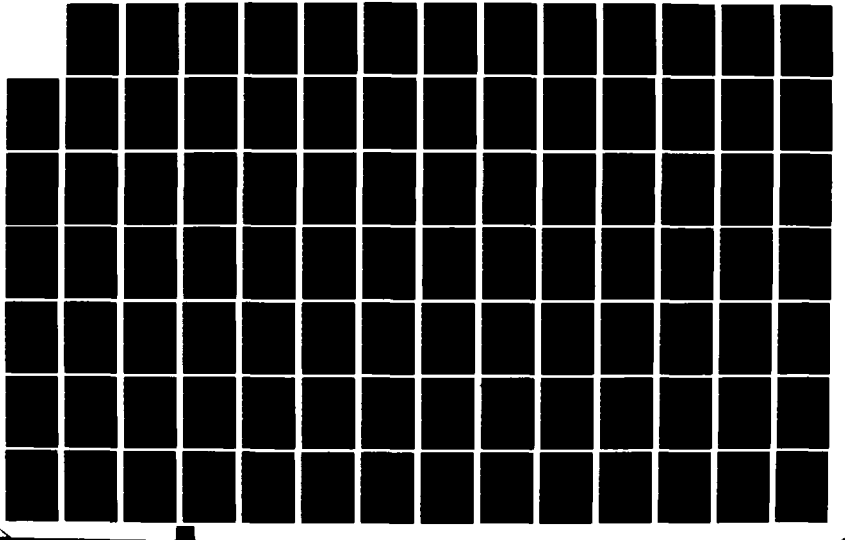
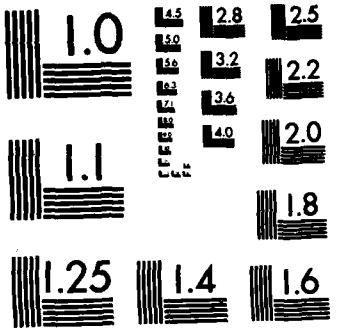
AD-A126 994 FAST KALMAN FILTERING FOR ARMA (AUTOREGRESSIVE MOVING AVERAGE) PROCESSES:.. (U) RHODE ISLAND UNIV KINGSTON DEPT OF ELECTRICAL ENGINEERING S SIGURDSSON 15 MAR 83 1/2

UNCLASSIFIED N00014-82-K-0300 F/G 12/1 NL





(12)

FAST KALMAN FILTERING FOR ARMA PROCESSES: FIXED POINT IMPLEMENTATION

Sigurdur Sigurdsson

Department of Electrical Engineering
Colorado State University
Fort Collins, Colorado 80523

March 15, 1983

Prepared for

OFFICE OF NAVAL RESEARCH (Code 411SP)
Statistics and Probability Branch
Arlington, Virginia 22217
under Contract N00014-82-K-0300
L. L. Scharf, Principal Investigator

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A126994
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Fast Kalman Filtering for ARMA Processes: Fixed Point Implementation	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Sigurdur Sigurdsson	8. CONTRACT OR GRANT NUMBER(s) N00014-82-K-0300
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Electrical Engineering University of Rhode Island Kingston, Rhode Island 02881	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (Code 411 SP) Department of the Navy Arlington, Virginia 22217	12. REPORT DATE March 15, 1983
	13. NUMBER OF PAGES 130
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Also submitted as the M.S. Thesis of Sigurdur Sigurdsson to the Electrical Engineering Department, Colorado State University, Ft. Collins, CO

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Kalman filtering, fixed-point implementation, scaling, round-off noise, microprocessor.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Kalman predictors and filters are implemented in fixed point arithmetic on a 16-bit INTEL 8086 microprocessor. Results from this implementation are compared with corresponding results for a 16-bit floating point implementation on an 8-bit 8080 microprocessor. Both implementations are carried out using an INTEL MDS 230 development system. Floating point code is written in FORTRAN and fixed point code is written in ASSEMBLY language.

(con't)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Abstract continued)

The Kalman filters and predictors are realized in a fast form that uses the so-called fast Kalman gain algorithm. This algorithm for the gain is inherently fixed point.

Scaling rules for Kalman filters and predictors are derived, and expressions are derived for rounding error variances. The numerical results show that low order fixed point realizations perform very close to the floating point realizations.

ABSTRACT

FAST KALMAN FILTERING FOR ARMA PROCESSES:
FIXED POINT IMPLEMENTATION

Kalman predictors and filters are implemented in fixed point arithmetic on a 16-bit INTEL 8086 microprocessor. Results from this implementation are compared with corresponding results for a 16-bit floating point implementation on an 8-bit 8080 microprocessor. Both implementations are carried out using an INTEL MDS 230 development system. Floating point code is written in FORTRAN and fixed point code is written in ASSEMBLY language.

The Kalman filters and predictors are realized in a fast form that uses the so-called fast Kalman gain algorithm. This algorithm for the gain is inherently fixed point.

Scaling rules for Kalman filters and predictors are derived, and expressions are derived for rounding error variances. The numerical results show that low order fixed point realizations perform very close to the floating point realizations.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Finite dimensional systems have Markovian or state space descriptions. This is well understood. Similarly, narrowband signals, such as speech, vibrations, and other natural phenomena have state space descriptions. In such descriptions, all relevant information is carried in the state. The Kalman filtering problem is one of estimating the state of a system based upon noisy measurements up to time t. Kalman prediction is the process of estimating the state based upon noisy measurements up to time t-1. The Kalman predictor is actually a one step predictor.

In general the Kalman filter (predictor) is the best filter among all linear filters. When the noise processes are Gaussian the Kalman filter (predictor) is the best linear filter among all filters linear or nonlinear. The goodness is measured in terms of the error covariance.

Among all state space models of a process there is one type of model which is of special interest. This is the so-called Innovations representation. The name comes from the fact that the model is driven by the innovations process of the output. It so happens that the innovations process of the Kalman predictor (filter) is the same as the innovations process of the signal model when there is no measurement noise. The Kalman predictor (filter) is an exact inverse of

this kind of signal model. This fact will be used later to define the Kalman predictor (filter).

In this thesis we restrict our attention to autoregressive moving average (ARMA) processes. We study the Kalman filter/ predictor both from a theoretical and from a practical point of view. Numerical results are presented for the Kalman filter implemented in floating point and fixed point arithmetic. What follows is a short overview of each chapter.

Chapter 2 is devoted to signal models for autoregressive moving average processes and their properties. The first two signal models we derive are the Markovian representations #1 and #2. They are linear time-invariant state space models. Innovations representations #1 and #2 are on the other hand linear time-varying state space models. The state vectors of these models are vectors of s-step predictors based on the infinite past for the Markovian state space models and based on the finite past for the Innovations representations.

In Chapter 3 we introduce the Kalman predictor (filter) as a sort of inverse to its assumed signal model. It is the Innovations representation #1 for the Kalman predictor and the Innovations representation #2 for the Kalman filter. Also we look at some of the properties of the Kalman predictor and filter. The Kalman gain vectors are composed of the first n+1 values of a time-varying unit pulse response. A so-called Impulse Response Algorithm is derived which allows us to calculate the unit pulse response recursively and obtain the Kalman gain vectors. Using the fact that the underlying process is an ARMA (p,q) a substantial savings in calculation is

achieved. The result is the so-called Fast Kalman Algorithm. Besides being fast, this algorithm is very well suited for fixed point implementation. It is shown that all the elements of the gain vectors are less than or equal to one in magnitude.

Chapter 4 is devoted to some of the problems which must be addressed when implementing the Kalman predictor and filter using fixed point arithmetic. Scaling of variables is used to limit the probability of overflow. This is seldom a problem in floating point arithmetic, but in fixed point arithmetic it is nearly always a problem. The effects of roundoff can be modeled as additive white noise sources and incorporated into the governing equations for the Kalman predictor and Kalman filter. Other effects as a result of coefficient truncation and input quantization are also important but not considered in this thesis. We do analyze computational complexity in terms of number of arithmetic operations.

In Chapter 5 numerical results are presented. The Kalman filter is implemented using both floating point and fixed point arithmetic and the results are compared. As it turns out the fixed point results compare very favorably with the floating point results even though the most simple scaling procedure is used. As mentioned before, the Fast Kalman Algorithm is very well suited for fixed point implementation and the numerical results confirm this. In Chapter 6 we summarize the main results of this thesis and suggest further research.

CHAPTER 2

SIGNAL MODELS FOR ARMA PROCESSES

This chapter deals with state space representations of an autoregressive moving average (ARMA) process. Markovian representations are linear time-invariant state space models. Their corresponding state vectors are vectors of s-step ahead predictors based on the infinite past. Innovations representations are a linear time varying state space models with state vectors of s-step ahead predictors based on the finite past.

In section 2.1 we derive two closely related Markovian state space models which we call Markovian representations #1 and #2 (MR1 and MR2). Section 2.2 is devoted to some of the properties of the Markovian representations. Then in section 2.3 two Innovations representations (IR1 and IR2) are introduced as time-varying Markovian counterparts. In section 2.4 the properties of the Innovations representations are studied and their properties are compared with those of the Markovian state space models.

2.1 Markovian Representations

In this section we derive two closely related state space representations of an autoregressive moving average process.

Autoregressive moving average process. A second order stationary (SOS) time series $y(t)$ is called an autoregressive moving

average process of order (p,q), (ARMA(p,q)), if it satisfies the equation

$$y(t) + a(1)y(t-1) + \ldots + a(p)y(t-p) = b(0)u(t) + b(1)u(t-1) + \ldots + b(q)u(t-q) \quad (2.1.1)$$

In operator form this equation is

$$A(z)\, y(t) = B(z)\, u(t) \quad (2.1.2)$$

where

$$A(z) = 1 + a(1)z^{-1} + \ldots + a(p)z^{-p} \quad (2.1.3)$$

$$B(z) = b(0) + b(1)z^{-1} + \ldots + b(q)z^{-q} \quad (2.1.4)$$

The input u(t) is a white noise process:

$$E[u(t)] = 0 \text{ and } E[u(t)u(t+i)] = \begin{cases} \sigma_u^2 & i=0 \\ 0 & i \text{ not zero} \end{cases}$$

The zeros of A(z) and B(z) are assumed to lie inside the unit circle. See Figure 2.1.1 for a block diagram. The unit pulse response of the ARMA(p,q) process is

$$h(t) + a(1)h(t-1) + \ldots + a(p)h(t-p) = b(0)\delta(t) + b(1)\delta(t-1) + \ldots + b(q)\delta(t-q) \quad (2.1.5)$$

Best linear predictor. The fact the zeros of A(z) and B(z) lie inside the unit circle allows us to write

$$y(t) = A^{-1}(z)B(z)u(t) = \sum_{i=0}^{\infty} h(i)u(t-1) = y(1/t-1) + h(0)u(t) \quad (2.1.6)$$

and

$$u(t) = B^{-1}(z)A(z)y(t) = \sum_{i=0}^{\infty} d(i)y(t-i) = d(0)[y(t) - y(1/t-1)] \quad (2.1.7)$$

where y(1/t-1) is the best (one step) linear predictor (BLP) for

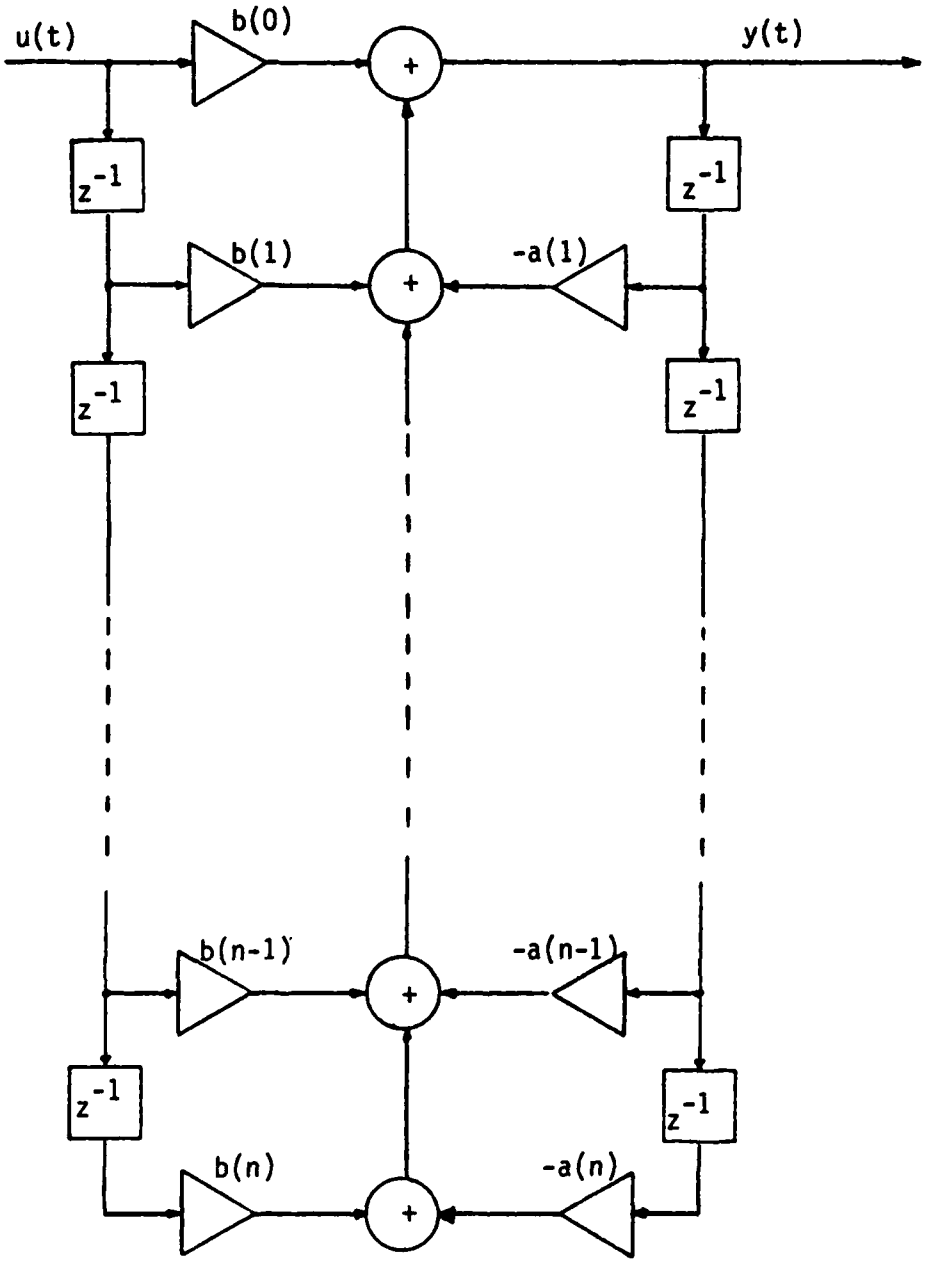


Fig. 2.1.1. Direct form realization of an ARMA(n,n) process. n=max(p,q).

y(t) given y(t-i) i = 1,2, ... It is defined as

$$d(0)y(1/t-1) = -d(1)y(t-1) - d(2)y(t-2) - \ldots =$$

$$- \sum_{i=1}^{\infty} d(i)y(t-i) \qquad (2.1.8)$$

where the coefficients d(i) i=1,2,3, . . . minimize

$$E[y(t) - y(1/t-1)]^2$$

In terms of u(t-i) i=1,2,3, . . . the BLP is

$$y(1/t-1) = \sum_{i=1}^{\infty} h(i)u(t-i) \qquad (2.1.9)$$

and the prediction error e(t) is then

$$e(t) = y(t) - y(1/t-1) = h(0)u(t) \qquad (2.1.10)$$

More generally the best linear s-step predictor given y(t-i) i=1,2,3, . . . is

$$y(s/t-1) = \sum_{i=0}^{\infty} h(s+i)u(t-1-i) \quad s= 1,2,3, \ldots \qquad (2.1.11)$$

or

$$y(s/t) = \sum_{i=0}^{\infty} h(s+i)u(t-i) \quad s= 1,2,3, \ldots \qquad (2.1.12)$$

Markovian Representation #1. We can now write

$$y(s/t) = h(s)u(t) + h(s+1)u(t-1) + \ldots$$

$$y(s+1/t-1) = \quad h(s+1)u(t-1) + \ldots$$

and after subtracting the second equation from the first

$$y(s/t) = y(s+1/t-1) + h(s)u(t) \qquad (2.1.13)$$

Using the fact that (see Box and Jenkins [31])

$$y(s/t) + a(1)y(s-1/t) + \ldots + a(n)y(s-n/t) = 0 \quad s>=n \qquad (2.1.14)$$

or

$$y(n/t-1) = -a(1)y(n-1/t-1) - \ldots -a(n)y(1/t-1) \qquad (2.1.15)$$

where n = max(p,q) we can write down a state space representation using equation (2.1.13) for s=1,2, . . ., n and (2.1.15)

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{h}(1)u(t) \tag{2.1.16}$$

$$y(t) = y(1/t-1) + h(0)u(t)$$

$$= \underline{c}'\underline{x}(t/t-1) + h(0)u(t) \tag{2.1.17}$$

Here

$$\underline{x}'(t/t-1) = [y(1/t-1), y(2/t-1), \ldots y(n/t-1)] \tag{2.1.18}$$

$$\underline{h}'(1) = [h(1), h(2), \ldots, h(n)] \tag{2.1.19}$$

$$\underline{c}' = (1 \quad 0 \quad 0 \ldots 0) \tag{2.1.20}$$

and

$$A = \begin{bmatrix} 0 & 1 & 0 & . & . & 0 \\ 0 & 0 & 1 & 0 & . & 0 \\ & & & & & \\ & & & & & 1 \\ -a(n). & & . & . & . & -a(1) \end{bmatrix} \tag{2.1.21}$$

The output y(t) is simply the predicted value plus the prediction error. See Figure 2.1.2 for a block diagram. We will refer to equations (2.1.16) and (2.1.17) as the Markovian representation #1. Note:

The state vector consists of s-step ahead predictors for s=1,2,3, . . . n based on the infinite past:

$\underline{x}'(t/t-1) = [y(1/t-1), y(2/t-1), \ldots, y(n/t-1)]$

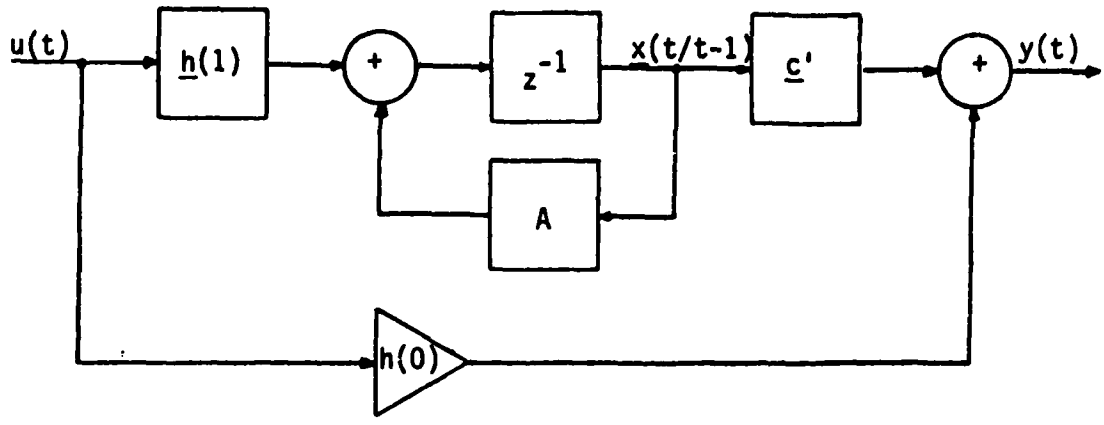
Markovian Representation #2. Now we write

$y(s/t+1) = h(s)u(t+1) + h(s+1)u(t) + \ldots$

$y(s+1/t) = \qquad\qquad h(s+1)u(t) + \ldots$

and after subtracting the second equation from the first:

$$y(s/t+1) = y(s+1/t) + h(s)u(t+1) \tag{2.1.22}$$



Fit. 2.1.2. Markovian representation #1.

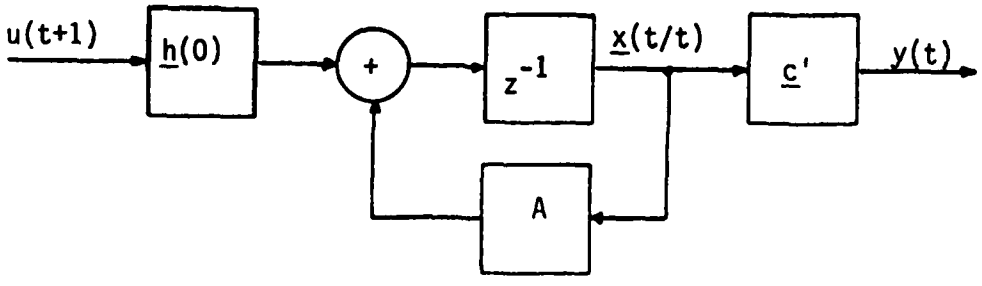


Fig. 2.1.3. Markovian representation #2.

Note that

$$y(n/t) = -a(n)y(n-1/t) - \ldots - a(1)y(0/t)$$

We can write for s=0,1, . . .,n-1 another state space representation

$$\underline{x}(t+1/t+1) = A\underline{x}(t/t) + \underline{h}(0)u(t+1) \qquad (2.1.23)$$

$$y(t) = \underline{c}'\underline{x}(t/t) \qquad (2.1.24)$$

Here

$$\underline{x}'(t/t) = [y(0/t), y(1/t), \ldots, y(n-1/t)] \qquad (2.1.25)$$

$$\underline{h}'(0) = [h(0), h(1), \ldots, h(n-1)] \qquad (2.1.26)$$

and $\underline{c}$ and A are defined as before. The output y(t) is the first element in the state vector since y(0/t) = y(t) by definition. We will refer to equations (2.1.23) and (2.1.24) as the Markovian representation #2. This representation is the same as derived in the paper by Akaike [2] and the previous derivation follows his closely. Figure 2.1.3 shows a block diagram of the Markovian representation #2. Note:

The state vector consists of s-step ahead predictors for s=0,1,2, . . ., n-1 based on the infinite past:

$$\underline{x}'(t/t) = [y(0/t), y(1/t), \ldots, y(n-1/t)]$$

The relation between the two state vectors $\underline{x}(t/t-1)$ and $\underline{x}(t/t)$ is simply

$$\underline{x}(t/t-1) = A\,\underline{x}(t-1/t-1) \qquad (2.1.27)$$

Also note that

$$\underline{h}(1) = A\,\underline{h}(0) + \underline{d}\,b(n) \qquad (2.1.28)$$

$$\underline{d}' = [0 \;\; 0 \ldots 1] \qquad (2.1.29)$$

which follows from equation (2.1.5).

2.2 Properties of the Markovian Representations

First and second order statistics of the two Markovian representations are the main subject of this section. Let's begin with the Markovian representation #1.

Markovian Representation #1

Expected value of the state vector. The MR1 is described by

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{h}(1)u(t) \tag{2.2.1}$$

$$y(t) = \underline{c}'\underline{x}(t/t-1) + h(0)u(t) \tag{2.2.2}$$

Rewrite equation (2.2.1) as

$$\underline{x}(t/t-1) = A^t\underline{x}(0/-1) + \sum_{i=1}^{t} A^{i-1}\underline{h}(1)u(t-i)$$

$$= \sum_{i=1}^{\infty} A^{i-1}\,\underline{h}(1)u(t-i) \tag{2.2.3}$$

Taking the expectation on both sides of equation (2.2.3)

$$E[\underline{x}(t/t-1)] = \sum_{i=1}^{\infty} A^{i-1}\,\underline{h}(1)E[u(t-i)] = \underline{0} \tag{2.2.4}$$

since

$$E[u(t-i)] = 0 \quad \forall\ i >= 0$$

Unit pulse response, h(t). The first n+1 values of the unit pulse response are determined from equation (2.1.5):

$$h(t) = b(t) - \sum_{i=1}^{n} a(i)h(t-i) \quad t=0,1,\ldots,n \tag{2.2.5}$$

where n=max(p,q) as before. For t > n, h(t) is simply

$$h(t) = -\sum_{i=1}^{n} a(i)h(t-i) \quad t > n \tag{2.2.6}$$

or

$$h(t) = \begin{cases} 0 & t>0 \\ h(0) & t=0 \\ \underline{c}'A^{t-1}\underline{h}(1) & t>=1 \end{cases} \qquad (2.2.7)$$

$$\underline{h}'(1) = [h(1),h(2), \ldots, h(n)]$$

This result can also be obtained by writing

$$y(t) = c'\underline{x}(t/t-1) + \underline{h}(0)u(t)$$

and substitute equation (3.2.3) for $\underline{x}(t/t-1)$:

$$y(t) = h(0)u(t) + \sum_{i=1}^{\infty} \underline{c}'A^{i-1}\underline{h}(1)u(t-i)$$

State variance, Q(t). The variance of state vector is given by

$$Q(t+1) = E[\underline{x}(t+1/t)\underline{x}'(t+1/t)]$$

$$E[(A\underline{x}(t/t-1)+\underline{h}(1)u(t))(A\underline{x}(t/t-1)+\underline{h}(1)u(t))']$$

$$= AQ(t)A' + \sigma_u^2\, \underline{h}(1)\underline{h}'(1)$$

Given initial condition in the infinite past, $Q(-\infty)$, it must hold that

$$\bar{Q}(0) = A^*\bar{Q}(0)A' + \sigma_u^2\underline{h}(1)\underline{h}'(1) \qquad (2.2.9)$$

$$Q(t) = \bar{Q}(0) \text{ for } t>=0$$

State covariance, R(t). The state covariance is given by

$$R(t) = [\underline{x}(s/s-1)\underline{x}'(s+t/s+t-1)]$$

$$E[\underline{x}(s/s-1)(A^t\underline{x}(s/s-1) + \sum_{i=1}^{t} A^{i-1}\, \underline{h}(1)u(t+s-i))']$$

$$= E[\underline{x}(s/s-1)\underline{x}'(s/s-1)](A^t)'$$

$$= A^tQ(s)$$

$$= A^t\bar{Q}(0) \qquad (2.2.10)$$

Output covariance, r(t). The covariance of y(t) is determined

by

$$r(t) = E[y(s)y'(s+t)]$$

$$E[(\underline{c}'\underline{x}(s/s-1) + h(0)u(s))(\underline{c}'\underline{x}(s+t/s+t-1) + h(0)u(s+t))']$$

$$= \underline{c}'E[\underline{x}(s/s-1)\underline{x}'(s+t/s+t-1)]\underline{c} + h(0)E[u(s)\underline{x}'(s+t/s+t-1)]\underline{c}$$

$$= \underline{c}'R(t)\,\underline{c} + \sigma_u^2\, h(0)h(t) \qquad (2.2.11)$$

Here we have used

$$E[u(s)\underline{x}'(s+t/s+t-1)] = \sigma_u^2\,(A^{t-1}\underline{h}(1))' \qquad (2.2.12)$$

$$h(t) = \underline{c}'A^{t-1}\underline{h}(1)$$

Also note that $r(t) = r(-t)$.

By using similar techniques we derive the corresponding properties for the Markovian representation #2.

Markovian Representation #2

Expected value of the state vector. The MR2 is described by

$$\underline{x}(t+1/t+1) = A\underline{x}(t/t) + \underline{h}(0)u(t+1)$$

$$y(t) = \underline{c}'\underline{x}(t/t)$$

Writing

$$\underline{x}(t/t) = A^{t+1}\underline{x}(-1/-1) + \sum_{i=0}^{t} A^i\underline{h}(0)\,u(t-i)$$

$$= \sum_{i=0}^{\infty} A^i\underline{h}(0)\,u(t-i) \qquad (2.2.13)$$

and taking the expectation on both sides:

$$E[\underline{x}(t/t)] = E[\sum_{i=0}^{\infty} A^i\underline{h}(0)u(t-i)] = 0 \qquad (2.2.14)$$

The unit pulse response, h(t). This follows by inspection from equation (2.2.13) and is given by

$$h(t) = \begin{cases} 0 & t<0 \\ \\ \underline{c}'A^t\,\underline{h}(0) & t>0 \end{cases} \qquad (2.2.15)$$

State variance, $Q(t)$. The state variance is given by

$$Q(t+1) = AQ(t)A' + \sigma_u^2 \underline{h}(0)\underline{h}'(0) \qquad (2.2.16)$$

The stationary variance satisfies

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2 \underline{h}(0)\underline{h}'(0) \qquad (2.2.17)$$

State covariance, $R(t)$. The state covariance is given by

$$R(t) = A^t \bar{Q}(0) \quad t>=0 \qquad (2.2.18)$$

Output covariance, $r(t)$. The output covariance of MR2 equals that of the MR1 and is given by

$$r(t) = \underline{c}' R(t)\, \underline{c} \quad t>=0 \qquad (2.2.19)$$

Table 1 summarizes the basic properties of the two Markovian representations we have derived is this section.

2.3 Innovations Representations

In the time invariant Markovian state space representations studied in the previous sections the non-zero initial conditions are random vectors:

$$\underline{x}(0/-1) : N(\underline{0},\bar{Q}(0))$$

$$\underline{x}(0/0) : N(\underline{0},\bar{Q}(0))$$

where $\bar{Q}(0)$ and $\bar{Q}(0)$ are determined from

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2 \underline{h}(1)\underline{h}'(1)$$

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2 \underline{h}(0)\underline{h}'(0)$$

One way of deriving the innovations representations for an ARMA(p,q) process is to introduce them as inverses of a Kalman predictor/filter. This is done by Anderson and Moore [4]. It may be done more directly by treating them as time varying counterparts of the Markovian representations as done by Gueguen and Scharf [1]. The Kalman predictor/filter is introduced as a kind of inverse of the Innovations representations. I choose the latter way.

Table 1. Governing equations for the Markovian representation #1 and #2.

Markovian Representation #1	Markovian Representation #2
State Space Model	
$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{h}(1)u(t);\ \underline{x}(0/-1):N(\underline{0},\bar{Q}(0))$	$\underline{x}(t+1/t+1) = A\underline{x}(t/t) + \underline{h}(0)u(t+1)$
$y(t) = \underline{c}'\underline{x}(t/t-1) + h(0)u(t)$	$y(t) = \underline{c}'\underline{x}(t/t);\ \underline{x}(0/0) : N(\underline{0},\tilde{Q}(0))$
Unit Pulse Response	
$h(i) = \begin{cases} 0 & i<0 \\ h(0) & i=0 \\ \underline{c}'A^{i-1}\underline{h}(1) & i>0 \end{cases}$	$h(i) = \begin{cases} 0 & i<0 \\ \underline{c}'A^{i}\underline{h}(0) & i\geq 0 \end{cases}$
State Variance	
$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2\underline{h}(1)\underline{h}'(1)$	$\tilde{Q}(0) = A\tilde{Q}(0)A' + \sigma_u^2\underline{h}(0)\underline{h}'(0)$
State Covariance	
$R(t) = A^t\bar{Q}(0)$	$R(t) = A^t\tilde{Q}(0)$
Output Covariance	
$r(t) = \underline{c}'R(t)\underline{c} + \sigma_u^2h(0)h(t)$	$r(t) = \underline{c}'R(t)\underline{c}$
$\underline{h}'(0) = [h(0)\ h(1)\ \ldots\ h(n-1)]$	$A = \begin{bmatrix} 0 & 1 & 0 & . & . & . & 0 \\ 0 & 0 & 1 & 0 & . & . & 0 \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & & & 1 \\ -a(n) & . & . & . & . & & -a(1) \end{bmatrix}$
$\underline{h}'(1) = [h(1)\ h(2)\ \ldots\ h(n)]$	
$\underline{c}' = (1\ 0\ \ldots\ 0)$	

Replacing $\underline{h}(1)$ by $\underline{k}(t)$, $\underline{h}(0)$ by $\underline{k}(t+1)$ and σ_u^2 by $v(t)$ in the Markovian representations we get time-varying representations called the Innovarions representations.

Innovations representation #1. The defining equations are

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t) \tag{2.3.1}$$

$$y(t) = \underline{c}'\underline{x}(t/t-1) + h(0)u(t) \tag{2.3.2}$$

where $\underline{k}(t)$ is a time-varying vector

$$\underline{k}'(t) = [k^t(1), k^t(2), \ldots, k^t(n)] \tag{2.3.3}$$

and $u(t)$ is the innovations sequence

$$u(t) : N(0,v(t)) \tag{2.3.4}$$

Innovations representation #2. The defining equations are

$$\underline{x}(t+1/t+1) = A\underline{x}(t/t) + \underline{k}(t+1)u(t+1) \tag{2.3.5}$$

$$y(t) = \underline{c}'\underline{x}(t/t) \tag{2.3.6}$$

where $\underline{k}(t)$ is a time varying vector

$$\underline{k}'(t) = [k^t(1), k^t(2), \ldots, k^t(n)] \tag{2.3.7}$$

and $u(t)$ the innovations sequence

$$u(t) : N(0,v(t)) \tag{2.3.8}$$

The initial conditions on the state vectors $\underline{x}(t/t-1)$ and $\underline{x}(t/t)$ are chosen to be identically zero:

$$\underline{x}(0/-1) = \underline{0} \tag{2.3.9}$$

$$\underline{x}(-1/-1) = \underline{0} \tag{2.3.10}$$

The definitions for A and $\underline{c}$ are the same as for the Markovian representations.

The Innovations representations are the time-varying state space models that start from zero initial conditions rather than from random initial conditions as in the Markovian representations [1]. The state vectors of the Innovations representations are still

vectors of s-step predictors but based on the finite past:

$$\underline{x}'(t/t-1)=[y(1/t-1),y(2/t-1), \ldots, y(n/t-1)] \qquad (2.3.11)$$

$$\underline{x}'(t/t) = [y(0/t),y(1/t), \ldots, y(n-1/t)] \qquad (2.3.12)$$

Figures 2.3.1 and 2.3.2 show block diagrams of the two Innovations representations.

In section 2.4 we show how to choose $\underline{k}(t)$, $\underline{k}(t)$ and $v(t)$ in order to generate stationary data $y(t)$ with the same correlations sequence as that of the Markovian representations.

2.4 Properties of the Innovations Representations

As in section 2.2 we derive some first and second order statistics for the two Innovations representations and then show what conditions must be satisfied to match the output covariance of the Innovations representations to the one of the Markovian representations. We start with the Innovations representation #1.

Innovations Representation #1

Expected value of the state vector. The IR1 is described by

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$$

$$y(t) = \underline{c}'\underline{x}(t/t-1) + h(0)u(t)$$

The first equation can be written as

$$\underline{x}(t/t-1) = A^t\underline{x}(0/-1) + \sum_{i=1}^{t} A^{i-1}\underline{k}(t-i)u(t-i)$$

$$= \sum_{i=1}^{\infty} A^{i-1}\underline{k}(t-i)u(t-i) \qquad (2.4.1)$$

Taking the expectation on both sides we get

$$E[\underline{x}(t/t-1)] = \sum_{i=1}^{\infty} A^{i-1}\underline{k}(t-i)E[u(t-i)] = \underline{0} \qquad (2.4.2)$$

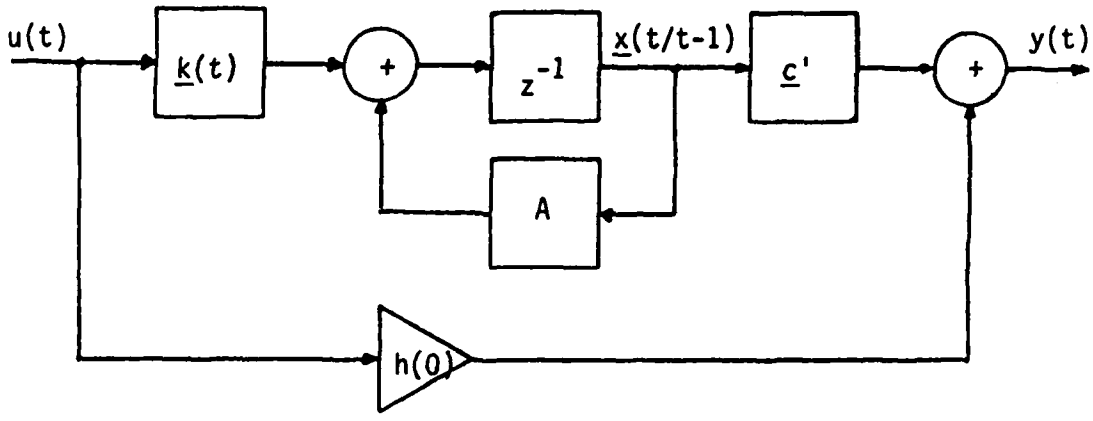


Fig. 2.3.1. Innovations representation #1.

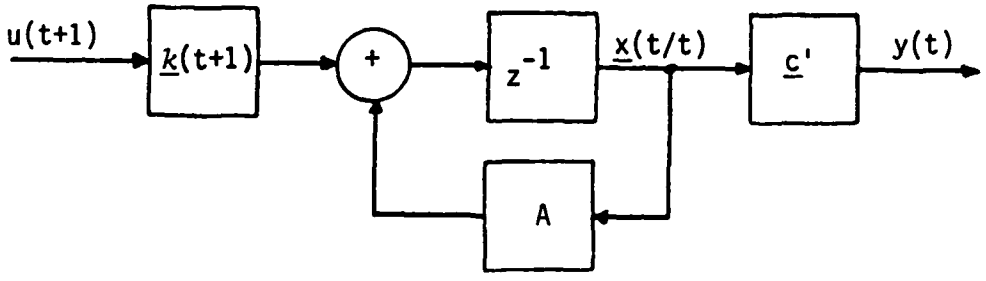


Fig. 2.3.2. Innovations representation #2.

Unit pulse response, $h^t(i)$. We know that the unit pulse response must satisfy

$$y(t) = \sum_{i=0}^{\infty} h^t(i)u(t-i) \tag{2.4.3}$$

Now

$$y(t) = \underline{c}'\underline{x}(t/t-1) + h(0)u(t) \tag{2.4.4}$$

Substituting equation (2.4.1) for $\underline{x}(t/t-1)$ in (2.4.4) to obtain

$$y(t) = h(0)u(t) + \sum_{i=1}^{\infty} A^{i-1}\underline{k}(t-i)u(t-i) \tag{2.4.5}$$

Compare equations (2.4.3) and (2.4.5) to obtain the unit pulse response:

$$h^t(i) = \begin{cases} 0 & i<0 \\ h(0) & i=0 \\ \underline{c}'A^{i-1}\underline{k}(t) & i>0 \end{cases} \tag{2.4.6}$$

Note that $h^t(i)$ is the response at time t+i to an impulse applied at time t.

State variance, Q(t). The variance of the state vector is given by

$$\begin{aligned} Q(t+1) &= E[\underline{x}(t+1/t)\underline{x}'(t+1/t)] = \\ &E[(A\underline{x}(t/t-1)+\underline{k}(t)u(t))(A\underline{x}(t/t-1)+\underline{k}(t)u(t))'] \\ &= AQ(t)A' + v(t)\underline{k}(t)\underline{k}'(t) \end{aligned} \tag{2.4.7}$$

Here

$$Q(0) = 0 \tag{2.4.8}$$

$$v(t) = E[u(t)u'(t)] \tag{2.4.9}$$

State covariance, $R^t(k)$. The state covariance is given oy

$$R^t(k) = [\underline{x}(t/t-1)\underline{x}'(t+k/t+k-1)]$$

$$= E[\underline{x}(t/t-1)(A^k\underline{x}(t/t-1) + \sum_{i=1}^{k} A^{i-1}\underline{k}(t+k-i)u(t+k-i))']$$

$$= A^kQ(t) \qquad (2.4.10)$$

since

$$E[\underline{x}(t/t-1)u'(t+k-i)] = \underline{0} \text{ for } i=1,2, \ldots, k$$

Output covariance, $r^t(k)$. The covariance of y(t) is determined from

$$r^t(k) = E[y(t)y'(t+k)]$$

$$= E[(\underline{c}'\underline{x}(t/t-1)+h(0)u(t))(\underline{c}'\underline{x}(t+k/t+k-1)+h(0)u(t+k))']$$

$$= \underline{c}'R^t(k)\ \underline{c} + v(t)h(0)h^t(k) \qquad (2.4.11)$$

This follows from

$$E[u(t)\underline{x}'(t+k/t+k-1)] = v(t)(A^{k-1}\underline{k}(t))' \qquad (2.4.12)$$

and equation (2.4.6).

