rho > 0 \tag{2.8.3.28}$$

$$\frac{\partial E_0(\rho,\underline{q})}{\partial\rho} > 0 \qquad \text{for } \rho > 0 \tag{2.8.3.29}$$

$$\frac{\partial E_0(\rho,\underline{q})}{\partial\rho}\Big|_{\rho=0} = I(\underline{q}) \tag{2.8.3.30}$$

$$\frac{\partial^2 E_0(\rho,\underline{q})}{\partial\rho^2} \leq 0 \tag{2.8.3.31}$$

with equality in (2.8.3.31) if and only if

$$\ln\Big\{\frac{p'(z(k)|x)}{\sum_{x\in X^e} q(x)\ p'(z(k)|x)}\Big\} = I(\underline{q}) \tag{2.8.3.32}$$

for all $x \in X^e$ and all $z(k)$ in the space of all possible observations at time k such that $q(x)\ p'(z(k)|x) > 0$. Therefore, for a given $\underline{q}$, $E_0(\rho,\underline{q})$ is a positive increasing convex $\cap$ function of $\rho \in [0,\infty)$ with a slope at the origin equal to $I(\underline{q})$. Also, the following function

$$\rho\,\frac{\ln(M-1)}{L} + \frac{\rho}{1+\rho}\,\frac{\ln G}{L}$$

is a convex $\cap$ function of $\rho \in [0,\infty)$. Hence we can easily perform the maximization in (2.8.3.25) over $\rho \in [0,1]$ for a given $q(\cdot)$ so that $E(M,G,L)$ can be expressed parametrically as[16,17]

$$E(M,G,L) = \begin{cases} 0 & \text{If } R > C \\ \max_{\underline{q}} E_0(\rho,\underline{q}) - \rho\,\dfrac{\ln(M-1)}{L} - \dfrac{\rho \ln G}{(1+\rho)L} & \\ \dfrac{\partial}{\partial\rho}\left[\max_{\underline{q}} E_0(\rho,\underline{q})\right] = \dfrac{\ln(M-1)}{L} + \dfrac{\ln G}{(1+\rho)^2 L} & \text{If } \dfrac{\partial}{\partial\rho}\left[\max_{\underline{q}} E_0(\rho,\underline{q})\right]\Big|_{\rho=1} \le R \le C \\ \max_{\underline{q}} E_0(1,\underline{q}) - \dfrac{\ln(M-1)}{L} - \dfrac{\ln G}{2L} & \text{If } R \le \dfrac{\partial}{\partial\rho}\left[\max_{\underline{q}} E_0(\rho,\underline{q})\right]\Big|_{\rho=1} \end{cases} \qquad (2.8.3.33)$$

where

$$R \triangleq \frac{\ln(M-1)}{L} + \frac{\ln G}{4L}$$

$$C \triangleq \max_{\underline{q}} I(\underline{q}) \qquad (2.8.3.34)$$

The maximization of the Gallager function $E_0(\rho,q)$ and the average mutual information $I(q)$ over the space of all possible probability density functions on X^e has been treated in the literature. Two theorems related to this maximization are stated. Their proof can be found in References [16] and [17].

Theorem 2.8.3.2

A probability density function $q_0(\cdot)$ on X^e maximizes the function $E_0(\rho,\underline{q}_0)$ for a given $\rho \ge 0$ if an only if the following holds

$$\int_{z(k)} [p'(z(k)|x)]^{\frac{1}{1+\rho}} [a(z(k),\underline{q}_0)]^{\rho} \le \int_{z(k)} [a(z(k),\underline{q}_0)]^{1+\rho} \quad \text{for all } x \in X^e \qquad (2.8.3.35)$$

with equality for all $x \in X^e$ for which $q_0(x) > 0$ where

$$a(z(k),\underline{q}_0) \triangleq \sum_{x \in X^e} q_0(x)\,[p'(z(k)|x)]^{\frac{1}{1+\rho}} \qquad (2.8.3.36)$$

Theorem 2.8.3.3

A probability density function $q_0(\cdot)$ on X^e maximizes the average mutual information $I(\underline{q}_0)$ if and only if the following holds

$$\int_{z(k)} p'(z(k)|x) \ln\left[\frac{p'(z(k)|x)}{\sum_{x \in X^e} q_0(x)\,p'(z(k)|x)}\right] \le I(q_0) \quad \text{for all } x \in X^e \qquad (2.8.3.37)$$

equality for all x for which $q_0(x) > 0$.

It then follows that neither of these theorems is very useful in finding the maximum of the Gallager function or the average mutual information. But both are useful in verifying that a hypothesized solution is indeed a solution. For example, using these theorems, it can be verified that the uniform distribution on X^e, that is

$$q(x) = \frac{1}{N^e} \quad \text{for all } x \in X^e \qquad (2.8.3.38)$$

is not the optimum (maximizing) distribution for either $E_0(\rho,q)$ or $I(q)$. In general the maximization over the probability density functions on X^e must be performed numerically. Even if the optimum (maximizing) probability density function is known, the evaluation of the Gallager function or the average mutual information is, in general, not easy at all since the related expressions contain multidimensional integrals, hence the evaluation of them must be performed numerically. Throughout this chapter, as the ensemble upper bound, the bound using $\rho = 1$ and the uniform distribution for $q(\cdot)$ in (2.8.3.20) is used by virtue of the nice feature arising from the uniform distribution (which is stated in the next theorem) and the fact that the simplest function to calculate among $E_0(\rho,\underline{q})$ is

$E_0(1,q)$. Obviously this bound is, in general, not as tight as the bound in (2.8.3.24). Substituting $\rho = 1$ and $(1/N^e)$ for $q(x)$ in (2.8.3.20), we obtain the following bound for the ensemble upper bound for the overall error probability

$$\bar{P}_E \leq D \prod_{k=1}^{L} \{ \int_{z(k)} [\sum_{x \in X^e} (p'(z(k)|x))^{1/2}]^2 \}$$

$$\triangleq B^e \qquad (2.8.3.39)$$

where

$$D \triangleq (M-1) [\prod_{k=1}^{L} (\frac{\pi_k^{max}}{\pi_k^{min}})^{1/2}] (\frac{1}{N^e})^{2L} \qquad (2.8.3.40)$$

If the function $g(k, ., ., .)$ in the observation model being considered and the statistics of the observation noise $v(k)$ and the interference $I(k)$ (in the presence of interference) are time-invariant, then the bound in (2.8.3.39) becomes

$$\bar{P}_E \leq D \{ \int_{z(k)} [\sum_{x \in X^e} (p'(z(k)|x))^{1/2}]^2 \}^L \qquad (2.8.3.41)$$

Let us now prove the following theorem which gives us the reason that a uniformly weighted ensemble bound (like (2.8.3.39)) is used as the performance measure. A uniformly weighted ensemble bound is an ensemble bound obtained by using the uniform density function $Q^e(\cdot)$ on the ensemble $\mathcal{E}$, i.e.,

$$q(x) = \frac{1}{N^e} \qquad \text{for all } x \in X^e \qquad (2.8.3.42)$$

Hence

$$Q(T^e) = \frac{1}{(N^e)^{LM}} \qquad \text{for all } T^e \in \mathcal{E}$$

Theorem 2.8.3.4

For a given uniformly weighted ensemble upper bound B^e for the overall error probability, there exists a subset, denoted by $\mathcal{E}N_g$, of the ensemble $\mathcal{E}N$ such that $\mathcal{E}N_g$ contains at least half of the elements in $\mathcal{E}N$, and every element (trellis diagram) in $\mathcal{E}N_g$ must have an overall error probability which is less than or equal to two times B^e, i.e.,

$$P_E(T) \leq 2B^e \qquad \text{for all } T \in \mathcal{E}N_g \qquad (2.8.3.43)$$

where $P_E(T)$ is the overall error probability of T (i.e., the overall error probability for the detection of the path through T most likely followed by the target associated with the trellis diagram T).

Proof: Let us assume that $\overline{\mathcal{E}N_g}$ is the complement of $\mathcal{E}N_g$, i.e.,

$$P_E(T) > 2B^e \qquad \text{for all } T \in \overline{\mathcal{E}N_g} \qquad (2.8.3.44)$$

and the number, denoted by K, of elements in $\overline{\mathcal{E}N_g}$ is greater than half of the number of elements in $\mathcal{E}N$ (otherwise there is nothing to prove), i.e.,

$$K > \frac{(N^e)^{LM}}{2} \qquad (2.8.3.45)$$

Since both $Q^e(T^e)$ and $P_E(T)$ are positive for all T in $\mathcal{E}N$, and $\overline{\mathcal{E}N_g}$ is contained in $\mathcal{E}N$, the ensemble error probability $\bar{P}_E$ defined by (2.8.3.2) can be lower bounded by

$$\bar{P}_E \geq \sum_{T \in \overline{\mathcal{E}N_g}} Q^e(T^e) P_E(T) \qquad (2.8.3.46)$$

Substituting the bound in (2.8.3.44) for the overall error probability of T, the ensemble error probability can be further lower bounded as

$$\bar{P}_E > 2B^e \sum_{T \in \mathcal{C}M_s} Q^e(T^e) \qquad (2.8.3.47)$$

Substituting (2.8.3.42) into (2.8.3.47) (since the uniformly weighted ensemble bound is considered) we get

$$\bar{P}_E > 2B^e \sum_{T \in \mathcal{C}M_s} \frac{1}{(N^e)^{LM}}$$

$$= 2B^e K \frac{1}{(N^e)^{LM}} \qquad (2.8.3.48)$$

Further, substituting the lower bound in (2.8.3.45) for K in (2.8.3.48), the bound in (2.8.3.48) can again be lower bounded so that we have

$$\bar{P}_E > B^e$$

This contradicts the assumption that B^e is an ensemble upper bound. This completes the proof.

If the trellis diagram associated with the target being considered is a member of $\mathcal{C}M_s$, then the overall error probability of this trellis is upper bounded by $2B^e$; otherwise, we do not have any idea about this overall error probability. Since $\mathcal{C}M_s$ contains at least half of the elements in the ensemble $\mathcal{C}M$, there is a good chance that the trellis diagram being considered belongs to $\mathcal{C}M_s$. However, for nonuniformly weighted ensembles, a large subset (of $\mathcal{C}M$), every element of which has an overall error probability bounded by a constant bound, may not easily be obtained since the arguments must include the effect of nonuniform weighting. That is why uniformly weighted ensemble bounds are used as the performance measure.

2.9 STACK SEQUENTIAL DECODING BASED SMOOTHING ALGORITHM

In tracking a target from time zero to time L by using the optimum decoding based smoothing algorithm (ODSA), the path most likely followed by the target [multiple (composite) hypothesis testing or decoding problem] is decided by simply finding the path with the largest metric through a trellis diagram from time zero to time L. ODSA does this by using the Viterbi decoding algorithm, which systematically examines (searches) all possible paths in the trellis diagram. Hence, if the number of possible paths in the trellis diagram is very large, ODSA requires a huge amount of memory and computation.

If there were a way to guess the correct path without calculating the metric of every path in the trellis diagram, most of the computation and memory requirement in ODSA could be avoided. One way to do so is to use a smoothing algorithm using a stack sequential decoding algorithm (which is suboptimum, i.e., it does not minimize the overall error probability)[15,16]. Such a smoothing algorithm, which at any time (step) stores a "stack" of already searched paths of varying length ordered according to their metrics, is presented below. This algorithm is referred to as Stack Sequential Decoding Based Smoothing Algorithm (SSDSA).

Initial Step - Reducing the target motion model to a finite state model, as described before, obtain a trellis diagram for the target motion model from time zero to time, say L, until which the target will be tracked. Calculate the metrics of all initial paths (by convention, an initial node and its metric are referred to as an initial path and the metric of this initial path respectively). Then store these paths and their metrics and order them according to their metrics.

Recursive Step - Compute the metrics of the paths which are the single-branch continuations of the best path in the stack (see Definition 2.9.1 below) and replace the best path and its metric by these paths and their metrics. If any of the newly added paths merges with a path already in the stack, discard the one with smaller metric. Then reorder the remaining paths. If the best path in the stack terminates in a final node of the trellis diagram, stop and choose the best path as the path most probably followed by the target; otherwise repeat the process (i.e., continue to search by extending the best path in the stack).

Definition 2.9.1 The best path in a stack of already searched paths of varying length is the one with the largest metric. If there is more than one path with the same largest metric, then the best path is the one with the longest length. If there is more than one path with the same largest metric and length, then the best path is only one of these paths (this is chosen at random).

The following example illustrates the stack sequential decoding based smoothing algorithm.

2.9.1 An example

Let us consider a target whose motion from time zero to time 2 is described by Figure 2.9.1. Using SSDSA, we would like to find the path (through the trellis diagram) most probably followed by the target from time zero to time 2. Let us first adopt the following conventions.

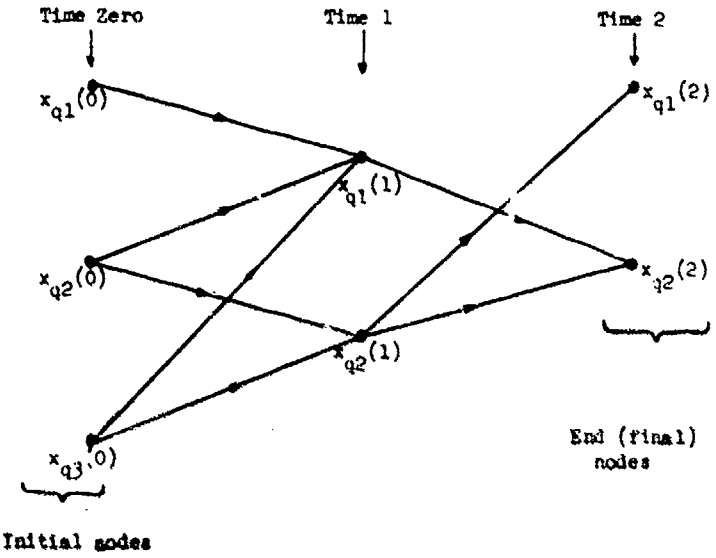


Figure 2.9.1. The Trellis Diagram for the Example for the Stack Sequential Decoding based Smoothing Algorithm

I. In a stack: a searched path is represented by the node sequences that the path passes through followed by its metric. The sequence and the metric are separated by a comma. The metric of the path is denoted by MS(·) where the term in parentheses is the final node of the path. Further, different searched paths are separated by semicolons.

II. A stack is ordered in such a way that the best path is placed at the end of the stack and the path with the second largest metric is placed before the best path, etc.

Then the path most likely followed by the target is obtained as follows.

Initial Step - Assuming that the metrics of the initial paths (i.e., initial nodes) $x_{q1}(0)$, $x_{q2}(0)$, and $x_{q3}(0)$ are such that

$$MS(x_{q2}(0)) \geq MS(x_{q3}(0)) \geq MS(x_{q1}(0)) \tag{2.9.1.1}$$

where $MS(\cdot) \triangleq MM(\cdot)$ is defined by (2.7.14), the following stack is obtained

$$x_{q1}(0),\ MS(x_{q1}(0));\ x_{q3}(0),\ MS(x_{q3}(0));\ x_{q2}(0),\ MS(x_{q2}(0)) \tag{2.9.1.2}$$

First Step - The paths $x_{q2}(0)\,x_{q1}(1)$ and $x_{q2}(0)\,x_{q2}(1)$ are the single-branch continuations of the best path $x_{q2}(0)$ in (2.9.1.2). Hence calculating the metric of these paths, that is,

$$MS(x_{q1}(1)) = M(x_{q2}(0) \to x_{q1}(1)) + MS(x_{q2}(0))$$

$$MS(x_{q2}(1)) = M(x_{q2}(0) \to x_{q2}(1)) + MS(x_{q2}(0)) \tag{2.9.1.3}$$

where $M(x \to y)$ is defined by (2.7.15), and replacing the best path and its metric [i.e., $x_{q2}(0)$, $MS(x_{q2}(0))$] by these paths and their metrics, we get

$$x_{q1}(0),MS(x_{q1}(0));x_{q3}(0),MS(x_{q3}(0));x_{q2}(0)x_{q1}(1),MS(x_{q1}(1));x_{q2}(0)x_{q2}(1),MS(x_{q2}(1)) \tag{2.9.1.4}$$

Now assuming that

$$MS(x_{q3}(0)) \geq MS(x_{q2}(1)) \geq MS(x_{q1}(1)) \geq MS(x_{q1}(0)), \tag{2.9.1.5}$$

and then reordering the paths in (2.9.1.4) according to their metrics we obtain the stack

$$x_{q1}(0),MS(x_{q1}(0));x_{q2}(0)x_{q1}(1),MS(x_{q1}(1));x_{q2}(0)x_{q2}(1),MS(x_{q2}(1));x_{q3}(0),MS(x_{q3}(0)) \qquad (2.9.1.6)$$

Hence, the best path $x_{q3}(0)$ does not terminate in a final node in Figure 2.9.1. Therefore, we shall continue to search by extending the best path $x_{q3}(0)$ similarly.

Second Step - The paths $x_{q3}(0)\ x_{q1}(1)$ and $x_{q3}(0)\ x_{q2}(1)$ are the single-branch continuations of the best path $x_{q3}(0)$ in (2.9.1.6). Thus calculating the metrics of these paths and then replacing the best path and its metric in (2.9.1.6) by these paths and their metrics yields

$$x_{q1}(0),MS(x_{q1}(0));x_{q2}(0)x_{q1}(1),MS(x_{q1}(1));x_{q2}(0)x_{q2}(1),MS(x_{q2}(1));x_{q3}(0)x_{q1}(1),MS(x_{q1}(1));x_{q3}(0)x_{q2}(1),$$
$$MS(x_{q2}(1)). \qquad (2.9.1.7)$$

Hence the newly added paths $x_{q3}(0)\ x_{q1}(1)$ and $x_{q3}(0)\ x_{q2}(1)$ merge with the paths $x_{q2}(0)\ x_{q1}(1)$ and $x_{q2}(0)\ x_{q2}(1)$ (which are already in the stack) respectively. Assuming that

$$MS(x_{q1}(1)) \text{ of } x_{q2}(0)\ x_{q1}(1) \geq MS(x_{q1}(1)) \text{ of } x_{q3}(0)\ x_{q1}(1)$$

and

$$MS(x_{q2}(1)) \text{ of } x_{q2}(0)\ x_{q2}(1) \geq MS(x_{q2}(1)) \text{ of } x_{q3}(0)\ x_{q2}(1)\ ,$$

the paths $x_{q3}(0)\ x_{q1}(1)$ and $x_{q3}(0)\ x_{q2}(1)$ and their metrics are discarded. Hence we have the stack

$$x_{q1}(0),\ MS(x_{q1}(0));\ x_{q2}(0)\ x_{q1}(1),\ MS(x_{q1}(1));\ x_{q2}(0)\ x_{q2}(1),\ MS(x_{q2}(1)) \qquad (2.9.1.8)$$

Still, the best path $x_{q2}(0)\ x_{q2}(1)$ in (2.9.1.8) does not terminate in one of the end nodes of the trellis diagram. Hence the best path $x_{q2}(0)\ x_{q2}(1)$ is extended.

Third Step - Computing the metrics of the paths $x_{q2}(0)\ x_{q2}(1)\ x_{q1}(2)$ and $x_{q2}(0)\ x_{q2}(1)\ x_{q2}(2)$, namely,

$$MS(x_{q1}(2)) = M(x_{q2}(1) - x_{q1}(2)) + MS(x_{q2}(1))$$

$$MS(x_{q2}(2)) = M(x_{q2}(1) - x_{q2}(2)) + MS(x_{q2}(1))\ , \qquad (2.9.1.9)$$

replacing the best path and its metric in (2.9.1.8) by these paths and their metrics, and then reordering all the paths according to their metrics, let us assume that the following stack is obtained

$$x_{q1}(0),MS(x_{q1}(0));x_{q2}(0)x_{q1}(1),MS(x_{q1}(1));x_{q2}(0)x_{q2}(1)x_{q2}(2),MS(x_{q2}(2));x_{q2}(0)x_{q2}(1)x_{q1}(2),MS(x_{q1}(2)).$$
$$(2.9.1.10)$$

Since the best path $x_{q2}(0)\ x_{q2}(1)\ x_{q1}(2)$ terminates in the final node $x_{q1}(2)$ in the trellis diagram, it is decided that the path $x_{q2}(0)\ x_{q2}(1)\ x_{q1}(2)$ was most probably followed by the target.

Having established the stack sequential decoding based smoothing algorithm, its performance is going to be discussed in the following two sections.

2.9.2 An Upper Bound for the Overall Error Probability

Let us consider a target whose motion from time zero to time L is described by a trellis diagram with N possible paths $H_1, H_2, \ldots,$ and H_N from time zero to time L. Let H_m and $H_{m'}$ be two paths through the trellis diagram such that H_m is the correct path (i.e., it is the one actually followed by the target) and $H_{m'}$ is the incorrect one (see Figure 2.9.2). It is easily verified that since at each step the stack sequential decoding based smoothing algorithm (SSD3A) extends only the best path in the stack by only one branch, the path $H_{m'}$ cannot be chosen as the one most probably followed by the target (as the decision)[16]

$$\text{If} \quad M(x_q^m(i)) > M(x_q^{m'}(j)) \qquad \text{for all} \quad i \in S \qquad \text{and some} \quad j \in S \qquad (2.9.2.1)$$

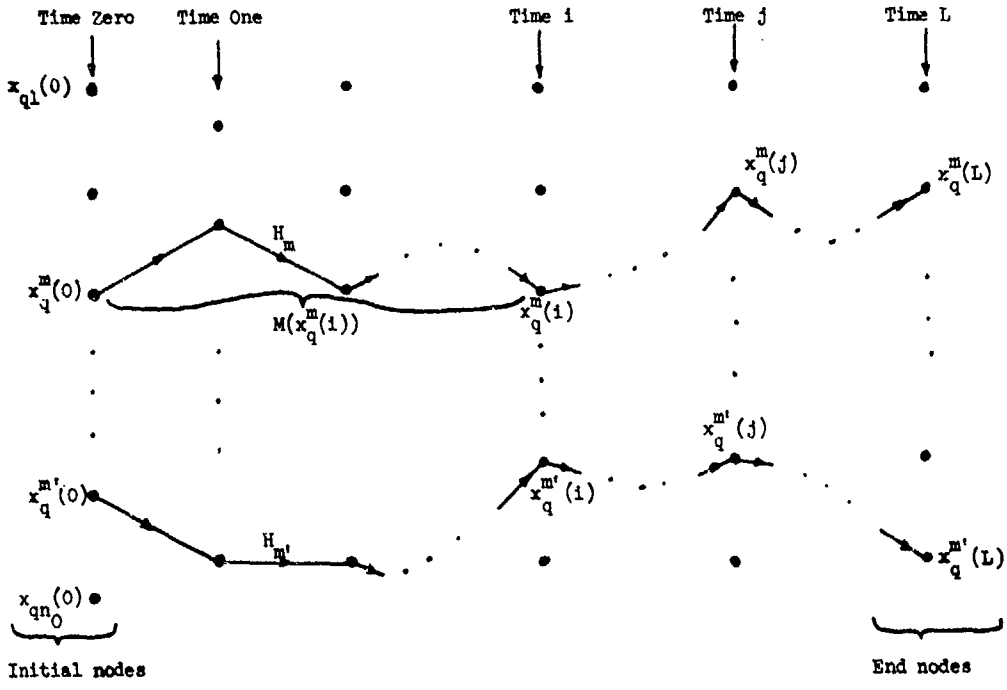


Figure 2.9.2. The Trellis Diagram for the Performance Analysis of the Stack Sequential Decoding based Smoothing Algorithm

or

$$\text{If} \quad \gamma_m > M(x_q^{m'}(j)) \qquad \text{for some} \qquad j \in S;$$

however, the path H_m, may be chosen as the decision

$$\text{If} \quad M(x_q^{m'}(j)) \geq \gamma_m \qquad \text{for all} \qquad j \in S \tag{2.9.2.2}$$

where $M(x_q^m(i))$ is the metric of the portion between the nodes $x_q^m(0)$ and $x_q^m(i)$ of the path H_m, which is defined by (2.7.16), and

$$S \triangleq \{0, 1, 2, \ldots, L\}$$

$$\gamma_m \triangleq \min_{i \in S} M(x_q^m(i)) \tag{2.9.2.3}$$

Let Γ_m be a subset of the observation space D such that

$$\Gamma_m \triangleq \{z^L: \; M(x_q^{m'}(j)) \geq \gamma_m \text{ for all } j \in S \text{ and all } m' \neq m\} \tag{2.9.2.4}$$

Since any incorrect path $H_{m'}$ (the one which was not followed by the target) can be chosen as the decision only if (2.9.2.2) is valid, the set Γ_m contains the complement, $\bar{D}_m$, of the decision region D_m for the path H_m (the decision region D_m is by definition the subset of the observation space such that whenever the observation sequence z^L falls within this subset, SSDSA decides the path H_m as the one most likely followed by the target). Let $\tilde{\Gamma}_m$ be another subset of the observation space D such that

$$\tilde{\Gamma}_m = \{z^L: \; M(x_q^m(L)) \geq \gamma_m \text{ for all } m' \neq m\} \tag{2.9.2.5}$$

Since the inequality in (2.9.2.4) implies the inequality in (2.9.2.5) (the converse is not true), Γ_m is contained in $\tilde{\Gamma}_m$. Hence $\bar{D}_m$ is a subset of $\tilde{\Gamma}_m$. Therefore, the error probability of the path H_m (see (2.7.18)) can be upper bounded by

$$P_{E_m}(H_1, H_2, \ldots, H_M) \leq \int_{z^L \in \tilde{\Gamma}_m} p(z^L|H_m)\, dz^L$$

$$\triangleq \int_{z^L} p(z^L|H_m)\, \phi(z^L)\, dz^L \tag{2.9.2.6}$$

where

$$\phi(z^L) \triangleq \begin{cases} 1 & \text{if} \quad z^L \in \tilde{\Gamma}_m \\ 0 & \text{elsewhere} \end{cases} \tag{2.9.2.7}$$

Moreover, the function $\phi(z^L)$ can be upper bounded by

$$\phi(z^L) \leq [\sum_{m' \neq m} \exp \alpha(M(x_q^{m'}(L)) - \gamma_m)]^\rho \qquad \text{for any } \alpha,\ \rho \geq 0 \tag{2.9.2.8}$$

The reason that (2.9.2.8) is valid is as follows: If $z^L \in \tilde{\Gamma}_m$, it follows from (2.9.2.5) that there exists at least an $m' \neq m$ such that

$$M(x_q^{m'}(L)) - \gamma_m \geq 0$$

Consequently

$$\exp \alpha(M(x_q^{m'}(L)) - \gamma_m) \geq 1 \qquad \text{for any } \alpha \geq 0$$

Therefore the summation in brackets in (2.9.2.8) is greater than or equal to one; obviously any non-negative power, say ρ, of it is greater than or equal to one. On the other hand if $z^L \notin \tilde{\Gamma}_m$, the term in brackets in (2.9.2.8) is at least non-negative. Hence (2.9.2.8) is valid. Further, substituting (2.9.2.8) into (2.9.2.6) yields

$$P_{E_m}(H_1,H_2,\ldots,H_M) \leq \int_{z^L} p(z^L|H_m)\ (\exp - \alpha\rho\gamma_m)\ [\sum_{m' \neq m} \exp \alpha\, M(x_q^{m'}(L))]^\rho\, dz^L \qquad \text{for any } \alpha,\rho \geq 0 \tag{2.9.2.9}$$

Also from (2.9.2.3), we have

$$\exp - \alpha\rho\gamma_m \leq \sum_{i=0}^{L} \exp - \alpha\rho\, M(x_q^m(i)) \tag{2.9.2.10}$$

since at least one term in the summation is equal to the left hand side of the inequality and the other terms in the summation are at least non-negative. Substituting (2.9.2.10) into (2.9.2.9), we get

$$P_{E_m}(H_1,H_2,\ldots,H_M) \leq \int_{z^L} p(z^L|H_m)[\sum_{i=0}^{L} \exp - \alpha\rho\, M(x_q^m(i))][\sum_{m' \neq m} \exp \alpha\, M(x_q^{m'}(L))]^\rho dz^L$$

$$\text{for any } \alpha,\rho \geq 0 \tag{2.9.2.11}$$

Moreover, using (2.7.11), (2.7.16) and the equality

$$\exp(\ln a) = a \qquad \text{for all } a \geq 0\ ,$$

(2.9.2.11) can be rewritten as

$$P_{E_m}(H_1,\ldots,H_M) \leq \int_{z^L} [(\pi_0^m)^{-\alpha\rho} \cdot \prod_{k=1}^{L} p'(z(k)|x_q^m(k)) + \sum_{i=1}^{L} (\pi_0^m)^{-\alpha\rho} \cdot \prod_{k=1}^{i} (\pi_k^m)^{-\alpha\rho}$$

$$\cdot (p'(z(k)|x_q^m(k)))^{1-\alpha\rho} \cdot \prod_{j=i+1}^{L} p'(z(j)|x_q^m(j))] \cdot [\sum_{m' \neq m} [\pi_0^{m'} \prod_{k=1}^{L} \pi_k^{m'}\, p'(z(k)|x_q^{m'}(k))]^\alpha]^\rho\, dz^L$$

$$\text{for any } \alpha,\rho \geq 0 \tag{2.9.2.12}$$

Moreover, using the following inequalities

$$\pi_k^{\min} \leq \pi_k^m \leq \pi_k^{\max} \qquad \text{for all } m \text{ and } k \tag{2.9.2.13}$$

the bound in (2.9.2.12) can be upper bounded further so that we obtain the following bound for the error probability of the path H_m.

$$P_{E_m}(H_1,H_2,\ldots,H_M) \le \int_{z^L} [(\pi_0^{\min})^{-\alpha\rho} \cdot \prod_{k=1}^{L} p'(z(k)|x_q^m(k)) + \sum_{i=1}^{L} (\pi_0^{\min})^{-\alpha\rho} \prod_{k=1}^{i} (\pi_k^{\min})^{-\alpha\rho}$$

$$\cdot (p'(z(k)|x_q^m(k)))^{1-\alpha\rho} \prod_{j=i+1}^{L} p'(z(j)|x_q^m(j))] \{ \sum_{m' \neq m} [\pi_0^{\max} \prod_{k=1}^{L} \pi_k^{\max} p'(z(k)|x_q^{m'}(k))]^{\alpha}\}^{\rho} dz^L$$

$$\text{for any } \alpha,\rho \ge 0 \qquad (2.9.2.14)$$

where $p'(z(k)|x_q^m(k))$ is given by (2.7.10) and $\pi_k^{\min}$, π_k^m, and $\pi_k^{\max}$ are defined in Section 2.5. Moreover, substituting this bound into (2.7.19), we obtain an upper bound for the overall error probability. Since it contains multidimensional integrals as in (2.9.2.14), it can not be evaluated easily. Therefore, an ensemble upper bound on the overall error probability is considered in the next section.

