APPLICATION OF THE METHOD OF LEAST SQUARES TO ADAPTIVE SYSTEMS IDENTIFICATION

by

DAVID LYNN SOLDAN

B. S., Kansas State University, 1969

A MASTER'S THESIS

submitted in partial fulfullment of the

requirements for the degree

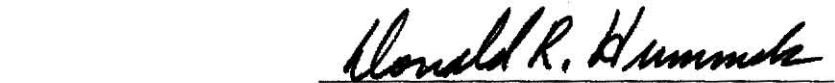
MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1976

Approved by:



Major Professor

TABLE OF CONTENTS

PAGE

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The problem considered is that of modeling an unknown system with an adaptive digital system. This problem has recently been considered by Widrow [1,2,3], Stearns [4] and Feintuch [5] who have each applied the method of steepest descent to obtain different control algorithms for adapting the digital system used to model the unknown system. The general approach taken is shown in Figure 1. The unknown system is forced by a signal x(t), samples of which are also input to the adaptive system and the controller. Samples of the output of the unknown system, perhaps corrupted by additive noise, n(t), are then compared with the outputs of the adaptive digital system to obtain an error signal. The error signal, e(k), is the major input to the controller which uses an algorithm to compute the appropriate adjustments in the adaptive system. Algorithms and control strategies are designed with the goal of minimizing the square of the error. They differ in their convergence properties and their error performance.

The approach of Figure 1 can be derived by application of a least squares procedure without resorting to the method of steepest descent, and with the aid of simplifying assumptions most of the previous algorithms are easily obtained as special cases of the least squares approach.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

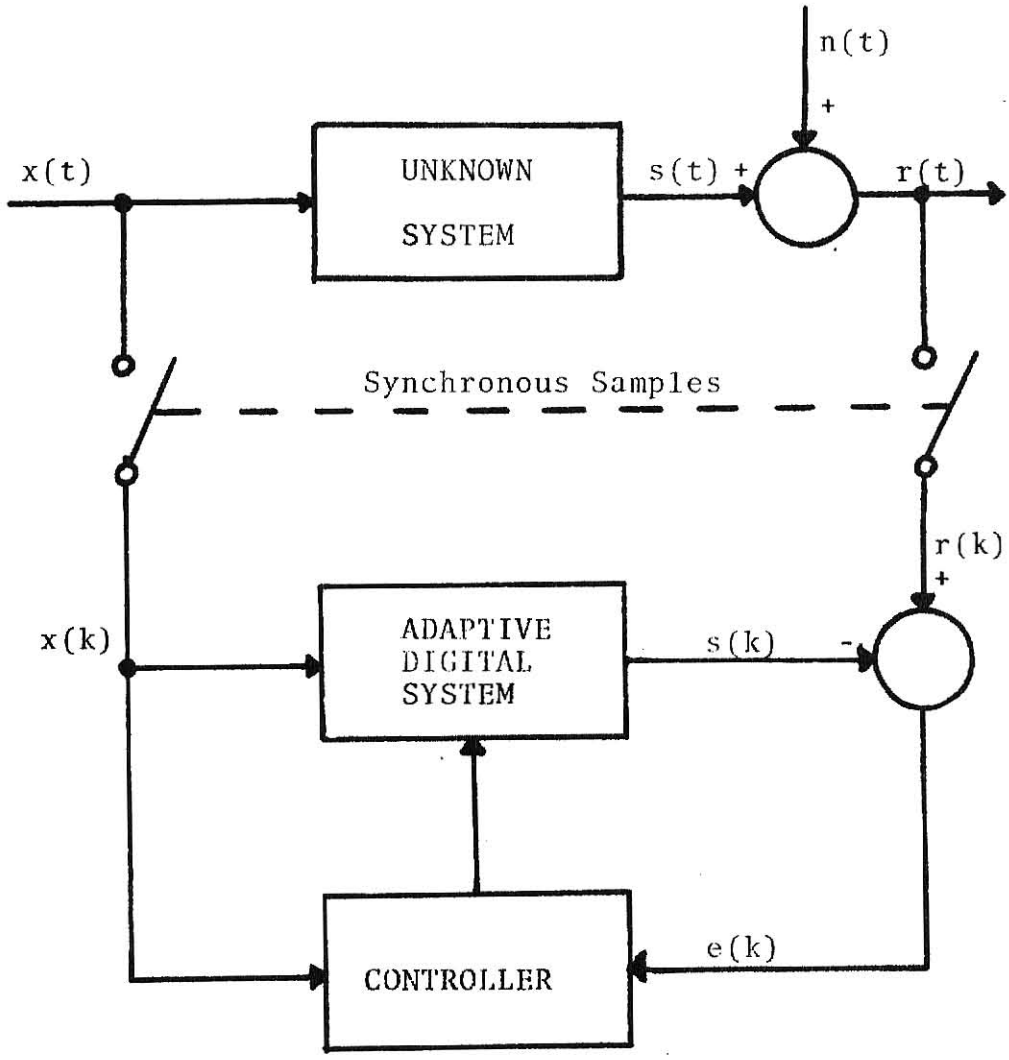


Figure 1. Modeling an unknown system with an adaptive digital system.

The simplifications necessary to arrive at Widrow's and Stearns' algorithms are given and a new algorithm which is a variation of Stearns' approach is developed.

The least squares algorithm is compared to Widrow's and Stearns' algorithms for the systems identification problem. Simple FIR and IIR unknown systems are used. The advantages of the least squares algorithm are its fast convergence time and its ability to track changes in the unknown system.

CHAPTER II

LEAST SQUARES ESTIMATION

A. VECTOR OBSERVATIONS

The following results are well documented in the literature [6,7,8,9,10]. Details for the risk function used in this system modeling application are given. There are a number of different techniques for obtaining a recursive least squares algorithm. The approach taken is basically that of Swerling [6] and Blackman [7].

Suppose that an estimate of a parameter vector α of dimension N is desired to the k-th stage, and that the estimate is to be obtained as an improvement of the estimate $\hat{\alpha}_{k-1}$ from the previous stage. The improvement is to be based on a new observation or measurement r_k where r_k is a vector of dimension M. If $s_k(\alpha)$ is a vector function of α, possibly nonlinear, then a suitable scalar risk function for finding a least squares estimate of α is

$$R(\alpha) = (\alpha - \alpha_{k-1})^T C_{k-1}^{-1} (\alpha - \alpha_{k-1}) + [r_k - s_k(\alpha)]^T C_n^{-1} [r_k - s_k(\alpha)], \quad (1)$$

where C_{k-1}^{-1} and C_n^{-1} are symmetric positive definite weighting matrices. When such data are available, they are usually taken as the inverse covariance matrices of the previous estimate, α_{k-1}, and the observation error, n_k. Statistical

descriptions, however, are not required in a least squares approach.

Expanding $s_k(\alpha)$ in a Taylor series about $\hat{\alpha}_{k-1}$ and retaining only the linear terms for substitution in (1) yields

$$\begin{aligned} R(\alpha) \simeq\ & (\alpha - \hat{\alpha}_{k-1})^T C_{k-1}^{-1} (\alpha - \hat{\alpha}_{k-1}) \\ & + \{r_k - s_k(\hat{\alpha}_{k-1}) - [\nabla_\alpha s_k^T(\hat{\alpha}_{k-1})]^T \\ & \cdot (\alpha - \hat{\alpha}_{k-1})\}^T C_n^{-1} \{r_k - s_k(\hat{\alpha}_{k-1}) \\ & - [\nabla_\alpha s_k^T(\hat{\alpha}_{k-1})]^T (\alpha - \hat{\alpha}_{k-1})\} \end{aligned} \tag{2}$$

where ∇_α is a differential operator used to indicate differentiation with respect to the vector α and

$$\nabla_\alpha s_k^T(\hat{\alpha}_{k-1}) \equiv \nabla_\alpha s_k^T(\alpha)\Big|_{\alpha=\hat{\alpha}_{k-1}} . \tag{3}$$

The following notation will be used:

$$\nabla S \equiv \nabla_\alpha s_k^T(\hat{\alpha}_{k-1}) \quad \text{and} \quad \nabla S^T \equiv [\nabla_\alpha s_k^T(\hat{\alpha}_{k-1})]^T \tag{4}$$

Minimizing $R(\alpha)$ requires taking the vector derivative of (2) and setting it to zero, i.e.,

$$\nabla_\alpha R(\alpha)\Big|_{\alpha=\hat{\alpha}_k} = 0 . \tag{5}$$

Differentiation of (3) yields

$$\begin{aligned} \nabla_\alpha R(\alpha) = & \ 2C_{k-1}^{-1}(\alpha - \hat{\alpha}_{k-1}) + 2\nabla_\alpha [-(\nabla S^T)\alpha]^T C_n^{-1} \\ & \cdot [r_k - s_k(\hat{\alpha}_{k-1}) - \nabla S^T(\alpha - \hat{\alpha}_{k-1})] \end{aligned} \tag{6}$$

