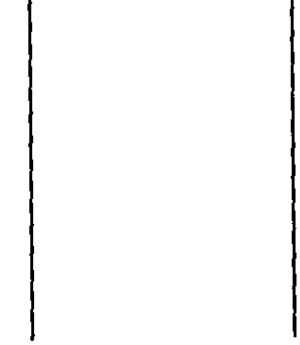

Estimation and Detection Techniques for Doubly Selective Channels in Wireless Communications

Thesis submitted to Loughborough University in candidature for the degree of Doctor of Philosophy.

Muhammad Tariq Nawaz Qaisrani

Advanced Signal Processing Group
Loughborough University
2008

Abstract

A fundamental problem in communications is the estimation of the channel. The signal transmitted through a communications channel undergoes distortions so that it is often received in an unrecognizable form at the receiver. The receiver must expend significant signal processing effort in order to be able to decode the transmit signal from this received signal. This signal processing requires knowledge of how the channel distorts the transmit signal, i.e. channel knowledge. To maintain a reliable link, the channel must be estimated and tracked by the receiver.

The estimation of the channel at the receiver often proceeds by transmission of a signal called the 'pilot' which is known a priori to the receiver. The receiver forms its estimate of the transmitted signal based on how this known signal is distorted by the channel, i.e. it estimates the channel from the received signal and the pilot. This design of the pilot is a function of the modulation, the type of training and the channel.

The pilot can be multiplexed into the unknown data in what is called time multiplexed or explicit training. It may also be added at low power onto the unknown data in a scheme called superimposed training. In this thesis, iterative semiblind channel estimation based on superimposed training is studied for orthogonal frequency division multiplexing in frequency selective channels. Iterative semiblind estimation of doubly selective channels with superimposed training is then studied for single carrier communications. Finally, iterative semiblind estimation of doubly selective channels based on time multiplexed training is studied. Significant performance gains are demonstrated with the application of iterative semiblind channel estimation in all three cases.

Contents

Statement of Originality

The novelty of chapter 3 is in semiblind channel estimation based on superimposed training for orthogonal frequency division multiplexing. The contribution of chapter 4 is to the application of iterative semiblind channel estimation based on superimposed training for basis exponential model based doubly selective channels. The contribution of chapter 5 is the application of iterative semiblind channel estimation based on time multiplexed training for basis exponential model based doubly selective channels. The novelty of the work is attested by the following publications:

1. M. Qaisrani, S. Lambotharan, 'An Iterative (Turbo) Channel Estimation and Symbol Detection Technique for Doubly Selective Channels', in IEEE Vehicular Technology Conference, 2007 Spring, Dublin, pp 2253 - 2256.

2. M. Qaisrani, S. Lambotharan, 'An Iterative (Turbo) Channel Estimation and Symbol Detection Technique for Doubly Selective Channels', in IEEE Vehicular Technology Conference, 2008 Spring, Singapore, pp 1316 - 1319.

3. M. Qaisrani, S. Lambotharan, 'Semiblind Channel Shortening based on Superimposed Training for MIMO OFDM', submitted to IET Proceedings on Communications, 28 April 2008.

4. M. Qaisrani, S. Lambotharan, 'Iterative Estimation, Equalization and Decoding For Doubly Selective MIMO Channels', submitted to IET Proceedings on Communications, 05 May 2008.

5. M. Qaisrani, S. Lambotharan, 'Superimposed Training with Turbo Equalization for Doubly Selective Channels', submitted to IET Proceedings on Communications, 08 June 2008 .

Acknowledgements

I would like to thank my supervisor Dr. Sangarapillai Lambotharan for his invaluable support and guidance throughout this PhD. His patience with my ignorance never ceased to surprise me.

I would like to thank the external and internal examiners whose insightful comments have greatly helped to improve the readability of this work. I am also grateful to all my friends at the Advanced Signal Processing Group for providing a stable environment within the Advanced Signal Processing group. In particular Vimal Sharma, Mohsin Naqvi, Kanapathipillai Cumanan, and Ranaji Krishna deserve special mention.

Finally, I must express my gratitude to my family for their unconditional love & prayers for my success and to Allah who gives nothing to those who keep their arms crossed.

List of Acronyms

ADSL	Asymmetric Digital Subscriber Line
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelenik and Raviv algorithm
BEM	Basis Expansion Model
BER	Bit error rate
BLAST	Bell Labs Layered Space Time
BLUE	Best Linear Unbiased Estimator
CMA	Constant Modulus Algorithm
COFDM	Coded Orthogonal Frequency Division Multiplexing
CRLB	Cramer Rao Lower Bound
CSI	Channel State Information
DFT	Discrete Fourier Transform
DMT	Discrete Multitone
DVB	Digital video broadcast
ECC	Error Control Coding
EXIT	Extrinsic Information Transfer Chart

FEC	Forward Error Correction
FMT	Filtered Multitone
GSM	Global System for Mobile Communications
HDTV	High definition television
IBI	Inter-block Interference
ICI	Intercarrier Interference
IIR	Infinite Impulse Response
ISI	Intersymbol Interference
LTI	Linear Time Invariant
LTV	Linear Time Variant
MAI	Multiple Access Interference
MAP	Maximum a posteriori probability
MERRY	Multicarrier Equalization by Restoration of Redundancy
MIMO	Multiple Input Multiple Output
MIMO-MA	Multiple input multiple output - multiple access
MISO	Multiple Input Single Output
ML	Maximum Likelihood
MLSE	Maximum Likelihood Sequence Estimation
MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error
MSI	Multistream Interference

MSSNR	Maximal Shortening Signal to Noise Ratio
MVU	Minimum Variance Unbiased Estimator
OFDM	Orthogonal Frequency Division Multiplexing
OSIC	Ordered Successive Interference Cancellation
PCCC	Parallel Concatenated Convolutional Code
PSAM	Pilot Symbol Assisted Modulation
QAM	Quadrature Amplitude Modulation
SAM	Sum Squared Autocorrelation Minimization Algorithm
SIMO	Single Input Multiple Output
SINR	Signal to Interference Noise Ratio
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
SOVA	Soft Output Viterbi Algorithm
ZF	Zero Forcing

List of Symbols

$\lfloor . \rfloor$	Floor operator
N	Block length
M_T	Number of transmit antennas
M_R	Number of receive antennas
$\odot$	Hadamard product
$\otimes$	Kronecker product
$\mathbf{I}_K$	$K \times K$ Identity matrix
$\mathbf{F}$	DFT matrix
$\mathbf{1}_K$	$K \times 1$ vector of ones
$\mathbf{e}_{K,k}$	k^{th} column of a $K \times K$ identity matrix
L	Channel order
L_{cp}	Cyclic prefix order
$\mathbf{v}$	Noise
$\mathbf{x}$	Transmit signal
$\mathbf{y}$	Received signal
$\boldsymbol{s}$	Data for transmission

$\mathbf{c}$	Training
$\mathbf{H}$	Channel convolution matrix
$\mathbf{G}$	Block equalizer
$\mathbf{g}$	Serial equalizer
$E(.)$	Expectation operator
$I(.;.)$	Mutual information
$\|.\|_p$	l_p norm
$\|.\|$	Euclidean norm
$(.)^H$	Hermitian transpose operator
$(.)^T$	Transpose operator
$(.)^\dagger$	Pseudo-inverse
$\mathbf{\Lambda}$	Log likelihood ratios
$\mathbf{diag(b)}$	Diagonal matrix with vector
$\mathbf{R}$	Correlation matrix
$\mathbf{sgn}(.)$	Signum function
μ	Step size of gradient descent
$Q(.)$	Q function / Gaussian probability tail function

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Introduction

Communications has evolved a long way since the electrical telegraph. The invention of the radio telegraph by Guglielmo Marconi began the radio wireless revolution. Powerline communications technology has made it possible to connect users with reduced wiring. Light amplification by stimulated emission of radiation (Laser) has propelled growth in free space optics and fiber optic communications. The communications industry has seen unprecedented growth in the last couple of decades spurred on by Shannon's promise of arbitrarily reliable communication at previously unimaginable data rates.

Wireless communications has given the end user a degree of flexibility that would have been unimaginable to the technology outsider two decades ago. Indeed what we take for granted in today's networks had its detractors even in the communications field [2]. The technology has been made possible through the use of radio frequencies and substantial signal processing. Wireless communications technology has evolved into much more than the original concept of a cell phone. It has provided people with connectivity to the internet where wireline communications simply cannot provide services. Thanks to wireless communications, we can now be connected seamlessly with nothing but a portable hand held unit. Upcoming fourth generation systems aim to provide wireless connectivity on an anytime anywhere basis[1].

[1]Standards for fourth generation systems have not been developed yet but they are projected to be completely packet switched networks which aim to provide data rates from 100Mbs to 1Gbs on an anytime anywhere basis.

This revolution in wireless communications has further propelled research in wireline communications. In order to have any appeal to the customer, wireline communications must offer him or her substantially higher data rates to offset the advantages of ubiquitous connectivity. This rivalry between wireless and wireline communications has dramatically transformed the concept of communications. Today's end user requires much more than just voice calls and text messaging. Networks are willing to offer streaming rich multimedia content, video calls, high definition television (HDTV), digital video broadcast (DVB) and much more.

All these services require high data rates and come with increasing signal processing complexity for both wireless and wireline communications. The increase in complexity is more pronounced for wireless communications as the end user expects to be provided the services regardless of his location and any motion. Networks are also keen to profit by providing services everywhere. The signal processing at the receiver to maintain a wireless communications link requires knowledge of how the physical environment (channel) distorts the signal as it propagates through it. This knowledge is more difficult to obtain when the channel to the user is changing as the user is vehicle borne. The focus of this work is mainly on the estimation of such channels.

1.2 Thesis Outline

Many signal processing algorithms to extract the transmit information from the received signal require channel knowledge at the receiver. The channel needs to be estimated for the receiver signal processing. The focus of this work is on the study of channel estimation for communications.

Chapter 2 surveys techniques for processing the received signal to deal with channel distortions. Error control coding (ECC) is briefly introduced, with emphasis on convolutional and turbo codes, as an effective technique

to combat the effects of thermal noise. Equalization techniques that attempt to detect the transmit signal by inverse filtering are explained and the more complex sequence estimation algorithms are described. The decision feedback equalizer is introduced as a compromise between the two. Blind algorithms are also introduced as an alternative to the channel estimation approach we use in this thesis.

Chapter 3 focuses on orthogonal frequency division multiplexing (OFDM). Channel estimation based on superimposed pilots is explained and an iterative channel estimation approach is proposed. The design of channel shortening equalizers is explained. Two separate cases of single input multiple output (SIMO) and multiple input multiple output (MIMO) are considered. The MIMO-OFDM is space time coded for spatial multiplexing and successive interference cancelation is introduced and the comparison to the decision feedback equalizer drawn. Simulation results indicate significant performance enhancement with the proposed iterative approach.

Chapter 4 studies superimposed training for single carrier communications over doubly selective channels. The separate cases of single input single output (SISO) and MIMO are studied. The turbo equalization and space time turbo equalization architectures are explained and an iterative semiblind approach is proposed for decision directed channel estimation.

Chapter 5 studies doubly selective channel estimation using time multiplexed training. Optimal training is designed based on a training based capacity where the training placement, choice and number of training symbols and training power is optimized and a semiblind iterative channel estimation is proposed.

Conclusions are drawn in chapter 6. A brief summary is also provided and potential areas of future research are identified.

Chapter 2

CODING AND EQUALIZATION

2.1 Introduction

This chapter provides an overview of the the two most fundamental components in any wireless communications system, namely coding and equalization. A general block diagram of a communications system is given in figure 2.1. The channel coding and signal processing aspects are described next.

2.2 Coding

Contemporary communications systems usually employ matched filtering to maximize the signal to noise ratio at the receiver. Nevertheless, the received signal to noise ratio may still be insufficient to achieve the target bit error rate (BER). Historically, repetition coding was applied to meet the required BER performance until Shannon introduced the concept of channel capacity in his ground breaking paper [3]. This paper laid the foundations for entirely new fields like coding theory and information theory. For communications theory, the implications of Shannon's work were profound. His work laid bare the fallacy that BER performance alone is a benchmark in communications and instead showed that performance is appropriately measured by a three tuple of signal to noise ratio (SNR), the bit error rate (BER) and the data rate.

Shannon introduced the concept of channel capacity and established the existence of codes that could, at least theoretically, achieve it. However, the real journey from the uncoded BER to the capacity started afterwards. Shannon's codes were constructed from randomly chosen codewords that entailed excessive decoding complexity. Shannon opened the way for the real problem in coding theory, i.e. to design codes with reasonable encoding / decoding complexity that can get close to the capacity. From this perspective, repetition coding does not get anywhere because although the BER is reduced, so is the data rate.

Coding theory has produced numerous milestones in the quest to achieve channel capacity. Linear block and convolutional codes have been discovered along with the Viterbi and BCJR algorithms that allow these codes to be decoded with an acceptable complexity. The turbo principle has been applied to codes with remarkable results.

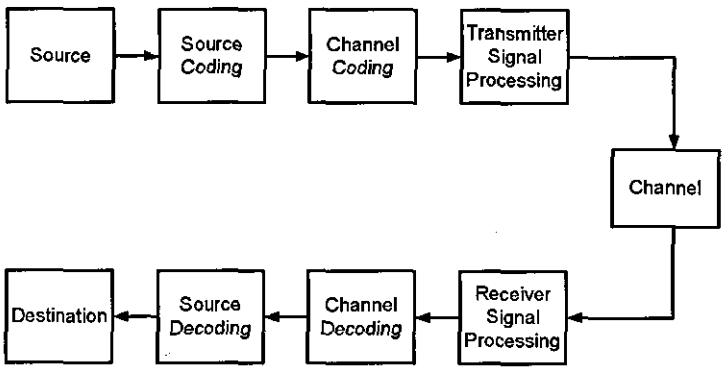

Figure 2.1. General communications block diagram.

Error correction codes introduce redundancy in the data stream before transmission in such a way that errors introduced can be corrected at the decoder. An (n, k) linear code can be considered as a k dimensional subspace in the n dimensional space $(\mathbb{F}_{q^m})^n$ over the finite field $\mathbb{F}_{q^m}$. These codes essentially try to place codewords as far apart as possible in all dimensions thereby introducing temporal diversity which averages out the effect of noise. Convolutional codes are the workhorse for wireless communications.

This chapter will therefore focus on convolutional and turbo codes.

2.2.1 Convolutional Codes

Binary linear convolutional codes have gained widespread popularity for use over the power limited regime. An $(n, 1, d_{free})$ binary linear convolutional encoder is a single input n output linear time invariant (LTI) system that may be defined using the D transform[1] as

$$\mathcal{C} = \{\mathbf{y}(D) = u(D)\mathbf{g}(D), u(D) \in \mathbb{F}_2((D))\} \tag{2.2.1}$$

where $\mathbf{g}(D)$ is $1 \times n$ vector of impulse responses, $\mathbb{F}_2((D))$ is the set of all formal Laurent series [4] and d_{free} is the free distance of the code. All vectors in this section are row vectors as opposed to the use of column vectors in the remainder of this dissertation. While this does inevitably introduce some unconformity, it has been done to maintain consistency with the respective fields of coding and communication theory which have used these notations.

Two generator n-tuples $\mathbf{g}(D)$ and $\mathbf{g}'(D)$ are equivalent if they generate the same code, i.e. if $\mathcal{C} = \mathcal{C}'$. In general, a minimal encoder is chosen for generating a code as it minimizes the complexity and ensures that the generator is not catastrophic[2]. A generator is minimal if and only if it is non catastrophic and there are no transitions to or from the zero state with all zero outputs except possibly a self loop.

Codes of a rate $\frac{k}{n}$ can be generated using either a $k \times n$ generator matrix of impulse responses or alternatively puncturing a rate $\frac{1}{n}$ code. Puncturing is achieved by omitting bits from the encoded sequence according to a puncturing matrix and it allows for a variable rate system without requiring

[1] The D transform is somewhat analogous to the z transform with two major differences. D is just a placeholder so that $\mathbf{u}(D)$ is still in the time domain and a unit delay is given by D rather than $\mathbf{z}^{-1}$. The D transform is thus not a transform in the strict sense as it only provides a convenient form of notation and does not transform from one domain to another.

[2] For a catastrophic generator, an infinite input sequence can generate a finite output sequence. Its state transition diagram thus has cycle with all zero outputs other than the zero state self loop

multiple encoders / decoders. This requires that the low rate mother code is chosen such that its codewords have the codewords of higher rate codes embedded in them and the puncturing matrix is accordingly chosen so that the higher rate code is retained after puncturing. Such mother codes are termed as rate compatible convolutional codes.

The parameter d_{free} is called the free distance of the code and it determines the error correction and error detection capabilities of the code. The free distance of a convolutional code is defined as the minimum distance between any two codewords. By the group property of linear codes, any sum of two codewords is a codeword. The free distance is then equal to the weight of the minimum weight codeword. The probability of an error event for the code may then be predicted with the union bound estimate as [4]

$$P_c = K_{min}(\mathcal{C})Q\left(\sqrt{\gamma_c \frac{2E_b}{N_o}}\right) \tag{2.2.2}$$

where $\gamma_c(\mathcal{C}) = \frac{kd_{free}}{n}$ is the nominal coding gain of the convolutional code $\mathcal{C}$ and K_{min} is the number of error events of weight d_{free} per unit time. The effective coding gain γ_{eff} depends both on γ_c and K_{min}. The function $Q(.)$ is called the Q function in communications and it measures the tail of a Gaussian pdf

$$Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{\frac{-x^2}{2}} dx \tag{2.2.3}$$

It is apparent from the union bound estimate that the key to the performance improvement offered by convolutional codes lies in the free distance properties of the convolutional code[3]. These codes have thus been traditionally designed and chosen to have large minimum distance to achieve better error performance. The BER performance of a good rate-$\frac{1}{3}$ convolutional code (i.e. a code with good free distance properties) and a punctured rate $\frac{1}{2}$ code obtained from this code are shown in figure 2.2. The BER performance of the equivalent recursive systematic convolutional code are also shown in

[3]The corresponding quantity in linear block codes is called the minimum distance.

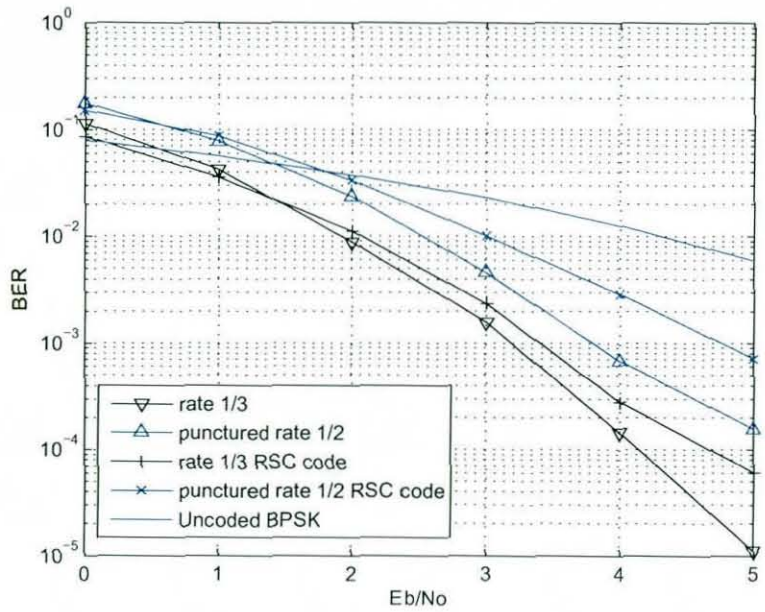

Figure 2.2. BER performance of rate convolutional codes with generator polynomial (31 27)

the figure along with the uncoded BER for reference.

2.2.2 Turbo Codes

Increasing constraint length implies strong coding behavior but also entails exponentially increasing computational complexity. Coding theory has worked around this by concatenating two shorter codes. Optimal decoding of these concatenated codes would still be computationally infeasible, but the concatenated structure allows for a computationally feasible albeit suboptimal approach. The codes can be decoded separately by using their own decoders in a concatenated structure. Both block and convolutional codes may be used and they may be concatenated serially or in parallel.

Concatenated codes [4] provide a tradeoff between coding gain and decoding complexity. The encoders can be concatenated in parallel so that they both operate on the same bit stream (turbo codes) or they may be concatenated in serial so that one operates on the output of the other (se-

rial concatenated convolutional codes). Serial concatenated codes have a much higher interleaver gain and minimum free distance than their equivalent parallel concatenated codes. Thus while parallel concatenated codes tend to perform better in the waterfall region of the bit error rate curve, the serial concatenated codes exhibit a much lower error floor. In this section, the focus is on turbo, hence parallel concatenated codes, but the concept of serial concatenated codes leads to a very useful concept of turbo equalization which will be discussed later. These concatenated codes operate in the low to moderate signal to noise ratio (SNR) region and they can operate within a fraction of a dB of the channel capacity. Turbo codes [4] were the first codes to operate within a dB of Shannon capacity [4, 5].

Turbo Encoder

Turbo codes are also called parallel concatenated convolutional codes (PCCC) because the turbo encoder is formed from the concatenation of two systematic (typically also recursive) convolutional encoders separated by an interleaver [6]. Thus the second encoder operates on an interleaved version of the input to the first encoder. This is illustrated in figure 2.3.

