USC–SIPI REPORT #161

A Linearly Constrained adaptive Algorithm for Constant Modulus Signal Processing

by

Michael J. Rude

August 1990

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

CSI-90-08-01

A Linearly Constrained Adaptive Algorithm for Constant ModulusSignal Processing

by

Michael J. Rude

Communications Sciences Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
Powell Hall of Engineering
University Park/MC-0272
Los Angeles, CA 90089 U.S.A.

August 1990

This work was supported by the Garland Division of E-Systems, Inc. under Contract 7307-EP-76673

Acknowledgements

I am grateful to Professor Lloyd Griffiths, my research advisor, for his guidance, encouragement and good humor. His broad-minded approach to research and graduate study has made the last 5 years much more rewarding. I would also like to thank Professors Scholtz, Weber, Leahy and Harris for their participation on my dissertation committee as well as their academic and career guidance. Special thanks to Professors Scholtz and Weber for introducing me to communication problems and for involving me in activities of the Communication Science Institute (CSI.)

Over the past 5 years, E-SYSTEMS, Inc. has been a generous sponsor of my research efforts on a contract basis. In addition to this financial support, the interaction with E-SYSTEMS employees has been and continues to be of great value.

The Signal and Image Processing Institute (SIPI) has been a comfortable place to be a student and carry out research. My thanks to Professors Sawchuk and Chellappa for providing a first-rate computing environment. Special credit is due to Allan Weber for his knowledge of the hardware, software and networks in use at SIPI and for his patience to teach me about them. Both SIPI and CSI are fortunate to have able staff members such as Gloria Bullock, Delsa Tan, Linda Varilla and Milly Montenegro. All have been good company and very helpful in

solving bureaucratic problems. My fellow students in SIPI have not only been of great help in research and scholastic matters but have become friends as well. From among these individuals, I would like to thank George Zunich, Ching-Yih Tseng and David Feldman. Their friendship and willingness to share knowledge has made graduate study much more satisfying.

I would like to thank my housemates Andy Ragan, Miriam Docter and Eric Dullea for being good friends and tolerating the sour moods that occasionally befall Ph.D. students. I would also like to thank Christine Peterson for steering my personal life back to the activities I have always enjoyed and at the same time giving me warmth, companionship and love. Without her, the final stages of graduate school would have been much more difficult.

When I moved to Los Angeles I left my family and most of my relatives behind. Kris Kelly, my cousin and closest relative living in LA, has been a good family confidante over the past five years. I would also like to thank my sister Karen and brother Peter who have given me regular encouragement and support even though they both live thousands of miles away.

No amount of gratitude can express the debt I owe to my parents David and Laura who have always given me unconditional love. They encouraged me in all of my educational endeavors and provided me with a home environment where learning was a natural part of life. With love, this dissertation is dedicated to them.

Contents

List of Tables

List of Figures

Abstract

This dissertation introduces an adaptive technique that is suitable for signal processing tasks in severe co-channel interference and multipath environments. The new approach is based on the minimization of complex envelope variations at the processor output subject to a set of linear constraints on the processor coefficients. As a result, it does not require a training, reference or desired signal for adaptation. This Linearly Constrained Constant Modulus (LCCM) method combines two existing untrained adaptive techniques: linearly constrained minimization of output power and unconstrained minimization of complex envelope or modulus variations.

LCCM addresses the signal cancellation problem that occurs when untrained adaptive processors are used in hostile signal environments; minimization of output power can lead to signal cancellation in multipath scenarios and unconstrained minimization of envelope variations can lead to signal cancellation in applications that include constant envelope co-channel interference. Analysis of a narrowband array application shows that the LCCM approach will eliminate the signal cancellation problem when appropriate *a priori* information is incorporated into the constraint set. An additional result is that the signal of interest does not necessarily need to have the constant envelope property.

Adaptive implementation of the LCCM method is accomplished using the common technique of Stochastic Gradient Descent (SGD.) Stable adaptive recursions are derived via comparison with other SGD recursions that are used for existing adaptive approaches. In addition, the roles of convergence parameters are described and practical values are specified. For special cases, convergence in the mean is shown.

In a typical signal processing scenario, LCCM is used for initial acquisition of a signal and then adaptive processing is switched to a decision directed mode. Simulation results for LCCM in the start-up or acquisition mode are presented for both single and multi-channel reception of a digitally modulated signal. The single channel experiment consists of a fractionally spaced LCCM equalizer with a signal subspace based linear constraint. The multi-channel experiment consists of a narrowband array with a look direction constraint. In both cases, the LCCM processor rejects the co-channel interference and compensates for the multipath distortion.

Chapter 1

Introduction

Modern radio communication receivers bear increasing resemblance to special purpose digital computers. Technological advances have made it possible to sample and quantize a radio signal waveform essentially as it is received and down-converted at the antenna. Receiver tasks such as carrier recovery, synchronization and matched filtering which have traditionally been done with analog circuits can now be implemented using digital signal processing (DSP) techniques.

One of the many advantages of implementing individual receiver tasks with DSP techniques is that the overall receiver operation can be made adaptive. An adaptive implementation allows the receiver to adjust to new or different operating conditions. For example, the *average* characteristics of the channel transfer function and noise field are taken into account at the transmitter by using appropriate coding and modulation schemes. This approach is extremely effective in most signal environments. However, an adaptive receiver has the potential to improve performance even further by adjusting the receiver to the *particular* characteristics of the signal environment.

Adaptive receivers can take many forms with varying degrees of complexity. They may be designed for analog or digital modulation. They may operate on single or multi-channel input data. They may have a single adaptive component, e.g. equalization, or they may have several adaptive elements, e.g. acquisition, synchronization and detection. Clearly, the range of application for adaptive processing in communication receivers is very wide. Likewise, the number of adaptive algorithms that are available for these applications is quite high. Most of these algorithms, however, are implementations that are based on a relatively small number of adaptive signal processing concepts.

All adaptive signal processing concepts are in some way based on general assumptions about the signal environment. When these assumptions are not valid the adaptive implementation of these concepts will break down. Although most environments do satisfy the general assumptions, there are many scenarios of interest that do not. This dissertation proposes a new adaptive signal processing concept that is suitable for such scenarios.

1.1 Organization of Dissertation

This dissertation is divided into 7 chapters. The size of the chapters varies since each chapter has a specific goal which can be addressed in fewer pages in some chapters than in others. Generally, detail is only introduced as it is required and most background material is referenced rather than re-derived.

Chapter 2 reviews the problem of interest and describes the adaptive approaches that are available to solve it. The basic problem is one of adaptive operation in severe co-channel interference and multipath signal environments.

Optimal methods based on reference signals are unsuitable because no reference signal is available in the case of interest. It is shown that each of the existing untrained or reference-free methods has certain shortcomings which limit its application. The goal of this chapter is to narrow the topic and demonstrate a need for yet another adaptive signal processing concept.

Chapter 3 develops the Linearly-Constrained Constant Modulus (LCCM) approach to untrained adaptive signal processing. The LCCM concept combines the ideas behind two existing methods. Techniques for constructing linear constraint sets are described and the dependence of LCCM on a particular parameter, the modulus factor, is noted. The goal of this chapter is to define the LCCM performance criterion and describe all of its variables, constants and parameters.

Chapter 4 analyzes the LCCM approach for a narrowband array application. The analysis covers correlated and uncorrelated signals in both noise-present and noise-absent environments. Modulus factor effects are also examined. Numerical techniques are used to evaluate analytically intractable problems. The goal of this chapter is to establish the utility of the LCCM method and justify derivation of an adaptive LCCM algorithm.

Chapter 5 derives adaptive algorithms based on the LCCM performance criterion. The algorithms are derived using the method of Stochastic Gradient Descent (SGD.) Stability and convergence issues are addressed via comparison with related SGD algorithms. The goal of this chapter is to derive stable and robust recursions for the adaptive solution to the LCCM performance criterion.

Chapter 6 simulates the LCCM algorithm in two applications. One is a narrowband array and the other is a fractionally spaced equalizer. The signal environment includes co-channel interference and multipath distortion. In the array

simulation, LCCM performance is compared with two related algorithms. The goal of this chapter is to demonstrate the potential of the LCCM algorithm and identify its drawbacks.

Chapter 7 reviews the results and contributions of this dissertation. In addition, extensions and future directions of research are identified.

1.2 Summary of Contributions

Because of the way this dissertation is organized, most of the work after Chapter 2 is in a sense a contribution. The LCCM approach is new and accordingly each result that is presented regarding LCCM is new as well. Without being too redundant, the following is a list of general contributions that are made in this dissertation.

- A unified presentation and critical review of existing concepts for untrained adaptive signal processing.
- The introduction of a new untrained approach: Linearly Constrained Constant Modulus.
- The analysis of LCCM local minima for correlated and uncorrelated signal environments, with noise present and noise absent.
- A convergence and stability analysis of the LCCM adaptive algorithm.
- The application of LCCM to a multi-channel problem: a narrowband array, and to a single channel problem: a fractionally spaced equalizer.

1.3 Notation

Unless otherwise specified, all signals are assumed to be in sampled form with the time index (n). The index n also specifies iteration number in adaptive recursions.

Scalar signals (real or complex) are in lower case italic type.

Examples: $x(n)$, $y(n)$.

Vectors are in lower case bold Roman type. They are assumed to be column vectors unless otherwise specified. They may be fixed or time-varying.

Examples: $\mathbf{x}(n)$, $\mathbf{w}_q$.

Matrices are in upper case bold Roman type.

Examples: $\mathbf{R}_{xx}$, $\mathbf{W}_s$.

Integers are in upper case italic type.

Examples: N, L.

Scalar parameters are generally Greek letters.

Examples: $\alpha_{\mathbf{w}}$, σ_g^2, δ.

The dagger symbol † denotes a conjugate transpose operation.

Examples: $\mathbf{w}^\dagger\mathbf{x}$, $\mathbf{w}^\dagger\mathbf{R}_{xx}\mathbf{w}$.

Expectation is denoted by either $E\{\cdot\}$ or $\overline{\{\cdot\}}$

Conventions

- The dB measure of power assumes a unity power signal as reference, i.e., 0 dB corresponds to a power of 1, 20 dB corresponds to a power of 100, etc.

- Gains given in dB are always power gains, e.g. a -3 dB gain on a 20 dB signal results in a new signal with power equal to 50 (17 dB.)

- All signals, whether they are noise or information bearing, are assumed to be zero mean.

Abbreviations

LMS	Least Mean Square
CM	Constant Modulus
UCM	Unconstrained Constant Modulus
LCCM	Linearly Constrained Constant Modulus
LCMP	Linearly Constrained Minimum Power
DF	Decision Feedback
SGD	Stochastic Gradient Descent
QPSK	Quaternary Phase Shift Keying
FM	Frequency Modulation
SOI	Signal Of Interest
ISI	InterSymbol Interference

Chapter 2

Background

This chapter describes the problem of interest and critically reviews some of the existing signal processing techniques that have been used to solve it. Its basic purpose is to narrow the scope of the problem. Along the way, notation is developed that is used throughout the remaining chapters.

2.1 The General Problem

At its highest level, the problem of interest is to improve the performance of communication receivers that operate in unknown and non-stationary signal environments. This is much too broad a topic for a dissertation, of course, so the scope of the problem must be narrowed. This is not done by restricting the application of interest to array receivers or channel equalizers. Nor is it done by restricting the modulation types to be analog or digital. Instead, the problem is made manageable by narrowing the investigation to a specific approach to dealing with the problem that has wide application: adaptive processing. Even adaptive

processing is too broad a research area, however, so in this chapter the type of adaptive approach is restricted.

2.1.1 Channel Estimation

In many communication environments today, the distribution of the noise field and the characteristics of the channel are largely unknown. Without these parameters, it is difficult to design an optimum receiver because the structure of an optimum receiver obviously depends on the channel and noise that it operates in. One approach to this problem is to simply measure the channel characteristics and the noise field and then build the optimum receiver accordingly.

Unfortunately, the noise field and channel characteristics are time-varying for many modern communication environments. This is particularly true for mobile communication where multipath fading can be extremely rapid. Since the environment is non-stationary, the parameters of the channel and noise field must be estimated or tracked and then down-loaded to an adaptive receiver structure. If the estimates keep pace with the changes in the environment then these adaptive receivers, although complex, are considered to be optimal. A good example of a technique of this type is the adaptive maximum likelihood receiver[1]. Another is the RAKE receiver which uses channel sounding techniques to determine a filter that optimally combines resolvable multipath [2].

The problem with this type of adaptive receiver is that *direct* estimation of channel and noise field parameters is not possible in many communication scenarios of interest. In some cases, estimates of either the channel or the noise parameters are available but knowledge of both is necessary to design the optimal receiver filter. Although direct estimation cannot always be employed, there are

other approaches to adaptive receiver design. The most common technique is to optimize a cost function that is related to the performance of the receiver. In a sense, this approach is *indirect* because it compensates for channel and noise effects without estimating the relevant parameters directly. The most common performance criteria are based on the mean square or least square error formulations.

2.1.2 Mean Square and Least Square Error Formulations

It should be noted that direct measurement of channel or noise parameters is often implemented using the techniques of mean or least square estimation. In this section, however, the minimum squared error methods are considered only as optimization criteria for the coefficients of an adaptive processor.

Part of the appeal of the minimum squared error methods is that they have a solid theoretical foundation: the mean square approach is based on discrete Weiner filtering and the least squares approach is related to Kalman filtering. The mean square error criterion is

$$\min_{\mathbf{w}} E\left\{|d(n) - y(n)|^2\right\} \tag{2.1}$$

where $y = \mathbf{w}^\dagger \mathbf{x}$ is the inner product of the weight vector $\mathbf{w}$ and the data vector $\mathbf{x}$ and d is the desired or reference signal. For stationary signal environments, the unique weight vector that solves (2.1) is

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1}\mathbf{p} \tag{2.2}$$

where $\mathbf{R}_{xx}$ is the autocorrelation matrix of the data and $\mathbf{p}$ is the cross-correlation vector of the data $\mathbf{x}$ and the conjugate of reference signal d. This weight vector minimizes the mean square error between signals $d(n)$ and $y(n)$.

LMS: Least Mean Square

As stated in the last section, the environment of interest is unknown and time varying and therefore $\mathbf{R}_{xx}$ and $\mathbf{p}$ are in general unknown and only approximately constant over a time interval. This means that the optimum solution cannot be fixed at (2.2) once and for all time, it must adapt and follow the changes in the signal environment. The LMS adaptive algorithm is a method that recursively estimates the optimum vector (2.2) and can track its variations if they change at a relatively slow rate [3].

There are many ways to derive the LMS adaptive recursion. One of the simplest starts with the steepest descent recursion [4],

$$\mathbf{w}(n+1) = (\mathbf{I} - \mu\mathbf{R}_{xx})\mathbf{w}(n) + \mu\mathbf{p}. \tag{2.3}$$

This recursion converges to (2.2) for an appropriately chosen value of μ. In LMS adaptation, the statistical quantities $\mathbf{R}_{xx}$ and $\mathbf{p}$ are replaced by their instantaneous approximations to yield the following recursion

$$\begin{aligned} \mathbf{w}(n+1) &= (\mathbf{I} - \mu\mathbf{x}\mathbf{x}^\dagger)\mathbf{w}(n) + \mu d^*(n)\mathbf{x}(n) && (2.4) \\ &= \mathbf{w}(n) + \mu e^*(n)\mathbf{x}(n) && (2.5) \end{aligned}$$

where

$$e^*(n) = d^*(n) - y^*(n). \tag{2.6}$$

LMS is easily the most widely applied and analyzed adaptive algorithm. This is for good reason, it is a simple and robust method with well understood convergence characteristics. If a good reference signal $d(n)$ is available, an adaptive receiver that incorporates the LMS filter will compensate for channel distortion

and additive interference. Indeed, this is the case in the telephone network today where LMS-based equalizers are used extensively. Figure 2.1 shows a block diagram of LMS that symbolically shows the feedback of the error signal into the update of the weight vector.

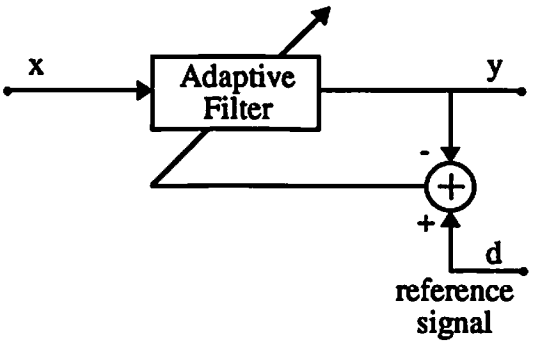

Figure 2.1: Block diagram of the LMS algorithm

Least Squares

A least squares adaptive filter uses the same reference signal $d(n)$ as LMS but updates its weights quite differently. At each iteration it minimizes the sum of the squared errors rather than the mean square error,

$$\sum_{n=1}^{n} |d(k) - y(k)|^2. \tag{2.7}$$

Each term in the summation (2.7) corresponds to one linear equation in a system of n linear equations that grows by one equation for every iteration step n. A least squares filter finds the least squares solution to this system of overdetermined equations at every iteration step n. This approach requires more computation than LMS but it usually converges at a faster rate. In a sense it is "converged" at every step since it solves for the least squared error at every step n. In a stationary

environment, however, both the LMS and least squares approaches will converge to (2.2) given the same reference signal [5].

2.1.3 Reference Signal Availability

The mean square and least square error methods are widely used in a variety of applications. In many practical problems of interest, however, a reference signal may not always be available. This is especially true in communication applications. The signal $d(n)$ that is assumed to be known at the receiver is typically the signal that one is attempting to receive in the first place and is therefore not available for training the adaptive processor. This difficulty is resolved in practice by a variety of techniques.

The simplest and most common approach is to initialize the receiver filter with a training sequence of short duration. This sequence, which is either stored or generated locally at the receiver, is synchronized with an identical sequence that is sent by the transmitter and the receiver filter is then adapted accordingly. In this case the receiver indeed knows exactly what the transmitter has sent. It is assumed that by the time the training sequence ends the adaptive filter will have converged and the weights can then be fixed for the normal transmission that follows. In some cases the weights are continually updated after the training period using the technique of decision feedback which is discussed in Section 2.2.1. Another way to allow continuous adaptation is to multiplex a predetermined reference signal onto the transmitted signal. In this way, an adaptive receiver can be continuously updated throughout the signal transmission.

As noted, in many instances either direct or indirect methods can be used for channel estimation. This is the case with both the start-up/training sequence

and multiplex reference examples described above. An important distinction is that indirect estimation takes both the channel and additive noise into account while direct estimation typically just measures the channel transfer function; the characteristics of the noise field are usually assumed to be known.

In many communication systems, a reference or training signal is unavailable. This dissertation is concerned with such systems. A variety of approaches have been developed over the past 30 years to adapt single and multi-channel filters without a reference signal. Most can be divided into two categories: those that synthesize or extract some sort of the reference signal from the data that can then be used in a minimum square error criterion and those that abandon the m.s.e. criterion altogether in favor of some alternative performance measure. In Section 2.2, methods from both categories will be reviewed and critically examined on a conceptual basis.

2.2 Existing Approaches

The block diagrams included in the following descriptions of untrained adaptive methods are not intended to represent actual implementations but to express the signal processing concepts that underly each technique. For this reason, no distinction is made between single and multi-channel inputs. The input data vector $\mathbf{x}$ could come from an array of sensors, a tapped delay line or perhaps an array of tapped delay lines.

2.2.1 Decision Feedback

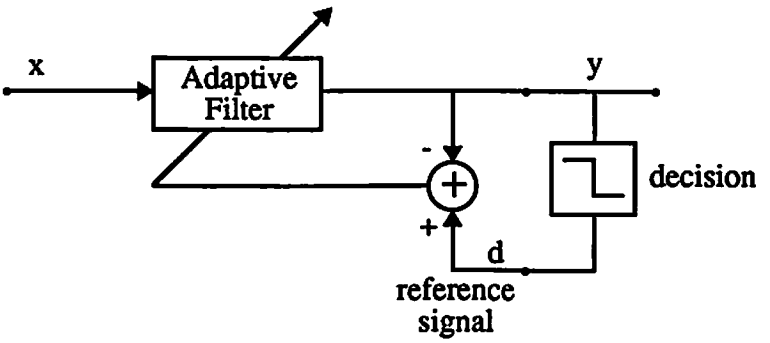

Figure 2.2: Block diagram of the Decision Feedback algorithm

Figure 2.2 shows a block diagram of the decision feedback adaptive filter. This popular channel equalization technique takes its name from the fact that the detected symbols are fed back into the filter update algorithm [6]. Essentially, these detected symbols are used as the reference signal for a mean square or least square adaptive algorithm. When the symbol decisions are made correctly the reference signal is perfect, but when errors are made the feedback nature of the algorithm can make the filter diverge.

In many equalizers, decision direction is combined with the training sequence approach. A training sequence is first used to initialize the filter and greatly reduce the error rate. Then normal data transmission begins and the filter is adapted in a decision directed mode to allow for a non-stationary environment. In short, the training sequence frees the decision feedback filter from having to bootstrap itself out of the initial, possibly high error stage. This combined approach is very effective in those applications where a training sequence is available. Of course, in the applications of interest here a start-up training sequence is unavailable. (Also, decision feedback cannot be used on analog modulated signals such as FM.)

2.2.2 Adaptive Line Enhancement/Cancellation

Adaptive line enhancement/cancellation [3] is another approach for generating a reference signal from the received signal which is completely different from decision feedback in concept. In this method, it is assumed that a narrowband signal is present in broadband noise and the autocorrelation of the noise is zero at lag values greater than Δ_T. Because of its narrower bandwidth, the autocorrelation of the signal is assumed to be non-zero at this value of Δ_T. That is,

$$r_{ss}(\Delta_T) \neq 0 \tag{2.8}$$

$$r_{nn}(\Delta_T) = 0. \tag{2.9}$$

In other words, a delay of Δ_T will decorrelate the broadband noise but it will not completely decorrelate the narrowband signal.

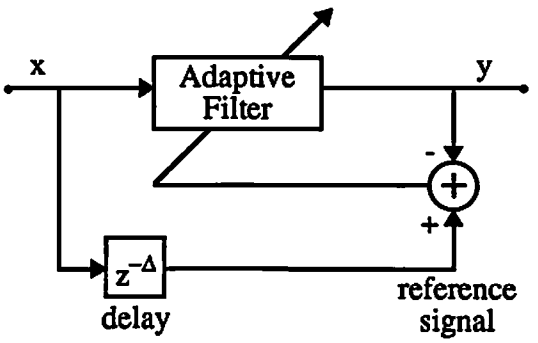

Figure 2.3: Block diagram of the Adaptive Line Enhancer

Figure 2.3 shows the adaptive line canceller in block diagram form. The received signal serves as the input to the adaptive filter and after a delay it also serves as the reference input. The delay Δ shown in the figure includes the decorrelation delay Δ_T as well as the delay or order of the adaptive filter Δ_L, i.e.,

$$\Delta = \Delta_T + \Delta_L \tag{2.10}$$

The difference between the adaptive line enhancer and the canceller is that the canceller uses the error signal as output (Fig. 2.3) whereas the enhancer uses the actual adaptive filter output.

Although the noise component in the reference signal is uncorrelated with the noise component in the filter output, the presence of the noise in the reference input can impair the dynamic performance of the adaptive filter. This effect is discussed in [7]. In any case, adaptive line enhancement/cancellation cannot be used in many signal environments of interest due to its restrictive correlation requirements.

2.2.3 Self-Coherence Restoral

The decorrelation approach of adaptive line enhancement is taken one step further in the Self-Coherence Restoral (SCORE) method The block diagram of the SCORE filter in Figure 2.4 shows that, in addition to the time delay, a local oscillator is placed in the reference signal path. This modulation adds the dimension of frequency to the dimension of time in the noise decorrelation process. [8].

Self-Coherence is a term from the theory of cyclo-stationary signal processing. A signal $x(t)$ is self-coherent at frequency α if its cross-correlation with its conjugate $x^*(t)$ frequency shifted by α is nonzero for some time lag τ, i.e.,

$$\mathbf{R}_{xx}{}^{\alpha}(\tau) = E\{x(t+\tau/2)x^*(t-\tau/2)e^{-j2\pi\alpha t}\} \neq 0 \qquad (2.11)$$

for some value of τ. The choice of the name SCORE is perhaps misleading since the self-coherence property is not really in need of restoral as the name would suggest. Indeed, self-coherence is the invariant property of the signal that the adaptive filter is based upon.

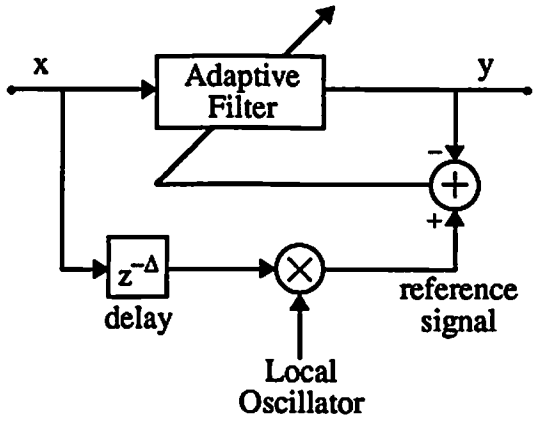

Figure 2.4: Block diagram of the SCORE algorithm

Since most communication signals are self-coherent at some frequency α, the combination of a delay and a frequency shift is an effective way to decorrelate the noise while only partially decorrelating the signal. SCORE is thus a useful generalization of the adaptive line enhancement approach which used only a time delay to decorrelate the noise. It is not without drawbacks, however. SCORE, like adaptive line enhancement, suffers from a noisy reference signal which adversely affects convergence. In addition, its performance is degraded in environments that include multipath or other signals which are self-coherent at the same frequency.

2.2.4 Noise Cancellation With Available Noise Reference

In noise cancellation, the filtering problem is turned around so that rather than using a replica of the signal of interest (SOI) as the desired signal, a representation of the noise is used instead [3]. This is shown symbolically in Figure 2.5 where it should be noted that the adaptive filter now operates on the reference signal rather than on the received signal. This approach is much different than the

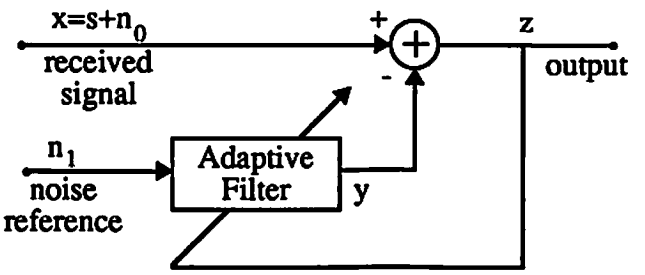

Figure 2.5: Block diagram of the adaptive noise canceller

previously described methods; the noise term n_0 in the received signal is now predicted rather than the signal term s itself and the error signal z is now the "output" instead of the usual adaptive filter output y.

The noise reference n_1, which is measured or extracted from the environment, is assumed to be correlated with the noise n_0 in the received signal but uncorrelated with the signal component s. Thus with appropriate filtering of the noise reference, the noise term n_0 in the received signal can be subtracted out in a mean square sense. The resultant output z becomes a best mean square estimate of the signal term s. To see this, first observe that the the output (error) can be written as

$$z = s + n_0 - y \tag{2.12}$$

and the mean square error then becomes

$$\begin{aligned} E\{z^2\} &= E\{s^2 + (n_0 - y)^2 + 2s(n_0 - y)\} && (2.13) \\ &= E\{s^2\} + E\{(n_0 - y)^2\} && (2.14) \end{aligned}$$

where the cross term is zero since s is uncorrelated with n_0 and y. Note that the signal power $E\{s^2\}$ will be unaffected when the filter is adjusted to minimize $E\{z^2\}$. In other words, minimizing $E\{z^2\}$ is equivalent to minimizing $E\{(n_0-y)^2\}$

and y is thus a mean square estimate of the noise term n_0. Moreover, minimizing $E\{(n_0 - y)^2\}$ is equivalent to minimizing $E\{(z - s)^2\}$ since

$$(z - s) = (n_0 - y.) \tag{2.15}$$

Therefore, adapting the filter to minimize the total output power $E\{z^2\}$ is tantamount to making the output z a mean square estimate of the signal s for the given filter structure and for the given noise reference.

Noise references are available in many signal processing applications but unfortunately they are difficult to obtain in most communications environments. This is because it is difficult to measure the noise alone without any of the signal s leaking into the measurement. If s is somehow represented in the noise reference, then both the noise reference n_1 and the filter output y would be correlated with the signal term s and signal cancellation would result. What is desired, then, is a technique whereby a noise reference can be extracted from the received signal $s+n_0$ rather than measured from the environment. Such a method, called linearly constrained power minimization, is described in the next section.

2.2.5 Linearly Constrained Power Minimization

Linearly constrained power minimization was not originally proposed as an extension or generalization of the noise cancellation approach but as a way to adapt an antenna array without a desired signal [9]. By imposing a set of linear constraints on the tap weights, the response of an adaptive array can be such that the signal of interest passes through to the output without distortion. For example, the constraints can be constructed such that the gain in a particular direction is constant over all frequencies so any signal that arrives from that direction is passed

without distortion to the output. Then when the output power is minimized the interfering sources are nulled and, at the same time, the constraints ensure that the signal of interest is unaffected.

Formally, this approach is stated as follows:

$$\min_{\mathbf{w}} \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w} \qquad \text{subject to} \qquad \mathbf{C}^{\dagger}\mathbf{w} = \mathbf{f}, \tag{2.16}$$

where the expression $\mathbf{C}^{\dagger}\mathbf{w} = \mathbf{f}$ specifies a system of linear equations that the weight vector $\mathbf{w}$ must satisfy. The solution to this constrained minimization problem is given by

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1}\mathbf{C}(\mathbf{C}^{\dagger}\mathbf{R}_{xx}^{-1}\mathbf{C})^{-1}\mathbf{f}. \tag{2.17}$$

A simple derivation and further elaboration can be found in [10].

At first, it may not seem that linearly constrained power minimization can be viewed as a noise cancellation approach, an interpretation implied at the end of the last section. This is because the interpretation lies not in the problem *formulation* but in a particular adaptive *implementation* [11]. This implementation, called the Generalized Sidelobe Canceller (GSC) structure, will be discussed in more detail in Section 5.1.2. In concept, the GSC is represented by Figure 2.6. The GSC extracts a noise reference signal n_1 from the received signal and uses it in the same fashion as the noise canceller in Figure 2.5.

Even though it solves the noise reference problem, the technique of linearly constrained minimum power (LCMP) has its own shortcomings. There are basically three problems associated with LCMP. First, suitable constraints cannot always be constructed for the signal processing task at hand. Second, even when constraints can be found LCMP is very sensitive to inaccuracies in these constraints

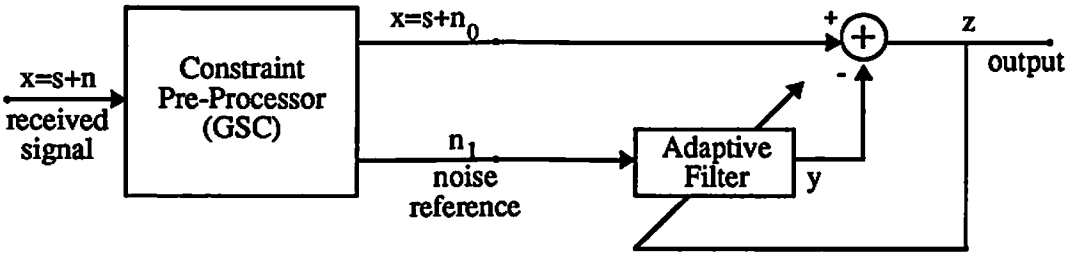

Figure 2.6: Block diagram of the LCMP algorithm

[12]. Finally, the method fails in multipath and other correlated interference environments for the same reason that the noise canceller fails when its noise reference is correlated with the signal of interest.

Of the three, the failure in correlated interference environments is probably the most significant since it will occur even if the constraint set is perfect. This failure, results in signal cancellation and has been analyzed and addressed in several papers [13, 14]. It is best understood using a simple and admittedly artificial example. Consider a narrowband signal A impinging on a linear array at 0 degrees and another signal B arriving at 45 degrees. A single constraint is employed to ensure that the gain of the array is unity at 0 degrees. When B is uncorrelated with A, a null is placed on signal B and the minimum power is that of signal A alone. However, when B is correlated with A, the minimum power can be reduced much further. For example, if signal B is identical to A the minimum power solution from (2.17) will specify that a gain of -1 be placed on signal B which is arriving from 45 degrees. This causes the signal at the output to be completely cancelled even though the constraint has still been met. This is certainly the minimum power solution but it is hardly acceptable to the end user. Although this example used a second signal that was perfectly correlated (it was

in this case the *same* signal), the effect is similar for interfering signals that are only partially correlated.

It should be noted that the LCMP algorithm is not restricted to multi-channel or array applications. Recent work describes a single channel filtering application of LCMP [15].

2.2.6 P-Vector

The P-Vector algorithm [16] does not fit neatly into the two broad categories of untrained adaptive algorithms in that it does not make use of an extracted reference signal and it does not use an alternate performance measure. Instead, the P-Vector algorithm is derived by manipulating the steepest descent recursion in the same way that was used to derive the LMS algorithm (Section 2.1.2.)

Recall that the minimum mean square error solution, given by (2.2), can be found adaptively using a stochastic approximation of the steepest descent recursion, namely LMS. LMS assumes that both $\mathbf{R}_{xx}$ and $\mathbf{p}$ are unknown but that a desired signal is available. The P-Vector approach assumes that $\mathbf{R}_{xx}$ and the desired signal are unknown but that the cross-correlation vector $\mathbf{p}$ is known. With an approximation along the same lines as LMS, the P-Vector algorithm is easily derived from the true steepest descent recursion

$$\mathbf{w}(n+1) = (\mathbf{I} - \mu\mathbf{R}_{xx})\mathbf{w}(n) + \mu\mathbf{p} \tag{2.18}$$

by replacing $\mathbf{R}_{xx}$ with its instantaneous equivalent,

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \mu\mathbf{x}\mathbf{x}^{\dagger}\mathbf{w}(n) + \mu\mathbf{p} && (2.19) \\ &= \mathbf{w}(n) + \mu\left(\mathbf{p} - y^{*}(n)\mathbf{x}(n)\right). && (2.20) \end{aligned}$$

It is interesting to note that given the same *a priori* information, i.e. a known $\mathbf{p}$, the LCMP method can be used to adapt a filter to within a scale factor of the same optimum solution [15]. This is achieved by requiring the inner product between the weight vector $\mathbf{w}$ and $\mathbf{p}$ to be constrained to a constant, e.g. unity. The LCMP solution in (2.17) then reduces to

$$\mathbf{w}_{opt} = \frac{1}{\alpha}\mathbf{R}_{xx}^{-1}\mathbf{p} \tag{2.21}$$

where $\alpha = \mathbf{p}^{\dagger}\mathbf{R}_{xx}^{-1}\mathbf{p}$ is a constant scale factor. Except for this scale factor, this is the expression for the minimum mean square error filter (2.2.) The P-Vector method can thus be cast as a LCMP approach to the minimum mean square error problem.

Normally, calculation of the $\mathbf{p}$ vector assumes that the desired signal and interference are uncorrelated. This simplifies the calculation, of course, but it is not always an accurate assumption, particularly when the environment includes multipath. If correlated interference is not taken into account in the calculation of the $\mathbf{p}$ vector, signal cancellation may result. And so the P-Vector approach, like LCMP, is not well suited for multipath and other correlated interference environments.

2.2.7 Unconstrained Constant Modulus (UCM)

For many modulation types such as FM and QPSK, the signal that is transmitted possesses the constant envelope property[1]. The signal that is received, however, may have lost this property due to common sources of distortion such as multipath fading, interference, and channel fluctuations. The constant modulus approach to

[1]This assumes that the QPSK signal has not been processed with a pulse shaper.

untrained adaptive filtering operates under the assumed principle that by restoring the constant envelope property to the signal, the sources of distortion can be removed or corrected [17, 18, 19, 20]. Since these same sources usually reduce SNR (e.g. in FM) or increase the probability of error (e.g. in QPSK), UCM processing can offer a significant improvement in performance.

The UCM approach is unlike the other untrained algorithms because it can combat multipath distortion. (In fact, one of its original applications was the correction of multipath distortion for analog FM signals [19].) Most other adaptive methods that do not use a reference signal will fail when the interference is correlated with the signal of interest (SOI) as it is in the multipath case. For instance, the LCMP approach does not need a reference signal but, as stated in Section 2.2.5, it is susceptible to signal cancellation in a multipath environment.

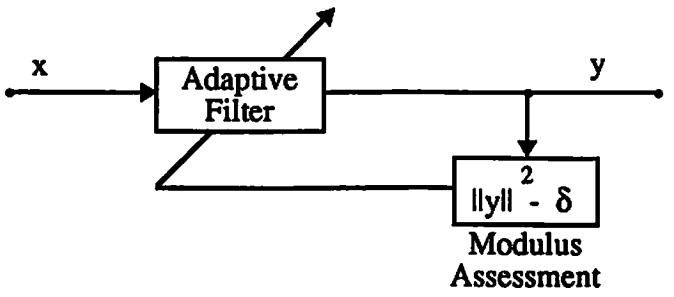

Figure 2.7: Block diagram of the CM algorithm

The goal of UCM adaptive processing is to find a weight vector w that minimizes fluctuations in the complex envelope of the filter output $y(n)$ where

$$y(n) = \mathbf{w}^{\dagger}\mathbf{x}(n). \tag{2.22}$$

Accordingly, the first step is to define a cost function $\mathcal{J}$ that penalizes variations in the output envelope. One such function is

$$\mathcal{J}(\mathbf{w}) = E\left\{\left(|y(n)|^2 - \delta\right)^2\right\}. \tag{2.23}$$

This cost function $\mathcal{J}(\mathbf{w})$ simply measures the average variation of the output envelope from an arbitrary constant value of δ.

