Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

NJIT New Jersey's Science & Technology University

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

AN ADAPTIVE CORRELATOR RECEIVER FOR COMBINED SUPPRESSION OF CO-CHANNEL INTERFERENCE AND NARROW-BAND JAMMERS IN A SLOWLY FADING CHANNEL

by
Raymond Carbone

This work deals with the adaptive correlation of a direct sequence spread spectrum signal in the presence of narrow-band, multipath and multiple user interference. The Least Mean Square and Recursive Least Square algorithms are employed for the adaptive convergence of the correlator receiver to minimize the mean squared error.

The performance of the adaptive correlator is compared with the matched filter correlator receiver and the conventional prediction filter for the suppression of narrow-band interference by calculating the bit error probability rate. The adaptive correlator is also compared with the RAKE receiver for multipath suppression and compared to the decorelating detector for the suppression of multiple user interference. It is shown that the adaptive correlator is capable of suppressing interference when the spread spectrum signal is corrupted by a combination of disturbances, such as narrow-band jammers and multipath or multiple users on the same channel.

AN ADAPTIVE CORRELATOR RECEIVER FOR COMBINED SUPPRESSION OF CO-CHANNEL INTERFERENCE AND NARROW-BAND JAMMERS IN A SLOWLY FADING CHANNEL

by
Raymond Carbone

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

October 1994

APPROVAL PAGE

AN ADAPTIVE CORRELATOR RECEIVER FOR COMBINED SUPPRESSION OF CO-CHANNEL INTERFERENCE AND NARROW-BAND JAMMERS IN A SLOWLY FADING CHANNEL

Raymond Carbone

Dr. Alexander M. Haimovich, Thesis Advisor / Date
Associate Professor of Electrical and Computer Engineering,
New Jersey Institute of Technology

Dr. Nirwan Ansari, Committee Member / Date
Associate Professor of Electrical and Computer Engineering,
New Jersey Institute of Technology

Dr. Zoran Siveski, Committee Member / Date
Assistant Professor of Electrical and Computer Engineering,
New Jersey Institute of Technology

BIOGRAPHICAL SKETCH

Author: Raymond Carbone

Degree: Master of Science in Electrical Engineering

Date: October 1994

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, New Jersey, 1994

- Bachelor of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, New Jersey, 1992

Major: Electrical Engineering

This work is dedicated to
Leandro, Pasqualina and Marco Carbone

ACKNOWLEDGMENT

I would like to thank Dr. Alexander Haimovich for his guidance, support and enormous patience without which this work would not have been possible. I would also like to thank Dr. Nirwan Ansari and Dr. Zoran Siveski for their much needed input and contributions.

Special thanks go to my friends and colleagues who have helped me along the way to seeing this work complete; notably, Aparna Vadhri and Andrew Bateman for their insights on the topic, and Karl Hauck and Mike Castellano for their support. I would also like to thank Lisa Fitton for her review of this work.

I would especially like to thank my family who have supported and guided me through not only the completion of this work but throughout all my studies.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Spread spectrum is a form of modulation in which a signal with narrow bandwidth is distributed over a wide range of frequencies. This signal is then transmitted through a channel to a receiver which collapses the signal to its original frequency. Hence, the essence of a spread spectrum system is to take the power of a narrow-band signal and *spread* it into a wide-band prior to transmission over a channel. The original narrow-band signal is then recovered and *despread* at the receiver. Spread spectrum modulation has been used until recently solely in military communications, where the spreading process makes the transmitted signal look like wide-band noise, thereby disguising it from unfriendly receivers [15]. In addition to *masking* the transmitted signal, the main advantage of spread spectrum is its ability to suppress interferences. New uses for spread spectrum technology are being found in commercial applications, such as in mobile communications, cellular phones, and computer local area networks. This work, however, will focus solely on the interference rejection capabilities of a spread spectrum receiver.

Spread spectrum is a digital technology in which the desired signal is transmitted as a stream of *bits*. Spread spectrum modulation has inherent immunities to various disturbances such as jammers and multipath interference as well as interferences from other users on the channel. These inherent immunities include:

1. Reducing the ratio of power to frequency, which causes a lower potential for disruption of the signal from interferences.

2. Built-in security, since traditional narrow-band receivers will only see a small part of the signal.

3. Reduction in information loss due to a narrow-band jammer, since only part of the wide-band signal is lost due to the interference.

These inherent immunities are sufficient to suppress narrow-band interferences if the power level of the interference is low compared with the power of the spread spectrum signal. However, if excessive interference exists in the channel, or the power level of the interference is much higher than the spread spectrum signal, additional signal processing will be necessary to recover the original signal.

There are a number of different techniques which can be used for spread spectrum technology, the most popular being time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA). The most common modulation techniques are Direct Sequence (DS), Frequency Hopping (FH), Chirp, and Time Hopping (TH) [1]. This work considers only DS spread spectrum modulation. DS spread spectrum is accomplished by modulating each transmitted data bit by a pseudo noise (PN) code sequence composed of a specified number of *chips*, the length of this code being equal to the duration of each transmitted data bit. This signal is then transmitted through a channel where it may be corrupted by noise and interference. At the receiver, this signal is demodulated with the same PN code sequence (assuming perfect synchronization between transmitter and receiver) to recover the original signal.

The spread spectrum receiver consists of a PN code demodulator in series with a low-pass filter. This receiver essentially acts as a matched filter with the receiver matched to the PN code sequence. Assuming that the spread spectrum signal is corrupted with additive white Gaussian noise (AWGN) and interference, the demodulator will *spread* the interference over the entire spread spectrum bandwidth while simultaneously *despreading* or collapsing the data signal back into its original bandwidth. The interference will assume the properties of AWGN and can effectively be removed by the low-pass filter, thus recovering the original signal, albeit slightly

distorted. This procedure results in an inherent processing gain in the system equal to the ratio of the duration of the data bit over the PN chip duration. By extension, this process reduces the noise power of the signal by an amount equal to the processing gain. The filtered signal is then passed through a decision circuit (i.e., threshold device) to decide what data bit symbol was originally transmitted.

This conventional receiver works well in recovering the data bit in environments with low signal-to-noise ratios (SNR) and low signal-to-interference ratios (SIR), yielding low bit error probabilities. However, when the noise power or the interference (jammers, multipath or multiple users) power is substantially greater than the data power, the performance of this receiver degrades quickly, making it essentially useless for signal recovery. In this case, additional circuitry is required to eliminate the interference.

As has been shown in previous works dealing with interference suppression in a spread spectrum signal, notably [5],[7],[8],[9] and [10], a tapped-delay line prediction filter can be employed to suppress narrow-band interference and recover the original signal. This filter notches out the interference signal prior to spread spectrum demodulation. The Wiener filter exploits the high coherency property of the narrow-band interference to estimate the interference. This estimate is then subtracted from the incoming received signal to produce a signal containing mostly AWGN noise and the spread spectrum signal. The data bit can then be recovered by demodulating this signal with the PN code sequence. The statistics of the noise, signal and interference must be known in order to design the optimum filter. Once these statistics are known, the filter yields the minimum mean squared error (MMSE) achievable. If the statistics are not known *a-priori*, an estimate of the correlation matrix may be generated by averaging squared values of the received signal. The estimate improves as more elements are taken in the averaging.

When the estimate of the correlation matrix can not be found before hand, such as in a non-stationary environment, the tap weights of the Wiener filter can be updated at each incoming signal sample using an adaptive algorithms such as the LMS and RLS, as described by Haykin [3], and Qureshi [18]. The algorithm adjusts the tap weights of the filter such that the MSE at the output is reduced. The performance of each algorithm depends, among other properties, on how fast it converges to a tap weight vector which produces an error close to the MMSE.

In addition to narrow-band interference, the spread spectrum signal may also be corrupted by multiple propagation paths caused by a fading channel. The multipath acts as additional interference to the spread spectrum signal, thus lowering the signal-to-interference ratio. Traditionally, a RAKE receiver, as the one described by Proakis [17] and by Price and Green [16], is used to suppress the multipath in a slowly fading channel when the channel response is known. The RAKE receiver eliminates multipath by combining the energies of the different paths. However, the RAKE receiver performs poorly when other disturbances are present in the channel as well, such as narrow-band jammers and multiple users.

In this work, an adaptive correlator (ADC) receiver, much like the one in [12],[13] and [14], which utilizes the LMS and RLS adaptive algorithms for weight convergence, is used to jointly suppress both the multipath and jammer interference. Bit probability error rates (BER) for the ADC utilizing both LMS and RLS algorithms are calculated for different scenarios involving multipath and narrow-band jammers. These error probabilities are then compared with the conventional matched filter receiver and the RAKE receiver, and the results are plotted. It is shown that the ADC outperforms either filter in eliminating interference, performing near optimum at low SNR's.

The ADC is also used to address the interference caused by other users on the same channel, referred to as multiple access interference (MAI). The conventional

way of demodulating multiple users is to have separate matched filters (*conventional multi-user detector*), each one matched to the user's corresponding PN signature code [22]. However, optimum decision performance is achieved only when the signature codes are orthogonal to each other. When they are not, this receiver cannot successfully recover the originally transmitted data bit with a high degree of accuracy. In addition, the conventional multi-user detector performs poorly when the power of the interfering user is substantially greater than the desired user's power level, such as when the interfering user's transmitter is closer to the receiver than the desired user's transmitter. This is called the *near-far* effect [22]. Another type of detector used to suppress interference caused in multiple users is the *decorrelating detector*, described in [2],[20],[21] and [22]. This receiver utilizes the inverse of the cross-correlation matrix of the signature codes to eliminate the multiple user interference. It will be shown that the ADC utilizing a fast converging algorithm such as the RLS produces satisfactory results in eliminating interference caused by multiple users. The BER for the ADC is calculated and compared with both the conventional multi-user detector and the decorrelating detector.

The ADC correlator will be shown to be a convolution of different FIR filters, each one performing a different task. One filter removes narrow-band interference, another filter performs channel estimation and removes the multipath interference, and yet another filter demodulates the spread spectrum signal. The ADC is an integrated, versatile structure which jointly removes narrow-band, multipath, and multiple user interference.

In Chapter 2, the spread spectrum system and the theory used in this work will be covered. The conventional matched filter receiver and the tapped-delay line receiver are introduced, and it is shown how they demodulate the spread spectrum signal in the presence of interference. The adaptive receiver is introduced as well. The LMS and RLS adaptive algorithms are also derived. Chapter 2 concludes with

theory on the RAKE receiver for multipath suppression and the decorrelating receiver used for multiple user suppression. Chapter 3 will explain the ADC and the signal model used to suppress narrow-band, multipath, and multiple user interferences. The performance of the RAKE receiver for multipath suppression and the decorrelating detector for multiple user interference suppression is analyzed and compared with the ADC by calculating the BER. In Chapter 4, the simulations are given along with discussions of the results. Chapter 5 provides the conclusions and discussion of possible future work.

CHAPTER 2

THE SPREAD SPECTRUM SYSTEM AND THEORY

This chapter deals with the spread spectrum signal model and the various filter designs used to suppress interference and recover the original transmitted data bit. The conventional matched filter spread spectrum receiver and prediction filter for jammer suppression are presented. The theory of a fading channel is also introduced and how the RAKE receiver is employed to combat the multipath interference. MAI theory is described and how the decorrelating detector used to suppress MAI. The LMS and RLS adaptive algorithms are then described in the context of an adaptive receiver.

2.1 The Spread Spectrum Signal Model

As described in the previous chapter, DS spread spectrum is a form of modulation in which a data bit stream is multiplied by a PN code sequence prior to transmission. We assume binary phase-shift keying (BPSK) data transmission with data bits $d(t)$ having values in the signal set $\{+1, -1\}$ with duration T_b. The PN code sequence $s(t)$ consists of L *chips*, each having a duration of T_c where $LT_c = T_b$. Figure 2.1 depicts a typical spread spectrum modulation transmitter, and Figure 2.2 shows a diagram illustrating the modulation scheme. As can be seen in Figure 2.2, the output of the spread spectrum transmitter will have an increase in bandwidth from $\frac{1}{T_b}$ to $\frac{1}{T_c}$. The more chips that are used in the PN code, the greater the bandwidth of the transmitted signal will be. With a large number of chips, the transmitted signal assumes a noise-like appearance.

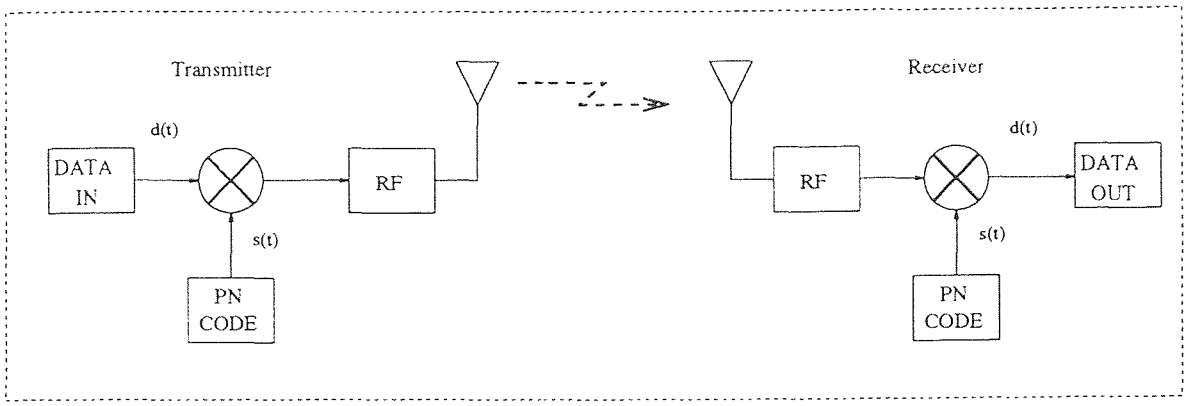


Figure 2.1 The spread spectrum transmitter and receiver.

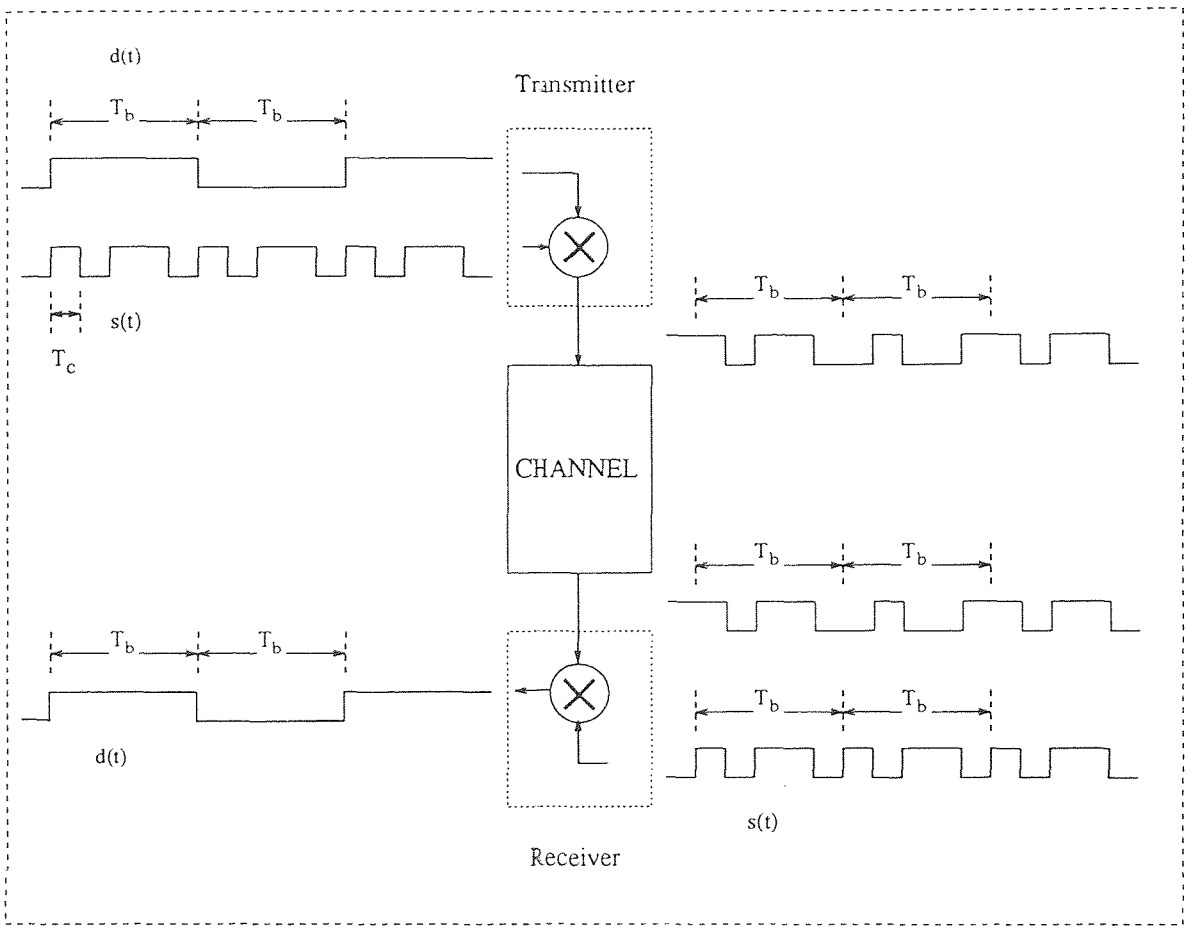


Figure 2.2 The modulation and demodulation of the data bit $d(t)$.

The original data bit is recovered by multiplying the received spread spectrum signal with the PN code sequence, and collapsing the signal back to narrow-band (see Figure 2.2).

When the channel contains interference, the spread spectrum signal model is described as:

$$y(t) = \sqrt{a}d(t)s(t) + j(t) + n(t) \tag{2.1}$$

where $y(t)$ is the received signal, $n(t)$ is AWGN, the signal $j(t)$ is the jammer, and a is the energy of the data bit, assumed to be constant for all bits for sake of simplicity. In order to simplify the analysis, the following assumptions will be made:

1. The data bits $d(t)$ have zero mean, are independent, and have a variance $\sigma_d^2 = 1$.
2. The noise samples $n(t)$ have zero mean and are uncorrelated with variance σ_n^2 equal to the power spectral density of the noise, $\frac{N_o}{2}$.
3. The jammer signal $j(t)$ is stationary, Gaussian with zero mean.

Two types of jammers will be considered in this work. The first type, a tone jammer, is be represented as:

$$j(t) = A_j \cos(\omega_j t + \theta_j) \tag{2.2}$$

where A_j is the amplitude of the jammer and ω_j is the frequency of the jammer. The phase θ_j is equal to a random variable uniformly distributed over the range $[0, 2\pi]$.

When the jammer has zero mean, the auto-correlation of $j(t)$ is defined as:

$$\begin{aligned} R_{jj}(t_1, t_2) &= E[j(t_1)j^*(t_2)] \\ &= A_j^2 E[\cos(\omega_j t_1 + \theta_j)\cos(\omega_j t_2 + \theta_j)] \\ &= A_j^2 E[\frac{1}{2}\cos(\omega_j(t_1 - t_2)) + \frac{1}{2}\cos(\omega_j(t_1 + t_2) + 2\theta_j)] \\ &= A_j^2 E[\frac{1}{2}\cos(\omega_j(t_1 - t_2))] \\ &= A_j^2 \cos(\omega_j \tau) \\ &= J\cos(\omega_j \tau) \end{aligned} \tag{2.3}$$

where J is the jammer power. The second type of jammer considered here is a narrow-band, zero mean, Gaussian jammer with an ideal bandpass spectrum and auto-correlation equal to:

$$R_{jj} = J\left(\frac{\sin(\omega_a \frac{\tau}{2})}{\omega_a \frac{\tau}{2}}\right)\cos(\omega_a \tau) \tag{2.4}$$

where ω_a is the jammer bandwidth [11].

2.2 Spread Spectrum Receivers

2.2.1 The Conventional Matched Filter Spread Spectrum Receiver

The conventional spread spectrum receiver is depicted in Figure 2.3. As can be seen in the figure, this receiver is just a correlator receiver consisting of a PN code generator followed by a summer, which is just a matched filter for a digital signal. It is assumed that the carrier frequency has been removed in a previous stage in the receiver so as to only be concerned with a baseband signal. If samples are taken every lT_c seconds, then the received signal takes the form:

$$y(lT_c) = \sqrt{a}\, d(lT_c)s(lT_c) + j(lT_c) + n(lT_c) \tag{2.5}$$

where the signals d, s, a, j and n were defined in the previous section.
The received signal in vector format for a one-bit interval is:

$$\mathbf{y}_i = \sqrt{a}\, d_i \mathbf{s} + \mathbf{j} + \mathbf{n} \tag{2.6}$$

where each vector contains L elements corresponding to a sampling interval of T_b, and the time subscript i denotes the i-th data bit.

The matched filter receiver is the optimum solution when the interference and noise are modeled as white and Gaussian. The maximum achievable SNR is attained when the filter is matched to the PN code sequence, producing an inherent processing gain. This gain is equal to L, the number of chips in the PN code sequence. The

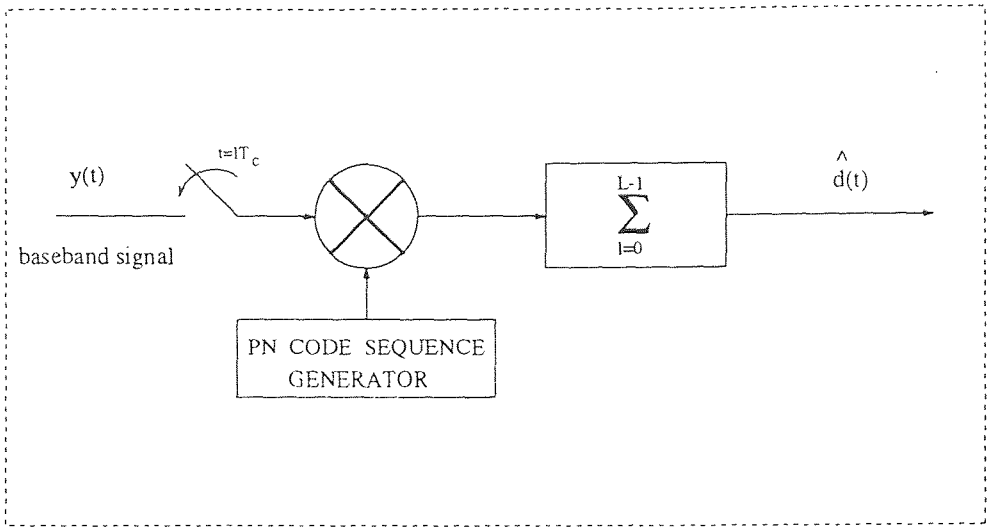


Figure 2.3 The conventional matched filter spread spectrum receiver

processing gain is due to the spreading of the narrow-band interference over a much wider frequency bandwidth when modulated by the PN code sequence in the receiver. The effective power of this interference is reduced by a factor of the processing gain when the signal is passed through the summer, which serves as a low pass filter. This processing gain is given as:

$$G = \frac{T_b}{T_c}. \tag{2.7}$$

Therefore, the SNR is increased by the amount equal to the processing gain.

