# 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

# ABSTRACT

## FAST ADAPTIVE ALGORITHMS FOR SIGNAL SEPARATION

by
Robert A. Manzo

LMS and RLS type algorithms are suggested for decorrelation of multi-channel systems outputs. These algorithms act as signal separators when applied to unknown linear combinations of the inputs. The performance of the suggested algorithms is compared with that of the conventional LMS and RLS algorithms that minimize the mean square error. It is shown that the correlation matrix eigenvalue spread associated with the LMS decorrelator is always smaller than the eigenvalue spread corresponding to the conventional LMS, resulting in faster convergence speed for the decorrelator. A new RLS type decorrelator algorithm is suggested. The RLS decorrelator is shown to be faster than the LMS decorrelator, not affected by the eigenvalue spread, and comparable in speed with the conventional RLS algorithm. Convergence analysis by simulation shows that the RLS algorithms and the LMS decorrelator have wider regions of convergence than the conventional LMS.

# FAST ADAPTIVE ALGORITHMS FOR SIGNAL SEPARATION

by
Robert A. Manzo

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

May 1994

APPROVAL PAGE

# FAST ADAPTIVE ALGORITHMS FOR SIGNAL SEPARATION

Robert A. Manzo

Dr. Alexander Haimovich, Thesis Advisor / Date
Associate Professor of Electrical and Computer Engineering,
NJIT

Dr. Nirwan Ansari, Committee Member Date
Associate Professor of Electrical and Computer Engineering,
NJIT

Dr. Zoran Siveski, Committee Member Date
Assistant Professor of Electrical and Computer Engineering,
NJIT

## BIOGRAPHICAL SKETCH

**Author:** Robert A. Manzo

**Degree:** Master of Science in Electrical Engineering

**Date:** May 1994

**Undergraduate and Graduate Education:**

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1994
- Bachelor of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1992

**Major:** Electrical Engineering

This work is dedicated to
my parents, Settimio and Teresa Manzo, without their love and support this work would not have been possible.

# ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to his Master's Thesis advisor, Dr. Alexander Haimovich, for his guidance and support throughout this research.

Special thanks to Dr. Nirwan Ansari and Dr. Zoran Siveski for serving as members of the committee.

Finally, the author thanks all the Communications Laboratory members for their help and suggestions.

TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

Figure Page

Figure Page

# CHAPTER 1

# INTRODUCTION

In today's world, there is an ever increasing demand to process large amounts of information accurately and efficiently. As a result, technology has to be pushed to accommodate the need while still providing reliable results. At the heart of the technology are digital signal processing chips which perform the necessary functions required to process the information. These chips perform their tasks via algorithms designed to filter out unwanted signals while enhancing the desired signal. Additionally, these algorithms must be adaptive to combat the random characteristics of the transmission medium. Increasing number of applications for such adaptive algorithms, such as detection in multiple-access communication systems, separating multiple speech signals and canceling cross-polarized interference in dual polarized systems, has led to the need for faster, more efficient adaptive algorithms to perform such tasks. In this thesis, we will analyze and compare existing algorithms with a newly developed algorithm and discuss the results. We will deal primarily with the issue of separation of recovered, unknown independent sources from observations of a linear mixture of sources. Our system will be multi-channel input multi-channel output where each output is an unknown linear combination of the inputs.

Two approaches of addressing the problem of signal separation are considered here. The first is to treat the problem of undesired signals as interference and implement interference cancellation techniques. The signal-to-noise-and- interference ratio at each output channel is enhanced by suppressing co-channel interference. The mean-squared error (MSE) between the system output and a reference signal is minimized by a Wiener filter. This Weiner filter can be implemented using a variety of stochastic gradient algorithms, in particular the Least Mean Square (LMS) and Recursive Least Squares (RLS). An example of such an application is the cross-

polarized signal separator in a dual-polarized M-ary QAM system as suggested by Kavehard [21]. With this approach, however, the interference signal is indiscriminantly suppressed thus making this type of approach inadequate for systems trying to isolate independent signals.

The other approach treats the separation of superimposed signals directly. In the early eighties, a class of signal separators was proposed that in effect estimates the parameters of the mixture between the sources [4]. A steady state analysis of three cross-coupled "noise cancelers" structures was detailed in [2], and a comparison with other methods was detailed in [10]. The signals to be separated were assumed to be uncorrelated and separation at the output could be had provided that some reference was available by which the signal could be discriminated. The reference signal is comprised of the superimposition of the filtered outputs excluding the desired output. Since this structure successively improved the purity of the reference inputs and hence the corresponding outputs, the algorithm became known as the *bootstrap* algorithm [3].

Later, similar structures and adaptive rules were proposed independently by other researchers. Since then, work has been done on different structures and adaptive algorithms [12], [1], [22]. The latter works implemented a supervised training signal. In [1], Aazhang *et al.*, proposed a multi-layer neural network as a multiuser receiver. The adaptation of the weight values was performed with the implementation of the back-propagation algorithm. Their work was strictly empirical and suffered several disadvantages. The algorithm required supervised training, the network size increased exponentially with the increase in the number of users and it lacked convergence under realistic situations.

Mitra and Poor [22] proposed a single-layer perceptron scheme, with both linear and nonlinear adaptive algorithms for single user demodulation in a multiuser channel. In particular, they showed that the single-layer perceptron weights

converged to the optimal values for the noiseless multiuser case both in the synchronous and asynchronous transmission cases. In the presence of additive Gaussian noise, the neural network performed satisfactorily and was less susceptible to the noise compared to other linear algorithms. Two drawbacks were its slow rate of convergence and the need for supervised training.

The one distinguishing feature of the previous two methods is the use of supervised learning for proper operation. In some applications, access to the known source signal is not possible. Consequently, a more desirable and pragmatic method of signal separation is the implementation of a unsupervised learning scheme through a mode known as the decision-directed mode. This is the method of choice and is implemented in a class of signal separators called *blind* signal separators and is studied by [18], [24], [5], [6]. In [18], Jutten and Herault addressed the problem based on a linear feedback neural network. Their algorithm assumed the unknown sources to be statistically independent. Therefore, in order to have a good solution to the problem of source separation, the outputs had to be statistically independent and not just uncorrelated. Since it was difficult to devise a criterion for testing statistical independence, a cost function for decorrelation of the outputs had to be designed. Thus, independence was not achieved directly but was approached by minimizing higher order cross moments utilizing stochastic gradient algorithms with cost functions defined by nonlinear functions of the outputs [19]. The algorithm did have its drawbacks in convergence and stability [28]. In [20], Jutten *et. al.* proposed an algorithm as an extension to the work in [18]. They generalized the algorithm to apply to a more interesting case of convolutive mixtures. In this case, the available information at the adaptive filter is a superposition of unknown sources after unknown filtering in unknown linear (FIR) filters which model the properties of the signals from sources to detectors. Thus, separation of unknown sources can be achieved by estimating an inverse FIR filter. The HJ algorithm is implemented

to estimate the various coefficients of the FIR filters. Results proved satisfactory for simulated signals but due to the simplicity of the model chosen, performance in real situations was insufficient.

In [24], Moreau and Macchi proposed two other structures, feedforward and mixed, in comparison to the one in [18]. While [18] required implementation of constraints for realizability, the two structures in [24] did not. Also, utilizing the same adaptation rule as in [18], the feedforward structure exhibited a faster convergence for certain signal mixtures. However, drawbacks reported in [18] remain unsolved by these researchers. In [5], Burel proposed a new approach to the work of Jutten and Herault based on the ideas of back propagation learning on neural networks. His contributions include the development of an algorithm designed to minimize a cost function and a means of dealing with nonlinear mixtures. Satisfactory results were obtained for both the linear and nonlinear cases and his algorithm was resistant to noise corruption. Convergence, however, was still not guaranteed.

Along similar lines, Compernolle and Van Gerven [8] and [7] developed a symmetric adaptive algorithm for noise cancellation and signal separation. The algorithm was derived from the interpretation of an adaptive noise canceler as a decorrelator between signal estimate and noise, where the noise reference was replaced with a "signal free" noise estimate. Thus a symmetric adaptive decorrelator for signal separation was obtained. Convergence of the algorithm, however, to the desired solution could only be guaranteed for a subclass of signal separation problems.

Finally, the structures introduced earlier by Bar-Ness in [2] for decorrelating the outputs of multi-channel systems can also be applied to multi-dimensional blind adaptive signal separation. The implementation was performed and analyzed in [13], [14], [11], [15]. This adaptation operated on the premise that cross-correlations of the outputs or of nonlinear functions of the outputs, are used to control filter weights

of both feedforward and feedback structures. Simulations showed that this LMS decorrelating algorithm is faster, for any number of channels, than the conventional LMS algorithm.

In this thesis, we analyze and compare various blind separation algorithms based on two criteria: the Mean Squared Error (MSE) and decorrelation. Our focus, will be to (1) prove , by means of mathematical analysis and computer simulations, that the eigenvalue spread associated with the LMS decorrelating algorithm is smaller than the eigenvalue spread associated with the LMS algorithm thus providing an explanation for previously reported results regarding the speed of convergence of the LMS decorrelating algorithm [11], [15] and (2) introduce and analyze our new decorrelating RLS type algorithm.

In Chapter 2, the system model is presented. In Chapter 3, we define the separation criteria and based on them compute the expressions for the steady state weight vector. In Chapter 4, the adaptive algorithms and convergence analysis of the LMS and LMS decorrelating algorithms are developed. The performance measures are defined in Chapter 5 and the various algorithms are compared via computer simulations in Chapter 6. The conclusion of the thesis is given in Chapter 7.

# CHAPTER 2

## SYSTEM MODEL

The model studied here reflects a multi-user communication system, such as CDMA, of $N$ sources transmitted through $N$ channels as shown in Figure 2.1.

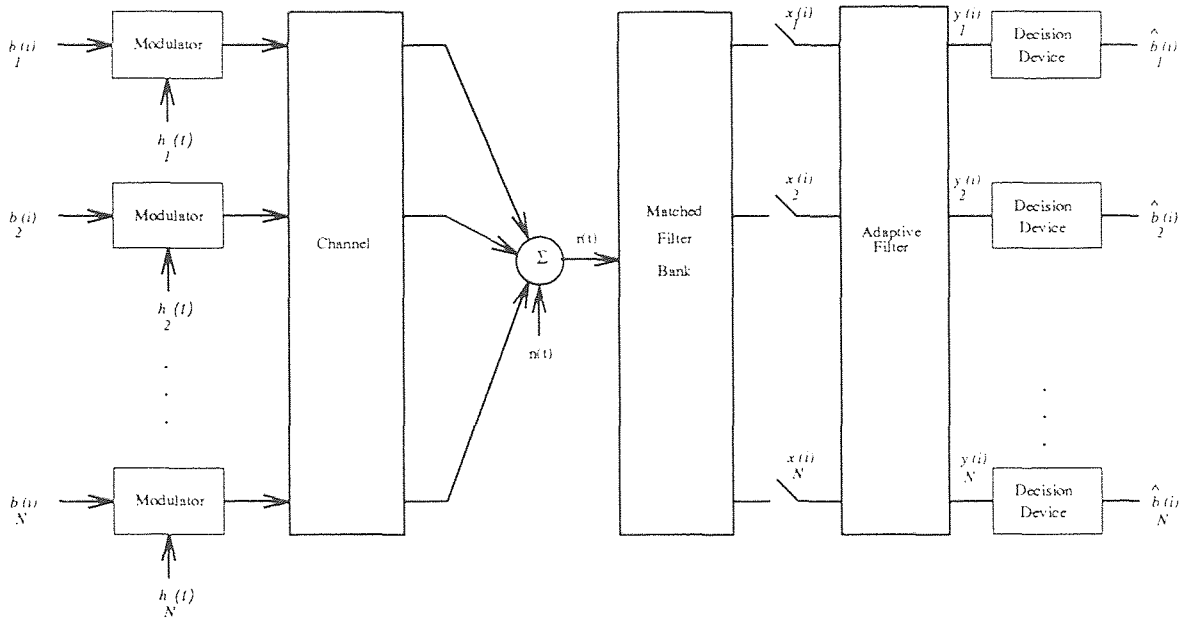


**Figure 2.1** Multi-user communication system model

Before proceeding, we make the following assumptions:

1. The sources are independent.

2. The outputs of the channels consist of a linear mixture of the inputs.

3. The number of inputs is equal to the number of sources.

4. The information bits from the $N$ sources are transmitted simultaneously.

5. The information bits are statistically independent and equiprobable.

Define $\mathbf{b}(i) = [b_1(i), b_2(i), \cdots, b_N(i)]^T$ as a vector corresponding to the $N$ sources' information bits transmitted at the $i$-th time interval, where $b_n \in \{-1, 1\}$.

$1 \leq n \leq N$, is the $n$-th information bit at the $i$-th time interval. Each information bit has a duration time of $T_s$. Define $\mathbf{h}(t) = [h_1(t), h_2(t,), \cdots, h_N(t)]^T$ as the vector corresponding to the signature waveforms of the $N$ sources, where $h_n(t)$, $1 \leq n \leq N$, is the $n$-th signature waveform with its time interval between $[0, T_s)$. The energy of $h_n(t)$ is normalized to 1, i.e. $\int_0^{T_s} |h_n(t)|^2 dt = 1$.

Each of the $N$ sources has each of the information bits, $b_n(i)$, encoded by the signature waveform $h_n(t)$. Subsequently, all the $N$ sources are transmitted through the same channel every $i$-th time interval where the channel is assumed to be nondispersive and possibly slowly, time varying. Define $n(t)$ as the background channel additive white Gaussian noise with a zero mean and a power density of $\sigma_{\mathbf{n}}^2$. In the case where all users in the system are bit-synchronous, the signal $r(t)$ received at the output of the channel is the sum of all the $N$ signals and the noise $n(t)$, expressed as:

$$r(t) = \sum_i \mathbf{E}\mathbf{b}\mathbf{h}^T + n(t) \tag{2.1}$$

or

$$r(t) = \sum_{n=1}^{N} \sum_{m=1}^{N} \sqrt{\xi_n} b_n(i) h_n(t) + n(t) \tag{2.2}$$

The received signal $r(t)$ is then demodulated by a bank of matched filters. The output $x_n(t)$ is the result of the convolution of $r(t)$ and $h_n(t)$, which results in

$$x_n(t) = \sum_i \sum_{m=1}^{N} a_{nm} \sqrt{\xi_m} b_m(i) + \nu(t) \tag{2.3}$$

for $1 \leq n \leq N$. $\nu(t)$ is Gaussian noise, possibly correlated, resulting from the additive white Gaussian noise, $n(t)$, passing through the matched filter bank. $\xi_n$ is the energy associated with the $n$-th user. The mixture cross-correlation coefficients due to the matched filtering, $a_{ij}$, where $i \neq j$, $i, j = 1, 2, \cdots, N$, are assumed to be less than unity in magnitude. The diagonal coefficients, $a_{ii}$, $i = 1, 2, \cdots, N$, are assumed, without loss of generality, to be unity. Since all the users are assumed to be bit-synchronous, we can drop the time dependency for convenience of presentation

and represent the matched filter output $\mathbf{x}(t)$ in vector form such as:

$$\mathbf{x} = \mathbf{AEb} + \mathbf{v}, \tag{2.4}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_N]^T$, $\mathbf{v} = [\nu_1, \nu_2, \cdots, \nu_N]^T$, $\mathbf{A}$ is the mixture matrix and $\mathbf{E}$ is a diagonal matrix, $\mathrm{diag}\mathbf{E} = \left[\sqrt{\xi_1}, \sqrt{\xi_2}, \cdots, \sqrt{\xi_N}\right]^T$. The sampled noise is has a zero mean and a covariance of

$$\begin{aligned} E\left[\mathbf{vv}^T\right] &= E\left[\mathbf{hnn}^T\mathbf{h}^T\right] & (2.5)\\ &= \sigma_{\mathrm{n}}^2\mathbf{Ihh}^T & (2.6)\\ &= \sigma_{\mathrm{n}}^2\mathbf{H} & (2.7) \end{aligned}$$

where $\mathbf{n}$ is the vector representing the background additive white Gaussian noise and $\mathbf{H}$ is the known matrix of the noise cross-correlations between the outputs of the matched filters.

The $n$-th user output of the adaptive filter, represented by the weight vector $\mathbf{w}_n^T = [w_{n1}, w_{n2}, \cdots, w_{nN}]^T$, $1 \leq n \leq N$, is defined as:

$$y_n = \mathbf{w}_n^T\mathbf{x}. \tag{2.8}$$

The adaptive filter structure of interest in this work will be the feed-forward structure shown in Figure 2.2 for the case of two users.

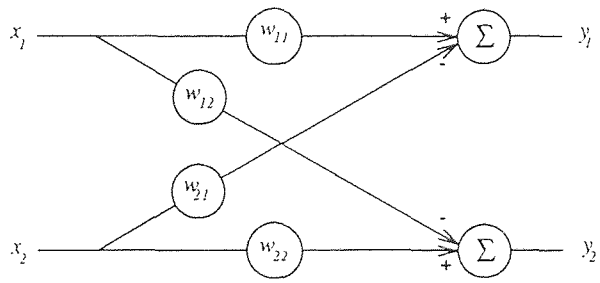


**Figure 2.2** Feed-forward filter structure

where $w_{ij}$ are the filter's adaptive weights.

For BPSK transmission, a nonlinear function can be used to detect the data bit $\widehat{b}_n$, defined as:

$$\widehat{b}_n = f(y_n). \tag{2.9}$$

where $f(\cdot)$ can be either the signum function, $\mathrm{sgn}(\cdot)$, for hard decisions or the hyperbolic tangent function, $\tanh(\cdot)$, for soft decisions.

# CHAPTER 3

# SIGNAL SEPARATION CRITERIA

The two signal separation criteria considered here are (1) the Mean Squared Error (MSE) separation and (2) signal decorrelation criteria. The criteria are used to develop control algorithms for adaptive filter weights. Our focus will be on separation of the $n$-th user signal from the rest of the signals, other channels being treated similarly.

## 3.1 MSE Separation Criterion

The MSE signal separator minimizes the mean squared error between its output and a reference signal. Typically, the reference signal is initially supplied by a training signal. After the adaptive weights have converged and the errors between the training signal and the output are small, the detector is switched to operate in the decision-directed mode. The reference signal is then supplied by the estimate output signal. The MSE separator is in effect an optimum linear detector for the $n$-th user, acting as a canceler for the co-channel interference from other users.

Consider the system depicted in Figure 3.1.

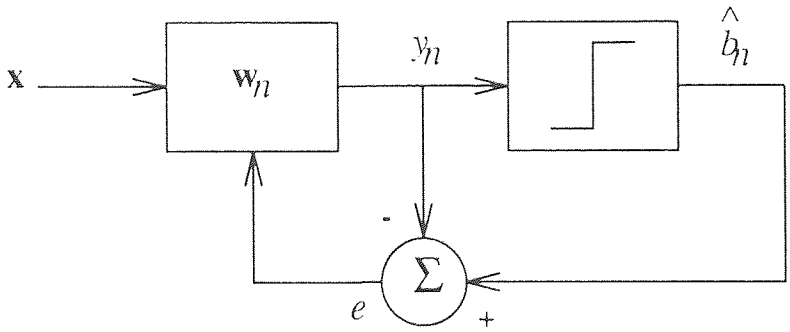


**Figure 3.1** Adaptive MSE Signal Separator

In the decision-directed mode, the MSE separator minimizes the error between the reference signal, in this case the estimated output symbol denoted by $\widehat{b}_n(k)$ and the output $y_n(k)$ . The estimations error can then be expressed as:

$$\epsilon(k) = \widehat{b}_n(k) - y_n(k) \tag{3.1}$$

The MSE is correspondingly denoted by $\epsilon = E\left[|e(k)|^2\right]$. Since the signals are assumed stationary, we can suppress the time dependency for convenience of presentation.

To establish the MSE criterion, we need to develop the steady state weight vector equation for separating the $n$-th user signal. Let $\nabla$ denote the gradient vector of the MSE. Differentiating the MSE, $\epsilon$, with respect to $\mathbf{w}_n^T$ [17], we get

$$\begin{aligned}
\nabla &= \frac{\partial \epsilon}{\partial \mathbf{w}_n^T} \\
&= -2\frac{\partial \epsilon}{\partial \mathbf{w}_n^T}\epsilon \\
&= -2E\left[-\mathbf{x}\left(\widehat{b}_n - \mathbf{x}^T\mathbf{w}_n\right)\right] \\
&= -E\left[\widehat{b}_n\mathbf{x}\right] + 2E\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{w}_n
\end{aligned} \tag{3.2}$$

Ignoring the scaling constant and minimizing (3.2) to obtain the optimum weight vector results in

$$\begin{aligned}
-E\left[\widehat{b}_n\mathbf{x}\right] + E\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{w}_n &= 0 \\
E\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{w}_n &= E\left[\widehat{b}_n\mathbf{x}\right] \\
\mathbf{w}_n &= \left(E\left[\mathbf{x}\mathbf{x}^T\right]\right)^{-1} E\left[\widehat{b}_n\mathbf{x}\right] \\
\mathbf{w}_n &= \mathbf{R}_x^{-1}\mathbf{r}_{\widehat{b}_n x}.
\end{aligned} \tag{3.3}$$

where $\mathbf{R}_x = E\left[\mathbf{x}\mathbf{x}^T\right]$ is the input correlation matrix and $\mathbf{r}_{\widehat{b}_n x} = E\left[\widehat{b}_n\mathbf{x}\right]$ is the cross-correlation vector between the input and the estimated output symbol. The resulting steady state weight vector equation is the well known Weiner-Hopf equation.

Inserting the input vector given by (2.4) leads to the input correlation matrix written as:

$$\begin{aligned}
\mathbf{R}_x &= E\left[\mathrm{x}\mathrm{x}^T\right] \\
&= E\left[(\mathbf{A}\mathbf{E}\mathrm{b}+\mathrm{v})\left(\mathrm{b}^T\mathbf{E}^T\mathbf{A}^T+\mathrm{v}^T\right)\right] \\
&= E\left[\mathbf{A}\mathbf{E}\mathrm{b}\mathrm{b}^T\mathbf{E}^T\mathbf{A}^T\right]+E\left[\mathbf{A}\mathbf{E}\mathrm{b}\mathrm{v}^T\right]+E\left[\mathrm{v}\mathrm{b}^T\mathbf{E}^T\mathbf{A}^T\right]+E\left[\mathrm{v}\mathrm{v}^T\right] \\
&= \mathbf{A}\mathbf{E}E\left[\mathrm{b}\mathrm{b}^T\right]\mathbf{E}^T\mathbf{A}^T+\mathbf{A}\mathbf{E}E\left[\mathrm{b}\mathrm{v}^T\right]+E\left[\mathrm{v}^T\mathrm{b}^T\right]\mathbf{E}^T\mathbf{A}^T+E\left[\mathrm{v}\mathrm{v}^T\right]. \quad (3.4)
\end{aligned}$$

Since it is assumed that the sources and the additive Gaussian noise are uncorrelated, then $E\left[\mathrm{v}\mathrm{b}^T\right]=E\left[\mathrm{b}\mathrm{v}^T\right]=\mathbf{0}$ which simplifies (3.4) to

$$\begin{aligned}
\mathbf{R}_x &= \mathbf{A}\mathbf{E}\mathbf{E}^T\mathbf{A}^T+\sigma_{\mathrm{n}}^2\mathbf{H} \\
&= \mathbf{A}\mathbf{E}^2\mathbf{A}^T+\sigma_{\mathrm{n}}^2\mathbf{H} \quad (3.5)
\end{aligned}$$

where $E\left[\mathrm{b}\mathrm{b}^T\right]=\mathbf{I}$ and $\mathbf{H}$ is the noise correlation matrix defined in Chapter 2.