The problem with this cost function (and all others that have been proposed so far) is that it is non-convex and no closed form expression for $\mathbf{w}$ that minimizes (2.23) exists. Even though the weight vector that minimizes $\mathcal{J}(\mathbf{w})$ cannot be calculated directly, an adaptive algorithm can still be derived that minimizes $\mathcal{J}(\mathbf{w})$ recursively.

The first step is to take the gradient of (2.23) with respect to $\mathbf{w}$,

$$\nabla_{\mathbf{w}} = 4E\left\{\left(|y(n)|^2 - \delta\right) y^*(n)\mathbf{x}(n)\right\} \tag{2.24}$$

The UCM adaptive algorithm is created by taking the stochastic or "instantaneous" approximation of (2.24) and using it in the standard steepest descent iteration,

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\left(|y(n)|^2 - \delta\right) y^*(n)\mathbf{x}(n) \tag{2.25}$$

where the scale factor on the gradient has been absorbed into μ. UCM is thus a SGD algorithm just like LMS.

Although UCM has been applied to a variety of signal processing problems, it has several shortcomings which limit its use in many applications. One is that the algorithm may "capture" a CM interferer rather than the CM signal of interest. That is, when more than one signal in the environment is constant modulus, each by itself is a valid output and so the filter will pass one and null the

others. Determining which signal will be passed and which ones will be filtered out is very difficult and has only been addressed for idealized cases. In these ideal cases, capture depends on the relative powers of the signals and the filter initial conditions [21]. The capture problem with UCM is far from being completely understood, however, and is still largely unpredictable.

Another problem is that *a priori* signal or environment knowledge is difficult to incorporate into the algorithm. For example, if the direction of arrival or carrier frequency of the SOI is known, the only way of exploiting this information with UCM at present is by fixed pre-filtering or by manipulating the filter initial conditions. These are unwieldy approaches to the problem and, furthermore, they are difficult to implement if the environment is non-stationary.

2.3 Summary and Problem Statement

Each of the previously described untrained approaches offers an alternative to the proven methods of minimum mean square and least square adaptive filtering. The price of operating without a reference signal, though, is a compromise in performance or a restriction to specific signal environments: decision feedback fails if the error rate is too high, a noise reference is not always available for noise cancellation, and all the other techniques except UCM are degraded by correlated interference and multipath. Although UCM can be used in multipath environments, it has the undesirable property of locking onto other co-channel CM signals rather than the CM signal of interest.

This chapter began with the general problem of how to improve the performance of communication receivers that operate in unknown and non-stationary

signal environments. Of the many possible techniques that can improve reception, the choice was limited to adaptive methods. Next, the type of adaptive method was restricted to those that do not require a reference signal and are thus untrained. The type of untrained adaptive method was further restricted to those that can operate in severe co-channel interference and multipath. It was then shown that the existing untrained adaptive approaches are not well suited for such difficult signal environments. In the next chapter, an untrained adaptive technique that is suitable for such signal environments is developed.

Chapter 3

LCCM Development

This chapter introduces LCCM: a Linearly Constrained Constant Modulus approach to untrained adaptive filtering [22]. To begin, the relationship between LCCM and two existing untrained approaches is described. This is followed by a general discussion of linear constraints and a description of the specific types of constraints that are suitable for LCCM. The chapter ends with a discussion of some of the special considerations of the LCCM method.

3.1 LCCM Concept

The LCCM approach combines features of LCMP and UCM into a new adaptive approach. As such, it can be motivated from two points of view: as a modification of LCMP or as an extension of UCM.

3.1.1 A Modification of LCMP

As noted in Section 2.2.5, one problem observed in LCMP adaptive arrays is the phenomena of signal cancellation [13]. Under certain conditions such as correlated interference, the SOI power at the output of the adaptive array can actually be less than it would be if the array was not adaptive but simply steered to the SOI direction with fixed weights. Rather than improving signal quality, in this case the adaptive processor is making the situation even worse.

The phenomena of signal cancellation depends on a variety of parameters such as SNR, SIR, and SWNR. At its root, however, it is a consequence of the underlying cost function that is being minimized in LCMP processing. In LCMP (Section 2.2.5), the output power is minimized subject to a set of linear constraints on the weight vector. Note that if the weights are unconstrained, the power is minimized by simply setting the weights to zero; a degenerate solution. The constraints, then, are designed to ensure that SOI is not nulled when the total power is minimized. However, if there is inaccuracy in the constraints or if correlated interference is present, it is possible that the SOI will be cancelled along with the interference.

The root of the problem of signal cancellation is the power minimization criterion. To avoid the signal cancellation problem, one approach is to modify the criterion of power minimization. In a sense, minimization of power is an inherently non-robust criterion because errors in the constraint equation usually lead directly to distortion of the SOI. This degradation is not necessarily graceful either: small errors can lead to large losses in signal power, particularly in high SNR cases.

The signal cancellation problem motivates the use of performance criteria other than minimum power. One alternative criterion is the constant modulus cost function of Section 2.2.7. Replacement of the power minimization criterion with the CM cost function yields the problem statement of LCCM,

$$\min_{\mathbf{w}} E\left\{\left(|y|^2-\delta\right)^2\right\} \qquad \text{subject to} \qquad \mathbf{C}^\dagger\mathbf{w}=\mathbf{f}. \tag{3.1}$$

In many ways, LCCM has the potential to be more robust than LCMP. For instance, if the constraint equation is eliminated from LCCM the result is regular UCM which is still useful. In contrast, LCMP is useless without the constraint equation. It is reasonable to assume, then, that the LCCM method is less sensitive to constraint inaccuracy. In addition, since the CM cost function was originally developed for multipath correction, LCCM is appropriate for correlated interference environments.

3.1.2 An Extension of UCM

In the previous section, the LCCM approach was motivated from the LCMP point of view. It is also possible to develop the LCCM concept from a UCM point of view. That is, to begin with UCM and extend it to LCCM.

A significant drawback of UCM is the capture problem discussed in Section 2.2.7. The capture problem is due to the ambiguous nature of the UCM cost function. The cost function, which penalizes variations in the complex envelope of the filter output, is meaningful only when a single CM signal is received. If more than one CM signal is present on the channel, an adaptive implementation of UCM will pass one signal and suppress the others. When the signal-of-interest

is suppressed and a CM interferer is passed to the output, the UCM processor is captured by the interference [21].

Analysis of the UCM capture problem has shown that it depends on the relative powers of the various signals and the initial conditions of the adaptive filter coefficients [21]. In recent work, the capture problem has been addressed by controlling the relative signal powers [23] or by specifying the filter initial conditions [24]. However, analysis and resolution of the capture problem is very difficult in general and has only been attempted for relatively simple cases.

Although capture depends on signal powers and initial conditions at the surface, the problem is fundamentally one of *degrees of freedom.* The number degrees of freedom available to an adaptive weight vector is equal to the dimension of the basis of all possible weight vector candidates. For an unconstrained weight vector, the dimension of the basis is simply equal to the number of weights N. These N degrees of freedom are employed by the adaptive processor to reject interference and compensate for distortion. In an UCM filter the N degrees of freedom also allow the weight vector to capture interfering signals rather than the SOI. If the degrees of freedom available to a UCM weight vector can be reduced in an appropriate fashion it follows that the capture problem may be mitigated.

One of the most effective ways to reduce the degrees of freedom available to an adaptive filter is to impose linear constraints on the filter coefficients as is done with LCMP. In LCMP, the constraint equation keeps the weight vector from degenerating to the zero vector. In the same way, a system of constraint equations can be employed to prevent the UCM weight vector from being captured by CM interference. The LCCM method is the combination of the these constraint

equations and the constant modulus cost function,

$$\min_{\mathbf{w}} E\left\{\left(|y|^2-\delta\right)^2\right\} \qquad \text{subject to} \qquad \mathbf{C}^{\dagger}\mathbf{w}=\mathbf{f}. \tag{3.2}$$

The constraints also offer a means to incorporate *a priori* signal and environment knowledge into the UCM method.

3.1.3 LCCM Cost Function

The conceptual development of Sections 3.1.1 and 3.1.2 showed that LCCM has a dual motivation that stems from the drawbacks associated with the UCM and LCMP methods. In this section LCCM is described more formally. The first step in the development of LCCM is to define a set of M linear constraints on the N dimensional weight vector $\mathbf{w}$. Each constraint is expressed in terms of a constraint vector $\mathbf{c}_k$ and a corresponding scalar constraint value f_k. This can be compactly written in matrix notation as

$$\mathbf{C}^{\dagger}\mathbf{w}=\mathbf{f} \tag{3.3}$$

where the matrix $\mathbf{C}$ contains the linearly independent $\mathbf{c}_k$ vectors as columns and $\mathbf{f}$ is the vector of constraint values f_k. The determination of the elements of $\mathbf{C}$ and $\mathbf{f}$ is described in Section 3.2.

The general LCCM problem statement is

$$\min_{\mathbf{w}} E\left\{||y|^p-\delta|^q\right\} \qquad \text{subject to} \qquad \mathbf{C}^{\dagger}\mathbf{w}=\mathbf{f}. \tag{3.4}$$

where p and q are positive integers and $\delta>0$. The cost function in (3.4) penalizes variations in the modulus of y for all values of p and q but for practical purposes described below p and q are restricted to be 1 or 2.

Although the p-q CM cost functions [25] are not the only functions that penalize variations in output modulus, they are probably the simplest. The selection of p and q is usually based on issues that arise in the adaptive implementations of constant modulus based algorithms. These issues include computational complexity and dynamic range. For instance, the 2-2 cost function calculates the fourth moment of $|y|$. This would require a very large dynamic range in a fixed point implementation. In this dissertation, only the 2-1 and 2-2 cost functions are used for LCCM. No attempt is made here to determine the "optimum" CM cost function. Recent results [26] suggest that the differences between cost functions are relatively minor.

In any case, the 2-1 and 2-2 cost functions are equivalent from an analytical point of view. The 2-1 function is simply the square root of the 2-2 function. Since the square root operation is a monotonically increasing functional, the finite local minimums of the 2-1 and 2-2 functions correspond exactly.

Since they are equivalent, the 2-2 version of LCCM is used in Chapter 4 because it greatly simplifies the analysis of the LCCM. The 2-1 LCCM version is used in Chapters 5 and 6 because it leads to the most stable adaptive form. Unless otherwise noted, the phrase "LCCM algorithm" in this work refers to the adaptive or recursive solution to (3.4) with $p = 2$ and $q = 1$.

3.2 Constraint Selection

This section describes in general terms the construction of practical constraint sets from typical *a priori* information. More detailed discussions can be found in [27, 10].

3.2.1 Response Constraints

The simplest and most widely used type of constraint is the response constraint. A response constraint ensures that some portion of the response of an adaptive processor is fixed, regardless of subsequent adaptation. Typically, the constraint controls the frequency response of an adaptive FIR filter or the spatial/temporal response of an adaptive array.

One commonly employed response constraint in the adaptive signal processing literature is the spatial gain constraint on a narrowband array. This constraint, an example of a point constraint, fixes the gain and phase response of the array to some specified value for a direction of interest. The general constraint matrix C in this case consists of a single vector c called a steering or direction vector. The general response vector f reduces to a scalar in this case that fixes the array gain in the direction of interest. For a given angle and array configuration, this constraint is very simple to calculate [28].

In a similar manner, a point response constraint can be imposed in the frequency domain. For instance, the magnitude and phase response of an adaptive FIR filter at a particular frequency can be constrained. The constraint vector c in this case is simply the Fourier vector which corresponds to the frequency of interest and the constraint value f corresponds to the desired magnitude and phase response at that frequency. When greater control over the response is necessary, the constraint vectors $\mathbf{c}_k$ can be grouped together to form the constraint matrix C.

Other methods of response control include combined spatio-temporal point constraints for broadband arrays [27] and derivative constraints [29]. A derivative constraint simply forces the magnitude response of the array/filter to have

a derivative of zero at some point. This ensures that the response is flat in a neighborhood around the particular point and adds a degree of robustness to the adaptive processor.

3.2.2 Signal Subspace Constraints

In some way, every response constraint incorporates *a priori* knowledge of the signal environment. This happens directly when an angle of arrival or a frequency of interest is used to construct a constraint vector and indirectly when signal environment uncertainty suggests the use of a derivative constraint. In the next section, another representation of *a priori* signal knowledge is described. It is called the *signal subspace* representation and it is based on the second order statistics of the received data. The signal subspace approach is a powerful and general technique for constructing constraints and hence reducing degrees of freedom. Although this approach is more general than the response constraint method, many response constraints can be interpreted from a signal subspace point of view.

The motivation for signal subspace or eigenvector constraints arose in broadband antenna array processing [27]. In a typical broadband array problem, the response is constrained to be flat over a specified frequency bandwidth for some look direction of interest. The fact that this can require a large number of constraints led to work on efficient methods for constraint design.

The starting point for the signal subspace approach to constraint design is the auto-correlation matrix of the signal of interest. If the general data vector $\mathbf{x}$ is given by

$$\mathbf{x} = \mathbf{s} + \mathbf{n} \tag{3.5}$$

where s and n are the uncorrelated signal and noise[1] components respectively, then the SOI auto-correlation matrix is given by

$$\mathbf{R}_{ss} = E\{\mathbf{s}\mathbf{s}^{\dagger}\} \tag{3.6}$$

The span of $\mathbf{R}_{ss}$ is called the signal subspace.

Since $\mathbf{R}_{ss}$ is Hermitian and non-negative definite by construction, an orthogonal eigenvector decomposition is possible [30],

$$\mathbf{R}_{ss} = \mathbf{U}\Lambda\mathbf{U}^{\dagger}, \tag{3.7}$$

where the orthogonal eigenvectors are the columns of $\mathbf{U}$ and the entries of the diagonal matrix Λ are ordered from largest λ_1 to smallest λ_N. The eigenvectors in $\mathbf{U}$ that correspond to the non-zero eigenvalues are an orthogonal basis for the span of the matrix $\mathbf{R}_{ss}$. Moreover, they form an efficient basis [27]. This is useful if $\mathbf{R}_{ss}$ is full rank and an efficient, reduced rank approximation of the signal subspace is necessary.

In the signal subspace approach, the columns of constraint matrix $\mathbf{C}$ are the M most significant eigenvectors from the decomposition (3.7.) The constraint value vector $\mathbf{f}$ is designed based on considerations of SOI power or distortion [27, 10]. For maximum SOI power $f_i = \lambda_i$ where λ_i is the eigenvalue of the corresponding eigenvector.

If the SOI auto-correlation matrix $\mathbf{R}_{ss}$ has a rank of one then the eigenvector constraint can usually be viewed as a response constraint. For example, the rank one $\mathbf{R}_{ss}$ matrix of an SOI impinging on a narrowband array has one significant eigenvector: the direction vector of the SOI. Recall that the direction vector is

[1]The noise n includes both white noise and any interference.

what is used to construct a simple point response constraint. Similarly, a complex sinusoid SOI as received by a tapped delay line also yields a rank one $\mathbf{R}_{ss}$. The sole eigenvector of this $\mathbf{R}_{ss}$ matrix corresponds to a frequency domain point constraint.

Use of the signal subspace/eigenvector approach does not normally imply that the matrix $\mathbf{R}_{ss}$ is measured from the environment. Instead, $\mathbf{R}_{ss}$ is calculated using available *a priori* knowledge such as direction of arrival, carrier frequency, or modulation type. In the case of a rank one signal source, $\mathbf{R}_{ss}$ is basically a conceptual aid since the eigenvector constraint can typically be considered a point constraint without resorting to subspace considerations. This is not always the case, however. In Section 6.3 a subspace constraint that does not correspond to a point constraint is described and used in a simulation.

3.2.3 LCCM Constraints

The purpose of the constraint equation in the LCCM approach is different than it is for LCMP. With LCMP, the purpose is to sufficiently represent the signal subspace in order to preserve the SOI when output power is minimized. The purpose of the constraint equation in LCCM is to prevent capture of the algorithm by CM interference.

Although the constraint motivation is different for LCCM and LCMP, the actual constraint vectors used are the same. The only difference is that the number of constraint vectors necessary for control of LCCM can be much less than needed for LCMP. This is because the signal subspace need not be completely represented in order to prevent capture. For instance, the rank of $\mathbf{R}_{ss}$ for a broadband signal such as FM as received by a tapped delay line will certainly be greater than one but it is conjectured that a single point constraint at its carrier frequency (or a

constraint on the principal eigenvector of $\mathbf{R}_{ss}$ if the carrier is suppressed) is enough to ensure that LCCM will lock onto the FM signal rather than be captured by a strong co-channel CM interferer. In many common LCCM scenarios, however, the signal subspace is of rank one. The analysis in Chapter 4 and the simulations in Chapter 6 assume that $\mathbf{R}_{ss}$ is rank one and the signal subspace can be represented with one constraint.

3.3 Modulus Factor Selection

There are three design quantities in the LCCM problem statement (3.4) other than p and q These are $\mathbf{C}$, $\mathbf{f}$ and δ. The constraint matrix $\mathbf{C}$ and the vector $\mathbf{f}$ were described in Section 3.2. The parameter δ, termed the *modulus factor*, has not received much attention. Recall that δ specifies the modulus value about which any complex envelope variations are penalized. This section describes the role of δ in more detail.

3.3.1 Fixed Modulus Factor

The optimal value of δ is very simple to calculate since its value does not directly depend on the constraint equation. Expanding the 2-2 cost function yields

$$\delta^2 - 2\delta E\left\{|y|^2\right\} + E\left\{|y|^4\right\}, \tag{3.8}$$

which can easily be minimized by setting δ such that

$$\delta_{min} = E\left\{|y|^2\right\} = \mathbf{w}^\dagger \mathbf{R}_{xx}\mathbf{w}. \tag{3.9}$$

Note that any properly scaled weight vector $\mathbf{w}$ that satisfies $\mathbf{C}^\dagger\mathbf{w} = \mathbf{f}$ will minimize the 2-2 cost function with respect to δ. The optimum δ, then, is given by the

path loss. It is also important in non-cooperative schemes where the power may fluctuate by design.

The adaptive determination of δ is discussed in more detail in Chapters 5 and 6. For now, δ is treated as just another free variable of minimization.

It is interesting to note that the role of δ is not discussed in much detail in the literature dealing with UCM. In most cases it is simply set to unity, i.e. $\delta = 1$. Although UCM is not sensitive to δ in the same way as LCCM, the choice of δ still has an effect on the ultimate performance of the adaptive implementation of UCM. This is because the value of δ, together with the relative powers of the received signals and the initial filter coefficients, plays a role in the UCM capture phenomenon. The only published reference found for the optimum value of δ is for an UCM equalizer application [18]. Interestingly enough, the signal in this case had Quadrature Amplitude Modulation (QAM) and did not possess the constant envelope property.

Although it affects capture, the role of δ in UCM is on the whole less significant than it is in LCCM. With UCM the weight vector $\mathbf{w}$ is unconstrained and the corresponding output power can be increased or decreased by simply scaling $\mathbf{w}$. So, for instance, if the output power is much greater than δ then the weight vector can be scaled down so the power matches δ. This offers a much greater reduction in cost then reducing the envelope variation alone and makes UCM less sensitive to the the choice of δ. Since the UCM weights are unconstrained, however, the value of δ cannot be made adaptive. This is because the UCM weight vector $\mathbf{w}$ and δ would both go to zero since they are both unconstrained.

3.4 Summary

This chapter introduced the LCCM approach and described how it has the potential to solve some of the drawbacks of LCMP and UCM. LCCM can viewed as a constrained version of UCM or alternatively as a CM version of LCMP. LCCM has the potential to improve robustness for both methods. The CM criterion in place of the minimum power criterion may mitigate the signal cancellation problem of LCMP and the linear constraints may eliminate the capture problem of UCM. Constraints also allow incorporation of *a priori* knowledge of the signal or environment characteristics.

The type of constraints that could be potentially used in LCCM are the same as those used in LCMP. Out of the many possible, this chapter described response constraints and the more general signal subspace constraints. The motivation for the use of constraints in LCCM is different than their use in LCMP. In LCMP they are to prevent signal distortion due to signal cancellation. With LCCM they are to prevent capture by interfering CM signals.

The modulus factor, δ, was shown to be an important parameter in the LCCM problem statement. The modulus factor can be treated as fixed parameter or as a free variable of minimization. When it is fixed, its value has a direct impact on the weight vector $\mathbf{w}$ that minimizes the overall cost. As such, the selection of a fixed δ affects the performance of the eventual adaptive LCCM implementation of Chapter 5. Making the modulus factor a free variable eliminates the sensitivity to δ in the cost function but it also adds another variable to the adaptive implementation.

The general claim of this chapter is that LCCM has advantages over UCM and LCMP. This claim is explored in Chapter 6 where adaptive implementations of UCM, LCMP and LCCM are compared via simulation. Before resorting to simulations, however, the LCCM problem statement can be studied analytically. The next chapter analyzes the LCCM method for a narrowband array application.

Chapter 4

Analysis of LCCM

4.1 Introduction

As is the case with the UCM cost function (2.23), direct analysis of the LCCM problem statement (3.4) is very difficult in general. That is, it is virtually impossible to solve for the closed form weight vector $\mathbf{w}$ that minimizes cost. The foremost reason for this is the inherent difficulty of calculating the the expectation operation $E\{\cdot\}$ which involves higher order moments. Even when the moments can be calculated, as they are in [25] for the UCM cost function, the result is still a non-convex function of a complex vector that is more or less intractable.

To make analysis tractable, published work on the UCM cost function invokes a narrowband array assumption on the data [26, 31]. This allows the analysis to proceed indirectly in terms of the gains of narrowband signals rather than directly on the elements of the weight vector. In other words, the analytical problem is to determine what combination of gains on the narrowband signals result in local minimums of the UCM cost function. This reduces the complexity of the analysis

considerably since the optimization is carried out in a smaller dimension with real gain variables rather than complex weight vector elements.

The same narrowband assumption used to simplify analysis of the UCM cost function can be used to study the local minima of LCCM. In addition, the constraint equation makes this task even simpler for LCCM than it is for the UCM cost function. This is because the constraint equation fixes the response of the array to have unity gain in the direction of the signal of interest. In this way, the dimension of the problem is reduced by one with respect to the UCM cost function. (The gain on the SOI is a free variable in the UCM cost function.) A reduction in the number of free variables by one is not as insignificant as it sounds. In many cases of interest it transforms a function of 3 variables into a more manageable function of 2 variables.

4.2 Narrowband Array Model

The analysis of this section follows closely the work published in [31] on the unconstrained UCM cost function. The first step is to establish the necessary notation. The output beam $y(n)$ of an N sensor adaptive array can be written as

$$y(n) = \mathbf{w}^{\dagger}\mathbf{x}(n) \tag{4.1}$$

where $\mathbf{w}$ is the vector of array weights and $\mathbf{x}(n)$ is the vector of signals received at the array sensor elements.

The data vector $\mathbf{x}(n)$ represents the sum of $M+1$ narrowband[1] signals. Since each signal is narrowband, it can be modeled as a scalar signal multiplied by a

[1]The bandwidth conditions necessary for the narrowband assumption are discussed in texts such as [32, 33].

fixed direction vector $\mathbf{d}(\theta)$. The decomposition of the received data vector into narrowband components is

$$\mathbf{x}(n) = a(n)\mathbf{d}(\theta_a) + b_1(n)\mathbf{d}(\theta_{b1}) + b_2(n)\mathbf{d}(\theta_{b2}) + \cdots + b_M(n)\mathbf{d}(\theta_{bM}) \tag{4.2}$$

where $a(n)$ is the signal of interest and $b_1(n) \ldots b_M(n)$ are the interfering signals. Note that for this model the signal environment is assumed to be noiseless, i.e. no ambient noise is present.

The direction vector $\mathbf{d}(\theta)$ consists of the complex phase shifts that correspond to a plane wave impinging on the array from an angle θ. For compactness, the set of $M+1$ linearly independent direction vectors are grouped as columns of a matrix $\mathbf{T}$,

$$\mathbf{T} \triangleq \left[\begin{array}{cccc} \mathbf{d}(\theta_a) & \mathbf{d}(\theta_{b1}) & \mathbf{d}(\theta_{b2}) & \cdots \quad \mathbf{d}(\theta_{bM}) \end{array} \right] \tag{4.3}$$

and the set of $M+1$ signals are grouped to form the vector $\mathbf{s}(n)$,

$$\mathbf{s}(n) \triangleq \left(\begin{array}{c} a(n) \\ b_1(n) \\ b_2(n) \\ \vdots \\ b_M(n) \end{array} \right) = \left(\begin{array}{c} a(n) \\ \mathbf{b}(n) \end{array} \right). \tag{4.4}$$

This allows the data vector $\mathbf{x}(n)$ to be rewritten as

$$\mathbf{x}(n) = \mathbf{T}\mathbf{s}(n) \tag{4.5}$$

and the output to be rewritten as

$$y(n) = \mathbf{w}^\dagger \mathbf{T}\mathbf{s}(n). \tag{4.6}$$

In this analysis, the system of linear equations $\mathbf{C}^\dagger\mathbf{w} = \mathbf{f}$ in the LCCM problem statement (3.4) is assumed to consist of a single constraint which ensures that the

gain of the array in the direction of the signal of interest is unity, i.e.,

$$\mathbf{w}^\dagger \mathbf{d}(\theta_a) = 1. \tag{4.7}$$

The gain of the array in the direction, θ_{b_i}, of each interfering signal is defined as

$$v_i^* \triangleq \mathbf{w}^\dagger \mathbf{d}(\theta_{b_i}). \tag{4.8}$$

Therefore, the product of the weight vector $\mathbf{w}$ and the direction or steering matrix $\mathbf{T}$ is given by

$$\mathbf{w}^\dagger \mathbf{T} = \begin{pmatrix} 1 & v_1^* & v_2^* & \cdots & v_M^* \end{pmatrix} \tag{4.9}$$

$$= \begin{pmatrix} 1 & \mathbf{v}^\dagger \end{pmatrix} \tag{4.10}$$

and the output (4.6) can be rewritten as

$$y(n) = a(n) + \mathbf{v}^\dagger \mathbf{b}(n). \tag{4.11}$$

In a sense, a change of variables has taken place. The vector of array weights $\mathbf{w}$ has been replaced by the vector of gains on the interfering signals $\mathbf{v}$. It is assumed that for any $\mathbf{v}$ there is at least one $\mathbf{w}$ such that (4.10) is satisfied. For this to be true, the $N-1$ degrees of freedom available to the weights must be greater than or equal to the M interfering signals. (The constraint equation (4.7) reduces the original N degrees of freedom to $N-1$.)

4.2.1 Multiple Independent Signals

In the previous section, the LCCM problem statement

$$\min_{\mathbf{w}} E\left\{\left||y|^2 - \delta\right|^q\right\} \qquad \text{subject to} \qquad \mathbf{C}^\dagger \mathbf{w} = \mathbf{f}. \tag{4.12}$$

was transformed from a function of the weight vector $\mathbf{w}$ to the gain vector $\mathbf{v}$,

$$\min_{\mathbf{v}} E\left\{\left||y|^2-\delta\right|^q\right\} \tag{4.13}$$

Note that the constraint equation has been eliminated from (4.13), the vector $\mathbf{v}$ is unconstrained. For the analysis in this chapter, only the 2-2 version of the LCCM cost expression will be used, i.e., $q=2$.

It is convenient to define

$$\mathcal{J}(\mathbf{v}) \triangleq E\left\{\left||y(n)|^2-\delta\right|^2\right\} \tag{4.14}$$

as the new transformed LCCM cost function. In the analysis to follow, the expectation operator $E\{\cdot\}$ is replaced by an overbar and the time index (n) is dropped. With these changes and with $\delta=1$, the cost function (4.14) becomes

$$\mathcal{J}(\mathbf{v}) = \overline{\left|\left(a(n)+\mathbf{v}^\dagger\mathbf{b}(n)\right)\left(a(n)+\mathbf{v}^\dagger\mathbf{b}(n)\right)^*-1\right|^2} \tag{4.15}$$

$$= \overline{\left||a|^2-\mathbf{v}^\dagger\mathbf{b}\mathbf{b}^\dagger\mathbf{v}-a^*\mathbf{v}^\dagger\mathbf{b}-a\mathbf{b}^\dagger\mathbf{v}-1\right|^2} \tag{4.16}$$

which can be expanded to yield

$$\begin{aligned}\mathcal{J}(\mathbf{v}) = {} & 1-\overline{|a|^2}-\mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger}\mathbf{v}-\mathbf{v}^\dagger\overline{\mathbf{b}a^*}-\overline{a\mathbf{b}^\dagger}\mathbf{v} \\ & -\overline{|a|^2}+\overline{|a|^4}+\mathbf{v}^\dagger\overline{|a|^2\mathbf{b}\mathbf{b}^\dagger}\mathbf{v}+\mathbf{v}^\dagger\overline{\mathbf{b}|a|^2a^*}+\overline{|a|^2a\mathbf{b}^\dagger}\mathbf{v}-\mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger}\mathbf{v} \\ & +\mathbf{v}^\dagger\overline{|a|^2\mathbf{b}\mathbf{b}^\dagger}\mathbf{v}+\mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger\mathbf{v}\mathbf{v}^\dagger\mathbf{b}\mathbf{b}^\dagger}\mathbf{v}+\mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger\mathbf{v}\mathbf{v}^\dagger\mathbf{b}a^*}+\mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger\mathbf{v}\mathbf{b}^\dagger\mathbf{v}a} \\ & -\mathbf{v}^\dagger\overline{\mathbf{b}a^*}+\mathbf{v}^\dagger\overline{\mathbf{b}|a|^2a^*}+\mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger\mathbf{v}\mathbf{v}^\dagger\mathbf{b}a^*}+\overline{(a^*)^2(\mathbf{v}^\dagger\mathbf{b})^2}+\mathbf{v}^\dagger\overline{|a|^2\mathbf{b}\mathbf{b}^\dagger}\mathbf{v} \\ & -\overline{a\mathbf{b}^\dagger}\mathbf{v}+\overline{|a|^2a\mathbf{b}^\dagger}\mathbf{v}+\mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger\mathbf{v}a\mathbf{b}^\dagger}\mathbf{v}+\mathbf{v}^\dagger\overline{|a|^2\mathbf{b}\mathbf{b}^\dagger}\mathbf{v}+\overline{a^2(\mathbf{b}^\dagger\mathbf{v})^2}\end{aligned} \tag{4.17}$$

To simplify (4.17), the following assumptions are made regarding the statistical properties of signals a and b_i:

1 Statistically Independence

2 Zero Mean

3 Unity Power

4 Uncorrelated I-Q Components

5 Equal Power I-Q Components

Assumption 1, which is the most important, implies that

$$\overline{f(a)f(\mathbf{b})} = \overline{f(a)}\,\overline{f(\mathbf{b})}. \tag{4.18}$$

The combination of assumptions 1 and 2 reduces the number of cross terms in (4.17) considerably. Assumptions 4 and 5 reduce the number of terms even further. To see this let

$$a = a_{re} + ja_{im} \tag{4.19}$$

Then, since a is zero mean and has uncorrelated I-Q components of equal power

$$\overline{a^2} = \overline{a_{re}^2} - \overline{a_{im}^2} + 2j\overline{a_{re}a_{im}} = 0 \tag{4.20}$$

$$\overline{(a^*)^2} = \overline{a_{re}^2} - \overline{a_{im}^2} - 2j\overline{a_{re}a_{im}} = 0. \tag{4.21}$$

Making use of the assumptions, the cost function (4.17) can be compactly rewritten as

$$\mathcal{J}(\mathbf{v}) = k_a - 1 + 2\mathbf{v}^\dagger\mathbf{v} + \mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger\mathbf{v}\mathbf{v}^\dagger\mathbf{b}\mathbf{b}^\dagger}\mathbf{v} \tag{4.22}$$

where the parameter

$$k_a \triangleq \frac{\overline{|a|^4}}{\left(\overline{|a|^2}\right)^2}. \tag{4.23}$$

is called the *kurtosis* of signal a.

In order to evaluate the last term in (4.22), the inner products are replaced by summations

$$\mathbf{v}^\dagger\overline{\mathbf{b}\mathbf{b}^\dagger\mathbf{v}\mathbf{v}^\dagger\mathbf{b}\mathbf{b}^\dagger}\mathbf{v} = \overline{\sum_{i=1}^{M} v_i^* b_i \sum_{j=1}^{M} v_j b_j^* \sum_{k=1}^{M} v_k^* b_k \sum_{l=1}^{M} v_l b_l^*} \tag{4.24}$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{l=1}^{M} v_i^* v_j v_k^* v_l \overline{b_i b_j^* b_k b_l^*} \tag{4.25}$$

Making use of all 5 assumptions, there are only 3 combinations of i, j, k and l that have non-zero expectation. They are:

$$\begin{array}{cc} \underline{\text{Case}} & \underline{\text{Corresponding Term}} \\ i = j = k = l & \sum_{i=1}^{M} k_i |v_i|^4 \\ i = j, k = l & \sum_{i=1}^{M}\sum_{j=1}^{M} |v_i|^2 |v_j|^2 \quad (i \neq k) \\ i = l, j = k & \sum_{i=1}^{M}\sum_{j=1}^{M} |v_i|^2 |v_j|^2 \quad (i \neq j) \end{array} \tag{4.26}$$

where

$$k_i \triangleq \frac{\overline{|b_i|^4}}{\left(\overline{|b_i|^2}\right)^2} \tag{4.27}$$

is the kurtosis of each interfering signal b_i. All other combinations of i, j, k and l lead to zero terms.

The following identity

$$(\mathbf{v}^\dagger \mathbf{v})^2 - \sum_{i=1}^{M} |v_i|^4 = \sum_{i=1}^{M}\sum_{j=1}^{M} |v_i|^2 |v_j|^2 \quad (i \neq j) \tag{4.28}$$

together with (4.26) allows equation (4.22) to be rewritten as

$$\mathcal{J}(\mathbf{v}) = k_a - 1 + 2\mathbf{v}^\dagger \mathbf{v} + 2(\mathbf{v}^\dagger \mathbf{v})^2 + \sum_{i=1}^{M} (k_i - 2)|v_i|^4. \tag{4.29}$$

Note that the cost function $\mathcal{J}(\mathbf{v})$ only depends on the magnitude $|v_i|$ of the complex gain on each interfering signal. Therefore, the cost function can be redefined as function of the real vector $\mathbf{u}$,

$$\mathbf{u} \triangleq \begin{pmatrix} |v_1| \\ |v_2| \\ \vdots \\ |v_M| \end{pmatrix} \tag{4.30}$$

and the resultant cost function is

$$\mathcal{J}(\mathbf{u}) = k_a - 12\mathbf{u}^\dagger\mathbf{u} + 2(\mathbf{u}^\dagger\mathbf{u})^2 + \sum_{i=1}^{M}(k_i - 2)u_i^4. \tag{4.31}$$

The stationary points of $\mathcal{J}(\mathbf{u})$ are found by setting its gradient equal to zero,

$$\nabla_{\mathbf{u}}\mathcal{J} = 4\mathbf{u} + 8\mathbf{u}^\dagger\mathbf{u}\mathbf{u} + 4\begin{pmatrix} (k_1 - 2)u_1^3 \\ (k_2 - 2)u_2^3 \\ \vdots \\ (k_M - 2)u_M^3 \end{pmatrix} = \mathbf{0}. \tag{4.32}$$

Clearly, $\mathbf{u} = \mathbf{0}$ is a solution to (4.32) and is therefore a stationary point of (4.31). Other solutions may exist however. In fact, additional stationary points exist for any vector $\mathbf{u}$ whose elements u_i satisfy

$$4 + 8\mathbf{u}^\dagger\mathbf{u} + 4(k_i - 2)u_i^2 = 0 \tag{4.33}$$

which can be rewritten as

$$u_i = \sqrt{\frac{-(1 + 2\mathbf{u}^\dagger\mathbf{u})}{k_i - 2}}. \tag{4.34}$$

In general, any combination of zero elements and elements that satisfy (4.34) produces a vector $\mathbf{u}$ that is a stationary point of (4.31). Note that since $1 + 2\mathbf{u}^\dagger\mathbf{u} > 0$ then $k_i < 2$ must be true for a real solution to (4.34) to exist. In other words, if a real and non-zero solution for u_i exists then the kurtosis of the i^{th} interfering signal must be less than 2.

Without a loss of generality, the M elements of $\mathbf{u}$ can be reordered such that the L non-zero values are at the top of the vector.

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_L \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{4.35}$$

The next step is to solve for u_i in terms of the kurtosis values. To begin, note that

$$\begin{aligned} \mathbf{u}^\dagger\mathbf{u} &= u_1^2 + u_2^2 + \cdots + u_L^2 && (4.36) \\ &= -\frac{(1+2\mathbf{u}^\dagger\mathbf{u})}{k_1 - 2} - \frac{(1+2\mathbf{u}^\dagger\mathbf{u})}{k_2 - 2} - \cdots - \frac{(1+2\mathbf{u}^\dagger\mathbf{u})}{k_L - 2} && (4.37) \\ &= -(1+2\mathbf{u}^\dagger\mathbf{u})\sum_{j=1}^{L}\frac{1}{k_j - 2}. && (4.38) \end{aligned}$$

Solving for $\mathbf{u}^\dagger\mathbf{u}$ yields

$$\mathbf{u}^\dagger\mathbf{u} = \frac{-\sum_{j=1}^{L}\frac{1}{k_j-2}}{1+2\sum_{j=1}^{L}\frac{1}{k_j-2}} \tag{4.39}$$

and it is straightforward to show that

$$1 + 2\mathbf{u}^\dagger\mathbf{u} = \frac{1}{1+2\sum_{j=1}^{L}\frac{1}{k_j-2}}. \tag{4.40}$$

Substituting (4.40) into (4.34) allows the gain vector elements to be rewritten as

$$u_i = \begin{cases} \sqrt{\frac{-\beta_i}{1+2\sum_{j=1}^{N}\beta_j}} & 1 \le i \le L \\ 0 & L < i \le M \end{cases} \tag{4.41}$$

where

$$\beta_i = \frac{1}{k_i - 2} \tag{4.42}$$

Equation (4.41) specifies the stationary points of (4.31) in terms of the kurtosis values of the interfering signals $b_i(n)$. To determine if these stationary points correspond to local minimums it is necessary to calculate the Hessian matrix.

