

AD-A102 707

NAVAL POSTGRADUATE SCHOOL MONTEREY CA

F/G 12/1

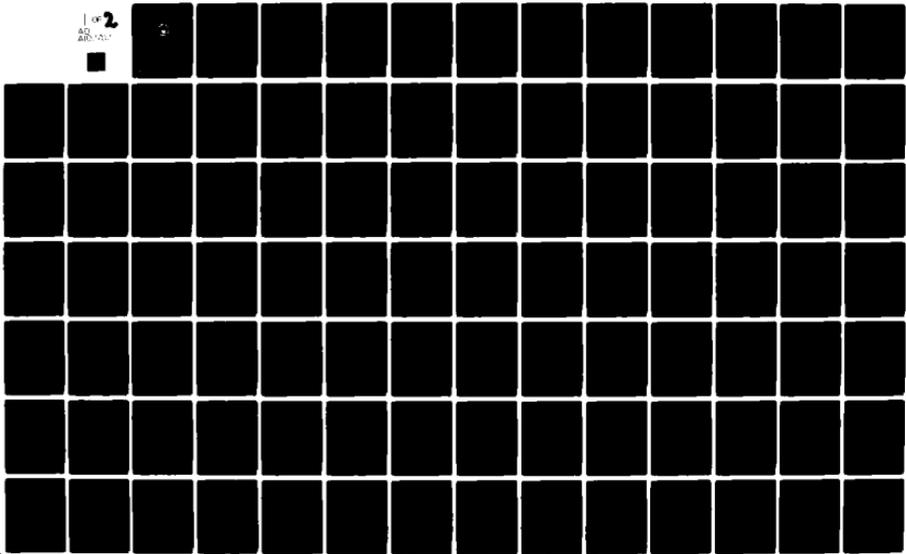
AN ADAPTIVE KALMAN IDENTIFIER AND ITS APPLICATION TO LINEAR AND--ETC(U)

MAR 81 L M MAYORAL

UNCLASSIFIED

NL

1 of 2  
AD  
ARJ-002



LEVEL

2

AD A102707

# NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC  
SELECT  
AUG 11 1981

## THESIS

AN ADAPTIVE KALMAN IDENTIFIER  
AND ITS APPLICATION TO  
LINEAR AND NON-LINEAR ARMA MODELING.

by

Leopoldo M. Mayoral

11 March 1981

Thesis Advisor:

S.R. Parker

Approved for public release; distribution unlimited.

DTIC FILE COPY

81 8 11 035

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A102707	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Adaptive Kalman Identifier and Its Application to Linear and Non-Linear ARMA Modeling		5. TYPE OF REPORT & PERIOD COVERED Engineer's Thesis; March 1981
7. AUTHOR(s) Leopoldo M. Mayoral		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE March 1981
		13. NUMBER OF PAGES 162
		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) Kalman Filter                      Autoregressive Moving-Average (ARMA) Adaptive Kalman Identifier      Least Mean Squares (LMS) Filter ARMA Modeling                    Least Mean Squares (LMS) Recursive Autoregressive (AR)              Filter Moving Average (MA)             Adaptive Lattice Filter		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  The problem of accurately replicating the parameters which define a given system for the purposes of implementing modern control strategies is important. Using an Autoregressive-Moving Average (ARMA) representation for the unknown system, a model is identified by processing input/output data to estimate the coefficients associated with the ARMA equation. Identification of unknown system parameters using Kalman filtering methods		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE  
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

#20 - ABSTRACT - (CONTINUED)

was accomplished by augmenting the state vector. In this thesis the Kalman filter is formulated so that parameters can be identified explicitly. We call this approach the Adaptive Kalman Identifier (AKI).

It is shown that the Adaptive least mean square (LMS) Adaptive Recursive LMS and Adaptive Lattice filters are special suboptimal cases of the AKI. The convergence and modeling properties are compared with those of the AKI by simulation using various types of data.

With minor modification, the AKI algorithm was used to identify the linear and non-linear ARMA models of the phase locked loop (PLL). A discrete PLL using a forward Euler integration scheme was used as a source of non-linear data. The AKI technique appears to enable one to discern when a potential non-linear system enters into its non-linear mode of operation.

#19 - KEY WORDS - (CONTINUED)

System Identification  
Non-Linear ARMA Modeling  
Overmodeling

Accession For	<input checked="" type="checkbox"/>
NTIS CRA&I	<input type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

Approved for public release; distribution unlimited.

An Adaptive Kalman Identifier  
and Its Application to  
Linear and Non-Linear ARMA Modeling

by

Leopoldo M. Mayoral  
Lieutenant, United States Navy  
B.S., University of Texas, 1974  
M.S., Naval Postgraduate School, 1980

Submitted in partial fulfillment of the  
requirements for the degree of

ELECTRICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL  
March 1981

Author

Leopoldo M. Mayoral

Approved by:

Lydney R. Parker

Thesis Advisor

Glen A. Myers

Second Reader

D. E. Kirk

Chairman, Department of Electrical Engineering

William M. Jolley

Dean of Science and Engineering

## ABSTRACT

The problem of accurately replicating the parameters which define a given system for the purposes of implementing modern control strategies is important. Using an Autoregressive-Moving Average (ARMA) representation for the unknown system, a model is identified by processing input/output data to estimate the coefficients associated with the ARMA equation. Identification of unknown system parameters using Kalman filtering methods was accomplished by augmenting the state vector. In this thesis the Kalman filter is formulated so that parameters can be identified explicitly. We call this approach the Adaptive Kalman Identifier (AKI).

It is shown that the Adaptive least mean square (LMS), Adaptive Recursive LMS and Adaptive Lattice filters are special suboptimal cases of the AKI. The convergence and modeling properties are compared with those of the AKI by simulation using various types of data.

With minor modifications, the AKI algorithm was used to identify the linear and non-linear ARMA models of the phase locked loop (PLL). A discrete PLL using a forward Euler integration scheme was used as a source of non-linear data. The AKI technique appears to enable one to discern when a potential non-linear system enters into its non-linear mode of operation.

TABLE OF CONTENTS

I.	INTRODUCTION -----	13
	A. BACKGROUND -----	13
	B. INVESTIGATIONS AND CONTRIBUTIONS -----	16
	C. ORGANIZATION -----	17
II.	FORMULATION OF THE PROBLEM -----	18
	A. DETERMINISTIC CASE -----	18
	B. NON DETERMINISTIC CASE -----	20
	1. Autoregressive Form ( $a_i = 0$ ) -----	24
	2. Moving Average Form ( $b_i = 0$ ) -----	27
	3. Autoregressive-Moving Average Form -----	30
	C. OBSERVATIONS -----	33
III.	COMPARISON BETWEEN THE ADAPTIVE MA KALMAN IDENTIFIER AND WIDROW LMS ADAPTIVE FILTER -----	36
	A. PRELIMINARIES -----	36
	B. THE COMPARISON -----	38
	C. OBSERVATIONS -----	41
IV.	COMPARISON BETWEEN THE ADAPTIVE KALMAN IDEN- TIFIER AND FEINTUCH ADAPTIVE RECURSIVE LMS FILTER -----	43
	A. PRELIMINARIES -----	43
	B. THE COMPARISON -----	47
	C. OBSERVATIONS -----	53
V.	SYNTHETIC DATA GENERATING PLANTS -----	54
	A. AUTOREGRESSIVE-MOVING AVERAGE DATA -----	54
	1. Moving Average Data (Case 1) -----	55
	2. Autoregressive Data (Case 2) -----	55
	3. Autoregressive-Moving Average Data (Case 3) -----	56

B.	PHASE LOCKED LOOP DATA -----	57
VI.	SOFTWARE -----	67
A.	ADAPTIVE KALMAN IDENTIFIER -----	67
B.	LMS ADAPTIVE FILTER -----	69
C.	ADAPTIVE RECURSIVE LMS FILTER -----	73
VII.	NON-LINEAR IDENTIFICATION -----	75
VIII.	FINDINGS, CONCLUSION -----	81
A.	ORDER OF THE UNKNOWN SYSTEM IS KNOWN -----	81
1.	AKI vs LMS Adaptive Filter -----	81
2.	AKI Applied to Autoregressive Data -----	91
3.	AKI Applied to ARMA Data -----	93
4.	AKI vs Adaptive Recursive LMS Filter -----	98
5.	AKI Applied to PLL Data (Linear Region) -	103
6.	AKI Applied to PLL Data (Non-linear Region) -----	103
B.	ORDER OF THE UNKNOWN SYSTEM IS NOT KNOWN (OVERMODELING) -----	109
1.	AKI vs LMS Adaptive Filter -----	109
2.	AKI Applied to AR and ARMA Data -----	114
C.	CONCLUSION -----	115
D.	TOPICS FOR FURTHER CONSIDERATION -----	118
APPENDIX A:	The Discrete Weiner Problem -----	120
APPENDIX B:	Program ARMA -----	124
APPENDIX C:	Program PLL -----	126
APPENDIX D:	Program ADAPTSN -----	131
APPENDIX E:	Program LMS -----	147
APPENDIX F:	Program LMSR -----	151

LIST OF REFERENCES ----- 158

INITIAL DISTRIBUTION LIST ----- 161

LIST OF TABLES

TABLE		
6.1	AKI Flag Meanings -----	68
6.2	Valid Flag Combinations -----	69
7.1	Non-linear Weighting Coefficients Calculated Using Three Methods -----	80
8.1	Actual vs AKI Estimates for Coefficients of Equation 8.3 -----	93
8.2	Actual vs AKI Estimates for Coefficients of Equation 8.4 at the 371st Iteration -----	94
8.3	Actual vs AKI and Adaptive Recursive LMS Filter Estimates for Coefficients of Equation 8.6 -----	100
8.4	Actual vs AKI Estimates of the ARMA Repre- sentation for the PLL (Linear Region) -----	103
8.5	Overmodeling of a MA Process Using the AKI ( $k = 250$ ) -----	113
8.6	Overmodeling of a MA Process Using the AKI of orders $AR = 2$ , $MA = 6$ -----	113

## LIST OF FIGURES

FIGURE		
2.1	Adaptive Kalman Identifier -----	34
3.1	LMS Adaptive Filter Implementation of System Identification -----	37
4.1	Adaptive Recursive LMS Filter Applied to System Identification -----	48
5.1	ARMA Digital Plant -----	56
5.2	PLL Block Diagram -----	58
5.3	Final PLL Block Diagram -----	59
5.4	PLL Step Response, Linear Region -----	60
5.5	PLL Step Response, Non-linear Region -----	61
5.6	Discrete PLL Block Diagram -----	62
5.7	Programmable PLL Implementing Forward Euler Integration -----	63
5.8	Discrete PLL Step Response, Linear Region ----	64
5.9	Discrete PLL Step Response, Non-linear Region -----	65
6.1	AKI General Program Flow -----	70
6.2	LMS Filter Program Flow -----	72
6.3	LMSR Filter Program Flow -----	74
7.1	PLL, Third Order Taylor Series Approximation -	76
8.1	LMS Adaptive Filter Weights, $k_s = -.200$ -----	83
8.2	LMS Adaptive Filter Weights, $k_s = -.300$ -----	84
8.3	LMS Adaptive Filter Weights, $k_s = -.400$ -----	85
8.4	LMS Adaptive Filter Weights, $k_s = -.500$ -----	86
8.5a	LMS Adaptive Filter Weights, $k_s = -.600$ -----	87

FIGURE

8.5b	LMS Adaptive Filter Weights, $k_s = -.200$ , with Increased Measurement Noise -----	88
8.6	AKI Moving Average Plant Coefficients -----	89
8.7	LMS Adaptive Filter Weights -----	89
8.8	AKI Averaged Gains for Moving Average Model --	90
8.9	AKI Moving Average Instantaneous Gains -----	90
8.10	AKI Autoregressive Plant Coefficients -----	92
8.11	AKI Autoregressive Averaged Gains -----	92
8.12	AKI(4,5) Computed Coefficients, Perry's Example -----	95
8.13	AKI(4,5) Average Gain Values -----	95
8.14a	Pole/zero Models Produced by AKI and Lattice Algorithms for a Plant with the Character- istic Transfer Function of Equation (8.4) ----	96
8.14b	Pole/zero Models Produced by AKI and Lattice Algorithms for a Plant with the Character- istic Transfer Function of Equation (8.4) at the 371st Iteration -----	97
8.15	Instantaneous Gain Values, AKI(4,5) Model ----	98
8.16	AKI(2,2) Computed Coefficients, Feintuch Example -----	101
8.17	LMSR Computed Coefficients, Feintuch Example -	101
8.18	AKI(2,2) Averaged Gains, Feintuch Example ----	102
8.19	Pole/zero Models Produced by AKI and Adaptive Recursive LMS Algorithms for a Plant with the Characteristic Transfer Function of Equation 8.6 -----	102
8.20	PLL, AKI(2,3) Computed Coefficients -----	104
8.21	Averaged Gain Values, AKI(2,3) PLL Model ----	104
8.22	PLL Root Locus Analysis vs Computed Roots of the AKI ARMA Model -----	106

FIGURE

8.23	Variation of $\bar{\alpha}$ vs Input Noise Signal Power ---	108
8.24	Comparison of $\frac{\sin x}{x}$ and P.D.F. of Input Noise Signal -----	110
8.25a	AKI(0,4) Model of a Third Order MA Process ----	112
8.25b	Fourth Order LMS Filter Weights of a Third Order MA Process -----	112
8.26	AKI(3,1) Overmodeled AKI(4,2) -----	116
8.27	AKI(3,1) Overmodeled AKI(5,5) -----	116
8.28	AKI(4,5) Overmodeled AKI(5,6) -----	117
A.1	The Discrete Weiner Problem -----	121

### ACKNOWLEDGMENT

I wish to express my gratitude and appreciation for the encouragement which I received, the patience which was extended to me, and the many countless hours which my thesis advisor, Professor S.R. Parker, unselfishly spent discussing and exchanging the many ideas that went into this thesis. Without Professor Parker's guidance this work could easily have been insurmountable.

I, however, am in much greater debt to my wife, Alice, whose tenacious and unwavering love and confidence in me allowed me to successfully progress through seemingly fruitless avenues of research. Her love, patience and understanding was neverfailing.

Lastly, I thank my son, Joshua, for demonstrating to me what life is all about.

## I. INTRODUCTION

The accomplishments in the area of microprocessor technology in the last decade have made a noticeable impact in the area of modern control. The desire to implement modern control theories taking advantage of these advancements, has made it imperative that the engineer attempt to mathematically replicate the system he ultimately desires to control. Efforts in this regard have, hence, generated a growing interest in the area of system identification [Ref. 1,2,3,4]. By implication this thesis concerns itself with discrete/digital signal processing.

### A. BACKGROUND

System identification or modeling can be accomplished by innovative application of existing techniques which were generally considered filtering or state estimation methods [Ref. 5]. Previous research efforts have, for obvious reasons, focused on linear modeling; however, there is a rising interest in non-linear modeling methods [Ref. 6,7,8].

An attractive form for the representation of an unknown system is the Autoregressive-Moving Average (ARMA) equation,

$$y(k) = \sum_{i=0}^{\infty} a_i u(k-i) - \sum_{i=1}^{\infty} b_i y(k-i) \quad (1.1)$$

which states that the present output,  $y(k)$ , is a linear combination of past outputs,  $y(k-i)$ , and of past and present

inputs,  $u(k)$ . Its attractiveness lies in its linear character and easily implementable structure using microprocessor/computer algorithms. Its relationship to the state space representation of a linear system has already been established [Ref. 9,10] making it germane to consider that a system is identifiable by an equation of the form (1.1). The system identification problem thus entails identifying the coefficients  $a_i$  and  $b_i$ .

There are several methods for computing the coefficients,  $a_i$  and  $b_i$ , however, it is not the intent of this brief introduction to attempt to develop even the majority of them. Nevertheless, it is practical to present a referenced history of those methods encountered in this thesis.

Adaptive algorithms for the purpose of estimating the coefficients of (1.1) have always been of interest. Widrow, using a least means square error criterion and implementing the method of steepest descent, developed an adaptive algorithm which estimated the coefficients of the moving average process associated with equation (1.1) [Ref. 11]. That is, the moving average model,

$$y(k) = a'_0 u(k) + a'_1 u(k-1) + a'_2 u(k-2) + \dots \quad (1.2)$$

where,

$$a'_0 = a_0$$

$$a'_1 = (a_1 - a_0 b_1)$$

$$\begin{aligned}
 a'_2 &= (a_2 - a_0 b_2) - b_1 (a_1 - a_0 b_1) \\
 &\vdots \\
 &\vdots
 \end{aligned}$$

was identified. The LMS theory has since been extended to include block LMS filtering methods [Ref. 12,13,14] using various search techniques [Ref. 15,4(Chapter 5)]. These methods have enjoyed much popularity in the area of Linear Prediction and digital speech processing [Ref. 16,17].

The shortcoming of representing equation (1.1) by its equivalent moving average model (1.2) lies in the practical aspect of its implementation. That is, the infinite series represented by equation (1.2) must be truncated at some point resulting in an approximation which may not adequately represent equation (1.1). Hence, efforts to adaptively estimate the coefficients for both the autoregressive ( $b_i$ ) and moving average ( $a_i$ ) processes continued [Ref. 18,19,20] with various degrees of success. Feintuch's Adaptive Regressive LMS procedure [Ref. 18] is still a controversial issue [Ref. 19,20].

From yet a different direction, Anderson and Moore suggested that the Kalman filtering algorithms can be used as a means to identify the  $a_i$  and  $b_i$  coefficients<sup>1</sup> of equation (1.1) [Ref. 21, pp. 50-52]. This thesis exploits this application.

---

<sup>1</sup>Anderson and Moore in fact formulate the technique to compute the coefficients  $a_i, b_i$   $i = 1, 2, \dots, p$  where  $p = n+m$ .

## B. INVESTIGATIONS AND CONTRIBUTIONS

This thesis formulates an adaptive application of the Kalman filter to identify the coefficients of the general ARMA equation (1.1). It is shown that the LMS adaptive filter [Ref. 11] and the Adaptive Recursive LMS filter [Ref. 18] are (1) special cases of the Adaptive ARMA Kalman identifier and (2) sub-optimal with respect to the underlying least means square (LMS) error criterion upon which the LMS adaptive filter, the adaptive recursive filter and the Kalman filter are based. It is also demonstrated that the adaptive Kalman identifier has excellent convergence properties.

By comparison, it is noted that the Kalman algorithm accounts for measurement noise where the LMS algorithms do not, a heretofore unapproached problem by LMS algorithms. The results indicate that the suggested modification by Griffiths [Ref. 22] of the convergence factor,  $k_s$ , for the LMS adaptive filter is justified.

The application of the Kalman filter algorithm is extended to identification of the coefficients associated with a special case of the generalized non-linear ARMA model [Ref. 8]. The results of the non-linear simulations suggest a technique for determining when a potential non-linear system enters its non-linear operating regime. Such a technique can prove valuable when on-line performance evaluation of a known non-linear system is required.

Lastly, the connection between the Kalman filter algorithm and the lattice filter algorithm [Ref. 6] is made. An example

which demonstrates and compares the performance of both algorithms is given.

### C. ORGANIZATION

Chapter II presents and exploits the Kalman filter equations emphasizing its connections with the Yule-Walker and the discrete Weiner-Hopf equations. The theory is further developed to investigate the general ARMA case. Chapter III pursues the theoretical comparisons between the discrete Weiner-Hopf equation upon which the adaptive LMS filter is based and the MA form of the Kalman filter. The theoretical comparison between the Adaptive Recursive LMS filter and the general ARMA form of the Kalman filter is made in Chapter IV. The software methods by which linear and non-linear synthetic data is generated are discussed in Chapter V. A short user's description of the data processing programs and the options provided is given in Chapter VI. A non-linear application for the identification of the parameters of a non-linear plant is presented in Chapter VII followed by a discussion and presentation of the results in Chapter VIII.

## II. FORMULATION OF THE PROBLEM

Identifying the coefficients of the Autoregressive-Moving Average (ARMA) equation,

$$y(k) + b_1 y(k-1) + \dots + b_n y(k-n) = a_0 u(k) + a_1 u(k-1) + \dots + a_m u(k-m) \quad (2.1a)$$

$$y(k) = \sum_{i=0}^m a_i u(k-i) - \sum_{i=1}^n b_i y(k-i) \quad (2.1b)$$

can be formulated as an adaptive Kalman identification problem. That is, instead of using the well known Kalman Filter equations [Ref. 23,24] to recursively estimate the states of a system, one can utilize the Kalman filter equations to adaptively estimate the coefficients of either an Autoregressive (AR), Moving Average (MA), or Autoregressive-Moving Average (ARMA) process by proper definition of the quantities involved. We call this the Adaptive Kalman Identifier for obvious reasons.

### A. THE DETERMINISTIC CASE

Consider for the moment that the  $a_i$  and  $b_i$  are constant. Then by collecting a sufficient number of measurements of the input  $u(k)$  and the output  $y(k)$ , one can readily solve a set of linear equations for the  $a_i$  and  $b_i$  coefficients. The "sufficient number" that is needed is  $n+m+1$  which define the

$n+m+1$  linear equations which solve the  $n+m+1$  unknown coefficients. The matrix equation to be solved takes the form:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(m) \\ y(m+1) \\ \vdots \\ y(m+n) \end{bmatrix} = \begin{bmatrix} u(0) & 0 & \dots & 0 & | & 0 & \dots & \dots & 0 \\ u(1) & u(0) & & & | & y(0) & & & 0 \\ \vdots & \vdots & \vdots & \vdots & | & \vdots & \vdots & \vdots & \vdots \\ u(m) & u(m-1) & & u(0) & | & y(m-1) & y(m-2) & y(0) & 0 \\ \hline u(m+1) & & & u(1) & | & y(m) & y(m-1) & y(1) & 0 \\ \vdots & & & \vdots & | & \vdots & \vdots & \vdots & \vdots \\ u(m+n) & \dots & & u(m) & | & y(n-1) & y(n-2) & & y(0) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \\ b_1 \\ b_2 \\ \vdots \\ \cdot \end{bmatrix} \quad (2.2)$$

$$\underline{y} = T \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} \quad (2.3)$$

It is interesting to note that the  $(n+m+1) \times (n+m+1)$  matrix,  $T$ , is block symmetric. The solution to (2.3) is readily apparent, namely,

$$\begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} = T^{-1} \underline{y} \quad (2.4)$$

where  $T^{-1}$  is the inverse matrix of  $T$ . Since we have assumed that the elements of  $T$  are perfect noiseless measurements, and that we somehow know a-priori the number of unknown

coefficients, then  $T$  is of full rank, namely  $\text{rank}(T) = n+m+1$ , and the inverse of  $T$  exists.

#### B. THE NONDETERMINISTIC CASE

Rarely can the coefficients be modeled so ideally. A more prudent and realistic model admits that the ARMA equation coefficients are subject to random perturbations. Further, it can be said that in general measurement devices introduce noise into the measurement data. Hence, the measurement model should reflect this fact. Developing the Adaptive Kalman Identifier along the same lines as [Ref. 21] we let,

$$a_i(k+1) = a_i(k) + w_i(k) \quad i = 0, 1, 2, \dots, m \quad (2.5a)$$

$$b_j(k+1) = b_j(k) + w_j(k) \quad j = 1, 2, \dots, n \quad (2.5b)$$

where the  $\{w_i(k), w_j(k)\}$  are samples from a zero mean, white, gaussian random process. Additionally, we assume that the noise sources are uncorrelated,

$$E[w_{a_r}(k)w_{a_s}(k)] = 0 \quad \text{for } r \neq s \quad (2.6a)$$

$$E[w_{b_r}(k)w_{b_s}(k)] = 0 \quad \text{for } r \neq s \quad (2.6b)$$

$$E[w_{a_r}(k)w_{b_s}(k)] = 0 \quad \text{for all } r, s \quad (2.6c)$$

Similarly, allowing for noisy measurement devices, the measurement equation (2.1b), is modelled as,

$$y(k) = \sum_{i=0}^m a_i u(k-i) - \sum_{i=1}^n b_i y(k-i) + v(k) \quad (2.7)$$

where the  $\{v(k)\}$  are samples from a zero mean, white, gaussian random process. It is also assumed that,

$$E[w_i(k) v(k)] = 0 \text{ for all } i \quad (2.8a)$$

$$E[w_j(k) v(k)] = 0 \text{ for all } j \quad (2.8b)$$

That is, it is assumed that the noise perturbations associated with one coefficient are independent of the noise perturbations associated with any other coefficient, and that the measurement noise and the coefficient perturbations are independent.

At this point it is necessary to use judgment and experience and utilize all the information known about the physical system which equation (2.7) represents to assign variances for the random processes  $\{w_i(k)\}$ ,  $\{w_j(k)\}$  and  $\{v(k)\}$ . Let

$$Q = E \left\{ \begin{array}{c} \begin{bmatrix} w_i(k) \\ w_j(k) \end{bmatrix} \\ [w_i(k) \ w_j(k)] \end{array} \right\} \quad (2.9a)$$

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \quad (2.9b)$$

where,  $Q_1 = \text{diag} (\sigma_{w_i}^2)$ ,  $Q_2 = \text{diag} (\sigma_{w_j}^2)$  and,

$$R = E[v(k) v(k)] \quad (2.9c)$$

and  $E[x]$  is the expected or average value of  $x$ .

Instead of using  $x$ 's denoting the commonly used notation for state variables, we retain the flavor of the problem by using the  $a_i$ 's and  $b_i$ 's as the "states" of the Adaptive Kalman Identifier. It is hoped that a more meaningful understanding of the Adaptive Kalman Identifier may thus be gained. Therefore, define the state vector,

$$\begin{bmatrix} a_0(k) \\ a_1(k) \\ \cdot \\ \cdot \\ a_m(k) \\ \text{-----} \\ b_1(k) \\ b_2(k) \\ \cdot \\ \cdot \\ \cdot \\ b_n(k) \end{bmatrix} = \begin{bmatrix} \underline{a} \\ \text{-----} \\ \underline{b} \end{bmatrix} \quad (2.10)$$

Combining equations (2.5) and (2.10), we have the discrete, first order Gauss-Markov process [Ref. 3],

$$\begin{bmatrix} \underline{a}(k+1) \\ \text{-----} \\ \underline{b}(k+1) \end{bmatrix} = \begin{bmatrix} \underline{a}(k) \\ \text{-----} \\ \underline{b}(k) \end{bmatrix} + \underline{w}(k) \quad (2.11)$$

where,

$$\underline{w}(k) = \begin{bmatrix} \underline{w}_i(k) \\ \text{-----} \\ \underline{w}_j(k) \end{bmatrix} . \quad (2.12)$$

Note that  $\underline{w}(k)$  is an  $n+m+1$  vector whose elements are a concatenation of the noise sequences associated with the  $a_i$  and  $b_i$ . To complete the Adaptive Kalman Identifier formulation we define the measurement vector,

$$H(k) = [u(k) \ u(k-1) \ \dots \ u(k-m) \ -y(k-1) \ \dots \ -y(k-n)] \quad (2.13)$$

and its associated measurement equation,

$$y(k) = H(k) \begin{bmatrix} \underline{a}(k) \\ \text{-----} \\ \underline{b}(k) \end{bmatrix} + v(k). \quad (2.14)$$

Then the solution to the Adaptive Kalman Identifier problem [Ref. 21] is,

$$\begin{bmatrix} \hat{\underline{a}}(k+1|k) \\ \text{-----} \\ \hat{\underline{b}}(k+1|k) \end{bmatrix} = [I - K(k)H(k)] \begin{bmatrix} \hat{\underline{a}}(k|k-1) \\ \text{-----} \\ \hat{\underline{b}}(k|k-1) \end{bmatrix} + K(k)y(k) \quad (2.15a)$$

$$K(k) = P(k|k-1) H^T(k) [H(k) P(k|k-1) H^T(k) + R]^{-1} \quad (2.15b)$$

$$P(k+1|k) = P(k|k-1) - K(k)H(k)P(k|k-1) + Q. \quad (2.15c)$$

Equations (2.15) are initialized by assuming an initial value for the coefficients (2.10) and assigning to our assumption a measure of our confidence in the initial guess. That is, we pick the values,

$$\begin{bmatrix} \hat{\underline{a}}(0|-1) \\ \text{-----} \\ \hat{\underline{b}}(0|-1) \end{bmatrix} \text{ and } P(0|-1) \quad (2.16)$$

where  $P(0|-1)$  is defined as the error covariance of the coefficients,

$$P(k|k-1) = E \left\{ \begin{bmatrix} \underline{a}(k) \\ \underline{b}(k) \end{bmatrix} - \begin{bmatrix} \hat{\underline{a}}(k|k) \\ \hat{\underline{b}}(k|k) \end{bmatrix} \begin{bmatrix} \underline{a}(k) \\ \underline{b}(k) \end{bmatrix} - \begin{bmatrix} \hat{\underline{a}}(k|k) \\ \hat{\underline{b}}(k|k) \end{bmatrix}^T \right\} \quad (2.17)$$

for  $k = 0$ . Equation (2.15b) is generally referred to as the Kalman gain. Two special cases are of interest: case (1);  $a_i = 0$  for all  $i$ , and case (2);  $b_i = 0$  for all  $i$ .

1. Autoregressive Form ( $a_i = 0$ )

For case 1, equation (2.7) takes the form,

$$y(k) = - \sum_{i=1}^n b_i y(k-i) + v(k). \quad (2.18)$$

This is a recursive equation stating that the present output is a linear combination of past outputs corrupted with additive gaussian noise,  $v(k)$ . Equation (2.18) is more formally recognized as an autoregressive process of order  $n$  [Ref. 25, 26]. Writing the Kalman solution for the coefficients of the AR process,

$$\hat{\underline{b}}(k+1|k) = [I - K(k)H(k)] \hat{\underline{b}}(k|k-1) + K(k)y(k) \quad (2.19a)$$

$$K(k) = P(k|k-1) H^T(k) [H(k)P(k|k-1) H^T(k) + R]^{-1} \quad (2.19b)$$

$$P(k+1|k) = P(k|k-1) - K(k)H(k)P(k|k-1) + Q. \quad (2.19c)$$

Alternate forms of the Kalman equations given by Maybeck [Ref. 27] are,

$$\begin{aligned} \hat{\underline{b}}(k+1|k) &= P(k|k-1) P(k|k-1)^{-1} \hat{\underline{b}}(k|k-1) \\ &\quad + P(k+1|k) H^T(k) R^{-1} y(k) \end{aligned} \quad (2.20a)$$

$$P(k+1|k) = (P(k|k-1))^{-1} + H^T(k) R^{-1} H(k)^{-1}. \quad (2.20b)$$

We can model the fact that we have no a-priori knowledge about the initial values of the coefficients by letting,

$$[P(0|-1)]^{-1} \approx 0. \quad (2.21)$$

Further, if we remain ignorant and totally doubt our previous estimate, then  $P(k|k-1)$  is modeled as,

$$[P(k|k-1)]^{-1} \approx 0. \quad (2.22)$$

Equations (2.20) reduce to,

$$\hat{\underline{b}}(k+1|k) = [H(k) R^{-1} H^T(k)]^{-1} H(k) R^{-1} y(k), \quad (2.23)$$

the weighted least squares estimate, previously encountered by Swerling [Ref. 28,29,30], of the coefficients,  $\underline{b}$ . Carrying the analysis further and letting  $R^{-1} = 1$ , we have the Penrose pseudo-inverse solution [Ref. 31,32],

$$\hat{\underline{b}}(k+1|k) = [H(k) H^T(k)]^{-1} H(k) y(k) \quad (2.24)$$

Swerling [Ref. 28] has shown that if the weighting matrix in (2.23) is not the inverse of the covariance matrix of the measurement errors, then the accuracy of the estimated coefficients (2.24) will be degraded.

The aforementioned notwithstanding, we press further into the analysis of equation (2.24). The product  $H(k)H^T(k)$  can be written as,

$$H(k)H^T(k) = \begin{bmatrix} y(k-1) \\ y(k-2) \\ \vdots \\ y(k-n) \end{bmatrix} [y(k-1) \ y(k-2) \ \dots \ y(k-n)] \quad (2.25)$$

$$H(k)H^T(k) = \begin{bmatrix} y^2(k-1) & y(k-1)y(k-2) & \dots & y(k-1)y(k-n) \\ y(k-2)y(k-1) & y^2(k-2) & & \vdots \\ \vdots & \vdots & & \vdots \\ y(k-n)y(k-1) & \dots & \dots & y(k-n)y(k-n) \end{bmatrix} \quad (2.26)$$

As we let  $k \rightarrow \infty$  the expected value of (2.26) becomes the autocorrelation matrix,

$$E\{\lim_{k \rightarrow \infty} H(k) H^T(k)\} = R_{yy}(k) \quad (2.27)$$

where,

$$R_{YY}(k) = \begin{bmatrix} R_{YY}(0) & R_{YY}(1) & \dots & R_{YY}(n) \\ R_{YY}(-1) & R_{YY}(0) & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R_{YY}(-n) & \dots & \dots & R_{YY}(0) \end{bmatrix} \quad (2.28)$$

Similarly, the product  $H(k)y(k)$  as  $k \rightarrow \infty$  becomes

$$r_{YY}(k) = \begin{bmatrix} R_{YY}(-1) \\ R_{YY}(-2) \\ \cdot \\ \cdot \\ R_{YY}(-n) \end{bmatrix} \quad (2.29)$$

The steady state solution for the estimate of the coefficients  $\hat{\underline{b}}(k+1|k)$  is,

$$\hat{\underline{b}}_{ss}(k+1|k) = R_{YY}^{-1}(k) r_{YY}(k) \quad (2.30)$$

Equation (2.30) is one of the starting points from which Perry [Ref. 6] develops his Lattice modeling algorithms. Equation (2.30) can also be recognized as the solution to the Yule-Walker or Normal equations [Ref. 26].

## 2. Moving Average Form ( $b_i = 0$ )

Referring once again to equation (2.7) and setting the coefficients  $b_i = 0$  for all  $i$  we have case (2),

$$y(k) = \sum_{i=0}^m u(k-i) + v(k) \quad (2.31)$$

Since  $v(k)$  is by definition noise associated with the measurement, we can combine  $y(k)$  and  $v(k)$  such that,

$$z(k) = y(k) - v(k) = \sum_{i=0}^m a_i u(k-i). \quad (2.32)$$

Equation (2.32) simply states that the present measurement is a linear combination of past and present inputs or by definition, a Moving Average (MA) process [Ref. 25,33]. The Adaptive Kalman Identifier estimate for the coefficients of the MA process is as before,

$$\hat{\underline{a}}(k+1|k) = [I - K(k)H(k)] \hat{\underline{a}}(k+1|k) + K(k)z(k) \quad (2.33a)$$

$$K(k) = P(k|k-1) H^T(k) [H(k)P(k|k-1) H^T(k) + R]^{-1} \quad (2.33b)$$

$$P(k+1|k) = P(k|k-1) - K(k)H(k)P(k|k-1) + Q. \quad (2.33c)$$

Using the alternate forms of the above equations we arrive at,

$$\begin{aligned} \hat{\underline{a}}(k+1|k) &= [P(k+1|k) [P(k|k-1)]^{-1}] \hat{\underline{a}}(k|k-1) \\ &\quad + [P(k+1|k) H^T(k) R^{-1}] z(k) \end{aligned} \quad (2.34a)$$

$$P(k+1|k) = [[P(k|k-1)]^{-1} + H^T(k) R^{-1} H(k)]^{-1}. \quad (2.34b)$$

Arguing as we did for the autoregressive case, no a-priori knowledge about the initial values of the moving average coefficients, implies that,

$$[P(0|-1)]^{-1} \approx 0. \quad (2.35)$$

And even if after we processed one measurement we still admitted no knowledge as to the accuracy of the previous estimate, we imply that,

$$[P(k|k-1)]^{-1} \approx 0. \quad (2.36)$$

Substituting these implications into equations (2.34) our estimate becomes,

$$\hat{\underline{a}}(k+1|k) = [H(k) R^{-1} H^T(k)]^{-1} H(k) R^{-1} z(k) \quad (2.37)$$

the by now familiar weighted least squares estimate [Ref. 28, 29,30]. If once again we allow  $R^{-1}$  to be unity, let  $k$  tend toward infinity and take the expectation of equation (2.37), one arrives at the discrete form of the Wiener-Hopf equation [Ref. 11], namely

$$\hat{\underline{a}}(k+1|k) = R_{uu}^{-1}(k) r_{uz}(k) \quad (2.38)$$

where it can easily be shown that,

$$R_{uu}(k) = \begin{bmatrix} R_{uu}(0) & R_{uu}(-1) & \dots & R_{uu}(-m) \\ R_{uu}(-1) & R_{uu}(0) & \dots & R_{uu}(1-m) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R_{uu}(-m) & \dots & \dots & R_{uu}(0) \end{bmatrix} \quad (2.39)$$

and,

$$R_{uy}(k) = \begin{bmatrix} R_{uy}(0) \\ R_{uy}(-1) \\ \cdot \\ \cdot \\ R_{uy}(-m) \end{bmatrix} \quad (2.40)$$

Equation (2.38) provided another point of departure from which Perry [Ref. 6] develops the MA Lattice modeling algorithms. Appendix A develops the Wiener-Hopf equation otherwise known as the all-zero model from yet another approach.

The all-zero model is fundamental to Widrow least mean square (LMS) adaptive filters [Ref. 11] and linear prediction theory [Ref. 16].

