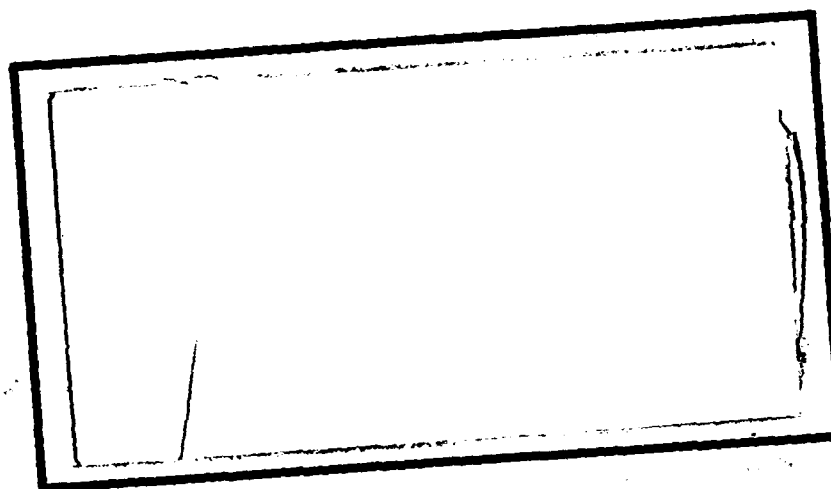


DTIC FILE COPY

AD-A202 578



DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

This document has been approved
for public release and sales in
distribution is unlimited.

DTIC
ELECTE
19 JAN 1989
S D

89 1 17 084

AFIT/GE/ENG/88D-35

PERFORMANCE OF AN ADAPTIVE
MATCHED FILTER USING THE
GRIFFITHS ALGORITHM

THESIS

Peter D. Pasko
Captain, USAF

AFIT/GE/ENG/88D-35

DTIC
ELECTE
S 19 JAN 1989 D
E

Approved for public release; distribution unlimited

AFIT/GE/ENG/88D-35

PERFORMANCE OF AN ADAPTIVE MATCHED FILTER
USING THE GRIFFITHS ALGORITHM

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Peter D. Pasko, B.S.

Captain, USAF

December 1988

Approved for public release; distribution unlimited

Preface

The concept of adaptive filters and adaptive communication systems is destined to play a major role in the future. With this thought in mind, I embarked on a study that analyzed and simulated an adaptive matched filter that uses the Griffiths algorithm. The original idea came from readings offered to me by Martin DeSimio of Wright-Patterson Air Force Base, Ohio. This idea blossomed into this thesis, which is a result of approximately nine months of effort.

This work could not have been accomplished without the help of many people at the Air Force Institute of Technology. The help of my advisor, Major Glenn E. Prescott, was extremely valuable. I also appreciated the suggestions for improvement from Major David M. Norman and Captain Robert Williams. Finally, I would like to thank my wife, Debra, and sons, Max and Jake, for their kindness and understanding these past nine months.



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Table of Contents

	Page
Preface	ii
List of Figures	v
List of Tables	vii
List of Symbols and Notation	.viii
Abstract	xi
CHAPTER 1 : PRELIMINARIES	1
1-1 Introduction	1
1-2 Justification	1
1-3 Background	3
1-4 Statement of Problem	7
1-5 Scope	7
CHAPTER 2 : FUNDAMENTAL CONCEPTS	9
2-1 Introduction	9
2-2 Basic Digital Filtering	9
2-3 Matched Filtering	12
2-4 Adaptive Matched Filtering	14
2-5 Griffiths Algorithm	18
CHAPTER 3 : ADAPTIVE SYSTEM MODELING	21
3-1 Introduction	21
3-2 A System Model for Analysis	21
3-3 Signal and Composite Channel Model	25
3-4 Adaptive Matched Filter Model	31
3-5 Griffiths Algorithm Model	33
3-6 Interference and Noise Model	35
3-7 General Simulation Model	38
CHAPTER 4 : ANALYSIS OF RESULTS	40
4-1 Introduction	40
4-2 Performance Criteria	40
4-2-1 Power Criteria	40
4-2-2 Fixed Criteria	40
4-2-3 Average Error Criterion	42
4-2-4 Defining $\mathbf{p}_I(k)$ and $\mathbf{p}_Q(k)$	45
4-2-5 Defining the P-vector	46

4-3	Jamming Results	51
4-3-1	Effects of Jamming Power and Jamming Frequency (phase constant)	51
4-3-2	Effects of Jamming Phase (frequency and power constant)	63
4-3-3	Effects of Noise on Filter	63
4-3-4	LMS Algorithm	63
4-4	Comparisons	85
4-4-1	Comparisons of No Noise and No Jamming Results	85
4-4-2	Comparisons of Jamming Results	88
4-4-2A	Effects of Jamming Power	88
4-4-2B	Effects of Jamming Frequency/Phase	88
4-4-2C	Effects of Noise Jamming	88
4-4-2D	Comparisons to LMS	88
CHAPTER 5 : CONCLUSION		90
5-1	Summary	90
5-2	Recommendations for Further Study	91
Appendix A: Software Simulation Programs		93
Appendix B: Definitions of Selected Terms		116
Bibliography		117
Vita		119

List of Figures

Figure	Page
2-1. Block Diagram of a Digital Filter	11
2-2. Block Diagram for a Matched Filter Receiver for BPSK	13
2-3. Tapped-delay-line Matched Filter	15
2-4. Adaptive Matched Filter	17
3-1. General Block Diagram of the System Model	22
3-2. Composite Channel Filter	26
3-3. Raised Cosine Frequency Spectrum with Roll-Off Factor Equal to 1	29
3-4. FFT Implementation	32
4-1. 5-Stage Linear Feedback Shift Register	43
4-2. Normalized Autocorrelation Function of $\underline{s}(k)$	44
4-3. Case I: Zero Distortion	47
4-4. Case II: Linear Delay Distortion	48
4-5. Case III: Quadratic Delay Distortion	49
4-6. Case IV: Cubic Delay Distortion	50
4-7. No Jamming/No Distortion	52
4-8. No Jamming/Linear Delay Distortion	53
4-9. No Jamming/Quadratic Delay Distortion	54
4-10. No Jamming/Cubic Delay Distortion	55
4-11. Bar Graph Comparing Various Delay Distortion Levels	56
4-12. -10 DB Plot of Jamming Results (Phase=0)	60
4-13. 0 DB Plot of Jamming Results (Phase=0)	61
4-14. 10 DB Plot of Jamming Results (Phase=0)	62

4-15.	Average Error due to Phase (0 DB and 1 Cycle per PN Length)	70
4-16.	Average Error due to Phase (0 DB and 2 Cycles per PN Length)	71
4-17.	Average Error due to Phase (0 DB and 8 Cycles per PN Length)	72
4-18.	Average Error due to Phase (-10 DB and 1 Cycle per PN Length)	73
4-19.	Average Error due to Phase (-10 DB and 2 Cycles per PN Length)	74
4-20.	Average Error due to Phase (-10 DB and 8 Cycles per PN Length)	75
4-21.	Actual Error due to Noise (SNR = 10 DB)	77
4-22.	Actual Error due to Noise (SNR = 0 DB)	78
4-23.	Actual Error due to Noise (SNR = -10 DB)	79
4-24.	Linear Delay Distortion, No Jamming, LMS Algorithm	80
4-25.	Cubic Delay Distortion, No Jamming, LMS Algorithm	81
4-26.	LMS Algorithm (No Distortion), JSR = 0 DB, 1 Cycle per PN Length	82
4-27.	LMS Algorithm (No Distortion), JSR = 10 DB, 8 Cycles per PN Length	83

List of Tables

Table	Page
1. Average Error of Filter Output for an Input that has No Distortion	57
2. Average Error of Filter Output for an Input that has Quadratic Delay Distortion	58
3. Average Error of Filter Output for an Input that has Cubic Delay Distortion	59
4. Average Error of Filter Output for Different Phase Constants (c1). JSR = 0 DB and Frequency = 1 Cycle per PN Length	64
5. Average Error of Filter Output for Different Phase Constants (c1). JSR = 0 DB and Frequency = 2 Cycles per PN Length	65
6. Average Error of Filter Output for Different Phase Constants (c1). JSR = 0 DB and Frequency = 8 Cycles per PN Length	66
7. Average Error of Filter Output for Different Phase Constants (c1). JSR = -10 DB and Frequency = 1 Cycle per PN Length	67
8. Average Error of Filter Output for Different Phase Constants (c1). JSR = -10 DB and Frequency = 2 Cycles per PN Length	68
9. Average Error of Filter Output for Different Phase Constants (c1). JSR = -10 DB and Frequency = 8 Cycles per PN Length	69
10. Average Error of Filter Output for Various SNRs (No Distortion)	76
11. Comparisons of Griffiths and LMS Algorithms for Selected Runs	84
12. Comparisons of Various Distortion Levels (No Jamming or Noise)	86

List of Symbols and Notation

A	- the amplitude response as a function of frequency for the composite channel filter
B	- complex correlation coefficient
c1	- the jamming phase component
CW	- continuous wave
D(k)	- the desired signal (scalar)
d	- delay coefficient in the phase response of the composite channel filter
e(k)	- the error signal (scalar)
E	- the expectation operator
FFT	- fast Fourier transform
F _J	- the jamming frequency
h(k)	- discrete time value of a digital filter's impulse response (scalar)
I	- in-phase
IFFT	- inverse FFT
Im	- imaginary part
$\underline{i}(k)$	- interference vector
j	- the imaginary operator
JPWR	- the jamming power
JSR	- jamming to signal power ratio
k	- discrete time index
k ₀ ..k ₄	- delay component in the composite channel filter impulse response
LMS	- least mean square
m _y	- the mean value of y
m _x	- the mean value of x

$\underline{m}$ - a mean vector
 N - indicates a quantity or length of a vector
 n - summation index
 $\underline{n}(k)$ - additive white gaussian noise vector
 $\underline{p}(k)$ - impulse response of the composite channel filter
 P - transfer function of the composite channel filter
 PN - pseudonoise
 PN_{bit} - value of a pseudonoise bit
 $\underline{P}$ - the P-vector $\underline{P} = E\{ D(k) \underline{r}^*(k) \}$
 Q - quadrature
 Re - real part
 R_x - autocorrelation of a random process X
 $r(k)$ - discrete time data value (scalar)
 $\underline{r}(k)$ - a received data vector, input to Griffiths filter
 $\underline{s}(k)$ - a signal vector which represents a real random bit stream
 SNR - signal to noise ratio
 t - time
 T - symbol duration time
 T_J - the period of the jammer
 U - a uniform random variable between 0 and 1
 $w_1 \dots w_N$ - individual weights of an N-weight tapped-delay-line
 $\underline{w}(k)$ - tapped-delay-line weight vector
 X - representation of a random process X
 $y(k)$ - discrete time data value (scalar)
 $Y(k)$ - output of the Griffiths filter (scalar)

$z(k)$ - output of the composite channel filter (scalar)
 $\underline{z}(k)$ - vector output of the composite channel filter
 ϵ - mean-square value of the error signal
 η - eigenvalue
 μ - convergence rate constant
 α - amplitude term of a narrowband signal
 θ - phase term of a narrowband signal
 β - independent frequency variable
 Φ - the frequency phase response of the composite channel filter
 σ - the standard deviation of the noise process
 π - the value of Pi
 Ω - the minimum Nyquist bandwidth
 τ - delay variable
 $\delta(k)$ - the Dirac delta function
 Γ - average error over a certain number of adaptations
 $*$ - convolution operator
 $\underline{z}^*(k)$ - complex conjugate of the vector $\underline{z}$
 $\underline{z}^T(k)$ - the transpose of the vector $\underline{z}$

Abstract

This thesis presents a CW and noise jamming analysis of an adaptive matched filter that (1) uses the Griffiths algorithm and (2) has a pseudonoise sequence as an input. The analysis is conducted over several jamming powers, frequencies, and phases. The Griffiths adaptive matched filter is shown to converge for raised cosine pulses that experience no distortion, quadratic delay distortion, and cubic delay distortion. The Griffiths adaptive matched filter diverges for pulses that experience linear delay distortion even though the convergence rate constant is within limits. Throughout the analysis the P-vector is determined apriori and held constant. The Griffiths filter is shown to converge for CW jamming and noise jamming. Noise jamming is shown to be more effective in the higher power ranges. A comparison is made between the Griffiths adaptive matched filter and an adaptive matched filter that uses the LMS algorithm. The degradation in performance of the Griffiths filter compared to an LMS filter that uses a stored reference is calculated for several selected runs. The actual computer programs used are presented in the Appendix.

**PERFORMANCE OF AN ADAPTIVE
MATCHED FILTER USING THE
GRIFFITHS ALGORITHM**

CHAPTER 1 : PRELIMINARIES

1-1 INTRODUCTION

This chapter addresses basic preliminaries. A justification for this research effort is stated along with selected background information. The concept of adaptive matched filtering is explained along with differences from several related concepts. A statement of the problem is presented, and the scope of the research effort is defined.

1-2 JUSTIFICATION

Performance of a military, digital communications receiver in a jamming or intentional interference environment is a primary consideration for military planners. Unexpected performance of a receiver on the electronic battlefield can have disastrous consequences for the user. The Air Force has realized the importance of

secure and effective electronic communications and has incorporated the concept into its basic aerospace doctrine.

Commanders rely on...secure...communications. Communications are the means through which a commander transmits and receives information about the enemy, coordinates with friendly forces, and commands and controls assigned forces (1:2-21).

The criticality of good communications to military operations cannot be overemphasized. Secure, jam resistant communications must be established (2:8).

Since secure and jam-resistant communications are important to the Air Force, various electronic communication receivers have been deployed to counteract the expected jamming threat. These receivers use many different methods; however, one trend has been toward adaptive signal processing. This trend toward adaptive signal processing is expected since the electronic battlefield is a changing environment, and one can not predict what type of electronic signals will be present.

Adaptivity is needed...because the environment is apriori unknown, time varying, and noncooperative. The present day systems have fixed architecture and a limited degree of adaptivity. This corresponds to a relatively simple hardware but provides non-optimal performance when the environment differs from that assumed in the radar design....Adaptivity should be used extensively in each subsystem of the radar (3:178).

1-3 BACKGROUND

One use of adaptive signal processing to help counter interference or jamming is in the matched filter circuitry of a digital communications receiver. In this situation, fixed coefficient, tapped-delay-line matched filters are designed so that they can adapt to received waveforms. This would be beneficial since one could not predict what type of waveform would be received in a jamming or interference environment. Once the matched filters have adapted to the received waveform, the standard decision circuitry and test statistics that were used in the fixed coefficient case can then be used.

The ideal adaptive matched filter adapts itself to the received waveform so that the filter becomes matched to the received waveform. The matched filter is made to adapt so that "the filter is optimized by minimizing the mean-square value of the error signal" (10:27). A filter which minimizes the mean-square error is optimum in the mean-square sense. The adaptive matched filter minimizes the mean-square error between the actual filter output and a desired signal. By minimizing the mean-square error, the adaptive matched filter's output, or test statistic, can remain approximately constant for a given transmitted

symbol, and the signal to noise ratio at the output will be maximized at the sampling time.

Several other concepts closely parallel adaptive matched filtering but will not be covered in this thesis. These other concepts are (1) equalization, (2) adaptive interference suppression, and (3) adaptive beam forming.

In equalization, the distortion caused by a channel is compensated for so that intersymbol interference (ISI) can be reduced. Equalization is used to combat ISI (9:545). In equalization, the sampling rate is the symbol rate (11:105). By contrast, in adaptive matched filtering the sampling rate is an integer multiple of the symbol rate. Adaptive matched filtering is used to optimally detect in the mean-square sense the received digital waveform, and, therefore, it will combat intentional interference and noise. It is not intended to correct or eliminate ISI.

In adaptive interference suppression, the interference is notched out in the frequency domain to obtain an approximation of the signal. The method is used in such devices as adaptive notch filters and adaptive line enhancers. The basic idea is to suppress the interference or noise and extract the signal (19:1698). By contrast, in adaptive matched filtering, no attempt is made to suppress the interference in the frequency domain. Instead, the adaptive matched filter maximizes the signal to noise ratio

at the output by minimizing the mean-square value of the error signal.

In adaptive beam forming an antenna radiation pattern is purposely nulled out in specific directions in order to suppress interference (8:409). The method works well if the signal and interference can be separated spatially. In adaptive matched filtering the concept of look-directions has no physical meaning and matched filtering is not accomplished in the antenna section. In adaptive matched filtering the suppression of interference is accomplished by maximizing the signal to noise ratio at the receiver decision circuitry.

The concepts of equalization, adaptive interference suppression, adaptive beam forming, and adaptive matched filtering all have basically the same goal. They all try, in some way, to provide reliable communication in a hostile (or less than optimum) environment. The actual concept employed will depend on the expected threat and the decisions made by senior commanders (20:28). In this thesis, the concept of adaptive matched filtering will be covered. Also, the performance of an adaptive matched filter against an expected jamming threat will be covered.

An adaptive matched filter is a finite impulse response (FIR) filter whose impulse response is adjusted in some optimum way. The procedure for adjusting the adaptive matched filter is known as the adaptive algorithm. Two of

the more popular algorithms are the conventional least-mean-square (LMS) algorithm developed by Widrow and Hoff and, a LMS variant, the Griffiths algorithm. This thesis is mainly concerned with the Griffiths algorithm.

The Griffiths algorithm, which is also known as the P-vector algorithm or modified LMS, was developed by L.J. Griffiths in 1969 and has an advantage in that the desired signal term is not used directly in the algorithm (4:1696). Since the desired signal is not used directly in the algorithm, the requirement of the receiver knowing the desired signal (i.e having a stored reference) is no longer necessary. Also, Feuer and Weinstein have shown that the Griffiths algorithm produces lower *misadjustment* (see Appendix B) than the conventional LMS algorithm when the correlation is low between the received data and the desired signal (5:226).

Due to its lower misadjustment compared to the conventional LMS, the Griffiths algorithm is showing somewhat increased usage in the reception of signals in the low SNR environment. A recent trend has been to use the Griffiths adaptive algorithm for adaptive processing of signals in the matched filter circuitry of digital communication receivers.

1-4 STATEMENT OF PROBLEM

The Griffiths algorithm is being implemented in the matched filter circuitry of digital communication receivers due to its expected better misadjustment in a low SNR environment; however, no comprehensive study has been accomplished concerning the performance of the Griffiths algorithm in a jamming or intentional interference environment with direct application to a communications system. The question that needs to be answered is whether the Griffiths adaptive matched filter provides any advantages over conventional methods (i.e. LMS) in a communications environment. For example, Feuer and Weinstein's work investigated performance in Gaussian noise only.

Since performance in a jamming environment is important, the analysis of a Griffiths adaptive filter in a jamming environment (with direct applications to communications) needs to be accomplished. This thesis will investigate performance against a continuous wave (CW) jammer and a noise jammer.

1-5 SCOPE

This thesis will describe a simulation, via computer program, of an adaptive matched filter that uses

the complex Griffiths algorithm. The following objectives will be accomplished in this thesis:

A. Perform a theoretical analysis of the Griffiths adaptive matched filter.

B. Model the Griffiths adaptive matched filter via a computer program.

C. Input a pseudonoise (PN) test signal of length 31 into the Griffiths adaptive matched filter and analyze performance. Performance will be analyzed over a range of jamming to signal power ratios (-10 DB, 0 DB, and 10 DB) and signal phase distortion levels. Signal phase distortion levels will be one of the following four types:

- 1) No distortion
- 2) Linear delay distortion
- 3) Quadratic delay distortion
- 4) Cubic delay distortion

A comparison of results to the LMS adaptive matched filter will be accomplished. The jammer will be modeled as both a CW jammer and noise jammer.

CHAPTER 2 : FUNDAMENTAL CONCEPTS

2-1 INTRODUCTION

In this chapter the fundamental concepts of Griffiths adaptive matched filtering will be explained. First, the simple definition of a digital filter will be addressed and then the more advanced concepts of matched filtering and adaptive filtering will be presented. Finally, an explanation of the complex Griffiths algorithm will be presented along with its relationship to the conventional LMS algorithm.

2-2 BASIC DIGITAL FILTERING

Basic digital filtering is concerned with the extraction of information from some type of data. In a digital communication system, the information is usually extracted from a received data sequence. The received data sequence contains a signal of interest corrupted by noise. The digital filter operates on the received data sequence in order to make an estimate of the signal of interest. In many cases, such as matched filtering, the digital filter produces a test statistic which indicates which symbol waveform was transmitted.

Basic digital filters are usually implemented in (1) filtering operations, (2) smoothing operations, or (3) prediction operations (10:1). In this thesis, only the concept of filtering will be treated. Filtering is defined to be "the extraction of information about a quantity of interest at time t by using data measured up to time t " (10:1).

Digital filters are usually represented by the block diagram given in Figure 2-1. $r(k)$, which is a sampled version of a cyclostationary (see Appendix B) random process, is the input sequence. $r(k)$ is also assumed to have been sampled at or above the Nyquist rate. $h(k)$ is the filter impulse response, and $y(k)$ is the filter output sequence. The representation in Figure 2-1 is the time domain representation. An equivalent frequency domain representation could be obtained by taking the appropriate Fourier transforms (7:101). In this thesis, only time domain representations will be considered.

Digital filters are classified as either infinite impulse response (IIR) or finite impulse response (FIR). This thesis will be concerned with causal FIR filters. Causality restricts the filter to have an output based only on present and past inputs. FIR filters are also known as transversal filters or tapped-delay-line filters.

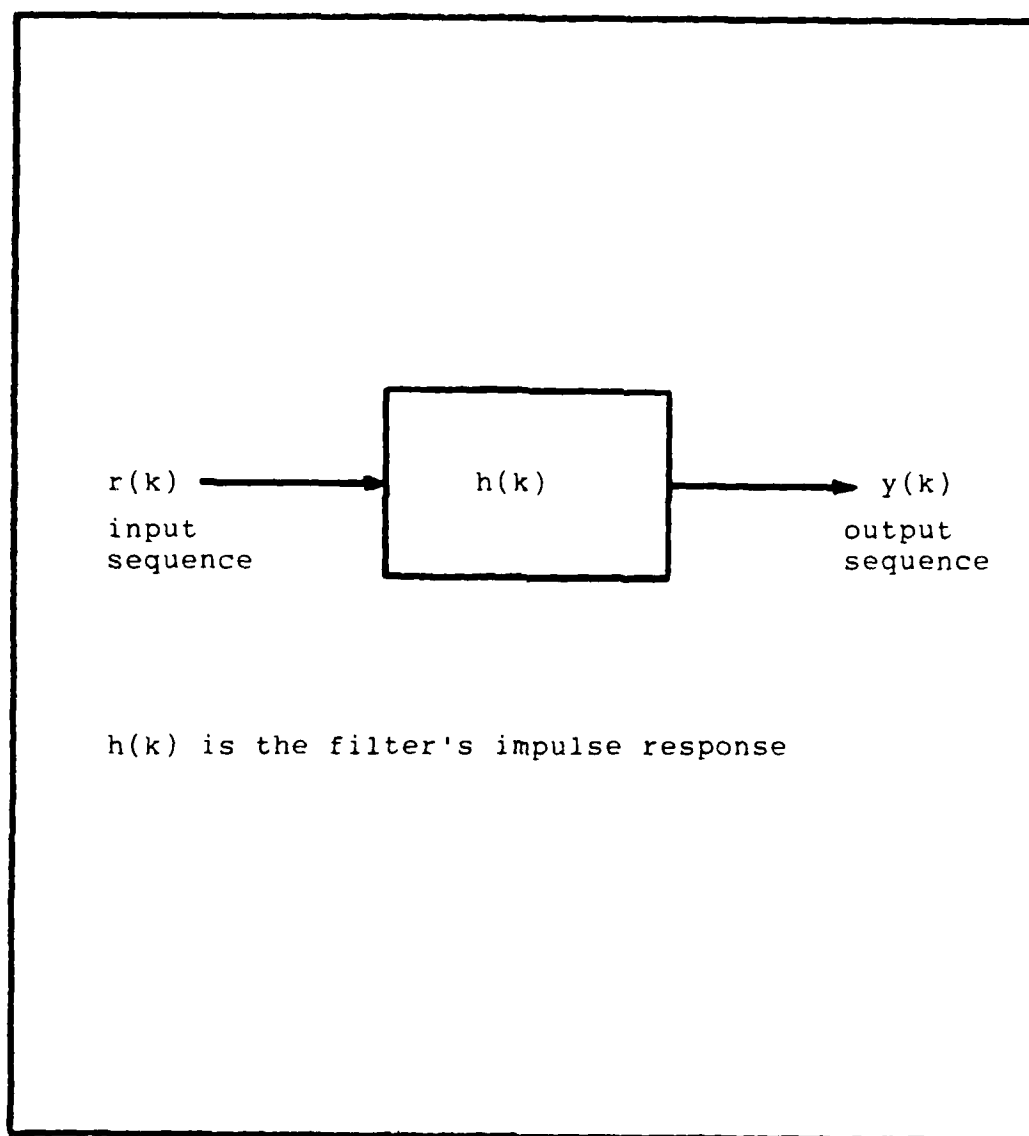


Figure 2-1 Block Diagram of a Digital Filter

2-3 MATCHED FILTERING

In this thesis we are concerned with digital filters implemented in the matched filter circuitry of a digital communication receiver. "A matched filter is a linear filter designed to provide the maximum signal-to-noise power ratio at its output for a given transmitted symbol waveform" (11:88). Figure 2-2 shows a typical block diagram of a matched filter receiver for binary phase shift keyed (BPSK) signals.

In Figure 2-2 the received data vector $\underline{x}(k)$ is processed in two matched filters. The outputs of the matched filters are sampled at time $t=T$. The sampling time T corresponds to the symbol time and occurs when $\underline{x}(k)$ is completely justified within the matched filter (sampling time is also related to filter length and the eye opening of the eye diagram). The two sampled outputs (or test statistics) are then compared, and a decision is then made on which symbol was sent. Throughout the entire process synchronization is assumed to be maintained.