The SNR at the input of the system is equal to the power of the desired signal divided by the power of the interference

$$\begin{aligned} \mathrm{SNR}_I &= \frac{(\sqrt{a}\, d_i)^2}{J + \sigma_n^2} \\ &= \frac{a}{J + \sigma_n^2}. \end{aligned} \tag{2.8}$$

The output SNR is thus equal to the input SNR multiplied by the processing gain, i.e.,

$$\begin{aligned} \mathrm{SNR}_O &= \mathrm{G}\,\mathrm{SNR}_I \\ &= \frac{\mathrm{G}\,a}{J + \sigma_n^2}. \end{aligned} \tag{2.9}$$

When the jammer signal power is large enough that the processing gain alone is insufficient to successfully recover the transmitted data bit, a prediction filter can be used to suppress the jammer.

2.2.2 The Transversal Prediction Filter for Interference Suppression

Prediction is the estimation of the future value $y(t+\tau)$ of a process $y(t)$ in terms of its past values $y(t-\tau)$, $t > 0$. If we assume the statistics of the jammer, $j(t)$, are known and are stationary, then $j(t)$ can be predicted from past values of $y(t)$. As was stated in Chapter 1, a tapped-delay line filter can be used to predict the narrow-band interference and remove it prior to spread spectrum demodulation. A circuit diagram of this filter is given in Figure 2.4. The tap weights $\{b_l\}$ of the filter are chosen to minimize the difference between the received input signal and the estimated jammer signal. The tap weight vector of the complete filter including the first tap (which is set to unity) is $[1 \ -\mathbf{b}]$, where $\mathbf{b}$ is the adaptive portion of the tap weight vector which estimates the jammer.

The signal is sampled at the spread spectrum chip rate. Each delay tap is set equal to a duration of a chip; thus, the total delay of the filter is one data bit in duration. The sampled received signal is shifted into the filter one chip at a time and then multiplied into its corresponding tap weight. The sum of these products produces an estimate of the current received signal which is then subtracted from the actual received signal sample to produce the difference or the error:

$$e_l = y_l - \sum_{n=1}^{L-1} b_n y_{l-n} \tag{2.10}$$

where l represents a chip. The tap weights $\{b_l\}$ are adjusted to minimize this error. Since the signal vector $\sqrt{a}\, d\mathbf{s}$ and noise vector $\mathbf{n}$ from equation 2.6 are uncorrelated with the received signal $\mathbf{y}$ and have zero mean, the output of the linear predictor will produce an estimate of the highly correlated jammer signal. Therefore, minimizing

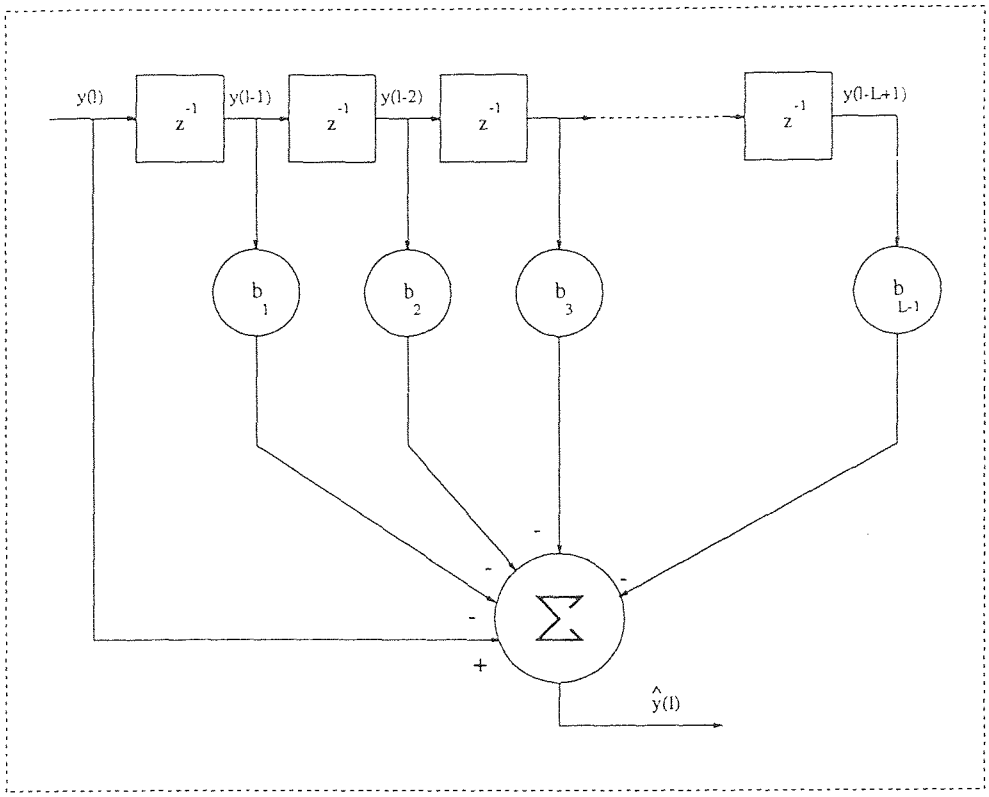


Figure 2.4 Tapped-delay line prediction filter.

the above equation is equivalent to minimizing:

$$e_l = y_l - \hat{j}_l. \tag{2.11}$$

For the i-th bit interval the error signal e_l contains the spread spectrum signal plus noise and some residual jammer energy:

$$\begin{aligned} e_l &= y_l - \hat{j}_l \\ &= \sqrt{a}\, ds_l + n_l + j_l - \hat{j}_l \\ &= \sqrt{a}\, ds_l + n_l + \gamma \end{aligned} \tag{2.12}$$

where γ represents the residual jammer component. The signal is then passed through the PN decorrelator of Figure 2.3 to collapse the data signal back to its original narrow-band state, while jointly spreading the residual jammer energy over

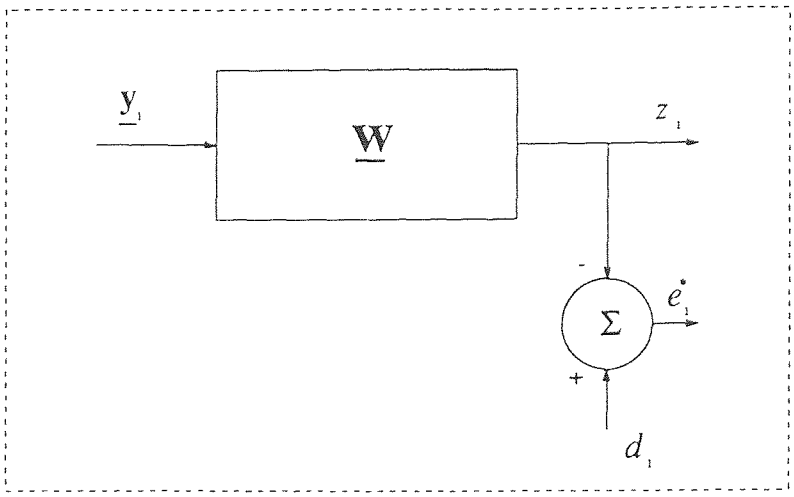


Figure 2.5 Wiener filter.

the entire spread spectrum bandwidth. The summer acts as a low pass filter having a bandwidth equal to the data bit bandwidth $\left(\frac{1}{T_b}\right)$. All wide-band interference is removed and the data bit is recovered. Advantages of this filter include easy control of bandwidth, an infinite null, and the capability of tracking the exact frequency of the interference.

2.2.3 Wiener Filter

Instead of using a filter which minimizes the error between the received signal and the estimate of the jammer signal $\hat{j}_l$, we design a filter which minimizes the error between the received signal and the desired bit d_i. Thus, an error signal will be produced at every bit interval instead of at every chip interval as in the case of a prediction filter just described. When the correlation matrix of the interference is known, the Wiener-Hopf equation can be used to calculate the optimum tap weight values which minimize the MSE between the output of the filter and the desired response d_i. The filter is depicted in Figure 2.5.

Expressing the signals in vector form, we have:

$$\mathbf{w}^T = [w_0, w_1, w_2, \ldots, w_{L-1}]^T \tag{2.13}$$

$$\mathbf{y}_i^T = [y_0, y_1, y_2, \ldots, y_{L-1}]^T \tag{2.14}$$

where the superscript T denotes the transpose of the matrix. The desired response signal estimate is the vector dot-product of the vectors $\mathbf{y}_i$ and $\mathbf{w}$ given as:

$$z_i = \mathbf{w}^H \mathbf{y}_i, \tag{2.15}$$

in which the bit estimate is calculated as:

$$\hat{d}_i = \mathrm{sgn}(z_i). \tag{2.16}$$

The error signal is

$$\begin{aligned} e_i^* &= d_i - z_i \\ &= d_i - \mathbf{w}^H \mathbf{y}_i \end{aligned} \tag{2.17}$$

where the superscript H denotes the *Hermitian Transpose* and the superscript, $*$, represents complex conjugation. The MSE is therefore,

$$\begin{aligned} \mathcal{J}(\mathbf{w}) &= E[|e^*|^2] \\ &= E[(d_i - z_i)(d_i - z_i^*)]. \end{aligned} \tag{2.18}$$

If we assume that $\mathbf{y}_i$ and the data bit d_i are jointly stationary, the following interpretations can be made:

1. The variance of the desired response is

$$\begin{aligned} \sigma_d^2 &= E[d_i d_i^*] \\ &= 1. \end{aligned} \tag{2.19}$$

2. The expectation $E[\mathbf{y}_i d_i^*]$ is an Lx1 cross-correlation vector:

$$\mathbf{P}_{dy} = [p_0, p_1, p_2, \ldots, p_{L-1}]^T. \tag{2.20}$$

3. The Hermitian of $\mathbf{P}$ is

$$\mathbf{P}_{dy}^{H} = E[d_i \mathbf{y}^{H}]. \tag{2.21}$$

4. The expectation of the received signal is an LxL auto-correlation matrix:

$$\mathbf{R}_{yy} = E[\mathbf{y}_i \mathbf{y}_i^{H}]. \tag{2.22}$$

With these assumptions, the MSE from equation 2.18 becomes:

$$\mathcal{J}(\mathbf{w}) = 1 - \mathbf{P}_{dy}^{H}\mathbf{w} - \mathbf{w}^{H}\mathbf{P}_{dy} + \mathbf{w}^{H}\mathbf{R}_{yy}\mathbf{w}. \tag{2.23}$$

The point at which $\mathcal{J}(\mathbf{w})$ achieves the minimum value is called $\mathcal{J}_{\min}$ and the corresponding $\mathbf{w}$ is called $\mathbf{w}_{\text{opt}}$. The resulting filter achieves optimum mean-square error.

The weight vector $\mathbf{w}_{\text{opt}}$ is found by using the well known Wiener-Hopf equation [3], given as:

$$\mathbf{R}_{yy}\mathbf{w}_{\text{opt}} = \mathbf{P}_{dy}. \tag{2.24}$$

Multiplying both sides of equation 2.24 by $\mathbf{R}_{yy}^{-1}$ gives the optimum weight vector as:

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{yy}^{-1}\mathbf{P}_{dy}. \tag{2.25}$$

Therefore,

$$\mathcal{J}_{\min} = 1 - \mathbf{P}_{dy}^{H}\mathbf{w}_{\text{opt}} - \mathbf{w}_{\text{opt}}^{H}\mathbf{P}_{dy} + \mathbf{w}_{\text{opt}}^{H}\mathbf{R}_{yy}\mathbf{w}_{\text{opt}}. \tag{2.26}$$

The major difference between the prediction filter and the Wiener filter when both are used for interference suppression of a spread spectrum signal is that the prediction filter tries to predict the value of the interference and then remove it from the received signal prior to PN deconvolution. On the other hand, the Wiener filter method tries to estimate the desired transmitted bit d_i and in the process of doing so, adjust the tap weights so as to perform both interference removal and PN demodulation all at once. Hence, there is no need for a separate PN decorrelator.

2.3 Adaptive Transversal Filter

An adaptive transversal filter automatically adjusts the tap-weights of the filter in accordance to an algorithm to continually keep the weight vector at a near optimum value. This is used in a changing or non-stationary environment. The error signal e_i is fed back and is used in the adaptive control algorithm to adjust the tap weights of the filter. A new tap weight vector $\mathbf{w}_{i+1}$ is computed with each incoming signal vector $\mathbf{y}_i$. In this section, the LMS and RLS adaptive algorithms will be presented.

2.3.1 The Least Mean Square Adaptive Algorithm (LMS)

The LMS is an adaptive algorithm which may be employed to adaptively adjust the tap weight vector $\mathbf{w}_i$ to minimize the MSE between d_i and $\hat{d}_i$ by developing an estimate of the auto-correlation matrix $\mathbf{R}_{yy}$ and the cross-correlation vector $\mathbf{P}_{dy}$. The LMS is advantageous because it does not require knowledge of the correlation functions, nor does it need to perform matrix inversion, making it simple to implement. The tap weight vector of the filter is updated according to the following equation [3]:

$$\begin{aligned} \hat{\mathbf{w}}_{i+1} &= \hat{\mathbf{w}}_i + \mu \mathbf{y}_i [d_i^* - \mathbf{y}_i^H \hat{\mathbf{w}}_i] \\ &= \hat{\mathbf{w}}_i + \mu \mathbf{y}_i e_i^* \end{aligned} \tag{2.27}$$

where

$$e_i^* = d_i - \hat{\mathbf{w}}_i^H \mathbf{y}_i \tag{2.28}$$

and the symbol $\wedge$ represents the variable as an estimate.

The LMS will recursively update/steer the weight vector $\mathbf{w}_i$ toward the MSE $\mathcal{J}_{\min}$. The speed of this convergence is dependent upon the step-size parameter, μ. The LMS will be stable and converge to a steady-state value if the constant μ satisfies the condition:

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{2.29}$$

where $\lambda_{\max}$ is the largest eigenvalue of the correlation matrix $\mathbf{R}_{yy}$. This ensures that the error, $E[\hat{\mathbf{w}}_i - \mathbf{w}_{\text{opt}}]$ approaches zero.

The LMS adaptive algorithm is given in the following steps:

1. Choose the step-size parameter μ to satisfy

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{2.30}$$

   where $\lambda_{\max}$ is the total power of the signal.

2. Initialize the weight vector to $\hat{\mathbf{w}}_0 = 0$.

3. Collect the received vector $\mathbf{y}_0$ when given d_0.

4. Compute the error estimate:

$$e(0) = d_0 - \hat{\mathbf{w}}_0 \mathbf{y}_0. \tag{2.31}$$

5. Update the weight vector:

$$\hat{\mathbf{w}}_1 = \hat{\mathbf{w}}_0 + \mu \mathbf{y}_0 e_0^*. \tag{2.32}$$

6. Repeat steps 3-5.

The estimates of $\mathbf{w}_i$ will converge toward $\mathbf{w}_{\text{opt}}$ to yield $\mathcal{J}_{\min}$.

Convergence Analysis of the LMS Adaptive Algorithm

Because the LMS estimates the correlation functions to calculate an estimate of $\mathbf{w}_i$, it will converge to a steady-state value close to but not exactly equal to $\mathcal{J}_{\min}$. This excess MSE is the difference between the $\mathcal{J}_{\min}$, given by the Wiener-Hopf equation and the actual MSE given by the LMS algorithm [3]:

$$\mathcal{J}_{\infty}^{(ex)} = \mathcal{J}_i - \mathcal{J}_{\min} \tag{2.33}$$

$$\mathcal{J}_{\infty}^{(ex)} = \mathcal{J}_{\infty} - \mathcal{J}_{\min}. \tag{2.34}$$

The *misadjustment* is defined as the dimensionless ratio $\mathcal{M}$ of the expected value of the average excess MSE to $\mathcal{J}_{\min}$ [3]:

$$\begin{aligned} \mathcal{M} &= \frac{E[\mathcal{J}_{\infty}^{(ex)}]}{\mathcal{J}_{\min}} \\ &= \frac{\mu \sum_{i=1}^{L} \lambda_i}{2 - \mu \sum_{i=1}^{L} \lambda_i}. \end{aligned} \tag{2.35}$$

The misadjustment is used to provide a measure of the cost of the adaptive process by showing how much the excess error is greater than $\mathcal{J}_{\min}$.

2.3.2 The Recursive Least Square Adaptive Algorithm (RLS)

The RLS is another adaptive algorithm which automatically adjusts the filter tap weight vector to converge to $\mathbf{w}_{\text{opt}}$. Like the LMS, the RLS utilizes previous samples of the vector $\mathbf{y}_i$ to update the weight vector estimate $\hat{\mathbf{w}}_{i+1}$. An important advantage of the RLS algorithm is that it utilizes all the information contained in the input data, extending back to the the instant of time the algorithm was started. This results in a faster rate of convergence as compared with the LMS algorithm. Although its performance proves superior to the that of the LMS, it has computational complexity significantly greater than the LMS.

Tap Weight Vector Update

The tap weight vector $\hat{\mathbf{w}}_i$ is calculated in manner similar to the LMS, given as:

$$\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + \mathbf{K}_i \alpha_i^* \tag{2.36}$$

where

$$\mathbf{K}_i = \frac{\lambda^{-1} \mathbf{q}_{i-1} \mathbf{y}_i}{1 + \lambda^{-1} \mathbf{y}_i^H \mathbf{q}_{i-1} \mathbf{y}_i} \tag{2.37}$$

is defined as the *gain vector* which is analogous to the step-size parameter μ in the LMS algorithm.

The vector $\mathbf{q}_i$ is the inverse of the auto-correlation matrix of $\mathbf{y}_i$ and is given as:

$$\mathbf{q}_i = \lambda^{-1}\mathbf{q}_{i-1} - \lambda^{-1}\mathbf{K}_i\mathbf{y}_i^H\mathbf{q}_{i-1} \tag{2.38}$$

where λ is a variable less than but close to 1. The greater the value of λ, the more memory the system has, i.e., more samples of the received signal from the beginning are taken into account when calculating $\mathbf{w}_{\text{opt}}$. The variable α_i is the estimate error and is calculated as:

$$\alpha_i = d_i - \hat{\mathbf{w}}_{i-1}\mathbf{y}_i. \tag{2.39}$$

This term is analogous to the error term e_i in the LMS algorithm.

In short, the RLS estimates a new inverse correlation matrix $\mathbf{q}_i$ given previous data. The RLS adaptive algorithm is calculated in the following manner [3]:

1. Initialize the algorithm: $\mathbf{q}_0 = \delta^{-1}\mathbf{I}$, where δ is a small positive constant and $\hat{\mathbf{w}}_0 = 0$. The constant δ should be small compared with $0.01\sigma_y^2$, where σ_y^2 is the variance of the data $\mathbf{y}_i$.

2. Collect a new vector $\mathbf{y}$ when given d_1 and compute:

$$\mathbf{K}_1 = \frac{\lambda^{-1}\mathbf{q}_0\mathbf{y}_1}{1 + \lambda^{-1}\mathbf{y}_1^H\mathbf{q}_0\mathbf{y}_1}. \tag{2.40}$$

3. Calculate the output error:

$$\alpha_1 = d_1 - \hat{\mathbf{w}}_0\mathbf{y}_1. \tag{2.41}$$

4. Update the weight vector:

$$\hat{\mathbf{w}}_1 = \hat{\mathbf{w}}_0 + \mathbf{K}_1\alpha_i^*. \tag{2.42}$$

5. Update the estimate for the auto-correlation matrix:

$$\mathbf{q}_1 = \lambda^{-1}\mathbf{q}_0 - \lambda^{-1}\mathbf{K}_1\mathbf{y}_1^H\mathbf{q}_0. \tag{2.43}$$

6. Repeat steps 2-5.

2.4 Multipath Interference in a Slowly Fading Channel

Multipath is a condition in communication channels in which the transmitted signal is propagated to the receiver along many paths [17]. Different echoes of the transmitted signal arrive at the receiver, with each echo having a randomly varying phase and amplitude. The received signal is a vector sum of the individually delayed signals. There are large variations of the strength of the received signal at a single frequency as a function of time. This phenomenon is termed *fading*. In order to suppress the fading in the channel, the echo signals may be individually detected using a correlation method and then added algebraically. The receiver is supplied with several replicas of the same information signal over independently fading channels. Thus, the probability that all signal components will fade simultaneously will be reduced [17]. This section introduces the RAKE receiver used for multipath suppression and then derives the signal model for multipath used in this work.

2.4.1 RAKE Receiver

The optimum multipath receiver for a wide-band signal is the RAKE receiver, [17] and [19]. The RAKE is a tapped-delay line filter through which the received signal is passed. A diagram of the RAKE receiver is depicted in Figure 2.6. The tap weights of the filter are given as $\{v_l\}$, where $l = 0, 1, 2, 3, \ldots, L-1$. The filter collects the signal energy from all the received signal paths that fall within the span of the delay line and carry the same information.

