1982-06

Systematic experimental determination of discrete-time models for nonlinear systems.

Mandelberg, Martin

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SYSTEMATIC EXPERIMENTAL DETERMINATION OF DISCRETE-TIME MODELS FOR NONLINEAR SYSTEMS

by

Martin Mandelberg

June 1982

Thesis Advisor: R. W. Hamming

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Systematic Experimental Determination of Discrete-Time Models for Nonlinear Systems

5. TYPE OF REPORT & PERIOD COVERED
Doctoral Dissertation
June 1982

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Martin Mandelberg

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE
June 1982

13. NUMBER OF PAGES
264

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

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

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Modeling, Performance Modeling, Linear Systems, Nonlinear Systems, Least Squares Error Minimization, Search Indicators, Search Indicator Growth Algorithm

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
Techniques are presented for experimentally computing discrete-time model equations from a finite set of samples observations of the system inputs and outputs. Existing modeling techniques typically consider simple model forms, and often make limiting assumptions and simplifications for mathematical convenience. This research extends these techniques to efficiently obtain a more accurate model

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

equation. Four key points are examined: (1) form of the model equation, (2) choice of the error minimization technique, (3) efficiency of model determination and evaluation algorithms, and (4) interpretation of the obtained model equations in typical applications.

A new algorithm for efficient model determination, the Search Indicator Growth Algorithm, is presented. This iterative algorithm efficiently evaluates a set of model terms and eliminates the undesired terms. The technique produces more accurate and robust model equations, and offers significant computational advantages over existing techniques. Computer simulated experiments illustrate the effectiveness of this method.

Approved for public release; distribution unlimited

Systematic Experimental Determination of Discrete-Time Models for Nonlinear Systems

by

Martin Mandelberg
B.S., Drexel Institute of Technology, 1969
M.S., The University of Connecticut, 1973
M.S., United States Naval War College, 1976

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

from the

NAVAL POSTGRADUATE SCHOOL
June 1982

ABSTRACT

Techniques are presented for experimentally computing discrete-time model equations from a finite set of sampled observations of the system inputs and outputs. Existing modeling techniques typically consider simple model forms, and often make limiting assumptions and simplifications for mathematical convenience. This research extends these techniques to efficiently obtain a more accurate model equation. Four key points are examined: (1) form of the model equation, (2) choice of the error minimization technique, (3) efficiency of model determination and evaluation algorithms, and (4) interpretation of the obtained model equations in typical applications.

A new algorithm for efficient model determination, the Search Indicator Growth Algorithm, is presented. This iterative algorithm efficiently evaluates a set of model terms and eliminates the undesired terms. The technique produces more accurate and robust model equations, and offers significant computational advantages over existing techniques. Computer simulated experiments illustrate the effectiveness of this method.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I am grateful to my advisor, Professor Richard W. Hamming, for his guidance, encouragement of my research, and his assistance in helping me to focus my efforts. I would like to thank all the members of my doctoral committee for following this lengthy dissertation effort, and contributing their time, knowledge, and suggestions.

I also wish to acknowledge financial assistance provided by a Department of the Navy Long Term Graduate Fellowship from the Naval Underwater Systems Center (NUSC). Additional research support was provided by NUSC, the Office of Naval Research (ONR), and my current employer, the Coast Guard Research and Development Center (R&DC). Particular mention is given to the contributions of Dr. William Von Winkle of NUSC, Captain Stacy Holmes and Dr. Joel Morris, formerly of ONR, and Commander Allen E. Rolland of R&DC.

Finally, I wish to thank my wife, Lois, and my family for their patience, understanding, and countless forms of support. The completion of this thesis would not have been possible without their love, and their encouragement to persevere.

I. INTRODUCTION

A. PRESENTATION OF THE RESEARCH PROBLEM

This research examines the problem of experimentally developing discrete-time model equations to represent, or approximate, the input-output behavior of both linear and nonlinear systems based on a finite set of sampled observations of the system inputs and outputs.

The traditional approach to the modeling problem involves selecting a particular model form, estimating the unknown coefficients of the model from the observations, and finally verifying the quality of the model. The particular model form is commonly chosen for mathematical convenience or from some physical understanding of the structure of the system. Given a specific model equation to represent a system, a number of techniques exist for estimating the values of the model coefficients that minimize a function of the fitting error between the model and the system.

We are interested in developing the techniques needed to obtain a suitable model when the underlying structure of the system is unknown. With a suitable model, a variety of current applications can be approached, including the detection and evaluation of failures that affect system performance. The problem is how to obtain a useful model from the available observations of the system.

There are four key parts of this modeling problem. First, the allowed functional forms for the model equation must be sufficiently general to permit adequate approximation of the behavior of systems of interest without requiring an unmanageable number of terms. (The adequacy of an approximation is an application dependent consideration and will be discussed later.) For most nonlinear systems of interest, the discrete-time model form typically used in the literature is the Volterra series model. This form does not adequately satisfy the accuracy or compact representation requirements except for a very restricted class of systems.

The second key part is the determination of the best error minimization technique for use in evaluating any model equation. This becomes an area of concern in terms of both accuracy and computational efficiency when we are faced with finite length data sequences and measurement noise.

The third key part is the development of a general technique for evolving or "growing" a model equation in a computationally efficient and accurate manner. Existing techniques for approaching this part of the problem are very limited as to the functional model form they can handle, and often make somewhat artificial assumptions and approximations to obtain even a partial solution. The result is typically an inferior model of the system, with insufficient prediction accuracy or an excessive number of terms in the model equation.

The fourth part is determining if the obtained model is the "best" model, in some sense, for use in a particular application. This last point involves an investigation of whether any obtained model equation is a preferred representation of the system, or just one model from a set of functionally equivalent models. We examine each of these four areas in this research and extend the existing techniques for developing model equations.

Most researchers have approached modeling from the coefficient estimation perspective, and there is a large body of literature on techniques for efficiently estimating the values of the coefficients of specific models once the model form has been chosen. Levinson [Ref. 1], Durbin [Ref. 2], Robinson [Ref. 3], and Morf [Ref. 4 and 5] have developed computationally efficient "recursive-in-order" algorithms for iteratively estimating the coefficient values of certain linear models. Recently, Lee [Ref. 6], Friedlander [Ref. 7], Perry [Ref. 8], and Parker and Perry [Ref. 9] have reported on extensions of these algorithms that estimate the coefficient values of a wider class of model forms. These techniques are shown to be inadequate for the more general problem of an unknown system form.

We approach the modeling problem from a different perspective, that of systematically growing a model that minimizes the error residual signal (performance modeling), rather than just estimating the coefficients of an arbitrary

model form. We have shifted our concern to performance modeling since we are really interested in the behavior of the system, and not the values of the coefficients of one particular approximation of it. This may seem at first to be a relatively minor difference in approach, but the development in the following chapters has uncovered some new capabilities for more efficient model determination.

B. DISCUSSION

Modern systems are both dynamic and complex, yet they generally work by cause and effect, i.e. the set of inputs operating on the system produces a set of outputs. Knowledge of this relationship between input and output, enables us to approach various practical applications. We can predict reaction based on the action applied, control the output by modification of the input, adapt the behavioral characteristics so that a given stimulus will result in a desired response, diagnose the cause by observing the effect, and detect and evaluate changes (failures) in a system's performance.

For failure detection applications, it is necessary to have some standard or reference by which to make the determination that a failure has occurred. An often used concept provides redundancy in the form of one or more additional systems (or subsystems) operating in parallel with the original system, compares the various outputs, and uses an appropriate criterion to detect a failed system.

It is not feasible to have redundant systems in many applications so a simulation is used, such as a mathematical model that approximates the system's performance characteristics. In some cases, this model can be designed from a detailed knowledge of both the structure and components of the system [Ref. 10]. Because extensive detail is typically required for creating this type of model, we refer to this as "microscopic modeling". But there is often insufficient knowledge of the system structure and/or component behavior of real world systems, or the computational complexity is excessive, and an alternate modeling technique is needed.

One concept used by earlier researchers [Ref. 11 and 12] selects a specific model form and uses it to characterize, or approximate, the performance of the system from input and output measurements of the system. This concept is referred to as "input-output" or "macroscopic" modeling, and the approximation must be done in some meaningful sense if it is to be useful. The choice of this mathematical form determines both the quality of the model (extent to which it approximates the system behavior) and the meaning of the model coefficient estimates. If the mathematical model is an exact or equivalent representation of the system, there is a correspondence between the set of system parameters and the set of model coefficients. Various properties of the model and the error residual (difference between the system

and realized model outputs under identical input conditions) can be used in applications including failure detection and evaluation. The characteristics of the measured input signal, and of any measurement noise, has a direct effect on the model performance[1].

Macroscopic characterization also involves model building (growing), which adds terms to a given model equation to better fit the observed data. If a model fit is not adequate for an application, then the standard technique in the literature [Ref. 11] guesses at a "better" model and fits to it the same data. This "brute-force" technique makes little use of the specific features of the unacceptable model, and blindly continues until (hopefully) an adequate fit is obtained.

This technique makes some physical and practical sense when dealing with a simple model form corresponding to the known form of the system. Examples include linear transfer functions (ARMA models) and static polynomials. In these cases, each successively larger model provides a better fit and the preceding fit can be considered as a reduced order

---

1 Two interesting cases regarding the input signal are; (1) we have little or no control over the characteristics of the input signal, and (2) the input signal (or probe) is under our complete control for the system characterization process. For the work that follows, we consider case (2) with the assumption that the input measurements are representative of the normal system operation. Section E of Chapter VI examines both of these situations. The impact of measurement noise is also addressed in Chapter III.

model of the system. This technique is of dubious value when the form of the system is unknown. We need to develop alternative growth techniques that are more useful in the general case.

A particular recursive algorithm was introduced by Levinson [Ref. 1], and adapted by Durbin [Ref. 2], to efficiently obtain the solution of a specially structured set of linear equations. In using this algorithm, a crucial simplification was often made by earlier researchers [Ref. 4-9, 13 and 14]. By limiting the form of the model and assuming that the input sampled data sequence is ergodic, a special mathematical structure can be induced into the model evaluation equations, and exploited to save a significant amount of mathematical computation. This ergodic assumption has been rationalized from different points of view, and has resulted in various related evaluation techniques. The simplification will be examined in detail in Chapter III, since its true effects do not appear clearly in the literature. While the simplification appears reasonable in the isolated context in which it was made, it will be shown that the net effect of model evaluation using this simplification is typically that the obtained model is suboptimal in prediction performance.

Even without these assumptions, the recursive solution algorithm has limited usefulness. While feasible for typical linear model forms, it is shown that the recursive

algorithm rapidly becomes computationally prohibitive when considering general nonlinear models. Alternative model growth techniques are therefore needed.

It is often overlooked that from the same input sequence, different system equations can give the same output sequence. The result is indistinguishable performance characteristics from structurally different systems, or equivalently, multiple different (and often independent) model equations each adequately describing the performance characteristics of a single system.

This point will be discussed further because it has implications for the fault detection and evaluation application. Using the integer n as the discrete time index, the system input sequence is denoted as {u(n)}, the output sequence as {y(n)}, and the residual sequence due to inaccuracies in the model equation as {e(n)}. Assume a suitable model has been obtained and is used in conjunction with the real system as shown below.

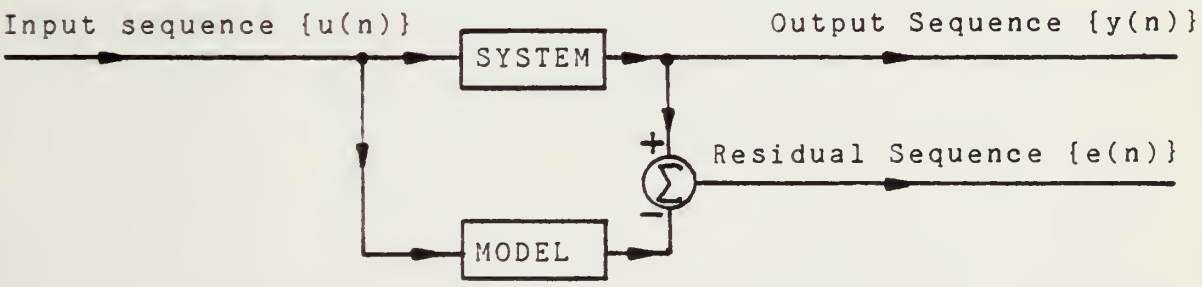


Figure 1: Configuration for Fault Detection

A significant change in the characteristics of the error residual sequence {e(n)} could be used to indicate that there has been a change in the behavior of the system (fault detection). There are applications where we would like to uniquely determine the change in the system (fault evaluation). This last capability requires that there exist a unique one-to-one relationship between the value of the system parameter that changed, and the resulting value of a model coefficient. This is obviously not possible when there are two or more structurally different but equivalently performing model equations.

The existence of a model equation that exactly describes a finite set of input-output measurements does not imply any uniqueness properties [Ref. 15]. This can be demonstrated by considering particular examples of structurally different but equivalently performing models for a given input.

Example 1.1: The time series of measurements $\{u(n);n=0,1,2,3,\ldots,7,\ldots\} = \{1,1,3,7,17,41,99,239,\ldots\}$ and $\{y(n);n=0,1,2,3,\ldots,7,\ldots\} = \{0,1,2,5,12,29,70,169,\ldots\}$ are equivalently described over any interval with $n>1$ by both of the linearly independent model equations;

$$y(n) = u(n) - y(n-1) \qquad \{1.1\}$$

and

$$y(n) = u(n-1) + y(n-1) \qquad \{1.2\}$$

Example 1.2: The time series of measurements

{u(n);n=0,1,2,...,7,...} = {1,1,-1,2,-5,10,-22,47,...} and

{y(n);n=0,1,2,...,7,...} = {0,1,0,2,-3,7,-15,32,...} are

equivalently described over any interval with n>2 by both of

the linearly independent model equations;

$$y(n) = u(n) + y(n-1) \qquad \{1.3\}$$

and

$$y(n) = u(n-2) - y(n-1) + y(n-2) \qquad \{1.4\}$$

Example 1.3: The equation

$$y(n) = .9u(n) - .5u(n-1)y(n-1) \qquad \{1.5\}$$

and the equation

$$y(n) = .9u(n) - .45u(n-1)u(n-1) + .25u(n-1)u(n-2)y(n-2) \qquad \{1.6\}$$

can be realized as shown in Figures 2 and 3 respectively.

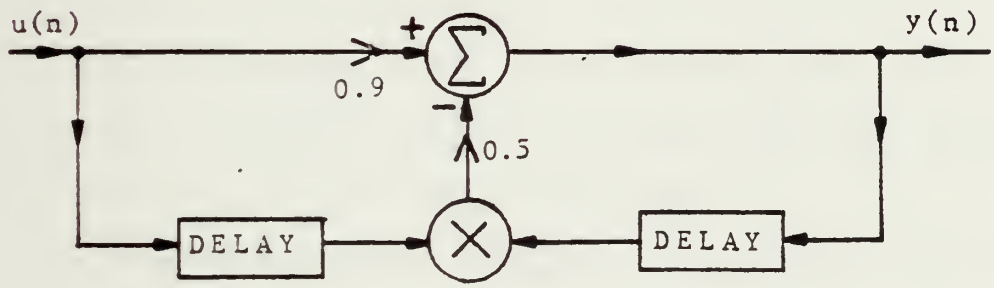


Figure 2: A block diagram realization of Equation {1.5}

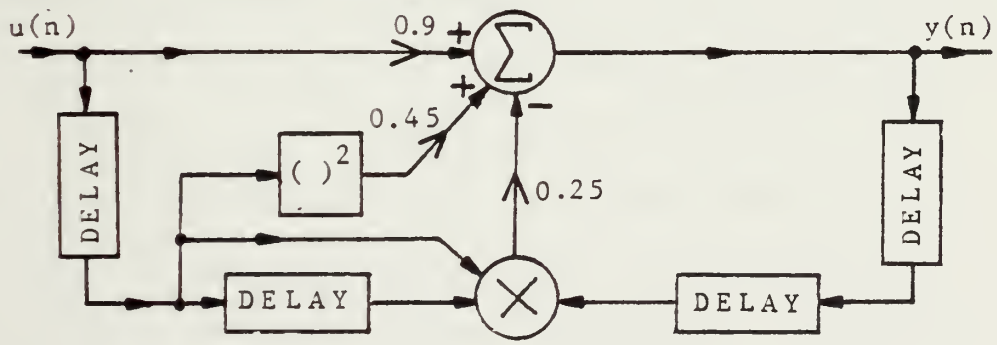


Figure 3: A block diagram realization of Equation {1.6}

These two different equations will produce the identical output sequence {y(n);S<=n<=T} for any given input sequence {u(n);S<=n<=T}. Note that {1.5} and {1.6} are not independent since {1.6} can be obtained from {1.5} directly by the following recursion.

Start by replacing n by n-1 in {1.5}.

$$y(n-1) = .9u(n-1) - .5u(n-2)y(n-2) \qquad \{1.7\}$$

Substituting {1.7} into the right side of {1.5} yields:

$$y(n) = .9u(n) - .5u(n-1)\,[.9u(n-1) - .5u(n-2)y(n-2)]$$

$$= .9u(n) - .45u(n-1)u(n-1) + .25u(n-1)u(n-2)y(n-2) \qquad \{1.8\}$$

Since {1.8} is the same as {1.6}, therefore {1.5} and {1.6} are not independent. There could be the case where {1.5} was the system and {1.6} was the model, or vice-versa. A change in one system parameter would not be uniquely related to a resulting change in one model coefficient. This also explains why an experimentally obtained model may explain predictability, but not cause and effect.

Measurement data of a physical system can be matched by more than one model equation for a number of reasons. It may be by coincidence; the input may keep the output of each model exactly the same. It was recently shown [Ref. 16] that if the system is linear, the existence of linearly independent models that match the input-output data can be attributed to the particular inputs that generate the measurements from which these independently equivalent models are computed.

The preceding discussion shows that the following problems must be considered. If we obtain a set of models each acceptably describing the input-output performance of a system, can we determine if any of these models include the structural properties of the actual system that produced the data? Naturally, we should expect that we will rarely be able to obtain a model with the same detailed structure as the system. The second problem is how to determine the "best" model for a particular application from those candidate models in the equivalence set. Chapter VIII will address these problems and present some new results in the context of a particular application.

C. OVERVIEW

Chapter II provides a review of existing general linear and nonlinear model forms and presents an extension to a unifying general model that we will use. Chapter III presents various error minimization techniques for evaluating candidate model equations and proves the generally superior modeling performance of one particular technique known as the "Covariance" least squares method, over the least squares technique commonly used in the literature. Additive measurement noise is examined, and new expressions are developed for the resulting distortion of the fitting error and of the coefficient estimates of linear recursive models.

Chapter IV discusses the equations required to evaluate the performance of a model as more terms are included, and presents a recursive technique for evaluating the solution of a large and useful class of equations. The computational advantages of this technique are compared with the direct least squares evaluation of each model.

Chapter V discusses the existing model growth techniques based on the parameter estimation approach. It shows that the recursive technique of Chapter IV reduces to the commonly used parameter estimation technique known as Levinson's Algorithm when two restrictive assumptions are made. The use of these assumptions typically produced inferior models compared to those produced by the more general technique. Chapter V also discusses several possible nonlinear model growth algorithms that are logical extensions of existing linear growth techniques.

Chapter VI presents a new concept in iterative model growth based on the developments in the preceding chapters, and analyzes the advantages and limitations. This heuristic technique is shown to offer significant improvements over the existing and previously discussed methods. Chapter VII gives the results of computer simulations and real world experimental comparisons of the model growth techniques developed in this research.

Chapter VIII examines three additional applications for the modeling methods developed in this thesis. They are

fault detection, fault evaluation, and reduced order modeling. Specific techniques are discussed and a number of concepts are proposed. Conclusions are given on the key results of this work and areas for future research are outlined.

II. CHOICE OF THE MODEL EQUATION FORM

A. EXISTING MODEL FORMS

We are concerned with the determination of discrete-time models for both linear and nonlinear systems. The logical starting point is a discussion of existing linear model forms. We limit discussion to single-input, single-output systems for simplicity, but using a vector notation, the results can be directly extended to the multiple-input, multiple-output case. We also limit consideration to models whose input-output relationships can be described by time-invariant difference equations[2].

After a discussion of linear models, the few general dynamic nonlinear models forms found in the literature are presented. A more general nonlinear model form that subsumes the preceding linear and nonlinear models is then discussed. One particular version of this general form is then developed in greater detail, and utilized in the remainder of this work.

---

2 It is recognized that there are systems where an input-output relationship cannot be exactly described by a difference equation. Consider a discrete quantizer whose output y(n) at any instant n is equal to the integral part of the input u(n) if the fractional part of u(n) is less than 0.5, and is equal to 1 plus the integral part of u(n) if the fractional part of u(n) is greater than or equal to 0.5. Clearly the input-output relationship of such a system cannot be accurately described by a difference equation.

Linear dynamic discrete models include the moving average (MA), autoregressive (AR), and autoregressive-moving average (ARMA) forms. They are well described in the literature [Ref. 11, 17, and 18] and are briefly discussed here for completeness.

The MA model predicts the current value of the output of a system as a weighted summation of the current and q consecutive preceding values of the system input, where q is known as the order (or memory) of the MA model. Following the previously used convention, the sampled observations of the system input are denoted as {u(n);S<=n<=T}, the system output as {y(n);S<=n<=T}, and the residual error due to inaccuracies in the model as {e(n);S<=n<=T}. The model equation can be written as;

$$y(n) = a^{(q)}(0)\, u(n) + a^{(q)}(1)\, u(n-1) + \ldots + a^{(q)}(q)\, u(n-q) + e(n)$$

$$= \sum_{i=0}^{q} a^{(q)}(i)\, u(n-i) + e(n) \qquad \{2.1\}$$

The coefficients are the $a^{(q)}(i)$ factors that multiply each corresponding $(i)^{th}$ delayed input term. The (q) superscript is used to emphasize the dependency of each coefficient value on the order of the model, and the superscript notation is dropped when no ambiguity results.

These models are called moving average because the current output is a weighted average of a finite "window"

passing over the present and past input values. Models of the form of Eq. {2.1} are denoted as MA(q).

The p^{th} order AR model predicts the current value of the system output as a weighted summation of p consecutive preceding output values. Using similar notation, the equation for this model is written as;

$$y(n) = b^{(p)}(1)\, y(n-1) + b^{(p)}(2)\, y(n-2) + \ldots + b^{(p)}(p)\, y(n-p) + e(n)$$

$$= \sum_{j=1}^{p} b^{(p)}(j)\, y(n-j) + e(n) \qquad \{2.2\}$$

The coefficients are the $b^{(p)}(j)$ factors that multiply each corresponding $(j)^{th}$ delayed output term. Models of the form of Eq. {2.2} are denoted as AR(p).

Despite their simple form, MA(q) and AR(p) modeling of even simple linear systems often requires an excessively large number of model terms (a high order model). A natural extension of these two models is a combination of both. Such mixed models are called autoregressive-moving average, or ARMA, models of orders p and q, and are often written as ARMA(p,q). The ARMA model predicts the current value of the system output as a weighted summation of the current value of the system input, q consecutive preceding values of the system input, and p consecutive preceding values of the system output. Combining the previous notations produces the model equation;

$$y(n) = \alpha^{(q)}(0)\, u(n) + \alpha^{(q)}(1)\, u(n-1) + \ldots + \alpha^{(q)}(q)\, u(n-q)$$

$$+ \beta^{(p)}(1)\, y(n-1) + \beta^{(p)}(2)\, y(n-2) + \ldots + \beta^{(p)}(p)\, y(n-p) + e(n)$$

$$= \sum_{i=0}^{q} \alpha^{(q)}(i)\, u(n-i) + \sum_{j=1}^{p} \beta^{(p)}(j)\, y(n-j) + e(n) \qquad \{2.3\}$$

The coefficients are the $\alpha^{(q)}(i)$ and $\beta^{(p)}(j)$ factors that multiply each corresponding $(i)^{th}$ delayed input term and $(j)^{th}$ delayed output term, respectively. Real world linear systems typically include feedback, and can be adequately modeled with the smallest number of terms by an ARMA model.

The literature discusses two general dynamic discrete-time nonlinear models. The Volterra mkdel [Ref. 19, 20 and 21] can be thought of as a nonlinear generalization of the MA model. This model predicts the current value of the system output as a linearly weighted summation of increasing degree products of the current and m consecutive preceding input values. Using the typical notation followed in the literature as a guide, the equation for this model form is written as;

$$y(n) = \sum_{k=1}^{d} f_k[u(n-i); i=0,1,2,\ldots,m] + e(n) \qquad \{2.4\}$$

where

$$f_k = \sum_{g_1=0}^{m} \sum_{g_2=g_1}^{m} \cdots \sum_{g_k=g_{k-1}}^{m} a_k(g_1, g_2, \ldots, g_k) \prod_{i=1}^{k} u(n-g_i) \qquad \{2.5\}$$

and $a_k(g_1, g_2, \ldots, g_k)$ is called the k^{th} degree Volterra kernel.

When the degree d equals 1, Eq. {2.4} reduces to the form of the MA model.

The Volterra model is based only on a sum of products of past and present input values. Because of this nonrecursive form, the Volterra model is unable to compactly represent a system that includes significant feedback. A model of the form of Eq. {2.4} is denoted as VOL(d,m). The lower limits of the summations in Eq. {2.5} were purposely chosen to eliminate redundant terms, and therefore we have minimized the number of equations that need to be solved in the evaluation of any particular VOL(d,m). The upper summation limits of Eq. {2.5} are all set equal to the integer m for notational clarity at this point. We could, of course, use a more general notational convention for the upper summation limits (e.g. m ; i=1,2,...). A more general upper summation limit notation would produce more complexity in the equations, and offers no specific advantages for the problem examined in this thesis. The Volterra model of a system may not require all of these terms indicated by Eq. {2.5}.

The Bilinear model [Ref. 22, 23, 24 and 25] predicts the current output of the system as a linearly weighted summation of the current and m consecutive preceding input values, plus a linearly weighted term composed of the product of one of m preceding output values with the current or one of m preceding input values. Using the typical notation found in the literature, the equation for this model form is written as;

$$y(n) = \sum_{i=0}^{m} a^{(m)}(i)\, u(n-1)$$

$$+ \sum_{i=0}^{m} \sum_{j=1}^{m} c^{(m)}(i,j)\, u(n-i)y(n-j) + e(n) \qquad \{2.6\}$$

This form includes bilinear terms composed of the products of specific input and output factors, a feature not found in the previously discussed model forms. However, it is limited to the one type of nonlinear form shown above. Models of the form of Eq. {2.6} are denoted as BIL(m). . The restriction to equal upper summation limits is again used for clarity. The Bilinear model of a system may not require all of these terms indicated by Eq. {2.6}.

B. DEVELOPMENT OF A MORE GENERAL MODEL FORM

When used for the modeling of a typical nonlinear system, the Volterra model form suffers from the same limitations as the MA model form does for linear systems. The existence of any feedback in a system will usually result in the requirement for the order m to be very large in Eq. {2.1} in the linear case, or in Eq. {2.4} in the nonlinear case. This property of nonrecursive model forms results in the need for an unacceptably large number of model terms to adequately represent the behavior of typical systems.

A natural extension of the Volterra model is to include Volterra-like terms of the output of the system, in a manner similar to the relationship between the MA and the ARMA models. An investigation of the effect of feedback in some common nonlinear systems leads to the conclusion that it is also useful to include terms that are extensions of the Bilinear model form. Such an extension has been made and some partial results concerning different versions of this new model form have been published [Ref. 9, 26 and 27].

One version of this model form was called the Nonlinear ARMA model in references 9 and 26. To better distinguish the properties of a more general form of the model, it is denoted as the Bivariate Volterra Model (BVM) in reference 27 and in the work that follows.

The coefficient notation of the previously discussed linear and nonlinear models forms follows the conventions found in the literature. Considerable thought was given to the need for suitable notation for the more general and complex model form, and also for the developments that follow. It was decided to have a uniform coefficient notation that could be applied to any of the models of this chapter. Following is a compact equation for BVM(d,m), a BVM of degree d and memory m in terms of this uniform coefficient notation.

$$y(n) = \sum_{r=1}^{d} \left[\sum_{g_1=0}^{m} \sum_{g_2=g_1}^{m} \cdots \sum_{g_r=g_{r-1}}^{m} \theta_{r;0}(g_1,g_2,\ldots,g_r) \prod_{i=1}^{r} u(n-g_i) \right]$$

$$+ \sum_{s=1}^{d} \left[\sum_{h_1=0}^{m} \sum_{h_2=h_1}^{m} \cdots \sum_{h_s=h_{s-1}}^{m} \theta_{0;s}(h_1,h_2,\ldots,h_s) \prod_{j=1}^{s} y(n-h_j) \right]$$

$$+ \sum_{r=1}^{d-1} \sum_{s=1}^{d-r} \left[\sum_{g_1=0}^{m} \sum_{g_2=g_1}^{m} \cdots \sum_{g_r=g_{r-1}}^{m} \sum_{h_1=1}^{m} \sum_{h_2=h_1}^{m} \cdots \sum_{h_s=h_{s-1}}^{m} \right.$$

$$\left. \theta_{r;s}(g_1,\ldots,g_r;h_1,\ldots,h_s) \prod_{i=1}^{r} u(n-g_i) \prod_{j=1}^{s} y(n-h_j) \right] + e(n) \qquad \{2.7\}$$

The coefficients are the factors starting with $\theta_{r;0}$, $\theta_{0;s}$, and $\theta_{r;s}$ in Eq. {2.7}. Note that two subscripts and one or more parameters in parenthesis are included for each coefficient. A $\theta_{r;0}$ coefficient is used in conjunction with a term composed exclusively of r products of past and present input factors. Likewise, a $\theta_{0;s}$ coefficient is used in conjunction with a term composed exclusively of s products of past output values. The subscript in each of these cases distinguishes the number of such factors in the corresponding model term. Finally, we use $\theta_{r;s}$ for the coefficient of the model term composed of r input factors and s output factors. In all cases, the parameters in parenthesis in each model coefficient distinguish the particular lag factors.

A model of this general form can be completely described by either specifying a particular degree d and memory m, or by just providing the distinguishing parameters and

subscripts of each desired coefficient. Following is an example of a second degree, first order BVM equation which is denoted as BVM(2,1).

$$\begin{aligned} y(n) = {} & \theta_{1;0}(0)\, u(n) + \theta_{1;0}(1)\, u(n-1) + \theta_{2;0}(0,0)\, u(n)^2 \\ & + \theta_{2;0}(0,1)\, u(n)u(n-1) + \theta_{2;0}(1,1)\, u(n-1)^2 \\ & + \theta_{1;1}(0,1)\, u(n)y(n-1) + \theta_{1;1}(1,1)u(n-1)y(n-1) \\ & + \theta_{0;2}(1)\, y(n-1)^2 + \theta_{0;2}(1,1)\, y(n-1)^2 + e(n) \end{aligned} \qquad \{2.8\}$$

This coefficient notation completely specifies the model terms, as demonstrates in the following examples.

$\theta_{3;0}(1,2,3)$ is the coefficient of term $u(n-1)u(n-2)u(n-3)$

$\theta_{2;1}(0,1,1)$ is the coefficient of term $u(n)u(n-1)y(n-1)$

The choice of the various lower summation limits in Eq. {2.7} eliminates redundant model terms. The upper summation limits are set equal to m for notational clarity, as was done for the VOL and BIL forms. Because the upper summation limits of Eq. {2.7} were all set equal to m in the preceding pages, a compact expression for the number of coefficients in a full BVM of degree d and memory m can be written as;

$$c(d,m) = \sum_{r=1}^{d} \frac{\prod_{i=1}^{r}(m+i)}{r!} + \sum_{s=1}^{d} \frac{\prod_{j=1}^{s}(m-1+j)}{s!} + \sum_{r=1}^{d-1} \sum_{s=1}^{d-r} \frac{\prod_{i=1}^{r}(m+i) \prod_{j=1}^{s}(m-1+j)}{r!\, s!} \qquad \{2.9\}$$

This equation is used in subsequent chapters when the computational complexity of evaluating this full model form is considered.

The BVM form defined in Eq. {2.7} is limited to the summation of products of integer powers of past and present input terms, and past output terms. Other functional forms besides sums of integer products are possible, but this form appears to be most tractable for our modeling. Examination of Eq. {2.1} through Eq. {2.6} confirms that the BVM form subsumes the AR, MA, ARMA, VOL, and BIL model forms.

For example, an ARMA(p,q) is subsumed by a BVM(1,m) when the degree d of the BVM is set equal to 1, and m = maximum (p,q). This only allows terms with descriptive coefficients $\Theta_{1;0}(i)$ or $\Theta_{0;1}(j)$; where $0<=i<=m$ and $1<=j<=m$. This includes all the terms of an ARMA(p,q).

The BVM form is emphasized because it includes the other general forms discussed in the chapter and commonly used in the literature. Using this BVM form rather than the ARMA, VOL or BIL forms, typically produces a more compact and accurate representation of nonlinear systems with feedback. Chapter VII contains several computer simulated experiments demonstrating this point.

III. EXAMINATION OF ERROR MINIMIZATION TECHNIQUES

A. DISCUSSION

Assume we are given any particular model form linear in some finite number c of unknown coefficients, and a set of input-output data of length N>c. We are interested in determining the particular model equation that, in some meaningful sense, best approximates the performance of the system that produced the N output measurements from the corresponding N input measurements. There are different error criteria, including least squares and minimax, that could be used in minimizing the error residual. Least squares techniques minimize the average squared residual sequence value in a given interval, while minimax techniques minimize the maximum absolute value of the residual sequence in the interval.

Difficulties with least squares, including degraded modeling performance under noisy conditions, have been reported [Ref. 13]. Nevertheless, it has been decided to investigate the use of least squares techniques for the following reasons; (1) least squares minimization for models linear in the coefficients leads to a set of tractable linear equations in the unknowns, (2) there exists a large body of parameter estimation techniques in the literature based on least squares, and it is possible to extend some of these for our model growth and evaluation problem.

This chapter presents both the theoretical differences and results of computer simulated experimental comparison of various least squares formulations when applied to systems characterization. In the simulation study, two criteria used for comparison purposes are: (1) average squared residual value (fitting error), and (2) accuracy in estimating model coefficient values. An examination of the effect of additive output noise is presented in the last section.

An example of a linear-in-the-coefficients nonlinear difference equation using the coefficient notation introduced in Chapter II is;

$$y(n) = \theta_{1;0}(1)\, u(n) + \theta_{0;2}(1,1)\, y(n-1)y(n-1) + \theta_{1;1}(0,1)\, u(n)y(n-1) \qquad \{3.1\}$$

Eq. {3.1} contains both linear and nonlinear terms in the input u(n) and output y(n). Since the coefficients $\theta_{r;s}$ all enter in a linear fashion, this equation can be expressed in compact vector notation (all vectors in this thesis are column vectors). Defining a coefficient vector $\underline{\theta}$, where

$$\underline{\theta}^T = [\, \theta_{1;0}(1),\ \theta_{0;2}(1,1),\ \theta_{1;1}(0,1)] \qquad \{3.2\}$$

and a term vector $\underline{x}(n)$, where

$$\underline{x}(n)^T = [u(n-1), y(n-1)y(n-1), u(n)y(n-1)] \qquad \{3.3\}$$

Eq. {3.1} can now be expressed in the vector form;

$$y(n) = \underline{\theta}^T \underline{x}(n) \qquad \{3.4\}$$

Assume that we are given a finite set of measurements of the input sequence {u(n);S<=n<=T} and the corresponding output sequence {y(n);S<=n<=T} of a time-invariant and

causal system of unknown structure. To reproduce the input-output behavior of this system within some moderately small error, we choose a linear-in-the-unknown coefficients predictor model equation of the following form.

$$y(n) = \underline{\theta}^T \underline{x}(n) + e(n) \qquad \{3.5\}$$

where e(n) is the equation error of the model at time n; $\underline{\theta}$ is a vector of length c containing as yet unknown coefficients corresponding to each model term; and $\underline{x}(n)$ is a vector of c model terms, each of which is formed from the product of a finite number of input and output factors from the set:

$$[u(n-m),u(n-m+1),\ldots,u(n-1),u(n),y(n-m),y(n-m+1),\ldots,y(n-1)] \qquad \{3.6\}$$

where m is some finite number called the memory (or order) of the model. The maximum number of input factors or output factors in any such product combination will be called the degree d of the model. Note that the above description fits the BVM(d,m) introduced in Chapter II.

The following example is used repeatedly for illustration. Consider a linear MA model with q = m = 2, and the coefficient and term vectors;

$$\underline{\theta}^T = [\theta_{1;0}(0), \theta_{1;0}(1), \theta_{1;0}(2)] \qquad \{3.7\}$$

$$\underline{x}(n)^T = [u(n),u(n-1),u(n-2)] \qquad \{3.8\}$$

This model form is linear in the unknown coefficients once the $\underline{x}(n)$ is specified, and we choose to minimize the following nonnegative least squares error criterion;

$$J^2 = \frac{1}{(n_3-n_2+1)} \sum_{n=n_2}^{n_3} e(n)^2 = \frac{1}{N} \sum_{n=n_2}^{n_3} e(n)^2 \qquad \{3.9\}$$

where n_2 and n_3 take on fixed integer values.

To carry out the least squares fit in compact vector-matrix notation, we use underscored lower case letters to represent vectors and capital letters to represent matrices. Scalars are represented with lower case letters whenever possible, but occasionally capital letters are used.

Define the output vector $\underline{y}$, where

$$\underline{y}^T = [y(n_2), y(n_2+1), \ldots, y(n_3)] \qquad \{3.10\}$$

and the data measurement matrix X, where

$$X^T = \left[\underline{x}(n_2) \quad \underline{x}(n_2+1) \quad \ldots \quad \underline{x}(n_3)\right] \qquad \{3.11\}$$

Substituting {3.5}, {3.7}, {3.10}, and {3.11} into {3.9} yields;

$$J^2 = \frac{1}{N} (\underline{y} - X^T\underline{\theta})^T(\underline{y} - X^T\underline{\theta}) = \frac{1}{N}\left[\underline{y}^T\underline{y} - \underline{\theta}^T X^T\underline{y} - \underline{y}^T X\underline{\theta} + \underline{\theta}^T X^T X\underline{\theta}\right] \qquad \{3.12\}$$

Following standard least squares theory [Ref. 11 - 14], the evaluation equation for the coefficient vector $\underline{\theta}$ that minimizes Eq. {3.12}, and the corresponding value of the minimum error criterion are determined. The details of the well known least squares derivation are included for notational development. Differentiating Eq. {3.12} with respect to the vector $\underline{\theta}$ using matrix calculus and equating the result to zero yields;

$$\frac{\partial J^2}{\partial \underline{\theta}} = 0 = -\frac{2}{N} X^T\underline{y} + \frac{2}{N} X^T X\underline{\theta} \qquad \{3.13\}$$

Simplifying Eq. {3.13} produces;

$$\frac{1}{N} X^T X\underline{\theta} = \frac{1}{N} X^T\underline{y} \qquad \{3.14\}$$

It is convenient to use the following compact notation for this set of c simultaneous linear equations.

$$R\underline{\theta} = \underline{r} \qquad \{3.15\}$$

where $$R = \frac{1}{N} X^T X \qquad \{3.16\}$$

is the positive semi-definite least squares matrix of size c, and

$$\underline{r} = \frac{1}{N} X^T \underline{y} \qquad \{3.17\}$$

is a column vector of size c. The factor 1/N is retained in these definitions for subsequent distinction.

To insure that Eq. {3.13} represents a unique minimum, the second derivative of J^2 with respect to $\underline{\theta}$ must be positive. Applying this result to Eq. {3.12} yields the added condition that;

$$\frac{\partial^2 J^2}{\partial \underline{\theta}^T \partial \underline{\theta}} = \frac{2}{N} \left[X^T X \right] = 2R > 0 \qquad \{3.18\}$$

Equations {3.15} are known as the normal equations, and there is a unique solution if and only if matrix R is positive definite. This unique solution is;

$$\underline{\theta} = R^{-1} \underline{r} \qquad \{3.19\}$$

Using {3.16}, {3.17}, and {3.19} with {3.12}, the minimum value of J^2 is;

$$J^2 = \frac{1}{N} \underline{y}^T \underline{y} - \underline{r} R^{-1} \underline{r}$$

$$= \frac{1}{N} \underline{y}^T \underline{y} - \underline{r}^T \underline{\theta} \qquad \{3.20\}$$

We now investigate how the effect of the choice of the values of n_2 and n_3 in Eq. {3.9}, changes the resulting value of the error criterion J^2. The main purpose of this investigation is to identify the reasons for the observed numerical differences in the results of various least squares formulations that commonly are used in the literature. Many recent researchers [Ref. 1 - 9, 13, 14, and 18 - 29] put emphasis on computational simplicity and make assumptions or approximations related to the values of n_2 and n_3 that induce special structure into the solution equations {3.19} for $\underline{\theta}$. We consider a number of distinct cases that are discussed but not clearly compared in the literature.

(1) If $n_2<S+m$ and $n_3<=T$, this is equivalent to the assumption that $u(n)=0$ for $n<S$ and $y(n)=0$ for $n<S$, and is known in the literature as the "Prewindowed" case [Ref. 4, 6, 28 and 29].

(2) If $n_2>=S+m$ and $n_3>T$, this is equivalent to the assumption that $u(n)=0$ for $n>T$ and $y(n)=0$ for $n>T$, is known in the literature as the "Postwindowed" case [Ref. 29], and is seldom used.

(3) If $n_2<S+m$ and $n_3>T$, we get both prewindowing and postwindowing since this is equivalent to assuming that $u(n)=0$ for $n<S$ and $n>T$, and also that $y(n)=0$ for $n<S$ and $n>T$. This is equivalent to rectangularly windowing the measurements. It is

known as the "Autocorrelation" case [Ref. 7, 14, 25, and 28 - 30], and is the typically used method.

(4) If $n_2 >= S+m$ and $n_3 <= T$, no window is applied to the observed measurements, and the so called "Covariance" case is realized [Ref. 28, 29 and 31].

Depending upon the specific choice of n_2 and n_3, there are many different least squares error criterion values $J^2(n_2,n_3)$, and related model coefficient estimates $\underline{\theta}(n_2,n_3)$, for a given set of input-output data. The literature typically reports the use of the Autocorrelation method for statistical considerations and because this can often lead to an efficient solution algorithm. This point is discussed further in a subsequent section.

Examination of these four different methods from the unifying framework of the least squares equation {3.9}, reveals an interesting comparison basis for explaining the subsequent differences in form and performance. This development does not appear in the systems identification literature and clearly indicates which error minimization method should be used for the performance modeling approach to the general model growth problem. The main result is that the Covariance method generally gives superior modeling results in terms of lower fitting error J^2 and more accurate model coefficients in the vector $\underline{\theta}$. The differences in these four methods are described analytically in terms of the following example, generalized in the theorem that

follows, and finally demonstrated in a computer simulated experiment.

EXAMPLE: Let S = 1 and T = 10. Then we have the data

$$\{u(n)\} = \{u(1), u(2), u(3), \ldots, u(10)\} \qquad \{3.21\}$$

$$\{y(n)\} = \{y(1), y(2), y(3), \ldots, y(10)\} \qquad \{3.22\}$$

Let the model be given by the equation:

$$y(n) = \theta_{1;0}(0)\, u(n) + \theta_{1;0}(1)\, u(n-1) + \theta_{1;0}(2)\, u(n-2) + e(n) \qquad \{3.23\}$$

Using least squares

$$J^2 = \frac{1}{N} \sum_{n=n_2}^{n_3} e(n)^2 \qquad \{3.24\}$$

where $N = n_3 - n_2 + 1$, and where the coefficient vector is;

$$\underline{\theta}^T = [\, \theta_{1;0}(0),\ \theta_{1;0}(1),\ \theta_{1;0}(2)\,] \qquad \{3.25\}$$

leads to

$$\frac{1}{N} [\, X^T X\,]\, \underline{\theta} = \frac{1}{N} X^T \underline{y} \qquad \{3.26\}$$

where

$$\underline{y}^T = [y(1), y(2), y(3), \ldots, y(10)] \qquad \{3.27\}$$

and X is the N x 3 data matrix involving {u(n)}, whose contents depends upon the choice of n_2 and n_3 as shown in the following four cases.

Case 1: Prewindowed method

Here $n_2 = S = 1$, $n_3 = T = 10$, and the data matrix becomes;

$$X = \begin{bmatrix} u(1) & 0 & 0 \\ u(2) & u(1) & 0 \\ u(3) & u(2) & u(1) \\ u(4) & u(3) & u(2) \\ u(5) & u(4) & u(3) \\ u(6) & u(5) & u(4) \\ u(7) & u(6) & u(5) \\ u(8) & u(7) & u(6) \\ u(9) & u(8) & u(7) \\ u(10) & u(9) & u(8) \end{bmatrix} \qquad \{3.28\}$$

The solution of the normal equations {3.26} is now given by;

$$\begin{bmatrix} \theta_{1;0}(0) \\ \hline \theta_{1;0}(1) \\ \hline \theta_{1;0}(2) \end{bmatrix} = \left[\begin{array}{c|c|c} \frac{1}{10}\sum_{n=1}^{10} u(n)^2 & \frac{1}{10}\sum_{n=2}^{10} u(n)u(n-1) & \frac{1}{10}\sum_{n=3}^{10} u(n)u(n-2) \\ \hline & \frac{1}{10}\sum_{n=2}^{10} u(n-1)^2 & \frac{1}{10}\sum_{n=3}^{10} u(n-1)u(n-2) \\ \hline \text{SYMMETRIC} & & \frac{1}{10}\sum_{n=3}^{10} u(n-2)^2 \end{array}\right]^{-1}$$

$$\begin{bmatrix} \frac{1}{10}\sum_{n=1}^{10} u(n)y(n) \\ \hline \frac{1}{10}\sum_{n=2}^{10} u(n-1)y(n) \\ \hline \frac{1}{10}\sum_{n=3}^{10} u(n-2)y(n) \end{bmatrix} \qquad \{3.29\}$$

Note that the square matrix in Eq. {3.29} has different summation limits along each diagonal parallel to the main diagonal.

Case 2: Postwindowed method

Here $n_2 = S+m = 1+2 = 3$, $n_3 = T+m = 10+2 = 12$, and the data matrix becomes;

$$X = \begin{bmatrix} u(3) & u(2) & u(1) \\ u(4) & u(3) & u(2) \\ u(5) & u(4) & u(3) \\ u(6) & u(5) & u(4) \\ u(7) & u(6) & u(5) \\ u(8) & u(7) & u(6) \\ u(9) & u(8) & u(7) \\ u(10) & u(9) & u(8) \\ 0 & u(10) & u(9) \\ 0 & 0 & u(10) \end{bmatrix} \qquad \{3.30\}$$

The solution of the normal equations {3.26} is now given by;

$$\begin{bmatrix} \theta_{1;0}(0) \\ \theta_{1;0}(1) \\ \theta_{1;0}(2) \end{bmatrix} = \begin{bmatrix} \frac{1}{10}\sum_{n=3}^{10} u(n)^2 & \frac{1}{10}\sum_{n=3}^{10} u(n)u(n-1) & \frac{1}{10}\sum_{n=3}^{10} u(n)u(n-2) \\ & \frac{1}{10}\sum_{n=2}^{10} u(n)^2 & \frac{1}{10}\sum_{n=2}^{10} u(n)u(n-1) \\ \text{SYMMETRIC} & & \frac{1}{10}\sum_{n=1}^{10} u(n)^2 \end{bmatrix}^{-1}$$

$$\begin{bmatrix} \frac{1}{10}\sum_{n=3}^{10} u(n)y(n) \\ \frac{1}{10}\sum_{n=3}^{10} u(n-1)y(n) \\ \frac{1}{10}\sum_{n=3}^{10} u(n-2)y(n) \end{bmatrix} \qquad \{3.31\}$$

Note that the square matrix in Eq. {3.31} has a different set of summation limits along each diagonal parallel to the main diagonal.

Case 3: Autocorrelation method

Here $n_2 = S = 1$, $n_3 = T+m = 10+2 = 12$, and the data matrix becomes.

$$X = \begin{bmatrix} u(1) & 0 & 0 \\ u(2) & u(1) & 0 \\ u(3) & u(2) & u(1) \\ u(4) & u(3) & u(2) \\ u(5) & u(4) & u(3) \\ u(6) & u(5) & u(4) \\ u(7) & u(6) & u(5) \\ u(8) & u(7) & u(6) \\ u(9) & u(8) & u(7) \\ u(10) & u(9) & u(8) \\ 0 & u(10) & u(9) \\ 0 & 0 & u(10) \end{bmatrix} \qquad \{3.32\}$$

The solution of the normal equations {3.26} is now given by;

$$\begin{bmatrix} \theta_{1;0}(0) \\ \theta_{1;0}(1) \\ \theta_{1;0}(2) \end{bmatrix} = \begin{bmatrix} \frac{1}{12}\sum_{n=1}^{10} u(n)^2 & \frac{1}{12}\sum_{n=2}^{10} u(n)u(n-1) & \frac{1}{12}\sum_{n=3}^{10} u(n)u(n-2) \\ & \frac{1}{12}\sum_{n=1}^{10} u(n)^2 & \frac{1}{12}\sum_{n=2}^{10} u(n)u(n-1) \\ \text{SYMMETRIC} & & \frac{1}{12}\sum_{n=1}^{10} u(n)^2 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{12}\sum_{n=1}^{10} u(n)y(n) \\ \frac{1}{12}\sum_{n=2}^{10} u(n-1)y(n) \\ \frac{1}{12}\sum_{n=3}^{10} u(n-2)y(n) \end{bmatrix} \qquad \{3.33\}$$

Note that the square matrix in Eq. {3.33} is symmetric, Toeplitz (equal values along every diagonal parallel to the main diagonal), and the summation limits are all the same along any diagonal parallel to the main diagonal.

The particular structure of the symmetric Toeplitz matrix in Eq. {3.33} was developed here strictly from a consideration of the error minimization limits. The literature contains numerous references to least squares matrices with this special structure, but it is usually just stated or developed along different lines[3]. After presentation of the fourth case, we will discuss the implications of each.

---

3 For example, Baheti [Ref. 23 and 24], Hsia [Ref. 14], and Iserman [Ref. 13] all utilize what they call "correlation analysis" where they assume that the input and output sequences are ergodic, such that this special Toeplitz structure results. This "ergodic assumption" can be described mathematically as follows [Ref. 14, pp. 44]. Consider a finite length discrete-time sequence of measurements of some signal denoted by {s(n)}. If this is a representative sample of an ergodic process, then the following condition will hold. The value obtained from the expression;

$$\frac{1}{N+1} \sum_{n=i}^{i+N} s(n)s(n-j)$$

is invariant with respect to the integer i. If this special condition holds, or is assumed, then the Toeplitz structure of Eq. {3.33} will result because of the relationships;

$$\frac{1}{N+1} \sum_{n=i}^{i+N} s(n)s(n-j) = \frac{1}{N+1} \sum_{n=i}^{i+N} s(n+1)s(n+1-j) = \frac{1}{N+1} \sum_{n=1+1}^{i+N+1} s(n)s(n-j)$$

$$= \frac{1}{N+1} \sum_{n=i}^{i+N} s(n+2)s(n+2-j) = \frac{1}{N+1} \sum_{n=i+2}^{i+N+2} s(n)s(n-j)$$

Case 4: Covariance method

Here $n_2 = S+m = 1+2 = 3$, $n_3 = T = 10$, and the data matrix becomes;

$$X = \begin{bmatrix} u(3) & u(2) & u(1) \\ u(4) & u(3) & u(2) \\ u(5) & u(4) & u(3) \\ u(6) & u(5) & u(4) \\ u(7) & u(6) & u(5) \\ u(8) & u(7) & u(6) \\ u(9) & u(8) & u(7) \\ u(10) & u(9) & u(8) \end{bmatrix} \qquad \{3.34\}$$

The solution of the normal equations {3.26} is now given by;

$$\begin{bmatrix} \theta_{1;0}(0) \\ \hline \theta_{1;0}(1) \\ \hline \theta_{1;0}(2) \end{bmatrix} = \left[\begin{array}{c|c|c} \frac{1}{8}\sum_{n=3}^{10} u(n)^2 & \frac{1}{8}\sum_{n=3}^{10} u(n)u(n-1) & \frac{1}{8}\sum_{n=3}^{10} u(n)u(n-2) \\ \hline & \frac{1}{8}\sum_{n=3}^{10} u(n-1)^2 & \frac{1}{8}\sum_{n=3}^{10} u(n-1)u(n-2) \\ \hline \text{SYMMETRIC} & & \frac{1}{8}\sum_{n=3}^{10} u(n-2)^2 \end{array}\right]^{-1}$$

$$\begin{bmatrix} \frac{1}{8}\sum_{n=3}^{10} u(n)y(n) \\ \hline \frac{1}{8}\sum_{n=3}^{10} u(n-1)y(n) \\ \hline \frac{1}{8}\sum_{n=3}^{10} u(n-2)y(n) \end{bmatrix} \qquad \{3.35\}$$

Note that the square matrix in Eq. {3.35} is symmetric but not Toeplitz, and the summation limits are all the same.

The main reason for the preceding four-case development is to point out the specific condition under which the least squares matrix becomes Toeplitz. This property is exploited in Levinson's algorithm [Ref. 1], which solves the normal equations with order of complexity proportional to the size

of the least squares matrix R, rather than proportional to the cube of the size of this matrix as occurs in the other three cases shown. Details of this algorithm are discussed later in Chapter V and Appendix B. For models other than simple moving average, other researchers [Ref. 3 - 9 and 21 - 25] constrained their model forms and used the Autocorrelation method to form least squares matrices with Toeplitz principle submatrices, and therefore gain some computational advantage when solving these equations using variations of Levinson's Algorithm. This chapter proves that this technique is cumbersome, unnecessary, and more importantly, generally produces inferior models compared to those obtained by the Covariance method.

The constrained Autocorrelation method models are inferior in two main ways: (1) Only specifically related sets of model terms necessary for the special Toeplitz structure are allowed in the model. This limits model growth flexibility, increases the computational burden, and degrades the model performance. Further discussion on these points is given in Chapter V. (2) The particular choice of data interval described by the Autocorrelation least squares method (or its statistical equivalent) typically produces a model with significantly higher fitting error, and substantially larger coefficient error than the Covariance method. This key point is discussed in more detail in the next two sections.

