

2

AD A 681 605

NAVAL POSTGRADUATE SCHOOL Monterey, California

LEVEL



DTIC
ELECTE
MAR 11 1980
D
C

THESIS

EFFICIENT ADAPTIVE FIR AND IIR FILTERS

by

Uzi Ben-Yakov

December 1979

Thesis Advisor:

S. R. Parker

Approved for public release; distribution unlimited

80 3 10 018

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) <u>EFFICIENT ADAPTIVE FIR AND IIR FILTERS</u>		5. TYPE OF REPORT & PERIOD COVERED Ph.D. Dissertation; December 1979
7. AUTHOR(s) 10) Uzi Ben-Yakov		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) 2786		12. REPORT DATE Dec 79
		13. NUMBER OF PAGES 147
		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) Adaptive Filters, FIR, IIR, Recursive, Random Search, Digital Filters, ALE, Performance Surface, LMS, Discrete Time.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Self adaptive filters adjust their parameters to perform an almost optimal filtering operation without apriori knowledge of the input signal statistics. Two approaches to the design of efficient self adaptive discrete filtering algorithms are considered. For non-recursive (FIR) adaptive filters, simplified estimations of the gradient of the performance function to be minimized are considered. These algorithms result in reduced complexity of.		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE/When Data Entered.

implementation, improved dynamic operating range with about the same misadjustment errors and convergence time as the classic LMS (Least Means Squared) algorithm. An analysis of the simplified gradient approach is presented and confirmed experimentally for the specific example of an adaptive line enhancer (ALE). The results are used to compare the simplified gradient approaches with each other and the LMS algorithm. This comparison is done using a new graphic presentation of adaptive filter operating characteristics and a complexity index. This comparison indicates that the simplified gradient estimators are superior to the LMS algorithm for filters of equal complexity.

For recursive (IIR) adaptive filters a combined random and gradient search (RGS) algorithm is proposed, analyzed and tested. Since for the IIR filter, the performance surface is multimodal in the feedback parameters and unimodal in the feedforward parameters, random search is used to adjust the feedback parameters and gradient search to adjust the feedforward parameters. Convergence to the globally optimal filter parameters is guaranteed for sufficiently long adaptation time. Convergence time estimation for the RGS algorithm is derived and supported by simulation results for the ALE example. Finally, apriori knowledge of the optimal filter structure is taken into account in the formulation of an improved version of the basic RGS algorithm. This improvement is confirmed with the ALE example.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
AVAILABILITY STATEMENTS	
Dist	Avail and/or special
A	

DD Form 1473
1 Jan 73
S/N 0102-014-6601

UNCLASSIFIED

2 SECURITY CLASSIFICATION OF THIS PAGE/When Data Entered

Approved for public release; distribution unlimited

EFFICIENT ADAPTIVE FIR AND IIR FILTERS

by

Uzi Ben-Yakov
Lieutenant Commander, Israeli Navy
B.S., Technion, Israel Institute of Technology, 1967

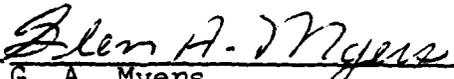
Submitted in partial fulfillment of the
requirements for the degree of

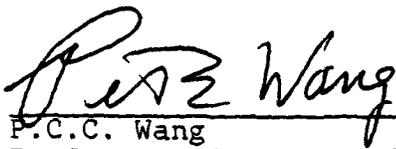
DOCTOR OF PHILOSOPHY

from the
NAVAL POSTGRADUATE SCHOOL
December 1979

Author 

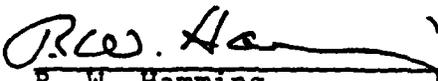
Approved by:


G. A. Myers
Associate Professor of
Electrical Engineering


P.C.C. Wang
Professor of Mathematics
and National Security
Affairs

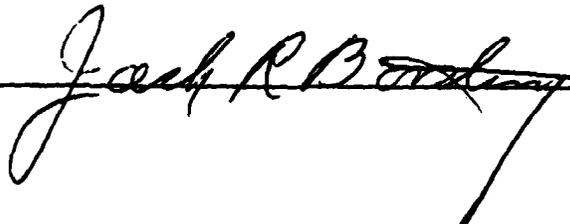

R. Panholzer
Professor of Electrical
Engineering


S. R. Parker
Professor of Electrical
Engineering
Dissertation Supervisor


R. W. Hamming
Professor of Computer
Science

Approved by 

Chairman, Department of Electrical Engineering

Approved by 

Academic Dean

ABSTRACT

Self adaptive filters adjust their parameters to perform an almost optimal filtering operation without apriori knowledge of the input signal statistics. Two approaches to the design of efficient self adaptive discrete filtering algorithms are considered.

For non-recursive (FIR) adaptive filters, simplified estimations of the gradient of the performance function to be minimized are considered. These algorithms result in reduced complexity of implementation, improved dynamic operating range with about the same misadjustment errors and convergence time as the classic LMS (Least Means Squared) algorithm. An analysis of the simplified gradient approach is presented and confirmed experimentally for the specific example of an adaptive line enhancer (ALE). The results are used to compare the simplified gradient approaches with each other and the LMS algorithm. This comparison is done using a new graphic presentation of adaptive filter operating characteristics and a complexity index. This comparison indicates that the simplified gradient estimators are superior to the LMS algorithm for filters of equal complexity.

For recursive (IIR) adaptive filters a combined random and gradient search (RGS) algorithm is proposed, analyzed and tested. Since for the IIR filter, the performance surface is multimodal in the feedback parameters and unimodal in the feedforward parameters, random search is used to adjust the

feedback parameters and gradient search to adjust the feed-forward parameters. Convergence to the globally optimal filter parameters is guaranteed for sufficiently long adaptation time. Convergence time estimation for the RGS algorithm is derived and supported by simulation results for the ALE example. Finally, apriori knowledge of the optimal filter structure is taken into account in the formulation of an improved version of the basic RGS algorithm. This improvement is confirmed with the ALE example.

TABLE OF CONTENTS

	Page No.
I. INTRODUCTION	9
1.1 BACKGROUND	9
1.2 FIR ADAPTIVE FILTERS	14
1.3 THE IIR ADAPTIVE FILTER	22
1.4 INTRODUCTION TO ADAPTIVE FIR FILTERS USING SIMPLIFIED GRADIENT ESTIMATIONS	27
1.5 INTRODUCTION TO ADAPTIVE IIR FILTERS USING RANDOM SEARCH TECHNIQUES	32
II. ADAPTIVE FIR FILTERS USING SIMPLIFIED GRADIENT ESTIMATIONS	35
2.1 SIMPLIFIED GRADIENT ALGORITHMS	35
2.2 CONVERGENCE AND STABILITY	40
2.3 CONVERGENCE TIME ESTIMATION	46
2.4 STEADY-STATE ERROR AND MISADJUSTMENT	54
2.5 DESIGN CONSIDERATIONS	64
2.6 CONCLUSION	71
III. RANDOM SEARCH IIR ADAPTIVE FILTERS	74
3.1 IIR PERFORMANCE SURFACE	74
3.2 THE RANDOM SEARCH CONCEPT	78
3.3 OPERATION OF THE RANDOM SEARCH ALGORITHM	85
3.4 RANDOM AND GRADIENT SEARCH (RGS)	89
3.5 CONVERGENCE OF THE RGS IIR FILTER	95
3.6 A-PRIORI STRUCTURED ADAPTIVE FILTERS (ASAF)	120
3.7 CONCLUSION	125
IV. SUMMARY	127

	Page No.
APPENDIX A - SIMULATION	130
A.1 - THE SIMULATION METHOD	130
A.2 - FIR SIMULATION PROGRAM	133
A.3 - IIR SIMULATION PROGRAM	136
APPENDIX B - RANDOM AND GRADIENT SEARCH SUBROUTINE	140
REFERENCES	142

ACKNOWLEDGEMENT

I would like to thank all those who supported and helped me in this work.

I wish to express here a special thanks to my wife, Rachel, and my daughters, Adi and Tali, for their love and understanding.

I would like to thank Professor Sydney R. Parker for his efforts and the many discussions which led to this dissertation.

I would like to thank the members of my doctoral committee, Professor R. W. Hamming, Professor G. A. Myers, Professor R. Panholzer and Professor P. C. C. Wang for their help and cooperation.

I wish to thank the Naval Postgraduate School computer lab staff for their helpful advice and cooperation in generating the simulation results.

Last but not least I would like to thank Ms. Elaine Christian for transforming a messy draft to a neatly typed report.

ACKNOWLEDGEMENT

I would like to thank all those who supported and helped me in this work.

I wish to express here a special thanks to my wife, Rachel, and my daughters, Adi and Tali, for their love and understanding.

I would like to thank Professor Sydney R. Parker for his efforts and the many discussions which led to this dissertation.

I would like to thank the members of my doctoral committee, Professor R. W. Hamming, Professor G. A. Myers, Professor R. Panholzer and Professor P. C. C. Wang for their help and cooperation.

I wish to thank the Naval Postgraduate School computer lab staff for their helpful advice and cooperation in generating the simulation results.

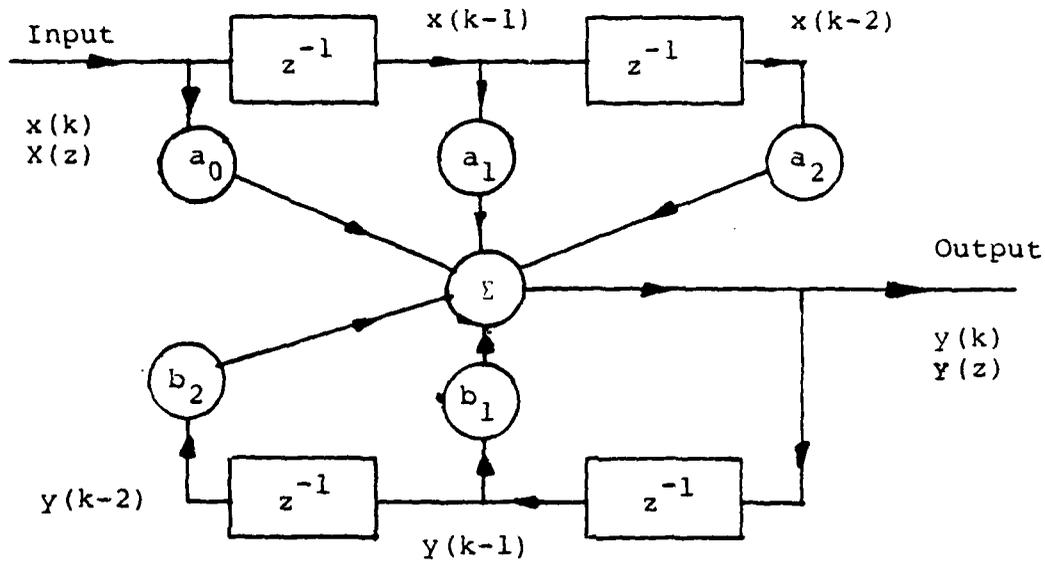
Last but not least I would like to thank Ms. Elaine Christian for transforming a messy draft to a neatly typed report.

I. INTRODUCTION

1.1. BACKGROUND

In a broad sense the term filter implies an operation on an input signal or collection of data in order to smooth, predict, or estimate a desired property hidden in the input. Fig. 1.1-1 presents the block diagram of a discrete time linear recursive digital filter. An optimal filter is one designed to be optimum or best with respect to a performance criterion that measures or expresses its effectiveness. The most commonly used approach to optimal filter design is the linear filter optimized with respect to a Minimum Mean Squared Error (MMSE), where the error is defined as the difference between the filter output and a desired signal.

This optimal filter is usually called the Wiener filter. Filter realization may be for: (a) analog signals and continuous time, (b) analog signals and discrete time, (c) digital signals and discrete time. This dissertation is applicable to cases (b) and (c). A basic discussion of discrete Wiener filters is presented by Nahi [28, Ch 5]. As expected the parameters of the optimal filter depend upon properties of the input and desired signals. For example, the Wiener filter solution depends upon the second order statistics of the input signal and the desired signal.



- z^{-1} - unit delay
 a - multiplier, a
 Σ - summing junction

Filter operation:

$$\begin{aligned}
 \text{time domain: } y(k) &= a_0 x(k) + a_1 x(k-1) + a_2 x(k-2) \\
 &\quad + b_1 y(k-1) + b_2 y(k-2) \\
 &= \sum_{i=0}^2 a_i x(k-i) + \sum_{i=1}^2 b_i y(k-i)
 \end{aligned}$$

z - transform domain:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

Fig. 1.1-1
Discrete Linear Filter

The performance surface describes the filter performance criterion as a function of its weights (parameters, coefficients- a_i , b_i , of Fig. 1.1-1). Each point of the surface is the value of the performance criterion with specific weights of the filter. The term performance function will be used to describe the performance criterion values as function of time during the adaptation process. The optimal filter weights are those at the global minimum point of the performance surface.

In those cases where the information (input statistics) needed to design an optimal filter is not available, or in those cases where the filter is required to operate under statistically nonstationary input signal conditions, the usual optimal design approach is not applicable. In some of these cases, a self adaptive filter can be used to overcome this lack of information. The adaptive filter tries to adjust its parameters dynamically to variations in the statistics of the input signal. For the weight adjustment, or adaptation, the adaptive filter uses an error signal. Ideally this error is the difference between the filter output and a desired signal. In many applications the desired signal is not available per se, so that a reference signal, related to the desired signal in some way, is used to develop the error signal. Fig. 1.1-2 presents a block diagram of an adaptive filter with its input, output and reference signals. The adaptive filter thus includes a signal processing section which is similar to a non-adaptive filter, except that the

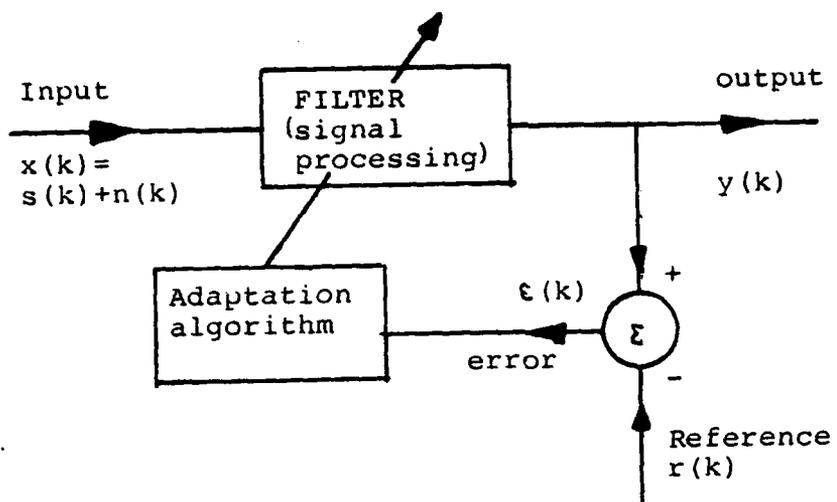


Fig. 1.1-2
Adaptive Filter

filter weights are adjustable and controlled by the second portion of the adaptive filter-namely the weight adaptation algorithm. The weight optimization algorithm typically estimates the gradient of the performance surface and adjusts the weights in the direction of steepest descent. For a statistically stationary situation, after some transient, the adaptive filter can be expected to reach a steady-state condition at which the parameters jitter around the minimum point of the performance surface.

The generation of the reference signal is a key consideration in adaptive filter implementation. There are various practical methods as discussed in [1, 2, 3, 7, 22, 24, 26, 29, 32, 37, 38, 39]. In many of these applications the reference signal is not identical to the signal we would like to have as output of the filter because if we had the desired output we wouldn't need the filter. In spite of the approximations involved, the adaptive filter is still able to operate and optimize the weights in many practical applications.

This dissertation investigates two approaches to efficient adaptive filters. Chapter II discusses simplified gradient estimation methods for non-recursive filters and Chapter III discusses recursive filters based on a combined random and gradient search adaptation technique.

1.2 FIR* ADAPTIVE FILTERS

The FIR filter is the simplest form of digital filter. The processing operation produces an output which is the linear sum of weighted delayed input samples. The impulse response of this filter is given by the sequence of values of the filter weights. Because of its relative simplicity, the FIR adaptive filter historically was the starting point for the development of adaptive filters.

A very important property of this filter is that its performance surface is quadratic so we have one and only one minimum, i.e. it is a unimodal surface as shown by Widrow [1]. For a unimodal surface, a gradient minimum seeking algorithm will converge to the minimum (a formal proof is presented in [1]), and this property is the key to the success of the Least Mean Squared (LMS) algorithm, discussed later. Interest in the area of adaptive filtering started in the late 50's and early 60's. The most successful approach is Widrow's LMS algorithm. Widrow in [1] presents the classic LMS algorithm and summarizes most of the previous work on the subject. The LMS algorithm and its basic properties are presented later. In [3] Widrow et al introduces the concept of noise cancelling which uses a reference signal that is related only to the noise to estimate the noise portion of the input. The

* FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) are generally used by the signal processing community to denote non-recursive and recursive filters respectively and are so used in this work. It is noted though, that some recursive filters can have a finite impulse response.

output is produced by subtracting the noise estimate from the input signal. In [4] Widrow et al extended the analysis to non-stationary operation of the LMS algorithm. In this study they identify two sources for misadjustment (a measure of the distance of the actual steady-state error from the optimal steady-state error) with nonstationary input signal. The first is due to gradient estimation errors (or gradient noise) which also exists with stationary inputs. The second cause of misadjustment with a non-stationary input is due to the changing statistics, and results in a lag in updating the filter weights after the optimal solution. This analysis gives some insight to the problem and provides basic design information. In [5] Widrow and McCool present a random search FIR filter and compare it to the LMS algorithm. Using the unimodal property of the FIR filter they modify the random search algorithm so that high performance function value points (which in regular random search methods are discarded) contribute to convergence towards the optimum. Their conclusion is that the LMS is a better algorithm; it converges faster and produces less steady-state misadjustment. In [6] Widrow et al present versions of the LMS algorithm that operate on complex data. This concept has recently become important because of the use of adaptive techniques in the frequency domain, Dentino [16] and Zentner [17]. Lucky, [7], introduces a Minimum Magnitude performance criterion to derive an adaptive equalizer. Digital

communication systems use equalizers to reduce the inter-symbol interference in a communication channel. Lucky's solution involves transmission of a special training sequence which is known at the receiver and is used there as the reference signal. Another interesting point in his solution is the use of quantized variables in the adaptation algorithm.

Finally Frost [26], Owsley [29], Widrow et al [38], Griffiths and Jim [41] and many others discuss the use of the LMS algorithm for adaptive control of sensor beamforming arrays. We will not discuss these applications in this dissertation because of their specialized nature. However, it is noted that the simplified algorithms presented here are general and may be used to advantage in antenna arrays.

From the references the importance of the LMS algorithm is very clear. Surprisingly enough, very little was done to improve the basic algorithm, the emphasis being primarily on applications of the concept. Gersho [40] discusses adaptation in a quantized parameter space. Gersho's discussion is of a general nature, i.e. no specific performance criterion was assumed, and his main results is that for unimodal performance surfaces and deterministic gradient (i.e. no need for stochastic gradient estimation), the quantized algorithm will converge to the neighborhood of the optimal solution.

Noschner [27] is the only published attempt to derive computationally more efficient versions of the basic LMS weight adaptation algorithm, and these results have not

been used in practice. Griffiths and Jim in a recent paper [41] discuss a simplified adaptive system from another point of view. Their concern is to simplify the signal processing section in order to achieve high frequency operation. They propose a 3 level weight quantization, with no multiplications in the signal processing portion. The resulting weight adaptation scheme is based on the LMS algorithm, and it is necessary to store past quantizations. Hence it is more complicated, but the goal of high frequency operation is achieved.

Summary of LMS Algorithm

Because of its importance, the LMS adaptive algorithm is presented here following the basic references [1, 2, 3, 4, 5]. The basic filter output is given by:

$$y(k) = \sum_{i=0}^{N_a-1} a_i(k)x(k-i) \quad (1.2-1)$$

where: k is the time index

N_a is the number of filter weights

$a_i(k)$ is the i th weight at time k

$x(k) = s(k) + n(k)$ is the input signal consisting of desired signal $s(k)$ and additive noise $n(k)$.

We want to minimize the performance function:

$$J(k) = E\{\epsilon_s^2(k)\} = E\{[y(k) - s(k)]^2\} \quad (1.2-2)$$

where: $\epsilon_s(k) = y(k) - s(k)$ is the error

In order to perform the adaptation algorithm we need the gradient of the performance surface:

$$\nabla_{a_i}(k) = \frac{\partial J(k)}{\partial a_i} \quad i = 0, 1, \dots, N_a - 1 \quad (1.2-3)$$

In practice we don't have $J(k)$ since $s(k)$ is not known nor do we have an ensemble of processes to perform the expectation operation of (1.2-2). Thus we must use an estimate of the performance function:

$$\hat{J}(k) = \epsilon_r^2(k) = \{y(k) - r(k)\}^2 \quad (1.2-4)$$

where $r(k)$ is a reference signal, not necessarily identical to $s(k)$.

The gradient estimate is given by:

$$\begin{aligned} \hat{\nabla}_{a_i}(k) &= \frac{\partial \hat{J}(k)}{\partial a_i} = \frac{\partial \epsilon_r^2(k)}{\partial a_i} = 2\epsilon_r(k) \frac{\partial \epsilon_r(k)}{\partial a_i} = \\ &= 2\epsilon_r(k) \frac{\partial y(k)}{\partial a_i} = 2\epsilon_r(k) x(k-i) \end{aligned} \quad (1.2-5)$$

Using the gradient estimate of (1.2-5), the LMS weights adaptation are given by:

$$a_i(k+1) = a_i(k) - \mu_a \hat{\nabla}_{a_i} = a_i(k) - 2\mu_a \epsilon_r(k)x(k-i)$$

$$i = 1, 2, \dots, N_a - 1 \quad (1.2-6)$$

where μ_a is the adaptation gain controlling the convergence and steady-state properties of the filter.

Reference [4] assumes a stationary input with uncorrelated samples and derives formulas for the stability region, convergence time, and misadjustment as follows.

Stable convergence of the adaptation algorithm is limited to values of μ_a given by:

$$0 < \mu_a < 1/[N_a R_{xx}(0)] \quad (1.2-7)$$

where $R_{xx}(m) = E\{x(k)x(k-m)\}$ is the autocorrelation function of the input. Equation (1.2-7) was derived using the mean of the gradient estimate. So, in practice, in order to be stable at all times we need

$$\mu_a \ll 1/[N_a R_{xx}(0)].$$

The approximate Mean Squared Error (MSE) convergence time constant is given by

$$\tau_{\text{MSE}} = 1/[4\mu_a R_{xx}(0)] \quad (1.2-8)$$

The misadjustment, M, is defined as the ratio of the excess Mean Squared Error (MSE), due to adaptive filter steady-state jitter around the optimal solution, to the minimum MSE:

$$M = \frac{J_{\text{steady-state}} - J_{\text{min}}}{J_{\text{min}}} = \mu_a N_a R_{xx}(0) \quad (1.2-9)$$

where

$$J_{\text{ss}} = J_{\text{steady-state}} = \lim_{k \rightarrow \infty} J(k)$$

$$J_{\text{min}} = J_{\text{ss}} \text{ [of the optimal filter]} = \text{Minimum MSE}$$

The misadjustment estimate (1.2-9) was derived for an ideal reference signal, $r(k) = s(k)$, and does not apply to cases of noisy reference.

The derivations in [1, 2, 3, 4, 5] are based upon the use of eigenvalue eigenvector analysis. To obtain practical estimation formulas the eigenvalues based equations are approximated by correlation functions. The analysis presented

in this dissertation makes the approximations at the start of the derivations and uses correlation functions throughout. The advantage of this approach is that it provides better insight into the nature of the approximations.

1.3 THE IIR* ADAPTIVE FILTER

An IIR filter uses previous output values to compute the present filter output:

$$y(k) = \sum_{i=0}^{N_a-1} a_i x(k-i) + \sum_{i=1}^{N_b} b_i y(k-i) \quad (1.3-1)$$

Because of the feedback in (1.3-1) the impulse response may be infinite and is designated IIR.

Because of inherent savings due to the use of previous calculated values (the existence of poles in the transfer function), the IIR filter is the most efficient filtering scheme for many applications.

Since it uses feedback, the IIR filter can be unstable. This presents a design problem for the conventional IIR filter, and a basic requirement for an IIR adaptation algorithm is to assure that the resulting filter is stable. A second disadvantage of the IIR adaptive filter is the multimodal nature of its performance surface as discussed in section 3.1 .

White [8] was the first to suggest the use of IIR structures for an adaptive filter. He indicates a possible use of several performance criteria and derives the gradient expression for the Minimum Mean Squared Error (MMSE) performance criterion. In [9], Stearns et al presents an all adaptive IIR

* FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) are generally used by the signal processing community to denote non-recursive and recursive filters respectively and are so used in this work. It is noted though, that some recursive filters can have a finite impulse response.

filter. Stearns' algorithm is rather complex, i.e. the number of operations (multiplications and additions) is proportional to $\alpha_1 N_a N_b + \alpha_2 N_b^2$, compared to the relative simplicity of the LMS where the number of operations is proportional to N_a . Stearns' algorithm is discussed later and its gradient estimation method is presented with details.

In [10] Feintuch presents a much simpler adaptive IIR filter which consists of two LMS adaptive sections, one controls the feedforward weights adaptation and the second controls the feedback weights adaptation. Feintuch's algorithm gradient estimation method is presented later on in this section. Feintuch's algorithm works in some cases but, as pointed out by several investigators [11, 12], the derivation has errors and the filter, at least in the examples presented in [11], does not converge to the optimal solution.

In [13] Parikh and Ahmed used the same examples presented in [11] to demonstrate the convergence properties of Stearns' algorithm. Reference [13] shows that Stearns' algorithm does converge to a minimum point, but with a multimodal performance surface the steady-state might be around a local minimum or the global minimum depending upon the starting point of the adaptation process. McMurray, [14], investigates the dependence of Feintuch's algorithm stability on the values of its adaptation gains. The region of stable operation turns out to be a triangle in the adaptation gains space. In [15] McMurray investigates the convergence time for Feintuch's algorithm IIR filtering of narrow band signals and compares operation in the

time and frequency domains. In both cases the convergence time is inversely proportional to the square root of the multiplication of four factors: feedforward adaptation gain, feedback adaptation gain, number of feedforward weights, and the number of feedback weights. An additional conclusion was that the convergence time is shorter for the time domain operation. Parker and Ko, [18], extend the adaptive IIR filter for image processing. In [35] Treichler, Larimore and Johnson modify Feintuch's algorithm by passing the error term through a FIR filter. This modification allows for convergence to a minimum (not necessarily global), and its use is limited by the information needed for the design of the error term filter. The existing IIR adaptive algorithms are based upon Stearns' and Feintuch's algorithms which are summarized briefly in the following.

In order to have a practical adaptation method we use a performance function estimate:

$$\hat{J}(k) = \epsilon_r^2(k) = \{y(k) - r(k)\}^2 \quad (1.3-2)$$

where $r(k)$ is the reference signal and $y(k)$ is given by (1.3-1) with weights $a_i(k)$ and $b_i(k)$ being a function of time. The gradient estimate is given now:

$$\hat{\nabla} a_i = \frac{\partial \hat{J}(k)}{\partial a_i} = 2\epsilon_r \frac{\partial \epsilon_r(k)}{\partial a_i} = 2\epsilon_r \frac{\partial y(k)}{\partial a_i} \quad (1.3-3)$$

the derivative $\frac{\partial y(k)}{\partial a_i}$ is not as simple as in (1.2-5) because of the feedback terms such as $b_j y(k-j)$ present in $y(k)$.

The final form of the gradient estimates of Stearns' algorithm are given by:

$$\hat{\nabla}_{a_i}(k) = 2\varepsilon_r(k) \alpha_i(k) \quad i = 0, 1, \dots, N_a - 1 \quad (1.3-4)$$

where:

$$\alpha_i(k) = x(k-i) + \sum_{j=1}^{N_b} b_j(k) \alpha_i(k-j) \quad (1.3-5)$$

and:

$$\hat{\nabla}_{b_i}(k) = 2\varepsilon_r(k) \beta_i(k) \quad i = 1, 2, \dots, N_b \quad (1.3-6)$$

where:

$$\beta_i(k) = y(k-i) + \sum_{j=1}^{N_b} b_j(k) \beta_i(k-j) \quad (1.3-7)$$

Equations (1.3-4, 5, 6, 7) are the gradient estimates of Stearns' algorithm. Feintuch's, [10], algorithm uses only the first terms in the expressions for α_i (1.3-5), and β_i (1.3-7) and the resulting gradient estimates are:

$$\hat{\nabla}_{a_i}(k) = 2\varepsilon_r(k) x(k-i) \quad (1.3-8)$$

$$\hat{\nabla}_{b_i}(k) = 2\varepsilon_r(k) y(k-i) \quad (1.3-9)$$

With both algorithms the weights adaptation is given by:

$$a_i(k+1) = a_i(k) - \mu_a \hat{v}_{a_i}(k) \quad i=0,1,\dots,N_a-1 \quad (1.3-10)$$

$$b_i(k+1) = b_i(k) - \mu_b \hat{v}_{b_i}(k) \quad i=1,2,\dots,N_b \quad (1.3-11)$$

where μ_a and μ_b are the feedforward and feedback adaptation gains. These adaptive IIR filtering schemes are not satisfactory solutions to the IIR filtering problem. Stearn's algorithm is not satisfactory because of the following reasons:

- The instability problem mentioned by Elliott, Jacklin and Stearns, [25]; this problem is discussed later.
- The algorithm does not assure convergence to the global minimum.
- It is a complicated algorithm.

The Feintuch algorithm is not satisfactory mainly because, in some cases it fails to converge to a minimum point, and in all cases does not assure convergence to the global minimum of the performance surface.

1.4 INTRODUCTION TO ADAPTIVE FIR FILTERS USING SIMPLIFIED GRADIENT ESTIMATIONS

The LMS algorithm is being used in many adaptive filtering applications [1:6, 16, 17, 22, 24, 26, 29, 32, 34, 37, 38, 39, 41], with satisfactory results. The possibility of using simplified algorithms, with hardware and time savings, has not received much attention. Gersho [40], and Moschner [27], and recently Griffiths and Jim [41] (which discusses a somewhat different problem of simplifying the signal processing portion with more complicated adaptation algorithm) appear to be the only publications in this area. All applications, except [41], seem to select the classical LMS algorithm and not a simplified version. A possible reason for this fact might be the lack of confidence in the performance of a simplified algorithm, compared with the many satisfactory results obtained with the use of the LMS algorithm. This dissertation will demonstrate analytically, and by extensive simulation, the advantages and savings associated with the use of the simplified algorithms. One natural simplified algorithm investigated here is the use of a positive or negative Fixed Step Correction (FSC) in the adaptation, instead of the LMS correction which is proportional to the value of the gradient. This gradient estimation is given by:

$$\hat{v}_{FSC}(i,k) = \text{Sgn}\{\hat{v}_{LMS}(i,k)\} = \text{Sgn}\{\varepsilon(k)\} \text{Sgn}\{x(k-i)\} \quad (1.4-1)$$

where:

$$\text{Sgn}[\cdot] = \begin{cases} 1 & \text{if } [\cdot] \geq 0 \\ -1 & \text{if } [\cdot] < 0 \end{cases}$$

The second algorithm investigated here is to use a modified FSC with the step size proportional to the magnitude of the error. This algorithm is called here the Simplified LMS (SLMS), Moschner [27] called this the clipped LMS. The SLMS has the following gradient estimate:

$$\hat{\nabla}_{\text{SLMS}}(i,k) = \epsilon_r(k) \text{Sgn}\{x(k-i)\} \quad (1.4-2)$$

Chapter II discusses these algorithms and presents an analysis and simulation of adaptive FIR filter operation using these algorithms.

