ADAPTIVE FRACTIONALLY-SPACED EQUALIZATION WITH EXPLICIT SIDELOBE CONTROL USING INTERIOR POINT OPTIMIZATION TEXHNIQUES

ADAPTIVE FRACTIONALLY-SPACED EQUALIZATION WITH EXPLICIT SIDELOBE CONTROL USING INTERIOR POINT OPTIMIZATION TEXHNIQUES

By

ASHISH MITTAL

B.Tech, EE Dept, IIT Madras, India

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Masters of Applied Science

McMaster University

Master of Applied Science (2005) McMaster University
(Electrical and Computer Engineering) Hamilton, Ontario

TITLE:	Adaptive Fractionally-Spaced Equalization with Explicit Sidelobe Control using Interior Point Optimization Techniques
AUTHOR:	Ashish Mittal Bachelor of Technology, Electrical Engineering Indian Institute of Technology, Madras, India
SUPERVISOR:	Dr. T. N. Davidson
NUMBER OF PAGES:	xii, 103

To my parents

and

brother

Abstract

This thesis addresses the design of fractionally-spaced equalizers for a digital communication system which is susceptible to Adjacent Channel Interference (ACI). ACI can render an otherwise well designed system prone to excess bit errors. Algorithms for a trained adaptive FIR linear fractionally-spaced equalizer (FSE) with explicit sidelobe control are developed in order to provide robustness to ACI. The explicit sidelobe control is achieved by imposing a quadratic inequality constraint on the frequency response of the equalizer at a discrete set of frequency points in the sidelobe region.

Algorithms are developed for both block adaptive and symbol-by-symbol adaptive modes. These algorithms use interior point optimization techniques to find the optimal equalizer coefficients. In the block adaptive mode, the problem is reformulated as a Second Order Cone Program (SOCP). In the symbol-by-symbol adaptive mode, the philosophy of the barrier approach to interior point methods is adopted. The concept of a central path and the Method of Analytic Centers (MAC) are used to develop two practically implementable algorithms, namely IPM2 and SBM, for performing symbol-by-symbol adaptive, fractionally-spaced equalization, with multiple quadratic inequality constraints.

The performance of the proposed algorithms is compared to that of the Wiener filter, and the standard RLS algorithm with explicit diagonal loading. In the computer simulations, the proposed algorithms perform better in the sense that they provide the desired robustness when the communication model is prone to intermittent interferers in the sidelobe region of the frequency response of the FSE. Although the proposed algorithms have a moderately higher computational cost, their insensitivity to the deleterious effects of ACI make them an attractive choice in certain applications.

Acknowledgments

I would like express my sincere thanks to Dr. T. N. Davidson for his guidance as my supervisor and continuous support during the period of this work. In particular, for his precious time and valuable comments which helped shape this thesis. I am grateful to him for igniting my interest in the field of interior point optimization, and its applicability to constrained adaptive equalization. Working with him has been an enormous learning experience.

I would like to thank our graduate secretary Cheryl Gies for being very supportive and considerate during my stay here. My thanks to Helen Jachna for always being patient with my requests. I would also like to extend my thanks to Cosmin and Terry for making sure our computing facilities stay functional without interruptions.

Many thanks to Dr. Abhijit Sinha and Mehran for the insightful discussions and valuable suggestions. To all my good friends and coffee break enthusiasts, esp. Pravesh and KK. I wish everyone the very best in their work and life. Finally, I would like to thank my parents and my dear brother Nitish, for their endless support and encouragement.

Overall, my goal was to gain knowledge and in the process be able to contribute towards the ongoing research in our group. I am grateful for the opportunity and express hope that this work will be continued further. I am sure future researchers will experience great thrill, as we did, in exploring the possibilities of this exciting area.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Equalization is an important component of the receiver in many digital communication schemes. Therefore, it is one of the better studied components in digital communications. Over the years, many researchers have made significant contributions to this field, and there is a wealth of knowledge available regarding several useful algorithms and the instances of those algorithms that are used in individual standards [1], [2]. In particular, the use of fractionally-spaced equalizers, instead of symbol-spaced equalizers, has greatly improved the equalization capability, especially when using an FIR filter as the equalizer. The use of adaptive equalizers further allows equalization over time-varying channels. However, most of the available equalization algorithms address unconstrained equalization problems. There can be scenarios in which one would like to implement an equalization algorithm with multiple equality and inequality constraints. In such a scenario, the conventional equalization algorithms/architectures are insufficient. In such a case, the path to the development of a new equalization algorithm is to model the equalization problem as a convex optimization problem with multiple equality and inequality constraints, and this is the approach that will be taken in this thesis.

In this work, we solve a training-based constrained fractionally-spaced adaptive equalization problem using the theory and concepts of convex optimization. The problem of adaptive equalization has been treated several times, but the novelty of the proposed approach lies in the way in which we handle multiple quadratic inequality constraints. We develop algorithms for adaptive constrained fractionally-spaced equalization, in block adaptive and symbol-by-symbol adaptive forms. These algorithms are based on the principles of interior point methods [4], [8] for the solution of convex optimization problems with multiple quadratic inequality constraints.

The motivation for the development of our constrained fractionally-spaced adaptive equalizer comes from applications in which there is intermittent adjacent channel interference. That is, interference in the sidelobe region of the frequency response of the fractionally-spaced equalizer that might not be present during the training phase of the equalizer. A conventional adaptive equalization algorithm, like LMS or RLS [1], [3], might have unacceptably high sidelobes in the frequency response of the equalizer. Should an unexpected interferer strike the sidelobe region during data transmission, it might result in a severe symbol error rate at the detector.

In order to achieve robustness to intermittent interferers in the sidelobe region of the fractionally-spaced equalizer, we constrain the frequency response of the equalizer to be below a desired sidelobe level over the sidelobe region. This is achieved by imposing multiple quadratic inequality constraints on the equalizer coefficients, and hence the designer can exercise explicit control over the sidelobe level.

Using this approach, the problem of performing constrained adaptive equalization is translated into solving an optimization problem with multiple quadratic inequality constraints. We develop both block adaptive and symbol-by-symbol adaptive equalizers with explicit sidelobe control. In these scenarios, we use different approaches to implement the principles of interior point methods. To solve the optimization problem in the block adaptive case, we model our problem as a Second Order Cone Program (SOCP), and use well-developed primal-dual interior point methods, such as that implemented in SeDuMi [4]. The symbol-by-symbol adaptive equalization problem is more challenging because a new filter that satisfys the sidelobe constraints must be computed for each received symbol. We solve this problem using the principles of the barrier method [8] approach. In order to evaluate the performance of the proposed equalizers, we compare their performance to that of standard equalizers, such as the Wiener Filter and the RLS algorithm with explicit diagonal loading, and discuss the advantages and disadvantages of our approach.

1.1 Adjacent Channel Interference (ACI)

In a practical communications scheme, transmitter imperfections result in adjacent channel emissions which lie in the transmitted band of spectrally neighboring schemes. This occurs due to insufficiently stringent limitations on the acceptable leakage at the transmitter. These interfering signals in the adjacent channel pose a threat to the successful communication for an operator using this adjacent frequency band, and are said to constitute adjacent channel interference (ACI). The affected operator in the adjacent band has no control over interference from other users. Therefore, it is possible that a majority of bit errors might be due to ACI and not due to the given channel or the inherent noise at the receiver. The ability of a receiver to accept signals at the desired frequencies, and suppressing any signals from the adjacent channels is called the selectivity of the receiver, or adjacent channel rejection.

In cellular mobile radio systems that employ TDMA and FDMA, there is substantial amount of frequency re-use in order to achieve high spectral efficiency. Spectrally adjacent channels are kept in separate cells, so as to minimize the impact of spectral overlap. However, adjacent channel interference can still occur due to mobile users in different cells. Spectral overlap between the signal of interest, and interfering signals from adjacent channels, can result in severe performance degradation. In 2G systems, such as GSM, the ACI was managed by allocating guard bands of unused spectrum between operators. This would generally ensure that the adjacent channel interference would attenuate to acceptable levels. However, in 3G systems using wideband code division multiple access (WCDMA), the higher channel bandwidth requirement means that this approach is not feasible.

When the level of ACI is a significant fraction of the desired signal power, it can lead to disruption of signal quality at the receiver. Sometimes ACI can lead to formation of dead zones in the network where the adjacent operator blinds the mobile user and the QoS target cannot be achieved. Therefore, due to the wide bandwidth requirements, a typical WCDMA system is limited by the extent of interference; hence ACI has become an important issue for WCDMA.

To combat the problem of ACI, it makes sense to try to curb the problem at the source itself, by incorporating better transmitter designs at the base station which would reduce emissions into the neighboring channel. This would certainly help the operator using a neighboring channel. Ironically, this operator could be a competitor, in which case there might be little motivation for the wireless company to improve on the transmitter design. Besides, the user can experience ACI from other mobile users as well. Hence, it is prudent to make the receivers robust to ACI.

In summary, the design of the receivers ought to incorporate measures to combat ACI, along with the control of ISI and noise. Conventional receivers with no treatment of the sidelobe regions often have unacceptably high sidelobe levels and can be quite susceptible to degradation caused by intermittent interferers in adjacent channels. To ensure robustness to ACI, explicit control over the sidelobe regions of the spectrum of the equalizer presents itself as an attractive proposition. This opens up an exciting avenue to improve upon conventional receiver designs, especially if the user can adjust the robustness of the receiver at will, depending the extent of interference that is expected. Considering the potentially adverse effects of ACI, it does not appear unreasonable to invest a moderately higher computational cost, if required, to attain explicit control over the sidelobe region of the receiver.

In this thesis several ACI immune receiver structures are proposed. We investigate trade-offs between performance degradation due to ACI and computational complexity. We compare our performance with the standard RLS algorithm with explicit diagonal loading. This is a good choice for comparison as it has a lower complexity than our interior point based algorithms, and also boasts of marginal sidelobe suppression due to explicit diagonal loading.

1.2 Contribution and Organization

The focus of this work is to develop practically implementable block adaptive and symbol-by-symbol adaptive fractionally-spaced equalization algorithms with multiple quadratic inequality constraints on the sidelobe region of the frequency response of the equalizer. We demonstrate the usefulness of interior point based algorithms in achieving this task. The aim is to be able to analyze and and develop algorithms which solve the core problem of attaining a good sub-optimal solution with a limited number of available training symbols. For simplicity we use linear equalizers and test our symbol-by-symbol adaptive algorithms in a block fading scenario. Our algorithms can be used in scenarios prone to ACI, as discussed in the previous section. At a moderately higher computational cost, now we have a class of receiver structures that is insensitive to intermittent ACI, and in which the level of robustness can be explicitly controlled by the designer.

The thesis is organized as follows. Chapter 2 provides a general background on popular equalization algorithms and architectures using a linear filter. This chapter considers scenarios involving unconstrained equalization. In Chapter 3 we develop a mathematical formulation of the problem from a block adaptive standpoint, and model the problem as a Second Order Cone Program (SOCP). Chapter 4 presents the theory of interior point based techniques for solving optimization problems with multiple quadratic constraints, and barrier methods in particular. We discuss the concept of the central path, and its importance in using the barrier methods. The application of barrier methods to perform symbol-by-symbol constrained adaptive equalization is developed in Chapter 5. We discuss different approaches and their limitations. Chapter 6 demonstrates the application of the theoretical concepts by computer simulations. Finally we present conclusions and potential avenues for future research in Chapter 7.

Chapter 2

Equalization

2.1 Introduction

Equalizers are a fundamental component of many digital communication systems [1], [2]. In a practical scenario, the channel over which the symbols are transmitted is often non-ideal. Therefore, its frequency response characteristics are such that the channel introduces amplitude and phase distortion in the frequency response of the received signal. This, in turn, causes overlapping of some of the received pulses, leading to intersymbol interference (ISI). If left unchecked, the ISI can be a major source of symbol errors at moderate-to-high signal-to-noise ratios (SNRs), at the receiver, and hence there is a need to provide a mechanism by which the receiver can compensate for the channel distortion and free the received signal from ISI, or at least minimize its deleterious effects. Hence the need for an equalizer, which compensates or equalizes the channels degrading effects, thereby reducing ISI.

There are several standard approaches to performing equalization. We can perform optimum detection from a probability of error point-of-view at the receiver by doing a Maximum Likelihood Sequence Estimation (MLSE) using the Viterbi Algorithm, where decisions are made on entire sequences of received symbols. However, that approach has a high computational cost, which increases exponentially with the length of the channel (in symbol intervals). In a practical scenario, a good sub-optimal equalizer with a lower computational cost might be more desirable, than a computationally intensive optimal equalizer. Hence, several sub-optimal equalization algorithms have been proposed, which work well in different scenarios. If the channel is unknown at the receiver, and/or time varying, then the equalizer needs to be consistently updated to keep track of the changing channel conditions. Hence, the need for *adaptive equalization.* An adaptive equalizer can either be block adaptive or symbol-by-symbol adaptive, depending on the application. Both approaches will be elaborated on later in the chapter.

Equalization is a very well studied field and in the subsequent sections we give a general idea of some of the prevalent approaches to equalization. This chapter attempts to provide a good background which will enable the reader to get a glimpse of the amount of effort made to perform unconstrained adaptive equalization. This will in turn enable the reader to understand and appreciate the adaptive equalization algorithms with multiple quadratic inequality constraints (both block adaptive and symbol-by-symbol adaptive) we develop later on the the thesis.

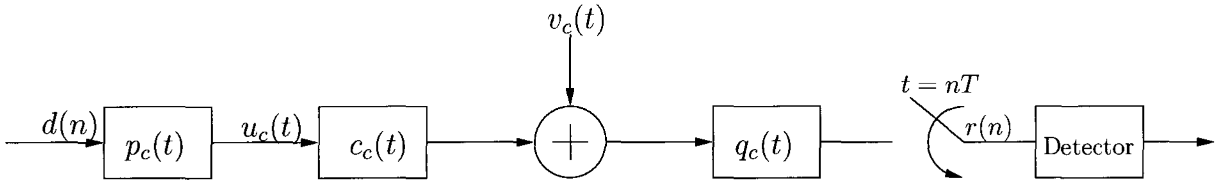

Figure 2.1: Standard model for baseband PAM.

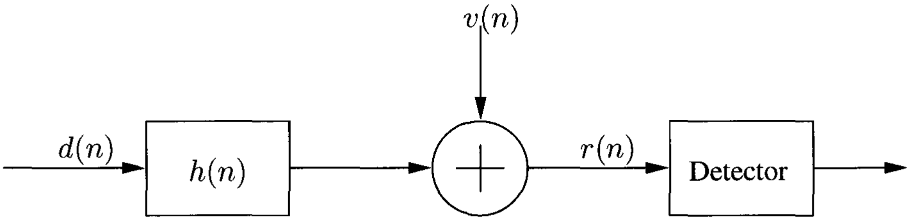

Figure 2.2: Equivalent discrete-time model for baseband PAM.

2.2 Models

Consider the standard model for baseband digital communication shown in Fig. 2.1. The data is waveform coded by pulse amplitude modulation (PAM). The transmitted waveform $u_c(t)$ can be written as

$$u_c(t) = \sum_n d(n)p_c(t - nT),$$

where $d(n)$ is the input data sequence, n is the symbol spaced time index, T is the symbol interval, and $p_c(t)$ is the pulse shaping filter. The signal $u_c(t)$ is transmitted through the linear time-invariant (LTI) baseband equivalent channel $c_c(t)$ and the noise, which is assumed to be zero-mean, white and Gaussian, is represented by $v_c(t)$. In general, all quantities are complex valued. This noisy signal is fed to the demodulator $q_c(t)$. A popular choice is to choose the demodulator to be the so-called "matched" filter, $q_c(t) = p_c^*(-t)$. This is a good choice for an AWGN channel where are errors are due to noise. The sampled output of the matched filter $q_c(t)$ is a sequence which has sufficient statistics for making a correct decision on the received signal [2]. Ideally for ISI channels, the prefilter $q(t)$ should be matched to the cascade of the transmit filter, and the channel model. However, more often the channel characteristics are not known a priori at the receiver. Therefore, in general the receiver prefilter $q(t)$ is chosen as a matched filter to the transmit filter.

It is convenient to view the communication model in Fig. 2.1 as a discrete time system. The equivalent discrete time model for baseband PAM is shown in Fig. 2.2. From Fig. 2.1 and Fig. 2.2, the equivalent discrete time channel model is

$$h(n) = \int c_c(\lambda) r_{pq}(nT - \lambda) d\lambda,$$

where

$$r_{pq}(\nu) = \int q_c(\vartheta) p_c(\nu - \vartheta) d\vartheta$$

and

$$v(n) = \int q_c(\lambda) v_c(nT - \lambda) d\lambda.$$

We need to correctly extract the information contained in the samples in order to successfully estimate the transmitted symbols. There are several detection schemes that can be employed for optimal or sub-optimal detection depending on the application. We discuss some of the standard approaches in the following sections.

2.3 Optimum Receivers

Compensation for ISI or optimum detection of transmitted symbols at the receiver are the aim of most communication receivers. In an ideal case, the channel is AWGN, and does not introduce any distortion in the received signal. In that case, the channel corrupts the transmitted signal only by the addition of white Gaussian noise. An optimum receiver in such a case would comprise of a matched filter, a symbol rate sampler and a memoryless optimum detector. The probability of error can be minimized by performing optimum detection at the demodulator output. This decision criterion can be expressed as a conditional probability function (likelihood function). For optimum detection we maximize the likelihood function, hence the name Maximum-Likelihood (ML) criterion. This in turn translates itself into computing a set of correlation metrices [2] and selecting the signal corresponding to the largest correlation metric. This symbol-by-symbol detection scheme is optimum when the signal has no memory.

However, in the scenario when the signal has memory, the optimum detector must make decisions on a sequence of received signals, and estimate an entire sequence of transmitted symbols, instead of performing symbol-by-symbol detection. The received signals in successive symbol intervals are now interdependent, and the optimal detector must calculate the probability of correct decision on a sequence of received signals. This is done by calculating a joint pdf of the outputs of the demodulator, and then selecting a sequence which maximizes this likelihood function. Therefore, this is called the Maximum-Likelihood Sequence Detector. Due to the Gaussian nature of the noise, maximizing the conditional PDFs translates itself into minimizing a Euclidean Distance function [2]. If the size of the symbol alphabet is M and the length of the sequence is K, the task is to calculate the Euclidean distance function for all the M^K possible sequences, and then choose the sequence for which the Euclidean distance function is minimum.

The computational complexity of the ML sequence detector can be reduced by using the Viterbi algorithm (VA). This algorithm acts as a Maximum-Likelihood Sequence Estimator (MLSE) and performs a sequential trellis search, while eliminating sequences with each new received signal. It reduces the number of paths searched in the trellis to find a sequence that minimizes the Euclidean distance function and in turn maximizes the probability of correct decision on a sequence of received signals. Even though VA brings down the initial computational complexity, it still requires M^{L-1} channel states [1], where L is the length of the channel. Hence, MLSE is difficult to implement in a practical scenario having a large signal constellation, or long channels. This kind of situation might be seen when we compare the use of MLSE in the Global System for Mobile Communications (GSM) and in Enhanced Data rates for GSM evolution (EDGE). One of the standard channel models used in GSM and EDGE is of length 7 [14] . Then, for GSM which uses binary modulation, the VA would

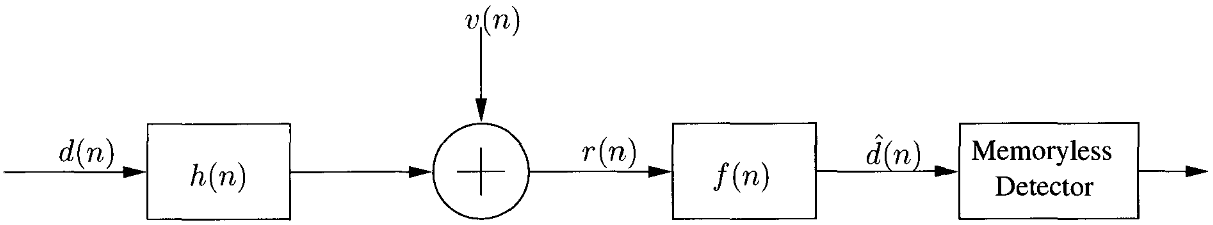

Figure 2.3: Equivalent discrete-time model for baseband PAM using an equalizer.

need $2^6 = 64$ states. On the other hand for EDGE, where 8-ary PSK is used, the VA would require $8^6 = 262144$ states.

2.4 Sub-Optimal Receivers

In order to reduce computational cost, sub-optimal channel equalization techniques have become popular in many practical applications. One popular choice is to use a linear transversal equalizer after the matched filter, followed by a memoryless detector. A linear transversal filter offers lower computational complexity to perform channel equalization to compensate for ISI. In the MLSE the cost rises exponentially with the channel length, whereas in this case it is a linear function of the length of the channel. However, we have to settle for a sub-optimal solution. A scheme involving the linear transversal equalizer in a communication model is shown in Fig. 2.3.

There are different approaches to implement a linear transversal equalizer. We discuss two popular criteria, namely the peak distortion criterion and mean squared error criterion.

Zero-Forcing Equalizer

The worst-case intersymbol interference at the output of the equalizer is called the *peak distortion*. An equalizer designed with the aim of minimizing the peak distortion is said to be based on the peak distortion criterion. It is called a Zero-Forcing Equalizer and it derives its name from the fact that it attempts to achieve zero ISI. (Zero ISI might not be achievable using a finite length filter.) Consider the communication scheme depicted in Fig. 2.3. For convenience, we define, $g(n) = h(n) \otimes f(n)$, where $\otimes$ denotes convolution [20], so that

$$g(n) = \sum_i f(i)h(n-i).$$

The output $\hat{d}(n)$ can be written as

$$\begin{aligned}\hat{d}(n) &= d(n) \otimes g(n) + v(n) \otimes f(n) && (2.1)\\ &= \sum_i g(i)d(n-i) + \sum_j f(j)v(n-j) && (2.2)\\ &= g(\delta)d(n-\delta) + \sum_{i\neq\delta} g(i)d(n-i) + \sum_j f(j)v(n-j). && (2.3)\end{aligned}$$

We assume that the output $\hat{d}(n)$ is an estimate of the transmitted symbol $d(n-\delta)$, where δ represents the system delay. Therefore, the error between the input and output symbols can be defined as

$$e(n) = \hat{d}(n) - d(n-\delta). \tag{2.4}$$

We see that the first term in (2.3) is a scaled version of the desired response. We normalize $g(\delta)$ to unity for convenience. Therefore, we have that

$$e(n) = \sum_{i\neq\delta} g(i)d(n-i) + \sum_j f(j)v(n-j), \tag{2.5}$$

where the first term is the ISI term and the second term is due to noise. Here we mathematically see that the symbol error results from a combination of the ISI and noise.

The optimum taps of the linear equalizer are those for which the peak value of the interference, or the peak distortion

$$\sum_{i\neq\delta} |g(i)|,$$

is minimized. For an infinite number of taps it is possible, under the conditions outlined below, to obtain a set of filter coefficients for which the peak distortion for the given system goes to zero. Therefore, it is possible to completely eliminate the ISI and thus the symbol errors would only be due to noise.

It is useful to investigate the given problem using the z transform. If $H(z)$ is the z transform of the discrete time channel, then the z transform $F(z)$ of a zero-forcing equalizer would be given by $1/H(z)$. Due to the inverse relation of the z transforms of the channel and the equalizer, the zeros of $H(z)$ are the poles of $F(z)$, and vice-versa. Therefore, for the equalizer to be causal and stable, the channel should be minimum phase. A minimum phase system is one for which the zeros and poles in the z transform are restricted to lie inside the unit circle [20].