B. A THEOREM DESCRIBING THE CONDITION FOR SUPERIOR PERFORMANCE OF THE COVARIANCE METHOD

The four previous methods use exactly the same form of computation; they differ only in the specific data measurements used. The Prewindowed, Postwindowed, and Autocorrelation methods supply missing zeros, either before or after the measured data, or both. Thus these methods are arithmetically equivalent to the Covariance method operating on a discontinuous function, and it is well known that it is hard for least squares or any other minimization method to handle discontinuous functions. An alternate explanation is that the first three methods utilize constraints on the data values, and it is generally found that a constrained solution is inferior to the optimum (minimum valued) solution. This suggests that the Prewindowed, Postwindowed, and Autocorrelation methods would be inferior to the Covariance method.

Simple computer simulation experiments given in the next section confirm this reasoning. It remains, therefore, to mathematically express this feeling and these results that supplying missing data by a run of zeros is a poor method to use. We are, of course, concerned with the quality of the fit, the sum of the squares of the residuals.

The first step is to examine under what circumstances the Prewindowed, Postwindowed, or Autocorrelation methods would produce a lower average error than the Covariance method. Some mathematical notation is needed.

Consider a finite set of dynamic input observations $\{u(n);S<=n<=T\}$ and corresponding output observations $\{y(n);S<=n<=T\}$ of some system, and a linear-in-the-coefficients model equation relating the present value of y(n) to functions of past values of y(n) and present and past values of u(n). Denote the integer m as the maximum discrete lag (order) of term of the model equation, and $\underline{\theta}$ as the coefficient vector. The model equation can be written:

$$y(n)=f[\underline{\theta},u(n-i),y(n-j);i=0,1,2,\ldots,m;j=1,2,\ldots,m] +e(n) \quad \{3.36\}$$

Let $\{e_1(n)\}$ represent the error residual and J_1^2 represent the average squared error obtained when a least squares minimization is performed over the interval (n_2,n_3).

$$J_1^2 = \frac{1}{N_1} \sum_{n=n_2}^{n_3} e_1(n)^2 \quad \{3.37\}$$

$$\text{where } n_2 = S+m \quad \{3.38\}$$

$$\text{and } n_3 = T \quad \{3.39\}$$

$$\text{and } N_1 = n_3 - n_2 + 1 \quad \{3.40\}$$

Let the length of the error minimization interval be increased by a small amount $N_2>0$ to a larger region (n_1,n_4) where $n_1<n_2$ and/or $n_4>n_3$. This new region includes the first region and available data points on either or both sides of the first region. Missing data points in the new data matrix X are padded with zero values. Using the same model form, perform a least squares minimization over the interval (n_1,n_4). Denote the new error residual as $\{e_2(n)\}$ and the average least squares error as J_2^2.

$$J_2^2 = \frac{1}{N_1+N_2} \sum_{n=n_2}^{n_3} e_2(n)^2$$

$$= \frac{1}{N_1+N_2}\left[\sum_{n=n_1}^{n_2-1} e_2(n)^2 + \sum_{n=n_2}^{n_3} e_2(n)^2 + \sum_{n=n_3+1}^{n_4} e_2(n)^2 \right] \quad \{3.41\}$$

where $N_2 = (n_4-n_1+1) - N_1$ {3.42}

Since J_1^2 is the least squares fit over (n_2,n_3), it must be less than or equal to the quantity;

$$\frac{1}{N_1} \sum_{n=n_2}^{n_3} e_1(n)^2 \quad \{3.43\}$$

Let E^2 be the nonnegative value representing this loss of fit.

$$E^2 = \frac{1}{N_1} \sum_{n=n_2}^{n_3} \left[e_2(n)^2 - e_1(n)^2 \right] \quad \{3.44\}$$

THEOREM 1:

A necessary and sufficient condition for

$$J_2^2 < J_1^2 \quad \{3.45\}$$

is that the following condition must be met:

$$\frac{1}{N_2}\left[\sum_{n=n_1}^{n_2-1} e_2(n)^2 + \sum_{n=n_3+1}^{n_4} e_2(n)^2 \right] < \frac{1}{N_1} \sum_{n=n_2}^{n_3} e_2(n)^2 - \left[\frac{N_1}{N_2} + 1 \right] E^2 \quad \{3.46\}$$

PROOF:

Substituting {3.37}, {3.41}, and {3.44} into {3.45} yields:

$$\frac{1}{N_1+N_2}\left[\sum_{n=n_1}^{n_2-1} e_2(n)^2 + \sum_{n=n_2}^{n_3} e_2(n)^2 + \sum_{n=n_3+1}^{n_4} e_2(n)^2 \right] < \frac{1}{N_1} \sum_{n=n_2}^{n_3} e_2(n)^2 - E^2 \quad \{3.47\}$$

Multiply by N_1+N_2 and transpose the middle term from the left.

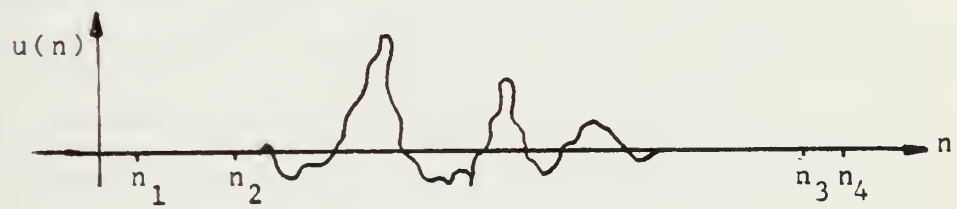
$$\sum_{n=n_1}^{n_2-1} e_2(n)^2 + \sum_{n=n_3+1}^{n_4} e_2(n)^2 < \left[\frac{N_1+N_2}{N_1} - 1 \right] \sum_{n=n_2}^{n_3} e_2(n)^2 - \left[N_1+N_2 \right] E^2 \quad \{3.48\}$$

Dividing both sides by N yields our desired condition.

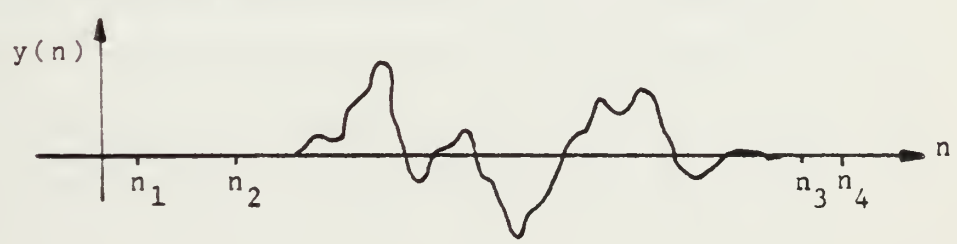
$$\frac{1}{N_2}\left[\sum_{n=n_1}^{n_2-1} e_2(n)^2 + \sum_{n=n_3+1}^{n_4} e_2(n)^2\right] < \frac{1}{N_1}\sum_{n=n_2}^{n_3} e_2(n)^2 - \left[\frac{N_1}{N_2} + 1\right] E^2 \qquad \{3.46\}$$

It is logical to ask how the condition of Eq. {3.46} could arise. The condition states that the average fit over the added end regions must be less than the average fit over the middle region minus the last term on the right side. Since $N_1 >> N_2$, the last term on the right side of Eq. {3.46} would be significant, and the error of the end regions must be much smaller than the average error over (n_2, n_3) for Eq. {3.46} to hold. Two obvious special cases can arise that satisfy the condition of Eq. {3.46}.

[1] In the case where the forced zero-valued data points in the expanded region correspond to the actual input sequence and the natural system dynamics (and no noise), then it follows that E^2 in {3.44} will be zero. As an example, consider a stable system excited by the following waveform {u(n)}.



The output {y(n)} might have a similar shape.



In the preceding figure, the expanded data region contains actual zero-valued input-output values, $e_2(n)$ will be zero in the expanded regions, and $J_2^2 <= J_1^2$.

[2] If the first region (n_2,n_3) contains data values that don't exactly fit the model equation, and the additional measurements in the larger region (n_1,n_4) happened to contain data that exactly, or almost exactly fit the model equation, then the average error over the larger region could be lower.

Both of these special cases are possible, but it appears highly unlikely that either will occur in practice. The special requirements on the data sequences for these cases are examples of pathological situations. The probability of their occurrence is so small as not to be meaningful.

The value of Theorem 1 resides not in the elegance of a mathematical proof, but because its proof is so simple and its meaning so important. Theorem 1 basically proves that any least squares error minimization method other than the Covariance method, will produce a higher average fitting error in all but unlikely pathological cases. Therefore, any systems characterization or parameter estimation technique based on a least squares minimization method different than the Covariance method (e.g. Prewindowed, Postwindowed, Autocorrelation, etc) will generally produce suboptimal fitting error results. This result is important for the work that follows, since it is well known that

approximations made early in certain recursive algorithms often grow and lead to significant errors later on, and we want to use a recursive algorithm to efficiently evaluate the model growth.

The next section provides some computer simulated experimental verification of the results of this section. Other factors affecting the accuracy of systems characterization and parameter estimation are also examined.

C. SIMULATION EXPERIMENTS

Experiment 1:

DESCRIPTION: An investigation of the effects of various least squares error methods, and the length of the observed data [S<=n<=T], on the accuracy of the characterization of known typical linear and nonlinear systems.

CRITERION: Square root of the average sum squared fitting error, J. Note that we minimize J^2 but examine J. This is done for clarity of graphical presentation.

For the first part of the experiment, we synthesize the MA(3) system;

$$y(n) = 1.0u(n) + .8u(n-1) + .6u(n-2) - .3u(n-3) \qquad \{3.49\}$$

The following Test Procedure is used repeatedly.

TEST PROCEDURE:

Generate a random sequence for {u(n)}, uniformly distributed between the amplitude limits [-5,5], and start this input through the system (with zero initial conditions)

at discrete time n=1. Record the observations {u(n);S<=n<=T} and {y(n);S<=n<=T} for S and T specified below, and use them to minimize the least square equation error of {3.12}. Examine the use of the Prewindowed (P), Postwindowed (W), Autocorrelation (A), and Covariance (C) methods. The value of S is chosen to be 11, and T varies from 50 to 1000 in steps of 50. The experiment is carried out over an ensemble of ten (10) runs, with different, but equivalently distributed, random input sequences. For the data obtained from the ensemble of ten runs, plot the maximum, minimum, and average value of J, as a function of the value of T and of the choice of minimization method.

For the second part of the experiment, synthesize the following ARMA(2,2) system and repeat the test procedure.

$$y(n) = 1.0u(n) + .8u(n-1) + .6u(n-2) - .9y(n-1) - .7y(n-2) \quad \{3.50\}$$

For the third part of the experiment, synthesize the following BVM system and repeat the test procedure.

$$\begin{aligned} y(n) = {} & 1.0u(n) + .8u(n-1) + .6u(n-2) - .9y(n-1) - .7y(n-2) \\ & + .2u(n)u(n) + .15u(n-1)u(n-4) - .3y(n-2)y(n-4) \\ & - .16\,u(n-1)y(n-1) + .05u(n-2)y(n-4) \end{aligned} \quad \{3.51\}$$

Figures 4 through 7 present the maximum, minimum, and average values of J versus T and the choice of the least squares error minimization method for the MA(3) model of Eq. {3.49}. As expected, the A, P, and W methods show improved performances with increasing T, but even at T=1000, these methods are significantly inferior to the C method.

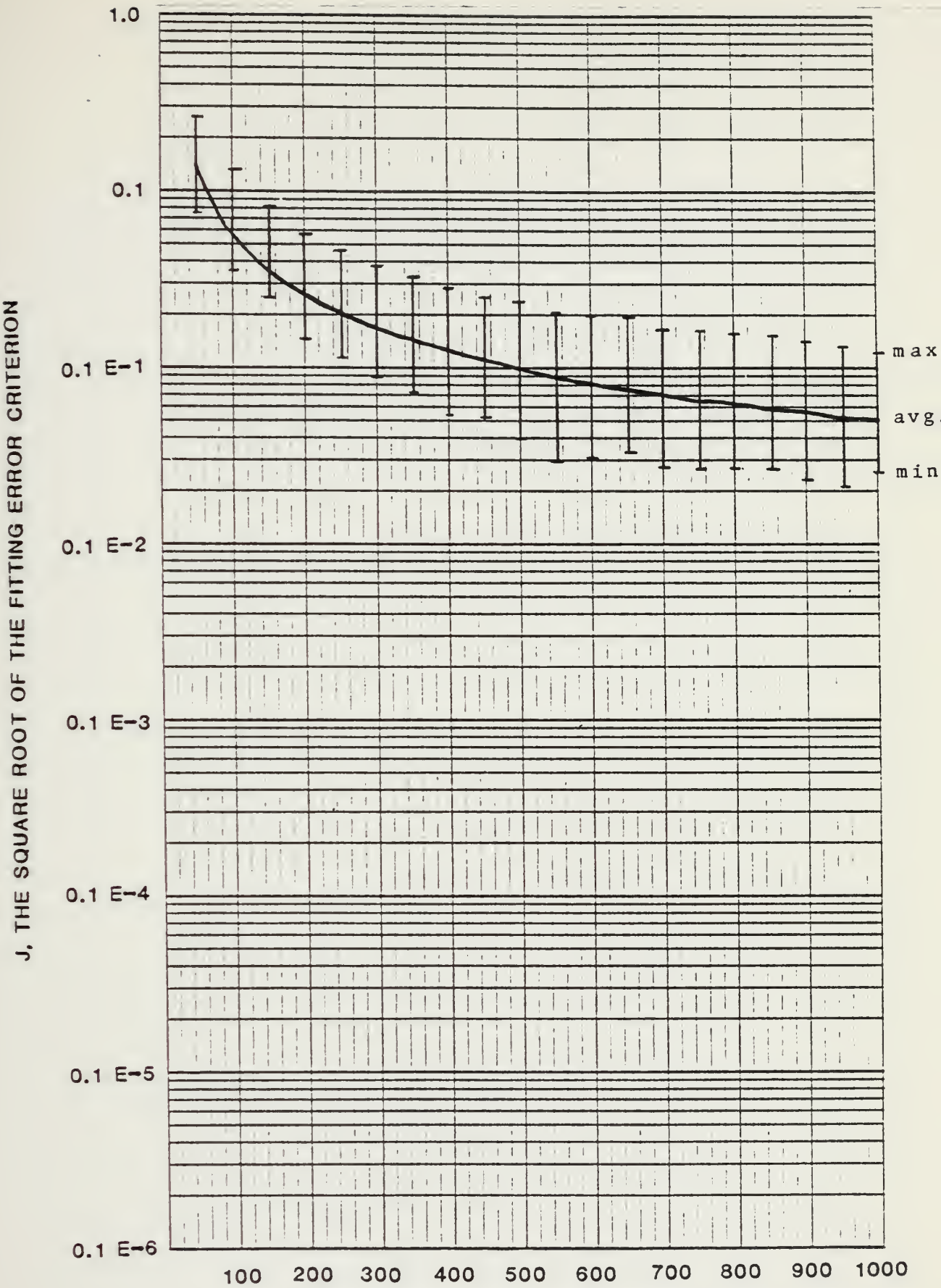


Figure 4: PREWINDOWED ANALYSIS OF A MA(3) SYSTEM

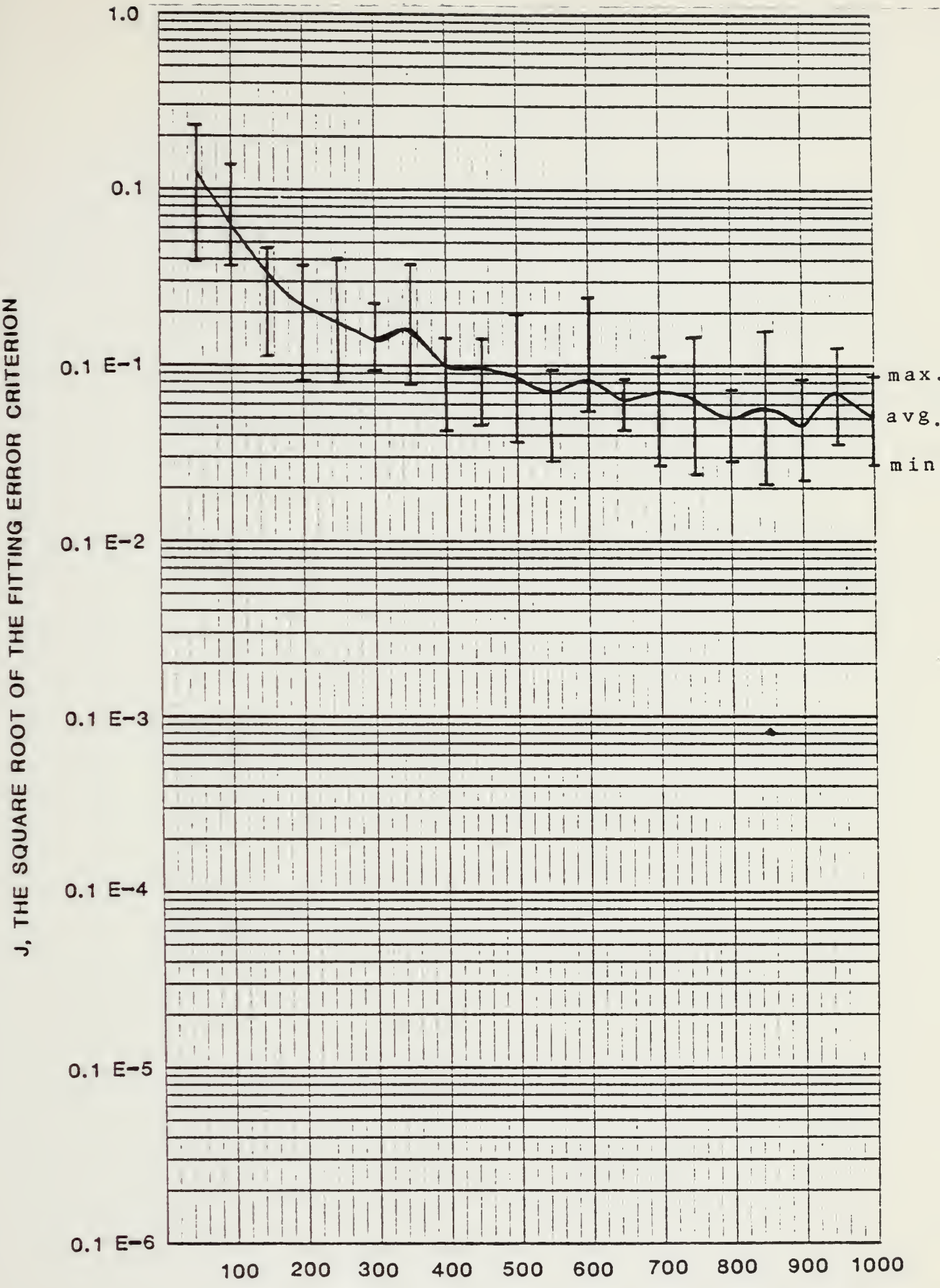


Figure 5: POSTWINDOWED ANALYSIS OF A MA(3) SYSTEM

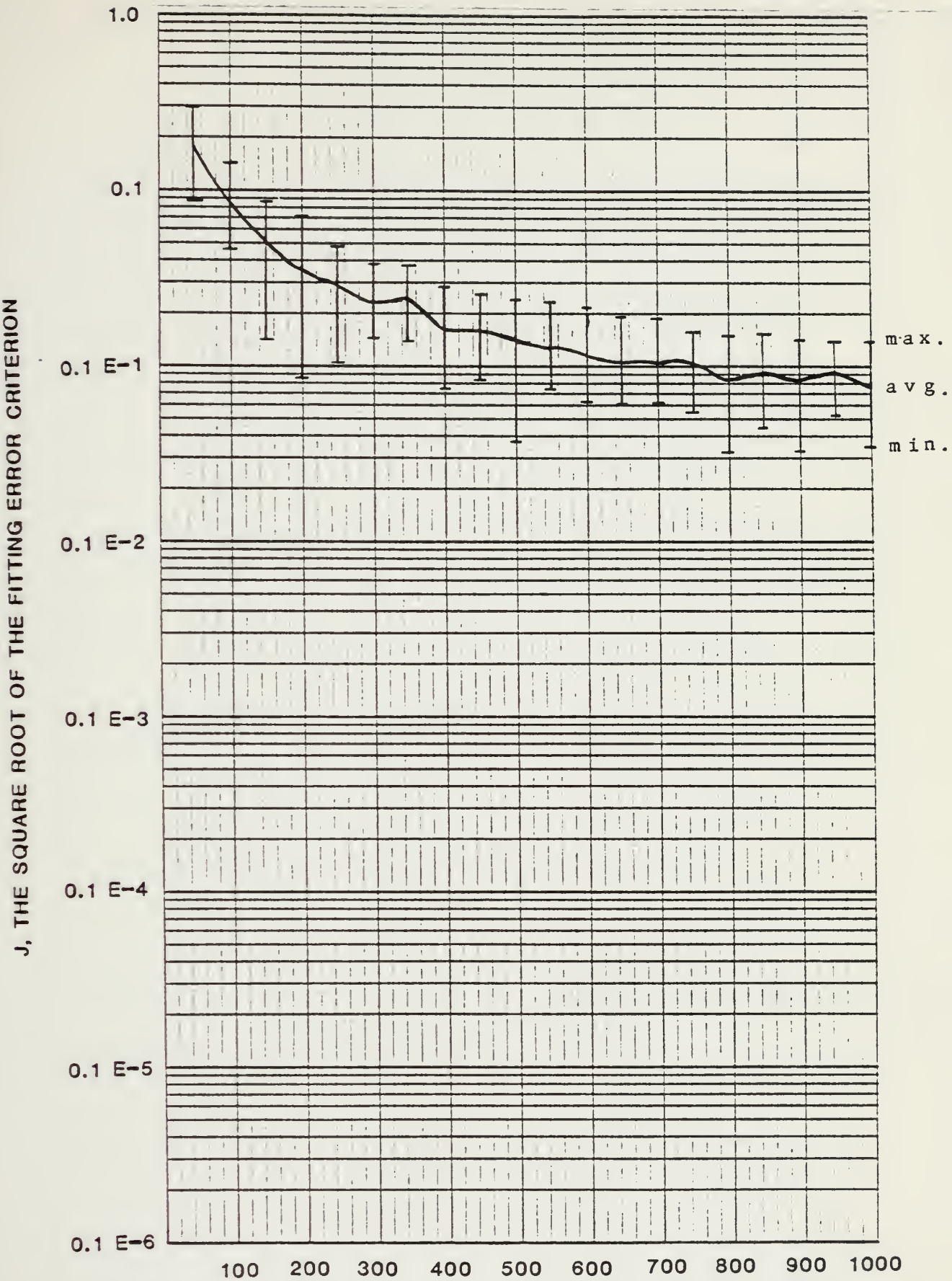


Figure 6: AUTOCORRELATION ANALYSIS OF A MA(3) SYSTEM

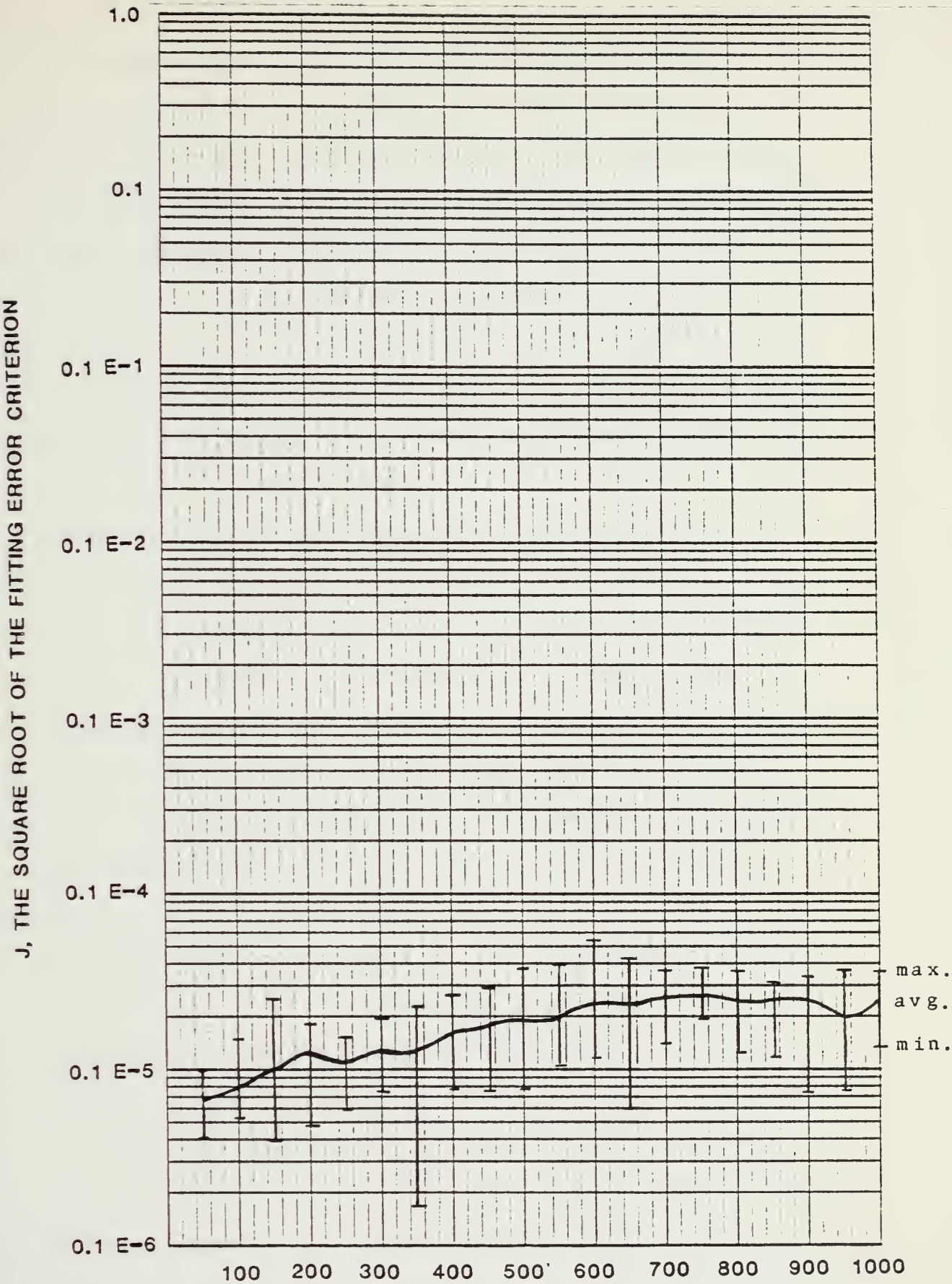


Figure 7: COVARIANCE ANALYSIS OF A MA(3) SYSTEM

It is conjectured that the slight increase in the average value of J as T increases with the Covariance method, is due to the finite precision of the computer used for these experiments. The experiments could be repeated using double precision variables in an attempt to verify this conjecture. We are actually approximating the N equations $X\underline{\theta} = \underline{y}$ for the 4 unknowns $\underline{\theta}$. Since there are many more measurement equations than constraint equations, it is natural that the average error should be slightly higher as T (and therefore N) gets larger.

Figure 8 shows how the choice of the four error minimization methods affect the matrices and vectors involved in the evaluation of the MA(3) model, for different values of T. Note that the R matrix and $\underline{r}$ vector have been normalized by dividing each element by the first-row, first-column entry of R. This does not affect the answer and it provides for an easier comparison of the twelve cases shown.

Since the Covariance method uses only the exact data measurements, we denote as exact the values of R and $\underline{r}$ in the Covariance method of Figure 8. The corresponding matrix and vector in the other three methods can therefore be considered to have errors. The important thing to recognize here is that errors in the third decimal place in the values of the R matrix in these other methods, translate to more significant errors in the inverse of R, and ultimately into substantial errors in the estimates of J and $\underline{\theta}$.

PREWINDOWED METHOD WITH T=50

$$R = \begin{bmatrix} .10000E+1 & .37507E-1 & .29789E-1 & .25930E+0 \\ .37507E-1 & .98776E+0 & .23452E-1 & .26258E-1 \\ .29789E-1 & .23452E-1 & .97185E+0 & .19733E-1 \\ .25930E+0 & .26258E-1 & .19733E-1 & .97092E+0 \end{bmatrix}$$

$$r^T = \begin{bmatrix} .97406E+0 & .83384E+0 & .62672E+0 & .86882E-3 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} .10762E+1 & -.32671E-1 & -.26392E-1 & -.28597E+0 \\ -.32671E-1 & .10158E+1 & -.23140E-1 & -.18277E-1 \\ -.26392E-1 & -.23140E-1 & .10306E+1 & -.13272E-1 \\ -.28599E+0 & -.18277E-1 & -.13272E-1 & .11071E+1 \end{bmatrix}$$

J = .110651E-1 Condition Number = .171425E+1

$$a^T = \begin{bmatrix} .100422E+1 & .800673E+0 & .600887E+0 & -.301163E+0 \end{bmatrix}$$

POSTWINDOWED METHOD WITH T=50

$$R = \begin{bmatrix} .10000E+1 & .12108E-1 & .97792E-2 & .27275E+0 \\ .12108E-1 & .10200E+1 & .25271E-1 & .31335E-1 \\ .97792E-2 & .25271E-1 & .10287E+1 & .39453E-1 \\ .27275E+0 & .31335E-1 & .39453E-1 & .10519E+1 \end{bmatrix}$$

$$r^T = \begin{bmatrix} .93373E+0 & .81489E+0 & .61138E+0 & .91391E-3 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} .10761E+1 & -.42194E-2 & .57127E-1 & -.27893E+0 \\ -.42194E-2 & .98185E+0 & -.23023E-1 & -.27291E-1 \\ .57127E-1 & -.23034E-1 & .97407E+0 & -.35996E-1 \\ -.27893E+0 & -.27291E-1 & -.35996E-1 & .10252E+1 \end{bmatrix}$$

J = .765225E-1 Condition Number = .174448E+1

$$a^T = \begin{bmatrix} .100146E+1 & .782047E+0 & .577257E+0 & -.303756E+0 \end{bmatrix}$$

PREWINDOWED METHOD WITH T=250

$$R = \begin{bmatrix} .10000E+1 & .53729E-2 & .96696E-1 & .98361E-2 \\ .53729E-2 & .98958E+0 & -.23251E-2 & .90248E-1 \\ .96696E-1 & -.23251E-1 & .98389E+0 & -.70876E-2 \\ .98361E-2 & .90248E-1 & -.70876E-2 & .97990E+0 \end{bmatrix}$$

$$r^T = \begin{bmatrix} .10599E+1 & .76868E+0 & .68743E+0 & -.21619E+0 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} .10097E+1 & -.47658E-2 & -.99322E-1 & -.10415E-1 \\ -.47658E-2 & .10191E+1 & .22011E-2 & -.93796E-1 \\ -.99322E-1 & .22011E-1 & .10262E+1 & .82167E-1 \\ -.10415E-1 & -.93796E-1 & .82167E-2 & .10293E+1 \end{bmatrix}$$

J = .157909E-2 Condition Number = .122546E+1

$$a^T = \begin{bmatrix} .100053E+1 & .800110E+0 & .600083E+0 & -.300015E+0 \end{bmatrix}$$

POSTWINDOWED METHOD WITH T=250

$$R = \begin{bmatrix} .10000E+1 & .18795E-2 & .94562E-1 & .99020E-2 \\ .18795E-2 & .10026E+1 & .35784E-2 & .97344E-1 \\ .94562E-1 & .35784E-2 & .10037E+1 & .54089E-2 \\ .99020E-2 & .97344E-1 & .54089E-2 & .10067E+1 \end{bmatrix}$$

$$r^T = \begin{bmatrix} .10553E+1 & .76579E+0 & .68586E+0 & -.21764E+0 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} .10091E+1 & -.64452E-3 & -.95016E-1 & -.93526E-2 \\ -.64452E-3 & .10069E+1 & -.30045E-2 & -.97340E-1 \\ -.95016E-1 & -.30045E-2 & .10053E+1 & -.41762E-2 \\ -.93526E-2 & -.97340E-1 & -.41762E-2 & .10029E+1 \end{bmatrix}$$

J = .494679E-1 Condition Number = .122783E+1

$$a^T = \begin{bmatrix} .100123E+1 & .789512E+0 & .587831E+0 & -.305518E+0 \end{bmatrix}$$

PREWINDOWED METHOD WITH T=1000

$$R = \begin{bmatrix} .10000E+1 & -.21069E-1 & .11255E-1 & -.63557E-1 \\ -.21069E-1 & .99994E+0 & -.21154E-1 & .11457E-1 \\ .11255E-1 & -.21154E-1 & .99981E+0 & -.20846E-1 \\ -.63557E-1 & .11456E-1 & -.20846E-1 & .99908E+0 \end{bmatrix}$$

$$r^T = \begin{bmatrix} .10091E+1 & .76178E+0 & .60051E+0 & -.36662E+0 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} .10046E+1 & .20237E-1 & -.95566E-2 & .63474E-1 \\ .20237E-1 & .10010E+1 & .20748E-1 & -.97592E-2 \\ -.95566E-2 & .20748E-1 & .10012E+1 & .20044E-1 \\ .63474E-1 & -.97592E-2 & .20044E-1 & .10055E+1 \end{bmatrix}$$

J = .187530E-3 Condition Number = .115708E+1

$$a^T = \begin{bmatrix} .100013E+1 & .800029E+0 & .600030E+0 & -.299991E+0 \end{bmatrix}$$

POSTWINDOWED METHOD WITH T=1000

$$R = \begin{bmatrix} .10000E+1 & -.21945E-1 & .10608E-1 & -.63659E-1 \\ -.21945E-1 & .10006E+1 & -.21540E-1 & .11273E-1 \\ .10608E-1 & -.21540E-1 & .10009E+1 & -.21102E-1 \\ -.63659E-1 & .11273E-1 & -.21102E-1 & .10016E+1 \end{bmatrix}$$

$$r^T = \begin{bmatrix} .10079E+1 & .76209E+0 & .59999E+0 & -.36721E+0 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} .10046E+1 & .21127E-1 & -.88558E-2 & .63425E-1 \\ .21127E-1 & .10004E+1 & .21106E-1 & -.94719E-2 \\ -.88558E-2 & .21106E-1 & .10001E+1 & .20270E-1 \\ .63425E-1 & -.94719E-2 & .20270E-1 & .10030E+1 \end{bmatrix}$$

J = .183869E-2 Condition Number = .115779E+1

$$a^T = \begin{bmatrix} .100003E+1 & .799832E+0 & .599761E+0 & -.299431E+0 \end{bmatrix}$$

Figure 8: CONTENTS OF MATRICES AND VECTORS UNDER VARYING CONDITIONS, FOR A MA(3) MODEL AND SYSTEM (Note: Matrix R and vector r have been normalized by the first row, first column entry of R.)

AUTOCORRELATION METHOD WITH T=50

$$R = \begin{bmatrix} .10000E+1 & .37507E-1 & .29789E-1 & .25930E+0 \\ .37507E-1 & .10000E+1 & .37507E-1 & .29789E-1 \\ .29789E-1 & .37507E-1 & .10000E+1 & .37507E-1 \\ .25930E+0 & .29789E-1 & .37507E-1 & .10000E+1 \end{bmatrix}$$

$$\underline{r}^T = \begin{bmatrix} .97406E+0 & .83384E+0 & .62672E+0 & .46882E-1 \end{bmatrix}$$

$$\bar{R}^{-1} = \begin{bmatrix} .10735E+1 & -.31256E-1 & -.20431E-1 & -.27666E+0 \\ -.31256E-1 & .10031E+1 & -.35927E-1 & -.20431E-1 \\ -.20431E-1 & -.35927E-1 & .10031E+1 & -.31256E-1 \\ -.27666E+0 & -.20431E-1 & -.31256E-1 & .10735E-1 \end{bmatrix}$$

J = .718937E-1 Condition Number = .172618E+1

$$\underline{a}^T = \begin{bmatrix} .100656E+1 & .783466E+0 & .578799E+0 & -.305178E+0 \end{bmatrix}$$

COVARIANCE METHOD WITH T=50

$$R = \begin{bmatrix} .10000E+1 & .12108E-1 & .97792E-2 & .27275E+0 \\ .12108E-1 & .10060E+1 & .97792E-2 & .27275E+0 \\ .97792E-2 & .10487E-1 & .99906E+0 & .20757E-1 \\ .27275E+0 & .27621E-1 & .20757E-1 & .10214E+1 \end{bmatrix}$$

$$\underline{r}^T = \begin{bmatrix} .93373E+0 & .81449E+0 & .61138E+0 & .91391E-1 \end{bmatrix}$$

$$\bar{R}^{-1} = \begin{bmatrix} .10786E+1 & -.50323E-2 & -.45249E-2 & -.28783E+0 \\ -.50434E-2 & .99493E+0 & -.98674E-2 & -.25363E-1 \\ -.45249E-2 & -.98674E-2 & .10015E+1 & -.18879E-1 \\ -.28783E+0 & -.25363E-1 & -.18879E-1 & .10571E+1 \end{bmatrix}$$

J = .111785E-5 Condition Number = .174774E+1

$$\underline{a}^T = \begin{bmatrix} .100000E+1 & .800000E+0 & .600000E+0 & -.300000E+0 \end{bmatrix}$$

AUTOCORRELATION METHOD WITH T=250

$$R = \begin{bmatrix} .10000E+1 & .53729E-2 & .96696E-1 & .98361E-1 \\ .53729E-2 & .10000E+1 & .53729E-2 & .96696E-1 \\ .96696E-1 & .53729E-2 & .10000E+1 & .53729E-2 \\ .98361E-1 & .96696E-1 & .53729E-2 & .10000E+1 \end{bmatrix}$$

$$\underline{r}^T = \begin{bmatrix} .10599E+1 & .76868E+0 & .68743E+0 & -.21613E+0 \end{bmatrix}$$

$$\bar{R}^{-1} = \begin{bmatrix} .10095E+1 & -.40282E-2 & -.97549E-1 & -.90164E-2 \\ -.40282E-2 & .10095E+1 & -.45102E-2 & -.97549E-1 \\ -.97549E-1 & -.45102E-2 & .10000E+1 & -.40282E-2 \\ -.90164E-2 & -.97549E-1 & -.40282E-2 & .10095E+1 \end{bmatrix}$$

J = .487747E-1 Condition Number = .123157E+1

$$\underline{a}^T = \begin{bmatrix} .100181E+1 & .789680E+0 & .587958E+0 & -.305561E+0 \end{bmatrix}$$

COVARIANCE METHOD WITH T=250

$$R = \begin{bmatrix} .10000E+1 & .18795E-2 & .94562E-1 & .99020E-2 \\ .18795E-2 & .99209E+0 & -.41712E-2 & .90853E-1 \\ .94562E-1 & -.41712E-2 & .98748E+0 & -.71351E-1 \\ .99020E-2 & .90853E-1 & -.71351E-2 & .98647E+0 \end{bmatrix}$$

$$\underline{r}^T = \begin{bmatrix} .10553E+1 & .76579E+0 & .68586E+0 & -.21764E+0 \end{bmatrix}$$

$$\bar{R}^{-1} = \begin{bmatrix} .10093E+1 & -.13382E-2 & -.96729E-1 & -.10707E-1 \\ -.13382E-2 & .10166E+1 & .37459E-2 & -.93584E-1 \\ -.96729E-1 & .37459E-2 & .10220E+1 & .80181E-2 \\ .10707E-1 & -.93584E-1 & .80181E-2 & .10225E+1 \end{bmatrix}$$

J = .141219E-5 Condition Number = .121941E+1

$$\underline{a}^T = \begin{bmatrix} .100000E+1 & .800000E+0 & .600000E+0 & -.300000E+0 \end{bmatrix}$$

AUTOCORRELATION METHOD WITH T=1000

$$R = \begin{bmatrix} .10000E+1 & -.21069E-1 & .11255E-1 & -.63557E-1 \\ -.21069E-1 & .10000E+1 & -.21069E-1 & .11255E-1 \\ .11255E-1 & -.21069E-1 & .10000E+1 & -.21069E-1 \\ -.63557E-1 & .11255E-1 & -.21069E-1 & .10000E+1 \end{bmatrix}$$

$$\underline{r}^T = \begin{bmatrix} .10091E+1 & .76278E+0 & .60051E+0 & -.36662E+0 \end{bmatrix}$$

$$\bar{R}^{-1} = \begin{bmatrix} .10046E+1 & .20250E-1 & -.95433E-2 & .63418E-1 \\ .20250E-1 & .10010E+1 & .20660E-1 & -.95433E-2 \\ -.95433E-2 & .20660E-1 & .10010E+1 & .20250E-1 \\ .63418E-1 & -.95433E-2 & .20250E-1 & .10046E+1 \end{bmatrix}$$

J = .184394E-2 Condition Number = .115703E+1

$$\underline{a}^T = \begin{bmatrix} .100016E+1 & .799862E+0 & .599792E+0 & -.299422E+0 \end{bmatrix}$$

COVARIANCE METHOD WITH T=1000

$$R = \begin{bmatrix} .10000E+1 & -.21945E-1 & .10608E-1 & -.63659E-1 \\ -.21945E-1 & .10006E+1 & -.21625E-1 & .11475E-1 \\ .10608E-1 & -.21625E-1 & .10007E+1 & -.20880E-1 \\ -.63659E-1 & .11475E-1 & -.20880E-1 & .10007E+1 \end{bmatrix}$$

$$\underline{r}^T = \begin{bmatrix} .10079E+1 & .76209E+0 & .59999E+0 & -.36721E+0 \end{bmatrix}$$

$$\bar{R}^{-1} = \begin{bmatrix} .10046E+1 & .21114E-1 & -.88689E-2 & .63481E-1 \\ .21114E-1 & .10005E+1 & .21194E-1 & -.96877E-1 \\ -.88689E-2 & .21194E-1 & .10003E+1 & .20064E-1 \\ .63481E-1 & -.96877E-2 & .20064E-1 & .10019E+1 \end{bmatrix}$$

J = .223429E-5 Condition Number = .115782E+1

$$\underline{a}^T = \begin{bmatrix} .100000E+1 & .800000E+0 & .600000E+0 & -.300000E+0 \end{bmatrix}$$

Figure 8: (CONTINUED)

While we are interested in determining the least squares method that provides the minimum J, we have also evaluated the typical offset in the coefficient estimates that result from the use of these four methods, and present this information in Figures 9 through 12. The erratic behavior of the A method in estimating the coefficient values appears to be the result of the padded zeros on both ends of the data matrix X. The appearance of similar curves in Figures 9 through 12 for the A and W cases can be explained as follows. Both the Autocorrelation (A) and Postwindowed (W) cases have padded zeros at the bottom end of the matrix X given by Eq. {3.30} and {3.32} respectively. The effect of the padded zeros in the Autocorrelation case X matrix decreases as N gets large, and the A and W cases approach each other in this limit.

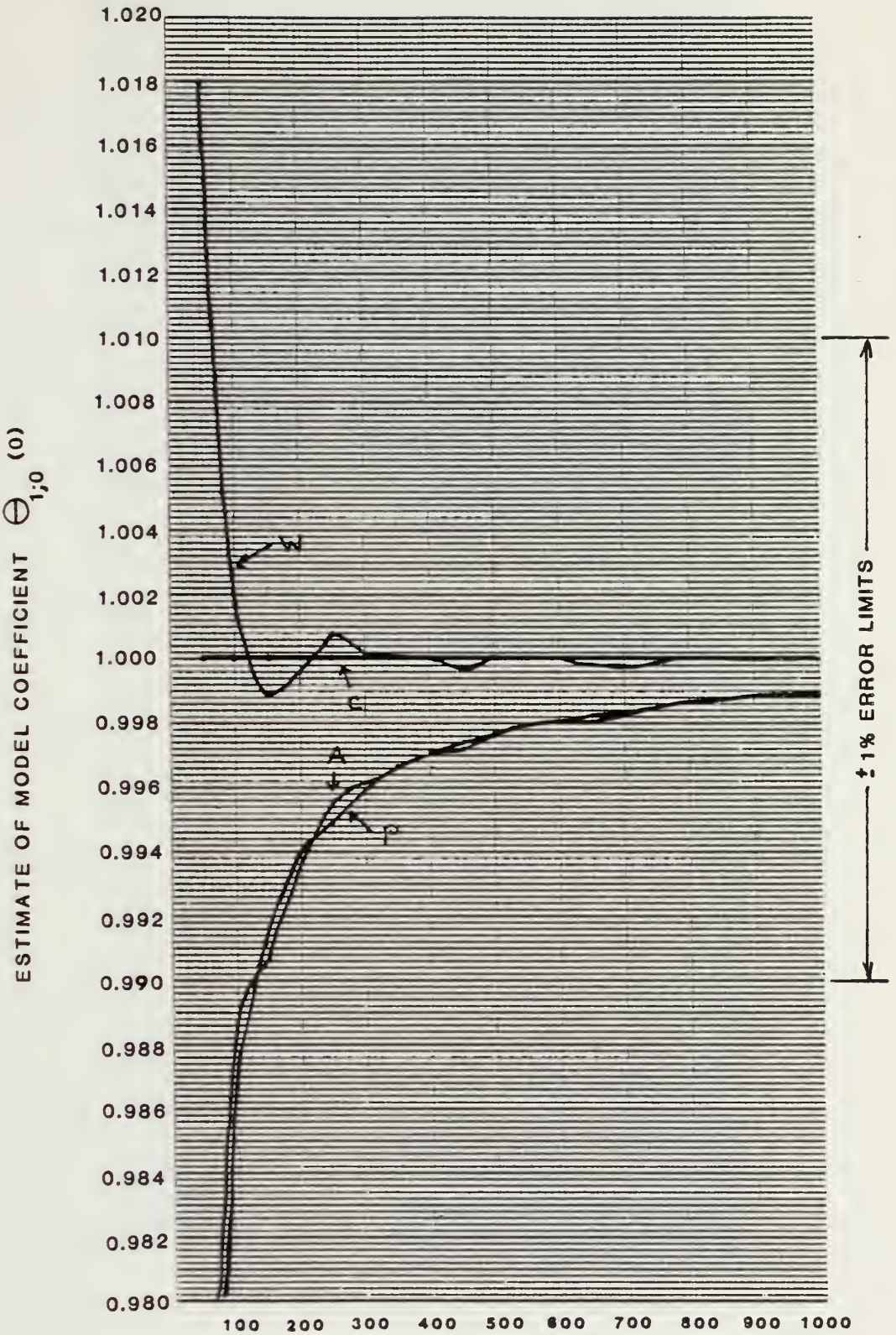


Figure 9: COEFFICIENT ESTIMATION OF A MA(3) SYSTEM

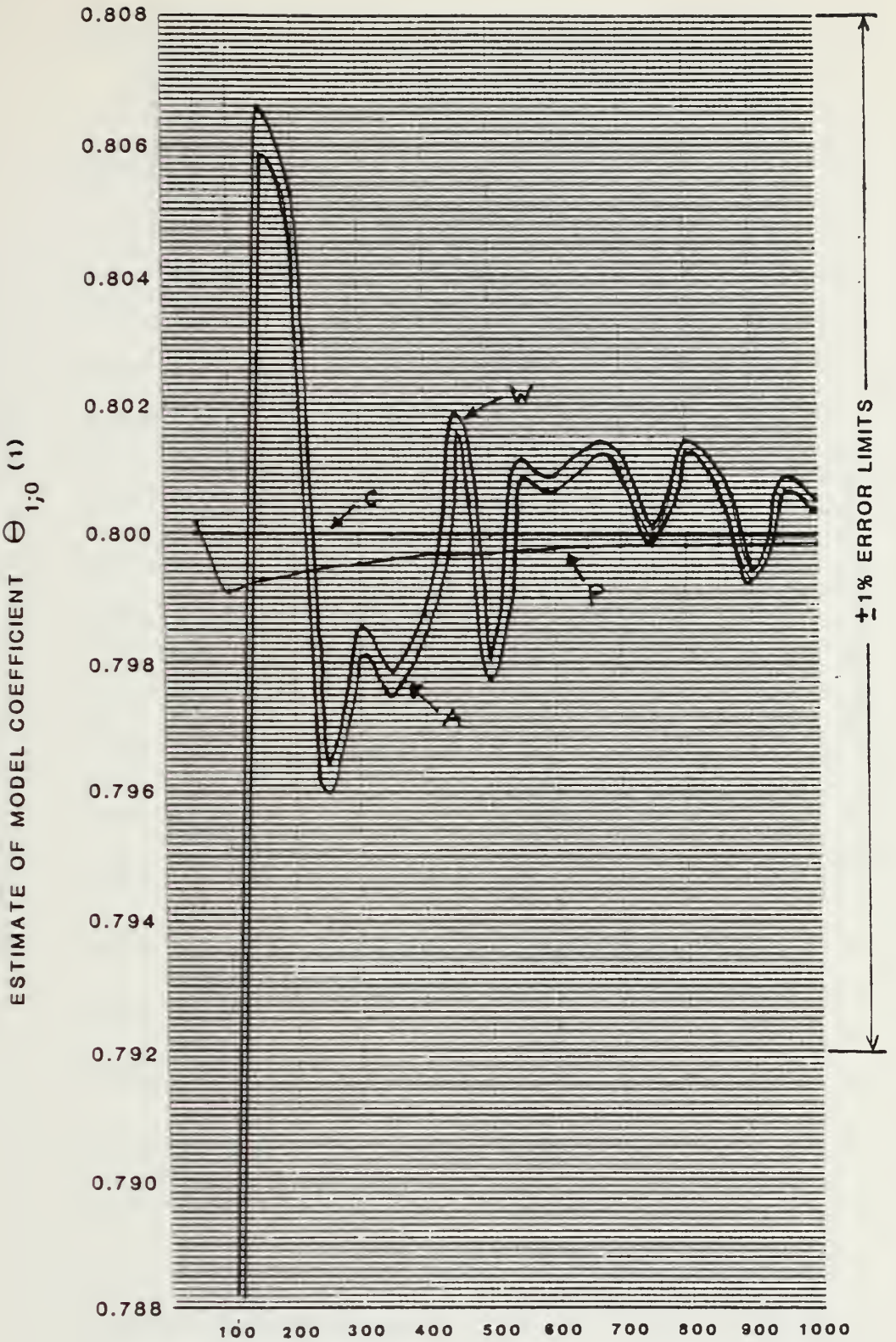


Figure 10: COEFFICIENT ESTIMATION OF A MA(3) SYSTEM

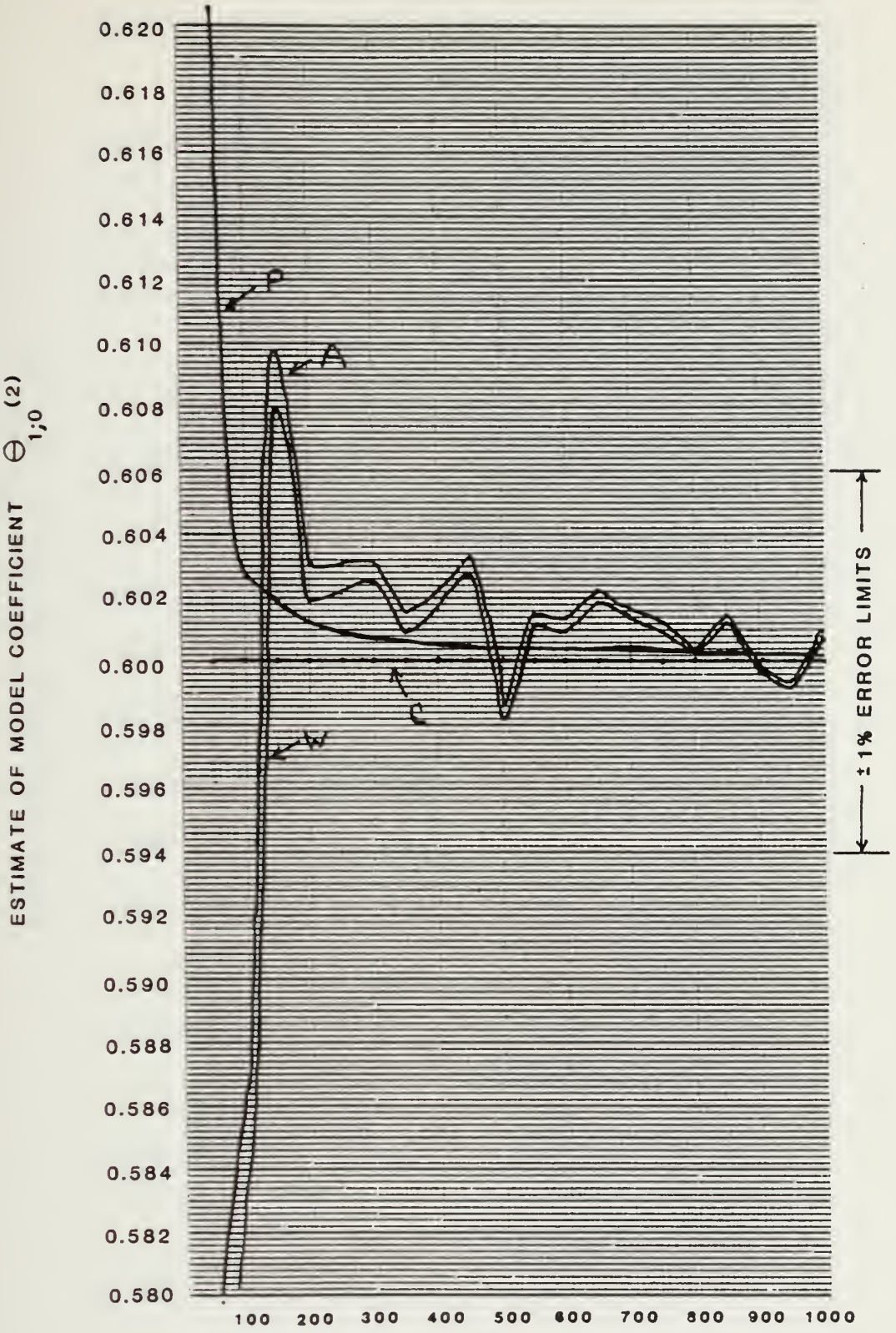


Figure 11: COEFFICIENT ESTIMATION OF A MA(3) SYSTEM

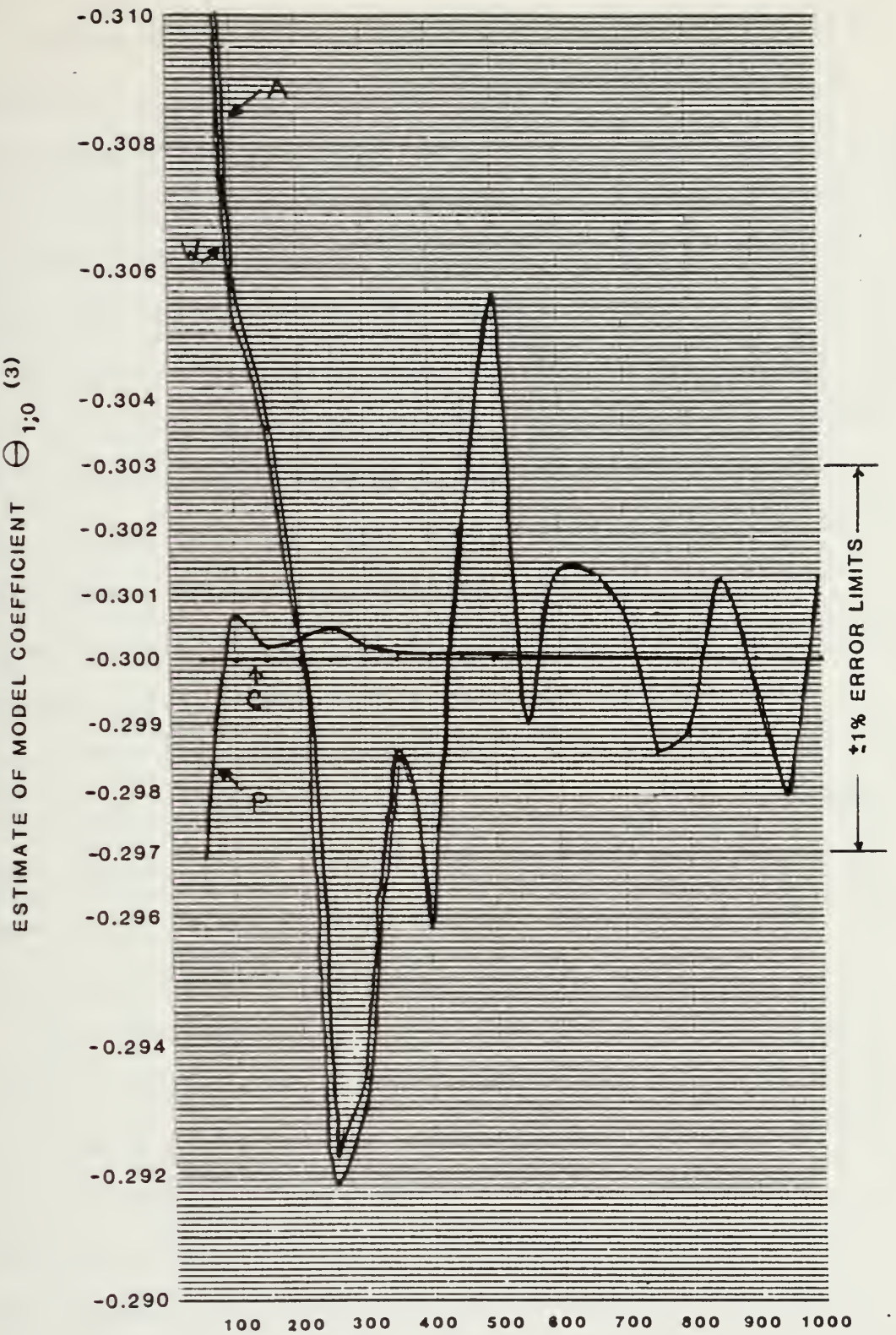


Figure 12: COEFFICIENT ESTIMATION OF A MA(3) SYSTEM

Figures 4 through 12 show the significant differences resulting from the choice of error minimization method. This choice directly affects the contents of R and $\underline{r}$, which in turn affects $\underline{\theta}$ and therefore J. A logical conjecture is that the condition number (ratio of largest to smallest eigenvalue) of the matrix R, could be a good indicator of the quality of the least squares fit. In other words, the more well conditioned the matrix (lower condition number), the lower the corresponding fitting error J. While esthetically pleasing, this conjecture is not born out by experience. In over 30 cases of linear and nonlinear systems modeled using each of these four error minimization methods, the corresponding R matrices were all well conditioned (low condition number), there was no significant difference in condition number between the four methods, and there was no direct correlation between lowest condition numbers and lowest fitting error J. Condition number data is included in the typical results of Figure 8.