### 3. Autoregressive-Moving Average Form

Returning to the alternate form (equations 2.20) of the Kalman Filter equations (2.19) the development of the Autoregressive Moving Average (ARMA) Adaptive Kalman Identifier follows. The estimate for the  $a_i$  and  $b_i$  coefficients is,

$$\begin{bmatrix} \hat{\underline{a}}(k+1|k) \\ \text{-----} \\ \hat{\underline{b}}(k+1|k) \end{bmatrix} = [P(k+1|k) [P(k|k-1)]^{-1}] \begin{bmatrix} \hat{\underline{a}}(k|k-1) \\ \text{-----} \\ \hat{\underline{b}}(k|k-1) \end{bmatrix} + [P(k+1|k) H^T(k) R^{-1}] y(k) \quad (2.41a)$$

$$P(k+1|k) = [(P(k|k-1))^{-1} + H^T(k) R^{-1} H(k)]^{-1} \quad (2.41b)$$

Progressing in the same manner as in the previous two cases, we assume,

$$[P(0|-1)]^{-1} \approx 0 \quad (2.42a)$$

$$[P(k|k-1)]^{-1} \approx 0 . \quad (2.42b)$$

The estimate then becomes,

$$\begin{bmatrix} \hat{\underline{a}}(k+1|k) \\ \text{-----} \\ \hat{\underline{b}}(k+1|k) \end{bmatrix} = [H^T(k) R^{-1} H(k)]^{-1} H(k) R^{-1} y(k) . \quad (2.43)$$

Taking equation (2.43) a step further by letting  $R^{-1}$  be unity, letting  $k$  approach infinity and taking the expectation we have,

$$\begin{bmatrix} \hat{\underline{a}}(k+1|k) \\ \text{-----} \\ \hat{\underline{b}}(k+1|k) \end{bmatrix} = \begin{bmatrix} R_{nu}(k) & -R_{uy}(k-1) \\ \text{-----} \\ -R_{uy}^T(k-1) & R_{yy}(k) \end{bmatrix}^{-1} \begin{bmatrix} R_{uy}(k) \\ \text{-----} \\ R_{yy}(k) \end{bmatrix} \quad (2.44)$$

Note that the time varying measurement vector,  $H(k)$ , is of the form

$$H(k) = [u(k) \ u(k-1) \ \dots \ u(k-n) \ | \ -y(k-1) \ -y(k-2) \ \dots \ -y(k-n)] \quad (2.45)$$

Shown in detail, equation (2.44) has the characteristic form (equation 2.46, next page). It is the Toeplitz and symmetric nature of equation (2.46) that is exploited by the Levinson [Ref. 34] and Lattice [Ref. 35] algorithms.

$$\begin{bmatrix} \hat{a}(k+1|k) \\ \hline \hat{b}(k+1|k) \end{bmatrix}$$

$$= \begin{bmatrix} R_{uu}(0) & R_{uu}(-1) & \dots & R_{uu}(-m) & -R_{uy}(-1) & -R_{uy}(-2) & \dots & -R_{uy}(-n) \\ R_{uu}(-1) & R_{uu}(0) & \dots & R_{uu}(1-m) & -R_{uy}(0) & -R_{uy}(-1) & \dots & -R_{uy}(1-n) \\ \vdots & \vdots \\ R_{uu}(1-m) & \dots & R_{uu}(0) & -R_{uy}(m-1) & \dots & \dots & -R_{uy}(m-n) \\ \hline -R_{yu}(-1) & -R_{yu}(0) & \dots & -R_{yu}(m-1) & R_{yy}(0) & R_{yy}(-1) & \dots & R_{yy}(-n) \\ -R_{yu}(-2) & -R_{yu}(-1) & \dots & \dots & R_{yy}(-1) & R_{yy}(0) & \dots & \dots \\ \vdots & \vdots \\ -R_{yu}(-n) & \dots & -R_{yu}(m-n) & R_{yy}(-n) & \dots & \dots & R_{yy}(0) \end{bmatrix}^{-1} \begin{bmatrix} R_{uy}(0) \\ R_{uy}(1) \\ \vdots \\ R_{uy}(m) \\ R_{yy}(1) \\ R_{yy}(2) \\ \vdots \\ R_{yy}(n) \end{bmatrix}$$

(2.46)

The Adaptive Kalman ARMA modeling technique can be best visualized by a block diagram. Referring to Figure 2.1, the unknown system is excited by a white, zero mean, gaussian noise sequence of sample values from a random process. The input and its associated output are then passed through M and N delays respectively in serial form. The parallel inputs (M+1) and the outputs (N) are concatenated to form the measurement vector  $H(k)$  which is represented by equation (2.13). The inputs N and M are selected a-priori as the model orders for the Autoregressive and Moving Average processes one desires to identify. It can be easily seen that the remainder of the figure simply implements equation (2.15a). The outputs of the Adaptive Kalman Identifier are an estimate of the coefficients,

$$\begin{bmatrix} \hat{\underline{b}}(k|k) \\ \text{-----} \\ \hat{\underline{a}}(k|k) \end{bmatrix}$$

and a one step prediction,  $\hat{y}(k|k)$ .

### C. OBSERVATIONS

In this section it has been shown that a direct connection can be established between the Adaptive Kalman Identifier and the Yule-Walker equations associated with an AR process. Secondly, the steady state Adaptive Kalman Identifier closely resembles the discrete Weiner-Hopf equation associated with the MA process when the measurement matrix,  $H(k)$ , is of the form,

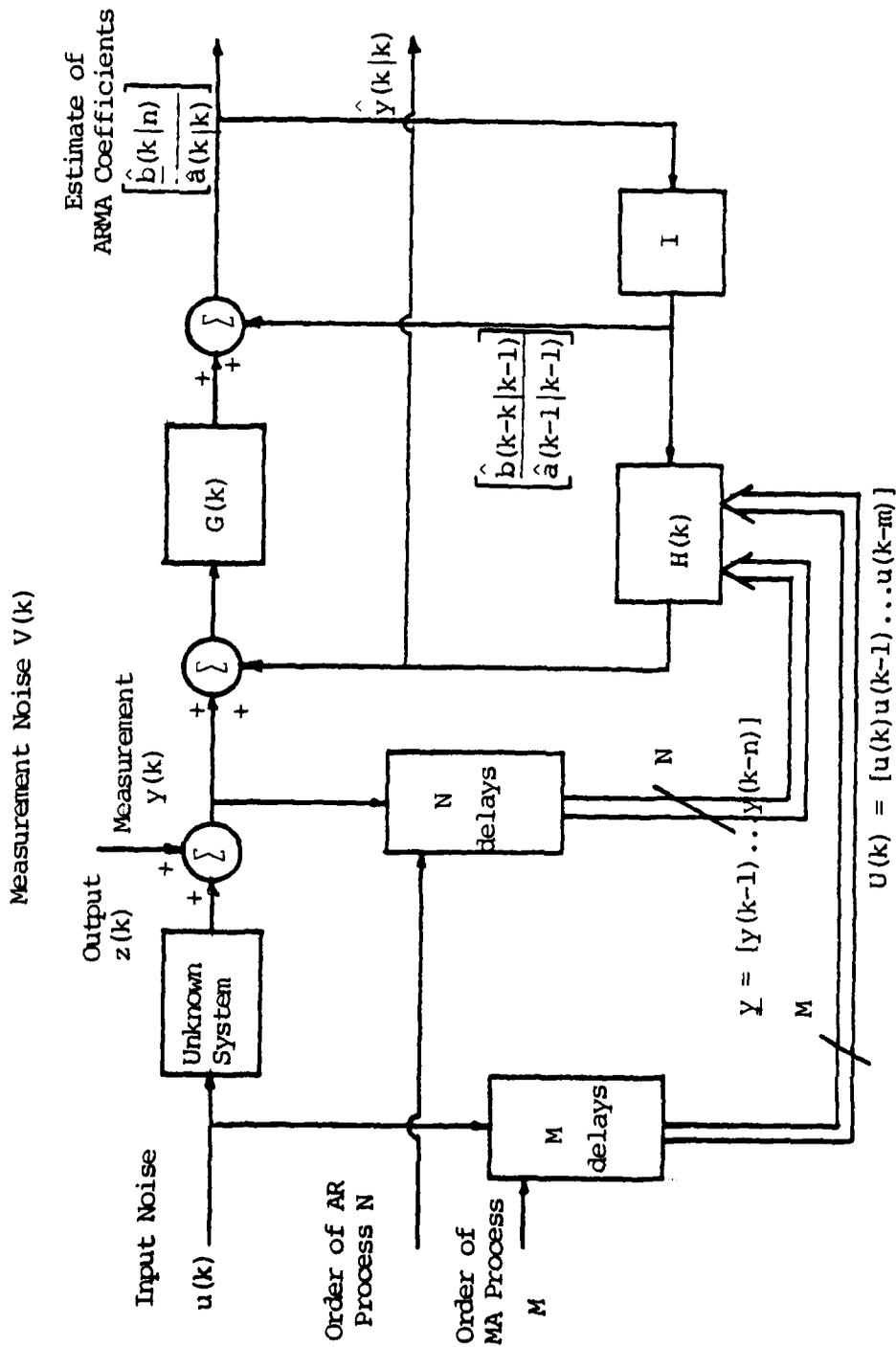


Figure 2.1. Adaptive Kalman Identifier

$$H(k) = [u(k) \ u(k-1) \ u(k-2) \ \dots \ u(k-m)]$$

And, thirdly under the same assumptions as in the previous two cases, the Adaptive Kalman ARMA Identifier is similar to the form Perry [Ref. 6] exploits using Lattice modeling.

### III. COMPARISON BETWEEN THE ADAPTIVE MA KALMAN IDENTIFIER AND THE WIDROW LMS ADAPTIVE FILTER

#### A. PRELIMINARIES

It is instructive to investigate the similarities between the Adaptive Kalman Identifier and the Widrow LMS Adaptive Filter when the concepts are applied to system identification. Basically, the LMS algorithm implementation of system identification considers a block diagram as is shown in Figure 3.1. The output,  $y(k)$ , of the Adaptive filter is simply a weighted linear combination of the past and present known inputs. The same input is fed into both the unknown system to be identified and the adaptive filter. The output of the unknown system is designated the desired response,  $d(k)$ , from which an error signal,  $\epsilon(k)$ , is derived. The error signal,  $\epsilon(k)$ , provides the criterion through which the weights,  $w_i$ , are adjusted such that the error,  $\epsilon(k+1)$ , is driven toward its minimum. A more detailed analysis of the operation of the LMS algorithm can be found in Ref. 11.

The weights  $w_i$  are adjusted from time step to time step in the following manner,

$$\underline{w}(k+1) = \underline{w}(k) - 2k_s \epsilon(k) \underline{X}(k) \quad (3.1)$$

where we define the following quantities,

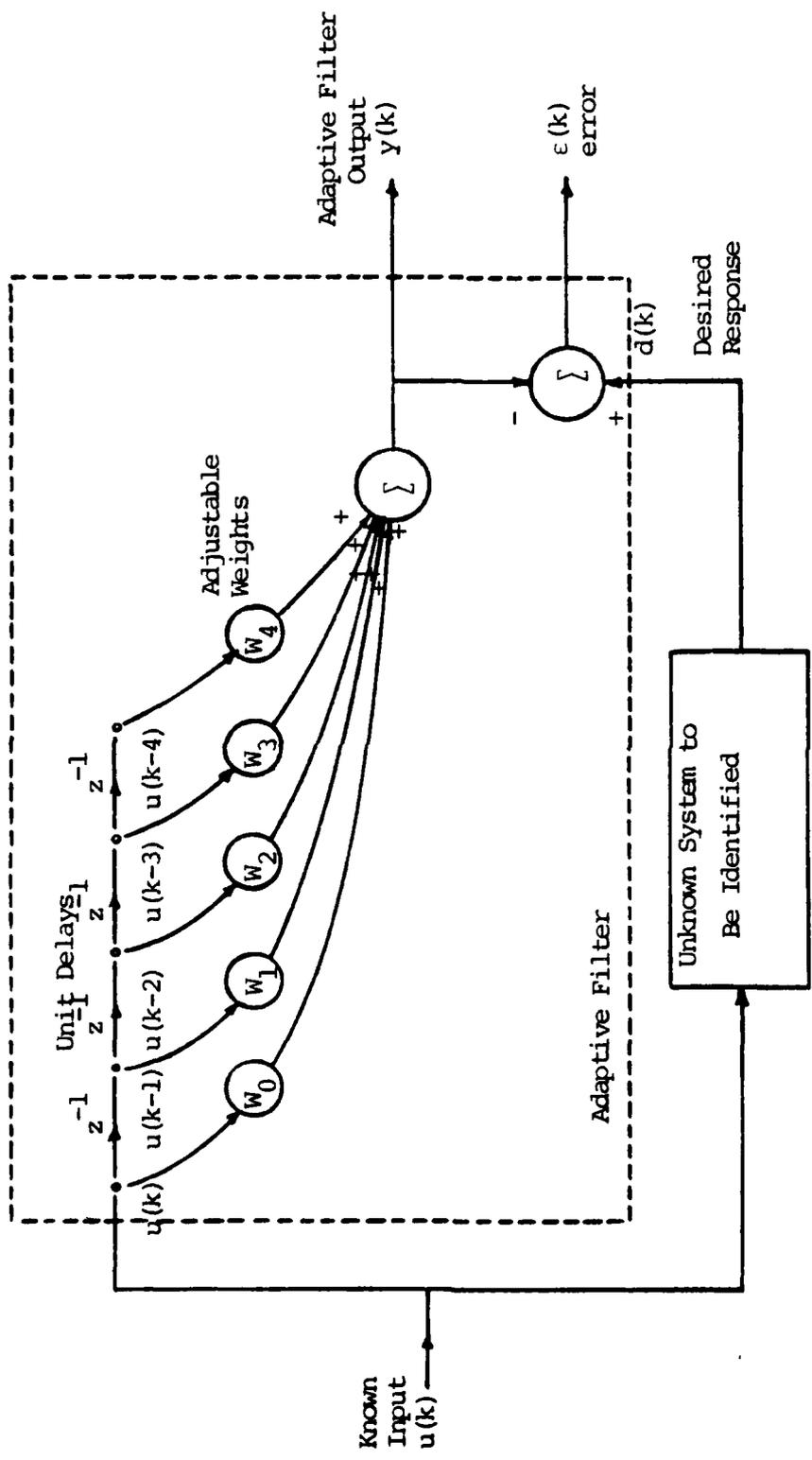


Figure 3.1. LMS Adaptive Filter Implementation of System Identification

$$\underline{w}(k) = \begin{bmatrix} w_0(k) \\ w_1(k) \\ \cdot \\ \cdot \\ w_m(k) \end{bmatrix} \quad (3.2)$$

the weight vector at time step  $k$ ,

$$\underline{X}(k) = \begin{bmatrix} u(k) \\ u(k-1) \\ \cdot \\ \cdot \\ u(k-m) \end{bmatrix} \quad (3.3)$$

the input signal vector at time step  $k$ ,

$$\varepsilon(k) = d(k) - y(k) \quad (3.4)$$

$$y(k) = \underline{w}^T(k) \underline{X}(k) \quad (3.5)$$

the error,  $\varepsilon(k)$ , and the output,  $y(k)$ , at time step  $k$ , and,  $k_s$ , a scalar constant controlling the rate of convergence and the stability of the adaptive filter.

#### B. THE COMPARISON

Substituting (3.4) and (3.5) into (3.1) gives,

$$\underline{w}(k+1) = \underline{w}(k) - 2k_s \underline{X}(k) [d(k) - \underline{w}^T(k) \underline{X}(k)]. \quad (3.6)$$

Recalling the MA form of the Adaptive Kalman Identifier, equation (2.33a), rewritten here for convenience as

$$\hat{\underline{a}}(k+1|k) = \hat{\underline{a}}(k|k-1) + K(k) [z(k) - H(k) \hat{\underline{a}}(k|k-1)], \quad (3.7)$$

one can make the following associations:

$$\hat{\underline{a}}(k+1|k) \langle \Rightarrow \underline{w}(k+1) \quad (3.8a)$$

$$\hat{\underline{a}}(k|k-1) \langle \Rightarrow \underline{w}(k) \quad (3.8b)$$

$$z(k) \quad \langle \Rightarrow \underline{d}(k) \quad (3.8c)$$

$$H^T(k) \quad \langle \Rightarrow \underline{x}(k) \quad (3.8d)$$

$$K(k) \quad \langle \Rightarrow -2k_s \underline{x}(j) \quad (3.8e)$$

Recall that for the MA form of the AKI, the measurement vector  $H(k)$  represents a vector of past and present inputs. Namely,

$$H(k) = [u(k) \ u(k-1) \ \dots \ u(k-m)] . \quad (3.9)$$

Therefore the associations (3.8a)-(3.8d) are straightforward. However, it is not so clear as to what is meant by the association (3.8e). Digressing a moment to present an equivalent expression [Ref. 27] for the Kalman gain,  $K(k)$ ,

$$K(k) = P(k|k) H^T(k) R^{-1} \quad (3.10)$$

and substituting (3.10) into the association (3.8e) and entertaining the conjecture that the quantities are equivalent

under certain conditions, we have:

$$P(k|k) H^T(k) R^{-1} = -2k_s \underline{X}(k) . \quad (3.11)$$

The conditions alluded to are (1) that the Adaptive Kalman Identifier is in steady state and (2) that the statistics of the input forcing function are stationary. Denoting the steady state error covariance,  $P(k|k)$ , as  $P_\infty$ , equation (3.11) can be solved for  $k_s$ ,

$$k_s I = -\frac{1}{2} R^{-1} P_\infty . \quad (3.12)$$

Invoking the entire Kalman gain history in its more popular form of equation (2.33b) and equating it to the Widrow gain (3.8e), we instead arrive at,

$$-2k_s \underline{X}(k) = P(k|k-1) H^T(k) [H(k) P(k|k-1) H^T(k) + R]^{-1} \quad (3.13a)$$

$$H^T(k) = \underline{X}(k) . \quad (3.13b)$$

Solving for the convergence factor,  $k_s$ , we obtain,

$$-2k_s \underline{X}(k) \underline{X}^T(k) = P(k|k-1) \underline{X}(k) [\underline{X}^T(k) P(k|k-1) \underline{X}(k) + R]^{-1} \underline{X}^T(k)$$

$$-2k_s I = P(k|k-1) \underline{X}(k) [\underline{X}^T(k) P(k|k-1) \underline{X}(k) + R]^{-1} \underline{X}^T(k) [\underline{X}(k) \underline{X}^T(k)]^{-1} .$$

But  $[\underline{X}^T(k) P(k|k-1) \underline{X}(k) + R]^{-1}$  is a scalar. Therefore,

$$-2k_s I = \frac{P(k|k-1)}{\underline{X}^T(k) P(k|k-1) \underline{X}(k) + R} \quad (3.14)$$

Griffiths [Ref. 22] suggests that the convergence factor  $k_s$ , which he denotes as  $\mu(n)$ , should be chosen such that,

$$\mu(n) = \frac{\alpha}{L \hat{\sigma}_x^2(n)} \quad (3.15)$$

where  $0 < \alpha < 1$  is a normalized adaptive stepsize parameter and the term  $\hat{\sigma}_x^2(n)$  is an estimate of the input power level which may be computed using a geometrically-weighted average for an L weight adaptive filter,

$$\hat{\sigma}_x^2(n) = (1 - \frac{\alpha}{L}) \hat{\sigma}_x^2(n-1) + \frac{\alpha}{L} x^2(n) . \quad (3.16)$$

Comparing equation (3.14) and (3.15), one observes that if  $P(k, k-1)$  is equal to the identity matrix then we obtain a term proportional to the input power. Further if measurement noise is considered negligible,  $R \rightarrow 0.0$ , then 3.14 and 3.15 are essentially equivalent. The salient point to make, however, is that the choice of the convergence constant,  $k_s$ , in the LMS algorithm provides no clue as to how to deal with measurement noise whereas the Adaptive MA Kalman Identifier explicitly incorporates measurement error into its algorithm.

### C. OBSERVATIONS

It has been known that the LMS algorithm was suboptimal since the actual gradient of equation (3.1) is replaced by the approximation,  $2\varepsilon(k) x(k)$ , [Ref. 24]. However, the degree of suboptimality is difficult to quantify. In previous linear prediction research no mention was made regarding the

role of measurement noise. Equation (3.14) gives us some insight into wiser selections of the rate of convergence constant,  $k_s$ , since it takes into account the effects of measurement noise. Further, it appears that the LMS adaptive filter is a special case of the Adaptive MA Kalman Identifier.

#### IV. COMPARISON BETWEEN THE ADAPTIVE ARMA KALMAN IDENTIFIER AND THE ADAPTIVE RECURSIVE LMS FILTER

This chapter endeavors to investigate the similarities and differences between the Adaptive ARMA Kalman Identifier and the Adaptive Recursive LMS Filter [Ref. 18]. LMS filters [Ref. 11] have enjoyed much popularity in the recent past due to their ease of implementation, simple unimodal algorithm, robustness and ability to "adapt" to the unknown statistics of the signal environment. Being transversal in nature, they have a finite impulse response being able to produce only zeros in the input/output transfer function. One may, however, decide to model the transfer function,

$$H_1(z) = \frac{1}{1 - .9z^{-1}} \quad (4.1)$$

using a transversal filter, only to realize that a large number of delays are required in order to arrive at an adequate approximation. The germane point which is being made is that one weighted feedback tap can realize an infinite string of feed-forward coefficients. Moreover, it is very desirable to adjust the feedback weights adaptively in some optimal fashion to the statistics of the signal environment.

##### A. PRELIMINARIES

The approach taken by Feintuch [Ref. 18] is patterned after the analysis first presented by Widrow [Ref. 11]. A summary of Feintuch's derivation will be presented here with emphasis on its application to ARMA modeling.

Recall the general ARMA equation (2.1b),

$$y(k) = \sum_{i=0}^m a_i u(k-i) + \sum_{i=1}^n b_i y(k-i) \quad (4.2a)$$

which can be rewritten in vector notation,

$$y(k) = \underline{a}^T \underline{u} + \underline{b}^T \underline{y} \quad (4.2b)$$

where,

$$\underline{a}^T = [a_0 \ a_1 \ \dots \ a_m] \quad (4.3a)$$

$$\underline{b}^T = [b_1 \ b_2 \ \dots \ b_n] \quad (4.3b)$$

$$\underline{y} = [y(k-1) \ y(k-2) \ \dots \ y(k-n)]^T \quad (4.3c)$$

$$\underline{u} = [u(k) \ u(k-1) \ \dots \ u(k-m)]^T \quad (4.3d)$$

Given that (4.2) is the assumed mathematical description of the unknown system where  $\underline{u}$  and  $\underline{y}$ , the input and output data sequences, are known, then by solving (4.2) for  $\underline{a}$  and  $\underline{b}$ , identification of the unknown system can be made. The LMS algorithms, in general, employ a "desired" signal  $d(k)$  with which to "train" the adaptive filter. If the desired signal was assumed to be the response of some unknown system to a known input signal, then the algorithm presented by Feintuch [Ref. 18] can be used as a means by which to identify the system parameters.

The first step in the Recursive LMS derivation is to form the error between the desired signal,  $d(k)$ , and the output

of the filter,  $y(k)$ ,

$$e(k) = d(k) - y(k) = d(k) - \underline{a}^T \underline{u} - \underline{b}^T \underline{y} \quad (4.4)$$

Forming the square of (4.4) and taking the expectation, we have the mean square error representation of the filter,

$$\begin{aligned} E\{e^2(k)\} &= E\{d^2(k)\} + \underline{a}^T R_{uu} \underline{a} + \underline{b}^T R_{yy} \underline{b} \dots \\ &\quad - 2\underline{a}^T R_{du} - 2\underline{b}^T R_{dy} + 2\underline{a}^T R_{uy} \underline{b} \end{aligned} \quad (4.5)$$

where,

$$R_{uu}(k) = E\{u(k) u^T(k)\} \quad (4.6a)$$

$$R_{yy}(k-1) = E\{y(k-1) y^T(k-1)\} \quad (4.6b)$$

$$R_{du}(k) = E\{d(k) u(k)\} \quad (4.6c)$$

$$R_{dy}(k) = E\{d(k) y(k)\} \quad (4.6d)$$

$$R_{uy}(k-1) = E\{u(k-1) y(k-1)\} \quad (4.6e)$$

It is assumed that the statistics (4.6) are constants allowing the gradient of (4.5) to be taken with respect to the  $a_i$  and  $b_i$ . This assumption does not stretch the theory since in practice one uses an input test signal with stationary characteristics and if the unknown system's output process is not stationary, then no identification of the system parameters can be made. The respective gradients are,

$$\frac{\partial}{\partial \underline{a}} [E\{e^2(k)\}] = 2R_{uu}(k)\underline{a} - 2R_{du}(k) + 2R_{uy}(k)\underline{b} \quad (4.7a)$$

$$\frac{\partial}{\partial \underline{b}} [E\{e^2(k)\}] = 2R_{yy}(k)\underline{b} - 2R_{dy}(k) + 2R_{uy}^T(k)\underline{a} . \quad (4.7b)$$

Since the second order statistics are assumed to be known, equations (4.7) can be solved for  $\underline{a}$  and  $\underline{b}$  by setting the gradients equal to zero. In matrix partitioned form we have,

$$\begin{bmatrix} R_{uu}(k) & R_{uy}(k) \\ R_{uy}^T(k) & R_{yy}(k) \end{bmatrix} \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} = \begin{bmatrix} R_{du}(k) \\ R_{dy}(k) \end{bmatrix} . \quad (4.8)$$

At this point we digress momentarily to mention that Johnson and Larimore [Ref. 19] and Widrow and McCool [Ref. 20] agree with Feintuch's derivation. The following steps, however, are controversial [Ref. 19,20].

Feintuch continues and states that in general, the statistics involved in (4.8) are not available a-priori. One method of estimating the statistics is to make the filter adaptive in an iterative fashion using the method of steepest descent. The method of steepest descent employs an algorithm of the form,

$$\underline{a}(k+1) = \underline{a}(k) + k_a \frac{\partial}{\partial \underline{a}} [E\{e^2(k)\}] \quad (4.9a)$$

$$\underline{b}(k+1) = \underline{b}(k) + k_b \frac{\partial}{\partial \underline{b}} [E\{e^2(k)\}]. \quad (4.9b)$$

The gradient involved in (4.9) can be approximated using the techniques outlined in [Ref. 11]. The iterative LMS algorithm

for estimating the coefficients  $\underline{a}$  and  $\underline{b}$  is,

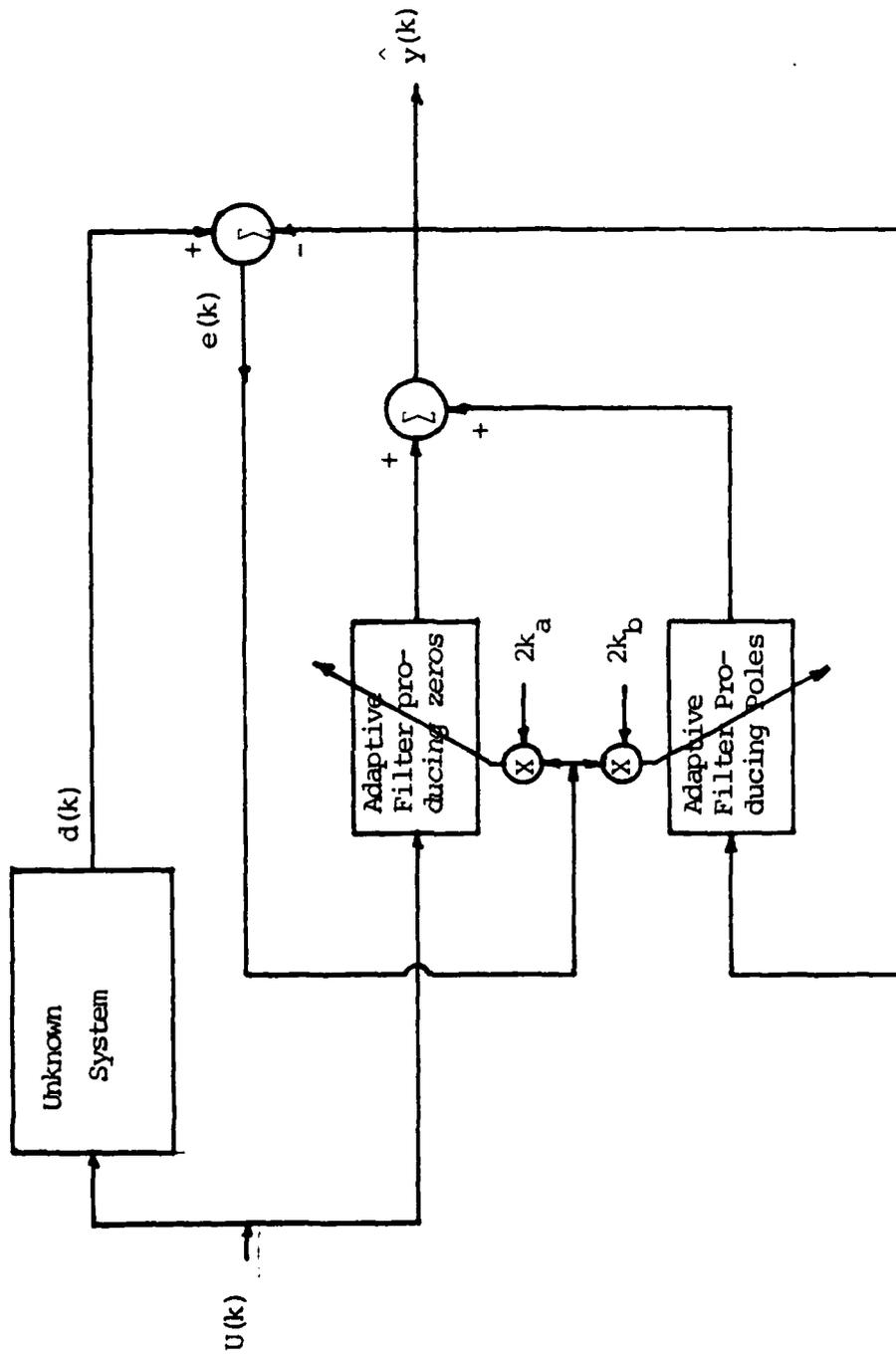
$$\underline{a}(k+1) = \underline{a}(k) - 2k_a e(k) \underline{u}(k) \quad (4.10a)$$

$$\underline{b}(k+1) = \underline{b}(k) - 2k_b e(k) \underline{y}(k). \quad (4.10b)$$

Figure 4.1 is a block diagram implementation of the Adaptive Recursive LMS Filter algorithm as applied to system identification. The input signal,  $u(k)$ , to both the unknown system and the adaptive filter is a zero mean, white gaussian noise sequence of samples from a random process. The output of the unknown system is  $d(k)$  which is used as the training signal for the filter. The output of the adaptive filter,  $y(k)$ , is compared with the desired signal,  $d(k)$ , from which an error signal is derived. The error signal,  $e(k)$ , is then used in the LMS algorithm, equation (4.10), to iteratively adjust the  $\underline{a}$  and  $\underline{b}$  coefficients. Theoretically, when the coefficients of the zeros-producing adaptive filter and the poles-producing adaptive filter correspond exactly with those of the unknown system, the error sequence should be zero and the unknown system is modeled by the Adaptive Recursive LMS Filter. The convergence constants,  $k_a$  and  $k_b$ , are chosen a-priori as are the initial values,  $\underline{a}(0)$  and  $\underline{b}(0)$ , for the coefficients.

## B. THE COMPARISON

One can begin the comparison between the Adaptive Kalman Identifier and the Adaptive Recursive LMS filter by noting



Filter 4.1. Adaptive Recursive LMS Filter Applied to System Identification

the similarities between the equations (2.43) and (4.8), repeated here for convenience,

$$\begin{bmatrix} \hat{\underline{a}}(k+1|k) \\ \hat{\underline{b}}(k+1|k) \end{bmatrix} = \begin{bmatrix} R_{uu}(k) & -R_{uy}(k-1) \\ -R_{yu}^T(k-1) & R_{yy}(k) \end{bmatrix}^{-1} \begin{bmatrix} R_{uy}(k) \\ R_{yy}(k) \end{bmatrix} \quad (2.43a)$$

$$A_1 = B_1 C_1 \quad (2.43b)$$

$$\begin{bmatrix} R_{uu}(k) & R_{uy}(k-1) \\ R_{uy}^T(k-1) & R_{yy}(k) \end{bmatrix} \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} = \begin{bmatrix} R_{du}(k) \\ R_{dy}(k) \end{bmatrix} \quad (4.8a)$$

$$\begin{bmatrix} \hat{\underline{a}} \\ \hat{\underline{b}} \end{bmatrix} = \begin{bmatrix} R_{uu}(k) & R_{uy}(k-1) \\ R_{uy}^T(k-1) & R_{yy}(k) \end{bmatrix}^{-1} \begin{bmatrix} R_{du}(k) \\ R_{dy}(k) \end{bmatrix} \quad (4.8b)$$

$$A_2 = B_2 C_2 \quad (4.8c)$$

The upper left partitions of  $B_1$  and  $B_2$  are obviously identical, in that the AKI and the Adaptive Recursive LMS both implement instantaneous estimates of the input covariance functions,  $R_{uu}(k)$ . The lower right partitions of  $B_1$  and  $B_2$  are similar with one subtle difference. Both are instantaneous covariance functions; however,  $B_1$  employs the covariance,  $R_{yy}(k)$ , of the actual data whereas  $B_2$  uses, in fact, previous filter estimates to compute the output covariance,  $R_{yy}(k)$ , more properly denoted by  $R_{yy}^{\hat{\hat{}}}(k)$ . The negative sign of the upper right and lower left partitions in  $B_1$ , in comparison with  $B_2$ , is a result of the initial definition for the recursive

weights,  $\underline{b}_i$ . That is, the transfer function for the general ARMA equation can be written as,

$$H_{\text{ARMA}}(z) = \frac{a_0 + \sum_{i=1}^m a_i z^{-i}}{1 - \sum_{i=1}^m b_i z^{-i}} \quad (4.9a)$$

or,

$$H'_{\text{ARMA}}(z) = \frac{a_0 + \sum_{i=1}^m a_i z^{-i}}{1 + \sum_{i=1}^n b_i z^{-i}} \quad (4.9b)$$

The second and more important observation is that cross-correlations employed by equation (4.8b) use the instantaneous values for the past estimates of the outputs while (2.43a) uses the actual past values for  $y(k)$ . Similarly, comparing  $C_1$  with  $C_2$ , one finds that the lower partition of  $C_2$  uses the cross-correlation between the desired signal and the past estimates of the adaptive recursive LMS filter.

Equation (4.8a) can convey the above information more clearly if it is instead written as,

$$\begin{bmatrix} \hat{\underline{a}} \\ \hat{\underline{b}} \end{bmatrix} = \begin{bmatrix} R_{uu}(k) & R_{uy}(k-1) \\ R_{uy}^T(k-1) & R_{yy}(k) \end{bmatrix}^{-1} \begin{bmatrix} R_{du}(k) \\ R_{dy}(k) \end{bmatrix} \quad (4.10)$$

The convergence factors,  $k_a$  and  $k_b$ , can be compared to the steady state Adaptive Kalman Identifier gains by making

the identical analogies (3.8) as was done in the previous chapter. One also observes that since the output data that is processed by the Adaptive Recursive LMS filter are in actuality filter estimates, a more correct version of Feintuch's original algorithm (4.10) is written as,

$$\underline{a}(k+1) = \underline{a}(k) - 2k_a e(k) \underline{u}(k) \quad (4.11a)$$

$$\underline{b}(k+1) = \underline{b}(k) - 2k_b e(k) \hat{\underline{y}}(k) . \quad (4.11b)$$

In augmented form, equation (4.11) is,

$$\begin{bmatrix} \underline{a}(k+1) \\ \underline{b}(k+1) \end{bmatrix} = \begin{bmatrix} \underline{a}(k) \\ \underline{b}(k) \end{bmatrix} - \begin{bmatrix} 2k_a \underline{u}(k) \\ 2k_b \underline{y}(k) \end{bmatrix} [d(k) - \underline{a}(k) \underline{u}(k) - \underline{b}(k) \underline{y}(k)]$$

$$\begin{bmatrix} \underline{a}(k+1) \\ \underline{b}(k+1) \end{bmatrix} = \begin{bmatrix} \underline{a}(k) \\ \underline{b}(k) \end{bmatrix} - \begin{bmatrix} 2k_a \underline{u}(k) \\ 2k_b \underline{y}(k) \end{bmatrix} \left[ d(k) - [\underline{u}(k) \quad \underline{y}(k)] \begin{bmatrix} \underline{a}(k) \\ \underline{b}(k) \end{bmatrix} \right] . \quad (4.12)$$

Recalling equation (2.15a) in a slightly different form we have,

$$\begin{bmatrix} \hat{\underline{a}}(k+1|k) \\ \hat{\underline{b}}(k+1|k) \end{bmatrix} = \begin{bmatrix} \hat{\underline{a}}(k|k-1) \\ \hat{\underline{b}}(k|k-1) \end{bmatrix} + K(k) \left[ z(k) - H(k) \begin{bmatrix} \hat{\underline{a}}(k|k-1) \\ \hat{\underline{b}}(k|k-1) \end{bmatrix} \right] . \quad (4.13)$$

It is a simple matter to explore the similarities between (4.12) and (4.13) and make the following associations,

AKI

Adaptive Recursive LMS

$$\begin{bmatrix} \hat{\underline{a}}(k+1|k) \\ \text{-----} \\ \hat{\underline{b}}(k+1|k) \end{bmatrix} \Leftrightarrow \begin{bmatrix} \underline{a}(k+1) \\ \text{-----} \\ \underline{b}(k+1) \end{bmatrix} \quad (4.14a)$$

$$\begin{bmatrix} \hat{\underline{a}}(k|k-1) \\ \text{-----} \\ \hat{\underline{b}}(k|k-1) \end{bmatrix} \Leftrightarrow \begin{bmatrix} \underline{a}(k) \\ \text{-----} \\ \underline{b}(k) \end{bmatrix} \quad (4.14b)$$

$$z(k) \Leftrightarrow d(k) \quad (4.14c)$$

$$H(k) = \begin{bmatrix} u(k) & \dots & & \\ u(k-m) & & y(k-1) & \\ & \dots & & y(k-n) \end{bmatrix} \Leftrightarrow \begin{bmatrix} \underline{u}(k) & \underline{y}(k) \end{bmatrix} \quad (4.14d)$$

$$K(k) \Leftrightarrow \begin{bmatrix} 2k_a \underline{u}(k) \\ \text{-----} \\ 2k_b \underline{\quad}(k) \end{bmatrix} \quad (4.14e)$$

The associations (4.14d) and (4.14e) are the major differences between the AKI and the Adaptive Recursive LMS. It has been shown that the mean square error surface employing not only estimated feed forward coefficients but also estimated feedback (recursive) coefficients is in general multimodal [Ref. 36]. Hence, the Adaptive Recursive LMS algorithm does not minimize the mean square error [Ref. 19,20], and in general the gradient algorithm in this case does not seek the global minimum.

### C. OBSERVATIONS

The complex nature of estimating recursive coefficients is yet a formidable problem [Ref. 19,20]. It seems that Feintuch's algorithm is successful due to the a-priori knowledge of the minimal order generating the desired signal. This knowledge is used in setting the order of the adaptive recursive filter.

## V. SYNTHETIC DATA GENERATING PLANTS

The Adaptive ARMA Kalman Identifier was tested using computer derived data from several models. The input/output data was collected by driving a known plant with a zero mean, unit variance, white gaussian noise sequence of samples. The AKI algorithm was repeatedly tested using data from progressively more complex models.