Interleaving plays a very crucial part in the performance of turbo codes. It is necessary for the suboptimum decoding algorithm based on information exchange between two component decoders that the inputs to the two encoders be independent of one another. The interleaver serves the purpose of decorrelating the inputs to the encoders. Consequently, there will be a high probability that the errors that remain uncorrected after one decoder can be corrected by the other decoder. There are different kinds of interleavers including block interleavers, convolutional interleavers, random interleavers, and code matched interleavers. Only recursive systematic convolutional (RSC) component encoders can produce an interleaving perfor-

[4] Low density parity check (LDPC) codes can operate even closer to the Shannon capacity and lack of any patents provides a very attractive incentive over turbo codes, nevertheless they have higher encoding complexity and decoding latency.

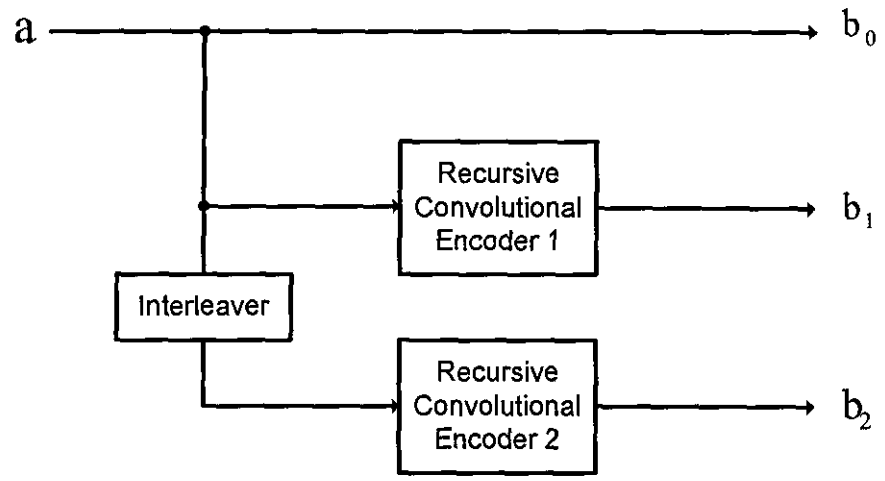

Figure 2.3. Turbo Encoder

mance gain which is vital for overall performance gain [7].

As with convolutional codes, trellis termination in a known state will improve performance in turbo codes. Recursive encoders cannot be terminated by a trailing sequence of zeros. Rather, they are terminated in a state of all zeros by switching the input to the feedback in the encoder for the termination sequence. Termination of both encoders in a known state is somewhat more complicated with negligible performance gain over termination of only one encoder in a known state and is therefore beyond the scope of this work.

We consider the most general case of a rate $\frac{1}{3}$ turbo code. Various codes of higher rates can be obtained from this code by puncturing, i.e. simply omitting bits of the code according to a puncturing pattern called the puncturing table. The receiver then simply decodes the code without assuming any channel observations related to the punctured bit. We used the recursive systematic convolutional encoders with the generator polynomial $[1 \quad 1 + D + D^2 + D^3 + D^4]$. Other rate codes can also be achieved using component encoders with different rates in the turbo encoder. For example, if the rate of the first encoder is R_1 and that of the second encoder is R_2,

then the rate of the overall turbo code R will satisfy [7]

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} - 1$$

Turbo Decoder

The turbo decoder is composed of two component decoders which may be based on the maximum aposteriori (MAP) algorithm or the soft output Viterbi (SOVA) algorithm. The decoders are concatenated with the code interleaver as shown in figure 2.4. The first MAP decoder takes the systematic sequence, the sequence produced by the first encoder and the apriori information from the second decoder to produce output extrinsic information on the decoded sequence. Similarly, the second decoder uses this a priori information in addition to the interleaved systematic information and the output sequence of the second encoder to decode. Thus information is iteratively passed from one decoder to another and this can improve performance over one shot decoding. This feedback loop in the decoder is similar to the feedback of engine heated air in the turbo engine, hence the name turbo codes. In fact, it is not the code, but this decoding algorithm that is responsible for the name 'turbo code'.

To explain the working of turbo algorithms, it is necessary to understand the concepts of a priori, intrinsic, a posteriori and extrinsic information. In an iterative algorithm with information exchange between constituent blocks, the a priori information for a constituent block is the information provided by the previous block. The intrinsic information about a bit is the information about that bit which is formed directly from its channel observations. The a posteriori information for any constituent block is the information generated at the output of that constituent block. The extrinsic information of a constituent block is obtained by subtracting the a priori and intrinsic information at the input of a constituent block from the a posteriori information at the output of the constituent block.

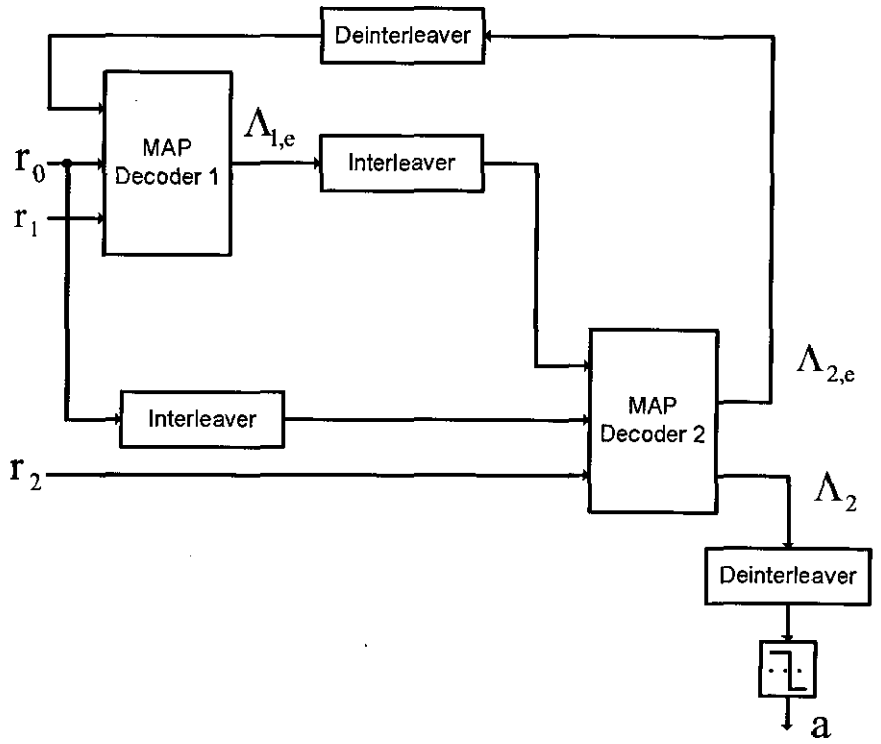

Figure 2.4. Turbo decoder block diagram.

It is essential to the performance of turbo algorithms that only extrinsic information is fed back. Feeding back intrinsic information generally results in faster convergence but the decoded data is typically far from the global minimum and could also result in limit cycles. Similarly, the a priori information must also be removed from the a posteriori information so that each constituent block only provides new information to the next block. Also interleaving is essential and the interleaver should decorrelate the sequences input to the two encoders, otherwise both decoders will fail to correct the same errors and the iterations will be useless. Although the intrinsic and a priori information are removed, the forward backward algorithms like the BCJR and the SOVA tend to produce outputs that are locally highly correlated so that a direct feedback effect will still persist. The interleaver is introduced in the feedback loop to suppress these correlations so that the direct feedback effect is avoided. An increasing interleaver size will better

suppress these correlations and improve the BER performance.

The soft information for these iterative algorithms is expressed in log likelihood ratios. The log likelihood ratio of a bit c_k is expressed as

$$\Lambda(c_k) = \log \frac{p(c_k = 0)}{p(c_k = 1)} \tag{2.2.4}$$

The extrinsic information at the output of the first decoder is given by [7]

$$\Lambda_{1e}^{(i)}(c_k) = \underbrace{\Lambda_1^i(c_k)}_{aposteriori} - \underbrace{\frac{2}{\sigma^2} r_{k,0}}_{intrinsic} - \underbrace{\widetilde{\Lambda}_{2e}^{(i-1)}(c_k)}_{apriori} \tag{2.2.5}$$

where Λ denotes log likelihood ratios. The second term on the right hand side represents the intrinsic information and the third term represents a priori information - both of which have to be removed. Similarly, at the output of the second decoder we have

$$\Lambda_{2e}^{(i)}(c_k) = \Lambda_2^i(c_k) - \frac{2}{\sigma^2} \widetilde{r}_{k,0} - \widetilde{\Lambda}_{2e}^{(i)}(c_k) \tag{2.2.6}$$

The two MAP decoders operating in an iterative manner are shown in figure 2.4 where the notations A and E are used for a priori and extrinsic information respectively.

The BER improvements over the iterations of a turbo code with an interleaver depth of 420 in an additive white Gaussian noise channel are shown in figure 2.6. It can be seen that the improvement in the BER performance decreases with increasing number of iterations. After a certain number of iterations, the decoders stop making progress and the final decoded sequence can be obtained by taking hard decisions at the output of the second decoder. The BER curve of a turbo code consists of three regions. The region of low Eb/No[5] where there is almost no improvement over the iterations,

[5]Eb/No is the bit energy per noise power spectral density also called the signal to noise ratio per bit. For a coded system with code rate r, it is related to the SNR as Eb/No = SNR/r.

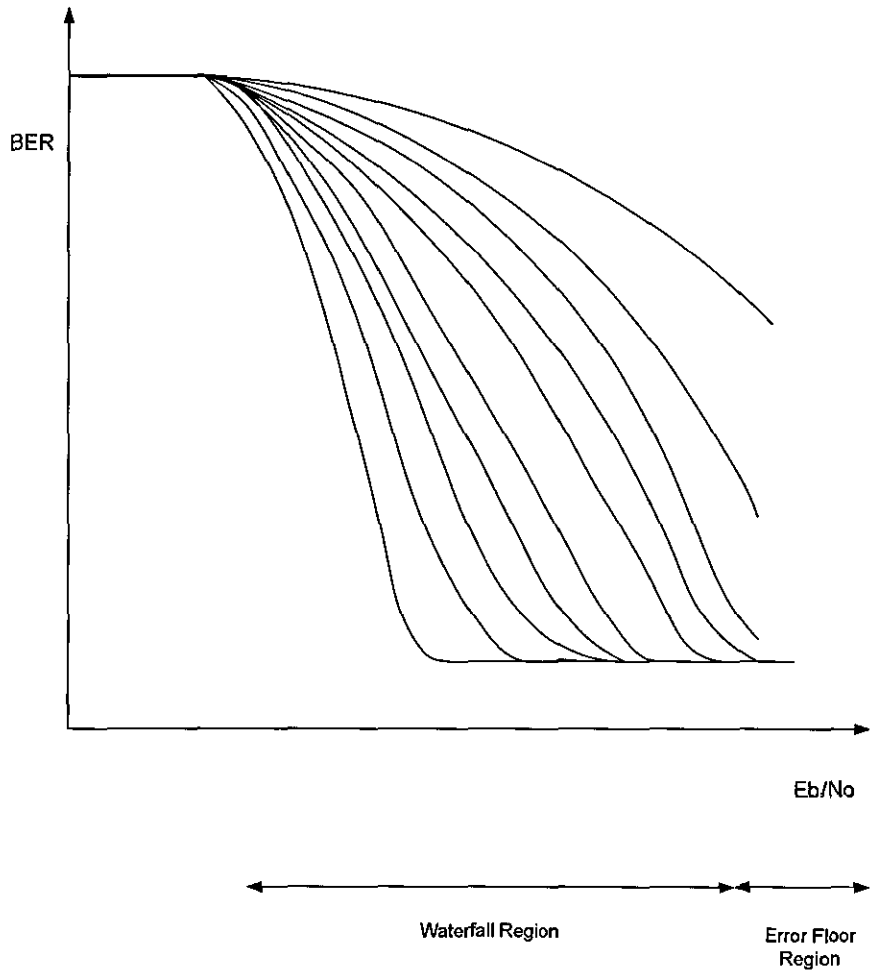

Figure 2.5. The regions of the BER curve of a turbo algorithm.

the 'waterfall' (also turbo cliff region) region where there is persistent BER reduction over many iterations and the error floor region where there a low BER is achieved over a few iterations. These regions are depicted in figure 2.5. Figure 2.6 shows the waterfall region of the code where there are substantial performance improvements over eight iterations.

EXIT Charts

Extrinsic information transfer charts (EXIT) were introduced by ten Brink in [8] as a useful graphical tool for the analysis and comparison of turbo codes and was later extended for turbo equalization in [9]. EXIT charts provide useful insight into the convergence behavior of the iterative decoding

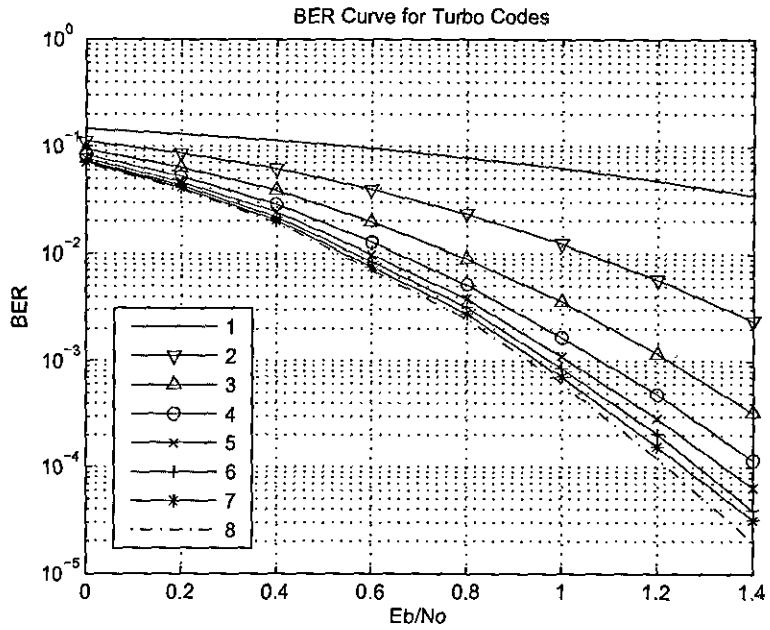

Figure 2.6. Convergence of turbo code waterfall region in the AWGN channel

scheme. They graphically depict the flow of extrinsic information through the decoding process based on mutual information. Ten Brink showed that the averaged decoding trajectory is bounded on both sides by the transfer characteristics of the two decoders [8]. Decoding is therefore possible until the transfer characteristics pinch off. In the low Eb/No region, the transfer characteristics are pinched off, hence no improvement is made over the iterations. In the waterfall region, there is a narrow bottleneck between the transfer characteristics that the decoding trajectory has to cross. The trajectory takes many small steps in crossing this bottleneck explaining why there is persistent BER improvement over many iterations in this region. In the error floor region, the transfer characteristics are wide open allowing fast convergence in just a couple of iterations. The transfer characteristics are obtained through stand alone operation of the decoder. The behavior of the turbo code can thus be predicted without actually having to simulate the entire decoder. For sufficient interleaver depth, the waterfall will lie at the

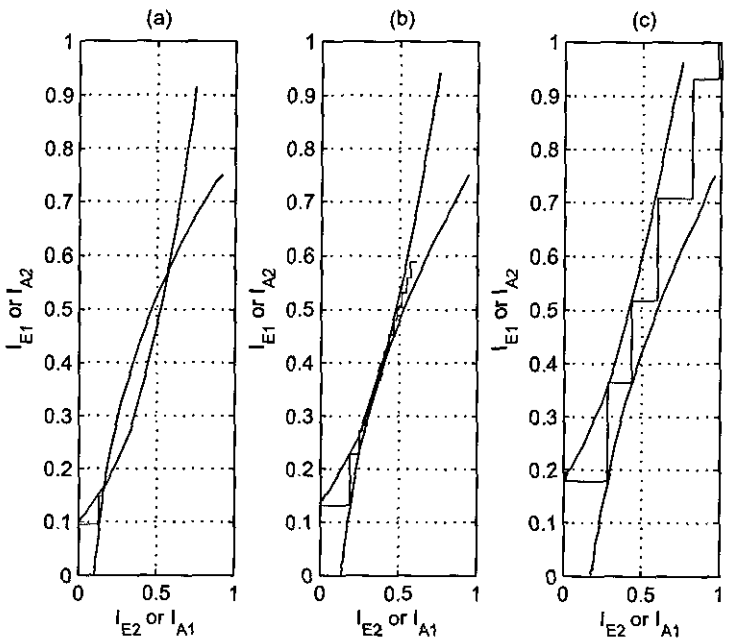

Figure 2.7. EXIT charts for a punctured rate $\frac{1}{2}$ code at (a) Eb/No = 0.4 dB (b) Eb/No = 0.7 dB (c) Eb/No = 1 dB

Eb/No at which the transfer characteristics pinch off.

Figure 2.7 shows the EXIT charts for a punctured rate $\frac{1}{2}$ turbo code at various values of Eb/No. Initially, the averaged decoding trajectory matches well with the transfer characteristics but increasing correlations of extrinsic information begin to cause mismatch as the iterations progress. At Eb/No = 0.7 dB, the decoding trajectory barely manages to sneak between the bottleneck formed by the transfer characteristics suggesting the location of the turbo cliff for this code.

2.3 Channel Characteristics

In order to understand the signal processing that is carried out in a communications system, it is necessary to understand the physical characteristics of the channel. In this section we explain how the transmit signal is distorted by the channel as it propagates through it. The signal experiences several different phenomenon. The channel distorts and spreads the signal

in different dimensions so that, many times, the signal is unrecognizable in its received form. Substantial signal processing is then required on the received signal before the transmit information may be extracted from it. These signal processing techniques are explained in the next section. While they do increase receiver complexity and are sometimes detrimental to communication, these channel effects may not always be harmful and can even be exploited for gains at times.

As the signal propagates through the environment, it experiences path loss. Another phenomenon induced by the wireless channel is fading whereby the received signal fluctuates. These fluctuations are observed in both the received signal amplitude and in its envelope. Fast fading, or short term fading refers to rapid fluctuations of the received signal caused by scattering off objects near a moving mobile. Under the assumption of a large number of wavefronts with random amplitudes and angles of arrival with uniformly distributed phases, the in-phase and quadrature components of the vertical electrical field are Gaussian distributed. The envelope of the received signal follows a Rayleigh density function [10]

$$p(x) = \begin{cases} \frac{x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}}, & x \geq 0; \\ 0, & x < 0. \end{cases} \tag{2.3.1}$$

where σ^2 is the variance. The fluctuations in the envelope are termed slow fading while those in the signal itself are called fast fading. The slow fading, or long term fading is caused by shadowing effects of buildings or landscape features and it is determined by the local mean of the received waveform. The slow fading is dependant on antenna heights, terrain features and operating frequency. The received power averaged over a Rayleigh fading distribution

follows a log normal distribution

$$p(x) = \begin{cases} \frac{1}{\sqrt{\pi}\sigma x} e^{\frac{-(\log x - \mu)^2}{2\sigma^2}}, & x \geq 0; \\ 0, & x < 0. \end{cases} \tag{2.3.2}$$

The signal received over a wireless channel consists of many time delayed and scaled replicas of the transmit signal arriving at the receiver[6]. These replicas are called multipaths and they arise from scattering, reflection, refraction and / or diffraction of the radiated energy off objects between the transmitter and receiver. If these multiple paths arrive nearly simultaneously, the effective channel seen by the receiver is a single path with a gain equal to the sum of the gains of these multiple paths and the channel is termed frequency flat. However, if the replicas are significantly delayed so that some replica corresponding to one transmission actually arrives nearly simultaneously with other replicas from preceding transmissions, the channel is termed as frequency selective. Both the forward and reverse links see the same multipaths. This is the reason behind the principle of reciprocity, i.e. the forward and reverse channels are the same in a time division duplex (TDD) system if the ping pong time[7] is much smaller than the channel coherence time. However, the path amplitudes and phases are a function of the frequency so the principle of reciprocity does not carry over to frequency division duplex (FDD) systems.

Frequency selectivity can be caused by reflections from multiple scatterers when the difference between the arrival times of the replicas formed by the scatterers is larger than the reciprocal symbol transmission rate. Frequency selective behavior is induced by remote scatterers, i.e. scatterers neither in the vicinity of the mobile nor the base but in between the two. Frequency selectivity can also be induced by limited bandwidth in the ab-

[6] A similar phenomenon to echoes in acoustics

[7] The ping pong time is the time between the transmission and reception of a frame in TDD systems.

sence of physical scatters. Limited bandwidth smears the transmit pulse in time[8] so that interference is introduced. Frequency selective channels provide delay diversity which is a valuable resource in combating channel fading but it also introduces intersymbol interference[9]. This behavior is also shown by wired channels and even data storage systems like magnetic disks where the head senses changes in flux and cannot pass DC. Frequency selective behavior is characterized by the delay spread or the coherence bandwidth. The delay spread is the span of path delays. The coherence bandwidth is the maximum frequency separation for which the frequency domain channel responses remain strongly correlated and is inversely proportional to the delay spread.

