RL-TR-95-294
Final Technical Report
February 1996

MULTICHANNEL SIGNAL PROCESSING EXTENSIONS

Kaman Sciences Corporation

Robert Vienneau

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be releasable to the general public, including foreign nations.

RL-TR-95-294 has been reviewed and is approved for publication.

APPROVED: James H. Michels

JAMES H. MICHELS, Ph.D.
Project Engineer

FOR THE COMMANDER: Gary D. Barmore

GARY D. BARMORE, Major, USAF
Deputy Director of Surveillance & Photonics

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify Rome Laboratory/ (OCSM), Rome NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1996	Final Apr 94 - Aug 95

4. TITLE AND SUBTITLE

MULTICHANNEL SIGNAL PROCESSING EXTENSIONS

5. FUNDING NUMBERS

C - F30602-94-C-0101
PE - 61102F
PR - 2304
TA - E8
WU - PC

6. AUTHOR(S)

Robert Vienneau

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Kaman Sciences Corporation
258 Genesee Street
Utica NY 13502-4627

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratory/OCSM
26 Electronic Pky
Rome NY 13441-4514

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-95-294

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: James H. Michels, Ph.D./OCSM/(315) 330-4432

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This effort extends the signal processing capabilities of the Multichannel Signal Processing Simulation System (MSPSS), including:

- the analysis of a signal detection algorithm for a constant magnitude signal of unknown amplitude in white or colored noise;
- the addition of a capability to generate Weibull-distributed noise or clutter;
- the implementation of the representative model that provides user-control over statistical properties of simulated space-time data, as calculated from certain parameters characterizing the phased-array radar platform, the clutter environment, signals, and jammers; and
- additional diagnostic capabilities.

The MSPSS was developed to assess the performance of multichannel signal processing algorithms for detection and estimation analyses. Specific emphasis has been given to the implementation of multichannel parametric model-based methods.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Multichannel, Detection, Estimation, Non-Gaussian, Weibull	108
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

1. OVERVIEW

This document is the final report for Multichannel Signal Processing Extensions, an effort conducted by Kaman Sciences Corporation (KSC) under Rome Laboratory's Broad Agency Announcement (BAA) number 93-07. This effort was performed for

Dr. James H. Michels
Rome Laboratory
RL/OCTM
Griffiss AFB, NY 13441
(315) 330-4432

1.1 Objective

The objective of this effort was to extend the signal processing capabilities of the Multichannel Signal Processing Simulation System (MSPSS), including:

- The analysis of a signal detection algorithm for a signal with constant but unknown amplitude in (white or colored) noise
- The addition of a capability to generate Weibull-distributed noise or clutter
- The implementation of the Representative Model which provides user-control over statistical properties of simulated space-time data, as calculated from certain parameters characterizing the phased-array radar platform, the clutter environment, signals, and jammers
- Additional diagnostic capabilities.

1.2 Background

Dr. James Michels' ongoing research program investigates an original approach in multichannel signal processing (Michels 89, 90a, 90b, 91, 92a, 92b), including the sponorship of related work (e.g. Rangaswamy 92 and Roman 93). This research program has developed:

- A synthesis procedure for simulating multichannel Autoregressive (AR) processes in which intertemporal and interchannel correlations are controlled parametrically.
- Extensions to this synthesis procedure to handle non-Gaussian Spherically Invariant Random Processes (SIRPs) for K distributions.
- Diagnostics for examining statistical properties of synthesized processes.
- A multichannel signal detection algorithm based on a generalized loglikelihood ratio using an innovations approach, including ratios for Gaussian processes and K-distributed SIRPs.
- A Kalman filter structure for a state-space version of the signal detection algorithm.
- A sensor fusion application of the signal detection algorithm.
- A Monte Carlo approach for exploring the performance of the signal detection algorithms.
- An extension of the Monte Carlo approach for calculating thresholds for given false alarm rates based on approximating the tail of a distribution by a Pareto distribution.

1.2.1 The Multichannel Signal Processing Simulation System

Kaman Sciences Corporation has designed and implemented the Multichannel Signal Processing Simulation System (MSPSS) to support Dr. Michels' research program. The system architecture was defined early, and capabilities were added over a series of projects (Kaman 91a, 92a, 92b and Vienneau 93 and 94).

The MSPSS is comprised of two major subsystems, a menu-based subsystem and the User Front-end Interface (UFI) based subsystem. The menu-based subsystem interacts with the user by a series of menus and prompts that can be displayed on a line-oriented "glass terminal." It is implemented as a collection of Fortran programs providing the desired signal processing capabilities. Multichannel data is passed between these programs by user-defined files. The menus themselves are Unix C-shells where the bottom-level menu automatically compiles, links, and executes the desired program. Figure 1-1 shows an example of the invocation of a Fortran program from the menu-based subsystem.

The structure of the menu-based subsystem provides the user with a great deal of analytical flexibility. The individual Fortran programs provide certain analytical capabilities, but minimal constraints are imposed on the order in which the user can invoke these functions. A number of diagnostic capabilities are provided, including parameter estimation for certain models, correlation function estimation, and graphical display of data. Thus the user is provided with a system that supports an exploratory style of analysis.

The UFI-based subsystem is designed for more repetitive analyses characteristic of the determination of detection probabilities for given false alarm probabilities and given algorithms. In such analyses, the user needs to invoke certain signal processing synthesis and analysis functions in a predetermined order with certain parameters. The UFI-based subsystem supports this need by allowing the user to construct an "experiment" specified by various algorithms and parameters. Once an experiment has been completely described, a computer program for performing an experiment is automatically generated, compiled, and executed. Since these are simulation experiments, the generated program can run for quite some time. Typically, a single experiment will be used to analyze the performance of the signal detection algorithm in terms of the false alarm probability and the probability of detection. A detailed example of the use of this subsystem is provided in the *Software User's Manual for the Multichannel Signal Processing Simulation System* (Kaman 92a).

1.2.2 Detection of Unknown Amplitude Signal

The signal detection algorithm, illustrated in Figure 1-2, is quite general. The radar returns $x(n)$ are entered into two parallel filters. The structure of the filters is based on models of the returns with and without a signal. If no deterministic signal is present, the output of the null hypothesis filter, $v_0(n)$, will be a white noise error signal. If a deterministic signal is present, the output of the alternative hypothesis filter, $v_1(n)$, will be a white noise error signal. The statistic Λ is used to determine which filter output provides the minimum white noise error signal. If it is above some threshold value, a signal is determined to be detected. Filters with different structures correspond to models of different clutter environments and different signals. A variety of parameter estimation algorithms may exist for a given model. The appropriate loglikelihood statistic differs between Gaussian and non-Gaussian noise.

```
                                          The user invokes the menu-based
isaiah.1-52% mchan                        multichannel system from the Unix shell.

                Multi-Channel Detection Algorithm

                        MAIN MENU

     Single Channel                  Multichannel
M1 -- Process Synthesis          M11 -- Process Synthesis
M2 -- Filtering Methods          M12 -- Filtering Methods
M3 -- Diagnostics                M13 -- Diagnostics

L1  -- List data files           L2  -- show the change log
Enter a command or Q to quit: m11          The user chooses a submenu.
              MULTI-CHANNEL (M/C) PROCESS SYNTHESIS MENU

M0  -- Gaussian noise            M2  -- Method 2 (MC2) synthesis
M1  -- Method 1 (MC1) synthesis  M3  -- Representative model
M4  -- Method 2 unconstrained quadrature process synthesis
M5  -- State space synthesis
M6  -- Apply Levinson-Wiggins-Robinson algorithm
M7  -- Display Multi-Channel (M/C) signal stats
M8  -- Plot M/C data
M9  -- Perform Nuttall-Strand or Vieira-Morf estimation
M10 -- Perform Yule-Walker estimation
M11 -- Estimate AR parameters for each channel
M12 -- Estimate covariance matrix
M13 -- Estimate state space parameters
M14 -- Estimate state space parameters from exact covariance matrix
M15 -- Estimate amplitude of constant signal
M16 -- Perform M/C correlation (temporal)
M17 -- Perform M/C correlation (ensemble)
M18 -- Split a M/C input         M23 -- Join inputs
M19 -- Add M/C signals           M24 -- Subtract M/C signals
M20 -- Convert AR parameters to state space
M21 -- Display complex data      M25 -- Display coefficient file
M22 -- Catenate M/C signals      M26 -- Sum channels

L1  -- List data files           L2  -- show the change log

Enter a command or Q to return to the MAIN menu: m3
                                          The user invokes and interacts
                                          with a Fortran program.
```

Figure 1-1: Invoking a Fortran Program from the Menu-Based System

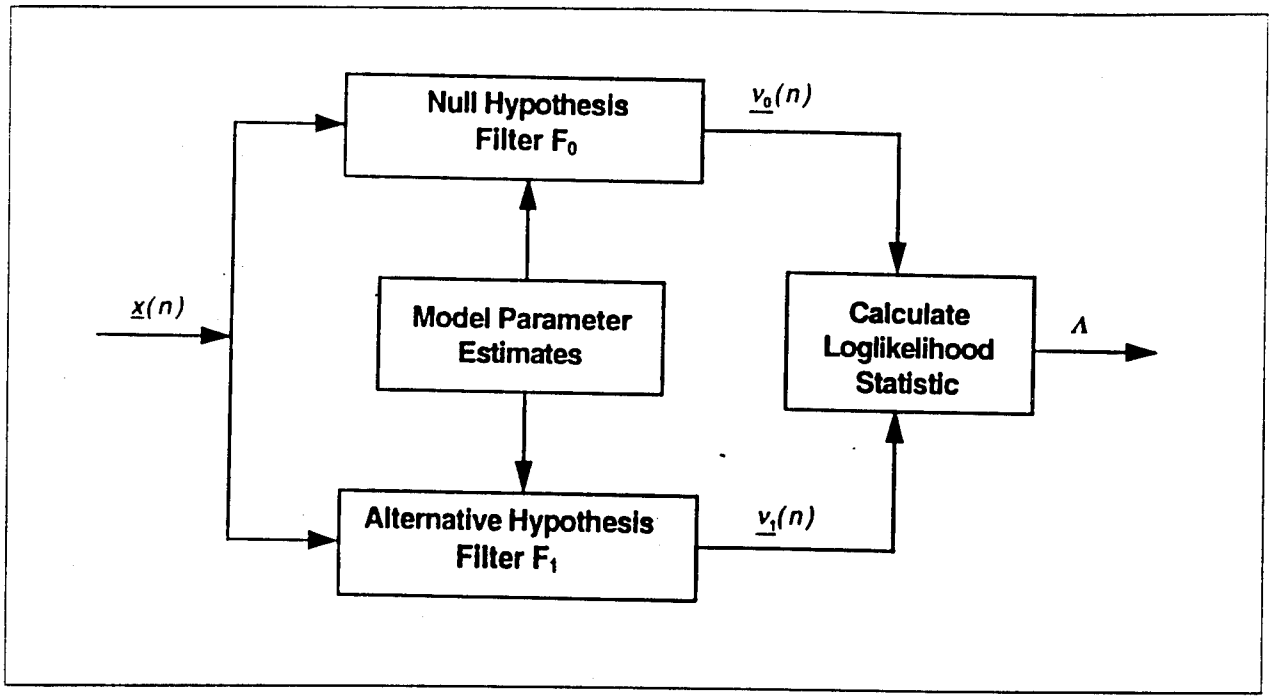

Figure 1-2: The Signal Detection Algorithm

Not all instantiations of this signal detection algorithm have been implemented in the MSPSS. Recently, capabilities were added to analyze the detection of a constant magnitude signal, assuming its amplitude is known (Vienneau 94). Dr. Michels has extended the detection algorithm to a model of a signal with constant but unknown amplitude. Capabilities for analyzing this case were added under this effort.

1.2.3 Weibull Distribution

Clutter and noise have been modeled as stochastic processes. For example, capabilities exist in the MSPSS to synthesize the clutter as an AR process. Driving noise for AR processes and white noise were first implemented as Gaussian noise. Recent research at Syracuse University and Rome Laboratory has investigated non-Gaussian processes known as Spherically Invariant Random Processes (SIRPs). SIRPs can be used to model many probability distributions (Rangaswamy 91, 92, 93a, and 93b). A K-distribution was the only non-Gaussian SIRP synthesized by the MSPSS prior to this effort. Now a capability exists for synthesizing Weibull SIRPs in the MSPSS.

1.2.4 Representative Model

The Representative Model is closely related to the synthesis procedure existing in the MSPSS at the start of this effort. The Representative Model provides control over statistical properties of simulated space-time data, as calculated from certain parameters characterizing the phased-array radar platform, the clutter environment, signals, and jammers. The Representative Model differs from the Physical Model in that it provides greater control over statistical properties of the simulated radar returns. The radar is described at a higher level of abstraction, while the Physical Model simulates specific radars, signals, and jammers.

The Representative Model parameters determine "shaping functions" used to calculate block covariance matrices for the clutter, signal, and interference. The block covariance matrices, in turn, are used to find either a Cholesky decomposition of the block covariance

matrix, or Autoregressive (AR) coefficients used in simulating the radar returns. The simulated radar returns are appropriate for testing the performance of a signal detection algorithm, examining the performance of estimation algorithms, passing through a Fast Fourier Transform (FFT), etc.

An interesting question is the performance of the signal detection algorithm analyzed by the MSPSS when applied to data characterized by the Representative Model. The Representative Model was implemented in the MSPSS under this effort. This implementation permits the use of diagnostics to characterize data synthesized by the Representative Model, but does not support convenient examination of the performance of the signal detection algorithm applied to Representative Model data.

1.2.5 Diagnostics

The MSPSS implemented certain signal processing diagnostic capabilities before the start of this project. More were added. A previous effort added a capability to estimate the parameters of a state space model (Vienneau 94). This capability is difficult to test since estimated matrices may differ from theoretical values by a basis transformation of the state space and since statistical variation may combine with numerical effects to yield inaccurate estimates. A capability was added under this project to calculate a theoretically exact block covariance matrix for an AR process, which in turn is used to estimate the state space parameters.

Certain capabilities existed at the start of this effort for estimating correlation functions. Insight into the underlying processes can be obtained by examining the spectrum of correlation functions. Accordingly, this effort included the addition of a capability to calculate a two dimensional Fast Fourier Transform (2D-FFT).

The radar system simulated by the Representative Model can contain up to 14 channels. Therefore, the number of channels in processes synthesized and analyzed by the MSPSS was increased from the previously existing bound of four channels, namely to 16 channels.

1.3 Overview of Report

This section consists of an introduction and an overview of the remainder of the report.

Section 2 provides a brief description of the signal detection problem and an overview of the approach that can be analyzed by the MSPSS.

Section 3 defines new capabilities added to the MSPSS during this effort. Subsections present the analysis of the detection of a signal with unknown amplitude, the synthesis of a Weibull SIRP, the representative model, the calculation of state space parameters, and a two dimensional Fast Fourier Transform (2D-FFT).

Section 4 discusses implementation details for the new capabilities. Examples of how the new capabilities are invoked are provided.

Section 5 provides references, while appendices describe notation, acronyms, and certain mathematical details.

2. THE MULTICHANNEL SIGNAL PROCESSING SIMULATION SYSTEM

The Multichannel Signal Processing Simulation System (MSPSS) was developed by Kaman Sciences to support RL/OCTM research. This section briefly describes the signal processing problems that can be analyzed with the MSPSS and the use of the MSPSS.

2.1 Signal Detection and Statistical Hypothesis Testing

The primary purpose of the MSPSS is to analyze the performance of a signal detection algorithm. Signal detection can be thought of as a problem in statistical hypothesis testing. Let x denote a multichannel vector stochastic process representing the radar returns. Radar returns consist of a time series of complex vectors, where the dimension of the vectors is the number of radar elements (channels). Let s denote a signal, c denote the clutter, and n denote white noise. Consider deciding between the null hypothesis

- H_0: $x = c + n$

and the alternative hypothesis

- H_1: $x = s + c + n$.

Deciding that a signal is present is to accept the alternative hypothesis.

The Neyman-Pearson theory of statistical hypothesis testing provides control over probabilities of making an erroneous decision. The significance level, α, of a statistical test is the probability that a decision rule will accept the alternative hypothesis when the null hypothesis is true:

$$\alpha = Pr\,(Accept\ H_1 \mid H_0\ true\,). \tag{2.1-1}$$

Since to accept the alternative is to decide a signal is present, the significance level is the probability of false alarm in the signal detection problem.

Another type of erroneous decision is possible. A decision rule can result in a decision that no signal is present when, in fact, a signal exists. The probability of this mistake, known as a Type II error, is usually denoted by β. The power of a test denotes the probability of correctly deciding in favor of the alternative hypothesis. The power is related to the probability of a Type II error, as shown by Equation 2.1-2:

$$1 - \beta = Pr\,(Accept\ H_1 \mid H_1\ true\,). \tag{2.1-2}$$

In signal detection, the power of a test is known as the probability of detection.

2.2 A Model-Based Approach

The MSPSS supports the analysis of a model-based approach to signal detection. In other words, both the synthesis and the analysis of radar returns are based on certain parameterized models. (Non-model based approaches are also known as non-parametric approaches.) Synthesis models currently implemented in the MSPSS include:

- Multichannel Gaussian, K-distributed, and Weibull noise
- Spherically Invariant Random Processes (SIRPs) in which each realization is Gaussian but the distribution of a time sample across realizations is K-distributed or Weibull.
- Autoregressive (AR) processes with driving noise from SIRP or white noise

- Autoregressive Moving Average (ARMA) models implemented as the sum of AR models and white noise
- A state space model
- Multipath processes

These models are implemented in an innovations representation. The output process is formed by modifying a driving noise term. The filters in the signal detection algorithm in Figure 1-2 are designed to produce estimates of the innovations process of the model corresponding to the appropriate hypothesis. A different filter structure corresponds to each different model structure. The corresponding loglikelihood statistic is designed to determine which filter output more closely resembles the modeled process.

For example, Figure 2-1 shows the system structure for synthesizing an AR process. The input process $v(n)$ is white noise uncorrelated both in time and across channels. The operator T adds correlation across channels to produce $\varepsilon(n)$. The remainder of the structure adds correlation in time and modifies the correlation across channels. The appropriate filter to estimate the innovations for this AR model has a tapped delay line structure.

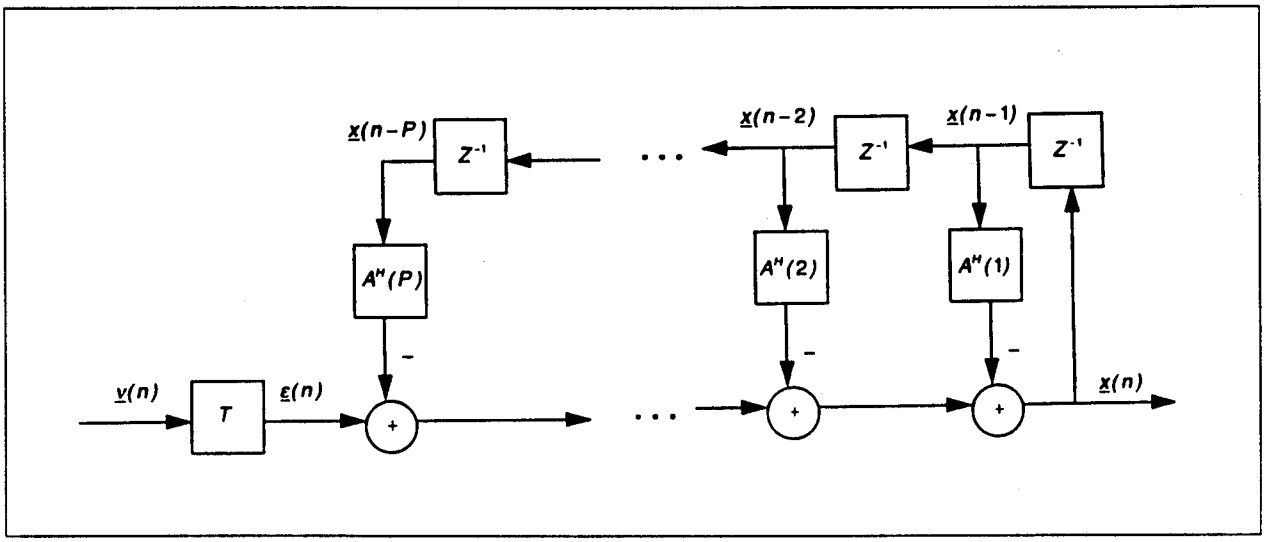

Figure 2-1: Synthesis of an AR Process

A state space model (Figure 2-2) provides another powerful example of an innovations model. In this model, an internal process, $\alpha(n)$, is used to inject intertemporal correlation. In this case, a Kalman filter provides the appropriate structure for estimating the innovations process.

The MSPSS provides instantiations of the signal detection algorithm shown in Figure 1-2. An instantiation is constructed by combining a synthesis model, parameter estimation algorithms, filters for estimating innovations, and an appropriate test statistic. The MSPSS allows the user to analyze the resulting algorithm. Thresholds can be calculated for given false alarm probabilities, and then a corresponding probability of detection can be determined.

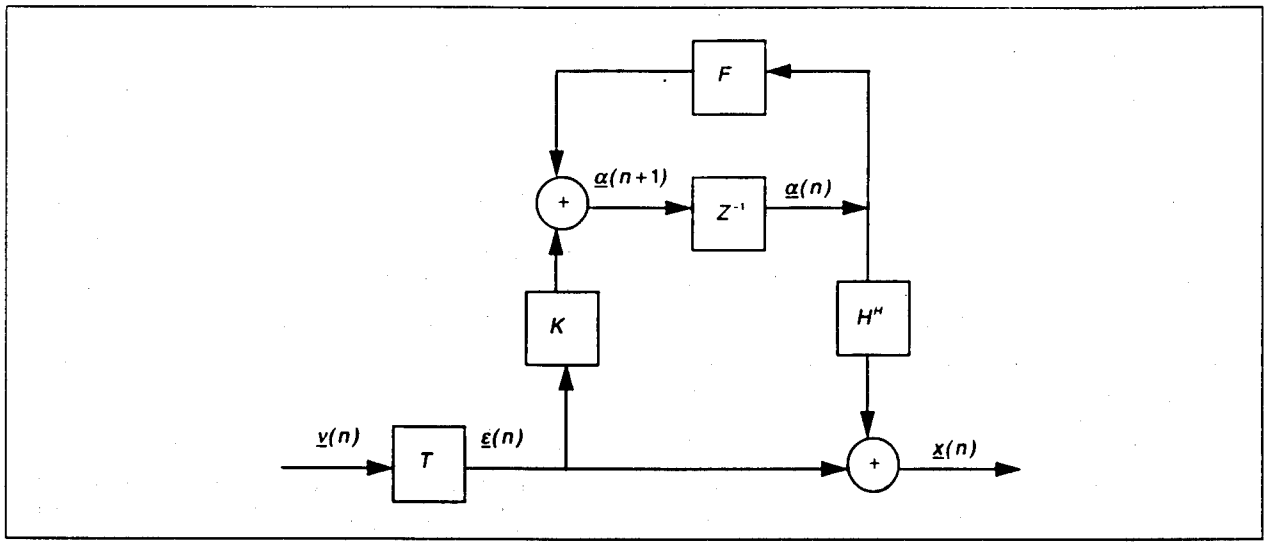

Figure 2-2: Innovations Representation of a State Space Model

2.3 Diagnostics

Synthesis procedures in the MSPSS provide user-control over characteristics of the synthesized processes, such as temporal and cross-channel correlation. Since these are stochastic processes, the synthesized radar returns exhibit a range of random variability. Diagnostics provide the user a number of ways to examine radar processes.

The MSPSS provides a graphing capability integrated from Khoros (Rasure 92). Graphical displays can be rotated, annotated, otherwise manipulated, and printed. Statistical routines are provided for means, variances, correlation functions, parameter estimation, distribution identification and testing, periodograms, and Fast Fourier Transforms of spectra. Additional capabilities include various filters and data manipulation capabilities such as adding processes, combining channels, and splitting channels.

2.4 Signal Detection

The User Front-end Interface (UFI) based subsystem provides powerful capabilities for analyzing the signal detection algorithm. Analysis sequences allow the user to specify the model parameters. A Monte Carlo approach is used to estimate filter parameters, thresholds for given false alarm probabilities, probabilities of detection, and variability in thresholds and probabilities of detection. The extreme value method (Chakravarthi 92 and Vienneau 94) gives efficient estimates of thresholds for extremely small false alarm probabilities.

The MSPSS allows the user to determine how the probability of detection varies with false alarm probabilities and characteristics of the radar returns such as signal to noise ratios. Different parameter estimation algorithms and different filtering structures can be explored. In short, the MSPSS allows the user to explore the robustness of the signal detection algorithm under a wide range of instantiations and circumstances. Published results using this system are positive so far (Michels 94 and 95).

Section 4 describes the use of the MSPSS for the capabilities added under this effort. General user information is provided in (Vienneau 93) and (Vienneau 94). (Kaman 92a) describes how to use the User Front-end Interface (UFI) based subsystem. Procedures for adding additional capabilities are discussed in (Vienneau 93).

3. NEW CAPABILITIES

Kaman Sciences has implemented and enhanced the Multichannel Signal Processing Simulation System (MSPSS) over a series of contracts. The MSPSS was extended under this effort to include the following capabilities:

- The analysis of the detection of a signal with constant but unknown amplitude in clutter
- The synthesis of a Weibull-distributed Spherically Invariant Random Process (SIRP)
- The synthesis of the Representative Model, originally designed for the Rome Laboratory Space-Time Adaptive Processing Algorithm Development Tool (RLSTAP/ADT)
- The estimation of state space model parameters from a theoretically exact block covariance matrix for an AR process
- A two-dimensional Fast Fourier Transform (2D-FFT)
- The extension of all capabilities to 16 channels.

This section defines the capabilities implemented under this effort.

3.1 A Signal with Unknown Amplitude

A new model-based signal detection analysis capability was added. The signal detection analysis capabilities allow the user to determine the false alarm probability and probability of detection for specific algorithms under controlled conditions. These signal detection algorithms are based on models of the signal, clutter, and noise. In this case, the algorithm decides between the null hypothesis:

$$H_0: \; x(n) = y(n), \; n = 1,2,\ldots,N, \tag{3.1-1}$$

and the alternative hypothesis:

$$H_1: \; x(n) = s(n) + y(n), \; n = 1,2,\ldots,N. \tag{3.1-2}$$

The radar return, $x(n)$, is a J-element complex vector. $y(n)$ represents clutter, while $s(n)$ is the signal.

3.1.1 Clutter Synthesis

The clutter process, $y(n)$, is modeled as an Autoregressive (AR) process with white driving noise:

$$y(n) = -\sum_{k=1}^{p} A^H(k)\, y(n-k) + \varepsilon(n), \tag{3.1-3}$$

where $y(n)$ and $\varepsilon(n)$ are J-element complex column vectors and $A^H(1)$, $A^H(2)$, ..., $A^H(p)$ are $J \times J$ AR matrix coefficients for an AR process of order p.

The driving noise process, $\varepsilon(n)$, is uncorrelated in time, but possibly correlated across channels. The correlation across channels is expressed by the $J \times J$ covariance matrix Σ_ε:

$$\Sigma_\varepsilon = E\left[\varepsilon(n)\varepsilon^H(n)\right]. \tag{3.1-4}$$

The driving noise process is synthesized based on one of three decompositions of the covariance matrix. The Cholesky decomposition is specified by Equation 3.1-5:

$$\Sigma_\varepsilon = C_\varepsilon C_\varepsilon^H, \tag{3.1-5}$$

where C_ε is a lower triangular complex matrix. The LDU decomposition is specified by Equation 3.1-6:

$$\Sigma_\varepsilon = L_\varepsilon D_\varepsilon L_\varepsilon^H, \tag{3.1-6}$$

where L_ε is a lower triangular complex matrix with unity along the principal diagonal and D_ε is a diagonal matrix. For a correlation matrix the Singular Value Decomposition (SVD) reduces to the unitary similarity transformation and is given by Equation 3.1-7:

$$\Sigma_\varepsilon = Q_\varepsilon \Lambda_\varepsilon Q_\varepsilon^H, \tag{3.1-7}$$

where Λ_ε is a diagonal matrix whose elements are eigenvalues of Σ_ε. The columns of Q_ε are the corresponding right hand eigenvectors:

$$\Sigma_\varepsilon (Q_\varepsilon)_{.j} = (\Lambda_\varepsilon)_{j,j} (Q_\varepsilon)_{.j}. \tag{3.1-8}$$

The rows of Q_ε^H are the left hand eigenvectors:

$$(Q_\varepsilon^H)_{i.} \Sigma_\varepsilon = (\Lambda_\varepsilon)_{i,i} (Q_\varepsilon^H)_{i.}. \tag{3.1-9}$$

Since Σ_ε is Hermitian, the Hermitian transpose of Q_ε is also the inverse of Q_ε.

The driving noise is generated based on either C_ε, L_ε and D_ε, or Q_ε and Λ_ε. For each time sample in the driving noise term, the J-element complex column vector $v_\varepsilon(n)$ is generated. $v_\varepsilon(n)$ is from a K-distributed Spherically Invariant Random Process (SIRP), a Weibull SIRP, or a Gaussian distribution. $v_\varepsilon(n)$ is uncorrelated across channels and across time. If the user chose a Cholesky decomposition, the jth channel of $v_\varepsilon(n)$ has a variance of unity. If the user specifed a LDU decomposition or SVD, the jth channel has a variance of $(D_\varepsilon)_{j,j}$ or $(\Lambda_\varepsilon)_{j,j}$, respectively. The driving noise term is then given by Equation 3.1-10:

$$\varepsilon(n) = T_\varepsilon v_\varepsilon(n), \tag{3.1-10}$$

where T_ε is either C_ε, L_ε, or Q_ε.

The driving noise covariance matrix Σ_ε and the AR coefficients $A^H(1)$, $A^H(2)$, ..., $A^H(p)$ are determined based on a "shaping function" approach (Michels 94). The correlation matrix for the AR process for the kth lag is defined by Equation 3.1-11:

$$R_y(k) = E[y(n) y^H(n-k)], \tag{3.1-11}$$

where $R_y(k)$ is JxJ. By definition, Equation 3.1-12 follows:

$$R_y(-k) = R_y^H(k). \tag{3.1-12}$$

The correlation matrix and the AR parameters are related by the Yule-Walker equations. For example, the Yule-Walker equations for three lags, $p = 3$, are given by Equation 3.1-13:

$$\begin{bmatrix} I & A^H(1) & A^H(2) & A^H(3) \end{bmatrix} \begin{bmatrix} R_y(0) & R_y(1) & R_y(2) & R_y(3) \\ R_y(-1) & R_y(0) & R_y(1) & R_y(2) \\ R_y(-2) & R_y(-1) & R_y(0) & R_y(1) \\ R_y(-3) & R_y(-2) & R_y(-1) & R_y(0) \end{bmatrix} = \begin{bmatrix} \Sigma_\varepsilon & 0 & 0 & 0 \end{bmatrix} \tag{3.1-13}$$

The values of $R_y(k)$ are determined by either a Gaussian or Exponential shaping function, and the Yule-Walker equations are solved to obtain the AR parameters. The synthesis of clutter as an AR process was already implemented when this project began. See (Vienneau 93) for further details, including the specification of the shaping functions.

3.1.2 Synthesis of the Deterministic Signal

The deterministic signal, $s(n)$, is modeled by Equation 3.1-14:

$$s(n) = \alpha s_0(n) = \alpha \begin{bmatrix} s_{0,1}(n) \\ s_{0,2}(n) \\ \vdots \\ s_{0,J}(n) \end{bmatrix}, \tag{3.1-14}$$

where α is a complex constant equal across all channels and

$$s_{0,i}(n) = e^{j2\pi(i-1)\frac{d}{\lambda}\sin(\theta_s)} e^{j2\pi(n-1)f_d T \sin(\theta_s)}. \tag{3.1-15}$$

The complex amplitude is specified as

$$\alpha = A e^{j\phi_0} \tag{3.1-16}$$

In synthesizing the signal, the user specifies the constant real magnitude A, the normalized Doppler $f_d T$, the normalized element spacing $\frac{d}{\lambda}$, and the angle of arrival θ_s of the signal. The phase ϕ_0 is a random real number between 0 and 2π. This number is fixed for each realization, but varies across realizations.