Simplification of (6) leads to

$$0 = C_{k-1}^{-1}(\hat{\alpha}_k - \hat{\alpha}_{k-1}) - \nabla S(C_n^{-1}) \cdot [r_k - s_k(\hat{\alpha}_{k-1}) - \nabla S^T(\hat{\alpha}_k - \hat{\alpha}_{k-1})] \quad . \tag{7}$$

Solving for $\hat{\alpha}_k$, the estimate of α yields

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + [C_{k-1}^{-1} + \nabla S\ C_n^{-1}\ \nabla S^T]^{-1} \cdot \nabla S\ C_n^{-1}\ [r_k - s_k(\hat{\alpha}_{k-1})] \quad . \tag{8}$$

If a statistical description is given for α, it is not difficult to show [7] that the covariance matrix C_k of the estimate $\hat{\alpha}_k$ is given by the lengthy inverse term of (8), i. e.,

$$C_k = [C_{k-1}^{-1} + \nabla S\ C_n^{-1}\ \nabla S^T]^{-1} \quad . \tag{9}$$

In the absence of a statistical description it is still desirable to use C_k^{-1} as a weighting matrix at stage k+1. Equations (8) and (9) thus define a recursive procedure for calculating the least squares estimate of α. A very similiar result is derived by Sage [8] using a risk function based on all previous observations.

B. SCALAR OBSERVATIONS

While vector observations occur often, the application to systems modeling considered has scalar observations. When r_k is a scalar it is possible to manipulate (8) to eliminate the cumbersome inverse. Examples of this procedure are given in [7] and [8]. C_n becomes a scalar and will be written as

σ_n^2 to emphasize it scalar nature. Equation (8) becomes

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + [C_{k-1}^{-1} \sigma_n^2 + \nabla S \nabla S^T]^{-1} \nabla S [r_k - s_k(\hat{\alpha}_{k-1})] . \quad (10)$$

With γ a scalar defined as

$$\gamma = \nabla S^T C_{k-1} \nabla S + \sigma_n^2 \quad (11)$$

∇S can be written as

$$\nabla S = \frac{1}{\gamma}[\nabla S \nabla S^T C_{k-1} \nabla S + \sigma_n^2 C_{k-1}^{-1} C_{k-1} \nabla S] , \quad (12)$$

since $C_{k-1}^{-1} C_{k-1} = I$ where I is the identity. Factoring $C_{k-1} \nabla S$ to the right yields

$$\nabla S = \frac{1}{\gamma}[\nabla S \nabla S^T + \sigma_n^2 C_{k-1}^{-1}] C_{k-1} \nabla S . \quad (13)$$

Substitution of (13) into (10) yields

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + [\nabla S \nabla S^T + \sigma_n^2 C_{k-1}^{-1}]^{-1} \frac{1}{\gamma} \cdot [\nabla S \nabla S^T + \sigma_n^2 C_{k-1}^{-1}] C_{k-1} \nabla S [r_k - s_k(\hat{\alpha}_{k-1})] , \quad (14)$$

which reduces to

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + \frac{1}{\gamma} C_{k-1} \nabla S [r_k - s_k(\hat{\alpha}_{k-1})] \quad (15)$$

with γ as defined by (11). Blackman [7] has shown that the covariance of the estimate is

$$C_k = [C_{k-1} + \nabla S C_n^{-1} \nabla S^T]^{-1} . \quad (16)$$

Some simplification of (16) is necessary to give an expression that can be used to recursively calculate C_k. As before

$C_n = \sigma_n^2$. Subtracting C_{k-1} from each side of (16) yields

$$C_k - C_{k-1} = \sigma_n^2 \left[\nabla S \, \nabla S^T + \sigma_n^2 \, C_{k-1}^{-1}\right]^{-1} - C_{k-1} \, . \tag{17}$$

or

$$\left[\nabla S \, \nabla S^T + \sigma_n^2 \, C_{k-1}^{-1}\right](C_k - C_{k-1}) = \sigma_n^2 \, I$$

$$- \left[\nabla S \, \nabla S^T + \sigma_n^2 \, C_{k-1}^{-1}\right] C_{k-1} \, . \tag{18}$$

Combining (13) and (16) gives

$$\nabla S = \frac{1}{\gamma \, \sigma_n^2} \, C_k^{-1} \, C_{k-1} \, \nabla S \, . \tag{19}$$

Substitution of (19) into (18) yields

$$\frac{1}{\sigma_n^2} \, C_k^{-1} \, (C_k - C_{k-1}) = \sigma_n^2 \, I$$

$$- \left[\frac{1}{\gamma \, \sigma_n^2} \, C_k^{-1} \, C_{k-1} \, \nabla S \, \nabla S^T + \sigma_n^2 \, C_{k-1}^{-1}\right] C_{k-1} . \tag{20}$$

Multiplying both sides by C_k and then solving for C_k

$$C_k = C_{k-1} - \frac{1}{\gamma} \, C_{k-1} \, \nabla S \, \nabla S^T \, C_{k-1} \, . \tag{21}$$

The least squares algorithm is now defined by (11), (15) and (21). The algorithm is summarized below.

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + \frac{1}{\gamma} \, C_{k-1} \, \nabla S \, [r_k - s_k(\hat{\alpha}_{k-1})]$$

$$C_k = C_{k-1} - \frac{1}{\gamma} \, C_{k-1} \, \nabla S \, \nabla S^T \, C_{k-1}$$

$$\gamma = \nabla S^T \, C_{k-1} \, \nabla S + \sigma_n^2 \, .$$

The vectors $\hat{\alpha}_k$ and $\hat{\alpha}_{k-1}$ are still N-dimensional and matrices

C_k and C_{k-1} are N x N. $s_k(\hat{\alpha}_{k-1})$ is a scalar and ∇S is a vector of dimension N.

CHAPTER III

APPLICATIONS IN SYSTEMS MODELING

A. GENERAL MODEL

In systems modeling the function $s_k(\alpha)$ is determined by the form assumed for the model. Both finite impulse response (FIR) and infinite impulse response (IIR) forms will be considered. The components of the vector α are the parameters of the model and the goal is to adjust these for the least squared error between the output of the unknown system and the model. The system output depends both on the system parameters α and the input x. Since the problem considered is that of estimating α the x dependence has been suppressed in the preceding. The application of least squares procedures to the system modeling problem suggests the structure illustrated in Figure 2. This structure is an implementation of (11), (15) and (21), the least squares approach.

B. FINITE IMPULSE RESPONSE SYSTEMS

One of the initial steps in applying the preceding results is to choose a specific $s_k(\alpha)$. In what follows next it is assumed that a finite impulse response (FIR) digital filter is the most suitable form. This corresponds to the case of Widrow [1,2,3]. For an FIR filter the output can be represented by