Doppler spread is the spreading in frequency of a pure tone. It is also caused by local to terminal scatterers in conjunction with mobility of the mobile terminal. It is also caused by oscillator drifts and carrier frequency offsets. This causes time selectivity so that such a channel varies rapidly with time. The Doppler power spectrum is defined as the Fourier transform of the autocorrelation function of the channel impulse response. The Doppler spread is the support of the Doppler spectrum. The time selective nature can be characterized by the Doppler spread or the coherence time. The coherence time represents the time separation over which the channel impulse response remains strongly correlated. Time selective channels can provide Doppler diversity to combat fading.

Space selective fading is caused by local to base scatterers and remote scatterers. These scatterers induce angle spread, i.e. they spread the angles of arrival of the multipaths at the receiver. At the transmitter, the angle spread refers to the spread of the departure angles of the multipaths. As a result, the channel fades in the spatial dimension. The space selective fading can be characterized by the coherence distance or the angle spread. The

[8] multiplication with a rect() in frequency appears as convolution with a sinc() in time

[9] Intersymbol interference or ISI is self interference that occurs when successive transmissions from a source interfere with each other.

coherence distance is the distance over which the channel impulse response remains strongly correlated. The coherence distance is inversely proportional to the angle spread. Space selective behavior allows exploitation of the spatial dimension for beamforming and spatial multiplexing because it causes variations of the channel from antenna to antenna.

The delay, Doppler and angle spread present in a cell depends on the physical environment. A list of typical values is shown in table 2.1 [1]. However, the selectivity seen in different dimensions depends on the system parameters in addition to the environment. Although, the underlying parameters are determined by the physical channel, their effects can be masked or highlighted by the system parameters. For example the global system for mobile communications (GSM) uses a bandwidth of 200kHz which is much larger than the Doppler spread typical of a hilly environment. Thus GSM does not suffer from time selectivity and does not require the channel to be tracked during a transmission. It uses a very short symbol period of $3.7\mu s$ while the delay spread in a hilly environment is $20\mu s$. Thus we can see that there will be substantial ISI with up to six paths and substantial effort must be expended to deal with it. Another standard, the interim standard 54 (IS-54) uses a symbol period of $41.6\mu s$ so there is negligible ISI even in a hilly environment. On the other hand, it uses a bandwidth of 30 kHz and a time slot of 6.66ms which is larger than the coherence time of the channel thus showing time selectivity. Thus while IS-54 does not require any equalization effort, it must track the channel variations during a transmission.

Similarly, the universal mobile telecommunications system (UMTS) uses wideband code division multiple access (CDMA) so that all users transmit over the entire bandwidth. The bandwidth is shared by all the users and distinct chip sequences or codes are used for multiple access. Because of the wide bandwidth employed for transmission, the users will typically experience frequency selective propagation effects which are dealt with using

Environment	Delay Spread	Angle Spread	Doppler Spread
Flat Rural (Macrocell)	0.5 μs	1°	190 Hz
Urban (Macrocell)	5 μs	20°	120 Hz
Hilly (Macrocell)	20 μs	30°	190 Hz
Mall (Microcell)	0.3 μs	120°	10 Hz
Indoors (Picocell)	0.1 μs	360°	5 Hz

Table 2.1. Typical delay, angle and Doppler spreads in cellular applications [1]

a RAKE receiver[10]. Also because of the wide bandwidth used for transmission, time selectivity induced by Doppler spread will be much smaller as compared to that induced in IS-54 and GSM. However, UMTS poses its own distinct set of challenges. The UMTS standard has to contend with multiple access interference (MAI) and has to maintain stringent power control over the users to deal with the near far effect[11]

In this work, we will be dealing with frequency selective and doubly selective channels. Frequency selective channels are dispersive in time while doubly selective channels are dispersive in both time and frequency. Thus doubly selective channels have both delay and Doppler spread. Although angle spread will not be explicitly discussed further, we will also be implicitly taking it into consideration in our study of multiple input multiple output (MIMO) systems. This is because the antenna arrays have to be designed so that the separation between antennas is more than the coherence distance so that the antennas see uncorrelated channels in order to achieve spatial diversity. Thus in exploiting the spatial dimension, the underlying assumption is that there is sufficient angle spread in the system.

Apart from all these channel induced effects, there is the ubiquitous thermal noise. This omnipresent noise is generally Gaussian distributed but not necessarily white. A simple noise whitening filter may be used at the receiver before any other processing. A matched filter is used to maximize the signal

[10]The RAKE consists of multiple correlators followed by combining(typically maximum ratio combining (MRC)).

[11]By the near far effect, the weak signal from a far user is swamped by the MAI from a strong signal of a nearby user.

to noise ratio and channel coding techniques may be used to average out the effects of noise.

2.4 Equalization

Channel equalization is required to mitigate the effects of fading and to remove interference. Equalization has been widely studied after the pioneering paper by Lucky [11]. Some of the techniques developed are introduced in this section.

Although channel equalization is generally performed at the receiver (post equalization), it may also be carried out at the transmitter (pre equalization) or balanced between the two [12]. Pre equalization is rarely applied as it requires CSI at the transmitter which is generally unavailable. However, with CSI at the transmitter, pre equalization can avoid the noise enhancement problem, particularly if the channel matrix is ill conditioned. Balanced equalization may or may not require CSI at the transmitter. An example of the former is vector coding where the singular value decomposition of a multiple input multiple output (MIMO) channel is used to open parallel data pipes between the transmitter and receiver [10]. [12] On the other hand, orthogonal frequency division multiplexing (OFDM) uses balanced equalization but CSI is not required at the transmitter. This is because the matrix is diagonalized by means of a channel independent transformation [13][13].

CSI can be obtained at the receiver with a known training signal from the transmitter and may be used to design an equalizer. Acquiring CSI at the transmitter is much more difficult and is usually achieved by feeding back from the receiver or by exploiting some invariance properties to estimate the forward channel from the reverse channel. Since the design of an equalizer

[12] In vector coding, the data streams are modulated with the right singular vectors at the transmitter while they are demodulated with the the left singular vectors at the receiver and this ensures that the streams are received without interference. Because different channels can have different singular vectors, CSI is required at the transmitter to find the right singular vectors.

[13] Circulant matrices have the property that any circulant matrix is diagonalized by the DFT matrix. The Fourier basis are eigenvectors of the circulant matrices.

requires a matrix inversion that is computationally complex, methods have also been developed for direct equalizers that circumvent the estimation of the channel. Much research has also focussed on the design of blind equalizers that can equalize the channel without the need for training or CSI at the receiver.

Equalization may be serial or block based. Each of these approaches have their own merits and drawbacks. Serial equalizers for example have smaller design and implementation complexities and they can also tradeoff complexity for performance. However, channel invertibility and symbol detection are not always guaranteed. Block equalizers can provide better performance and guarantee channel invertibility [12].

A length 5 frequency selective channel with impulse response

$$h[n] = 0.227\delta[n] + 0.46\delta[n-1] + 0.688\delta[n-2] + 0.46\delta[n-3] + 0.227\delta[n-4] \tag{2.4.1}$$

will be used in this section to illustrate the performance of the different equalization algorithms discussed. This channel causes severe ISI [14] and is difficult to equalize as its four zeros are very close to the unit circle and its frequency response is characterized by a deep null as shown in figure 2.8.

2.4.1 Zero Forcing Equalization

Zero forcing equalization is based on the minimization of the peak distortion criterion in order to eliminate intersymbol interference altogether [14] and perfectly recover the input signal in the absence of noise. Finite length baud spaced serial zero forcing equalization is not possible for a frequency selective channel and it takes an IIR filter to invert an FIR filter. If the channel transfer function is $\mathbf{H}(z)$, the transfer function of the serial zero forcing equalizer is

$$G_{ZF}(z) = \frac{1}{H(z)} \tag{2.4.2}$$

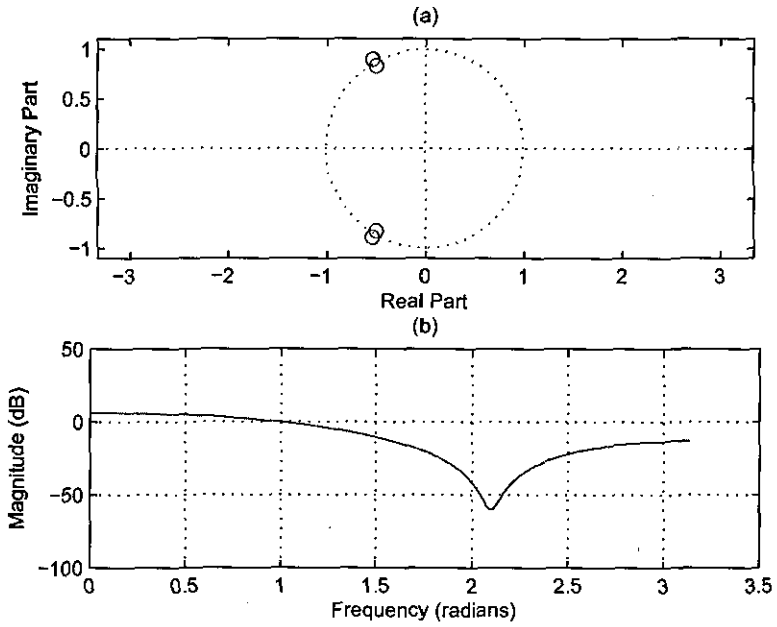

Figure 2.8. (a) Locations of the channel zeros (b) Magnitude response of the channel.

which is only causal and stable for minimum phase channels. Even theoretically, the IIR serial zero forcing equalizer can only be used to equalize minimum phase channels.

It has been shown that it is possible to invert a channel with a finite length equalizer if oversampling of the output is employed [15] in the absence of noise. This fractionally spaced equalization can be modelled by both a multirate system model or a multichannel system model. Consequently it is also possible to perfectly recover the input signal in the absence of noise if the oversampling is applied in either the spatial dimension (SIMO system) or the temporal dimension. Given a $T/2$ spaced system where T is the baud rate, L is the channel length, and L_{eq} is the equalizer length, the overall channel equalizer response is given by

$$\mathbf{f} = \mathbf{H}_{sl}\mathbf{g} = \begin{bmatrix} \mathbf{H}_o & \mathbf{H}_e \end{bmatrix} \begin{bmatrix} \mathbf{g}_e \\ \mathbf{g}_o \end{bmatrix} \tag{2.4.3}$$

where $\mathbf{H}_o$ ($\mathbf{H}_e$)is the $L+L_{eq}-1 \times L_{eq}$ baud spaced convolution matrix of the odd (even) samples of the fractionally sampled impulse response and $\mathbf{H}_{sl}$ is a Sylvester matrix constructed from two convolution matrices.

$$\mathbf{H}_o = \begin{bmatrix} h_0^{odd} & & & \\ h_1^{odd} & h_0^{odd} & & \\ \vdots & h_1^{odd} & & \\ h_{L-1}^{odd} & \vdots & \ddots & h_0^{odd} \\ & h_{L-1}^{odd} & & h_1^{odd} \\ & & & \vdots \\ & & & h_{L-1}^{odd} \end{bmatrix} \tag{2.4.4}$$

and $\mathbf{H}_e$ is defined similarly from the even samples of the fractionally sampled channel impulse response. For perfect source recovery with a delay δ, the combined channel and equalizer response vector must be

$$\mathbf{f}_\delta = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T \tag{2.4.5}$$

where the unity coefficient is in the δth position. The system of equations $\mathbf{f}_\delta = \mathbf{H}_{sl}\mathbf{g}$ must have a solution which requires that $\mathbf{H}_{sl}$ must be full row rank, i.e

$$L + L_{eq} - 1 \leq 2L_{eq} \Rightarrow L_{eq} \geq L - 1 \tag{2.4.6}$$

which is the minimum length of the equalizer required to obtain a zero forcing solution in $T/2$ spaced systems. This is known as the length condition. Notice that $\mathbf{H}_{sl}$ is a Sylvester matrix and by the properties of Sylvester matrices, $\mathbf{H}_{sl}$ will be rank deficient if the polynomials $\mathbf{h}_e$ and $\mathbf{h}_o$ share a common root. Hence the subchannels in a fractionally spaced system must not share a common root for the zero forcing solution to exist.

As opposed to serial equalization, block zero forcing equalization can be performed on a SISO baud spaced system. Assuming a block of N output

samples, the output samples may be expressed as

$$\mathbf{y}[n] = \sqrt{E_x}\mathbf{H}\mathbf{x}[n] + \mathbf{v}[n]$$

where

$$\mathbf{y}[n] = [y[n] \quad \dots \quad y[n+N-1]]^T$$

$$\mathbf{x}[n] = [x[n-L+1] \quad \dots \quad x[n+N-1]]^T$$

$$\mathbf{v}[n] = [v[n] \quad \dots \quad v[n+N-1]]^T$$

and $\mathbf{H}$ is a $(N+L-1) \times N$ convolution matrix. The zero forcing solution $\mathbf{G}_{zf}$ is any $N \times (N+L-1)$ matrix that satisfies

$$\mathbf{G}_{zf}\mathbf{H} = \mathbf{I}_N \tag{2.4.7}$$

The minimum norm block zero forcing solution is given by [10]

$$\mathbf{G}_{zf} = (\mathbf{H}^H\mathbf{R}_v^{-1}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{R}_v^{-1} \tag{2.4.8}$$

where $\mathbf{R}_v = E(\mathbf{v}[n]\mathbf{v}^H[n])$ is the autocorrelation matrix of the vector of noise samples. This solution reduces to the Moore Penrose matrix pseudo inverse for independent and identically distributed white noise [1]. Looking closely at Eq. (2.4.8), it is obvious that the zero forcing operation involves noise whitening, and a projection onto the signal subspace.

The zero forcing equalizer is also called the decorrelating detector. Its primary disadvantage is noise enhancement. Although it eliminates ISI, it also enhances noise thereby degrading performance at the receiver. If CSI is available at the transmitter, pre-equalization based on the zero forcing principle may be employed prior to transmission and the problem of noise enhancement will be avoided at the price of possibly having to transmit higher power. In maintaining a steady received signal to noise ratio, channel

inversion can require a high amount of power which might not be possible as many systems are peak power constrained. Even if inversion were truly possible, the caveat here is that channel inversion is strictly suboptimal from the information theoretic point of view as it is the exact opposite of the waterfilling power allocation strategy that maximizes the system capacity. Furthermore, channel knowledge at the transmitter is often hard to come by. Nevertheless channel inversion does find some applications in slow fading scenarios where waterfilling is not possible [13].

2.4.2 MMSE Equalization

Since noise is not taken into consideration in the design of zero forcing equalizers, they blow up noise while inverting the channel. The use of zero forcing equalization is limited to channels with mild interference. It has been proposed to design equalizers according to some criterion that balances interference reduction with noise enhancement. One possible criterion is to minimize the mean squared error between the transmitted signal and the detected signal. The equalizer transfer function is

$$G_{MMSE}(z) = \frac{H^*(z^{-1})}{H(z)H^*(z^{-1}) + N_0} \tag{2.4.9}$$

The block minimum mean squared error equalizer is given by [10]

$$\mathbf{G}_{MMSE} = (\mathbf{H}^H \mathbf{R}_v^{-1} \mathbf{H} + \mathbf{R}_x^{-1})^{-1} \mathbf{H}^H \mathbf{R}_v^{-1} \tag{2.4.10}$$

where $\mathbf{R}_{vv} = E(\mathbf{v}\mathbf{v}^H)$ is the noise correlation matrix and $\mathbf{R}_{xx} = E(\mathbf{x}\mathbf{x}^H)$ is the correlation matrix of the transmit signal and $\mathbf{H}$ is a $(N + L - 1) \times 1$ dimensional Toeplitz matrix of the channel taps. For the serial equalizer

$$\mathbf{g}_{mmse} = \arg\min_{g} \|x(n) - z(n - \delta)\|^2 \tag{2.4.11}$$

where the serial equalizer is applied to detect the transmitted signal as

$$z(n) = \sum_{k=0}^{L_{eq}-1} g^*_{mmse}(k)y(n-k) \tag{2.4.12}$$

Defining $\dot{\mathbf{H}}$ as an $(L + L_{eq} - 1) \times L_{eq}$ convolution matrix formed from the channel taps, the serial equalizer is [10]

$$\mathbf{g}_{mmse} = (\dot{\mathbf{H}}^H \dot{\mathbf{R}}_v^{-1} \dot{\mathbf{H}} + \dot{\mathbf{R}}_x^{-1})^{-1} \dot{\mathbf{H}}^H \dot{\mathbf{R}}_v^{-1} \mathbf{e}_\delta \tag{2.4.13}$$

where $\mathbf{e}_\delta$ is a column vector of zeros with a 1 in position δ, termed the equalization delay.

The equalization delay is an important parameter in the design of MMSE equalizers. The choice of the delay will determine the mean square error (MSE) and BER performance. The optimal equalization delay will depend on the channel response's center of mass. The earlier equalization delay will perform better for minimum phase channels and later equalization cursors will perform better for maximum phase channels[14]. The optimal equalization delay is given by the index of the minimum diagonal entry of the MSE matrix [10]

$$\delta = \arg\min_n [\dot{\mathbf{R}}_v^{-1} \dot{\mathbf{H}} (\dot{\mathbf{H}}^H \dot{\mathbf{R}}_v^{-1} \dot{\mathbf{H}} + \dot{\mathbf{R}}_x^{-1})^{-1} \dot{\mathbf{H}}^H \dot{\mathbf{R}}_v^{-1}]_{n,n} \tag{2.4.14}$$

For a block size N, the block equalizer design requires an inversion of a $N \times N$ matrix and thus has a design complexity[15] of $\mathcal{O}(N^3)$ while the equalization itself requires multiplication by an $N \times (N + L - 1)$ matrix so the implementation complexity is approximately $\mathcal{O}(N^2)$ flops. For the se-

[14] A minimum phase system is a system whose zeros are all located within the unit circle [16]. A maximum phase system has all its zeros located outside the unit circle. If the zeros are scattered on both sides of the unit circle, the system is termed as mixed phase.

[15] By exploiting the Toeplitz structure of the convolution matrix, this complexity can be reduced to $\mathcal{O}(N^2)$

rial equalizer, the design complexity is $\mathcal{O}(L_{eq}^3)$ flops because inversion of a $L_{eq} \times L_{eq}$ matrix is required. The implementation complexity is $\mathcal{O}(NL_{eq})$. For $L_{eq} \ll N$, the complexities are considerably less for serial equalization.

Fractionally spaced equalizers can also be designed according to the MMSE criterion. While baud spaced equalizers would seem computationally more feasible, the difference in complexity is not very pronounced [15]. In order to achieve the same performance, baud spaced equalizers have to be several times the length of the channel while, at least theoretically, fractionally spaced equalizers only need to satisfy the length and zero conditions.

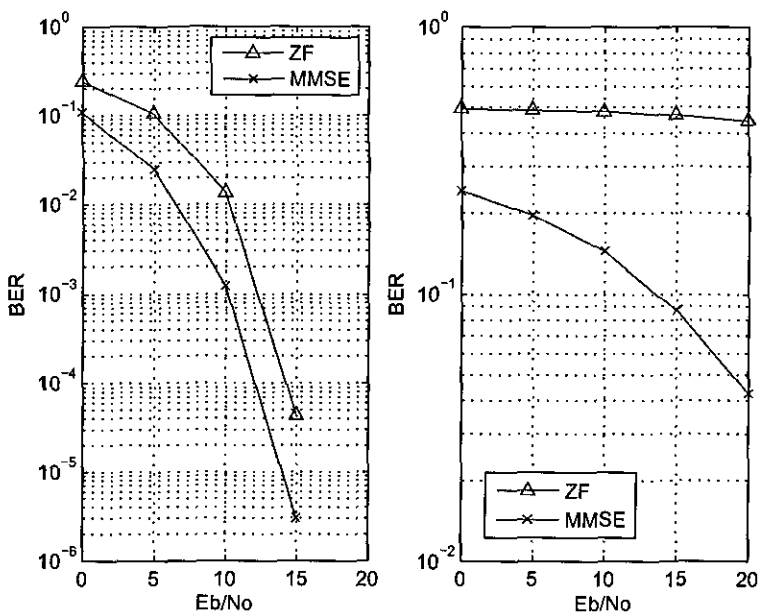

Figure 2.9. (a) Block equalizer (2.4.10) performance for channel (2.4.15) which has mild ISI. (b) Block equalizer (2.4.10) performances for channel (2.4.1) which has severe ISI.

Figure 2.9 compares the BER performance of the block zero forcing and MMSE equalizers. The comparison is made for both the channel with severe

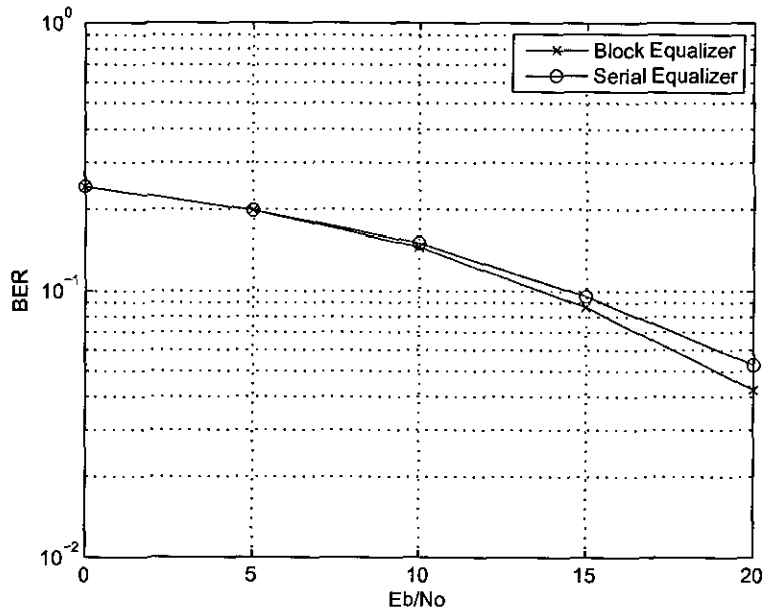

Figure 2.10. BER performance of block MMSE equalizer (2.4.10) and serial MMSE equalizer (2.4.13) for channel (2.4.1).