3.1.3 Amplitude Estimation

A program was written to estimate the amplitude α of the signal. The parameters of the steering vector $s_0(n)$ are assumed known. These known parameters consist of

- The number of time samples N
- The number of channels J
- The normalized Doppler $f_d T$
- The normalized element spacing $\frac{d}{\lambda}$
- The angle to the signal θ_s

In addition to the signal amplitude, the AR parameters of the clutter are assumed unknown.

Amplitude estimation proceeds in two stages. In the first stage, one of three methods is used to estimate the covariance matrix of the clutter from data generated from the null hypothesis; that is, with just clutter and noise. In the second stage, the amplitude is estimated from the estimated covariance matrix and data generated from the alternative hypothesis. Data from the alternative hypothesis consists of the sum of signal, clutter, and noise.

3.1.3.1 The Sample Covariance Matrix

Suppose K realizations of the interference process are observed. Let $y^k(1)$, $y^k(2)$, ..., $y^k(N)$ be the interference for the kth realization, where $y^k(n)$ is a J-element complex column vector, J is the number of channels, and N is the number of time samples. The concatenated JN column vector y^k is formed as in Equation 3.1-17:

$$y^k = \begin{bmatrix} y^k(1) \\ y^k(2) \\ \cdot \\ \cdot \\ \cdot \\ y^k(N) \end{bmatrix}. \tag{3.1-17}$$

The covariance matrix is estimated as the sample covariance matrix given in Equation 3.1-18:

$$\hat{\Sigma} = \frac{1}{K} \sum_{k=1}^{K} y^k (y^k)^H . \tag{3.1-18}$$

Equation 3.1-18 can be expressed in block matrix form:

$$\hat{\Sigma} = \begin{bmatrix} \hat{r}_1(0) & \hat{r}_1(-1) & \cdots & \hat{r}_1(1-N) \\ \hat{r}_2(1) & \hat{r}_2(0) & \cdots & \hat{r}_2(2-N) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \hat{r}_N(N-1) & \hat{r}_N(N-2) & \cdots & \hat{r}_N(0) \end{bmatrix}, \tag{3.1-19}$$

where

$$\hat{r}_n(l) = \frac{1}{K} \sum_{k=1}^{K} y^k(n)[y^k(n-l)]^H . \tag{3.1-20}$$

3.1.3.2 Time/Space Averaged Estimator

In a stationary process, it seems reasonable that estimates of the same lagged correlation matrix based on different time samples, e.g. $\hat{r}_{n_1}(l)$ and $\hat{r}_{n_2}(l)$, should be close in some sense. For nonnegative lags, time averaged estimates are found by averaging over time samples:

$$\hat{r}_T^k(l) = \frac{1}{N} \sum_{n=l+1}^{N} y^k(n)[y^k(n-l)]^H \tag{3.1-21}$$

for biased estimates, and

$$\hat{r}_T^k(l) = \frac{1}{N-l} \sum_{n=l+1}^{N} y^k(n)[y^k(n-l)]^H \tag{3.1-22}$$

for unbiased estimates. Equation 3.1-23 can be used to estimate lagged correlation matrices for negative lags:

$$\hat{r}_T^k(-l) = [\hat{r}_T^k(l)]^H . \tag{3.1-23}$$

The time averaged estimates of the lagged correlation matrices can now be averaged over many realizations of the clutter. Since in practice these realizations will correspond to different range cells, this step is known as space averaging. The time/space averaged estimators of the correlation matrices are given by Equation 3.1-24:

$$\hat{r}_{TS}(l) = \frac{1}{K} \sum_{k=1}^{K} \hat{r}_T^k(l). \tag{3.1-24}$$

The time/space averaged estimate of the covariance matrix is a block matrix formed from the time/space estimates of the lagged correlation matrices:

$$\hat{\Sigma} = \begin{bmatrix} \hat{r}_{TS}(0) & \hat{r}_{TS}(-1) & \cdots & \hat{r}_{TS}(1-N) \\ \hat{r}_{TS}(1) & \hat{r}_{TS}(0) & \cdots & \hat{r}_{TS}(2-N) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_{TS}(N-1) & \hat{r}_{TS}(N-2) & \cdots & \hat{r}_{TS}(0) \end{bmatrix}. \tag{3.1-25}$$

This matrix estimate has block Toeplitz form. However, we emphasize that this estimate is not necessarily positive definite.

3.1.3.3 Time/Space Averaged Estimator with Time Clipping

More data is available for lower lags in calculating time/space averaged estimators of the lagged correlation matrices. Estimates of lagged correlation matrices for high lags will be less accurate and show more variation than those for lower lags. Time clipping (Huang 88) is a modification to the time/space averaged estimators intended to correct for this effect. For nonnegative lags l, the biased time averaged estimate with time clipping of the lagged correlation matrices is given by Equation 3.1-26:

$$\left[\hat{r}^k_{TC}(l)\right]_{i,j} = \frac{1}{N}\sum_{n=l+1}^{N} y_i^k(n)[y_j^k(n-l)]^* \frac{\sum_{n=1}^{N} y_i^k(n)[y_j^k(n)]^*}{\sum_{n=l+1}^{N} y_i^k(n)[y_j^k(n)]^*} \tag{3.1-26}$$

The unbiased estimate for nonnegative lags is given by Equation 3.1-27:

$$\left[\hat{r}^k_{TC}(l)\right]_{i,j} = \frac{1}{N-l}\sum_{n=l+1}^{N} y_i^k(n)[y_j^k(n-l)]^* \frac{\sum_{n=1}^{N} y_i^k(n)[y_j^k(n)]^*}{\sum_{n=l+1}^{N} y_i^k(n)[y_j^k(n)]^*} \tag{3.1-27}$$

For negative lags, the biased estimate with time clipping is given by Equation 3.1-28:

$$\left[\hat{r}^k_{TC}(l)\right]_{i,j} = \frac{1}{N}\sum_{n=|l|+1}^{N} y_i^k(n-|l|)[y_j^k(n)]^* \frac{\sum_{n=1}^{N} y_i^k(n)[y_j^k(n)]^*}{\sum_{n=|l|+1}^{N} y_i^k(n)[y_j^k(n)]^*} \tag{3.1-28}$$

The unbiased estimate for negative lags is given by Equation 3.1-29:

$$\left[\hat{r}^k_{TC}(l)\right]_{i,j} = \frac{1}{N-|l|}\sum_{n=|l|+1}^{N} y_i^k(n-|l|)[y_j^k(n)]^* \frac{\sum_{n=1}^{N} y_i^k(n)[y_j^k(n)]^*}{\sum_{n=|l|+1}^{N} y_i^k(n)[y_j^k(n)]^*} \tag{3.1-29}$$

The remainder of the algorithm closely resembles the time/space averaged estimate. The time/space averaged estimate with time clipping is found by averaging the time averaged estimate with time clipping over many realizations of the clutter:

$$\hat{r}_{TSC}(l) = \frac{1}{K} \sum_{k=1}^{K} \hat{r}_{TC}^{k}(l). \tag{3.1-30}$$

The estimate of the covariance matrix is formed in the usual way:

$$\hat{\Sigma} = \begin{bmatrix} \hat{r}_{TSC}(0) & \hat{r}_{TSC}(-1) & \cdots & \hat{r}_{TSC}(1-N) \\ \hat{r}_{TSC}(1) & \hat{r}_{TSC}(0) & \cdots & \hat{r}_{TSC}(2-N) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_{TSC}(N-1) & \hat{r}_{TSC}(N-2) & \cdots & \hat{r}_{TSC}(0) \end{bmatrix}. \tag{3.1-31}$$

3.1.3.4 Estimating the Amplitude

The amplitude α consists of a magnitude A and phase ϕ_0.

$$\alpha = A\, e^{j\phi_0} \tag{3.1-32}$$

Given a realization of the sum of signal and clutter, the amplitude α can be estimated as in Equation 3.1-33:

$$\hat{\alpha} \doteq \frac{s_0^H \hat{\Sigma}^{-1} x}{s_0^H \hat{\Sigma}^{-1} s_0}, \tag{3.1-33}$$

where x is a JN-element complex vector formed by concatenating together the radar returns for each time sample:

$$x = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix}. \tag{3.1-34}$$

The radar returns are assumed to be the sum of signal and clutter in estimating the amplitude. s_0 is the (known) JN-element complex steering vector:

$$s_0 = \begin{bmatrix} s_0(1) \\ s_0(2) \\ \vdots \\ s_0(N) \end{bmatrix} \tag{3.1-35}$$

and $s_0(n)$ is given by Equation 3.1-15.

The phase ϕ_0 can be estimated if the magnitude A of the amplitude is known. The estimator is

$$\hat{\phi}_0 = \frac{1}{2} \tan^{-1} \frac{\hat{\beta}_Q}{\hat{\beta}_I}, \tag{3.1-36}$$

where

$$\hat{\beta}_I + j\,\hat{\beta}_Q = \frac{s_0^H \hat{\Sigma}^{-1} x}{x_0^H \hat{\Sigma}^{-1} s_0}. \tag{3.1-37}$$

If the covariance matrix is known, the actual value Σ may be used instead of an estimate in Equations 3.1-33 and 3.1-37.

3.1.3.5 Inverting the Covariance Matrix

The inverse of the estimate of the covariance matrix is used in Equations 3.1-31 and 3.1-37. Since $\hat{\Sigma}$ can be a large matrix, numerical problems may arise in calculating an inverse. A clever method of inverting $\hat{\Sigma}$ is based on an SVD decomposition. Let

$$\hat{\Sigma} = U \Lambda V^H, \tag{3.1-38}$$

where Λ is a diagonal matrix of singular values, the columns of U are the corresponding left singular vectors, and the columns of V are the right singular vectors. Consider the product:

$$\hat{\Sigma}^{-1} = (U \Lambda V^H)^{-1} = (V^H)^{-1} \Lambda^{-1} U^{-1}. \tag{3.1-39}$$

Since U and V are unitary matrices:

$$V^H = V^{-1} \tag{3.1-40}$$

and

$$U^H = U^{-1} \tag{3.1-41}$$

Thus,

$$\hat{\Sigma}^{-1} = V (\Lambda)^{-1} U^H \tag{3.1-42}$$

For a positive definite Hermetian matrix $\hat{\Sigma}$, $U = V$ so that the SVD reduces to the eigen decomposition:

$$\hat{\Sigma} = Q \Lambda Q^H, \tag{3.1-43}$$

where Λ now contains the eigenvalues of $\hat{\Sigma}$ and $Q = U = V$. The problem of finding an inverse for the estimated covariance matrix has then been reduced to an SVD and finding the inverse of a diagonal matrix.

3.1.4 Filtering

Prediction error filters are provided to transform the radar returns to white noise uncorrelated in time and across channels. The results of these transformations are estimates of the innovations process for the models upon which the filters are based.

3.1.4.1 Innovations for White Noise

A simple special case arises when the clutter is white noise uncorrelated in time, but possibly correlated across channels. This case can be regarded as a special case of the AR process described in Section 3.1.1, where the order p is zero. If the radar returns consist solely of clutter modeled by white noise, then Equation 3.1-44 holds:

$$x(n) = y(n) = \varepsilon(n) = T\,v(n), \quad n = 1,2,\ldots,N,. \tag{3.1-44}$$

where some subscripts have been dropped. The corresponding linear filter is defined by Figure 3-1 or Equation 3.1-45:

$$v(n) = T^{-1} x(n), \quad n = 1,2,\ldots,N. \tag{3.1-45}$$

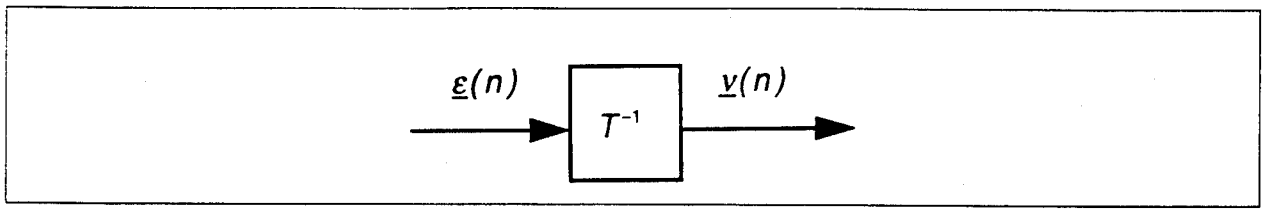

Figure 3-1: The Linear Filter for White Noise

Multiplying the radar returns by the inverse of T removes correlation between channels, where T is formed from the decomposition of the JxJ covariance matrix for the noise. In practice, the covariance matrix must be estimated. A procedure for estimating the (zero-lag) covariance matrix already existed at the start of this effort. The linear filter in Figure 3-1 was previously implemented as well.

3.1.4.2 Innovations for Temporally Correlated Clutter

The more general case of clutter, or the sum of clutter and noise, modeled as an AR process also already existed. If the radar returns consist solely of such an AR process, the linear filter shown in Figure 3-2 removes correlation in time and across channels. The tapped delay line structure implements Equation 3.1-46:

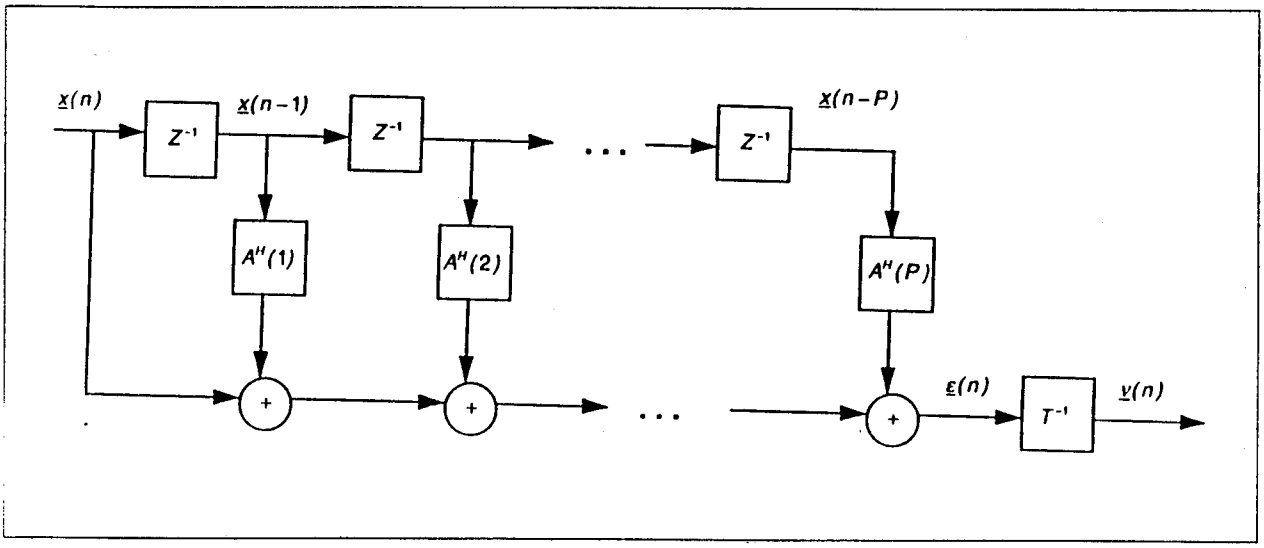

Figure 3-2: The Linear Filter for an AR Process

$$\varepsilon(n) = x(n) + \sum_{k=1}^{p} A^H(k)\,x(n-k),\quad n = 1,2,\ldots,N. \tag{3.1-46}$$

Cross-channel correlation is removed by Equation 3.1-47:

$$v(n) = T^{-1}\varepsilon(n),\quad n = 1,2,\ldots,N. \tag{3.1-47}$$

AR parameters can be estimated by the Yule-Walker, Nuttal-Strand, or Vieira-Morf algorithms (Marple 87).

3.1.4.3 Innovations for a Constant Signal in Clutter

A new filter was written to estimate the innovations in the model given by Equation 3.1-2. In this model, the radar returns consist of a signal with constant but unknown amplitude in clutter modeled as an AR process. The corresponding filter is shown in Figure 3-3. The first

part of the filter, given by Equation 3.1-48, removes the signal:

$$y(n) = x(n) - \hat{\alpha} s_0(n), \quad n = 1,2,\ldots,N. \tag{3.1-48}$$

The linear filter F_0 corresponds to the filter for the null hypothesis as shown in either Figure 3-1 or Figure 3-2. The outputs from the null hypothesis filter F_0 and the alternative hypothesis filter, F_1 shown in Figure 3-3, are used to calculate a generalized loglikelihood statistic used in signal detection.

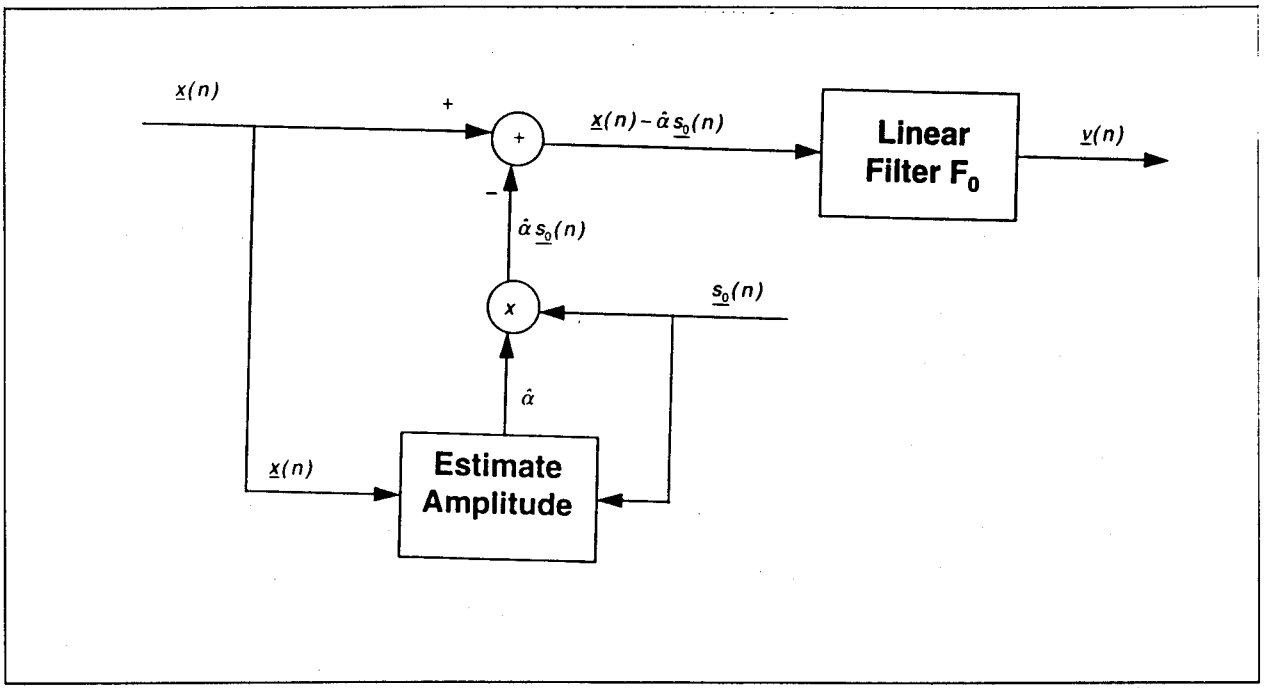

Figure 3-3: The Linear Filter for Signal and Clutter

3.2 Weibull Clutter Synthesis

This effort included the addition of a capability to synthesize Weibull-distributed Spherically Invariant Random Processes (SIRPs). Weibull SIRPs are useful for modeling terrain clutter. Time samples from a given range bin might follow a Gaussian distribution, but a given time sample might be Weibull-distributed across range bins (Rangaswamy 91).

3.2.1 Weibull SIRP White Noise

Let $w(1)$, $w(2)$, ..., $w(N)$ be white noise uncorrelated in time but possibly correlated across channels. Each time sample $w(n)$ is a J element (column) vector, where J is the number of channels. The cross-channel correlation is specified by the J x J covariance matrix Σ_w defined in Equation 3.2-1:

$$\Sigma_w = E[w(n) w^H(n)]. \tag{3.2-1}$$

$w^H(n)$ denotes the Hermitian transpose of $w(n)$. Equation 3.2-1 implies that the covariance matrix Σ_w is Hermitian.

The user of the MSPSS has the option of either specifying the covariance matrix Σ_w directly or the Cholesky, LDU, or Singular Value Decomposition of the covariance matrix. The Cholesky decomposition is given by Equation 3.2-2:

$$\Sigma_w = C_w C_w^H. \tag{3.2-2}$$

where C_w is a lower diagonal complex matrix. The LDU decomposition is

$$\Sigma_w = L_w D_w L_w^H. \tag{3.2-3}$$

where L_w is a lower triangular matrix with unity along its diagonal and D_w is a diagonal matrix.

Since Σ_w is Hermetian, the Singular Value Decomposition (SVD) of Σ_w is given by Equation (3.2-4):

$$\Sigma_w = Q_w \Lambda_w Q_w^H. \tag{3.2-4}$$

Λ_w is a diagonal matrix whose elements are the singular values of Σ_w. In this case, the singular values are also the eigenvalues of the matrix Σ_w. The columns of Q_w are right hand eigenvectors of Σ_w, and the rows of Q_w^H are left hand eigenvectors. Furthermore, Q_w^H is the inverse of Q_w.

The decomposition of the covariance matrix allows for the control of cross channel correlation of white noise. Let

$$w(n) = T_w v(n), \tag{3.2-5}$$

where T_w is either C_w, L_w, or Q_w, depending on which decomposition is chosen by the user. $v(n)$ is a white noise process uncorrelated both in time and across channels. If the Cholesky decomposition is chosen, each channel of $v(n)$ has a variance of unity. If the LDU decomposition is desired, the variance of the jth channel of $v(n)$ is the corresponding element along the principal diagonal of D_w. Finally, if a SVD is chosen, the channel variances are the diagonal elements of Λ_w. Equation 3.2-5 then ensures that the resulting white noise process $w(n)$ has the desired cross channel correlations and autocorrelations.

Formerly, the MSPSS supported the generation of Gaussian and K-distributed Spherically Invariant Random Processes (SIRPs). Under these synthesis procedures, $v(n)$ is synthesized from the desired distribution, either Gaussian or a K-distributed SIRP. The resulting process is then transformed to introduce cross-channel correlation.

The MSPSS now supports the synthesis of Weibull distributed SIRPs. Appendix B describes the Weibull distribution. The generation of SIRPs is most easily explained by first considering the special case in which the channel variances are unity and no correlation exists either across channels or in time. Accordingly, let $\xi(1)$, $\xi(2)$, ..., $\xi(N)$ be a Weibull-distributed SIRP with a mean of zero and identity as the correlation matrix. Each time sample, $\xi(n)$, in the SIRP is a J element column vector:

$$\xi(n) = \begin{bmatrix} \xi_1(n) \\ \xi_2(n) \\ \vdots \\ \xi_J(n) \end{bmatrix}, \tag{3.2-6}$$

where J is the number of channels. The entire process can be expressed as a JN element column vector formed by concatenation:

$$\xi = \begin{bmatrix} \xi(1) \\ \xi(2) \\ \cdot \\ \cdot \\ \cdot \\ \xi(N) \end{bmatrix}. \tag{3.2-7}$$

The distribution of a SIRP is completely specified by the mean, correlation matrix, and a certain quadratic form, q. Let M be the block-structured correlation matrix:

$$M = E[\xi \xi^H]. \tag{3.2-8}$$

In the special case considered here, the correlation matrix is the identity matrix. The quadratic form q is then defined as follows:

$$q = \xi^H M \xi. \tag{3.2-9}$$

If the SIRP under consideration is complex, the Probability Density Function (PDF) of the quadratic form q is given by:

$$f_q(q) = \frac{1}{2^{NJ}\Gamma(NJ)} q^{NJ-1} h_{2NJ}(q), \quad q > 0, \tag{3.2-10}$$

where $\Gamma(\ )$ is the Gamma function. For Weibull-distributed SIRPs,

$$h_{2NJ}(q) = (-2)^{NJ} \sum_{k=1}^{NJ} B_k \frac{A^k}{k!} q^{\frac{kb}{2} - NJ} \exp(-A q^{\frac{b}{2}}), \tag{3.2-11}$$

$$A = a \sigma^b, \tag{3.2-12}$$

$$B_k = \sum_{m=1}^{k} (-1)^m \binom{k}{m} \prod_{i=0}^{NJ-1} \left(\frac{mb}{2} - i \right), \tag{3.2-13}$$

$$\sigma^2 = \frac{1}{2} \left(\frac{1}{a} \right)^{\frac{2}{b}} \Gamma\left(1 + \frac{2}{b} \right) = 1. \tag{3.2-14}$$

σ^2 is the variance of the Weibull distribution. The parameter b, $0 < b \leq 2$, is specified by the user, while a is set such that the variance is unity. If the SIRP is real, then Equation 3.2-10 is replaced by Equation 3.2-15:

$$f_q(q) = \frac{1}{2^{\frac{NJ}{2}} \Gamma\left(\frac{NJ}{2} \right)} q^{\frac{NJ}{2} - 1} h_{NJ}(q), \quad q > 0. \tag{3.2-15}$$

The generation of Weibull-distributed SIRPs relies on the properties of a generalized spherical transformation of the SIRP. This coordinate transformation expresses the SIRP as a function of the JN random variables R, Θ, $\Phi_{j,n}$, $n = 1, 2, \ldots, N$; $j = 1, 2, \ldots, J$ if $n < N$, $j = 1, 2, \ldots, J-2$ if $n = N$. The generalized spherical transformation is given by Equations 3.2-16 through 3.2-21:

$$\xi_1(1) = R \cos(\Phi_{1,1}), \tag{3.2-16}$$

$$\xi_j(1) = R\cos(\Phi_{j,1})\prod_{k=1}^{j-1}\sin(\Phi_{k,1}),\ j = 2,3,\ldots,J. \tag{3.2-17}$$

$$\xi_j(n) = R\cos(\Phi_{j,n})\left[\prod_{k=1}^{j-1}\sin(\Phi_{k,n})\right]\left[\prod_{k_1=1}^{J}\prod_{k_2=1}^{n-1}\sin(\Phi_{k_1,k_2})\right],\ j=1,\ldots,J,\ n=2,\ldots,N-1, \tag{3.2-18}$$

$$\xi_j(N) = R\cos(\Phi_{j,N})\left[\prod_{k=1}^{j-1}\sin(\Phi_{k,N})\right]\left[\prod_{k_1=1}^{J}\prod_{k_2=1}^{N-1}\sin(\Phi_{k_1,k_2})\right],\ j = 1,2,\ldots,J-2 \tag{3.2-19}$$

$$\xi_{J-1}(N) = R\cos(\Theta)\left[\prod_{k=1}^{J-2}\sin(\Phi_{k,N})\right]\left[\prod_{k_1=1}^{J}\prod_{k_2=1}^{N-1}\sin(\Phi_{k_1,k_2})\right], \tag{3.2-20}$$

$$\xi_J(N) = R\sin(\Theta)\left[\prod_{k=1}^{J-2}\sin(\Phi_{k,N})\right]\left[\prod_{k_1=1}^{J}\prod_{k_2=1}^{N-1}\sin(\Phi_{k_1,k_2})\right]. \tag{3.2-21}$$

The random variable R is the distance of the Spherically Invariant Random Vector from the origin in the generalized spherical coordinate system. Its distribution does not depend on the other generalized spherical coordinates. This independence is the defining property of Spherically Invariant Random Vectors and SIRPs.

Note that Equation 3.2-22 holds for the special case of a SIRP in which the correlation matrix M is identity:

$$R^2 = \xi^H \xi = q. \tag{3.2-22}$$

When the SIRP is complex, the distribution of R is:

$$f_R(r) = \frac{r^{2NJ-1}}{2^{NJ-1}\Gamma(NJ)} h_{2NJ}(r^2). \tag{3.2-23}$$

When the SIRP is real, the distribution of R is:

$$f_R(r) = \frac{r^{NJ-1}}{2^{\frac{NJ}{2}-1}\Gamma\left(\frac{NJ}{2}\right)} h_{NJ}(r^2). \tag{3.2-24}$$

Note that $h_{NJ}(r^2)$ is only defined for even NJ. Either the number of channels or the number of time samples in a real Weibull SIRP must be even.

This brief discussion of some of the properties of SIRPs provides some indication of the theory underlying the algorithm for generating Weibull-distributed SIRPs that is implemented in the MSPSS:

- **Step 1:** Generate a white Gaussian process $z(1)$, $z(2)$, ..., $z(N)$ with a mean of zero and a correlation matrix of identity. The Gaussian process should be complex or real, depending on whether the synthesized SIRP is complex or real, respectively. Each element of the Gaussian process $z(n)$ should have J channels.
- **Step 2:** Calculate the norm R_G of the generated Gaussian process. The norm is defined by Equation 3.2-25:

$$R_G = \sqrt{\sum_{j=1}^{J}\sum_{n=1}^{N}|z_j(n)|^2}. \tag{3.2-25}$$

- **Step 3:** Generate a single realization of the random variable R. R has the distribution given by the PDF in Equations 3.2-23 or 3.2-24. R is generated by the rejection method, as described in Appendix C.
- **Step 4:** Generate the white SIRP $\xi(n)$:

$$\xi(n) = \frac{z(n)}{R_G} R . \tag{3.2-26}$$

In the general case, each channel of $\nu(n)$ may have different variances, each of which may differ from unity. The process $\nu(n)$ is generated from $\xi(n)$ by use of Equation 3.2-27:

$$\nu_j(n) = \sigma_{\nu_j} \xi_j(n), \quad j = 1,2,....,J; n = 1,2,....,N , \tag{3.2-27}$$

where σ_{ν_j} is the standard deviation of the jth channel of $\nu(n)$. Cross-channel correlation is then added using the decomposed covariance matrix as in Equation 3.2-5.

The MSPSS generates a user-specified number of trials of white noise. The user is given the option of generating each trial as a single realization of the SIRP or generating each time sample within the white noise process as a separate realization of the SIRP. Furthermore, separate channels can be a single realization of the SIRP, or each channel can be generated independently. These options leave the user with four choices, only three of which are possible for real processes:

- The white noise process is a single realization of the SIRP. Each element $\xi_j(n)$ appears as a Gaussian process within a single realization of the white noise. (Either the number of channels or the number of time samples must be even in a real SIRP.)
- The white noise process is generated as J realizations of the SIRP with each channel generated separately. (The number of time samples must be even in a real SIRP.)
- Each time sample is generated as a separate realization of the SIRP with J channels. (The number of channels must be even in a real SIRP.)
- Each channel and each time sample is generated as a separate realization of the SIRP. In effect, each element $\xi_j(n)$ is generated from a Weibull distribution. (This option is not possible in a real SIRP.)

These options affect the appropriate values to use for the number of channels and the number of time samples in the synthesis procedure defined above.

3.2.2 An AR Process with Weibull Driving Noise

Capabilities were added to use Weibull SIRPs as driving noise terms in various models with temporal correlation. These capabilities were added by modifying previously implemented models. For example, consider a clutter process, $y(n)$, modeled as an AR process as in Equation 3.2-38:

$$y(n) = -\sum_{k=1}^{p} A^H(k) y(n-k) + T\nu(n), \tag{3.2-28}$$

where T is either C, L, or Q in the Cholesky, LDU, or Singular Value Decomposition, respectively, of the driving noise covariance matrix Σ. $\nu(n)$ can be a Weibull SIRP synthesized as described in Section 3.2.1.

3.3 The Representative Model

A capability for synthesizing "Representative Model" processes was added to the MSPSS under this effort. Multichannel radar returns $x(n)$ are synthesized in the Representative Model as the sum of signal, clutter, and interference processes:

$$x(n) = s(n) + c(n) + I(n), \tag{3.3-1}$$

where $s(n)$ is the signal, $c(n)$ is the clutter, and $I(n)$ is the interference. One can think of each realization of the Representative Model as representing the radar returns from another range bin.