2.9.3 An Ensemble Upper Bound for the Overall Error Probability

Setting α equal to $1/(1+\rho)$, the bound for the error probability of the path H_m in (2.9.2.14) can be rewritten as

$$P_{E_m}(T) \le \int_{z^L} A_m(z^L) \{ \sum_{m' \neq m} B_{m'}(z^L)\}^{\rho} dz^L \qquad \text{for any } \rho \ge 0 \qquad (2.9.3.1)$$

where

$$A_m(z^L) \triangleq (\pi_0^{\min})^{-\frac{\rho}{1+\rho}} \prod_{k=1}^{L} p'(z(k)|x_q^m(k)) + \sum_{i=1}^{L} \{(\pi_0^{\min})^{-\frac{\rho}{1+\rho}} \prod_{k=1}^{i} (\pi_k^{\min})^{-\frac{\rho}{1+\rho}} p'(z(k)|x_q^m(k))^{\frac{1}{1+\rho}}$$

$$\cdot \prod_{j=i+1}^{L} p'(z(j)|x_q^m(j))\}$$

$$B_{m'}(z^L) \triangleq (\pi_0^{\max})^{\frac{1}{1+\rho}} \prod_{k=1}^{L} (\pi_k^{\max})^{\frac{1}{1+\rho}} p'(z(k)|x_q^{m'}(k))^{\frac{1}{1+\rho}} \qquad (2.9.3.2)$$

Averaging this bound over the ensemble $\mathcal{E}$ as in Section 2.8.3, we obtain the following bound for the ensemble error probability of the path H_m.

$$\overline{P_{E_m}(T)} \le \{ \prod_{k=1}^{L} \sum_{H_k^e \in \underline{H}^e} Q(H_k^e)\} \int_{z^L} A_m(z^L) \{ \sum_{m' \neq m} B_{m'}(z^L)\}^{\rho} dz^L$$

$$= \int_{z^L} \{ \sum_{H_m^e \in \underline{H}^e} Q(H_m^e) A_m(z^L)\} \{ \prod_{\substack{k=1 \\ k \neq m}}^{L} (\sum_{H_k^e \in \underline{H}^e} Q(H_k^e) \{ \sum_{m' \neq m} B_{m'}(z^L)\}^{\rho}) dz^L \qquad (2.9.3.3)$$

where ρ is any non-negative number. Hence, restricting ρ in the interval [0, 1], and using Jensen's inequality (as in Section 2.8.3), the term in braces in (2.9.3.3) can be upper bounded as follows

$$\overline{P_{E_m}(T)} \le \int_{z^L} [\sum_{H_m^e} Q(H_m^e) A_m(z^L)] [\sum_{m' \neq m} \sum_{H_{m'}^e} Q(H_{m'}^e) B_{m'}(z^L)]^{\rho} dz^L$$

$$= (M-1)^{\rho} \int_{z^L} [\sum_{H_m^e} Q(H_m^e) A_m(z^L)] [\sum_{H_{m'}^e} Q(H_{m'}^e) B_{m'}(z^L)]^{\rho} dz^L \quad \text{for any } \rho \in [0,1] \qquad (2.9.3.4)$$

The last equality follows from the fact that the summations run over the same space $\underline{H}^e$. Further using (2.8.3.1) with an argument similar to the one in Section 2.8.3, we can easily obtain the following equalities

$$\sum_{H_m^e} Q(H_m^e) A_m(z^L) = (\pi_0^{\min})^{-\frac{\rho}{1+\rho}} \prod_{k=1}^{L} \overline{p'(z(k)|x)} + \sum_{i=1}^{L} (\pi_0^{\min})^{-\frac{\rho}{1+\rho}} \prod_{k=1}^{i} (\pi_k^{\min})^{-\frac{\rho}{1+\rho}}$$

$$\cdot \overline{p'(z(k)|x)}^{\frac{1}{1+\rho}} \prod_{j=i+1}^{L} \overline{p'(z(j)|x)} \qquad (2.9.3.5)$$

and

$$\sum_{H_{m'}^{e}} Q(H_{m'}^{e}) B_{m'}(z^{L}) = \left[\prod_{k=0}^{L} (\pi_k^{max})^{\frac{1}{1+\rho}}\right] \prod_{k=1}^{L} \overline{p'(z(k)|x)^{\frac{1}{1+\rho}}} \qquad \text{for any } \rho \in [0,1]$$

where

$$\overline{p'(z(k)|x)} \triangleq \sum_{x \in X^{e}} q(x)\, p'(z(k)|x)$$

$$\overline{p'(z(k)|x)^{\frac{1}{1+\rho}}} \triangleq \sum_{x \in X^{e}} z(x)\, p'(z(k)|x)^{\frac{1}{1+\rho}} \qquad (2.9.3.6)$$

where $p'(z(k)|x)$ is given by (2.7.10), $q(\cdot)$ is an arbitrary probability density function on the set X^{e}, and X^{e} is defined in 2.8.3. Substituting these equalities into (2.9.3.4), and changing the order of integrations and multiplications, we obtain the following bound for the ensemble error probability of the path H_m

$$\overline{P_E(T)} \le F \cdot \left\{\prod_{k=1}^{L} C_k + \sum_{i=1}^{L} \prod_{k=1}^{i} (\pi_k^{min})^{-\frac{\rho}{1+\rho}} D_k \prod_{j=i+1}^{L} C_j\right\} \qquad \text{for any } \rho \in [0,1] \qquad (2.9.3.7)$$

where

$$C_k \triangleq \int_{z(k)} \overline{p'(z(k)|x)}\, \left[\overline{p'(z(k)|x)^{\frac{1}{1+\rho}}}\right]^{\rho} dz(k)$$

$$D_k \triangleq \int_{z(k)} \left[\overline{p'(z(k)|x)^{\frac{1}{1+\rho}}}\right]^{1+\rho} dz(k)$$

$$F \triangleq (M-1)^{\rho} \left(\frac{\pi_0^{max}}{\pi_0^{min}}\right)^{\frac{\rho}{1+\rho}} \left[\prod_{k=1}^{L} (\pi_k^{max})^{\frac{1}{1+\rho}}\right] \qquad (2.9.3.8)$$

where π_k^{max}, π_k^{min}, and M are as defined in Section 2.5.

If the function $g(k, \cdot, \cdot, \cdot)$ in the observation model being considered, the statistics of the observation noise $v(k)$, and the interference $I(k)$ (in the presence of interference) are all time invariant, then C_k and D_k are time invariant. Hence in this case, the bound in (2.9.3.7) becomes

$$\overline{P_{E_m}(T)} \le F \cdot \left(C_k^{L} + \sum_{i=1}^{L} \left[\prod_{k=1}^{i} (\pi_k^{min})^{-\frac{\rho}{1+\rho}}\right] D_k^{i} \cdot C_k^{L-i}\right) \qquad \text{for any } \rho \in [0,1] \qquad (2.9.3.9)$$

Since the bound in (2.9.3.7) does not depend on the path (i.e., m), it follows from (2.8.3.3) that this bound is also an upper bound for the ensemble error probability $\bar{P}_E$, for the detection of the path most probably followed by the target being considered. Furthermore, the integrals in (2.9.3.7) may not be evaluated for any $\rho \in (0,1)$. Hence the easiest bound to calculate is the one for $\rho = 1$. This fact with the nice feature of the uniformly weighted ensemble bound (see Theorem 2.8.3.4) leads us to using the bound with $\rho = 1$ and the uniform density function for $q(\cdot)$, i.e., $q(\cdot) = 1/N^{e}$ where N^{e} is defined in Section 2.8.3)) in (2.9.3.7) as the performance measure of the stack sequential decoding based smoothing algorithm.

2.10 SUBOPTIMUM DECODING BASED SMOOTHING ALGORITHM

As described before, in order to decide the path most likely followed by a target from time zero to time L, the optimum decoding based smoothing algorithm (ODSA) and the stack sequential decoding based smoothing algorithm (SSDSA) first obtain a trellis diagram (denoted by T) for the target motion (model) from time zero to time L; then use the Viterbi decoding algorithm (VDA) and stack sequential decoding algorithm respectively. The number of paths in the trellis diagram depends on L as well as n_0, n_1, n_2, ..., n_L, and the gate sizes used to reduce the target motion model to a finite state model where n_0 and n_k are the number of possible values of the discrete random vectors $x_j(0)$ and $w_j(k)$ (see Section 2.3). In particular, if L is very large (in other words, the target needs to be tracked for a long time), the trellis diagram may contain a huge amount of paths. In such a case, SSDSA may require a very large memory for the storage of stacks of searched paths and comparisons to reorder the paths in stacks according to their metrics while ODSA requires a huge memory and computation to compare the metrics of all paths in the trellis diagram. Hence, these smoothing algorithms are impractical. Therefore, a smoothing algorithm which requires a constant memory for the path most probably followed by the target from time zero to any time L is needed. In this section, such an algorithm is presented. It is based on a suboptimum decoding algorithm. Hence it does not minimize the overall error probability for the

detection of the path most likely followed by the target. It (smoothing algorithm) is referred to as Suboptimum Decoding based Smoothing Algorithm (SDSA) which is as follows:

Initial Step - After obtaining the first L(1) observations (i.e., the observation sequence from time one to time L(1)), using ODSA find the path most probably followed by the target from time zero to time L(1). Let this path be $\hat{H}^1$ (see Figure 2.10.1).

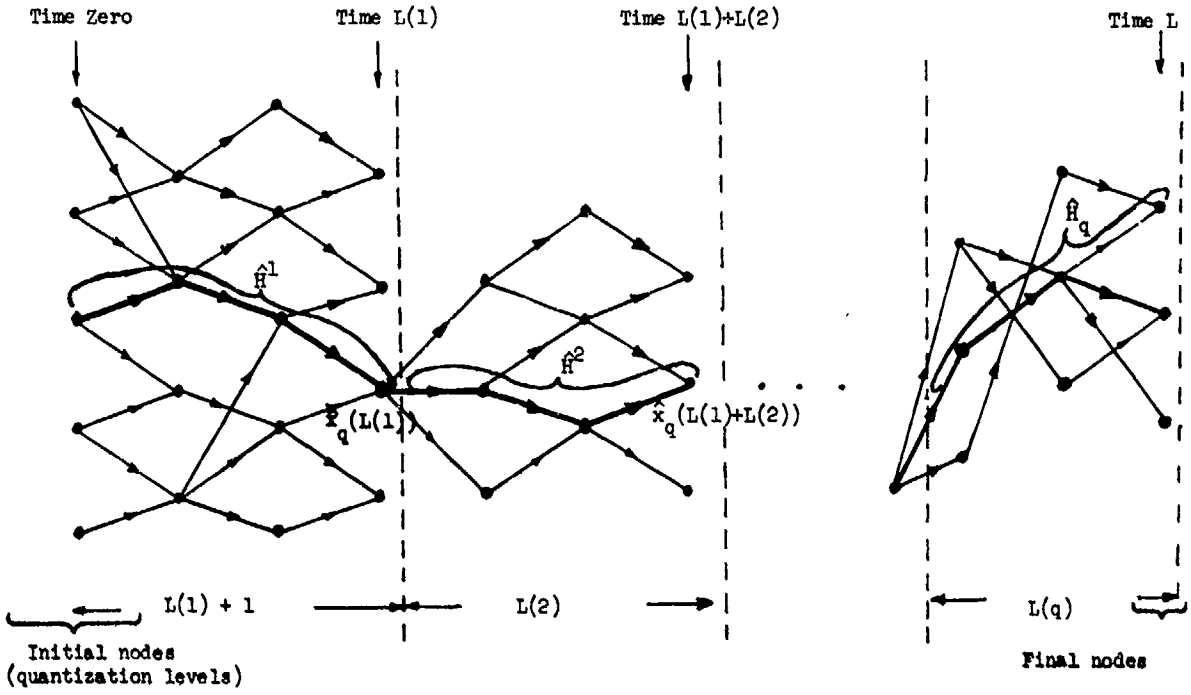


Figure 2.10.1. The Trellis Diagram for the Suboptimum Decoding based Smoothing Algorithm

First Step - Obtain the next L(2) observations (i.e., the observation sequence from time L(1)+1 to time L(1) + L(2)) and assume that the target in fact followed the path $\hat{H}^1$ from time zero to time L(1) (in other words, assume that the target was at the end point, denoted by $\hat{x}_q(L(1))$, of $\hat{H}^1$ at time L(1) with probability one). Then using ODSA, find the path most probably followed by the target from time L(1)+1 to L(1) + L(2). Let this path be $\hat{H}^2$.

Second Step - Obtain the following L(3) observations (i.e., the observation sequence from time L(1) + L(2) + 1 to time L(1) + L(2) + L(3)) and assume that the path $\hat{H}^1\hat{H}^2$ actually followed by the target from time zero to time L(1) + L(2) (in other words, assume that the target was at the end node, denoted by $\hat{x}_q(L(1) + L(2))$ of the path $\hat{H}^2$ with probability one). Then using ODSA, find the path most probably followed by the target from time L(1) + L(2) + 1 to time L(1) + L(2) + L(3). Let this path be $\hat{H}^3$. The other steps similarly continue until

$$L = \sum_{k=1}^{q} L(k)$$

At the end, decide the path composed of the paths $\hat{H}^1$, $\hat{H}^2$, ..., $\hat{H}^q$ as the path ($\hat{H}$) most probably followed by the target from time zero to time L, i.e.,

$$\hat{H} = \hat{H}^1 \hat{H}^2 \ldots \hat{H}^q$$

where q is the number of observation sequences considered from time zero to time L. The number, L(i), of observations in the i^{th} observation sequence is chosen such that at the $(i-1)^{th}$ setp of (SDSA), ODSA finds the path $\hat{H}^i$ without requiring a huge memory and computation.

Let us divide the trellis diagram T into q parts such that the first part contains L(1) + 1 columns of quantization levels, starting from time zero; the second part contains the next L(2) columns; the third part contains the following L(3) columns of quantization levels; and so on. Now we are going to define some symbols which will be used in the analyses (see Figure 2.10.2).

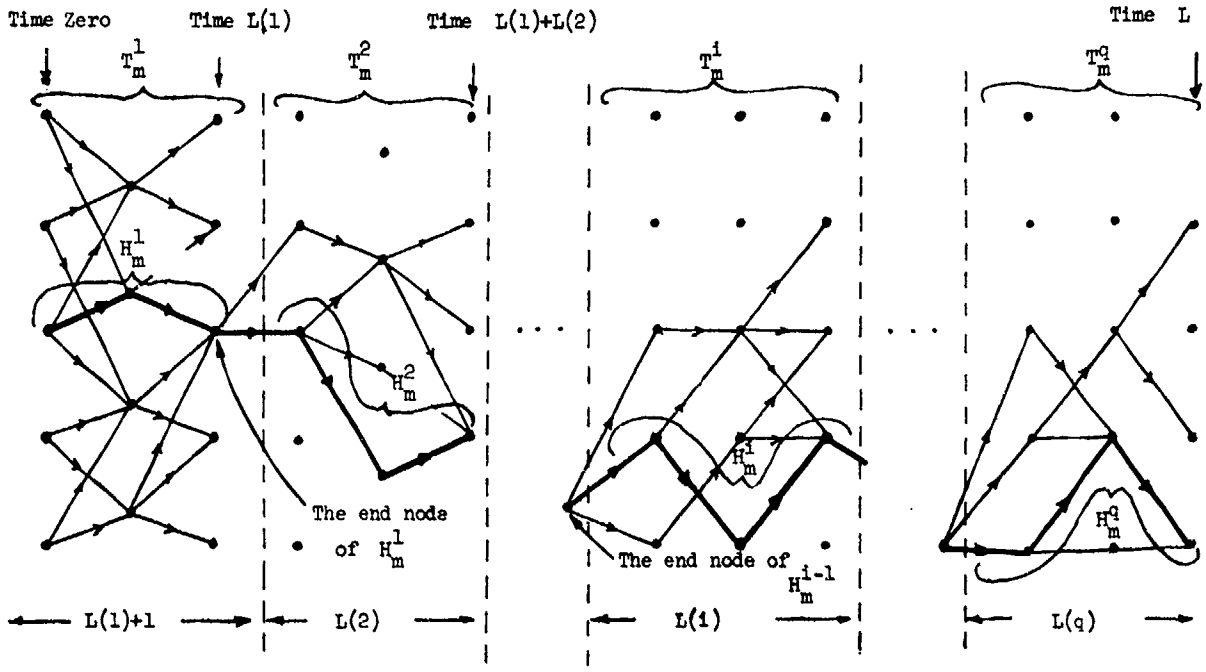


Figure 2.10.2. The Trellis Diagram for the Performance Analysis of the Suboptimum Decoding based Smoothing Algorithm

H_m is the m^{th} path through the trellis diagram T

H_m^i is the portion of H_m in the i^{th} part of T where i = 1, 2, ..., q

$H_{\hat{m}}$ is the path (throughout T) which SDSA decided that was most probably followed by the target when the target actually followed the path H_m

$H_{\hat{m}}^i$ is the portion of $H_{\hat{m}}$ in the i^{th} part of T where i = 1, 2, ..., q

T_m^1 is the first part of the trellis diagram T

T_m^i is the trellis diagram composed of the paths in the i^{th} part of T which start off at the end (final) node H_m^{i-1} where i = 2, 3, ..., q

T_m is the trellis diagram which is composed of T_m^1, T_m^2, ..., and T_m^q

M is the number of possible paths (hypotheses) through T

M_m^i is the number of possible paths through T_m^i

M_m is the multiplication of M_m^1, M_m^2, ..., and M_m^q

X_m^e is the set of all quantization levels in T_m, except the quantization levels at time zero

X_m^{1e} is the set of all quantization levels in T_m^1 except the quantization levels at time zero

X_m^{ie} is the set of all quantization levels in T_m^i where i = 2, 3, ..., q

H_m^{ie} is the set of all L(i)-tuples of X_m^{ie}

H_m^e is the cartesian product of the sets H_m^{1e}, H_m^{2e}, ..., and H_m^{qe}, that is $H_m^e = H_m^{1e} \times H_m^{2e} \times \ldots \times H_m^{qe}$

$\mathcal{E}_m^i$ is the ensemble (or set) of all M_m^i-tuples of H_m^{ie}

$\mathcal{E}_m$ is the ensemble (or set) of all M_m-tuples of H_m^e

$z^{L(i)}$ is the observation sequence used at the (i-1) step of SDSA (i.e., the observation sequence associated with the i^{th} part of T), that is

$$z^{L(i)} \triangleq \{z(n+1), z(n+2), \ldots, z(n+L(i))\}$$

where

$$n \triangleq \sum_{k=1}^{i-1} L(k); \text{ by definition, } n = 0 \text{ for } i = 1$$

The performance of the suboptimum decoding based smoothing algorithm is discussed in the next two sections.

2.10.1 An Upper Bound for the Overall Error Probability

Let us start calculating the correct detection probability of the path H_m, which is defined by the probability of choosing H_m as the path most likely followed by the target when the target in fact followed the path H_m. In other words, the correct detection probability $P_{C_m}(H_1, H_2, \ldots, H_M)$ is the probability of choosing $H_m (= H_m^1 H_m^2 \ldots H_m^q)$, given the observation sequence z^L and that the path H_m was followed by the target. Hence, we have

$$\begin{aligned} P_{C_m}(H_1,H_2,\ldots,H_M) &\triangleq \text{Prob}\{H_{\hat{m}}^1 = H_m^1, H_{\hat{m}}^2 = H_m^2, \ldots, H_{\hat{m}}^q = H_m^q | z^L, H_m\} \\ &= \text{Prob}\{H_{\hat{m}}^q = H_m^q | H_m, z^L, H_{\hat{m}}^1 = H_m^1, \ldots, H_{\hat{m}}^{q-1} = H_m^{q-1}\} \\ &\quad \cdot \text{Prob}\{H_{\hat{m}}^{q-1} = H_m^{q-1} | H_m, z^L, H_{\hat{m}}^1 = H_m^1, \ldots, H_{\hat{m}}^{q-2} = H_m^{q-2}\} \\ &\quad \ldots \text{Prob}\{H_{\hat{m}}^1 = H_m^1 | H_m, z^L\} \end{aligned} \tag{2.10.1.1}$$

On the other hand, for the suboptimum decoding based smoothing algorithm, the probability of choosing H_m^i as the decision (i.e., $H_{\hat{m}}^i = H_m^i$), given the observation sequence z^L, that the path H_m was followed by the target, and the correct detection of the paths $H_m^1, H_m^2, \ldots, H_m^{i-1}$, is the probability of choosing H_m^i, given the observation sequence $z^{L(i)}$, and that the path H_m^i was followed by the target, that is,

$$\text{Prob}\{H_{\hat{m}}^i = H_m^i | H_m, z^L, H_{\hat{m}}^1 = H_m^1, \ldots, H_{\hat{m}}^{i-1} = H_m^{i-1}\} = \text{Prob}\{H_{\hat{m}}^i = H_m^i | z^{L(i)}, H_m^i\} \tag{2.10.1.2}$$

Also we have

$$\text{Prob}\{H_{\hat{m}}^i = H_m^i | z^{L(i)}, H_m^i\} = 1 - P_{E_m}^i(T_m^i) \tag{2.10.1.3}$$

where

$$P_{E_m}^i(T^i) \triangleq \text{Prob}\{H_{\hat{m}}^i \neq H_m^i | z^{L(i)}, H_m^i\} \tag{2.10.1.4}$$

which is the error probability of H_m^i when at the $(i-1)^{th}$ step of SDSA, ODSA is used for only the trellis diagram T_m^i with the observation sequence $z^{L(i)}$. First substituting (2.10.1.3) into (2.10.1.2) and then into (2.10.1.1), we obtain the correct detection probability of H_m

$$P_{C_m}(H_1,H_2,\ldots,H_M) = \prod_{i=1}^{q} [1 - P_{E_m}^i(T_m^i)] \tag{2.10.1.5}$$

Moreover, from the definitions of the correct detection and error probabilities of the path H_m (see Definition 2.7.4), we have

$$\begin{aligned} P_{E_m}(H_1, H_2, \ldots, H_M) &\triangleq P_{E_m}(T) \\ &= 1 - P_{C_m}(H_1, H_2, \ldots, H_M) \end{aligned} \tag{2.10.1.6}$$

Hence substituting (2.10.1.5) into (2.10.1.6) we get the error probability of H_m as

$$P_{E_m}(H_1, H_2, \ldots, H_M) = 1 - \prod_{i=1}^{q} [1 - P_{E_m}^i(T_m^i)] \tag{2.10.1.7}$$

Furthermore, using a bound, denoted by $B_m^i(T_m^i)$, for the error probability of H_m^i, the error probability of H_m can be upper bounded by

$$P_{E_m}(H_1, H_2, \ldots, H_M) \leq 1 - \prod_{i=1}^{q} [1 - B_m^i(T_m^i)] \qquad (2.10.1.8)$$

where, for example, $B_m^i(T_m^i)$ is a bound of Gallager's type for the trellis diagram $T_m^{(i)}$ with the observation sequence $z^{L(i)}$ (see 2.8.2.8). Therefore substituting the bound in (2.10.1.8) for the error probability of the path H_m in (2.7.19) yields a bound for the overall error probability.

2.10.2 An Ensemble Upper Bound for the Overall Error Probability

An ensemble bound for the overall error probability for the detection of the path most likely followed by the target being considered can be obtained as follows:

First, for every path, say H_m, through the trellis diagram, T, a probability density function $Q_m^e(\cdot)$ is defined on the ensemble $\mathcal{E}_m$ such that

$$Q_m^e(T_m^e) = \prod_{i=1}^{q} Q_m^{ie}(T_m^{ie}) \qquad \text{for all } T_m^e \in \mathcal{E}_m \qquad (2.10.2.1)$$

where $Q_m^{ie}(\cdot)$ is a probability density function on the ensemble $\mathcal{E}_m^i$, and $T_m^{ie} \in \mathcal{E}_m^i$. Then averaging the error probability of the path H_i over the ensemble $\mathcal{E}_m$ with respect to the probability density fucntion in (2.10.2.1) yields the following ensemble error probability, denoted by $\overline{P_m(T)}$, of the path H_m.

$$\overline{P_{E_m}(T)} = \sum_{T_m^e \in \mathcal{E}_m} Q_m^e(T_m^e)\, P_{E_m}(T) \qquad (2.10.2.2)$$

Further, substituting (2.10.1.7) and (2.10.2.1) into (2.10.2.2), changing the order of multiplications and summations, and recognizing that the summations are performed over the entire ensembles, we obtain

$$\overline{P_{E_m}(T)} = 1 - \prod_{i=1}^{q} [1 - \overline{P_{E_m}^i(T_m^i)}] \qquad (2.10.2.3)$$

where

$$\overline{P_{E_m}^i(T_m^i)} \triangleq \sum_{T_m^{ie} \in \mathcal{E}_m^i} Q_m^{ie}(T_m^{ie})\, P_{E_m}(T_m^i) \qquad (2.10.2.4)$$

which is the ensemble error probability of the path H_m^i when only the trellis diagram T_m^i is considered. In other words, it is the ensemble average of the error probability of H_m^i (when only the trellis diagram T_m^i is considered) over the ensemble $\mathcal{E}_m^i$ with respect to a probability density function $Q_m^{ie}(\cdot)$ (see Section 2.8.3). Similarly, using (2.10.1.8), (2.10.2.1), and (2.10.2.2), we obtain the following ensemble bound for the error probability of the path H_m.

$$\overline{P_{E_m}(T)} \leq 1 - \prod_{i=1}^{q} [1 - \overline{B_m^i(T_m^i)}] \qquad (2.10.2.5)$$

where

$$\overline{B_m^i(T_m^i)} \triangleq \sum_{T_m^{ie} \in \mathcal{E}_m^i} Q_m^{ie}(T_m^{ie})\, B_m^i(T_m^i) \qquad (2.10.2.6)$$

which is an ensemble average of a bound $B_m^i(T_m^i)$ for the error probability of the path H_m^i when only the trellis diagram T_m^i is considered. Further, substituting (2.10.2.3) for the ensemble path error probabilities in (2.8.3.3) yields an ensemble average of the overall error probability for the detection of the path most likely followed by the target being considered (i.e., the ensemble error probability $\bar{P}_E$) as follows.

$$\bar{P}_E = \sum_{m=1}^{M} p(H_m) \left[1 - \prod_{i=1}^{q} [1 - \overline{P_{E_m}^i(T_m^i)}]\right)$$

$$= 1 - \sum_{m=1}^{M} p(H_m) \prod_{i=1}^{q} [1 - \overline{P_{E_m}^i(T_m^i)}] \qquad (2.10.2.7)$$

Similarly substituting (2.10.2.5) for the ensemble path error probabilities in (2.8.3.3), we obtain the following ensemble bound for the overall error probability for the detection of the path most likely followed by the target being considered.

$$\bar{P}_E \leq 1 - \sum_{m=1}^{M} p(H_m) \prod_{i=1}^{q} [1 - \overline{B_m^i(T_m^i)}] \qquad (2.10.2.8)$$

where $p(H_m)$ is as defined in Chapter 2.6.

3. APPLICATIONS OF THE SMOOTHING ALGORITHMS

As discussed before, the new smoothing algorithms developed in the previous chapters, can be used for (linear or nonlinear) discrete models with arbitrary independent (of each other and from sample to sample) random interference and noise. The following chapters consider applications of these smoothing algorithms to some discrete models with Gaussian noise and with or without arbitrary random interference in the time interval [0,L].

3.1 An Example with Gaussian Disturbance and Observation Noises

This chapter deals with the following discrete models

$$x(k+1) = f(k, x(k), u(k), w(k)) \qquad \text{(Motion Model)}$$

$$z(k) = g(k, x(k)) + v(k) \qquad \text{(Observation Model)} \qquad (3.1.1)$$

where

$x(0)$ is an n x 1 initial state Gaussian random vector with mean m_0 and covariance R_0

$w(k)$ is a p x 1 Gaussian disturbance noise vector with zero mean and covariance $R_w(k)$

$x(k)$, $u(k)$, $z(k)$ and $f(x,x(k),u(k),w(k))$ are as described in Chapter 2.1

$g(k,x(k))$ is an r x 1 (linear or nonlinear) vector

$v(k)$ is an r x 1 Gaussian observation noise vector with zero mean and covariance $R_v(k)$.

Moreover, the random vectors $x(0)$, $w(j)$, $w(k)$, $v(\ell)$ and $v(m)$ are assumed to be independent for all j, k, ℓ, m.

3.1.1 The Metric of a Branch

The observation $z(k)$ in (3.1.1) is a linear function of the Gaussian observation noise $v(k)$. Hence the conditional probability density function of $z(k)$, given that $x(k) = x_q^i(k)$, is a multivariate Gaussian density function. Thus we have

$$p'(z(k)|x_q^i(k)) \triangleq p(z(k)|x(k) = x_q^i(k))$$

$$= \frac{\exp - \{[z(k) - g(k, x_q^i(k))]^T R_v^{-1}(k)[z(k) - g(k, x_q^i(k))]/2\}}{(2\pi)^{r/2} [\det R_v(k)]^{1/2}} \qquad (3.1.1.1)$$

Substituting this into (2.7.15) yields the metric of the branch between the nodes $x_q^i(k-1)$ and $x_q^i(k)$ as

$$M(x_q^i(k-1) \to x_q^i(k)) = \ln x_k^i - \ln\{(2\pi)^{r/2}[\det R_v(k)]^{1/2}\} - [z(k) - g(k,x_q^i(k))]^T R_v^{-1}(k)[z(k) - g(k,x_q^i(k))]/2 \qquad (3.1.1.2)$$

3.1.2 The Optimum Decoding based Smoothing Algorithm

3.1.2.1 A Union Upper Bound. As mentioned before, the bound for the overall error probability, given in Chapter 2.8.2 is very complex to evaluate; hence the ensemble bound in Chapter 2.8.3 was considered. However for the models in (3.1.1), a union bound (which is very easy to evaluate) for the detection of the path (most probably followed by the target from time zero to time L) by using ODSA can be derived as follows.