In the development of the matched filter (11:88), a known signal $\underline{g}(k)$ and additive white gaussian noise $\underline{n}(k)$ are assumed to be input to the matched filter circuitry. Hence, the received signal $\underline{x}(k)$ is of the form

$$\underline{x}(k) = \underline{g}(k) + \underline{n}(k) \quad (1)$$

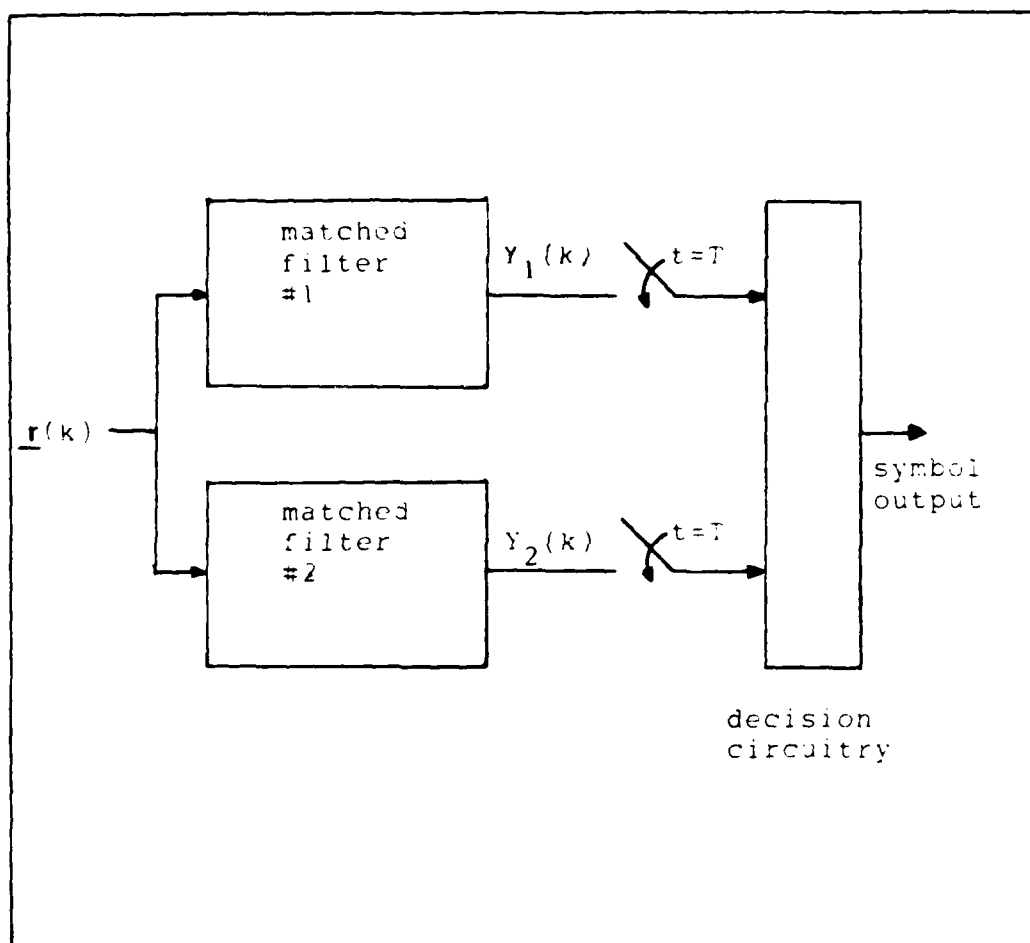


Figure 2-2 Block Diagram for a Matched Filter Receiver for BPSK

The matched filter can be implemented digitally as a tapped-delay-line. A tapped-delay-line matched filter is shown in Figure 2-3. In Figure 2-3, five samples occur each symbol duration T . The weights w_1 thru w_5 are determined by the signal vector $\underline{s}(k)$ and are fixed and known apriori. The output of the tapped-delay-line is defined by the matrix equation

$$Y(k) = \underline{x}^T(k) \underline{w}(k) \quad (2)$$

Note that a scalar (or test statistic) is output for each vector input. When $Y(k)$ is sampled at time $t=T$, the symbol duration time, the signal to noise ratio is maximized. Note that $\underline{x}(k)$, $\underline{n}(k)$, $\underline{s}(k)$, $\underline{w}(k)$, and $Y(k)$ can all be complex quantities.

2-4 ADAPTIVE MATCHED FILTERING

In the previous two sections, discussion has centered on basic digital filtering and matched filtering. It was shown that a matched filter could be implemented as a tapped-delay-line FIR filter. When the signal vector $\underline{s}(k)$ is known exactly and the noise vector $\underline{n}(k)$ is an additive white gaussian noise process, the matched filter can be implemented as a fixed weight tapped-delay-line.

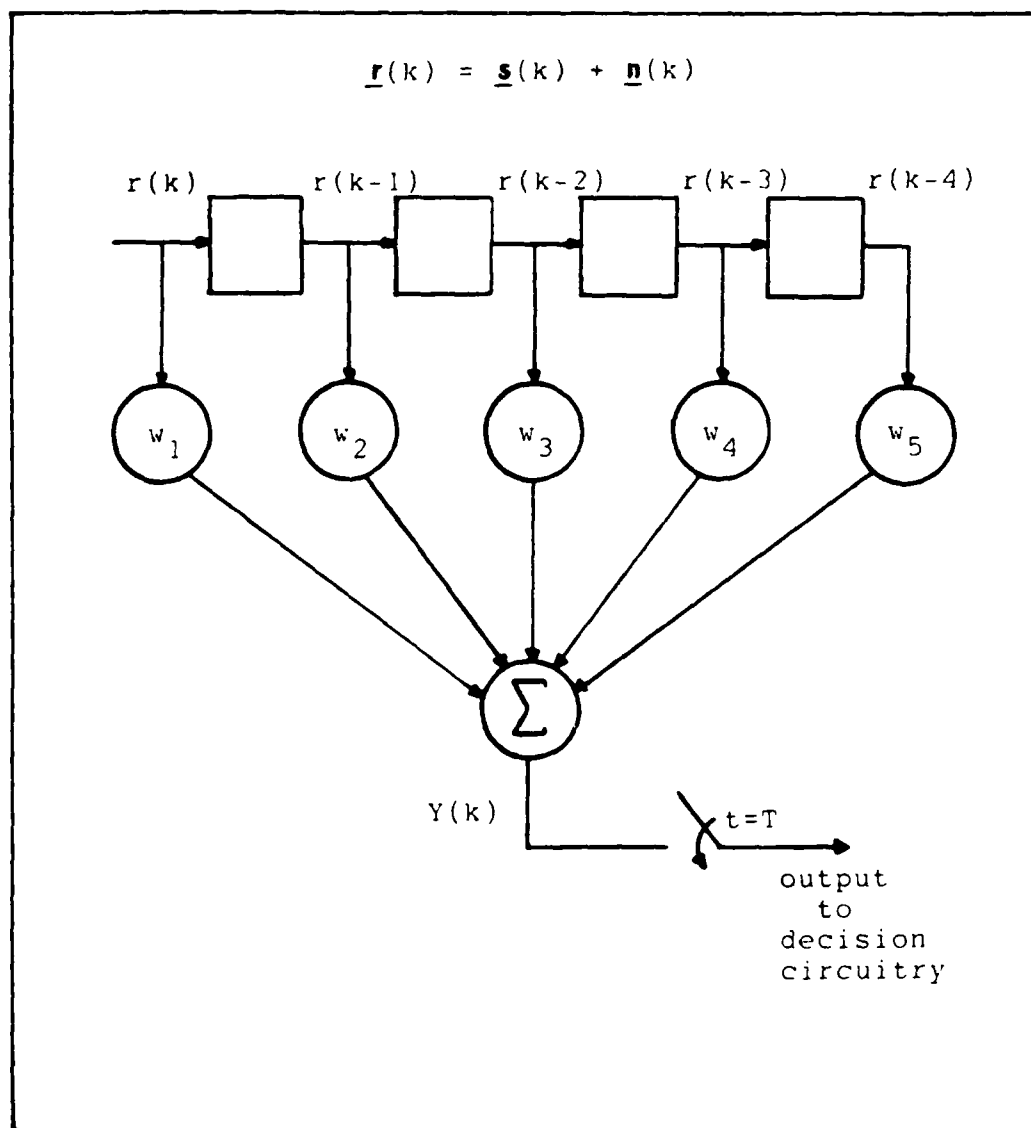


Figure 2-3 Tapped-delay-line Matched Filter

In most instances, however, the signal $s(k)$ is not known exactly. Distortion in the communication channel may distort $s(k)$ in amplitude and phase. Also, intentional interference or jamming may affect $s(k)$ and contribute to $r(k)$. In cases where $s(k)$ is distorted, the fixed weight matched filter is less than optimum since, in the derivation of the matched filter, $s(k)$ was assumed to be known exactly.

One way of overcoming the problem of nonoptimum performance is to make the fixed-weight matched filter adapt to the received vector $r(k)$. This can be accomplished by modifying the diagram of Figure 2-3 to that of Figure 2-4.

Figure 2-4 shows the classic tapped-delay-line modified to perform the adaptive filtering operation (10:91). In this case, the weight vector $w(k)$ is adjusted to match the received vector $r(k)$. Once the weights have adjusted to $r(k)$, the filter becomes matched. When the weight vector $w(k)$ has completely adapted, $y(k)$ will approximately equal $d(k)$ and the mean-square value of the error signal will be minimized.

The advantage of having an adaptive matched filter is that, if $r(k)$ changes due to amplitude, phase, or interference distortion, the filter can adapt to the new $r(k)$.

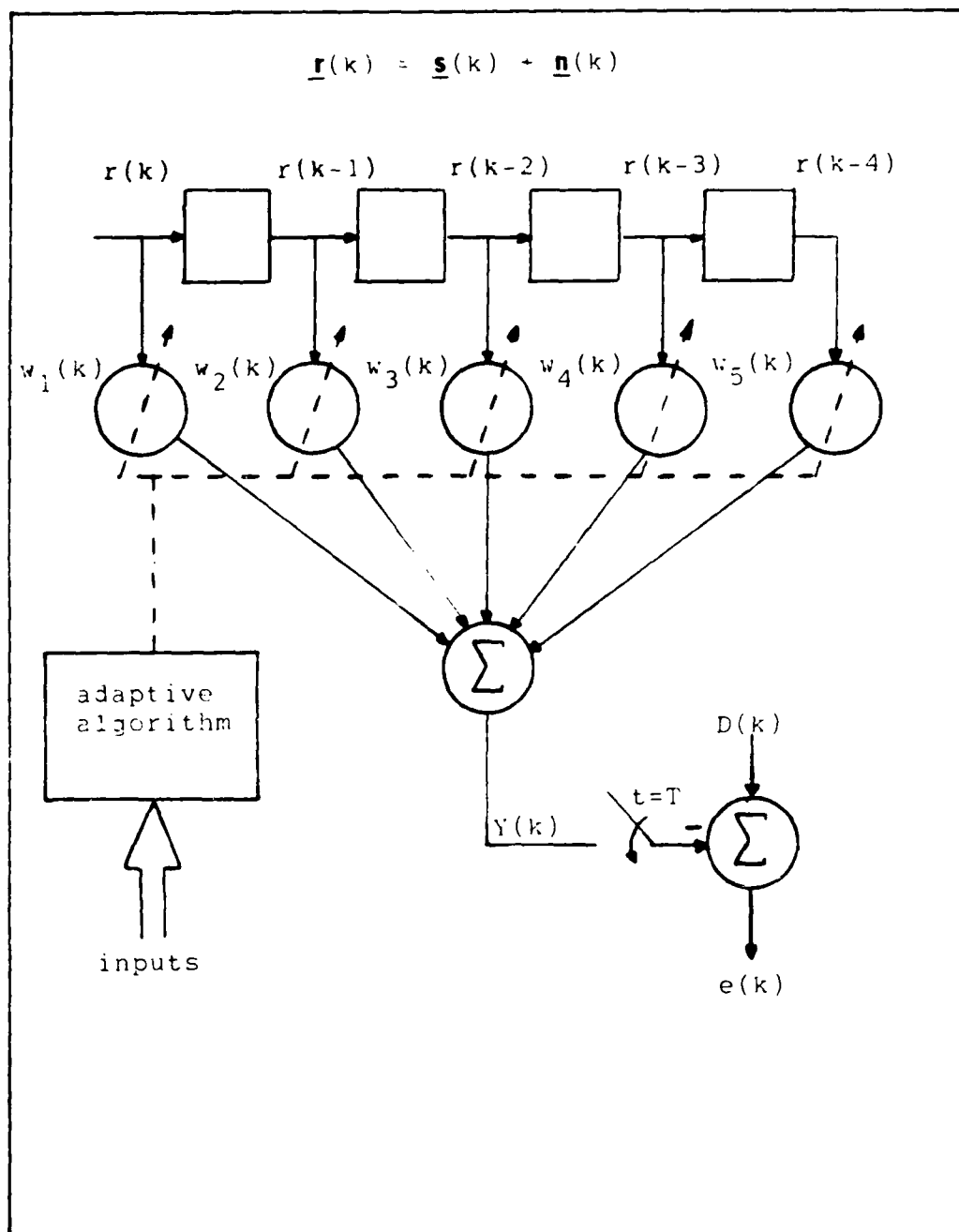


Figure 2-4 Adaptive Matched Filter

2-5 GRIFFITHS ALGORITHM

The adaptive algorithm in Figure 2-4 is the procedure by which the weights are updated. In this thesis, the complex Griffiths algorithm will be implemented as the adaptive algorithm.

The complex Griffiths algorithm can be derived from the Widrow-Hoff LMS algorithm (8:412). The complex Widrow-Hoff LMS algorithm is given as (15:719)

$$\underline{w}(k+1) = \underline{w}(k) + 2\mu e(k) \underline{x}^*(k) \quad (3)$$

where

k = the k^{th} iteration or clock cycle

$\underline{w}(k+1)$ = the updated complex weight vector

$\underline{w}(k)$ = the previous complex weight vector

μ = convergence rate constant

$e(k)$ = complex error signal

$\underline{x}^*(k)$ = the complex conjugate of the received vector

If the substitution $e(k) = D(k) - Y(k)$, where $D(k)$ is the desired signal and $Y(k)$ is the actual output, is made in equation (3) then

$$\underline{w}(k+1) = \underline{w}(k) + 2\mu [D(k) - Y(k)] \underline{x}^*(k) \quad (4)$$

After expanding, equation (4) becomes

$$\underline{w}(k+1) = \underline{w}(k) + 2\mu D(k) \underline{x}^*(k) - 2\mu Y(k) \underline{x}^*(k) \quad (5)$$

Equation (5) is an expanded form of the LMS algorithm. In 1969 L.J. Griffiths investigated the possibility of removing the $D(k)$ term from the algorithm. The motivation for this was that in certain adaptive receiving antenna arrays a desired signal is not available for use in the adaptive algorithm, and methods such as the LMS algorithm, which require a desired signal, cannot be directly applied (6:705). Griffiths' work led him to replace the $D(k) \underline{x}^*(k)$ term with its expected value (4:1699). With this substitution, equation (5) becomes

$$\underline{w}(k+1) = \underline{w}(k) + 2\mu E\{ D(k) \underline{x}^*(k) \} - 2\mu Y(k) \underline{x}^*(k) \quad (6)$$

and let $\underline{p} = E\{ D(k) \underline{x}^*(k) \}$

$$\underline{w}(k+1) = \underline{w}(k) + 2\mu \underline{p} - 2\mu Y(k) \underline{x}^*(k) \quad (7)$$

Equation (7), as shown above, is the complex Griffiths algorithm (6:705). The middle term in equation (7), also known as the P-vector, is a constant determined from apriori information (4:1699). The P-vector in this thesis will be determined from a signal that has no

distortion. The degradation in filter performance will be analyzed when the actual received signal has phase distortion added (which is different from what the P-vector was determined from).

Equation (7) can be shown to converge to the best mean-square estimate of the optimum Wiener solution as the number of adaptations approaches infinity (8:413). This convergence is guaranteed as long as the convergence rate constant satisfies

$$0 < \mu < \frac{1}{\eta_{\max}} \quad (8)$$

where

$\eta_{\max}$ = the largest eigenvalue of the input correlation matrix

"Like the LMS algorithm, the Griffiths algorithm is unbiased and produces at convergence an expected steady-state solution that is the true least-squares solution" (8:413). The advantages of the Griffiths algorithm are that the desired signal, $D(k)$, is not directly used in the algorithm and it produces lower misadjustment than the Widrow-Hoff LMS when the correlation is low between the received vector and the desired response (6:705).

CHAPTER 3 : ADAPTIVE SYSTEM MODELING

3-1 INTRODUCTION

This chapter will outline the model used to test and simulate the Griffiths adaptive matched filter. First, an overall system model will be presented. This system model will then be expanded and broken down into individual subsystems. Each subsystem will be analyzed separately. The following models will be explained in this chapter:

- 1) system model
- 2) signal and composite channel filter model
- 3) adaptive matched filter model
- 4) Griffiths algorithm model
- 5) interference and noise model

3-2 A SYSTEM MODEL FOR ANALYSIS

The purpose of this thesis is to analyze the performance of a Griffiths adaptive matched filter over a range of signal inputs. The general block diagram of the system model is shown in Figure 3-1. The input signal is $\underline{s}(k)$, which is a real random bit stream of a nonreturn-to-zero level (NRZ-L) waveform (11:79). The signal $\underline{s}(k)$ will then pass through a composite channel filter. The composite

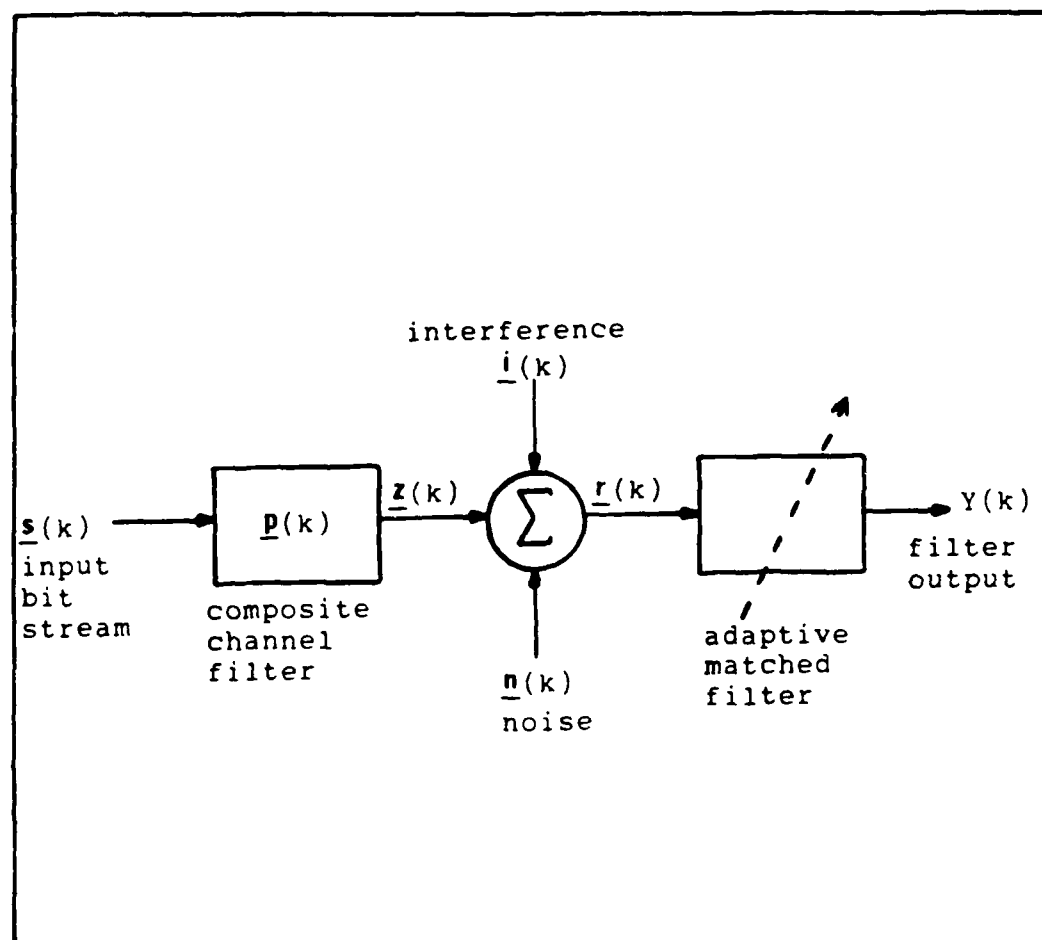


Figure 3-1 General Block Diagram of the System Model

channel filter will represent a filter that both pulse shapes and phase distorts the original bit stream. The composite channel pulse shaping filter will be of the raised cosine type and will introduce various levels of phase distortion (see Appendix B). The composite channel filter finite impulse response is $p(k)$. The output of the composite channel filter is $\underline{x}(k)$. After the composite channel, the signal will have noise and interference added to it. The signal $\underline{x}(k)$, which has channel distortion along with noise and interference added, will serve as input to the adaptive matched filter. The response of the filter, $Y(k)$, to various inputs (that have distortion, noise, and interference added) will then be accomplished.

The signals used in this thesis will be assumed to be narrowband. This implies that the spectral components of the signals of interest are confined to a band that is small compared to the band center frequency, or carrier frequency.

In general, a narrowband representation for the input waveform can be expressed as (12:57)

$$s(k) = a_s(k) \cos[\beta_0 k + \theta_s(k)] \quad (9)$$

and the narrowband representation for the composite channel filter can be expressed as

$$p(k) = 2 a_p(k) \cos[\beta_0 k + \theta_p(k)] \quad (10)$$

where

$\alpha(k)$ = amplitude or envelope modulation

$\theta(k)$ = angle modulation

β_0 = carrier frequency

k = the discrete time index

Note that in equation (10) a constant factor of two has been added for convenience. It is assumed that the carrier frequency is much greater than the bandwidth of the signal and channel filter.

In general some form of modulation is required to transmit information across a communication channel. In this thesis, the specific form of modulation will not be considered; instead, baseband effects will be analyzed and the complex, low-pass, baseband representation of all sequences will be used. This means the various vectors in Figure 3-1 can be defined in the following way

$$\underline{p}(k) = \underline{p}_I(k) + j \underline{p}_Q(k) \quad (11)$$

$$\underline{x}(k) = \underline{x}_I(k) + j \underline{x}_Q(k) \quad (12)$$

$$\underline{i}(k) = \underline{i}_I(k) + j \underline{i}_Q(k) \quad (13)$$

$$\underline{n}(k) = \underline{n}_I(k) + j \underline{n}_Q(k) \quad (14)$$

$$\underline{r}(k) = \underline{r}_I(k) + j \underline{r}_Q(k) \quad (15)$$

$$\underline{y}(k) = \underline{y}_I(k) + j \underline{y}_Q(k) \quad (16)$$

where

j = imaginary operator

subscript I = the in-phase, or I channel, component

subscript Q = the quadrature, or Q channel, component

k = the k^{th} iteration or clock cycle

Note that the input signal $s(k)$ is real and has no imaginary component.

3-3 SIGNAL AND COMPOSITE CHANNEL FILTER MODEL

The input signal is $s(k)$, which is a real random bit stream. For instance, $s(k)$ could possibly be the information content of a radio frequency pulse train generated by a radar or a digitally coded pulse train of a radio transmission. $s(k)$ will be a sampled version of a NRZ-L waveform. Impulse sampling will be assumed and each sample will be represented by a Dirac delta function, which is also known as an impulse function (14:50). The bit stream will then pass through a composite channel filter which will pulse shape the impulse samples and introduce various types of phase distortion.

The composite channel filter can be represented as shown in Figure 3-2. In Figure 3-2 the input bit stream is passed through the composite channel filter and the output

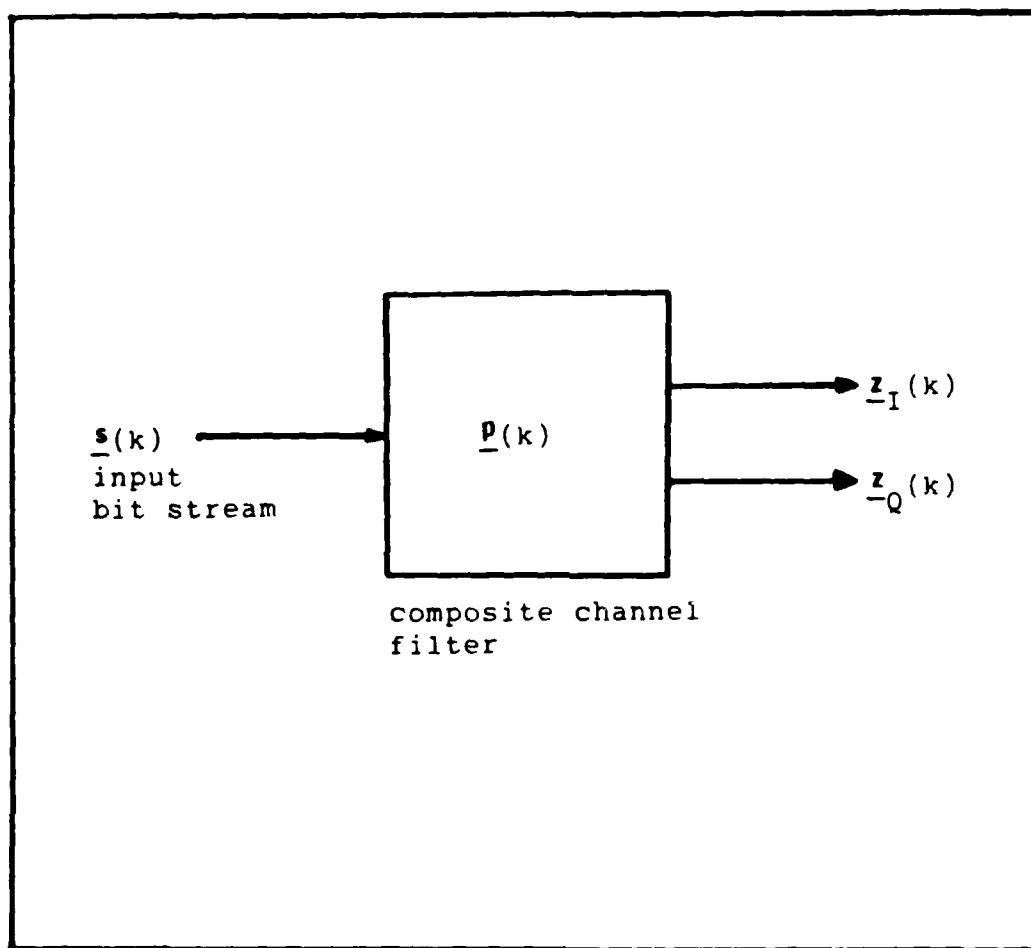


Figure 3-2 Composite Channel Filter

is two signals, $\underline{x}_I(k)$ and $\underline{x}_Q(k)$, which represent the in-phase and quadrature components of $\underline{x}(k)$.

The transfer function of the composite channel filter can be expressed as

$$P(\beta) = A(\beta) e^{-j \Phi(\beta)} \quad (17)$$

where

$P(\beta)$ = the channel transfer function

$A(\beta)$ = the amplitude response

$\Phi(\beta)$ = the phase response

β = the independent frequency variable

In order to generate pulses that exhibit some form of phase distortion, $A(\beta)$ is set equal to one and $\Phi(\beta)$ is defined the following way

$$\Phi(\beta) = d_0 + d_1\beta + d_2\beta^2 + d_3\beta^3 + d_4\beta^4 \quad (18)$$

where

d_0 = constant phase term

d_1 = constant delay term

d_2 = linear delay distortion term

d_3 = quadratic delay distortion term

d_4 = cubic delay distortion term

Note that more terms could be added to equation (18) to completely describe any phase characteristic. "The constant phase term and the constant delay term alone introduce no distortion; however, they play an important part in mathematically describing some arbitrary phase characteristic" (17:9).