A channel is said to be *frequency selective* if $\Delta f_c \ll W$, where W is the bandwidth of the bandpass signal and Δf_c is defined as the *coherence bandwidth* of the channel. Signals that are scattered by more than Δf_c are affected differently by the channel. The coherence bandwidth Δf_c is defined as:

$$\Delta f_c = \frac{1}{T_m} \tag{2.44}$$

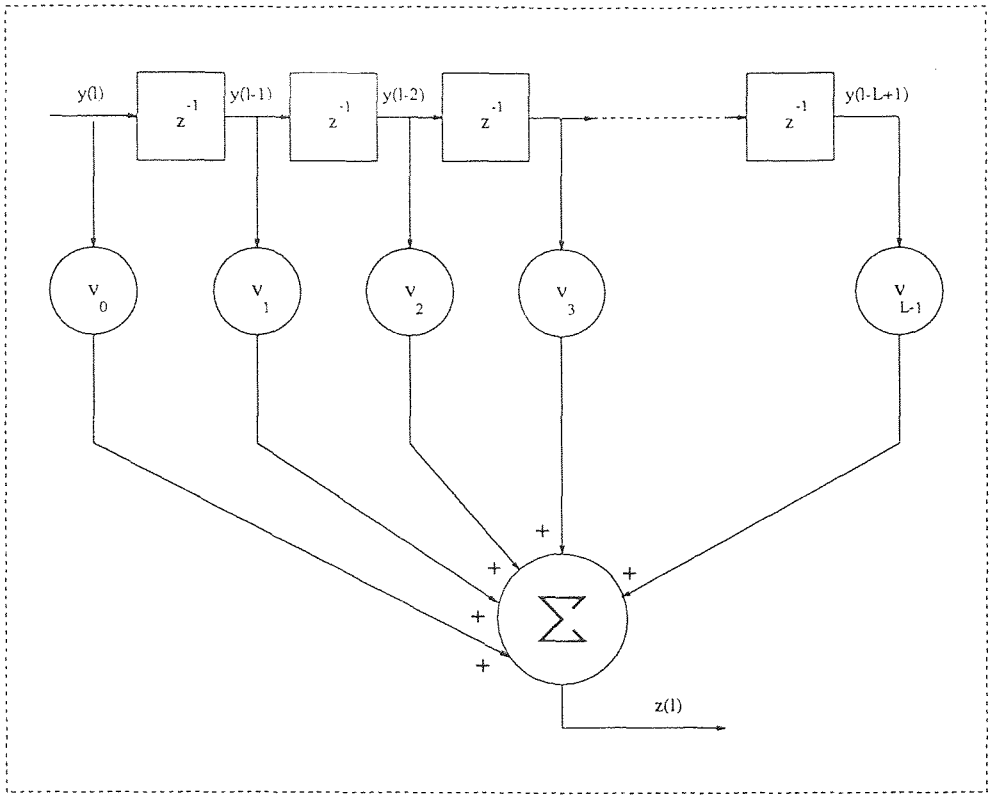


Figure 2.6 The RAKE receiver.

where T_m is the *multipath spread* of the channel, defined as the range of values of the time delay over which the average power output as a result of multipath is non-zero. We define our low-pass transmitted signal to be:

$$u(t) = \sqrt{a}\, d(t)s(t) \tag{2.45}$$

where $d(t)$ is the data bit, $s(t)$ is the PN code sequence and a is the energy of the transmitted signal. The bandwidth of $u(t)$ is W and the equivalent band occupancy is $|f| \leq \frac{W}{2}$. Sampling the signal $u(t)$ results in:

$$u(t) = \sum_{n=-\infty}^{\infty} u\left(\frac{n}{W}\right) \text{sinc}\left[w\left(t - \frac{n}{W}\right)\right]. \tag{2.46}$$

The Fourier Transform of $u(t)$ is:

$$U(f) = \begin{cases} \frac{1}{W} \sum_{n=-\infty}^{\infty} u\left(\frac{n}{W}\right) e^{\frac{-j2\pi fn}{W}} & |f| \leq \frac{W}{2} \\ 0 & |f| > \frac{W}{2} \end{cases}. \tag{2.47}$$

The received signal $y(t)$ from the frequency selective channel is expressed as:

$$y(t) = \int_{-\infty}^{\infty} H(f;t)U(f)e^{j2\pi ft}df \tag{2.48}$$

where $H(f;t)$ is the time-variant transfer function of the channel. Substituting for $U(f)$ in the above equation results in:

$$\begin{aligned} y(t) &= \frac{1}{W}\sum_{n=-\infty}^{\infty} u\left(\frac{n}{W}\right)\int_{-\infty}^{\infty} H(f;t)e^{j2\pi f\left(t-\frac{n}{W}\right)}df \\ &= \frac{1}{W}\sum_{n=-\infty}^{\infty} u\left(\frac{n}{W}\right)h\left(t-\frac{n}{W};t\right). \end{aligned} \tag{2.49}$$

Since equation 2.49 is a convolutional sum, it may be expressed as:

$$\begin{aligned} y(t) &= \frac{1}{W}\sum_{n=-\infty}^{\infty} u\left(t-\frac{n}{W}\right)h\left(\frac{n}{W};t\right) \\ &= \frac{1}{W}\sum_{n=-\infty}^{\infty} u\left(t-\frac{n}{W}\right)h_n(t) \end{aligned} \tag{2.50}$$

where $h_n(t) = \frac{1}{W}h(\frac{n}{W};t)$. Equation 2.50 states that the signal $y(t)$ is the output of a tapped-delay line filter. The time-variant frequency selective channel $h_n(t)$ can be modeled as a tapped-delay line filter with tap spacings of $\frac{1}{W}$ and tap weights $\{h_n(t)\}$. The impulse response of the channel is:

$$h(\tau;t) = \sum_{n=-\infty}^{\infty} h_n(t)\delta(\tau-\frac{n}{W}), \tag{2.51}$$

with the time variant transfer function given as:

$$H(f;t) = \sum_{n=-\infty}^{\infty} h_n(t)e^{\frac{-j2\pi fn}{W}}. \tag{2.52}$$

The following assumptions are made concerning multipath and interference characteristics in this work:

- The channel *Doppler spread* is small compared with the baseband bandwidth W (slowly fading). The multipath parameters are constant over several bit intervals.

- The interference is locally wide-sense stationary (WSS) during the estimation.
- Decision-directed mode can be applied for channel and interference estimation since the Doppler spread is small.

The tap spacing for the channel is $T_s = \frac{1}{2B}$ where B is the bandwidth. For a DS BPSK spread spectrum signal with chip interval T_c, the bandwidth is approximately equal to $\frac{1}{T_c}$. Therefore, the signal is band-limited to $|f| \leq B$, and the tap spacing becomes $T_s = \frac{T_c}{2}$. The number of taps, $\mathcal{N}$, in the channel model is:

$$\begin{aligned} \mathcal{N} &= \frac{T_m}{\frac{1}{W}} + 1 \\ &= T_m W + 1 \\ &= L_h \end{aligned} \tag{2.53}$$

where T_m is the multipath spread and $\frac{1}{W}$ is the tap spacing. Therefore, equation 2.50 for the received signal $y(t)$ becomes:

$$y(t) = \sum_{n=0}^{L_h-1} h_n(t) u\left(t - \frac{n}{W}\right). \tag{2.54}$$

If Rayleigh fading is assumed, the channel tap spacing $\{h_n(t)\}$ are zero-mean complex-valued stationary Gaussian random processes; or equivalently, the magnitudes $|h_n(t)|$ are Rayleigh distributed and the phases are uniformly distributed.

If the scattering is uncorrelated, the tap weights $\{h_n(t)\}$ are mutually uncorrelated and since they are Gaussian, they are statistically independent. This model is called *Wide-Sense Stationary Channel with Uncorrelated Scattering* (WSSUS). The impulse response of the channel is thus given as:

$$h(\tau; t) = \sum_{n=0}^{L_h-1} h_n(t)\delta(\tau - \frac{n}{W}). \tag{2.55}$$

The low-pass spread spectrum signal may be represented as:

$$\begin{aligned} u(t) &= \sum_i \sqrt{a}\, d_i s(t - iT_b) \\ &= \sum_i \sqrt{a}\, d_i \sum_{l=0}^{L_f-1} s_l p(t - lT_c) \end{aligned} \tag{2.56}$$

where L_f is the number of chips, $T_b = L_f T_c$, d_i is the data bit in the i-th bit interval and $s(t - T_m)$ is the PN code sequence. The signal $p(t - lT_c)$ is:

$$p(t - lT_c) = \begin{cases} 1 & 0 \leq t \leq T_c \\ 0 & \text{otherwise} \end{cases} . \tag{2.57}$$

The frequency selective channel can be written as:

$$h(\tau; t) = \sum_{l=0}^{L_h - 1} h_l \delta \left(\tau - \frac{lT_c}{2} \right) . \tag{2.58}$$

Since the tap weights of the tapped-delay line representation of the channel are spaced $\frac{T_c}{2}$ apart, the samples of $u(t)$ are also taken at a rate of $\frac{T_c}{2}$ in order to keep sampling rates equal. Therefore, our spread spectrum receiver will now sample twice per spreading chip, as opposed to only once per chip. If we drop the subscript i and observe only a one bit interval, the new spread spectrum signal becomes:

$$u(l\frac{T_c}{2}) = \sum_{n=0}^{2L-1} \sqrt{a}\, ds \left(l\frac{T_c}{2} - n\frac{T_c}{f} \right) . \tag{2.59}$$

Expressing the signal and channel impulse response in vector form for one data bit gives:

$$\mathbf{u} = \sqrt{a}\, d[s_0, s_0, s_1, s_1, s_2, s_2, \ldots, s_{L-1}, s_{L-1}]^T \tag{2.60}$$

$$\mathbf{h} = [h_0, h_1, h_2, \ldots, h_{L_h - 1}]^T \tag{2.61}$$

where s_l represents a PN code chip.

Since the frequency-selective channel is represented as an FIR filter, the output of the channel is the convolution of the vectors $\mathbf{u}$ and $\mathbf{h}$, i.e.,

$$\mathbf{v} = \mathbf{u} \otimes \mathbf{h}. \tag{2.62}$$

The length of $\mathbf{u}$ is $2L$ and the length of $\mathbf{h}$ is L_h. Therefore, after convolution of the two signals, the length of $\mathbf{v}$ is $N = 2L + L_h$ elements. The vector $\mathbf{v}$ is the tap weight vector of the RAKE receiver (see Figure 2.6).

The signal received by the RAKE is:

$$\begin{aligned} \mathbf{y} &= \mathbf{u} \otimes \mathbf{h} + \mathbf{n} \\ &= \mathbf{v} + \mathbf{n} \end{aligned} \tag{2.63}$$

where the vector $\mathbf{n}$ is AWGN with zero-mean.

The output z of the RAKE receiver is the vector dot-product of the vectors $\mathbf{v}$ and $\mathbf{y}$:

$$z = \mathbf{v}^H \mathbf{y}. \tag{2.64}$$

This output is passed through a threshold device to yield an estimate of the data bit d:

$$\hat{d} = \mathrm{Re}[\mathrm{sgn}(z)]. \tag{2.65}$$

The BER is simply the probability of producing a wrong decision at the threshold device, given as:

$$\begin{aligned} P_e &= \mathrm{Prob}[\mathrm{Re}(z) < 0 | d = 1] \;=\; \mathrm{Prob}[\mathrm{Re}(z) > 0 | d = -1] \\ &= \mathrm{Prob}[\mathbf{v}^H \mathbf{y} < 0 | d = 1] \\ &= \mathrm{Prob}[\sqrt{a}\, d \mathbf{v}^H \mathbf{v} + \mathbf{v}^H \mathbf{n} < 0 | d = 1] \\ &= \mathrm{Prob}[\sqrt{a}\, \mathbf{v}^H \mathbf{v} + \mathbf{v}^H \mathbf{n} < 0] \\ &= Q\left(\frac{\sqrt{a}\, \mathbf{v}^H \mathbf{v}}{\sqrt{\sigma_n^2 \mathbf{v}^H \mathbf{v}}} \right) \end{aligned} \tag{2.66}$$

where the variance of $\mathbf{v}^H \mathbf{n}$ is $\sigma_n^2 \mathbf{v}^\mathbf{H} \mathbf{v}$ with σ_n^2 the variance of $\mathbf{n}$, and Q is the Q - function defined as:

$$Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_\alpha^\infty \exp^{-\frac{t^2}{2}}\, dt. \tag{2.67}$$

When a jammer signal is also present in the received signal, the probability of error equation is similar to the above equation with the exception of the auto-correlation matrix of the jammer included in the denominator:

$$P_e = Q\left(\frac{\sqrt{a}\,\mathbf{v}^H\mathbf{v}}{\sqrt{\mathbf{v}^T\mathbf{R}_{jj}\mathbf{v} + \sigma_n^2\mathbf{v}^T\mathbf{v}}}\right) \tag{2.68}$$

where $\mathbf{R}_{jj}$ is the correlation matrix of the jammer.

2.5 Multiple-Access Interference (MAI) Suppression

In a CDMA system, several users transmit information simultaneously over a channel, with each user having a separate code waveform. Having knowledge of the codes, the receiver must demodulate the desired user's signal from the sum of all the multiple users on the channel. Each separate receiver has the code required to demodulate its desired user's signal.

We consider a BPSK DS spread spectrum system with K simultaneous users, each one having been modulated with a separate PN code sequence. If the PN code sequences are mutually orthogonal, we can successfully demodulate the received signal into separate signals by simply using a bank of matched filter spread spectrum receivers like the one described in Section 2.2.1. Each matched filter would be matched to a separate user's PN code sequence. This type of receiver is called a *conventional multi-user detector.* However, if it is not possible to obtain mutually orthogonal code sequences, the performance of this receiver deteriorates drastically. Furthermore, the conventional multi-user detector is not capable of demodulating the desired signal in the presence of a narrow-band jammer or multipath. This will also be shown below and in Chapter 4.

Another receiver used to demodulate a CDMA signal is the *decorrelating detector.* The primary performance feature of this receiver is its invariance to the signal energies of the interfering users. Although this receiver successfully excises the interference from simultaneous users, it does not perform well in the presence of

multipath or a narrow-band jammer. This section will introduce the conventional multi-user detector and decorrelating detector used to suppress MAI.

2.5.1 Multi-User Detection with a Matched Filter

As described above, a CDMA signal can be demodulated using a conventional multi-user detector if the signature codes are orthogonal. The conventional multi-user detector circuit consists of a bank of matched filters, each matched to a specific user's signature code, followed by a threshold device. This model is equivalent to a tapped-delay line filter with the tap weights equal to the PN code sequence of each particular user. The equation for the received signal for a one bit interval and K users is:

$$y(t) = \sum_{k=1}^{K} \sum_{i} \sqrt{a_k}\, d_k(i) s_k(t - iT) + n(t) \tag{2.69}$$

where $d_k(i) \in \{-1, +1\}$ is the k-th user's data bit in the i-th bit interval, a_k is the k-th user's energy and s_k is the k-th user's PN code sequence with a duration T_b. Each signal s_k contains L chips with duration T_c where $LT_c = T_b$. To simplify the analysis, we will limit ourselves to the observation of one data bit interval and drop the subscript i. Sampling at the chip rate for one bit interval produces the following signal in vector form:

$$\mathbf{y} = \sum_{k=1}^{K} \sqrt{a_k}\, d_k \mathbf{s}_k + \mathbf{n} \tag{2.70}$$

where the subscript k represents a separate user.

The output from the bank of matched filters is sampled at every i-th bit interval; thus, the output from each matched filter, x_k, is equivalent to the dot product of the two vectors $\mathbf{s}_k$ and $\mathbf{y}$ as in the Wiener filter of Figure 2.5. The output of the matched filter for the n-th user is the vector dot-product of $\mathbf{s}_n$ and $\mathbf{y}$:

$$\begin{aligned} x_n &= \mathbf{s}_n^H \mathbf{y} \\ &= \mathbf{s}_n^H \left(\sum_{k=1}^{K} \sqrt{a_k}\, d_k \mathbf{s}_k + \mathbf{n} \right) \end{aligned}$$

$$= \mathbf{s}_n^H \sum_{k=1}^{K} \sqrt{a_k}\, d_k \mathbf{s}_k + \mathbf{s}_n^H \mathbf{n}$$
$$= \sum_{k=1}^{K} \sqrt{a_k}\, d_k \mathbf{s}_n^H \mathbf{s}_k + \mathbf{s}_n^H \mathbf{n}. \tag{2.71}$$

The output x_n is fed into a threshold device to produce an estimate of the transmitted bit $\hat{d}_k$. The probability of bit error is the probability of detecting an error from the n-th user's signal. Assuming equal probability for $d_n = 1$ and $d_n = -1$ for all n, we therefore have,

$$\begin{aligned} P_e &= \text{Prob}[x_n < 0 | d_n = 1] = \text{Prob}_n[x_n > 0 | d_n = -1] \\ &= \text{Prob}[\sum_{k=1}^{K} \sqrt{a_k}\, d_k \mathbf{s}_n^H \mathbf{s}_k + \mathbf{s}_n^H \mathbf{n} < 0 | d_n = 1] \\ &= \text{Prob}[\sum_{k=1 \;\; k \neq n}^{K} \sqrt{a_k}\, \mathbf{s}_n^H \mathbf{s}_k + \sqrt{a_n}\, \mathbf{s}_n^H \mathbf{s}_n + \mathbf{s}_n^H \mathbf{n} < 0]. \end{aligned} \tag{2.72}$$

Using Bayes' Theorem results in the following equation for the probability of error for the conventional multi-user detector with only 2 users:

$$\begin{aligned} P_e &= \frac{1}{2}\text{Prob}[\sqrt{a_1}\, \mathbf{s}_1^H \mathbf{s}_1 + \sqrt{a_2}\, \mathbf{s}_1^H \mathbf{s}_2 + \mathbf{s}_1^H \mathbf{n} < 0] \\ &\quad + \frac{1}{2}\text{Prob}[\sqrt{a_1}\, \mathbf{s}_1^H \mathbf{s}_1 - \sqrt{a_2}\, \mathbf{s}_1^H \mathbf{s}_2 + \mathbf{s}_1^H \mathbf{n} < 0] \\ &= \frac{1}{2} Q\left[\frac{\sqrt{a_1}\, \mathbf{s}_1^H \mathbf{s}_1 + \sqrt{a_2}\, \mathbf{s}_1^H \mathbf{s}_2}{\sqrt{\sigma_n^2 \mathbf{s}_1^H \mathbf{s}_1}} \right] \\ &\quad + \frac{1}{2} Q\left[\frac{\sqrt{a_1}\, \mathbf{s}_1^H \mathbf{s}_1 - \sqrt{a_2}\, \mathbf{s}_1^H \mathbf{s}_2}{\sqrt{\sigma_n^2 \mathbf{s}_1^H \mathbf{s}_1}} \right] \end{aligned} \tag{2.73}$$

where the subscript 1 denotes the first or desired user and the subscript 2 denotes the second or interfering user. The term $\mathbf{s}_1^T \mathbf{s}_2$ represents the cross-correlation between the code sequences. If the codes are orthogonal, this term will equal to zero. Therefore, we note that non-orthogonal codes would introduce additional interference in the signal x_n which will affect the bit estimate $\hat{d}_n$ from the threshold device.

2.5.2 Decorrelating Detector

A decorrelating detector circuit for MAI suppression is shown in Figure 2.7. Expressing the signals $\{x_k\}$ in a matrix format, we can represent the output from the bank of matched filters as:

$$\begin{aligned}
\mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix} \\
&= \begin{bmatrix} \sqrt{a_1}\, d_1 \mathbf{s}_1^H \mathbf{s}_1 + \sqrt{a_2}\, d_2 \mathbf{s}_1^H \mathbf{s}_2 + \ldots + \sqrt{a_K}\, d_K \mathbf{s}_1^H \mathbf{s}_K + \mathbf{n}_1 \\ \sqrt{a_1}\, d_1 \mathbf{s}_2^H \mathbf{s}_1 + \sqrt{a_2}\, d_2 \mathbf{s}_2^H \mathbf{s}_2 + \ldots + \sqrt{a_K}\, d_K \mathbf{s}_2^H \mathbf{s}_K + \mathbf{n}_2 \\ \vdots \\ \sqrt{a_1}\, d_1 \mathbf{s}_K^H \mathbf{s}_1 + \sqrt{a_2}\, d_2 \mathbf{s}_K^H \mathbf{s}_2 + \ldots + \sqrt{a_K}\, d_K \mathbf{s}_K^H \mathbf{s}_K + \mathbf{n}_K \end{bmatrix} \\
&= \mathbf{S}\mathbf{d} + \mathbf{m}
\end{aligned} \tag{2.74}$$

where the matrix $\mathbf{S}$ is the cross-correlation matrix of the PN signature codes defined as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1^H \mathbf{s}_1 & \mathbf{s}_1^H \mathbf{s}_2 & \mathbf{s}_1^H \mathbf{s}_3 & \ldots & \mathbf{s}_1^H \mathbf{s}_K \\ \mathbf{s}_2^H \mathbf{s}_1 & \mathbf{s}_1^H \mathbf{s}_2 & \mathbf{s}_2^H \mathbf{s}_3 & \ldots & \mathbf{s}_2^H \mathbf{s}_K \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{s}_K^H \mathbf{s}_1 & \mathbf{s}_K^H \mathbf{s}_2 & \mathbf{s}_K^H \mathbf{s}_3 & \ldots & \mathbf{s}_K^H \mathbf{s}_K \end{bmatrix} \tag{2.75}$$

and the vector $\mathbf{d}$ is the vector containing the transmitted data bit defined as:

$$\mathbf{d} = [d_1, d_2, d_3 \ldots, d_K]^H. \tag{2.76}$$

The the vector $\mathbf{m}$ is the Gaussian vector with variance $\sigma_m^2 = \sigma_n^2 \mathbf{S}$ and is defined as:

$$\begin{aligned}
\mathbf{m} &= [\mathbf{s}_1^H \mathbf{n}_1, \mathbf{s}_2^H \mathbf{n}_2, \mathbf{s}_3^H \mathbf{n}_3 \ldots, \mathbf{s}_K^H \mathbf{n}_K]^H \\
&= [m_1, m_2, m_3, \ldots, m_K]^H.
\end{aligned} \tag{2.77}$$

If the signal set is linearly independent (that is if the matrix $\mathbf{S}$ is invertible), we can multiply the output of the bank of matched filters $\mathbf{x}$ by the inverse of the cross-correlation matrix $\mathbf{S}^{-1}$ to recover the original transmitted data bits d_k. An inverse cross-correlation stage is added after the bank of matched filters which multiplies the vector $\mathbf{x}$ by $\mathbf{S}^{-1}$, followed by a bank of threshold devices to obtain the bit estimates

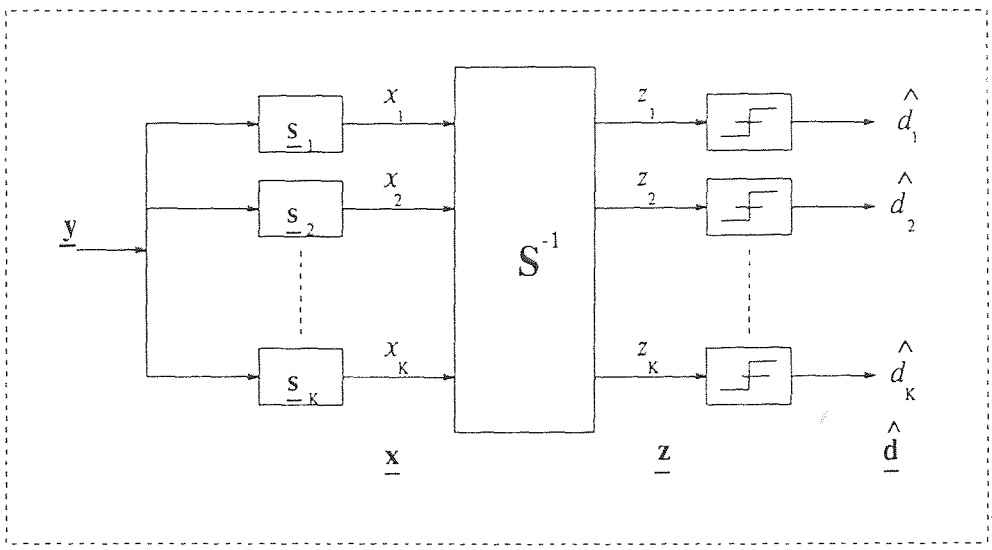


Figure 2.7 The decorrelating detector.