We observe that the peak distortion criterion gives no consideration to the white Gaussian noise. Due to the inverse nature of the of z transform of the zero-forcing equalizer, it might enhance noise power when the channel zeros are close to the unit circle. Therefore, for deep nulls, and even for moderate depressions in the channel frequency response, the performance of a linear infinite length zero-forcing equalizer might be poor. In a practical scenario, a finite length (FIR) filter is used instead of the infinite impulse response (IIR) filter that we have discussed up until this point. For the FIR filter, the taps are chosen so that the peak distortion is minimized, however it is not possible, in general to make the peak distortion to go to zero completely due to a finite number of taps.

Minimum Mean Squared Error (MMSE) Equalizer

While the focus of a zero-forcing equalizer is only on reducing the ISI, an MMSE equalizer takes into account the effects of both ISI and noise. The tap weights of the linear MMSE equalizer are adjusted so as to minimize the mean squared value of the error,

$$J = E\{|\hat{d}(n) - d(n-\delta)|^2\}, \tag{2.6}$$

where $E\{\cdot\}$ denotes the expectation operator. We assume that the data symbols are independent and identically distributed (i.i.d), with zero mean, and are uncorrelated to the zero mean white Gaussian noise. The z transform $F(z)$ of the optimal infinite length equalizer is given by $1/(H(z) + N_0)$, where $H(z)$ is the z transform of the discrete time channel, and N_0 is the noise spectral density [2]. Due to the special consideration given to the noise along with ISI, the MSE for an MMSE equalizer is always lower then the MSE at the output of an zero-forcing equalizer, although for higher SNRs, both MMSE and zero-forcing equalizers are expected to give a similar performance. Nevertheless, in the presence of noise, the ISI never goes to zero for a linear infinite length MMSE equalizer. That said, it is still the preferred choice over zero-forcing equalizer in many applications due to its treatment of noise along with ISI.

To find the optimal tap weights (optimum in a mean-squared-error sense), the MSE defined in (2.6) is minimized. This leads to solving the *Wiener-Hopf* equations [3], whose solution requires true knowledge of the channel characteristics, and the resulting filter is called the Wiener Filter. In a practical implementation a finite length linear equalizer is used instead of an IIR filter. For a M-tap linear filter, the MSE criterion leads to solving a system of M simultaneous equations, to get the optimal tap weights. Consider an input vector

$$\mathbf{r}_n = \Big(r(n), r(n-1), r(n-2), ..., r(n-M+1)\Big)^T,$$

where $r(n)$ is the input to the equalizer in Fig. 2.3 and $(\cdot)^T$ denotes transpose. The output $\hat{d}(n)$ becomes

$$\hat{d}(n) = \mathbf{f}^H \mathbf{r}_n,$$

where $(\cdot)^H$ denotes Hermitian transpose, and

$$\mathbf{f} = \Big(f^*(0), f^*(1), f^*(2), ..., f^*(M-1)\Big)^T,$$

where $(*)$ denotes complex conjugate. Equation (2.6) now becomes

$$E\{|e(n)|^2\} = \mathbf{f}^H\mathbf{R}\mathbf{f} - \mathbf{f}^H\mathbf{p} - \mathbf{p}^H\mathbf{f} + \sigma_d^2, \tag{2.7}$$

where

$$\mathbf{R} = E\{\mathbf{r}_n\mathbf{r}_n^H\}$$

is the $M \times M$ auto-correlation matrix of the equalizer input vector $\mathbf{r}_n$, and the cross-correlation vector is,

$$\mathbf{p} = E\{\mathbf{r}_n d^*(n-\delta)\},$$

where $d(n)$ is the transmitted symbol, δ represents system delay and $(*)$ denotes complex conjugation. Minimizing the mean squared error in (2.7) yields the optimal FIR MMSE filter [2]

$$\mathbf{f}_{opt} = \mathbf{R}^{-1}\mathbf{p}. \tag{2.8}$$

The minimum MSE in a linear symbol-spaced equalizer is dependent on the sampler phase at the receiver, which might be a source of errors. Next we consider the design of a fractionally-spaced linear equalizer, which offers several advantages over a symbol spaced equalizer.

Fractionally Spaced Equalizers

Another design using a linear filter is the so called *fractionally-spaced* equalizer. In this design, the received signal is sampled at a rate higher than the baud rate at which the symbols are transmitted. The motivation for that is the fact that there is an excess bandwidth present in the received signal, due to spectral roll-offs at the band edges. Hence, it makes sense to sample at a higher rate than the symbol rate to avoid aliasing. Consequently, the tap spacing of the equalizer is a fraction of the input symbol period.

A standard model for baseband PAM communication using a fractionally spaced equal-

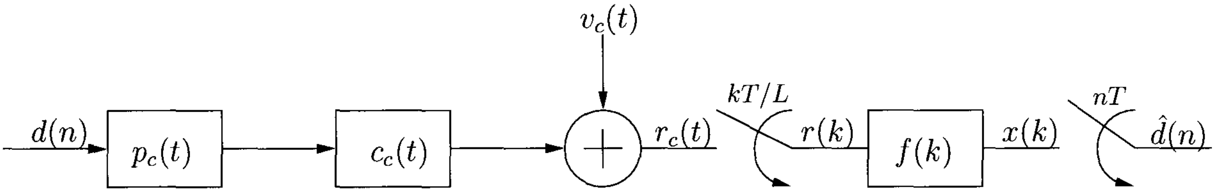

Figure 2.4: Standard model for baseband PAM using a FSE.

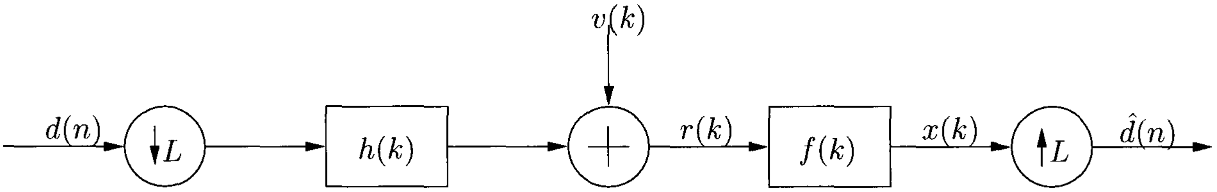

Figure 2.5: Equivalent discrete-time (multirate) model for baseband PAM using FSE.

izer (FSE) is shown in Fig. 2.4. The T-spaced symbols $d(n)$ are transmitted through the continuous time pulse shaping filter $p_c(t)$ and continuous time linear time-invariant (LTI) channel $c_c(t)$. The received signal $r_c(t)$ is corrupted by additive noise $v_c(t)$ which is assumed to be zero-mean white, and Gaussian. The noisy received signal is oversampled, where the oversampling factor L is a positive integer. The fractionally-spaced samples $r(k)$ are then linearly combined with the FIR FSE tap weights $f(k)$ to yield the equalizer output $x(k)$. For clarity, the index n ($n \in \mathbb{Z}$) is used to represent the symbol spaced quantities, and the index k ($k \in \mathbb{Z}$) is used for representing fractionally-spaced quantities. The output $x(k)$ of the equalizer is then downsampled to the symbol rate, so that the input and output symbol rates are the same. We assume that the output $\hat{d}(n)$ is an estimate of the transmitted symbol $d(n-\delta)$, where δ is the system delay.

Multirate Model

It is convenient to visualize the baseband model of Fig. 2.4 as a discrete time system. An equivalent discrete time (multirate) model for the baseband PAM communication scheme using an FSE is shown in Fig. 2.5. In this model, two successive input symbols are spaced by $L-1$ zeros, and the output of the FSE is decimated by a factor of L. The oversampled noisy received signal $r(k)$ at the input of the FSE is

$$r(k) = r_c(t)|_{t=kT/L}, \tag{2.9a}$$

$$= \sum_n d(n)h_c\Big(\frac{kT}{L} - nT\Big) + v_c\Big(\frac{kT}{L}\Big) \tag{2.9b}$$

where

$$h_c(\lambda) = \int p_c(\tau)c_c(\lambda - \tau)d\tau.$$

For simplicity, we assume that the length of the FIR FSE is NL, where L is the oversampling factor, and N is a positive integer. The output of the FSE in Fig. 2.5 is

$$x(k) = \sum_{m=0}^{NL-1} f(m)r(k-m).$$

The output symbol $\hat{d}(n) = x(Ln)$, so that

$$\hat{d}(n) = \sum_{m=0}^{NL-1} f(m)r(nL-m) \tag{2.10}$$

$$= \mathbf{f}^H\mathbf{r}_n, \tag{2.11}$$

where the vectors $\mathbf{f}$ and $\mathbf{r}_n$ are defined according to (2.10).

From (2.11), the mean squared error between the transmitted symbol $d(n)$ and its estimate $\hat{d}(n)$ (assuming a system delay of δ) becomes

$$\begin{aligned} J &= E\{|\mathbf{f}^H\mathbf{r}_n - d(n-\delta)|^2\} \\ &= \mathbf{f}^H\mathbf{A}\mathbf{f} - \mathbf{f}^H\mathbf{b} - \mathbf{b}^H\mathbf{f} + \sigma_d^2, \end{aligned}$$

where

$$\mathbf{A} = E\{\mathbf{r}_n\mathbf{r}_n^H\}$$

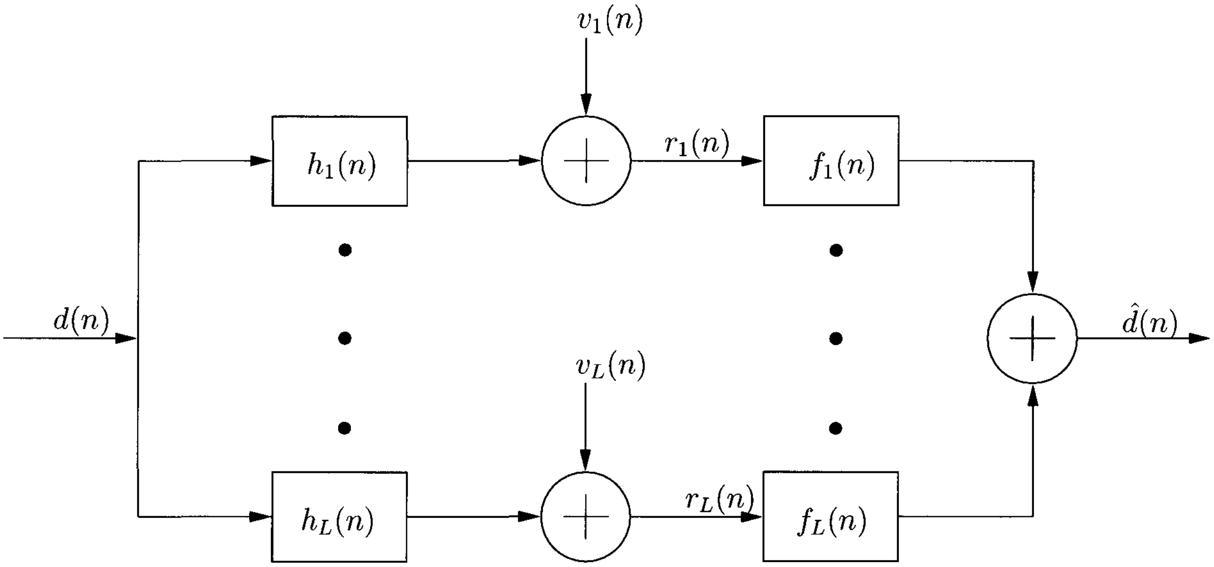

Figure 2.6: Multi-Channel Model

is the $LN \times LN$ auto-correlation matrix of the equalizer input vector $\mathbf{r}_n$, and the $LN \times 1$ cross-correlation vector $\mathbf{b}$ between the FSE input and transmitted symbols is given as

$$\mathbf{b} = E\{\mathbf{r}_n d^*(n-\delta)\}.$$

Multi-Channel Model

Another useful discrete time interpretation of the continuous time baseband model in Fig. 2.4 is the multi-channel model of the FSE as shown in Fig. 2.6. From the baseband equivalent model in Fig. 2.4 we have

$$r(k) = \sum_n d(n) h_c\Big(\frac{kT}{L} - nT\Big) + v_c\Big(\frac{kT}{L}\Big).$$

We define the polyphase components of the received signal $r(k)$, channel impulse response $h(k)$ and additive noise $v(k)$ in Fig. 2.5 as

$$\begin{aligned} r_q(p) &= r(pL+q-1), \\ h_q(p) &= h_c\Big(pT+(q-1)\frac{T}{L}\Big), \\ \text{and} \quad v_q(p) &= v_c\Big(pT+(q-1)\frac{T}{L}\Big), \end{aligned}$$

where $q = 1, 2, ..., L$ and $p \in \mathbb{Z}$. For Fig. 2.5, the discrete time channel $h(k)$ of length ML and equalizer $f(k)$ of length NL can be interpreted as being divided into L sub-channels each of length M and N, so that they are decimated versions of the discrete time channel and fractionally-spaced equalizer in Fig. 2.5. The outputs of each L symbol-spaced sub-equalizers are added to provide the output $\hat{d}(n)$ at symbol rate.

Therefore, the received signal at the the input to the i-th sub-equalizer is

$$r_i(n) = \sum_{j=0}^{M-1} h_i(j)d(n-j) + v_i(n). \tag{2.12}$$

Using (2.12) we can obtain a useful expression for the input vector to the NL tap equalizer filter bank at the n^{th} symbol. Defining

$$\mathbf{r}(n) = (r_1(n), r_2(n), ..., r_L(n))^T,$$

$$\mathbf{v}(n) = (v_1(n), v_2(n), ..., v_L(n))^T,$$

the $L \times M$ channel matrix

$$\mathbf{H} = \begin{pmatrix} h_1(0) & \cdots & h_1(M-1) \\ \vdots & \cdots & \vdots \\ h_L(0) & \cdots & h_L(M-1) \end{pmatrix}$$

and the transmitted symbol vector

$$\mathbf{d}(n) = (d(n), ..., d(n-M+1))^T,$$

we get

$$\mathbf{r}(n) = \mathbf{H}\mathbf{d}(n) + \mathbf{v}(n). \tag{2.13}$$

Zero-Forcing Conditions for Fractionally Spaced Equalizers

From Fig. 2.6, assuming a noiseless condition, the overall impulse response $g(n)$ between the transmitted symbol and the baud spaced equalizer output can be written as

$$g(n) = \sum_{m=1}^{L} h_m(n) \otimes f_m(n),$$

where h_m is the m-th subchannel and f_m is the corresponding symbol spaced sub-equalizer. Taking the z-transform of both sides in the above relation we get

$$G(z) = \sum_{m=1}^{L} H_m(z) F_m(z)$$

For a perfect zero-forcing model, we require that $G(z) = z^{-\delta}$, where δ is a non-negative integer representing system delay. That is, we require that,

$$z^{-\delta} = \sum_{m=1}^{L} H_m(z) F_m(z). \tag{2.14}$$

This relation given in (2.14) is known as the Bezout relation [25], [26] and it leads directly to the perfect equalization requirements concerning sub-channel roots. That is, to attain perfect equalization using a FIR equalizer, there should not exist a common root for all the sub-channel z-transforms.

This can be easily illustrated by an example [27]. Suppose a common root does exist between the sub-channel polynomials, and let it be represented by say, $K(z) = k_0 + k_1 z^{-1}$, so that $H_m(z) = K(z)\hat{H}_m(z)$. Extracting out this common roots from all sub-channel polynomials we get

$$z^{-\delta} = (k_0 + k_1 z^{-1}) \sum_{m=1}^{L} \hat{H}_m(z) F_m(z) \tag{2.15}$$

$$= (k_0 + k_1 z^{-1}) A(z). \tag{2.16}$$

Hence there is contradiction, because there is no finite length polynomial $A(z)$ that when multiplied by $K(z)$ would result in the delay operator $z^{-\delta}$. In order to satisfy (2.15), we would require

$$A(z) = \frac{z^{-\delta}}{k_0 + k_1 z^{-1}},$$

which is an IIR system.

A fractionally spaced equalizer is a better design then a conventional symbol spaced equalizer. Not only does it simply the demodulator design by reducing restrictions on the front end filter, it also provides a robustness to variations in the sampler phase at the receiver. Furthermore, it does so at only a modest increase in implementation cost. Consequently, fractionally-spaced equalization is very popular, and has replaced the conventional symbol-spaced equalization in several applications involving a linear receiver structure [22]. However, for the same length both the fractionally-spaced and symbol-spaced equalizers have similar convergence properties with respect to mean squared error [16]. A common practice is to use a $T/2$ spaced equalizer, in which the input signal is sampled at twice the symbol rate.

2.5 Adaptive Equalization

In the design of the optimum and sub-optimum receivers discussed up until this point, it was assumed that the channel characteristics are known at the receiver. However, in a practical communication system this is seldom the case. The channel may be different for different transmissions, as in a switched telephone network where the channel is different for each new dialed number [2], or the channel may be time varying even during the signal transmission, as in many wireless channels. In many cases, the channels are dispersive, and therefore distort the transmitted signal by causing intersymbol interference. Therefore, we need equalizer structures which are self adjusting to changing channel conditions, and do not rely on apriori knowledge of the channel characteristics. Such self-adjusting receivers are called *adaptive equalizers.* In this section we discuss popular adaptive filtering approaches which are widely used in modern digital communications. There are two broad modes in which adaptive equalizers are applied. One is the block adaptive mode, and the other, more widely used mode, is the symbol-by-symbol adaptive mode. We now discuss these modes in detail.

2.5.1 Block Adaptive Equalizers

Consider the communication model in Fig. 2.3. This model was discussed earlier for a static channel, assuming that true channel characteristics are known at the receiver. The block adaptive approach to equalization can be used when the channel characteristics may be different for two separate transmissions, but remain unchanged for a given transmission. Therefore, the equalizer needs to be calculated only once per block of transmitted symbols, for which the channel remains unchanged.

In Section 2.4, an FIR MMSE equalizer was calculated in (2.8). For convenience, we will call this equalizer the Wiener Filter. Hence for the discrete-time communication model of

Fig. 2.3, an FIR MMSE Wiener Filter $\mathbf{f}_W$ of length M is calculated as,

$$\mathbf{f}_W = \mathbf{R}^{-1}\mathbf{p}, \tag{2.17}$$

where $\mathbf{R}$ is the the $M \times M$ auto-correlation matrix of the equalizer input signal $r(n)$, and $\mathbf{p}$ is the length M cross-correlation vector of the transmitted data symbols with the equalizer input. Recall, that for Fig. 2.3, if $r(n)$ is the input to the equalizer, then the equalizer input vector is defined as,

$$\mathbf{r}_n = \Big(r(n), r(n-1), r(n-2), ..., r(n-M+1)\Big)^T. \tag{2.18}$$

Then we can define the auto-correlation matrix as

$$\mathbf{R} = E\{\mathbf{r}_n\mathbf{r}_n^H\} \tag{2.19}$$

and the cross-correlation vector becomes

$$\mathbf{p} = E\{\mathbf{r}_n d^*(n-\delta)\}. \tag{2.20}$$

The Wiener filter $\mathbf{f}_W$ calculated above requires knowledge of the true channel auto-correlation matrix and cross-correlation vector at the receiver. In practice, the equalizer needs to estimate the channel characteristics in order to adjust itself to the frequency characteristics of the present channel. It is a common practice to have a training period for the equalizer to adapt itself to the channel, before actual data is transmitted. A training sequence is a finite sequence of symbols known *apriori* at the receiver. The covariance matrix of this sequence should have the same structure as that of the data (typically a scaled identity), and the length of training should at least be equal to the length of the equalizer. If N, $N \geq M$, training symbols are transmitted for a given channel, the estimates of the channel auto-correlation matrix, or the sample auto-correlation matrix for a length M input vector $\mathbf{r}_n$ (2.18) is given by,

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{i=\delta}^{N+\delta-1} \mathbf{r}_n \mathbf{r}_n^H,$$

and the sample cross-correlation vector becomes

$$\hat{\mathbf{p}} = \frac{1}{N} \sum_{i=\delta}^{N+\delta-1} \mathbf{r}_n d^*(n-\delta).$$

The resulting estimate of the FIR MMSE filter is

$$\hat{\mathbf{f}}_W = \hat{\mathbf{R}}^{-1}\hat{\mathbf{p}}.$$

Hence, by using a part of the transmitted block of symbols as training, we can estimate the Wiener Filter. Therefore, we have a block adaptive equalizer structure, which is not dependent on the knowledge of true channel characteristics, and tries to adapt itself to the channel for different transmissions, as long as the channel does not change for a given block transmission.

2.5.2 Symbol-by-Symbol Adaptive Equalizers

The symbol-by-symbol adaptive approach is more commonly used in equalizer design than the block adaptive approach. In the symbol-by-symbol adaptive case, the equalizer can track changing channel conditions and update itself with each new symbol arriving at its input. Thus, the equalizer is able to account for time-varying channels, and provide a more adaptive design as compared to the block adaptive equalizer. Again, there needs to be a training period for the equalizer, but in the case of a time-varying channel, even after the training period, the equalizer can still operate in a decision directed mode, in which the equalizer continues to update itself using its output symbols instead of the true transmitted symbols, assuming a high probability of correct decision.

Two very popular approaches to symbol-by-symbol adaptive equalization are the standard Least Mean Squares (LMS) algorithm, which has a computational complexity of $O(M)$, and the standard Recursive Least Squares (RLS) algorithm, which has a computational complexity $O(M^2)$, but has a better convergence rate then the LMS algorithm [1], [3]. Both approaches are discussed in the following sections.

LMS algorithm

Before understanding the actual LMS algorithm, it is useful to discuss the method of steepest descent. A virtue of the MSE criterion is that it is a quadratic function of the taps of the FIR linear equalizer. Therefore, the error performance surface may be thought of as an M dimensional paraboloid [1] such that its bottom represents the optimal tap weights of the FIR filter. Therefore, one may progress towards this optimal solution iteratively from an arbitrary starting point. A popular approach is to use the *steepest descent* algorithm which is based on a first-order approximation [3] of the error performance surface around the present iterate. This presents itself as a simple to implement approach and has been a widely accepted method due to its computational simplicity [2]. Now we do not have to invert the matrix to get the Wiener solution, but we can instead approach it in a step-by-step manner. As the name suggests, the taps of the FIR filter are adjusted in the direction of the steepest descent, i.e. opposite to the gradient of cost function (MSE) at the present iterate. Therefore, if $\mathbf{f}_n$ is the M tap FIR equalizer after n iterations, and $J(\mathbf{f}_n)$ is the MSE, then the relationship between successive iterates is

$$\mathbf{f}_{n+1} = \mathbf{f}_n + \frac{1}{2}\mu \mathbf{g}_n$$

where, μ is a positive constant, and $\mathbf{g}$ gives the direction in which the tap weight vector is incremented, and it is the negative gradient of the MSE at the present iterate

$$\mathbf{g}_n = -\frac{\partial J(\mathbf{f})}{\partial \mathbf{f}}\bigg|_{\mathbf{f}=\mathbf{f}_n}.$$

The positive constant μ is called the *step-size parameter*, and governs the convergence of the algorithm. For a given the auto-correlation matrix $\mathbf{R}$ and cross-correlation vector $\mathbf{p}$, the MSE is

$$J(\mathbf{f}_n) = \mathbf{f}_n^H\mathbf{R}\mathbf{f}_n - \mathbf{f}_n^H\mathbf{p} - \mathbf{p}^H\mathbf{f}_n + \sigma_d^2,$$

where σ_d^2 is the variance of the desired response $d(n)$. Therefore, the update relation of the tap weight vector can be written as

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \mu(\mathbf{R}\mathbf{f}_n - \mathbf{p}).$$

The method of steepest descent is a deterministic gradient algorithm, and iteratively calculates the Wiener Filter. However, it relies on the knowledge of the true correlation matrices, which are seldom available in a practical scenario. Therefore, we resort to the very popular *Least Mean Square* (LMS) algorithm which is a stochastic gradient algorithm. In this approach the update relation becomes,

$$\hat{\mathbf{f}}_{n+1} = \hat{\mathbf{f}}_n + \mu\mathbf{r}_n e^*(n),$$

where $\mathbf{r}_n$ is the length M input to the equalizer and $e(n)$ is the estimation error

$$e(n) = d(n) - \mathbf{f}_n^H\mathbf{r}_n.$$

The computational complexity of the algorithm is $O(M)$, due to which it is quite popular. However, it is sensitive to the eigenvalue spread in the input signal auto-correlation matrix.