The optimal Wiener filter depends upon the statistics of the input signal and the desired signal, the steady-state behavior of an adaptive filter depends upon the corresponding statistics. Since the desired signal is not available for the adaptive filter, and it uses a reference signal which is only related to the desired signal, it is obvious that the properties of this filter differ depending upon the application and manner in which the reference signal is provided. In Chapter II an adaptive FIR filter is used as an adaptive line enhancer (ALE) [3, 34, 37, 39] which is a typical signal processing application and utilizes a noisy reference.

Appendix A describes the simulation details.

The discussion in Chapter II includes for each algorithm the following topics:

- convergence and stability, section 2.2.
- convergence time (TC), section 2.3.
- steady-state misadjustment (M), section 2.4.
- implementation complexity, section 2.1.
- dynamic range, section 2.6.

Sections 2.3 and 2.4 include derivations of estimation formulas to the convergence time, TC, and misadjustment, M, of the FSC and SLMS algorithms.

The simulation experiment, described in Appendix A, shows good agreement to these misadjustment and convergence time formulas.

Fig. 1.4-1 presents a typical operation of the adaptive FIR filter with LMS, FSC, and the SLMS algorithms. This figure shows a typical weight, a_1 , and the Mean Squared Error (MSE), as a function of time for the three algorithms as noted. On each plot we have drawn the optimal value of the weight or the MSE, an ensemble average of 100 runs as well as the convergence of an individual filter (single run). In Fig. 1.4-1 all of the algorithms perform, on the average, about the same.

For more accurate comparison, a graphic presentation of adaptive filter properties is introduced in Section 2.1. This graphic presentation, the Adaptive Filter Operating Characteristic (AFOC) is used to compare equal degree and equal complexity filters with different algorithms.

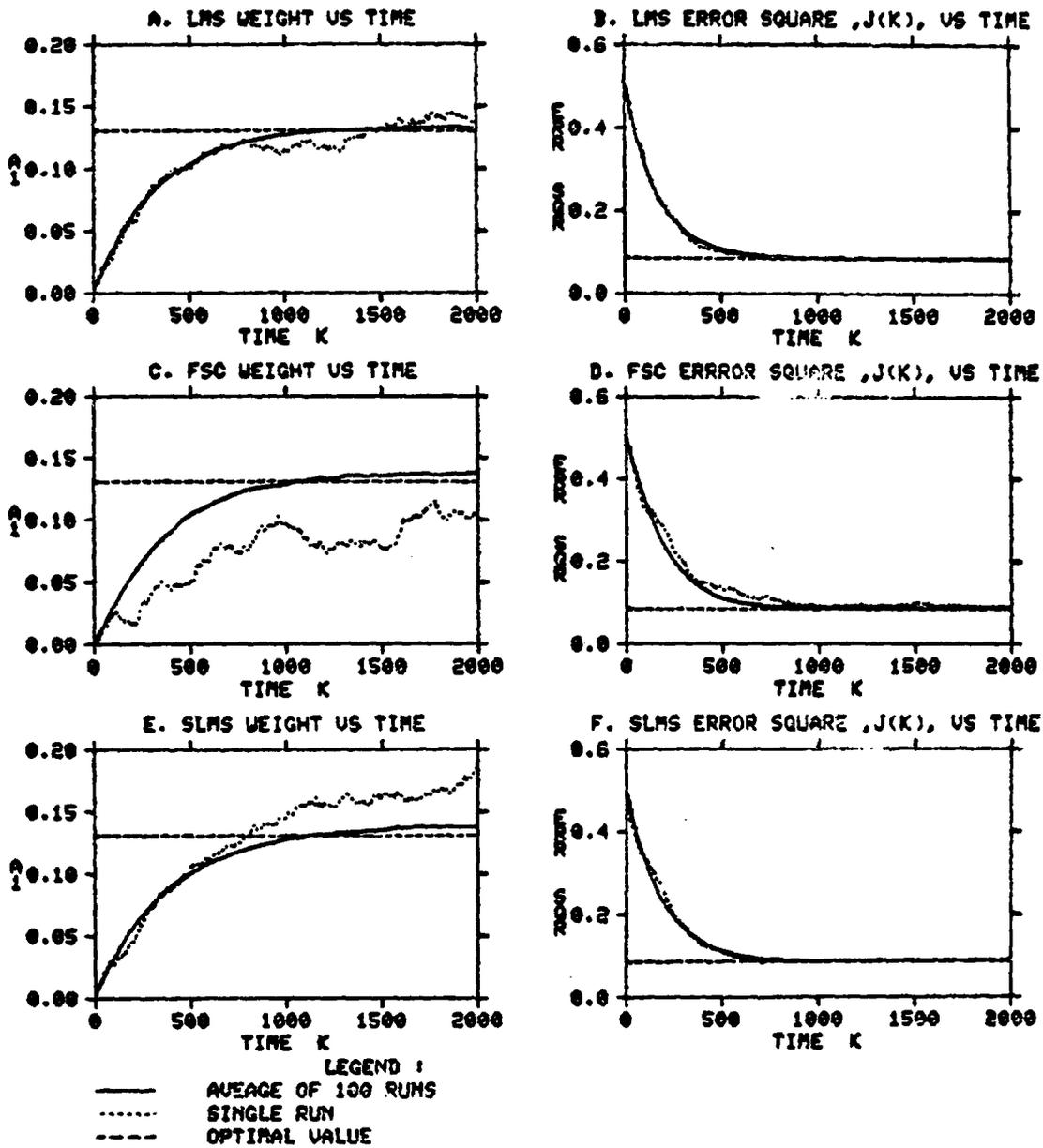


Fig. 1.4-1

Typical FIR adaptive filters operation with $N_a = 15$, $\mu_{LMS} = .0005$

$\mu_{FSC} = \mu_{SLMS} = .001$, ALE experiment

If one looks ahead to Fig. 2.1-1 it is apparent that the simplified gradient algorithms (FSC and SLMS), when compared to the LMS algorithm with equal complexity (cost) and equal convergence time, are more effective and provide more processing gain (processing gain is defined later as a measure of filter effectiveness).

1.5 INTRODUCTION TO ADAPTIVE IIR FILTERS USING RANDOM SEARCH TECHNIQUES

Adaptive IIR filters based on gradient methods have one major disadvantage which is the multimodal structure of the performance surface as discussed in section 3.1. Thus there is no inherent way to assure a steepest descent gradient convergence to the global minimum. The convergence problem and additional disadvantages of Stearns' and Feintuch's algorithms, as discussed in section 1.3, suggests that gradient methods may not be the best adaptation scheme for the IIR filter. Thus a different adaptation technique, namely, random search, is considered here. The basic concept of random search is discussed in section 3.2.

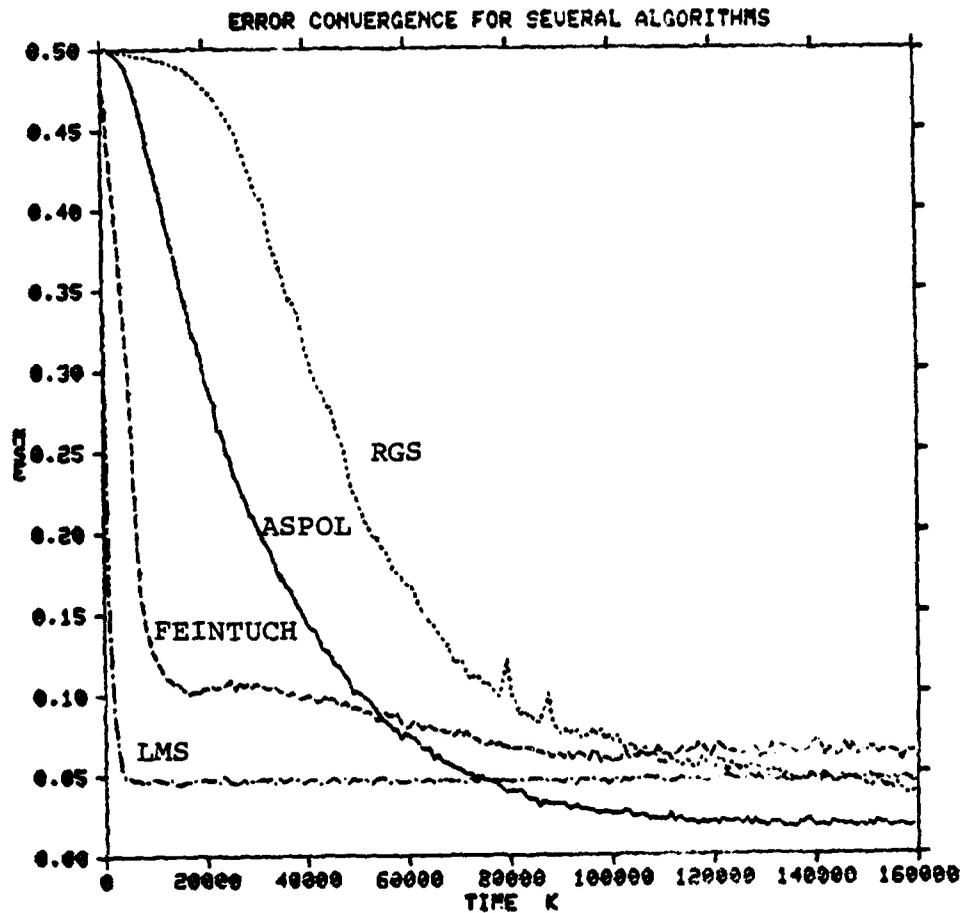
A random search IIR filter is presented and discussed in section 3.3. It is concluded there that this scheme is not satisfactory. The fact that the IIR filter's performance surface is quadratic in the feedforward weights (Elliott et al [25]) is the key for the hybrid Random and Gradient Search (RGS) algorithm developed in section 3.4. This new algorithm provides for satisfactory operation of an IIR adaptive filter. Convergence analysis of the RGS algorithm and convergence time estimation for a typical signal processing situations is given in section 3.5.

For cases where information is available on the structure of the optimal filter, a constrained, or apriori structured filter algorithm can be implemented. This concept is discussed in section 3.6 and shows good results. Fig. 1.5-1

presents the error convergence of four filters - the LMS FIR filter with 20 weights, a RGS IIR filter, an a priori structure adaptive pole (ASPOL, section 3.6) IIR filter, and Feintuch algorithm IIR filter. The IIR filters have two feedback and three feedforward weights. For this example it is seen that:

1. The LMS algorithm converges fastest.
2. The RGS converges slower but reaches a lower steady-state error.
3. The ASPOL converges to the lowest steady-state error, faster than the RGS.
4. Feintuch algorithm converges to the highest steady-state error.

These examples are typical.



LEGEND :

—	ASPOL	$\mu_a = .00001$	$\mu_b = .05$	$R = 500$	$\rho = .99$
.....	RGS	$\mu_a = .00001$	$\mu_b = .1$	$R = 500$	
- · - · -	FEINTUCH ALGORITHM	$\mu_a = .0001$	$\mu_b = .0075$		
- - - -	LMS	$\mu_a = .0001$	$N_A = 20$		

ρ is a parameter constant (pole magnitude)

R is the number of output samples used in estimating the performance surface value for a fixed set of parameters (random search interval).

Fig. 1.5-1

Error Convergence For Several Adaptive Filters

II. ADAPTIVE FIR FILTERS USING SIMPLIFIED GRADIENT ESTIMATIONS

2.1 TWO SIMPLIFIED GRADIENT ALGORITHMS

Two simplified gradient algorithms are considered:

(a) The Fixed Step Correction (FSC) adaptation scheme is given by:

$$a_i(k+1) = a_i(k) - \mu_a \text{Sgn}\{\hat{V}(i, k)\} \quad (2.1-1)$$

This formulation is essentially binary and was motivated by the general success of bang-bang type controllers. The adaptation gain μ_a is the size of the fixed correction step. We define the FSC gradient estimate as:

$$\hat{V}_{\text{FSC}}(i, k) = \text{Sgn}\{\hat{V}_{\text{LMS}}(i, k)\} = \text{Sgn}\{\epsilon(k)\} \text{Sgn}\{x(k-i)\} \quad (2.1-2)$$

It should be noted that the sign of the gradient, $\text{Sgn}\{\hat{V}(i, k)\} = \text{Sgn}\{x(k-i)\} \text{Sgn}\{\epsilon(k)\}$, is identical for both error magnitude and mean square error estimates, that is

$$\text{Sgn}\left\{\frac{\partial |\epsilon(k)|}{\partial a_i}\right\} = \text{Sgn}\left\{\frac{\partial \epsilon^2(k)}{\partial a_i}\right\}, \text{ so that (2.1-2) can be}$$

derived from either error magnitude or mean squared error.

Large correction steps result in fast convergence to the steady-

state near the optimal filter weights with a large steady-state jitter around the optimal filter. These contradicting effects call for engineering compromise in choosing the size of the correction step μ_a .

(b) The second approach is to use a variable size step. A natural possibility is to consider

$$\mu_a = \mu' |\epsilon(k)| \quad (2.1-3)$$

The combination of (1.2-2), (2.1-1), and (2.1-3) gives:

$$a_i(k+1) = a_i(k) - \mu' |\epsilon(k)| \text{Sgn} \{ \epsilon(k) \} \text{Sgn} \{ x(k-i) \} \quad (2.1-4)$$

We can use the regular adaptation gain symbol μ_a instead of μ' and write

$$a_i(k+1) = a_i(k) - \mu_a \epsilon(k) \text{Sgn} \{ x(k-i) \} \quad (2.1-5)$$

(2.1-5) is the simplified LMS (SLMS) algorithm with the gradient estimate given by:

$$\hat{v}_{\text{SLMS}}(i,k) = \epsilon(k) \text{Sgn} \{ x(k-i) \} \quad (2.1-6)$$

Typical operation of the LMS, FSC and SLMS algorithms are presented in Fig. 1.4-1.

A useful graphic presentation of adaptive filter properties is given by a plot of processing gain (PG) as a function of convergence time (TC). The processing gain measures the filter effectiveness and is defined as:

$$PG = 10 \log[R_{nn}(o)/J_{ss}] \quad (2.1-7)$$

where $R_{nn}(o)$ is the input noise power and J_{ss} , defined in (1.2-9), is the output error power.

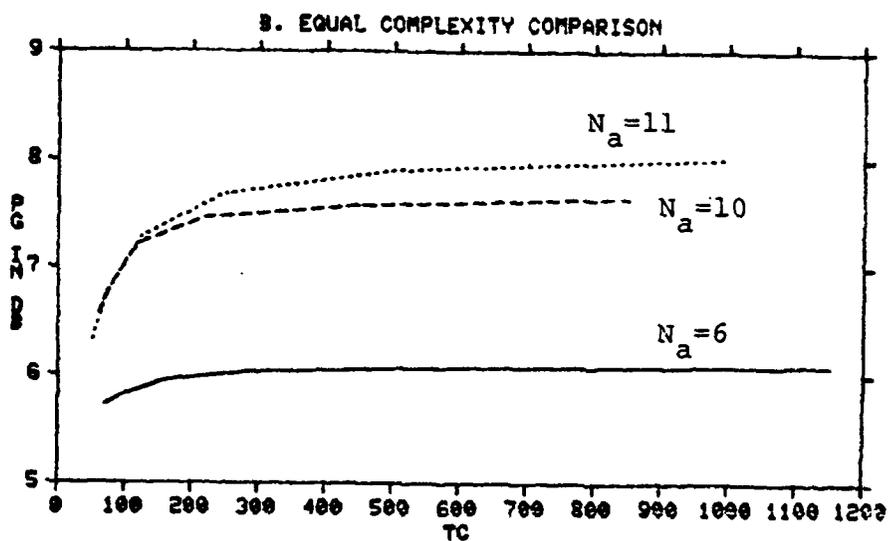
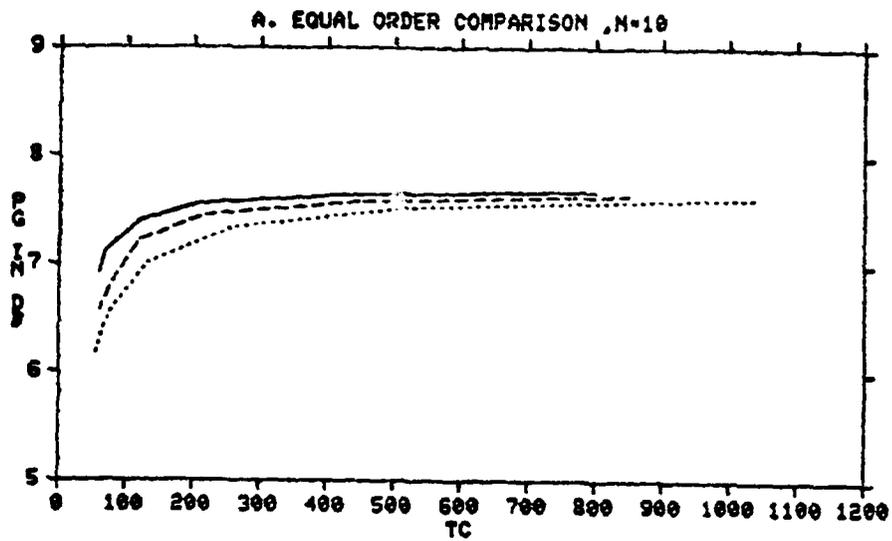
The convergence time, TC, is the time required to reduce 90% of the initial excess MSE. The value of the performance function at the time TC is:

$$J(TC) = J_{ss} + 0.1[J(o) - J_{ss}] \quad (2.1-8)$$

This plot, named the Adaptive Filter Operating Characteristic (AFOC), can be used for design when the number of filter weights, N_a , is a parameter. It also provides a method of comparison for different adaptation schemes. Curves for the LMS, FSC, and SLMS algorithms are presented in Fig. 2.1-1A for the ALE experiment of Appendix A.

We define the following complexity index (CF) for comparing adaptation schemes.

$$CF = \alpha_1 N_{MUL} + \alpha_2 N_{ADD} + \alpha_3 N_{CON} \quad (2.1-9)$$



LEGEND :

— LMS
 FSC
 - - - - SLMS

Fig. 2.1-1
 AFOC Comparison, ALE Experiments
 Average Of 100 Runs

where N_{MUL} , N_{ADD} , N_{CON} are the number of multiplication, addition and control operations used in one iteration. α_1 , α_2 , α_3 are weighting coefficients representing the cost of each operation. A reasonable approximation which neglects control operations is

$$CF = \alpha N_{MUL} + N_{ADD} \quad (2.1-10)$$

Using the equations for LMS, FSC and SLMS techniques we have the following complexity indices as a function of the number of the filter weights, N_a :

$$CF_{LMS} = (2N_a + 1)\alpha + 2N_a + 1 \quad (2.1-11)$$

$$CF_{FSC} = N_a\alpha + 2N_a + 1 \quad (2.1-12)$$

$$CF_{SLMS} = (N_a + 1)\alpha + 2N_a + 1 \quad (2.1-13)$$

As a reasonable numerical example, using $\alpha = 5$, we have approximately equal complexity with $N_{LMS} = 6$, $N_{FSC} = 11$, $N_{SLMS} = 10$. The AFOC comparison for this complexity is presented in Fig. 2.1-1B and indicates that for a given convergence time the simplified gradient methods provide higher processing gain.

2.2 CONVERGENCE AND STABILITY

In this section we discuss conditions for the convergence and the stability of the simplified gradient estimates. A stable adaptive filter is one that converges to a near optimal steady-state. We now define the convergence ratio,

$C_i(k)$:

$$C_i(k) = \frac{a_i(k+1) - a_i^*}{a_i(k) - a_i^*} \quad (2.2-1)$$

where a_i^* is the optimal value for the weight a_i .

Following Widrow et al [1, 3, 4] we define the weight noise, $V_i(k)$, as:

$$V_i(k) = a_i(k) - a_i^* \quad (2.2-2)$$

Combining (2.2-1) and (2.2-2) gives:

$$C_i(k) = \frac{V_i(k+1)}{V_i(k)} \quad (2.2-3)$$

From (2.2-2) and (1.2-6) we get:

$$V_i(k+1) = V_i(k) - \mu_a \hat{\nabla}_{a_i}(k) \quad (2.2-4)$$

Combining (2.2-3) and (2.2-4) we get:

$$C_i(k) = 1 - \mu_a \frac{\hat{\nabla}_{a_i}(k)}{V_i(k)} \quad (2.2-5)$$

The steady state average convergence ratio is defined as:

$$\bar{C}_i = E\{C_i(k)\} = 1 - \mu_a E\left\{\frac{\hat{\nabla}_{a_i}(k)}{V_i(k)}\right\} \quad (2.2-6)$$

where k is large enough for operation of the filter to be in steady-state. From this point we proceed with the specific case of the SLMS, with its gradient estimate given by (2.1-6).

The error as function of the weight noise is given from (2.2-2) and (1.2-4) as:

$$\epsilon_r(k) = \epsilon_r^*(k) + \sum_{j=0}^{N_a-1} V_j(k) X(k-j) \quad (2.2-7)$$

where $\epsilon_r^*(k)$ is the optimal error at time k and is given by:

$$\epsilon_r^*(k) = \sum_{i=0}^{N_a-1} a_i^* X(k-i) - r(k) \quad (2.2-8)$$

Inserting (2.2-7) to (2.1-6) results in the equation:

$$\begin{aligned} \hat{V}_{SLMS}(i,k) &= \epsilon_r^*(k) \text{Sgn}\{X(k-i)\} + \\ &+ \sum_{j=0}^{N_a-1} V_j(k) X(k-j) \text{Sgn}\{x(k-i)\} \end{aligned} \quad (2.2-9)$$

Inserting (2.2-9) into (2.2-6) we get:

$$\begin{aligned} \bar{C}_i &= 1 - \mu_a \left\{ E \left\{ \frac{\epsilon_r^*(k) \text{Sgn}\{x(k-i)\}}{V_i(k)} \right\} \right. \\ &+ \left. \sum_{j=0}^{N_a-1} E \left\{ \frac{V_j(k)}{V_i(k)} \cdot X(k-j) \text{Sgn}\{x(k-i)\} \right\} \right\} \end{aligned} \quad (2.2-10)$$

$\epsilon_r^*(k)$ is independent of $x(k-i)$ and of $V_i(k)$ so that:

$$E\left\{\frac{\epsilon_r^*(k) \text{Sgn}\{x(k-i)\}}{V_i(k)}\right\} = E\{\epsilon_r^*(k)\}E\left\{\frac{\text{Sgn}\{x(k-i)\}}{V_i(k)}\right\} = 0 \quad (2.2-11)$$

because $E[\epsilon_r^*(k)] = 0$.

To continue with the simplification of (2.2-10) we make the following assumptions:

$$\begin{aligned} \text{(a) } & V_i(k) \text{ and } x(k-j) \text{ are uncorrelated} \\ \text{(b) } & E\{V_j(k)/V_i(k)\} = 1 \end{aligned} \quad (2.2-12)$$

Assumption (a) is similar to the uncorrelated input assumption used by Widrow in [1] and seems to be justified by his results. Assumption (b) is made for mathematical convenience and can be justified by the dependence of the weight noises on the common error terms and the uniform statistics of the input signal over the filter memory.

Using (2.2-11) and (2.2-12) in (2.2-10) we get:

$$\bar{C}_i = 1 - \mu_a \sum_{j=0}^{N_a-1} E\{x(k-j) \text{Sgn}\{x(k-i)\}\} \quad (2.2-13)$$

Since $x(k-j) \text{Sgn}\{x(k-i)\} \leq |x(k-j)|$ we can write:

$$\bar{C}_i \leq 1 - \mu_a \sum_{j=0}^{N_a-1} E|x(k-j)| \quad (2.2-14)$$

For stationary input signals $E\{x(k-j)\} = E\{x(k)\}$ with all values of j and we get:

$$\bar{c}_i \leq 1 - \mu_a N_a E|x(k)| \quad (2.2-15)$$

For stable operation, as well as for convergence to the optimal weight values, we require

$$|\bar{c}_i| < 1 \quad (2.2-16)$$

Manipulating (2.2-15) and (2.2-16) to obtain the stability condition for μ_a yields for the SLMS algorithm:

$$0 < \mu_{\text{SLMS}} < \frac{2}{N_a E|x(k)|} \quad (2.2-17)$$

To express (2.2-17) as function of the input power, $R_{xx}(0)$, we can define the input signal form factor, F_x , as:

$$F_x = E|x(k)| / \sqrt{E\{X^2(k)\}} \quad (2.2-18)$$

Now inserting (2.2-18) in (2.2-17) results in:

$$0 < \mu_{\text{SLMS}} < \frac{2}{N_a F_x \sqrt{R_{xx}(0)}} \quad (2.2-19)$$

For the LMS algorithm we can use (2.2-6) and the LMS gradient estimate. Following the above derivation and using the assumptions of (2.2-12) we get

$$0 < \mu_{\text{LMS}} < \frac{1}{N_a R_{xx}(0)} \quad (2.2-20)$$

(2.2-20) is equivalent to equation (32) in [4] which was derived in a different manner but with similar assumptions.

To derive the stability region of the FSC algorithm we use (2.2-17) and the relationship between the FSC and the SLMS algorithms, we define an equivalent adaptation gain, μ_{eq} , by the formula

$$\mu_{FSC} = \mu_{eq} E|\epsilon_r(k)| \quad (2.2-21)$$

It is interesting to note that we are now using the derivation process of Section 2.1 for the SLMS algorithm in a reverse direction. The case of greatest interest is that of a low signal to noise ratio. For this case we use the following approximations:

$$\epsilon_r(k) = y(k) - s(k+1) - n(k+1) \approx -n(k+1) \approx -r(k)$$

and

$$E|\epsilon_r(k)| \approx E|r(k)| = E|x(k)| \quad (2.2-22)$$

Inserting μ_{eq} from (2.2-21) into the SLMS relation, given by (2.2-17), with the use of (2.2-22) results in the following.

$$0 < \mu_{FSC} < 2/N_a \quad (2.2-23)$$

The foregoing relationships (2.2-17), (2.2-20) and (2.2-23), are based upon average behavior of the algorithms. In

practice, to avoid numerical overflow, we must use adaptation gain values much smaller than the upper limit indicated in the above relations. An additional consideration that also results in a smaller adaptation gain is the misadjustment. For all the algorithms, the use of the upper bound value for the adaptation gain results in a misadjustment of the order of the optimal filter gain (PF), which means that practically we are restricted to much lower values of the adaptation gain. The results of this section are included in Table 2.6-1.

2.3 CONVERGENCE TIME ESTIMATION

In order to estimate the convergence time of an adaptive filter one may visualize the process as changing the weights with some average step, Δ , taken in most of the iterations towards the optimal value of the weight. Assuming an initial value of zero for all the weights, the longest convergence time will be associated with weight having the largest absolute value, a_{\max} . From the above it is reasonable to assume the following relationship:

$$TC = \alpha_1 \frac{a_{\max}}{\Delta} \cdot N_a^{\alpha_2} \quad (2.3-1)$$

where: N_a is the number of filter weights, and α_1, α_2 are unknown coefficients. a_{\max}/Δ is the exact number of steps needed for convergence if the correction is always in the right direction. In practice the gradient estimation causes errors in the direction, and the number of iterations required to converge to the optimal value of the weights is modified by a factor that depends in some non-linear way on the number of weights N_a . This modification is represented in (2.3-1) by the factor $\alpha_1 N_a^{\alpha_2}$. Also α_1 depends on the exact definition of TC (i.e. 10% or e^{-1} of the initial error squared). Filter operation involves a linear combination of input values. Since the reference amplitude is independent of N_a , when we combine more input samples the relative weight associated with each sample should be smaller, mathematically:

$$a_{\max} = \frac{\alpha_3}{N_a} \quad (2.3-2)$$

In general, α_3 , depends upon the input signal to noise ratio as discussed in the literature [3, 33]. This dependency is not taken into account in the derivations which follow in order to simplify comparison of the new algorithms with existing algorithms. The results of reference [37] can be used to modify the results presented here to include the dependence upon input signal to noise.

When looking at specific applications, such as Adaptive Line Enhancement (ALE), one can determine the value of α_3 in (2.3-2) exactly. Inserting (2.3-2) to (2.3-1) and absorbing α_3 into α_1 , we write:

$$TC = \frac{\alpha_1 N_a^{\alpha_2}}{\Delta N_a} \quad (2.3-3)$$

Δ in (2.3-3) depends on the adaptive scheme. It is the fixed step size in the FSC algorithm and an average step size for the LMS and the SLMS algorithms. Thus for these three cases we define:

$$\Delta_{\text{FSC}} = \mu_{\text{FSC}} \quad (2.3-4)$$

$$\Delta_{\text{LMS}} = E\{|\mu_{\text{LMS}} \hat{v}_{\text{LMS}}|\} = 2\mu_{\text{LMS}} E\{|\epsilon(k)x(k-i)|\} \quad (2.3-5)$$

$$\Delta_{\text{SLMS}} = E\{|\mu_{\text{SLMS}} \hat{v}_{\text{SLMS}}|\} = \mu_{\text{SLMS}} E\{|\epsilon(k) \text{Sgn } x(k-i)|\} \quad (2.3-6)$$

Using (2.3-3) and (2.3-4) and the empirical coefficients $\alpha_1 = 1.65$ $\alpha_2 = 1/2$ as evaluated using the simulations described in Appendix A, we get the following FSC convergence time to 10% of the initial squared error:

$$TC = \frac{1.65}{\mu_{FSC} \sqrt{N_a}} \quad (2.3-7)$$

where TC is the time required to reduce the error to 10% as defined by (2.1-8). Fig. 2.3-1 presents a verification of (2.3-7) using simulation results with several values of μ_{FSC} , N_a , and the input power $R_{xx}(0)$.