If the composite channel filter of Figure 3-2 is implemented as a raised cosine filter with *roll-off factor* (see Appendix B) equal to one (see Figure 3-3) and the spectrum of the filter is given by

$$P(\beta) = \begin{cases} \frac{\pi}{2\Omega} \cos^2 \left[\frac{\pi\beta}{4\Omega} \right] & 0 \leq |\beta| < 2\Omega \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

where

$\pi/2\Omega$ = a scaling factor chosen for convenience

$2/T$ = the signaling rate

$T = \pi/\Omega$

2Ω = the maximum frequency in the passband

Ω = the minimum Nyquist bandwidth

then the components of the composite channel filter can be expressed as (17:12)

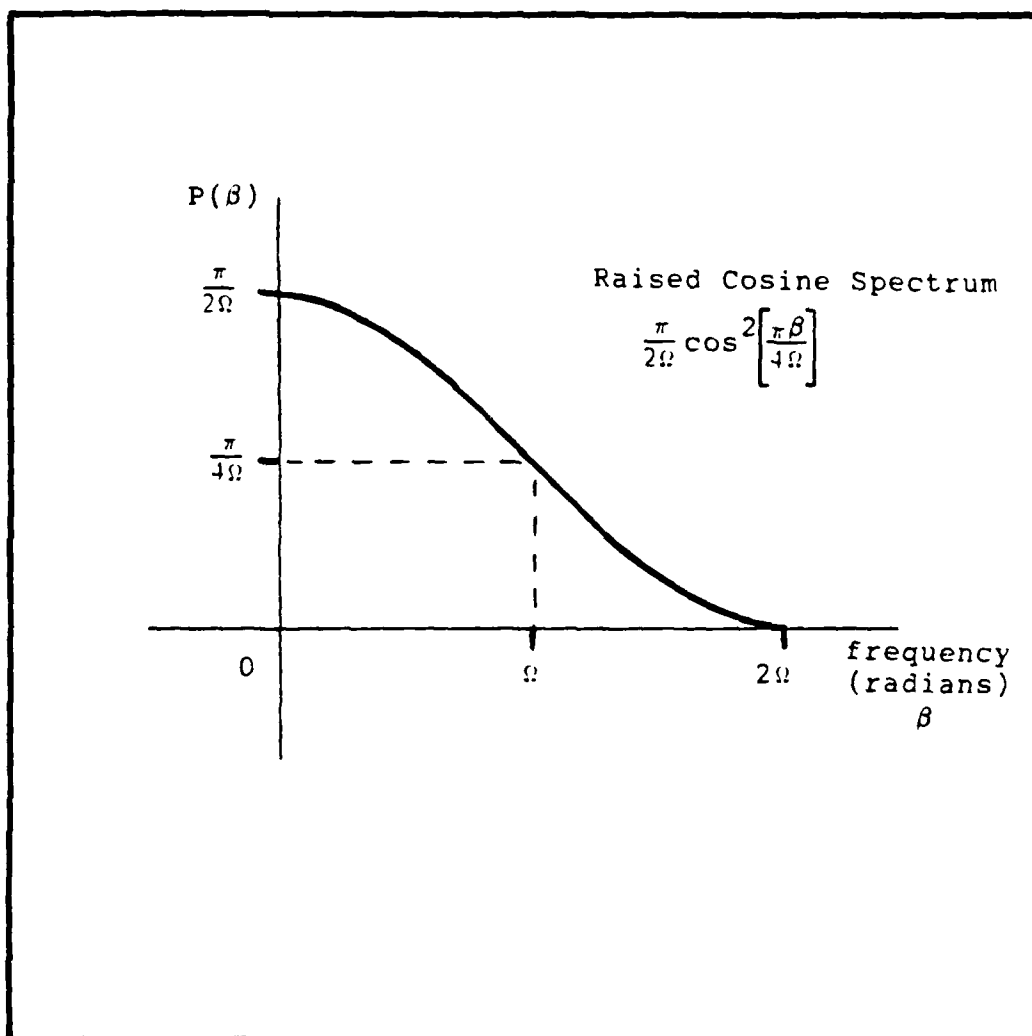


Figure-3-3 Raised Cosine Frequency Spectrum
with Roll-Off Factor Equal to 1

$$p_I(k) = \frac{2}{\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^2 x \cos \left[\frac{4tx}{T} - k_0 - \frac{4k_1}{T}x - \frac{4k_2}{\pi T}x^2 - \frac{16k_3}{3\pi^2 T}x^3 - \frac{8k_4}{\pi^3 T}x^4 \right] dx \quad (20)$$

$$p_Q(k) = \frac{2}{\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^2 x \sin \left[\frac{4tx}{T} - k_0 - \frac{4k_1}{T}x - \frac{4k_2}{\pi T}x^2 - \frac{16k_3}{3\pi^2 T}x^3 - \frac{8k_4}{\pi^3 T}x^4 \right] dx \quad (21)$$

where

$\frac{k}{T}$ = normalized time variable in number of bit intervals

k_0 = constant phase component expressed in radians

k_1 = constant delay component

k_2 = linear delay component

k_3 = quadratic delay component

k_4 = cubic delay component

Equations (20) and (21) give the composite channel filter impulse response for the in-phase and quadrature components.

The outputs of the composite channel filter are then defined by the following equations:

$$\underline{z}_I(k) = \underline{s}(k) * \underline{p}_I(k) \quad (22)$$

$$\underline{z}_Q(k) = \underline{s}(k) * \underline{p}_Q(k) \quad (23)$$

By selecting different values of k_0 thru k_4 , one can generate different forms of phase distortion on the input bit stream $\underline{s}(k)$.

In Figure 3-1, the system model, the input $\underline{s}$ needs to be convolved with the channel response. To facilitate the convolution of these functions the fast Fourier transform (FFT) implementation of convolution will be used. Figure 3-4 shows how the FFTs will be used to perform the linear convolution. The input sequences need to be augmented with zeros so that the required FFT multiplication results in a linear convolution (7:270).

3-4 ADAPTIVE MATCHED FILTER MODEL

The adaptive matched filter will be implemented as a tapped-delay-line. Figure 2-4 shows the implementation. In general, there are a total of N samples per symbol waveform, where N is an integer. In this thesis $N=8$ and the vector length of $\underline{x}(k)$ will be 8. $Y(k)$ is sampled at

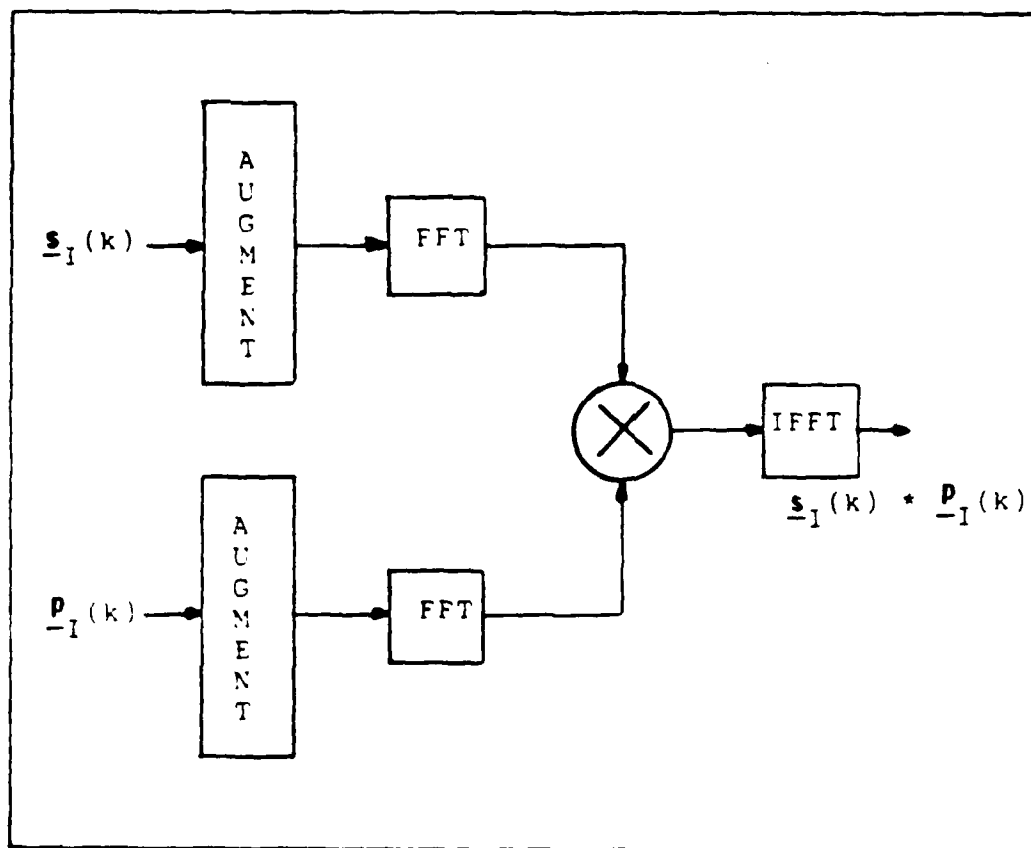


Figure 3-4 FFT Implementation

time $t=T$, which is also the time required to justify $\underline{x}$ within the matched filter. Thus one sample of $Y(k)$ occurs for each vector input $\underline{x}$.

The weight vector $\underline{w}(k)$, which is composed of w_1 thru w_N , is adaptive so that the individual weights can be updated through the adaptive algorithm.

3-5 GRIFFITHS ALGORITHM MODEL

The complex Griffiths algorithm was derived previously and is given by

$$\underline{w}(k+1) = \underline{w}(k) + 2\mu \underline{P} - 2\mu Y(k) \underline{x}^*(k) \quad (24)$$

where $\underline{P} = E\{ D(k) \underline{x}^*(k) \}$

All terms in the algorithm are known apriori or can be measured directly from observed data. The P-vector requires further elaboration to gain a full understanding.

The complex P-vector is defined to be

$$\underline{P} = E\{ D(k) \underline{x}^*(k) \} \quad (25)$$

where

E = the statistical expectation operator

$D(k)$ = the complex desired signal

$\mathbf{r}^*(k)$ = the complex conjugate of the received vector

$D(k)$, for a communication application, is a complex constant determined by the initial receiver design and signal constellation. With this assumption the P-vector can be written as

$$\mathbf{P} = D(k) \mathbf{E}\{ \mathbf{r}^*(k) \} \quad (26)$$

Assuming now that we have no interference and using notation consistent with Figure 3-1, the P-vector can be written as

$$\mathbf{P} = D(k) \mathbf{E}\{ \mathbf{r}_I(k) - j \mathbf{r}_Q(k) \} \quad (27)$$

$$\mathbf{P} = D \mathbf{E}\{ (\mathbf{z}_I + \mathbf{n}_I) - j (\mathbf{z}_Q + \mathbf{n}_Q) \} \quad (28)$$

where the discrete time index, k , has been dropped for convenience. Assuming that the noise is a zero mean process, the P-vector becomes

$$\mathbf{P} = D [\mathbf{E}\{ \mathbf{z}_I \} - j \mathbf{E}\{ \mathbf{z}_Q \}] \quad (29)$$

$$\mathbf{P} = (D_I + j D_Q) [\mathbf{E}\{ \mathbf{z}_I \} - j \mathbf{E}\{ \mathbf{z}_Q \}] \quad (30)$$

The values $\mathbf{z}_I$ and $\mathbf{z}_Q$ are outputs of the composite channel filter. It can be shown (13:172) that for a causal, time-invariant linear filter, the mean of the output is given by

$$m_Y(k) = m_X(k) * h(k) \quad (31)$$

where

$m_Y(k)$ = the mean of the output

$m_X(k)$ = the mean of the input

$h(k)$ = a channel's finite impulse response

Therefore, the expectations given in equation (30) can be written as

$$E\{\underline{z}_I\} = \underline{m}_{S_I} * \underline{p}_I \quad (32)$$

$$E\{\underline{z}_Q\} = \underline{m}_{S_I} * \underline{p}_Q \quad (33)$$

Equations (24), (30), (32), and (33) define the Griffiths algorithm as implemented in the simulation.

3-6 INTERFERENCE AND NOISE MODEL

Both interference and noise will be modeled as complex, low-pass, baseband signals. Therefore, both can be defined as

$$\underline{i}(k) = \underline{i}_I(k) + j \underline{i}_Q(k) \quad (34)$$

$$\underline{n}(k) = \underline{n}_I(k) + j \underline{n}_Q(k) \quad (35)$$

The noise process will be approximated as an additive white gaussian process with zero mean. In order to calculate one element of the noise matrix the following equation will be used

$$n(k) = \sigma \left[\sum_{i=1}^{12} U_i - 6 \right] \quad (36)$$

where

$n(k)$ = a gaussian noise value

σ = the standard deviation of the noise process

U_i = a uniformly distributed random variable
between 0 and 1

Equation (36) illustrates how the noise is generated in the computer simulation. The basis for equation (36) lies in the central limit theorem. This theorem states that the sum of N independent, identically distributed random variables becomes a Gaussian distribution as N becomes large (18:243). In equation (36) $N=12$, and the Gaussian approximation is based on the sum of 12 uniform random variables. If the noise is also assumed to be ergodic (see Appendix B) then the standard deviation is related to the noise power by

$$\sigma = \sqrt{\text{noise power}} \quad (37)$$

Equation (37) enables one to specify a noise power and generate noise at this power level.

The interference will also be additive. The interference will be narrowband and be in the same frequency band as the signal of interest; therefore, $i(k)$ takes the form of equation (34). The in-phase and quadrature components can be represented as

$$i_I(k) = \sqrt{2 \text{ JPWR}} \cos[2\pi F_J T_J k + 2\pi c1] \quad (38)$$

$$i_Q(k) = \sqrt{2 \text{ JPWR}} \sin[2\pi F_J T_J k + 2\pi c1] \quad (39)$$

where

JPWR = the jamming power

F_J = the frequency of the jammer

T_J = the period

k = the discrete time index

$c1$ = the phase component

Equations (38) and (39) define the interference model as used in the simulation.

3-7 GENERAL SIMULATION MODEL

The purpose of this thesis is to analyze the performance of a complex Griffiths adaptive matched filter for various input signals. The software chosen for this analysis is MathCAD developed by Mathsoft, Inc. A complete listing of the simulation software program is found in Appendix A. The general procedure for the simulation is as follows

- STEP 1: define $\underline{s}(k)$.
- STEP 2: define $\underline{p}_I(k)$ and $\underline{p}_Q(k)$.
- STEP 3: define $D(k)$, the desired response.
- STEP 4: define $\underline{i}_I(k)$, $\underline{i}_Q(k)$, $\underline{n}_I(k)$, $\underline{n}_Q(k)$.
- STEP 5: input the defined signals and run the simulation.
- STEP 6: compare the output $Y(k)$ against the various defined input signals
- STEP 7: compare the performance of the adaptive filter against the various defined input signals.

Note that the simulation requires the user to define the various input sequences.

Throughout the development of the thesis model various assumptions were made concerning the filter and model. All assumptions are repeated here for compactness:

1) input signal is at least cyclostationary and sampled at or above the Nyquist rate. The input signal is real, has odd length, and is antipodal.

2) synchronization is maintained during all operations.

3) all signals are modeled as narrowband, complex, low-pass, baseband signals.

4) only filtering operations are considered.

5) all filters are stable, causal, FIR filters.

6) channel filter is linear, time-invariant.

7) the adaptive filter is implemented as a FIR structure and uses closed loop adaptation.

8) complex Griffiths algorithm is implemented

9) noise and interference are both additive.

Interference has a zero mean.

10) noise is approximated as a zero mean, independent, additive white gaussian noise process.

11) noise, interference, and input signal are ergodic.

CHAPTER 4 : ANALYSIS OF RESULTS

4-1 INTRODUCTION

This chapter analyzes performance of the Griffiths adaptive matched filter over a range of signal inputs. All notation will be consistent with that of Figures 3-1 and 3-2. First, several performance criteria are established. This discussion will lay the fundamental framework for the jamming analysis that follows. The jamming analysis will be conducted over a range of signal distortion levels and jamming to signal power ratios (JSR).

4-2 PERFORMANCE CRITERIA

4-2-1 POWER CRITERIA

Throughout the analysis the terms signal power, noise power, and jamming power will be used. Signal power will be the power in $\underline{s}(k)$, noise power will be the power in $\underline{n}(k)$, and jamming power will be the power in $\underline{j}(k)$.

4-2-2 FIXED CRITERIA

In order to make comparisons of performance, certain parameters need to be held fixed or constant.

Throughout the analysis the following constants will be assumed

$$\mu = .01$$

$$D = 1 + j$$

$$\text{signal power} = .375$$

$$\text{noise power} = 0$$

$$\underline{s}(k) = \text{the same sequence of binary data}$$

The reason for setting the noise power to zero is to explicitly analyze the effects of jamming and delay distortion on the filter.

The convergence rate constant, μ , is set equal to .01 so that jamming power levels can be set equal to -10 DB, 0 DB, and 10 DB. In general, the convergence rate constant should satisfy (8:103)

$$\mu < \frac{1}{(\text{number of tap weights})(\text{received power})} \quad (40)$$

rearranging and substituting

$$\text{received power} < \frac{1}{(8) (.01)} \quad (41)$$

$$\text{received power} < 12.5 \quad (42)$$

Since the transmitted signal power is .375, the combined noise and jamming power must be no greater than 12.125. This implies that the maximum JSR is approximately 15 DB. With the convergence rate constant equal to .01, analysis of JSRs equal to -10 DB, 0 DB, and 10 DB is possible.

$\underline{s}(k)$ will be a real bit stream of length 31.
 $\underline{s}(k)$ will be generated as a pseudonoise sequence of length 31 and take the following form in binary data

1000 0100 1011 0011 1110 0011 0111 010

The sequence for $\underline{s}(k)$ was generated from a 5-stage linear feedback shift register as shown in Figure 4-1 with an initial shift register state of 00001. The sequence for $\underline{s}(k)$ is of maximal length and can be shown to obey the properties of pseudonoise sequences such as balance, run, and correlation (11:546). A plot of the normalized autocorrelation function is shown in Figure 4-2.

4-2-3 AVERAGE ERROR CRITERION

For this thesis the error will be defined as

$$e(k) = \sqrt{[\text{Re}(D) - \text{Re}(Y) \text{PN}_{\text{bit}}]^2 + [\text{Im}(D) - \text{Im}(Y) \text{PN}_{\text{bit}}]^2} \quad (43)$$

where

$e(k)$ = the error signal

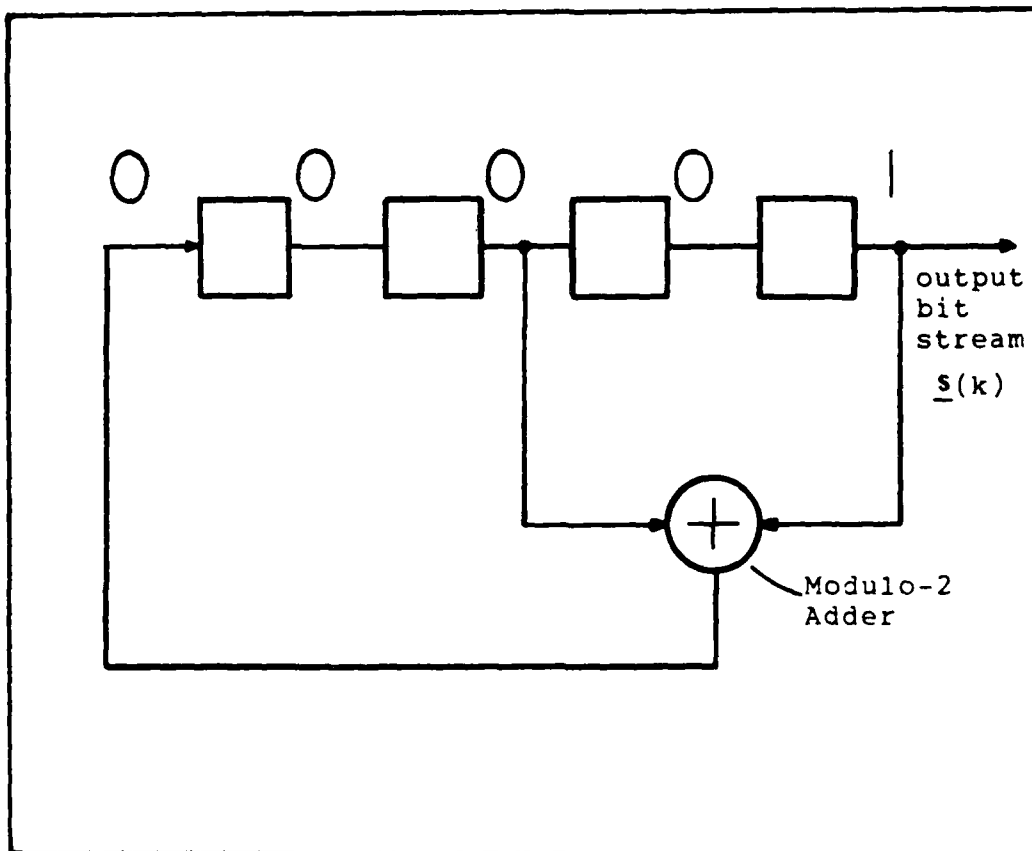


Figure 4-1 5-Stage Linear Feedback Shift Register

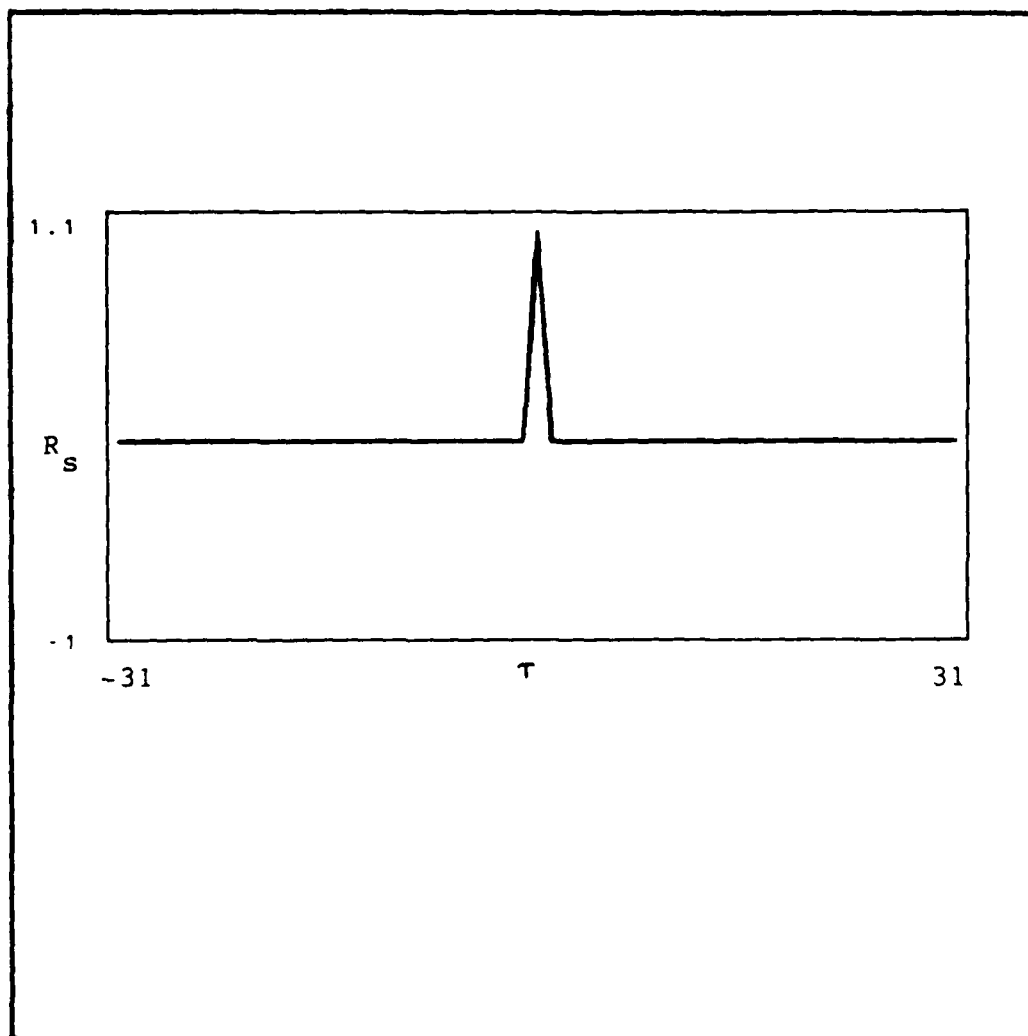


Figure 4-2 Normalized Autocorrelation Function
of $\underline{s}(k)$

D = the desired signal

Y = the actual filter output

PN_{bit} = the bit transmitted, +1 or -1

$e(k)$ is simply the distance between D and $Y PN_{bit}$. If $e(k)$ is the actual error at a specific adaptation cycle, then the average error will be defined as

$$\Gamma = \frac{1}{N} \sum_{k=86}^{185} e(k) \quad (44)$$

where

N = the number of adaptation cycles (100)

Γ = the average error

The average error will represent the average of the error over the last 100 adaptations. This is done to minimize transient effects.

To make a comparison to other input signals the number of adaptations will be fixed at 186. 186 was chosen because after this many adaptations the filter has settled down well into its steady state for $\mu=.01$ (for instance, see Figure 4-7).

4-2-4 DEFINING $p_I(k)$ AND $p_Q(k)$

The in-phase and quadrature response of the composite channel filter are defined by equations (20) and

(21). For this thesis we will be concerned with four cases.
These cases are

- CASE I : no distortion
- CASE II : linear delay distortion
- CASE III : quadratic delay distortion
- CASE IV : cubic delay distortion

The representations for $p_I(k)$ and $p_Q(k)$ are given for each of the four cases in Figures 4-3 thru 4-6. The amount of delay distortion is set at $k/T = 2$.

4-2-5 DEFINING THE P-VECTOR

The P-vector was defined in equation (25). One of the requirements of the Griffiths algorithm is that the P-vector is determined apriori (4:1699). For this thesis the P-vector will be determined from Figure 4-3 for the case of no distortion. With this assumption the P-vector becomes

$$\underline{P} = D \begin{bmatrix} 0 \\ .17 \\ .5 \\ .849 \\ 1 \\ .849 \\ .5 \\ .17 \end{bmatrix}$$

Note that the P-vector is determined from the main lobe of the in-phase channel and consists of 8 sample values. The

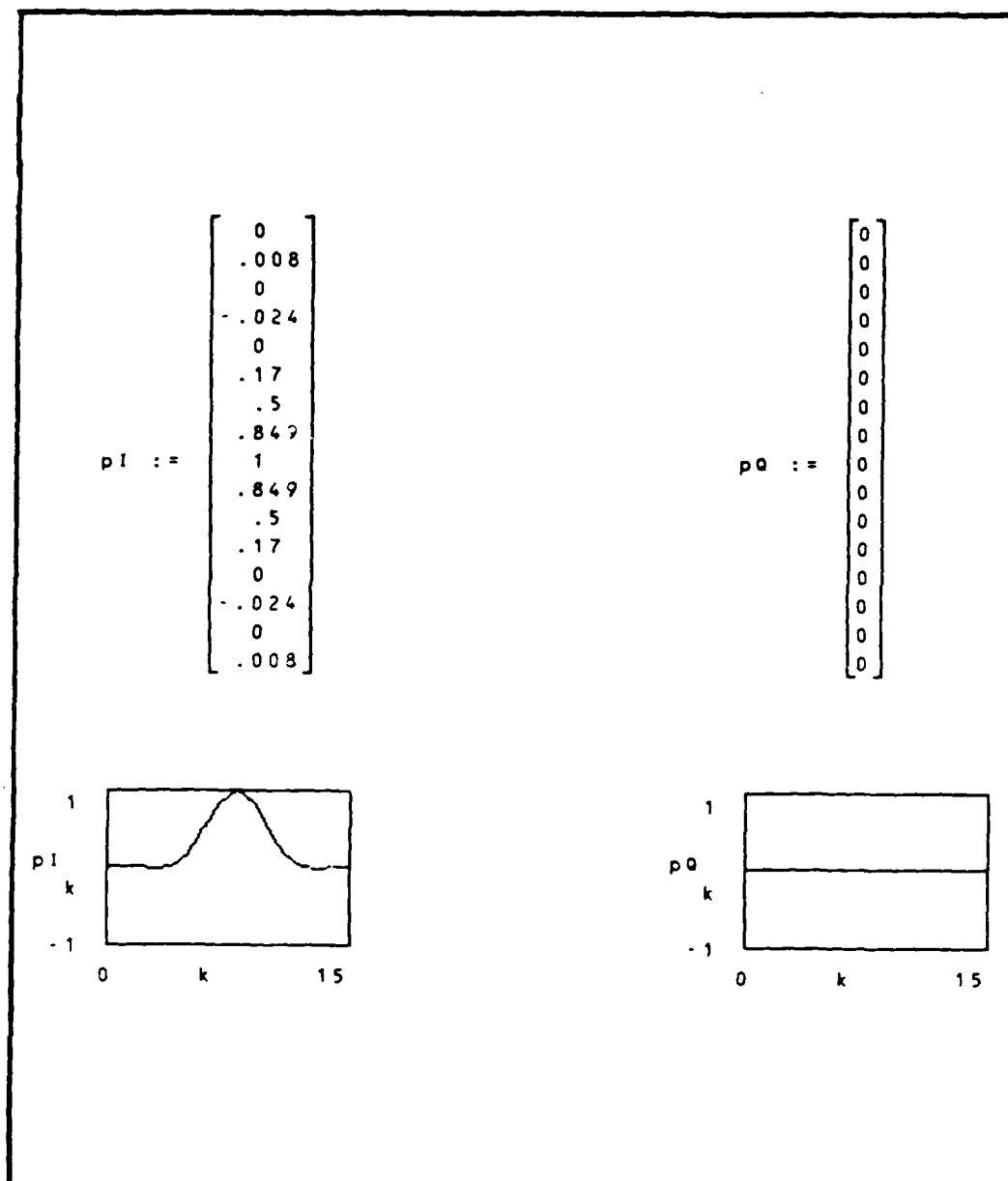


Figure 4-3 Case I. Zero Distortion.
 Calculated from Equations
 (20) and (21) with $T=.5$, $k_0=0$,
 $k_1=2$, $k_2=0$, $k_3=0$, $k_4=0$.