The cross-correlation vector is given by

$$\begin{aligned}
\mathrm{r}_{\hat{b}_n x} &= E\left[\hat{b}_n\mathrm{x}\right] \\
&= E\left[\hat{b}_n\left(\mathbf{A}\mathbf{E}\mathrm{b}+\mathrm{v}\right)\right] \\
&= \mathbf{A}\mathbf{E}E\left[\hat{b}_n\mathrm{b}\right]+E\left[\hat{b}_n\mathrm{v}\right]. \quad (3.6)
\end{aligned}$$

To simplify (3.6) further, we claim that in the decision-directed mode

$$E\left[\hat{b}_n\mathrm{b}\right]=q_n\mathrm{u}_n \quad (3.7)$$

and

$$E\left[\hat{\mathrm{b}}\mathrm{b}^T\right]=\mathbf{Q}, \quad (3.8)$$

where $\mathbf{u}_n$ is a unit vector with the $n$-th element equal to 1, $q_n$ is a constant defined as:

$$q_n=E\left[\hat{b}_n b_n\right] \quad (3.9)$$

and $\mathbf{Q}$ is a diagonal matrix, $\mathrm{diag}\mathbf{Q}=\left[q_1,\ \cdots\ q_N\right]^T$. The proof of (3.7) and (3.8) is given in Appendix A.

Substituting (3.7) back into (3.6), we get

$$\mathbf{r}_{\widehat{b}_n x} = q_n \mathbf{A}\mathbf{E}\mathbf{u}_n + E\left[\widehat{b}_n \mathbf{v}\right] \tag{3.10}$$

In the absence of noise, the MSE criterion indeed does lead to complete signal separation. This can be shown by substituting (3.5) and (3.10) into (3.3) and getting

$$\begin{aligned} \mathbf{w}_n &= \mathbf{R}_x^{-1}\mathbf{r}_{\widehat{b}_n x} \\ &= \left(\mathbf{A}\mathbf{E}^2\mathbf{A}^T\right)^{-1}\left(q_n \mathbf{A}\mathbf{E}\mathbf{u}_n\right) \\ &= q_n \mathbf{A}^{-T}\mathbf{E}^{-1}\mathbf{E}^{-1}\mathbf{A}^{-1}\mathbf{A}\mathbf{E}\mathbf{u}_n \\ &= q_n \mathbf{A}^{-T}\mathbf{E}^{-1}\mathbf{u}_n. \end{aligned} \tag{3.11}$$

The input vector is then applied to the filter and a scaled version of the information bit is recovered and given by:

$$\begin{aligned} y_n &= \mathbf{w}_n^T \mathbf{x} \\ &= \left(q_n \mathbf{A}^{-T}\mathbf{E}^{-1}\mathbf{u}_n\right)^T (\mathbf{A}\mathbf{E}\mathbf{b}) \\ &= q_n \mathbf{u}_n^T \mathbf{E}^{-T}\mathbf{A}^{-T}\mathbf{A}\mathbf{E}\mathbf{b} \\ &= q_n \mathbf{u}_n^T \mathbf{b} \\ &= q_n b_n. \end{aligned} \tag{3.12}$$

## 3.2 Decorrelation Signal Separation

The decorrelation signal separator shown in Figure 3.2 differs from the MSE signal separator in the way the weight vector is controlled. The MSE weight vector is controlled by the error derived solely from the output corresponding to that user. The decorrelation weight vector, on the other hand, is controlled by the outputs of all the channels. In short, the separation process is the decorrelation of the output $y_n$ from the other users.

In a mode similar to the decision-directed mode of the MSE separator, the decorrelation criterion is devised to decorrelate the output $y_n$ from the rest of the

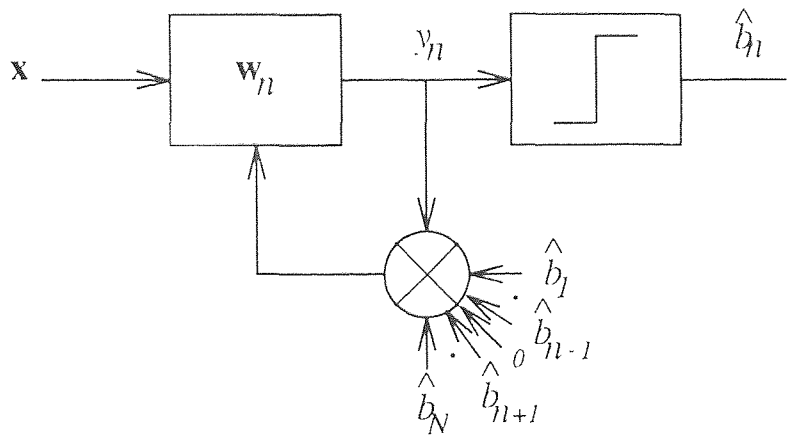


**Figure 3.2** Decorrelating Signal Separator

signals. Decorrelation is achieved when

$$E\left[\hat{\mathbf{b}}y_n\right] = c_n \mathbf{u}_n. \tag{3.13}$$

where $c_n$ is a constant.

To develop the criterion, we will first define and optimize a cost function to compute the steady state decorrelating weight vector. We begin by examining the recursive equation based on the conventional LMS algorithm adapted for the decorrelating signal separator [11]

$$\mathbf{w}_n(k) = \mathbf{w}_n(k-1) - \mu y_n \hat{\mathbf{B}}_n, \tag{3.14}$$

where $\hat{\mathbf{B}}_n$ is the estimated output vector of all but the $n$-th user defined as:

$$\begin{aligned} \hat{\mathbf{B}}_n &= \left[\hat{b}_1, \cdots \hat{b}_{n-1}, 0, \hat{b}_{n+1}, \cdots \hat{b}_N\right]^T \\ &= \left(\mathbf{I} - \mathbf{u}_n \mathbf{u}_n^T\right)\hat{\mathbf{b}} \end{aligned} \tag{3.15}$$

where the $n$-th element is equal to zero. After examining (3.14), it is obvious that the algorithm converges when

$$E\left[\mathbf{w}_n(k)\right] = E\left[\mathbf{w}_n(k-1)\right] \tag{3.16}$$

or alternatively, when the correction term $E\left[\hat{\mathbf{B}}_n y_n\right]$ is equal to zero. This condition for convergence can be rewritten as:

$$\begin{aligned} E\left[\hat{\mathbf{B}}_n y_n\right] &= 0 \\ E\left[\hat{\mathbf{B}}_n \mathbf{x}^T\right] \mathbf{w}_n &= 0 \\ \mathbf{R}_{\hat{\mathbf{B}}_n x} \mathbf{w}_n &= 0 \end{aligned} \tag{3.17}$$

where $\mathbf{R}_{\hat{\mathbf{B}}_n x}$ is the correlation matrix of the *other* estimated bits and the input.

Twice the result calculated in (3.17) is equivalent to the result of the minimization of the cost function $J$ with respect to $\mathbf{w}_n^T$ denoted by

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{w}_n^T} &= 2\mathbf{R}_{\hat{\mathbf{B}}_n x} \mathbf{w}_n \\ \nabla J &= 2\mathbf{R}_{\hat{\mathbf{B}}_n x} \mathbf{w}_n. \end{aligned} \tag{3.18}$$

Subsequently, the cost function $J$ is calculated to be

$$J = \mathbf{w}_n^T \mathbf{R}_{\hat{\mathbf{B}}_n x} \mathbf{w}_n \tag{3.19}$$

However, $\mathbf{R}_{\hat{\mathbf{B}}_n x}$ should be expressed in terms of the unaltered $\hat{\mathbf{b}}$ vector as follows:

$$\mathbf{R}_{\hat{\mathbf{B}}_n x} = \left(\mathbf{I} - \mathbf{u}_n \mathbf{u}_n^T\right) \mathbf{R}_{\hat{b}x}, \tag{3.20}$$

where $\mathbf{R}_{\hat{b}x} = E\left[\hat{\mathbf{b}}\mathbf{x}^T\right]$ and is the correlation matrix of *all* the estimated bits and the input. Therefore the minimization of the cost function $J = \mathbf{w}_n^T \mathbf{R}_{\hat{\mathbf{B}}_n x} \mathbf{w}_n$ is equivalent to the minimization of the cost function $J = \mathbf{w}_n^T \left(\mathbf{I} - \mathbf{u}_n \mathbf{u}_n^T\right) \mathbf{R}_{\hat{b}x}$. This is shown by considering, without loss of generality, the case where $n$=1 and defining the vector partitions

$$\hat{\mathbf{B}}_n^T = \begin{pmatrix} 0 & \vdots & \hat{\mathbf{B}}'^T \end{pmatrix}, \; \mathbf{x}^T = \begin{pmatrix} x_1 & \vdots & \mathbf{X}^T \end{pmatrix}.$$

where $\hat{\mathbf{B}}'^T = \left[\hat{b}_2, \hat{b}_3, \cdots, \hat{b}_N\right]^T$ and $\mathbf{X} = [x_2, x_3, \cdots, x_N]^T$. Next, we can represent the correlation matrix $\mathbf{R}_{\hat{\mathbf{B}}_n x}$ as:

$$\mathbf{R}_{\hat{\mathbf{B}}_n x} = E\left[\hat{\mathbf{B}}_n \mathbf{x}^T\right]$$

$$= E\left[\begin{pmatrix} 0 & \vdots & \mathbf{0}^T \\ \cdots & \vdots & \cdots \\ \hat{\mathbf{B}}'^T x_1 & \vdots & \hat{\mathbf{B}}'^T \mathbf{X}^T \end{pmatrix}\right]$$

where $\hat{\mathbf{B}}'^T \mathbf{X}^T$ is the principal submatrix and

$$E\left[\hat{\mathbf{B}}'^T \mathbf{X}^T\right] = \mathbf{R}_{\hat{B}'x} \tag{3.21}$$

Similarly, we define the following vector partitions for the correlation matrix $\mathbf{R}_{\hat{b}x}$ as:

$$\hat{\mathbf{b}}^T = \begin{pmatrix} \hat{b}_1 & \vdots & \hat{\mathbf{B}}'^T \end{pmatrix}, \ \mathbf{x}^T = \begin{pmatrix} x_1 & \vdots & \mathbf{X}^T \end{pmatrix}$$

where the only difference is that all the estimated bits are included. Consequently, $\mathbf{R}_{\hat{b}x}$ can be represented as:

$$\begin{aligned} \mathbf{R}_{\hat{b}x} &= E\left[\hat{\mathbf{b}}\mathbf{x}^T\right] \\ &= E\left[\begin{pmatrix} \hat{b}_1 x_1 & \vdots & \hat{b}_1 \mathbf{X}^T \\ \cdots & \vdots & \cdots \\ \hat{\mathbf{B}}'^T x_1 & \vdots & \hat{\mathbf{B}}'^T \mathbf{X}^T \end{pmatrix}\right] \end{aligned}$$

Since the $n$-th row of $\mathbf{R}_{\hat{\mathbf{B}}_n x}$ consists of all zeroes, an adaptive algorithm would have no control over $w_{nn}$. Consequently, $w_{nn}$ can be set arbitrarily, for example let $w_{nn} = 1$. Therefore after analyzing both partitioned matrices, we can state that the minimization of (3.19) is equivalent to the minimization of

$$J = \mathbf{w}_n^T \mathbf{R}_{\hat{b}x} \mathbf{w}_n, \tag{3.22}$$

subject to the constraint $\mathbf{w}_n^T \mathbf{u}_n = 1$.

Indeed, the minimization of the cost function $J$ given in (3.22) does satisfy the decorrelation condition defined in (3.13) as shown below:

$$\begin{aligned} \nabla J &= 2\mathbf{R}_{\hat{b}x} \mathbf{w}_n \\ &= 2E\left[\hat{\mathbf{b}}\mathbf{x}^T\right] \mathbf{w}_n \\ &= 2E\left[\hat{\mathbf{b}}\mathbf{x}^T \mathbf{w}_n\right] \\ &= 2E\left[\hat{\mathbf{b}}\, y_n\right] \qquad (3.23) \\ &= c_n \mathbf{u}_n, \qquad (3.24) \end{aligned}$$

The steady state decorrelating weight vector equation is calculated implementing the method of Lagrange multipliers [27],

$$\begin{array}{lrcl} \text{minimize} & J & = & \mathbf{w}_n^T \mathbf{R}_{\widehat{bx}} \mathbf{w}_n \\ \text{subject to} & \mathbf{w}_n^T \mathbf{u}_n & = & 1 \end{array} \tag{3.25}$$

$$\begin{aligned} J &= \mathbf{w}_n^T \mathbf{R}_{\widehat{bx}} \mathbf{w}_n - \beta \left( \mathbf{w}_n^T \mathbf{u}_n - 1 \right) \\ \frac{\partial J}{\partial \mathbf{w}_n^T} &= 2\mathbf{R}_{\widehat{bx}} \mathbf{w}_n - \beta \mathbf{u}_n \\ 2\mathbf{R}_{\widehat{bx}} \mathbf{w}_n - \beta \mathbf{u}_n &= 0 \\ \mathbf{w}_n &= \frac{1}{2} \beta \mathbf{R}_{\widehat{bx}}^{-1} \mathbf{u}_n. \end{aligned} \tag{3.26}$$

where $\beta$ is the scaling factor necessary to meet the linear constraint and is calculated as follows:

$$\begin{aligned} \mathbf{w}_n^T \mathbf{u}_n &= \frac{1}{2} \left( \beta \mathbf{R}_{\widehat{bx}}^{-1} \mathbf{u}_n \right)^T \mathbf{u}_n \\ \frac{1}{2} \beta \mathbf{u}_n^T \mathbf{R}_{\widehat{bx}}^{-1} \mathbf{u}_n &= 1 \\ \beta &= 2 \left( \mathbf{u}_n^T \mathbf{R}_{\widehat{bx}}^{-1} \mathbf{u}_n \right)^{-1}. \end{aligned} \tag{3.27}$$

For the noiseless case, we can show that the criterion in (3.13) indeed does lead to signal separation. Substituting the noiseless input vector into (3.13) results in

$$\begin{aligned} E\left[ \widehat{\mathbf{b}} y_n \right] &= E\left[ \widehat{\mathbf{b}} \left( \mathbf{w}_n^T \mathbf{A} \mathbf{E} \mathbf{b} \right) \right] \\ &= E\left[ \widehat{\mathbf{b}} \mathbf{b}^T \right] \mathbf{E}^T \mathbf{A}^T \mathbf{w}_n. \end{aligned} \tag{3.28}$$

Substituting (3.9) into (3.28) we get

$$E\left[ \widehat{\mathbf{b}} y_n \right] = \mathbf{Q} \mathbf{E}^T \mathbf{A}^T \mathbf{w}_n \tag{3.29}$$

which is equivalent to

$$E\left[ \widehat{\mathbf{b}} y_n \right] = c_n \mathbf{u}_n. \tag{3.30}$$

# CHAPTER 4

# ADAPTIVE ALGORITHMS

The MSE and decorrelation criteria were developed for a stationary environment. However, when the signal environment is nonstationary an adaptive algorithm is needed to obtain the optimum weight vector. In our case, we use steepest-descent type algorithms to calculate the weight vectors for the MSE and decorrelating separators.

## 4.1 LMS Error Algorithm

The LMS algorithm for updating the weight vector of the MSE separator is given by:

$$\begin{aligned} \mathbf{w}_n(k) &= \mathbf{w}_n(k-1) + \mu \epsilon(k)\mathbf{x}(k) \\ \epsilon(k) &= \widehat{b}_n(k) - \mathbf{x}^T(k)\mathbf{w}_n(k-1). \end{aligned} \tag{4.1}$$

To distinguish it from the *LMS decorrelating* algorithm or *LMS decorrelator*, we refer to this algorithm as the *LMS error* algorithm for obvious reasons. Since the steepest-descent algorithm involves the presence of feedback, this type of algorithm is subject to the possibility of becoming unstable. From (4.1), the two possible sources of instability are (1) the step-size parameter $\mu$, and (2) the input vector $\mathbf{x}(k)$. The convergence analysis of the LMS error algorithm is well known [17]. Consequently, since the eigenvalues of $\mathbf{R}_x$ are all real and positive, it follows that the necessary condition for convergence of the LMS error algorithm is that the step-size parameter $\mu$ satisfy the condition

$$0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{R}_x)}, \tag{4.2}$$

where $\lambda_{\max}(\mathbf{R}_x)$ is the largest eigenvalue of $\mathbf{R}_x$. Therefore, provided the step-size parameter is set within the bounds defined by (4.2), the weight vector computed by

using the LMS error algorithm converges the optimum solution, defined in (3.3), as the number of iterations approaches infinity.

## 4.2 RLS Error Algorithm

The *RLS error* algorithm for the MSE separator is based on a recursive implementation of the Weiner-Hopf equation. The requirement of unsupervised training resulted in the modification of the typical RLS recursive equations to an alternative weighting scheme that permits past data to be regarded either as less important or more important than the current data is [23], [9]. The estimated updates for the covariance matrix $\mathbf{R}_x$ and the cross-correlation vector $\mathbf{r}_{\hat{b}_n x}$ are expressed as follows:

$$\mathbf{R}_x(k) = (1-\alpha)\,\mathbf{R}_x(k-1) + \alpha \mathbf{x}(k)\mathbf{x}^T(k) \tag{4.3}$$

$$\mathbf{r}_{\hat{b}_n x}(k) = (1-\alpha)\,\mathbf{r}_{\hat{b}_n x}(k-1) + \alpha \mathbf{x}(k)\hat{b}_n(k) \tag{4.4}$$

where $\alpha$ is the forgetting factor. To implement (3.3), we first must determine the inverse of the covariance matrix using the Matrix Inversion Lemma [17]

$$\begin{aligned}\mathbf{R}_x^{-1}(k) &= \frac{1}{1-\alpha}\mathbf{R}_x^{-1}(k-1) - \\ &\quad \frac{\frac{1}{1-\alpha}\mathbf{R}_x^{-1}(k-1)\mathbf{x}(k)\mathbf{x}(k)^T\frac{1}{1-\alpha}\mathbf{R}_x^{-1}(k-1)}{\frac{1}{\alpha}+\mathbf{x}(k)^T\frac{1}{1-\alpha}\mathbf{R}_x^{-1}(k-1)\mathbf{x}(k)}.\end{aligned} \tag{4.5}$$

For convenience of computation, we define

$$\mathbf{g}(k) = \frac{\mathbf{R}_x^{-1}(k-1)\mathbf{x}(k)}{\frac{1-\alpha}{\alpha}+\mathbf{x}(k)^T\mathbf{R}_x^{-1}(k-1)\mathbf{x}(k)} \tag{4.6}$$

as the variable gain unique to the RLS error algorithm replacing the fixed-valued step-size parameter $\mu$ in the LMS error algorithm. The variable gain is the essential reason for its fast adaptive nature.

Using (4.6), we can rewrite (4.5) as

$$\mathbf{R}_x^{-1}(k) = \frac{1}{1-\alpha}\left[\mathbf{I} - \mathbf{g}(k)\mathbf{x}(k)^T\right]\mathbf{R}_x^{-1}(k-1). \tag{4.7}$$

Then we substitute (4.4) into (3.3) and get

$$\mathbf{w}_n = \mathbf{R}_x^{-1}(k)\left[(1-\alpha)\,\mathbf{r}_{\hat{b}_n x}(k-1)\right] + \alpha \mathbf{x}(k)\,\hat{b}_n \mathbf{R}_x^{-1}(k). \tag{4.8}$$

Next, we substitute (4.7) in the first term on the right side of (4.8) and get

$$\begin{aligned} \mathbf{w}_n &= \mathbf{R}_x^{-1}(k-1)\,\mathbf{r}_{\hat{b}_n x}(k-1) \\ &\quad -\mathbf{g}(k)\,\mathbf{x}(k)^T\,\mathbf{R}_x^{-1}(k-1)\,\mathbf{r}_{\hat{b}_n x}(k-1) \\ &\quad +\alpha \mathbf{R}_x^{-1}(k)\,\mathbf{x}(k)\,\hat{b}_n. \end{aligned} \tag{4.9}$$

Acknowledging the fact that $\mathbf{R}_x^{-1}(k)\,\mathbf{x}(k)$ equals the gain vector $\mathbf{g}(k)$, we get the desired recursive equation as follows:

$$\mathbf{w}_n(k) = \mathbf{w}_n(k-1) + \mathbf{g}(k)\,e(k) \tag{4.10}$$

where

$$e(k) = \alpha \hat{b}_n - \mathbf{x}^T(k)\,\mathbf{w}_n(k-1) \tag{4.11}$$

The algorithm is summarized below:

$$\mathbf{g}(k) = \frac{\mathbf{R}_x^{-1}(k-1)\,\mathbf{x}(k)}{\frac{1-\alpha}{\alpha} + \mathbf{x}^T(k)\,\mathbf{R}_x^{-1}(k-1)\,\mathbf{x}(k)} \tag{4.12}$$

$$\mathbf{R}_x^{-1}(k) = \frac{1}{1-\alpha}\left[\mathbf{I} - \mathbf{g}(k)\,\mathbf{x}^T(k)\right]\mathbf{R}_x^{-1}(k-1) \tag{4.13}$$

$$e(k) = \alpha \hat{b}_n - \mathbf{x}^T(k)\,\mathbf{w}_n(k-1) \tag{4.14}$$

$$\mathbf{w}_n(k) = \mathbf{w}_n(k-1) + \mathbf{g}(k)\,e(k) \tag{4.15}$$

## 4.3 LMS Decorrelator Algorithm

The LMS decorrelator is a steepest-descent algorithm that seeks to null the instantaneous estimate of the gradient of the cost function $J$ reproduced here for convenience

$$J = \mathbf{w}_n^T \mathbf{R}_{\hat{b}_n} \mathbf{w}_n.$$

The instantaneous estimate of $\nabla J = \mathbf{R}_{\hat{b}x}\mathbf{w}_n$ is given as

$$\begin{aligned} \widehat{\nabla} J &= \hat{\mathrm{b}}(k)\,\mathrm{x}^T(k)\,\mathrm{w}_n(k) \\ &= \hat{\mathrm{b}}(k)\,y_n(k). \end{aligned} \tag{1.16}$$

According to our formulation, the constraint of $\mathbf{w}_n^T\mathbf{u}_n = 1$ sets $w_{nn} = 1$. This results in the $n$-th component of the input to be transferred directly to the output which is the distinguishing feature of the decorrelator structure. In order to obtain this constraint, we premultiply the gradient by $\mathbf{U}_n = \left(\mathbf{I} - \mathbf{u}_n\mathbf{u}_n^T\right)$ which zeros its $n$-th component. The decorrelator algorithm is then given simply by:

$$\begin{aligned} \mathrm{w}_n(k) &= \mathrm{w}_n(k-1) - \mu y_n(k)\,\mathbf{U}_n\hat{\mathrm{b}}(k) \\ w_{nn} &= 1 \end{aligned} \tag{4.17}$$