### A. AUTOREGRESSIVE-MOVING AVERAGE DATA

Autoregressive-Moving Average (ARMA) data was generated using an equation of the form,

$$\begin{aligned} y(k) = & b_1 y(k-1) + b_2 y(k-2) + \dots + b_n y(k-n) \\ & + a_0 u(k) + \dots + a_m u(k-m) \end{aligned} \quad (5.1)$$

Taking the Z-transform of (5.1) we have,

$$\begin{aligned} Y(z) = & b_1 z^{-1} Y(z) + b_2 z^{-2} Y(z) + \dots + b_n z^{-n} Y(z) \\ & + a_0 U(z) + \dots + a_m z^{-m} U(z) \end{aligned} \quad (5.2)$$

where  $Y(z)$  ( $U(z)$ ) represents the Z-transform of the output (input) and  $z^{-n_0} Y(z)$  ( $z^{-n_0} U(z)$ ) represents the Z-transform of the output (input) delayed  $n_0$  time steps. Equation (5.1) has already been defined as the general form of the ARMA process provided that the sequence  $u(k-i)$   $i = 0, 1, \dots$  comes from a gaussian random process [Ref. 25]. The ratio,

$$H(z) = \frac{Y(z)}{U(z)} = \frac{a_0 + a_1 z^{-1} + \dots + a_m z^{-m}}{1 - b_1 z^{-1} - \dots - b_n z^{-n}} \quad (5.3)$$

can then be readily recognized as the general transfer function [Ref. 37] of a digital plant. That is, the  $u(k)$  is the observed input and the  $y(k)$  is the perfectly measured output. Consider that the measurements of the output include some random error,  $v(k)$ , due to the inaccuracy of the measuring instruments, then the resulting situation is as depicted in Figure 5.1. A practical, judicious and reasonable description of the measurement error is that this be a stochastic process whose distribution is zero mean gaussian. The noisy measurement,  $z(k)$ , is then the output,  $y(k)$ , plus the measurement error,  $v(k)$ .

There are three cases of interest: Case 1:  $b_i = 0$  for all  $i$ ; Case 2:  $a_i = 0, i = 1, 2, \dots, m$ ; Case 3:  $a_i \neq 0, b_i \neq 0$  for all  $i$ .

#### 1. Moving Average Data (Case 1)

Case one is readily recognized as the all zero plant which produces moving average data. A simple second order plant was defined where  $a_0 = 1.0, a_1 = 2.0$  and  $a_2 = 3.0$ . The gaussian noise sequences,  $u(k)$  and  $v(k)$ , were obtained using the general purpose IMSL subroutine, GGNML, provided by the computer center at the Naval Postgraduate School.

#### 2. Autoregressive Data (Case 2)

Case two results in the all pole plant producing autoregressive data. The coefficients were selected such that the

plant of Figure 5.1 produced stable data. In precise terms, the poles of the function,

$$H(z) = \frac{1}{1 + \frac{26}{24} z^{-1} + \frac{9}{24} z^{-2} + \frac{1}{24} z^{-3}} \quad (5.4)$$

were located completely within the unit circle in the z-plane ensuring a stable plant [Ref. 37,38]. The noise sequences were obtained as before.

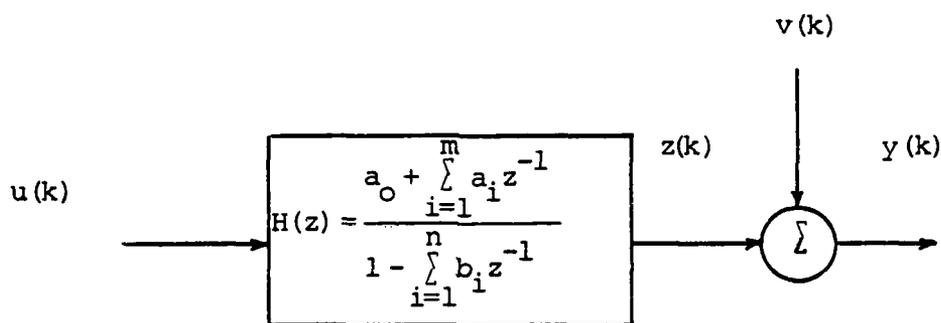


Figure 5.1. ARMA Digital Plant

### 3. Autoregressive-Moving Average Data (Case 3)

Case three follows from a logical combination of the two previous cases. That is, the transfer function,  $H(z)$ , now has both zeros and poles. In order to compare with previous work done in the area of system identification [Ref. 6], one of Perry's models [Ref. 6] was used. Namely, the transfer function  $H(z)$  used for this case was,

$$H(z) = \frac{1 + 1.4z^{-1} + .98z^{-2}}{1 - 1.14z^{-1} + 1.4549z^{-2} - .88490z^{-3} + .40745z^{-4}} \quad (5.5)$$

The input,  $u(k)$ , was obtained as in the previous two cases using the IMSL subroutine GGNML set to unit variance and zero mean. The output measurement noise sequence, however, was set to a variance of .0001. The basic software, program ARMA, which generated the different data is included as Appendix B. Basically, the coefficients A1-A10 change the character of the general ARMA equation (5.1) which can be selected to produce the desired plant data. The input/output data is then written onto a disk file for subsequent analysis.

In order to compare some of the findings in this thesis with the results previously obtained by Feintuch [Ref. 18], the transfer function,

$$H(z) = \frac{0.05 - 0.40z^{-1}}{1.0 - 1.1314z^{-1} + 0.25z^{-2}} \quad (5.6)$$

was also used as a source of synthetic data. Feintuch used (5.6) as a source of data with which to analyze the operation of his Adaptive Recursive LMS Filter.

#### B. PHASE LOCKED LOOP DATA

The second class of data used to exercise the Adaptive ARMA Kalman Identifier was derived from a computer simulated phase locked loop (PLL) developed by Romeo [Ref. 7]. The basic PLL algorithm, however, implemented a forward Euler integration scheme.

Briefly, in block diagram form, the phase tracking characteristics of the PLL in the frequency domain can be depicted as in Figure 5.2.

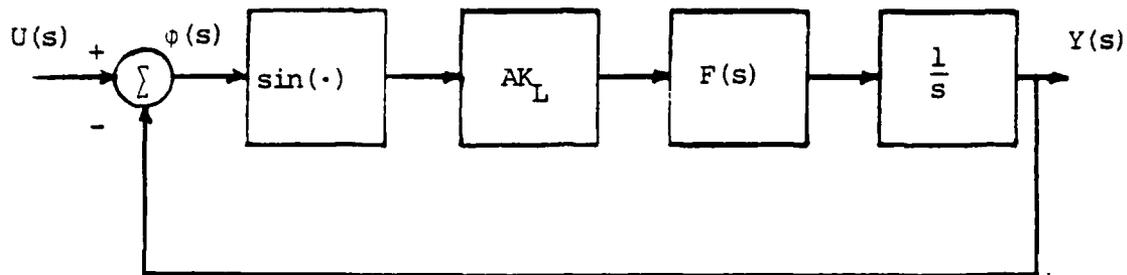


Figure 5.2. PLL Block Diagram

Where:

A is the input voltage

K is the loop gain

F(s) is the filter transfer function

For most applications, it is assumed that  $\phi(t) \ll \pi/2$  allowing one to make the approximation,  $\sin \alpha \approx \alpha$ . This is the linear mode of operation. Romeo [Ref. 7] chose the filter characteristics  $F(s) = 1 + K/S$  and the same characteristics were used in this thesis in order to compare results. The parameters of the overall system were adjusted to obtain a damping coefficient,  $\zeta$ , of 0.3 and a natural frequency,  $\omega_n$  of 3.33 rad/sec. This resulted in a step response overshoot of about 46.7% at  $t \approx .75$  sec. Solving for the time domain step response analytically in the linear region one obtains,

$$y(t) = 1 + 1.048e^{-t} \sin(3.178t - 1.266) \quad (5.7)$$

The final block diagram of the PLL system is shown in Figure 5.3.

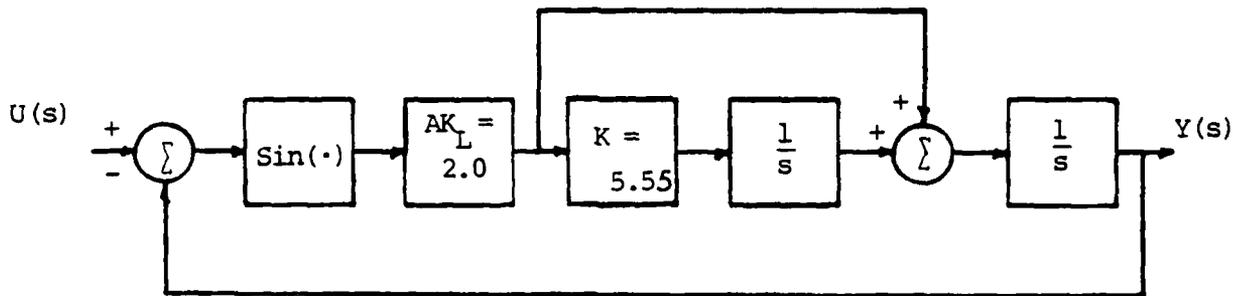
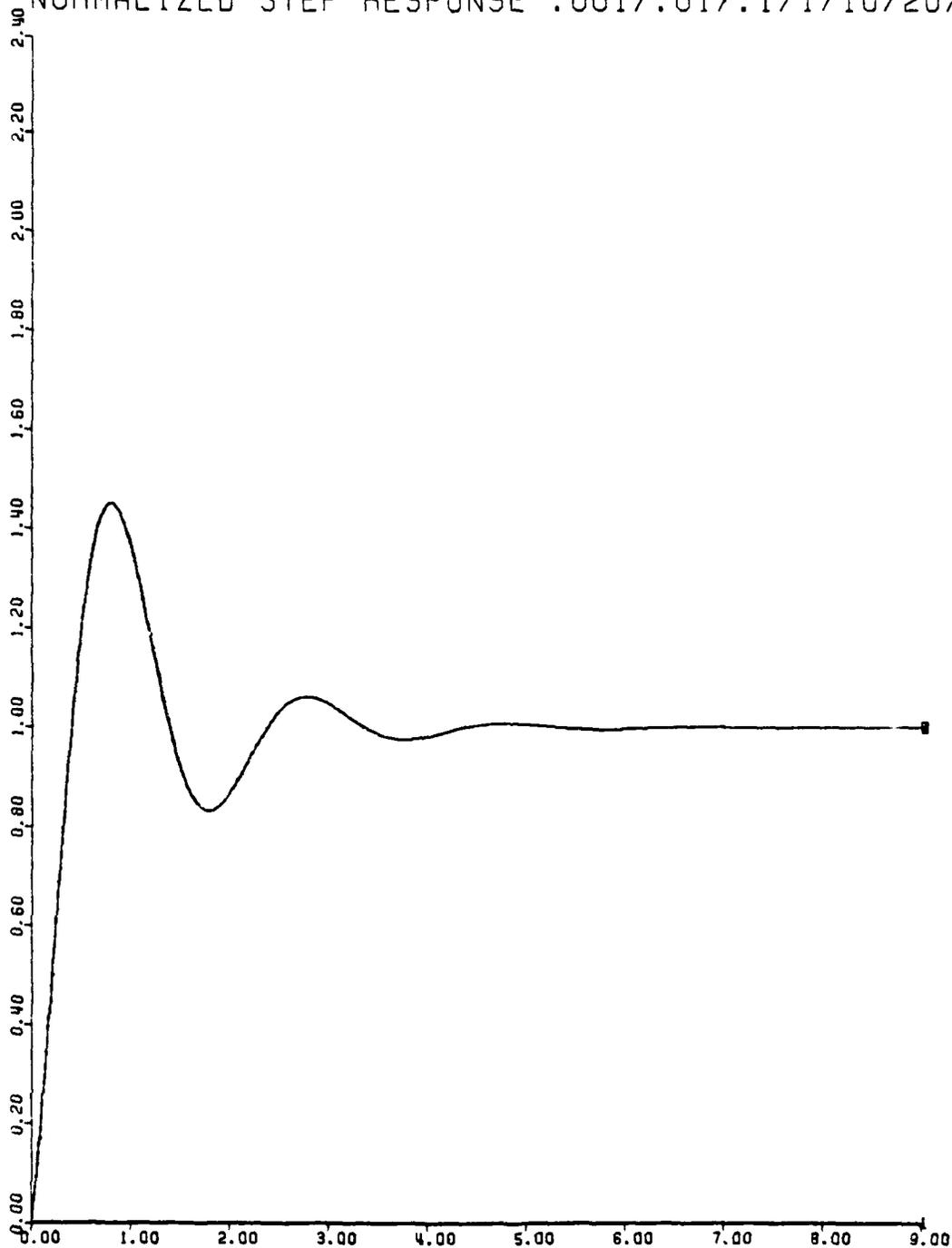


Figure 5.3. Final PLL Block Diagram

Normalized step responses for several step magnitudes using Digital Simulation Language (DSL) were obtained. Figure 5.4 shows the step responses of several step inputs in the range  $0 < u(t) < 30$ . It is obvious that the PLL is in its approximate linear region and does not exhibit any discernible distortion. Figure 5.5 is a repeat of the above test; however, the range of step inputs are  $30 \leq u(t) \leq 170$ . It is apparent from Figure 5.5 that the  $\sin(\cdot)$  nonlinearity begins to have some effect on the operation of the PLL in that the amplitude of the responses are reduced and delayed.

The DSL PLL system, however, was not used as a source of synthetic data because of the problem of nonstationary statistics of the input signal using the random sequence generators available under DSL already discussed in Ref. 7. These simulations were nevertheless used as a basis for comparison of the operation of the discrete PLL developed in Reference 7 but modified to implement a forward Euler integration scheme which is used in this thesis.

2ND ORDER PLL -- TRAPEZOIDAL INTEGRATION  
NORMALIZED STEP RESPONSE .001/.01/.1/1/10/20/30

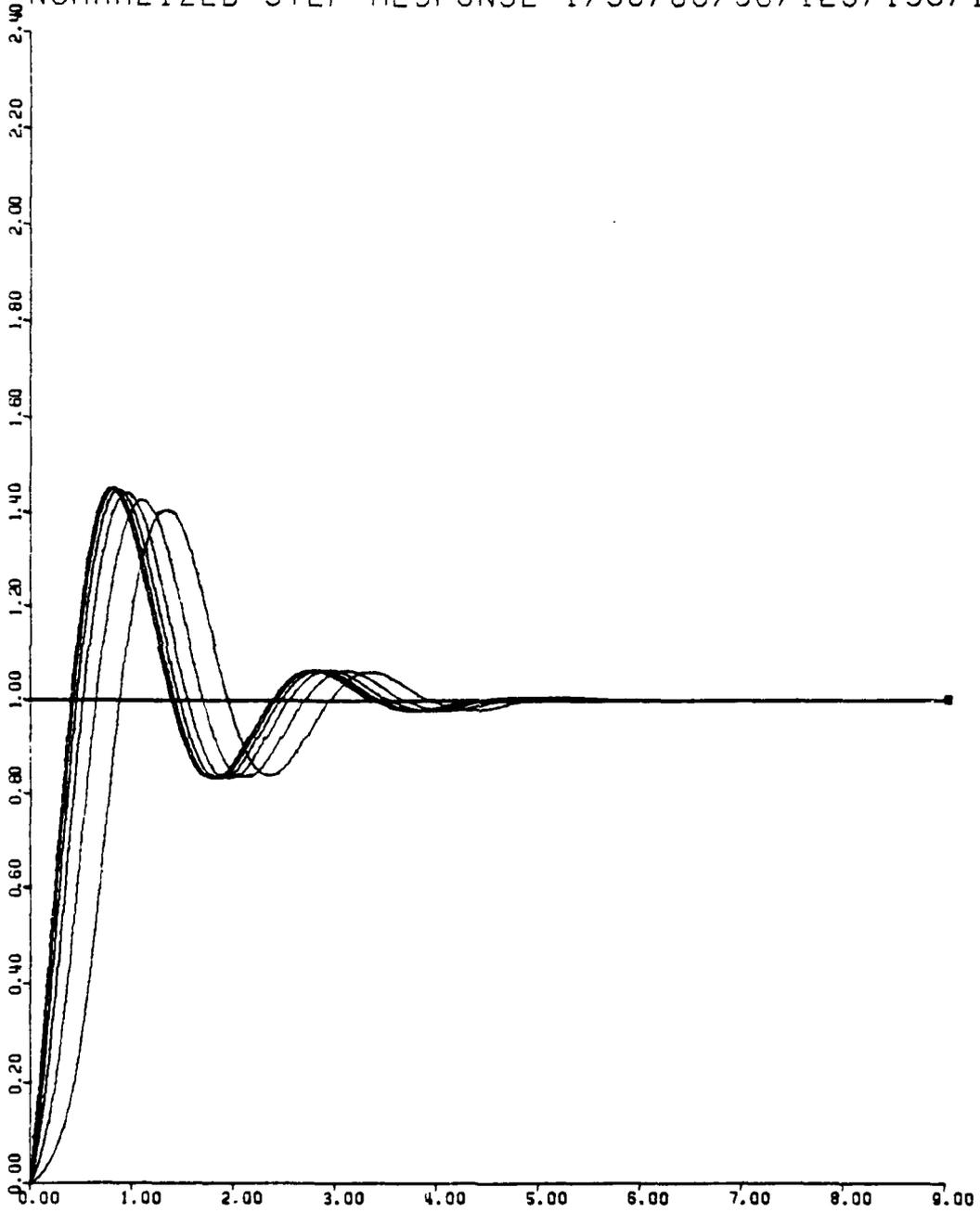


XSCALE= 1.00      UNITS/INCH  
YSCALE= 0.20      UNITS/INCH

RUN NO. 1  
PLOT NO. 1

Figure 5.4. PLL Step Response, Linear Region

2ND ORDER PLL -- TRAPEZOIDAL INTEGRATION  
NORMALIZED STEP RESPONSE 1/30/60/90/120/150/170



XSCALE= 1.00  
YSCALE= 0.20

UNITS/INCH  
UNITS/INCH

RUN NO. 1  
PLOT NO. 1

Figure 5.5. PLL Step Response, Non-Linear Region

The forward Euler integrator takes the form,

$$\frac{1}{s} \iff \frac{Tz^{-1}}{1-z^{-1}}, \quad T = .01 \quad (5.8)$$

The block diagram of Figure 5.3 changes character to appear as in Figure 5.6.

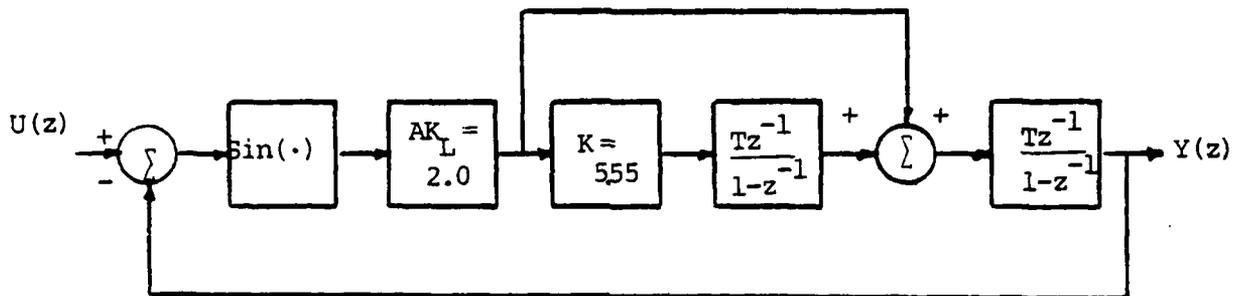


Figure 5.6. Discrete PLL Block Diagram

Applying Mason's gain rule to the block diagram of Figure 5.6 in the linear region we arrive at,

$$\frac{Y(z)}{U(z)} = H(z) = \frac{0.02z^{-1} - 0.018890z^{-2}}{1 - 1.980z^{-1} + .981110z^{-2}} \quad (5.9)$$

the linearized PLL transfer function. The discrete PLL system has zeros at (0.0, .9445) and a complex conjugate pole pair at (0.990 ± j.03178).

The discrete PLL system was implemented using a Fortran program on the IBM 370 in double precision. The source code for the discrete PLL is included as Appendix C. The discrete

PLL was tested using various magnitude step inputs. Results were similar to those from the DSL simulation. That is, the discrete PLL exhibited the same approximate overshoot, rise time, natural frequency and nearly identical time of max overshoot in the linear region. To implement the PLL using forward Euler integration, the block diagram of Figure 5.6 was reconfigured to appear as Figure 5.7. Note that Figure 5.7 does not employ any delay free loops, hence it is easily programmable.

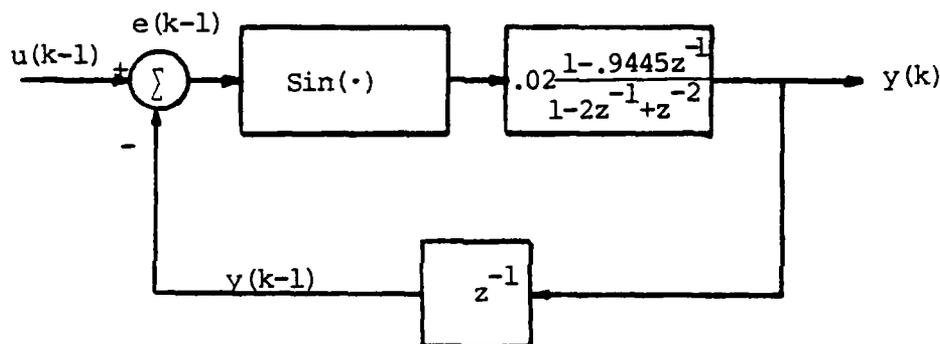


Figure 5.7. Programmable PLL Implementing Forward Euler Integration

Figure 5.8 summarizes the results of the discrete PLL simulation for normalized step inputs in the range  $0 < u(t) \leq 30$ . Similarly, the discrete PLL was tested in the nonlinear region using normalized step inputs in the range  $30 \leq u(t) \leq 170$ . The discrete PLL displayed the same characteristics of decreased output amplitude and delay which was first observed in the DSL simulation. Figure 5.9 summarizes the test results of the discrete PLL in the nonlinear region.

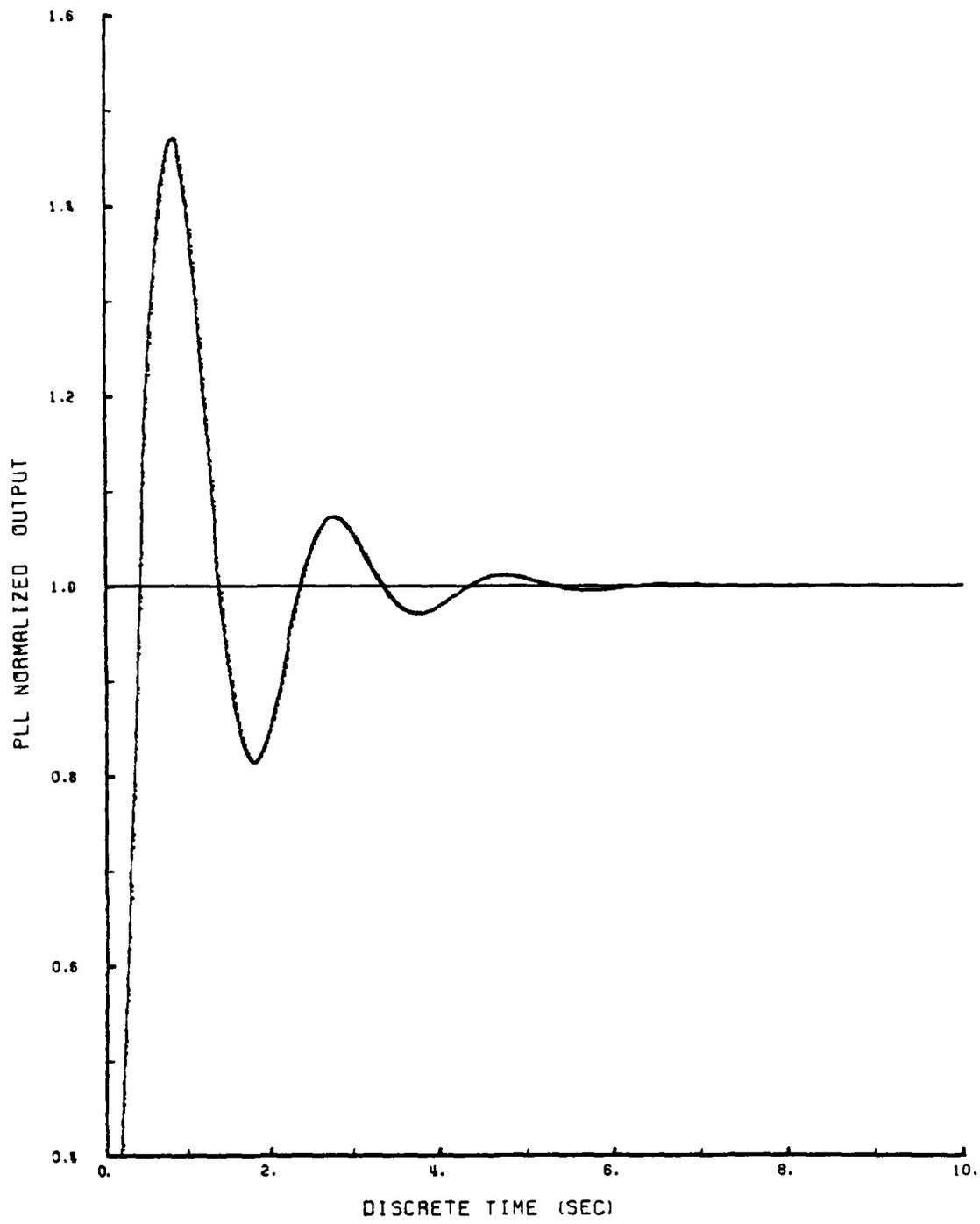


Figure 5.8. Discrete PLL Step Response, Linear Region

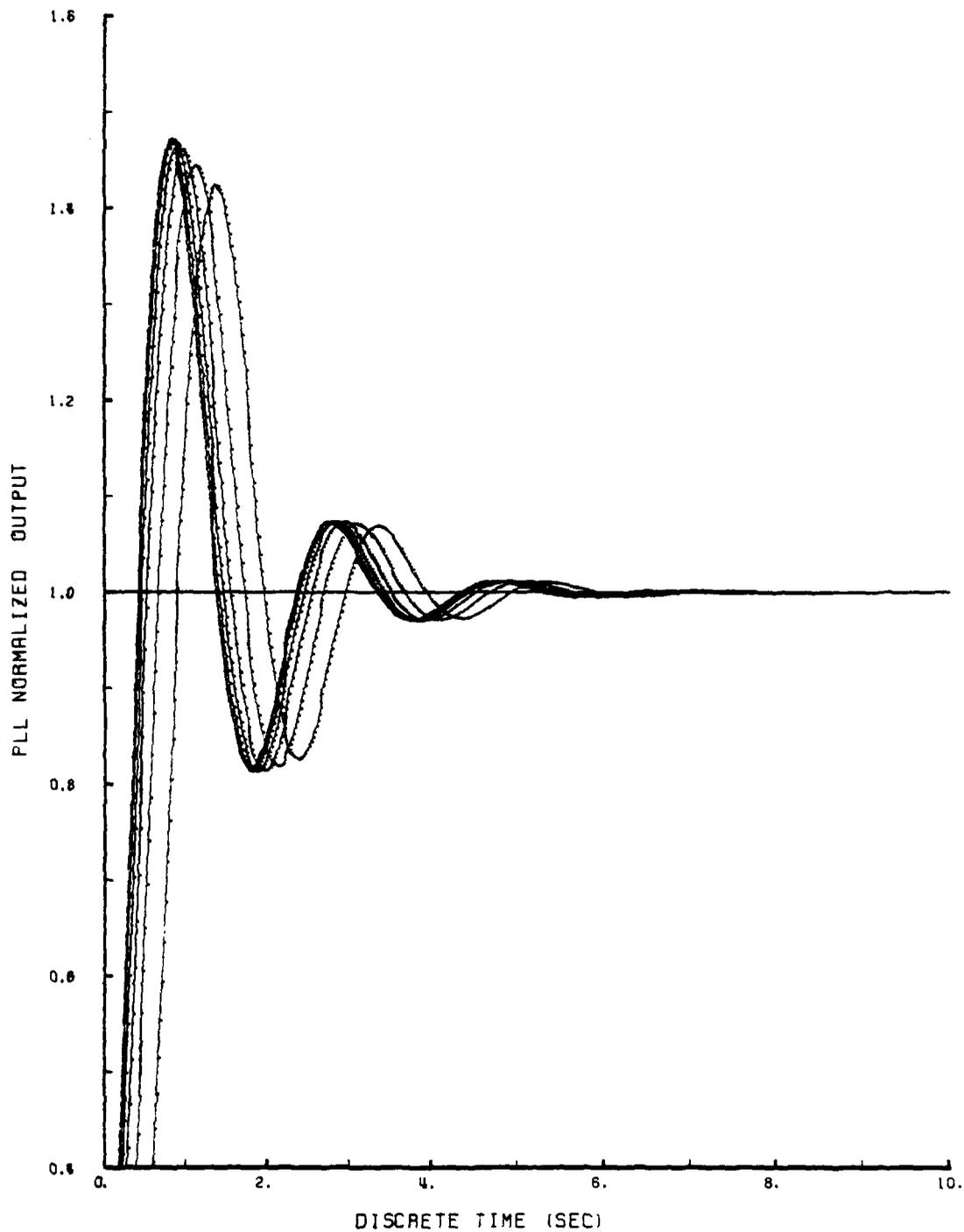


Figure 5.9. Discrete PLL Step Response, Nonlinear Region

It was felt that the PLL was adequately modeled for use as a source of data to be subsequently analyzed by the Adaptive Kalman Identifier.

## VI. MODEL IDENTIFICATION SOFTWARE

Three basic Fortran IV programs for use on the IBM 370 were developed:

- (1) ADAPTSN: Adaptive Kalman Identifier
- (2) LMS: Adaptive Least Mean Square Filter
- (3) LMSR: Adaptive Recursive Least Mean Square Filter

All three programs used double precision arithmetic to minimize the effects of truncation error, limit cycles and roundoff error. In all cases the orders of the autoregressive and moving average processes were read in unformatted form from a disk file along with the input/output data of the unknown system to be analyzed. An attempt was made to minimize memory storage, but not at the expense of program flow and clarity.

### A. ADAPTIVE KALMAN IDENTIFIER

Program ADAPTSN is a versatile Fortran IV software program which implements equations (2.15) to identify the coefficients of an MA, AR, or ARMA process. Additionally, at each iteration the poles and zeros of the evolving transfer function are computed. ADAPTSN can also be used to perform regression analysis on non-linear terms as discussed in Chapter VII. The versatility of the AKI lies in the various options which are available to the user. By properly selecting the flags N, M, and NL, the user can perform regression analysis on data whose combined order  $(N+M+1)$  is less than or equal to 20. Table 6.1 lists the ascribed meanings of the various flags.

Table 6.1

AKI FLAG MEANINGS

- N: the order of the autoregressive process
- M: one greater than the order of the moving average process
- NL = 0: AKI is used to identify the coefficients associated with the general ARMA equation (2.1b) (linear regression)
- NL = 1: AKI is used to identify the coefficients of the linear general ARMA equation and the weighting coefficient associated with  $u^3(k-i)$
- NL = 2: AKI is used as in NL = 0,1 and additionally identifies the weighting coefficient associated with  $y^3(k-i)$
- NL = 3: AKI is used as in NL = 0,1,2 and additionally identifies the weighting coefficient associated with  $u^2(k-i)y(k-i)$
- NL = 4: AKI is used as in NL = 0,1,2,3, and additionally identifies the weighting coefficient associated with  $y^2(i-i)u(k-i)$
- NL = 5: AKI is implemented to analyze time series and identify the ARMA coefficients associated with the Box-Jenkins model [Ref. 26].

The option given by NL = 5 was not extensively tested and as such only limited results are available. Table 6.2 outlines the allowable flag combinations which will produce a valid analysis of the data.

For purposes of this thesis, the non-linear (NL) options were configured to implement the expansion terms associated with the Taylor series expansion of the sine function. This, however, can be changed at the user's discretion to implement any other series expansion by inserting the appropriate Fortran statements at the proper location in the AKI program.

Table 6.2

## VALID FLAG COMBINATIONS

OPTION (NL)	MAX COMBINED ORDER
0	$N + M \leq 20$
1	$N + 2M \leq 20; N, M \neq 0$
2	$2N + 2M \leq 20; N, M \neq 0$
3	$3N + 2M \leq 20; N, M \neq 0$
4	$4N + 2M \leq 20; N, M \neq 0$
5	$N + M \leq 20$

The overall structure of the AKI program uses subroutine calls to compute the various quantities necessary for the eventual computation of the system ARMA coefficients. These subroutines are: GAIN, RECKON, PRINT, NEXT. A brief description of the function each performs is given at the beginning of each subroutine. Figure 6.1 describes the general program flow of the AKI. The source code for program ADAPTSN is included as Appendix D.

#### B. ADAPTIVE LEAST MEAN SQUARE FILTER

Program LMS uses the equations developed in Chapter III to compute the coefficients (or weights) associated with a moving average process. The LMS filter program is capable of computing up to 12 MA coefficients; or equivalently, is capable of identifying an eleventh order moving average process. The general program flow of the LMS filter is presented in Figure 6.2 and the annotated source code is included as Appendix E.

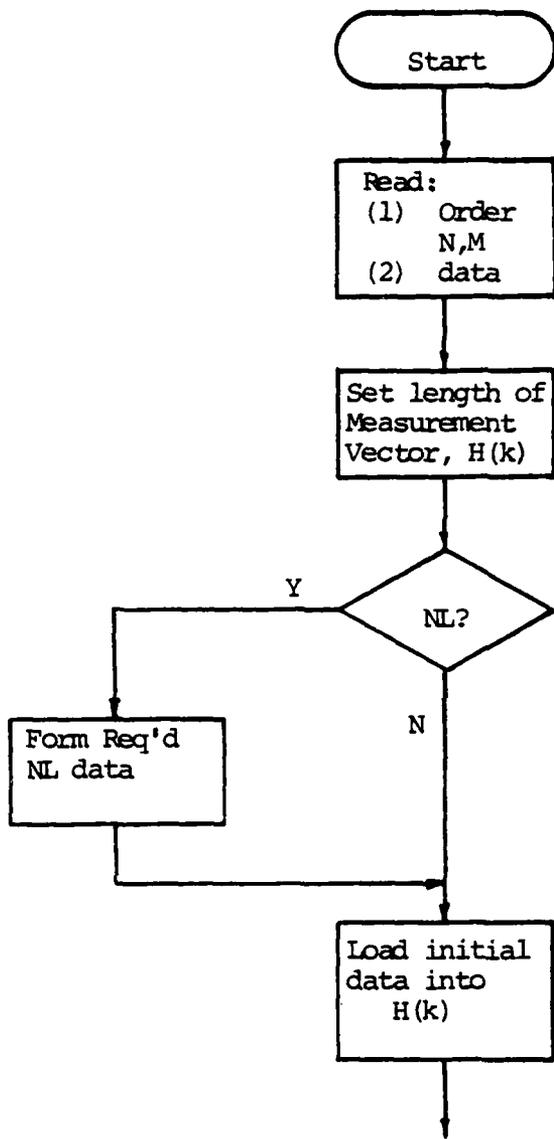


Figure 6.1. AKI General Program Flow

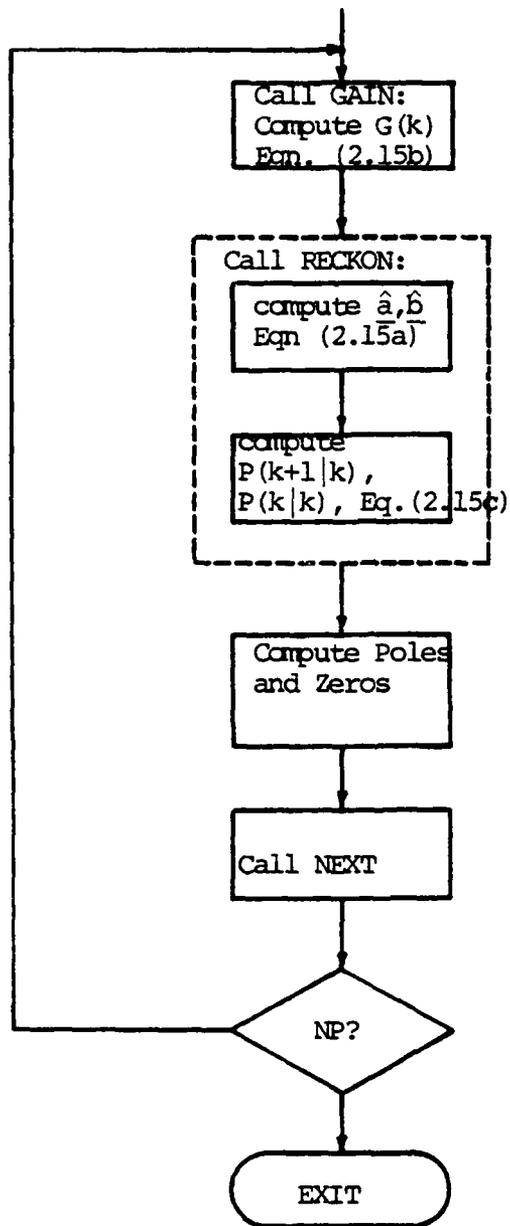


Figure 6.1. (CONTINUED)

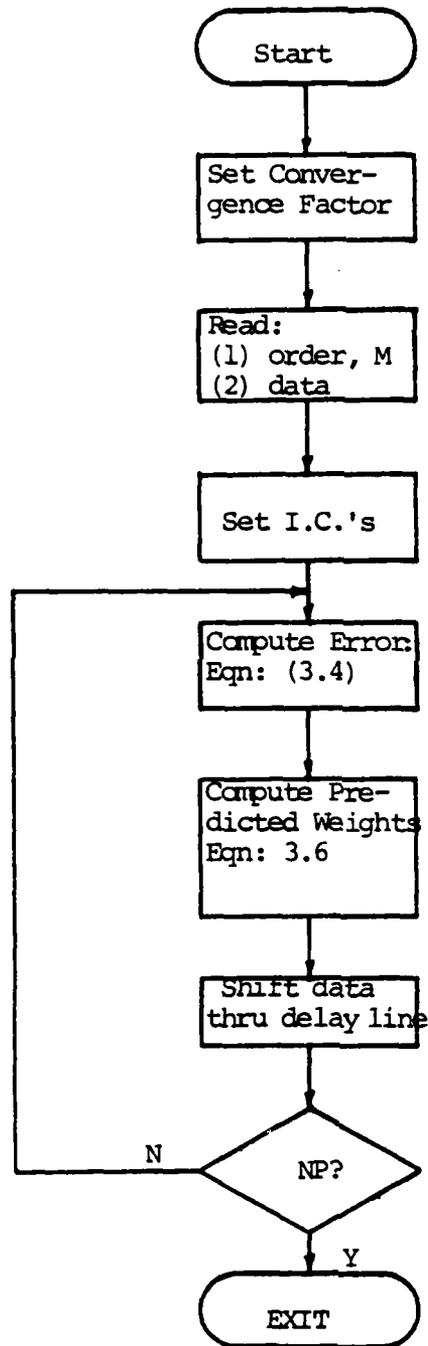


Figure 6.2. LMS Filter Program Flow

### C. ADAPTIVE RECURSIVE LEAST MEAN SQUARE FILTER

Program LMSR realizes the algorithm proposed by Feintuch and recapped in Chapter IV. It was designed to handle a combined AR and MA order of 11. That is, the program can be used to estimate  $m + 1 = M$  coefficients associated with the MA process and  $n = N$  coefficients associated with the AR process such that  $N + M \leq 12$ . The general program flow for the Adaptive Recursive LMS filter is shown in Figure 6.3 and the source code is included as Appendix F.