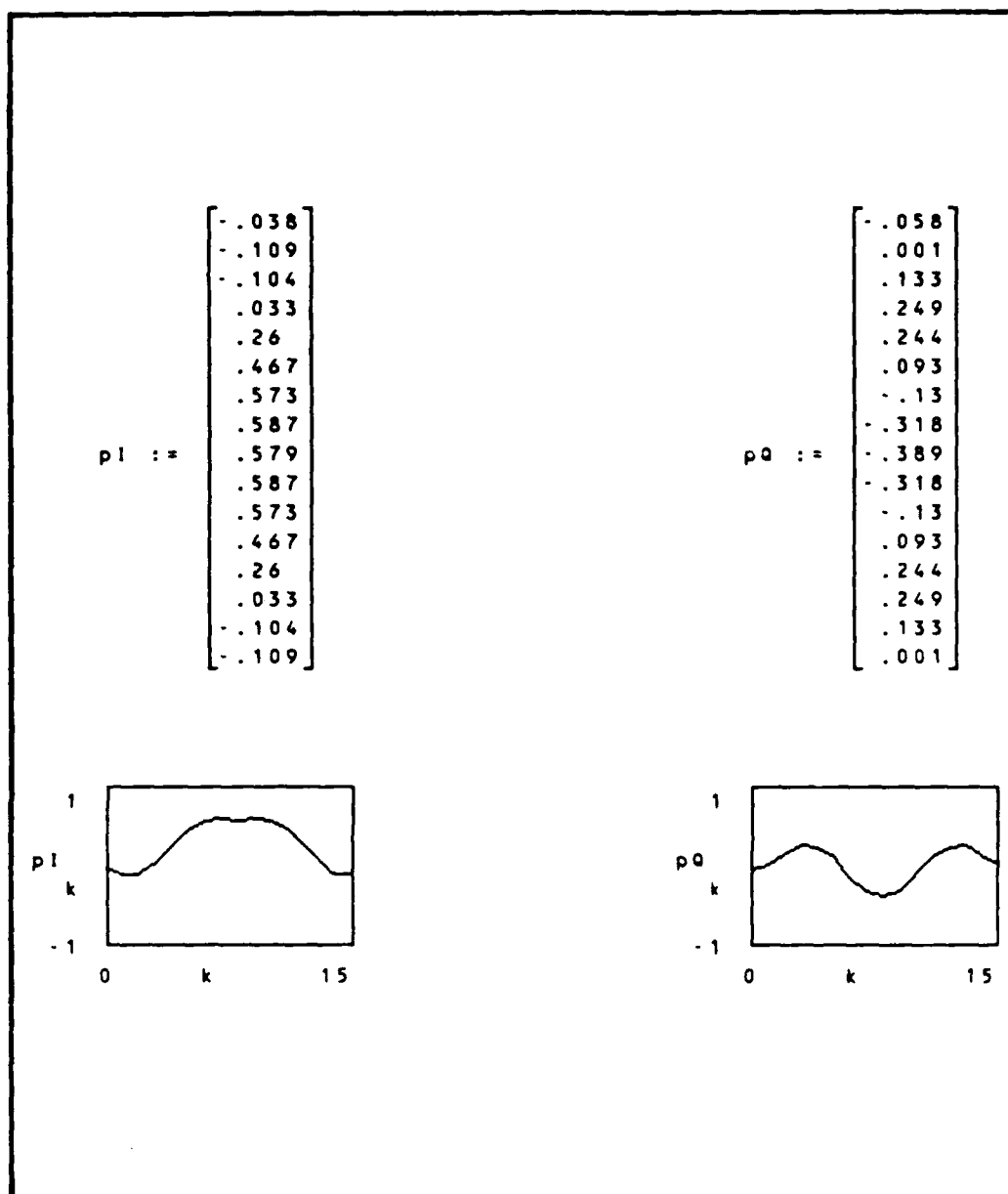


Figure 4-4 Case II. Linear Delay Distortion.
Calculated from Equations
(20) and (21) with $T=.5$, $k_0=0$,
 $k_1=2$, $k_2=1$, $k_3=0$, $k_4=0$.

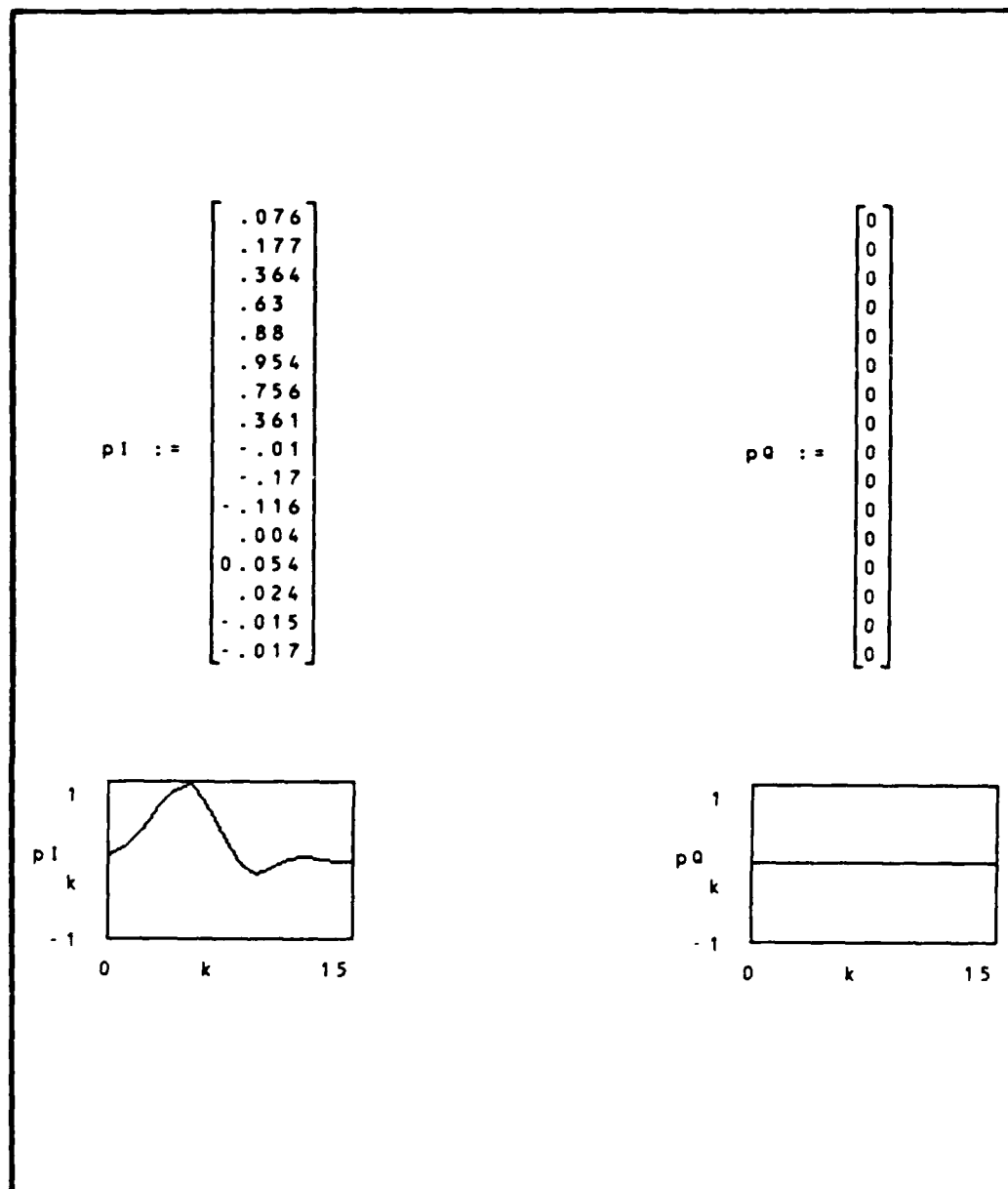


Figure 4-5 Case III. Quadratic Delay Distortion.
 Calculated from Equations
 (20) and (21) with $T=.5$, $k_0=0$,
 $k_1=2$, $k_2=0$, $k_3=1$, $k_4=0$.

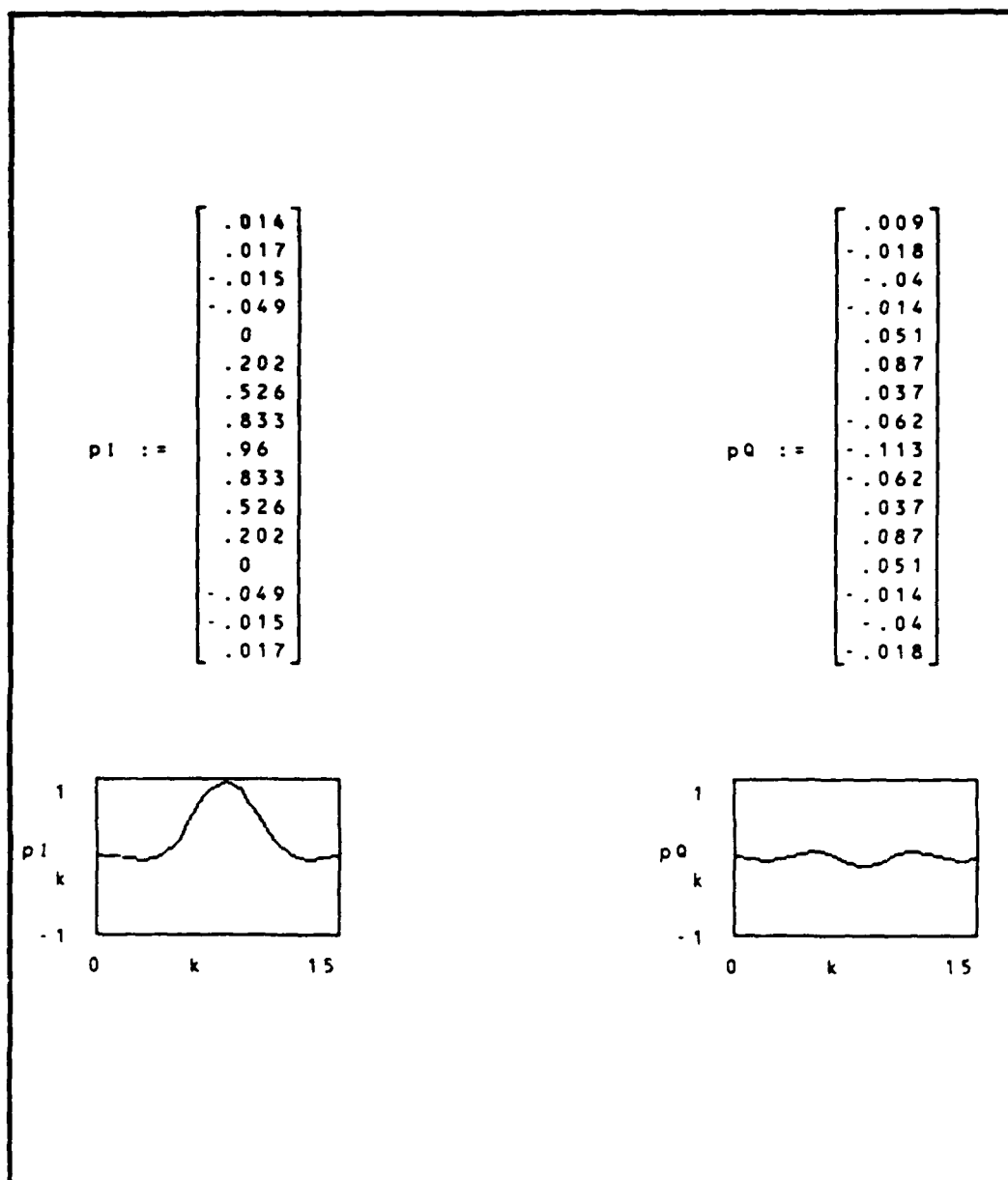


Figure 4-6 Case IV. Cubic Delay Distortion.
 Calculated from Equations
 (20) and (21) with $T=.5$, $k_0=0$,
 $k_1=2$, $k_2=0$, $k_3=0$, $k_4=1$.

P-vector has length 8 because this is the length of the filter.

4-3 JAMMING RESULTS

This section will present the results of the simulation. The first section will present the effects of jamming power and jamming frequency on the filter output. The second section will present the effects of jamming phase on the filter output. Subsequent sections will compare CW jamming and noise jamming and make comparisons to the LMS algorithm.

4-3-1 EFFECTS OF JAMMING POWER AND JAMMING FREQUENCY (PHASE CONSTANT)

The first set of results is that for zero jamming power and zero noise power. These results are presented in Figures 4-7 thru 4-11. The actual values for the average error are presented in Tables 1 thru 3 for a JSR = $-\infty$. Note that pulses that experience linear delay distortion will not enable the algorithm to converge. This will be explained further in the COMPARISONS section.

The results of the simulation dealing with jamming power and jamming frequency (phase constant) are presented in Tables 1 thru 3. A graphical presentation of these results are presented in Figures 4-12 thru 4-14.

NO JAMMING No Distortion

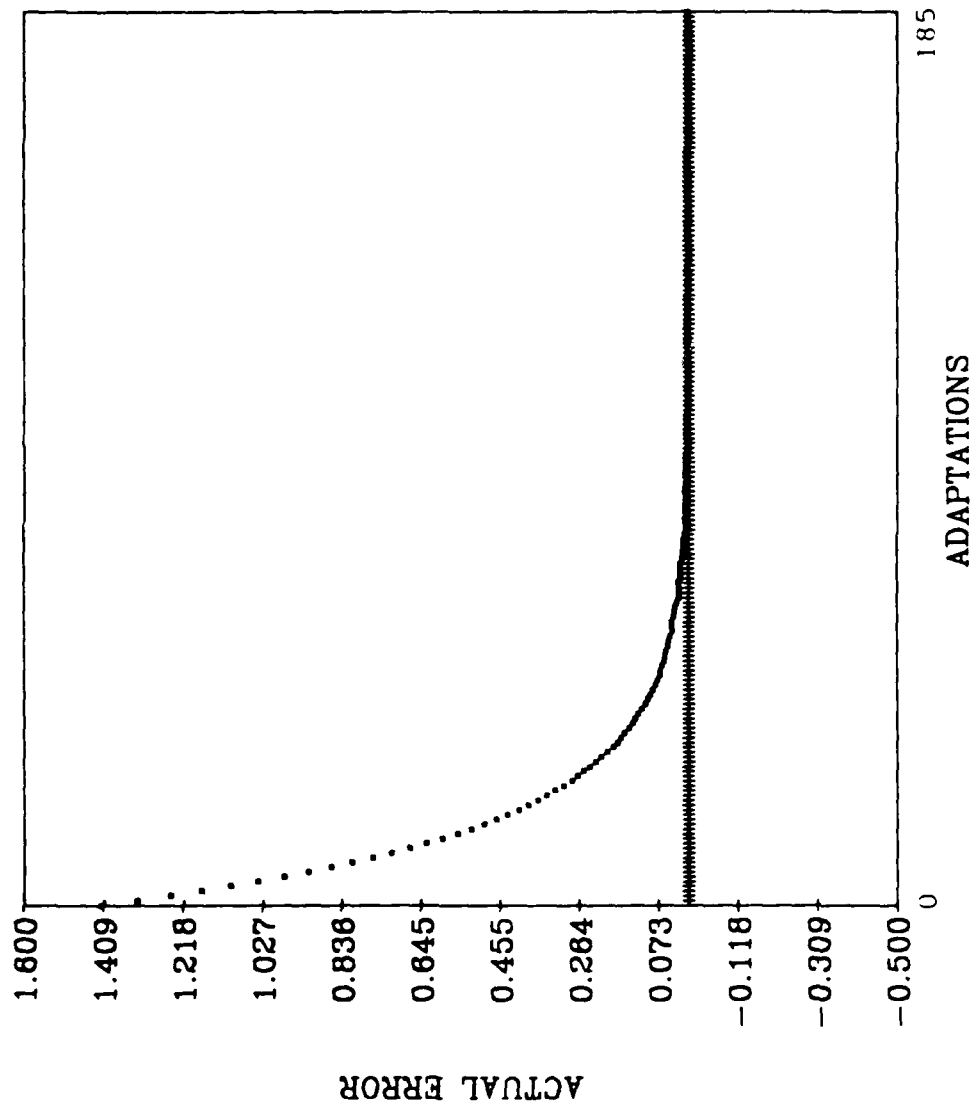


Figure 4-7 No Jamming/No Distortion

NO JAMMING

Linear Delay

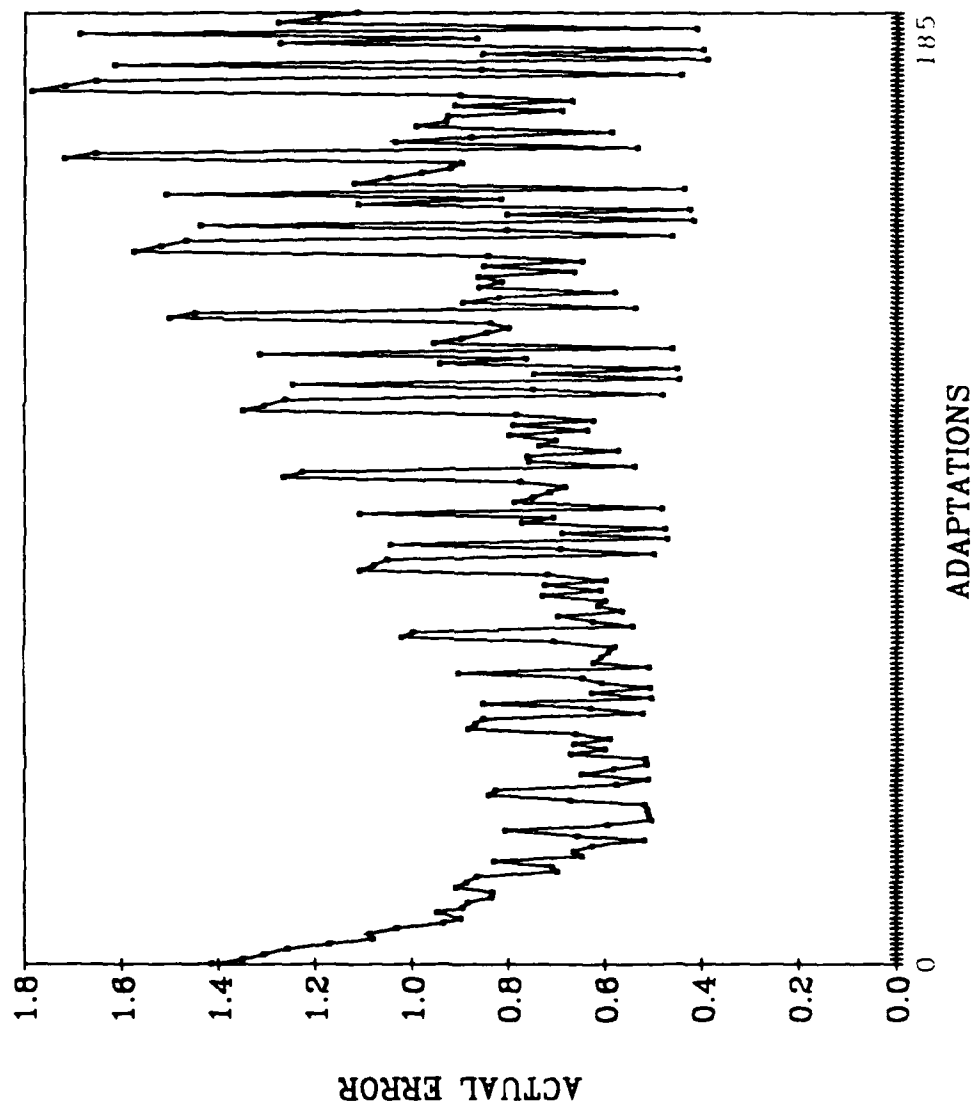


Figure 4-8 No Jamming/Linear Delay Distortion

NO JAMMING

Quadratic Delay

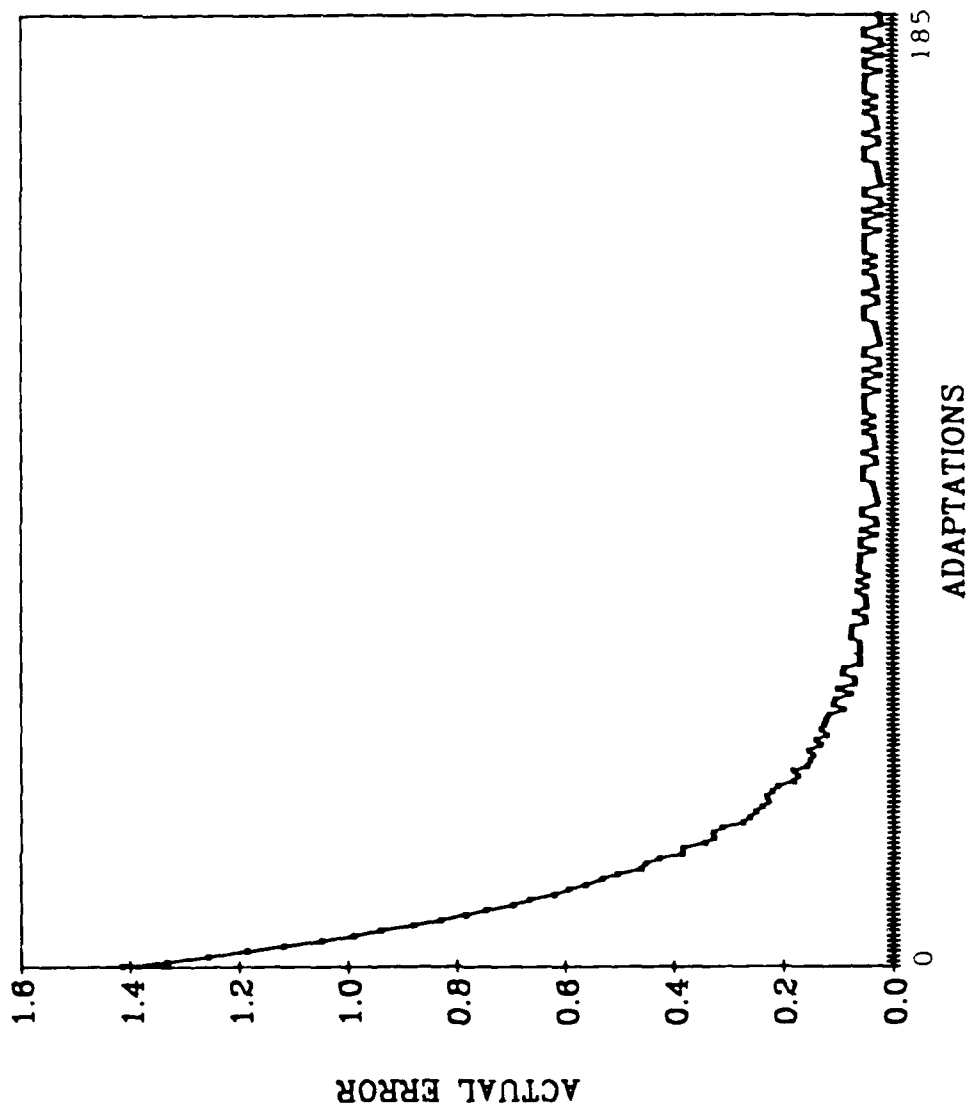


Figure 4-9 No Jamming/Quadratic Delay Distortion

NO JAMMING

Cubic Delay

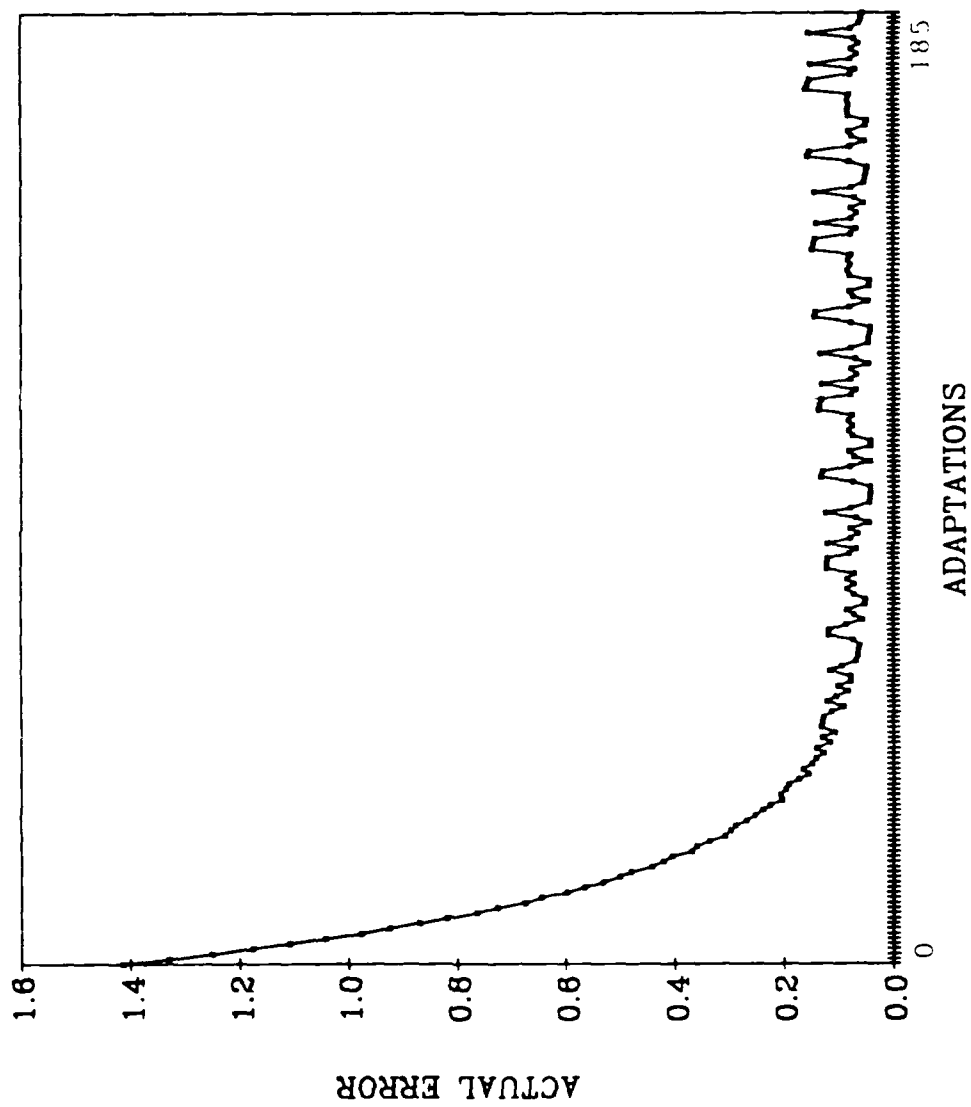


Figure 4-10 No Jamming/Cubic Delay Distortion

NO JAMMING

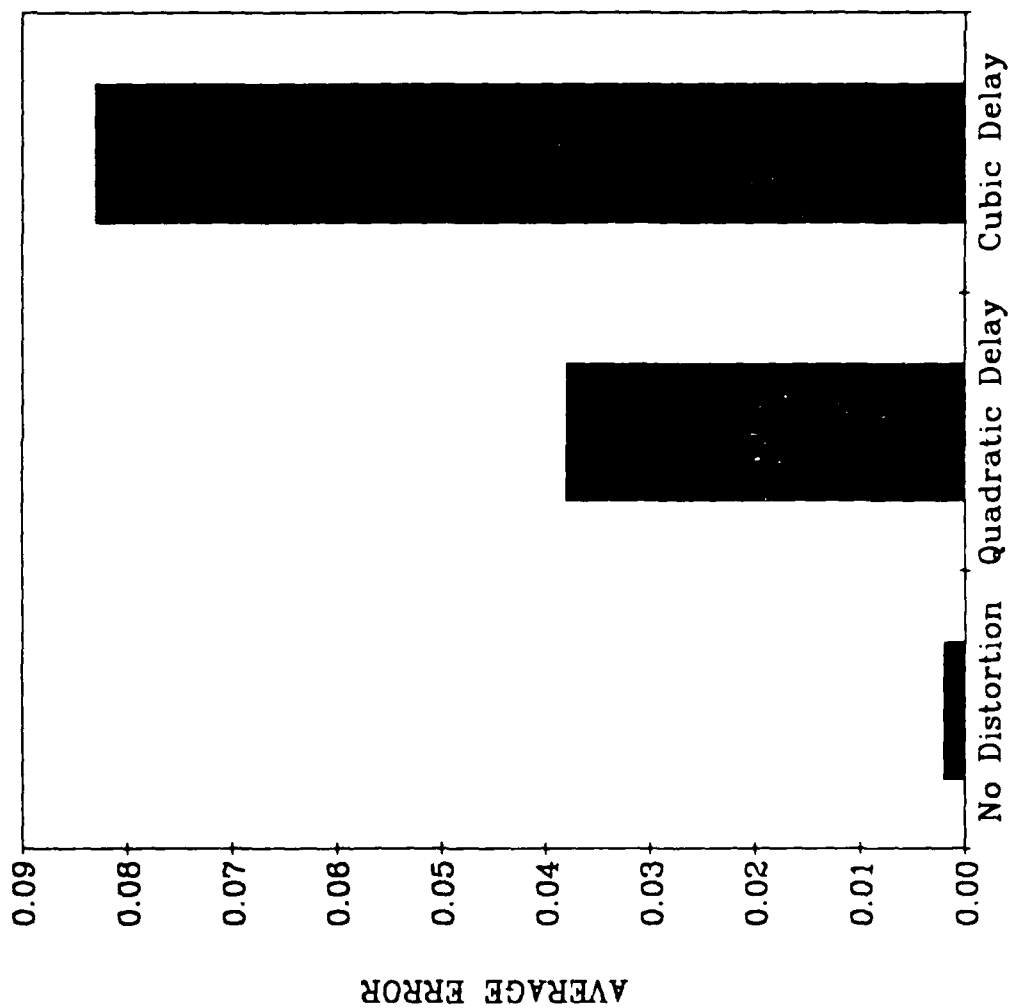


Figure 4-11 Bar Graph Comparing Various Delay Distortion Levels

TABLE 1 Average error of filter output for
an input that has no distortion

JSR (DB)	JAMMING CYCLES per PN LENGTH	AVERAGE ERROR (Γ)
$-\infty$	-	.002
-10	1	.332
0	1	.371
10	1	.744
-10	2	.26
0	2	.296
10	2	.751
-10	8	.2
0	8	.237
10	8	.896
-10	31	.079
0	31	.127
10	31	.799