Also, compared to the RLS algorithm it has a slow convergence [3].

RLS algorithm

The Recursive Least Squares (RLS) algorithm is one of the most popular adaptive filtering algorithms. Although it has a higher computational complexity $\big(O(M^2)\big)$ than the LMS algorithm, it has gained wide acceptance due to its better convergence rate, which is an order of magnitude faster than the simple LMS algorithm [3].

The idea behind the RLS algorithm is to find optimum tap weights for the FIR equalizer, that minimizes the sum of squared errors,

$$E(n) = \sum_{i=\delta}^{n} \lambda^{n-i} |d(i-\delta) - \mathbf{f}_n^H \mathbf{r}_n|^2,$$

where λ is an exponential forgetting factor $(0 < \lambda \leq 1)$. The solution to the above problem is the filter estimate

$$\hat{\mathbf{f}}_n = \boldsymbol{\Psi}_n^{-1} \mathbf{z}_n.$$

Here, $\boldsymbol{\Psi}_n$ is a time average correlation matrix of the tap-input vector $\mathbf{r}_i$,

$$\boldsymbol{\Psi}_n = \sum_{i=\delta}^{n} \lambda^{n-i} \mathbf{r}_i \mathbf{r}_i^H + \xi \lambda^n \mathbf{I},$$

where, ξ is a small positive real constant and is called the regularization parameter, and $\mathbf{I}$ is an identity matrix of appropriate dimensions. The regularizing term is required to ensure the invertibity of $\boldsymbol{\Psi}_n$ in the initial stages of the iterations when insufficient data are available. The effect of the diagonal loading dissipates with time due to the forgetting factor. The cross-correlation vector $\mathbf{z}_n$ between the tap inputs and desired response is given as

$$\mathbf{z}_n = \sum_{i=\delta}^{n} \lambda^{n-i} \mathbf{r}_i d^*(i-\delta)$$

The estimate of the FIR filter can be calculated as,

$$\hat{\mathbf{f}}_n = \boldsymbol{\Psi}_n^{-1}\mathbf{z}_n.$$

However, in practice, we avoid calculating the inverse of the correlation matrix $\boldsymbol{\Psi}_n$. Instead, we develop a recursive update relation for the estimate of tap weight vector. The calculation and update of the inverse matrix $\boldsymbol{\Psi}_n^{-1}$ is carried out using the matrix inversion lemma [3], which enables us to avoid direct matrix inversion. If $\mathbf{P}_n = \boldsymbol{\Psi}_n^{-1}$, then from the matrix inversion lemma [3],

$$\mathbf{P}_n = \lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}\mathbf{k}_n\mathbf{r}_n^H\mathbf{P}_{n-1},$$

where,

$$\mathbf{k}_n = \frac{\mathbf{P}_{n-1}\mathbf{r}_n}{\lambda + \mathbf{r}_n^H\mathbf{P}_{n-1}\mathbf{r}_n},$$

and $\mathbf{P}_0 = \zeta^{-1}\mathbf{I}$ can be used for initialization. The recursive update of the FIR equalizer is given by

$$\hat{\mathbf{f}}_n = \hat{\mathbf{f}}_{n-1} + \mathbf{k}_n\xi_n^*,$$

where ξ_n is the a priori estimation error,

$$\xi_n = d(n) - \hat{\mathbf{f}}_{n-1}^H\mathbf{r}_n.$$

and $\mathbf{f}_0 = \mathbf{0}$ can be used for initialization.

Therefore, by using the matrix inversion lemma, the RLS algorithm avoids direct calculation of the inverse of the auto-correlation matrix, and provides a recursive relation for the calculation of the tap weight vector estimate with a computational complexity of $O(M^2)$ calculations per sample. The RLS algorithm enjoys better convergence properties than the

LMS algorithm, which render it an attractive choice for adaptive filtering, even at a higher computational cost.

The discussion presented in this chapter was aimed at providing the reader with an overview of the standard algorithms and approaches for performing "unconstrained" equalization. We introduce the interesting problem of performing "constrained" equalization in the future chapters. We will discuss the multiple inequality constrained, block adaptive, fractionally spaced equalization approaches first, and then provide the theory and algorithms for performing multiple inequality constrained, symbol-by-symbol adaptive, fractionally-spaced equalization.

Chapter 3

Block Adaptive Equalization with Explicit Sidelobe Control

3.1 The Communication Model

In this chapter we develop a block adaptive, fractionally-spaced equalizer with explicit sidelobe control. We impose multiple quadratic inequality constraints, and model the problem as a Second Order Cone Program (SOCP). We then use well developed interior point methods [4] to obtain an optimal solution. We use the multirate model of Fig. 2.5 as the basic communication scheme. The oversampling factor is $L = 2$, and we assume the presence of an interferer in the sidelobe region of the equalizer frequency response. This modified communication model is shown in Fig. 3.1 consisting of a Pulse Shaping Transmit Filter $p(n)$, a

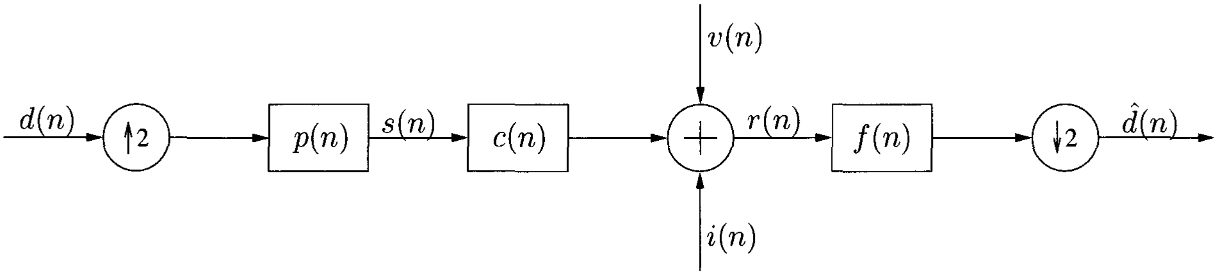

Figure 3.1: The Communication Model

dispersive channel $c(n)$, additive noise $v(n)$, adjacent channel interference (ACI) $i(n)$ and a receiver filter $f(n)$. The dispersive nature of the channel causes distortion of the transmitted pulses and gives rise to Inter Symbol Interference (ISI). The noisy and distorted input to the $T/2$ spaced equalizer is $r(n)$. A popular approach is to design $f(n)$ as a Minimum Mean squared Error (MMSE) Equalizer, which minimizes the joint effects of noise and ISI. This design may work well if ISI and noise are the only sources of symbol errors at the receiver. However, a practical system may also suffers from ACI. For example, in FDMA cellular radio systems, to enhance spectral efficiency and traffic carrying capacity, two neighboring cells might be using spectrally adjacent carrier frequencies. In such a scenario, there might be a partial overlap between the power spectral density spectrums of these two channels. This leads to Adjacent Channel Interference as the signal from the adjacent channel is seen as an interference at the present channel. Therefore robustness to intermittent adjacent channel interference needs to incorporated in the optimum filter design to ensure good performance at the receiver.

The fractionally-spaced MMSE equalizer mentioned above might have unacceptably high sidelobes in the frequency response of the equalizer. This maybe a potential source of performance degradation if unexpected interferences are present in the sidelobe region due to ACI. In order to ensure robustness to ACI, special treatment of this sidelobe region is required. Therefore, explicit sidelobe control of the frequency response of the equalizer is required. The aim here is to design an optimum receiver, such that the mean square error between the received and the desired signal is minimized and the sidelobe of the receiver baseband filter in the frequency stopband $\omega_{stop} \leq \omega \leq \pi$ (where ω denotes frequency) should be less then a prescribed level.

Let $d(n)$ be an input symbol to the communication system in Fig. 3.1 and $\hat{d}(n)$ denote the estimated symbol at the output of the receiver filter. As in (2.4), the error between the input and output symbols is

$$e(n) = \hat{d}(n) - d(n - \delta),$$

where δ is the system delay. The receiver output $\hat{d}(n)$ can be written as

$$\hat{d}(n) = \mathbf{f}^H \mathbf{r}_{2n},$$

where $\mathbf{f}$ is a vector

$$\mathbf{f} = \Big(f(0)^*, f(1)^*, f(2)^*, ..., f(M-1)^*\Big)^T,$$

containing the M tap coefficients of the FIR filter $f(n)$, and

$$\mathbf{r}_{2n} = \Big(r(2n), r(2n-1), r(2n-2), ..., r(2n-M+1)\Big)^T.$$

As shown in (2.7), the Mean Squared Error (MSE) is,

$$E\{|e(n)|\}^2 = \mathbf{f}^H \mathbf{R}_{rr} \mathbf{f} - \mathbf{f}^H \mathbf{p}_{dr} - \mathbf{p}_{dr}^H \mathbf{f} + \sigma_d^2,$$

where

$$\mathbf{p}_{dr} = E\{\mathbf{r}_{2n} d^*(n-\delta)\}$$

and

$$\mathbf{R}_{rr} = E\{\mathbf{r}_{2n} \mathbf{r}_{2n}^H\}.$$

Note that we assume the knowledge of true channel characteristics here. We shall develop our algorithm to calculate the desired optimal equalizer using these true matrices, so that it can be used as a benchmark in our simulations. This analysis can be directly applied to the block adaptive approach, as will be discussed in Section 3.6.

3.2 Sidelobe Control

The degrading effects of ACI might render an otherwise well-designed system prone to errors. Therefore, there is a need for the use of specialized techniques, tailor-made to constrain the

ACI power to a desired level of attenuation. We aim at exercising explicit control over the sidelobe region of the equalizer, so that the designer may change the sidelobe suppression at will.

In order to explicitly control the sidelobe level, we introduce quadratic inequality constraints in the sidelobe region of the FIR fractionally-spaced equalizer. These constraints guarantee that the sidelobe level is not higher then the prescribed value. We also enforce an additional constraint to put an upper bound on the norm of the FIR filter $\mathbf{f}$ to bound the output noise at the receiver. The problem of designing the M-tap equalizer that minimizes the MSE subject to these constraints can be formulated as

$$\begin{aligned} \min_{\mathbf{f}} \quad & \mathbf{f}^H \mathbf{R}_{rr} \mathbf{f} - \mathbf{f}^H \mathbf{p}_{dr} - \mathbf{p}_{dr}^H \mathbf{f} + \sigma_d^2 && (3.1\text{a}) \\ \text{subject to} \quad & |\mathbf{s}(\omega)^H \mathbf{f}| \leq \varepsilon \quad \forall \omega \in \Psi_{sl}, && (3.1\text{b}) \\ & \|\mathbf{f}\| \leq R, && (3.1\text{c}) \end{aligned}$$

where ε is the prescribed value of the sidelobe level, $\Psi_{sl} = \{\omega_{stop} \leq |\omega| \leq \pi\}$ denotes the sidelobe region on either side of the mainlobe, R is a scalar and $\mathbf{s}(\omega)$ is the vector

$$\mathbf{s}(\omega) = (1, e^{j\omega}, e^{j2\omega}, \ldots, e^{j(M-1)\omega})^T.$$

The above problem is convex, as it has a convex quadratic objective function ($\mathbf{R}_{rr}$ is positive semi-definite) and convex quadratic inequality constraints. However, the sidelobe frequency ω is considered to be continuous and hence we have an infinite number of quadratic inequality constraints. In order to approximate the above optimization problem in a convex and finite manner we can chose to implement a finite subset of the quadratic constraints by discretizing the sidelobe region. As will be shown in the following sections, the (convex) approximation of the problem in (3.1) can then be formulated as a Second Order Cone

Program (SOCP), and hence can be solved efficiently using well-established interior point methods; e.g. [4], [8].

3.3 Discretization

As discussed in the previous section, the original optimization problem (3.1) with an infinite number of quadratic inequality constraints can be approximated by one with a finite number of quadratic inequality constraints. We approximate the semi-infinite sidelobe constraint by the corresponding discretized constraints at a set of N_d frequencies in the sidelobe region Ψ_{sl}, $\omega_i \in \Psi_{sl}, i = 1, 2, \ldots, N_d$. The approximated optimization problem can be written as,

$$\min_{\mathbf{f}} \quad \mathbf{f}^H \mathbf{R}_{rr} \mathbf{f} - \mathbf{f}^H \mathbf{p}_{dr} - \mathbf{p}_{dr}^H \mathbf{f} + \sigma_d^2 \tag{3.2a}$$

$$\text{subject to} \quad |\mathbf{s}(\omega_i)^H \mathbf{f}| \leq \varepsilon \quad i = 1, 2, ..., N_d, \tag{3.2b}$$

$$||\mathbf{f}|| \leq R. \tag{3.2c}$$

Care should be taken to ensure that the chosen grid of frequencies ω_i, $i = 1, 2, ..., N_d$, is sufficiently dense in the sidelobe region Ψ_{sl}, otherwise some of the sidelobe constraints in (3.1) between two consecutive frequency points in the grid may be significantly violated. However, a large value of N_d results in a large computational complexity ($\sim O(N_d^{3.5})$) [18] . Another approach is to choose a slightly tighter prescribed sidelobe level, say $\varepsilon - \epsilon$ instead of ε. The feasible set of $\mathbf{f}$ for problem (3.2) shrinks for a tighter sidelobe level, therefore we must be careful to make sure that ϵ/ε is sufficiently small. There is generally a trade-off between the number of discretizations N_d and the reduction in sidelobe level ϵ. As N_d is increased ϵ can be reduced. In practice, the prescribed sidelobe level is not altered when N_d is sufficiently large ($N_d \approx 15M$) [6].

3.4 Second Order Cone Programming (SOCP)

Each quadratic inequality constraint in (3.2) can be imposed by modeling it as a Second Order Cone (SOC) constraint. An n-dimensional Second Order Cone (also known as a quadratic cone or Lorentz cone is defined as [34],

$$SOC^n := \{(x_1, \mathbf{x}_2 \in \mathcal{R} \times \mathcal{C}^{n-1} | x_1 \geq ||\mathbf{x}_2||\},$$

where $||\cdot||$ denotes the Euclidean norm. We intend to rewrite (3.2) in the dual standard form of a convex cone optimization problem, namely

$$\max_{\mathbf{y}} \mathbf{b}^T\mathbf{y} \qquad \textit{subject to} \qquad \mathbf{c} - \mathbf{A}^T\mathbf{y} \in \mathcal{K}, \tag{3.3}$$

where $\mathcal{K}$ is a symmetric cone [1], $\mathbf{y}$ is a variable vector, $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{A}$ are arbitrary vectors and a matrix of appropriate dimensions, all of which can be complex valued. Since all the constraints in (3.2) can be represented by Second Order Cones, we now reformulate our optimization problem as a Second Order Cone Programming (SOCP) problem.

3.5 Reformulation using SOCP

We re-write the quadratic objective function of (3.2) in a more suitable form as

$$\mathbf{f}^H\mathbf{R}_{rr}\mathbf{f} - \mathbf{f}^H\mathbf{p}_{dr} - \mathbf{p}_{dr}^H\mathbf{f} + \sigma_d^2 = ||\mathbf{L}\mathbf{f}_1||^2,$$

[1] "A symmetric cone is a Cartesian product of a non-negative orthant, quadratic cones and cones of positive semidefinite matrices", [4] where each elementary cone corresponds to a constraint of the optimization problem.

where the $(M+1) \times (M+1)$ upper triangular matrix $\mathbf{L}$ is a Cholesky factor of $\mathbf{Q}$ (i.e. $\mathbf{L}^H\mathbf{L} = \mathbf{Q}$), where

$$\mathbf{Q} = \begin{bmatrix} \mathbf{R} & -\mathbf{p} \\ -\mathbf{p}^H & \sigma_d^2 \end{bmatrix}$$

and

$$\mathbf{f}_1 = \begin{bmatrix} \mathbf{f} & 1 \end{bmatrix}^T.$$

Therefore, minimizing $||\mathbf{L}\mathbf{f}_1||$ is equivalent to minimizing the cost function in (3.2). We further write $\mathbf{L} = [\mathbf{L}_1, \mathbf{L}_2]$, where $\mathbf{L}_1$ is an $(M+1) \times M$ matrix and $\mathbf{L}_2$ is an $(M+1) \times 1$ vector; hence $||\ \mathbf{L}\mathbf{f}_1\ ||$ can be written as $||\mathbf{L}_1\mathbf{f} + \mathbf{L}_2||$. We now introduce a non-negative scalar τ and a new constraint $||\mathbf{L}_1\mathbf{f} + \mathbf{L}_2|| \leq \tau$, so that the original problem in (3.2) becomes,

$$\min_{\tau,\mathbf{f}} \quad \tau \tag{3.4a}$$

$$\text{subject to} \quad ||\mathbf{L}_1\mathbf{f} + \mathbf{L}_2|| \leq \tau \tag{3.4b}$$

$$|\mathbf{s}(\omega_i)^H\mathbf{f}| \leq \varepsilon \quad i = 1, 2, ..., N_d \tag{3.4c}$$

$$||\mathbf{f}|| \leq R. \tag{3.4d}$$

Now we will reformulate (3.4) as the SOCP program in (3.3). We define an $(M+1) \times 1$ dimensional variable vector $\mathbf{y}$, such that

$$\mathbf{y} = \begin{bmatrix} y_1 & \mathbf{y}_2^T \end{bmatrix}^T$$

where $y_1 = \tau$ and $\mathbf{y}_2 = \mathbf{f}$. We define an $(M+1) \times 1$ vector $\mathbf{b}$,

$$\mathbf{b} = [-1, 0, \cdots, 0]^T,$$

and hence

$$-y_1 = \mathbf{b}^T\mathbf{y}.$$

Therefore, the problem in (3.4) becomes

$$\max_{\mathbf{y}} \quad \mathbf{b}^T\mathbf{y} \tag{3.5a}$$

$$\text{subject to} \quad ||\mathbf{L}_1\mathbf{f} + \mathbf{L}_2|| \leq \; y_1 \tag{3.5b}$$

$$|\mathbf{s}(\omega_i)^H\mathbf{y}_2| \leq \varepsilon \quad i = 1, 2, ..., N_d \tag{3.5c}$$

$$||\mathbf{y}_2|| \leq R. \tag{3.5d}$$

Now we formulate the constraints in (3.5) as second order cone constraints. The constraint $||\mathbf{L_1}\mathbf{f} + \mathbf{L_2}|| \leq \; y_1$ takes the form

$$\begin{pmatrix} y_1 \\ \mathbf{L}_1\mathbf{y}_2 + \mathbf{L}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{L}_2 \end{pmatrix} - \begin{pmatrix} -1 & \mathbf{0} \\ \mathbf{0} & -\mathbf{L}_1 \end{pmatrix} \mathbf{y} \equiv \mathbf{c}_1 - \mathbf{A}_1^T\mathbf{y} \in SOC^{M+2}. \tag{3.6}$$

Similarly, $|\mathbf{s}(\omega_i)^H\mathbf{y}_2| \leq \varepsilon$ becomes

$$\begin{pmatrix} \varepsilon \\ \mathbf{s}(\omega_i)^H\mathbf{y}_2 \end{pmatrix} = \begin{pmatrix} \varepsilon \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & \mathbf{0} \\ 0 & -\mathbf{s}(\omega_i)^H \end{pmatrix} \mathbf{y} \equiv \mathbf{c}_{1+i} - \mathbf{A}_{1+i}^T\mathbf{y} \in SOC^2 \tag{3.7}$$

for $i = 1, \ldots, N_d$, and $||\mathbf{y}_2|| \leq R$ can be written as

$$\begin{pmatrix} R \\ \mathbf{y}_2 \end{pmatrix} = \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} 0 & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{pmatrix} \mathbf{y} \equiv \mathbf{c}_{N_d+2} - \mathbf{A}_{N_d+2}^T\mathbf{y} \in SOC^{M+1}. \tag{3.8}$$

Now we define a vector $\mathbf{c}$ and a matrix $\mathbf{A}^T$ so that

$$\mathbf{c} := \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_{N_d+1} \\ \mathbf{c}_{N_d+2} \end{bmatrix}, \qquad \mathbf{A}^T := \begin{bmatrix} \mathbf{A}_1^T \\ \mathbf{A}_2^T \\ \vdots \\ \mathbf{A}_{N_d+1}^T \\ \mathbf{A}_{N_d+2}^T \end{bmatrix}. \tag{3.9}$$

Hence, from (3.9), the optimization problem in (3.5) can be expressed as

$$\max_{\mathbf{y}} \mathbf{b}^T\mathbf{y} \qquad \textit{subject to} \qquad \mathbf{c} - \mathbf{A}^T\mathbf{y} \in \mathcal{K}, \tag{3.10}$$

where $\mathcal{K}$ is the symmetric cone corresponding to the constraints, i.e., $\mathcal{K}$ is a Cartesian product of the elementary cones,

$$\mathcal{K} := SOC^{M+2} \times \underbrace{SOC^2 \times \cdots \times SOC^2}_{N_d} \times SOC^{M+1}.$$

This indicates that the first $(M+2)$ terms of the vector $\mathbf{c}-\mathbf{A}^T\mathbf{y}$ lie in the $(M+2)$ dimensional SOC, next N_d pairs each lie in the 2 dimensional (complex) SOC, and the last $(M+1)$ terms belong to the $(M+1)$ dimensional Second Order Cone. Thus, we have reformulated the original optimization problem in (3.2) into a Second Order Cone Program which can be efficiently solved by using interior point methods e.g. [8]. A convenient implementation of these methods is the SeDuMi tool [4]. If the convex optimization problem is feasible, the optimal solution for $\mathbf{y}$ yields the optimal equalizer for the system.

3.6 Block Adaptive Equalizer with Explicit Sidelobe Control

The analysis in the preceding sections of this chapter assumed prior knowledge of the true channel characteristics. However, this is seldom the case. Recall from Section 2.5.1, that in a scenario where the channel characteristics remain unchanged during a transmission, but may be different for two separate transmissions, a block adaptive approach to estimate the optimal (FIR) equalizer can be adopted. As in the unconstrained case, for the present problem of constrained equalization, the auto-correlation matrix $\mathbf{R}_{rr}$ of the equalizer inputs and cross-correlation vector $\mathbf{p}_{dr}(n)$ can be estimated using a finite length training sequence. If N, $N \geq M$, training symbols are transmitted for a given channel, the estimates of the channel auto-correlation matrix, or the sample auto-correlation matrix for a length M input vector $\mathbf{r}_{2n}$ is given by,

$$\hat{\mathbf{R}}_{rr} = \frac{1}{N} \sum_{i=\delta}^{N+\delta-1} \mathbf{r}_{2n}\mathbf{r}_{2n}^{H},$$

and the sample cross-correlation vector becomes

$$\hat{\mathbf{p}}_{dr} = \frac{1}{N} \sum_{i=\delta}^{N+\delta-1} \mathbf{r}_{2n} d^{*}(n-\delta).$$

Hence, the problem (3.2) becomes

$$\min_{\mathbf{f}} \quad \mathbf{f}^{H}\hat{\mathbf{R}}_{rr}\mathbf{f} - \mathbf{f}^{H}\hat{\mathbf{p}}_{dr} - \hat{\mathbf{p}}_{dr}^{H}\mathbf{f} + \sigma_d^2 \tag{3.11a}$$

$$\text{subject to} \quad |\mathbf{s}(\omega_i)^{H}\mathbf{f}| \leq \varepsilon \quad i = 1, 2, ..., N_d, \tag{3.11b}$$

$$||\mathbf{f}|| \leq R, \tag{3.11c}$$

which can be implemented using the same analysis provided in Section 3.5, to yield an estimate of the optimal filter.