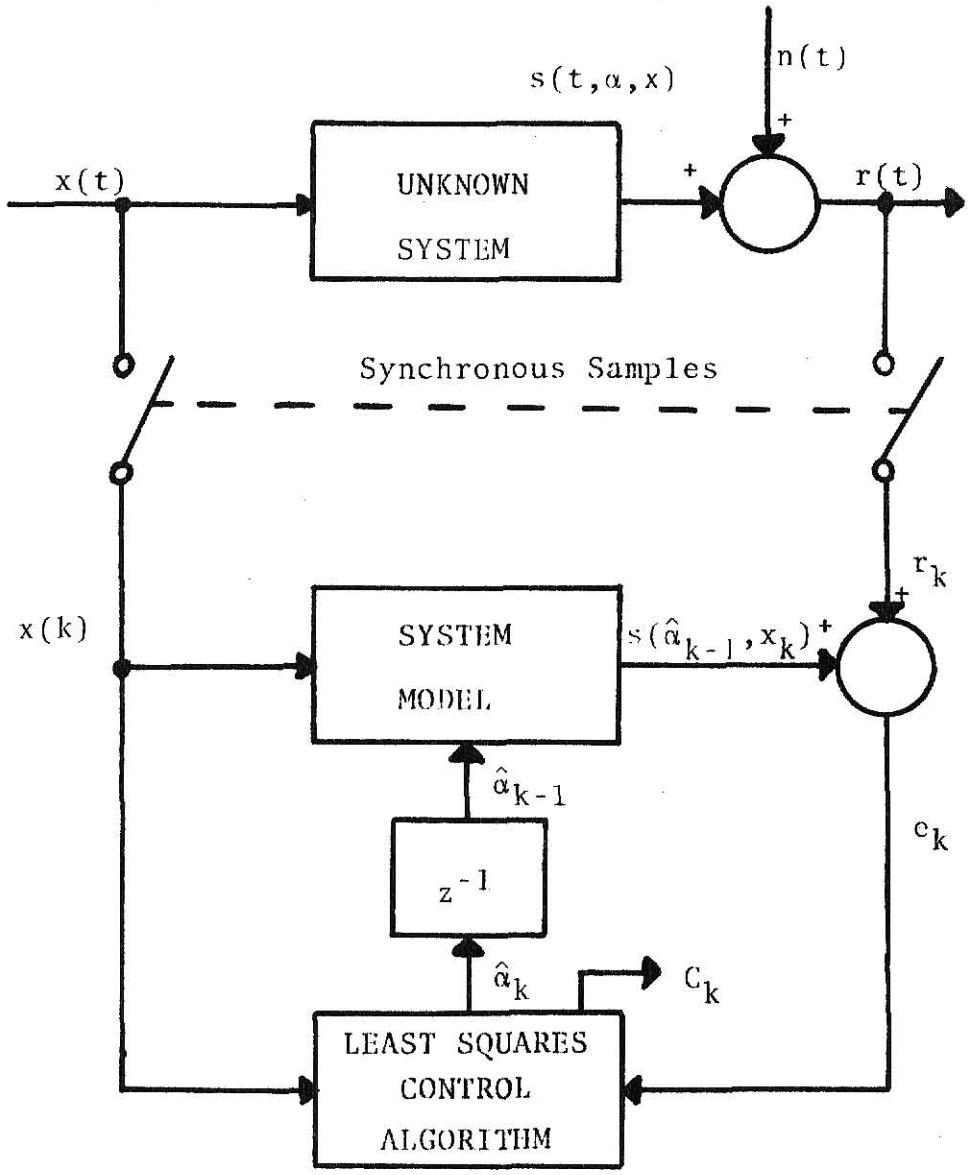


Figure 2. Least squares implementation.

$$s_k(\alpha) = \sum_{i=0}^{N-1} w_i \, x(k-i) \quad . \tag{22}$$

Defining the vectors α and x_k as

$$\alpha^T = [w_0 \quad w_1 \cdots w_{N-1}] \tag{23}$$

$$x_k^T = [x(k) \quad x(k-1) \cdots x(k-N-1)] \tag{24}$$

leads to a more convenient vector form

$$s_k(\alpha) = x_k^T \alpha \quad . \tag{25}$$

The derivative indicated in (8) and (15) is then

$$\nabla S = x_k \tag{26}$$

A Least Squares Algorithm

Substitution of (25) and (26) into (11), (15) and (21) yields a recursive least squares algorithm for estimating the weights of the FIR system. The resultant algorithm is

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + \frac{1}{\gamma} C_{k-1} \, x_k \, e_k \quad , \tag{27}$$

$$\gamma = x_k^T C_{k-1} \, x_k + \sigma_n^2 \quad , \tag{28}$$

$$C_k = C_{k-1} - \frac{1}{\gamma} C_{k-1} \, x_k \, x_k^T \, C_{k-1} \quad , \tag{29}$$

where $$e_k = r_k - x_k^T \hat{\alpha}_{k-1} \quad . \tag{30}$$

A similiar result has been derived by Graupe [8] using a slightly different approach.

Reduction to the LMS Algorithm

Widrow [1,2,3] has used the method of steepest descent to develop an algorithm for the FIR model. His result may be expressed as

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + 2\,\mu\, x_k\, e_k \,. \tag{31}$$

The parameter μ is a free parameter used to control stability and rate of convergence.

Reduction of the least squares result (27) to Widrow's form is most readily achieved by starting at an earlier point with (8) and (9). For the case of an FIR model, (8) becomes

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + (C_{k-1}^{-1} + \frac{1}{\sigma_n^2}\, x_k\, x_k^T)^{-1}\, x_k\, \frac{1}{\sigma^2}\, e_k \tag{32}$$

and (9) reduces to

$$C_k = (C_{k-1}^{-1} + \frac{1}{\sigma_n^2}\, x_k\, x_k^T)^{-1} \,. \tag{33}$$

Considerable simplification can be obtained by arbitrarily avoiding calculation of the inverse in (33) and substituting a constant matrix in its place. Since this matrix corresponds to the covariance matrix of the estimate, this amounts to assuming that the quality of the estimate remains constant as time proceeds. While it is anticipated that the convergence properties of the algorithm will be impaired by this choice, it is expected that the desired estimate should still be obtained unless a particularly adverse choice is made. Assume that C_{k-1} is a constant matrix of the form

$$C_{k-1} = \sigma_\alpha^2 \, I \tag{34}$$

where I is the identity. Next replace the matrix $x_k \, x_k^T$ by its average. If white Gaussian noise is used for the input, x(t), the appropriate substitution is

$$E[x_k \, x_k^T] = \sigma_x^2 \, I \; . \tag{35}$$

Substituting in (32) and manipulating yields

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + \frac{\sigma_\alpha^2}{\sigma_n^2 + \sigma_\alpha^2 \, \sigma_x^2} \, x_k \, e_k \; . \tag{36}$$

If the observation error is small so that $\sigma_n^2 << \sigma_\alpha^2 \, \sigma_x^2$ further simplification may be achieved so that (36) becomes

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + \frac{1}{\sigma_x^2} \, x_k \, e_k \; . \tag{37}$$

With $\sigma_x^2 = 1/2\mu$ this is the LMS algorithm developed by Widrow.

Comparison of the FIR Algorithms

A comparison of the convergence characteristics of these two algorithms shows the superior performance of the least squares approach. The models used for comparison are that of Figure 1 for the LMS algorithm and Figure 2 for the least squares algorithm. Each adaptive system had five coefficients or weights. The "unknown" system was a four weight FIR digital filter characterized by the difference equation

$$s_k(\alpha) = a_0 \, x(k) + a_1 \, x(k-1) + a_2 \, x(k-2) + a_3 \, x(k-3)$$

$$\text{with} \qquad a_0 = 0.50 \qquad a_1 = 0.50 \qquad a_2 = 0.50 \qquad a_3 = 0.50 \tag{38}$$

This represents a low pass filter.

The algorithms were simulated on an IBM System 370. The programs used are given in Appendix A. The input to the system was white Gaussian noise generated by the standard IBM Scientific Subroutine Gaussian number generator. The noise had zero mean with $\sigma^2 = 0.1$. The value of μ in (31) was chosen as $1/\lambda_{max}$ to achieve minimum convergence time [1,2,3].

The results shown in Figures 3 and 4 are the average of five simulations using different input sequences. Figure 3 represents the average squared error curves for both algorithms. The least squares algorithm converges much more rapidly than the LMS algorithm. This rapid convergence characteristic allows the least squares algorithm to track changes in the unknown system within a few iterations while the LMS algorithm requires many iterations to adjust. To demonstrate this tracking ability the unknown was changed from (38) to

$$\begin{aligned} a_0 &= 0.75 & a_1 &= 0.50 \\ a_2 &= 0.25 & a_3 &= 0.25 \end{aligned} \tag{39}$$

after 100 iterations. The resulting squared error curves are shown in Figure 4.

C. INFINITE IMPULSE RESPONSE SYSTEMS

A recursive form is used for the infinite impulse response (IIR) model. Let

$$s_k(\alpha) = \sum_{\ell=0}^{L} a_\ell \, x(k-\ell) + \sum_{j=1}^{J} b_j \, s_{k-j}(\alpha) \, . \tag{40}$$