Note: Noise power is zero and the phase of the jammer is zero

TABLE 2 Average error of filter output for
an input that has quadratic delay
distortion

JSR (DB)	JAMMING CYCLES per PN LENGTH	AVERAGE ERROR (Γ)
$-\infty$	-	.038
-10	1	.319
0	1	.416
10	1	.623
-10	2	.28
0	2	.376
10	2	.647
-10	8	.318
0	8	.425
10	8	.911

Note: Noise power is zero and the phase of the jammer is
zero

TABLE 3 Average error of filter output for
an input that has cubic delay distortion

JSR (DB)	JAMMING CYCLES per PN LENGTH	AVERAGE ERROR (Γ)
$-\infty$	-	.083
-10	1	.369
0	1	.403
10	1	.76
-10	2	.304
0	2	.352
10	2	.751
-10	8	.262
0	8	.325
10	8	.888

Note: Noise power is zero and the phase of the jammer is zero

JSR = -10 DB

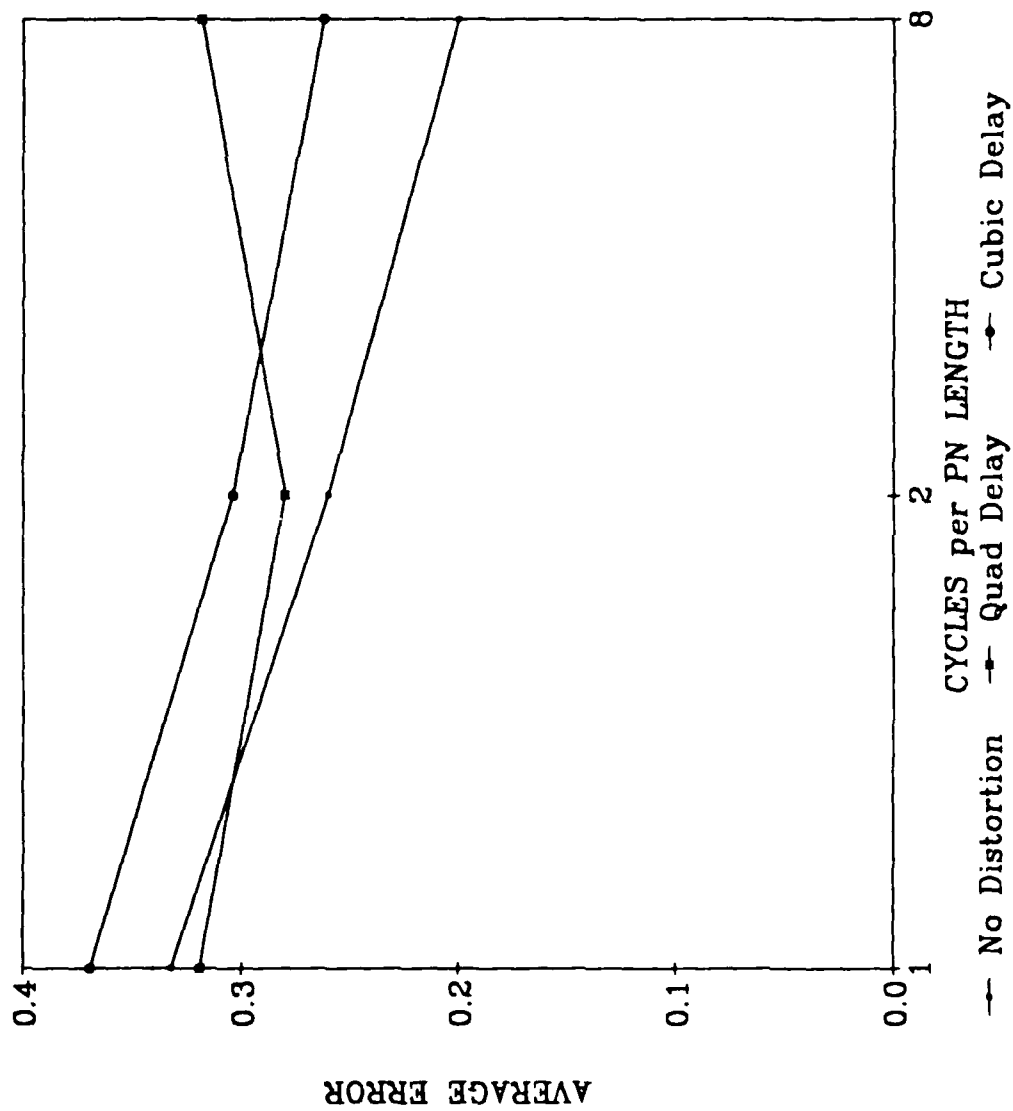


Figure 4-12 -10 DB Plot of Jamming Results (Phase=0)

JSR = 0 DB

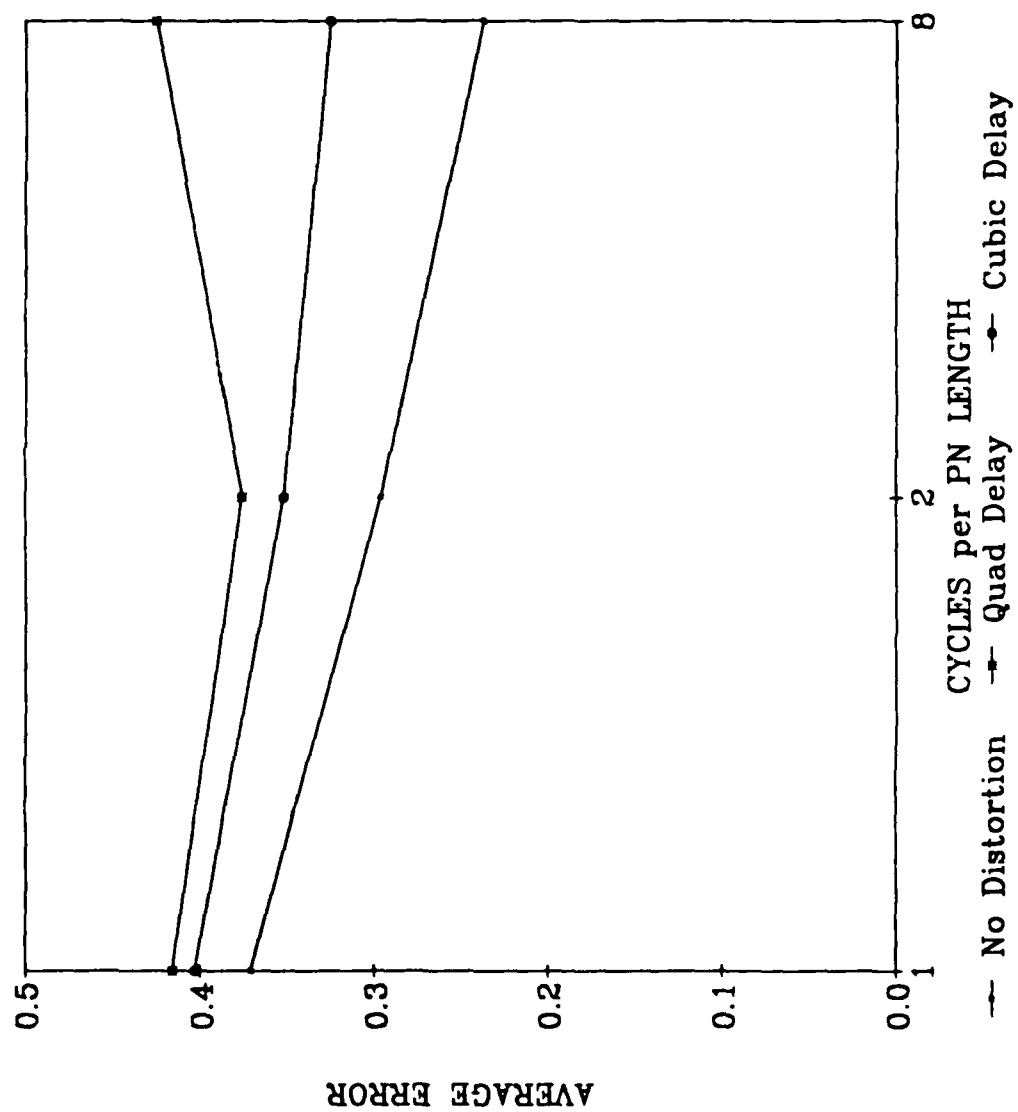


Figure 4-13 0 DB Plot of Jamming Results (Phase=0)

JSR = 10 DB

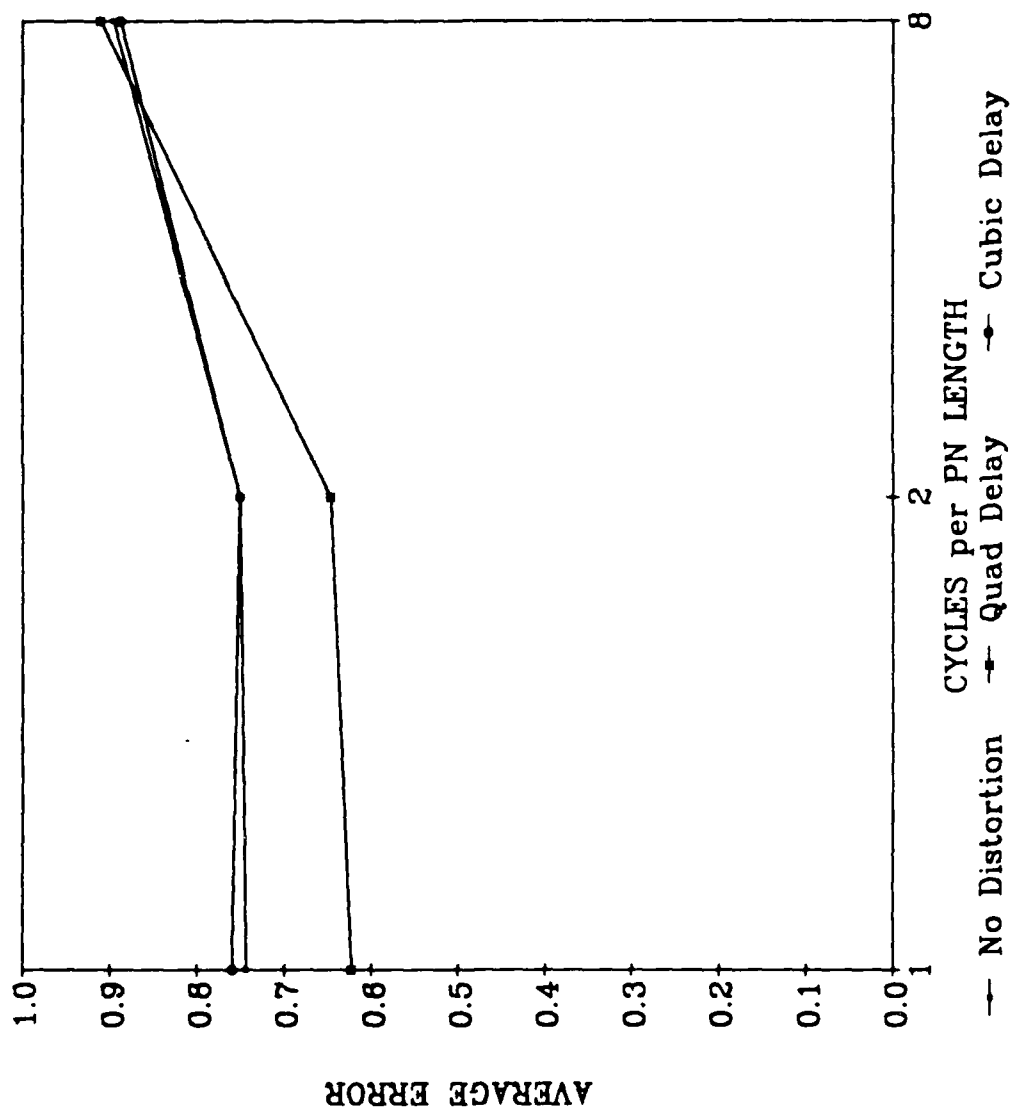


Figure 4-14 10 DB Plot of Jamming Results (Phase=0)

4-3-2 EFFECTS OF JAMMING PHASE (FREQUENCY AND POWER CONSTANT)

The effects of jamming phase at 0 DB and -10 DB are presented in Tables 4 thru 9. Figures 4-15 thru 4-20 are a graphical presentation of these results.

4-3-3 EFFECTS OF NOISE ON FILTER

In order to compare between noise jamming and continuous wave jamming, a one-time run was made on a signal that has no distortion at various signal-to-noise ratios (SNR). These results are presented in Table 10. Actual error output due to various SNRs is graphically depicted in Figures 4-21 thru 4-23.

4-3-4 LMS ALGORITHM

The purpose of this thesis has been to analyze an adaptive matched filter that uses the Griffiths algorithm. For comparison several selected runs were repeated using the LMS algorithm. The LMS algorithm is implemented as

$$\underline{w}(k+1) = \underline{w}(k) + 2\mu P N_{bit} D \underline{x}^*(k) - 2\mu Y(k) \underline{x}^*(k) \quad (45)$$

where all terms have been defined previously. The LMS runs are presented in Figures 4-24 thru 4-27 and Table 11.

TABLE 4 Average error of filter output for
different phase constants (c1).
JSR=0 DB and Frequency = 1 cycle per
PN Length.

<i>No Distortion</i>	<i>c1</i>	<i>Γ</i>
	0	.371
	.25	.263
	.5	.273
	.75	.386
<i>Quadratic Delay</i>		
	0	.416
	.25	.342
	.5	.313
	.75	.393
<i>Cubic Delay</i>		
	0	.403
	.25	.349
	.5	.434
	.75	.513

TABLE 5 Average error of filter output for
different phase constants (c1).
JSR=0 DB and Frequency = 2 cycles per
PN Length.

No Distortion	c1	Γ
	0	.296
	.25	.269
	.5	.367
	.75	.4
Quadratic Delay		
	0	.376
	.25	.334
	.5	.363
	.75	.431
Cubic Delay		
	0	.352
	.25	.417
	.5	.504
	.75	.441

TABLE 6 Average error of filter output for
different phase constants (c1).
JSR=0 DB and Frequency = 8 cycles per
PN Length.

No Distortion	c1	Γ
	0	.237
	.25	.309
	.5	.387
	.75	.325
<i>Quadratic Delay</i>		
	0	.425
	.25	.326
	.5	.378
	.75	.529
<i>Cubic Delay</i>		
	0	.325
	.25	.44
	.5	.455
	.75	.318

TABLE 7 Average error of filter output for
different phase constants (c1).
JSR=-10 DB and Frequency = 1 cycle per
PN Length.

No Distortion	c1	r
	0	.332
	.25	.241
	.5	.257
	.75	.341
Quadratic Delay		
	0	.319
	.25	.255
	.5	.251
	.75	.315
Cubic Delay		
	0	.369
	.25	.291
	.5	.313
	.75	.385

TABLE 8 Average error of filter output for
different phase constants (c1).
JSR=-10 DB and Frequency = 2 cycles per
PN Length.

No Distortion	c1	Γ
	0	.26
	.25	.234
	.5	.327
	.75	.344
Quadratic Delay		
	0	.28
	.25	.241
	.5	.295
	.75	.333
Cubic Delay		
	0	.304
	.25	.3
	.5	.371
	.75	.382

TABLE 9 Average error of filter output for
different phase constants (c1).
JSR=-10 DB and Frequency = 8 cycles per
PN Length.

<i>No Distortion</i>	<i>c1</i>	<i>r</i>
	0	.2
	.25	.275
	.5	.34
	.75	.279
<i>Quadratic Delay</i>		
	0	.318
	.25	.263
	.5	.344
	.75	.386
<i>Cubic Delay</i>		
	0	.262
	.25	.341
	.5	.387
	.75	.311

PHASE ERROR AT 0 DB 1 CYCLE per PN LENGTH

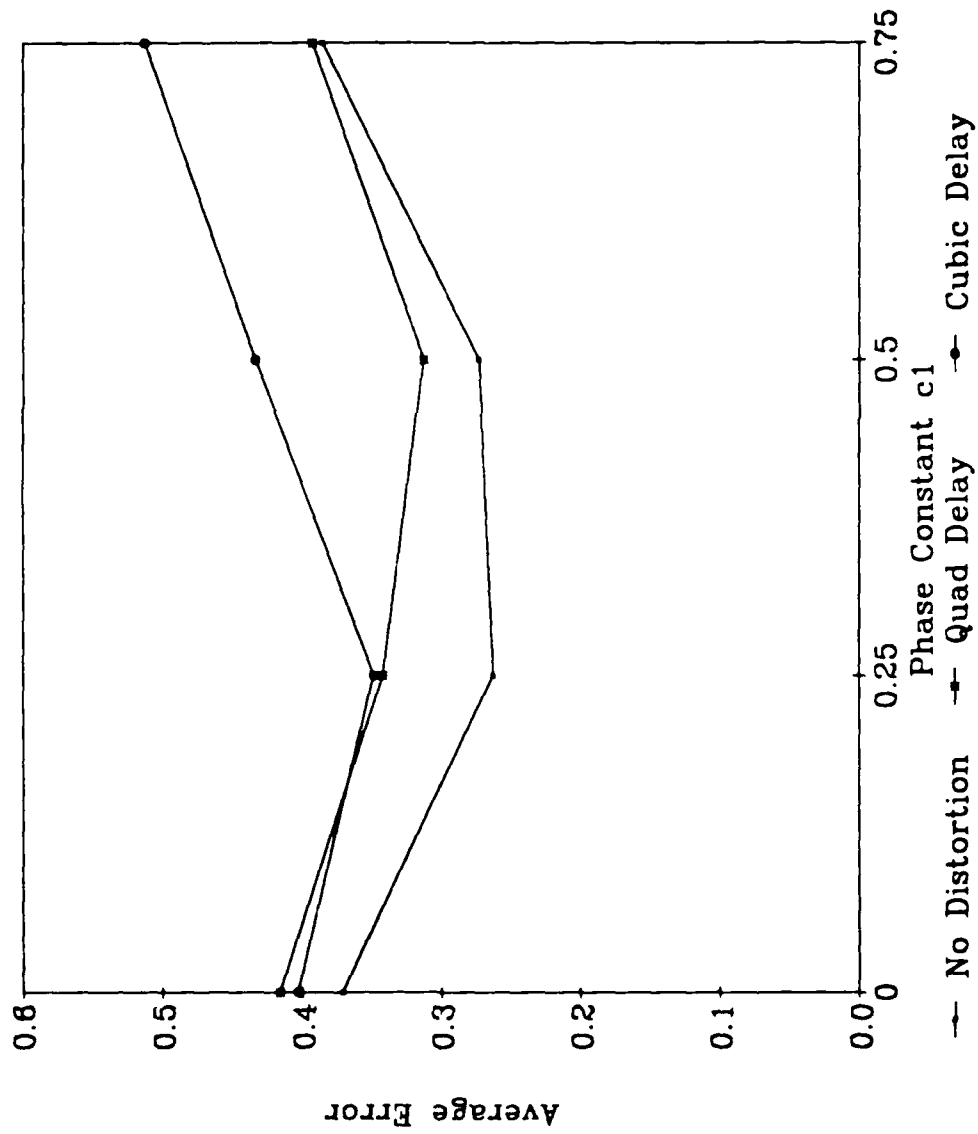


Figure 4-15 Average Error Due to Phase (0 DB and 1 Cycle per PN Length)

PHASE ERROR AT 0 DB 2 CYCLES per PN LENGTH

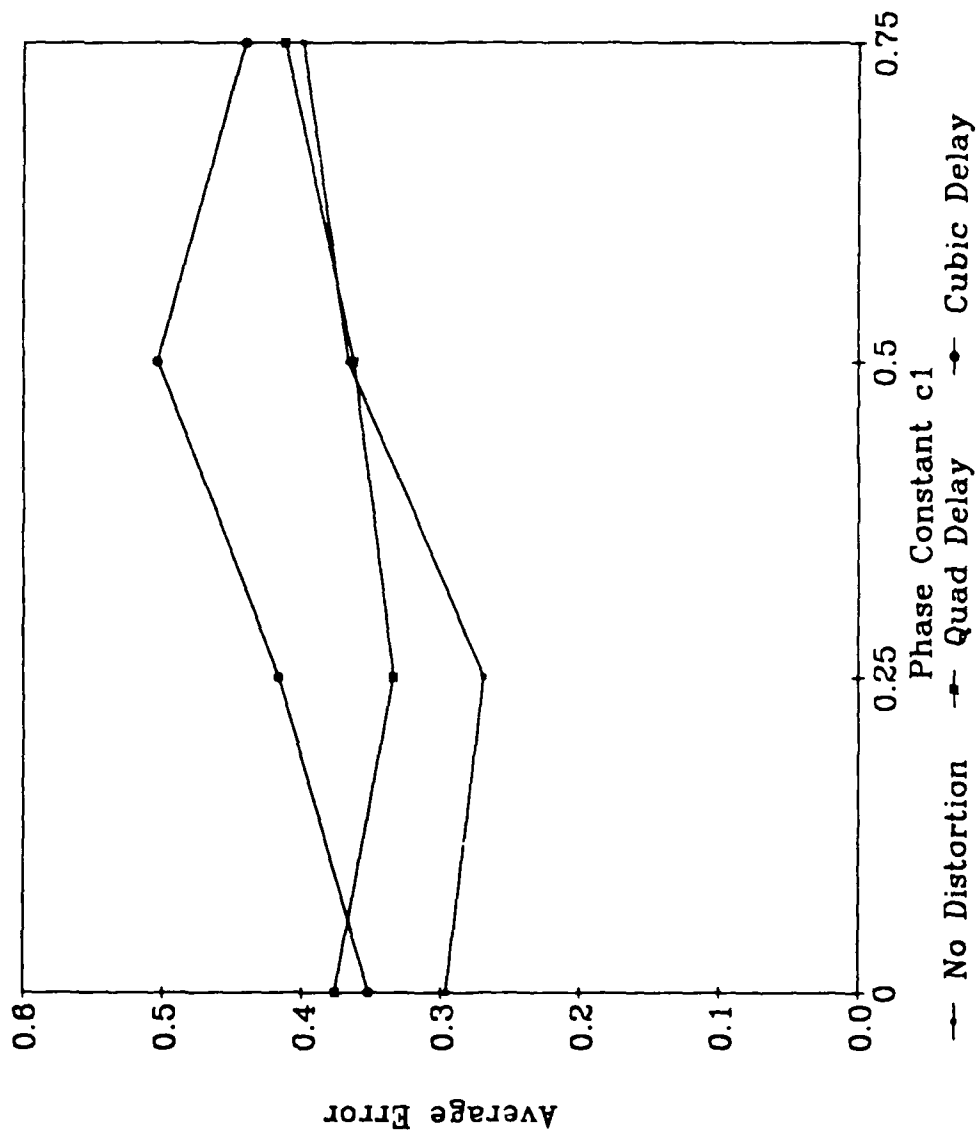


Figure 4-16 Average Error Due to Phase (0 DB and 2 Cycles per PN Length)