Chapter 4

Interior Point Methods for Convex Inequality Constrained Optimization Problems

4.1 Introduction

Interior Point Methods have recently emerged as a powerful tool for solving a host of convex optimization problems, including those with multiple convex quadratic inequality constraints. In this chapter we discuss the use of interior point methods for solving convex optimization problems with multiple inequality constraints and analyse some of the prevalent approaches. The underlying idea behind all these approaches is that a solution to the original optimization problem can be approached by solving a sequence of unconstrained problems which are easier to solve. In particular, we will discuss the "barrier method" approach and the concept of the central path. An alternate approach that uses the concept of "analytic centers" is also presented. The application of both these approaches to solve the dynamic problem of performing symbol-by-symbol constrained adaptive equalization with quadratic inequalities will be dealt with in the next chapter.

4.2 Standard Barrier Method

The standard form of the barrier method is discussed in detail in [8], on which the following discussion based. Consider the following generic convex optimization problem with inequality constraints,

$$\min \quad f_0(\mathbf{x}) \tag{4.1a}$$

$$\text{subject to} \quad f_i(\mathbf{x}) \leq 0 \quad i = 1, ..., m \tag{4.1b}$$

where $f_0, ..., f_m : \mathcal{C}^n \rightarrow \mathcal{R}$ are convex and twice continuously differentiable. To simplify the analysis, we will assume that there exists a strictly feasible point $\mathbf{x}_0$; i.e. there exists an $\mathbf{x}_0$ such that

$$f_i(\mathbf{x}_0) < 0 \quad i = 1, ..., m.$$

The set of all strictly feasible points will be represented by the set

$$\Phi = \{\mathbf{x} | f_i(\mathbf{x}) < 0, i = 1, ..., m\}.$$

A "barrier function" for the set Φ is a (smooth convex) function that tends to infinity at the boundary of the set. A good choice of a barrier function for (4.1) is the *logarithmic barrier function*,

$$\phi(\mathbf{x}) = -\sum_{i=1}^{m} \log(-f_i(\mathbf{x})). \tag{4.2}$$

Not only is its domain the set Φ defined above, the function $\phi(\mathbf{x})$ also grows without bound as $\mathbf{x}$ approaches the boundary of this set.

The logarithmic barrier function can be used to approximate the original optimization problem (4.1) by the unconstrained optimization problem

$$\min \; t f_0(\mathbf{x}) + \phi(\mathbf{x}), \tag{4.3}$$

where t is a non-negative parameter that gives a measure of the accuracy of the approximation. As the parameter t grows, the approximation becomes more accurate. The function $\phi(\mathbf{x})$ is convex and differentiable, hence (4.3) is also a smooth convex optimization problem. Note that we have "lifted" the non-linear constraints of the original optimization problem in (4.1) into the objective in (4.3) using the logarithmic barrier function. The standard barrier methods approach a solution to original problem in (4.1) by solving a sequence of unconstrained problems of the form in (4.3). For a given value of t, the unconstrained optimization problem in (4.3) has a unique solution that can be found using standard methods for solving convex unconstrained problems, such as Newton's Method. While solving this sequence of unconstrained minimization problems, the solution to the previous problem is taken as the starting point for the Newton's Procedure in the present problem. The rate of update of the parameter t from one unconstrained problem to the next determines the aggressiveness of the barrier method.

We call the set of solutions of (4.3) for non-negative values of t the *central path*, and we let $\mathbf{x}^\star(t)$ denote the *central point* on that path corresponding to the given value of t. The point $\mathbf{x}^\star(0)$ is the *analytic center* of the feasible set Φ (see Section 4.3) and the goal of a barrier method is to successively increase t and to solve (4.3) for the next central point so that we follow the central path to an optimal solution. Each central point $\mathbf{x}^\star(t)$ is a strictly feasible point, (i.e., $f_i(\mathbf{x}^\star(t)) < 0$, $i = 1, ..., m$) and since it is the minimizer of (4.3), we have

$$t\nabla f_0(\mathbf{x}^\star(t)) + \nabla\phi(\mathbf{x}^\star(t)) = 0. \tag{4.4}$$

From the definition of $\phi(\mathbf{x})$ in (4.2), we have that

$$\nabla\phi(\mathbf{x}^\star(t)) = \sum_{i=1}^{m} \frac{-1}{f_i(\mathbf{x}^\star(t))}\nabla f_i(\mathbf{x}^\star(t)). \tag{4.5}$$

Using (4.5) in (4.4) we get the relation,

$$\nabla f_0(\mathbf{x}^\star(t)) + \sum_{i=1}^{m} \frac{-1}{tf_i(\mathbf{x}^\star(t))} \nabla f_i(\mathbf{x}^\star(t)) = 0. \tag{4.6}$$

Defining,

$$\lambda_i^\star(t) = \frac{-1}{tf_i(\mathbf{x}^\star(t))}, \tag{4.7}$$

where for $t > 0$, the fact that $f_i(x^\star(t)) < 0$, $i = 1, ..., m$, implies that

$$\lambda_i^\star(t) > 0 \quad i = 1, ..., m. \tag{4.8}$$

Using (4.7) in (4.6) we have that

$$t\nabla f_0(\mathbf{x}^\star(t)) + \sum_{i=1}^{m} \lambda_i^\star(t) \nabla f_i(\mathbf{x}^\star(t)) = 0. \tag{4.9}$$

From (4.9) we observe that the central point $x^\star(t)$ minimizes the Lagrangian of the primal problem (4.1)

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i(t) f_i(\mathbf{x}).$$

From (4.8) we know also that $\lambda_i^\star(t) > 0$. Therefore we may conclude that $\boldsymbol{\lambda}^*(t)$ (where the i^{th} element of $\boldsymbol{\lambda}^*(t)$ is $\lambda_i^\star(t)$) is a dual feasible point. We define the dual function $g(\boldsymbol{\lambda}^*(t))$ to be the minimum value of the Lagrangian over $\mathbf{x}$. That is

$$g(\boldsymbol{\lambda}^\star(t)) = f_0(\mathbf{x}^\star(t)) + \sum_{i=1}^{m} \lambda_i^\star(t) f_i(\mathbf{x}^\star(t)) \tag{4.10}$$

$$= f_0(\mathbf{x}^\star(t)) - m/t.$$

Hence an important property of the central path is that for each central point there exists a dual function, and hence there exists a lower bound on the optimal value $p^\star$ [8], where $p^\star$ denotes the optimal value of the objective of the primal problem given in (4.1). From (4.10) we see that for a given value of t, the duality gap between the primal and dual solutions is m/t. Since the dual function $g(\boldsymbol{\lambda})$ yields a lower bound on the optimal value $p^\star$ of the primal problem (4.1), we can say that,

$$f_0(\mathbf{x}^\star(t)) - p^\star \leq m/t,$$

i.e., "$\mathbf{x}^\star(t)$ is no more than m/t suboptimal" [8]. This supports the fact the parameter t is gives us a measure of the accuracy of the approximation in problem in (4.1). When $\mathbf{x}$ is strictly feasible, we see that as $t \to \infty$, the duality gap becomes vanishingly small and the central path converges to an optimal solution of the original optimization problem in (4.1).

From the above discussion, we know that for a given value of t the barrier method provides us with an m/t suboptimal solution, and that a certificate of accuracy (a lower bound on $p^\star$) is provided by the dual feasible point $\boldsymbol{\lambda}^\star(t)$. The barrier method involves the solution of a sequence of unconstrained minimization problems with increasing values of t, which can be solved using a popular technique like the Newton procedure, with the solution of the previous problem as a starting point for the present problem, to arrive at an ϵ suboptimal solution of the original problem in (4.1). This approach is called the *Sequential Unconstrained Minimization Technique* (SUMT) [8].

In the algorithm described in Table 4.1, each *centering step* corresponds to an *outer* iteration in which a new point on the central path is calculated using the previously calculated point on the central path as the starting point. Each centering step may require several Newton steps, or *inner iterations* to arrive at the optimal solution for a given value of t. The algorithm is designed to follow the central path to approach an optimal solution of problem in (4.1) as $t \to \infty$. Hence it is also called a *path following method.*

The rate at which the parameter t increases defines the behavior of the algorithm. This is so because the objective function in (4.3) is dependent on t, as the parameter t decides the relative importance of $f_0(\mathbf{x})$ and the barrier term $\phi(\mathbf{x})$. If t is updated conservatively,

Table 4.1: Sequential Unconstrained Minimization Technique (SUMT)

•**Step 1: Initialization.** Let $m, \mathbf{x}_0, t_0 > 0, \mu > 1, \epsilon$ be given. Set

$$\mathbf{x} := \mathbf{x}_0, \quad t := t_0,$$

where $\mathbf{x}_0 \in \Phi$ is a strictly feasible point. One possible value for $\mathbf{x}_0$ is $\mathbf{x}^\star(0)$.

•**Step 2: Centering Step.** The new central point $\mathbf{x}^\star(t)$ is obtained by solving the unconstrained problem

$$\min t f_0(\mathbf{x}) + \phi(\mathbf{x})$$

with $\mathbf{x}$ as the starting point.

•**Step 3: Updating.** Set

$$\mathbf{x} := \mathbf{x}^\star(t), \quad t := \mu t.$$

•**Stopping Criterion.** QUIT if $m/t < \epsilon$, else RETURN to **Step2**.

i.e., with each outer iteration t increases only by a small value, then the previous iterate is a very good starting point for the Newton procedure, and a small number of Newton steps (inner iterations) would be sufficient for precise centering, i.e., calculation of the next iterate. In contrast if t is increased aggressively, then the problem is changing very fast, and the previous iterate might not be a very good starting point, thus requiring more inner iterations.

The barrier method described above is a standard approach for solving inequality constrained convex optimization problems. In the next section we discuss an alternate approach for efficiently solving the same problem by using the concept of *analytic centers.*

4.3 Method of Analytic Centers

The method of analytic centers is an alternate approach to the barrier method described in the previous section, based on another parametrization of the central path. In this approach the driving force of the algorithm is to find a "central" feasible point in a set defined by convex inequalities. The idea is to include the original objective function $f_0(x)$ into the constraints and define a convex feasibility region Ω_n. One measure of the "center" of this convex feasibility region is called the *analytic center* of Ω_n.

Recall that the original optimization problem in (4.1) was

$$\min \quad f_0(\mathbf{x}) \tag{4.11a}$$

$$\text{subject to} \quad f_i(\mathbf{x}) \leq 0 \quad i = 1, ..., m \tag{4.11b}$$

where $f_0, ..., f_m : \mathcal{C}^n \rightarrow \mathcal{R}$ are convex and twice continuously differentiable. To formulate this problem as a sequence of convex feasibility problems, we define a convex feasibility region,

$$\Omega = \{\mathbf{x} \in \mathcal{C}^n | f_0(\mathbf{x}) \leq \tau, f_1(\mathbf{x}) \leq 0, \ldots, f_m(\mathbf{x}) \leq 0\}, \tag{4.12}$$

where τ is a scalar, which may be determined adaptively, or could remain constant depending on the approach. The inequality constrained convex optimization problem in (4.11) can be modeled as the problem of finding a feasible point which satisfys all the convex quadratic inequalities in (4.12). Now we define the following *potential function* on Ω,

$$\psi(\mathbf{x}) = -\log(\tau - f_0(\mathbf{x})) - \rho \sum_{i=1}^{m} \log(-f_i(\mathbf{x})), \tag{4.13}$$

where ρ is a non-negative scalar. The function $\psi(\mathbf{x})$ is a *logarithmic barrier function*, and is convex and differentiable. We assume that the minimizer of $\psi(\mathbf{x})$ over Ω exits and is unique. A point in the deep interior of Ω will have a low potential, whereas towards the boundary the potential tends towards infinity. The unique minimizer of this potential function is called the *analytic center* of Ω, and hence the name of the algorithm. It is important to note here that the analytic center is different from the geometric center of Ω. Whereas the geometric center depends only on the geometric shape of Ω, the analytic center depends on the algebraic representation of Ω as well. A desirable property of the analytic center is that it is reasonably easy to compute. In contrast, the geometric center can be difficult to compute. We discuss two approaches to find the optimal solution of the original optimization problem in (4.11) using the concept of analytic centers. The idea is to solve a sequence of convex feasibility problems with either τ or ρ as the parameter that drives the algorithm towards an optimal solution of problem in (4.11).

4.3.1 Method of Analytic Centers-1 (MAC1)

In this approach we let the parameter τ drive the algorithm to the optimal solution, i.e., τ is made adaptive, whereas the parameter $\rho = 1$. Therefore, the potential function defined on the convex feasibility region Ω is,

$$\psi(\mathbf{x}) = -\log(\tau - f_0(\mathbf{x})) - \sum_{i=1}^{m} \log(-f_i(\mathbf{x})).$$

The unique minimizer of this potential function is called the analytic center of Ω. The set of analytic centers of each of these convex feasibility problems for different values of τ form the central path. By solving a sequence of convex feasibility problems with decreasing values of τ the algorithm follows the central path from a feasible starting point to an optimal solution of problem in (4.11). We use the previous analytic center as a starting point for the Newton procedure to arrive at the analytic center of the present problem. This is called the recentering step or an *outer* iteration, and the Newton steps needed to arrive at the new analytic center are called the *inner* iterations. The mathematical analysis of this approach is subject to the definition of τ which in turn is dependent on the nature of the problem at hand.

An interesting analogy between the Standard Barrier Method described in the previous section, and MAC1 exists. Both are used to solve the same optimization problem given in (4.11). Let $\mathbf{x}^\star(t)$ denote a central point for the Standard Barrier Method for a given value of t ($t > 0$) which minimizes (4.3), and let $\mathbf{z}^\star(\tau)$ denote the analytic center of Ω for $\tau > p^\star$, where $p^\star$ is the optimal value of the problem (4.9). "For each $\tau > p^\star$, there is a $t > 0$ for which $\mathbf{x}^\star(t) = \mathbf{z}^\star(\tau)$, and conversely, for each $t > 0$, there is a $\tau > p^\star$ for which $\mathbf{z}^\star(\tau) = \mathbf{x}^\star(t)$" [8]. Hence, we can deduce that the central path formed by the analytic centers in MAC1 is another parametrization of the central path formed by the central points in the Standard Barrier Method.

The success of this scheme is dependent on an appropriate definition of τ for each problem. Instead of having to redefine τ for each new problem, we can use a different approach based on the method of analytic centers. This approach is discussed in the next section.

4.3.2 Method of Analytic Centers-2 (MAC2)

In the second approach, we let τ remain a constant, and decrement ρ with each iteration. The unique minimum of $\psi(\mathbf{x})$ over Ω is called the *ρ-analytic center* of Ω. The function $\psi(\mathbf{x})$ is governed by the relative importance of the objective function term and the inequality constraint terms. For each value of ρ a convex feasibility problem is solved, and the solution (called a ρ-analytic center) is used as the starting point for the Newton procedure in the next problem. As $\rho \longrightarrow 0$, we approach an optimal solution via a central path formed by all the ρ-analytic centers calculated for decreasing values of ρ. Let $\tilde{\mathbf{x}}^\star(\rho)$ be a ρ-analytic center. Therefore, it satisfys $\nabla\psi(\tilde{\mathbf{x}}^\star(\rho)) = 0$. Hence we have that

$$\nabla f_0(\tilde{\mathbf{x}}^\star(\rho)) + \rho \sum_{i=1}^{m} \frac{-(\tau - f_0(\tilde{\mathbf{x}}^\star(\rho)))}{f_i(\tilde{\mathbf{x}}^\star(\rho))} \nabla f_i(\tilde{\mathbf{x}}^\star(\rho)) = 0. \tag{4.14}$$

Now we define $\tilde{\lambda}_i^\star(\rho)$ such that,

$$\tilde{\lambda}_i^\star(\rho) = \frac{-\rho(\tau - f_0(\tilde{\mathbf{x}}^\star(\rho))}{f_i(\tilde{\mathbf{x}}^\star(\rho))}. \tag{4.15}$$

Since $\rho > 0$ and since the ρ-analytic center $\tilde{\mathbf{x}}^\star(\rho)$ belongs to the convex feasibility region Ω in (4.12), we have that

$$\tilde{\lambda}_i^\star(\rho) > 0 \quad i = 1, ..., m.$$

Using (4.15) in (4.14), we have that

$$\nabla f_0(\tilde{\mathbf{x}}^\star(\rho)) + \sum_{i=1}^{m} \tilde{\lambda}_i^\star(\rho) \nabla f_i(\tilde{\mathbf{x}}^\star(\rho)) = 0. \tag{4.16}$$

However, from (4.16) we observe that the ρ-analytic center $\tilde{\mathbf{x}}^\star(\rho)$ would minimize the Lagrangian for the primal problem in (4.11),

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x),$$

for $\boldsymbol{\lambda} = \tilde{\boldsymbol{\lambda}}^{\star}(\rho)$, i.e., $\tilde{\boldsymbol{\lambda}}^{\star}(\rho)$ is a dual feasible point. We define a dual function $g(\tilde{\boldsymbol{\lambda}}^{\star}(t))$ to be the minimum value of the Lagrangian over $\mathbf{x}$. That is,

$$g(\tilde{\boldsymbol{\lambda}}^{\star}(\rho)) = f_0(\tilde{\mathbf{x}}^{\star}(\rho)) + \sum_{i=1}^{m} \tilde{\lambda}_i^{\star}(\rho) f_i(\tilde{\mathbf{x}}^{\star}(\rho)) \tag{4.17a}$$

$$= f_0(\tilde{\mathbf{x}}^{\star}(\rho)) - m\rho(\tau - f_0(\tilde{\mathbf{x}}^{\star}(\rho)), \tag{4.17b}$$

where τ is a constant, and $f_0(\tilde{\mathbf{x}}^{\star}(\rho))$, the value of original objective function at the ρ-analytic center is measurable for each ρ. For simplicity, we define,

$$\Delta_{\rho,\tau} = \rho(\tau - f_0(\tilde{\mathbf{x}}^{\star}(\rho)). \tag{4.18}$$

The duality gap between the primal and dual solution is $m\Delta_{\rho,\tau}$, where $\Delta_{\rho,\tau}$ is a measurable quantity, and m is given. Also, since the dual function $g(\tilde{\boldsymbol{\lambda}})$ is such that,

$$g(\tilde{\boldsymbol{\lambda}}) \leq p^{\star}$$

we have from (4.15),

$$f_0(\tilde{\mathbf{x}}^{\star}(\rho)) - p^{\star} \leq m\Delta_{\rho,\tau}, \tag{4.19}$$

i.e., $\tilde{\mathbf{x}}^{\star}(\rho)$ is no more than $m\Delta_{\rho,\tau}$ suboptimal. Therefore, we see that the accuracy of the solution via this approach using the method of analytic centers depends on the choice of constant τ.

As the non-negative parameter ρ is decreased with each iteration, the duality gap also reduces, and we approach an optimal solution of problem in (4.11) as $\rho \to 0$, irrespective of the value of the constant τ. Therefore, we see that it is possible to solve an inequality constrained convex optimization problem by solving a sequence of convex feasibility problems. The Newton procedure is used to arrive at the analytic center for each new convex feasibility problem using the analytic center for the previous problem as the starting point.

Table 4.2: Method of Analytic Centers-2 (MAC2)

•**Step 1: Initialization.** Let $m, \tilde{\mathbf{x}}_0 \in \Omega, \rho_0 > 0, \beta < 1, \epsilon$ and τ be given. Set

$$\tilde{\mathbf{x}} := \tilde{\mathbf{x}}_0, \quad \rho := \rho_0,$$

•**Step 2: Recentering.** Solve

$$\min \quad -\log(\tau - f_0(\tilde{\mathbf{x}})) - \rho \sum_{i=1}^{m} \log(-f_i(\tilde{\mathbf{x}}))$$

with $\tilde{\mathbf{x}}$ as the starting point.

•**Step 3: Updating.** Set

$$\tilde{\mathbf{x}} := \tilde{\mathbf{x}}^\star(\rho),$$

$$\rho := \beta\rho,$$

$$\Delta_{\rho,\tau} = \rho(\tau - f_0(\mathbf{x}^\star(\rho))$$

•**Stopping Criterion.** QUIT if $m\Delta_{\rho,\tau} < \epsilon$, else RETURN to **Step2**.

The accuracy of the analytic center with respect to the original optimization problem in (4.11) is given by $m\Delta_{\rho,\tau}$ as proved earlier in (4.19). Hence, an algorithm which provides us with a guaranteed specified accuracy ϵ for a constant τ and decreasing ρ, is given in Table 4.2. Each recentering step is an *outer* iteration, and may require several *inner* iterations (Newton steps), starting from the analytic center of the previous convex feasibility problem. The constant β ($0 < \beta < 1$) is a measure of the aggressiveness of the algorithm, a small value indicating a more aggressive approach towards the optimal solution of the inequality constrained convex optimization problem in (4.11). As stressed earlier, a wise choice of the constant τ would improve the performance of the algorithm. However, eventual convergence of the algorithm to an optimal solution as $\rho \to 0$ is independent of τ.

In the next chapter, we develop practically implementable algorithms based on the Standard Barrier Method, and Method of Analytic Centers for performing, quadratic inequality constrained, symbol-by-symbol adaptive, fractionally-spaced equalization.

Chapter 5

Adaptive equalization with explicit sidelobe control using interior point methods

5.1 Introduction

The problem of performing adaptive equalization with strict sidelobe control control was described in the Chapter 3. This problem was reformulated as a Second Order Cone Programming (SOCP) problem, which can be solved using standard interior point methods, such as that in the SeDuMi tool [4]. The method however had its limitations because it could only perform block adaptive equalization. In this chapter, we develop algorithms based on interior point methods for performing fractionally-spaced symbol-by-symbol adaptive equalization with explicit sidelobe control. This is important because it enables us to perform symbol by symbol adaptive equalization while maintaining strict control on the sidelobe region of the equalizer. Therefore it renders itself useful in applications where the filter needs to be updated adaptively with each new incoming symbol, for example in the case of a time varying channel with ACI in the sidelobe region. The algorithms developed in this chapter are practically implementable and are based on the theory presented in the previous chapter.

We write the problem of minimizing the mean squared error with explicit sidelobe control as a convex optimization problem with multiple inequality constraints. We discuss the application of the Sequential Unconstrained Minimization Technique (SUMT) and Method of analytic centers (MAC) approaches to solve the problem in a symbol-by-symbol fashion. The changing channel conditions and practical considerations like computational cost encourage us to make adjustments and approximations to the algorithm, which will be discussed later in the chapter.