Innovations Representation #2

Expected value of the state vector. The IR2 is described by

$$\underline{x}(t+1/t+1) = A\underline{x}(t/t) + \underline{k}(t+1)u(t+1)$$

$$y(t) = \underline{c}'\underline{x}(t/t)$$

The state equation can be written as

$$\underline{x}(t/t) = A^{t+1}\underline{x}(-1/-1) + \sum_{i=0}^{t} A^i\underline{k}(t-i)u(t-i) \qquad (2.4.13)$$

Taking the expectation on both sides results in

$$E[\underline{x}(t/t)]=E[A^{t+1}\underline{x}(-1/-1) + \sum_{i=0}^{t} A^i\underline{k}(t-i)u(t-i)]=\underline{0} \qquad (2.4.14)$$

since $\underline{x}(-1/-1) = \underline{0}$ and $E[u(t-i)]=0 \quad \forall i=>0$

Unit pulse response, $h^t(i)$. We have

$$y(t) = \underline{c}'\underline{x}(t/t) = \underline{c}' \sum_{i=0}^{t} A^i \underline{k}(t-i)u(t-i)$$

so the unit pulse response is simply

$$h^t(i) = \begin{cases} 0 & i<0 \\ \underline{c}'A^i\underline{k}(t) & i\geq 0 \end{cases} \qquad (2.4.15)$$

By comparison with equation (2.4.6) the relation between $\underline{k}(t)$ and $\underline{k}(t)$ is

$$\underline{k}(t) = A\,\underline{k}(t) \qquad (2.4.16)$$

State variance, $Q(t)$. The state variance is determined from

$$Q(t)=E[\underline{x}(t/t)\underline{x}'(t/t)]$$

$$= E(A\underline{x}(t-1/t-1)+\underline{k}(t)u(t))(A\underline{x}(t-1/t-1)+\underline{k}(t)u(t))']$$

$$= AQ(t-1)A' + v(t)\underline{k}(t)\underline{k}'(t) \qquad (2.4.17a)$$

or

$$Q(t) = Q(t) + v(t)\underline{k}(t)\underline{k}'(t) \qquad (2.4.17b)$$

Q(t) is the state variance of the Innovations rep. #1.

State covariance, $R^t(k)$. The state covariance is given by

$$R^t(k)=E[\underline{x}(t/t)\underline{x}'(t+k/t+k)] =$$

$$E[\underline{x}(t/t)(A^k\underline{x}(t/t) + \sum_{i=0}^{k-1} A^i\underline{k}(t+k-i)u(t+k-i))']$$

$$= A^kQ(t) \qquad (2.4.18)$$

since

$$E[\underline{x}(t/t)u'(t+k-i)] = \underline{0} \text{ for } i=0,1,2, \ldots, k-1$$

Output covariance, $r^t(k)$. The covariance of y(t) is determined by

$$r^t(k) = E[y(t)y'(t+k)] = \underline{c}'E[\underline{x}(t/t)\underline{x}'(t+k/t+k)]\underline{c}$$

$$= \underline{c}'\, R^t(k)\, \underline{c} \qquad (2.4.19)$$

Table 2 summarizes the basic properties of the two Innovations representations we have derived in this section.

The output covariance $r^t(k)$ of an Innovations representation is generally non-stationary but we would like to make it stationary or more specifically match it to that of the corresponding Markovian representation. For the Innovation representations #1

$$r^t(k) = c'R^t(k)c + v(t)h(0)h^t(k)$$

$$= \underline{c}'A^kQ(t)\underline{c} + h(0)\underline{c}'A^{k-1}\underline{k}(t)v(t) \qquad (2.4.20)$$

Gueguen and Scharf [1] show that by choosing

$$\underline{k}(t)v(t) = -AQ(t)\underline{c} + A\bar{Q}(0)\underline{c} + \sigma_u{}^2\underline{h}(1) \qquad (2.4.21)$$

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u{}^2\underline{h}(1)\underline{h}'(1)$$

$r^t(k)$ is made equal to $r(k)$ for the Markovian representation #1.

Plug equation (2.4.21) into (2.4.20)

$$r^t(k)=h(0)\underline{c}'A^k\bar{Q}(0)\underline{c}+\sigma_u{}^2h(0)\underline{c}'A^{k-1}\underline{h}(1)+(1-h(0))\underline{c}'A^kQ(t)\underline{c}$$

or

$$r^t(k) = \underline{c}'A^k\bar{Q}(0)\underline{c} + \sigma_u{}^2h(k) = r(k) \qquad (2.4.22)$$

The variance of the innovations sequence u(t), v(t), becomes

$$v(t)=E[u(t)u'(t)]=E[(y(t)-\underline{c}'\underline{x}(t/t-1))(y(t)-\underline{c}'\underline{x}(t/t-1))']$$

$$= E[y(t)y'(t)]-\underline{c}'E[\underline{x}(t/t-1)\underline{x}'(t/t-1)]\underline{c}$$

$$= r(0) - \underline{c}'Q(t)\underline{c}$$

$$= c'(\bar{Q}(0)-Q(t))c + \sigma_u{}^2 \qquad (2.4.23)$$

since $r(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_u{}^2$.

The output covariance for Innovations representations #2 is

$$r^t(k)=\underline{c}'A^kQ(t)\underline{c}=\underline{c}'A^k(AQ(t-1)A'+v(t)\underline{k}(t)\underline{k}'(t))\underline{c}$$

$$= \underline{c}'A^{k+1}Q(t-1)A'\underline{c} + \underline{c}'h(0)A^k\underline{k}(t)v(t) \qquad (2.4.24)$$

Table 2. Governing equations for the Innovations representation #1 and #2.

Innovations Representation #1	Innovations Representation #2
State Space Model	
$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$	$\underline{x}(t+1/t+1) = A\underline{x}(t/t) + \underline{k}(t+1)u(t+1)$
$y(t) = \underline{c}'\underline{x}(t/t-1) + h(0)u(t);\ \underline{x}(0/-1)=0$	$y(t) = \underline{c}'\underline{x}(t/t);\ \underline{x}(-1/-1)=\underline{0}$
Unit Pulse Response	
$h^t(i) = \begin{cases} 0 & i<0 \\ h(0) & i=0 \\ \underline{c}'A^{i-1}\underline{k}(t) & i>0 \end{cases}$	$h^t(i) = \begin{cases} 0 & i<0 \\ \underline{c}'A^i\underline{k}(t) & i\geq 0 \end{cases}$
State Variance	
$Q(t+1) = AQ(t)A' + v(t)\underline{k}(t)\underline{k}';\ Q(0)=0$	$Q(t) = AQ(t-1)A' + v(t)\ \underline{k}(t)\underline{k}'(t);\ Q(-1)=0$
State Covariance	
$R^t(k) = A^kQ(t)$	$R^t(k) = A^kQ(t)$
Output Covariance	
$r^t(k) = \underline{c}'R^t(k)\underline{c} + v(t)h(0)h^t(k)$	$r^t(k) = \underline{c}'R^t(k)\underline{c}$
$\underline{k}'(t) = [k^t(0)k^t(1) \ldots k^t(n-1)]$	$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \vdots \\ -a(n) & \cdots & & & -a(1) \end{bmatrix}$
$\underline{\ }'(t) = [k^t(1)\ k^t(2) \ldots k^t(n)]$	
$\underline{c}' = (1\ 0 \ldots 0)$	

Now we choose $\underline{k}(t)v(t)$ as follows

$$\underline{k}(t)v(t) = -AQ(t-1)A'\underline{c} + \bar{Q}(0)\underline{c} \tag{2.4.25}$$

where $\bar{Q}(0)$ satisfies

$$\bar{Q}(0) = A^*\bar{Q}(0)^*A' + \sigma_u^2\underline{h}(0)\underline{h}'(0)$$

Plug equation (2.4.25) into (2.4.24)

$$r^t(k) = \underline{c}'A^{k+1}Q(t-1)A'\underline{c}+\underline{c}'h(0)A^k[-AQ(t-1)A'+\bar{Q}(0)]\underline{c}$$

$$= \underline{c}'h(0)A^k\bar{Q}(0)\underline{c} + (1-h(0))\underline{c}'A^{k+1}\bar{Q}(t-1)A'\underline{c} \tag{2.4.26}$$

or

$$r^t(k) = r(k) = \underline{c}'A^k\bar{Q}(0)\underline{c} \tag{2.4.27}$$

The variance of the innovations sequence u(t), v(t), is now

$$v(t) = E[u(t)u'(t)] = r(0) - c'AQ(t)A'c$$

$$= \underline{c}'A[\bar{Q}(0)-Q(t-1)]A'\underline{c} + \sigma_u^2 \tag{2.4.28}$$

The main point is this. It is possible to generate stationary data, using the Innovations representations, having the same first and second order statistics as the Markovian representations. For the Innovation representation #1 this is done by choosing

$$\underline{k}(t)v(t) = -A^*Q(t)\underline{c} + A^*\bar{Q}(0)\underline{c} + \sigma_u^2h(1)$$

$$h(0) = 1$$

and for the Innovations representation #2 by choosing

$$\underline{k}(t)v(t) = -A^*Q(t-1)^*A'\underline{c} + \bar{Q}(0)\underline{c}$$

$$h(0) = 1$$

REFERENCES FOR CHAPTER 2

1. C. Gueguen and L.L. Scharf, "Exact Maximum Likelihood Identification of ARMA Models; A Signal Processing Perspective," Proc. EUSIPCO, Lausanne, Switzerland, Sept. 1980.

2. H. Akaike, "Markovian Representation of Stochastic Processes and its Application to the Analysis of Autoregressive Moving Average Processes," Ann. Inst. Statist. Math., 26, (1974).

3. G.E.P. Box and G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco (1976).

4. B.D.O. Anderson and J.B. Moore, Optimal Filtering, Prentice-Hall Inf. Syst. Series, 1979.

5. C.T. Mullis and R.A. Roberts, "The Use of Second-Order Information in the Approximation of Discrete-Time Linear Systems," IEEE Trans. Acoust., Speech, Signal Proc., ASSP-24, No. 3, June 1976.

6. L.L. Scharf, EE612; Estimation Theory, Class Notes, Colorado State University, 1981.

7. J.P. Dugre', "Parametric Spectrum Analysis of Stationary Random Sequences," Ph.D. Dissertation, Colorado State University, Summer 1981.

8. J.S. Williams, ST525; Time Series Analysis, Class Notes, Colorado State University, 1981.

CHAPTER 3

FAST KALMAN FILTERING FOR ARMA PROCESSES

Using the two time-varying signal models introduced in Chapter 2 we derive the Kalman predictor and filter as inverses of these signal models. In the case of the Kalman predictor the assumed signal model is the Innovations representation #1 and for the Kalman filter it is the Innovations representation #2. Anderson and Moore [2] introduced the Innovation representation #1 as a sort of inverse to a given Kalman predictor but the idea of doing the opposite is attributed to Gueguen and Scharf [1]. They derived the Innovations representation #1 as the time-varying counterpart of the Markovian representation #1. In section 3.1 and 3.2 we introduce the Kalman predictor and Kalman filter, respectively.

The calculations of the Kalman gain vectors, $\underline{k}(t)$ and $\underline{k}(t)$, are the main computational problems in implementing the Kalman predictor and filter. Solving a Ricatti type of equation is not the most efficient way and a better algorithm is desired. The problem of solving the Kalman gain vectors translates into that of calculating the first n+1 values of the time-varying impulse response for eact t. A so-called Impulse Response Algorithm can be used to calculate the impulse response recursively. The number of steps must be known ahead of time. This drawback is bypassed by using the fact that the underlying process is an ARMA(p,q). This gives the so-called Fast Kalman Algorithm

which calculates the Kalman gain vectors recursively. Dugre' [7] derived this algorithm for the predictor gain vector $\underline{k}(t)$ and we extend it for the filter gain vector also. This is the main objective of section 3.3.

Properties of the Kalman predictor and filter are the subject of section 3.4. Both the signal models and their corresponding predictor and filter structures are asymptotically time invariant and in the case of the Innovations representations #1 and #2 they converge to the time invariant Markovian representations #1 and #2, respectively. The Kalman predictor and filter converge to their corresponding Markovian predictor and filter structures.

3.1 The Kalman Predictor

The (one step) prediction problem is one of estimating the state vector, $\underline{x}(t)$, of the signal model given measurement data up till time t-1. For more information on the general problem of optimal prediction see references [2], [3], [4], [5], and [6]. We consider two different cases:

1) Noise-Free Measurement Process: $z(t) = y(t)$

2) Noisy Measurement Process: $z(t) = y(t) + n(t)$

Here $n(t)$ is a sequence of independent $N(0,\sigma_n^2)$ random variables.

Noise-free measurement processes. The Innovations rep. #1 is described by

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t);\ \underline{x}(0/-1)=\underline{0} \tag{3.1.1}$$

$$y(t) = \underline{c}'\underline{x}(t/t-1) + u(t);\ h(0)=1 \tag{3.1.2}$$

Solve for $u(t)$ in equation (3.1.2):

$$u(t)=y(t)-\underline{c}'\underline{x}(t/t-1)=y(t)-\hat{y}(t)=z(t)-\hat{y}(t) \tag{3.1.3}$$

Plug u(t) into equation (3.1.1) to obtain the Kalman predictor:

$$\underline{x}(t+1/t)=A\underline{x}(t/t-1)+k(\underline{t})[z(t)-\hat{y}(t)];\ \underline{x}(0/-1)=\underline{0} \tag{3.1.4}$$

$$\hat{y}(t) = \underline{c}'\underline{x}(t/t-1) \tag{3.1.5}$$

A derivation similar to this one can be found in [1]. In [2] the Innovations representation #1 is derived as an inverse to the Kalman predictor. Figure 3.1.1 shows the Innovations representation #1 (the signal model) with the Kalman predictor. In the case of a noise-free measurement process the Kalman predictor reconstructs the innovations sequence of the signal model (the Innovation rep. #1) and estimates the state vector exactly. Remember, though, that the perfectly estimated state consists of imperfect estimate of the underlying sequence y(t). The Kalman predictor can also be said to be a whitening filter if we think of u(t) as the output. The input is a correlated data sequence z(t) and the output is uncorrelated data sequence u(t). The defining equations for the Kalman predictor are then:

$$\underline{x}(t+1/t)=A\underline{x}(t/t-1)+\underline{k}(t)[z(t)-\hat{y}(t)];\ \underline{x}(0/-1)=\underline{0}$$

$$\hat{y}(t) = \underline{c}'\underline{x}(t/t-1)$$

$$\underline{k}(t)v(t) = A[\bar{Q}(0)-Q(t)]\underline{c}' + \sigma_u{}^2\underline{h}(1) \tag{3.1.6}$$

$$v(t) = r(0) - \underline{c}'Q(t)\underline{c} \tag{3.1.7}$$

$$Q(t+1) = AQ(t)A' + v(t)\underline{k}(t)\underline{k}'(t);\ Q(0)=0 \tag{3.1.8}$$

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u{}^2\underline{h}(1)\underline{h}'(1) \tag{3.1.9}$$

$$r(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_u{}^2 \tag{3.1.10}$$

These are basically the equations derived in section 2.4.

Noisy measurement process. When there is additive noise, the input to the Kalman predictor is

$$z(t) = y(t) + n(t) \tag{3.1.11}$$

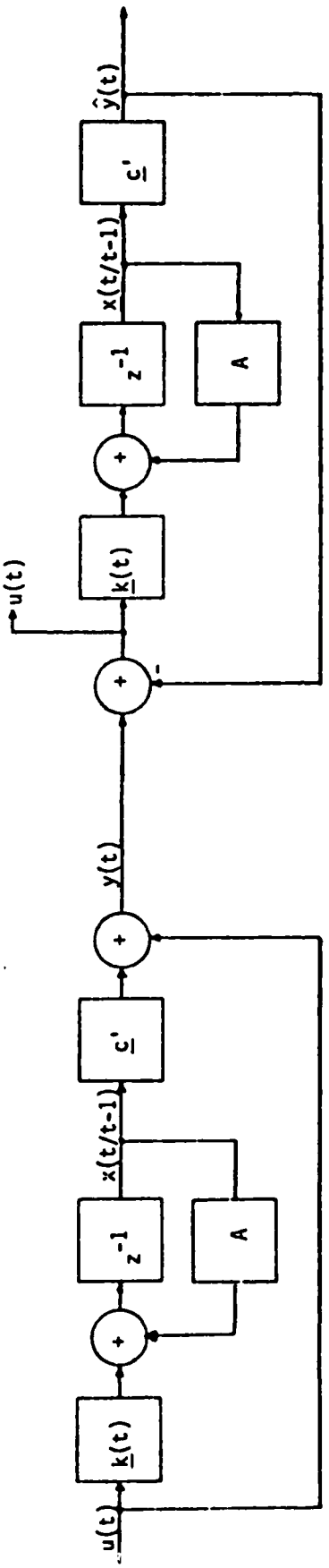


Fig. 3.1.1. The Kalman predictor and signal model Innovations rep. #1. Noise free case.

where n(t) is a sequence of independent $N(0,\sigma_n^2)$ random variables. The noise added to the output is only going to affect the output covariance of the signal model. The output covariance becomes

$$\bar{r}(k) = E[z(t)z'(t+k)] = E[(y(t)+n(t))(y(t)+n(t))']$$
$$= r(k) + \sigma_n^2\delta(k) \qquad (3.1.12)$$

and for k=0,

$$\bar{r}(0) = r(0) + \sigma_n^2 \qquad (3.1.13)$$

Then the defining equations for the Kalman predictor are

$$\underline{\hat{x}}(t+1/t) = A\underline{\hat{x}}(t/t-1) + \underline{\hat{k}}(t)\hat{u}(t);\ \underline{\hat{x}}(0/-1)=\underline{0} \qquad (3.1.14)$$

$$\hat{u}(t) = z(t) - \hat{y}(t) \qquad (3.1.15)$$

$$\hat{y}(t) = c'\underline{\hat{x}}(t/t-1) \qquad (3.1.17)$$

$$\underline{\hat{k}}(t)\hat{v}(t) = A[\bar{Q}(0)-\hat{Q}(t)]\underline{c}' + \sigma_u^2\underline{h}(1) \qquad (3.1.18)$$

$$v(t) = \bar{r}(0) - \underline{c}'\hat{Q}(t)\underline{c};\ \hat{v}(0)=\bar{r}(0) \qquad (3.1.19)$$

$$\bar{r}(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_u^2 + \sigma_n^2 = r(0) + \sigma_n^2 \qquad (3.1.20)$$

$$\hat{Q}(t+1) = A\hat{Q}(t)A' + \hat{v}(t)\underline{\hat{k}}(t)\underline{\hat{k}}'(t);\ \hat{Q}(0)=0 \qquad (3.1.21)$$

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2\underline{h}(1)\underline{h}'(1) \qquad (3.1.22)$$

Figure 3.1.2 shows the Kalman predictor with the assumed signal model, the Innovations representation #1, for the noisy case. By inverting the Kalman predictor a new signal model is obtained. This new model is driven by the innovations sequence of the Kalman predictor. The noisy measurement process is included in this model since the output is z(t)=y(t)+n(t) not just y(t). Figure 3.1.3 shows this new signal model along with the Kalman predictor. Because of how this model is derived the state vector of this new model is equal to the one of the Kalman predictor is so we can say that the Kalman predictor estimates the state of this new model with no error and reconstructs the innovations

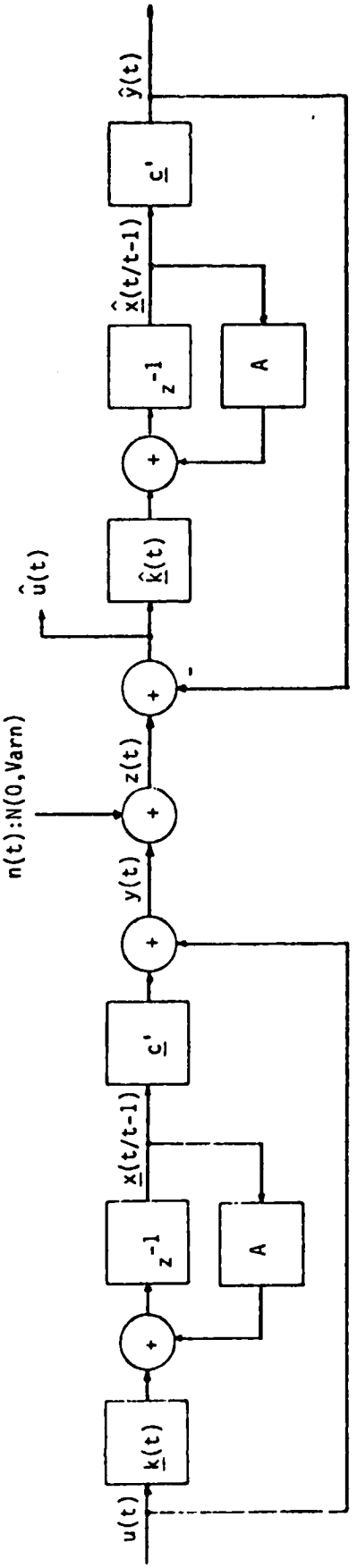


Fig. 3.1.2. The Kalman predictor and signal model the Innovations rep. #1. Noisy case.

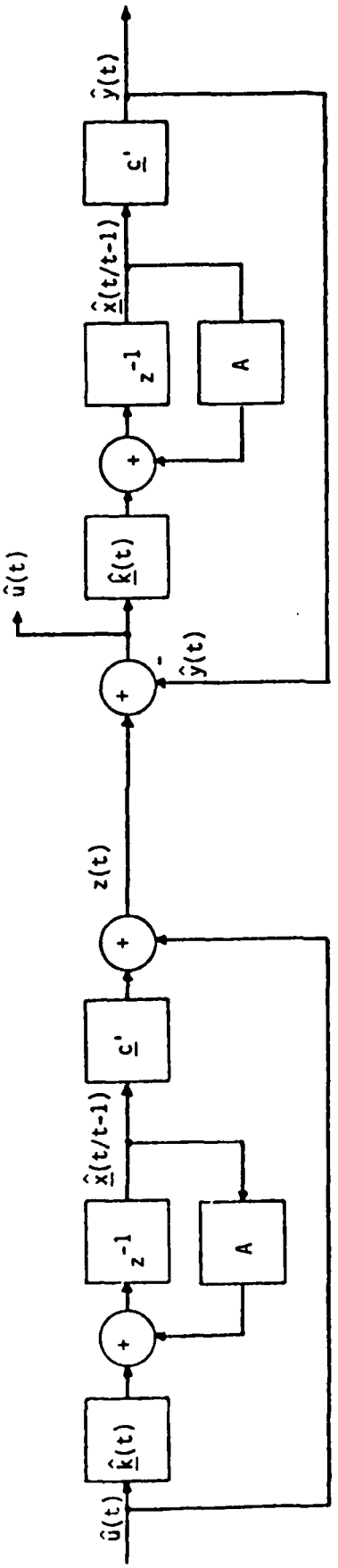


Fig. 3.1.3. The Kalman predictor and Innovations rep. #1. Measurement noise included in the signal model.

sequence perfectly. But of course the new state vector is not equal to the state vector of the old signal model.

3.2 The Kalman Filter

We said in section 3.1 that (one step) prediction is the process of estimating the state vector, $\underline{x}(t)$, of a signal model given measurement data up to time t-1. Estimating the state vector given data up to time t is said to be a filtering process. The Kalman filter is an inverse of the Innovations representation #2 and as in the Kalman predictor the innovations sequence, $u(t)$, plays the main role. We consider two cases:

1) Noise-Free Measurement Process: $z(t) = y(t)$

2) Noisy Measurement Process: $z(t) = y(t) + n(t)$

As before $n(t)$ is the measurement noise.

Noise-free measurement processes. The signal model is now the Innovations representation #2:

$$\underline{x}(t/t)=A\underline{x}(t-1/t-1)+\underline{k}(t)u(t);\ \underline{x}(-1/-1)=\underline{0} \tag{3.2.1}$$

$$y(t-1)=\underline{c}'\underline{x}(t-1/t-1) \tag{3.2.2}$$

The idea is to reconstruct the innovations sequence, $u(t)$, from the measurement data.

Assume we have a predicted output $\hat{y}(t)$. Define an innovations sequence as

$$u(t) = y(t) - \hat{y}(t) \tag{3.2.3}$$

Plug it into equation (3.2.1):

$$\begin{aligned}\underline{x}(t/t) &= A\underline{x}(t-1/t-1) + \underline{k}(t)u(t)\\ &= A\underline{x}(t-1/t-1) + \underline{k}(t)[z(t)-\hat{y}(t)]\end{aligned} \tag{3.2.4}$$

Figure 3.2.1 shows this process. We have not said how to get the estimate $\hat{y}(t)$. One way is to have the Kalman predictor produce it but

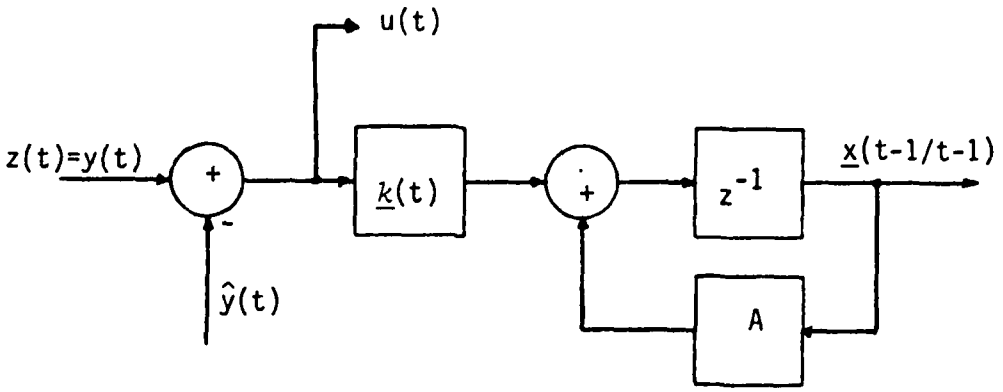


Fig. 3.2.1. The Kalman filter.

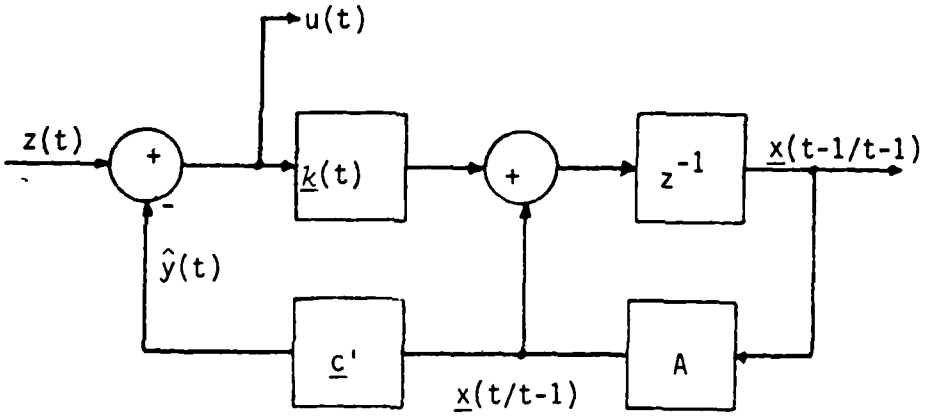


Fig. 3.2.2. The Kalman filter complete.

that does not look very attractive computationally. First note that the state vector in the Kalman predictor can be written as

$$\underline{x}(t/t-1) = A\underline{x}(t-1/t-1) \tag{3.2.5}$$

and then the predicted output becomes

$$\hat{y}(t) = \underline{c}'\underline{x}(t/t-1) = \underline{c}'A\underline{x}(t-1/t-1) \tag{3.2.6}$$

This means we can produce the estimate of the output $\hat{y}(t)$ using the old state $\underline{x}(t-1/t-1)$ in the Kalman filter. Figure 3.2.2 shows the complete Kalman filter. Like the Kalman predictor the Kalman filter reconstructs the innovations sequence, u(t), and estimates the state of the signal model (the Innovations representation #2) exactly in the case of a noise-free measurement process. Figure 3.2.3 show the Kalman filter along with the assumed signal model.

Note that we get the Kalman predictor for free in the Kalman filter since

$$\underline{x}(t/t-1) = A\,\underline{x}(t-1/t-1)$$

The Kalman filter is also a whitening filter if we think of u(t) as the output. The defining equations for the Kalman filter in the noise-free case are

$$\underline{x}(t/t) = A\underline{x}(t-1/t-1) + \underline{k}(t)u(t);\ \underline{x}(-1/-1)=\underline{0} \tag{3.2.7}$$

$$u(t) = z(t) - \hat{y}(t) \tag{3.2.8}$$

$$\hat{y}(t) = \underline{c}'A\underline{x}(t-1/t-1) \tag{3.2.9}$$

$$\underline{k}(t)v(t) = [\bar{Q}(0)-AQ(t-1)A']\underline{c} \tag{3.2.10}$$

$$v(t) = r(0) - \underline{c}'AQ(t-1)A'\underline{c};\ v(0)=r(0) \tag{3.2.11}$$

$$Q(t) = AQ(t-1)A' + v(t)\underline{k}(t)\underline{k}'(t);\ Q(-1)=0 \tag{3.2.12}$$

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2\underline{h}(0)\underline{h}'(0) \tag{3.2.13}$$

$$r(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_u^2 \tag{3.2.14}$$

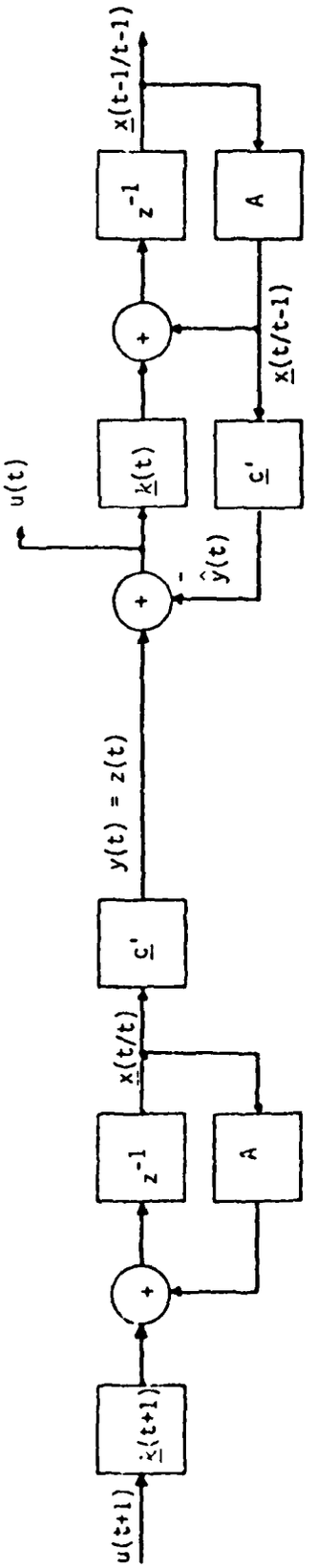


Fig. 3.2.3. The Kalman filter and signal model the Innovations rep. #2. Noise free case.

Noisy measurement process. With additive noise, the input to the Kalman filter is

$$z(t) = y(t) + n(t) \tag{3.1.15}$$

where n(t) is the measurement noise. The added noise only affects the variance of z(t)

$$\bar{r}(k) = E[z(t)z'(t+k)] = E[(y(t)+n(t))(y(t)+n(t))']$$
$$= r(k) + \sigma_n^2\delta(k) \tag{3.2.16}$$

and for k=0,

$$\bar{r}(0) = r(0) + \sigma_n^2 = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_n^2 \tag{3.2.17}$$

The defining equations for the Kalman filter are

$$\hat{\underline{x}}(t/t) = A\hat{\underline{x}}(t\text{-}1/t\text{-}1) + \hat{\underline{k}}(t)\hat{u}(t);\ \hat{\underline{x}}(-1/-1)=\underline{0} \tag{3.2.18}$$

$$\hat{u}(t) = z(t) - \hat{y}(t) \tag{3.2.19}$$

$$\hat{y}(t) = \underline{c}'A\hat{\underline{x}}(t\text{-}1/t\text{-}1) \tag{3.2.20}$$

$$\hat{\underline{k}}(t)\hat{v}(t) = [\bar{Q}(0) - A\hat{Q}(t\text{-}1)A']\underline{c} \tag{3.2.21}$$

$$\hat{v}(t) = \bar{r}(0) - \underline{c}'A\hat{Q}(t\text{-}1)A'\underline{c};\ \hat{v}(0)=\bar{r}(0) \tag{3.2.22}$$

$$\hat{Q}(t) = A\hat{Q}(t\text{-}1)A' + \hat{v}(t)\hat{\underline{k}}(t)\hat{\underline{k}}'(t);\ \hat{Q}(-1)=0 \tag{3.2.23}$$

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2\underline{h}(0)\underline{h}'(0) \tag{3.2.24}$$

$$\bar{r}(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_n^2 \tag{3.2.26}$$

Figure 3.2.4 shows the Kalman filter with the assumed signal model in the noisy case. Inverting the Kalman filter a new signal model is obtained. This signal model is driven by the innovations sequence of the Kalman filter and includes the measurement noise. Figure 3.2.5 shows this new model along with the Kalman filter. The Kalman filter reconstructs the innovations sequence of this new model with no error and estimates the state perfectly. Note however that the perfectly estimated state consists of imperfect estimates of the underlying sequence y(t). Table 3 lists the governing equations for the Kalman predictor and Kalman filter.

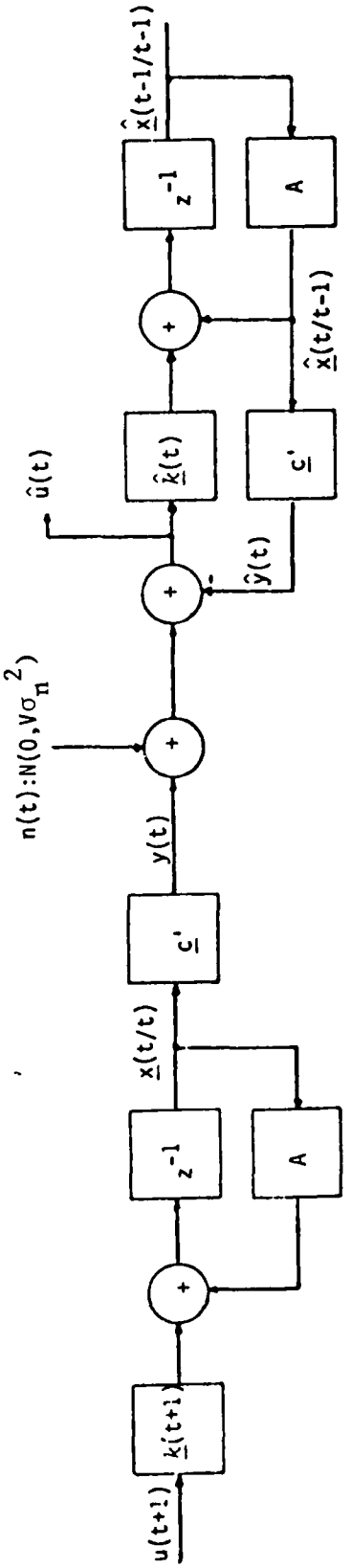


Fig. 3.2.4. The Kalman filter and signal model the Innovations rep. #2. Noisy case.

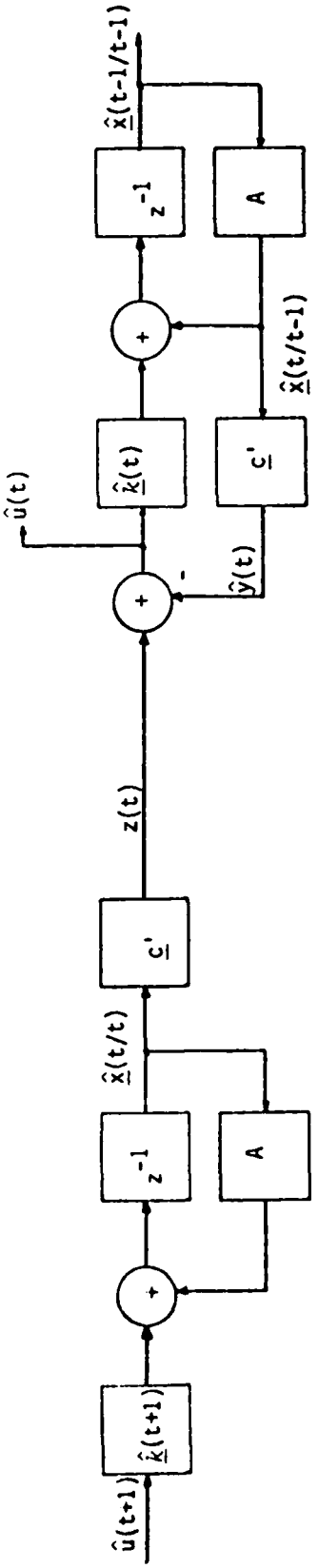


Fig. 3.2.5. The Kalman filter and Innovations rep. #2. Measurement noise included in the signal model.

Table 3. Governing equations for the Kalman predictor and Kalman filter.

The Kalman Predictor	The Kalman Filter
State Space Model	
$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$	$\underline{x}(t/t) = A\underline{x}(t-1/t-1) + \underline{k}(t)u(t)$
$u(t) = z(t) - \hat{y}(t)$	$u(t) = z(t) - \hat{y}(t)$
$\hat{y}(t) = \underline{c}'\underline{x}(t/t-1);\ \underline{x}(0/-1)=\underline{0}$	$\hat{y}(t) = \underline{c}'A\underline{x}(t-1/t-1);\underline{x}(-1/-1)=0$
Measurement Process	
$z(t) = y(t) + n(t)$	$z(t) = y(t) + n(t)$
$n(t);\ N(0,\sigma_n^2)$ r.v.s.	$n(t);\ N(0,\sigma_n^2)$ r.v.s.
The Kalman Gain	
$\underline{k}(t)v(t) = A[\bar{Q}(0) - Q(t)]\underline{c} + \sigma_u^2\underline{h}(1)$	$\underline{k}(t)v(t) = [\bar{Q}(0) - AQ(t-1)A']\underline{c}$
$v(t) = \bar{r}(0) - \underline{c}'Q(t)\underline{c};\ v(0) = \bar{r}(0)$	$v(t) = \bar{r}(0) - \underline{c}'AQ(t-1)A'\underline{c};\ v(0)=\bar{r}(0)$
$Q(t+1) = AQ(t)A' + v(t)\underline{k}(t)\underline{k}'(t);\ Q(0)=0$	$Q(t) = AQ(t-1)A' + v(t)\underline{k}(t)\underline{k}'(t);\ Q(-1)=0$
$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2\underline{h}(1)\underline{h}'(1)$	$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2\underline{h}(0)\underline{h}'(0)$
$\bar{r}(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_u^2 + \sigma_n^2 = r(0) + \sigma_n^2$	$\bar{r}(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_n^2 = r(0) + \sigma_n^2$
$\underline{k}'(t) = [k^t(0)\ k^t(1)\ \ldots\ k^t(n-1)]$	$A = \begin{bmatrix} 0 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & 1 & 0 & \cdot & \cdot & 0 \\ \cdot & & & \cdot & & & \cdot \\ \cdot & & & & \cdot & & \cdot \\ \cdot & & & & & & 1 \\ -a(n) & \cdot & \cdot & \cdot & & & -a(1) \end{bmatrix}$
$\underline{k}'(t) = [k^t(1)\ k^t(2)\ \ldots\ k^t(n)]$	
$\underline{c}' = (1\ 0\ \ldots\ 0)$	

3.3 The Kalman Gain Calculations

The main computational problem in implementing the Kalman predictor or filter is the calculation of the Kalman gain vectors, $\underline{k}(t)$ and $\underline{\bar{k}}(t)$. The most obvious approach is to use the equations derived in section 3.1 and 3.2. The governing equations for the predictor gain, $\underline{k}(t)$ are

$$\underline{k}(t)v(t) = A[\bar{Q}(0)-Q(t)]\underline{c}' + \sigma_u^2\underline{h}(1)$$

$$v(t) = \bar{r}(0) - \underline{c}'Q(t)\underline{c};\ v(0)=\bar{r}(0)$$

$$Q(t+1) = AQ(t)A' + v(t)\underline{k}(t)\underline{k}'(t);\ Q(0)=0$$

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2\underline{h}(1)\underline{h}'(1)$$

$$\bar{r}(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_u^2+\sigma_n^2 = r(0) + \sigma_n^2$$

On the other hand the governing equations for the filter gain, $\underline{\bar{k}}(t)$ are

$$\underline{\bar{k}}(t)v(t) = [\bar{Q}(0)-AQ(t-1)A']\underline{c}$$

$$v(t) = \bar{r}(0) - \underline{c}'AQ(t-1)A'\underline{c};\ v(0)=\bar{r}(0)$$

$$Q(t) = AQ(t-1)A' + v(t)\underline{\bar{k}}(t)\underline{\bar{k}}'(t);\ \bar{Q}(-1)=0$$

$$\bar{Q}(0) = A\bar{Q}(0)A' + \sigma_u^2\underline{h}(0)\underline{h}'(0)$$

$$\bar{r}(0) = \underline{c}'\bar{Q}(0)\underline{c} + \sigma_n^2 = r(0) + \sigma_n^2$$

These equations are of Ricatti type and are not very attractive from a computational point of view. A simpler and faster algorithm is desired.