Let us now consider the set Γ_i defined by (2.8.2.1). Substituting (3.1.1.1) into the metric of the path H_i (i.e., (2.7.17)) and then substituting this metric into (2.8.2.1), we obtain the set

Γ_i as

$$\Gamma_i = \{z^L:\ M'(H_j) \geq M'(H_i) \text{ for some } j \neq i\}$$

$$= \bigcup_{j=i} \{z^L:\ M'(H_j) \geq M'(H_i)\} \tag{3.1.2.1}$$

where

$$M'(H_i) \triangleq 2 \ln \pi_0^i + \sum_{k=1}^{L} \{2 \ln \pi_k^i - [z(k) - g(k, x_q^i(k))]^T R_v^{-1}(k)[z(k) - g(k, x_q^i(k))]\} \tag{3.1.2.2}$$

Recall that the set Γ_i contains the complement, $\bar{D}_m$, of the decision region D_m for the hypothesis H_m. Then from the axioms of probability, we obtain the following bound for the error probability of the path H_i

$$P_{E_i}(H_1, H_2, \ldots, H_M) \leq \text{Prob}\{z^L \in \Gamma_i | H_i\}$$

$$\leq \sum_{j \neq i} \text{Prob}\{z^L:\ M'(H_j) \geq M'(H_i) | H_i\} \tag{3.1.2.3}$$

Substituting (3.1.2.2) for $M'(H_i)$, we get

$$\text{Prob}\{z^L:\ M'(H_j) \geq M'(H_i) | H_i\} = \text{Prob}\{J_{ij} \geq A_{ij} | H_i\} \tag{3.1.2.4}$$

where

$$J_{ij} \triangleq \sum_{k=1}^{L} 2[g(k, x_q^j(k)) - g(k, x_q^i(k))]^T R_v^{-1}(k)\ z(k)$$

$$A_{ij} \triangleq 2 \ln\left[\prod_{k=0}^{L} \frac{\pi_k^i}{\pi_k^j}\right] + \sum_{k=1}^{L} \{g^T(k, x_q^j(k)) R_v^{-1}(k)\ g(k, x_q^j(k)) - g^T(k, x_q^i(k)) R_v^{-1}(k)\ g(k, x_q^i(k))\}$$

It follows that J_{ij} is a linear function of $z(k)$, which is a multivariate Gaussian density function when H_i is given. Hence, the conditional density function of J_{ij}, given H_i, is a Normal density function with

$$\text{mean} = E(J_{ij} | H_i)$$

$$= \sum_{k=1}^{L} 2[g(k, x_q^j(k)) - g(k, x_q^i(k))]^T R_v^{-1}(k)\ g(k, x_q^i(k)) \tag{3.1.2.5}$$

and

$$\text{Var}(J_{ij} | H_i) = 4 \sum_{k=1}^{L} [g(k, x_q^j(k)) - g(k, x_q^i(k))]^T R_v^{-1}(k)[g(k, x_q^j(k)) - (k, x_q^i(k))] \tag{3.1.2.6}$$

Therefore we have

$$\text{Prob}\{J_{ij} \geq A_{ij} | H_i\} = \int_{A_{ij}}^{\infty} [2\pi\ \text{Var}(J_{ij} | H_i)]^{-1/2} \exp - \frac{[J_{ij} - E(J_{ij} | H_i)]^2}{2\ \text{Var}(J_{ij} | H_i)}\ dJ_{ij}$$

$$\triangleq Q\left(\frac{A_{ij} - E(J_{ij} | H_i)}{\sqrt{\text{Var}(J_{ij} | H_i)}}\right) \tag{3.1.2.7}$$

where $Q(\cdot)$ is sometimes referred to as the Gaussian integral function and it is defined by

$$Q(y) \triangleq \int_y^{\infty} \frac{1}{\sqrt{2\pi}} \exp - \frac{x^2}{2}\ dx \tag{3.1.2.8}$$

Combining (3.1.2.7), (3.1.2.4), and (3.1.2.3) we obtain the following upper bound for the error probability of the path H_m

$$P_{E_i}(H_1, H_2, \ldots, H_M) \leq \sum_{j \neq i} Q\left(\frac{A_{ij} - E\{J_{ij}|H_i\}}{\sqrt{Var\{J_{ij}|H_i\}}}\right) \tag{3.1.2.9}$$

This bound is sometimes referred to as the union bound for the error probability of H_i. Further, substituting this bound for the path error probabilities in (2.7.19) we obtain an upper bound for the overall error probability.

3.1.2.2 The Upper Bound for the Overall Error Probability. Setting ρ equal to one in (2.8.2.8), and changing the order of multiplication and integration, we get

$$P_{E_i}(H_1,H_2,\ldots,H_M) \leq \sum_{j \neq i} \left(\prod_{k=0}^{L} \frac{\pi_k^j}{\pi_k^i}\right)^{1/2} \prod_{k=1}^{L} \int_{z(k)} [p'(z(k)|x_q^i(k))\ p'(z(k)|x_q^j(k))]^{1/2}\, dz(k) \tag{3.1.2.10}$$

From (3.1.1.1) and (A.4) in Appendix A, we have

$$B(x_q^i(k), x_q^j(k)) \triangleq \int_{z(k)} [p'(z(k)|x_q^i(k))\ p'(z(k)|x_q^j(k))]^{1/2}\, dz(k)$$

$$= \exp - \frac{[g(k, x_q^i(k)) - g(k, x_q^j(k))]^T R_v^{-1}(k)[g(k, x_q^i(k)) - g(k, x_q^j(k))]}{8} \tag{3.1.2.11}$$

Hence substituting (3.1.2.11) into (3.1.2.10) we obtain the following bound for the error probability of the path H_i.

$$P_{E_i}(H_1,H_2,\ldots,H_M) \leq \sum_{j \neq i} \left(\prod_{k=0}^{L} \frac{\pi_k^j}{\pi_k^i}\right)^{1/2} \prod_{k=1}^{L} B(x_q^i(k), x_q^j(k)) \tag{3.1.2.12}$$

Substituting this bound for the path error probabilities in (2.7.19) yields a bound for the overall error probability.

3.1.2.3 The Ensemble Upper Bound for the Overall Error Probability. The ensemble bound in (2.8.3.39) is used as the performance measure for the optimum decoding based smoothing algorithm. Using (A.4) with (3.1.1.1) in this bound, we obtain the following bound for the ensemble error probability for the detection of the path most likely followed by the target being considered

$$\bar{P}_E \leq D \cdot \prod_{k=1}^{L} \left(\sum_{x_1 \in X^e} \sum_{x_2 \in X^e} B(x_1, x_2)\right) \tag{3.1.2.13}$$

where D and $B(x_1, x_2)$ are given by (2.8.3.40) and (3.1.2.11), respectively and X^e is defined in Section 2.8.3. If the function $g(k,\cdot)$ in the observation model in (3.1.1) and the statistics of the observation noise $v(k)$ (i.e., $R_v(k)$) are time invariant, then the ensemble bound in (3.1.2.13) becomes

$$\bar{P}_E \leq D \cdot \left(\sum_{x_1 \in X^e} \sum_{x_2 \in X^e} B(x_1, x_2)\right)^L \tag{3.1.2.14}$$

3.1.3 The Stack Sequential Decoding based Smoothing Algorithm

3.1.3.1 The Ensemble Upper Bound for the Overall Error Probability. In the bound in (2.9.3.7), substituting one for ρ and $(1/N^e)$ for $q(x)$ for all x, we obtain the following bound for the ensemble error probability

$$\bar{P}_E \leq F\left(\prod_{k=1}^{L} C_k + \sum_{i=1}^{L} \prod_{k=1}^{i} (\pi_k^{min})^{-1/2} D_k \prod_{j=i+1}^{L} C_j\right) \tag{3.1.3.1}$$

where F is given in (2.9.3.8), and

$$C_k \triangleq \int_{z(k)} \overline{[p'(z(k)|x_1)]}\, \overline{[p'(z(k)|x_2))^{1/2}]}\, dz(k) \tag{3.1.3.2}$$

$$D_k \triangleq \int_{z(k)} \overline{[(p'(z(k)|x))^{1/2}]^2}\, dz(k) \tag{3.1.3.3}$$

Substituting (2.9.3.6) with (3.1.1.1) into C_k and then using (A.5) in Appendix A, we get

$$C_k = \left(\frac{1}{N^e}\right)^2 \frac{(2/9\pi)^{r/4}}{(\det R_v(k))^{1/4}} \left\{ \sum_{x_1 \in X^e} \sum_{x_2 \in X^e} \exp - \frac{[g(k,x_1) - g(k,x_2)]^T R_v^{-1}(k)[g(k,x_1) - g(k,x_2)]}{6} \right\} \quad (3.1.3.4)$$

Similarly, substituting (2.9.3.6) with (3.1.1.1) into D_k and using (A.4) we obtain

$$D_k = \left(\frac{1}{N^e}\right)^2 \left\{ \sum_{x_1 \in X^e} \sum_{x_2 \in X^e} B(x_1, x_2) \right\} \quad (3.1.3.5)$$

where $B(x_1,x_2)$ is given by (3.1.2.11); N^e and X^e are defined in Section 2.8.3.

If the covariance matrix $(R_v(k))$ of the observation sequence v(k) and the function g(k,.) are time invariant, then the bound in (3.1.3.1) becomes

$$\bar{P}_E \leq F\{C_k^L + \sum_{i=1}^{L} [\prod_{k=1}^{i} (\pi_k^{min})^{-1/2}] D_k^i C_k^{L-i}\} \quad (3.1.3.6)$$

3.2 An Example With Interference and Gaussian Disturbance and Observation Noises

In this section, we consider the following models:

$$x(k+1) = f(k, x(k), u(k), x(k)) \qquad \text{(Motion Model)}$$

$$z(k) = g(k, x(k), I(k)) + h(k, x(k), I(k))\, v(k) \qquad \text{(Observation Model)} \quad (3.2.1)$$

where

x(0), x(k), u(k), w(k), z(k) and f(k, x(k), u(k), w(k)) are as described in Section 3.1,

g(k, x(k), I(k)) and h(k, x(k), I(k)) are r x 1 and r x ℓ dimensional (linear or nonlinear) matrices, respectively,

v(k) is an ℓ x 1 Gaussian observation noise vector with zero mean and covariance $R_v(k)$,

I(k) is an m x 1 interference vector with known statistics.

Furthermore, the following assumptions are made:

(1) The random vectors x(0), w(j), w(k), v(ℓ), v(m), I(n) and I(p) are independent for all j, k, ℓ, m, n, p.

(2) $[h(k, x(k), I(k))\, R_v(k)\, h^T(k, x(k), I(k))]^{-1}$ exists for all k.

3.2.1 The Metric of a Branch

Let us consider the observation model in (3.2.1). The observation z(k) is a linear function of the normal observation noise vector v(k). Therefore, the conditional probability density function of z(k), given that $x(k) = x_q^i(k)$ and I(k) is a multivariate normal density function, namely,

$$p(z(k)|x_q^i(k), I(k)) \triangleq p(z(k)|x(k) = x_q^i(k), I(k))$$

$$= A \exp - \frac{B}{2} \quad (3.2.1.1)$$

where

$$A \triangleq (2\pi)^{-r/2} \left(\det[h(k, x_q^i(k), I(k))\, R_v(k)\, h^T(k, x_q^i(k), I(k))]\right)^{-1/2}$$

$$B \triangleq [z(k) - g(k,x_q^i(k),I(k))]^T [h(k,x_q^i(k),I(k))R_v(k)\, h^T(k,x_q^i(k),I(k))]^{-1}[z(k) - g(k,x_q^i(k),I(k))] \quad (3.2.1.2)$$

From (2.7.10), we have

$$p'(z(k)|x_q^i(k)) = \begin{cases} \int_{I(k)} p(z(k)|x_q^i(k),\ I(k))\ p(I(k))\ dI(k) & \text{using (2.5.1)} \\ \sum_{\ell=1}^{r_k} p(z(k)|x_q^i(k),\ I_{d\ell}(k))\ p(I_{d\ell}(k)) & \text{using (2.7.7)} \end{cases} \qquad (3.2.1.3)$$

where

$$p(z(k)|x_q^i(k),\ I_{d\ell}(k)) \triangleq p(z(k)|x_q^i(k),\ I(k) = I_{d\ell})$$

which is given by (3.2.1.1). Substituting (3.2.1.3) into (2.7.15) yields the metric of the branch between the nodes $x_q^i(k-1)$ and $x_q^i(k)$, i.e.,

$$M(x_q^i(k-1) \to x_q^i(k)) = \ln \pi_k^i + \ln p'(z(k)|x_q^i(k)) .$$

3.2.2 The Optimum Decoding based Smoothing Algorithm

3.2.2.1 An Upper Bound for the Overall Error Probability. Setting ρ equal to one in (2.8.2.8) yields the bound in (3.1.2.10). Hence substituting (3.2.1.3) into (3.1.2.10) we obtain a bound for the error probability of the path H_i.

If the observation model is approximated by (2.7.7), then an upper bound, which is very easy to evaluate, for the error probability of H_i can be obtained as follows: substitute the second equation (3.2.1.3) into (3.1.2.10) and then use the following inequality

$$\left(\sum_i a_i\right)^\lambda \le \sum_i a_i^\lambda \qquad \text{for any} \quad a_i \ge 0 \text{ and } \lambda \in [0,1] \qquad (3.2.1.4)$$

to obtain the following bound for the error probability of the path H_i

$$P_{E_i}(H_1,\ldots,H_M) \le \sum_{j\neq i} \left(\prod_{k=0}^{L} \frac{\pi_k^j}{\pi_k^i}\right)^{1/2} \prod_{k=1}^{L} \left\{\sum_{i=1}^{r_k}\sum_{j=1}^{r_k} [p(I_{di}(k))\ p(I_{dj}(k))]^{1/2} \cdot \int_{z(k)} [p(z(k)|x_q^i(k),\ I_{di}(k))\ p(z(k)|x_q^j(k),\ I_{dj}(k))]^{1/2}\, dz(k)\right\} \qquad (3.2.1.5)$$

where $I_{di}(k)$ and $I_{dj}(k)$ are summed over the set of all discrete values of $I_d(k)$ (see Section 2.7). The integral can be evaluated by using (3.2.1.1) in (A.1) so that we have

$$\int_{z(k)} [p(z(k)|x_q^i(k),\ I_{di}(k))\ p(z(k)|x_q^j(k),\ I_{dj}(k))]^{1/2}\, dz(k) = A' \exp + \frac{B'}{4} \qquad (3.2.1.6)$$

where

$$A' \triangleq (\det[2(R_i^{-1} + R_j^{-1})^{-1}])^{1/2} / (\det R_i)^{1/4}\ (\det R_j)^{1/4}$$

$$B' \triangleq (b_{ij}^T (R_i^{-1} + R_j^{-1})^{-1} b_{ij} - g^T(k,x_q^i(k),I_{di}(k))R_i^{-1}\, g(k,x_q^i(k),I_{di}(k)) - g^T(k,x_q^j(k),I_{dj}(k))R_j^{-1}\, g(k,x_q^j(k),I_{dj}(k)))$$

$$R_i \triangleq h(k,x_q^i(k),\ I_{di}(k))R_v(k)\ h^T(k,\ x_q^i(k),\ I_{di}(k))$$

$$R_j \triangleq h(k,x_q^j(k),\ I_{dj}(k))R_v(k)\ h^T(k,x_q^j(k),\ I_{dj}(k))$$

$$b_{ij} \triangleq R_i^{-1}\, g(k,x_q^i(k),\ I_{di}(k)) + R_j^{-1}\, g(k,x_q^j(k),\ I_{dj}(k)). \qquad (3.2.1.7)$$

Substituting this bound for the error probability of H_i in (2.7.19), we get an upper bound for the overall error probability.

3.2.2.2 An Ensemble Upper Bound for the Overall Error Probability. Substituting (3.2.1.3) with (3.2.1.1) into (2.8.3.39), we obtain an ensemble bound for detecting the path (most likely followed by the target being considered) by the optimum decoding based smoothing algorithm.

If the observation model is approximated by (2.7.7), then the ensemble bound mentioned above can be further upper bounded by using the inequality (3.2.1.4) so that we obtain the following ensemble bound, which is easy to evaluate, for the overall error probability

$$\bar{P}_E \le D \prod_{k=1}^{L} \Big\{ \sum_{x_1 \in X^e} \sum_{x_2 \in X^e} \sum_{i=1}^{r_k} \sum_{j=1}^{r_k} [p(I_{di}(k))\, p(I_{dj}(k))]^{1/2} \int_{z(k)} [p(z(k)|x_1, I_{di}(k)) p(z(k)|x_2, I_{dj})]^{1/2}\, dz(k) \Big\}$$

(3.2.2.1)

where D is defined by (2.8.3.40), and the integral is given by (3.2.1.6).

If the functions $g(k,\cdot,\cdot)$ and $h(k,\cdot,\cdot)$ in the observation model in (3.2.1), the covariance, $R_v(k)$, of the observation noise, and the statistics of the interference I(k) are time invariant, then the bound in (3.2.2.1) becomes

$$\bar{P}_E \le D\Big(\sum_{x_1 \in X^e} \sum_{x_2 \in X^e} \sum_{i=1}^{r_k} \sum_{j=1}^{r_k} [p(I_{di}(k)) p(I_{dj}(k))]^{1/2} \int_{z(k)} [p(z(k)|x_1, I_{di}(k)) p(z(k)|x_2, I_{dj}(k))]^{1/2}\, dz(k) \Big)^L$$

(3.2.2.2)

3.2.3 The Stack Sequential Decoding based Smoothing Algorithm

3.2.3.1 An Ensemble Upper Bound for the Overall Error Probability. In the bound in (2.9.3.7), substituting one for ρ and $(1/N^e)$ for q(x) for all x, and then using (3.2.1.3), we obtain an ensemble bound for the overall error probability for detecting the path (most likely followed by the target being considered) by the stack sequential decoding based smoothing algorithm.

If the observation model is approximated by (2.7.7), then the ensemble bound mentioned above can be further upper bounded by using inequality (3.2.1.4) so that we can obtain an ensemble bound which is easy to evaluate as follows. Using the inequality (3.2.3.1), we get

$$[p'(z(k)|x)]^{1/2} \le \sum_{i=1}^{r_k} [p(z(k)|x, I_{di}(k))]^{1/2}\, [p(I_{di}(k))]^{1/2} \qquad (3.2.3.1)$$

then substituting this bound into C_k and D_k in (2.9.3.8), we obtain

$$C_k\Big|_{\substack{\rho=1 \\ q(\cdot)=1/N^e}} \le \Big(\frac{1}{N^e}\Big)^2 \Big\{ \sum_{i=1}^{r_k} \sum_{j=1}^{r_k} p(I_{di}(k))\, [p(I_{dj}(k))]^{1/2} \sum_{x_1 \in X^e} \sum_{x_2 \in X^e} \Big[\int_{z(k)} p(z(k)|x_1, I_{di}(k)) \cdot (p(z(k)|x_2, I_{dj}(k)))^{1/2}\, dz(k) \Big] \Big\} \qquad (3.2.3.2)$$

$$\triangleq C_k^b$$

and

$$D_k\Big|_{\substack{\rho=1 \\ q(\cdot)=1/N^e}} \le \Big(\frac{1}{N^e}\Big)^2 \Big\{ \sum_{i=1}^{r_k} \sum_{j=1}^{r_k} [p(I_{di}(k))\, p(I_{dj}(k))]^{1/2} \sum_{x_1 \in X^e} \sum_{x_2 \in X^e} \int_{z(k)} [p(z(k)|x_1, I_{di}(k)) \cdot p(z(k)|x_2, I_{dj}(k))]^{1/2}\, dz(k) \Big\} \qquad (3.2.3.3)$$

$$\triangleq D_k^b$$

where the integral in (3.2.3.3) is given by (3.2.1.6) and the integral in C_k can be evaluated by substituting (3.2.1.1) into it and then using (A.3) in Appendix A so that we have

$$\int_{z(k)} p(z(k)|x_1, I_{di}(k))\, [p(z(k)|x_2, I_{dj}(k))]^{1/2}\, dz(k) = A' \exp \frac{B'}{4} \qquad (3.2.3.4)$$

where

$$A' \triangleq (\det[(R_1^{-1} + \frac{R_2^{-1}}{2})^{-1}])^{1/2}/(2\pi)^{r/4}\, (\det R_1)^{1/2}\, (\det R_2)^{1/4}$$

$$B' \triangleq b_{12}^T [2R_1^{-1} + R_2^{-1}]^{-1} b_{12}' - 2g^T(k,x_1,I_{di}(k))R_1^{-1} g(k,x_1,I_{di}(k)) - g^T(k,x_2,I_{dj}(k))R_1^{-1} g(k,x_2,I_{dj}(k))$$

$$b'_{12} \triangleq 2R_1^{-1} g(k,x_1,I_{di}(k)) + R_2^{-1} g(k,x_2,I_{dj}(k))$$

$$R_1 \triangleq h(k,x_1,I_{di}(k))\ R_v(k)\ h^T(k,x_1,I_{di}(k))$$

$$R_2 \triangleq h(k,x_2,I_{dj}(k))\ R_v(k)\ h^T(k,x_2,I_{dj}(k)).$$

Finally, substituting these bounds for C_k and D_k in (2.9.3.7), we obtain the following ensemble bound for the overall error probability

$$\bar{P}_E \leq F\{\prod_{k=1}^{L} C_k^b + \sum_{i=1}^{L} \prod_{k=1}^{i} (\pi_k^{min})^{-1/2}\ D_k^b \prod_{j=i+1}^{L} C_j^b\} \qquad (3.2.3.5)$$

where F is as given in (2.9.3.8). If the functions $g(k,\cdot,\cdot)$ and $h(k,\cdot,\cdot)$, and the covariance matrix, $R_v(k)$, of the observation noise v(k) are time invariant, then C_k^b and D_k^b become time invariant. Hence, in this case, the bound in (3.2.3.5) can be rewritten as follows

$$\bar{P}_E \leq F\{(C_k^b)^L + \sum_{i=1}^{L} [\prod_{k=1}^{i} (\pi_k^{min})^{-1/2}]\ (D_k^b)^i\ (C_k^b)^{L-i}\}\ . \qquad (3.2.3.6)$$

4. NUMERICAL EXPERIMENTS

The purpose of simulating was to find out how well the smoothing algorithms, developed in Section 2, perform both in a clear environment and in the presence of interference.

In a clear environment, the aim was to compare the smoothing algorithms with the Kalman filter algorithm for linear discrete models and the extended Kalman filter algorithm for nonlinear discrete models. However, in the presence of interference, the smoothing algorithms may not be compared with the (extended) Kalman filter algorithm since it cannot handle the case of interference. Therefore, the purpose was to discover how good the estimates produced by the smoothing algorithms are, and also to observe the estimates obtained by the (extended) Kalman filter algorithm (which considers only observation noise, i.e., with zero interference). All of this was done for both linear and nonlinear discrete models with interference.

For all simulations, the IBM Systems/370 Model 3033, Fortran IV, and IMSL library were used. For each simulation, the disturbance noise w(k), observation noise v(j), initial state x(0), and interference I(ℓ) (i.e., in the presence of interference) were taken to be white Gaussian and also independent of each other. For a discrete random variable (with a given number of possible values) which approximates the Gaussian random variable with mean μ and variance σ^2, the one in (B.9) in Appendix B was used. In addition, the approximate observation model (2.7.7) was used in all the cases of interference.

Simulation results are presented in figures. At the top left corner of each figure, the used models, noise statistics, gate size, and the number of possible values of the discrete random variables $w_d(\cdot)$, $I_d(\cdot)$, and $x_d(0)$ (which approximate the disturbance noise $w(\cdot)$, interference $I(\cdot)$ and initial state x(0)) are provided.

The following abbreviations and terms are used in the figures:

AAEK represents the average absolute error for the (extended) Kalman filter estimates. The absolute error at time j and the average absolute error are defined as follows:

$$\text{ABSOLUTE ERROR (at time j)} \triangleq |x(j) - \hat{x}_k(j|j)| \qquad (4.1)$$

$$\text{AAEK} \triangleq \frac{1}{L+1} \sum_{j=0}^{L} |x(j) - \hat{x}_k(j|j)| \qquad (4.2)$$

where L is the time which the target was tracked up to and including, $\hat{x}_k(j|j)$ is the (extended) Kalman estimate of the state x(j), given the observation sequence from time one to time j.

AAEOP represents the average absolute error for the estimates obtained by the smoothing algorithm used. The absolue error at time j and the average absolute error are defined as follows:

$$\text{ABSOLUTE ERROR (at time j)} \triangleq |x(j) - \hat{x}_s(j)| \qquad (4.3)$$

$$\text{AAEOP} \triangleq \frac{1}{L+1} \sum_{j=0}^{L} |x(j) - \hat{x}_s(j)| \qquad (4.4)$$

where L is as defined above, and $\hat{x}_s(j)$ is the estimate of the state x(j), obtained by the smoothing algorithm used.

ACTUAL stands for the actual values of the states

BOUND represents the bound in (3.1.2.14) for the optimum decoding based smoothing algorithm (ODSA) using an example without interference, the bound in (3.2.2.2) for ODSA using an example with interference, the bound in (3.1.3.6) for the stack sequential decoding based smoothing algorithm (SSDSA) using an example without interference, or the bound in (3.2.3.6) for SSDSA using an example with interference.

ER.COV. represents the estimation error covariance matrix for the (extended) Kalman filter algorithm. The estimation error covariance matrix at time j is defined by

$$E\{(x(j) - \hat{x}_k(j|j))\ (x(j) - \hat{x}_k(j|j))^T\}$$

where E{ } stands for expectation. Obviously in a scalar case, this matrix reduces to the mean square error

EX.KAL. represents the extended Kalman filter algorithm used

E(A(·)) stands for the expectation of the random variable A(·)

GATE SIZE represents the gate size used for the quantization

KALMAN stands for the Kalman filter algorithm used

NUM. OF DISC. FOR A(·) represents the number of possible values of the discrete random variable (used for the simulation) which approximates the random variable A(·)

OPD stands for the optimum decoding based smoothing algorithm used

SOD stands for the suboptimum decoding based smoothing algorithm used

SSD stands for the stack sequential decoding based smoothing algorithm used

VAR(A(·)) stands for the variance of the random variable A(·).

4.1 The Optimum Decoding Based Smoothing Algorithm

Many examples were simulated with the optimum decoding based smoothing algorithm and the (extended) Kalman filter algorithm. The simulation results of some of them are presented in Figure 4.1.1a-4c. The simulations were stopped after seven steps because of the exponentially growing memory requirement of the optimum decoding based smoothing algorithm. For each example, the simulation results are presented in three figures.

The first figure presents the variations of the actual and estimated values of the states versus time. The actual values are marked by Symbol ○ , the (extended) Kalman filter estimates by Symbol △ , and the OPD estimates (i.e., the estimates obtained by the optimum decoding based smoothing algorithm) by +. The second figure presents the variation of the estimation error covariance matrix (for the (extended) Kalman filter algorithm) versus time as well as the bound in (3.1.2.14) (if the example does not have any interference) or the bound in (3.2.2.2) (if the example contains interference). This bound is used as the performance measure of the optimum decoding based smoothing algorithm while the error covariance matrix is used as the performance measure of the (extended) Kalman filter algorithm. The third figure presents two curves as well as the average absolute errors for the (extended) Kalman estimates and the OPD estimates. One of these curves shows the variation of the absolute errors for the (extended) Kalman estimates versus time and is marked by △ . The other curve shows the variation of the absolute error for the OPD estimates, and is marked by Symbol +.

4.2 The Stack Sequential Decoding Based Smoothing Algorithm

A large number of examples were simulated with the stack sequential decoding based smoothing algorithm and the (extended) Kalman filter algorithm. The simulation results of some of them are presented in Figure 4.2.1a-4c. For each example, the simulation results are presented in three figures (as in Section 4.1).

The first figure presents the variations of the actual values, the (extended) Kalman estimates, and the SSD estimates (i.e., the estimates obtained by the stack sequential decoding based smoothing algorithm) of the states versus time. The second figure presents the estimation error covariance matrix versus time as well as the bound in (3.1.3.6) (if the example does not have any interference) or the bound in (3.2.3.6) (if the example contains interference). This bound is used as the performance measure of the stack sequential decoding based smoothing algorithm. The third figure presents the variations of the absolute errors, and the average absolute errors for the (extended) Kalman estimates and the SSD estimates.

4.3 The Suboptimum Decoding Based Smoothing Algorithm

Many examples were simulated with the (extended) Kalman filter algorithm and the suboptimum decoding based smoothing algorithm considering three steps at each of which six observations were

were used. The simulation results of some of them are presented in Figure 4.3.1a-4c. For each example, the simulation results are presented in three figures (as in Section 4.1).

The first figure presents the variations of the actual values, the (extended) Kalman estimates, and the SOD estimates (i.e., the estimates obtained by the suboptimum decoding based smoothing algorithm) of the states versus time. The second one presents the variation of the estimation error covariance matrix for the (extended) Kalman filter algorithm. The third figure presents the average absolute errors, and the variations of the absolute errors for the (extended) Kalman filter estimates and the SOD estimates.

4.4 Comments

Let m_k, n_0 and r_k be the numbers of possible values of the discrete random variables $w_d(k)$, $x_d(0)$, and $I_d(k)$, which approximate the disturbance noise $w(k)$, initial state $x(0)$, and interference $I(k)$, respectively. These numbers were taken to be time invariant for all simulations. Definitely, the performance of the smoothing algorithms depends on these numbers as well as the gate size used. The smoothing algorithm produces better estimates of the states for a suitable gate size, and larger m_k, n_0, r_k. This follows from the fact that the disturbance noise, initial state, and interference are approximated better for larger m_k, n_0, and r_k. For a large gate size, good estimates of the states are not expected since more quantization errors are made.