### 4.3.1 Convergence Analysis of LMS Decorrelator Algorithm

The LMS decorrelator has been shown to converge faster than the LMS error algorithm [11]. In this section we will prove, through the use of the eigenvalue spreads of both LMS algorithms, that the spread of the LMS decorrelator algorithm is smaller and converges faster than the LMS error algorithm. First we determine the necessary and sufficient conditions for convergence of the LMS decorrelator. Define $\hat{\mathbf{B}}_n = \left(\mathbf{I} - \mathbf{u}_n\mathbf{u}_n^T\right)\hat{\mathrm{b}}$, we can rewrite (4.17)

$$\begin{aligned} \mathrm{w}_n(k) &= \mathrm{w}_n(k-1) - \mu y_n(k)\,\hat{\mathbf{B}}_n(k) \\ &= \left[\mathbf{I} - \mu\hat{\mathbf{B}}_n(k)\,\mathrm{x}^T(k)\right]\mathrm{w}_n(k-1) \end{aligned} \tag{4.18}$$

We define the weight-error vector for the LMS decorrelating algorithm as:

$$\mathrm{e}(k) = \mathrm{w}_n(k) - \mathrm{w}_o, \tag{4.19}$$

similar to the one in [17]. Then rewriting (4.18) in terms of the weight-error vector results in

$$\mathrm{e}(k) = \mathrm{e}(k-1) - \mu y_n(k)\,\hat{\mathbf{B}}_n(k) \tag{4.20}$$

Recognizing that $y_n(k)\hat{\mathbf{B}}_n(k) = \hat{\mathbf{B}}_n(k)y_n(k) = \hat{\mathbf{B}}_n(k)\mathbf{x}^T(k)\mathbf{w}_n(k-1)$ ,we get

$$\begin{aligned}\mathbf{e}(k) &= \mathbf{e}(k-1) - \mu\hat{\mathbf{B}}_n(k)\mathbf{x}^T(k)\mathbf{w}_n(k-1) \\ &= \mathbf{e}(k-1) - \mu\hat{\mathbf{B}}_n(k)\mathbf{x}^T(k)(\mathbf{e}(k-1) - \mathbf{w}_o). \end{aligned} \tag{4.21}$$

Taking the expected value of both sides of (4.21) results in

$$\begin{aligned}E[\mathbf{e}(k)] &= E[\mathbf{e}(k-1)] \\ &\quad -\mu E\left[\hat{\mathbf{B}}_n(k)\mathbf{x}^T(k)\mathbf{e}(k-1)\right] - \mu E\left[\hat{\mathbf{B}}_n(k)\mathbf{x}^T(k)\mathbf{w}_o\right]. \end{aligned} \tag{4.22}$$

Implementing the assumptions of independence made in [17] between the weight vector and the input vector and subsequently the weight-error vector and the input vector, we can simplify (4.22) as follows:

$$\begin{aligned}E[\mathbf{e}(k)] &= E[\mathbf{e}(k-1)] - \mu\mathbf{R}_{\hat{B}_n x}E[\mathbf{e}(k-1)] - \mu\mathbf{R}_{\hat{B}_n x}\mathbf{w}_o \\ &= \left(\mathbf{I} - \mu\mathbf{R}_{\hat{B}_n x}\right)E[\mathbf{e}(k-1)] - \mu\mathbf{R}_{\hat{B}_n x}\mathbf{w}_o, \end{aligned} \tag{4.23}$$

where $\mathbf{R}_{\hat{B}_n x} = E\left[\hat{\mathbf{B}}_n(k)\mathbf{x}^T\right]$ . Comparing the expected value of the weight-error vector equation in [17] to (4.23), we observe that they both are of the same mathematical form except for the constant $\mu\mathbf{R}_{\hat{B}_n x}\mathbf{w}_o$. Therefore, the convergence condition of the LMS decorrelator would seem to be dependent on the eigenvalues of the $\mathbf{R}_{\hat{B}_n x}$. However, since the $n$-th row of the correlation matrix $\mathbf{R}_{\hat{B}_n x}$ consists of all zeros, the convergence condition is actually dependent on the eigenvalues of the correlation matrix $\mathbf{R}_{\hat{B}' x}$ as defined in (3.21). Subsequently, the mean of the weight-error vector $\mathbf{e}(k)$ converges to zero as $n$ approaches infinity, provided that the following condition holds:

$$0 < \mu < \frac{2}{\lambda_{\max}\left(\mathbf{R}_{\hat{B}' x}\right)} \tag{4.24}$$

where $\lambda_{\max}\left(\mathbf{R}_{\hat{B}' x}\right)$ is the largest eigenvalue of $\mathbf{R}_{\hat{B}' x}$.

To analyze the convergence speed, we will estimate the eigenvalue spread for the LMS decorrelator and compare it with the spread for the LMS error. Using

the assumptions stated in [17] for the LMS error algorithm analysis, we can rewrite (4.18) as:

$$\mathrm{w}_n(k) = \left[\mathbf{I} - \mu \mathbf{R}_{\widehat{B}_n x}\right] \mathrm{w}_n(k-1). \tag{4.25}$$

Then taking the expected value of both sides of (4.25), we get

$$E[\mathrm{w}_n(k)] = \left[\mathbf{I} - \mu \mathbf{R}_{\widehat{B}x}\right] E[\mathrm{w}_n(k-1)] \tag{4.26}$$

$$= \left[\mathbf{I} - \mu \mathbf{U}_n \mathbf{E} \mathbf{A}^T\right] E[\mathrm{w}_n(k-1)] \tag{4.27}$$

where we have defined $\mathbf{U}_n = \left(\mathbf{I} - \mathrm{u}_n \mathrm{u}_n^T\right)$, the identity matrix with the $n$-th element zeroed, and used the relations

$$\mathbf{R}_{\widehat{B}_n x} = \mathbf{U}_n \mathbf{R}_{\widehat{b}x} = \mathbf{U}_n E\left[\hat{\mathrm{b}}\left(\mathrm{b}^T \mathbf{E} \mathbf{A}^T + \mathrm{v}\right)\right] = \mathbf{U}_n E\left[\hat{\mathrm{b}}\mathrm{b}^T\right] \mathbf{E}\mathbf{A}^T = \mathbf{U}_n \mathbf{E}\mathbf{A}^T. \tag{4.28}$$

For example, when $N = 2$, $U_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$, and the LMS decorrelator algorithm for controlling $\mathrm{w}_1^T = [w_{11}, w_{12}]$ consists of two scalar and uncoupled relations: the fixed weight $w_{11}(k) = 1$ and the equation $w_{12}(k) = w_{12}(k-1) - \mu y_1(k) \hat{b}_2(k)$. Consequently, for this example, the LMS decorrelator algorithm is faster than its LMS error counterpart.

Without loss of generality, consider the separation of source $n = 1$, from the other sources. Define the following matrix and vector partitions:

$$\mathrm{w}_1^T = \begin{pmatrix} w_{11} & \vdots & \mathbf{W}^T \end{pmatrix}, \ \mathbf{I} = \begin{pmatrix} 1 & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & \mathbf{I}' \end{pmatrix} \tag{4.29}$$

$$\hat{\mathbf{B}}_1^T = \begin{pmatrix} 0 & \vdots & \hat{\mathbf{B}}'^T \end{pmatrix}, \ \mathrm{x}^T = \begin{pmatrix} x_1 & \vdots & \mathbf{X}^T \end{pmatrix} \tag{4.30}$$

where $\mathbf{W} = [w_{12}, w_{13}, \cdots, w_{1N}]^T$, $\mathbf{I}'$ is an $(N-1) \times (N-1)$ identity matrix, $\hat{\mathbf{B}}'^T = \left[\hat{b}_2, \hat{b}_3, \cdots, \hat{b}_N\right]^T$ and $\mathbf{X}^T = [x_2, x_3, \cdots, x_N]^T$. Consequently we can write equation (4.18) as follows:

$$\begin{pmatrix} w_1(k) \\ \cdots \\ \mathbf{W}(k) \end{pmatrix} = \left[ \begin{pmatrix} 1 & \vdots & 0 \\ \cdots & \vdots & \cdots \\ 0 & \vdots & \mathbf{I}' \end{pmatrix} - \mu \begin{pmatrix} 0 \\ \cdots \\ \hat{\mathbf{B}}'(k) \end{pmatrix} \begin{pmatrix} x_1(k) & \vdots & \mathbf{X}^T(k) \end{pmatrix} \right] \begin{pmatrix} w_1(k-1) \\ \cdots \\ \mathbf{W}(k-1) \end{pmatrix} \tag{4.31}$$

$$= \begin{pmatrix} w_1(k-1) \\ \cdots \\ \left[\mathbf{I}' - \mu \hat{\mathbf{B}}'(k) X^T(k)\right] \mathbf{W}(k-1) - \mu \hat{\mathbf{B}}'(k) x_1(k) w_1(k-1) \end{pmatrix}$$

Taking expected value of both sides of (4.31) we get

$$E[\mathbf{W}(k)] = \left[\mathbf{I}' - \mu \mathbf{R}_{\hat{B}'x}\right] E[\mathbf{W}(k-1)] - \mu E\left[\hat{\mathbf{B}}'(k) x_1(k) w_1(k-1)\right]. \tag{4.32}$$

We deduce that the convergence speed of the LMS decorrelator is controlled by the spread of the eigenvalues of $\mathbf{R}_{\hat{B}'x}$.

*Proposition 1: The following relation exists between the largest eigenvalues of* $\mathbf{R}_{\hat{B}'x}$ *and* $\mathbf{R}_{\hat{b}x}$:

$$\lambda_{\max}\left(\mathbf{R}_{\hat{b}x}\right) \geq \lambda_{\max}\left(\mathbf{R}_{\hat{B}'x}\right) \tag{4.33}$$

*Proof:* The matrix $\mathbf{R}_{\hat{B}'x}$ can be expressed as a partition of the matrix $\mathbf{R}_{\hat{b}x}$:

$$\mathbf{R}_{\hat{b}x} = \mathbf{E}\mathbf{A}^T = \begin{pmatrix} \sqrt{\xi_1} & \vdots & \mathbf{r}_1^T \\ \cdots & \vdots & \cdots \\ \mathbf{r}_1 & \vdots & \mathbf{R}_{\hat{B}'x} \end{pmatrix}. \tag{4.34}$$

Define the ratio

$$\rho(z) = \frac{\mathbf{z}^T \mathbf{R}_{\hat{b}x} \mathbf{z}}{\mathbf{z}^T \mathbf{z}} \tag{4.35}$$

where $\mathbf{z}$ is the eigenvector of the eigenvalue $\lambda\left(\mathbf{R}_{\hat{b}x}\right)$. According to the Courant-Fisher Theorem [16],

$$\lambda_{\max}\left(\mathbf{R}_{\hat{b}x}\right) = \max_{\tilde{z}} \rho(z). \tag{4.36}$$

Expanding both numerator and denominator of (4.35), we get

$$\mathbf{z}^T \mathbf{R}_{\hat{b}x} \mathbf{z} = \mathbf{z}^T \begin{pmatrix} \sqrt{\xi_1} & \vdots & \mathbf{r}_1^T \\ \cdots & \vdots & \cdots \\ \mathbf{r}_1 & \vdots & \mathbf{R}_{\hat{B}'x} \end{pmatrix} \mathbf{z} \tag{4.37}$$

$$= \mathbf{z}^T \begin{pmatrix} \sqrt{\xi_1} z_1 + \mathbf{r}_1^T \mathbf{Z} \\ \mathbf{r}_1 z_1 + \mathbf{R}_{\hat{B}'x} \mathbf{Z} \end{pmatrix} \tag{4.38}$$

$$= \sqrt{\xi_1} z_1^2 + z_1 \mathbf{r}_1^T \mathbf{Z} + \mathbf{Z}^T \mathbf{r}_1 z_1 + \mathbf{Z}^T \mathbf{R}_{\hat{B}'x} \mathbf{Z}. \tag{4.39}$$

and

$$\mathbf{z}^T \mathbf{z} = z_1^2 + \mathbf{Z}^T \mathbf{Z}. \tag{4.40}$$

where $\mathbf{z} = \left[ z_1 : \mathbf{Z} \right]^T$ and $\mathbf{Z} = [z_2, z_3, \cdots, z_N]^T$. Then substituting (4.37) and (4.40) back into (4.36), we get

$$\lambda_{\max} \left( \mathbf{R}_{\hat{b}x} \right) = \max_{\tilde{z}} \frac{\sqrt{\xi_1} z_1^2 + z_1 \mathbf{r}_1^T \mathbf{Z} + \mathbf{Z}^T \mathbf{r}_1 z_1 + \mathbf{Z}^T \mathbf{R}_{\hat{B}'x} \mathbf{Z}}{z_1^2 + \mathbf{Z}^T \mathbf{Z}} \tag{4.41}$$

Next, we maximize the right side of (4.41) over the restriction $z_1 = 0$ and get

$$\begin{aligned} \max_{z_1 = 0} \rho(z) &= \max_{Z} \frac{\mathbf{Z}^T \mathbf{R}_{\hat{B}'x} \mathbf{Z}}{\mathbf{Z}^T \mathbf{Z}} \\ &= \lambda_{\max} \left( \mathbf{R}_{\hat{B}'x} \right). \end{aligned} \tag{4.42}$$

where $\lambda_{\max} \left( \mathbf{R}_{\hat{B}'x} \right)$ is the maximum eigenvalue of the partition matrix $\mathbf{R}_{\hat{B}'x}$. Consequently, we can rewrite (4.36) as:

$$\lambda_{\max} \left( \mathbf{R}_{\hat{b}x} \right) \geq \lambda_{\max} \left( \mathbf{R}_{\hat{B}'x} \right). \tag{4.43}$$

where $\geq$ is due to the restriction placed on $z_1$.

*Proposition 2: The following relation exists between the smallest eigenvalues of* $\mathbf{R}_{\hat{B}'x}$ *and* $\mathbf{R}_{\hat{b}x}$:

$$\lambda_{\min} \left( \mathbf{R}_{\hat{b}x} \right) \leq \lambda_{\min} \left( \mathbf{R}_{\hat{B}'x} \right) \tag{4.44}$$

*Proof:* According to a corollary to the Courant-Fisher Theorem known as the Interlacing Property [16], and noting that $\mathbf{R}_{\hat{B}'x}$ is the $(N-1) \times (N-1)$ leading

principal submatrix of $\mathbf{R}_{\hat{b}x}$, we have:

$$\lambda_{\min}\left(\mathbf{R}_{\hat{b}x}\right) = \lambda_N\left(\mathbf{R}_{\hat{b}x}\right) \leq \lambda_{N-1}\left(\mathbf{R}_{\hat{B}'x}\right) = \lambda_{\min}\left(\mathbf{R}_{\hat{B}'x}\right) \tag{4.45}$$

*Proposition 3: If the noise is negligible then the eigenvalues of* $\mathbf{R}_x$ *are equal to the square of the eigenvalues of* $\mathbf{R}_{\hat{b}x}$.

*Proof:* We have the relation $\mathbf{R}_{\hat{b}x} = \mathbf{E}\mathbf{A}^T$, and for negligible noise we have $\mathbf{R}_x = \mathbf{A}\mathbf{E}\mathbf{E}\mathbf{A}^T$. Consequently, $\mathbf{R}_x = \mathbf{R}_{\hat{b}x}^T\mathbf{R}_{\hat{b}x}$. It follows that $\lambda_i(\mathbf{R}_x) = \lambda_i^2\left(\mathbf{R}_{\hat{b}x}\right)$ for $1 \leq i \leq N$. In particular,

$$\chi(\mathbf{R}_x) = \frac{\lambda_{\max}(\mathbf{R}_x)}{\lambda_{\min}(\mathbf{R}_x)} = \left(\frac{\lambda_{\max}\left(\mathbf{R}_{\hat{b}x}\right)}{\lambda_{\min}\left(\mathbf{R}_{\hat{b}x}\right)}\right)^2 = \chi^2\left(\mathbf{R}_{\hat{b}x}\right) \tag{4.46}$$

*Proposition 4: The eigenvalue spread* $\chi(\mathbf{R}_x)$ *associated with the LMS error algorithm, is larger than the square of the spread* $\chi\left(\mathbf{R}_{\hat{B}'x}\right)$ *associated with the LMS decorrelator provided the noise is negligible.*

*Proof:* From *Propositions* 1,2 and 3 we have:

$$\chi(\mathbf{R}_x) = \chi^2\left(\mathbf{R}_{\hat{b}x}\right) \geq \chi^2\left(\mathbf{R}_{\hat{B}'x}\right) \tag{4.47}$$

*Proposition 4* provides the explanation why the LMS decorrelator algorithm is faster than the LMS error algorithm.

*Proposition 5: The upper bound of the LMS error algorithm step-size parameter* $\mu_{\max}(\mathbf{R}_x)$ *is smaller than or equal to one half the square of the upper bound of the LMS decorrelator* $\mu_{\max}\left(\mathbf{R}_{\hat{B}'x}\right)$*:*

$$\mu_{\max}(\mathbf{R}_x) \leq \frac{1}{2}\mu_{\max}^2\left(\mathbf{R}_{\hat{B}'x}\right), \tag{4.48}$$

for the same values of the input vector $\mathbf{x}$.

*Proof:* From *Proposition 1* and *3*, it follows that

$$\begin{aligned}\mu_{\max}(\mathbf{R}_x) &= \frac{2}{\lambda_{\max}^2(\mathbf{R}_x)} \\ &= \frac{2}{\lambda_{\max}^2\left(\mathbf{R}_{\hat{b}x}\right)}\end{aligned}$$

$$\leq \frac{2}{\lambda^2_{\max}\left(\mathbf{R}_{\widehat{B}'x}\right)}$$
$$\leq \frac{1}{2}\mu^2_{\max}\left(\mathbf{R}_{\widehat{B}'x}\right) \tag{4.19}$$

## 4.4 RLS Decorrelator

We introduce a new decorrelating RLS type algorithm referred to as the *RLS decorrelator*. Paralleling the development of the RLS algorithm from the MSE weight vector in (3.3), we developed the following RLS algorithm for the decorrelator weight vector in (3.26).

First we recall the optimum weight vector equation for the LMS decorrelator given by:

$$\mathbf{w}_n = \frac{1}{2}\beta\mathbf{R}^{-1}_{\widehat{bx}}\mathbf{u}_n. \tag{4.50}$$

where $\beta = 2\left(\mathbf{u}_n^T\mathbf{R}^{-1}_{\widehat{bx}}\mathbf{u}_n\right)^{-1}$, and the recursive equation for the cross-correlation matrix between the estimated output $\widehat{\mathbf{b}}$ and the input vector $\mathbf{x}$ is given by:

$$\mathbf{R}_{\widehat{bx}}(k) = (1-\alpha)\,\mathbf{R}_{\widehat{bx}}(k-1) + \alpha\widehat{\mathbf{b}}(k)\,\mathbf{x}^T(k). \tag{4.51}$$

where $\alpha$ is the forgetting factor. As in the case of the RLS error algorithm's derivation, we implement the Matrix Inversion Lemma and determine the inverse of $\mathbf{R}_{\widehat{bx}}(k)$ which results in

$$\begin{aligned}\mathbf{R}^{-1}_{\widehat{bx}}(k) &= \frac{1}{1-\alpha}\mathbf{R}^{-1}_{\widehat{bx}}(k-1) \\ &\quad - \frac{\frac{1}{1-\alpha}\mathbf{R}^{-1}_{\widehat{bx}}(k-1)\,\widehat{\mathbf{b}}(k)\,\mathbf{x}(k)^T\,\frac{1}{1-\alpha}\mathbf{R}^{-1}_{\widehat{bx}}(k-1)}{\frac{1}{\alpha} + \widehat{\mathbf{b}}(k)^T\,\frac{1}{1-\alpha}\mathbf{R}^{-1}_{\widehat{bx}}(k-1)\,\widehat{\mathbf{b}}(k)} \\ &= \frac{1}{1-\alpha}\left[\mathbf{R}^{-1}_{\widehat{bx}}(k-1) - \mathbf{g}(k)\,\mathbf{x}(k)^T\,\mathbf{R}^{-1}_{\widehat{bx}}(k-1)\right]\end{aligned} \tag{4.52}$$

where as before $\mathbf{g}(k)$ is the variable gain vector defined as:

$$\mathbf{g}(k) = \frac{\mathbf{R}^{-1}_{\widehat{bx}}(k-1)\,\widehat{\mathbf{b}}(k)}{\frac{1-\alpha}{\alpha} + \mathbf{x}(k)^T\,\mathbf{R}^{-1}_{\widehat{bx}}(k-1)\,\widehat{\mathbf{b}}(k)} \tag{4.53}$$

Then, we substitute (4.52) into the optimum weight vector equation and get:

$$\mathbf{v}_n(k) = \frac{\beta}{1-\alpha}\left[\hat{\mathbf{R}}_{bx}^{-1}(k-1)\,\mathbf{u}_n - \mathbf{g}(k)\,\mathbf{x}^T(k)\,\hat{\mathbf{R}}_{bx}^{-1}(k-1)\,\mathbf{u}_n\right]. \quad (4.54)$$

Recognizing the fact that $\hat{\mathbf{R}}_{bx}^{-1}(k-1)\,\mathbf{u}_n = \frac{\mathbf{w}_n(k-1)}{\beta}$ we can simplify (4.54) and obtain the recursive equation for the weight vector as

$$\mathbf{v}_n(k) = \frac{1}{1-\alpha}\left[\mathbf{w}_n(k-1) - \mathbf{g}(k)\,\mathbf{x}^T(k)\,\mathbf{w}_n(k-1)\right] \quad (4.55)$$

Additionally, we recall that the optimum weight vector equation was derived under the linear constraint that $\mathbf{w}_n^T\mathbf{u}_n = 1$. In order to adhere to this constraint, we normalize (4.55) and get

$$\mathbf{w}_n(k) = \frac{\mathbf{v}_n(k)}{\mathbf{u}_n^T\mathbf{v}_n(k)} \quad (4.56)$$

The RLS decorrelating algorithm is summarized as follows:

$$\begin{aligned}
\mathbf{g}(k) &= \frac{\hat{\mathbf{R}}_x^{-1}(k-1)\,\hat{\mathbf{b}}(k)}{\frac{1-\alpha}{\alpha} + \mathbf{x}^T(k)\,\hat{\mathbf{R}}_x^{-1}(k-1)\,\hat{\mathbf{b}}(k)} \\
\hat{\mathbf{R}}_x^{-1}(k) &= \frac{1}{1-\alpha}\left[\mathbf{I} - \mathbf{g}(k)\,\mathbf{x}^T(k)\right]\hat{\mathbf{R}}_x^{-1}(k-1) \\
\mathbf{v}_n(k) &= \mathbf{w}_n(k-1) - \mathbf{g}(k)\,\mathbf{x}^T(k)\,\mathbf{w}_n(k-1) \\
\mathbf{w}_n(k) &= \frac{\mathbf{v}_n(k)}{\mathbf{u}_n^T\mathbf{v}_n(k)}
\end{aligned}$$

## 4.5 Soft Decision Detection

The classical decision-directed approach to estimating the transmitted data bits $\mathbf{b}$ is to implement the signum function at the output of the adaptive filter. This is known as *hard* decision. Nowlan and Hinton [25] proposed a more complicated nonlinearity known as *soft* decision. Their new soft decision-based algorithm would converge in channels with twice the initial bit error rate for which the hard decision-based LMS algorithm was convergent. Consequently, we have applied this nonlinearity to the various algorithms studied above in order to see its effect on the convergence behavior

of the algorithms. While the derivation in [25] is based on the application of MSE algorithms, we generalize the derivation and apply it to the decorrelating algorithms as well. The derivation is performed in Appendix B.