Remember that the unit pulse response $h^t(i)$ of the two Innovarions representations can be written as

$$h^t(i) = \begin{cases} 0 & i<0 \\ h(0) & i=0 \\ c'A^{i-1}\underline{k}(t) & i>=1 \end{cases} \qquad (3.3.1)$$

or

$$h^t(i) = \begin{cases} 0 & i<0 \\ c'A^i\underline{k}(t) & i>=0 \end{cases} \qquad (3.3.2)$$

Define the vector $\underline{h}^t(i)$ as

$$\underline{h}^t(i)' = [h^t(i)\ h^t(i+1)\ \ldots\ h^t(i+n-1)] \qquad (3.3.3)$$

This gives the following fundamental relations:

$$\underline{k}(t) = \underline{h}^t(1) \qquad (3.3.4)$$

$$\underline{k}(t) = \underline{h}^t(0) \qquad (3.3.5)$$

All that is necessary to get $\underline{k}(t)$ and $\underline{k}(t)$ is to calculate the first n+1 of the time varying impulse responses for each t. This is shown in Gueguen and Scharf [1].

LeRoux and Gueguen [7] introduced an algorithm, the so-called Impulse Response Algorithm, to calculate the impulse response recursively. By using the fact that the underlying process is an ARMA(p,q) a substantial savings in calculations is possible as Dugre' [8] points out. This results in the so-called Fast Kalman Algorithm of Morf, Sidhu and Kailath [9], which they derived using Chandrasekar type equation for vector processes. The scalar version of this fast algorithm is the same as discussed by Pearlman [10].

What follows is a derivation of the Impulse Response Algorithm for the scalar case. This gives a fast way of computing Kalman gains and leads to the Fast Kalman Algorithm. I will follow the derivation of Dugre' [8] closely but extend it to cover both $\underline{k}(t)$ and $\underline{k}(t)$. He derived the algorithm only for the predictor gain vector, $\underline{k}(t)$. Having recursions for both the predictor and filter gain vectors bypasses the problem of requiring the inverse of A to exist to get the filter

gain vector $\underline{k}(t)$ as

$$\underline{k}(t) = A^{-1}\,\underline{k}(t)$$

Impulse response algorithm. The first n+1 values of the output y(t) can be written in matrix form as

$$Y(t) = K'(t)\,U(t) \tag{3.3.6}$$

where

$$X'(t) = [y(t)\ y(t-1)\ \ldots\ y(1)\ y(0)] \tag{3.3.7}$$

$$U'(t) = [u(t)\ u(t-1)\ \ldots\ u(1)\ u(0)] \tag{3.3.8}$$

and

$$K'(t) = \begin{bmatrix} h^{t}(0) & & & & & h'(t-1) & h^{0}(t) \\ & h^{t-1}(0) & & & & & \\ & & \ddots & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & h^{0}(1) & \\ & & & & & & h^{0}(0) \end{bmatrix} \tag{3.3.9}$$

Then the covariance matrix $R(t) = E[Y(t)Y'(t)]$ can be written as [11]

$$R(t) = K'(t)\ \mathrm{diag}[v(t)\ \ldots\ v(0)]K(t) \tag{3.3.10}$$

or

$$R(t)A(t) = K'(t)\mathrm{diag}[v(t)\ \ldots\ v(0)] \tag{3.3.11}$$

Here we define A(t) as follows:

$$A(t)=K^{-1}(t) = \begin{bmatrix} 1 & 0 & & & & & 0 \\ a^{t}(1) & 1 & & & & & \\ \vdots & & \ddots & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & 1 & 0 \\ a^{t}(t) & a^{t-1}(t-1) & \ldots & & \ldots & a'(1) & 1 \end{bmatrix} \tag{3.3.12}$$

Now the right hand term of (3.3.11) is

$$K'(t)\text{diag}[v(t) \ldots v(0)] =$$

$$\begin{bmatrix} s^t(0) & & & & & & s^1(t-1) & s^0(t) \\ \cdot & \cdot & & & & & & \cdot \\ \cdot & & \cdot & & & & & \cdot \\ \cdot & & & \cdot & & & & \cdot \\ \cdot & & & & \cdot & & & \cdot \\ \cdot & & & & & \cdot & & \cdot \\ \cdot & & & & & & \cdot & \cdot \\ \cdot & & & & & & & s^0(1) \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & s^0(0) \end{bmatrix} \tag{3.3.13}$$

where

$$s^t(i) = E[y(t+i)v'(t)] = h^t(i)v(t) \tag{3.3.14}$$

The term $s^t(i)$ is a scaled version of the impulse response $h^t(i)$. It is also the cross correlation between the input at t and the output at t+i. For $i<0$ $s^t(i)$ can be interpreted as an anticausal impulse response. In matrix form it satisfies

$$\begin{bmatrix} r(1+t) & \cdot & \cdot & \cdot & \cdot & r(1) \\ \cdot & \cdot & & & & \cdot \\ \cdot & & \cdot & & & \cdot \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & & \cdot & \cdot \\ r(t+1+t) & \cdot & \cdot & \cdot & \cdot & r(t+1) \end{bmatrix}$$

$$A(t) = \begin{bmatrix} s^t(-t-1) & & & s^1(-2) & s^0(-1) \\ \cdot & \cdot & & & s^0(-2) \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ s^t(-2t-2) & \cdot & \cdot & \cdot & s^0(-t-1) \end{bmatrix} \tag{3.3.15}$$

For example

$$s^{k-1}(-k) = r(k)+r(k-1)a^{k-1}(1) + \ldots r(1)a^{k-1}(k-1) \tag{3.3.16}$$

By inspecting equation (3.3.11) it follows that

$$s^t(k) = r(k) + r(k+1)a^t(1) + \ldots + r(k+t)a^t(t) \tag{3.3.17}$$

and

$$s^{t+1}(k) = r(k) + r(k+1)a^{t+1}(1) + \ldots + r(k+t+1)a^{t+1}(t+1) \quad (3.3.18)$$

What we need is a recursion for $s^t(k)$.

Levison-Durbin algorithm. The Levinson-Durbin algorithm gives a recursion for $a^t(i)$ [12], [13]:

$$a^{t+1}(i) = a^t(i) + \lambda(t+1)a^t(t+1-i) \quad i=1,2,\ldots,t \quad (3.3.19)$$

$$\lambda(t+1) = -[r(t+1)+a^t(1)r(t)+\ldots+a^t(t)r(1)]/v(t) \quad (3.3.20)$$

$$v(t+1) = v(t)[1 - \lambda^2(t+1)] \quad (3.3.21)$$

$$a^{t+1}(t+1) = \lambda(t+1) \text{ and } a^{t+1}(0) = a^t(0) = 1$$

We can substitute for $a^{t+1}(k)$ in equation (3.3.18) to obtain

$$s^{t+1}(k) = (r(k)+r(k+1)a^t(1)+\ldots+r(k+t)a^t(t)) + \lambda(t+1)[r(k+1)a^t(t)+r(k+2)a^t(t-1)+\ldots+r(k+t)a^t + r(k+t+q)] = s^t(k) + \lambda(t+1)s^t(-t-1-k) \quad (3.3.22)$$

Now we need a recursion for $s^t(-t-1-k)$. It follows from inspection of equation (3.3.15). We write

$$s^{t+1}(-t-1-k)=r(t+1+k)+r(t+k)a^{t+1}(1)+r(t+k-1)a^{t+1}(2)+\ldots+r(k)a^{t+1}(t+1) = [r(t+1+k)+r(t+k)a^t(1)+\ldots+r(k+1)a^t(t)] + \lambda(t+1)[r(k+t)a^t(t)+\ldots+r(k+1)a^t(1)+r(k)] = s^t(-t-1-k) + \lambda(t+1)s^t(k) \quad (3.3.23)$$

Here we have substituted $a^t(k)+\lambda(t+1)a^t(t+1-k)$ for $a^{t+1}(k)$ as before. Now we can write the recursion for $s^t(k)$ as

$$s^{t+1}(k) = s^t(k) + \lambda(t+1)s^t(-t-1-k) \quad (3.3.24)$$

$$s^{t+1}(-t-1-k) = s^t(-t-1-k) + \lambda(t+1)s^t(k) \quad (3.3.25)$$

$$\lambda(t+1) = -s^t(-t-1)/s^t(0) \quad (3.3.26)$$

with initial conditions

$$s^0(k) = r(k) \quad \forall\ k \quad \text{and} \quad \lambda(0) = -r(1)/r(0)$$

This is the so-called Impulse Response Algorithm [7]. By substituting $s^t(0) = v(t)$ and $s^t(k) = h^t(k)v(t)$, and adding on the recursion for $a^t(k)$ from the Levinson-Durbin Algorithm, we get a simultaneous recursion for $h^t(k)$ and $a^t(k)$ as:

$$h^{t+1}(k) = [h^t(k) + \lambda(t+1)h^t(-t-1-k)]/(1 - \lambda^2(t+1)) \quad (3.3.27)$$

$$h^{t+1}(-t-1-k)=[h^t(-t-1-k)+\lambda(t+1)h^t(-t-1-k)]/(1-\lambda^2(t+1)) \quad (3.3.28)$$

$$\lambda(t+1) = -h^t(-t-1)v(t)/v(t) = -h^t(-t-1) \quad (3.3.29)$$

$$a^{t+1}(k) = a^t(k) + \lambda(t+1)a^t(t+1-k), \quad k=1,2,\ldots, t \quad (3.3.30)$$

$$a^{t+1}(t+1) = \lambda(t+1) \quad (3.3.31)$$

with initial conditions $h^0(k)=r(k)/r(0)$ and $\lambda(0)=-r(1)/r(0)$.

Note that we can write $s^{t+1}(0)$ as

$$s^{t+1}(0) = s^t(0)(1 - \lambda^2(t+1)) \quad (3.3.32)$$

because of equation (3.3.21) and $s^t(0)=v(t)$.

An obvious drawback in using the Impulse Response Algorithm is that one must know t ahead of time to calculate $s^t(-t-1-k)$ [8]. This can be seen in the following way. Write $s^t(-t-1-k)$ as

$$\begin{aligned} s^t(-t-1-k) &= s^{t-1}(-t-1-k) + \lambda(t)s^{t-1}(k) \\ &= s^{t-2}(-t-1-k) + \lambda(t-1)s^{t-2}(k) + \lambda(t)s^{t-1}(k) \\ &= s^0(-t-1-k) + \sum_{i=0}^{t-1} \lambda(t-i)s^{t-1-i}(k) \end{aligned}$$

It is clear that one must start with $s^0(-t-1-k)$ to get $s^t(-t-1-k)$. This depends on t. In particular for k=n one must start with $s^0(-t-1-n)$ to calculate $s^t(-t-1-n)$ where $n=\max(p,q)$. This drawback can be corrected by noting the following:

$$[r(n+1+t)\ r(n+t)\ \ldots\ r(n+1)] =$$

$$[-a(n)\ \ldots\ -a(2)\ -a(1)] \begin{bmatrix} r(t+1) & . & . & . & r(1) \\ & . & & & . \\ & . & & & . \\ r(t+n) & . & . & . & r(n) \end{bmatrix}$$

Multiplying both sides by $[1\ a^t(1)\ \ldots\ a^t(t)]'$ results in

$$s^t(-t-1-n) = -[a(n)\ \ldots\ a(1)][s^t(-t-1)\ \ldots\ s^t(-t-n)]' \qquad (3.3.33)$$

So there is no need to go all the way down to $s^0(-t-1-n)$ to get $s^t(-t-1-n)$. One can stop at $s^0(-n-1)$ and proceed recursively [8].

Fast Kalman algorithm. Using equation (3.3.24) we write

$$\begin{bmatrix} s^{t+1}(0) \\ s^{t+1}(1) \\ . \\ s^{t+1}(n-1) \end{bmatrix} = \begin{bmatrix} s^t(0) \\ s^t(1) \\ . \\ s^t(n-1) \end{bmatrix} - \lambda(t+1) \begin{bmatrix} s^t(-t-1) \\ s^t(-t-2) \\ . \\ s^t(-t-n) \end{bmatrix}$$

and

$$\begin{bmatrix} s^{t+1}(1) \\ s^{t+1}(2) \\ . \\ s^{t+1}(n) \end{bmatrix} = \begin{bmatrix} s^t(1) \\ s^t(2) \\ . \\ s^t(n) \end{bmatrix} - \lambda(t+1) \begin{bmatrix} s^t(-t-2) \\ s^t(-t-3) \\ . \\ s^t(-t-n-1) \end{bmatrix}$$

In matrix form these are

$$\underline{K}^{t+1}(0) = \underline{K}^t(0) - (\underline{c}'\underline{L}^t/v(t))\underline{L}^t$$

$$\underline{K}^{t+1}(1) = \underline{K}^t(1) - (c'\underline{L}^t/v(t))A\ \underline{L}^t$$

where

$$\underline{K}^t(0)' = [s^t(0)s^t(1)\ \ldots\ s^t(n-1)]$$

$$\underline{K}^t(1)' = [s^t(1)s^t(2)\ \ldots\ s^t(n)]$$

$$(L^t)' = [s^t(-t-1)s^t(-t-2)\ \ldots\ s^t(-t-n)]$$

$$(A\ \underline{L}^t)' = [s^t(-t-2)s^t(-t-3)\ \ldots\ s^t(-t-n-1)]$$

$$s^t(-t-1) = \underline{c}'\underline{L}^t \text{ and } s^t(0) = v(t)$$

All we need now is a recursion for $\underline{L}^t$. Using equation (3.3.25) we write

$$\begin{bmatrix} s^{t+1}(-t-2) \\ s^{t+1}(-t-3) \\ \vdots \\ s^{t+1}(-t-n-1) \end{bmatrix} = \begin{bmatrix} s^t(-t-2) \\ s^t(-t-3) \\ \vdots \\ s^t(-t-n-1) \end{bmatrix} - (\underline{c}'\underline{L}^t/v(t)) \begin{bmatrix} s^t(1) \\ s^t(2) \\ \vdots \\ s^t(n) \end{bmatrix}$$

or

$$\underline{L}^{t+1} = A\underline{L}^t - (\underline{c}'\underline{L}^t/v(t))\underline{K}^t(1)$$
$$= A[\underline{L}^t - (\underline{c}'\underline{L}^t/v(t))\underline{K}^t(0)]$$

Here we have used the fact that

$$\underline{K}^t(1) = v(t)\underline{k}(t) = v(t)A\underline{k}(t) = A\underline{K}^t(0)$$

Summary of Recursions for $K^t(0)$ and $K^t(1)$

For $\underline{K}^t(0)$:

$$\underline{K}^{t+1}(0) = \underline{K}^t(0) - (\underline{c}'\underline{L}^t/v(t))\underline{L}^t \tag{3.3.34}$$

$$\underline{L}^{t+1} = A\,[\underline{L}^t - (\underline{c}'\underline{L}^t/v(t))]\underline{K}^t(0) \tag{3.3.35}$$

$$v(t+1) = v(t)[1 - (\underline{c}'\underline{L}^t/v(t))^2] \tag{3.3.36}$$

Initial conditions:

$$\underline{K}^0(0)' = [r(0)\ r(1)\ \ldots\ r(n-1)] \tag{3.3.37}$$

$$\underline{L}^0(0)' = [r(1)\ r(2)\ \ldots\ r(n)] \tag{3.3.38}$$

$$v(0) = r(0) \tag{3.3.39}$$

For $\underline{K}^t(1)$:

$$\underline{K}^{t+1}(1) = \underline{K}^t(1) - (\underline{c}'\underline{L}^t/v(t))A\underline{L}^t \tag{3.3.40}$$

$$\underline{L}^{t+1} = A\underline{L}^t - (\underline{c}'\underline{L}/v(t))\underline{K}^t(1) \tag{3.3.41}$$

$$v(t+1) = v(t)\,[1 - (\underline{c}'\underline{L}^t/v(t))^2] \tag{3.3.42}$$

Initial conditions:

$$\underline{K}^0(0)' = \underline{L}^0(0)' = [r(1)\ r(2)\ \ldots\ r(n)] \tag{3.3.43}$$

$$v(0) = r(0) \tag{3.3.44}$$

The Fast Kalman algorithm follows by substituting $v(t)\underline{k}(t)$ for $\underline{K}^t(0)$, $v(t)\underline{k}(t)$ for $\underline{K}^t(1)$ and $v(t)\underline{l}(t)$ for $\underline{L}^t$ and solving for $\underline{k}(t)$ and $\underline{k}(t)$.

The Filter Gain Vector, $k(t)$

$$\underline{k}(t+1) = [\underline{k}(t) - (\underline{c}'\underline{l}(t))\underline{l}(t)]/d(t) \tag{3.3.45}$$

$$\underline{l}(t+1) = A[\underline{l}(t) - (\underline{c}'\underline{l}(t))\underline{k}(t)]/d(t) \tag{3.3.46}$$

$$d(t) = 1 - (\underline{c}'\underline{l}(t))^2 \tag{3.3.47}$$

$$v(t+1) = v(t)d(t) \tag{3.3.48}$$

Initial conditions are

$$\underline{k}'(0) = [r(0)\ r(1)\ \ldots\ r(n-1)]/v(0) \tag{3.3.49}$$

$$\underline{l}'(0) = [r(1)\ r(2)\ \ldots\ r(n[]/v(0) \tag{3.3.50}$$

$$v(0) = r(0) \tag{3.3.51}$$

The Predictor Gain Vector, $k(t)$

$$\underline{k}(t+1) = [\underline{k}(t) - (\underline{c}'\underline{l}(t))A\ \underline{l}(t)]/d(t) \tag{3.3.52}$$

$$\underline{l}(t+1) = [A\ \underline{l}(t) - (\underline{c}'\underline{l}(t))\underline{k}(t)]/d(t) \tag{3.3.53}$$

$$d(t) = 1 - (\underline{c}'\underline{l}(t))^2 \tag{3.3.54}$$

$$v(t+1) = v(t)d(t) \tag{3.3.55}$$

Initial conditions are

$$\underline{k}'(0) = \underline{l}'(0) = [r(1)\ r(2)\ \ldots\ r(n)]/v(0) \tag{3.3.56}$$

$$v(0) = r(0) \tag{3.3.57}$$

In the case of noisy measurement process then we must use

$$v(0) = \bar{r}(0) = r(0) + \sigma_n^2 \tag{3.3.58}$$

instead of $v(0)=r(0)$. Nothing else changes.

3.4 Properties of the Kalman Predictor/Filter

In section 3.1 and 3.2 we derived the Kalman predictor and filter as inverses of the Innovations representations #1 and #2, respectively. It should be no surprise that the Kalman predictor and filter share many of the same properties as their corresponding signal models. In this section we will establish the relation between the state variance of the Kalman predictor/filter and the state variance of the corresponding signal models. We also show that the two time-varying signal models are asymptotically time-invariant, converging to the Markovian representations #1 and #2, respectively. The Kalman predictor and filter converge to their corresponding time invariant Markovian predictor and filter structures.

Consider first the Innovations representation #1

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$$

$$y(t) = \underline{c}'\underline{x}(t/t-1) + u(t)$$

and its corresponding Kalman predictor

$$\hat{\underline{x}}(t+1/t) = A\hat{\underline{x}}(t/t-1) + \hat{\underline{k}}(t)[z(t)-\hat{y}(t)]$$

$$\hat{y}(t) = \underline{c}'\hat{\underline{x}}(t/t-1)$$

$$z(t) = y(t) + n(t)$$

The state variance of the signal model Q(t) satisfies

$$Q(t+1) = A\ Q(t)\ A' + v(t)\underline{k}(t)\underline{k}(t);\ Q(0)=0$$

This can be written as

$$Q(t)=E[\underline{x}(t/t-1)\underline{x}'(t/t-1)] =$$

$$E[(\underline{x}(t/t-1)-\hat{\underline{x}}(t/t-1)+\hat{\underline{x}}(t/t-1))(\underline{x}(t/t-1)-\hat{\underline{x}}(t/t-1)+\hat{\underline{x}}(t/t-1))']$$

$$=E[(\underline{x}(t/t-1)-\hat{\underline{x}}(t/t-1))(\underline{x}(t/t-1)-\hat{\underline{x}}(t/t-1))']+E[\hat{\underline{x}}(t/t-1)\hat{\underline{x}}'(t/t-1)]$$

$$= E(t) + \hat{Q}(t) \qquad (3.4.1)$$

where $\hat{Q}(t)$ is the state variance of the Kalman predictor

$$\hat{Q}(t+1) = A\hat{Q}(t)A' + \hat{v}(t)\underline{\hat{k}}(t)\underline{\hat{k}}'(t);\ \hat{Q}(0)=0$$

and $E(t)$ is the error variance. It follows that both $E(t)$ and $\hat{Q}(t)$ are bounded by $Q(t)$ [2]

$$0 <= \hat{Q}(t) <= Q(t) \tag{3.4.2}$$

$$0 <= E(t) <= Q(t) \tag{3.4.3}$$

The state variance of the Kalman predictor $\hat{Q}(t)$ is independent of the particular signal model used, depending only on the covariance of the measurement process $z(t)$. This is so because the Kalman gain vector $\underline{k}(t)$ and $v(t)$ depend only on the covariance of the measurement process, $\bar{r}(k) = r(k) + \sigma_n^2\delta(k)$ [2]. On the other hand the error variance is dependent on the particular signal model and on the noise variance σ_n^2 of the measurement process. In the case $n(t)=0$ the Kalman predictor estimates the state of the Innovations representation #1 with no error and then

$$\underline{\hat{x}}(t/t-1) = \underline{x}(t/t-1) \tag{3.4.4}$$

$$E(t) = 0 \tag{3.4.5}$$

$$\hat{Q}(t) = Q(t) \tag{3.4.6}$$

Increasing the noise variance of the measurement process results in a smaller state variance $\hat{Q}(t)$ and larger error variance for fixed state variance, $Q(t)$, of the signal model.

The Innovations representation #2 is described by

$$\underline{x}(t+1/t+1) = A\underline{x}(t/t) + \underline{k}(t+1)u(t+1)$$

$$y(t) = c'\underline{x}(t/t)$$

and its corresponding Kalman filter

$$\underline{\hat{x}}(t/t) = A\underline{\hat{x}}(t-1/t-1) + \underline{\hat{k}}(t)[z(t)-\hat{y}(t)]$$

$$\hat{y}(t) = \underline{c}'A\hat{\underline{x}}(t-1/t-1)$$

$$z(t) = y(t) + n(t)$$

Proceeding in a same way as we did before the state variance of the Innovations representation #2 satisfies

$$Q(t) = AQ(t-1)A' + v(t)\,\underline{k}(t)\underline{k}'(t);\ Q(-1)=0$$

This can be written as

$$Q(t)=E[(\underline{x}(t/t)-\hat{\underline{x}}(t/t)+\hat{\underline{x}}(t/t))(\underline{x}(t/t)-\hat{\underline{x}}(t/t)+\hat{\underline{x}}(t/t))'] =$$

$$E[(\underline{x}(t/t)-\hat{\underline{x}}(t/t))(\underline{x}(t/t)-\hat{\underline{x}}(t/t))']+E[\hat{\underline{x}}(t/t)\hat{\underline{x}}'(t/t)]$$

$$= E(t) + \hat{Q}(t) \qquad (3.4.7)$$

Here $Q(t)$ is the Kalman filter state variance:

$$Q(t) = AQ(t-1)A' + \hat{v}(t)\hat{\underline{k}}(t)\hat{\underline{k}}'(t);\ Q(-1)=0$$

and $E(t)$ is the error variance. Both the error and the filter state variance are bounded by the state variance of the signal model Q(t):

$$0 = E(t) = Q(t) \qquad (3.4.8)$$

$$0 = Q(t) = Q(t) \qquad (3.4.9)$$

In case we have n(t)=0 the Kalman filter estimates the state of the Innovations representation #2 with no error and we have

$$\hat{\underline{x}}(t/t) = \underline{x}(t/t) \qquad (3.4.10)$$

$$E(t) = 0 \qquad (3.4.11)$$

$$\hat{Q}(t) = Q(t) \qquad (3.4.12)$$

As we mentioned before the two Innovations representations and hence the Kalman predictor and filter are time-varying but asymptotically time invariant. For the case n(t)=0 the Kalman predictor estimates the state of the Innovations representation #1 perfectly and so does the Kalman filter for the Innovations representation #2. As a result we have

$$\hat{Q}(t) = Q(t)$$

$$\hat{\mathcal{Q}}(t) = \mathcal{Q}(t)$$

Assume that the Kalman predictor is estimating the state of the Markovian representation #1 (n(t)=0). We have then

$$\bar{Q}(0) = > E(t) + Q(t);\ E(0)=\bar{Q}(0) \tag{3.4.13}$$

Anderson and Moore [2] show that the state variance, Q(t) is monotone and converges to a limit Q(∞) since it is bounded by $\bar{Q}(0)$. We need to show that this limit is $\bar{Q}(0)$. If Q(t) converges to a limit so does $\mathcal{Q}(t)$ of the Innovations representation #2 since $\mathcal{Q}(t)=AQ(t-1)A'$. Now the output covariance of the Innovations representation #2 at time t=∞ is

$$r(k) = \underline{c}'A^k\mathcal{Q}(\infty)\underline{c}$$

and if it is to match that of the Markovian representation #2 then

$$r(k)=\underline{c}'A^k\mathcal{Q}(\infty)\underline{c} = r(k)=\underline{c}'A^k\bar{\mathcal{Q}}(0)\underline{c}$$

and it must hold that

$$\mathcal{Q}(\infty) = \bar{\mathcal{Q}}(0) \tag{3.4.14}$$

As a consequence since $\bar{Q}(0)=A*\bar{\mathcal{Q}}(0)*A'$ it follows that

$$Q(\infty) = \bar{Q}(0) \tag{3.4.15}$$

Also

$$\lim_{t\to\infty} v(t) = \sigma_n^2 \tag{3.4.16}$$

$$\lim_{t\to\infty} \underline{k}(t) = \underline{h}(0) \tag{3.4.17}$$

$$\lim_{t\to\infty} \underline{k}(t) = \underline{h}(1) \tag{3.4.18}$$

The main point is this. The Innovations representations are both asymptotically time-invariant and converge to the Markovian representation #1 and #2, respectively. For the case n(t)=0 the Kalman predictor and filter converge to the Markovian predictor and filter

structure. Figures 3.4.1 and 3.4.2 show the Markovian predictor and filter structure, respectively.

In the special case the process is an AR(p) instead of an ARMA(p,q) then the signal models converge in only p steps (Pearlman [10]). The same thing holds for the Kalman filter and predictor. The reason for this is that after p steps the Impulse Response Algorithm (see section 3.3) has reached steady state:

$$h^{p+1}(i) = h^{p}(i) \qquad i=0,1, \ldots, p$$

$$v(p+1) = v(p)$$

since

$$\lambda(p+1) = -h^{p}(-p-1) = a^{p+1}(p+1) = 0$$

As a consequence

$$\underline{k}(p+1) = \underline{k}(p)$$

$$\underline{k}(p+1) = \underline{k}(p)$$

and

$$Q(p+1) = Q(p)$$

$$Q(p+1) = Q(p)$$

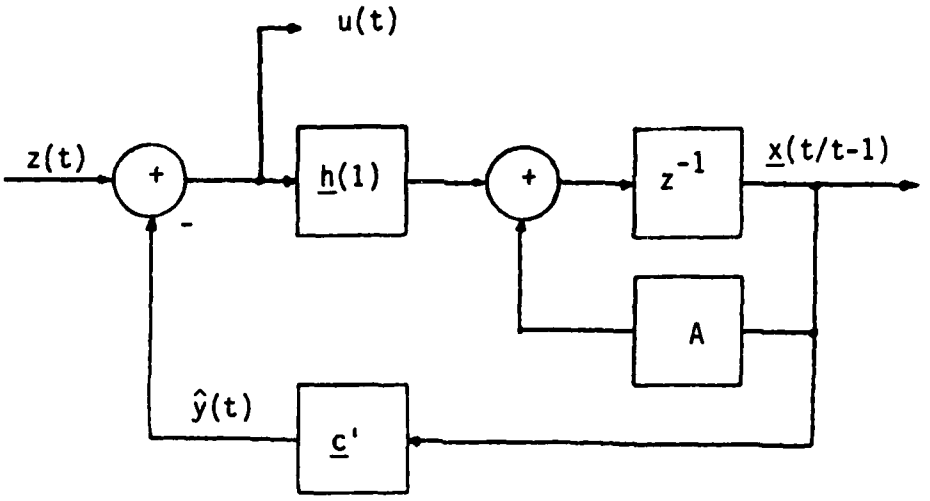


Fig. 3.4.1. The Markovian predictor.

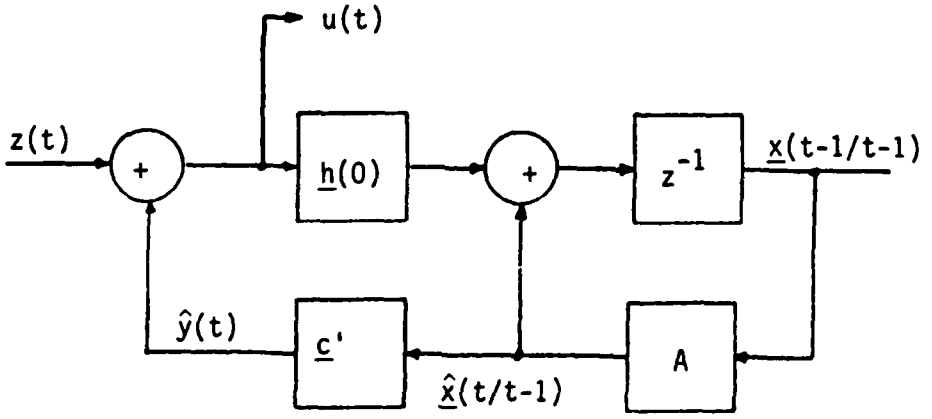


Fig. 3.4.2. The Markovian filter.

REFERENCES FOR CHAPTER 3

1. C. Gueguen and L.L. Scharf, "Exact Maximum Likelihood Identification of ARMA Models; A Signal Processing Perspective," Proc. EUSIPCO, Lausanne, Switzerland, Sept. 1980.

2. B.D.O. Anderson and J.B. Moore, Optimal Filtering, Prentice-Hall Inf. Syst. Series, 1979.

3. J.S. Meditch, Stochastic Optimal Linear Estimation and Control, McGraw-Hill Book Company, NY, 1969.

4. A. Gelb, Applied Optimal Estimation, The MIT Press, 1974.

5. A.P. Sage and J.L. Melsa, Estimation Theory with Applications to Communications and Control, McGraw-Hill Book Company, NY, 1971.

6. J. Makhoul, "Linear Prediction: A Tutorial Review," Proc. of IEEE, Vol. 63, No. 4, April 1975.

7. J. LeRoux and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients," Trans. Acoust., Speech, and Signal Processing, Vol. 25, No. 2, pp. 257-259, June 1977.

8. J.P. Dugre', "Parametric Spectrum Analysis of Stationary Random Sequences," Ph.D. Dissertation, Colorado State University, Summer 1981.

9. M. Morf, G.S. Sidhu and T. Kailath, "Some new Algorithms for Recursive Estimation on Constant Linear Discrete Time Systems," IEEE Trans. on Automat. Cont., AC-19, August 1974.

10. J.G. Pearlman, "An Algorithm for exact likelihood of high-order autoregressive-moving average process," Biometrica, 67, 1, 1980.

11. L.L. Scharf, EE612; Estimation Theory, Class Notes, Colorado State University, 1981.

12. N. Levinson, "The Wiener RMS (root-mean-square) Error Criterion in Filter Design and Prediction," J. Math. Phys., Vol. 25, January 1947.

13. J. Durbin, "Fitting of Time Series Models," Rev. Int. Stat. Inst., Vol. 28, 1960.

CHAPTER 4

DIGITAL IMPLEMENTATION

In this chapter we address some of the problems encountered in implementing the Kalman predictor and Kalman filter using fixed point arithmetic. The finite wordlength effects are (1) overflow, (2) round-off, and (3) input and coefficient quantization. A great deal of work has been done to study these effects and others in time invariant digital filters. For a review see for example references [9] and [10]. Scaling problems are treated in [3] and [4] and scaling of state space digital filters is covered in [1] and [2]. Roundoff noise in state space digital filters is covered in references [1], [5], [6], and [7]. Issues in digital implementation of control compensators have been treated in [8] and [8]. Many of the results of these studies can be used the Kalman predictor/filter problem.

In section 4.1 we look at the scaling problem. Section 4.2 deals with the effects of roundoff on the performance of the Kalman predictor and filter. Then in section 4.3 we examine the computational complexity of the predictor/filter in terms number of arithmetic operations.

4.1 Scaling of Variables

In this section we show how to scale each element in the state vectors of the Kalman predictor and filter to limit the probability

overflow. The scaling rule used will be the so-called l_2 scaling rule of Jackson [3], [4].

The set of numbers that can be represented using a fixed word-length of m-bits constitutes the dynamic range of variables. The variables take on discrete values

$$i\varepsilon, \quad i=0,+/-1, \ldots, +/-2^{m-2} \tag{4.1.1}$$

where ε is the quantization step size. The dynamic range is therefore bounded by

$$-\varepsilon 2^{m-1} < \varepsilon i < +\varepsilon 2^{m-1} \qquad \forall\ i \tag{4.1.2}$$

If the bounds are forced to be +/-1 then the step size must be

$$\varepsilon = 2^{-(m-1)} \tag{4.1.3}$$

For m = 16 (as in a 16-bit processor) the step size is

$$\varepsilon = 2^{-15} = 3.05176*10^{-5} \tag{4.1.4}$$

A filter is said to be l_2 scaled if for each variable

$$\delta^2\sigma_u^2\ ||f_k||^2 = \delta^2\sigma_u^2 \sum_{i=1}^{\infty} f_k(i)^2 = [\varepsilon 2^{m-1}]^2 \tag{4.1.5}$$

Here $f_k(i)$ is the response of the k^{th} variable to a unit pulse sequence applied at the input. σ_u^2 is the variance of the signal applied at the input. $\sigma_u^2\ ||f_k||^2$ is the variance of the k^{th} filter state variable when the input is an i.i.d. r.v. of variance σ_u^2. The parameter δ determines the probability of overflow. Increasing δ decreases the probability of overflow.

Scaling of Variables in the Kalman Predictor

The state space equations for the Kalman predictor are

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$$

$$y(t) = \underline{c}'\underline{x}(t/t-1)$$

$$u(t) = z(t) - y(t)$$

Figure 4.1.1 shows a block diagram of the Kalman predictor. To scale the system we introduce a diagonal scaling matrix S

$$S = \begin{bmatrix} s(1) & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & s(n) \end{bmatrix} \tag{4.1.6}$$

and perform the transformation

$$\underline{x}(t/t-1)=S^{-1}\,\underline{x}(t/t-1) \tag{4.1.7}$$

The scaled Kalman predictor is then

$$\underline{x}(t+1/t) = (S^{-1}\,AS)\,\underline{x}(t/t-1) + (S^{-1}\underline{k}(t))u(t) \tag{4.1.8}$$

$$y(t) = (\underline{c}'S)\underline{x}(t/t-1) \tag{4.1.9}$$

Figure 4.1.2 shows the scaled Kalman predictor. The l_2 scaling rule for the Kalman predictor is

$$\delta^2\, S^{-1}\,[Q(t)]_{kk}\,S^{-T} = (\varepsilon 2^{m-1})^2 \quad k=1,2,\ldots,n \tag{4.1.10}$$

Here Q(t) is the state variance of the Kalman predictor

$$Q(t) = v(t)\sum_{i=1}^{t}(A^{i-1}\underline{k}(t-i))(A^{i-1}\underline{k}(t-i))' \tag{4.1.11}$$

These are time-varying scaling constraints and not very practical. A better solution is to find the worst case scaling constraints and use them. The state variance is monotonically increasing and converges in the limit Q(∞)

$$\lim_{t\to\infty} Q(t) = Q(\infty)$$

So the worst case scaling constraints are then

$$\delta^2\, S^{-1}[Q(\infty)]_{kk}\,S^{-T} = (\varepsilon 2^{m-1})^2 \quad k=1,2,\ldots,n \tag{4.1.12}$$

which can be written as

$$\delta^2\, s^{-2}(k)q(k,k) = (\varepsilon 2^{m-1})^2 \quad k=1,2,\ldots n \tag{4.1.13}$$

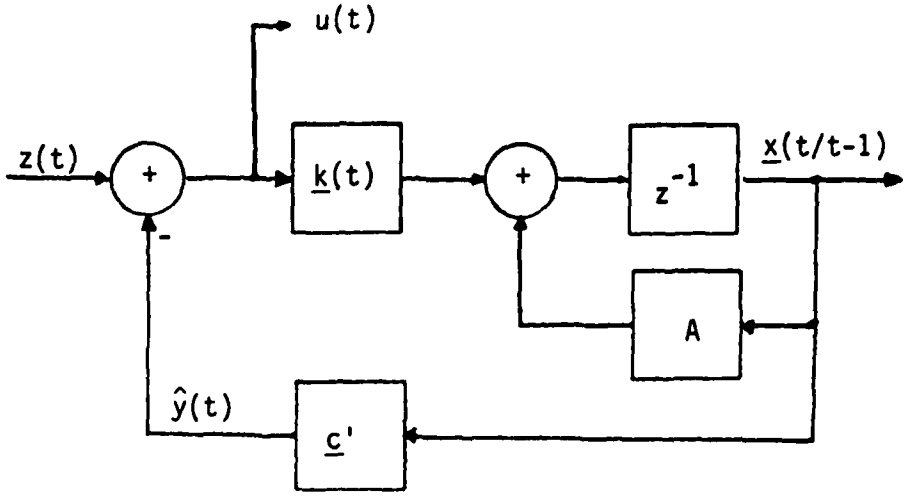


Fig. 4.1.1. The Kalman predictor.