The ensemble bounds of Gallager's type is used as the performance measure of the smoothing algorithms. The values of these bounds depend on the models as well as the quantization (i.e., gate size, m_k, n_0, and r_k) used. Since some inequalities are used to derive these bounds, they can be greater than or equal to one for some models or quantization used. It should also be noted that even in the cases where these ensemble bounds are less than one, they do not give complete information about the performance of the smoothing algorithms (see Theorem 2.8.3.4).

5. CONCLUSIONS

Three completely new smoothing algorithms have been presented for the following type of discrete models with or without random interference: they can be linear or nonlinear; the disturbance noise, observation noise, and interference can be any independent (of each other, and from sample (time) to sample) not necessarily Gaussian noises; the disturbance noise and interference can affect the motion (model) and observation (model) in a general and not necessarily linear way; functions which define the models do not even have to be continuous.

The new smoothing algorithms are based on the quantization of states to a finite set of states and Decoding Technique of Information Theory. If the quantization errors are neglected, the first smoothing algorithm, which is referred to as the optimum decoding based smoothing algorithm, is optimum with respect to the minimum error probability criterion (Bayes' decision rule). However, this requires an evergrowing amount of memory with time for its implementation. Hence the second algorithm, which is referred to as the stack sequential decoding based smoothing algorithm, has been proposed. Even this algorithm requires a growing amount of memory with time (but not as fast as the first algorithm) for its implementation. Therefore, the third smoothing algorith, which is referred to as the suboptimum decoding based smoothing algorithm, has been proposed. This algorithm requires a finite amount of memory for any time (i.e., no matter how long the target is tracked).

In target tracking, it is very difficult to determine a motion model which is analytically tractable and, at the same time, satisfactorily close to reality. Therefore, by making many approximations, an analytically tractable motion model is obtained. However, the smoothing algorithms presented in this chapter uses this motion model only to determine the transition probabilities of the tracked target from gate to gate. If these transition probabilities can be determined in another way, these smoothing algorithms can be used with only an observation model so that the motion model (which is approximate) is not necessary any more. Hence the more accurate the transition probabilities are determined, the more accurate the estimates will be.

The ensemble bounds of Gallager's type have been derived and used as the performance measure of the new smoothing algorithms. These bounds are sometimes totally useless since they become numbers which are greater than or equal to one.

In order to test the smoothing algorithms, Digital Computer Simulations have been performed. Some of the simulation results are presented in detail. These results show that for both linear and nonlinear discrete models with interference, these smoothing algorithms perform very well even though neither the Kalman nor the extended Kalman filter algorithm is capable of handling interference. Also, these smoothing algorithms perform better than the extended Kalman filter algorithm for some nonlinear models with Gaussian noises (and without interference) while they perform almost as well as the Kalman filter algorithm for linear models with Gaussian noises (and without interference).

APPENDICES

A. Theorem

Let $p(z|x_i)$ and $p(z|x_j)$ be two r-dimensional (multivariate) Gaussian density functions such that

$$p(z|x_i) = (2\pi)^{-r/2} (\det R_i)^{-1/2} \exp - \frac{(z - g(x_i))^T R_i^{-1} (z - g(x_i))}{2}$$

$$\triangleq N(g(x_i), R_i)$$

and

$$p(z|x_j) \triangleq N(g(x_j), R_j).$$

Then the following equalities hold:

$$I \triangleq \int_z [p(z|x_i)\ p(z|x_j)]^{1/2}\ dz = A \exp\{B/4\} \tag{A.1}$$

where

$$A \triangleq \{\det[2(R_i^{-1} + R_j^{-1})^{-1}]\}^{1/2}/(\det R_i)^{1/4}\ (\det R_j)^{1/4}$$

$$B \triangleq \{b_{ij}^T (R_i^{-1} + R_j^{-1})^{-1}\ b_{ij} - g(x_i)^T R_i^{-1}\ g(x_i) - g(x_j)^T R_j^{-1}\ g(x_j)\}$$

$$b_{ij} \triangleq R_i^{-1}\ g(x_i) + R_j^{-1}\ g(x_j) \tag{A.2}$$

$$II \triangleq \int_z p(z|x_i)\ [p(z|x_j)]^{1/2}\ dz = A' \exp\{B'/4\} \tag{A.3}$$

where

$$A' \triangleq \{\det[(R_i^{-1} + \frac{R_i^{-1}}{2})^{-1}]\}^{1/2}/(2\pi)^{r/4}\ (\det R_i)^{1/2}\ (\det R_j)^{1/4}$$

$$B' \triangleq b_{ij}'^T [2R_i^{-1} + R_j^{-1}]^{-1}\ b_{ij}' - 2g(x_i)^T R_i^{-1}\ g(x_i) - g(x_j)^T R_j^{-1}\ g(x_j)$$

$$b_{ij}' \triangleq 2R_i^{-1}\ g(x_i) + R_j^{-1}\ g(x_j).$$

If the covariance matrices R_i and R_j are equal, say, $R_i = R_j \triangleq R_v$, then the equalities above become

$$III \triangleq \int_z [p(z|x_i)p(z|x_j)]^{1/2}\ dz = \exp - \frac{[g(x_i) - g(x_j)]^T R_v^{-1}[g(x_i) - g(x_j)]}{8} \tag{A.4}$$

and

$$IV \triangleq \int_z p(z|x_i)[p(z|x_j)]^{1/2}\ dz = G \exp - \frac{[g(x_i) - g(x_j)]^T R_v^{-1}[g(x_i) - g(x_j)]}{6} \tag{A.5}$$

where

$$G \triangleq (\frac{2}{9\pi})^{r/4}/(\det R_v)^{1/4}\ .$$

Proof: From the definitions of the probability density functions, it is easily obtained that

$$[p(z|x_i)\ p(z|x_j)]^{1/2} = C \exp - \frac{D}{4} \tag{A.6}$$

where

$$C \triangleq 1/[(2\pi)^{r/2}\ (\det R_i)^{1/4}\ (\det R_j)^{1/4}]$$

$$D \triangleq (z - g(x_i))^T R_i^{-1}(z - g(x_i)) + (z - g(x_j))^T R_j^{-1}(z - g(x_j)) \tag{A.7}$$

Also, D can be rewritten as

$$D = (z - a)^T (R_i^{-1} + R_j^{-1})\ (z - a) - B \tag{A.8}$$

where

$$a \triangleq (R_i^{-1} + R_j^{-1})^{-1} b_{ij}$$

$$b_{ij} \triangleq R_i^{-1} g(x_i) + R_j^{-1} g(x_j) \tag{A.9}$$

B is as defined in (A.2).

Therefore, we have

$$[p(z|x_i)p(z|x_j)]^{1/2} = \frac{\exp[-\frac{(z-a)^T (R_i^{-1} + R_j^{-1})(z-a)}{4} + \frac{B}{4}]}{(2\pi)^{r/2} (\det R_i)^{1/4} (\det R_j)^{1/4}} \tag{A.10}$$

Multiplying the denominator and numerator by

$$\{\det[2(R_i^{-1} + R_j^{-1})^{-1}]\}^{1/2}$$

gives

$$[p(z|x_i)p(z|x_j)]^{1/2} = [A \exp \frac{B}{4}] \{\frac{\exp - [(z-a)^T \frac{R_i^{-1} + R_j^{-1}}{2} (z-a)/2]}{(2\pi)^{r/2}[\det[2(R_i^{-1} + R_j^{-1})^{-1}]]^{1/2}}\} \tag{A.11}$$

It should be noted that the term in the braces is a multivariate Gaussian density function with mean a and covariance = $2(R_i^{-1} + R_j^{-1})^{-1}$. Hence, integrating this term over all z yields one, so that we obtain the equality in (A.1).

For the proof of the equality in (2.3), let us rewrite the integral in (2.3) as

$$\int_z p(z|x_i)[p(z|x_j)]^{1/2} dz = \int_z [p(z|x_i)^2 p(z|x_j)]^{1/2} dz \tag{A.12}$$

On the other hand, from the equality that

$$\det R_i = 2^r \det(R_i/2) \tag{A.13}$$

we have

$$p^2(z|x_i) = G \cdot p_1(z|x_i) \tag{A.14}$$

where

$$p_1(z|x_i) \triangleq N(g(x_i), \frac{R_i}{2})$$

and

$$G \triangleq 1/2^{r/2} (2\pi)^{r/2} (\det R_i)^{1/2} \tag{A.15}$$

Substituting (A.14) into (A.12) yields

$$\int_z p(z|x_i)[p(z|x_j)]^{1/2} dz = G^{1/2} \int_z [p_1(z|x_i)p(z|x_j)]^{1/2} dz \tag{A.16}$$

and using the quality in (A.1) for the integral on the right hand side in (A.16), we can easily obtain the equality in (A.3).

If $R_i = R_j \triangleq R_v$, substituting R_v for R_i and R_j in the equalities in (A.1) and (A.3), the equalities in (A.4) and (A.5) can readily be verified. This proves the theorem.

B. Approximation of an Absolutely Continuous Random Vector by a Discrete Random Vector

Let n be a given positive integer and let D^m be the set of all distribution functions of all m x 1 discrete random vectors with n possible values where superscript m stands for the dimensionality of random vectors. Then the problem of approximating an absolutely continuous m x 1 random vector X^m with distribution function $F_{X^m}(\cdot)$ by an m x 1 discrete random vector with n possible values is to find a distribution function $F_{y_o^m}(\cdot) \in D^m$ which minimizes the following objective function over the set D^m.

$$J_m(F_{y^m}(\cdot)) \triangleq \int_{R^m} [F_{X^m}(a) - F_{y^m}(a)]^2 \, da; \quad F_{y^m}(\cdot) \in D^m \tag{B.1}$$

That is

$$J(F_{y_o^m}(\cdot)) = \min_{F_{y^m}(\cdot) \in D^m} J(F_{y^m}(\cdot)) \tag{B.2}$$

The discrete random vector defined by $F_{y_o^m}(\cdot)$ is referred to as the optimum discrete random vector approximating the random vector X^m.

Here, the approximation of an absolutely continuous random variable X with distribution $F_X(\cdot)$ by a discrete random variable with n possible values is considered. The necessary conditions that the optimum discrete random variable approximating X must satisfy is obtained. Finally, discrete random variables approximating normal random variables are obtained.

Let us now state two theorems and define some symbols which will be used. The proofs of the theorems are given in Reference [18].

Theorem B.1[18]

Let $f(y) \triangleq f(y_1, y_2, \ldots, y_\ell)$ be a real valued function on an open set Γ of R^ℓ and let f(y) have finite partial derivatives $\partial f(y)/\partial y_k$, $k = 1, 2, \ldots, \ell$ at each point of Γ. If f(y) has a local minimum at the point $y_o \triangleq (y_{1,0}, y_{2,0}, \ldots, y_{\ell,0})$ in Γ, then

$$\frac{\partial}{\partial y_k} f(y)\Big|_{y=y_0} = 0 \text{ for each } k = 1, 2, \ldots, \ell$$

Theorem B.2[18]

Let $f(y) \triangleq f(y_1, y_2, \ldots y_\ell)$ be a real valued function on an open set Γ of R^ℓ and let f(y) have continuous second-order partial derivatives on Γ. Let $y_0 \triangleq (y_{1,0}, y_{2,0}, \ldots, y_{\ell,0})$ be a point of Γ for which

$$\frac{\partial}{\partial y_k} f(y)\Big|_{y=y_0} = 0 \text{ for each } k = 1, 2, \ldots, \ell$$

Assume that the determinant

$$G \triangleq \det([\nabla^2 f(y)]|_{y=y_0}) \neq 0$$

where

$$[\nabla^2 f(y)]_{ij} \triangleq \frac{\partial^2}{\partial y_i \partial y_j} f(y)$$

Let G_{n-k} be the determinant obtained from G by deleting the last k rows and columns. If the n numbers $G_1, G_2, \ldots, G_\ell$ are all positive, then f(y) has a local minimum at y_0.

D is the set of all distribution functions of all discrete random variables with n possible values where n is a given positive integer.

S is the set of all step function with n+1 steps where a step function with n+1 steps (where n is given) is defined throughout R (real line) and is constant on each one of n+1 non-intersecting intervals whose union is R. This constant is zero on the interval containing $-\infty$ and is one on the interval containing ∞. That is

$$S \triangleq \{g(x): g(x) = 0 \text{ for } x < y_1;\ g(x) = P_i \text{ for } y_i \leq x < y_{i+1},\ P_i \in (0,1);\ g(x) = 1 \text{ for } x \geq y_n;\ y_{i+1} > y_i,\ y_i \in (-\infty,\infty);\ i = 1, 2, \ldots, n-1\}$$

In order to find an optimum discrete random variable with n possible values (where n is a given positive integer) that approximates an absolutely continuous random variable X with distribution function $F_x(\cdot)$, we must find a distribution function $F_{y_0}(\cdot)$ which minimize the following objective function over the set D

$$J(F_y(\cdot)) \triangleq \int_{-\infty}^{\infty} [F_x(a) - F_y(a)]^2 \, da \tag{B.3}$$

Namely,

$$J(F_{y_0}(\cdot)) = \min_{F_y(\cdot) \in D} J(F_y(\cdot)) \tag{B.4}$$

$$= \min_{g(\cdot) \in S} J(g(\cdot)) \tag{B.5}$$

The last equality follows from the following arguments. Let a step function $g_0(\cdot) \in S$ minimize (B.3) over the set S; since the distribution function is nondecreasing, $g_0(\cdot)$ must be nondecreasing, hence it is a nondecreasing step function (from zero to one); therefore $g_0(\cdot) \in D$. Thus the aim is to find a step function $g_0(\cdot) \in S$ which minimize (B.3) over S · i.e., we would like to minimize the following function over $y_i \in (-\infty,\infty)$ and $P_j \in (0,1)$ (where $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, n-1$).

$$J(g(\cdot)) = \int_{-\infty}^{y_1} F_x^2(a)\, da + \int_{y_1}^{y_2} [F_x(a) - P_1]^2 \, da + \int_{y_2}^{y_3} [F_x(a) - P_2]^2 \, da + \ldots + \int_{y_{n-1}}^{y_n} [F_x(a) - P_{n-1}]^2 \, da + \int_{y_n}^{\infty} [F_x(a) - 1]^2 \, da \tag{B.6}$$

It follows from Theorem B.1 that if $g_0(x)$ which is defined by

$$g_0(x) = \begin{cases} 0 & x < y_{1,0} \\ P_{i,0} & \text{If } y_{i,0} \leq x < y_{i+1,0};\ i = 1, 2, \ldots, n-1 \\ 1 & x > y_{n,0} \end{cases} \tag{B.7}$$

is a step function (distribution function) which minimize (B.6). This must satisfy the following set of equations

$$
\begin{aligned}
P_{1,0} &= 2F_x(y_{1,0}) \\
P_{1,0} + P_{2,0} &= 2F_x(y_{2,0}) \\
P_{2,0} + P_{3,0} &= 2F_x(y_{3,0}) \\
&\vdots \\
P_{n-2,0} + P_{n-1,0} &= 2F_x(y_{n-1,0}) \\
1 + P_{n,0} &= 2F_x(y_n) \\
P_{1,0}(y_{2,0} - y_{1,0}) &= \int_{y_{1,0}}^{y_{2,0}} F_x(a)\, da \\
P_{2,0}(y_{3,0} - y_{2,0}) &= \int_{y_{2,0}}^{y_{3,0}} F_x(a)\, da \\
&\vdots \\
P_{n-1,0}(y_{n,0} - y_{n-1,0}) &= \int_{y_{n-1,0}}^{y_{n,0}} F_x(a)\, da
\end{aligned}
\qquad (B.8)
$$

Using these equations and Theorem B.2, the discrete random variables which approximate the normal random variable with zero mean and unit variance have been numerically obtained, and they are tabulated in Table B.1.

Let y_0 be the discrete random variable with n possible values $y_{1,0}, y_{2,0}, \ldots, y_{n,0}$ which approximate the normal random variable with zero mean and unit variance and let $P_{i,0}$ be such that

$$P_{i,0} = \text{Prob}(y_0 = y_{i,0})$$

Let z_0 be the discrete random variable with n possible values $z_{1,0}, z_{2,0}, \ldots, z_{n,0}$ which approximates the normal random variable with mean μ and variance σ^2 and let $P'_{i,0}$ be such that

$$P'_{i,0} \triangleq \text{Prob}(z = z_{i,0}) .$$

Through the equations in (B.8), it can easily be verified that

$$
\begin{aligned}
z_{i,0} &= \sigma y_{i,0} + \mu \\
P'_{i,0} &= P_{i,0}
\end{aligned}
\qquad i = 1, 2, \ldots, n \qquad (B.9)
$$

TABLE B.1. Discrete Random Variables Approximating the Gaussian Random Variable with Zero Mean and Unit Variance

Number of Possible Values of y_0^*	n possible Values and Corresponding Probabilities of y_0^*								
	i**	1	2	3	4	5	6	7	8
n = 1	$y_{i,0}$	0.000							
	$P_{i,0}$	1.000							
n = 2	$y_{i,0}$	-0.675	0.675						
	$P_{i,0}$	0.500	0.500						
n = 3	$y_{i,0}$	-1.005	0.0	1.005					
	$P_{i,0}$	0.315	0.370	0.315					
n = 4	$y_{i,0}$	-1.219	-0.355	0.355	1.219				
	$P_{i,0}$	0.223	0.277	0.277	0.223				
n = 5	$y_{i,0}$	-1.376	-0.592	0.0	0.592	1.376			
	$P_{i,0}$	0.169	0.216	0.230	0.216	0.169			
n = 6	$y_{i,0}$	-1.499	-0.767	-0.242	0.242	0.767	1.499		
	$P_{i,0}$	0.134	0.175	0.191	0.191	0.175	0.134		
n = 7	$y_{i,0}$	-1.599	-0.905	-0.423	0.0	0.423	0.905	1.599	
	$P_{i,0}$	0.110	0.145	0.162	0.166	0.162	0.145	0.110	
n = 8	$y_{i,0}$	-1.683	-1.018	-0.567	-0.183	0.183	0.567	1.018	1.683
	$P_{i,0}$	0.093	0.123	0.139	0.145	0.145	0.139	0.123	0.093

*y_0^* is the discrete random variable with n possible values $y_{1,0}, y_{2,0}, \ldots, y_{n,0}$ which approximates the normal random variable with zero mean and unit variance.

**$y_{i,0}$ is the i^{th} possible value of y_0^*

$P_{i,0} \triangleq \text{Prob}(y_0^* = y_{i,0})$

REFERENCES

1. Demirbaş, K. — *Target Tracking in the Presence of Interference*, Ph.D Thesis, University of California at Los Angeles, Los Angeles, California, January 1981.

2. Jazwinski, A. H. — *Limited Memory Optimal Filtering*, IEEE Trans. Automatic Contr., Vol. AC-13, October 1968.

3. Thorp, J. S. — *Optimal Tracking of Maneuvering Targets*, IEEE Trans. Aerospace Electronics Systems, Vol. AES-9, July 1973.

4. Singer, R. A. — *Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets*, IEEE Trans. Aerospace Electronics Systems, July 1970.

5. Howard, R. A. — *System Analysis of Semi-Markov Processes*, IEEE Trans. Mil. Electron., Vol. MIL-8, pp. 114-124, April 1974.

6. Gholson, N. H.
Moose, R. L. — *Maneuvering Target Tracking Using Adaptive State Estimation*, IEEE Trans. Aerospace Electronics Systems, May 1977.

7. Moose, R. L.
VanLandingham, H. F.
McCabe, D. H. — *Modelling and Estimation for Tracking Maneuvering Targets*, IEEE Trans. Aerospace Electronics Systems, Vol. AES-15, No. 3, May 1979.

8. Farina, A.
Pardini, S. — *Multiradar Tracking Systems Using Radial Velocity Measurements*, IEEE Trans. Aerospace Electronics Systems, Vol. AES-15, No. 4, July 1979.

9. Sage, A. P.
Melsa, J. L. — *Estimation Theory with Applications to Communications and Control*, New York, McGraw-Hill, 1971.

10. Kailath, T. — *An Innovations Approach to Least Square Estimation, Part 1: Linear Filtering in Additive White Noise*, IEEE Trans. Automation Control, Vol. AC-13, No. 6, December 1968.

11. Makhoul, J. — *Linear Prediction: A Tutorial Review*, Proc. IEEE, Vol. 63, pp. 561-580, 1975.

12. Kailath, T. — *A View of Three Decades in Linear Filtering Theory*, IEEE Trans. Information Theory, Vol. IT-20, No. 2, March 1974.

13. Medich, J. S. — *A Survey of Data Smoothing for Linear and Nonlinear Dynamic Systems*, Automatica, Vol. 9, pp. 151-162, Pergamond Press, 1973.

14. Van Trees, H. L. — *Detection Estimation and Modulation: Part 1*, Wiley and Sons, New York, 1968.

15. Forney, G. D., Jr. — *Convolution Codes II. Maximum-Likelihood Decoding, and Convolution Codes III. Sequential Decoding*, Information Control, Vol. 25, pp. 222-247, 1974.

16. Viterbi, A. J.
Omura, J. K. — *Principles of Digital Communication and Coding*, New York, McGraw-Hill, 1979.

17. Gallager, R. G. — *A Simple Derivation of the Coding Theorem and Some Applications*, IEEE Trans. Information Theory, Vol. IT-11, pp. 3-18, January 1965.