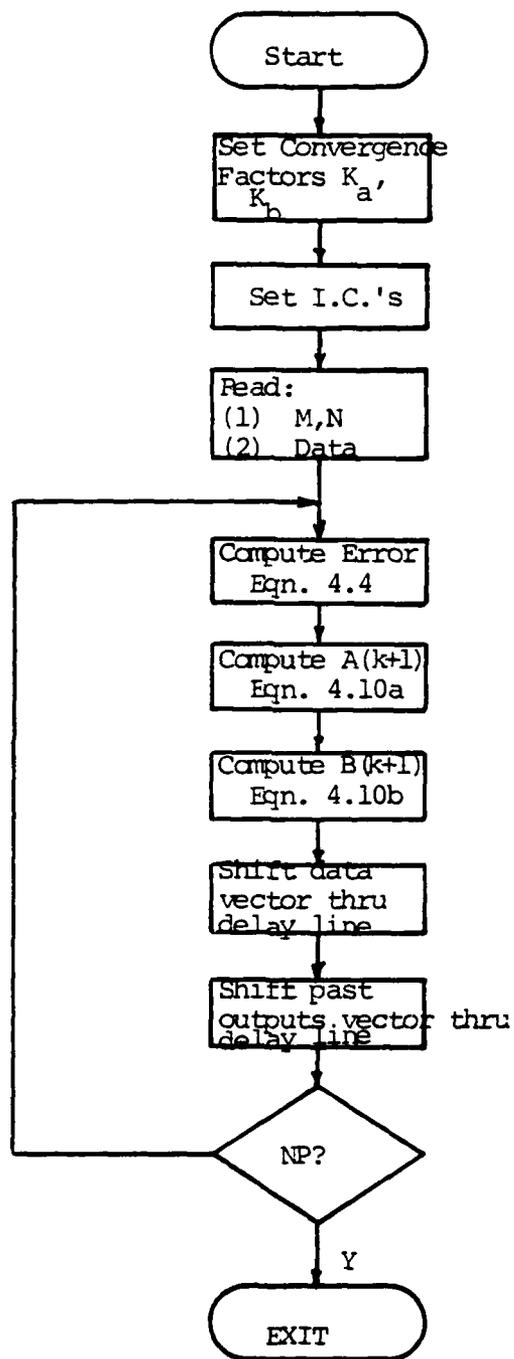


Figure 6.3. LMSR Filter Program Flow

## VII. NON-LINEAR IDENTIFICATION

The Adaptive Kalman Identifier can be modified to estimate the weighting coefficients associated with higher order terms produced by some non-linearity within the unknown system. In general, much has to be known about the non-linearity such that the functional description chosen for it is a close approximation to the effects it causes. In this thesis one possible application of the Adaptive Kalman Identifier toward identifying a system with a known non-linearity is explored. No attempt has been made to present an extensive treatment of non-linear analysis techniques.

The approach taken has been previously explored by Parker [Ref. 8]. A special case of the generalized non-linear ARMA model [Ref. 8],

$$\begin{aligned}
 y(k) = & \sum_{i_1=0}^{\infty} a(i_1)u(k-i_1) + \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} a(i_1, i_2)u(k-i_1)u(k-i_2) + \dots \\
 & + \sum_{i_1=0}^{\infty} \dots \sum_{i_m=0}^{\infty} a(i_1, \dots, i_m)u(k-i_1) \dots u(k-i_m) \\
 & + \sum_{j_1=1}^{\infty} b(j_1)y(k-j_1) + \sum_{j_1=1}^{\infty} \sum_{j_2=1}^{\infty} b(j_1, j_2)y(k-j_1)y(k-j_2) + \dots \\
 & + \sum_{j_1=1}^{\infty} \dots \sum_{j_n=1}^{\infty} b(j_1, j_2, \dots, j_n)y(k-j_1)y(k-j_2) \dots y(k-j_n) \\
 & + \sum_{i_j=0}^{\infty} \sum_{j_1=1}^{\infty} c(i_1, j_1)u(k-i_1)y(k-j_1) + \dots + \sum_{i_1=0}^{\infty} \dots \sum_{i_m=0}^{\infty} \sum_{j_1=1}^{\infty} \\
 & \dots \sum_{j_n=1}^{\infty} c(i_1, \dots, i_m, j_1 \dots j_n)u(k-i_1) \dots u(k-i_m)y(k-j_1) \dots y(k-j_n)
 \end{aligned}
 \tag{7.1}$$

is used to model the input/output relationships, non-linear transfer function, of the phase locked loop. The non-linear element, the sine function, of the PLL can be replaced by a Taylor power series expansion,

$$\sin(x) = x - \frac{1}{3!} x^3 + \frac{1}{5!} x^5 - \dots \quad (7.2)$$

Other expansions can be used, e.g., Legendre polynomials, Volterra series, ..., etc.; however, only the Taylor expansion was investigated in this thesis. Practical implementation of (7.2) suggests truncation at the third order term. The sine function is therefore approximated as

$$\sin(x) = x - \frac{1}{3!} x^3 . \quad (7.3)$$

Substituting (7.3) into the block diagram of Figure (5.6) we have Figure 7.1.

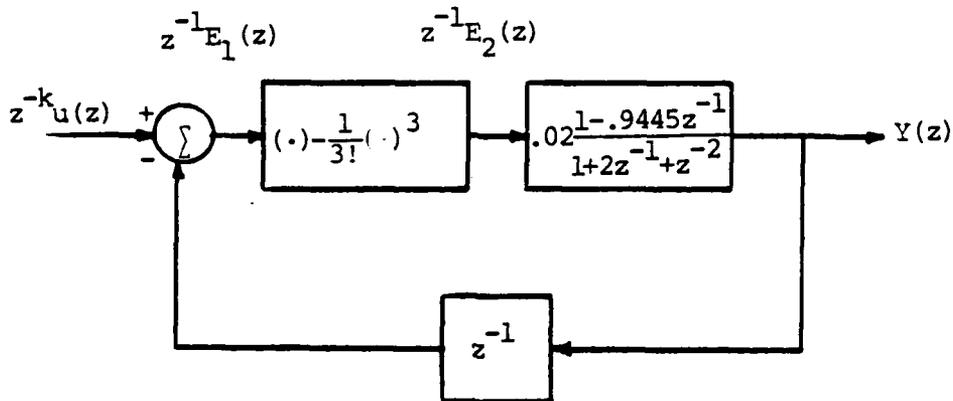


Figure 7.1. PLL, Third Order Taylor Series Approximation

The discrete time domain equations at the different nodes are:

$$e_1(k-1) = u(k-1) - y(k-1) \quad (7.4a)$$

$$e_2(k-1) = e_1(k-1) - \frac{1}{3!} e_1^3(k-1) \quad (7.4b)$$

$$y(k) = .02e_2(k-1) - .01889e_2(k-2) + 2.0y(k-1) - y(k-2) \quad (7.4c)$$

Manipulating equations (7.4a)-(7.4c) gives

$$\begin{aligned} y(k) = & .02u(k-1) - .01889u(k-2) + 1.98y(k-1) - .98111y(k-2) \\ & - .003333u^3(k-1) + .003148333u^3(k-2) \\ & + .003333y^3(k-1) - .003148333y^3(k-2) \quad (7.5) \\ & + .01u^2(k-1)y(k-1) - .09445u^2(k-2)y(k-2) \\ & - .01u(k-1)y^2(k-1) + .09445u(k-2)y^2(k-2). \end{aligned}$$

Equation (7.5) indicates which non-linear terms of equation (7.1) should be retained. Therefore, the third order non-linear approximation model should contain the following terms:

$$u(k-1), u(k-2), y(k-1), y(k-2) \quad \text{linear terms} \quad (7.6a)$$

$$u^3(k-1), u^3(k-2) \quad \text{input cubic terms} \quad (7.6b)$$

$$y^3(k-1), y^3(k-2) \quad \text{output cubic terms} \quad (7.6c)$$

$$\left. \begin{array}{l} u^2(k-1)y(k-1), u^2(k-2)y(k-2) \\ y^2(k-1)u(k-1), y^2(k-2)u(k-2) \end{array} \right\} \begin{array}{l} \text{third order cross} \\ \text{product terms} \end{array} \quad (7.6d)$$

The AKI algorithm is primarily modified to include the hybrid signals (7.6) in the structure of  $H(k)$ , the measurement vector. Since now the measurement vector,  $H(k)$  takes the form,

$$\begin{aligned} H(k) = & [u(k) \dots u(k-m), -y(k-1) \dots -y(k-n), u^3(k) \dots \\ & \dots u^3(k-m), -y^3(k-1) \dots -y^3(k-n), -(u^2(k-1)y(k-1)) \dots \\ & \dots -(u^2(k-n)y(k-n)), (u(k-1)y^2(k-1)) \dots (u(k-n)y^2(k-n))] \end{aligned} \quad (7.7)$$

the AKI algorithm calculates the coefficients associated with the special case of the generalized non-linear ARMA model, equation (7.1).

Romeo using a similar approach [Ref. 7] computes a least squares curve fit for the third order truncation of the Taylor series for the sine function over the interval  $(0, \pi/2)$ . Beginning with the approximation,

$$\sin(x) \approx \alpha x + \beta x^3 \quad (7.8)$$

Romeo finds the values for  $\alpha$  and  $\beta$  to be .865 and -.095 respectively. Performing the same operations as those used in obtaining equation (7.5) we arrive at,

$$\begin{aligned}
y(k) = & .02\alpha u(k-1) - .01889\alpha u(k-2) + (2.0 - .02\alpha)y(k-1) \\
& + (.01889\alpha - 1)y(k-2) - .02\beta u^3(k-1) + .01889\beta u^3(k-2) \\
& - .01889\beta y^3(k-2) + .06\beta u^2(k-1)y(k-1) + .02\beta y^3(k-1) \\
& - .056670\beta u^2(k-2)y(k-2) - .06\beta y^2(k-1)u(k-1) \\
& + .056670\beta y^2(k-2)u(k-2)
\end{aligned} \tag{7.9}$$

Table 7.1 tabulates the resulting weighting coefficients using the three methods just described: (1) analytically calculated values using the truncated Taylor series, (2) least squares estimate of the third order sine approximation, as computed by Romeo [Ref. 7] and (3) the coefficient estimates as computed using the AKI algorithm.

The tabulation clearly shows that the AKI outperforms the first two methods overall. That is, the linear terms were identified without question; however, the AKI failed to identify the coefficients associated with the  $y^3(i-1)$ ,  $y^3(k-2)$ ,  $u^2(k-2)y(k-2)$  and  $y^2(k-2)u(k-2)$  terms. The reason for the failure is not known and was not investigated.

Table 7.1

NON-LINEAR WEIGHTING COEFFICIENTS CALCULATED  
USING THREE METHODS

(1)	(2)	(3)
TAYLOR SERIES $\sin(x) = x - \frac{1}{3!}x^3$	LEAST SQUARES EST. $\sin(x) = \alpha x + \beta x^3$ $\alpha = .865, \beta = -.095$	AKI USING ACTUAL sin(x) data base
y(k-1)	1.98	1.982700
y(k-2)	-.98111	-.98366015
u(k)	0.0	0.0
u(k-1)	0.2	.01730000
u(k-2)	-.01889	-.01633985
u <sup>3</sup> (k)	0.0	0.0
u <sup>3</sup> (k-1)	-.003333	.00190
u <sup>3</sup> (k-2)	.003148333	-.00179455
y <sup>3</sup> (k-1)	.003333	-.00190
y <sup>3</sup> (k-2)	-.003148333	.00179455
u(k-1)y(k-1)	.01	-.005700
u <sup>2</sup> (k-2)y(k-2)	-.09445	.00538365
y <sup>2</sup> (k-1)y(k-1)	-.01	.005700
y <sup>2</sup> (k-2)u(k-2)	.09445	-.00538365
		.01913

## VIII. FINDINGS AND CONCLUSION

The performance of the Adaptive Kalman Identifier was compared to the LMS Adaptive Filter and the Adaptive Recursive LMS Filter using data derived from the models discussed in Chapter V. Results for the case where the order of the model (that is,  $m$  and  $n$  of equation (2.1b)) are assumed known are presented first followed by the analysis of the PLL data. The findings for the overmodeled case are presented next. The graphs show typical runs and do not represent ensemble averages.

### A. ORDER OF THE UNKNOWN SYSTEM IS KNOWN

#### 1. AKI vs. LMS Adaptive Filter

Using the MA form of the Adaptive Kalman Identifier, its performance can be compared against that of the LMS adaptive filter. Synthetic data derived from a plant whose transfer function is,

$$H(z) = 1.0 + 2.0z^{-1} + 3.0z^{-2} \quad (8.1)$$

was used. However, a fair comparison necessitated that the LMS Adaptive filter be "tuned" by adjustment of the convergence factor,  $k_s$ . Several values for  $k_s$  in the range  $[-.600, -.200]$  were used. The objective of tuning the LMS filter was to achieve a fast convergence time with little or no steady state error while not compromising filter stability. The three filter weights were normalized and plotted for five

convergence factor values. It can be seen from Figures 8.1-8.3 that convergence is essentially reached by step thirty-five, ( $k \approx 35$ ). As the convergence factor is further increased it can be noted in Figure 8.4 that the filter weights become more noisy, and that when  $k_s = -.600$ , filter stability is being compromised (Figure 8.5a).

Using the same data, the AKI converges to the MA coefficients in less than five iterations. Referring to Figure 8.5b, it was also noted that as the measurement noise  $v(k)$  was increased, the LMS algorithm yielded more noisy estimates whereas the AKI tended to compensate for measurement noise. This is intuitively reasonable since the AKI incorporates into its algorithm the effects of measurement error due to noisy sensors. Comparing Figure 8.6 with Figure 8.7 the latter figure clearly shows the faster convergence of the AKI to the plant coefficient.

It was shown in Chapter III that the AKI gains would approach a steady state value and indeed they did, as Figure 8.8 shows. However, the instantaneous Kalman gains display a more erratic pattern as shown in Figure 8.9. The information of Figure 8.8 was gleaned from Figure 8.9 by computing the average of the individual gains at each iteration using the algorithm,

$$E\{\underline{G}(k+1)\} = E\{\underline{G}(k)\} + \frac{1}{k+1}[\underline{G}(k) - E\{\underline{G}(k)\}] \quad (8.2)$$

where,

$E\{\underline{G}(k+1)\}$  is the one step prediction of the average value for the gains,

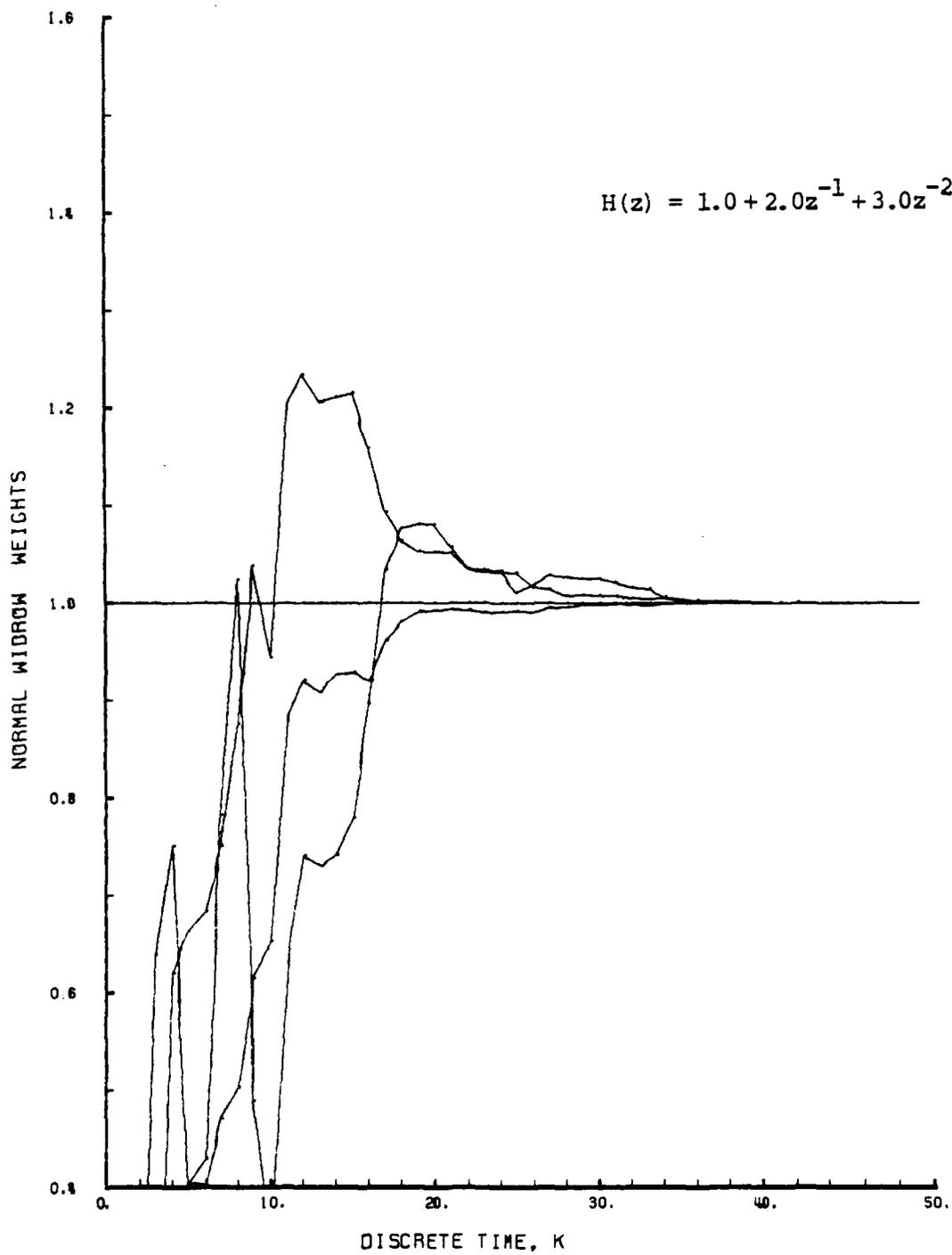


Figure 8.1. LMS Adaptive Filter Weights,  $k_s = -.200$

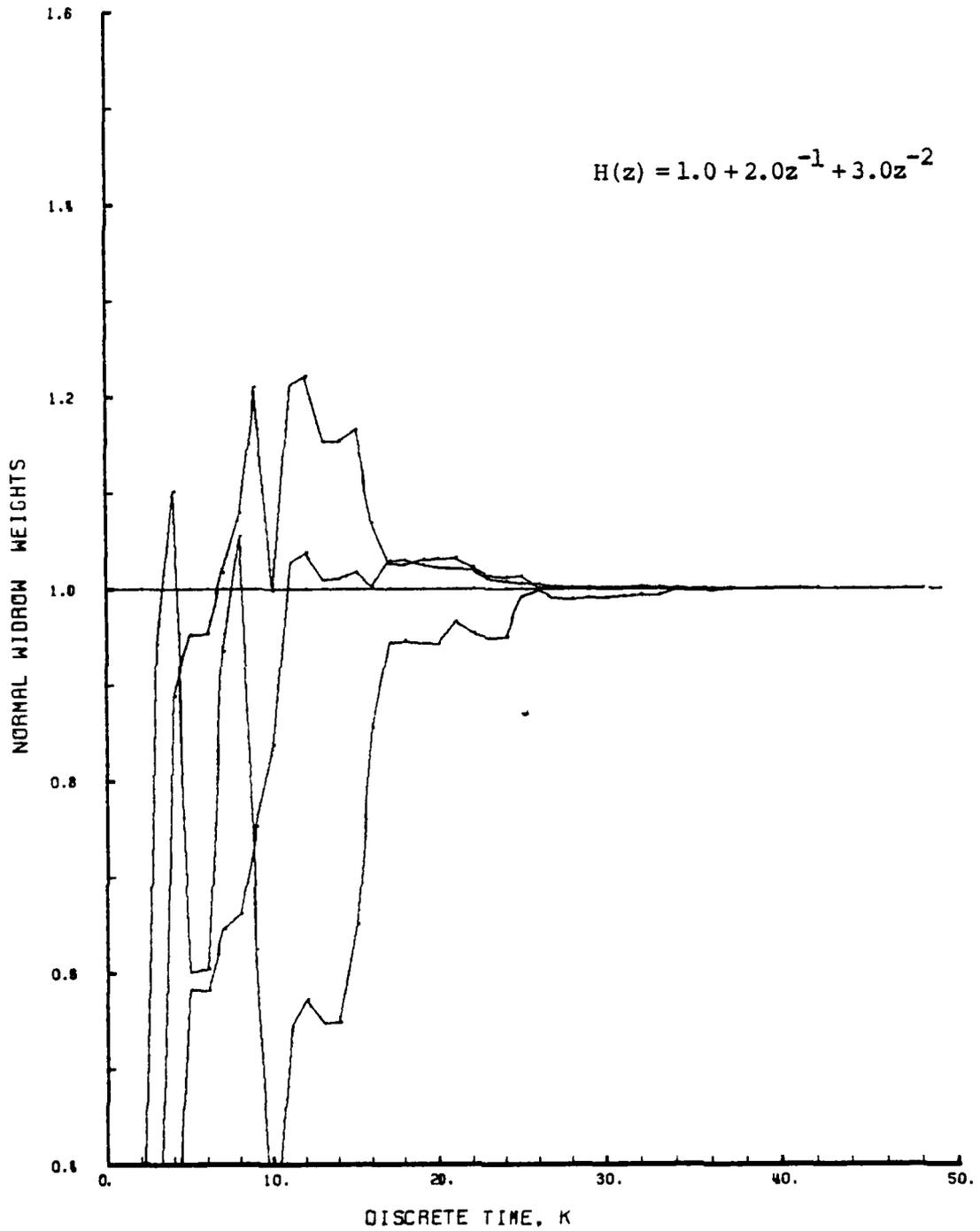


Figure 8.2. LMS Adaptive Filter Weights,  $k_s = -.300$

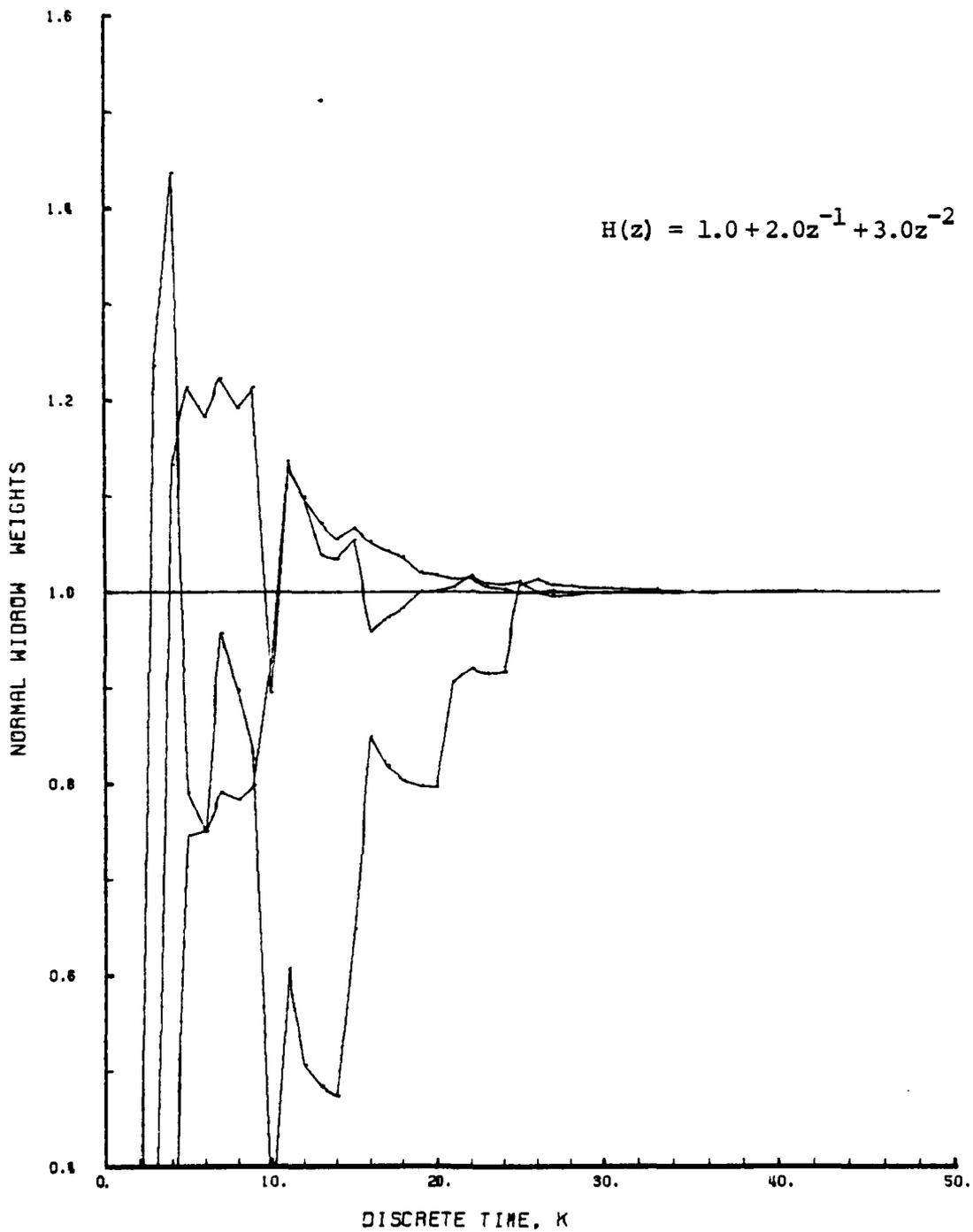


Figure 8.3. LMS Adaptive Filter Weights,  $k_s = -.400$

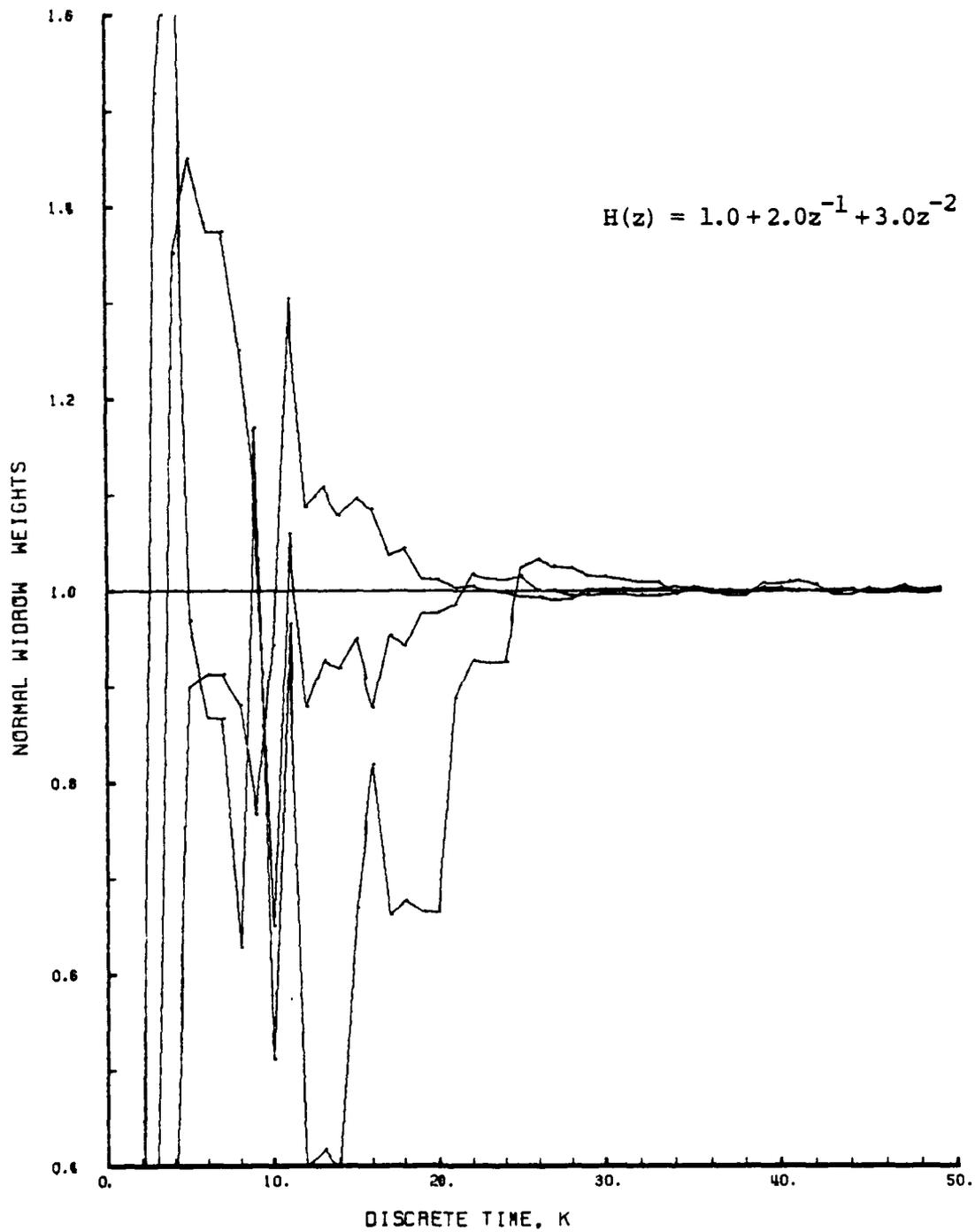


Figure 8.4. LMS Adaptive Filter Weights,  $k_s = -.500$

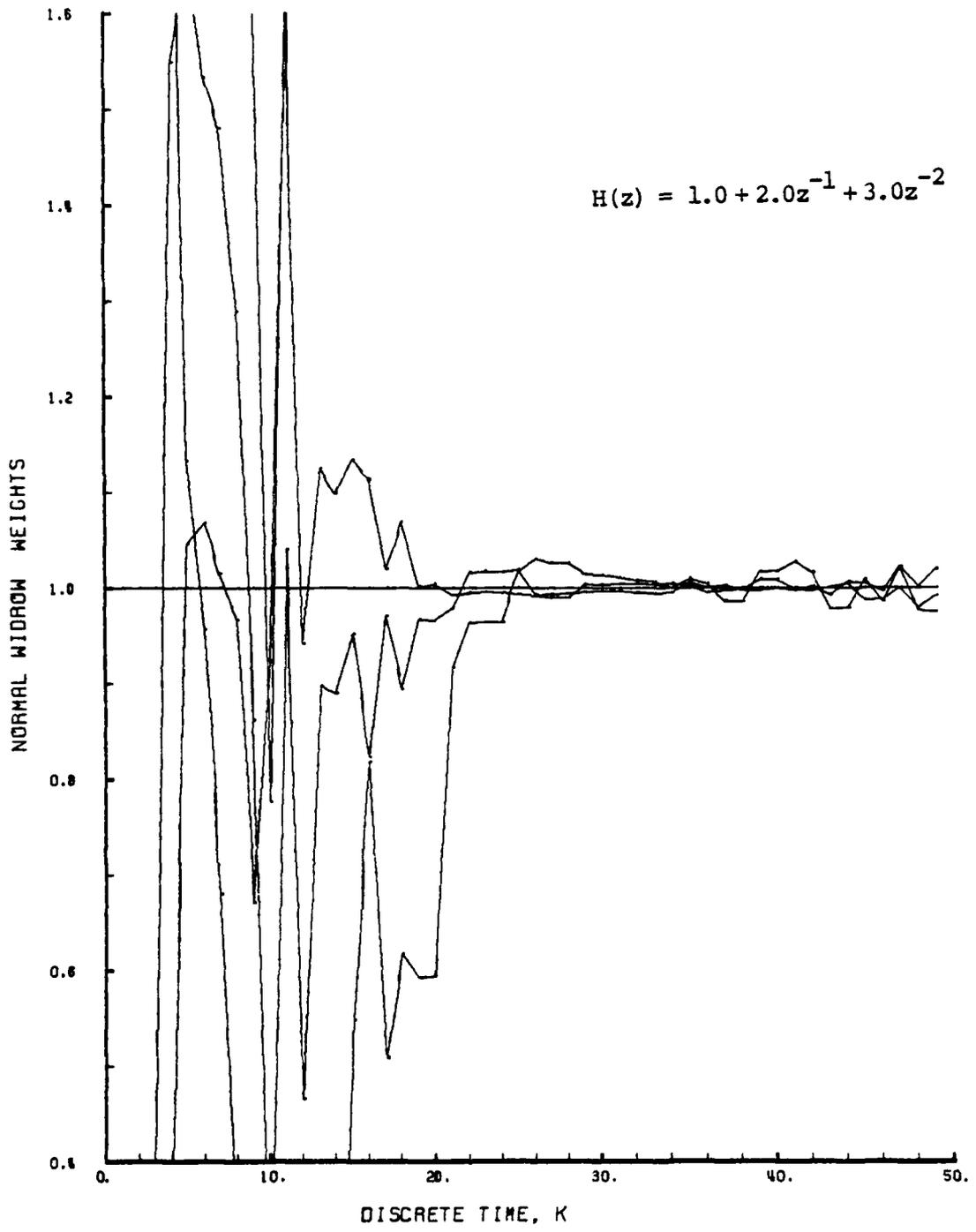


Figure 8.5a. LMS Adaptive Filter Weights,  $k_s = -.600$

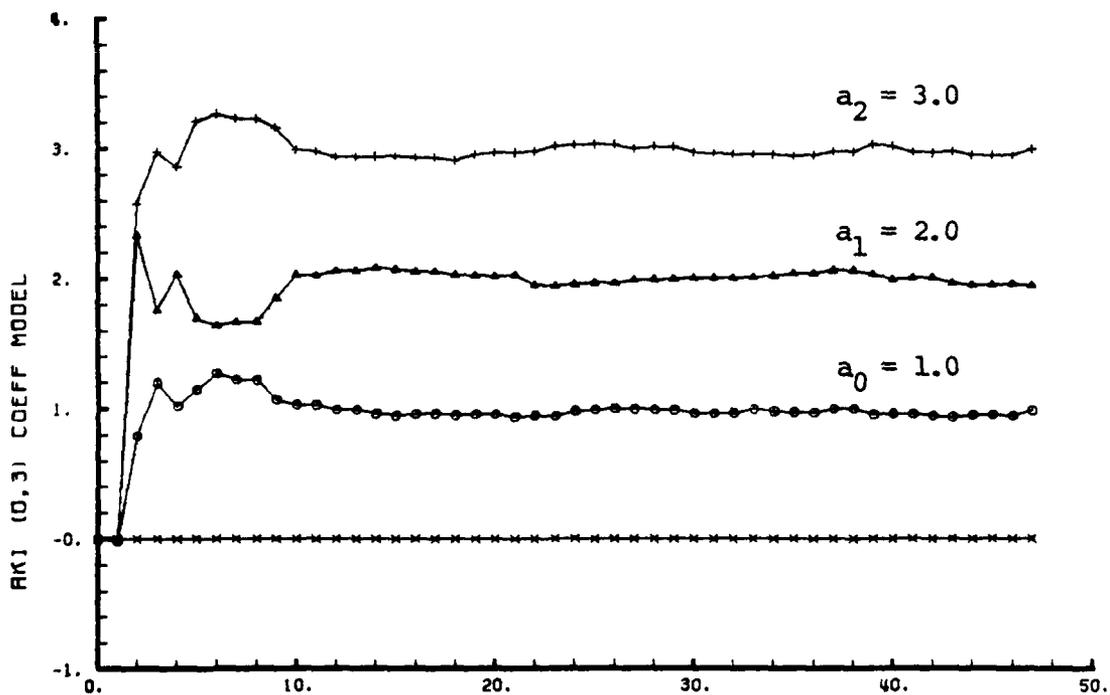
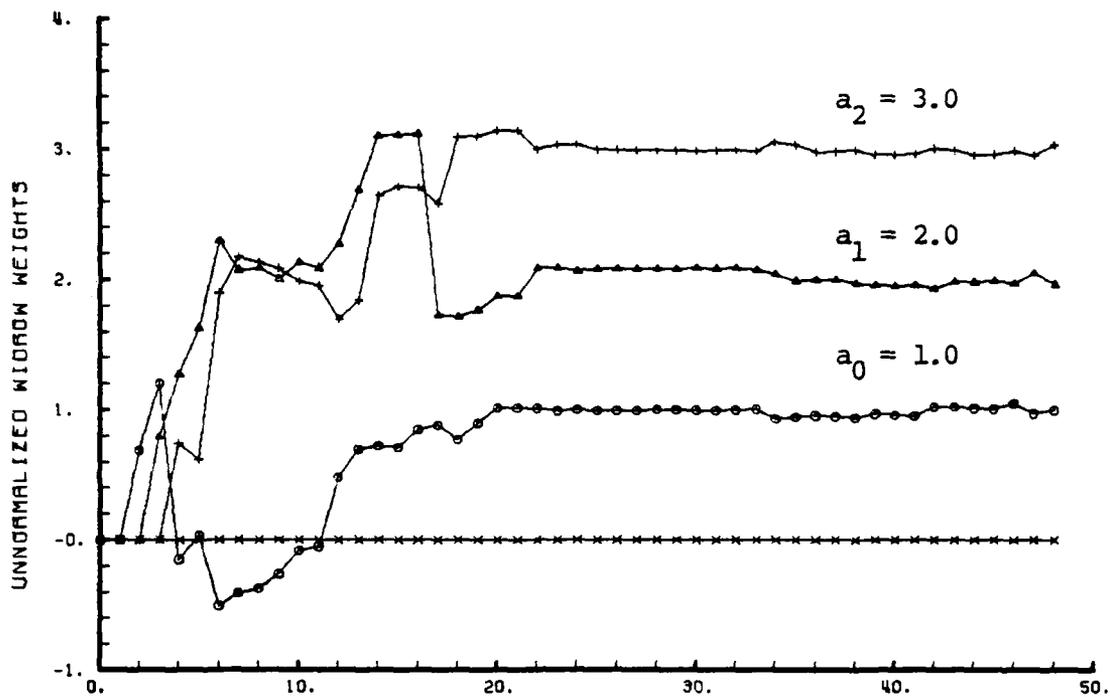