This is explained by the following. The fit J is a function of the entire coefficient vector $\underline{\theta}$ and the vector $\underline{r}$. Since the coefficient vector $\underline{\theta}$ is a function of both the matrix R (actually the inverse of R) and the vector $\underline{r}$, the condition number of R is an insufficient measure of the accuracy of $\underline{\theta}$, and therefore is an insufficient measure of the quality of the fit J.

Figures 13 through 16 present the results of the experiment for the ARMA(2,2) model of Eq. {3.50}. Figures 17 through 20 present the results of the experiment for the BVM(2,4) model of Eq. {3.51}. Both of these sets of figures indicate the superior performance of the Covariance (C) method in minimizing the fitting error criterion J.

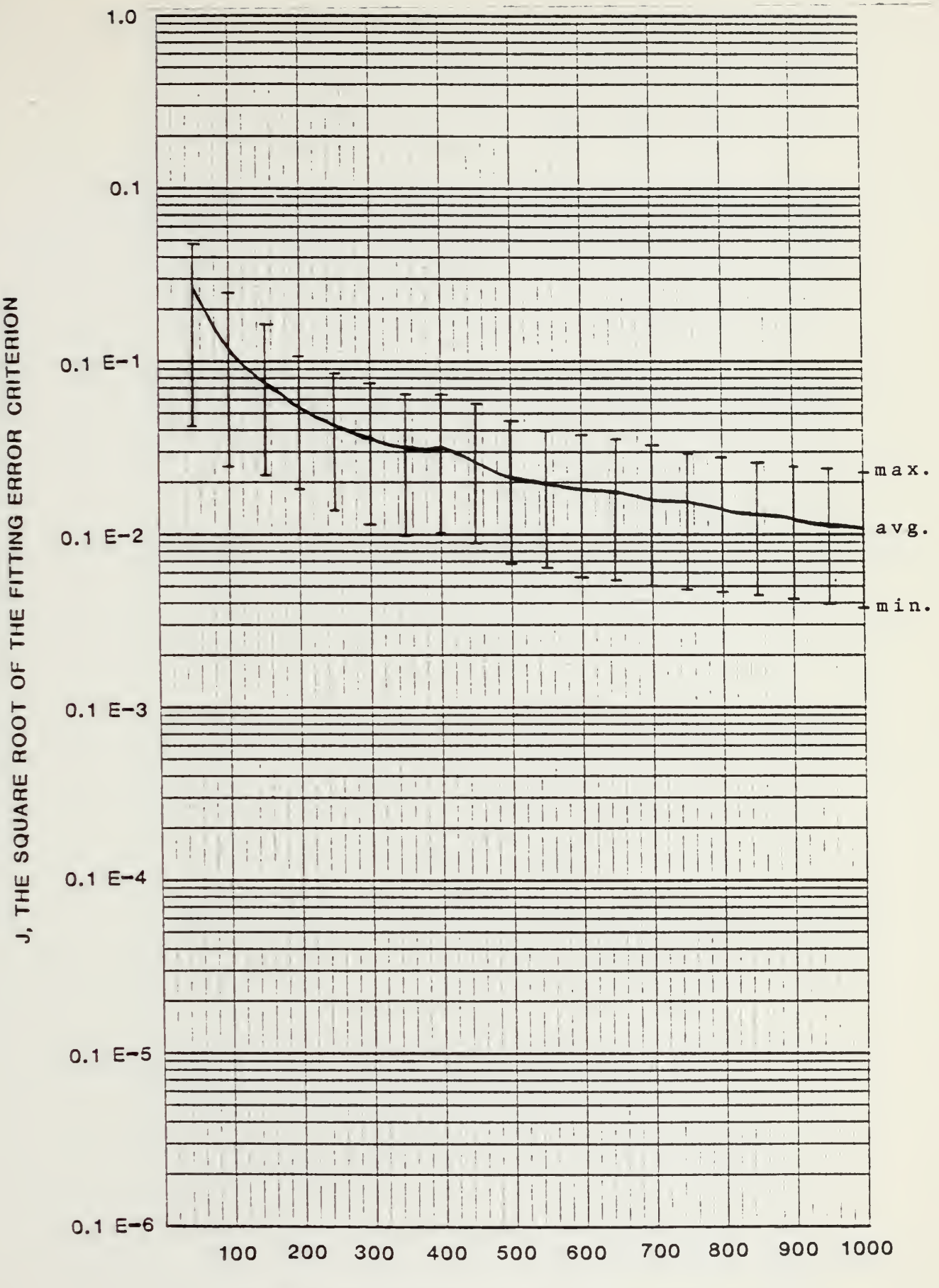


Figure 13: PREWINDOWED ANALYSIS OF AN ARMA(2,2) SYSTEM

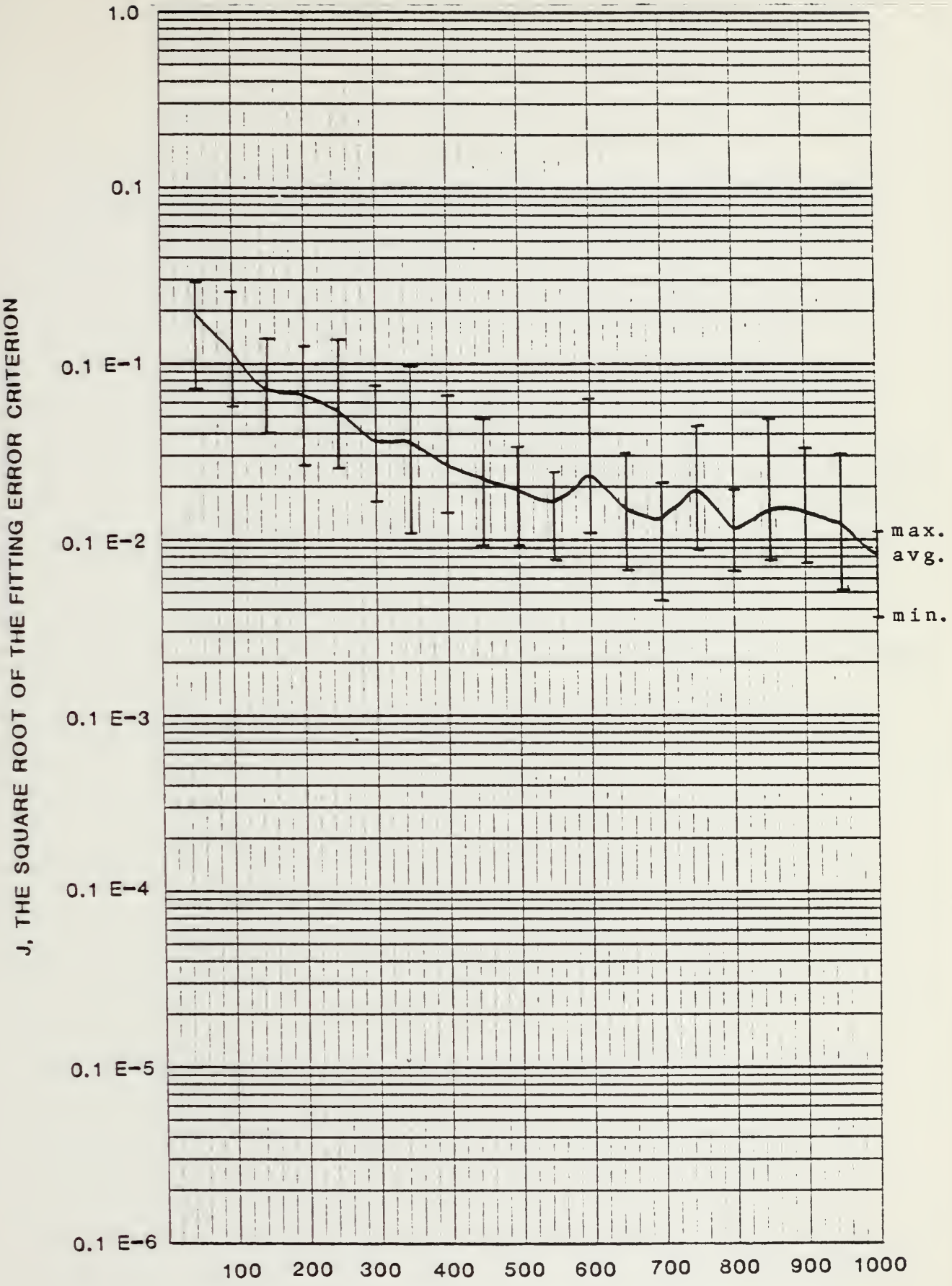


Figure 14: POSTWINDOWED ANALYSIS OF AN ARMA(2,2) SYSTEM

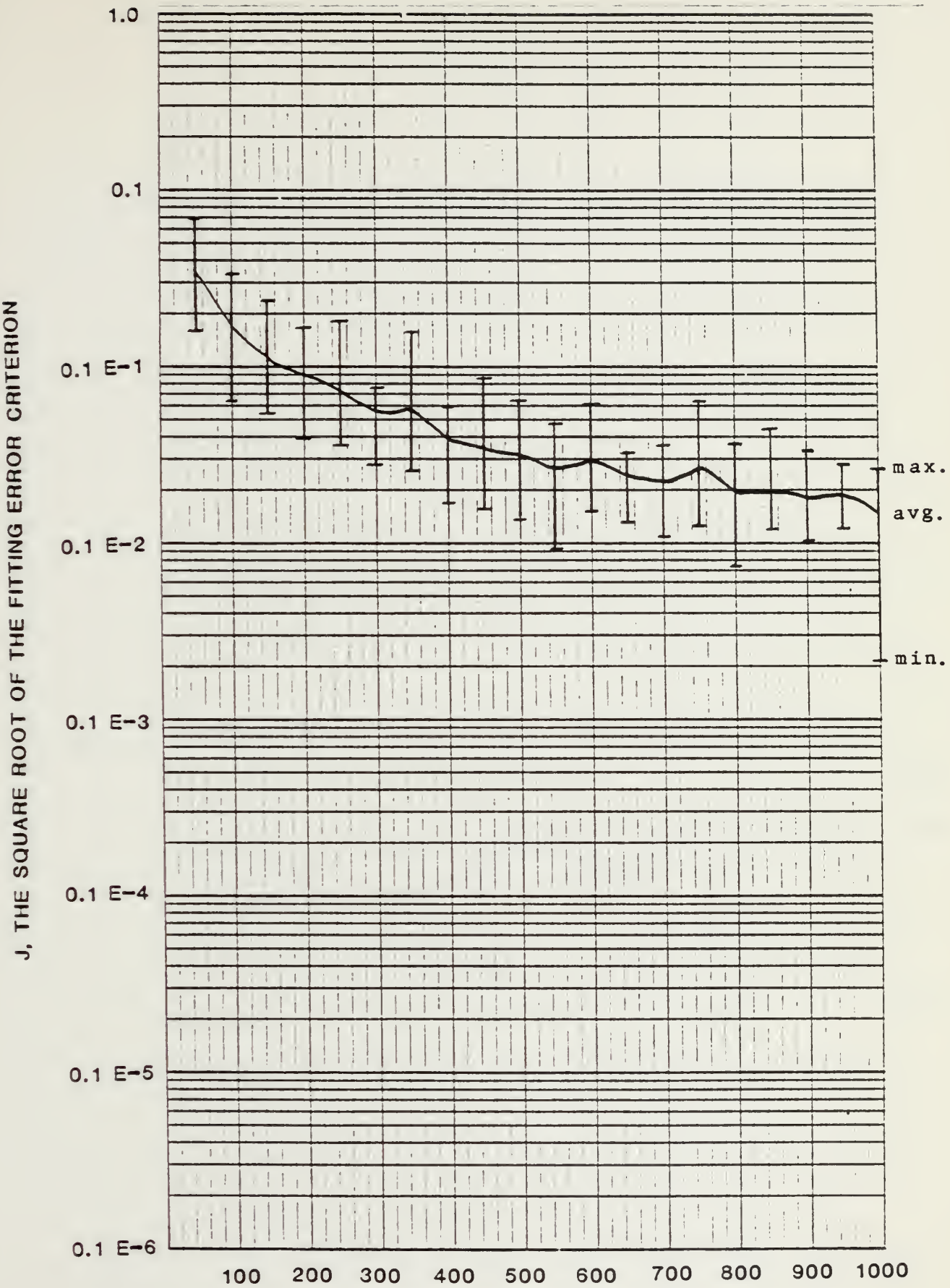


Figure 15: AUTOCORRELATION ANALYSIS OF AN ARMA(2,2) SYSTEM

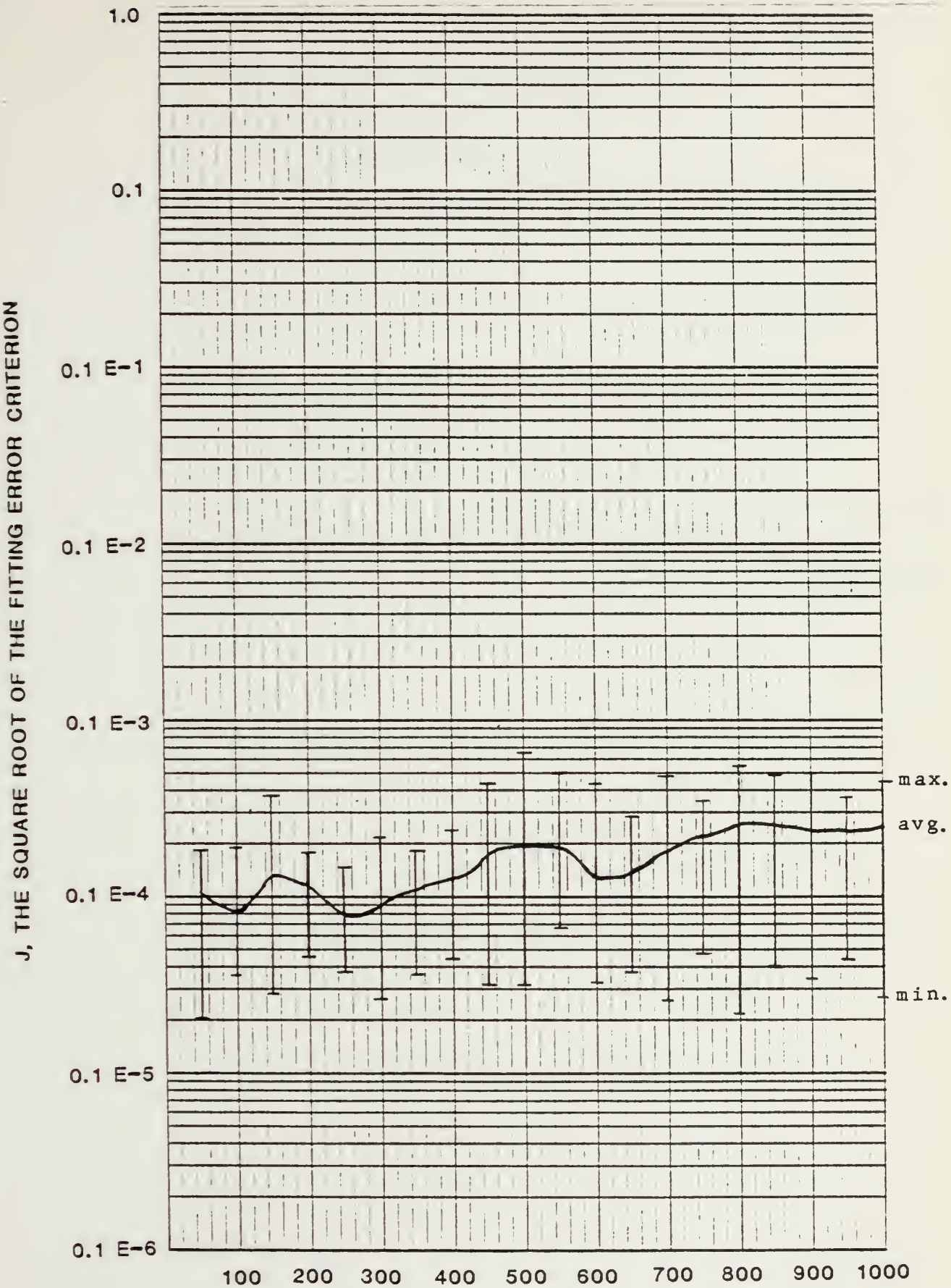


Figure 16: COVARIANCE ANALYSIS OF AN ARMA(2,2) SYSTEM

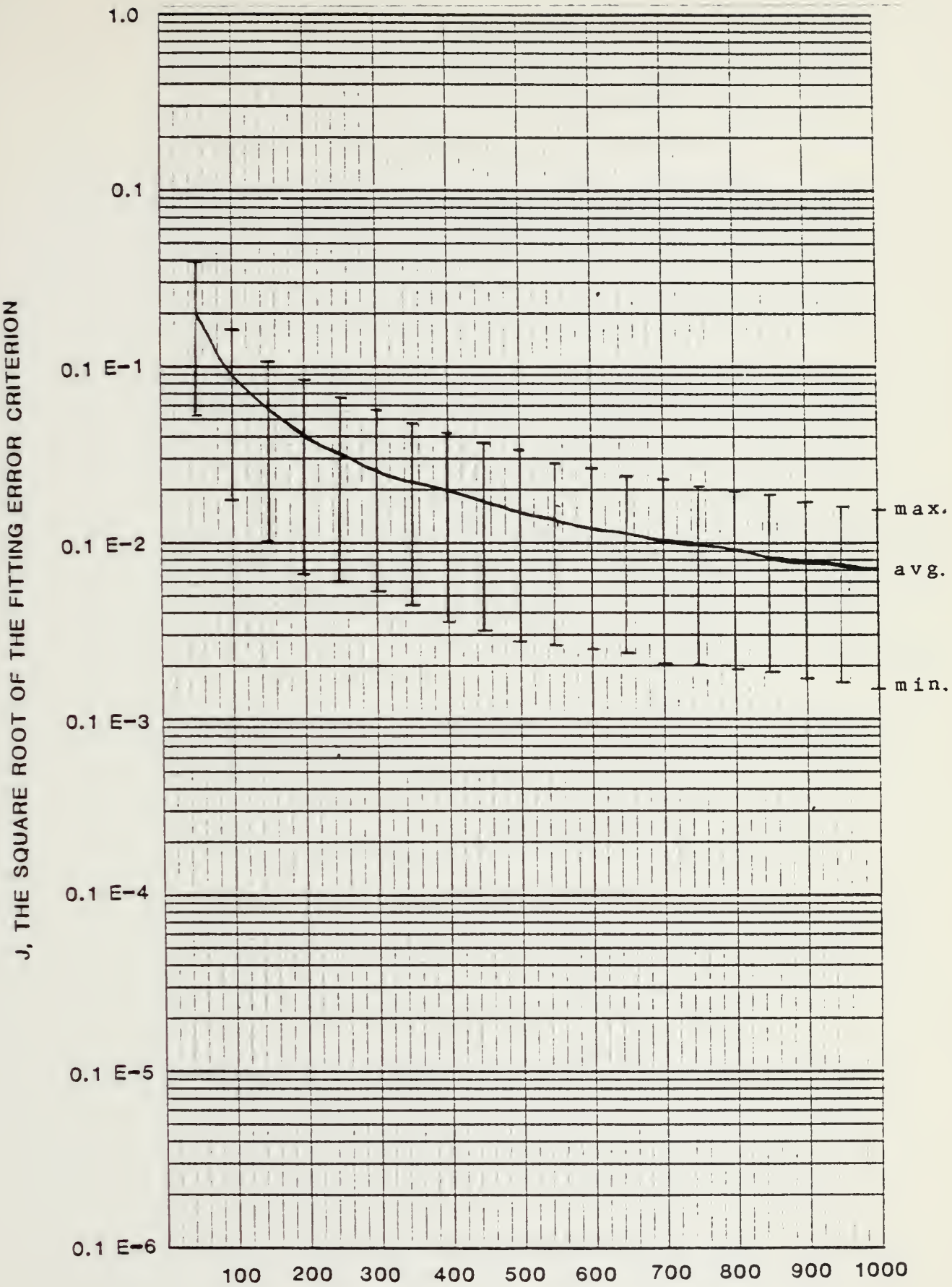


Figure 17: PREWINDOWED ANALYSIS OF A BVM(2,4) SYSTEM

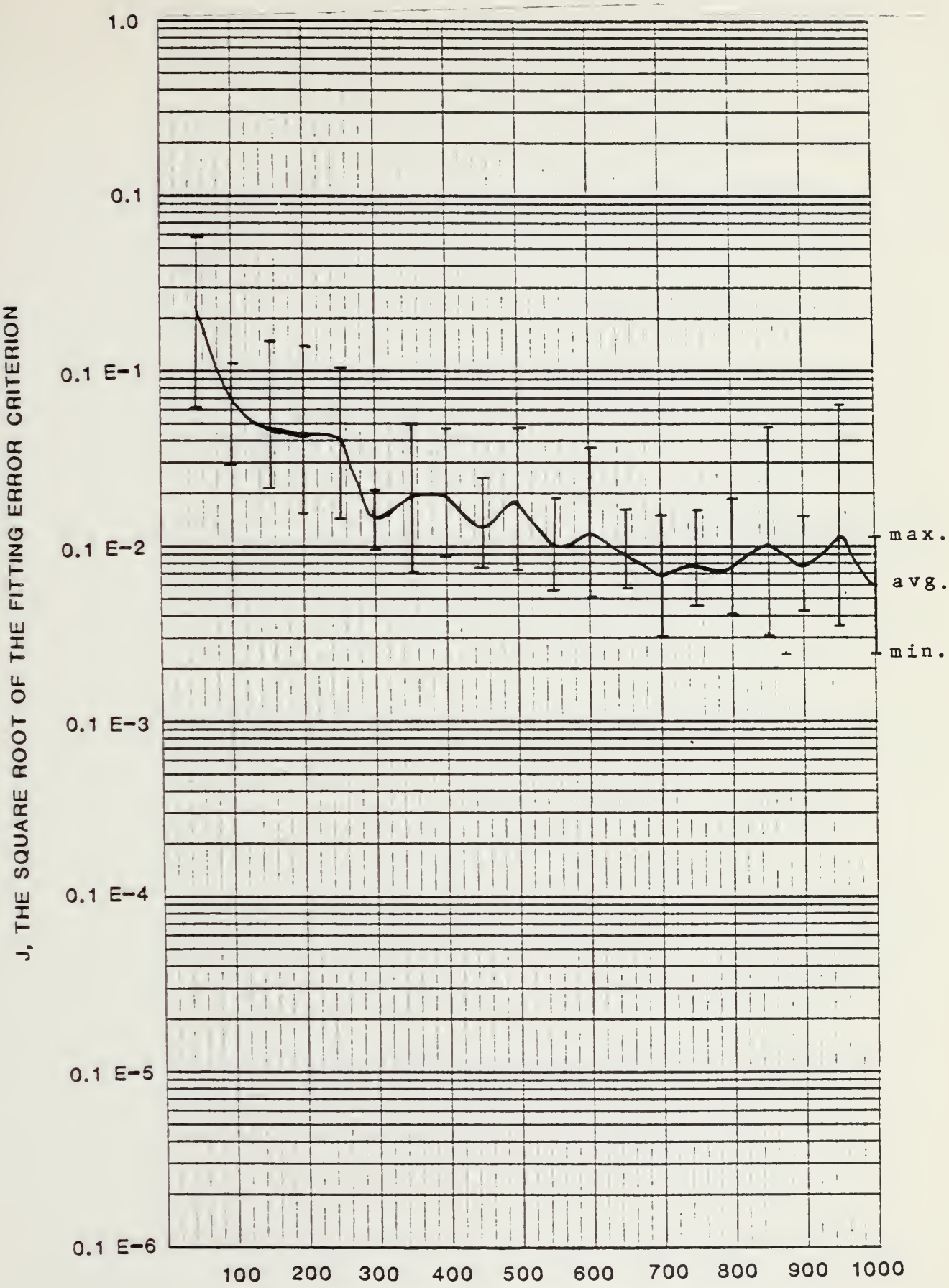


Figure 18: POSTWINDOWED ANALYSIS OF A BVM(2,4) SYSTEM

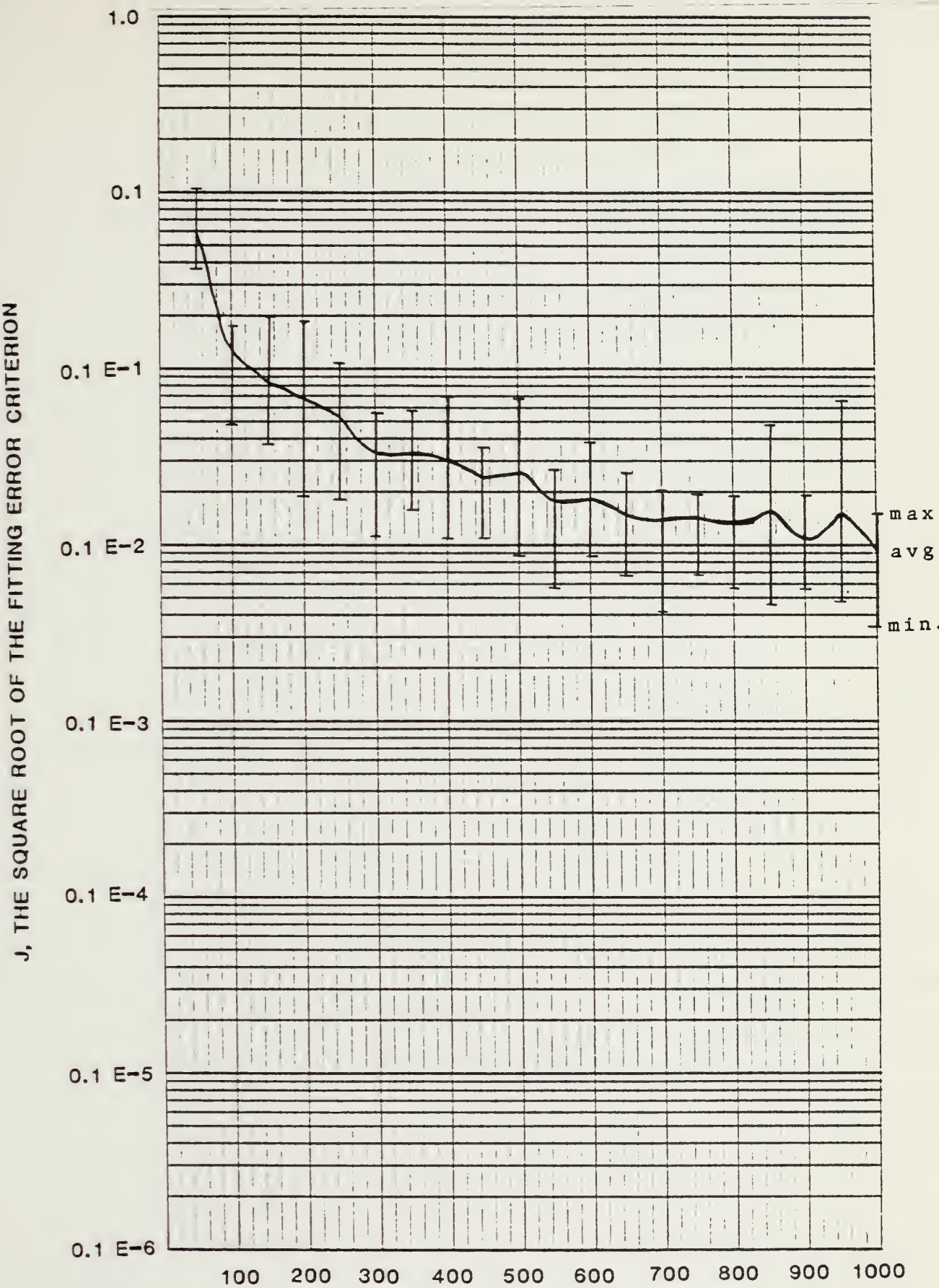


Figure 19: AUTOCORRELATION ANALYSIS OF A BVM(2,4) SYSTEM

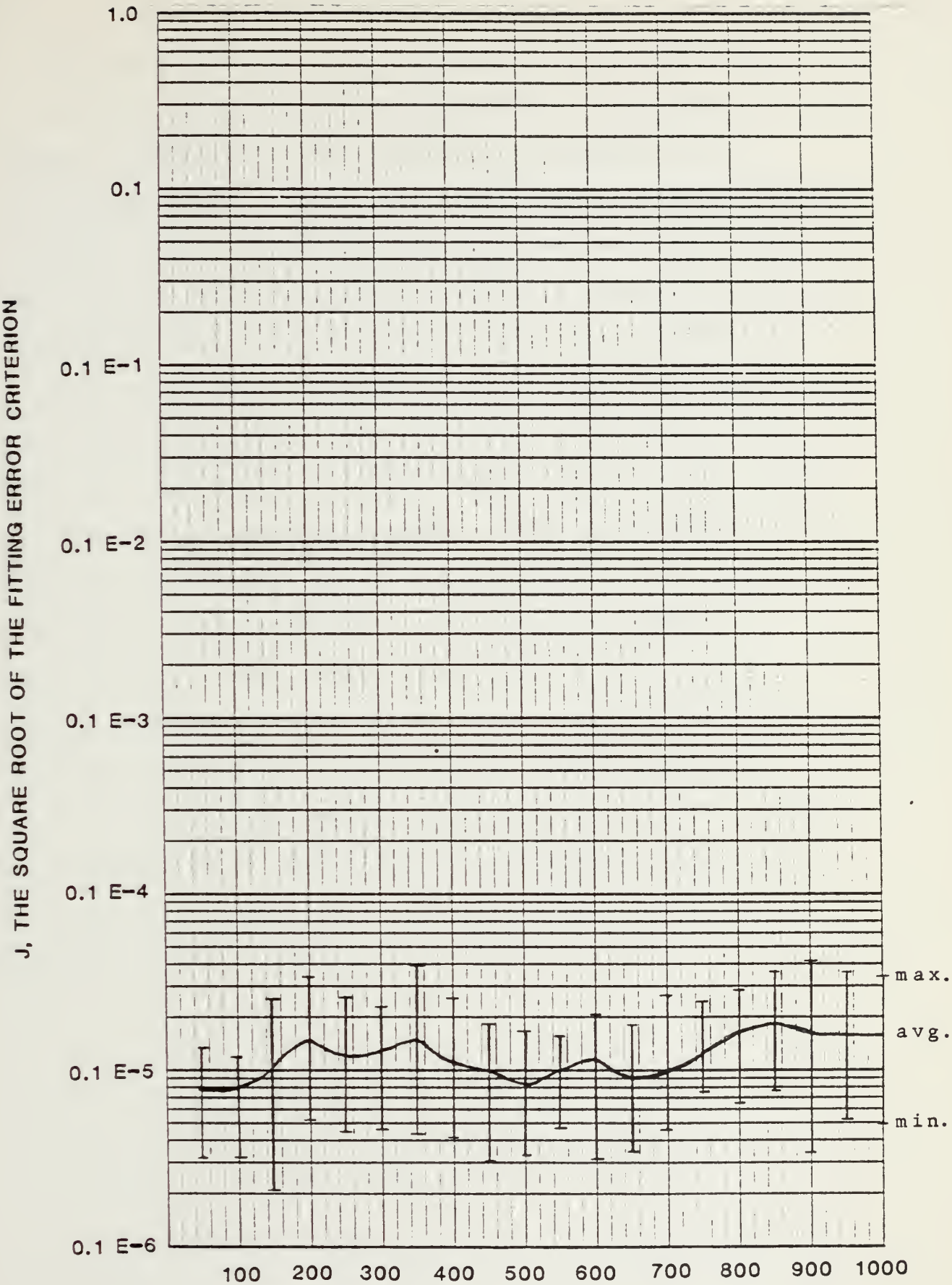


Figure 20: COVARIANCE ANALYSIS OF A BVM(2,4) SYSTEM

This experiment has lead to a greater understanding of the accuracy of the different minimization techniques with respect to the size of the observation sequences for a nonrecursive model, a recursive model, and a nonlinear model of the BVM form. For the models examined, the Covariance least squares error minimization method is superior to the Prewindowed, Postwindowed, and Autocorrelation methods.

These results are representative of those obtained with other system equations. The conclusion to be drawn from simulation Experiment 1 is that the Covariance method is the most accurate of the four methods. Since our primary problem is accurately characterizing systems whose exact mathematical form is unknown, the Covariance method is adopted for the rest of the work in this thesis. This avoids introducing offset errors in J and $\underline{\theta}$ that might give misleading results later in our model growth techniques.

The next section examines another factor that can affect the estimates of J^2 and $\underline{\theta}$; output measurement noise. This is included for convenience at the present time, and is referred to again later on.

D. EFFECTS OF OUTPUT MEASUREMENT NOISE

If the system output is contaminated with additive noise $\{v(n)\}$, then the evaluation of the model and the estimates of the model coefficients may be affected[4]. If the additive noise is uncorrelated with the system input, and the model is nonrecursive (e.g. no $\theta_{0;q}$ or $\theta_{p;q}$ terms in a BVM which thereby reduces to a MA or VOL model), then the effect of the additive noise on the coefficient estimates will generally be small. This effect approaches zero in the limit as the size of the data segment (T-S+1) gets large. This property of a nonrecursive model is well known in the literature [Ref. 13 and Ref. 14, pp 41 and pp 144].

To better understand the effect of uncorrelated additive output noise in the case of a linear recursive model[5], denote the noisy output sequence as $\{z(n)\}$;

$$z(n) = y(n) + v(n) \quad \text{for all } S<=n<=T \qquad \{3.52\}$$

To utilize the equation error minimization techniques discussed at the beginning of this chapter, we substitute $z(n)$ for $y(n)$ in the evaluation equations, and Eq. {3.5} becomes;

$$z(n) = \underline{\hat{\theta}}^T \underline{\hat{x}}(n) + e(n) \qquad \{3.53\}$$

---

4 Other measurement noises such as additive input noise or multiplicative input and/or output noise are also possible, but are not considered.

5 The following analysis holds for linear recursive or nonrecursive models (e.g. ARMA). There appears no tractable way to extend it to recursive nonlinear models (e.g. BVM).

where under the condition of additive output noise, $\hat{\underline{\theta}}$ is the coefficient vector, and $\hat{\underline{x}}(n)$ is the corresponding term vector based on u(n) and z(n), instead of u(n) and y(n). The "^" is used to denote factors affected by the noise.

$$\hat{\underline{x}}(n) = \underline{x}(n) \Big|_{\{y(n)\} = \{z(n)\}}$$

$$= \underline{x}(n) \Big|_{\{y(n)\} = \{y(n)\}} + \underline{x}(n) \Big|_{\substack{\{u(n)\} = 0 \\ \{y(n)\} = \{v(n)\}}}$$

$$= \underline{x}(n) + \underline{x}_v(n) \qquad \{3.54\}$$

Note that if the model is nonrecursive, $\underline{x}_v(n) = \underline{0}$, and $\hat{\underline{x}}(n) = \underline{x}(n)$. A more interesting example is as follows; If $\underline{x}(n)^T = [\, u(n), u(n-1), y(n-1) \,]$, then $\hat{\underline{x}}(n)^T = [\, u(n), u(n-1), z(n-1)] = [\, u(n), u(n-1), y(n-1)+v(n-1) \,]$ and $\underline{x}_v(n)^T = [\, 0, 0, v(n-1) \,]$

Substituting Eq. {3.53} and Eq. {3.54} into Eq. {3.9}, and minimizing with respect to $\hat{\underline{\theta}}$ by matrix calculus, yields the least squares solution equations;

$$\frac{1}{N} [\, \hat{X}^T \hat{X} \,] \hat{\underline{\theta}} = \frac{1}{N} \hat{X}^T \underline{z} \qquad \{3.55\}$$

where $\underline{z}^T = [z(n_2), z(n_2+1), \ldots, z(n_3)]$ {3.56}

and

$$\hat{X}^T = [\, \hat{\underline{x}}(n_2) \;\; \hat{\underline{x}}(n_2+1) \;\; \ldots \;\; \hat{\underline{x}}(n_3) \,] \qquad \{3.57\}$$

Substituting Eq. {3.54} and {3.57} into {3.55} yields;

$$[\, R + R_v \,] \hat{\underline{\theta}} = \underline{r} + \underline{r}_v \qquad \{3.58\}$$

where the following matrices and vectors are defined;

$$R = \frac{1}{N} X^T X \qquad \{3.59\}$$

$$R_v = \frac{1}{N} X_v^T X_v \qquad \{3.60\}$$

$$X^T = [\, \underline{x}(n_2) \;\; \underline{x}(n_2+1) \;\ldots\; \underline{x}(n_3) \,] \qquad \{3.61\}$$

$$X_v^T = [\, \underline{x}_v(n_2) \;\; \underline{x}_v(n_2+1) \;\ldots\; \underline{x}_v(n_3) \,] \qquad \{3.62\}$$

$$\underline{r} = \frac{1}{N} X^T \underline{y} \qquad \{3.63\}$$

$$\underline{r}_v = \frac{1}{N} X_v^T \underline{v} \qquad \{3.64\}$$

$$\underline{v}^T = [v(n_2), v(n_2+1), \ldots\; v(n_3)] \qquad \{3.65\}$$

Solving {3.58} for the model coefficient vector $\hat{\underline{\theta}}$;

$$\hat{\underline{\theta}} = [\, R + R_v \,]^{-1} [\, \underline{r} + \underline{r}_v \,] \qquad \{3.66\}$$

The first term on the right side of Eq. {3.66} can be simplified when the inverse exists.

$$\begin{aligned} [\, R + R_v \,]^{-1} &= [\, R\, [\, I + R^{-1} R_v \,]\,]^{-1} \\ &= [I + R^{-1} R_v]^{-1} R^{-1} \end{aligned} \qquad \{3.67\}$$

Substituting {3.67} into {3.66} and using $\underline{\theta} = R^{-1}\underline{r}$ from Eq. {3.19} yields;

$$\begin{aligned} \hat{\underline{\theta}} &= [\, I + R^{-1} R_v]^{-1} R^{-1} [\, \underline{r} + \underline{r}_v] \\ &= [\, I + R^{-1} R_v]^{-1} R^{-1} \underline{r} + [\, I + R^{-1} R_v]^{-1} R^{-1} \underline{r}_v \\ &= [\, I + R^{-1} R_v]^{-1} \underline{\theta} + [\, I + R^{-1} R_v]^{-1} R^{-1} \underline{r}_v \end{aligned} \qquad \{3.68\}$$

Note that when the measurement noise {v(n);S<=n<T} is equal to zero, R_v reduces to the null matrix, $\underline{r}_v$ reduces to the null vector, and Eq. {3.68} yields $\hat{\underline{\theta}} = \underline{\theta}$. We are interested in the noisy measurement case, and denote the distortion in the model coefficients as $\underline{\theta}_d$, where

$$\underline{\theta}_d = \hat{\underline{\theta}} - \underline{\theta} \qquad \{3.69\}$$

Substituting Eq. {3.69} into Eq. {3.68} yields an expression for this distortion in the model coefficients.

$$\underline{\theta}_d = [\, I + R^{-1}R_v]^{-1}\underline{\theta} + [\, I + R^{-1}R_v]\; R^{-1}\underline{r}_v - \underline{\theta}$$
$$= [\, I + R^{-1}R_v]^{-1}[\, \underline{\theta} + R^{-1}\underline{r}_v]^{-1} - \underline{\theta} \qquad \{3.70\}$$

Eq. {3.70} gives an exact expression for the coefficient distortion due to additive output noise, but its meaning is hard to appreciate directly because of the four inversions. The first term on the right side can be expanded in a geometric series;

$$[\, I + R^{-1}R_v]^{-1} = I - R^{-1}R_v + [\, R^{-1}R_v]^{+2} - [\, R^{-1}R_v]^{+3} + \ldots \qquad \{3.71\}$$

This series is valid when the absolute values of the eigenvalues of matrix $[\, R^{-1}R_v]$ are all less than 1. Matrix powers greater than one are negligible when the eigenvalues are small compared to one. These conditions are met when the total power of the additive noise is small in comparison to the total power of the system output; i.e. high SNR. Using this assumption, Eq. {3.71} is approximated by the first two terms of the expansion, and Eq. {3.70} becomes;

$$\underline{\theta}_d \cong [\, I - R^{-1}R_v][\, \underline{\theta} + R^{-1}\underline{r}_v] - \underline{\theta}$$
$$= \underline{\theta} - R^{-1}R_v\underline{\theta} + R^{-1}\underline{r}_v - R^{-1}R_vR^{-1}\underline{r}_v - \underline{\theta}$$
$$= R^{-1}[\, [\, I - R_vR^{-1}]\underline{r}_v - R_v\underline{\theta}] \qquad \{3.72\}$$

The above equation can be interpreted as describing the model coefficient distortion vector as composed of the difference of two vectors. One is a constant term, and the other is a multiplicative function of the noise-free model coefficients. Note that the only inversion needed for this approximation is that of the matrix R. Also, both vectors are directly proportional to the inverse of the matrix R,

which is independent of the particular additive noise characteristics.

The distortion on the estimates of the model coefficients is therefore the difference of two vector;

$$\underline{\theta}_c = R^{-1} [\underline{r}_v - R_v R^{-1} \underline{r}_v] = [I - R^{-1} R_v] R^{-1} \underline{r}_v \quad \{3.73\}$$

and

$$\underline{\theta}_m = R^{-1} R_v \underline{\theta} \quad \{3.74\}$$

where $\underline{\theta}_d = \underline{\theta}_c - \underline{\theta}_m$ {3.75}

This shows how the coefficient distortion of a linear recursive model depends upon the choice of the particular model terms, time averages of the system input and output, and time averages of the additive output noise.

As an illustrative example of the effect of noise on a linear recursive model, consider the ARMA(1,1) model;

$$y(n) = \theta_{1;0}(0)\, u(n) + \theta_{1;0}(1)\, u(n-1) + \theta_{0;1}(1)\, y(n-1) \quad \{3.76\}$$

The following matrices and vectors are written by inspection;

$$\underline{x}(n)^T = [u(n),\ u(n-1),\ y(n-1)] \quad \{3.77\}$$

$$\underline{x}_v(n)^T = [\,0,\ 0,\ v(n-1)] \quad \{3.78\}$$

$$X = \begin{bmatrix} u(n_2) & u(n_2-1) & y(n_2-1) \\ u(n_2+1) & u(n_2) & y(n_2) \\ \vdots & \vdots & \vdots \\ u(n_3) & u(n_3-1) & y(n_3-1) \end{bmatrix} \quad \{3.79\}$$

$$X_v = \begin{bmatrix} 0 & 0 & v(n_2-1) \\ 0 & 0 & v(n_2) \\ \vdots & \vdots & \vdots \\ 0 & 0 & v(n_3-1) \end{bmatrix} \quad \{3.80\}$$

$$\underline{r} = \begin{bmatrix} u(n_2) & u(n_2+1) & \dots & u(n_3) \\ u(n_2-1) & u(n_2) & \dots & u(n_3-1) \\ y(n_2-1) & y(n_2) & \dots & y(n_3-1) \end{bmatrix} \begin{bmatrix} y(n_2) \\ y(n_2+1) \\ \vdots \\ y(n_3) \end{bmatrix}$$

$$= \left[\begin{array}{c} \frac{1}{N}\sum_{n=n_2}^{n_3} u(n)y(n) \\ \hline \frac{1}{N}\sum_{n=n_2}^{n_3} u(n-1)y(n) \\ \hline \frac{1}{N}\sum_{n=n_2}^{n_3} y(n-1)y(n) \end{array}\right] \qquad \{3.81\}$$

$$\underline{r}_v = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ v(n_2-1) & v(n_2) & \dots & v(n_3-1) \end{bmatrix} \begin{bmatrix} v(n_2) \\ v(n_2+1) \\ \vdots \\ v(n_3) \end{bmatrix}$$

$$= \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline \frac{1}{N}\sum_{n=n_2}^{n_3} v(n)v(n-1) \end{array}\right] \qquad \{3.82\}$$

$$R = \begin{bmatrix} u(n_2) & u(n_2+1) & \dots & u(n_3) \\ u(n_2-1) & u(n_2) & \dots & u(n_2-1) \\ y(n_2-1) & y(n_2) & \dots & y(n_3-1) \end{bmatrix} \begin{bmatrix} u(n_2) & u(n_2-1) & y(n_2-1) \\ u(n_2+1) & u(n_2) & y(n_2) \\ \vdots & \vdots & \vdots \\ u(n_3) & u(n_3-1) & y(n_3-1) \end{bmatrix}$$

$$= \left[\begin{array}{c|c|c} \frac{1}{N}\sum_{n=n_2}^{n_3} u(n)^2 & \frac{1}{N}\sum_{n=n_2}^{n_3} u(n)u(n-1) & \frac{1}{N}\sum_{n=n_2}^{n_3} u(n)y(n-1) \\ \hline & \frac{1}{N}\sum_{n=n_2}^{n_3} u(n-1)^2 & \frac{1}{N}\sum_{n=n_2}^{n_3} u(n-1)y(n-1) \\ \hline \text{SYMMETRIC} & & \frac{1}{N}\sum_{n=n_2}^{n_3} y(n-1)^2 \end{array}\right] \qquad \{3.83\}$$

$$R_v = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & & 0 \\ v(n_2-1) & v(n_2) & \cdots & v(n_3-1) \end{bmatrix} \begin{bmatrix} 0 & 0 & v(n_2-1) \\ 0 & 0 & v(n_2) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0 & 0 & v(n_3-1) \end{bmatrix}$$

$$= \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & \frac{1}{N}\sum_{n=n_2}^{n_3} v(n-1)^2 \end{array}\right] \qquad \{3.84\}$$

$$R_v\underline{\theta} = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & \frac{1}{N}\sum_{n=n_2}^{n_3} v(n-1)^2 \end{array}\right] \left[\begin{array}{c} \theta_{1;0}(0) \\ \hline \theta_{1;0}(1) \\ \hline \theta_{0;1}(1) \end{array}\right]$$

$$= \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline \theta_{0;1}(1) \quad \frac{1}{N}\sum_{n=n_2}^{n_3} v(n-1)^2 \end{array}\right] \qquad \{3.85\}$$

Representing matrix R of Eq. {3.83} in the following shorthand;

$$R = \left[\begin{array}{c|c|c} a & b & c \\ \hline b & d & e \\ \hline c & e & f \end{array}\right] \qquad \{3.86\}$$

the inverse of matrix R can be written as shown below.

$$R^{-1} = \left[\begin{array}{c|c|c} g & h & k \\ \hline h & m & q \\ \hline k & q & s \end{array}\right] = \left[\begin{array}{c|c|c} (df - e^2)/|R| & (ce - bf)/|R| & (be - cd)/|R| \\ \hline (ce - bf)/|R| & (af - c^2)/|R| & (bc - ae)/|R| \\ \hline (be - cd)/|R| & (bc - ae)/|R| & (ad - b^2)/|R| \end{array}\right] \qquad \{3.87\}$$

where $|R| = adf + 2bec - c^2d - b^2f - e^2a$ $\{3.88\}$

Substituting {3.85} and {3.87} into {3.74} yields;

$$\underline{\theta}_m = \theta_{0;1}(1) \frac{1}{N} \sum_{n=n_2}^{n_3} v(n-1)^2 \begin{bmatrix} k \\ q \\ s \end{bmatrix} \qquad \{3.89\}$$

Substituting {3.82}, {3.84}, and {3.85} into {3.73} yields;

$$\underline{\theta}_c = \frac{1}{N} \sum_{n=n_2}^{n_3} v(n)v(n-1) \left[1 - \frac{s}{N} \sum_{n=n_2}^{n_3} v(n-1)^2 \right] \begin{bmatrix} k \\ q \\ s \end{bmatrix} \qquad \{3.90\}$$

Substituting {3.89} and {3.90} into {3.75} provides an expression for the distortion of the coefficient vector;

$$\underline{\theta}_d = \left\{ \frac{1}{N} \sum_{n=n_2}^{n_3} v(n)v(n-1) \left[1 - \frac{s}{N} \sum_{n=n_2}^{n_3} v(n-1)^2 \right] - \theta_{0;1}(1) \frac{1}{N} \sum_{n=n_2}^{n_3} v(n-1)^2 \right\} \begin{bmatrix} k \\ q \\ s \end{bmatrix} \qquad \{3.91\}$$

where k, q, and s are elements of the inverse of matrix R.

We can now directly examine the effect of additive noise on the coefficient distortion vector, in terms of the time averages of the additive noise. If the additive output noise is ergodic and uncorrelated with itself, the first term on the right side of Eq. {3.91} is small and will approach zero in the limit as $N \rightarrow \infty$. The magnitude of the coefficient distortion values will be directly proportional to both the sample autocorrelation of the output noise and the value of the recursive coefficient. Under this condition the coefficient distortion vector equals the negative of Eq. {3.89}. The value of $\underline{\theta}_d$ indicated by Eq. {3.91} has been observed in simulation experiments.

The preceding example serves to demonstrate the new insight that is available as a result of the development of equations {3.52} through {3.75}. The effect of the presence and properties of the additive noise on the distortion of the model coefficients follows directly from an examination of these equations. The impact of this will be addressed again in Chapters VI and VII.