18. Apostol, A. M. — *Mathematical Analysis*, Addison-Wesley, USA, 1958.

19. Ash, R. B. — *Real Analysis and Probability*, Academic Press, New York, 1972.

20. Ferguson, S. T. — *Mathematical Statistics - A Decision Theoretic Approach*, Academic Press, New York, 1967.

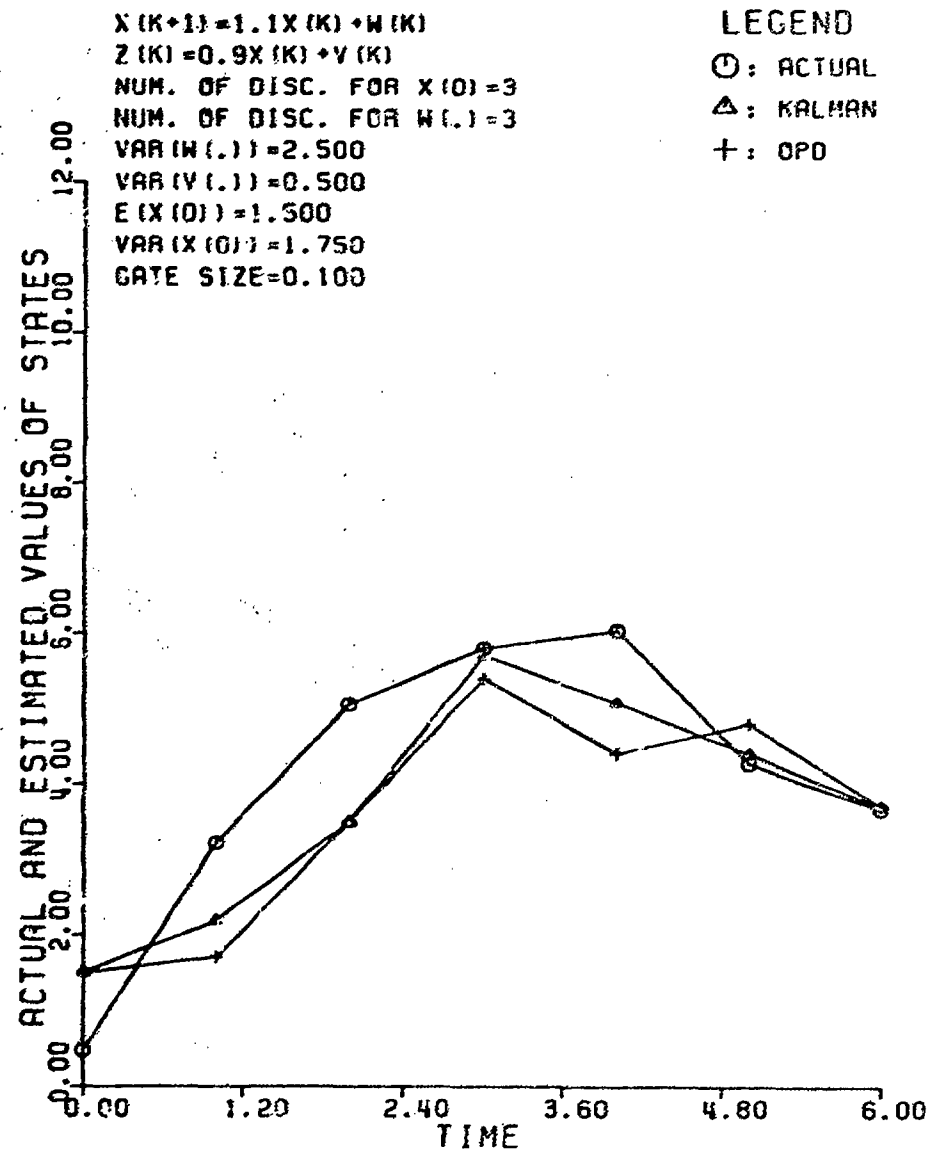


Fig.4.1.1(a) Actual and estimated values of states

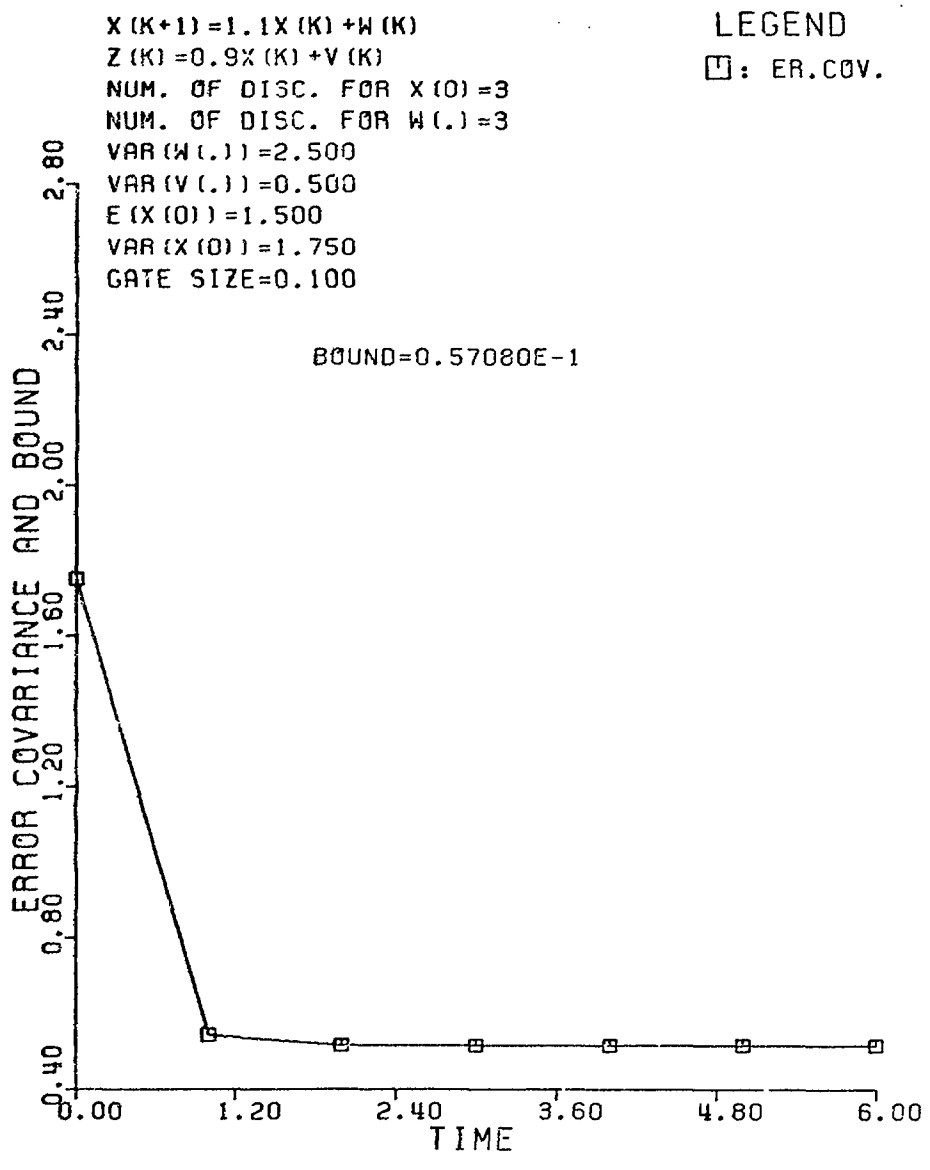


Figure 4.1.1(b) Error covariance and bound

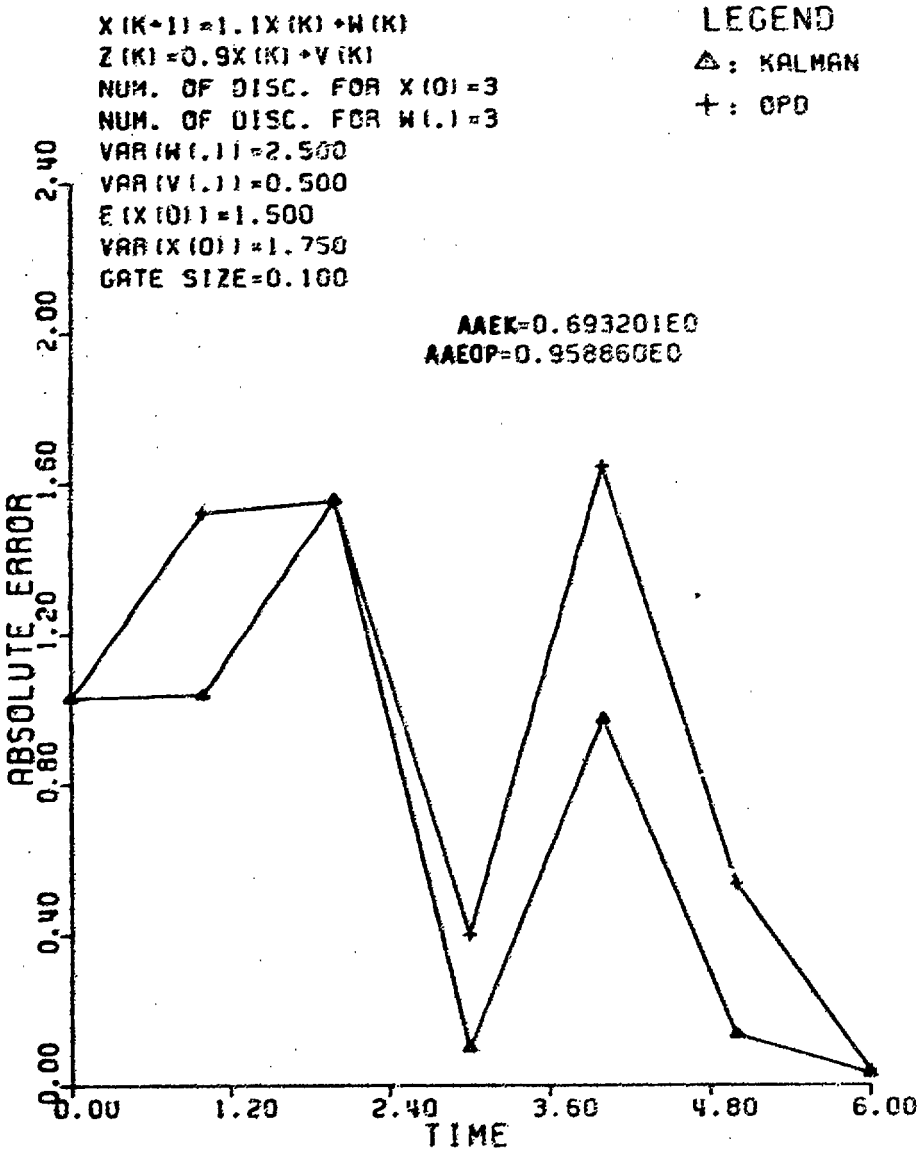


Figure 4.1.1(c) Absolute and average absolute errors

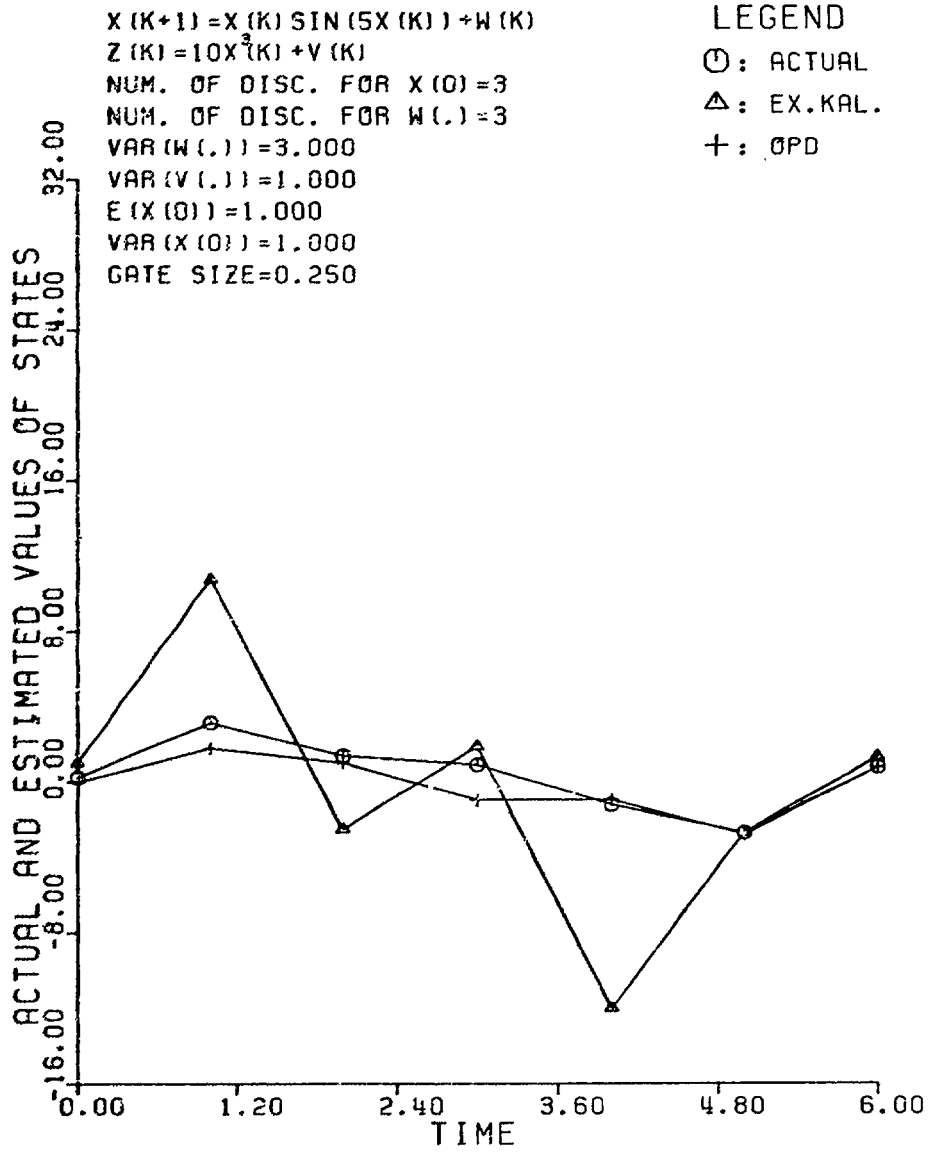


Figure 4.1.2(a) Actual and estimated values of states

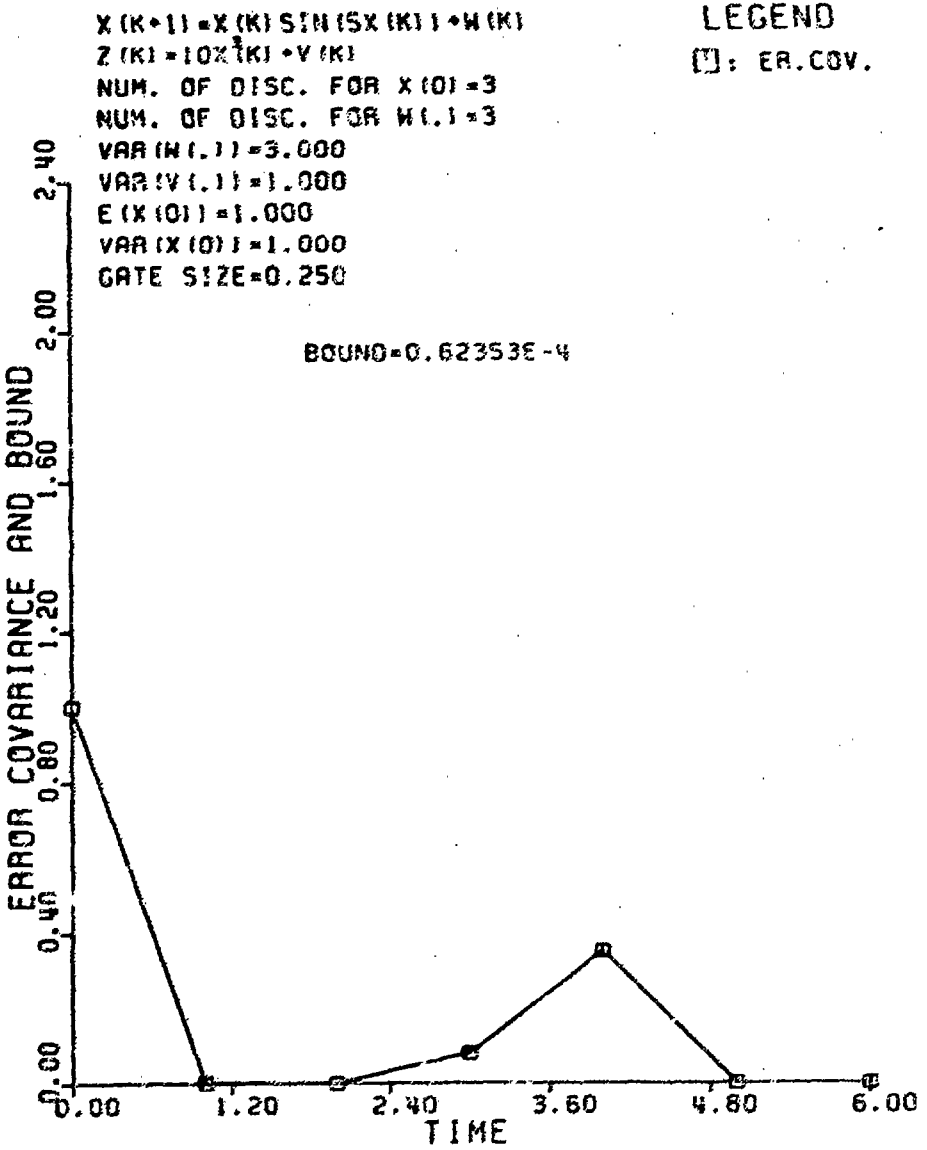


Figure 4.1.2(b) Error covariance and bound

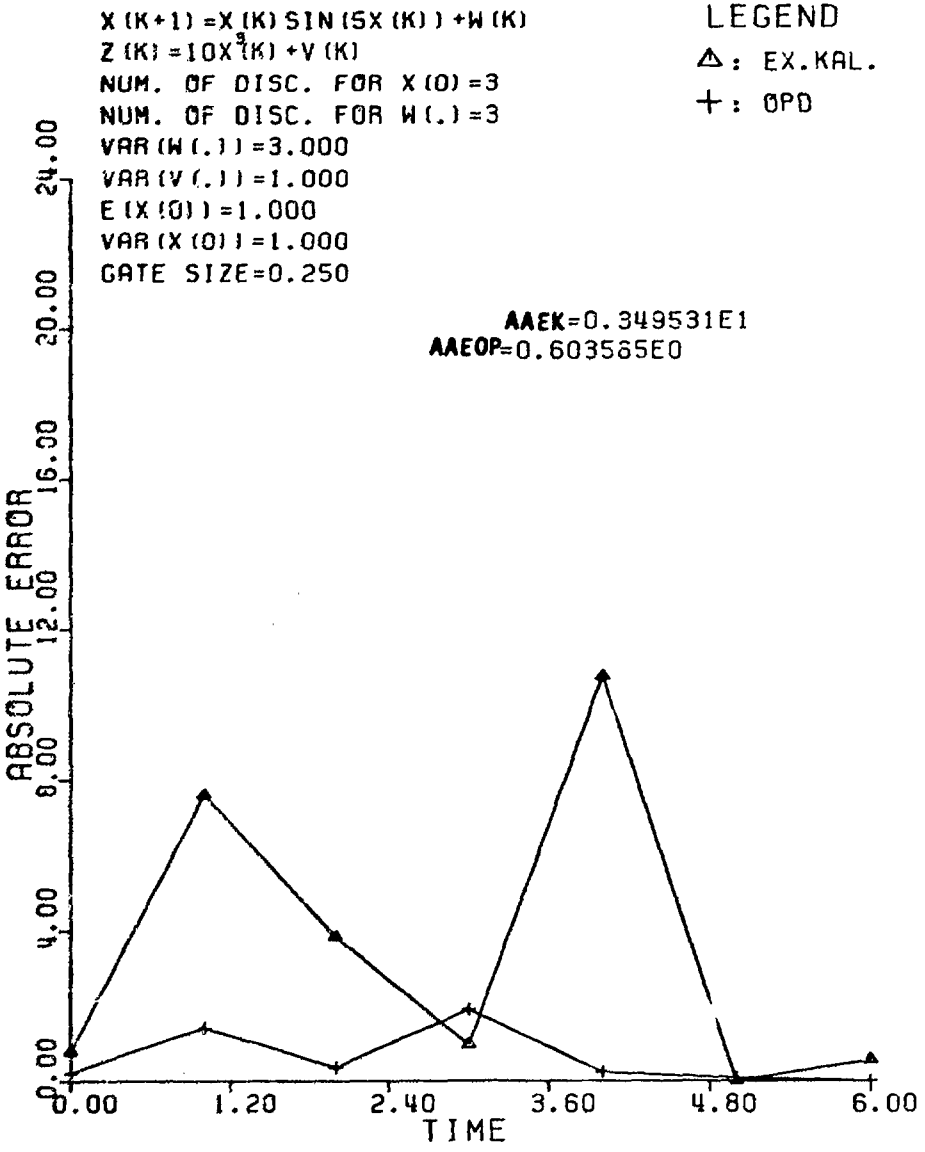


Figure 4.1.2(c) Absolute and average absolute errors

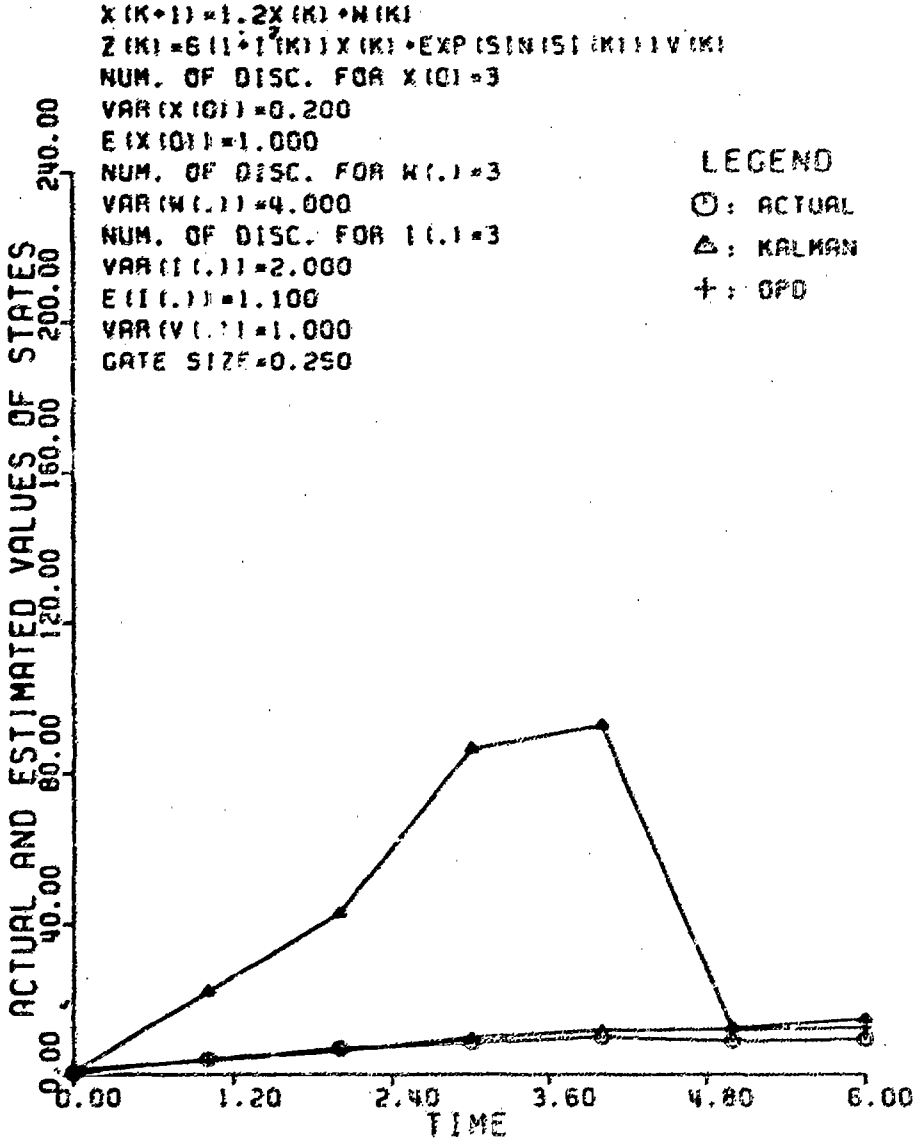


Figure 4.1.3(a) Actual and estimated values of states

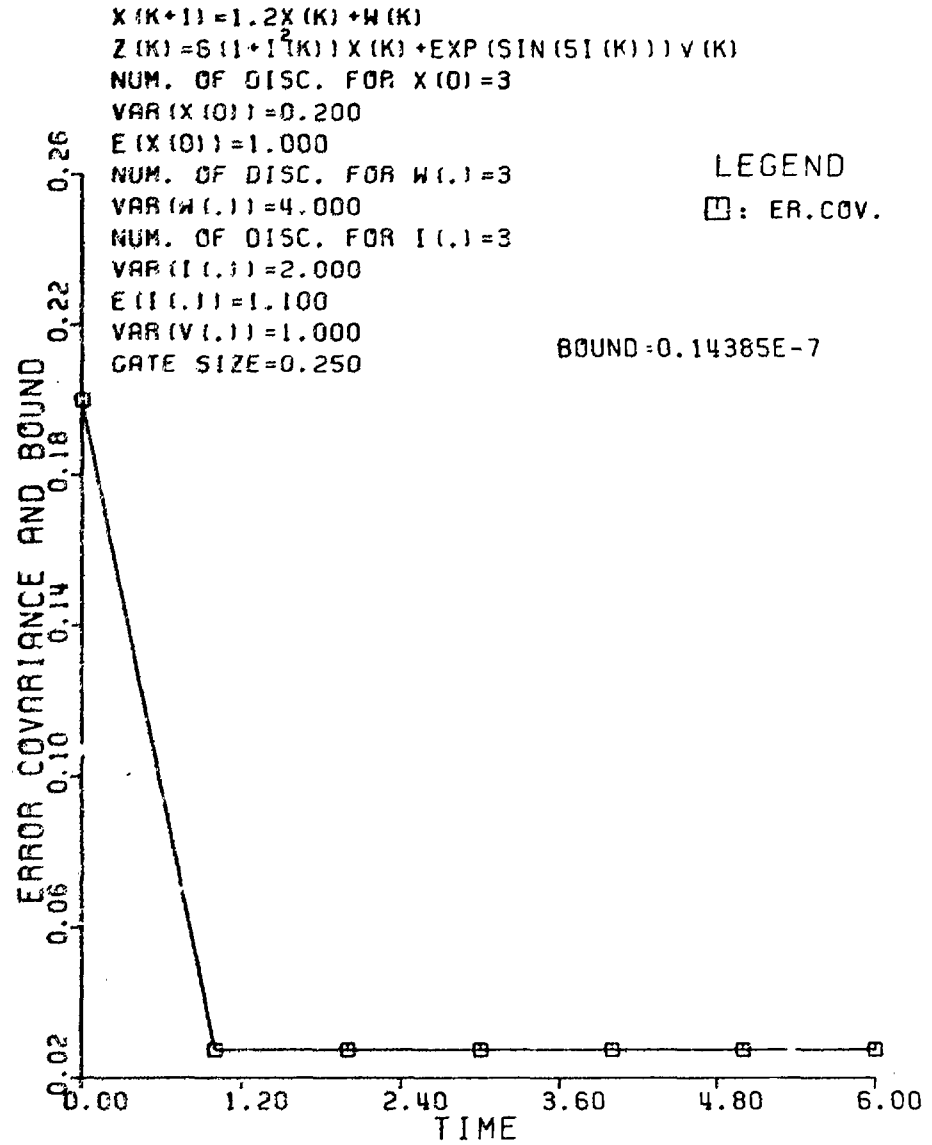


Figure 4.1.3(b) Error covariance and bound

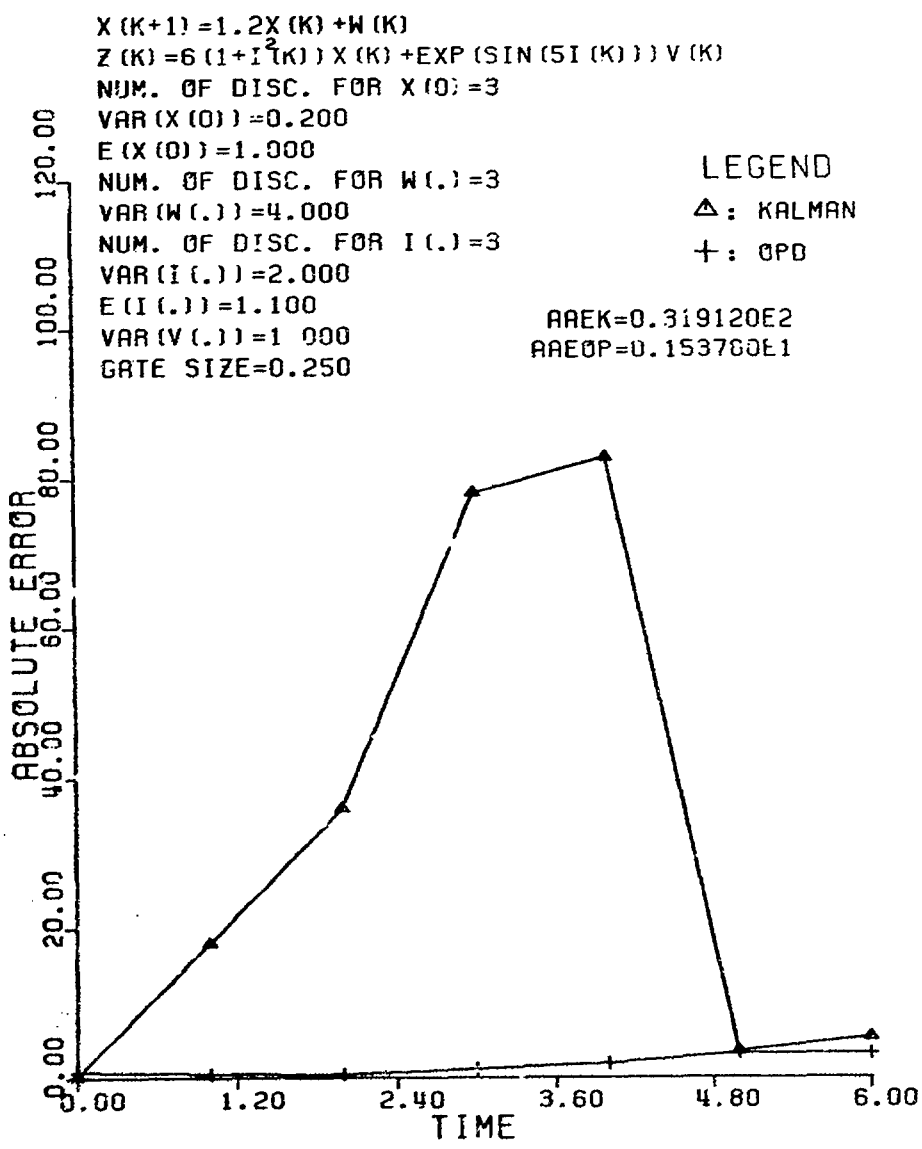


Figure 4.1.3(c) Absolute and average absolute errors

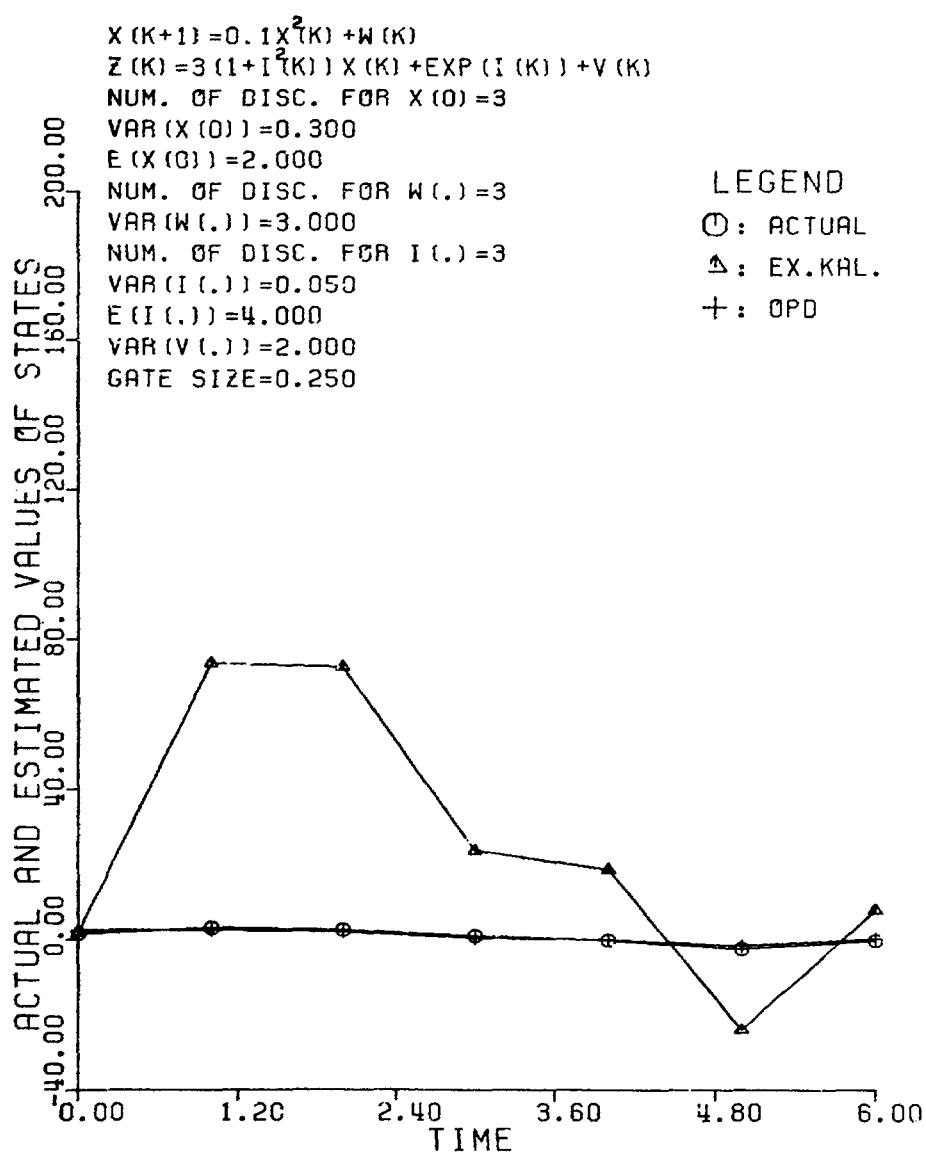


Figure 4.1.4(a) Actual and estimated values of states

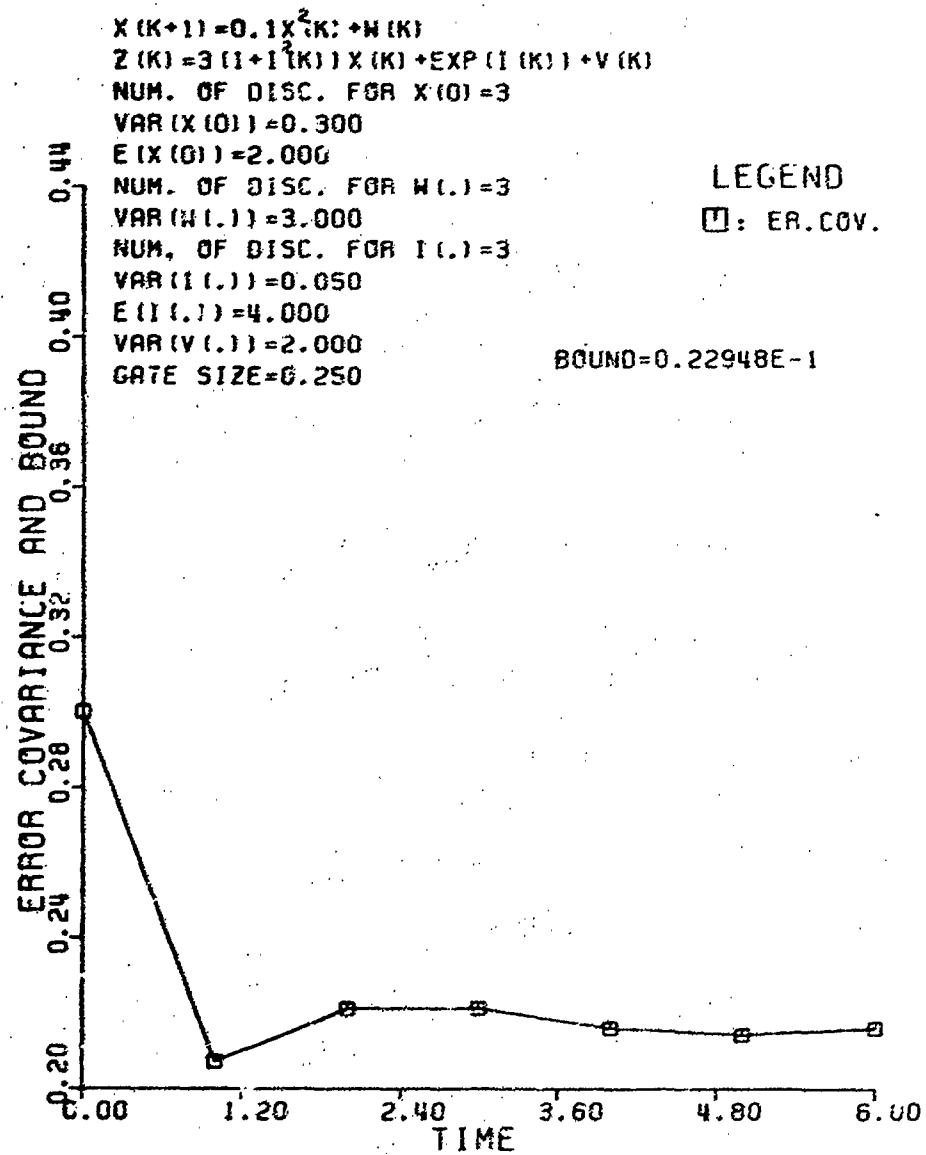


Figure 4.1.4(b) Error covariance and bound

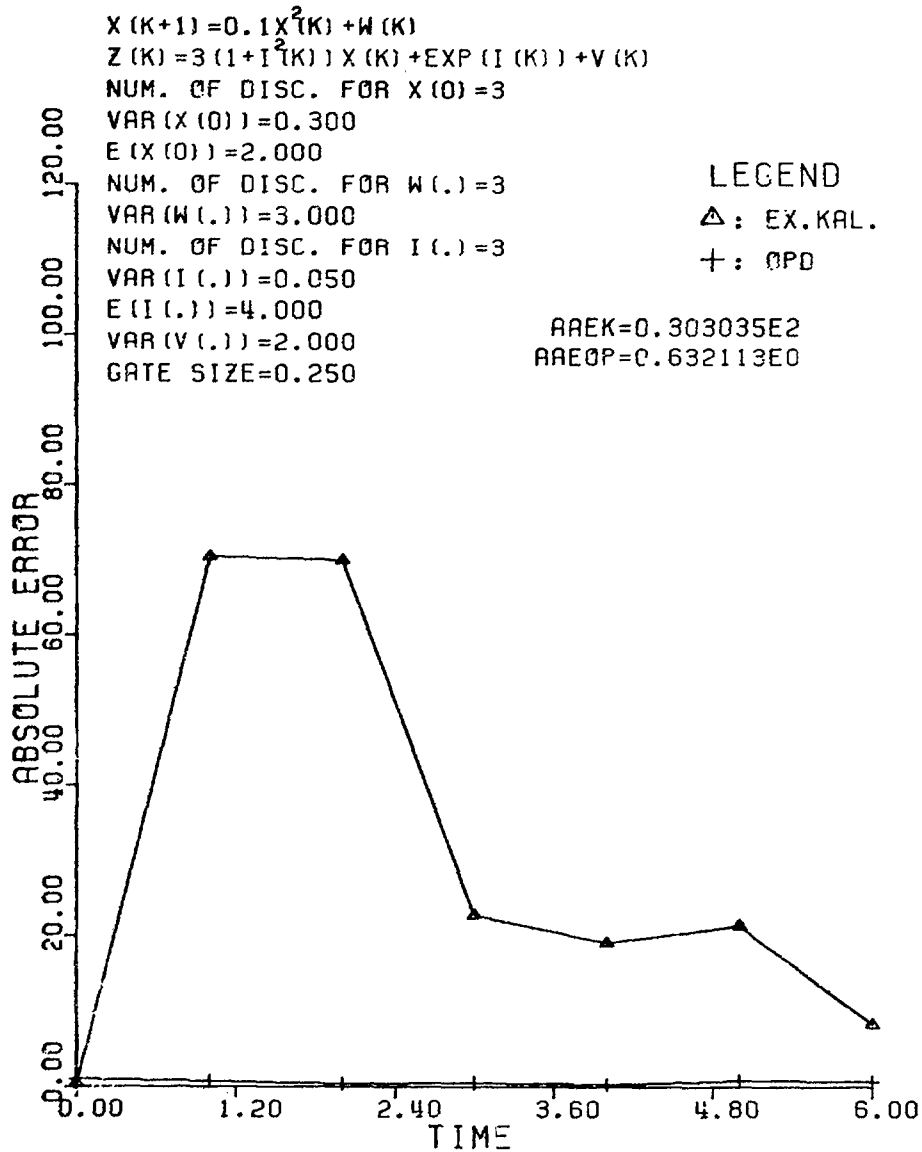


Figure 4.1.4(c) Absolute and average absolute errors

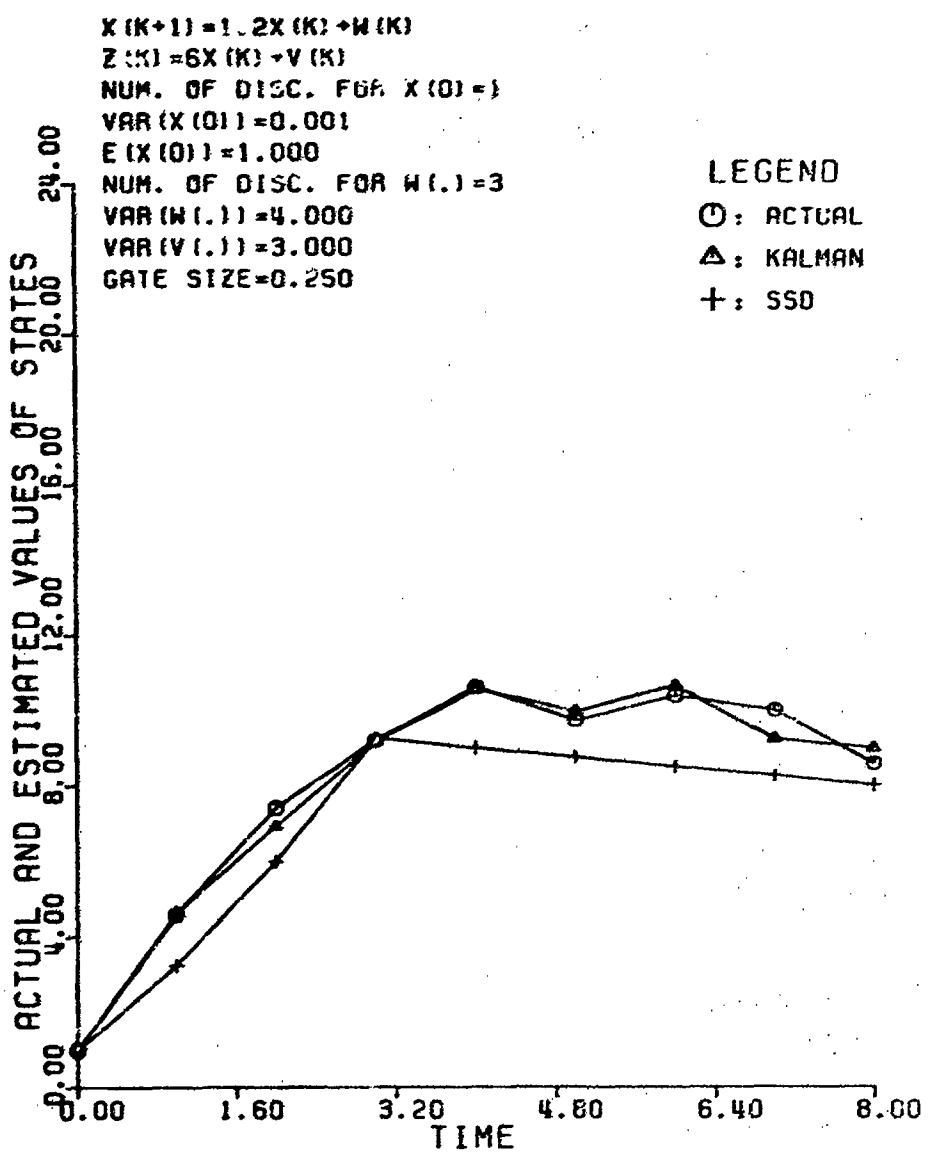


Figure 4.2.1(a) Actual and estimated values of states

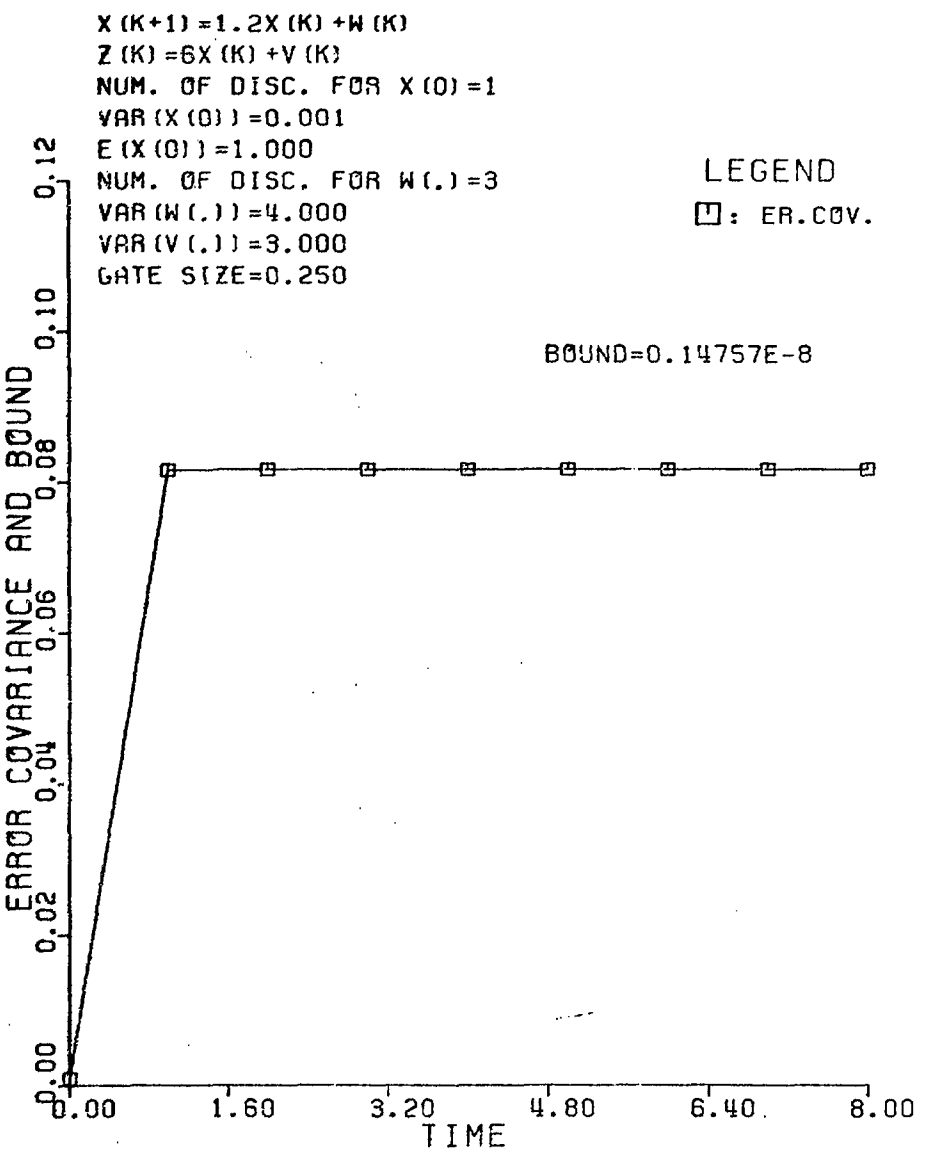


Figure 4.2.1(b) Error covariance and bound

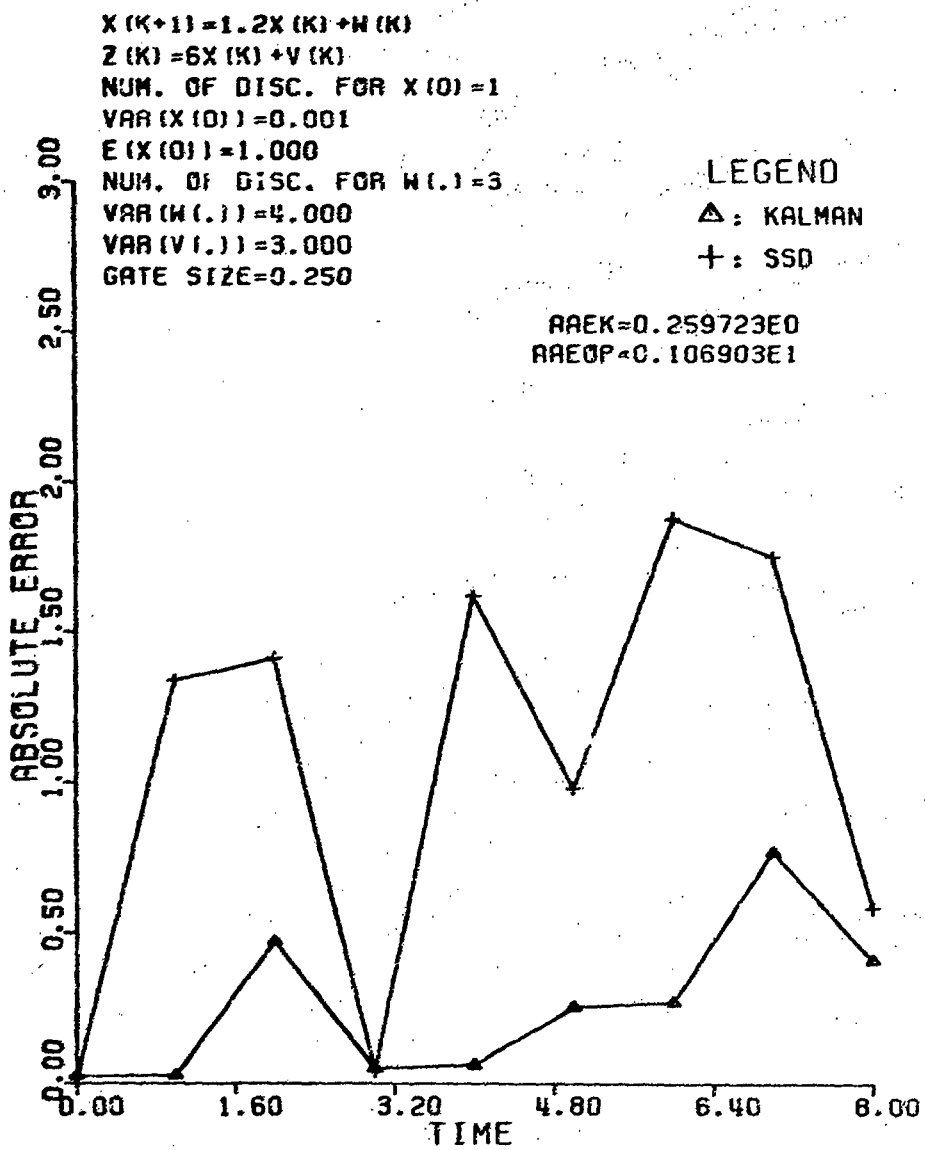


Figure 4.2.1(c) Absolute and average absolute errors

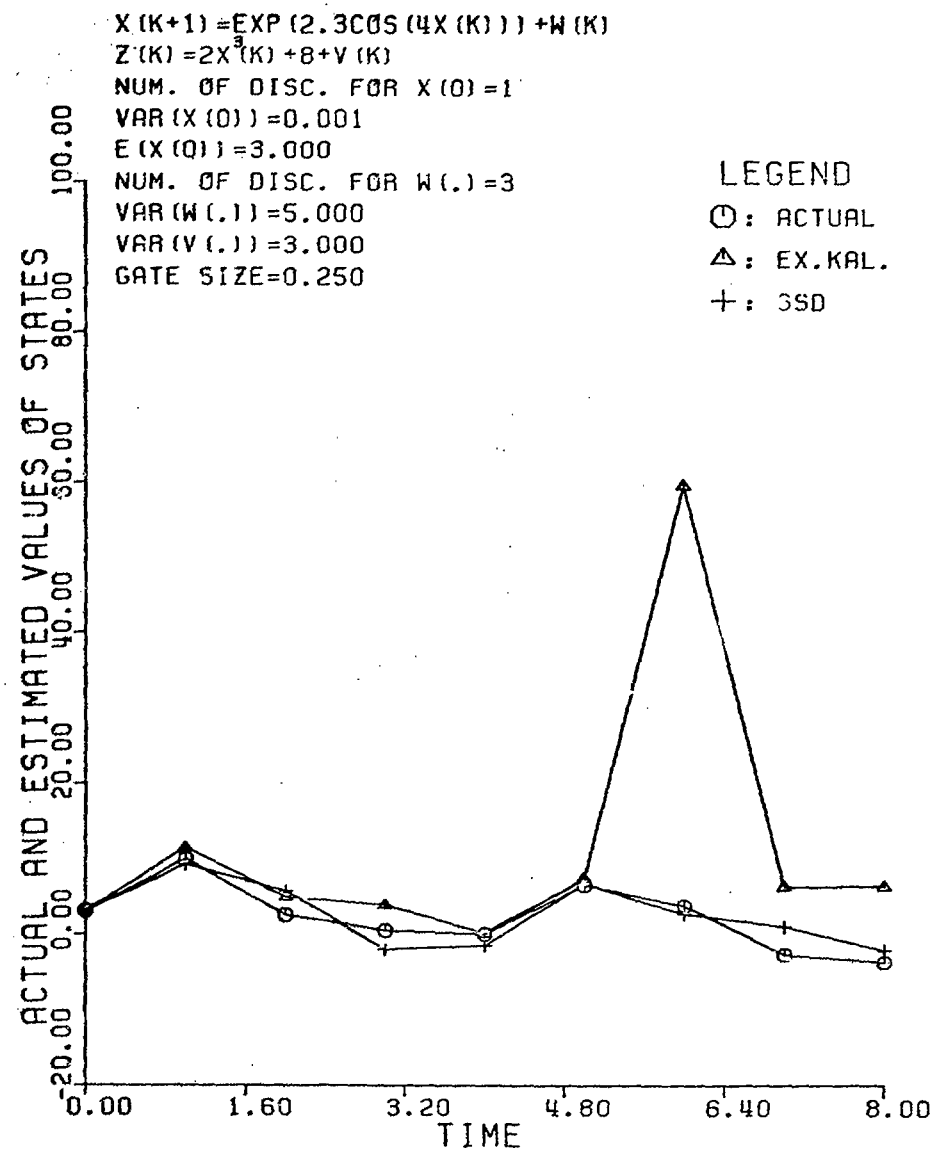


Figure 4.2.2(a) Actual and estimated values of states

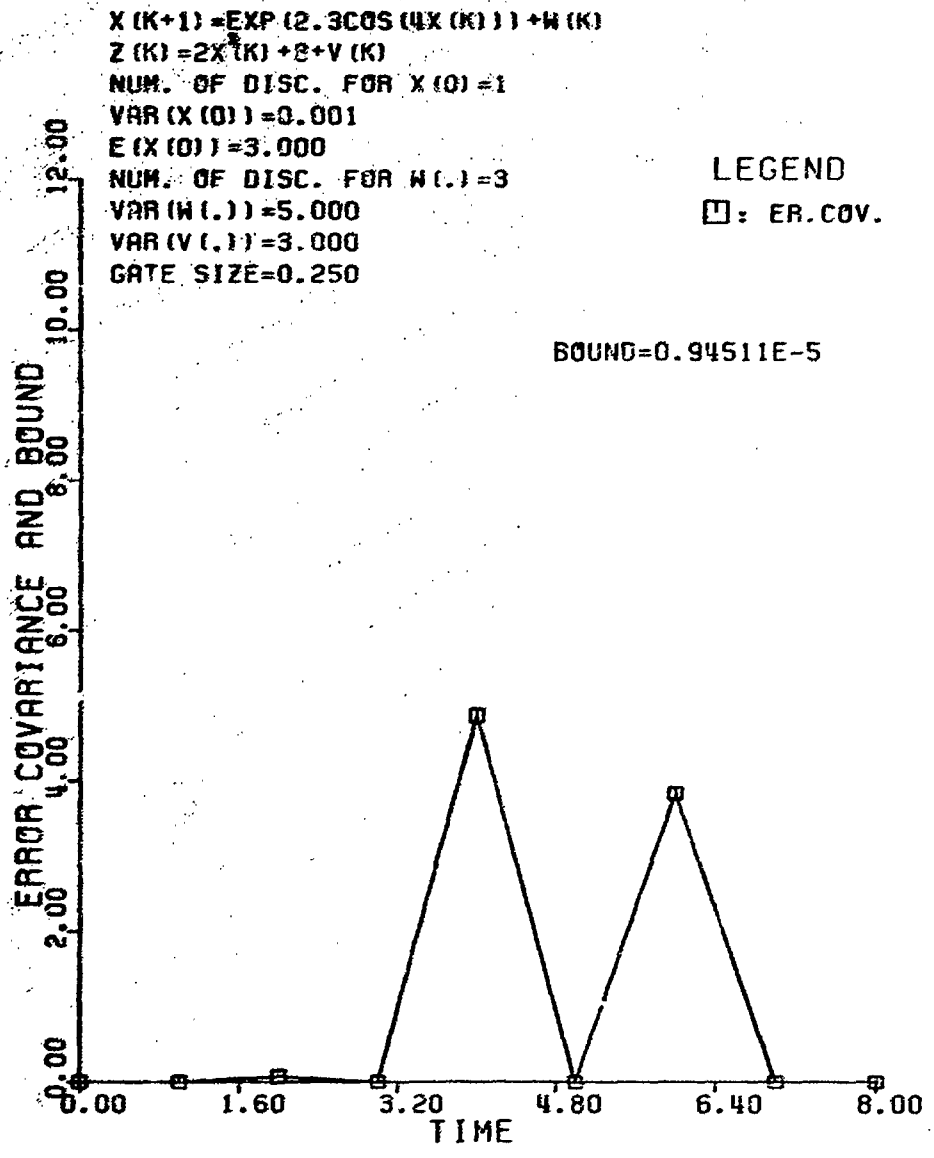


Figure 4.2.2(b) Error covariance and bound

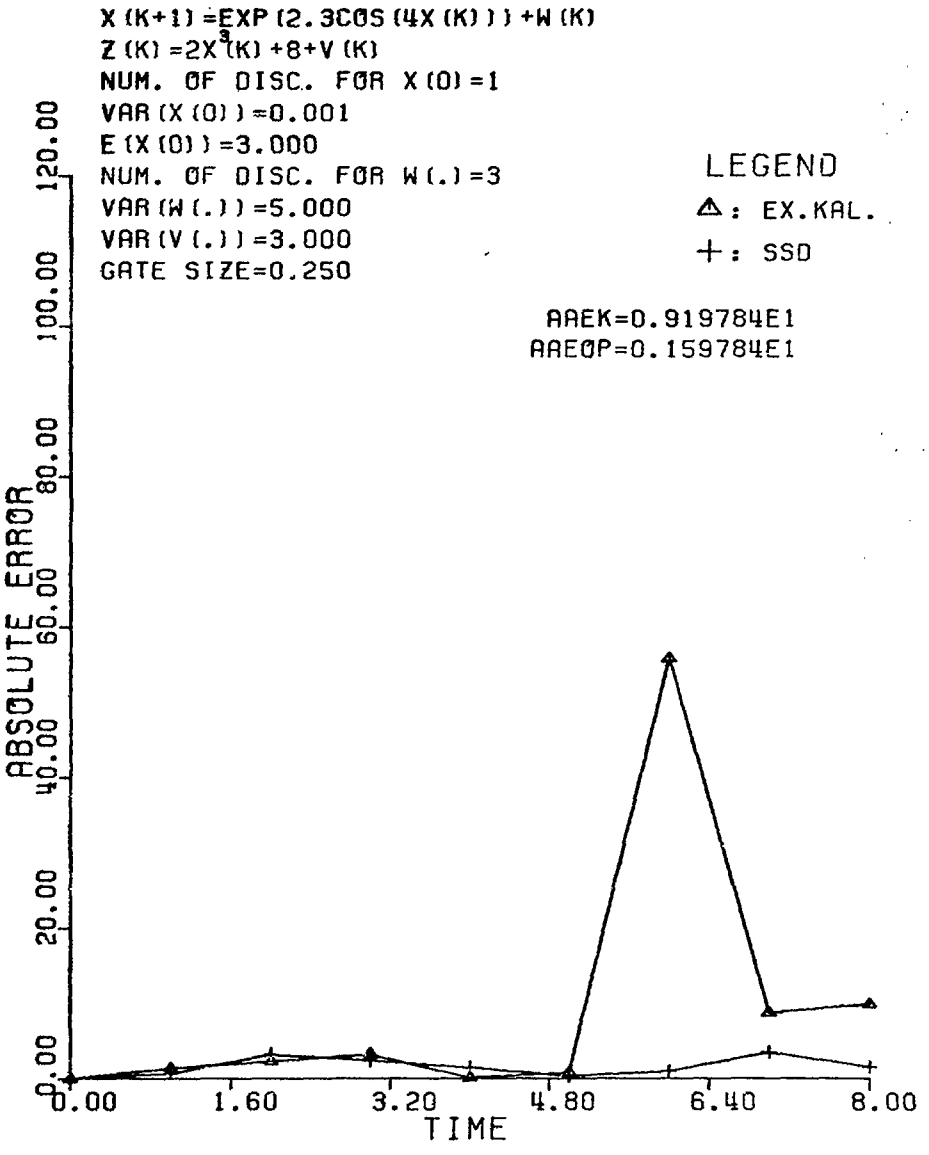


Figure 4.2.2(c) Absolute and average absolute errors

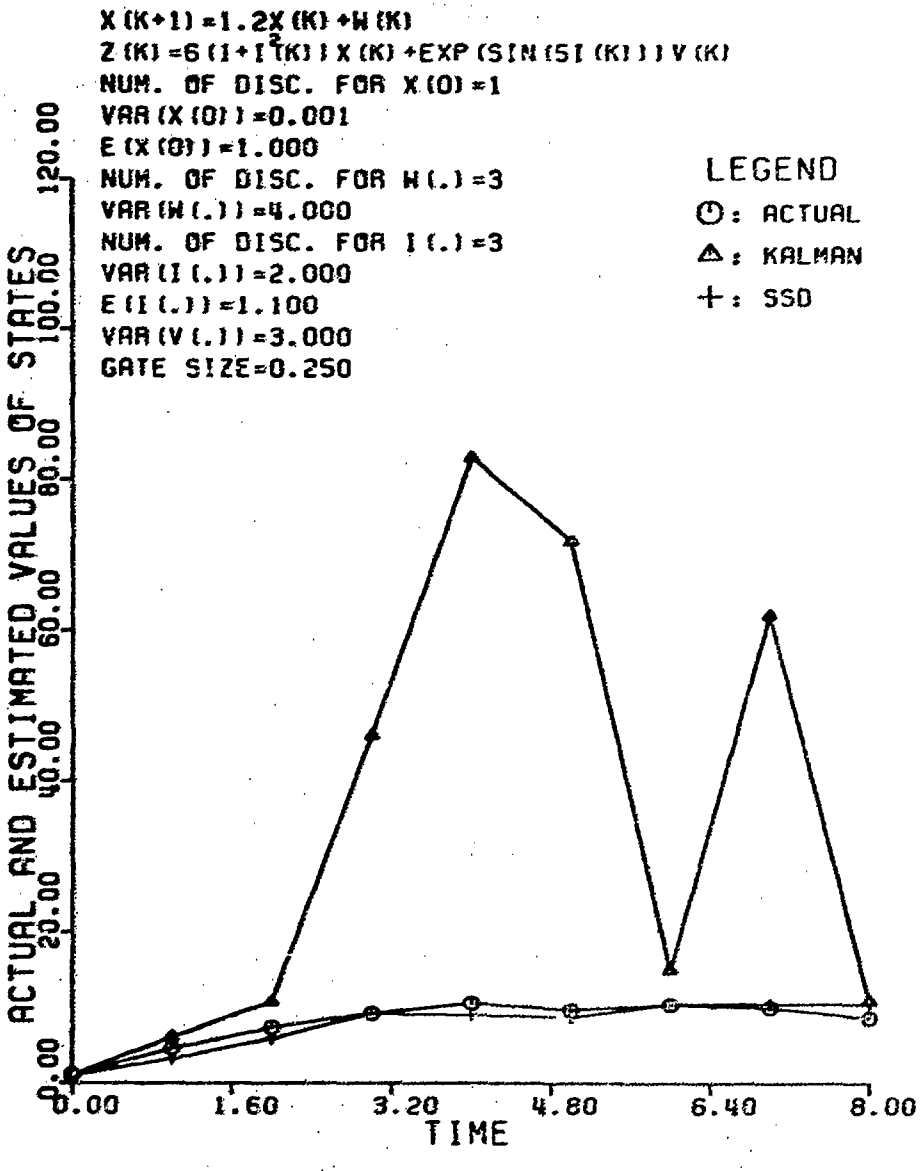


Figure 4.2.3(a) Actual and estimated values of states

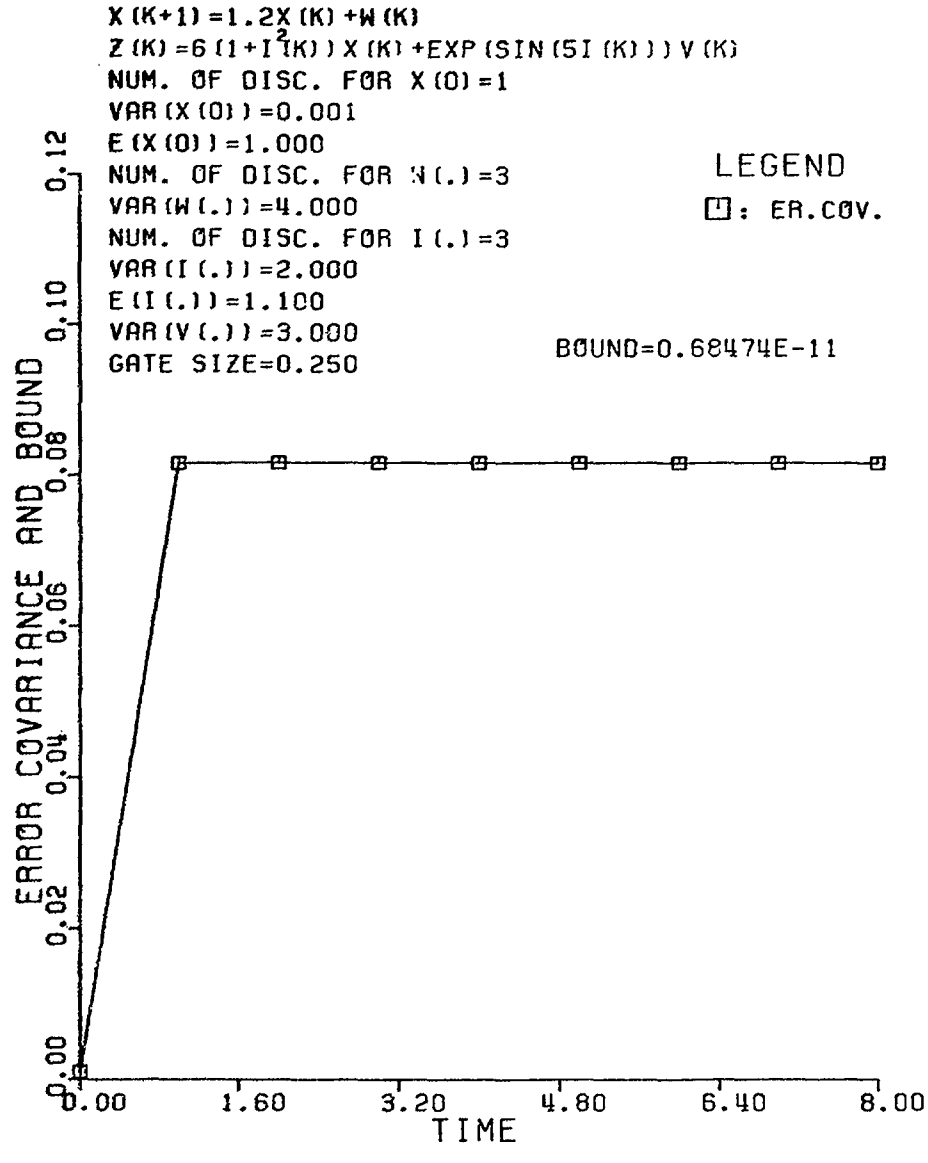


Figure 4.2.3(b) Error covariance and bound