Hessian Calculation

The Hessian is defined as the matrix of partial derivatives

$$\mathbf{H}(\mathcal{J}) \triangleq \left[\frac{\partial \mathcal{J}}{\partial_{u_i}\partial_{u_j}}\right]. \tag{4.43}$$

For the cost function $\mathcal{J}(\mathbf{u})$ given in (4.31), the Hessian is given by

$$\mathbf{H} = (4 + 8\mathbf{u}^\dagger\mathbf{u})\mathbf{I} + 16\mathbf{u}\mathbf{u}^\dagger + \sum_{i=1}^{M}(k_i - 2)u_i^2\mathbf{e}_i\mathbf{e}_i^\dagger \tag{4.44}$$

where $\mathbf{e}_i$ is the i^{th} basis vector, i.e. a vector of zeros except for a value of 1 in the i^{th} position. Note that $\mathbf{H}$ evaluated at $\mathbf{u} = \mathbf{0}$ leads to a diagonal, positive definite matrix and therefore $\mathbf{u} = \mathbf{0}$ is a local minimum.

For other local minimums to exist, $\mathbf{H}$ must be positive definite for some $\mathbf{u}$. The simplest positive definite check for the Hessian of (4.44) is to calculate its eigenvalues directly. Using (4.40) and (4.41), $\mathbf{H}$ evaluated at $\mathbf{u}$ can be rewritten as

$$\mathbf{H} = \frac{4}{1 + 2\sum_{j=1}^{L}\beta_j}\mathbf{I} + 16\mathbf{u}\mathbf{u}^\dagger - \frac{12}{1 + 2\sum_{j=1}^{L}\beta_j}\mathbf{e}_i\mathbf{e}_i^\dagger. \tag{4.45}$$

Recall that the vector $\mathbf{u}$ has $M - L$ zero elements. This allows the Hessian matrix to be partitioned as follows,

$$\mathbf{H} = 16\mathbf{u}\mathbf{u}^\dagger + \begin{bmatrix} -\frac{8}{1+2\sum_{j=1}^{L}\beta_j}\mathbf{I}_{L\times L} & \mathbf{0} \\ \mathbf{0} & \frac{4}{1+2\sum_{j=1}^{L}\beta_j}\mathbf{I}_{(M-L)\times(M-L)} \end{bmatrix} \tag{4.46}$$

By inspection, one eigenvector of $\mathbf{H}$ is $\mathbf{u}$. The corresponding eigenvalue is found as follows

$$\mathbf{Hu} = \left(16\mathbf{u}^\dagger\mathbf{u} - \frac{8}{1+2\sum_{j=1}^{L}\beta_j}\right)\mathbf{u} \tag{4.47}$$

$$= \left(16\mathbf{u}^\dagger\mathbf{u} - 8(1+2\mathbf{u}^\dagger\mathbf{u})\right)\mathbf{u} \tag{4.48}$$

$$= -8\mathbf{u}. \tag{4.49}$$

Clearly $\mathbf{H}$ is not positive definite since it has at least one negative eigenvalue, $\lambda = -8$. Therefore the stationary points given by (4.41), for $L \neq 0$ correspond to either local maximums or saddle points. The only local minimum is when $L = 0$ or equivalently when $\mathbf{u} = \mathbf{0}$.

The $\mathbf{u} = \mathbf{0}$ solution corresponds to a weight vector $\mathbf{w}$ that nulls all interfering signals. This is the desired result. Further, no assumption on the nature of the signal of interest is necessary, i.e., $a(n)$ may or may not be constant modulus. This is in contrast to the analysis in [31] where it was shown that the UCM cost function has local minimums for any signal with kurtosis less than 2. Not only does the LCCM approach have only one local minimum, there are no modulus or kurtosis restrictions on the SOI. The implication here is that LCCM can be applied to a broader class of independent signals than those that possess the CM property.

4.2.2 Variable Modulus Factor

In the previous section, the minima of the LCCM cost function were examined for the case where the modulus factor δ was fixed to unity. In this section, δ is assumed to be a variable. The same signal model used in the previous section is used here except that only one independent interfering signal is present rather

than M signals. The reason for this is to keep the number of variables to a manageable number. In any case, it is no great loss of generality to assume that only one interfering signal exists. The focus here is to investigate the effect of a variable δ in the LCCM problem statement.

Since only one interfering signal is present, the array output can be written as

$$y(n) = a(n) + \gamma b(n) \tag{4.50}$$

where $a(n)$ is the SOI and γ is the gain on the interfering signal $b(n)$. Again, since the constraint fixes the gain on $a(n)$ to be unity, the LCCM cost function for this case is

$$\begin{aligned} \mathcal{J}(\gamma,\delta) &\triangleq E\left\{\left(|y|^2-\delta\right)^2\right\} && (4.51)\\ &= E\left\{\left(|a|^2+|\gamma|^2|b|^2+a\gamma^*b^*+a^*\gamma b-\delta\right)^2\right\} && (4.52)\end{aligned}$$

The cost function is now a function of two variables: the complex gain γ and the real parameter δ.

Expanding (4.52) yields the following expression,

$$\begin{aligned} \mathcal{J}(\gamma,\delta) = E\Big\{&|a|^4+|\gamma|^2|a|^2|b|^2+|a|^2a\gamma^*b^*+|a|^2a^*\gamma b-\delta|a|^2\\ &+|a|^2|\gamma|^2|b|^2+|\gamma|^4|b|^4+|\gamma|^2|b|^2a\gamma^*b^*+|\gamma|^2|b|^2a^*\gamma b-|\gamma|^2|b|^2\delta\\ &+|a|^2a\gamma^*b^*+|\gamma|^2|b|^2a\gamma^*b^*+a^2(\gamma^*)^2(b^*)^2+|a|^2|\gamma|^2|b|^2-a\gamma^*b^*\delta\\ &+|a|^2a^*\gamma b+|\gamma|^2|b|^2a^*\gamma b+|a|^2|\gamma|^2|b|^2+(a^*)^2\gamma^2b^2-a^*\gamma b\delta\\ &-|a|^2\delta-|\gamma|^2|b|^2\delta-a\gamma^*b^*\delta-a^*\gamma b\delta+\delta^2\Big\}. \end{aligned}$$

Since a and b are independent, the previous expression can be reduced to

$$\begin{aligned} \mathcal{J}(\gamma,\delta) = &\overline{|a|^4}+4|\gamma|^2\overline{|a|^2}\,\overline{|b|^2}+|\gamma|^4\overline{|b|^4}+\delta^2-2\delta\overline{|a|^2}\\ &-2\gamma^2\delta\overline{|b|^2}+(\gamma^*)^2\overline{a^2}\,\overline{(b^*)^2}+\gamma^2\overline{(a^*)^2}\,\overline{b^2}. \end{aligned} \tag{4.53}$$

The last two terms in (4.53) can be eliminated by assuming that, as in Section 4.2.1, the two signals have equal power and uncorrelated I-Q components. An additional simplification is possible since the cost function only depends on the magnitude of the complex gain γ. By defining $\alpha \triangleq |\gamma|$, the cost function can be rewritten as a function of two real variables

$$\mathcal{J}(\alpha, \delta) = \overline{|a|^4} + 4\alpha^2\sigma_a^2\sigma_b^2 + \alpha^4\overline{|b|^4} + \delta^2 - 2\sigma_a^2\delta - 2\alpha^2\sigma_b^2\delta. \tag{4.54}$$

In order to find the stationary points of the cost function, the partial derivatives are found,

$$\frac{\partial \mathcal{J}}{\partial \alpha} = 8\alpha\sigma_a^2\sigma_b^2 + 4\alpha^3\overline{|b|^4} - 4\alpha\sigma_b^2\delta \tag{4.55}$$

$$\frac{\partial \mathcal{J}}{\partial \delta} = 2\delta - 2\sigma_a^2 - 2\alpha^2\sigma_b^2. \tag{4.56}$$

A stationary point exists when both partial derivatives equal zero. Clearly this occurs when $\alpha = 0$ and $\delta = \sigma_a^2$. To check for other stationary points, α is assumed to be non-zero and is divided from (4.55.) The system of non-linear equations then becomes

$$2\sigma_a^2\sigma_b^2 + \alpha^2\overline{|b|^4} - \sigma_b^2\delta = 0 \tag{4.57}$$

$$\delta - \sigma_a^2 - \alpha^2\sigma_b^2 = 0. \tag{4.58}$$

By defining the kurtosis of signal b to be

$$k_b \triangleq \frac{\overline{|b|^4}}{(\sigma_b^2)^2} \tag{4.59}$$

equation (4.57) can be rewritten as

$$2\sigma_a^2 - \delta + \alpha^2\sigma_b^2 k_b = 0 \tag{4.60}$$

Substituting (4.58) into (4.60) yields

$$2\sigma_a^2 - \left(\sigma_a^2 + \alpha^2\sigma_b^2\right) + \alpha^2\sigma_b^2 k_b = 0 \tag{4.61}$$

which can be easily solved for α,

$$\alpha = \pm\sqrt{\frac{\sigma_a^2}{\sigma_b^2(1-k_b)}}. \tag{4.62}$$

Since the kurtosis of any signal always satisfies $k \geq 1$, no real, non-zero value of α can correspond to a stationary point. Therefore, only one stationary point exists: when $\alpha = 0$ and $\delta = \sigma_a^2$.

Next, the Hessian matrix is formed to verify that the stationary point $(0, \sigma_a^2)$ is a local minimum. The second partials are

$$\frac{\partial^2 \mathcal{J}}{\partial \alpha \partial \delta} = -4\alpha\sigma_b^2 \tag{4.63}$$

$$\frac{\partial^2 \mathcal{J}}{\partial \delta \partial \alpha} = -4\alpha\sigma_b^2 \tag{4.64}$$

$$\frac{\partial^2 \mathcal{J}}{\partial \alpha^2} = 8\sigma_a^2\sigma_b^2 + 12\alpha^2\overline{|b|^4} - 4\sigma_b^2\delta \tag{4.65}$$

$$\frac{\partial^2 \mathcal{J}}{\partial \delta^2} = 2. \tag{4.66}$$

The Hessian evaluated at $\alpha = 0$ and $\delta = \sigma_a^2$ is then

$$\mathbf{H} = \begin{bmatrix} 8\sigma_a^2\sigma_b^2 & 0 \\ 0 & 2 \end{bmatrix} \tag{4.67}$$

which is clearly positive definite and therefore the stationary point $(0, \sigma_a^2)$ is a local minimum.

Again as in the previous section, the only local minimum of the cost function corresponds to a null on the interfering signal. Like Section 4.2.1, this is true whether or not $a(n)$ is CM. The difference here is that δ is variable not fixed and only one interfering signal is present. The implication of this is that if both the weights and δ are made adaptive, they will both converge to the desired solution using standard SGD techniques.

The next extension to this analysis is to remove the independent signal assumption and allow the signals to be correlated. This analysis is important because the correlated signal environment is a prime motivation for the development of LCCM.

4.3 Correlated Signal Model

Correlated signals make analysis of LCCM very difficult. When the signals are independent, most terms in the cost function drop out. In the correlated case, all of these terms must be retained adding significant complexity. Non-zero correlation also means that the cost function is no longer a function of the magnitude of the gain on the interferer but of the phase as well. Having to deal with a complex gain rather than its magnitude adds one more variable to the analysis.

Correlated signals present such a problem in this analysis that the signals $a(n)$, $b(n)$ and the gain γ are assumed to be real not complex. This makes the problem tractable analytically and its solution offers insight into the more general complex case. However, this assumption results in a significant loss of generality due to the fact that the real correlated signal model no longer corresponds to a true narrowband signal as it impinges on an array. (It should be noted that single channel applications exist in communication systems where a similar real signal model is applicable.)

Let the real output be written as

$$y = a + \gamma b \tag{4.68}$$

With this model, the LCCM cost function is

$$\begin{aligned}
\mathcal{J}(\gamma,\delta) &= E\left\{\left(y^2-\delta\right)^2\right\} && (4.69)\\
&= E\left\{\left(a^2+\gamma^2b^2+2ab\gamma-\delta\right)^2\right\} && (4.70)\\
&= \overline{a^4}+\gamma^2\overline{a^2b^2}+2\gamma\overline{a^3b}-\sigma_a^2\delta \\
&\quad +\gamma^2\overline{a^2b^2}+\gamma^4\overline{b^4}+2\gamma^3\overline{ab^3}-\gamma^2\sigma_b^2\delta \\
&\quad +2\gamma\overline{a^3b}+2\gamma^3\overline{ab^3}+4\gamma^2\overline{a^2b^2}-2\gamma\overline{ab}\delta \\
&\quad -\sigma_a^2\delta-\gamma^2\sigma_b^2\delta-2\gamma\overline{ab}\delta+\delta^2. && (4.71)
\end{aligned}$$

The partial derivatives of (4.71) are

$$\frac{\partial\mathcal{J}}{\partial\gamma}=4\overline{b^4}\gamma^3+12\overline{ab^3}\gamma^2+\left(12\overline{a^2b^2}-4\sigma_b^2\delta\right)\gamma+4\overline{a^3b}-4\overline{ab}\delta \qquad (4.72)$$

and

$$\frac{\partial\mathcal{J}}{\partial\delta}=-2\sigma_a^2-2\gamma^2\sigma_b^2-4\gamma\overline{ab}+2\delta. \qquad (4.73)$$

Before setting the partials to zero, an additional simplification is necessary, it is assumed that signal a is constant modulus. This implies that a^2 is not random but is a constant and can therefore be pulled out of any expectation operation. That is,

$$\sigma_a^2\triangleq E\{a^2\}=a^2 \qquad (4.74)$$

which implies that

$$\overline{a^2b^2}=\sigma_a^2\sigma_b^2 \qquad (4.75)$$

and

$$\overline{a^3b}=\sigma_a^2\overline{ab}. \qquad (4.76)$$

Making use of (4.75) and (4.76), the partial derivative (4.72) is set to zero yielding the following equation,

$$4\overline{b^4}\gamma^3+12\overline{ab^3}\gamma^2+\left(12\sigma_a^2\sigma_b^2-4\sigma_b^2\delta\right)\gamma+4\sigma_a^2\overline{ab}-4\overline{ab}\delta=0. \qquad (4.77)$$

Setting (4.73) to zero yields the following expression for δ,

$$\delta = \sigma_a^2 + \gamma^2\sigma_b^2 + \gamma\overline{ab}. \tag{4.78}$$

Substituting (4.78) into (4.77) eliminates δ and yields the following expression in γ

$$4\left(\overline{b^4} - (\sigma_b^2)^2)\gamma^3 + 12\left(\overline{ab^3} - \overline{ab}\sigma_b^2\right)\gamma^2 + 8\left(\sigma_a^2\sigma_b^2 - \left(\overline{ab}\right)^2\right)\gamma = 0. \tag{4.79}$$

The real roots of this equation correspond to stationary points of the cost function given in (4.71). Note that $\gamma = 0$ is a root of this cubic equation. Also, if signal b is constant modulus, i.e., b^2 is deterministic and constant and satisfies

$$\sigma_b^2 \triangleq E\{b^2\} = b^2, \tag{4.80}$$

then the cubic equation (4.79) degenerates to

$$\frac{(\overline{ab})^2}{\sigma_a^2\sigma_b^2} = 1 \tag{4.81}$$

which is independent of γ.

Equation (4.81) can be equivalently stated as

$$\rho = \pm 1 \tag{4.82}$$

where ρ is the correlation coefficient between signal a and b. This implies that if a and b are perfectly correlated (positive or negative), then a stationary point exists for all values of γ. This is obvious since the sum of two perfectly correlated CM signals is again a constant modulus signal independent of the gains on the two signals.

To check for additional roots it is assumed that b is not CM and that $\gamma \neq 0$, then γ can be divided from (4.79) yielding a quadratic

$$\left(\overline{b^4} - (\sigma_b^2)^2\right)\gamma^2 + 3\left(\overline{ab^3} - \overline{ab}\sigma_b^2\right)\gamma + 2\left(\sigma_a^2\sigma_b^2 - (\overline{ab})^2\right) = 0. \tag{4.83}$$

If real roots exist, the discriminant of (4.83) must be greater than zero,

$$9\left((\overline{ab^3})^2+\left(\overline{ab}\sigma_b^2\right)^2-2\overline{ab^3}\,\overline{ab}\sigma_b^2\right)-8\left(\left(\overline{b^4}-(\sigma_b^2)^2\right)\left(\sigma_a^2\sigma_b^2-(\overline{ab})^2\right)\right)>0. \quad (4.84)$$

Combining terms yields

$$9(\overline{ab^3})^2+\left(\overline{ab}\sigma_b^2\right)^2-18\overline{ab^3}\,\overline{ab}\sigma_b^2-8\left(\overline{b^4}\sigma_b^2\sigma_a^2-\overline{b^4}(\overline{ab})^2-(\sigma_a^2\sigma_b^2)^3\right)>0 \quad (4.85)$$

which can be reduced to the normalized form

$$9\frac{(\overline{ab^3})^2}{\sigma_a^2\sigma_b^2\overline{b^4}}+\frac{(\overline{ab})^2\sigma_b^2}{\sigma_a^2\overline{b^4}}-18\frac{\overline{ab^3}\,\overline{ab}}{\sigma_a^2\overline{b^4}}-8\left(1-\frac{(\overline{ab})^2}{\sigma_a^2\sigma_b^2}-\frac{1}{k_b}\right)>0, \quad (4.86)$$

where k_b is the kurtosis of signal b. This equation is "normalized" in the sense that each statistical term is in the form of a higher order correlation coefficient. Although normalized in terms of units, their magnitudes are not necessarily bounded by 1 for general signals a and b. For the specific signal case at hand, however, the magnitude of each is bounded by 1. This is because the signal a is constant modulus and the maximum cross correlation between a and b occurs when $b = a$. (If $b = a$ then all of the higher order correlation coefficients are equal to 1.)

Numerical tests for signal models described in Section 4.5 show that the discriminant (4.86) is positive in many cases. Therefore two real, non-zero stationary points may exist in addition to the one at $\gamma = 0$. In principle, the values of the two non-zero stationary points could be found using the quadratic root formula. and then the Hessian could be evaluated at these points and their nature (local maximum, minimum or saddle point) determined as a function of the signal statistics. However, the general purpose of this analysis is not to solve for the local minimums exactly but to gain insight into the correlated signal case. It is enough to note here that other local minimums are possible at non-zero values of the multipath gain γ.

In contrast to the intractable stationary points at non-zero values of γ, the analysis of the stationary point at $\gamma = 0$ is straightforward. The Hessian matrix at $\gamma = 0$ and $\delta = \sigma_a^2$ is given by

$$\mathbf{H}(\gamma = 0, \delta = \sigma_a^2) = \begin{bmatrix} 8\sigma_a^2\sigma_b^2 & -4\overline{ab} \\ -4\overline{ab} & 2 \end{bmatrix} \tag{4.87}$$

which must be positive definite for a local minimum.

Rather than making the cumbersome eigenvalue calculation directly, $\mathbf{H}$ can be shown to have positive eigenvalues indirectly. First of all, $\mathbf{H}$ is not negative-definite since

$$\mathbf{e}^\dagger\mathbf{H}\mathbf{e} > 0 \tag{4.88}$$

where $\mathbf{e}$ is either unit vector. Second, the determinant of $\mathbf{H}$

$$\det(\mathbf{H}) = \lambda_1\lambda_2 = 16\left(\sigma_a^2\sigma_b^2 - (\overline{ab})^2\right) \tag{4.89}$$

is positive unless a and b are perfectly correlated in which case it is zero. The eigenvalues cannot both be negative because of (4.88) and cannot be opposite in sign because of (4.89). Therefore $\mathbf{H}$ is positive definite as long as the two signals are not perfectly correlated. It follows that the stationary point at $\gamma = 0$ and $\delta = \sigma_a^2$ is a local minimum.

This analysis of the correlated signal case with variable δ has shown that the desired local minimum at $(0, \sigma_a^2)$ exists and that other local minimums are possible. The nature of these other local minimums will be investigated numerically in Section 4.5. If it is assumed that δ is fixed rather than variable, then these other local minimums can be investigated analytically rather than numerically.

Fixed δ case

Investigating the correlated signal case is far simpler if it is assumed that δ is constant rather than variable. It was determined in Section 4.2.2 that a local minimum exists when $\delta = \sigma_a^2$. For the following analysis, δ is fixed to σ_a^2,

$$\delta = a^2 = \sigma_a^2. \tag{4.90}$$

(Recall that since signal a is CM, a^2 is deterministic and equal to σ_a^2.)

Under these assumptions, the cost function reduces to

$$\begin{aligned} \mathcal{J}(\gamma) &= E\left\{\left(\gamma^2 b^2 + 2ab\gamma\right)^2\right\} && (4.91) \\ &= \overline{b^4}\gamma^4 + 4\overline{ab^3}\gamma^3 + 4\sigma_a^2\sigma_b^2\gamma^2 && (4.92) \end{aligned}$$

and is now a function of only one variable, γ.

The stationary point is found when the derivative is zero,

$$\frac{\partial \mathcal{J}}{\partial \gamma} = 4\overline{b^4}\gamma^3 + 12\overline{ab^3}\gamma^2 + 8\sigma_a^2\sigma_b^2\gamma = 0. \tag{4.93}$$

Clearly $\gamma = 0$ is one solution, if it is assumed that $\gamma \neq 0$ then γ can be divided out to produce the quadratic,

$$\overline{b^4}\gamma^2 + 3\overline{ab^3}\gamma + 2\sigma_a^2\sigma_b^2 = 0. \tag{4.94}$$

For other real and distinct roots to exist the discriminant must be greater than zero,

$$9(\overline{ab^3})^2 - 8\overline{b^4}\sigma_a^2\sigma_b^2 > 0 \tag{4.95}$$

or equivalently,

$$\frac{(\overline{ab^3})^2}{\overline{b^4}\sigma_a^2\sigma_b^2} > \frac{8}{9}. \tag{4.96}$$

The left hand side of inequality (4.96) is the square of the correlation measure $\hat{\rho}$ defined as,

$$\hat{\rho} \triangleq \frac{(\overline{ab^3})}{\sigma_a \sigma_b \sqrt{\overline{b^4}}}. \tag{4.97}$$

The parameter $\hat{\rho}$ has a maximum of 1 and a minimum of -1 since a is constant modulus. It is one of many possible *higher order* correlation coefficients. Similar measures make up equation (4.86). To test the nature of the roots, the second derivative is formed,

$$\frac{\partial^2 \mathcal{J}}{\partial \gamma^2} = 12\overline{b^4}\gamma^2 + 24\overline{ab^3}\gamma + 8\sigma_a^2\sigma_b^2 \tag{4.98}$$

Clearly, $\gamma = 0$ is a local minimum.

The only point of interest with the other two stationary points, (when (4.96) is satisfied,) is whether or not they correspond to local minimums or maximums. The most obvious approach is to solve for the roots of (4.94) and then substitute them into (4.98). This is unwieldy, however, and a simpler and more intuitive approach is possible. Since (4.93) has a positive leading coefficient and three distinct roots including $\gamma = 0$, it can only take on the three cases of Figure 4.1. It is also known that the derivative of (4.93) given by (4.98) is positive at $\gamma = 0$ and thus Case I can be eliminated. Both Case II and Case III have positive derivatives at one of their non-zero roots. Therefore, one local minimum other than $\gamma = 0$ exists if (4.96) is satisfied. This is verified in the numerical analysis of Section 4.5.

To be conservative, it is assumed in this noiseless and real signal study that while the negative results will generalize to the complex case, the positive results may not. One important negative result is that spurious local minima are possible

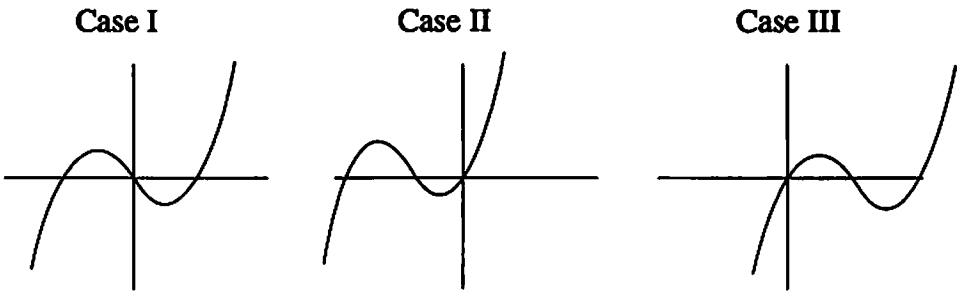

Figure 4.1: Pictorial representation of the three possible cases of a cubic with three distinct real roots, one root at zero and a positive leading coefficient.

for the correlated signal case even when an appropriate linear constraint is employed. A positive aspect of this result is that the existence of the spurious local minima depends on the degree of correlation between the signals; the correlation must be greater than a specific value for an undesired local minimum to exist. It is probable that this general relationship between correlation and local minima is found in the complex case as well.

It has been assumed up to this point that the signal environment of the narrowband array model contains no ambient white noise. This over-restrictive assumption is relaxed in the next section.

4.4 Noise Model

It is not surprising that other local minima exist for the noiseless case. After all, even the LCMP cost function (2.16) has local minima if the environment is noiseless. (The correlation matrix $\mathbf{R}_{xx}$ is singular in this case.) For power minimization, though, an infinitesimal amount of white noise makes $\mathbf{R}_{xx}$ non-singular and guarantees a unique solution (2.17) for the weight vector. The dependence of

the the LCCM cost function local minima on noise is unknown. It will be explored in this section.

4.4.1 Derivation

In order to study the effects of noise on the LCCM cost expression it is necessary to model the relationship between the gain on the interfering signal and the resultant increase or decrease in noise output power. The model is derived by assuming that the gain on the interfering signal b is fixed to γ and then solving for the minimum noise output power that is possible given that gain. This assumes that for any given gain γ the LCCM cost expression is minimized when the output noise component is minimized. This assumption may not be valid for the fixed δ case, however.

For this study, the received narrowband array vector (N elements) is modeled as

$$\mathbf{x}(n) = a(n)\mathbf{d}(\theta_a) + b(n)\mathbf{d}(\theta_b) + \mathbf{g}(n) \tag{4.99}$$

where $a(n)$ and $b(n)$ are the SOI and interfering signals respectively and the vector $\mathbf{g}(n)$ represents the white Gaussian noise received at each array sensor.

The approach that is used to solve for the minimum noise power is Linearly-Constrained Power Minimization (Section 2.2.5.) Since both the gain on the SOI and the gain on the interferer are constrained, the constraint matrix for this analysis is given by

$$\mathbf{C} \triangleq \left[\begin{array}{cc} \mathbf{d}(\theta_a) & \mathbf{d}(\theta_b) \end{array} \right] \tag{4.100}$$

and the overall constraint equation is

$$\mathbf{C}^{\dagger}\mathbf{w} \triangleq \mathbf{f} \tag{4.101}$$

$$= \begin{pmatrix} 1 \\ \gamma^* \end{pmatrix} \tag{4.102}$$

where $\mathbf{f}$ is the response vector. This equation forces the weight vector $\mathbf{w}$ to have a gain of 1 on the SOI and a gain of γ on the interfering signal.

The next step is to minimize the total output power. Minimizing the total output power is equivalent to minimizing the noise output power because the gains on the two signals are constrained. The weight vector $\mathbf{w}_{opt}$ that minimizes the total output power satisfies

$$\min_{\mathbf{w}} E\left\{|y|^2\right\} \qquad \text{subject to} \qquad \mathbf{C}^{\dagger}\mathbf{w} = \mathbf{f} \tag{4.103}$$

or equivalently

$$\min_{\mathbf{w}} \mathbf{w}^{\dagger}\mathbf{R}_{xx}\mathbf{w} \qquad \text{subject to} \qquad \mathbf{C}^{\dagger}\mathbf{w} = \mathbf{f}. \tag{4.104}$$

where

$$\begin{aligned} \mathbf{R}_{xx} &= \sigma_a^2 \mathbf{d}(\theta_a)\mathbf{d}^{\dagger}(\theta_a) + \sigma_b^2 \mathbf{d}(\theta_b)\mathbf{d}^{\dagger}(\theta_b) + \overline{ab^*}\mathbf{d}(\theta_a)\mathbf{d}^{\dagger}(\theta_b) \\ &\quad + \overline{a^*b}\mathbf{d}(\theta_b)\mathbf{d}^{\dagger}(\theta_a) + \sigma_g^2\mathbf{I} \end{aligned} \tag{4.105}$$

The solution to (4.104) is given by

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1}\mathbf{C}(\mathbf{C}^{\dagger}\mathbf{R}_{xx}^{-1}\mathbf{C})^{-1}\mathbf{f}. \tag{4.106}$$

The minimum power weight vector (4.106) can be reduced to a much simpler form by making use of a special decomposition [10],

$$\mathbf{w}_{opt} = \mathbf{C}^{\dagger}(\mathbf{C}^{\dagger}\mathbf{C})^{-1}\mathbf{f} - \mathbf{W}_s(\mathbf{W}_s^{\dagger}\mathbf{R}_{xx}\mathbf{W}_s)^{-1}\mathbf{W}_s^{\dagger}\mathbf{R}_{xx}\mathbf{C}(\mathbf{C}^{\dagger}\mathbf{C})^{-1}\mathbf{f} \tag{4.107}$$

where $\mathbf{W}_s$ is in general any $N \times (N - M)$ full rank matrix that satisfies

$$\mathbf{C}^\dagger \mathbf{W}_s = \mathbf{0}, \tag{4.108}$$

in this case $M = 2$.

Note that the second term in (4.107) includes the product $\mathbf{W}_s^\dagger \mathbf{R}_{xx} \mathbf{C}$. By construction, (equations (4.100) and (4.108)), the matrix $\mathbf{W}_s$ is orthogonal to $\mathbf{d}(\theta_a)$ and $\mathbf{d}(\theta_b)$. Therefore, using (4.105),

$$\mathbf{W}_s^\dagger \mathbf{R}_{xx} \mathbf{C} = \sigma_g^2 \mathbf{W}_s^\dagger \mathbf{I} \mathbf{C} = \mathbf{0} \tag{4.109}$$

and (4.106) becomes

$$\mathbf{w}_{opt} = \mathbf{C}(\mathbf{C}^\dagger \mathbf{C})^{-1} \mathbf{f}. \tag{4.110}$$

The next step is to determine the white noise output power that corresponds to the weight vector in (4.110). This power is given by

$$\sigma_g^2 \mathbf{w}_{opt}^\dagger \mathbf{I} \mathbf{w}_{opt} = \sigma_g^2 (\mathbf{C}(\mathbf{C}^\dagger \mathbf{C})^{-1} \mathbf{f})^\dagger \mathbf{C}(\mathbf{C}^\dagger \mathbf{C})^{-1} \mathbf{f} \tag{4.111}$$

$$= \sigma_g^2 \mathbf{f}^\dagger (\mathbf{C}^\dagger \mathbf{C})^{-1} \mathbf{f} \tag{4.112}$$

The matrix $(\mathbf{C}^\dagger \mathbf{C})^{-1}$ is necessary to compute the noise power. Let

$$\mathbf{C}^\dagger \mathbf{C} \triangleq \begin{bmatrix} 1 & \nu \\ \nu^* & 1 \end{bmatrix} \tag{4.113}$$

where

$$\nu = \mathbf{d}^\dagger(\theta_a) \mathbf{d}(\theta_b) \tag{4.114}$$

Note that

$$|\nu| < 1 \tag{4.115}$$

since the direction vectors are normalized.

The inverse of (4.113) is given by

$$(\mathbf{C}^\dagger\mathbf{C})^{-1} = \frac{1}{1-|\nu|^2}\begin{bmatrix} 1 & -\nu \\ -\nu^* & 1 \end{bmatrix} \tag{4.116}$$

and the resultant output white noise power is

$$\sigma_g^2\mathbf{f}^\dagger(\mathbf{C}^\dagger\mathbf{C})^{-1}\mathbf{f} = \sigma_g^2\frac{1-(\nu\gamma^*+\nu^*\gamma)+|\gamma|^2}{1-|\nu|^2}. \tag{4.117}$$

This expression correctly reflects the fact that, for certain ν, a non-zero gain on the interference may result in lower noise output power than if the interference is completely nulled. In fact, in some cases the *total* output power may be reduced by a non-zero γ depending on the strength of the interferer. This is the reason why a true null ($\gamma = 0$) on an interferer is not always the best solution when noise is present. A tradeoff exists between the null depth and total output noise power and this is accounted for in (4.117.)

4.4.2 Correlated Signals

In this analysis, ν defined in (4.114) is assumed to be zero which implies that the direction vectors $\mathbf{d}(\theta_a)$ and $\mathbf{d}(\theta_b)$ are orthogonal. This small compromise in generality greatly simplifies the analysis and allows easier interpretation of the results.

Using the noise model of the previous section, the real array output can be written as

$$y = a + \gamma b + \sqrt{1+\gamma^2}g \tag{4.118}$$

where the real amplitude gain on the noise results in a power gain of $(1+\gamma^2)$ on the noise term in accordance with (4.117.)

Squaring the output yields

$$y^2 = a^2 + \gamma^2 b^2 + 2ab\gamma + (1+\gamma^2)g^2 + 2bg\gamma\sqrt{1+\gamma^2} + 2ag\sqrt{1+\gamma^2} \qquad (4.119)$$

which is then substituted into the LCCM cost function

$$\mathcal{J}(\gamma, \delta) \triangleq E\{(y^2 - \delta)^2\}. \qquad (4.120)$$

Sparing details, the expanded LCCM cost function is

$$\begin{aligned} \mathcal{J}(\gamma,\delta) \;=\; & (\overline{b^4} + 6\overline{b^2}\sigma_g^2 + 3\sigma_g^4)\gamma^4 + (4\overline{ab^3} + 12\overline{ab}\sigma_g^2)\gamma^3 \\ & + (6\overline{a^2b^2} + 6\sigma_a^2\sigma_g^2 + 6\sigma_b^2\sigma_g^2 + 6\sigma_g^4 - 2\sigma_b^2\delta - 2\sigma_g^2\delta)\gamma^2 \\ & + (12\overline{ab}\sigma_g^2 + 4\overline{a^3b} - 4\overline{ab}\delta)\gamma \\ & + (6\sigma_a^2\sigma_g^2 + 3\sigma_g^4 + \overline{a^4} - 2\overline{a^2}\delta - 2\sigma_g^2\delta + \delta^2) \end{aligned} \qquad (4.121)$$

where the real white Gaussian noise g has a fourth order moment given by

$$\overline{g^4} = 3(\sigma_g^2)^2 = 3\sigma_g^4. \qquad (4.122)$$

The partial with respect to γ is set to zero yielding

$$\frac{\partial \mathcal{J}}{\partial \gamma} = 4(\cdot)\gamma^3 + 3(\cdot)\gamma^2 + 2(\cdot)\gamma + (\cdot) = 0 \qquad (4.123)$$

where each shorthand notation $(\cdot)$ corresponds to the appropriate coefficient of (4.121). Setting the partial with respect to δ equal to zero

$$\frac{\partial \mathcal{J}}{\partial \delta} = 2\delta - 2\sigma_b^2\gamma^2 - 2\sigma_g^2\gamma^2 - 4\overline{ab}\gamma - 2\sigma_a^2 - 2\sigma_g^2 = 0 \qquad (4.124)$$

leads to the following expression for δ,

$$\delta = \sigma_a^2 + \sigma_b^2\gamma^2 + (1+\gamma^2)\sigma_g^2 + 2\overline{ab}\gamma. \qquad (4.125)$$

Substituting (4.125) into (4.123) and collecting terms in γ yields

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \gamma} &= 4(\overline{b^4} - \sigma_b^4 + 4\sigma_b^2\sigma_g^2 + 2\sigma_g^4)\gamma^3 \\ &\quad + 12(2\sigma_g^2\overline{ab} + \overline{ab^3} - \overline{ab}\sigma_b^2)\gamma^2 \\ &\quad + 4(3\overline{a^2b^2} - \sigma_a^2\sigma_b^2 + 2\sigma_a^2\sigma_g^2 + 2\sigma_b^2\sigma_g^2 + 2\sigma_g^2 - 2(\overline{ab})^2)\gamma \\ &\quad + 4(2\overline{ab}\sigma_g^2 + \overline{a^3b} - \overline{ab}\sigma_a^2) = 0 \end{aligned} \tag{4.126}$$

It is interesting to note that, unlike the previous noise-absent cases, $\gamma = 0$ is not a stationary point in general. That is, a weight vector that places a null on the interfering signal is not a local minimum. This is true in spite of the fact that, with reference to (4.117), if $\nu = 0$ a non-zero γ can only *increase* noise power at the output. The tradeoff between noise power and interferer power does not exist when $\nu = 0$. Clearly, there is some other cost-reducing mechanism involved here that has not been taken into account. Rather than continuing this study analytically, however, the cost function for this case (4.121) is investigated numerically in Section 4.5.

4.4.3 Independent Signals with Background Noise

Although the general case of correlated signals in noise is analytically intractable, if the signals are independent the analysis is straightforward. Assuming a and b are independent, (4.126) can be rewritten as

$$\frac{\partial \mathcal{J}}{\partial \gamma} = 4(\overline{b^4} - \sigma_b^4 + 4\sigma_b^2\sigma_g^2 + 2\sigma_g^4)\gamma^3 + 8(\sigma_a^2\sigma_b^2 + \sigma_a^2\sigma_g^2 + \sigma_b^2\sigma_g^2 + \sigma_g^4)\gamma = 0 \tag{4.127}$$

which has a root at $\gamma = 0$.