In this chapter, we also provide a discussion of the Interior Point Least Squares (IPLS) algorithm proposed by Afkhamie *et al.* [10]. When applied to unconstrained problems, the IPLS algorithm enjoys a better transient performance than the RLS algorithm [10]. This is due to the implicit adaptive diagonal loading in the IPLS algorithm which provides robustness to variations in the initialization of the algorithm. IPLS has a slightly higher computational complexity than RLS, namely $O(M^{2.2})$ operations per symbol, as compared to $O(M^2)$. Afkhamie *et al.* briefly discussed the use of IPLS algorithms in solving adaptive filtering problems with linear equality constraints [11]. They reported a computational complexity of $O(M^3)$ operations per symbol. We analyse the applicability of this approach to inequality constrained optimization problems and demonstrate that the structure of the IPLS algorithm is not directly applicable to inequality constrained optimization scenarios. The inability of the IPLS algorithm to directly solve inequality constrained convex optimization problems is the primary motivation for this thesis. We provide algorithms (IPM2 and SBM) in this chapter (and their detailed mathematical analysis) which are successfully applied to such problems.

The concept of central path and related parameters is of fundamental importance in the success of the iterative interior point algorithms. We have provided discussion and mathematical analysis on these concepts in the last chapter. In the next section we apply the Method of Analytic centers to solve our constrained adaptive equalization problem. In the following section we solve the same problem using an approximation of the standard barrier method. Lastly we discuss the computational complexity of these algorithms, and its comparison with standard RLS algorithms.

5.2 Analytic Center based estimators

The Method of Analytic Centers has been discussed in detail in the previous chapter. A convex feasibility region is defined which encompasses all the inequality constraints as well as the objective function. Then we take multiple Newton steps from a starting point to arrive at a unique minimizer of a potential function defined on this convex feasibility region. This unique minimizer is called the analytic center of the convex feasibility region, and is used as the starting point for the next convex feasibility problem. Each centering step requires one or more Newton steps to arrive at the optimal solution. In order to closely follow the central path, one must perform several Newton steps (inner iterations) so that one obtains a sufficiently *accurate* minimizer of the problem. However, calculating a *good* minimizer instead of an *accurate* minimizer can substantially reduce the computational cost. Therefore, in a practical adaptive filtering scenario, a reasonable approach is to aim for a good minimizer while taking only one Newton step. Therefore, in all future discussions, it will be assumed that only one Newton step is taken per outer iteration.

As discussed in the previous chapter, there are two approaches for using the Method of Analytic Centers for solving inequality constrained optimization problems, namely MAC1 and MAC2. Afkhamie *et al.* [11] proposed two versions of the IPLS algorithm, namely IPM1 and IPM2, to solve Adaptive Filtering problems with linear equality constraints. In the light of the Method of Analytic centers discussed in the previous chapter, we modify IPM1 and IPM2 to accommodate quadratic inequality constraints as well. The first approach is called the Interior Point Method−1 (IPM1). It uses a parameter associated with the original objective function to follow the desired central path whereas the second approach, called

Interior Point Method−2 (IPM2), uses a parameter with the constraint functions to drive the algorithm towards an optimal solution. Both approaches are described in detail in the following sections and their limitations (if any) are discussed as well.

5.2.1 Interior Point Method-1 (IPM1)

In an attempt to perform symbol-by-symbol adaptive equalization with multiple quadratic constraints, we introduce the IPM1 approach, based on the principles of the MAC1 approach discussed in the previous chapter.

The quadratically constrained optimization problem described in Chapter 3 can be re-written as

$$\min_{\mathbf{f}} \quad J_n(\mathbf{f}) \tag{5.1a}$$

$$\text{subject to} \quad |\mathbf{s}(\omega_i)^H\mathbf{f}|^2 \le \varepsilon^2 \quad i = 1, 2, ..., N_d, \tag{5.1b}$$

$$||\mathbf{f}||^2 \le R^2, \tag{5.1c}$$

where

$$J_n(\mathbf{f}) = \mathbf{f}^H\hat{\mathbf{R}}_{rr}\mathbf{f} - 2\mathbf{f}^H\hat{\mathbf{p}}_{dr} + \sigma_d^2$$

is the Mean Squared Error (MSE) between the input and output in the communication model described in the Chapter 3, and

$$\hat{\mathbf{p}}_{dr} = \frac{1}{n-\delta+1}\sum_{i=\delta}^{n} \mathbf{r}_{2i} d^*(i-\delta),$$

$$\hat{\mathbf{R}}_{rr} = \frac{1}{n-\delta+1}\sum_{i=\delta}^{n} \mathbf{r}_{2i}\mathbf{r}_{2i}^H,$$

are the length M sample cross-correlation and $M \times M$ sample auto-correlation matrix, respectively. The M tap equalizer is denoted by $\mathbf{f}$, and

$$\mathbf{r}_{2n} = \Big(r(2n), r(2n-1), r(2n-2), ..., r(2n-M+1)\Big)^T$$

with $r(n)$ being the input to the equalizer $\mathbf{f}$. The parameter δ is the system delay, and $(*)$ denotes complex conjugation. The approach of IPM1 is to view this problem as a convex feasibility problem. To do so, we define a convex feasibility region,

$$\Omega_n = \{\mathbf{f} \in \mathcal{C}^M | J_n(\mathbf{f}) \leq \tau_n, |\mathbf{s}_i^H \mathbf{f}|^2 \leq \varepsilon^2, ||\mathbf{f}||^2 \leq R^2\}, \tag{5.2}$$

where $\tau_n \geq 0$ is a scalar and $R > 0$ is a constant. The threshold τ_n is determined adaptively and is, in fact, the driving force behind the IPLS algorithm. The quadratically constrained adaptive equalization problem can be modeled as the problem of finding a feasible point which satisfys all the convex quadratic inequalities in (5.2). For convenience, let us define

$$s_n(\mathbf{f}) = \tau_n - J_n(\mathbf{f})$$

$$t_n(\mathbf{f}) = R^2 - ||\mathbf{f}||^2$$

and

$$\gamma_i(\mathbf{f}) = \varepsilon^2 - |\mathbf{s}_i^H \mathbf{f}|^2.$$

Now we define the following *potential function* on Ω_n,

$$\psi_n(\mathbf{f}) = -\log(s_n(\mathbf{f})) - \log(t_n(\mathbf{f})) - \frac{1}{N_d}\sum_{i=1}^{N_d} \log(\gamma_i(\mathbf{f})). \tag{5.3}$$

The sidelobe constraint term is divided by N_d in order to take into account the average effect of the sidelobe constraints. The function $\psi_n(\mathbf{f})$ is a *logarithmic barrier function.* It is strictly convex and has a unique minimizer over Ω_n. A point in the deep interior of Ω_n will have a low potential, whereas towards the boundary the potential tends towards infinity. The unique minimizer of this potential function is the analytic center of Ω_n.

If $\mathbf{f}_n^a$ is the unique minimizer of $\psi_n(\mathbf{f})$ for a given τ_n and R, then setting $\nabla\psi_n(\mathbf{f}_n^a) = 0$, we obtain the expression

$$\frac{\nabla J_n(\mathbf{f}_n^a)}{s_n(\mathbf{f}_n^a)} + \frac{1}{N_d}\sum_{i=1}^{N_d}\frac{2\mathbf{s}_i\mathbf{s}_i^H\mathbf{f}_n^a}{\gamma_i(\mathbf{f}_n^a)} + \frac{2\mathbf{f}_n^a}{t_n(\mathbf{f}_n^a)} = 0. \tag{5.4}$$

Now using

$$\nabla J_n(\mathbf{f}_n^a) = -2\mathbf{p}_{dr}(n) + 2\mathbf{R}_{rr}(n)\mathbf{f}_n^a$$

in (5.4) we have that

$$\mathbf{f}_n^a = \left(\alpha_n\mathbf{I} + s_n\frac{1}{N_d}\sum_{i=1}^{N_d}\frac{\mathbf{s}_i\mathbf{s}_i^H}{\gamma_i} + \mathbf{R}_{rr}(n)\right)^{-1}\mathbf{p}_{dr}(n), \tag{5.5}$$

where $\alpha_n = \frac{s_n}{t_n}$. The term $\alpha_n\mathbf{I}$ can be viewed as a regularization term for the matrix $\mathbf{R}_{rr}(n)$ (which maybe poorly conditioned). Recall that $s_n(\mathbf{f}) = \tau_n - J_n(\mathbf{f})$. Thus, this analysis confirms that τ_n is the parameter that drives algorithm towards the optimal solution. In order to reach a good minimizer we would therefore expect $s_n \to 0$ close to the optimal solution. For an efficient algorithm, we need to decide how to update τ_n effectively. Depending on how the parameter τ_n is varied, several versions of IPM1 are possible.

One candidate is the IPLS algorithm, which is also an approximation of an analytic center based estimator. It would be interesting to find out if this algorithm, originally designed for unconstrained problems, would perform in inequality constrained scenarios. In their unconstrained IPLS algorithm, Afkhamie *et. al.* [10] suggest that τ_n be updated as

$$\tau_n = J_n(\mathbf{f}_{n-1}^a) + K\|\nabla J_n(\mathbf{f}_{n-1}^a)\|_2, \tag{5.6}$$

where K is a constant. At first glance this looks like a good idea. Indeed, this formulation works well for the unconstrained least squares problem where $\nabla J_n(\mathbf{f}) \rightarrow 0$ towards the optimal solution. While this might be a candidate for the unconstrained case, in the presence of multiple quadratic sidelobe constraints $\nabla J_n(\mathbf{f})$ does not become vanishingly small at the optimal solution, due to the sidelobe constraints. Hence, (5.6) is not a very good assignment for τ_n in the constrained case. If this method is used, then the algorithm would be limited by unacceptable residual sub-optimality in the steady state. Since we do not expect the IPM1 algorithm with the adaptation of τ_n in (5.6) to deliver a good sub-optimal solution, we will not investigate the IPM1 algorithm any further in this chapter. The interested reader can find a description of the IPLS algorithm in Appendix-A. We now move on to an extension of the IPM2 algorithm, which is based on the MAC2 approach. IPM2 does provide us with a good sub-optimal solution to the quadratic inequality constrained least squares problem in the steady state.

5.2.2 Interior Point Method-2 (IPM2)

This is the second approach for solving the quadratic inequality constrained least squares problem using the Method of Analytic Centers. In this method we aim at solving a different sequence of convex feasibility problems to arrive at the optimal solution of problem in (5.1). The difference lies in the definition of the potential function $\psi_n(\mathbf{f})$ for the convex feasibility region Ω_n. We recall the convex feasibility region

$$\Omega_n = \{\mathbf{f} \in \mathcal{C}^M | J_n(\mathbf{f}) \leq \tau_n, \| \mathbf{s}_i^H \mathbf{f} \|^2 \leq \varepsilon^2, \| \mathbf{f} \|^2 \leq R^2\}. \tag{5.7}$$

The analytic center depends on the algebraic representation of the potential function $\psi_n(\mathbf{f})$ of Ω_n. As in MAC2, we write the potential function as

$$\psi_n(\mathbf{f}) = -\log(s_n(\mathbf{f})) - \rho_n \left(\log(t_n(\mathbf{f})) + \frac{1}{N_d} \sum_{i=1}^{N_d} \log(\gamma_i(\mathbf{f})) \right), \tag{5.8}$$

where

$$s_n(\mathbf{f}) = \tau - J_n(\mathbf{f}),$$

$$t_n(\mathbf{f}) = R^2 - \|\mathbf{f}\|^2,$$

and

$$\gamma_i(\mathbf{f}) = \varepsilon^2 - |\mathbf{s}_i^H \mathbf{f}|^2$$

where τ is a constant. The parameter ρ drives the algorithm towards the optimal solution, and is varied as $\rho_n = \beta\rho_{n-1}$, where $0 < \beta < 1$. Therefore, $\psi_n(\mathbf{f})$ is a weighted logarithmic barrier function of Ω_n. The function ψ_n is convex over Ω_n and has a unique minimum which we call the ρ-analytic center of Ω_n. Again, as emphasized earlier, the main idea is to stay close to the central path and make sufficient progress with each iteration towards the desired optimal solution. The function $\psi_n(\mathbf{f})$ is governed by the relative importance of the objective function term and the quadratic inequality constraint terms, which in turn, is governed by the parameter ρ. Hence, β can thought of as a measure of the "aggressiveness" of the algorithm. As $\rho \rightarrow 0$, the algorithm approaches the optimal solution. This is verified by the following mathematical analysis: If $\mathbf{f}_n^a$ is the unique minimizer of $\psi_n(\mathbf{f})$ for a given β and R, then setting $\nabla\psi_n(\mathbf{f}_n^a) = 0$, we have that

$$\frac{\nabla J_n(\mathbf{f}_n^a)}{s_n(\mathbf{f}_n^a)} + \rho_n \left(\frac{1}{N_d} \sum_{i=1}^{N_d} \frac{2\mathbf{s}_i\mathbf{s}_i^H\mathbf{f}_n^a}{\gamma_i(\mathbf{f}_n^a)} + \frac{2\mathbf{f}_n^a}{t_n(\mathbf{f}_n^a)} \right) = 0. \tag{5.9}$$

Now using

$$\nabla J_n(\mathbf{f}_n^a) = -2\hat{\mathbf{p}}_{dr}(n) + 2\hat{\mathbf{R}}_{rr}(n)\mathbf{f}_n^a$$

in (5.9) we have that

$$\mathbf{f}_n^a = \left(\alpha_n \mathbf{I} + \rho_n s_n \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{\mathbf{s}_i\mathbf{s}_i^H}{\gamma_i(\mathbf{f}_n^a)} + \hat{\mathbf{R}}_{rr}(n) \right)^{-1} \hat{\mathbf{p}}_{dr}(n). \tag{5.10}$$

where $\alpha_n = \rho_n \frac{s_n}{t_n}$. Here, it is the $\alpha_n \mathbf{I}$ term that provides sufficient regularization for the matrix $\hat{\mathbf{R}}_{rr}(n)$ in the initial stage, when it is poorly conditioned. Since $\rho_n \to 0$, as n grows, the effect of regularization is gradually eliminated, as desired. Note, however, that as $\rho_n \to 0$, some of the sidelobe terms $\gamma_i(\mathbf{f}_n^a) \to 0$. That means that even though the coefficient of the first matrix under the inverse in (5.10) becomes vanishingly small at the optimal solution, the second matrix, which carries the effect of the sidelobe constraints, remains non-trivial. Therefore, $\nabla J_n \neq 0$ at the optimal solution for MMSE equalizer with quadratic inequality constraints. This is the effect of the sidelobe constraints on the equalizer.

As discussed earlier, in order to reduce the computational cost it is desirable to take only one Newton step per outer iteration. As long as we stay close to the central path by taking only one Newton step, we can still hope to arrive at a good sub-optimal solution. Hence an approximate analytic center of Ω_n obtained using only one inner iteration starting from an approximation to $\mathbf{f}_{n-1}^a$ (previous iterate) might be sufficient, as long as β is not too aggressive (i.e., β is not too small).

The Newton iteration is given as

$$\mathbf{f}_n = \mathbf{f}_{n-1} - \left(\nabla^2 \psi_n(\mathbf{f}_{n-1})\right)^{-1} \nabla \psi_n(\mathbf{f}_{n-1}), \tag{5.11}$$

where $\mathbf{f}_n$ represents the approximate analytic center of Ω_n. The gradient and Hessian of $\psi_n(\mathbf{f})$ are given by,

$$\nabla \psi_n(\mathbf{f}) = \frac{\nabla J_n(\mathbf{f})}{s_n(\mathbf{f})} + \rho_n \left(\frac{1}{N_d} \sum_{i=1}^{N_d} \frac{2\mathbf{s}_i \mathbf{s}_i^H \mathbf{f}}{\gamma_i(\mathbf{f})} + \frac{2\mathbf{f}}{t_n(\mathbf{f})} \right) \tag{5.12}$$

and

$$\nabla^2 \psi_n(\mathbf{f}) = \frac{\nabla^2 J_n(\mathbf{f})}{s_n(\mathbf{f})} + \frac{\nabla J_n(\mathbf{f}) \nabla J_n(\mathbf{f})^H}{s_n^2(\mathbf{f})} + \rho_n \left(\frac{1}{N_d} \sum_{i=1}^{N_d} \frac{2\mathbf{s}_i \mathbf{s}_i^H}{\gamma_i(\mathbf{f})} + \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{4\mathbf{s}_i \mathbf{s}_i^H \mathbf{f}\mathbf{f}^H \mathbf{s}_i \mathbf{s}_i^H}{\gamma_i^2(\mathbf{f})} + \frac{4\mathbf{f}\mathbf{f}^H}{t_n^2(\mathbf{f})} + \frac{2\mathbf{I}}{t_n(\mathbf{f})} \right), \tag{5.13}$$

respectively. The gradient and Hessian of $J_n(\mathbf{f})$ are given by

$$\nabla J_n(\mathbf{f}) = -2\hat{\mathbf{p}}_{dr}(n) + 2\hat{\mathbf{R}}_{rr}(n)\mathbf{f} \tag{5.14a}$$

$$\text{and} \quad \nabla^2 J_n(\mathbf{f}) = 2\hat{\mathbf{R}}_{rr}(n), \tag{5.14b}$$

respectively. Thus by using an approximate analytic center instead of an exact minimizer we obtain an implementable analytic center based estimator for a quadratic inequality constrained convex optimization problem, where the approximate analytic centers are updated by a single Newton iteration. The details of the algorithm are summarized in Table 5.1. For the first Newton step, we use $\mathbf{f} = \mathbf{0}$, as the starting point, which is the analytic center of the constraint set. As discussed in this section, we have a practically implementable interior point method based algorithm that is useful for efficiently solving quadratically constrained adaptive filtering problems. It is based on the Method of Analytic Centers (MAC2), and works well if appropriate values of the parameters are chosen. In the next section we analyse the problem of performing adaptive equalization with strict sidelobe control using the standard barrier method.

5.3 Standard Barrier Method based estimator

The standard barrier method can also be employed to efficiently perform quadratically constrained least squares minimization. A general idea of the standard barrier method was given in the previous chapter. Now we intend to use it to perform symbol-by-symbol adaptive equalization with strict sidelobe control. A sequence of unconstrained minimization problems are solved to arrive at the optimal solution of the original problem defined in (5.2).

The original optimization problem in (5.1) can be approximated as

$$\min_{\mathbf{f}} \psi(\mathbf{f}) = J_n(\mathbf{f}) + \frac{1}{t_n}\phi(\mathbf{f}) \tag{5.17}$$

Table 5.1: IPM2

- **Step 1: Initialization.** Let $\varepsilon, N_d, R, \beta$ be given. Set

$$\mathbf{f}_o = \mathbf{0}, \quad \mathbf{p}_{dr}(0) = \mathbf{0}, \quad \mathbf{R}_{rr}(0) = \mathbf{0}, \quad \nabla J_0(\mathbf{0}) = \mathbf{0}.$$

- **Step 2: Updating.** For $n \geq 1$, update $\mathbf{r}_{2n}$ and $d(n-\delta)$, and then update

$$\hat{\mathbf{p}}_{dr}(n) = \frac{n-1}{n}\hat{\mathbf{p}}_{dr}(n-1) + \frac{1}{n}\mathbf{r}_{2n}d^*(n-\delta) \tag{5.15a}$$

$$\hat{\mathbf{R}}_{rr}(n) = \frac{n-1}{n}\hat{\mathbf{R}}_{rr}(n-1) + \frac{1}{n}\mathbf{r}_{2n}\mathbf{r}_{2n}^H. \tag{5.15b}$$

Update

$$\nabla J_n(\mathbf{f}_{n-1}) = -2\hat{\mathbf{p}}_{dr}(n) + 2\hat{\mathbf{R}}_{rr}(n)\mathbf{f}_{n-1}$$

$$t_n(\mathbf{f}_{n-1}) = R^2 - ||\mathbf{f}_{n-1}||^2$$

$$\gamma_i(\mathbf{f}) = \varepsilon^2 - |\mathbf{s}_i^H\mathbf{f}_{n-1}|^2 \quad i = 1, 2, ..., N_d$$

$$\nabla\psi_n(\mathbf{f}_{n-1}) \quad \nabla^2\psi_n(\mathbf{f}_{n-1}).$$

using (5.12) and (5.13), and

$$\rho_n = \beta\rho_{n-1} \tag{5.16}$$

- **Step 3: Recentering.** The new approximate center of Ω_n is obtained by taking one Newton step with $\mathbf{f}_{n-1}$ as the starting point,

$$\mathbf{f}_n = \mathbf{f}_{n-1} - \left(\nabla^2\psi_n(\mathbf{f}_{n-1})\right)^{-1}\nabla\psi_n(\mathbf{f}_{n-1}).$$

Set $n := n+1$, and return to **Step 2**.

where

$$\phi(\mathbf{f}) = \log(u(\mathbf{f})) + \frac{1}{N_d}\sum_{i=1}^{N_d}\log(\gamma_i(\mathbf{f})) \tag{5.18}$$

where ϕ is the logarithmic barrier for the original problem and

$$u(\mathbf{f}) = R^2 - ||\mathbf{f}||^2$$

and

$$\gamma_i(\mathbf{f}) = \varepsilon^2 - |\mathbf{s}_i^H\mathbf{f}|^2$$

are defined for convenience. The log barrier function $\phi(\mathbf{f})$ is convex and differentiable over the domain of points that satisfy the inequality constraints in (5.1). The parameter $t_n > 0$ gives a measure of the accuracy of the approximation of the original optimization problem to an unconstrained modified objective function with a logarithmic barrier, the approximation becomes more accurate as t_n increases. The function $\psi_n(\mathbf{f})$ is governed by the relative importance of the objective function and the log barrier, which is dependent on the parameter t_n. The changes in the parameter t_n define the aggressiveness of the algorithm, and drive the algorithm to an optimal solution as $t_n \longrightarrow \infty$. We verify this by a similar mathematical analysis to that provided in the previous section. Let $\mathbf{f}_n^a$ denote the minimizer of the problem in (5.17) at the n^{th} iteration. Therefore, we would expect $\nabla\psi_{n,t}(\mathbf{f}_n^a) = 0$, i.e.,

$$\nabla J_n(\mathbf{f}_n^a) + \frac{1}{t_n}\left(\frac{1}{N_d}\sum_{i=1}^{N_d}\frac{2\mathbf{s}_i\mathbf{s}_i^H\mathbf{f}_n^a}{\gamma_i(\mathbf{f}_n^a)} + \frac{2\mathbf{f}_n^a}{u_n(\mathbf{f}_n^a)}\right) = 0. \tag{5.19}$$

Now using (5.14b) in the above equation, we have that

$$\mathbf{f}_n^a = \left(\alpha_n\mathbf{I} + \frac{1}{t_n}\frac{1}{N_d}\sum_{i=1}^{N_d}\frac{\mathbf{s}_i\mathbf{s}_i^H}{\gamma_i(\mathbf{f}_n^a)} + \hat{\mathbf{R}}_{rr}(n)\right)^{-1}\hat{\mathbf{p}}_{dr}(n). \tag{5.20}$$

where $\alpha_n = \frac{1}{t_n u_n}$. Here, it is the $\alpha_n\mathbf{I}$ term that again provides the regularization required by the auto-correlation matrix $\mathbf{R}_{rr}$ in the initial phase. As $t \longrightarrow \infty$ close to the optimal solution, the effect of the regularization term is gradually eliminated. Note that some of the sidelobe terms $\gamma_i(\mathbf{f}_n^a) \rightarrow 0$ as $t_n \rightarrow \infty$. Therefore, though the coefficient of the first matrix under the inverse in (5.20) becomes vanishingly small at the optimal solution, the second matrix which carries the effect of the sidelobe constraints remains non-trivial as $t_n \rightarrow \infty$. Therefore, the optimal solution is not equal to the Wiener solution, and in particular $\nabla J_n \neq 0$ at the optimal solution.

In the algorithm, we choose an initial starting point that is strictly feasible, and proceed using the Newton method, trying to stay as close to the central path as possible. We solve a sequence of unconstrained minimization problems, using the most recent approximation of a central point as the starting point for the next unconstrained minimization problem.