3.3.1 Signal

The signal can be modeled as a constant magnitude signal or a random signal. Let N denote the number of time samples in the radar returns, and let M be the number of channels. Then each time sample in the signal is a M-element vector, as shown in Equation 3.3-2:

$$s(n) = \begin{bmatrix} s_1(n) \\ s_2(n) \\ \cdot \\ \cdot \\ \cdot \\ s_M(n) \end{bmatrix}. \tag{3.3-2}$$

3.3.1.1 Constant Magnitude Signal

The constant magnitude signal model is given by Equation 3.3-3:

$$s_m(n) = a\, e^{2\pi\sqrt{-1}(m-1)\frac{d}{\lambda}\sin(\phi_s)}\, e^{2\pi\sqrt{-1}(n-1)(f_{dP}T + f_{dS}T)\sin(\phi_s)+\theta}, \quad n=1,2,\ldots,N;\ m=1,2,\ldots,M, \tag{3.3-3}$$

where

- a is the amplitude
- $\frac{d}{\lambda}$ is the normalized element spacing
- $f_{dP}T$ is the normalized platform Doppler center frequency
- $f_{dS}T$ is the normalized signal Doppler center frequency
- ϕ_s is the angular direction to the signal (in radians)
- θ is the initial phase.

The initial phase can be either user-specified or a random variable. If the initial phase is random, θ is from a uniform distribution on $[0, 2\pi]$. When random, θ is constant for each realization of the signal process, but varies from realization to realization.

3.3.1.2 Random Signal

The random signal model is specified by transforming a one-channel stochastic process, $s'(n)$, to a multichannel process, as shown by Equation 3.3-4:

$$s_m(n) = s'(n)\, e^{2\pi\sqrt{-1}(m-1)\frac{d}{\lambda}\sin(\phi_s)}, \quad n=1,2,\ldots,N;\ m=1,2,\ldots,M. \tag{3.3-4}$$

This model assumes that the process $s'(n)$ contains all the temporal correlation. The signal

amplitude is the same for all channels but is phase-shifted across channels.

The autocorrelation of the process $s'(n)$ is controlled by a "shaping function" approach. The lagged autocorrelation function for the process is defined by Equation 3.3-5:

$$R_s(l_t) = E[s'(n)s'^*(n - l_t)], \tag{3.3-5}$$

where $s'^*(n)$ is the complex conjugate of $s'(n)$. In the shaping function approach, the lagged autocorrelation function is specified as in Equation 3.3-6:

$$R_s(l_t) = \sigma_s^2 F_t^s(l_t) e^{2\pi\sqrt{-1}\, l_t (f_{dP} T + f_{dS} T)\sin(\phi_s)}, \tag{3.3-6}$$

where $F_t^s(l_t)$ is a temporal "shaping function," either Gaussian or exponential. Equation 3.3-7 defines the Gaussian temporal shaping function:

$$F_t^s(l_t) = (\mu_t)^{l_t^2}. \tag{3.3-7}$$

The exponential temporal shaping function is defined by Equation 3.3-8:

$$F_t^s(l_t) = (\mu_t)^{|l_t|}. \tag{3.3-8}$$

The covariance matrix is constructed from the individual lags as in Equation 3.3-9:

$$R_s = \begin{bmatrix} R_s(0) & R_s(1) & \cdots & R_s(p) \\ R_s(-1) & R_s(0) & \cdots & R_s(p-1) \\ \vdots & \vdots & & \vdots \\ R_s(-p) & R_s(-p+1) & \cdots & R_s(0) \end{bmatrix}, \tag{3.3-9}$$

The one-channel stochastic process $s'(n)$ is synthesized by either a block procedure or as an AR process. In the block procedure, a Cholesky decomposition is found for the covariance matrix:

$$R_s = C_s C_s^H, \tag{3.3-10}$$

where C_s is a lower triangular matrix. The desired process $s'(n)$ is found by transforming a zero-mean, unit variance process $v(n)$:

$$\begin{bmatrix} s'(1) \\ s'(2) \\ \vdots \\ s'(p) \end{bmatrix} = C_s \begin{bmatrix} v(1) \\ v(2) \\ \vdots \\ v(p) \end{bmatrix}, \tag{3.3-11}$$

$$s'(n) = \sum_{j=1}^{p} (C_s)_{p,j}\, v(n + j - p), \quad n = p+1, p+2, \ldots, N, \tag{3.3-12}$$

where $v(n)$ is uncorrelated in time. The user is given the option of choosing $v(n)$ from a Gaussian distribution, K-distributed SIRP, or Weibull-distributed SIRP. Equation 3.3-11 shows how to generate the first p elements of $s'(n)$. Theoretically, the order p of the covariance matrix should be set equal to the number of time samples N. The user, however, is given the option of choosing an order p less than N, in which case Equation 3.3-12 is used to synthesize the last $N-p$ time samples.

In synthesizing $s'(n)$ as an AR process, the Levinson-Wiggins-Robinson algorithm is used to solve the Yule-Walker equations shown in Display 3.3-13:

$$\begin{bmatrix} 1 & a_s^*(1) & a_s^*(2) & \cdots & a_s^*(p) \end{bmatrix} \begin{bmatrix} R_s(0) & R_s(1) & \cdots & R_s(p) \\ R_s(-1) & R_s(0) & \cdots & R_s(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_s(-p) & R_s(-p+1) & \cdots & R_s(0) \end{bmatrix} = \begin{bmatrix} \sigma_s^2 & 0 & \cdots & 0 \end{bmatrix} \quad (3.3\text{-}13)$$

The process $s'(n)$ is synthesized as the AR process shown in Equation 3.3-14:

$$s'(n) = -\sum_{k=1}^{p} a_s^*(k)\, s'(n-k) + v(n), \quad (3.3\text{-}14)$$

where $v(n)$ is a zero mean process uncorrelated in time. The variance of $v(n)$ is σ_s^2, and the user is given the option of choosing $v(n)$ from a Gaussian distribution, K-distributed SIRP, or Weibull-distributed SIRP.

3.3.2 Clutter

The spatial and temporal properties of the clutter do not separate so elegantly. The clutter model is specified by controlling the block covariance matrix for the clutter, shown in Equation 3.3-15:

$$R_c = \begin{bmatrix} R_c(0) & R_c(1) & \cdots & R_c(p) \\ R_c(-1) & R_c(0) & \cdots & R_c(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_c(-p) & R_c(-p+1) & \cdots & R_c(0) \end{bmatrix}, \quad (3.3\text{-}15)$$

where

$$R_c(l_t) = E\left[c(n)\, c^H(n-l_t)\right], \quad (3.3\text{-}16)$$

and $c(n)$ is the clutter process. Since the radar is assumed to have M channels, each time sample in the clutter process is an M-element vector. Therefore each entry in the block covariance matrix R_c is itself an $M \times M$ matrix. Equation 3.3-17 shows an expansion of one matrix:

$$R_c(l_t) = \begin{bmatrix} R_{1,1}(l_t) & R_{1,2}(l_t) & \cdots & R_{1,M}(l_t) \\ R_{2,1}(l_t) & R_{2,2}(l_t) & \cdots & R_{2,M}(l_t) \\ \vdots & \vdots & \ddots & \vdots \\ R_{M,1}(l_t) & R_{M,2}(l_t) & \cdots & R_{M,M}(l_t) \end{bmatrix} \quad (3.3\text{-}17)$$

Block covariance matrix elements are controlled by the function in Equation 3.3-18:

$$R_{i,j}(l_t) = \sigma_i\, \sigma_j\, F_t(l_A)\, F_s(l_B)\, e^{2\pi\sqrt{-1}\, l_s \frac{d}{\lambda} Sin(\phi_0)}\, e^{2\pi\sqrt{-1}\, l_t f_{dP} T Sin(\phi_0)}, \quad (3.3\text{-}18)$$

where

- σ_i and σ_j are the standard deviations of the two channels
- $F_t(l_A)$ and $F_s(l_B)$ are temporal and spatial shaping functions, respectively
- $\frac{d}{\lambda}$ is the normalized element spacing
- ϕ_0 is the angular beam direction
- $f_{dP}\,T$ is the normalized platform Doppler center frequency.

The spatial index is the difference of two channels:

$$l_s = j - i\,. \tag{3.3-19}$$

The indices l_A and l_B represent a rotation of the spatial and temporal indices to account for platform rotation:

$$\begin{bmatrix} l_A \\ l_B \end{bmatrix} = \begin{bmatrix} Cos\,\alpha & -Sin\,\alpha \\ Sin\,\alpha & Cos\,\alpha \end{bmatrix} \begin{bmatrix} l_t \\ l_s \end{bmatrix}, \tag{3.3-20}$$

where

$$\alpha = Tan^{-1}\left[\frac{f_{dP}\,T}{d/\lambda}\right] \tag{3.3-21}$$

The temporal shaping function, $F_t(l_A)$, can be a Gaussian, exponential correlation, or exponential power spectrum temporal shaping function. The Gaussian temporal shaping function is defined by Equation 3.3-22:

$$F_t(l_A) = \mu_t^{\,l_A^2}, \tag{3.3-22}$$

where μ_t is the one-lag temporal correlation parameter. Equation 3.3-23 defines the exponential correlation temporal shaping function:

$$F_t(l_A) = \mu_t^{\,|l_A|} \tag{3.3-23}$$

Equation 3.3-24 defines the exponential power spectrum temporal shaping function

$$F_t(l_A) = \frac{\zeta^2}{\zeta^2 + (2\pi)^2 l_A^2}, \quad 0.1 \le \zeta. \tag{3.3-24}$$

The spatial shaping function, $F_s(l_B)$, can be a Hanning, Hamming, Blackman-Harris or Gaussian spatial shaping function. The Hanning, Hamming, and Blackman-Harris spatial shaping functions are special cases of the function defined by Equation 3.3-25:

$$F_s(l_B) = \left\{\beta + (1-\beta)\,Cos\left[\frac{\pi\, l_B}{(N-1)\sin\alpha + (J-1)\cos\alpha}\right]\right\} Cos^2(\phi_0) \tag{3.3-25}$$

The Hanning function results when $\beta = 0.5$, the Hamming when $\beta = 0.543478261$, and the Blackman-Harris when $\beta = 0.53856$. The Gaussian spatial shaping function is defined by Equation 3.3-26:

$$F_s(l_B) = \mu_s^{\,l_B^2}\, Cos^2(\phi_0) \tag{3.3-26}$$

As with the Representative Model of the signal, both a block procedure and an AR process model are provided for synthesizing the clutter. The block procedure is based on a Cholesky decomposition of the block covariance matrix:

$$R_c = C_c\, C_c^H, \tag{3.3-27}$$

where C_c is a lower triangular matrix, which can be thought of as a block matrix also. The clutter is synthesized as defined by Equations 3.3-28 and 3.3-29:

$$\begin{bmatrix} c(1) \\ c(2) \\ \cdot \\ \cdot \\ \cdot \\ c(p) \end{bmatrix} = C_c \begin{bmatrix} v(1) \\ v(2) \\ \cdot \\ \cdot \\ \cdot \\ v(p) \end{bmatrix}, \tag{3.3-28}$$

$$c(n) = \sum_{j=1}^{p} (C_c)_{p,j}\, v(n+j-p), \quad n = p+1, p+2, \ldots, N, \tag{3.3-29}$$

where $(C_c)_{p,j}$ is the (p,j)th element of the matrix C_c.

The AR process model is based on the solution to the Yule-Walker equations given in Display 3.3-30:

$$\begin{bmatrix} I & A_c^H(1) & A_c^H(2) & \ldots & A_c^H(p) \end{bmatrix} \begin{bmatrix} R_c(0) & R_c(1) & \ldots & R_c(p) \\ R_c(-1) & R_c(0) & \ldots & R_c(p-1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R_c(-p) & R_c(-p+1) & \ldots & R_c(0) \end{bmatrix} = \begin{bmatrix} \Sigma_c & 0 & \ldots & 0 \end{bmatrix} \tag{3.3-30}$$

The solution is found by the Levinson-Wiggins-Robinson algorithm. The clutter process $c(n)$ is synthesized as in Equation 3.3-31:

$$c(n) = -\sum_{k=1}^{p} A_c^H(k)\, c(n-k) + T\, v(n), \tag{3.3-31}$$

where T is either C, L, or Q in the Cholesky, LDU, or Singular Value Decomposition of the driving noise covariance matrix Σ_c. $v(n)$ is a zero-mean process uncorrelated both temporally and spatially. Channel variances of $v(n)$ are unity for the Cholesky decomposition, and $v(n)$ can be synthesized from a K-distributed SIRP, Weibull-distributed SIRP, or Gaussian distribution.

3.3.3 Interference

Two models are provided for interference, a direct path white noise interference model and a direct path partially correlated noise interference model. Capabilities exist in the MSPSS for adding together several synthesized processes. Thus, the user can synthesize interference for several jammers separately, and then add the results together.

3.3.3.1 Direct Path White Noise Interference Model

The direct path interference model is defined by Equation 3.3-32:

$$I_m(n) = I'(n)\, e^{2\pi\sqrt{-1}(m-1)\frac{d}{\lambda}\mathrm{Sin}(\phi_I)}\, e^{\pi\sqrt{-1}\,n\,(f_{dP}T + f_{dI}T)\mathrm{Sin}(\phi_I)}, \quad n = 1,2,\ldots,N;\ m = 1,2,\ldots,M, \tag{3.3-32}$$

where $I'(n)$ is a zero-mean white noise process with variance σ_I^2. Other parameters consist of:

- $\frac{d}{\lambda}$ is the normalized element spacing
- ϕ_I is the angular direction to the jammer (in radians)
- $f_{dP}\,T$ is the normalized platform Doppler center frequency
- $f_{dI}\,T$ is the normalized interference Doppler center frequency

The user is given the option of choosing $I'(n)$ from a Gaussian distribution, K-distributed SIRP, or Weibull-distributed SIRP.

3.3.3.2 Direct Path Partially Correlated Noise Interference Model

The direct path partially correlated noise interference model closely resembles the random signal model, with a difference of a factor of two in some terms. The interference is found by transforming a one-channel partially correlated noise process $I'(n)$, as shown in Equation 3.3-33:

$$I_m(n) = I'(n)\,e^{2\pi\sqrt{-1}(m-1)\frac{d}{\lambda}Sin(\phi_I)},\ n=1,2,\ldots,N;\ m=1,2,\ldots,M. \tag{3.3-33}$$

Intertemporal correlation of $I'(n)$ is controlled by the function defined in Equation 3.3-34:

$$R_I(l_t) = E[I'(n)I'^{*}(n)] = \sigma_I^2 F_t^I(l_t)\,e^{\pi\sqrt{-1}n(f_{dP}T + f_{dI}T)Sin(\phi_I)}, \tag{3.3-34}$$

where $F_t^I(l_t)$ is either a Gaussian or exponential temporal shaping function. The Gaussian temporal shaping function is defined by Equation 3.3-35:

$$F_t^I(l_t) = (\mu_t)^{(l_t^2)}, \tag{3.3-35}$$

where μ_t is the one-lag temporal correlation parameter. The Exponential temporal shaping function is defined by Equation 3.3-36:

$$F_t^I(l_t) = (\mu_t)^{|l_t^2|}. \tag{3.3-36}$$

The covariance matrix for the interference is constructed from the individual lags:

$$R_I = \begin{bmatrix} R_I(0) & R_I(1) & \ldots & R_I(p) \\ R_I(-1) & R_I(0) & \ldots & R_I(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_I(-p) & R_I(-p+1) & \ldots & R_I(0) \end{bmatrix}, \tag{3.3-37}$$

Both a block procedure and an AR model are provided for synthesizing the one-channel stochastic process $I'(n)$. A Cholesky decomposition of the block covariance matrix is used in the block procedure. The Cholesky decomposition is defined by Equation 3.3-38:

$$R_I = C_I\,C_I^H, \tag{3.3-38}$$

where C_I is a lower triangular matrix. The desired process $I'(n)$ is found by transforming a zero-mean, unit variance process $\nu(n)$:

$$\begin{bmatrix} I'(1) \\ I'(2) \\ \cdot \\ \cdot \\ \cdot \\ I'(p) \end{bmatrix} = C_I \begin{bmatrix} v(1) \\ v(2) \\ \cdot \\ \cdot \\ \cdot \\ v(p) \end{bmatrix}, \tag{3.3-39}$$

$$I'(n) = \sum_{j=1}^{p} (C_I)_{p,j}\, v(n+j-p), \quad n = p+1, p+2, \ldots, N, \tag{3.3-40}$$

where $v(n)$ is uncorrelated in time. The user is given the option of choosing $v(n)$ from a Gaussian distribution, K-distributed SIRP, or Weibull-distributed SIRP. The first p time samples in $I'(n)$ are synthesized as in Equation 3.3-39, while the remaining $N-p$ time samples, if any, are synthesized by Equation 3.3-40.

The synthesis process for $I'(n)$ as an AR process begins by using the Levinson-Wiggins-Robinson algorithm to solve the Yule-Walker equations shown in Display 3.3-41:

$$\begin{bmatrix} 1 & a_I^*(1) & a_I^*(2) & \ldots & a_I^*(p) \end{bmatrix} \begin{bmatrix} R_I(0) & R_I(1) & \ldots & R_I(p) \\ R_I(-1) & R_I(0) & \ldots & R_I(p-1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R_I(-p) & R_I(-p+1) & \ldots & R_I(0) \end{bmatrix} = \begin{bmatrix} \sigma_I^2 & 0 & \ldots & 0 \end{bmatrix} \tag{3.3-41}$$

The process $I'(n)$ is synthesized as the AR process shown in Equation 3.3-42:

$$I'(n) = -\sum_{k=1}^{p} a_I^*(k)\, I'(n-k) + v(n), \tag{3.3-42}$$

where $v(n)$ is a zero mean process uncorrelated in time. The variance of $v(n)$ is σ_I^2, and the user is given the option of choosing $v(n)$ from a Gaussian distribution, K-distributed SIRP, or Weibull-distributed SIRP.

3.4 A Two-Dimensional FFT of Clutter Covariance Matrix

A capability was added to graph a two-dimensional Fast Fourier Transform (FFT) of the covariance matrix for the Representative Model clutter (Section 3.3.2). This covariance matrix is not estimated, but is determined using the shaping functions.

The clutter covariance matrix, R_{cc}, can be represented as in Equation 3.4-1:

$$R_{cc} = \begin{bmatrix} r_{cc}(1-J,1-N) & r_{cc}(1-J,2-N) & \ldots & r_{cc}(1-J,N-1) \\ r_{cc}(2-J,1-N) & r_{cc}(2-J,2-N) & \ldots & r_{cc}(2-J,N-1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ r_{cc}(J-1,1-N) & r_{cc}(J-1,2-N) & \ldots & r_{cc}(J-1,N-1) \end{bmatrix}. \tag{3.4-1}$$

Each element of the $2J-1$ x $2N-1$ matrix R_{cc} specifies a correlation:

$$r_{cc}(l_s, l_t) = E\left[c_{j_1}(n)\, c_{j_2}^*(n-l_t)\right] \tag{3.4-2}$$

where

$$l_s = j_2 - j_1, \quad j_1, j_2 = 1, 2, \ldots, J, \tag{3.4-3}$$

and J is the number of channels. The spatial correlation is assumed to depend only on the difference between channels, not the particular pair of channels selected. Using the shaping function approach, the elements of the covariance matrix are specified by Equation 3.4-4:

$$r_{cc}(l_s, l_t) = \sigma^2 F_t(l_A) F_s(l_B) e^{2\pi\sqrt{-1}\, l_s \frac{d}{\lambda} Sin(\phi_0)} e^{2\pi\sqrt{-1}\, l_t f_{dP} T Sin(\phi_0)}, \tag{3.4-4}$$

where

- σ is the channel standard deviation (assumed constant across channels)
- $F_t(l_A)$ and $F_s(l_B)$ are temporal and spatial shaping functions, respectively, as specified in Equations 3.3-22 through 3.3-26
- $\frac{d}{\lambda}$ is the normalized element spacing
- ϕ_0 is the angular beam direction
- $f_{dP} T$ is the normalized platform Doppler center frequency.

The indices l_A and l_B are found by rotating the spatial and temporal lags, as given by Equations 3-3-20 and 3-3-21. Equations 3.3-20 and 3.3-21 are repeated for convenience as Equations 3.4-5 and 3.4-6:

$$\begin{bmatrix} l_A \\ l_B \end{bmatrix} = \begin{bmatrix} Cos\,\alpha & -Sin\,\alpha \\ Sin\,\alpha & Cos\,\alpha \end{bmatrix} \begin{bmatrix} l_t \\ l_s \end{bmatrix}, \tag{3.4-5}$$

$$\alpha = Tan^{-1}\left[\frac{f_{dP} T}{d/\lambda}\right] \tag{3.4-6}$$

The covariance matrix R_{cc} is first windowed before applying a two-dimensional FFT. The window is created by using the Dolph-Chebychev weights given by Equation 3.4-7:

$$W_k(p) = \frac{2}{p}\left\{S + 2\sum_{n=1}^{K} T_{p-1}\left[z_0 \cos\left[\frac{\pi n}{p}\right]\right] \cos\left[\frac{2\pi n}{p}\left[k - \frac{p+1}{2}\right]\right]\right\}, \quad k = 1, 2, \ldots, p, \tag{3.4-7}$$

where

$$S = 10^{\frac{\alpha}{20}}, \tag{3.4-8}$$

α is the sidelobe level (in decibels),

$$K = \begin{cases} \frac{p-1}{2}, & p \text{ odd} \\ \frac{p}{2} - 1, & p \text{ even} \end{cases} \tag{3.4-9}$$

$$z_0 = \cosh\left[\frac{\cosh^{-1}(S)}{p-1}\right], \tag{3.4-10}$$

and

$$T_{p-1}(z) = \begin{cases} \cos[(p-1)Cos^{-1}(z)] & \text{if } |z| \le 1 \\ \cosh[(p-1)Cosh^{-1}(z)] & \text{if } |z| \le 1 \end{cases} \tag{3.4-11}$$

The Dolph-Chebychev weights for the temporal dimension are given by Equation 3.4-12:

$$\Omega_T = \begin{bmatrix} \omega_{1-N} & 0 & \cdots & 0 \\ 0 & \omega_{2-N} & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & \omega_{N-1} \end{bmatrix}, \tag{3.4-12}$$

where

$$\omega_n = W_{n+N}(2N-1), \quad n = 1-N, 2-N, \ldots, N-1. \tag{3.4-13}$$

The Dolph-Chebychev weights for the spatial dimension are given by Equation 3.4-14:

$$\Omega_S = \begin{bmatrix} \omega_{1-J} & 0 & \cdots & 0 \\ 0 & \omega_{2-J} & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & \omega_{J-1} \end{bmatrix} \tag{3.4-14}$$

where

$$\omega_j = W_{j+J}(2J-1), \quad j = 1-J, 2-J, \ldots, J-1. \tag{3.4-15}$$

The window is formed as in Equation 3.5-15:

$$R'_{cc} = \Omega_S R_{cc} \Omega_T \tag{3.4-16}$$

A two dimensional FFT is the applied to R'_{cc}. The particular implementation used in the MSPSS is based on the algorithm used in Khoros.

3.5 Calculation of State Space Parameters

State space synthesis and analysis capabilities were implemented in the MSPSS under a previous effort (Vienneau 94). These capabilities included a fairly complex algorithm for estimating the parameters of an innovations representation of a state space process based on the block covariance matrix. The block covariance matrix is estimated from realizations of the process. The state space parameter estimates are often difficult to validate due to the possibility of a basis transformation of the state vector. Thus, it is difficult to test the estimation algorithm.

A capability was implemented under this effort for determining state space parameters from a block covariance matrix calculated analytically. The covariance matrix used as input to the state space estimation algorithm no longer contains statistical variation, unlike those estimated from a synthesized process. This capability is intended to simplify the testing of the estimation algorithm. The block covariance matrices are limited to those corresponding to an Autoregressive (AR) process.

A stationary state space process can be described by the innovations representation given in Equations 3.5-1 and 3.5-2:

$$\alpha(n+1) = F\,\alpha(n) + K\,\varepsilon(n), \tag{3.5-1}$$

$$x(n) = H^H \alpha(n) + \varepsilon(n). \tag{3.5-2}$$

$\alpha(n)$ is an m element (column) vector representing the state variables, where m is the order of

the process. $x(n)$ is a J element (column) vector denoting the process output. The mxm matrix F determines how the state variables evolve through time and is known as the system dynamics matrix. K is an mxJ matrix known as the Kalman gain, and H is the mxJ observations matrix. The J element (column) vector $\varepsilon(n)$ denotes driving noise, and is called the innovations process. The covariance matrix for the innovations process is defined by Equation 3.5-3:

$$\Sigma = E[\varepsilon(n)\varepsilon^H(n)]. \tag{3.5-3}$$

Many of the capabilities provided by the MSPSS relate to AR models. A pth order AR model is defined by Equation 3.5-4:

$$x(n) = -\sum_{k=1}^{p} A^H(k)x(n-k) + \varepsilon(n). \tag{3.5-4}$$

This AR process can be cast into an innovations representation of a state space process. The pJ state variables consist of the first p lagged values of the process:

$$\alpha(n) = \begin{bmatrix} x(n-1) \\ x(n-2) \\ \vdots \\ x(n-p) \end{bmatrix} \tag{3.5-5}$$

The system dynamics matrix for an AR process is given by Equation 3.5-6:

$$F = \begin{bmatrix} -A^H(1) & -A^H(2) & \cdots & -A^H(p-1) & -A^H(p) \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \tag{3.5-6}$$

The Kalman gain is shown in Equation 3.5-7:

$$K = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.5-7}$$

The Hermitian transpose of the observation matrix is shown in Equation 3.5-8:

$$H^H = [-A^H(1) \; -A^H(2) \; \cdots \; -A^H(p)] \tag{3.5-8}$$

Throughout the MSPSS, AR processes are specified by the parameters of shaping functions for a block covariance matrix. The block covariance matrix for an order p AR process is given by Equation 3.5-9:

$$R = \begin{bmatrix} R(0) & R(1) & \cdots & R(p) \\ R(-1) & R(0) & \cdots & R(p-1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R(-p) & R(-p+1) & \cdots & R(0) \end{bmatrix}, \tag{3.5-9}$$

where $R(l)$ is given by Equation 3.5-10:

$$R(l) = E[x(n)x^H(n-l)]. \tag{3.5-10}$$

Lagged covariance matrices need only be determined for positive lags. Covariances for negative lags can be found from Equation 3.5-11, which follows from Equation 3.5-10:

$$R(-l) = R^H(l). \tag{3.5-11}$$

Two shaping functions are provided for the capability of estimating state space parameters from a known covariance matrix. The Gaussian shaping function for the cross-channel correlation function (between channels i and j) is given by Equation 3.5-12:

$$R_{i,j}(k) = K_{i,j} \frac{\lambda_{i,j}^{[(k-l_{i,j})^2]}}{\lambda_{i,j}^{(l_{i,j}^2)}} e^{2\pi\phi T k\sqrt{-1}}, \quad i \le j \tag{3.5-12}$$

where $K_{i,j}$ is a magnitude which relates to the cross-channel correlation, $\lambda_{i,j}$ is the one-lag temporal cross-channel correlation parameter, $l_{i,j}$ is the lag at which the function peaks, ϕ is the Doppler shift, and T is the sampling period. The shaping function is constructed to ensure Equation 3.5-13 holds:

$$R_{i,j}(k) = R_{j,i}^*(-k), \quad i > j. \tag{3.5-13}$$

Equation 3.5-14 defines the exponential shaping function:

$$R_{i,j}(k) = K_{i,j} \frac{\lambda_{i,j}^{|k-l_{i,j}|}}{\lambda_{i,j}^{|l_{i,j}|}} e^{2\pi\phi T k\sqrt{-1}}, \quad i \le j \tag{3.5-14}$$

The AR parameters are determined using the Levinson-Wiggins-Robinson algorithm to solve the Yule-Walker equations, which are based on the block covariance matrix of the same order as the AR process. Equation 3.5-15 shows an example of the Yule-Walker system of equations, for an AR process of order three:

$$\begin{bmatrix} I & A^H(1) & A^H(2) & A^H(3) \end{bmatrix} \begin{bmatrix} R(0) & R(1) & R(2) & R(3) \\ R(-1) & R(0) & R(1) & R(2) \\ R(-2) & R(-1) & R(0) & R(1) \\ R(-3) & R(-2) & R(-1) & R(0) \end{bmatrix} = \begin{bmatrix} \Sigma & 0 & 0 & 0 \end{bmatrix} \tag{3.5-15}$$

Notice that when Equations 3.5-12 or 3.5-14 are used to generate the lagged covariance matrices, such a set of parameters may not correspond to an AR process exactly for any order. In those cases, the AR model obtained via the Yule-Walker equations is a minimum mean square model.

3.5.1 State Space Closed Form Method

A capability to test the state space Canonical Correlations algorithm (Roman 93 and Vienneau 94) was implemented under this contract. The Canonical Correlations algorithm estimates the state space parameters F, K, H, and Σ from the block covariance matrix. The

Canonical Correlations algorithm estimates the state space model order, and the block covariance matrix input into the algorithm may be larger than the order of the state space model.

The Gaussian and Exponential shaping functions are used to determine the elements of the block covariance matrix up to the order of the AR process. But how should theoretically exact elements of the covariance matrix be determined for higher orders? Two methods were implemented for answering this question, the State Space Closed Form and the AR Recursion Method. Appendix E shows that these two methods are theoretically equivalent when the correlation functions describe the AR process exactly. This section describes the State Space Closed Form Method, while the next section describes the AR Recursion Method.

The State Space Closed Form Method begins with the values of the state space parameters F, K, H, and Σ for the correct order. For an AR process, these parameters are as in Equations 3.5-6, 3.5-7, and 3.5-8, where the AR coefficients and Σ are found by solving the Yule-Walker equations. The outputs of the State Space Closed Form Method are the values of the lagged covariance matrices $R(0)$, $R(1)$, $R(2)$, ... $R(m)$, where $p \leq m$.

Intermediate matrices P and Γ are used in the State Space Closed Form Method. P is the solution of Equation 3.5-16:

$$P = F P F^H + K \Sigma K^H , \tag{3.5-16}$$

while Γ is found from Equation 3.5-17:

$$\Gamma = F P H + K \Sigma \tag{3.5-17}$$

Equation 3.5-16 defines P, but does not give a closed form solution that can be used in an algorithm. The solution to Equation 3.5-16 can be found as the limit of a matrix series:

$$P = \lim_{k \to \infty} P(k), \tag{3.5-18}$$

where

$$P(0) = I \tag{3.5-19}$$

and

$$P(k+1) = F P(k) F^H + K \Sigma K^H . \tag{3.5-20}$$

The program implementing the State Space Closed Form Method sets $P(0)$ to the identity matrix, as in Equation 3.5-19. Successive values of $P(k)$ are calculated as in Equation 3.5-20 until the spectral radius of the difference in successive values is below some appropriate small bound. As is summarized in Appendix D, this condition imposes a lower bound on all "natural norms" of the difference in successive values of $P(k)$.

The final results of the State Space Closed Form Method, lagged covariance matrices are calculated as in Equations 3.5-21 and 3.5-22:

$$R(0) = H^H P H + \Sigma \tag{3.5-21}$$

$$R(l) = H^H F^{l-1} \Gamma, \quad l = 1, 2, ..., m . \tag{3.5-22}$$

In the cases where the lagged covariance matrices used to obtain the AR parameters do not correspond to a true AR process of order p, the lags generated via Equations 3.5-21 and 3.5-22 for $l = 0, 1, ..., p$ will be different from the lags used at the start of the procedure.

3.5.2 AR Recursion Method

The AR Recursion Method is more straightforward. The first p lagged covariance matrices, where p is the order of the AR process, are found from the shaping function. These are the lagged covariance matrices used in the Yule-Walker Equations to determine the AR coefficients and driving noise covariance matrix. Lagged covariance matrices for higher order lags are found by the recursion relationship shown in Equation 3.5-23:

$$R(l) = -\sum_{k=1}^{M} A^H(k) R(l-k), \quad l = p+1, p+2, \ldots, m. \tag{3.5-23}$$

However, with this method there will be a model inconsistency in the cases where the shaping function lags for $l = 0, 1, \ldots, p$ do not correspond exactly to a true AR process of order p. This is likely to be the case more often than not because the lagged products from the shaping functions are not analytically related to an AR process. In contrast, once the state-space model parameters F, K, H, Σ, P, and Γ are calculated and covariance lags are generated using Equations 3.5-21 and 3.5-22, such a set of covariance lags corresponds exactly to an AR process.