Assuming $\gamma \neq 0$,

$$\gamma^2 = \frac{-2(\sigma_a^2\sigma_b^2 + \sigma_a^2\sigma_g^2 + \sigma_b^2\sigma_g^2 + \sigma_g^4)}{\overline{b^4} - \sigma_b^4 + 4\sigma_b^2\sigma_g^2 + 2\sigma_g^4} \tag{4.128}$$

$$= \frac{-2(\sigma_a^2\sigma_b^2 + \sigma_a^2\sigma_g^2 + \sigma_b^2\sigma_g^2 + \sigma_g^4)}{\sigma_b^4(k_b - 1) + 4\sigma_b^2\sigma_g^2 + 2\sigma_g^4}. \tag{4.129}$$

The right hand side of (4.129) is always negative since $k_b \geq 1$. Since γ must be real it follows that the only stationary point of $\mathcal{J}$ is when $\gamma = 0$ and $\delta = \sigma_a^2 + \sigma_g^2$.

Evaluation of the second partials that form the Hessian at $\gamma = 0$ and $\delta = \sigma_a^2 + \sigma_g^2$ yields

$$\frac{\partial^2 \mathcal{J}}{\partial \gamma^2} = 8(\sigma_a^2\sigma_b^2 + \sigma_a^2\sigma_g^2 + \sigma_b^2\sigma_g^2 + \sigma_g^4) \tag{4.130}$$

$$\frac{\partial^2 \mathcal{J}}{\partial \gamma \partial \delta} = \frac{\partial^2 \mathcal{J}}{\partial \delta \partial \gamma} = 0 \tag{4.131}$$

$$\frac{\partial^2 \mathcal{J}}{\partial \delta^2} = 2 \tag{4.132}$$

Clearly, the Hessian is positive definite and therefore the only local minimum of $\mathcal{J}$ when a and b are independent is if $\gamma = 0$ and $\delta = \sigma_a^2 + \sigma_g^2$, the sum of the SOI and noise powers.

Fixed δ Case

An even simpler case to examine is if in addition to a and b being independent, the parameter δ is fixed rather than variable. If δ is fixed to its optimum value

$$\delta = \sigma_a^2 + \sigma_g^2 \tag{4.133}$$

then the derivative of $\mathcal{J}$ becomes

$$\frac{\partial \mathcal{J}}{\partial \gamma} = 4(\overline{b^4} + 6\sigma_b^2\sigma_g^2 + 3\sigma_g^4)\gamma^3 + 8(\sigma_a^2\sigma_b^2 + \sigma_a^2\sigma_g^2 + \sigma_b^2\sigma_g^2 + \sigma_g^4)\gamma. \tag{4.134}$$

Clearly, $\gamma = 0$ is the only real valued stationary point and hence the only local minimum for this case as well.

A general conclusion that can be drawn from the analysis up to this point is that the independent signal case does not present a problem for the LCCM

cost expression. In other words, for the case when a and b are independent only one local minimum exists: the desired local minimum corresponding to a null on the interfering signal b. So far, this is true whether or not the SOI is constant modulus, whether or not background noise is present and whether or not δ is variable.

The analytical results for correlated signals are not as clear. For one thing, existence of multiple local minimums has only been shown for the fixed δ, noise-absent case. All other correlated signal cases are so complex that the difficulty of analysis far outweighs the significance of any potential result. Rather than continuing this line of analysis with contrived cases, the correlated signal case of Section 4.3 is reexamined in the next section from a numerical point of view.

4.5 Numerical Models

In this section, the LCCM cost function expressions for real correlated signals are investigated numerically. The principle objective is to determine the existence and general locations of local minima for equation (4.121.) Both the noise-absent and noise-present cases are examined. (Note that for the noise-absent case, equation (4.121) reduces to equation (4.71.) The approach taken is to construct signal models for the correlated signals a and b, calculate the appropriate statistics, and then evaluate the cost expression as a function of the gain on the correlated interferer γ and the modulus factor δ. This process is repeated for two different signal models and a variety of cross-correlation values.

Although standard non-linear optimization techniques can be used to find the local minima, for the purposes of this analysis contour plots of the cost function

surface offer more insight. This is because the precise location of the the minima is less important than the general relationship between the existence of the local minima and the correlation of the two real signals. It is possible that this relationship may generalize to the more realistic complex signal case while the exact locations of the local minimums will certainly not.

4.5.1 Correlated Signal Model # 1: Multipath

The first signal model emulates the sampled output of a BPSK matched filter. The contribution at the output of the matched filter due to the SOI is given by

$$a(n) = \sigma_a m(n) \tag{4.135}$$

where $m(n)$ is an uncorrelated bipolar message sequence

Obviously, the signal $a(n)$ is constant modulus and has power σ_a^2. The contribution at the output of the matched filter due to the multipath signal is modeled as

$$b(n) = \xi\left((1-\beta)a(n) + \beta a(n-1)\right) \tag{4.136}$$

where β is the fractional baud delay and ξ combines the attenuation and phase on the correlated interferer. The phase takes on only two values for this real signal case: 0° and 180°.

The multipath time delay is represented by β the fractional baud delay. For example, if $\beta = 1/3$ then the matched filter window includes 2/3 of current symbol and 1/3 of the previous symbol (with reference to the SOI.) This is shown pictorially in Figure 4.2 for a baseband BPSK signal.

In Section 4.3, it was determined that the higher order correlation coefficient $\hat{\rho}$ is an important parameter that determines the existence of sub-optimal local

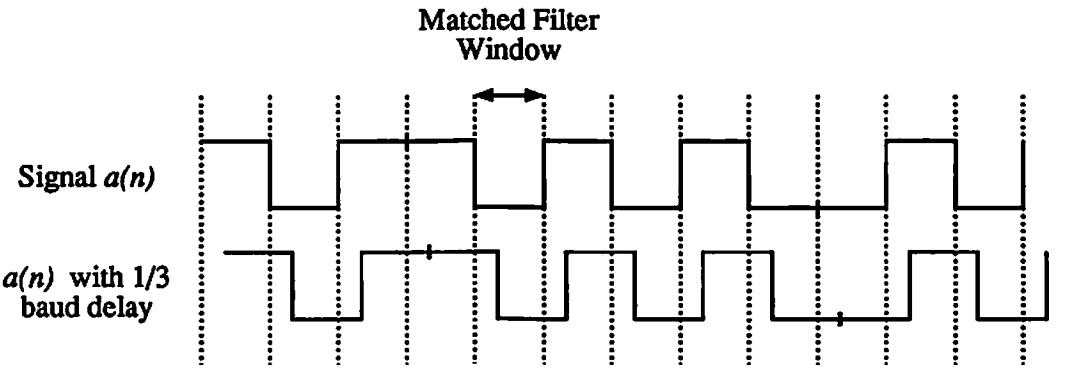

Figure 4.2: Baseband representation of a multipath signal delayed by 1/3 of a symbol interval with respect to the SOI.

minima for the fixed δ case. The parameter $\hat{\rho}$ and the traditional correlation coefficient ρ are plotted in Figure 4.3 as a function of fractional baud delay. Note that unlike ρ, the higher order correlation coefficient $\hat{\rho}$ is not a monotonically decreasing function of multipath delay. Further, $\hat{\rho}$ reaches its maximum of unity at two values of fractional baud delay: 0 and 1/2. It turns out that with a delay of 1/2, the multipath signal has a kurtosis equal to 1 just as it does with no delay. The fact that $\beta = 0$ yields a signal b with unity kurtosis is no surprise because for $\beta = 0$ the signal b is constant modulus since it is equal to signal a. But when $\beta = 1/2$, signal the multipath signal b is not constant modulus and yet the kurtosis is again unity. Although it is true that all CM signals have unity kurtosis, evidently not all unity kurtosis signals are constant modulus. (Another signal model that does not have the non-monotonicity phenomenon in $\hat{\rho}$ is used in Section 4.5.2.)

Figures 4.4a through 4.7b plot the cost function given in equation (4.121) for the parameters which are listed in Table 4.1. For each value of β, the cost

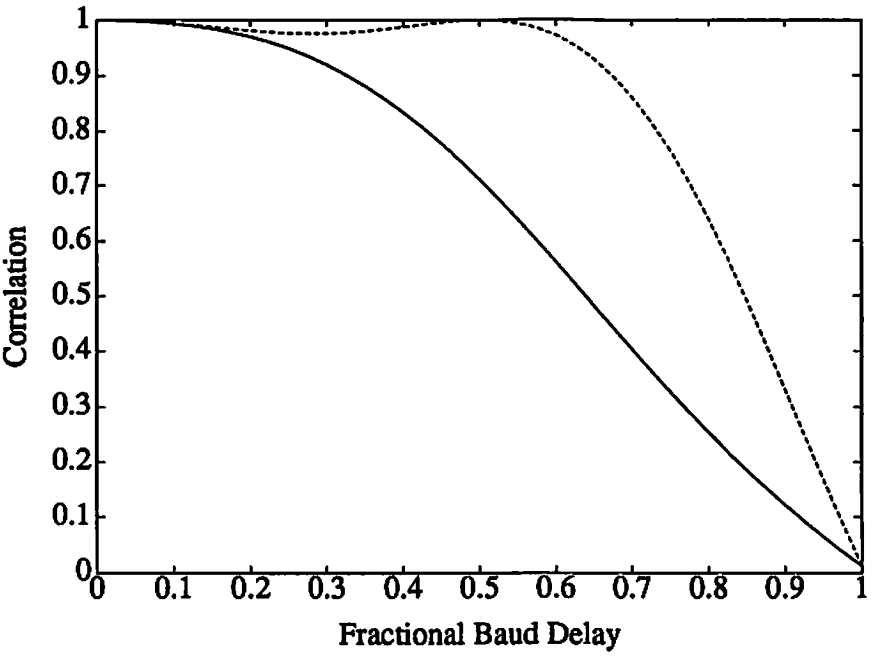

Figure 4.3: Correlation between signals $a(n)$ and $b(n)$ as function of fractional baud delay. ρ: solid line. $\hat{\rho}$: dashed line.

function is plotted with and without noise, i.e. $\sigma_g^2 = 1$ and $\sigma_g^2 = 0$ respectively. This emphasizes the significant effect that additive noise has on the cost function.

	σ_a^2 [dB]	ξ	β	ρ	$\hat{\rho}$	σ_g^2
Figure 4.4a	15	$1/\sqrt{2}$	0.81	0.24	0.61	0
Figure 4.4b	15	$1/\sqrt{2}$	0.81	0.24	0.61	1
Figure 4.5a	15	$1/\sqrt{2}$	0.65	0.46	0.92	0
Figure 4.5b	15	$1/\sqrt{2}$	0.65	0.46	0.92	1
Figure 4.6a	15	$1/\sqrt{2}$	0.60	0.56	0.98	0
Figure 4.6b	15	$1/\sqrt{2}$	0.60	0.56	0.98	1
Figure 4.7a	15	$1/\sqrt{2}$	0.28	0.93	0.98	0
Figure 4.7b	15	$1/\sqrt{2}$	0.28	0.93	0.98	1

Table 4.1: Parameter Set for Signal Model #1

Note that in the plots without noise, Figures 4.4a, 4.5a, 4.6a and 4.7a, there is always a local minimum at the point where $\gamma = 0$ and $\delta = 15$ dB. This verifies the analysis of Section 4.3. In addition, these same plots all contain a second local minimum for some negative value of the multipath gain parameter γ. This is true even when the two signals have a relatively small degree of correlation as is the case in Figure 4.4a where $\rho = 0.24$. Therefore, for this signal model and under noiseless conditions a linear constraint on the SOI does not guarantee that the cost function will have only one local minimum.

The effects of noise are shown in Figures 4.4b, 4.5b, 4.6b and 4.7b. For the plots corresponding to the three largest values of β, which are the most uncorrelated, the undesired local minimum disappears leaving only the desired minimum at the point where γ is approximately zero. In contrast, for the smallest delay (highest correlation) case plotted in Figure 4.7b the desired minimum at $\gamma \approx 0$ disappears leaving only the undesired minimum. This is clearly sub-optimal and

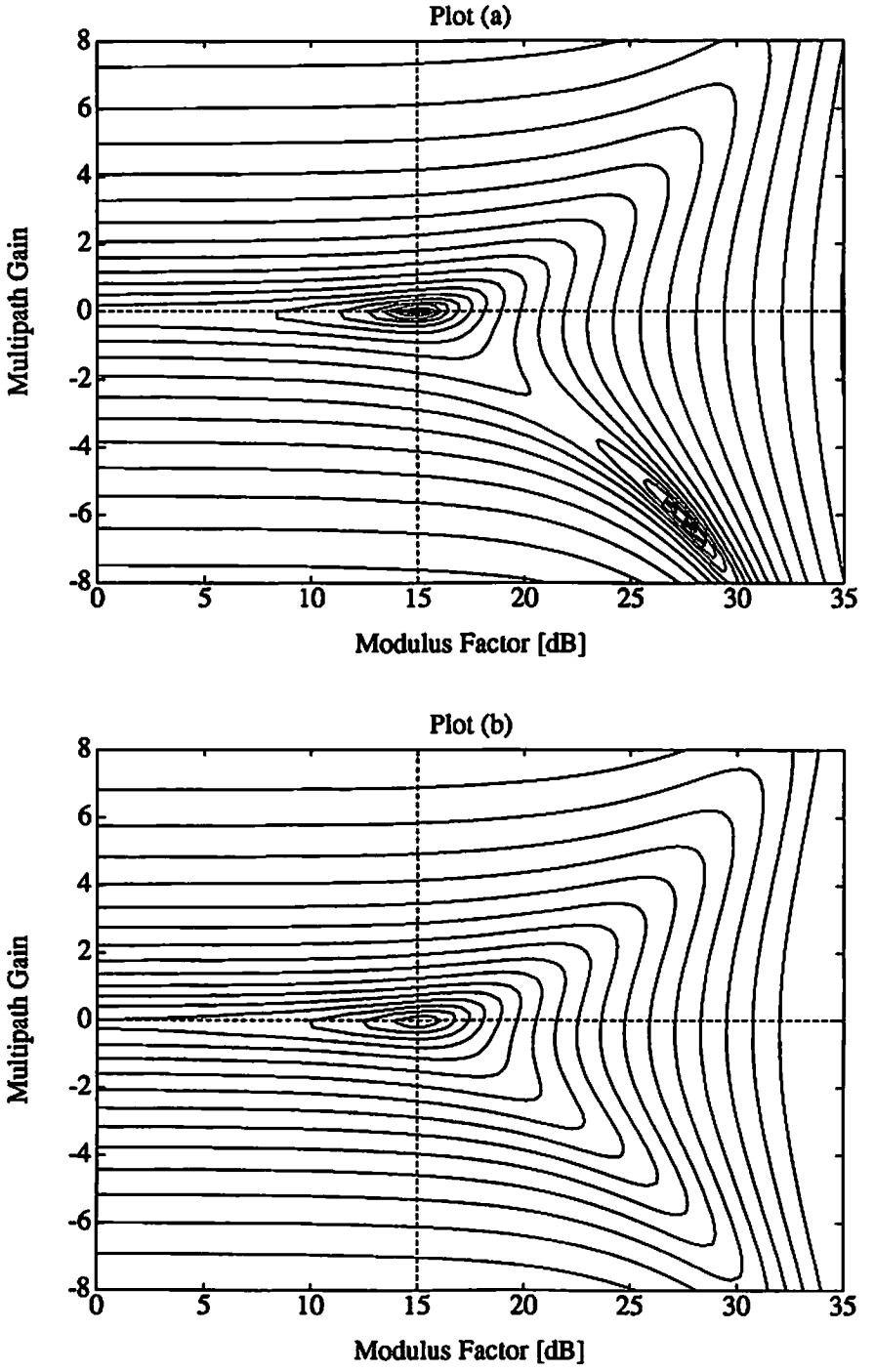

Figure 4.4: Contour plots of LCCM cost as a function of δ and the gain on the multipath signal. Signal model #1 with ρ =0.24 and $\hat{\rho}$=0.61. Plot (a): Noise absent. Plot (b): Noise present.

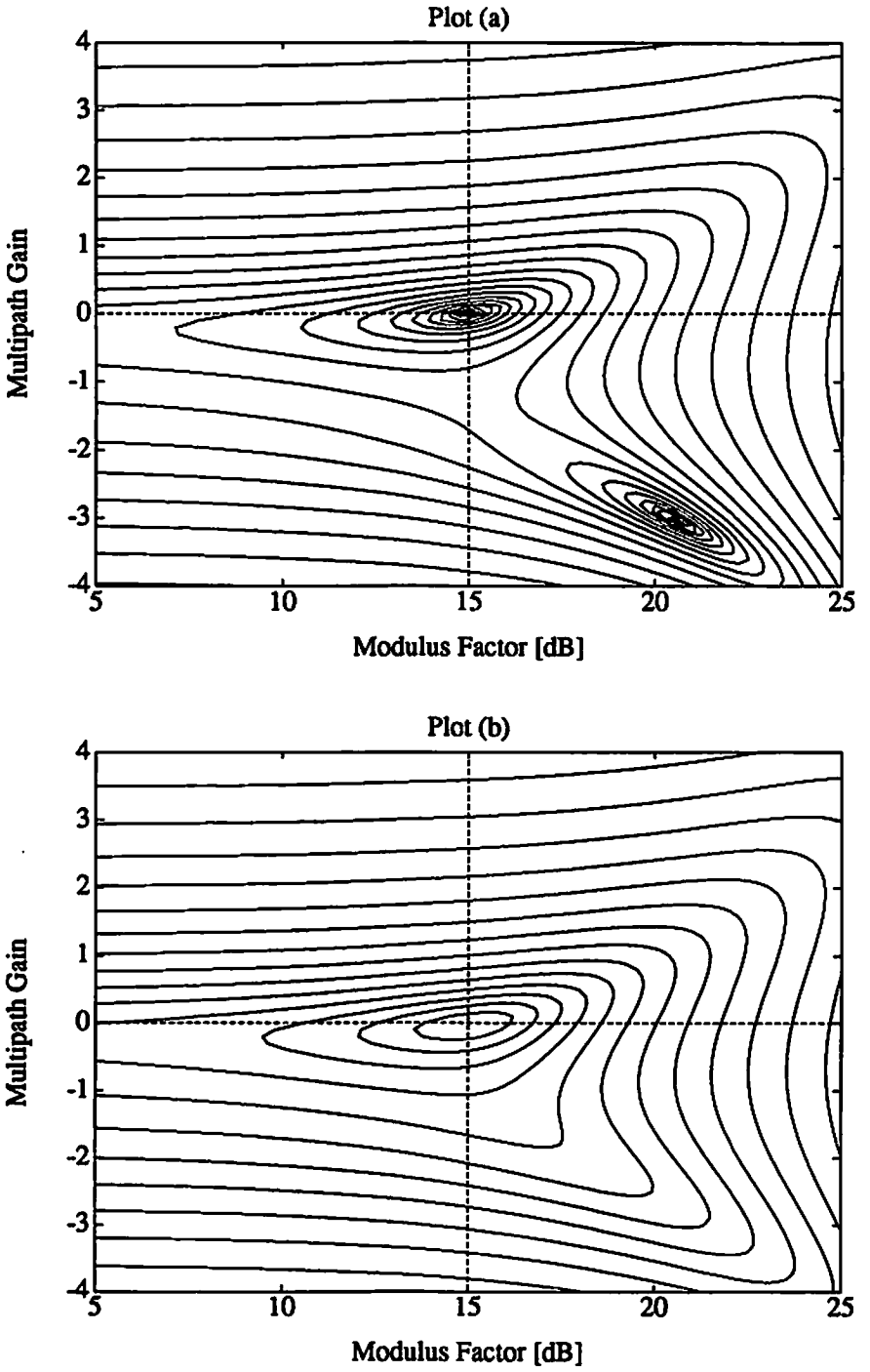

Figure 4.5: Contour plots of LCCM cost as a function of δ and the gain on the multipath signal. Signal model #1 with ρ =0.46 and $\hat{\rho}$=0.92. Plot (a): Noise absent. Plot (b): Noise present.

demonstrates that linear constraints may not solve all of the problems that are associated with the constant modulus cost function. It is important to note, however, that this is a special case and simulations of the LCCM adaptive algorithm (Chapter 6) for conditions that are approximately equivalent to this case have not produced an analogous sub-optimal result.

Figures 4.6a and 4.7a show the noise-absent cost function for identical values of $\hat{\rho}$ but different values of ρ. Although their $\hat{\rho}$ values are the same, the resultant surfaces are quite different. Figures 4.6b and 4.7b show the corresponding noise-present plots. Again, the surfaces are very different. Clearly, the parameter $\hat{\rho}$ is not a reliable indicator of local minima position for the case when δ is a free variable. Analysis in Section 4.3 did show that it is an important parameter for the fixed δ case, however.

Fixed δ

The fixed δ case is another important comparison between the figures. When δ is fixed the cost expression (4.121) becomes a function of the single gain variable γ. This is represented on the plots by vertical slices or cross-sections through the cost function surface. A dashed line is drawn to indicate the cross-section that corresponds to the optimal value of δ for the noise-absent case. Recall from Section 4.3 that if

$$\frac{(\overline{ab^3})^2}{\overline{b^4}\sigma_a^2\sigma_b^2} < \frac{8}{9}. \tag{4.137}$$

is satisfied, then $\gamma = 0$ is the only local minimum. Figures 4.4a and 4.5a plot the cost function of signals that satisfy (4.137). For $\delta = 15$ dB, only the minimum at $\gamma = 0$ exists and thus the inequality in (4.137) is verified. Table 4.1 shows that $\hat{\rho}$ for Figures 4.6a and 4.7a does not satisfy (4.137.) Accordingly, close observation

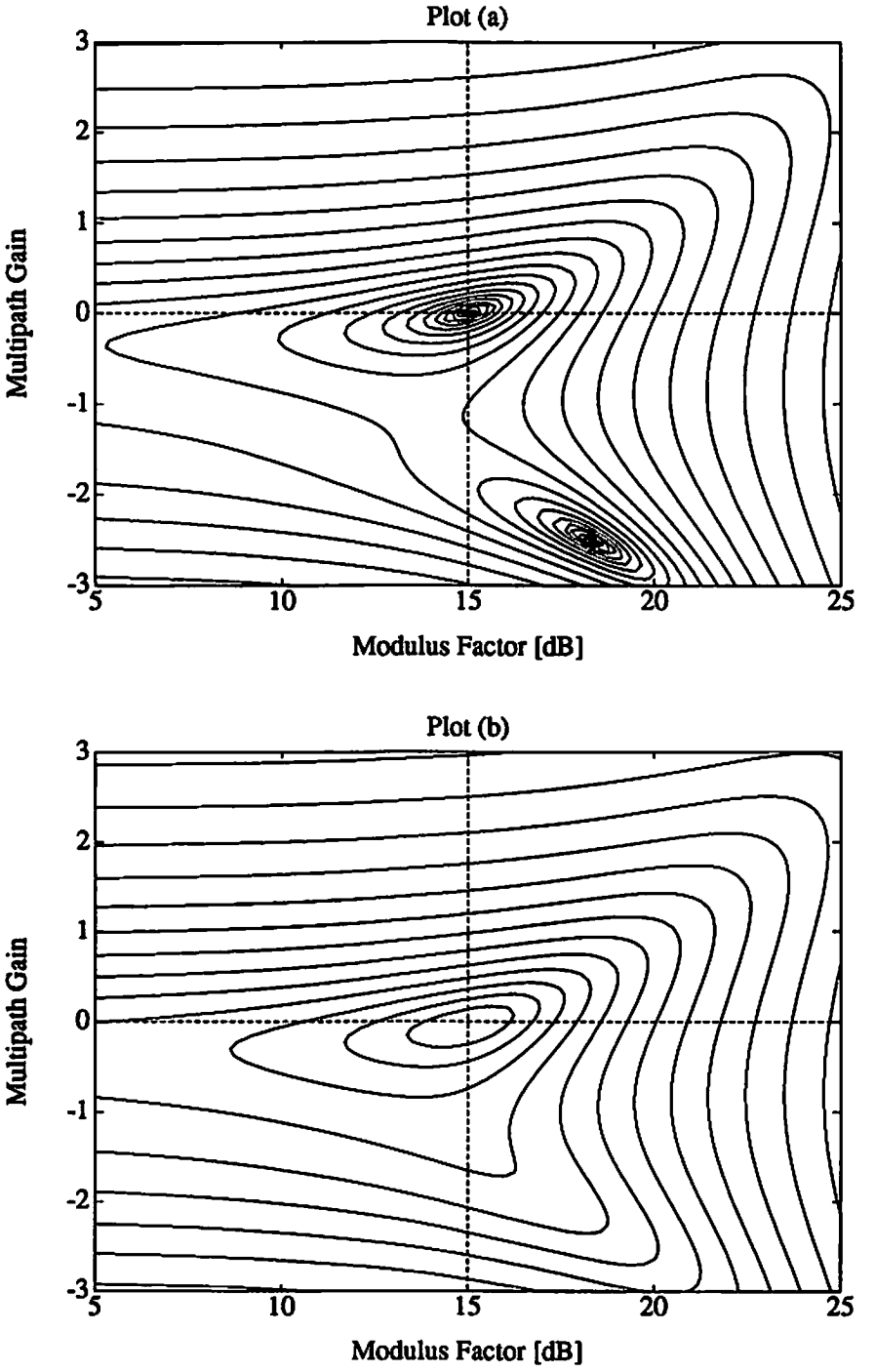

Figure 4.6: Contour plots of LCCM cost as a function of δ and the gain on the multipath signal. Signal model #1 with ρ =0.56 and $\hat{\rho}$=0.98. Plot (a): Noise absent. Plot (b): Noise present.

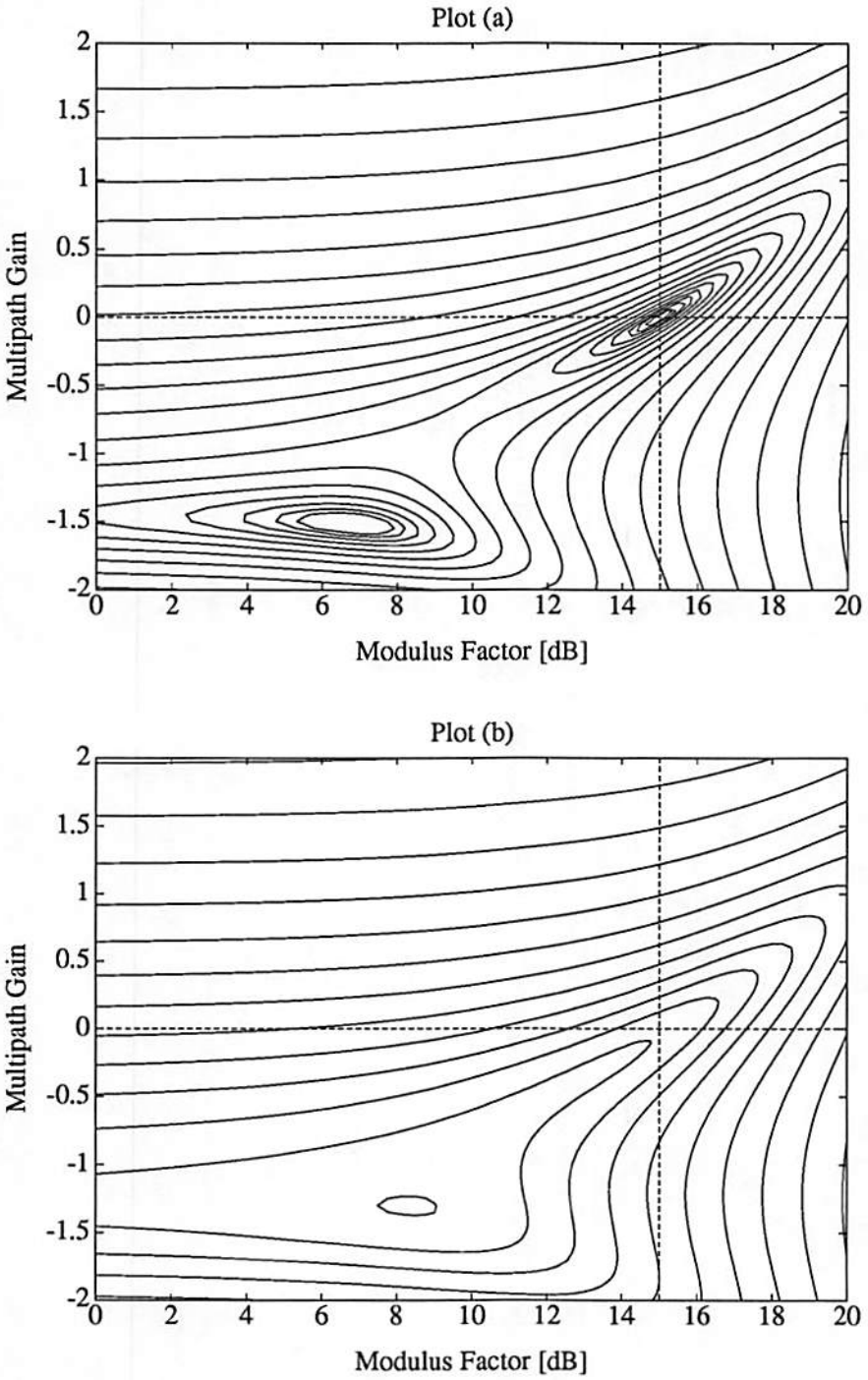

Figure 4.7: Contour plots of LCCM cost as a function of δ and the gain on the multipath signal. Signal model #1 with ρ =0.93 and $\hat{\rho}$=0.98. Plot (a): Noise absent. Plot (b): Noise present.

of Figures 4.6a and 4.7a reveals that indeed, for $\delta = 15$ dB, there are two local minimums. Again this simply verifies the analytical results of Section 4.3.

The noise-present cost expression (4.121) was not examined analytically for the fixed δ case due to its complexity. However, the effects of noise can be easily observed from the plots. Figures 4.4b and 4.5b have only one minimum along the $\delta = 15$ dB slice and retain it when noise is added to the signal model. Although Figure 4.6a has two minimums along the 15 dB slice, Figure 4.6b evidently has only one. In this case at least, the noise has a positive effect on the cost function: undesired local minima along a fixed δ slice are eliminated. This also implies that the inequality (4.137) is not necessarily valid for the noise-present case. That is, even though $\hat{\rho} > 2\sqrt{2}/3$ and two local minima are predicted for the noise-absent case, only one is observed for the noise-present case. Close study of Figure 4.7b shows that two local minima exist along the $\delta = 15$ dB slice. The first one is at $\gamma \approx 0$ point and the second is off the plot but it is implied by the local maximum at $\gamma \approx -1$. This is again a positive result because no local minimum exists at $\gamma = 0$ when δ is a free variable. For this case, a fixed δ is preferred over a variable δ. If δ is variable the gain that corresponds to the one local minimum is $\gamma \approx -1.25$. This yields the minimum cost but it is undesirable nevertheless. Fixing $\delta = 15$ dB forces γ to be on a surface cross section that, although it has two local minimums, it at least has one at $\gamma \approx 0$.

These plots show the relationship between correlation and the cost function local minima for one simple signal model. An important question to ask is how dependent these results are on the particular signal model used. In the next section, a second model is developed to address this very question.

4.5.2 Correlated Signal Model # 2: Signal Copy & Noise

Like the signal model of Section 4.5.1, this model emulates the output of a BPSK matched filter. The SOI signal $a(n)$ is the same as (4.135) but the multipath is modeled differently,

$$b(n) = a(n) + \gamma_0 g_0(n) \tag{4.138}$$

where γ_0 is the gain on a white Gaussian noise signal $g_0(n)$. The white noise added to the signal serves to decorrelate signal $b(n)$ from $a(n)$. Although the signal $b(n)$ is not a conventional multipath signal, it will be described as such in the following, i.e. *multipath gain* refers to the gain on $b(n)$.

Figure 4.8 shows the correlation coefficients ρ and $\hat{\rho}$ as functions of the noise gain parameter γ_0. Note that both ρ and $\hat{\rho}$ are monotonically decreasing functions of the noise gain γ_0. This contrasts with Figure 4.3 from the previous signal model of Section 4.5.2 in which $\hat{\rho}$ was not monotonically decreasing.

Figures 4.9a through 4.10b plot the cost function given in equation (4.121) for the parameters which are listed in Table 4.2. For each value of γ_0, the cost function is plotted with and without background noise, i.e. $\sigma_g^2 = 1$ and $\sigma_g^2 = 0$ respectively. It is important to distinguish between the *background* noise signal $g(n)$ and the *decorrelation* noise signal $g_0(n)$. The background noise signal $g(n)$ is as described in Section 4.4 while the decorrelation noise signal $g_0(n)$ is part of the signal $b(n)$ (4.138). Both are zero-mean, white Gaussian noise random processes but their effects are much different.

As was the case with the previous signal model, the noise-absent plots in Figures 4.9a and 4.10a both have local minimums at ($\gamma = 0, \delta = 15$ dB.) Note that the term "noise-absent" refers to the background noise not the decorrelation

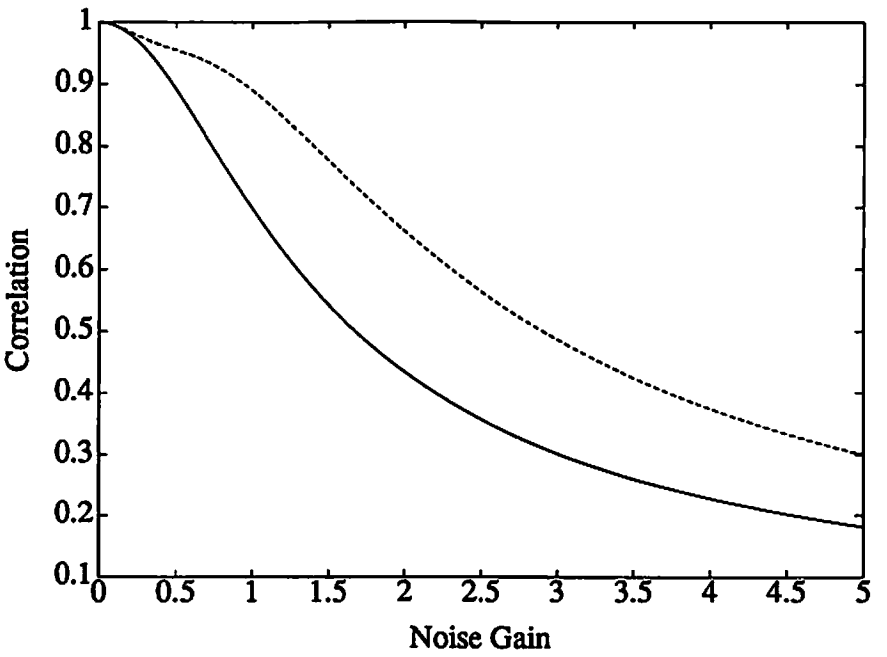

Figure 4.8: Correlation between signals $a(n)$ and $b(n)$ as function of noise gain in Signal Model # 2. ρ: solid line. $\hat{\rho}$: dashed line.

	σ_a^2 [dB]	ξ	γ_0	ρ	$\hat{\rho}$	σ_g^2
Figure 4.9a	15	$1/\sqrt{2}$	0.85	0.76	0.92	0
Figure 4.9b	15	$1/\sqrt{2}$	0.85	0.76	0.92	1
Figure 4.10a	15	$1/\sqrt{2}$	0.28	0.96	0.98	0
Figure 4.10b	15	$1/\sqrt{2}$	0.28	0.96	0.98	1

Table 4.2: Parameter Set for Signal Model #2

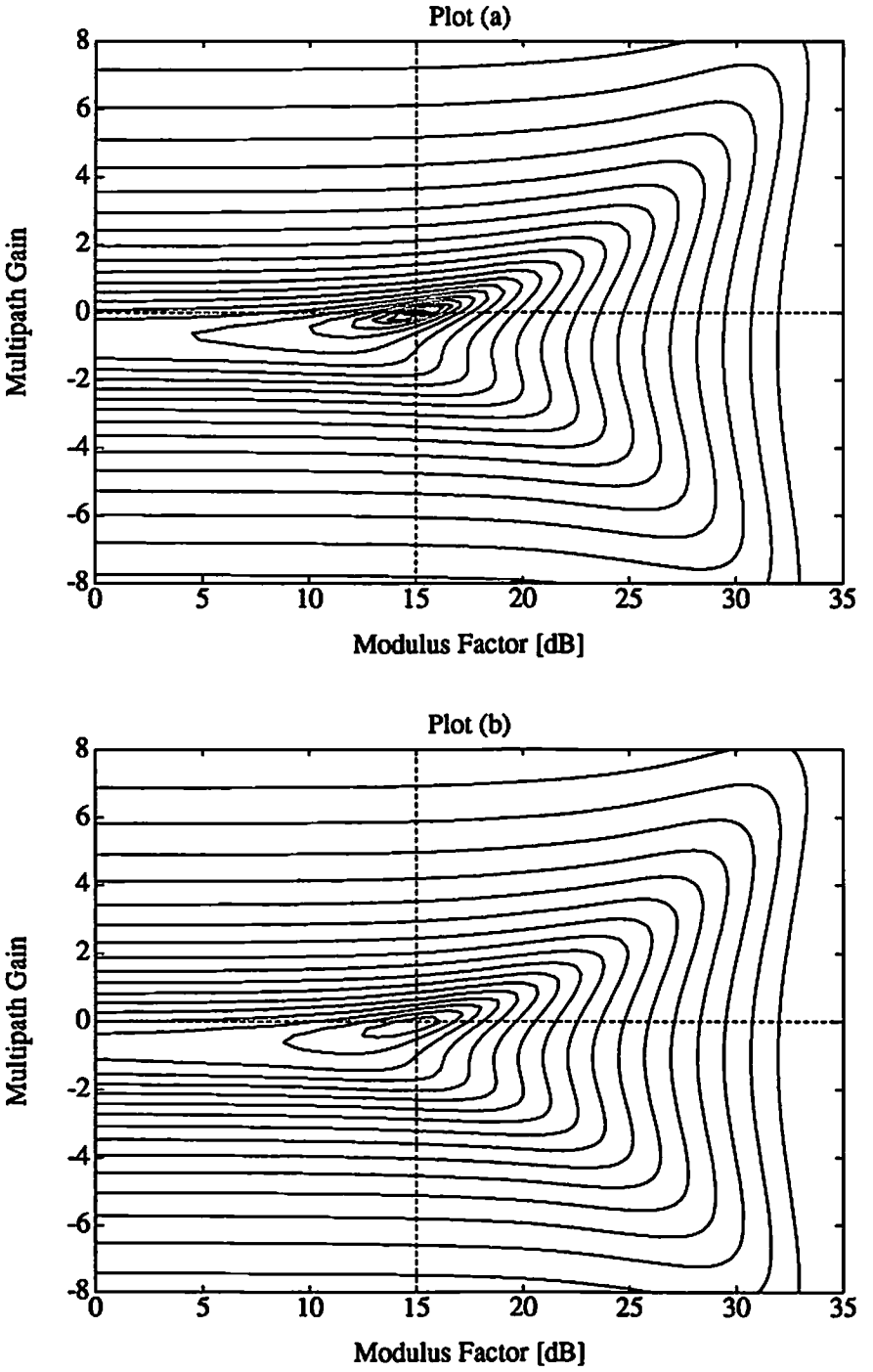

Figure 4.9: Contour plots of LCCM cost as a function of δ and the gain on the correlated signal. Signal model #2 with ρ =0.76 and $\hat{\rho}$=0.92. Plot (a): Noise absent. Plot (b): Noise present.

noise. The gain γ_0 on the decorrelation noise for Figure 4.9a was chosen such that the value of $\hat{\rho}$ was approximately equal to that used in Figure 4.5a (refer to Tables 4.1 and 4.2.) Interestingly enough, there is no second local minimum in Figure 4.9a as there is in Figure 4.5a. This is true in spite of the fact that the correlation coefficient ρ is actually higher for the scenario of Figure 4.9a. One would think that this would make the second local minimum even more prominent but in fact a second minimum does not exist at all. This is a demonstration that the nature of the cost function depends on more than just the two statistical parameters ρ and $\hat{\rho}$.