# CHAPTER 5

# PERFORMANCE MEASURES ANALYSIS

In this chapter we analyze the measures used to evaluate the performance of the various algorithms. The main performance measures are the probability of error of detecting the transmitted data bits of the $n$-th user and the signal-to-noise and signal-to-signal ratios.

## 5.1 Probability of Error Analysis

We first begin by evaluating the probability bit error for the system model detailed in Chapter 2. Without loss of generality, we consider the case of finding the probability of error in the first user given a two user $N = 2$ environment for presentation purposes, and then generalize the results for the $n$-th user in an $N$ user environment. The probability of an error occurring at the estimated output $\widehat{b}_1$ is written as:

$$P_1(\epsilon) = P\left(\widehat{b}_1 \neq b_1, b_1 = 1\right) + P\left(\widehat{b}_1 \neq b_1, b_1 = -1\right). \tag{5.1}$$

where $P\left(\widehat{b}_1 \neq b_1, b_1 = 1\right)$ and $P\left(\widehat{b}_1 \neq b_1, b_1 = -1\right)$ are defined as the joint probability of $\widehat{b}_1 \neq b_1$ and $b_1 = \pm 1$. Applying Bayes Theorem to (5.1), we get

$$\begin{aligned} P_1(\epsilon) &= P\left(\widehat{b}_1 \neq b_1 | b_1 = 1\right) P(b_1 = 1) \\ &\quad + P\left(\widehat{b}_1 \neq b_1 | b_1 = -1\right) P(b_1 = -1) \\ &= P\left(\widehat{b}_1 = -1 | b_1 = 1\right) P(b_1 = 1) \\ &\quad + P\left(\widehat{b}_1 = 1 | b_1 = -1\right) P(b_1 = -1). \end{aligned} \tag{5.2}$$

Since the data bits can take on one of two equiprobable values, i.e. $b_n \in \{-1, 1\}$, $P(b_n = 1) = P(b_n = -1) = 0.5$,

$$P_1(\epsilon) = 0.5\left[P\left(\widehat{b}_1 = -1 | b_1 = 1\right) + P\left(\widehat{b}_1 = 1 | b_1 = -1\right)\right]. \tag{5.3}$$

and we can take the two terms in (5.1) to be equal and simplify the equation as follows:

$$P_1(e) = P\left(\widehat{b}_1 = 1 | b_1 = -1\right) \tag{5.4}$$

Substituting the value of $\widehat{b}_1 = \text{sgn}(y_1)$, for hard decision detection, in (5.4) results in

$$\begin{aligned} P_1(e) &= P(y_1 > 0 | b_1 = -1) \\ &= P\left(\mathbf{w}_1^T \mathbf{x} > 0 | b_1 = -1\right), \end{aligned} \tag{5.5}$$

where $y_1 = \mathbf{w}_1^T \mathbf{x}$. Expanding the term $\mathbf{w}_1^T \mathbf{x}$ results in:

$$\begin{aligned} P_1(e) &= P\left(\mathbf{w}_1^T (\mathbf{AEb} + \mathbf{v}) > 0 | b_1 = -1\right) \\ &= P\left(\mathbf{w}_1^T \mathbf{AEb} + \mathbf{w}_1^T \mathbf{v} > 0 | b_1 = -1\right) \\ &= P(t_{11} b_1 + t_{12} b_2 + \eta_1 > 0 | b_1 = -1) \\ &= P(-t_{11} + t_{12} b_2 + \eta_1 > 0) \\ &= P(\eta_1 > t_{11} - t_{12} b_2), \\ &= \sum_{b_2} P(\eta_1 > t_{11} - t_{12} b_2, b_2) \\ &= P(\eta_1 > t_{11} - t_{12} b_2 | b_2 = 1) P(b_2 = 1) \\ &\quad + P(\eta_1 > t_{11} - t_{12} b_2 | b_2 = -1) P(b_2 = -1) \\ &= 0.5 \left[P(\eta_1 > t_{11} - t_{12}) + P(\eta_1 t_{11} + t_{12})\right] \end{aligned} \tag{5.6}$$

where $t_{11}$ and $t_{12}$ are the two elements of the vector

$$\begin{aligned} \mathbf{t}_1 &= \mathbf{w}_1^T \mathbf{AE} \\ &= [t_{11}\ t_{12}] \end{aligned}$$

and $\eta_1$ is the filtered Gaussian noise.

Since $\eta_1$ is a Gaussian random variable with a density function of

$$f_\eta(z) = \frac{1}{\sqrt{2\pi}\sigma_\eta} \exp^{-\frac{z^2}{2\sigma_\eta^2}}, \tag{5.7}$$

we can express (5.6) in terms of (5.7) which results in

$$P_1(e) = 0.5\left[\frac{1}{\sqrt{2\pi}\sigma_\eta}\int_{t_{11}-t_{12}}^{\infty}\exp^{-\frac{z^2}{2\sigma_\eta^2}}dz + \frac{1}{\sqrt{2\pi}\sigma_\eta}\int_{t_{11}+t_{12}}^{\infty}\exp^{-\frac{z^2}{2\sigma_\eta^2}}dz\right]. \quad (5.8)$$

After some observation, we can express (5.8) in terms of the $Q$-function, which was defined earlier, as:

$$P_1(e) = 0.5\left[Q\left(\frac{t_{11}-t_{12}}{\sigma_\eta}\right) + Q\left(\frac{t_{11}+t_{12}}{\sigma_\eta}\right)\right] \quad (5.9)$$

$$= 0.5\left[Q\left(\frac{t_{11}-t_{12}}{\sigma_n\sqrt{\mathbf{w}_n^T\mathbf{H}\mathbf{w}_n}}\right) + Q\left(\frac{t_{11}+t_{12}}{\sigma_n\sqrt{\mathbf{w}_n^T\mathbf{H}\mathbf{w}_n}}\right)\right] \quad (5.10)$$

Generalizing now for the probability of error of the $n$th user in an $N$ user environment results in:

$$P_n(e) = 2^{1-N}\sum_{\substack{b\in\{-1,1\}^N \\ b_n=-1}} Q\left(\frac{t_{nn}-\sum_{j\neq n}^{N} t_{nj}b_j}{\sigma_n\sqrt{\mathbf{w}_n^T\mathbf{H}\mathbf{w}}}\right) \quad (5.11)$$

However, since the numerical analysis of the probability of error for the $N$ user case is difficult to track, we will only consider the two and four user cases.

## 5.2 SNR and SSR

Two measures utilized frequently in evaluating a system are the input signal-to-background noise ratio SNR and the input signal-to-signal ratio SSR. We will also use these measures and define them below. The signal-to-noise ratio of the $n$-th user is defined as the bit energy of the $n$-th user to the energy of the background Gaussian noise such as:

$$\mathrm{SNR}_n = \frac{\xi_n}{\sigma_n^2 g_{nn}^2} \quad (5.12)$$

where $\xi_n$ is the energy of the detected signal from the $n$-th user, $g_{nn}$ is the $nn$-th noise cross-correlation element of the noise matrix $\mathbf{H}$ defined between the outputs of the matched filters and $\sigma_n^2$ is the power of the additive white Gaussian background

noise. The signal-to-signal ratio of the $n$-th user is defined as the bit energy of the $n$-th user to the bit energy of the rest of the users such as:

$$\mathrm{SSR}_n = \frac{\xi_n}{a_{ij}^2 \xi_m} \tag{5.13}$$

where $n \neq m$ and $a_{ij}$ is the cross-correlation coefficient.

## 5.3 Analysis of the Figure of Merit for the Convergence Plots

Convergence regions for each algorithm are determined using the approach suggested in [25]. There a figure of merit $\gamma$ is defined that relates the initial and final probabilities of error, $P_{ei}$ and $P_{ef}$ respectively. $\gamma$ was defined as

$$\gamma = 1 - \frac{P_{ef}}{P_{ei}}. \tag{5.14}$$

$P_{ei}$ is determined from the initial weight vector and $P_{ef}$ is determined from the resulting weight vector after a predetermined number of iterations. The SNR is varied from a negative value to a positive value so that the initial probabilities of error are indicated on the abscissa. The point of these curves is to demonstrate, after any given number of iterations and initial probabilities of error, where the algorithm stops converging. When $P_{ef} \ll P_{ei}$, $\gamma \approx 1$ and the algorithm always converges. When $P_{ef} = P_{ei}$, $\gamma = 0$, no convergence occurs.

# CHAPTER 6

## SIMULATION RESULTS

In this chapter, we present the computer simulations performed to verify the theoretical performance of the RLS decorrelating algorithm in comparison to the other algorithms. In order to take into account all the variables discussed previously, we have performed simulations with two and four users to show how the algorithm's performance varies with the number of users. Also, we have implemented both *hard* and *soft* decisions to show how the decision device used in a decision-directed mode effects the algorithm's performance. We focus on three basic types of plots. First, the *Learning Curve* plots to demonstrate the speed of the various algorithms simultaneously at various levels of interference powers. Secondly, the *Convergence Region* plots to demonstrate the regions at which the various algorithms cease to converge and their performance upto that point. Lastly, *Probability of Error vs.* SSR plots to demonstrate the performance of the various algorithms over a range of realistic operating SSR's. Additionally, we prove via computer simulations the theoretical results concerning the speed of convergence of the LMS decorrelator in comparison to the LMS error algorithm.

In all simulations to follow, the initial $n$-th user weight vector $\mathbf{w}_n(0)$ for each algorithm was set so that $w_{nn} = 1$ and $w_{nj} = 0$ for $j \neq n$. Also, the inital value of the inverse correlation matrices for the RLS type algorithms was set to $\mathbf{R}_x^{-1}(0) = \mathbf{R}_{\hat{b}x}(0)^{-1} = \frac{1}{\delta}\mathbf{I}$ where $\delta$ is a small constant. Finally, the legend implemented in Figures 6.1 through 6.28 denoting the performance of the four algorithms under various conditions is reproduced here to facilitate the analysis of the plots.

Table 6.1 Legend corresponding to Figures 6.1 - 6.28.

| ALGORITHM | NOTATION | LINE TYPE |
|---|---|---|
| RLS Decorrelator | RLS-dec | ——— |
| LMS Decorrelator | LMS-dec | . . . . . |
| RLS Error | RLS-err | - - - - - |
| LMS Error | LMS-err | - . - . - . - . - |

## 6.1 Learning Curves

One important issue in the development of adaptive algorithms is their speed of convergence to their steady state value. Here, we utilized the probability of error of the first user in a two and four user environment to show the convergence rates of the various algorithms.

### 6.1.1 Two Users-Hard Decision

In this section, we analyze the two user environment implementing hard decision detection at three SSR's: -5,-10,-15dB. The SNR is set at 8dB, which is a practical and realistic setting for most applications. The adaptive process is performed over 500 iterations and 10 independent trials. From Figure 6.1 for a SSR=-5dB, we see that all four algorithms converge to the approximately the same steady state value with the two RLS type algorithms converging slightly faster. In Figure 6.2 with SSR=-10dB, we note that the two LMS type algorithms start to converge slower, while the RLS type algorithms' behavior is unchanged. The increase in interference power has noticeably affected the two LMS type algorithms.

In Figure 6.3, for a SSR=-15dB, the LMS error algorithm completely diverges with further increases in the interference power making it unreliable in high interference environments. The LMS decorrelator converges, but at a slower rate compared to the RLS type algorithms.

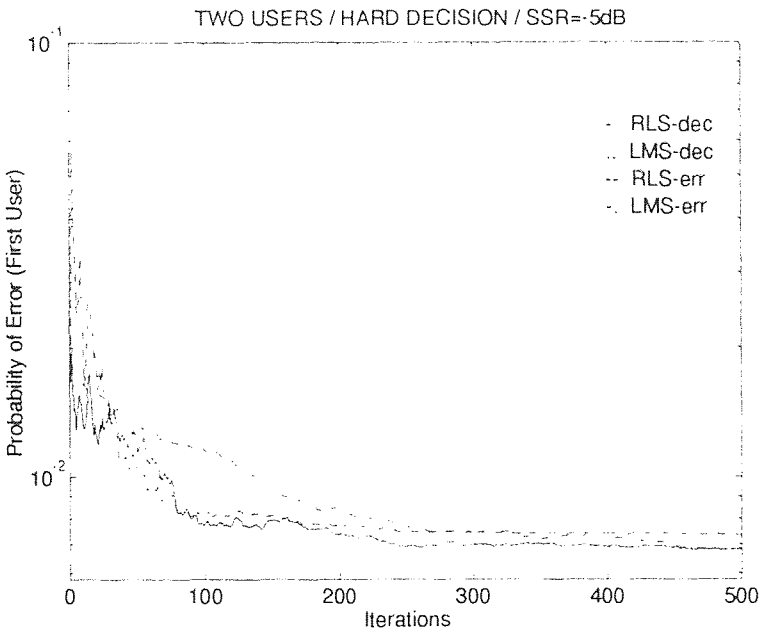


**Figure 6.1** Learning curve of the probability of error of the first user (Hard Decision,SNR=8dB,SSR=-5dB,N=2,$a_{ij}$=0.15)

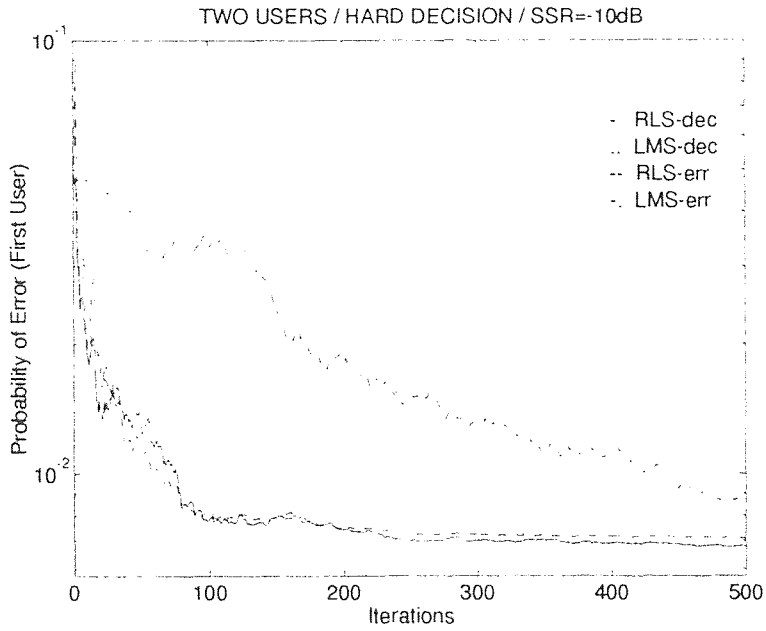


**Figure 6.2** Learning curve of the probability of error of the first user (Hard Decision,SNR=8dB,SSR=-10dB,N=2,$a_{ij}$=0.15)

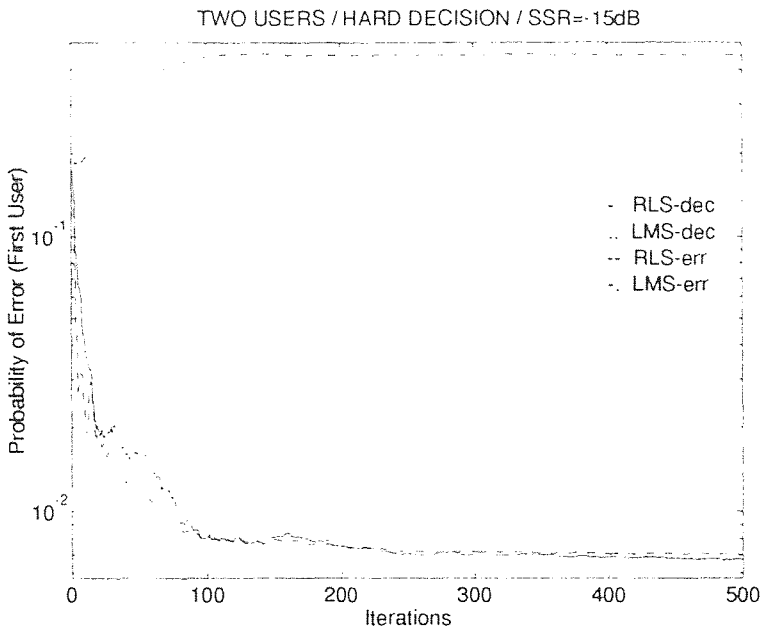


**Figure 6.3** Learning curve of the probability of error of the first user (Hard Decision,SNR=8dB,SSR=-15dB,N=2,$a_{ij}$=0.15)

### 6.1.2 Two Users-Soft Decision

In this section, we perform the same simulations as in Section 6.1.1 but implementing soft decision detection. For the SSR=-5dB, convergence speed has decreased for all four algorithms which implies that for the two user case, hard decision detection is more efficient as can be seen in Figure 6.4. Additionally, the steady state converging point of the RLS error algorithm has been increased. This can be attributed to the already less than desired performance of the RLS error algorithm at lower levels of interference due to its dependence on the forgetting factor $\alpha$, which introduces a bias in the estimations error equation [see Eq. (4.11)]. The forgetting factor was chosen to work optimally at higher interference levels. Similarly, in Figures 6.5 and 6.6, we notice little improvement in all but the LMS error algorithm which shows some signs of convergence.

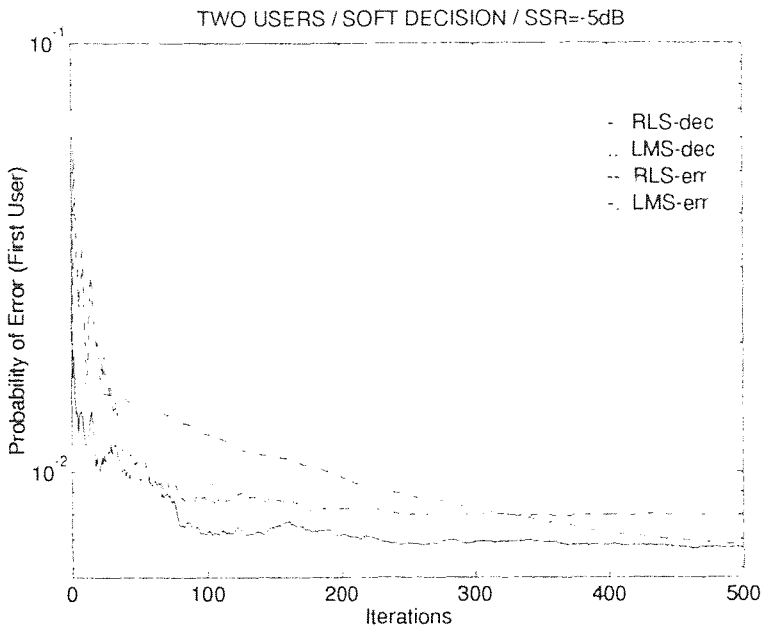


**Figure 6.4** Learning curve of the probability of error of the first user (Soft Decision, SNR=8dB,SSR=-5dB,N=2,$a_{ij}$=0.15)

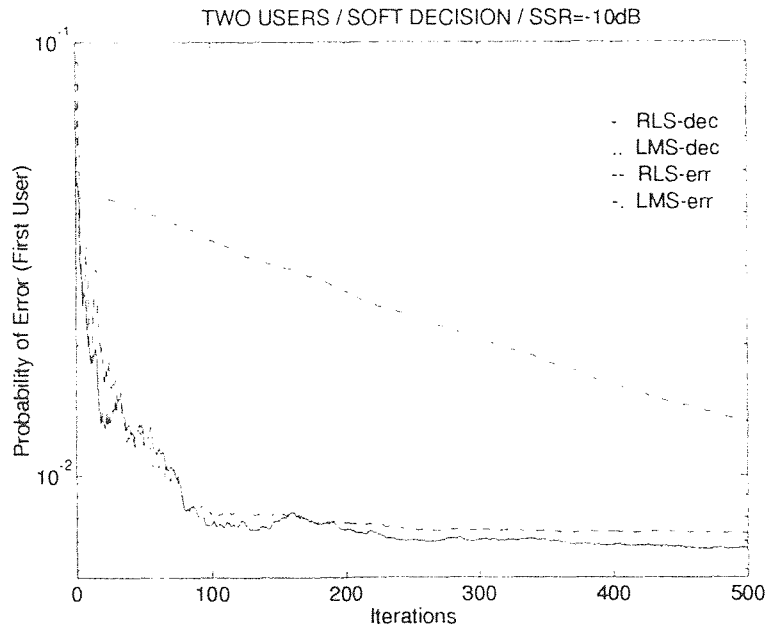


**Figure 6.5** Learning curve of the probability of error of the first user (Soft Decision, SNR=8dB,SSR=-10dB,N=2,$a_{ij}$=0.15)

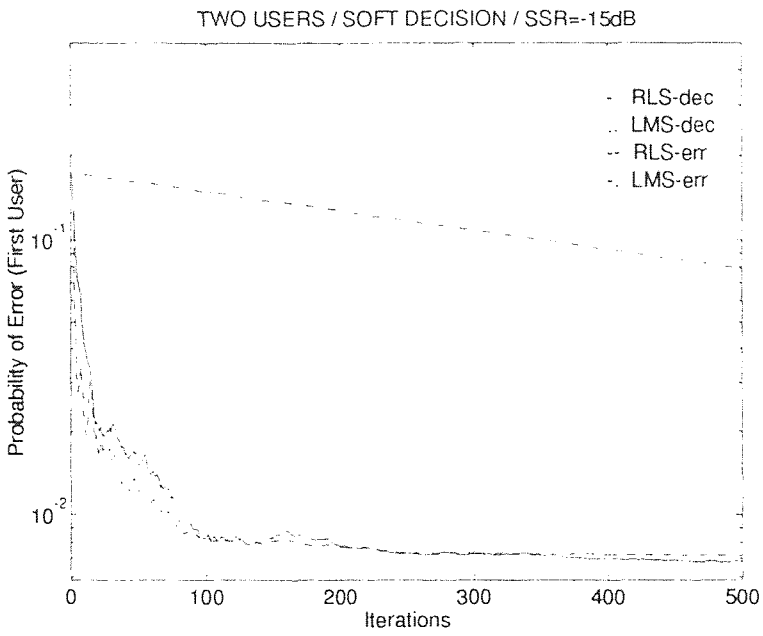


**Figure 6.6** Learning curve of the probability of error of the first user (Soft Decision, SNR=8dB,SSR=-15dB,N=2,$a_{ij}$=0.15)

### 6.1.3 Four Users-Hard Decision

In this section, the number of users is increased to exploit the robustness of the decorrelating algorithms to larger interference levels. While four users is not necessarily a practical application, it does provide some insight on the behavior of the algorithms. Again, we take the same SNR and SSR's as the previous two cases: SNR= $8dB$ and SSR= $-5,-10,-15dB$.

In Figure 6.7, we note the convergence for all four algorithms with the RLS type algorithms maintaining their edge in this measure of performance. In Figures 6.8 and 6.9, the increase in interference power has minimal detrimental effect on the performance of the all but the LMS error which clearly, as seen in the two user case, cannot cope with such levels of interference. The RLS type algorithms still outperform the LMS decorrelator in convergence speed, increasingly so with increases in the interference power. Additionally, the decorrelating algorithms behave more

smoothly with increases in interference power which is due the fact that the increased interference power from other signals assists the algorithm in decorrelating the desired signal from the others. So the stronger the other signals are easier it becomes to decorrelate them from the desired signal.