$\hat{d}_k$. Thus, the output following the inverse stage is given as:

$$\begin{aligned} \mathbf{z} &= \mathbf{S}^{-1}\mathbf{x} \\ &= \mathbf{S}^{-1}\mathbf{S}\mathbf{d} + \mathbf{S}^{-1}\mathbf{m} \\ &= \mathbf{d} + \mathbf{o} \end{aligned} \tag{2.78}$$

where the vector $\mathbf{o}$ represents the new Gaussian noise component with variance $\sigma_o^2 = \sigma_m^2 \mathbf{S}^{-1}$. The estimate obtained from the threshold device is:

$$\hat{\mathbf{d}} = \mathrm{sgn}(\mathbf{z}). \tag{2.79}$$

The probability of bit error is thus equal to the probability of recovering an incorrect bit from the output of the threshold device given in [2] as:

$$P_e = Q\left[\sqrt{\frac{a_k}{\sigma_n^2 \mathbf{S}_{kk}^{-1}}}\right] \tag{2.80}$$

where a_k is the energy of the desired data bit and $\mathbf{S}_{kk}^{-1}$ represents a diagonal term on the inverse of the cross-correlation matrix. The advantage of this detector is that it does not require knowledge of the energies of any of the active users, thus making it

immune to strong multiple-user power levels. The decorrelating detector receiver is not affected by the near-far problem that was described in Chapter 1. It produces a constant P_e for strong or weak multiple user interference power levels.

CHAPTER 3

THE ADAPTIVE CORRELATOR RECEIVER (ADC)

In this chapter the ADC receiver will be introduced as a means of demodulating a BPSK DS spread spectrum signal in the presence of various disturbances. The disturbances considered here will include a tone jammer, a narrow-band jammer, multipath, and multiple users on the channel. This chapter will introduce the ADC receiver as a means of demodulating a spread spectrum signal when either of the above mentioned disturbances are present in the received signal. The circuit of the ADC is introduced in the first section. The following sections will present three cases. The first case explains the signal model which will be used for the ADC when the transmitted signal is corrupted by a jammer. The second case will show how the ADC removes multipath interference, and the last case will show how the ADC suppresses MAI for a two-user case. The probability of bit error is derived for all three cases. The last section of this chapter will prove how the ADC is a convolution of separate FIR filters which separately remove the interferences of the three cases, showing how the ADC is a versatile filter for spread spectrum demodulation.

3.1 The ADC

The ADC receiver that is analyzed in this work consists of a tapped delay line filter with the number of taps equal to the number of chips (L) in the PN spreading code. Each tap delay is equal to one chip duration; thus, one full data bit can be applied to the filter input. The ADC circuit diagram is shown in Figure 3.1. The tap weights of the filter are adjusted so that the output of the filter is an estimate of the desired data bit $\hat{d}_i$. The error signal ϵ_i^* is the difference between $\hat{d}_i$ and the actual data bit

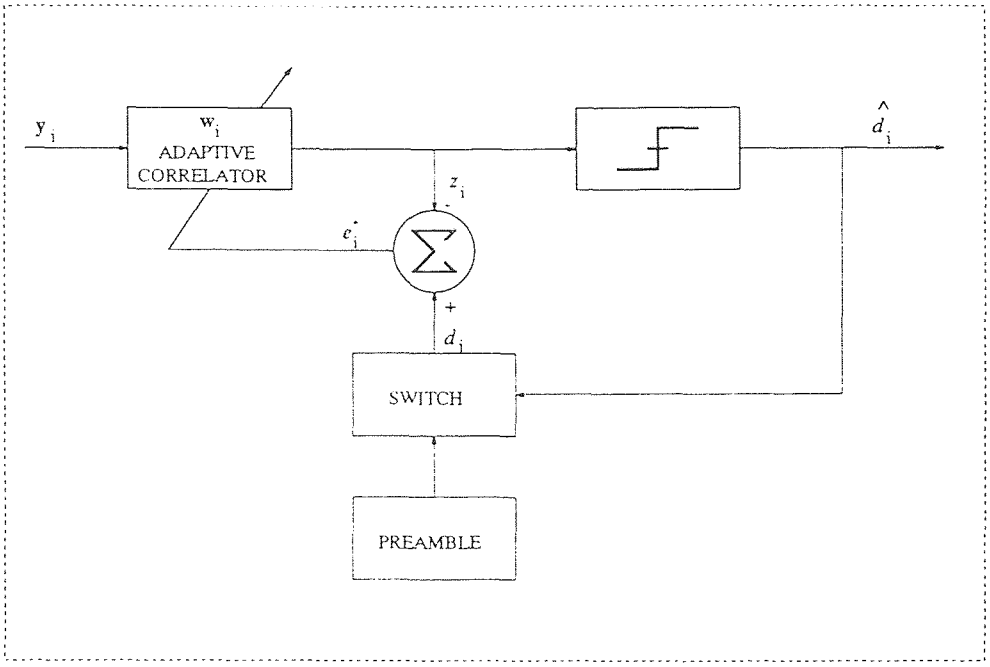


Figure 3.1 The adaptive correlator receiver.

and is used in an LMS or RLS algorithm to adaptively adjust the tap weight vector $\mathbf{w}_i$.

The received signal is sampled at a rate equal to the chip duration. If we assume that the system is synchronized with the incoming signal, samples are taken at each chip transition interval for L chips to produce the vector $\mathbf{y}_i$. The signal is then shifted into the filter. When the full data bit vector $\mathbf{y}_i$ is aligned in the filter, each sample is multiplied by its appropriate weight. Then, the products are summed to produce the signal z_i. Subtracting this estimate from the desired bit d_i produces the error term e_i, which is used to adapt the tap weights using the LMS or RLS algorithm. After the weights are adjusted, the process is repeated for each new signal sample.

Initially, the data bits d_i are supplied to the equalizer via a preamble of data sequence. This process allows the adaptive algorithm to converge to the optimal tap weights. When the system is functioning properly, a fairly good replica of the transmitted sequence is being produced at the output of the threshold device. Consequently, this output can be used as the *desired* response when calculating the error e_i for adaptive equalization. This is called *decision directed* adaptive equalization, because the receiver tries to *learn* by using its own decisions [3]. This procedure works if the average BER is relatively small, in which case the decisions made by the receiver are accurate most of the time. However, if the BER is high, decision directed equalization will cause error propagations throughout the sequence at the output of the receiver. After convergence of the weight vector $\mathbf{w}_i$, we can use the output of the decision circuit for the estimate of d_i. The decision circuit is a threshold device which produces a $+1$ for a value of $\hat{d}_i$ above zero and a -1 for a value of $\hat{d}_i$ below zero. When the ADC is operating in this mode it is said to be *decision directed.*

Unlike the notch filter receiver described in the previous chapter, the ADC does not find the MMSE between the output of the filter and the received signal. Instead, the ADC's tap weights are adjusted so as to achieve an MMSE between the output of the filter z_i and the desired data bit d_i.

If the ADC receiver is used to demodulate a DS spread spectrum signal in the presence of AWGN, the tap weights of the filter will converge to the PN code sequence. This eliminates the need for a separate PN code sequence demodulator. In addition to this *automatic* PN demodulation, the tap weights will automatically adjust to values which eliminate any undesired interference in the spread spectrum signal. The filter will act as a notch filter when a narrow-band interference signal is present. In addition to a notch filter, the ADC acts as a channel estimator when multipath interference is present. Essentially, the filter learns the frequency response of the channel and adjusts the tap weights accordingly so the frequency response of

the weight vector matches the channel's frequency response in order to demodulate the desired signal. Therefore, this receiver serves the same function as the RAKE receiver of Chapter 2. In essence, the ADC is a filter whose tap-weights are adjusted to be the convolution of three separate filters; one to suppress narrow-band interference, one to perform channel equalization, and yet another to do PN demodulation.

3.2 Case 1: The ADC for Narrow-Band Jammer Suppression

In this section, we develop the signal model for the ADC when a tone or narrow-band jammer is present in the received signal. The auto-correlation of the received signal and the optimum weight vector for the jammer scenario is derived. Finally, the BER for the ADC with a jammer in the received signal is calculated.

The signal model for BPSK DS spread spectrum with narrow-band jamming and one user on the channel is given by equation 2.1, re-written here for convenience:

$$y(t) = \sqrt{a}\, d(t)s(t) + j(t) + n(t) \tag{3.1}$$

where $d(t)$ is the data bit with duration T_b and signal energy a. The signal $s(t)$ is the PN code sequence also with duration T_b and with L chips, each with duration T_c, where $LT_c = T_b$. The term $n(t)$ is AWGN and the term $j(t)$ is either a Gaussian tone jammer or a narrow-band Gaussian jammer. The tone jammer signal is given as:

$$j_T(t) = A_j \cos(\omega_j t + \theta_j) \tag{3.2}$$

where θ_j ,the jammer phase, is a uniform random variable in the interval $[0, 2\pi]$. The auto-correlation of the tone jammer is derived from equation 2.3 and is given as:

$$\mathbf{R}_{jj} = \begin{bmatrix} J & J\cos(\omega_j) & \dots & J\cos((L-1)\omega_j) \\ J\cos(-\omega_j) & J & \dots & J\cos((L-2)\omega_j) \\ J\cos(-2\omega_j) & J\cos(-\omega_j) & \dots & J\cos((L-3)\omega_j) \\ \vdots & \vdots & & \vdots \\ J\cos(-(L-1)\omega_j) & J\cos(-(L-2)\omega_j) & \dots & J \end{bmatrix}. \tag{3.3}$$

Similarly, the auto-correlation matrix for a narrow-band Gaussian jammer is derived from equation 2.4 and is given as:

$$\mathbf{R}_{jj}^{NB} = \begin{bmatrix} J & J\frac{\sin(\omega_a \frac{1}{2})}{\omega_a \frac{1}{2}}\cos(\omega_j) & \dots & J\frac{\sin(\omega_a \frac{L-1}{2})}{\omega_a \frac{L-1}{2}}\cos(\omega_j) \\ J\frac{\sin(-\omega_a \frac{1}{2})}{-\omega_a \frac{1}{2}}\cos(\omega_j) & J & \dots & J\frac{\sin(\omega_a \frac{L-2}{2})}{\omega_a \frac{L-2}{2}}\cos(\omega_j) \\ J\frac{\sin(-\omega_a)}{-\omega_a}\cos(\omega_j) & J\frac{\sin(-\omega_a \frac{1}{2})}{-\omega_a \frac{1}{2}}\cos(\omega_j) & \dots & J\frac{\sin(\omega_a \frac{L-3}{2})}{\omega_a \frac{L-3}{2}}\cos(\omega_j) \\ \vdots & \vdots & & \vdots \\ J\frac{\sin(-\omega_a \frac{L-1}{2})}{-\omega_a \frac{L-1}{2}}\cos(\omega_j) & J\frac{\sin(-\omega_a \frac{L-2}{2})}{-\omega_a \frac{L-2}{2}}\cos(\omega_j) & \dots & J \end{bmatrix}. \tag{3.4}$$

For our model, the number of taps of our ADC is equal to the number of chips of the PN code (L). If the received signal $y(t)$ is sampled at the spreading chip rate, we may represent the received signal as:

$$y(lT_c) = \sqrt{a}\, d(lT_c)s(lT_c) + j(lT_c) + n(lT_C). \tag{3.5}$$

The received signal in vector form with L elements over one bit interval is:

$$\mathrm{y}_i = d_i\mathrm{s} + \mathrm{j} + \mathrm{n}. \tag{3.6}$$

If we limit ourselves to observe one bit interval we can drop the subscript i and assuming that $T_c = 1$, we can represent the received signal vectors as:

$$\mathrm{y} = [y_0, y_1, y_2, \dots, y_{L-1}]^T \tag{3.7}$$

$$\mathrm{s} = [s_0, s_1, s_2, \dots, s_{L-1}]^T \tag{3.8}$$

where the elements of s form the bi-polar PN code sequence and are either $+1$ or -1. The interference vectors are represented as:

$$\mathrm{j} = [j_0, j_1, j_2, \dots, j_{L-1}]^T \tag{3.9}$$

$$\mathrm{n} = [n_0, n_1, n_2, \dots, n_{L-1}]^T. \tag{3.10}$$

The data bit d is a random variable equal to $+1$ or -1 with probability of $\frac{1}{2}$. The noise samples n_l have zero mean with variance σ_n^2. The auto-correlation matrix of y

is given as:

$$\begin{aligned}
\mathbf{R}_{yy} &= E[\mathrm{y}\mathrm{y}^H] \\
&= E[(\sqrt{a}\, d\mathrm{s} + \mathrm{j} + \mathrm{n})(\sqrt{a}\, d\mathrm{s}^H + \mathrm{j}^H + \mathrm{n}^H)] \\
&= E[ad^2\mathrm{s}\mathrm{s}^H + \sqrt{a}\, d\mathrm{s}\mathrm{j}^H + \sqrt{a}\, d\mathrm{s}^H\mathrm{n} + \sqrt{a}\, d\mathrm{j}\mathrm{s}^H \\
&\quad + \mathrm{j}\mathrm{j}^H + \mathrm{j}\mathrm{n}^H + \sqrt{a}\, d\mathrm{n}\mathrm{s}^H + \mathrm{n}\mathrm{j}^H + \mathrm{n}\mathrm{n}^H)]. \qquad (3.11)
\end{aligned}$$

We assume that the signals s, j, and n are statistically independent, with s and n having zero mean. Therefore equation 3.11 for the auto-correlation matrix becomes:

$$\begin{aligned}
\mathbf{R}_{yy} &= E[ad^2\mathrm{s}\mathrm{s}^H + \mathrm{j}\mathrm{j}^H + \mathrm{n}\mathrm{n}^H] \\
&= ad^2E[\mathrm{s}\mathrm{s}^H] + E[\mathrm{j}\mathrm{j}^H] + E[\mathrm{n}\mathrm{n}^H] \\
&= aE[\mathrm{s}\mathrm{s}^H] + E[\mathrm{j}\mathrm{j}^H] + E[\mathrm{n}\mathrm{n}^H] \\
&= a\mathrm{s}\mathrm{s}^H + \mathbf{R}_{jj} + \sigma_{\mathrm{n}}^2\mathbf{I} \qquad (3.12)
\end{aligned}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{R}_{jj}$ is the auto-correlation matrix of the jammer signal given as equation 3.3 for a tone jammer and equation 3.4 for a narrow-band jammer.

The error e^* is the difference between the output of the correlator z and the desired bit d. It is given as:

$$\begin{aligned}
e^* &= d - z \\
&= d - \mathbf{w}^H\mathbf{y}. \qquad (3.13)
\end{aligned}$$

The value z is the demodulated spread spectrum estimate bit with residual noise components and $\mathbf{w}^H$ is the tap weight vector. The MMSE is:

$$\mathcal{J}(\mathbf{w}) = E[ee^*]. \qquad (3.14)$$

The optimum weight vector $\mathbf{w}_{\text{opt}}$ is calculated using the Wiener-Hopf equation if the system is considered stationary:

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{yy}^{-1}\mathbf{P}_{dy} \tag{3.15}$$

where the cross-correlation between $\mathbf{y}$ and the data bit d is given as:

$$\mathbf{P}_{dy} = E[\mathbf{y}d^*]. \tag{3.16}$$

Therefore, the MMSE is given as:

$$\begin{aligned}
\mathcal{J}_{\text{min}} &= E[ee^*] \\
&= E[(d-\hat{d})(d-\hat{d})^*] \\
&= E[(d-\mathbf{w}_{\text{opt}}^H\mathbf{y})(d^*-\mathbf{w}_{\text{opt}}^H\mathbf{y}^*)] \\
&= 1-\mathbf{P}_{dy}^H\mathbf{w}_{\text{opt}}-\mathbf{w}_{\text{opt}}^H\mathbf{P}_{dy}+\mathbf{w}_{\text{opt}}^H\mathbf{R}_{yy}\mathbf{w}_{\text{opt}}.
\end{aligned} \tag{3.17}$$

Since the signals are statistically independent, the cross-correlation vector is reduced to:

$$\begin{aligned}
\mathbf{P}_{dy} &= E[\mathbf{s}d^*] \\
&= E[\mathbf{s}] \\
&= \mathbf{s}.
\end{aligned} \tag{3.18}$$

Substituting equations 3.12 and 3.18 into equation 3.15 gives the optimum weight vector as:

$$\begin{aligned}
\mathbf{w}_{\text{opt}} &= \mathbf{R}_{yy}^{-1}\mathbf{P}_{dy} \\
&= [\mathbf{s}\mathbf{s}^H+\mathbf{R}_{jj}+\sigma_n^2\mathbf{I}]^{-1}\mathbf{s}
\end{aligned} \tag{3.19}$$

where $\mathbf{R}_{jj}$ is either equation 3.3 or equation 3.4, depending on the type of jammer. The output of the optimum filter is:

$$\begin{aligned}
z &= \mathbf{w}_{\text{opt}}^T\mathbf{y} \\
&= \mathbf{P}_{dy}^H\mathbf{R}_{yy}^{-1}\mathbf{y}
\end{aligned} \tag{3.20}$$

and the estimate of the desired bit d is simply:

$$\hat{d} = \text{sgn}(z). \tag{3.21}$$

In order to gauge the performance of the system, the BER is calculated. If the ADC's weight vector has converged to a near optimum value, and the system is operated in a decision-directed mode, then the probability that a wrong decision is made at the threshold device is the probability of a bit error. The output of the ADC is:

$$z = \mathbf{w}^T\mathbf{y} \tag{3.22}$$

where $\mathbf{w}$ is the value of the weight vector which the ADC has converged to. Assuming that $d = 1$ and $d = -1$ with equal probability, the probability of bit error is:

$$\begin{aligned} P_e &= \text{Prob}[z < 0|d = 1] = \text{Prob}[z < 0|d = -1] \\ &= \text{Prob}[\mathbf{w}^T\mathbf{y} < 0|d = 1] \\ &= \text{Prob}[\sqrt{a}\, d\mathbf{w}^T\mathbf{s} + \mathbf{w}^T\mathbf{j} + \mathbf{w}^T\mathbf{n} < 0|d = 1] \\ &= \text{Prob}[\sqrt{a}\, \mathbf{w}^T\mathbf{s} + \mathbf{w}^T\mathbf{j} + \mathbf{w}^T\mathbf{n} < 0]. \end{aligned} \tag{3.23}$$

Using the fact that the noise signal and the narrow-band jammer are Gaussian distributed, equation 3.23 reduces to

$$\begin{aligned} P_e &= \text{Prob}[\mathbf{w}^T\mathbf{j} + \mathbf{w}^T\mathbf{n} < -\sqrt{a}\, \mathbf{w}^T\mathbf{s}] \\ &= Q\left(\frac{\sqrt{a}\, \mathbf{w}^T\mathbf{s}}{\sqrt{\mathbf{w}^T\mathbf{R}_{jj}\mathbf{w} + \sigma_n^2\mathbf{w}^T\mathbf{w}}}\right) \end{aligned} \tag{3.24}$$

where $\mathbf{w}^T\mathbf{R}_{jj}\mathbf{w}$ is the variance of $\mathbf{w}^T\mathbf{j}$, $\sigma_n^2\mathbf{w}^T\mathbf{w}$ is the variance of $\mathbf{w}^T\mathbf{n}$, and σ_n^2 is the variance of the noise.

Equation 3.24 is similar to the result obtained for a BPSK signal with AWGN in [4]:

$$P_e = \text{Q}\left(\sqrt{\text{SNR}}\right), \tag{3.25}$$

the difference being the addition of the interference covariance in the demodulator in equation 3.24. Equation 3.24 is also exactly equivalent to the expression for the probability of bit error which was derived in [8] when Gaussian interference is assumed and the probability of occurrence of each data bit equal to $\frac{1}{2}$.

3.3 Case 2: The ADC for Multipath Suppression

In this section, we develop the signal model for the ADC when the received signal is corrupted by multipath. The optimum weight vector for the ADC is calculated for this model and the BER is derived. In order to remove the negative effects of multipath from the received signal, the ADC receiver is used to estimate the channel. Since the sampling rate is $\frac{T_c}{2}$, the number of tap-delays of the ADC will be increased by a factor of 2 to $2L$ (see Section 2.4). Due to the multipath effect, the received signal vector will also be increased from $2L$ to $2L + L_h$ elements, where L_h is the number of tap weights of the channel model. As was explained in Section 2.4, this increase is due to the convolution of $\mathbf{u}$ with $\mathbf{h}$. This increase in length will cause the samples of the previous data bit to spill over into the sampling interval of the current bit, thus adding further interference to the signal. Figure 3.2 illustrates this phenomena. The signal will be the sum of the tail-end of the previous signal and the beginning of the current signal. We define this signal as:

$$\mathbf{f}_i = (\mathbf{u}_i \otimes \mathbf{h})_{0 \text{to} L-1} + (\mathbf{u}_{i-1} \otimes \mathbf{h})_{L-1 \text{to} L-1+L_h}. \tag{3.26}$$

We keep the time subscript i in this analysis to distinguish between the current bit and the previous bit. The first term on the right side of the equation is the current data bit and code, and the second term on the right hand side represents the previous data bit and code that arrives late to the receiver. Therefore, the new signal model which includes the multipath interference effect is given in vector form as:

$$\mathbf{y}_i = \mathbf{f}_i + \mathbf{j} + \mathbf{n}$$

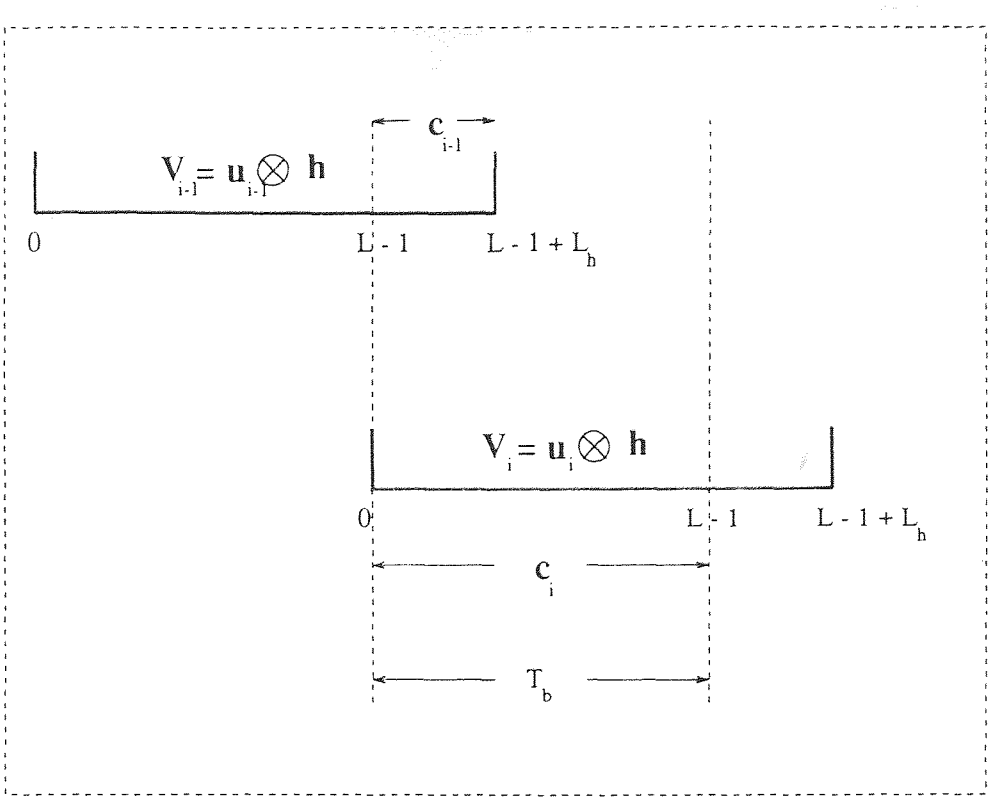


Figure 3.2 Effect of multipath on the received signal vector.