19-58

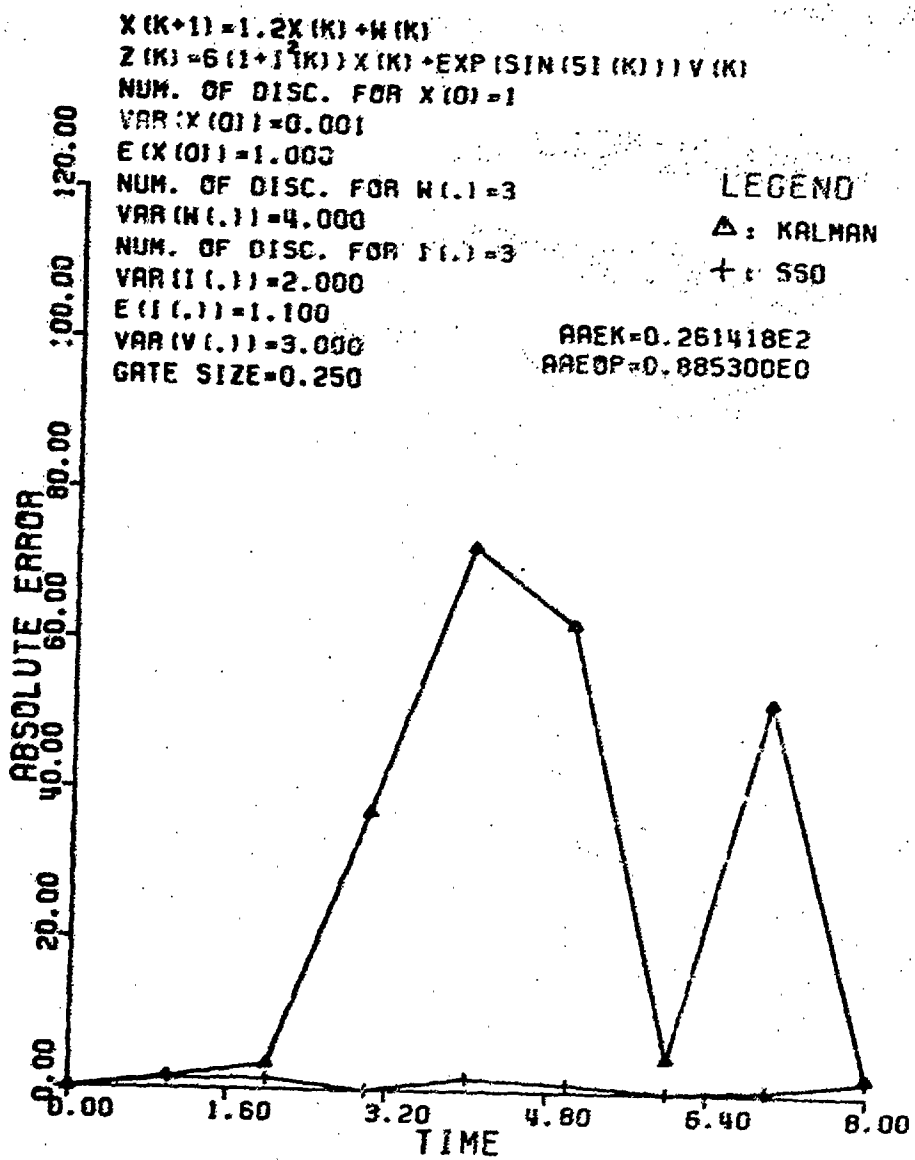


Figure 4.2.3(c) Absolute and average absolute errors

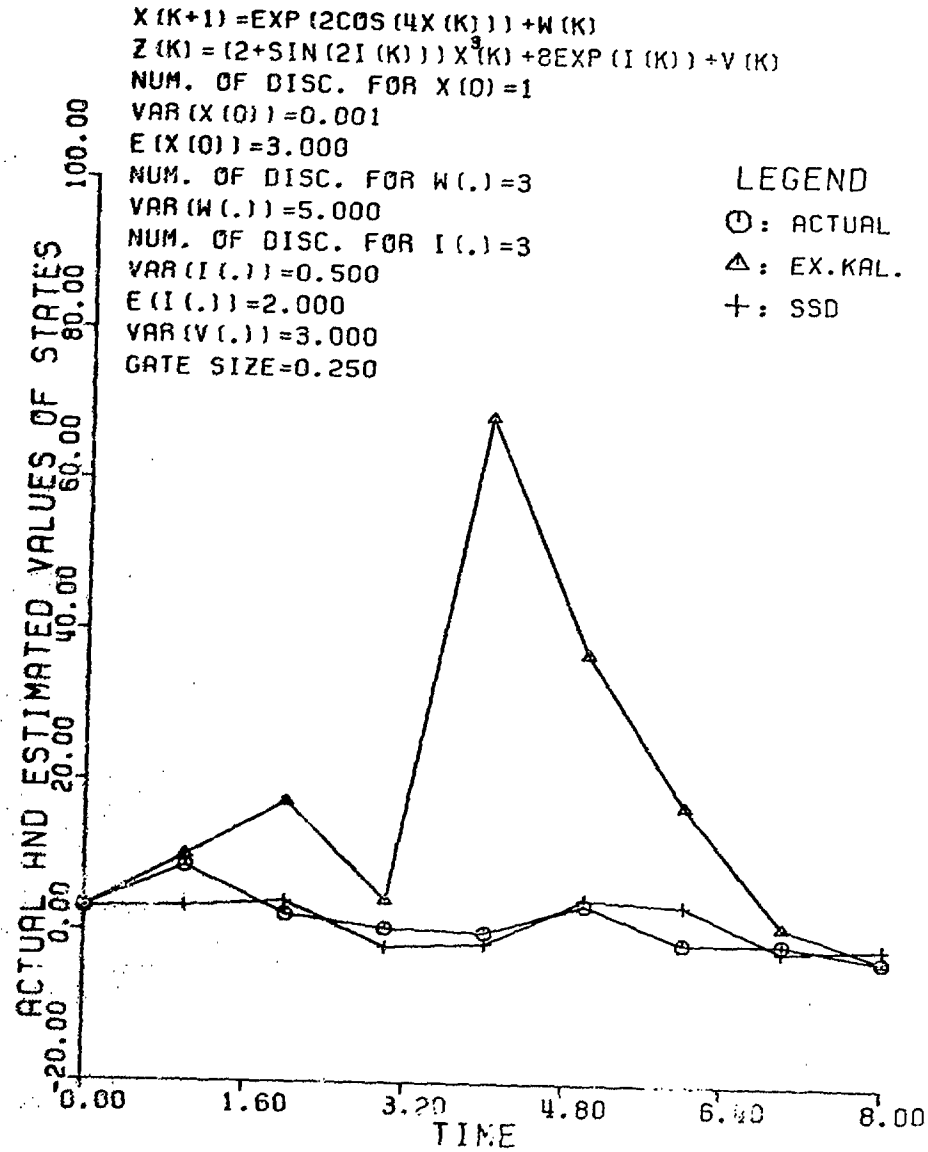


Figure 4.2.4(a) Actual and estimated values of states

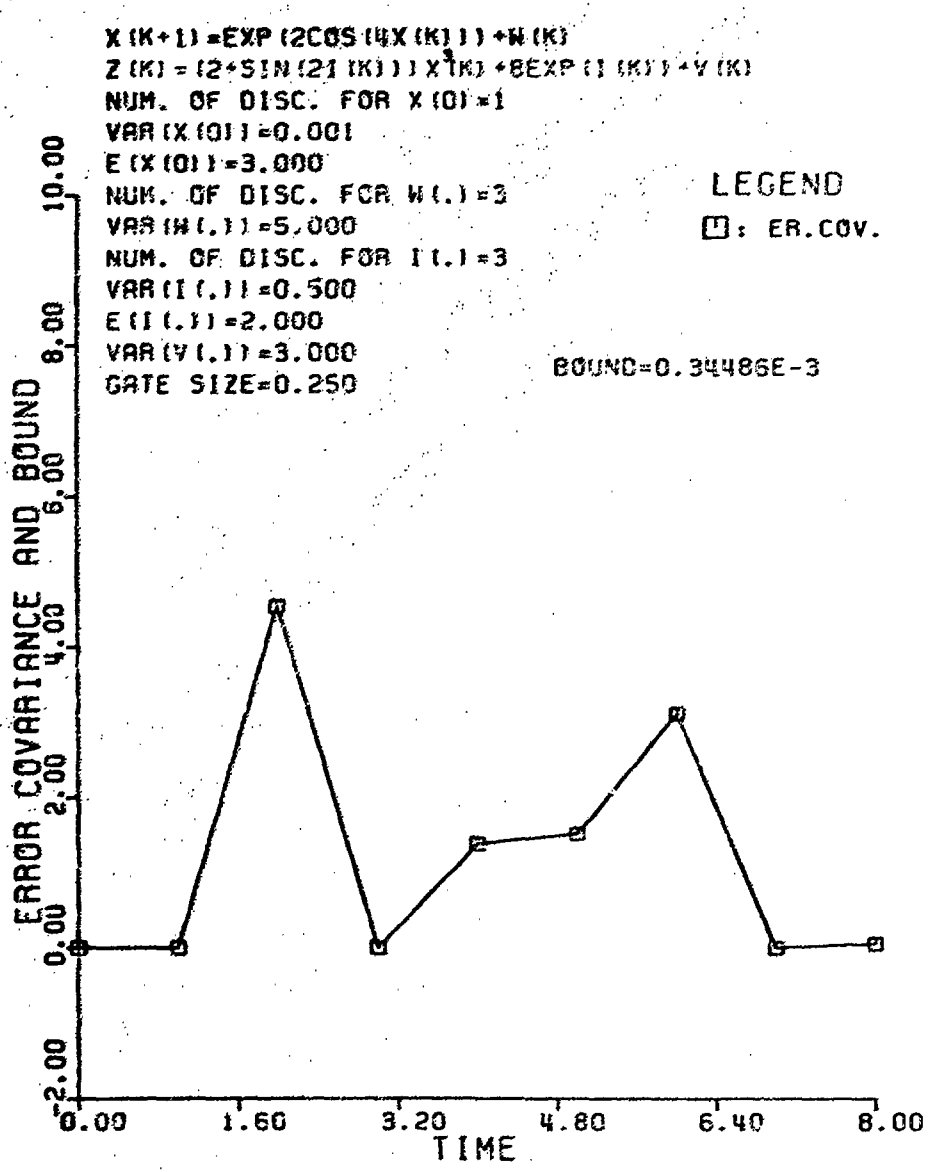


Figure 4.2.4(b) Error covariance and bound

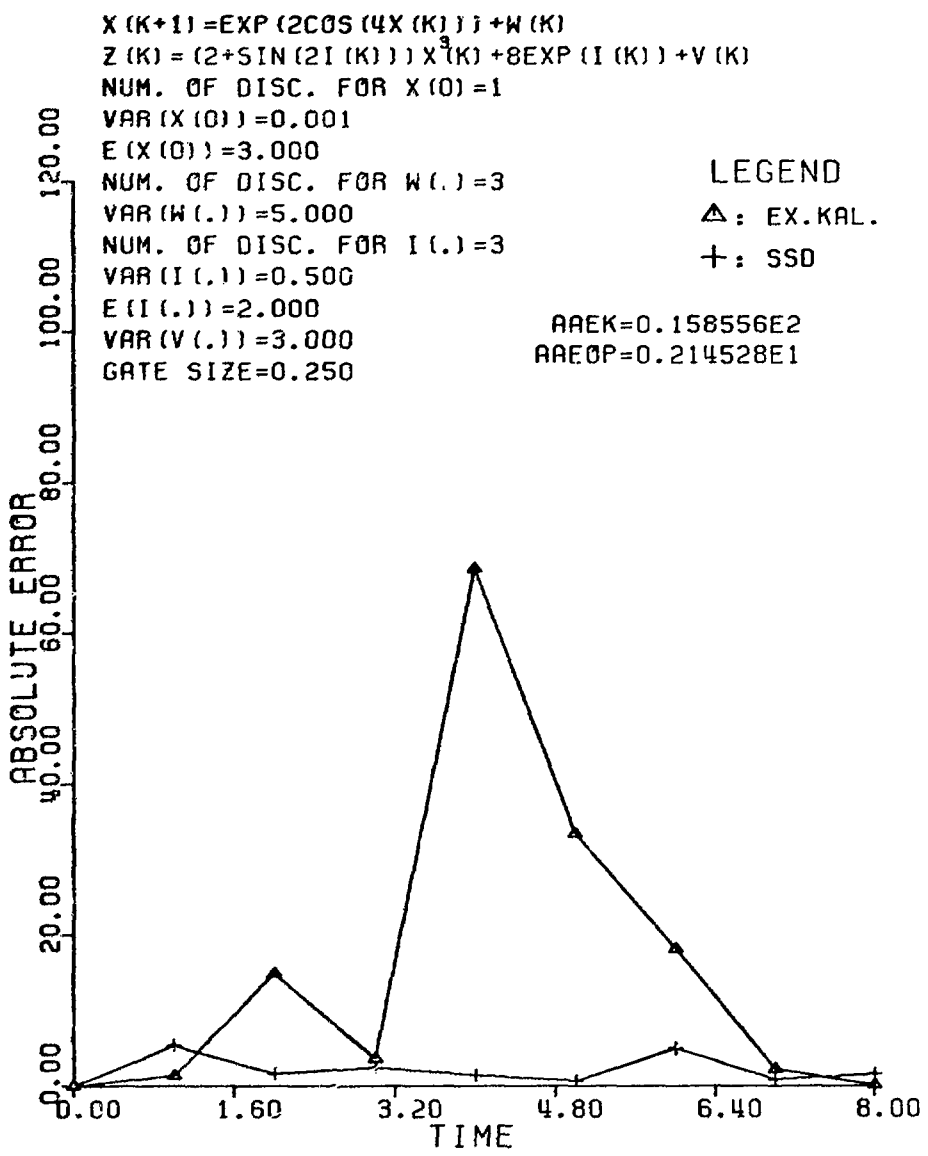


Figure 4.2.4(c) Absolute and average absolute errors

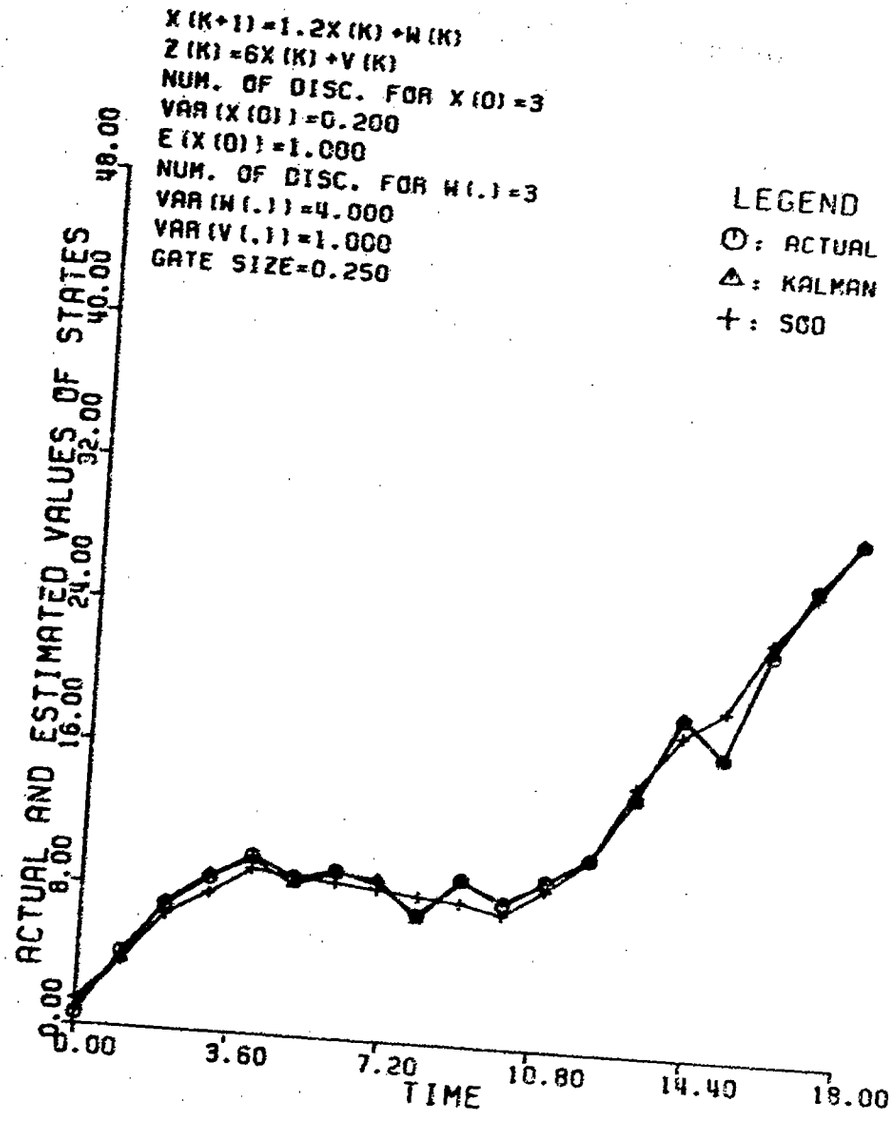


Figure 4.3.1(a) Actual and estimated values of states

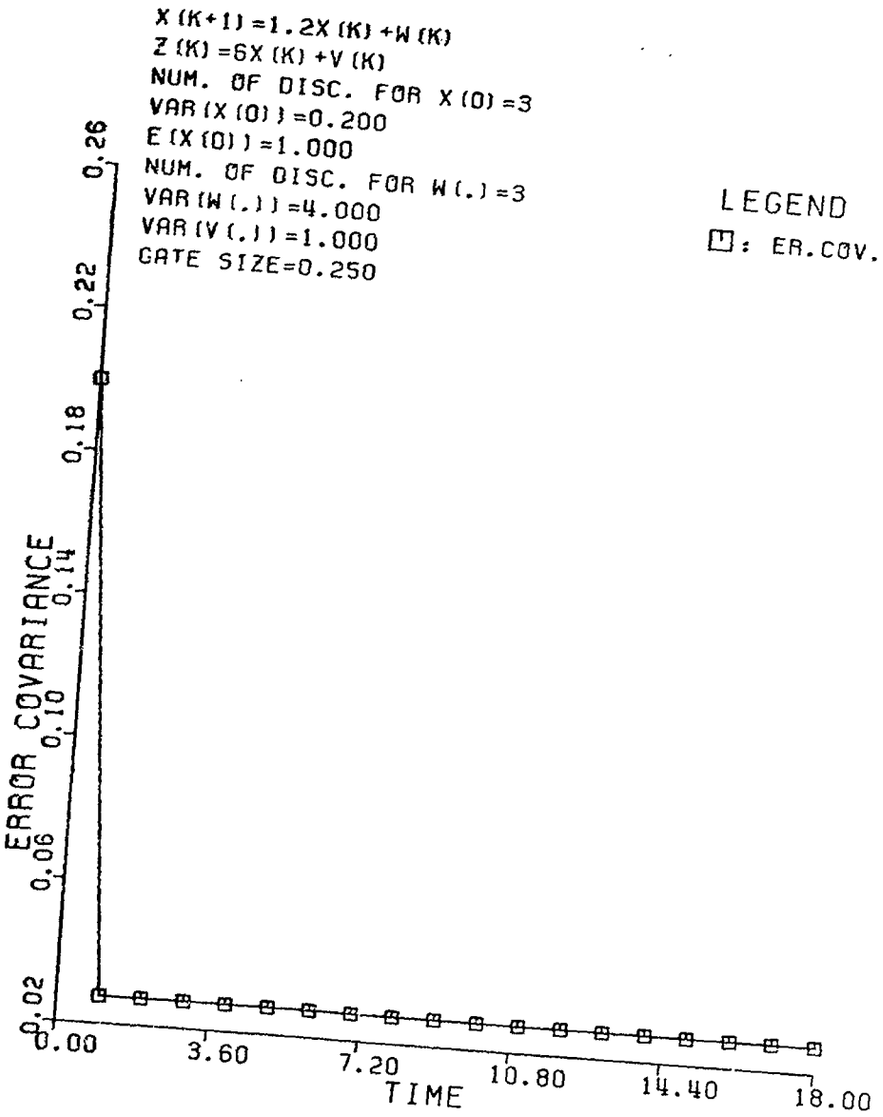


Figure 4.3.1(b) Error covariance

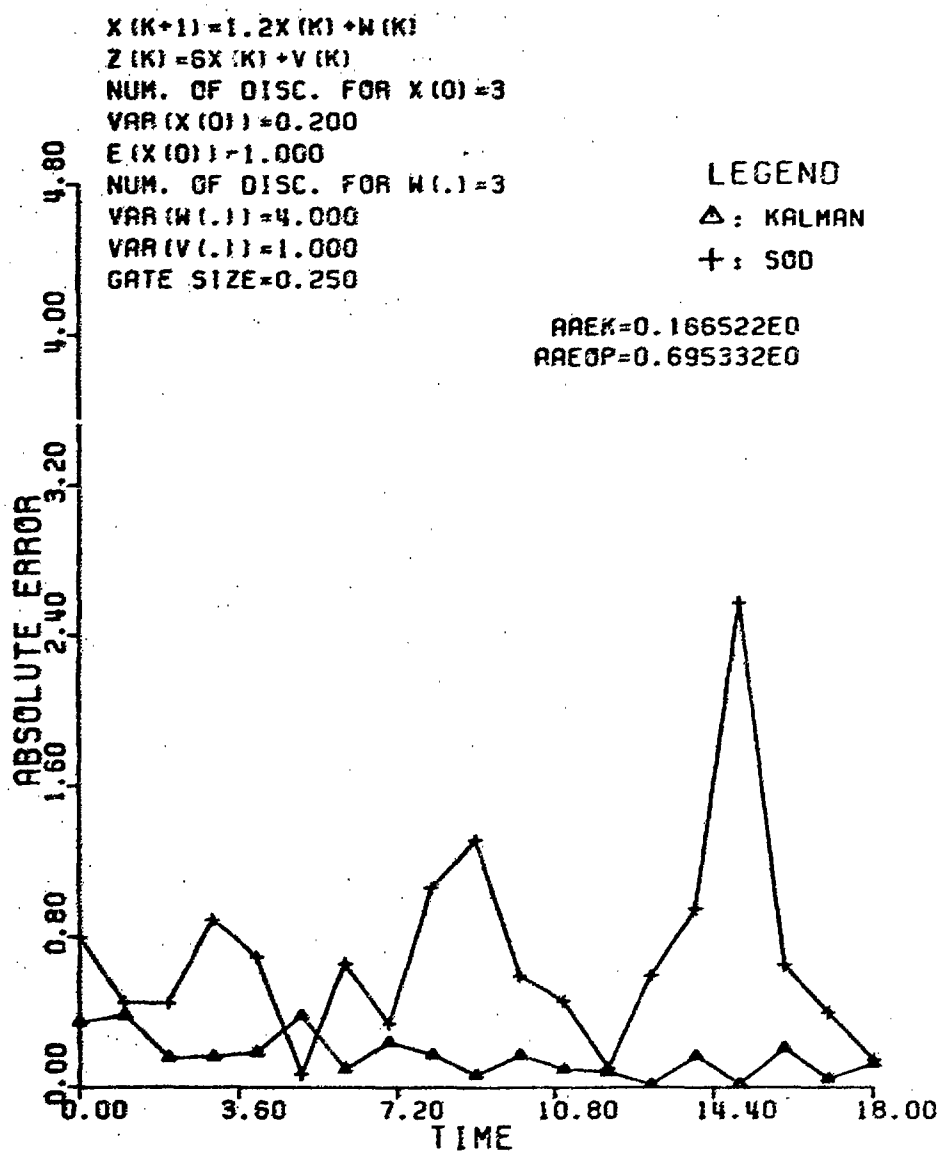


Figure 4.3.1(c) Absolute and average absolute errors

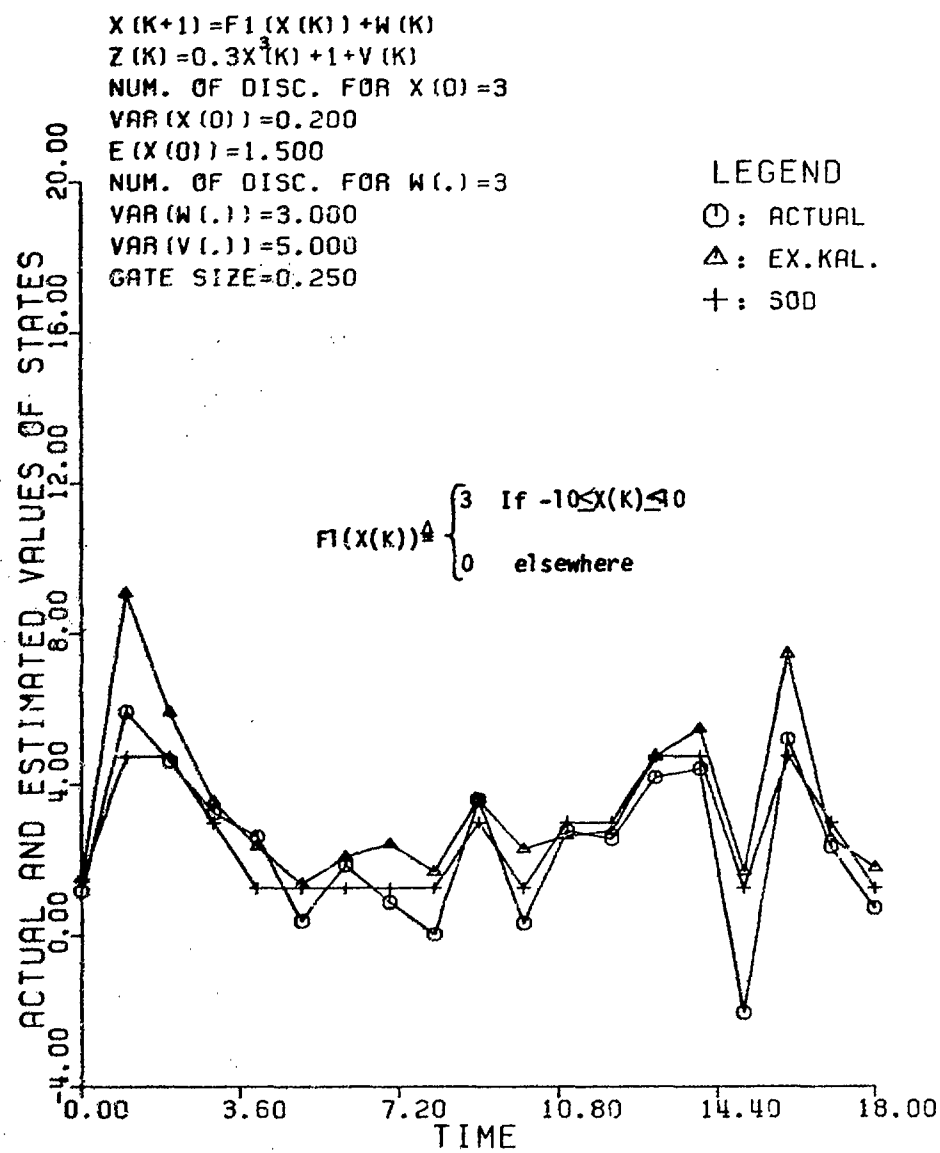


Figure 4.3.2(a) Actual estimated values of states

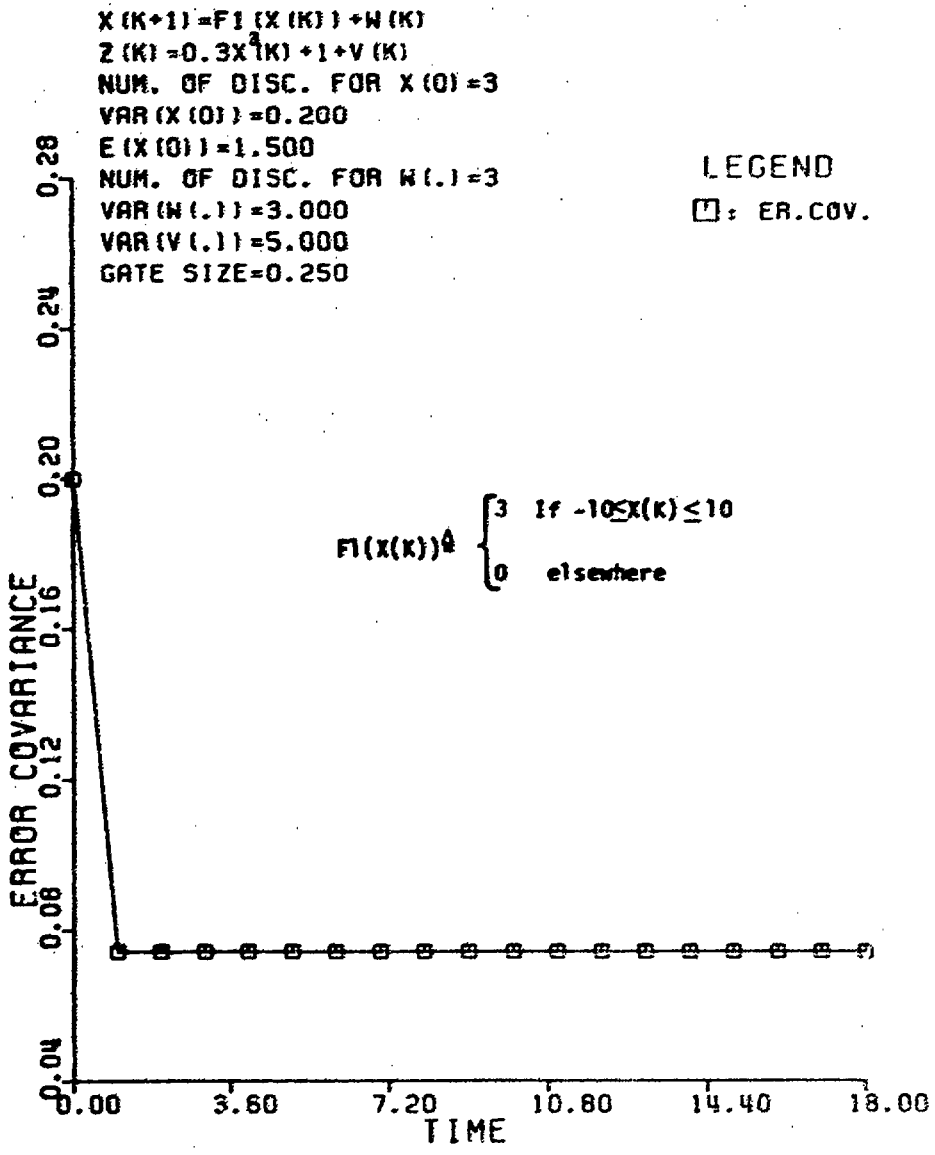


Figure 4.3.2(b) Error covariance

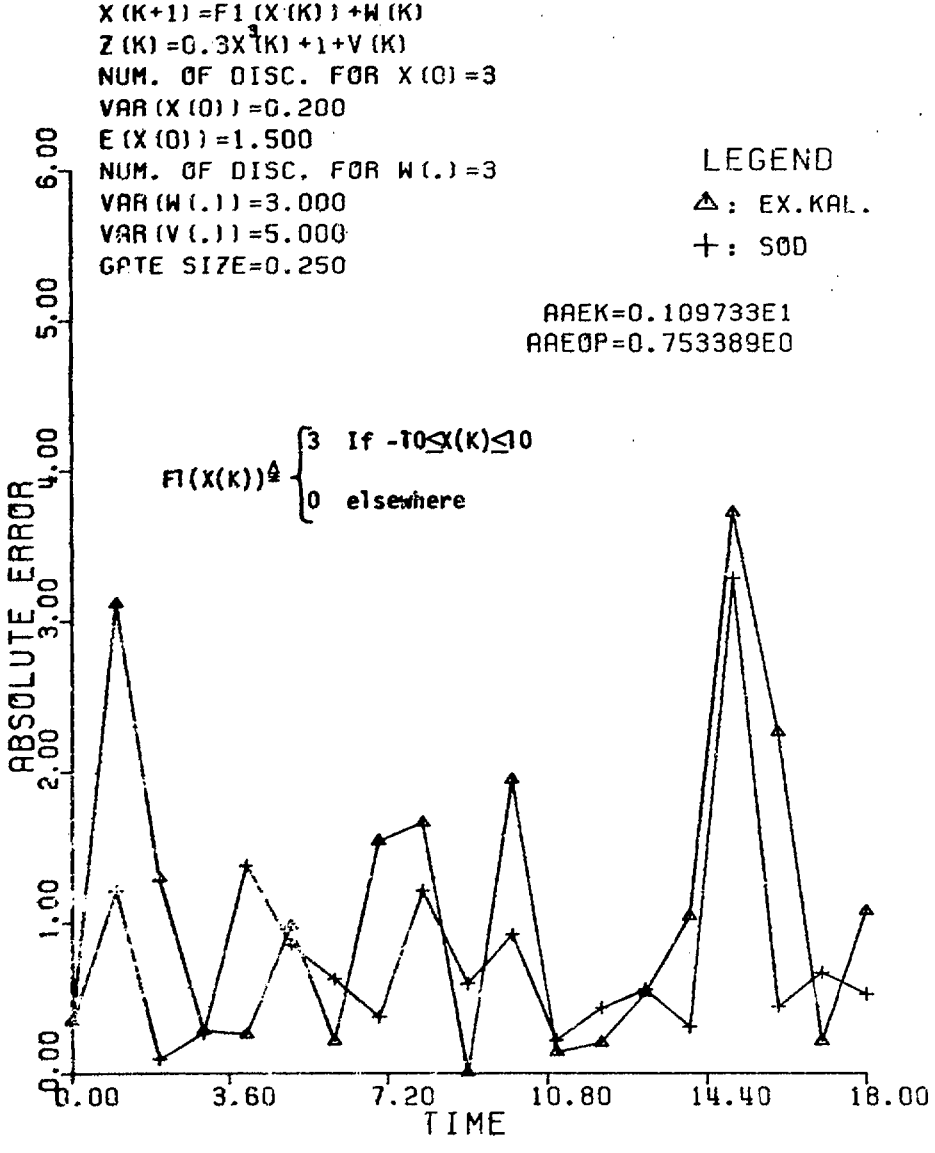


Figure 4.3.2(c) Absolute and average absolute errors

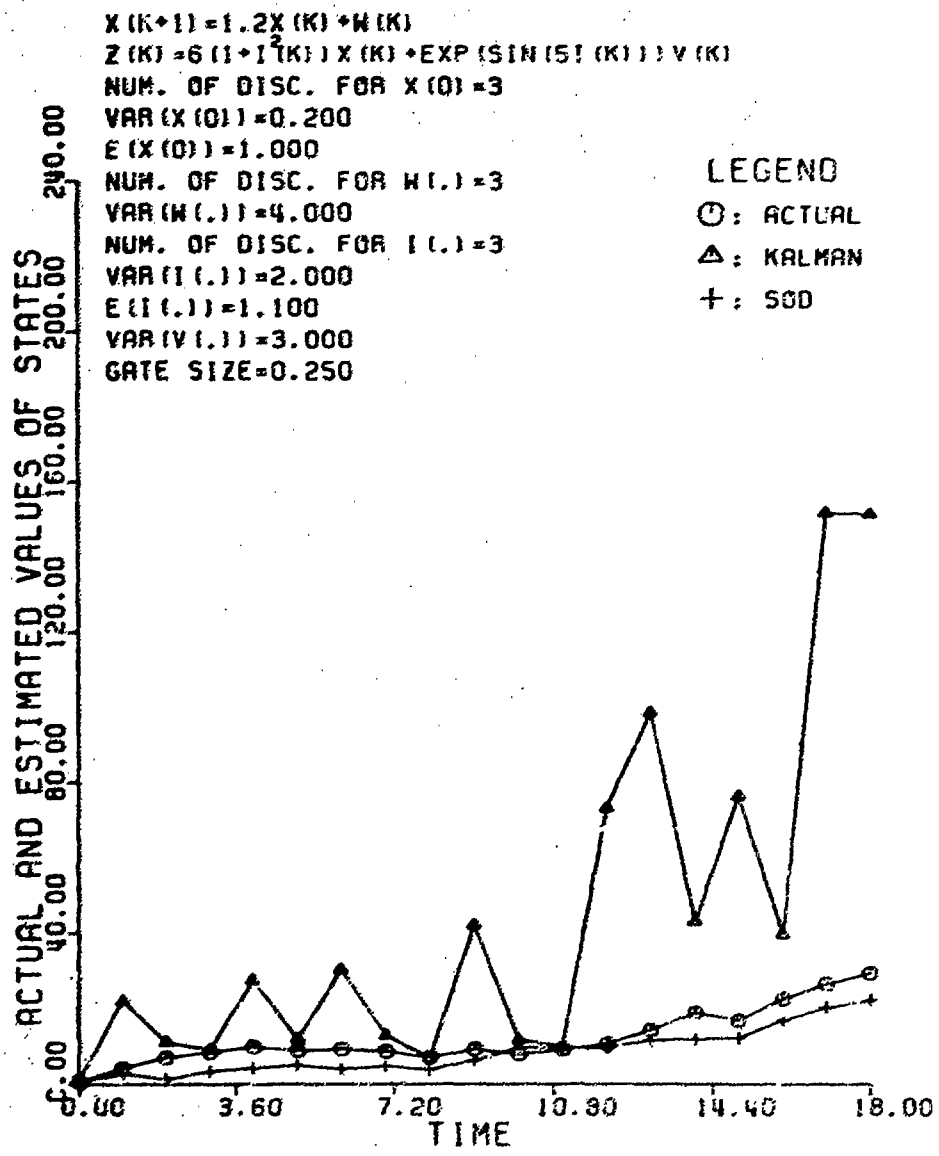


Figure 4.3.3(a) Actual and estimated values of states

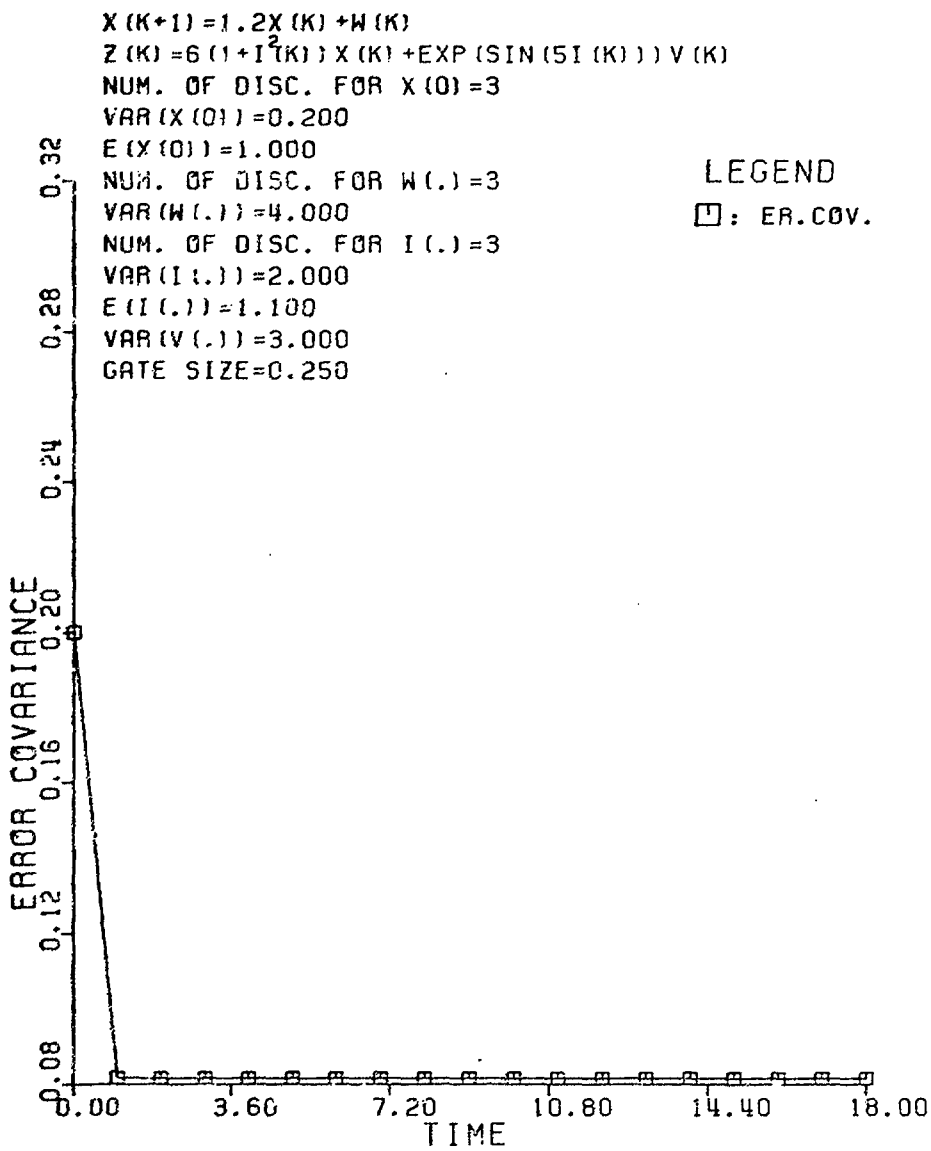


Figure 4.3.3(b) Error covariance

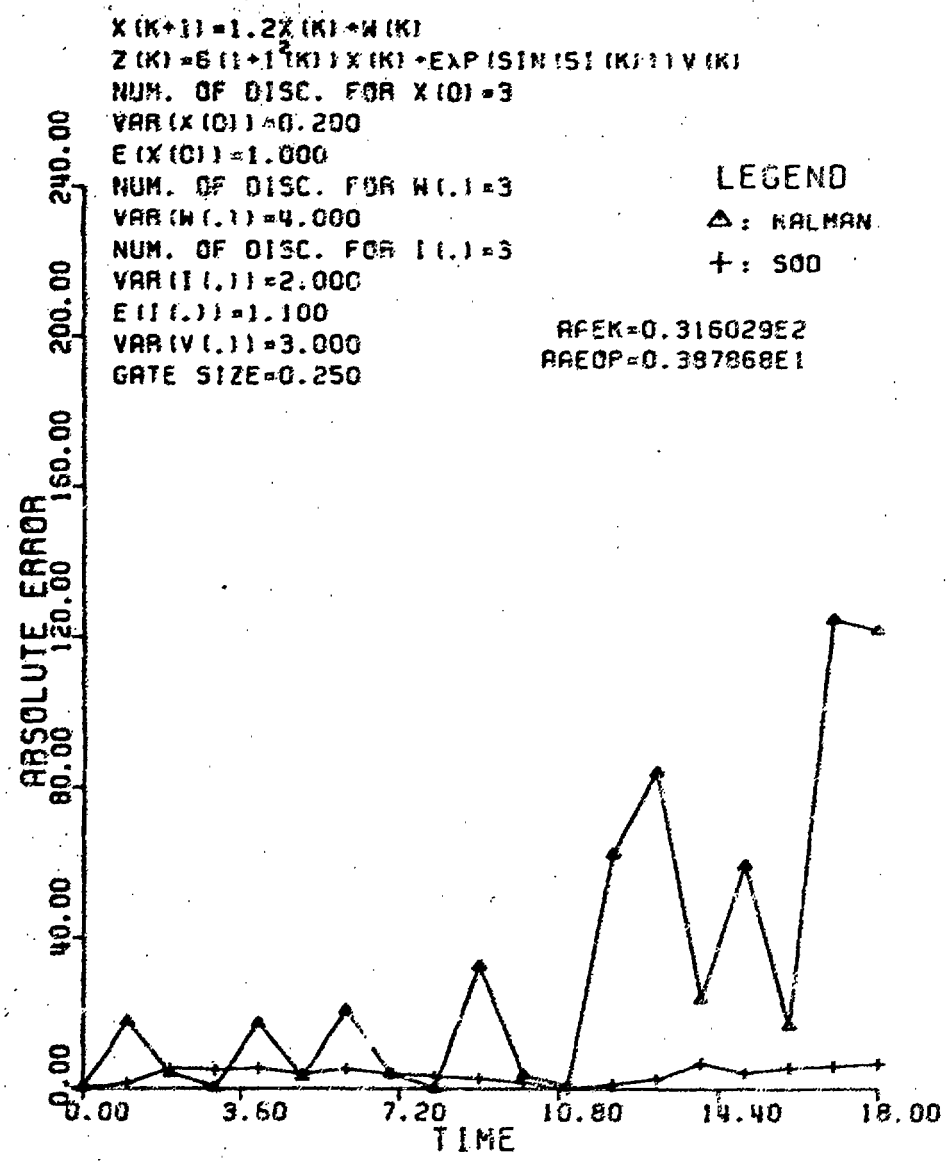


Figure 4.3.3(c) Absolute and average absolute errors

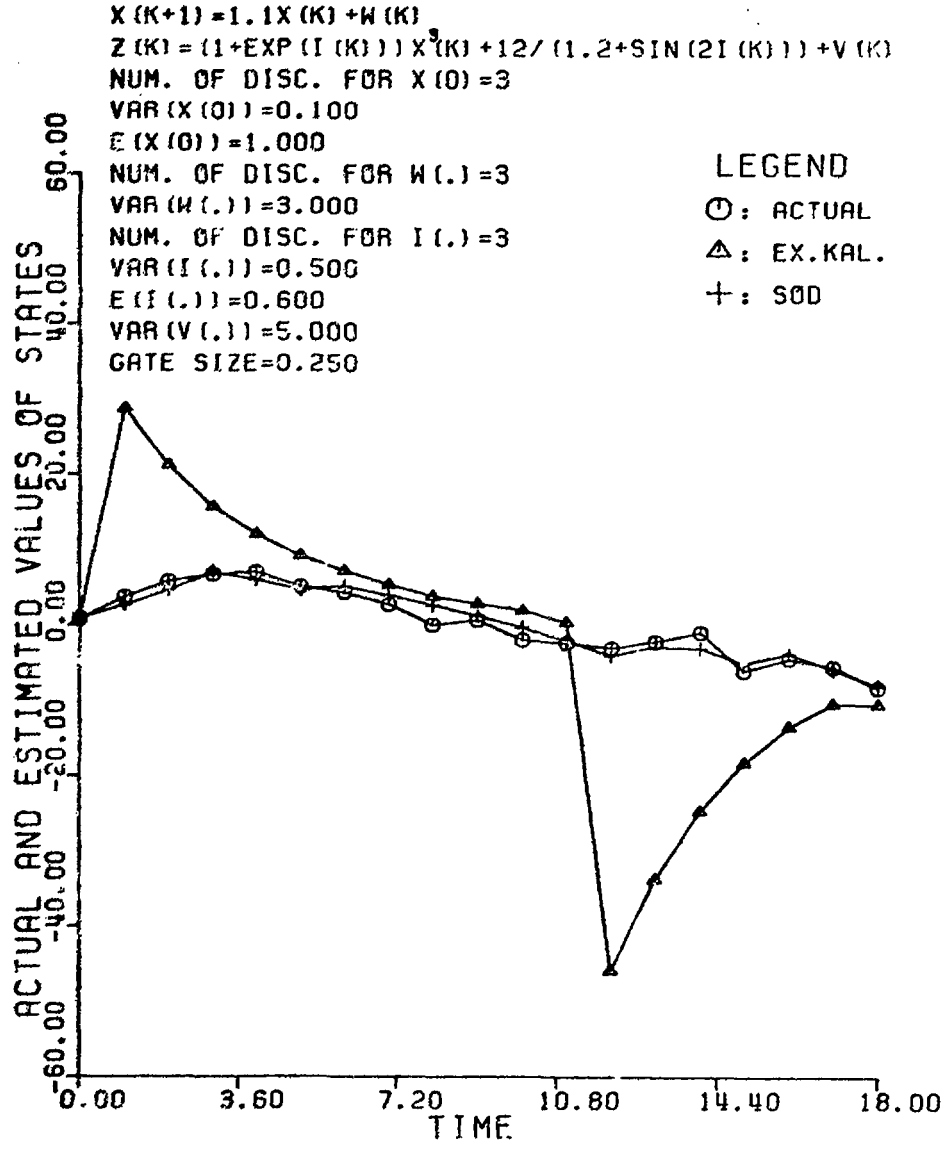


Figure 4.3.4(a) Actual and estimated values of states

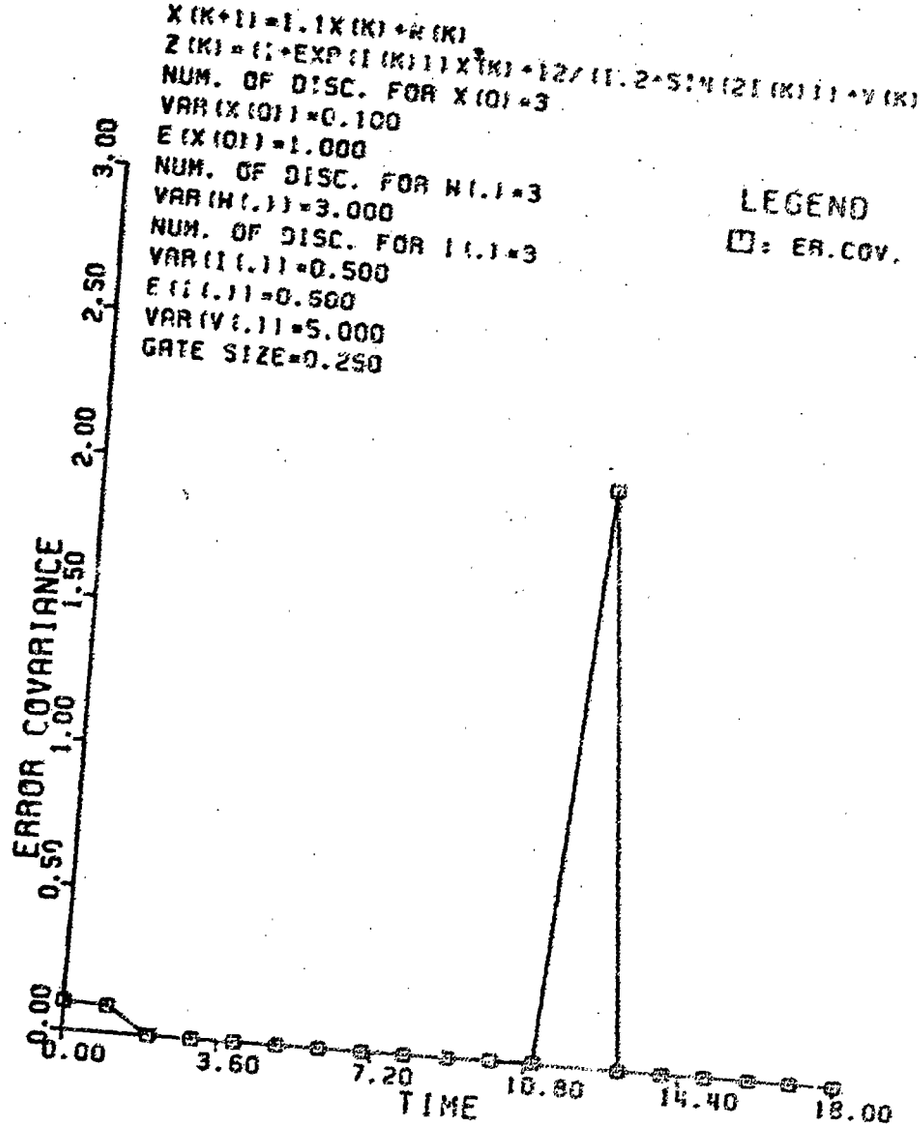


Figure 4.3.4(b) Error covariance

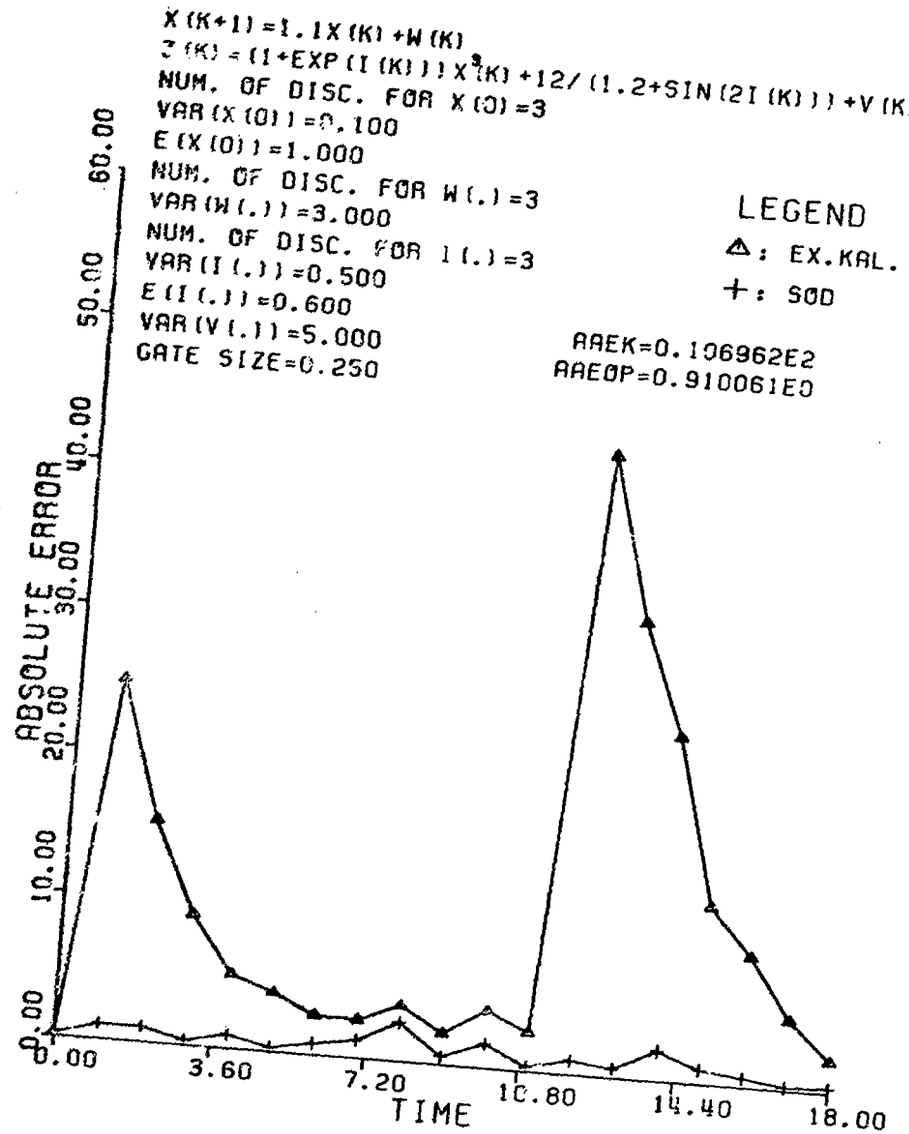


Figure 4.3.4(c) Absolute and average absolute errors

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
	AGARD-AG-256	ISBN 92-835-1418-1	UNCLASSIFIED

5. Originator Advisory Group for Aerospace Research and Development
North Atlantic Treaty Organization
7 rue Ancelle, 92200 Neuilly sur Seine, France

6. Title ADVANCES IN THE TECHNIQUES AND TECHNOLOGY OF THE APPLICATION OF NONLINEAR FILTERS AND KALMAN FILTERS

7. Presented at

8. Author(s)/Editor(s) Edited by Professor C.T.Leondes, Ph.D.

9. Date March 1982

10. Author's/Editor's Address School of Engineering and Applied Science
University of California, Los Angeles
7620 Boelter Hall
Los Angeles, California 90024, USA

11. Pages 538