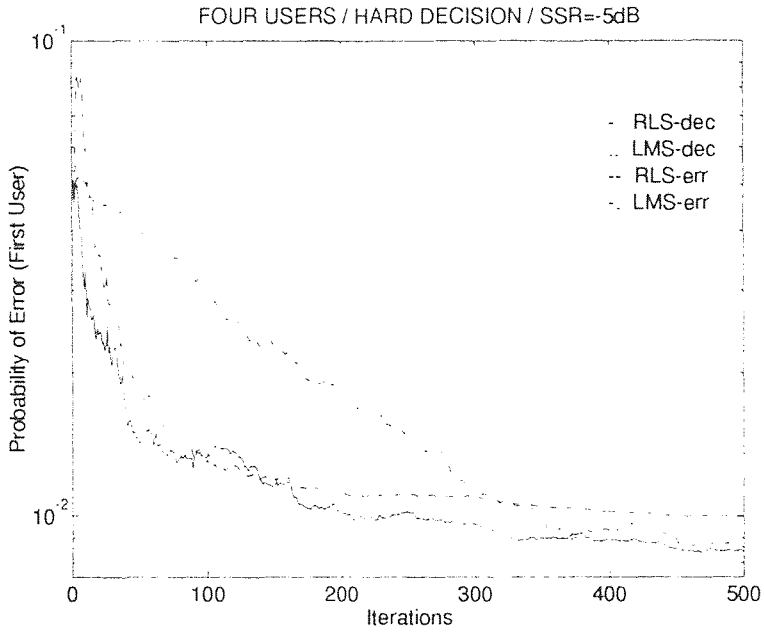


**Figure 6.7** Learning curve of the probability of error of the first user (Hard Decision,SNR=8dB,SSR=-5dB,N=4,$a_{ij}$=0.15)

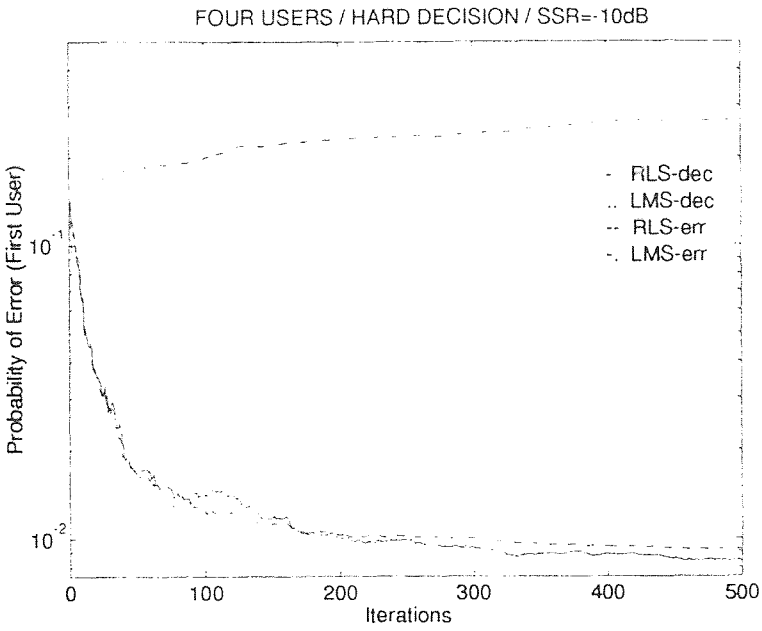


**Figure 6.8** Learning curve of the probability of error of the first user (Hard Decision,SNR=8dB,SSR=-10dB,N=4,$a_{ij}$=0.15)

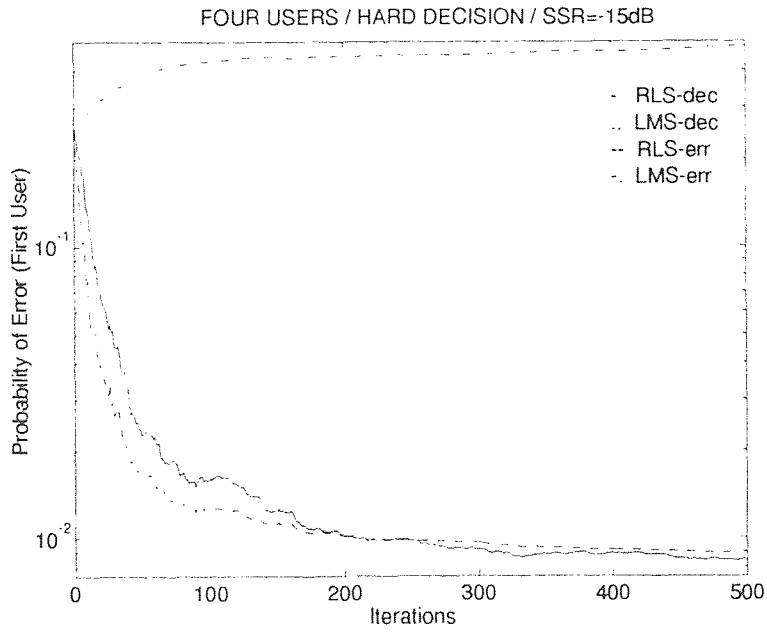


**Figure 6.9** Learning curve of the probability of error of the first user (Hard Decision,SNR=8dB,SSR=-15dB,N=4,$a_{ij}$=0.15)

### 6.1.4 Four Users-Soft Decision

The soft decision nonlinearity is applied to the four user case with all other parameters kept the same. From Figure 6.10, we notice the convergence of all four algorithms similar to previous simulations. However, soft decision detection has, as in the two user case for the same parameters, increased the steady state converging point of the RLS error algorithm. Again this is due to the selection of the forgetting factor $\alpha$ for optimization at high interference levels.

Increases in the interference levels brings down the steady state probability of error point for the RLS error algorithm. Additionally, two more observations can be made from Figures 6.11 and 6.12. The first is that the soft decision detection, as seen in Section 6.1.2, permits the convergence of the LMS error algorithm and secondly, it has minimal effect on the convergence of the other algorithms.

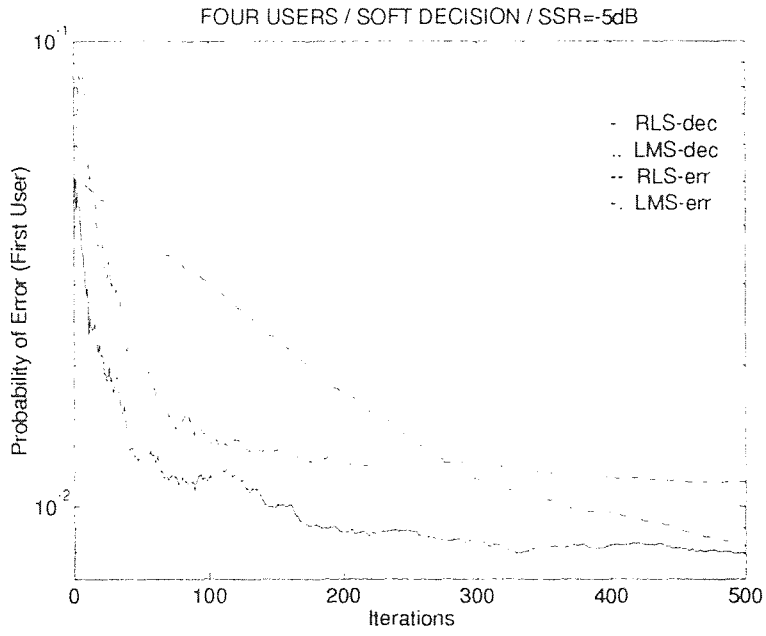


**Figure 6.10** Learning curve of the probability of error of the first user (Soft Decision, SNR=8dB,SSR=-5dB,N=4,$a_{ij}$=0.15)

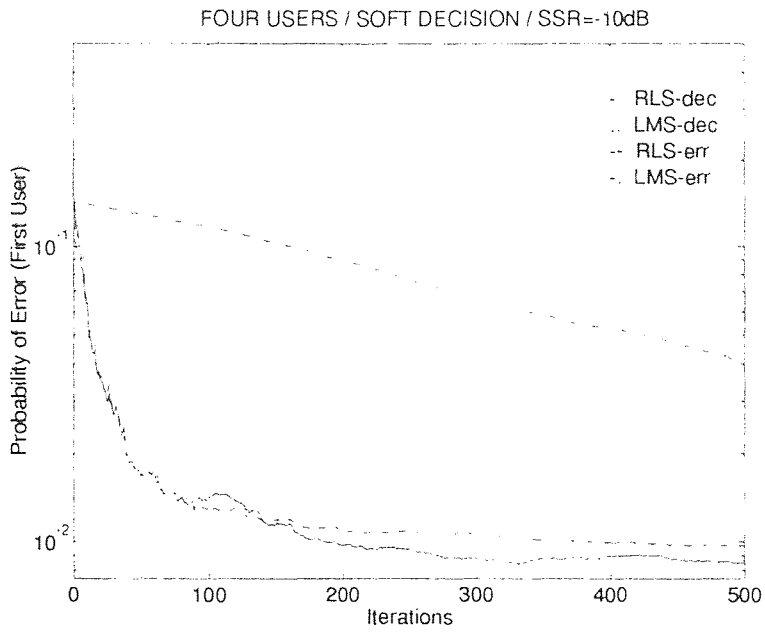


**Figure 6.11** Learning curve of the probability of error of the first user (Soft Decision, SNR=8dB,SSR=-10dB,N=4,$a_{ij}$=0.15)

In comparing all the simulation results of the convergence rate over the various SSR's for both decision methods, we can make some final observations. First, the algorithm of choice for both speed and robustness in the midst of a changing interference environment seems to be the RLS decorrelator. While the RLS error algorithm has comparable speed, in our unsupervised training version it responds uncharacteristically at lower interference levels mainly due to the choice of the forgetting factor $\alpha$. Secondly, the LMS error algorithm is more sensitive to changes in the interference levels, or equivalently eigenvalue spread, than the LMS decorrelator which supports our claims made earlier. Finally, the use of soft decision detection over hard decision detection proved to have minor improvement on the convergence of the algorithms except for the LMS error.

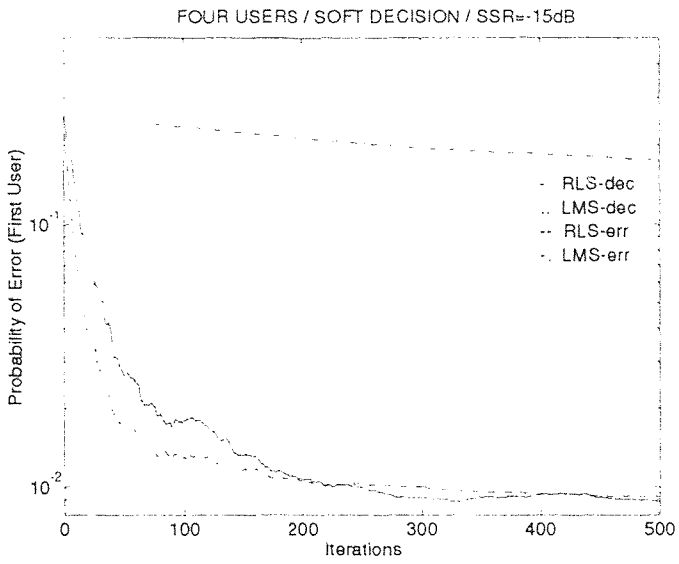


**Figure 6.12** Learning curve of the probability of error of the first user (Soft Decision, SNR=8dB,SSR=-15dB,N=4,$a_{ij}$=0.15)

## 6.2 Convergence Plots

The convergence plots are produced using the figure of merit analyzed in Section 5.3. In this case, 100 iterations and 10 independent trials were used. The SNR was varied from -10dB to +10dB. Also as before, we look at how the soft decision nonlinearity and different number of users affect the performance of the algorithms at the same three SSR's: -5, -10 and -15dB.

### 6.2.1 Two Users-Hard Decision

In Figure 6.13, we notice that all four algorithms converge with the LMS error slightly behind the others. In Figure 6.14 the LMS error has fallen behind considerably in its convergence region while the others maintain comparable performance and show significant movement in light of increasing interference levels. In Figure 6.15, the LMS error diverges completely while the others continue to widen their convergence regions with increased interference.

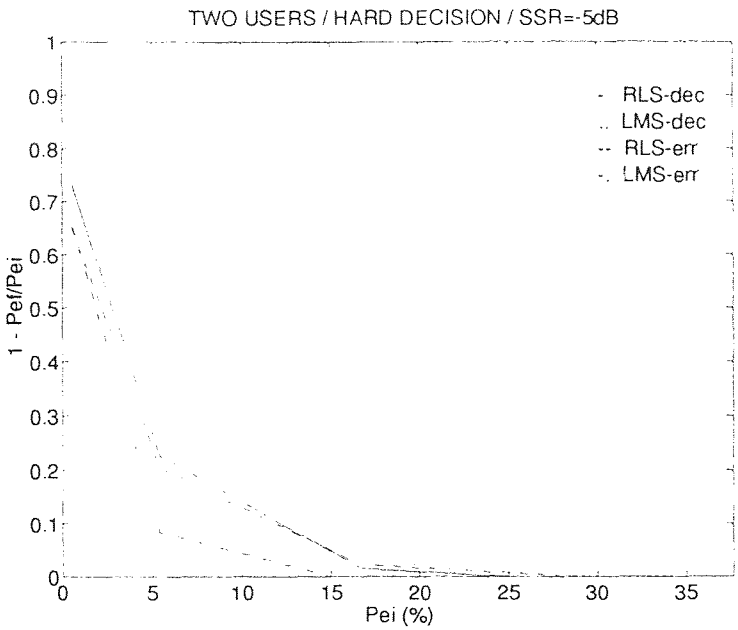


**Figure 6.13** Convergence regions after 100 iterations for the probability of error of the first user (Hard Decision,SSR=-5dB,N=2,$a_{ij}$=0.15)

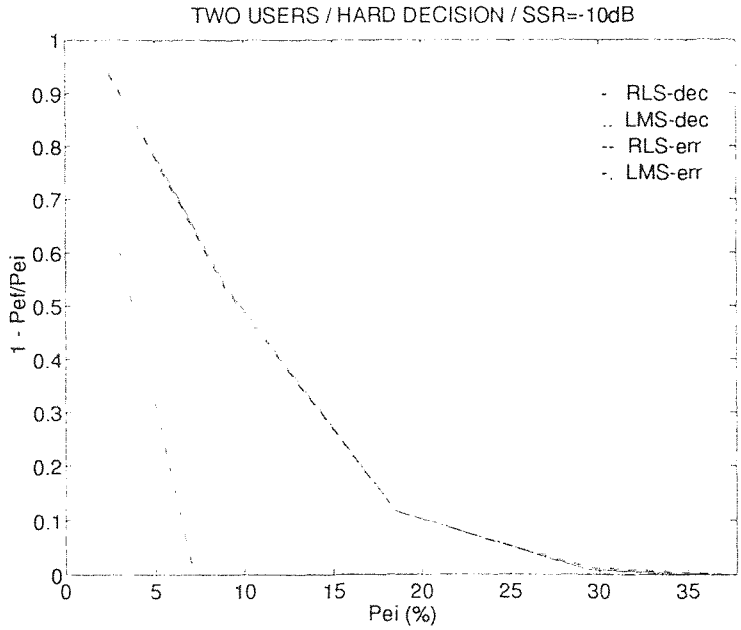


**Figure 6.14** Convergence regions after 100 iterations for the probability of error of the first user (Hard Decision,SSR=-10dB,N=2,$a_{ij}$=0.15)

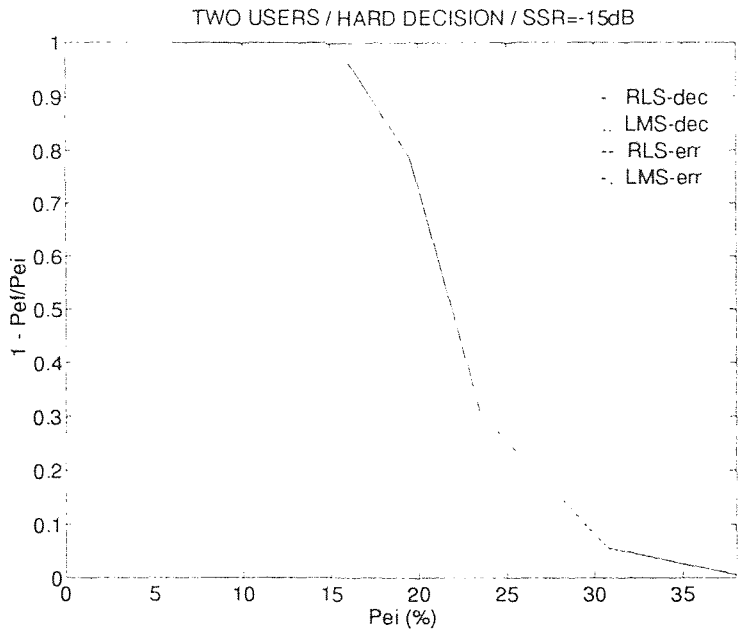


**Figure 6.15** Convergence regions after 100 iterations for the probability of error of the first user (Hard Decision,SSR=-15dB,N=2,$a_{ij}$=0.15)

### 6.2.2 Two Users-Soft Decision

The two decorrelating and RLS error algorithms show similar performance as seen in Figures 6.16, 6.17 and 6.18. The LMS error, however, demonstrates wider regions of convergence and in the latter two figures continues to show convergence in cases where it failed to do so with the hard decision. Here we can see the desired effects of the soft decision nonlinearity, at least in the case of the LMS error algorithm. It allowed the algorithm to converge where it otherwise would not.

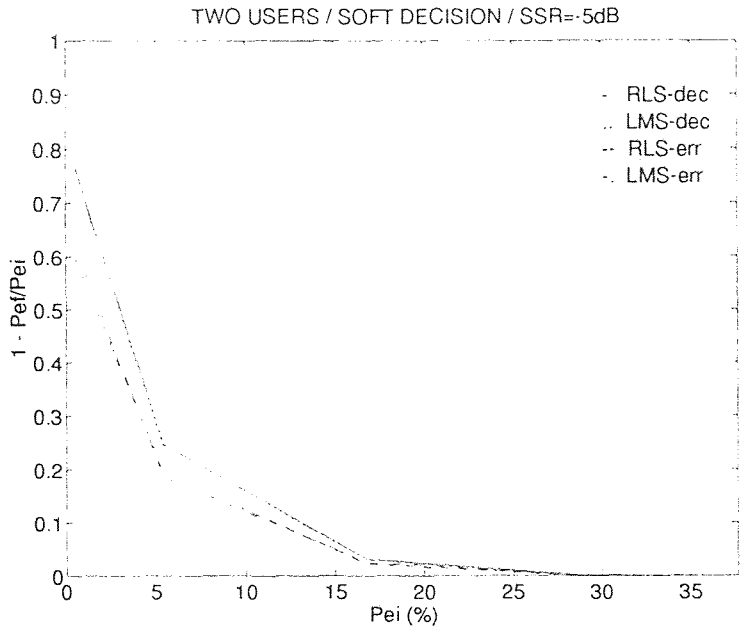


**Figure 6.16** Convergence regions after 100 iterations for the probability of error of the first user (Soft Decision,SSR=-5dB,N=2,$a_{ij}$=0.15)

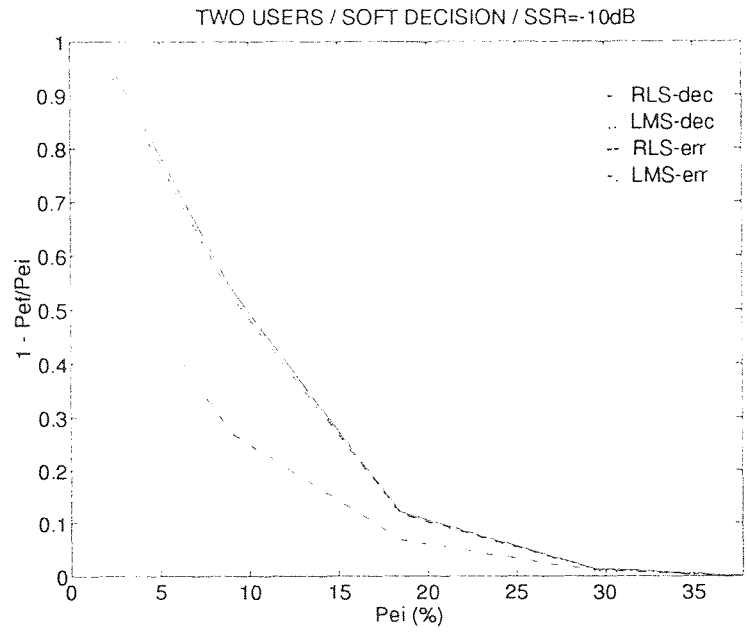


**Figure 6.17** Convergence regions after 100 iterations for the probability of error of the first user (Soft Decision,SSR=-10dB,N=2,$a_{ij}$=0.15)

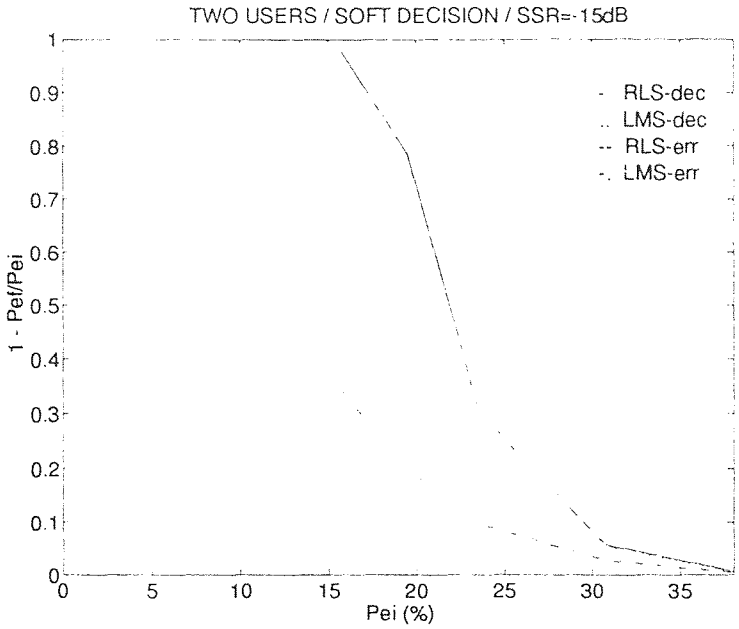


**Figure 6.18** Convergence regions after 100 iterations for the probability of error of the first user (Soft Decision,SSR=-15dB,N=2,$a_{ij}$=0.15)

### 6.2.3 Four Users-Hard Decision

With the additional two users, we observe performance behavior similar to the two user case coupled with the widening of the convergence regions for all four algorithms at each of the three interference levels. In Figure 6.19, all four algorithms converge with the LMS error again falling slightly behind. In the next two scenarios, Figures 6.20 and 6.21, the LMS error no longer converges while the others maintain their robustness. Additionally, we observe that the increase in the number of users has resulted in wider regions of convergence rather than smaller. This supports our claim that higher levels of distortion and interference aid the decorrelating algorithms in separating the desired signals from the others. The RLS error also demonstrates similar characteristics.