ISI in (2.4.1) and also for the channel with mild ISI[16]

$$h(n) = -0.418\delta(n) - 0.2365\delta(n-1) - 0.1025\delta(n-2) + 0.85\delta(n-3) - 0.1864\delta(n-4) \qquad (2.4.15)$$

Figure 2.10 compares block MMSE equalization with serial MMSE equalization. Figure 2.11 shows the magnitude response of the channel, the MMSE serial equalizer at 10 dB and the combined channel equalizer magnitude response. It can be seen from figure 2.11 that because of the deep null in the channel frequency response and the noise, the equalizer does not attempt to invert the channel.

[16]Channel (2.4.1) has severe ISI and is very difficult to equalize because of its zeros which are located very close to the unit circle. These zeros induce highly frequency selective behavior as the unit circle represents the axis of Fourier frequencies and the zeros cause large attenuation at frequencies close to their location. The zeros of channel (2.4.15) are further removed from the unit circle so the frequency response shows less variation.

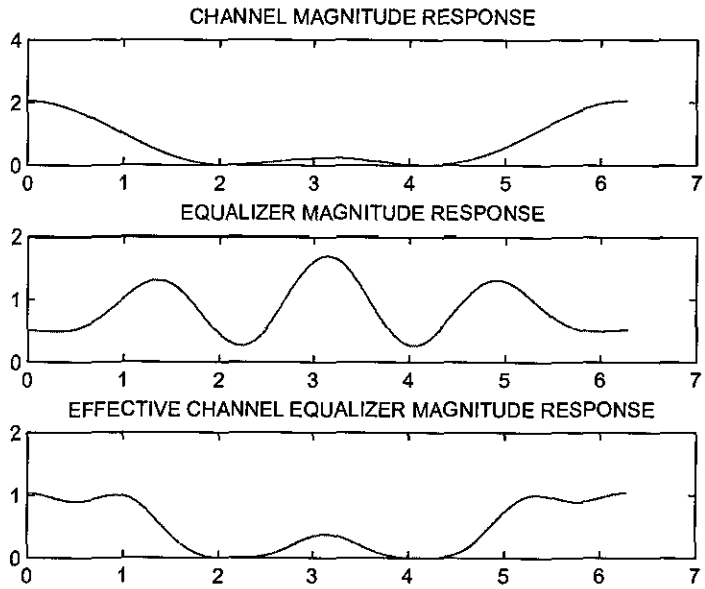

Figure 2.11. Frequency Responses of the channel (2.4.1), MMSE equalizer at 10dB and the response of the effective equalized channel on a linear scale.

2.4.3 ML Sequence Estimation

The equalizers discussed thus far are linear. Linearity is a desirable property which leads to simple implementations in systems. Linear equalizers are, however, outperformed by non linear detection techniques. The optimal detection strategy employs Bayesian estimation and chooses the sequence with maximum aposteriori probability.

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) \tag{2.4.16}$$

If all sequences are equally probable, maximization of the aposteriori probability is equivalent to maximization of the likelihoods. Thus maximum likelihood (ML) sequence estimation is the optimal detection strategy for equiprobable transmitted sequences. The maximum likelihood estimate of $\mathbf{x}$

is chosen to be the vector that maximizes the likelihood function

$$p(\mathbf{y}|\mathbf{x}) = p(y(1), y(2), \ldots, y(N)|\mathbf{x}) \tag{2.4.17}$$

Since the noise encountered in communication systems is Gaussian and because of the concavity of the natural logarithm, it is easier to work with the log likelihood function. The ML detector thus tries to find the sequence that minimizes

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left((\mathbf{y} - \mathbf{H}\mathbf{x})^H \mathbf{R}_v^{-1} (\mathbf{y} - \mathbf{H}\mathbf{x})\right) \tag{2.4.18}$$

The complexity of this equalization grows exponentially in the length of the channel to be equalized and the size of the constellation. A brute force approach would involve computing the metric in (2.4.18) for all possible sequences and choosing the sequence with the least metric. Such an approach would also increase complexity exponentially with the sequence length and be infeasible for all but very short sequences. Viterbi [17] reduced the complexity of ML detection to be linear in the length of the data sequence with his algorithm. However, the complexity remains exponential in the modulation order and the length of the channel to be equalized so that computational complexity of ML detection can often be very large even with the Viterbi algorithm.

In practice the algorithm is too complex to use for detection with anything but very short channels. Several authors have looked into channel shortening to reduce the length of the channel and then apply ML detection [18, 19].

2.4.4 Decision Feedback Equalization

The decision feedback equalizer is a nonlinear equalizer that seeks to combine the simple implementation of linear equalizers while avoiding the noise enhancement caused by them. It consists of a feedforward filter g and a feedback filter d as shown in figure 2.12. Previous detected symbols are fed back

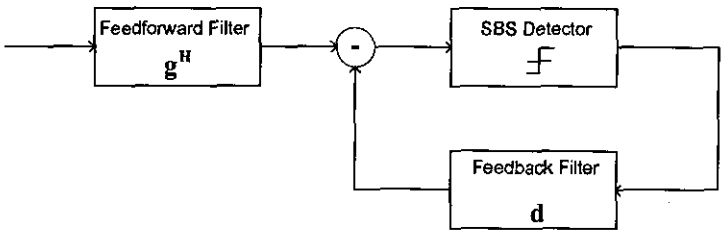

Figure 2.12. Decision feedback equalizer.

through the feedback filter to help cancel residual ISI. Under the assumption of correct decision feedback, we have

$$z_n = \underbrace{c_\delta x_{n-\delta}}_{\text{information bearing cursor}} + \underbrace{\sum_{i=0}^{\delta-1} c_i x_{n-i}}_{\text{residual precursor ISI}}$$
$$+ \underbrace{\sum_{i=1}^{N_d} (c_{i+\delta} - d_i) x_{n-\delta-i}}_{\text{modeled postcursor ISI}} + \underbrace{\sum_{i=\delta+N_d+1}^{N_c} c_i x_{n-i}}_{\text{residual postcursor ISI}} + \underbrace{\sum_{i=0}^{N_f} g_i^* v_{n-i}}_{\text{filtered noise}} \quad (2.4.19)$$

where $c = g^* * h$ is the combined channel and feedforward filter response.

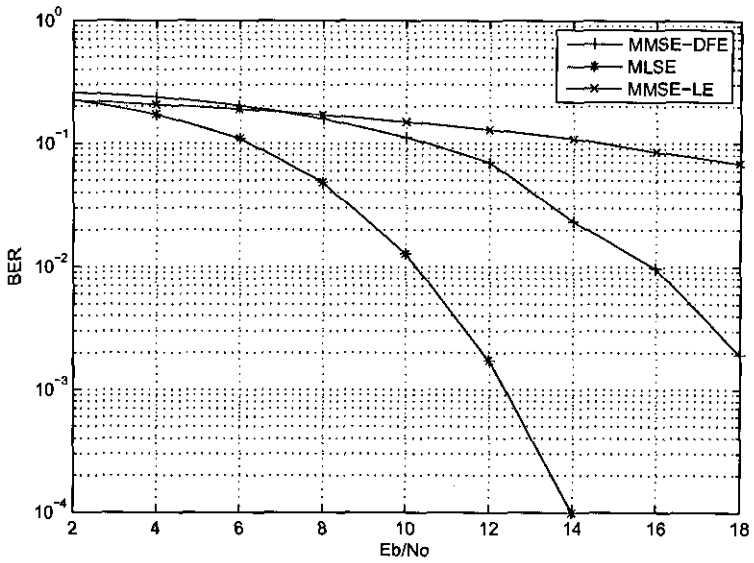

Figure 2.13. BER performance comparison of an MMSE linear equalizer with an MMSE decision feedback and MLSE equalizer for the channel (2.4.1)

The objective of the feedforward filter is to minimize the residual pre and postcursor ISI with a strong information bearing cursor while keeping the noise gain $\|\mathbf{g}\|^2$ small. The DFE uses the feedback filter to cancel out the modeled postcursor ISI by matching the feedback taps to the channel equalizer equivalent response $d_i = c_i, 1 \leq i \leq N_d$. The role of the feedback filter is thus to cancel out some interference without introducing noise gain. Although decision feedback equalization reduces noise enhancement, it suffers from error propagation on account of incorrect estimates being fed back[17]. It has been shown that the DFE is a canonical[18] transmission system in that it can achieve capacity on the equivalent ISI free channel when combined with sufficient coding [20–22]. A comparison of the performance of linear MMSE, decision feedback MMSE and MLSE is provided in figure 2.13 where it can be seen that the performance of the decision feedback MMSE equalizer lies in between the linear MMSE equalizer and the MLSE.

2.4.5 Turbo Equalization

When forward error correction (FEC) is applied to a frequency selective channel for data protection, the optimal detector will compare the received sequence with all possible transmit sequences and thus find the ML transmit sequence [9]. This optimal detector will jointly equalize and decode. Such a detector is too complex to implement and the equalization and decoding must be performed separately. The obvious way of doing so involves taking hard decisions at the output of the equalizer for the decoder to operate on. Hard decisions at the output of the equalizer destroy the soft information in the signal which could have been used to improve the performance of the receiver.

Since an optimal joint equalization and decoding algorithm is infeasi-

[17]This error propagation can be avoided by shifting the feedback section of the DFE to the transmitter where errors do not occur. The increase in transmit power can be limited through the use of the Tomlinson Harashima precoder [20].

[18]A canonical system is one which can, at least theoretically achieve the maximum possible transmission rate offered by the physical channel.

ble, the problem has to be split into two subproblems. Turbo equalization achieves this by using separate equalization and decoding blocks that exchange soft information. It is an iterative equalization and decoding scheme that was proposed in [23]. Some complexity reduction schemes for turbo equalization were introduced in [9] while [24] later provided a very good overview. The encoder and the channel are separated by the channel interleaver. The operation of the turbo equalizer is similar to the decoding of serial concatenated convolutional codes where the inner code is replaced by the channel. In fact the ISI channel acts as the inner rate-1 convolutional code, hence the decoding algorithm for serial concatenated convolutional codes is directly applied to channel equalization as 'turbo equalization'. The iterative equalizer and the decoder exchange soft information according to the turbo principle. Apart from decoding and equalization, the turbo principle has also found application in iterative demapping [25] and iterative multiuser detection [26, 27]. Turbo equalization combined with turbo codes has been proposed in [28].

The turbo equalizer was originally envisioned as a trellis based symbol detector like the MAP or MLSE operating jointly with a MAP decoder. Linear equalization can also be used as a simpler alternative. For turbo equalization, the transmitter has an encoder followed by an interleaver. The interleaved encoded sequence is transmitted over the channel and is iteratively processed for equalization and decoding. Linear processing of the output $\mathbf{y}_k$ to obtain an estimate of the input $\hat{x}_k$, we have

$$\hat{x}_k = \mathbf{g}_k^T \mathbf{y}_k + b_k \tag{2.4.20}$$

where $\mathbf{g}_k$ is the linear equalizer chosen to minimize the mean square error $E(|x_k - \hat{x}_k|^2)$ between a symbol and its estimate. This can be achieved by computing the estimate as [9]

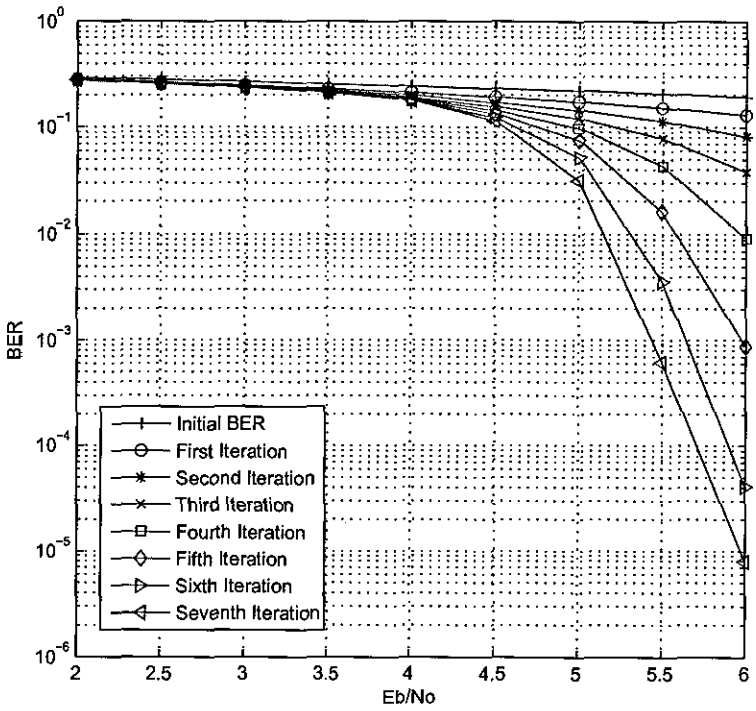

Figure 2.14. BER performance of turbo equalization over the iterations.

$$\hat{x}_k = E(x_k) + \mathbf{g}_k^T(\mathbf{y}_k - E(\mathbf{y}_k)) \tag{2.4.21}$$

where

$$\mathbf{g}_k = \mathrm{Cov}(\mathbf{y}_k, \mathbf{y}_k)^{-1}\mathrm{Cov}(\mathbf{y}_k, x_k) \tag{2.4.22}$$

and

$$\mathrm{Cov}(\mathbf{y}_k, \mathbf{y}_k) = \sigma^2\mathbf{I} + \tilde{\mathbf{H}}\mathrm{Cov}(\mathbf{x}_k, \mathbf{x}_k)\tilde{\mathbf{H}}^T \tag{2.4.23}$$

$$\mathrm{Cov}(\mathbf{y}_k, x_k) = \tilde{\mathbf{H}}\mathrm{Cov}(\mathbf{x}_k, x_k) \tag{2.4.24}$$

$$E(\mathbf{y}_k) = \tilde{\mathbf{H}}E(\mathbf{x}_k) \tag{2.4.25}$$

where $\tilde{\mathbf{H}}$ is a Toeplitz matrix constructed according to the equalizer length. Figure 2.14 shows the performance of the turbo equalization algorithm over the iterations for the channel in (2.4.1).

EXIT charts are an invaluable tool for design and analysis of turbo equalizers. Fig. 2.15 shows the EXIT charts for turbo equalization of the channel (2.4.1) at various bit energy to noise power spectral densities where it can be observed that the gap between the equalizer and decoder transfer characteristics increases with Eb/No so that convergence is achieved with fewer iterations.

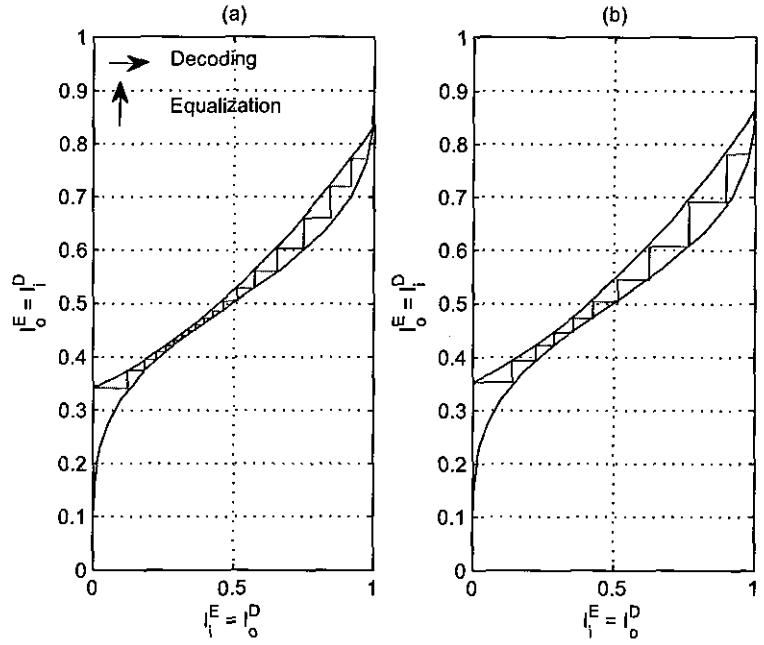

Figure 2.15. EXIT chart for turbo equalization for channel (2.4.1) at (a) Eb/No = 4.5 dB (b) Eb/No = 5 dB.

2.4.6 Blind Equalization

The design of an equalizer requires a matrix inversion which is computationally intensive. It also require CSI at the receiver which implies training that might be impractical in some situations. Much work has been done on self recovering [19] adaptive equalizers. Blind equalizers are generally based on property restoral algorithms. They try to equalize the channel by restoring some property of the input signal that has been degraded by the channel.

[19] also known as blind or unsupervised

These algorithms can exploit some structure in the transmitted signal like possibly a constant modulus or a finite alphabet. A mean cost function based on such a structure or property is defined. Stochastic gradient descent minimization of that cost is then applied to equalize the channel [1,29].

Constant Modulus Algorithm

The constant modulus algorithm (CMA) exploits a constant modulus property in the transmitted signal for blind equalization. The CM cost function is

$$J_{CM} = E\{(\gamma - |z_n|^2)^2\} \tag{2.4.26}$$

where $\gamma = E(|x|^4)/\sigma_x^2$ is the dispersion constant of the transmit constellation. For constant modulus signals like QPSK, γ is one. It has been shown that in a noiseless environment, CM minima match exactly with the MSE cost minima [30] in the combined channel equalizer space. With noise, the minima for both criteria move towards the origin. The dislocation is more pronounced in the CM cost. The CM cost surface is deformed by violations in the ideal zero cost conditions like noise, under modeling of channel length, non CM source and source correlation.

Let the received samples at the input to the CM equalizer at time n be stacked in a vector $\mathbf{y}_n$. Then the output sample of the CM equalizer would be $z_n = \mathbf{w}_n^H \mathbf{y}_n$. The CM update rule is obtained by substituting the gradient of the CM cost into the gradient descent algorithm. This gives [30]

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu(1 - |z_n|^2)\mathbf{y}_n z_n^* \tag{2.4.27}$$

where μ is the step size. Convergence of the constant modulus algorithm typically depends on the initialization. Convergence may be very slow along the smaller modes, i.e. the smaller eigenvalues of the autocorrelation matrix of the received signal $\mathbf{R}_{yy} = E(\mathbf{y}\mathbf{y}^H)$. The prewhitened constant modulus (PW-CM) algorithm has been introduced to improve the convergence behav-

ior of the algorithm. The convergence of constant modulus equalizer taps is shown in figure 2.16.

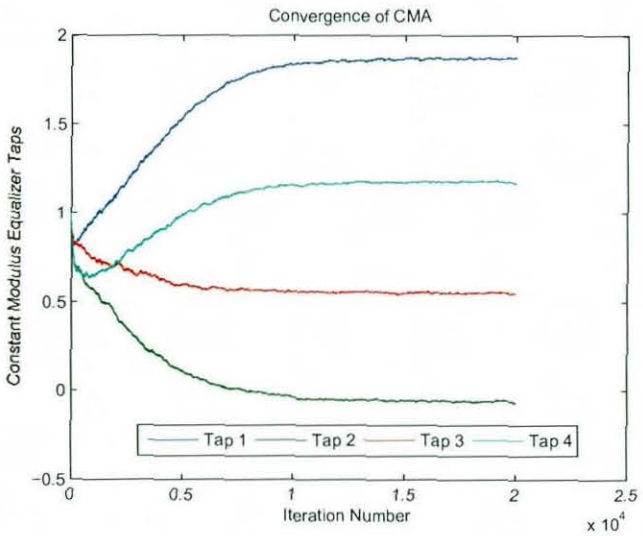

Figure 2.16. Convergence behavior of the constant modulus algorithm

Shalvi Weinstein Equalization

It has been shown in [31] that a sufficient condition for equalization is that the probability distribution of the equalized signal should be identical to the probability distribution of the input signal. [31] also showed that second order statistics are insufficient for phase identification. This leads to the very interesting observation that the problem of blind equalization cannot be solved when the input is Gaussian distributed.