PHASE ERROR AT 0 DB 8 CYCLES per PN LENGTH

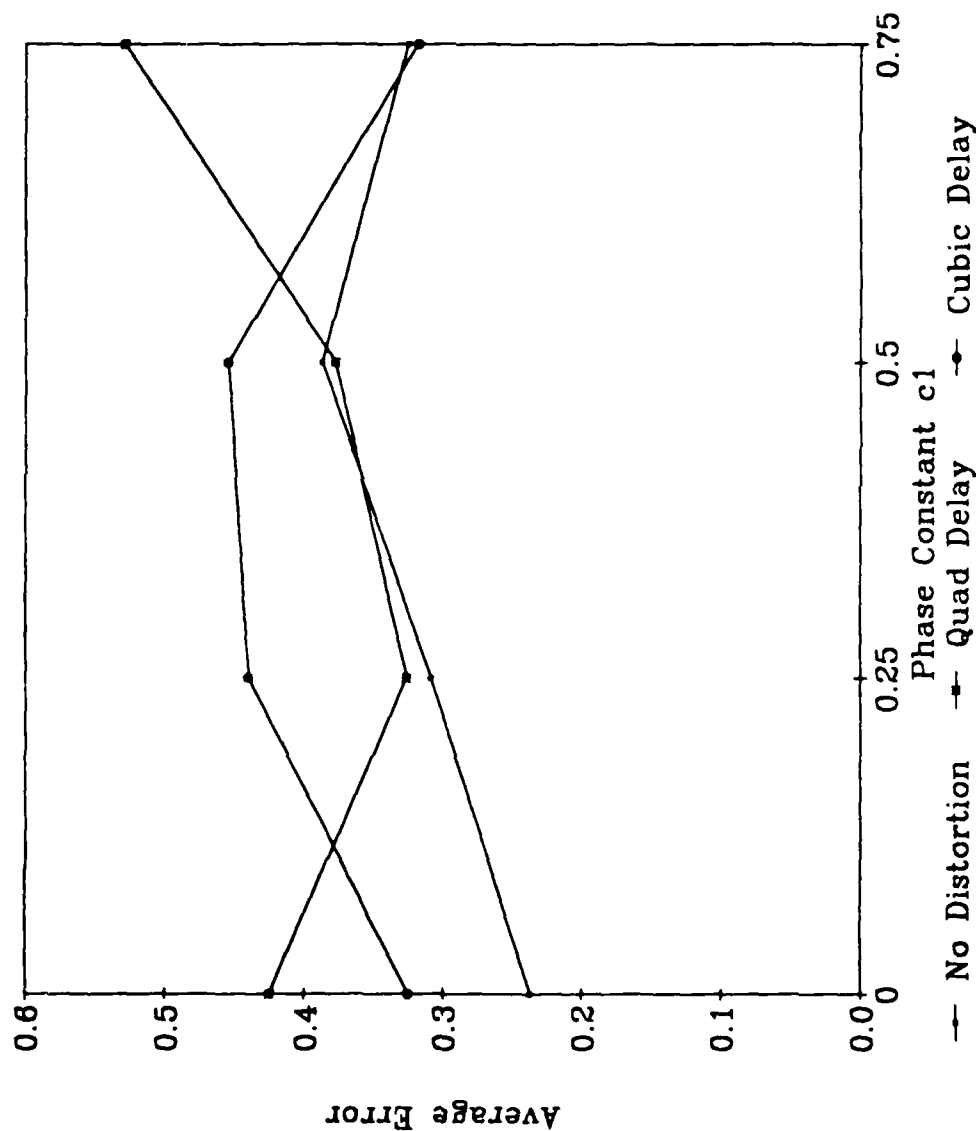


Figure 4-17 Average Error Due to Phase (0 DB and 8 Cycles per PN Length)

PHASE ERROR AT -10 DB 1 CYCLE per PN LENGTH

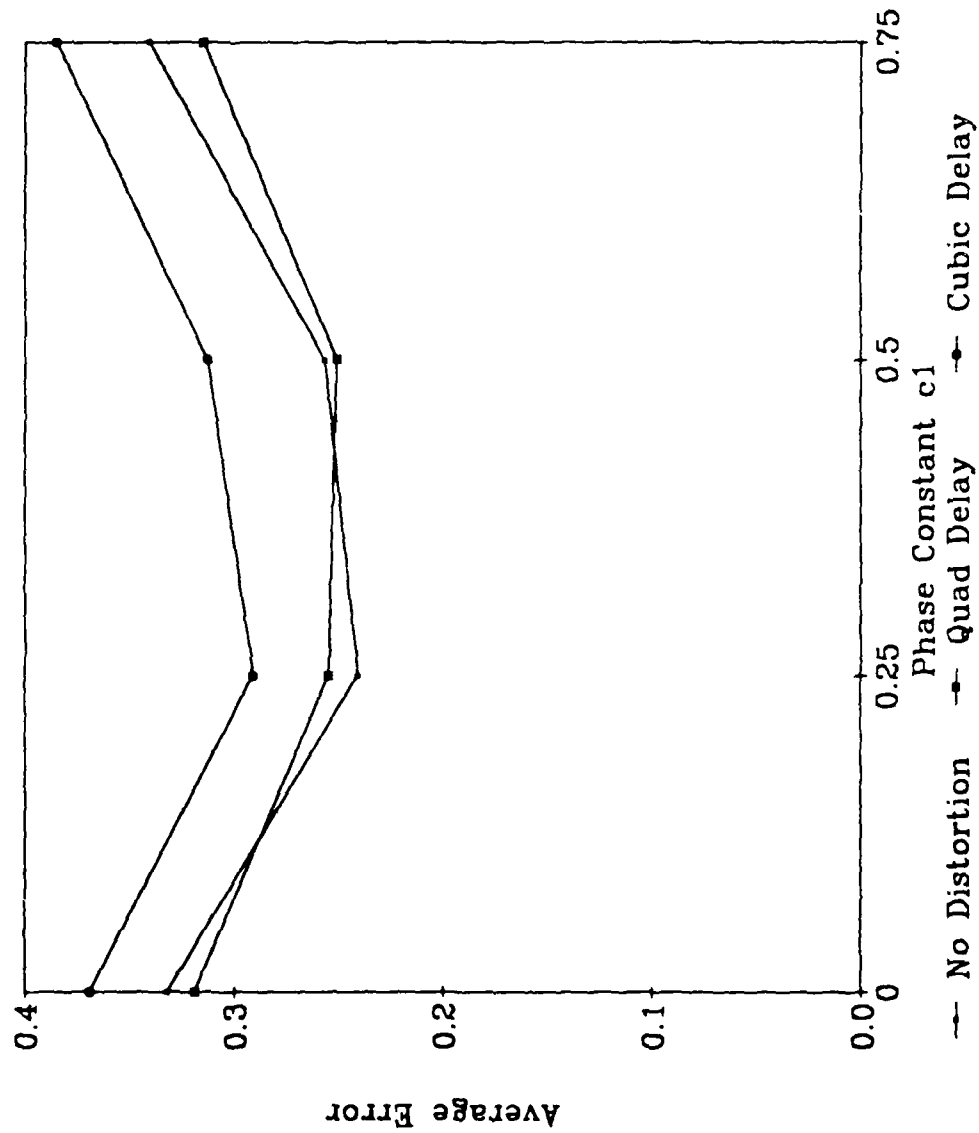


Figure 4-18 Average Error Due to Phase (-10 DB and 1 Cycle per PN Length)

PHASE ERROR AT -10 DB 2 CYCLES per PN LENGTH

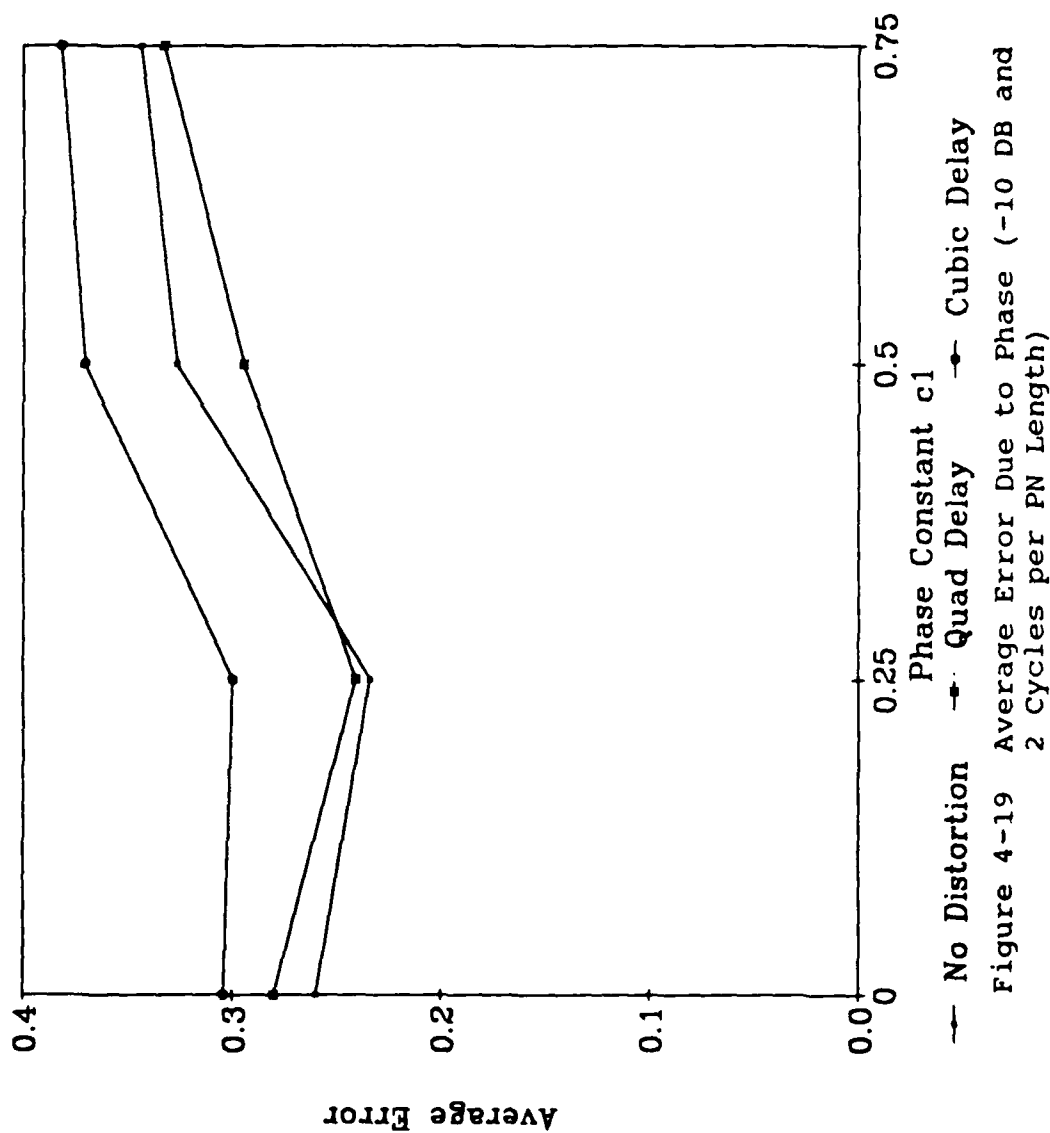


Figure 4-19 Average Error Due to Phase (-10 DB and 2 Cycles per PN Length)

PHASE ERROR AT -10 DB 8 CYCLES per PN LENGTH

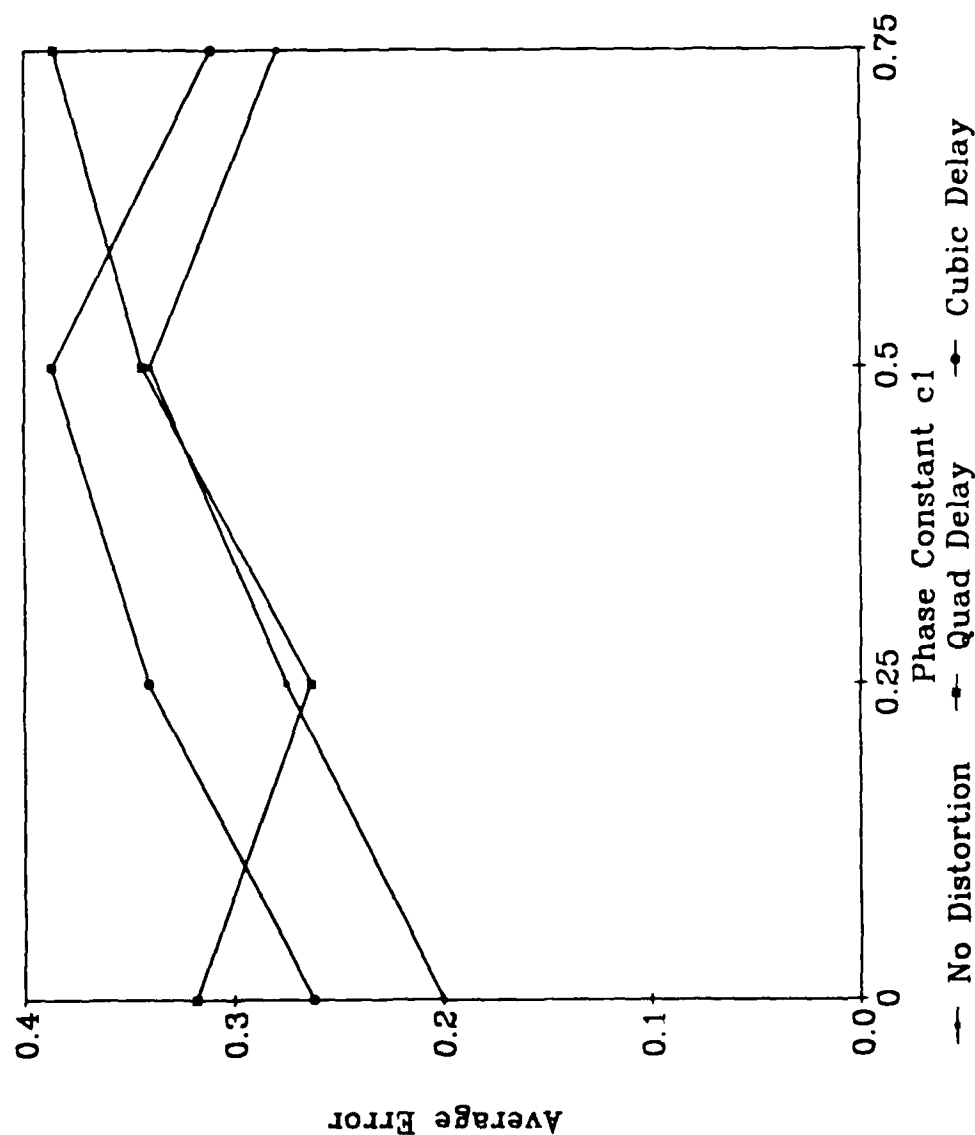


Figure 4-20 Average Error Due to Phase (-10 DB and 8 Cycles per PN Length)

TABLE 10 Average error of filter output
for various SNRs (No Distortion)

SNR (DB)	Γ
-10	1.038
0	.425
10	.138

No Distortion

SNR = 10 DB

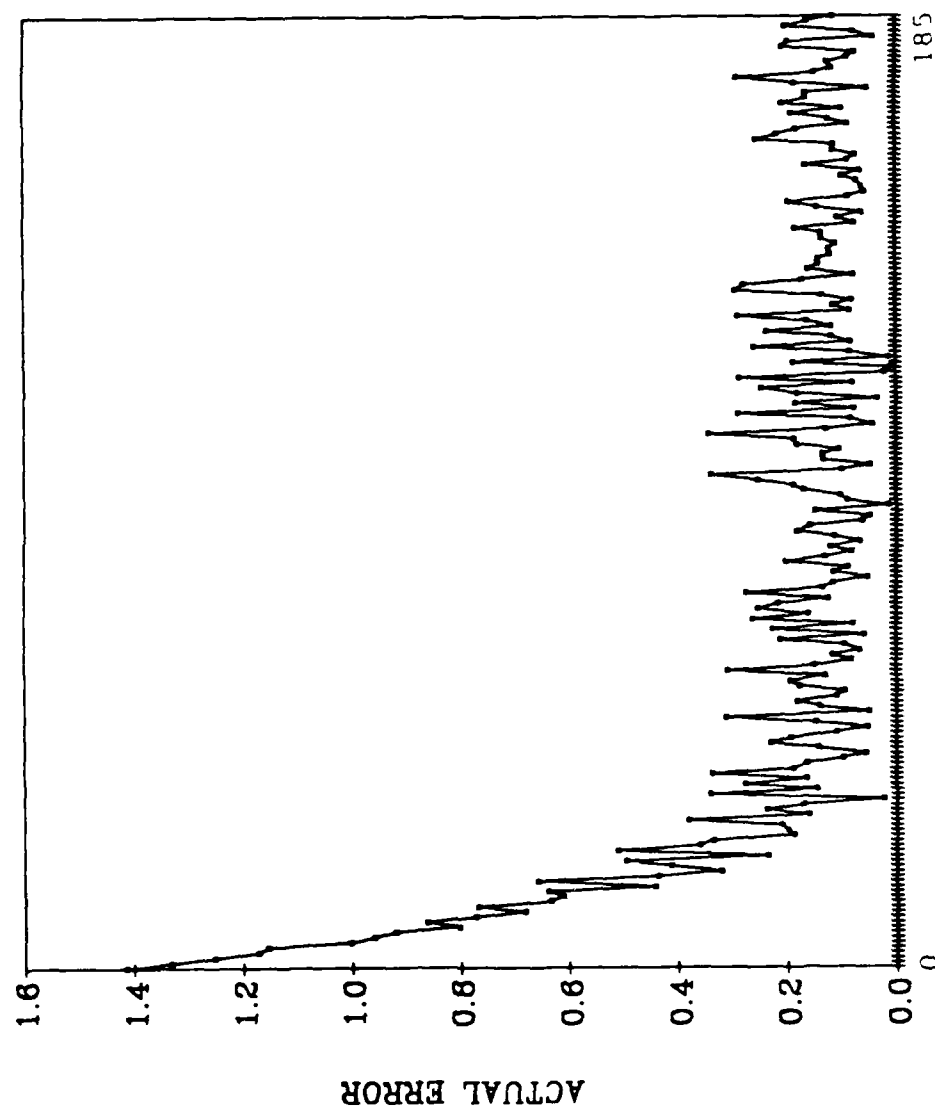


Figure 4-21 Actual Error Due to Noise (SNR = 10 DB)

No Distortion
SNR = 0 DB

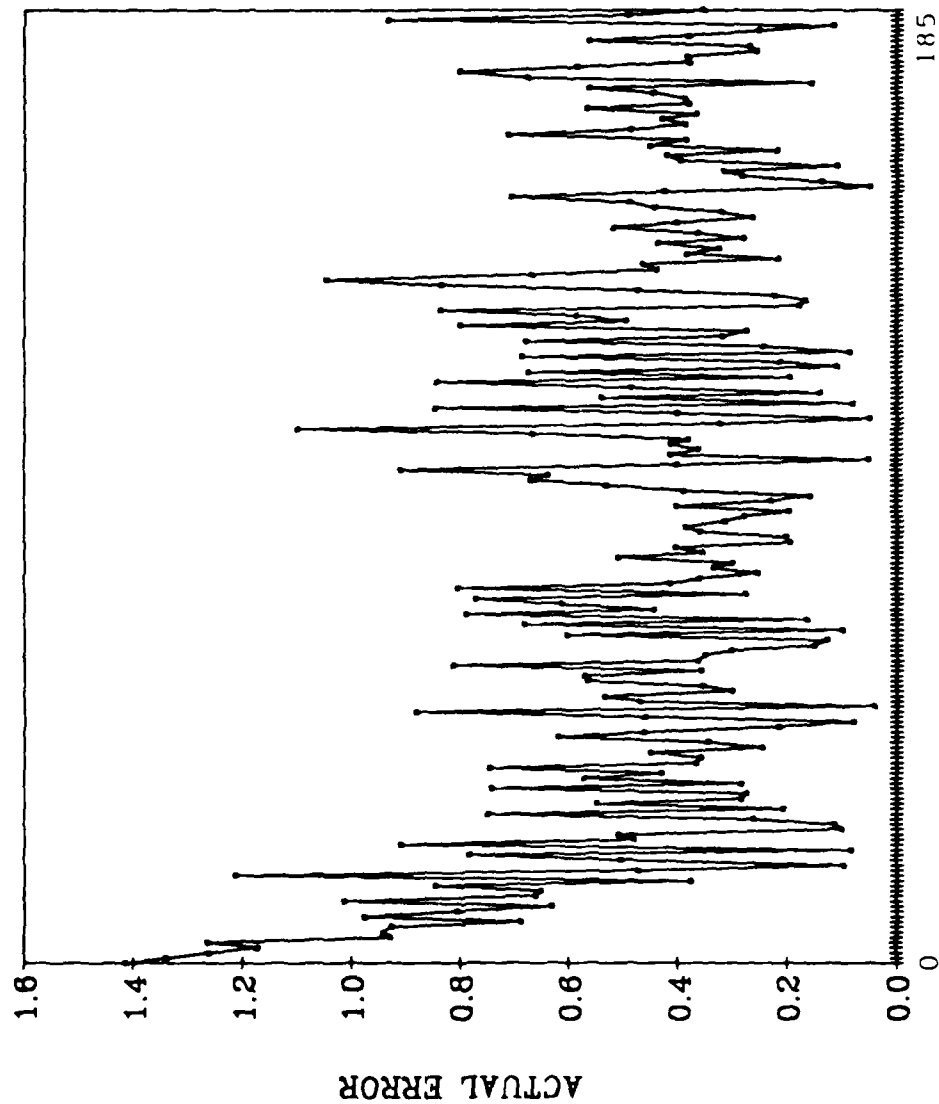
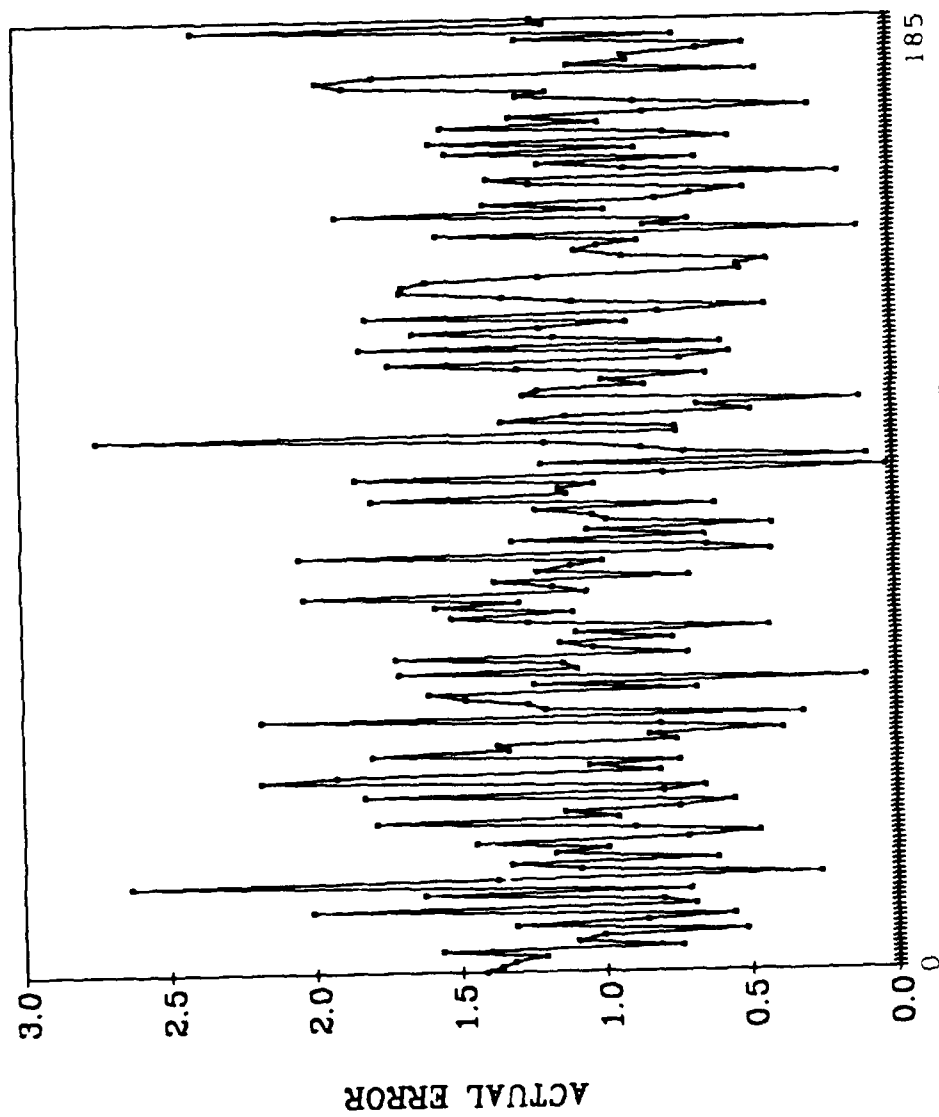


Figure 4-22 Actual Error Due to Noise (SNR = 0 DB)

No Distortion SNR = -10 DB



ADAPTATIONS
Figure 4-23 Actual Error Due to Noise (SNR = -10 DB)

LMS ALGORITHM

Linear Delay Distortion (No Jamming)

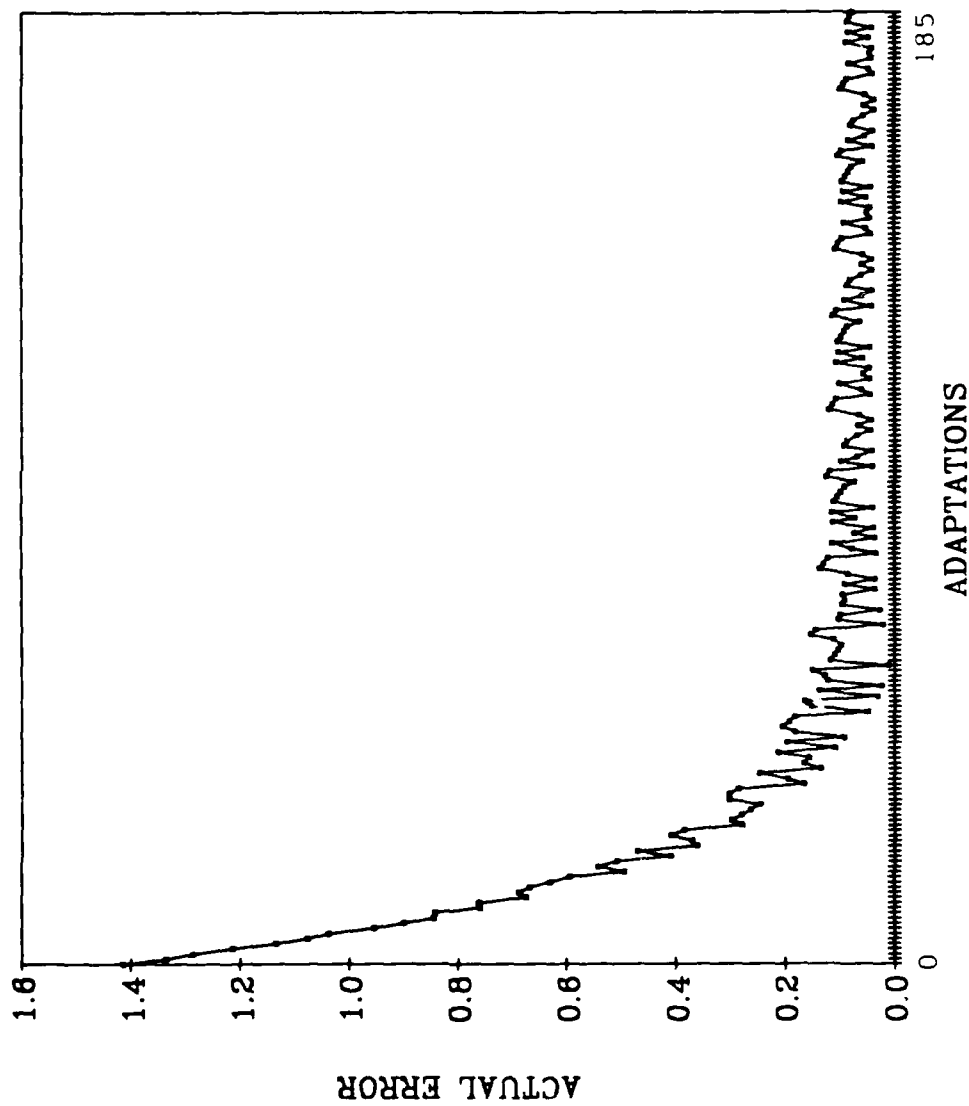


Figure 4-24 Linear Delay Distortion, No Jamming,
LMS Algorithm

LMS ALGORITHM

Cubic Delay Distortion (No Jamming)