For Figure 4.10a, the parameter γ_0 was selected such that the resultant value of $\hat{\rho}$ was approximately equal to that used in Figure 4.7a (refer to Tables 4.1 and 4.2.) These two figures and their noise-present counterparts (Figures 4.7b and 4.10b) are very similar. Like Figure 4.7b, Figure 4.10b does not have a local minimum at $\gamma = 0$ for δ a free variable. A local minimum does exist at $\gamma \approx 0$ along the $\delta = 15$ dB slice, however, just as it does in Figure 4.7b. Again, it is evident that for high correlation cases that a fixed δ may be preferable to a variable δ.

The two signal models used in this numerical investigation demonstrate the complexity of the constant modulus cost function. In some comparable cases the results are very similar while for others the plots look much different. And this is true for signal models that are extremely simple and unlikely to be encountered in practice. The one general conclusion that seems fair is that the robustness of the LCCM method may break down when the signals are very highly correlated.

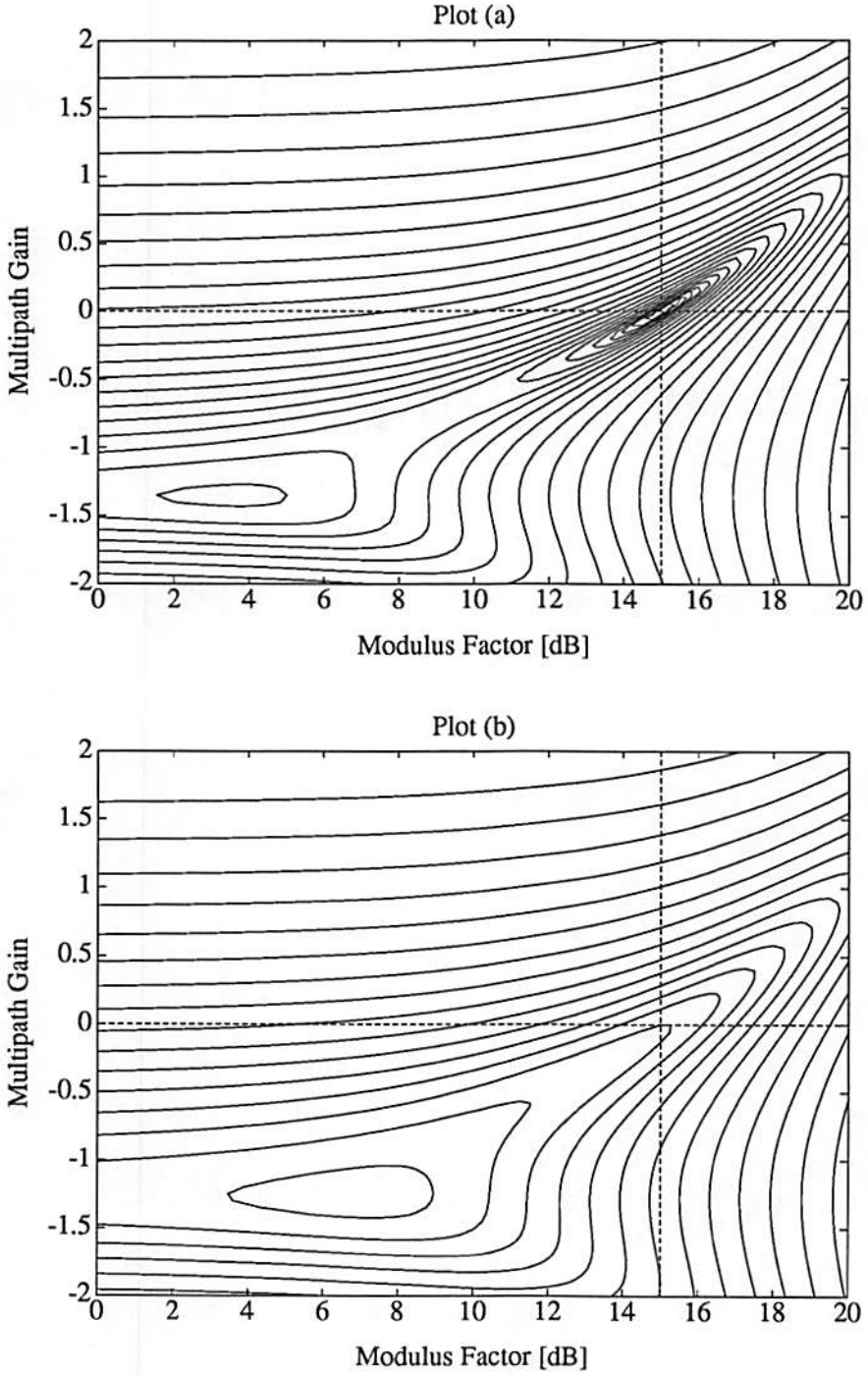

Figure 4.10: Contour plots of LCCM cost as a function of δ and the gain on the correlated signal. Signal model #2 with ρ =0.96 and $\hat{\rho}$=0.98. Plot (a): Noise absent. Plot (b): Noise present.

4.6 Summary

On the whole, the results of this chapter support the notion that linear constraints can improve the characteristics of the constant modulus cost function. In all the cases analyzed where signals are independent the LCCM cost expression had only one local minimum: when the interfering signals are nulled. In contrast, the unconstrained CM cost function has a local minimum for every signal that has kurtosis less than two [31]. Each local minimum corresponds to an instance of capture by an interfering signal. Recall that the capture phenomenon was a principal motivation for the LCCM method. Clearly, the results of this chapter show that linear constraints eliminate the capture problem for independent signals.

The notion of making δ a free variable rather than a fixed parameter was also supported by the results of this chapter. Analysis and experiments demonstrated that simultaneous minimization with respect to δ and γ resulted in local minimums at their desired values in most cases. This implies that if the optimum δ is unknown, δ can be made adaptive in the same way that the weights can be made adaptive with reasonable assurance that it will reach its desired value. The counter-examples to this conclusion were scenarios where the signals had very high correlation. In these cases an adaptive δ would be detrimental but it was also shown that if δ was fixed rather than variable the results would probably be satisfactory. Although the general question of whether δ should be fixed or adaptive was addressed here, it was not answered unequivocally. In most cases, an adaptive δ would be preferred for reasons of flexibility. However, for highly correlated signals a fixed value δ may be more appropriate. To be sure, any adaptive implementation of the LCCM method must allow δ to be fixed as well

as variable. The issue of fixed vs. variable δ will be revisited in the simulations of Chapter 6.

It is important to realize that all of the analysis in this chapter was for one application: the narrowband adaptive array. The application of the LCCM approach is not limited to arrays, however. Like the UCM algorithm, the LCCM approach is applicable to equalizers and other single channel adaptive filters. In fact, the equalizer may be its most useful application. The problem is that the single channel LCCM implementation is far more difficult to analyze than the narrowband array implementation. As such, simulations become the analysis tool for applications other than the narrowband array. An equalizer application of LCCM is simulated in Section 6.3.

The goal of this chapter was to determine whether or not the LCCM approach offered enough potential to justify the development of an adaptive algorithm. The analysis and numerical results of this chapter suggest that this potential exists. In the next chapter, the LCCM adaptive algorithm is derived from the LCCM problem statement.

Chapter 5

LCCM Adaptive Algorithms

5.1 Derivation

The derivation of the LCCM adaptive algorithm from the LCCM cost function is straightforward and divided into three parts: transformation to an unconstrained form, implementation on a special filter structure, and derivation of the adaptive recursions.

5.1.1 Constraint Elimination

The first step is to transform the LCCM problem statement

$$\min_{\mathbf{w}} E\left\{\left||y|^2-\delta\right|^q\right\} \qquad \text{subject to} \qquad \mathbf{C}^\dagger\mathbf{w}=\mathbf{f} \tag{5.1}$$

into an unconstrained form. This is accomplished through a constraint elimination approach.

Clearly, any weight vector $\mathbf{w}$ that satisfies

$$\mathbf{C}^\dagger\mathbf{w}=\mathbf{f} \tag{5.2}$$

can be decomposed into constrained and unconstrained components. The constrained component is a linear combination of the columns of the constraint matrix C and the unconstrained component is a linear combination of the basis vectors of $C^{\perp}$, the orthogonal subspace of C. With this decomposition in mind, the change of variables

$$\mathbf{w} = \mathbf{w}_q - \mathbf{W}_s \mathbf{w}_a \tag{5.3}$$

is made.

The vector $\mathbf{w}_q$ corresponds to the constrained component of w and is defined as

$$\mathbf{w}_q \triangleq \mathbf{C}\left(\mathbf{C}^{\dagger}\mathbf{C}\right)^{-1}\mathbf{f}. \tag{5.4}$$

It is the unique combination of the columns of C that satisfies the constraint equation (5.2).

Likewise, the product $\mathbf{W}_s\mathbf{w}_a$ corresponds to the unconstrained component. The columns of matrix $\mathbf{W}_s$ are defined to be any basis for $C^{\perp}$, i.e. $\mathbf{W}_s$ is full rank and satisfies

$$\mathbf{C}^{\dagger}\mathbf{W}_s = \mathbf{0}. \tag{5.5}$$

The vector $\mathbf{w}_a$, therefore, determines a particular linear combination of the basis vectors of $C^{\perp}$.

The decomposition property is easily verified by substituting the the decomposed weight vector (5.3) into the constraint equation (5.2),

$$\mathbf{C}^{\dagger}\mathbf{w} = \mathbf{C}^{\dagger}\left(\mathbf{w}_q - \mathbf{W}_s\mathbf{w}_a\right) = \mathbf{C}^{\dagger}\mathbf{w}_q - \mathbf{C}^{\dagger}\mathbf{W}_s\mathbf{w}_a = \mathbf{C}^{\dagger}\mathbf{C}\left(\mathbf{C}^{\dagger}\mathbf{C}\right)^{-1}\mathbf{f} = \mathbf{f} \tag{5.6}$$

where the matrix product $\mathbf{C}^{\dagger}\mathbf{W}_s = \mathbf{0}$ by definition (5.5). Clearly, the constraint equation (5.2) is satisfied for all values of $\mathbf{w}_a$.

Through the change of variables in (5.3), the constrained minimization problem in (5.1) has been transformed into an unconstrained problem of reduced dimension,

$$\min_{\mathbf{w}_a} E\left\{\left||y|^2 - \delta\right|^q\right\}. \tag{5.7}$$

Note that the minimization is now over the unconstrained vector $\mathbf{w}_a$ rather than the constrained vector $\mathbf{w}$. If the vector $\mathbf{w}$ has dimension N then $\mathbf{w}_a$ has dimension $N - M$ where M is the number of linear constraints.

5.1.2 GSC Filter Structure

In its adaptive form, the weight vector decomposition is time-varying,

$$\mathbf{w}(n) = \mathbf{w}_q - \mathbf{W}_s\mathbf{w}_a(n) \tag{5.8}$$

Note that $\mathbf{w}_q$ and $\mathbf{W}_s$ are fixed, non-adaptive components. This decomposition suggests that, given $\mathbf{w}_a(n)$, the overall weight vector $\mathbf{w}(n)$ must be computed for every adaptive iteration and then applied to the data $\mathbf{x}(n)$ to get an output,

$$y(n) = \mathbf{w}^\dagger(n)\mathbf{x}(n). \tag{5.9}$$

This is clearly a computational burden.

A more efficient approach to linearly-constrained processing is to consider the decomposition of $\mathbf{w}(n)$ as it applies to the data vector $\mathbf{x}(n)$. By substituting (5.8) into (5.9), the output can be rewritten as

$$\begin{aligned} y(n) &= \mathbf{w}_q^\dagger\mathbf{x}(n) - \mathbf{w}_a^\dagger(n)\mathbf{W}_s^\dagger\mathbf{x}(n) && (5.10)\\ &= y_q(n) - \mathbf{w}_a^\dagger(n)\mathbf{x}_a(n) && (5.11)\\ &= y_q(n) - y_a(n) && (5.12) \end{aligned}$$

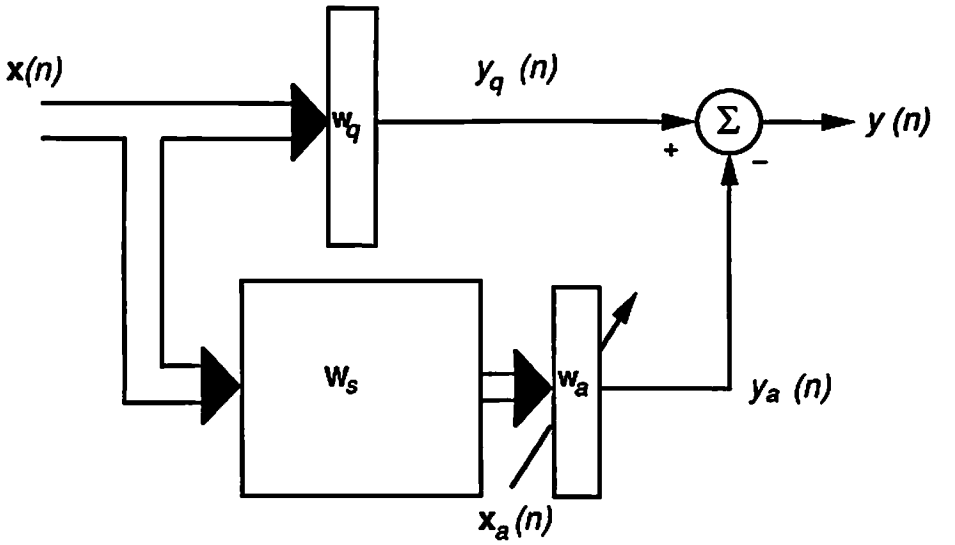

Figure 5.1: The Generalized Sidelobe Canceller Structure (GSC)

where $\mathbf{x}_a(n) = \mathbf{W}_s^\dagger \mathbf{x}(n)$ is the transformed data vector, $y_q(n) = \mathbf{w}_q^\dagger \mathbf{x}(n)$, and $y_a(n) = \mathbf{w}_a^\dagger(n)\mathbf{x}_a(n)$. Clearly, if $y_q(n)$ and $\mathbf{x}_a(n)$ are formed from the data vector $\mathbf{x}(n)$ then the overall weight vector (5.8) does not need to be computed to generate an output $y(n)$. The output is found by simply subtracting $y_a(n)$ from $y_q(n)$.

A convenient structure for forming $y_q(n)$ and $\mathbf{x}_a(n)$ is the Generalized Sidelobe Canceller (GSC) [11, 34, 35, 36]. A block diagram of the GSC which shows signal paths for $y_q(n)$ and $\mathbf{x}_a(n)$ is given in Figure 5.1. As is seen, the elements of the GSC structure correspond directly to the weight vector decomposition (5.3).

Although the GSC is basically a matrix filter, the flexibility available in the design of $\mathbf{W}_s$ can greatly reduce the amount of computation required. Recall that the only requirement for $\mathbf{W}_s$ is that it be full rank and satisfy (5.5). It has been shown [10] that for a filter of order N with M constraints, a canonical form for $\mathbf{W}_s$ exists that only requires $O(MN)$ computation. In most LCCM applications only

one constraint is necessary and so a $\mathbf{W}_s$ matrix can be constructed that needs only $O(N)$ computations.

In the single constraint case the matrix of constraint vectors $\mathbf{C}$ becomes a single vector of non-zero, complex elements.

$$\mathbf{C} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-2} \\ c_{N-1} \end{bmatrix}. \tag{5.13}$$

Given $\mathbf{C}$, a simple full rank matrix $\mathbf{W}_s$ that satisfies $\mathbf{C}^\dagger \mathbf{W}_s = \mathbf{0}$ is

$$\mathbf{W}_s = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\frac{c_0^*}{c_1^*} & 1 & & 0 \\ 0 & -\frac{c_1^*}{c_2^*} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & & 1 \\ 0 & 0 & \cdots & -\frac{c_{N-2}^*}{c_{N-1}^*} \end{bmatrix} \tag{5.14}$$

Note that the $N-1$ columns of $\mathbf{W}_s$ are linearly independent and orthogonal to $\mathbf{C}$. They therefore form a particularly simple basis for the subspace that is orthogonal to $\mathbf{C}$. When a simple $\mathbf{W}_s$ like this is used in the GSC (Fig. 5.1), the overall computation required is of the same order as a vector not a matrix.

5.1.3 Weight Vector Recursions

The constraint elimination technique of Section 5.1.1 allows the minimization over the vector $\mathbf{w}$ (3.4) to be reformulated as an unconstrained minimization over the

reduced dimension vector $\mathbf{w}_a$.

$$\min_{\mathbf{w}_a} E\left\{\left||y|^2 - \delta\right|^q\right\} \tag{5.15}$$

In turn, the GSC structure of Section 5.1.2 offers an efficient implementation of the constraint elimination technique. Unfortunately, whether constrained or unconstrained, the expectation operation $E\{\cdot\}$ of the CM cost function has only been studied for the special cases examined in Chapter 4 and is difficult to evaluate in general. No closed form solution for the weight vector $\mathbf{w}_a$ is known.

Since solving for $\mathbf{w}_a$ is impossible directly, LCCM makes use of the iterative technique Stochastic Gradient Descent (SGD.) The SGD method is basically an approximation of the Steepest Descent approach which is widely used in optimization problems. In Steepest Descent, a local minimum of the function is found by using the negative gradient in a simple recursion. For LCCM the recursion is

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) - \mu \nabla_{\mathbf{w}_a} \mathcal{J} \tag{5.16}$$

where the parameter μ controls convergence and $\mathcal{J}$ represents the CM cost function,

$$\mathcal{J} = E\left\{\left||y|^2 - \delta\right|^q\right\} \tag{5.17}$$

In SGD, an estimate of the gradient, $\tilde{\nabla}$, is used,

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) - \mu \tilde{\nabla}_{\mathbf{w}_a} \tag{5.18}$$

For LCCM, as in most SGD algorithms, the gradient estimate is found by dropping the expectation, i.e.,

$$\nabla_{\mathbf{w}_a} \mathcal{J} = \nabla_{\mathbf{w}_a} E\left\{\left||y(n)|^2 - \delta\right|^q\right\} \tag{5.19}$$

$$\downarrow \qquad \downarrow \tag{5.20}$$

$$\tilde{\nabla}_{\mathbf{w}_a} \mathcal{J} = \nabla_{\mathbf{w}_a} \left||y(n)|^2 - \delta\right|^q \tag{5.21}$$

Although no explicit averaging is done, some implicit smoothing does occur due to the recursive nature of the steepest descent iteration.

In the adaptive filter literature, these approximate gradients are described as *instantaneous* or *noisy* [37] since they may vary widely from iteration to iteration. The general instantaneous LCCM gradient is calculated using the chain rule yielding,

$$\nabla_{\mathbf{w}_a}\left(\left||y(n)|^2-\delta\right|^q\right) = q\left||y(n)|^2-\delta\right|^{q-1}\operatorname{sgn}\left(|y(n)|^2-\delta\right)\nabla_{\mathbf{w}_a}\left(|y(n)|^2-\delta\right). \tag{5.22}$$

Since the filter output $y(n)$ is given by

$$y(n) = \mathbf{w}_q^\dagger \mathbf{x}(n) - \mathbf{w}_a^\dagger \mathbf{W}_s^\dagger \mathbf{x}(n) \tag{5.23}$$

it can be shown that

$$\nabla_{\mathbf{w}_a}\left(|y(n)|^2\right) = -2y^*(n)\mathbf{W}_s^\dagger \mathbf{x}(n) \tag{5.24}$$

$$= -2y^*(n)\mathbf{x}_a(n) \tag{5.25}$$

where $\mathbf{x}_a(n) = \mathbf{W}_s^\dagger \mathbf{x}(n)$ (Figure 5.1.)

Therefore, the general LCCM gradient is

$$\nabla_{\mathbf{w}_a}\left(\left||y(n)|^2-\delta\right|^q\right) = -2q\left||y(n)|^2-\delta\right|^{q-1}\operatorname{sgn}\left(|y(n)|^2-\delta\right)y^*(n)\mathbf{x}_a(n) \tag{5.26}$$

and the general LCCM algorithm is found by substituting the approximate gradient (5.26) into the SGD recursion (5.18),

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) + 2\mu\, q\left||y(n)|^2-\delta\right|^{q-1}\operatorname{sgn}\left(|y(n)|^2-\delta\right)y^*(n)\mathbf{x}_a(n) \tag{5.27}$$

For this dissertation, the two most useful LCCM forms are given when $q = 1$ and when $q = 2$, they are the so-called 2-1 and 2-2 algorithms. They are given by

$$\begin{aligned} \text{2-1} \quad \mathbf{w}_a(n+1) &= \mathbf{w}_a(n) + \mu \operatorname{sgn}\left(|y(n)|^2 - \delta\right) y^*(n)\mathbf{x}_a(n) \\ \text{2-2} \quad \mathbf{w}_a(n+1) &= \mathbf{w}_a(n) + \mu \left(|y(n)|^2 - \delta\right) y^*(n)\mathbf{x}_a(n) \end{aligned} \tag{5.28}$$

where the constants have been absorbed into μ. The details of the convergence parameter μ are discussed in Section 5.2 which deals with stability.

It is important to realize that the LCCM iterations do not, in themselves, constitute the LCCM algorithm. The LCCM algorithm is either of the 2-1 or 2-2 iterations together with the GSC filter structure (Figure 5.1.)

5.1.4 Adaptive Modulus Factor Recursions

Since the optimal value of δ is unknown in some cases and time-varying in others it is necessary to have an adaptive approach available for its determination.

Partitioned Gradient

One way to make δ adaptive is to first define a new adaptive weight vector $\mathbf{h}$ that is partitioned or stacked to include δ and $\mathbf{w}_a$

$$\mathbf{h} = \begin{pmatrix} \mathbf{w}_a \\ \delta \end{pmatrix} \tag{5.29}$$

The minimization problem then becomes

$$\min_{\mathbf{h}} E\left\{\left||y(n)|^2 - \delta\right|\right\} \tag{5.30}$$

for which a stochastic gradient can easily be calculated and an adaptive algorithm formed.

The stochastic gradient is partitioned as,

$$\tilde{\nabla}_{\mathbf{h}} = \begin{pmatrix} \tilde{\nabla}_{\mathbf{w}_a} \\ \tilde{\nabla}_{\delta} \end{pmatrix} \tag{5.31}$$

where $\tilde{\nabla}_{\mathbf{w}_a}$ is identical to (5.26) and $\tilde{\nabla}_{\delta}$ is given by

$$\tilde{\nabla}_{\delta} = -\text{sgn}\left(|y(n)|^2 - \delta(n)\right). \tag{5.32}$$

The stochastic gradient is then substituted into the standard steepest descent iteration yielding,

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu\,\text{sgn}\left(|y(n)|^2 - \delta(n)\right) \begin{pmatrix} y^*(n)\mathbf{x}_a(n) \\ 1 \end{pmatrix}. \tag{5.33}$$

Although it is the most straightforward technique for adapting δ, the partitioned approach results in the slowest converging algorithm. The reason for this is that the same convergence parameter μ is used for both the $\mathbf{w}_a$ and δ partitions of the recursion. This is appropriate if $\mathbf{w}_a$ and δ are strongly coupled but needlessly slows convergence if they are not.

Independent Minimization

In most LCCM applications, the values of $\mathbf{w}_a$ and δ are relatively uncoupled and it is preferable to separate the update for $\mathbf{h}$ into two recursions,

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) + \mu_{\mathbf{w}}\text{sgn}\left(|y(n)|^2 - \delta(n)\right) y^*(n)\mathbf{x}_a(n) \tag{5.34}$$

and

$$\delta(n+1) = \delta(n) + \mu_{\delta}\,\text{sgn}\left(|y(n)|^2 - \delta(n)\right) \tag{5.35}$$

where $\mu_{\mathbf{w}}$ and μ_{δ} are chosen independently. This allows much faster convergence since the value of μ_{δ} can typically be made an order of magnitude greater than the value of $\mu_{\mathbf{w}}$.

The fact that $\mathbf{w}_a$ and δ are relatively uncoupled also makes it possible to have adaptive iterations for $\mathbf{w}_a$ and δ that are based on different cost functions. This approach offers even more flexibility than using different values for μ_δ and $\mu_{\mathbf{w}}$. For instance, the 2-1 cost function can be used for minimization w.r.t. $\mathbf{w}_a$ and the 2-2 cost function can be used for the minimization w.r.t. δ. The 2-2 minimization expression for δ is

$$\min_{\delta} E\left\{\left||y|^2 - \delta\right|^2\right\} \tag{5.36}$$

from which the adaptive recursion

$$\delta(n+1) = \delta(n) + \mu_\delta\left(|y(n)|^2 - \delta(n)\right) \tag{5.37}$$

can be easily derived.

Rearrangement of the iteration in (5.37) yields

$$\delta(n+1) = (1-\mu_\delta)\delta(n) + \mu_\delta|y(n)|^2 \tag{5.38}$$

which is an unbiased recursive estimator of the output power $E\{|y|^2\}$. In this form, the parameter μ_δ controls the roll-off of the exponential function used to window the magnitude squared output samples $|y(n)|^2$. For a fixed weight vector $\mathbf{w}_a$, this will converge to the average output power. If the weight vector $\mathbf{w}_a$ is at its optimal value then (5.37) will converge to δ_{opt} (3.10).

However, the weight vector $\mathbf{w}_a$ is not fixed to its optimal value but instead is typically adapting toward it. The convergence of δ is thus dependent on the dynamics of the relationship between δ and $\mathbf{w}_a$. These dynamics and the selection of $\mu_{\mathbf{w}}$ and μ_δ are discussed in the next section.

5.2 Stability & Convergence

The two parameters μ and δ are the only undefined quantities in the preceding section. All others are defined by the GSC structure together with the input data vector $\mathbf{x}$. For now it will be assumed that the parameter δ is fixed and non-adaptive. The adaptive case will be discussed in Section 5.2.2.

5.2.1 Fixed δ Case

Any discussion of the stability of SGD algorithms begins with the convergence parameter μ. The parameter μ controls the stability and convergence rate of LCCM in the same way that it does for other stochastic gradient descent algorithms such as LMS. To implement an algorithm based on SGD it is desirable to determine the bounds on μ for stable operation. Although it is relatively straightforward to calculate the bound for other algorithms such as LMS, it is more difficult to do so for UCM and LCCM. With LMS, the bounds on μ are found by first determining the mean weight vector recursion [28]. In this method, the expected value of the adaptive update is evaluated to determine the bounds on μ such that the weight vector converges in mean. (Convergence of the variance of weight elements is a more restrictive condition that is described in [4, 38, 7].) An approach based on the convergence of the mean weight vector is not easy to extend to the recursions used in UCM or LCCM. This is because the calculation of the mean weight vector recursion for UCM or LCCM is made intractable by the presence of the non-linear term $|y|^2$ in the update iteration. In other words, the same problem that makes the local minima of the LCCM cost function difficult to determine makes the the stability of the LCCM adaptive algorithm difficult to analyze.

Since convergence in mean is difficult if not impossible to show, an alternative approach is used in the stability analysis of the LCCM algorithm. Basically, the LCCM recursion is compared with the recursion used in the LCMP method. The stability and convergence of the LCMP iteration is well established and bounds on μ are derived in [11]. By comparison with LCMP, the stability of the LCCM algorithm can be studied.

The LCMP approach uses the same GSC decomposition structure as LCCM and a very similar recursion for the adaptive weight vector $\mathbf{w}_a$ given by

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) + \mu y^*(n)\mathbf{x}_a(n). \tag{5.39}$$

A loose but practical bound on μ for this recursion [11] is given by

$$0 \leq \mu < \frac{2}{\mathrm{tr}(\mathbf{R}_{x_a x_a})} \tag{5.40}$$

where the matrix $\mathbf{R}_{x_a x_a} = \mathbf{W}_s^\dagger \mathbf{R}_{xx} \mathbf{W}_s$ and $\mathbf{W}_s$ is the GSC blocking matrix defined in Section 5.1.2.

If it is assumed that

$$\left| |y(n)|^2 - \delta \right| > 0 \tag{5.41}$$

then the 2-1 LCCM recursion

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) + \mu \,\mathrm{sgn}\left(|y(n)|^2 - \delta\right) y^*(n)\mathbf{x}_a(n) \tag{5.42}$$

is identical to the LCMP recursion (5.39). This implies that as long as (5.41) holds, the LCCM iteration is identical to the LCMP iteration and therefore LCCM minimizes total output power. It follows that as long as (5.41) is satisfied, the bounds on μ for the LCMP algorithm (5.40) apply to the LCCM algorithm as well. The fact that (5.41) is not always satisfied is deferred for the moment in order to derive a normalized version of μ.

In practice, the trace of $\mathbf{R}_{x_a x_a}$ is unknown and will be time varying if the signal environment is non-stationary. Therefore the the value of $\mathrm{tr}(\mathbf{R}_{x_a x_a})$ must be estimated and tracked to ensure that μ satisfies (5.40). Tracking of $\mathrm{tr}(\mathbf{R}_{x_a x_a})$ can be easily accomplished using an unbiased recursive estimator of the following form

$$\sigma_{xa}^2(n) = \gamma \sigma_{xa}^2(n-1) + (1-\gamma)\left|\mathbf{x}_a(n)\right|^2. \tag{5.43}$$

where $\sigma_{xa}^2 \triangleq \mathrm{tr}(\mathbf{R}_{x_a x_a})$. This recursion makes use of the fact that

$$\mathrm{tr}(\mathbf{R}_{x_a x_a}) = \mathrm{tr}\left(E\left\{\mathbf{x}_a \mathbf{x}_a{}^{\dagger}\right\}\right) = \mathrm{tr}\left(E\left\{\mathbf{x}_a{}^{\dagger}\mathbf{x}_a\right\}\right). \tag{5.44}$$

The parameter γ in (5.43) can be thought of as a "forgetting factor" used to weigh recent data in the estimate more heavily than older data. More precisely, it defines the roll-off of the exponential function used to window the samples of $\mathbf{x}_a^{\dagger}(n)\mathbf{x}_a(n)$ This recursive estimator, actually a one pole filter, is BIBO stable for $|\gamma| < 1$. Typical practical values for γ range from 0.95 to 0.999 depending on the degree of non-stationarity present in the signal environment.

A convenient time-varying form for μ that incorporates the non-stationary nature of $\mathrm{tr}(\mathbf{R}_{x_a x_a})$ is

$$\mu(n) \triangleq \frac{\alpha}{\sigma_{xa}^2(n)}. \tag{5.45}$$

By comparison with (5.40), it is clear that α must satisfy $0 < \alpha < 2$. The advantage of this re-parameterization is that α is normalized whereas μ is data dependent [39]. It is normalized in the sense that similar convergence performance can be expected when the same value of α is used in the application of LCCM to two very different signal environments. This is not the true if the same value of μ is used.

In addition, the value of α gives an indication of the variance of the adaptive weight vector elements. The variance of each weight vector element is typically combined into a single measure called misadjustment [28]. As a general rule, when the convergence rate is improved by increasing α, the average misadjustment also increases. Like other SGD algorithms, a tradeoff exists between convergence rate and misadjustment. For highly non-stationary applications a relatively large value of α is appropriate while for other applications convergence rate may be less important and a small α is used to keep misadjustment to a minimum. A practical range for α is

$$0.01 < \alpha < 0.05 \tag{5.46}$$

since experience with LCCM and other SGD algorithms shows that it offers a good compromise between convergence rate and misadjustment.

For greater algorithm clarity, $\mu(n)$ can be redefined to include the sgn$(\cdot)$ operation

$$\mu(n) \triangleq \text{sgn}\left(|y(n)|^2 - \delta\right) \frac{\alpha}{\sigma_{xa}^2(n)} \tag{5.47}$$

and the weight vector iteration then becomes

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) + \mu(n) y^*(n) \mathbf{x}_a(n). \tag{5.48}$$

As long as (5.41) is satisfied the above recursion is stable and converges to

$$\mathbf{w}_a \rightarrow \left(\mathbf{W}_s^\dagger \mathbf{R}_{xx} \mathbf{W}_s\right)^{-1} \mathbf{W}_s^\dagger \mathbf{R}_{xx} \mathbf{w}_q, \tag{5.49}$$

the minimum power solution for the reduced dimension weight vector $\mathbf{w}_a$ [11].

Of course, the inequality in (5.41) cannot be valid for all iterations. If it were the LCCM algorithm would simply converge to the minimum power solution all

the time and there would be no difference between LCCM and LCMP. What happens instead is that when the algorithm iterates to a point where the squared modulus is less than δ, i.e. $||y(n)|^2 - \delta| < 0$, $\mu(n)$ changes sign. In effect, when the sign of $\mu(n)$ changes the LCCM algorithm is no longer steepest *descent* but steepest *ascent*. The weight vector then iterates in a direction that generally increases output power. This cannot continue indefinitely however since at some point $|y(n)|^2$ will again be greater than δ, i.e. (5.41) is satisfied once again, and the cycle will start over again. It is important to note that adaptation is still taking place during these cycles because the sign changes in $\mu(n)$ are correlated with the incoming data vector $\mathbf{x}_a(n)$. It is this correlation that directs the LCCM algorithm to a useful solution rather than simply causing the output power to randomly oscillate about the value of δ. The algorithm converges when the output is such that the sign changes in $\mu(n)$ are uncorrelated with $y^*(n)\mathbf{x}_a(n)$. Unfortunately, calculation of the weight vector that corresponds to this case is impossible in general due to the difficulty of evaluating the expectation necessary for the correlation.

There is a special case, however, where the correlation between $\mu(n)$ and $y^*(n)\mathbf{x}_a(n)$ is known and the optimal weight vector can be calculated. This occurs when the value of δ is significantly less than the Signal of Interest (SOI) power. The value of δ is significantly less than the output power if

$$\text{sgn}\left(E\left\{|y(n)|^2\right\} - \delta\right) = \text{sgn}\left(|y(n)|^2 - \delta\right) \tag{5.50}$$

is true. If the SOI lies in the constraint subspace and the interference is uncorrelated with the SOI, then the output power can never be less than δ. There is no way to add uncorrelated signals such that the power of their sum is less than the

sum of their powers. If (5.50) is true then the sign of $\mu(n)$ will be positive for all adaptive weight iterations. As previously described, under these circumstances the LCCM algorithm is identical to the LCMP algorithm in all respects and it will converge to the minimum power solution. When the power is minimized, $E\{y^*\mathbf{x}_a\} = 0$ which is equivalent to $\mathbf{W}_s^\dagger \mathbf{R}_{xx}\mathbf{w} = 0$. This equation together with the constraint equation $\mathbf{C}^\dagger\mathbf{w} = \mathbf{f}$ defines a unique solution for the weight vector $\mathbf{w}$ given by

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1}\mathbf{C}(\mathbf{C}^\dagger\mathbf{R}_{xx}^{-1}\mathbf{C})^{-1}\mathbf{f}, \tag{5.51}$$

the minimum power solution for the full weight vector $\mathbf{w}$.

Although convergence of the LCCM algorithm can be proven for this special case, it cannot be shown for the majority of applications. This is no surprise since no closed form expression exists for the minimums of the LCCM cost function. The important result is that a stable form of the recursion can be derived by comparing the LCCM and LCMP adaptive iterations.

The stabilized 2-1 LCCM algorithm for fixed δ is as follows:

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) + \mu(n)y^*(n)\mathbf{x}_a(n) \tag{5.52}$$

$$\sigma_{xa}^2(n) = \gamma\sigma_{xa}^2(n-1) + (1-\gamma)|\mathbf{x}_a(n)|^2 \tag{5.53}$$

$$\mu(n) = \mathrm{sgn}(|y|^2 - \delta(n))\frac{\alpha_{\mathbf{w}}}{\sigma_{xa}^2(n)}. \tag{5.54}$$

5.2.2 Adaptive δ Case

The last section dealt with the stability and convergence of the LCCM algorithm when the modulus factor δ was fixed. In Section 3.3, however, the need for a variable δ was motivated and in Section 5.1.4 stable recursions were derived to

implement the adaptive determination of δ. This section deals with the stability of the LCCM algorithm when δ is adaptive.