From the preceding development of an expression for the distortion in the model coefficients resulting from additive output noise, an expression for the related increase in the fitting error can also be obtained. Substitute z(n) for y(n) in the development of Eq. {3.20}, denote the minimum error fitting criterion resulting from the noisy data as $\hat{J}$, and make use of {3.52} through {3.69}.

$$\hat{J}^2 = \frac{1}{N}\sum_{n=n_2}^{n_3} z(n)^2 - \hat{\underline{r}}^T\hat{\underline{\theta}}$$

$$= \frac{1}{N}[\,\underline{y} + \underline{v}\,]^T[\,\underline{y} + \underline{v}\,] - [\,\underline{r} + \underline{r}_v\,]^T[\,\underline{\theta} + \underline{\theta}_d\,]$$

$$= \frac{1}{N}\underline{y}^T\underline{y} + \frac{2}{N}\underline{v}^T\underline{y} + \frac{1}{N}\underline{v}^T\underline{v} - \underline{r}^T\underline{\theta} - \underline{r}^T\underline{\theta}_d - \underline{r}_v^T\underline{\theta} - \underline{r}_v^T\underline{\theta}_d \quad \{3.92\}$$

Denote the distortion in the minimum value of $\hat{J}^2$ as J_d^2.

$$J_d^2 = \hat{J}^2 - J^2 \quad \{3.93\}$$

Substituting {3.92} into {3.93} and using {3.20} yields;

$$J_d^2 = \frac{2}{N}\underline{v}^T\underline{y} + \frac{1}{N}\underline{v}^T\underline{v} - \underline{r}^T\underline{\theta}_d - \underline{r}_v^T\underline{\theta} - \underline{r}_v^T\underline{\theta}_d \quad \{3.94\}$$

Using the assumption that the additive noise {v(n)} is ergodic and independent of the system output {y(n)}, the first term on the right side of {3.94} is small, and

approaches zero as $N \rightarrow \infty$. Using this common assumption and substituting {3.72} into {3.94} yields;

$$J_d^2 = \frac{1}{N} \underline{v}^T \underline{v} - \underline{r}^T R^{-1} \{ [I - R_v R^{-1}] \underline{r}_v - R^T \underline{\theta} \} - \underline{r}_v^T \underline{\theta}$$

$$- \underline{r}_v^T R^{-1} \{ [I - R_v R^{-1}] \underline{r}_v - R_v \underline{\theta} \} \qquad \{3.95\}$$

From Eq. {3.19} we have $\underline{\theta}^T = \underline{r}^T R^{-1}$. Substituting this into Eq. {3.95} and simplifying gives;

$$J_d^2 = \frac{1}{N} \underline{v}^T \underline{v} - \underline{\theta}^T [\underline{r}_v - R_v R^{-1} \underline{r}_v - R_v \underline{\theta} + \underline{r}_v] - \underline{r}_v^T R^{-1} \underline{r}_v$$

$$+ \underline{r}_v^T R^{-1} R_v R^{-1} \underline{r}_v + \underline{r}_v^T R^{-1} R_v \underline{\theta}$$

$$= \frac{1}{N} \underline{v}^T \underline{v} + \underline{\theta}^T R_v \underline{\theta} - [\underline{r}_v^T R^{-1} + 2 \underline{\theta}^T] [\underline{r}_v - R_v R^{-1} \underline{r}_v] \qquad \{3.96\}$$

Equation {3.96} is a new expression for the distortion in the fitting error criterion of a linear recursive model caused by additive output noise. Note that if the model had been nonrecursive, vector $\underline{r}_v$ would reduce to the null vector, and matrix R_v would reduce to the null matrix. In this special case, Eq. {3.96} reduces to;

$$J_d^2 = \frac{1}{N} \underline{v}^T \underline{v} = \frac{1}{N} \sum_{n=n_2}^{n_3} v(n)^2 \qquad \{3.97\}$$

and the distortion in the fitting error criterion would be equal to the average power of the additive output noise, as expected.

The ARMA model of Eq. {3.76} is used again in an illustrative example of the effect of noise on the fitting error of a linear recursive model. Substituting Eq. {3.76} through {3.88} into {3.96} produces;

$$J_d^2 = \frac{1}{N}\sum_{n=n_2}^{n_3} v(n)^2 + \underline{\theta}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{N}\sum_{n=n_2}^{n_3} v(n-1)^2 \end{bmatrix} \underline{\theta}$$

$$- \begin{bmatrix} 0 & 0 & \frac{1}{N}\sum_{n=n_2}^{n_3} v(n)v(n-1) \end{bmatrix} \begin{bmatrix} g & h & k \\ h & m & q \\ k & q & s \end{bmatrix} + 2\underline{\theta}^T \cdot$$

$$\left\{ \begin{bmatrix} 0 \\ 0 \\ \frac{1}{N}\sum_{n=n_2}^{n_3} v(n)v(n-1) \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{N}\sum_{n=n_2}^{n_3} v(n-1)^2 \end{bmatrix} \begin{bmatrix} g & h & k \\ h & m & q \\ k & q & s \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{N}\sum_{n=n_2}^{n_3} v(n)v(n-1) \end{bmatrix} \right\}$$

$$= \frac{1}{N}\sum_{n=n_2}^{n_3} v(n)^2 + \left[\theta_{0;1}(1)\right]^2 \frac{1}{N}\sum_{n=n_2}^{n_3} v(n-1)^2$$

$$- \left[\frac{1}{N}\sum_{n=n}^{n} v(n)v(n-1)\right]^2 \left[s + 2\,\theta_{0;1}(1)\right]\left[1 - \frac{s}{N}\sum_{n=n_2}^{n_3} v(n-1)^2\right] \quad \{3.98\}$$

If the noise $\{v(n)\}$ is ergodic and uncorrelated with itself, the factor premultiplying the third term on the right side of Eq. {3.98} is small, and approaches zero in the limit as $N \longrightarrow \infty$. Using this common assumption, the fitting error distortion reduces to the following;

$$J_d = \frac{1}{N}\sum_{n=n_2}^{n_3} v(n)^2 + \left[\theta_{0;1}(1)\right]^2 \frac{1}{N}\sum_{n=n_2}^{n_3} v(n-1)^2$$

$$\cong \left[1 + \left[\theta_{0;1}(1)\right]^2\right] \frac{1}{N}\sum_{n=n_2}^{n_3} v(n)^2 \qquad \text{for large } N \quad \{3.99\}$$

The preceding equation shows that for uncorrelated additive output noise, the increase in the fitting error criterion is proportional to both the power of the additive noise, and one plus the square of the magnitude of the recursive coefficient. If the model form had more than one recursive term, the resulting equation for J_d^2 would appear more complex, but would follow a related form as this example.

The preceding development provides new insight to the actual effect of additive output noise on the characterization of linear recursive systems.

IV. EVALUATION OF MODEL EQUATIONS

A. EXISTING TECHNIQUES

Given a set of input and output measurements and a model equation that is a function of these measurements and linear in a set of coefficients, Chapter III showed how to obtain estimates for the coefficient values and the error residual. The literature reports [Ref. 17] that the ARMA model can reasonably represent most linear systems of interest using orders less than m = 10. A current problem is the efficiency of computation when larger and more general model forms like VOL and BVM are considered. Regardless of the degree or memory of the model, the calculation of the model fit involves solving the normal equations {3.15}.

This section discusses the traditional direct least squares model evaluation technique. The next section develops a unified solution technique for the more efficient recursive evaluation of a wide class of models. The last section compares the computational features of these evaluation techniques.

The traditional modeling technique starts by selecting a first model $y(n) = \underline{\theta}^T \underline{x}(n)$. We include the index parameter 1 to identify this first model, and write the prediction form equation as follows.

$$y(n,1) = \underline{\theta}(1)^T \underline{x}(n,1) + e(n,1) \qquad \{4.1\}$$

Using {4.1} in place of {3.5}, and following the same least squares development as Chapter III, yields the normal equations corresponding to Eq. {3.14}. The index parameter is included where needed.

$$\frac{1}{N}\left[X(1)^T X(1)\right]\underline{\theta}(1) = \frac{1}{N} X(1)^T \underline{y} \qquad \{4.2\}$$

This leads to the model error evaluation and coefficient estimation equations in terms of this indexed notation;

$$J^2(1) = \frac{1}{N}\underline{y}^T\underline{y} - \underline{r}(1)^T R(1)^{-1}\underline{r}(1) \qquad \{4.3\}$$

$$\underline{\theta}(1) = R(1)^{-1}\underline{r}(1) \qquad \{4.4\}$$

where

$$R(1) = \frac{1}{N} X(1)^T X(1) \qquad \{4.5\}$$

and

$$\underline{r}(i) = \frac{1}{N} X(i)^T \underline{y} \qquad \{4.6\}$$

If the fitting error $J^2(1)$ is too large for the application, the traditional systems identification technique is to select a larger model that contains the terms of the first model plus some additional terms. This second prediction form model is written as shown below.

$$y(n,2) = \underline{\theta}(2)^T \underline{x}(n,2) + e(n,2) \qquad \{4.7\}$$

The technique forms equations like {4.3} and {4.4} for the model of {4.7}, and continues until the fit $J^2(i)$ of model number i is within some acceptable limits. This is a brute-force and inefficient approach since the evaluation of the second (and subsequent models) does not take advantage of the solution calculated for the previous model(s).

To appreciate the above point, we digress momentarily to demonstrate the computational complexity (as measured by the number of multiplications or divisions) involved with calculating the inverse of the matrix R(i) when the model form is the BVM(d,m) introduced in Chapter II. Equation {2.9} gives the number of coefficients in a BVM as a function of the choice of the degree and the memory. Table 1 shows the number of coefficients, and therefore the size of the corresponding R(i) matrix, for any BVM of degree up to 6 and memory up to 10. In this chapter, the notation c(i) is used for the size of the $(i)^{th}$ model regardless of its form.

	d=1	d=2	d=3	d=4	d=5	d=6
m=0	1	2	3	4	5	6
m=1	3	9	19	34	55	83
m=2	5	20	55	125	251	461
m=3	7	35	119	329	791	1715
m=4	9	54	219	714	2001	5004
m=5	11	77	363	1364	4367	12375
m=6	12	104	559	2379	8567	27131
m=7	15	135	815	3875	15503	54263
m=8	17	170	1139	5984	26333	100946
m=9	19	209	1539	8854	42503	177099
m=10	21	252	2023	12649	65779	296009

TABLE 1: Number of coefficients in a BVM of degree d and memory m

The computational cost of inverting a matrix R(i) of size c(i), is of the order of 1/3 times the cube of c(i) multiplicative operations. Table 2 shows the approximate number of such operations required by the direct least squares technique for the inversion of the matrix R(i) corresponding to a BVM of degree d and memory m.

	d=1	d=2	d=3	d=4	d=5	d=6
m=0	1	3	9	22	42	72
m=1	9	243	2287	13100	55450	190600
m=2	42	2667	55460	651000	5271000	3.266E7
m=3	115	1.429E4	5.617E5	1.187E7	1.650E8	1.681E9
m=4	243	5.249E4	3.501E6	1.213E8	2.671E9	3.176E10
m=5	444	1.522E5	1.594E7	8.459E8	2.776E10	6.317E11
m=6	733	3.750E5	5.822E7	4.488E9	2.096E11	6.657E12
m=7	1125	8.200E5	1.804E8	1.940E10	1.242E12	5.326E13
m=8	1638	1.638E6	4.925E8	7.143E10	6.087E12	3.429E14
m=9	2287	3.043E6	1.215E9	2.314E11	2.560E13	1.852E15
m=10	3087	5.334E6	2.760E9	6.746E11	9.487E13	8.646E15

TABLE 2: Number of multiplication operations required for the matrix inversion involved in the direct least squares evaluation of a BVM of degree d and memory m.

It is clear that for degrees above 3, the inversion of the matrix R(i) required for this direct least squares evaluation of model i, rapidly becomes prohibitively expensive for increasing d or m. For problems of interest, however, we want to evaluate such higher degree and/or memory models of the BVM form.

It should be mentioned that for large c(i), the computation of the elements of the R(i) matrix requires approximately c(i)N operations. This can dominate the computation time if N >> c(i), as is typically the case in the literature. Even though the correct model forms were used in the three examples of experiment number 1, the Autocorrelation, Prewindowed, and Postwindowed methods still required a large N to obtain a small fitting error. On the other hand, the Covariance method gave superior performance without requiring N >> c(i). These results are typical of those obtained with other computer simulated experiments,

and indicate that an equivalent performing model solution can be obtained more economically with the Covariance method.

B. PRESENTATION OF A RECURSIVE EVALUATION TECHNIQUE

To develop efficient algorithms for evaluating models of the BVM form of equation {2.7}, we make a change in notation that will allow us to relate the equations and solutions of various models. This notational change is important for subsequent developments. We reorder $\underline{x}(n,i)$ and $\underline{\theta}(1)$, respectively, in a manner described below. We denote the reordered $\underline{x}(n,i)$ as $\underline{w}(n,i)$, and the reordered $\underline{\theta}(i)$ as $\underline{p}(i)$, such that Eq. {4.7} becomes;

$$y(n,2) = \underline{p}(2)^T \underline{w}(n,2) + e(n,2) \qquad \{4.8\}$$

where $$\underline{w}(n,2)^T = [\ \underline{w}(n,1)^T \ | \ \underline{w}(n,2/1)^T\ ] \qquad \{4.9\}$$

and $$\underline{w}(n,1) = \underline{x}(n,1) \qquad \{4.10\}$$

and $\underline{w}(n,2/1)$ is a vector formed by starting with $\underline{x}(n,2)$, deleting all of the terms that also exist in $\underline{w}(n,1)$, and reducing the size of the resultant vector by eliminating the spaces of any deleted elements.

and where $$\underline{p}(2)^T = [\ \underline{p}(1/2)^T \ | \ \underline{p}(2/1)^T\ ] \qquad \{4.11\}$$

and $$\underline{p}(1/2) = \underline{p}(1) \text{ evaluated at the } 2^{nd} \text{ iteration} \qquad \{4.12\}$$

and $\underline{p}(2/1)$ is a vector formed by starting with $\underline{\theta}(2)$, deleting all of the terms that also exist in $\underline{p}(1)$, and reducing the size of the resultant vector by eliminating the spaces of any deleted elements.

It remains to show that the evaluation of model equation {4.8} can be accomplished more efficiently than the evaluation of the same model given instead by Eq. {4.7}. Before demonstrating this result, the preceding notation is generalized for models beyond the first and second.

We recursively define an equation for model i of size c(i) in terms of model i-1 of size c(i-1), where c(i) > c(i-1).

$$y(n,i) = \underline{p}(i)^T \underline{w}(n,i) + e(n,i) \qquad \{4.13\}$$

where $\underline{w}(n,i)$ and $\underline{p}(i)$ are size c(i) vectors defined in the same manner as Eq. {4.8} through {4.12}, such that,

$$\underline{w}(n,i)^T = [\ \underline{w}(n,i-1)^T \mid \underline{w}(n,i/i-1)^T\ ] \qquad \{4.14\}$$

and

$$\underline{p}(i)^T = [\ \underline{p}(i-1/i)^T \mid \underline{p}(i/i-1)^T\ ] \qquad \{4.15\}$$

Following the standard least squares development yields the normal equations corresponding to equation {4.13}:

$$\frac{1}{N}\left[W(i)^T W(i)\right]\underline{p}(i) = \frac{1}{N} W(i)^T \underline{y} \qquad \{4.16\}$$

where we have the c(i) x N transposed data matrix;

$$W(i)^T = [\ \underline{w}(n_2,1) \quad \underline{w}(n_2+1,i) \quad \ldots \quad \underline{w}(n_3,i)\ ] \qquad \{4.17\}$$

The solution of {4.16} is the coefficient estimation equation;

$$\underline{p}(i) = [\ W(i)^T W(i)\ ]^{-1} W(i)^T \underline{y} \qquad \{4.18\}$$

The solution for the model fitting error criterion is;

$$J^2(i) = \frac{1}{N}\underline{y}^T\underline{y} - \underline{d}(i)^T D(i)^{-1} \underline{d}(i) \qquad \{4.19\}$$

where the c(i) x c(i) least squares matrix is;

$$D(i) = \frac{1}{N} W(i)^T W(i) \qquad \{4.20\}$$

and the c(i) x 1 vector $\underline{d}(i)$ is defined;

$$\underline{d}(i) = \frac{1}{N} W(i)^T \underline{y} \qquad \{4.21\}$$

Instead of solving Eq. {4.18} and {4.19}, we use {4.14} and {4.15} to develop a set of recursive model evaluation and model coefficient estimation equations[6]. Define q(i) to represent the number of terms in the (i)th model that are not contained in the (i-1)st model.

$$q(i) = c(i) - c(i-1) \qquad \{4.22\}$$

---

6 The recursive model evaluation equations presented in this chapter were independently developed; but they turn out to have many similar features with a published algorithm by Hsia [Ref. 14, pp 27]. Hsia's algorithm is designed for recursively computing the least squares coefficient estimates of a model as the number of terms in the model increases. The equations developed in this chapter have equivalent recursive capabilities for coefficient estimation, and also include some features that are used later in the thesis for the more general model growth problem.

The form of our equations turn out to be somewhat different since we have some matrices and vectors that do not appear explicitly in Hsia's algorithm. These matrices and vectors are basic to the development of the main model growth results of Chapter VI. We also include a recursive equation for the model fitting error based on a simple extension of our particular form of the coefficient evaluation equations, without explicitly solving for the model coefficient estimates. Using our equations, we are able to show how it reduces to a special but widely used efficient recursive parameter estimation technique (Durbin's [Ref. 2] modification of Levinson's Algorithm) when two restrictive (from a performance modeling growth perspective) assumptions are made. Since we need our form of the equations and these other features in the remainder of this thesis, the equations are developed at this point.

Substitute {4.14} into {4.17}, and define the following.

$$W(i) = \left[\begin{array}{c|c} \underline{w}(n_2,i-1)^T & \underline{w}(n_2,i/i-1)^T \\ \underline{w}(n_2+1,i-1)^T & \underline{w}(n_2+1,i/i-1)^T \\ \vdots & \vdots \\ \underline{w}(n_3,i-1) & \underline{w}(n_3,i/i-1)^T \end{array}\right]$$

$$= [\ W(i-1) \ | \ W(i/i-1)\ ] \qquad \{4.23\}$$

where $W(i)$ is the N x c(i) data matrix for the $(i)^{th}$ model, $W(i-1)$ is the N x c(i-1) data matrix for the $(i-1)^{st}$ model, and $W(i/i-1)$ is the N x q(i) data matrix for the new terms in the $(i)^{th}$ model that are not in the $(i-1)^{st}$ model.

Substituting Eq. {4.23} into Eq. {4.16} and simplifying;

$$\left[\begin{array}{c|c} A(i-1) & B(i/i-1) \\ \hline B(i/i-1)^T & A(i/i-1) \end{array}\right] \underline{p}(i) = \left[\begin{array}{c} \underline{h}(i-1) \\ \hline \underline{h}(i/i-1) \end{array}\right] \qquad \{4.24\}$$

where the following definitions are made for convenience.

$$A(i-1) = \frac{1}{N} W(i-1)^T W(i-1) \ , \text{ a } c(i-1) \text{ x } c(i-1) \text{ matrix} \qquad \{4.25\}$$

$$B(i/i-1) = \frac{1}{N} W(i-1)^T W(i/i-1) \ , \text{ a } c(i-1) \text{ x } q(i) \text{ matrix} \qquad \{4.26\}$$

$$A(i/i-1) = \frac{1}{N} W(i/i-1)^T W(i/i-1) \ , \text{ a } q(i) \text{ x } q(i) \text{ matrix} \qquad \{4.27\}$$

$$\underline{h}(i-1) = \frac{1}{N} W(i-1)^T \underline{y} \ , \text{ a } c(i-1) \text{ column vector} \qquad \{4.28\}$$

$$\underline{h}(i/i-1) = \frac{1}{N} W(i/i-1)^T \underline{y} \ , \text{ a } q(i) \text{ column vector} \qquad \{4.29\}$$

The set of linear equations {4.24} is a special permuted form of the normal equations for the $(i)^{th}$ model. This special form results from the ordering or $\underline{w}(n,i)$ described by Eq. {4.14}, and leads to an efficient set of recursive

solution equations for $J^2(i)$ and $\underline{p}(i)$. It also provides the basis for our unified approach to the model determination and growth problem examined in the next chapters.

Recursive Model Growth Solution and Evaluation Equations

The set of simultaneous linear equations {4.24} has a compact solution for $\underline{p}(i)$ and $J^2(i)$, based on the previously obtained $\underline{p}(i-1)$ and $J^2(i-1)$. Appendix A contains this development and we only state and use the results here. It is convenient to use the following definitions;

$$F(i) = -A(i-1)^{-1} B(i/i-1) \text{ , a } c(i-1) \text{ x } q(i) \text{ matrix} \quad \{4.30\}$$

$$G(i) = A(i/i-1) - B(i/i-1)^T A(i-1)^{-1} B(i/i-1)$$
$$= A(i/i-1) + B(i/i-1)^T F(i) \text{ , a } q(i) \text{ x } q(i) \text{ matrix} \quad \{4.31\}$$

$$\underline{g}(i) = \underline{h}(i/i-1) - B(i/i-1)^T \underline{p}(i-1)$$
$$= \underline{h}(i/i-1) + F(i)^T \underline{h}(i-1) \text{ , a } q(i) \text{ column vector} \quad \{4.32\}$$

$$\underline{k}(i) = G(i)^{-1} \underline{g}(i) \text{ , a } q(i) \text{ column vector} \quad \{4.33\}$$

$$\text{As long as } | A(i/i-1) | \neq 0 \quad \{4.34\}$$

then the solution of {4.24} is given by

$$\underline{p}(i) = \begin{bmatrix} \underline{p}(i-1) \\ \hline \underline{0} \end{bmatrix} + \begin{bmatrix} F(i) \\ \hline I \end{bmatrix} \underline{k}(i) \quad \{4.35\}$$

where $\underline{0}$ is the null vector and I is the identity matrix. The resulting minimum average sum squared error value is;

$$J^2(i) = J^2(i-1) - \underline{g}(i)^T \underline{k}(i) \quad \{4.36\}$$

In addition, the following recursive relationships exist;

$$A(i)^{-1} = \left[\begin{array}{c|c} A(i-1)^{-1} + F(i)G(i)^{-1}F(i)^T & F(i)G(i)^{-1} \\ \hline G(i)^{-1}F(i)^T & G(i)^{-1} \end{array}\right] \quad \{4.37\}$$

$$\underline{h}(i)^T = [\, \underline{h}(i-1)^T \mid \underline{h}(i/i-1)^T \,] \quad \{4.38\}$$

C. CAPABILITIES OF THE RECURSIVE EVALUATION TECHNIQUE

We now demonstrate some of the advantages of using equation {4.35} as an alternative solution to {4.18} for model i. We showed that evaluation of equation {4.18} for a BVM(d,m) requires the inversion of a matrix of size c(d,m) given by equation {2.9}. Examination of {4.30} through {4.38} reveals that only one smaller inversion of size q(i) need be performed to evaluate Eq. {4.35} and Eq. {4.36}. This is the inversion of G(i) required for the calculation of $\underline{k}(i)$ in Eq. {4.33}.

For a BVM, the size of matrix G(i) at the (i)th iteraion is given by the following equation.

$$\text{size of } G(i) = q(i) = c(d_i, m_i) - c(d_{i-1}, m_{i-1}) \qquad \{4.39\}$$

where d_i, m_i, d_{i-1}, and m_{i-1} are the BVM degree and memory at iterations i and i-1, respectively. The computational cost of recursively evaluating models using Eq. {4.35} and Eq. {4.36} is a function of the degree and memory of the various models 1,2,...,i-1,i.

Table 3 represents an example where we consider three different paths from the BVM with d=1 and m=1, to the BVM with d=4 and m=4. The paths are denoted with arrows and oversized letters. The order of complexity involved in this example is shown in Table 4. A direct evaluation of the BVM with d=4 and m=4 is given for comparison purposes.

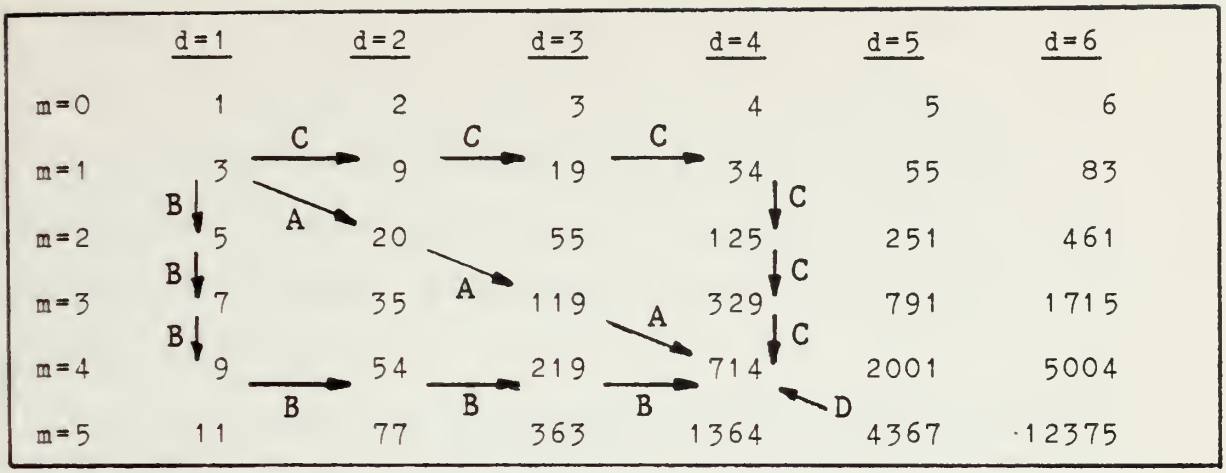


TABLE 3: Flow of four growth paths through the chart of the number of coefficients in a BVM of degree d and memory m.

Path	i	Model d_i, m_i	$c(d_i, m_i)$	Size of Matrix to be inverted $q(i)$	Inversion Operations $[q(i)]^3/3$	Total
A	1	(1,1)	3	3	9	
	2	(2,2)	20	17	1638	
	3	(3,3)	119	99	3.234E+5	
	4	(4,4)	714	595	7.021E+7	7.054E+7
B	1	(1,1)	3	3	9	
	2	(1,2)	5	2	3	
	3	(1,3)	7	2	3	
	4	(1,4)	9	2	3	
	5	(2,4)	54	45	3.037E+4	
	6	(3,4)	219	165	1.497E+6	
	7	(4,4)	714	495	4.043E+7	4.196E+7
C	1	(1,1)	3	3	9	
	2	(2,1)	9	6	72	
	3	(3,1)	19	10	334	
	4	(4,1)	34	15	1125	
	5	(4,2)	125	91	2.510E+5	
	6	(4,3)	329	204	2.830E+5	
	7	(4,4)	714	385	1.900E+7	2.208E+7
D	1	(4,4)	714	714	1.213E+8	1.213E+8

TABLE 4: Order of Complexity for four growth paths.

Paths A, B, and C each result in a lower total computational complexity than the direct evaluation of the BVM(4,4) model described by path D. Other paths are possible but this example is representative of the computational savings that result from the use of the recursive algorithm.

The development of the recursive algorithm is based on three assumptions.

(1) All model equations are linear in their respective coefficient vectors.

(2) The equation of the $(i)^{th}$ model includes all of the terms contained in the $(i-1)^{st}$ model, plus some new terms. This is described mathematically in equations {4.9}, {4.10}, and {4.14}.

(3) The determinant of A(i/i-1) is not zero.

We explicitly avoided any assumptions on the relationship between $\underline{w}(n,i-1)$, the terms in the $(i-1)^{st}$ model, and $\underline{w}(n,i/i-1)$, the new terms appearing in the $(i)^{th}$ model. This results in a general recursive solution algorithm that is applicable for any type of model growth we care to consider[7]. The following chapter shows that the

---

7 We still use the limitation on the form of each term that we defined in Eq. {3.6} for continuity of presentation, but mention here that other functional forms besides integer products of observations could be used as long as the resulting model equation is still linear in the unknown coefficients.

existing "recursive-in-order" algorithms (e.g. Levinson's [Ref. 2 - 4 and 20 - 25] and Lattice [Ref. 5 - 8 and 39] are special cases of the recursive evaluation equations presented here.

Chapter V develops several new techniques for specifying possible model term vectors $\underline{w}(n,i/i-1)$; for recursive growth using the BVM. These are less restrictive than existing techniques, and allow for more accurate and compact modeling of typical systems of interest.

V. TECHNIQUES FOR GROWING MODELS

A. OVERVIEW OF MODEL GROWTH

The objective of our analysis is to determine patterns or other key behavior properties from the measured data, and use this information to efficiently formulate a suitable mathematical model. This model relationship is evaluated against both its ability to predict behavior of the output time series within some reasonable and statistically quantifyable degree of accuracy, and its compactness of form.

Earlier work in model development was generally limited to an assumed linear relationship, and started with techniques like harmonic analysis and mathematical transform theory [Ref. 32, 33 and 34]. In the late 1960's, time series statistical analysis methods were developed by Box and Jenkins [Ref. 17]. These methods are related to transformations of the spectral methods, and approach the characterization problem from the different perspective of prediction form models [Ref. 35]. These techniques are not closed form solutions to the system characterization problem, but are instead multistage approaches that have been widely used for the time series analysis of real world systems [Ref. 36].

The Box and Jenkins technique assumes a general class of time series models which has been found, experimentally, to

be extremely rich. The procedure continues as a trial-and-error process with decision points where the analyst is required to select the next step based on the available information [Ref. 37].

Since existing linear time series techniques are not closed form solution algorithms permitting full analysis without human intervention and decisions, it would not be surprising that we are unable to find a complete closed form algorithm for the more general nonlinear relationship case. But nonlinear systems characterization is interesting and important, and a solution is still worth pursuing.

Chapter IV introduced a general, recursive set of equations for efficiently evaluating related sets of model equations. Efficiently handling the system characterization problem requires a method for determining what new model terms to add at each iteration; how to "grow" the model.

This chapter discusses existing techniques for recursive model growth (e.g. "recursive-in-order") which have been applied to some linear and nonlinear systems. Six variants on this type of "block-form" technique are developed for the more general BVM form, and the capabilities and limitations of all these techniques are investigated.

B. EXISTING TECHNIQUES FOR MODEL GROWTH

The systems identification literature [Ref. 2 - 8, 20 - 25, 38 and 39] contains two different techniques for both specifying and recursively estimating model coefficients,

and no explicit techniques for just the recursive evaluation of model fitting error. Both techniques are based on the concept of considering new model terms that have a unique "increasing order" relationship to existing model terms, and take advantage of the special Toeplitz matrix structure that can be made to occur in the resulting equations for the coefficient estimates [Ref. 2 - 4 and 20 - 25]. This Toeplitz structure is limited to a restrictive class of models, and requires the use of the Autocorrelation error minimization method. The iterative solution technique is based on Durbin's simpliciation of Levinson's algorithm [Ref. 2], and is well known in the literature. This technique is a special restrictive case of the general recursive algorithm presented in Chapter IV, and the details of the relationship are given in Appendix B.

Theorem 1 and Experiment 1 show that the use of the Autocorrelation method typically produces a suboptimal fit when the data sequences are finite. In terms of nonlinear models, the requirement for Toeplitz (or even Block-Toeplitz) structure for the least squares matrix severely limits the choice of allowable models. The "regular-form" kernel Nonlinear ARMA model used by Perry [Ref. 8] is a typical example of a restricted choice of terms in the model. This is discussed again in the next section.

The second recursive coefficient estimation and growth technique is based on "Lattice-filtering" [Ref. 5 - 8, 38 and 39]. This is a prediction-error version of Levinson's algorithm using a lattice structure implementation rather than the more conventional tapped delay line type of implementation. The error residual signal after each stage of the lattice has converged, is used in the computation of the lattice parameter estimates and signals used in subsequent stages. The lattice technique is a special implementation of the general algorithm of Chapter IV which has limited applicability. This technique offers no advantages for the unrestricted model growth we wish to consider.

C. RECURSIVE MODEL GROWTH WITH THE BVM

Chapter II introduced the BVM and showed that it subsumes the MA, AR, ARMA, Volterra, and Bilinear model forms. The recursive model evaluation and coefficient estimation algorithm presented in Chapter IV can be used for efficient and meaningful model growth of any of these model forms.

This section presents six extensions of the recursive-in-order techniques which apply directly to the general BVM form. In all cases, the first model is evaluated by a direct least squares fit using Eq {4.1} through Eq. {4.6}. Subsequent models are evaluated using the recursive

relationships presented in Eq {4.8} through {4.37}. We restricted the upper limit of the lag on the model terms to be equal to the memory m for both the input and output terms, for clarity of presentation. This restriction is removed in the model growth technique presented in the next chapter.

The first growth technique starts with the base model, BVM(d,m) = BVM(1,1) and uses the following fixed procedure. The fitting error J(d,m) = J(1,1) is evaluated for this first model, and for subsequent models BVM(1,1+i) where i=1,2,3,... until J(1,1+i) stops decreasing significantly (the meaning of which will be discussed later). This last significant model is denoted as the new base model BVM(1,m). The fitting error J(1+j,m) is evaluated for subsequent models BVM(1+j,m) where j=1,2,3,... until J(1+j,m) stops decreasing significantly. This last model is denoted as the new base model BVM(d,m) and the iteration starts on the memory m again. This two-phase iteration is continued as long as meaningful reduction in fitting error is obtained. This search strategy is denoted as the "M Directed Growth" because the initial phase involves growth in memory m (Fig. 21.a).

A second growth technique involves a similar algorithm with the difference that the first phase iteration is on the degree d (Fig. 21.b). This technique is therefore denoted as "D Directed Growth". The choice of an appropriate

significance test for switching between the two phases of these directed growth algorithms remains an open question.

A third growth method is denoted as "Diagonal Growth", and again starts with the evaluation of the base model BVM(d,m) = BVM(1,1). The fitting error of successive models BVM(1+i,1+i) for i=1,2,3,... is evaluated until J(1+i,1+i) is acceptably small (Fig. 21.c).

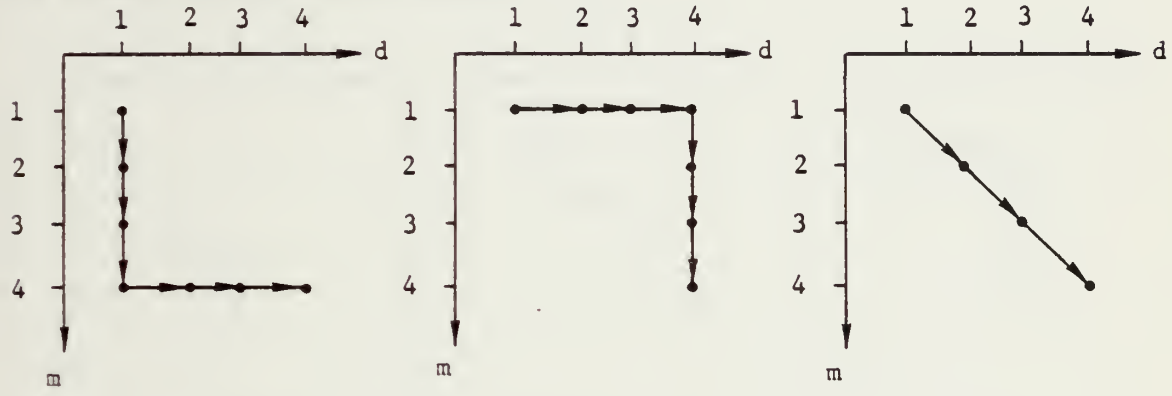


Figure 21.a Figure 21.b Figure 21.c

FIGURE 21: Three Growth Methods for the BVM

A fourth technique is denoted as "M-D Zig-Zag Growth" and starts with the evaluation of the base model BVM(1,1). The memory m and degree d are alternately incremented until the fitting error of the resulting model is acceptable (Fig. 22.c).

A fifth technique denoted as "D-M Zig-Zag Growth" involves a similar algorithm with the key difference that the first phase iteration is on the degree d (Fig. 22.b).

A sixth growth strategy is denoted as "Neighbor Growth". It starts with the base model BVM(1,1) and evaluates its

fitting error J(1,1). Next we evaluate models that differ from the base model by one increment of degree, one increment of memory, or both. In this starting case we evaluate J(1,2), J(2,1), and J(2,2). We denote the model with the lowest fitting error J as the new base model and continue this iteration process until the decrease in J is no longer significant (Fig. 22.c).

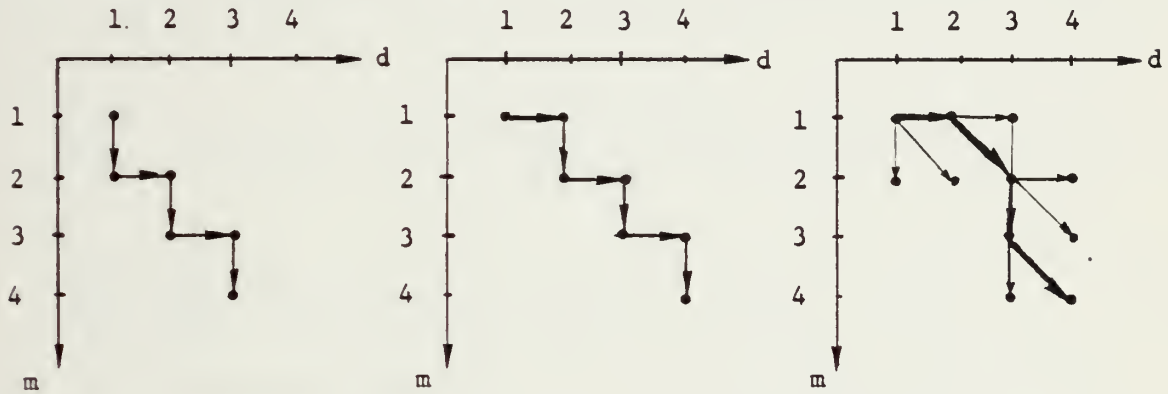


Figure 22.a Figure 22.b Figure 22.c

FIGURE 22: Three Additional Growth Methods for the BVM

These six techniques all fall under the general type of growth we refer to as "block-form" techniques. At each iteration the model growth is accomplished by automatically including one of a predetermined set of model terms.

One other model growth technique has been proposed by Perry [Ref. 8]. A special and restricted form of the nonlinear ARMA model (earlier version of the BVM) and the Autocorrelation error minimization method are used to develop a special multichannel lattice form parameter

estimation solution. Reference 8 does not contain any experimental verification of this technique. We analyze this technique to show its intrinsic weaknesses.

A quadratic nonlinear ARMA model of this special form can be written as shown below (a translation of Ref 5, pp. 193, Eq {4.29} into the notation of this thesis).

$$y(n) = \sum_{h_1=0}^{M_1} \theta_{1;0}(h_1)u(n-h_1) + \sum_{h_2=0}^{M_2} \sum_{h_1=0}^{M_1} \theta_{2;0}(h_1,h_2)u(n-h_1)u(n-h_1-h_2)$$

$$+ \sum_{h_1=1}^{M_1} \theta_{0;1}(h_1)y(n-h_1) + \sum_{h_3=0}^{M_3} \sum_{h_1=0}^{M_1} \theta_{0;2}(h_1,h_3)y(n-1-h_1)y(n-1-h_1-h_3)$$

$$+ \sum_{h_4=0}^{M_4} \sum_{h_1=0}^{M_1} \theta_{1;1}(h_1,h_4)u(n-h_1)y(n-1-h_1-h_4) + e(n) \qquad \{5.1\}$$

The proposed technique of Reference 8 requires that the user prespecify the integer values of the upper summation limits M_2, M_3, and M_4. Then Eq. {5.1} becomes a recursive-in-order M_1 model equation that can partially evaluated by least squares lattice techniques. Some of the model terms do not fit the restrictions of the lattice solution and must be evaluated separately by direct least squares. Despite the attractiveness of the potentially efficient lattice form, the requirement to prespecify all but one of the upper summation limits of Eq {5.1} excessively complicates any systematic model growth with this method. Reference 8 does not provide any suggestions on how to select these upper limits. Since this technique automatically involves a fixed

set of terms at each iteration, it falls under our definition as a block-form technique. It basically is an attempt at fitting a problem (parameter estimation) to a particular form of solution (lattice form recursive-in-order), and is inferior to the growth methods of this chapter and the following.

These block-form techniques can all eventually subsume any system of the BVM form. They unfortunately require a significant amount of computations as the degree and memory increase (See section VI.D). The first five techniques are all nonlinear extension of the linear recursive-in-order concept discussed earlier, and are brute-force methods. The Neighbor Growth technique offers an appealing approach for potentially more efficient model growth, but also suffers from the problem of high computational cost when used to evaluate the addition of a large number of model terms. We continue the discussion of efficient model growth along a related line in the next chapter.

VI. SEARCH INDICATORS FOR EFFICIENT MODEL GROWTH

A. DISCUSSION

The block-form growth techniques of Chapter V follow from the conventional concept of model growth in linear systems. Closer examination reveals that these techniques have a number of serious flaws when used for nonlinear model growth. Given a finite amount of measurement data, we can only consider evaluating models when the number of model terms is less than or equal to the number of data samples. Table 1 reveals that many of these nonlinear models can only be evaluated with extremely long sequences of data measurements. For example, the model BVM(3,5) has 363 different terms. Only a few of these terms may be needed in modeling any third degree dynamic system whose equation involves the factor u(n-5) or y(n-5). However, all of these 363 terms are involved in the full model evaluation when the block form growth techniques are used, and sufficient data measurements must be available. The following points summarize the results of many computer simulated model growth experiments (See Chapter VII later).

As the degree or memory increase, all block-form modeling techniques automatically consider an increasing number of terms at each subsequent growth iteration. This results in rapidly increasing computational cost, and often

produces an ill-conditioned least squares matrix A(i) due to the inclusion of several model terms with nearly equivalent properties in terms of values in this matrix. By ill-conditioned, we mean the condition number (the ratio of the largest to smallest eigenvalue) is numerically large (e.g. greater than 10000).

A higher condition number for matrix A(i) is related to less accurate estimates for the coefficients $\underline{p}(i)$, and higher fitting error $J^2(i)$. In some cases the matrix A(i) becomes so ill-conditioned that it is no longer positive definite, and the resulting model evaluation is no longer optimum in any least squares sense. It has been experimentally verified that the general use of these block-form modeling techniques often produces poor results for nonlinear systems; namely offset model coefficient estimates, high fitting error, and the inclusion of terms that are not actually needed. An example is provided in Chapter VII.

One possible approach to overcoming the preceding problems is to start with some base model such as BVM(1,1), and use one of the block-form techniques just to specify a new set of q(i) model terms. Instead of evaluating these candidate terms as a set, assume that only a subset of them is actually needed. The problem is how to find the particular subset of these terms required in the final model.

This approach is related to standard stepwise regression analysis [Ref. 40], and also to a recently published technique known as GMDH, the Group Method of Data Handling [Ref. 41]. These preceding techniques are general enough to permit consideration of a wide variety of model terms and both can avoid the ill-conditioned solution, but they still have the following major problem. With q(i) potential new model terms, there are $2^{q(i)}$ possible model equations to consider. Except for small values of q(i), the exhaustive evaluation of each of the corresponding solution equations rapidly becomes prohibitively expensive.

There is the additional problem of a stopping criterion. Examination of Eq. {4.33} and Eq. {4.36} shows that the fitting error $J^2(i)$ is monotonically decreasing when new model terms are added and matrix $G(i)^{-1}$ is positive definite. Therefore, while only some number r out of these particular q(i) candidate model terms may be needed in the final model, the fitting error with r+1 terms will still be lower (with the exception of numerical errors or an exact model fit).

In all practical situations, we have only finite measurement data and finite computer resources, and are left with an interesting and not unusual problem. When the preceding model growth techniques yield ill-conditioned solutions or increase to a point (1) greater than the finite data can handle, or (2) beyond the computational resources;

are there any prudent procedures that we can employ? The field of artificial intelligence has provided some motivation in related problems including chess playing programs and voice-recognition methods. The basis for comparison is the intractability of the exhaustive solution when there is only finite data, time, and computational power. Since the performance modeling problem is interesting, and has practical applications, we develop a semi-heuristic technique to follow when there are not enough resources for the exhaustive solution.

This chapter presents the new concept of "search indicators" for efficiently growing a model of an unknown linear or nonlinear system from a finite set of input-output measurements. Rather than attempting to solve the typically large set of normal equations for all of the new candidate model terms, the proposed concept uses an easily computable search indicator (scalar value), or a set of such search indicators, for each candidate model term contained in $\underline{w}(n,i/i-1)$ at each growth iteration i. The relative values of these search indicators are used to systematically exclude those terms expected[8] to have insignificant effect

8 The word expected is used to acknowledge the heuristic nature of some of these search indicators and of their use. We have been unable to prove that any technique based on their use is guaranteed to pick the optimum model terms at each growth iteration. These indicators are logical factors based on the recursive model evaluation equations, and the results of many experiments show that techniques using some search indicators provide for highly efficient model growth.

on reducing the fitting error. The remaining term or terms are retained in the vector $\underline{w}(n,i/i-1)$ for this $(i)^{th}$ model, and used to form a much smaller set of normal equations that are efficiently evaluated by the general recursive equations presented in Chapter IV.

This proposed two-phase concept offers a number of improved capabilities over the existing growth techniques.

(1) Since we compute the search indicators for each candidate model term seperately, we can consider more potential model terms than the number of data measurements. As a result, the terms of nonlinear models with large degree and memory can now be considered. We must, of course, eliminate enough terms such that the reduced model form evaluated in the second phase has fewer unknowns than the number of data measurements.

(2) This technique allows the evaluation of widely different model terms at any iteration. Unlike the recursive-in-order (or more general block-form) techniques, there is no longer the implicit restriction that the current model contain all of the possible set of input, output, and bivariate terms specified by a particular degree and memory.

(3) The initial set of model terms $\underline{w}(n,1)$ can be better chosen by the search indicator concept, rather than by blindly picking a predefined base

model like BVM(1,1). The computer simulated experiments of Chapter VII show that this property allows for the efficient characterization of a general class of systems having an input-output delay L (e.g. where terms containing the factor u(n-k) for k=0,1,2,...,L-1 are not needed in the final model). In cases where the system under consideration has a delay factor L, the block-form techniques fail to recognize and exploit this property, and often converge on a more complex model.

(4) The search indicator technique selects one or more candidate model terms in the first phase, produces a much smaller matrix A(i/i-1), and therefore significantly reduces the computational burden. It is also capable of efficiently handling the previously discussed problem of ill-conditioning caused by nearly equivalent model terms.

These features are demonstrated in the following sections.

The next section defines various possible search indicators based on the signals, vectors, and matrices contained in the recursive model growth solution and evaluation equations introduced in Chapter IV. Some physical interpretation is given for each of these search indicators, and the set is reduced to a smaller set worthy of further investigation. Results of many computer

simulated experiments have shown the superior growth capabilities of the proposed concept, and confirmed that the search indicator technique provides significant computational savings and accuracy improvements over all of the previously discussed growth techniques. Examples of model growth are provided in the computer simulated and real world experiments of Chapter VII.

B. DEVELOPMENT OF SEARCH INDICATORS

The notation $w_j(n,i/i-1)$ is used to represent the $(j)^{th}$ model term (out of the candidate set of terms) that we consider adding at the $(i)^{th}$ iteration, given that we have previously evaluated and accepted a model at the $(i-1)^{st}$ iteration. We let q(i) still represent the number of candidate model terms considered at the $(i)^{th}$ iteration, and therefore $j=1,2,3,\ldots,q(i)$.

The following development is partially based on the notation for the signals, vectors, and matrices contained within the recursive solution and evaluation equations of Chapter IV. One important note of clarification needs to be made at this point to minimize potential confusion.

The set of equations {4.8} through {4.38} in Chapter IV was developed to evaluate the improvement in model fitting error and calculate the new coefficient estimates based on adding the entire candidate set of terms $\underline{w}(n,i/i-1)$ to the model with the existing set of terms $\underline{w}(n,i-1)$. The set of search indicators developed in this chapter is to be

calculated for each of the candidate model terms $w_j(n,i/i-1)$ in the candidate set $\underline{w}(n,i/i-1)$. These indicators are designed to each give some partial metric or measure for the improvement in model fitting error. As a result of this development, many of the matrices and vectors defined in Eq. {4.8} through Eq. {4.38} for the evaluation of multiple model terms are used in this chapter in a reduced form (e.g. vectors and scalars, respectively) for the search indicator evaluation of each term. Whenever possible we use the lower case vector version of the matrix designation to represent the corresponding reduced form vector (e.g. $\underline{w}_j(i/i-1)$ for $W(i/i-1)$). Likewise we use the scalar representation to describe the corresponding reduced form of a vector (e.g. $h_j(i/i-1)$ for $\underline{h}(i/i-1)$).

These reductions are made only for clarity in the development of the search indicators. Once a subset of model terms is selected by the search indicators, the full form equations of Chapter IV are used to evaluate the fitting error and coefficient estimates. It is noted, however, that some of the factors calculated in the evaluation of the search indicators for the candidate model terms can be used again in the actual evaluation of the model performance. Thus, some of these computations will serve double duty.

A primary concern is efficiency of computation, so the numerical complexity (number of multiplications and

divisions) involved in calculating each search indicator has been analyzed. The notational convention O(N) will be used to denote N multiplication or division operations. This complexity notation is included with the development of each search indicator, and summarized with examples in Table 5 and Table 6 after the development of all of the indicators.

Denote the size c(i-1) of the $(i-1)^{st}$ model as P, and the number of data points in the error minimization as N. Since we have completed the the evaluation of the $(i-1)^{st}$ model, the following matrices and vectors are available.

W(i-1) = a N x P matrix given by Eq. {4.23}

A(i-1) = a P x P matrix given by Eq. {4.25}

$\underline{h}(i-1)$ = a P x 1 column vector given by Eq. {4.28}

We also have $A(i-1)^{-1}$ and $\underline{p}(i-1)$, the coefficient vector.

Some preliminary vectors needed for the development of the search indicators are presented at this point.

$$\underline{w}_j(i/i-1)^T = [w_j(n_2,i/i-1), w_j(n_2+1,i/i-1), \ldots, w_j(n_3,i/i-1)] \quad \{6.1\}$$

= a N x 1 transposed vector of the signal specified by the $(j)^{th}$ candidate model term over the interval (n_2,n_3). This is the reduced version of the data matrix W(i/i-1) given by Eq. {4.23}, in the case of just the $(j)^{th}$ candidate model term.

$$e(n,i-1) = y(n) - \underline{w}(n,i-1)\,\underline{p}(i-1) \quad \text{for } n_2 <= n <= n_3 \quad \{6.2\}$$

= value of the error residual at discrete time n from the $(i-1)^{st}$ model iteration. This can be computed with P multiplications per point.

$$\underline{e}(i-1)^T = [\ e(n_2,i-1),\ e(n_2+1,i-1),\ldots,\ e(n_3,i-1)\ ] \qquad \{6.3\}$$

= a N x 1 column vector of the residual sequence values from the $(i-1)^{st}$ iteration over the interval (n_2,n_3). This requires PN multiplications.

$$\underline{b}_j(i/i-1)^T = \frac{1}{N}\, W(i-1)^T \underline{w}_j(i/i-1) \qquad \{6.4\}$$

= the reduced version of matrix B(i/i-1) given by Eq. {4.26} in the case of just the $(j)^{th}$ candidate model term. Since this is a P x N matrix times a N x 1 column vector, the cost is O(PN+1).

$$\underline{f}_j(i/i-1)^T = -A(i-1)^{-1}\underline{b}_j(i/i-1) \qquad \{6.5\}$$

= the reduced version of matrix F(i) given by Eq. {4.30} in the case of just the $(j)^{th}$ candidate model term. Since $A(i-1)^{-1}$ is a P x P matrix that we already computed, and $\underline{b}_j(i/i-1)$ is a P x 1 column vector obtained in Eq. {6.4} at a cost of O(PN+1), the total cost of computing $\underline{f}_j(i/i-1)$ is $O(P^2) + O(PN+1) = O(P^2+PN+1)$.

Twelve different search indicators were developed and examined in this work. The initial set of search indicators I(j,1) through I(j,8) was developed from an algebraic perspective; i.e. these relationships arose from an examination of the general recursive evaluation equations of Chapter IV, under the condition of adding a single new model term $w_j(n,i/i-1)$. Each search indicator therefore has a direct relationship to the actual system characterization experiment.

The first search indicator is the time average of the product of the signal specified by the $(j)^{th}$ candidate model term, and the output signal of the system.

$$I(j,1) = \frac{1}{N} \underline{w}_j(i/i-1)^T \underline{y} = \frac{1}{N} \sum_{n=n_2}^{n_3} w_j(n,i/i-1)y(n) \qquad \{6.6\}$$

Since $\underline{w}_j(i/i-1)$ is a N x 1 vector and $\underline{y}$ is a N x 1 vector, the calculation of $I(j,1)$ requires $O(N+1)$ operations for each candidate model term.

This indicator corresponds to the scalar version $h_j(i/i-1)$ of the vector $\underline{h}(i/i-1)$ defined by Eq. {4.29}. While intuitively appealing as the "empirical" cross-correlation between the output of the system under test and the signal specified by the candidate model term, this indicator has a basic flaw. It is a function only of the output of the system and the candidate model term, and as such, does not depend on the particular terms in the previous model. Numerous computer simulated experiments have verified that $I(j,1)$, taken alone, is unsuitable as a reliable search indicator for model growth.

The second search indicator is the value corresponding to the reduced version $g_j(i)$ of the vector $\underline{g}(i)$ of Eq. {4.32}, in the case of just the $(j)^{th}$ candidate model term.

$$I(j,2) = \frac{1}{N} \underline{w}_j(i/i-1)^T \underline{y} + \underline{f}_j(i/i-1)^T \underline{h}(i-1)$$

$$= I(j,1) + \underline{f}_j(i/i-1)^T \underline{h}(i-1) \qquad \{6.7\}$$

where $\underline{w}_j(i/i-1)$ is a N x 1 vector and $\underline{y}$ is a N x 1 vector.

Since $\underline{f}_j(i/i-1)$ is a P x 1 vector obtained at the cost $O(P^2+PN+1)$, and $\underline{h}(i-1)$ is a P x 1 vector, the calculation of $I(j,2)$ requires a total of $O(P^2+PN+P+N+2)$ operations for each candidate model term.

We digress momentarily to examine some of the characteristics of the full vector $\underline{g}(i)$ given by Eq. {4.32}.