Figure 8.5b. LMS Adaptive Filter Weights,  $k_s = -.200$ , with increased measurement noise

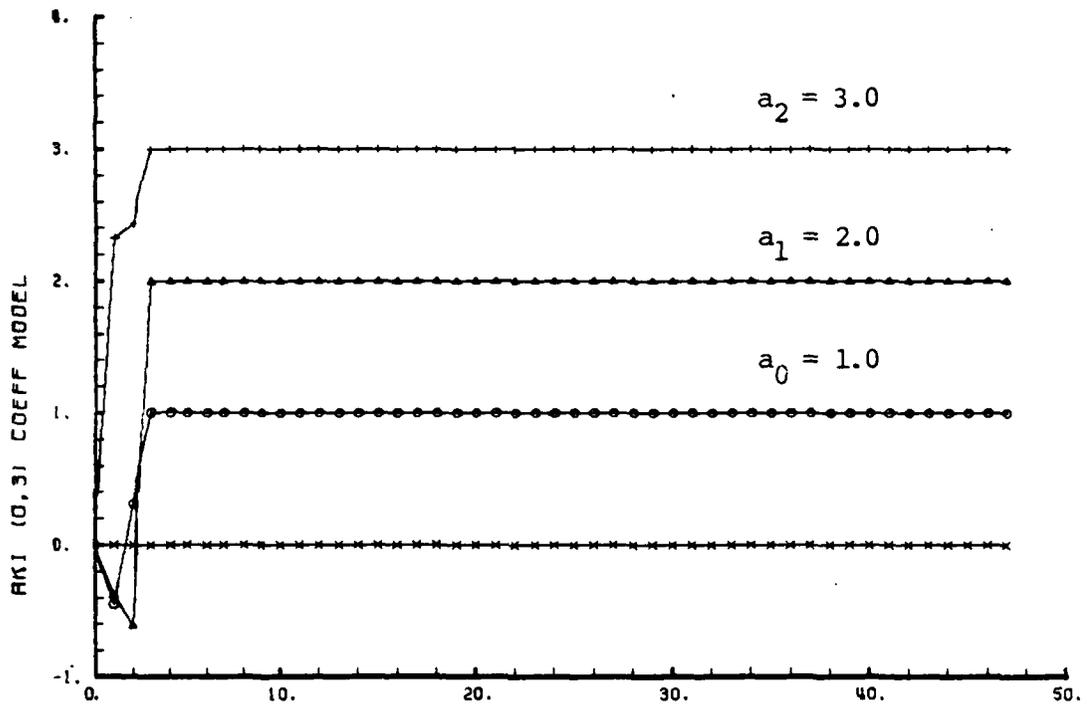


Figure 8.6. AKI Moving Average Plant Coefficients

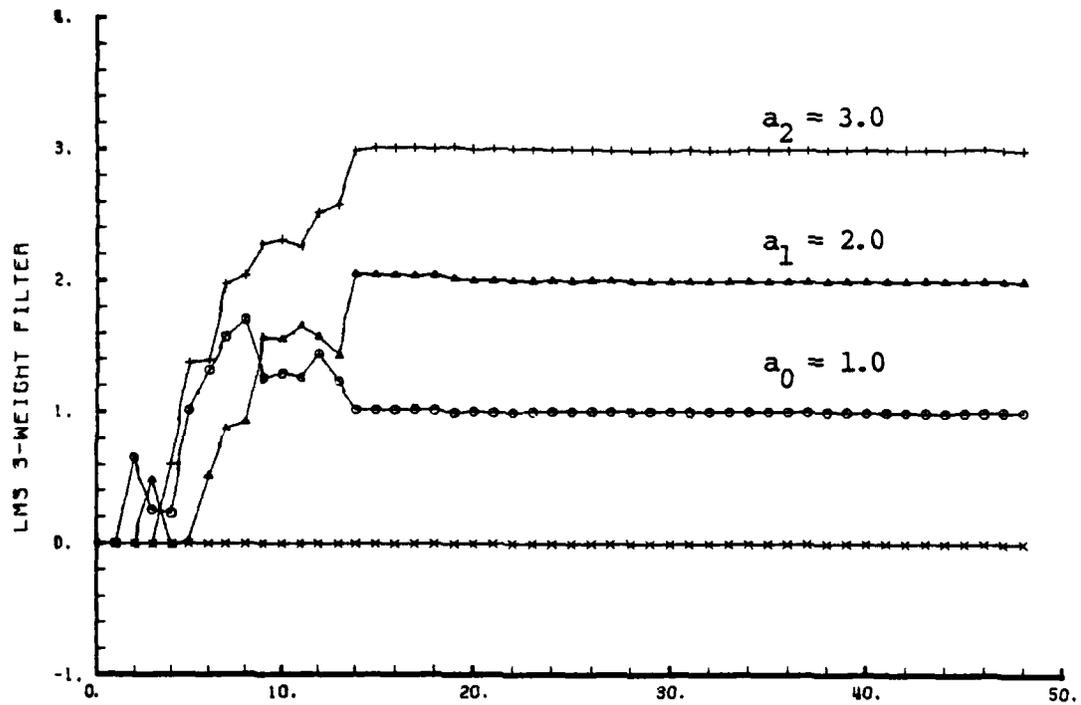


Figure 8.7. LMS Adaptive Filter Weights

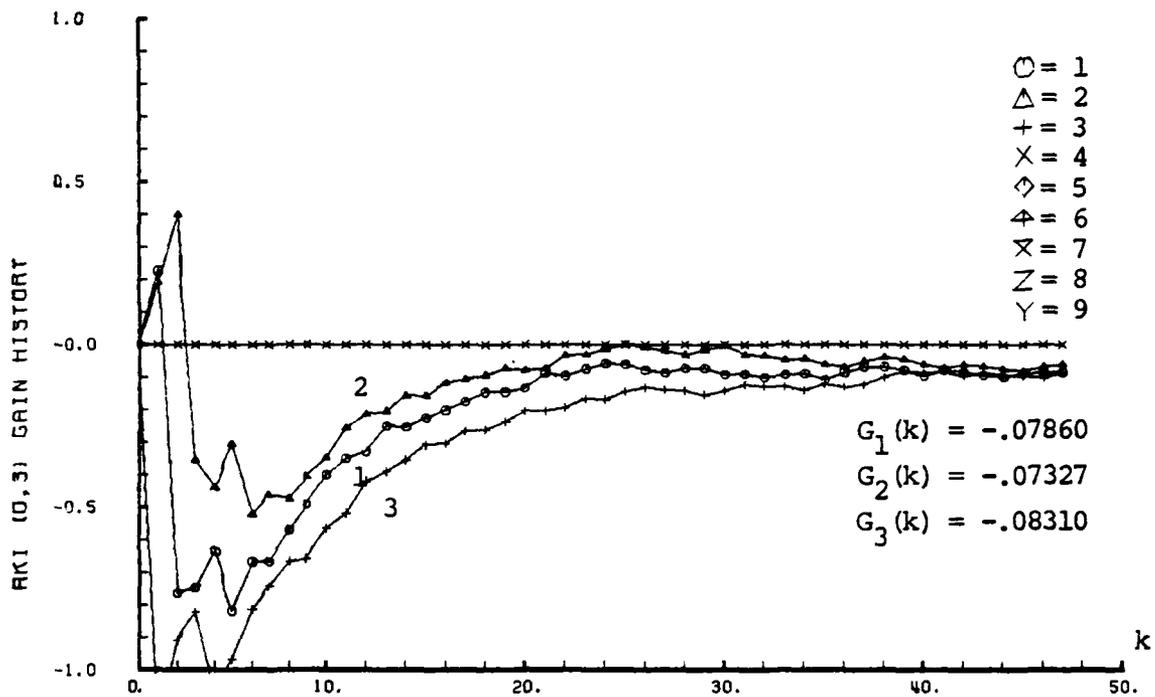


Figure 8.8. AKI Averaged Gains for Moving Average Model

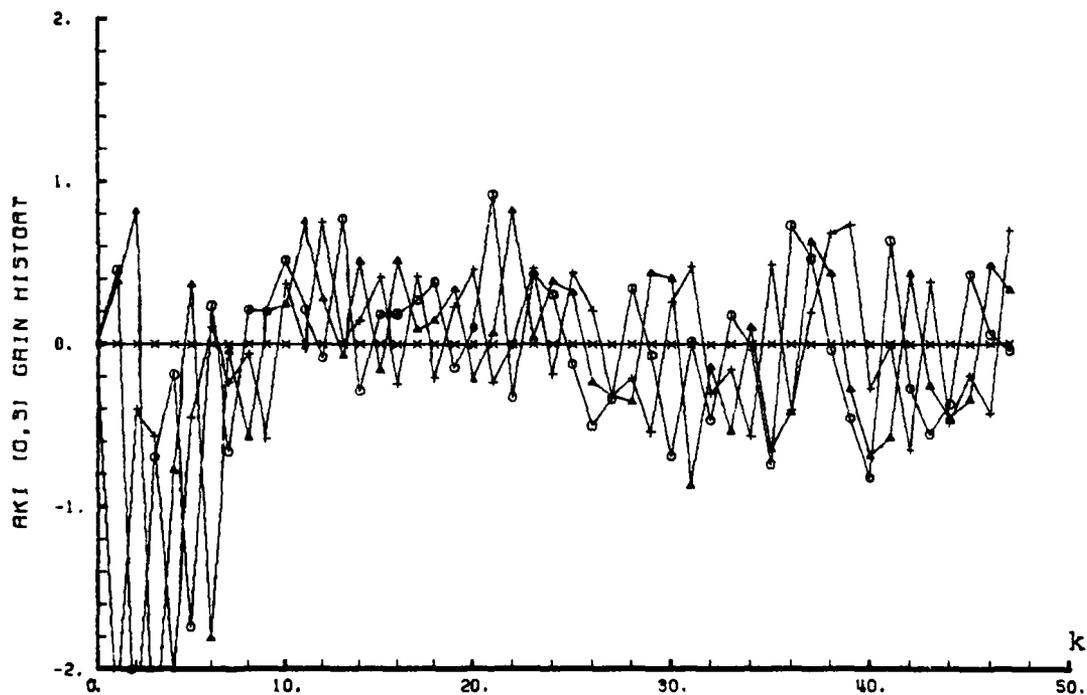


Figure 8.9. AKI Moving Average Instantaneous Gains

$E\{\underline{G}(k)\}$  is the present average value for the gain

$\underline{G}(k)$  is the present value of the instantaneous gains

$\frac{1}{k+1}$  is the averaging gain.

This method of obtaining a "running" average is generally well known, (see for example Ref. 39). The example presented was not the only one used but a good representative of the operation of the AKI when the data was derived from a moving average process.

## 2. AKI Applied to Autoregressive Data

Before applying the AKI to identifying the coefficients of a complex ARMA plant, it was first tested using data derived from a plant whose transfer function specified an autoregressive process. The transfer function used was,

$$\begin{aligned} H(z) &= \frac{1}{1.0 + \frac{26.0}{24.0}z^{-1} + \frac{9.0}{24.0}z^{-2} + \frac{1.0}{24.0}z^{-3}} \\ &= \frac{a_0}{1 - b_1z^{-1} - b_2z^{-2} - b_3z^{-3}} \end{aligned} \quad (8.3)$$

The AKI had no problem converging on the plant coefficients (Figure 8.10) producing the following estimates at the 42<sup>nd</sup> iteration (Table 8.1). It essentially converged on the plant coefficients in approximately five iterations. The gains computed by the AKI for this case also displayed convergence to a steady state value. The gain history for this example is depicted in Figure 8.11.

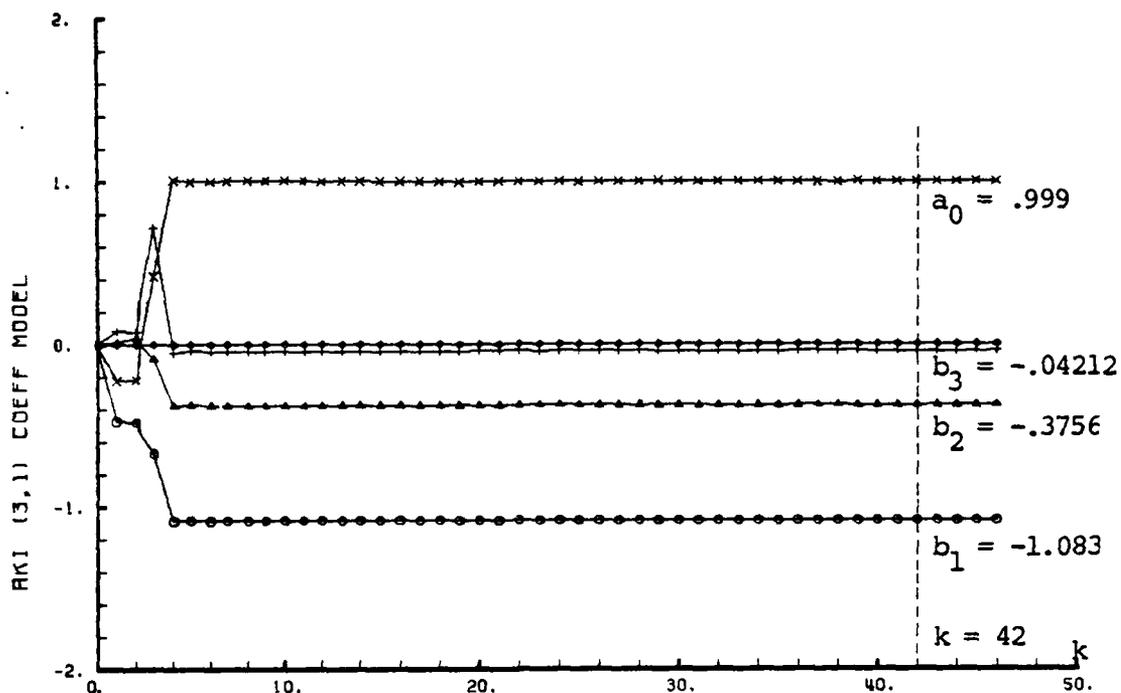


Figure 8.10. AKI Autoregressive Plant Coefficients

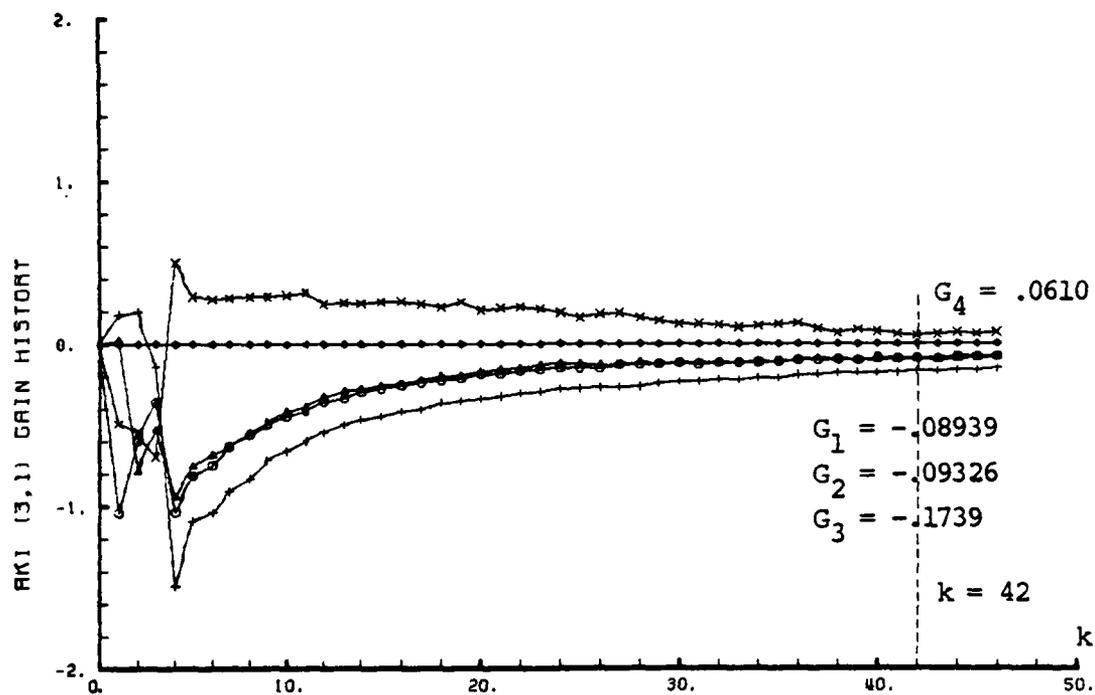


Figure 8.11. AKI Autoregressive Averaged Gains

Table 8.1

ACTUAL VS AKI ESTIMATES FOR COEFFICIENTS OF EQUATION 8.3

	ACTUAL	AKI ESTIMATES
$a_0$	1.0	0.9995
$b_1$	-1.08333 ...	-1.083
$b_2$	-0.3750	-0.3756
$b_3$	-0.04166 ...	-0.04212

### 3. AKI Applied to ARMA Data

The next logical step was to use the AKI to identify the coefficients of a general ARMA process. One of Perry's models [Ref. 6] was used for this purpose. Specifically, the transfer function of the plant was,

$$\begin{aligned}
 H(z) &= \frac{1.0 + 1.4z^{-1} + .98z^{-2}}{1 - 1.14z^{-1} + 1.4549z^{-2} - .88490z^{-3} + .40745z^{-4}} \\
 &= \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 - b_1z^{-1} - b_2z^{-2} - b_3z^{-3} - b_4z^{-4}} \quad (8.4)
 \end{aligned}$$

The ARMA plant was subjected to the same conditions which Perry [Ref. 6] describes. That is, unit variance, zero mean, white gaussian noise was used as the input. The reader is reminded that the output signal processed by the AKI was corrupted by measurement noise as described in Chapter V. The output data used in Perry's examples, however, reflects noiseless measurements. Table 8.2 tabulates the results for the coefficient estimates computed by the AKI. Even though the results presented in Table 8.2 represent the coefficient

Table 8.2

ACTUAL VS AKI ESTIMATES FOR COEFFICIENTS OF EQUATION  
8.4 AT THE 371ST ITERATION

	ACTUAL	AKI ESTIMATES
$a_0$	1.00000	0.98760
$a_1$	1.40000	1.40100
$a_2$	0.98000	0.99280
$a_3$	0.00000	0.01252
$a_4$	0.00000	-0.00922
$b_1$	1.14000	1.14000
$b_2$	-1.45490	-1.45900
$b_3$	0.88490	.88740
$b_4$	-0.40745	-.40980

estimates at the 371<sup>st</sup> iteration, it can be seen from Figure 8.12 that the AKI has essentially converged by the 28<sup>th</sup> iteration. We note also the characteristic convergence of the averaged gains to steady state values in Figure 8.13. As a means of comparison with Perry's results, the poles and zeros of the AKI estimates at the 28<sup>th</sup>, 90<sup>th</sup> and 371<sup>st</sup> iteration are plotted in Figure 8.14a and Figure 8.14b. Perry's results Figure 3.7 [Ref. 6:pp. 107,108] for his lattice modeling of the plant represented by equation (8.4) are reproduced for convenience.

As was noted for the previous cases, the instantaneous gains appeared erratic, Figure 8.15, whereas the averaged gains converged to some steady state value.

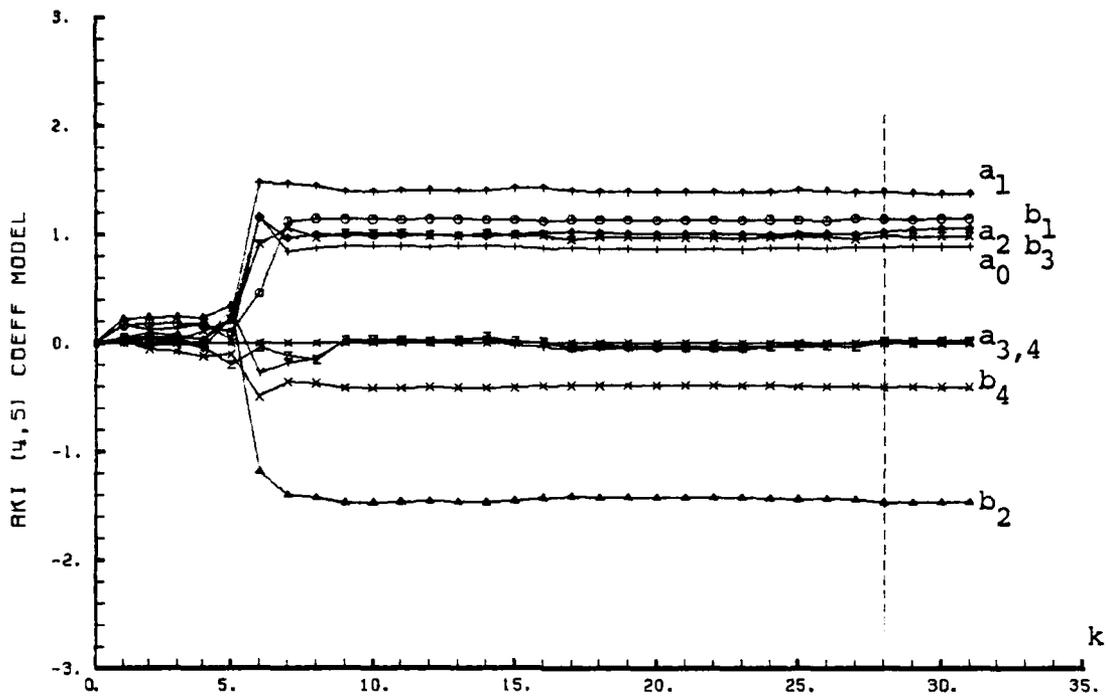


Figure 8.12. AKI(4,5) Computer Coefficients, Perry's Example

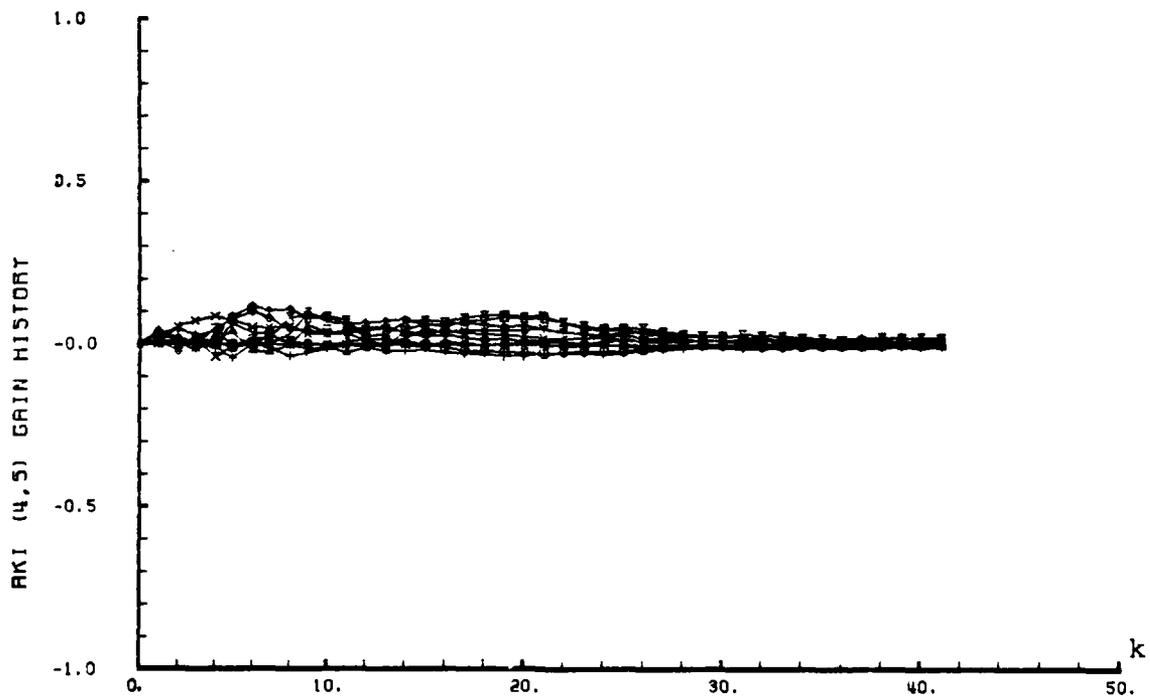
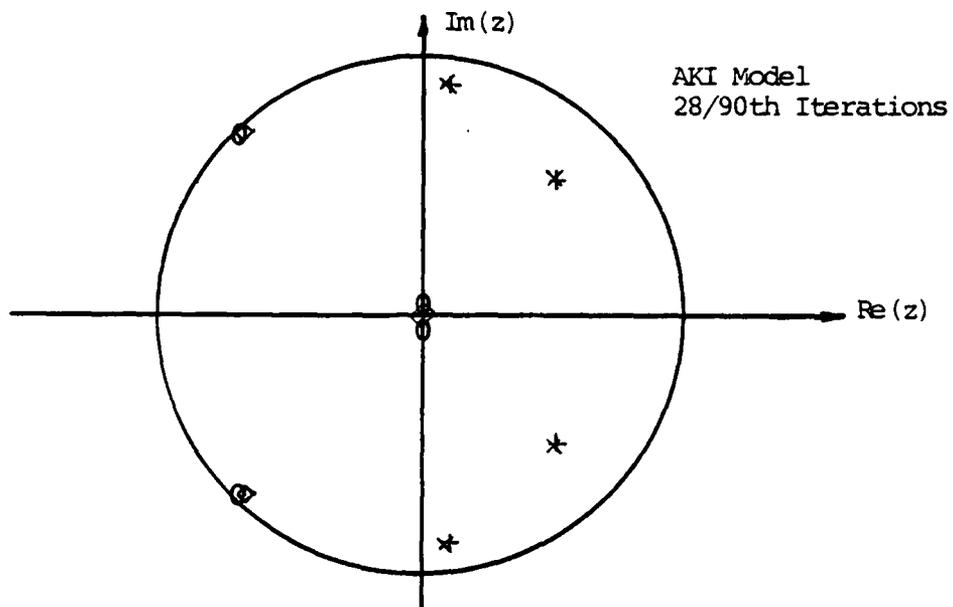


Figure 8.13. AKI(4,5) Averaged Gain Values





X Model Pole  
 O Model Zero  
 + System Pole  
 ◊ System Zero

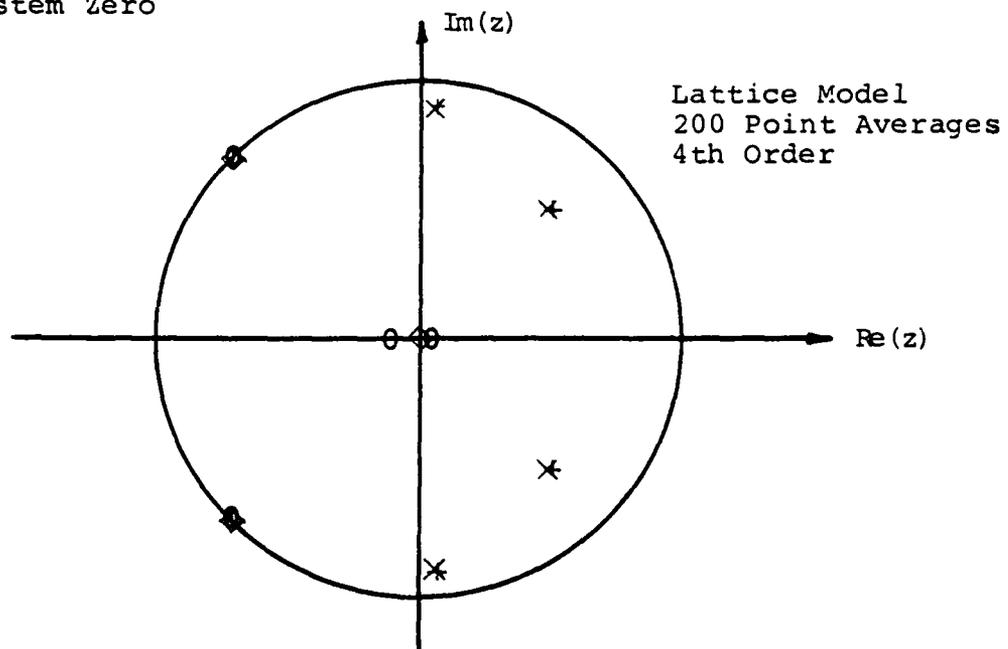


Figure 8.14a. Pole/Zero Models Produced by AKI and Lattice Algorithms for a Plant with the Characteristic Transfer Function of Eqn. (8.4)

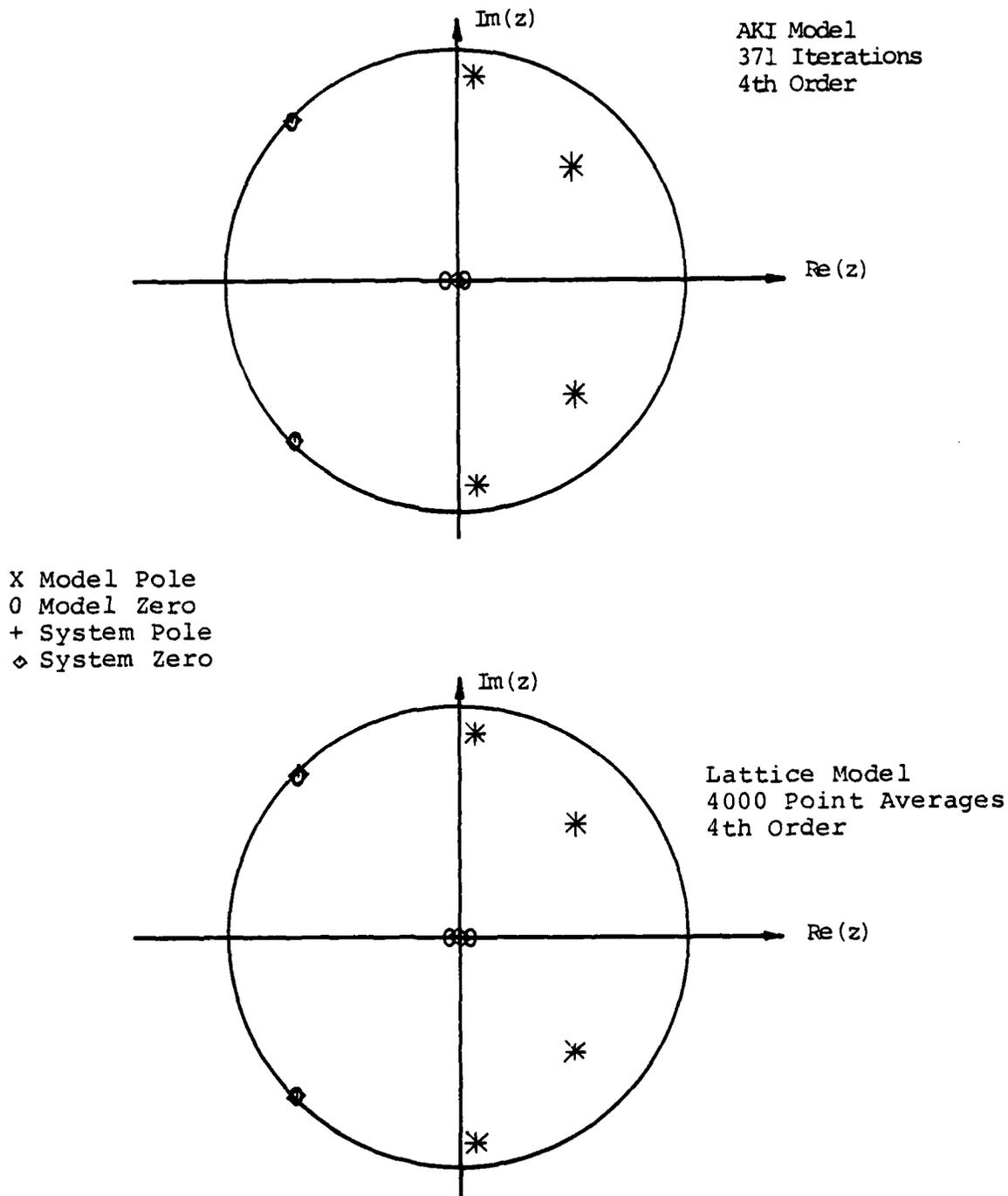


Figure 8.14b. Pole/Zero Models Produced by AKI and Lattice Algorithms for a Plant with the Characteristic Transfer Function of Eqn. (8.4) at 371st Iteration

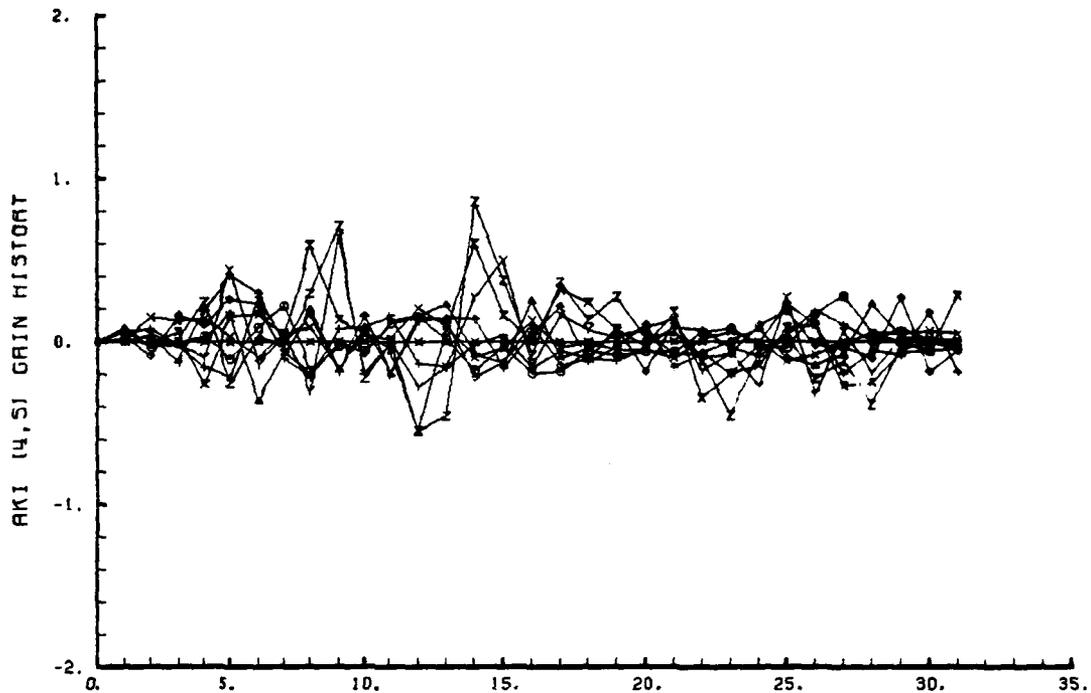


Figure 8.15. Instantaneous Gain Values, AKI(4,5) Model

#### 4. AKI vs Adaptive Recursive LMS Filter

Using Feintuch's proposed algorithm [Ref. 18] and repeating the simulation presented in the rebuttal to Johnson and Larimore [Ref. 19], a comparison was made between the operation of the AKI and the Adaptive Recursive LMS filter. The Adaptive Recursive LMS filter required "tuning" of the convergence factors,  $k_a$  and  $k_b$ , to what appeared to be the optimal performance features of (1) fast convergence and

(2) stable estimates. The "tuning" process was a trial and error procedure beginning with the convergence constants given by Feintuch,

$$k_a = -4.3 \times 10^{-4} \quad (8.5a)$$

$$k_b = -4.3 \times 10^{-4} . \quad (8.5b)$$

The Adaptive Recursive LMS algorithm was then implemented to estimate the coefficients of the plant whose transfer function was,

$$H(z) = \frac{0.05 - 0.40z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}} \quad (8.6a)$$

$$= \frac{a_0 + a_1z^{-1}}{1 - b_1z^{-1} - b_2z^{-2}} \quad (8.6b)$$

The best response obtained by trial and error resulted in the convergence constants,

$$k_a = -4.3 \times 10^{-3} \quad (8.7a)$$

$$k_b = -14.3 \times 10^{-3} . \quad (8.7b)$$

Feintuch reported that after  $8,192^2$  iterations the estimates had converged on the coefficients of (8.6a) with a .2096 normalized rms error. Using the convergence constants (8.7) convergence was essentially reached by the  $4,000^{\text{th}}$  iteration.

---

<sup>2</sup>Feintuch reports the resulting estimates for the coefficients at several intermediate iterations from 8,192 to 65,536, however 8,192 was the minimum number of iterations reported.

It was felt that the operation of the adequately tuned Adaptive Recursive LMS filter could be fairly compared with the operation of the AKI.

Table 8.3 tabulates the performance of the two system identification methods and Figures 8.16 and 8.17 graphically present the responses. Figure 8.18 not only shows at what point the gains of the AKI reach a constant value but also

Table 8.3

ACTUAL VS AKI AND ADAPTIVE RECURSIVE LMS FILTER  
ESTIMATES FOR COEFFICIENTS OF EQUATION 8.6

	ACTUAL	AKI ESTIMATES (k = 150)	ADAPTIVE RECURSIVE LMS (k = 3,990)
$a_0$	0.05	0.05107	0.05045
$a_1$	-0.40	-0.4007	-0.3990
$b_1$	1.1314	1.13200	1.12800
$b_2$	-0.25	-0.25090	-0.2442

provides a measure of confidence that the unknown system has been identified. It is evident from Figure 8.19 that both methods seem to identify the poles and zeros of the actual plant; however, the poles computed by the AKI are closer. The zeros computed by the AKI are approximately .080 units farther from the true zero than is the Adaptive Recursive LMS filter estimate. All aspects considered, the AKI takes considerably fewer iterations to arrive at its estimate.