The significance of these results is that they confirm that the convergence time is inversely proportional to the adaptation gain and the square root of the number of weights. Assuming in (2.3-5) that

$$E\{|\epsilon(k)| \cdot |x(k-i)|\} = E|\epsilon(k)| \cdot E|x(k-i)| \text{ we get:}$$

$$\Delta_{LMS} = 2\mu_{LMS} E|\epsilon(k)| E|x(k-i)| \quad (2.3-8)$$

At the start of the adaptation process the initial weights have a value of zero, so that $y(0) = 0$ and $\epsilon(0) = r(0)$. For the correlated reference case the reference power is essentially the same as the input power and we have:

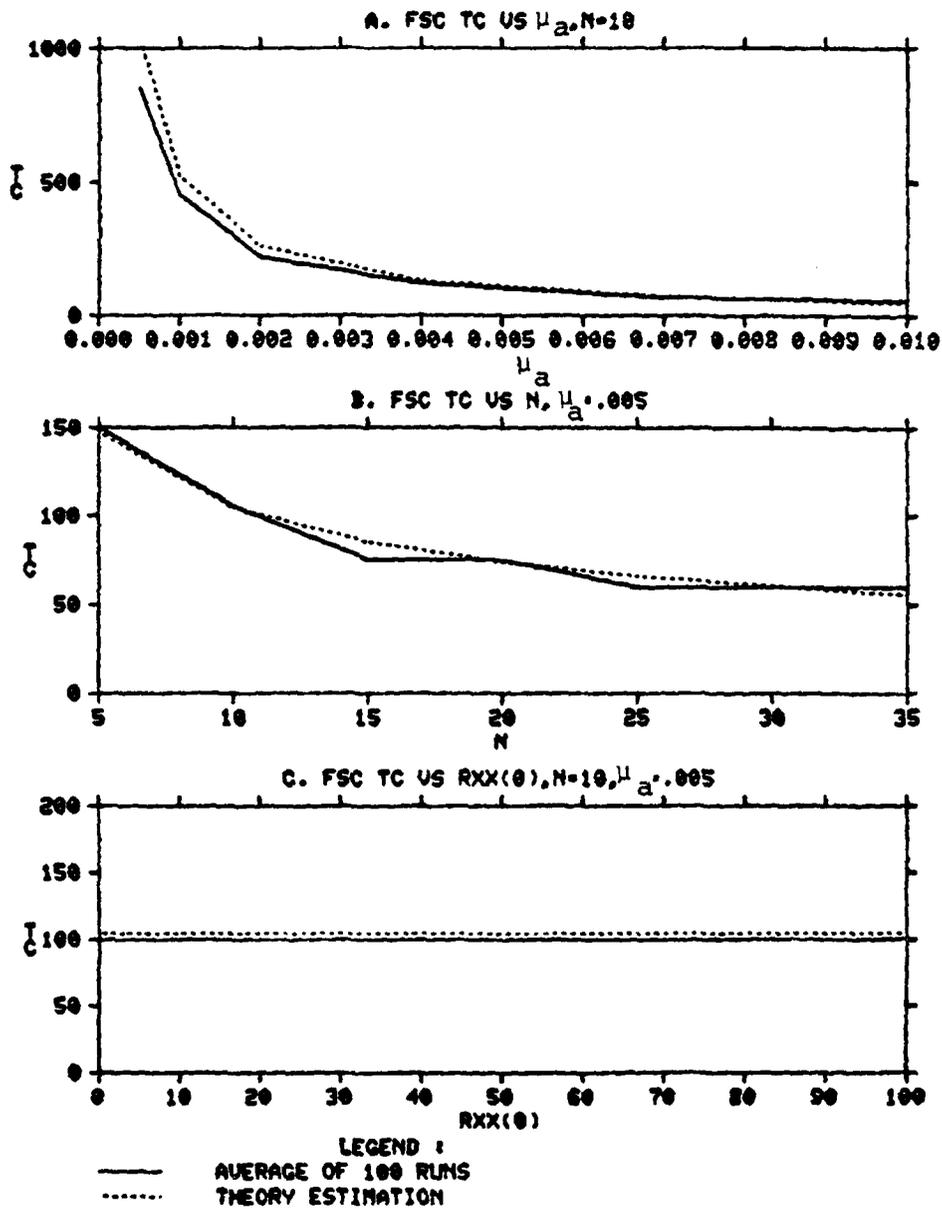


Fig. 2.3-1

Fixed Size Correction Convergence Time
 Theory And Simulation Results

$$\Delta_{LMS} = 2\mu_{LMS} E|x(k)| \cdot E|x(k)| \quad (2.3-10)$$

Using expression (2.3-10) for the LMS average step size in (2.3-3) with $\alpha_2 = 1/2$ we have

$$TC_{LMS} = \frac{\alpha_{LMS}}{\mu_{LMS} (E|x(k)|)^2 \sqrt{N}_a} \quad (2.3-11)$$

In [4] the classical LMS convergence time estimate is given by:

$$TC_{LMS} = \frac{\ln 10}{4 \mu_{LMS} R_{xx}(0)} \quad (2.3-12)$$

Based on the simulation described in appendix A we select

$$\alpha_{LMS} = .555.$$

Fig. 2.3-2 presents a comparison of simulation results with the classical convergence time formula, (2.3-12), and the new convergence time formula, (2.3-11). This figure indicates clearly that the convergence time depends upon the number of weights, N_a , as developed in (2.3-11), and that this formulation is more accurate than that of (3.2-12) which was developed in reference [4]. In a similar way (2.3-6) and (2.3-3) gives

$$TC_{SLMS} = \frac{\alpha_{SLMS}}{\mu_{SLMS} \sqrt{N}_a E|x(k)|} \quad (2.3-13)$$

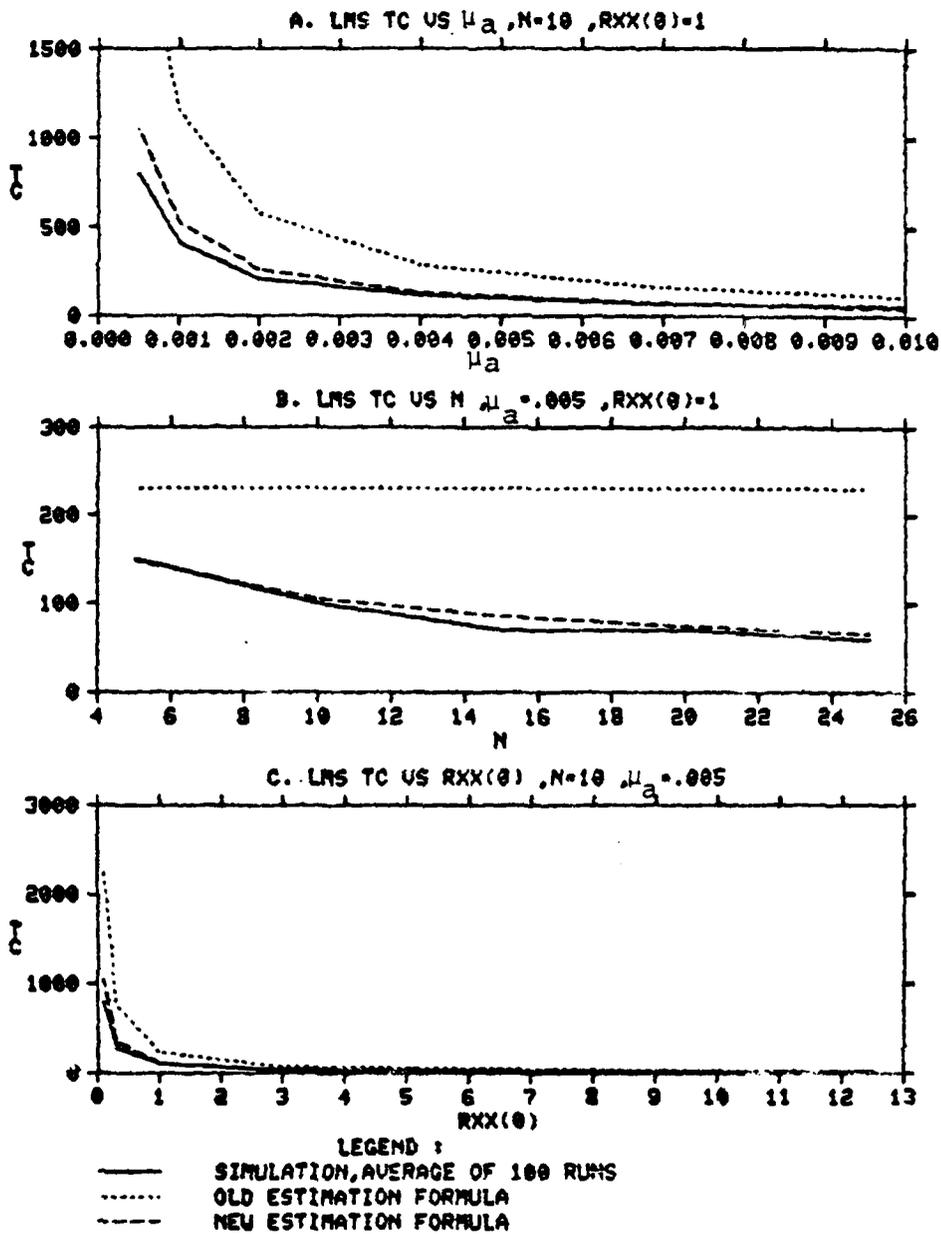


Fig. 2.3-2

LMS Convergence Time, Simulation Results And Theory

Fig. 2.3-3 presents a comparison of (2.3-13) with simulation results, with $\alpha_{\text{SLMS}} = 1.4$, based upon the results of simulations described in Appendix A. The comparison confirms (2.3-13). The key formulas of this section are included in Table 2.6-1.

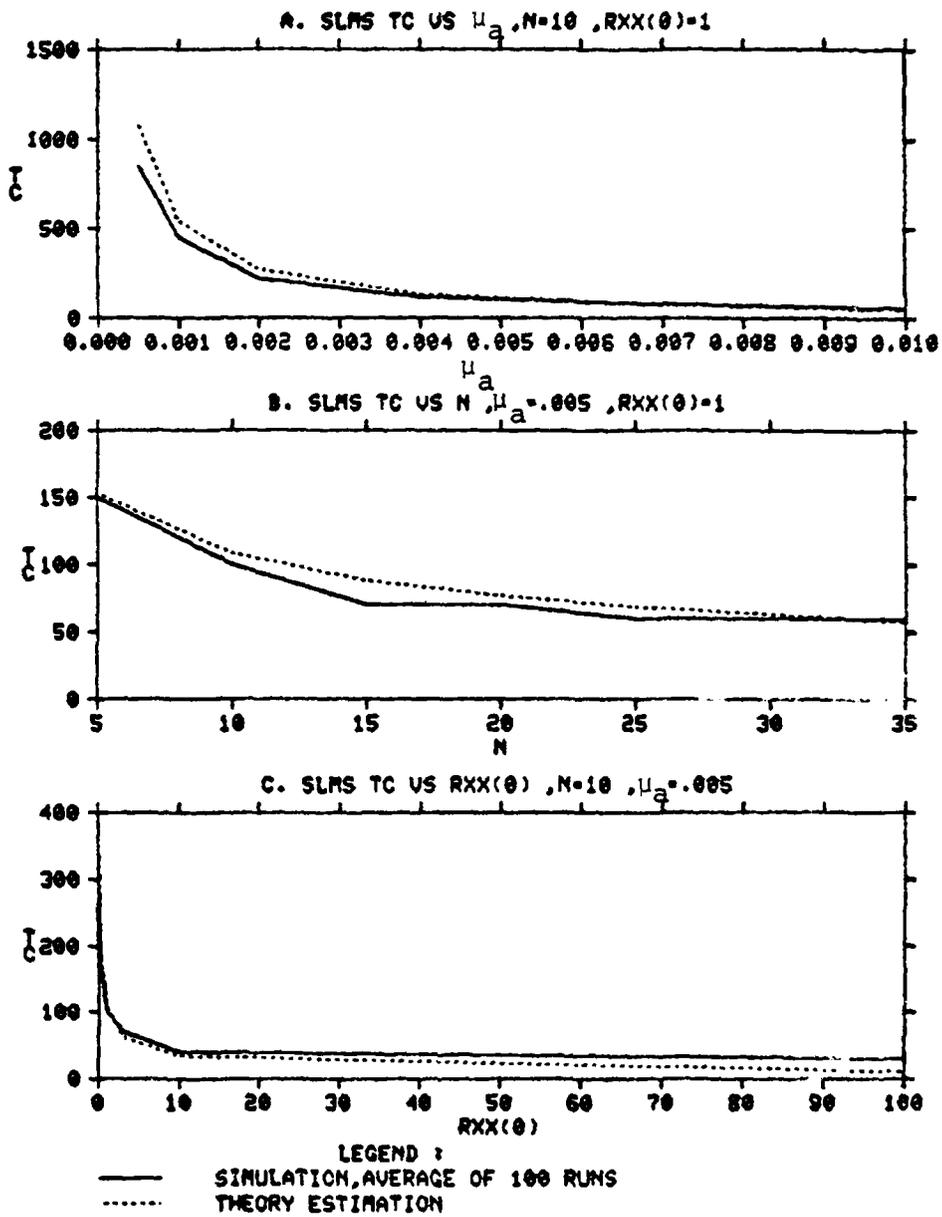


Fig. 2.3-3
 SLMS Convergence Time, Simulation Results And Theory

2.4 STEADY STATE ERROR AND MISADJUSTMENT

In order to evaluate the steady-state error we start with general relationships. First following [1, 4, 3] we define the weight noise $v_i(k)$ as:

$$v_i(k) = a_i(k) - a_i^* \quad (2.4-1)$$

where a_i^* is the optimal i th weight.

$$\begin{aligned} y(k) &= \sum_{i=0}^{N_a-1} a_i(k) x(k-i) = \sum_{i=0}^{N_a-1} a_i^* x(k-i) \\ &\quad + \sum_{i=0}^{N_a-1} v_i(k) x(k-i) \end{aligned} \quad (2.4-2)$$

Define

$$\begin{aligned} \epsilon_s(k) &= y(k) - s(k) = \sum_{i=0}^{N_a-1} a_i^*(k) x(k-i) - s(k) \\ &\quad + \sum_{i=0}^{N_a-1} v_i(k) x(k-i) \end{aligned} \quad (2.4-3)$$

We can now define the optimal instantaneous error:

$$\epsilon_s^*(k) = \sum_{i=0}^{N_a-1} a_i^* x(k-i) - s(k) \quad (2.4-4)$$

Using this value the minimum mean squared error is:

$$J_{\min} = E \{ \epsilon_s^*(k)^2 \}, \text{ with } k \text{ in the steady state.} \quad (2.4-5)$$

From (2.4-3) and (2.4-4) it follows that:

$$\begin{aligned} J_{ss} &= J(k) \mid k \text{ in the steady state} \\ &= E \left\{ \left\{ \epsilon_s^*(k) + \sum_{i=0}^{N_a-1} v_i(k) x(k-i) \right\}^2 \right\} \\ &= E \{ \epsilon_s^{*2}(k) \} + E \left\{ 2 \sum_{i=0}^{N_a-1} v_i(k) \epsilon_s^*(k) x(k-i) \right\} \\ &\quad + E \left\{ \sum_{i_1=0}^{N_a-1} \sum_{i_2=0}^{N_a-1} v_{i_1}(k) v_{i_2}(k) x(k-i_1) x(k-i_2) \right\} \end{aligned} \quad (2.4-6)$$

The foregoing assumes:

- (a) The expectation of $v_i(k) x(k-i)$ is factorable.
- (b) $E\{v_i(k)v_j(k)\} = \overline{v^2} \delta_{ij}$

$$\text{where } \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Assumptions (a) and (b) appear to be well justified in the case of a correlated reference signal, as confirmed by the agreement obtained between the derived formulas and the simulation results.

The second term of (2.4-6) can be factored:

$$\{v_i(k) \varepsilon_s^*(k) x(k-i)\} = E\{v_i(k)\} \cdot E\{\varepsilon_s^*(k) \cdot x(k-i)\} \quad (2.4-7)$$

However, $E\{\varepsilon_s^*(k) x(k-i)\} = 0$ (because of the orthogonality of the optimal solution) so that the second term is zero and (2.4-6), using assumptions (a) and (b), becomes:

$$\begin{aligned} J_{ss} &= J_{\min} + \sum_{i=0}^{N_a-1} \overline{v^2} E\{x(k-i) x(k-i)\} \\ &= J_{\min} + N_a \overline{v^2} R_{xx}(0) \end{aligned} \quad (2.4-8)$$

We now define the excess MSE as follows:

$$J_e = J_{ss} - J_{\min} = N_a \overline{v^2} R_{xx}(0) \quad (2.4-9)$$

and the Misadjustment as:

$$M = \frac{J_e}{J_{\min}} = \frac{N_a \overline{v^2} R_{xx}(0)}{J_{\min}} \quad (2.4-10)$$

the foregoing depends upon J_{\min} , N_a , $\overline{v^2}$, and $R_{xx}(0)$. $R_{xx}(0)$ depends upon the statistics of the input. J_{\min} depends upon the input statistics as well as N_a . However, $\overline{v^2}$ depends upon the nature of the specific algorithm and will now be considered for the SLMS and FSC algorithms.

From (2.4-1) and (1.2-6) we have:

$$v_i(k+1) = v_i(k) - \mu_a \hat{v}_{a_i}(k) \quad (2.4-11)$$

Squaring both sides we get:

$$v_i^2(k+1) = v_i^2(k) + \mu_a^2 \hat{v}_{a_i}^2(k) - 2\mu_a v_i(k) \hat{v}_{a_i}(k) \quad (2.4-12)$$

For the SLMS algorithm:

$$\begin{aligned} v_i(k) \hat{v}_{a_i}(k) &= v_i(k) \varepsilon_r(k) \text{Sgn} \{x(k-i)\} = \\ &= v_i(k) \left\{ \varepsilon_r^*(k) + \sum_{j=0}^{N_a-1} v_j(k)x(k-j) \right\} \text{Sgn} \{x(k-i)\} \end{aligned} \quad (2.4-13)$$

where

$$\varepsilon_r(k) = y(k) - r(k) \quad (2.4-14)$$

and

$$\varepsilon_r^*(k) = \sum_{i=0}^{N_a-1} a_i^* x(k-i) - r(k) \quad (2.4-15)$$

$\varepsilon_r(k)$ and $\varepsilon_r^*(k)$ depend on the reference signal $r(k)$, and in many cases, including the correlated reference case, those errors and the previously defined $\varepsilon_s(k)$ and $\varepsilon_s^*(k)$ have completely different statistics.

$v_i(k)$ and $x(k-i)$ are independent of $\varepsilon_r^*(k)$ so we have:

$$E \{v_i(k) \varepsilon_r^*(k) \text{Sgn} \{x(k-i)\}\} =$$

$$E \{\varepsilon_r^*(k)\} \cdot E\{v_i(k) \text{Sgn} \{x(k-i)\}\} = 0 \quad (2.4-16)$$

we get:

$$E\{v_i(k) \hat{v}_{a_i}(k)\} = \sum_{j=0}^{N_a-1} E\{v_i(k) v_j(k) x(k-j) \text{Sgn}\{x(k-i)\}\} \quad (2.4-17)$$

Using assumptions (a) and (b) in (2.4-17) we have:

$$E\{v_i(k) \hat{v}_{a_i}(k)\} = \bar{v}^2 E|x(k)| \quad (2.4-18)$$

taking the expectation of (2.4-12) in the steady state and using (2.4-18) we have:

$$\begin{aligned} E[v_i^2(k+1)] &= E[v_i^2(k)] + \mu_a^2 E\{\varepsilon_r^2(k) \text{Sgn}^2[x(k-i)]\} \\ &\quad - 2\mu_a \bar{v}^2 E|x(k)| \end{aligned} \quad (2.4-19)$$

In the steady state $E[v_i^2(k+1)] = E[v_i^2(k)]$ and

from (2.4-19) we can express \bar{v}^2 as:

$$\bar{v}^2 = \frac{\mu_a E[\varepsilon_r^2(k)]}{2 E|x(k)|} \quad (2.4-20)$$

Now we can insert (2.4-20) to (2.4-10) and get:

$$M = \frac{\mu_a N_a E[\epsilon_r^2(k)] R_{xx}(0)}{2 J_{\min} E|x(k)|} \quad (2.4-21)$$

$E[\epsilon_r^2(k)]$ depends upon the type of reference used. For the correlated reference we have:

$$\epsilon_r(k) = y(k) - r(k) = y(k) - [s(k+1) + n(k+1)] \quad (2.4-22)$$

By squaring and taking the expectation of (2.4-22) we get:

$$\begin{aligned} E[\epsilon_r^2(k)] &= E\{[y(k) - s(k+1)]^2\} + E[n^2(k+1)] \\ &\quad - 2E\{y(k) n(k+1) - s(k+1) n(k+1)\} \end{aligned} \quad (2.4-23)$$

In the third term of (2.4-23), $n(k+1)$ is independent of $s(k+1)$, and the present output $y(k)$ is independent of the future noise $n(k+1)$, so this term's expectation is zero. Because of the correlation of $s(k+1)$ and $s(k)$, which is a basic requirement for the use of the correlated reference, the first term of (2.4-23) will be:

$$E\{[y(k) - s(k+1)]^2\} = E\{[y(k) - s(k)]^2\} = E[\epsilon_s^2(k)] = J_{ss} \quad (2.4-24)$$

For stationary noise we have

$$R_{nn}(0) = E[n^2(k)] = E[n^2(k+1)] \quad (2.4-25)$$

Using (2.4-24) and (2.4-25) and the above reasoning about the 3rd term, (2.4-23) becomes:

$$E[\epsilon_r^2(k)] = J_{ss} + R_{nn}(0) \quad (2.4-26)$$

For reasonable processing gain $J_{ss} \ll R_{nn}(0)$ and

$$E[\epsilon_r^2(k)] = R_{nn}(0) \quad (2.4-27)$$

The optimal processing factor (PF) of a filter is defines as:

$$PF = \frac{R_{nn}(0)}{J_{min}} \quad (2.4-28)$$

PF express the optimal noise reduction possible by an optimal filter of order N_a . PF depends upon N_a and the signal statistics, and does not depend upon the adaptation algorithm and the adaptation gain.

Inserting (2.4-27) into (2.4-21) we get:

$$M = \frac{\mu_a N_a R_{nn}(0) R_{xx}(0)}{2 E|x(k)| J_{min}} \quad (2.4-29)$$

Using definition (2.4-28) in (2.4-29) we get:

$$M = \frac{\mu_a N_a R_{xx}(0)}{2 E|x(k)|} \cdot PF \quad (2.4-30)$$

Fig. 2.4-1 presents a verification of (2.4-30), using simulation results with several values of μ_{SLMS} , N_a , and the input power $R_{xx}(0)$.

Equation (2.4-30) was derived for the SLMS algorithm. To derive an equivalent expression for the FSC algorithm, we use (2.4-30) and the relationship between the FSC and the SLMS algorithms given by (2.2-21). The case of greatest interest is that of a low signal to noise ratio, for this case we can use the approximation given by (2.2-22). Inserting μ_{eq} from (2.2-21) into the SLMS relation, given by (2.4-30), with the use of (2.2-22), results in the following:

$$M = \frac{\mu_{FSC} N_a R_{xx}(0)}{2 [E|x(k)|]^2} PF \quad (2.4-31)$$

Equation (2.4-31) provides an estimate of the misadjustment of the FSC algorithm and Fig. 2.4-2 illustrates its agreement with the simulations. It should be noted that because of the approximation of (2.4-31), the accuracy of (2.2-22), and the accuracy of (2.4-31) should improve for lower signal to noise ratios.

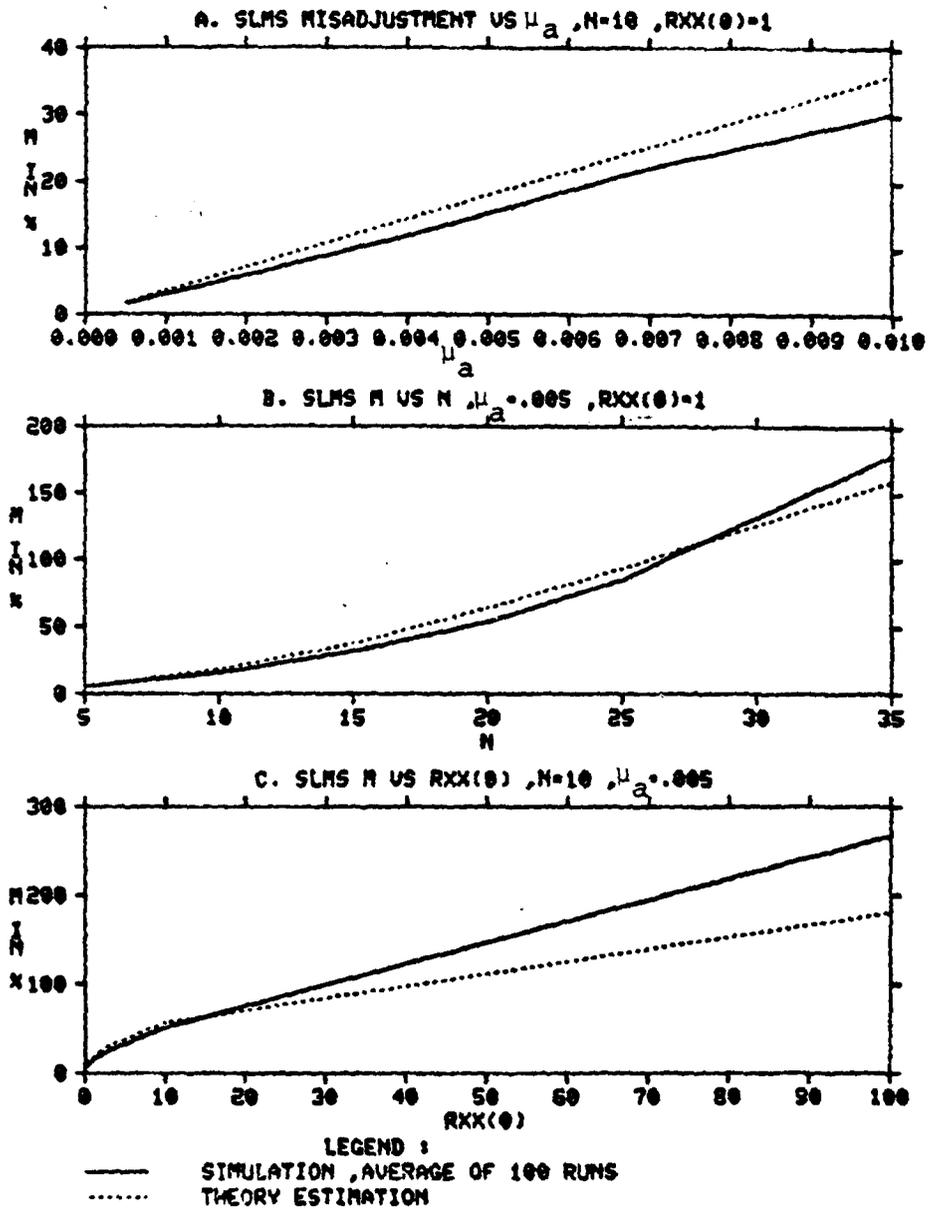


Fig. 2.4-1

SLMS Misadjustment, Simulation Results and Theory

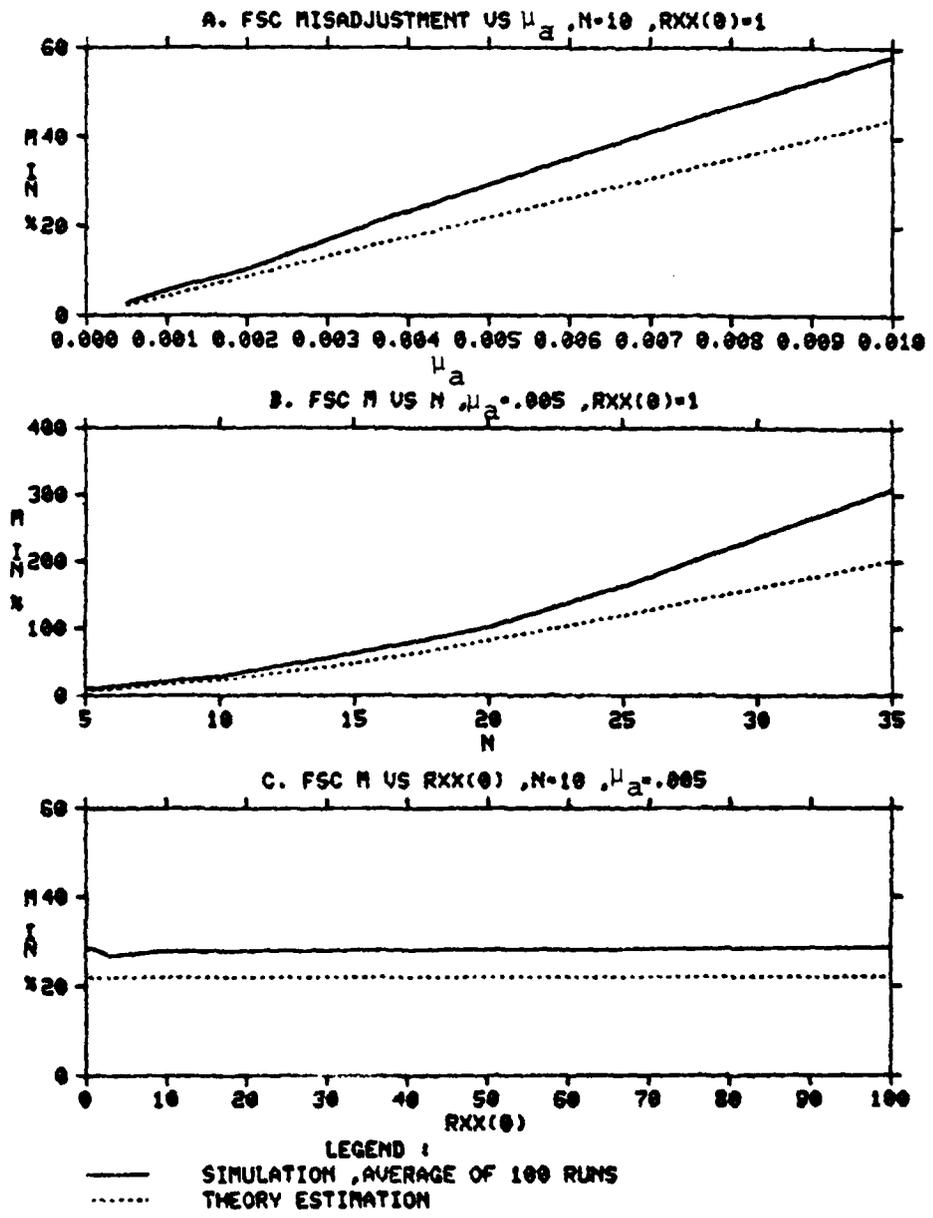


Fig. 2.4-2

FSC Misadjustment, Simulation Results And Theory

2.5 DESIGN CONSIDERATIONS

The design problem of a FIR adaptive filter involves the following major points:

- (a) Selection of an algorithm: LMS, FSC, SLMS.
- (b) Determination of the order of the filter, N_a .
- (c) Determination of the adaptation gain, μ_a .