$$\begin{aligned} &= (\mathbf{u}_i \otimes \mathbf{h}) + (\mathbf{u}_{i-1} \otimes \mathbf{h}) + \mathbf{j} + \mathbf{n} \\ &= \sqrt{a}\, d^l \mathbf{c}_i + \sqrt{a}\, d^r \mathbf{c}_{i-1} + \mathbf{j} + \mathbf{n} \end{aligned} \tag{3.27}$$

where $\mathbf{c}_i$ is the PN code sequence with multipath effects for the current bit d^l and $\mathbf{c}_{i-1}$ is the PN code with multipath effects for the previous bit d^r. The signal a is the energy of the data bit d, which is constant for all bits. The data bits d^l and d^r are discrete random variables equal to $+1$ or -1 with equal probability. If no multipath is present, $\mathbf{c}_i$ will be equal to the PN code $\mathbf{s}$, while $\mathbf{c}_{i-1}$ will equal zero. The auto-correlation of the signal $\mathbf{y}$ thus becomes:

$$\begin{aligned} \mathbf{R}_{yy} &= E[\mathbf{y}_i \mathbf{y}_i^H] \\ &= E[(\sqrt{a}\, d^l \mathbf{c}_i + \sqrt{a}\, d^r \mathbf{c}_{i-1} + \mathbf{j} + \mathbf{n}) \\ &\quad (\sqrt{a}\, d^l \mathbf{c}_i + \sqrt{a}\, d^r \mathbf{c}_{i-1} + \mathbf{j} + \mathbf{n})^H]. \end{aligned} \tag{3.28}$$

Following the same logic behind equation 3.12, the auto-correlation matrix becomes:

$$\mathbf{R}_{yy} = a\mathbf{c}_i\mathbf{c}_i^H + a\mathbf{c}_{i-1}\mathbf{c}_{i-1}^H + \mathbf{R}_{jj} + \sigma_n^2\mathbf{I}. \tag{3.29}$$

Similarly, the cross-correlation vector becomes:

$$\begin{aligned} \mathbf{P}_{dy} &= E[\mathbf{y}\,(d^l)^*] \\ &= E[(\sqrt{a}\,d^l\mathbf{c}_i + \sqrt{a}\,d^r\mathbf{c}_{i-1} + \mathbf{j} + \mathbf{n})\,(d^l)^*] \\ &= \sqrt{a}\,\mathbf{c}_i \end{aligned} \tag{3.30}$$

and the optimum weight vector $\mathbf{w}_{\text{opt}}$ for the ADC is:

$$\begin{aligned} \mathbf{w}_{\text{opt}} &= \mathbf{R}_{yy}^{-1}\mathbf{P}_{dy} \\ &= (a\mathbf{c}_i\mathbf{c}_i^H + a\mathbf{c}_{i-1}\mathbf{c}_{i-1}^H + \mathbf{R}_{jj} + \sigma_n^2\mathbf{I})^{-1}\sqrt{a}\,\mathbf{c}_i \end{aligned} \tag{3.31}$$

where $\mathbf{R}_{jj}$ is either equation 3.3 or equation 3.4, depending on the type of jammer used. The output of the ADC is:

$$\begin{aligned} z_i &= \mathbf{w}^H\mathbf{y}_i \\ &= \mathbf{c}_i^H\mathbf{R}_{yy}^{-1}\mathbf{y}. \end{aligned} \tag{3.32}$$

When an adaptive algorithm is used to find the MMSE, the weights will converge to values which jointly suppress both multipath and interference while also demodulating the spread spectrum signal. If there is no narrow-band interference in the received signal $\mathbf{y}_i$, the ADC will perform channel estimation. In other words, the impulse response of the tap weights will be equal to the impulse response of the channel.

The bit probability is calculated as before, in which the probability of error is equal to the probability of the decision device producing an error. When z_i is the output of the ADC and $\mathbf{w}_i$ is the weight vector which the ADC converges to, the equation is given as:

$$P_e = \text{Prob}[z_i < 0 | d^l = 1] = \text{Prob}[z_i > 0 | d^l = -1]$$

$$= \text{Prob}[\sqrt{a}\, d^l \mathbf{w}^H \mathbf{c}_i + \sqrt{a}\, d^r \mathbf{w}^H \mathbf{c}_{i-1} + \mathbf{w}^H \mathbf{j} + \mathbf{w}^H \mathbf{n} < 0 | d^l = 1]$$

$$= \text{Prob}[\sqrt{a}\, d^l \mathbf{w}^H \mathbf{c}_i + \sqrt{a}\, d^r \mathbf{w}^H \mathbf{c}_{i-l} + \mathbf{w}^H \mathbf{j} + \mathbf{w}^H \mathbf{n} < 0]. \quad (3.33)$$

Using Bayes' Theorem, equation 3.33 is further resolved:

$$\begin{aligned} P_e &= \frac{1}{2}\text{Prob}[\sqrt{a}\, d^l \mathbf{w}^H \mathbf{c}_i + \sqrt{a}\, d^r \mathbf{w}^H \mathbf{c}_{i-1} + \mathbf{w}^H \mathbf{j} + \mathbf{w}^H \mathbf{n} < 0] \\ &\quad + \frac{1}{2}\text{Prob}[\sqrt{a}\, \mathbf{w}^H \mathbf{c}_i - \sqrt{a}\, \mathbf{w}^H \mathbf{c}_{i-1} + \mathbf{w}^H \mathbf{j} + \mathbf{w}^H \mathbf{n} < 0] \\ &= \frac{1}{2} Q\left(\frac{\sqrt{a}\, \mathbf{w}_i^H \mathbf{c}_i + \sqrt{a}\, \mathbf{w}_i^H \mathbf{c}_{i-1}}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\ &\quad + \frac{1}{2} Q\left(\frac{\sqrt{a}\, \mathbf{w}_i^H \mathbf{c}_i - \sqrt{a}\, \mathbf{w}_i^H \mathbf{c}_{i-1}}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right). \end{aligned} \quad (3.34)$$

3.4 Case 3: The ADC for MAI Suppression

In this section, we develop the signal model for the ADC for co-channel interference. The optimum weight vector for the ADC is calculated for this interference and the BER equation is derived.

The ADC receiver can be used in CDMA to suppress MAI in much the same way the conventional multi-user detector is used to demodulate the data bit. The circuit consists of a bank of ADC's which adjusts the tap weights to remove the MAI, followed by a bank of threshold devices. The concept is similar to the conventional multi-user detector except that the taps of the ADC are automatically adjusted to suppress the MAI. The result is that the data bit is demodulated from the received signal and the other users are removed even if the PN code signatures are not mutually orthogonal. The tap weights are adjusted so as to produce a MMSE between the output of the tap-delay line and the desired data bit. The signal model is given in vector form for one bit interval as equation 2.70:

$$\mathbf{y} = \sum_{k=1}^{K} \sqrt{a_k}\, d_k \mathbf{s}_k + \mathbf{n} \quad (3.35)$$

where the subscript k represents a user, and the signal a_k is the energy of each user.

The optimum weight vector $\mathbf{w}_{\mathrm{opt}_k}$ for each ADC is given by the Wiener-Hopf equation as before as:

$$\mathbf{w}_{\mathrm{opt}_k} = \mathbf{R}_{yy}^{-1}\mathbf{P}_{dy}. \tag{3.36}$$

where $\mathbf{w}_{\mathrm{opt}_k}$ is the optimum weight vector for the k-th user and $\mathbf{R}_{yy}$ is the new auto-correlation matrix defined as:

$$\begin{aligned} \mathbf{R}_{yy} &= E[\mathbf{y}\mathbf{y}^H] \\ &= \sum_{k}^{K} a_k \mathbf{s}_k \mathbf{s}_k^H + \sigma_n^2 \mathbf{I}. \end{aligned} \tag{3.37}$$

The cross-correlation vector $\mathbf{P}_{dy}$ is defined for the k-th user as:

$$\begin{aligned} \mathbf{P}_{dy} &= E[\mathbf{y}d_k^*] \\ &= \sqrt{a_k}\,\mathbf{s}_k. \end{aligned} \tag{3.38}$$

The output of the tap-delay line is the vector dot-product of the ADC's weight vector with the received signal, and given for the k-th user as:

$$z_k = \mathbf{w}_k^H \mathbf{y}. \tag{3.39}$$

The estimate data bit is given as

$$\hat{d}_k = \mathrm{sgn}(z_k). \tag{3.40}$$

In a manner similar to the way in which the probability of error for the conventional multi-user detector was calculated (see Section 2.5.1), the probability of error for the ADC with two users is given as:

$$\begin{aligned} P_e &= \frac{1}{2}Q\left[\frac{\sqrt{a_1}\,\mathbf{w}_1^H\mathbf{s}_1 + \sqrt{a_2}\,\mathbf{w}_1^H\mathbf{s}_2}{\sqrt{\sigma_n^2\mathbf{w}_1^H\mathbf{w}_2}}\right] \\ &\quad + \frac{1}{2}Q\left[\frac{\sqrt{a_1}\,\mathbf{w}_1^H\mathbf{s}_1 - \sqrt{a_2}\,\mathbf{w}_1^H\mathbf{s}_2}{\sqrt{\sigma_n^2\mathbf{w}_1^H\mathbf{w}_1}}\right] \end{aligned} \tag{3.41}$$

where $\sigma_n^2\mathbf{w}_1^H\mathbf{w}_1$ is the variance of the term $\mathbf{w}_1^H\mathbf{n}$ and the subscript 1 and 2 denote user 1 and 2, respectively. User 1 is the desired user and the vector $\mathbf{w}_1$ is the weight vector which the ADC for user 1 converges to.

If a jammer is present in the received signal vector $\mathbf{y}$, the signal model now becomes:

$$\mathbf{y} = \sum_{k=1}^{K} \sqrt{a_k}\, d_k \mathbf{s}_k + \mathbf{j} + \mathbf{n}. \tag{3.42}$$

The equation for the auto-correlation matrix then becomes:

$$\mathbf{R}_{yy} = \sum_{k}^{K} a_k \mathbf{s}_k \mathbf{s}_k^H + \mathbf{R}_{jj} + \sigma_n^2 \mathbf{I}. \tag{3.43}$$

The probability of bit error for the ADC with two users is thus:

$$\begin{aligned} P_e &= \frac{1}{2} Q\left(\frac{\sqrt{a_1}\, \mathbf{w}_1^T \mathbf{s}_1 + \sqrt{a_2}\, \mathbf{w}_1^T \mathbf{s}_2}{\sqrt{\mathbf{w}_1^T \mathbf{R}_{jj} \mathbf{w}_1 + \sigma_n^2 \mathbf{w}_1^T \mathbf{w}_1}} \right) \\ &\quad + \frac{1}{2} Q\left(\frac{\sqrt{a_1}\, \mathbf{w}_1^T \mathbf{s}_1 - \sqrt{a_2}\, \mathbf{w}_1^T \mathbf{s}_2}{\sqrt{\mathbf{w}_1^T \mathbf{R}_{jj} \mathbf{w}_1 + \sigma_n^2 \mathbf{w}_1^T \mathbf{w}_1}} \right) \end{aligned} \tag{3.44}$$

where $\mathbf{w}_1^T \mathbf{R}_{jj} \mathbf{w}_1$ is the variance of the jammer, and $\sigma_n^2 \mathbf{w}_1^T \mathbf{w}_1$ is the variance of $\mathbf{w}_1^T \mathbf{n}$.

3.5 Convolution of FIR Filters

The intent of this work is to introduce a tapped-delay line filter capable of suppressing both multipath and jammer interferences from a spread spectrum signal in a multiple-user environment while also demodulating the spread spectrum signal. The filter introduced here performs the function of separate FIR filters; one filter to remove the interference, another filter to remove the co-channel interference, a RAKE to mitigate the multipath interference, and yet another filter to demodulate the spread spectrum signal and retrieve the transmitted bit. The adaptive correlator receiver performs the same function as these structures combined in series (See Figure 3.3). This section will show that the convolution of separate filters will yield the same output as generated for the ADC in the previous sections.

If we assume that our signal consists solely of the PN code sequence and a jammer, a prediction filter, as depicted in Figure 2.4, may be used to remove the narrow-band interference signal by, relying on the strong correlation of the interference. This filter is followed by a PN decorrelator to demodulate the spread

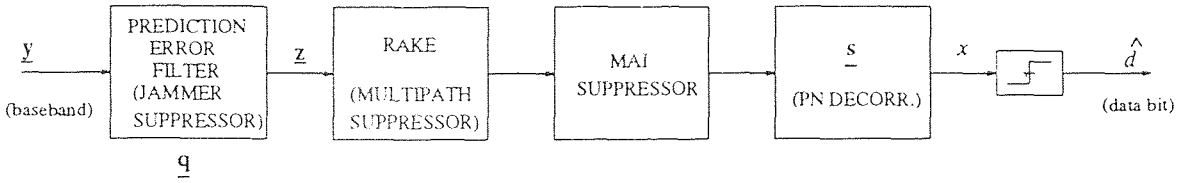


Figure 3.3 FIR filters in series.

spectrum signal [8],[9] and [11]. The receiver would consist of two stages; one stage to remove the jammer signal, and the other stage to demodulate the spread spectrum signal with tap weights of this filter equal to the PN code sequence. To show that this two stage structure is equivalent to equation 3.20 when only a jammer is present, we proceed as follows. Let $\mathbf{q}$ be the coefficient vector of the interference suppression filter. The interference suppression filter is designed such that its output is a vector $\mathbf{z}$ obtained by suppressing the interference in the input vector $\mathbf{y}$. Conceptually, this requires a set of L parallel filters, each reproducing one of the L chips in the PN code. A diagram of this filter is depicted in Figure 3.4. Thus z_l, the l-th element of the output vector, is given by:

$$z_l = \mathbf{q}_l^T \mathbf{y} \qquad l = 0, 1, 2, 3, \ldots, L-1 \tag{3.45}$$

where $\mathbf{y}$ is the filter coefficient found by minimizing the output power $\mathbf{q}_l^T \mathbf{R}_{yy} \mathbf{q}_l$ subject to the constraint of passing the l-th bit, i.e., $\mathbf{q}_l^T \mathbf{u}_l = 1$, where $\mathbf{u}_l^T$ is a unit vector with the l-th element set to unity.

The solution to this constrained optimization is given by:

$$\mathbf{q}_l = \mathbf{R}_{yy}^{-1} \mathbf{u}_l. \tag{3.46}$$

Hence, the output vector $\mathbf{z}$ can be written as:

$$\begin{aligned} \mathbf{z} &= [\mathbf{q}_0^T \mathbf{y}, \mathbf{q}_1^T \mathbf{y}, \mathbf{q}_2^T \mathbf{y}, \ldots, \mathbf{q}_{L-1}^T \mathbf{y}]^T \\ &= [(\mathbf{R}_{yy}^{-1} \mathbf{u}_\mathbf{y}^T]^T \end{aligned}$$

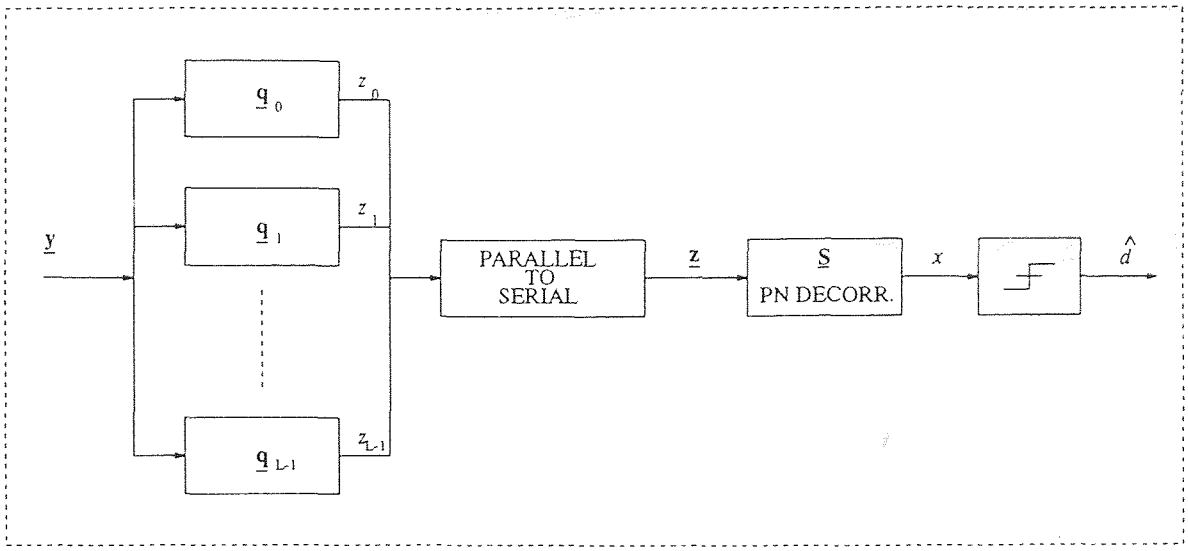


Figure 3.4 Interference-suppressant filter used to demodulate the data bit.

$$\begin{aligned} &= [\mathbf{u}_0^T\mathbf{R}_{yy}^{-1}\mathbf{y}, \mathbf{u}_1^T\mathbf{R}_{yy}^{-1}\mathbf{y}, \ldots, \mathbf{u}_{L-1}^T\mathbf{R}_{yy}^{-1}\mathbf{y}] \\ &= \begin{bmatrix} \mathbf{u}_0^T \\ \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_{L-1}^T \end{bmatrix} \mathbf{R}_{yy}^{-1}\mathbf{y} \\ &= \mathbf{I}\mathbf{R}_{yy}^{-1}\mathbf{y} \\ &= \mathbf{R}_{yy}^{-1}\mathbf{y}. \end{aligned} \tag{3.47}$$

The output x is the correlation of $\mathbf{z}$ with the PN code sequence $\mathbf{s}$:

$$\begin{aligned} x &= \mathbf{s}^T\mathbf{z} \\ &= \mathbf{s}^T\mathbf{R}_{yy}^{-1}\mathbf{y} \end{aligned} \tag{3.48}$$

which is exactly equal to equation 3.22 for the ADC, re-written here for convenience:

$$\begin{aligned} x &= \mathbf{w}^T\mathbf{y} \\ &= \mathbf{s}^T\mathbf{R}_{yy}^{-1}\mathbf{y}. \end{aligned} \tag{3.49}$$

If multipath interference is present in the received signal, we can include a RAKE receiver to remove it. The structure would look similar to the one depicted

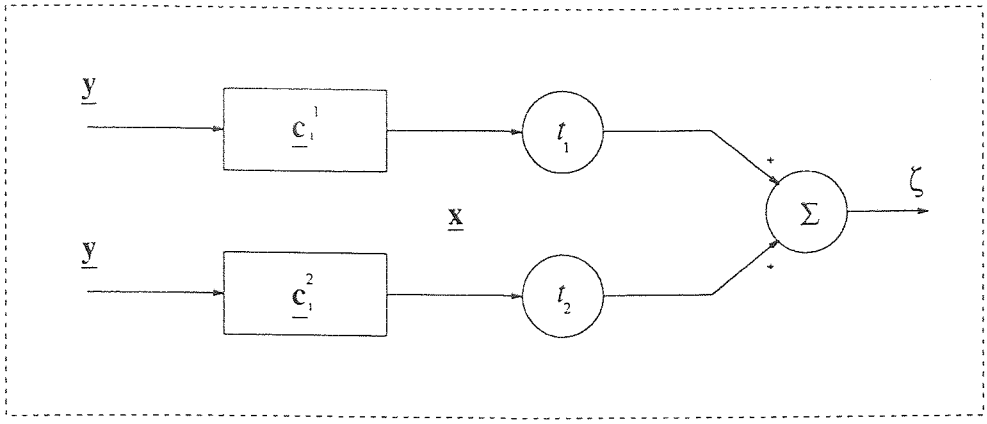


Figure 3.5 Separate filter structure used to remove multipath and MAI.

in Figure 3.4 with the exception of the PN code vector s replaced with the vector $\mathbf{c}_i$ in the decorrelating stage. The signal $\mathbf{c}_i$ was introduced in Section 3.3 and was shown to be the convolution of $\sqrt{a}\,d\mathbf{s}$ with $\mathbf{h}$ in the time interval T_b. Therefore, equation 3.49 may re-defined as:

$$x = \mathbf{c}_i^H \mathbf{R}_{yy}^{-1} \mathbf{y} \tag{3.50}$$

which is exactly equal to equation 3.32.

We now show how the ADC is equivalent to a structure which illuminates the MAI as well as multipath. This structure would consist of a bank of separate matched filters for each separate user. The tap weights of each matched filter would be matched to the convolution of each separate PN code with the channel model. This structure for two users is shown in Figure 3.5. The first stage would remove the multipath in the signal. The output of the matched filters would produce a vector $\mathbf{x}$ consisting of K elements, where K is the number of users. In our model we have only two users. To remove the MAI, $\mathbf{x}$ will be passed through a separate filter consisting of tap weights $\{t_k\}$ to produce the output ζ.

This output, ζ, is the vector dot-product of $\mathbf{t}$ with $\mathbf{x}$:

$$\zeta = \mathbf{t}^H \mathbf{x}. \tag{3.51}$$

The vector $\mathbf{t}$ is calculated using the Wiener-Hopf equation and given as:

$$\mathbf{t} = \mathbf{R}_{xx}^{-1}\mathbf{P}_{dx} \tag{3.52}$$

where $\mathbf{R}_{xx}^{-1}$ is the inverse auto-correlation matrix of $\mathbf{x}$:

$$\mathbf{R}_{xx} = E[\mathbf{x}\mathbf{x}^H] \tag{3.53}$$

and $\mathbf{P}_{dx}$ is the cross-correlation vector of $\mathbf{x}$ and d:

$$\mathbf{P}_{dx} = E[\mathbf{x}d^*] \tag{3.54}$$

where d is the desired data bit.

To prove that the ADC is equivalent to the structure in Figure 3.5, we need to show that ζ is equivalent to z_i, the output of the ADC when multipath, MAI, and a jammer is present in the signal. The output z_i is given as:

$$\begin{aligned} z_i &= \mathbf{w}^H\mathbf{y} \\ &= (\mathbf{c}_i^1)\mathbf{R}_{yy}^{-1}\mathbf{y} \end{aligned} \tag{3.55}$$

where $\mathbf{R}_{yy}$ is the auto-correlation of $\mathbf{y}$ and $\mathbf{c}_i^1$ is the PN code sequence with multipath effects for the current bit interval i. The auto-correlation matrix of $\mathbf{y}$ is:

$$\begin{aligned} \mathbf{R}_{yy} &= E[\mathbf{y}\mathbf{y}^H] \\ &= \mathbf{c}_i^1(\mathbf{c}_i^1)^H + \mathbf{c}_{i-1}^1(\mathbf{c}_{i-1}^1)^H + \mathbf{c}_i^2(\mathbf{c}_i^2)^H + \mathbf{c}_{i-1}^2(\mathbf{c}_{i-1}^2)^H + \mathbf{R}_{jj} + \sigma_n^2\mathbf{I} \end{aligned} \tag{3.56}$$

where the superscripts 1 and 2 denote the first and second users respectively, and $\mathbf{c}_{i-1}$ is the PN code sequence with multipath effects for the previous bit interval $i-1$.