3.6 Sixteen Channels

The MSPSS imposes a maximum limit to the number of channels in all radar signals to be synthesized or analyzed. This limit was four channels at the start of this effort. It was increased to 16 under this effort. The FFT described in Section 3.4 uses a matrix where one of its dimensions is determined by number of spatial lags. The limit for the number of spatial lags in the FFT capability is 32.

4. IMPLEMENTATION

Section 3 describes new capabilities implemented under this effort. These capabilities were implemented as a combination of programs under the menu-based subsystem, modifications to existing programs, and new analysis sequences under the UFI-based system. Two new menu-based programs and four analysis sequences provide capabilities for analyzing a constant magnitude signal with unknown amplitude. Existing synthesis programs were modified to provide for the synthesis of Weibull SIRPs. The Representative Model was implemented as a single program under the menu-based subsystem. A new program for the menu-based subsystem provides the capability of estimating state space parameters from the AR parameters of a specified correlation function, and some subroutines were modified to correct a bug in the program for estimating state space parameters for a synthesized process. A two dimensional FFT of the Representative Model correlation function was implemented as a single program under the menu-based subsystem. The maximum number of channels was raised to 16 through a global change to the MSPSS, in both the menu-based and UFI-based subsystems.

4.1 A Signal with Unknown Amplitude

Two new routines were written in the menu-based system to implement the functionality needed for the analysis of a constant magnitude signal with unknown amplitude in interference. One of the existing menu-based synthesis routines was modified. Four new sequences were added to the UFI-based MSPSS subsytem.

The use of the modified menu-based synthesis routine is illustrated in Figure 4-1. This figure shows the creation of a file containing five realizations of the sum of signal and white noise. User inputs are shown in boldface; all values are chosen for illustrative purposes only. The real and imaginary parts of the synthesized process are graphed in Figures 4-2 and 4-3. Both the real and imaginary parts of each channel vary sinusoidally, but the magnitude of each channel is two. The two new menu-based routines are for

- Estimating the covariance matrix for the interference and the unknown amplitude of the signal
- Subtracting the estimated signal from the radar returns as described in Figure 3-3.

4.1.1 Covariance Matrix and Amplitude Estimation

The use of the amplitude estimation routine is shown in Figures 4-4 and 4-5. The names of two files must be input by the user. The first file should have data synthesized under the null hypothesis so that no signal is present. The second file contains data from which the signal amplitude is estimated. The radar returns in these files should contain the same number of time samples, but the number of realizations can vary between the two files. The data from the first file is used to estimate the covariance matrix. The user can choose to see the covariance estimate from each realization, or, as in the example, only the average covariance matrix estimate. The covariance matrix estimate used in amplitude estimation is found by averaging over all the realizations in the first file. Once all the data in the first file is processed, the amplitude is estimated for each trial in the second file. Thus, the second file is treated as if each realization contains the sum of signal and clutter, where the signal is modeled as described in Section 3.1.2. Note that this program can estimate the phase of a constant magnitude signal with known magnitude, as well as the unknown amplitude estimate shown in Figures 4-4 and 4-5.

```
        MULTI-CHANNEL (M/C) PROCESS SYNTHESIS MENU

M0  -- Generate Gaussian noise                  Choose appropriate synthesis
M1  -- Method 1 (MC1) process synthesis         option from submenu
M2  -- Method 2 (MC2) process synthesis
                .
                .
                .
Enter a command or Q to return to the MAIN menu: m2
Linking program msirp, please stand by ...
f77 -O -C -pipe -cg89 sun4/msirp.o -Lsun4 -lmc -lvec -lmat -lmc -L/home/marlin/tom/lib/sun4 -lfsl  -o
MSIRP -- Multichannel AR Process Generation with SIRPs
Version 1.16
Do you want to set the pseudo random generator seeds (y/n) [N] ?
Number of channels ? 3                          Boxed values are defaults
Number of points to be generated and saved ? 200
Enter the number of trials to be generated ? 1
Add a signal component (y/n) [Y] ? y
ENTER PARAMETERS FOR THE SIGNAL.                Describe the signal
Do you want a Deterministic, Generalized deterministic or an AR process (d/g/a) [D] ? d
Enter the magnitude of the amplitude ? 2.0
Enter the normalized Doppler (fd T) ? 0.01
Enter the normalized element spacing ? 0.083333333
Enter the angle to the signal in radians ? 1.570796327
Amplitude magnitude:  2.0000e+00
Normalized Doppler f(d) T:  1.0000e-02
Normalized element spacing:  8.3333e-02
Angle to the signal:  1.5708e+00
Initial phase PHI is random.
Initial phase is constant within each trial.
Are these parameters okay (y/n) [Y] ?
Add a clutter component (y/n) [Y] ? n
Add a white noise component (y/n) [Y] ? n
Total output file name ? rlv
Title of this dataset ? Constant magnitude signal
Signal output file name ?
No file name entered, is this okay ? y
ok?
```

Figure 4-1: Synthesizing a Constant Magnitude Signal in White Noise

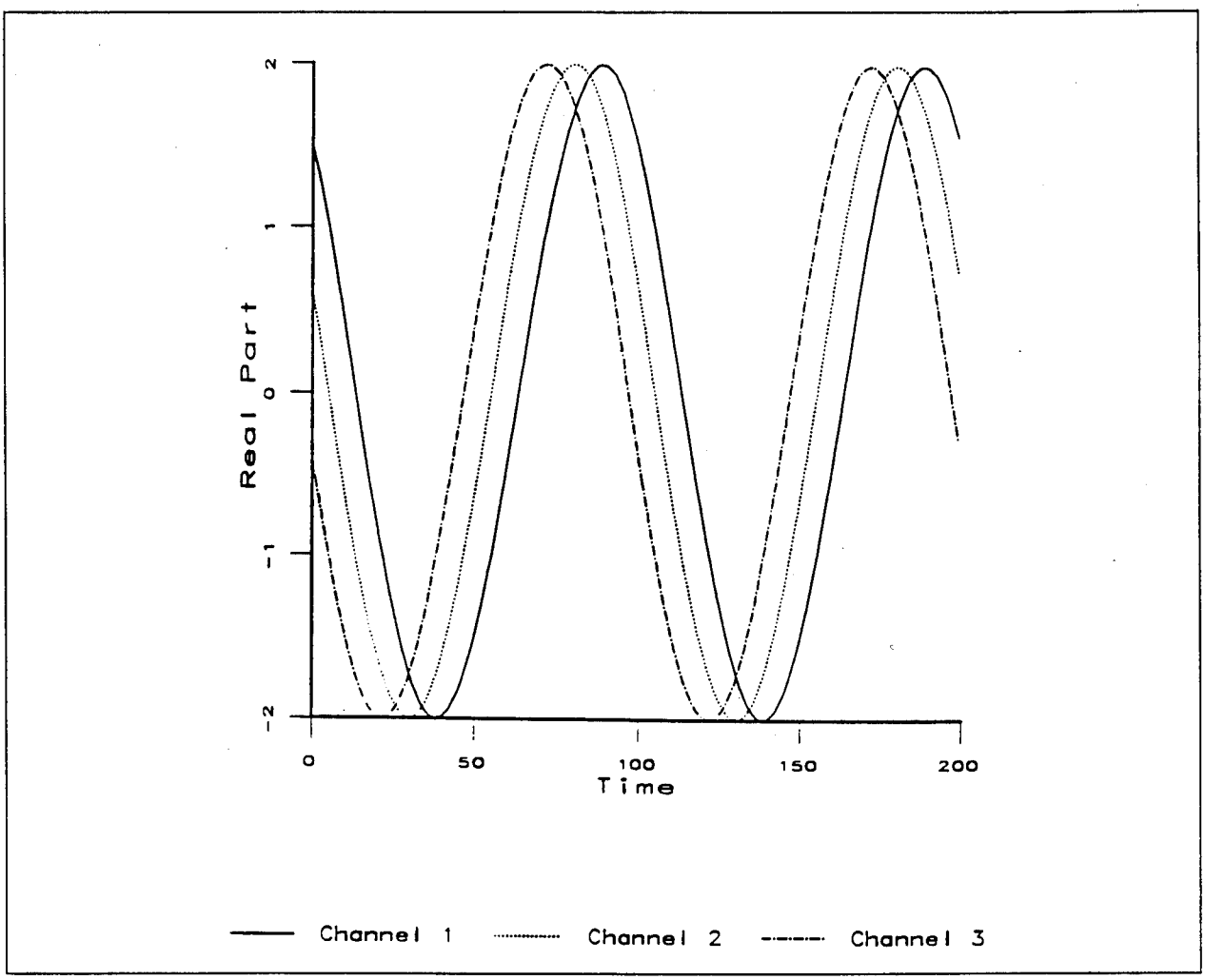

Figure 4-2: Real Part of a Constant Magnitude Signal

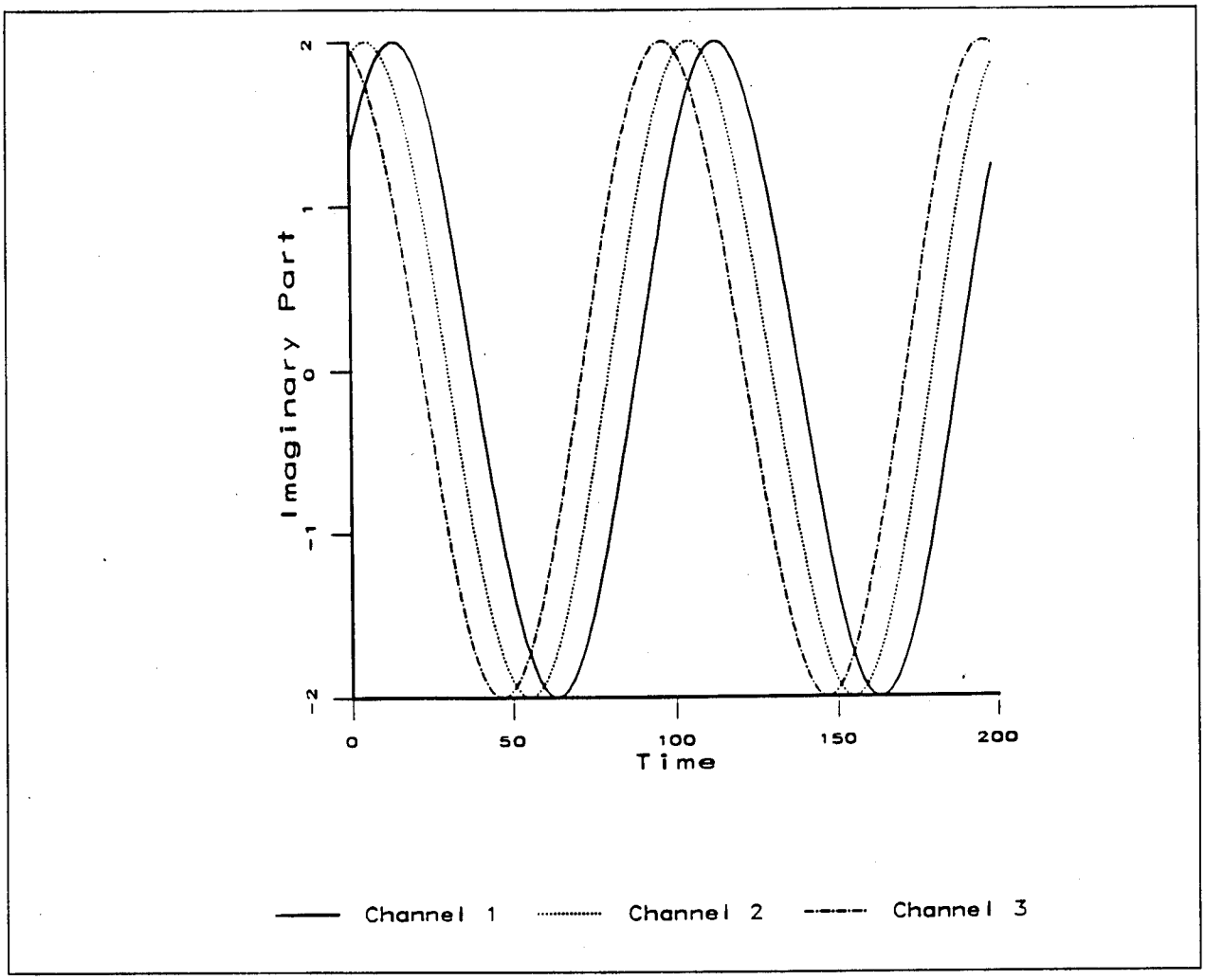

Figure 4-3: Imaginary Part of a Constant Magnitude Signal

```
Linking program mamp, please stand by ...
f77 -O -C -pipe -cg89 sun4/mamp.o -Lsun4 -lmc -lvec -lmat -lmc -L/home/marlin/tom/lib/sun4 -lfsl  -o
MAMP -- Estimates amplitude of multichannel constant signal in clutter.
Version 1.14
Null hypothesis input file name ? rlv0              Already existing noise file with 2 time samples
Title: Gaussian noise, covariance matrix is identity
Alternate hypothesis input file name ? rlv1         Sum of signal and noise
Title: Signal + Noise, Amplitude 1.25, Doppler 0.0, Spacing 0.125 Angle 90o
Amplitude estimates output file name ? rlva         Amplitude file is created
Title of this dataset ? Amplitude estimates         Help is available
Estimation: sample Matrix, time/space Average, time Clipping(m/a/c) [M] ? ?
Enter "m" to use sample covariance matrix estimate.
Enter "a" to use the time/space averaged estimator for the correlation
matrix.
Enter "c" to use the time/space averaged estimator with time clipping.
Estimation: sample Matrix, time/space Average, time Clipping(m/a/c) [M] ? a
Use biased estimate (y/n) [N] ? y
Enter the normalized Doppler (fd T) ? 0.0
Enter the normalized element spacing ? 0.125
Enter the angle to the signal in radians ? 1.570796327
Is the magnitude of the amplitude known (y/n) [N] ?
Display covariance estimates for each trial (y/n) [N] ?
Mean value for 10000 trials.
  Average covariance matrix:                        Noise was synthesized
Columns 1 through 2                                   with Identity covariance
  (9.921660e-01,0.000000e+00)   (-8.681035e-03,-3.323868e-03)
  (-8.681035e-03,3.323868e-03)  (1.006634e+00,0.000000e+00)
  (3.554484e-03,-1.095882e-03)  (-6.401982e-03,4.316789e-03)
  (-3.996205e-03,-3.282540e-03)  (-4.838127e-03,-9.185376e-04)

Columns 3 through 4
  (3.554484e-03,1.095882e-03)  (-3.996205e-03,3.282540e-03)
  (-6.401982e-03,-4.316789e-03)  (-4.838127e-03,9.185376e-04)
  (9.921660e-01,0.000000e+00)  (-8.681035e-03,-3.323868e-03)
  (-8.681035e-03,3.323868e-03)  (1.006634e+00,0.000000e+00)


  Inverse of average covariance matrix:
Columns 1 through 2
  (1.008024e+00,0.000000e+00)  (8.688107e-03,3.324494e-03)
  (8.688107e-03,-3.324494e-03)  (9.935789e-01,-2.576044e-09)
  (-3.545582e-03,1.096502e-03)  (6.415665e-03,-4.301259e-03)
  (4.019544e-03,3.300156e-03)  (4.840136e-03,8.899141e-04)
```

Figure 4-4: Estimating the Covariance Matrix and Amplitude

```
 Columns 3 through 4
  (-3.545597e-03,-1.096502e-03)   (4.019544e-03,-3.300156e-03)
  (6.415665e-03,4.301277e-03)   (4.840136e-03,-8.899141e-04)
  (1.008057e+00,4.493454e-09)   (8.709729e-03,3.317986e-03)
  (8.709729e-03,-3.317978e-03)   (9.935464e-01,-1.200157e-08)

Trial: 1   Amplitude Estimate: (1.5886e+00,1.2140e+00)
                  Magnitude: 1.9994e+00
display: Next trial, All trials, or Quit (n/a/q) [N] ? a
Trial: 2   Amplitude Estimate: (-1.0848e+00,-8.5688e-01)
                  Magnitude: 1.3824e+00
Trial: 3   Amplitude Estimate: (-5.6062e-01,-1.3987e+00)
                  Magnitude: 1.5069e+00
                     .
                     .
                     .
Trial: 9998   Amplitude Estimate: (-3.3731e-01,1.2878e+00)
                  Magnitude: 1.3312e+00
Trial: 9999   Amplitude Estimate: (1.0322e-01,-9.3869e-01)
                  Magnitude: 9.4435e-01
Trial: 10000   Amplitude Estimate: (1.8662e-01,8.7653e-01)
                  Magnitude: 8.9618e-01
Mean value for 10000 trials.
  Average amplitude estimate: (1.3166e-02,2.2880e-03)
                Variance: 1.8137e+00
        Average magnitude: 1.3018e+00          Actual magnitude was 1.25
                Variance: 1.1894e-01
ok?
```

Figure 4-5: Estimating the Covariance Matrix and Amplitude (Cont.)

4.1.2 Subtraction Filter Routine

Figure 4-6 shows the subtraction filter routine. The input consists of two files. The first file contains the radar returns from which the estimated signal is to be subtracted. The second file contains the estimated amplitudes produced by the amplitude estimating routine. There should be one amplitude estimate for each realization of the radar returns in the first file. Optionally, the user can specify a single amplitude to be used for all realizations. The output is an estimate of the clutter, formed by subtracting the estimated signal from the inputs. The output is suitable for use in a signal detection algorithm, or for diagnostics. For example, mean descriptive statistics over the 10,000 trials in the output from the subtraction filter can be calculated. The sample variance along the two channels is 1.027 and 1.001, respectively. The sample estimates can be compared with the theoretical value for these innovations, unity in this case.

4.1.3 Constant Signal Analysis Routines

Four new analysis sequences were written in the UFI-based system for the analysis of constant magnitude signals with unknown amplitudes. Figure 4-7 shows the UFI menu for choosing one of these analysis sequences. The first sequence is used to examine the performance of the covariance matrix and amplitude estimation algorithms. The other three sequences are used to analyze the signal detection algorithm illustrated in Figure 4-8.

```
Linking program msubfil, please stand by ...
f77 -O -C -pipe -cg89 sun4/msubfil.o -Lsun4 -lmc -lvec -lmat -lmc -L/home/marlin/tom/lib/sun4 -lfsl  -o
MSUBFIL -- Filters input by subtracting constant signal
Version 1.5
Input file name ? rlv1
Title: Signal + Noise, Amplitude 1.25, Doppler 0.0, Spacing 0.125 Angle 90o
Output file name ? rlvsub
Title of this dataset ? Estimated noise
Enter amplitude or read from File (ef) [F] ?
File containing amplitude estimates ? rlva
Title: Amplitude estimates
Enter the normalized Doppler (fd T) ? 0.0
Enter the normalized element spacing ? 0.125
Enter the angle to the signal in radians ? 1.570796
Filtered 10000 trials.
ok?
```

Figure 4-6: Routine for Subtracting the Estimated Signal

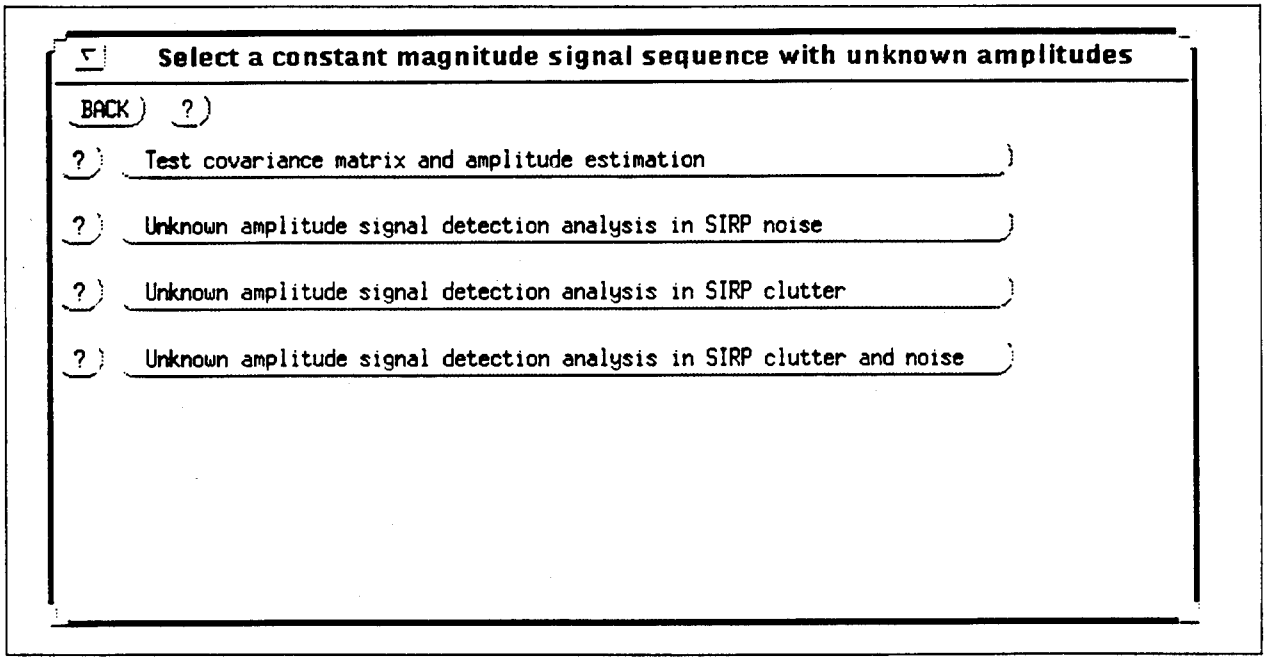

Figure 4-7: Unknown Amplitude Analysis Sequences

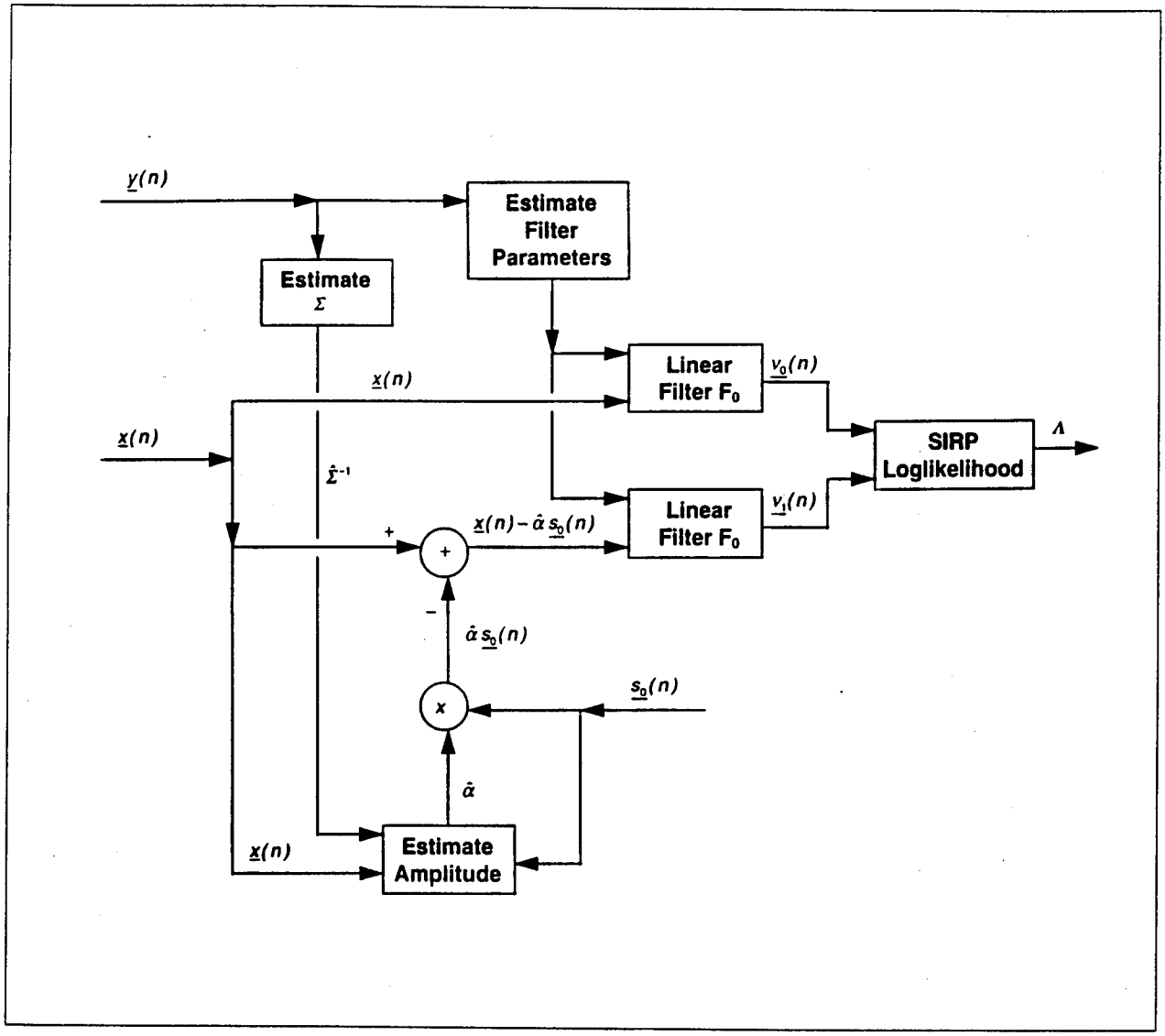

Figure 4-8: The Signal Detection Algorithm for a Constant Signal

The analysis sequence for analyzing the amplitude estimation algorithm is fairly straightforward. The user is asked to enter the number of trials used in synthesizing clutter and the number of trials used in synthesizing the sum of signal and clutter. Synthesized clutter is used in estimating the block covariance matrix. The sum of clutter and the signal is used to estimate the amplitude. The user enters AR process parameters for the clutter and parameters for the constant magnitude signal. The clutter is an AR process with Gaussian, K-distributed SIRP, or Weibull SIRP driving noise. The covariance matrix and the amplitudes are estimated as in the menu-based routines. Estimates of the covariance matrix and the amplitude are sent to the output file. Figures 4-9, 4-10, and 4-11 show a sample output of this analysis sequence.

```
Set random number seeds
Set random number generator seed : n
Get number of channels.
Number of channels  : 2
Get some common parameters for signal, clutter, and noise.
Length of the generated process in points  : 3
Enter the number of points to be generated before saving data : 500
Get AR process parameters with SIRP driver
Order of the AR process : 3
Gaussian, K-distributed SIRP, or Weibull SIRP driving noise (g/k/w) : k
Do you want S to vary within each realization/trial (y/n)  : y
Do you want the diagonal elements of S to all be equal (y/n)  : y
Specify Cholesky or L-D-U decomposition (c/l)  : c
Driving noise:  real only, imaginary only, or complex (r/i/c)  : c
Noise Alpha, the shape parameter for Gamma distribution of S  : 0.5
Noise B, a parameter for the distribution of V  : 1.0
Use Gaussian or Exponential shaped function (g/e) : g
Enter the amplitude matrix for the AR process : 1.0,0.0
0.0,1.0
Enter the intertemporal correlation matrix for the AR process : 0.1,0.0
0.0,0.1
Enter the matrix of lag values for the AR process : 0,0
0,0
Reference doppler frequency for the AR process : 0.0
Enter the sample interval for the AR process : 0.01
Intertemporal correlation coefficients:
R(0)=(1.0000e+00,0.0000e+00) (0.0000e+00,0.0000e+00)
     (0.0000e+00,0.0000e+00) (1.0000e+00,0.0000e+00)

R(1)=(1.0000e-01,0.0000e+00) (0.0000e+00,0.0000e+00)
     (0.0000e+00,0.0000e+00) (1.0000e-01,0.0000e+00)

R(2)=(1.0000e-04,0.0000e+00) (0.0000e+00,0.0000e+00)
     (0.0000e+00,0.0000e+00) (1.0000e-04,0.0000e+00)

R(3)=(1.0000e-09,0.0000e+00) (0.0000e+00,0.0000e+00)
     (0.0000e+00,0.0000e+00) (1.0000e-09,0.0000e+00)

Determinants of principle minors:
(1.0000e+00,0.0000e+00)(1.0000e+00,0.0000e+00)(9.9000e-01,0.0000e+00)(9.8010e-01,0.0000e+00)...
Coefficients for the AR process:
a(1)= (-1.0101e-01,0.0000e+00) (0.0000e+00,0.0000e+00)
      (0.0000e+00,0.0000e+00) (-1.0101e-01,0.0000e+00)

a(2)= (1.0101e-02,0.0000e+00) (0.0000e+00,0.0000e+00)
      (0.0000e+00,0.0000e+00) (1.0101e-02,0.0000e+00)

a(3)= (-1.0000e-03,0.0000e+00) (0.0000e+00,0.0000e+00)
      (0.0000e+00,0.0000e+00) (-1.0000e-03,0.0000e+00)
```

Figure 4-9: Amplitude Estimation Results

```
Covariances:  (9.8990e-01,0.0000e+00) (0.0000e+00,0.0000e+00)
              (0.0000e+00,0.0000e+00) (9.8990e-01,0.0000e+00)
Correlations: (9.9494e-01,0.0000e+00) (0.0000e+00,0.0000e+00)
              (0.0000e+00,0.0000e+00) (9.9494e-01,0.0000e+00)
CONST0P - Reads constant magnitude signal parameters from user
Amplitude of constant magnitude signal  : 1.25
Normalized Doppler of constant magnitude signal  : 0.0
Normalized element spacing  : 0.125
Angle to the signal in radians  : 1.5708
MCOVP - Asks the user to enter covariance matrix estimation options
Estimation: sample Matrix, time/space Average, time Clipping(m/a/c) : a
Use biased estimate (y/n) : y
Output each covariance matrix estimate (y/n)  : n
ACOV0P - Asks user for info about average covariance matrix estimate
Output the average covariance matrix estimate (y/n)  : y
MAMP0P - Ask user whether amplitude estimates should be output
Output each amplitude estimate (y/n)  : n
Is the amplitude magnitude known (y/n)  : n
Mean value for 10000 trials.
  Average covariance matrix:
Columns 1 through 2
  (9.739912e-01,0.000000e+00)   (2.712762e-03,5.375063e-03)
  (2.712762e-03,-5.375063e-03)  (9.901970e-01,0.000000e+00)
  (6.350853e-02,5.049611e-03)   (-2.736679e-03,-7.572205e-03)
  (-8.788312e-04,2.111209e-03)  (6.849832e-02,9.498819e-04)
  (-2.587584e-03,-7.809592e-06)  (-8.939112e-04,-1.153310e-03)
  (-8.176130e-04,2.022152e-04)  (-3.452280e-04,1.530773e-03)

Columns 3 through 4
  (6.350853e-02,-5.049611e-03)  (-8.788312e-04,-2.111209e-03)
  (-2.736679e-03,7.572205e-03)  (6.849832e-02,-9.498819e-04)
  (9.739912e-01,0.000000e+00)   (2.712762e-03,5.375063e-03)
  (2.712762e-03,-5.375063e-03)  (9.901970e-01,0.000000e+00)
  (6.350853e-02,5.049611e-03)   (-2.736679e-03,-7.572205e-03)
  (-8.788312e-04,2.111209e-03)  (6.849832e-02,9.498819e-04)

Columns 5 through 6
  (-2.587584e-03,7.809592e-06)  (-8.176130e-04,-2.022152e-04)
  (-8.939112e-04,1.153310e-03)  (-3.452280e-04,-1.530773e-03)
  (6.350853e-02,-5.049611e-03)  (-8.788312e-04,-2.111209e-03)
  (-2.736679e-03,7.572205e-03)  (6.849832e-02,-9.498819e-04)
  (9.739912e-01,0.000000e+00)   (2.712762e-03,5.375063e-03)
  (2.712762e-03,-5.375063e-03)  (9.901970e-01,0.000000e+00)
```

Figure 4-10: Amplitude Estimation Results (Cont'd)

```
 Variances:
Columns 1 through 4
 1.543065e+00  9.388222e-01  2.419760e-01  2.206363e-01
 9.388222e-01  1.547744e+00  2.341915e-01  2.381257e-01
 2.419760e-01  2.341915e-01  1.543065e+00  9.388222e-01
 2.206363e-01  2.381257e-01  9.388222e-01  1.547744e+00
 1.050504e-01  1.100442e-01  2.419760e-01  2.341915e-01
 1.063162e-01  1.076555e-01  2.206363e-01  2.381257e-01

Columns 5 through 6
 1.050504e-01  1.063162e-01
 1.100442e-01  1.076555e-01
 2.419760e-01  2.206363e-01
 2.341915e-01  2.381257e-01
 1.543065e+00  9.388222e-01
 9.388222e-01  1.547744e+00


 Inverse of average covariance matrix:
Columns 1 through 2
 (1.031222e+00,0.000000e+00)  (-3.135458e-03,-6.307036e-03)
 (-3.135458e-03,6.307013e-03)  (1.014910e+00,1.124075e-08)
 (-6.778821e-02,-5.359376e-03)  (3.187418e-03,8.721530e-03)
 (1.331642e-03,-3.049985e-03)  (-7.064067e-02,-8.784747e-04)
 (7.148132e-03,7.088222e-04)  (5.463213e-04,2.312660e-05)
 (6.700754e-04,1.828000e-04)  (5.255580e-03,-1.441172e-03)

Columns 3 through 4
 (-6.778820e-02,5.359375e-03)  (1.331642e-03,3.049985e-03)
 (3.187418e-03,-8.721530e-03)  (-7.064067e-02,8.784608e-04)
 (1.035732e+00,1.429922e-09)  (-3.431246e-03,-7.086769e-03)
 (-3.431261e-03,7.086784e-03)  (1.019806e+00,-8.951288e-09)
 (-6.778634e-02,-5.365778e-03)  (3.181972e-03,8.725807e-03)
 (1.310989e-03,-3.038183e-03)  (-7.064237e-02,-8.721282e-04)

Columns 5 through 6
 (7.148176e-03,-7.088184e-04)  (6.700791e-04,-1.828074e-04)
 (5.463175e-04,-2.309680e-05)  (5.255580e-03,1.441169e-03)
 (-6.778634e-02,5.365772e-03)  (1.310974e-03,3.038183e-03)
 (3.181983e-03,-8.725822e-03)  (-7.064235e-02,8.721200e-04)
 (1.031291e+00,-9.782253e-10)  (-3.096633e-03,-6.345093e-03)
 (-3.096640e-03,6.345086e-03)  (1.014843e+00,-8.683358e-10)


Mean value for 10000 trials.
 Average amplitude: (-2.0039e-02,8.3590e-03)  Variance: 1.7575e+00
 Average magnitude: 1.2911e+00  Variance: 9.1017e-02
```
Actual magnitude 1.25

Figure 4-11: Amplitude Estimation Results (Cont'd)

The remaining sequences estimate the threshold and the probability of detection for user-specified false alarm probabilities. Variances in these estimates are calculated. In all three cases discussed here, the signal is a constant magnitude signal, as described in Section 3.1.2. The three sequences vary among the null hypotheses in which no signal is present. In one case, the clutter is modeled as K-distributed SIRP white noise uncorrelated in time. In this case, the filter F_0 merely decorrelates the input across channels, as in Figure 3-1. In the second case, the clutter is modeled as an AR process with K-distributed SIRP driving noise. In the third case, the clutter is modeled as the sum of an AR process and white noise. In both of these cases, the filter F_0 decorrelates the input across time and across channels, as in Figure 3-2.