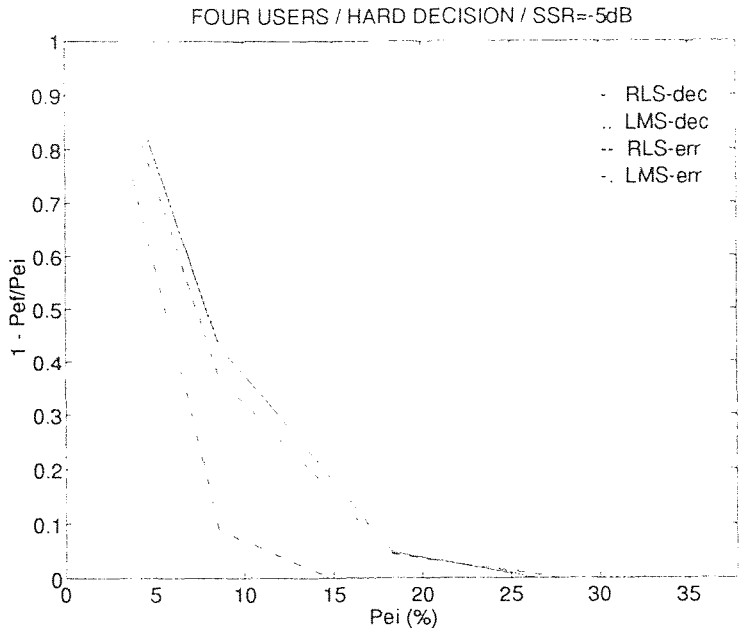


**Figure 6.19** Convergence regions after 100 iterations for the probability of error of the first user (Hard Decision,SSR=-5dB,N=4,$a_{ij}$=0.15)

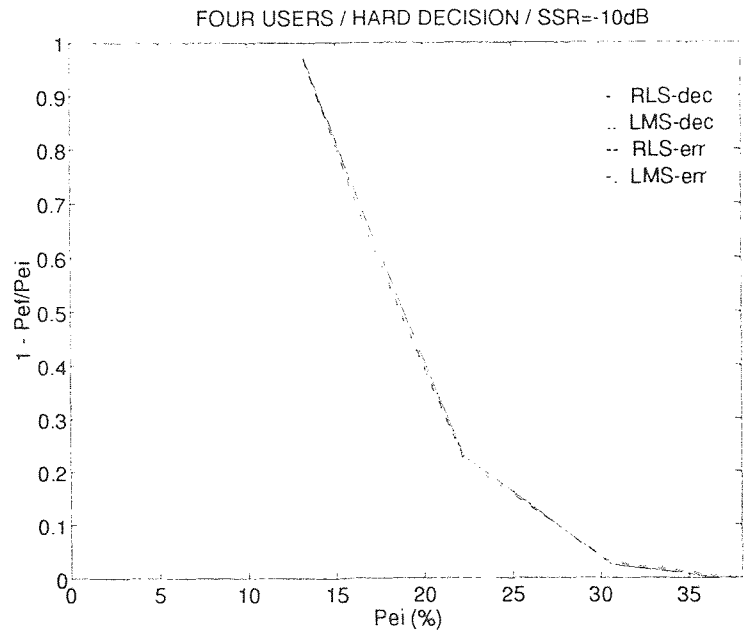


**Figure 6.20** Convergence regions after 100 iterations for the probability of error of the first user (Hard Decision,SSR=-10dB,N=4,$a_{ij}$=0.15)

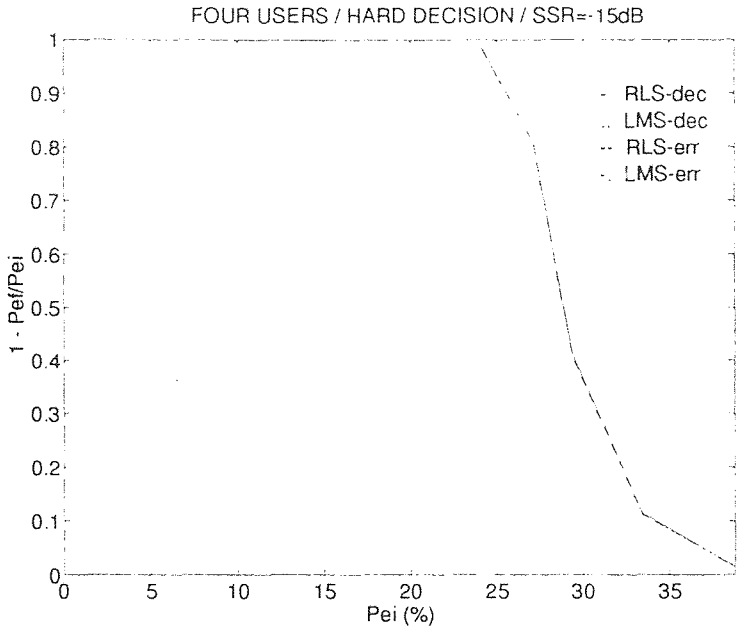


**Figure 6.21** Convergence regions after 100 iterations for the probability of error of the first user (Hard Decision,SSR=-15dB,N=4,$a_{ij}$=0.15)

### 6.2.4 Four Users-Soft Decision

Applying the soft decision nonlinearity to the four user case results in improved performance only for the LMS error algorithm as seen in Figures 6.22, 6.23 and 6.24. The others maintain their performance regardless of the nonlinearity used.

The convergence region curves proved to be useful in answering the two questions about the decorrelating algorithms' performance. First, the increase in the number of users assisted the algorithms in performing the separation of signals as expected. Secondly, the application of the soft decision nonlinearity does outperform the hard decision nonlinearity because it enabled the convergence of the LMS error algorithm in regions and interference levels it would not have converged otherwise.

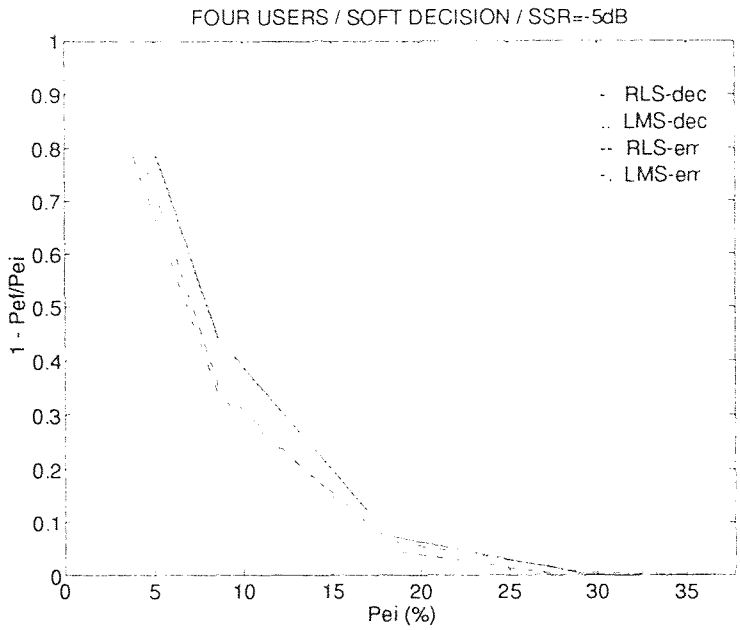


**Figure 6.22** Convergence regions after 100 iterations for the probability of error of the first user (Soft Decision,SSR=-5dB,N=4,$a_{ij}$=0.15)

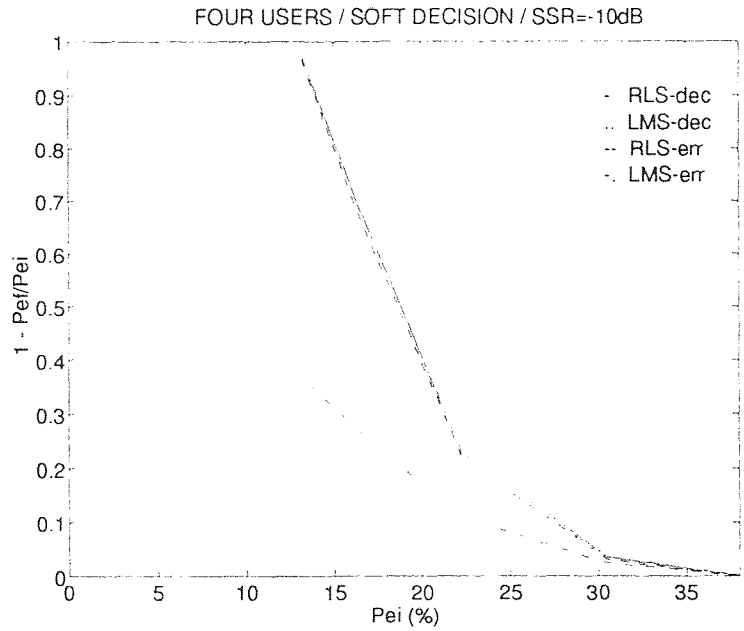


**Figure 6.23** Convergence regions after 100 iterations for the probability of error of the first user (Soft Decision,SSR=-10dB,N=4,$a_{ij}$=0.15)

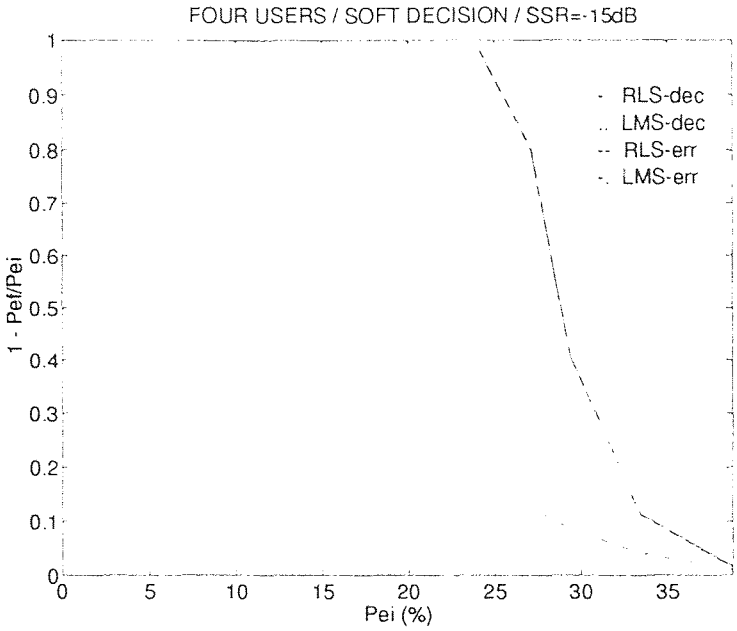


**Figure 6.24** Convergence regions after 100 iterations for the probability of error of the first user (Soft Decision,SSR=-15dB,N=4,$a_{ij}$=0.15)

## 6.3 Probability of Error Comparison of Algorithms

In this section, we examine the effects of varying the SNR of the other users with respect to the first user or in other words keeping the noise power constant and varying the SSR. Our main goal is to study further the behavior of the four algorithms under varying SSR's. We vary the difference in SNR between the first user and the other users from -10 to +10dB. The probability of error as always is with respect to the first user taken after 500 iterations and 10 independent trials. Additionally, we can compare the responses of the various algorithms to the response of an ideal decorrelator where the weight matrix, composed of the $N$ weight vectors, is equal to the inverse of the mixture matrix, i.e. $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N]^T = \mathbf{A}^{-1}$. The response of the ideal decorrelator is constant throughout the variations in SSR. Also, to prevent further cluttering of the plots with an additional line, the probability of

error of the first user for the ideal decorrelator is given below for the two and four users cases:

$$P_e\,(\text{two users}) = .0065$$

$$P_e\,(\text{four users}) = .0072$$

### 6.3.1 Two User Case

We analyze the hard decision nonlinearity first. In Figure 6.25, we observe some interesting behavior. First, the two decorrelating algorithms show unequaled robustness throughout the various SNR's as expected. Secondly, also as expected, the LMS error algorithm exhibits decreasing probability of error as $\text{SNR}_1$ increases beyond $\text{SNR}_2$. However, the RLS error algorithm shows uncharacteristic behavior at more positive SSR's. This can be attributed to the selection of the forgetting factor $\alpha$ to optimize performance at higher interference levels. The forgetting factor introduces a bias term into the estimations error equation of (4.11) which adversely affects the performance of the algorithm at more positive SSR's. This bias term is the disadvantage with our unsupervised RLS algorithm.

With the application of the soft decision nonlinearity, the MSE algorithms behave similarly but with greater discrepancies between their performance and the decorrelating algorithms' performance as shown in Figure 6.26.

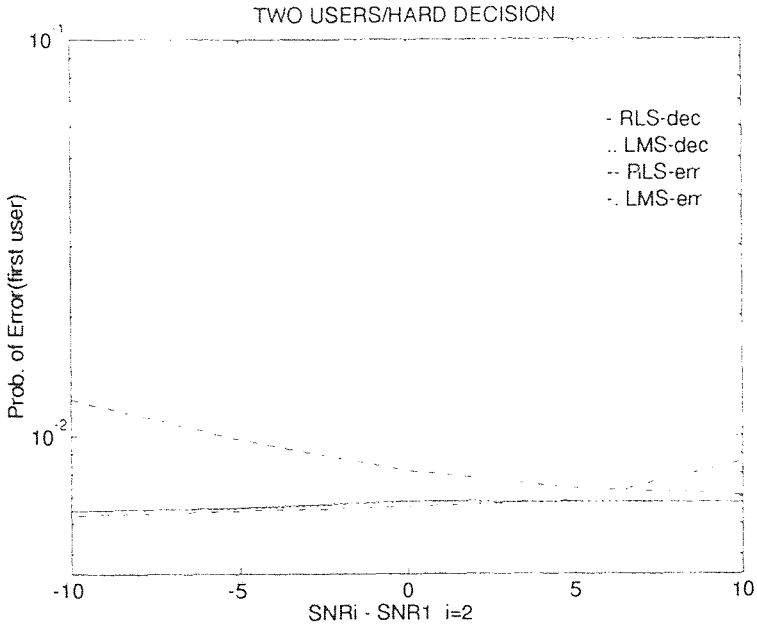


**Figure 6.25** Probability of error of the first user of each of the four algorithms (Hard Decision, N=2, $a_{ij}$=0.15)

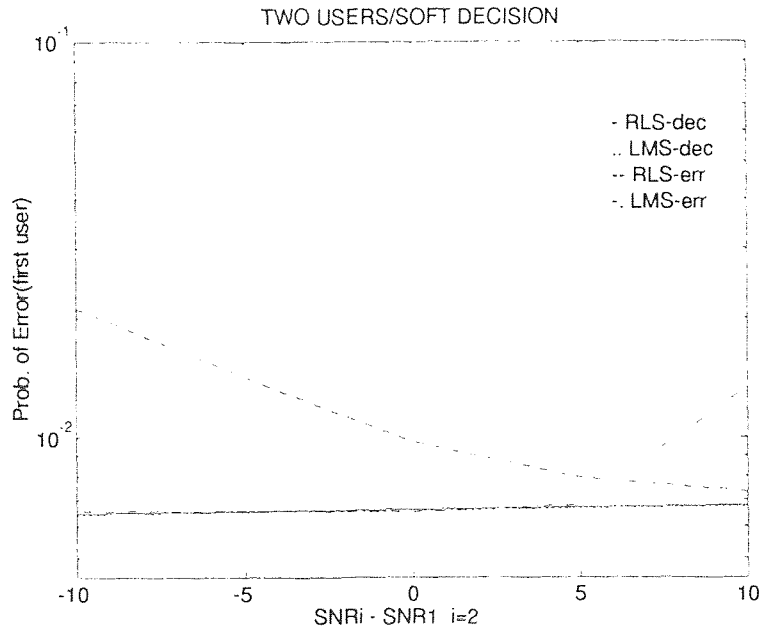


**Figure 6.26** Probability of error of the first user of each of the four algorithms (Soft Decision, N=2, $a_{ij}$=0.15)

### 6.3.2 Four User Case

For the hard decision nonlinearity, two additional users detrimentally affect the performance of the MSE algorithms. Both show behavior similar to the two user case but with greater discrepancies between them and the decorrelating algorithms as shown in Figure 6.27. Furthermore, the lowest probability of error for all the algorithms increased slightly from the two user case, a characteristic observed in learning curves as well.

Implementing the soft decision nonlinearity slightly improved the performance of the MSE algorithms having no expected effect on the decorrelating algorithms as shown in Figure 6.28.

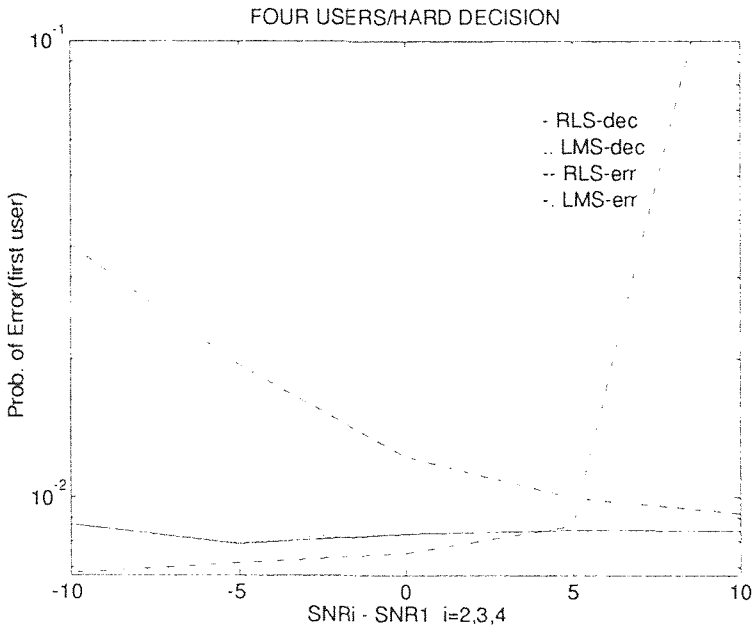


**Figure 6.27** Probability of error of the first user for each of the four algorithms (Hard Decision,N=4,$a_{ij}$=0.15)

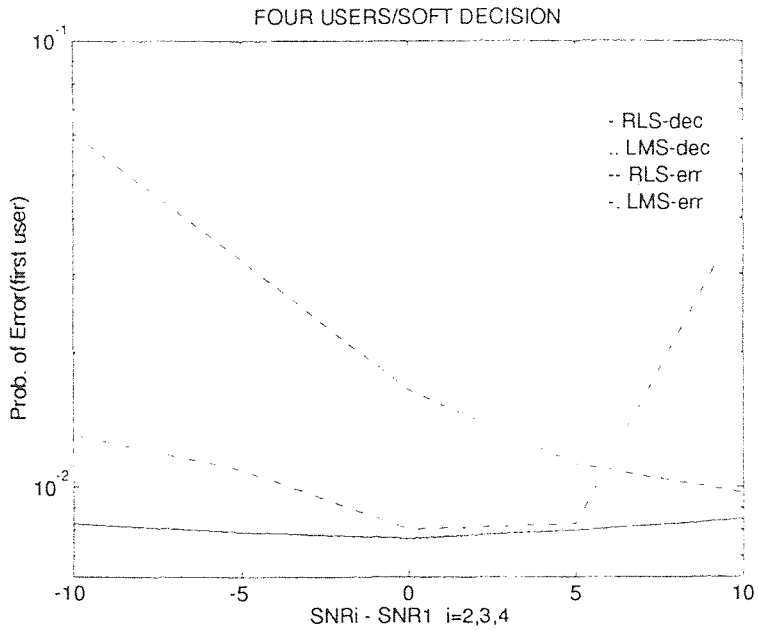


**Figure 6.28** Probability of error of the first user for each of the four algorithms (Soft Decision,N=4,$a_{ij}$=0.15)

## 6.4 Probability of Error Comparison Among Users

In this section, we analyze the extend to which each algorithm suppresses the other users with respect to the first. The parameters are kept the same as in the previous section and only the more interesting case of a four user environment with both hard and soft decision detection is examined. For the hard decision nonlinearity, the four algorithms show significant suppression for negative SSR's as shown in Figures 6.29-6.32. For positive SSR's, the suppression of the other users is no longer affective and now the other users contribute to the interference environment. However, each algorithm maintains a fairly constant probability of error for the first user.