Both of the recursions for δ in equations (5.35) and (5.37) are clearly linked to the recursion for $\mathbf{w}_a$ in equation (5.52). The recursion for $\mathbf{w}_a$ has the variable $\mu(n)$ which explicitly includes $\delta(n)$. Likewise, the both of the recursions for δ have the quantity $|y(n)|^2$ which is a function of $\mathbf{w}_a(n)$. Although they are linked, the adaptive updates for $\mathbf{w}_a$ and δ are relatively uncoupled and so it is assumed that the stability of the two recursions operating simultaneously can be ensured if the two recursions are independently stable.

The 2-1 adaptive iteration for δ rewritten here

$$\delta(n+1) = \delta(n) + \mu_\delta \operatorname{sgn}\left(|y(n)|^2 - \delta(n)\right) \tag{5.55}$$

is very robust and is actually BIBO stable for all bounded values of μ_δ. The $\operatorname{sgn}(\cdot)$ operation in the update makes this recursion extremely stable.

If the magnitude squared of the output is constant, i.e.,

$$|y(n)|^2 = C \tag{5.56}$$

then it can be shown that the recursion (5.55) is locally stable. To prove this let

$$v(n) \triangleq |y(n)|^2 - \delta(n) \tag{5.57}$$

then substituting (5.57) into (5.55) yields

$$v(n+1) = v(n) - \mu_\delta \operatorname{sgn}(v(n)) \tag{5.58}$$

where

$$\operatorname{sgn}(v(n)) = \frac{v(n)}{|v(n)|} \tag{5.59}$$

and it is assumed that $v(n) \neq 0$. For local stability, $|v(n)| \leq \mu_\delta$ implies that $|v(n+1)| \leq \mu_\delta$. Rewriting (5.58) yields

$$v(n+1) = \left(1 - \mu\delta \frac{1}{|v(n)|}\right) \tag{5.60}$$

$$|v(n+1)| = \left|1 - \mu_\delta \frac{1}{|v(n)|}\right| |v(n)| \tag{5.61}$$

$$= \left||v(n)| - \mu_\delta\right|. \tag{5.62}$$

Now, since $|v(n)| \leq \mu_\delta$ then

$$\max\left\{\left||v(n)| - \mu_\delta\right|\right\} = \mu_\delta \tag{5.63}$$

and therefore

$$v(n+1)| \leq \mu_\delta. \tag{5.64}$$

Equation (5.37), the 2-2 δ iteration, is rewritten here in its recursive estimator form

$$\delta(n+1) = (1 - \mu_\delta)\delta(n) + \mu_\delta |y(n)|^2. \tag{5.65}$$

This simple first order filter is BIBO stable for $0 < \mu_\delta < 2$ and will converge in the mean to the average filter output power for a fixed weight vector $\mathbf{w}_a$. As an estimator with an exponential window, it has better tracking characteristics than the 2-1 recursion which incorporates the sgn(·) operation (5.55.) For consistency, however, the 2-1 adaptive δ iteration is used exclusively in the simulations of Chapter 6.

To review, the stabilized 2-1 LCCM algorithm with adaptive δ is as follows:

$$\mathbf{w}_a(n+1) = \mathbf{w}_a(n) + \mu(n) y^*(n) \mathbf{x}_a(n) \tag{5.66}$$

$$\delta(n+1) = \delta(n) + \alpha_\delta \mathrm{sgn}\left(|y(n)|^2 - \delta(n)\right) \tag{5.67}$$

$$\sigma^2_{xa}(n) = \gamma \sigma^2_{xa}(n-1) + (1-\gamma)|\mathbf{x}_a(n)|^2 \tag{5.68}$$

$$\mu(n) \;=\; \mathrm{sgn}\left(|y|^2 - \delta(n)\right) \frac{\alpha_{\mathbf{w}}}{\sigma^2_{xa}(n)}. \tag{5.69}$$

Note that α_δ has replaced μ_δ in (5.67.) This is for consistency with (5.69). (The parameter α_δ is in effect μ_δ normalized by 1.)

5.3 Summary

In this chapter, two versions of the LCCM adaptive algorithm have been derived: one with a fixed modulus factor the other with an adaptive modulus factor. Both versions use the GSC decomposition structure to implement the linear constraint equation which allows a great deal of flexibility in meeting computational requirements.

Both versions of the LCCM algorithm can be stabilized by appropriately normalizing the convergence parameter μ. This normalization, into a new parameter α, allows easier control over the trade-off between convergence rate and misadjustment. Practical bounds on α as well as the another minor parameter γ do not depend on the received data and are thus easy to meet in practice.

Under certain conditions, the LCCM algorithm converges to exactly the same solution as the LCMP solution. This result is a positive aspect of LCCM since under these same conditions the LCMP solution is normally the desired solution. General convergence is difficult to prove, however. Likewise, general stability for the adaptive δ version of LCCM is unknown due to the dynamic interaction between the modulus factor and the weight vector. This interaction and the overall LCCM convergence behavior are investigated in the next chapter.

Chapter 6

Applications & Simulations

6.1 Introduction

This chapter describes simulation results for single and multi-channel applications of the adaptive LCCM algorithm. The first application is the reception of a QPSK modulated signal by a 4 element adaptive array. In this case, the performance of the LCCM algorithm is compared to UCM and LCMP for a signal environment that includes severe co-channel interference as well as multipath. Both fixed and adaptive modulus factor versions of LCCM are simulated. For the fixed δ case, the effect of error in the modulus factor estimate is also investigated. The second application is a fractionally spaced equalizer that operates in an environment that, like the array, includes multipath and co-channel interference. These two applications are chosen to demonstrate the performance of the LCCM algorithm and highlight its advantages and potential disadvantages. They are not intended to be highly accurate representations of practical communication scenarios.

6.2 An Untrained Adaptive Array

The adaptive array set-up that is simulated in this section is based on a scenario in a recent paper [23]. This paper described and simulated the implementation of the UCM algorithm for a narrowband adaptive array. The signal environment consisted of a QPSK signal and its multipath[1] returns in a background of white noise with no co-channel interference present. It was shown that the UCM adaptive algorithm selects the signal with the highest initial power, normally the primary path signal, and nulls the multipath returns. This acquisition of the primary signal was done "blindly" in the sense that the direction of arrival of the primary path signal was assumed to be unknown. If this algorithm is used in an environment with severe co-channel interference, however, the array will capture an interfering signal if it has the highest initial power. This results in the suppression of the signal of interest.

Although it was assumed to be unknown in [23], there are many situations where the direction of arrival of the primary path signal is known. This *a priori* knowledge is difficult to exploit with UCM but is very simple to incorporate into LCCM as a spatial gain constraint. In the following simulations, the angle of arrival of the SOI is assumed to be known.

6.2.1 Experiment Details

In the experiments to follow, a four element square array with half-wavelength spacing is used to receive a QPSK signal arriving from a known direction. At each

[1]The delay of the multipath returns in [23] is long enough for them to be uncorrelated with the primary path signal. Basically, they are uncorrelated interferers.

array element, the received signal is quadrature downconverted, matched filtered, and synchronously sampled to yield one complex sample per QPSK symbol. These samples are then multiplied by the array weights and added together to form a single channel beam out of the multi-channel input. Finally, the beam is fed into the symbol decision stage of the receiver which completes the detection sequence.

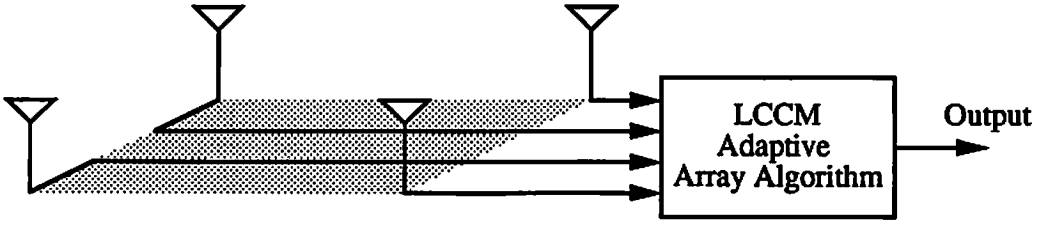

Figure 6.1: Block diagram of a four element square array for the reception of a QPSK signal. At each antenna, the received signal is match filtered and synchronously sampled. These samples are the input to the LCCM algorithm.

The array of matched filter outputs is used as the data vector $\mathbf{x}$ for several reasons. One reason is that QPSK modulated signals are often pre-processed with pulse shapers which reduce the signal bandwidth but also cause the signal to lose its CM property. However, the symbols themselves and the outputs of the matched filters are still CM. In the absence of pulse shaping the QPSK signal would be constant envelope and conceivably LCCM could used to process the signal before matched filtering. This is a much more involved simulation, though, and working with the matched filter outputs is much simpler.

The adaptive array receiver simulated herein operates in a signal environment that is summarized in Table 6.1. Note that the relative signal powers in this table are calculated after the matched filters rather than at the array sensors. This is

Signal	Arrival Angle	Power [dB]
QPSK SOI	0°	20
3/10 Baud Delay SOI	30°	15
CM Interferer # 1	−50°	20
CM Interferer # 2	140°	30
Complex White Noise	—	-10

Table 6.1: The Signal Environment for the Array Simulations with Multipath

because the adaptive processor operates on the matched filter outputs rather than on the raw received signal itself.

For this experiment, the signal environment contains both correlated and uncorrelated interference. The uncorrelated interference consists of two constant modulus interferers together with white noise. The correlated interference takes the form of a multipath return that is delayed 3/10 baud with respect to the SOI. This relatively short delay of 3/10 of a symbol interval is chosen to ensure that the multipath signal is correlated with the SOI. If the delay is greater than one baud the multipath will be uncorrelated with the SOI since the QPSK message sequence is assumed to be white.

Chapter 4 presents analysis of the LCCM optimization problem for the correlated signal case. These results cannot be applied directly to this simulation scenario because they are derived under a real signal assumption and the QPSK signals in the following experiments are complex. Still, it is useful to calculate the cross correlation between the SOI and the multipath signals. The 3/10 baud delay results in a significant correlation coefficient, $|\rho| = 0.92$. The higher order coefficient $\hat{\rho}$ is defined for the real case in (4.97). A complex $\hat{\rho}_c$ can be defined as

$$\hat{\rho}_c \triangleq \frac{\overline{ab|b|^2}}{\sigma_a \sigma_b \sqrt{\overline{|b|^4}}}. \tag{6.1}$$

For this experiment, $|\hat{\rho}_c|$=0.94.

6.2.2 Algorithm Comparison

The nature of this signal environment limits the choice of adaptive algorithms. Since the array must adapt without a reference signal, methods such as LMS cannot be used. It is possible to use decision direction because the SOI is digitally modulated, but the interference is so severe and the initial errors are so frequent that it is unlikely that a decision feedback algorithm would be able to "bootstrap" itself. Linearly constrained power minimization is another possible approach but the correlated multipath signal could lead to signal cancellation. As for UCM, each constant modulus interferer corresponds to a local minimum of the UCM cost function (2.23) which in turn would lead to capture and subsequent loss of the QPSK SOI.

These problems are verified in the following simulation that compares the adaptive performance of UCM and LCMP with the fixed δ implementation of LCCM. In all three algorithms the initial weight vector is the steer vector for the direction of arrival of the SOI and the convergence parameters are chosen to give comparable misadjustment performance. For the two constrained algorithms, the gain in the direction of the SOI, 0°, is constrained to be unity. Finally, the modulus factor δ for LCCM is fixed to 20 dB which is the sum of the SOI and the ambient white noise power. This is not the optimal value but is a very close approximation. The optimal value is slightly higher since the nulling of the

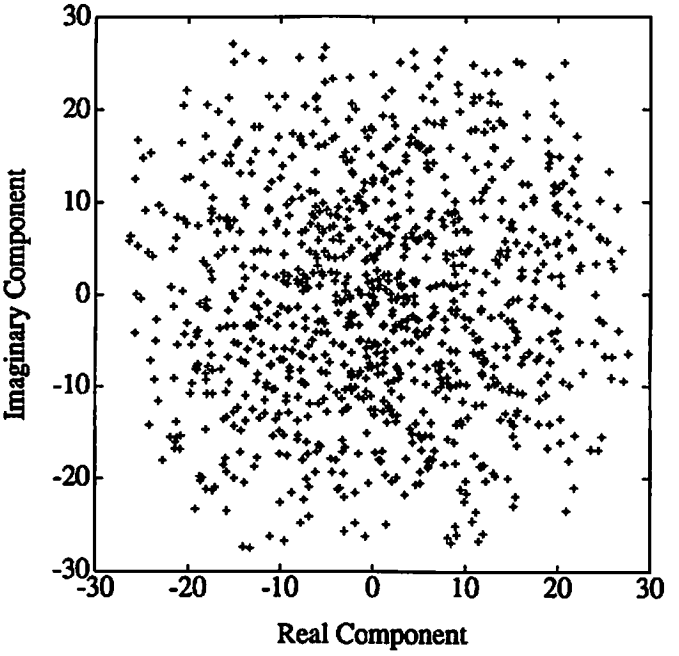

Figure 6.2: Output constellation for array steered to the direction of the SOI with no adaptation.

interference will cause an increase in output white noise power and a corresponding increase in the optimal value of δ.

Figure 6.2 shows the signal constellation that results when the array weights are fixed at their initial condition, steered towards the SOI. Note that the underlying QPSK constellation is completely masked by the interference. This is the starting point for all three algorithms. Figures 6.3a, 6.4a and 6.5a show the spatial response of the three arrays after they have converged (approx. 8000 iterations.) In the LCCM response (Fig. 6.3a), the two CM interferers are nulled along with the multipath return as desired and the signal constellation that results, Figure 6.3b, has easily resolvable QPSK symbols. The response of the LCMP array (Fig. 6.4a) looks very much like the LCCM response except that the multipath

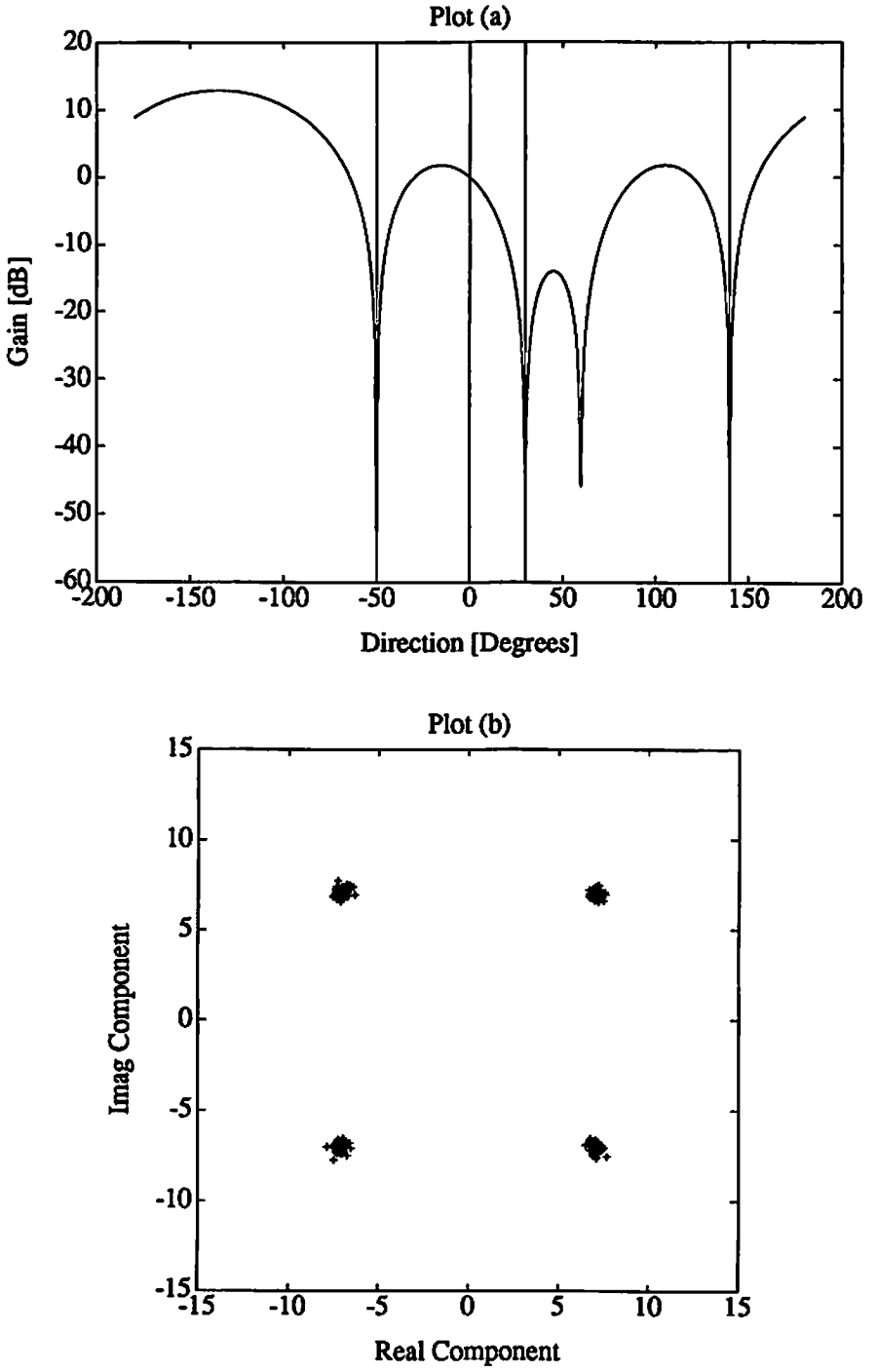

Figure 6.3: Plot (a): Converged Spatial Response of LCCM array. Plot (b): LCCM Constellation at Convergence.

signal is not nulled. Instead, the weight vector w of the LCMP algorithm puts a gain and phase on the multipath signal that minimizes the total output power and consequently causes signal cancellation. This is clearly shown in the LCMP signal constellation (Fig. 6.4b) which has less power than the LCCM constellation but no discernible QPSK symbols. The UCM algorithm, on the other hand, has a response much different than the other two methods. As can be seen in the beam response of Figure 6.5a, the UCM algorithm has been captured by one of the CM interferers and the SOI has been nulled along with the other signals. The signal constellation that results (Fig. 6.5b) shows a single CM interferer alone. Although this is certainly a minimum of the UCM cost function, it is obviously not the desired result.

In additional simulations, the three algorithms were tested in environments that had a variety of signal powers and arrival angles. In no case was the LCCM algorithm captured by a constant modulus interferer. To be fair, UCM was not always captured by constant modulus interference either but the previous simulation was not an unusual result. The LCMP algorithm, in contrast, suffered from signal cancellation in every simulation that included multipath distortion.

6.2.3 Modulus Factor Effects

This section deals with the effect of δ on the performance of the LCCM method. With LCCM there are two ways of choosing δ: it can be fixed to some constant value or it can be made adaptive. For stationary signal environments, an optimum δ exists but it requires knowledge of the optimum filter weights and is therefore impossible to calculate exactly beforehand. For non-stationary environments, δ_{opt} is time-varying and the adaptive δ LCCM implementation is preferable. This

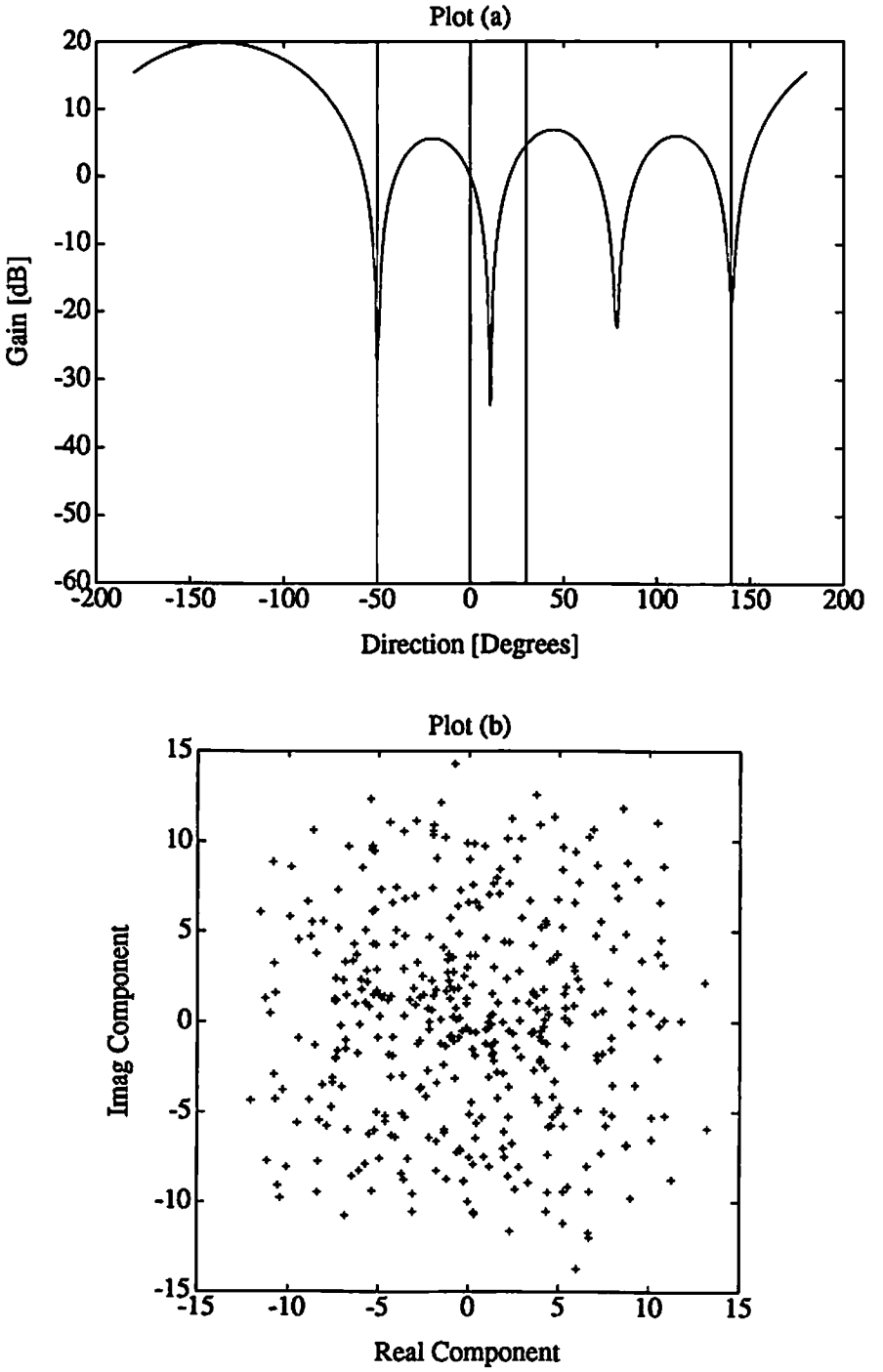

Figure 6.4: Plot (a): Converged Spatial Response of LCMP array. Plot (b): LCMP Constellation at Convergence.

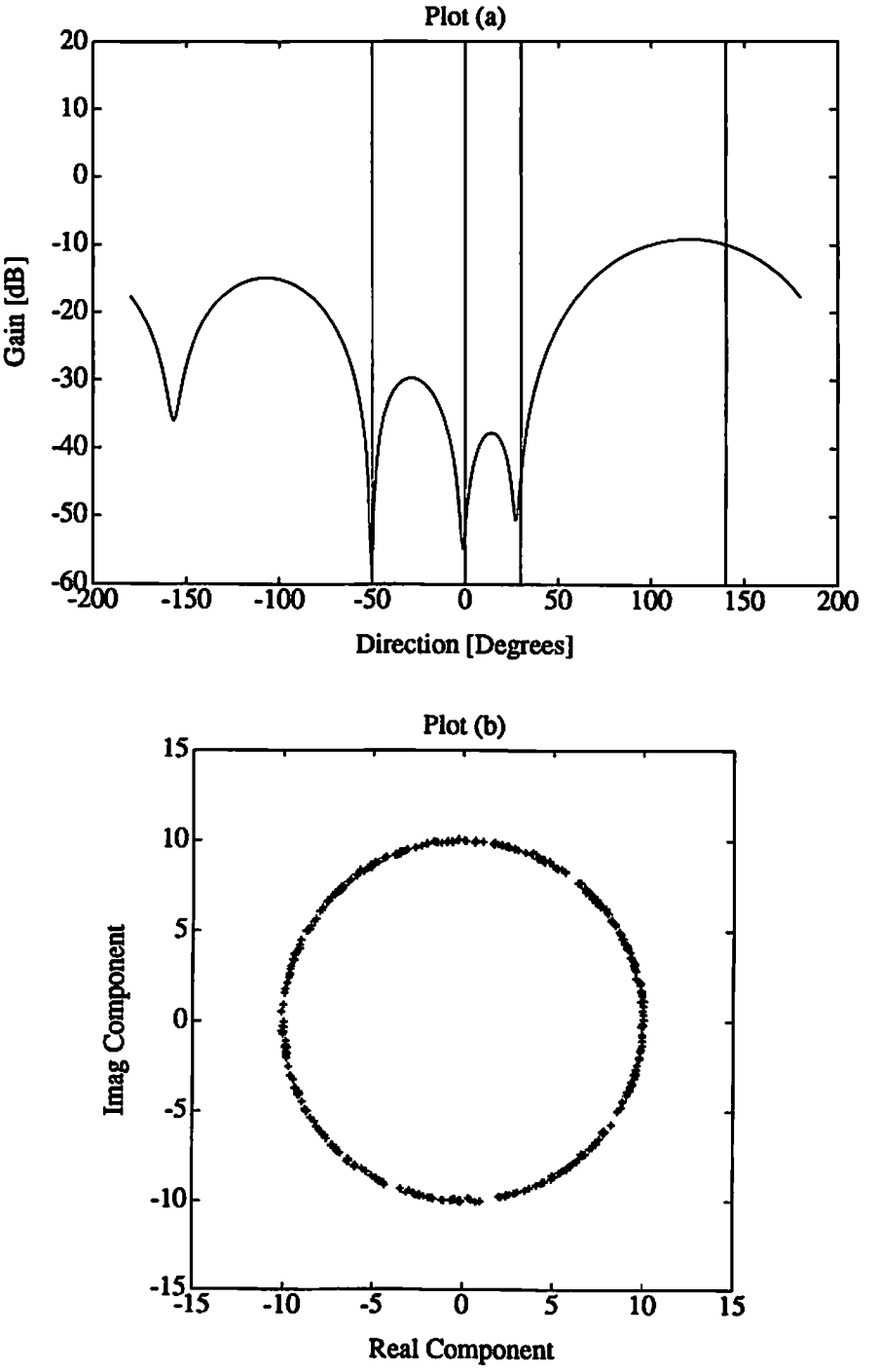

Figure 6.5: Plot (a): Converged Spatial Response of UCM array. Plot (b): UCM Constellation at Convergence.

version can also be used to estimate δ in stationary environments. For each of the following experiments, the effect of δ fixed to a value other than δ_{opt} is investigated. In addition, for each experiment the adaptive δ version of LCCM is simulated. The effect of δ on LCCM performance is different for correlated and uncorrelated interference environments. For this reason, the simulations are divided into multipath-present and multipath-absent cases.

Uncorrelated Interference: Multipath Absent

The array simulations that follow use the same array receiver structure that was described in Section 6.2.1. In addition, all of the simulations use the same value for the convergence parameter, $\alpha_{\mathbf{w}} = 0.025$. The signal environment parameters are summarized in Table 6.2.

Signal	Arrival Angle	Power
QPSK SOI	0°	20 dB
CM Interferer # 1	−50°	20 dB
CM Interferer # 2	140°	30 dB
Complex White Noise	—	0 dB

Table 6.2: Signal Environment for Modulus Factor Comparison: Multipath Absent

As in Section 6.2.2, the optimum δ value for this simulation is very close to 20 dB and it will be assumed to be exactly 20 dB for simplicity. To investigate the effects of errors in the selection of δ, three fixed values of δ are simulated: 15, 20, and 25 dB. Figure 6.6a shows the output modulus as a function of iteration for the 20 dB simulation. Convergence occurs in approximately 1000 iterations. Although they are not included here, the output modulus plots for the 15 and 25 dB cases converged in $\approx$ 1000 iterations as well. Figure 6.6b shows the converged

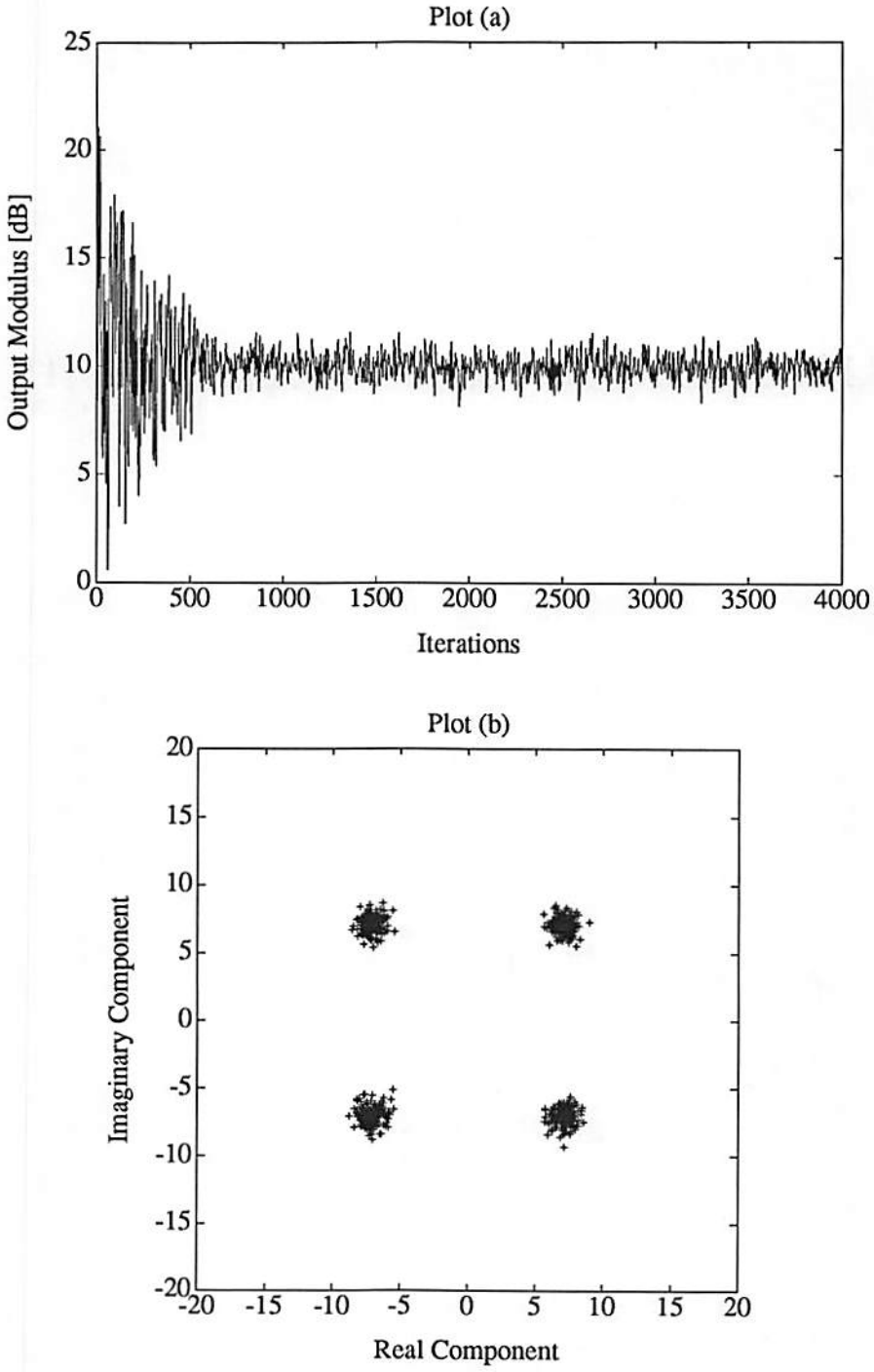

Figure 6.6: Plot (a): Output modulus $|y(n)|$ of the LCCM array as a function of adaptive iteration for δ= 20 dB and no multipath. Plot(b): Converged Constellation.

QPSK constellation for the 20 dB case. Note that the CM interfering signals have been removed and only the residual noise is left at the symbol points. The converged constellation for the 15 dB case is virtually identical to Figure 6.6b. The reason for this is that 15 dB is significantly less than the SOI power of 20 dB and equation (5.50) is satisfied. As explained in Section 3.3, the output power cannot be decreased by reducing the gain on the SOI because this gain is constrained. Nor can the power be decreased by appropriately adding interfering signals because these signals are uncorrelated with the SOI. Instead the squared output modulus $|y(n)|^2$ satisfies

$$|y(n)|^2 > \delta \tag{6.2}$$

for all[2] iterations and $\mu(n)$ therefore does not change sign. When the sign of $\mu(n)$ is positive the LCCM recursion is identical to the LCMP recursion and the LCCM weight vector w converges to the minimum power solution (2.16.) This is the desired solution for an uncorrelated interference scenario. Evidently, underestimating δ has no deleterious effect on the performance of LCCM if the interference is uncorrelated.

On the other hand, overestimating the value of δ has a significant effect on quality of the converged constellation. This is seen in the converged output constellation for the 25 dB case (Fig. 6.7.) Since the gain on the SOI is constrained and therefore cannot be increased, the LCCM algorithm increases the output power in order to match the 25 dB value of δ. It does this by reducing the null depth on one of the CM interferers. The presence of this CM interferer can be seen as a rotating phasor on each QPSK point in Figure 6.7. Overestimating the

[2]The random noise may occasionally bring the instantaneous squared modulus below δ but the effect is negligible.

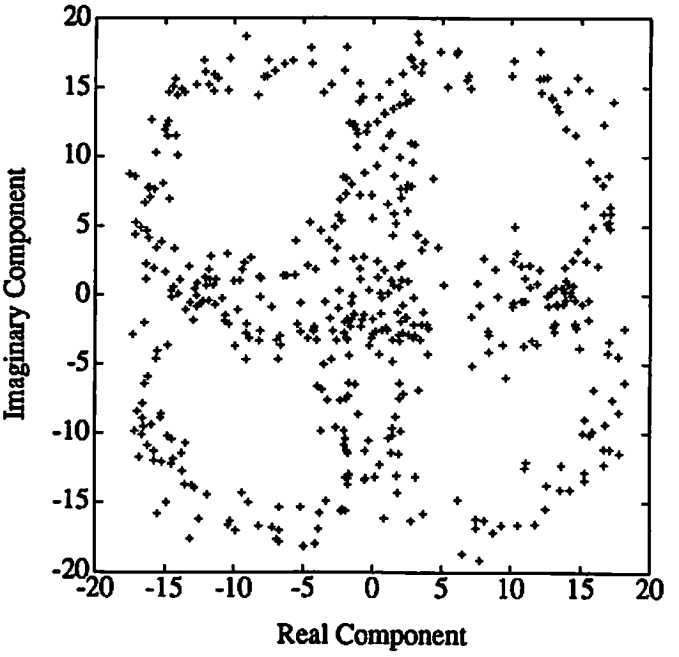

Figure 6.7: Converged QPSK signal constellation for LCCM array simulation with δ=25 dB and no multipath.

value of δ degrades performance for this particular simulation and is probably a bad practice in general.

For the same signal environment, the modulus factor is now made adaptive by letting $\alpha_\delta = 0.25$. Figure 6.8a shows that the output modulus converges in approximately 1000 samples. Comparison with Figure 6.6a shows that there is very little difference in the convergence of the output modulus between the $\delta = 20$ dB and adaptive δ cases. Both converge in ≈ 1000 samples. In addition, the converged constellation for adaptive δ (Fig. 6.8b) is virtually identical to the constellation when $\delta = 20$ dB (Fig. 6.6b.)

A consideration with the adaptive δ LCCM algorithm is how to select the initial value of δ. The effect of the this initial value is shown in Figure 6.9 where the adaptive modulus factor is plotted as a function of iteration for two initial conditions. It makes no difference if the initial δ estimate is greater or less than the optimal δ, both converge to the optimum value near 20 dB and produce the constellation shown in Figure 6.8b.

Correlated Interference: Multipath Present

The last experiment is now repeated with a correlated signal present. The signal environment for this simulation is the same as that given in Table 6.1 except that the white noise is at 0 dB. With the array simply steered to the SOI and no adaptation, the QPSK constellation is completely masked by interference and multipath distortion (Fig. 6.2.)