The discussion which follows does not consider the following points:

- (a) A possible IIR filter solution.
- (b) Implementation details.
- (c) Minimization of a given design criteria, such as: cost, volume, weight ... etc.

The adaptive filter is usually part of a larger system which sets its design requirements. The adaptive filter specifications that we consider here are: a desired processing gain and an upper limit to the convergence time. The additional information required for the design is some specification of the expected input signal to the adaptive filter. Realizing that a complete analysis of adaptive filter behavior is not possible for complicated signals, we consider a design procedure based upon simulation and a graphic presentation of the adaptive filter properties, the adaptive filter operating characteristic (AFOC) as defined in section 2.1. As an example, we considered enhancement of a single sine wave of unknown frequency with a signal to white noise ratio of 0 dB. The desired processing gain is 8 dB and the allowed convergence time is 100 iterations.

Table 2.5-1 outlines the suggested design procedure and presents the application of this procedure to the foregoing example. Since we do not specify implementation details, step (6) of the procedure cannot be carried out for the example. Hence the example is done for the SLMS algorithm only.

Table 2.5-1

FIR Adaptive Filter Design Procedure

Step	Description	Example
1 Test Signal Selection	Define a test signal (or several test signals) for which the filter performance will be evaluated. The dynamic range of the input signal should be considered at this point, and might influence the selection of the test signals. The test signal might be average, worst case, or several typical signals.	Sine wave plus white noise with signal to noise ratio of 0 dB.
2 Simula- tion	Use simulation to generate an Adaptive Filter Operating Characteristic (AFOC) with N_a as a parameter for the LMS, FSC, SLMS algorithms for each of the test signals selected in step 1.	Fig. 2.5-1
3 Determin- ation of N_a for each test signal	For each of the algorithms draw, on the AFOC plots, lines for the desired processing gain and convergence time. For each algorithm select the smallest number of weights that meets the requirements. At this point the designer might consider trade-off in N_a , PG, TC.	In Fig. 2.5-1 we select $N_a=14$
4 Determin- ation of μ_a for each test signal	Since each curve on the AFOC is constructed for several values of μ_a , one can use this data and the values of N_a , TC and PG selected in ^a step 3 to determine the appropriate value for the adaptation gain.	Fig. 2.5-2 we select $\mu_a=.004$.

Table 2.5-1 continued

Step	Description	Example
<p>5 Select optimal parameters for each algorithm</p>	<p>Using the information from all the test signals we need to: (1) Select a single N_a^* and μ_a^* for each algorithm. (2) For the three algorithms evaluate the performance with this N_a^*, μ_a^* for each test signal. (3) Determine for the LMS and SLMS algorithms whether adjustment for dynamic range is required.</p>	<p>Fig. 2.5-3 we have $PG=7.9$ dB, $TC=114$. This performance is marginal and a higher order filter should be considered</p>
<p>6 Selection of algorithm</p>	<p>At this point in order to complete the specification of the filter the designer can compare the resulting complexity of the three candidates and select the best one. The decision depends upon implementation details.</p>	

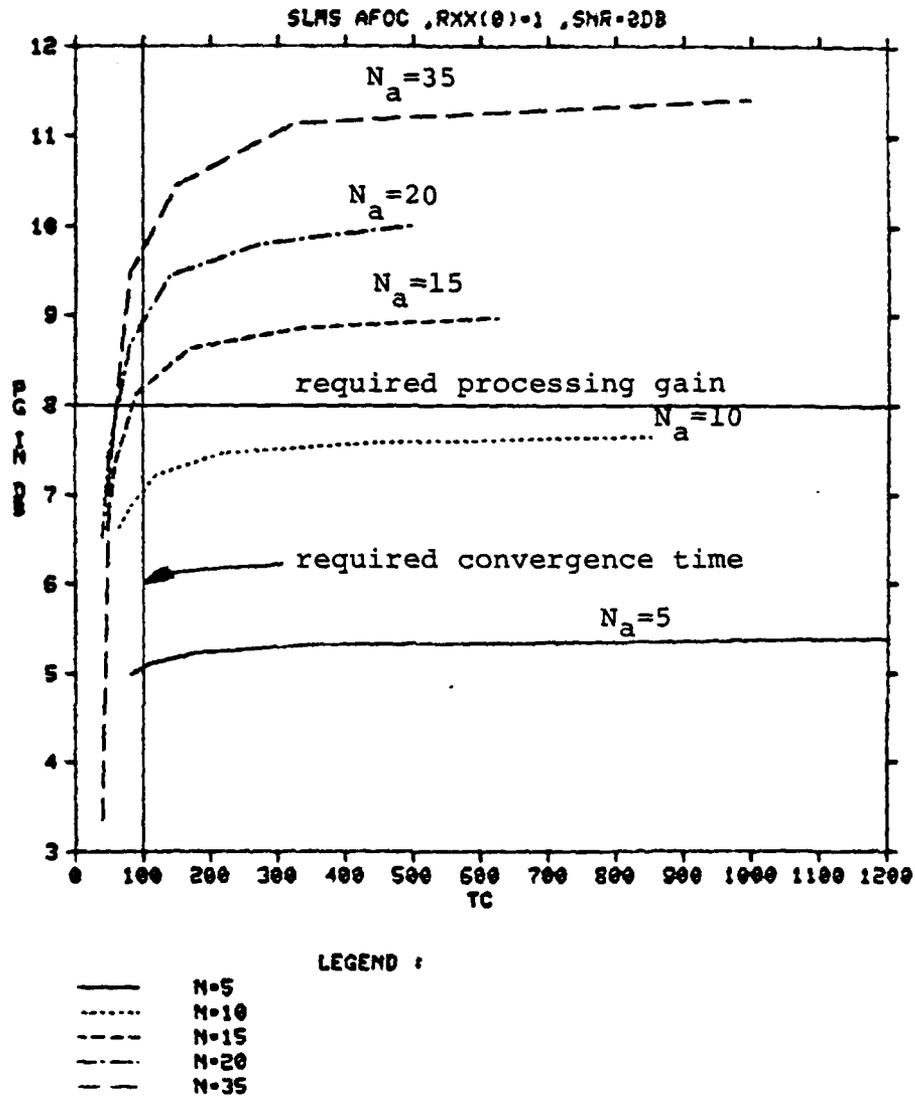


Fig. 2.5-1

AFOC For The SLMS Algorithm, With Details Of The Design Example. Input Signal Is A Sine Wave Plus White Noise With SNR= 0 dB.

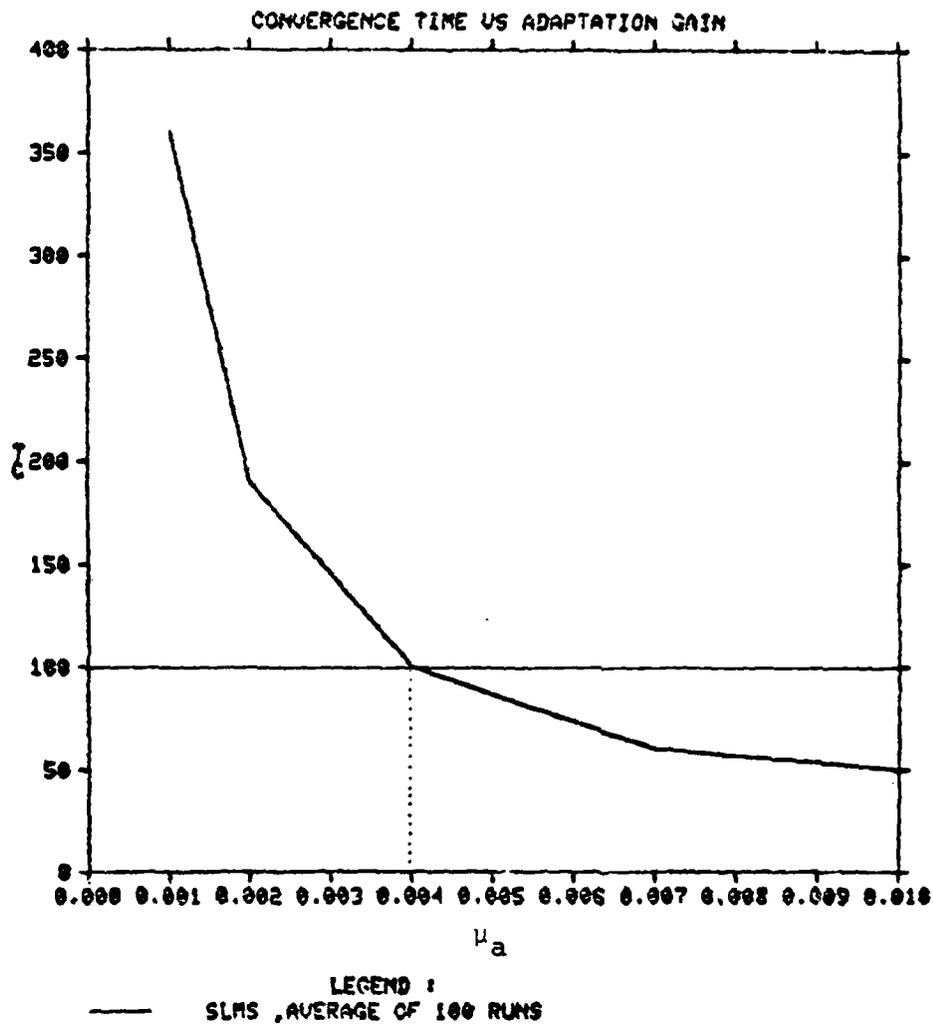


Fig. 2.5-2

Convergence Time As Function Of Adaptation Gain For SLMS With $N_a=14$. For $TC=100$ We Select $\mu_a=.004$

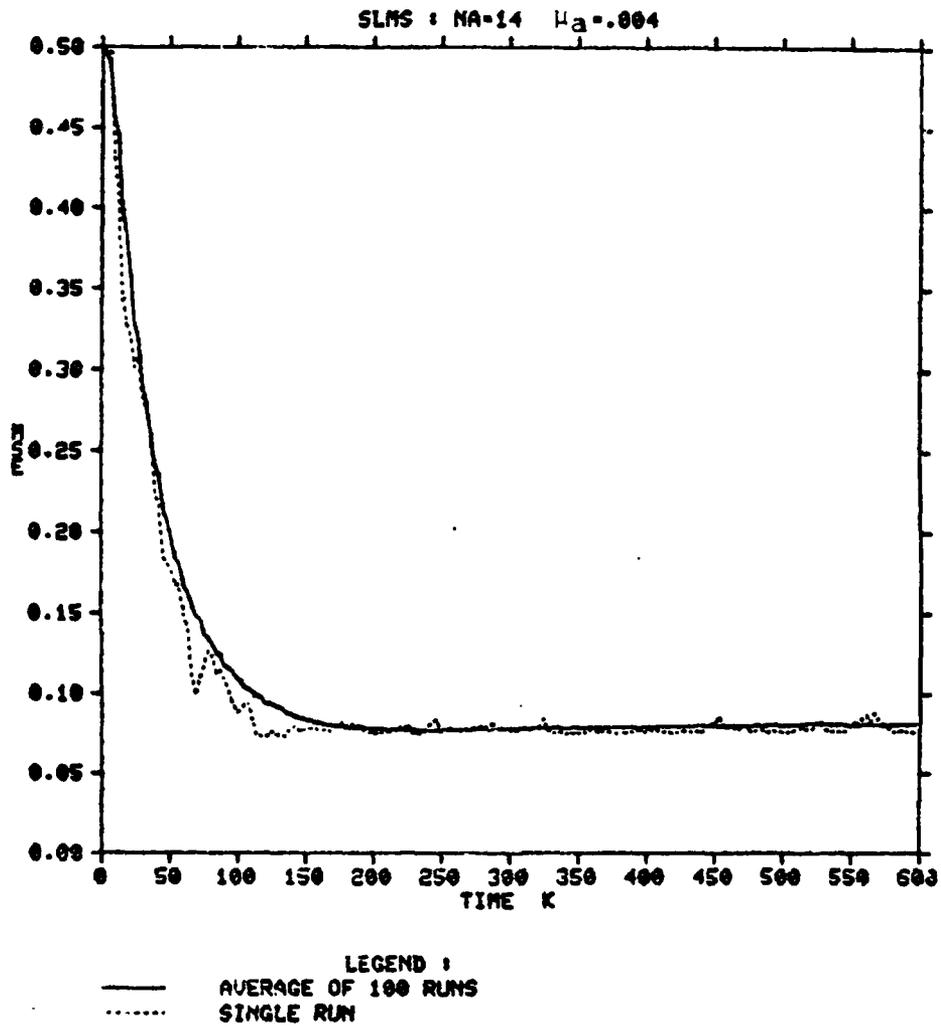


Fig. 2.5-3

Typical Operation Of The Filter With The Selected Parameters.

2.6 CONCLUSION

The simplified gradient estimation algorithms, FSC and SLMS, have processing gain and convergence time similar to the classical LMS algorithm as shown in section 2.1. This similarity of performance has been confirmed when all the filters which were compared have the same order, N_a . Thus, when one considers the implementation savings of the simplified algorithms, the comparison favors the simplified versions.

Analytical comparison of the algorithms is possible using the results in sections 2.2, 2.3, and 2.4, which were developed for the adaptive line enhancer.

A summary of these properties is presented in Table 2.6-1 and compared with the LMS algorithm properties taken from [4].

Since $E|x(k)| = k \sqrt{R_{xx}}(0)$, it is clear from Table 2.6-1 that the dynamic range of the FSC algorithm is the best, because M and TC are not functions of $R_{xx}(0)$. The LMS algorithm has the poorest dynamic range, since M and TC depend on $R_{xx}(0)$. The SLMS algorithm is in the middle since M and TC depend on the square root of $R_{xx}(0)$. Fig. 2.6-1 presents the dynamic range properties of these algorithms. Finally a systematic approach to efficient adaptive filter design has been outlined.

Table 2.6-1

Summary of Algorithms Properties
Derived for ALE Example

	Misadjustment-M	Convergence Time-TC	Stability Limit
LMS	$[\mu_a N_a R_{xx}(0)]PF$ (see note 2)	$\frac{\ln 10}{4\mu_a R_{xx}(0)}$ (Eq. 2.3-12, see note 1)	$1/[N_a R_{xx}(0)]$ (Eq. 2.2-20)
FSC	$[\frac{\mu_a N_a R_{xx}(0)}{2(E x(k))^2}]PF$ (Eq. 2.4-31)	$\frac{1.65}{\mu_a \sqrt{N_a}}$ (Eq. 2.3-7)	$2/N_a$ (Eq. 2.2-23)
SLMS	$[\frac{\mu_a N_a R_{xx}(0)}{2E x(k) }]PF$ (Eq. 2.4-30)	$\frac{1.4}{\mu_a \sqrt{N_a} E x(k) }$ (Eq. 2.3-14, with $\alpha_{SLMS} = 1.4$)	$2/[N_a E x(k)]$ (Eq. 2.2-17)

$$PF = \frac{R_{nn}(0)}{J_{\min}}$$

Notes:

- (1) A new LMS convergence time estimate, using (2.3-11) with $\alpha_{LMS} = .555$, is given by:

$$TC = \frac{0.555}{\mu_a (E|x(k)|)^2 \sqrt{N_a}} = \frac{\alpha}{\mu_a R_{xx}(0) \sqrt{N_a}}$$

- (2) The LMS relationship taken from [4] with modification for the ALE example. This modification is similar to the derivation of (2.4-30).

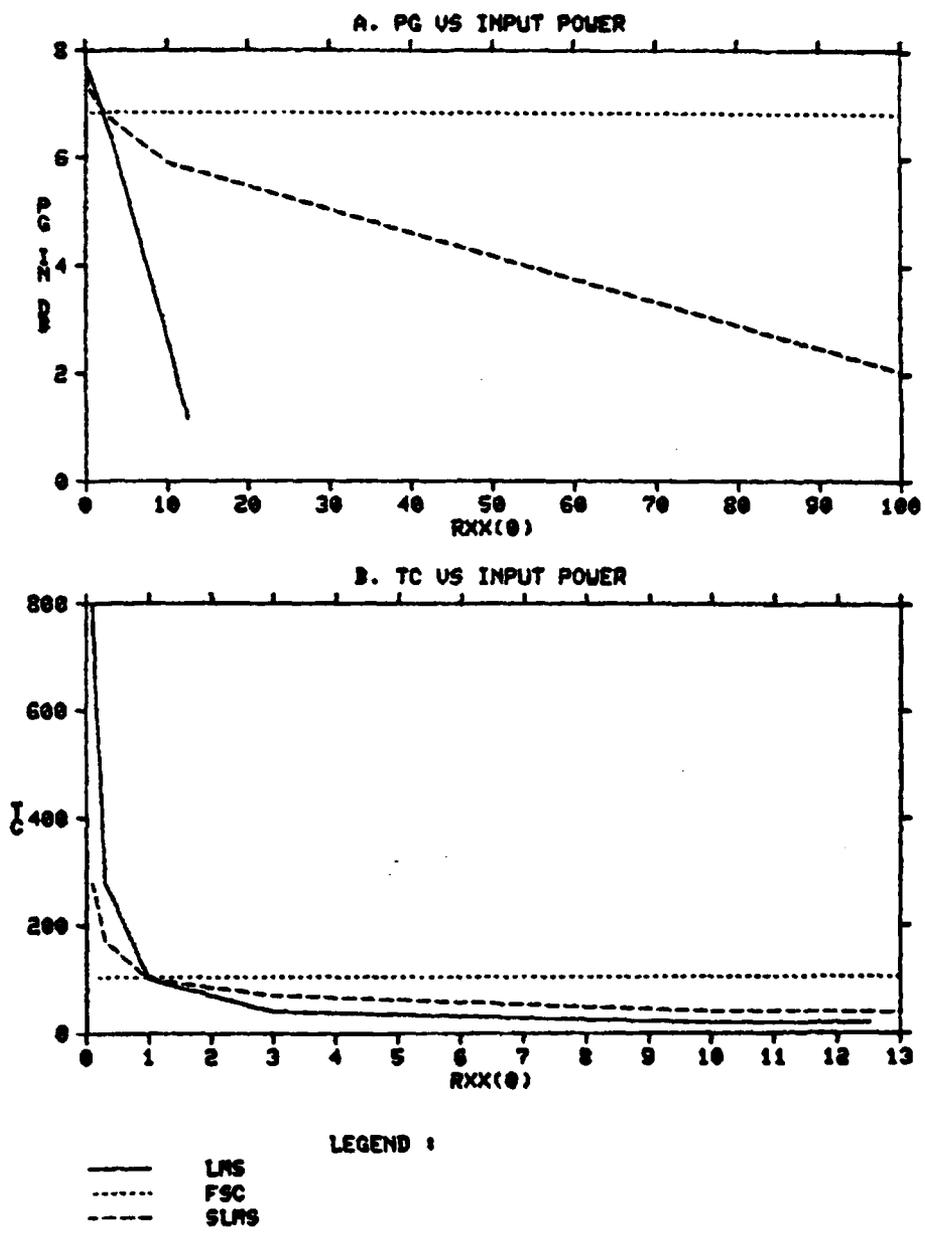


Fig. 2.6-1
 Dynamic range of the algorithms $N_a=10$, $\mu_{LMS} = .0025$
 $\mu_{FSC} = \mu_{SLMS} = .005$, ALE simulation results average of
 100 runs.

III. RANDOM SEARCH IIR ADAPTIVE FILTERS

3.1 IIR PERFORMANCE SURFACE

The performance surface of the IIR filter is much more complicated than the FIR performance surface because of the feedback of previous output values used to form the present output.

Consider the filter:

$$y(k) = \sum_{i=0}^{N_a-1} a_i x(k-i) + \sum_{i=1}^{N_b} b_i y(k-i) \quad (3.1-1)$$

and the error $\epsilon_s(k) = y(k) - s(k)$. The MSE performance surface is given by:

$$J(\{a_i\}, \{b_i\}) = E[\epsilon_s^2(k)] = E\left\{ \left[\sum_{i=0}^{N_a-1} a_i x(k-i) + \sum_{i=1}^{N_b} b_i y(k-i) - s(k) \right]^2 \right\} \quad (3.1-2)$$

In (3.1-2) k is large enough for operation of the filter to be in steady-state.

Manipulating (3.1-2) we get:

$$J(\{a_i\}, \{b_i\}) = \sum_{i=0}^{N_a-1} \sum_{j=0}^{N_a-1} a_i a_j R_{xx}(i-j) \quad (3.1-3)$$

$$+ \sum_{i=1}^{N_b} \sum_{j=1}^{N_b} b_i b_j R_{yy}(i-j) + R_{ss}(0) - 2 \sum_{j=1}^{N_b} b_j R_{sy}(j)$$

$$+ 2 \sum_{i=0}^{N_a-1} \sum_{j=1}^{N_b} a_i b_j R_{xy}(i-j) - 2 \sum_{i=0}^{N_a-1} a_i R_{sx}(i)$$

where the correlation functions are:

$$R_{xx}(m) = E[x(k) x(k-m)]$$

$$R_{yy}(m) = E[y(k) y(k-m)]$$

$$R_{sx}(m) = E[s(k) x(k-m)]$$

$$R_{sy}(m) = E[s(k) y(k-m)]$$

$$R_{xy}(m) = E[x(k) y(k-m)]$$

$$R_{ss}(0) = E[s(k) s(k)]$$

Equation (3.1-3) appears to be quadratic in the weights, but actually $R_{yy}(m)$, $R_{xy}(m)$ and $R_{sy}(m)$ also depend on the weights.

The dependence of the performance surface (3.1-3) on the weights is of high order, and the surface has several minima, only one of which is the global minimum. To demonstrate the complexity of this performance surface consider the simple filter:

$$y(k) = ax(k) + by(k-1) \quad (3.1-4)$$

One can recursively insert successive expressions for $y(k-i)$.

Thus:

$$y(k) = ax(k) + b[ax(k-1) + b[ax(k-2) + b[ax(k-3) + \dots$$

(3.1-5)

or in a compact form:

$$y(k) = a \sum_{i=0}^k b^i x(k-i) \quad (3.1-6)$$

$$\begin{aligned} J(a,b) &= E\{[y(k)-s(k)]^2\} \\ &= a^2 \sum_{i=0}^k \sum_{j=0}^k b^i b^j E[x(k-i)x(k-j)] \\ &\quad + E[s^2(k)] - 2a \sum_{i=0}^k b^i E[x(k-i)s(k)] \\ &= a^2 \sum_{i=0}^k \sum_{j=0}^k b^{i+j} R_{xx}(j-i) + R_{ss}(0) \\ &\quad - 2a \sum_{i=0}^k b^i R_{sx}(i) \end{aligned} \quad (3.1-7)$$

Since $R_{xx}(m)$ and $R_{sx}(m)$ do not depend on the values of the weights a, b the degree of the performance surface given by (3.1-7) is already $2k$ for b and quadratic in a . When the filter operates for a long time, $k \rightarrow \infty$ and (3.1-7) is an infinite sum and an infinite degree polynomial.

Elliott, Jacklin and Stearns, [25], presents an expression for the performance surface which is derived for the general case, with N_a forward weights and N_b backwards weights, this expression is similar to (3.1-7).

The general case also has quadratic dependence on the a 's and infinite polynomial dependence on the b 's. Thus the use of gradient search methods to optimize multimodal performance surface can be expected to result in a steady

state around one of the minima point which is not necessarily the desired global minimum. The steady-state minimum point depends upon the initialization of the adaptive filter. This behavior has been demonstrated for the Stearns' algorithm by Parikh and Ahmed. [13]

3.2 THE RANDOM SEARCH CONCEPT

A random search method consists of evaluation of estimates of the performance surface at discrete sets of the filter weights. After evaluation of these performance surface estimates, a comparison is made and a minimum point is selected.

The method of selection of filter weights for which the performance surface estimate is to be evaluated, is very important. In order to have a useful adaptation scheme for non-stationary input signals a continuous search method is needed, in contrast to possible two phase method that has a global search phase and then fine tuning. From the several methods in the literature [19, 20, 21, 23, 30, 31, 33, 36] the needed continuity of operation is provided by the moving center method.

The center is the point in the parameter space $\{W_i\}$ with the lowest estimate of the performance function among the points that have been tested so far. The set of filter weights (or in general, system parameters) to be tested, at the l th random search interval, $\{\tilde{W}_i\}_l$, is given by:

$$\tilde{W}_{i,l} = W_{i,l} + \mu g \quad \text{for all } i \quad (3.2-1)$$

where:

i is the parameter index

$\{W_i\}_l$ is the value of the center at the l th random search interval, $W_{i,l}$ is its i th element

g is a number independently generated for each weight from a gaussian random number generator, with zero mean and unity variance

$\mu > 0$ determines the range covered by one step of the search and is taken to be the same for all weights which need not be the case

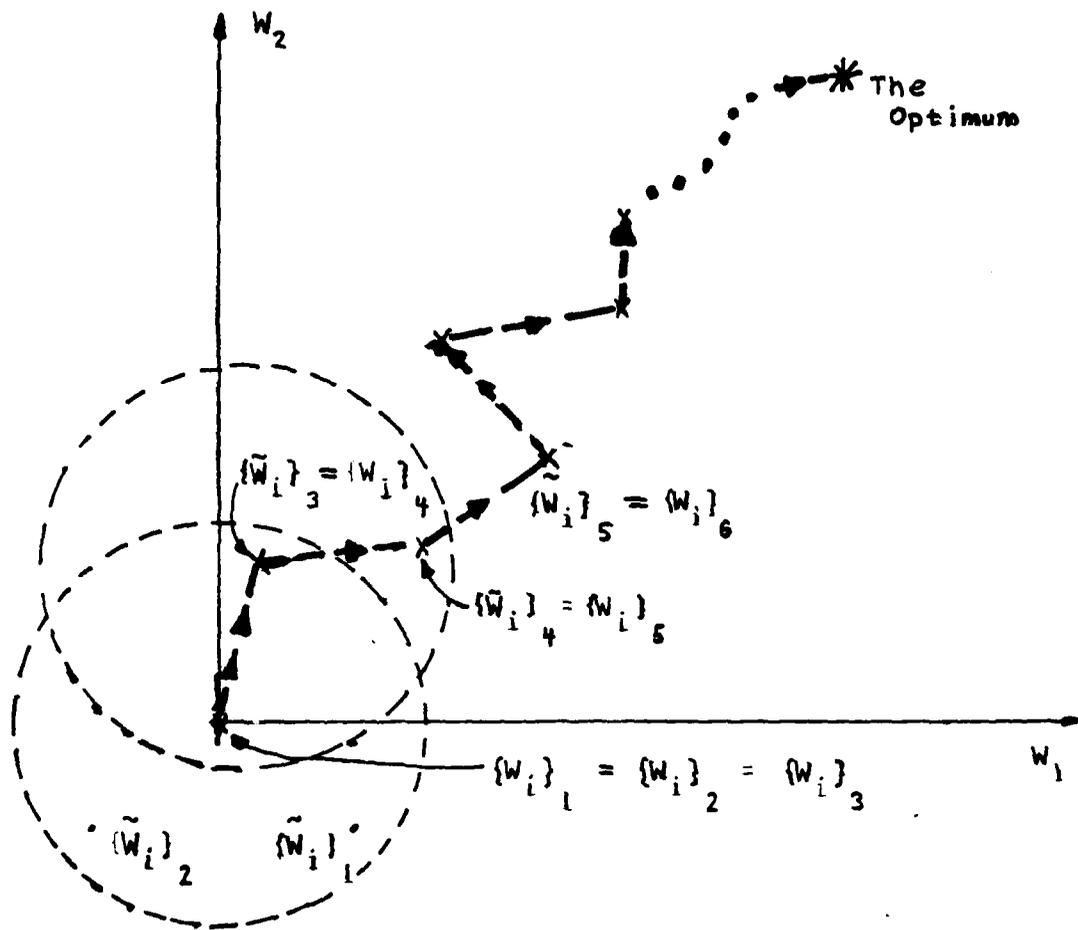
$\{\tilde{W}_i\}_\ell$ is a set of randomly selected parameter values around the center point $\{W_i\}_\ell$, during the ℓ th random search evaluation interval. $\tilde{W}_{i,\ell}$ is its i th element.

The test point (the set $\{\tilde{W}_i\}_\ell$) is tested; that is the value of the performance surface, \tilde{J}_ℓ , is estimated as \hat{J}_ℓ , and compared to the current center estimated value, \hat{J}_ℓ .

If $\hat{J}_\ell \leq \tilde{J}_\ell$, a new point in parameter space is selected using equation (3.2-1). If $\tilde{J}_\ell < \hat{J}_\ell$ the test point corresponds to a lower performance surface value estimate, and the center moves to a new location; that is we set $W_{i,\ell+1} = \tilde{W}_{i,\ell}$ for all i . Now (3.2-1) is used again to evaluate another point to be tested.

Fig. 3.2-1 presents a two parameter example of a moving center random search process. It should be noted that \hat{J}_ℓ and \tilde{J}_ℓ are only an estimate of the performance surface points J_ℓ and \tilde{J}_ℓ because the latter properly involves averaging over an infinite ensemble.

In order to use the random search method in adaptive filters we need to specify the performance function and to define some estimate of that function. Since we are comparing the performance function and not evaluating it's gradient one



Legend:

- tested and rejected point
- x tested and accepted, a new center
- (x) area with high probability of selection points to be tested
- the movement of the center

Fig. 3.2-1

Moving Center Random Search With Two Parameters

can select a complex non-analytic performance function. This possibility might be used to great advantage. However, in the following discussion we use the standard criteria of minimum mean squared error.

To evaluate an estimate to the performance surface we use two filters in parallel. One uses a set of weights that are the current center, so that this filter produces the output, $y(k)$. The second set of filter weights correspond to a test point in the weight space. The filter output at the test point, $\tilde{y}(k)$, is used only during the adaptation process.