12. Distribution Statement This document is distributed in accordance with AGARD policies and regulations, which are outlined on the Outside Back Covers of all AGARD publications.

13. Keywords/Descriptors

Linear filtering
Kalman filters
Linearization techniques
Nonlinear filtering
Extended Kalman filters
Correlation techniques

14. Abstract

This AGARDograph addresses recent trends and requirements for the application of advanced filtering technology and techniques. The following topics are covered:

- Advanced topics in nonlinear and linear filters,
- Computational techniques in nonlinear and linear filters, and
- Advanced nonlinear and Kalman filter application techniques and methodologies.

This AGARDograph has been prepared and edited at the request of the Guidance and Control Panel of AGARD.

AGARDograph No.256
Advisory Group for Aerospace Research and Development, NATO
ADVANCES IN THE TECHNIQUES AND TECHNOLOGY OF THE APPLICATION OF NONLINEAR FILTERS AND KALMAN FILTERS
Edited by Professor C.T.Leondes, Ph.D.
Published March 1982
538 pages

This AGARDograph addresses recent trends and requirements for the application of advanced filtering technology and techniques. The following topics are covered:

– Advanced topics in nonlinear and linear filters

P.T.O.

AGARD-AG-256

Linear filtering
Kalman filters
Linearization techniques
Nonlinear filtering
Extended Kalman filters
Correlation techniques

AGARDograph No.256
Advisory Group for Aerospace Research and Development, NATO
ADVANCES IN THE TECHNIQUES AND TECHNOLOGY OF THE APPLICATION OF NONLINEAR FILTERS AND KALMAN FILTERS
Edited by Professor C.T.Leondes, Ph.D.
Published March 1982
538 pages

This AGARDograph addresses recent trends and requirements for the application of advanced filtering technology and techniques. The following topics are covered:

– Advanced topics in nonlinear and linear filters

P.T.O.

AGARD-AG-256

Linear filtering
Kalman filters
Linearization techniques
Nonlinear filtering
Extended Kalman filters
Correlation techniques

AGARDograph No.256
Advisory Group for Aerospace Research and Development, NATO
ADVANCES IN THE TECHNIQUES AND TECHNOLOGY OF THE APPLICATION OF NONLINEAR FILTERS AND KALMAN FILTERS