Recall that the Newton iteration is given as,

$$\mathbf{f}_n = \mathbf{f}_{n-1} - \Big((\nabla^2 \psi_n(\mathbf{f}_{n-1}))\Big)^{-1} \nabla \psi_n(\mathbf{f}_{n-1}), \tag{5.21}$$

where $\mathbf{f}_n$ represents the approximate central point of the unconstrained problem at the n^{th} iteration. The gradient and Hessian of $\psi_n(\mathbf{f})$ are given as,

$$\nabla \psi_n(\mathbf{f}) = \nabla J_n(\mathbf{f}) + \frac{1}{t_n}\left(\frac{1}{N_d}\sum_{i=1}^{N_d} \frac{2\mathbf{s}_i\mathbf{s}_i^H\mathbf{f}}{\gamma_i(\mathbf{f})} + \frac{2\mathbf{f}}{u_n(\mathbf{f})}\right) \tag{5.22}$$

and

$$\nabla^2 \psi_n(\mathbf{f}) = \nabla^2 J_n(\mathbf{f}) + \frac{1}{t_n}\left(\frac{1}{N_d}\sum_{i=1}^{N_d} \frac{2\mathbf{s}_i\mathbf{s}_i^H}{\gamma_i(\mathbf{f})} + \frac{1}{N_d}\sum_{i=1}^{N_d} \frac{4\mathbf{s}_i\mathbf{s}_i^H\mathbf{f}\mathbf{f}^H\mathbf{s}_i\mathbf{s}_i^H}{\gamma_i^2(\mathbf{f})} + \frac{4\mathbf{f}\mathbf{f}^H}{u_n^2(\mathbf{f})} + \frac{2\mathbf{I}}{u_n(\mathbf{f})}\right). \tag{5.23}$$

Therefore, we have an implementable interior point based estimater obtained by approximating the standard barrier method. This algorithm, in which a single Newton step is used to arrive at the approximate central point of the next unconstrained problem, is similar to IPM2 algorithm in implementation. It is given in Table 5.2.

The update of t_n depends on the parameter $\mu \geq 1$, and it is used to control the aggres-

Table 5.2: SBM

•**Step 1: Initialization.** Let $\varepsilon, N_d, R, \mu > 1$ be given. Set

$$\mathbf{f}_o = \mathbf{0}, \quad \hat{\mathbf{p}}_{dr}(0) = \mathbf{0}, \quad \hat{\mathbf{R}}_{rr}(0) = \mathbf{0}, \quad \nabla J_0(\mathbf{0}) = \mathbf{0}.$$

•**Step 2: Updating.** For $n \geq 1$, update $\mathbf{r}_{2n}$ and $d(n-\delta)$, and then update

$$\hat{\mathbf{p}}_{dr}(n) = \frac{n-1}{n}\hat{\mathbf{p}}_{dr}(n-1) + \frac{1}{n}\mathbf{r}_{2n}d^*(n-\delta) \tag{5.24a}$$

$$\hat{\mathbf{R}}_{rr}(n) = \frac{n-1}{n}\hat{\mathbf{R}}_{rr}(n-1) + \frac{1}{n}\mathbf{r}_{2n}\mathbf{r}_{2n}^H. \tag{5.24b}$$

Update

$$\nabla J_n(\mathbf{f}_{n-1}) = -2\hat{\mathbf{p}}_{dr}(n) + 2\hat{\mathbf{R}}_{rr}(n)\mathbf{f}_{n-1}$$

$$u_n(\mathbf{f}_{n-1}) = R^2 - \|\mathbf{f}_{n-1}\|^2$$

$$\gamma_i(\mathbf{f}) = \varepsilon^2 - |\mathbf{s}_i^H\mathbf{f}_{n-1}|^2 \quad i = 1, 2, ..., N_d$$

$$\nabla\psi_{n,t}(\mathbf{f}_{n-1}), \quad \nabla^2\psi_{n,t}(\mathbf{f}_{n-1})$$

using (5.22) and (5.23) and

$$t_n = \mu t_{n-1} \tag{5.25}$$

•**Step 3: Recentering.** The new approximate central point is obtained by taking one exact Newton step with $\mathbf{f}_{n-1}$ as the starting point,

$$\mathbf{f}_n = \mathbf{f}_{n-1} - \left(\nabla^2\psi_n(\mathbf{f}_{n-1})\right)^{-1}\nabla\psi_n(\mathbf{f}_{n-1}).$$

Set $n := n + 1$, and return to **Step 2**.

siveness of the algorithm. A small value of μ would mean that the algorithm progresses conservatively towards the optimal solution, whereas a large value would make it more aggressive.

Thus we have two implementable interior point based algorithms that efficiently solve the symbol-by-symbol adaptive equalization with strict sidelobe control problem to a good suboptimal solution. Both the algorithms use a Newton step in the update of the weight vector, which involves the calculation of the inverse of the Hessian $\nabla^2\psi_n(\mathbf{f}_{n-1})$. This requires $O(M^3)$ arithmetic operations per iteration. Although this is higher then the per iteration complexity of $O(M^2)$ offered by the RLS algorithm, we are able to efficiently impose quadratic inequality constraints into adaptive filtering problems.

Chapter 6

MMSE equalization with explicit sidelobe control: Simulation Results

6.1 Introduction

In this chapter, we implement the Minimum Mean Squared Error (MMSE) linear equalizers with explicit sidelobe control presented in the preceding chapters. We make some standard approximations and assumptions in our implementation, and provide simulation results for a host of different scenarios. We perform both block adaptive, as well as symbol-by-symbol adaptive equalization. In both cases we employ discretized frequency constraints, as proposed in the previous chapters. We discuss the merits and limitations of each approach, and provide quantitative comparisons. Overall we demonstrate the utility of interior point based algorithms in solving the problem of adaptive equalization with multiple quadratic inequality constraints. Finally, we compare our results with the standard RLS algorithm with explicit diagonal loading. This is a good choice for comparison because of the implicit sidelobe suppression provided by the diagonal loading and because that algorithm has lower computational cost. Sidelobe suppression is required to suppress unexpected interferers in the sidelobe region.

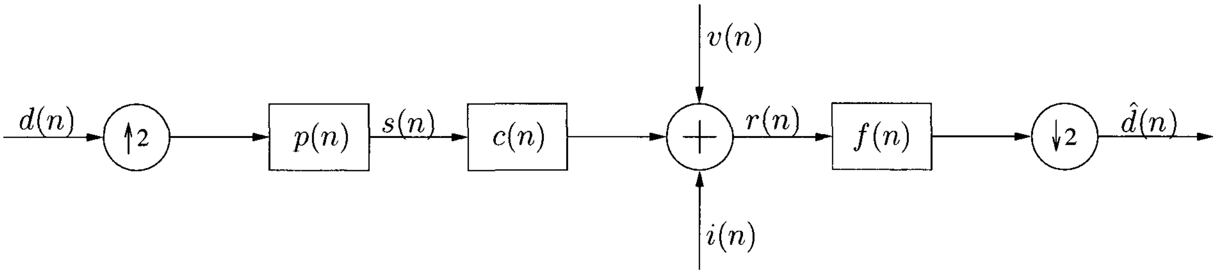

Figure 6.1: The Communication Model

In our implementations we send training symbols preceding the data symbols to estimate the statistics of the channel output. An interferer in the sidelobe region may or may not be present during the training phase. In this chapter, we analyse both scenarios, and also compare the simulation results for diagonally loaded RLS, as well as the interior point based algorithms.

6.2 Block Equalization

When communicating over a static channel, the block adaptive approach to the design of a MMSE equalizer with explicit sidelobe constraints can be adopted. A static channel is one whose coefficients do not change during the duration of message transmission. As discussed in Chapter 3, we formulate the problem as a Second Order Cone Programming (SOCP) problem, and then use the SeDuMi Toolbox [4] to obtain the FIR filter. If the true statistics of the channel output are available, then a good SOCP solver [4] will provide us with an optimal solution to the problem. However, in practice we use estimates of the channel auto-correlation matrix, and the cross-correlation vector between the receiver input and transmitted data. Therefore, SeDuMi provides us with an estimate of the optimal FIR filter. The sub-optimality of this filter depends on the accuracy of the sample auto-correlation matrix of the receiver input and the sample cross-correlation vector between the receiver input and transmitted data symbols.

For convenience, we recall the communication model (Fig. 6.1) presented in Chapter 3. In our experiments, the FIR equalizer has $M = 32$ taps and we provide $N = 64$ training symbols to estimate true the auto-correlation matrix of the receiver input and cross-correlation vector of the receiver input and transmitted data symbols. The input data stream is antipodal with each point in the alphabet $[+1, -1]$ having equal probability. The transmit filter is a 13 tap truncated root raised cosine filter with excess bandwidth 22%. The tap weights are

$$\mathbf{p} = \begin{bmatrix} -0.0270 \\ 0.0546 \\ 0.0350 \\ -0.1269 \\ -0.0406 \\ 0.4425 \\ 0.7504 \\ 0.4425 \\ -0.0406 \\ -0.1269 \\ 0.0350 \\ 0.0546 \\ -0.0270 \end{bmatrix}.$$

We assume that the 4 tap channel is unknown and its coefficients remain constant for the duration of the data transmission. In our simulations we have used the channel

$$\begin{pmatrix} 0.7985 & -0.4819 & 0.3561 & 0.0590 \end{pmatrix}^T.$$

Two of the three zeros of this channel makes angles of ±60° with the positive x-axis and they form a complex conjugate pair with a magnitude of 0.74. The third zero is at an angle of 180° and has a magnitude of 0.135. Thus, this is a minimum-phase channel.

We need to define a sidelobe region in the frequency response of the equalizer, and then choose discrete frequency points at which the quadratic inequality constraints will be applied. In the frequency spectrum of the equalizer we define the sidelobe region $\Psi_{sl} := [-\pi, -\text{edge}] \cup [\text{edge}, \pi]$ and use edge $= 2$ in our experiments. We make $N_d = 400$ uniformly separated discretizations on Ψ_{sl} such that there are 200 discretizations on each sidelobe. The desired sidelobe suppression is 30 dB. Zero mean white Gaussian noise is added to the receiver input and in our experiments we operate at an SNR of 40 dB, where

$$\text{SNR} = \frac{E\{s(n)^2\}}{\sigma_v^2},$$

where σ_v^2 is the variance of the AWGN term, $v(n)$. Given our definitions of $\mathbf{p}$ and $d(n)$, SNR$= 1/2\sigma_v^2$. This is a good choice because here we are trying to show some of the fundamental issues of the problem. Therefore, testing the system performance at a high SNR is not unreasonable. The system delay is $\delta = 8$ for all experiments. For simplicity we have used real valued channel, transmit filter and data, but the same algorithm can be used for complex valued data and filter taps as well.

Experiment 1

In the first experiment, we assume that there is no interferer present in the training phase. The aim of this experiment is to implement the sidelobe constraints in the equalizer, and demonstrate the applicability of the SOCP to solve the quadratically constrained optimization problem.

Fig. 6.2 shows the frequency response of the optimal equalizer obtained using the true auto-correlation and cross-correlation matrices (dashed curve). It is clearly shown that the sidelobe region is successfully contained below the desired sidelobe level. Next we estimate this true filter using 64 training symbols. This kind of scenario is more often witnessed in a practical communication system, as the channel characteristics are unlikely to be known at

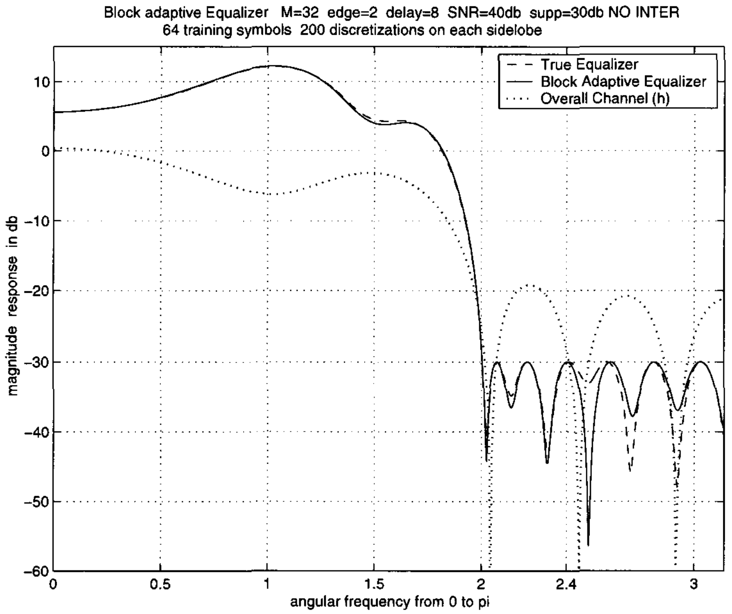

Figure 6.2: Frequency Response for SOCP based Sidelobe Constrained Equalizer

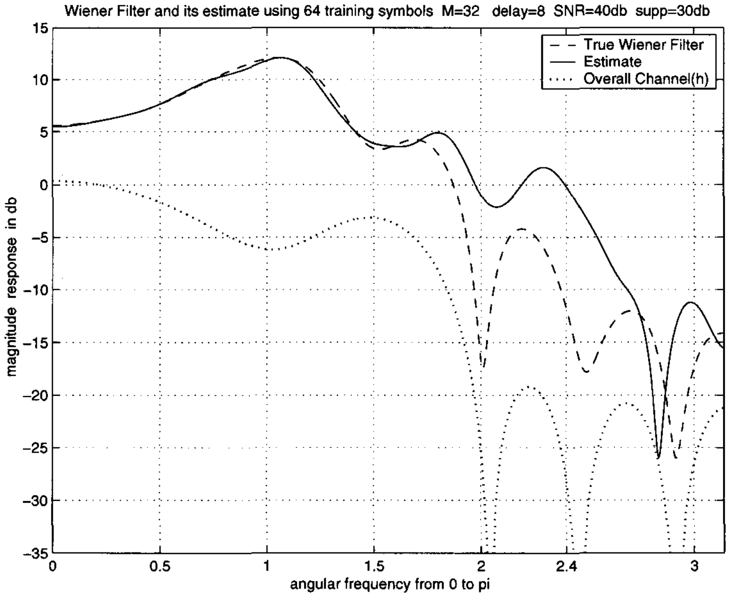

Figure 6.3: Frequency Response for the Wiener Filter and its estimate

the receiver. The frequency response of this estimate of the optimal equalizer is also shown in Fig. 6.2. We see that the sidelobes of both the filters are constrained at -30 dB as desired, thereby ensuring robustness to external interferers in the sidelobe region.

In Fig. 6.3 we see the performance of the Wiener filter and its estimate using 64 training symbols on the same static channel. We observe that the sidelobe region for the estimate of the Wiener filter is high, and hence it is quite susceptible to even moderately powerful interferers due to Adjacent Channel Interference (ACI); especially in the frequency band $2 - 2.4$ rad/sample. Our block adaptive sidelobe constrained equalizer is however robust to such interferers.

In Fig. 6.1, for convenience we define the overall channel $\mathbf{h} = \mathbf{p} \otimes \mathbf{c}$, where $\otimes$ denotes convolution and $\mathbf{g} = \mathbf{h} \otimes \mathbf{f}$. For Fig. 6.1 the receiver output $\hat{d}(n)$ can be written as,

$$\hat{d}(n) = \sum_{i} g(2i)d(n-i) + \sum_{j} f(j)v(2n-j) \tag{6.1a}$$

$$= g(2\delta)d(n-\delta) + \sum_{i \neq \delta} g(2i)d(n-i) + \sum_{j} f(j)v(2n-j), \tag{6.1b}$$

where δ is the system delay. In (6.1b), the second term is the ISI term and the third term is due to noise. We can use $g(n)$ to asses the measure of ISI performance. A good equalizer will naturally tend to suppress the ISI, and in turn the sum of even indexed terms of $g(n)$

$$\sum_{i \neq \delta} |g(2i)|.$$

For the optimal equalizer shown in Fig. 6.2, we plot the $g(n)$ in Fig. 6.4 for a system delay of $\delta = 8$. Note that the even indexed terms of $g(n)$ are small, except at $n = 2\delta$, which is close to 1.

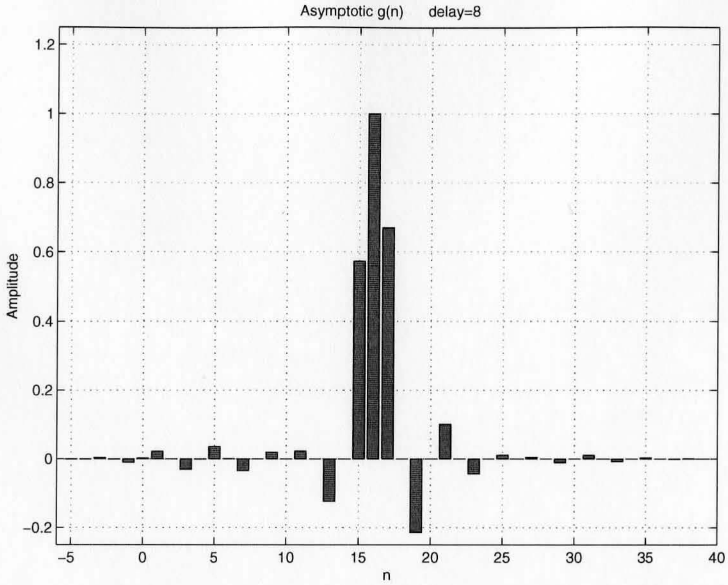

Figure 6.4: Overall impulse response $g(n)$

Experiment 2

In this experiment, we provide the Asymptotic Mean Squared Error (MSE) and Asymptotic Inter Symbol Interference (ISI) performance using the same static channel for different sidelobe regions and sidelobe levels. This provides an indication of the trade-offs incurred when imposing tight constraints. This is important as it can be used as a limiting performance index while implementing algorithms which estimate the optimal equalizer using training. As discussed in the previous section we can use the peak distortion as measure for ISI. This can be used as a measure of the quality of performance of the equalizer.

As we see in Figs. 6.5 and 6.6, when the constraints are relaxed to 0 dB, both the MSE and ISI performance are same as the Wiener performance for all sidelobe regions. However, when tighter sidelobe constraints are applied for a given sidelobe region, the feasible set for the proposed equalizer becomes smaller. The rise in MSE and ISI is only due to the tightening of the sidelobe constraints for a given sidelobe region. Also, we see that for a given sidelobe level, both the MSE and ISI are higher for a larger sidelobe region, which is expected as now we constrain a larger sidelobe region.

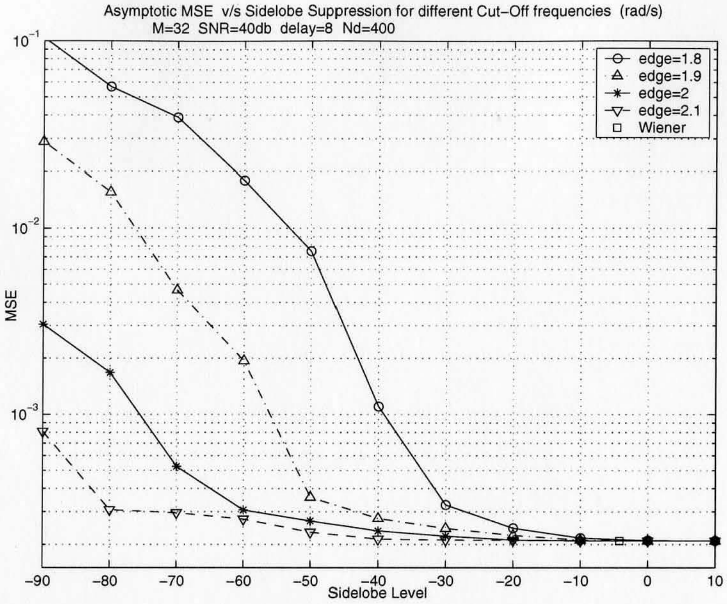

Figure 6.5: Asymptotic MSE performance index

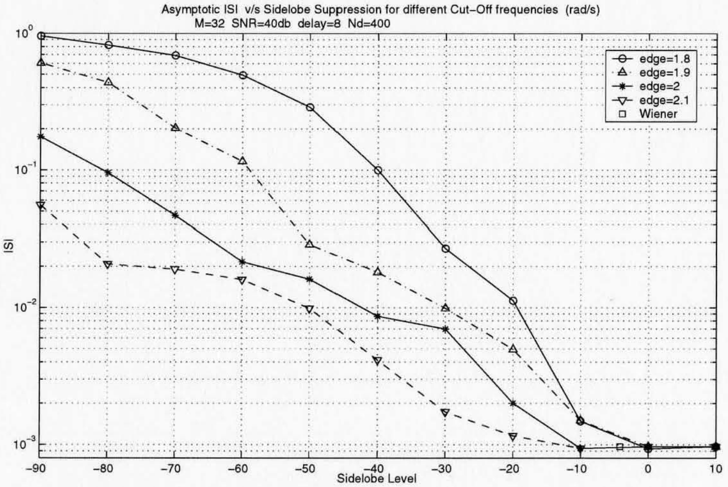

Figure 6.6: Asymptotic ISI performance index

6.3 MMSE symbol-by-symbol adaptive equalizer with explicit sidelobe control

In this section we implement a constrained symbol-by-symbol adaptive equalizer using the theory and analysis provided in Chapter 5. We discussed two approaches to solve the problem, namely Interior Point Method-2 (IPM2) and the Standard Barrier Method (SBM).

Unlike the block adaptive approach, where a block of training symbols needs to be transmitted before the equalizer can be implemented, now we implement the equalizer from the first training symbol itself. Furthermore, we are not dependent on the availability of the SeDuMi tool [4], but instead we use standard methods like the Newton method. First, we do some experiments on a static channel, and then we use a block fading channel.

6.3.1 Static Channel

The aim here is to analyse the convergence properties of the equalizer when sufficient training is provided. We use the static channel that was used in the experiments in the previous section; i.e., $c(n) = (\ 0.7985 \quad -0.4819 \quad 0.3561 \quad 0.0590\ )^T$. We test the performance of the IPM2 algorithm for different values of β. Recall from Chapter 5 that β is the parameter that controls the aggressiveness of the algorithm. Since we are taking only one Newton step, a smaller value of β would ensure proximity to the central path with each iteration. If sufficient training is provided, this in turn would provide us with a good sub-optimal equalizer with explicit sidelobe constraints. On the other hand, using an aggressive value of β could result in the solution deviating from the central path, if an insufficient number of inner iterations are performed. In IPM2, we do only one inner iteration, hence in our implementations we intend to use a conservative β. We shall use the mean squared tap-error plot of the FIR filter obtained using IPM2 as a measure of performance. The Mean Squared tap error for IPM2 is defined as,

$$e_{\text{ipm}}(n) = E\{\|\mathbf{f}_n^{\text{ipm}} - \mathbf{f}_{\text{S}}\|_2^2\},$$

where $\mathbf{f}_n^{\mathrm{ipm}}$ is the IPM2 based estimate of the optimal (block adaptive) equalizer $\mathbf{f}_{\mathrm{S}}$ obtained using the true channel statistics. This filter was obtained using SeDuMi [4] in Section 6.2. Similarly, we plot the Mean Squared tap error performance of RLS as well, which is defined as

$$e_{\mathrm{rls}}(n) = E\{\|\mathbf{f}_n^{\mathrm{rls}} - \mathbf{f}_{\mathrm{W}}\|_2^2\},$$

where $\mathbf{f}_{\mathrm{W}}$ is the Wiener filter and $\mathbf{f}_n^{rls}$ is the RLS filter at the n^{th} instant.