We want to show that:

$$\zeta = \mathbf{P}_{dx}^H\mathbf{R}_{xx}^{-1}\mathbf{y} = (\mathbf{c}_i^1)^H\mathbf{R}_{yy}^{-1}\mathbf{y}. \tag{3.57}$$

We proceed by showing that the vector $\mathbf{x}$ is equal to:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \mathbf{c}_i^1\mathbf{y} \\ \mathbf{c}_i^2\mathbf{y} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{c}_i^1 \\ \mathbf{c}_i^2 \end{bmatrix}\mathbf{y}. \end{aligned} \tag{3.58}$$

Therefore, the vector $\mathbf{P}_{dx}$ is equal to:

$$\begin{aligned} \mathbf{P}_{dx} &= E[\mathrm{x}d_1^*] \\ &= \begin{bmatrix} (\mathrm{c}_i^1)^H \mathrm{c}_i^1 \\ (\mathrm{c}_i^2)^H \mathrm{c}_i^1 \end{bmatrix} \end{aligned} \tag{3.59}$$

and its transpose can be written as:

$$\mathbf{P}_{dx}^H = (\mathrm{c}_i^1)^H \begin{bmatrix} \mathrm{c}_i^1 & \mathrm{c}_i^2 \end{bmatrix}. \tag{3.60}$$

Equation 3.51 can now be written as:

$$\zeta = (\mathrm{c}_i^1)^H \begin{bmatrix} \mathrm{c}_i^1 & \mathrm{c}_i^2 \end{bmatrix} \mathbf{R}_{xx}^{-1} \begin{bmatrix} (\mathrm{c}_i^1)^H \\ (\mathrm{c}_i^2)^H \end{bmatrix} \mathrm{y}. \tag{3.61}$$

To show that equation 3.57 is true we need to show that:

$$\mathbf{R}_{yy}^{-1} = \begin{bmatrix} \mathrm{c}_i^1 & \mathrm{c}_i^2 \end{bmatrix} \mathbf{R}_{xx}^{-1} \begin{bmatrix} (\mathrm{c}_i^1)^H \\ (\mathrm{c}_i^2)^H \end{bmatrix}. \tag{3.62}$$

To do this, we show that the auto-correlation matrix $\mathbf{R}_{xx}$ can be written as:

$$\begin{aligned} \mathbf{R}_{xx} &= E\left[\begin{bmatrix} \mathrm{c}_i^1 \\ \mathrm{c}_i^1 \end{bmatrix} \mathrm{y}\mathrm{y}^H \begin{bmatrix} (\mathrm{c}_i^1)^H \\ (\mathrm{c}_i^2)^H \end{bmatrix} \right] \\ &= \begin{bmatrix} \mathrm{c}_i^1 \\ \mathrm{c}_i^1 \end{bmatrix} \mathbf{R}_{yy} \begin{bmatrix} (\mathrm{c}_i^1)^H \\ (\mathrm{c}_i^2)^H \end{bmatrix} \\ &= \mathbf{C}^H \mathbf{R}_{yy} \mathbf{C}. \end{aligned} \tag{3.63}$$

With this, we may write equation 3.62 as:

$$\mathbf{R}_{yy}^{-1} = \mathbf{C}\mathbf{R}_{xx}^{-1}\mathbf{C}^H. \tag{3.64}$$

We show that this equation is equivalent to equation 3.63 as follows:

$$\mathbf{I} = \mathbf{R}_{yy}^{-1}\mathbf{R}_{yy} = \mathbf{C}\mathbf{R}_{xx}^{-1}\mathbf{C}^H\mathbf{R}_{yy} \tag{3.65}$$

$$\mathbf{C} = \mathbf{C}\mathbf{R}_{xx}^{-1} \underbrace{\mathbf{C}^H\mathbf{R}_{yy}\mathbf{C}}_{\mathbf{R}_{xx}}, \tag{3.66}$$

and thus equation 3.62 and equation 3.57 are shown to be true. This concludes the proof which shows that the ADC is equivalent to separate FIR filters designed to remove separate interferences.

CHAPTER 4

SIMULATION RESULTS

In this chapter, the results of the various simulations utilizing the ADC for suppression of narrow-band interferences, co-channel interference and multipath effects in a spread spectrum signal are given. The simulation model for the BPSK DS spread spectrum used in the simulations is explained. The LMS and RLS adaptive algorithms are analyzed and their speed of convergence learning curves plotted. The frequency response of the ADC is calculated and shown to act as a notch filter when the signal is corrupted by a narrow-band jammer. It is also shown that the ADC estimates the channel frequency response to suppress multipath interference. In an effort to gauge the ADC's performance in interference and multipath suppression, probability of error curves are generated and compared with the conventional matched filter and the RAKE receiver. Probability of error curves are also plotted to compare the conventional multi-user and decorrelating detector with the ADC in MAI suppression.

4.1 Simulation Model

The simulations explained in this chapter were generated using MATLAB version 4.0a on a SUN SPARC workstation. A PN code sequence of 15 chips was used and sampled at a rate of twice per chip, producing a signal vector with 30 elements per bit. The reason for using this sampling rate as opposed to once per chip is due to the fact that the fading channel is modeled as a tapped-delay line with a tap delay of $\frac{T_c}{2}$, as explained in Chapter 2. Therefore, the ADC is modeled as having 30 tap-delays. The jammer has a signal power of 20dB greater than the transmitted data bit, with

a jammer frequency of 1 rad/sec. The noise vector has a zero mean and a variance of $\sigma_n^2 = 1$.

The multipath channel was modeled after the one used in [6]. It has four energy paths (one direct and three delayed) and is described by the following transfer function:

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} \tag{4.1}$$

where $z = e^{j\omega}$. The coefficient h_k represents the amplitude for the four different paths and z^{-k} represents the phase shift associated with each path. The channel is modeled as having zeros at frequencies 2, 3, and 4 radians and with $H(0) = 1$. The coefficients can then be found as follows:

$$\begin{aligned} H(0) &= h_0 = 1 \\ H(z_0^2) &= 1 + h_1 z_0^{-2} + h_2 z_0^{-4} + h_3 z_0^{-6} = 0 \\ H(z_0^3) &= 1 + h_1 z_0^{-3} + h_2 z_0^{-6} + h_3 z_0^{-9} = 0 \\ H(z_0^4) &= 1 + h_1 z_0^{-4} + h_2 z_0^{-8} + h_3 z_0^{-12} = 0 \end{aligned} \tag{4.2}$$

where $z_0 = e^j$. In vector format the above equation becomes:

$$\begin{bmatrix} 1 & z_0^{-1} & z_0^{-2} & z_0^{-3} \\ 1 & z_0^{-2} & z_0^{-4} & z_0^{-6} \\ 1 & z_0^{-4} & z_0^{-8} & z_0^{-12} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{B}\ \mathbf{h}, \tag{4.3}$$

and is solved for the vector $\mathbf{h}$.

Having calculated the coefficient, the signal vector $\mathbf{c}_i$ for multipath is given in equation 3.27 as the sum of shifted code vectors multiplied by the appropriate coefficient:

$$\begin{aligned} \mathbf{c}_i &= h_0[s_0, s_0, s_1, s_1, \ldots, s_{L-1}, s_{L-1}]^T \\ &\quad + h_1[0, s_0, s_0, s_1, s_1, \ldots, s_{L-2}, s_{L-1}]^T \\ &\quad + h_2[0, 0, s_0, s_0, s_1, s_1, \ldots, s_{L-2}, s_{L-2}]^T \\ &\quad + h_3[0, 0, 0, s_0, s_0, s_1, s_1, \ldots, s_{L-3}, s_{L-2}]^T. \end{aligned} \tag{4.4}$$

The vector $\mathbf{c}_i$ represents the multipath effect on the current data bit. The signal vector for $\mathbf{c}_{i-1}$ in equation 3.27 which corresponds to the multipath left over from the previous data bit is given in a similar fashion as:

$$\begin{aligned} \mathbf{c}_{i-1} &= h_1[s_{L-1}, 0, 0, \ldots, 0]^T \\ &\quad + h_2[s_{L-1}, s_{L-1}, 0, 0, \ldots, 0]^T \\ &\quad + h_3[s_{L-2}, s_{L-1}, s_{L-1}, 0, 0, \ldots,]^T. \end{aligned} \tag{4.5}$$

In order to obtain analytical results, the signal vectors $\mathbf{c}_i$ and $\mathbf{c}_{i-1}$ are normalized to obtain:

$$\mathbf{c}_{N_m} = \frac{\sqrt{L}\mathbf{c}_m}{\sqrt{\mathbf{c}_i^T\mathbf{c}_i + \mathbf{c}_{i-1}^T\mathbf{c}_{i-1}}} \quad m = i, i-1 \tag{4.6}$$

where $\mathbf{c}_{N_m}$ are the normalized vectors for $\mathbf{c}_i$ and $\mathbf{c}_{i-1}$ and L is the number of chips, which is equal to 30 for our case. This normalization is done in order to make sure that the multipath channel does not inadvertently enhance instead of corrupt our signal $\mathbf{y}$. The received power from the transmitted data bit is the sum of the power received in the current bit interval i and the power that spills over into the next bit interval, $i-1$. The normalization procedure sets the received power of the transmitted signal equal to unity.

The narrow-band jammer used in the simulations was modeled after the one used in [6] and [12]. It was produced by taking the output of a Butterworth filter of order 16 fed by white Gaussian noise. The center frequency of the filter's passband is 1 radian, with a total bandwidth of 0.5 radians. The tone jammer is the output of a sine-wave generator with a uniform random phase in the range [0,2π].

4.2 Learning Curves

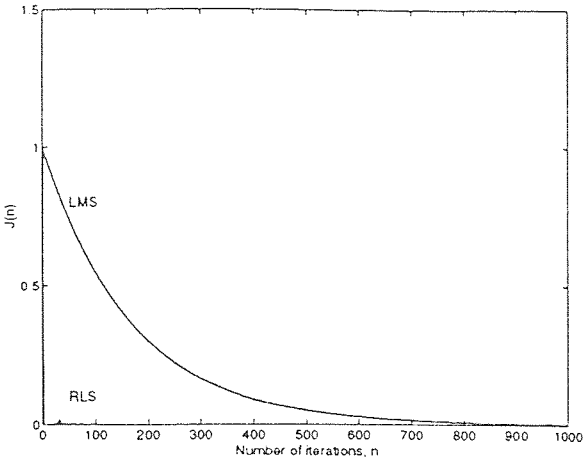
In order to show the speed of convergence of the LMS and RLS algorithms, the error squared from the output of the ADC was plotted against the number of iterations

needed for the algorithm to converge to a minimum squared error. The squared error e_i^2 was computed for each algorithm over 1000 iterations. The experiment was repeated 100 times for both RLS and LMS cases. The MSE was computed by averaging the e_i^2 over the 100 independent experiments. This results in only an approximation of the actual ensemble average of e_i^2, but is nevertheless sufficient for the purpose of this work.

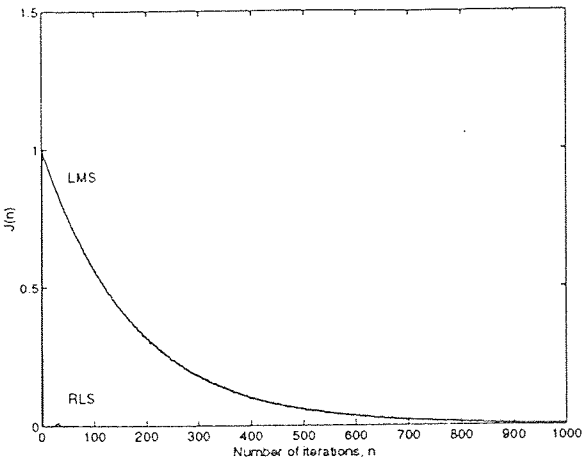
The value for the step-size parameter μ in the LMS algorithm was chosen such that the LMS converges as rapidly as possible but still satisfies the restriction $0 < \mu < \frac{2}{\lambda_{max}}$ for convergence of the algorithm. For the RLS, λ was set equal to unity and δ was set equal to 0.001 to ensure convergence of the algorithm. The reference was supplied by the actual data bit to update the weight vector $\mathbf{w}_{i+1}$ as opposed to the output of the threshold device. The SNR for each simulation was set to 30dB so that the noise would not significantly affect the convergence rate of each algorithm. Different scenarios for a number of combinations of interferences corrupting the transmitted signal were tested. The resulting plots are shown in Figure 4.1.

Plot 1 of Figure 4.1 shows the learning curve for both the RLS and LMS algorithms when no interference is present in the received signal. Plots 2, 3 and 4 show the learning curves for the RLS and LMS with multipath interference, and multipath with a tone jammer with an SIR of -10dB and -20dB respectively. Learning curves for MAI with signal-to-signal ratios (SSR) of -5dB, 0dB, 5dB (SNR2-SNR1) were also computed and are shown in Plots 5, 6 and 7, respectively. The other plots in Figure 4.1 show the learning curves for combinations of multipath interference, MAI, and tone and narrow-band jammers. Convergence of the algorithm is assumed to occur when the learning curve begins to level out at a constant MSE value.

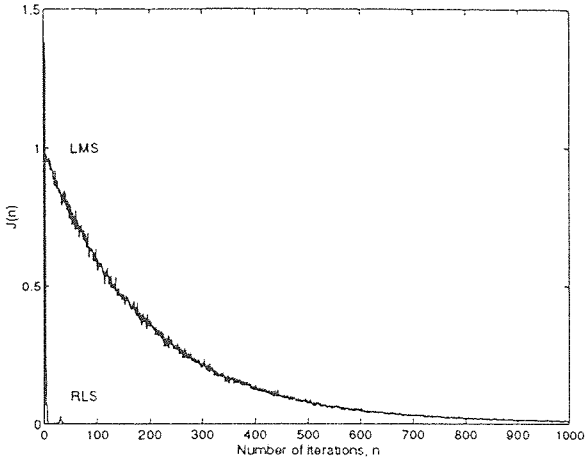
In all cases, the RLS was shown to converge to the optimum weight vector, $\mathbf{w}_{opt}$, faster than the LMS, taking less than 25 iterations in most cases. The LMS,



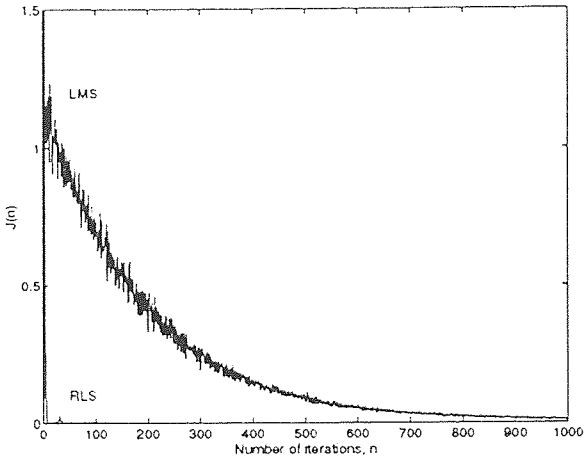
Plot 1: No interference.



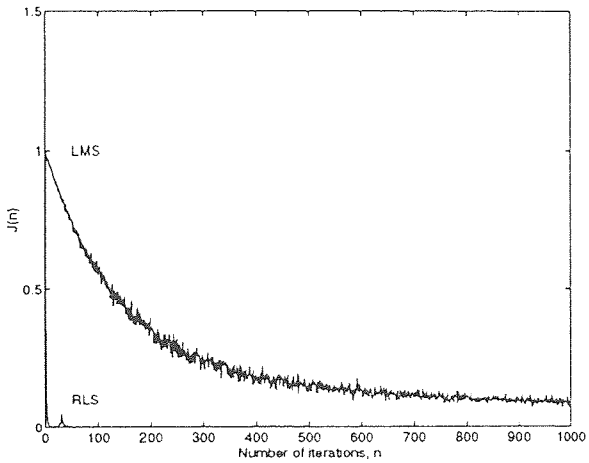
Plot 2: Multipath.



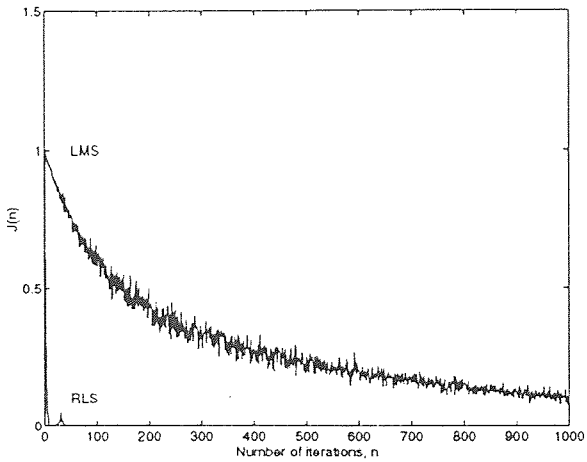
Plot 3: Tone jammer(SIR=-10dB) with multipath.



Plot 4: Tone jammer(SIR=-20dB) with multipath.

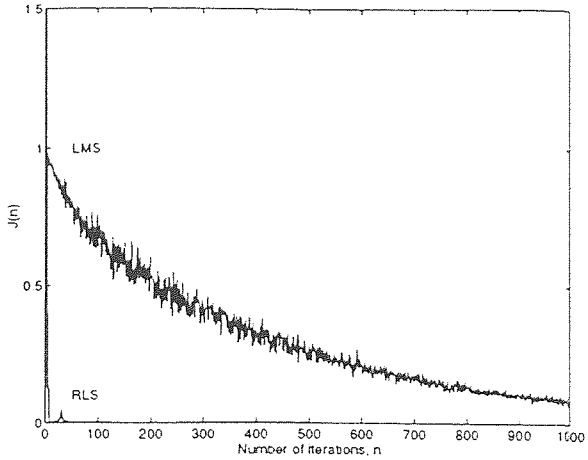


Plot 5: MAI(SSR=-5dB).

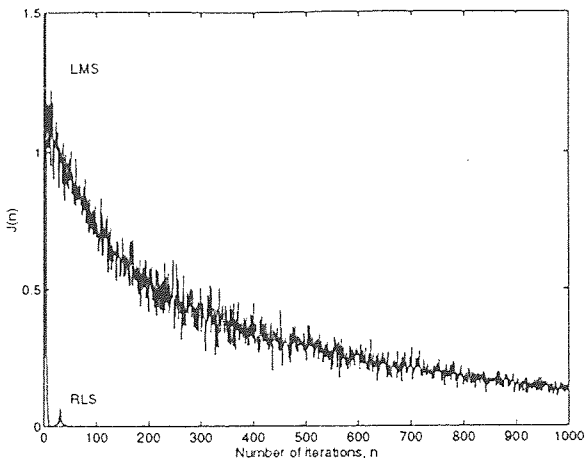


Plot 6: MAI(SSR=0dB).

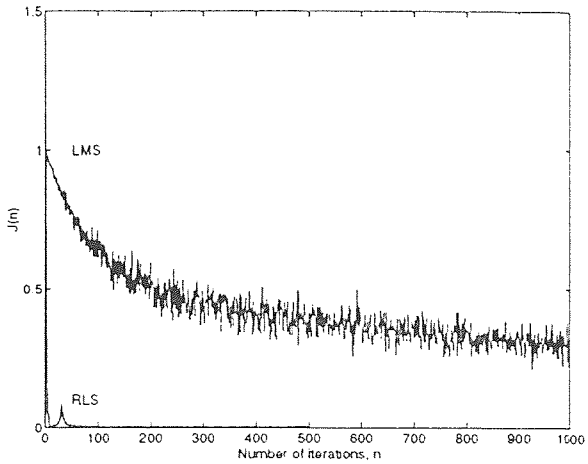
Figure 4.1 Learning Curves for RLS and LMS algorithms. SNR=30dB. Curves are averages of 100 separate runs. For the LMS, the step-size parameter $\mu = 0.0001$. For the RLS, $\lambda = 1$ and $\delta = 0.0001$.



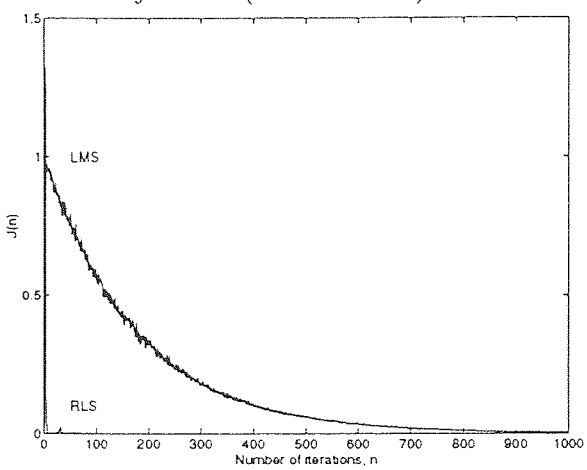
Plot 7: MAI(SSR=5dB).



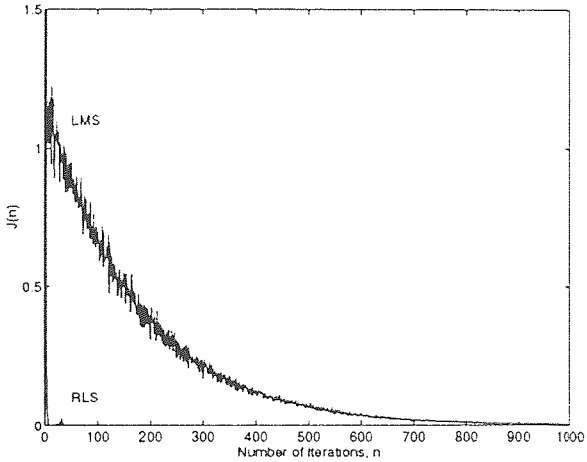
Plot 8: MAI(SSR=0dB) with tone jammer(SIR=-20dB).



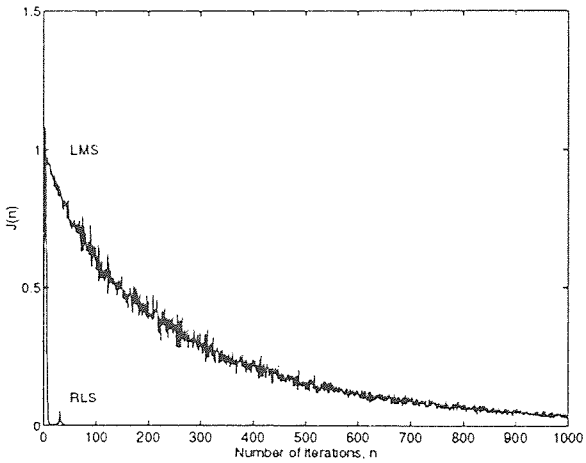
Plot 9: MAI(SSR=0dB) with multipath.



Plot 10: Tone jammer(SIR=-10dB).