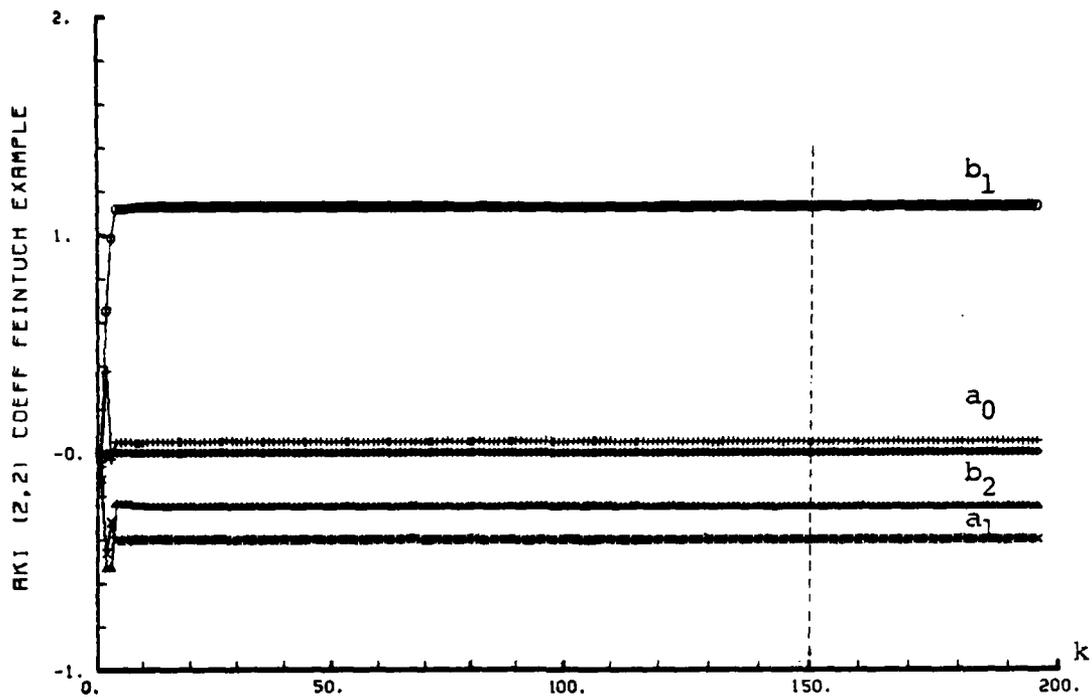


Figure 8.16. AKI(2,2) Computed Coefficients, Feintuch Example

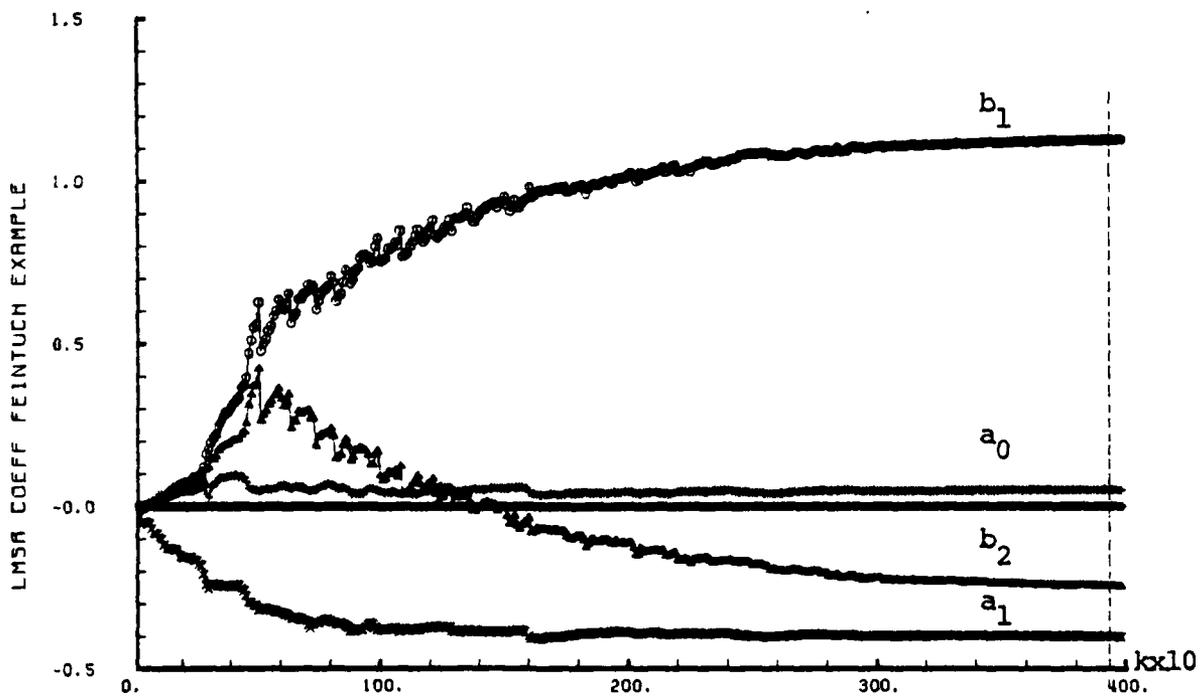


Figure 8.17. LMSR Computed Coefficients, Feintuch Example

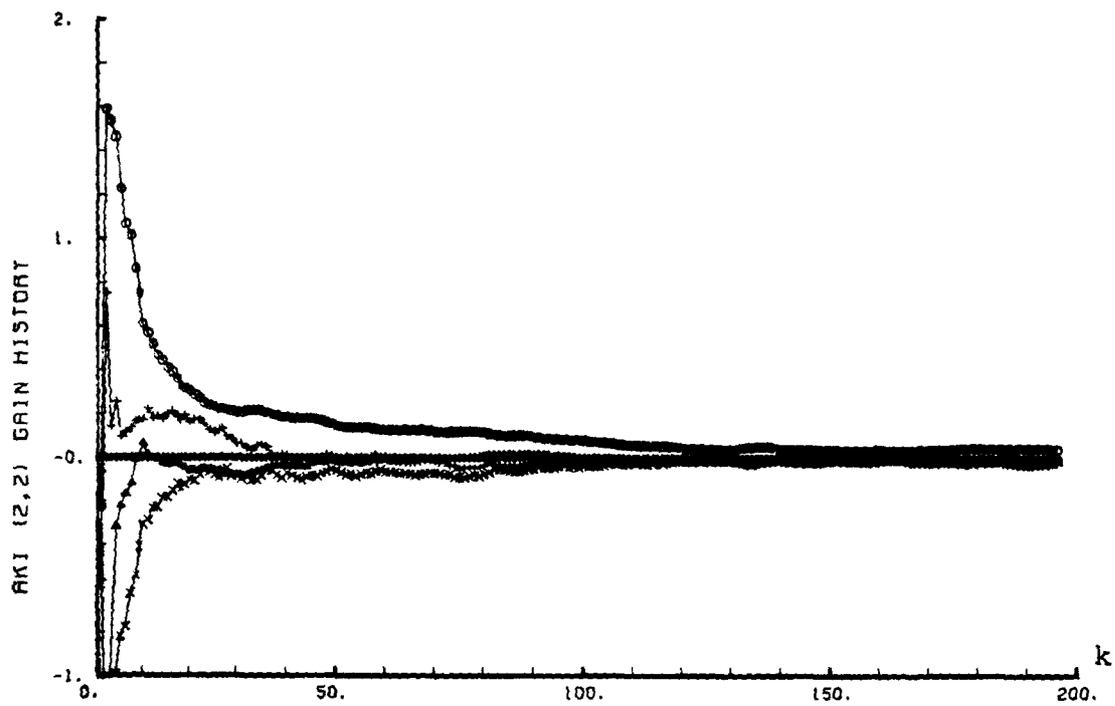


Figure 8.18. AKI(2,2) Averaged Gains, Feintuch Example

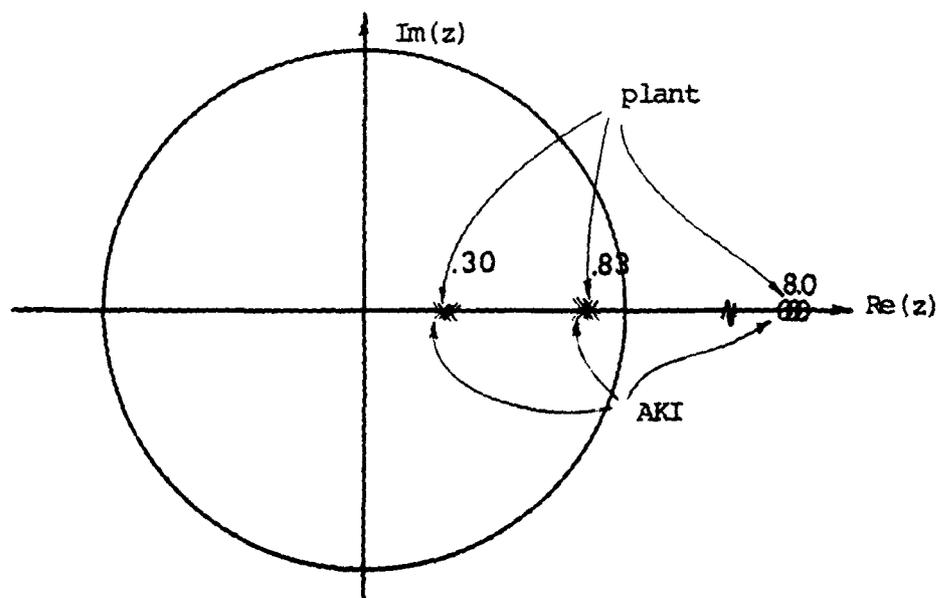


Figure 8.19. Pole/Zero Models Produced by AKI and Adapt. Rec. LMS Algorithms for a Plant with the Characteristic Transfer Function of Eqn. (8.6)

5. AKI Applied to PLL Data (Linear Region)

To identify the coefficients associated with the general autoregressive-moving average representation for the phase locked loop, the input noise signal power (mean square value) was kept at  $(10 \text{ deg})^2$ . In this manner, the linear region of the PLL was invoked. The input/output data was analyzed by the AKI resulting in the estimates shown in Table 8.4. Figures 8.20 and 8.21 show the response of the AKI

Table 8.4

ACTUAL VS AKI ESTIMATES OF THE ARMA  
REPRESENTATION FOR THE PLL (LINEAR REGION)

	ACTUAL	AKI ESTIMATES (k = 82)	AKI ESTIMATES (k = 350)
$a_0$	0.000000	-0.0001046	-0.000692
$a_1$	0.020000	0.0198500	0.0199800
$a_2$	-0.018890	-0.018720	-0.0188900
$b_1$	1.980000	1.980000	1.978000
$b_2$	-0.981110	-0.981000	-0.9790

when applied to the identification of the PLL data. It was noted that though the AKI correctly identified the coefficients of the linear PLL, the AKI gains were large due to the weak signals (input/output data) incorporated in the measurement vector,  $H(k)$ .

6. AKI Applied to PLL Data (Non-linear Region)

When analyzing any non-linear system the engineer must bring to bear all his analysis techniques on the problem. The PLL was therefore studied using classical root locus

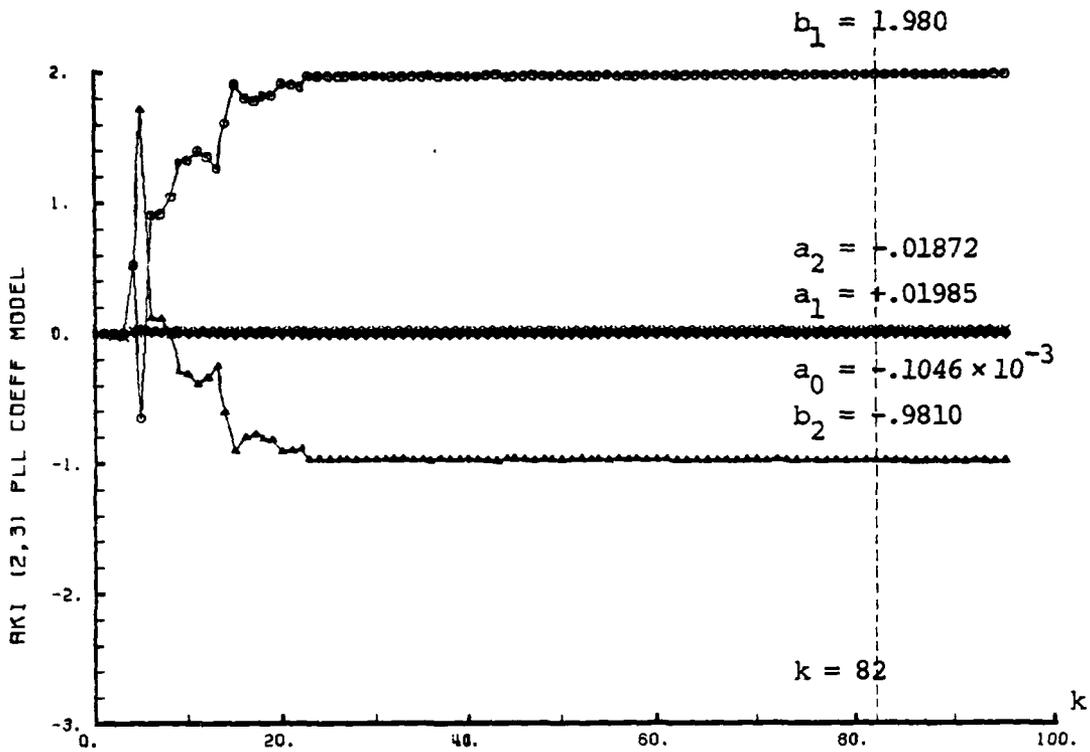


Figure 8.20. PLL, AKI(2,3) Computed Coefficients

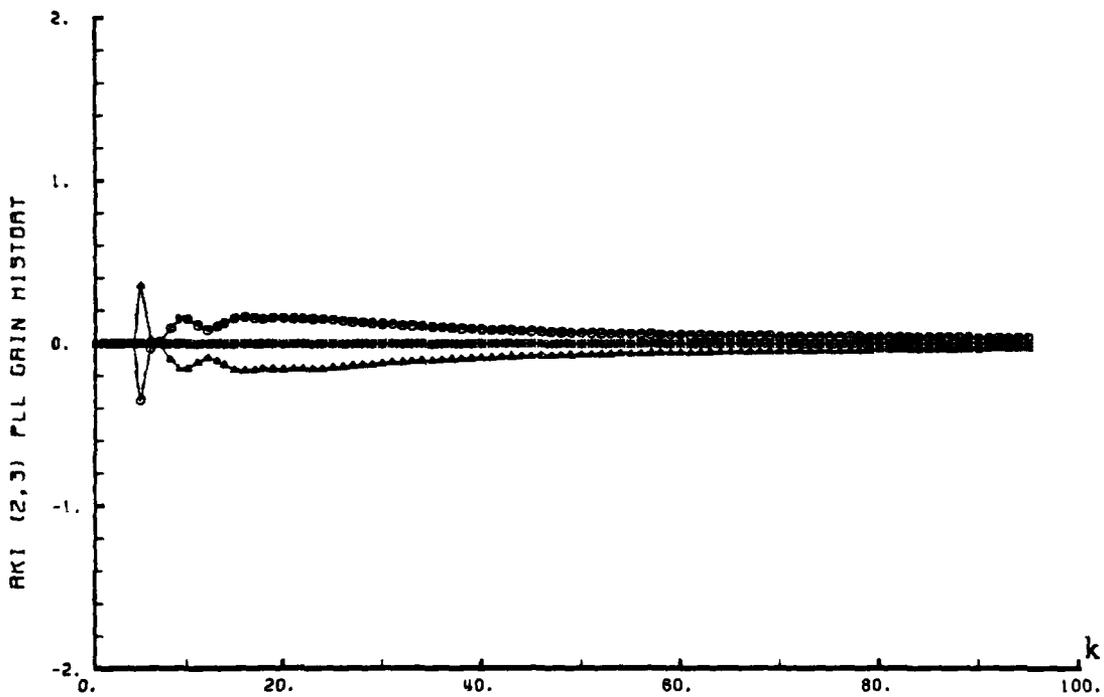


Figure 8.21. Averaged Gain Values, AKI(2,3) PLL Model

techniques to preview the possible outcomes when being modeled by the AKI. Stability analysis of a more complex PLL system using the root locus technique has been previously presented in the literature [Ref. 40].

The root locus technique can be applied to the PLL presented in this thesis by first assuming that the  $\sin(\cdot)$  block of Figure 5.7 is a variable gain,  $\lambda$ . This is not a restrictive assumption since the PLL during operation generally tracks small deviations. The loop gain can be written by inspection as,

$$\begin{aligned} L(z) &= \frac{\lambda(.02)(1 - .9445z^{-1})z^{-1}}{(1 - z^{-1})^2} & (8.8) \\ &= \frac{\lambda(.02)(z - .9445)z}{(z - 1)^2} \end{aligned}$$

From equation (8.8) the root locus of  $L(z)$  is drawn as shown in Figure 8.22. Even though the root locus technique is generally used when the signals in the system are considered deterministic, it is not surprising that some of the results obtained are nevertheless valid. When a moderately strong input noise identification signal  $[E\{u^2(u)\} < (25 \text{ deg})^2]$  was used, the pole-zero locations of the PLL seem to follow the classical root locus analysis. However, when the input noise identification signal power is increased beyond  $(25 \text{ deg})^2$  the pole-zero locations do not follow the expected behavior predicted using the root locus method as can be seen in Figure 8.22.

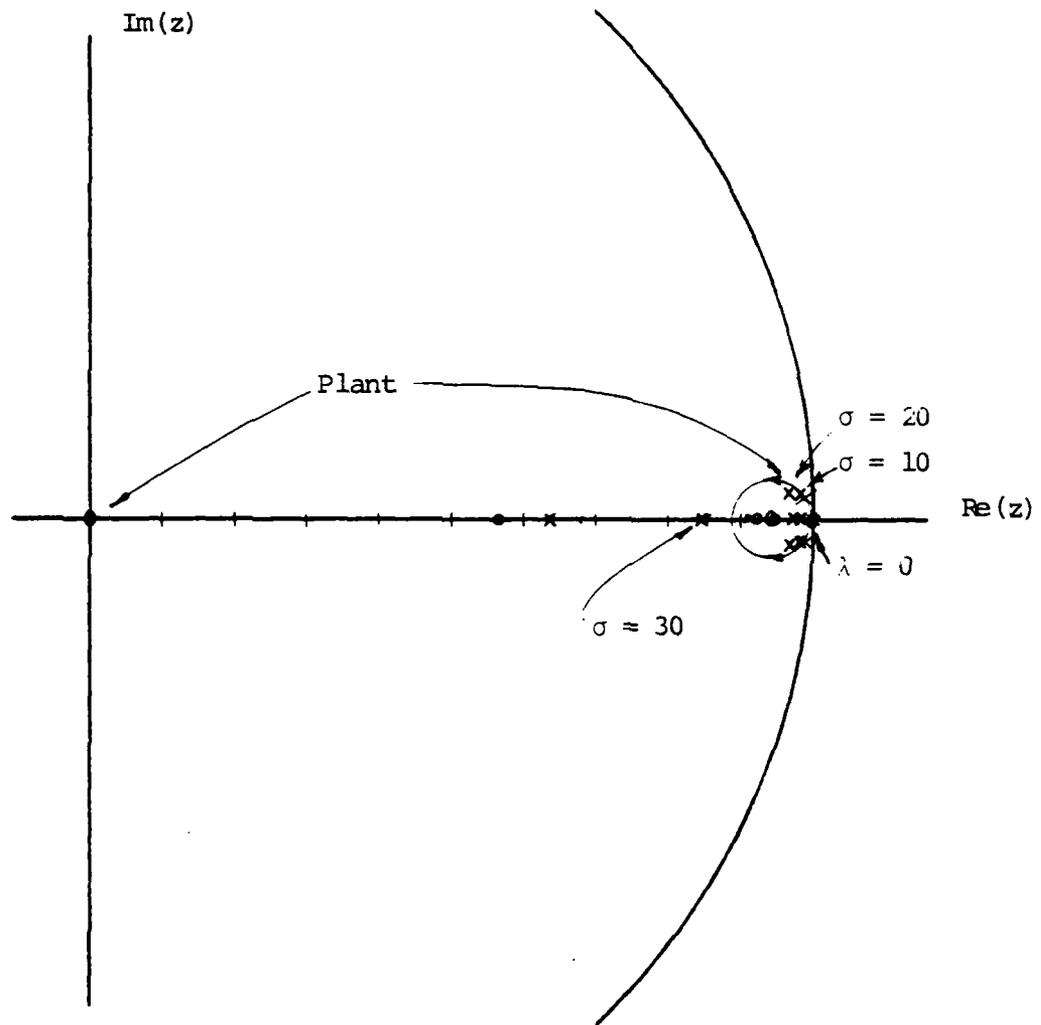


Figure 8.22. PLL Root Locus Analysis vs Computer Roots of the AKI ARMA Model

The departure of the pole-zero behavior from what was expected was analyzed by closer investigation of the linear terms of equation (7.8).

$$\begin{aligned} y(k) = & \text{lin} \quad .02\alpha u(k-1) - .01889\alpha u(k-2) + (2.0 - .02\alpha)y(k-1) \\ & + (.01889\alpha - 1)y(k-2) \end{aligned} \quad (8.9)$$

Recalling that a third order Taylor series approximation of the sine is the functional expressed by equation (7.7),

$$\sin(x) \approx \alpha x + \beta x^3 \quad (7.7)$$

one notes that the linear region is described when  $\alpha = 1$  and  $\beta = 0$ . It is therefore reasonable to study the variation of  $\alpha$  with respect to input noise power.

The average value  $\bar{\alpha}$  of  $\alpha$ , was computed by equating the estimated AKI coefficients to the coefficients of like terms in equation (8.9) for several input noise power levels. The relation used was

$$\bar{\alpha} = \frac{1}{4} \left[ \frac{2.0 - \hat{b}_1}{.02} + \frac{1.0 + \hat{b}_2}{.01889} + \frac{\hat{a}_1}{.02} - \frac{\hat{a}_2}{.01889} \right] \quad (8.10)$$

at the 350th iteration. The relationship between  $\bar{\alpha}$  and the input noise power level is readily apparent from Figure 8.23. This result is plausible since if one considers the input-output relationship of the sine block of Figure 5.7 one obtains Figure 8.24. Superimposing the gaussian probability functions of the different input noise signals, it can be seen that for

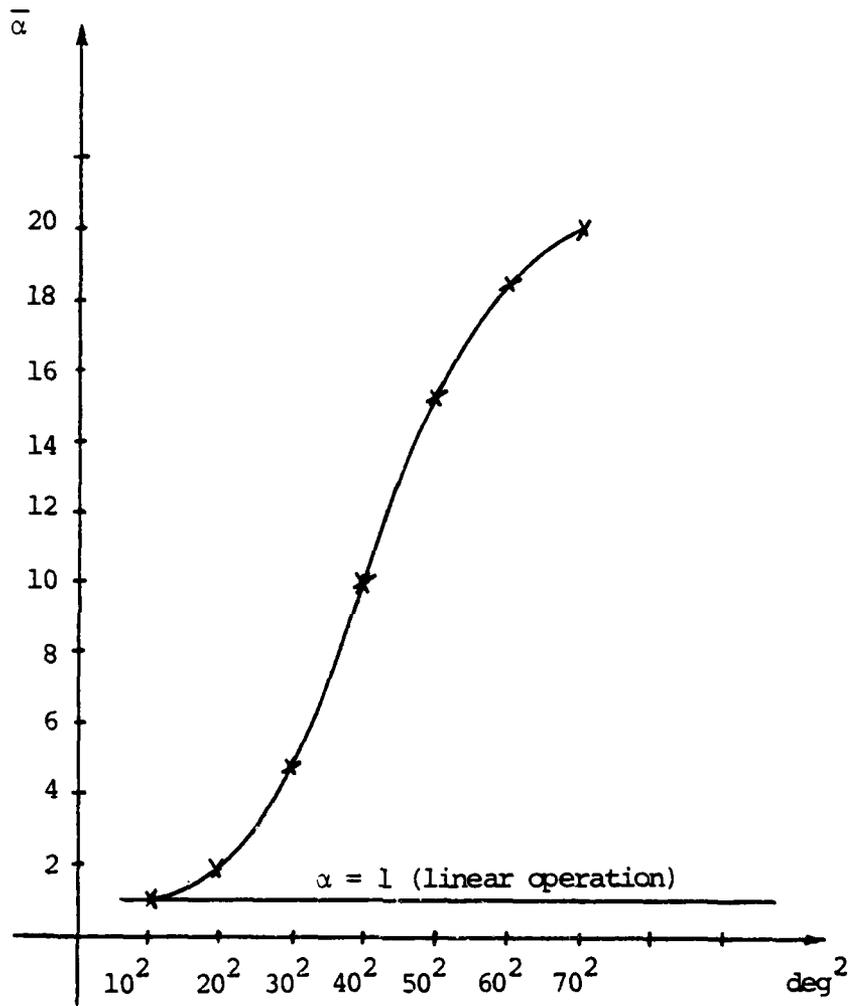


Figure 8.23. Variation of  $\bar{\alpha}$  vs Input Noise Signal Power

low power levels of the input signal, identification of the linear parameters of the overall system (equation 5.9) can be made. As the input power is increased beyond  $(25 \text{ deg})^2$  the input-output characteristics of the sin x block are no longer approximately linear, showing its effect in Figure 8.24 as a departure from its linear operation,  $\bar{\alpha} = 1$ . Therefore, by monitoring  $\bar{\alpha}$  one can determine when the overall system is entering its non-linear operating regime.

Since the same functional dependence between each of the AKI estimates and  $\bar{\alpha}$  did not exist for all of the input noise powers considered, knowing  $\bar{\alpha}$  did not provide any information of the pole locations but did provide a measure of the degree of non-linear operation.

## B. ORDER OF THE UNKNOWN SYSTEM IS NOT KNOWN (OVERMODELING)

### 1. AKI vs LMS Adaptive Filter

Using the data derived by operating a plant with the transfer characteristic of equation (8.1) the operation of the AKI and the LMS adaptive filter was compared when the orders used in the identifier and the LMS filter were greater than the known process that generated it. The following overmodeling cases were studied:

- (1) MA model is greater than plant MA process
- (2) ARMA model is fitted to MA plant

When the MA model order is greater than that of the plant MA process, it was found that both the AKI and the LMS filter would compute coefficients close to zero for the higher

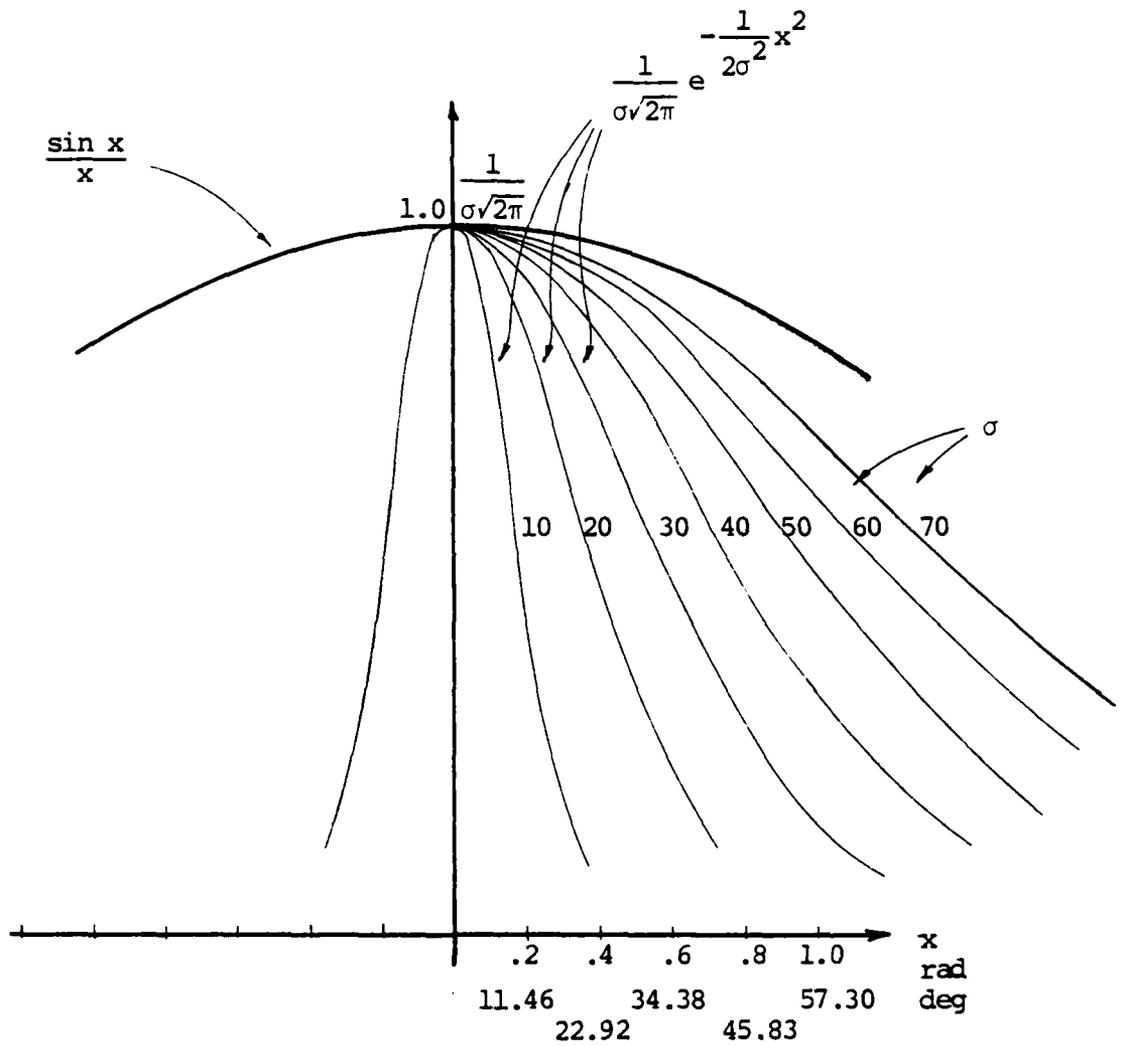


Figure 8.24. Comparison of  $\frac{\sin x}{x}$  and p.d.f. of input noise signals

order coefficients. This effect can be seen in Figures 8.25a and 8.25b when a fourth order MA process is used to model the actual second order process represented by equation (8.1). Since these are a "zeros only" plant and model the over-modeling essentially causes zeros to appear at the origin of the Z-plane of the model transfer function. Table 8.5 summarizes the results for two overmodeling conditions at the 250th iteration using the AKI. When a purely moving average process represented by equation (8.1) was modeled as an ARMA process, one must direct his attention to the poles and zeros of the model transfer function which the AKI computed and compare them to the actual plant poles and zeros. It was not readily apparent from the resulting coefficients that the plant had been identified. The following example will help clarify what is happening.

For the data produced by the second order plant (equation 8.1), an autoregressive moving average (ARMA) model of orders 2 and 5 respectively was fitted. It can be seen from Table 8.6 that no firm conclusions can be drawn about which coefficients actually identify the plant. We, the analysts, knowing the form of the plant which produced the data, could qualitatively state that  $b_1$ ,  $b_2$ ,  $a_3$ ,  $a_4$  and  $a_5$  are small enough to be ignored. Hence, we can identify the plant correctly. However, inspection of the poles and zeros of the transfer function which the AKI computed,

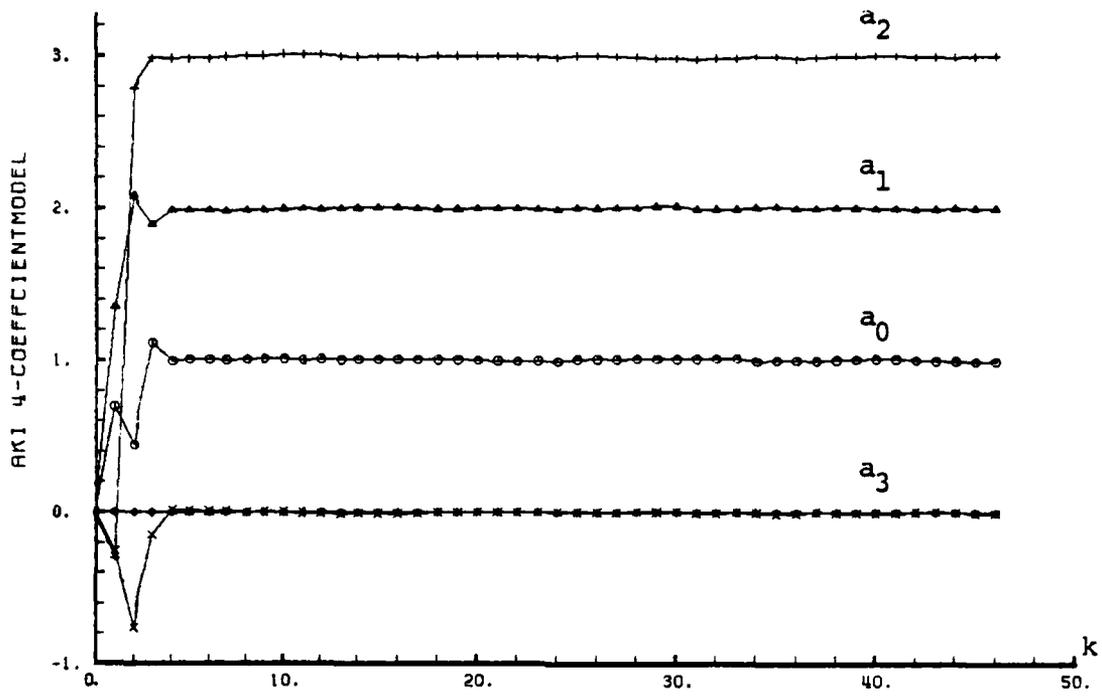


Figure 8.25a. AKI(0,4) Model of a Third Order MA Process

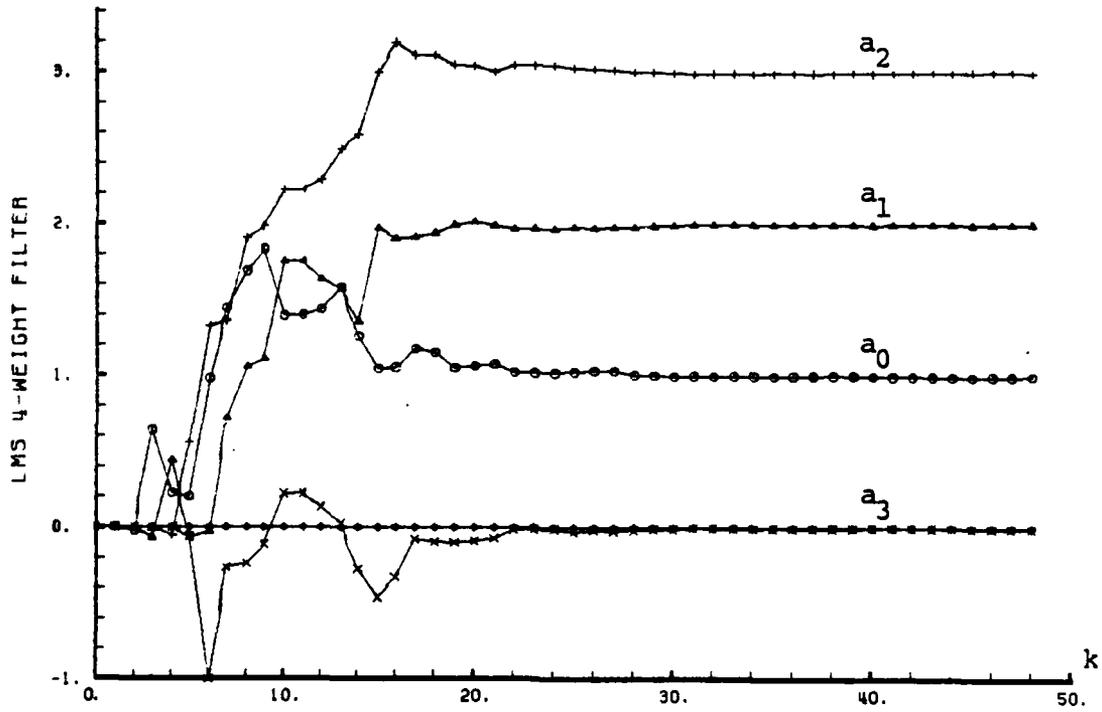


Figure 8.25b. Fourth Order LMS Filter Weights of a Third Order MA Process

Table 8.5

OVERMODELING OF A MA PROCESS USING THE AKI ( $k = 250$ )

	Actual Model Coeffs	3rd Order AKI MA Model	5th Order AKI MA Model
$a_0$	1.0	.9995	.9991
$a_1$	2.0	2.0000	2.000
$a_2$	3.0	3.0000	3.000
$a_3$	0.0	$.893 \times 10^{-3}$	.0003191
$a_4$	0.0	---	.001082
$a_5$	0.0	---	.0003198
zeros:	$-1.0 \pm 1.414j$	$-1.000 \pm 1.414j$	$-1.001 \pm 1.414j$
		$-.2977 \times 10^{-3}$	$.0227 \pm .0428j$
			-.4534

Table 8.6

OVERMODELING OF A MA PROCESS USING THE AKI OF ORDERS  
AR = 2, MA = 6

	Actual Plant Coefficients	AKI(2,6) Model Coefficients
$b_1$	0.0	.00467
$b_2$	0.0	.10500
$a_0$	1.0	1.0000
$a_1$	2.0	1.9940
$a_2$	3.0	2.8840
$a_3$	0.0	-0.2257
$a_4$	0.0	-0.3142
$a_5$	0.0	0.0001517

$$H_{AKI(2,6)}(z) = \frac{1.0 + 1.994z^{-1} + 2.8840z^{-2} - 0.2257z^{-3} - 0.3142z^{-4} + 0.001517z^{-5}}{1 - .00467z^{-1} - .1050z^{-2}} \quad (8.11)$$

results in:

poles: -.8217  
.3264  
zeros: -.9996 ± 1.414j  
.4827 × 10<sup>-3</sup>  
-.3212  
.3262

The observation to be made is that there are two pole-zero combinations which are near cancellation. This suggests that the AKI algorithm be rerun with the AR and MA orders reduced by at least two.

The implication of the analysis of this section is that a model can in theory be found for a given set of input/output data. Further, a parsimonious model can be identified by careful observation of pole-zero combinations which are near cancellation and of stray zeros near the origin.

## 2. AKI Applied to AR and ARMA Data

Essentially the same characteristic results found in Section VIII.B.1 were confirmed when the data produced by plants defined by the transfer functions of equations (8.3) and (8.4) were analyzed by the overmodeled AKI. That is, pole-zero pairs near cancellation and zeros near the origin

were produced by the AKI. Figures 8.26 through 8.28 have been chosen as representative pole-zero plots of the transfer functions computed by the AKI for the AR plant of Section VIII.A.2 and for Perry's model (Section VIII.A.3).