Substituting {4.26} and {4.29} into {4.32} produces;

$$\begin{aligned}\underline{g}(i) &= \frac{1}{N} W(i/i-1)^T \underline{y} - \frac{1}{N} W(i/i-1)^T W(i-1)\underline{p}(i-1) \\ &= \frac{1}{N} W(i/i-1)^T [\, \underline{y} - W(i-1)\underline{p}(i-1) \,] \\ &= \frac{1}{N} W(i/i-1)^T \underline{e}(i-1) \\ &= \frac{1}{N} \underline{e}(i-1)^T W(i/i-1) \end{aligned} \qquad \{6.8\}$$

Since $\{e(n,i-1)\}$ is the prediction error sequence of the $(i-1)^{st}$ model, and the vector $\underline{e}(i-1)$ contains the values of $\{e(n,i-1)\}$, we see that $\underline{g}(i)$ is a vector whose $(j)^{th}$ element is the normalized inner product of $\underline{e}(i-1)$ and the $(j)^{th}$ column of $W(i/i-1)$. Examination of Eq. {4.23} and Eq. {6.1} reveals that the $(j)^{th}$ column of $W(i/i-1)$ is the vector $\underline{w}_j(i/i-1)$, and yields the following expression.

$$\begin{aligned} g_1(i) &= \frac{1}{N} \underline{e}(i-1)^T \underline{w}_1(i/i-1) \\ g_2(i) &= \frac{1}{N} \underline{e}(i-1)\ \underline{w}_2(i/i-1) \\ &\vdots \\ g_{q(i)}(i) &= \frac{1}{N} \underline{e}(i-1)^T \underline{w}_{q(i)}(i/i-1) \end{aligned} \qquad \{6.9\}$$

where $\underline{g}(i)^T = [\, g_1(i), g_2(i), \ldots, g_j(i), \ldots, g_{q(i)}(i) \,]$ {6.10}

and where $q(i) = c(i) - c(i-1)$ {6.11}

Examination of Eq. {6.9} shows that the value $g_j(i)$ is the time average of the product of the error residual signal and a signal formed by products and powers of products, of the input-output measurements corresponding to the specification of the $(j)^{th}$ candidate model term. This gives physical interpretation and increased meaning to the value $g_j(i)$, which is contained in Eq. {6.7} (search indicator two), and equivalently in Eq. {6.12} below as search indicator three.

$$I(j,3) = \frac{1}{N} \underline{w}_j(i/i-1)^T \underline{e}(i-1) = \frac{1}{N} \sum_{n=n_2}^{n_3} w_j(n,i/i-1)e(n,i-1) = g_j(i) \quad \{6.12\}$$

Since $\underline{w}_j(i/i-1)$ is a N x 1 vector and $\underline{e}(i-1)$ is a N x 1 vector obtained at the cost O(PN), the calculation of I(j,3) requires O(PN+N+1) operations for the first candidate model term. For the second and subsequent candidate model terms the cost is reduced to O(N+1) since $\underline{e}(i-1)$ has already been calculated. Tables 5 and 6 show that I(j,3) can be computed much more efficiently than I(j,2). It is also intuitively appealing to be using the error residual from the previous model in evaluating the usefulness of a new candidate model term.

If the $(i-1)^{st}$ model produces an exact match to the measured input-output data {u(n)} and {y(n)}, then $g_j(i)$ is zero. This occurs regardless of the choice of the new term

$w_j(n,i/i-1)$ considered for inclusion in the $(i)^{th}$ model. It follows that the absolute value of $g_j(k)$ should be a useful measure at any step $k < i$ in the growth iteration. It is conjectured that this represents a measure of the relative benefit of that particular model term as compared with other possible choices of terms. In this regard the term that produces the largest absolute value of $g_j(k)$ would also probably result in the smallest value of $J^2(i)$, the error fitting criteria. This last point remains to be demonstrated.

The above discussion indicates that $I(j,3)$ should be a potentially good search indicator, either alone, or in combination with other factors. We will later consider other search indicators based on $g_j(i)$.

The fourth search indicator $I(j,4)$ is the time average of the square of the signal specified by the $(j)^{th}$ candidate model term.

$$I(j,4) = \frac{1}{N} \underline{w}_j(i/i-1)^T \underline{w}_j(i/i-1) = \frac{1}{N} \sum_{n=n_2}^{n3} \left[w_j(n,i/i-1)\right]^2 \quad \{6.13\}$$

Since $\underline{w}_j(i/i-1)$ is a N x 1 vector, the calculation of $I(j,4)$ requires $O(N+1)$ operations for each candidate model term. This corresponds to the reduced version of matrix $A(i/i-1)$ given by Eq. {4.25}. It can be efficiently computed, but suffers the same flaws as $I(j,1)$.

The fifth search indicator is the scalar corresponding to the reduced version of matrix $G(i)$ given by Eq. {4.32} in the case of one additional coefficient in the $(i)^{th}$ model.

$$I(j,5) = \frac{1}{N} \underline{w}_j(i/i-1)^T \underline{w}_j(i/i-1) + \underline{b}_j(i/i-1)^T \underline{f}_j(i/i-1)$$

$$= I(j,4) + \underline{b}_j(i/i-1)^T \underline{f}_j(i/i-1) \qquad \{6.14\}$$

Since $\underline{b}_j(i/i-1)$ is a P x 1 vector, and $\underline{f}_j(i/i-1)$ is a P x 1 vector, the cost of computing $\underline{f}_j(i/i-1)$ is $O(P^2+PN+1)$ and includes the cost of computing $\underline{b}_j(i/i-1)$. Therefore the calculation of I(j,5) requires a total of $O(P^2+PN+P+N+2)$ operations for each candidate model term. Examination of Eq. {4.33 and {4.36} reveals that the scalar value I(j,5) is inversely related to the reduction in the fitting error that results if the single candidate model term is brought into the model. As such, there is reason to expect that I(j,5) would be a good search indicator, either alone or in combination with other factors. Unfortunately, the high cost for I(j,5) precludes its general use.

The sixth search indicator is the scalar value corresponding to the reduced version of vector $\underline{k}(i)$ from Eq. {4.33} in the case of one additional model term.

$$I(j,6) = I(j,3)/I(j,5)$$

$$= \frac{\frac{1}{N} \underline{w}_j(i/i-1)^T \underline{e}(i-1)}{\frac{1}{N} \underline{w}_j(i/i-1)^T \underline{w}_j(i/i-1) + \underline{b}_j(i/i-1)^T \underline{f}_j(i/i-1)} \qquad \{6.15\}$$

where $\underline{w}_j(i/i-1)$ is a N x 1 vector, $\underline{e}(i-1)$ is a N x 1 vector obtained at the cost O(PN), and $\underline{f}_j(i/i-1)$ is a P x 1 vector obtained at the cost $O(P^2+PN+1)$ which includes the cost of computing the P x 1 vector $\underline{b}_j(i/i-1)$. Therefore, the

calculation of I(j,6) requires $O(P^2+2PN+2N+P+4)$ operations for the first candidate model term. For the second and subsequent candidate model terms the cost is reduced to $O(P^2+PN+2N+P+4)$ since $\underline{e}(i-1)$ has already been calculated.

Examination of Eq. {4.36} reveals that the value of I(j,6) is directly related to the reduction in the fitting error $J^2(i)$ that results from including the candidate term in the model. Tables 5 and 6 indicate, however, that there is a very high computational cost associated with this search indicator.

The seventh search indicator is the value of the change in the error criterion $J^2(i)$ as a result of including the candidate model term. It is based on Eq. {4.32}, {4.33}, and {4.36}.

$$I(j,7) = I(j,2)/I(j,5)$$

$$= \frac{\frac{1}{N}\ \underline{w}_j(i/i-1)^T\underline{y} + \underline{f}_j(i/i-1)^T\underline{h}(i-1)}{\frac{1}{N}\ \underline{w}_j(i/i-1)^T\underline{w}_j(i/i-1) + \underline{b}_j(i/i-1)^T\underline{f}_j(i/i-1)} \quad \{6.16\}$$

where $\underline{w}_j(i/i-1)$ is a N x 1 vector, $\underline{y}$ is a N x 1 vector, $\underline{h}(i-1)$ is a P x 1 vector, and $\underline{f}_j(i/i-1)$ is a P x 1 vector obtained at the cost $O(P^2+PN+1)$ which includes the cost of computing the P x 1 vector $\underline{b}_j(i/i-1)$. Therefore the total cost of computing I(j,7) requires $O(P^2+PN+2P+2N+5)$ operations for each candidate model term.

This is the <u>exact</u> value of the reduction in the error criterion resulting from including the candidate model term.

As such, it probably should not be called an "indicator". It is included as a control indicator since it has the desired property of exactly describing the performance improvement. Tables 5 and 6 show that this indicator is extremely expensive to compute. We next reduce the computational complexity using I(j,3).

The eighth search indicator is the value of the change in the error criterion $J^2(i)$, as a result of including the candidate model term and using the error residual signal of the model from the previous growth iteration. It is based on Eq. {6.8}, {4.33}, and {4.36}. I(j,8) has the following form;

$$I(j,8) = I(j,3)^2/I(j,5)$$

$$= \frac{\left[\frac{1}{N}\ \underline{w}_j(i/i-1)^T\underline{e}(i-1)\right]^2}{\frac{1}{N}\ \underline{w}_j(i/i-1)^T\underline{w}_j(i/i-1) + \underline{b}_j(i/i-1)^T\underline{f}_j(i/i-1)} \quad \{6.17\}$$

where $\underline{w}_j(i/i-1)$ is a N x 1 vector, $\underline{e}(n-1)$ is a N x 1 vector obtained at the cost O(PN), and $\underline{f}_j(i/i-1)$ is a P x 1 vector obtained at the cost $O(P^2+PN+1)$ which includes the cost of computing the P x 1 vector $\underline{b}_j(i/i-1)$. Therefore the calculation of I(j,8) requires $O(P^2+2PN+2N+P+5)$ operations for the first candidate model term. For the second and subsequent candidate model terms the cost is reduced to $O(P^2+PN+2N+P+5)$ since $\underline{e}(i-1)$ has already been calculated. This is the exact value of the reduction in the error criterion resulting from including the candidate term, using

the alternate and less costly computation for $g_j(i)$ discussed previously. Unfortunately the cost of computing the denominator of Eq. {6.17} predominates, and we have an alternative, but still costly, directly related search indicator.

The next three search indicators were developed in an attempt to recognize some additional factors that could be used to reduce the computational burden of the original set. Their physical interpretations are not as clear, but they are logical extensions to consider.

The ninth search indicator is the value of the L2-norm of the vector $\underline{b}_j(i/i-1)$ given by Eq. {6.4}, that is;

$$I(j,9) = \text{norm } \underline{b}_j(i/i-1) = \left[\underline{b}_j(i/i-1)^T \underline{b}_j(i/i-1)\right]^{1/2} \quad \{6.18\}$$

Since $\underline{b}_j(i/i-1)$ is a P x 1 vector obtained at the cost O(PN+1), the calculation of I(j,9) requires O(PN+P+2) operations for each candidate model term. This is the L2-norm of a vector composed of time averages between the signals specified by each of the existing model terms and the signal specified by the new candidate model term. Since this vector corresponds to the reduced version of matrix B(i/i-1) appearing in Eq. {4.30} through {4.32} and Eq. {6.4}, it was conjectured that its length might have some significance. Unfortunately, it also has a high cost and therefore offers no advantages.

The tenth search indicator is the value of the L2-norm of the vector $\underline{f}_j(i/i-1)$ given by Eq. {6.5}.

$$I(j,10) = \text{norm } \underline{f}_j(i/i-1) = \left[\underline{f}_j(i/i-1)^T \underline{f}_j(i/i-1)\right]^{1/2} \quad \{6.19\}$$

Since $\underline{f}_j(i/i-1)$ is a P x 1 vector obtained at the cost $0(P^2+PN+1)$, the calculation of $I(j,10)$ requires $0(P^2+PN+P+2)$ operations for each candidate model term. This indicator is the L2-norm of the matrix product of the preceding vector of time averages in $\underline{b}_j(i/i-1)$ and the inverse of the previous model least squares matrix $A(i-1)$. This resulting vector corresponds to the reduced version of matrix $F(i)$, appearing in Eq. {4.30}, Eq. {4.31}, and Eq. {4.37}. It was conjectured that the length of this vector might have some significance to the growth problem. Tables 5 and 6 show that is suffers from a similar high computational cost.

The eleventh search indicator is the inner product of the vectors $\underline{b}_j(i/i-1)$ and $\underline{f}_j(i/i-1)$.

$$I(j,11) = \underline{b}_j(i/i-1)^T \underline{f}_j(i/i-1) \quad \{6.20\}$$

This value appears in the calculation of matrix $G(i)$ in Eq. {4.31} and also in $I(j,5)$. Since $\underline{f}_j(i/i-1)$ is a P x 1 vector obtained at the cost $0(P^2+PN+1)$ which includes the cost of computing the P x 1 vector $\underline{b}_j(i/i-1)$, the calculation of $I(j,11)$ requires $0(P^2+PN+P+1)$ operations for each candidate model term. This second group of search indicators $I(j,9)$ through $I(j,11)$ do not appear to offer any advantages over the first group of indicators.

At this point we will leave the domain of proven results and use experimental analysis to develop other search indicators for the model growth problem. We provide

mathematical justification wherever possible, but these are the results of mathematical rationalization based on experimental findings. The following factor is the main result obtained after many detailed experiments using computer simulated systems and a controlled input sequence.

The twelfth search indicator is defined as the ratio of the square of the values of the third search indicator, divided by the fourth search indicator.

$$I(j,12) = I(j,3)^2 / I(j,4) \qquad \{6.21\}$$

This twelfth indicator was experimentally developed as a heuristic compromise to the computational and performance limitations of some of the preceding indicators. One explanation of the meaning for this search indicator is described below.

The improvement in the fitting error resulting from the involvement of just the $(j)^{th}$ candidate model term is defined as $J_j^2(i/i-1)$, and can be obtained from Eq. {4.36} by reducing this general vector equation to its simpler one-term model form. Since $\underline{g}(i)$ and $\underline{k}(i)$ become $g_j(i)$ and $k_j(i)$, respectively, we obtain;

$$J_j^2(i/i-1) = J^2(i) - J^2(i-1) = \left[\underline{g}(i)^T \underline{k}(i)\right]_j = g_j(i)\, k_j(i) \qquad \{6.22\}$$

Substituting Eq. {4.33} into {6.22} yields;

$$J_j^2(i/i-1) = g_j(i) G_j(i)^{-1} g_j(i) = [g_j(i)]^2 / G_j(i) \qquad \{6.23\}$$

Substituting Eq. {6.4} and {6.5} into {6.14} produces another expression for the reduced version of the matrix $G(i)$ to the scalar $G_j(i)$.

$$G_j(i) = \frac{1}{N} \underline{w}_j(i/i-1)^T \underline{w}_j(i/i-1)$$

$$- \frac{1}{N} \underline{w}_j(i/i-1)^T W(i-1)A(i-1)^{-1} W(i-1)^T \underline{w}_j(i/i-1)$$

$$= \frac{1}{N} \underline{w}\,(i/i-1)^T \left[I - W(i-1)A(i-1)^{-1} W(i-1)^T \right] \underline{w}\,(i/i-1)$$

$$= \frac{1}{N} \underline{w}_j(i/i-1)^T H(i-1)\, \underline{w}_j(i/i-1) \qquad \{6.24\}$$

where $H(i-1) = \left[I - W(i-1)A(i-1)^{-1} W(i-1)^T \right]$ $\{6.25\}$

The matrix H(i-1) is a function of the preceding model, and not a function of the candidate model term. Therefore it can be considered as a constant scaling factor for each candidate term evaluation at any model iteration step. Matrix H(i-1) is positive semi-definite since the scalar $G_j(i)$ cannot be negative, and H(i-1) is also idempotent. Since $G_j(i)$ is a quadratic form, we can use a quadratic identity [Ref. 18, pp. 254], and write it as:

$$G_j(i) = \text{trace} \left[\frac{1}{N} \underline{w}_j(i/i-1)\, \underline{w}_j(i/i-1)^T H(i-1) \right] \qquad \{6.26\}$$

After many attempts, we are still unable to reduce Eq. {6.26} to a form that can be more efficiently computed. Based on the properties of matrix H(i-1), and the heuristic belief that the trace of the matrix in the square brackets of Eq. {6.26} is an important factor to consider, we make the following approximation. Justification for this approximation will be given in a subsequent theorem. Using Eq. {6.24}, {6.25}, and {6.26}, approximate $G_j(i)$ by its maximum value, $G_j(i)^+$.

$$G_j(i)^+ = \max G_j(i) = \text{trace}\left[\frac{1}{N}\,\underline{w}_j(i/i-1)\,\underline{w}_j(i/i-1)^T\right] \qquad \{6.27\}$$

Analysis of Eq. {6.27} and Eq. {6.13} lead to the recognition that the trace of the matrix in square brackets of Eq. {6.27} equals search indicator I(j,4).

$$G_j(i)^+ = I(j,4) \qquad \{6.28\}$$

Substituting Eq. {6.28} into Eq. {6.23}, and using I(j,3) for $g_j(i)$, results in the new search indicator;

$$I(j,12) = \frac{I(j,3)^2}{I(j,4)} = \frac{\left[\frac{1}{N}\,\underline{w}_j(i/i-1)^T\underline{e}(i-1)\right]^2}{\frac{1}{N}\,\underline{w}_j(i/i-1)^T\underline{w}_j(i/i-1)} \qquad \{6.29\}$$

Since $\underline{w}_j(i/i-1)$ is a N x 1 vector, and $\underline{e}(i-1)$ is a N x 1 vector obtained at the cost O(PN), the calculation of I(j,12) requires O(PN+2N+4) operations for the first candidate model term. For the second and subsequent candidate model terms the cost is reduced to O(2N+4) since $\underline{e}(i-1)$ has already been calculated. Note that I(j,12) is a normalized version of the square of $g_j(i)$, and therefore should be a better indicator than I(j,3). It is also cheaper to calculate than I(j,8). These preceding order of complexity equations appear in Table 5 and Table 6 along with some numerical examples.

SEARCH INDICATOR	COMPLEXITY	N=50 P=5	N=50 P=10	N=100 P=5	N=100 P=50	N=500 P=5	N=500 P=50
1	$O(N+1)$	51	51	101	101	501	501
2	$O(P^2+NP+N+P+2)$	332	662	632	7652	3032	28052
3	$O(NP+N+1)$	301	551	601	5101	3001	25501
4	$O(N+1)$	51	51	101	101	501	501
5	$O(P^2+NP+N+P+2)$	332	662	632	7652	3032	28052
6	$O(P^2+2NP+2N+P+4)$	634	1214	1234	12754	6034	53554
7	$O(P^2+NP+2N+2P+4)$	390	725	740	7805	3540	28605
8	$O(P^2+2NP+2N+P+5)$	635	1215	1235	12755	6035	53555
9	$O(NP+P+2)$	257	512	507	5052	2507	25052
10	$O(P^2+NP+P+2)$	282	612	532	7552	2532	27552
11	$O(P^2+NP+P+1)$	281	611	531	7551	2531	27551
12	$O(NP+2N+4)$	354	604	704	5204	3504	26004

TABLE 5: Order of complexity (Number of multiplications or divisions) required to compute each search indicator value for a single candidate model term. Various examples of model size P and measurement sequence length N are included.

Some of the factors required in the calculation of the search indicator values of the first candidate model term at each growth iteration, can be used in the calculation of other search indicator values for this term and subsequent model terms. This can be exploited to produce a lower computational complexity for each candidate model term beyond the first as shown in Table 6.

SEARCH INDICATOR	COMPLEXITY	N=50 P=5	N=50 P=10	N=100 P=5	N=100 P=50	N=500 P=5	N=500 P=50
1	$O(N+1)$	51	51	101	101	501	501
2	$O(P^2+NP+N+P+2)$	332	662	632	7652	3032	28052
3	$O(N+1)$	51	51	101	101	501	501
4	$O(N+1)$	51	51	101	101	501	501
5	$O(P^2+NP+N+P+2)$	332	662	632	7652	3032	28052
6	$O(P^2+NP+2N+P+4)$	384	714	734	7754	3534	28554
7	$O(P^2+NP+2N+2P+4)$	390	725	740	7805	3540	28605
8	$O(P^2+NP+2N+P+5)$	385	715	735	7755	3535	28555
9	$O(NP+P+2)$	257	512	507	5052	2507	25052
10	$O(P^2+NP+P+2)$	282	612	532	7552	2532	27552
11	$O(P^2+NP+P+1)$	281	611	531	7551	2531	27551
12	$O(2N+4)$	104	104	204	204	1004	1004

TABLE 6: Order of complexity (Number of multiplications or divisions) required to compute each search indicator value for subsequent model terms beyond the first. Various examples of model size P and measurement sequence length N are included.

Based on the preceding development of I(j,12), we state and prove the following theorem.

THEOREM 2: LOWER BOUND ON REDUCTION IN FITTING ERROR

I(j,12) is a lower bound on the improvement in the fitting error resulting from including the single model term $w_j(n,i/i-1)$ at the $(i)^{th}$ growth iteration.

PROOF:

From Eq. {6.12}, $I(j,3) = g_j(i)$. Substituting this into Eq. {6.23} yields;

$$J_j^2(i/i\text{-}1) = I(j,3)^2 / G_j(i) \qquad \{6.30\}$$

From the development of Eq. {6.24} and Eq. {6.27}, we see that $G_j(i)$ can be written as;

$$G_j(i) = G_j(i)^+ - P_j = I(j,4) - P_j \qquad \{6.31\}$$

where P_j is nonnegative and $P_j < I(j,4)$. Therefore $I(j,4)$ is an upper bound on $G_j(i)$. Applying this last result to Eq. {6.30} and Eq. {6.29} yields the result that $I(j,12)$ is a lower bound on $J_j^2(i/i\text{-}1)$.

We have shown how the value of $I(j,12)$ is related to the improvement in the error fitting criterion. Tables 5 and 6 show that this search indicator can be computed with a very low computational cost. In fact, the cost in Table 6 is not a function of the size P of the existing model, only of the number N of data measurements.

The power of this new search indicator is significant. The computer simulated and real world experiments we have performed indicate that it is an excellent indicator of the fitting error improvement that results from including the candidate model term. Because the value of $I(j,12)$ is proportional to the square of $I(j,3)$, it rarely happens that a term with low $I(j,12)$ will have a significantly large value of $I(j,8)$, the actual fitting error improvement. The

fact that I(j,12) is easily computed adds to its significance.

For starting the model growth, we can select the subset of terms in $\underline{w}(n,1)$ with the one or two largest values of I(j,12). At this first iteration there is no error residual signal since there is no existing model, so we use the total model output sequence $\{y(n)\}$ in place of $\{e(n,0)\}$. While we have not been able to prove that this manner of specifying this subset of $\underline{w}(n,1)$ prevents inclusion of unneeded terms, results of many experiments show this method provides a good set of starting terms and generally yields more compact models.

We have examined the characteristics of search indicators I(j,1) through I(j,12) under experimental conditions. This involved numerous experiments with synthesized systems, a controlled input probe sequence, and the assumption of no additive output noise. A subsequent section examines the robustness of model growth in cases where preceding assumptions are relaxed.

A thirteenth search indicator is designed for a special purpose. We previously discussed the potential problem of nearly equivalent performance from different model terms. This leads to ill-conditioning of the least squares matrix and the possibility of multiple solutions.

The thirteenth search indicator is the maximum result chosen from the set of squared and normalized time averages

obtained from the product of the signal specified by the candidate model term, and the signals specified from each of the other q(i) candidate model terms at this iteration.

$$I(j,13) = \underset{\substack{1<=k<=q(i) \\ j\neq k}}{\text{maximum}} \left\{ \frac{\left[\frac{1}{N} \sum_{n=n_2}^{n_3} w_j(n,i/i-1)\, w_k(n,i/i-1) \right]^2}{\frac{1}{N} \sum_{n=n_2}^{n_3} w_j(n,i/i-1)^2 \; \frac{1}{N} \sum_{n=n_2}^{n_3} w_k(n,i/i-1)^2} \right\} \quad \{6.32\}$$

Examination of Eq. {4.27} and Eq. {6.32} shows that the value of I(j,13) equals the maximum ratio of the square of each off-diagonal element of the $(j)^{th}$ row of A(i/i-1), and the product of the diagonal elements of the corresponding column and the $(j)^{th}$ row. In physical terms, large I(j,13) means there is the possibility of significant correlation between the signal specified by the $(j)^{th}$ candidate model term and the signal specified by another candidate model term. This is related to the multiple correlation coefficient in regression analysis. The following set of theorems show that a necessary condition for the least squares solution to represent a unique minimum is that A(i/i-1) be positive definite at each growth iteration. They also shown how this condition is related to the values of I(j,13). Theorem 3 is well known in the linear algebra and matrix literature, and is included for completeness.

THEOREM 3:

The elements of a positive definite matrix $D = [d_j]$ must satisfy the inequality;

$$d_{jk}^2 < d_{jj}\, d_{kk} \qquad \text{for all } j \neq k \qquad \{6.33\}$$

PROOF:

Assume that the converse of {6.33} existed for some d_{jk}. We could interchange the $(1)^{st}$ column of D with the $(j)^{th}$ column, the $(2)^{nd}$ column with the $(k)^{th}$ column, the $(1)^{st}$ row with the $(j)^{th}$ row, and the $(2)^{nd}$ row with the $(k)^{th}$ row without affecting the definiteness of D. From the converse of {6.33}, the determinant of this 2 x 2 principle submatrix of D would now be less than or equal to zero, and D could not be positive definite. Therefore {6.33} is a necessary condition.

THEOREM 4:

A necessary condition for the matrix A(i/i-1) given by Eq. {4.27} to be positive definite, is that the value of I(j,13) for each of the q(i) terms in the $(i)^{th}$ iteration must satisfy the following inequality.

$$I(j,13) < 1 \qquad \text{for all } j,\ j=1,2,\ldots,q(i) \qquad \{6.34\}$$

PROOF:

From Eq. {4.27}, all diagonal elements of A(i/i-1) are nonnegative. From Eq. {6.32} and Theorem 3 we see that {6.34} is equivalent to {6.33}, and therefore {6.34} is a necessary condition for A(i/i-1) to be positive definite.

THEOREM 5:

A necessary condition for the uniqueness of the solution of the normal equations {4.24}, and repeated below;

$$A(i)\underline{\theta}(i) = \left[\begin{array}{c|c} A(i-1) & B(i/i-1) \\ \hline B(i/i-1)^T & A(i/i-1) \end{array}\right] \underline{\theta}(i) = \underline{h}(i) \qquad \{6.35\}$$

is that the value of I(j,13) for each of the q(i) terms in the $(i)^{th}$ iteration must satisfy the following inequality.

$$I(j,13) < 1 \quad \text{for all } j,\ j=1,2,\ldots,q(i) \qquad \{6.36\}$$

PROOF

From Chapter III, Eq. {3.18} describes the condition that the least squares matrix A(i) must be positive definite for $J^2(i)$ to represent a unique minimum. If A(i) is not positive definite, the system of equations given by Eq, {6.35} contains more than one set of solutions that equivalently minimize the fitting error criterion. The least squares error minimization may become extremely unstable since the minimum will tend to lie on a line or surface in parameter space, rather than at a point. From Theorem 4, Eq. {6.34} is a necessary condition that A(i/i-1) is positive definite. It follows directly from the proof of Theorem 3, that a necessary condition for A(i) to be positive definite is that A(i/i-1) is also positive definite. Therefore we see that {6.34}, or equivalently {6.36} is a necessary condition.

Other possible search indicators are contained in the area of the patterns in the residual sequence of the $(i-1)^{st}$ model. We know that when we have completely modeled a system using noise free measurements, the error residual

should be a random sequence without any trends or patterns. In fact, we expect that this case will produce an error residual sequence composed of a series of very short segments of alternating sign. It is reasonable to expect that the pattern of segments we see in the residual when we have undermodeled the system, is representative of the missing term(s) in the model. The problem is to learn how to decode this information from the patterns in the $(i-1)^{st}$ model, to aid us in selecting the missing term or terms.

C. SEARCH INDICATOR GROWTH ALGORITHM

Our proposed Search Indicator Growth Algorithm is represented in Figure 23. We start by specifying a very large set of candidate model terms. Our algorithm picks the subset of candidate model terms whose I(j,12) values are greater than some specified value of the variable h_1 (e.g. 70% of the maximum value of I(j,12) for any candidate term). Before adding the selected term(s) to the model for subsequent evaluation of the fitting error, we calculate the value of I(j,13) for each selected term using Eq. {6.32}. A second heuristic variable h_2 is used to indicate when significant colinearity is present. Values of I(j,13) close to 1 indicate that the $(j)^{th}$ candidate model term (out of the selected set) is nearly linearly dependent on another candidate model term. This other term is used in the calculation for the $(k)^{th}$ row of A(i/i-1), and contributes to the large I(j,13). When I(j,13) is greater than h_2, we

discard the candidate model term of this pair that has the lower value of I(j,12), re-estimate I(j,13) for the remaining term, and continue until all values of I(j,13) are sufficiently small (e.g. less than 0.85).

This iterative two-phase growth technique is based on the terms selected by the search indicators, and has a much lower computational cost than the complete evaluation of $J^2(i)$ for each combination of possible new terms.

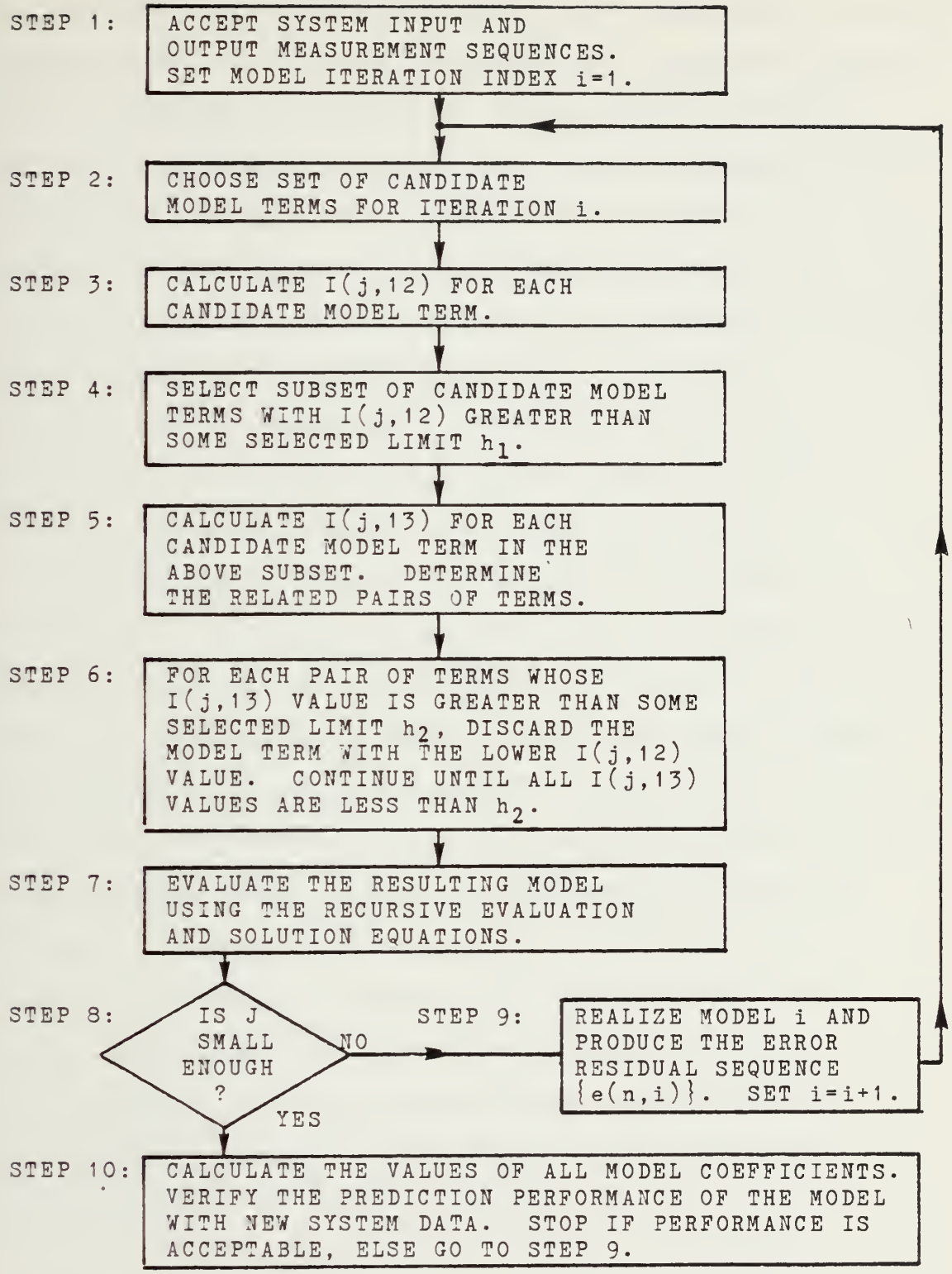


FIGURE 23: Flow Diagram of the Search Indicator Growth Algorithm

The heuristic variable h_1 determines the number of terms selected for inclusion in the model, and can be set based on the distribution of the values of the search indicator I(j,12). If there is a grouping of terms with high values for I(j,12), they should probably all be accepted into the model. If there are only a few terms with high values of I(j,12), we should select them all, plus possibly a few more with slightly lower values of I(j,12). There is a disadvantage of selecting h_1 too small, since this can result in the requirement for extra iterations in order to obtain all of the needed terms in the final model.

The heuristic variable h_2 determines the amount of colinearity allowed between model terms. If chosen too low, it will delay or prevent acceptance of actually needed model terms that happen to be somewhat correlated with existing model terms. If chosen too high, it allows extra terms into the model and thereby increase the ill-conditioning of the least squares matrix. We have experimentally found the range $0.7 <= h_2 <= 0.85$ to be most effective.

The next section examines the computational cost of model growth using the techniques discussed to this point. The result is that model growth using the search indicator techniques developed in this chapter offers a new and efficient means of obtaining models.

D. COMPUTATIONAL COMPARISON OF GROWTH TECHNIQUES

This section examines the algorithms and computational cost associated with the model growth techniques presented in this thesis. We use N to denote the number of data points, c(i) for the number of model terms in iteration i, and q(i) for the number of candidate new terms at this iteration. Therefore q(i) = c(i)-c(i-1). Each growth technique is presented in algorithmic form as a series of steps, and the number of multiplicative or division computations required at each step is indicated. The details of these order of complexity calculations are based on the size of the various matrices, vectors, and sequences used in the model growth, and are included in Appendix C for the interested reader. The computational cost equation is formed for a full iteration at the end of each technique, and an example is used to emphasize the differences in the growth techniques.

Technique 1: Direct Least Squares		Computational Cost
Step 1:	Set i = 1, form term vector $\underline{x}(n,i)$	
Step 2:	Form R(i) using Eq. {4.5}	[N+1]c(i)[c(i)+1]/2
Step 3:	Form $\underline{r}(i)$ using Eq. {4.6}	c(i)[N+1]
Step 4:	Invert R(i)	[c(i)**3]/6
Step 5:	Solve for $J^2(i)$ using Eq. {4.3}	[c(i)**2]+c(i)+N+1
Step 6:	If $J^2(i)$ < acceptable level, stop. Else;	
Step 7:	Set i = i+1, form a new term vector $\underline{w}(n,i)$. Go to Step 2	

Total cost for Steps 1 through 7 is;

$$O(n) = [c(i)**3]/6 + [[N+3]/2][c(i)**2] + [[3N+5]/2]c(i) + N+1$$

Example: N = 500, c(1) = 10, c(2) = 20, c(3) = 30

Iteration	Number of multiplicative operations
1	33343
2	117485
3	253926
	TOTAL = 404754

Technique 2: Block Form Recursive Growth		Computational Cost
Step 1:	Set i = 1, form term vector $\underline{x}(n,i)$	
Step 2:	Form R(i) using Eq. {4.5}	[N+1]c(i)[c(i)+1]/2
Step 3:	Form $\underline{r}(i)$ using Eq. {4.6}	c(i)[N+1]
Step 4:	Invert R(i)	[c(i)**3]/6
Step 5:	Solve for $J^2(i)$ using Eq. {4.3}	[c(i)**2]+c(i)+N+1
Step 6:	If $J^2(i)$ < acceptable level, stop. Else;	
Step 7:	Set i = i+1, form a new term vector $\underline{w}(n,i/i-1)$	
Step 8:	Form A(i/i-1) using Eq. {4.27}	[N+1]q(i)[q(i)+1]/2
Step 9:	Form B(i/i-1) using Eq, {4.26}	q(i)P[N+1]
Step 10:	Form $\underline{h}(i/i-1)$ using Eq. {4.29}	q(i)[N+1]
Step 11:	Form F(i) using Eq, {4.30}	q(i)[P**2]
Step 12:	Form G(i) using Eq. {4.31}	P[q(i)**2]
Step 13:	Form $\underline{g}(i)$ using Eq. {4.32}	Pq(i)
Step 14:	Invert G(i)	[q(i)**3]/6
Step 15:	Form $\underline{k}(i)$ using Eq. {4.33}	q(i)**2
Step 16:	Solve for $J^2(i)$ using Eq. {4.36}	q(i)
Step 17:	If $J^2(i)$ < acceptable level, stop. Else;	
Step 18:	Form inverse of A(i) using Eq. {4.37}	q(i)[P**2] + P[q(i)**2]
Step 19:	Go to Step 7	

Cost for Steps 1 through 7 is the same as Technique 1.

Total cost for Steps 8 through 17 are;

O(n) = [q(i)**3]/6 + [P+N+3][q(i)**2]/2 + q(i)[NP+[P**2]+2P+[3N+5/2]

Cost for Step 18 is; O(n) = q(i)[P**2] + P[q(i)**2]

Example: N = 500, c(1) = 10, c(2) = 20, c(3) = 30

Iteration	Steps	Number of multiplicative operations	
1	1 - 7		33343
2	8 - 18	84542+2000(for step 18) =	86542
3	8 - 17		138240
		TOTAL =	258127

Technique 3: Search Indicator Growth Algorithm		Computational Cost
Step 1:	Set i = 1, form term vector $\underline{x}(n,i)$	
Step 2:	Form R(i) using Eq. {4.5}	[N+1]c(i)[c(i)+1]/2
Step 3:	Form $\underline{r}(i)$ using Eq. {4.6}	c(i)[N+1]
Step 4:	Invert R(i)	[c(i)**3]/6
Step 5:	Solve for $J^2(i)$ using Eq. {4.3}	[c(i)**2]+c(i)+N+1
Step 6:	If $J^2(i)$ < acceptable level, stop. Else;	
Step 7:	Set i = i+1, form a new term vector $\underline{w}(n,i/i-1)$	
Step 8:	Form I(j,12) for each term in $\underline{w}(n,i/i-1)$ using Eq. {6.29}	NP + [2N+4]q(i)
Step 9:	Select the subset of k terms with values of I(j,12) greater than a specified level h_1. Reduce the vector $\underline{w}(n,i/i-1)$ to only contain this subset of k terms	No cost
Step 10:	Form A(i/i-1) using the reduced vector $\underline{w}(n,i/i-1)$ in Eq. {4.27}	k[k+1][N+1]/2
Step 11:	Form B(i/i-1) using the reduced vector $\underline{w}(n,i/i-1)$ in Eq. {4.26}	kP[N+1]
Step 12:	Form $\underline{h}(i/i-1)$ using the reduced vector $\underline{w}(n,i/i-1)$ in Eq. {4.29}	k[N+1]
Step 13:	Form F(i) using the reduced vector $\underline{w}(n,i/i-1)$ in Eq. {4.30}	k[P**2]
Step 14:	Form G(i) using the reduced vector $\underline{w}(n,i/i-1)$ in Eq. {4.31}	P[k**2]

Step 15:	Form $\underline{g}(i)$ using the reduced vector $\underline{w}(n,i/i-1)$ in Eq. {4.32}	Pk
Step 16:	Invert G(i)	[k**3]/6
Step 17:	Form $\underline{k}(i)$ using Eq. {4.33}	[k**2]
Step 18:	Solve for $J^2(i)$ using Eq. {4.36}	k
Step 19:	If $J^2(i)$ < acceptable level, stop. Else,	
Step 20:	Form inverse of A(i) using Eq. {4.37}	[P**2]k + P[k**2]
Step 21:	Go to Step 7	

Cost for Steps 1 through 7 is the same as Technique 1 and Technique 2. Total cost for Steps 8 through 19 is;

O(n) = [k**3]/6 + [k**2][P+[N+3]/2] + k[2P+PN+[P**2]+[3N+5]/2]
+ NP +[2N+4]q(i)

Cost for Steps 20 through 21 is;

O(n) = [P**2]k + P[k**2]

Example: N = 500, c(1) = 10, c(2) = 20, c(3) = 30, Let k = 3

Iteration	Steps	Number of multiplicative operations	
1	1 - 7		33343
2	8 - 21	35017+390(for step 20)	35407
3	8 - 19		49305
		TOTAL =	118055

Note: Additional savings can be realized when I(j,13) is used to eliminate highly colinear terms in Step 9.

The preceding example shows that the Search Indicator Growth Algorithm can require substantially lower computational cost than the other two techniques. Table 7 summarized the results of the example. Because of this lower cost, we can consider a greater number of candidate terms during each iteration than would be possible with the direct or block-form techniques. This increases the probability that we will consider the terms actually needed in the model. The performance of this algorithm will be demonstrated in the experiments of Chapter VII.

Technique	Cost of Iteration 1	Cost of Iteration 2	Cost of Iteration 3	Total Cost
1 Direct Least Squares	33343	117485	253926	404754
2 Block form Recursive	33343	86542	138240	258127
3 Search Indicator Growth	33343	35407	49305	118055

TABLE 7: Computation Cost (Number of Multiplications or Divisions) Required in example of Section D, Chapter VI.

E. FACTORS AFFECTING MODEL EVALUATION AND GROWTH

Chapters I and II mentioned that there were two main factors that can limit the ability to accurately model a system from input and output measurements. These are; (1) the ability to control the input signal applied to the system, and (2) the presence of output measurement noise. The four permutations of these two factors are represented in the following table.

	SYSTEM INPUT		OUTPUT MEASUREMENTS	
Case	Uncontrollable	Controllable	No Noise	Additive Noise
1A	X		X	
1B	X			X
2A		X	X	
2B		X		X

TABLE 8: System Characterization Conditions

Other factors include the form of the system and the model (e.g. other than BVM), choice of error minimization method, and selection of sampling interval (over or under sampling is a possibility). It is assumed that these last two factors are not a problem in the examples we consider.

We have been primarily concerned with Case 2A in this thesis because it allows us to focus on just the choice of model terms. In the computer simulated experiments, we generate an input probe using a uniformly distributed pseudo-random number generator. The amplitude values of this sequence are scaled to cover the known (or assumed) operating range of the nominal system input. We then apply this input sequence to the system, and use the resulting output sequence along with the input probe sequence to grow the model by any of the techniques discussed in the preceding chapters.

A uniform distribution was chosen for the input probe rather than the gaussian distribution typically mentioned in

the literature, based on the following argument. Nonlinear terms contribute to the output sequence in a nonlinear amplitude dependent manner. Since we don't know the form of the nonlinear system terms, select an input probe that is equally likely to take on any value in the allowed range. One could, of course, postulate system examples where a nonuniform input probe amplitude distribution provides more efficient model growth (e.g. more significant differences in the behavior of the candidate model terms).

Case 2B has additive output noise contaminating the system output sequence $\{y(n)\}$. This is the next step towards the situation we must face in the real world. If it is reasonable to consider this additive noise to be zero mean, stationary, and uncorrelated with the system input, then we can perform some filtering to reduce the distortion examined in Chapter III. Since we still control the input sequence, we can measure and record the noisy output sequence for M repeated applications of the identical input sequence. A point-for-point ensemble average of the noisy system output can then be performed, which reduces the variance of $\{y(n)\}$ by the factor M. This filters the output variation due to the additive noise, and we can grow a model using the input sequence, and the output sequence corresponding to the average of the noisy output sequences. This technique has been tried experimentally and produces improved results. An example is presented in Chapter VII.

Case 1B has no additive output noise but we must work with the given input sequence (e.g., we cannot probe the system ourselves). If the given input sequence is sufficiently wideband or "persistently exciting" [Ref. 14, pp 42], then the least squares matrix A(i) will be well conditioned at each growth iteration i, and the growth techniques provide useful results. Each specific case of input signal, system output, and model form must be examined experimentally to determine if the evaluation equations are well conditioned. Examination of the amplitude distribution and the empirical sample autocorrelation of the particular input sequence gives a qualitative measure of the suitability of the available input for systems characterization. Much work needs to be done in rating a given input signal for use in systems characterization, and this is suggested as an area for future research. Ultimately, it is the value of the obtained final model in the intended application that determines the adequacy of the input signal used in the characterization.

Case 1A is the most difficult set of conditions for any model growth technique. Even if we knew the exact form of the final model, and were therefore just doing parameter estimation, the output noise would degrade the model growth and evaluation error. We may not obtain a useful system characterization under these conditions.

Case 1A is the situation normally encountered when we don't have control of the experiment that obtains the measurement data. Two possible techniques to try are as follows. Using the given input and output measurement sequences, we could use the search indicator growth algorithm until we reach a limiting number of model terms (e.g. N/10), or until there was no significant improvement in the fitting error. At this point we "freeze" the current model and simulate it on the computer. By probing this mathematical model with the given input sequence $\{u(n);\ S<=n<=T\}$, we can produce the model output sequence $\{\hat{y}(n)\}$. Using a nonlinear iterative algorithm such as Marquardt [Ref. 17], we could perform an iterative nonlinear analysis in an attempt to refine the parameter estimates and reduce the magnitude of the output error $\hat{e}(n) = y(n) - \hat{y}(n)$. Using this corrected model, and the least squares matrix and vector corresponding to it, we could then grow from this point using the Search Indicator Growth Algorithm. This two-phase process could continue until no significant decrease in J is obtained.

Another proposed technique would be to grow a nonrecursive model like the VOL(d,m) from the input and noisy output measurements, using the Search Indicator Growth Algorithm. The noisy output data would not distort the coefficients of a nonrecursive model, and it might be possible to obtain a reasonable fit. Since there is noise

added to the system output, a stopping criterion such as independence of the residual sequence {e(n,i)}, as measured by its autocorrelation sequence, would make more sense than the magnitude of the fitting error $J^2(i)$. When a nonrecursive model with a limiting number of terms (e.g. N/10) is obtained, or {e(n)} is found to be uncorrelated, then a second phase would be used. The previously determined nonrecursive model would be used along with the input signal to produce the model output $\{\bar{y}(n)\}$. The input signal {u(n)} and the nonrecursive model output $\{\bar{y}(n)\}$ would then be used to grow a more general and probably more compact recursive model like the BVM and using the Search Indicator Growth Algorithm. This concept could be expected to reduce the effect of the additive output noise. We denote this as the "N-R" technique because it uses both nonrecursive and recursive models.

This concept is related to a recently developed two-stage least squares parameter estimation algorithm for linear systems [Ref. 42]. The method presented here is more powerful since it is applicable to model growth for nonlinear systems and uses the efficient search indicator growth algorithm developed in the previous section. Experimental analysis of the method discussed in this section is provided in Chapter VII.

VII. EXPERIMENTS IN SYSTEM CHARACTERIZATION

A. DISCUSSION

The preceding chapter developed the Search Indicator Growth Algorithm and showed the computational advantages that result from its use. The next step is the experimental evaluation of the performance of this proposed algorithm in characterizing systems. These evaluations include comparisons with the performance of the block-form techniques developed in Chapter V.

This chapter contains several experiments designed to demonstrate the strengths and limitations of the model growth techniques presented in this thesis. In the first six experiments we synthesize a given system equation on a computer, and generate a finite length pseudo-random input sequence {u(n)} uniformly distributed between chosen amplitude limits. Each case involves probing the system equation with the input sequence to create an output sequence {y(n)}. These input and output sequences are then used as data points for the model growth techniques. Various system features and measurement noise conditions are included for illustrative purposes.

The advantage of using synthesized systems is that it allows us to examine the properties of the model growth techniques under conditions that do not obscure the key

differences. We can more clearly see the weaknesses of some techniques and verify how other techniques can compensate for related problems. The Covariance error minimization method is used for each growth technique because of its superior performance (Chapter III).

The third section of this chapter examines the capabilities of our best growth techniques on a real world example, where we must work with the single set of available measurement sequences (Case 1A in Chapter VI). Verification of the modeling results is not as direct in this case since the actual system equation is unknown. This real example verifies some of the inherent weaknesses of model growth techniques when we are faced with Case 1A conditions. The final section summarizes the experimental findings.

B. CONTROLLED EXPERIMENTS

The systems used in these experiments were not selected to bias the findings in favor of any technique. We have not excluded any examples or experiments that produced contrary results. The following set of experiments are honestly considered to fairly examine the basic properties of the various model growth techniques. We start these experiments with Experiment 2, since Experiment 1 is contained in Chapter III.

Experiment 2

The purpose of this experiment is to demonstrate that the restricted growth properties of the block-form model

growth techniques generally lead to a high condition number for the least squares matrix A(i). This situation can extend to the extreme point of ill-conditioning where these growth techniques fail to converge on an adequate model. The Search Indicator Growth Algorithm allows unrestricted model growth, is robust to ill-conditioning, and typically finds an acceptable model when block-form techniques fail.

We synthesize the following nonlinear system.

$$y(n) = 1.0\,u(n) + .8\,u(n-1) + .6\,u(n-2) - .9\,y(n-1) - .7\,y(n-2) + .4\,u(n)u(n) - .2\,u(n-1)u(n-1)y(n-3) - .1\,y(n-1)y(n-2)y(n-3) - .12\,u(n)y(n-3)y(n-3) \quad \{7.1\}$$

A random input probe $\{u(n); 1<=n<=200\}$ is generated uniformly distributed between the amplitude limits of -2 and +2. The system output sequence $\{y(n)\}$ is produced by probing the system of Eq. {7.1} with the input sequence $\{u(n)\}$. Starting with evaluation of the base model BVM(1,1), we recursively grow models by each of the six block-form growth techniques[9] of Chapter V and the Search Indicator Growth Algorithm of Chapter VI. The condition number and error fit for each model are evaluated at each iteration, and the results presented in Table 9. We also include the results

---

9 Both the "M Directed" and "D Directed" growth algorithms require a significance test for switching between their two phases (See Figure 21 and the discussion in Chapter V). For clarity of presentation, we assume that there is a test that recognizes the place to change phases after going one increment too far (e.g. we turn after m=3 or d=3, respectively). The tables for each of the following experiments show where these phase changes are made.

of a direct least squares model evaluation using the exact form of the system as a comparison basis. Table 10 contains additional details of the more successful characterization by the Search Indicator Growth Algorithm.

METHOD NAME	ITERATION i	MODEL	NEW TERMS	SELECTED TERMS	TOTAL TERMS	CONDITION NUMBER	SQUARE ROOT OF FITTING ERROR; J
M DIRECTED	1	BVM(1,1)	3	3	3	.1047E+02	.5183E+00
	2	BVM(1,2)	2	2	5	.1769E+02	.4737E+00
	3	BVM(1,3)	2	2	7	.3330E+02	.4532E+00
	4	BVM(1,4)	2	2	9	.4177E+02	.4114E+00
(see footnote 9)	4*	BVM(2,3)	28	28	35	.1178E+05	.1540E+00
	5	BVM(3,3)	84	84	119	.1349E+08	.1668E-02
D DIRECTED	1	BVM(1,1)	3	3	3	.1047E+02	.5183E+00
	2	BVM(2,1)	6	6	9	.1386E+03	.3421E+00
	3	BVM(3,1)	10	10	19	.4105E+04	.3280E+00
	4	BVM(4,1)	21	21	40	.4075E+06	.2959E+00
(see footnote 9)	4*	BVM(3,2)	36	36	55	.6731E+05	.1832E+00
	5	BVM(3,3)	64	64	119	.1334E+08	.2632E-02
DIAGONAL	1	BVM(1,1)	3	3	3	.1047E+02	.5183E+00
	2	BVM(2,2)	17	17	20	.9233E+03	.2375E+00
	3	BVM(3,3)	99	99	119	.1265E+08	Diverging
M-D ZIG-ZAG	1	BVM(1,1)	3	3	3	.1047E+02	.5183E+00
	2	BVM(1,2)	2	2	5	.1769E+02	.4737E+00
	3	BVM(2,2)	15	15	20	.9233E+03	.2375E+00
	4	BVM(2,3)	15	15	35	.1178E+05	.1540E+00
	5	BVM(3,3)	84	84	119	.1349E+08	.1668E-02
D-M ZIG-ZAG	1	BVM(1,1)	3	3	3	.1047E+02	.5183E+00
	2	BVM(2,1)	6	6	9	.1386E+03	.3421E+00
	3	BVM(2,2)	11	11	20	.9233E+03	.2375E+00
	4	BVM(3,2)	35	35	55	.6731E+05	.1832E+00
	5	BVM(3,3)	64	64	119	.1334E+08	.2632E-02
NEIGHBOR	1	BVM(1,1)	3	3	3	.1047E+02	.5183E+00
	2A	BVM(2,1)	6	6	9	.1386E+03	.3421E+00
	2B	BVM(1,2)	3	3	6	.1769E+02	.4737E+00
	2C	BVM(2,2)*	17	17	20	.9233E+03	.2375E+00
	3A	BVM(3,2)	35	35	55	.6744E+05	.1832E+00
	3B	BVM(2,3)	15	15	35	.1178E+05	.1540E+00
	3C	BVM(3,3)	99	99	119	.1334E+08	Diverging
SEARCH INDICATOR	1	BVM(1,1)	3	3	3	.1047E+02	.5183E+00
	2	BVM(3,3)	116	1	4	.1125E+02	.3941E+00
	3	BVM(3,3)	115	7	11	.1311E+03	.2346E+00
	4	BVM(3,3)	108	4	15	.3711E+03	.8208E-01
	5	BVM(3,3)	104	2	17	.6262E+03	.1037E-05
EXACT MODEL OF THE SYSTEM			9	9	9	.3134E+02	.4176E-06

TABLE 9: Summary Results from Experiment 2

Experiment: 2 Iteration: 1 Candidate Model: BVM(1,1)

Number of candidate model terms, q(i) = 3

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)	.6583E+00	-	-
2*	y(n-1)	.3040E-01	-	-
3*	u(n-1)	.2155E-02	-	-

Number of terms in final subset (marked with *), N_f = 3

Total number of terms in resulting model, c(i) = 3

Condition Number of least squares matrix A(i), N_c = .1047E+02

Square root of the fitting error, J(i) = .5183E+00

Remarks: We chose to select all the candidate model terms.