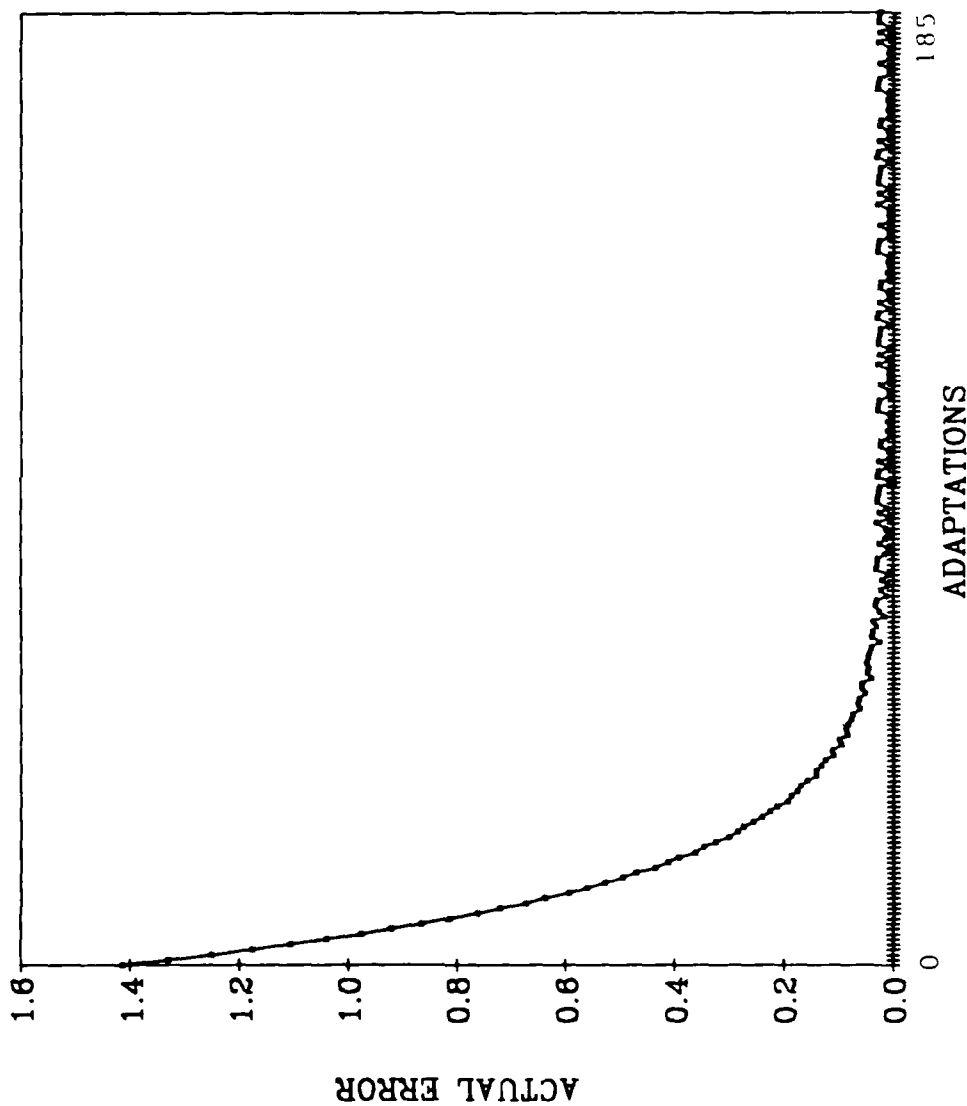
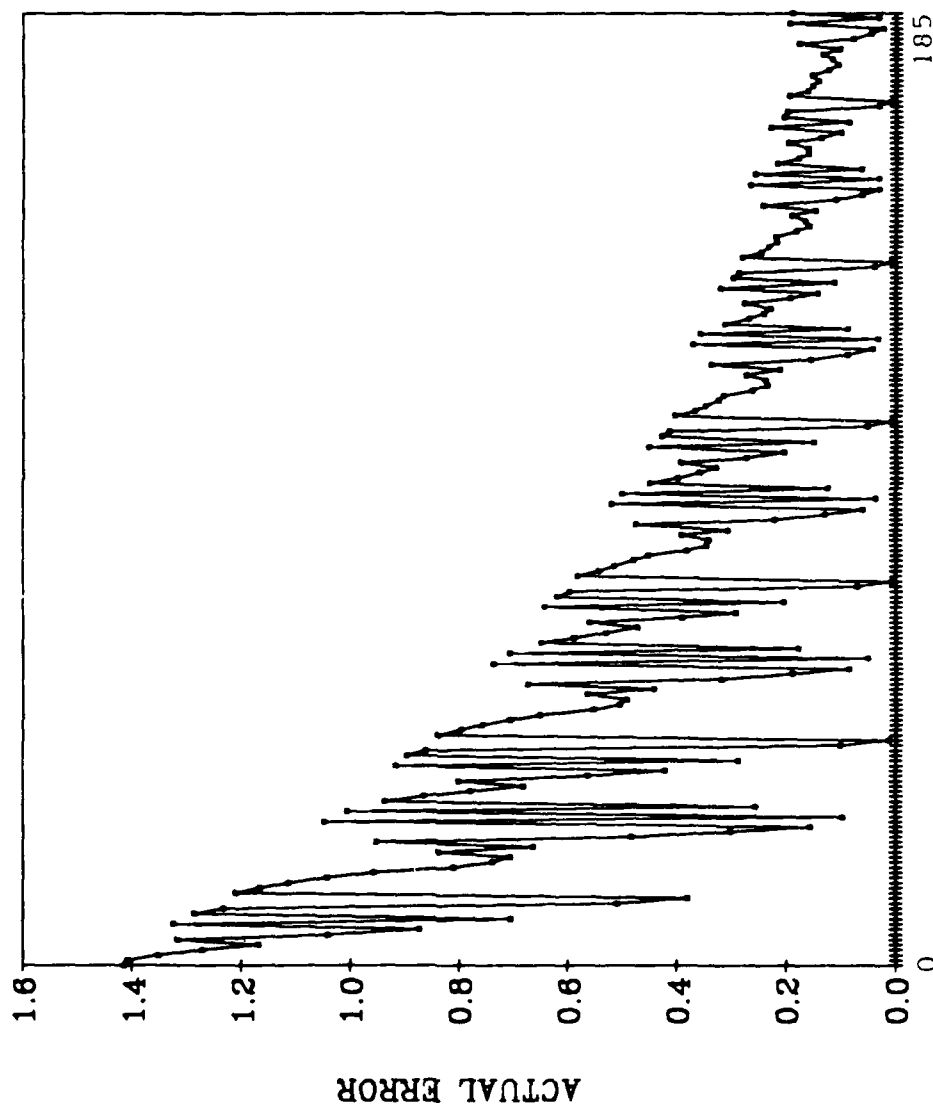


Figure 4-25 Cubic Delay Distortion, No Jamming,
LMS Algorithm

LMS ALGORITHM (No Distortion)

JSR=0 DB, 1 CYCLE per PN LENGTH



ADAPTATIONS

Figure 4-26 LMS Algorithm (No Distortion), JSR = 0 DB,
1 Cycle per PN Length

LMS ALGORITHM (No Distortion)

JSR=10 DB, 8 CYCLES per PN LENGTH

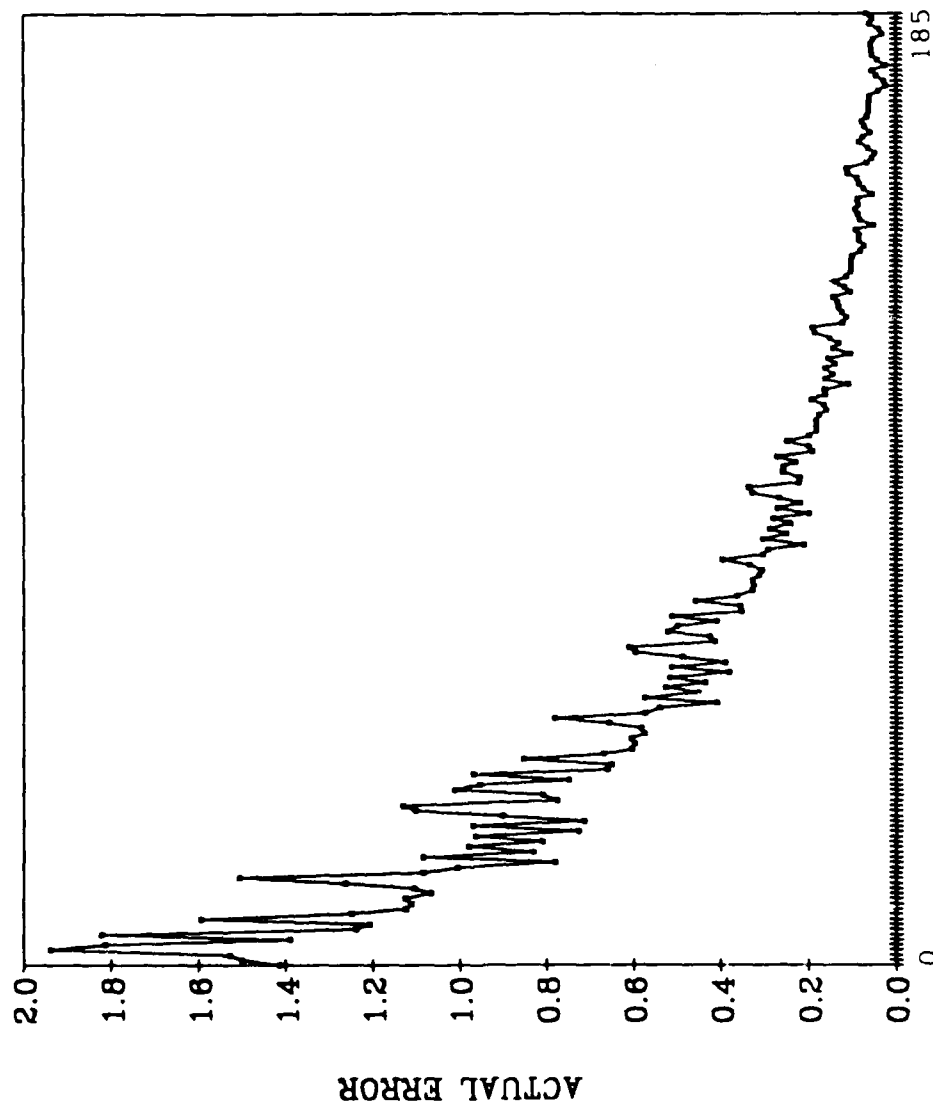


Figure 4-27 LMS Algorithm (No Distortion), JSR = 10 DB,
8 Cycles per PN Length

TABLE 11 Comparisons of Griffiths and LMS
algorithms for selected runs

LMS	Γ	GRIFFITHS	RUN
.073		DIVERGES	Linear Delay (No Noise, No Jamming)
.017		.083	Cubic Delay (No Noise, No Jamming)
.203		.371	No Distortion, JSR=0 DB 1 Cycle per PN Length
.124		.896	No Distortion, JSR=10 DB 8 Cycles per PN Length

4-4 COMPARISONS

This section will compare various results obtained in Section 4-3.

4-4-1 COMPARISONS OF NO NOISE AND NO JAMMING RESULTS

One of the most significant results is that the Griffiths algorithm diverges when pulses that experience linear delay distortion are processed by the filter. This divergence occurs even though the received signal power is well within the limits established by equation (42).

The linear delay distortion run of Figure 4-8 was executed again for (1) $\mu = .001$, adaptations = 930 and (2) $\mu = .01$, adaptations = 1085. The results were similar to Figure 4-8 again -- the algorithm was not converging.

By looking at Table 12 it is shown that for the case of linear delay distortion the power in the main lobe (i.e. that portion the matched filter uses to integrate upon) is the least compared to the other types of distortions. This means linear delay distortion causes the most spreading of the pulse. Also, linear delay distortion is the least correlated with the P-vector. The complex correlation coefficient is defined as (21:291)

$$B = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \quad (46)$$

TABLE 12 Comparisons of various distortion levels (no jamming or noise)

Distortion Level	Correlation Coefficient		% of power in main lobe
	B	B	
No Distortion	1	1	100%
Linear Delay	.773 - .345j	.847	72.8%
Quad Delay	.959	.959	98.4%
Cubic Delay	.99 - .05j	.991	98.1%

where

B = the complex correlation coefficient

X, Y = complex vectors

Cov = the covariance

σ = the standard deviation of one complex vector

Table 12 shows that linear delay distortion is the least correlated with the P-vector and has the most pulse spreading. These results show that a combination of (1) correlation with the P-vector and (2) pulse spreading have a direct impact on the convergence of the algorithm even though received power is within limits.

The LMS algorithm is able to handle linear delay distortion. This is shown in Figure 4-24. Also, the LMS algorithm produces lower average error than the Griffiths algorithm for cubic delay distortion ($\Gamma=.083$ for Griffiths and $\Gamma=.017$ for LMS).

From the results of no noise/no jamming it can be seen that the LMS adaptive matched filter is superior to the Griffiths adaptive matched filter in handling delay distortion with PN sequences. The LMS is superior because it uses a stored reference which represents the desired signal exactly.

4-4-2 COMPARISONS OF JAMMING RESULTS

4-4-2A Effects of Jamming Power

The results of Tables 1 thru 3 show that by increasing jamming power for a given frequency the average error will increase.

4-4-2B Effects of Jamming Frequency/Phase

The Griffiths filter responds well to frequency and phase changes. The average error, in general, changes with frequency and phase changes. The actual magnitude and direction of change can be found from Tables 1 thru 9. The Griffiths filter is able to converge to a steady state with frequency and phase changes in a jammer.

4-4-2C Effects of Noise Jamming

Increasing the power of the noise will result in a larger average error. Noise jamming produces a higher average error for JSRs of 10 DB and 0 DB compared to a CW jammer. At -10 DB JSR, on the average, a CW jammer produces a higher average error.

4-4-2D COMPARISONS TO LMS

The selected LMS runs are compared to the same Griffiths runs in Table 11. Based on these results the LMS

adaptive matched filter is superior to the Griffiths
adaptive matched filter for PN sequences and CW jamming.
The LMS is superior because it has a desired signal
available in the form of a stored reference.

CHAPTER 5 : CONCLUSION

5-1 SUMMARY

This thesis presented a CW and noise jamming analysis of an adaptive matched filter that (1) has a PN sequence as an input and (2) uses the Griffiths algorithm. The Griffiths filter is shown to converge over JSRs of -10 DB, 0 DB, and 10 DB and frequencies of 1, 2, and 8 cycles per PN length. Also, the filter is shown to converge over jamming phases of 0, $.5\pi$, π , and 1.5π radians.

The Griffiths filter is shown to converge for raised cosine pulses that experience quadratic delay distortion and cubic delay distortion. For pulses that experience linear delay distortion, the filter is shown to diverge even though the convergence rate constant is well within limits. This divergence is due to the lack of correlation between the input signal and the P-vector. This divergence for pulses that experience linear delay distortion is a disadvantage of using the Griffiths adaptive matched filter. Throughout the analysis the P-vector is determined apriori and held constant.

The Griffiths adaptive matched filter is shown to converge for both CW jamming and noise jamming. Noise jamming is more effective than CW jamming in the 0 DB and 10

DB JSR ranges. CW jamming, on the average, is more effective in the -10 DB JSR range.

A comparison is made between the Griffiths adaptive matched filter and an adaptive matched filter that uses the LMS algorithm. The LMS adaptive matched filter is shown to be superior to the Griffiths adaptive matched filter when a stored reference of the desired signal is available for use with the LMS.

5-2 RECOMMENDATIONS FOR FURTHER STUDY

The following are areas recommended for further study.

- 1) The P-vector was held constant during the analysis. Further research should be conducted in the area of updating the P-vector with each new processing block (i.e. determine the P-vector from received data) to see if performance can be improved.
- 2) The effects of high jamming frequencies need to be explored on the Griffiths adaptive matched filter.
- 3) The desired signal was a constant in this thesis. Further research should be conducted with a non-constant desired signal to see if performance can be improved.
- 4) It was shown that pulses that experience linear delay distortion cause the filter to diverge even though the

convergent rate constant, μ , is within limits. This phenomenon should be studied further. A more detailed correlation analysis concerning the received signal with the P-vector should be accomplished in order to explain this divergence more thoroughly.

Appendix A : Software Simulation Programs

This appendix lists the computer programs used in the simulation. All programs are MathCAD programs. User instructions for MathCAD can be found in reference [16]. This appendix contains the following programs:

- 1) JAM.MCD
- 2) PNCODE.MCD
- 3) REVEC.MCD
- 4) PVEC.MCD
- 5) GRIFFPV.MCD
- 6) CONTPV.MCD
- 7) NOISE.MCD

NAME OF PROGRAM: JAM.MCD

PURPOSE: Generates interference vectors $\underline{i}_I(k)$ and $\underline{i}_Q(k)$.

SELECTED VARIABLE IDENTIFICATION:

kmax - Number of rows in jamming matrix. Corresponds to number of samples per symbol.

IMAX - Number of columns in jamming matrix. Corresponds to the number of adaptation cycles.

VMAX - Number of elements in jamming matrix.

k, ITER, VLEN - Counting variables.

JPWR - Jamming power.

FJ - Jamming frequency.

T - Jamming period.

c1 - Jamming phase.

iI - In-phase vector.

iQ - Quadrature vector.

JAMI - In-phase matrix.

JAMQ - Quadrature matrix.

PWRI - Power in the in-phase components.

PWRQ - Power in the quadrature components.

FILES USED:

JAMI.dat - data file that stores the in-phase values.

JAMQ.dat - data file that stores the quadrature values.

```

kmax := 7          k := 0 .. kmax          Program JAM.MCD
IMAX := 30         ITER := 0 .. IMAX
VMAX := 247        VLEN := 0 .. VMAX

```

```
JPWR := .01875 · 0
```

```
FJ := 1
```

```

T :=  $\frac{1}{248}$           c1 := 0

```

```

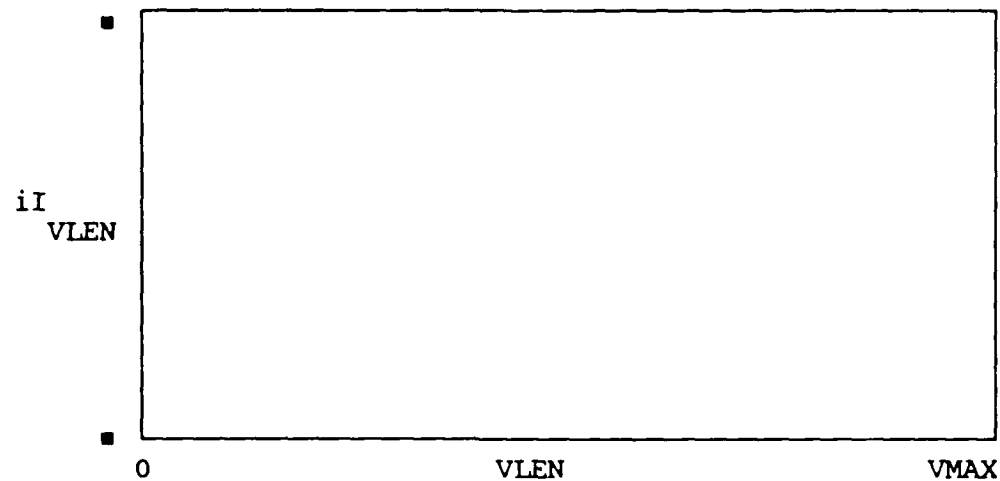
iI      :=  $\sqrt{2 \cdot \text{JPWR} \cdot \cos(2 \cdot \pi \cdot \text{FJ} \cdot \text{T} \cdot \text{VLEN} + 2 \cdot \pi \cdot \text{c1})}$ 
VLEN

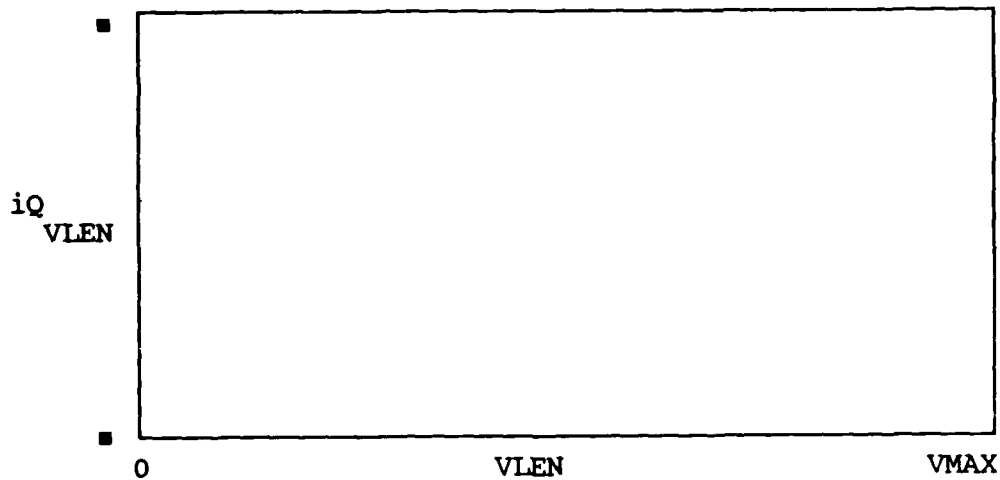
```

```

iQ      :=  $\sqrt{2 \cdot \text{JPWR} \cdot \sin(2 \cdot \pi \cdot \text{FJ} \cdot \text{T} \cdot \text{VLEN} + 2 \cdot \pi \cdot \text{c1})}$ 
VLEN

```





JAMI_{k,ITER} := iI_{k+(kmax+1) · ITER}

JAMQ_{k,ITER} := iQ_{k+(kmax+1) · ITER}

WRITEPRN[JAMI_{dat}] := JAMI

WRITEPRN[JAMQ_{dat}] := JAMQ

$$PWRI := \frac{1}{VMAX + 1} \sum_{VLEN} \left[iI_{VLEN} \right]^2$$

$$PWRQ := \frac{1}{VMAX + 1} \sum_{VLEN} \left[iQ_{VLEN} \right]^2$$

PWRI =

PWRQ =

NAME OF PROGRAM: PNCODE.MCD

PURPOSE: Generates values for $z_I(k)$ and $z_Q(k)$.

SELECTED VARIABLE IDENTIFICATION:

VMAX - FFT vector length.

BMAX, kmax - Number of rows. Corresponds to number of samples per symbol.

TMAX, IMAX - Number of columns. Corresponds to number of adaptations.

PN - Pseudonoise sequence.

BIT - Column vector which gives length and definition of the bit.

T - Column vector used in FFT calculations.

BITS - Matrix of pseudonoise bits.

PULSI, PULSQ - I and Q channels of the pulse shaping impulse response.

P1 thru P5 - FFT processing arrays.

GI, GQ - I and Q channel outputs of the pulse shaping filter.

V1MAX, V1 - Variables used in power calculations.

RI, RQ - Arrays that store I and Q values of $z(k)$.

RAI, RAQ - Arrays used in wrap around processing.

PWRI, PWRQ - Power in the I and Q components.

FILES USED:

ZII.dat - data file that stores the in-phase values.

ZQQ.dat - data file that stores the quadrature values.

```

VMAX := 255      VLEN := 0 .. VMAX      Program PNCODE.MCD
BMAX := 7        b := 0 .. BMAX
TMAX := 30       t := 0 .. TMAX
PN := (1  -1  -1  -1  -1  1  -1  -1  1  -1  1  1  -1  -1  1  1  1  1  1
      -1  -1  -1  1  1  -1  1  1  1  -1  1  -1)

```

```

      [1]
      0
      0
      0
      0
      0
      0
      0
      0]
BIT :=

      T      := 0      PULSI      := 0
      VLEN      VLEN
      PULSQ      := 0
      VLEN

```

```

      <t>      <t>
BITS      := BIT · PN
T
b+(BMAX+1) · t      := BITS      b,t
PULSI      := -.038      PULSQ      := -.058
0
PULSI      := -.109      PULSQ      := .001
1
PULSI      := -.104      PULSQ      := .133
2
PULSI      := .033      PULSQ      := .249
3
PULSI      := .26      PULSQ      := .244
4
PULSI      := .467      PULSQ      := .093
5
PULSI      := .573      PULSQ      := -.13
6
PULSI      := .587      PULSQ      := -.318
7
PULSI      := .579      PULSQ      := -.389
8

```

PULSI ₉ := .587	PULSQ ₉ := -.318
PULSI ₁₀ := .573	PULSQ ₁₀ := -.13
PULSI ₁₁ := .467	PULSQ ₁₁ := .093
PULSI ₁₂ := .26	PULSQ ₁₂ := .244
PULSI ₁₃ := .033	PULSQ ₁₃ := .249
PULSI ₁₄ := -.104	PULSQ ₁₄ := .133
PULSI ₁₅ := -.109	PULSQ ₁₅ := .001

$\text{xfft}(a) := \left[(\text{VMAX} + 1)^{.5} \right] \cdot \text{icfft}(a)$

Put FFT's in Ludeman's
terms (7:286).

$\text{ixfft}(a) := \left[\frac{1}{(\text{VMAX} + 1)^{.5}} \right] \cdot \text{cfft}(a)$

P1 := xfft(T) P2 := xfft(PULSI) P3 := xfft(PULSQ)

P4 := $\overline{(P1 \cdot P2)}$ P5 := $\overline{(P1 \cdot P3)}$

GI := Re(ixfft(P4)) GQ := Re(ixfft(P5))

kmax := 7 k := 0 ..kmax

IMAX := 30 ITER := 0 ..IMAX

VIMAX := (IMAX + 1) · (kmax + 1) - 1

V1 := 0 ..VIMAX

RI_{k,ITER} := GI_{k+(kmax+1) · ITER}

RQ_{k,ITER} := GQ_{k+(kmax+1) · ITER}

RAI_{k,0} := GI_{k+248}

RAQ_{k,0} := GQ_{k+248}

RI_{k,0} := RI_{k,0} + RAI_{k,0} Wrap-around

RQ_{k,0} := RQ_{k,0} + RAQ_{k,0}

WRITEPRN[ZII_{dat}] := RI

WRITEPRN[ZQQ_{dat}] := RQ

$$PWRI := \frac{1}{VIMAX + 1} \sum_k \sum_{ITER} [RI_{k,ITER}]^2$$

$$PWRQ := \frac{1}{VIMAX + 1} \sum_k \sum_{ITER} [RQ_{k,ITER}]^2$$

PWRQ =

PWRI =

NAME OF PROGRAM: REVEC.MCD

PURPOSE: Generates $\underline{x}(k)$.

SELECTED VARIABLE IDENTIFICATION:

RI, NI, JI - Arrays which contain the in-phase components of $\underline{x}$, $\underline{n}$, $\underline{i}$.

RQ, NQ, JQ - Arrays which contain the quadrature components of $\underline{x}$, $\underline{n}$, $\underline{i}$.

RII, NII, JII - Vectors which contain the in-phase components of $\underline{x}$, $\underline{n}$, $\underline{i}$.

RQQ, NQQ, JQQ - Vectors which contain the quadrature components of $\underline{x}$, $\underline{n}$, $\underline{i}$.

REVECI, REVECQ - Arrays which contain the I and Q components of $\underline{x}(k)$.

REVECII, REVECQQ - Vectors which contain the I and Q components of $\underline{x}(k)$.

ZZZI, ZZZQ - Bit synchronization processing vectors.

FILES USED:

REVECI.dat - Stores I channel components of $\underline{x}(k)$.

REVECQ.dat - Stores Q channel components of $\underline{x}(k)$.

Note: All other variables have been defined previously.