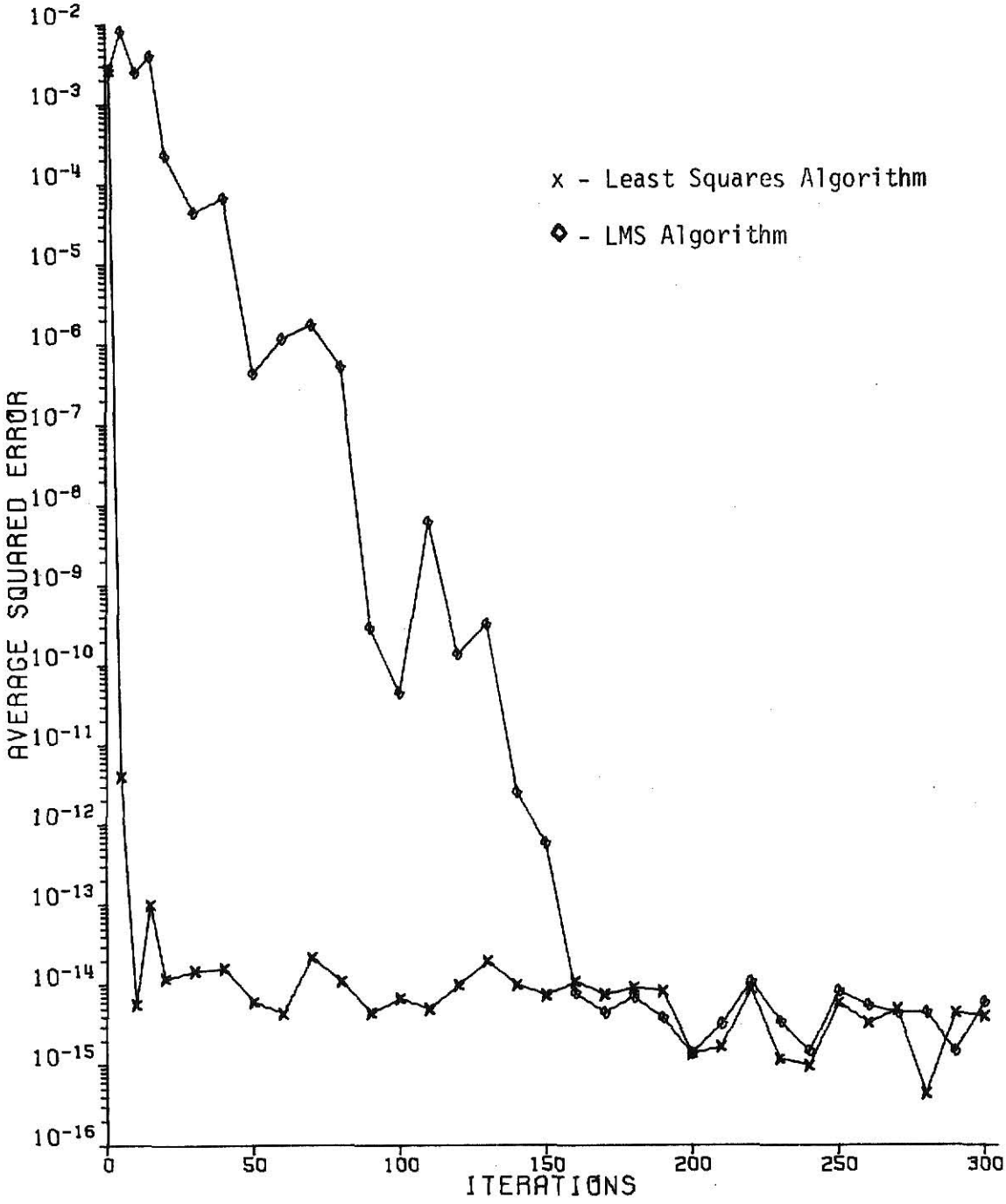


Figure 3. Convergence characteristics of FIR algorithms.

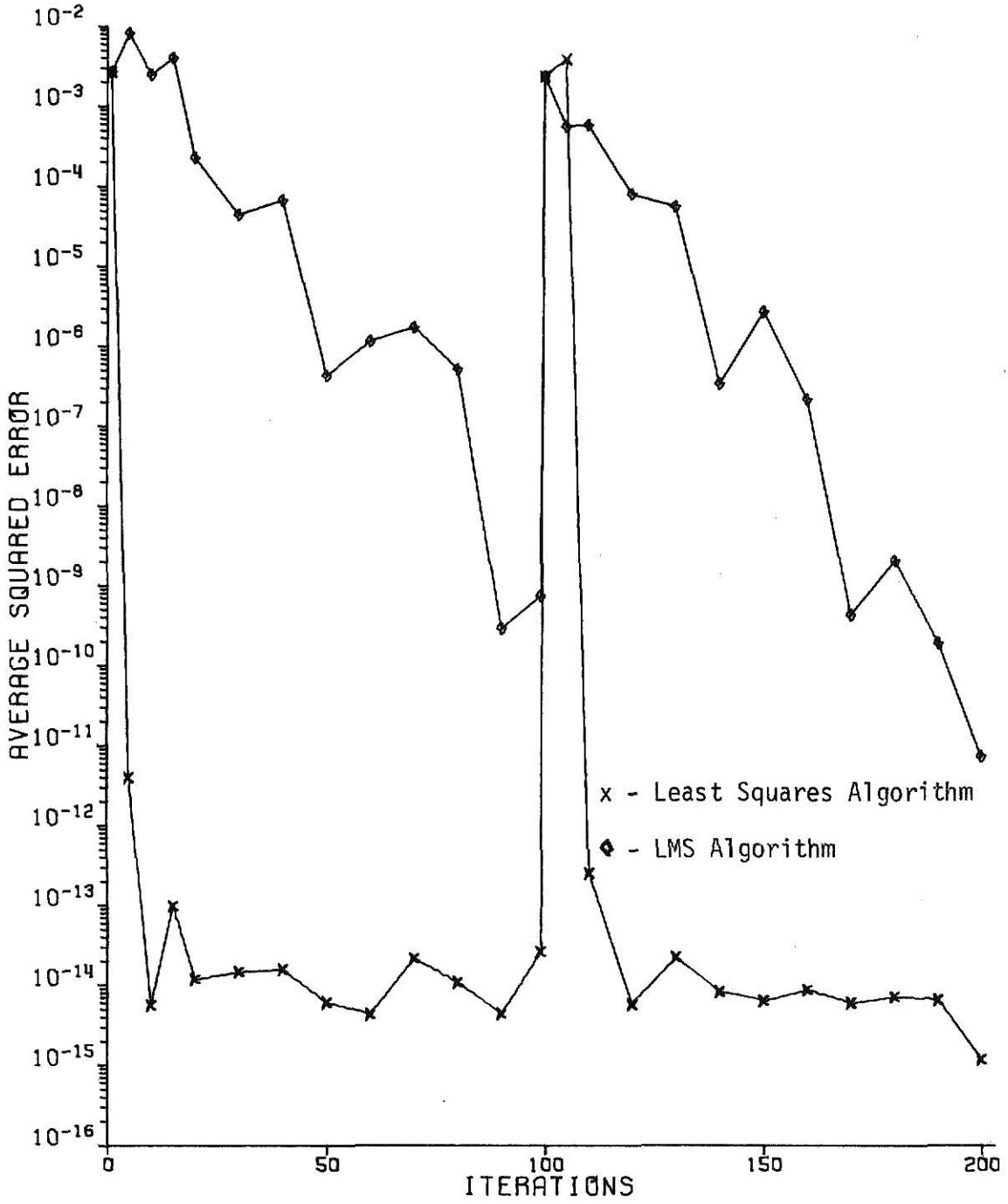


Figure 4. Tracking characteristics of FIR algorithms.

In this case, the output of the system not only depends on the input but also on the previous outputs. By defining vectors

$$\alpha^T = [a_0 \quad a_1 \cdots a_L \quad b_1 \quad b_2 \cdots b_J] \tag{41}$$

and

$$\beta_k^T = [x(k) \; x(k-1) \cdots x(k-L) \; s_{k-1}(\alpha) \cdots s_{k-J}(\alpha)] \; , \tag{42}$$

$s_k(\alpha)$ can be expressed in vector form as

$$s_k(\alpha) = \beta_k^T \alpha \; . \tag{43}$$

A Least Squares Algorithm

Using (41), (42) and (43) the derivatives of $s_k(\alpha)$ are

$$\nabla_\alpha s_k(\alpha) = \begin{bmatrix} \frac{\partial s_k(\alpha)}{\partial a_0} \\ \vdots \\ \frac{\partial s_k(\alpha)}{\partial a_L} \\ \frac{\partial s_k(\alpha)}{\partial b_1} \\ \frac{\partial s_k(\alpha)}{\partial b_J} \end{bmatrix} = \begin{bmatrix} x(k) + b_1 \frac{\partial s_{k-1}(\alpha)}{\partial a_0} + \cdots + b_J \frac{\partial s_{k-J}(\alpha)}{\partial a_0} \\ \vdots \\ x(k-L) + b_1 \frac{\partial s_{k-1}(\alpha)}{\partial a_L} + \cdots + b_J \frac{\partial s_{k-J}(\alpha)}{\partial a_L} \\ s_{k-1}(\alpha) + b_1 \frac{\partial s_{k-1}(\alpha)}{\partial b_1} + \cdots + b_J \frac{\partial s_{k-J}(\alpha)}{\partial b_1} \\ s_{k-J}(\alpha) + b_1 \frac{\partial s_{k-1}(\alpha)}{\partial b_J} + \cdots + b_J \frac{\partial s_{k-J}(\alpha)}{\partial b_J} \end{bmatrix} \tag{44}$$

or written more compactly

$$\nabla S = \beta_k + b_1 \nabla_\alpha s_{k-1}(\alpha) + \cdots + b_J \nabla_\alpha s_{k-J}(\alpha) \; . \tag{45}$$

Using (43) and (45) together with (11), (15) and (21) yields a least squares algorithm for the IIR model. Although the algorithm does involve matrices, it is free from matrix inversions and not unduly complicated. It differs from the FIR case in that calculations of both the filter output and its derivatives are retained for calculations of subsequent estimates.

For many physical systems which have infinite or extremely long impulse responses, a good FIR model would require a very large number of weights, i. e., a vector α of large dimension would be needed. In such cases it is expected that implementation of the IIR model will be simpler.