Experiment: 2 Iteration: 2 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 116

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)u(n)	.1073E+00	-	-

Number of terms in final subset (marked with *), N_f = 1

Total number of terms in resulting model, c(i) = 4

Condition Number of least squares matrix A(i), N_c = .1125E+02

Square root of the fitting error, J(i) = .3941E+00

Remarks: This term had an I(j,12) more than twice as large as all other terms, so only this term was selected.

TABLE 10: Search Indicator Growth Algorithm results of Experiment 2 (continued on next page).

Experiment: 2 Iteration: 3 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 115

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	y(n-2)y(n-2)y(n-2)	.2945E-01	.9583E+00	u(n-2)y(n-2)y(n-2)
2*	y(n-2)	.2701E-01	.7454E+00	u(n-2)y(n-2)y(n-2)
3	u(n-2)y(n-2)y(n-2)	.2410E-01	.9583E+00	y(n-2)y(n-2)y(n-2)
4*	u(n-1)u(n-1)u(n-3)	.2237E-01	.1137E-01	u(n)u(n)y(n-2)
5*	y(n-1)y(n-2)y(n-3)	.1975E-01	.8015E+00	u(n-2)y(n-1)y(n-3)
6*	u(n-2)y(n-1)y(n-3)	.1956E-01	.8015E+00	y(n-1)y(n-2)y(n-3)
7*	u(n)u(n)y(n-2)	.1895E-01	.5954E+00	y(n-2)
8*	u(n-1)u(n-1)	.1822E-01	.6420E-01	u(n-2)y(n-2)y(n-2)

Number of terms in final subset (marked with *), N_f = 7

Total number of terms in resulting model, c(i) = 11

Condition Number of least squares matrix A(i), N_c = .1311E+03

Square root of the fitting error, J(i) = .2346E+00

Remarks: The first phase picked terms with I(j,12) > .17E+00
and the second phase kept terms with I(j,13) < 0.90

TABLE 10: (continued)

Experiment: 2 Iteration: 4 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 108

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)y(n-3)y(n-3)	.8122E-02	.9072E+00	u(n)u(n-3)y(n-3)
2*	u(n-2)u(n-2)u(n-2)	.6511E-02	.8535E+00	u(n-2)
3*	u(n-2)	.6317E-02	.8535E+00	u(n-2)u(n-2)u(n-2)
4	u(n)u(n-3)y(n-2)	.5187E-02	.9072E+00	u(n)y(n-3)y(n-3)
5*	u(n-1)u(n-1)u(n-2)	.5140E-02	.6022E+00	u(n)

Number of terms in final subset (marked with *), N_f = 4

Total number of terms in resulting model, c(i) = 15

Condition Number of least squares matrix A(i), N_c = .3709E+03

Square root of the fitting error, J(i) = .8208E-01

Remarks: The first phase picked terms with I(j,12) > 0.50E-02
and the second phase kept terms with I(j,13) < 0.90

TABLE 10: (continued)

Experiment: 2 Iteration: 5 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 108

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n-1)u(n-1)y(n-3)	.9677E-03	.6222E+00	u(n-1)y(n-1)y(n-3)
2*	u(n-1)y(n-1)y(n-3)	.5825E-03	.6222E+00	u(n-1)u(n-1)y(n-3)

Number of terms in final subset (marked with *), N_f = 2

Total number of terms in resulting model, c(i) = 17

Condition Number of least squares matrix A(i), N_c = .6262E+03

Square root of the fitting error, J(i) = .1037E-05

Remarks: The first phase picked terms with I(j,12) > 0.50E-03
and the second phase kept terms with I(j,13) < 0.90

TABLE 10: (continued)

Table 9 indicates that two of the block-form techniques produced excessively ill-conditioned least squares matrices and were unable to be solved. Matrix G(i), given by Eq. {4.31}, became singular and this stopped the evaluation. The other four block-form techniques had very high condition numbers but were able to converge on the BVM(3,3) which subsumes the system of Eq. {7.1}. Each case produced a considerable number of unnecessary model terms and many had significant coefficient values (as large as .10E-01). It would be difficult to identify these terms as unnecessary without having knowledge of the system equation. Both the square root of the fitting error and the condition number of

the least squares matrices corresponding to the models from each of these block form techniques are much larger than the exact model case. This leads us to question the value of the resulting models.

The search indicator technique did not suffer from these problems, and settled on the model equation described below.

$$
\begin{aligned}
y(n) = {} & 1.0000E{+}0\ u(n) + 0.7999E{+}0\ u(n-1) + 0.6000E{+}0\ u(n-2) \\
& - 0.8999E{+}0\ y(n-1) - 0.7000E{+}0\ y(n-2) + 0.4000E{+}0\ u(n)u(n) \\
& - 0.2000E{+}0\ u(n-1)u(n-1)y(n-3) - 0.1000E{+}0\ y(n-1)y(n-2)y(n-3) \\
& - 0.1200E{+}0\ u(n)y(n-3)y(n-3) - 0.2384E{-}6\ u(n-1)u(n-1) \\
& + 0.1490E{-}6\ u(n-1)u(n-1)u(n-1) - 0.2962E{-}6\ y(n-2)y(n-2)y(n-2) \\
& - 0.6985E{-}7\ u(n)u(n)y(n-2) - 0.5765E{-}6\ u(n-2)y(n-1)y(n-3) \\
& - 0.3073E{-}6\ u(n-1)u(n-1)u(n-2) - 0.1612E{-}5\ u(n-2)u(n-2)u(n-2) \\
& - 0.1219E{-}6\ u(n-1)y(n-1)y(n-3) \qquad \{7.2\}
\end{aligned}
$$

It is obvious that we can ignore the terms beyond the ninth term in Eq. {7.2}. Table 9 shows that the square root of the fitting error from the Search Indicator Growth Algorithm was better than three orders of magnitude lower than any error obtained by the block form techniques. The condition number and fitting error produced by this algorithm are realistically close to the values produced by direct analysis of the exact model. The square root of the fitting error for the exact model was not exactly zero, which indicates that some numerical roundoff error existed in the computer program. We actually computed $J^2(i)$ and then took the square root. The non-zero value for the square root of

the exact model fitting error J(i) = .4176E-6 translates to a $J^2(i)$ of .1744E-12, which is within the expected numerical range of zero for the computer.

Table 10 shows the operation of the Search Indicator Growth Algorithm. Notice how rapidly this technique selected the critical model terms. The line titled "Remarks" gives a summary of the heuristic decision making rules used for acceptance of the particular candidate model terms in each phase of the algorithm.

This experiment demonstrated the weakness of the block form techniques resulting from their restricted form of model growth. It is logical to expect that as one arbitrarily adds more and more sets of model terms, the probability increases that two or more terms will be nearly linearly dependent (colinear). This would result in a large increase in the condition number of the least squares matrix A(i). This conjecture was also tested by evaluating search indicator I(j,13) for <u>all</u> of the terms added at each growth iteration by the block form techniques. In all of these cases, there were numerous occurrences of I(j,13) values greater than 0.90, and this appears to explain the observed ill-conditioning. Any growth techniques that do not check for and somehow handle colinearity among the model terms will have similar problems in characterizing systems. The Search Indicator Growth Algorithm effectively handles this problem with search indicator I(j,13).

Experiment 3

This experiment examines the performance of the various growth techniques when the system under test actually has a significant delay factor L (previously discussed in Chapter VI). The block-form techniques do not have any provision for recognizing this condition during the growth iterations, and therefore include unnecessary model terms.

We synthesize the following nonlinear system.

$$y(n) = 1.0\,u(n-4) + .8\,u(n-5) - .4y(n-1) + .15\,u(n-5)y(n-2) \quad \{7.3\}$$

Using the same input sequence (length N = 200) as Experiment 2, we probe Eq. {7.3} to produce the system output sequence {y(n)}. We grow models by the M Directed, D Directed, Neighbor, and Search Indicator techniques. The other three block-form techniques would require more than the available 200 measurements to evaluate a BVM(2,5) model, and it was decided not to include them.

We started the Search Indicator Growth Algorithm by initially considering the candidate terms in BVM(1,9), the highest memory linear model that could be handled by the computer program. The largest value of I(j,12) was used to specify which term to include in the first model. Using the degree and memory of this first selected term, we heuristically consider the candidate set specified by the BVM with one increase in degree and one increase in memory. The condition number and error fit for each model are evaluated at each iteration, and the results are presented

in Table 11. Table 12 contains the full details of the more compact characterization by the Search Indicator Growth Algorithm.

METHOD NAME	ITERATION i	MODEL	NEW TERMS	SELECTED TERMS	TOTAL TERMS	CONDITION NUMBER	SQUARE ROOT OF FITTING ERROR; J
M DIRECTED	1	BVM(1,1)	3	3	3	.1254E+01	.1266E+01
	2	BVM(1,2)	2	2	5	.1813E+01	.1250E+01
	3	BVM(1,3)	2	2	7	.2411E+01	.1244E+01
	4	BVM(1,4)	2	2	9	.2820E+01	.3829E+00
	5	BVM(1,5)	2	2	11	.4635E+02	.2274E+00
	6	BVM(1,6)	2	2	13	.2048E+03	.2272E+00
(see footnote 9)	6*	BVM(2,5)	66	66	77	.1425E+05	.2293E-04
D DIRECTED	1	BVM(1,1)	3	3	3	.1254E+01	.1266E+01
	2	BVM(2,1)	6	6	9	.7462E+01	.1244E+01
	3	BVM(3,1)	10	10	19	.3730E+03	.1224E+01
(see footnote 9)	3*	BVM(2,2)	11	11	20	.1705E+02	.1194E+01
	4	BVM(2,3)	15	15	35	.3965E+02	.1124E+01
	5	BVM(2,4)	19	19	54	.7172E+02	.1929E+00
	6	BVM(2,5)	23	23	77	.1424E+05	.1974E-05
NEIGHBOR	1	BVM(1,1)	3	3	3	.1254E+01	.1266E+01
	2A	BVM(2,1)	6	6	9	.7462E+01	.1244E+01
	2B	BVM(1,2)	3	3	6	.1813E+01	.1250E+01
	2C	BVM(2,2)*	17	17	20	.1705E+02	.1194E+01
	3A	BVM(3,2)	35	35	55	.1035E+04	.1062E+01
	3B	BVM(2,3)	15	15	35	.3965E+02	.1124E+01
	3C	BVM(3,3)*	99	99	119	.5051E+04	.7120E+00
(footnote 10)	4	BVM(2,4)	19	19	138	.6516E+04	.8605E-01
	5	BVM(2,5)	23	23	161	.2571E+06	.5546E-04
SEARCH INDICATOR	1	BVM(1,9)	19	1	1	.1000E+01	.5803E+00
	2	BVM(2,5)	76	2	3	.1802E+02	.2316E+00
	3	BVM(2,5)	74	2	5	.3206E+02	.6653E-06
EXACT MODEL OF THE SYSTEM			4	4	4	.1802E+02	.5986E-06

TABLE 11: Summary Results from Experiment 3

---

10 Note that the Neighbor Search technique selected the BVM(3,3) model over the others at iteration 3. Since we only have 200 data points in our measurement sequences, we cannot continue to evaluate the BVM neighbors of BVM(3,3). We chose to modify the neighbor growth algorithm and consider the new terms specified by the next model we can encompass, BVM(2,4). Therefore at iteration 4, our overall model contains terms from both BVM(3,3) and BVM(2,4). In a similar manner we cannot evaluate BVM(3,4) because it would require at least 219 data points. We chose to use the same modification at iteration 5 and consider the new terms specified by the next model we can encompass, BVM(2,5). Therefore at iteration 5, our overall model contains terms from both BVM(3,3) and BVM(2,5). Had we not made this modification, results similar to experiment number 4 would be obtained. If we had more data points, we would have been able to follow the unmodified growth algorithms and would have obtained results similar to those of experiment number 2.

Experiment: 3 Iteration: 1 Candidate Model: BVM(1,9)

Number of candidate model terms, q(i) = 19

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n-4)	.1409E+01	-	-

Number of terms in final subset (marked with *), N_f = 1

Total number of terms in resulting model, c(i) = 1

Condition Number of least squares matrix A(i), N_c = .1000E+01

Square root of the fitting error, J(i) = .5803E+00

Remarks: We picked the one candidate model term from the candidate set with the highest I(j,12).

Experiment: 3 Iteration: 2 Candidate Model: BVM(2,5)

Number of candidate model terms, q(i) = 76

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n-5)	.2278E+00	.7982E+00	y(n-1)
2*	y(n-1)	.1029E+00	.7982E+00	u(n-5)

Number of terms in final subset (marked with *), N_f = 2

Total number of terms in resulting model, c(i) = 3

Condition Number of least squares matrix A(i), N_c = .1802E+02

Square root of the fitting error, J(i) = .2316E+00

Remarks: The first phase picked terms with I(j,12) > 0.70E-01 and the second phase kept terms with I(j,13) < 0.90

Table 12: Search Indicator Growth Algorithm results of Experiment 3 (continued on next page).

Experiment: 3 Iteration: 3 Candidate Model: BVM(2,5)

Number of candidate model terms, q(i) = 74

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n-5)y(n-2)	.5363E-01	.8158E+00	y(n-1)y(n-2)
2*	y(n-1)y(n-2)	.4383E-01	.8158E+00	u(n-5)y(n-2)

Number of terms in final subset (marked with *), N_f = 2

Total number of terms in resulting model, c(i) = 5

Condition Number of least squares matrix A(i), N_c = .3206E+02

Square root of the fitting error, J(i) = .6653E-06

Remarks: The first phase picked terms with I(j,12) > 0.10E-01
and the second phase kept terms with I(j,13) < 0.90

TABLE 11: (continued)

This experiment shows that the Search Indicator Growth Algorithm can provide a better conditioned solution (over 2 orders of magnitude lower) than the other growth techniques when the system has a significant delay factor L. The block-form techniques used in this experiment converged on a larger model with reasonably small fitting error. These solutions however had higher condition numbers and required a significantly larger number of multiplicative operations.

The search indicator algorithm converged on the following model equation;

$$y(n) = 1.0000E+0\ u(n-4) + .8000E+0\ u(n-5) - .4000E+00\ y(n-1) + .1897E-7\ u(n-1)u(n-2) + .1500E+0\ u(n-5)y(n-2) \quad \{7.4\}$$

We experimentally developed the technique used to specify the subsequent sets of candidate model terms based on the terms selected in the first iteration. It is denoted as the Candidate Model Specification Technique in the work that follows. This heuristic technique works well but it is acknowledged that there undoubtedly are cases where it may fail to specify a suitably inclusive set of candidate model terms. The resulting model may be suboptimal in these cases, and other candidate model term specification techniques need be considered.

The major strength of the Search Indicator Growth Algorithm is its ability to efficiently select the best performing model terms from the candidate set. It is important to insure that the candidate set is large enough. There is no known way to guarantee ahead of time that this goal is met. It remains necessary for the user of this algorithm to recognize when, and if, the candidate set is to be expanded.

Experiment 4

The purpose of this fourth experiment is to show that even for linear systems, the Search Indicator Growth Algorithm can provide more efficient systems characterization than the widely used recursive-in-order techniques like those of Box and Jenkins [Ref. 17]. This is a simplified example of what can also happen when block-form techniques are used on more general nonlinear systems.

Consider the following linear system equation.

$$y(n) = 1.0\,u(n) + .5\,u(n-3) + .3\,u(n-8) - .6\,y(n-3) - .4\,y(n-7) \quad \{7.5\}$$

Using the same input sequence (length N = 200) as Experiment 2, we probe Eq. {7.5} with {u(n)} to produce the system output sequence {y(n)}. We then grow models by the M Directed Growth technique (with d=1) and the Search Indicator Growth Algorithm. Fixing the degree at d=1 reduces the M Directed technique to an equivalent form of the Box and Jenkins technique. It is obvious that the other block form techniques would add many unneeded nonlinear terms, and they are therefore not considered here.

The condition number and error fit for each model are evaluated at each iteration, and the results are presented in Table 13. We also include the results of a direct least squares model evaluation using the exact form of the system as a comparison basis. Additional details of the characterization by the Search Indicator Growth Algorithm are presented in Table 14.

METHOD NAME	ITERATION i	MODEL	NEW TERMS	SELECTED TERMS	TOTAL TERMS	CONDITION NUMBER	SQUARE ROOT OF FITTING ERROR: J
M DIRECTED	1	BVM(1,1)	3	3	3	.6055E+01	.9201E+00
	2	BVM(1,2)	2	2	5	.1821E+02	.8724E+00
	3	BVM(1,3)	2	2	7	.2351E+02	.4932E+00
(footnote 1:)	4	BVM(1,4)	2	2	9	.9344E+02	.4931E+00
	5	BVM(1,5)	2	2	11	.1413E+03	.4728E+00
	6	BVM(1,6)	2	2	13	.1820E+03	.4630E+00
	7	BVM(1,7)	2	2	15	.2506E+03	.1820E+00
	8	BVM(1,8)	2	2	17	.1607E+04	.5616E-05
SEARCH	1	BVM(1,9)	19	1	1	.1000E+01	.1254E+01
INDICATOR	2	BVM(1,9)	18	2	3	.2525E+01	.5393E+00
	3	BVM(1,9)	16	3	6	.1378E+02	.5186E-06
EXACT MODEL OF THE SYSTEM			5	5	5	.9855E+01	.7104E-06

TABLE 13: Summary Results from Experiment 4

Experiment: 4 Iteration: 1 Candidate Model: BVM(1,9)

Number of candidate model terms, q(i) = 19

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)	.1163E+01	-	-

Number of terms in final subset (marked with *), N_f = 1

Total number of terms in resulting model, c(i) = 1

Condition Number of least squares matrix A(i), N_c = .1000E+01

Square root of the fitting error, J(i) = .1254E+01

Remarks: We picked the one candidate model term from the candidate set with the highest I(j,12).

Table 14: Search Indicator Growth Algorithm results of Experiment 4 (continued on next page).

Experiment: 4 Iteration: 2 Candidate Model: BVM(1,9)

Number of candidate model terms, q(i) = 18

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	y(n-7)	.1030E+01	.1288E+00	y(n-3)
2*	y(n-3)	.6925E+00	.1288E+00	y(n-7)

Number of terms in final subset (marked with *), N_f = 2

Total number of terms in resulting model, c(i) = 3

Condition Number of least squares matrix A(i), N_c = .2523E+01

Square root of the fitting error, J(i) = .5393E+00

Remarks: The first phase picked terms with I(j,12) > 0.60E+00 and there was no required reduction in phase 2.

Experiment: 4 Iteration: 3 Candidate Model: BVM(1,9)

Number of candidate model terms, q(i) = 16

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n-8)	.1272E+00	.3897E+00	y(n-8)
2*	y(n-8)	.1088E+00	.3897E+00	u(n-8)
3*	u(n-3)	.7227E-01	.8644E-02	y(n-8)

Number of terms in final subset (marked with *), N_f = 3

Total number of terms in resulting model, c(i) = 6

Condition Number of least squares matrix A(i), N_c = .1378E+02

Square root of the fitting error, J(i) = .5186E-06

Remarks: The first phase picked terms with I(j,12) > 0.50E-01 and there was no required reduction in phase 2.

TABLE 14: (continued)

This experiment shows that the Search Indicator Growth Algorithm can converge on an accurate model of a linear system in fewer iterations than the M Directed Growth technique. The condition number of the least squares matrix A(i) is significantly lower, and therefore the variance of the model coefficients is lower when the Search Indicator technique is used. We also achieved a lower error and found that there was little dependency among the final model terms. The following model was obtained with the Search Indicator Growth Algorithm.

$$y(n) = 1.0000E+1\ u(n) - .6000E+0\ y(n-3) - .4000E+0\ y(n-7)$$
$$- .5000E+0\ u(n-3) + .3000E+0\ u(n-8) - .3847E-7\ y(n-8) \quad \{7.6\}$$

The main reason these results were obtained, is the particular form of Eq. {7.5}. The M Directed technique could not take advantage of the fact that there were unnecessary terms in a full BVM(1,8) form, and consequently had to include all 17 of the terms. Tables 13 and 14 show how the Search Indicator Growth Algorithm efficiently converged on an adequate model, based on performance evaluation of the set of candidate model terms.[11]

---

11 A conventional growth stopping criterion in the literature [Ref. 17] is when the fitting error J stops decreasing significantly. In this example, the M Directed growth algorithm (with d=1) could therefore indicate that growth should stop at iteration 4; this could result in an inferior model.

Experiment 5

This experiment shows how the finite length of the measurement sequences prevents the block-form growth techniques from converging to an accurate model. Chapter VI described how these block-form techniques could not be used to evaluate models with more terms than the number of data measurements in the sequences. The net effect is that only a limited set of model terms can be considered, based on the number of available measurements. The Search Indicator Growth Algorithm is shown to be unaffected by the size of the data sequences, and is capable of considering a nearly unlimited number of candidate model terms. The ability to efficiently evaluate a very large set of candidate model terms, and cut down to a small and meaningful subset, is one of the main strengths of the algorithm.

We synthesize the following nonlinear system.

$$\begin{aligned} y(n) = {} & 1.0\,u(n) + .8\,u(n-1) + .6\,u(n-2) + .45\,u(n-3) \\ & -.9\,y(n-1) - .7\,y(n-2) - .25\,y(n-3) + .1\,u(n)u(n-1)u(n-2) \\ & -.15\,y(n-1)y(n-2)y(n-3) - .35\,u(n-2)y(n-3) \\ & +.05\,y(n-1)y(n-2) - .18\,y(n-2)y(n-2)y(n-3) \end{aligned} \qquad \{7.7\}$$

We use a random input probe (length N = 100) uniformly distributed between the limits of -1 and +1. The system output sequence {y(n)} is produced by probing the system of Eq. {7.7} with the input sequence {u(n)}. Starting with the evaluation of the base model BVM(1,1), we recursively grow the model by each of the six block-form growth techniques of

Chapter V and the Search Indicator Growth Algorithm of Chapter VI. Whenever a growth technique reaches the point where insufficient data measurements are available, we stop the growth. The Candidate Model Specification Technique is used for the search indicator growth, starting with the initial model BVM(1,9).

The condition number and fitting error for each model are evaluated at each iteration, and the results are presented in Table 15. We include the results of a direct least squares model evaluation using the exact form of the system as a comparison basis. Table 16 contains additional details of the more successful characterization by the Search Indicator Growth Algorithm.

METHOD NAME	ITERATION i	MODEL	NEW TERMS	SELECTED TERMS	TOTAL TERMS	CONDITION NUMBER	SQUARE ROOT OF FITTING ERROR: J
M DIRECTED	1	BVM(1,1)	3	3	3	.2181E+02	.2347E+00
	2	BVM(1,2)	2	2	5	.4791E+02	.1901E+00
	3	BVM(1,3)	2	2	7	.1495E+03	.1683E+00
	4	BVM(1,4)	2	2	9	.2941E+03	.1630E+00
	4*	BVM(2,3)	28	28	35	.1886E+06	.5282E-01
D DIRECTED	1	BVM(1,1)	3	3	3	.2181E+02	.2347E+00
	2	BVM(2,1)	6	6	9	.5142E+03	.2170E+00
	3	BVM(3,1)	10	10	19	.3333E+05	.1889E+00
	4	BVM(4,1)	21	21	40	.4042E+07	.1502E+00
	4*	BVM(3,2)	36	36	55	.1931E+07	.7899E-01
DIAGONAL	1	BVM(1,1)	3	3	3	.2181E+02	.2347E+00
	2	BVM(2,2)	17	17	20	.5166E+04	.1526E+00
M-D ZIG-ZAG	1	BVM(1,1)	3	3	3	.2181E+02	.2347E+00
	2	BVM(1,2)	2	2	5	.4791E+02	.1901E+00
	3	BVM(2,2)	15	15	20	.5166E+04	.1526E+00
	4	BVM(2,3)	15	15	35	.1886E+06	.5282E-01
D-M ZIG-ZAG	1	BVM(1,1)	3	3	3	.2181E+02	.2347E+00
	2	BVM(2,1)	6	6	9	.5142E+03	.2170E+00
	3	BVM(2,2)	11	11	20	.5166E+04	.1526E+00
	4	BVM(3,2)	35	35	55	.1931E+07	.7899E-01
NEIGHBOR	1	BVM(1,1)	3	3	3	.2181E+02	.2347E+00
	2A	BVM(2,1)	6	6	9	.5142E+03	.2170E+00
	2B	BVM(1,2)	3	3	6	.4791E+02	.1901E+00
	2C	BVM(2,2)*	17	17	20	.5166E+04	.1526E+00
	3A	BVM(3,2)	35	35	55	.1931E+07	.7899E-01
	3B	BVM(2,3)*	15	15	35	.1886E+06	.5282E-01
	4	BVM(2,4)	19	19	54	.2706E+07	.3024E-01
	5	BVM(2,5)	23	23	77	.3300E+08	.1313E-01
SEARCH INDICATOR	1	BVM(1,9)	19	2	2	.1686E+01	.2878E+00
	2	BVM(3,3)	117	4	6	.1037E+03	.1667E+00
	3	BVM(3,3)	113	4	10	.2329E+03	.1438E+00
	4	BVM(3,3)	109	1	11	.2343E+03	.1287E+00
	5	BVM(3,3)	108	8	19	.6544E+03	.1006E+00
	6	BVM(3,3)	100	3	22	.6858E+03	.8167E-01
	7	BVM(3,3)	97	7	29	.1137E+04	.6570E-01
	8	BVM(3,3)	90	3	32	.1303E+04	.1429E-01
	9	BVM(3,3)	87	1	33	.1405E+04	.6545E-02
	10	BVM(3,3)	86	2	35	.1966E+04	.3882E-05
EXACT MODEL OF THE SYSTEM			12	12	12	.1707E+03	.7572E-06

TABLE 15: Summary Results from Experiment 5

Experiment: 5 Iteration: 1 Candidate Model: BVM(1,9)

Number of candidate model terms, q(i) = 2

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)	.3986E+00	-	-
2*	y(n-2)	.2369E-01	-	-

Number of terms in final subset (marked with *), N_f = 2

Total number of terms in resulting model, c(i) = 2

Condition Number of least squares matrix A(i), N_c = .1686E+01

Square root of the fitting error, J(i) = .2878E+00

Remarks: These candidate model terms has values of I(j,12) that were far greater than those of the other terms.

TABLE 16: Search Indicator Growth Algorithm results of Experiment 5 (continued on next page).

Experiment: 5 Iteration: 2 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 117

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n-1)u(n-3)y(n-2)	.2714E-01	.8895E+00	u(n-1)y(n-2)y(n-3)
2*	u(n-2)y(n-3)	.2600E-01	.1002E+00	u(n-1)u(n-2)u(n-3)
3*	u(n-1)u(n-2)u(n-3)	.2520E-01	.8434E+00	u(n-1)u(n-3)y(n-2)
4	u(n-1)y(n-2)y(n-3)	.2484E-01	.8895E+00	u(n-1)u(n-3)y(n-2)
5*	u(n-2)u(n-2)y(n-1)	.2339E-01	.1927E+00	u(n-1)u(n-2)u(n-3)

Number of terms in final subset (marked with *), N_f = 4

Total number of terms in resulting model, c(i) = 6

Condition Number of least squares matrix A(i), N_c = .1037E+03

Square root of the fitting error, J(i) = .1667E+00

Remarks: The first phase picked terms with I(j,12) > .20E-01
and the second phase kept terms with I(j,13) < 0.85

TABLE 16: (continued)

Experiment: 5 Iteration: 3 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 113

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	y(n-3)	.5916E-02	.8253E+00	u(n-3)
2*	u(n-3)	.4575E-02	.8253E+00	y(n-3)
3*	u(n-1)u(n-1)y(n-3)	.4070E-02	.6100E+00	y(n-3)
4*	u(n)u(n)y(n-3)	.3733E-02	.5592E+00	y(n-3)

Number of terms in final subset (marked with *), N_f = 4

Total number of terms in resulting model, c(i) = 10

Condition Number of least squares matrix A(i), N_c = .2329E+03

Square root of the fitting error, J(i) = .1438E+00

Remarks: The first phase picked terms with I(j,12) > 0.3200E-02
and the second phase kept terms with I(j,13) < 0.85

TABLE 16: (continued)

Experiment: 5 Iteration: 4 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 109

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n-1)y(n-3)	.3760E-02	.8650E+00	u(n-1)u(n-3)
2	u(n-1)u(n-3)	.2548E-02	.8650E+00	u(n-1)y(n-3)

Number of terms in final subset (marked with *), N_f = 1

Total number of terms in resulting model, c(i) = 11

Condition Number of least squares matrix A(i), N_c = .2343E+03

Square root of the fitting error, J(i) = .1287E+00

Remarks: The first phase picked terms with I(j,12) > 0.2000E-02
and the second phase kept terms with I(j,13) < 0.85

TABLE 16: (continued)

Experiment: 5 Iteration: 5 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 108

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)u(n-3)y(n-1)	.1194E-02	.8799E+00	u(n)y(n-1)y(n-3)
2	u(n)y(n-1)y(n-3)	.1081E-02	.8799E+00	u(n)u(n-3)y(n-1)
3*	u(n)u(n-2)u(n-3)	.8264E-03	.8426E+00	u(n)u(n-2)y(n-3)
4*	y(n-2)y(n-2)y(n-3)	.7172E-03	.6808E+00	u(n-3)y(n-2)y(n-2)
5*	y(n-1)	.6958E-03	.6116E+00	u(n)u(n)y(n-1)
6*	u(n-3)y(n-2)y(n-2)	.6735E-03	.6808E+00	y(n-2)y(n-2)y(n-3)
7*	u(n)u(n)y(n-1)	.6681E-03	.6116E+00	y(n-1)
8*	u(n-3)y(n-1)y(n-3)	.6619E-03	.9303E+00	y(n-1)y(n-3)y(n-3)
9	y(n-1)y(n-3)y(n-3)	.6322E-03	.9303E+00	u(n-3)y(n-1)y(n-3)
10*	u(n)u(n-2)y(n-3)	.6199E-03	.8426E+00	u(n)u(n-2)u(n-3)

Number of terms in final subset (marked with *), N_f = 8

Total number of terms in resulting model, c(i) = 19

Condition Number of least squares matrix A(i), N_c = .6544E+03

Square root of the fitting error, J(i) = .1006E+00

Remarks: The first phase picked terms with I(j,12) > .6000E-03
and the second phase kept terms with I(j,13) < 0.85

TABLE 16: (continued)

Experiment: 5 Iteration: 6 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 100

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)u(n-3)	.4679E-03	.4053E-01	u(n-3)y(n-1)
2*	u(n-3)y(n-1)	.4498E-03	.4053E-01	u(n)u(n-3)
3*	u(n-3)u(n-3)	.4242E-03	.3640E-01	u(n)u(n-3)

Number of terms in final subset (marked with *), N_f = 3

Total number of terms in resulting model, c(i) = 22

Condition Number of least squares matrix A(i), N_c = .6858E+03

Square root of the fitting error, J(i) = .8167E-01

Remarks: The first phase picked terms with I(j,12) > 0.33E-03
and the second phase kept terms with I(j,13) < 0.85

TABLE 16: (continued)

Experiment: 5 Iteration: 7 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 97

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n-1)u(n-1)u(n-1)	.1280E-03	.8497E+00	u(n-1)
2*	u(n-1)u(n-3)u(n-3)	.1189E-03	.5654E+00	u(n-1)
3*	y(n-1)y(n-2)	.1173E-03	.7726E+00	u(n-2)y(n-1)
4*	u(n)u(n-1)y(n-2)	.1002E-03	.6958E-01	u(n-2)y(n-1)
5*	u(n-2)y(n-1)	.9235E-04	.7726E+00	y(n-1)y(n-2)
6*	u(n-2)u(n-2)	.9088E-04	.3713E-01	u(n-1)u(n-3)u(n-3)
7*	u(n-1)	.9021E-04	.8497E+00	u(n-1)u(n-1)u(n-1)

Number of terms in final subset (marked with *), N_f = 7

Total number of terms in resulting model, c(i) = 29

Condition Number of least squares matrix A(i), N_c = .1137E+04

Square root of the fitting error, J(i) = .6570E-01

Remarks: The first phase picked terms with I(j,12) > 0.90E-04
and the second phase kept terms with I(j,13) < 0.85

TABLE 16: (continued)

Experiment: 5 Iteration: 8 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 90

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)y(n-1)	.1754E-03	.8446E+00	u(n)u(n-1)
2*	u(n-2)	.1292E-03	.1742E-01	u(n)u(n-1)
3*	u(n)u(n-1)	.1242E-03	.8466E+00	u(n)y(n-1)

Number of terms in final subset (marked with *), N_f = 3

Total number of terms in resulting model, c(i) = 32

Condition Number of least squares matrix A(i), N_c = .1303E+04

Square root of the fitting error, J(i) = .1429E-01

Remarks: The first phase picked terms with I(j,12) > 0.105E-03 and the second phase kept terms with I(j,13) < 0.85

Experiment: 5 Iteration: 9 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 87

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	y(n-1)y(n-2)y(n-3)	.1032E-04	-	-

Number of terms in final subset (marked with *), N_f = 1

Total number of terms in resulting model, c(i) = 33

Condition Number of least squares matrix A(i), N_c = .1405E+04

Square root of the fitting error, J(i) = .6545E-02

Remarks: The first phase picked terms with I(j,12) > 0.60E-05

TABLE 16: (continued)

Experiment: 5 Iteration: 10 Candidate Model: BVM(3,3)

Number of candidate model terms, q(i) = 86

Candidate Model Terms After First Phase Reduction:

#	Term	I(j,12)	I(j,13)	Related to Term
1*	u(n)u(n-2)y(n-1)	.3232E-05	.8468E+00	u(n)u(n-1)u(n-2)
2*	u(n)u(n-1)u(n-2)	.2892E-05	.7467E+00	u(n)u(n-2)y(n-1)

Number of terms in final subset (marked with *), N_f = 2

Total number of terms in resulting model, c(i) = 35

Condition Number of least squares matrix A(i), N_c = .1966E+04

Square root of the fitting error, J(i) = .3882E-05

Remarks: The first phase picked terms with I(j,12) > 0.15E-06
and the second phase kept terms with I(j,13) < 0.85

TABLE 16: (continued)

Table 15 shows that the first six growth techniques all failed to converge on an adequate model because they exhausted the available data. The Neighbor Growth technique came closest to generating an adequate model, but it also had to restrict its model growth choices.

Only the Search Indicator Growth Algorithm found an acceptable model. Note that the condition number of this last model (iteration 10) is reasonably close to that of the exact model of the system, despite the fact that we have nearly three times the required number of model terms.

The Search Indicator Growth Algorithm considered a total of 131 different model terms specified by our selection technique, even though there were only 100 data points.

This ability enabled it to locate the best model terms to accept over the ten growth iterations. There were a number of iterations where near colinearity was detected, and Table 16 shows how the corresponding term with the lowest value of I(j,12) was deleted from consideration. This last model contains 23 extra model terms, but these unnecessary terms are easily identifiable by their low coefficient estimates. The equation obtained for this model is;

$$
\begin{aligned}
y(n) = {} & 1.0000E{+}1\ u(n) - .7000E{+}0\ y(n-2) - .3500E{+}0\ u(n-2)y(n-3) \\
& - .5051E{-}5\ u(n-1)u(n-2)u(n-3) - .6385E{-}5\ u(n-2)y(n-1) \\
& - .1186E{-}5\ u(n-2)u(n-2)y(n-1) + .4500E{+}0\ u(n-3) \\
& - .2500E{+}0\ y(n-3) - .1266E{-}4\ u(n-1)u(n-3)y(n-2) \\
& + .4830E{-}5\ u(n)u(n)y(n-3) - .1116E{-}4\ u(n-1)u(n-1)u(n-1) \\
& - .9000E{+}0\ y(n-1) + .2841E{-}4\ u(n-1)u(n-3)u(n-3) \\
& - .5353E{-}5\ u(n-1)u(n-1)y(n-3) - .1489E{-}4\ u(n-1)y(n-3) \\
& - .2538E{-}4\ u(n)u(n-2)u(n-3) - .1800E{+}0\ y(n-2)y(n-2)y(n-3) \\
& - .4268E{-}5\ u(n)u(n)y(n-1) + .2956E{-}4\ u(n)u(n-2)y(n-3) \\
& + .1132E{-}5\ u(n)u(n-3)y(n-1) - .2190E{-}4\ u(n-3)y(n-1)y(n-3) \\
& + .1779E{-}5\ u(n-3)y(n-2)y(n-2) + .9621E{-}6\ u(n)u(n-3) \\
& + .4792E{-}5\ u(n-3)u(n-3) + .1190E{-}4\ u(n-3)y(n-1) \\
& + .8000E{+}0\ u(n-1) - .1500E{+}0\ y(n-1)y(n-2)y(n-3) \\
& - .2880E{-}5\ u(n-2)u(n-2) + .5000E{-}1\ y(n-1)y(n-2) \\
& + .8200E{-}5\ u(n)u(n-1) - .9998E{-}1\ u(n)u(n-1)u(n-2) \\
& - .1520E{-}5\ u(n)u(n-1)y(n-2) + .6000E{+}0\ u(n-2) \\
& - .1210E{-}4\ u(n)y(n-1) + .2016E{-}4\ u(n)u(n-2)y(n-1) \qquad \{7.8\}
\end{aligned}
$$

Experiment 6

This experiment investigates the degraded model growth resulting from additive output noise, and the improvement that can result when we have control over the input sequence. Chapter 6 discussed a method that uses repeated application of the identical system input, and calculates a point-for-point ensemble average of the corresponding sets of system outputs to form an "averaged" system output sequence $\{\bar{y}(n)\}$. Model growth is then attempted using the input sequence $\{u(n)\}$ and this averaged sequence $\{\bar{y}(n)\}$.

We synthesize the following nonlinear system.

$$y(n) = 1.0\,u(n) + .8\,u(n-1) - .4\,y(n-1) + .15\,u(n-1)y(n-2) + v(n) \qquad \{7.9\}$$

We generate a random input sequence $\{u(n); 1<=n<=100\}$ uniformly distributed between the amplitude limits (-2,2), and a random additive noise sequence $\{v(n); 1<=n<=100\}$ uniformly distributed between the amplitude limits (-1,1). The sequence $\{v(n)\}$ is produced with a different random seed and is uncorrelated with the input. We produce the noisy system output from Eq. {7.9} and grow models by the Search Indicator Growth Algorithm. Growth is halted when the fitting error stops decreasing significantly, or when the condition number jumps drastically. These results are summarized in the first section of Table 17.

After reapplying the input probe to the system a number of times, an ensemble average of the corresponding system

output sequences is performed, and we form the "averaged" system output sequence $\{\tilde{y}(n)\}$. Various experiments are conducted with an increasing number of output sequences (ensemble members) used to produce $\{\tilde{y}(n)\}$. These results are included in Table 17 for 4, 10, 40 and 100 ensemble member averages. A direct least squares evaluation of the model, with the exact form of the system and no measurement noise, is included for comparison purposes.

METHOD NAME	ITERATION i	MODEL	NEW TERMS	SELECTED TERMS	TOTAL TERMS	CONDITION NUMBER	SQUARE ROOT OF FITTING ERROR; J
SEARCH	1	BVM(1,9)	19	2	2	.1307E+01	.7014E+00
INDICATOR	2	BVM(2,2)	18	2	4	.3273E+01	.5950E+00
(1 AVERAGE)	3	BVM(2,2)	16	2	6	.1553E+02	.5550E+00
	4	BVM(2,2)	14	2	8	.4703E+02	.5392E+00
	5	BVM(2,2)	12	1	9	.4740E+02	.5370E+00
(See Eq. [7.10])	6	BVM(2,2)	11	6	15	.3704E+03	.5221E+00
SEARCH	1	BVM(1,9)	19	2	2	.1307E+01	.3784E+00
INDICATOR	2	BVM(2,2)	18	2	4	.1745E+02	.2671E+00
(4 AVERAGES)	3	BVM(2,2)	16	1	5	.1766E+02	.1955E+00
	4	BVM(2,2)	15	1	6	.1846E+02	.1832E+00
	5	BVM(2,2)	14	1	7	.2340E+02	.1671E+00
(See Eq. [7.11])	6	BVM(2,2)	13	2	9	.3672E+03	.1419E+00
SEARCH	1	BVM(1,9)	19	2	2	.1307E+01	.3492E+00
INDICATOR	2	BVM(2,2)	18	1	3	.2443E+01	.2285E+00
(10 AVERAGES)	3	BVM(2,2)	17	1	4	.2613E+01	.1448E+00
(See Eq. [7.12])	4	BVM(2,2)	16	4	8	.6615E+03	.5956E-01
SEARCH	1	BVM(1,9)	19	2	2	.1307E+01	.3422E+00
INDICATOR	2	BVM(2,2)	18	1	3	.2443E+01	.2196E+00
(40 AVERAGES)	3	BVM(2,2)	17	1	4	.2618E+00	.1350E+00
(See Eq. [7.13])	4	BVM(2,2)	16	3	7	.5214E+03	.1617E-01
SEARCH	1	BVM(1,9)	19	2	2	.1307E+01	.3415E+00
INDICATOR	2	BVM(2,2)	18	1	3	.2444E+01	.2189E+00
(100 AVERAGES)	3	BVM(2,2)	17	1	4	.2619E+01	.1346E+00
(See Eq. [7.14])	4	BVM(2,2)	16	3	7	.5190E+03	.6459E-02
EXACT (NO NOISE) SYSTEM MODEL			4	4	4	.2597E+02	.1208E-05

TABLE 17: Summary Results from Experiment 6

This experiment shows how control over the system input can be used to reduce the distorting effect of additive output noise on system characterization. Both the fitting error and the condition number are reduced as we average more data sequences. The Search Indicator Growth Algorithm converges quicker and on a more compact model. The averaging technique reduces the effect of the output noise by a factor equal to the reciprocal of the number of ensemble averages used. It is interesting to note that J(i), the square root of the fitting error, dropped by approximately the same factor.

Equations {7.10}, {7.11}, {7.12}, {7.13} and {7.14} are the resulting model equations obtained from the last iteration of the growth tests with 1, 4, 10, 40 and 100 ensemble averages respectively. The actual system equation is repeated below for comparison.

$$y(n) = 1.0\,u(n) + .8\,u(n-1) - .4\,y(n-1) + .15\,u(n-1)y(n-2) + v(n) \qquad \{7.9\}$$

<u>One Average</u>:

$$\begin{aligned} y(n) = {} & .10149E+1\,u(n) + .45434E+0\,u(n-1) - .18148E+0\,u(n-2) \\ & + .20995E+0\,u(n-1)y(n-2) + .14323E+0\,u(n)u(n) \\ & + .19578E-1\,y(n-2) - .24651E-1\,u(n-2)u(n-2) \\ & - .81504E-1\,u(n-2)y(n-2) - .96112E-1\,u(n)u(n-1) \\ & + .59792E-1\,u(n-1)u(n-1) - .78196E-1\,y(n-1)y(n-2) \\ & + .41998E-1\,y(n-2)y(n-2) + .67933E-1\,u(n)y(n-1) \\ & - .18208E-1\,u(n)y(n-2) - .36490E-1\,u(n-1)y(n-1) \qquad \{7.10\} \end{aligned}$$

Note that the model growth did not select the system term y(n-1). The additive output noise is degrading the model growth capabilities.

Four Averages:

$$y(n) = .10179E+1\ u(n) + .59065E+0\ u(n-1) - .94970E-2\ u(n-1)u(n-2) - .14650E+0\ u(n-2) + .46875E-1\ u(n)u(n) - .29741E-1\ u(n-2)y(n-2) - .18258E+0\ y(n-1) + .59640E-1\ y(n-2) + .16499E+0\ u(n-1)y(n-2) \quad \{7.11\}$$

Ten Averages:

$$y(n) = .10058E+1\ u(n) + .74664E+0\ u(n-1) + .15180E+0\ u(n-1)y(n-2) - .31808E-1\ u(n-2) - .34365E+0\ y(n-1) + .90579E-2\ y(n-2) + .16271E-1\ u(n)u(n) - .47212E-2\ y(n-2)y(n-2) \quad \{7.12\}$$

Forty Averages:

$$y(n) = .10003E+1\ u(n) + .80620E+0\ u(n-1) + .14979E+0\ u(n-1)y(n-2) + .76613E-2\ u(n-2) - .40594E+0\ y(n-1) - .60805E-2\ y(n-2) + .13304E-2\ y(n-2)y(n-2) \quad \{7.13\}$$

One Hundred Averages:

$$y(n) = .10001E+1\ u(n) + .80304E+0\ u(n-1) + .14990E+0\ u(n-1)y(n-2) + .35466E-2\ u(n-2) - .40295E+0\ y(n-1) - .26935E-2\ y(n-2) + .56228E-3\ y(n-2)y(n-2) \quad \{7.14\}$$

The results presented in Table 17, and in the preceding equations, clearly demonstrate the significant improvements in model growth and model accuracy that can be obtained when we can reduce the effect of additive output noise by averaging. This averaging and growth technique is useful whenever the statistics of the additive output noise do not change during the experiment.

Experiment 7

The purpose of this experiment is to demonstrate the model growth improvement resulting from use of the two-stage "N-R" technique discussed at the end of Chapter VI. This technique is applicable to model growth when we have only one set of system input and output measurement sequences, and the output sequence contains additive noise. We also develop alternate criteria for evaluating the fit of a model.

For this experiment, we use the same nonlinear system as in Experiment 6.

$$y(n) = 1.0\,u(n) + .8\,u(n-1) - .4\,y(n-1) + .15\,u(n-1)y(n-2) + v(n) \qquad \{7.15\}$$

The input sequence $\{u(n); 1<=n<=1000\}$ is uniformly distributed between the amplitude limits (-2,2), and the additive noise sequence $\{v(n)\}$ is uniformly distributed between the amplitude limits (-.2,.2). These two sequences are uncorrelated, both with themselves and each other. After generating the noisy system output sequence $\{y(n)\}$

corresponding to Eq. {7.15}, we grow a recursive model using the Search Indicator Growth Algorithm and the Candidate Model Specification Technique. The additive output noise degrades the growth but this step is included to show the typical results obtained with noisy data sequences. This first modeling example is denoted as Test 1 and the results are summarized in Table 18.

Using the VOL model form of Eq. {2.4} and the Search Indicator Growth Algorithm, we next grow a nonrecursive, nonlinear model from the available measurement sequences (first phase of the N-R technique). We selected the system given by Eq. {7.15} to be of recursive nonlinear BVM form, and therefore any finite VOL model produced by our growth algorithm can only approximate the performance of the system. Since we used the Search Indicator Growth Algorithm, the resulting VOL model is more compact than any block form model using the VOL form. We give extra freedom (larger d and m) to the candidate VOL model terms to allow for improved growth performance. The results of this second modeling example are given as Test 2 in Table 18.

After evaluating the coefficients of our final VOL model from the preceding growth step, we synthesize it on the computer and probe it with our stored system input sequence {u(n)}. The resulting model output sequence {$\hat{y}$(n)} is stored, and used with {u(n)} to grow a recursive model of the BVM form using the Search Indicator Growth Algorithm

(second phase of the N-R technique). The results of this third modeling example are summarized as Test 3 in Table 18.

We include two direct least squares modeling examples using the exact form of the system. First we use $\{u(n)\}$ and $\{y(n)\}$ to obtain an evaluation of the correct model using actual system data. This is summarized as Test 4 in Table 18. Finally we use $\{u(n)\}$ and $\{\hat{y}(n)\}$ to obtain an evaluation of the correct model using the output data from the nonrecursive VOL model realization obtained in Test 2. This is summarized as Test 5 in Table 18.

TEST DESCRIPTION	TOTAL TERMS	CONDITION NUMBER	SQUARE ROOT OF FITTING ERROR; J	MINIMUM VALUE OF RESIDUAL SEQUENCE $\{e(n)\}$	MAXIMUM VALUE OF RESIDUAL SEQUENCE $\{e(n)\}$
1: SIGA GROWTH OF A RECURSIVE BVM USING NOISY SYSTEM DATA.	9	516.7	.1250	-.2893	.3027
2: SIGA GROWTH OF NON-RECURSIVE VOL MODEL USING NOISY SYSTEM DATA.	19	2.3	.1193	-.3048	.3401
3: SIGA GROWTH OF A RECURSIVE BVM USING OUTPUT DATA FROM VOL MODEL OF TEST 2.	7	423.6	.1147	-.2941	.2572
4: DIRECT EVALUATION OF EXACT MODEL USING NOISY SYSTEM DATA.	4	19.3	.1273	-.2971	.3169
5: DIRECT EVALUATION OF EXACT MODEL USING OUTPUT DATA FROM VOL MODEL OF TEST 2.	4	20.06	.1152	-.2961	.2510

TABLE 18: Summary Results from Experiment 7

Equations {7.16}, {7.17}, {7.18}, {7.19} and {7.20} are the resulting model equations obtained from the preceding Test 1, Test 2, Test 3, Test 4 and Test 5, respectively. The actual system equation is repeated below for comparison.

$$y(n) = 1.0\,u(n) + .8\,u(n-1) - .4\,y(n-1) + .15\,u(n-1)y(n-2) + v(n) \quad \{7.15\}$$

$$\begin{aligned} y(n) = {} & .10059E{+}1\,u(n) + .69807E{+}0\,u(n-1) + .10798E{-}3\,u(n-1)u(n-2) \\ & + .13629E{+}0\,u(n-1)y(n-2) - .79330E{-}3\,u(n-2) - .29862E{+}0\,y(n-1) \\ & + .35210E{-}3\,y(n-2) - .13068E{-}3\,y(n-2)y(n-2) \\ & + .14619E{-}3\,u(n-2)u(n-2) \end{aligned} \quad \{7.16\}$$

$$\begin{aligned} y(n) = {} & .10064E{+}1\,u(n) + .39759E{+}0\,u(n-1) - .16452E{+}0\,u(n-2) \\ & + .14916E{+}0\,u(n-1)u(n-2) + .64402E{-}3\,u(n-2) \\ & - .66326E{-}3\,u(n-2)u(n-3) - .26126E{-}3\,u(n-1)u(n-4) \\ & - .26433E{-}3\,u(n-2)u(n-4) + .28719E{-}3\,u(n-3)u(n-4) \\ & + .54585E{-}3\,u(n-1)u(n-3) + .85071E{-}4\,u(n-3)u(n-5) \\ & - .15249E{-}3\,u(n-4) + .12135E{-}3\,u(n-5) \\ & - .10625E{-}3\,u(n-4)u(n-6) + .59477E{-}4\,u(n)u(n-8) \\ & + .63433E{-}4\,u(n-1)u(n-5) + .56957E{-}4\,u(n-1)u(n-7) \\ & + .66455E{-}4\,u(n-2)u(n-5) - .61506E{-}4\,u(n-4)u(n-8) \end{aligned} \quad \{7.17\}$$

$$\begin{aligned} y(n) = {} & .10050E{+}1\,u(n) + .84321E{+}0\,u(n-1) + .14686E{+}0\,u(n-1)y(n-2) \\ & + .33216E{-}1\,u(n-2) - .44315E{+}0\,y(n-1) - .20647E{-}1\,y(n-2) \\ & + .25836E{-}3\,y(n-1)y(n-2) \end{aligned} \quad \{7.18\}$$

$$\begin{aligned} y(n) = {} & .10059E{+}1\,u(n) + .79303E{+}0\,u(n-1) - .39294E{+}0\,y(n-1) \\ & + .14394E{+}0\,u(n-1)y(n-1) \end{aligned} \quad \{7.19\}$$

$$\begin{aligned} y(n) = {} & .10047E{+}1\,u(n) + .81040E{+}0\,u(n-1) - .41073E{+}0\,y(n-1) \\ & + .14214E{+}0\,u(n-1)y(n-2) \end{aligned} \quad \{7.20\}$$

This experiment was designed to show the typical modeling improvement resulting from the two-stage "N-R" growth algorithm. The nonrecursive VOL model growth phase (Test 2) was continued until a fitting error value less than the model of Test 1 was obtained. Table 18 shows that a significantly larger but manageable number of model terms are required in this phase. The final VOL model has a very low condition number. This is due to both the nonrecursive nature of the VOL model form, and the property of the Search Indicator Growth Algorithm which only picks model terms offering substantial reduction in the fitting error.

Table 18 shows that the model of Test 3 has fewer terms, lower condition number, and lower fitting error than the model of Test 1, but not by much. The error of Test 4 is higher than Tests 1 through 3, but this is primarily due to the lower number of final model terms. The error of Test 5 is the lowest for this number of model terms.

We know that the models of Test 4 and Test 5 should be better than the models of Test 1 and Test 3, respectively, but it is difficult to recognize this from the values in the table. The additive output noise causes an offset in the fitting error and we cannot use just this scalar performance criterion J(i) to rate the quality of fit. We instead must find some additional characteristics of our obtained model fit to demonstrate that we have a meaningful and useful growth technique when there is additive output noise.

One possible measure of the quality of fit is the amplitude range of the error residual sequence {e(n)} at the end of each test. A wide spread between the maximum and minimum residual values would generally indicate a poor model fit. Conversely, a small spread (compared to the spread of the system output sequence {y(n)}) would generally indicate that we have a good fit. Table 18 includes the maximum and minimum values of {e(n)} for the last model obtained from each of the five tests. There is some difference in the spread in each of the five tests, but nothing significant enough to use as a criterion. The magnitude of the additive noise masks these performance properties and we must look for another characteristic.

Chapter VI mentioned that the residual error sequence would be a random sequence without any identifiable trends or patterns if we have adequately modeled the system. This condition can be qualitatively evaluated using a standard statistical technique in the literature [Ref. 17]. The normalized sample autocorrelation plot {r(k); k=0,1,2,...} of a random sequence should approach the following form.