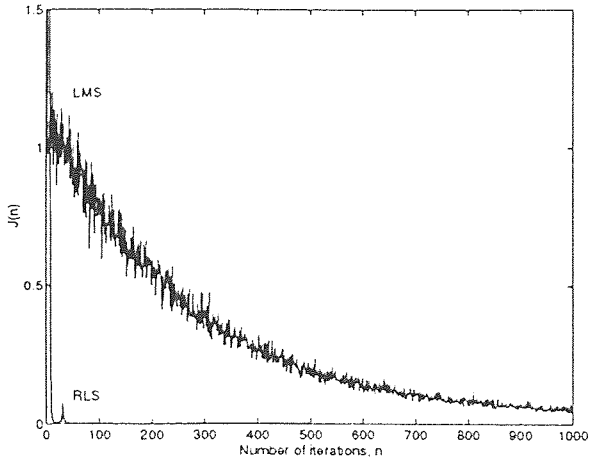


Plot 11: Tone jammer(SIR=-20dB).

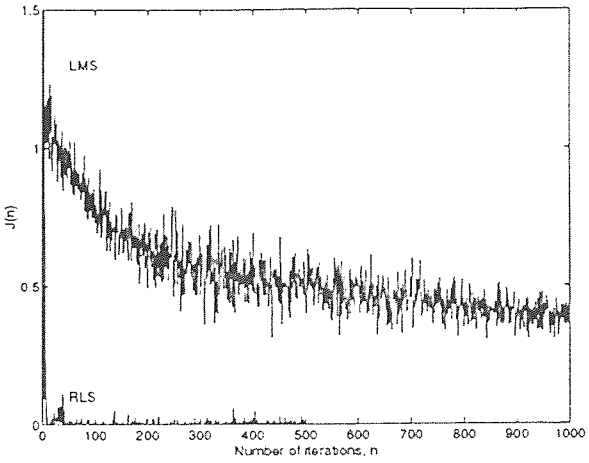


Plot 12: Narrow-band jammer (SIR=-10db).

Figure 4.1 cont. Learning curves for RLS and LMS algorithms. SNR=30dB. Curves are averages of 100 separate runs. For the LMS, the step-size parameter $\mu = 0.0001$. For the RLS, $\lambda = 1$ and $\delta = 0.0001$.



Plot 13: Narrow-band jammer (SIR = -20dB).



Plot 14: Tone jammer (SIR = -20dB) with MAI (SSR = 0dB) and multipath

Figure 4.1 cont. Learning curves for RLS and LMS algorithms. SNR = 30 dB. Curves are averages of 100 separate runs. For the LMS, the step-size parameter $\mu = 0.0001$. For the RLS, $\lambda = 1$ and $\delta = 0.0001$.

on the other hand, did not converge until 400 or more iterations, and in some cases (such as the case with multipath, jammer, and MAI in Plot 14) it converged after 1000 iterations.

The reason the RLS converged so much faster than the LMS in the simulations was because of the low SNR value used. If, however, a larger SNR was used, both the RLS and LMS converge at about the same rate [3]. This is because the steady-state value of the ensemble-averaged squared error produced from the RLS algorithm is much smaller than that produced from the LMS algorithm with either a high or low SNR. Unlike the LMS algorithm, the RLS algorithm is insensitive to the eigenvalue spread of the correlation matrix $\mathbf{R}_{yy}$.

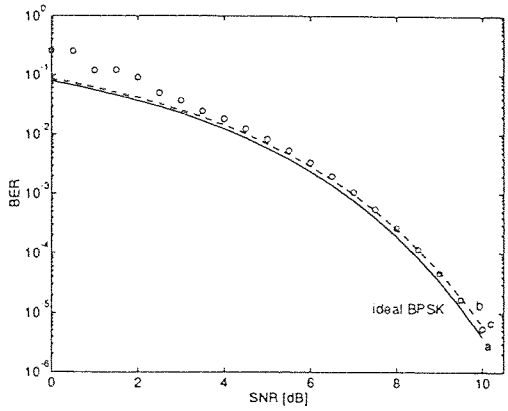
It is also observed that the LMS produced a large *misadjustment* in Plots 5,6,8,9 and 14, while the RLS converged to an MSE of zero every time. This difference is caused by the adaptive mechanism used to control the weight vector with the LMS as opposed to a deterministic approach used with the Wiener-Hopf equation.

4.3 Probability of Error Curves

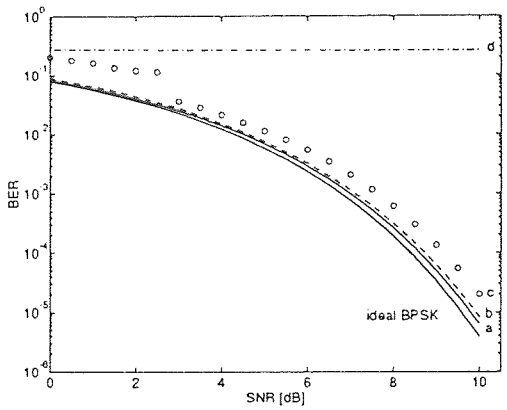
In order to gauge the performance of the RLS and LMS algorithms as applied to the ADC, the BER was calculated versus the SNR. For these simulations, both the RLS and LMS algorithms were run for 1000 iterations. For the first 500 iterations the actual transmitted bit was used to calculate the error e_i. After 500 iterations, the ADC was run in decision-directed mode, using the output of the threshold device as the reference to calculate e_i and update the weight vector $\mathbf{w}_i$. The value of μ for the LMS was chosen to allow rapid convergence of the LMS and to satisfy the restriction $0 < \mu < \frac{2}{\lambda_{\max}}$. The RLS algorithm was run with $\lambda = 0.9999$ and $\delta = 0.1$ in order to allow it to converge. The probability of error curves were calculated using the probability of error equations in Chapter 3. The ADC was run utilizing the LMS and RLS for 1000 iterations. After the algorithms converged, the weight vector was used to calculate the probability of error. It was assumed that 1000 iterations was enough to let each algorithm converge to a weight vector close to optimum. Different scenarios of different combinations of interferences were run. For the jammer case, an SIR of -20dB was used for both tone and narrow-band jammers. The multipath model used was that described in Section 4.1. For the MAI case, an SSR of 0dB was used throughout. These curves are plotted in Figure 4.2 through Figure 4.6.

In addition to these curves, the theoretical value of the weight vector $\mathbf{w}_i$ using the Wiener-Hopf equations, was also plotted in order to compare how well each algorithm converged. A curve for an ideal BPSK system is also included in each plot to compare each scenario to an ideal case. Observing the plots in the figure, we see that when the ADC was run using the RLS algorithm, the probability of error curve tracks the theoretical curve closely, signifying that the RLS converged within the 1000 iterations. However, observing the curves for the LMS case, we see that they do not track the theoretical curves as closely. In fact, when a jammer, or a combination of a jammer with multipath and MAI is present in the received

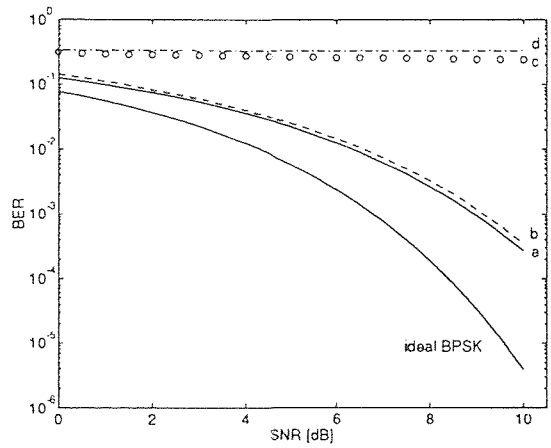
signal y, the probability of error curve for the LMS is considerably higher than the theoretical curve (see Plot 3 of Figure 4.4, Plot 2 of Figure 4.5 and both plots of Figure 4.6). The high irregularities of the LMS curve for high SNR, particularly the plots of Figure 4.4 and Plot 1 of Figure 4.5, are due to the fact that the LMS did not converge within the allotted 1000 iterations. The LMS diverged and produced error propagations.



Plot 1: No interference.

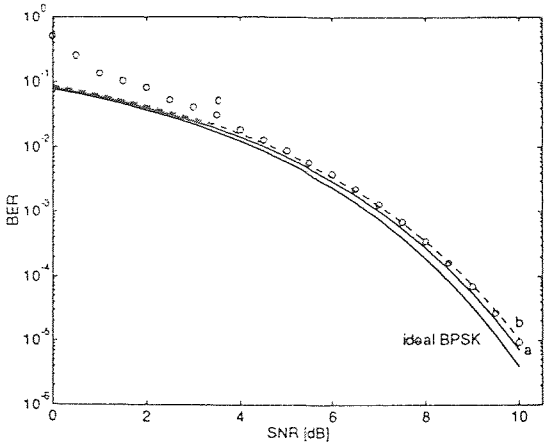


Plot 2: Tone Jammer(SIR=-20dB).

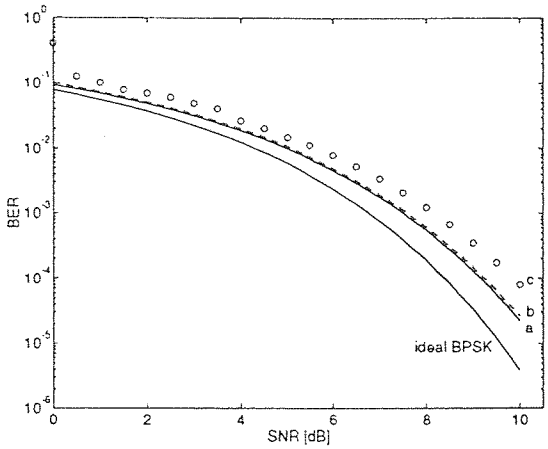


Plot 3: Narrow-band jammer(SIR=-20dB).

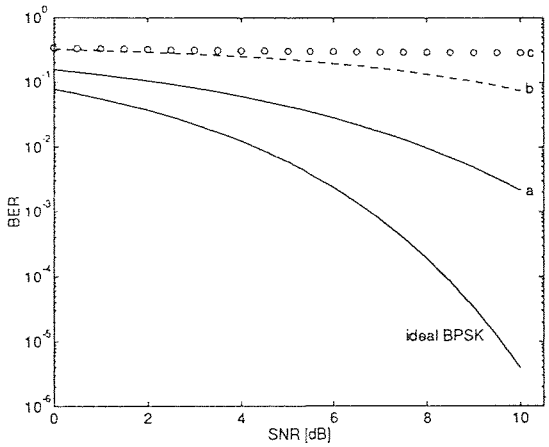
Figure 4.2 Probability of error vs . SNR for the ADC with a jammer
[a]Theoretical curve using Wiener-Hopf [b]ADC utilizing RLS
[c]ADC utilizing LMS [d]conventional matched-filter.



Plot 1: Multipath.

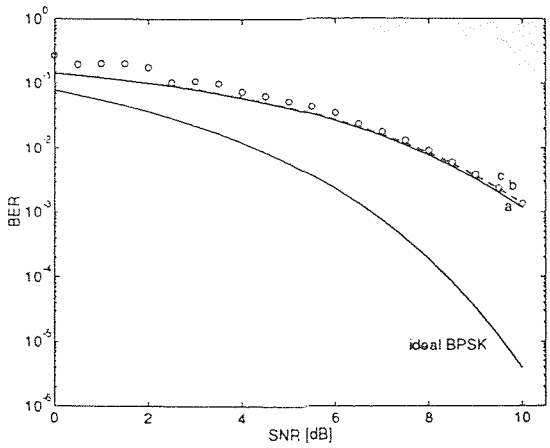


Plot 2: Multipath with tone jammer.

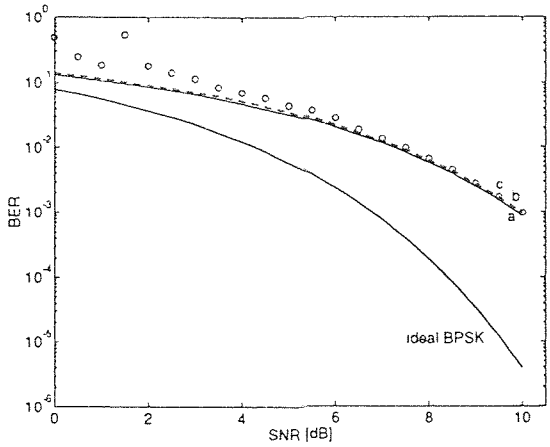


Plot 3: Multipath with narrow-band jammer.

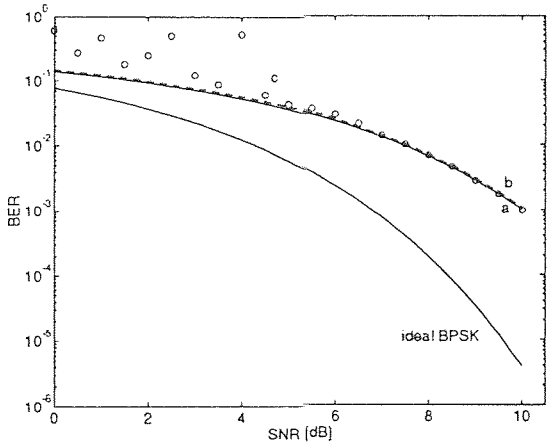
Figure 4.3 Probability of error vs . SNR for the ADC with multipath and jammer. SIR of jammer is -20dB.
[a]Theoretical curve using Wiener-Hopf [b]ADC utilizing RLS
[c]ADC utilizing LMS.



Plot 1: MAI(SSR=-5dB).

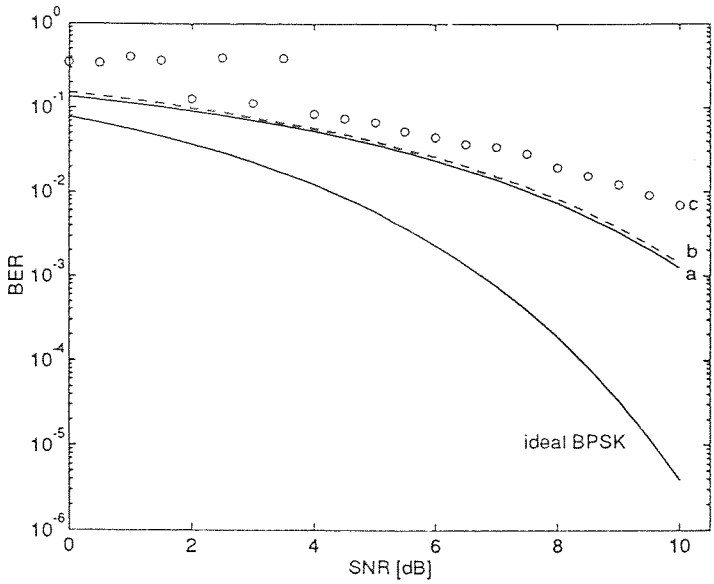


Plot 2: MAI(SSR=0dB).

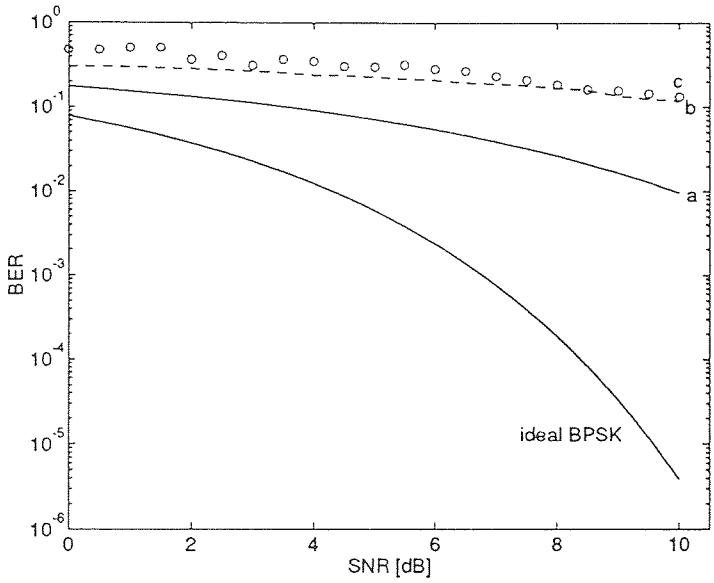


Plot 3: MAI(SSR=5dB).

Figure 4.4 Probability of error vs . SNR for the ADC with MAI.
[a]Theoretical curve using Wiener-Hopf [b]ADC utilizing RLS
[c]ADC utilizing LMS.

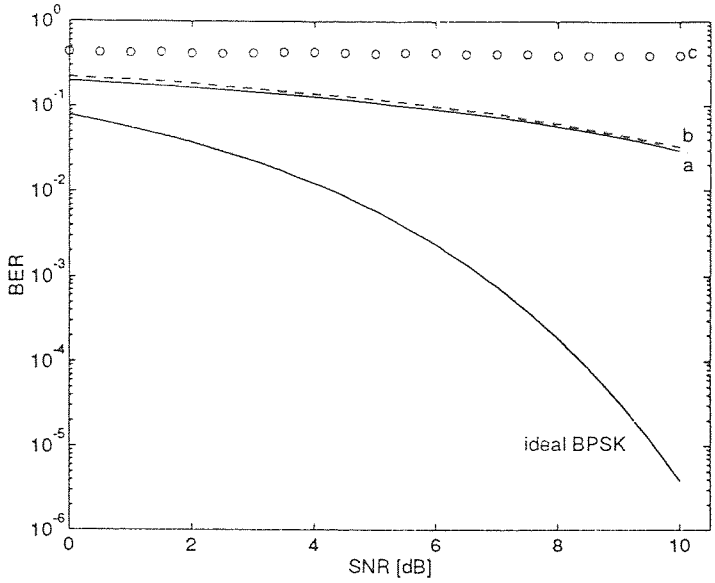


Plot 1: MAI with tone jammer.

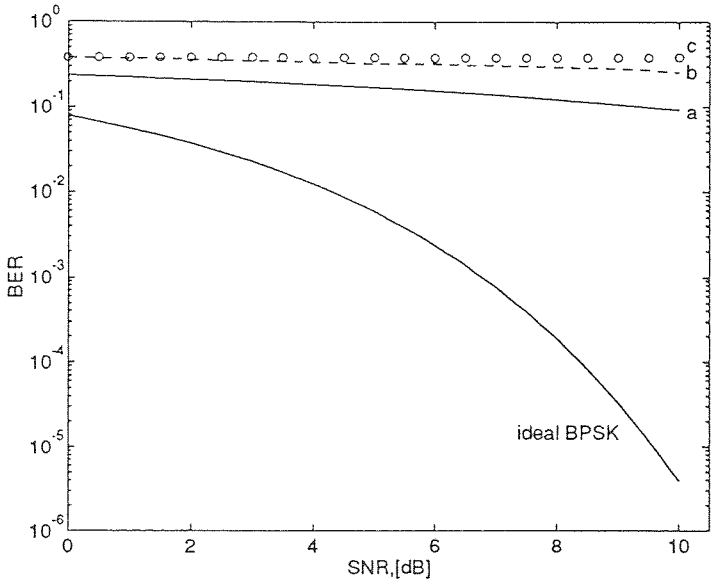


Plot 2: MAI with narrow-band jammer.

Figure 4.5 Probability of error vs . SNR for the ADC with MAI and jammer. SIR of jammer is -20dB. SSR=0dB.
[a]Theoretical curve using Wiener-Hopf [b]ADC utilizing RLS
[c]ADC utilizing LMS.



Plot 1: Multipath with MAI and tone jammer.



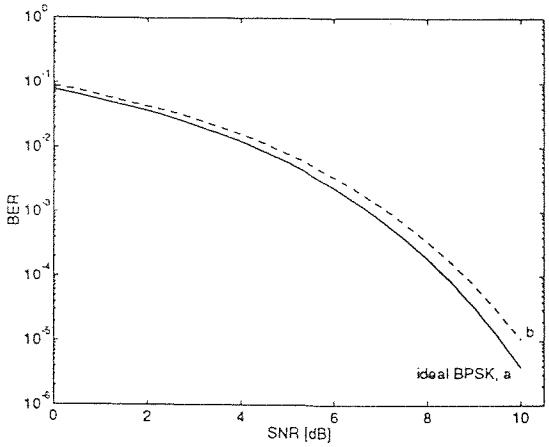
Plot 2: Multipath with MAI and narrow-band jammer.

Figure 4.6 Probability of error vs . SNR for the ADC with multipath, MAI and jammer. SIR of jammer is -20dB. SSR=0dB.
[a]Theoretical curve using Wiener-Hopf [b]ADC utilizing RLS
[c]ADC utilizing LMS.

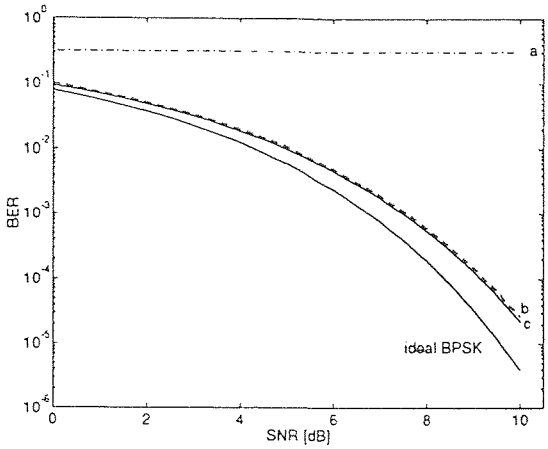
4.4 The ADC Compared With a RAKE Receiver

Probability of error curves similar to those produced in the previous section were produced to compare the ADC receiver with the RAKE receiver when the received signal is corrupted by multipath and narrow-band interferences. The plots in Figure 4.7 compare the RAKE with the ADC, and were run utilizing the RLS algorithm. The RLS was run in non-decision-directed mode for the first 500 iterations, then switched to decision-directed mode from 500 to 1000 iterations, in which case it was assumed that the algorithm had converged to an optimum $\mathbf{w}_i$. Equation 3.34 was used to calculate the BER utilizing the weight vector $\mathbf{w}_i$ from the ADC and the channel model of Section 4.1. The BER curve for the RAKE was calculated using equation 2.66. Plot 1 of Figure 4.7 shows that the RAKE receiver gives an optimum solution when the channel model is known. Hence, it is ideal BPSK. The curve for the ADC utilizing the RLS is slightly higher, since it will converge to that given by equation 3.34.