These sequences begin with an a priori estimation phase. Many realizations of the clutter $y(n)$ are used to estimate the covariance matrix Σ and the filter parameters. When the clutter is modeled as white noise, the filter has only one parameter, related to the JxJ zero-lag covariance matrix. Otherwise, AR coefficients and the driving noise covariance matrix are estimated.

The next step is to synthesize many realizations of the radar returns $x(n)$ under the null hypothesis of no signal. These realizations are used to determine the loglikelihood statistic Λ for given false alarm probabilities. The statistic is calculated for each realization, as shown in Figure 4-8. The null hypothesis innovations $v_0(n)$ are estimated by the appropriate linear filter, Figure 3-1 or Figure 3-2. The signal amplitude is estimated for each realization based on the 'a priori' estimate of the covariance matrix and the known steering vector. The user has the option of specifying the amplitude magnitude as known, in which case only the phase is estimated. The amplitude estimate is used to subtract the estimated signal from the radar returns, and the result is then filtered by F_0 to produce the alternative hypothesis innovations $v_1(n)$. These innovations are used to calculate the appropriate loglikelihood statistic for a K-distributed SIRP. An estimate of the probability distribution for Λ under the null hypothesis is found from the many realizations of the loglikelihood distribution. This allows the program to calculate the statistic value (threshold) for the tail probability given by the false alarm probability.

Another set of realizations of the radar returns are used to estimate the probability of detection. This set of realizations contains the constant magnitude signal, as well as clutter. The signal is resynthesized for each realization, allowing the signal phase to vary randomly across realizations. The same signal detection algorithm (Figure 4-8) is used to calculate the loglikelihood statistic for each realization. If it is above the threshold determined before by an user-specified false alarm probability, the algorithm decides that a signal is present. The detection probability is estimated as the ratio of the number of realizations in which a signal is decided to be present to the total number of realizations.

The user can specify that this process of a priori estimation, threshold determination, and probability of detection estimation be repeated a number of times. The different results of these repetitions are used to estimate the variance in the estimates of the threshold and the probability of detection.

4.2 Weibull Clutter

A number of subroutines were modified, both in the menu-based subsystem and in sequences in the UFI-based subsystem, to implement the synthesis of Weibull-distributed Spherically Invariance Random Processes (SIRPs). Figure 4-12 shows an invocation of this

```
                MULTI-CHANNEL (M/C) PROCESS SYNTHESIS MENU
                .
                .
                .
M2  -- Method 2 (MC2) process synthesis
                .
                .
                .
Enter a command or Q to return to the MAIN menu: m2
Linking program msirp, please stand by ...
'bin/sun4/msirp' is up to date.
MSIRP -- Multichannel AR Process Generation with SIRPs
Version 1.14
Do you want to set the pseudo random generator seeds (y/n) [N] ? y
Enter an integer seed value ? 4
Enter an integer seed value ? 6
Enter an integer seed value ? 8
Enter an integer seed value ? 1
Number of channels ? 1
Number of points to be generated and saved ? 100
Enter the number of trials to be generated ? 10
Add a signal component (y/n) [Y] ? n
Add a clutter component (y/n) [Y] ? n
Add a white noise component (y/n) [Y] ? y
ENTER PARAMETERS FOR THE NOISE.
Gaussian, K SIRP, or Weibull SIRP noise (g/k/w) [G] ? w
Do you want R to vary within each realization/trial (y/n) [Y] ? y
Do you want R to be constant across channels (y/n) [Y] ? y
Specify Cholesky, L-D-U, or SVD (c/l/s) [C] ? c
Do you want the noise to be real, imaginary, or complex (r/i/c) [C] ? c
Enter Noise B, a parameter for the Weibull distribution ? 1.0
Do you want the Cholesky decomposition matrix C to be identity (y/n) [Y] ? y
Complex Weibull-distributed SIRP noise will be generated.
SIRP random variable R varies across time samples.
R will be constant across channels.
B for the SIRP: 1.0000e+00
Correlations:  (1.0000e+00,0.0000e+00)
Are these parameters okay (y/n) [Y] ? y
Total output file name ? rlv
Title of this dataset ? White noise, magnitude is Weibull
Noise output file name ?
No file name entered, is this okay ? y
 Note: the following IEEE floating-point arithmetic exceptions
 occurred and were never cleared; see ieee_flags(3M):
 Inexact;  Underflow;  Overflow;  Invalid Operand;
 Sun's implementation of IEEE arithmetic is discussed in
 the Numerical Computation Guide.
ok?                    Warning message is common
```

Figure 4-12: Synthesizing Weibull Noise

capability. In this case, temporally uncorrelated white noise is generated. Each time sample is a complex one-channel SIRP with another realization of the spherical distance from the origin R. From Equations 3.2-23 and 3.2-11 through 3.2-14, one can show the distribution of the magnitude of the process is the Weibull distribution given by Equation 4.2-1:

$$f_R(r) = a\, b\, r^{b-1} e^{-a r^b}, \quad r > 0. \tag{4.2-1}$$

Since the user specifies b to be equal to unity in the example, both the scale and shape parameters of this Weibull distribution are unity. (See Appendix B.)

The synthesized process can be analyzed by any capabilities provided in the MSPSS. For example, Figure 4-13 shows the results of applying Ozturk's algorithm (Ozturk 90) to the magnitude of the first realization of the processes synthesized in Figure 4-12. The hypothesis that the magnitude is Weibull-distributed, with a shape parameter of unity, should be rejected if the endpoint of the curve corresponding to the sample falls outside the double circles. In this case, Ozturk's statistic leads to the acceptance of a Weibull distribution. So this case illustrates a test showing the Weibull-synthesis capability functions correctly.

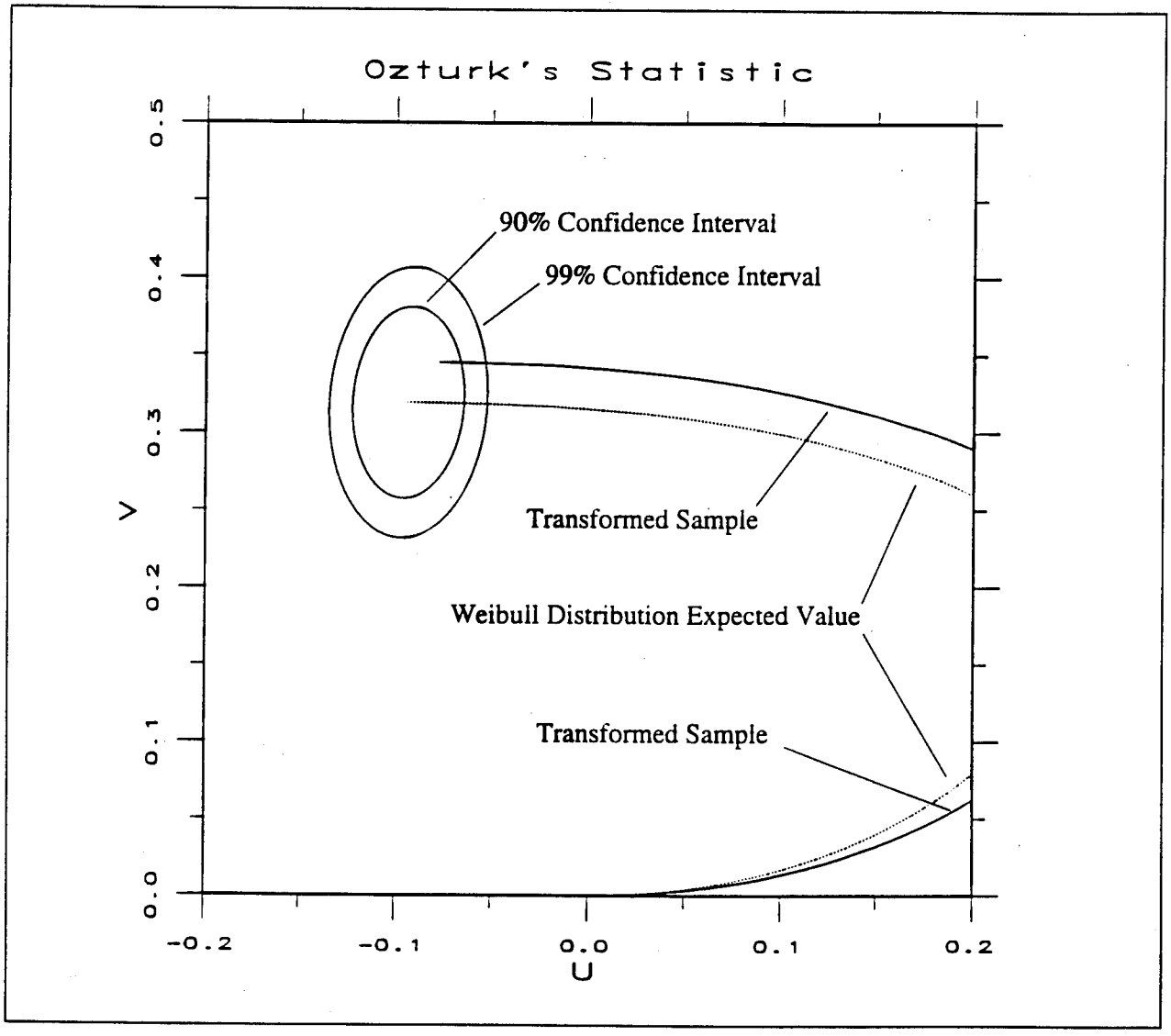

Figure 4-13: Ozturk Test for Weibull Distribution

Some difficulties arose in implementing the Weibull distribution. Appropriate parameters for the triangular distributions described in Appendix C were found numerically. The synthesis process can be quite slow for Weibull SIRPs with a large number of time samples. Procedures to remove these limitations should be investigated.

4.3 Representative Model

The Representative Model was implemented as one routine in the menu-based subsystem. Figures 4-14 through 4-17 illustrate the use of this routine. The user enters parameters describing the signal, the clutter, and interference. Two models are provided for both the signal and the interference. The synthesized process need not contain signal, clutter, and interference. Any of the three can be absent. The user is able to save each to a separate file, as well as any AR parameters that have been calculated for the respective models. Only the sum of signal, clutter, and interference is saved in the illustrated example. Figure 4-18 shows a graph of the real part of the synthesized process. In this case, the synthesized process consists of two periods of three channels of a phase-shifted sine wave (for the signal), with the addition of clutter and interference.

```
STAP -- Synthesizes multichannel representative model processes
Version 1.13
Do you want to set the pseudo random generator seeds (y/n) [N] ?
Number of channels ? 3                    Describe radar.
Number of points to be generated and saved ? 100
Enter the number of trials to be generated ? 100
Enter the normalized element spacing ? 0.25
Enter the normalized platform Doppler center frequency ? 0.014142136

Channels/elements: 3
Time samples: 100
Realizations/trials: 100

Radar parameters:
  Normalized element spacing: 2.50000e-01
  Normalized element spacing: 1.41421e-02
Are these parameters okay (y/n) [Y] ?
Add a signal component (y/n) [Y] ?                    Describe signal.
ENTER PARAMETERS FOR THE SIGNAL.
Do you want a Constant amplitude or Random signal (c/r) [C] ? c
Enter the normalized signal Doppler center frequency ? 0.014142136
Enter the angular direction to signal ? 0.785398163
Enter the magnitude of the signal ? 1.5
Do you want to specify the initial phase (y/n) [Y] ?
Enter the initial phase ? 0.0
Constant amplitude signal parameters:
  Normalized signal Doppler center frequency: 1.41421e-02
  Angular direction to the signal: 7.85398e-01
  Magnitude: 1.50000e+00
  Initial phase: 0.00000e+00
Are these parameters okay (y/n) [Y] ?
```

Figure 4-14: Representative Model Synthesis

```
Add a clutter component (y/n) [Y] ?                    Describe clutter.
ENTER PARAMETERS FOR THE CLUTTER.
Enter the angular beam direction ? 0.0
Do you want the clutter to be real, imaginary, or complex (r/i/c) [C] ?
Gaussian, K-distributed SIRP, or Weibull SIRP driving noise (g/k/w) [G] ?
Constant clutter standard deviation across channels (y/n) [Y] ?
Enter the channel standard deviation ? 0.5
Do you want to use the Block procedure or an AR process (b/a) [A] ? a
Enter the number of points to be generated before saving data ? 1000
Enter the order of the AR process ? 4
Specify Cholesky, L-D-U, or SVD decomposition (c/l/s) [C] ? c
Gaussian, exponential Correlation, or exponential Power temporal shaping [G] ? g
Enter the one lag temporal correlation parameter ? 0.90
Gaussian or Hanning/hamming/blackman-harris spatial shaping function (g/h) [G] ? g
Enter the one lag spatial correlation parameter ? 0.9
Clutter parameters:
  Angular beam direction: 0.00000e+00
  Driving noise is complex.
  Cholesky decomposition is used in synthesizing driving noise.
  Clutter has Gaussian driving noise.
  Channel standard deviations:
    5.00000e-01 5.00000e-01 5.00000e-01
  Clutter will be synthesized as an AR process.
  Points to prime the clutter process: 1000
  Order of the AR process: 4
  A Gaussian temporal shaping function will be used.
  One lag temporal correlation parameter: 9.00000e-01
  A Gaussian spatial shaping function will be used.
  One lag spatial correlation parameter: 9.00000e-01
Are these parameters okay (y/n) [Y] ? y

Block covariance matrix for lag 0
  2.500000e-01  2.250000e-01  1.640250e-01
  2.250000e-01  2.500000e-01  2.250000e-01
  1.640250e-01  2.250000e-01  2.500000e-01
        .                          The program calculates the
        .                          block covariance matrix.
        .
Block covariance matrix for lag 4
  4.632548e-02  4.169293e-02  3.039415e-02
  4.169293e-02  4.632548e-02  4.169293e-02
  3.039415e-02  4.169293e-02  4.632548e-02
```

Figure 4-15: Representative Model Synthesis (Continued)

Coefficients for the AR process:

A(1):

-2.698014e+00 -1.227813e-03 1.968449e-05
-2.500378e-03 -2.694599e+00 -2.502176e-03
4.966523e-04 -2.732538e-03 -2.697129e+00

A(2):

3.308810e+00 2.801777e-03 5.090439e-05
6.143676e-03 3.300308e+00 6.123936e-03
-1.290442e-03 6.827475e-03 3.306435e+00

A(3):

-2.186008e+00 -2.527904e-03 -1.548294e-04
-6.042001e-03 -2.177529e+00 -5.998279e-03
1.349216e-03 -6.845814e-03 -2.183456e+00

A(4):

6.564693e-01 9.012222e-04 1.044273e-04
2.379417e-03 6.530809e-01 2.352715e-03
-5.674362e-04 2.756119e-03 6.553717e-01

Driving noise covariance matrix:

4.342771e-03 3.905334e-03 2.844440e-03
3.907508e-03 4.342770e-03 3.907157e-03
2.840573e-03 3.901698e-03 4.340220e-03

Decomposed driving noise covariance matrix:

6.589971e-02 0.000000e+00 0.000000e+00
5.929477e-02 2.875588e-02 0.000000e+00
4.310448e-02 4.680183e-02 1.708250e-02

The program calculates the AR coefficients.

Figure 4-16: Representative Model Synthesis (Continued)

```
Add interference (y/n) [Y] ? y                    Describe interference.
ENTER PARAMETERS FOR THE INTERFERENCE.
Direct path White noise or partially Correlated noise (w/c) [W] ? w
Enter the normalized interference Doppler center frequency ? -0.014142136
Enter the angular direction to jammer ? 0.523598776
Do you want the interference to be real, imaginary, or complex (r/i/c) [C] ? c
Gaussian, K-distributed SIRP, or Weibull SIRP driving noise (g/k/w) [G] ? g
Enter the shaping function standard deviation ? 0.707106781
Direct path white noise interference parameters:
  Normalized interference Doppler center frequency: -1.41421e-02
  Angular direction to the jammer: 5.23599e-01
  Driving noise is complex.
  Cholesky decomposition is used in synthesizing driving noise.
  Interference has Gaussian driving noise.
  Standard deviation:
    7.07107e-01
Are these parameters okay (y/n) [Y] ? y
Total output file name ? rlv                      Save synthesized results.
Title of this dataset ? Rep. Model signal + clutter + interference
Signal output file name ?
No file name entered, is this okay ? y
Clutter output file name ?
No file name entered, is this okay ? y
Clutter output coefficient file name ?
No file name entered, is this okay ? y
Interference output file name ?
No file name entered, is this okay ? y
ok?
```

Figure 4-17: Representative Model Synthesis (Continued)

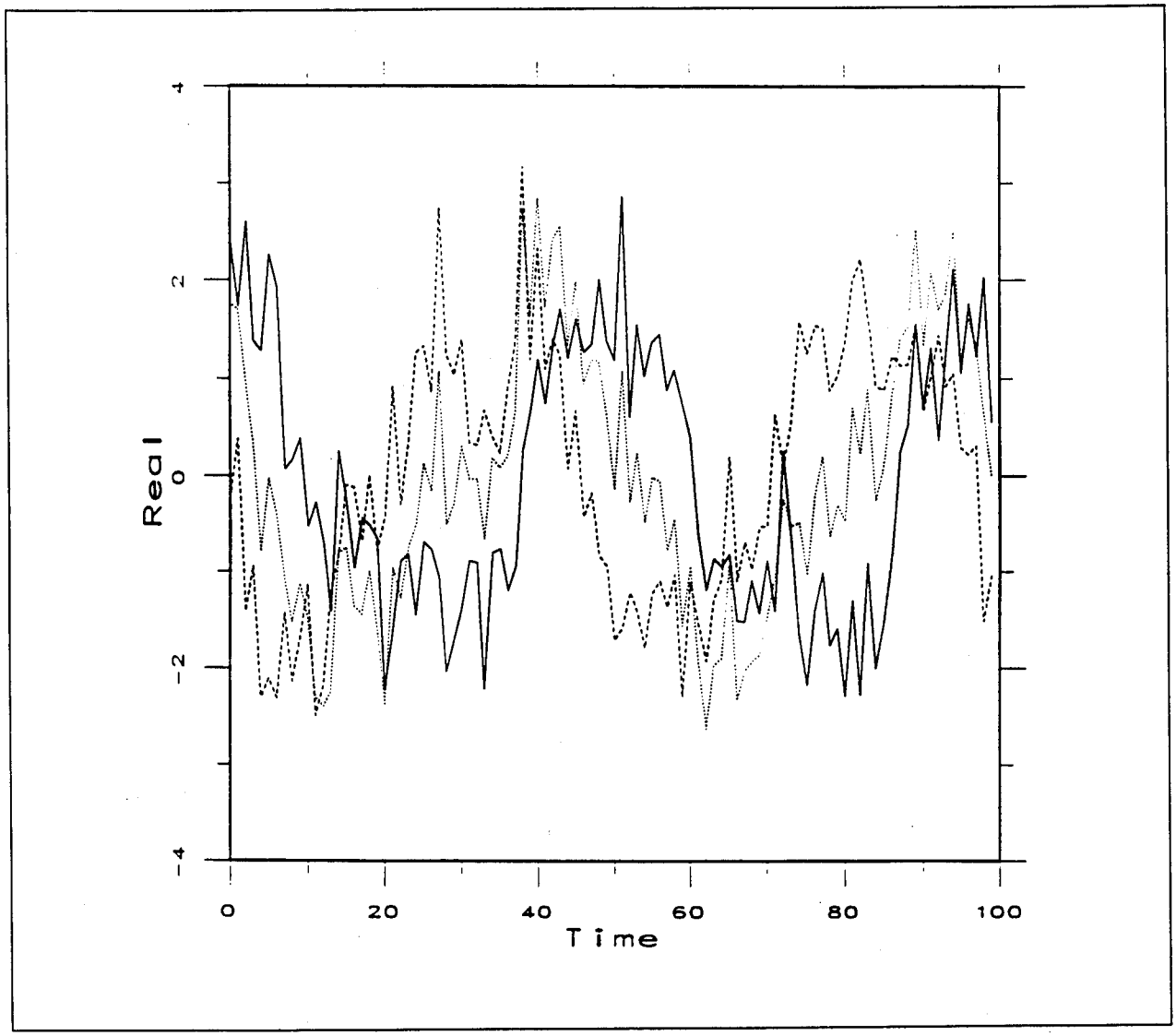

Figure 4-18: Real Part of Sum For the Three Channels

4.4 Two-Dimensional FFT

A two-dimensional FFT of the covariance matrix for the Representative Model clutter was implemented as one routine in the menu-based subsystem. The 2-D FFT algorithm was based on that implemented under Khoros. A 2-D FFT is implemented in Khoros as a combination of C and Fortran source code. The C source code was translated to Fortran under this effort. Figures 4-19 through 4-23 show an invocation of the two-dimensional FFT capability. The user enters parameters describing parameters of the correlation function for the Representative Model clutter. The 2-D FFT must be run under X Windows. A new window appears at various points in the program for displaying graphs. These displays use the Khoros program Xprism. Xprism allows the user to manipulate, rotate, and annotate the results.

```
Performs 2-D FFT on Representative Model Covariance Matrix.
Version 1.14
Number of channels ? 12
Number of time samples ? 12
Enter the normalized element spacing ? 0.5
Enter the normalized platform Doppler center frequency ? 0.5
Enter the angular beam direction ? 0.0
Enter the channel standard deviation ? 2.0
Gaussian, exponential Correlation, or exponential Power temporal shaping [G] ?
Enter the one lag temporal correlation parameter ? 0.9975
Gaussian or Hanning/hamming/blackman-harris spatial shaping function (g/h) [G] ?
Enter the one lag spatial correlation parameter ? 0.54
Do you want to do Dolph-Chebychev filtering (y/n) [Y] ? y
Sidelobe level (in decibels relative to peak) ? 40.0

Channels/elements: 12
Time samples: 12
Normalized element spacing: 5.0000000e-01
Normalized platform Doppler: 5.0000000e-01
Angular beam direction: 0.0000000e+00
Channel standard deviations:  2.0000000e+00
A Gaussian temporal shaping function will be used.
One lag temporal correlation parameter: 9.9750000e-01
A Gaussian spatial shaping function will be used.
One lag spatial correlation parameter: 5.4000002e-01
Dolph-Chebychev filtering will be performed.
Sidelobe level: 4.0000000e+01
Are these parameters okay (y/n) [Y] ?
Plotting covariance matrix...                    Displays graph in Figure 4-20
Complex data conversion: Real, Imaginary, Magnitude, or Angle [R] ?
Plotting covariance matrix...                    Displays graph in Figure 4-21
Complex data conversion: Real, Imaginary, Magnitude, or Angle [R] ? r
Plotting FFT of covariance matrix...             Displays graph in Figure 4-22
Complex data conversion: Real, Imaginary, Magnitude, or Angle [R] ? m
Plotting FFT of covariance matrix...             Displays graph in Figure 4-23
Complex data conversion: Real, Imaginary, Magnitude, or Angle [R] ?
ok?
```