The performance surface estimates are a time average, which is the only reasonable estimate of the ensemble average that we can calculate on line, and are given by:

$$\hat{J}_\ell = \sum_{j=0}^{R-1} [y(k-j) - r(k-j)]^2 \quad (3.2-2)$$

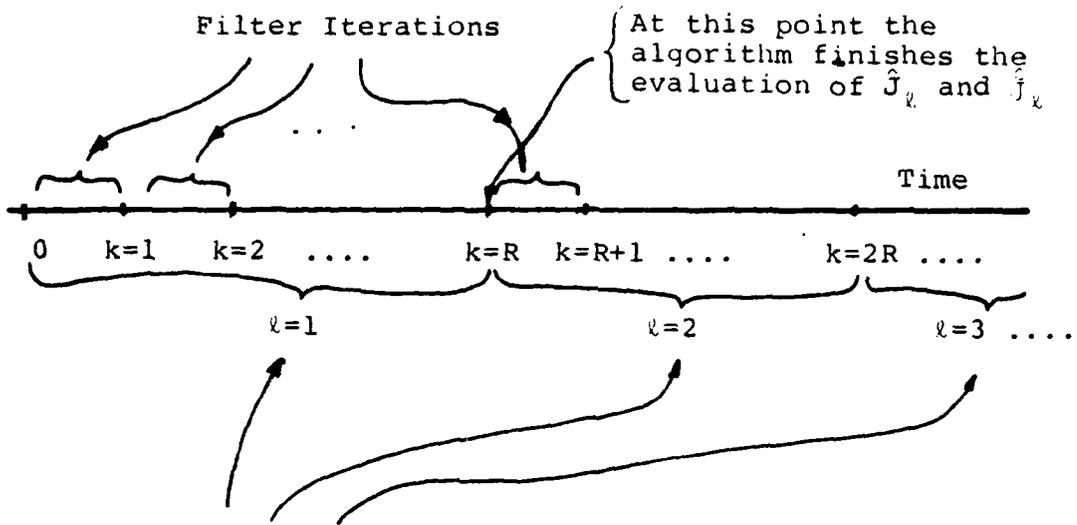
$$\hat{J}_\ell = \sum_{j=0}^{R-1} [\tilde{y}(k-j) - r(k-j)]^2 \quad (3.2-3)$$

where R is the number of input samples used to estimate the performance function for a given random search interval.

We have two types of iterations. First filter iterations which process each new input sample with a fix set of filter weights and produce the outputs $y(k)$ and $\tilde{y}(k)$. The second type of iteration involves the random search selection of a new set of filter parameters which occurs after R filter

iterations. For each random search interval a new set of parameters is selected and tested. Fig. 3.2-2 presents the relationship of filter iterations to the random search interval.

Fig. 3.2-3 presents a flow chart of the basic random search adaptive filter algorithm.



Random Search Intervals
 Each with a fix set of parameters.

Fig. 3.2-2

Filter Iterations And
 Random Search Intervals

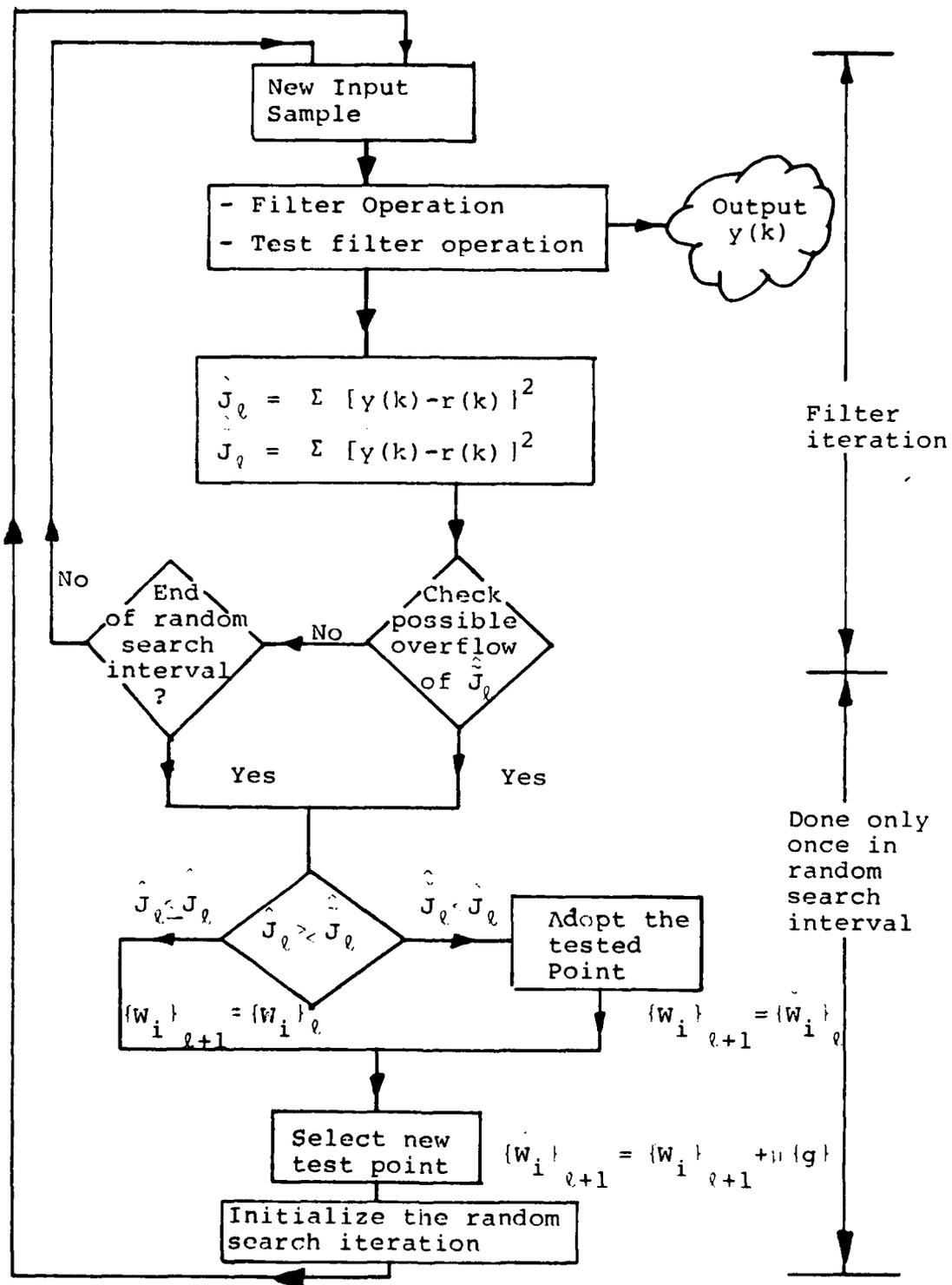


Fig. 3.2-3

The Basic Random Search Adaptive Filter Algorithm

3.3 OPERATION OF THE RANDOM SEARCH ALGORITHM

We now use the random search algorithm presented in Fig. 3.2-3 to implement an adaptive IIR filter. The filter operation is given by:

$$y(k) = \sum_{i=0}^{N_a-1} a_i(k)X(k-i) + \sum_{i=1}^{N_b} b_i(k)y(k-i) \quad (3.3-1)$$

The weights $\{a_i(k)\}$ and $\{b_i(k)\}$ are functions of time and their variation is controlled by the algorithm of Fig. 3.2-3 and equation (3.2-1).

Details of the simulation are given in Appendix A.

Fig. 3.3-1 presents the operation of a random search IIR filter with $N_a=3$, $N_b=2$, $\mu_a=.01$, $\mu_b=.1$.

We now discuss these results starting with some basic filtering considerations.

The poles and zeros of a filter should be located so that the desired spectral components pass through the filter and the unwanted components are rejected.

For an adaptive filter we also need to match the filter output amplitude to the reference signal amplitude, that is there is a gain factor which must be adjusted accurately in the adaptive filter.

The filter (3.1-1) has the following transfer function

$$H(z) = a_0 \frac{\prod_{i=1}^M (z-q_i)}{\prod_{i=1}^M (z-p_i)} \quad (3.3-2)$$

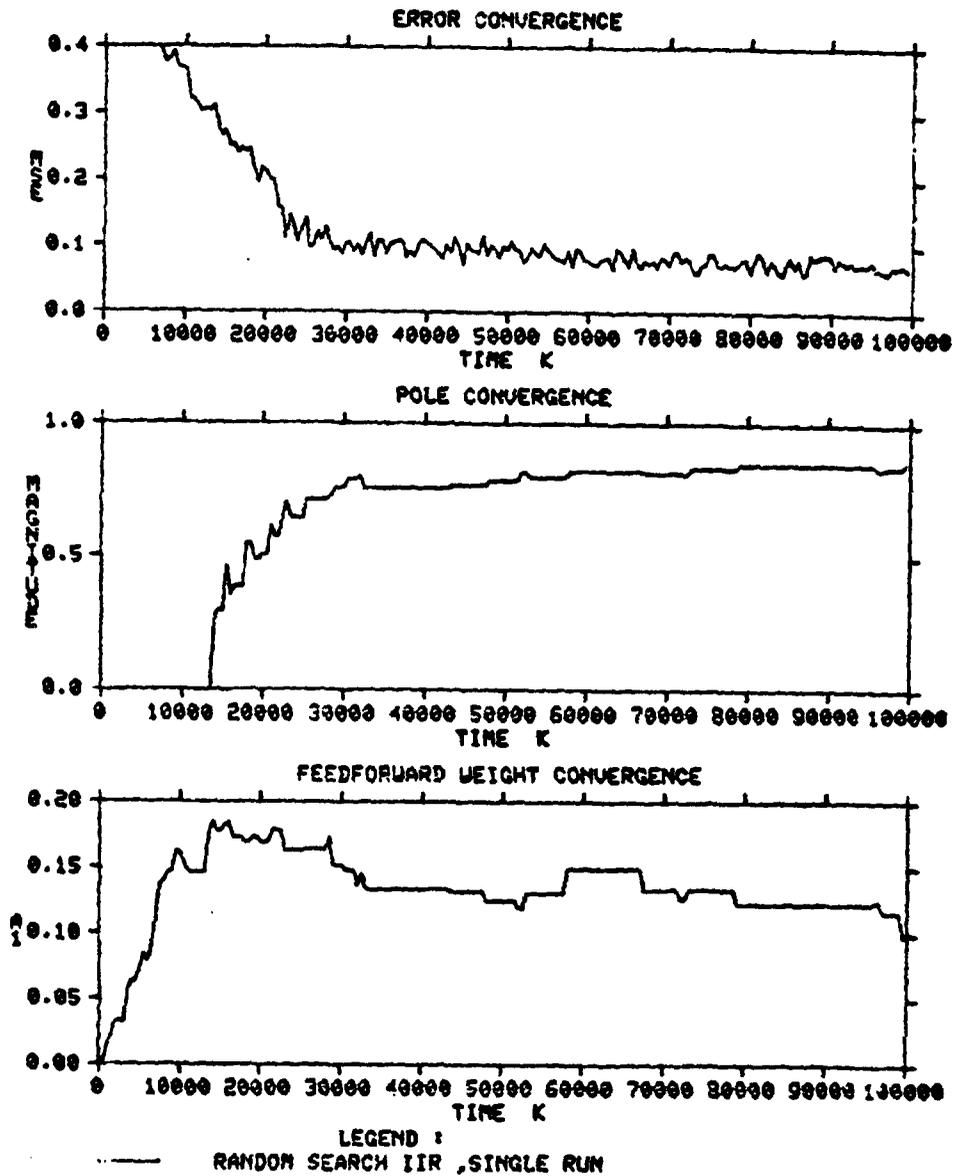


Fig. 3.3-1

Random Search IIR Filter Operation With $N_a=3$, $N_b=2$, $\mu_a=.01$,
 $\mu_b=.1$, $R=200$

where:

q_i are the filter zeros. There are M_z zeros.

p_i are the filter poles. There are M_p poles.

The parameter a_0 controls the gain of the filter. For the adaptive filter we can't use the concept of transfer function because the filter weights are time varying. However we can consider an average steady state transfer function with weights that are the mean of the time varying weights. Thus a_0 is required to match the filter maximum output magnitude to the reference amplitude. In order to enhance a desired spectral component the filter should have a pole (or poles) near the spectral component, close to the unit circle. The effect of this pole on a signal at the same frequency would be to multiply its amplitude by a gain factor of $(1/1-\rho)$ where ρ is the pole's magnitude. Thus the output of the filter at the pole frequency is given by $a_0/(1-\rho)$ times the magnitude of the input signal multiplied by a factor which depends upon the location of the other poles and zeros. If this output magnitude is to be equal to the reference signal amplitude, $a_0/(1-\rho)$ must have a specific accurate value because all the other factors that determine the output amplitude (namely the location of the other poles and the zeros) have only one optimal value, and thus for the steady-state near optimal filter are fixed.

For good selectivity ρ is only a little smaller than unity, and $a_0/(1-\rho)$ is the ratio of two very small numbers. It is difficult to achieve accuracy for this ratio with

random search adaptation on the forward weights (the a's) and the backwards weights (the b's or the poles).

As a result of the mismatch of filter output and reference signal amplitudes, decision mistakes occur when comparing the performance function estimates, $\hat{J}_k > \hat{J}_k$, with the result that a bad set of weights is sometimes chosen. These decision mistakes cause slower convergence and smaller value of the steady-state ρ (which means lower processing gain), and possible instability. This type of behavior was experienced in our simulation. One solution that we tried was to use smaller variance for the search on the feedforward weights. This approach turned out to be inferior to a new approach (which is presented in the next section) based on the use of gradient search on the feedforward weights and random search on the feedback weights.

3.4 RANDOM AND GRADIENT SEARCH (RGS)

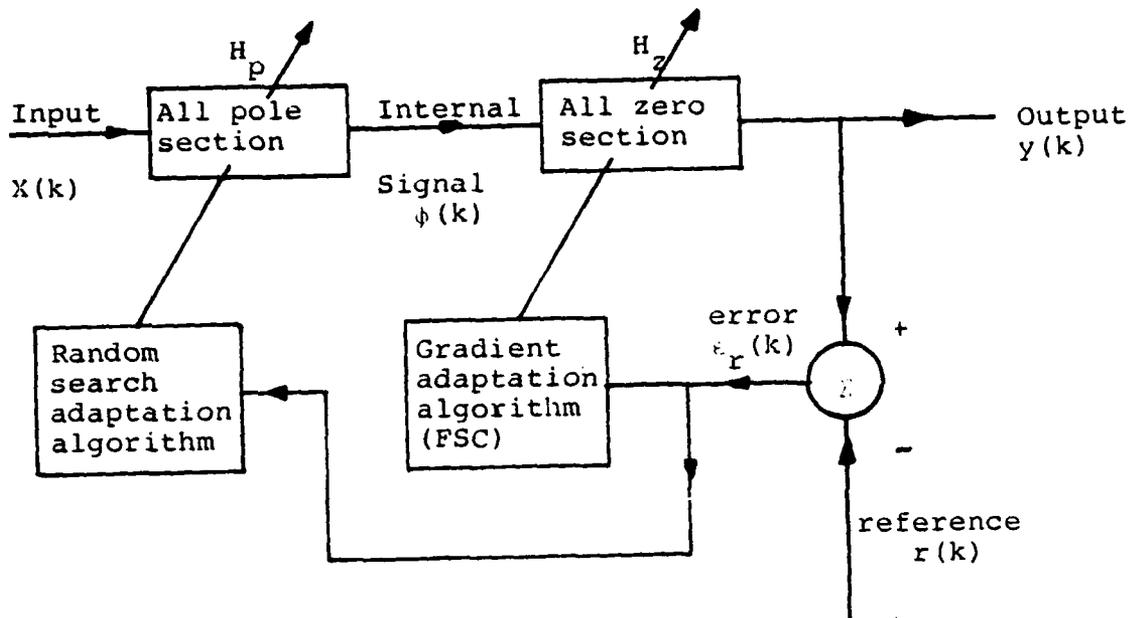
The fact that the IIR filter's performance surface is a quadratic function of the feedforward weights, as discussed in section 3.1, means that for non-varying feedback weights the performance surface with respect to the a's is unimodal. A unimodal surface may be handled best using a gradient method, and it is possible to achieve any desired accuracy to overcome the problem of the purely random search scheme discussed in section 3.3.

Widrow and McCool [5], have compared a random search technique for a FIR filter with the LMS algorithm. The random search technique used was tailored to the unimodal situation and, nonetheless, resulted in inferior performance compared with the LMS steepest decent gradient search.

The question is how to make the feedback weights converge first, so that the feedforward weights would then converge to the global minimum.

The cascaded arrangement, as shown in Fig. 3.4-1 is suggested. The all pole section comes first, and is adaptively controlled by a random search algorithm. A second all zeros section is then adaptively controlled by a gradient algorithm to produce, with suitable values of the adaptation gains, the desired effect of pole convergence followed by zero convergence.

The optimal values of the adaptation gains is a compromise of two contradicting considerations. The first factor is the requirement that the poles converge faster, and calls



Filter operation:

	H_p	H_z
$\phi(k) = x(k) + \sum_{i=1}^{N_b} b_i(k) \phi(k-i)$		$y(k) = \sum_{i=0}^{N_a-1} a_i(k) \phi(k-i)$
$H_p(z) \Big _{\text{average steady state}} = \frac{1}{1 - \sum_{i=1}^{N_b} b_i^{\omega} z^{-i}}$		$H_z(z) \Big _{\text{average steady state}} = \sum_{i=0}^{N_a-1} a_i^{\omega} z^{-i}$

where a_i^{ω} , b_i^{ω} are the steady state average value of the weights.

Composite filter response:

$H(z) \Big _{\text{average steady state}} = H_p(z) \cdot H_z(z)$		$= \frac{\sum_{i=0}^{N_a-1} a_i^{\omega} z^{-i}}{1 - \sum_{i=1}^{N_b} b_i^{\omega} z^{-i}}$
--	--	---

Fig. 3.4-1

The Random and Gradient Search (RGS) Filter Concept

for low μ_a . The second factor is a desire to maintain enough randomness in the weight adaptation process so that we have a reasonable probability of transition from a local minimum zone to the global minimum zone. The effect of the later factor depends upon the specific shape of the performance surface.

The internal signal $\phi(k)$ has, at least during the pole convergence, non-stationary characteristics. In particular, the magnitude variations of $\phi(k)$ are important to the adaptation operation in the all zero section. As discussed in section 2.6 the FSC algorithm has no dynamic range limitations so that it is ideally suited for the RGS algorithm.

The RGS algorithm is presented in three ways: Fig. 3.4-2 presents its flow diagram, Fig. 3.4-3 presents its block diagram, and Appendix B is a FORTRAN realization of the RGS algorithm. Fig. 3.4-4 presents typical operation of this algorithm and shows the convergence and steady-state operation of the filter. Some further analysis and more simulation results are included in the next section.

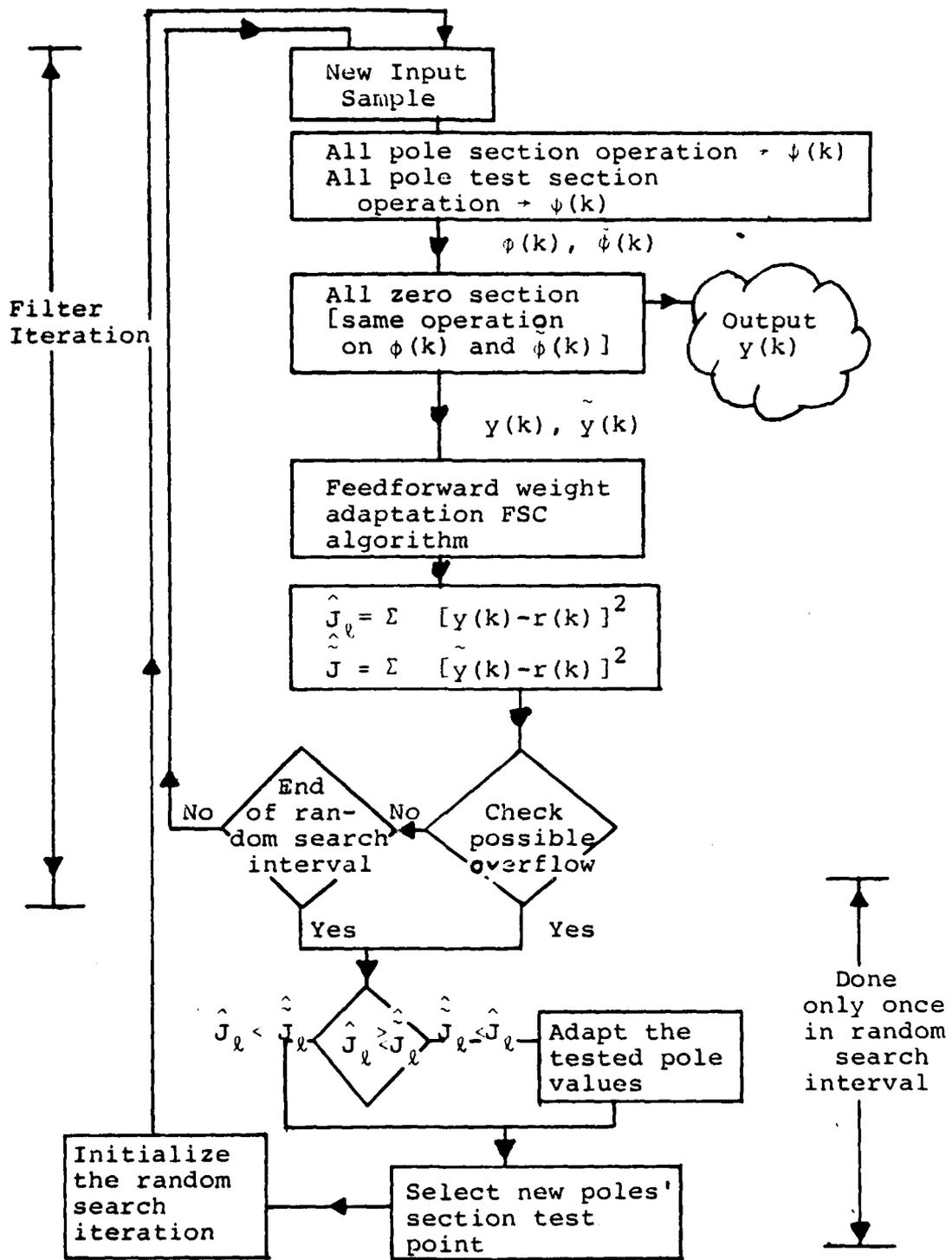


Fig. 3.4-2
RGS Flow Diagram

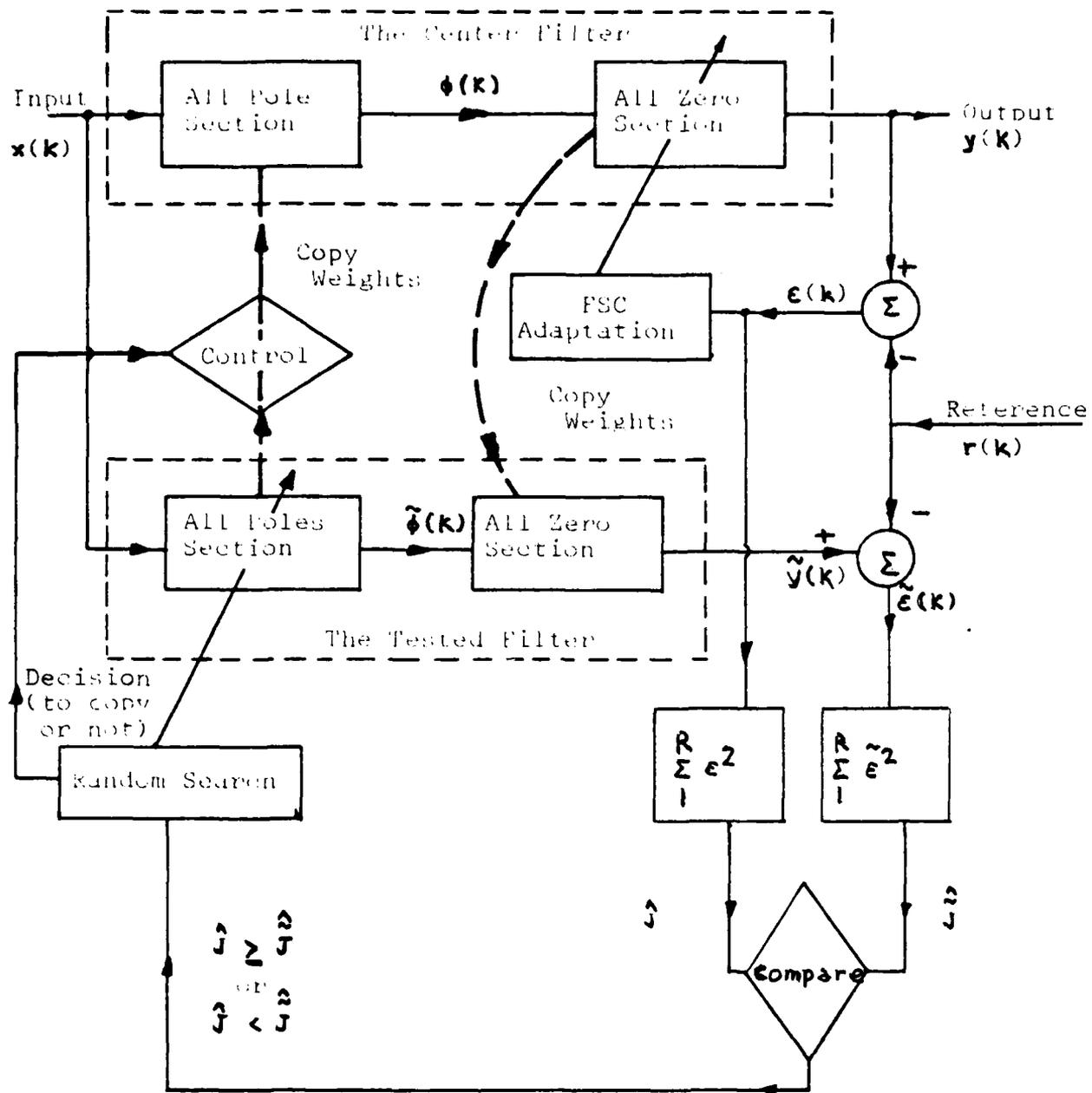


Fig. 3.4-3
RGS Block Diagram

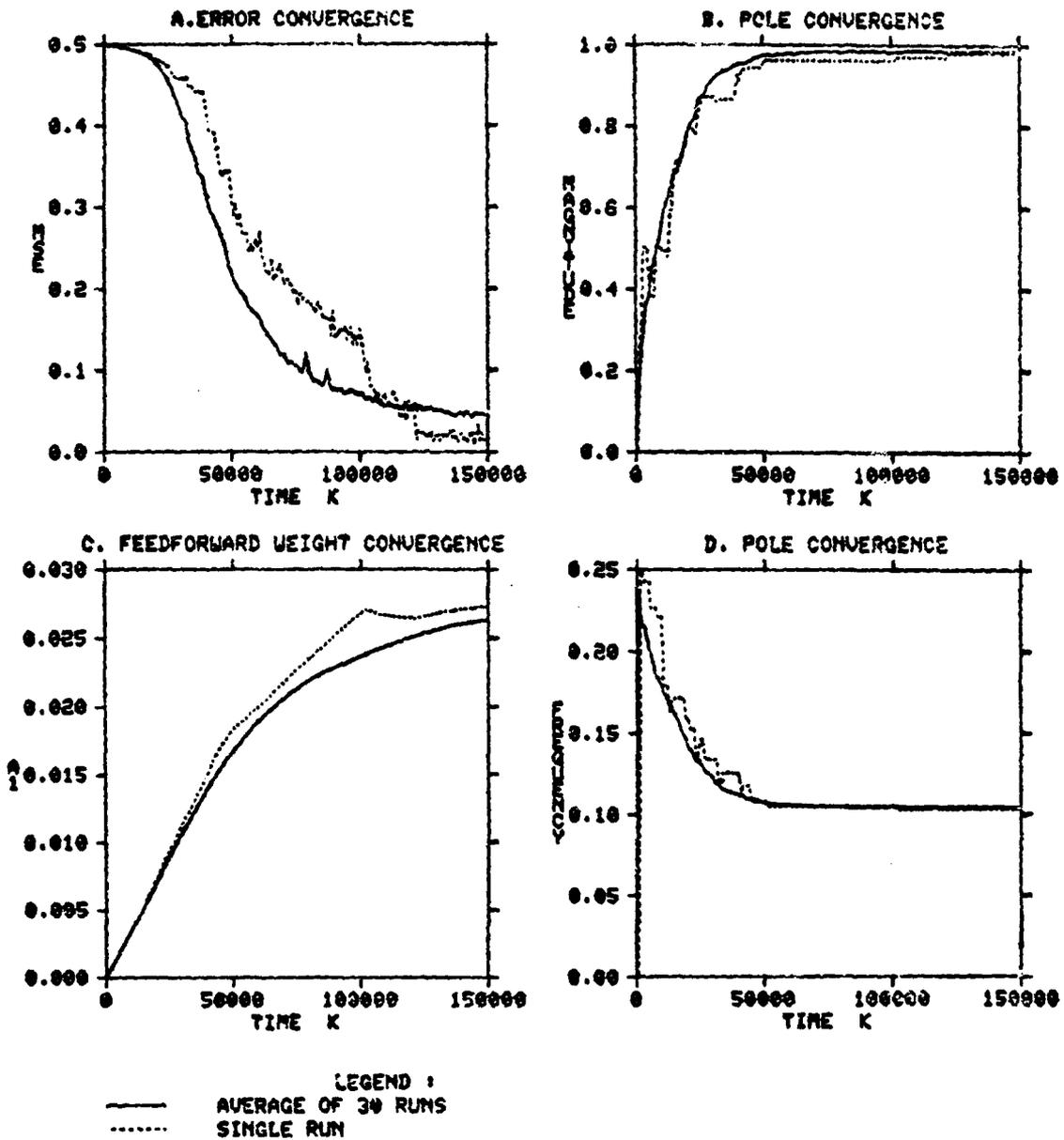


Fig. 3.4-4
 RGS Typical Operation. ALE example with $N_a=3$, $N_b=2$, $\mu_a=10^{-6}$,
 $\mu_b=.1$, $R=500$.

AD-A081 605

NAVAL POSTGRADUATE SCHOOL MONTEREY CA
EFFICIENT ADAPTIVE FIR AND IIR FILTERS. (U)
DEC 79 U BEN-YAKOV

F/6 17/5

UNCLASSIFIED

NL

2 of 2
3/2/80



END
DATE
FILMED
4-80
DTIC

3.5 CONVERGENCE OF THE RGS IIR FILTER

Convergence of the RGS IIR filter is composed of two processes: the random search on the feedback weights, and the gradient search on the feedforward weights. These processes are coupled through the cascade structure of the filter, the common error expression and the dependency of the feedforward weights on the poles magnitude, as discussed in section 3.2. In order to analyze this situation we first assume independent operation and analyze each of the sections separately. We then combine the convergence time estimates with a correction factor to account for the fact that both processes converge simultaneously.

Consider first the analysis of the random search algorithm used in the RGS IIR filter in a general environment. In the simple case of single parameter, W , define the convergence zone:

$$|W_{\ell} - W^*| \leq \Delta W \quad (3.5-1)$$

where:

W_{ℓ} is the parameter, W , at the ℓ th random search interval. W^* is the optimal value of W (the global minimum of the performance surface)

$\Delta W > 0$ is the limits of the convergence zone around W^* .

A test point is selected by:

$$\tilde{W}_{\ell} = W_{\ell} + \mu g \quad (3.5-2)$$

where:

\tilde{W}_ℓ is the test point in the random search's ℓ th interval.

$\mu > 0$ is convergence control parameter adaptation gain.

g is a number generated by $N(1,0)$ random number generator.