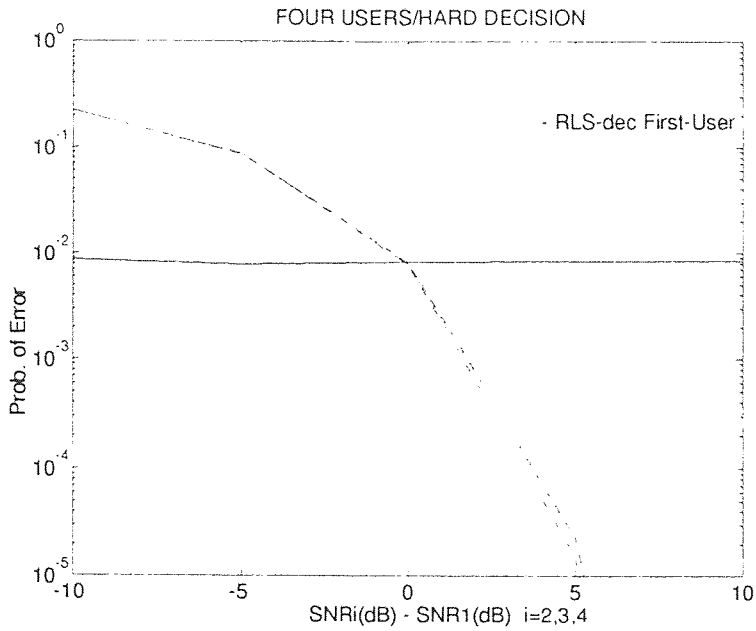


**Figure 6.29** Probability of error of all four users for the RLS decorrelating algorithm (Hard Decision,N=4,$a_{ij}$=0.15)

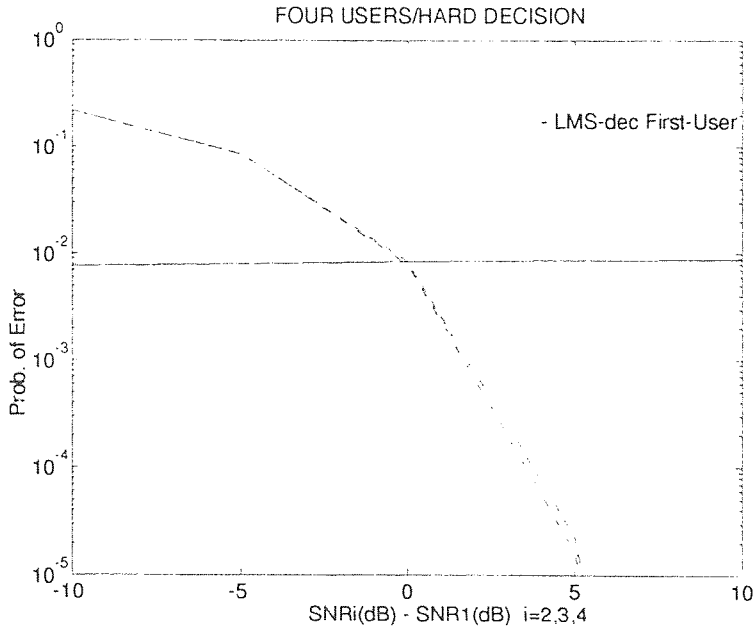


**Figure 6.30** Probability of error of all four users for the LMS decorrelating algorithm (Hard Decision,N=4,$a_{ij}$=0.15)

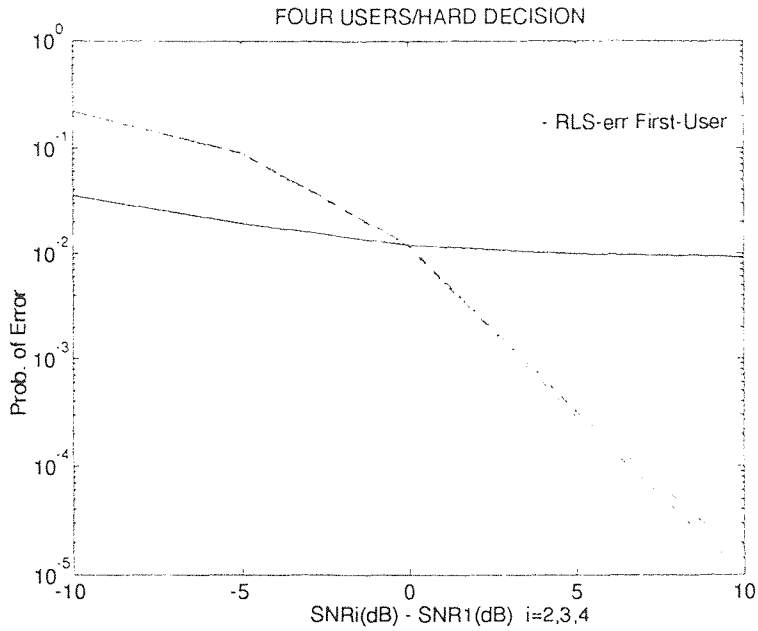


**Figure 6.31** Probability of error of all four users for the RLS error algorithm (Hard Decision,N=4,$a_{ij}$=0.15)

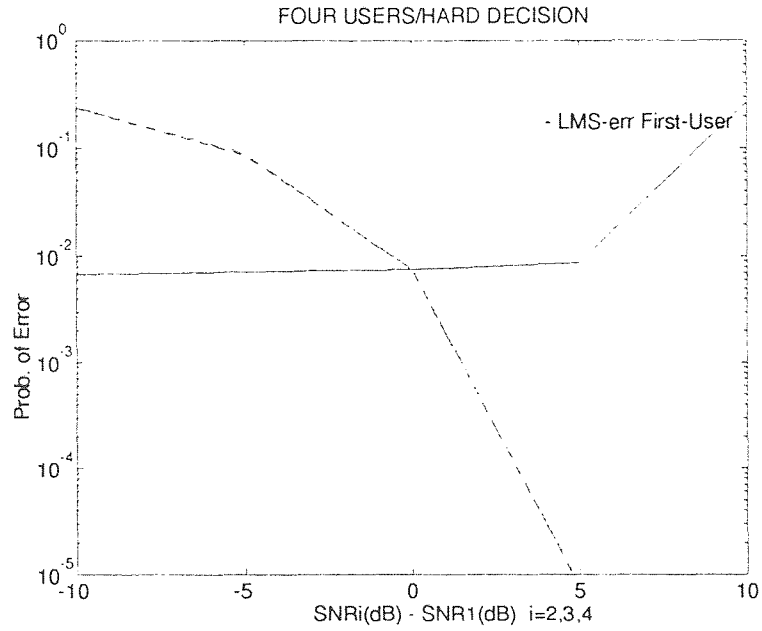


**Figure 6.32** Probability of error of all four users for the LMS error algorithm (Hard Decision,N=4,$a_{ij}$=0.15)

Similarly for the application of the soft decision nonlinearity, the performance is unchanged except for the RLS error algorithm.

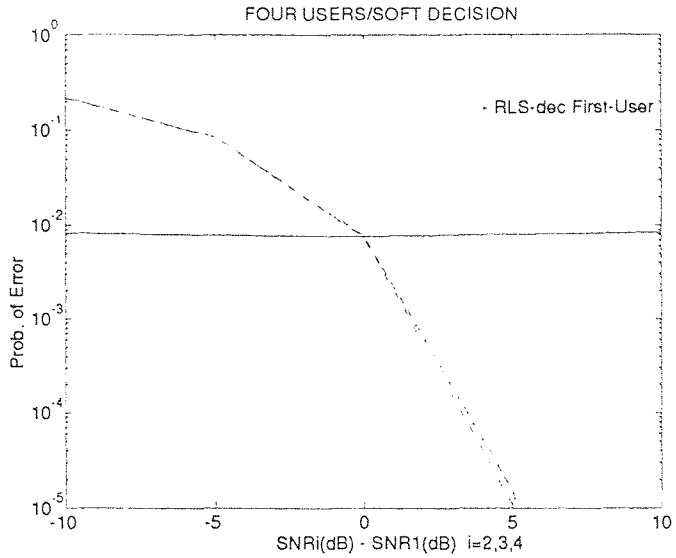


**Figure 6.33** Probability of error of all four users for the RLS decorrelating algorithm (Soft Decision,N=4,$a_{ij}$=0.15)

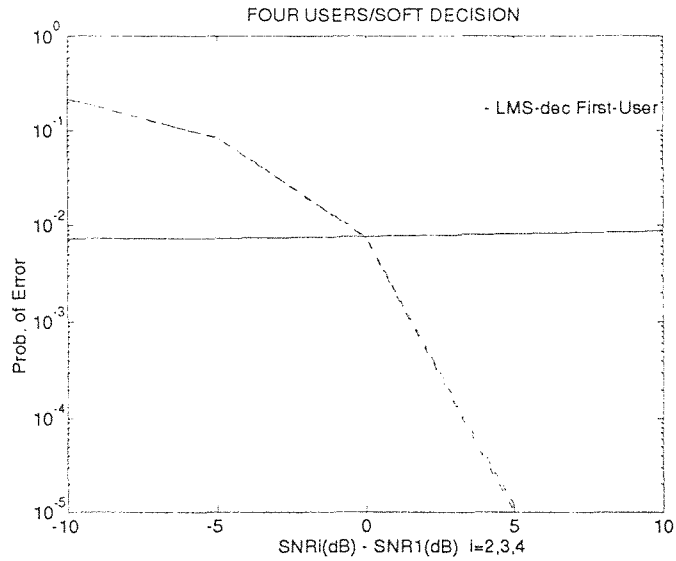


**Figure 6.34** Probability of error of all four users for the LMS decorrelating algorithm (Soft Decision,N=4,$a_{ij}$=0.15)

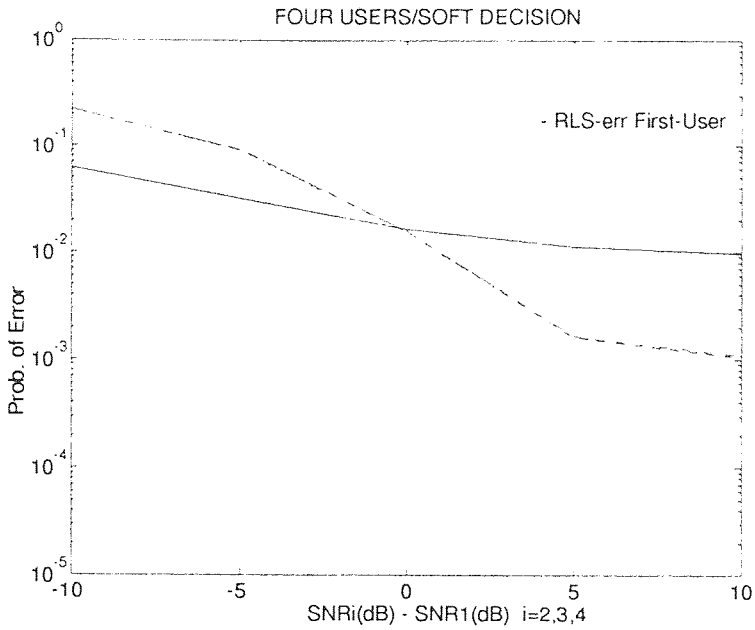


**Figure 6.35** Probability of error of all four users for the RLS error algorithm (Soft Decision,N=4,$a_{ij}$=0.15)

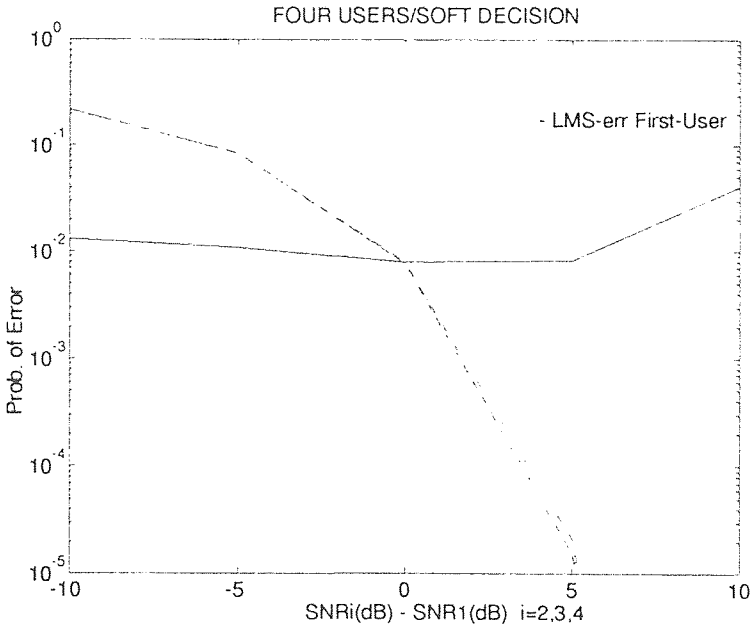


**Figure 6.36** Probability of error of all four users for the LMS error algorithm (Soft Decision,N=4,$a_{ij}$=0.15)

Evaluating the probability of error comparison curves, we observed two interesting features. First, the RLS error algorithm, as previously noted in the learning curves for an SSR=-5dB, behaves uncharacteristically at favorable SSR's with respect to the first user. Again, we attribute this to the forgetting factor $\alpha$. Secondly, the decorrelating algorithms are robust and maintain a fairly constant response throughout the various scenarios.

## 6.5 Eigenvalue Spread

In this section, we verify the theoretical analysis performed in Section 4.3.1 pertaining to the eigenvalue spread of the LMS error, $\chi(\mathbf{R}_x)$, and LMS decorrelating, $\chi\left(\mathbf{R}_{\widehat{B}'x}\right)$, algorithms. We examine the more interesting case of a four user environment.

In order to determine the eigenvalue spread of the algorithms, we implemented recursive equations to calculate the estimated correlation matrices as shown below:

$$\begin{aligned}\widehat{\mathbf{R}}_x(k) &= \widehat{\mathbf{R}}_x(k-1) + \mathbf{x}(k)\mathbf{x}^T(k) \\ \widehat{\mathbf{R}}_{\hat{b}x}(k) &= \widehat{\mathbf{R}}_{\hat{b}x}(k-1) + \hat{\mathbf{b}}(k)\mathbf{x}^T(k)\end{aligned}$$

The estimated correlation matrix of the LMS decorrelator is a submatrix of $\widehat{\mathbf{R}}_{\hat{b}x}$ defined as:

$$\widehat{\mathbf{R}}_{\hat{b}x} = \begin{pmatrix} \sqrt{\xi_1} & \vdots & \mathbf{r}_1{}^T \\ \cdots & \vdots & \cdots \\ \mathbf{r}_1 & \vdots & \widehat{\mathbf{R}}_{\widehat{B}'x} \end{pmatrix}$$

Therefore as the number of iterations increases, the estimated correlation matrices approach the actual correlation matrices. The eigenvalue spread is then calculated as the defined in (4.46). The recursive equations were calculated over 2500 iterations and 10 independent trials with negligible background noise.

In Figure 6.37, we note that both $\chi(\mathbf{R}_x)$ and $\chi\left(\mathbf{R}_{\widehat{B}'x}\right)$ settle to their steady state value after approximately 1500 iterations. Also, $\chi\left(\mathbf{R}_{\widehat{B}'x}\right)$ is approximately equal to the square root of $\chi(\mathbf{R}_x)$, i.e. $\chi\left(\mathbf{R}_{\widehat{B}'x}\right) \leq \sqrt{\chi(\mathbf{R}_x)}$. To be more precise

the actual computer simulations values for the eigenvalue spreads of both algorithms are:

$$\chi\left(\mathbf{R}_{\widehat{B}'x}\right) = 1.5360$$
$$\chi\left(\mathbf{R}_x\right) = 2.9995$$

This is in accordance with *Proposition 4*. To further support our claims, we examined the eigenvalue spread of $\mathbf{R}_{\widehat{bx}}^T\mathbf{R}_{\widehat{bx}}$, $\chi^2\left(\mathbf{R}_{\widehat{bx}}\right)$, in comparison with $\chi\left(\mathbf{R}_x\right)$ to verify *Proposition 3*. In Figure 6.38, we observe that $\chi^2\left(\mathbf{R}_{\widehat{bx}}\right)$ and $\chi\left(\mathbf{R}_x\right)$ do settle to approximately the same steady state value. The actual computer simulations value for the eigenvalue spread of $\mathbf{R}_{\widehat{bx}}^T\mathbf{R}_{\widehat{bx}}$ is:

$$\chi^2\left(\mathbf{R}_{\widehat{bx}}\right) = 3.0955$$

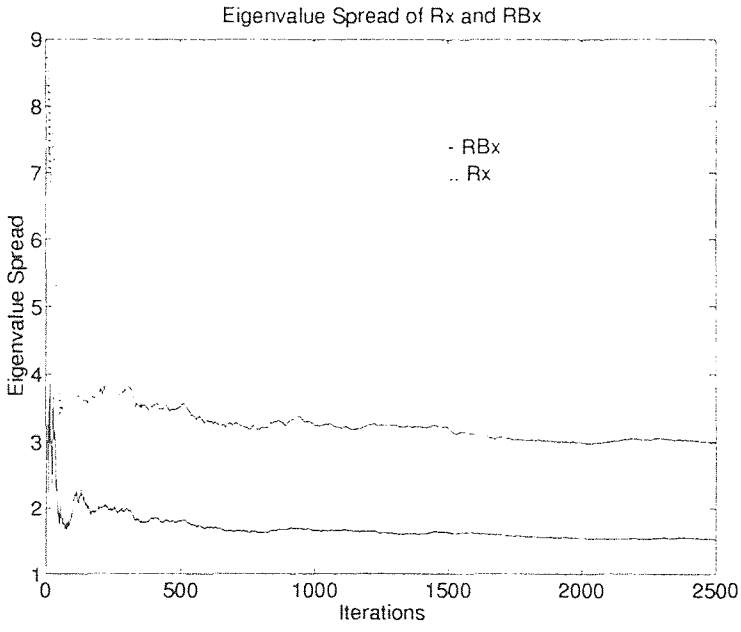


**Figure 6.37** Eigenvalue spread of the LMS decorrelating and LMS error algorithms (SNR=8dB,N=4,$a_{ij}$=0.15)

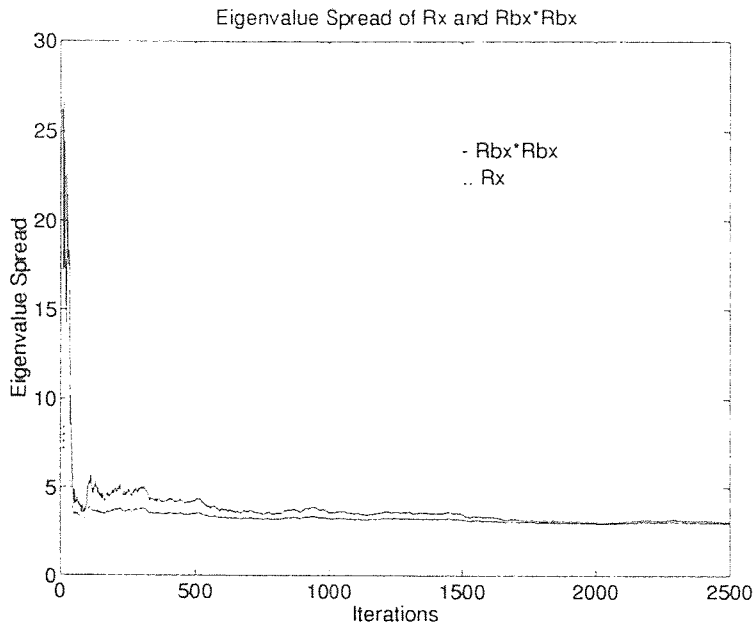


**Figure 6.38** Eigenvalue spread of the LMS decorrelating and LMS error algorithms (SNR=8dB,N=4,$a_{ij}$=0.15)

# CHAPTER 7

# CONCLUSIONS

In this thesis, various algorithms were analyzed and compared. Our focus was to demonstrate the validity of a new RLS type decorrelating algorithm and to show how it performed in comparison to the other algorithms. Also, we analyzed further the convergence of the LMS decorrelator and LMS error algorithms and implemented a relatively new decision nonlinearity known as soft decision to all the algorithms. Our work resulted in some noteworthy contributions:

- The LMS decorrelator algorithm is faster than the LMS error algorithm due to a smaller eigenvalue spread.
- The RLS decorrelator is faster than the LMS type algorithms and comparable to speed with the conventional RLS error algorithm.
- The RLS decorrelator and LMS decorrelator have comparable regions of convergence wider than than the LMS error algorithm.
- The LMS error algorithm's step-size parameter upper bound is smaller than or equal to one half the square of the LMS decorrelator's upper bound.
- The soft decision nonlinearity only significantly improved the performance of the LMS error algorithm, allowing it to converge at higher levels of interference.

# APPENDIX A

## PROOF OF THE JOINT STATISTICS OF $\hat{b}_n b_n$

We will analyze and verify that the cross-correlation matrix between the estimated output bits, $\widehat{\mathbf{b}}$, and the transmitted information bits, $\mathbf{b}$, is equal to the a diagonal matrix $\mathbf{Q}$, $\mathrm{diag}\mathbf{Q} = [q_1, q_2, \cdots, q_N]^T$, where $q_n$ is a constant defined as:

$$E\left[\hat{b}_n b_n\right] = q_n \tag{A.1}$$

and

$$E\left[\hat{b}_m b_n\right] = 0, \tag{A.2}$$

where $n \neq m$. Without loss of generality, we will evaluate only the two users case, $N = 2$, with respect to the first user. Later we will generalize the results for the $n$-th user.

We begin with

$$E\left[\widehat{\mathbf{b}}^T \mathbf{b}\right] = \begin{bmatrix} E\left(\hat{b}_1 b_1\right) & E\left(\hat{b}_2 b_1\right) \\ E\left(\hat{b}_1 b_2\right) & E\left(\hat{b}_2 b_2\right) \end{bmatrix}. \tag{A.3}$$

Evaluating the first term, we get

$$E\left[\hat{b}_1 b_1\right] = \sum_{\hat{b}_1, b_1} \hat{b}_1 b_1 P\left(\hat{b}_1, b_1\right). \tag{A.4}$$

Since $b_n \in \{-1, 1\}$, we expand (A.4) into

$$\begin{aligned} E\left[\hat{b}_1 b_1\right] &= \left(\hat{b}_1 = 1\right)(b_1 = 1) P\left(\hat{b}_1 = 1, b_1 = 1\right) \\ &\quad + \left(\hat{b}_1 = -1\right)(b_1 = 1) P\left(\hat{b}_1 = -1, b_1 = 1\right) \\ &\quad + \left(\hat{b}_1 = 1\right)(b_1 = -1) P\left(\hat{b}_1 = 1, b_1 = -1\right) \\ &\quad + \left(\hat{b}_1 = -1\right)(b_1 = -1) P\left(\hat{b}_1 = -1, b_1 = -1\right). \end{aligned} \tag{A.5}$$

Using Bayes Theorem [26]

$$P(A, B) = P(A|B) P(B), \tag{A.6}$$

where $A$ and $B$ are random variables, we can express (A.5) as:

$$\begin{aligned} E\left[\hat{b}_1 b_1\right] &= P\left(\hat{b}_1 = 1 | b_1 = 1\right) P\left(b_1 = 1\right) \\ &\quad - P\left(\hat{b}_1 = -1 | b_1 = 1\right) P\left(b_1 = 1\right) \\ &\quad - P\left(\hat{b}_1 = 1 | b_1 = -1\right) P\left(b_1 = -1\right) \\ &\quad + P\left(\hat{b}_1 = -1 | b_1 = -1\right) P\left(b_1 = -1\right). \end{aligned} \tag{A.7}$$

Noting that $P\left(b_n = 1\right) = P\left(b_n = -1\right) = 0.5$, (A.7) can be rewritten as:

$$\begin{aligned} E\left[\hat{b}_1 b_1\right] &= 0.5\left[P\left(\hat{b}_1 = 1 | b_1 = 1\right)\right. \\ &\quad - P\left(\hat{b}_1 = -1 | b_1 = 1\right) \\ &\quad - P\left(\hat{b}_1 = 1 | b_1 = -1\right) \\ &\quad \left. + P\left(\hat{b}_1 = -1 | b_1 = -1\right)\right]. \end{aligned} \tag{A.8}$$

After examining (A.8), we can recognize that for $b_1 = 1$

$$\begin{aligned} P\left(\hat{b}_1 = 1 | b_1 = 1\right) &= P\left(\text{correct decision in } \hat{b}_1 | b_1 = 1\right) \\ &= P\left(C_1 | b_1 = 1\right) \end{aligned} \tag{A.9}$$

and

$$\begin{aligned} P\left(\hat{b}_1 = -1 | b_1 = 1\right) &= P\left(\text{error decision in } \hat{b}_1 | b_1 = 1\right) \\ &= P\left(E_1 | b_1 = 1\right). \end{aligned} \tag{A.10}$$

Similarly, for $b_1 = -1$

$$\begin{aligned} P\left(\hat{b}_1 = 1 | b_1 = -1\right) &= P\left(\text{correct decision in } \hat{b}_1 | b_1 = -1\right) \\ &= P\left(C_1 | b_1 = -1\right) \end{aligned} \tag{A.11}$$

and

$$\begin{aligned} P\left(\hat{b}_1 = -1 | b_1 = -1\right) &= P\left(\text{error decision in } \hat{b}_1 | b_1 = -1\right) \\ &= P\left(E_1 | b_1 = -1\right). \end{aligned} \tag{A.12}$$

Therefore, (A.8) can be rewritten as

$$\begin{aligned} E\left[\hat{b}_1 b_1\right] &= 0.5\left[P\left(C_1|b_1=1\right)-P\left(E_1|b_1=1\right)\right. \\ &\quad \left.-P\left(E_1|b_1=-1\right)+P\left(C_1|b_1=-1\right)\right]. \end{aligned} \tag{A.13}$$

Furthermore, from the Total Probability Theorem [26]

$$P(A)=P(A|B_1)\,P(B_1)+P(A|B_2)\,P(B_2)+\cdots+P(A|B_n)\,P(B_n), \tag{A.14}$$

where $A, B_1, \cdots, B_n$ are random variables, we can combine the first and fourth terms and the second and third terms on the right side of (A.13) and get

$$\begin{aligned} E\left[\hat{b}_1 b_1\right] &= 0.5\left[\left\{P\left(C_1|b_1=1\right)+P\left(C_1|b_1=-1\right)\right\}\right. \\ &\quad \left.-\left\{P\left(E_1|b_1=1\right)+P\left(E_1|b_1=-1\right)\right\}\right] \\ &= 0.5\left[2P\left(C_1\right)-2P\left(E_1\right)\right] \\ &= P\left(C_1\right)-P\left(E_1\right). \end{aligned} \tag{A.15}$$