### C. CONCLUSION

This work indicates that the Kalman filter algorithm heretofore generally used as a state estimator, or in augmented form to estimate parameters (in which case the parameters are treated as states), can also be formulated in an adaptive manner to iteratively estimate the coefficients of an ARMA equation explicitly. This approach, termed the Adaptive Kalman Identifier (AKI), summarily identifies the unknown system whose input/output data is being processed. The LMS adaptive algorithm of Widrow, and its modification by Griffiths (in which the convergence factor is selected to be inversely proportional to the input signal power) are shown to be sub-optimal cases of the AKI. An additional insight provided by the AKI is that it indicates clearly how measurement noise might be taken into account in the LMS adaptive formulation.

The operation of the AKI was checked by way of simulation and compared with two existing identification techniques: (1) the LMS Adaptive nonrecursive (MA) algorithm and (2) the Adaptive Recursive ARMA LMS algorithm. It was found that not only are the two LMS filtering techniques special suboptimal cases of the AKI; but, further, the AKI exhibits superior convergence and modeling properties for the cases where (1) the

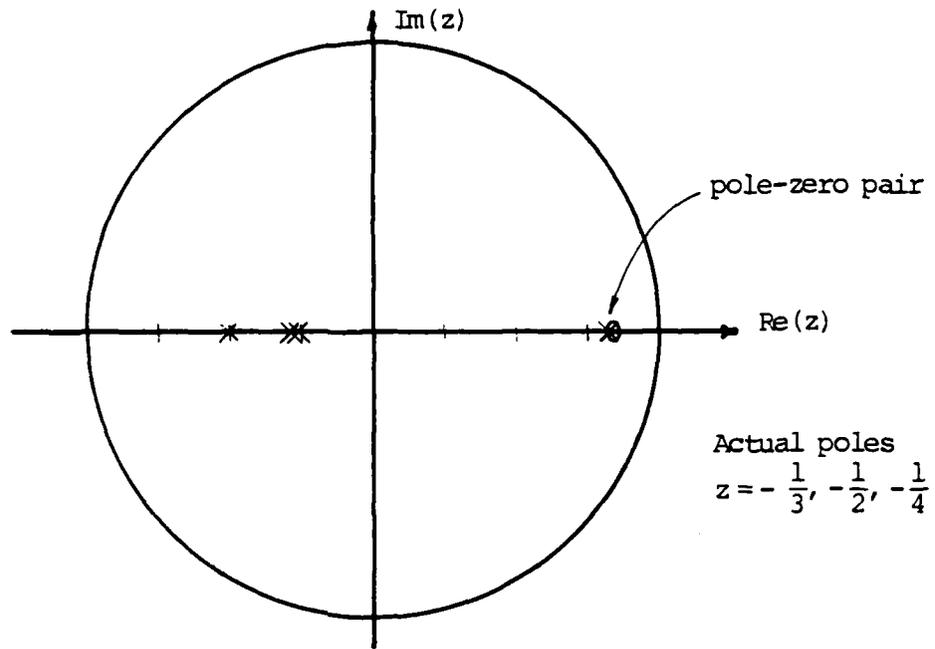


Figure 8.26. AKI(3,1) Overmodeled AKI(4,2)

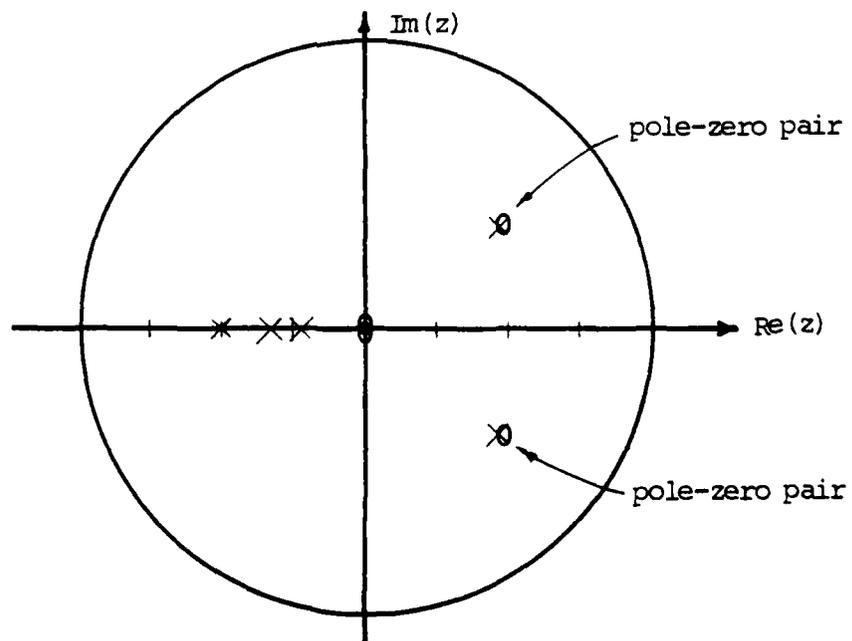


Figure 8.27. AKI(3,1) Overmodeled AKI(5,5)

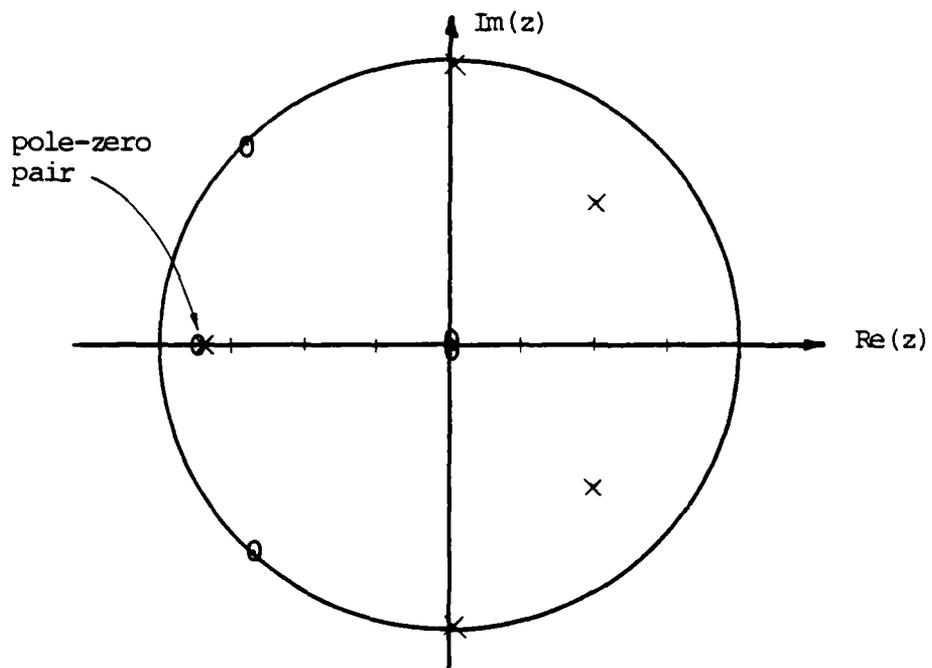


Figure 8.28. AKI(4,5) Overmodeled AKI(5,6)

order of the unknown plant is known and (2) the order of the plant is overmodeled. Additionally, the simulations indicate that accuracies similar to those obtained using lattice modeling methods, can be achieved using the AKI at a decidedly smaller number of iterations.

By making minor modifications to the measurement vector,  $H(k)$ , that is by using hybrid signals, the AKI was used to identify the linear and non-linear ARMA representations of a phase locked loop with success. Interestingly, the AKI technique appears to enable one to discern when a potential non-linear system enters its non-linear mode of operation, by closely monitoring the coefficients of the linear portion of the generalized non-linear ARMA model.

#### D. TOPICS FOR FURTHER CONSIDERATION

Several areas for further study directly and indirectly related to the AKI were uncovered. Foremost, a rigorous convergence proof is desirable. Although the connection was made between the AKI algorithm and the initial equation from which the lattice modeling algorithm is developed, similar comparisons (such as Chapters III and IV) could provide more insight into the operation of both. The multichannel AKI is the logical development of the single channel AKI presented. And, lastly, refinement of the AKI software (using the NL = 5 option) to include ARMA modeling of time series using a Box-Jenkins approach [Ref. 26] is feasible. A limited number of simulations using Monterey rain data and series C [Ref. 26]

data seem to indicate that an application exists for the AKI  
in this area.

## APPENDIX A

### THE DISCRETE WEINER PROBLEM

The discrete Wiener problem considers optimally filtering a desired signal from unwanted stationary noise. The criterion of optimality used is minimization of the mean squared error between the output and the desired signal. Generally one desires that the optimal filter, which is the device being sought, be time invariant and that it be able to accept a signal,  $s(k)$ , and noise,  $n(k)$ , where each are samples from stationary random processes. In other words, we want a device which can accept

$$x(k) = s(k) + n(k) \quad (\text{A.1})$$

as an input and produce at its output  $s(k+\Delta)$  or some linear function thereof where  $\Delta$  is some known delay.

Assuming that the desired output,  $d(k)$ , is the response to a specified sampled data linear system whose transfer function,  $H_d(z)$ , is given, then the error between the desired output and the output of the filter,  $y(k)$ , we seek is,

$$e(k) = d(k) - y(k). \quad (\text{A.2})$$

Figure A.1 depicts the discrete Wiener problem formulation. The derivation from here follows the one presented by Maybeck [Ref. 27] for the continuous case. From the block diagram we have,

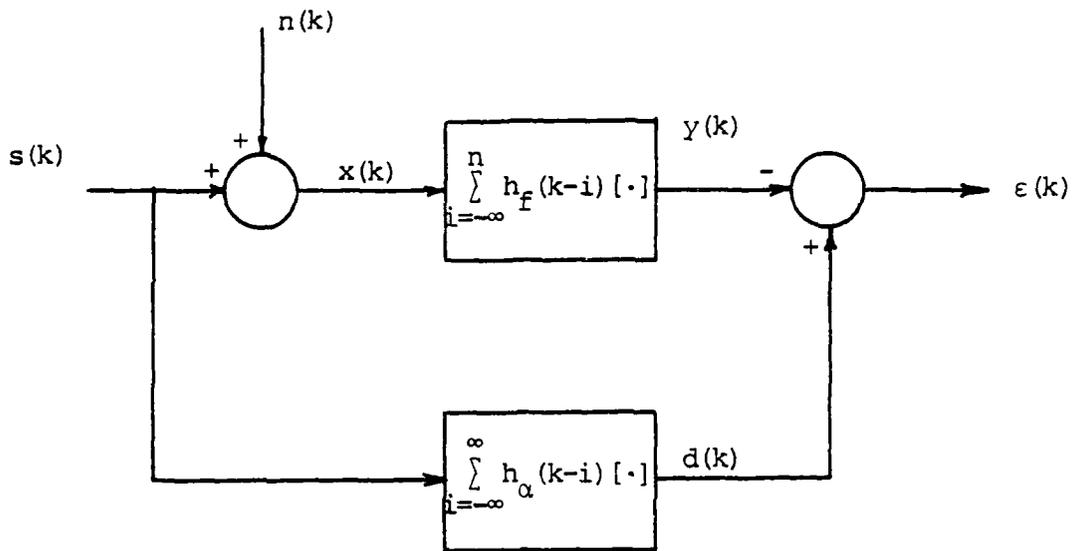


Figure A.1. The Discrete Wiener Problem

$$y(k) = \sum_{i=-\infty}^{\infty} h_f(k-i) x(i) \quad (\text{A.3})$$

or equivalently,

$$y(k) = \sum_{i=-\infty}^{\infty} x(k-i) h_f(i) \quad (\text{A.4})$$

Substituting equation A.4 into A.2 and squaring we have,

$$e^2(k) = d^2(k) - 2d(k) \sum_{i=-\infty}^{\infty} h_f(i) x(k-i) + \left[ \sum_{i=-\infty}^{\infty} x(i) h_f(k-i) \right] \left[ \sum_{i=-\infty}^{\infty} x(i) h_f(k-i) \right] \quad (\text{A.5})$$

Taking the expected value of A.5 we obtain,

$$\begin{aligned}
 E\{e^2(k)\} &= \psi_{dd}(0) - 2 \sum_{i=-\infty}^{\infty} h_f(i) \psi_{dx}(i) \\
 &+ \sum_{i=-\infty}^{\infty} h_f(i) \sum_{k=-\infty}^{\infty} h_f(k) \psi_{xx}(k-i)
 \end{aligned}
 \tag{A.6}$$

where notation  $\psi_{uv}(m)$  denotes the expected value of the product of  $u(k)$  and  $v(k+m)$ . That is,

$$\psi_{uv}(m) = E\{u(k)v(k+m)\} = E\{u(k)v(k-m)\}
 \tag{A.7}$$

otherwise known as the autocorrelation of  $u(k)$  and  $v(k)$ .

Using variational techniques and letting,

$$h_f(i) = h_{opt}(i) + \epsilon \Delta h(i)
 \tag{A.8}$$

substituting equation A.8 into equation A.6, taking the partial derivative with respect to epsilon,  $\epsilon$ , and setting the partial derivative equal to zero we have:

$$\sum_{n=-\infty}^{\infty} \Delta h(n) \left[ \sum_{i=-\infty}^{\infty} h_{opt}(i) \psi_{xx}(n-i) - \psi_{xd}(n) \right] = 0
 \tag{A.9}$$

The term within the brackets is the discrete form of the Weiner-Hopf equation and must be equal to zero if equation A.9 is to be valid since  $\Delta h(n) > 0$  by definition. Therefore,

$$\sum_{i=-\infty}^{\infty} h_{opt}(i) \psi_{xx}(n-i) = \psi_{xd}(n)
 \tag{A.10}$$

Equation A.10 is the most often encountered in truncated form in linear prediction theory, Pade approximation, adaptive filtering and lattice filtering. The truncated version of A.10 which is generally used in solving the discrete Wiener problem is,

$$\sum_{i=0}^M h_{\text{opt}}(i) \psi_{\text{xx}}(n-i) = \psi_{\text{xd}}(n) \quad (\text{A.11})$$

or in matrix form,

$$\begin{bmatrix} \psi_{\text{xx}}(0) & \psi_{\text{xx}}(-1) & \dots & \psi_{\text{xx}}(-n) \\ \psi_{\text{xx}}(1) & \psi_{\text{xx}}(0) & \dots & \psi_{\text{xx}}(1-n) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{\text{xx}}(n) & \psi_{\text{xx}}(n-1) & \dots & \psi_{\text{xx}}(0) \end{bmatrix} \begin{bmatrix} h_{\text{opt}}(0) \\ h_{\text{opt}}(1) \\ \vdots \\ h_{\text{opt}}(n) \end{bmatrix} = \begin{bmatrix} \psi_{\text{xd}}(0) \\ \psi_{\text{xd}}(1) \\ \vdots \\ \psi_{\text{xd}}(n) \end{bmatrix} \quad (\text{A.12})$$

APPENDIX B

```

PROGRAM ADMA: A TRACKLESSIVE-MOVING AVERAGE SIMULATOR
      THIS PROGRAM GENERATES THE VARIOUS TYPES OF LINEAR SYSTEMS
      DATA WHICH IS PROCESSED BY THE ADAPTIVE KALMAN IDENTIFIER (AKI).
      THE PROGRAM PERFORMS ALL COMPUTATIONS IN DOUBLE PRECISION AND
      SIZES THE OUTPUT DATA ON A DISK FILE FOR ANALYSIS BY THE AKI. IF
      LINKERS OF THE AUTOREGRESSIVE (AR) AND MOVING-AVERAGE (MA) PROCESSES
      PRECEDE THE DATA IN UNCOMPRESSED FORM.
      REAL*4 Y(6000),U(8000),Z(6000),V(8000)
      REAL*4 W(8000),W2(8000)
      WRITE (6,*)N
      DO UNTIL (I=1,IX,MI,ME,DX,SH,IPUT,PK,6MULTIPLY,4X,LINEADJUSTMENT,/)
      CALL=25545117.00
      SIGW = 1.00E-3
      SIGU = 1.00E-3
      NPE = NUMBER OF DATA POINTS GENERATED
      ME = NUMBER OF AR COEFFICIENTS
      MA = NUMBER OF MA COEFFICIENTS
      ME = MODE OF OPERATION FOR THE AKI
      MPE = TOTAL ORDER OF SYSTEM
      NPE=400
      ME=0
      MA=0
      MPE=400
      WRITE (6,*)N,ME,MA
      CALL SAMPLE (USE1,ME,MA)
      CALL SAMPLE (USE1,ME,MA)
      DO 10 I=1,NPE
      V(I) = SIGV*DBLE(W2(I))
      U(I) = SIGU*DBLE(W(I))
      10
      A1=0.0000
      A2=0.0000
      A3=0.0000
      A4=0.0000
      A5=1.0000
      A6=0.0000
      A7=0.0000
      A8=0.0000
      AF000050
      AF000055
      AF000060
      AF000065
      AF000070
      AF000075
      AF000080
      AF000085
      AF000090
      AF000095
      AF000100
      AF000105
      AF000110
      AF000115
      AF000120
      AF000125
      AF000130
      AF000135
      AF000140
      AF000145
      AF000150
      AF000155
      AF000160
      AF000165
      AF000170
      AF000175
      AF000180
      AF000185
      AF000190
      AF000195
      AF000200
      AF000205
      AF000210
      AF000215
      AF000220
      AF000225
      AF000230
      AF000235
      AF000240
      AF000245
      AF000250
      AF000255
      AF000260
      AF000265
      AF000270
      AF000275
      AF000280
      AF000285
      AF000290
      AF000295
      AF000300
      AF000305
      AF000310
      AF000315
      AF000320
      AF000325
      AF000330
      AF000335
      AF000340
      AF000345
      AF000350
      AF000355
      AF000360
      AF000365
      AF000370
      AF000375
      AF000380
      AF000385
      AF000390
      AF000395
      AF000400
      AF000405
      AF000410
      AF000415
      AF000420
      AF000425
      AF000430
      AF000435
      AF000440
      AF000445
      AF000450
      AF000455
      AF000460
      AF000465
      AF000470
      AF000475
      AF000480
      AF000485
      AF000490
      AF000495
      AF000500
      AF000505
      AF000510
  
```

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC









```

111 01470
112 01471
113 01472
114 01473
115 01474
116 01475
117 01476
118 01477
119 01478
120 01479
121 01480
122 01481
123 01482
124 01483
125 01484
126 01485
127 01486
128 01487
129 01488
130 01489
131 01490
132 01491
133 01492
134 01493
135 01494
136 01495
137 01496
138 01497
139 01498
140 01499
141 01500
142 01501
143 01502
144 01503
145 01504
146 01505
147 01506
148 01507
149 01508
150 01509
151 01510
152 01511
153 01512
154 01513
155 01514
156 01515
157 01516
158 01517
159 01518
160 01519
161 01520
162 01521
163 01522
164 01523
165 01524
166 01525
167 01526
168 01527
169 01528
170 01529
171 01530
172 01531
173 01532
174 01533
175 01534
176 01535
177 01536
178 01537
179 01538
180 01539
181 01540
182 01541
183 01542
184 01543
185 01544
186 01545
187 01546
188 01547
189 01548
190 01549
191 01550
192 01551
193 01552
194 01553
195 01554
196 01555
197 01556
198 01557
199 01558
200 01559
201 01560
202 01561
203 01562
204 01563
205 01564
206 01565
207 01566
208 01567
209 01568
210 01569
211 01570
212 01571
213 01572
214 01573
215 01574
216 01575
217 01576
218 01577
219 01578
220 01579
221 01580
222 01581
223 01582
224 01583
225 01584
226 01585
227 01586
228 01587
229 01588
230 01589
231 01590
232 01591
233 01592
234 01593
235 01594
236 01595
237 01596
238 01597
239 01598
240 01599
241 01600
242 01601
243 01602
244 01603
245 01604
246 01605
247 01606
248 01607
249 01608
250 01609
251 01610
252 01611
253 01612
254 01613
255 01614
256 01615
257 01616
258 01617
259 01618
260 01619
261 01620
262 01621
263 01622
264 01623
265 01624
266 01625
267 01626
268 01627
269 01628
270 01629
271 01630
272 01631
273 01632
274 01633
275 01634
276 01635
277 01636
278 01637
279 01638
280 01639
281 01640
282 01641
283 01642
284 01643
285 01644
286 01645
287 01646
288 01647
289 01648
290 01649
291 01650
292 01651
293 01652
294 01653
295 01654
296 01655
297 01656
298 01657
299 01658
300 01659
301 01660
302 01661
303 01662
304 01663
305 01664
306 01665
307 01666
308 01667
309 01668
310 01669
311 01670
312 01671
313 01672
314 01673
315 01674
316 01675
317 01676
318 01677
319 01678
320 01679
321 01680
322 01681
323 01682
324 01683
325 01684
326 01685
327 01686
328 01687
329 01688
330 01689
331 01690
332 01691
333 01692
334 01693
335 01694
336 01695
337 01696
338 01697
339 01698
340 01699
341 01700
342 01701
343 01702
344 01703
345 01704
346 01705
347 01706
348 01707
349 01708
350 01709
351 01710
352 01711
353 01712
354 01713
355 01714
356 01715
357 01716
358 01717
359 01718
360 01719
361 01720
362 01721
363 01722
364 01723
365 01724
366 01725
367 01726
368 01727
369 01728
370 01729
371 01730
372 01731
373 01732
374 01733
375 01734
376 01735
377 01736
378 01737
379 01738
380 01739
381 01740
382 01741
383 01742
384 01743
385 01744
386 01745
387 01746
388 01747
389 01748
390 01749
391 01750
392 01751
393 01752
394 01753
395 01754
396 01755
397 01756
398 01757
399 01758
400 01759
401 01760
402 01761
403 01762
404 01763
405 01764
406 01765
407 01766
408 01767
409 01768
410 01769
411 01770
412 01771
413 01772
414 01773
415 01774
416 01775
417 01776
418 01777
419 01778
420 01779
421 01780
422 01781
423 01782
424 01783
425 01784
426 01785
427 01786
428 01787
429 01788
430 01789
431 01790
432 01791
433 01792
434 01793
435 01794
436 01795
437 01796
438 01797
439 01798
440 01799
441 01800
442 01801
443 01802
444 01803
445 01804
446 01805
447 01806
448 01807
449 01808
450 01809
451 01810
452 01811
453 01812
454 01813
455 01814
456 01815
457 01816
458 01817
459 01818
460 01819
461 01820
462 01821
463 01822
464 01823
465 01824
466 01825
467 01826
468 01827
469 01828
470 01829
471 01830
472 01831
473 01832
474 01833
475 01834
476 01835
477 01836
478 01837
479 01838
480 01839
481 01840
482 01841
483 01842
484 01843
485 01844
486 01845
487 01846
488 01847
489 01848
490 01849
491 01850
492 01851
493 01852
494 01853
495 01854
496 01855
497 01856
498 01857
499 01858
500 01859
501 01860
502 01861
503 01862
504 01863
505 01864
506 01865
507 01866
508 01867
509 01868
510 01869
511 01870
512 01871
513 01872
514 01873
515 01874
516 01875
517 01876
518 01877
519 01878
520 01879
521 01880
522 01881
523 01882
524 01883
525 01884
526 01885
527 01886
528 01887
529 01888
530 01889
531 01890
532 01891
533 01892
534 01893
535 01894
536 01895
537 01896
538 01897
539 01898
540 01899
541 01900
542 01901
543 01902
544 01903
545 01904
546 01905
547 01906
548 01907
549 01908
550 01909
551 01910
552 01911
553 01912
554 01913
555 01914
556 01915
557 01916
558 01917
559 01918
560 01919
561 01920
562 01921
563 01922
564 01923
565 01924
566 01925
567 01926
568 01927
569 01928
570 01929
571 01930
572 01931
573 01932
574 01933
575 01934
576 01935
577 01936
578 01937
579 01938
580 01939
581 01940
582 01941
583 01942
584 01943
585 01944
586 01945
587 01946
588 01947
589 01948
590 01949
591 01950
592 01951
593 01952
594 01953
595 01954
596 01955
597 01956
598 01957
599 01958
600 01959
601 01960
602 01961
603 01962
604 01963
605 01964
606 01965
607 01966
608 01967
609 01968
610 01969
611 01970
612 01971
613 01972
614 01973
615 01974
616 01975
617 01976
618 01977
619 01978
620 01979
621 01980
622 01981
623 01982
624 01983
625 01984
626 01985
627 01986
628 01987
629 01988
630 01989
631 01990
632 01991
633 01992
634 01993
635 01994
636 01995
637 01996
638 01997
639 01998
640 01999
641 02000
642 02001
643 02002
644 02003
645 02004
646 02005
647 02006
648 02007
649 02008
650 02009
651 02010
652 02011
653 02012
654 02013
655 02014
656 02015
657 02016
658 02017
659 02018
660 02019
661 02020
662 02021
663 02022
664 02023
665 02024
666 02025
667 02026
668 02027
669 02028
670 02029
671 02030
672 02031
673 02032
674 02033
675 02034
676 02035
677 02036
678 02037
679 02038
680 02039
681 02040
682 02041
683 02042
684 02043
685 02044
686 02045
687 02046
688 02047
689 02048
690 02049
691 02050
692 02051
693 02052
694 02053
695 02054
696 02055
697 02056
698 02057
699 02058
700 02059
701 02060
702 02061
703 02062
704 02063
705 02064
706 02065
707 02066
708 02067
709 02068
710 02069
711 02070
712 02071
713 02072
714 02073
715 02074
716 02075
717 02076
718 02077
719 02078
720 02079
721 02080
722 02081
723 02082
724 02083
725 02084
726 02085
727 02086
728 02087
729 02088
730 02089
731 02090
732 02091
733 02092
734 02093
735 02094
736 02095
737 02096
738 02097
739 02098
740 02099
741 02100
742 02101
743 02102
744 02103
745 02104
746 02105
747 02106
748 02107
749 02108
750 02109
751 02110
752 02111
753 02112
754 02113
755 02114
756 02115
757 02116
758 02117
759 02118
760 02119
761 02120
762 02121
763 02122
764 02123
765 02124
766 02125
767 02126
768 02127
769 02128
770 02129
771 02130
772 02131
773 02132
774 02133
775 02134
776 02135
777 02136
778 02137
779 02138
780 02139
781 02140
782 02141
783 02142
784 02143
785 02144
786 02145
787 02146
788 02147
789 02148
790 02149
791 02150
792 02151
793 02152
794 02153
795 02154
796 02155
797 02156
798 02157
799 02158
800 02159
801 02160
802 02161
803 02162
804 02163
805 02164
806 02165
807 02166
808 02167
809 02168
810 02169
811 02170
812 02171
813 02172
814 02173
815 02174
816 02175
817 02176
818 02177
819 02178
820 02179
821 02180
822 02181
823 02182
824 02183
825 02184
826 02185
827 02186
828 02187
829 02188
830 02189
831 02190
832 02191
833 02192
834 02193
835 02194
836 02195
837 02196
838 02197
839 02198
840 02199
841 02200
842 02201
843 02202
844 02203
845 02204
846 02205
847 02206
848 02207
849 02208
850 02209
851 02210
852 02211
853 02212
854 02213
855 02214
856 02215
857 02216
858 02217
859 02218
860 02219
861 02220
862 02221
863 02222
864 02223
865 02224
866 02225
867 02226
868 02227
869 02228
870 02229
871 02230
872 02231
873 02232
874 02233
875 02234
876 02235
877 02236
878 02237
879 02238
880 02239
881 02240
882 02241
883 02242
884 02243
885 02244
886 02245
887 02246
888 02247
889 02248
890 02249
891 02250
892 02251
893 02252
894 02253
895 02254
896 02255
897 02256
898 02257
899 02258
900 02259
901 02260
902 02261
903 02262
904 02263
905 02264
906 02265
907 02266
908 02267
909 02268
910 02269
911 02270
912 02271
913 02272
914 02273
915 02274
916 02275
917 02276
918 02277
919 02278
920 02279
921 02280
922 02281
923 02282
924 02283
925 02284
926 02285
927 02286
928 02287
929 02288
930 02289
931 02290
932 02291
933 02292
934 02293
935 02294
936 02295
937 02296
938 02297
939 02298
940 02299
941 02300
942 02301
943 02302
944 02303
945 02304
946 02305
947 02306
948 02307
949 02308
950 02309
951 02310
952 02311
953 02312
954 02313
955 02314
956 02315
957 02316
958 02317
959 02318
960 02319
961 02320
962 02321
963 02322
964 02323
965 02324
966 02325
967 02326
968 02327
969 02328
970 02329
971 02330
972 02331
973 02332
974 02333
975 02334
976 02335
977 02336
978 02337
979 02338
980 02339
981 02340
982 02341
983 02342
984 02343
985 02344
986 02345
987 02346
988 02347
989 02348
990 02349
991 02350
992 02351
993 02352
994 02353
995 02354
996 02355
997 02356
998 02357
999 02358
1000 02359
1001 02360
1002 02361
1003 02362
1004 02363
1005 02364
1006 02365
1007 02366
1008 02367
1009 02368
1010 02369
1011 02370
1012 02371
1013 02372
1014 02373
1015 02374
1016 02375
1017 02376
1018 02377
1019 02378
1020 02379
1021 02380
1022 02381
1023 02382
1024 02383
1025 02384
1026 02385
1027 02386
1028 02387
1029 02388
1030 02389
1031 02390
1032 02391
1033 02392
1034 02393
1035 02394
1036 02395
1037 02396
1038 02397
1039 02398
1040 02399
1041 02400
1042 02401
1043 02402
1044 02403
1045 02404
1046 02405
1047 02406
1048 02407
1049 02408
1050 02409
1051 02410
1052 02411
1053 02412
1054 02413
1055 02414
1056 02415
1057 02416
1058 02417
1059 02418
1060 02419
1061 02420
1062 02421
1063 02422
1064 02423
1065 02424
1066 02425
1067 02426
1068 02427
1069 02428
1070 02429
1071 02430
1072 02431
1073 02432
1074 02433
1075 02434
1076 02435
1077 02436
1078 02437
1079 02438
1080 02439
1081 02440
1082 02441
1083 02442
1084 02443
1085 02444
1086 02445
1087 02446
1088 02447
1089 02448
1090 02449
1091 02450
1092 02451
1093 02452
1094 02453
1095 02454
1096 02455
1097 02456
1098 02457
1099 02458
1100 02459
1101 02460
1102 02461
1103 02462
1104 02463
1105 02464
1106 02465
1107 02466
1108 02467
1109 02468
1110 02469
1111 02470
1112 02471
1113 02472
1114 02473
1115 02474
1116 02475
1117 02476
1118 02477
1119 02478
1120 02479
1121 02480
1122 02481
1123 02482
1124 02483
1125 02484
1126 02485
1127 02486
1128 02487
1129 02488
1130 02489
1131 02490
1132 02491
1133 02492
1134 02493
1135 02494
1136 02495
1137 02496
1138 02497
1139 02498
1140 02499
1141 02500
1142 02501
1143 02502
1144 02503
1145 02504
1146 02505
1147 02506
1148 02507
1149 02508
1150 02509
1151 02510
1152 02511
1153 02512
1154 02513
1155 02514
1156 02515
1157 02516
1158 02517
1159 02518
1160 02519
1161 02520
1162 02521
1163 02522
1164 02523
1165 02524
1166 02525
1167 02526
1168 02527
1169 02528
1170 02529
1171 02530
1172 02531
1173 02532
1174 02533
1175 02534
1176 02535
1177 02536
1178 02537
1179 02538
1180 02539
1181 02540
1182 02541
1183 02542
1184 02543
1185 02544
1186 02545
1187 02546
1188 02547
1189 02548
1190 02549
1191 02550
1192 02551
1193 02552
1194 02553
1195 02554
1196 02555
1197 02556
1198 02557
1199 02558
1200 02559
1201 02560
1202 02561
1203 02562
1204 02563
1205 02564
1206 02565
1207 02566
1208 02567
1209 02568
1210 02569
1211 02570
1212 02571
1213 02572
1214 02573
1215 02574
1216 02575
1217 02576
1218 02577
1219 02578
1220 02579
1221 02580
1222 02581
1223 02582
1224 02583
1225 02584
1226 02585
1227 02586
1228 02587
1229 02588
1230 02589
1231 02590
1232 02591
1233 02592
1234 02593
1235 02594
1236 02595
1237 02596
1238 02597
1239 02598
1240 02599
1241 02600
1242 02601
1243 02602
1244 02603
1245 02604
1246 02605
1247 02606
1248 02607
1249 02608
1250 02609
1251 02610
1252 02611
1253 02612
1254 02613
1255 02614
1256 02615
1257 02616
1258 02617
1259 02618
1260 02619
1261 02620
1262 02621
1263 02622
1264 02623
1265 02624
1266 02625
1267 02626
1268 02627
1269 02628
1270 02629
1271 02630
1272 02631
1273 02632
1274 02633
1275 02634
1276 02635
1277 02636
1278 02637
1279 02638
1280 02639
1281 02640
1282 02641
1283 02642
1284 02643
1285 02644
1286 02645
1287 02646
1288 02647
1289 02648
1290 02649
1291 02650
1292 02651
1293 02652
1294 02653
1295 02654
1296 02655
1297 02656
1298 02657
1299 02658
1300 02659
1301 02660
1302 02661
1303 02662
1304 02663
1305 02664
1306 02665
1307 02666
1308 02667
1309 02668
1310 02669
1311 02670
1312 02671
1313 02672
1314 0
```



APPENDIX D

```

PROGRAM ALAP10: ADAPTIVE KALMAN IDENTIFIER (AKI)
      THIS PROGRAM REPRESENTS THE MAIN SOFTWARE DEVELOPED WHICH IM-
      PLEMENTS THE ADAPTIVE KALMAN IDENTIFIER (AKI) FORMULATED IN THE
      BODY OF THE THESIS. THE ADAPTIVE KALMAN IDENTIFIER (AKI) IDENTIFIES
      THE COEFFICIENTS OF THE GENERAL ALGEBRAIC SYSTEM (AKI) IDENTIFIES
      (AKI) REPRESENTATION OF AN UNKNOWN SYSTEM. HERETOFORE, IDENTIFYING
      SYSTEM PARAMETERS WERE SIMPLY AUGMENTED TO THE STATE VECTOR OF THE
      GENERAL KALMAN FILTERS AND, HENCE, ESTIMATED. USING THE AKI PRO-
      CEEDURE, SUCH AUGMENTATION OR EXTENSION IS MADE; BUT, RATHER,
      THE PARAMETERS OF THE SYSTEM (OR EQUIVALENTLY THE COEFFICIENTS OF
      THE ALGEBRAIC) ARE IDENTIFIED DIRECTLY.
      IDENTIFYING THE AKI COEFFICIENTS REQUIRES PROCESSING THE
      INPUT/OUTPUT DATA WHICH IS TEMPORARILY STORED ON A DISK FILE PRIOR
      TO BEING READ BY THE AKI PROGRAM. THE COLUMNS OF THE AUTOREGRESSIVE
      (AKI) AND MOVING-AVERAGE (MA) PROCESS AND THE CPU TIME FLAG PRECEDE
      THE UNFORMATTED DATA ALSO IN UNORDERED FORM. THE AKI PROGRAM
      HAS SEVERAL MULTIPURPOSE FLAGS WHICH ARE DESCRIBED THROUGHOUT THE
      SOURCE CODE IN THE FORM OF COMMENT STATEMENTS.
      REAL*8 A(20,30),H,PKK,PKK1,D,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,Y11,
      COMMON A(20,30),E(20,30),I(20,20),PKK(20,20),PKKAL(20,20)
      LOGICAL YY(800),YY(800),YL(800),YL(800),YC(20,800)
      DIMENSION Y(60),G(20,20),E(20,20),Z,R,N,M,N101,KK,I01AG,NI
      DATA 11
      REAL*8 S100
      DIMENSION V1(30),V2(30),A(20),B(20)
      COMPLEX*16 C1(20),C2(20)
      INTEGER N101,N102,N103,N104,N105,N106,N107,N108,N109,N110,N111,N112,N113,N114,N115,N116,N117,N118,N119,N120,N121,N122,N123,N124,N125,N126,N127,N128,N129,N130,N131,N132,N133,N134,N135,N136,N137,N138,N139,N140,N141,N142,N143,N144,N145,N146,N147,N148,N149,N150,N151,N152,N153,N154,N155,N156,N157,N158,N159,N160,N161,N162,N163,N164,N165,N166,N167,N168,N169,N170,N171,N172,N173,N174,N175,N176,N177,N178,N179,N180,N181,N182,N183,N184,N185,N186,N187,N188,N189,N190,N191,N192,N193,N194,N195,N196,N197,N198,N199,N200,N201,N202,N203,N204,N205,N206,N207,N208,N209,N210,N211,N212,N213,N214,N215,N216,N217,N218,N219,N220,N221,N222,N223,N224,N225,N226,N227,N228,N229,N230,N231,N232,N233,N234,N235,N236,N237,N238,N239,N240,N241,N242,N243,N244,N245,N246,N247,N248,N249,N250,N251,N252,N253,N254,N255,N256,N257,N258,N259,N260,N261,N262,N263,N264,N265,N266,N267,N268,N269,N270,N271,N272,N273,N274,N275,N276,N277,N278,N279,N280,N281,N282,N283,N284,N285,N286,N287,N288,N289,N290,N291,N292,N293,N294,N295,N296,N297,N298,N299,N300,N301,N302,N303,N304,N305,N306,N307,N308,N309,N310,N311,N312,N313,N314,N315,N316,N317,N318,N319,N320,N321,N322,N323,N324,N325,N326,N327,N328,N329,N330,N331,N332,N333,N334,N335,N336,N337,N338,N339,N340,N341,N342,N343,N344,N345,N346,N347,N348,N349,N350,N351,N352,N353,N354,N355,N356,N357,N358,N359,N360,N361,N362,N363,N364,N365,N366,N367,N368,N369,N370,N371,N372,N373,N374,N375,N376,N377,N378,N379,N380,N381,N382,N383,N384,N385,N386,N387,N388,N389,N390,N391,N392,N393,N394,N395,N396,N397,N398,N399,N400,N401,N402,N403,N404,N405,N406,N407,N408,N409,N410,N411,N412,N413,N414,N415,N416,N417,N418,N419,N420,N421,N422,N423,N424,N425,N426,N427,N428,N429,N430,N431,N432,N433,N434,N435,N436,N437,N438,N439,N440,N441,N442,N443,N444,N445,N446,N447,N448,N449,N450,N451,N452,N453,N454,N455,N456,N457,N458,N459,N460,N461,N462,N463,N464,N465,N466,N467,N468,N469,N470,N471,N472,N473,N474,N475,N476,N477,N478,N479,N480,N481,N482,N483,N484,N485,N486,N487,N488,N489,N490,N491,N492,N493,N494,N495,N496,N497,N498,N499,N500,N501,N502,N503,N504,N505,N506,N507,N508,N509,N510,N511,N512,N513,N514,N515,N516,N517,N518,N519,N520,N521,N522,N523,N524,N525,N526,N527,N528,N529,N530,N531,N532,N533,N534,N535,N536,N537,N538,N539,N540,N541,N542,N543,N544,N545,N546,N547,N548,N549,N550,N551,N552,N553,N554,N555,N556,N557,N558,N559,N560,N561,N562,N563,N564,N565,N566,N567,N568,N569,N570,N571,N572,N573,N574,N575,N576,N577,N578,N579,N580,N581,N582,N583,N584,N585,N586,N587,N588,N589,N590,N591,N592,N593,N594,N595,N596,N597,N598,N599,N600,N601,N602,N603,N604,N605,N606,N607,N608,N609,N610,N611,N612,N613,N614,N615,N616,N617,N618,N619,N620,N621,N622,N623,N624,N625,N626,N627,N628,N629,N630,N631,N632,N633,N634,N635,N636,N637,N638,N639,N640,N641,N642,N643,N644,N645,N646,N647,N648,N649,N650,N651,N652,N653,N654,N655,N656,N657,N658,N659,N660,N661,N662,N663,N664,N665,N666,N667,N668,N669,N670,N671,N672,N673,N674,N675,N676,N677,N678,N679,N680,N681,N682,N683,N684,N685,N686,N687,N688,N689,N690,N691,N692,N693,N694,N695,N696,N697,N698,N699,N700,N701,N702,N703,N704,N705,N706,N707,N708,N709,N710,N711,N712,N713,N714,N715,N716,N717,N718,N719,N720,N721,N722,N723,N724,N725,N726,N727,N728,N729,N730,N731,N732,N733,N734,N735,N736,N737,N738,N739,N740,N741,N742,N743,N744,N745,N746,N747,N748,N749,N750,N751,N752,N753,N754,N755,N756,N757,N758,N759,N760,N761,N762,N763,N764,N765,N766,N767,N768,N769,N770,N771,N772,N773,N774,N775,N776,N777,N778,N779,N780,N781,N782,N783,N784,N785,N786,N787,N788,N789,N790,N791,N792,N793,N794,N795,N796,N797,N798,N799,N800,N801,N802,N803,N804,N805,N806,N807,N808,N809,N810,N811,N812,N813,N814,N815,N816,N817,N818,N819,N820,N821,N822,N823,N824,N825,N826,N827,N828,N829,N830,N831,N832,N833,N834,N835,N836,N837,N838,N839,N840,N841,N842,N843,N844,N845,N846,N847,N848,N849,N850,N851,N852,N853,N854,N855,N856,N857,N858,N859,N860,N861,N862,N863,N864,N865,N866,N867,N868,N869,N870,N871,N872,N873,N874,N875,N876,N877,N878,N879,N880,N881,N882,N883,N884,N885,N886,N887,N888,N889,N890,N891,N892,N893,N894,N895,N896,N897,N898,N899,N900,N901,N902,N903,N904,N905,N906,N907,N908,N909,N910,N911,N912,N913,N914,N915,N916,N917,N918,N919,N920,N921,N922,N923,N924,N925,N926,N927,N928,N929,N930,N931,N932,N933,N934,N935,N936,N937,N938,N939,N940,N941,N942,N943,N944,N945,N946,N947,N948,N949,N950,N951,N952,N953,N954,N955,N956,N957,N958,N959,N960,N961,N962,N963,N964,N965,N966,N967,N968,N969,N970,N971,N972,N973,N974,N975,N976,N977,N978,N979,N980,N981,N982,N983,N984,N985,N986,N987,N988,N989,N990,N991,N992,N993,N994,N995,N996,N997,N998,N999,1000
      PRINT*P=100
      IPE = A PRINTING FLAG; IF IPE=1 SUBROUTINE CAPI IS CALLED
      I01AG = A DIAGNOSTIC FLAG; IF I01AG=1 MATRIX INVERSE DATE
      VALUES ARE PRINTED OUT; I01AG = A RESULT CALCULATION FLAG;
      I01AG = 1 IDENTIFIED VALUES FOR THE RESULTS ARE COMPUTED.
      I01AG = A PRINTING FLAG; IDENTIFIED OUTPUT RESULTS FOR K201.
      IPE=1
      I01AG=0
      I01AG=1
  