In this experiment, the IPM2 algorithm is tuned so that it stays close to the central path while taking only one Newton step per iteration. Hence we provide sufficient training, and chose conservative values for β. We plot three Mean Squared tap error plots for training sequence of length $N = 1000$ and different values of β, namely, $\beta = 0.99$, $\beta = 0.975$ and $\beta = 0.95$. In all the 3 cases we use $\mathbf{f} = \mathbf{0}$ as a convenient feasible starting point and we choose $R = 10$.[1] For RLS we use $\xi = 10^{-4}$, where ξ is the amount of regularization added to the sample auto-correlation matrix $\hat{\mathbf{R}}_{rr}(n)$ to ensure that the sample auto-correlation matrix remains non-singular at all iterations. The regularization term decays with the iterations, as the RLS algorithm implicitly computes the solution $\mathbf{f}_n^{rls} = \left((\frac{\xi}{n}\mathbf{I} + \hat{\mathbf{R}}_{rr}\right)^{-1}\hat{\mathbf{p}}_{dr}$, where $\hat{\mathbf{p}}_{dr}$ is the sample cross-correlation vector between the receiver input and desired symbol, and $\mathbf{I}$ is an identity matrix of appropriate dimensions.

The mean squared tap error plots for $\beta = 0.99$, 0.975 and 0.95, along with that of RLS are given in Fig. 6.7. We first analyse the $\beta = 0.99$ case. No damping of the Newton method was observed in this case, indicating that for this conservative value of β one Newton step (inner iteration) keeps us close enough to the central path. Now we make the algorithm more aggressive by reducing β to 0.975. The Mean Squared tap error plot for this case is shown in Fig. 6.7 (solid line). We observe that the algorithm is more aggressive in this case as expected. After nearly 700 iterations damping was required. Therefore, for this value

[1] In our implementation of the Newton Method, we use damping to control the numerical properties close to the boundary. The details of this modification are provided in Appendix B. This numerical control is necessary when the combination of an aggressive value for β and the use of only one Newton step per outer iteration allows the sequences of estimates of the analytic center to drift away from the central path.

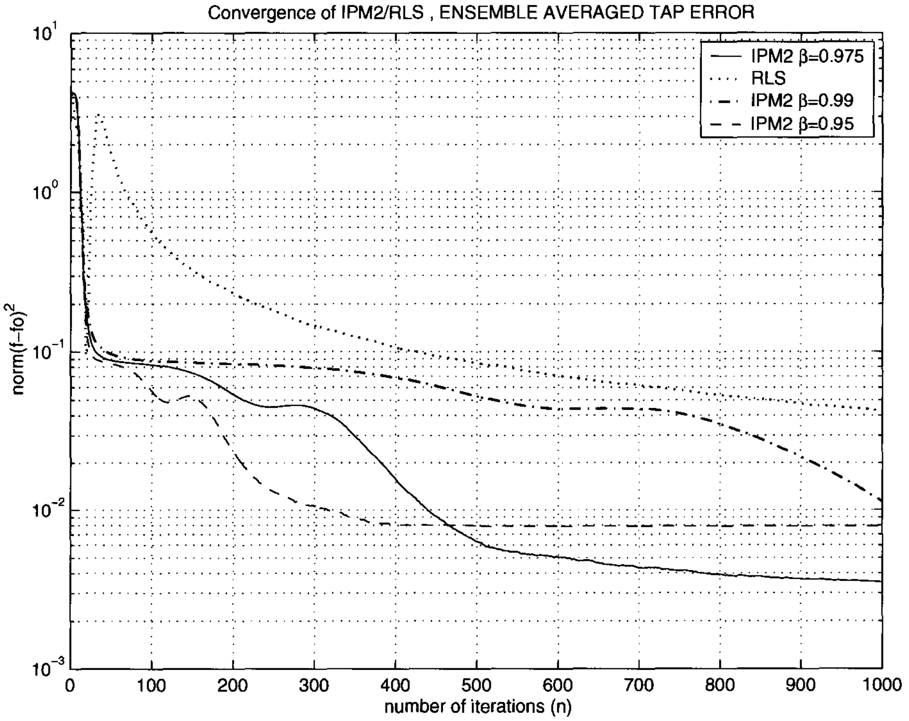

Figure 6.7: Convergence of RLS and IPM2

of β, one inner iteration is insufficient to stay close enough to the central path. However, compared to the performance for $\beta = 0.99$, in this case we arrive at a much better sub-optimal solution given the amount of training available. Therefore, there is a trade-off between the best sub-optimal solution attainable and the aggressiveness of the algorithm. We deviate from the central path to the optimal solution for an aggressive β and eventually the performance tends to be limited at the boundary. This is so, because the Hessian tends to vary rapidly near the boundary of the feasible set. Due to this inherent ill-conditioning of barrier methods at the boundary, for an aggressive β, the algorithm might not perform well. This is highlighted in the next case when the algorithm is deliberately made more aggressive to verify the trade off between aggressiveness and limiting performance. We choose a more aggressive value of $\beta = 0.95$. The damping starts after 300 iterations in this case. This is because with the more aggressive β, one inner iteration is unable to keep the sequence of equalizers close to the central path. In fact, the sequence deviates substantially from the central path, and it approaches the boundary quickly, but at some distance from an optimal solution. From there, the progress of the algorithm towards an optimal solution is limited by the ill-conditioning of the Hessian of the barrier function near the boundary. Note however that the transient convergence of IPM2 improves with an aggressive β.

In all the three cases, we see that the transient performance of IPM2 is better then that of RLS. This is attributed to the implicit adaptive diagonal loading present in the algorithm, as illustrated in Chapter 5. This gives a substantial advantage over RLS algorithm which relies on a fixed explicit diagonal loading and takes a long time to phase out the effects of initialization.

6.3.2 Block Fading Channel

In this section we test the performance of our algorithm over a *block fading channel.* Here we assume that the channel is stationary for a finite block of transmitted symbols. Hence all symbols of a block experience the same fade. This standard model represents slow fading channel scenarios. In our implementation we assume that the channel realization in each block is independent from those in other blocks. Each of these independent blocks is a 4 tap channel with uncorrelated Gaussian coefficients. We assume that these coefficients have exponentially decreasing variances, i.e., $\sigma_{c,n}^2 = \sigma_c^2 e^{\gamma n}$ for $n = 0, ..., L_c - 1$, where L_c is the length of the FIR channel. In our simulations we have used $\gamma = -2$. The distribution of the magnitude of zeros for 10000 channels is shown in Fig. 6.10.

This kind of scenario may arise in wireless transmission in urban areas [14]. Therefore, attenuation of the delayed signal is not unexpected. This kind of channel typically has its zeros located inside the unit circle. This is important because for deep nulls in the channel frequency spectrum due to zeros close to the unit circle, the performance of the linear equalizer may be severely affected due to noise enhancement. Therefore, by choosing a certain class of channels, the majority of which have zeros inside the unit circle, we have a low probability of encountering channels for which a linear equalizer would be inappropriate.

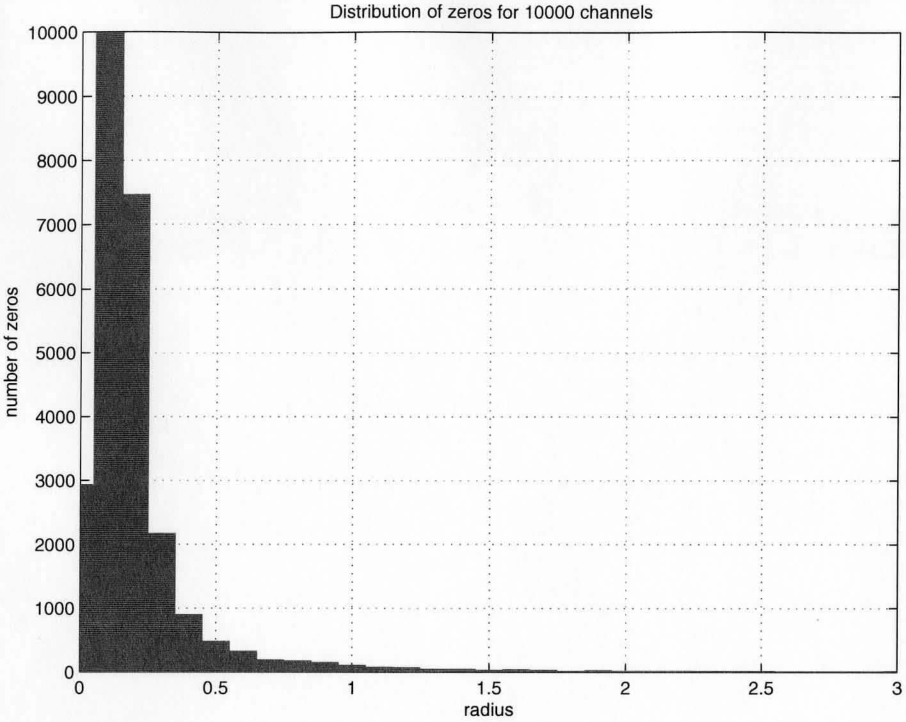

Figure 6.8: Distribution of zeros for 10000 channels for $\gamma = -2$

Notice from Fig. 6.8 that even after this favorable modeling of the channels, there are still some channels which have zeros clustered around the unit circle. These channels are inherently difficult to equalize with a linear equalizer. However, on the other hand, in a practical scenario one cannot imagine to have all channels to be favorable either. Hence, we proceed to use this class of channels in our simulations with a linear equalizer.

We use an $M = 32$ tap FIR filter as the equalizer, and have 100 discrete frequency constraints in the sidelobe region. The desired sidelobe suppression is 30 dB. In a practical scenario there might be a limitation on the amount of training available to the receiver. In our implementation, we send a message signal of length 256 symbols of which the first 64 symbols are used for training, and the remaining 192 are the data symbols. We want to test the performance of the receiver in the presence of an interferer in the sidelobe region due to ACI. For testing purposes we model the interferer as a sinusoid of frequency 2.2 rad./sample at an Interference to Signal ratio (ISR) of 0 dB. We first present simulation results for IPM2,

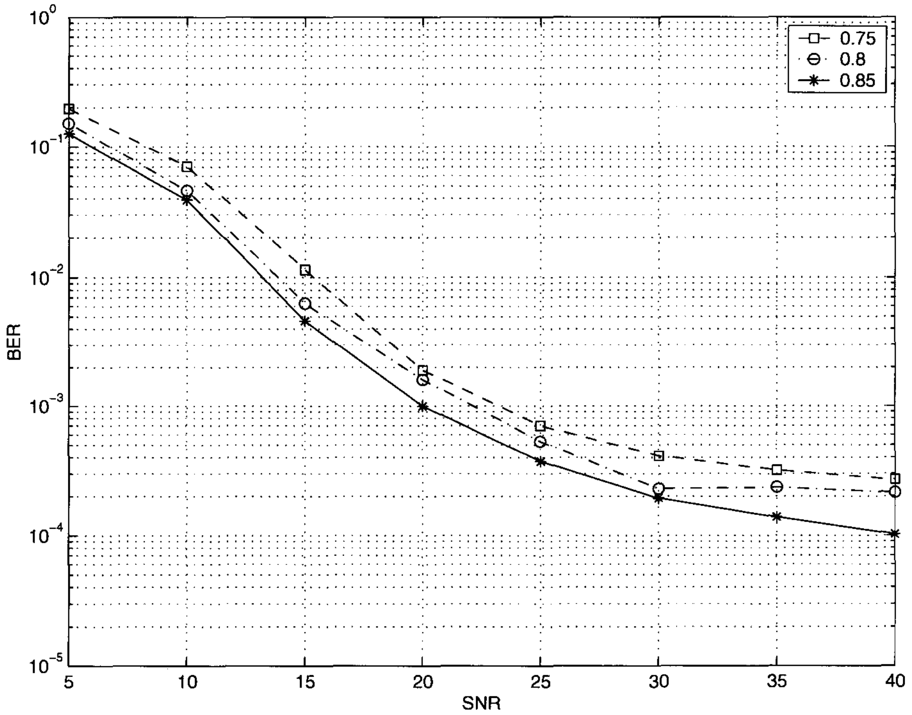

Figure 6.9: Equalization performance of IPM2 with sidelobe suppression of 30 dB for $\beta = 0.75, 0.8$ and 0.85.

and then compare the IPM2 method with the SBM approach.

Experiment 1

In the first experiment, we demonstrate the BER performance of the IPM2 algorithm for three different values of β. We assume the presence of ACI in the data block only, and that no interferer is present in the training. Due to limited training available, we choose aggressive values of β, namely $\beta = 0.75$, $\beta = 0.8$ and $\beta = 0.85$. The results are plotted in Fig. 6.9. We see that the performance improves for higher (less aggressive) values of β. One explanation for this behavior is that for a more aggressive value, the chances of the solution deviating from the central path are higher than that in the case of a less aggressive β. We observe that there is a limiting BER performance towards the higher SNR values in all the three cases. This is because of the error in the receiver due to residual ISI present because of multiple sidelobe constraints. This is the cost of ensuring robustness in the equalizer to

ACI.

Experiment 2

In order to fully appreciate the constrained equalization performance of the IPM2 based equalizer, we now compare its performance to that of a RLS algorithm with explicit diagonal loading. This is a good choice because additional diagonal loading leads to a moderate reduction in the sidelobe levels. This is so because the sidelobe levels depend on the eigenvalue spread of the sample auto-correlation matrix of the receiver inputs [12]. The eigenvalue spread is the ratio $\frac{\lambda_{max}}{\lambda_{min}}$, where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of the sample auto-correlation matrix of the receiver inputs, respectively. For fewer samples (limited training), the eigenvalue spread might be higher. By explicitly adding a diagonal matrix $\alpha\mathbf{I}$ to the $M \times M$ sample auto-correlation matrix $\hat{\mathbf{R}}_{rr}$ in the RLS algorithm, where α is a positive constant, all the M individual eigen values can be increased. This might lead to a reduction in the eigenvalue spread as some of the smaller eigenvalues might be increased and compressed at the loading level α. Therefore, some additional sidelobe suppression might be achieved in the RLS algorithm by doing explicit diagonal loading. Note that this is different from the regularization term needed in the beginning of the RLS algorithm when data might be insufficient to ensure the invertibity of the sample auto-correlation matrix. This regularization term ξ decays at a rate of $O(1/n)$, and therefore is trivial in comparison to the explicit diagonal loading term. That is, the RLS algorithm with explicit diagonal loading, implicitly computes the solution $\mathbf{f}_n^{DLRLS} = ((\frac{\xi}{n} + \alpha)\mathbf{I} + \hat{\mathbf{R}}_{rr})^{-1}\hat{\mathbf{p}}_{dr}$, where $\hat{\mathbf{p}}_{dr}$ is the sample cross-correlation vector between the receiver input and desired symbols. In our experiments, we have chosen $\xi = 10^{-4}$ and $\alpha = 10^{-3}$.

In this experiment, we assume that there is no interferer present in the training, and that an Adjacent Channel Interferer appears only during the data transmission. We plot the BER curves for IPM2, and for RLS, both with and without explicit diagonal loading. For IPM2, based on Fig. 6.9, we choose $\beta = 0.85$. We provide the BER curves using the optimal

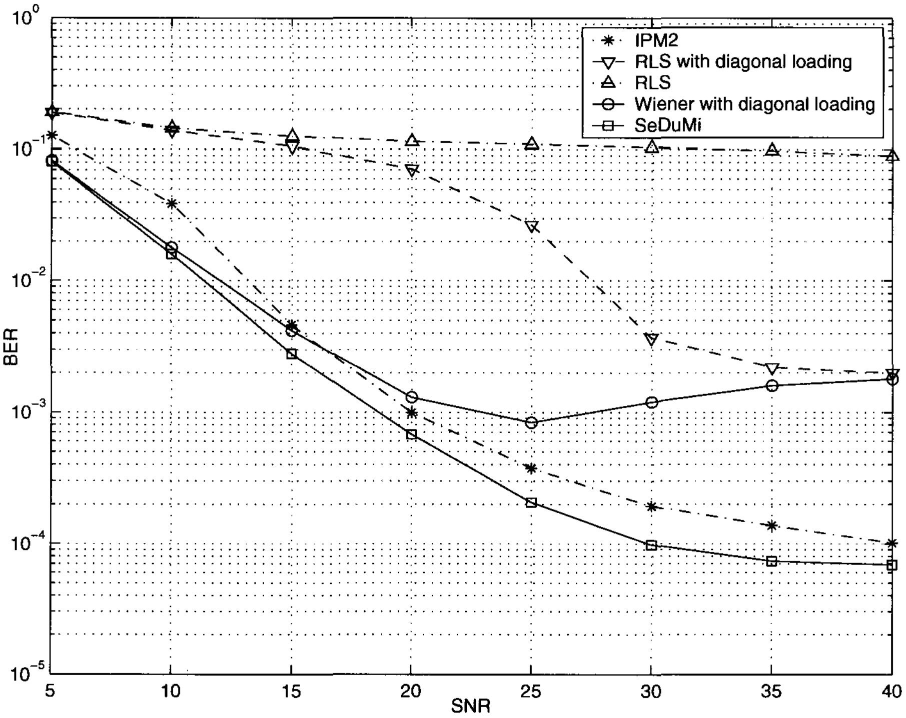

Figure 6.10: Equalization performance for IPM2 and diagonally loaded RLS along with their asymptotic performance. $\beta = 0.85$, ISR=0 dB.

equalizers calculated with true statistics, which are used as a benchmark against which the performance of our filter estimates is measured. All the results are shown in Fig. 6.10, where the interferer-to-signal ratio

$$\text{ISR} = \frac{\text{power of interferer}}{\text{power of desired signal}}$$

is set to 1, i.e., 0 dB. We see that the standard RLS algorithm is extremely sensitive to the interferer in the sidelobe region. Its BER performance is severely affected and is clearly not a suitable choice for environments in which intermittent ACI is expected. The explicit diagonal loading slightly improves the performance, in particular in the higher SNRs, but is eventually restricted due to the presence of the diagonal loading itself. The performance of IPM2 is however significantly better then diagonally loaded RLS especially in the normal operational region where the SNR is around $15 - 20$ dB. The limiting performance of the algorithm at higher SNR is due to the residual ISI induced due to the sidelobe constraints.

We observe that the BER performance of the diagonally loaded Wiener Filter appears to be counter-intuitive at first sight, since we might expect the BER performance to improve with increasing SNR. However, it is important to note that the diagonal loading term is a source of errors; like noise, it corrupts the sample auto-correlation matrix. One plausible explanation for the rise in BER performance can be that for higher SNRs, due to less external noise, the Wiener filter tends to have higher sidelobes, and hence is more susceptible to ACI. Hence the rise in BER performance with increase in SNR. Therefore, at a higher SNRs the number of errors attributable to interferer increases faster than those due to noise decrease; hence the rise in BER at higher SNRs.

Experiment 3

In the next experiment, we consider three scenarios namely, (a) no interferer present at any time during transmission, (b) interferer present only during the data phase of transmission

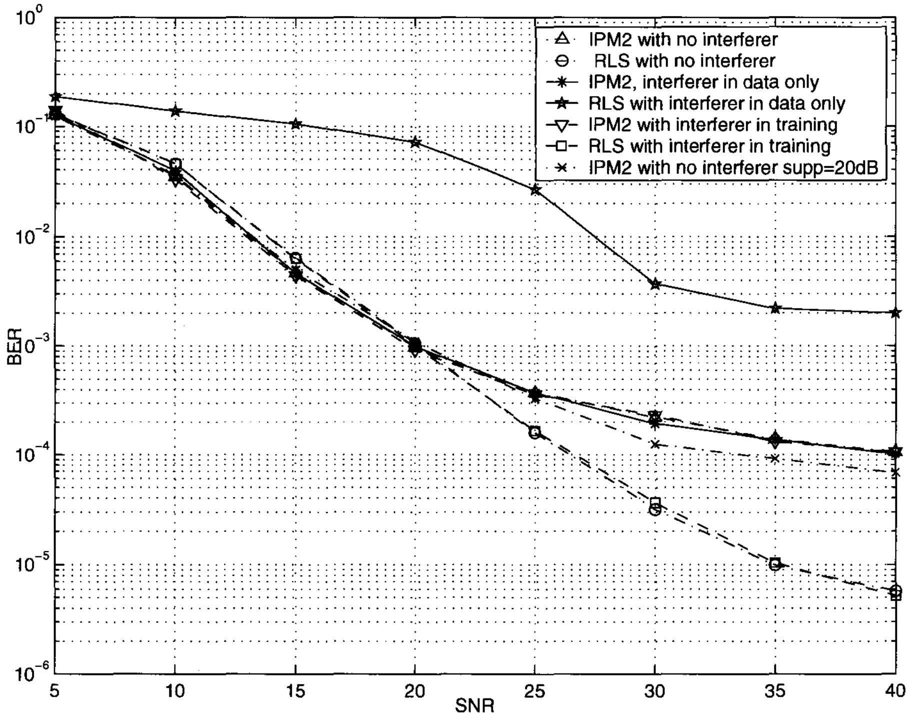

Figure 6.11: Equalization performance of IPM2 and diagonally loaded RLS in different scenarios. $\beta = 0.85$, ISR=0 dB.

and, (c) interferer present in the training as well as the data mode. All three are for a sidelobe suppression of 30 dB. We analyse the BER performance of IPM2 and diagonally loaded RLS algorithm in all the three cases. The results are shown in Fig. 6.11. We see that IPM2 performance in all three cases remains fairly similar. Therefore, it provides a dependable receiver, robust to unwanted interferers in the sidelobe region which may strike occasionally or be present all the time. We also show the BER performance of the proposed equalizer in the absence of the interferer for a sidelobe suppression of 20 dB, rather than 30 dB. At high SNRs, the performance is improved, because the relaxed constraint allows greater ISI suppression. This comparison provides an indication of the way in which the flexibility to choose the desired sidelobe level can be exploited by the designer.

Fig. 6.11 shows that the diagonally loaded RLS performance fluctuates significantly, becoming much worse if an interferer strikes in the data mode. The presence of diagonal loading also limits its BER performance for higher SNRs in all the three scenarios for RLS. This happens when the distortion due to the diagonal loading exceeds the distortion due to noise. Therefore, in a practical scenario, a receiver based on diagonally-loaded RLS remains susceptible to any intermittent interferences in the sidelobe region. On the other hand, IPM2 provides us with the necessary robustness to suppress any such undesirable interferers. We also observe that IPM2 outperforms diagonally loaded RLS in the normal operational SNR around $15-20$ dB. This is because the number of errors due to noise in this range are greater than the errors due to residual ISI attributed to the sidelobe constraints. At higher SNRs however, the number of errors due to noise is lower than the number of errors due to the residual ISI.