Figure 4-19: An FFT Example

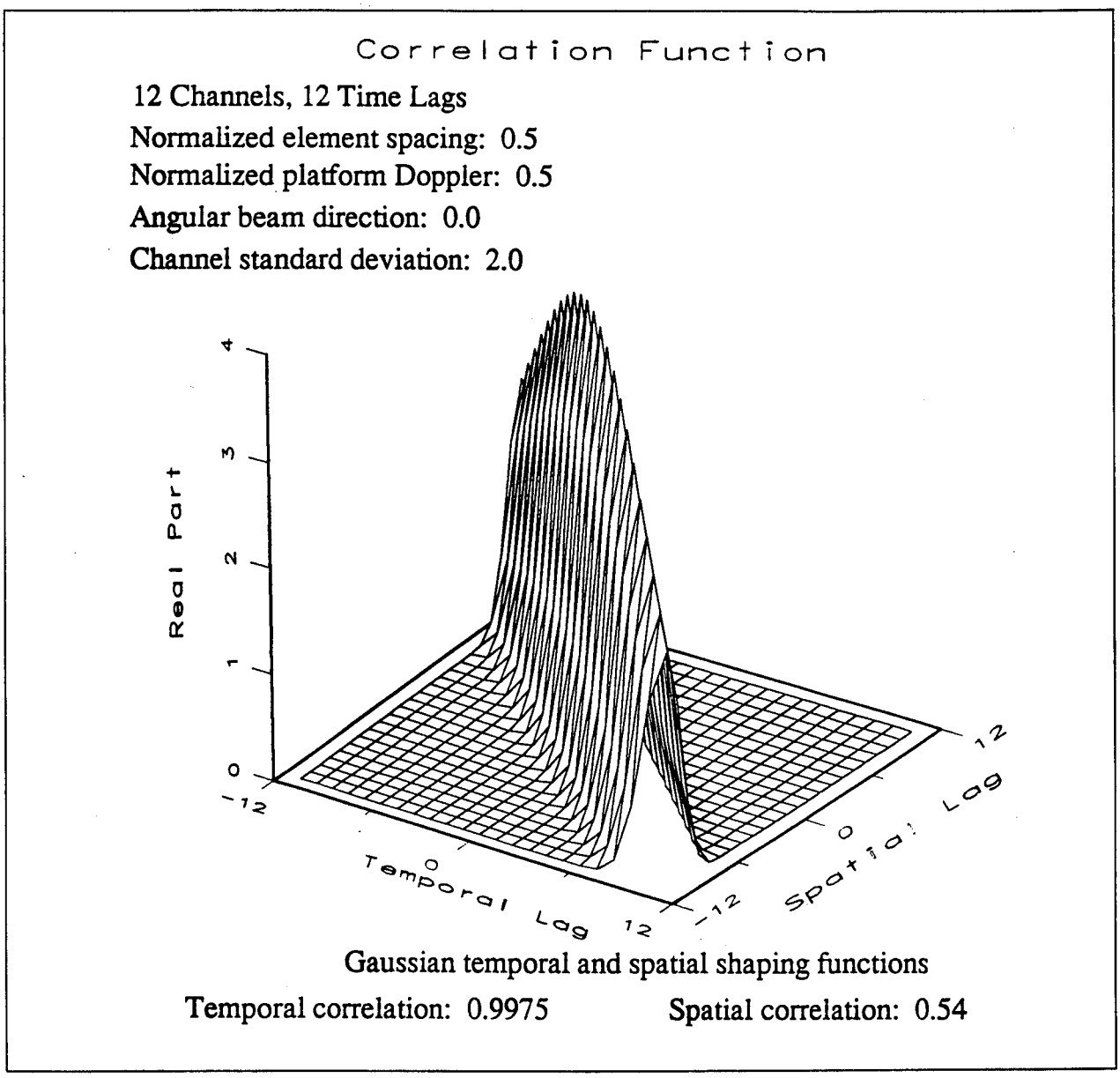

Figure 4-20: Graph of Correlation Function

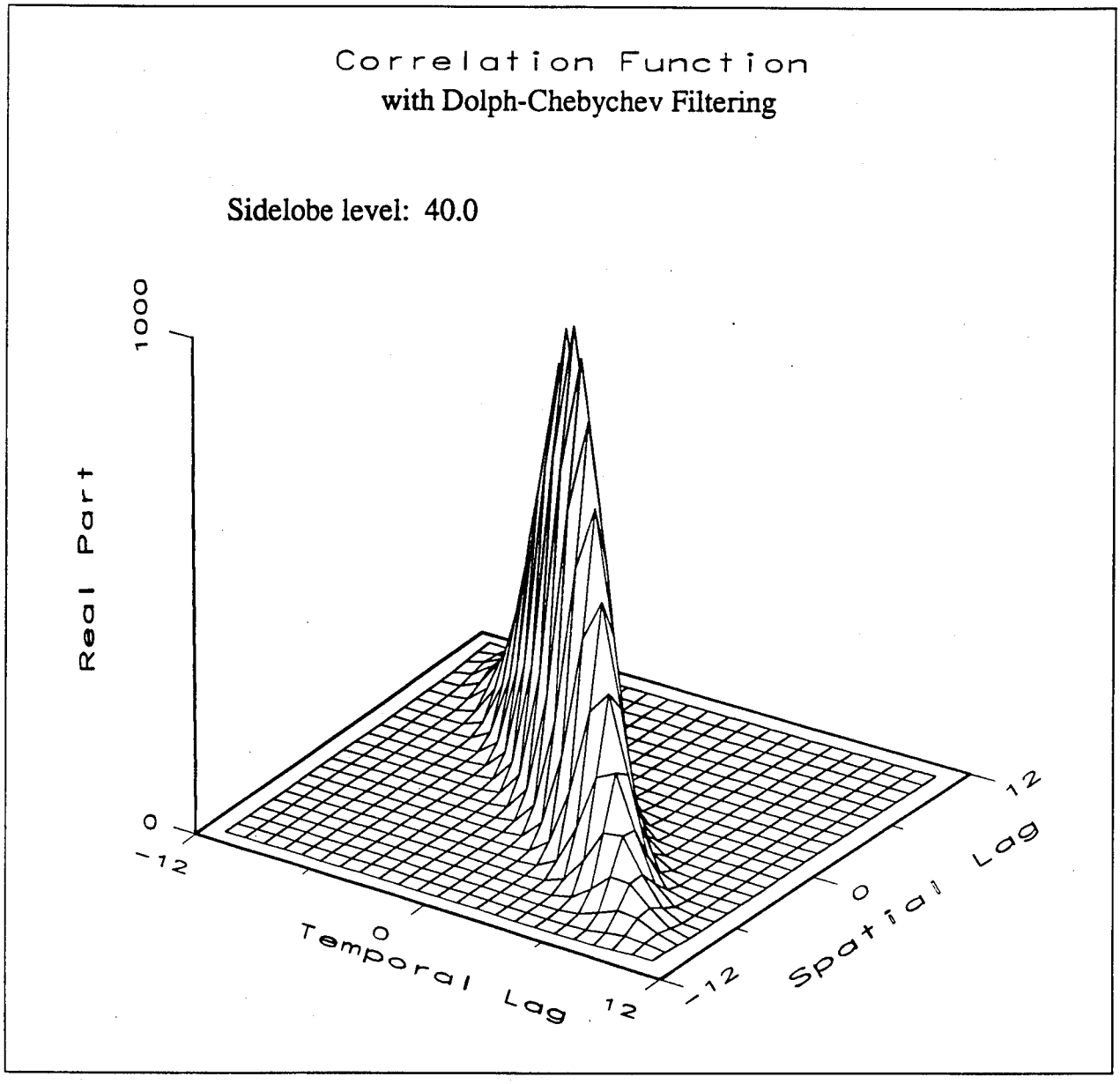

Figure 4-21: Dolph-Chebychev Filtering of Correlation Function

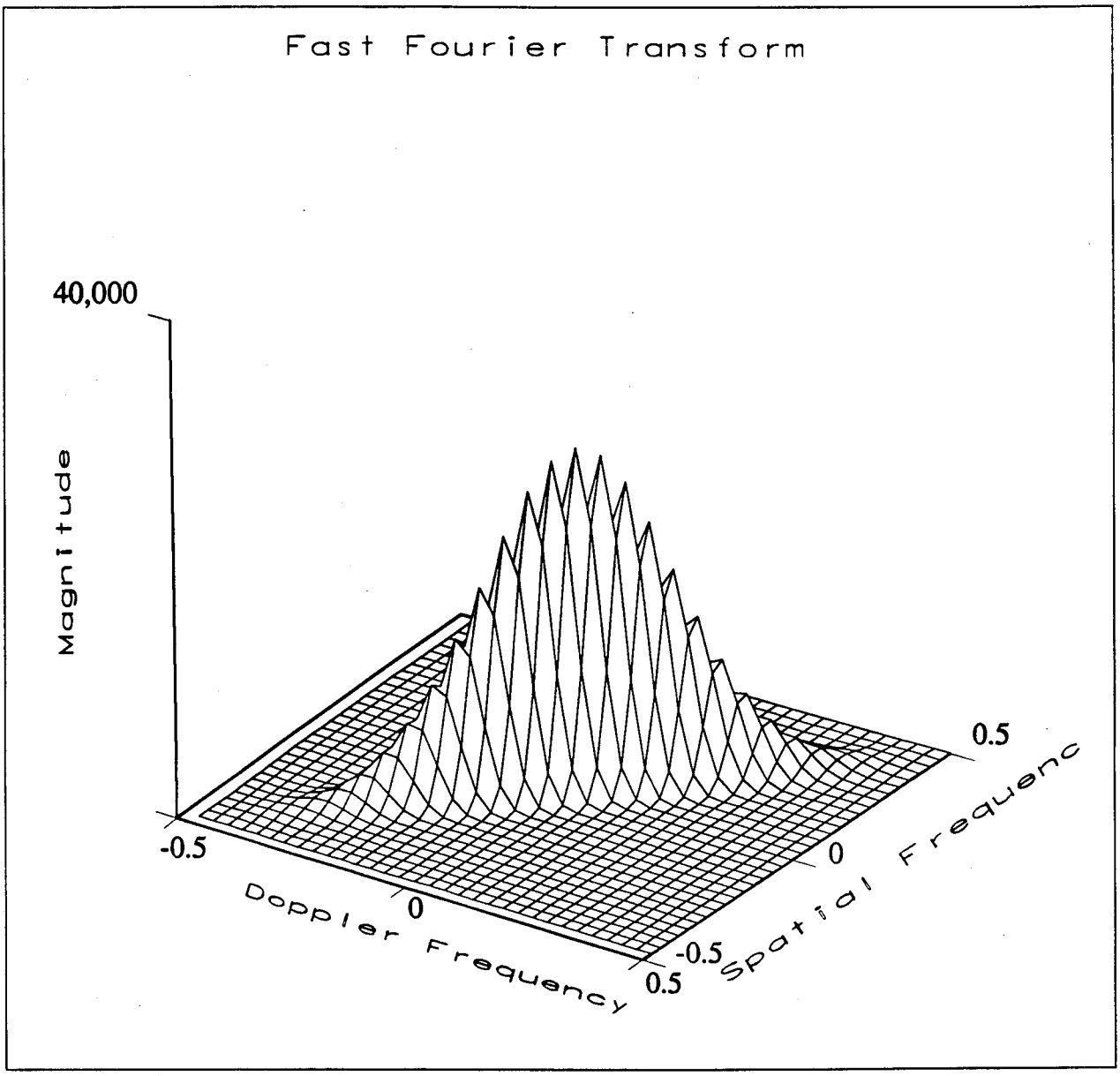

Figure 4-22: Graph of FFT

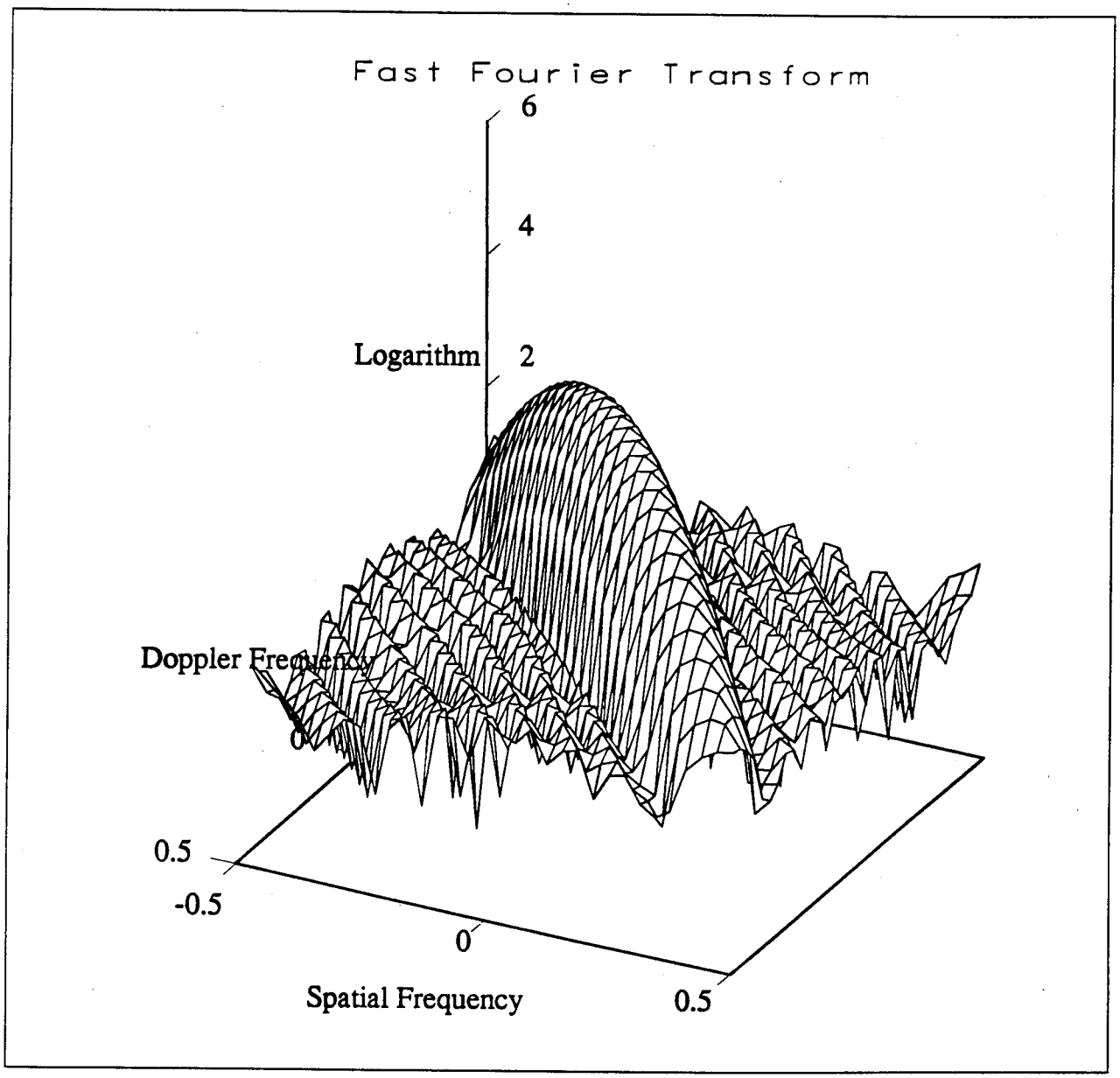

Figure 4-23: Logarithm of Magnitude of FFT

4.5 State Space Capabilities

Figures 4-24, 4-25, and 4-26 illustrate the use of the new state space routine added under this effort. The user enters parameters for a "shaping function" defining the block covariance matrices for lags equal to the user-selected order of an AR process. The Levinson-Wiggins-Robinson algorithm is used to solve the Yule-Walker equations to find the AR parameters. Block covariance matrices for a larger number of lags are found through either the State Space Closed Form Method (Section 3.5.1) or the AR Recursion Method (Section 3.5.2). This block covariance matrix is used to find state space parameters by the canonical correlations algorithm. (Previously, the block covariance matrix needed to be estimated from a synthesized process, thus introducing statistical variation.) As the example illustrates, the resulting state space estimates can exhibit a basis transformation.

```
SSC2 -- Estimates state space model parameters by Scientific Studies algorithm
Version 1.8
Number of channels ? 2
Order of the AR process ? 2
Use Gaussian or Exponential shaped function (g/e) [G] ?
Enter row 1 of the amplitude matrix:
Enter a row ? 1.995, 0.0
Enter row 2 of the amplitude matrix:
Enter a row ? 0.0, 1.995
Enter row 1 of the intertemporal correlation matrix:
Enter a row ? 0.9, 0.0
Enter row 2 of the intertemporal correlation matrix:
Enter a row ? 0.0, 0.9
Enter row 1 of the lag matrix:
Enter a row ? 0,0
Enter row 2 of the lag matrix:
Enter a row ? 0,0
Reference doppler frequency ? 0.0
Enter the sample interval ? 0.01
An AR( 2 ) process will be analyzed.
Gaussian shaped function.
Amplitude matrix:
  1.995000e+00   0.000000e+00
  0.000000e+00   1.995000e+00

1-lag Temporal Correlation Matrix:
  9.000000e-01   0.000000e+00
  0.000000e+00   9.000000e-01

Lag matrix:
  0  0
  0  0

Reference doppler frequency:  0.0000e+00 Hz.
Sample interval:  1.0000e-02 seconds.
Are these parameters okay (y/n) [Y] ?
Intertemporal correlation coefficients:
```

Figure 4-24: A State Space Example

```
R( 0 ):
   1.995000e+00   0.000000e+00
   0.000000e+00   1.995000e+00

R( 1 ):
   1.795500e+00   0.000000e+00
   0.000000e+00   1.795500e+00

R( 2 ):
   1.308919e+00   0.000000e+00
   0.000000e+00   1.308919e+00

Coefficients for the AR process:
A( 1 ):
  -1.629000e+00   0.000000e+00
   0.000000e+00  -1.629000e+00

A( 2 ):
   8.099999e-01   0.000000e+00
   0.000000e+00   8.099999e-01

Driving noise covariance matrix:
   1.303554e-01   0.000000e+00
   0.000000e+00   1.303554e-01

Output file name for parameter estimates ?
No file name entered, is this okay ? y
Upper bound, L, on size of state space block vector ? 3
State space closed Form or AR recursion method (f/a) ? a
Covariance matrix for lag 0
   1.995000e+00   0.000000e+00
   0.000000e+00   1.995000e+00

Covariance matrix for lag 1
   1.795500e+00   0.000000e+00
   0.000000e+00   1.795500e+00

Covariance matrix for lag 2
   1.308919e+00   0.000000e+00
   0.000000e+00   1.308919e+00

Covariance matrix for lag 3
   6.778746e-01   0.000000e+00
   0.000000e+00   6.778746e-01

Covariance matrix for lag 4
   4.403305e-02   0.000000e+00
   0.000000e+00   4.403305e-02
```

The program calculates AR parameters

Figure 4-25: A State Space Example (Cont'd)

```
Covariance matrix for lag 5
  -4.773485e-01   0.000000e+00
   0.000000e+00  -4.773485e-01

Covariance matrix for lag 6
  -8.132674e-01   0.000000e+00
   0.000000e+00  -8.132674e-01

MODEL ORDER MENU
0 - Enter the model order directly
1 - Use the canonical correlations to calculate the model order
2 - Use the normalized running sum of the canonical correlations
3 - Use the squares of the canonical correlations
4 - Use the normalized running sum of squares
5 - Use the log parameters
6 - Use the normalized mutual information parameters
Model order calculation method ? 1                 The program can find the model order
Upper bound on canonical correlations ? 0.05
Display diagnostic information (y/n) ? n           Intermediate calculations could have been displayed
Canonical correlations:
9.794061e-01 9.794056e-01 6.698955e-01 6.698954e-01 1.653733e-07 1.066407e-07
Order: 4
System Dynamics matrix F:
  8.740377e-01   3.144643e-15   3.874717e-01  -3.395807e-09
  9.068543e-16   8.740378e-01  -3.395813e-09  -3.874718e-01
 -3.874723e-01   3.395806e-09   7.549624e-01   9.210561e-15
  3.395801e-09   3.874724e-01  -1.060677e-14   7.549620e-01


Hermetian transpose of Observation matrix H:
  1.363761e+00  -9.999875e-09   2.536587e-01   2.253127e-09
  9.999872e-09   1.363760e+00   2.253115e-09  -2.536590e-01


Kalman Gain:
  1.001023e+00   2.130526e-09
 -2.130539e-09   1.001017e+00
  1.040161e+00  -3.170894e-08
 -3.170937e-08  -1.040140e+00


Covariance matrix for innovations:                 Matches theoretical value
  1.303552e-01  -3.080869e-15
 -3.080869e-15   1.303568e-01


ok?
```

Figure 4-26: A State Space Example (Cont'd)

5. REFERENCES

(Burden 81) Richard L. Burden, J. Douglas Faires, Albert C. Reynolds, *Numerical Analysis*, Second Edition; Prindle, Weber & Schmidt; 1981.

(Chakravarthi 92) Prakosh R. Chakravarthi, "On Determining the Radar Threshold for Non-Gaussian Processes from Experimental Data," Ph. D. Thesis, Syracuse University, 1992.

(Huang 88) X. Huang et. al., "A Method for Upgrading the Estimates of the Model Parameters of a Signal When Given a Truncated Measurement," *Fourth Annual ASSP Workshop on Spectrum Estimation and Modeling*, Minneapolis, MN, 3-5 August 1988.

(Hogg 78) Robert V. Hogg and Allen T. Craig, *Introduction to Mathematical Statistics*, Fourth Edition, Macmillan, 1978.

(Kaman 91a) Kaman Sciences Corporation, *Man Machine Interface Experiment*, January 23, 1991.

(Kaman 91b) Kaman Sciences Corporation, *User Front-End Interface Knowledge Base (UFIKB) Builders' Guide*, February 1991.

(Kaman 92a) Kaman Sciences Corporation, *Software User's Manual for the Multichannel Signal Processing Simulation System*, May 30, 1992.

(Kaman 92b) Kaman Sciences Corporation, *Enhancement of Multichannel Signal Processing Simulation System*, December 3, 1992.

(Marple 87) S. Lawrence Marple, Jr., *Digital Spectral Analysis*, Prentice-Hall, Inc., 1987.

(Michels 89) James H. Michels, *A Parametric Detection Approach Using Multichannel Processes*, Rome Air Development Center, RADC-TR-89-306, November 1989.

(Michels 90a) James H. Michels, *Synthesis of Multichannel Autoregressive Random Processes and Ergodicity Considerations*, Rome Air Development Center, RADC-TR-90-211, July 1990.

(Michels 90b) James H. Michels, *Multichannel Linear Prediction and Its Association with Triangular Matrix Decomposition*, Rome Air Development Center, RADC-TR-90-226, November 1990.

(Michels 91) James H. Michels, *Multichannel Detection Using the Discrete-Time Model-Based Innovations Approach*, Rome Laboratory, RL-TR-91-269, August 1991.

(Michels 92a) James H. Michels, *Multichannel Detection of Partially Correlated Signals in Clutter*, Rome Laboratory, RL-TR-92-332, December 1992.

(Michels 92b) James H. Michels, *Considerations of the Error Variances of Time-Averaged Estimators for Correlated Processes*, Rome Laboratory, RL-TR-92-339, December 1992.

(Michels 94) James H. Michels, Pramod K. Varshney, and Donald D. Weiner, "Synthesis of

Correlated Multichannel Random Processes," *IEEE Transactions on Signal Processing*, Vol. 42, No. 2, February 1994, pp. 367-375.

(Michels 95) James H. Michels, Pramod K. Varshney, and Donald D. Weiner, "Multichannel Signal Detection Involving Temporal and Cross-Channel Correlation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 31, No. 3, July 1995, pp. 866-879.

(Ozturk 90) A. Ozturk and E. J. Dudewicz, *A New Statistical Goodness-of-Fit Test Based on Graphical Representation*, Technical Report no. 53, Department of Mathematics, Syracuse University, 1990.

(Rangaswamy 91) M. Rangaswamy, D. D. Weiner, and A. Ozturk, "Simulation of Correlated Non-Gaussian Interference for Radar Signal Detection," *Proceedings of Twentyfifth Asilomar Conference on Signals and Computers*, Pacific Grove, CA, 1991.

(Rangaswamy 92) M. Rangaswamy, D. D. Weiner, and J. H. Michels, "Innovations Based Detection Algorithm for Correlated Non-Gaussian Random Process," , 1992.

(Rangaswamy 93a) Muralidhar Rangaswamy, Donald Weiner, and Aydin Ozturk, "Computer Generation of Correlated Non-Gaussian Radar Clutter," unpublished manuscript, April 8, 1993.

(Rangaswamy 93b) M. Rangaswamy et. al., *Signal Detection in Correlated Gaussian and Non-Gaussian Radar Clutter*, Rome Laboratory, May 1993.

(Rasure 92) Rasure and Mark Young, "An Open Environment for Image Processing Software Development," *1992 SPIE/IS&T Symposium on Electronic Imaging*, SPIE Proceedings Vol. 1659, 14 February 1992.

(Roman 93) Jaime R. Roman and Dennis W. Davis, *State-Space Models for Multichannel Detection*, RL-TR-93-146, Rome Laboratory, July 1993.

(Taylor 80) Angus E. Taylor and David C. Lay, *Introduction to Functional Analysis*, Second Edition, John Wiley & Sons, 1980.

(Vienneau 93) Robert L. Vienneau, David Dekkers, and Sue Eilers, *Generalized Multichannel Signal Detection*, RL-TR-93-154, Rome Laboratory, July 1993.

(Vienneau 94) Robert L. Vienneau, *Fusion of Correlated Sensor Observations*, RL-TR-94-134, Rome Laboratory, August 1994.

Appendix A.
NOTATION AND ACRONYMS

A.1 Notation

A	(1) The amplitude in the constant magnitude signal model. (2) A constant used in defining the Probability Density Function for the quadratic form q. (3) A matrix.
$\hat{A}$	An estimate of the amplitude in the constant magnitude signal model.
$A^H(k)$	An AR coefficient.
$A_c^H(k)$	An AR coefficient for the Representative Model clutter.
B_k	A constant used in defining the Probability Density Function for the quadratic form q.
C_1, C_2	Some sets.
C_I	The Cholesky decomposition of the covariance matrix R_I.
C^n	The vector space consisting of n-dimensional complex vectors.
C_c	The Cholesky decomposition of the covariance matrix R_c.
C_k	A constant used in the rejection method.
C_s	The Cholesky decomposition of the covariance matrix R_s.
C_w	The Cholesky decomposition of the covariance matrix Σ_w.
C_e	The Cholesky decomposition of the covariance matrix Σ_e.
D_w	The diagonal matrix in the LDU decomposition of the covariance matrix Σ_w.
D_e	The diagonal matrix in the LDU decomposition of the covariance matrix Σ_e.
F	System dynamics matrix in state space model.
F_0	The null hypothesis filter.
F_1	The alternative hypothesis filter.
H	The observations matrix in the state space model.
H_0	The null hypothesis.
H_1	The alternative hypothesis.
$I(n)$	An interference process.
$I'(n)$	A single-channel process which is transformed to form the interference process.
J	The number of channels (elements) in a radar return.
K	(1) The number of realizations of a process. (2) Kalman gain matrix in state space model. (3) Amplitude matrix in a shaping function.
L_w	The lower triangular matrix in the LDU decomposition of the covariance matrix Σ_w.
L_e	The lower triangular matrix in the LDU decomposition of the covariance matrix Σ_e.

M	(1) A block covariance matrix related to the quadratic form q. (2) The number of channels in a radar return.
N	The number of time samples in a radar return.
P	A matrix used in the State Space Closed Form Method.
$P(k)$	A sequence of matrices used in the State Space Closed Form Method.
Q	The matrix of eigenvectors in the SVD of $\hat{\Sigma}$, an estimated block covariance matrix.
Q_w	The matrix of eigenvectors in the SVD of Σ_w.
Q_ϵ	The matrix of eigenvectors in the SVD of Σ_ϵ.
R	(1) A coordinate in a generalized spherical transformation of a SIRP. (2) A block covariance matrix
R_G	The norm of a Gaussian process.
R_I	The covariance matrix for interference in the Representative Model.
R_c, R_{cc}	The block covariance matrix for the clutter in the Representative Model.
R_s	The covariance matrix for the signal in the Representative Model.
$R_y(k)$	The kth lagged correlation matrix for the AR process $y(n)$.
T	(1) Either the Cholesky, LDU, or Singular Value Decomposition of Σ_ϵ. (2) A linear operator. (3) The sampling period in a shaping function.
T_ϵ	Either the Cholesky, LDU, or Singular Value Decomposition of Σ_ϵ.
T_c	Either the Cholesky, LDU, or Singular Value Decomposition of Σ_c.
T_w	Either the Cholesky, LDU, or Singular Value Decomposition of Σ_w.
U_1	A random variable used in the rejection method.
X	(1) A random variable from a Weibull distribution. (2) A vector space.
Y	A random variable from a Weibull distribution.
a	(1) A parameter of a Weibull distribution. (2) An amplitude of a signal.
a'	A constant used in the rejection method.
$a_s^*(k)$	An AR coefficient for the signal in the Representative Model.
b	A parameter of a Weibull distribution.
b'	A constant used in the rejection method.
c	(1) A clutter process. (2) A parameter in a linear transformation of a Weibull-distributed random variable. (3) A constant used in the rejection method.
$c(n)$	A clutter process.
d	A constant used in the rejection method.
d_1	A constant used in the rejection method.
$\frac{d}{\lambda}$	Normalized element spacing in a phased array radar.

$f_d T$	The normalized Doppler of a constant magnitude signal.
$f_{dI} T$	Normalized interference/jammer Doppler center frequency.
$f_{dP} T$	Normalized platform Doppler center frequency.
$f_{dS} T$	Normalized signal Doppler center frequency.
i	A channel index.
j	A channel index.
k	(1) A index for lags in an AR process. (2) A constant used in the rejection method.
l	A matrix of lag values at which a shaping function for a block covariance matrix peaks.
l_A	A transformation of a spatial and temporal lag index.
l_B	A transformation of a spatial and temporal lag index.
$l^p(n)$	A normed vector space.
l_s	A spatial lag index.
l_t	A temporal lag index.
m	A channel index.
n	(1) A time index; (2) A noise process.
p	The order of an AR process.
q	A quadratic form used in the definition of SIRPs.
r	A realization of the random variable R resulting from a generalized spherical coordinate transformation of a SIRP.
$\hat{r}_n(l)$	An estimate of the lth lag of a covariance matrix.
$\hat{r}_T^k(l)$	Time averaged estimate of the lth lag of a covariance matrix from the kth realization of the clutter process.
$\hat{r}_{TC}^k(l)$	Time averaged estimate of the lth lag of a covariance matrix with time clipping, from the kth realization of the clutter process.
$\hat{r}_{TS}(l)$	Time/space averaged estimate of the lth lag of a covariance matrix.
$\hat{r}_{TSC}(l)$	Time/space averaged estimate of the lth lag of a covariance matrix with time clipping.
s	(1) A signal process. (2) A constant used in the rejection method.
$s(n)$	A signal process.
$s'(n)$	A process used in synthesizing the representative model signal.
s_0, $s_0(n)$	The normalized constant magnitude signal. s_0 is a block vector formed by concatenating the time samples $s_0(n)$.
$s_{0,i}(n)$	A channel in the normalized constant magnitude signal.
$s_m(n)$	A channel in a signal process.
u_1, u_2	Realizations of random variables used in the rejection method.

$w(n)$	Weibull-distributed white noise.
x	(1) Radar returns. x is a block vector formed by concatenating the time samples $x(n)$. (2) A realization of the Weibull-distributed random variable X. (3) A vector in a vector spce.
$x(n)$	Radar returns.
y	(1) A clutter process. y is a block vector formed by concatenating the time samples $y(n)$. (2) A vector in a vector spce.
$y(n)$	A clutter process, either white noise or a stochastic process correlated across time (e.g. an AR process).
y^k, $y^k(n)$	The kth realization of a clutter process. y^k is a block vector formed by concatenating the time samples $y^k(n)$.
z	A vector in a vector space.
$z(n)$	A unit-variance zero-mean Gaussian process.
Θ	A coordinate in a generalized spherical transformation of a SIRP.
Γ	A matrix in the State Space Closed Form Method.
Λ	(1) The loglikelihood statistic. (2) The diagonal matrix of eigenvalues in the SVD of $\hat{\Sigma}$, an estimated block covariance matrix.
Λ_w	The diagonal matrix of eigenvalues in the SVD of Σ_w.
Λ_ε	The diagonal matrix of eigenvalues in the SVD of Σ_ε.
Σ	The covariance matrix for the innovations in the innovations representation of the state space model.
$\hat{\Sigma}$	Estimated block covariance matrix.
Σ_c	The driving noise covariance matrix for the Representative Model clutter
Σ_w	The covariance matrix for weibull noise $w(n)$.
Σ_ε	The covariance matrix for the driving noise $\varepsilon(n)$ in an AR process.
$\Phi_{j,n}$	A coordinate in a generalized spherical transformation of a SIRP.
α	(1) A significance level, that is, the probability of erroneously accepting an alternative hypothesis. In radar, the probability of false alarm. (2) An angle. (3) A scalar.
α'	(1) A tail probability. (2) An unknown complex amplitude to modify a steering vector. (3) An angle in the Representative Model of the clutter.
$\alpha(n)$	State variables in state space model.
β	(1) The probability of erroneous accepting a null hypothesis. In radar, 1 minus the probability of detection. (2) A parameter of a spatial shaping function.
ε	(1) A driving noise term in an AR process. ε is a block vector formed by concatenating the time samples $\varepsilon(n)$. (2) The innovations in the innovations representation of the state space model.

ζ	A parameter of the exponential power spectrum temporal shaping function.
θ	The initial phase.
θ_s	Angle to the signal.
λ	(1) An eigenvalue. (2) A matrix of intertemporal one-lag correlation parameters in a shaping function.
μ_X	The mean of the random variable X.
μ_Y	The mean of the random variable Y.
μ_s	One lag spatial correlation parameter.
μ_t	One lag temporal correlation parameter.
$\nu(n)$	A white noise process uncorrelated both in time and across channels.
$\nu_\varepsilon(n)$	A white noise process uncorrelated both in time and across channels.
$\nu_0(n)$	The innovations output of the null hypothesis filter.
$\nu_1(n)$	The innovations output of the alternative hypothesis filter.
ξ	A Weibull-distributed Spherically Invariant Random Process (SIRP). ξ is a block vector formed by concatenating the time samples $\xi(n)$.
π	A constant; the ratio of the circumference of a circle to its diameter.
σ	The standard deviation of a Weibull distribution.
σ_I	A standard deviation for interference in the Representative Model.
σ_X	The standard deviation of the random variable X.
σ_Y	The standard deviation of the random variable Y.
σ_i	Standard deviation for the ith channel.
σ_s	A standard deviation.
σ_{ν_j}	The standard deviation of the jth channel of the process $\nu(n)$.
ϕ	The Doppler shift in a shaping function.
ϕ_0	Angular beam direction.
ϕ_I	Angular direction to interference/jammer.
ϕ_s	Angular direction to signal.
$\|\ \|$	A norm mapping a vector into the real numbers.
$\|\ \|_p$	The $l^p(n)$ norm.
$\|\ \|_\infty$	The $l^\infty(n)$ norm.
$F_t^I(l_t)$	A temporal shaping function for interference.
$F_t^s(\ )$	A temporal shaping function for the signal process.
$M(\ )$	An event used in the rejection method.
$f(\ )$	A Probability Density Function.
$f_R(\ )$	The Probability Density Function of the random variable R.

$f_{U_1}(\ )$	The Probability Density Function of the random variable U_1.
$f_{U_1,U_2}(\ )$	A joint Probability Density Function.
$f_q(\ )$	The Probability Density Function of the quadratic form q.
$g(\ )$	(1) A Probability Density Function. (2) A function.
$h(\ )$	(2) A triangular function.
$h_{NJ}(\ ), h_{2NJ}(\ )$	A function used in defining the Probability Density Function for the quadratic form q.
$\Gamma(\ )$	A standard mathematical function.
$\rho(\ )$	The spectral radius of a matrix.

A.2 Acronyms

AR	Autoregressive
ARMA	Autoregressive Moving Average
BAA	Broad Agency Announcement
CDF	Cumulative Distribution Function
COTS	Commercial Off The Shelf
FFT	Fast Fourier Transform
GUI	Graphical User Interface
KSC	Kaman Sciences Corporation
MSPSS	Multichannel Signal Processing Simulation System
PDF	Probability Density Function
RL	Rome Laboratory
RLSTAP/ADT	Rome Laboratory Space-Time Adaptive Processing Algorithm Development Tool
SIRP	Spherically Invariant Random Process
STAP	Space-Time Adaptive Processing
SVD	Singular Value Decomposition
UFI	User Front-end Interface

Appendix B.
THE WEIBULL DISTRIBUTION

Let X be a random variable from the Weibull distribution. The Probability Density Function (PDF) for X is:

$$f(x) = a\,b\,x^{b-1} Exp\left[-a\,x^{b}\right],\ 0 \le x. \tag{B-1}$$

The Cumulative Distribution Function (CDF) is:

$$F(x) = 1 - Exp\left[-a\,x^{b}\right],\ 0 \le x. \tag{B-2}$$

The mean of X is $a^{\frac{1}{b}}\Gamma\left[1+\frac{1}{b}\right]$ and the variance is $a^{\frac{2}{b}}\left[\Gamma\left[1+\frac{2}{b}\right] - \Gamma^{2}\left[1+\frac{1}{b}\right]\right]$.