The probability density function of \tilde{W}_ℓ given W_ℓ is:

$$P(\tilde{W}_\ell/W_\ell) = \frac{1}{\sqrt{2\pi\mu}} \exp\left[-\frac{(\tilde{W}_\ell - W_\ell)^2}{2\mu^2}\right] \quad (3.5-3)$$

and the probability of selecting a test point in the convergence zone is:

$$P_\ell = P_r[|\tilde{W} - W^*| \leq \Delta W/W_\ell] = \int_{W^* - \Delta W}^{W^* + \Delta W} P(\tilde{W}_\ell/W_\ell) d\tilde{W}_\ell \quad (3.5-4)$$

Fig. 3.5-1 illustrates the situation and the probabilities defined above.

Selecting a correct value for \tilde{W}_ℓ is not enough. After the testing of this point we need a correct decision that the tested point is better than W_ℓ . Thus we have the estimates \hat{J}_ℓ and \hat{J}_ℓ , and the selection of the correct weight depends upon their comparison. The probability of a correct decision depends upon the values of J_ℓ and \tilde{J}_ℓ , their difference, and the estimation parameters, mainly R . To simplify the analysis we define $P_{CD}(\ell)$ as the probability of a correct decision given W_ℓ , averaged over all possible values of \tilde{W}_ℓ .

We can write the probability of convergence to the convergence zone (3.5-1), $\theta_{\ell+1}$ at the $(\ell+1)$ random search interval, given W_ℓ , as:

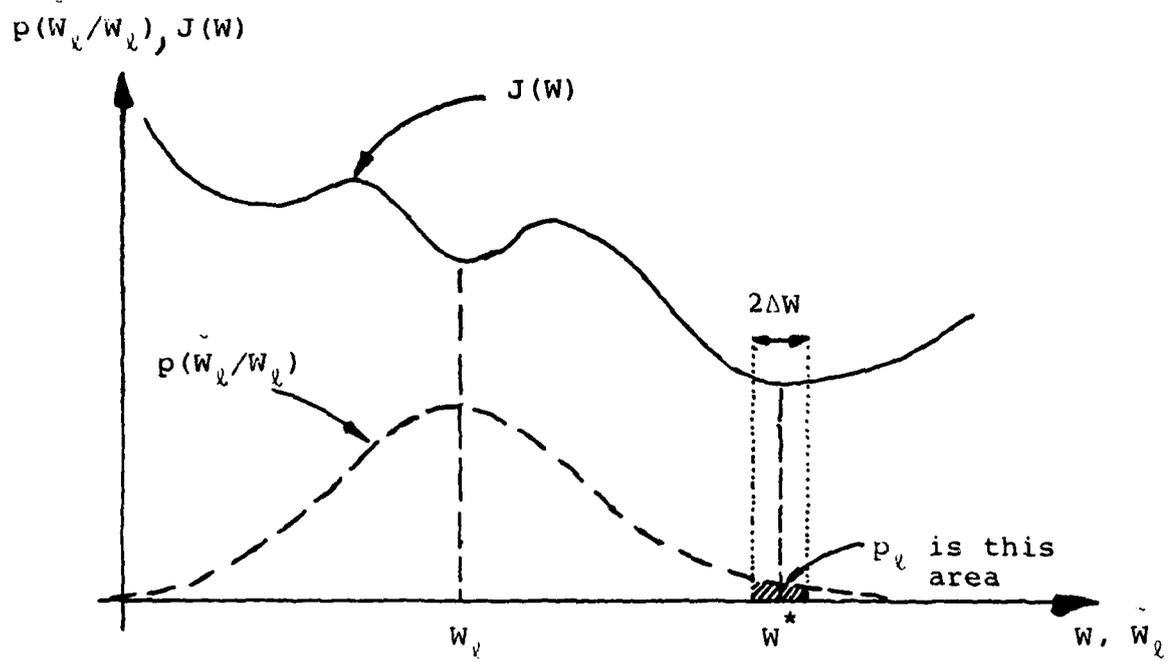


Fig. 3.5-1

The Probabilities Associated With
The Selection Of A New Test Point

$$\begin{aligned} \theta_{\ell+1} &= P_r[\text{selection of } W_\ell \text{ in the convergence zone}] \cdot P_r[\text{correct decision}] = \\ &= P_\ell \cdot P_{CD}(\ell) \end{aligned} \quad (3.5-5)$$

The probability, Q_ℓ , that the process does not converge in the first ℓ iterations is given by:

$$Q_\ell = P_r[\text{no convergence in the 1st R.S. interval}] P_r[\text{no conv. in sec. inter.}] \dots$$

$$\dots P_r[\text{no conv. in } \ell\text{th inter.}] = \prod_{i=1}^{\ell} [1-\theta_i] \quad (3.5-6)$$

The probability, P_ℓ , that the process does converge in the first ℓ iterations is given by:

$$P_\ell = 1 - Q_\ell = 1 - \prod_{i=1}^{\ell} [1-\theta_i] \quad (3.5-7)$$

So far we have discussed the single parameter case. It is now convenient to introduce the general case, namely M parameters. We can define the multidimensional convergence zone as given by:

$$\| \underline{W}_\ell - \underline{W}^* \| \leq \underline{\Delta W} \quad (3.5-8)$$

where:

\underline{W}_ℓ is a vector of M parameters, $W_{i,\ell}$ $i=1,\dots,M$.

\underline{W}^* is a vector of optimal values, W_i^* $i=1,\dots,M$.

$\underline{\Delta W}$ is a vector of deviations from the optimal values of the parameters defining the convergence zone.

$\| \quad \|$ is a norm defined on the parameter space.

The multidimensional version of (3.5-2) is given by:

$$\underline{\tilde{W}}_\ell = \underline{W}_\ell + \mu \underline{G} \quad (3.5-9)$$

where:

\underline{G} is a vector of M independent random numbers each of which is $N(1,0)$.

Because of the independence of the parameters a multidimensional version of (3.5-4) is given by:

$$\psi_\ell = \prod_{i=1}^M P_{i,\ell} \quad (3.5-10)$$

where $P_{i,\ell}$ is the single parameter probability given by 3.5-4.

The probability of convergence to the convergence zone at the $\ell+1$ iteration given \underline{W}_ℓ is given by:

$$\theta_{\ell+1} = \psi_\ell \cdot P_{CD}(\ell) \quad (3.5-11)$$

Equations (3.5-6) and (3.5-7) remain the same for the multi-parameter case. An essential property of any optimization algorithm is its ability to converge to the optimum. We will now prove that the random search algorithm used in the RGS filter, converges to the convergence zone defined in (3.5-8). To observe this point we examine equation (3.5-7). Since $(1-\theta_i)$ is a number always less than 1, the multiplication

$\prod_{i=1}^{\ell} (1-\theta_i)$ becomes smaller as ℓ increases. Thus

$$P_{\infty} = \lim_{\ell \rightarrow \infty} P_{\ell} = 1 - \lim_{\ell \rightarrow \infty} \left\{ \prod_{i=1}^{\ell} [1-\theta_i] \right\} = 1 - 0 = 1 \quad (3.5-12)$$

$P_{\infty} = 1$ means that after enough time the process converges to the convergence zone of the global minimum; that is, convergence with probability 1. Equation (3.5-12) does not provide quantitative information, namely an estimate of the convergence time. This problem is treated later in this section.

Fig. 3.5-2 presents the results of a parameter identification experiment, the details of which are presented in Appendix A. This example was taken from reference [11] where it was used to demonstrate how Feintuch's algorithm converges to a point on the performance surface which is not a minimum. Ref. [13] uses the same example to demonstrate how Stearns' algorithm converges to either a local or global minima depending upon the initialization point. Our results show that the RGS IIR filter converges to the global minimum even when started

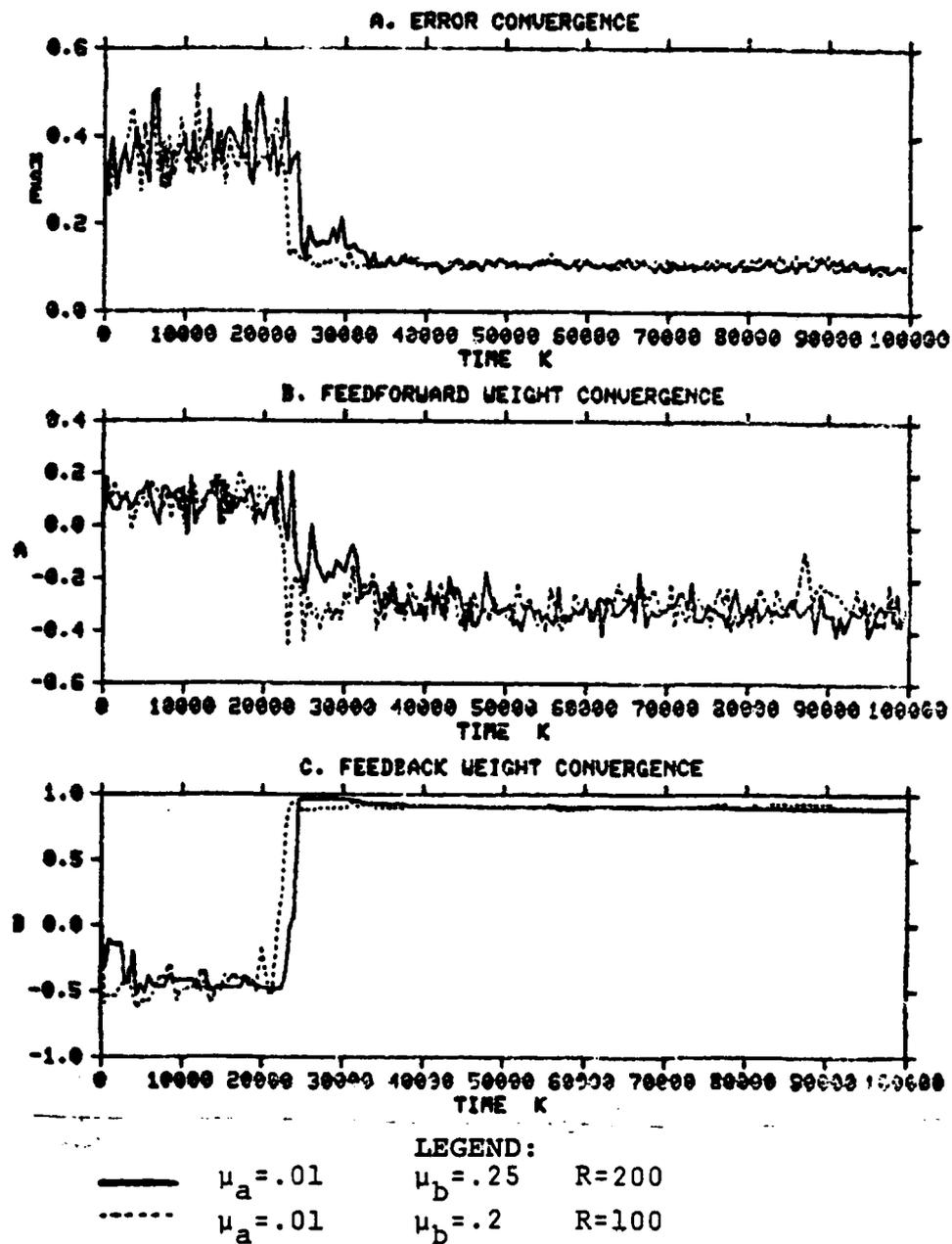


Fig. 3.5-2

RGS IIR filter transition from a local to the global minimum for a parameter identification example. The figure shows the results of two independent experimental runs. See Appendix A for details of the simulation.

in a local minimum point. Fig. 3.5-2 presents two experiments with typical convergence.

The adaptation of a random search process has two types of parameter changes, or steps:

- zone transitions - where after the movement, the center is near a new minimum
- small steps - where after the movement, the center is near the same minimum.

Fig. 3.5-3 illustrates the two step types. In signal filtering it turns out that the performance surface values at the local minima are typically much higher than at the global minimum, as shown in Fig. 3.5-4. This difference in the performance surface value means that the random search process, when comparing values of the performance surface estimations, is insensitive to the local minima. In terms of step types, we neglect the analytically complex zone transition steps and analyze the situation typical of signal processing applications, assuming convergence with small steps only.

We can define, for the general case of equation (3.5-9), the average step size, S_{av} , as:

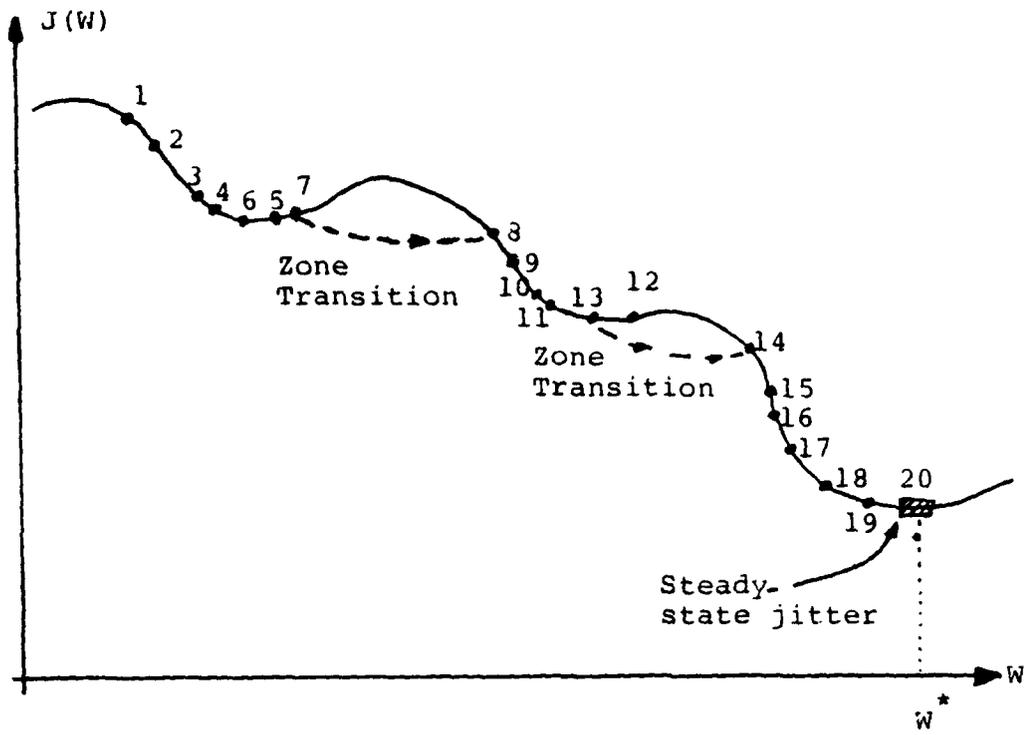
$$S_{av} = E[W_{i,l+1} - W_{i,l}] \quad (3.5-13)$$

where: $W_{i,l+1}$ and $W_{i,l}$ are the i th component of \underline{W}_{l+1} and \underline{W}_l respectively.

Using (3.5-2) as the equation for each component of (3.5-9)

we have

$$\tilde{W}_{i,l} - W_{i,l} = \mu g \quad (3.5-14)$$



Legend:

- -center value at the random search i th interval

Fig. 3.5-3

Random Search Convergence,
 Zone Transitions: 7 to 8, 13 to 14
 Small Steps: the Other Transitions

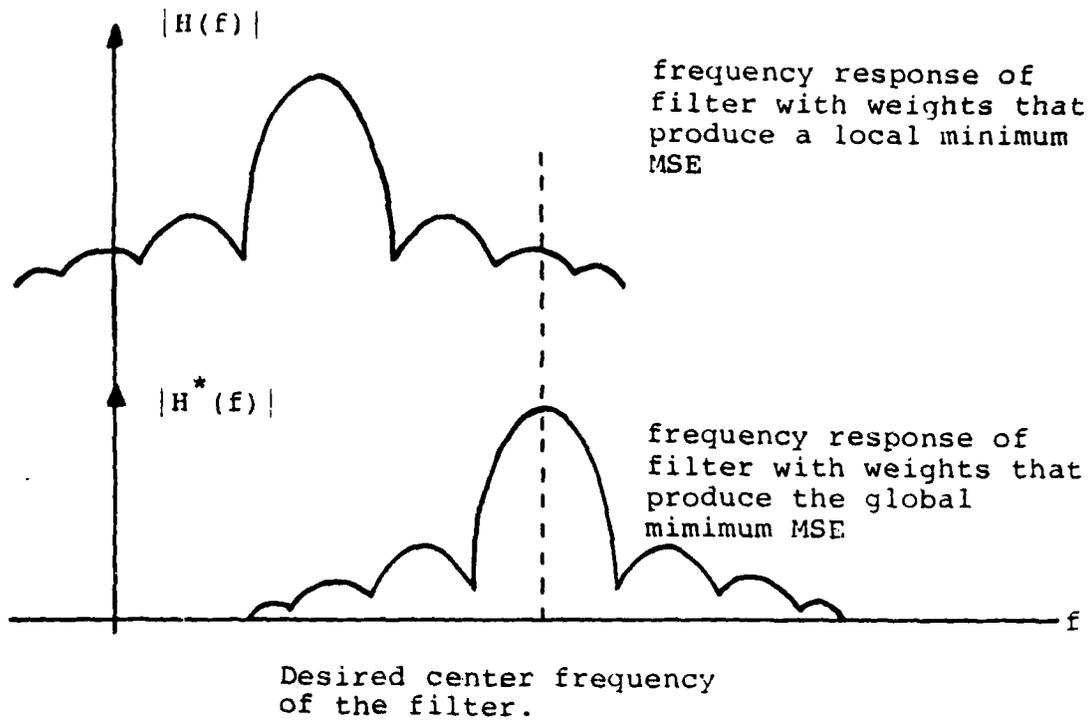


Fig. 3.5-4

Global and Local Minimum,
Signal Processing Example

Since in small step convergence the direction of convergence is always in the same direction towards W^* , on the average only half of the test points are accepted, that is those where $\mu g \geq 0$ (or those with $\mu g < 0$). This assumes no decision errors when comparing two points. Using the above reasoning, from (3.5-13) and the definition of $g=N(1,0)$ we get:

$$S_{av} = \int_0^{\infty} \mu g p(g) dg = \mu \int_0^{\infty} \frac{g}{\sqrt{2\pi}} e^{-g^2/2} dg = \frac{\mu}{\sqrt{2\pi}} \quad (3.5-15)$$

The above analysis ignores the coupling of the M parameters through the common error expression. At any interval one of the parameters is dominant, that is it contributes to the error term more than the others. As a result the correction of the value of this parameter dominates even though the changes in the values of the other parameters may be in the wrong direction.

Practically the value of a parameter can be expected to jitter around some value until this parameter becomes the dominant one. Then it's value would be corrected (and other parameters would jitter). If all the parameters were dominant for equal portions of the process the average step size would be $1/M$ of that given by (3.5-15), but parameters with larger numerical value get more attention, and the reduction in the average step size is given by:

$$S_{av}^D = \frac{1}{M^{\alpha_0}} \cdot \frac{\mu}{\sqrt{2\pi}} \quad (3.5-16)$$

S_{av}^D is the average step size for the dominant, large

valued, parameter. $0 < \alpha_0 < 1$ is unknown factor.

Assuming that there is a correct decision, the mean number of random search intervals needed for convergence is W_{\max}/S_{av}^D , where W_{\max} is the value of the largest parameter;

$$TC = \alpha_1' \frac{W_{\max}}{S_{\text{av}}^D} \cdot R \quad (3.5-17)$$

where: α_1' is a proportionality constant that depends upon the exact definition of TC (10%, e^{-1} of initial error, etc.)

TC in (3.5-17) is given in filter iterations. We can combine (3.5-16) and (3.5-17) and include $\sqrt{2\pi}$ in α_1 to yield:

$$TC = \alpha_1 \cdot \frac{W_{\max}}{\mu} \cdot M^{\alpha_0} \cdot R \quad (3.5-18)$$

Equation (3.5-18) provides a convergence time estimation for a general case of random search operation with the above assumptions. Let us turn now to the RGS IIR filter and get a specific expression for its' convergence time estimate. For the RGS IIR filter case equation (3.5-18), in terms of the algorithm parameters as defined in section 3.3, becomes

$$TC_b = \alpha_1 \frac{b_{\max}}{\mu_b} N_b^{\alpha_0} R \quad (3.5-19)$$

To get an estimate to b_{\max} we consider the transfer function of the filter given by equation (3.1-1):

$$H(z) = \frac{\sum_{i=0}^{N_a-1} a_i z^{-i}}{1 - \sum_{i=1}^{N_b} b_i z^{-i}} = \frac{\sum_{i=0}^{N_a-1} a_i z^{-i}}{\prod_{j=1}^{N_s} (1 + 2\rho_j \zeta_j z^{-1} - \rho_j^2 z^{-2})} \quad (3.5-20)$$

where:

N_s is the number of 2-order sections of the filter, $N_s = N_b/2$.

ρ_j is the magnitude of the j th pole

$\zeta_j = \cos(2\pi f/fs)$

f - pole frequency, f_s - the sampling frequency.

For a stable filter $\rho_j < 1$ and the largest possible b_{\max} is given by additions of the terms $2\rho_j \zeta_j$ which is the coefficient of z^{-1} when the multiplication of (3.5-20) is expanded

$$b_{\max} = \sum_{j=1}^{N_s} 2\rho_j \zeta_j < \sum_{j=1}^{N_s} 2 = 2N_s = 2(N_b/2) = N_b \quad (3.5-21)$$

Inserting (3.5-21) to (3.5-19) gives:

$$TC_b = \alpha_1 \frac{N_b^{\alpha_2}}{\mu_b} R \quad (3.5-22)$$

where:

$\alpha_2 = \alpha_0 + 1$ is expected to be in the range $1 < \alpha_2 < 2$.

Equation (3.5-22) estimates the convergence time of the feedback portion of the RGS IIR filter. To estimate the convergence time of the feedforward portion we can use the FSC relation (2.3-1). As discussed in section 3.3, the largest

value of a feedforward weight should be proportional to $(1-\rho)$

$$a_{\max} = \alpha_3(1-\rho) \quad (3.5-23)$$

where ρ is the pole's magnitude. In addition to the discussion of section 3.3 which relates the gain factor a_0 to the dominant pole magnitude ρ , it is also noted that the adaptation gain is also related to the pole magnitude because the smaller the adaptation gain the finer the control of the ratio $a_0/(1-\rho)$ and the closer the pole magnitude can be adjusted to the unit circle. Thus we write the following relationship:

$$(1-\rho) = \alpha_4 \mu_a^{\alpha_5} \quad (3.5-24)$$

where α_4 and α_5 are proportionality constants. Fig. 3.5-5 shows the transient and steady state values of the pole magnitude for several values of μ_a . The non-linear relationship between ρ and μ_a , as suggested by (3.5-24) seems to be reasonable.

Combining (2.3-1), (2.3-4), (3.5-23), (3.5-24) and value of $\alpha_1=1/2$ in (2.3-1) we get:

$$TC_a = \alpha \frac{a_{\max} \sqrt{N_a}}{\mu_a} = \alpha_6 \mu_a^{\alpha_7} \sqrt{N_a} \quad (3.5-25)$$

We now determine the convergence time for the RGS IIR filter by adding (3.5-25) and (3.5-22):

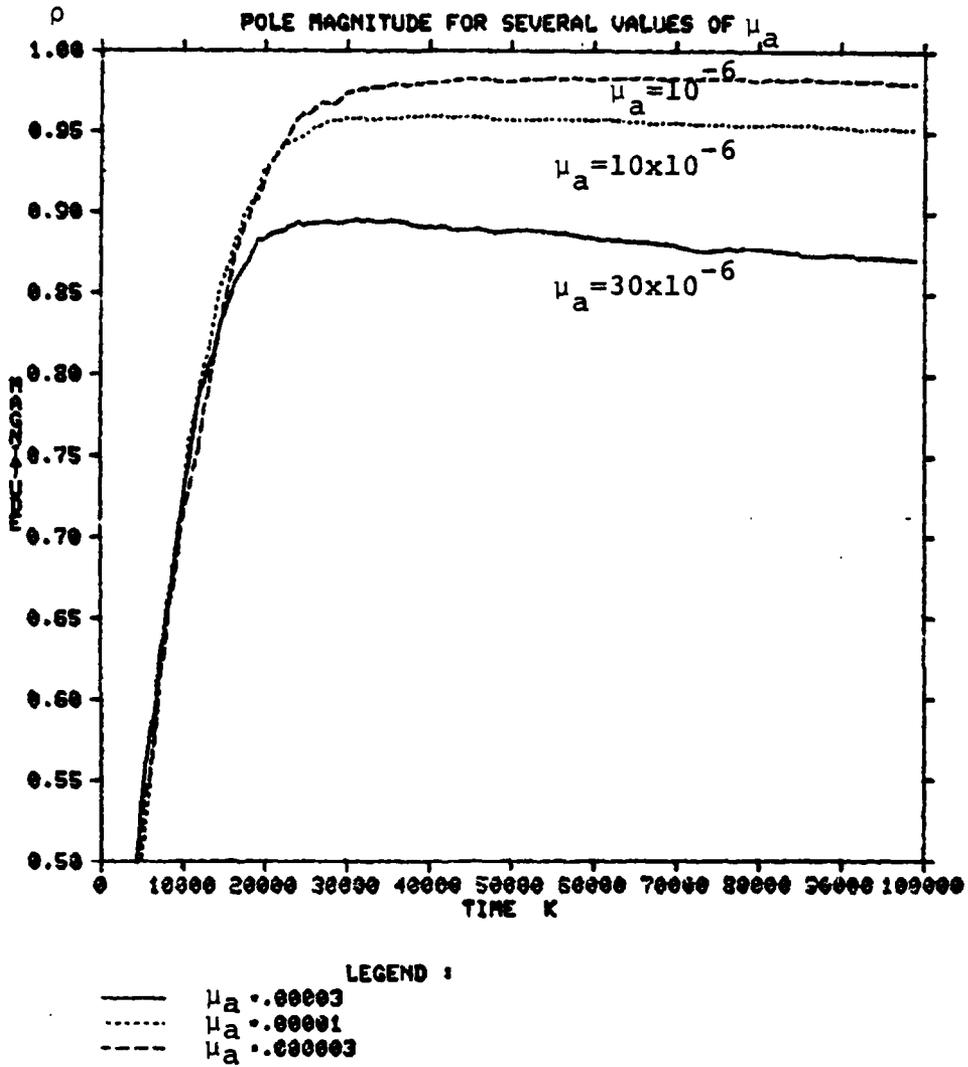


Fig. 3.5-5

Pole convergence and steady state value for several values of μ_a . Average of 25 runs, all the experiments were with $\mu_b = .1$ and $R=500$.

$\alpha_1, \alpha_2, \alpha_6, \alpha_7$ are experimentally evaluated constants. The values of α_1 and α_6 can be adjusted to compensate for the simultaneous convergence of the all pole and all zero sections of the RGS IIR filter.

A special difficulty encountered is to confirm the dependence of the random search convergence time upon the number of weights, N_b , in the random search process. A change in the number of feedback weights in the RGS IIR filter (with or without changing the input signal) causes major changes in the nature of the problem to be solved. For example if there are more poles than necessary, only one of them needs to converge, and the others are cancelled by the zeros. Any experiment in which the number of poles is varied, (with or without changing the signal) will combine the effects of changes in the nature of the problems to be solved, the effect of any changes in the input signal statistics, as well as the effect of more parameters upon the random search convergence time. A simpler approach is to construct a random search FIR filter and to use this filter to verify the analysis of the random search process and the dependence of its convergence upon the number of weights.

We start with relation (3.5-18) using FIR notation. Thus

$$TC = \alpha_1 \frac{\mu_a^{\max}}{\mu_a} \cdot N_a^{\alpha_0} R \quad (3.5-27)$$

From Treichler [37] we have:

$$a_{\max} = \frac{\text{SNR}}{1 + 0.5N_a \text{SNR}} \quad (3.5-28)$$

where SNR is the input signal to noise ratio.

Inserting (3.5-28) to (3.5-27) gives:

$$\text{TC} = \alpha_1 \frac{\text{SNR} \cdot N_a^{\alpha_0} \cdot R}{(1 + 0.5N_a \text{SNR}) \mu_a} \quad (3.5-29)$$

Fig. 3.5-6 presents a comparison of simulation results with convergence estimate (3.5-29) for several values of N_a , with α_0 and α_1 experimentally determined as $\alpha_0 = .4242$, $\alpha_1 = 1.557$. These results verify the ability of equation (3.5-18) to estimate the effect of the number of parameters on the convergence time of a random search filter.

An important assumption used in the convergence analysis of the RGS IIR filter was that the random search process does not make mistakes in the comparison of $\hat{J}_\ell > \hat{J}_\ell$. In order to define conditions for filter operation with no decision mistakes we investigate the effects of the random search interval, R , on the RGS filter performance. Fig. 3.5-7 presents operation with decision mistakes. This situation is typical in operation with relatively small random search interval. In the example of Fig. 3.5-7 we used $R = 100$.

For larger values of the random search interval, R , we have slower convergence as given by equation (3.5-26) and illustrated by the pole convergence of Fig. 3.5-8. Fig. 3.5-8

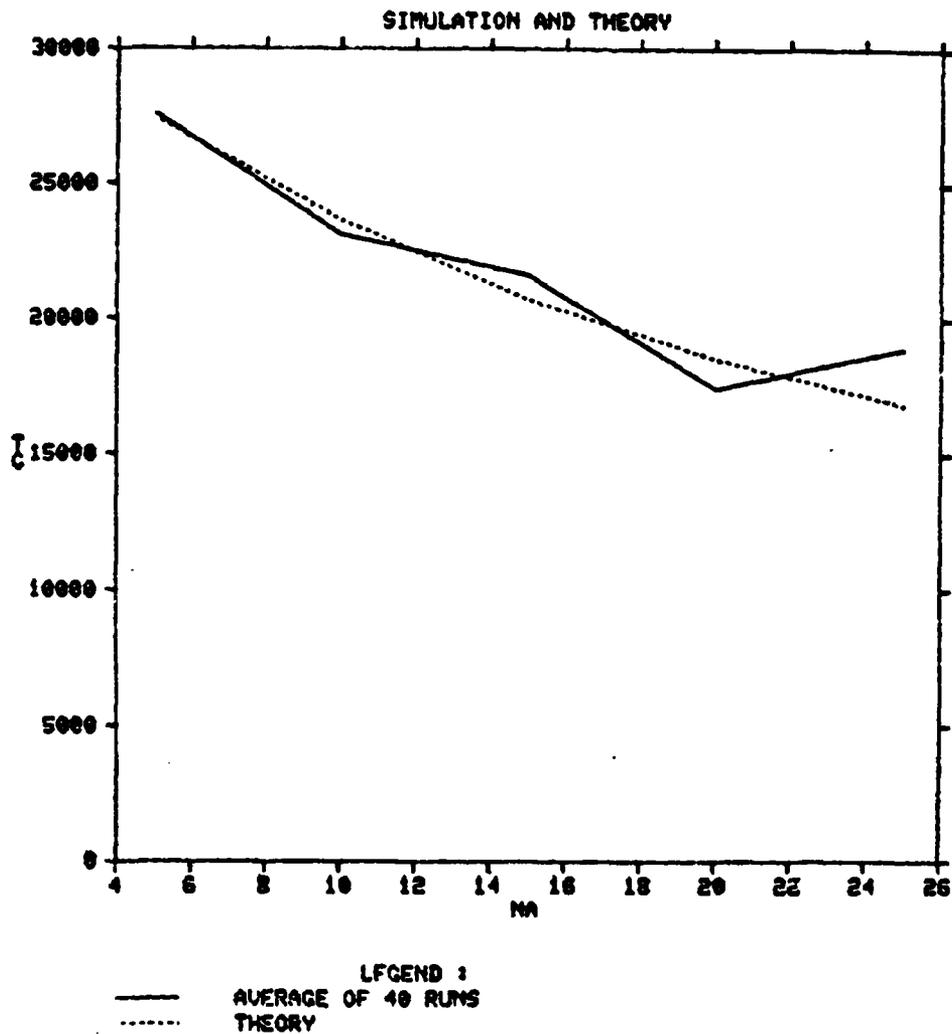


Fig. 3.5-6

Convergence time dependence upon the number of weight simulation and theory for a random search FIR filter. With $\mu_a = .0075$ $R=300$ for $SNR=-3$ dB.