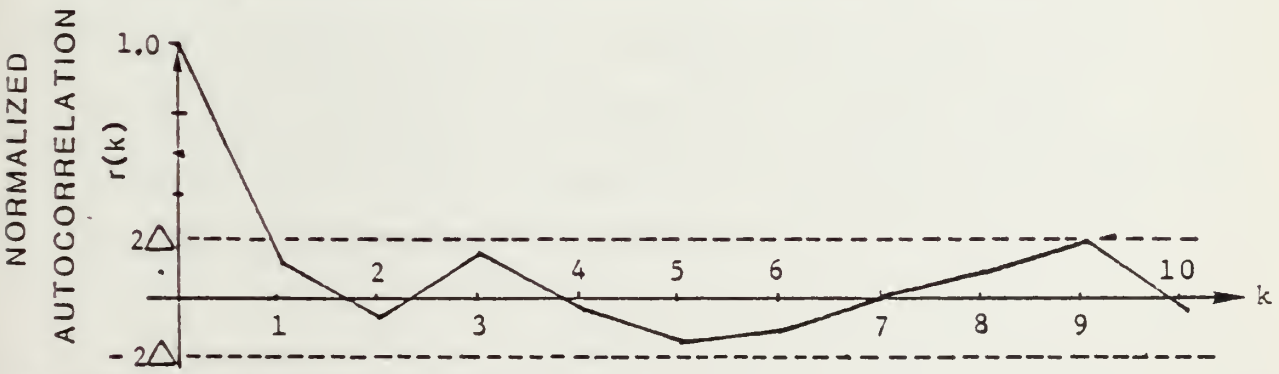


FIGURE 24: Typical autocorrelation plot of a random sequence

Here Δ equals the square root of the reciprocal of N, the number of data points used in the error minimization ($N = n_3 - n_2 + 1$). For example, N = 1000 produces the value $\Delta = 0.032$.

Following the conventions of Chapters II and IV, the following equation is used for the normalized sample autocorrelation of a signal {s(n)} at lag k.

$$r(k) = \frac{\frac{1}{N} \sum_{n=n_2}^{n_3} s(n)s(n-k)}{r(0)} \qquad \{7.21\}$$

A sequence is typically considered to be random if the values of {r(k);k=1,2,3,...} lie between $\pm 2\Delta$ for at least 95% of the normalized autocorrelation plot [Ref. 43].

The first seven normalized autocorrelation values of the error residual sequences from each of the five previous tests have been calculated from the experimental data. The autocorrelation values for the random additive noise sequence {v(n)} and the random input sequence {u(n)} have been calculated for comparison purposes, and these are denoted as Test 6 and Test 7, respectively. These autocorrelation values are summarized in Table 19 and graphically presented in Figure 25. A split format presentation is used in Figure 25 to better distinguish the different autocorrelation plots.

TEST DESCRIPTION	NORMALIZED AUTOCORRELATION VALUES r(1)	r(2)	r(3)	r(4)	r(5)	r(6)
1: SIGA GROWTH OF A RECURSIVE BVM USING NOISY SYSTEM DATA	.2928	.0095	.0018	-.0180	-.1320	.0112
2: SIGA GROWTH OF A NONRECURSIVE VOL MODEL USING NOISY SYSTEM DATA	.0258	.0286	.0020	-.0078	-.0572	.0451
3: SIGA GROWTH OF A RECURSIVE BVM USING OUTPUT DATA FROM VOL MODEL OF TEST 2	.0677	.0178	-.0127	-.0097	-.0641	.0391
4: DIRECT EVALUATION OF EXACT MODEL USING NOISY SYSTEM DATA	.3756	.0302	-.0106	-.0243	-.0446	.0111
5: DIRECT EVALUATION OF EXACT MODEL USING OUTPUT DATA FROM VOL MODEL OF TEST 2	.0570	.0160	-.0163	-.0092	-.0523	.0385
6: ADDITIVE NOISE SEQUENCE	.0618	.0174	-.0176	.0055	-.0685	.0514
7: INPUT SEQUENCE	.0162	.0542	.0256	-.0283	-.0352	.0504

TABLE 19: Autocorrelation values of various error residual and other random sequences in Experiment 7

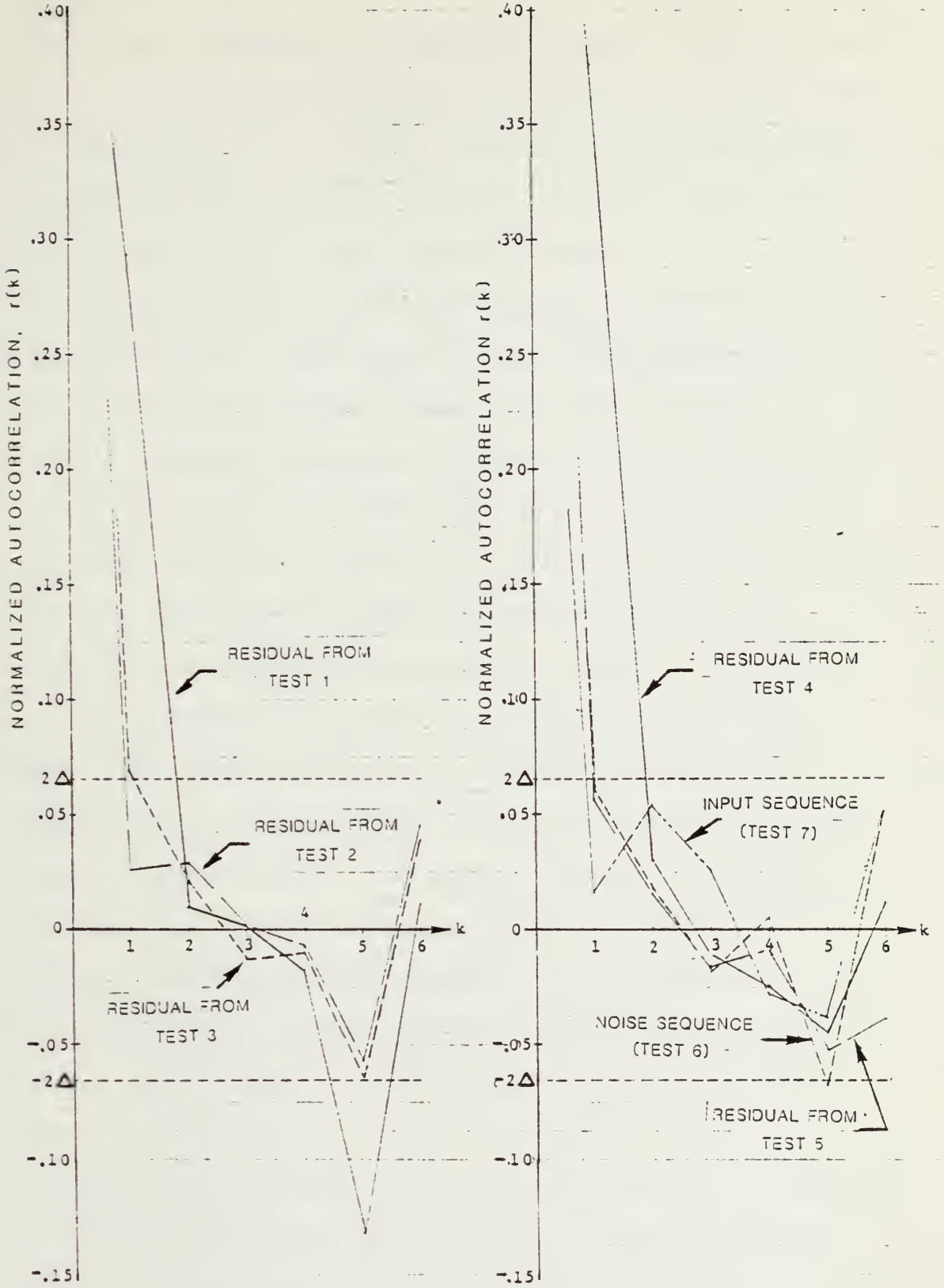


FIGURE 25: Normalized autocorrelation plots for various sequences in Experiment 7

Note the differences between the results of Test 1 and Test 3 in Table 19 and Figure 25. The residual signal from Test 1 has value $r(1) = .2928$, indicating that this signal is nonrandom. The residual signal from Test 3 has value $r(1) = .0258$, and all other autocorrelation values are between -2Δ and $+2\Delta$, indicating that this signal is reasonably random. This leads to the conjecture that the model of Test 3 is "better" than the model of Test 1, because the residual sequence from Test 3 is more random.

This conjecture is further supported by comparing the autocorrelation data from Test 4 and Test 5; the exact model fit cases. The residual signal from Test 4 has value $r(1) = .3756$ which indicates that the signal is nonrandom. The residual signal from Test 5 is significantly more random. The sample autocorrelation data and plots corresponding to the additive noise sequence (Test 6), and the input sequence (Test 7), provide examples of how the autocorrelation values should appear for typical random sequences. While some differences can be recognized in the plots of Figure 25, we would like to have another criterion that could more clearly indicate which sequence is more random.

We developed a measure for the randomness of a sequence based on the cumulative distribution of runs of varying lengths. A run of length k is defined as a contiguous sequence of k data points with the same sign, bordered by

data points with the opposite sign [Ref. 43]. As an example, consider the following sequence.

$\{s(n); 1<=n<=11\} = \{-.2, -.5, +.3, -.1, +.1, +.3, +.05, -.1, -.2, -.05, -.04\}$

This has 2 runs of length 1, 1 run of length 2, 1 run of length 3, and 1 run of length 4.

We define a factor for the "randomness" of a sequence $\{s(n)\}$, to be the percentage of runs with length less than or equal a small integer k.

$$\rho(k) = \frac{\sum_{j=1}^{k} \text{Number of runs of length } j \text{ in the sequence}}{\text{Total number of runs in the sequence}} \qquad \{7.22\}$$

A random sequence should primarily have runs of low size. Therefore $\rho(k)$ should increase rapidly for small values of k. Table 20 contains the values of $\rho(k)$ versus k for the five error residual sequences from Test 1 through Test 5. We also include the values of $\rho(k)$ versus k for both the random additive noise sequence $\{v(n)\}$ and the input sequence $\{u(n)\}$, as a comparison basis. These are denoted as Test 6 and Test 7, respectively. This calculated data is graphically presented in Figure 26. A split format presentation is used in Figure 26 to better distinguish the different distribution of runs plots. The abbreviation SIGA is used for the Search Indicator Growth Algorithm where space is limited.

TEST DESCRIPTION	CUMULATIVE DISTRIBUTION OF RUNS $\rho(1)$	$\rho(2)$	$\rho(3)$	$\rho(4)$	$\rho(5)$
1: SIGA GROWTH OF A RECURSIVE BVM USING NOISY SYSTEM DATA.	.412	.619	.779	.860	.914
2: SIGA GROWTH OF NON-RECURSIVE VOL MODEL USING NOISY SYSTEM DATA.	.475	.733	.857	.925	.951
3: SIGA GROWTH OF A RECURSIVE BVM USING OUTPUT DATA FROM VOL MODEL OF TEST 2.	.474	.711	.838	.904	.954
4: DIRECT EVALUATION OF EXACT MODEL USING NOISY SYSTEM DATA.	.363	.605	.768	.856	.912
5: DIRECT EVALUATION OF EXACT MODEL USING OUTPUT DATA FROM VOL MODEL OF TEST 2.	.463	.715	.846	.913	.959
6: ADDITIVE NOISE SEQUENCE	.483	.733	.866	.928	.969
7: INPUT SEQUENCE	.520	.745	.869	.930	.965

TABLE 20: Cumulative distribution of runs of varying length for the error residuals and other sequences in Experiment 7

Test 1: SIGA growth with recursive BVM and noisy system data
Test 2: SIGA growth with nonrecursive VOL and noisy system data
Test 3: SIGA growth with recursive BVM and output data from VOL model of Test 2
Test 4: Direct evaluation of exact model using noisy system data
Test 5: Direct evaluation of exact model using output data from VOL model of Test 2
Test 6: Additive noise sequence
Test 7: Input Sequence

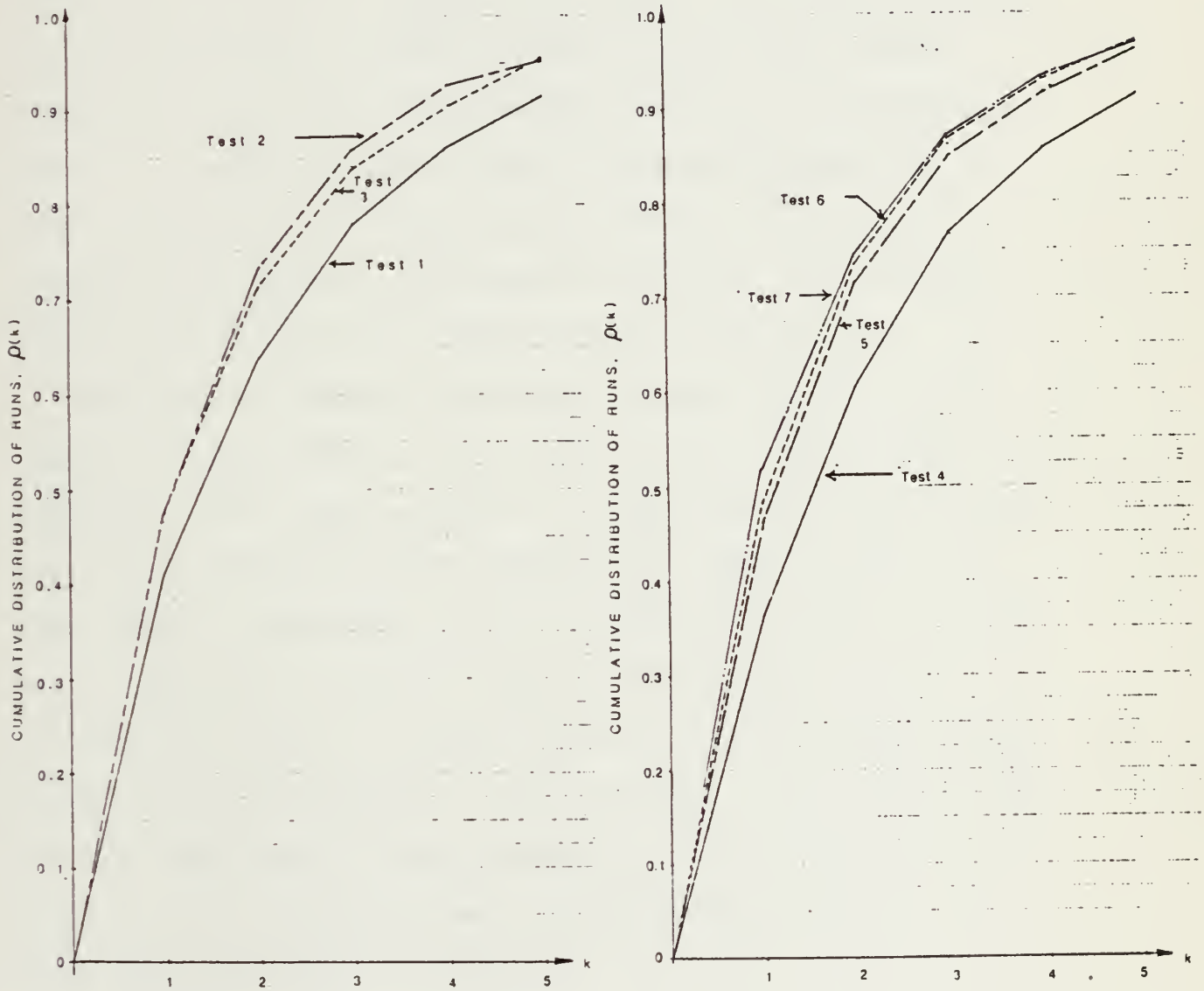


FIGURE 26: Plots of cumulative distribution of runs of varying length for the error residuals and other sequences in Experiment 7

Figure 26 demonstrates the power of the randomness factor $\rho(k)$. The curves corresponding to Test 6 and Test 7 represent the randomness of the most random sequences in this experiment, the additive output noise and the input probe. By comparing the cumulative distribution of runs curves for different residual sequences, we conjecture that the curve closest to that of Test 6 is the most random sequence. Except for pathological cases (e.g. no additive noise), it follows that no error residual sequence can be more random than our uncorrelated additive output noise. The plots of figures like Figure 26 provide an alternate means of evaluating the randomness of sequences.

Analysis of Figure 26 provides a clear picture of the results of Experiment 7. We see that the model of Test 3 is superior to the model of Test 1. Additionally, if we could improve our model growth technique and somehow obtain the exact form of the model, the N-R technique would provide us with the model of Test 5. This is significantly superior to the model of Test 4, the best we could hope to obtain using the existing techniques in the literature. We conclude that the N-R technique and the cumulative distribution plots can improve our systems characterization methods when we are faced with the Case 1A situation.

C. REAL WORLD EXPERIMENTS

This section presents the results of an experiment using real world data sequences. Unlike the controlled

experiments of the previous section, the results are not as dramatic. The actual form of the system equation is unknown, as are the specific properties of the input sequence and any measurement noise.

Experiment 8

The New London, Connecticut, Laboratory of the Naval Underwater Systems Center has been engaged in a continuing series of research efforts aimed at accurately modeling a particular path in the ocean. One set of experiments involved injecting a signal into a transmitting hydrophone, and measuring the resulting signal at a distant receiving hydrophone. Three sets of these input and output signals were sampled, converted to digital format, loaded into computer files, and made available for experimentation[12]. We denote the different input sequences of length 1024 as CH1IN, CH2IN, and CH3IN. The corresponding output sequences are denoted as CH1OUT, CH2OUT, and CH3OUT.

The sequences were measured over a suitably short time interval, and we therefore consider the acoustic path to be time invariant during the period of the measurements. It is expected that ambient noise and signals from other sources are received at the receiving hydrophone. We are

12 These computer files were made available by Mr. Steve Capizzano of NUSC on 12 October 1981. No details were available regarding any potential model form or the characteristics of the input or noise sequences.

therefore faced with a difficult real world example of the Case 1A conditions described in Chapter VI.

We first calculate the sample autocorrelation values for the three input sequences. These values are summarized in Table 21, and graphically presented in Figure 27.

	NORMALIZED AUTOCORRELATION VALUES						
INPUT SEQUENCE	r(1)	r(2)	r(3)	r(4)	r(5)	r(6)	r(7)
1: CH1IN	.0313	-.3069	.2131	-.1944	-.2422	.1959	.1260
2: CH2IN	-.1022	-.4193	.3630	-.1216	-.3309	.1695	.1222
3: CH3IN	-.0040	-.3026	.1913	-.2758	-.3072	.1630	.1366

TABLE 21: Normalized autocorrelation values for various input sequences in Experiment 8

We also calculate the cumulative distribution of runs values for these input sequences. These values are listed in Table 22, and graphically presented in Figure 28.

	CUMULATIVE DISTRIBUTION OF RUNS					
INPUT SEQUENCE	$\rho(1)$	$\rho(2)$	$\rho(3)$	$\rho(4)$	$\rho(5)$	$\rho(6)$
1: CH1IN	.377	.781	.876	.942	.986	.992
2: CH2IN	.375	.844	.912	.961	.980	.994
3: CH3IN	.395	.781	.877	.943	.984	.996

TABLE 22: Cumulative distribution of runs values for various input sequences in Experiment 8

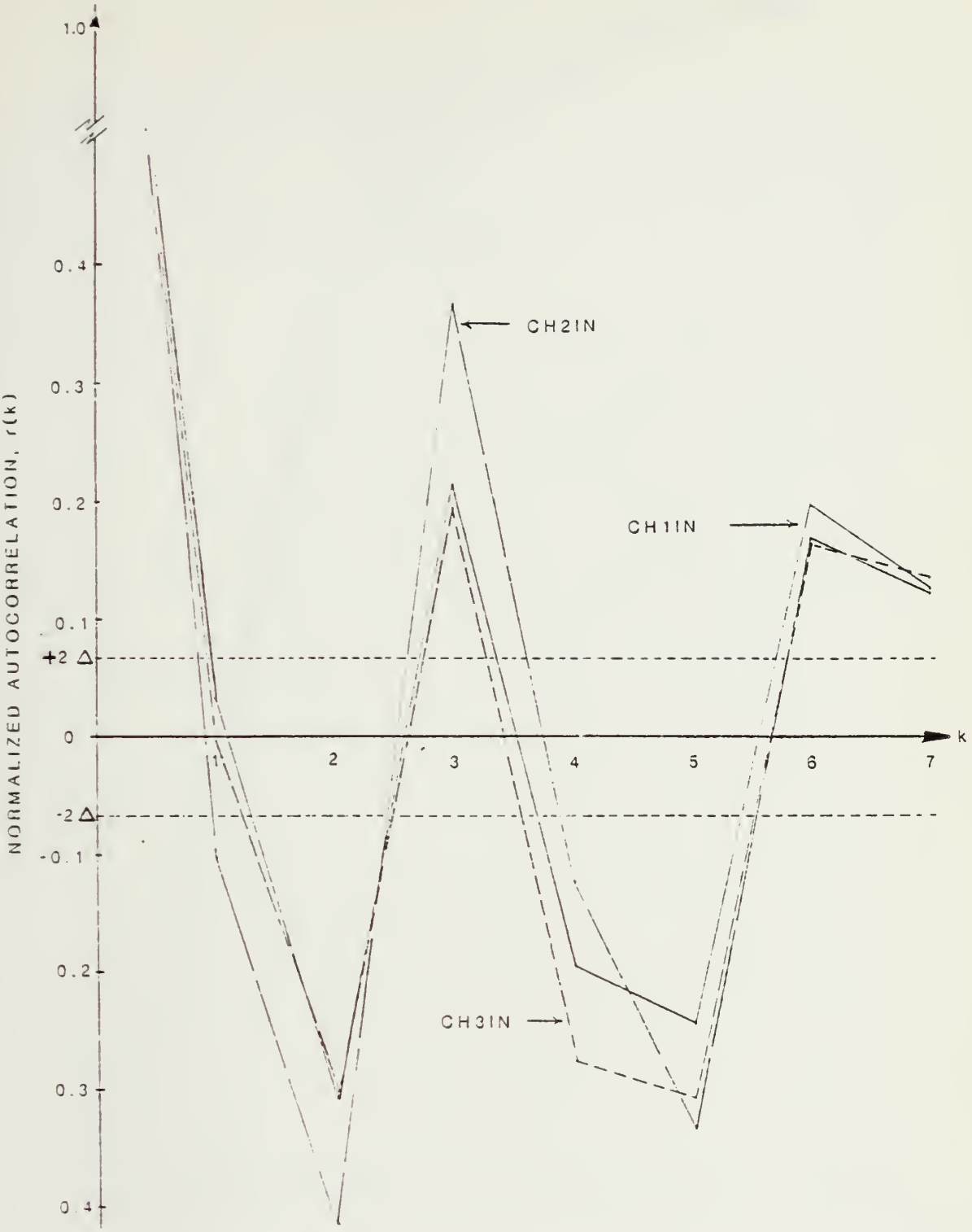


FIGURE 27: Normalized Autocorrelation plots for various input signals in Experiment 8

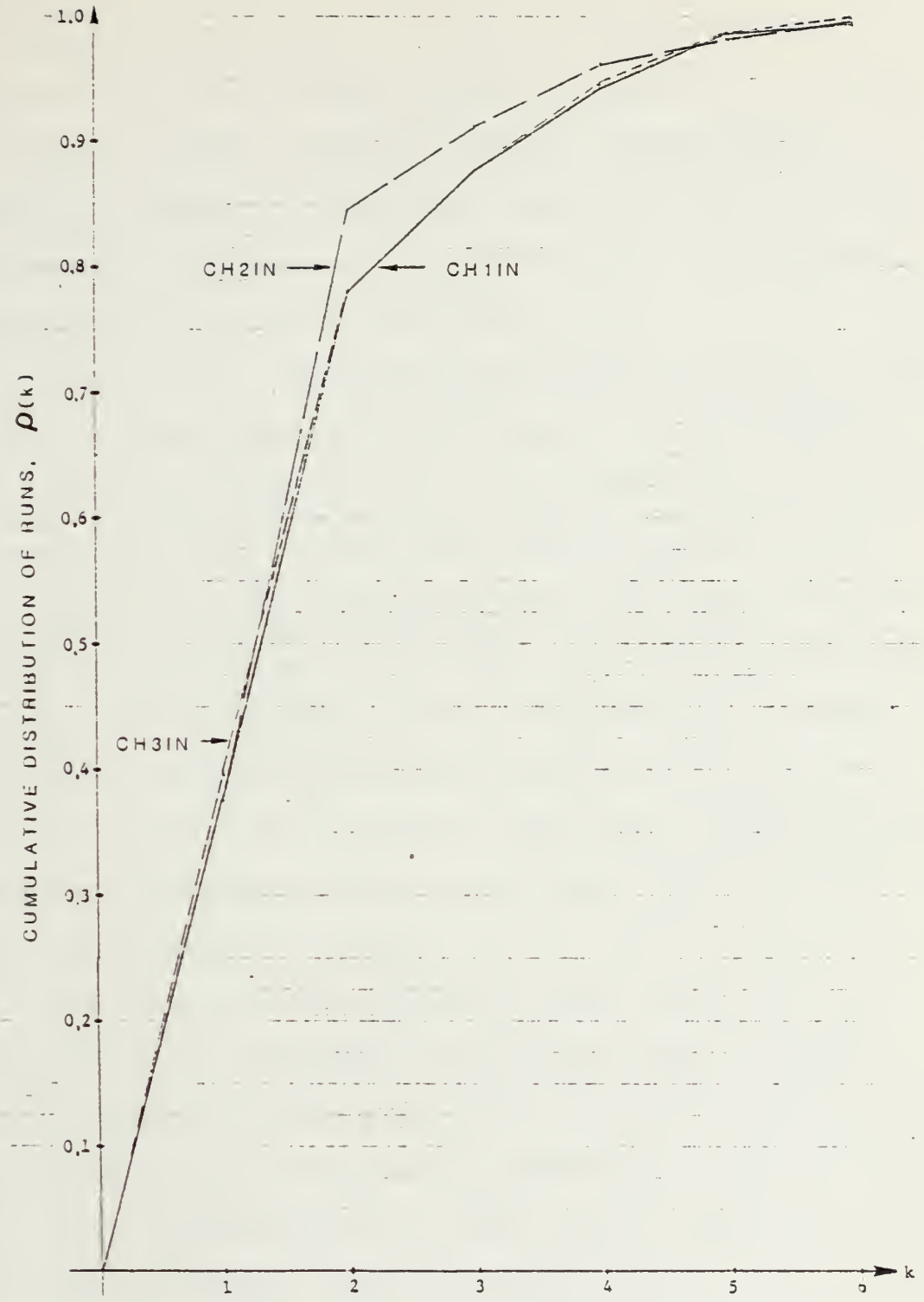


FIGURE 28: Plot of cumulative distribution of runs for various input sequences in Experiment 8

The form of the curves in Figure 27 and Figure 28 indicate that some problems should be expected. The input sequences are not as random as those of Experiments 1 through 7. Chapter VI mentioned that this condition may give our growth techniques some difficulty. We digress momentarily to expand on this point.

In our computer simulated experimental research of model growth, an input sequence {u(n)} based on an independent random generator distributed over the amplitude range of interest was used. In many real world problems, we are given input and output sequences that are simply time-series values of the available continuous time signals. Often the available input sequence may not be sufficiently random, and significant autocorrelations may result in the error residual, even with an adequate model form. Additive output noise contributing to the error residual may also result in significant autocorrelations.

Least squares model evaluation does not require any assumptions such as independence of the error residual values, but our candidate term selection and evaluation techniques can give degraded or misleading results in this case. The "goodness-of-fit" tests used in Experiment 7 may not produce suitable results in these cases.

This experiment is continued in an attempt to gain further insight into this common situation, but the expectations are limited for a successful characterization.

Based on the preceding figures, the first 500 data points of CH3IN and CH3OUT are selected as the input and output sequences for the characterization experiment. We first grow a recursive linear ARMA model by the M-Directed block form growth technique (with d=1). This is included for comparison since it is equivalent to the Box and Jenkins technique commonly used in the literature. This first modeling example is denoted as Test 1 and the results are presented in Table 23.

Chapter VI mentioned that it was possible to evaluate models by regression analysis. This involves considering a large set of candidate model terms, and evaluating the exact reduction in the fitting error that results if the best performing term is brought into the model, one at a time. This is similar to picking the one candidate model term with the largest value of search indicator I(j,8) given by Eq. {6.17}. Recall from Tables 5 and 6 that the cost of computing I(j,8) is much higher than the cost for I(j,12). The results of a regression analysis of the experimental data using the candidate model term set defined by a BVM(1,9) are included for comparison. This is denoted as Test 2, and results are presented in Table 24.

We next use the system data sequences to grow a recursive linear BVM using the Search Indicator Growth Algorithm. This is denoted as Test 3, and results are presented in Table 25. The model from Test 3 performs

better at each growth iteration than the model from Test 1. The results of Test 3 are almost identical to the regression analysis results of Test 2 at each growth iteration. It can be shown from the developments in Chapter VI, that the Search Indicator Growth Algorithm requires significantly fewer computations than regression analysis.

For Test 4, we use the system data sequences and grow a more general recursive nonlinear BVM using the Search Indicator Growth Algorithm. This enables us to see if a nonlinear model would provide a better fit than the previously analyzed linear ARMA form. The results are presented in Table 26.

Using the VOL model form of Eq. {2.4} and the Search Indicator Growth Algorithm, we next grow a nonrecursive nonlinear model (first phase of the N-R technique). We give the technique freedom to consider all terms in the VOL(2,9) model. This is denoted as Test 5, and results are given in Table 27.

After evaluating the coefficients of the final VOL model from Test 5, this model is synthesized on the computer, and probed with the stored version of the input sequence {u(n)}. The resulting model output sequence {$\hat{y}$(n)} is stored, and used with {u(n)} to grow a recursive model of the BVM form using the Search Indicator Growth Algorithm (second phase of the N-R technique). This model growth is denoted as Test 6, and the results are presented in Table 28.

1	CANDIDATE MODEL	NUMBER OF MODEL TERMS			COND. NUMBER	FITTING ERROR	RESIDUAL SEQUENCE		NORMALIZED AUTOCORRELATION OF RESIDUAL SEQUENCE							CUMULATIVE DISTRIBUTION OF RUNS OF THE RESIDUAL				
		NEW	SEL.	TOT.			MIN.	MAX.	r(1)	r(2)	r(3)	r(4)	r(5)	r(6)	r(7)	ρ(1)	ρ(2)	ρ(3)	ρ(4)	ρ(5)
1	BVM(1,1)	3	3	3	8.2	.57427	-1.501	1.436	-.0336	-.4234	.0199	-.1768	-.0600	.0794	.1022	.336	.770	.918	.975	.992
2	BVM(1,2)	2	2	5	12.4	.51918	-1.321	1.276	-.0077	-.1803	-.0767	-.4335	-.0173	.0952	.1407	.443	.779	.898	.980	.996
3	BVM(1,3)	2	2	7	18.8	.51915	-1.321	1.279	-.0076	-.1806	-.0714	-.4344	-.0216	.0963	.1401	.431	.760	.878	.963	.980
4	BVM(1,4)	2	2	9	35.5	.46986	-1.148	1.474	-.0616	-.1502	-.0595	-.1440	-.0584	-.1989	.2003	.477	.775	.912	.969	.996
5	BVM(1,5)	2	2	11	41.0	.46170	-1.076	1.378	-.0631	-.1766	-.0413	-.1461	-.0204	-.2233	-.1171	.497	.804	.946	.982	.996
6	BVM(1,6)	2	2	13	53.7	.42684	-1.096	1.248	-.0519	-.0963	-.0919	-.1327	-.0349	-.0449	.0306	.483	.786	.920	.962	.992
7	BVM(1,7)	2	2	15	63.8	.42121	-1.119	1.191	-.0285	-.0912	-.0950	-.1590	-.0249	-.0527	.0633	.469	.781	.906	.953	.980
8	BVM(1,8)	2	2	17	76.2	.41303	-1.112	1.148	-.0212	-.0696	-.0850	-.1169	-.0577	-.0254	.0602	.472	.756	.892	.952	.972
9	BVM(1,9)	2	2	19	88.3	.40925	-1.046	1.094	-.0293	-.0610	-.0585	-.1085	-.0477	-.0386	.0536	.476	.752	.906	.945	.972

TABLE 23: Summary Results of Test 1 (M Directed Growth) for Experiment 8

1	CANDIDATE MODEL	NUMBER OF MODEL TERMS			COND. NUMBER	FITTING ERROR	RESIDUAL SEQUENCE		NORMALIZED AUTOCORRELATION OF RESIDUAL SEQUENCE							CUMULATIVE DISTRIBUTION OF RUNS OF THE RESIDUAL				
		NEW	SEL.	TOT.			MIN.	MAX.	r(1)	r(2)	r(3)	r(4)	r(5)	r(6)	r(7)	ρ(1)	ρ(2)	ρ(3)	ρ(4)	ρ(5)
1	BVM(1,9)	19	1	1	1.0	.57548	-1.500	1.449	-.0782	-.4227	.0689	-.1691	-.0580	-.0764	-.0914	.373	.786	.929	.976	.992
2	BVM(1,9)	18	1	2	7.8	.52198	-1.341	1.311	-.1099	-.1797	.0306	-.4200	-.0033	.0905	.1053	.469	.796	.904	.965	.985
3	BVM(1,9)	17	1	3	11.2	.47377	-1.210	1.385	-.1655	-.1317	-.0026	-.1179	.0200	-.1926	.1680	.507	.813	.939	.982	.997
4	BVM(1,9)	16	1	4	12.0	.45309	-.9804	1.392	-.2109	-.0262	-.0095	-.0480	.0159	-.0116	.1838	.529	.805	.919	.960	.985
5	BVM(1,9)	15	1	5	12.2	.44934	-1.027	1.389	-.1130	-.0329	-.0872	-.0582	.0048	.0088	.2026	.519	.788	.905	.951	.981
6	BVM(1,9)	14	1	6	12.3	.44620	-1.028	1.346	-.1017	-.0399	-.1048	-.0392	.0977	.0058	.1517	.519	.774	.901	.950	.984
7	BVM(1,9)	13	1	7	12.7	.43992	-.9491	1.339	-.0193	-.0420	-.0084	-.0401	.0416	.0323	.1060	.496	.754	.879	.931	.967
8	BVM(1,9)	12	1	8	18.7	.43750	-1.012	1.337	-.0275	-.0470	-.0233	-.0541	.0289	.0194	.0906	.489	.760	.898	.953	.980
9	BVM(1,9)	11	1	9	34.3	.43455	-.9935	1.338	-.0450	-.0558	-.0459	-.0764	.0038	-.0045	.0600	.488	.750	.902	.957	.988
10	BVM(1,9)	10	1	10	46.3	.43227	-1.042	1.282	-.0616	-.0626	-.0680	-.0869	-.0136	-.0180	.0362	.477	.744	.902	.961	.988
11	BVM(1,9)	9	1	11	54.9	.42811	-1.082	1.245	-.0818	-.0686	-.0967	-.1001	-.0443	-.0318	.0095	.500	.760	.912	.969	.989
12	BVM(1,9)	8	1	12	62.0	.42527	-1.064	1.203	-.0656	-.0582	-.0804	-.1135	-.0177	-.0364	.0627	.504	.777	.913	.966	.985
13	BVM(1,9)	7	1	13	66.4	.42176	-.9989	1.150	-.0679	-.0302	-.0724	-.0609	-.0282	.0012	.0608	.533	.786	.919	.971	.989
14	BVM(1,9)	6	1	14	67.2	.42036	-1.024	1.154	-.0594	-.0376	-.0738	-.0694	-.0299	.0020	.0529	.530	.785	.915	.967	.989
15	BVM(1,9)	5	1	15	68.4	.41906	-1.017	1.138	-.0578	-.0447	-.0646	-.0784	-.0353	-.0035	.0423	.511	.773	.905	.962	.992
16	BVM(1,9)	4	1	16	69.8	.41649	-1.049	1.122	-.0433	-.0558	-.0740	-.0995	-.0445	-.0147	.0356	.472	.756	.902	.961	.980
17	BVM(1,9)	3	1	17	78.5	.41306	-1.086	1.107	-.0407	-.0612	-.0824	-.1057	-.0473	-.0227	.0376	.504	.771	.908	.962	.989
18	BVM(1,9)	2	1	18	85.2	.41000	-1.077	1.123	-.0270	-.0638	-.0788	-.1117	-.0658	-.0313	.0429	.504	.760	.907	.953	.976
19	BVM(1,9)	1	1	19	88.3	.40925	-1.046	1.094	-.0293	-.0610	-.0585	-.1085	-.0477	-.0386	.0536	.480	.756	.909	.949	.976

TABLE 24: Summary Results of Test 2 (Regression Analysis) for Experiment 8

i	CANDIDATE MODEL	NUMBER OF MODEL TERMS			COND. NUMBER	FITTING ERROR	RESIDUAL SEQUENCE		NORMALIZED AUTOCORRELATION OF RESIDUAL SEQUENCE							CUMULATIVE DISTRIBUTION OF RUNS OF THE RESIDUAL				
		NEW	SEL.	TOT.			MIN.	MAX.	r(1)	r(2)	r(3)	r(4)	r(5)	r(6)	r(7)	ρ(1)	ρ(2)	ρ(3)	ρ(4)	ρ(5)
1	BVM(1,9)	19	2	2	7.8	.52198	-1.341	1.311	-.1099	-.1797	.0306	-.4200	-.0033	.0905	.1053	.469	.796	.904	.965	.985
2	BVM(1,9)	17	2	3	11.2	.47377	-1.210	1.385	-.1655	-.1317	-.0026	-.1179	.0200	-.1926	.1680	.507	.813	.939	.982	.996
3	BVM(1,9)	16	1	4	12.0	.45309	-.9804	1.392	-.2097	-.0262	-.0095	-.0480	.0159	-.0116	.1838	.529	.805	.919	.960	.985
4	BVM(1,9)	15	1	5	12.2	.44934	-1.027	1.389	-.1130	-.0329	-.0872	-.0582	.0048	.0088	.2026	.519	.788	.905	.951	.981
5	BVM(1,9)	14	1	6	12.3	.44620	-1.028	1.346	-.1017	-.0399	-.1048	-.0392	.0977	.0058	.1517	.519	.775	.901	.950	.985
6	BVM(1,9)	13	1	7	12.7	.43992	-.9491	1.339	-.0193	-.0420	-.0084	-.0401	.0416	.0032	.1060	.496	.754	.879	.931	.968
7	BVM(1,9)	12	3	10	19.6	.43388	-1.063	1.337	-.0181	-.0544	-.0492	-.0709	-.0046	.0248	.0930	.484	.750	.893	.952	.980
8	BVM(1,9)	9	3	13	56.6	.42538	-1.082	1.243	-.0561	-.0683	-.1139	-.1156	-.0577	-.0234	.0169	.508	.769	.917	.962	.989
9	BVM(1,9)	6	1	14	60.5	.42350	-1.082	1.214	-.0738	-.0494	-.1079	-.0764	-.0689	.0026	.0018	.522	.781	.922	.970	.989
10	BVM(1,9)	5	2	16	81.3	.41898	-1.022	1.137	-.0595	-.0439	-.0578	-.0766	-.0286	-.0050	.0460	.515	.771	.901	.954	.989
11	BVM(1,9)	3	2	18	82.3	.41325	-1.018	1.108	-.0291	-.0592	-.0640	-.1057	-.0510	-.0283	.0456	.500	.773	.904	.958	.981

TABLE 25: Summary Results of Test 3 (Linear Model SIGA Growth) for Experiment 8

i	CANDIDATE MODEL	NUMBER OF MODEL TERMS			COND. NUMBER	FITTING ERROR	RESIDUAL SEQUENCE		NORMALIZED AUTOCORRELATION OF RESIDUAL SEQUENCE							CUMULATIVE DISTRIBUTION OF RUNS OF THE RESIDUAL				
		NEW	SEL.	TOT.			MIN.	MAX.	r(1)	r(2)	r(3)	r(4)	r(5)	r(6)	r(7)	$\rho(1)$	$\rho(2)$	$\rho(3)$	$\rho(4)$	$\rho(5)$
1	BVM(1,9)	19	2	2	7.8	.52198	-1.341	1.311	-.1099	-.1797	.0306	-.4200	-.0033	.0905	.1053	.469	.796	.904	.965	.985
2	BVM(2,9)	207	1	3	11.2	.47368	-1.210	1.385	-.1655	-.1317	-.0026	-.1179	.0200	-.1926	.1680	.507	.813	.939	.982	.996
3	BVM(2,9)	206	1	4	12.0	.45309	-.9804	1.392	-.2097	-.0262	-.0095	-.0480	.0159	-.0116	.1838	.529	.805	.919	.960	.985
4	BVM(2,9)	205	2	6	12.5	.43822	-1.042	1.304	-.2731	-.0563	-.0810	-.0916	-.0151	-.0573	.1345	.528	.836	.948	.983	.993
5	BVM(2,9)	203	3	9	12.8	.42667	-1.088	1.322	-.1571	-.0904	-.1962	-.0982	.0836	-.0508	.0872	.507	.783	.934	.982	.993
6	BVM(2,9)	200	1	10	13.1	.41618	-.9841	1.309	-.0611	-.1154	-.1084	-.1372	-.0259	-.0495	-.0001	.500	.799	.938	.982	.993
7	BVM(2,9)	199	3	13	14.1	.40519	-.9529	1.196	-.0828	-.0913	-.1102	-.1028	-.0580	-.0230	-.0211	.522	.793	.926	.967	.981
8	BVM(2,9)	196	4	17	49.8	.39585	-.8851	1.096	-.0377	-.0856	-.0843	-.1091	-.0446	-.0135	.0287	.480	.760	.898	.957	.980
9	BVM(2,9)	192	6	23	307.4	.38570	-.9146	1.122	-.0231	-.0715	-.0717	-.1291	-.0270	-.0253	.0226	.503	.779	.904	.962	.985
10	BVM(2,9)	186	6	29	574.4	.38071	-.8970	1.120	-.0049	-.0640	-.0683	-.1289	-.0434	-.0261	.0145	.484	.774	.885	.952	.972

TABLE 26: Summary Results of Test 4 (Nonlinear BVM SIGA Growth) for Experiment 8

i	CANDIDATE MODEL	NUMBER OF MODEL TERMS			COND. NUMBER	FITTING ERROR	RESIDUAL SEQUENCE		NORMALIZED AUTOCORRELATION OF RESIDUAL SEQUENCE							CUMULATIVE DISTRIBUTION OF RUNS OF THE RESIDUAL				
		NEW	SEL.	TOT.			MIN.	MAX.	r(1)	r(2)	r(3)	r(4)	r(5)	r(6)	r(7)	ρ(1)	ρ(2)	ρ(3)	ρ(4)	ρ(5)
1	VOL(1,9)	10	1	1	1.0	.57548	-1.500	1.449	-.0782	-.4227	.0688	-.1691	-.0580	.0764	.0914	.373	.786	.929	.976	.992
2	VOL(2,9)	64	5	6	40.4	.56844	-1.622	1.427	-.0770	-.4210	.0559	-.1663	-.0600	.0790	.0871	.374	.783	.933	.984	.996
3	VOL(2,9)	59	5	11	51.8	.55924	-1.575	1.449	-.0832	-.3979	.0670	-.1570	.0582	.0610	.0869	.392	.800	.935	.992	.996
4	VOL(2,9)	54	7	18	127.9	.55379	-1.413	1.501	-.0756	-.4055	.0597	-.1481	-.0490	.0470	.0753	.379	.789	.934	.988	.996
5	VOL(2,9)	47	1	19	128.5	.55165	-1.517	1.498	-.0727	-.4053	.0607	-.1449	-.0552	.0465	.0721	.384	.795	.938	.988	.996
6	VOL(2,9)	46	7	26	168.0	.54692	-1.457	1.499	-.0742	-.4162	-.0600	-.1424	-.0538	.0441	.0735	.367	.801	.941	.984	.992
7	VOL(2,9)	39	7	33	328.3	.54478	-1.511	1.472	-.0742	-.4125	.0603	-.1482	-.0540	.0506	.0714	.383	.785	.938	.988	.992

TABLE 27: Summary Results of Test 5 (Nonlinear VOL Model SIGA Growth; First Phase of N-R Technique) for Experiment 8

1	CANDIDATE MODEL	NUMBER OF MODEL TERMS			COND. NUMBER	FITTING ERROR	RESIDUAL SEQUENCE		NORMALIZED AUTOCORRELATION OF RESIDUAL SEQUENCE							CUMULATIVE DISTRIBUTION OF RUNS OF THE RESIDUAL				
		NEW	SEL.	TOT.			MIN.	MAX.	r(1)	r(2)	r(3)	r(4)	r(5)	r(6)	r(7)	ρ(1)	ρ(2)	ρ(3)	ρ(4)	ρ(5)
1	BVM(1,9)	19	2	2	1.3	.57254	-1.580	1.531	-.0768	-.4084	.0704	-.1796	-.0505	.0884	.0848	.361	.778	.933	.988	.996
2	BVM(2,9)	207	3	5	24.2	.55998	-1.589	1.646	-.0717	-.3962	.0671	-.1464	-.0692	.0423	.0802	.368	.788	.930	.976	.992
3	BVM(2,9)	204	6	11	50.4	.54503	-1.449	1.379	-.0684	-.3909	.0435	-.1231	-.0786	.0207	.0871	.366	.783	.933	.984	.992
4	BVM(2,9)	198	4	15	55.9	.53509	-1.467	1.257	-.0634	-.3846	.0315	-.1223	-.0896	.0115	.0902	.373	.778	.933	.984	.992
5	BVM(2,9)	194	3	18	59.4	.53077	-1.445	1.241	-.0620	-.3863	.0309	-.1228	.0966	.0164	.0983	.378	.787	.929	.980	.996
6	BVM(2,9)	191	5	23	65.9	.52239	-1.388	1.419	-.0641	-.3850	.0218	-.1114	-.0832	.0033	.1099	.395	.787	.942	.981	.996
7	BVM(2,9)	186	5	28	89.4	.51658	-1.328	1.342	-.0631	-.3805	.0237	-.1006	-.0760	-.0076	.1058	.388	.796	.946	.985	1.00
8	BVM(2,9)	181	4	32	101.9	.51204	-1.307	1.531	-.0626	-.3758	-.0249	-.1024	-.0753	-.0058	.1066	.389	.793	.926	.988	1.00
9	BVM(2,9)	177	6	38	113.8	.50744	-1.310	1.409	-.0638	-.3720	.0315	-.1050	-.0837	-.0079	.1121	.378	.768	.941	.984	1.00

TABLE 28: Summary Results of Test 6 (Nonlinesr BVM SIGA Growth from output of VOL Model from Test 5; Second Phase of N-R Technique) for Experiment 8

The preceding tables and figures support the previously stated concerns regarding the randomness of the input sequence {u(n)}. The results of Test 1 through Test 6 are all somewhat disappointing. Experience with many controlled experiments leads to the conclusion that the correlations within the input sequence are the main reason for these results in Experiment 8. It should be mentioned, however, that another possible contributing problem is that the form of the BVM might not be appropriate for the physical system we are attempting to represent.

The results obtained in this experiment are the reason we stated that systems characterization is a trial and error process (Chapter V). The choice of model form and the characteristics of the available (or hopefully controllable) input probe are extremely important. These ultimately must be selected by the user based on all available quantitative and non-quantitative factors.

Despite the high fitting error obtained in the various tests of Experiment 8, several results are imbedded within Tables 23 through 28. The least squares techniques are designed to minimize the fitting error $J^2(i)$ while growing the model. The performances of Test 1 through Test 4 are compared further. Figure 29 is a plot of J(i), the square root of the fitting error, versus the total number of model terms after each iteration.

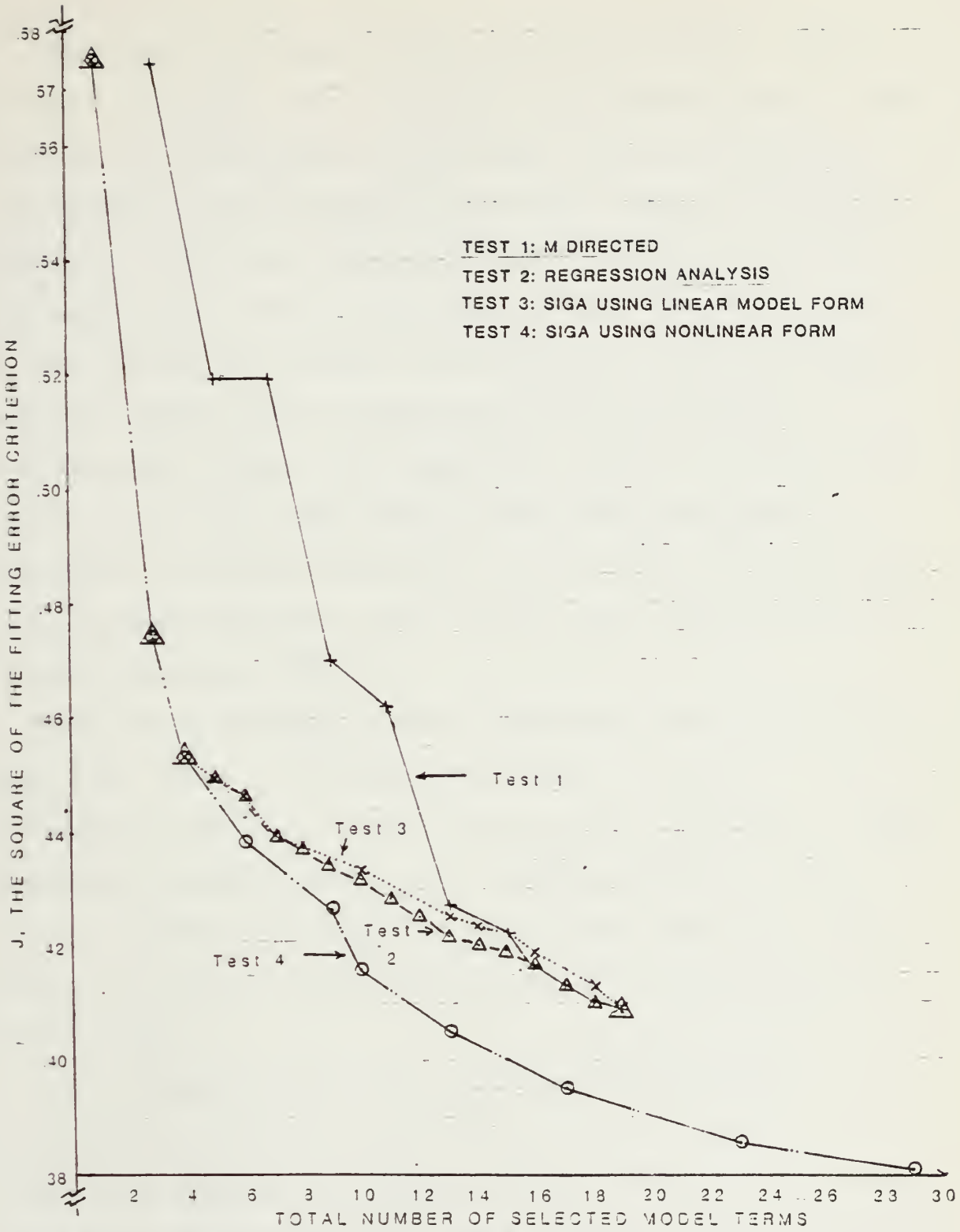


FIGURE 29: Plot of the square root of the fitting error versus the total number of model terms after each growth iteration, for various tests in Experiment 8

The results of Figure 29 are very interesting. The M Directed technique (Test 1) reduced the fitting error as the order of the ARMA model was increased, but the results are significantly poorer than the regression analysis technique (Test 2). The Search Indicator Growth Algorithm using the ARMA model form (Test 3) nearly duplicated the performance of the regression analysis. We previously showed that the Search Indicator Growth Algorithm offered substantial computational savings over regression analysis (Chapter VI). Figure 29 verifies that even for real world measurement sequences, the Search Indicator Growth Algorithm can perform systems characterization with results that are equivalent to the best existing technique.

The Search Indicator Growth Algorithm using the BVM(2,9) model form (Test 4) provided equivalent or better performance than the previous three growth techniques. This comparison includes models with the same number of total terms. Allowing the algorithm to consider nonlinear terms resulted in some of them being chosen over the candidate linear terms.

Even though we have been primarily interested in minimizing the fitting error $J^2(i)$, it is interesting to look at the maximum and minimum values in the error residual sequences resulting from each type of model growth. Figure 30 graphically presents this information obtained from Table 23 through Table 26.