The PDF of a Weibull distribution is often written in a slightly different form. This alternate expression is defined by a scale and shape parameter. In the above expression, $a^{1/b}$ is the scale parameter and b is the shape parameter. These parameters can be set so as to control the mean of the Weibull distribution. Equation B-3 ensures the mean is unity:

$$a = \left[\frac{1}{\Gamma\left[1+\frac{1}{b}\right]}\right]^{b}. \tag{B-3}$$

A linear transformation of a Weibull distribution is also Weibull. Let the random variable Y be formed from a linear transformation of X:

$$Y = c\,X. \tag{B-4}$$

Then Y has the following PDF:

$$g(y) = \left[\frac{a}{c^{b}}\right] b\,y^{b-1} Exp\left[-\left[\frac{a}{c^{b}}\right] y^{b}\right],\ y > 0. \tag{B-5}$$

The means of X and Y are related as follows:

$$\mu_Y = \frac{1}{c}\,\mu_X, \tag{B-6}$$

$$\sigma_Y^2 = \frac{1}{c^2}\,\sigma_X^2. \tag{B-7}$$

For example, if the mean of X is unity, then X should be multiplied by $\frac{1}{\mu_Y}$ to obtain a Weibull distribution with the indicated mean. The variance will be transformed as well.

Appendix C.
THE REJECTION METHOD

The synthesis of Weibull-distributed Spherically Invariant Random Process (SIRP) clutter relies on the distribution of a random variable R. The Probability Density Function (PDF) for R is quite complicated, and the corresponding Cumulative Distribution Function (CDF) cannot be found in closed form. Consequently, the generation of R from the appropriate distribution cannot be found by a simple transformation of an uniformly distributed random variable.

C.1 A Simple Version

The rejection method provides a Monte Carlo technique for generating random variates from a distribution with a known PDF $f_R(r)$. The more complicated and efficient method that was implemented can best be understood by first considering a simpler version of the rejection method. Let b' be such that

$$\int_0^{b'} f_R(r)\,dr \approx 1, \tag{C-1}$$

and c be such that

$$c = \max f_R(r). \tag{C-2}$$

The rejection method then proceeds as follows:

- **Step 1**: Generate u_1 from a distribution uniform on $(0, b')$.
- **Step 2**: Generate u_2 from a distribution uniform on $(0, c)$, where u_1 and u_2 are stochastically independent.
- **Step 3**: If $u_2 \le f_R(u_1)$, set R to u_1. Otherwise, reject u_1 and return to Step 1.

The proof of the rejection method is based on the theory of conditional probability. A basic definition is that the conditional probability of an event C_2, given the event C_1, is the ratio of the probability of the intersection of the events and the probability of C_1:

$$Pr(C_2 | C_1) = \frac{Pr(C_2 \cap C_1)}{Pr(C_1)}. \tag{C-3}$$

Define the event $M(U_1)$ as follows:

$$M(U_1) = \{(u_1, u_2) \mid 0 \le u_1 \le U_1 \text{ and } 0 \le u_2 \le f_R(u_1)\}. \tag{C-4}$$

The rejection method generates u_1, given (u_1, u_2) is in $M(b')$. What needs to be shown to guarantee the rejection method works is that a conditional probability associated with this procedure is as desired:

$$Pr[0 \le u_1 \le U_1 \mid (u_1, u_2) \text{ is in } M(b')] = \int_0^{U_1} f_R(u_1)\,du_1. \tag{C-5}$$

Proof: By the definition of conditional probability,

$$Pr[0 \le u_1 \le U_1 \mid (u_1, u_2) \text{ is in } M(b')] = \frac{Pr[(u_1, u_2) \text{ is in } M(U_1)]}{Pr[(u_1, u_2) \text{ is in } M(b')]}. \tag{C-6}$$

Since u_1 and u_2 are realizations of stochastically independent uniform random variables, the probability of the location of (u_1, u_2) is equally distributed on the rectangle $\{(u_1, u_2) \mid 0 \le u_1 \le b' \text{ and } 0 \le u_2 \le c\}$. The area of this rectangle is $b'c$. The area of the set $M(U_1)$ is the

integral $\int_0^{U_1} f_R(u_1)\,du_1$. Hence,

$$Pr\,[(u_1,u_2)\ is\ in\ M(U_1)] = \frac{1}{b'c}\int_0^{U_1} f_R(u_1)\,du_1. \tag{C-7}$$

Or

$$Pr\,[0 \le u_1 \le U_1 \mid (u_1,u_2)\ is\ in\ M(b')] = \frac{\frac{1}{b'c}\int_0^{U_1} f_R(u_1)\,du_1}{\frac{1}{b'c}}. \tag{C-8}$$

Equation C-8 implies Equation C-5, which was to be shown.

C.2 A Sophisticated Version

The simple version of the rejection method will reject a large proportion of the uniform variates generated. A more efficient method is available to reduce the number of rejected variates.

Let R have PDF $f_R(r)$. No special restrictions are imposed on f_R. Let u_1 have PDF $f_{U_1}(u_1)$ where there exists an a' greater than zero such that

$$a' f_R(u_1) \le f_{U_1}(u_1). \tag{C-9}$$

Define $g(u_1)$ by Equation C-10:

$$g(u_1) = \frac{a' f_R(u_1)}{f_{U_1}(u_1)}. \tag{C-10}$$

where $f_{U_1}(u_1)$ is strictly positive. Notice that over the region in which $g(u_1)$ is defined,

$$g(u_1) \le 1. \tag{C-11}$$

The rejection method is then defined by the following algorithm:

- **Step 1**: Generate u_1 from the distribution given by $f_{U_1}(u_1)$.
- **Step 2**: Generate u_2 from a distribution uniform on (0, 1), where u_1 and u_2 are stochastically independent.
- **Step 3**: If $u_2 \le g(u_1)$, set R to u_1. Otherwise, reject u_1 and return to Step 1.

Before proving that this method generates a variate from the desired distribution, it seems advisable to demonstrate the simple version is a special case of the more sophisticated version of the rejection method. Accordingly, assume there exists a b' and c such that Equations C-1 and C-2 hold. Let u_1 be uniformly distributed on (0, b') for this special case. Notice that Equation C-12 holds:

$$\frac{1}{b'c} f_R(u_1) \le f_{U_1}(u_1). \tag{C-12}$$

In other words Equation C-9 holds with a' set to $\frac{1}{b'c}$. Then $g(u_1)$ is defined as

$$g(u_1) = \frac{1}{c} f_R(u_1),\ 0 \le u_1 \le b'. \tag{C-13}$$

Let $c\,u_2$ be uniform on $(0, c)$. The simple version of the rejection method accepts u_1 if $c\,u_2 \le f_R(u_1)$. This rule is equivalent to accepting u_1 if $u_2 \le g(u_1)$. So the simple version is indeed a special case of the more sophisticated version of the rejection method.

The proof of the sophisticated version in its full generality relies on the theory of conditional probability. Define the event $M(U_1)$ by Equation C-14:

$$M(U_1) = \{(u_1,u_2) |\ u_1 \le U_1 \text{ and } 0 \le u_2 \le g(u_1)\}. \tag{C-14}$$

The rejection method ensures (u_1,u_2) will be in $M(\infty)$ when u_1 is accepted. Equation C-15 is what needs to be shown for the rejection method to work:

$$Pr[\ u_1 \le U_1 | (u_1,u_2) \text{ is in } M(\infty)] = \int_{-\infty}^{U_1} f_R(u_1)\,du_1 = F_R(U_1), \tag{C-15}$$

where F_R is the CDF for R.

Proof: By conditional probability,

$$Pr[u_1 \le U_1 | (u_1,u_2) \text{ is in } M(\infty)] = \frac{Pr[(u_1,u_2) \text{ is in } M(U_1)]}{Pr[(u_1,u_2) \text{ is in } M(\infty)]}. \tag{C-16}$$

Since u_1 and u_2 are stochasically independent, the joint PDF of (u_1,u_2) is the product of their individual PDFs:

$$f_{U_1,U_2}(u_1,u_2) = f_{U_1}(u_1),\ 0 \le u_2 \le 1. \tag{C-17}$$

The probability that (u_1,u_2) is in $M(U_1)$ is found as follows:

$$Pr[(u_1,u_2) \text{ is in } M(U_1)] = \int_{-\infty}^{U_1} \int_{0}^{g(u_1)} f_{U_1,U_2}(u_1,u_2)\,du_2\,du_1, \tag{C-18}$$

$$Pr[(u_1,u_2) \text{ is in } M(U_1)] = \int_{-\infty}^{U_1} \int_{0}^{g(u_1)} f_{U_1}(u_1)\,du_2\,du_1, \tag{C-19}$$

$$Pr[(u_1,u_2) \text{ is in } M(U_1)] = \int_{-\infty}^{U_1} f_{U_1}(u_1) \int_{0}^{g(u_1)} du_2\,du_1, \tag{C-20}$$

$$Pr[(u_1,u_2) \text{ is in } M(U_1)] = \int_{-\infty}^{U_1} f_{U_1}(u_1)\,g(u_1)\,du_1, \tag{C-21}$$

$$Pr[(u_1,u_2) \text{ is in } M(U_1)] = \int_{-\infty}^{U_1} f_{U_1}(u_1)\frac{a' f_R(u_1)}{f_{U_1}(u_1)}\,du_1, \tag{C-22}$$

$$Pr[(u_1,u_2) \text{ is in } M(U_1)] = a' \int_{-\infty}^{U_1} f_R(u_1)\,du_1, \tag{C-23}$$

$$Pr[(u_1,u_2) \text{ is in } M(U_1)] = a'\,f_R(U_1). \tag{C-24}$$

Hence,

$$Pr\,[u_1 \le U_1 \,|\, (u_1, u_2)\ is\ in\ M(\infty)] = \frac{a'\,F_R(U_1)}{a'\,F_R(\infty)} = F_R(U_1), \tag{C-25}$$

which was to be shown.

C.3 An Application

The rejection method is used in the synthesis of Weibull-distributed SIRPs. Specifically, it is used to generate a random variable R with the PDF given by Equation C-26:

$$f_R(r) = \frac{r^{2NJ-1}}{2^{NJ-1}\Gamma(NJ)}\, h_{2NJ}(r^2),\ r > 0, \tag{C-26}$$

where

$$h_{2NJ}(r^2) = (-2)^{NJ} \sum_{k=1}^{NJ} B_k \frac{A^k}{k!}\, r^{(kb-2NJ)}\, Exp\,(-A\,r^b), \tag{C-27}$$

$$A = a\,\sigma^b, \tag{C-28}$$

$$\sigma^2 = \frac{1}{2}\left(\frac{1}{a}\right)^{\frac{2}{b}} \Gamma\left(1 + \frac{2}{b}\right) \tag{C-29}$$

$$B_k = \sum_{m=1}^{k} (-1)^m \binom{k}{m} \prod_{i=0}^{NJ-1} \left(\frac{m\,b}{2} - i\right). \tag{C-30}$$

The parameter $0 < b \le 2$ is specified by the user, and a is chosen such that σ is unity:

$$a = \left[\frac{1}{2}\Gamma\left(1 + \frac{2}{b}\right)\right]^{\frac{b}{2}}. \tag{C-31}$$

For small values of b and NJ, the PDF either monotonically decreases for all positive values or peaks very close to the origin. For larger values, the PDF first increases and then decreases. In the first case, the PDF can be bounded by a triangle with a right angle at the origin. In the second case, the PDF can be bounded by a differently shaped triangle.

C.3.1 A Triangular Distribution

Consider the PDF $f_R(r)$ for small values of b and NJ. Suppose the PDF always lies below the triangular function $h(r)$ shown in Figure C-1 in the region (0, d). The parameter d is chosen such that the probability that R exceeds d is negligible. $h(r)$ is defined by:

$$h(r) = \begin{cases} s\left[1 - \frac{r}{d}\right], 0 \le r \le d \\ 0, else. \end{cases} \tag{C-32}$$

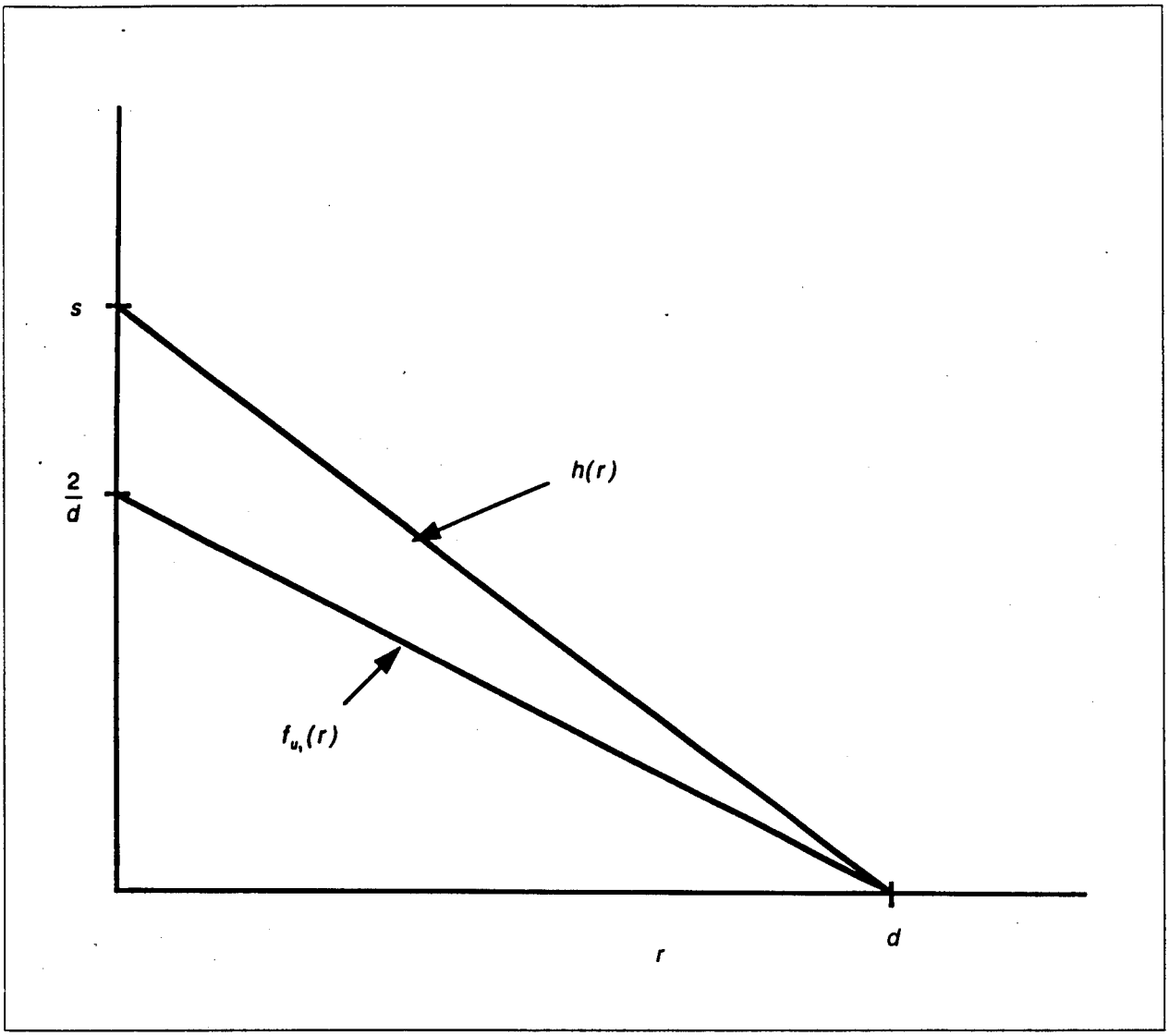

Figure C-1: Triangular Functions

$h(r)$ is not a PDF since the area under the triangle exceeds unity. But consider the function $f_{U_1}(u_1)$ defined as follows:

$$f_{U_1}(u_1) = \frac{2}{d\,s}\,h(u_1) = \begin{cases} \frac{2}{d}\left(1 - \frac{r}{d}\right), 0 \le r \le d \\ 0, \textit{else.} \end{cases} \tag{C-33}$$

$f_{U_1}(u_1)$ is a PDF. Furthermore, the inequality given by Display C-9 is satisfied for the PDF in Equation C-26 and

$$a' = \frac{2}{d\,s}. \tag{C-34}$$

Hence,

$$g(u_1) = \frac{f_R(u_1)}{h(u_1)} \tag{C-39}$$

The random variable U_1 can be generated based on the following transformation:

$$U_1 = d\,(1 - \sqrt{U}\,), \tag{C-40}$$

where U is from a uniform distribution on (0, 1).

In summary, a variate is generated from the PDF given by Equation C-26 as follows:

- **Step 1**: Generate u from a distribution uniform on (0, 1).
- **Step 2**: Transform to the triangular distribution with PDF $f_{U_1}(u_1)$ using Equation C-40.
- **Step 3**: Generate u_2 from a distribution uniform on (0, 1). u and u_2 are stochastically independent.
- **Step 4**: If $u_2 \le g\,(u_1)$, where g is given by Equation C-39, set R to u_1. Otherwise, reject u_1 and return to Step 1.

C.3.2 Another Triangular Distribution

Now consider the PDF $f_R\,(r)$ for larger values of b and NJ. In this case, suppose the PDF for R is bounded above by the triangular function $h\,(r)$ redefined by Equation C-41:

$$h\,(r) = \begin{cases} \dfrac{s}{d_1}\,r\,,\ 0 \le r \le d_1 \\ \dfrac{s\,(d - r)}{(d - d_1)}\,,\ d_1 \le r \le d\,. \end{cases} \tag{C-41}$$

The function $h\,(r)$ is shown in Figure C-2. Once again, define the PDF $f_{U_1}(u_1)$ as follows:

$$f_{U_1}(u_1) = \frac{2}{d\,s}\,h\,(u_1) = \begin{cases} \dfrac{2}{d\,d_1}\,u_1\,,\ 0 \le u_1 \le d_1 \\ \dfrac{2(d - u_1)}{d\,(d - d_1)}\,,\ d_1 \le u_1 \le d\,. \end{cases} \tag{C-42}$$

The inequality given by Display C-9 remains true for a' as defined in Equation C-34, and $g\,(u_1)$ is redefined by Equation C-35 in terms of the redefined function $h\,(u_1)$.

The random variable U_1 can be generated based on using the inverse of the CDF to transform a uniform variate. The CDF for U_1 is:

$$F_{U_1}(u_1) = \begin{cases} \dfrac{u_1^2}{d\,d_1}\,,\ 0 \le u_1 \le d_1 \\ \dfrac{u_1}{d} - \dfrac{(u_1 - d)(u_1 - d_1)}{d\,(d - d_1)}\,,\ d_1 \le u_1 \le d\,. \end{cases} \tag{C-43}$$

The inverse of the CDF is:

$$F_{U_1}^{-1}\,(u) = \begin{cases} \sqrt{d\,d_1\,u}\,,\ 0 \le u \le \dfrac{d_1}{d} \\ d - \sqrt{d^2 - [d\,d_1 + d\,(d - d_1)\,u\,]}\,,\ \dfrac{d_1}{d} \le u \le 1\,. \end{cases} \tag{C-44}$$

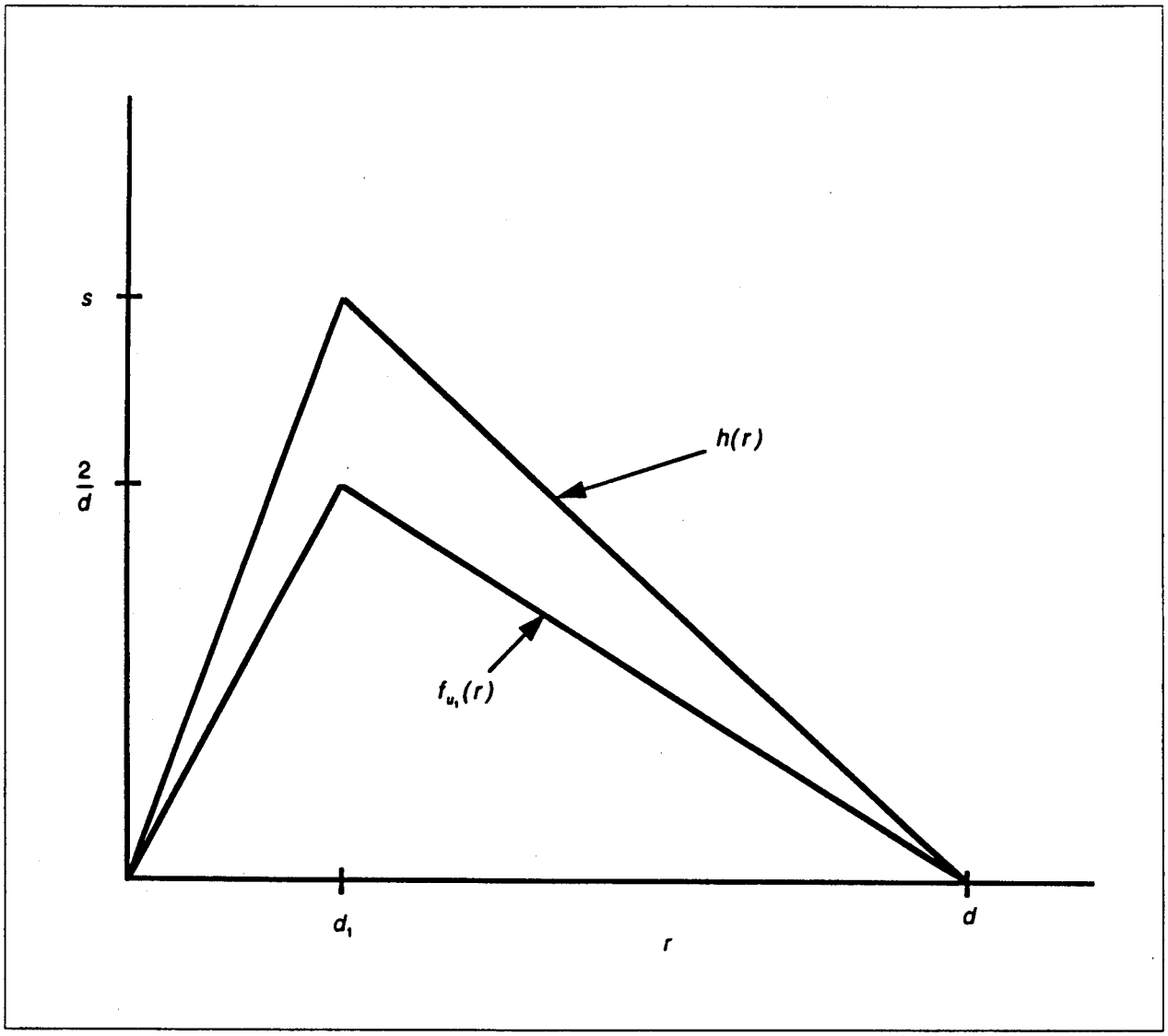

Figure C-2: More Triangular Functions

In summary, a variate is generated from the PDF given by Equation C-26 in this case as follows:

- **Step 1**: Generate u from a distribution uniform on (0, 1).
- **Step 2**: Transform to the triangular distribution with PDF $f_{U_1}(u_1)$:

$$u_1 = F_{U_1}^{-1}(u), \tag{C-45}$$

  where $F_{U_1}^{-1}$ is given by Equation C-41.
- **Step 3**: Generate u_2 from a distribution uniform on (0, 1). u and u_2 are stochastically independent.
- **Step 4**: If $u_2 \leq g(u_1)$, where g is given by Equation C-39, set R to u_1. Otherwise, reject u_1 and return to Step 1.

C.3.3 The Parameters of the Triangular Distribution

The above algorithm presumes that the parameters d, d_1, and s of the triangular functions $h(r)$ are known. d is chosen such that almost all the probability for R is concentrated on (0, d). Since the Weibull SIRP is used for false alarm analysis, the tail of R needs to be closely approximated for very large values of R.

Let α' denote the tail (false alarm) probability that is to be approximated. To ensure that R is synthesized such that this tail probability is accurate, points further out on the tail need to be synthesized. Accordingly, R is synthesized on the space (0, d) where

$$Pr(R > d) = \int_d^\infty f_R(r)\,dr, \tag{C-46}$$

and

$$\alpha = k\,\alpha'. \tag{C-47}$$

The constant k should be selected to be much less than unity. A default value of $k = 0.01$ is appropriate.

Equation C-46 defines d implicitly. Further progress requires the calculation of the integral on the right hand side of Equation C-46. Let C_k be defined by Equation C-48:

$$C_k = \frac{(-2)^{NJ}}{2^{NJ-1}\Gamma(NJ)} B_k \frac{A^k}{k!} = \frac{2(-1)^{NJ} A^k}{(NJ-1)!\,k!} \sum_{m=1}^{k} \left[(-1)^m \binom{k}{m} \prod_{i=0}^{NJ-1} \left(\frac{mb}{2} - i \right) \right]. \tag{C-48}$$

The PDF of R is then

$$f_R(r) = \sum_{k=1}^{NJ} C_k\, r^{bk-1} e^{-A r^b}. \tag{C-49}$$

Equation C-46 can be rewritten as follows:

$$Pr(R > d) = \int_d^\infty \sum_{k=1}^{NJ} C_k\, r^{bk-1} e^{-A r^b}\,dr, \tag{C-50}$$

$$Pr(R > d) = \sum_{k=1}^{NJ} C_k \int_d^\infty r^{bk-1} e^{-A r^b}\,dr. \tag{C-51}$$

Let $x = r^b$. Then,

$$Pr(R > d) = \frac{1}{b} \sum_{k=1}^{NJ} C_k \int_{d^b}^\infty x^{k-1} e^{-Ax}\,dx. \tag{C-52}$$

One can prove by induction the formula for the integral given in Equation C-53:

$$\int x^{k-1} e^{-Ax}\,dx = -\sum_{i=0}^{k-1} \frac{(k-1)!}{(k-i-1)!\,A^{i+1}} x^{k-i-1} e^{-Ax}, \quad k = 1, 2, 3, \cdots \tag{C-53}$$

Hence,

$$Pr(R > d) = -\frac{1}{b} \sum_{k=1}^{NJ} C_k \left[\sum_{i=0}^{k-1} \frac{(k-1)!}{(k-i-1)!\,A^{i+1}} x^{k-i-1} e^{-Ax} \right]_{d^b}^{\infty}. \tag{C-54}$$

$$Pr(R > d) = \frac{1}{b}\sum_{k=1}^{NJ} C_k \left[\sum_{i=0}^{k-1} \frac{(k-1)!}{(k-i-1)!A^{i+1}} d^{(k-i-1)b}\right] e^{-A d^b}. \quad \text{(C-55)}$$

Or,

$$\alpha = \frac{1}{b}\sum_{k=1}^{NJ} C_k \left[\sum_{i=0}^{k-1} \frac{(k-1)!}{(k-i-1)!A^{i+1}} d^{(k-i-1)b}\right] e^{-A d^b}. \quad \text{(C-56)}$$

So d is found numerically such that

$$\alpha \geq \frac{1}{b}\sum_{k=1}^{NJ} C_k \left[\sum_{i=0}^{k-1} \frac{(k-1)!}{(k-i-1)!A^{i+1}} d^{(k-i-1)b}\right] e^{-A d^b}. \quad \text{(C-57)}$$

d_1 is set equal to the point that maximizes the PDF for r. This value is found numerically by using the bisection method to find the value of r for which the derivative of the PDF is equal to zero. The maximum value s of the triangular function is found by stepping off the r axis in small increments. The smallest value of s ensuring that the PDF never exceeds the triangular function is thereby determined empirically.

Appendix D.
MATRIX NORMS

The State Space Closed Form algorithm for calculating the block covariance matrix corresponding to an AR process uses an iterative procedure to calculate an intermediate matrix P. This iterative procedure is said to converge when the norm of the difference between successive iterations is less than some small value. Thus, the implementation of this algorithm requires a decision on the most appropriate method for calculating matrix norms. This appendix briefly reviews the appropriate theory for matrix and vector norms.

Intuitively, a norm is the length of a vector. A vector is an element of a vector space, where a vector space is a set with addition and scalar multiplication operations and a certain formal structure (Taylor 80). The definition of a vector space generalizes and axiomatizes the intuitive concept of n dimensional real vectors. The space of all matrices of a specified size satisfies the needed properties. Consequently, a matrix can be thought of as a vector, and can have a norm.

A norm on a vector space X is a function $\| \ \|$ mapping a vector into the real numbers. The function $\| \ \|$ must have the following properties:

- For all vectors x and y, $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)
- For all vectors x and scalars α, $\|\alpha x\| \leq |\alpha| \|x\|$.
- For all vectors x, $\|x\| \geq 0$.
- $\|x\| = 0$ if and only if x is the zero vector.

The $l^p(n)$ spaces are probably the most obvious examples of norms. Consider the vector space C^n, the space of n dimensional complex vectors. Let $z = (z_1, z_2, ..., z_n)$ denote an element of that space. Then the $l^p(n)$ norm is defined by:

$$\|z\|_p = \left[\sum_{i=1}^{n} |z_i|^p \right]^{1/p}. \tag{D-1}$$

The $l^2(n)$ norm is also known as the Euclidean norm. The $l^\infty(n)$ norm is defined by

$$\|z\|_\infty = \max_i |z_i|. \tag{D-2}$$

A matrix can be thought of as a basis-specific representation of a linear operator mapping one vector space into another. If X and Y are normed vector spaces, a norm can be defined on the space $\{ T \mid T: X \rightarrow Y \}$ by Equation D-3:

$$\|T\| = \sup_{\|x\| \leq 1} \|Tx\| = \sup_{\|x\| = 1} \|Tx\| = \sup_{\|x\| \neq 1} \frac{\|Tx\|}{\|x\|} \tag{D-3}$$

where *sup* denotes the least upper bound (supremum) of a set. Any matrix norm induced by a vector norm on an $n \times n$ matrix is known as a natural norm (Burden 81).

Each $l^p(n)$ induces a natural matrix norm. The norm induced by the Euclidean (vector) norm is of particular interest. This norm is related to the spectral radius of a matrix. The spectral radius $\rho(A)$ of the matrix A is defined by Equation D-4:

$$\rho(A) = \max |\lambda|, \tag{D-4}$$

where λ is an eigenvalue of A. The matrix norm induced by the Euclidean norm is defined by Equation D-5:

$$\| A \|_2 = [\rho(A^H A)]^{1/2}. \tag{D-5}$$

The spectral radius of a matrix is related to all natural norms by Equation D-6:

$$\rho(A) \leq \| A \|, \tag{D-6}$$

for any natural norm $\| \ \|$.

These mathematical facts suggest two alternatives for choosing a norm in a specific numerical algorithm. One could select a matrix norm induced by a $l^p(n)$ norm. Values of p = 1, 2, and ∞ are the obvious candidates. The $l^2(n)$ norm is easily implemented. Since $A^H A$ is Hermitian, the singular values of $A^H A$, as found in a Singular Value Decomposition (SVD), are also eigenvalues of $A^H A$. Or one could consider the spectral radius of a matrix A. Minimizing this quantity is equivalent to minimizing a *lower* bound on all natural norms of the matrix A. This second choice was used in the State Space Closed Form algorithm for calculating the block covariance matrix corresponding to an AR process.

Appendix E.
THE EQUIVALENCE OF TWO METHODS

Two methods have been given for finding lagged covariance matrices for an AR process past the order m of the process. The State Space Closed Form Method is based on considering an AR process as a special case of an innovations representation of a state space process. Covariance matrices are then found for the state space representation. The AR Recursion Method proceeds more directly by considering certain properties of an AR process.

Assuming the solution of the State Space Closed Form Method is unique, these two methods are equivalent when the correlation functions describe the AR process exactly. This appendix demonstrates this proposition. They may, however, have different numerical stability results.

The AR process is defined by the AR coefficients $A^H(1)$, $A^H(2)$, ..., $A^H(m)$ and driving noise covariance matrix Σ. The lagged covariance matrices for lags $-m$, $-m+1$, ..., 1, 0, 1, ..., m are related to the AR coefficients through the Yule-Walker equations. The AR Recursion Method calculates block covariance matrices for larger lags by use of Equation E-1:

$$R(l) = -\sum_{k=1}^{m} A^H(k) R(l-k), \quad l = m+1, m+2, \cdots \tag{E-1}$$

The innovations representation of a state space process is defined by certain block covariance matrices - the system dynamics matrix F, the Kalman gain K, and the observation matrix H. For an AR process, the system dynamics matrix, the Kalman gain, and the observation matrix are given by Equations E-2, E-3, and E-4:

$$F = \begin{bmatrix} -A^H(1) & -A^H(2) & \cdots & -A^H(m-1) & -A^H(m) \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \tag{E-2}$$

$$K = \begin{bmatrix} I \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \tag{E-3}$$

$$H = \begin{bmatrix} -A(1) \\ -A(2) \\ \cdot \\ \cdot \\ \cdot \\ -A(m) \end{bmatrix} \tag{E-4}$$

The State Space Closed Form Method makes use of two intermediate matrices P and Γ. P is found by an iterative algorithm to solve Equation E-5:

$$P = F P K^H + K \Sigma K^H \tag{E-5}$$

Γ is defined by Equation E-6:

$$\Gamma = F P H + K \Sigma \tag{E-6}$$

The State Space Closed Form Method uses Equations E-7 and E-8 to find covariance matrices for all nonnegative lags:

$$R(0) = H^H P H + \Sigma \tag{E-7}$$

$$R(l) = H^H F^{l-1} \Gamma, \; l = 1, 2, \cdots \tag{E-8}$$

The remainder of this appendix shows that covariance matrices that satisfy the Yule-Walker equations and Equation E-1 will also satisfy Equations E-7 and E-8.