Shalvi and Weinstein [32] make use of the fact that input signals in digital communications have discrete distributions to deduce sufficient and necessary conditions for equalization. They show that it is sufficient to maximize the kurtosis subject to the constraint that the variance (and hence power) is one. They propose a scalar cost that is the absolute value of the kurtosis and apply stochastic gradient descent on it, with a normalization of the coefficients in each iteration. They also show analytically that this

cost has no local minima. The input is pre-whitened and the vector update equations are

$$g_l' = g_l + \mu \text{sgn}(K(x))|z_i|^2 z_i y_{i-l}^* \tag{2.4.28}$$

$$g_{l+1} = (1/\sqrt{\sum |g_l'|^2}_l) g_l' \tag{2.4.29}$$

where z_n is the sequence at the output of the equalizer, μ is the step size of the gradient descent algorithm and sgn() is the signum function. Fig. 2.17(a) shows the combined channel equalizer response for Shalvi Weinstein equalization of (2.4.1) with a length fifteen equalizer. Fig. 2.17(b) shows the ISI at the output of the equalizer as a function of the iterations while fig. 2.17(c) depicts the convergence of the equalizer taps. This work was

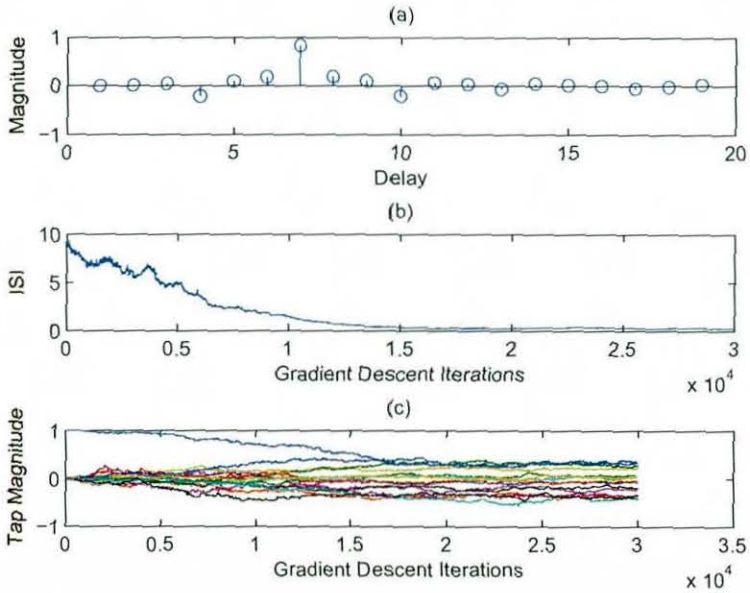

Figure 2.17. (a) Effective channel equalizer impulse response (b) Intersymbol interference over iterations (c) Convergence of equalizer taps.

later extended by Papadias in [33] for the blind source separation with their multiuser kurtosis maximization criterion (MUK) for separation of non convolutive mixtures which can find applications in communications to MIMO

flat fading channels.

2.4.7 Multicarrier Communications

The channel equalizers discussed so far entail varying amount of complexity. With a slight loss of throughput caused by the use of a cyclic prefix, the complexity of the equalization process may be reduced to approximately that of an FFT/IFFT pair by using multicarrier communications such as orthogonal frequency division multiplexing (OFDM) and discrete multitone (DMT) [10]. The cyclic prefix is needed to create the illusion of a circular convolution. This allows the broadband channel to be divided into many narrow band sub channels that are free of interference. Just as the frequency selective nature of the channel is a cause for receiver complexity, it is source of loss in throughput for multicarrier communications. Apart from the small complexity required by OFDM, it is very robust to impulsive noise [34]. Its major shortcoming is the high peak to average power ratio and sensitivity to time selective fading, carrier offsets, oscillator drifts, mobility etc. An alternative to multicarrier communications is single carrier frequency domain equalization [20] [35]. Any time variations in the channel cause the subcarriers to lose their orthogonality resulting in intercarrier and intersymbol interference. Multicarrier communications is generalized by discrete time channel partitioning. Discrete time channel partitioning including DMT and vector coding are canonical transmission systems.

2.5 Conclusion

The focus of this work is on the problem of channel estimation for communications. In order to gain a meaningful insight into the channel estimation problem, it is necessary to take a step back and understand both the physical channel and the communications systems. This chapter has discussed the dispersive effects of the communications channels that make channel

[20] also known as single carrier cyclic prefix

estimation necessary. These dispersive effects and their underlying causes have been explained and signal processing techniques to deal with or exploit these phenomenon have been introduced. Forward error correction (FEC) has been explained with an emphasis on convolutional and turbo codes as a means to combat AWGN and exploit the diversity effects of the channel. Equalization has been introduced as a means to combat the time dispersive nature of the channel and various equalization algorithms explained and their complexity versus performance tradeoffs discussed. The actual channel estimation problem is considered in the next three chapters which study channel estimation for multicarrier communications, and doubly selective channel estimation for single carrier communications.

Chapter 3

SUPERIMPOSED TRAINING FOR OFDM

3.1 Introduction

Multicarrier communications has found widespread acceptance and application in a very short span of time. It has been standardized for ADSL in ITU G.992.1 in the form of DMT. For wireless communications, it has been standardized in IEEE standards 802.11, 802.16 and 802.20 for wireless LAN (WiFi), Broadband Wireless Access (WiMax)[1] and Mobile Broadband Wireless Access (MBWA) [36–38] respectively in the form of coded OFDM (COFDM). This popularity of multicarrier communications can be attributed to its advantages including simple implementation and canonical performance[2].

The focus of this chapter is on superimposed training for channel estimation in OFDM which is typically used in wireless and broadcast applications as opposed to DMT which is employed for wireline communications. The contribution of this chapter lies in the use of semiblind techniques for the channel estimation problem. Single input multiple output (SIMO) and multiple input multiple output (MIMO) OFDM systems are considered in this chapter.

[1] Orthogonal frequency divsion multiple access is a competing technology for broadband cellular wireless access

[2] A communications system is said to be canonical if it can, atleast theoretically, achieve the maximum possible data rate offered by the physical channel.

As opposed to DMT, OFDM does not adapt the transmissions to the channel individually for the subcarriers[3]. The transmission of the cyclic prefix in the guard period results in a further loss of throughput which can be substantial for short to moderate size OFDM symbols, especially for very long channels. Several solutions to this problem have been proposed in literature including the use of filtered multitone (FMT) or the use of channel shortening equalizers for partial equalization of the channel impulse response so that most of the energy in the effective response spans a shorter duration than the original channel response and thus a shorter cyclic prefix may be used.

In addition to ensuring no inter block interference, the cyclic prefix also tricks the channel into believing the input is periodic. The cyclic prefix should span the channel memory and because only $N+L$ input symbols are processed at a time, where N is the number of symbols in a block and L is the channel memory, the convolution appears periodic. The IBI corrupted samples are discarded and receiver processing is performed in the frequency domain.

Channel estimation in multicarrier communications is generally performed by reserving tones for training purposes. With a fixed transmitter in OFDM, this results in a further loss of throughput over and above that due to cyclic prefix transmission. Superimposed training allows estimation of the channel at the expense of power instead of throughput. It is for this reason that we study superimposed training for OFDM.

3.2 Channel Shortening

Various literature has focussed on the design of channel shortening equalizers. One of the first channel shortening equalizers proposed was based on the maximal shortening signal to noise ratio (MSSNR) criterion proposed

[3]The OFDM transmitter does adapt to the overall channel quality by changing the modulation order so that a higher throughput is realized for better channels but there is no adaptation from subcarrier to subcarrier.

by Melsa in [39] and which has subsequently been shown to be a zero forcing design. MMSE channel shorteners were proposed in [40] and bit rate maximizing channel shorteners were proposed by [41,42]. It was later shown by [43] that the bit rate maximizing channel shorteners are in fact target impulse response optimized MMSE channel shorteners. Time domain channel shortening equalizers based on SINR maximization were studied in [44]. The problem of MIMO channel shortening was studied by [45]. A common feature of these works was that the channel shortener was implemented in the time domain, i.e. in direct cascade with the channel. [46] proposed per tone equalizers that were implemented at the output of the FFT block in the receiver with a different shortener for each tone. These per tone equalizers have similar run time complexities as time domain channel shorteners but their design complexity is considerably more if different equalizers are optimally designed for each tone. This complexity may be somewhat reduced by grouping similar tones together and designing a single equalizer for each group of tones. On the upside, a search of all delays is not required as per tone equalizer performance is relatively insensitive to equalization delay. Another advantage that is very pronounced in wireline communications is that while time domain channel shortening expends equal effort in equalizing all tones, per tone equalization offers the flexibility to focus equalization effort only on the desired tones while ignoring unused tones. The concept of per tone equalization was extended to MIMO-OFDM in [47].

3.3 Channel Identification

All this literature assumes perfect channel state information at the receiver. In the absence of CSI an adaptive filter for direct blind equalization may be employed. Such adaptive equalizers are based on a gradient descent of cost functions determined from the cyclic prefix or the autocorrelation [34]. The multicarrier equalization for restoration of redundancy (MERRY) al-

gorithm adapts the equalizer to minimize a cost function derived from the cyclic prefix [48, 49]. It tries to restore the cyclic nature that exists in the transmit signal by restoring a cyclic prefix of length one. It has been shown to be globally convergent and similar to the maximal shortening signal to noise ratio (MSSNR) criterion proposed by [39]. The sum squared autocorrelation minimization algorithm (SAM) adapts the equalizer to minimize the sum of square of the autocorrelation taps of the processed signal outside the shortened length [50]. Both algorithms do not require any training but are computationally complex. The MERRY is very slow to converge and it has large memory requirements as it updates the adaptive equalizer once per OFDM symbol. The SAM is much faster to converge but it suffers from undesirable minima. The algorithm converges to any of these minima depending on the initialization. It has been shown by the authors that these minima are related[4] to one another and an exhaustive search of these minima will lead to the optimal solution. Nevertheless, considerable complexity is introduced by an exhaustive search especially for longer channels which have more numerous undesirable minima. The upside of these algorithms is that there is no loss in throughput due to training.

As an alternative to these blind algorithms, training based algorithms have also been proposed for channel identification in OFDM. The channel is identified at the receiver with the help of pilot tones set aside for training [51]. These training based algorithms are computationally simple and practically feasible but there is an associated loss in throughput. Recently a superimposed training scheme has been proposed for single carrier communications which has the the desirable properties of both multiplexed training and blind techniques. Like blind schemes, no explicit time slots are allocated for train-

[4]The autocorrelation of a sequence is unchanged if any of the roots are reflected over the unit circle. Because the SAM cost is based on the autocorrelation function, it has minima at all $2^{L_{eq}}$ sequences that can be formed from the optimal equalizing solution by reflecting roots about the unit circle. Only two of these minima actually shorten the channel, but convergence depends on the initialization. Once convergence is achieved, an exhaustive search may be required to obtain the shortening solution.

ing and like trained systems the estimation is computationally simple and elegant. This superimposed scheme estimates the channel at the expense of power rather than bandwidth. It is thus suitable for the bandlimited regime and also for fixed transmitter schemes like OFDM. Superimposed training has previously been applied for channel shortening in multicarrier communications by [52]. However, we consider semiblind estimation for single input multiple output (SIMO) and multiple input multiple output (MIMO) systems where the detected data is reused for channel estimation in an iterative channel estimation and data detection process.

3.4 SIMO OFDM

We consider a baseband equivalent SIMO OFDM system where transmit and receive filtering are absorbed into the baseband channel coefficients. The multiple outputs may be due to fractional sampling in time or in space. The fundamental problem considered in this work is the problem of communication without CSI at the receiver. We study estimation of the channel and detection of the transmit signal at the receiver. We consider a general scenario where the channel memory exceeds the cyclic prefix length. We do not consider peak to average ratio reduction which has otherwise been extensively studied in literature and techniques like tone reservation and tone injection have been proposed. Further, we will consider perfect synchronization between the transmitter and the receiver. Previous research has shown that with a lack of synchronization between the transmitter and receiver, the channel estimate obtained is a cyclically rotated version of the actual channel. This rotated channel estimate cannot be used itself to design the receiver but a simplistic brute force approach is to check all rotated versions of the channel estimate. Other more feasible algorithms for synchronization exist in the literature and are beyond the scope of this work [53].

3.4.1 Superimposed Training

Superimposed training exploits the fact that the pilot added at low power onto the unknown data passes through the same channel as the unknown data. The superimposed pilot must be rich in modes for channel sounding, i.e. a persistent excitation signal . Because it shares the time and frequency dimensions with the unknown data, we must exploit some statistical properties to separate the training from the data at the receiver. The transmit signal during the k^{th} OFDM block,may be written as

$$\tilde{\mathbf{x}}_k = \mathbf{c} + \underbrace{\mathbf{T}_{cp}\mathbf{F}^H\mathbf{s}_k}_{\mathbf{x}_k} \tag{3.4.1}$$

where $\mathbf{c} \in \mathbf{C}^{(N+L_{cp})\times 1}$ is the training signal, $\mathbf{F}$ is the $N \times N$ DFT matrix whose $(m,n)^{th}$ element is given by

$$[\mathbf{F}]_{m,n} = \frac{1}{\sqrt{N}} e^{\frac{-j2\pi mn}{N}} \tag{3.4.2}$$

and $\mathbf{T}_{cp} \in \mathbf{R}^{(N+L_{cp})\times N}$ is the cyclic prefix insertion matrix

$$\mathbf{T}_{cp} = \begin{bmatrix} \mathbf{0}_{L_{cp}\times(N-L_{cp})} & \mathbf{I}_{L_{cp}} \\ & \mathbf{I}_N \end{bmatrix} \tag{3.4.3}$$

Although equation (3.4.1) gives the impression of a single carrier signal $\mathbf{c}$ being added onto a multicarrier signal, we note that $\mathbf{c}$ is periodic, hence cyclic. Thus we may consider the training to be multicarrier, $\mathbf{t}$, where $\mathbf{c} = \mathbf{T}_{cp}\mathbf{F}^H\mathbf{t}$ is the time domain training. As the channel is unknown at the transmitter, the optimal transmission strategy is for the transmit signal to be independent and identically distributed, i.e. $E(\mathbf{s}_k\mathbf{s}_k^H) = \mathbf{I}$. We note that a periodic sequence with period P is used for training, so the pilot is

$$\mathbf{c} = \mathbf{1}_R \otimes \mathbf{u} \tag{3.4.4}$$

where we assume there are R periods of the pilot in one block so that $R \times P = N + L$ and $\mathbf{1}_R$ is a $R \times 1$ vector of ones. Also, we let $\mathbf{u} \in \mathbf{C}^{P \times 1}$ be one period of the training. A periodic training signal is chosen as this simplifies the channel estimation. We will shortly see that with periodic training, an estimate can be formed from a sample averaging operation at the receiver. The product ($\otimes$) is the Kronecker product. The Kronecker product between a $m \times n$ matrix $\mathbf{A}$ and a $p \times q$ matrix $\mathbf{B}$ is defined as the $mp \times nq$ dimensional matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \tag{3.4.5}$$

We may define the training to information power ratio as

$$TIR = \frac{\rho}{1-\rho} = \frac{\|\mathbf{c}\|_2^2}{E(\|\mathbf{x}_k\|_2^2)} \tag{3.4.6}$$

The k^{th} signal block at the ν^{th} receive antenna $\mathbf{y}_k^\nu \in \mathbf{C}^{(N+L_{cp})\times 1}$ can be written as

$$\mathbf{y}_k^\nu = \tilde{\mathbf{X}}_k \mathbf{h}_\nu + \tilde{\mathbf{X}}_{k-1}^{ibi} \mathbf{h}_\nu + \mathbf{v}_k^\nu \tag{3.4.7}$$

where $\mathbf{h}_\nu = \begin{bmatrix} h_\nu[0] & \cdots & h_\nu[L] \end{bmatrix}^T$, $\mathbf{y}_k^\nu = \begin{bmatrix} y_k^\nu[0] & \cdots & y_k^\nu[N+L_{cp}-1] \end{bmatrix}^T$, $\mathbf{v}_k^\nu \in \mathbf{C}^{(N+L_{cp})\times 1}$ contains the additive white Gaussian noise samples and the matrix $\tilde{\mathbf{X}}_k$ is a Toeplitz matrix constructed from the transmit signal $\tilde{\mathbf{x}}_k$.

For example, for $k = 0$,

$$\tilde{\mathbf{X}}_{k=0} = \begin{bmatrix} \tilde{x}[0] & 0 & \cdots & 0 \\ & \tilde{x}[0] & & \vdots \\ & & \ddots & 0 \\ \vdots & & & \tilde{x}[0] \\ & & & \\ & & & \\ \tilde{x}[N+L_{cp}-1] & \tilde{x}[N+L_{cp}-2] & \cdots & \tilde{x}[N+L_{cp}-L-1] \end{bmatrix} \tag{3.4.8}$$

Where $\tilde{\mathbf{X}}_{k-1}^{ibi}$ is the Toeplitz matrix which introduces the interblock interference from the $(k-1)^{st}$ block to the k^{th} block. For example, for $k = 0$

$$\tilde{\mathbf{X}}_{k-1=-1}^{ibi} = \begin{bmatrix} 0 & x(-1) & x(-2) & \cdots & x(-L) \\ 0 & 0 & x(-1) & & \\ c & & & \ddots & \\ \vdots & & & & x(-1) \\ & & & & \vdots \\ 0 & & \cdots & & 0 \end{bmatrix} \tag{3.4.9}$$

Stacking the received blocks at all the M_R receive antennas together, we get

$$\begin{bmatrix} \mathbf{y}_k^1 \\ \vdots \\ \mathbf{y}_k^{M_R} \end{bmatrix} = \begin{bmatrix} (\tilde{\mathbf{X}}_k + \tilde{\mathbf{X}}_{k-1}^{ibi})\mathbf{h}_1 \\ \vdots \\ (\tilde{\mathbf{X}}_k + \tilde{\mathbf{X}}_{k-1}^{ibi})\mathbf{h}_{M_R} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_k^1 \\ \vdots \\ \mathbf{v}_k^{M_R} \end{bmatrix} \tag{3.4.10}$$

$$\begin{bmatrix} \mathbf{y}_k^1 \\ \vdots \\ \mathbf{y}_k^{M_R} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{X}}_k + \tilde{\mathbf{X}}_{k-1}^{ibi} & & \\ & \ddots & \\ & & \tilde{\mathbf{X}}_k + \tilde{\mathbf{X}}_{k-1}^{ibi} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{M_R} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_k^1 \\ \vdots \\ \mathbf{v}_k^{M_R} \end{bmatrix} \tag{3.4.11}$$

This may be concisely expressed as

$$\mathbf{y}_k = (\mathbf{I}_{M_R} \otimes \tilde{\mathbf{X}}_k)\mathbf{h} + (\mathbf{I}_{M_R} \otimes \tilde{\mathbf{X}}_{k-1}^{ibi})\mathbf{h} + \mathbf{v}_k \tag{3.4.12}$$

where $\mathbf{y}_k = \left[\mathbf{y}_k^{1T} \cdots \mathbf{y}_k^{M_R T}\right]^T \in \mathbf{C}^{M_R(N+L_{cp})\times 1}$, $\mathbf{h} = \left[\mathbf{h}_1^T \cdots \mathbf{h}_{M_R}^T\right]^T \in \mathbf{C}^{M_R(L+1)\times 1}$. We assume the channel to be constant over a symbol and vary from symbol to symbol, i.e. block fading model which is a first order approximation of a slow fading environment. The channel taps are assumed to be uncorrelated complex Gaussian fading gains. The noise is assumed to zero mean identically distributed spatially and temporally white Gaussian, i.e. $E(\mathbf{v}_k\mathbf{v}_k^H) = \sigma_v^2\mathbf{I}$.