```









```

100  DO 21 AT=1,NMUL
110  CALL C(1,2) (Q(1,0),J=1,NMUL)
120  CALL C(1,2) (Q(1,0),J=1,NMUL)
130  BEGIN FORN ITERATION LOOP
140  NNEW=
150  NOLD=N
160  DO 170 K=1,NMUL,1
170  NNEW=
180  COMPUTE THE CALLS, C(K), AND ERROR COVARIANCES, PNN(K) & PNNR(K)
190  CALL C(K)
200  COMPUTE ESTIMATE OF THE ARMA COEFFICIENTS
210  CALL RCON
220  ADAPTIVE KALMAN ARRAY COEFFICIENTS ARE NOW COMPUTE FOR
230  THE STEP BY 600 AND READY FOR PRINTING
240  IF (ABS(LEND) < 10.10)
250  CALL PRINT
260  IF (LEND < 0) GO TO 100
270  COMPUTE THE PULS AND ZEPS
280  RLEND=
290  IF (ABS(LEND) < 50.10)
300  RLEND=
310  GO TO 120
320  CALL C(K) = SQR(ABS(RLEND))
330  CALL ZEPSY (V1,PL,IER)
340  WRITE (1,13)
350  CALL C(1,2) (Q(1,0),J=1,NMUL)
360  CALL C(1,2) (Q(1,0),J=1,NMUL)
370  CALL C(1,2) (Q(1,0),J=1,NMUL)
380  CALL C(1,2) (Q(1,0),J=1,NMUL)
390  CALL C(1,2) (Q(1,0),J=1,NMUL)
400  CALL C(1,2) (Q(1,0),J=1,NMUL)
410  CALL C(1,2) (Q(1,0),J=1,NMUL)
420  CALL C(1,2) (Q(1,0),J=1,NMUL)
430  CALL C(1,2) (Q(1,0),J=1,NMUL)
440  CALL C(1,2) (Q(1,0),J=1,NMUL)
450  CALL C(1,2) (Q(1,0),J=1,NMUL)
460  CALL C(1,2) (Q(1,0),J=1,NMUL)
470  CALL C(1,2) (Q(1,0),J=1,NMUL)
480  CALL C(1,2) (Q(1,0),J=1,NMUL)
490  CALL C(1,2) (Q(1,0),J=1,NMUL)
500  CALL C(1,2) (Q(1,0),J=1,NMUL)
510  CALL C(1,2) (Q(1,0),J=1,NMUL)
520  CALL C(1,2) (Q(1,0),J=1,NMUL)
530  CALL C(1,2) (Q(1,0),J=1,NMUL)
540  CALL C(1,2) (Q(1,0),J=1,NMUL)
550  CALL C(1,2) (Q(1,0),J=1,NMUL)
560  CALL C(1,2) (Q(1,0),J=1,NMUL)
570  CALL C(1,2) (Q(1,0),J=1,NMUL)
580  CALL C(1,2) (Q(1,0),J=1,NMUL)
590  CALL C(1,2) (Q(1,0),J=1,NMUL)
600  CALL C(1,2) (Q(1,0),J=1,NMUL)
610  CALL C(1,2) (Q(1,0),J=1,NMUL)
620  CALL C(1,2) (Q(1,0),J=1,NMUL)
630  CALL C(1,2) (Q(1,0),J=1,NMUL)
640  CALL C(1,2) (Q(1,0),J=1,NMUL)
650  CALL C(1,2) (Q(1,0),J=1,NMUL)
660  CALL C(1,2) (Q(1,0),J=1,NMUL)
670  CALL C(1,2) (Q(1,0),J=1,NMUL)
680  CALL C(1,2) (Q(1,0),J=1,NMUL)
690  CALL C(1,2) (Q(1,0),J=1,NMUL)
700  CALL C(1,2) (Q(1,0),J=1,NMUL)
710  CALL C(1,2) (Q(1,0),J=1,NMUL)
720  CALL C(1,2) (Q(1,0),J=1,NMUL)
730  CALL C(1,2) (Q(1,0),J=1,NMUL)
740  CALL C(1,2) (Q(1,0),J=1,NMUL)
750  CALL C(1,2) (Q(1,0),J=1,NMUL)
760  CALL C(1,2) (Q(1,0),J=1,NMUL)
770  CALL C(1,2) (Q(1,0),J=1,NMUL)
780  CALL C(1,2) (Q(1,0),J=1,NMUL)
790  CALL C(1,2) (Q(1,0),J=1,NMUL)
800  CALL C(1,2) (Q(1,0),J=1,NMUL)
810  CALL C(1,2) (Q(1,0),J=1,NMUL)
820  CALL C(1,2) (Q(1,0),J=1,NMUL)
830  CALL C(1,2) (Q(1,0),J=1,NMUL)
840  CALL C(1,2) (Q(1,0),J=1,NMUL)
850  CALL C(1,2) (Q(1,0),J=1,NMUL)
860  CALL C(1,2) (Q(1,0),J=1,NMUL)
870  CALL C(1,2) (Q(1,0),J=1,NMUL)
880  CALL C(1,2) (Q(1,0),J=1,NMUL)
890  CALL C(1,2) (Q(1,0),J=1,NMUL)
900  CALL C(1,2) (Q(1,0),J=1,NMUL)
910  CALL C(1,2) (Q(1,0),J=1,NMUL)
920  CALL C(1,2) (Q(1,0),J=1,NMUL)
930  CALL C(1,2) (Q(1,0),J=1,NMUL)
940  CALL C(1,2) (Q(1,0),J=1,NMUL)
950  CALL C(1,2) (Q(1,0),J=1,NMUL)
960  CALL C(1,2) (Q(1,0),J=1,NMUL)
970  CALL C(1,2) (Q(1,0),J=1,NMUL)
980  CALL C(1,2) (Q(1,0),J=1,NMUL)
990  CALL C(1,2) (Q(1,0),J=1,NMUL)

```

A1AC0020  
 A1AC0021  
 A1AC0022  
 A1AC0023  
 A1AC0024  
 A1AC0025  
 A1AC0026  
 A1AC0027  
 A1AC0028  
 A1AC0029  
 A1AC0030  
 A1AC0031  
 A1AC0032  
 A1AC0033  
 A1AC0034  
 A1AC0035  
 A1AC0036

```

150 CALL (0,150)
160 PRINT (7,58), "LINE FILES: /,000,2 HK, 0,000,0 /"
170 CALL (0,150) (0,2,1),1=1,000)
180 PRINT (0,100) 113
190 PRINT (7,58), "INTERIOR CODE: 1116, /"
200 CONTINUE
210 PRINT (7,58), "SUBSEQUENT VECTORS, FAK), TO INCLUDE THE ROW DATA
220 CALL (0,150)
230 CALL (0,150)
240 CALL (0,150)
250 STOP
260 END
  
```

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC

















ALAC(19)  
 ALAC(20)  
 ALAC(21)  
 ALAC(22)  
 ALAC(23)  
 ALAC(24)  
 ALAC(25)  
 ALAC(26)  
 ALAC(27)  
 ALAC(28)  
 ALAC(29)  
 ALAC(30)  
 ALAC(31)  
 ALAC(32)  
 ALAC(33)  
 ALAC(34)  
 ALAC(35)  
 ALAC(36)  
 ALAC(37)  
 ALAC(38)  
 ALAC(39)  
 ALAC(40)  
 ALAC(41)  
 ALAC(42)  
 ALAC(43)  
 ALAC(44)  
 ALAC(45)  
 ALAC(46)  
 ALAC(47)  
 ALAC(48)  
 ALAC(49)  
 ALAC(50)

ALL INFORMATION CONTAINED HEREIN IS UNCLASSIFIED EXCEPT WHERE SHOWN OTHERWISE  
 DATE 08-14-2010 BY 60322 UCBAW/STP  
 (S) (U) (F) (M) (D) (N) (A) (C) (E) (G) (I) (O) (P) (Q) (R) (S) (T) (V) (W) (X) (Y) (Z) (AA) (AB) (AC) (AD) (AE) (AF) (AG) (AH) (AI) (AJ) (AK) (AL) (AM) (AN) (AO) (AP) (AQ) (AR) (AS) (AT) (AU) (AV) (AW) (AX) (AY) (AZ) (BA) (BB) (BC) (BD) (BE) (BF) (BG) (BH) (BI) (BJ) (BK) (BL) (BM) (BN) (BO) (BP) (BQ) (BR) (BS) (BT) (BU) (BV) (BW) (BX) (BY) (BZ) (CA) (CB) (CC) (CD) (CE) (CF) (CG) (CH) (CI) (CJ) (CK) (CL) (CM) (CN) (CO) (CP) (CQ) (CR) (CS) (CT) (CU) (CV) (CW) (CX) (CY) (CZ) (DA) (DB) (DC) (DD) (DE) (DF) (DG) (DH) (DI) (DJ) (DK) (DL) (DM) (DN) (DO) (DP) (DQ) (DR) (DS) (DT) (DU) (DV) (DW) (DX) (DY) (DZ) (EA) (EB) (EC) (ED) (EE) (EF) (EG) (EH) (EI) (EJ) (EK) (EL) (EM) (EN) (EO) (EP) (EQ) (ER) (ES) (ET) (EU) (EV) (EW) (EX) (EY) (EZ) (FA) (FB) (FC) (FD) (FE) (FF) (FG) (FH) (FI) (FJ) (FK) (FL) (FM) (FN) (FO) (FP) (FQ) (FR) (FS) (FT) (FU) (FV) (FW) (FX) (FY) (FZ) (GA) (GB) (GC) (GD) (GE) (GF) (GG) (GH) (GI) (GJ) (GK) (GL) (GM) (GN) (GO) (GP) (GQ) (GR) (GS) (GT) (GU) (GV) (GW) (GX) (GY) (GZ) (HA) (HB) (HC) (HD) (HE) (HF) (HG) (HH) (HI) (HJ) (HK) (HL) (HM) (HN) (HO) (HP) (HQ) (HR) (HS) (HT) (HU) (HV) (HW) (HX) (HY) (HZ) (IA) (IB) (IC) (ID) (IE) (IF) (IG) (IH) (II) (IJ) (IK) (IL) (IM) (IN) (IO) (IP) (IQ) (IR) (IS) (IT) (IU) (IV) (IW) (IX) (IY) (IZ) (JA) (JB) (JC) (JD) (JE) (JF) (JG) (JH) (JI) (JJ) (JK) (JL) (JM) (JN) (JO) (JP) (JQ) (JR) (JS) (JT) (JU) (JV) (JW) (JX) (JY) (JZ) (KA) (KB) (KC) (KD) (KE) (KF) (KG) (KH) (KI) (KJ) (KK) (KL) (KM) (KN) (KO) (KP) (KQ) (KR) (KS) (KT) (KU) (KV) (KW) (KX) (KY) (KZ) (LA) (LB) (LC) (LD) (LE) (LF) (LG) (LH) (LI) (LJ) (LK) (LL) (LM) (LN) (LO) (LP) (LQ) (LR) (LS) (LT) (LU) (LV) (LW) (LX) (LY) (LZ) (MA) (MB) (MC) (MD) (ME) (MF) (MG) (MH) (MI) (MJ) (MK) (ML) (MM) (MN) (MO) (MP) (MQ) (MR) (MS) (MT) (MU) (MV) (MW) (MX) (MY) (MZ) (NA) (NB) (NC) (ND) (NE) (NF) (NG) (NH) (NI) (NJ) (NK) (NL) (NM) (NO) (NP) (NQ) (NR) (NS) (NT) (NU) (NV) (NW) (NX) (NY) (NZ) (OA) (OB) (OC) (OD) (OE) (OF) (OG) (OH) (OI) (OJ) (OK) (OL) (OM) (ON) (OO) (OP) (OQ) (OR) (OS) (OT) (OU) (OV) (OW) (OX) (OY) (OZ) (PA) (PB) (PC) (PD) (PE) (PF) (PG) (PH) (PI) (PJ) (PK) (PL) (PM) (PN) (PO) (PP) (PQ) (PR) (PS) (PT) (PU) (PV) (PW) (PX) (PY) (PZ) (QA) (QB) (QC) (QD) (QE) (QF) (QG) (QH) (QI) (QJ) (QK) (QL) (QM) (QN) (QO) (QP) (QQ) (QR) (QS) (QT) (QU) (QV) (QW) (QX) (QY) (QZ) (RA) (RB) (RC) (RD) (RE) (RF) (RG) (RH) (RI) (RJ) (RK) (RL) (RM) (RN) (RO) (RP) (RQ) (RR) (RS) (RT) (RU) (RV) (RW) (RX) (RY) (RZ) (SA) (SB) (SC) (SD) (SE) (SF) (SG) (SH) (SI) (SJ) (SK) (SL) (SM) (SN) (SO) (SP) (SQ) (SR) (SS) (ST) (SU) (SV) (SW) (SX) (SY) (SZ) (TA) (TB) (TC) (TD) (TE) (TF) (TG) (TH) (TI) (TJ) (TK) (TL) (TM) (TN) (TO) (TP) (TQ) (TR) (TS) (TT) (TU) (TV) (TW) (TX) (TY) (TZ) (UA) (UB) (UC) (UD) (UE) (UF) (UG) (UH) (UI) (UJ) (UK) (UL) (UM) (UN) (UO) (UP) (UQ) (UR) (US) (UT) (UU) (UV) (UW) (UX) (UY) (UZ) (VA) (VB) (VC) (VD) (VE) (VF) (VG) (VH) (VI) (VJ) (VK) (VL) (VM) (VN) (VO) (VP) (VQ) (VR) (VS) (VT) (VU) (VV) (VW) (VX) (VY) (VZ) (WA) (WB) (WC) (WD) (WE) (WF) (WG) (WH) (WI) (WJ) (WK) (WL) (WM) (WN) (WO) (WP) (WQ) (WR) (WS) (WT) (WU) (WV) (WW) (WX) (WY) (WZ) (XA) (XB) (XC) (XD) (XE) (XF) (XG) (XH) (XI) (XJ) (XK) (XL) (XM) (XN) (XO) (XP) (XQ) (XR) (XS) (XT) (XU) (XV) (XW) (XX) (XY) (XZ) (YA) (YB) (YC) (YD) (YE) (YF) (YG) (YH) (YI) (YJ) (YK) (YL) (YM) (YN) (YO) (YP) (YQ) (YR) (YS) (YT) (YU) (YV) (YW) (YX) (YZ) (ZA) (ZB) (ZC) (ZD) (ZE) (ZF) (ZG) (ZH) (ZI) (ZJ) (ZK) (ZL) (ZM) (ZN) (ZO) (ZP) (ZQ) (ZR) (ZS) (ZT) (ZU) (ZV) (ZW) (ZX) (ZY) (ZZ)

THIS PAGE IS BEST QUALITY PRACTICABLE  
 FROM COPY FURNISHED TO DDC



```

100 BEGIN ADAPTIVE ITERATION LOOP
110 EQ=40 K=2,NP
120 D = Y(K-1)
130 IF (L1AG.EQ.0) GO TO 80
140 WRITE (6,500) D
150 FORMAT (7,5X,'DESIRED SIGNAL D(K):',5X,E15.5,/)
160 CONTINUE
170 COMPUTE ERROR
180 IF (L1AL.EQ.0) GO TO 110
190 WRITE (6,120)
200 FORMAT (7,5X,'W(I):',10X,'X(I):',/,)
210 DO 130 I=1,M
220 WRITE (6,140) W(I),X(I)
230 FORMAT (5X,E15.4)
240 CONTINUE
250 CALL SMLL(W,X,4,C)
260 E=D-C
270 WRITE (6,400) K,E,C,(W(I),I=1,M)
280 FORMAT (14,12F1.3)
290 KL=K-1
300 DO 150 I=1,M
310 W(I,KL) = SNGL(A(LI))
320 IF (L1JAL.EQ.0) GO TO 50
330 WRITE (6,500)
340 FORMAT (7,5X,'X(I):',/,)
350 WRITE (6,600) (X(I),I=1,M)
360 FORMAT (5X,E15.4)
370 CONTINUE
380 COMPUTE W(K+1): THE NEXT STEP PREDICTION
390 DO 40 I=1,M
400 W(I) = KSF*X(I)
410 W(I) = W(I)-XX(I)
420 SUBT THE INPUT DATA VECTOR THRU THE DELAY LINE BY ONE
430 L = M+1
440 N = M-1
450 DO 70 I=1,N
460 LL = L-I
470 X(LL) = X(LI-1)
480 X(LL) = U(K)

```

```

LMS00520
LMS00530
LMS00540
LMS00550
LMS00560
LMS00570
LMS00580
LMS00590
LMS00600
LMS00610
LMS00620
LMS00630
LMS00640
LMS00650
LMS00660
LMS00670
LMS00680
LMS00690
LMS00700
LMS00710
LMS00720
LMS00730
LMS00740
LMS00750
LMS00760
LMS00770
LMS00780
LMS00790
LMS00800
LMS00810
LMS00820
LMS00830
LMS00840
LMS00850
LMS00860
LMS00870
LMS00880
LMS00890
LMS00900
LMS00910
LMS00920
LMS00930
LMS00940
LMS00950
LMS00960
LMS00970
LMS00980
LMS00990

```

LMSCL000  
LMSCL010  
LMSCL020  
LMSCL030  
LMSCL040  
LMSCL050  
LMSCL060  
LMSCL070  
LMSCL080  
LMSCL090  
LMSCL100  
LMSCL110  
LMSCL120  
LMSCL130  
LMSCL140

```
      40 CONTINUE  
      NPP=MP-1  
      WRITE (4,160) N,NPP  
      160 FORMAT(2I5)  
      DO 170 I=1,NP  
      170 J=1,NPP  
      170 WRITE (4,180) MM(I,J)  
      180 FORMAT(4E20.5)  
      STOP  
      END
```

```

SUBROUTINE MULT(B,C,M,D)
REAL*8 B,C,D
DIMENSION B(12),C(12)
INTEGER M
      D=0.0D0
      DO 10 I=1,M
      D=D+B(I)*C(I)
10  NEXT I
      END

```

```

LMS01170
LMS01180
LMS01190
LMS01200
LMS01210
LMS01220
LMS01230
LMS01240
LMS01250
LMS01260

```



```

200 FORMAT (/ ,10X,'ONLY THE LAST',I3,' ITERATIONS ARE PRINTED',/)
      WRITE (6,300)
300 FORMAT (9X,'TIME ',10X,5HFERRR,5X,6HOUTPUT,/)
      TITLE NUMBER OF ITERATIONS IS NP
      NP = 4000
      NPRT = NP-IPRT
      IF ((%LT.2).OR.(N.LT.1)) GO TO 205
      DO 10 I=1,M
        A(I) = 0.000
        X(I) = 0.000
10   YA(I) = 0.000
      DO 11 I=1,N
        B(I) = 0.000
11   YI(I) = 0.000
      DO 20 I=1,NP
20   READ (4) V1(I),V2(I)
      NP = NP-10
      DO 30 I=1,NP
        U(I) = LBLE(V1(I+10))
        Y(I) = UNLE(V2(I+10))
30
      BEGIN ADAPTIVE ITERATION LOOP
      DO 40 K=2,NP
        U = Y(N-1)
        K3 = K-1
        IF (DIA0.E0.0) GO TO 80
        WRITE (6,900) K3,D
900 FORMAT (/ ,5X,'DESIRED SIGNAL D(',I3,'):',5X,F15.5,/)
        GO CONTINUE
      COMPUTE ERROR
      IF (C10A0.E0.0) GO TO 110
      WRITE (6,122)
122 FORMAT (/ ,2X,'COMPUTE THE ERROR WITH:',/)
      WRITE (6,120)
120 FORMAT (/ ,10X,'A(I):',10X,'Y(I):',/)
      DO 130 I=1,M
130   WRITE (6,140) A(I),X(I)
140 FORMAT (2A,2F15.4)

```

```

LMS00520
LMS00530
LMS00540
LMS00550
LMS00560
LMS00570
LMS00580
LMS00590
LMS00600
LMS00610
LMS00620
LMS00630
LMS00640
LMS00650
LMS00660
LMS00670
LMS00680
LMS00690
LMS00700
LMS00710
LMS00720
LMS00730
LMS00740
LMS00750
LMS00760
LMS00770
LMS00780
LMS00790
LMS00800
LMS00810
LMS00820
LMS00830
LMS00840
LMS00850
LMS00860
LMS00870
LMS00880
LMS00890
LMS00900
LMS00910
LMS00920
LMS00930
LMS00940
LMS00950
LMS00960
LMS00970
LMS00980
LMS00990

```

LMS01000  
 LMS01010  
 LMS01020  
 LMS01030  
 LMS01040  
 LMS01050  
 LMS01060  
 LMS01070  
 LMS01080  
 LMS01090  
 LMS01100  
 LMS01110  
 LMS01120  
 LMS01130  
 LMS01140  
 LMS01150  
 LMS01160  
 LMS01170  
 LMS01180  
 LMS01190  
 LMS01200  
 LMS01210  
 LMS01220  
 LMS01230  
 LMS01240  
 LMS01250  
 LMS01260  
 LMS01270  
 LMS01280  
 LMS01290  
 LMS01300  
 LMS01310  
 LMS01320  
 LMS01330  
 LMS01340  
 LMS01350  
 LMS01360  
 LMS01370  
 LMS01380  
 LMS01390  
 LMS01400  
 LMS01410  
 LMS01420  
 LMS01430  
 LMS01440  
 LMS01450  
 LMS01460  
 LMS01470

```

    WRITE (C1,C2)
    L21 FORMAT (/,'10X',O(1):',10X',Y(1):',/)
    DO 131 I=1,M
    L21 WRITE (C1,C2) B(I),YI(I)
    L11 CONTINUE
  C
    CALL MMULT (A,X,N,C1)
    CALL MULTS (C,YI,N,C2)
  C
    IF (10*ABS(EQ,0) GO TO 111
    WRITE (C1,C2) C1,C2
    L12 FORMAT (/,'5X',A*X=C1=',E15.4/,5X',B*Y=C2=',E15.4,/)
    L11 CONTINUE
  C
    YI(1)=C1+C2
    E=D-YI(1)
  C
    IF (ABS(MPFI) WRITE (6,400) K8,E,YI(1)
    FORMAT (110,'E15.4)
  C
    DO 150 L=1,M
    AA(L,K8) = SNGL(A(L))
    DO 151 L=2,N
    B(L) = SNGL(B(L))
  C
    IF (10*ABS(EQ,0) GO TO 50
    WRITE (C1,C2) X(1):',/
    FORMAT (/,'5X',X(1):',/
    WRITE (C1,C2) (X(I),I=1,M)
    DO 1
    WRITE (C1,C2) YI(1):',/
    FORMAT (/,'5X',YI(1):',/
    WRITE (C1,C2) (YI(I),I=2,N)
    DO
    CONTINUE
  C
    CC=CC+E*(K+1) AND D(K+1)= THE NEXT STEP PREDICTION
    DO 60 I=1,M
    XX(I) = KA*(X(I)
    AA(I) = A(I)-XX(I)
  C
    DO 61 I=2,N
    YY(I) = KB*(YI(I)
    B(I) = C(I)-YY(I)
  C
    IF (10*ABS(EQ,0) GO TO 62
    WRITE (C1,C2)
    L3 FORMAT (/,'10X',AFTER UPDATE:',/,'10X',A(I):',/
  C
  
```

LMS01480  
LMS01490  
LMS01500  
LMS01510  
LMS01520  
LMS01530  
LMS01540  
LMS01550  
LMS01560  
LMS01570  
LMS01580  
LMS01590  
LMS01600  
LMS01610  
LMS01620  
LMS01630  
LMS01640  
LMS01650  
LMS01660  
LMS01670  
LMS01680  
LMS01690  
LMS01700  
LMS01710  
LMS01720  
LMS01730  
LMS01740  
LMS01750  
LMS01760  
LMS01770  
LMS01780  
LMS01790  
LMS01800  
LMS01810  
LMS01820  
LMS01830  
LMS01840  
LMS01850  
LMS01860  
LMS01870  
LMS01880  
LMS01890  
LMS01900  
LMS01910  
LMS01920  
LMS01930  
LMS01940  
LMS01950

```

04 WRITE (6,64) (A(I), I=1,M)
   FORMAT (1X,F14.4)
05 WRITE (6,65)
   FORMAT (1X,10X,'AFTER UPDATE:',/,10X,'B(1):',/,/)
06 WRITE (6,64) (B(I), I=1,N)
07 CONTINUE

08 SHIFT THE INPUT DATA VECTOR THRU THE DELAY LINE BY ONE
   IF (M.EQ.1) GO TO 72
   L1 = M+1
   N1 = M-1
   DO 70 I=1,N1
   L1 = L1-1
70 X(L1) = A(L1-1)
72 X(L1) = U(K)

09 L2 = M+1
   L3 = M-1
   DO 71 I=1,L3
   L4 = L2-1
71 Y(L4) = Y(L4-1)

10 CONTINUE

11 WRITE THE ORDER AND # PTS TO OUTPUT FILE FOR GRAPHING
   NPPE=NP-1
   NPPI = 400
   N5=N-1
   WRITE (4,160) M,NPPI
   WRITE (5,160) N5,NPPI
160 FORMAT(2I5)

170 DO 170 I=1,M
   DO 170 J=1,NPPI,I
   WRITE (4,180) AA(I,J)
170 FORMAT (4,180) AA(I,J)

171 DO 171 I=1,N5
   DO 171 J=1,NPPI,I
171 WRITE (5,180) B5(I,J)
180 FORMAT (4L20.5)

191 WRITE (6,201) (C3A(I), I=1,M)
191 FORMAT (1H1,12(0X,A4),/)

202 DO 202 L=NPPI,NPPI
202 WRITE (6,202) (AA(I,L), I=1,M)

203 FORMAT (2X,1D11.5)

```

LMS01960  
 LMS01970  
 LMS01980  
 LMS01990  
 LMS02000  
 LMS02010  
 LMS02020  
 LMS02030  
 LMS02040  
 LMS02050  
 LMS02060  
 LMS02070  
 LMS02080  
 LMS02090  
 LMS02100  
 LMS02110  
 LMS02120

```

WRITE (6,201) (D(LH(I),I=1,N5)
DO 204 L=NPRI,NPP
204 WRITE (6,203) (D3(I,L),I=1,N5)
DO 206 L=1,200
206 WRITE (6,207)
207 FORMAT (1H1,/,5X, 'EITHER THE NUMERATOR OR THE DENOMINATOR ORDER'
1, ' IS INCORRECT; THAT IS, /,5X, 'EITHER M OR N IS INCORRECTLY'
2, ' SPLICED. READ IN THE CORRECT, /,5X, 'VALUES FOR M AND N AND'
3, ' TRY AGAIN.',/)
208 CONTINUE
STOP
END

```

LMSC2150  
LMSC2160  
LMSC2170  
LMSC2180  
LMSC2190  
LMSC2200  
LMSC2210  
LMSC2220  
LMSC2230  
LMSC2240  
LMSC2250  
LMSC2260  
LMSC2270

ROUTINE WMULT (A,B,N,D)  
REAL\*8 A(12),B(12),D  
INTEGER N  
L=0.000  
DO 10 I=1,N  
10 D=L+A(I)\*B(I)  
RETURN  
END



### LIST OF REFERENCES

1. Sage, A.P. and Melsa, J.L., System Identification, Academic Press, 1971.
2. Eykhoff, P., System Identification, Parameter and State Estimation, Wiley, 1974.
3. Astrom, K.J. and Eykhoff, P., "System Identification--A Survey," Automatica, Vol. 7, pp. 123-162, 1971.
4. Mehra, R.K. and Laniotis, D.G., System Identification: Advances and Case Studies, Academic Press, 1976.
5. Willsky, A.S., "Relationships Between Digital Signal Processing and Control and Estimation Theory," Proceedings of IEEE, Vol. 66, No. 9, September 1978.
6. Perry, F.A., Parametric Modeling of Linear and Nonlinear Systems, Doctoral Dissertation, Naval Postgraduate School, June 1980.
7. Romeo, M.; Multichannel ARMA Lattice Modeling with Applications to the Phase Locked Loop, Thesis, Naval Postgraduate School, December 1980.
8. Parker, S.R., "An Autoregressive Moving Average (ARMA) Discrete Nonlinear Model," Proc. of the 1980 Symposium on Circuits and Systems, pp. 918-920, April 1980.
9. Lee, R.C.K., Optimal Estimation, Identification, and Control, M.I.T. Press, Cambridge Mass., 1964.
10. Akaike, H., "Markovian Representation of Stochastic Processes and Its Application to the Analysis of Autoregressive Moving Average Processes," Institute of Statistical Mathematics, Tokyo. Annals., Vol. 26, No. 3, 1974.
11. Widrow, B., "Adaptive Filters I: Fundamentals," Stanford Electronics Labs, Stanford, Ca., Rept. SEL-66-126 (Tech. Rept 6764-6), December, 1966.
12. Clark, G.A., Block Adaptive Filtering and its Implementation to Seismic Event Detection, Doctoral Dissertation, Univ. of California, Santa Barbara, September 1980.
13. Burrus, C.S., "Block Implementation of Digital Filters," IEEE Trans. on Circuit Theory, Vol. CT-18, No. 6, pp. 697-701, November 1971.

14. Burrus, C.S., "Block Realization of Digital Filters," IEEE Trans. on Audio and Electroacoustics, Vol. AV-20, No. 4, pp. 230-235, October 1972.
15. Widrow, B. and McCool, J.M., "A Comparison of Adaptive Algorithms Based on Methods of Steepest Descent and Random Search," IEEE Trans. on Antennas and Propagation, Vol. AP-24, No. 5, pp. 615-637.
16. Markel, J.D. and Gray, Jr., A.H., Linear Prediction of Speech, Springer-Verlag, 1976.
17. Willsky, A.S., Digital Signal Processing and Control and Estimation Theory: Points of Tangency, Areas of Intersection and Parallel Directions, The MIT Press, 1979.
18. Feintuch, P.L., "An Adaptive Recursive LMS Filter," Proceedings IEEE, pp. 1622-1624, November 1976.
19. Johnson, C.R. and Larimore, M.G., "Comments on and Additions to an Adaptive Recursive LMS Filter," Proceedings IEEE, pp. 1399-1402.
20. Widrow, B. and McCool, J., "Comments on an Adaptive Recursive LMS Filter," Proceedings IEEE, pp. 1402-1404, September 1977.
21. Anderson, B.D.O. and Moore, J.R., Optimal Filtering, Prentice Hall, 1979.
22. Griffiths, L.J., "A Comparison of Lattice-Based Adaptive Algorithms," IEEE Symposium on Circuits and Systems, Proc., pp. 742-743, 1980.
23. Kalman, R.E., "New Methods and Results in Linear Prediction and Filtering Theory," Technical Report 61-1, Research Institute for Adv. Studies, Baltimore 12, Md., 1961.
24. Kalman, R.E., "A New Approach to Linear Filter and Prediction Problems," Journal of Basic Engineering, March 1961.
25. Gelb, A., editor, Applied Optimal Estimation, Analytical Science Corp., The M.I.T. Press, Cambridge, Mass., 1974.
26. Box, G.E.D. and Jenkins, G.M., Time Series Analysis, Rev. Ed., Holden-Day, 1967.
27. Maybeck, P.S., Stochastic Models, Estimation and Control, Vol. 1, Academic Press, 1979.
28. Swerling, P., "Topics in Generalized Least Squares Signal Estimation," SIAM J. Appl. Math., Vol. 14, pp. 998-1031, September 1966.

29. Swerling, P., "Modern State Estimation Methods from the Viewpoint of the Method of Least Squares," IEEE Trans. on Auto. Cont., Vol. AC-16, No. 6, December 1971.
30. Swerling, P., "Classes of Signal Processing Procedures Suggested by Exact Minimum Mean Square Error Procedures," SIAM J. Appl. Math., Vol. 14, No. 6, pp. 1199-1224, November 1966.
31. Penrose, R., "A Generalized Inverse for Matrices," Proceedings of the Cambridge Philosophical Society, Vol. 51, 1955.
32. Penrose, R., "On Best Approximate Solutions of Linear Matrix Equations," Proceedings of the Cambridge Philosophical Society, Vol. 52, 1956.
33. Anderson, T.W., The Statistical Analysis of Time Series, Wiley, 1971.
34. Levinson, N., "The Weiner RMS Error Criteria in Filter Design and Prediction," J. Math Physics, Vol. 25, pp. 261-278, January 1947.
35. Ahmed, N. and Fogler, R.J., "On an Adaptive Lattice Prediction and a Related Application," IEEE Circuits and Systems Magazine, Vol. 1, No. 4, pp. 17-23, 1979.
36. Ben-Yakov, U., Efficient Adaptive FIR and IIR Filters, Doctoral Dissertation, Naval Postgraduate School, December 1979.
37. Antoniou, A., Digital Filters Analysis and Design, McGraw-Hill, 1979.
38. Oppenheim, A.V. and Schafer, R.N., Digital Signal Processing, Prentice Hall, Inc., 1975.
39. Kirk, D.E., "Optimal Estimation: An Introduction to the Theory and Applications," Notes at the Naval Postgraduate School, 1975.
40. Carpenter, D.D., "The Effect of Time Delays on the Stability of Practical Phase Locked Receivers with Multiple Conversion by Utilizing Root Locus with Positive Feedback," 1973 IEEE Int. Conf. on Communications, Vol. II, June 11-13, 1973, pp. 33-14--33-19.

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 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4. Professor S.R. Parker, Code 62Px Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	6
5. Professor G.A. Myers, Code 62Mv Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Professor F.R. Richards, Code 55Rh Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
7. Professor D.J. Collins, Code 67Co Department of Aerospace Engineering Naval Postgraduate School Monterey, California 93940	1
8. Professor D.E. Kirk, Code 62Ki Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
9. Professor G. Thaler, Code 62Tr Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
10. Cdr. R.A. Persell, Code 32 Curricular Officer Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1

11. Lt. F.A. Perry, Code 71 1  
c/o Commander Naval Ocean Systems Command  
San Diego, California 92152
12. Lcdr. D.J. Grandia, SMC 2708 1  
Naval Postgraduate School  
Monterey, California 93940
13. Lcdr. D.J. Klich, SMC 1433 1  
Naval Postgraduate School  
Monterey, California 93940
14. LCOL. A. Feit, SMC 1073 1  
Naval Postgraduate School  
Monterey, California 93940
15. Lt. L.M. Mayoral 4  
c/o Commander Carrier Group Seven  
FPO San Francisco 96601

**DATE**  
**ILME**