E.1 Some Useful Formulas

The relationship between the parameters of an AR process and the corresponding correlation functions is expressed by Display E-9:

$$\begin{bmatrix} I & A^H(1) & \cdots & A^H(m) \end{bmatrix} \begin{bmatrix} R(0) & R(1) & \cdots & R(m) \\ R(-1) & R(0) & \cdots & R(m-1) \\ \vdots & \vdots & & \vdots \\ R(-m) & R(-m+1) & \cdots & R(0) \end{bmatrix} = \begin{bmatrix} \Sigma & 0 & \cdots & 0 \end{bmatrix} \tag{E-9}$$

The last m columns of this system of equations can be written as in Equation E-10:

$$R(l) + \sum_{k=1}^{m} A^H(k) R(l-k) = 0, \; l = 1, 2, \cdots, m. \tag{E-10}$$

Or,

$$R(l) = -\sum_{k=1}^{m} A^H(k) R(l-k), \; l = 1, 2, \cdots, m. \tag{E-11}$$

The AR Recursion Method extends Equation E-11 to all strictly positive l. Thus, Equation E-12 holds for $R(l)$ satisfying the AR Recursion method:

$$R(l) = -\sum_{k=1}^{m} A^H(k) R(l-k), \; l = 1, 2, 3, \cdots. \tag{E-12}$$

The Hermitian transpose of the Yule-Walker equations is given by Display E-13:

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(m) \\ R(-1) & R(0) & \cdots & R(m-1) \\ \vdots & \vdots & & \vdots \\ R(-m) & R(-m+1) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} I \\ A(1) \\ \vdots \\ A(m) \end{bmatrix} = \begin{bmatrix} \Sigma \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{E-13}$$

The first row of this system yields Equation E-14:

$$-\sum_{k=1}^{m} A(k) R(k) = R(0) - \Sigma. \tag{E-14}$$

The remaining rows yield Equation E-15:

$$R(-l) = -\sum_{k=1}^{m} A(k)R(k-l),\ l = 1, 2, \cdots, m. \tag{E-15}$$

E.2 The Proof

The proof of the equivalence of the two methods for the special case of AR processes is presented here by means of a succession of lemmas. The first lemma relates P to the covariance matrices:

Lemma E-1: Let P be the solution of Equation E-16:

$$P = F P K^H + K \Sigma K^H, \tag{E-16}$$

where F, K, and Σ are as in the state space representation of an AR process of order m. Then P is given by Equation E-17:

$$P = \begin{bmatrix} R(0) & R(1) & \cdots & R(m-1) \\ R(-1) & R(0) & \cdots & R(m-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(-m+1) & R(-m+2) & \cdots & R(0) \end{bmatrix}. \tag{E-17}$$

Proof: Assume the solution to Equation E-16 is unique. The proof proceeds by showing that when the conjectured value for P is plugged into the right hand side of Equation E-16, the desired left hand side results. To begin, calculate FP:

$$FP = \begin{bmatrix} -A^H(1) & -A^H(2) & \cdots & -A^H(m-1) & -A^H(m) \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} R(0) & R(1) & \cdots & R(m-1) \\ R(-1) & R(0) & \cdots & R(m-2) \\ R(-2) & R(-1) & \cdots & R(m-3) \\ \vdots & \vdots & \ddots & \vdots \\ R(-m+1) & R(-m+2) & \cdots & R(0) \end{bmatrix} \tag{E-18}$$

Thus,

$$FP = \begin{bmatrix} -\sum_{k=1}^{m} A^H(k)R(1-k) & -\sum_{k=1}^{m} A^H(k)R(2-k) & \cdots & -\sum_{k=1}^{m} A^H(k)R(m-k) \\ R(0) & R(1) & \cdots & R(m-1) \\ R(-1) & R(0) & \cdots & R(m-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(-m+2) & R(-m+3) & \cdots & R(1) \end{bmatrix} \tag{E-19}$$

Using Equation E-11, one obtains Equation E-20:

$$FP = \begin{bmatrix} R(1) & R(2) & R(3) & \cdots & R(m) \\ R(0) & R(1) & R(2) & \cdots & R(m-1) \\ R(-1) & R(0) & R(1) & \cdots & R(m-2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ R(-m+2) & R(-m+3) & R(-m+4) & \cdots & R(1) \end{bmatrix} \tag{E-20}$$

The next step is to determine FPF^H.

$$FPF^H =$$

$$\begin{bmatrix} R(1) & R(2) & \cdots & R(m) \\ R(0) & R(1) & \cdots & R(m-1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R(-m+3) & R(-m+4) & \cdots & R(2) \\ R(-m+2) & R(-m+3) & \cdots & R(1) \end{bmatrix} \begin{bmatrix} -A(1) & I & \cdots & 0 \\ -A(2) & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ -A(m-1) & 0 & \cdots & I \\ -A(m) & 0 & \cdots & 0 \end{bmatrix} \tag{E-21}$$

Thus,

$$FPF^H = \begin{bmatrix} -\sum_{k=1}^{m} A(k)R(k) & R(1) & R(2) & \cdots & R(m-1) \\ -\sum_{k=1}^{m} A(k)R(k-1) & R(0) & R(1) & \cdots & R(m-2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ -\sum_{k=1}^{m} A(k)R(k-m+1) & R(-m+2) & R(-m+3) & \cdots & R(0) \end{bmatrix} \tag{E-22}$$

Equations E-14 and E-15 yield Equation E-23:

$$FPF^H = \begin{bmatrix} R(0)-\Sigma & R(1) & R(2) & \cdots & R(m-1) \\ R(-1) & R(0) & R(1) & \cdots & R(m-2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ R(-m+1) & R(-m+2) & R(-m+3) & \cdots & R(0) \end{bmatrix} \tag{E-23}$$

Finally, one needs to calculate $K\Sigma K^H$ to complete the evaluation of the right hand side of Equation E-16:

$$K\Sigma K^H = \begin{bmatrix} I \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \Sigma \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} \tag{E-24}$$

$$K\,\Sigma K^H = \begin{bmatrix} \Sigma & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{E-25}$$

Therefore,

$$F\,P\,F^H + K\,\Sigma K^H = \begin{bmatrix} R(0) & R(1) & \cdots & R(m-1) \\ R(-1) & R(0) & \cdots & R(m-2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ R(-m+1) & R(-m+2) & \cdots & R(0) \end{bmatrix} \tag{E-26}$$

Equation E-26 shows that when the claimed value of P is substituted in Equation E-16, equality holds. This completes the proof of the lemma.

The block column vector Γ is used in the State Space Closed Form method to calculate covariance matrices with positive lags. The following lemma gives Γ explicitly:

Lemma E-2: Let Γ be defined by Equation E-27:

$$\Gamma = F\,P\,H + K\,\Sigma, \tag{E-27}$$

where F, H, K, P, and Σ are as in the state space representation of an AR process of order m. Then Γ is given by Equation E-28:

$$\Gamma = \begin{bmatrix} R(0) \\ R(-1) \\ \cdot \\ \cdot \\ \cdot \\ R(-m+1) \end{bmatrix}. \tag{E-28}$$

Proof: Using Equation E-20,

$$F\,P\,H = \begin{bmatrix} R(1) & R(2) & \cdots & R(m) \\ R(0) & R(1) & \cdots & R(m-1) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ R(-m+2) & R(-m+3) & \cdots & R(1) \end{bmatrix} \begin{bmatrix} -A(1) \\ -A(2) \\ \cdot \\ \cdot \\ \cdot \\ -A(m) \end{bmatrix} \tag{E-29}$$

$$F P H = \begin{bmatrix} -\sum_{k=1}^{m} A(k) R(k) \\ -\sum_{k=1}^{m} A(k) R(k-1) \\ \vdots \\ -\sum_{k=1}^{m} A(k) R(k-m+1) \end{bmatrix} \tag{E-30}$$

From Equations E-14 and E-15,

$$F P H = \begin{bmatrix} R(0) - \Sigma \\ R(-1) \\ \vdots \\ R(-m+1) \end{bmatrix} \tag{E-31}$$

Therefore,

$$F P H + K \Sigma = \begin{bmatrix} R(0) \\ R(-1) \\ \vdots \\ R(-m+1) \end{bmatrix} \tag{E-32}$$

So the lemma is proved.

Another useful lemma follows:

Lemma E-3:

$$F^{l-1} \Gamma = \begin{bmatrix} R(l-1) \\ R(l-2) \\ \vdots \\ R(l-m) \end{bmatrix}, \quad l = 1, 2, 3, \cdots \tag{E-33}$$

where F and Γ are as in Lemma E-2.

Proof: (by induction) The case $l = 1$ is a trivial corollary of Lemma E-2. Accordingly, assume the theorem for $l - 1$:

$$F^{l-2} \Gamma = \begin{bmatrix} R(l-2) \\ R(l-3) \\ \vdots \\ R(l-m-1) \end{bmatrix} \tag{E-34}$$

Consider the case of l:

$$F^{l-1}\Gamma = F\,(F^{l-2}\Gamma) \tag{E-35}$$

$$F^{l-1}\Gamma = \begin{bmatrix} -A^H(1) & -A^H(2) & \cdots & -A^H(m-1) & -A^H(m) \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} R(l-2) \\ R(l-3) \\ R(l-4) \\ \cdot \\ \cdot \\ \cdot \\ R(l-m-1) \end{bmatrix} \tag{E-36}$$

$$F^{l-1}\Gamma = \begin{bmatrix} -\sum_{k=1}^{m} A^H(k)R(l-1-k) \\ R(l-2) \\ \cdot \\ \cdot \\ \cdot \\ R(l-m) \end{bmatrix} \tag{E-37}$$

The lemma follows from Equation E-12.

The above lemmas allow a proof of the equivalence of the State Space Closed Form and AR Recursion Methods.

Theorem E-1: Let H, P, and Σ be as in the state space representation of an AR process. Then Equation E-38 holds:

$$R(0) = H^H P H + \Sigma \tag{E-38}$$

Proof:

$$H^H P = \begin{bmatrix} -A^H(1) & -A^H(2) & \cdots & -A^H(m) \end{bmatrix} \begin{bmatrix} R(0) & R(1) & \cdots & R(m-1) \\ R(-1) & R(0) & \cdots & R(m-2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R(-m+1) & R(-m+2) & \cdots & R(0) \end{bmatrix} \tag{E-39}$$

$$H^H P = \begin{bmatrix} -\sum_{k=1}^{m} A^H(k)R(1-k) & -\sum_{k=1}^{m} A^H(k)R(2-k) & \cdots & -\sum_{k=1}^{m} A^H(k)R(m-k) \end{bmatrix} \tag{E-40}$$

By Equation E-11,

$$H^H P = \begin{bmatrix} R(1) & R(2) & \cdots & R(m) \end{bmatrix} \tag{E-41}$$

Thus,

$$H^H P H = \begin{bmatrix} R(1) & R(2) & \cdots & R(m) \end{bmatrix} \begin{bmatrix} -A(1) \\ -A(2) \\ \cdot \\ \cdot \\ \cdot \\ -A(m) \end{bmatrix} \tag{E-42}$$

$$H^H P H = -\sum_{k=1}^{m} A(k) R(k) \tag{E-43}$$

By Equation E-14,

$$H^H P H = R(0) - \Sigma \tag{E-44}$$

Theorem 1 follows.

Theorem E-2: Let H, F, and Γ be as in the state space representation of an AR process. Then Equation E-45 holds:

$$R(l) = H^H F^{l-1} \Gamma, \quad l = 1, 2, 3, \cdots \tag{E-45}$$

Proof: Using Lemma E-3,

$$H^H F^{l-1} \Gamma = \begin{bmatrix} -A^H(1) & -A^H(2) & \cdots & -A^H(m) \end{bmatrix} \begin{bmatrix} R(l-1) \\ R(l-2) \\ \vdots \\ R(l-m) \end{bmatrix}, \quad l = 1, 2, 3, \cdots \tag{E-46}$$

Or,

$$H^H F^{l-1} \Gamma = -\sum_{k=}^{m} A^H(k) R(l-k), \quad l = 1, 2, 3, \cdots \tag{E-47}$$

By Equation E-12,

$$H^H F^{l-1} \Gamma = R(l), \quad l = 1, 2, 3, \cdots \tag{E-48}$$

The proof of Theorem 2 is complete. Theorems 1 and 2 state that the State Space Closed Form Method follows from the AR Recursion Method.

Appendix F.
AN EXAMPLE FOR THE STATE SPACE MODEL

A simple test case is presented here for the state space estimation algorithm. It illustrates the difficulties of testing, and presents some results that can be compared with other implementations. Consider a two-channel real Autoregressive (AR) process of order two. Consider the Gaussian autocorrelation shaping function (Equation 3.4-12) with the amplitude matrix:

$$\begin{bmatrix} 1.995 & 0.0 \\ 0.0 & 1.995 \end{bmatrix} \tag{F-1}$$

and the one-lag temporal correlation coefficient:

$$\begin{bmatrix} 0.9 & 0.0 \\ 0.0 & 0.9 \end{bmatrix}. \tag{F-2}$$

The lag matrix is the zero matrix, the Doppler frequency is 0.0, and the sample interval is chosen to be 0.01.

The shaping function determines the covariance matrix used to solve the Yule-Walker equations:

$$R = \begin{bmatrix} R(0) & R(1) & R(2) \\ R(-1) & R(0) & R(1) \\ R(-2) & R(-1) & R(0) \end{bmatrix} = \begin{bmatrix} 1.995 & 0.0 & 1.7955 & 0.0 & 1.3089 & 0.0 \\ 0.0 & 1.995 & 0.0 & 1.7955 & 0.0 & 1.3089 \\ 1.7955 & 0.0 & 1.995 & 0.0 & 1.7955 & 0.0 \\ 0.0 & 1.7955 & 0.0 & 1.995 & 0.0 & 1.7955 \\ 1.3089 & 0.0 & 1.7955 & 0.0 & 1.995 & 0.0 \\ 0.0 & 1.3089 & 0.0 & 1.7955 & 0.0 & 1.995 \end{bmatrix} \tag{F-3}$$

The AR parameters found by solving the Yule-Walker equations are given by Displays F-4 and F-5:

$$A(1) = \begin{bmatrix} -1.629 & 0.0 \\ 0.0 & -1.629 \end{bmatrix}, \tag{F-4}$$

$$A(2) = \begin{bmatrix} 0.81 & 0.0 \\ 0.0 & 0.81 \end{bmatrix}. \tag{F-5}$$

The driving noise covariance matrix is

$$\Sigma = \begin{bmatrix} 0.1303554 & 0.0 \\ 0.0 & 0.1303554 \end{bmatrix}. \tag{F-6}$$

F.1 True System State Space Parameters

Corresponding to the AR parameters are parameters for the innovations representation of the state space model. The system dynamics matrix is

$$F = \begin{bmatrix} 1.629 & 0.0 & -0.81 & 0.0 \\ 0.0 & 1.629 & 0.0 & -0.81 \\ 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \end{bmatrix}. \tag{F-7}$$

The Hermetian transpose of the observation matrix H is

$$H^H = \begin{bmatrix} 1.629 & 0.0 & -0.81 & 0.0 \\ 0.0 & 1.629 & 0.0 & -0.81 \end{bmatrix}. \quad \text{(F-8)}$$

The matrix Γ is

$$\Gamma = \begin{bmatrix} 1.995 & 0.0 \\ 0.0 & 1.995 \\ 1.7955 & 0.0 \\ 0.0 & 1.7955 \end{bmatrix}. \quad \text{(F-9)}$$

The Kalman gain matrix is

$$K = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}. \quad \text{(F-10)}$$

And the covariance matrix for the innovations is the covariance matrix for the driving noise in the AR model.

These true system parameters can be used for this test case to test any algorithm for estimating state space parameters. Such an algorithm may introduce a basis transformation of the state space, but the innovations covariance matrix should remain unchanged.

The AR parameters (and the state-space model parameters derived from them) are a minimum-mean-square fit to the matrix lags $R(0)$, $R(1)$, and $R(2)$ in Equation F-3. Thus, in order to have a numerically exact true model, the matrix lags $R(0)$, $R(1)$, and $R(2)$ must be recalculated using the state-space relations:

$$R(0) = H^H P H + \Sigma \quad \text{(F-11)}$$

$$R(m) = H^H F^{m-1} \Gamma, \; m = 1,2 \quad \text{(F-12)}$$

where P is the steady-state covariance matrix of the true system state vector. Matrix P is calculated as the solution to the following equation:

$$P = F P F^H + K \Sigma K^H. \quad \text{(F-13)}$$

For a simple low-order system, matrix P can be obtained analytically. For high-order systems a numerical recursion is preferred. The matrix lags generated in this manner correspond **exactly** with the state space model and the AR model.

F.2 The Scientific Studies Algorithm with Exact Order

This appendix presents step-by-step results of the Scientific Studies algorithm for estimating state space parameters. This test case shows the results when the upper bound on the matrix order of the state space model matches the actual value, two. The inputs to the algorithm are past block correlation matrix, the future block correlation matrix, the block Hankel matrix, and the column-shifted block Hankel matrix, shown in displays F-14, F-15, F-16, and F-17, respectively:

$$R_P = \begin{bmatrix} R(0) & R(1) \\ R(-1) & R(0) \end{bmatrix} = \begin{bmatrix} 1.995 & 0.0 & 1.7955 & 0.0 \\ 0.0 & 1.995 & 0.0 & 1.7955 \\ 1.7955 & 0.0 & 1.995 & 0.0 \\ 0.0 & 1.7955 & 0.0 & 1.995 \end{bmatrix} \tag{F-14}$$

$$R_F = \begin{bmatrix} R(0) & R(-1) \\ R(1) & R(0) \end{bmatrix} = \begin{bmatrix} 1.995 & 0.0 & 1.7955 & 0.0 \\ 0.0 & 1.995 & 0.0 & 1.7955 \\ 1.7955 & 0.0 & 1.995 & 0.0 \\ 0.0 & 1.7955 & 0.0 & 1.995 \end{bmatrix} \tag{F-15}$$

$$H_{LL} = \begin{bmatrix} R(1) & R(2) \\ R(2) & R(3) \end{bmatrix} = \begin{bmatrix} 1.7955 & 0.0 & 1.3089 & 0.0 \\ 0.0 & 1.7955 & 0.0 & 1.3089 \\ 1.3089 & 0.0 & 0.6778 & 0.0 \\ 0.0 & 1.3089 & 0.0 & 0.6778 \end{bmatrix} \tag{F-16}$$

$$\bar{H}_{LL} = \begin{bmatrix} R(2) & R(3) \\ R(3) & R(4) \end{bmatrix} = \begin{bmatrix} 1.3089 & 0.0 & 0.6778 & 0.0 \\ 0.0 & 1.3089 & 0.0 & 0.6778 \\ 0.6778 & 0.0 & 0.04403 & 0.0 \\ 0.0 & 0.6778 & 0.0 & 0.04403 \end{bmatrix} \tag{F-17}$$

Notice that in this example the past and future block correlation matrices are equal to each other. This is not true in general.

The higher lag covariance matrices $R(3)$ and $R(4)$ were found by the AR Recursion Method. These matrices could have been calculated equivalently using the Closed-Form Method based on the state-space parameters. For this example, they work out to be the least significant numerals in a difference, suggesting that nemerical difficulties can arise here as the order increases.

The singular value decomposition of the past and future block covariance matrices is given by display F-18:

$$\begin{bmatrix} \frac{1}{\sqrt{2}}I & \frac{1}{\sqrt{2}}I \\ \frac{1}{\sqrt{2}}I & -\frac{1}{\sqrt{2}}I \end{bmatrix} \begin{bmatrix} 3.7905I & 0.0 \\ 0.0 & 0.1995I \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}I & \frac{1}{\sqrt{2}}I \\ \frac{1}{\sqrt{2}}I & -\frac{1}{\sqrt{2}}I \end{bmatrix} = \begin{bmatrix} 1.995I & 1.7955I \\ 1.7955I & 1.995I \end{bmatrix} \tag{F-18}$$

Accordingly, the square root of the past and forward block covariance matrices are:

$$R_P^{1/2} = R_F^{1/2} = \begin{bmatrix} 1.196787I & 0.750133I \\ 0.750133I & 1.196787I \end{bmatrix} \tag{F-19}$$

The inverse of the square root of these block covariance matrices is:

$$R_P^{-1/2} = R_F^{-1/2} = \begin{bmatrix} 1.37625I & -0.86262I \\ -0.86262I & 1.37625I \end{bmatrix} \tag{F-20}$$

The matrix A is calculated as:

$$A = R_F^{-1/2} H_{LL} R_P^{-1/2} = \begin{bmatrix} 0.7973I & 0.5169I \\ 0.5169I & -0.4879I \end{bmatrix} \tag{F-21}$$

At least one significant digit is lost in some of the components of A. Since A is calculated

from the Hankel matrix, and numerical precision problems can arise in calculating the covariance sequence lags comprising the Hankel matrix, these numerical difficulties may be compounded. A Singular Value Decomposition of A is:

$$A = U_A S_A V_A^H = \begin{bmatrix} 0.9431I & -0.3323I \\ 0.3323I & 0.9432I \end{bmatrix} \begin{bmatrix} 0.9794I & 0 \\ 0 & 0.67I \end{bmatrix} \begin{bmatrix} 0.9431I & 0.3323I \\ 0.3323I & -0.9432I \end{bmatrix} \quad \text{(F-22)}$$

Equation F-22 shows the SVD used in the computer program and further calculations here.

The past and future transformation matrices are:

$$T_P = V_A^H R_P^{-1/2} = \begin{bmatrix} 1.0113I & -0.3562I \\ 1.2710I & -1.5847I \end{bmatrix} \quad \text{(F-23)}$$

$$T_F = U_A^H R_F^{-1/2} = \begin{bmatrix} 1.0113I & -0.3562I \\ -1.2710I & 1.5847I \end{bmatrix} \quad \text{(F-24)}$$

The matrix $\bar{\Delta}$ is:

$$\bar{\Delta} = T_F \bar{H} T_P^H = \begin{bmatrix} 0.8560I & 0.3142I \\ -0.3142I & 0.5054I \end{bmatrix} \quad \text{(F-25)}$$

The matrix Z_H is

$$Z_H = (H_{L,L} T_P^H)_{1:2,\,1:4} = \begin{bmatrix} 1.350I & 0.2079I \end{bmatrix} \quad \text{(F-26)}$$

The matrix Z_Γ is

$$Z_\Gamma = (T_F H_{L,L})_{1:4,\,1:2} = \begin{bmatrix} 1.3496I \\ -0.2079I \end{bmatrix} \quad \text{(F-27)}$$

These matrices allow one to finally determine the state space parameters. The system dynamics matrix is:

$$F = S_A^{-1/2} \bar{\Delta} S_A^{-1/2} = \begin{bmatrix} 0.874I & 0.388I \\ -0.388I & 0.754I \end{bmatrix} \quad \text{(F-28)}$$

The Hermetian transpose of the observations matrix is:

$$H^H = Z_H S_A^{-1/2} = \begin{bmatrix} 1.36I & 0.254I \end{bmatrix} \quad \text{(F-29)}$$

The backward observation matrix Γ is:

$$\Gamma = S_A^{-1/2} Z_\Gamma = \begin{bmatrix} 1.364I \\ -0.254I \end{bmatrix} \quad \text{(F-30)}$$

The state space correlation matrix is:

$$\Pi = S_A = \begin{bmatrix} 0.9794I & 0 \\ 0 & 0.67I \end{bmatrix} \quad \text{(F-31)}$$

The driving noise covariance matrix is:

$$\Omega = R(0) - H^H \Pi H = 0.1304I \quad \text{(F-32)}$$

The calculation of Ω results in the loss of one significant digit in this case. The Kalman gain matrix is:

$$K = (\Gamma - F\,\Pi H)\Omega^{-1} = \begin{bmatrix} 0.999I \\ 1.04I \end{bmatrix} \tag{F-33}$$

The state space algorithm has introduced a transformation of the basis for the state vector. An element of the state space associated with the model parameters in Equations F-7 through F-10 is transformed to the state space associated with the parameters in Equations F-28 through F-33 by premultiplying it by the matrix T, where

$$T = \begin{bmatrix} 0.999I & -0.315I \\ 1.04I & -1.50I \end{bmatrix} \tag{F-34}$$

F.3 The Scientific Studies Algorithm with Higher Order

This test case shows the results when the upper bound on the matrix order of the state space model exceeds the actual value. In this case, the upper bound is three, while the matrix order of the model is two. The inputs to the algorithm, the past block correlation matrix, the future block correlation matrix, the block Hankel matrix, and the column-shifted block Hankel matrix, are shown in displays F-35 through F-38:

$$R_P = \begin{bmatrix} R(0) & R(1) & R(2) \\ R(-1) & R(0) & R(1) \\ R(-2) & R(-1) & R(0) \end{bmatrix} = \begin{bmatrix} 1.995I & 1.7955I & 1.3089I \\ 1.7955I & 1.995I & 1.7955I \\ 1.3089I & 1.7955I & 1.995I \end{bmatrix} \tag{F-35}$$

$$R_F = \begin{bmatrix} R(0) & R(-1) & R(-2) \\ R(1) & R(0) & R(-1) \\ R(2) & R(1) & R(0) \end{bmatrix} = \begin{bmatrix} 1.995I & 1.7955I & 1.3089I \\ 1.7955I & 1.995I & 1.7955I \\ 1.3089I & 1.7955I & 1.995I \end{bmatrix} \tag{F-36}$$

$$H_{LL} = \begin{bmatrix} R(1) & R(2) & R(3) \\ R(2) & R(3) & R(4) \\ R(3) & R(4) & R(5) \end{bmatrix} = \begin{bmatrix} 1.7955I & 1.3089I & 0.6779I \\ 1.3089I & 0.6779I & 0.04403I \\ 0.6779I & 0.04403I & -0.4773I \end{bmatrix} \tag{F-37}$$

$$\bar{H}_{LL} = \begin{bmatrix} R(2) & R(3) & R(4) \\ R(3) & R(4) & R(5) \\ R(4) & R(5) & R(6) \end{bmatrix} = \begin{bmatrix} 1.3089I & 0.6779I & 0.04403I \\ 0.6779I & 0.04403I & -0.4773I \\ 0.04403I & -0.4773I & -0.8133I \end{bmatrix} \tag{F-38}$$

(More significant digits were used in working out this example than are shown.)

The singular value decomposition of the past and future block covariance matrices is:

$$\begin{bmatrix} 0.5589I & -0.7071I & 0.4331I \\ 0.6125I & 0I & -0.7904I \\ 0.5589I & 0.7071I & 0.4331I \end{bmatrix} \begin{bmatrix} 5.2717I & 0 & 0 \\ 0 & 0.6861I & 0 \\ 0 & 0 & 0.02725I \end{bmatrix} \begin{bmatrix} 0.5589I & 0.6125I & 0.5589I \\ -0.7071I & 0I & 0.7071I \\ 0.4331I & -0.7904I & 0.4331I \end{bmatrix}$$

$$= \begin{bmatrix} 1.995I & 1.7955I & 1.3089I \\ 1.7955I & 1.995I & 1.7955I \\ 1.3089I & 1.7955I & 1.995I \end{bmatrix} \tag{F-39}$$

Accordingly, the square roots of the past and forward block covariance matrices are:

$$R_P^{1/2} = R_F^{1/2} = \begin{bmatrix} 1.1624I & 0.7296I & 0.3341I \\ 0.7296I & 0.9646I & 0.7296I \\ 0.3341I & 0.7296I & 1.1624I \end{bmatrix} \tag{F-40}$$

The inverse of the square root of these block covariance matrices is:

$$R_P^{-1/2} = R_F^{-1/2} = \begin{bmatrix} 1.8761I & -1.9247I & 0.6688I \\ -1.9247I & 3.9480I & -1.9247I \\ 0.6688I & -1.9247I & 1.8761I \end{bmatrix} \quad \text{(F-41)}$$

The matrix A is calculated as:

$$A = R_F^{-1/2} H_{LL} R_P^{-1/2} = \begin{bmatrix} 0.7522I & 0.4829I & 0.7522I \\ 0.4829I & -0.0902I & -0.2647I \\ 0.2272I & -0.2647I & -0.3525I \end{bmatrix} \quad \text{(F-42)}$$

Some of the elements of A in the corresponding computer print out are only accurate to four significant figures, already having lost some precision. An SVD of A is

$$\begin{bmatrix} 0.9157I & 0.3211I & -0.2415I \\ 0.3941I & -0.6015I & 0.6949I \\ 0.07792I & -0.7315I & -0.6774I \end{bmatrix} \begin{bmatrix} 0.9794I & 0 & 0 \\ 0 & 0.6699I & 0 \\ 0 & 0 & 3.1x10^{-6}I \end{bmatrix} \begin{bmatrix} 0.9157I & 0.3941I & 0.0779I \\ -0.3211I & 0.6015I & 0.7315I \\ 0.2415I & -0.6949I & 0.6774I \end{bmatrix} \quad \text{(F-43)}$$

The past and future transformation matrices are:

$$T_P = V_A^H R_P^{-1/2} = \begin{bmatrix} 1.0115I & -0.3564I & -5.6x10^{-7} \\ -1.2709I & -1.5847I & 4.4x10^{-6} \\ 2.2435I & -4.5119 & 2.7697 \end{bmatrix} \quad \text{(F-44)}$$

$$T_F = U_A^H R_F^{-1/2} = \begin{bmatrix} 1.0115I & -0.3564I & -5.6x10^{-7} \\ 1.2709I & -1.5847I & 4.4x10^{-6} \\ -2.2435I & 4.5119 & -2.7697 \end{bmatrix} \quad \text{(F-45)}$$

The matrix Z is

$$Z = \begin{bmatrix} 0.8560I & -0.3139I \\ 0.3139I & 0.5057I \end{bmatrix} \quad \text{(F-46)}$$

The matrix Z_H is:

$$Z_H = \begin{bmatrix} 1.3496I & -0.2076I \end{bmatrix} \quad \text{(F-47)}$$

The matrix Z_Γ is:

$$Z_\Gamma = \begin{bmatrix} 1.3496I & 0.2076I \end{bmatrix} \quad \text{(F-48)}$$

The system dynamics matrix is:

$$F = S_A^{-1/2} Z S_A^{-1/2} = \begin{bmatrix} 0.8740I & -0.3874I \\ 0.3874I & 0.7550I \end{bmatrix} \quad \text{(F-49)}$$

The Hermetian transpose of the observations matrix is:

$$H^H = Z_H S_A^{-1/2} = \begin{bmatrix} 1.3638I & -0.2537I \end{bmatrix} \quad \text{(F-50)}$$

The backward observation matrix Γ is:

$$\Gamma = S_A^{-1/2} Z_\Gamma = \begin{bmatrix} 1.3638I \\ 0.2537I \end{bmatrix} \quad \text{(F-51)}$$

The state space correlation matrix is:

$$\Pi = S_A = \begin{bmatrix} 0.9794I & 0 \\ 0 & 0.6699I \end{bmatrix} \quad \text{(F-52)}$$

The driving noise covariance matrix is:

$$\Omega = R(0) - H^H \Pi H = 0.1304 I \tag{F-53}$$

The Kalman gain matrix is:

$$K = (\Gamma - F \Pi H)\Omega^{-1} = \begin{bmatrix} 1.0010 I \\ -1.0402 I \end{bmatrix} \tag{F-54}$$

These values are close to the values obtaining when the upper limit on the state space order matches the actual order.

*U.S. GOVERNMENT PRINTING OFFICE: 1996-710-126-20144