The channel must now be estimated before any receiver processing to detect the data can take place. This estimation is done by exploiting the cyclostationarity induced in the received signal by the transmission of a periodic training signal. A sample averaging operation can be used to extract the channel information. Because of the cyclostationarity, the sample averaging is performed for the entire training period. This allows the training signal to add coherently while the unknown data and the noise add destructively. The received signal is processed as

$$\mathbf{z}_k = \frac{1}{R}(\mathbf{I}_{M_R} \otimes \mathbf{\Psi})\mathbf{y}_k \tag{3.4.13}$$

where $\mathbf{\Psi} = \mathbf{1}_R^T \otimes \mathbf{I}_P$ has the following structure

$$\mathbf{\Psi} = \left[\mathbf{I}_P \cdots \mathbf{I}_P\right] = \begin{bmatrix} 1 & & 1 & & & 1 & \\ & \ddots & & \ddots & \cdots & & \ddots \\ & & 1 & & 1 & & 1 \end{bmatrix} \tag{3.4.14}$$

This may be written as

$$\mathbf{z}_k = \frac{1}{R}(\mathbf{I}_{M_R} \otimes \mathbf{\Psi}\tilde{\mathbf{X}}_k)\mathbf{h} + \frac{1}{R}(\mathbf{I}_{M_R} \otimes \mathbf{\Psi}\tilde{\mathbf{X}}_{k-1}^{ibi})\mathbf{h} + \frac{1}{R}(\mathbf{I}_{M_R} \otimes \mathbf{\Psi})\mathbf{v}_k \tag{3.4.15}$$

Now $\tilde{\mathbf{X}}_k = \mathbf{X}_k + \mathbf{C}$, where $\mathbf{X}_k$ and $\mathbf{C}$ are the Toeplitz matrices constructed from $\mathbf{x}_k$ and $\mathbf{c}$ respectively. Similarly $\tilde{\mathbf{X}}_{k-1}^{ibi} = \mathbf{X}_{k-1}^{ibi} + \mathbf{C}^{ibi}$. Because of the periodic nature of $\mathbf{c}$, we have $\mathbf{C} + \mathbf{C}^{ibi} = \mathbf{1}_R \otimes \mathbf{U}$ where $\mathbf{U}$ is a $P \times (L+1)$

circulant matrix constructed from $\mathbf{u}$

$$\mathbf{U} = \begin{bmatrix} u_1 & u_P & \cdots & u_{P-L+1} \\ u_2 & u_1 & & \vdots \\ \vdots & \vdots & & u_P \\ u_P & u_{P-1} & & u_{P-L} \end{bmatrix} \tag{3.4.16}$$

Now by substituting $\tilde{\mathbf{X}}_k$, $\tilde{\mathbf{X}}_{k-1}^{ibi}$, $\mathbf{C} + \mathbf{C}^{ibi}$ and $\mathbf{\Psi} = \mathbf{1}_R^T \otimes \mathbf{I}_P$, we have

$$\mathbf{z}_k = \frac{1}{R}(\mathbf{I}_{M_R} \otimes \mathbf{1}_R^T \otimes \mathbf{I}_P)(\mathbf{I}_{M_R} \otimes \mathbf{1}_R \otimes \mathbf{U})\mathbf{h} + \frac{1}{R}\big(\mathbf{I}_{M_R} \otimes \mathbf{\Psi}(\mathbf{X}_k + \mathbf{X}_{k-1}^{ibi})\big)\mathbf{h} + \frac{1}{R}(\mathbf{I}_{M_R} \otimes \mathbf{\Psi})\mathbf{v}_k$$

which becomes

$$\mathbf{z}_k = \underbrace{(\mathbf{I}_{M_R} \otimes \mathbf{U})\mathbf{h}}_{\text{term due to training}} + \underbrace{\frac{1}{R}\big(\mathbf{I}_{M_R} \otimes \mathbf{\Psi}(\mathbf{X}_k + \mathbf{X}_{k-1}^{ibi})\big)\mathbf{h}}_{\text{term due to unknown data}} + \underbrace{\frac{1}{R}(\mathbf{I}_{M_R} \otimes \mathbf{\Psi})\mathbf{v}_k}_{\text{term due to noise}} \tag{3.4.17}$$

We note that the terms due to unknown data and noise involve a sum of R uncorrelated random variables. In the subsequent discussion, we assume that both the noise and the signal constellation are zero mean. We now use the fact that when uncorrelated random variables are added, the variance of the resulting signal is the sum of the variances of the original signals. Assuming zero mean random vectors $\mathbf{x}$ and $\mathbf{y}$, we have

$$E\big((\mathbf{x}+\mathbf{y})^H(\mathbf{x}+\mathbf{y})\big) = E(\mathbf{x}^H\mathbf{x}) + E(\mathbf{y}^H\mathbf{y}) + \underbrace{2 \times Real\{E(\mathbf{y}^H\mathbf{x})\}}_{=0} \tag{3.4.18}$$

We also know that by multiplying a random variable with a constant c, the power is changed by c^2

$$E(c\mathbf{x}^H c\mathbf{x}) = c^2 E(\mathbf{x}^H\mathbf{x}) \tag{3.4.19}$$

Using these facts, it is easy to see that the power of the second and third terms in equation (3.4.17) will be reduced by a factor of R due to the sample averaging operation. This is the idea behind periodic superimposed training. Assuming R can be made as large as desired, these undesirable terms can be made vanishingly small thereby facilitating estimation of the channel. The least squares approximation can then be used to obtain the channel as [54]

$$\hat{\mathbf{h}} = (\mathbf{I}_{M_R} \otimes \mathbf{U}^H\mathbf{U})^{-1}(\mathbf{I}_{M_R} \otimes \mathbf{U}^H)\mathbf{z}_k \tag{3.4.20}$$

As opposed to other estimators in classical estimation theory, the least squares estimator makes no probabilistic assumptions about the received data. For this reason, no optimality claims can be made about a least squares estimate in general. The advantage of not specifying the probability density function is that the least squares estimator can be used for a wide range of problems. However, it requires the assumption of a model for the received signal based on unknown parameters and then it tries to fit these unknown parameters in such a way so as to minimize the sum of the squares of the error between the received noisy samples and the noiseless modeled data. The error of the least squares approach can thus come from two sources, error caused by any observation noise and any modeling errors resulting from the choice of an inappropriate model for the problem. While the estimator always tries its best to fit the parameters according to the given model, an incorrect choice of a model can completely corrupt the estimate rendering it useless. Fortunately for our problem, the model is already known apriori and we do not need to choose one so the errors will only come from the observation noise which is composed of both AWGN and interference from the unknown data in this case.

The least squares estimator is generally used in situations where an optimal estimator cannot be found or where the computational burden of obtaining an optimal estimator may be cumbersome. The optimal minimum

variance unbiased (MVU) estimator may not always exist. Even if it does exist, it may not easily be found. Although the maximum likelihood estimator (ML) is asymptotically optimal and always exists, finding it is often not very easy and sometimes requires resorting to iterative methods to find an extremum of the likelihood function. In contrast, the least squares estimator always exists and always has a closed form solution. The least squares estimator coincides with the best linear unbiased estimator (BLUE) and the MVU estimator under some special circumstances. For estimation of the parameters of a linear model, the least squares estimator coincides with the BLUE if the observations are zero mean, independent and identically distributed. If in addition, the observations are Gaussian, the least squares estimator coincides with the MVU estimator.

In our situation, it is not straightforward to characterize the effective noise term because of the contribution of the unknown data and we thus have to resort to linear least squares without staking any claims to optimality. The choice of the training sequence $\mathbf{u}$ will affect the estimate through $\mathbf{U}^H\mathbf{U}$. The training sequence must be such that $\mathbf{U}^H\mathbf{U}$ is full rank for a solution to exist. Theoretically, any training sequence that generates a full rank matrix is equally good. However, practical considerations dictate that the condition number should be as small as possible to avoid instability in inverting $\mathbf{U}^H\mathbf{U}$ caused by roundoff errors [55]. Another issue that must be kept in mind when designing training sequences is the computational complexity. In both signal processing and coding for communications, the computational burden of any algorithm can be just as important an issue as the performance of the algorithm. Here we see that a single matrix inversion of $\mathbf{U}^H\mathbf{U}$ is required as

$$(\mathbf{I}_{M_R} \otimes \mathbf{U}^H\mathbf{U})^{-1} = (\mathbf{I}_{M_R} \otimes (\mathbf{U}^H\mathbf{U})^{-1}) \tag{3.4.21}$$

However, if the training sequence is chosen such that $\mathbf{U}$ is unitary, this matrix inversion is no longer necessary and

$$\hat{\mathbf{h}} = (\mathbf{I}_{M_R} \otimes \mathbf{U}^H)\mathbf{z}_k. \tag{3.4.22}$$

An example of such a sequence is the Kronecker Delta $\mathbf{u} = \begin{bmatrix} \rho P & 0 & \cdots & 0 \end{bmatrix}^T$ which results in a scaled identity matrix $\mathbf{U} = \rho P \mathbf{I}_R$ where P is the period of the periodic training.

In theory the noise terms in (3.4.17) can be made as small as desired by choosing large enough R to obtain a reliable estimate. In practice, the choice of R is constrained to be moderate by various factors including channel variations, processing delay, memory constraints, etc. This has some serious implications because we have already shown that the power of the noise terms only decreases linearly in R. Consequently, the unknown data will not completely cancel itself out with the sample averaging operation. It will interfere with the training. The channel estimator would be inaccurate even in a noise free environment and an error floor would be observed with respect to SNR due to the interference. Ideally, as R approaches infinity, the noise and interference power approaches zero and a perfect channel estimate can be obtained with superimposed training where the pilot is transmitted at an arbitrary small but non zero power ρ. However, with moderate R, the noise and data interference will be present. The accuracy of the channel estimate formed will depend on the training signal to data interference and noise ratio. While increasing ρ will improve the channel estimate accuracy, it will leave lesser power $1 - \rho$ for data transmission. On the other hand, a lower ρ will also affect the system through a mismatched channel estimate.

3.4.2 Iterative Channel Estimation

We therefore propose an iterative semiblind approach to channel estimation and data detection to overcome this difficulty. An initial channel estimate

is formed through the superimposed pilot and is used to detect the data. Since we are discussing multicarrier modulation, full channel equalization is not necessary and channel shortening equalizers are employed so that the interblock interference fits within the cyclic prefix which is to be discarded. This detected data is then assumed correct and used along with the received signal to re-estimate the channel. The new estimate is again used for detection and so on. The proposed scheme is summarized in algorithm 1. A

Algorithm 1 Iterative Superimposed Training Based Channel Estimation and Data Detection for OFDM

1. At the transmitter, choose a periodic training sequence with period greater than or equal to the channel delay spread and superimpose it onto the signal to be transmitted.
2. At the receiver form an initial estimate of the channel coefficients from the superimposed training signal.
3. Peel of the contribution of the superimposed training in the received signal using the channel estimate.
4. Design a channel shortener from the channel estimate if necessary and filter the signal with the channel shortener.
5. Demodulate the signal with an FFT followed by a single tap per tone equalizer and detect the transmit data.
6. Use the estimates of the transmit data to re-estimate the channel.
7. Iterate steps 3 to 6.

block diagram of the proposed receiver architecture for a SIMO OFDM system is shown in figure 3.1 where the dotted line from the detected data to the channel estimator indicates that the detected data is reused for channel estimation in the subsequent iterations.

3.4.3 MMSE Channel Shortening

Once the channel estimates are available, receiver processing for data detection can begin. If the cyclic prefix length used is sufficient, the processing involves peeling off the training, followed by discarding the cyclic prefix and

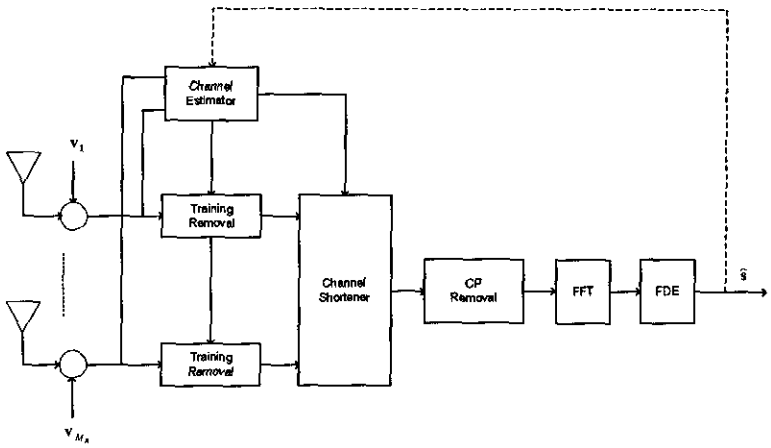

Figure 3.1. Block diagram

an FFT operation at each receive antenna. The signals at the receive antennas are then coherently combined after a single tap per tone equalization to remove any phase rotations. Mathematically, (3.4.12) can be written as

$$\mathbf{y}_k = \mathbf{H}(\mathbf{x}_k + \mathbf{c}) + \mathbf{H}^{ibi}(\mathbf{x}_{k-1} + \mathbf{c}) + \mathbf{v}_k \tag{3.4.23}$$

where $\mathbf{H} = \begin{bmatrix} \mathbf{H}_1^T & \cdots & \mathbf{H}_{M_R}^T \end{bmatrix}^T$ and $\mathbf{H}_\nu$ is the convolution matrix for the channel between the transmitter and receiver ν, i.e.

$$\mathbf{H}_\nu = \begin{bmatrix} h_\nu[0] & & & & & \\ h_\nu[1] & h_\nu[0] & & & & \\ \vdots & & & & & \\ h_\nu[L] & & \cdots & h_\nu[0] & & \\ & h_\nu[L] & & \cdots & h_\nu[0] & \\ & & & \ddots & & \\ & & & & h_\nu[L] \quad \cdots & h_\nu[0] \end{bmatrix} \tag{3.4.24}$$

and the IBI introducing matrix has the form

$$\mathbf{H}_{\nu}^{ibi} = \begin{bmatrix} 0 & \cdots & 0 & h_\nu[L] & \cdots & h_\nu[1] \\ & & & & \ddots & \\ & & & & & h_\nu[L] \\ \vdots & & & & & 0 \\ & & & & & \vdots \\ 0 & & \cdots & & & 0 \end{bmatrix} \tag{3.4.25}$$

If the cyclic prefix length is sufficient, these matrices form a circulant matrix, otherwise we require a channel shortening filter so that the effective channel gives a circulant structure. Using the channel estimate to peel off the training

$$\mathbf{y}_k - (\hat{\mathbf{H}} + \hat{\mathbf{H}}^{ibi})\mathbf{c} = \hat{\mathbf{H}}\mathbf{x}_k + \underbrace{\tilde{\mathbf{H}}(\mathbf{c}+\mathbf{x}_k) + \tilde{\mathbf{H}}^{ibi}(\mathbf{c}+\mathbf{x}_{k-1}) + \mathbf{v}_k}_{\tilde{\mathbf{v}}_k} \tag{3.4.26}$$

where $\tilde{\mathbf{H}} = \mathbf{H} - \hat{\mathbf{H}}$. The cyclic prefix removal and DFT operations are given by

$$\mathbf{r}_k = (\mathbf{I}_{M_R} \otimes \mathbf{FT}_{DCP})(\hat{\mathbf{H}}\mathbf{x}_k + \tilde{\mathbf{v}}_k) \tag{3.4.27}$$

where $\mathbf{T}_{DCP} = \begin{bmatrix} \mathbf{0}_{N \times L_{CP}} & \mathbf{I}_N \end{bmatrix}$. The cyclic prefix addition at the transmitter and removal at the receiver gives the channel matrices a circulant structure. A key property of circulant matrices is that they can be factorized as $\mathbf{H}_c = \mathbf{F}^H\mathbf{D}\mathbf{F}$, where $\mathbf{F}$ is the familiar DFT matrix and $\mathbf{D}$ is a diagonal matrix. Thus these matrices can be diagonalized through a channel independent transformation. This is the key idea behind OFDM and DMT - the interblock interference and the intercarrier interference are eliminated without the requirement of channel knowledge at the transmitter as opposed to vector coding. We thus get

$$\mathbf{r}_k = \underbrace{\begin{bmatrix} \mathbf{D}_1^T & \cdots & \mathbf{D}_{M_R}^T \end{bmatrix}^T}_{\boldsymbol{D}} \mathbf{s}_k + (\mathbf{I}_{M_R} \otimes \mathbf{FT}_{DCP})\tilde{\mathbf{v}}_k \tag{3.4.28}$$

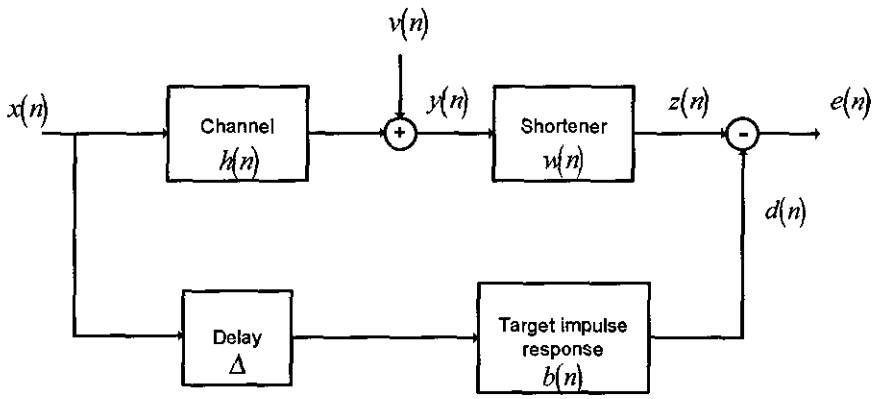

Figure 3.2. MMSE channel shortener.

and the transmitted signal is estimated as

$$\hat{\mathbf{s}}_k = (\mathbf{D}^H\mathbf{D})^{-1}\mathbf{D}^H\mathbf{r}_k \tag{3.4.29}$$

which is implemented as scaler multiplications because of the diagonal structure of $\mathbf{D}_i$.

However if the cyclic prefix length is insufficient, then the convolution matrix does not become circulant. The insufficient cyclic prefix length introduces interblock interference because all the corrupted samples are not discarded at the beginning of an OFDM symbol. With ICI and IBI, the OFDM receiver is considerably complicated as these must be mitigated. Because of the transmission of the cyclic prefix, it does not make sense to completely equalize the channel but instead to shorten it so that the effective channel order fits within the cyclic prefix. An MMSE channel shortening equalizer is thus introduced. The channel shortener is designed to minimize the mean square error between the effective shortened impulse response and a desired target impulse response of the target length as shown in figure 3.2. Mathematically

$$J = E(|e[n]|^2) = E\big(|\sum_{l=0}^{L_d} b^*(l)x(n-\Delta-l) - \sum_{l=0}^{L_s} w^*(l)y(n-l)|^2\big) \tag{3.4.30}$$

where L_s is the order of the channel shortener and L_d is the desired channel order. In vector form, the cost may be written as

$$J = \mathbf{b}^H \tilde{\mathbf{R}}_{xx}\mathbf{b} - \mathbf{b}^H \tilde{\mathbf{R}}_{xy}\mathbf{w} - \mathbf{w}^H \tilde{\mathbf{R}}_{yx}\mathbf{b} + \mathbf{w}^H \mathbf{R}_{yy}\mathbf{w} \tag{3.4.31}$$

where the $(m,n)^{th}$ entry of the convolution matrices is given by

$$[\tilde{\mathbf{R}}_{xx}]_{m,n} = r_{xx}(m-n) \quad 0 \le m < L_d, 0 \le n < L_d \tag{3.4.32}$$

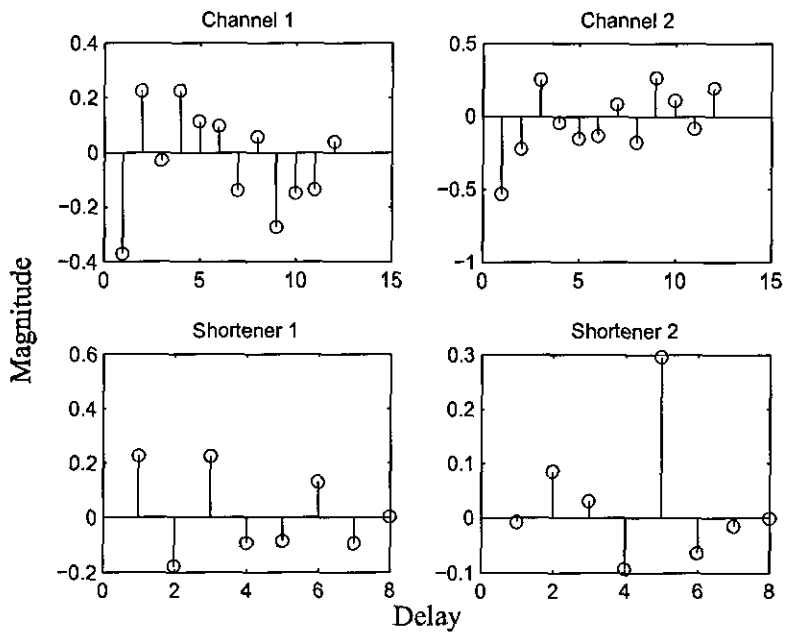

Figure 3.3. Impulse responses of the channels and the channel shortening equalizers.

$$[\tilde{\mathbf{R}}_{xy}]_{m,n} = r_{xy}(m-n+\delta) \quad 0 \le m < L_d, 0 \le n < L_s \tag{3.4.33}$$

$$[\mathbf{R}_{yy}]_{m,n} = r_{yy}(m-n+\delta) \quad 0 \le m < L_s, 0 \le n < L_s \tag{3.4.34}$$

where $r_{xx}(n) = E(x[m]x^*[m+n])$. Differentiating with respect the the target impulse response $\mathbf{w}$, we get [43]

$$\frac{\partial J}{\partial \mathbf{w}^*} = \mathbf{R}_{yy}\mathbf{w} - \tilde{\mathbf{R}}_{yx}\mathbf{b} \tag{3.4.35}$$

$$\Rightarrow \mathbf{w} = \mathbf{R}_{yy}^{-1}\tilde{\mathbf{R}}_{yx}\mathbf{b} \tag{3.4.36}$$

Substituting this value of the target impulse response into the cost gives [43]

$$J = \mathbf{b}^H \underbrace{(\tilde{\mathbf{R}}_{xx} - \tilde{\mathbf{R}}_{xy}\tilde{\mathbf{R}}_{yy}^{-1}\tilde{\mathbf{R}}_{yx})}_{\mathbf{R}_{x|y}}\mathbf{b} \tag{3.4.37}$$

which must be minimized subject to the unit energy constraint $\mathbf{b}^H\mathbf{b} = 1$.

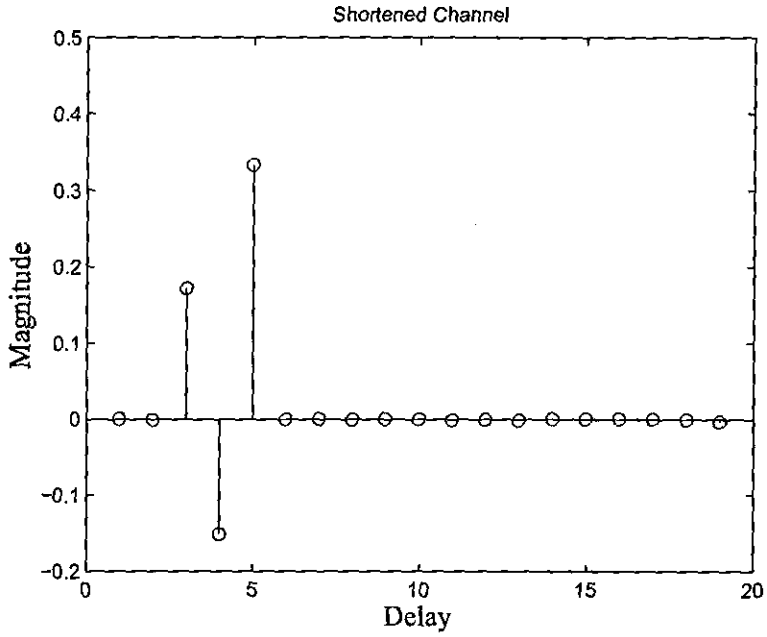

Figure 3.4. Shortened channel impulse response.