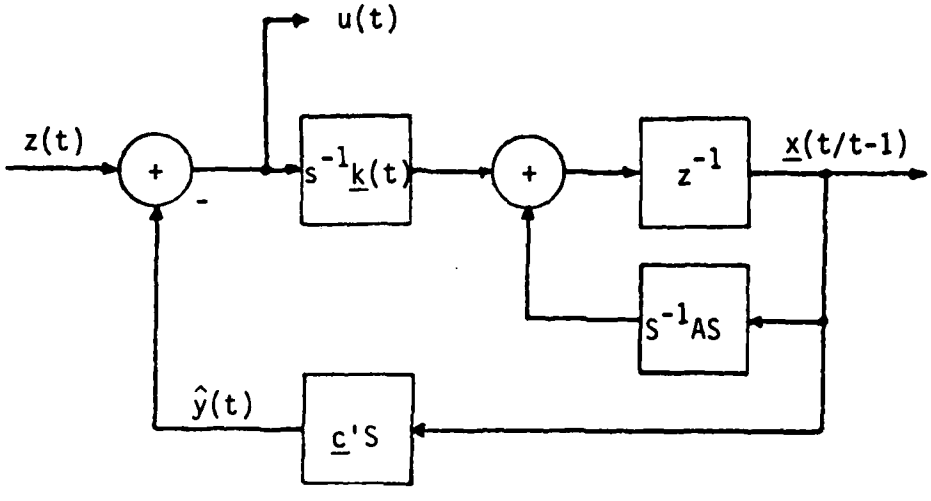


Fig. 4.1.2. The scaled Kalman predictor.

where $q(k,k) = [Q(\infty)]_{kk}$ and $s^{-2}(k) = [S^{-2}]_{kk}$. The scale factors are determined by

$$s(k) = \delta \frac{\sqrt{q(k,k)}}{(\varepsilon 2^{m-1})} \qquad k=1,2, \ldots, n \tag{4.1.14}$$

This is the same result obtained by Mullis and Roberts [1]. In the case $(\varepsilon 2^{m-1}) = 1$ the scale factors are chosen

$$s(k) = \delta\sqrt{q(k,k)} \qquad k=1,2,3, \ldots, n \tag{4.1.15}$$

This means the variance of each scaled state element must be less than one assuming $\delta > 1$. A nice feature of this scaling procedure is that it allows us to map the dynamic range of each state element to the range of the numerical representation. The price is an increased number of multiplications. A simpler procedure is to choose all scale factors equal to s

$$s = \max_k(s(k)) \tag{4.1.16}$$

Then the scaled Kalman predictor becomes

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + (1/s)\underline{k}(t)u(t) \tag{4.1.17}$$

$$y(t) = s\underline{c}'\underline{x}(t/t-1) \tag{4.1.18}$$

and only (n+1) multiplications are required for scaling. This is not a bad idea if all state elements have variance of the same order of magnitude. If not, it is possible that some state variables are underflowing. By moving the scale factor out in front of the summation node then only one multiplication is needed for scaling. Figure 4.1.3a and 4.1.3b show the Kalman predictor using this simplified scaling procedure.

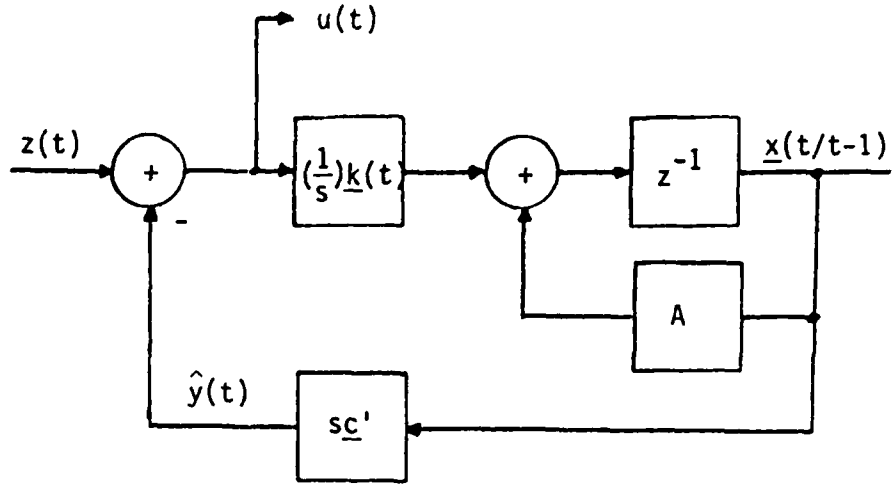


Fig. 4.1.3a. The scaled Kalman predictor. Simplified scaling procedure.

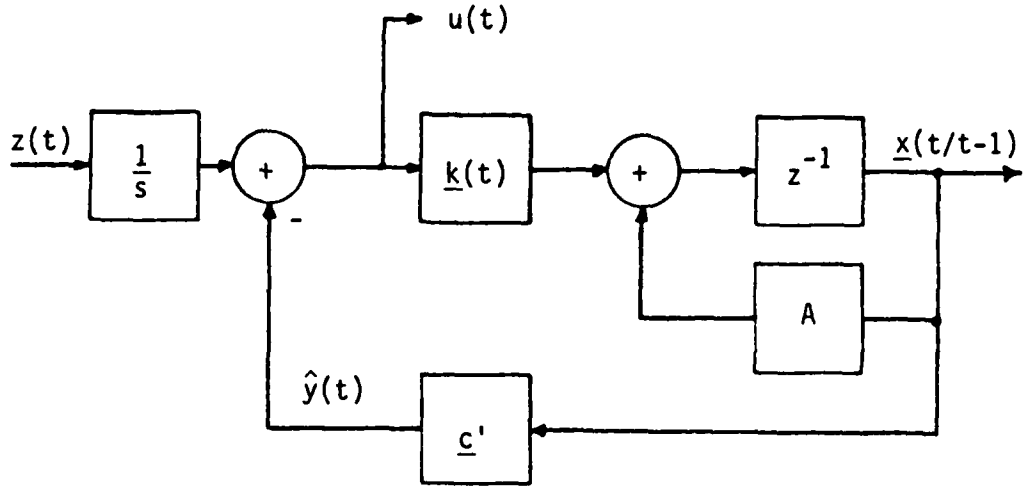


Fig. 4.1.3b. The scaled Kalman predictor. Simplified scaling procedure.

Scaling of Variables in the Kalman Filter

The Kalman filter is described by the state space equations

$$\underline{x}(t/t) = A\underline{x}(t-1/t-1) + \underline{k}(t)u(t)$$

$$y(t) = \underline{c}'A\underline{x}(t-1/t-1)$$

$$u(t) = z(t) - y(t)$$

Figure 4.1.4 shows a block diagram of the Kalman filter. With the scaling matrix S defined as before the scaled Kalman filter is

$$\underline{x}(t/t) = (S^{-1}AS)\underline{x}(t-1/t-1) + (S^{-1}\underline{k}(t))u(t) \qquad (4.1.19)$$

$$y(t) = (\underline{c}'AS)\underline{x}(t-1/t-1) \qquad (4.1.20)$$

Using similar arguments as for the Kalman predictor we write down the l_2 scaling constraints for the filter.

$$\delta^2 s^{-2}(k)q(k,k) = (\varepsilon 2^{m-1})^2 \qquad (4.1.21)$$

The scale factors are then

$$s(k) = \frac{\delta\sqrt{q(k,k)}}{(\varepsilon 2^{m-1})} \qquad k=1,2, \ldots, n \qquad (4.1.22)$$

Here

$$q(k,k) = [Q(\infty)]_{kk} \qquad (4.1.23)$$

where $Q(\infty)$ is the state variance of the Kalman filter in steady state. Figure 4.1.5 shows the scaled Kalman filter. The scaling procedure is simplified when all elements in the matrix S are made equal to s

$$s = \max_k(s(k)) \qquad (4.1.24)$$

The pros and cons of this simplification are the same as for the Kalman predictor. Figure 4.1.6a shows the Kalman filter with this kind of scaling and Figure 4.1.6b after having moved the scale factor out in front of the summation node. Table 4 summarizes the basic scaling procedures derived in this section.

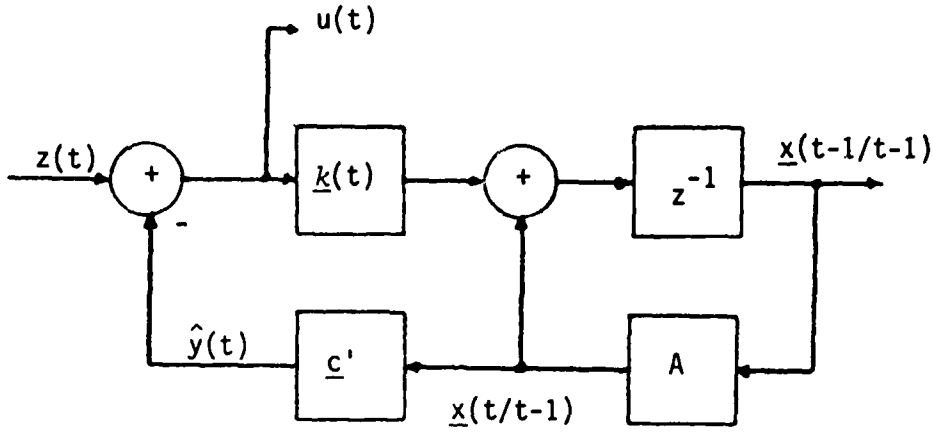


Fig. 4.1.4. The Kalman filter.

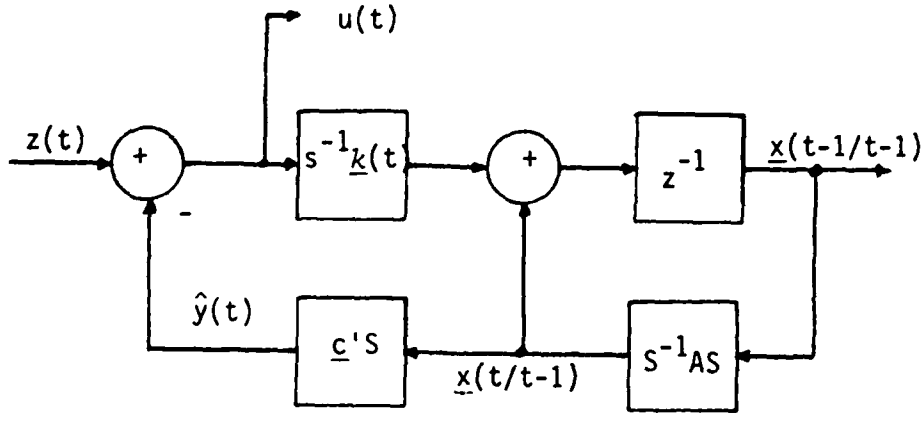


Fig. 4.1.5. The scaled Kalman filter.

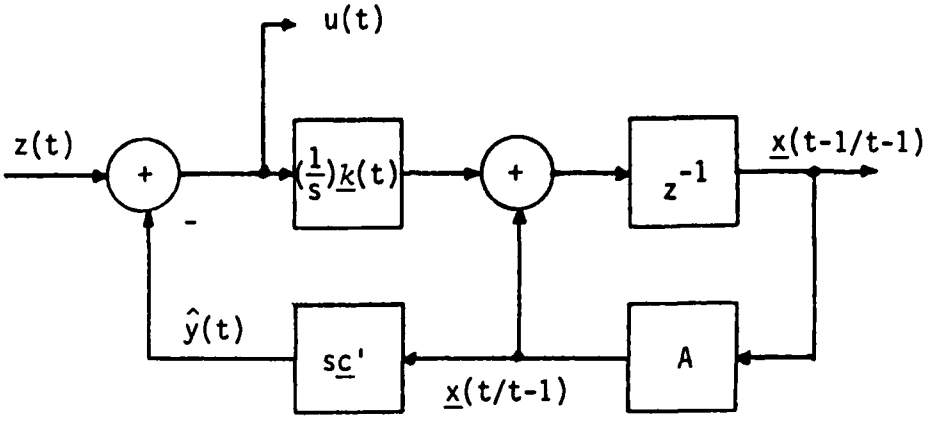


Fig. 4.1.6a. The scaled Kalman filter. Simplified scaling procedure.

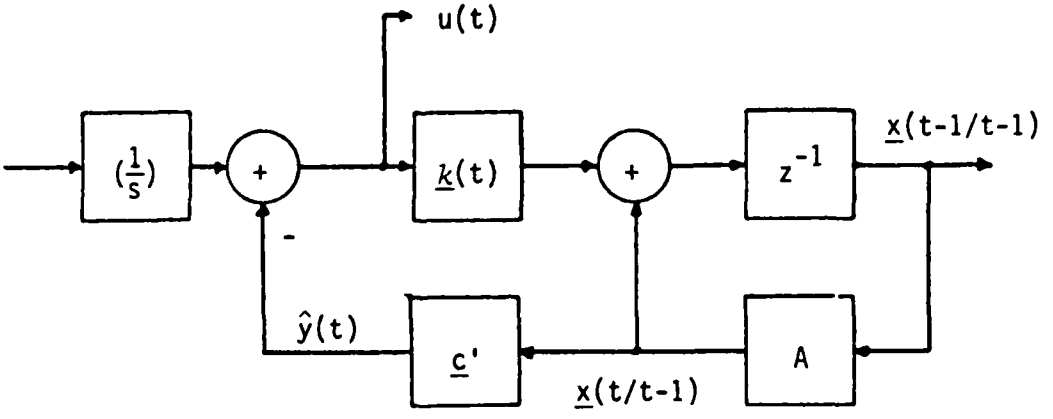


Fig. 4.1.6b. The scaled Kalman filter. Simplified scaling procedure.

Table 4. Unscaled and scaled Kalman predictor and Kalman filter.

The Kalman Predictor	The Kalman Filter
Unscaled State Space Model	
$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$	$\underline{x}(t/t) = A\underline{x}(t-1/t-1) + \underline{k}(t)u(t)$
$\hat{y}(t) = \underline{c}'\underline{x}(t/t-1)$; $\underline{x}(0/-1)=\underline{0}$	$\hat{y}(t) = \underline{c}'A\underline{x}(t-1/t-1)$; $\underline{x}(-1/-1)=\underline{0}$
$u(t) = z(t) - \hat{y}(t)$	$u(t) = z(t) - \hat{y}(t)$
Scaled State Vector	
$x(t/t-1) = S^{-1}x(t/t-1)$	$x(t/t) = S^{-1}x(t/t)$
S: a diagonal scaling matrix	S: a diagonal scaling matrix
Scaled State Space Model	
$\underline{x}(t+1/t) = (S^{-1}AS)\underline{x}(t/t-1) + (S^{-1}\underline{k}(t))u(t)$	$\underline{x}(t/t) = (S^{-1}AS)\underline{x}(t-1/t-1) + (S^{-1}\underline{k}(t))u(t)$
$\hat{y}(t) = (\underline{c}'S)\underline{x}(t/t-1)$; $\underline{x}(0/-1)=\underline{0}$	$\hat{y}(t) = (\underline{c}'AS)\underline{x}(t-1/t-1)$; $\underline{x}(-1/-1)=\underline{0}$
$u(t) = z(t) - \hat{y}(t)$	$u(t) = z(t) - \hat{y}(t)$
Scaling Constraints	
$s(k) = \delta\sqrt{q(k,k)}/(\varepsilon 2^{m-1})$	$s(k) = \delta\sqrt{q(k,k)}/(\varepsilon 2^{m-1})$
$q(k,k) = [Q(\infty)]_{kk}$	$q(k,k) = [Q(\infty)]_{kk}$
$Q(\infty) = AQ(\infty)A' + v(\infty)\underline{k}(\infty)\underline{k}'(\infty)$	$Q(\infty) = AQ(\infty)A' + v(\infty)\underline{k}(\infty)\underline{k}'(\infty)$
δ: param. that determines the probability of overflow	δ: param. that determines the probability of overflow
$(\varepsilon 2^{m-1})$: dynamic range bounds	$(\varepsilon 2^{m-1})$: dynamic range bounds
$s(k) = [S]_{kk}$	$s(k) = [S]_{kk}$

We have seen how to scale the state vectors in the Kalman predictor and filter. In the simplified scaling procedure we must also make sure that the reconstructed innovations sequence u(t) is bounded by the dynamic range of the numerical representation. The scale factor s must be chosen larger or equal to

$$s = \delta\sqrt{r(0) + \sigma_n^2}/(\varepsilon 2^{m-1}) \qquad (4.1.25)$$

to prevent overflow in u(t). There is a conflict between the scaling of the state variables and the scaling of the innovations sequence u(t). With increased measurements noise the state variance of the Kalman predictor/filter decreases in steady state. This means a smaller scale factor s. However the scale factor s must be larger with increased measurement noise as equation (4.1.25) shows.

The Fast Kalman Algorithm is very well suited for fixed point implementation since it is scaled by nature. This can be seen in the following way. We know that the gain vectors consists of the first n+1 values of the time varying impulse response $h^t(k)$. In the derivation of the Fast Kalman Algorithm we showed the cross correlation can be written as

$$s^t(k) = v(t)h^t(k) \qquad \forall\ k \qquad (4.1.26)$$

LeRoux and Gueguen [14] derived bounds on the cross correlations

$$|s^t(k)| \ <= \ r(0) \qquad \forall\ k \qquad (4.1.27)$$

Therefore we can write

$$|h^t(k)| \ = \left|\frac{s^t(k)}{v(t)}\right| \ <= \ \left|\frac{r(0)}{r(0)}\right| \ = 1 \qquad (4.1.28$$

This means all elements in the Kalman gain vector $\underline{k}(t)$ and $\underline{k}(t)$ are bounded by +/-1 and there is no need to scale the algorithm. When there is measurement noise the bound on $h^t(k)$ becomes

$$|h^t(k)| \ <= r(0)/(r(0) + \sigma_n^2) \ < 1 \quad \forall\ k \qquad (4.1.29)$$

4.2 Roundoff Noise

The effects of multiplication roundoff in the Kalman predictor and Kalman filter are considered in this section. We will be mainly concerned with how the roundoff affects the state vectors since these are considered to be the output of the Kalman predictor and filter. It is also of concern to know how the roundoff affects y(t) since this is fed back to the input and any added noise on y(t) will increase the variance of u(t). We assume that the effect of multiplication roundoff can be modeled by an additive white noise source with zero mean and variance $\varepsilon^2/12$ and that all such noise sources are independent from source to source. Here ε is the quantization step size as before. See for example references [9] and [10].

Roundoff Noise in the Kalman Predictor

The state vector of the Kalman predictor is updated as

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$$

So for each element in the state vector $\underline{x}(t+1/t)$ there are $\nu(i)$ multiplications required where

$$\nu(i) = \begin{cases} 1 & i=1,2, \ldots, n-1 \\ n+1 & i=n \end{cases} \qquad (4.2.1)$$

This follows from the structure of the state matrix A. For every multiplication we add one noise source so the total effect is an additive white noise source of zero mean and variance $\nu(i)\varepsilon^2/12$. The state update equation with additive noise included is then

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t) + \underline{\xi}(t) \qquad (4.2.2)$$

where $\underline{\xi}(t)$ is an uncorrelated vector sequence with zero mean and variance E:

$$E = \varepsilon^2/12 \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & n+1 \end{bmatrix} \tag{4.2.3}$$

This is the same formulation as given by Mullis and Roberts [1]. The state variance with the effect of roundoff included is then

$$Q(t) = AQ(t-1)A' + v(t-1)\underline{k}(t-1)\underline{k}'(t-1) + E \tag{4.2.4}$$

and in steady state

$$Q(\infty) = AQ(\infty)A' + v(\infty)\underline{k}(\infty)\underline{k}'(\infty) + E \tag{4.2.5}$$

The random vector sequence $\underline{\xi}(t)$ will increase the state variance from what it was with no roundoff.

Roundoff Noise in the Kalman Filter

For the Kalman filter the state vector is updated as

$$\underline{x}(t/t) = A\underline{x}(t-1/t-1) + \underline{k}(t)u(t)$$

which becomes

$$\underline{x}(t/t) = A\underline{x}(t-1/t-1) + \underline{k}(t)u(t) + \underline{\xi}(t) \tag{4.2.6}$$

when the effect of roundoff is included. The vector sequence $\underline{\xi}(t)$ is defined as before. The state variance of the Kalman filter is then

$$Q(t) = AQ(t-1)A' + v(t)\underline{k}(t)\underline{k}'(t) + E \tag{4.2.7}$$

and in steady state

$$Q(\infty) = AQ(\infty)A' + v(\infty)\underline{k}(\infty)\underline{k}'(\infty) + E \tag{4.2.8}$$

The variance of $\underline{x}(t/t-1) = A\underline{x}(t-1/t-1)$ assuming roundoff is

$$\begin{aligned} Q(t) &= AQ(t-1)A' \\ &= AQ(t-1)A' + v(t)\underline{k}(t)\underline{k}'(t) + AEA' \end{aligned} \tag{4.2.9}$$

For the Kalman predictor assuming roundoff we had

$$Q(t) = AQ(t-1)A' + v(t)\underline{k}(t)\underline{k}'(t) + E$$

This means that using the Kalman filter as a Kalman predictor gives worse prediction results in terms of state variance when the effect of roundoff is included. This increased state variance has an effect on the estimated output $\hat{y}(t)$

$$\hat{y}(t) = \underline{c}'\underline{x}(t/t-1) = \underline{c}'[A\underline{x}(t-\bar{1}/t-1)]$$

in terms of increased variance. In steady state the variance of $\hat{y}(t)$ of the Kalman predictor is

$$E[\hat{y}(t)\hat{y}'(t)] = \underline{c}'Q(\infty)\underline{c}$$

where $Q(\infty)$ is determined from equation (4.2.5). For the Kalman filter the variance of $\hat{y}(t)$ in steady state is

$$E[\hat{y}(t)\hat{y}'(t)] = \underline{c}'\mathit{Q}(\infty)\underline{c}$$

where $\mathit{Q}(\infty)$ is determined from equation (4.2.9). This increased variance of $\hat{y}(t)$ could affect the variance of the innovations sequence $u(t)$ since $u(t) = z(t)-\hat{y}(t)$. To what extent depends on the variance of the input sequence $z(t)$.

All derivations have been carried out using the unscaled Kalman predictor and filter. The corresponding results for the scaled predictor and filter follow by replacing A by $(S^{-1}AS)$, $\underline{k}(t)$ by $(S^{-1}\underline{k}(t))$, $\underline{\mathit{k}}(t)$ by $(S^{-1}\underline{\mathit{k}}(t))$ and $\underline{c}'$ by $(\underline{c}'S)$.

We have not considered the effect of coefficient and input quantization. This effect is basically deterministic. We have also not determined how roundoff errors propagate in the calculation of the Kalman gain vectors using fixed point arithmetic. It is unlikely that the roundoff error can be modeled as an additive white noise source since the gain values do not change that rapidly between samples. In Chapter 5 we give numerical results from calculating the Kalman gain vectors using fixed point arithmetic.

4.3 Computational Complexity

This section deals with the computational complexity of the Kalman predictor and filter in terms of the number of multiplications, divisions, additions, and subtractions. Memory requirements are also considered.

The Kalman predictor is described by the following equations:

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$$

$$u(t) = z(t) - \hat{y}(t)$$

$$\hat{y}(t) = \underline{c}'\underline{x}(t/t-1)$$

The Kalman filter is described by

$$\underline{x}(t/t) = A\underline{x}(t-1/t-1) + \underline{k}(t)u(t)$$

$$u(t) = z(t) - \hat{y}(t)$$

$$\hat{y}(t) = \underline{c}'A\underline{x}(t-1/t-1)$$

The computational burden can be divided into the following:

1) Update the Kalman gain vector ($\underline{k}(t)$ or $\underline{k}(t)$)

2) Update the state vector ($\underline{x}(t+1/t)$ or $\underline{x}(t/t)$)

Even though it is possible to calculate the Kalman gain vector ahead of time we will assume it is done in real time along with the updating of the state vector.

Complexity of the Kalman Gain Calculation

For the Kalman filter we have

$$\underline{k}(t+1) = [\underline{k}(t) - (\underline{c}'\underline{l}(t))\underline{l}(t)]/d(t)$$

$$\underline{l}(t+1) = A[\underline{l}(t) - (\underline{c}'\underline{l}(t))\underline{k}(t)]/d(t)$$

$$d(t) = 1 - (\underline{c}'\underline{l}(t))^2$$

For the Kalman predictor we have

$$\underline{k}(t+1) = [\underline{k}(t) - (\underline{c}'\underline{l}(t))A\underline{l}(t)]/d(t)$$

$$\underline{l}(t+1) = [A\underline{l}(t) - (\underline{c}'\underline{l}(t))\underline{k}(t)]/d(t)$$

$$d(t) = 1 - (\underline{c}'\underline{l}(t))^2$$

By inspection we determine the number of multiplications, divisions, additions, and subtractions required to update the Kalman gain vectors. Table 5 gives the results. The interesting thing is that it requires a total of (6n+2) multiplications and divisions to update the predictor gain vector but only (5n+2) to update the filter gain vector. Using the relationship

$$\underline{k}(t) = A\underline{k}(t)$$

to calculate the predictor gain vector requires the same number of arithmetic operations as using the defining equations.

Complexity in Updating the State Vectors

Again by inspection we determine the number of arithmetic operations required to update the state vectors. In Table 5 we list the results along with the total number of arithmetic operations needed to go through one cycle of the Kalman predictor and Kalman filter. The conclusion is that it takes a greater number of arithmetic operations to implement the Kalman predictor than the Kalman filter. The Kalman filter can act both as a filter and a predictor. So in all respect the Kalman filter is more efficient than the Kalman predictor, except that the rounding variance is higher is the filter than in the predictor. If the Kalman gain vector is calculated ahead of time then the Kalman predictor and filter are equal in terms of arithmetic operations. The number of multiplications and divisions required then is on the order of (2n).

Table 5. Computational requirements for the Kalman predictor and Kalman filter.

Updating	Number of Arithmetic Operations # Multiplications	# Divisions	# Additions	# Subtractions
	The Kalman Predictor			
The Gain Vector, $\underline{k}(t)$	4n + 2	2n	0	3
The state Vector, $\underline{x}(t+1/t)$	2n	0	1	1
The Kalman Predictor	6n + 2	2n	1	4
	The Kalman Filter			
The Gain Vector, $\underline{k}(t)$	3n + 2	2n	0	3
The state Vector, $\underline{x}(t/t)$	2n	0	1	1
The Kalman Filter	5n + 2	2n	1	4

Example

Assume that it takes 15μs to multiply and divide and 4μs to add and subtract. Then the total cycle times for the Kalman predictor and filter for the cases (1) n=5 and (2) n=10 are on the order of

1) n=5

Filter: (7*5+2)*15 + (1+4)*5 = 580μs

Predictor: (8*5+2)*15 + (1+4)*5 = 655μs

2) n=10

Filter: (7*10+2)*15 + (1+4)*5 = 1105μs

Predictor: (7*10+2)*15 + (1+4)*5 = 1255μs

Memory Requirements

For the Kalman predictor the storage requirements are on the order of (3n+1) assuming only the state and gain vector must be stored along with the denominator coefficients a(i) i=0,n. When it is required to store z(t) and u(t) the total memory requirement is (3n+3) per time sample. The memory requirements for the Kalman filter are on the order of (4n+3) assuming that the predictor state vector must be stored completely.

REFERENCES FOR CHAPTER 4

1. C.T. Mullis and R.A. Roberts, "Synthesis of Minimum Roundoff Fixed Point Digital Filters," IEEE Trans. on Circuits and Systems, Vol. CAS-23, No. 9, Sept. 1976.

2. S.Y. Hwang, "Dynamic Range Constraint in State-Space Digital Filtering," IEEE Trans. on ASSP, Vol. ASSP-23, December 1975.

3. L.B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," Bell Syst. Tech. J., Vol. 49, 1970.

4. L.B. Jackson, "Roundoff Noise Analysis for Fixed-Point Digital Filters Realized in Cascade of Parallel Form," IEEE Trans. on Audio and Electroacoust., Vol. AU-18, No. 2, June 1970.

5. S.Y. Hwang, "Roundoff Noise in State-Space Digital Filtering: A General Analysis," IEEE Trans. on ASSP, Vol. ASSP-24, December 1975.

6. C.T. Mullis and R.A. Roberts, "Roundoff Noise in Digital Filter Frequency Transformations and Invariants," IEEE Trans. on ASSP, Vol. ASSP-24, No. 6, December 1976.

7. S.Y. Hwang, "Minimum Uncorrelated Unit Noise in State-Space Digital Filtering," IEEE Trans. on ASSP, Vol. ASSP-25, No. 4, August 1977.

8. P.R. Be'langer and M.E. Ahmed, "Scaling and Roundoff in Fixed-Point Controllers," Proc. C.D.C. Conf., Albuquerque, December 1980.

9. P. Moroney, A.S. Willsky and P.K. Houpt, "The Digital Implementation of Control Compensators: The Coefficient Wordlength Issue," IEEE Trans. on Automatic Control, Vol. AC-25, August 1980.

10. A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Englewood Cliffs, N.J.: Prentice-Hall 1975.

11. A. Peled and B. Liu, Digital Signal Processing, Theory, Design, and Implementation. John Wiley & Sons 1976.

12. A.B. Skipad and D.L. Snyder, "A Necessary and Sufficient Condition for Quantization Errors to be Uniform and White," IEEE Trans. on ASSP, Vol. ASSP-25, No. 5, October 1977.

13. J.G. Pearlman, "An Algorithm for exact likelihood of high-order autoregressive-moving average process," Biometrica, 67, 1, 1980.

14. J. LeRoux and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients," Trans. Acoust.., Speech, and Signal Processing, Vol. 25, No. 2, pp. 257-259, June 1977.

CHAPTER 5

NUMERICAL RESULTS

In this chapter we give numerical results obtained by implementing the Kalman filter using floating point and fixed point arithmetic. The floating point calculations were carried out on an Intel Development System Model 230 using Fortran as the program language. The fixed point implementation was carried out using an Intel 8086 microprocessor which was a part of the SDK-86 single board microcomputer. The Intel 8086 has a wordlength of 16 bits. The actual program was written in assembly language. All the results from the *floating point and the fixed point implementations* can be stored on a file for later analysis.

The input data to the Kalman filter was generated by running the Innovations representation #1 with measurement noise added to $y(t)$ to obtain $z(t)=y(t)+n(t)$. The data $z(t)$ was generated on the Intel Development System using floating point arithmetic. For information on the software and the system configuration see Appendices A and B.

Section 5.1 covers results from the Kalman filter implemented using floating point arithmetic. In section 5.2 we look at the corresponding fixed point results.

5.1 Results Using Floating Point Arithmetic

In this section we present results for a Kalman filter implemented using floating point arithmetic. The example used is an

ARMA(3,2) process:

$$H(z) = \frac{1 - 1.75\ z^{-1} + 0.8\ z^{-2}}{1 - 1.5\ z^{-1} + 1.21\ z^{-2} - 0.4550\ z^{-3}}$$

To start the Kalman filter we need the denominator coefficients, the first 4 values of the covariance sequence of the ARMA process and the measurement noise variance. This is also needed to generate the input data z(t). The covariance sequence of the ARMA process is calculated using the method of Mullis and Roberts (ref [5] in Chapter 1). Figure 5.1.1 shows a plot of the first 50 values of the covariance sequence. The first 4 values are

$$\underline{r}' = (r(0),r(1),r(2),r(3)$$
$$= (2.27497,\ 0.43467,\ -1.10293,\ -1.14524)$$

The things we are interested in are the Kalman filter gain vector, $\underline{\hat{k}}(t)$, the innovations sequence $\hat{u}(t)$ and its calculated variance $\hat{v}(t)$, and the estimated output $\hat{y}(t)$. We also calculate the Kalman predictor gain vector $\underline{\hat{k}}(t)$ but it is not used for the Kalman filter. What follows are results from the Kalman filter run for three different cases

1) $\sigma_n^2 = 0.0,\ r(0)/\sigma_n^2 = \infty$
2) $\sigma_n^2 = 1.0,\ r(0)/\sigma_n^2 = 2.27497$
3) $\sigma_n^2 = 2.0,\ r(0)/\sigma_n^2 = 1.13749$

Case #1, $\sigma_n^2 = 0$

Figure 5.1.2 shows a plot of the Kalman filter gain vector $\underline{\hat{k}}(t)$ along with the Kalman gain vector of the Innovations representation #2 $\underline{k}(t)$. The numbers from 1 to 3 correspond to the elements in $\underline{k}(t)$ and the numbers 4 to 6 correspond to the elements in $\underline{\hat{k}}(t)$. As expected they are equal, i.e.,

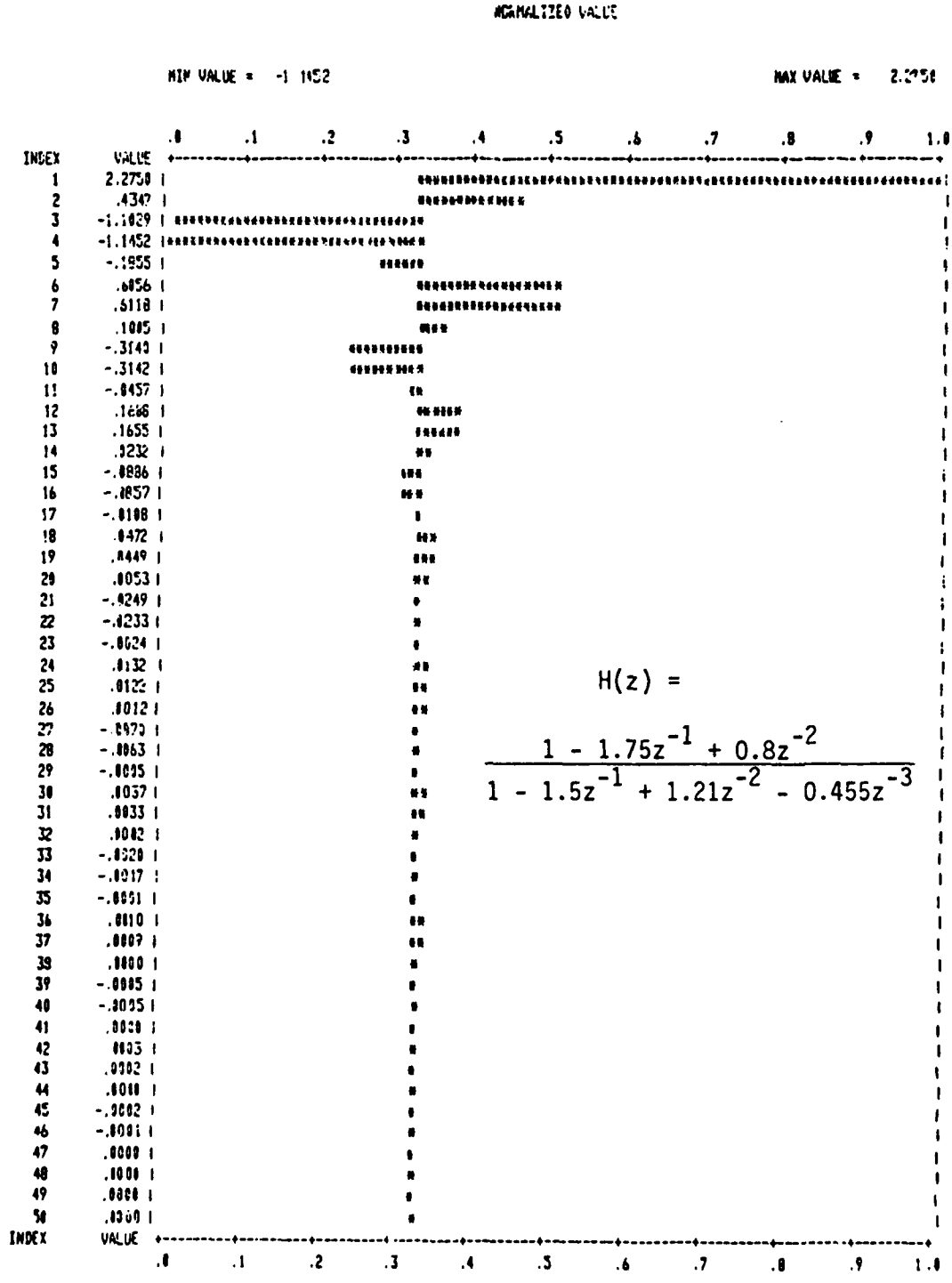


Fig. 5.1.1. The covariance seq. r(k) k = 0,49 for an ARMA(3.2) process.

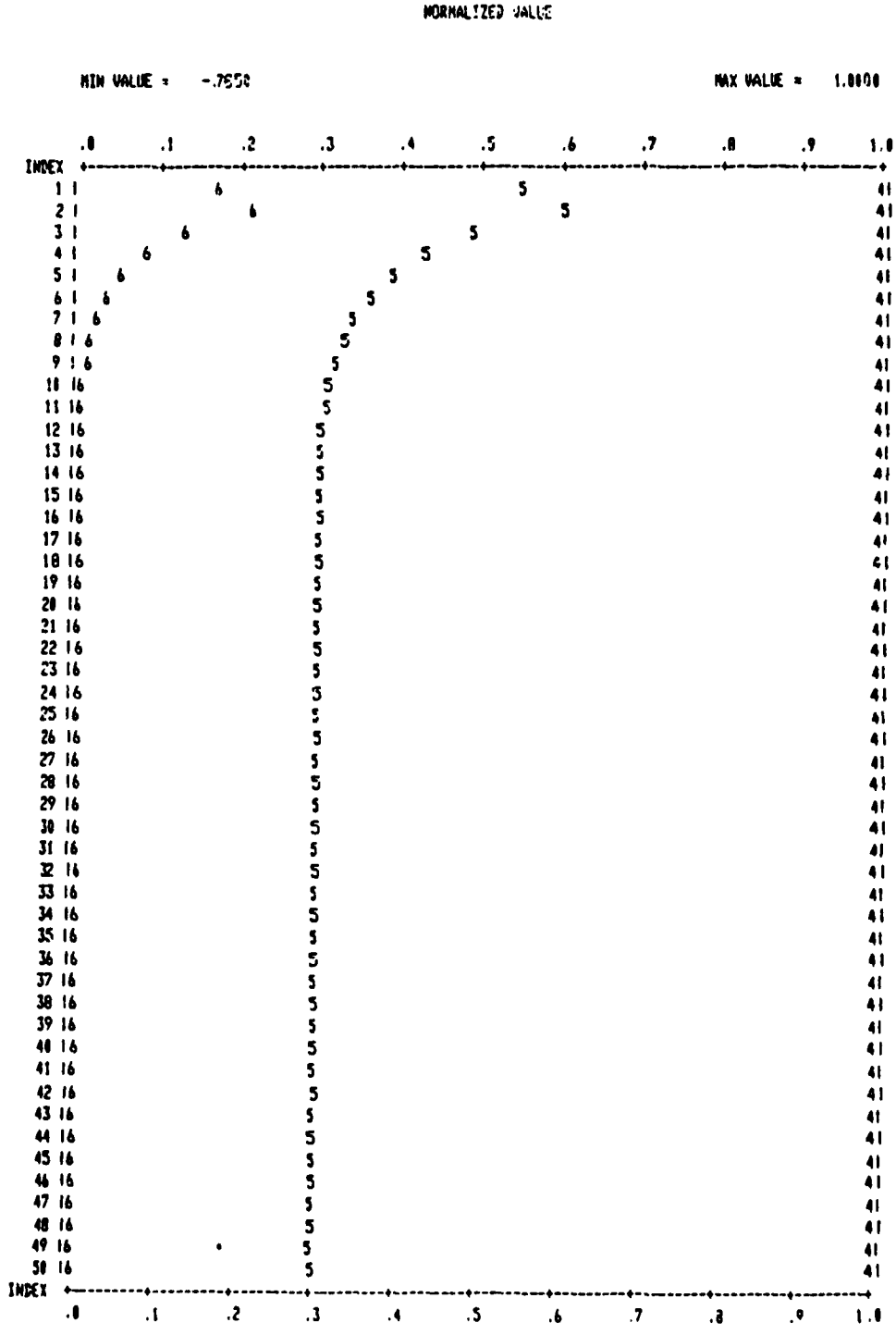


Fig. 5.1.2. The Kalman filter gain vectors $\underline{k}(t)$ and $\hat{\underline{k}}(t)$. Noise variance $\sigma_n^2 = 0$.

$$\hat{\underline{k}}(t) = \underline{k}(t)$$

In the figure 4 overwrites 1, 5 overwrites 2, . . .