Reduction to the Case of Stearns

The first work with an IIR model was that of Stearns [4]. As in the case of Widrow, Stearns used the method of steepest descent. By assuming a constant covariance matrix Stearns' algorithm can be derived from the least squares algorithm as given by (8) and (9). Let C be a constant diagonal matrix and replace the inverse in (8) by σ_n^2 C. For the IIR case this gives

$$\hat{\alpha}_k = \hat{\alpha}_{k-1} + C\,[\nabla_\alpha s_k(\hat{\alpha}_{k-1})]\, e_k \,. \tag{46}$$

Now, if the derivatives are formed using the recursion

$$\nabla S = \beta_k + b_1 \nabla_\alpha s_{k-1}(\hat{\alpha}_{k-2}) + \cdots + b_J \nabla_\alpha s_{k-J}(\hat{\alpha}_{k-1-J}) \,, \tag{47}$$

the result is an algorithm essentially the same as that of

Stearns. The constants of the diagonal matrix are selected for convergence and stability.

Another suboptimum but computationally simpler version has been suggested by Feintuch[5]. In this case the derivative is approximated by only the first term of (47), $\nabla S = \beta_k$.

A New IIR Algorithm

Equation (40) may be written in an FIR form by substituting previous values of $s_k(\alpha)$ in the second summation of (40). These previous values are derived from (40) itself by replacing k with k-1, k-2, etc. This yields [11]

$$s_k(\alpha) = \sum_{n=0}^{\infty} c_n \, x(k-n). \tag{48}$$

The c_n's are functions of the components of α as defined in (41).

$\nabla_\alpha s_k(\alpha)$ becomes a function only of α, the filter parameters, and X_k, the inputs. A method for determining the c_n and $\nabla_\alpha s_k(\alpha)$ has been developed and is given in Appendix B.

For the case L=J=2 in (40) and using the first five terms of the summation

$$\nabla s_k(\hat{\alpha}_{k-1}) = \begin{bmatrix} 1 & d_0 & d_1 & d_2 & d_3 \\ 0 & 1 & d_0 & d_1 & d_2 \\ 0 & 0 & 1 & d_0 & d_1 \\ 0 & d_4 & d_5 & d_6 & d_7 \\ 0 & 0 & d_4 & d_5 & d_6 \end{bmatrix} [X_k] \tag{49}$$

where

$$d_0 = b_1$$

$$d_1 = b_1^2 + b_2$$

$$d_2 = b_1^3 + 2b_1b_2$$

$$d_3 = b_1^4 + 3b_1^2b_2 + b_2^2$$

$$d_4 = a_0$$

$$d_5 = a_1 + 2b_1a_0$$

$$d_6 = 3b_1^2a_0 + 2b_2a_0 + 2b_1a_1 + a_2$$

$$d_7 = 4b_1^3a_0 + 6b_2b_1a_0 + 3b_1^2a_1 + 2b_2a_1 + 2b_1a_2$$

With this algorithm as defined by (46) and (49) ∇S is computed each iteration without using previous values of itself in the computation. In the systems modeling application discussed in the next section five terms are needed in (49) to achieve results that are equivalent to those of Stearns' algorithm.

Comparison of IIR Algorithms

The models used for comparison are the same as those used for the FIR case with Figure 1 representing Stearns' algorithm. Each adaptive system was a five weight IIR system with two zeroes and three poles. The input sequences were identical to those used in the FIR case. The "unknown" system was a five weight IIR low pass filter characterized by

$$s_k(\alpha) = a_0 x(k) + a_1 x(k-1) + a_2 x(k-2)$$
$$+ b_1 s_{k-1}(\alpha) + b_2 s_{k-2}(\alpha) \qquad (50)$$

where

$$a_0 = 1.00$$

$$a_1 = 0.00$$

$$a_2 = 0.49$$

$$b_1 = 0.50$$

$$b_2 = 0.06$$

The programs used for simulation on the IBM 370 appear in Appendix A. The value of C in (32) was chosen for minimum convergence time. The results shown in Figures 5 and 6 are an average of five simulations.

Figure 5 gives average squared error curves for all three algorithms. The new suboptimum algorithm uses five terms of the infinite series in its computation. The least squares algorithm converges much faster than the other two algorithms. This allows the least squares algorithm to track changes in the unknown much faster than the other algorithms. Figure 6 shows average squared error curves for the least squares and Stearns' algorithms when the unknown (50) is changed to

$$a_0 = 0.90$$

$$a_1 = 0.00$$

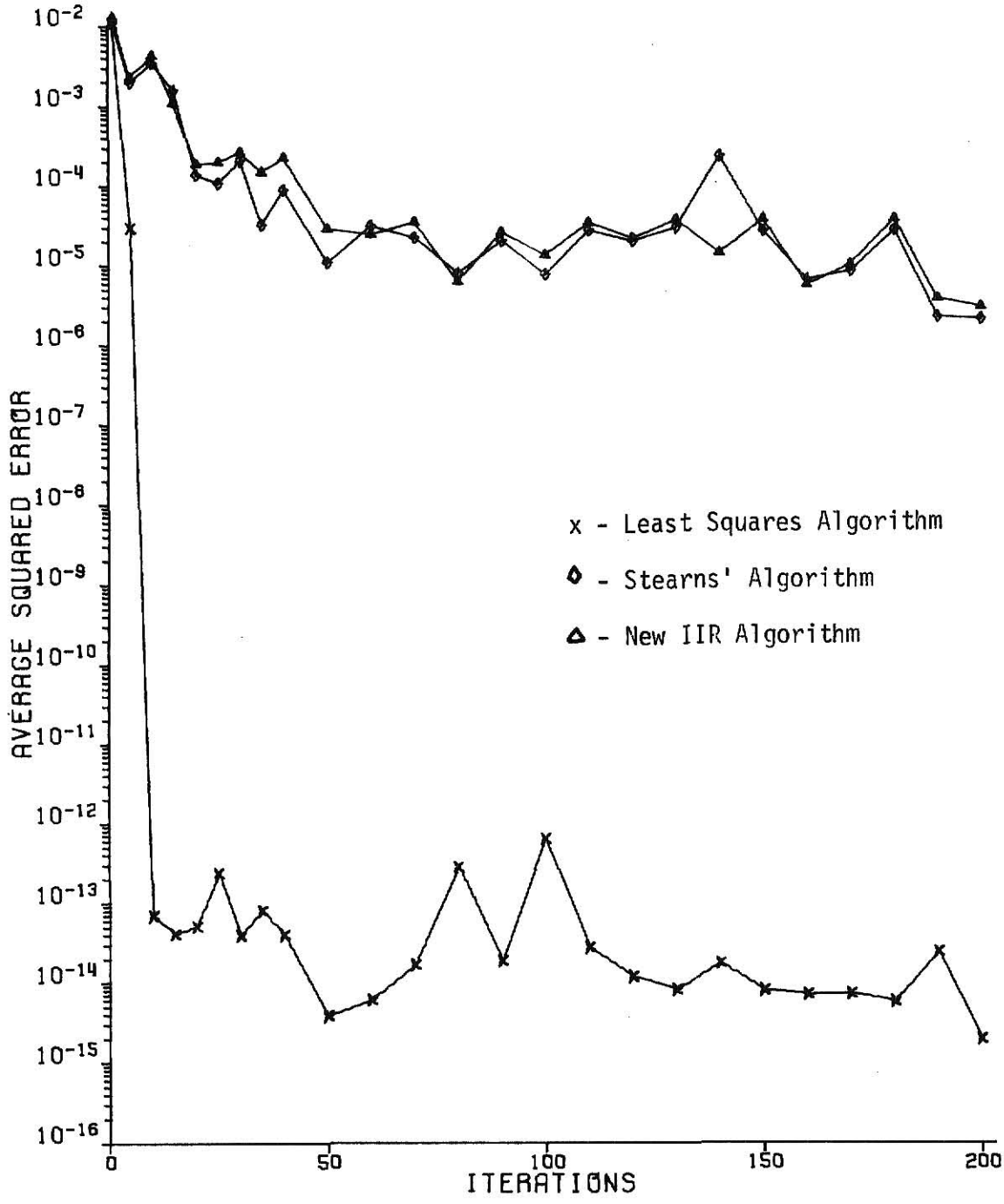


Figure 5. Convergence characteristics of IIR algorithms.