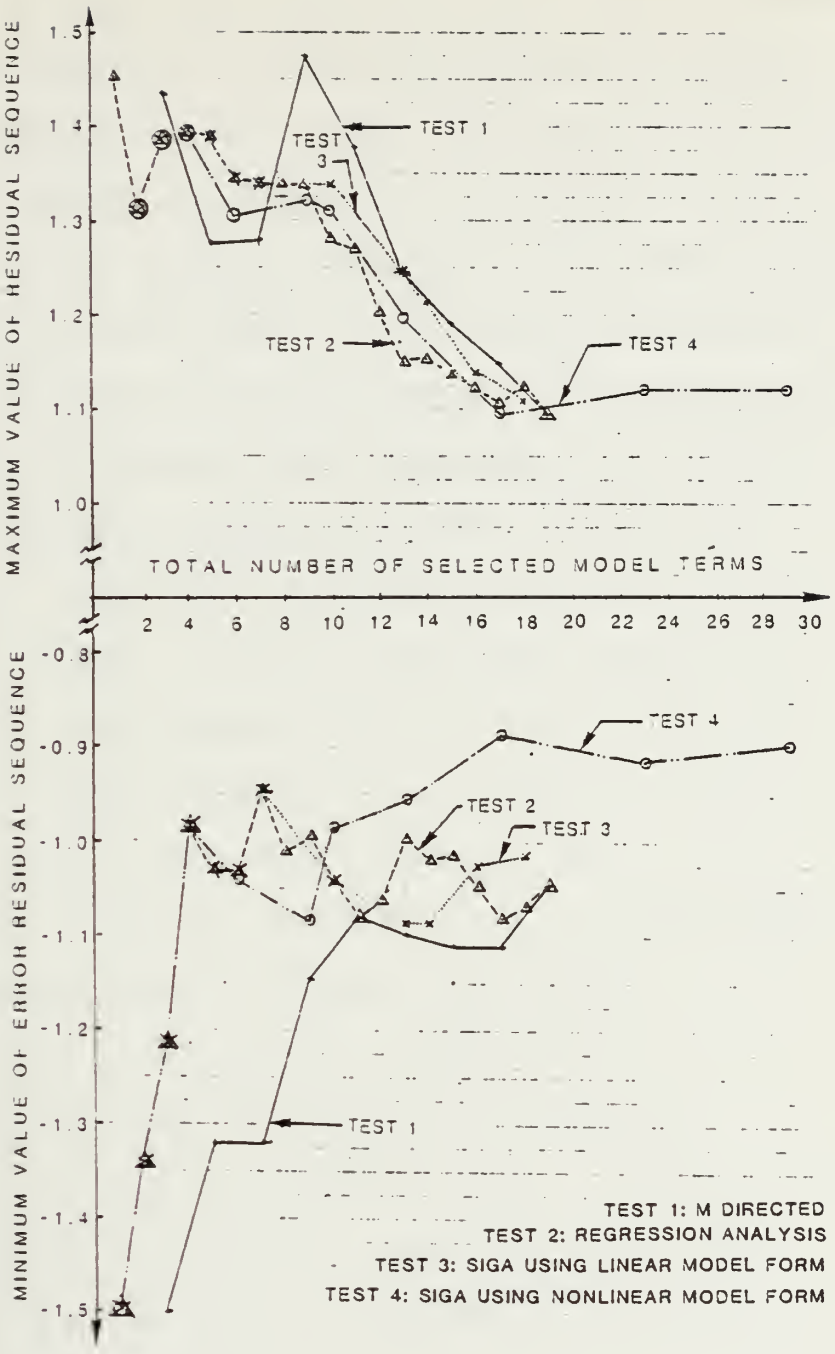


FIGURE 30: Plot of the maximum and minimum values of the error residual sequence versus the total number of model terms after each iteration, for various tests in Experiment 8

Figure 30 shows that there is a generally decreasing trend in the amplitude of both the maximum and minimum residual sequences as more model terms are added by the growth techniques. Note that the decreasing trend is more pronounced in the regression analysis case (Test 2) than the Box and Jenkins case (Test 1). The Search Indicator Growth Algorithm with ARMA model form (Test 3) performed reasonably close to Test 2, and the best performance was obtained when the Search Indicator Growth Algorithm was used with the BVM(2,9) form (Test 4). The model equation resulting from Iteration 8 of Test 4 provided the best combination of low fitting error, good autocorrelation properties of the residual, and good cumulative distribution of runs of the residual. This 17 term model equation is;

$$
\begin{aligned}
y(n) = {} & .9345E{-}1\ u(n-5) - .9122E{+}0\ y(n-2) - .8470E{+}0\ y(n-4) \\
& - .5365E{+}0\ y(n-6) + .1064E{+}0\ y(n-4)y(n-4) \\
& + .2415E{+}0\ y(n-9)y(n-9) - .3279E{+}0\ u(n-1) \\
& - .3585E{+}0\ u(n-5) + .1007E{+}0\ y(n-2)y(n-9) \\
& - .3653E{+}0\ u(n-3) - .1813E{+}0\ y(n-8) - .1254E{+}0\ y(n-7) \\
& + .1636E{+}0\ y(n-1)y(n-4) - .2062E{+}0\ y(n-5)y(n-7) \\
& - .3198E{+}0\ u(n-1)y(n-7) - .3023E{+}0\ u(n-5)y(n-6) \\
& + .2965E{+}0\ u(n-6)y(n-1) \qquad \{7.23\}
\end{aligned}
$$

Tables 27 and 28 indicate that the N-R technique failed to improve the characterization. Several reasons could exist for this result. The previously discussed problems with the characteristics of the input signal could be a

major factor. The normalized autocorrelation values and cumulative distribution of runs values in these tables indicate that the residuals are probably not random enough. If additive output noise exists in {y(n)}, it may be correlated with itself or with the input {u(n)}. The only conclusion we can reach on this point is that in this experiment, the N-R technique did not offer any advantages to the characterization problem.

E. SUMMARY OF EXPERIMENTAL RESULTS

The results of these experiments show that block-form recursive modeling of nonlinear systems typically produces (1) non-parsimonious models (e.g. contain unneeded terms with significantly non-zero coefficient values), (2) higher fitting error J, (3) higher condition number for the least squares matrix (and therefore larger variance in the coefficient estimates), and (4) distorted coefficient estimates on the correct model terms. The block-form techniques also require the availability of a larger number of data measurements, have a much larger computational cost, and often fail to converge on an adequate model because of excessive ill-conditioning of the least squares matrix or the limited amount of available data.

The block-form growth techniques make no provisions for handling near colinearity in the candidate model set. This typically results in abnormally high condition number for the least squares matrix (and therefore larger variance in

the coefficient estimates). Experiment 2 showed the typical results of this weakness.

The block-form techniques force a restrictive set of terms to be fully considered at each growth iteration. This typically results in a significant number of unnecessary terms in the model equation; increasing the computational cost and contributing to other problems. Experiments 2, 3, and 4 are examples of this situation.

The block-form techniques require the availability of a larger number of data measurements than the Search Indicator Growth Algorithm. Therefore, with limited data, there are many cases where we will be unable to grow an adequate model for an application using block-form techniques. Experiment 5 provided a meaningful example of this situation.

The Search Indicator Growth Algorithm can better select its starting base model by using I(j,12) at iteration 1. In this way it can recognize the existence of a delay factor L in the system, and can start the growth iteration at the appropriate term (Experiment 3).

Finally, the form of the Search Indicator Growth Algorithm allows for simple extensions that can be used to better handle real world conditions like additive output noise. The averaging algorithm discussed in Chapter VI offers some improvement when conditions permit its use (Experiment 6). The two-stage "N-R" algorithm proposed in Chapter VI has some capabilities for improving system

characterization when we cannot probe the system (Experiment 7). A real-world example (Experiment 8) shows that the Search Indicator Growth Algorithm can characterize a system with results that are equivalent to or better than existing techniques. The inability to control the input probe was shown to degrade all of these techniques.

Based on experience with many characterization experiments, an important factor appears to be the selected amplitude range for the input probe signal. If this range is too small, the resultant signals specified by the candidate model terms typically are highly colinear. This increases the ill conditioning of the least squares matrix and degrades the performance of the growth algorithm.

If the input probe is selected to be too large, the system output may be unbounded. It is not possible or meaningful to continue the characterization experiment in this case. Experimentation may be necessary to obtain a suitable input probe.

The Search Indicator based growth techniques have been shown to be superior to block form techniques, but further work still remains to be done. The whole systems characterization problem is not yet solved.

VIII. APPLICATIONS, CONCLUSIONS, AND AREAS FOR FURTHER RESEARCH

A. DISCUSSION OF APPLICATIONS

A main reason for experimentally modeling a system is to better understand the nature of what is actually happening in the system. Another reason for modeling is to use models in designing controllers or estimators, and for simulating systems to predict behavior. The model can serve to confirm existing beliefs about functional relationships. A stronger concept is that the use of model growing techniques could lead to the prediction of a physically significant effect of which the application user might be unaware. If new information regarding the system can be uncovered, we may be better able to understand the inner workings of the system.

Three current applications for accurate experimentally determined models are; (1) fault detection, (2) fault evaluation, and (3) reduced-order modeling. We discuss each of these in terms of the techniques of this research, and describe some new capabilities that appear to be useful.

Once a model with acceptable performance for a particular application has been obtained from the set of measurement data, we cannot be sure that there are no other equivalently performing models with different sets of model terms and coefficient values. Such equivalent models may have been uncovered during the model growth iterations. If

we somehow obtained equivalent models with different set of terms, we must have a means of picking the "best".

Chapter I mentioned that the criterion for best model is application dependent. In terms of performance modeling, the following criteria appear best. For fault detection applications, the optimum criterion is maximum sensitivity of the error residual to changes in each system parameter. For fault evaluation applications, the optimum criterion is maximum distinguishability of the system characteristic that has changed. For reduced-order modeling applications, the optimum criterion is the best performance of a finite term model in duplicating the behavior of the system.

One general criterion that appears to be a good compromise is based on Ockham's Razor, "... the simplest model is the best ...". We define the simplest BVM that adequately represents the system performance as the one with the smallest number of terms, and the lowest degree and memory (when the number of terms are the same). This criterion is probably not optimum for all applications.

Assuming that we have reduced our set of equivalent models to one "best" model, there still are two main concerns. First, we would like to know if any simpler equivalently performing model exists. It is conceivable that particular systems might be better modeled by a different functional form than BVM, but we will not consider this case. If a model is obtained by one of the block-form

techniques of Chapter V, the experiments of Chapter VII demonstrate that the model may contain a large number of extra terms. If the models were obtained by the Search Indicator Growth Algorithm, unneeded model terms may have been included at various iterations. We want the most parsimonious (minimum number of term) model that matches the performance of the system within acceptable error.

The second concern is how to efficiently use the model in applications. If the model is simulated and used as shown in Figure 1 of Chapter I, a running average of the squared difference between the system and model output can be monitored. When this average exceeds a threshold, a fault may have occurred in the system. Other factors that might also cause this condition include; (1) increased additive measurement noise, and (2) the current input signal exceeding the amplitude range used in this model's growth.

Once a fault has been detected, the coefficient values can be re-estimated and used as an indication of the possible kind of fault. This last step is the basis of the fault evaluation application. The work that follows is designed to improve the efficiency and accuracy of the approach to both of these modeling concerns.

A possible concept is to select the subset of model terms whose coefficients are most robust to variations in conditions that are unrelated to system faults. These coefficients are designated as the "syndromes" of the model, and the following method is proposed for their selection.

Step 1: Given the best performance fitting model, estimate the coefficient values $\underline{p}(i)$ that minimize the error criterion $J^2(i)$ for different random input probes, each with approximately the same amplitude distribution. Mark those coefficients whose estimates remain nearly constant from test to test as having the property "A".

Step 2: Using the same model, estimate the coefficient values $\underline{p}(i)$ that minimize the error criterion $J^2(i)$ for different ranges of input amplitude (continue to use a uniform amplitude distribution). Limit this range of input amplitude to the known or assumed range of the actual operating input of the system. Mark those coefficients whose estimates remain nearly constant from test to test as having the property "B".

Coefficients with both the "A" and "B" property are conjectured to be robust to variations in both the input probe level and the particular probe contents. These syndromes should therefore be most sensitive to changes in the system under test. It is logical to expect that the final model obtained by the performance modeling based Search Indicator Growth Algorithm of Chapter VI, would provide a superior fault detection signal threshold, and better fault evaluation syndromes, than the models obtained by the block-form growth algorithms of Chapter V.

All of the fault detection and evaluation methods in two well referenced survey reports [Ref. 44 and 45], and other recent papers [Ref. 46, 47 and 48], are based on using the full set of coefficients of the obtained model form. The preceding development suggests that the set of syndromes would provide a clearer reference for recognizing system faults than would the full set of model coefficients. It is also expected that these Search Indicator Growth Algorithm developed syndromes would be superior to the large set of Nonlinear ARMA lattice coefficients proposed for this application by Reference 5.

The reduced-order modeling problem has received considerable attention in the literature [Ref. 49 and 50]. The concept is to determine the particular finite size (number of terms) model that best matches the performance of some system. The existing techniques attempt to fit this problem to a particular parameter estimation form of solution (e.g. recursive-in-order model form and the use of Levinson's algorithm). The results of Chapter III and Appendix B indicate that these methods would generally lead to suboptimal models unless the restrictive assumptions are met. It makes more sense to grow a model using a general performance modeling technique like the Search Indicator Growth Algorithm, rather than disguising the problem in a parameter estimation form. It is also easier to optimize the model performance while limiting the number of model

terms. The experiments in Chapter VII contain many examples showing where the Search Indicator Growth Algorithm produced equal size models with vastly superior performance compared with recursive-in-order techniques.

B. CONCLUSIONS

The purpose of this research was to extend existing techniques for experimentally developing discrete-time model equations to represent the input-output behavior of linear and nonlinear systems.

We started by dividing the problem into four key parts; the functional form of the model, choice of error minimization method, efficiency of model selection and evaluation, and verification of the quality of the model for various current applications. After a discussion of existing discrete-time model forms, we adopted the more general Bivariate Volterra Model (BVM). Various error minimization methods were examined, and it was shown how the Covariance least squares method is generally significantly superior to the Autocorrelation method typically used in the literature. We next developed expressions for the distortions in both the fitting error and the coefficient estimates of a linear recursive model form when there exists additive output noise. These results clearly show the effects of the magnitude of the recursive coefficients and the sample autocorrelation values of the noise sequence.

A general set of recursive solution and evaluation equations was developed for computational savings and as a

unifying basis for the work that followed. This enabled us to easily evaluate a wide range of model equations without limiting the form of the model or making other unnecessarily restricting assumptions. Existing model growth techniques were examined and extended to allow consideration of the more general BVM form. Inherent limitations (e.g. maximum number of model terms less than or equal to the number of available data measurements) were recognized.

A major goal was the development of a growth technique that could perform better than existing techniques. The concept of Search Indicators was introduced, and led to the development of the Search Indicator Growth Algorithm and related special techniques. The physical interpretation and significant computational savings resulting from the use of this algorithm were presented, along with provisions for handling the important problem of colinearity among the model terms. Various conditions affecting the evaluation and growth of models were examined, and several special techniques were proposed.

The remainder of the thesis focused on experimental verification of the strengths and weaknesses of the various model growth techniques. These results clearly showed the improved performance of the Search Indicator Growth Algorithm. Some specific ideas for improving the development and use of mathematical models in several current applications were also presented.

C. AREAS FOR FURTHER RESEARCH

Various interesting questions were encountered during the development of this thesis. Those listed below can form the basis for extension of this work and are recommended as areas for further research.

1. The BVM form was emphasized in this research but the modeling techniques are not limited to terms that only contain integer powers, and products of powers, of past and present input values and past output values. The model form could be extended to include decaying exponentials, divide functions, and other factors of the measurements. These could be explicitly included, or we could approximate factors like exponentials with difference equations.
2. The Candidate Model Specification Technique is a first heuristic approach for specifying the candidate terms to use in the Search Indicator Growth Algorithm. Techniques for decoding the patterns in the residuals might lead to improved methods.
3. The development of the key search indicator I(j,12) included the definition of a matrix H(i-1) given by Eq. {6.25}. This matrix has several interesting properties (e.g. positive semi-definite and idempotent), and it might be possible to exploit these to produce improved search indicators.

4. The Search Indicator Growth Algorithm performance is dependent on the choice of the candidate model terms at each iteration, and the values of the two heuristic variables h_1 and h_2. These values could either be kept fixed throughout the modeling iterations, or possibly be made adaptive.
5. The Search Indicator Growth Algorithm has been shown to be useful for bringing terms into the model equation. Since the candidate set is allowed to expand after each growth iteration, it follows that unneeded terms might exist in the model. It might be possible to develop efficient search indicators that operate on the existing set of model terms and suggest which should be eliminated at each iteration. Existing, but expensive, techniques include backward regression [Ref. 40].
6. Chapter VI made a case for uniform amplitude distribution of the controllable input probe sequence, but mentioned that other distributions could provide better results in certain cases. For example, a distribution that put emphasis on higher amplitude values might be better suited for recognizing specific strong nonlinearities in the system.
7. Additional modeling experience needs to be gained with real world experiments.

APPENDIX A. GENERAL RECURSIVE SOLUTION OF A GROWING SET OF NORMAL EQUATIONS

The solution of a large set of normal equations occurs in various fields, including systems identification, linear prediction, and least squares estimation. Successive sets of normal equations must be solved, where the preceding set can be related to the subsequent set in the partitioned form described below.

Set 1: $$A(1)\underline{p}(1) = \underline{h}(1) \qquad \{A.1\}$$

Set 2: $$A(2)\underline{p}(2) = \underline{h}(2) \qquad \{A.2\}$$

where

$$A(2) = \left[\begin{array}{c|c} A(1) & B(2/1) \\ \hline B(2/1)^T & A(2/1) \end{array}\right] \qquad \{A.3\}$$

and

$$\underline{h}(2)^T = \left[\begin{array}{c|c} \underline{h}(1)^T & \underline{h}(2/1)^T \end{array}\right] \qquad \{A.4\}$$

.
.
.

Set i: $$A(i)\underline{p}(i) = \underline{h}(i) \qquad \{A.5\}$$

where

$$A(i) = \left[\begin{array}{c|c} A(i-1) & B(i/i-1) \\ \hline B(i/i-1)^T & A(i/i-1) \end{array}\right] \qquad \{A.6\}$$

and

$$\underline{h}(i)^T = \left[\begin{array}{c|c} \underline{h}(i-1)^T & \underline{h}(i/i-1)^T \end{array}\right] \qquad \{A.7\}$$

Each set is of size c(i), where c(i-1) < c(i).

Rather than solving each set independently, the general partitioned structure of these related sets of equations can be exploited to obtain the solution to each set in an efficient recursive manner. From Eq. {4.22}, we use the notation $q(i) = c(i) - c(i-1)$.

Given a set of c(i) linear normal equation;

$$A(i)\underline{p}(i) = \underline{h}(i) \qquad \{A.8\}$$

where we have a unique solution to the previous set of c(i-1) normal equations;

$$\underline{p}(i-1) = A(i-1)^{-1}\underline{h}(i-1) \qquad \{A.9\}$$

Substituting {A.6} and {A.7} into {A.8} yields;

$$\underline{p}(i) = \left[\begin{array}{c|c} A(i-1) & B(i/i-1) \\ \hline B(i/i-1)^T & A(i/i-1) \end{array}\right] \left[\begin{array}{c} \underline{h}(i-1) \\ \hline \underline{h}(i/i-1) \end{array}\right] \qquad \{A.10\}$$

The partitioned matrix inversion theorem [Ref. 12, pp. 18] permits exploitation of the symmetry of matrix A(i) whenever $|A(i/i-1)| \neq 0$.

$$\underline{p}(i) = \left[\begin{array}{c|c} A(i-1)^{-1} + F(i)G(i)^{-1}F(i)^T & F(i)G(i)^{-1} \\ \hline G(i)^{-1}F(i)^T & G(i)^{-1} \end{array}\right] \left[\begin{array}{c} \underline{h}(i-1) \\ \hline \underline{h}(i/i-1) \end{array}\right] \qquad \{A.11\}$$

where F(i) is a c(i-1) x q(i) matrix, and G(i) is a q(i) x q(i) matrix each defined below.

$$F(i) = -A(i-1)^{-1}B(i/i-1) \qquad \{A.12\}$$

$$G(i) = A(i/i-1) - B(i/i-1)^T A(i-1)^{-1} B(i/i-1)$$

$$= A(i/i-1) + B(i/i-1)^T F(i) \qquad \{A.13\}$$

Expanding Eq. {A.11} produces;

$$\underline{p}(i) = \left[\begin{array}{c} A(i-1)^{-1}\underline{h}(i-1) + F(i)G(i)^{-1}F(i)^T\underline{h}(i-1) + F(i)G(i)^{-1}\underline{h}(i/i-1) \\ \hline G(i)^{-1}F(i)^T\underline{h}(i-1) + G(i)^{-1}\underline{h}(i/i-1) \end{array}\right] \qquad \{A.14\}$$

Define two vectors of size q(i);

$$\underline{g}(i) = \underline{h}(i/i-1) + F(i)^T \underline{h}(i-1) \qquad \{A.15\}$$

$$\underline{k}(i) = G(i)^{-1} \underline{g}(i) \qquad \{A.16\}$$

Substituting {A.15} and {A.16} into {A.14}

$$\underline{p}(i) = \left[\frac{\underline{p}(i-1) + F(i)G(i)^{-1}\underline{g}(i)}{G(i)^{-1}\underline{g}(i)}\right]$$

$$= \left[\frac{\underline{p}(i-1) + F(i)\underline{k}(i)}{\underline{k}(i)}\right] \qquad \{A.17\}$$

$$= \left[\frac{\underline{p}(i-1)}{\underline{0}}\right] + \left[\frac{F(i)}{I}\right] \underline{k}(i) \qquad \{A.18\}$$

Equation {A.18} is our desired answer and is presented in Chapter IV as Eq. {4.35}.

A compact recursive expression for the resulting minimum average sum squared error, $J^2(i)$, can also be developed. Substituting {4.4} into {4.3} for the $(i)^{th}$ model produces;

$$J^2(i) = \frac{1}{N} \underline{y}^T\underline{y} - \underline{r}(i)^T R(i)^{-1} \underline{r}(i)$$

$$= \frac{1}{N} \underline{y}^T\underline{y} - \underline{r}(i)^T \underline{\theta}(i) \qquad \{A.19\}$$

Using the definitions of Eq. {4.9} through {4.12}, the vectors $\underline{r}(i)$ and $\underline{\theta}(i)$ can be rearranged in the form of vectors $\underline{h}(i)$ and $\underline{p}(i)$, respectively. Substituting these into Eq. {A.19} produces;

$$J^2(i) = \frac{1}{N} \underline{y}^T\underline{y} - \underline{h}(i)^T \underline{p}(i) \qquad \{A.20\}$$

Substituting {A.7} and {A.18} into {A.20} and simplifying yields;

$$J^2(i) = \frac{1}{N}\underline{y}^T\underline{y} - \left[\, \underline{h}(i-1)^T \;\vdots\; \underline{h}(i/i-1)^T \,\right] \underline{p}(i)$$

$$= \frac{1}{N}\underline{y}^T\underline{y} - \underline{h}(i-1)^T\underline{p}(i-1) - \underline{h}(i-1)^T F(i)\underline{k}(i) - \underline{h}(i/i-1)^T\underline{k}(i)$$

$$= J^2(i-1) - \left[\, \underline{h}(i-1)^T F(i) + \underline{h}(i/i-1)^T \,\right] \underline{k}(i)$$

$$= J^2(i-1) - \underline{g}(i)^T\underline{k}(i) \qquad \{A.21\}$$

where we made use of {A.15}, {A.16}, and $J^2(i-1)$, the previous evaluation of the $(i-1)^{st}$ model. This last expression appears in Chapter IV as Eq. {4.36}.

APPENDIX B. RELATIONSHIP OF THE GENERAL RECURSIVE ALGORITHM TO LEVINSON'S ALGORITHM

When we are given a set of simultaneous linear equations,

$$A(i)\underline{p}(i) = \underline{h}(i) \qquad \{B.1\}$$

where A(i) is a c(i) by c(i) real matrix, the direct solution of Eq. {B.1} requires on the order of $[c(i)]^3/3$ multiplicative operations. When A(i) is symmetric and positive definite, the solution can be accomplished with the order of $[c(i)]^3/6$ multiplicative operations by various techniques (e.g. Cholesky, LU decomposition, etc.) .

Appendix A developed a general recursive solution for $\underline{p}(i)$, based on the concept that the previous set of $c(i-1) < c(i)$ equations given by;

$$A(i-1)\underline{p}(i-1) = \underline{h}(i-1) \qquad \{A.9\}$$

has been previously evaluated for $\underline{p}(i)$ and $A(i-1)^{-1}$, where the following partitioned matrx relationship exist.

$$A(i) = \left[\begin{array}{c|c} A(i-1) & B(i/i-1) \\ \hline B(i/i-1)^T & A(i/i-1) \end{array}\right] \qquad \{A.6\}$$

and

$$\underline{h}(i)^T = \left[\begin{array}{c|c} \underline{h}(i-1)^T & \underline{h}(i/i-1)^T \end{array}\right] \qquad \{A.7\}$$

This general solution equation for $\underline{p}(i)$ is repeated below;

$$\underline{p}(i) = \left[\begin{array}{c} \underline{p}(i-1) \\ \hline \underline{0} \end{array}\right] + \left[\begin{array}{c} F(i) \\ \hline I \end{array}\right]\underline{k}(i) \qquad \{A.18\}$$

where $\underline{0}$ is the null vector, I is the identity matrix, and;

$$F(i) = -A(i-1)^{-1} B(i/i-1) \qquad \{A.12\}$$

$$G(i) = A(i/i-1) - B(i/i-1)^{T} A(i-1)^{-1} B(i/i-1)$$

$$= A(i/i-1) + B(i/i-1)^{T} F(i) \qquad \{A.13\}$$

$$\underline{g}(i) = \underline{h}(i/i-1) - B(i/i-1)^{T} \underline{p}(i-1)$$

$$= \underline{h}(i/i-1) + F(i)^{T} \underline{h}(i-1) \qquad \{A.15\}$$

$$\underline{k}(i) = G(i)^{-1} \underline{g}(i) \qquad \{A.16\}$$

The matrix F(i) given by Eq. {A.12} requires the use of the inverse of matrix A(i-1). If this was not explicitly solved for previously, it is needed at this point. The vector $\underline{k}(i)$ given by Eq. {A.16} requires the use of the inverse of matrix G(i).

In 1947, Norman Levinson published a paper in which he "in order to facilitate computational procedure, worked out an approximate, and one might say, mathematically trivial procedure" [Ref. 1; pp. 161]. Levinson was working in conjunction with Norbert Wiener on a problem involving linear moving average filter design using a least squares fit. This required solving a set of simultaneous linear equations of the form of Eq. {B.1}. Levinson developed an iterative procedure for obtaining $\underline{p}(i)$ based on the following conditions (given in terms of our notation).

(1) Matrix A(i) is symmetric, and can be represented in the form of Eq. {A.6}. {B.2}

(2) The solution for $\underline{p}(i-1)$ has been previously obtained. {B.3}

(3) A(i/i-1) is a 1 x 1 matrix (scalar), a(i/i-1) {B.4}

Later, Durbin [Ref. 2] simplified the solution for $\underline{p}(i)$ by adding a fourth condition.

(4) $B(i/i-1)$ reduces to a vector $\underline{b}(i/i-1)$ which equals the vector $\underline{h}(i-1)$ in reverse order. {B.5}

Despite this separate and later simplification by Durbin, the procedure is commonly referred to as Levinson's algorithm, and it can be shown that;

$$\underline{p}(i) = \begin{bmatrix} \underline{p}(i-1) \\ \hline 0 \end{bmatrix} + \begin{bmatrix} \underline{f}(i) \\ \hline 1 \end{bmatrix} k(i) \qquad \{B.6\}$$

The main property of this algorithm is that calculation of the vector $\underline{f}(i)$ and the scalar $k(i)$ do not require any matrix inversions. Additionally, the number of multiplicative operations required for the solution of Eq. {B.6} is $2c(i-1)+1$, where $c(i)-1 = c(i-1)$, and $c(i-1)$ is the size of matrix $A(i-1)$. The popularity of Levinson's algorithm is a result of this very small computational cost, which offers a significant savings over the direct matrix solution of Eq. {B.1} by the more conventional techniques.

We demonstrate that {B.6} is a <u>special case</u> of {A.18} based on the four conditions {B.2}, {B.3}, {B.4}, and {B.5} presented above. We first develop an expression for the factor $\underline{f}(i)$ in Eq. {B.6}. From {B.5} we can write

$$B(i/i-1) = \underline{h}(i-1) \text{ in reverse order} \equiv \tilde{\underline{h}}(i-1) \qquad \{B.7\}$$

where the "~" denotes an end-for-end reversal.

Substituting {B.7} into {A.12} produces;

$$F(i) = -A(i-1)^{-1} B(i/i-1)$$

$$= -A(i-1)^{-1} \tilde{\underline{h}}(i-1) \qquad \{B.8\}$$

From the symmetric condition {B.2}, Eq. {A.9} can be written

$$A(i-1)\, \tilde{\underline{p}}(i-1) = \tilde{\underline{h}}(i-1) \qquad \{B.9\}$$

Substituting {B.9} into {B.8} and simplifying yields;

$$F(i) = -A(i-1)^{-1} \tilde{\underline{h}}(i-1)$$

$$= -A(i-1)^{-1} A(i-1) \tilde{\underline{p}}(i-1)$$

$$= -\tilde{\underline{p}}(i-1) \qquad \{B.10\}$$

From {B.10}, we see that F(i) is obtained directly from the previous solution $\underline{p}(i-1)$ without any computational cost.

We now develop an equation for k(i) in Eq. {B.6}. Substitute {B.4}, {B.7}, and {B.10} into {A.13} and solve for the 1x1 matrix G(i).

$$G(i) = A(i/i-1) + B(i/i-1)^{T} F(i)$$

$$= a(i/i-1) - \tilde{\underline{h}}(i-1)^{T} \tilde{\underline{p}}(i-1) \qquad \{B.11\}$$

The solution of Eq. {B.11} requires c(i-1) multiplications.

Substituting {B.10} into {A.15} and simplifying yields;

$$\underline{g}(i) = \underline{h}(i/i-1) + F(i)^{T} \underline{h}(i-1)$$

$$= \underline{h}(i/i-1) - \tilde{\underline{p}}(i-1)^{T} \underline{h}(i-1) \qquad \{B.12\}$$

The solution of Eq. {B.12} requires c(i-1) multiplications.

Next substitute {B.11} and {B.12} into {A.16}.

$$\underline{k}(i) = G(i)^{-1} \underline{g}(i)$$

$$= [\, a(i/i-1) - \tilde{\underline{h}}(i-1)^{T} \tilde{\underline{p}}(i-1) \,]^{-1} [\, \underline{h}(i/i-1) - \tilde{\underline{p}}(i-1)^{T} \tilde{\underline{h}}(i-1) \,]$$

$$= k(i), \text{ a scalar} \qquad \{B.13\}$$

Therefore k(i) requires 2c(i-1)+1 multiplications, and does not involve any matrix inversion, other than the trivial scalar inversion of the 1 x 1 matrix G(i).

We see that under the four stated conditions, the general recursive solution algorithm {A.10} reduces to Levinson's algorithm, and has the same computational cost.

The simplification of Levinson's algorithm is critically dependent upon conditions {B.4} and {B.5}. This first condition, that A(i/i-1) is a 1 x 1 matrix, restricts Levinson's algorithm to be a single-step iterative technique (one increase in size of matrix A(i) over A(i-1)). This limits model growth to only one new term at a time in the general modeling problem. The second critical condition, that the transpose of B(i/i-1) equals $\underline{h}$(i-1) in reverse order, restricts Levinson's algorithm to both the limited cases of model growth that adds terms that are delayed versions of existing terms, and the use of the Autocorrelation error minimization method that produces least squares normal equations with this special structure. Note that matrix A(j) for j=1,2,...,i has to be Toeplitz, and the vector $\underline{h}$(j) must satisfy condition {B.5}.

Multiple channel versions of Levinson's Algorithm have been proposed [Ref. 3 - 9], but these all require the special "recursive-in-order" relationship between the model terms represented in A(i-1), and those terms represented in A(i). This is an unnecessary and suboptimal restriction for the general model growth problem.

APPENDIX C. DETAILS OF ORDER OF COMPLEXITY CALCULATIONS USED TO COMPARE THE GROWTH TECHNIQUES

Using the same convention as Chapter VI, we denote the size c(i-1) of the $(i-1)^{st}$ model as P, and the number of data points in the error minimization as N. When distinction is needed, we use the shorthand "Step D1", "Step B1", or "Step S1" to indicate that we are refering to Step 1 of the direct least squares technique, block-form recursive technique, or search indicator growth technique, respectively. When the computational cost is the same for all three techniques (e.g. steps 1 through 7), we use just the step number.

The computational cost for each of the first seven steps is developed as follows. Each of the three growth techniques uses the identical first seven steps.

Step 1: Set i = 1, and form the term vector $\underline{x}(n,i)$. No cost.

Step 2: Form R(1) using Eq. {4.5}

$$R(i) = \frac{1}{N} X(i)^T X(i) \qquad \{4.5\}$$

Since X(i) is a N x c(i) matrix, each element in the c(i) x c(i) matrix R(i) requires N multiplications and 1 division operation. Because of symmetry, there are c(i)[c(i)+1]/2 elements in matrix R(i) that must be calculated. Therefore the computational cost is [N+1]c(i)[c(i)+1]/2.

Step 3: Form $\underline{r}(i)$ using Eq. {4.6}

$$\underline{r}(i) = \frac{1}{N} X(i)^T \underline{y} \qquad \{4.6\}$$

Each of the c(i) elements in vector $\underline{r}(i)$ requires N multiplications and 1 division, therefore the computational cost is c(i)[N+1].

Step 4: Invert R(i)

Since R(i) is a symmetric matrix of size c(i), it can be inverted at a cost of [c(i)**3]/6 operations.

Step 5: Solve for $J^2(i)$ using Eq. {4.3}

$$J^2(i) = \frac{1}{N} \underline{y}^T \underline{y} - \underline{r}(i)^T R(i)^{-1} \underline{r}(i) \qquad \{4.3\}$$

Since $\underline{y}$ is a size N vector, the first term on the right side of {4.3} can be computed with N+1 operations. Using the preceding definitions for the sizes of vector $\underline{r}(i)$ and matrix R(i), the second term on the right side of {4.3} requires [c(i)**2] + c(i) operations. The total cost for this step is therefore [c(i)**2] + c(i) + N + 1.

Step 6 and Step 7 do not involve any computational cost. Adding costs results in the following complexity equation for the direct least squares technique.

$$O(n) = [c(i)**3]/6 + [c(i)**2][N+3]/2 + c(i)[3N+5]/2 + N+1 \qquad \{C.1\}$$

We continue with the computational cost of the block-form technique.

Step B8: Form A(i/i-1) using Eq. {4.27}

$$A(i/i-1) = \frac{1}{N} W(i/i-1)^T W(i/i-1) \qquad \{4.27\}$$

The N x q(i) matrix W(i/i-1) is the data matrix for the new model terms. Each element in matrix A(i/i-1) requires N multiplications and 1 division operation. Because of symmetry, there are q(i)[q(i)+1]/2 elements that must be calculated, therefore the computational cost is [N+1]q(i)[q(i)+1]/2.

Step B9: Form B(i/i-1) using Eq. {4.26}

$$B(i/i-1) = \frac{1}{N} W(i-1)^T W(i/i-1) \qquad \{4.26\}$$

The N x P matrix W(i-1) is the data matrix for the $(i-1)^{st}$ model obtained previously. Each element in matrix B(i/i-1) requires N multiplications and 1 division operation. Since there are q(i)P elements that must be calculated, the computational cost is [N+1]q(i)P.

Step B10: Form $\underline{h}(i/i-1)$ using Eq. {4.29}

$$\underline{h}(i/i-1) = \frac{1}{N} W(i/i-1)^T \underline{y} \qquad \{4.29\}$$

Using the preceding definitions of the sizes of matrix W(i/i-1) and vector $\underline{y}$, each of the q(i) elements in vector $\underline{h}(i/i-1)$ requires N multiplications and 1 division operation. Therefore the computational cost is q(i){N+1].

Step B11: Form F(i) using Eq. {4.30}

$$F(i) = -A(i-1)^{-1} B(i/i-1) \qquad \{4.30\}$$

The P x P matrix $A(i/i-1)^{-1}$ is the inverse of the least squares matrix for the $(i-1)^{st}$ model obtained previously. Since matrix B(i/i-1) is P x q(i), each element in matrix F(i) requires P multiplications. There are q(i)P elements in matrix F(i) and therefore the computational cost is q(i)[P**2].

Step B12: Form G(i) using Eq. {4.31}

$$G(i) = A(i/i-1) + B(i/i-1)^{T} F(i) \qquad \{4.31\}$$

Using the preceding definitions of the sizes for matrices B(i/i-1) and F(i), each of the [q(i)**2] elements in the result of the second term on the right side of {4.31} requires P multiplications. Since there are [q(i)**2] elements in this resulting matrix, the total computational cost is P[q(i)**2].

Step B13: Form $\underline{g}(i)$ using Eq. {4.32}

$$\underline{g}(i) = \underline{h}(i/i-1) + F(i)^{T} \underline{h}(i-1) \qquad \{4.32\}$$

Using the preceding definitions of the sizes for matrix F(i) and vector $\underline{h}(i/i-1)$, each of the q(i) elements in the result of the second term on the right side of {4.32} requires P multiplications. Therefore, the total computational cost is Pq(i).

Step B14: Invert G(i)

Since G(i) is a symmetric matrix of size q(i), it can be inverted at a cost of [q(i)**3]/6 operations.

Step B15: Form $\underline{k}(i)$ using Eq. {4.33}

$$\underline{k}(i) = G(i)^{-1}\underline{g}(i) \qquad \{4.33\}$$

Since the inverse matrix $G(i)^{-1}$ is size q(i) x q(i), and $\underline{g}(i)$ is a q(i) size vector, each of the q(i) elements of vector $\underline{k}(i)$ requires q(i) multiplications. Therefore, the computational cost is [q(i)**2].

Step B16: Solve for $J^2(i)$ using Eq. {4.36}

$$J^2(i) = J^2(i-1) - \underline{g}(i)^T\underline{k}(i) \qquad \{4.36\}$$

Since $\underline{g}(i)$ and $\underline{k}(i)$ are both size q(i) column vectors, the computational cost is q(i).

Step B17 does not involve any computational cost.

Based on the results of step B17, the growth may stop. Adding complexity notational for each step, results in the following complexity equation for the block-form technique (steps B8 through B17).

$$O(n) = [q(i)**3]/6 + [P+N+3][q(i)**2]/2 + q(i)[NP+[P**2]+2P+[3N+5]/2] \qquad \{C.2\}$$

If additional growth iterations are required for adequate modeling performance, one additional computational step is required before starting again at step B7.

Step B18: Form inverse of A(i) using Eq. {4.37}

$$A(i)^{-1} = \left[\begin{array}{c|c} A(i-1) + F(i)G(i)^{-1}F(i)^T & F(i)G(i)^{-1} \\ \hline G(i)^{-1}F(i) & G(i)^{-1} \end{array}\right] \qquad \{4.37\}$$

All of the indicated matrices in {4.37} have already been calculated. The only computations required are

those necessary to form the matrix factors $F(i)G(i)^{-1}F(i)^{T}$ and $F(i)G(i)^{-1}$. Using the previously defined sizes of these matrices, these factors can both be calculated with a total cost of $q(i)[P**2] + P[q(i)**2]$.

We continue with the computational costs of the search indicator growth technique. The cost of step 1 through step 7 is the same as first two growth techniques.

Step S8: Form $I(j,12)$ for terms in $\underline{w}(n,i/i-1)$ using the definition of Eq. {6.29}, repeated here;

$$I(j,12) = \frac{\left[\frac{1}{N}\,\underline{w}_j(i/i-1)^{T}\,\underline{e}(i-1)\right]^{2}}{\frac{1}{N}\,\underline{w}_j(i/i-1)^{T}\,\underline{w}_j(i/i-1)} \qquad \{6.29\}$$

Using the computational results of Table 5 and Table 6, the cost of $I(j,12)$ for the first term is NP+2N+4, and the cost for each of the second through the $q(i)^{th}$ term is 2N+4. Therefore, the total cost for the $q(i)$ indicators $I(j,12)$ is = $PN + [2N+4]q(i)$.

Step S9 involves selecting the subset of terms with values of $I(j,12)$ greater than a specified level h_1. Depending on the value of h_1 and the values of $I(j,12)$, there will be an integer k number of terms left. The size of the term vector $\underline{w}(n,i/i-1)$ is reduced from $q(i) \times 1$ to $k \times 1$. There is no significant cost of this step.

Step S10: Form A(i/i-1) using the reduced vector $\underline{w}(n,i/i-1)$ in Eq. {4.23} and substituting this into Eq. {4.27} to form;

$$A(i/i-1) = \frac{1}{N} W(i/i-1)^T W(i/i-1) \qquad \{4.27\}$$

The N x k matrix W(i/i-1) is now the reduced data matrix for the new model terms. Each element in matrix A(i/i-1) requires N multiplications and 1 division operation. Because of symmetry, there are k[k+1]/2 elements that must be calculated, therefore the computational cost is [N+1]k[k+1]/2.

Step S11: Form B(i/i-1) using Eq. {4.26}

$$B(i/i-1) = \frac{1}{N} W(i-1)^T W(i/i-1) \qquad \{4.26\}$$

The N x P matrix W(i-1) is the data matrix for the $(i-1)^{st}$ model obtained previously. Each element in matrix B(i/i-1) requires N multiplications and 1 division operation. Since there are kP elements that must be calculated, the computational cost is [N+1]kP.

Step S12: Form $\underline{h}(i/i-1)$ using Eq. {4.29}

$$\underline{h}(i/i-1) = \frac{1}{N} W(i/i-1)^T \underline{y} \qquad \{4.29\}$$

Using the preceding definitions of the sizes of matrix W(i/i-1) and vector $\underline{y}$, each of the k elements in vector $\underline{h}(i/i-1)$ requires N multiplications and 1 division operation. Total cost is k[N+1].

Step S13: Form F(i) using Eq. {4.30}

$$F(i) = -A(i-1)^{-1} B(i/i-1) \qquad \{4.30\}$$

The P x P inverse matrix $A(i/i-1)^{-1}$ is the least squares matrix for the $(i-1)^{st}$ model obtained previously. Since matrix B(i/i-1) is P x k, each element in matrix F(i) requires P multiplications. There are kP elements in matrix F(i) and therefore the computational cost is k[P**2].

Step S14: Form G(i) using Eq. {4.31}

$$G(i) = A(i/i-1) + B(i/i-1)^{T} F(i) \qquad \{4.31\}$$

Using the preceding definitions for the sizes of matrices B(i/i-1) and F(i), each element in the result of the second term on the right side of {4.31} requires P multiplications. Since there are [k**2] elements in this resulting matrix, the total computational cost is P[k**2].

Step S15: Form $\underline{g}(i)$ using Eq. {4.32}

$$\underline{g}(i) = \underline{h}(i/i-1) + F(i)^{T} \underline{h}(i-1) \qquad \{4.32\}$$

Using the preceding definitions of the sizes for matrix F(i) and vector $\underline{h}(i/i-1)$, each of the k elements in the result of the second term on the right side of {4.32} requires P multiplications. Therefore, the total computational cost is Pk.

Step S16: Invert G(i)

Since G(i) is a symmetric matrix of size k, it can be inverted at a cost of [k**3]/6 operations.

Step S17: Form vector $\underline{k}(i)$ using Eq. {4.33}

$$\underline{k}(i) = G(i)^{-1}\underline{g}(i) \qquad \{4.33\}$$

Since the inverse matrix $G(i)^{-1}$ is size k x k, and $\underline{g}(i)$ is a k size vector, each of the k elements of vector $\underline{k}(i)$ requires k multiplications. Therefore, the computational cost is [k**2].

Step S18: Solve for $J^2(i)$ using Eq. {4.36}

$$J^2(i) = J^2(i-1) - \underline{g}(i)^T\underline{k}(i) \qquad \{4.36\}$$

Since $\underline{g}(i)$ and $\underline{k}(i)$ are both size k column vectors, the computational cost is k multiplications.

Step S19 does not involve any computational cost.

Based on the results of step S19, the growth may stop. Adding complexity notation for each step, results in the following complexity equation for the block-form technique (steps S8 through S19).

$$O(n) = [k**3]/6 + [P+N+3][k**2]/2 + k[NP+[P**2]+2P+[3N+5]/2] + NP + [2N+4]q(i) \qquad \{C.3\}$$

If additional growth iterations are required for adequate modeling performance, one additional computational step is required before starting again at step S7.

Step B18: Form inverse of A(i) using Eq. {4.37}

$$A(i)^{-1} = \left[\begin{array}{c|c} A(i-1)^{-1} + F(i)G(i)^{-1}F(i)^T & F(i)G(i)^{-1} \\ \hline G(i)^{-1}F(i) & G(i)^{-1} \end{array}\right] \qquad \{4.37\}$$

All of the indicated matrices in {4.37} have already been calculated. The only computations required are those necessary to form the matrix factors

$F(i)G(i)^{-1}F(i)^{T}$ and $F(i)G(i)^{-1}$. Using the previously defined sizes of these matrices, these factors can both be calculated with a total cost of k[P**2] + P[k**2].

LIST OF REFERENCES

1. Levinson, N., "The Wiener RMS Error Criteria in Filter Design and Prediction", J. Math Phys., Vol 25, pp. 261-278, January 1947.

2. Durbin, J., "The fitting of Time Series Models", Rev. Inst. Int. Statist., Vol. 28, No. 3, pp. 233-243, 1960.

3. Robinson, E.A., Multichannel Time Series Analysis with Digital Computer Programs, Rev-Ed., Holden-Day, 1967.

4. Morf, M. and others, "Some New Algorithms for Recursive Estimation in Constant, Linear, Discrete Time Systems", IEEE Trans, Auto Control, Vol. AC-19, pp.315-323, Aug 1974

5. Morf, M. and others, "Ladder Forms for Identification and Speech Processing", Proceedings IEEE Dec. & Contr. Conf., pp. 1074-1078, 1977.

6. Lee, D. T. L. and others, "Recursive Least Square Ladder Estimation Algorithms", IEEE Trans. ASSP, Vol. ASSP-29, No. 3, pp. 627-641, June 1981.

7. Friedlander, B. and others, "Extended Levinson and Chandrasekhar Equations for the General Discrete-Time Linear Estimation Problems", IEEE Trans. Auto Control, Vol. 23, No. 4, pp. 653-659, Aug 1978

8. Perry, F. A. Parametric Modeling of Linear and Nonlinear Systems, Ph.D Dissertation, U.S. Naval Postgraduate School, 1980

9. Parker, S. R., and Perry, F. A., "A Discrete ARMA Model for Nonlinear System Identification", IEEE Trans. Ckts. and Systems, Vol. CAS-28, No. 3, pp. 224-233, March 1981.

10. Chua, L. O. and Lin, P., Computer Aided Analysis of Electronic Circuits: Algorithm and Computational Techniques, Prentice Hall, Englewood Cliffs, N.J., 1975

11. Eykoff, P., System Identification: Parameter and State Estimation, John Wiley, New York, 1974

12. Graybill, F. A., Theory and Application of the Linear Model, Duxbury Press, North Scituate, Mass., 1976

13. Iserman, R. and Bauer, U., "Two Step Process Identification with Correlation Analysis and Least Squares Parameter Estimation", Trans. A.S.M.E. (Journal of Dynamic Systems Measurement & Control), pp. 426-432, Dec 1974

14. Hsia, T. C., System Identification, D. C. Heath & Company, Lexington ,Mass., 1977

15. Wu, Y. C. and Rekasius, Z. V., "Deterministic Identification of Linear Dynamic Systems", IEEE Trans. Auto Control, Vol. AC-25, No. 3, pp. 501-504, June 1980

16. Wu, Y. C., Identification of Linear Dynamic Systems, Ph.D Dissertation, Northwestern University, 1978

17. Box, G. E. and Jenkins, G. M., Time Series Analysis, Holden-Day, Inc., San Francisco, 1976

18. Graupe, D., Identification of Systems, Robert E. Krieger Publishing Company, Huntington, New York, 2nd ed., 1976

19. Kalman, R. E., "Design of a Self-Optimizing Control System", Trans. A.S.M.E., Vol. 80, pp. 468-478, Feb 1958

20. Alper, P., "A Consideration of the Discrete Volterra Series", IEEE Trans. Auto Control, Vol. 10, pp. 322-327, July 1965

21. Fu, F. C. and Farison, J. B., "On the Volterra-Series Functional Identification of Non-Linear Discrete-Time Systems", Int. J. Control, Vol. 18, pp. 1281-1289, 1973

22. Beghelli, S. and Guidorzi, R., "Bilinear System Identification from Input-Output Sequences", Proc. IV IFAC Symposium Identification & System Parameter Estimation, Tbilisi, U.S.S.R., Sept. 1976

23. Baheti, R. S., Parameter and Volterra Kernel Estimation of Bilinear Systems, Ph.D Dissertation, Oregon State University, Vorvallis, 1977

24. Baheti, R. S. and others, "A New Approach to Volterra Kernel Estimation of Bilinear Systems", Proc. IEEE Conf. Dec. & Control, pp. 546-550, 1979

25. Baheti, R. S. and others, "Second-Order Correlation Method for Bilinear System Identification", IEEE Trans. Auto Control, Vol. AC-25, pp. 1141-1146, Dec 1980

26. Parker, S. R., "An Autoregressive Moving Average (ARMA) Discrete Nonlinear Model", 1980 IEEE Int. Symposium Ckts. & Systems, New York, pp. 918-920, April 1980

27. Mandelberg, M., "Systematic Experimental Determination of Discrete-Time Models for Nonlinear Systems", Record of 10th IFIP Conf. on System Modeling and Optimization, New York, pp. 233, Aug. 1981

28. Chandra, S. and Lin, W. C., "Experimental Comparison Between Stationary and Nonstationary Formulations of Linear Prediction Applied to Voiced Speech Analysis", IEEE Trans. ASSP, Vol. ASSP-22, No. 6, pp. 403-415, Dec. 1974

29. Makhoul, J., "Linear Prediction: A Tutorial Review", Proc. IEEE, Vol. 63, pp. 561-580, 1975

30. Beck, J. V. and Arnold, K. J., Parameter Estimation in Science and Engineering, John Wiley, New York, 1977

31. Marple, S. L., "Efficient Least Squares FIR System Identification", IEEE Trans. ASSP, Vol. ASSP-29, No. 1, pp. 62-73, Feb 1981

32. Cox, D. R. and Lewis, P. A. W., The Statistical Analysis of Series of Events, John Wiley, New York, 1966

33. Jenkins, G. M. and Watts, D. G., Spectral Analysis and its Applications, Holden-Day, San Francisco, 1968

34. Anderson, T. W. The Statistical Analysis of Time Series, John Wiley, New York, 1970

35. IEEE Trans. Auto Control, Special Issue on System Identification and Time Series Analysis, Vol. AC-19, No. 6, Dec. 1974

36. Wheelwright, S. C. and Mauridakis, S., Interactive Forecasting, Univariate and Multivariate Methods, Holden Day, San Francisco, 1978

37. Woodall, S. R., An Interactive Software Package for Time Series Analysis, Masters Thesis, U.S. Naval Postgraduate School, 1978

38. Makhoul, J., "Stable and Efficient Lattice Methods for Linear Prediction", IEEE Trans. ASSP, Vol. ASSP-25, No. 5, Oct. 1977

39. Satorius, E. H. and Pack, J. D., "On the Application of Lattice Algorithms to Data Equilization", in Record of 13th Annual Asilomar Conf. Circuits, Systems, & Computers, Pacific Grove, CA, Nov 1979

40. Wesolowsky, G. O., Multiple Regression and Analysis of Variance, John Wiley, New York, 1976

41. Farlow, S. J. "The GMDH Algorithm of Ivakhnenko", The American Statistican, Vol. 35, No. 4, pp. 210-215, November 1981

42. Hsia, T. C., "A Two-Stage Least-Squares Procedure for System Identification", IEEE Trans. Auto Control, Vol. AC-26, No. 3, pp. 742-745, June 1981

43. Fisz, M., Probability Theory and Mathematical Statistics, 3rd ed., John Wiley, New York, 1963

44. Garcia, R. F., Fault Isolation Computer Methods, NASA Contractor Report CR-1758, 1971

45. Willsky, A. S., "A Survey of Design Methods for Failure Detection in Dynamic Systems", Automatica, Vol. 12, pp. 601-611, Nov 1976

46. Saeks, R. and W-Liu, R., "Fault Diagnosis in Electronic Circuits", Jour. of the Soc. of Instr. and Control Engrgs., Vol 20., pp. 20-22, 1981

47. Clark, R. N., "The Dedicated Observer Approach to Instrument Failure Detection", Proc. 18th IEEE Conf. Dec. & Control, pp. 237-241, 1979

48. Sakla, A. A. and others, "A Sensitivity Algorithm for Fault Detection in Analog Circuits", Proc. 1980 IEEE Int. Symp. Ckts. & Systems, Houston, pp. 1075-1077

49. Sims, C. S., "Reduced Order Modeling", 13th Asilomar Conf., Pacific Grove, CA, pp. 215-218, 1979

50. Shamash, Y. "Stable Biased Reduced Order Models using the Routh Technique", Proc. 18th IEEE Conf. Dec. & Control, pp. 864, 1979

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 62 Naval Postgraduate School Monterey, California 93940	1
4.	Professor R. W. Hamming, Code 52Hg Department of Computer Science Naval Postgraduate School Monterey, California 93940	3
5.	Professor D. E. Kirk, Code 62Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Professor R. Panholzer, Code 62Pz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	Assoc. Professor P. H. Moose, Code 62Me Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
8.	Assoc. Professor P. C. C. Wang, Code 53Wg Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
9.	Assoc. Professor U. Kodres, Code 52Kr Department of Computer Science Naval Postgraduate School Monterey, California 93940	1

10. Professor S. R. Parker, Code 62Px — 1
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California 93940

11. Mr. Martin Mandelberg — 3
35 Highland Drive
Ledyard, Connecticut 06339

12. Dr. Joel Morris — 1
Code 427-ONR
Office of Naval Research
800 North Quincy
Arlington, Virginia 22217

13. Captain Stacy Holmes, USN — 1
Naval Electronics Systems Command
Code 310
2511 Jefferson Davis Highway
Arlington, Virginia 22202

14. Lieutenant Commander John Martinos — 1
Helenic Navy
Glafkis 22
Chalandri
Athens, Greece

15. Commanding Officer — 1
Naval Underwater Systems Center
New London Laboratory
New London, Connecticut 06320
ATTN: Dr. C. N. Pryor

16. Commanding Officer — 1
Naval Underwater Systems Center
New London Laboratory
New London, Connecticut 06320
ATTN: Dr. William Von Winkle

17. Commanding Officer — 1
Naval Underwater Systems Center
New London Laboratory
New London, Connecticut 06320
ATTN: Dr. Edward Eby

18. Commanding Officer — 1
Naval Underwater Systems Center
New London Laboratory
New London, Connecticut 06320
ATTN: Dr. Albert Nuttall

19. Commanding Officer — 2
Coast Guard Research and Development Center
Avery Point
Groton, Connecticut 06340
ATTN: Dr. Lloyd Breslau

20. Dr. Gene Sogliero — 1
Pfizer Central Research
Department of Clinical Research
Bldg. 118, Eastern Point Road
Groton, Connecticut 06340

21. Dr. Jimmy D. Woods — 1
RFD 2
Box 136B
Wheeler Road
Stonington, Connecticut 06378

22. Commander — 1
Naval Ocean Systems Center
San Diego, California 94305
ATTN: Lieutenant Frank Perry, USN

23. Commanding Officer — 1
Naval Civil Engineering Laboratory
Port Hueneme, California 93043
ATTN: Dr. Donald L. Birkimer

24. Dr. Michael Danchak — 1
Chairman, Department of Computer Science
Hartford Graduate Center
275 Windsor Street
Hartford, Connecticut 06120