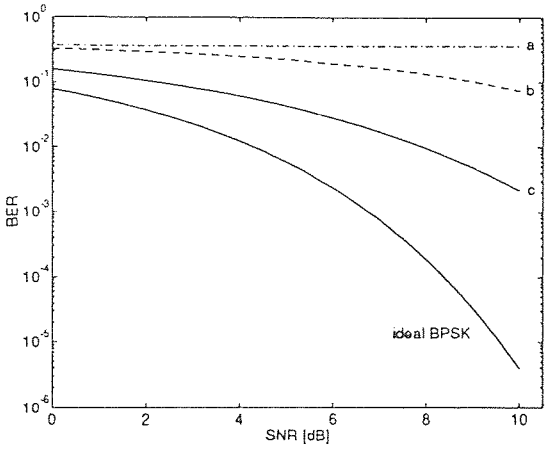
When a jammer is also present in the received signal, the situation changes. Since the RAKE receiver has knowledge of only the channel and not the jammer statistics, its performance will deteriorate enormously, as evidenced in Plots 2 and 3 of Figure 4.7. These plots show the performance of the RAKE and ADC in a multipath environment with a tone jammer and a narrow-band jammer, respectively. The ADC, on the other hand, tries to minimize the MMSE at the output of the filter and hence successfully removes the jammer **and** estimates the channel response in order to recover the transmitted bit, thus producing a lower BER, as shown in Plots 2 and 3.



Plot 1: Multipath only.



Plot 2: Multipath with tone jamme



Plot 3: Multipath with narrow-band jammer.

Figure 4.7 Probability of error vs . SNR to compare ADC and RAKE
SIR for jammer is -20dB.
[a]RAKE receiver [b]ADC utilizing RLS
[c]Theoretical curve using Wiener-Hopf.

4.5 The ADC Compared With a Decorrelating Detector

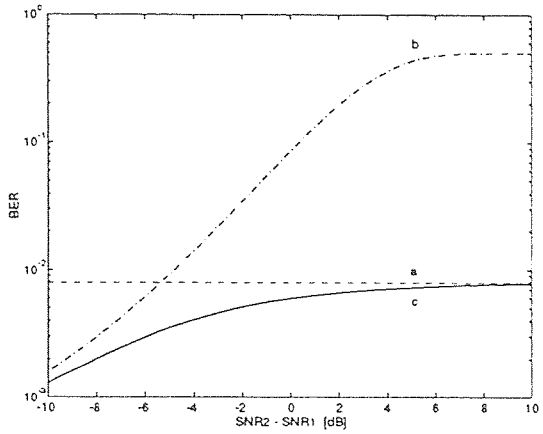
To see how well the ADC performs when used in a multiple user environment, it was compared with a conventional conventional multi-user detector and a decorrelating detector. The BER was calculated versus SSR of -10dB to 10dB for a simple two-user case. The cross-correlation between the two codes used, $\mathbf{s}_1^T\mathbf{s}_2$, was set equal to 22, which represents a high bandwidth case. The SNR of user 1 was fixed at 8dB for all the simulations and the SNR of user 2 relative to user 1 varied from -10dB to 10dB. The RLS algorithm was run on the ADC for 1000 iterations with decision-directed mode for the final 500 iterations. The probability of error for the ADC was calculated using equations 3.34 and 3.44 with the signal $\sqrt{a}\,d_2\mathbf{s}_2$ having a variable energy level and utilizing the final weight vector value from the RLS simulations.

Figure 4.8 shows the results of these simulations. As can be seen in Plot 1, the value of the BER for the decorrelating detector is constant. This is because the interfering signal is completely decorrelated at the output of the decorrelator, thus bearing no effect on the BER. Only the variance of the noise, or any other Gaussian interference, effects the BER. The BER curve for the ADC begins at a low BER for low SSR and increases logarithmically toward the decorrelator's curve for increasing values of SSR. This is because the decorrelator represents the maximum limit achievable when the only interference present is another user. The curve for a conventional multi-user detector is also plotted, and as expected, increases drastically as the interfering user's power level increases in this high bandwidth case.

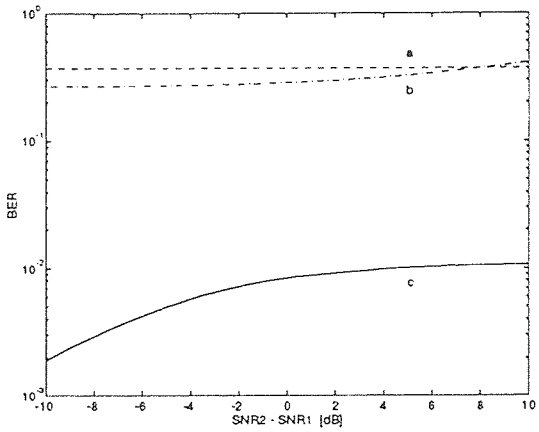
When additional interference from a jammer is present in the received signal, the decorrelating detector does not decorrelate the jammer signal, as evidenced in Plots 2 and 3 in Figure 4.8, for both a tone and narrow-band jammer. These plots show that the BER for both the decorrelator and conventional multi-user detector increases drastically. The BER curve for the ADC, however, only increase slightly.

This is because the weight vector for the ADC converges to a value which minimizes the error at the output such that the MAI is removed along with the jammer.

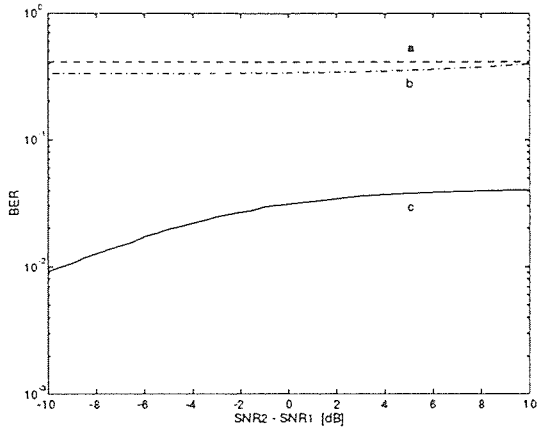
One limitation of the ADC concerning MAI removal is that it does not address the near-far user problem. The BER increases when the interfering user's power is higher than that of the desired user. This is shown in the increasing BER curves for the ADC with increasing SSR.



Plot 1: No interference.



Plot 2: Tone Jammer.



Plot 3: Narrow-band jammer.

Figure 4.8 Probability of error vs . SSR to compare ADC and decorrelating detector. SIR for the jammer is -20dB, SNR=8dB and $\rho = 0.7333$. [a]decorrelating detector [b]conventional multi-user detector [c]ADC utilizing RLS.

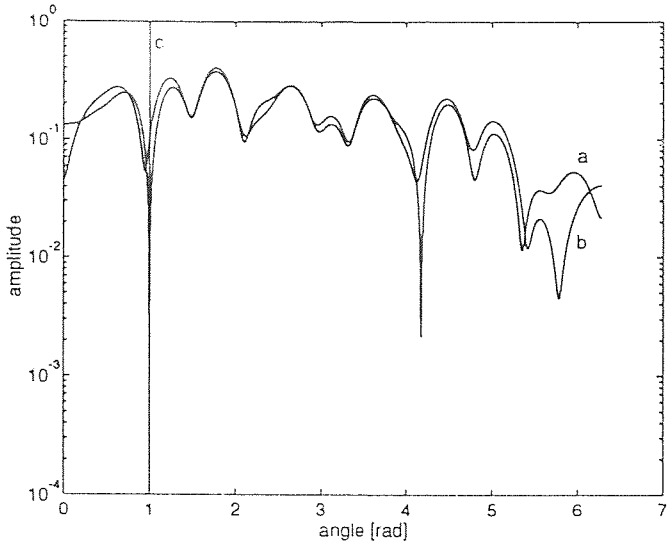
4.6 Frequency Response Curves

Frequency response curves are plotted to show the notch-filtering effect of the ADC for jammer suppression and channel estimation for multipath suppression. Both the tone and narrow-band jammers had a center frequency of 1 radian, with the narrow-band jammer having a bandwidth of 0.5 radians. The multipath model used was described in Section 4.1.

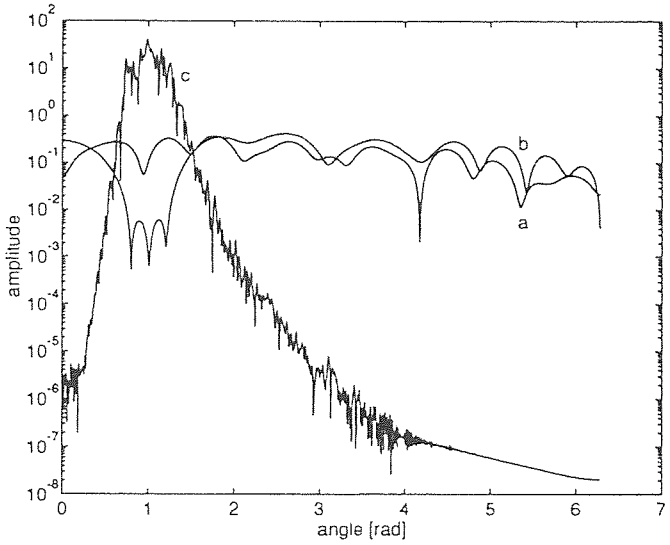
Figures 4.9 and 4.10 show the plots for power spectral density produced from the simulations. Plots 1 and 2 of Figure 4.9 show the spectrum for the weight vector of the ADC when a tone and narrow-band jammer are in the received signal. For comparison, the spectrum of the PN code sequence used in the conventional spread spectrum matched filter receiver was also plotted. In both of these spectrum plots we see that the ADC successfully notched-out the jammer at the proper frequency.

To show how the ADC removes multipath interference in order to demodulate the spread spectrum signal, the frequency response curve for the ADC with multipath was plotted. This curve is shown in the plot of Figure 4.10 along with the channel response curve. We see that the response of the ADC is nearly identical to the channels response. This is the required response to remove the multipath, just as a BPSK receiver would be matched to the transmitted carrier signal.

The difference between the ADC and an adaptive equalizer is that an adaptive equalizer tries to maintain a flat frequency response curve in a non-stationary environment by compensating for any perturbations in the channel, increasing the gain at drops in the channel frequency response and decreasing gain at peaks in the channel frequency response, achieving a white output. The ADC is not a true equalizer in the sense that it tries to account for the perturbations in the channel. Instead, the ADC performs channel equalization. The tap weights of the ADC adaptively converge to the response of the channel so as to demodulate the spread spectrum signal and retrieve the desired bit.

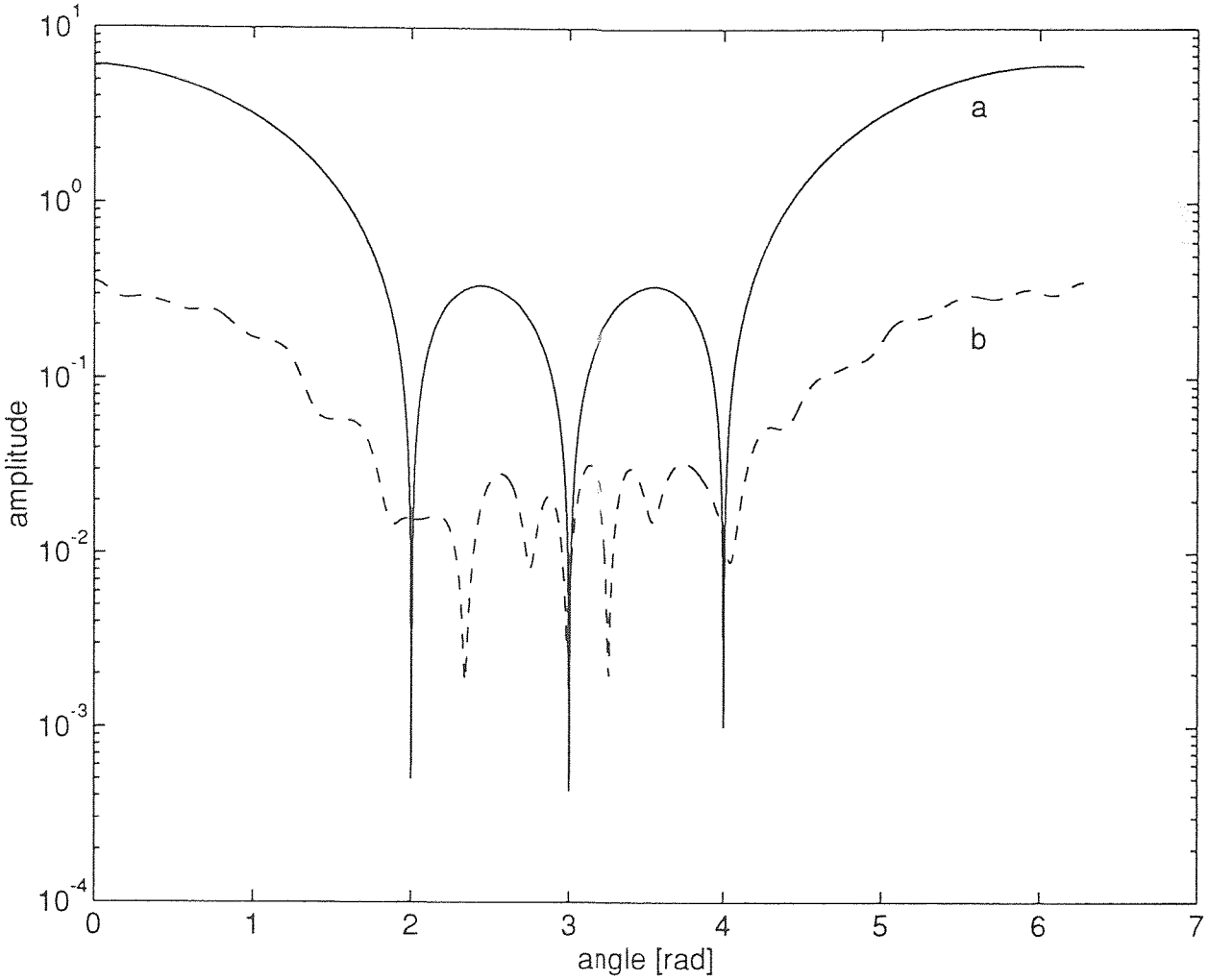


Plot 1: Frequency response of weight vector with a 20dB tone jammer.



Plot 2: Frequency response of weight vector with a 20dB narrow-band jammer.

Figure 4.9 Power spectral density plots with jammer. SNR=10dB.
[a] PN code sequence spectrum [b] ADC's spectrum
[c] jammer



Plot 1: Frequency response of channel and weight vector with multipath.

Figure 4.10 Frequency response plot with multipath. SNR=10dB.
[a] Channel frequency response [b] ADC's frequency response.

CHAPTER 5

CONCLUSIONS

A versatile filter capable of suppressing numerous types of interferences from a spread spectrum signal was presented in this work. The ADC receiver works on the basis of minimizing the MSE between the received signal and the desired data bit in order to jointly suppress interferences and demodulate the signal to recover the transmitted data bit. The types of interferences considered here were a narrow-band jammer, multipath, and MAI. The following conclusions were made from the simulations:

- The MMSE of the output of the ADC is achieved much faster utilizing the RLS algorithm as opposed to the LMS algorithm.
- The ADC is comparable to a RAKE receiver when the received signal is corrupted by multipath. The ADC estimates the channel response in order to recover the transmitted data bit.
- The ADC outperforms the RAKE receiver when the received signal is corrupted by both multipath and a narrow-band interference.
- The ADC is comparable to a decorrelating detector in removing MAI in the received signal and outperforms the conventional multi-user detector when the user's PN codes are not orthogonal.
- The ADC outperforms the decorrelating detector when, in addition to MAI, the received signal is corrupted by a narrow-band interference.

The ADC was shown to be a structure which is a convolution of a series of FIR filters which separately remove narrow-band interference, multipath, and MAI, in addition to demodulating the spread spectrum signal. Possible future work may focus on faster converging algorithms and decision-feedback filters.

APPENDIX A

PROBABILITY OF ERROR EQUATIONS

The probability of error equations which were used in the simulations for the ADC which were not derived in the previous chapters will be listed here. These equations are for various interference cases that were not addressed previously. The derivation of each equation is the probability that the output of the threshold device for the i-th bit produces an incorrect estimate $\hat{d}_i$ of the actual data bit d_i. The the probability of error is, given as:

$$P_e = \text{Prob}[z_i < 0 | d_i = 1] = \text{Prob}[z_i > 0 | d_i = -1] \tag{A.1}$$

The probability of error is derived using Bayes' Theorem given the fact that the output z_i is the vector dot-product of the ADC's weight vector $\mathbf{w}_i$ with the received signal vector $\mathbf{y}_i$.

Case: Multipath only

The received signal vector is:

$$\mathbf{y}_i = \sqrt{a}\, d^l \mathbf{c}_i + \sqrt{a}\, d^r \mathbf{c}_{i-1} + \mathbf{n} \tag{A.2}$$

where $\mathbf{c}_i$ is the PN code sequence with multipath effects for the current bit d^l and $\mathbf{c}_{i-1}$ is the PN code with multipath effects for the previous bit d^r. The signal a is the energy of the data bit d, which is constant for all bits. The data bits d^l and d^r are discrete random variables equal to $+1$ or -1 with equal probability. The signal $\mathbf{n}$ is a AWGN vector with zero mean. The probability of error equation is given as:

$$\begin{aligned} P_e &= \frac{1}{2} Q\left(\frac{\sqrt{a}\, \mathbf{w}_i^H \mathbf{c}_i + \sqrt{a}\, \mathbf{w}_i^H \mathbf{c}_{i-1}}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\ &\quad + \frac{1}{2} Q\left(\frac{\sqrt{a}\, \mathbf{w}_i^H \mathbf{c}_i - \sqrt{a}\, \mathbf{w}_i^H \mathbf{c}_{i-1}}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right). \end{aligned} \tag{A.3}$$

Case: Multipath and MAI

The received signal vector is:

$$\mathbf{y}_i = \sum_{k=1}^{K} \sqrt{a_k}\, d_k^l \mathbf{c}_i^k + \sqrt{a_k}\, d_k^r \mathbf{c}_{i-1}^k + \mathbf{n} \tag{A.4}$$

where k represents a separate user and a_k is the signal energy of the k-th user. The probability of error equation for two users is given as:

$$\begin{aligned}
P_e = \; & \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 + \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 + \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 + \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 + \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 - \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 - \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 - \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 - \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right)
\end{aligned} \tag{A.5}$$

where a_1 is the energy of user 1 and a_2 is the energy of user 2. The vector $\mathbf{c}_i^1$ is the PN code sequence vector with multipath effects of the first or desired user in the current bit interval, while $\mathbf{c}_{i-1}^1$ is the PN code sequence with multipath effects for the first user in the previous bit interval. Similarly, the vector $\mathbf{c}_i^2$ is the PN code sequence with multipath effects for the second or interfering user in the current bit

interval, while the vector $\mathbf{c}_{i-1}^2$ is the PN code sequence with multipath effects of the second user in the previous bit interval.

Case: Multipath, MAI with jammer

The received signal vector is:

$$\mathbf{y}_i = \sum_{k=1}^{K} \sqrt{a_k}\, d_k^l \mathbf{c}_i^k + \sqrt{a_k}\, d_k^r \mathbf{c}_{i-1}^k + \mathbf{j} + \mathbf{n} \tag{A.6}$$

where $\mathbf{j}$ is the jammer signal vector. The probability of error equation for two users is given as:

$$\begin{aligned}
P_e \;=\; & \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 + \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 + \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 + \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 + \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 - \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 - \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 - \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 + \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right) \\
& + \frac{1}{8} Q\left(\frac{\sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_i^1 - \sqrt{a_1}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^1 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_i^2 - \sqrt{a_2}\,\mathbf{w}_i^H \mathbf{c}_{i-1}^2}{\sqrt{\mathbf{w}_i^H \mathbf{R}_{jj} \mathbf{w}_i + \sigma_n^2 \mathbf{w}_i^H \mathbf{w}_i}} \right)
\end{aligned} \tag{A.7}$$

where $\mathbf{R}_{jj}$ is the auto-correlation of the jammer $\mathbf{j}$ given as equation 3.3 for a tone jammer and equation 3.4 for a narrow-band jammer.

REFERENCES

1. R. Dixon, *Spread Spectrum Systems*, J. Wiley, New York, second ed., 1984.

2. A. Duel-Hallen, "Decorrelating decision-feedback multiuser detector for synchronous code division multiple access channel," *IEEE Trans. Communications*, vol. 41, no. 2, pp. 285–290, February 1993.

3. S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, New Jersey, 2nd ed., 1991.

4. S. Haykin, *Communication Systems*, John Wiley and Sons, New York, second ed., 1983.

5. F. Hsu and A. Giordano, "Digital whitening techniques for improving spread spectrum communications performance in the presence of narrow-band jamming and interference," *IEEE Trans. Communications*, vol. 26, no. 2, pp. 209–216, February 1978.

6. R. Iltis, "A GLRT-based spread spectrum receiver for joint channel estimaton and interference suppression," *IEEE Trans. Communications*, vol. 37, no. 3, pp. 277–288, March 1989.

7. R. Iltis and L. Milstein, "Performance analysis of narrow-band interference rejection techniques in DS spread-spectrum systems," *IEEE Trans. Communications*, vol. 32, no. 11, pp. 1169–1177, November 1984.

8. J. Ketchum and J. Proakis, "Adaptive algorithms for estimating and suppressing narrow-band interference in PN spread-spectrum systems," *IEEE Trans. Communications*, vol. 30, no. 5, pp. 913–923, May 1982.

9. L. Li and L. Milstein, "Rejection of narrow-band interference in PN spread-spectrum systems using transversal filters," *IEEE Trans. Communications*, vol. 30, no. 5, pp. 925–928, May 1982.

10. L. Milstein, "Interference rejection techniques in spread spectrum communications," *Proc. IEEE*, vol. 76, no. 6, pp. 657–671, June 1988.

11. L. Milstein and R. Iltis, "Signal processing for interference rejection in spread spectrum communications," *IEEE ASSP Mag.*, pp. 14–31, April 1986.

12. C. Pateros and G. Saulnier, "Adaptive correlator receiver performance in direct sequence spread spectrum communication," *Proc. IEEE Military Communications Conf.*, 1992.

13. C. Pateros and G. Saulnier, "A decision directed adaptive correlator for a direct sequence spread spectrum receiver," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1992.

14. C. Pateros and G. Saulnier, "Interference suppression and multipath mitigation using an adaptive correlator direct sequence spread spectrum receiver," *Proc. IEEE Int. Conf. Communications*, 1992.

15. R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread-spectrum communications - a tutorial," *IEEE Trans. Communications*, vol. 30, no. 5, pp. 855–884, May 1982.

16. R. Price and P. Green, Jr, "A communication technique for multipath channels," *Proc. IRE*, no. 46, pp. 555–570, March 1958.

17. J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, 1983.

18. S. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, no. 9, pp. 1349–138, September 1985.

19. G. Turin, "Introduction to spread-spectrum antimultipath techniques and their application to urban digital radio," *Proc. IEEE*, vol. 68, no. 5, pp. 328–352, March 1980.

20. M. Varanasi and B. Aazhang, "Multistage detector in asynchronous code-division mulitple access communications," *IEEE Trans. Communications*, vol. 38, no. 4, pp. 509–519, April 1990.

21. M. Varanasi and B. Aazhang, "Near-optimum detector in synchronous code-division multiple-access systems," *IEEE Trans. Communications*, vol. 39, no. 5, pp. 725–736, May 1991.

22. S. Verdú, "Minimum probability of error for asynchronous gaussian multiple-access channels," *IEEE Trans. Information Theory*, vol. 32, no. 1, pp. 85–96, January 1986.