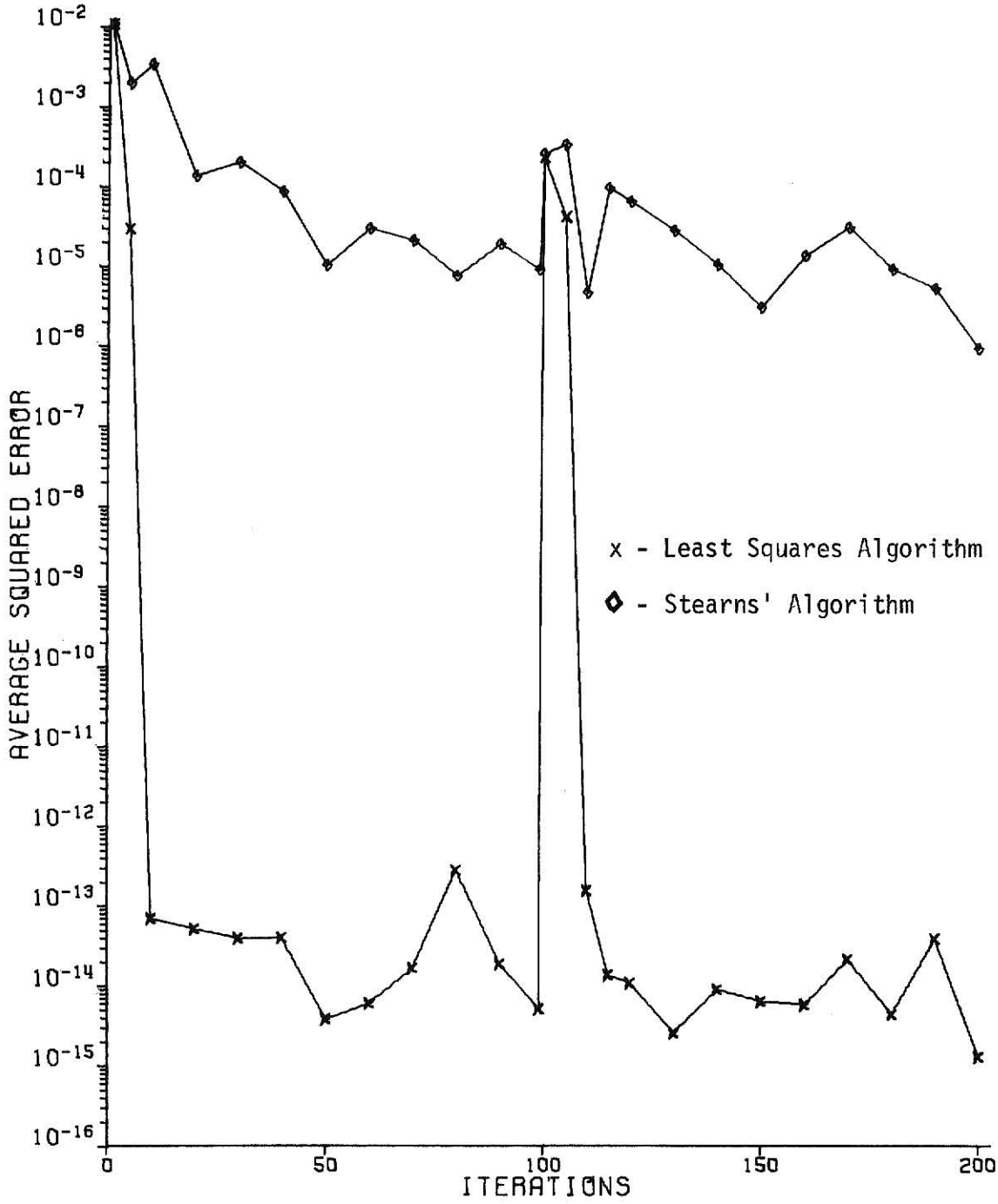


Figure 6. Tracking characteristics of IIR algorithms.

$$a_2 = 0.35$$

$$b_1 = 0.25$$

$$b_2 = 0.00$$

after 100 iterations.

CHAPTER IV

CONCLUSION

The method of least squares has been applied to derive an algorithm for modeling an unknown system with an adaptive digital system. Although this algorithm requires matrix operations, no matrix inversions are needed. Implementation of the least squares algorithm in software requires more statements than some previous algorithms but is straightforward. A comparison of the least squares algorithm with previous algorithms shows the superior convergence characteristics of the least squares approach. In addition the ability of the model to adapt to changes in the unknown system is superior to that of models based on previous algorithms.

A new IIR suboptimum algorithm that does not require storing previous values of the gradient was developed and compared to previous IIR algorithms. These comparisons showed equivalent convergence characteristics were achieved with five terms of the infinite sum for the new algorithm.

It has also been shown that the previous algorithms, both FIR and IIR, can be derived as special cases of the least squares approach.

Further study is needed in the area of modeling larger systems with adaptive digital systems. An anticipated problem in this area is the effect of coefficient round off error. Another area where the least squares approach should find many applications is that of adaptive noise cancelling.

APPENDIX A

FORTRAN PROGRAMS

This appendix contains copies of the FORTRAN programs used to simulate the modeling of unknown systems as described in Chapter 3. The WATFIV compiler was used. Work was done on the Kansas State University IBM 370 system. The following five programs are included.

1. Five Weight LMS Algorithm.
2. Five Weight FIR Least Squares Algorithm.
3. Stearns' Algorithm.
4. Five Weight IIR Algorithm.
5. Five Weight IIR Least Squares Algorithm.

1. FIVE WEIGHT LMS ALGORITHM

```
      DIMENSION X(500)
      REAL MU
  102 FORMAT(' ',I6,3F12.4,E12.4,5F12.4)
  104 FORMAT(5F5.3)
  105 FORMAT(F8.4,I9,I4,2F7.4)
      READ105,MU,IX,M,B,C
      READ104,W0,W1,W2,W3,W4
      DO 43 I=1,9
      X(I)=0.E0
   43 CONTINUE
      DO 45 I=10,M
      CALL GAUSS(IX,B,C,W)
   45 X(I)=W
      ERR=0.E0
      DO 10 I=10,M
      N=I-9
      W0=W0+MU*ERR*X(I-1)
      W1=W1+MU*ERR*X(I-2)
      W2=W2+MU*ERR*X(I-3)
      W3=W3+MU*ERR*X(I-4)
      W4=W4+MU*ERR*X(I-5)
      Y=W0*X(I)+W1*X(I-1)+W2*X(I-2)+W3*X(I-3)+W4*X(I-4)
      D=.5*(X(I)+X(I-1)+X(I-2)+X(I-3))
      ERR=D-Y
      ERR2=ERR*ERR
   10 PRINT102,N,X(I),D,Y,ERR2,W0,W1,W2,W3,W4
      STOP
      END
```

2. FIVE WEIGHT FIR LEAST SQUARES ALGORITHM

```
      DIMENSION X(500),A(8),B(8),D(8),F(8),C(8,8),E(8,8),Y(500)
  102 FORMAT(' ',I6,3F12.4,E14.4,6F12.4)
  104 FORMAT(I9,I4,2F7.4,I4)
      READ104,IX,MM,BB,CC,NN
      READ,((C(I,J),J=1,NN),I=1,NN)
      DO 10 I=1,NN
   10 A(I)=0.E0
      DO 43 I=1,9
      X(I)=0.E0
   43 CONTINUE
      DO 45 I=10,MM
      CALL GAUSS(IX,BB,CC,W)
   45 X(I)=W
      DO 99 I=10,MM
      NX=I-9
      S=A(1)*X(I)+A(2)*X(I-1)+A(3)*X(I-2)+A(4)*X(I-3)+A(5)*X(I-4)
      Z=.5*(X(I)+X(I-1)+X(I-2)+X(I-3))
      ERR=Z-S
      ERR2=ERR*ERR
      IF(ERR2.LT.1.E-13) GO TO 47
      GAMA=0.E0
      DO 15 N=1,NN
      F(N)=0.E0
      B(N)=0.E0
   15 D(N)=0.E0
      Y(I)=X(I)
      Y(I-1)=X(I-1)
      Y(I-2)=X(I-2)
      Y(I-4)=X(I-4)
      Y(I-3)=X(I-3)
      DO 60 N=1,NN
      DO 50 M=1,NN
   50 F(N)=F(N)+C(M,N)*Y(I+1-M)*Y(I+1-N)
   60 GAMA=GAMA+F(N)
      DO 30 K=1,NN
      DO 20 J=1,NN
      A(K)=A(K)+ERR*C(K,J)*Y(I+1-J)/GAMA
      B(K)=B(K)+C(K,J)*Y(I+1-J)
   20 D(K)=D(K)+C(J,K)*Y(I+1-J)
   30 CONTINUE
      DO 46 K=1,NN
      DO 55 J=1,NN
      E(K,J)=B(K)*D(J)
   55 C(K,J)=C(K,J)-E(K,J)/GAMA
   46 CONTINUE
   47 CONTINUE
   99 PRINT102,NX,X(I),Z,S,ERR2,A(1),A(2),A(3),A(4),A(5)
      STOP
      END
```