This is simply solved as an eigenvalue problem. The channel shortener is given by the eigenvector associated with the smallest eigenvalue of $\mathbf{R}_{x|y}$ and the mean squared error is given by this eigenvalue [43].

As an example, a SIMO system with one transmit antenna and two receive antennas and channel impulse response length of twelve taps is short-

ened to an effective length of three taps using channel shortening equalizers of length eight at the receive antennas at SNR = 10dB. The channel impulse responses and channel shortener impulse responses are shown in figure 3.3 while figure 3.4 shows the shortened impulse response.

3.4.4 Simulation Results

With the architecture described, one major issue is yet to be resolved. If the total transmission power is fixed, how should the power be divided between the training and data. We will resort to simulations to resolve this question. The estimation MSE is plotted against ρ (the ratio of the total power allocated to training) in figure 3.5 at 20 dB where we have used $M_R = 2$, $L_{cp} = 2$, $L_s = 8$ and $L = 6$. The modulation is 4 QAM. It can be observed that although the initial estimate improves with increasing ρ and hence increasing power allocation to training, our iterative scheme does not. For our iterative channel estimation and equalization scheme it can be observed that the channel estimate quality degrades with allocation of $\rho > 0.2$ to the training. So while in conventional training, it might make sense to allocate more resources to the pilot for a better channel estimate, we see that in our scheme doing so actually degrades the channel estimate. With our scheme, a sensible balance has to be achieved for power allocation at $\rho = 0.2$. It can be seen that the channel estimate and the BER performance improves with increasing ρ for $\rho < 0.2$. This is because of the large improvements in the quality of the initial channel estimate with increasing ρ. In this region, the performance of the iterative scheme is largely limited by the poor quality of the initial channel estimate available. However, for $\rho > 0.2$, the effect of the improvements in the initial channel estimate is offset by the limited power available for data transmission. Because the data transmission has less power to combat the noise, this results in more detection errors. The detected sequence is reused for channel estimation in the subsequent iterations so that the improvement in the channel estimate is limited by the errors in

data detection. In this region, the performance of the iterative scheme is therefore largely limited by the reduced data signal to noise ratio.

It can be observed that at $\rho = 0.2$, the iterative semiblind scheme can provide a better channel estimate than simple superimposed training with $\rho = 0.9$. The BER results with the same parameters shown in figure 3.6 indicate that the iterative process converges quickly within two iterations and indicates that $0.1 \leq \rho \leq 0.2$ minimizes the BER for the iterative semiblind algorithm.

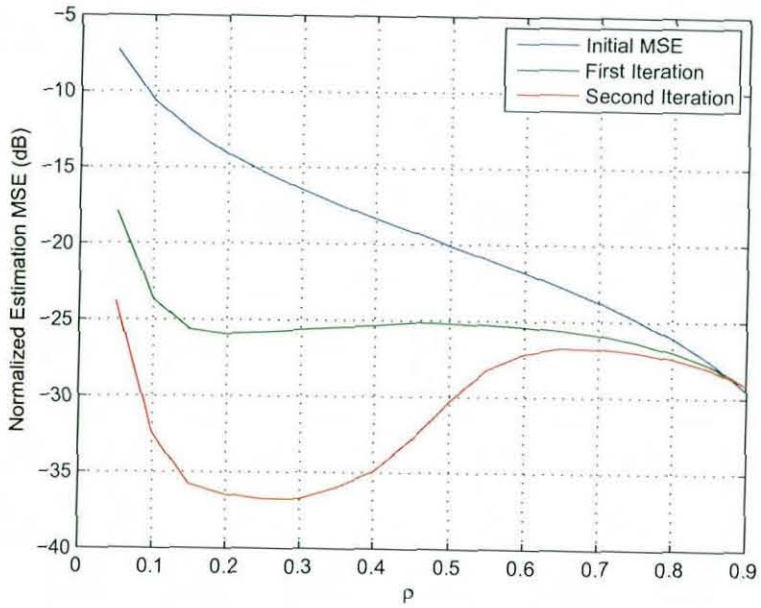

Figure 3.5. Normalized Estimation mean square error is plotted against ρ for various iterations at SNR = 20dB.

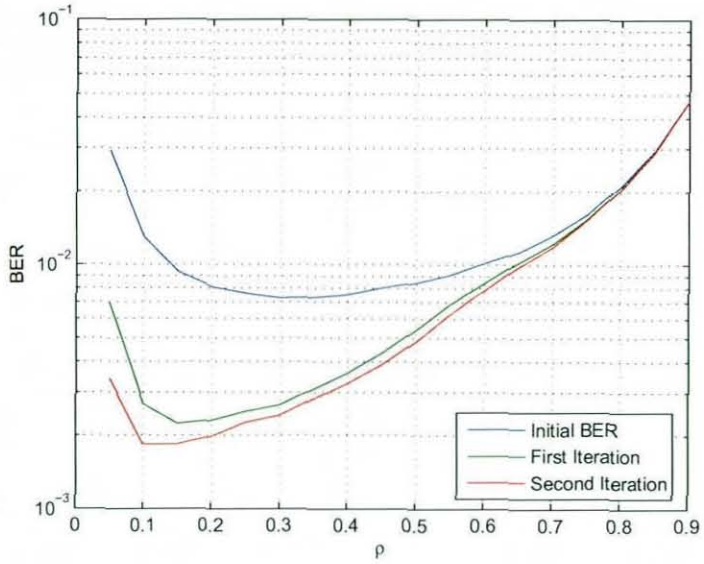

Figure 3.6. BER is plotted against ρ for various iterations at SNR = 20dB.

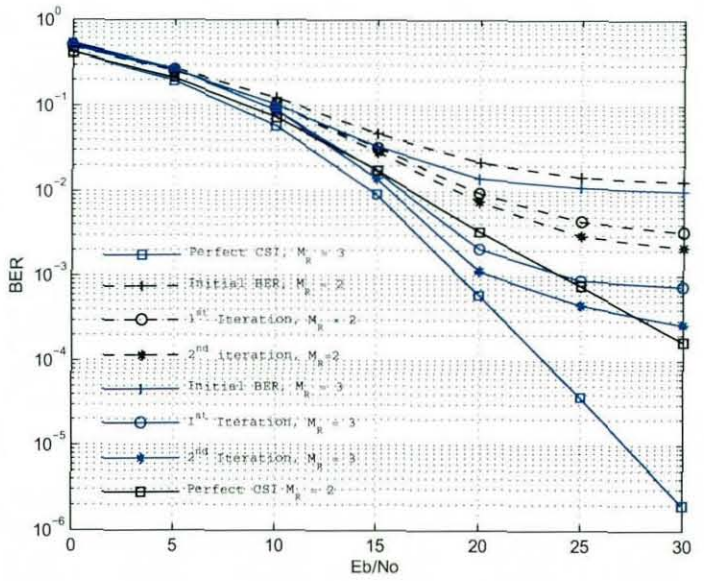

Figure 3.7. BER is plotted against Eb/No for various iterations with $\rho = 0.2$. The curves for the case of two receive antennas are dashed and those for three receive antennas are solid.

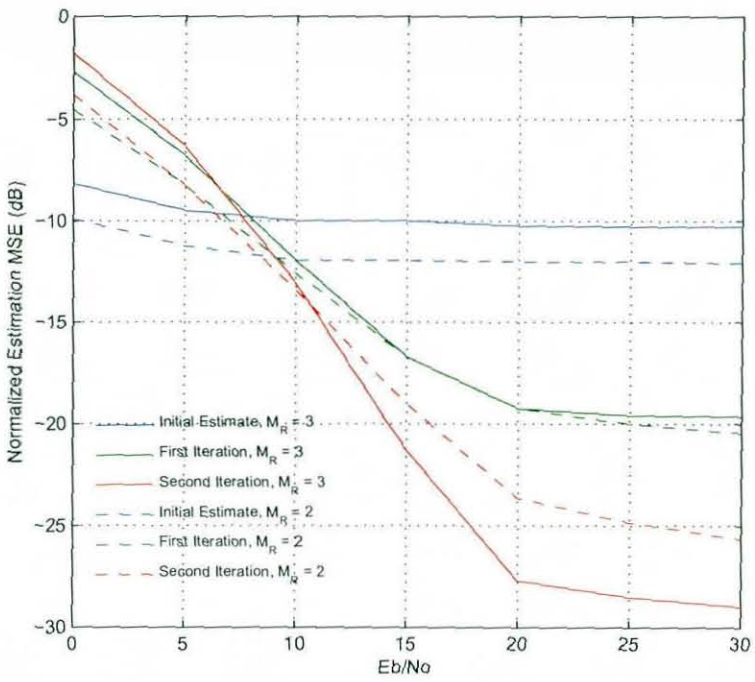

Figure 3.8. Normalized estimation mean square error is plotted against Eb/No for various iterations with $\rho = 0.2$. The curves for the case of two receive antennas are dashed and those for three receive antennas are solid.

Choosing $\rho = 0.2$, we simulated the same system for BER and MSE results against SNR. These are provided in figures 3.7 and 3.8 respectively. These results indicate that significant performance improvements are possible with the use of iterative detection and channel estimation for in conjunction with superimposed training in SIMO OFDM.

3.5 MIMO OFDM

We now consider semiblind channel estimation for a MIMO OFDM system in the context of point to point MIMO. MIMO communications has the capability to provide a host of advantages including array gain, diversity gain, interference reduction and spatial multiplexing gain that will be discussed in more detail later. However, all these gains come with an increased computational complexity as all taps in SISO communications are now replaced

by matrix taps. Apart from intersymbol interference, the system has to now contend with multiple stream interference[5] (MSI). Complete temporal equalization of MIMO systems could become prohibitive. This computational burden may be alleviated by the use of OFDM. OFDM technology is overlayed onto a SU-MIMO system where the IFFT/FFT and CP operations are performed at each of the transmit and receive antennas. With a sufficient cyclic prefix, a frequency selective MIMO channel is decoupled into a set of parallel frequency flat MIMO channels. Thus the multipaths are inherently resolved and the system has only to contend with MSI. However the transmission of a cyclic prefix is a loss in throughput which can be mitigated somewhat by a channel shortening equalizer that the computational resources allow.

3.5.1 Superimposed Training

We now consider superimposed training for a MIMO OFDM system for spatial multiplexing. Explicit training or frequency division multiplexed training for MIMO-OFDM has been studied in [56] among others. The material in this section is similar to the material on superimposed training for SIMO systems provided in section 3.4.1 with the essential difference being that multiple transmit antennas are now included into the mathematical framework and the required conditions to form a channel estimate are derived and training design for the multiple transmitters is discussed.

We consider a system with M_T transmit and M_R receive antennas, an OFDM symbol size N and maximum delay spread L and assume a cyclic prefix L_{cp}. The M_T input streams are parsed into blocks of size N to form OFDM symbols for transmission. The block index k is then related to the serial index as

$$k = \lfloor n/N \rfloor \tag{3.5.1}$$

[5] For a MIMO-MA channel, which is a uplink channel with multiple independent transmitters accessing one base, this interference is termed as multiple access interference (MAI).

The frequency domain vector corresponding to the k^{th} OFDM symbol is then $\mathbf{s}_k = \begin{bmatrix} \mathbf{s}^T[kN] & \cdots & \mathbf{s}^T[kN+N-1] \end{bmatrix}^T \in \mathbf{C}^{M_T N \times 1}$ and $\mathbf{s}[i] = \begin{bmatrix} s_1[i] & \cdots & s_{M_T}[i] \end{bmatrix}^T \in \mathbf{C}^{M_T \times 1}$. The average transmit power of each antenna is normalized to one. The frequency domain signal is converted to the time domain

$$\mathbf{x}_k = (\mathbf{F}^H \otimes \mathbf{I}_{M_T})\mathbf{s}_k \tag{3.5.2}$$

where $\mathbf{x}_k \in \mathbf{C}^{M_T N \times 1}$. A cyclic prefix is inserted and a pilot signal is superimposed onto the data to form the transmit signal

$$\tilde{\mathbf{x}}_k = (\mathbf{T}_{\mathbf{cp}} \otimes \mathbf{I}_{M_T})\mathbf{x}_k + \mathbf{c} \tag{3.5.3}$$

where $\tilde{\mathbf{x}}_k, \mathbf{c} \in \mathbf{C}^{(N+L_{cp})M_T \times 1}$. The received signal is then given by

$$\mathbf{y}_k = \tilde{\mathbf{X}}_k \mathbf{h} + \tilde{\mathbf{X}}_{k-1}^{ibi} \mathbf{h} + \mathbf{v}_k \tag{3.5.4}$$

where $\mathbf{y}_k \in \mathbf{C}^{(N+L_{cp})M_R \times 1}$ and $\tilde{\mathbf{X}}_k$ is a block Toeplitz matrix

$$[\tilde{\mathbf{X}}_k]_{m,n} = \begin{cases} \tilde{\mathbf{X}}[m-n+k(N+L_{cp})], & m-n \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{3.5.5}$$

For example, for $k = 0$ this is

$$\tilde{\mathbf{X}}_{k=0} = \begin{bmatrix} \tilde{\mathbf{X}}[0] & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{\mathbf{X}}[1] & \tilde{\mathbf{X}}[0] & & \vdots \\ & & \ddots & \mathbf{0} \\ \vdots & & & \tilde{\mathbf{X}}[0] \\ & & & \vdots \\ \tilde{\mathbf{X}}[N+L_{cp}-1] & \tilde{\mathbf{X}}[N+L_{cp}-2] & \cdots & \tilde{\mathbf{X}}[N+L_{cp}-L-1] \end{bmatrix} \tag{3.5.6}$$

where $\tilde{\mathbf{X}}[i] = \tilde{\mathbf{x}}^T[i] \otimes \mathbf{I}_{M_R}$, i.e.

$$\tilde{\mathbf{X}}[i] = \begin{bmatrix} \tilde{x}_1[i]\mathbf{I}_{M_R} & \tilde{x}_2[i]\mathbf{I}_{M_R} & \cdots & \tilde{x}_{M_T}[i]\mathbf{I}_{M_R} \end{bmatrix} \tag{3.5.7}$$

where $\tilde{x}_\kappa[i]$ is the i^{th} symbol transmitted from the κ^{th} transmit antenna and the interblock interference (IBI) is captured by the block Toeplitz matrix

$$[\tilde{\mathbf{X}}_{k-1}^{ibi}]_{m,n} = \begin{cases} \tilde{\mathbf{X}}[m-n+k(N+L_{cp})], & m-n \leq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{3.5.8}$$

Again for $k = 0$, this can be written as

$$\tilde{\mathbf{X}}_{k-1=-1}^{ibi} = \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{X}}[-1] & \cdots & \tilde{\mathbf{X}}[-L] \\ & \mathbf{0} & \ddots & \tilde{\mathbf{X}}[-L+1] \\ & & & \vdots \\ \vdots & & & \tilde{\mathbf{X}}[-1] \\ & & & \mathbf{0} \\ & & & \\ \mathbf{0} & \cdots & & \mathbf{0} \end{bmatrix} \tag{3.5.9}$$

The channel vector is defined as $\mathbf{h} = \begin{bmatrix} \mathbf{h}^T[0] & \cdots & \mathbf{h}^T[L] \end{bmatrix}^T$ and $\mathbf{h}[i] = \begin{bmatrix} \mathbf{h}_1^T[i] & \cdots & \mathbf{h}_{M_T}^T[i] \end{bmatrix}$ where $\mathbf{h}_j[i] = \begin{bmatrix} h^{(1,j)}[i] & \cdots & h^{(M_R,j)}[i] \end{bmatrix}$. We assume that the channel is Rayleigh fading and appropriately normalized with an average gain of one between any transmitter receiver pair.

The received signal must be processed for channel estimation. Again we assume that the pilot signal superimposed on the data was periodic but with period $P = M_T(L+1)$. Then we can estimate the channels at each receive antenna from all transmit antennas by a sample averaging operation

$$\mathbf{z}_k = \frac{1}{R}\Psi \mathbf{y}_k \tag{3.5.10}$$

where $\mathbf{\Psi} = \mathbf{1}_R^T \otimes \mathbf{I}_{PM_R} = \begin{bmatrix} \mathbf{I}_{PM_R} & \cdots & \mathbf{I}_{PM_R} \end{bmatrix}$. We can write $\tilde{\mathbf{X}}_k = \mathbf{X}_k + \mathbf{C}$ where $\mathbf{X}_k$ and $\mathbf{C}$ are appropriately formed from the vectors $\mathbf{x}$ and $\mathbf{c}$ respectively. Because of the periodic nature of the training, we can write $\mathbf{C} + \mathbf{C}^{ibi} = \mathbf{1}_R \otimes \mathbf{U}$ where $\mathbf{U}$ is a $M_R P \times M_R M_T (L+1)$ block circulant matrix formed from one period of the training

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}[0] & \mathbf{U}[P] & \cdots & \mathbf{U}[P-L+1] \\ \mathbf{U}[1] & \mathbf{U}[0] & & \\ \vdots & & & \vdots \\ \mathbf{U}[P] & \mathbf{U}[P-1] & \cdots & \mathbf{U}[P-L] \end{bmatrix} \tag{3.5.11}$$

where $\mathbf{U}[i] = \begin{bmatrix} u_1[i]\mathbf{I}_{M_R} & \cdots & u_{M_T}[i]\mathbf{I}_{M_R} \end{bmatrix}$. By substituting equation (3.5.4) into (3.5.10), and using $\tilde{\mathbf{X}}_k = \mathbf{X}_k + \mathbf{C}$ and $\mathbf{C} + \mathbf{C}^{ibi} = \mathbf{1}_R \otimes \mathbf{U}$, we get

$$\mathbf{z}_k = \frac{1}{R}(\mathbf{1}_R^T \otimes \mathbf{I}_{PM_R})\big((\mathbf{1}_R \otimes \mathbf{U})\mathbf{h} + \mathbf{X}_k\mathbf{h} + \mathbf{X}_{k-1}^{ibi}\mathbf{h} + \mathbf{v}_k\big) \tag{3.5.12}$$

$$\mathbf{z}_k = \mathbf{U}\mathbf{h} + \underbrace{\frac{1}{R}(\mathbf{1}_R^T \otimes \mathbf{I}_{PM_R})(\mathbf{X}_k\mathbf{h} + \mathbf{X}_{k-1}^{ibi}\mathbf{h} + \mathbf{v}_k)}_{\tilde{\mathbf{v}}_k} \tag{3.5.13}$$

Again, the channel can be obtained by the least squares approximation

$$\hat{\mathbf{h}} = (\mathbf{U}^H\mathbf{U})^{-1}\mathbf{U}^H\mathbf{z}_k \tag{3.5.14}$$

We note that because the dimensions of $\mathbf{U}$ are $M_R P \times M_R M_T (L+1)$, the condition required for a unique solution is $P \geq (L+1)M_T$. Also, the pilot signals from the M_T transmitters should be designed such that $\mathbf{U}^H\mathbf{U}$ is well conditioned. The matrix inversion can be avoided altogether if $\mathbf{U}$ is unitary, by an appropriate choice of training. An example of such a pilot is when $\mathbf{c}$ is a train of Kronecker deltas with a period of L. This corresponds to the situation where all transmitters are using an impulse train for channel identification and the other transmitters remain silent when any transmitter is sounding the channel. The interference from the unknown data due

to limited symbol length is still a major obstacle as it results in an early error floor. With constraints like processing delay, memory, etc. precluding an increase in the symbol length, the naive solution would be to transmit the pilot at a higher power allowing for a better channel estimate and link performance. This is not feasible for several reasons. It would increase interference in the network thereby automatically lowering network capacity. Higher transmission power would also reduce battery life. Although error control coding could be introduced, codes also require something to work on - bandwidth for convolutional and binary linear block codes or power for trellis coded modulation (TCM). Fortunately, all this can be avoided with iterative semiblind processing at the receiver. We propose to employ the detected data for channel re-estimation. The new channel estimate is then used for data detection. This process is repeated and it converges to a solution much closer to what would be obtained with perfect channel knowledge.

Once an initial channel estimate becomes available, receiver processing can begin. If the cyclic prefix length is sufficient, the received signal is processed by dropping the cyclic prefix and then translating it back to the frequency domain via FFT operations at each of the receive antennas. In the general case, the cyclic prefix will be insufficient and the received signal must be processed by a channel shortener. We choose to apply time domain channel shortener although a per tone equalizer [47] is equally possible. We next describe channel shortening before we proceed to further receiver processing.