The initial value of the Kalman filter gain vector is

$$\hat{\underline{k}}'(0) = [r(0)\ r(1)\ r(2)]/r(0)$$

$$= (1.00000 \quad 0.19107 \quad -0.48481)$$

At time t=49 the value of the gain vector is

$$\hat{\underline{k}}'(49) = (1.00000 \quad -0.24994 \quad -0.78495)$$

The final value should be

$$\hat{\underline{k}}'(\infty) = (h(0) \quad h(1) \quad h(2))$$

$$= (1.00000 \quad -0.2500 \quad -0.78500)$$

The h(k) is determined from equation (2.2.5).

Figure 5.1.3 shows a plot of the Kalman predictor gain vector $\hat{\underline{k}}(t)$ along with the Kalman gain vector of the Innovations Representation #1 $\underline{k}(t)$. As can be seen they are equal:

$$\hat{\underline{k}}(t) = \underline{k}(t)$$

Again the numbers 1, 2 and 3 are overwritten.

The initial value of the Kalman predictor gain vector is

$$\hat{\underline{k}}'(0) = [r(1)\ r(2)\ r(3)]/r(0)$$

$$= (0.19107 \quad 0.48481 \quad -0.50341)$$

At time t=49 the Predictor gain vector is

$$\hat{\underline{k}}'(49) = (-0.24994 \quad -0.78495 \quad -0.42001)$$

The steady state value of the Predictor gain vector should be

$$\hat{\underline{k}}'(\infty) = (-0.25000 \quad -0.78500 \quad -0.42000)$$

The anticausal impulse response vector of the signal model and the Kalman filter is plotted in Figure 5.1.4. The numbers 1 to 3 correspond to signal model and the numbers 4 to 6 to the Kalman filter. The two responses are equal. The initial value is

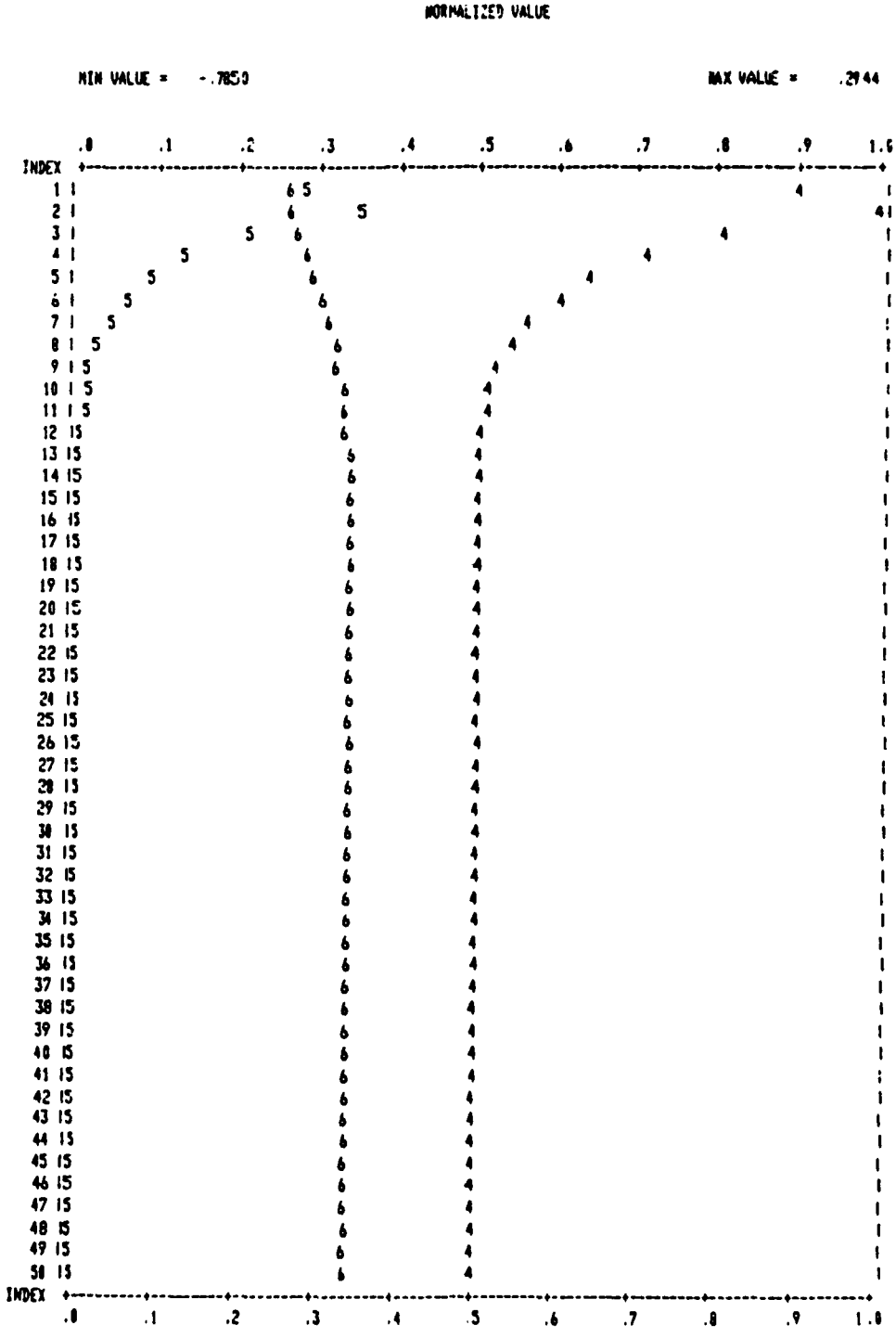


Fig. 5.1.3. The Kalman predictor gain vectors $\underline{k}(t)$ and $\hat{\underline{k}}(t)$. Noise variance $\sigma_n^2 = 0$.

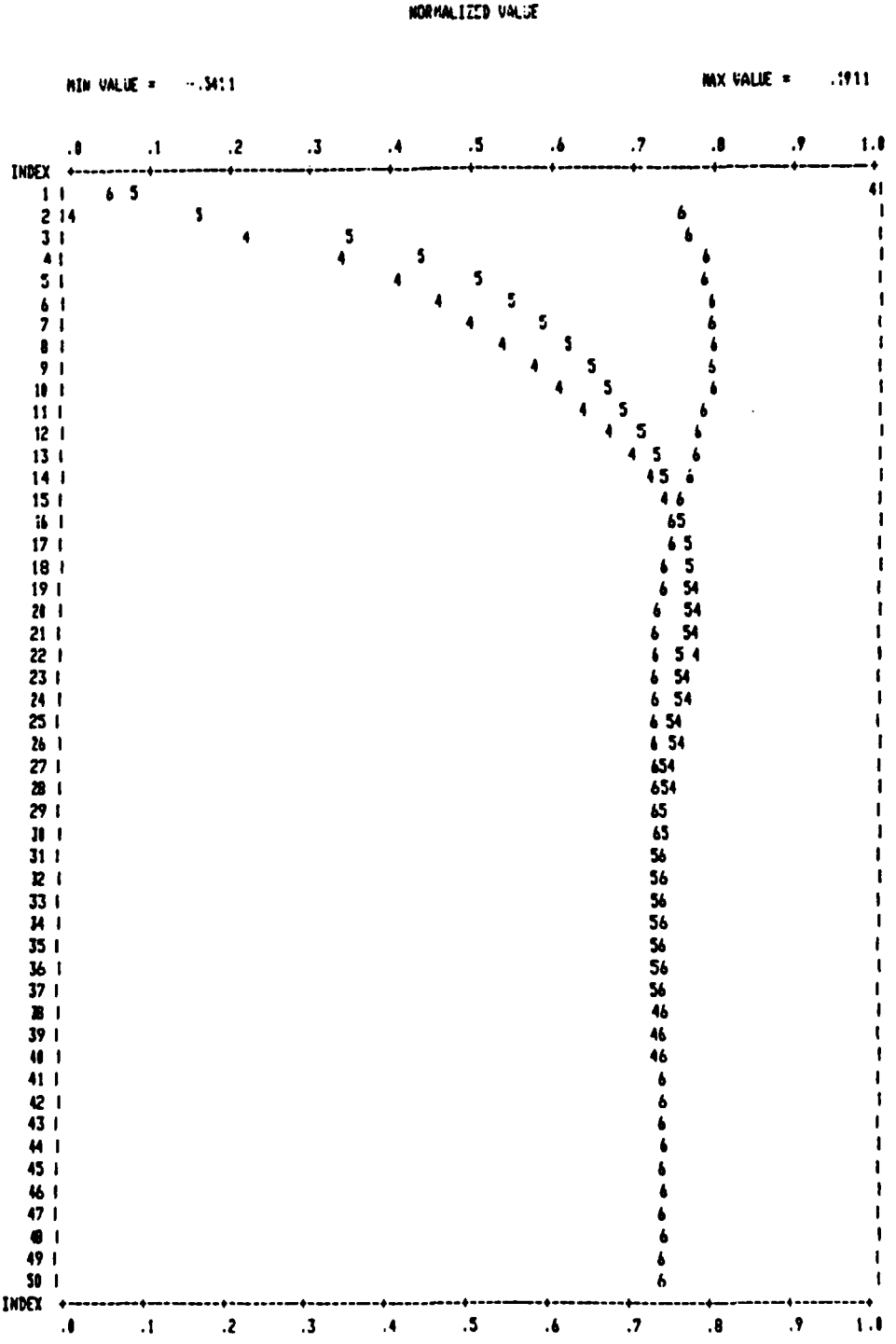


Fig. 5.1.4. The anticausal impulse response vectors $\underline{1}(t)$ and $\hat{\underline{1}}(t)$. Noise variance $\sigma_n^2 = 0$.

$$\underline{\hat{l}}(0) = \underline{\hat{k}}(0)$$

The steady state value should be zero, i.e.,

$$\underline{\hat{l}}'(\infty) = (0.0 \quad 0.0 \quad 0.0)$$

At time t=49 the value is

$$\underline{\hat{l}}'(49) = (0.00104 \quad 0.00076 \quad -0.00011)$$

Figure 5.1.5 shows a plot of the output of the signal model $y(t)$ and the estimated output, $\hat{y}(t)$. The difference between the two is the reconstructed innovations sequence since $\hat{u}(t)=y(t)-\hat{y}(t)$.

Figure 5.1.6 shows a plot of the two innovations sequences of the signal model and the Kalman filter, $u(t)$ and $\hat{u}(t)$. As can be seen the Kalman filter reconstructs the innovations sequence of the signal model exactly. The ones correspond to $u(t)$ and the twos to $\hat{u}(t)$. The 2s overwrite the 1s. Figure 5.1.7 shows a plot of the calculated variance of $u(t)$ and $\hat{u}(t)$. The initial value of $\hat{v}(t)$ is

$$\hat{v}(0) = \bar{r}(0) = 2.27497$$

The final value should be

$$\hat{v}(\infty) = \sigma_u^2 = 1.00000$$

At time t=49 the calculated variance is

$$\hat{v}(49) = 1.00006$$

Case #2, $\sigma_n^2 = 1.0$

Now we have a measurement noise. Figures 5.1.8 through 5.1.13 show the same kind of information as Figures 5.1.2 through 5.1.7. The Kalman gain vectors are now different from the noise free case. The Kalman filter does not reconstruct the innovations sequence of the signal model as before. The state variance of the filter is now less than in the noise free case.

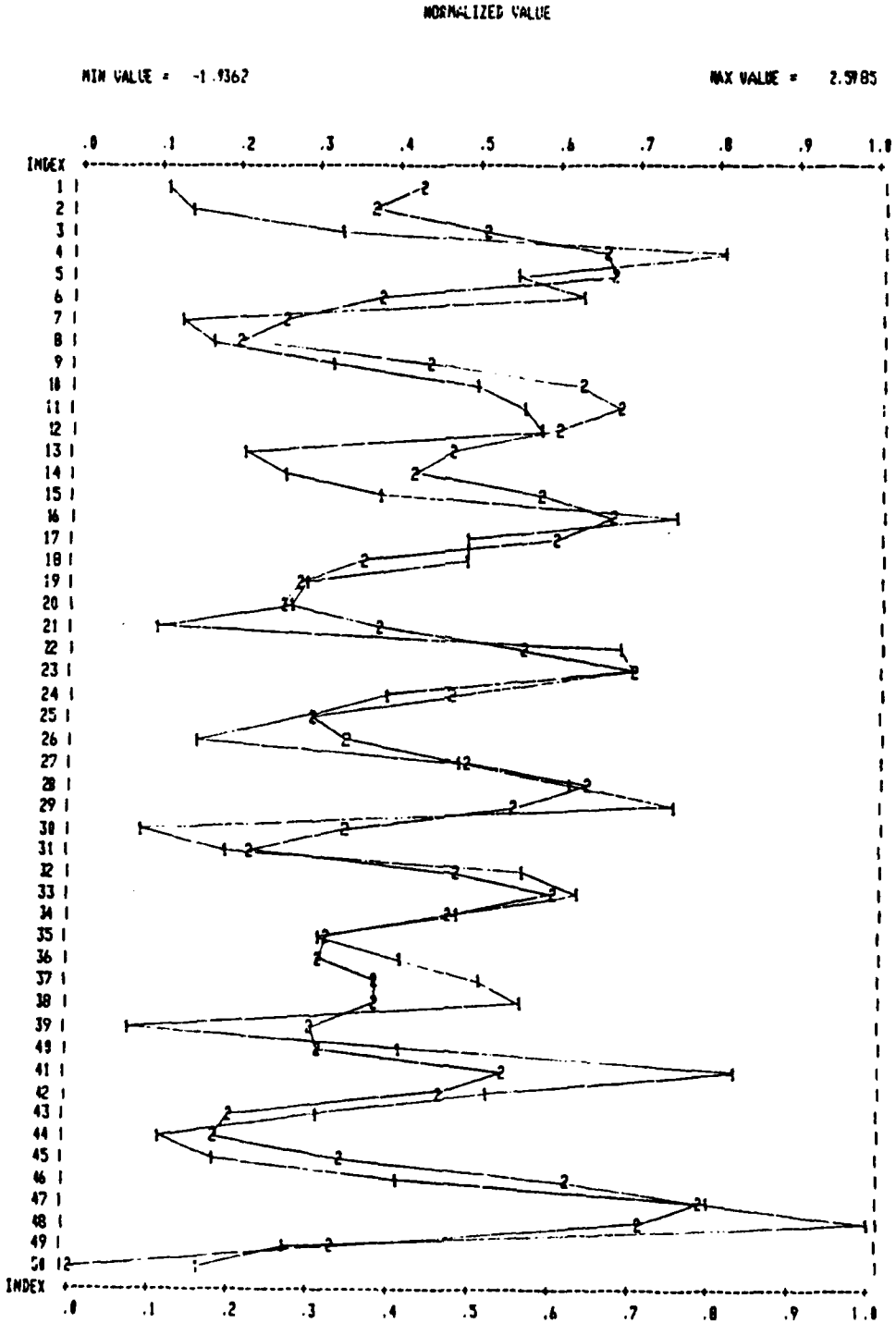


Fig. 5.1.5. The output y(t) and the estimated output $\hat{y}(t)$. Noise variance $\sigma_n^2 = 0$.

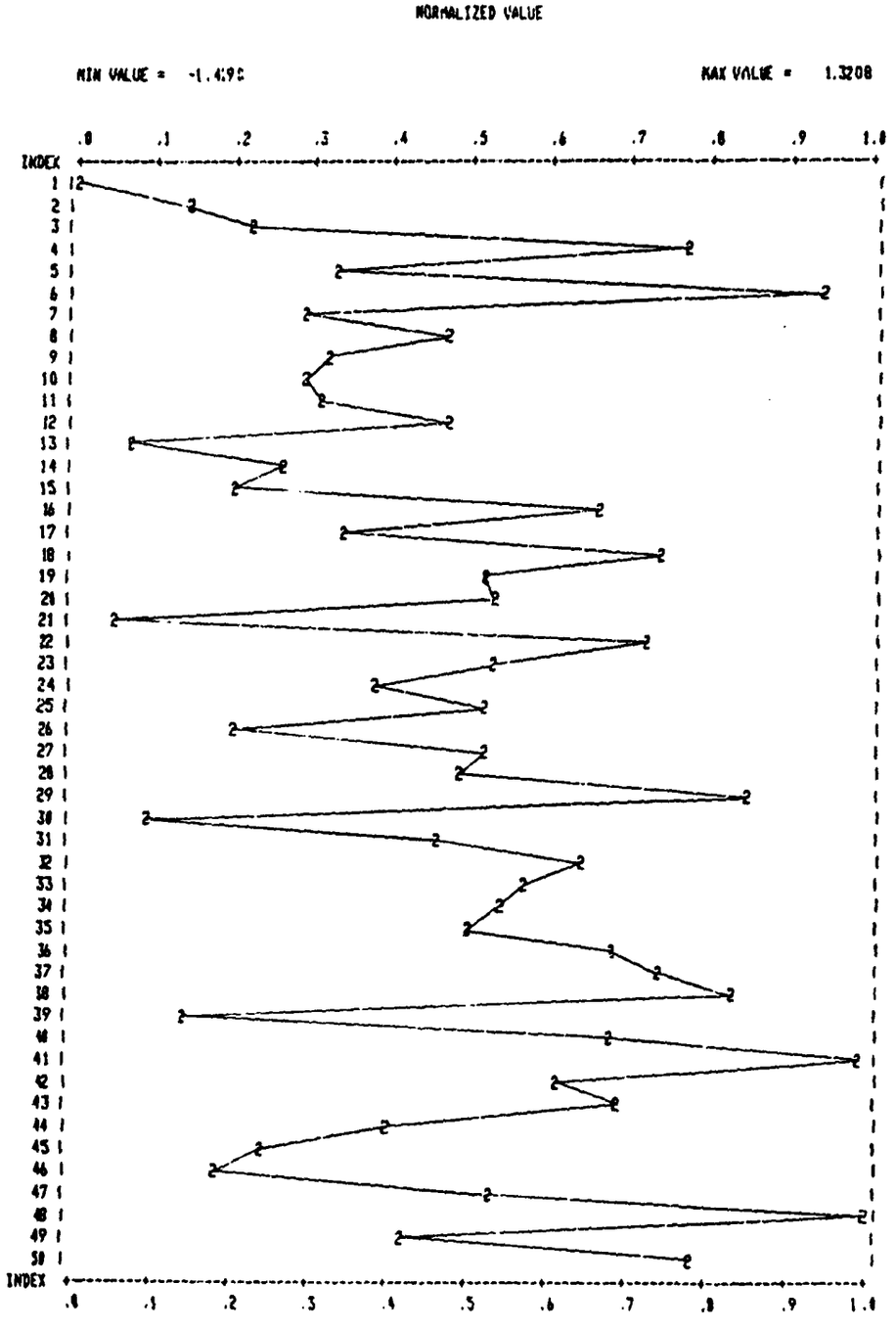


Fig. 5.1.6. The innovations sequences $u(t)$ and $\hat{u}(t)$. Noise variance $\sigma_n^2 = 0$.

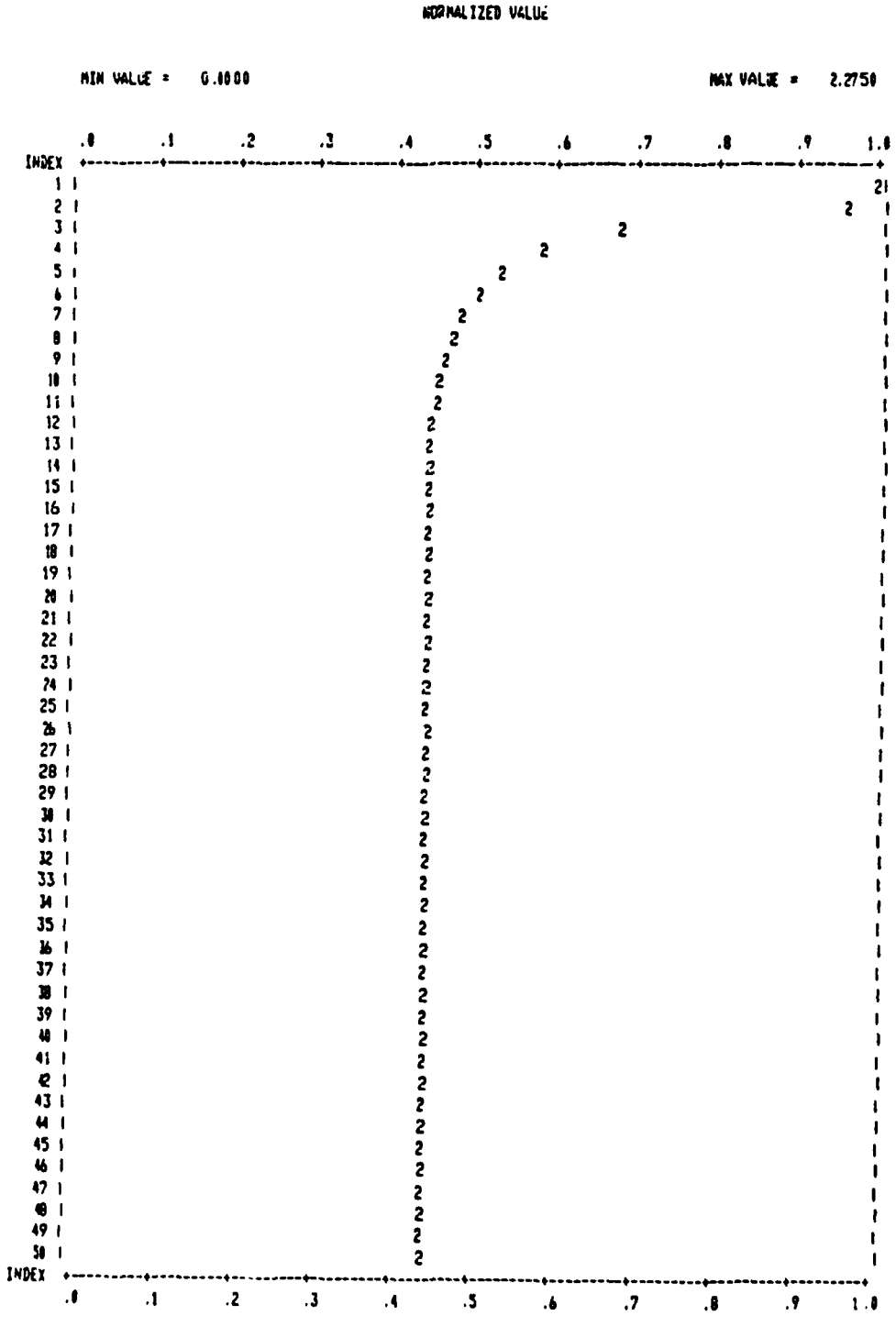


Fig. 5.1.7. The innovations variance v(t) and $\hat{v}(t)$. Noise variance $\sigma_n^2 = 0$.

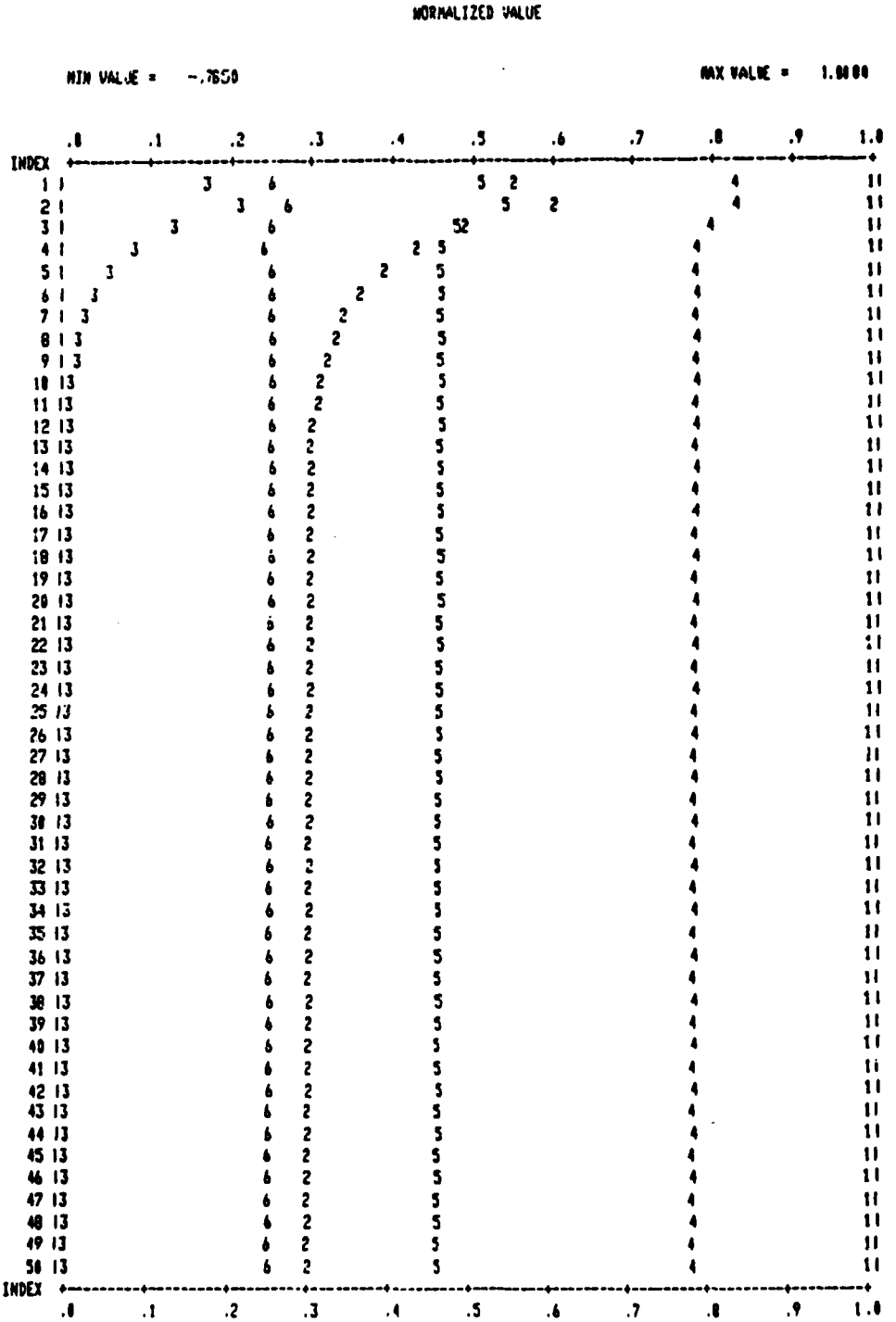


Fig. 5.1.8. The Kalman filter gain vectors $\underline{k}(t)$ and $\underline{\hat{k}}(t)$. Noise variance $\sigma_n^2 = 1.0$.

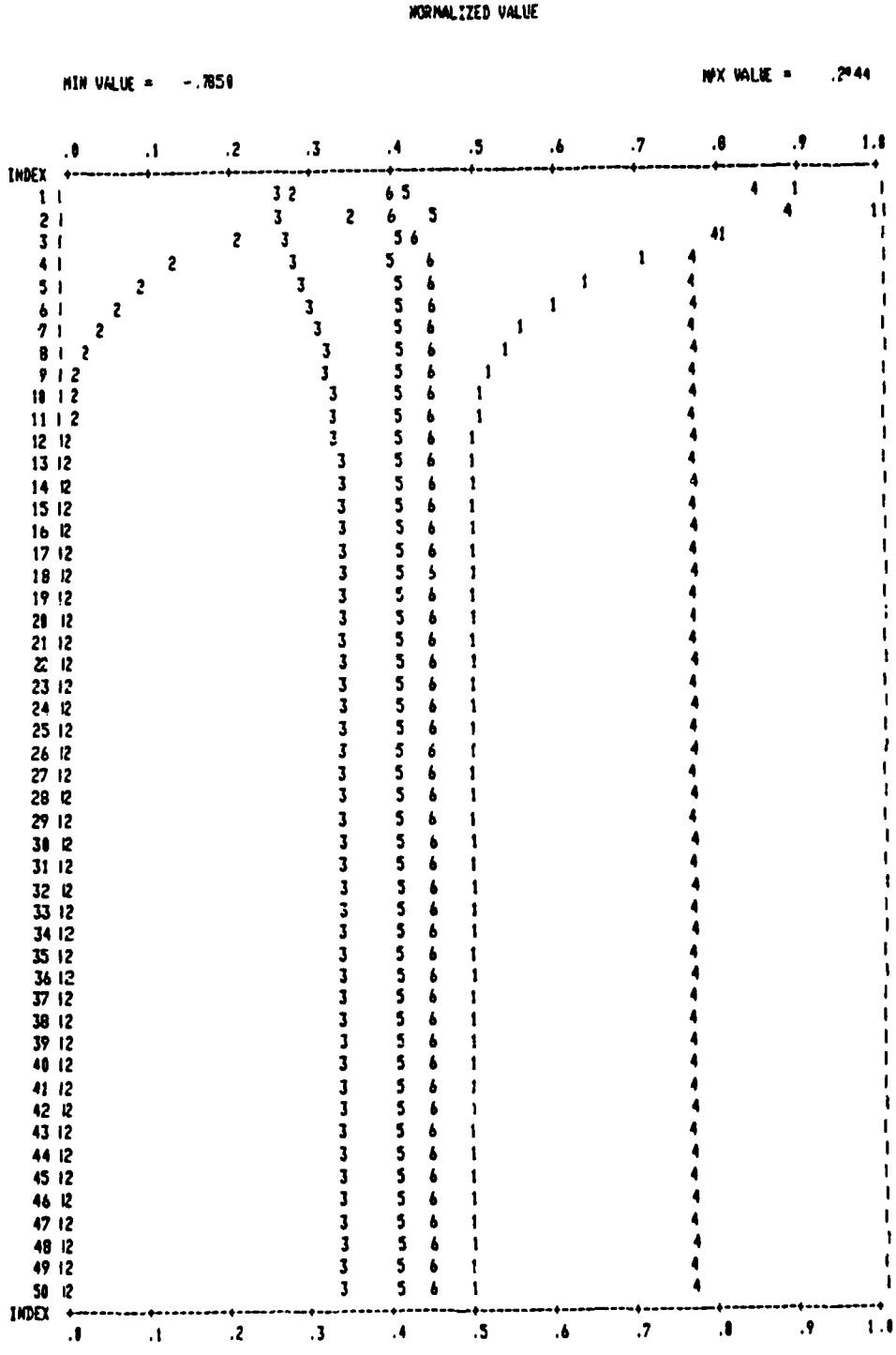


Fig. 5.1.9. The Kalman predictor gain vectors $\underline{k}(t)$ and $\hat{\underline{k}}(t)$. Noise variance $\sigma_n^2 = 1.0$.

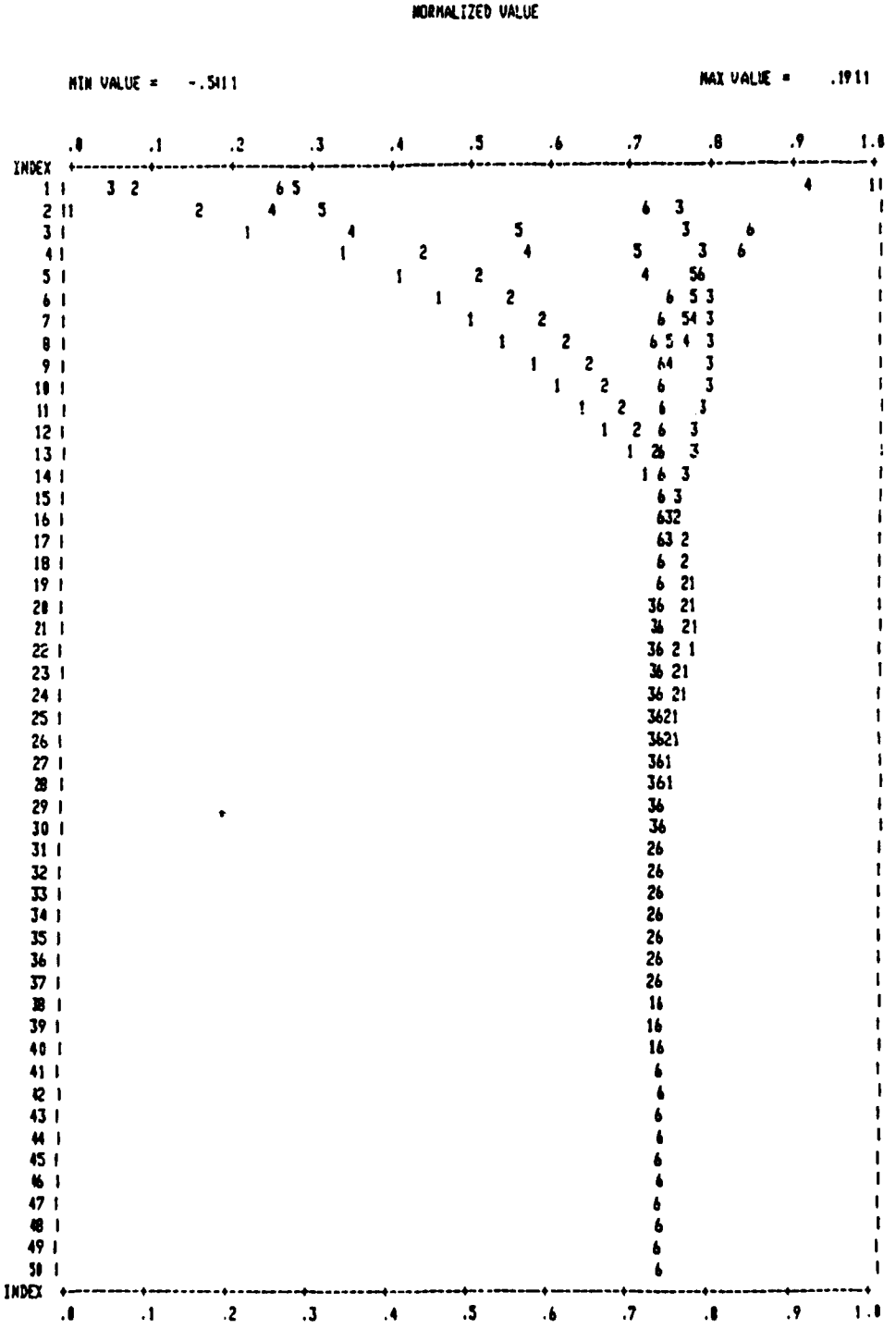


Fig. 5.1.10 The anticausal impulse response vectors $\underline{1}(t)$ and $\hat{\underline{1}}(t)$. Noise variance $\sigma_n^2 = 1.0$.

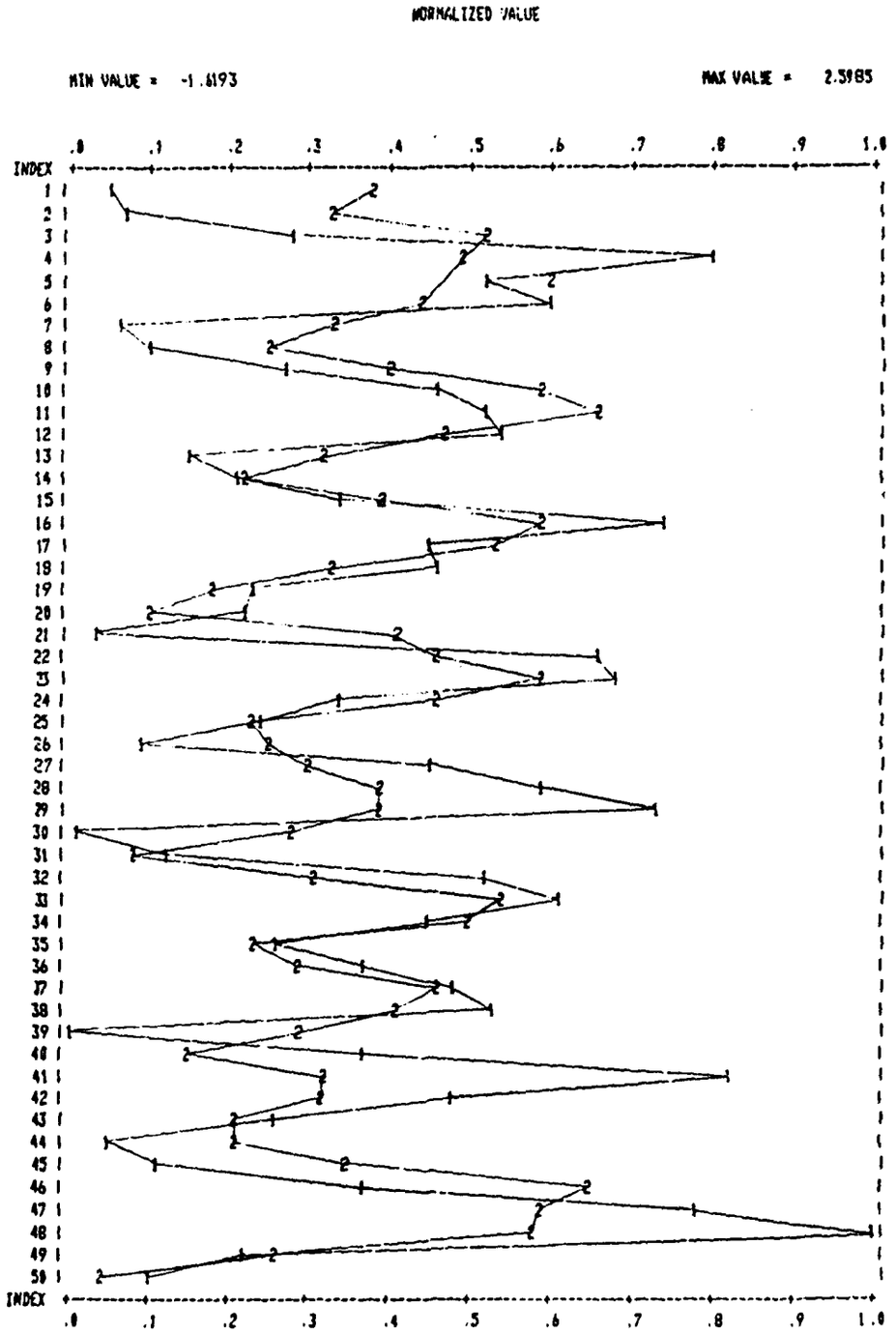
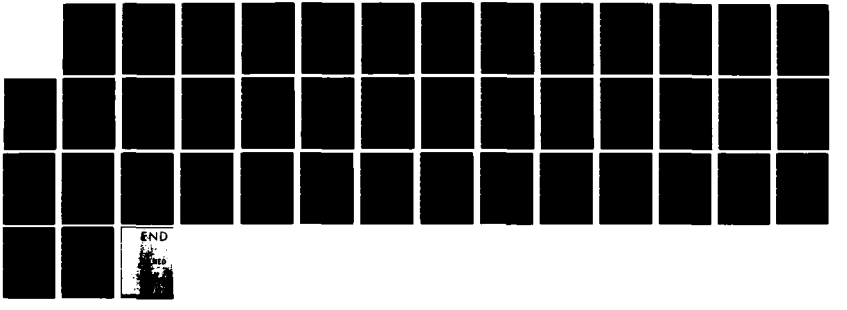
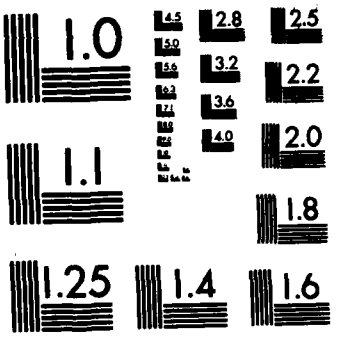


Fig. 5.1.11. The output y(t) and the estimated output $\hat{y}(t)$. Noise variance $\sigma_n^2 = 1.0$.