3. STEARNS' ALGORITHM

```
      DIMENSION X(300),Y(300),ALFA1(300),ALFA2)300),BETA0(300),
     *BETA1(300),BETA2(300)
      REAL MU1,MU2
  101 FORMAT(5F5.3)
  102 FORMAT(' ',I6,3F12.4,E14.4,5F12.4)
  104 FORMAT(2F8.4,I9,I4,2F7.4)
      READ104,MU1,MU2,IX,M,B,C
      READ101,A0,A1,A2,B1,B2
      DO 46 I=1,9
      X(I)=0.E0
      Y(I)=0.E0
      BETA0(I)=0.E0
      BETA1(I)=0.E0
      BETA2(I)=0.E0
      ALFA1(I)=0.E0
   46 ALFA2(I)=0.E0
      DO 44 I=10,M
      CALL GAUSS(IX,B,C,W)
   44 X(I)=W
      ERR=0.E0
      DO 10 I=10,M
      N=I-9
      Y(I)=A0*X(I)+A1*X(I-1)+A2*X(I-2)+B1*Y(I-1)+B2*Y(I-2)
      D=X(I)-.49*X(I-2)+.5*Y(I-1)+.06*Y(I-2)
      ERR=D-Y(I)
      ERR2=ERR*ERR
      ALFA1(I)=Y(I-1)+B1*ALFA1(I-1)+B2*ALFA1(I-2)
      ALFA2(I)=Y(I-2)+B1*ALFA2(I-1)+B2*ALFA2(I-2)
      BETA0(I)=X(I)+B1*BETA0(I-1)+B2*BETA0(I-2)
      BETA1(I)=X(I-1)+B1*BETA1(I-1)+B2*BETA1(I-2)
      BETA2(I)=X(I-2)+B1*BETA2(I-1)+B2*BETA2(I-2)
      B1=B1+MU2*ERR*ALFA1(I)
      B2=B2+MU2*ERR*ALFA2(I)
      A0=A0+MU1*ERR*BETA0(I)
      A1=A1+MU1*ERR*BETA1(I)
      A2=A2+MU1*ERR*BETA2(I)
   10 PRINT102,N,X(I),D,Y(I),ERR2,A0,A1,A2,B1,B2
      STOP
      END
```

4. FIVE WEIGHT IIR ALGORITHM

```
      DIMENSION X(500),S(500)
      REAL K1,K2,K3,K4,K5
  101 FORMAT(5F10.4)
  102 FORMAT(' ',I6,3F12.4,E14.4,5F12.4)
  104 FORMAT(I9,I4,2F7.4)
      READ101,K1,K2,K3,K4,K5
      READ101,A0,A1,A2,B1,B2
      READ104,IX,M,B,C
      DO 44 I=10,M
      CALL GAUSS(IX,B,C,W)
   44 X(I)=W
      DO 42 I=1,9
      X(I)=0.E0
      S(I)=0.E0
   42 CONTINUE
      DO 10 I=10,M
      N=I-9
      S(I)=A0*X(I)+A1*X(I-1)+A2*X(I-2)+B1*S(I-1)+B2*S(I-2)
      D=X(I)-.49*X(I-2)+.5*S(I-1)+.06*S(I-2)
      ERR=D-S(I)
      ERR2=ERR*ERR
      C1=B1**2+B2
      C2=B1**3+2.*B1*B2
      C3=A1+2.*B1*A0
      C4=3.*B1**2*A0+2.*B2*A0+2.*B1*A1+A2
      C5=B1**4+3.*B1**2*B2+B2**2
      C6=4.*B1**3*A0+6.*B2*B1*A0+3.*B1**2*A1+2.*B2*A1+2.*B1*A2
      C7=A0
      A0=A0+K1*ERR*(X(I)+B1*X(I-1)+C1*X(I-2)+C2*X(I-3)+C5*X(I-4))
      A1=A1+K2*ERR*(X(I-1)+B1*X(I-2)+C1*X(I-3)+C2*X(I-4))
      A2=A2+K3*ERR*(X(I-2)+B1*X(I-3)+C1*X(I-4))
      B1=B1+K4*ERR*(C7*X(I-1)+C3*X(I-2)+C4*X(I-3)+C6*X(I-4))
      B2=B2+K5*ERR*(C7*X(I-2)+C3*X(I-3)+C4*X(I-4))
   10 PRINT102,N,X(I),D,S(I),ERR2,A0,A1,A2,B1,B2
      STOP
      END
```

5. FIVE WEIGHT IIR LEAST SQUARES ALGORITHM

```
      DIMENSION X(500),A(8),B(8),D(8),F(8),C(8,8),E(8,8),Y(500)
  102 FORMAT(' ',I6,3F12.4,E14.4,6F12.4)
  104 FORMAT(I9,I4,2F7.4,I4)
      READ104,IX,MM,BB,CC,NN
      READ,((C(I,J),J=1,NN),I=1,NN)
      DO 10 I=1,NN
   10 A(I)=0.E0
      DO 43 I=1,9
      X(I)=0.E0
   43 CONTINUE
      DO 45 I=10,MM
      CALL GAUSS(IX,BB,CC,W)
   45 X(I)=W
      S1=0.E0
      S=0.E0
      DO 99 I=10,MM
      NX=I-9
      S2=S1
      S1=S
      S=A(1)*X(I)+A(2)*X(I-1)+A(3)*X(I-2)+A(4)*S1+A(5)*S2
      Z=X(I)-.49*X(I-2)+.5*S1+.06*S2
      ERR=Z-S
      ERR2=ERR*ERR
      IF(ERR2.LT.1.E-13) GO TO 47
      GAMA=0.E0
      DO 15 N=1,NN
      F(N)=0.E0
      B(N)=0.E0
   15 D(N)=0.E0
      Y(I)=X(I)
      Y(I-1)=X(I-1)
      Y(I-2)=X(I-2)
      Y(I-3)=S1
      Y(I-4)=S2
      DO 60 N=1,NN
      DO 50 M=1,NN
   50 F(N)=F(N)+C(M,N)*Y(I+1-M)*Y(I+1-N)
   60 GAMA=GAMA+F(N)
      DO 30 K=1,NN
      DO 20 J=1,NN
      A(K)=A(K)+ERR*C(K,J)*Y(I+1-J)/GAMA
      B(K)=B(K)+C(K,J)*Y(I+1-J)
   20 D(K)=D(K)+C(J,K)*Y(I+1-J)
   30 CONTINUE
      DO 46 K=1,NN
      DO 55 J=1,NN
      E(K,J)=B(K)*D(J)
   55 C(K,J)=C(K,J)-E(K,J)/GAMA
   46 CONTINUE
   47 CONTINUE
   99 PRINT102,NX,X(I),Z,S,ERR2,A(1),A(2),A(3),A(4),A(5)
      STOP
      END
```

APPENDIX B

EXPANSION OF AN IIR SYSTEM IN TERMS OF FIR SYSTEM PARAMETERS

The equation

$$s_k(\alpha) = \sum_{\ell=0}^{L} a_\ell \, x(k-\ell) + \sum_{j=1}^{J} b_j \, s_{k-j}(\alpha) \qquad \text{(B1)}$$

represents an IIR model for a system. This can be written in an FIR form as [11]

$$s_k(\alpha) = \sum_{n=0}^{\infty} c_n \, x(k-n) \; . \qquad \text{(B2)}$$

The expression for $\nabla_\alpha s_k(\alpha)$ formed from (B2) does not contain previous values of itself as does the $\nabla_\alpha s_k(\alpha)$ derived from (B1). To evaluate $s_k(\alpha)$ derived from (B2) previous values of $s_k(\alpha)$ are substituted into (B1). These previous values are derived from (B1) by replacing k with k-1, k-2, etc. Substituting for $s_{k-1}(\alpha)$ in (B1) yields

$$\begin{aligned} s_k(\alpha) = {} & a_0 x(k) + a_1 x(k-1) + \ldots + a_L x(k-L) \\ & + b_1 [a_0 x(k-1) + a_1 x(k-2) + \cdots + a_L x(k-L-1) \\ & + b_1 s_{k-2}(\alpha) + b_2 s_{k-3}(\alpha) + \cdots + b_J s_{k-J-1}(\alpha)] \\ & + b_2 s_{k-2}(\alpha) + b_3 s_{k-3}(\alpha) + \cdots + b_J s_{k-J}(\alpha) \; . \end{aligned} \qquad \text{(B3)}$$

Continued substitution for $s_{k-2}(\alpha)$, $s_{k-3}(\alpha)$, etc. yields an expression that contains terms like $s_{k-J-m}(\alpha)$ where $m \geq 1$. Assuming the system has not been functioning for all time a point will be reached where all terms containing a previous

value of the output will be identically zero and the present output will be expressed in terms of the inputs and the system parameters as in (B2).