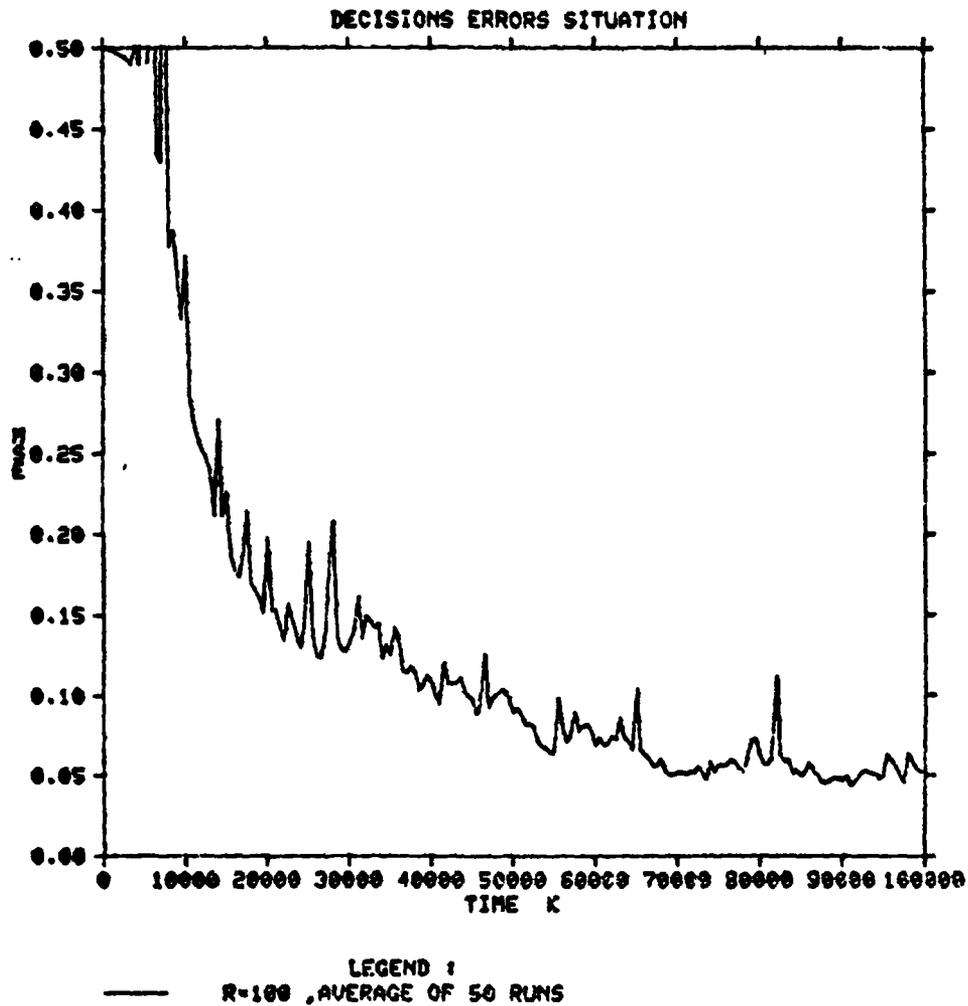


Fig. 3.5-7

RGS filter operation with decision mistakes. Filter parameters in this example are: $N_a=3$, $N_b=2$, $\mu_a=3 \times 10^{-6}$, $\mu_b=.1$, $R=100$.

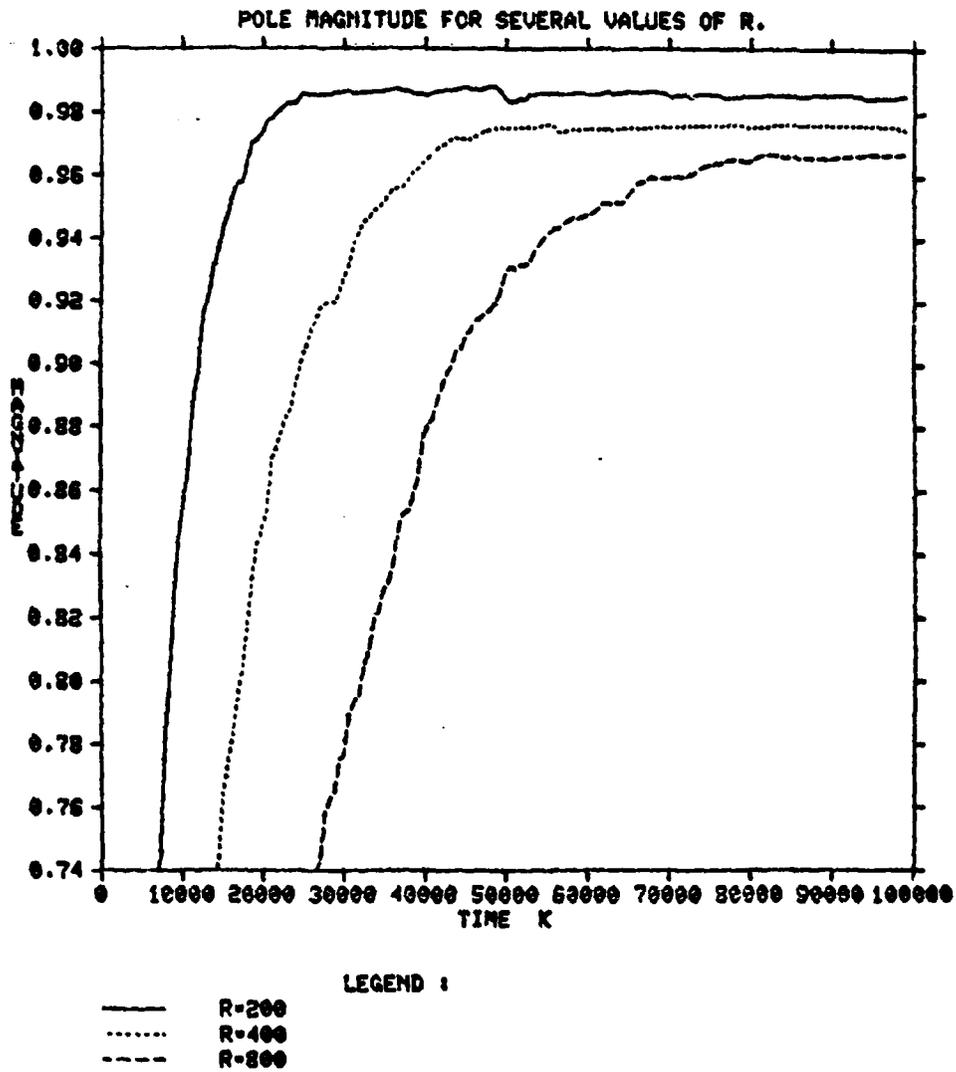


Fig. 3.5-8

Effects of random search interval, R , on pole convergence and pole's steady-state magnitude.

also shows that when the random search interval is too large ($R = 800$ in the example of Fig. 3.5-8) the pole magnitude is smaller and the resulting signal processing gain is lower. This unwanted effect is caused by decision mistakes that occur because with a long random search interval the feedforward weights provides better match for the weights in the center filter than for the tested filter for which the feedback feedforward weights are only copied (Fig. 3.4-3). This effect, of smaller pole magnitude, for long random search interval depends on the convergence of the all zero section and hence upon the value of μ_a .

The above discussion suggests that there is an optimal value for the random search interval, which depends upon the value of μ_a . From the results obtained in our experiments it seems that for μ_a in the range of 30×10^{-6} to 10^{-6} , the optimal values of R are in the range 300 to 500 iterations.

To verify the convergence time estimate of equation (3.5-26) we present experiments with several values of μ_a (Fig. 3.5-9) and several values of μ_b (Fig. 3.5-10); all of them with $R=500$ to assure that the operation is practically free from decision mistakes.

The effects of μ_a as discussed above are clear in Fig. 3.5-9:

- (1) The steady-state value of the pole magnitude is closer to 1 for smaller μ_a .
- (2) The convergence rate of the feedforward weight is proportional to μ_a .

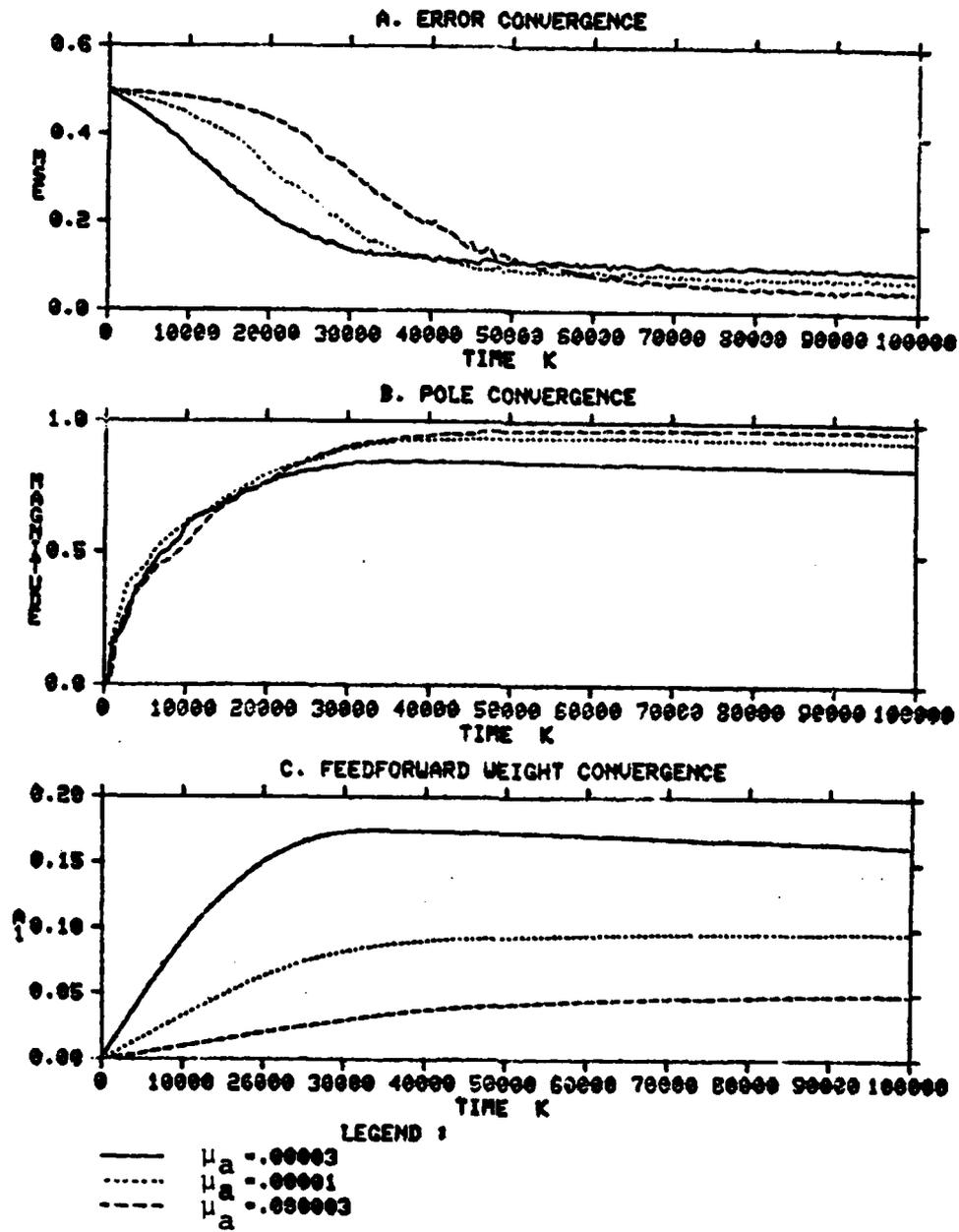


Fig. 3.5-9

RGS IIR filter operation for 3 values of μ_a . The results are simulation average of 32 runs with $\mu_b = .1$ and $R=500$.

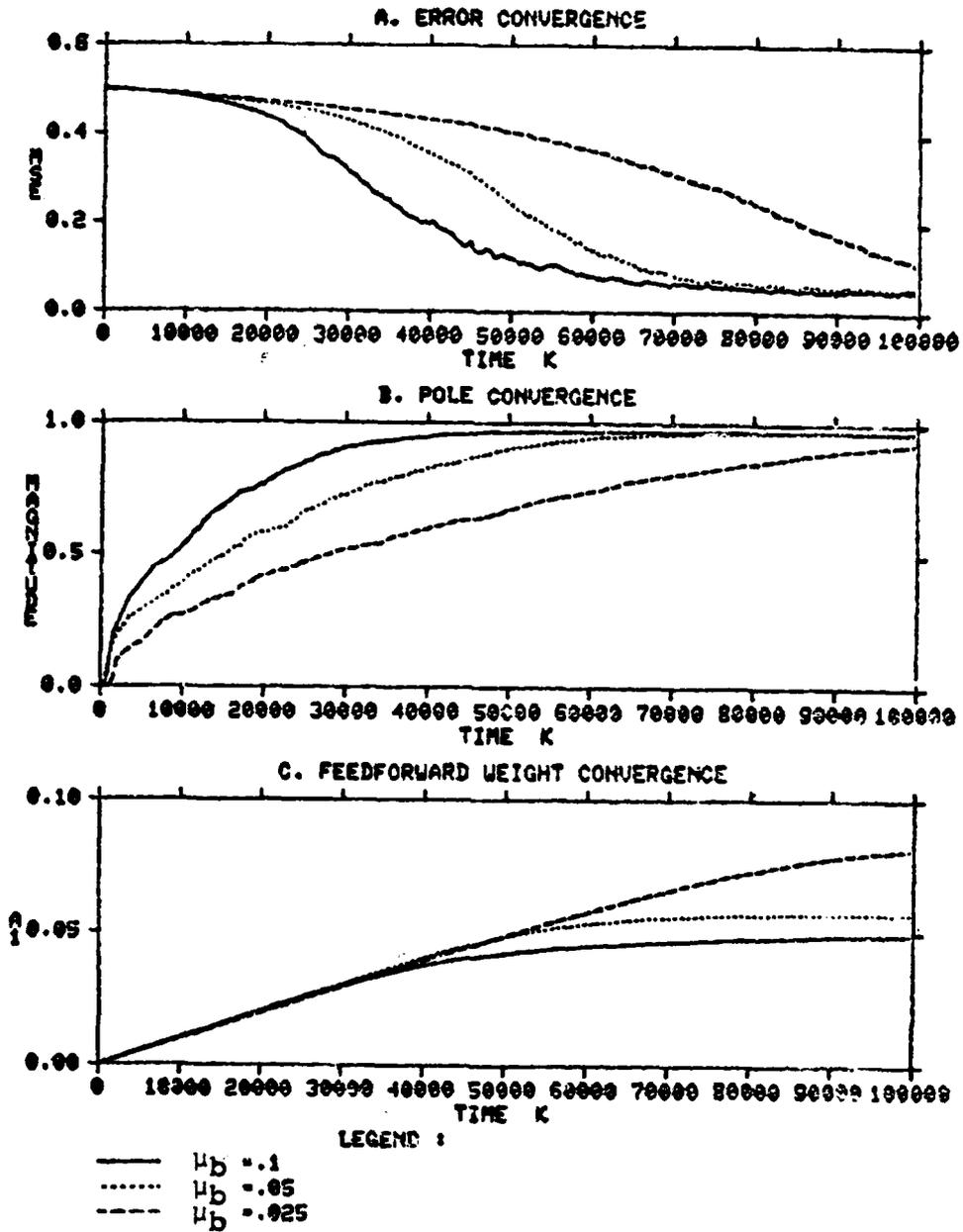


Fig. 3.5-10

RGS IIR filter operation for 3 values of μ_b . Simulation results are average of 32 runs with $\mu_a = 3 \times 10^{-6}$ and $R=500$.

- (3) The overall effect of the convergence and steady-state value of the output MSE is a combination of (1) and (2).

The effects of μ_b , as discussed above are clear in Fig. 3.5-10.

- (1) The convergence of the pole is proportional to μ_b .
- (2) The overall effect in the convergence of the output MSE is mainly the convergence rate.

It is interesting to note in Fig. 3.5-10 that the convergence rate of the feedforward weight is equal for all the values of μ_b .

Table 3.5-1 compares the convergence time measured in the experiments presented in Fig. 3.5-9 and Fig. 3.5-10 to the estimation given by (3.5-26) with experimentally determined proportionality constants. The modified estimation formula for $N_b=2$ is:

$$TC = 203.12\mu_a^{-.3392}\sqrt{N_a} + 3.8 R/\mu_b \quad (3.5-30)$$

Table 3.5-1 shows good agreement between experimental measurements of RGS filter convergence time and the estimations of (3.5-30). This agreement verifies the analysis of the RGS IIR filter convergence properties.

Table 3.5-1

RGS Convergence Time Measurement and Estimation

(Results are an average of 32 runs with $N_a=3$, $N_b=2$)

#	Results Presentation	Experiment Parameters			Convergence Time	
		μ_a	μ_b	R	Measured	Estimation (Eq.3.5-30)
1	Fig. 3.5-9	30×10^{-6}	.1	500	30,500	31,079
2	Fig. 3.5-9	10×10^{-6}	.1	500	41,000	36,517
3	Fig. 3.5-9 and Fig. 3.5-10	3×10^{-6}	.1	500	57,000	45,334
4	Fig. 3.5-10	3×10^{-6}	.05	500	66,000	64,372
5	Fig. 3.5-10*	3×10^{-6}	.025	500	100,000	102,448

NOTE

* Convergence time for #5 cannot be measured from Fig. 3.5-10 because only part of this experiment data is plotted here.

3.6 APRIORI STRUCTURED ADAPTIVE FILTERS (ASAP)

Motivated by the possibility to reduce the number of variables under random search adaptation, we now investigate the relationship of the filter's weights to a smaller set of variables. In some cases the structure of the optimal filter is known and only a few parameters are unknown and need to be evaluated. In other cases we might accept a sub-optimal simpler solution, namely an optimal filter with structural constraint. For the filter (3.1-1) we might have a smaller set of parameters, \underline{W} , such that:

$$a_i = f_i(\underline{W}) \quad i=0,1,\dots,N_a-1 \quad (3.6-1)$$

$$b_i = g_i(\underline{W}) \quad i=1,2,\dots,N_b \quad (3.6-2)$$

where: $f_i(\cdot)$ and $g_i(\cdot)$ are functions that connect each filter weight to the parameter vector \underline{W} .

Since we have a good solution to the feedforward weight adaptation, it is acceptable to use the same combined random and gradient search method that was used for the RGS IIR filter, for the proposed ASAF filter as presented in Fig. 3.6-1.

We will continue the discussion by considering the specific case of a pole close to the unit circle with adaptation of it's frequency only, see Fig. 3.6-2.

This type of adaptive filter is useful for the ALE

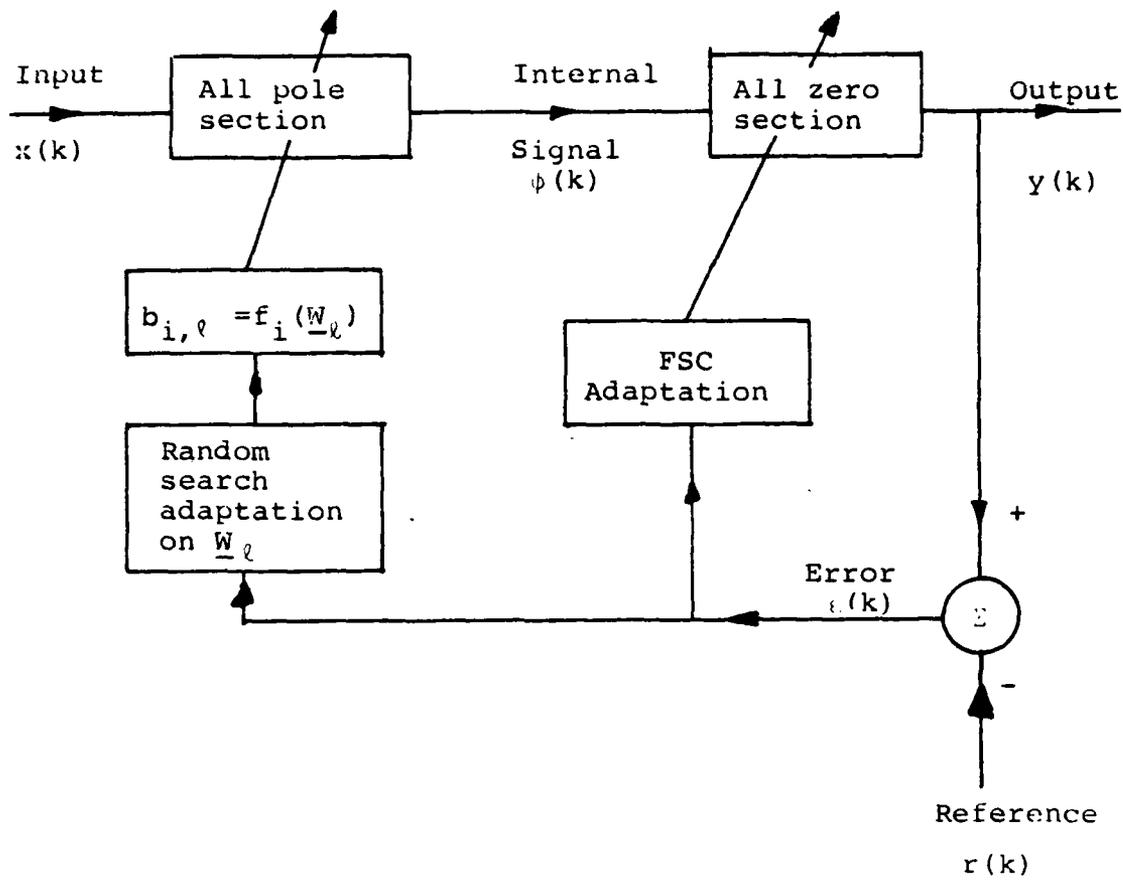
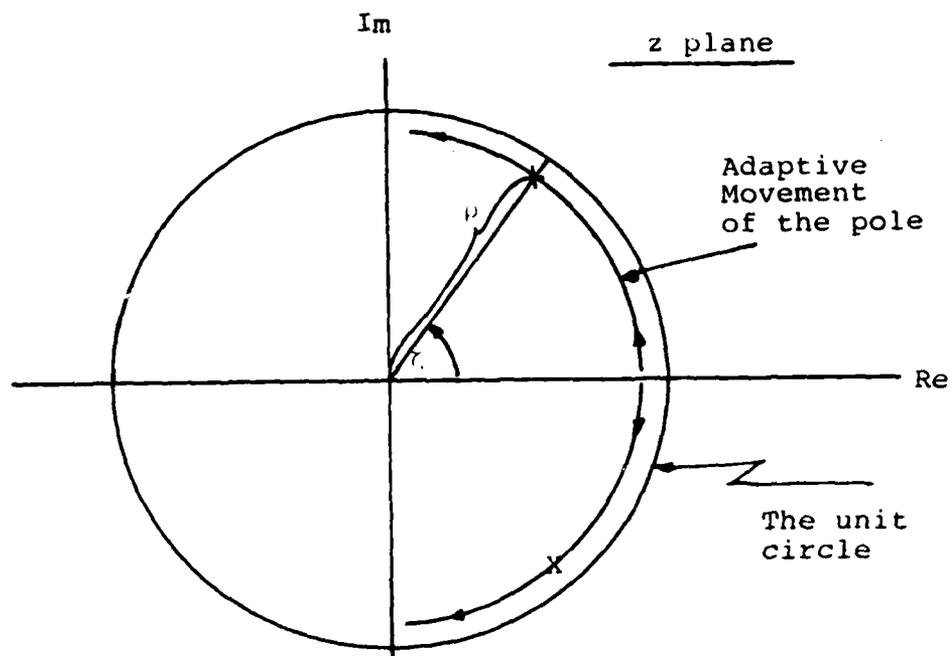


Fig. 3.6-1

Apriori Structured
Adaptive Filter (ASAF)



$$W = \rho = \cos(2\pi f/f_s)$$

f is the pole frequency

f_s is the sampling frequency

$$b_1 = 2\rho$$

$$b_2 = -\rho^2$$

Fig. 3.6-2

Apriori Structured Pole
Filter zeros are not shown.

application, or when several such sections are cascaded to enhance a multiple sine wave signal.

Fig. 3.6-3 presents a typical operation of an Apriori Structured pole, ASPOL, IIR adaptive filter with pole magnitude $\rho = .99$. Simulation details are presented in Appendix A. It is clear from Fig. 3.6-3 that this filter has high processing gain with relatively short convergence time. These advantages are also shown in Fig. 1.5-1.

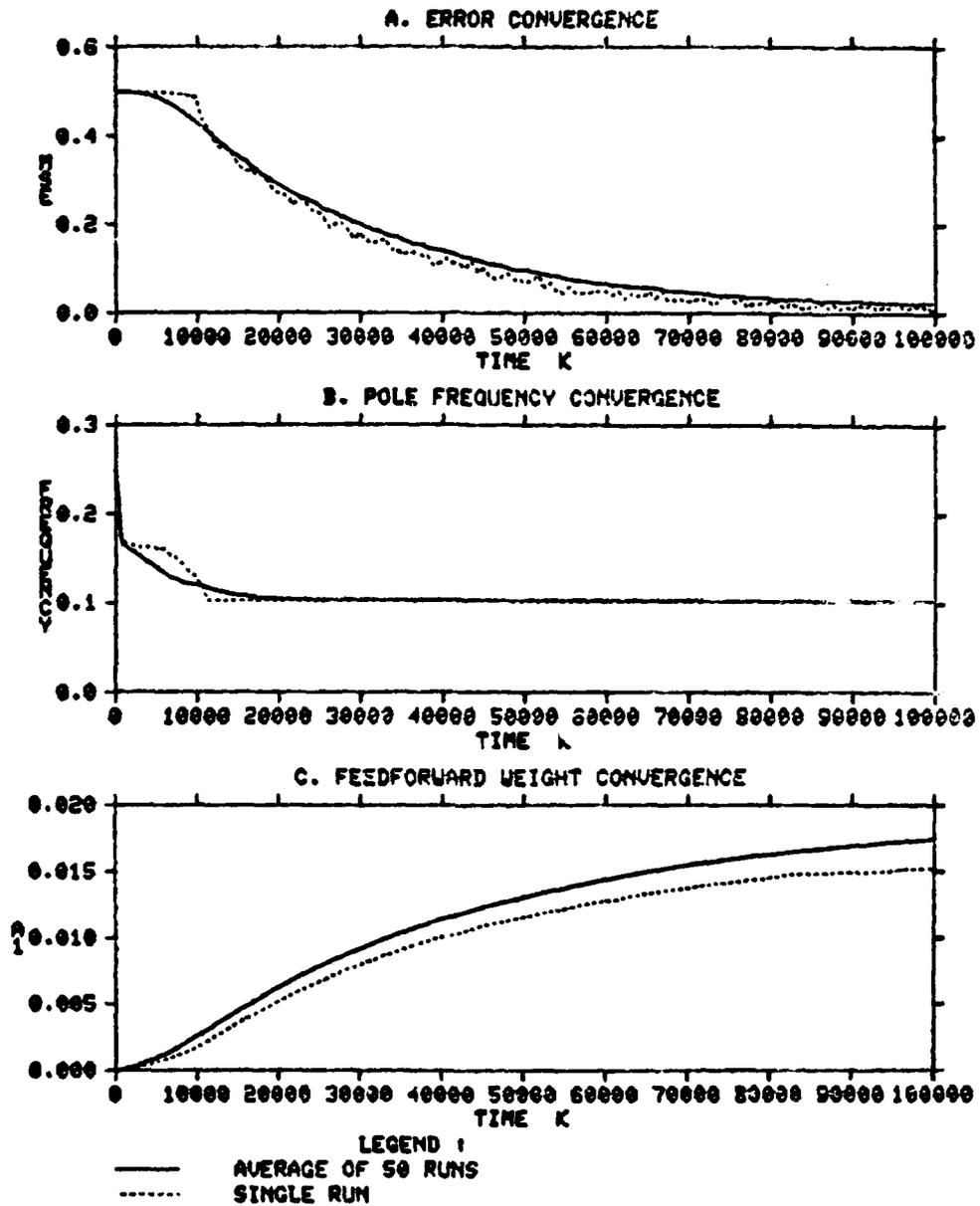


Fig. 3.4-3

Apriori Structured Pole (ASPOL) typical operation with $N_a=3$, $N_b=2$, $\mu_a=10^{-6}$, $\mu_b=.05$ and $R=500$.

3.7 CONCLUSION

An IIR filter has, in many cases, great advantages over a FIR filter because of the efficiency associated with the use of recursion and the existence of poles in the transfer function. However, realization of an adaptive IIR filter is a difficult task because of the multimodal nature of the IIR filter's performance surface, as well as the stability problem and the complexity of the performance surface gradient expression.

The proposed Random and Gradient Search (RGS) algorithm overcomes the complexity of the multimodal performance surface and converges to the global optimum with probability 1. This convergence is guaranteed for sufficiently large time. For the important special case of large difference in the value of the performance function between the global minimum and local minima, an estimation for the average convergence time has been derived and verified by simulation results.

The convergence time estimate is given by:

$$TC = \alpha_6 \mu_a^{\alpha_7} \sqrt{N_a} + \alpha_1 \cdot \frac{N_b^{\alpha_2}}{\mu_b} \cdot R \quad (3.7-1)$$

where $\alpha_6, \alpha_7, \alpha_1, \alpha_2$ are experimentally evaluated constants.

Stable operation of the RGS IIR filter was demonstrated in many hours of computer simulation without overflow problems. This stability is attributed to the detection of excessive MSE at the tested points, and the fact that the algorithm discards such points before overflow, which is caused by unstable filter weights, can be developed. Thus

the RGS is a practical candidate to realize an adaptive IIR filter.