Also we note that the complement of the probability of $A$ is 1 minus the probability of $A$, therefore

$$P\left(C_1\right)=1-P\left(E_1\right). \tag{A.16}$$

and (A.15) can be rewritten as:

$$E\left[\hat{b}_1 b_1\right] = 1-2P\left(E_1\right). \tag{A.17}$$

To determine the value of $P\left(E_1\right)$, we evaluate the term

$$\begin{aligned} P\left(E_1\right) &= P\left(E_1|b_1=1\right)+P\left(E_1|b_1=-1\right) \\ &= P\left(\hat{b}_1<0|b_1=1\right)+P\left(\hat{b}_1>0|b_1=-1\right). \end{aligned} \tag{A.18}$$

Due to symmetry, both terms in (A.18) are equal and therefore

$$P\left(E_1\right)=2P\left(E_1|b_1=1\right). \tag{A.19}$$

Expanding the term on the right hand side by noting that for the hard decision case,

$$\begin{aligned}\hat{b}_1 &= \operatorname{sgn}(y_1) \\ &= \operatorname{sgn}\left(\mathbf{w}_1^T \mathbf{x}\right) \\ &= \operatorname{sgn}\left(\mathbf{w}_1^T (\mathbf{AEb} + \mathbf{v})\right) \\ &= \operatorname{sgn}(t_{11} b_1 + t_{12} b_2 + \eta_1), \end{aligned} \tag{A.20}$$

where $t_{11}$ and $t_{12}$ are the elements of the vector $\mathbf{t}_1$ defined as:

$$\begin{aligned}\mathbf{t}_1 &= \mathbf{w}_1^T \mathbf{AE} \\ &= [t_{11}\ t_{12}], \end{aligned}$$

and $\eta_1$ is the filtered Gaussian noise defined as

$$\eta_1 = \mathbf{w}_1^T \mathbf{v}. \tag{A.21}$$

Therefore, we can express (A.19) as

$$\begin{aligned}P(E_1) &= 2P(E_1|b_1 = 1) \\ &= 2P(t_{11} b_1 + t_{12} b_2 + \eta_1 > 0 | b_1 = -1) \\ &= 2P(-t_{11} + t_{12} b_2 + \eta_1 > 0) \\ &= 2P(\eta_1 > t_{11} - t_{12} b_2). \end{aligned} \tag{A.22}$$

Again using the Total Probability Theorem on (A.22) over the two values of $b_2$ results in

$$\begin{aligned}P(E_1) &= 2[P(\eta_1 > t_{11} - t_{12} b_2 | b2 = 1) P(b2 = 1) \\ &\quad + P(\eta_1 > t_{11} - t_{12} b_2 | b2 = -1) P(b2 = -1)] \\ &= P(\eta_1 > t_{11} - t_{12}) + P(\eta_1 > t_{11} + t_{12}), \end{aligned} \tag{A.23}$$

where $P(b_2 = 1) = P(b_2 = -1) = 0.5$. Since $\eta_1$ is a Gaussian random variable with a density function equal to

$$f_\eta(z) = \frac{1}{\sqrt{2\pi}\sigma_\eta} \exp^{-\frac{z^2}{2\sigma_\eta^2}}. \tag{A.24}$$

where $\sigma_\eta^2$ is the filtered noise power defined as $\sigma_{\mathrm{n}}^2 \mathbf{w}_n^T \mathbf{H} \mathbf{w}_n$. We can rewrite (A.23) in terms of the corresponding density function with the appropriate limits such as:

$$P(E_1) = \frac{1}{\sqrt{2\pi}\sigma_\eta} \int_{t_{11}-t_{12}}^{\infty} \exp^{-\frac{z^2}{2\sigma_\eta^2}} dz + \frac{1}{\sqrt{2\pi}\sigma_\eta} \int_{t_{11}+t_{12}}^{\infty} \exp^{-\frac{z^2}{2\sigma_\eta^2}} dz \qquad (A.25)$$

After observing (A.25), we can state that with some manipulation of the limits and the implementation of the $Q$-function defined as

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} \exp^{-\frac{\zeta^2}{2}} d\zeta, \qquad (A.26)$$

we can rewrite (A.25) as

$$P(E_1) = Q\left(\frac{t_{11}-t_{12}}{\sigma_{\mathrm{n}}\sqrt{\mathbf{w}_n^T \mathbf{H} \mathbf{w}_n}}\right) + Q\left(\frac{t_{11}+t_{12}}{\sigma_{\mathrm{n}}\sqrt{\mathbf{w}_n^T \mathbf{H} \mathbf{w}_n}}\right). \qquad (A.27)$$

Consequently, we can insert (A.27) into (A.17) and obtain the joint statistic for $\hat{b}_1 b_1$.

$$\begin{aligned} E\left[\hat{b}_1 b_1\right] &= 1 - 2\left[Q\left(\frac{t_{11}-t_{12}}{\sigma_{\mathrm{n}}\sqrt{\mathbf{w}_n^T \mathbf{H} \mathbf{w}_n}}\right) + Q\left(\frac{t_{11}+t_{12}}{\sigma_{\mathrm{n}}\sqrt{\mathbf{w}_n^T \mathbf{H} \mathbf{w}_n}}\right)\right] \qquad (A.28) \\ q_1 &= 1 - 2\left[Q\left(\frac{t_{11}-t_{12}}{\sigma_{\mathrm{n}}\sqrt{\mathbf{w}_n^T \mathbf{H} \mathbf{w}_n}}\right) + Q\left(\frac{t_{11}+t_{12}}{\sigma_{\mathrm{n}}\sqrt{\mathbf{w}_n^T \mathbf{H} \mathbf{w}_n}}\right)\right]. \end{aligned}$$

Following the same procedure, we can evaluate the joint statistic of $E\left[\hat{b}_2 b_1\right]$.

$$\begin{aligned} E\left[\hat{b}_2 b_1\right] &= \sum_{\hat{b}_2, b_1} \hat{b}_2 b_1 P\left(\hat{b}_2, b_1\right) \\ &= \left(\hat{b}_2 = 1\right)(b_1 = 1) P\left(\hat{b}_2 = 1, b_1 = 1\right) \\ &\quad + \left(\hat{b}_2 = -1\right)(b_1 = 1) P\left(\hat{b}_2 = -1, b_1 = 1\right) \\ &\quad + \left(\hat{b}_2 = 1\right)(b_1 = -1) P\left(\hat{b}_2 = 1, b_1 = -1\right) \\ &\quad + \left(\hat{b}_2 = -1\right)(b_1 = -1) P\left(\hat{b}_2 = -1, b_1 = -1\right) \\ &= P\left(\hat{b}_1 = 1, b_1 = 1\right) - P\left(\hat{b}_1 = -1, b_1 = 1\right) \\ &\quad - P\left(\hat{b}_1 = 1, b_1 = -1\right) + P\left(\hat{b}_1 = -1, b_1 = -1\right). \qquad (A.29) \end{aligned}$$

Again using Bayes Theorem and noting that $P(b_1 = 1) = P(b_1 = -1) = 0.5$, we get

$$E\left[\hat{b}_2 b_1\right] = P\left(\hat{b}_2 = 1 | b_1 = 1\right) P(b_1 = 1)$$

$$
\begin{aligned}
& -P\left(\widehat{b}_2 = -1 | b_1 = 1\right) P\left(b_1 = 1\right) \\
& -P\left(\widehat{b}_2 = 1 | b_1 = -1\right) P\left(b_1 = -1\right) \\
& +P\left(\widehat{b}_2 = -1 | b_1 = -1\right) P\left(b_1 = -1\right) \\
= \; & 0.5\left[P\left(\widehat{b}_2 = 1 | b_1 = 1\right)\right. \\
& -P\left(\widehat{b}_2 = -1 | b_1 = 1\right) \\
& -P\left(\widehat{b}_2 = 1 | b_1 = -1\right) \\
& \left. + P\left(\widehat{b}_2 = -1 | b_1 = -1\right)\right] . \qquad \text{(A.30)}
\end{aligned}
$$

Evaluating the first term in (A.30) we see that

$$
\begin{aligned}
P\left(\widehat{b}_2 = 1 | b_1 = 1\right) &= \sum_{\widehat{b}_2} P\left(\widehat{b}_2 = 1 | b_1 = 1, b_2\right) P\left(b_2\right) \\
&= 0.5\left[P\left(\widehat{b}_2 = 1 | b_1 = 1, b_2 = 1\right)\right. \\
+ P\left(\widehat{b}_2 = 1 | b_1 = 1, b_2 = -1\right)\Big] . & \qquad \text{(A.31)}
\end{aligned}
$$

We can recall from equations (A.9)-(A.12), and recognize that

$$
\begin{aligned}
P\left(\widehat{b}_2 = 1 | b_1 = 1\right) &= P\left(\text{correct decision in } \widehat{b}_2 | b_1 = 1\right) \\
&= P\left(C_2 | b_1 = 1\right) \qquad \text{(A.32)}
\end{aligned}
$$

$$
\begin{aligned}
P\left(\widehat{b}_2 = 1 | b_1 = 1\right) &= P\left(\text{error decision in } \widehat{b}_2 | b_1 = 1\right) \\
&= P\left(E_2 | b_1 = 1\right) . \qquad \text{(A.33)}
\end{aligned}
$$

We can then rewrite (A.31) as

$$
P\left(\widehat{b}_2 = 1 | b_1 = 1\right) = 0.5\left[P\left(C_2 | b_1 = 1\right) + P\left(E_2 | b_1 = 1\right)\right] .
$$

Repeating these steps for the other three terms in (A.30) results in

$$
P\left(\widehat{b}_2 = -1 | b_1 = 1\right) = 0.5\left[P\left(C_2 | b_1 = 1\right) + P\left(E_2 | b_1 = 1\right)\right] \qquad \text{(A.34)}
$$

$$
P\left(\widehat{b}_2 = 1 | b_1 = -1\right) = 0.5\left[P\left(C_2 | b_1 = -1\right) + P\left(E_2 | b_1 = -1\right)\right] \qquad \text{(A.35)}
$$

$$P\left(\widehat{b}_2 = -1 | b_1 = -1\right) = 0.5\left[P\left(C_2|b_1 = -1\right) + P\left(E_2|b_1 = -1\right)\right]. \quad (A.36)$$

Gathering (A.34)-(A.36) and inserting them into (A.30) results in the cross-joint statistic of $\widehat{b}_2 b_1$ given by:

$$\begin{aligned} E\left[\widehat{b}_2 b_1\right] &= 0.5\left[P\left(C_2|b_1 = 1\right) + P\left(E_2|b_1 = 1\right)\right. \\ &\quad -P\left(C_2|b_1 = 1\right) - P\left(E_2|b_1 = 1\right) \\ &\quad -P\left(C_2|b_1 = -1\right) - P\left(E_2|b_1 = -1\right) \\ &\quad \left.+P\left(C_2|b_1 = -1\right) + P\left(E_2|b_1 = -1\right)\right] \\ &= 0 \end{aligned} \quad (A.37)$$

Therefore, the cross-joint statistics of the estimated output bits and the transmitted information bits is zero without any constraining assumptions.

Generalizing the results obtained above for the $n$-th user, results in

$$E\left[\widehat{b}_n b_n\right] = 1 - 2\left[\sum_{\substack{b\in\{-1,1\}^N \\ b_n=-1}} Q\left(\frac{t_{nn} - \sum_{j\neq n} t_{nj} b_j}{\sigma_n \sqrt{\mathbf{w}_n^T \mathbf{H} \mathbf{w}_n}}\right)\right] \quad (A.38)$$

and

$$E\left[\widehat{b}_m b_n\right] = 0, \quad (A.39)$$

where $n \neq m$.

# APPENDIX B

## SOFT DECISION NONLINEARITY DERIVATION

Here we derive the soft decision nonlinearity based on the work of Nowlan and Hinton [25]. Given the adaptive structure in Figure B.1, we obtain the fitered output as:

$$y(k) = b(k) + v(k) \tag{B.1}$$

where $b(k)$ is the information bit, discretely distributed with a zero mean and a unit variance and $v(k)$ is the sum contributions of residual multi-user interference and thermal noise. To make the analysis mathematically tractable, $v(k)$ is modeled as white Gaussian noise with zero mean and a variance of $\sigma^2$, which is statistically independent of $b(k)$.

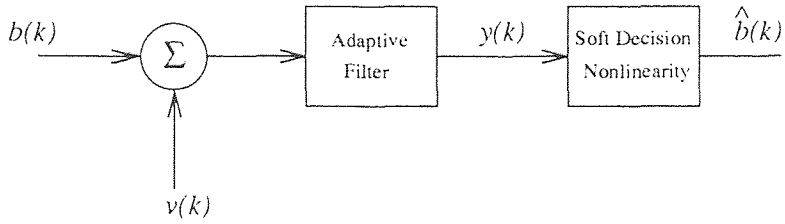


**Figure B.1** The adaptive structure implemented to derive the soft decision nonlinearity.

We derive a Bayes estimate of $b(k)$ optimized in the mean-square sense. For convenience of presentation, we will suppress the time dependence.

For the filtered output $y$, we write the conditional mean estimate $\widehat{b}$ of the random variable $b$ as $E\left[\widehat{b}|y\right]$. We then let $f_b(b|y)$ denote the conditional probability density function of $b$ given $y$ or

$$\begin{aligned} \widehat{b} &= E[b|y] \\ &= \int_{-\infty}^{\infty} b\, f_b(b|y)\, db. \end{aligned} \tag{B.2}$$

Using a fundamental property of conditional probability, we can rewrite $f_b(b|y)$ as

$$f_b(b|y) = \frac{f_y(y|b) f_b(b)}{f_y(y)} \tag{B.3}$$

and consequently we can rewrite (B.2) as

$$\hat{b} = \frac{1}{f_y(y)} \int_{-\infty}^{\infty} b f_y(y|b) f_b(b) db. \tag{B.4}$$

Next, we let

$$y = cb + v, \tag{B.5}$$

where $c$ is a scaling factor smaller than unity, included so as to keep $E[y^2] = 1$. Therefore keeping with the assumptions made earlier for the statistics of $b$ and $v$ and the restriction on $E[y^2]$, we can determine the value of $c$ as follows:

$$\begin{aligned} E\left[y^2\right] &= E[(cb+v)(cb+v)] \\ 1 &= c^2 E\left[b^2\right] + E\left[v^2\right] \\ 1 &= c^2 + \sigma^2 \\ c &= \sqrt{1-\sigma^2}. \end{aligned} \tag{B.6}$$

Furthermore, it follows from (B.5) that

$$f_y(y|b) = f_v(y - cb). \tag{B.7}$$

Accordingly, substituting (B.7) into (B.4) yields

$$\hat{b} = \frac{1}{f_y(y)} \int_{-\infty}^{\infty} b f_v(y - cb) f_b(b) db. \tag{B.8}$$

We then evaluate (B.8) as follows:

$$\begin{aligned} \int_{-\infty}^{\infty} b f_v(y - cb) f_b(b) db &= 0.5(1) f_v(y - c) + 0.5(-1) f_v(y + c) \\ &= 0.5 f_v(y - c) - 0.5 f_v(y + c). \end{aligned} \tag{B.9}$$

where $f_b(b)$ is defined as:

$$f_b(b) = 0.5\delta\left(b+\widehat{b}\right)+0.5\delta\left(b-\widehat{b}\right)$$

where $\delta(u)$ is the usual Kronecker delta function

$$\delta(u)=\begin{cases}1 & \text{u=0}\\ 0 & \text{otherwise.}\end{cases}$$

and $b\in\{-1,1\}$. Next, we evaluate $f_y(y)$ as

$$\begin{aligned} f_y(y) &= \int_{-\infty}^{\infty} f_v(y-cb)\,f_b(b)\,db \\ &= 0.5f_v(y-c)+0.5f_v(y+c) \end{aligned} \tag{B.10}$$

Substituting (B.9) and (B.10) into (B.8) yields

$$\begin{aligned} \widehat{b} &= \frac{0.5f_v(y-c)-0.5f_v(y+c)}{0.5f_v(y-c)+0.5f_v(y+c)} \\ &= \frac{f_v(y-c)-f_v(y+c)}{f_v(y-c)+f_v(y+c)}. \end{aligned} \tag{B.11}$$

Next, we define the density function of the Gaussian mixture $f_y(y-c)$ as

$$f_v(y-c) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp^{\frac{-(y-c)^2}{2\sigma^2}} \tag{B.12}$$

and substituting (B.12) back into (B.11) yields

$$\begin{aligned} \widehat{b} &= \frac{\frac{1}{\sqrt{2\pi\sigma^2}}\left[\exp^{\frac{-y^2}{2\sigma^2}}\exp^{\frac{-c^2}{2\sigma^2}}\left(\exp^{\frac{cy}{\sigma^2}}-\exp^{\frac{-cy}{\sigma^2}}\right)\right]}{\frac{1}{\sqrt{2\pi\sigma^2}}\left[\exp^{\frac{-y^2}{2\sigma^2}}\exp^{\frac{-c^2}{2\sigma^2}}\left(\exp^{\frac{cy}{\sigma^2}}+\exp^{\frac{-cy}{\sigma^2}}\right)\right]} \\ &= \frac{\exp^{\frac{cy}{\sigma^2}}-\exp^{\frac{-cy}{\sigma^2}}}{\exp^{\frac{cy}{\sigma^2}}+\exp^{\frac{-cy}{\sigma^2}}} \\ &= \tanh\left(\frac{cy}{\sigma^2}\right). \end{aligned} \tag{B.13}$$

Therefore, the optimum nonlinear function available for decision-directed algorithms is the hyperbolic tangent function

From (B.13), we can see that for $\frac{cy}{\sigma^2}\gg 1$, the $\tanh(\cdot)$ function approaches the characteristics of the sgn function. The weakness of the hard decision is apparent

when $y$ is very close to zero. A decision based on the sign of $y$ will most likely be incorrect, due to the effects of the random noise. Yet it is in these cases that the weights are changed the most. The soft decision nonlinearity has a much smaller magnitude when $y$ is close to zero, so it makes only small weight changes in this highly ambiguous case. It smooths over the sharp discontinuity of the sgn function. The difference between the two nonlinearity will be most apparent when there are numerous incorrect decisions, i.e. high probability of error. Simulation reported in [25] suggest that the use of soft decision nonlinearity can lead to more rapid initial convergence than the hard decision in channels with moderate to severe noise and distorion.

# REFERENCES


1. B. Aazhang, B. Paris, and G. Orsak, "Neural networks for multiuser detection in code division multiple access communications," *IEEE Trans. on Communications*, vol. 40, no. 7, pp. 1212–1222, July 1992.

2. Y. Bar-Ness, "Bootstrapped cross-pol interference cancelling techniques-steady state analysis," *Bell Lab Technical Report*, 1982.

3. Y. Bar-Ness, A. Dinc, and H. Messer, "Bootstrapped adaptive separation of superimposed signals : Analysis of the effect of thermal noise on performance," *submitted to IEE Proceedings-F, Radar and Signal Processing*, 1992.

4. Y. Bar-Ness and J. Rukach, "Cross-coupled bootstrapped interference canceler," *AP-S International Symposium, Conference Proceedings*, pp. 292–295, June 1981.

5. G. Burel, "Blind separation of sources: A nonlinear neural algorithm," *Neural Networks*, vol. 5, pp. 937–947, November/December 1992.

6. P. Comon, "Separation of sources using high-order cumulants," in *SPIE Conference on Adaptive Algorithms and Architectures for Signal Processing*, San Diego, CA, pp. 170–181, August 8-10 1989. paper 3.7.

7. D. Compernolle and S. V. Gerven, "Feedforward and feedback in a symmetric adaptive noise canceler:Stability analysis in a simplified case," in *Signal Processing VI: Theories and Applications*, vol. 2, Brussels, Belgium, August 24-27 1992.

8. D. Compernolle and S. V. Gerven, "Signal separation in symmetric adaptive noise canceler by output decorrelation," in *IEEE Int'l Conference on Acoustics, Speech and Signal Processing*, vol. 4, San Francisco, CA, March 23-26 1992.

9. R. T. Compton, *Adaptive Antennas*, Prentice Hall, Englewood Cliffs, NJ, 1988.

10. A. Dinc and Y. Bar-Ness, "Performance comparison of LMS, diagonalizer and bootstrapped adaptive cross-pol cancelers for M-ary QAM," in *Proceedings of Milcom'90*, Monterey, CA, October 1990. paper 3.7.

11. A. Dinc and Y. Bar-Ness, "Bootstrap: A fast adaptive signal separator," in *ICASSP 92*, pp. II.325–II.328, 1992.

12. A. Dinc and Y. Bar-Ness, "Comparison of three different structures of bootstrap blind adaptive algorithms for multisignal co-channel separation," in *MILCOM '92*, San Diego, CA, October 11-14 1992.

13. A. Dinc and Y. Bar-Ness, "Convergence and performance comparison of three different structures of bootstrap blind adaptive algorithms for mutiignal co-channel separation," in *MILCOM '92*, San Diego, CA, October 11-14 1992.

14. A. Dinc and Y. Bar-Ness, "Error probability of bootstrapped blind adaptive cross-pol cancelers for M-ary QAM," in *IEEE Int'l Conference on Communications*, vol. 2, Chicago, IL, June 14-18 1992. paper no. 353.5.

15. A. Dinc and Y. Bar-Ness, "A forward/backward bootstrapped structure for blind separation of signals in a multi-channel dispersive environment," in *IEEE Int'l Conference on Acoustics, Speech and Signal Processing*, vol. 3, Minneapolis, MN., pp. 376–379, April 27-30 1993.

16. G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

17. S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, second ed., 1991.

18. C. Jutten and J. Herault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetric architecture," *Signal Processing, Elsevier*, vol. 24, pp. 1–10, July 1991.

19. C. Jutten and J. Herault, "Blind separation of sources, Part II: Problems statement," *Signal Processing, Elsevier*, vol. 24, pp. 11–20, July 1991.

20. C. Jutten, L. Nguyenthi, E. Dukstra, E. Vittoz, and J. Caelen, "Blind separation of sources:An algorithm for separation of convolutive mixtures," in *Workshop High-Order Statistics*, Chamrousse, France, July 10-12 1991.

21. M. Kavehrad, "Performance of cross-polarized QAM signals over non-dispersive fading channels," *AT&T Bell Lab. Technical Journal*, vol. 63, pp. 499–521, March 1984.

22. U. Mitra and H. Poor, "Adaptive receiver algorithms for near-far resistant CDMA," in *The 3rd IEEE Intl. Symposium on Personal, Indoor and Mobile Radio Comm.*, Boston, MA, October 19-21 1992.

23. R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*, John Wiley and Sons Inc., New York, NY, 1980.

24. E. Moreau and O. Macchi, "Two novel architectures for self adaptive separation of signals," *IEEE*, February 1993.

25. S. J. Nowlan and G. E. Hinton, "A soft decision-directed LMS algorithm for blind equalization," *IEEE Trans. on Communications*, vol. 41, pp. 275-279, February 1993.

26. A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill Inc., New York, NY, third ed., 1991.

27. G. V. Reklaitis, A. Ravindran, and K. M. Ragsdett, *Engineering Optimization: Methods and Applications*, John Wiley and Sons Inc., New York, NY, 1983.

28. E. Sorouchyari, "Blind separation of sources, Part III: Stability analysis," *Signal Processing, Elsevier*, vol. 24, pp. 21–29, July 1991.