AD-A126 994 FAST KALMAN FILTERING FOR ARMA (AUTOREGRESSIVE MOVING AVERAGE) PROCESSES:..(U) RHODE ISLAND UNIV KINGSTON DEPT OF ELECTRICAL ENGINEERING S SIGURDSSON 15 MAR 83 2/2

UNCLASSIFIED N00014-82-K-0300 . F/G 12/1 NL





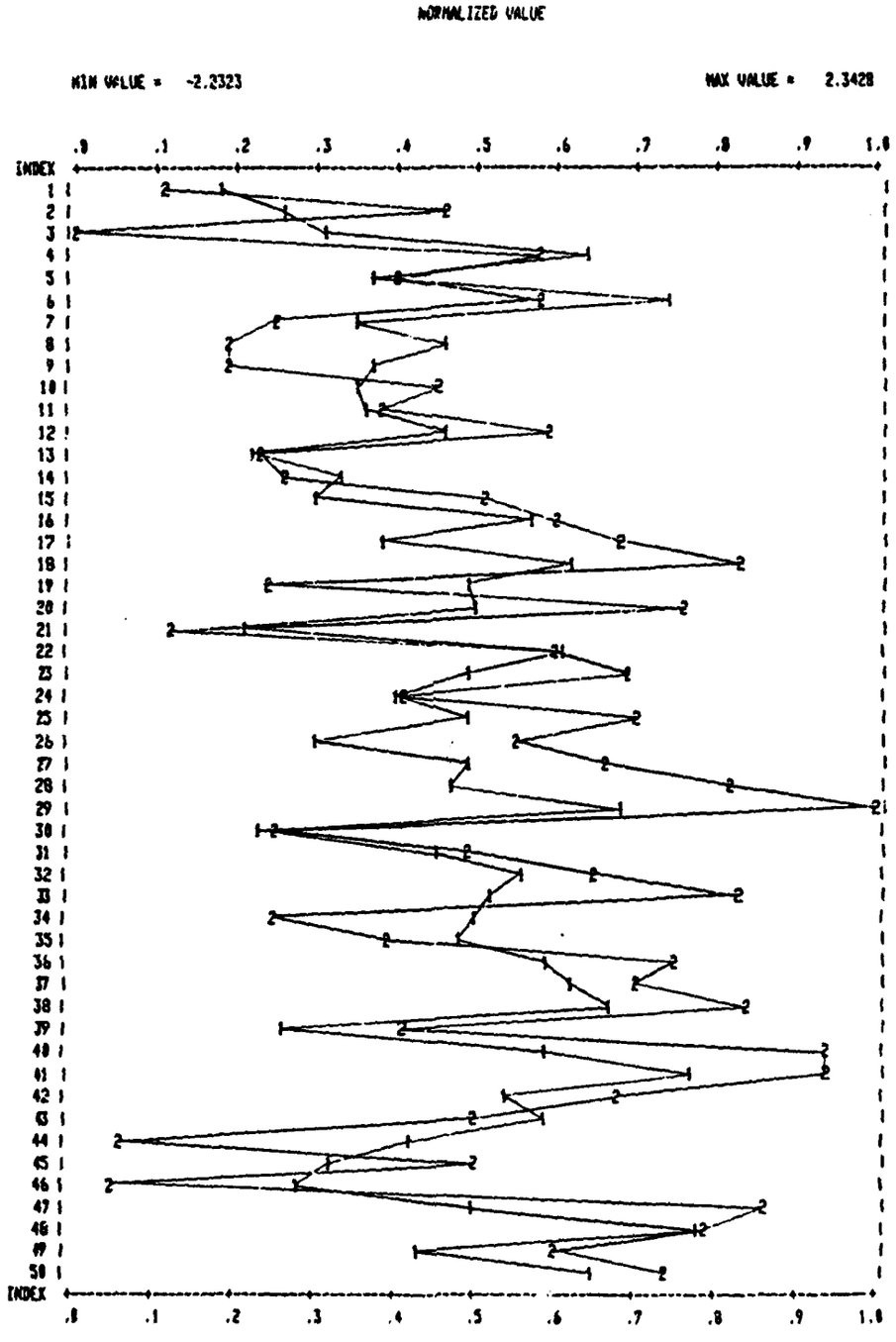


Fig. 5.1.12. The innovations sequences u(t) and $\hat{u}(t)$. Noise variance $\sigma_n^2 = 1.0$.

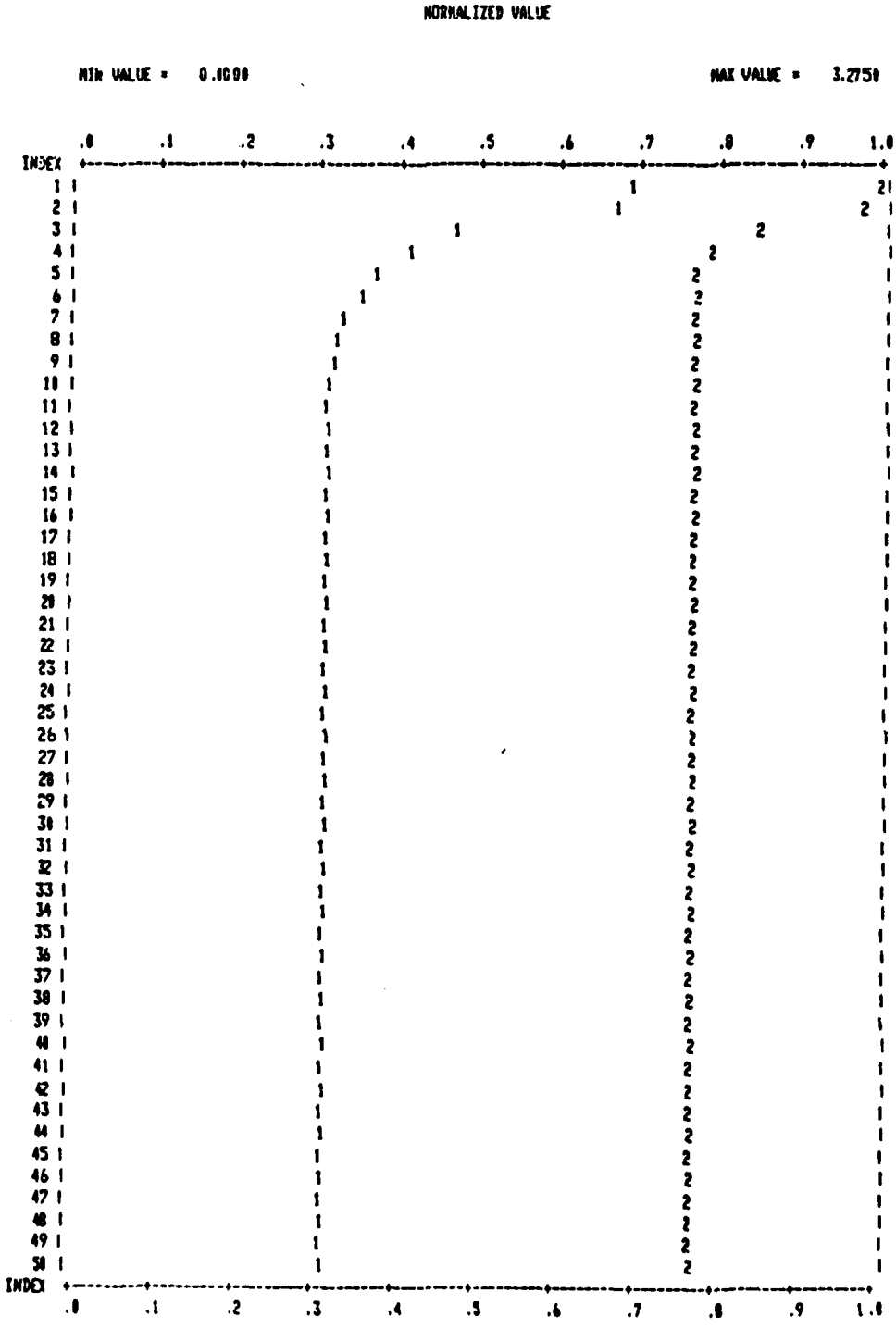


Fig. 5.1.13. The innovations variance v(t) and $\hat{v}(t)$. Noise variance $\sigma_n^2 = 1.0$.

Case #3, $\sigma_n^2 = 2.0$

The variance of the measurement noise is 2.0. Figures 5.1.14 through 5.1.19 show the same kind of result as before. Table 6 gives the initial conditions for the variables we have been looking at for the three different measurements noise level. The values at time t=49 are also given.

5.2 Results Using Fixed Point Arithmetic

In this section we give results from implementing the Kalman filter using fixed point arithmetic. We compare the fixed point results to the corresponding floating point results. The same example is used as in section 5.1:

$$H(z) = \frac{1 - 1.75\ z^{-1} + 0.8\ z^{-2}}{1 - 1.5\ z^{-1} + 1.21\ z^{-2} - 0.4550\ z^{-3}}$$

As in section 5.1 we run the fixed point version of the Kalman filter for three different cases

1) $\sigma_n^2 = 0.0, \quad r(0)/\sigma_n^2 = \infty$
2) $\sigma_n^2 = 1.0, \quad r(0)/\sigma_n^2 = 2.27497$
3) $\sigma_n^2 = 2.0, \quad r(0)/\sigma_n^2 = 1.13749$

The Kalman filter was scaled using the simplified scaling procedure shown in Figure 4.1.6b. The purpose of the scaling is to limit the probability of overflow in the state variables. The scale factor must be chosen large enough to make the scaled input z(t)/s less than +/-1. The necessary scaling of the input can be determined in the same way as for the state vector. The variance of z(t) is $r(0) + \sigma_n^2$. The scale factor for the input is chosen as

$$s = (\delta\sqrt{r(0) + \sigma_n^2})$$

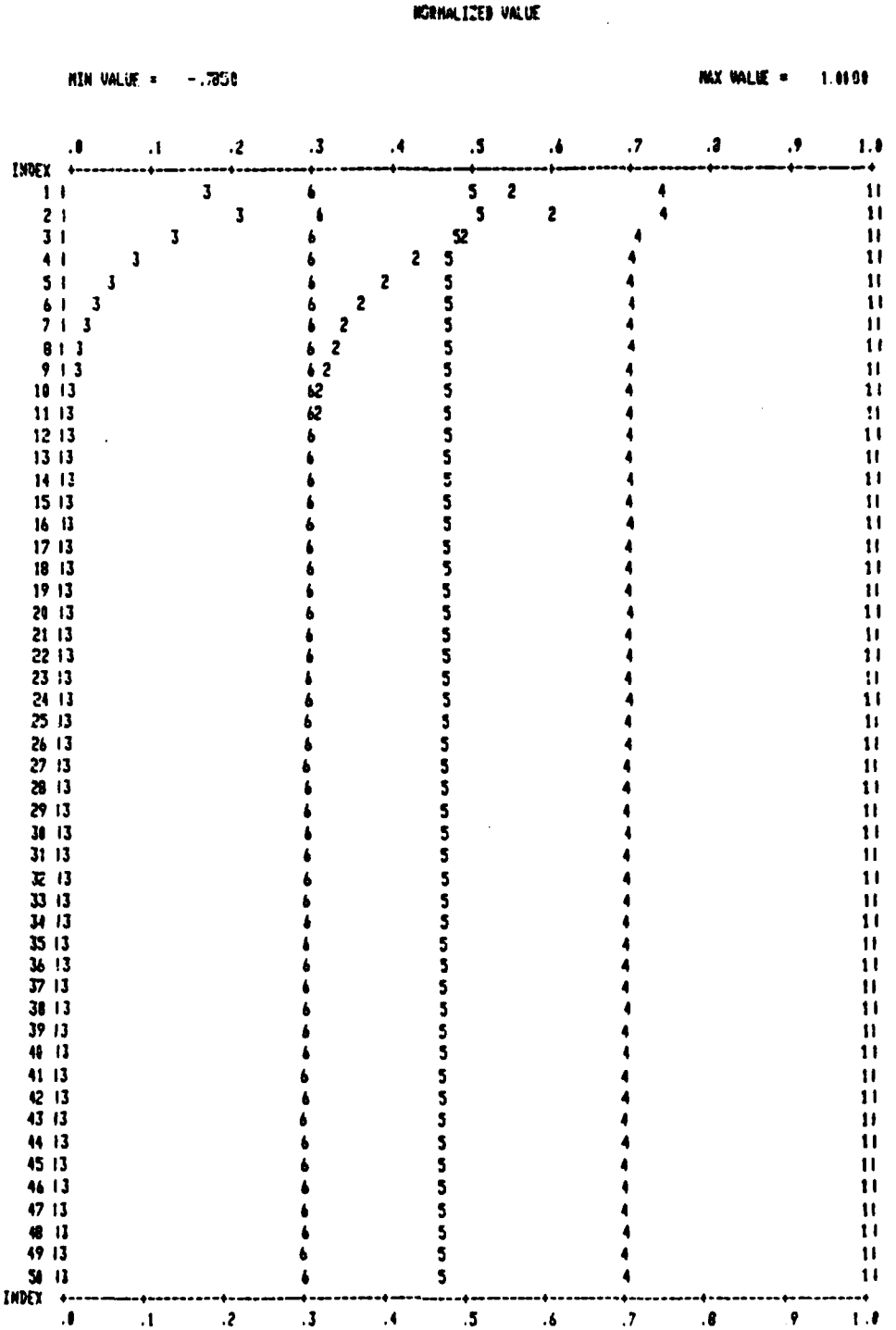


Fig. 5.1.14. The Kalman filter gain vectors $\underline{k}(t)$ and $\hat{\underline{k}}(t)$. Noise variance $\sigma_n^2 = 2.0$.

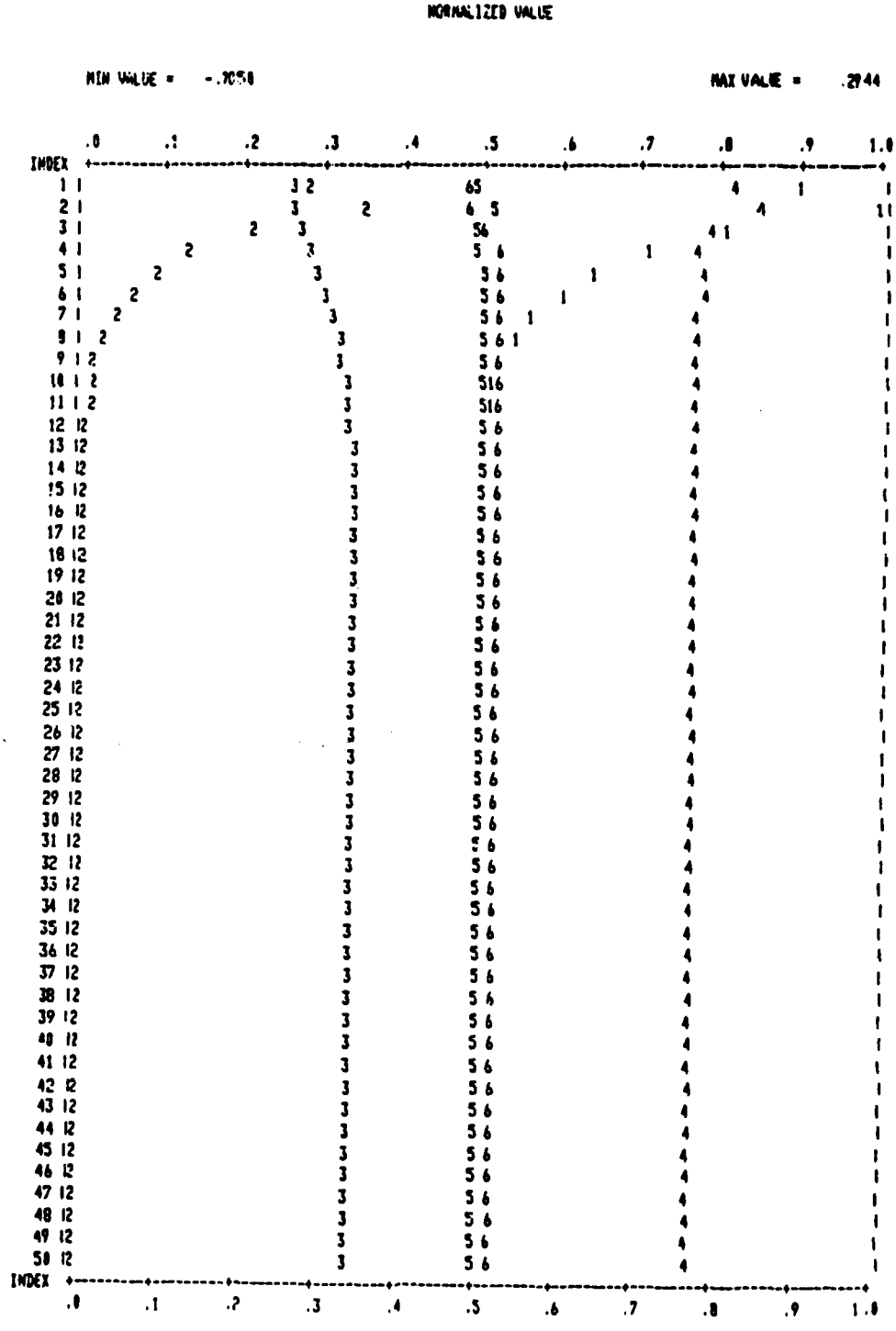


Fig. 5.1.15. The Kalman predictor gain vectors $\underline{k}(t)$ and $\hat{\underline{k}}(t)$. Noise variance $\sigma_n^2 = 2.0$.

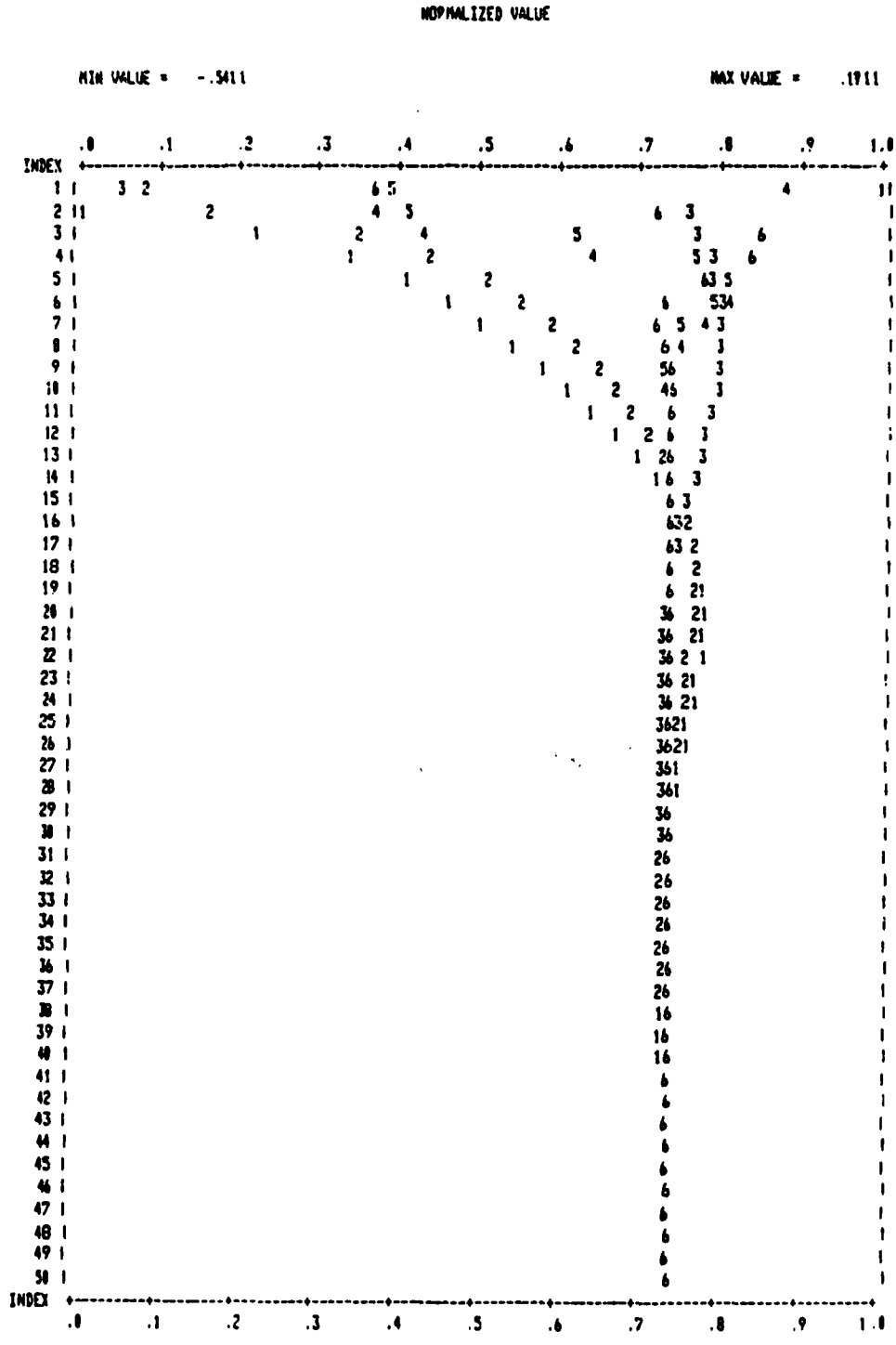


Fig. 5.1.16. The anticausal impulse response vectors $\underline{l}(t)$ and $\hat{\underline{l}}(t)$. Noise variance $\sigma_n^2 = 2.0$.

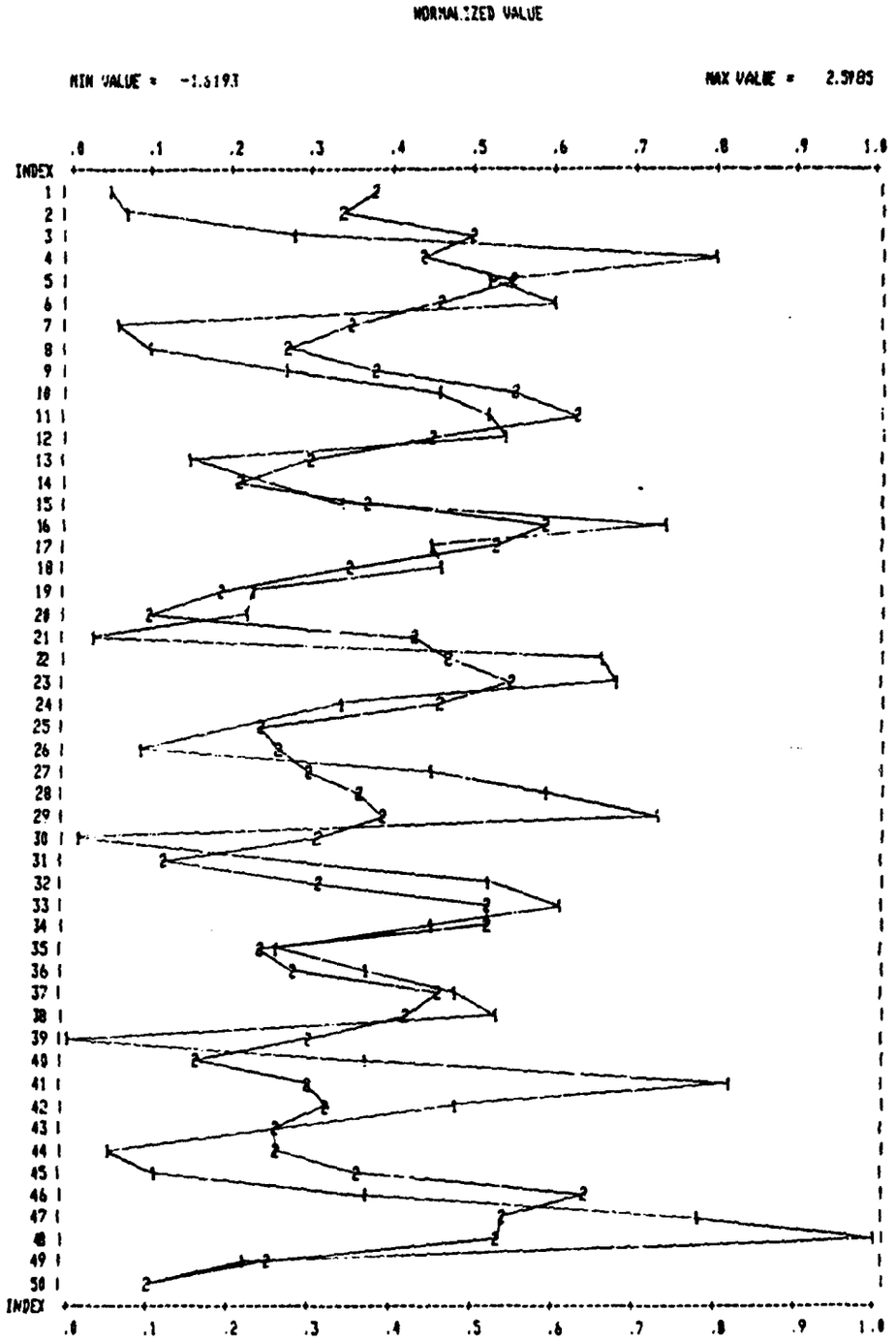


Fig. 5.1.17. The output y(t) and the estimated output $\hat{y}(t)$. Noise variance $\sigma_n^2 = 2.0$.

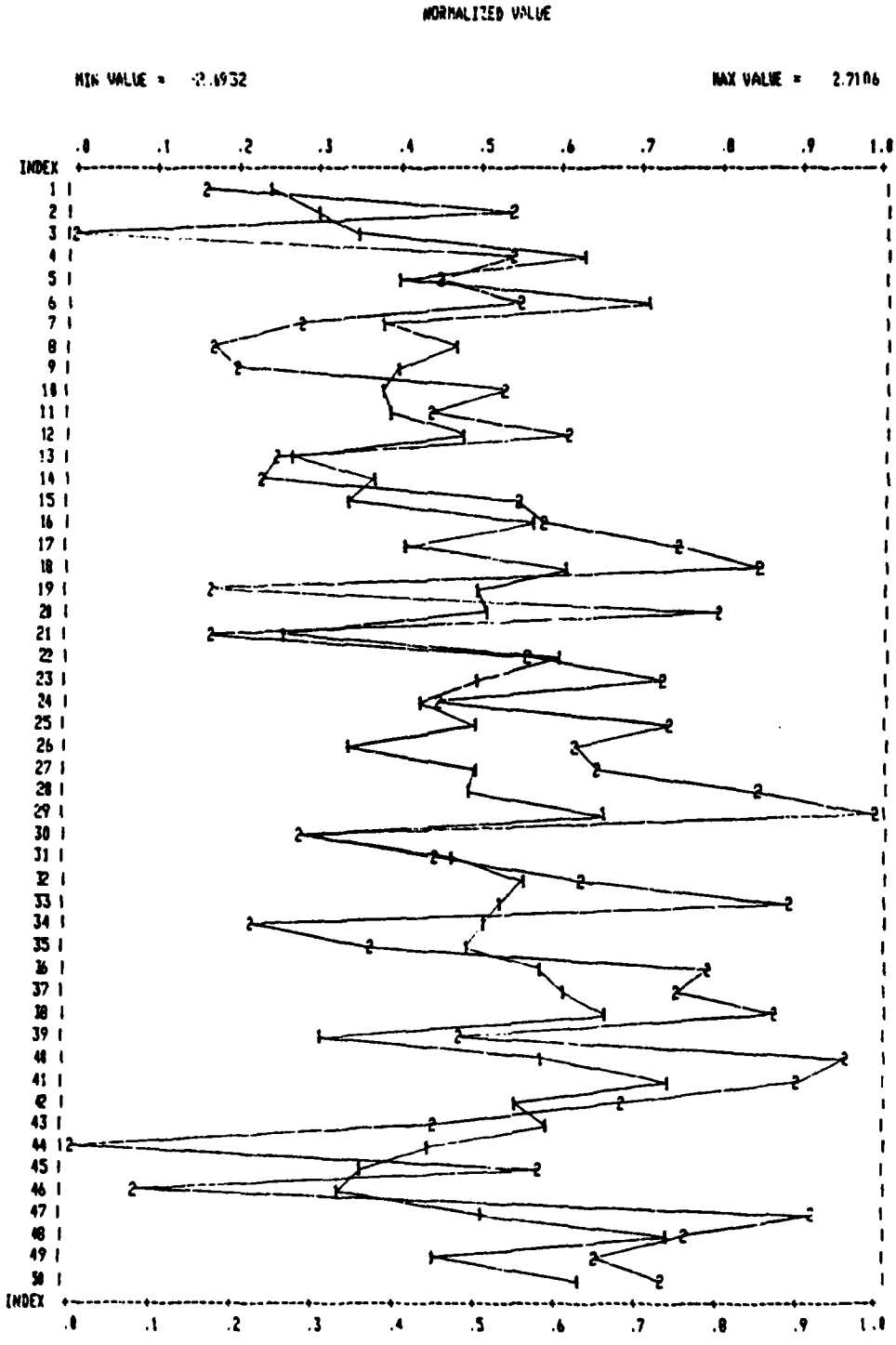


Fig. 5.1.18. The innovations sequences u(t) and $\hat{u}(t)$. Noise variance $\sigma_n^2 = 2.0$.

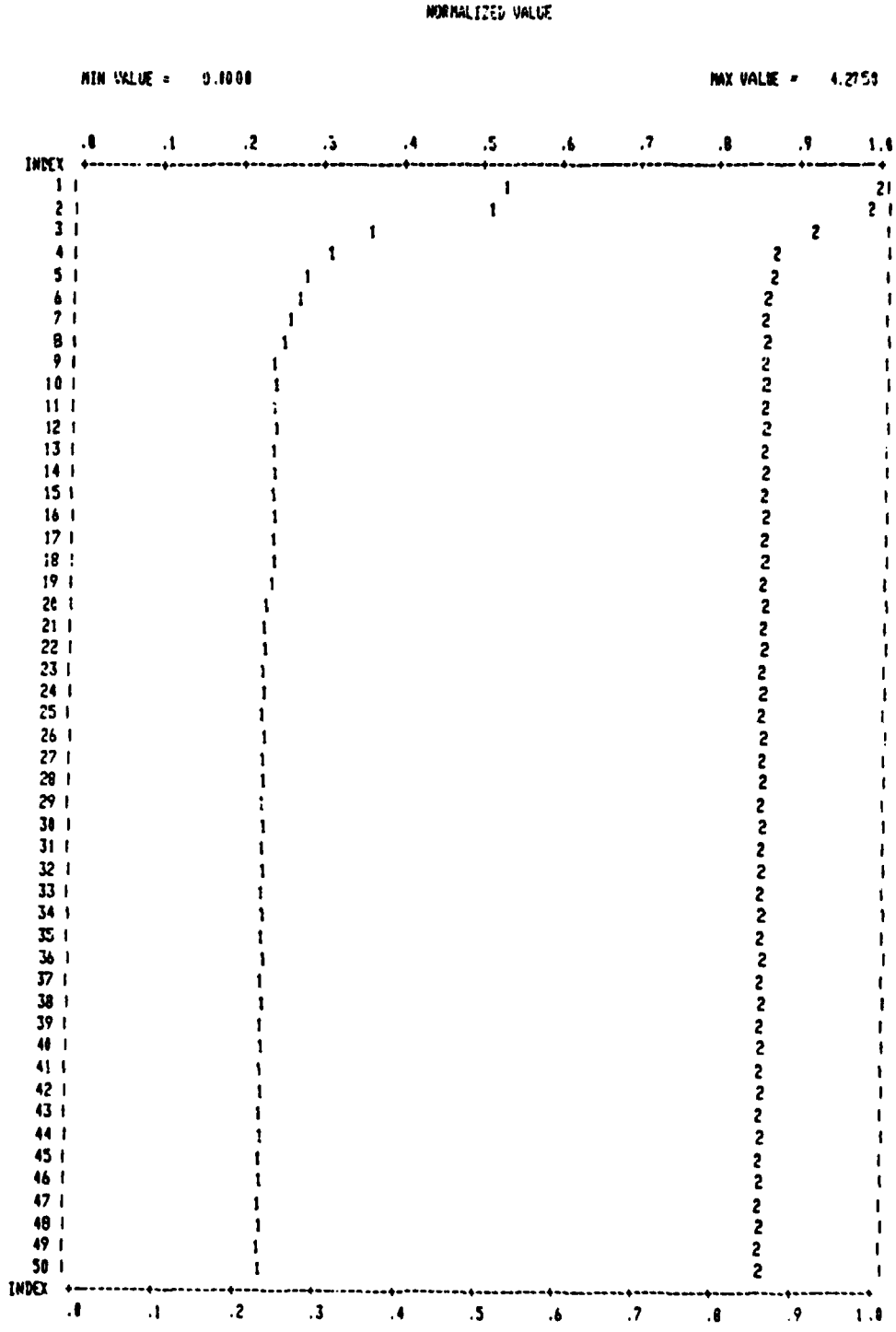


Fig. 5.1.19. The innovations variance v(t) and $\hat{v}(t)$. Noise variance $\sigma_n^2 = 2.0$.

Table 6. Results from the Kalman filter.

	Initial Values t = 0			Values at Time t = 49		
	$\sigma_n^2 = 0.0$	$\sigma_n^2 = 1.0$	$\sigma_n^2 = 2.0$	$\sigma_n^2 = 0.0$	$\sigma_n^2 = 1.0$	$\sigma_n^2 = 2.0$
			Floating Point Calculations			
$\underline{\cdot}(t)$	1.00000	0.69465	0.53216	1.00000	0.60495	0.45686
	0.19107	0.13272	0.10168	-0.24994	0.04156	0.05004
	-0.48481	-0.33678	-0.25800	-0.78495	-0.34615	-0.24812
$\underline{\cdot}(t)$	0.19107	0.13272	0.10168	-0.24994	0.04156	0.05004
	-0.48481	-0.33678	-0.25800	-0.78495	-0.34615	-0.24812
	-0.50341	-0.34969	-0.26789	-0.42001	-0.29425	-0.22486
$\underline{1}(t)$	0.19107	0.13272	0.10168	0.00104	0.00000	0.00000
	-0.48481	-0.33678	-0.25800	0.00076	0.00000	0.00000
	-0.50341	-0.34969	-0.26789	-0.00011	0.00000	0.00000
v(t)	2.27497	3.27497	4.27497	1.00006	2.53134	3.68229
			Fixed Point Calculations			
$\underline{k}(t)$	0.99997	0.69464	0.53217	0.99997	0.60492	0.45691
	0.19107	0.13272	0.10168	-0.22894	0.04199	0.05029
	-0.48477	-0.33673	-0.25797	-0.76813	-0.34485	-0.24725
$\underline{k}(t)$	0.19107	0.13272	0.10168	-0.22894	0.04199	0.05029
	-0.48477	-0.33673	-0.25797	-0.76813	-0.34435	-0.24725
	-0.50317	-0.34949	-0.26758	-0.42004	-0.29272	-0.22375
$\underline{1}(t)$	0.19107	0.13272	0.10168	0.00503	0.00140	0.00098
	-0.48477	-0.33673	-0.25797	0.00421	0.00140	0.00098
	-0.50317	-0.34949	-0.26758	0.00000	0.00098	0.00073
v(t)	2.2749	3.2749	4.2748	1.0149	2.5230	3.6707

In the case $\sigma_n^2 = 2.0$ the variance of z(t) becomes 4.27497 and then

$$s = 2.0676\delta$$

Choosing s = 4.00 corresponds to $\delta = 1.9$ which should be fine. The denominator coefficients are stored at their half value since the dynamic range is limited by +/-1.0 and the largest denominator coefficient is -1.5. The same scalefactor is used in all three cases, s=4.0. Also we used truncation instead of rounding.

Case #1, $\sigma_n^2 = 0$

Figures 5.2.1 through 5.2.6 show plots of $\hat{\underline{k}}(t)$, $\underline{\hat{k}}(t)$, $\hat{\underline{l}}(t)$, $\hat{y}(t)$, $\hat{u}(t)$, and $\hat{v}(t)$ calculated using floating and fixed point arithmetic. The lower numbers correspond to the floating point calculations. As can be seen the fixed point results follow the corresponding floating point results closely. These are very positive results.

Case #2, $\sigma_n^2 = 1.0$

Figures 5.2.7 through 5.7.12 show both the floating and fixed point results in the case the measurement noise variance is 1.0.

Case #3, $\sigma_n^2 = 2.0$

Figures 5.2.13 through 5.2.18 show the same thing as Figures 5.2.7 through 5.2.12. Now the measurement noise variance is 2.0. If anything the measurement noise seems to improve the accuracy of the fixed point calculation compared with the floating point ones. It is also worth noting that the rate of convergence increases with increased measurement noise. Compare for example Figures 5.2.1, 5.2.7, and 5.2.13. Table 6 gives the initial values for $\hat{\underline{k}}(t)$, $\underline{\hat{k}}(t)$, $\hat{\underline{l}}(t)$, and $\hat{v}(t)$ for both the fixed p int and floating point calculations. The

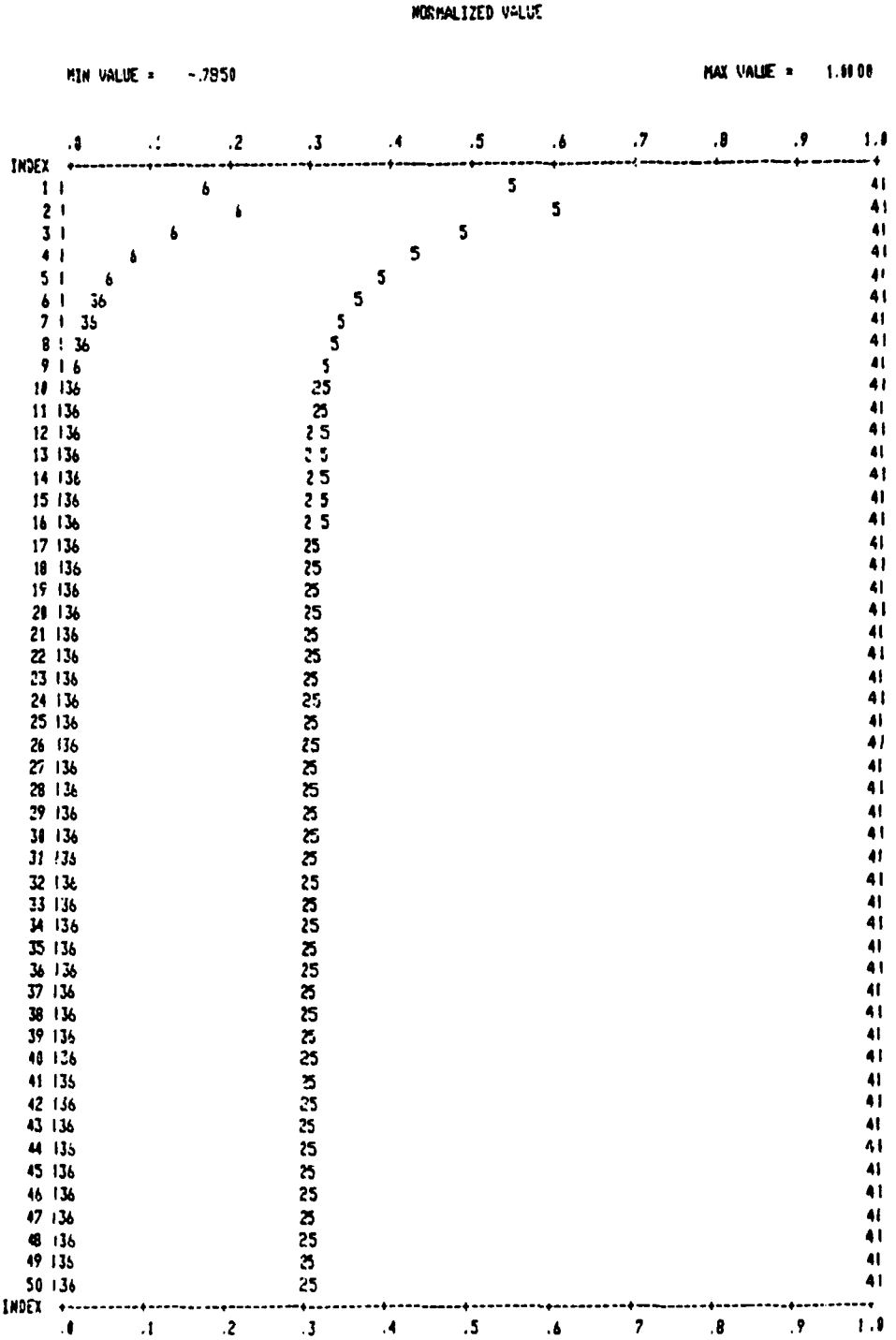


Fig. 5.2.1. The Kalman filter gain vector $\hat{\underline{k}}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 0$.