Edited by Professor C.T.Leondes, Ph.D.
Published March 1982
538 pages

This AGARDograph addresses recent trends and requirements for the application of advanced filtering technology and techniques. The following topics are covered:

– Advanced topics in nonlinear and linear filters

P.T.O.

AGARD-AG-256

Linear filtering
Kalman filters
Linearization techniques
Nonlinear filtering
Extended Kalman filters
Correlation techniques

AGARDograph No.256
Advisory Group for Aerospace Research and Development, NATO
ADVANCES IN THE TECHNIQUES AND TECHNOLOGY OF THE APPLICATION OF NONLINEAR FILTERS AND KALMAN FILTERS
Edited by Professor C.T.Leondes, Ph.D.
Published March 1982
538 pages

This AGARDograph addresses recent trends and requirements for the application of advanced filtering technology and techniques. The following topics are covered:

– Advanced topics in nonlinear and linear filters

P.T.O.

AGARD-AG-256

Linear filtering
Kalman filters
Linearization techniques
Nonlinear filtering
Extended Kalman filters
Correlation techniques

– Computational techniques in nonlinear and linear filters – Advanced nonlinear and Kalman filter application techniques and methodologies. This AGARDograph has been prepared and edited at the request of the Guidance and Control Panel of AGARD. ISBN 92-835-1418-1	– Computational techniques in nonlinear and linear filters – Advanced nonlinear and Kalman filter application techniques and methodologies. This AGARDograph has been prepared and edited at the request of the Guidance and Control Panel of AGARD. ISBN 92-835-1418-1
– Computational techniques in nonlinear and linear filters – Advanced nonlinear and Kalman filter application techniques and methodologies. This AGARDograph has been prepared and edited at the request of the Guidance and Control Panel of AGARD. ISBN 92-835-1418-1	– Computational techniques in nonlinear and linear filters – Advanced nonlinear and Kalman filter application techniques and methodologies. This AGARDograph has been prepared and edited at the request of the Guidance and Control Panel of AGARD. ISBN 92-835-1418-1