Experiment 4

All the previous simulations for symbol-by-symbol adaptive equalization scenario used the IPM2 approach. However, for the same experiments the standard barrier method (SBM) approach can also be used. In this experiment, we compare the performances of the IPM2

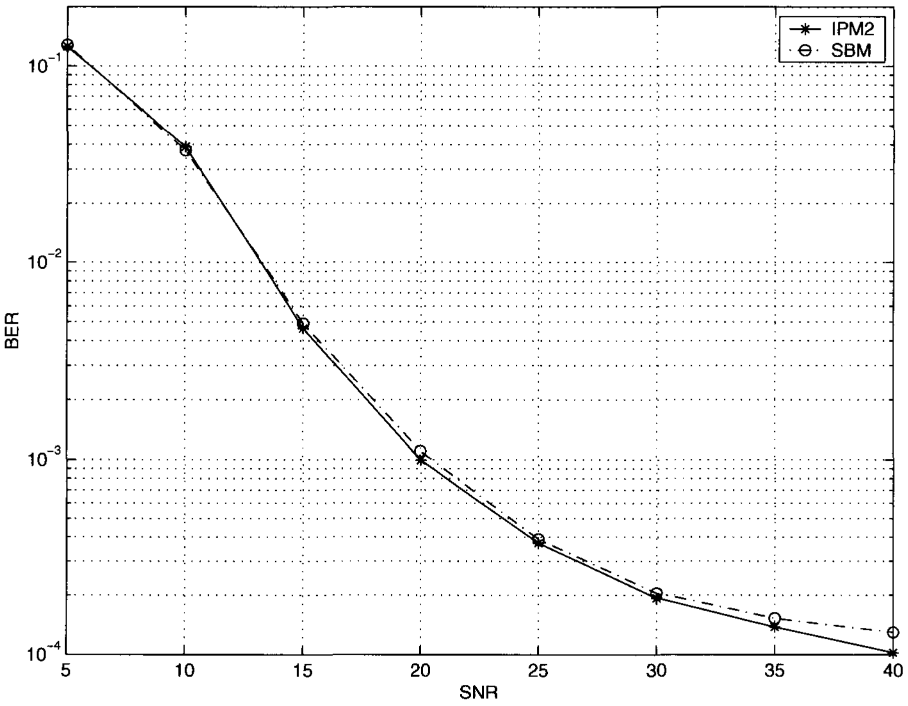

Figure 6.12: Equalization performance of IPM2 and SBM. $\beta = 0.85$, $\mu = 1.2$, ISR=0 dB.

and SBM based algorithms. We test the two algorithms in a scenario in which the interferer is present only in the data mode. As shown in Fig. 6.12 both algorithms give similar performance though IPM2 performs better towards the higher SNRs because the objective in IPM2 is inside a logarithm. However, both algorithms perform well and can be used to provide a receiver which ensures robustness to intermittent interferers in the sidelobe region.

Chapter 7

Summary and Conclusion

In this work we developed block adaptive and symbol-by-symbol adaptive fractionally-spaced equalization algorithms which are insensitive to ACI. This is an important requirement in modern day WCDMA based 3G cellular systems. We attempted to combat ACI by imposing multiple quadratic inequality constraints at discrete frequency points in the sidelobe region of the magnitude response of the equalizer. In Chapter 3 we wrote our adaptive equalization problem as an optimization problem with multiple quadratic inequality constraints. We modeled this problem as a Second Order Cone Program which was solved using well established interior point methods [4]. Thereby we presented an efficient approach to solve our optimization problem. However, this technique is limited to block adaptive implementations.

In Chapter 4 we analysed generic convex optimization problems with multiple inequality constraints. We demonstrated the potential application of barrier methods in solving such problems. We gained a deeper understanding of the theory in light of the concept of the central path, which is fundamental to interior point methods. This presented itself as a promising technique, but required multiple Newton steps to stay close to the central path. We developed and discussed two approaches based on the method of analytic centers (MAC) and the standard barrier method.

In Chapter 5 we successfully developed practically implementable interior point based algorithms which are able to perform symbol-by-symbol adaptive fractionally-spaced equalization with multiple inequality constraints. These algorithms, namely IPM2 and SBM are modifications of the MAC and standard barrier methods.

In Chapter 6 we demonstrated the applicability of these algorithms using computer simulations. We separately analysed the performance in both the block adaptive and symbol-by-symbol adaptive scenarios. We used limited training in both scenarios, and compared our performance in the presence of ACI with a RLS algorithm with explicit diagonal loading. We demonstrated that with explicit control over the sidelobe region, the designer can adjust the sidelobe levels at a desirable level, and thus can combat ACI much more effectively then the RLS algorithm with explicit diagonal loading.

The computational cost for IPM2 and SBM algorithms is $O(M^3)$, which is moderately higher then the computational cost for RLS $O(M^2)$. However in a communication scenario prone to ACI, such robust algorithms might prove to be indispensable.

7.1 Future Work

In this work we have solved the core problem of effectively performing symbol-by-symbol adaptive equalization using a linear equalizer over block fading channels. An obvious extension to the decision-directed mode is possible, when the probability of bit error is low. It can also be extended to a decision-feedback equalizer (DFE) over time-varying channels in a straight forward manner. Another area that needs attention is reducing the computational complexity of the algorithms. The use of conjugate-gradient methods does appear to be an attractive choice for this purpose, but such an approach might require more training. Another interesting possibility is the use of active set methods which are discussed in the next section.

Active Set Methods

An attractive avenue to reduce the computational complexity is to use Active Set Methods. For an inequality constrained convex optimization problem, the active set at any feasible point is the set of "active" inequality constraints for which the equality condition holds. Therefore, it might lead to substantial complexity reduction if applied efficiently. However, there are some of the potential problems associated with active set methods. The main challenge would be to determine the active set at the optimal solution at each stage of the algorithm. To handle this issue one natural approach could be to "guess" the active set, and refresh it recursively with successive iterations. This could however be a slow process, and might impose a lower bound on the number of iterations needed to reach optimality. To handle this problem, there could be approaches like using a "gradient projection method" [24] for rapid changes in the active set. Another potential problem in using active set method can be the existence of "strongly" active and "weakly" active constraints. This depends on the sensitivity of the optimal objective value to the presence of the constraint. The potential problem that may arise here is that sometimes it may be difficult for the algorithm to decide which of active constraints at the optimal solution are weakly active. This indecisiveness in the algorithm might result in successive iterations being wasted.

Therefore, using an active set method could provide an inexpensive route towards the optimal solution. However, it might require a large number of steps. Nevertheless, this is an exciting field to explore to achieve a lower computational complexity at an acceptable number of iterations.

Semi-infinite Constraints using SDP

In Chapter 3, instead of imposing a finite number of inequality constraints, it might be interesting to impose infinite constraints using semi-definite programming (SDP) to achieve "strict" control of the sidelobe regions using techniques developed in [5], [36]. Furthermore, it would be interesting to explore any possibilities of achieving strict sidelobe control in the

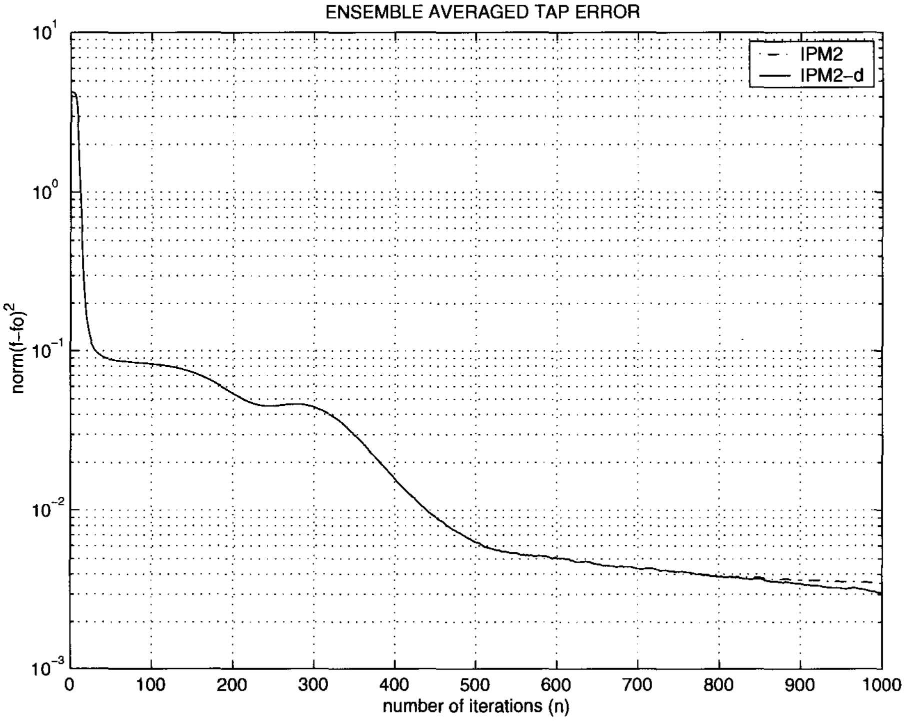

Figure 7.1: Comparison of IPM2 with IPM2-d

symbol-by-symbol adaptive scenario.

Improving on IPM1

One very interesting possibility is to come up with a suitable dependence of τ_n in the IPM1 algorithm. The present dependence, as used in the IPLS algorithm [10], is unsuitable for inequality constrained problems. This is so because it was originally proposed for unconstrained problems. Now τ_n should change in such a way that it takes into account the inequality constraints as well, and manages to keep the solution close to the central path in a single Newton step.

Improving on IPM2 and SBM

Another interesting avenue is the potential scope for improvement on the IPM2 and SBM algorithms provided in the thesis. We rely on the use of a damped Newton approach close to the boundary. Another possibility might be to take a second Newton step, when close to the boundary. By doing so, the solution should tend to stay closer to the central path. This appears to be better than spending iterations in finding a suitable direction at a point closer to the boundary. For example, in the IPM2 algorithm, instead of making ρ monotonic decreasing, we can make it monotonic non-increasing. That is, rather than updating ρ at each iteration, we revert back to the previous value of ρ, when very close to the boundary. Thus, we should stay closer to the central path. Although this approach (say, IPM2-d) provided with only a minor improvement over the IPM2 algorithm (see Fig. 7.1), for a training of 1000 symbols and $\beta = 0.975$, it might be interesting to investigate further in this direction.

Appendix A

The IPLS algorithm

The IPLS algorithm was originally proposed for unconstrained optimization problems [10]. In the IPLS algorithm, the parameter τ_n attempts to drive the algorithm to optimality. However, the definition of τ_n in [10], which does not treat inequality constraints, is inappropriate for constrained problems and severely limits the movement of the solution on the central path. For the interested reader, a mathematical treatment of the IPLS algorithm for the constrained optimization problem discussed in Chapter 5 is presented below. We assume that the reader is familiar with the notations, symbols, and the discussion provided in Chapter 5.

IPLS is an analytic center based estimator. Therefore, it defines a potential function $\psi_n(\mathbf{f})$, and tries to find a "center" for this potential function. The gradient and Hessian of $\psi_n(\mathbf{f})$ are given by

$$\nabla\psi_n(\mathbf{f}) = \frac{\nabla J_n(\mathbf{f})}{s_n(\mathbf{f})} + \left(\frac{1}{N_d}\sum_{i=1}^{N_d}\frac{2\mathbf{s}_i\mathbf{s}_i^H\mathbf{f}}{\gamma_i(\mathbf{f})} + \frac{2\mathbf{f}}{t_n(\mathbf{f})}\right) \tag{A.1}$$

and

$$\nabla^2\psi_n(\mathbf{f}) = \frac{\nabla^2 J_n(\mathbf{f})}{s_n(\mathbf{f})} + \frac{\nabla J_n(\mathbf{f})\nabla J_n(\mathbf{f})^H}{s_n^2(\mathbf{f})} + \left(\frac{1}{N_d}\sum_{i=1}^{N_d}\frac{2\mathbf{s}_i\mathbf{s}_i^H}{\gamma_i(\mathbf{f})} + \frac{1}{N_d}\sum_{i=1}^{N_d}\frac{4\mathbf{s}_i\mathbf{s}_i^H\mathbf{f}\mathbf{f}^H\mathbf{s}_i\mathbf{s}_i^H}{\gamma_i^2(\mathbf{f})} + \frac{4\mathbf{f}\mathbf{f}^H}{t_n^2(\mathbf{f})} + \frac{2\mathbf{I}}{t_n(\mathbf{f})}\right). \tag{A.2}$$

The gradient and Hessian of $J_n(\mathbf{f})$ are given by

$$\nabla J_n(\mathbf{f}) = -2\hat{\mathbf{p}}_{dr}(n) + 2\hat{\mathbf{R}}_{rr}(n)\mathbf{f}$$
$$\nabla^2 J_n(\mathbf{f}) = 2\hat{\mathbf{R}}_{rr}(n),$$

respectively. The IPLS algorithm is given in Table A.1.

This algorithm works satisfactorily for the unconstrained least squares problem where $\nabla J_n(\mathbf{f}) \rightarrow 0$ towards the optimal solution. However, in the presence of multiple quadratic sidelobe constraints $\nabla J_n(\mathbf{f})$ does not become vanishingly small at the optimal solution, due to the sidelobe constraints. Hence, using this algorithm in a scenario with multiple inequality constraints does not yield very good results.

Table A.1: IPLS

- **Step 1: Initialization.** Let $\varepsilon, N_d, R, \beta$ be given. Set

$$\mathbf{f}_o = \mathbf{0}, \quad \mathbf{p}_{dr}(0) = \mathbf{0}, \quad \mathbf{R}_{rr}(0) = \mathbf{0}, \quad \nabla J_0(\mathbf{0}) = \mathbf{0}.$$

- **Step 2: Updating.** For $n \geq 1$, update $\mathbf{r}_{2n}$ and $d(n-\delta)$, and then update

$$\hat{\mathbf{p}}_{dr}(n) = \frac{n-1}{n}\hat{\mathbf{p}}_{dr}(n-1) + \frac{1}{n}\mathbf{r}_{2n}d^*(n-\delta) \tag{A.3a}$$

$$\hat{\mathbf{R}}_{rr}(n) = \frac{n-1}{n}\hat{\mathbf{R}}_{rr}(n-1) + \frac{1}{n}\mathbf{r}_{2n}\mathbf{r}_{2n}^H. \tag{A.3b}$$

Update

$$\nabla J_n(\mathbf{f}_{n-1}) = -2\hat{\mathbf{p}}_{dr}(n) + 2\hat{\mathbf{R}}_{rr}(n)\mathbf{f}_{n-1}$$

$$t_n(\mathbf{f}_{n-1}) = R^2 - \|\mathbf{f}_{n-1}\|^2$$

$$\tau_n = J_n(\mathbf{f}^a_{n-1}) + \beta\frac{R}{\sqrt{2}}\|\nabla J_n(\mathbf{f}^a_{n-1})\|_2$$

$$\nabla\psi_n(\mathbf{f}_{n-1}), \quad \nabla^2\psi_n(\mathbf{f}_{n-1})$$

which are given by (A.1) and (A.2).

- **Step 3: Recentering.** The new approximate center of Ω_n is obtained by taking one Newton step with $\mathbf{f}_{n-1}$ as the starting point,

$$\mathbf{f}_n = \mathbf{f}_{n-1} - \left(\nabla^2\psi_n(\mathbf{f}_{n-1})\right)^{-1}\nabla\psi_n(\mathbf{f}_{n-1}).$$

Set $n := n+1$, and return to **Step 2**.

Appendix B

Damped Newton Method

Here we provide the damped Newton approach [8, pp. 487] used in IPM2 and SBM. We assume that reader is familiar with the notations, symbols, and the problem format discussed in Chapter 5. Only one Newton step is taken in each iteration to control computational cost. Close to the boundary, the Hessian matrix can become ill-conditioned and a conventional Newton update may produce an infeasible point. Such an outcome is obviously undesirable in an interior point method. To overcome this effect, we will adopt a conservative approach that enables us to avoid "wasting" iterations in an inappropriate direction. If a proposed Newton step breaks the boundary for a given iteration, then we will not perform the update. Instead we will use the feasible starting point of the present iteration as the starting point for the next iteration. We will the update the Newton direction in the standard way and will introduce damping (i.e., reduce step-size), in order to make conservative steps towards an optimal solution.

Table B.1: Damped Newton Method

- **Starting point:** Let the cost function $\psi(\mathbf{f})$ and a starting point $\mathbf{f}_o$ be given. Set, scaling factor $\alpha = 1$, damping factor $\lambda = 0.5$, $(0 < \lambda \leq 1)$, and tolerance threshold $\eta = \zeta \times 10^5$, $(\eta \ll 1)$, where $\zeta = 2.2204 \times 10^{-16}$ is the machine accuracy.

- **Step 1: Newton Update** For $n \geq 1$,

$$\text{Update} \quad \mathbf{f}_n = \mathbf{f}_{n-1} - \alpha\big(\nabla^2\psi_n(\mathbf{f}_{n-1})\big)^{-1}\nabla\psi_n(\mathbf{f}_{n-1})$$

$$\text{and calculate} \quad \gamma_i(\mathbf{f}_n) = \varepsilon^2 - |\mathbf{s}_i^H\mathbf{f}_n|^2 \quad i = 1, 2, ..., N_d$$

- **Step 2: Stopping Criterion** For $n \geq 1$,

$$\text{If} \quad \gamma_i(\mathbf{f}_n) > \eta \quad \text{for all} \quad i = 1, 2, ..., N_d$$

$$\text{Reset} \quad \alpha = 1$$

$$\text{Set} \quad n := n + 1$$

and return to **Step 1**.

$$\text{Else if} \quad \gamma_i(\mathbf{f}_n) \leq \eta \quad \text{for some} \quad i = 1, 2, ..., N_d$$

$$\text{Reset} \quad \mathbf{f}_n = \mathbf{f}_{n-1}$$

$$\text{Set} \quad \alpha = \lambda\alpha$$

$$\text{Set} \quad n := n + 1$$

and return to **Step 1**.

Bibliography

[1] S. U. H. Qureshi, "Adaptive equalization", *Proceedings of the IEEE*, vol. 73, pp. 1349-1387, Sept. 1985.

[2] J. G. Proakis, *Digital Communications*, fourth edition, McGraw-Hill, New York, 2000.

[3] S. Haykin, *Adaptive Filter Theory*, fourth edition, Prentice-Hall, New Jersey, 2002.

[4] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones",*Optimization Methods and Software*, vol. 11-12, pp. 625-653, Aug. 1999.

[5] T. N. Davidson, Z. -Q. Luo, and J. F. Sturm, "Linear matrix inequality formulation of spectral mask constraints", *IEEE Transactions on Signal Processing*, vol. 50 , no. 11, pp. 2702-2715, Nov. 2002.

[6] S. -P. Wu, S. Boyd, and L. Vandenberghe, "FIR filter design via spectral factorization and convex optimization", in *Applied and Computational Control, Signals, and Circuits*, B. Datta, Ed., vol. 1., Birkhauser, Boston, 1999.

[7] J. Liu, A. B. Gershman, Z. Q. Luo and K. M. Wong, "Adaptive beamforming with sidelobe control: a second-order cone programming approach", *IEEE Signal Processing Letters*, vol. 10, no. 11, pp. 331-334, Nov. 2002.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.

[9] K. H. Afkhamie, *Interior Point Least Squares Estimation*, Ph.D. Thesis, McMaster University, Hamilton, Ontario, Canada, May 2000.

[10] K. H. Afkhamie, Z.-Q, Luo, and K. M. Wong, "Adaptive linear filtering using interior point optimization techniques," *IEEE Transactions on Signal Processing*, vol. 48, no. 6, pp. 1637-1648, June 2000.

[11] K. H. Afkhamie, T. N. Davidson, and Z. -Q, Luo, "Constrained adaptive estimation using interior point optimization," in *Proceedings of the International Symposium on Image, Speech, Signal Processing and Robotics*, pp. 21-26, Hong Kong, Sep. 1998.

[12] B. D. Carlson, "Covariance matrix estimation errors and diagonal loading in adaptive arrays", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 4, pp. 397-401, July 1988.

[13] D. K. Borah, R. A. Kennedy, Z. Ding, and I. Fijalkow, "Sampling and prefiltering effects on blind equalizer design", *IEEE Transactions on Signal Processing*, vol. 49, no. 1, pp. 209-218, Jan. 2001.

[14] R. Schober and W.H. Gerstacker, "On the distribution of zeros of mobile channels with application to GSM/EDGE", *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1289-1299, July 2001 .

[15] K. H. Afkhamie, Z.-Q, Luo, and K. M. Wong, "Interior point least squares estimation: Transient convergence analysis and application to MMSE decision-feedback equalization", *In Proceedings of the IEEE Transactions on Signal Processing*, vol. 49, no. 7, pp. 1543-1555, July 2001.

[16] G. Ungerboeck, "Fractional tap-spacing equalizer and consequences for clock recovery in data modems", *IEEE Transactions on Communications*, vol. 24, no. 8, pp. 856-864, Aug 1976.

[17] T. N. Davidson, Z. -Q. Luo, and K. M. Wong, "Design of orthogonal pulse shapes for communications via semidefinite programming", *IEEE Transactions on Signal Processing*, vol. 48, no. 5, pp. 1433-1445, May 2000.

[18] J. Liu, *Adaptive Beamforming with Strict Sidelobe Control*, M.A.Sc. Thesis, McMaster University, April 2003.

[19] G. L. Stuber, *Principles of Mobile Communication*, second edition, Kluwer Acsdemic Publishers, Boston, 2001.

[20] A. V. Oppenheim, R. W. Schafer and J. R. Buck, *Discrete-Time Signal Processing*, second edition, Prentice Hall, New Jersey, 2000.

[21] T. N. Davidson, "Efficient design of waveforms for robust pulse amplitude modulation", *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3098-3111, Dec. 2001.

[22] J. R. Treichler, I. Fijalkow, and C. R. Johnson, Jr., "Fractionally spaced equalizers: How long should they really be ?", *IEEE Signal Processing Magzine*, vol. 13, pp. 65-81, May 1996.

[23] R. D. Gitlin and S. B. Weinstein, "Fractionally spaced equalization: An improved digital transversal equalizer", *Bell System Technical Journal*, vol. 60, pp. 275-296, Feb. 1981.

[24] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 2000.

[25] P. P. Vaidyanathan and B. Vrcelj, " Biorthogonal partners and applications", *IEEE Transactions on Signal Processing*, vol. 49, no. 5, pp. 1013-1027, May 2001.

[26] C. B. Papadias and D. T. M. Slock, "Fractionally spaced equalization of linear polyphase channels and related blind techniques based on multichannel linear prediction", *IEEE Transactions on Signal Processing*, vol. 47, no. 3, p.p. 641-654, March 1999.

[27] C. R. Johnson, Jr., P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown and R. A. Casas, "Blind equalization using the constant modulus criterion: A review", *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1927-1950, Oct. 1998.

[28] Z.-Q. Luo, "Applications of convex optimization in signal processing and digital communication", *Mathematical Programming*, series B, pp. 177-207, 2003.

[29] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems*, Plenum Press, New York, 1992.

[30] D. D. Falconer, F. Adachi and B. Gudmundson, "Time division multiple access methods for wireless personal communications", *IEEE Communications Magazine*, vol. 33, no. 1, pp. 50-57, Jan. 1995.

[31] J. Dumont, S. Lasaulce and J.-M. Chaufray, "Adjacent channel interference in WCDMA networks equipped with multiple antennas mobile stations", in *Proceedings of the 5th IEEE Workshop on Signal Processing Advances in Wireless Communications*, pp. 512-516, Lisbon, July 2004.

[32] S. W. Wales, "Modulation and equalisation techniques for HIPERLAN", *5th IEEE International Symposium on Wireless Networks - Catching the Mobile Future*, vol. 3, pp. 959 - 963, Sep 1994.

[33] J. G. Proakis, "Adaptive equalization for TDMA digital mobile radio", *IEEE Transactions on Vehicular Technology*, vol. 40, no. 2, p. p. 333-341, May 1991.

[34] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming", *Linear Algebra and Its Applications*, 284:193-228, Nov. 1998.

[35] S. Hamalainen, H. Lilja, A. Hamalainen, "WCDMA adjacent channel interference requirements", in *Proceedings of the 50th IEEE Vehicular Technology Conference*, vol. 5, pp. 2591 - 2595, Sep. 1999.

[36] M. R. Wilbur, T. N. Davidson and J. P. Reilly, "Efficient design of oversampled NPR GDFT filterbanks", *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1947-1963, July 2004.

[37] http://www.3GPP.org/