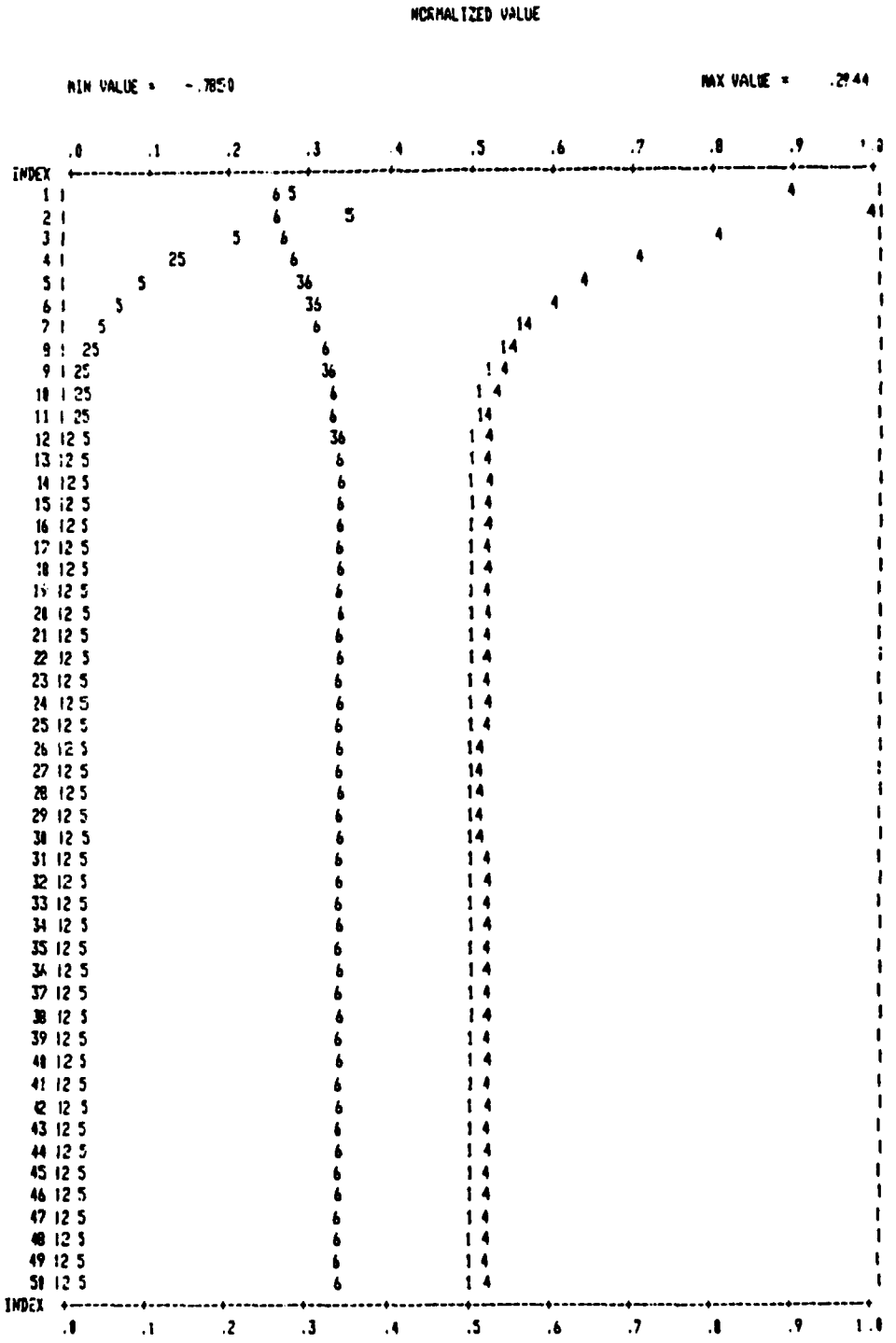


Fig. 5.2.2. The Kalman predictor gain vector $\underline{\hat{k}}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 0$.

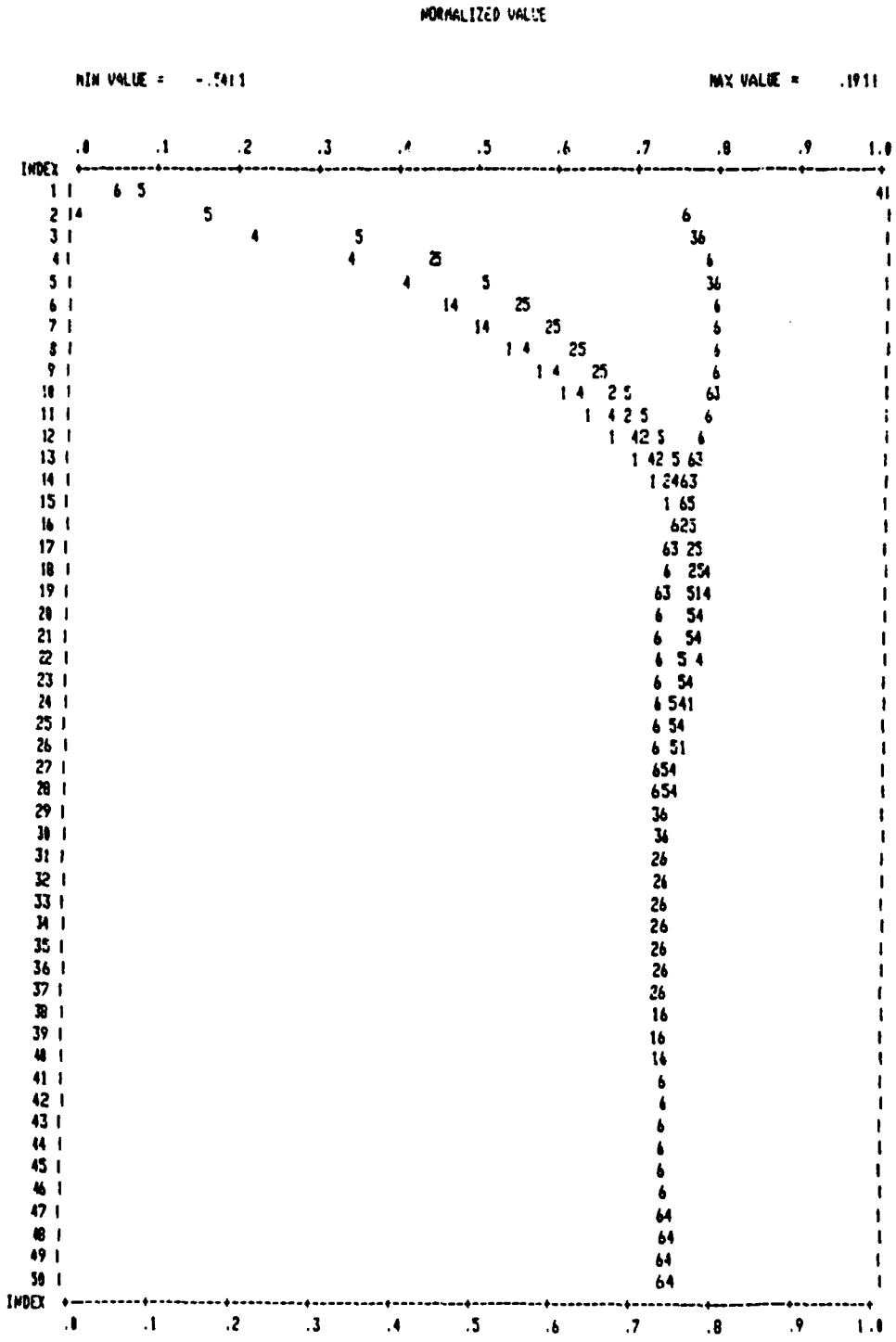


Fig. 5.2.3. The anticausal impulse response vector $\hat{\underline{l}}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 0$.

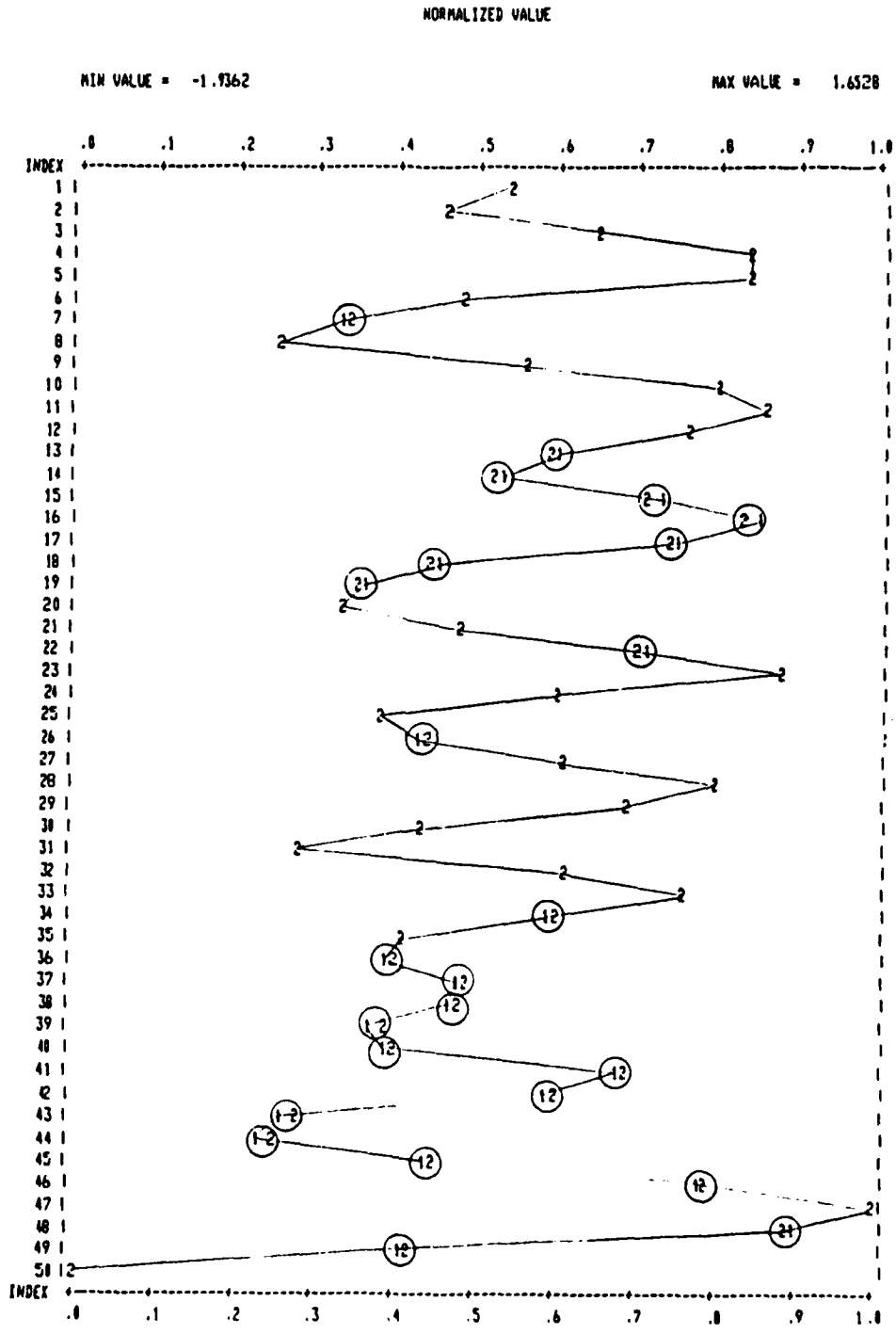


Fig. 5.2.4. The estimated output $\hat{y}(t)$ calculated using floating point and fixed point arithmetic. Noise variance $\sigma_n^2 = 0$.

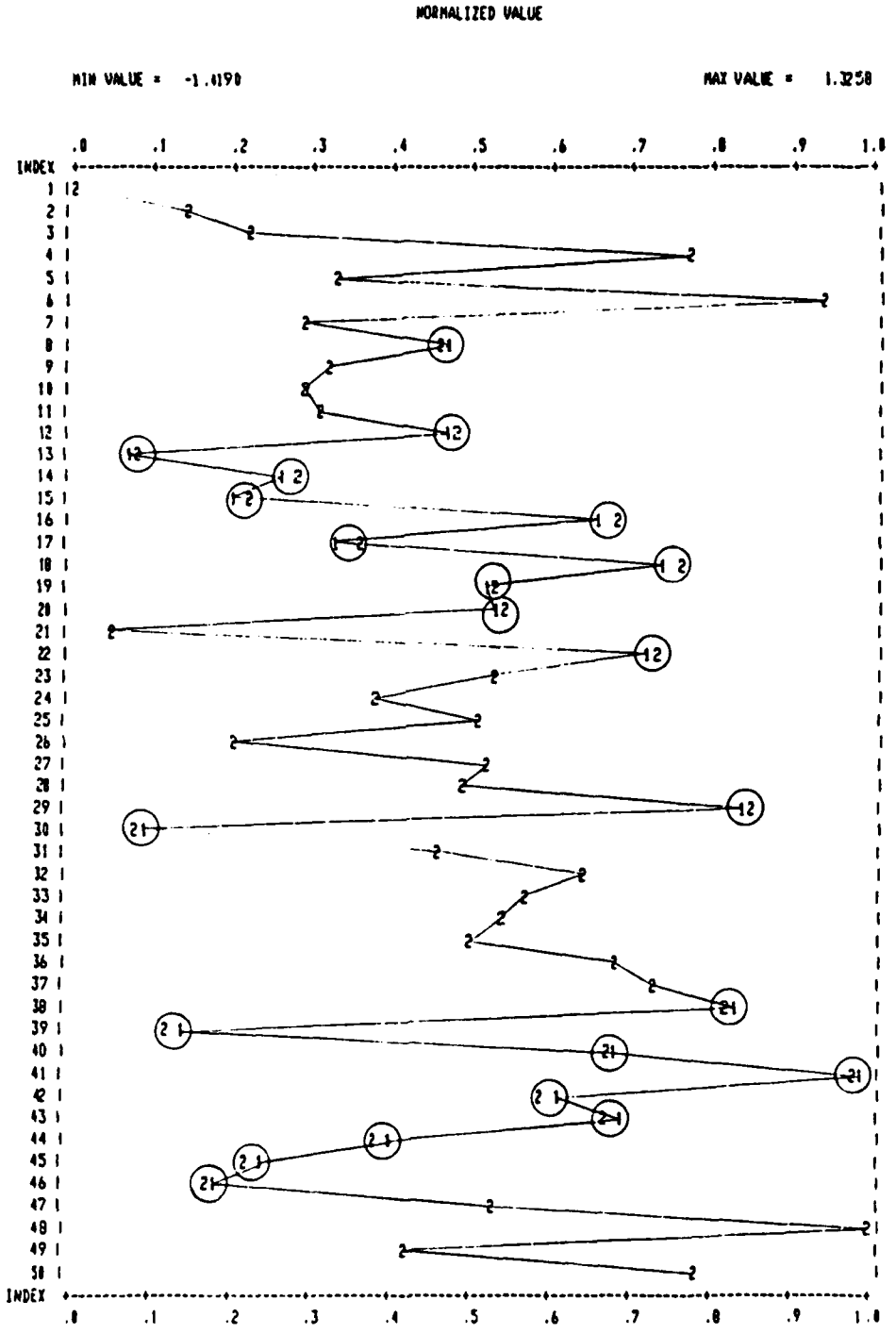


Fig. 5.2.5. The innovations sequence $\hat{u}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 0$.

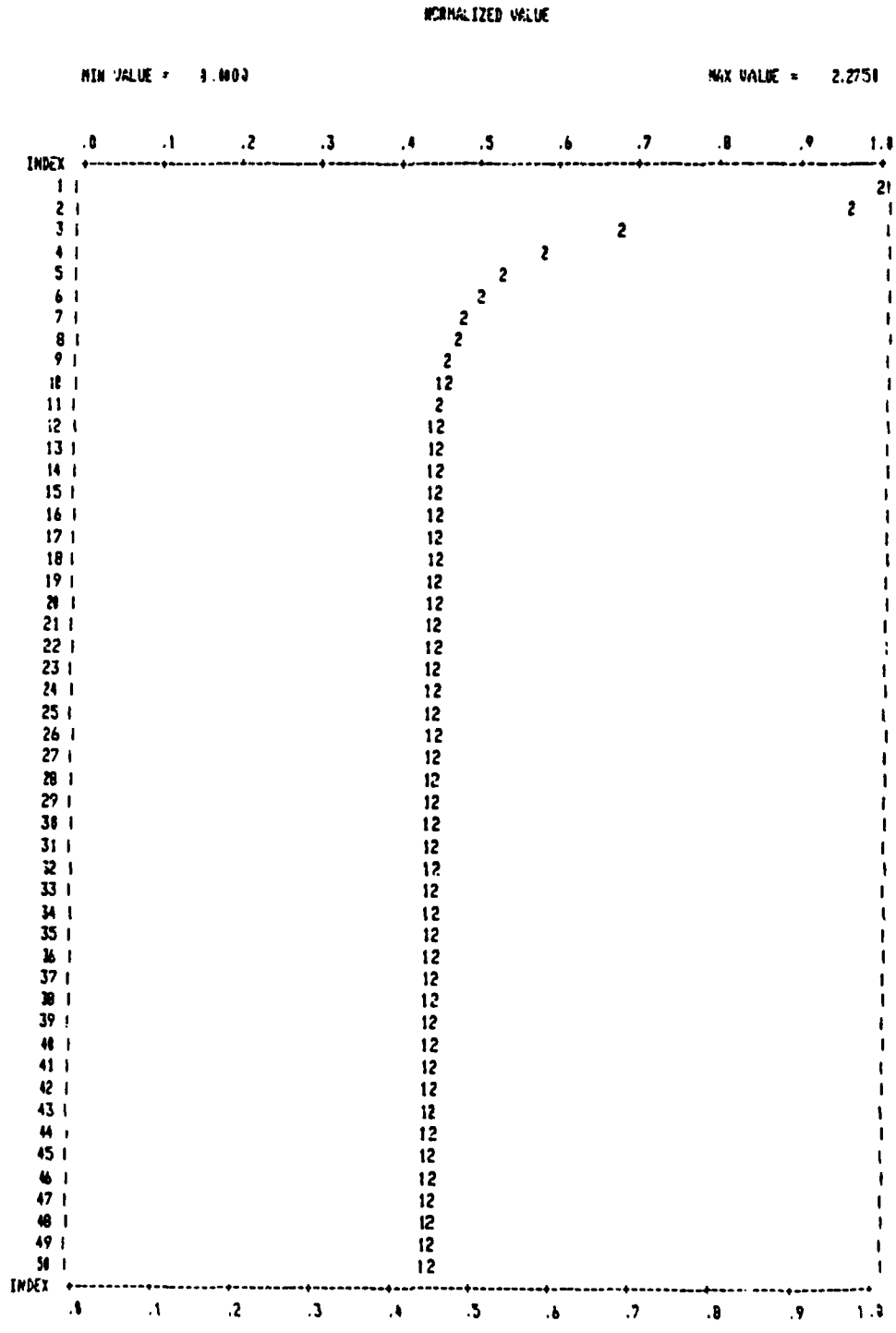


Fig. 5.2.6. The innovations variance $\hat{u}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 0$.

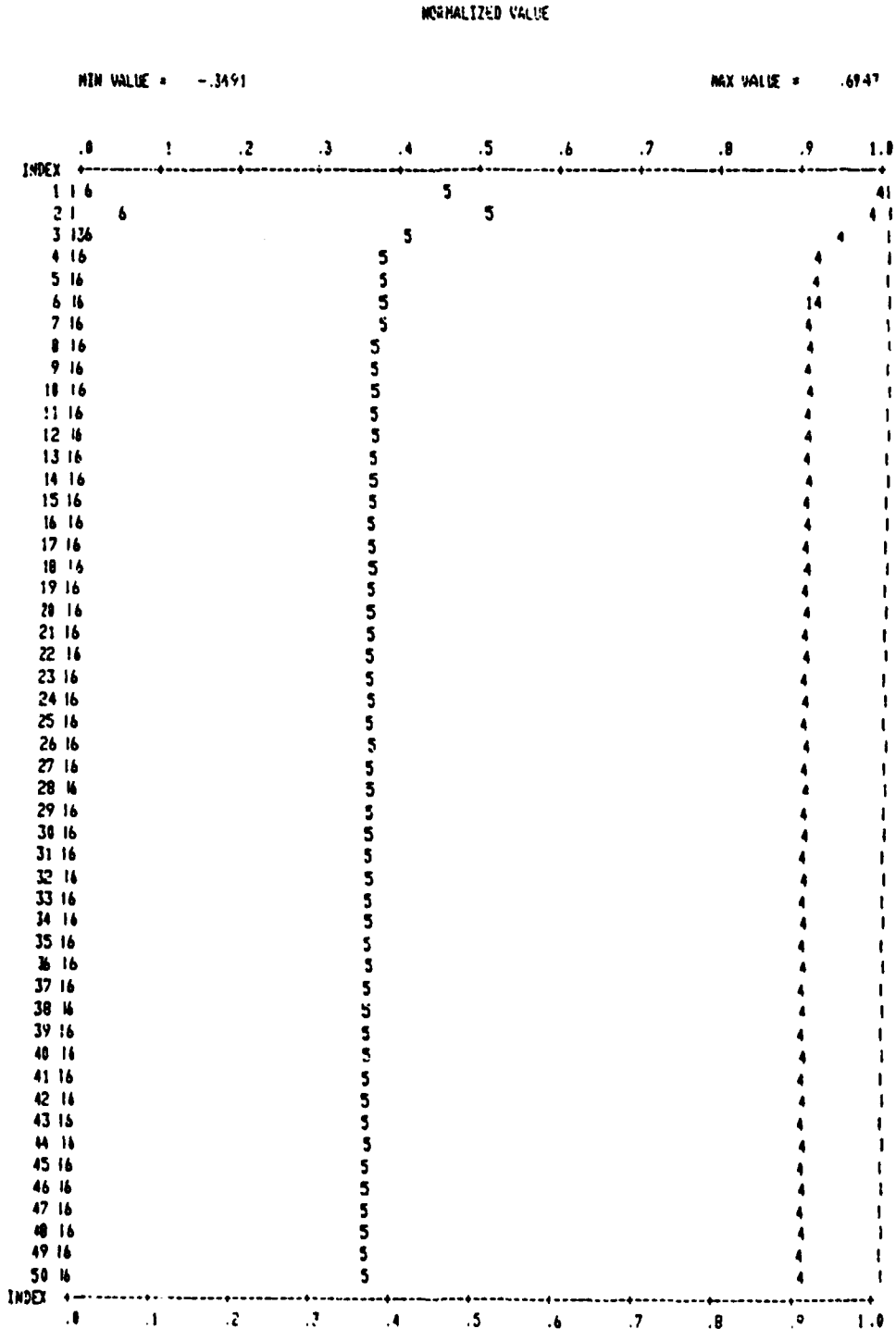


Fig. 5.2.7. The Kalman gain vector $\underline{\hat{k}}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 1.0$.

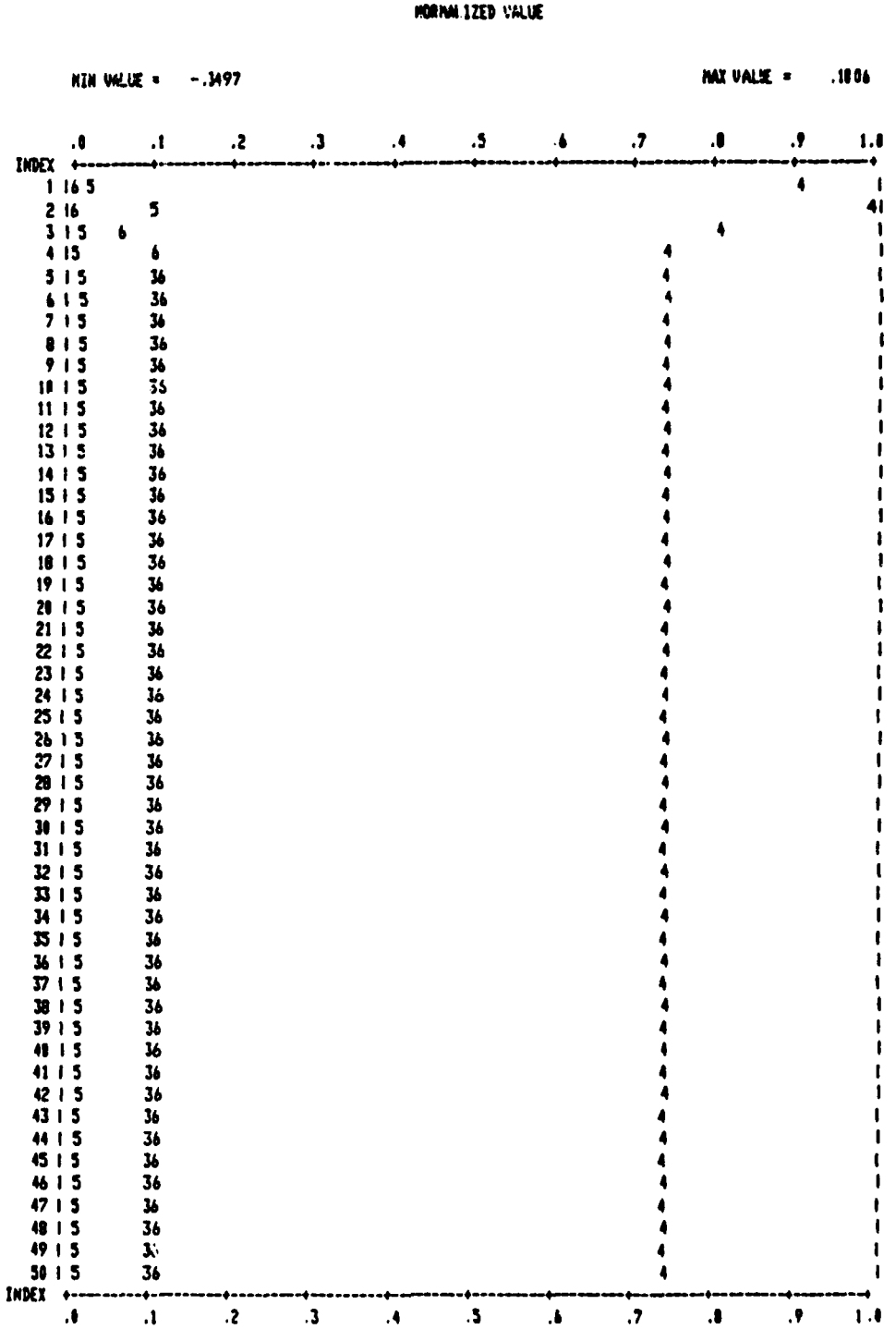


Fig. 5.2.8. The Kalman predictor gain vector $\underline{k}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 1.0$.

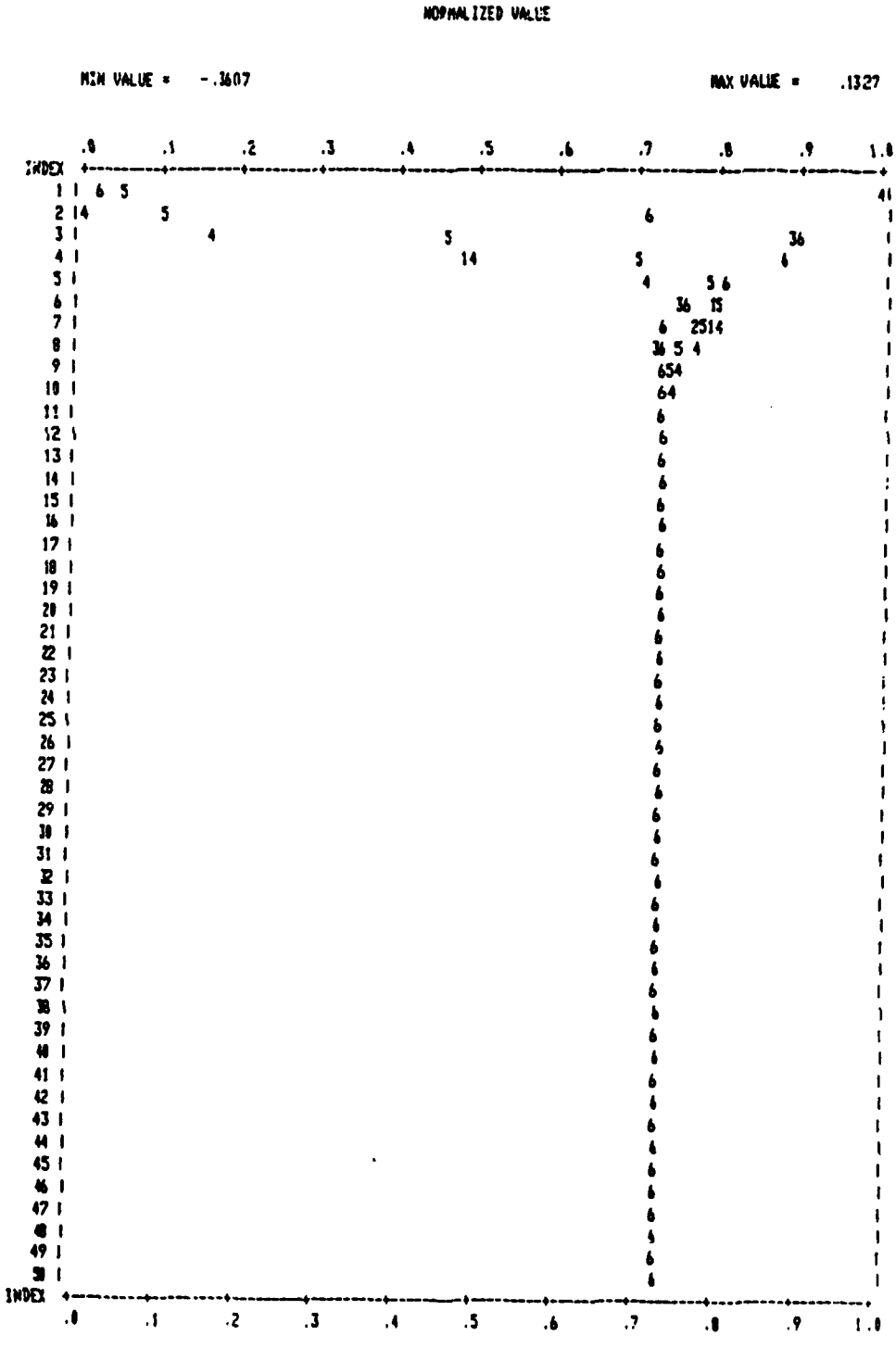


Fig. 5.2.9. The anticausal impulse response vector $\hat{\underline{i}}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 1.0$.

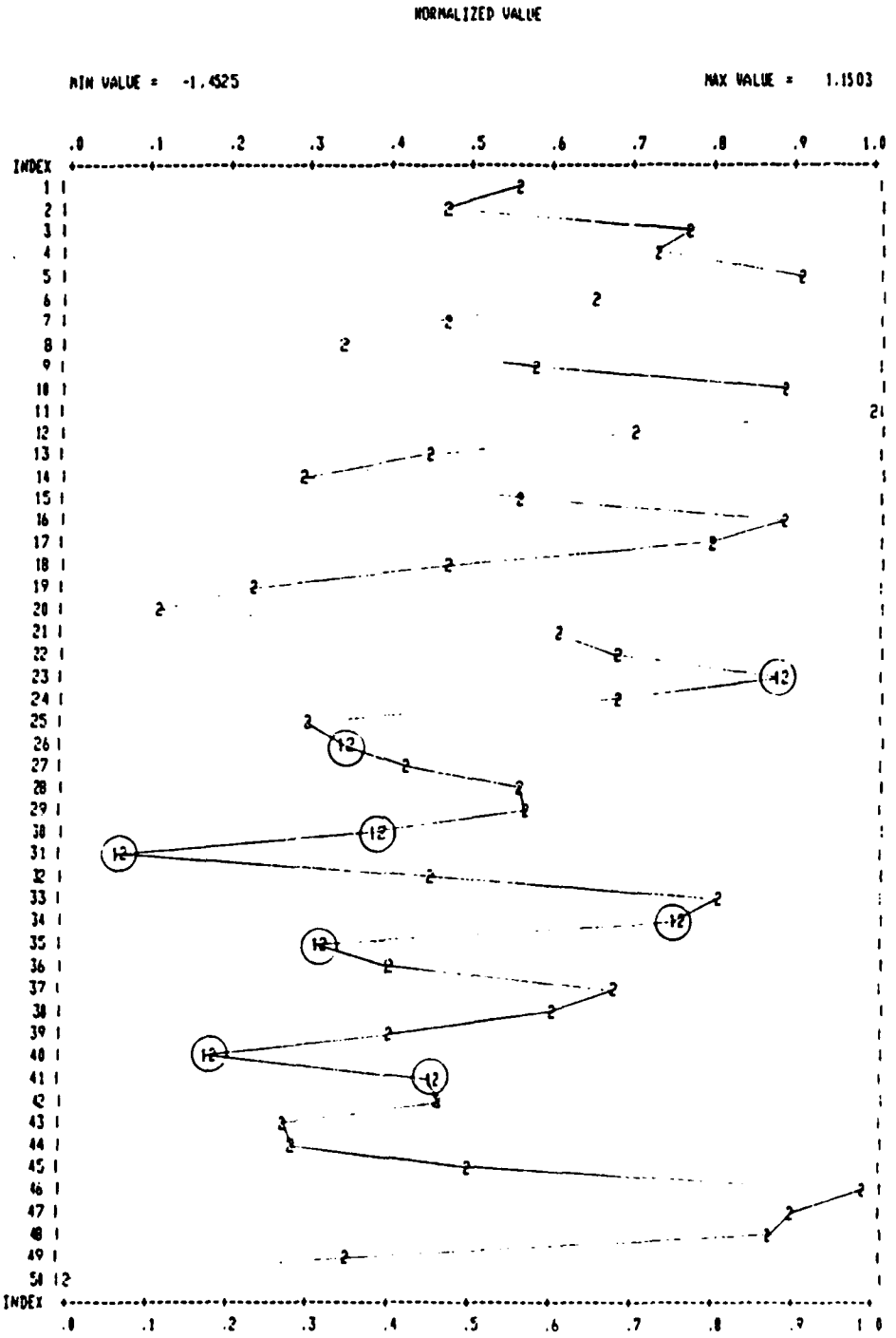


Fig. 5.2.10. The estimated output $\hat{y}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 1.0$.

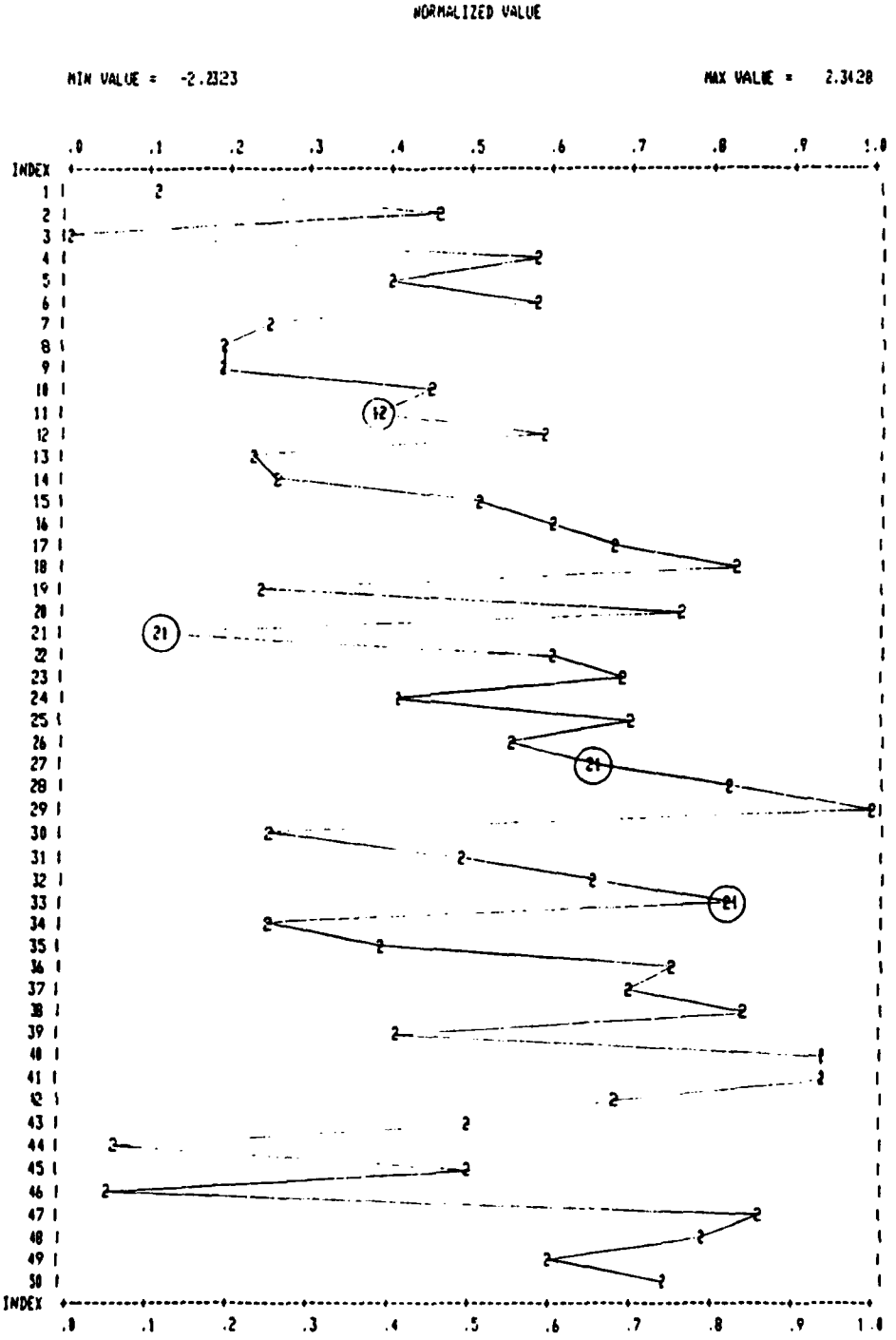


Fig. 5.2.11. The innovations sequence $\hat{u}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 1.0$.