As before, three trials are run with δ fixed to 15, 20, and 25 dB. When multipath is absent the performance of LCCM for 15 and 20 dB cases is essentially the

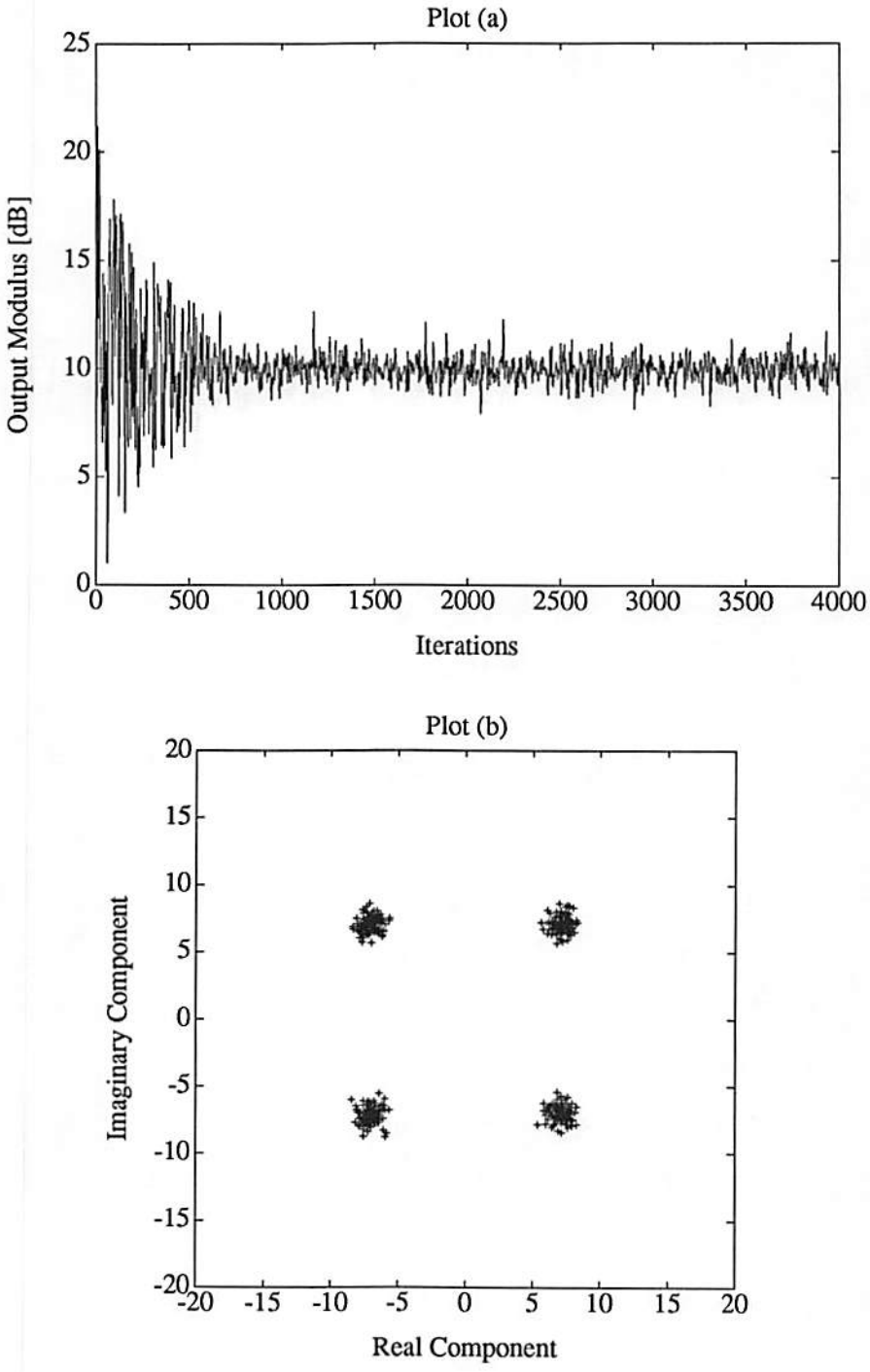

Figure 6.8: Plot (a): Output modulus $|y(n)|$ of the LCCM array with adaptive modulus factor as a function of adaptive iteration, no multipath. Plot (b): Converged Constellation.

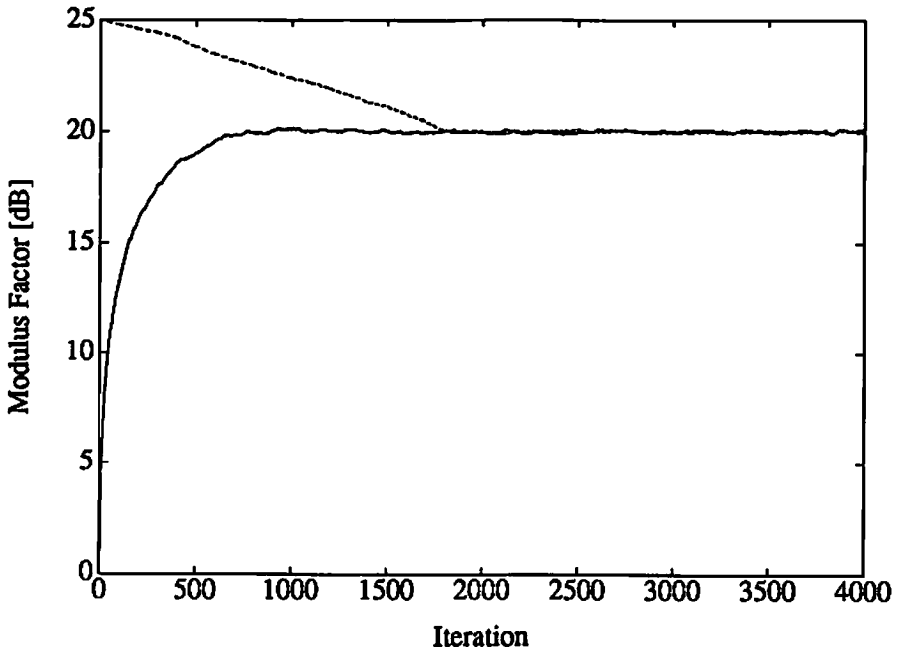

Figure 6.9: Adaptive modulus factor δ as a function of iteration for two initial conditions: $\delta(0)$=0 dB and $\delta(0)$=25 dB. No multipath present.

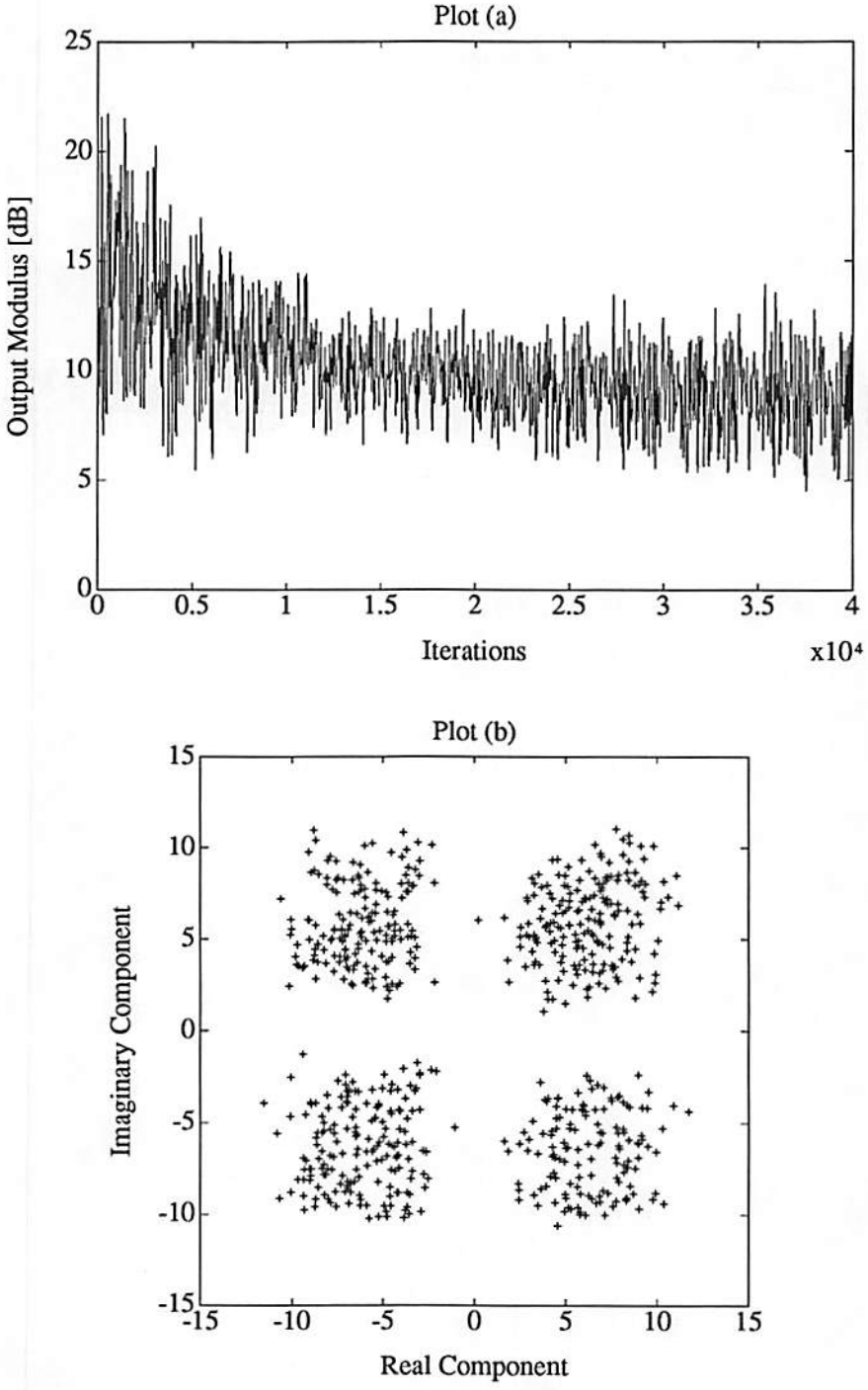

Figure 6.10: Plot (a): Output modulus $|y(n)|$ of the LCCM array as a function of adaptive iteration, δ=15 dB, multipath present. Plot (b): Converged Constellation.

same. When multipath is present, however, the situation is much different. Figure 6.10a shows the output modulus for the 15 dB trial as a function of iteration. Note that the variance decreases at first but then it increases again for the last 10,000 iterations. It appears at first that the output is diverging. However, as the variance increases, the mean actually decreases. That is, $Var\{|y|^2\}$ increases but $E\{|y|^2\}$ decreases, by about 1 dB over the last 20,000 iterations. This is precisely the behavior that was predicted in Section 3.3. Error in δ has a higher penalty in the LCCM cost function than variations in the complex envelope. To correct the error in δ, the LCCM weights reduce the output power by adding correlated interference destructively. This is evident in the converged spatial response of Figure 6.11. Note that the gain on the multipath signal at 30° is ≈ -6 dB. The phase is approximately $-180°$ which implies that the multipath is added to the SOI out of phase causing partial signal cancellation. Although this yields less cost, the constellation that results, Figure 6.10b, has much more variance than the case when $\delta = 20$ dB. Clearly, underestimating δ in correlated interference can degrade LCCM performance. (The constellation for the 20 dB case is not plotted but it is virtually identical to the one in Figure 6.14 where δ is adaptive.)

In the multipath-absent case, choosing a value of δ that is greater than the optimum leads to severe constellation distortion. The same is true for the multipath-present case as is can be seen in Figure 6.12 which is the constellation that results when $\delta = 25$ dB. Although both are poor, the constellation for the multipath-present case actually has a structure that bears more resemblance to QPSK. The reason for this is that in the correlated signal case, the multipath is added to the SOI to increase the total output power whereas in the uncorrelated case the

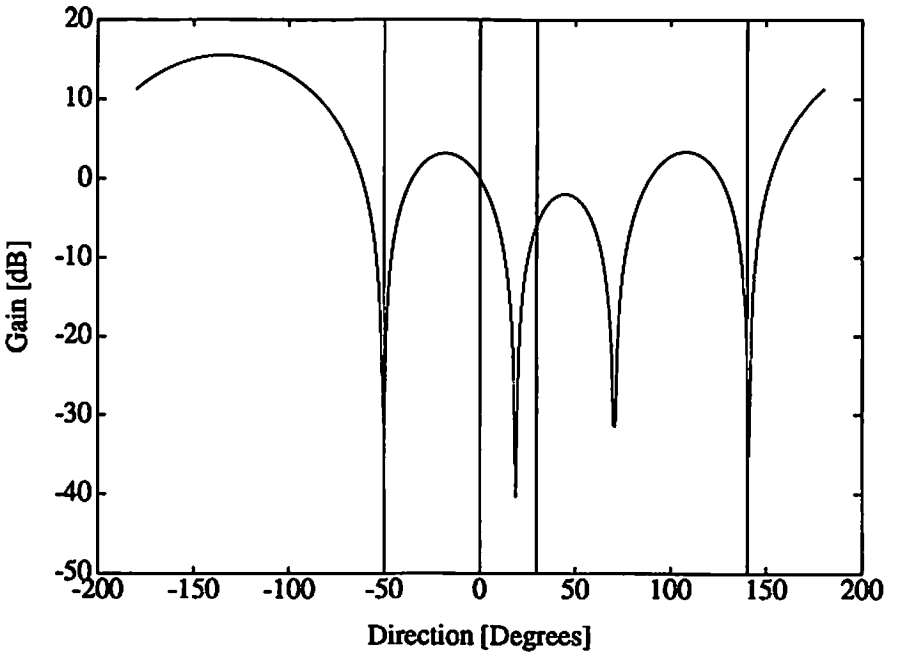

Figure 6.11: Converged spatial response of LCCM array for δ=15 dB. Multipath present.

CM interferer was added to the SOI. The correlation between the SOI and the multipath yields a somewhat better constellation. In any case, overestimating δ leads to distortion whether the interference is correlated or not.

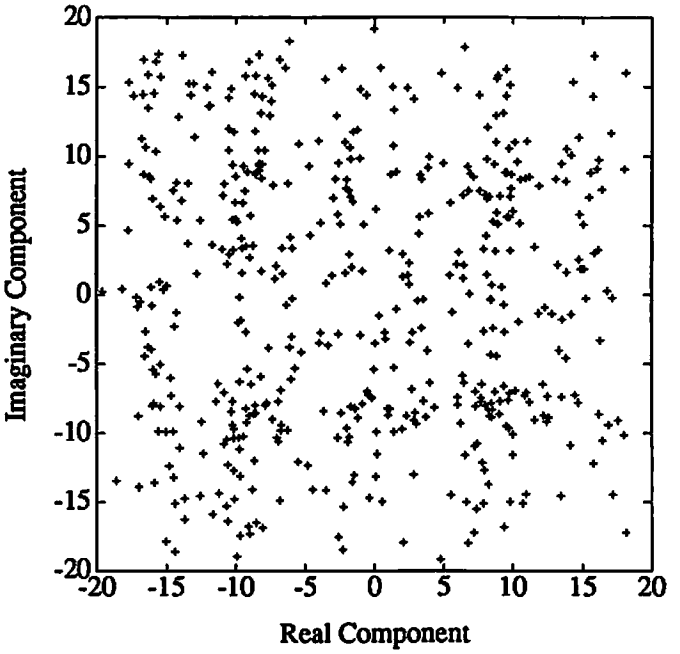

Figure 6.12: Converged signal constellation for LCCM array simulation with δ= 25 dB, multipath signal present.

Figures 6.13a and 6.13b show the output modulus plots for the 20 dB and adaptive δ cases respectively. The 20 dB simulation converges in approximately 15,000 iterations and the adaptive δ simulation converges in about 20,000 iterations. At convergence, both simulations have

$$\sigma^2_{|y|} \triangleq Var\{|y|\} = 0.34. \tag{6.3}$$

Their corresponding constellations are virtually identical. For brevity, only the constellation corresponding to the adaptive case is plotted (Fig. 6.14.)

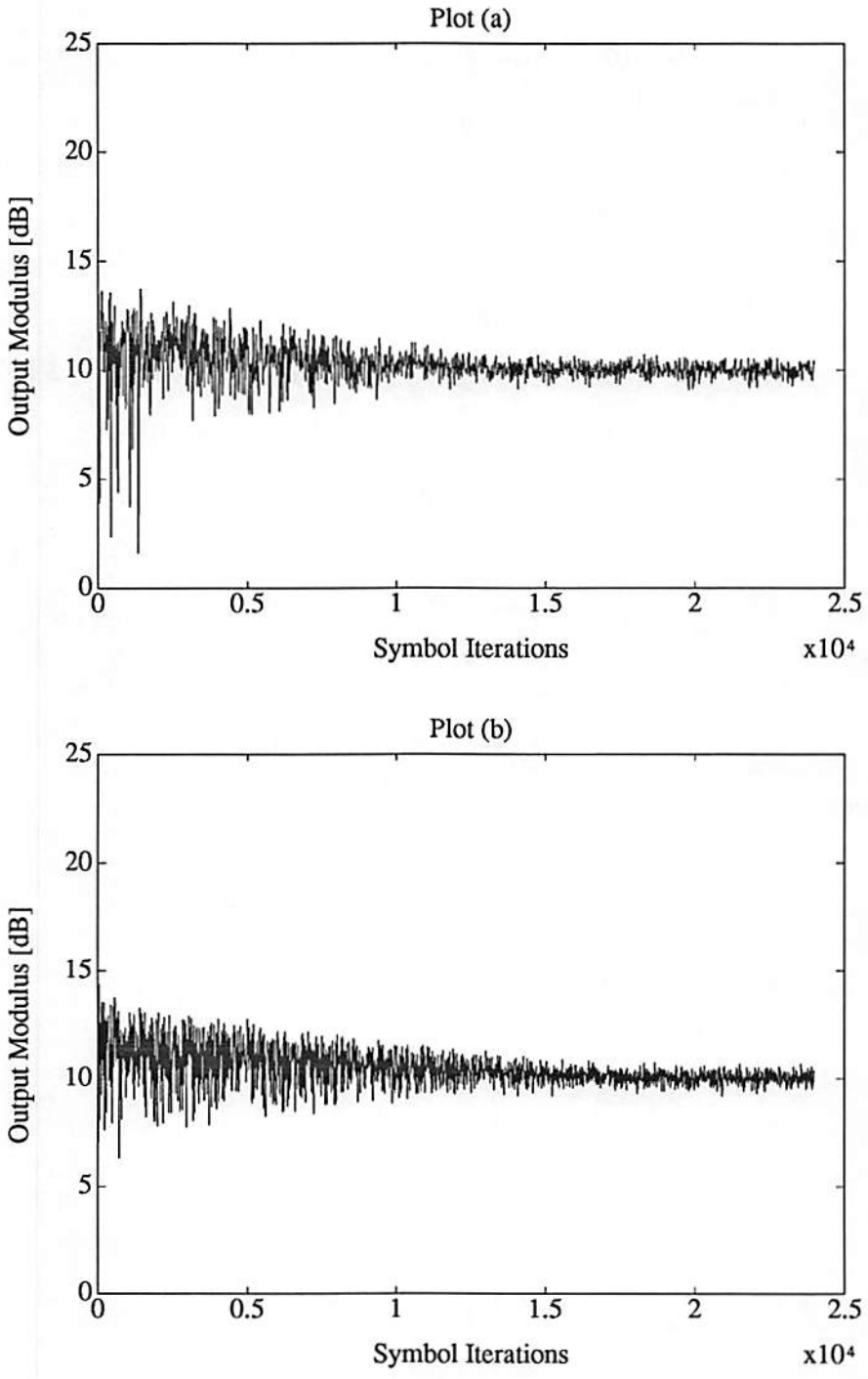

Figure 6.13: Output modulus $|y(n)|$ for the LCCM array as a function of adaptive iteration with multipath present. Plot (a): δ=20 dB. Plot (b): adaptive δ.

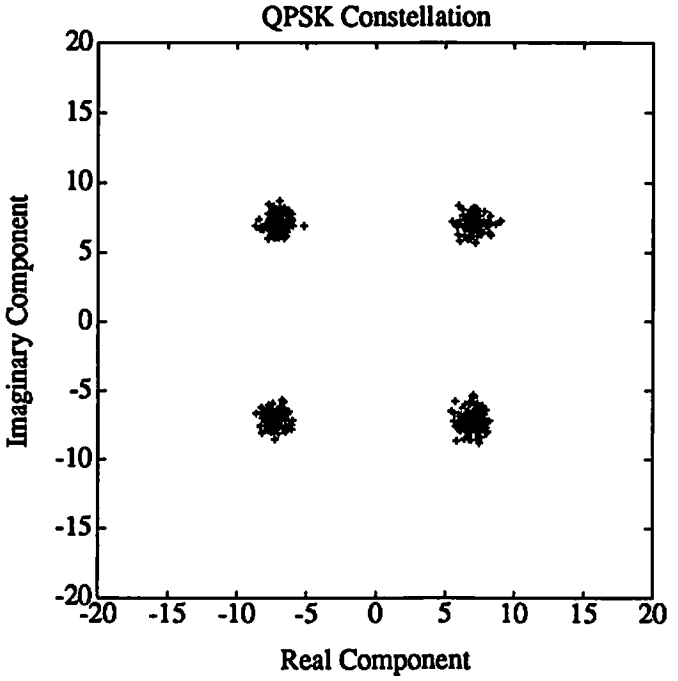

Figure 6.14: Converged QPSK signal constellation for LCCM array simulation with adaptive δ, multipath signal present.

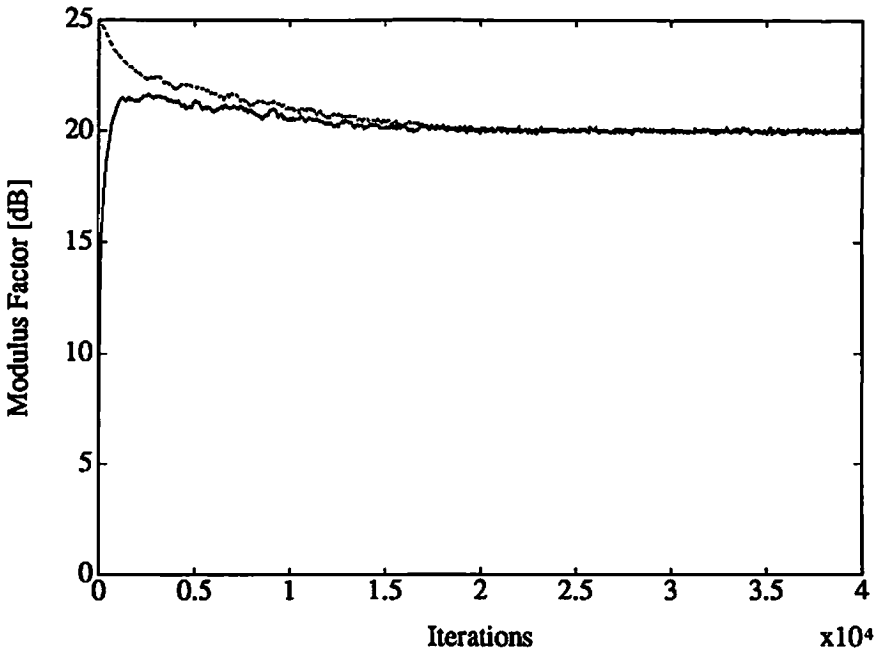

Figure 6.15: Adaptive modulus factor δ as a function of iteration for two initial conditions: $\delta(0)$=0 dB and $\delta(0)$=25 dB. Multipath signal present.

The adaptive case takes longer to converge because the modulus factor needs many iterations to converge. Figure 6.15 shows the convergence of the adaptive δ for both low and high initial conditions.

In both cases δ takes about 20,000 iterations to converge. Since the output modulus converges in about the same time the array weights and δ must be converging simultaneously. Better initial estimates of δ will of course lead to faster convergence. It is important to note the difference in convergence rate between the correlated and uncorrelated simulations. For the same values of $\alpha_{\mathbf{w}} = 0.025$ and $\alpha_{\delta} = 0.25$, convergence takes an order of magnitude more iterations for the case when multipath is present.

The general conclusion from the previous array simulations is simple: the adaptive δ version of LCCM is an effective technique for dealing with multipath and co-channel interference. The results indicate that, other than convergence rate, there is no performance loss for using an adaptive modulus factor. In many cases, the convergence rate can be improved by using a good initial estimate for the value of δ.

So far, this dissertation has dealt only with the adaptive array application of the LCCM method. Chapter 4 analyzed LCCM for narrowband array scenarios and in this section the LCCM adaptive array was simulated. The next section describes a single channel application of LCCM: fractionally spaced equalization.

6.3 A Fractionally Spaced Equalizer

Fractionally spaced equalization [40, 41] is a popular technique that was evidently first proposed in [42] for telephone channel compensation. The equalization scenario that is simulated in this section is drawn in part from a recent paper on digital radio reception [43]. This paper deals with the specification and design of a general purpose equalizer-demodulator for digital radio signals. The fractionally-spaced equalizer used in this design simultaneously performs adjacent and co-channel interference rejection, matched filtering, and ISI compensation. It accomplishes these tasks using a multi-mode, baseband equalizer that acquires the SOI with UCM and then switches to Decision Feedback (DF) when the error rate is low enough to reduce the effects of error propagation.

A significant drawback to acquisition of the SOI with UCM is that, as happened with the adaptive array in Section 6.2, UCM may acquire a interfering signal

rather than the SOI. This problem is specifically addressed in [24] for the adaptive receiver described in [43]. The approach taken in [24] is to initialize the equalizer taps such that capture by an interfering signal is precluded. Proper selection of initial conditions "immunizes" UCM from the bad local minima of the cost function. This approach may breakdown, however, in non-stationary environments because the cost function and the local minima are rapidly time-varying.

An alternative to initializing the equalizer tap weights is to make them satisfy a set of linear constraints. With an appropriate linear constraint equation, the problem of interference capture can potentially be eliminated. This was shown analytically in Chapter 4 and simulated for an array application in Section 6.2. In this section, the LCCM approach is applied to a single channel problem. The first step is to find a suitable linear constraint using the signal subspace method.

6.3.1 Signal Subspace-Based Equalization

It is important to distinguish between the *signal space* concept from communication theory [44] and the *signal subspace* concept from signal processing [27]. The two concepts are related but there are subtle differences which need clarification here. To review, the signal space is the space spanned by a set of P continuous time orthonormal functions $\{\phi_k(t)\}$. These functions are basis functions for a set of Q transmitted waveforms $\{s_i(t)\}$, i.e. each of the waveforms $\{s_i(t)\}$ is completely determined by a vector of coefficients

$$\mathbf{s}_i = \begin{pmatrix} s_{i1} & s_{i2} & \cdots & s_{iP} \end{pmatrix} \tag{6.4}$$

where

$$s_{ik} = \int_{-\infty}^{\infty} s_i(t)\phi_k(t)dt. \tag{6.5}$$

Many practical modulation schemes can be modeled as having only one basis function in baseband, the unit pulse

$$\phi(t) = \begin{cases} \sqrt{\frac{1}{T}} & 0 < t \leq T \\ 0 & \text{elsewhere.} \end{cases} \tag{6.6}$$

For example, a BPSK signal has two real coefficients $s_1 = 1$ and $s_2 = -1$. Two dimensional modulation such as QPSK is determined by four complex coefficients,

$$\begin{aligned} s_1 &= 1 + j \\ s_2 &= 1 - j \\ s_3 &= -1 + j \\ s_4 &= -1 - j. \end{aligned}$$

In practice, though, a QPSK signal is transmitted in I-Q form using two orthonormal basis functions. Nevertheless, it can be modeled as having a real pulse basis function and complex modulation. For the purposes of this study the signal space is one dimensional and consists of the pulse $\phi(t)$.

In a sense, the signal *subspace* is a discrete version of the signal space. The signal subspace is simply the vector of N samples of the continuous time function $\phi(t)$. This vector together with the complex modulation form a one dimensional complex subspace; any complex multiple of the sample vector is still a member of the subspace. How this relates to the case at hand is best shown by example.

Consider a baseband BPSK signal $x(t)$ that uses the unit pulse as its basis function. The signal is sampled and a data vector $\mathbf{x}(n)$ is formed

$$\mathbf{x}(n) = \begin{pmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{pmatrix}. \tag{6.7}$$

The significant eigenvectors of $\mathbf{R}_{xx}$ span the signal subspace of the BPSK signal for this particular sampling strategy[3]. All other things equal, the rank of the subspace is proportional to the bandwith of the transmitted signal and is generally greater than one. There is a special case, however, where the rank of the subspace is one. This happens when the sampling strategy meets the following conditions: the N samples are baud synchronous and are all taken from the same pulse interval. The first data vector only contains samples from the first pulse, the second data vector only contains samples from the second pulse, and so on. For example, let the BPSK signal be sampled at multiples of $T/2$ where T is the symbol period. If the sampling is synchronous, then the two samples from the same baud form a vector that has only two values: $\mathbf{x} = (\ 1 \quad 1\ )$ or $\mathbf{x} = (\ -1 \quad -1\ )$. The corresponding subspace has a rank of one. If the sampling crosses the baud boundary then the corresponding subspace will have a rank of two because the data vector can now take on the additional values $\mathbf{x} = (\ -1 \quad 1\ )$ or $\mathbf{x} = (\ 1 \quad -1\ )$. This notion generalizes to any system with a pulse as its fundamental baseband unit and is independent of the number of samples per symbol interval.

In practical communication systems, the received pulse does not resemble the unit pulse at all. The energy of the pulse stretches across the baud boundary due to intentional effects such as pulse shaping and unintentional effects such as channel distortion. The subspace that corresponds to this signal, even if it is created using the sampling strategy described previously, will not be rank one. However, the subspace can be reduced to one by redefining what is signal and what is interference. In the ideal case the signal consisted of all the pulses in

[3]Actually, a BPSK signal is cyclo-stationary with period T so the autocorrelation matrix $\mathbf{R}_{xx}$ is really an average of the periodic statistics over T.

the BPSK signal. For the case at hand where the pulse is smeared across baud boundaries it is assumed that only the pulse in the center of the sample window is *signal*. The parts of other pulses that leak into this sampling window are considered *interference*. With baud synchronous sampling, this assumption leads to a signal subspace of one.

In order to define the constraint used in LCCM equalization, it is necessary to examine the position of the LCCM equalizer in a receiver block diagram, Figure 6.16. The input to the equalizer is the $T/2$ sampled output of the matched filter. Ideally, the pulse at the output of the matched filter has the $T/2$ sampled Nyquist form of Figure 6.17. At all intervals of T other than the pulse center, the sampled Nyquist pulse is zero which implies that the ISI is zero. The vector of samples of the Nyquist pulse is the signal subspace of the received signal under ideal conditions and assuming that the other pulses are interference. This vector is the linear constraint used in the LCCM equalizer. It may seem inappropriate that the ideal pulse is used as the linear constraint for an equalizer that will operate under highly non-ideal conditions but the constraint vector represents the only *a priori* knowledge available. The matched filter accounts for any known characteristics of the channel; the compensation for unmodeled channel distortion, multipath and interference is the job of the LCCM equalizer and the decoder that follows

At this point it is helpful to review why a linear constraint is necessary in the first place. The UCM algorithm can be used to blindly adapt the fractionally spaced equalizer taps but it is susceptible to capture by CM interference. Imposing a linear constraint on the taps has the potential to eliminate this drawback.

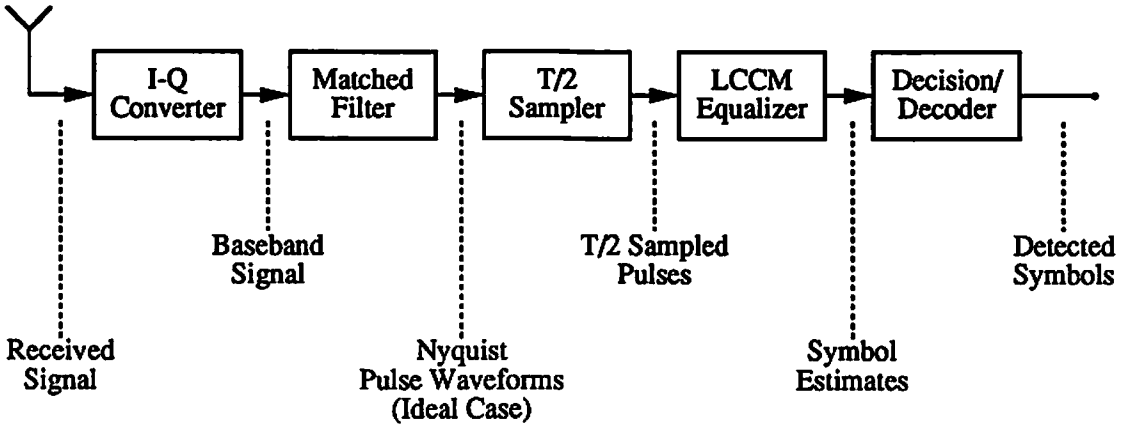

Figure 6.16: Block diagram of a receiver that includes the LCCM fractionally spaced equalizer.

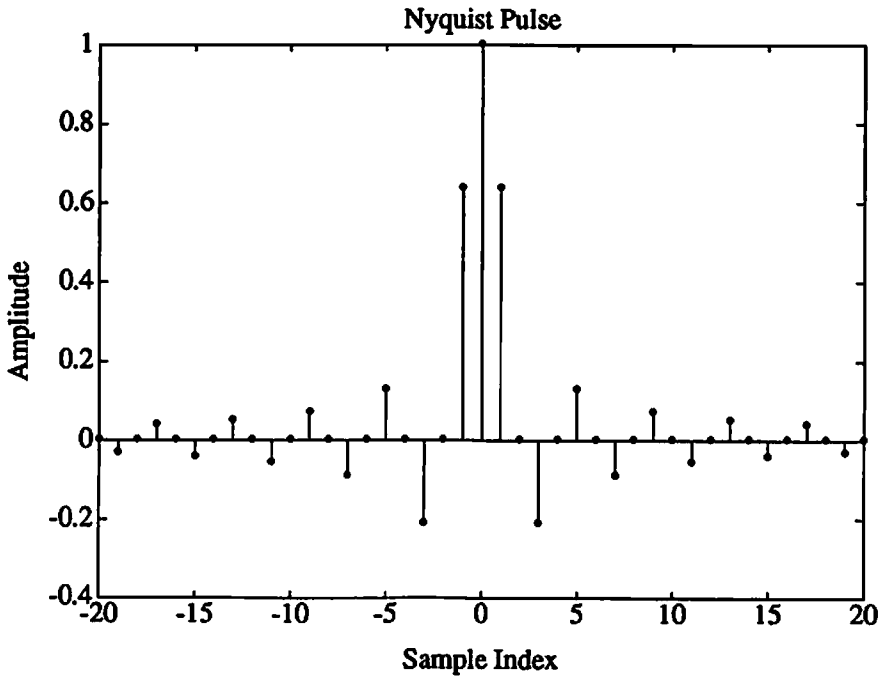

Figure 6.17: The Nyquist pulse. Used as the constraint vector for the LCCM adaptive equalizer.

In the absence of additional channel knowledge, the Nyquist pulse is the most appropriate linear constraint.

Like any implementation of the LCCM algorithm, the LCCM equalizer uses the GSC filter structure, Figure 5.1, to implement the constraint. The correspondence between the elements of the GSC structure and the LCCM equalizer is very simple. The elements of the input signal vector $\mathbf{x}(n)$ are the matched filter outputs sampled at $T/2$. The vector $\mathbf{w}_q$ is as defined in (5.4) with $\mathbf{C}$ equal to the pulse replica and $\mathbf{f}$ equal to unity. (Recall that $\mathbf{C}$ is a vector and $\mathbf{f}$ is a scalar in the single constraint case.) The output $y(n)$ is the equalizer output which is updated along with the weights at intervals of T.

6.3.2 Intersymbol Interference Correction

Now that a constraint has been found a good first test of the LCCM equalizer is its operation under ideal conditions: when the pulse at the output of the matched filter has perfect Nyquist shape. Actually, under these conditions no equalization is necessary because there is no ISI. The optimum equalizer under these conditions has a center tap of unity and all others equal to zero. These are not the initial conditions of the LCCM equalizer, however. At start-up, the vector $\mathbf{w}_a$ in the LCCM adaptive recursion is the zero vector. With $\mathbf{w}_a = \mathbf{0}$, the equalizer tap weights are equal to $\mathbf{w}_q$ (5.4), a scaled copy of the constraint vector[4]. Although it is the ideal pulse at the output of the matched filter, $\mathbf{w}_q$ is not the optimum equalizer. The vector $\mathbf{w}_q$ actually causes ISI where none existed. Figure 6.18

[4]The actual Nyquist pulse used as the constraint in this simulation decays more rapidly than the one shown in Figure 6.17.

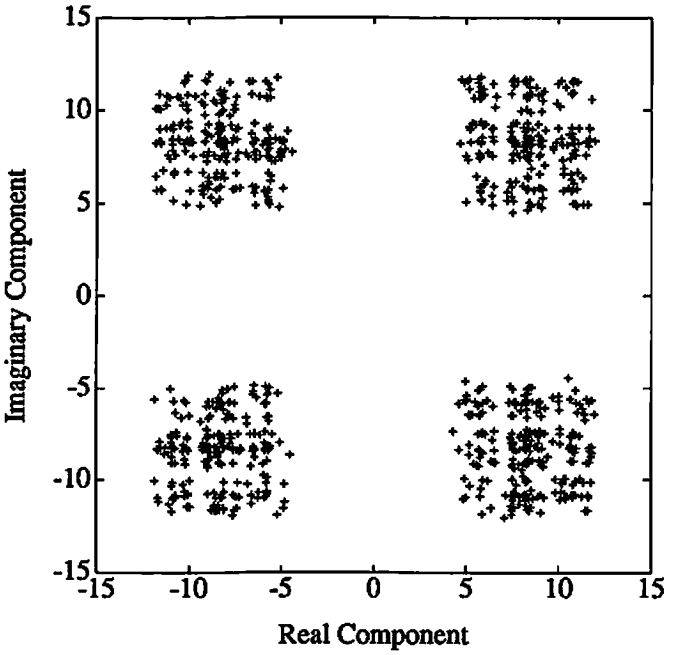

Figure 6.18: Start-up constellation for the LCCM equalizer. No co-channel interference, no multipath.

shows this artificially induced ISI for the case of a 7 tap equalizer operating on a 20 dB QPSK signal in -10 dB white noise.

Next, the LCCM equalizer is allowed to adapt under these same conditions with its operating parameters given by Table 6.3. Figure 6.19 shows that the adaptive modulus factor $\delta(n)$ converges in about 600 symbols. Convergence of the output modulus follows shortly afterward as shown by Figure 6.20. The clean QPSK constellation that results, Figure 6.21, reveals that the artificially induced ISI is removed adaptively. So, the LCCM equalizer operating under ideal conditions gives satisfactory results. Next it is tested in non-ideal signal environments that include co-channel interference and multipath.

LCCM Parameter	Value
$\alpha_{\mathbf{w}}$: Convergence Parameter for w	0.025
α_{δ} : Convergence Parameter for δ	0.25
γ : Tracking Factor for σ^2_{xa}	0.99
$\delta(0)$: Initial Value of Modulus Factor	0 dB

Table 6.3: Common LCCM Operating Parameters for Equalizer Simulations.