For the case L=J=2 in (B1)

$$s_k(\alpha) = a_0x(k) + a_1x(k-1) + a_2x(k-2) + b_1s_{k-1}(\alpha) + b_2s_{k-2}(\alpha) \tag{B4}$$

substituion for first $s_{k-1}(\alpha)$ and then $s_{k-2}(\alpha)$ yields

$$\begin{aligned} s_k(\alpha) = {} & a_0x(k) + a_1x(k-1) + a_2x(k-2) \\ & + b_1\, a_0x(k-1) + a_1x(k-2) + a_2x(k-3) \\ & + b_1[a_0x(k-2) + a_1x(k-3) + a_2x(k-4)] \\ & + b_2s_{k-3}(\alpha) + b_2[a_0x(k-2) + a_1x(k-3) \\ & + a_2x(k-4) + b_1s_{k-3}(\alpha) + b_2s_{k-4}(\alpha)] \ . \end{aligned} \tag{B5}$$

After substituting for $s_{k-3}(\alpha)$ so that all terms containing x(k), x(k-1), x(k-2) and x(k-3) will be contained in the expression for $s_k(\alpha)$ (B5) becomes

$$\begin{aligned} s_k(\alpha) = {} & a_0x(k) + [a_1 + b_1a_0]x(k-1) \\ & + [a_2 + b_1a_1 + b_1^2a_0 + b_2a_0\ ]x(k-2) \\ & + [b_1a_2 + b_1^2a_1 + 2b_2a_1 + b_1b_2a_0 + b_2b_1a_0]x(k-3) \\ & + R \end{aligned} \tag{B6}$$

where R is the additional terms not involving x(k), x(k-1),

x(k-2) and x(k-3). Inspection of (B6) and (B2) immediately yields

$$c_0 = a_0$$

$$c_1 = a_1 + b_1 a_0$$

$$c_2 = a_2 + b_1 a_1 + b_2 a_0 + b_1^2 a_0$$

$$c_3 = b_1 a_2 + b_1^2 a_1 + 2b_2 a_1 + 2b_2 b_1 a_0 \ . \qquad \text{(B7)}$$

Additional c_i's can be generated by defining a permutation function $P_i(b)$ such that:

$P_i(b)$ = A sum of products of the form $b_m b_n \ldots$ where $m+n+\ldots=i$. The sum is to include all permutations of the product. $P_0(b) = 1$ by definition.

For example let J=2 in (B1). The product permutations possible are:

b_1	$b_2 b_1$	$b_1 b_2 b_1$	$b_2 b_2 b_2$
b_2	$b_2 b_2$	$b_2 b_1 b_1$	etc.
$b_1 b_1$	$b_1 b_1 b_1$	$b_2 b_1 b_2$	
$b_1 b_2$	$b_1 b_1 b_2$	$b_2 b_2 b_1$	

This gives

$$P_0(b) = 1$$

$$P_1(b) = b_1$$

$$P_2(b) = b_2 + b_1 b_1$$

$$P_3(b) = b_1b_2 + b_2b_1 + b_1b_1b_1 = b_1^3 + 2b_1b_2$$

$$P_4(b) = b_2^2 + 3b_2b_1^2 + b_1^4$$

$$\vdots$$

Using this notation gives

$$s_k(\alpha) = P_0\, a_0\, x(k) + [P_0\, a_1 + P_1\, a_0]x(k-1) + [P_0\, a_2 + P_1\, a_1 + P_2\, a_0]x(k-2) + \cdots \tag{B8}$$

or

$$s_k(\alpha) = \sum_{j=0}^{\infty} \Big[\sum_{\substack{m,n \\ m+n=j}} P_m(b)\, a_n \Big] x(k-j) \tag{B9}$$

To form $\nabla_\alpha s_k(\alpha)$ the derivatives of $s_k(\alpha)$ with respect to α must be formed. Using (B9)

$$\frac{\partial s_k(\alpha)}{\partial a_0} = P_0x(k) + P_1x(k-1) + P_2x(k-2) + \cdots$$

$$\frac{\partial s_k(\alpha)}{\partial a_n} = \sum_{j=n}^{\infty} P_{j-n}\, x(k-j)$$

$$\frac{\partial s_k(\alpha)}{\partial b_i} = \sum_{j=0}^{\infty} \Big[\sum_{\substack{m,n \\ m+n=j}} \frac{\partial P_m(b)}{\partial b_i}\, a_n \Big]\, x(k-j)\ . \tag{B10}$$

Computation of the first few terms of the summation for the last equation in (B10) for the case J=2 yields

$$\frac{\partial P_0(b)}{\partial b_1} = 0 \qquad \frac{\partial P_0(b)}{\partial b_2} = 0$$

$$\frac{\partial P_1(b)}{\partial b_1} = 1 \qquad \frac{\partial P_1(b)}{\partial b_2} = 0$$

$$\frac{\partial P_2(b)}{\partial b_1} = 2b_1 \qquad \frac{\partial P_2(b)}{\partial b_2} = 1$$

$$\frac{\partial P_3(b)}{\partial b_1} = 3b_1^2 + 2b_2 \qquad \frac{\partial P_3(b)}{\partial b_2} = 2b_1$$

$$\frac{\partial P_4(b)}{\partial b_1} = 4b_1^3 + 6b_2b_1 \qquad \frac{\partial P_4(b)}{\partial b_2} = 3b_1^2 + 2b_2$$

$$\vdots \qquad\qquad \vdots$$

The complete computation of $\nabla_\alpha s_k(\alpha)$ involves infinte sums, therefore truncation is required in practical applications. For L=J=2 in (B1) and using the first five terms of the infinite sums gives the following expression for $\nabla_\alpha s_k(\alpha)$.

$$\nabla_\alpha s_k(\alpha) = \begin{bmatrix} 1 & d_0 & d_1 & d_2 & d_3 \\ 0 & 1 & d_0 & d_1 & d_2 \\ 0 & 0 & 1 & d_0 & d_1 \\ 0 & d_4 & d_5 & d_6 & d_7 \\ 0 & 0 & d_4 & d_5 & d_6 \end{bmatrix} [X_k] \qquad \text{(B11)}$$

where

$$d_0 = b_1$$

$$d_1 = b_1^2 + b_2$$

$$d_2 = b_1^3 + 2b_1b_2$$

$$d_3 = b_1^4 + 3b_1^2b_2 + b_2^2$$

$$d_4 = a_0$$

$$d_5 = a_1 + 2b_1a_0$$

$$d_6 = a_2 + 2b_1a_1 + 2b_2a_0 + 3b_1^2a_0$$

$$d_7 = 2b_1a_2 + 2b_2a_1 + 3b_1^2a_1 + 6b_2b_1a_0 + 4b_1^3a_0 .$$

REFERENCES

[1] B. Widrow, et al., "Adaptive Noise Cancelling: Principles and Applications," Proc. IEEE, Vol. 63, pp. 1692-1716, Dec. 1975.

[2] B. Widrow, et al., "Stationary and Non-stationary Learning Characteristics of the LMS Adaptive Filter," Proc. IEEE, Vol. 64, pp. 1151-1162, Aug. 1976.

[3] B. Widrow, "Adaptive Filters," Aspects of Network and System Theory, ed. R. E. Kalman and N. DeClaris. New York: Holt, Rinehard and Winston, Inc., 1971.

[4] G. R. Elliott, W. K. Jacklin, and S. D. Stearns, "The Adaptive Digital Filter," Tech. Rept. No. SAND 76-0360, August 1976, Sandia Laboratories, Albuquerque, New Mexico.

[5] P. L. Feintuch, "An Adaptive Recursive LMS Filter," Proc. IEEE, (Corresp.), to appear.

[6] P. Swerling, "Modern State Estimation Methods from the Viewpoint of the Method of Least Squares," IEEE Trans. Automat. Contr., Vol. AC-16, pp. 707-719, Dec. 1971.

[7] R. B. Blackman, "Methods of Orbit Refinement," The Bell System Technical Journal, Vol. XLIII, pp. 885-909, May 1964.

[8] D. Graupe, Identification of Systems. New York: Van Nostrand Reinhold Co., 1972, Chap. 6.

[9] A. P. Sage, Optimum Systems Control. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1968, Chap. 10.

[10] R. Deutsch, Estimation Theory. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1965, Chap. 8.

[11] S. D. Stearns, Digital Data Analysis. Rochelle Park, New Jersey: Hayden Book Co., Inc., 1975, Chap. 9.

ACKNOWLEDGEMENTS

I would like to express my utmost gratitude to Dr. Donald R. Hummels for the guidance and assistance he has given in the preperation of this thesis. My appreciation is given to Drs. Nasir Ahmed and Kendall F. Casey for serving on my committee.

The financial support of the Electrical Engineering Department of Kansas State University has made possible these studies and is appreciated.

The continuing encouragement received from my wife, Dorothy, has made this thesis a reality.

APPLICATION OF THE METHOD OF LEAST SQUARES TO ADAPTIVE SYSTEMS IDENTIFICATION

by

DAVID LYNN SOLDAN

B. S., Kansas State University, 1969

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1976

The method of least squares is applied to derive an algorithm for modeling an unknown system with an adaptive digital system. Both finite impulse response (FIR) and infinite impulse response (IIR) models are considered. A new suboptimum algorithm is also obtained for the IIR model. In comparisons with previously published algorithms, the least squares algorithm is found to converge faster. This characteristic enables the adaptive system to track changes in the unknown system with increased speed.