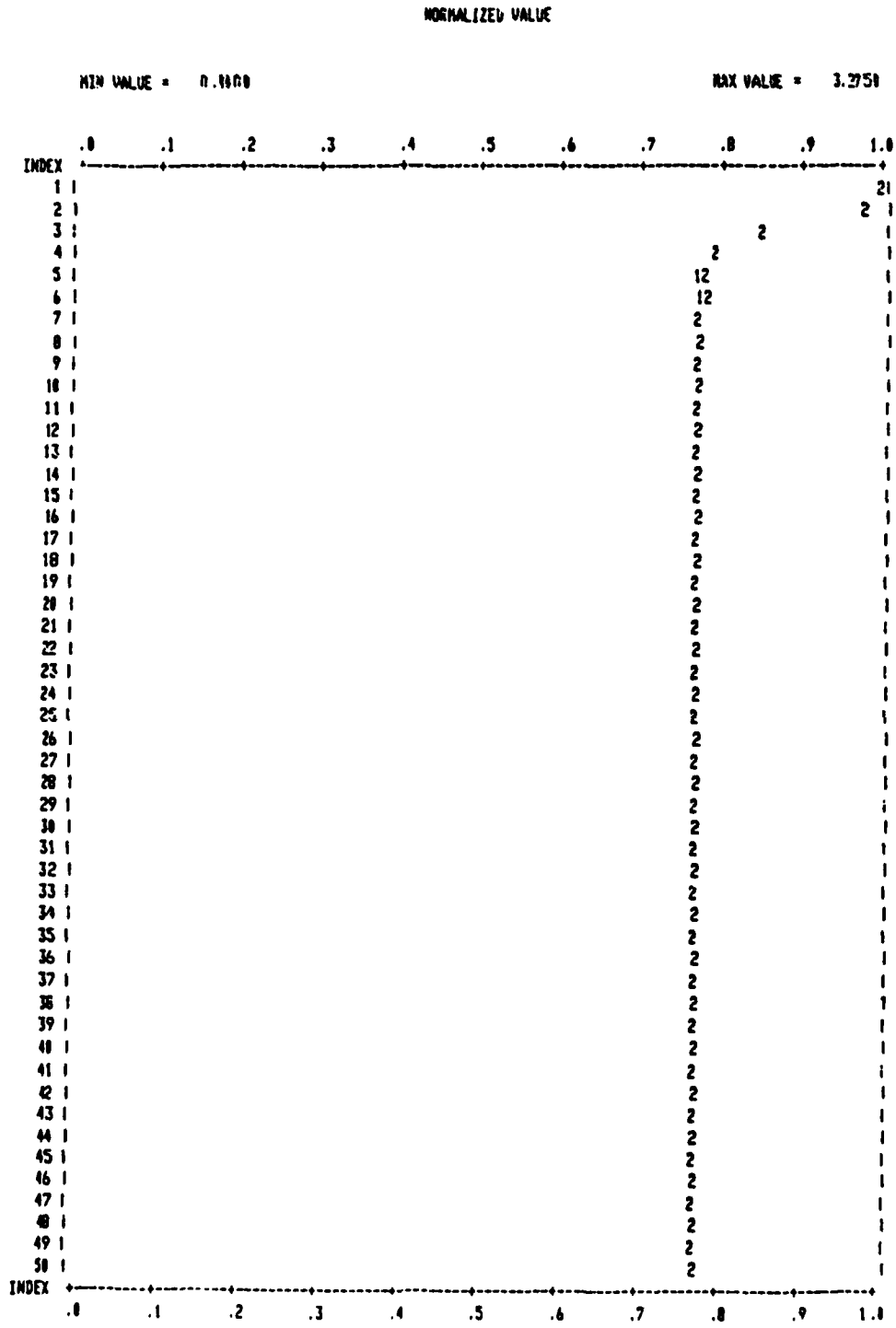


Fig. 5.2.12. The innovations variance $\hat{v}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 1.0$.

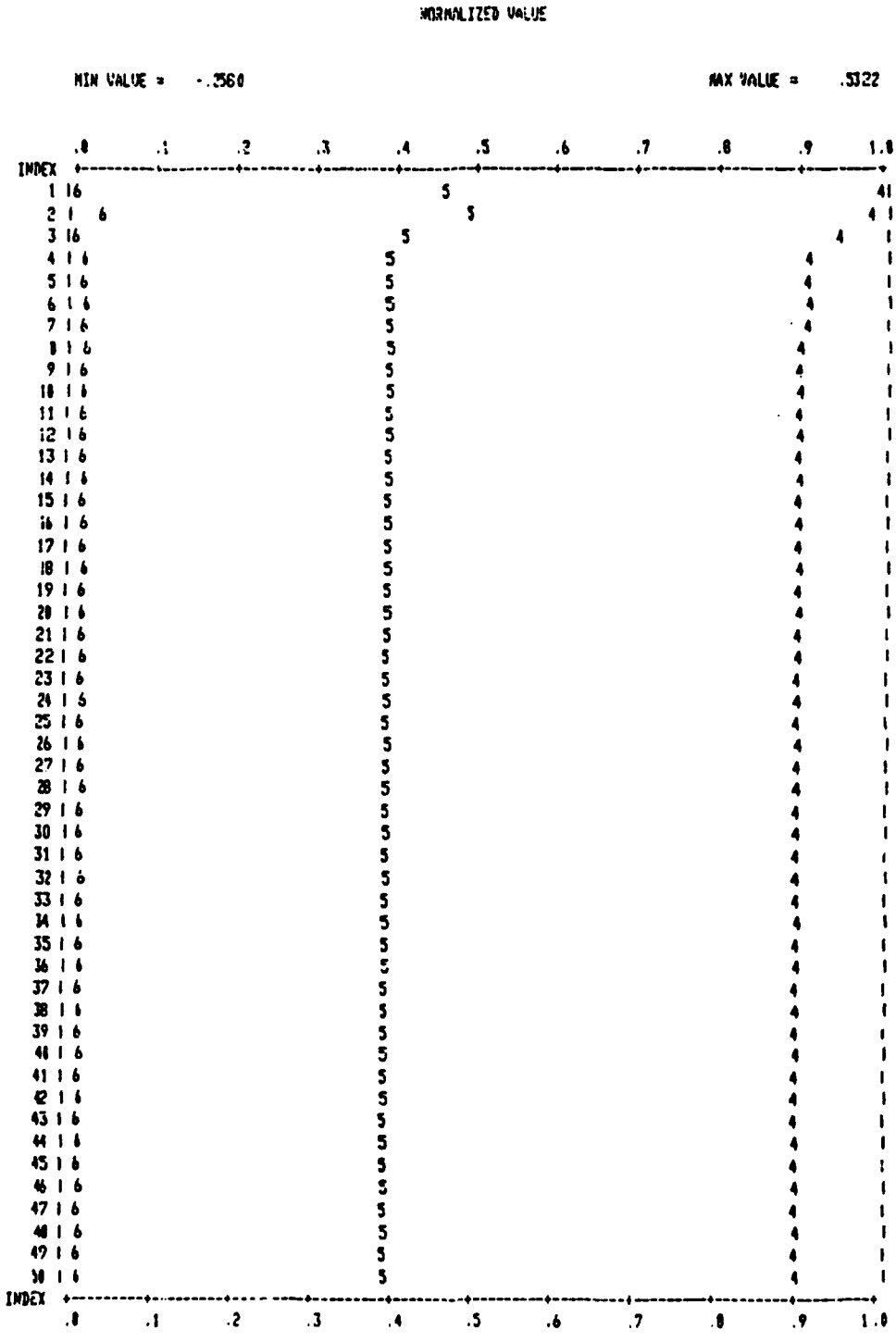


Fig. 5.2.13. The Kalman filter gain vector $\underline{\hat{k}}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 2.0$.

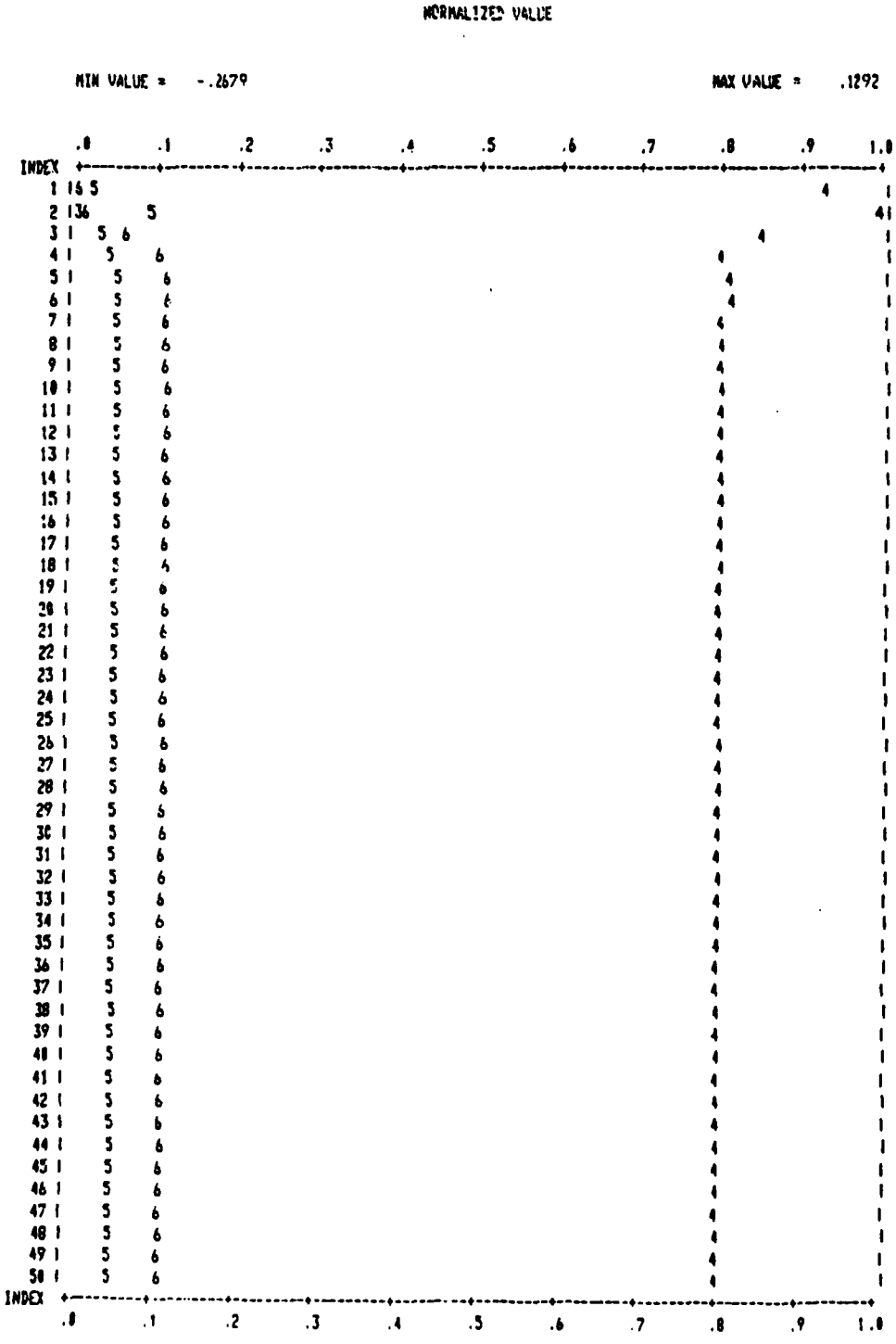


Fig. 5.2.14. The Kalman predictor gain vector $\underline{k}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 2.0$.

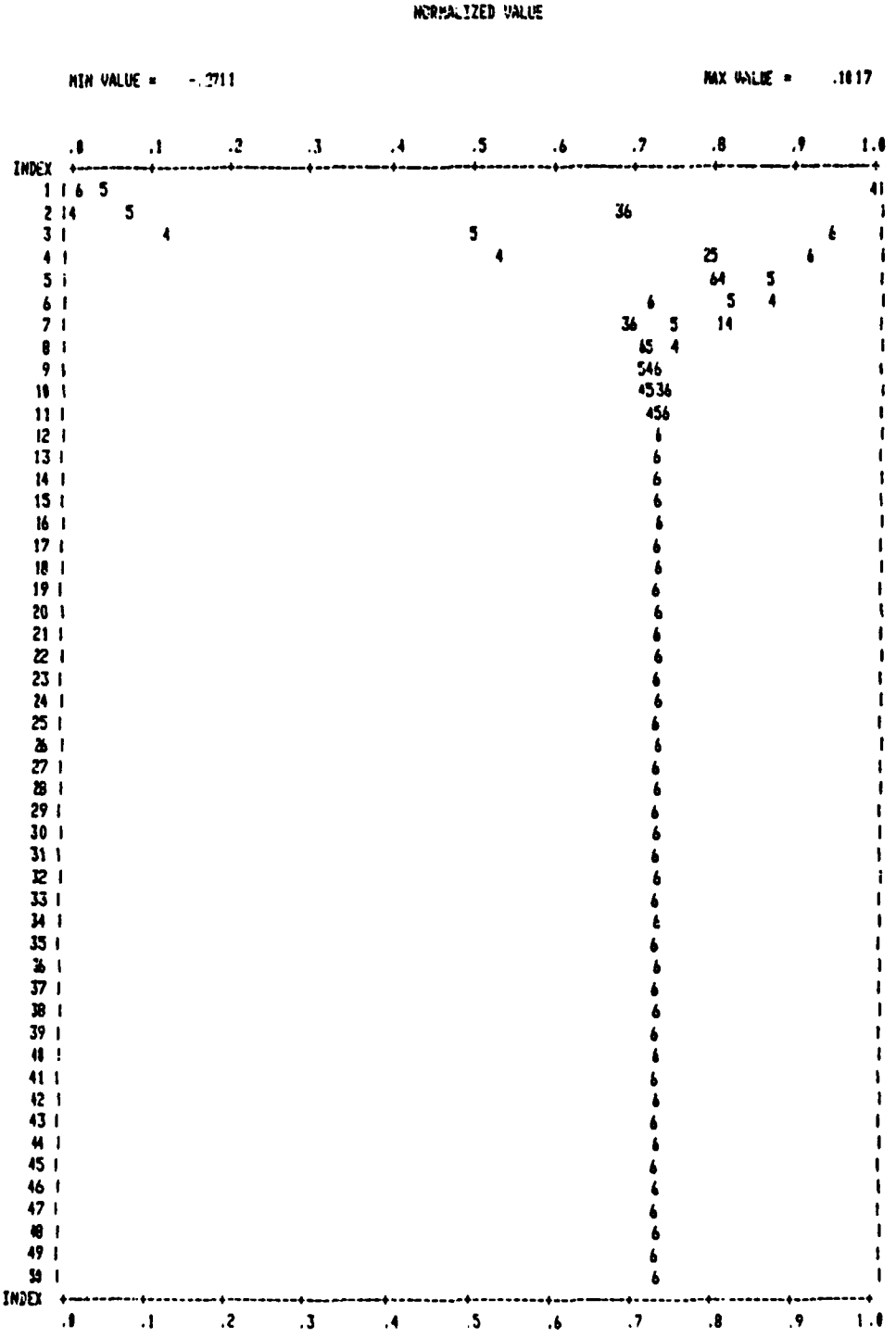


Fig. 5.2.15. The anticausal impulse response vector $\hat{\underline{l}}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 2.0$.

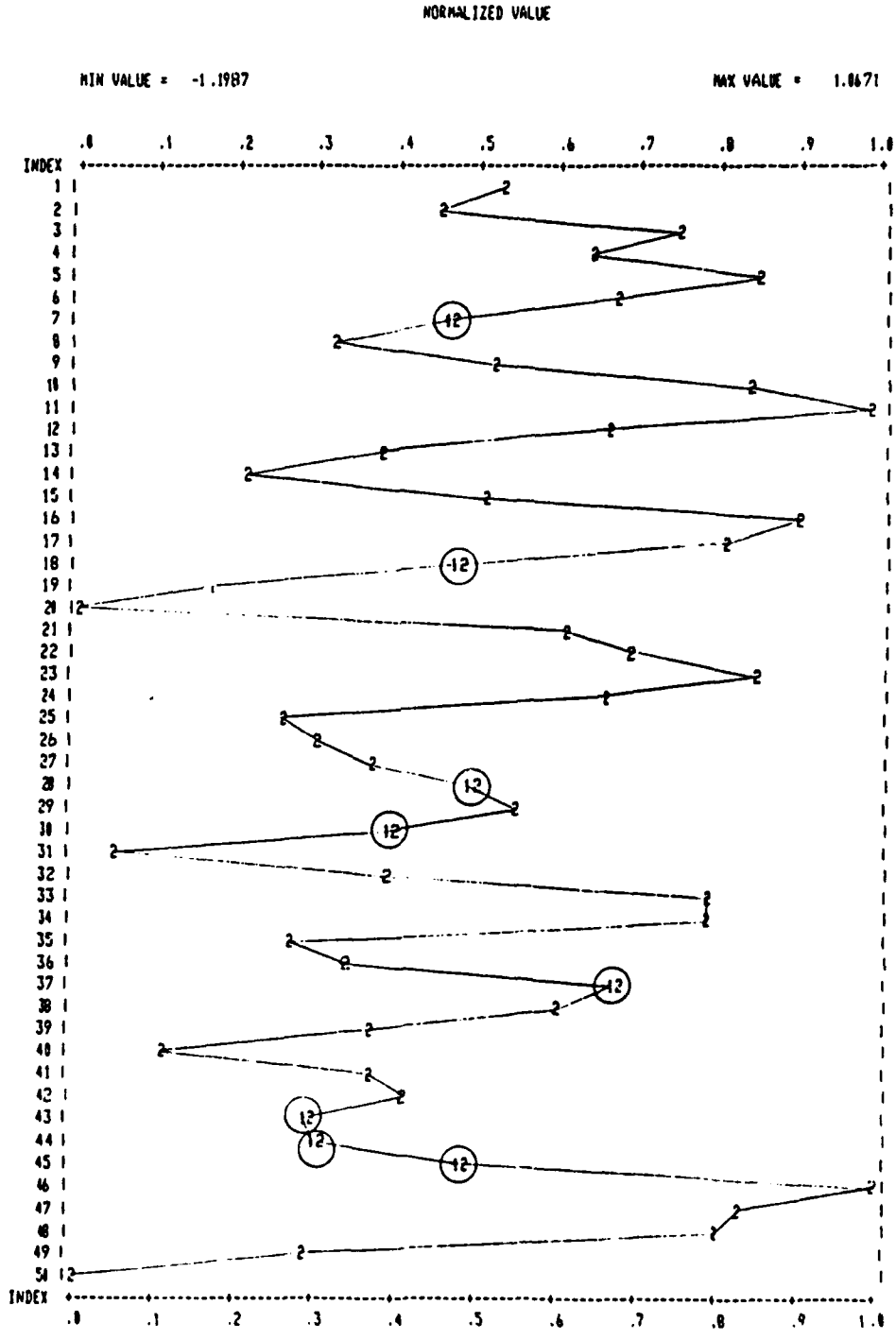


Fig. 5.2.16. The estimated output $\hat{y}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 2.0$.

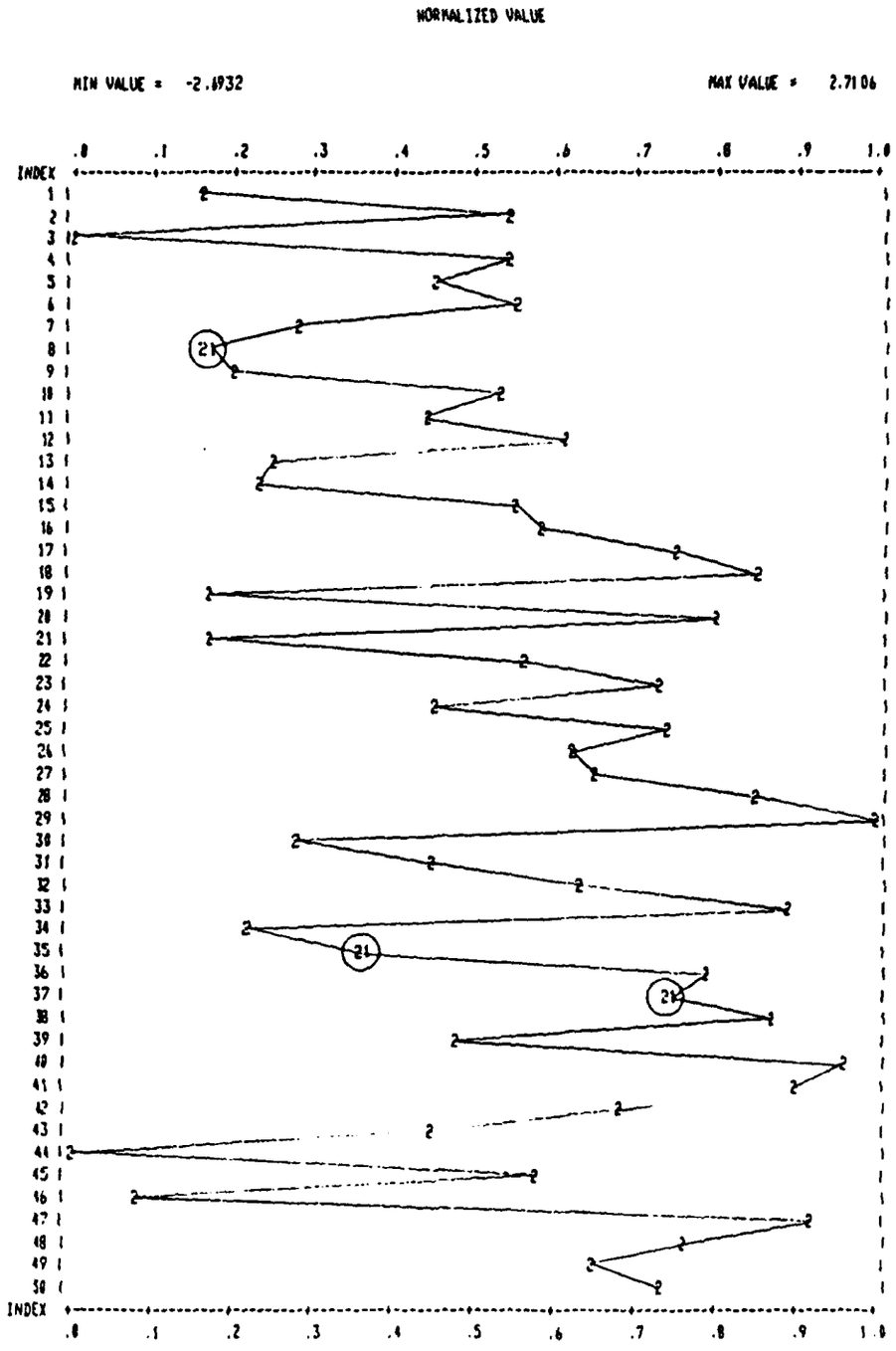


Fig. 5.2.17. The innovations sequence $\hat{u}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 2.0$.

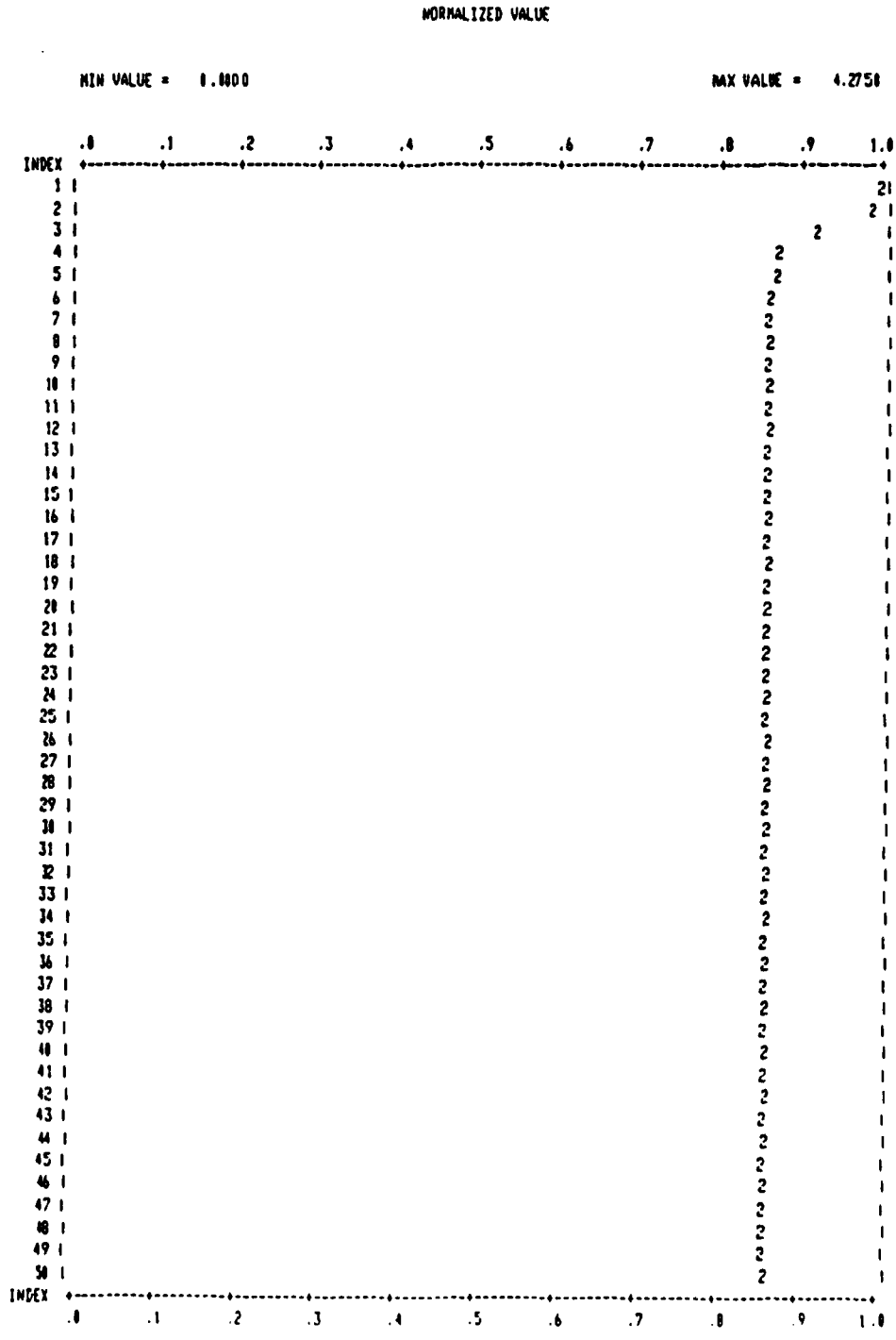


Fig. 5.2.18. The innovations variance $\hat{v}(t)$ calculated using floating and fixed point arithmetic. Noise variance $\sigma_n^2 = 2.0$.

values at time t=49 are also given. One of the major positive conclusions of this thesis is that these fixed point results compare very closely to the floating point results. Measurement noise is desirable since it enhances the performance of the fixed point version of the Kalman filter.

CHAPTER 6

CONCLUSIONS

In this thesis we have studied Kalman filtering of ARMA processes both from a theoretical and from a practical point of view. Numerical results have been presented for the Kalman filter implemented in floating point and fixed point arithmetic.

The Kalman predictor and Kalman filter are derived as inverses to the Innovations representations #1 and #2, respectively. The Innovations representations are linear time-varying state space models which are asymptotically time-invariant. In the limit the two converge to the so-called Markovian representations #1 and #2, respectively. As a result the Kalman predictor and filter converge to the Markovian predictor and filter structure.

A very efficient recursive algorithm called the Fast Kalman Algorithm is derived to calculate the Kalman gain vector recursively for both the filter and the predictor. The number of multiplications and divisions needed to update the Kalman filter gain vector is shown to be 5n+2 and for the predictor gain vector 6n+2. To update the state vector of the Kalman filter and predictor requires only 2n multiplications so that it is obvious that still the Kalman gain calculation is the main computational burden in implementing the Kalman filter and predictor.

To implement the Kalman filter and predictor using fixed point arithmetic the state variables must be scaled to limit the probability

of overflow. A so-called l_2 scaling rule is used. For the Kalman predictor/filter to be properly scaled the variance of each state element must be less than one in magnitude at all time. Scaling each state element properly requires a substantial number of multiplications. If the variance of each state element is on the same number of magnitude then the scaling can be accomplished by only one multiplication. When using this simplified scaling procedure the scale factor s must be chosen large enough so the scaled input z(t)/s is less than one in magnitude at all time. This can result in overscaled state when there is a large measurement noise. The Fast Kalman Algorithm is scaled by nature and is very well suited for fixed point implementation. The numerical results confirm this.

The effect of multiplication roundoff is also considered and modeled by an additive white noise vector in the state update equation.

Numerical results are presented for implementation of the Kalman filter in both floating point and fixed point arithmetic. The fixed point results compare very well with the floating point results as the figures in section 5.2 confirm. The fixed point results look even better considering that the simplified scaling procedure was used and chopping instead of rounding was used. It is also interesting to note that increased measurement noise seems to have a positive effect on the performance of the fixed point Kalman filter.

The major finding of this thesis is the surprisingly good performance of the fixed point implementation of the Kalman filter. The Fast Kalman Algorithm is certainly efficient but is still the major computational burden in implementing the Kalman filter.

Some of the things we have not covered in this thesis but should form the basis of future work are:

--The effects of roundoff in the Fast Kalman Gain Algorithm.

--Compute the Fast Kalman Gain Algorithm with other Kalman gain algorithms implemented in fixed point (Lattice, Ricatti).

--The wordlength issue in the Fast Kalman Gain Algorithm and in the Kalman filter/predictor.

--Input and coefficient truncation.

--Limit cycle behavior.

APPENDIX A

THE SOFTWARE

In this appendix we describe the software programs used in this study and describe how to use them. There are 5 main programs.

MAIN1. This program generates noisy data $z(t)=y(t)+n(t)$. The Innovations representation #1 is used to generate $y(t)$. The equations used are

$$\underline{x}(t+1/t) = A\underline{x}(t/t-1) + \underline{k}(t)u(t)$$

$$\underline{x}(t+1/t+1) = \underline{x}(t+1/t) + \underline{k}(t)u(t)$$

$$y(t) = \underline{c}'\underline{x}(t/t-1)$$

$$\underline{k}(t) = A\underline{k}(t)$$

The user gives the names of the files where the following information is stored:

i) the denominator coefficients a(i) i=0,n

ii) the first n+1 values of the covariance sequence of the ARMA structure: r(i) i=0,n.

The user also gives the length, nz, of the data sequence to be generated and the noise variance, Varn, of the measurement noise n(t).

At the end there is an option of saving all the data on a single file or all or part of the data on separate files. The data to be saved includes $z(t)$, $y(t)$, $u(t)$, $v(t)$, $\underline{k}(t)$, $\underline{k}(t)$, $\underline{l}(t)$, $\underline{x}(t/t-1)$, and $\underline{x}(t/t)$.

MAIN2. This program performs floating point Kalman filtering of the data sequence z(t). The equations used are

$$\underline{\hat{x}}(t/t) = \underline{\hat{x}}(t/t-1) + \underline{\hat{k}}(t)(z(t)-\hat{y}(t))$$

$$\underline{\hat{x}}(t/t-1) = A\underline{\hat{x}}(t-1/t-1)$$

$$\hat{y}(t) = \underline{c}'\underline{\hat{x}}(t/t-1)$$

$$\underline{\hat{k}}(t) = A\underline{\hat{k}}(t)$$

The user gives the names of the files where the following information is stored:

i) the denominator coefficients a(i) i=0,n ,

ii) the first n+1 values of the covariance sequence of the ARMA structure: r(i) i=0,n ,

iii) the noisy data sequence z(t).

The user also specifies how many values of the noisy sequence are desired and gives the assumed noise variance, Varn, of the measurement noise n(t).

At the end there is an option of saving all the data on a single file or all or part of the data on separate files. The data saved can include z(t), $\hat{y}(t)$, $\hat{u}(t)$, $\hat{v}(t)$, $\underline{\hat{k}}(t)$, $\underline{k}(t)$, $\underline{\hat{l}}(t)$, $\underline{\hat{x}}(t/t-1)$, and $\underline{\hat{x}}(t/t)$.

MAIN3. This program communicates with the MAIN86 which implements the Kalman filter using fixed point arithmetic on Intel's SDK86 single board computer. Before running the MAIN3 program on the Development system the hex version of MAIN86, MAIN86.HEX should be downloaded first and started.

The user gives the names of the files where the following information is stored:

i) the denominator coefficients a(i) i=0,n ,

ii) the first n+1 values of the covariance sequence of the ARMA structure: r(i) i=0,n ,

iii) the noisy data z(t).

The user also specifies how many values of the z(t) sequence are desired and gives the assumed noise variance, Varn, of the measurement noise n(t). The scale factors Scale_A and Scale_Z for the denominator coefficients and the noisy data sequence must also be given. The scale factor Scale_A should be chosen such that all the scaled denominator coefficients will be less than one in magnitude. The scale factor Scale_Z should be chosen such that the scaled data sequence (Scale_Z)z(t) and all state elements are less than one in magnitude at all times.

At the end the user has the option of saving all the data on a single file or all or part of the data on separate files. The data saved can include z(t), $\hat{y}(t)$, $\hat{u}(t)$, $\hat{v}(t)$, $\underline{\hat{k}}(t)$, $\underline{\hat{k}}(t)$, $\underline{\hat{1}}(t)$, $\hat{x}(t/t-1)$, and $\underline{\hat{x}}(t/t)$.

MAIN4. This program allows the user to create a data file and generate the covariance sequence r(i) i=0,nr corresponding to an ARMA process when given the file names where the denominator and nominator coefficients, a(i) i=0,n and b(i) i=0,n, are stored. The covariance is calculated using a method of Mullis and Roberts.

MAIN86. This is the fixed point implementation of the Kalman filter. The program is written as assembly language and runs on the Intel single board computer SDK86. See also the description of the MAIN3 program which communicates with the MAIN86. The hex version of MAIN86 must be downloaded to the SDK86 from the Development system and

executed before running the MAIN3 program. All the data generated by this MAIN86 program is sent to the development system so that it can be saved on a file(s) for later analysis. To save on the number of arithmetic operations we use truncation instead of rounding.

APPENDIX B

SYSTEM CONFIGURATION

The system configuration under which the software programs described in Appendix A run is the following:

1. Intellect Series II Model 230 Microcomputer Development System (MDS-230) which includes:
   - i) 64k bytes RAM
   - ii) Integral CRT and detachable keyboard
   - iii) Two flexible disk drives (1 million bytes)
2. SDK-86 single board microcomputer system with necessary software and hardware to interface to the development system (SDK-C86).
3. A line printer.

Figure 1 shows the basic configuration. For more information see for example the following reference manuals:

A Guide to Microcomputer Development Systems.

Microcomputer Development Packages.

The 8086 Family User's Manual (Appendix B, Development Tools).

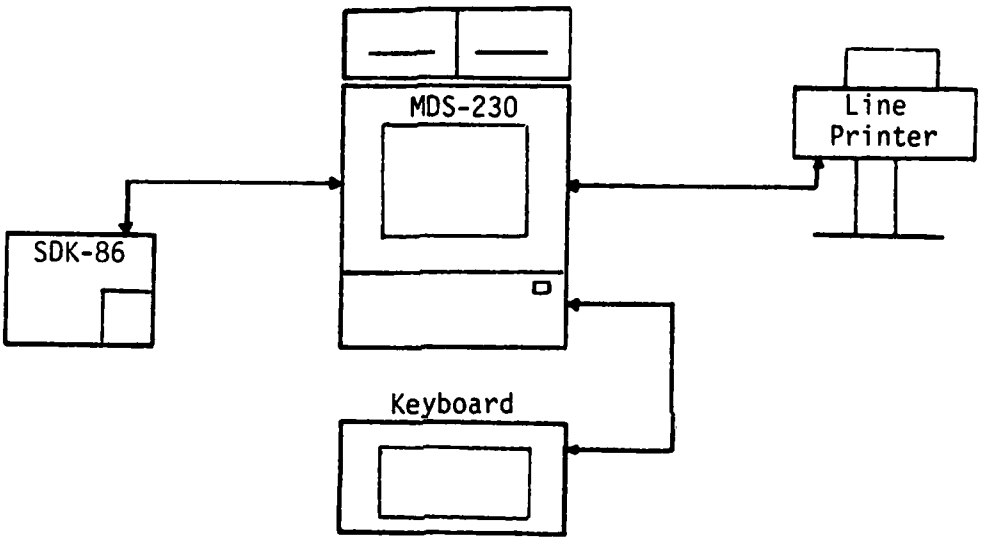


Fig. 1. Basic system configuration.

OFFICE OF NAVAL RESEARCH
STATISTICS AND PROBABILITY PROGRAM

BASIC DISTRIBUTION LIST
FOR
UNCLASSIFIED TECHNICAL REPORTS

FEBRUARY 1982

	Copies
Statistics and Probability Program (Code 411(SP)) Office of Naval Research Arlington, VA 22217	3
Defense Technical Information Center Cameron Station Alexandria, VA 22314	12
Commanding Officer Office of Naval Research Eastern/Central Regional Office Attn: Director for Science Barnes Building 495 Summer Street Boston, MA 02210	1
Commanding Officer Office of Naval Research Western Regional Office Attn: Dr. Richard Lau 1030 East Green Street Pasadena, CA 91101	1
U. S. ONR Liaison Office - Far East Attn: Scientific Director APO San Francisco 96503	1
Applied Mathematics Laboratory David Taylor Naval Ship Research and Development Center Attn: Mr. G. H. Gleissner Bethesda, Maryland 20084	1
Commandant of the Marine Coprs (Code AX) Attn: Dr. A. L. Slafkosky Scientific Advisor Washington, DC 20380	1
Navy Library National Space Technology Laboratory Attn: Navy Librarian Bay St. Louis, MS 39522	1
U. S. Army Research Office P.O. Box 12211 Attn: Dr. J. Chandra Research Triangle Park, NC 27706	1
Director National Security Agency Attn: R51, Dr. Maar Fort Meade, MD 20755	1
ATAA-SL, Library U.S. Army TRADOC Systems Analysis Activity Department of the Army White Sands Missile Range, NM 88002	1
ARI Field Unit-USAREUR Attn: Library c/o ODCSPER HQ USAEREUR & 7th Army APO New York 09403	1
Library, Code 1424 Naval Postgraduate School Monterey, CA 93940	1
Technical Information Division Naval Research Laboratory Washington, DC 20375	1
OASD (I&L), Pentagon Attn: Mr. Charles S. Smith Washington, DC 20301	1

Copies

Director
AMSAA
Attn: DRXSY-MP, H. Cohen
Aberdeen Proving Ground, MD 21005 1

Dr. Gerhard Heiche
Naval Air Systems Command (NAIR 03)
Jefferson Plaza No. 1
Arlington, VA 20360 1

Dr. Barbara Bailar
Associate Director, Statistical Standards
Bureau of Census
Washington, DC 20233 1

Leon Slavin
Naval Sea Systems Command (NSEA 05H)
Crystal Mall #4, Rm. 129
Washington, DC 20036 1

B. E. Clark
RR #2, Box 647-B
Graham, NC 27253 1

Naval Underwater Systems Center
Attn: Dr. Derrill J. Bordelon
Code 601
Newport, Rhode Island 02840 1

Naval Coastal Systems Center
Code 741
Attn: Mr. C. M. Bennett
Panama City, FL 32401 1

Naval Electronic Systems Command (NELEX 612)
Attn: John Schuster
National Center No. 1
Arlington, VA 20360 1

Defense Logistics Studies Information Exchange
Army Logistics Management Center
Attn: Mr. J. Dowling
Fort Lee, VA 23801 1

Copies

Reliability Analysis Center (RAC)
RADC/RBRAC
Attn: I. L. Krulac
Data Coordinator/
Government Programs
Griffiss AFB, New York 13441 1

Technical Library
Naval Ordnance Station
Indian Head, MD 20640 1

Library
Naval Ocean Systems Center
San Diego, CA 92152 1

Technical Library
Bureau of Naval Personnel
Department of the Navy
Washington, DC 20370 1

Mr. Dan Leonard
Code 8105
Naval Ocean Systems Center
San Diego, CA 92152 1

Dr. Alan F. Petty
Code 7930
Naval Research Laboratory
Washington, DC 20375 1

Dr. M. J. Fischer
Defense Communications Agency
Defense Communications Engineering Center
1860 Wiehle Avenue
Reston, VA 22090 1

Mr. Jim Gates
Code 9211
Fleet Material Support Office
U. S. Navy Supply Center
Mechanicsburg, PA 17055 1

Mr. Ted Tupper
Code M-311C
Military Sealift Command
Department of the Navy
Washington, DC 20390 1

Copies

Mr. F. R. Del Priori
Code 224
Operational Test and Evaluation
Force (OPTEVFOR)
Norfolk, VA 23511 1

and fixed point arithmetic. Noise variance $\sigma_n^2 = 2.0$.

END

FILMED

5-83

DTIC