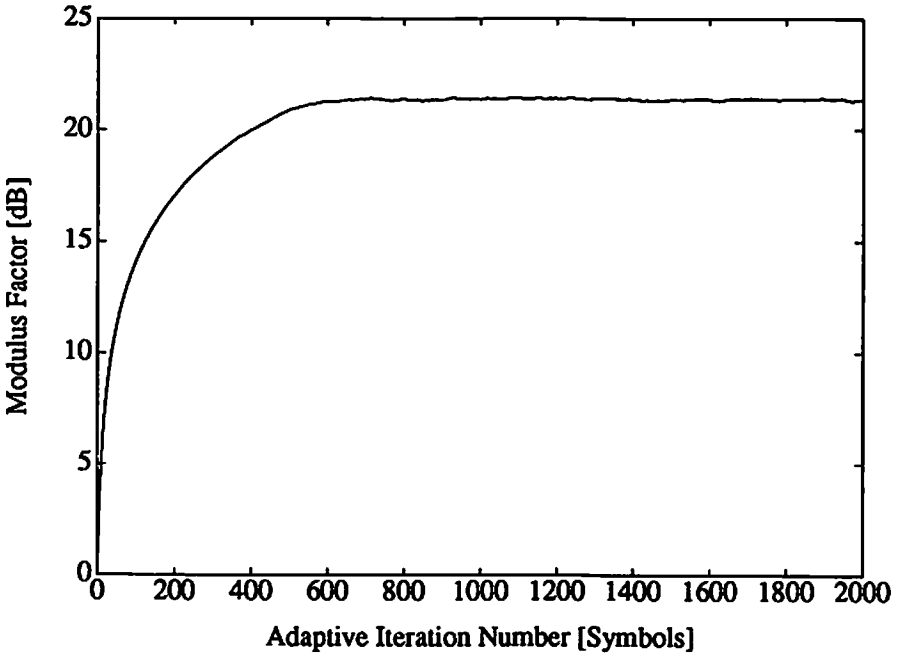

Figure 6.19: Adaptive modulus factor $\delta(n)$ of the LCCM equalizer as a function of iteration. No co-channel interference, no multipath.

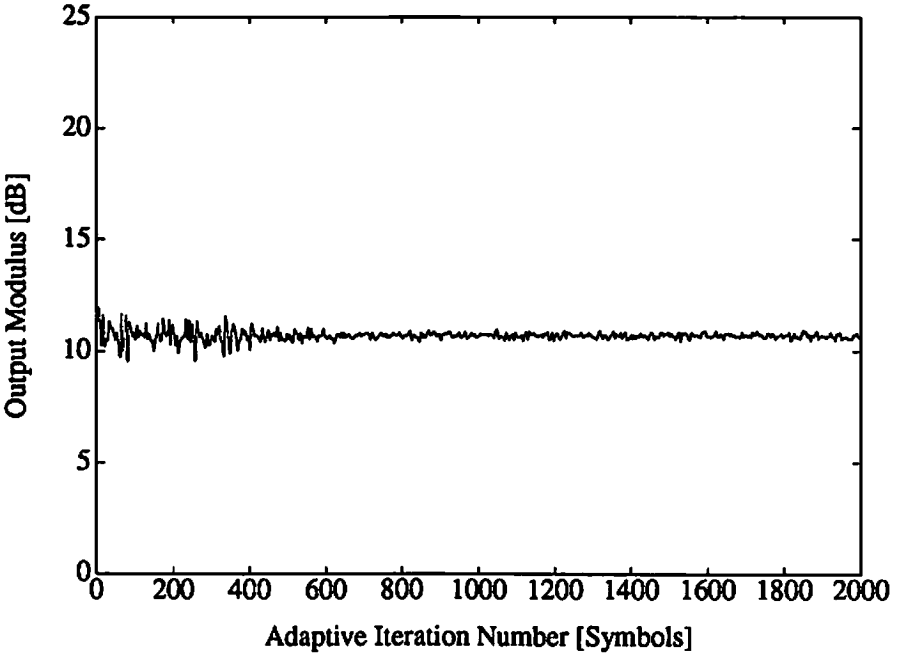

Figure 6.20: Output modulus $|y(n)|$ of the LCCM equalizer as a function of adaptive iteration. No co-channel interference, no multipath.

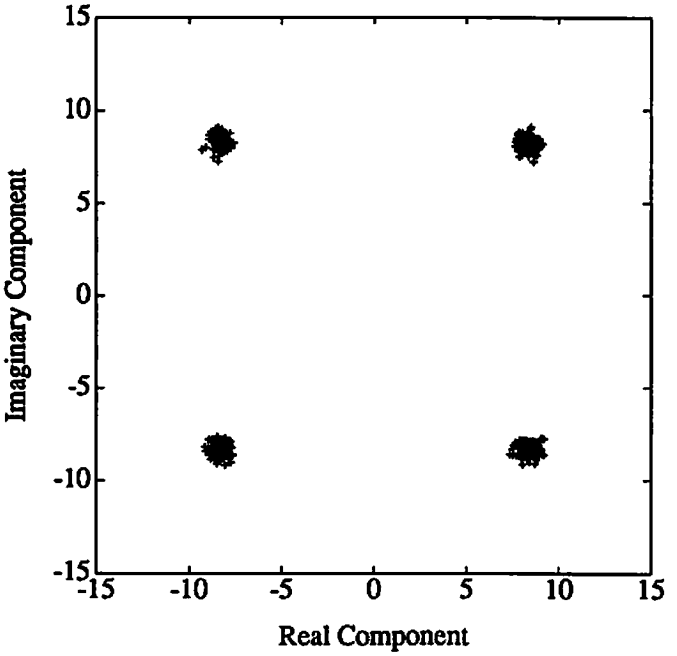

Figure 6.21: Converged output constellation of the LCCM equalizer. No co-channel interference, no multipath.

6.3.3 Co-Channel Interference & Multipath Correction

Table 6.4 describes the signal environment for an experiment that includes co-channel interference. The only difference between this simulation and the previous one is that this one includes a 30 dB co-channel CM interferer and the number of equalizer taps is increased to 9. Figure 6.22 shows that the adaptive modulus factor $\delta(n)$ converges in approximately 1000 symbols and Figure 6.23 indicates that the output modulus converges in about 1500 samples. The resultant

SOI Power	20 dB
Noise Power	-10 dB
Interference Power	30 dB
Interference Normalized Frequency	0.312

Table 6.4: Signal Environment for Figures 6.23 and 6.22.

constellation is virtually identical to that shown in Figure 6.21 which implies that the equalizer appropriately filters out the in-band CM interference. More importantly, LCCM is not captured by the strong constant modulus interferer as is possible with UCM.

Table 6.5 describes the signal environment for an experiment that includes multipath interference. In this case, the number of equalizer taps is increased to 45. A large number of taps is necessary because the equalizer needs many degrees of freedom to compensate for the multipath degradation. (In a sense, it uses an FIR equalizer to approximate an IIR response.) The effect of the number of taps on the equalizer performance will be investigated in later simulations.

The multipath used in this experiment is the QPSK SOI delayed by 3 samples ($3T/2$), phase shifted by 180 degrees, and attenuated by 3 dB. The sum of the

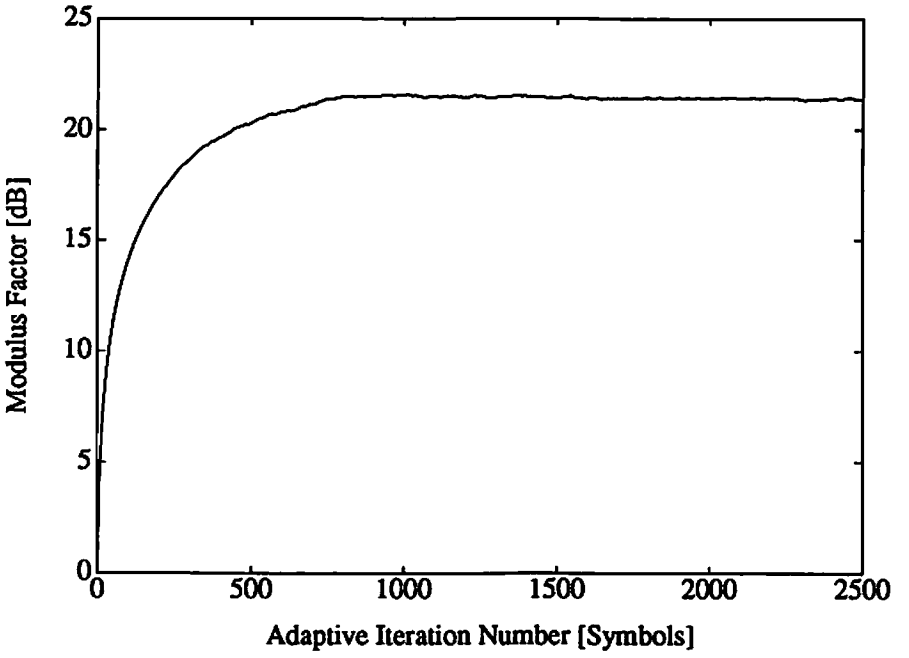

Figure 6.22: Adaptive modulus factor $\delta(n)$ of the LCCM equalizer as a function of iteration. Co-channel interference present, multipath absent.

SOI Power	20 dB
Noise Power	-10 dB
Multipath Delay	$3T/2$
Multipath Attenuation	-3 dB
Multipath Phase	180°

Table 6.5: Signal Environment for Figure 6.24.

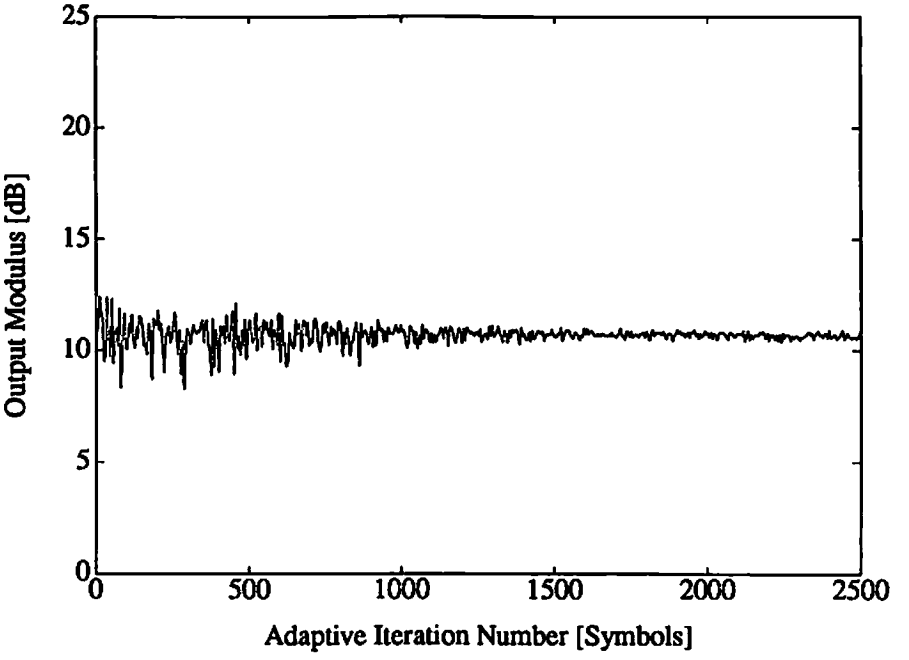

Figure 6.23: Output modulus $|y(n)|$ of the LCCM equalizer as a function of iteration. Co-channel interference present, multipath absent.

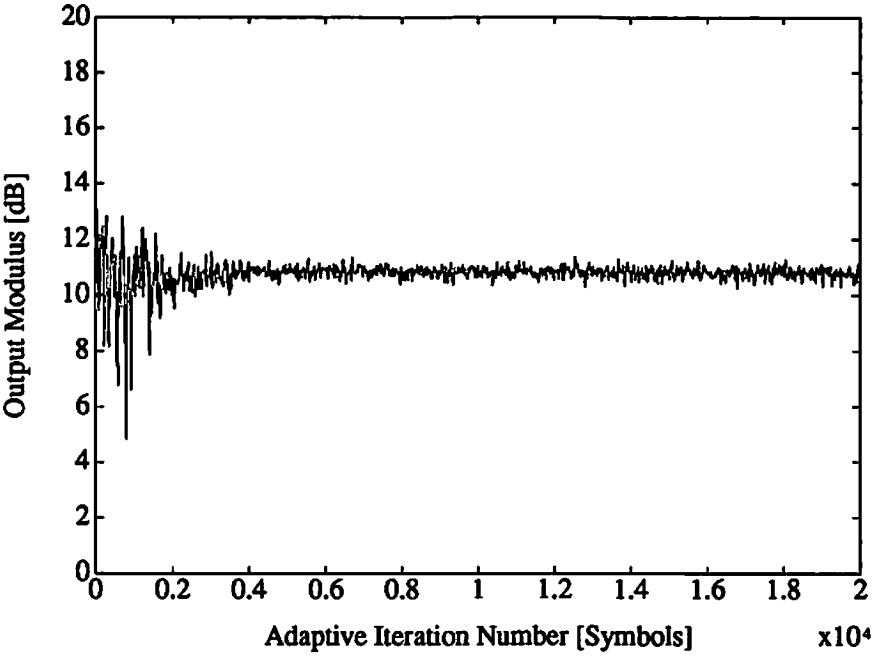

Figure 6.24: Output modulus $|y(n)|$ of the LCCM equalizer as a function of adaptive iteration. Multipath present, co-channel interference absent.

SOI and the multipath results in significant distortion of the constellation. Figure 6.24 shows that the output modulus converges in approximately 5000 symbol iterations to $Var\{|y|\} = 0.20$. Although the converged variance can be decreased slightly with additional equalizer taps, the more significant factors are the LCCM convergence parameters $\alpha_{\mathbf{w}}$ and α_{δ}. Reducing these parameters slows convergence but also decreases misadjustment which in turn decreases the variance of $|y|$ at convergence. The variance can also be decreased by freezing the weight vector at some point after convergence. This eliminates any variance due to misadjustment but unwise in non-stationary environments. The effects of both the number of taps and the value of $\alpha_{\mathbf{w}}$ are examined in the following simulation.

The next simulations include both co-channel interference and multipath. The signal environment parameters are given in Table 6.6. Before examining the adaptive results, it is interesting to check the constellation that results if no equalization is done. Figure 6.25 shows the constellation of matched filter outputs. The combined effects of co-channel interference and multipath completely mask the QPSK signal. This is not the start-up state for the LCCM equalizer it is the start-up state that a T-spaced DF or UCM equalizer would have. The initial error rate for this constellation is too high for the DF equalizer to bootstrap itself, and runaway error propagation would result. At the same time, the strength of the CM co-channel interferer is strong enough that the UCM equalizer would often experience capture resulting in the loss of the SOI. This is the type of signal environment that motivated the development of LCCM.

SOI Power	20 dB
Noise Power	-10 dB
Interference Power	30 dB
Interference Normalized Frequency	0.312
Multipath Delay	$3T/2$
Multipath Attenuation	-3 dB

Table 6.6: Signal Environment for Figures 6.26, 6.27, and 6.28.

Figure 6.26 shows the converged constellation of a 45 tap LCCM equalizer for the signal environment of Table 6.6 and the parameter set of Table 6.3. Clearly the 45 tap equalizer successfully compensated for distortion that is present. As mentioned previously, a typical role of LCCM is as an untrained start-up equalizer for the DF equalizer. The LCCM equalizer improves the constellation to a point where adaptation can be switched to a DF mode for higher performance. A DF

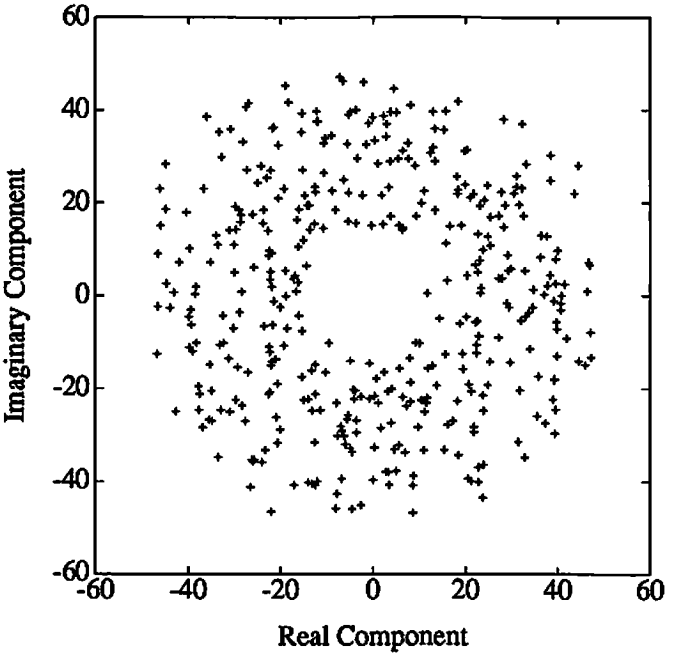

Figure 6.25: Constellation of matched filter outputs (slicing instant.) Both co-channel interference and multipath present.

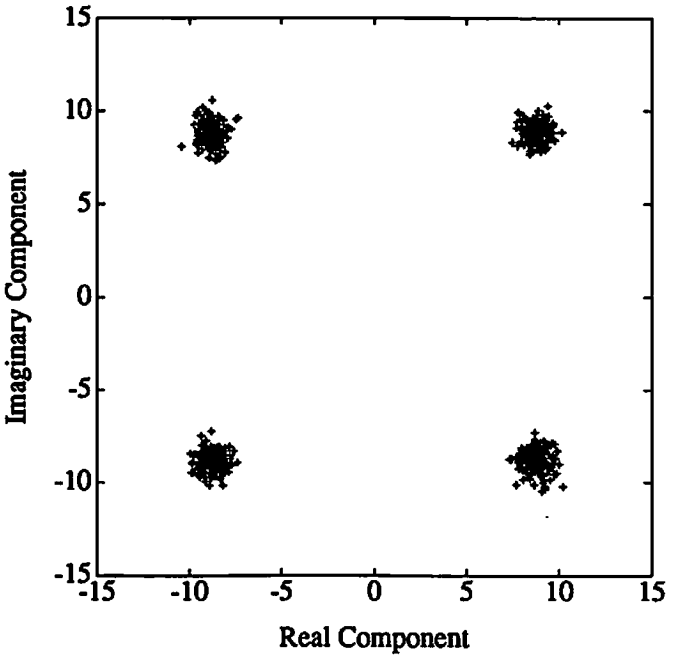

Figure 6.26: Converged constellation for a 45 tap LCCM equalizer. Both co-channel interference and multipath present.

equalizer would have no trouble adapting on the constellation of Figure 6.26. In fact, it is much cleaner than necessary and additional distortion could be tolerated without penalty. It is desirable to trade-off some of the constellation quality for a faster convergence rate.

Figure 6.27 shows the convergence of the output modulus that produced the constellation in Figure 6.26. Convergence occurs in approximately 25,000 symbol iterations. The convergence rate can be increased in three ways: by increasing $\alpha_{\mathbf{w}}$ and α_{δ}, by using fewer taps or by using a better initial estimate of the modulus factor. The first two methods trade off constellation quality for more rapid convergence rate. Figure 6.28 shows the output modulus as a function of iteration for a 25 tap LCCM equalizer with $\alpha_{\mathbf{w}} = 0.05$. Convergence occurs in approximately

5,000 symbols which is 1/5 the number required for the 45 tap, $\alpha_{\mathbf{w}} = 0.025$ case. Other combinations of $\alpha_{\mathbf{w}}$ and N are summarized in Table 6.7. Clearly, the convergence rate can be improved without a great increase in the variance of the constellation points.

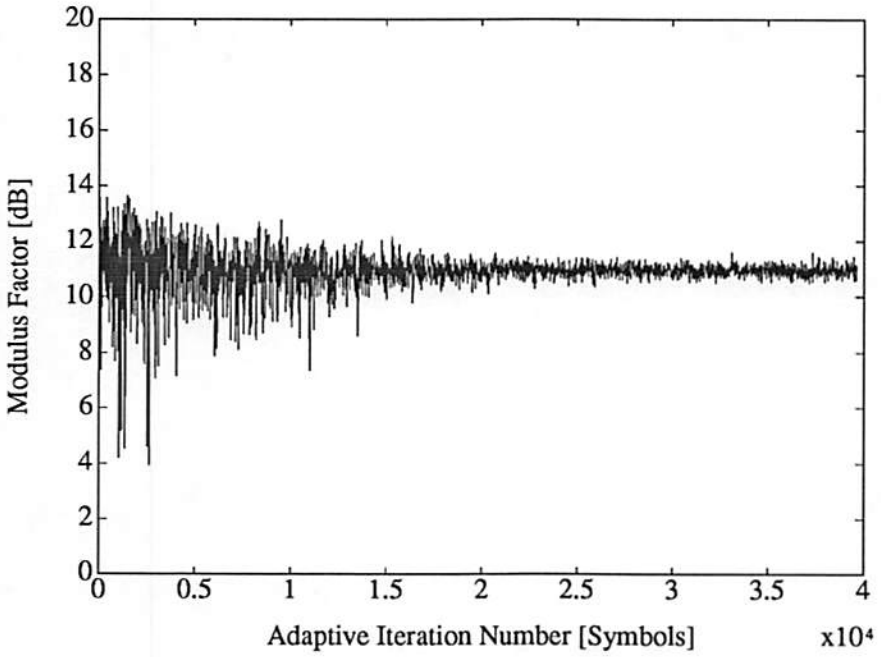

Figure 6.27: Output modulus $|y(n)|$ of the 45 weight LCCM equalizer as a function of iteration. Both co-channel interference and multipath present. $\alpha_{\mathbf{w}} = 0.025$.

$\alpha_{\mathbf{w}}$	N	Var$\{\|y\|\}$	Convergence [Symbols]
0.025	45	0.23	25K
0.025	25	1.07	10K
0.05	45	0.40	15K
0.05	25	1.23	5K

Table 6.7: Simulation Parameters and Results for LCCM Equalizer

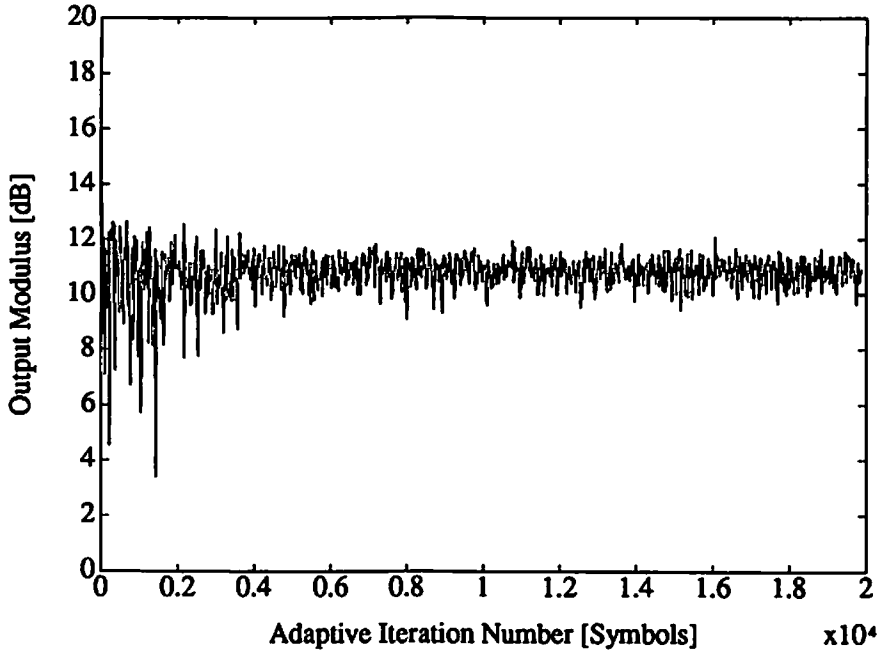

Figure 6.28: Output modulus $|y(n)|$ of the 25 weight LCCM equalizer as a function of iteration. Both co-channel interference and multipath present. $\alpha_{\mathbf{w}} = 0.05$.

6.4 Summary

The simulations of this chapter have demonstrated that the LCCM adaptive algorithm can be successfully applied to both single and multi-channel environments. For both the array and equalizer applications, LCCM removed co-channel interfering signals and compensated for any multipath distortion that was present. In no case was the LCCM algorithm captured by CM interfering signals as is possible with the UCM algorithm. In addition, a local minima problem for highly correlated signals was not observed. (Recall that the analytical results of Section 4.3 suggested that highly correlated interference may cause undesired local minima of LCCM.)

As predicted in Section 3.3, the modulus factor δ plays a fundamental role in the convergence of the adaptive weight vector and in the ultimate quality of the filtered signal. Both the fixed and adaptive δ versions of LCCM were implemented in the array application. The fixed δ LCCM implementation was adversely affected by error in the modulus factor estimate in all cases except one: when δ was underestimated in an uncorrelated interference environment. In the other cases the converged QPSK constellation was severely degraded. The adaptive δ LCCM implementation produced results that were nearly the same as those when δ was at its optimum value. The only disadvantage was that the adaptive δ version required more iterations for convergence.

In simulations of both the fixed and adaptive versions of LCCM, multipath interference significantly increased the number of iterations necessary for convergence. Typically, an order of magnitude more iterations were needed in both the array and equalizer applications. This same phenomenon is present in other SGD

algorithms such as LMS and is unavoidable. In fact, there are many similarities between the convergence of LCCM and LMS. Both SGD algorithms have faster convergence with fewer adaptive weights. Likewise, with both algorithms there is a trade-off between faster convergence and greater weight vector misadjustment for larger values of α.

As a practical matter, the choice of α and the choice between fixed and adaptive δ LCCM versions will be determined by the type of application. In an *acquisition* application, LCCM is a start-up processor that is later switched off in favor of a better but less robust algorithm such as decision feedback. It is not important that the signal distortion be optimally corrected by LCCM. What is important is that the signal quality is improved enough such that the follow-up processor can take over. Fast convergence is paramount which implies the use of fixed δ LCCM and a high value of $\alpha_{\mathbf{w}}$. In a *signal conditioning* application, LCCM is a pre-processor for some other filter that is designed for more benign signal environments. Since LCCM adapts continuously in this case, the adaptive δ implementation is more appropriate.

Chapter 7

Conclusion

The summaries at the end of each chapter give concise overviews of the results that are achieved in this dissertation. This final chapter is not intended to reiterate those results since each summary stands on its own for the most part. Instead, the purpose here is to put the results in a general framework that is then examined for strong and weak areas. In addition, future research topics are proposed.

7.1 Critical Review

The motivation for the research in this dissertation was the operation of untrained adaptive receivers in environments that include co-channel interference and multipath. It was shown that existing untrained adaptive approaches have characteristics that make them unsuitable for such hostile conditions. This led to the development of the LCCM approach and its subsequent analysis, derivation and simulation. The results presented indicate that LCCM is a promising alternative to the existing untrained algorithms.

The LCCM method was motivated in part as a way to prevent the capture problem that affects UCM. Chapter 4 shows that in a narrowband array application LCCM cannot be captured by independent interfering signals. This is true whether background noise is present or not. Moreover, the SOI does not need to be constant modulus for this to be true. This opens up the possibility of applying the LCCM approach to non-CM signals in the same way that UCM was applied to non-CM signals [31]. Of course, these results are restricted to the narrowband array case and further, the constraint equation is assumed to be perfect. That is, that the SOI lies only in the signal subspace that is represented by the constraint equation. This is a design goal that is usually not met exactly in practice. The correlated signal analysis of Chapter 4 also made an assumption that is not usually encountered in practice, i.e., the signals and gains are real. In fact, this actually contradicts the narrowband array assumption. Nevertheless, these results are still useful for their insights into the effects of correlation and noise on local minima of LCCM. There is relatively little published analysis for any adaptive method that is based on constant modulus principles [45]. There is even less for CM-based methods that are operated in correlated interference.

In contrast to the specialized local minima analysis, the LCCM adaptive recursion itself is very robust. The recursions that are derived in Chapter 5 never showed any signs of instability in simulation. The behavior was predictable and controllable in all cases examined. Even in the cases where δ was fixed to the wrong value the results were as expected. The adaptive δ version of LCCM performed predictably as well. Like other SGD algorithms, the convergence rate of LCCM can be improved at the cost of increased misadjustment. This may be appropriate in certain applications.

Both of the applications that are simulated in this dissertation can be viewed as LCCM in a start-up or *acquisition* mode. That is, LCCM is used in place of a reference or training signal in order to initialize a set of adaptive weights. These weights are then used as the start-up weights for another adaptive processor such as one based on decision feedback. In both the array and equalizer simulations the LCCM algorithm performed this role successfully. There are other possible modes of operation for LCCM, however. One is *signal conditioning* where LCCM is used as a pre-processor for some other receiver element. A signal conditioning application has not been simulated.

Indeed, there a many applications of LCCM that have not been simulated in this dissertation. In addition, there are still many unaddressed issues that concern the LCCM method. It is not the intent of this dissertation to consider every facet of LCCM. The emphasis of this dissertation was on the motivation, development and verification of the LCCM concept. Some of the more important issues that deserve research effort in the future are described in the next section.

7.2 Areas for Further Research

Robustness

The constraint equation is fundamental to the performance of the LCCM algorithm. It is natural to ask how sensitive LCCM is to errors in the constraints. Clearly it is not as sensitive to constraint error as is LCMP. After all, if the constraint equation is absent LCCM simply becomes UCM. The robustness issue is really one that is determined by the nature of the application. Although practical implementation is the most revealing, analysis and simulation can offer insight

into robustness as well. It is possible that the narrowband array application can be investigated analytically and research on this topic is ongoing. For simulations to be useful in the study of robustness the signal environments will have to be more realistic than those used in Chapter 6.

Non-Communication Applications

Communication scenarios are not the only applications of LCCM. The results of Chapter 4 suggest that LCCM can be used to adapt a narrowband array for signals that do not have a constant envelope. There is potential, then, that LCCM could be applied to acoustic array waveforms. For instance, LCCM could be used in place of LCMP to perform interference cancellation using arrays of acoustic sensors.

Linear Constraints for Other Adaptive Algorithms

In principle, the decomposition property of the GSC structure could be used to impose constraints on any adaptive algorithm. In this way the performance or robustness of the algorithm could potentially be improved in the same way that LCCM extends the utility of UCM. One possible candidate is the Decision Feedback technique. Other candidate algorithms have not been identified but new adaptive algorithms are published regularly.

Parallels with UCM

Research on the unconstrained constant modulus algorithm has been active in areas such as convergence rate improvement [46, 47, 48] and frequency domain implementations [49, 50]. Likewise, these are important areas for future research on LCCM.

References

[1] F. R. Magee and J. G. Proakis, "Adaptive maximum-likelihood sequence estimation for digital signaling in the presence of intersymbol interference," *IEEE Trans. Information Theory*, vol. IT-19:120–124, Jan. 1973.

[2] R. Price and P. E. Green, Jr, "A communication technique for multipath channels," *Proceedings, IRE*, Vol. 46:555–569, Mar. 1958.

[3] B. Widrow et al., "Adaptive noise cancelling: Principles and applications," *Proceedings, IEEE*, 63:1692–1716, Dec. 1975.

[4] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-33:222–229, Feb. 1985.

[5] C.F.N. Cowan and P.M. Grant, *Adaptive Filters*, Prentice-Hall, 1985.

[6] M. E. Austin, "Decision-feedback equalization for digital communication over dispersive channels," Technical Report # 437, MIT Lincoln Laboratory, Lexington, MA, August 1967.

[7] W.A. Gardner, "Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis and critique," *Signal Processing*, 6:113–133, 1984.

[8] B.G. Agee, S.V. Schell and W.A. Gardner, "Self-coherence restoral: A new approach to blind adaptation of antenna arrays," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, pages 589–593, Pacific Grove, CA, Nov. 1987.

[9] O.L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings, IEEE*, 60:926–935, Aug. 1972.

[10] C. Y. Tseng, *"Adaptive linearly constrained filtering: Principles and implementations"*, PhD thesis, University of Southern California, Los Angeles, CA, May 1990.

[11] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. on Antennas and Propagation*, AP-30:27–34, Jan. 1982.

[12] H. Cox et al., "Robust adaptive beamforming," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-35:1365–1376, Oct. 1987.

[13] B. Widrow et al., "Signal cancellation phenomena in adaptive antennas: Causes and cures," *IEEE Trans. on Antennas and Propagation*, AP-30:469–478, May 1982.

[14] Y.-L. Su et al., "Parallel spatial processing: A cure for signal cancellation in adaptive arrays," *IEEE Trans. on Antennas and Propagation*, AP-34:347–355, Mar. 1986.

[15] L. J. Griffiths and M. J. Rude, "Adaptive filtering without a desired signal," In *Proc. of the Int'l. Conference on Acoustics, Speech, and Signal Processing*, pages 105–108, Dallas, TX, Apr. 1987.

[16] L. J. Griffiths, "A simple adaptive algorithm for real-time processing in antenna arrays," *Proceedings, IEEE*, 57:1696–1704, Oct. 1969.

[17] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation systems,," *IEEE Trans. on Communications*, COM-23:679–682, June 1975.

[18] D. N. Godard, "Self-Recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. on Communications*, COM-28:1867–1875, Nov. 1980.

[19] J. R. Treichler and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-31:459–472, Apr. 1983.

[20] J. R. Treichler and M. G. Larimore, "New processing techniques based on the constant modulus adaptive algorithm," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-33:420–431, Apr. 1985.

[21] J.R. Treichler and M.G. Larimore, "The tone capture properties of CMA-based interference suppressors," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-33:946–958, Aug. 1985.

[22] M. J. Rude and L. J. Griffiths, "Incorporation of linear constraints into the Constant Modulus Algorithm," In *Proc. of the Int'l. Conference on Acoustics, Speech, and Signal Processing*, pages 968–971, Glasgow, Scotland, UK, May 1989.

[23] R.P. Gooch and J. Lundell, "The CM array: An adaptive beamformer for constant modulus signals," In *Proc. of the Int'l. Conference on Acoustics, Speech, and Signal Processing*, pages 2523–2526, Tokyo, Japan, Apr. 1986.

[24] R. P. Gooch and B. Daellenbach, "Prevention of interference capture in a blind (CMA-based) adaptive receive filter," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1989.

[25] M.G. Larimore and J.R. Treichler, "Convergence behavior of the constant modulus algorithm," In *Proc. of the Int'l. Conference on Acoustics, Speech, and Signal Processing*, pages 13–16, Boston, MA, Apr. 1983.

[26] B. G. Agee, "Convergent behavior of modulus-restoring adaptive arrays in Gaussian interference environments," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1988.

[27] K. M. Buckley, *"A source representation space approach to digital array processing"*, PhD thesis, University of Southern California, Los Angeles, CA, Aug. 1986.

[28] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, 1985.

[29] K. M. Buckley and L. J. Griffiths, "An adaptive generalized sidelobe canceller with derivative constraints," *IEEE Trans. on Antennas and Propagation*, AP-34:311–319, Mar. 1986.

[30] G. Strang, *Linear Algebra and Its Applications*, Academic Press, 1980.

[31] J. Lundell and B. Widrow, "Applications of the constant modulus adaptive beamformer to constant and non-constant modulus signals," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, pages 432–435, Pacific Grove, CA, Nov. 1987.

[32] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*, Wiley, 1980.

[33] R. T. Compton, Jr, *Adaptive Antennas*, Prentice-Hall, 1988.

[34] C. Y. Tseng and L. J. Griffiths, "Decomposing the blocking matrix in the generalized sidelobe canceller," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, pages 574–578, Pacific Grove, CA, Nov 1987.

[35] C. Y. Tseng and L. J. Griffiths, "A systematic procedure for implementing the blocking matrix in decomposed form," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, pages 808–812, Pacific Grove, CA, Nov 1988.

[36] C. Y. Tseng and L. J. Griffiths, "A unification and comparison of adaptive linearly-constrained beamforming structures," In *SPIE's 33rd Annual International Technical Symposium*, pages 245–256, San Diego, CA, August 1989.

[37] J. R. Treichler, C. R. Johnson, Jr. and M. G. Larimore, *Theory and Design of Adaptive Filters*, Wiley-Interscience, 1987.

[38] L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive array," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-29:722–736, June 1981.

[39] L. J. Griffiths, "An adaptive lattice structure for noise-cancelling applications," In *Proc. of the Int'l. Conference on Acoustics, Speech, and Signal Processing*, pages 87–90, Tulsa, OK, Apr. 1978.

[40] G. Ungerboeck, "Fractional tap-spacing equalizer and consequences for clock recovery in data modems," *com*, COM-24:856–864, 1976.

[41] R. D. Gitlin and S. B. Weinstein, "Fractionally-spaced equalization: An improved digital transversal equalizer," *Bell System Technical Journal*, Vol. 60:275–296, 1981.

[42] A. Gersho, "On combining adaptive equalization and filtering," Unpublished paper, 1969.

[43] V. Wolff et al., "Specification and development of an equalizer-demodulator for wideband microwave radio signals," In *Proc. of the IEEE Military Communications Conference*, 1988.

[44] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*, Wiley, 1965.

[45] J.O. Smith and B. Friedlander, "Global convergence of the constant modulus algorithm," In *Proc. of the Int'l. Conference on Acoustics, Speech, and Signal Processing*, pages 1161–1164, Tampa, FL, Apr. 1985.

[46] B. G. Agee, "The least-squares CMA: A new technique for rapid correction of constant modulus signals," In *Proc. of the Int'l. Conference on Acoustics, Speech, and Signal Processing*, pages 953–956, Tokyo, Japan, Apr. 1986.

[47] R. P. Gooch, M. Ready and J. Svoboda, "A lattice-based Constant Modulus adaptive filter ," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1986.

[48] J.O. Smith and B. Friedlander, "Extensions of the Constant Modulus algorithm," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1984.

[49] J. R. Treichler, S. L. Wood and M. G. Larimore, "Convergence rate limitations in certain frequency-domain adaptive filters," In *Proc. of the Int'l. Conference on Acoustics, Speech, and Signal Processing*, pages 960–963, Glasgow, Scotland, UK, May 1989.

[50] C. K. Chan, M. R. Petraglia and J. J. Shynk, "Frequency-domain implementations of the constant modulus algorithm," In *Proc. of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1989.