VMAX := 247 VLEN := 0 .. VMAX Program REVEC.MCD

kmax := 7 k := 0 .. kmax

IMAX := 30 ITER := 0 .. IMAX

RI := READPRN[$\begin{smallmatrix} \text{ZII} \\ \text{dat} \end{smallmatrix}$] RQ := READPRN[$\begin{smallmatrix} \text{ZQQ} \\ \text{dat} \end{smallmatrix}$]

NI := READPRN[$\begin{smallmatrix} \text{noiI} \\ \text{dat} \end{smallmatrix}$] NQ := READPRN[$\begin{smallmatrix} \text{noiQ} \\ \text{dat} \end{smallmatrix}$]

JI := READPRN[$\begin{smallmatrix} \text{JAMI} \\ \text{dat} \end{smallmatrix}$] JQ := READPRN[$\begin{smallmatrix} \text{JAMQ} \\ \text{dat} \end{smallmatrix}$]

REVECI $\begin{smallmatrix} \langle \text{ITER} \rangle \\ \text{:= RI} \end{smallmatrix}$ $\begin{smallmatrix} \langle \text{ITER} \rangle \\ + \text{NI} \end{smallmatrix}$ $\begin{smallmatrix} \langle \text{ITER} \rangle \\ + \text{JI} \end{smallmatrix}$

REVECQ $\begin{smallmatrix} \langle \text{ITER} \rangle \\ \text{:= RQ} \end{smallmatrix}$ $\begin{smallmatrix} \langle \text{ITER} \rangle \\ + \text{NQ} \end{smallmatrix}$ $\begin{smallmatrix} \langle \text{ITER} \rangle \\ + \text{JQ} \end{smallmatrix}$

RII $\begin{smallmatrix} \text{:= RI} \\ \text{k+(kmax+1) \cdot ITER} \end{smallmatrix}$ $\begin{smallmatrix} \text{k, ITER} \end{smallmatrix}$

NII $\begin{smallmatrix} \text{:= NI} \\ \text{k+(kmax+1) \cdot ITER} \end{smallmatrix}$ $\begin{smallmatrix} \text{k, ITER} \end{smallmatrix}$

JII $\begin{smallmatrix} \text{:= JI} \\ \text{k+(kmax+1) \cdot ITER} \end{smallmatrix}$ $\begin{smallmatrix} \text{k, ITER} \end{smallmatrix}$

RQQ $\begin{smallmatrix} \text{:= RQ} \\ \text{k+(kmax+1) \cdot ITER} \end{smallmatrix}$ $\begin{smallmatrix} \text{k, ITER} \end{smallmatrix}$

NQQ $\begin{smallmatrix} \text{:= NQ} \\ \text{k+(kmax+1) \cdot ITER} \end{smallmatrix}$ $\begin{smallmatrix} \text{k, ITER} \end{smallmatrix}$

JQQ $\begin{smallmatrix} \text{:= JQ} \\ \text{k+(kmax+1) \cdot ITER} \end{smallmatrix}$ $\begin{smallmatrix} \text{k, ITER} \end{smallmatrix}$

```
REVECII      := REVECI
      k+(kmax+1) · ITER      k, ITER
```

```
REVECQQ      := REVECQ
      k+(kmax+1) · ITER      k, ITER
```

```
-----
ZZZI      := REVECII
      VLEN      mod(VLEN+4, VMAX+1)
```

```
ZZZQ      := REVECQQ
      VLEN      mod(VLEN+4, VMAX+1)
```

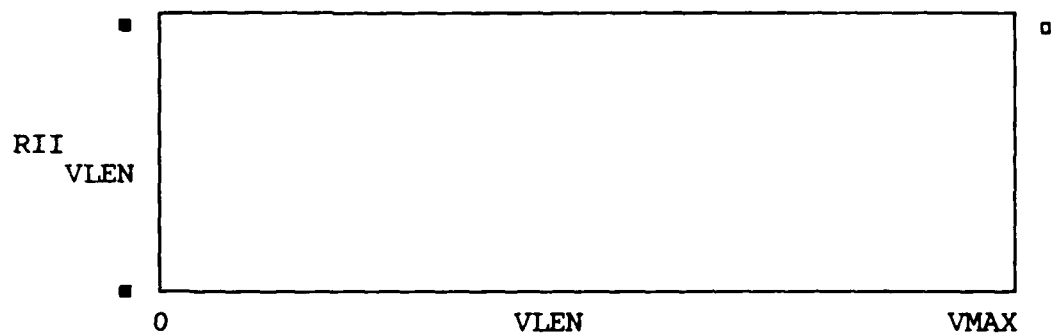
```
REVECII := ZZZI
```

```
REVECQQ := ZZZQ
```

Used for bit
synchronization

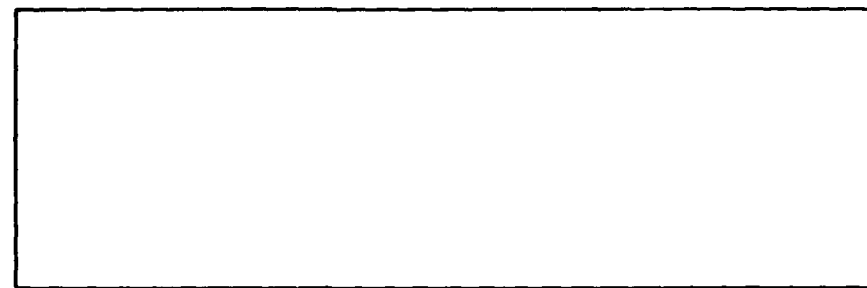
```
REVECI      := REVECII
      k, ITER      k+(kmax+1) · ITER
```

```
REVECQ      := REVECQQ
      k, ITER      k+(kmax+1) · ITER
-----
```



NII

VLEN



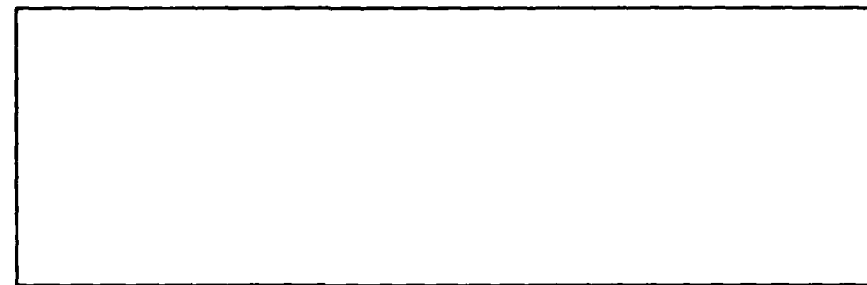
0

VLEN

VMAX

JII

VLEN



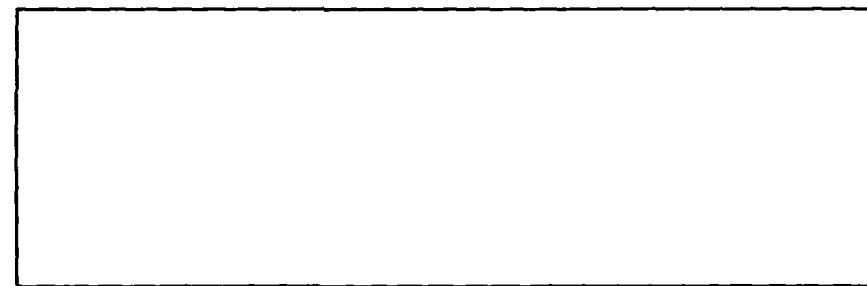
0

VLEN

VMAX

REVECII

VLEN



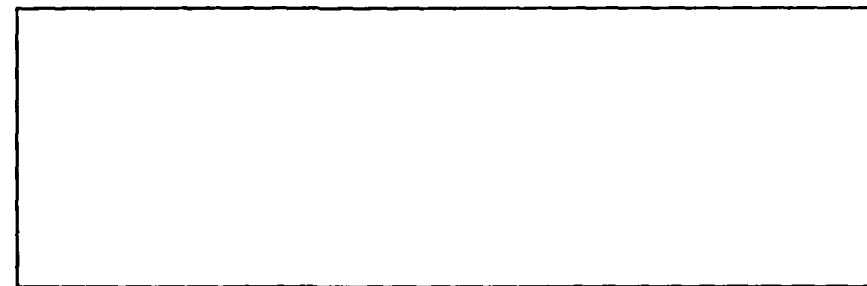
0

VLEN

VMAX

RQQ

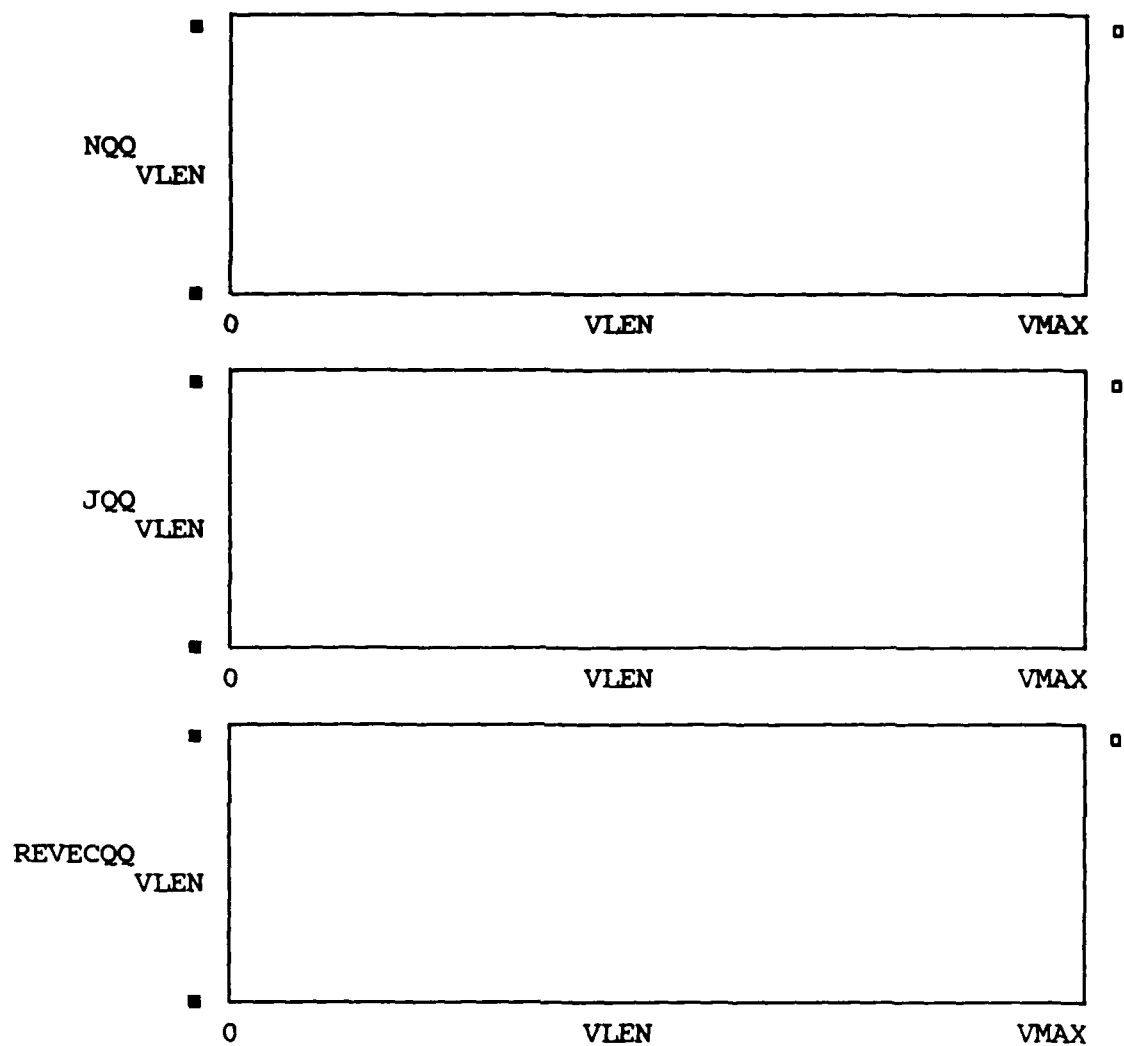
VLEN



0

VLEN

VMAX



WRITEPRN[REVECI
dat] := REVECI

WRITEPRN[REVECQ
dat] := REVECQ

$$PWRI := \frac{1}{VMAX + 1} \sum_{VLEN} \left[\frac{REVECII}{VLEN} \right]^2$$

PWRI =

$$PWRQ := \frac{1}{VMAX + 1} \sum_{VLEN} \left[\frac{REVECQQ}{VLEN} \right]^2$$

PWRQ =

NAME OF PROGRAM: PVEC.MCD

PURPOSE: Calculates the P-vector.

SELECTED VARIABLE IDENTIFICATION:

D - The desired signal.

RI - The in-phase component of $\mathbf{I}(k)$.

RQ - The quadrature component of $\mathbf{I}(k)$.

R - The received array $\mathbf{I}(k)$.

U, V - Processing arrays.

P - The P-vector.

FILES USED:

PVECI.dat - Stores the in-phase component of the
P-vector.

PVECQ.dat - Stores the quadrature component of the
P-vector.

Note: All other variables have been defined previously.

Program PVEC.MCD

First initialize and define constants.

kmax := 7 k := 0 ..kmax

IMAX := 30 ITER := 0 ..IMAX ITER1 := 1 ..IMAX + 1

D := 1 + j

RI := READPRN[REVECI_{dat}]

RQ := READPRN[REVECQ_{dat}]

R := RI + RQ·j

U := $\bar{R}$

$$V_k := \frac{1}{IMAX + 1} \sum_{ITER} U_{k,ITER}$$

Used for troubleshooting
purposes

P := D·V·31 a

$$P := \begin{bmatrix} 0 \\ .17 \\ .5 \\ .849 \\ 1 \\ .849 \\ .5 \\ .17 \end{bmatrix} \cdot D$$

WRITEPRN[PVECI_{dat}] := Re(P)

WRITEPRN[PVECQ_{dat}] := Im(P)

NAME OF PROGRAMS: GRIFFPV.MCD and CONTPV.MCD

PURPOSE: Generate adaptation cycles of the Griffiths filter. GRIFFPV.MCD performs the first "IMAX" iterations. CONTPV.MCD allows the user to perform as many adaptations as he desires in addition to those performed by GRIFFPV.MCD.

SELECTED VARIABLE IDENTIFICATION:

μ - Convergence rate constant.

W - The weight vector.

RI - In-phase component of the received vector.

RQ - Quadrature component of the received vector.

PI - In-phase component of the P-vector.

PQ - Quadrature component of the P-vector.

P - P-vector.

Y - The actual filter output.

ERR2 - the error signal.

FILES USED:

WI.dat - Stores in-phase values of the weight vector.

WQ.dat - Stores quadrature values of the weight
vector

MAGY.WKS - Stores the magnitude output of the filter.

ERR2.WKS - Stores the error output of the filter.

Note: All other variables have been defined previously.

Program GRIFFPV.MCD

First initialize and define constants.

kmax := 7 k := 0 ..kmax

IMAX := 30 ITER := 0 ..IMAX ITER1 := 1 ..IMAX + 1

μ := .01

D := 1 + j $W_k := 0$

PN := (1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 -1 -1 1 1 1 1 1
-1 -1 -1 1 1 -1 1 1 1 -1 1 -1)

RI := READPRN[REVECI_{dat}] RQ := READPRN[REVECQ_{dat}]

R := RI + RQ · j

PI := READPRN[PVECI_{dat}] PQ := READPRN[PVECQ_{dat}]

P := PI + PQ · j

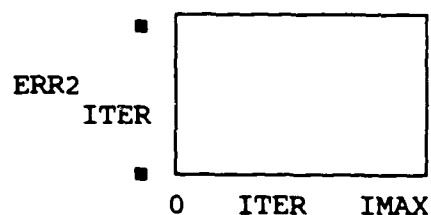
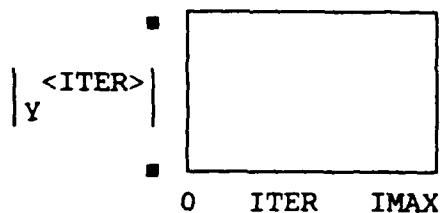
The next equation is the Griffiths algorithm

$$W^{<ITER1>} := \left[P - R \frac{\overline{<ITER1-1>}}{[R^{<ITER1-1>}]^T \cdot W^{<ITER1-1>}} \right] \cdot 2 \cdot \mu + W^{<ITER1-1>}$$

$$Y^{<ITER>} := \begin{bmatrix} <ITER> \\ R \end{bmatrix}^T \cdot W$$

$$Y^{<IMAX>} =$$

$$ERR2_{ITER} := \sqrt{\left[\text{Re}(D) - PN^{<ITER>} \cdot \text{Re}[Y^{<ITER>}] \right]^2 + \left[\text{Im}(D) - PN^{<ITER>} \cdot \text{Im}[Y^{<ITER>}] \right]^2}$$



$$WRITEPRN \begin{bmatrix} WI \\ dat \end{bmatrix} := \text{Re} \begin{bmatrix} <IMAX+1> \\ W \end{bmatrix}$$

$$WRITEPRN \begin{bmatrix} WQ \\ dat \end{bmatrix} := \text{Im} \begin{bmatrix} <IMAX+1> \\ W \end{bmatrix}$$

$$WRITEPRN \begin{bmatrix} MAGY \\ WKS \end{bmatrix} := \begin{bmatrix} <ITER> \\ Y \end{bmatrix}$$

$$WRITEPRN \begin{bmatrix} ERR2 \\ WKS \end{bmatrix} := ERR2_{ITER}$$

Program CONTPV.MCD
First initialize and define constants.

kmax := 7 k := 0 ..kmax

IMAX := 30 ITER := 0 ..IMAX ITER1 := 1 ..IMAX + 1

μ := .01

D := 1 + j $W_k := 0$

PN := (1 -1 -1 -1 -1 1 -1 -1 1 -1 1 1 -1 -1 1 1 1 1 1
-1 -1 -1 1 1 -1 1 1 1 -1 1 -1)

RI := READPRN[REVECI_{dat}] RQ := READPRN[REVECQ_{dat}]

R := RI + RQ j

PI := READPRN[PVECI_{dat}] PQr := READPRN[PVECQ_{dat}]

P := PI + PQr j

wI := READPRN[WI_{dat}]

wQ := READPRN[WQ_{dat}]

<0>
W := wI + wQ j

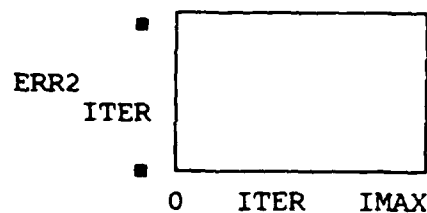
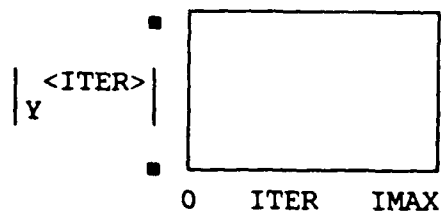
The next equation is the Griffiths algorithm

$$W^{<ITER1>} := \left[P - R \frac{\overline{<ITER1-1>}}{R} \left[<ITER1-1>^T \cdot <ITER1-1> \right] \cdot W^{<ITER1-1>} \right] \cdot 2 \cdot \mu + W^{<ITER1-1>}$$

$$Y^{<ITER>} := \left[R \right] \cdot W^{<ITER>}$$

$$Y^{<IMAX>} =$$

$$ERR2_{ITER} := \sqrt{\left[\text{Re}(D) - P_N^{<ITER>} \cdot \text{Re}[Y^{<ITER>}] \right]^2 + \left[\text{Im}(D) - P_N^{<ITER>} \cdot \text{Im}[Y^{<ITER>}] \right]^2}$$



$$WRITEPRN \left[\begin{matrix} WI \\ dat \end{matrix} \right] := \text{Re} \left[W^{<IMAX+1>} \right]$$

$$W^{<IMAX+1>} =$$

$$WRITEPRN \left[\begin{matrix} WQ \\ dat \end{matrix} \right] := \text{Im} \left[W^{<IMAX+1>} \right]$$

$$APPENDPRN \left[\begin{matrix} MAGY \\ WKS \end{matrix} \right] := |Y^{<ITER>}| \quad \square$$

$$APPENDPRN \left[\begin{matrix} ERR2 \\ WKS \end{matrix} \right] := ERR2_{ITER}$$

NAME OF PROGRAM: NOISE.MCD

PURPOSE: Generates values $\underline{n}_I(k)$ and $\underline{n}_Q(k)$.

SELECTED VARIABLE IDENTIFICATION:

NPWR - Noise power.

nI - The in-phase matrix.

nQ - The quadrature matrix.

STD - The standard deviation.

FILES USED:

noiI.dat - Stores the in-phase values.

noiQ.dat - Stores the quadrature values.

Note: All other variables have been defined previously.

This program generates the noise vectors used in the main program. Name of program "NOISE.MCD"

NPWR := 1.875 kmax := 7 k := 0 ..kmax VMAX := 247

STD := $\sqrt{\text{NPWR}}$ IMAX := 30 ITER := 0 ..IMAX VLEN := 0 ..127

m := 1 ..12

$$nI_{k, \text{ITER}} := \text{STD} \cdot \left[\sum_m (\text{rnd}(1)) - 6 \right]$$

$$nQ_{k, \text{ITER}} := \text{STD} \cdot \left[\sum_m (\text{rnd}(1)) - 6 \right]$$

WRITEPRN[noiI_{dat}] := nI

WRITEPRN[noiQ_{dat}] := nQ

$$\text{PWRI} := \frac{1}{\text{VMAX} + 1} \sum_k \sum_{\text{ITER}} \left[nI_{k, \text{ITER}} \right]^2$$

$$\text{PWRQ} := \frac{1}{\text{VMAX} + 1} \sum_k \sum_{\text{ITER}} \left[nQ_{k, \text{ITER}} \right]^2$$

PWRI =

PWRQ =

Appendix B : Definitions of Selected Terms

1. misadjustment - "misadjustment in an adaptive process is defined as the ratio of the excess mean-square error to the minimum mean-square error, and is thus a measure of how closely the adaptive process tracks the true Wiener solution, that is, a measure of the 'cost of adaptability' " (8:110). If the mean-square value of the error signal is defined as

$$\epsilon = E\{ e^2(k) \} \quad (B-1)$$

then misadjustment is defined as (10:115)

$$\text{misadjustment} = \frac{E\{ \epsilon(\infty) \} - \epsilon_{\min}}{\epsilon_{\min}} \quad (B-2)$$

2. cyclostationary - A random process $X(t)$ is cyclostationary in the wide sense if its mean and autocorrelation are periodic with some period T (13:302).

3. phase distortion - a linear system introduces phase distortion when the phase response of the system is not a linear function of frequency (18:66).

4. roll-off factor - the roll-off factor is defined to be the excess bandwidth used divided by the minimum Nyquist bandwidth. The roll-off factor takes on values from zero to one (9:368).

5. ergodic random process - Processes for which time averages and ensemble averages are interchangeable (18:266).

Bibliography

1. Air Force Manual 1-1. Basic Aerospace Doctrine of the United States Air Force, 16 March 1984.
2. Larsen, MGen Doyle E. "C3CM: Lessons Learned; Where Do We Go From Here?" Signal, 37: 37-40 (April 1983).
3. Farina, A. and G. Galati. "An Overview of Current and Advanced Processing Techniques for Surveillance Radar," IEEE 1985 International Radar Conference: 175-183 (May 1985).
4. Griffiths, L.J. "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," Proceedings of the IEEE, 57: 1696-1704 (October 1969).
5. Feuer, Arie and Ehud Weinstein. "Convergence Analysis of LMS filters with Uncorrelated Gaussian Data," IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-33: 222-229 (February 1985).
6. Griffiths, L.J. and others. "A Frequency Domain Implementation of the Block P-Vector Adaptive Algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-35: 705-707 (May 1987).
7. Ludeman, Lonnie C. Fundamentals of Digital Signal Processing. New York: Harper and Row, 1986.
8. Widrow, Bernard and Samuel D. Stearns. Adaptive Signal Processing. Englewood Cliffs, N.J.: Prentice-Hall, 1985.
9. Stremmer, Ferrel G. Introduction to Communication Systems (second edition). Reading, Massachusetts: Addison-Wesley Publishing Company, 1982.
10. Haykin, Simon. Introduction to Adaptive Filters. New York: Macmillan Publishing Company, 1984.
11. Sklar, Bernard. Digital Communications Fundamentals and Applications. Englewood Cliffs, N.J.: Prentice-Hall, 1988.
12. DiFranco, J.V. and W.L. Rubin. Radar Detection. Dedham, Massachusetts: Artech House, 1980.
13. Gardner, William A. Introduction to Random Processes. New York: Macmillan Publishing Company, 1986.

14. Gaskill, Jack D. Linear Systems, Fourier Transforms, and Optics. New York: John Wiley and Sons, 1978.
15. Widrow, Bernard and others. "The Complex LMS Algorithm," Proceedings of the IEEE, 63: 719-720 (April 1975).
16. Mathsoft Inc. MathCAD 2.0. Cambridge, MA: Mathsoft Inc., 1987 (software program and documentation manual).
17. Prescott, Glenn E. "A Numerical Evaluation of the Effects of Delay Distortion in a Noise Free Communication System," Handout distributed for EENG 799, Independent Study. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1988.
18. Ziemer, R.E. and W.H. Tranter. Principles of Communications Systems, Modulation, and Noise. Boston: Houghton Mifflin Company, 1985.
19. Widrow, Bernard and others. "Adaptive Noise Cancelling: Principles and Applications," Proceedings of the IEEE, 63: 1692-1716 (December 1975).
20. Schleher, D. Curtis. Introduction to Electronic Warfare. Dedham, Massachusetts: Artech House, 1986.
21. Urkowitz, Harry. Signal Theory and Random Processes. Norwood, Massachusetts: Artech House, 1983.

VITA

Captain Peter D. Pasko [REDACTED]
[REDACTED] [REDACTED]
[REDACTED] in 1975 [REDACTED] entered the U.S. Air Force

Academy at Colorado Springs, Colorado. In 1979 he graduated from the USAF Academy as an Honor Graduate with a Bachelor of Science in Electrical Engineering. He completed pilot training and received his wings in 1980. He then served as a C-130 pilot at Pope AFB, North Carolina and a T-37 instructor pilot at Vance AFB, Oklahoma. Captain Pasko entered the School of Engineering, Air Force Institute of Technology, in May 1987.

[REDACTED]
[REDACTED]

UNCLASSIFIED

Abstract

This thesis presents a CW and noise jamming analysis of an adaptive matched filter that (1) uses the Griffiths algorithm and (2) has a pseudonoise sequence as an input. The analysis is conducted over several jamming powers, frequencies, and phases. The Griffiths adaptive matched filter is shown to converge for raised cosine pulses that experience no distortion, quadratic delay distortion, and cubic delay distortion. The Griffiths adaptive matched filter diverges for pulses that experience linear delay distortion even though the convergence rate constant is within limits. Throughout the analysis the P-vector is determined apriori and held constant. The Griffiths filter is shown to converge for CW jamming and noise jamming. Noise jamming is shown to be more effective in the higher power ranges. A comparison is made between the Griffiths adaptive matched filter and an adaptive matched filter that uses the LMS algorithm. The degradation in performance of the Griffiths filter compared to an LMS filter that uses a stored reference is calculated for several selected runs. The actual computer programs used are presented in the Appendix.

UNCLASSIFIED

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. PORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GE/ENG/88D-35		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION School of Engineering	6b. OFFICE SYMBOL (if applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology WPAFB, OH 45433-6583		7b. ADDRESS (City, State, and ZIP Code)	
8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. PROJECT NO. TASK NO. WORK UNIT ACCESSION NO.	

11. TITLE (Include Security Classification)

PERFORMANCE OF AN ADAPTIVE MATCHED FILTER USING THE GRIFFITHS ALGORITHM

PERSONAL AUTHOR(S)

PETER D. PASKO, Captain, US Air Force

TYPE OF REPORT

MS Thesis

13b. TIME COVERED

FROM TO

14. DATE OF REPORT (Year, Month, Day)

December 1988

15. PAGE COUNT

132

SUPPLEMENTARY NOTATION

Approved for release in accordance with AFR 190-1

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Adaptive Filters, Matched Filters, Digital Communications, Jamming, Algorithms
FIELD	GROUP	SUB-GROUP	
25	02		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Advisor: Glenn E. Prescott, Major, USAF

Assistant Professor of Electrical and Computer Engineering

Approved for release in
accordance with AFR 190-1
10 Jan 89

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Major Glenn E. Prescott		22b. TELEPHONE (Include Area Code) 513-255-2024	22c. OFFICE SYMBOL AFIT/ENG