The Apriori Structured Adaptive Filter (ASAF) uses the random search method and additional structure information to improve adaptive IIR filter performance. The moving pole example, for instance, is guaranteed to be stable and has fewer parameters in the adaptation process.

IV. SUMMARY

Two approaches to efficient adaptive filtering have been investigated; a FIR filter with simplified gradient estimation methods, and IIR filters with a combined random search and gradient adaptation scheme.

Two simplified algorithms, the Fixed Step Correction (FSC) and the Simplified LMS (SLMS), are derived and compared to the classical LMS algorithm for the FIR filter. The comparison includes analysis of filter properties, and extensive simulation results are presented to verify the analysis.

Because the adaptive filter properties depend upon the statistics of the input signal, the desired signal, and the reference signal, when analyzing the operation of an adaptive filter one must assume some statistics for the above signals. Hence, the analysis is valid for a specific case or a class of cases.

Thus algorithm comparisons and the adaptive filter properties analysis has been carried out here for the application of the adaptive line enhancement (ALE). The analysis includes convergence time estimate, steady-state misadjustment, and filter processing gain. Estimates to these properties have been derived and verified by simulation results which compare the three algorithms (LMS, FSC, SLMS). The conclusion of the comparison is that for equal filter order the LMS

algorithm is somewhat better. However when one considers an equal complexity, which allows the use of a higher degree filter for the simplified gradient estimations, the result is that the FSC and the SLMS are better than the classical LMS.

The IIR filter offers, in many cases, computational savings over a FIR filter. However, the IIR filter has a multimodal performance surface and may be unstable. Because of these two problems, and the complexity of the gradient expression, the algorithms which have been proposed for the IIR adaptive filter by Feintuch and Stearns do not provide a satisfactory solution. The Random and Gradient Search algorithm (RGS) proposed here has the ability to converge to the global minimum of the multimodal performance surface, and convergence with probability one is guaranteed for sufficiently large time. For the important class of cases characterized by a global minimum much lower than the local minimum, an average convergence time estimate has been derived and verified with simulation results. The use of structure information of the optimal solution when known allows the construction of an Apriori Structured Adaptive Filter (ASAF). This version of the RGS IIR filter optimizes a smaller set of parameters and is advantageous in some practical applications. In summary this research has demonstrated that with the RGS scheme it is possible to realize an adaptive IIR filter which will operate properly and have a practical implementation.

At the end of this dissertation it is appropriate to indicate some of the subjects that call for further work: In the area of FIR adaptive filters with simplified gradient estimation, some topics are:

- Effects of finite arithmetic.
- The study of variations of convergence and steady-state behavior around the mean.
- The dependence of convergence and steady-state behavior on input signal to noise ratio.
- Operation with complicated signals.
- Extension of the discrete algorithms developed here to analog systems, including adaptive antenna arrays.
- Consideration of non-stationary input signals.
- Applications.

In the area of RGS IIR filters, topics include:

- The study of random search decision error dependence upon filter parameters.
- Analysis of possible processing gain dependence upon filter parameters.
- The effects of operation with an inaccurate random number generator.
- Possible configurations for a priori structured adaptive filters.
- Operation with complicated signals.
- Consideration of non-stationary input signals.
- Applications

APPENDIX A

SIMULATION

A.1 THE SIMULATION METHOD

Simulation was used to provide experimental data for comparison of adaptive filtering algorithms, and for verification of analytic formulas. The simulation program includes four basic functions:

(1) Execution initialization: Signal parameters (frequency, signal to noise ratio, power,..etc.) and filter parameters (filter type, number of weights, adaptation gain, ..., etc.) are loaded interactively into the computer to control the forthcoming execution.

(2) Experiment configuration and signals generation: A signal generator subroutine, determined by the signal parameters loaded in the execution initialization, prepares sequences of 100 samples of input and reference signals to be processed by a filter subroutine. The relationship of the input and reference samples determines the adaptive filter simulation to be evaluated, i.e. ALE or parameter identification.

(3) The filtering function: As controlled by the filter parameters loaded during the execution initialization, a filter subroutine is called upon to process the data in blocks of 100 samples.

(4) Experiment data extraction and storage: The program stores the values of $J(k)$ and 200 values (spread equally

over the time of the experiment) of up to six parameters, and at the end of the experiment the program evaluates the processing gain and the convergence time for that experiment. The foregoing information, together with the experiment parameters, are stored on disk files and are available for off-line use.

A simplified flow diagram of the simulation program is presented in Fig. A.1-1. It should be noted that although the FIR and IIR simulation programs have identical structure there are some differences as discussed in sections A.2 and A.3.

A data handling program is used to access the data files and present experimental results in tabular or graphic forms. The graphic option includes plots of variables as function of time and as functions of a filter or a signal parameters.

The simulation was done on a PDP-11/50 minicomputer under RSX-11M multiuser operating system.

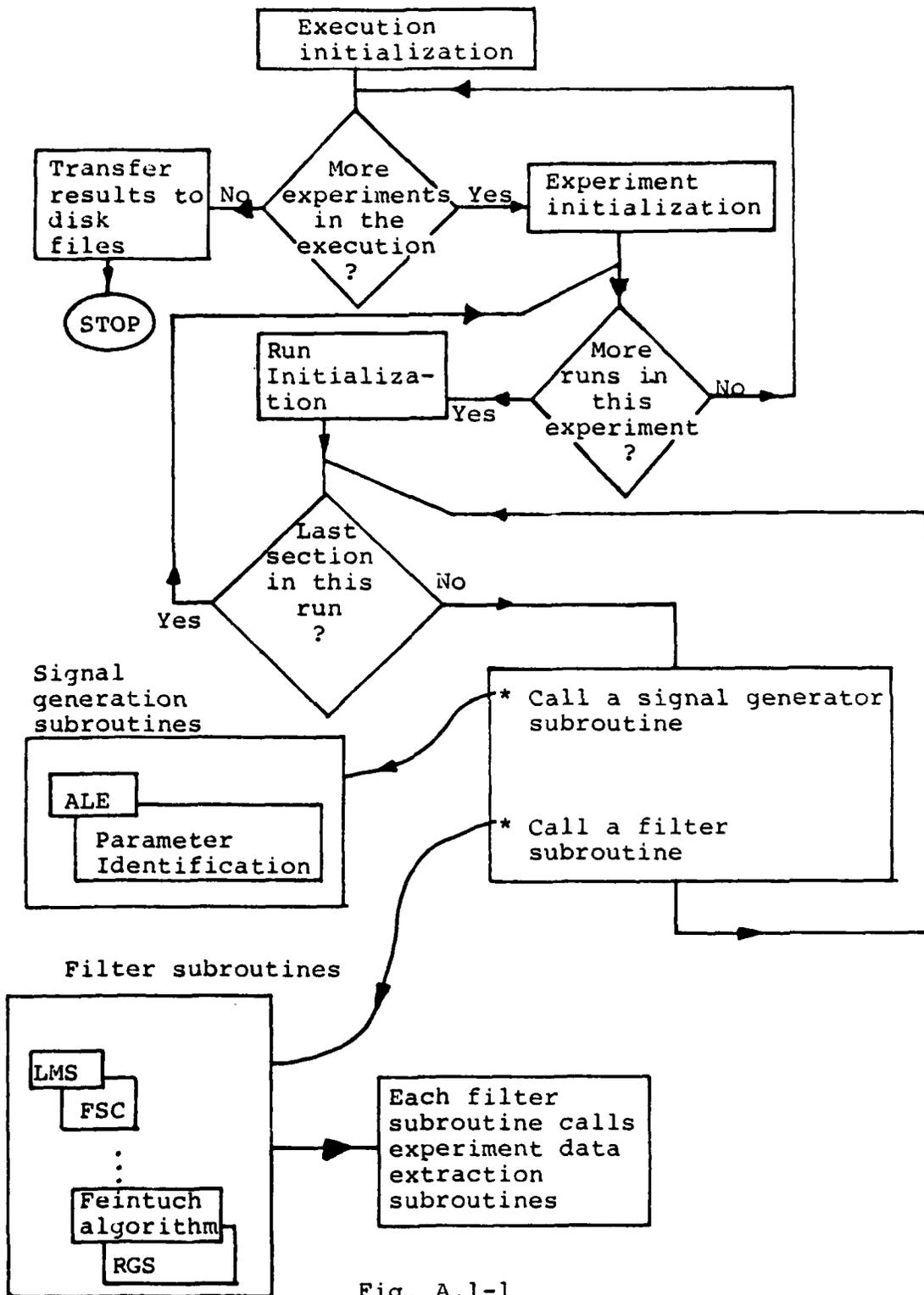


Fig. A.1-1

Simulation Simplified Flow Diagram

A.2 FIR SIMULATION PROGRAM

The FIR simulation uses the Adaptive Line Enhancement (ALE) configuration as presented in Fig. A.2-1. The desired signal is:

$$s(k) = \sqrt{2R_{ss}(0)} \cos \omega k \quad (\text{A.2-1})$$

where $R_{ss}(0)$ is the desired signal power.

And:

$$x(k) = s(k) + n(k)$$

where $n(k)$ is a white gaussian noise with variance $R_{nn}(0)$.

The execution initialization controls the parameters $R_{ss}(0)$, $R_{nn}(0)$, and ω . For each N_a and signal statistics used, the program evaluates the optimal values of the mean squared error, J_{\min} , and weights a_i^* as follows:

$$J_{\min} = \lim_{\substack{\mu_a \rightarrow 0 \\ k \rightarrow \infty}} J(k) \quad (\text{A.2-2})$$

$$a_i^* = \lim_{\substack{\mu_a \rightarrow 0 \\ k \rightarrow \infty}} a_i(k) \quad \text{for } i=0,1,\dots,N_a-1 \quad (\text{A.2-3})$$

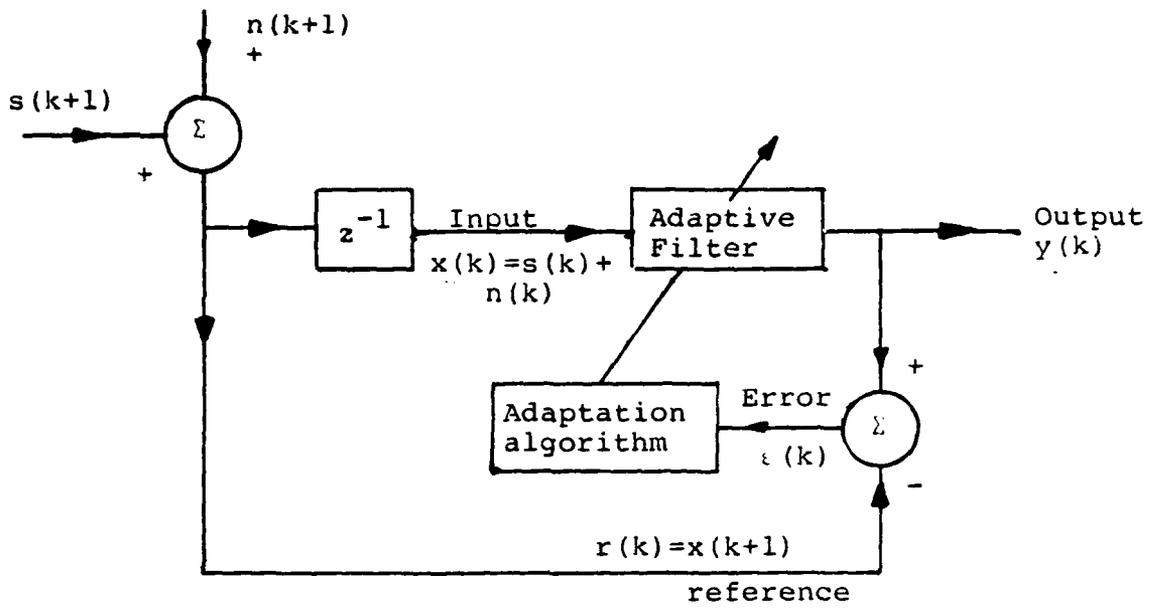


Fig. A.2-1

Adaptive Line Enhancer Block Diagram

For 200 points spread equally over the time of the experiment the program evaluates the performance function, $J(k)$:

$$J(k) = J_{\min} + [\underline{A}_k - \underline{A}^*]^T \underline{R} [\underline{A}_k - \underline{A}^*] \quad (\text{A.2-4})$$

where

\underline{A}_k is the vector of filter weights at time k .

\underline{A}^* is the vector of optimal weights determined by (A.2-3).

\underline{R} is the input signal autocorrelation matrix.

At the end of the experiment the program evaluates:

- (1) The steady-state MSE, J_{ss} , as the average value of $J(k)$ in the last 10% of the experiment.
- (2) Convergence time, TC, directly from the definition as the time required for the error $(J(k) - J_{ss})$ to be reduced to 10% of its initial value.
- (3) Misadjustment, M:

$$M = \frac{J_{ss} - J_{\min}}{J_{\min}} \quad (\text{A.2-5})$$

- (4) Processing gain, PG:

$$PG = 10 \log \left[\frac{R_{nn}(0)}{J_{ss}} \right] \quad (\text{A.2-6})$$

The program includes filter subroutines that perform the LMS, FSC, and the SLMS algorithms.

A.3 IIR SIMULATION PROGRAM

Three configurations were used for IIR adaptive filters simulation:

- ALE configuration as discussed in Section A.2. Differences from the FIR program are indicated later.
- Parameter Identification example as shown in Fig. A.3-1. This example was used to demonstrate convergence to the global minimum of a multimodal performance surface. This example is taken from Johanson and Larimore [11] and was used also by Parikh and Ahmed [13].
- Stability test, as shown in Fig. A.3-2. This test was used with the optimal location for the pole near the unit circle to demonstrate the stability of Feintuch and the random search algorithms, and the lack of stability of the Stearns' algorithm.

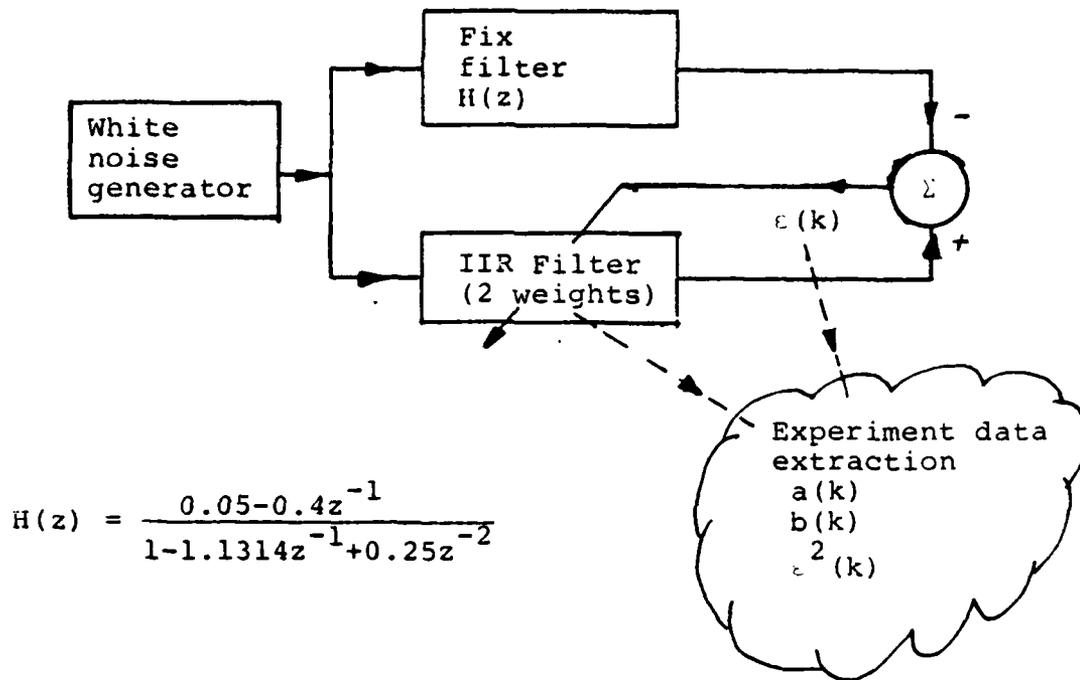
The ALE IIR experiments are similar to the FIR except for the following:

(1) The optimal values of (A.2-2) and (A.2-3) are not used.

(2) The MSE, $J(k)$, is evaluated by:

$$J(k) = \frac{1}{N_{av}} \sum_{j=0}^{N_{av}-1} [y(k-j) - s(k-j)]^2 \quad (A.3-1)$$

where N_{av} is an averaging interval, the values used were between 100 and 500. Obviously $J(k)$ is evaluated only once in each averaging interval.



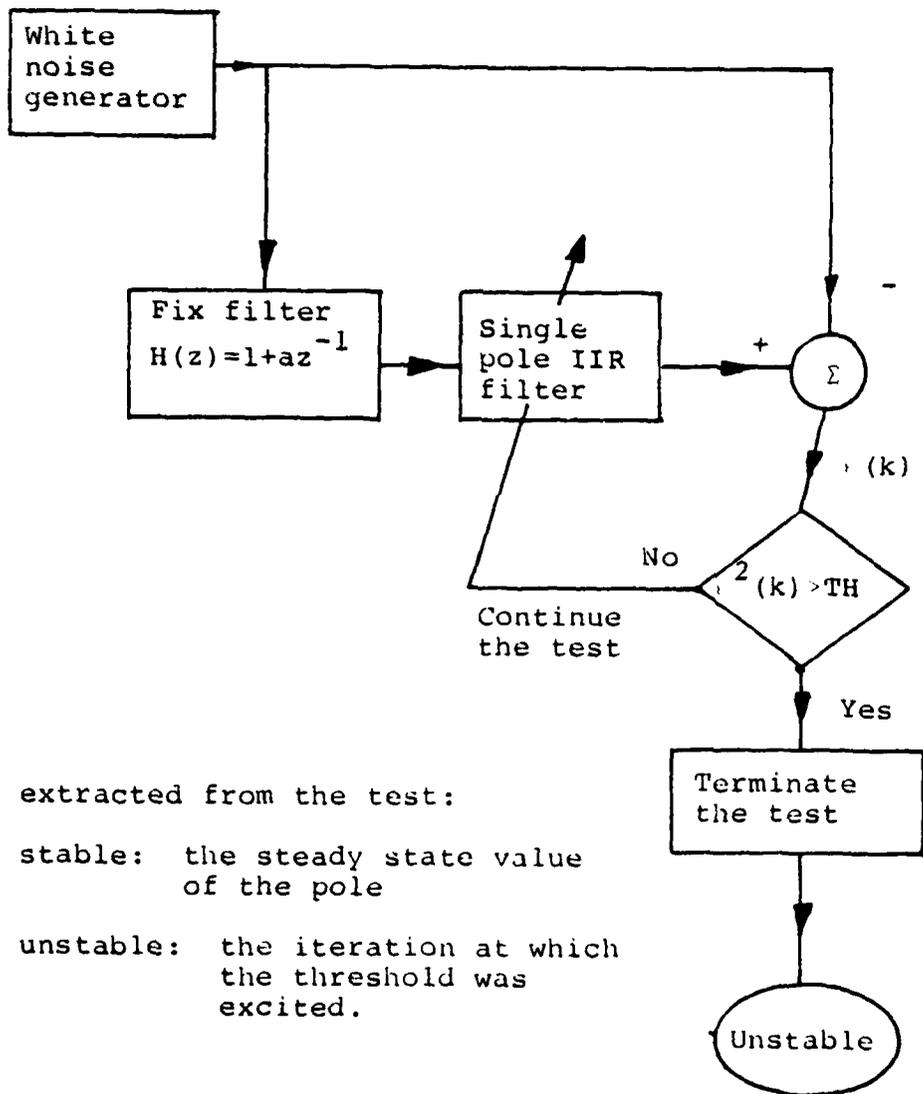
The performance surface has two minima:

local minimum at $a=0.114$, $b=0.519$

global minimum at $a=-0.311$, $b=0.906$

Fig. A.3-1

Parameter Identification Example



Data extracted from the test:

if stable: the steady state value of the pole

if unstable: the iteration at which the threshold was excited.

Fig. A.3-2
 Stability Test

(3) The misadjustment, M , of (A.2-5) is not used.

The program includes filter subroutines that use the following IIR algorithms:

- Feintuch
- Stearns
- Random Search
- RGS
- Apriori Structured Pole.

In order to present the algorithm details and as an example of a filter subroutine, the RGS subroutine is given in Appendix B.

APPENDIX B

SUBROUTINE RGS

```

C
C*****
C
C      RANDOM AND GRADIENT SEARCH SUBROUTINE
C
C      X(100)  INPUT DATA IN 100 ELEMENTS ARRAY
C      R(100)  REFERENCE DATA
C      Y(100)  OUTPUT DATA
C      A( . )  FEEDFORWARD WEIGHTS
C      B( . )  FEEDBACK WEIGHTS
C
C*****
C
C      THE COMMON BLOCK
C      INCLUDE 'AR.CMN'
C      DO 100 K=1,100
C          FEEDBACK SECTION PROCESSING,
C          BT(.) IS THE TESTED POINT.
C          YN=X(K)
C          YTN=X(K)
C          DO 101 J=1,NB
C              YN=YN+B(J)*YB(J)
C              YTN=YTN+BT(J)*YT(J)
101      CONTINUE
C          SHIFTING THE SIGNAL IN THE FILTER'S MEMORY
C          DO 102 J=1,NAB-1
C              YB(NAB-J+1)=YB(NAB-J)
C              YT(NAB-J+1)=YT(NAB-J)
102      CONTINUE
C          YB(1)=YN
C          YT(1)=YTN
C          FEEDFORWARD SECTION PROCESSING
C          Y(K)=0.0
C          ZT=0.0
C          DO 103 J=1,NA
C              Y(K)=Y(K)+A(J)*YB(J)
C              ZT=ZT+A(J)*YT(J)
103      CONTINUE
C          THE ERROR TERMS
C          ER=Y(K)-R(K)
C          ET=ZT-R(K)
C          FEEDFORWARD WEIGHTS' ADAPTATION
C          DO 104 J=1,NA
C              A(J)=A(J)-GA*SIGN(1.0,ER)*SIGN(1.0,YB(J))
104      CONTINUE
C          PERFORMANCE FUNCTION ESTIMATION

```

```

EC=EC+ER*ER
ECT=ECT+ET*ET
IF(ECT.GT.TH) L=LL
L=L+1
IF(L.LE.LL) GO TO 111
C          COMPARISON AND RANDOM SEARCH DECISION MAKING
L=0
IF(ECT.LT.EC) GO TO 200
GO TO 201
200      CONTINUE
DO 203 J=1,NB
        B(J)=BT(J)
        YB(J)=YT(J)
203      CONTINUE
201      CONTINUE
C          NEW TEST POINT SELECTION
DO 204 J=1,NB
        BT(J)=B(J)+GB*GAUSS(0)
204      CONTINUE
DO 205 J=1,NAB
        YT(J)=YB(J)
205      CONTINUE
EC=0.0
ECT=0.0
111      CONTINUE
C          EXPERIMENT DATA EXTRACTION
CALL AVERR(K)
IF(IOUT.EQ.2) CALL WTPR
100      CONTINUE
RETURN
END

```

REFERENCES

1. Widrow, B., Adaptive Filters, A chapter from Aspects of Network and System Theory. Edited by R. E. Kalman and N. De Claris, Holt, Rinehard and Winston, Inc., N.Y., 1970.
2. McCool, J. M., Widrow, B., Principles and Applications of Adaptive Filters: A Tutorial Review, Naval Undersea Center, San Diego, CA, MUC TP 530 AD-A037755, March 1977.
3. Widrow, B., et al, Adaptive Noise Cancelling: Principles and Applications, Proc. IEEE, December 1975.
4. Widrow, B., et al, Stationary and Non-stationary Learning Characteristics of the LMS Adaptive Filter, Proc. IEEE, August 1976.
5. Widrow, B., McCool, J. M., A Comparison of Adaptive Algorithms Based on the Method of Steepest Descent and Random Search, IEEE Trans. on Antennas and Propagation, September 1976.
6. Widrow, B., McCool, J. M., Ball, M., The Complex LMS Algorithm, Proc. IEEE, April 1975.
7. Lucky, R. W., Automatic Equalization for Digital Communication, BSTJ, April 1965.
8. White, S. A., An Adaptive Recursive Digital Filter, Proc. of Asilomar Conference on Circuit Systems and Computers, Pacific Grove, CA, 1975.
9. Stearns, S. D., Elliott, G. R., Ahmed, N., On Adaptive Recursive Filtering, Proc. of Asilomar Conference on Circuit Systems and Computers, Pacific Grove, CA, 1976.
10. Feintuch, P. L., An Adaptive Recursive LMS Filter, Proc. IEEE, November 1976.
11. Johnson, C. R., Larimore, M. G., Comments on and Additions to "An Adaptive Recursive LMS Filter", Proc. IEEE, September 1977.
12. Widrow, B., McCool, J. M., Comments on "An Adaptive Recursive LMS Filter", Proc. IEEE, September 1977.
13. Parikh, D., Ahmed, N., On An Adaptive Algorithm for IIR Filters, Proc. IEEE, May 1978.

14. McMurray, L. R., Stability Diagram for an Adaptive Recursive Filter, Proc. of Asilomar conference on Circuit Systems and Computers, Pacific Grove, CA, 1977.
15. McMurray, L. R., Adaptive Recursive Filter Turn-on Transient, Proc. of Asilomar conference on Circuit Systems and Computers, Pacific Grove, CA, 1978.
16. Dentino, M. J., Frequency Domain Adaptive Correlator, Proc. of Asilomar conference on Circuit Systems and Computers, Pacific Grove, CA, 1977.
17. Zentner, C. R., Frequency Domain Adaptive Decoupling in Multiple Output Array Processors, Pennsylvania State University, TM 75-223 contract N00017-73-C-1418, DD-A023137, September 1975.
18. Parker, S. R., Ko, S. J., A Two Dimensional Adaptive Recursive Digital Filter in Image Processing, IEEE int. symp. on Circuits and Systems, N.Y., May 1978.
19. Gran, R., On the Convergence of Random Search Algorithms In Continuous Time with Applications to Adaptive Control, IEEE Trans. on Systems Man. and Cyber, January 1973.
20. Karnopp, D. C., Random Search Techniques for Optimization Problems, Automatica Vol. 1, pp. 111-121, 1963.
21. McMurtry, G. J., Fu, K. S., A Variable Structure Automation Used as a Multimodel Searching Technique, IEEE Trans. on Automatic Control, July 1966.
22. Ahmed, N., A Study of Adaptive Digital Filters, Sandia Lab., Albuquerque, N.M., 87115, Sand 77-0102, August 1977.
23. Devroye, L. P., The Uniform Convergence of Nearest Neighbor Regression Function Estimators and their Application in Optimization, IEEE Trans. on Information Theory, March 1978.
24. Elliott, G. R., Stearns, S. D., Ahmed, N., Adaptive Transfer Filter Considerations, Asilomar Conference on Circuits Systems and Computers 1977, Pacific Grove, CA.
25. Elliott, G. R., Jacklin, W. L., Stearns, S. D., The Adaptive Digital Filter, Sandia Lab. Report Sand 76-0360, Albuquerque, N.M., August 1976.
26. Frost, O. L., An Algorithm for Linearly Constrained Adaptive Array Processing, Proc. of IEEE Vol. 60, No. 8, August 1972.

27. Moschner, J. L., Adaptive Filter with Clipped Input Data, Stanford University-Center for Systems Research, Technical Report No. 6796-1, June 1970.
28. Nahi, N. E., Estimation Theory and Applications, John Wiley & Sons Inc., New York, 1969.
29. Owsley, N. L., A Constrained Gradient Search Method with Application to Adaptive Sensor Arrays, Navy Underwater Sound Lab., Fort Trumbull, New-London, Connecticut, USL Technical Memorandum No. 2242-207-69, 3 September 1969, AD B 026993.
30. Price, W. L., A Controlled Random Search Procedure for Global Optimization, the Computer Journal Vol. 20, No. 4, 1976.
31. Rastrigin, L. A., Random Search in Problems of Optimization, Identification and Training of Control Systems, Journal of Cybernetics, 1974, 3,3, pp. 93-103.
32. Rickard, J. T., Zeidler, J. R., Second-Order Output Statistics of the Adaptive Line Enhancer, IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-27, No. 1, February 1979.
33. Schumer, M. A., Steiglitz, K., Adaptive Step Size Random Search, IEEE Trans. on Automatic Control Vol. AC-13, No. 3, June 1968.
34. Shensa, M., Time Constants and Learning Curves of LMS Adaptive Filters, Naval Ocean Systems Center, San Diego, CA, Technical Report 312, November 1978.
35. Treichler, J. R., Larimore, M. G., Johnson, C. R., Simple Adaptive IIR Filtering, Proc. of the IEEE International Conference on Acoustics Speech and Signal Processing, Tulsa, Oklahoma, April 1978.
36. Tsypkin, Ya. Z., Some Properties of Random Search, Automation and Remote Control, Vol. 38, November 1977.
37. Treichler, J. R., Transient and Convergent Behavior of the Adaptive Line Enhancer, IEEE Trans. on Acoustic Speech and Signal Processing, Vol. ASSP-27, No. 1, February 1979.
38. Widrow, B., et al, Adaptive Antenna Systems, Proc. IEEE, December 1967.
39. Zeidler, J. R., et al, Adaptive Enhancement of Multiple Sinusoids in Uncorrelated Noise, IEEE Trans. on Acoustics Speech and Signal Processing Vol. ASSP-26, No. 3, June 1978.

40. Gersho, A., Adaptation In a Quantized Parameter Space, Proc. Sixth Annual Allerton Conference on Circuit and System Theory, Monticello, Ill., October 1968.
41. Griffiths, L. J., Jim, C. W., A New Digital Adaptive Beamforming Systems For Microwave Radar Arrays, Septieme Colloque Sur Le Traitement Du Signal Et Ses Applications, Nice France, May 28 to June 2, 1979.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Electrical Engineering Department Chairman Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4. Professor Sydney R. Parker, Code 62Px Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	10
5. Professor G. A. Myers, Code 62Mv Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Professor R. M. Hamming, Code 56Hg Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
7. Professor R. Panholzer, Code 62Pz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
8. Professor P. C. C. Wang, Code 53Wg Mathematics and National Security Affairs Naval Postgraduate School Monterey, California 93940	1
9. Professor I. Bar-David Department of Electrical Engineering Technion Technion City, Haifa Israel	1
10. Uzi Ben-Yakov 25 Golomb Street Ranana, Israel	10

11. Lieutenant Colonel Adam Feit 1
SMC #1079
Naval Postgraduate School
Monterey, California 93940
12. Lieutenant Haim Amir 1
SMC #2641
Naval Postgraduate School
Monterey, California 93940
13. Lieutenant Commander John F. Martinos 1
22 Glafkis St. Halandri
Athens, Greece