3.5.2 MIMO Channel Shortening

We have seen from the previous sections that

$$\mathbf{y}_k = \tilde{\mathbf{X}}_k \mathbf{h} + \tilde{\mathbf{X}}_{k-1}^{ibi} \mathbf{h} + \mathbf{v}_k \tag{3.5.15}$$

Because of the commutativity of the product of a Toeplitz matrix and a vector, we may write

$$\mathbf{y}_k = \mathbf{H}\tilde{\mathbf{x}}_k + \mathbf{H}^{ibi}\tilde{\mathbf{x}}_{k-1} + \mathbf{v}_k \tag{3.5.16}$$

where $\mathbf{y}_k = \left[\mathbf{y}^T[k(N+L_{cp})+1] \quad \cdots \quad \mathbf{y}^T[k(N+L_{cp})+N+L_{cp}-1]\right]^T$, $\mathbf{y}[i] = \left[y_1[i] \quad \cdots \quad y_{M_R}[i]\right]^T$ and

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}[0] & \mathbf{0} & & \cdots & & \mathbf{0} \\ \vdots & \ddots & & & & \vdots \\ \mathbf{H}[L] & & \mathbf{H}[0] & \mathbf{0} & & \mathbf{0} \\ \mathbf{0} & & & \ddots & & \vdots \\ \vdots & \mathbf{0} & \mathbf{H}[L] & \cdots & \mathbf{H}[0] & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}[L] & \cdots & \mathbf{H}[0] \end{bmatrix} \tag{3.5.17}$$

where

$$\mathbf{H}[i] = \begin{bmatrix} h_{11}[i] & \cdots & h_{1M_T}[i] \\ & \ddots & \\ h_{M_R1}[i] & \cdots & h_{M_RM_T}[i] \end{bmatrix}$$

Similarly the IBI introducing matrix is

$$\mathbf{H}^{ibi} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}[L] & \cdots & \mathbf{H}[1] \\ & & & & \ddots & \\ & & & & & \mathbf{H}[L] \\ \vdots & & & & & \mathbf{0} \\ & & & & & \vdots \\ \mathbf{0} & & \cdots & & & \mathbf{0} \end{bmatrix} \tag{3.5.18}$$

Now we wish to apply a channel shortening equalizer to these received samples to shorten the effective channel. Let the order of the channel shortening equalizer be L_s so that it processes $L_s + 1$ samples at any time instant.

We note that because of the multiple antennas at the transmitter and the receiver, the channel taps are matrices. Similarly the channel shortening equalizer taps are matrices. Also, we denote the channel shortening equalizer as $\mathbf{W} = \begin{bmatrix} \mathbf{W}_0^T & \cdots & \mathbf{W}_{L_s}^T \end{bmatrix}^T$ where $\mathbf{W}_i$ is an $M_R \times M_T$ dimensional matrix tap. Then the output of the channel shortening equalizer at the n^{th} symbol in the k^{th} block, i.e. at $k(N + L_{cp}) + n$ is

$$\tilde{\mathbf{y}}_k = \mathbf{W}^H \underbrace{\begin{bmatrix} \mathbf{y}_{k(N+L_{cp})+n-L_s} \\ \vdots \\ \mathbf{y}_{k(N+L_{cp})+n} \end{bmatrix}}_{\mathbf{y}_{s,n}} \tag{3.5.19}$$

where $\tilde{\mathbf{y}}_k$ represents the output of the channel shortener. Similar to equation (3.5.16) we can write

$$\mathbf{y}_{s,n} = \underbrace{\begin{bmatrix} \mathbf{H}[L] & \cdots & \mathbf{H}[0] & \mathbf{0} & \cdots & \mathbf{0} \\ & & \ddots & & & \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}[L] & \cdots & \mathbf{H}[0] \end{bmatrix}}_{\mathbf{H}_s} \underbrace{\begin{bmatrix} \mathbf{x}_{k(N+L_{cp})+n-L_s-L-1} \\ \\ \vdots \\ \\ \mathbf{x}_{k(N+L_{cp})+n} \end{bmatrix}}_{\mathbf{x}_{s,n}} + \underbrace{\begin{bmatrix} \mathbf{v}_{k(N+L_{cp})+n-L_s} \\ \vdots \\ \mathbf{v}_{k(N+L_{cp})+n} \end{bmatrix}}_{\mathbf{v}_{s,n}}$$

We denote the target impulse response as $\mathbf{B} = \begin{bmatrix} \mathbf{B}_0^T & \cdots & \mathbf{B}_{L_d}^T \end{bmatrix}^T$ and $\mathbf{B}_i^T$ are matrix taps with dimensions $M_T \times M_T$. Then the desired channel and shortener cascade output is $\mathbf{B}^H \begin{bmatrix} \mathbf{x}_{k(N+L_{cp})+n-\Delta}^T & \cdots & \mathbf{x}_{k(N+L_{cp})+n-\Delta-L_d}^T \end{bmatrix}^T$ and the error vector would

be

$$\mathbf{e}_{s,n} = \tilde{\mathbf{B}}^H \mathbf{x}_{s,n} - \mathbf{W}^H \mathbf{y}_{s,n}$$

where

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{0}_{M_T\Delta \times M_T} \\ \mathbf{B}_0 \\ \vdots \\ \mathbf{B}_{L_d} \\ \mathbf{0}_{M_T(L_s+L-L_d-\Delta+1)\times M_T} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0}_{\Delta M_T \times (L_d+1)M_T} \\ \mathbf{I}_{(L_d+1)M_T} \\ \mathbf{0}_{(L+L_s-L_d-\Delta-1)M_T \times (L_d+1)M_T} \end{bmatrix}}_{\mathbf{\Gamma}} \mathbf{B}$$

Then by the principle of orthogonality $E(\mathbf{e}_{s,n}\mathbf{y}_{s,n}^H) = 0$ [14]. Thus we can write

$$E(\mathbf{e}_{s,n}\mathbf{y}_{s,n}^H) = E\big((\tilde{\mathbf{B}}^H \mathbf{x}_{s,n} - \mathbf{W}^H \mathbf{y}_{s,n})\mathbf{y}_{s,n}^H\big)$$

This implies that

$$\mathbf{W}^H = \tilde{\mathbf{B}}^H \mathbf{R}_{xy} \mathbf{R}_{yy}^{-1} \tag{3.5.20}$$

where $\mathbf{R}_{yy} = E(\mathbf{y}_{s,n}\mathbf{y}_{s,n}^H)$ and $\mathbf{R}_{xy} = E(\mathbf{x}_{s,n}\mathbf{y}_{s,n}^H)$. The error autocorrelation matrix is obtained as [45]

$$\mathbf{R}_{ee} = E(\mathbf{e}_{s,n}\mathbf{e}_{s,n}^H) = E\big((\tilde{\mathbf{B}}^H \mathbf{x}_{s,n} - \mathbf{W}^H \mathbf{y}_{s,n})(\mathbf{x}_{s,n}^H \tilde{\mathbf{B}} - \mathbf{y}_{s,n}^H \mathbf{W})\big) \tag{3.5.21}$$

Simplification and substitution of equation (3.5.20) gives [45]

$$\mathbf{R}_{ee} = \tilde{\mathbf{B}}^H \underbrace{(\mathbf{R}_{xx} - \mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\mathbf{R}_{yx})}_{\mathbf{R}_{x|y}} \tilde{\mathbf{B}} \tag{3.5.22}$$

The channel shortening equalizers are obtained by [45]

$$\mathbf{B}_{opt} = \arg\min_{\mathbf{B}} \operatorname{trace}(\mathbf{R}_{ee}) \quad \text{subject to} \quad \mathbf{B}^H\mathbf{B} = \mathbf{I}_{M_T} \tag{3.5.23}$$

The solution to this problem is given by the eigenvectors corresponding to the smallest eigenvalues of $\mathbf{\Gamma}^H \mathbf{R}_{x|y} \mathbf{\Gamma}$ which are the channel shorteners.

3.5.3 MIMO-OFDM Processing

Because of the commutativity of a Toeplitz matrix and a vector, the output in (3.5.4) can also be expressed as

$$\mathbf{y}_k = \mathbf{H}_k\tilde{\mathbf{x}}_k + \mathbf{H}_k^{ibi}\tilde{\mathbf{x}}_{k-1} + \mathbf{v}_k \tag{3.5.24}$$

where $\mathbf{H}_k$ is a Toeplitz matrix constructed from matrix channel taps

$$[\mathbf{H}_k]_{m,n} = \begin{cases} \mathbf{H}_k[m-n], & 0 \le m-n \le L; \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3.5.25}$$

and the matrix channel taps are

$$\mathbf{H}[i] = \begin{bmatrix} h^{(1,1)}[i] & \cdots & h^{(1,M_T)}[i] \\ & \ddots & \\ h^{(M_R,1)}[i] & \cdots & h^{(M_R,M_T)}[i] \end{bmatrix} \tag{3.5.26}$$

and $\mathbf{H}_k^{ibi}$ is defined accordingly.

The pilot is first peeled off the received signal

$$\mathbf{y}_k - \hat{\mathbf{H}}_k\mathbf{c} = \hat{\mathbf{H}}_k\mathbf{x}_k + \underbrace{\tilde{\mathbf{H}}_k(\mathbf{c}+\mathbf{x}_k) + \mathbf{H}_{k-1}^{ibi}\tilde{\mathbf{x}}_k + \mathbf{v}_k}_{\tilde{\mathbf{v}}_k} \tag{3.5.27}$$

Now in our model the effective noise is $\tilde{\mathbf{v}}_k$. We assume an equivalent Gaussian noise in designing the channel shortener and the processing at the receiver because this noise is difficult to characterize. It must be mentioned here that this approach is strictly sub optimal because the noise term is not properly characterized. If the cyclic prefix is insufficient, the signal is then processed by the channel shortening filter as follows

$$\tilde{y}_k = w_k * (y_k - \hat{h}_k * c_k) \tag{3.5.28}$$

where the $*$ denotes the convolution operation. The signal at the output of the channel shortener is then processed by discarding the cyclic prefix and translating back to the frequency domain

$$\mathbf{r}_k = (\mathbf{F}\mathbf{T}_{dcp} \otimes \mathbf{I}_{M_T})\tilde{\mathbf{y}}_k \tag{3.5.29}$$

The MIMO OFDM receiver is designed to process the received signal block by block. It processes $N + L_{cp}$ received samples per receive antenna at a time. Thus if the cyclic prefix is longer than or equal to the channel memory $L_{cp} \geq L - 1$, the signal appears periodic to the channel resulting in circular convolution. This ensures that IBI is avoided as the corrupted samples are discarded along with the cyclic prefix and allows for simple equalization in the frequency domain. For sufficient length cyclic prefix,

$$(\mathbf{F}\mathbf{T}_{dcp} \otimes \mathbf{I}_{M_T})(\mathbf{H} + \mathbf{H}^{ibi})(\mathbf{T}_{cp}\mathbf{F}^H \otimes \mathbf{I}_{M_T}) = \mathbf{D} \tag{3.5.30}$$

where

$$\mathbf{D} = \text{block diag}(\begin{bmatrix}\mathbf{D}[1] & \cdots & \mathbf{D}[N]\end{bmatrix}) \tag{3.5.31}$$

and

$$\mathbf{D}[i] = \begin{bmatrix} d^{(1,1)}[i] & \cdots & d^{(1,M_T)}[i] \\ & \ddots & \\ d^{(M_R,1)}[i] & \cdots & d^{(M_R,M_T)}[i] \end{bmatrix} \tag{3.5.32}$$

so that

$$\mathbf{r}_k = \mathbf{D}\mathbf{s}_k + \mathbf{v}_k \tag{3.5.33}$$

Note that there is no interference between the tones. The MIMO frequency selective channel has been transformed into N frequency flat MIMO channels.

3.5.4 BLAST

By noting that the FFT and IFFT are both unitary operations, hence simple rotations that do not change the properties of the circularly symmetric complex Gaussian noise, we see that the orthogonal frequency division multiplexing has already resolved the multipath without any noise enhancement. In detecting the transmit signal, it remains to deal with the MSI. The inputs to each of the frequency flat MIMO channels can be recovered through a zero forcing (ZF) equalization or MMSE equalization. Once again, the transmit signal was $\mathbf{s}_k = \begin{bmatrix}\mathbf{s}^T[kN] & \cdots & \mathbf{s}^T[kN+N-1]\end{bmatrix}^T$ where $\mathbf{s}[i] = \begin{bmatrix}s_1[i] & \cdots & s_{M_T}[i]\end{bmatrix}^T$ and the received signal at the output of the FFT demodulators is $\mathbf{r}_k = \begin{bmatrix}\mathbf{r}^T[kN] & \cdots & \mathbf{r}^T[kN+N-1]\end{bmatrix}^T$ where $\mathbf{r}[i] = \begin{bmatrix}r_1[i] & \cdots & r_{M_R}[i]\end{bmatrix}$. The received signal is a mixture of the transmit symbols that can be separated with MMSE equalization as

$$\hat{\mathbf{s}}[i] = \underbrace{(\mathbf{D}[i]^H\mathbf{D}[i] + \sigma^2\mathbf{I}_{M_T})^{-1}\mathbf{D}[i]^H}_{\mathbf{W}_i^H}\mathbf{r}[i]. \tag{3.5.34}$$

Note here that the index i stands for the subcarrier. The zero forcing solution is obtained by simply setting the noise variance to zero in the MMSE solution. These linear techniques are however suboptimal as they suffer from noise enhancement. The best we could possibly do is to use ML detection which would require checking all possible transmit vectors for the vector closest[6] to the received vector. The complexity of optimal ML detection is of the order M^{M_T} where we assume that M is the order of the modulation. This is clearly infeasible for a large number of transmit antennas or large constellations. It is possible to use a suboptimal scheme called ordered successive interference cancelation (OSIC) which is analogous to the decision feedback

[6]The definition of the closest depends on the distribution of the noise. For example, for Gaussian noise, which is the commonly encountered, it is in terms of the Euclidean norm. Similarly, for Laplacian noise, it would be in terms of the taxi cab norm (l_1 norm).

equalizer. It has also been shown that OSIC can achieve the ergodic capacity of MIMO-SU[7] channels in a fast fading environment [13]. In slow fading, the outage capacity is still achieved but with diagonal encoding with the D-BLAST architecture as opposed to the V-BLAST we use here [57]. SIC uses a divide and conquer strategy as it peels off layers that have already been detected and linearly equalizes the remaining layers thus mitigating noise enhancement. It does introduce error propagation in the process which is why detection has to be ordered from the strongest stream to the weakest stream. By detecting the stream with the highest post equalization SNR at each stage, the errors propagated into the other streams by peeling off this stream are minimized. While ML equalization can offer M_R order diversity and linear equalizers only provide $M_R - M_T + 1$ order diversity, OSIC gives a compromise between the two. With OSIC, the streams detected from the first to the last achieve diversity orders of $M_R - M_T + 1, M_R - M_T + 2, \cdots M_R$.

For OSIC with MMSE equalization, the detection order is determined by the mean squared error. The stream with the smallest mean squared error is detected first using a MMSE equalizer to suppress the other streams. The contribution of this detected stream is peeled off (canceled) from the received signal. Then the stream with the smallest MSE of the remaining streams is detected using an MMSE equalizer to suppress the other streams (excluding the one that has been peeled off). This detection of the streams proceeds iteratively with fewer streams to suppress with each iteration as more have been peeled off, until only one stream is left and is simply detected. The p^{th} iteration involves MMSE equalization

$$\hat{s}_m[i] = \mathbf{e}_m^T(\mathbf{D}_p^H[i]\mathbf{D}_p^H[i] + \sigma^2\mathbf{I}_{M_T})^{-1}\mathbf{D}_p^H[i]\mathbf{z}_p \qquad (3.5.35)$$

[7]For MIMO-MAC, the capacity region is attained by ordered successive user cancelation (OSUC) which is essentially identical.

where $\mathbf{e}_m$ is the m^{th} column of the M_T identity matrix. The stream to be detected in this iteration is determined by

$$m = \arg\min_m [(\mathbf{D}_p^H[i]\mathbf{D}_p[i] + \sigma^2\mathbf{I}_{M_T})^{-1}]_{m,m} \tag{3.5.36}$$

The vector of received samples to be processed is updated by peeling off the detected stream as

$$\mathbf{z}_{p+1}[i] = \mathbf{z}_p[i] - \mathbf{d}_m[i]\hat{s}_m[i] \tag{3.5.37}$$

where $\mathbf{d}_m[i]$ is the m^{th} column of $\mathbf{D}_p[i]$ and $\mathbf{D}_p[i]$ is updated as

$$\mathbf{D}_{p+1}[i] = \mathbf{D}_p[i]\begin{bmatrix}\mathbf{e}_1 & \cdots & \mathbf{e}_{m-1} & \mathbf{0} & \mathbf{e}_{m+1} & \cdots & \mathbf{e}_{M_T}\end{bmatrix} \tag{3.5.38}$$

The OSIC is initialized with $\mathbf{z}_1[i] = \mathbf{r}[i]$ and $\mathbf{D}_1[i] = \mathbf{D}[i]$.

SIC is generally carried out through a QR decomposition. Let the QR decomposition of $\mathbf{D}[i]$ be $\mathbf{D}[i] = \mathbf{QR}$ where $\mathbf{Q}$ is a unitary matrix and $\mathbf{R}$ is an upper triangular matrix. Then for zero forcing SIC

$$\tilde{\mathbf{s}}[i] = \mathbf{Q}^H\mathbf{r}[i] = \mathbf{R}\mathbf{s}[i] + \tilde{\mathbf{v}}[i] \tag{3.5.39}$$

where $\tilde{\mathbf{v}}[i] = \mathbf{Q}^H\mathbf{v}[i]$. The properties of the noise are unchanged because of the unitary nature of $\mathbf{Q}$. Because of the upper triangular structure of $\mathbf{R}$ [10]

$$\tilde{s}_k = r_{kk}s_k + \sum_{i=k+1}^{M_T} r_{ki}s_i + \tilde{v}_k \tag{3.5.40}$$

Thus the k^{th} stream is free of interference from streams $1 \cdots k-1$. This allows stream M_T to be detected as $\hat{s}_{M_T} = \frac{\tilde{s}_{M_T}}{r_{M_T M_T}}$. Assuming correct detection of s_{M_T}, the detection of the s_{M_T-1} can proceed using the estimate of s_{M_T} and so on until the whole vector $\mathbf{s}$ is detected.

Although considerably less than ML detection, the computational complexity of BLAST can still be large [58]. This complexity can be significantly

reduced [59, 60] by settling for a suboptimal ordering that still manages to salvage most of the gain. Detection in SIC proceeds from stream M_T to stream 1 and to minimize error propagation, it is desired that SIC be ordered from the stream with the largest SNR to the one with the least. By changing the order of the columns of $\mathbf{D}[i]$ and the elements of $\mathbf{s}[i]$ we can get a new QR factorization which corresponds to a different order of detection. In a QR factorization, the SNR of each layer is given by $|r_{kk}|^2$. To minimize error propagation, it is necessary to maximize $|r_{kk}|$ for $k = M_T, M_T - 1, \cdots 1$. However QR factorization proceeds from r_{11} to $r_{M_T M_T}$. Thus to avoid having to search exhaustively for the QR factorization which maximizes r_{kk} for $k = M_T, M_T - 1, \cdots 1$, we can while performing Gram Schmidt orthogonalization try to minimize r_{kk} for $k = 1, 2 \cdots M_T$ [59]. The logic is that by choosing a smaller $|r_{kk}|$ for initial values of k, we will get larger values in subsequent steps. This procedure is the sorted QRD and it does not always result in the optimal order but the losses are small. For OSIC with MMSE equalization, the augmented channel matrix is defined $\Theta = \begin{bmatrix} \mathbf{D}^T[i] & \sigma_v \mathbf{I} \end{bmatrix}^T$, then the output of the MMSE filter is given by the form for the zero forcing filter

$$\hat{\mathbf{s}} = (\Theta^H \Theta)^{-1} \Theta^H \underbrace{\begin{bmatrix} \mathbf{r}^T[i] & \mathbf{0}^T \end{bmatrix}^T}_{\underline{\mathbf{r}}[i]} \tag{3.5.41}$$

BLAST is accomplished by a QR factorization on this augmented channel matrix.

$$\Theta = \underbrace{\begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix}}_{\underline{\mathbf{Q}}} \underline{\mathbf{R}} \tag{3.5.42}$$

From this, we see that $\mathbf{Q}_2 = \sigma_v \underline{\mathbf{R}}^{-1}$ and $\mathbf{Q}_1 \underline{\mathbf{R}} = \mathbf{D}[i]$ [59].

$$\tilde{\mathbf{s}}[i] = \underline{\mathbf{Q}}^H \underline{\mathbf{r}}[i] = \mathbf{Q}_1^H (\mathbf{Q}_1 \underline{\mathbf{R}} \mathbf{s}[i] + \mathbf{v}[i]) = \underline{\mathbf{R}} \mathbf{s}[i] - \underbrace{\sigma_v \mathbf{Q}_2^H \mathbf{s}[i]}_{\text{interference}} + \underbrace{\mathbf{Q}_1^H \mathbf{v}[i]}_{\text{noise}} \tag{3.5.43}$$

The second term is a lower triangular matrix that constitutes interference that is not removed. We have that [59]

$$\hat{\mathbf{s}}[i] = (\mathbf{D}^H[i]\mathbf{D}[i] + \sigma^2\mathbf{I})^{-1}\mathbf{D}^H[i]\mathbf{r}[i] \tag{3.5.44}$$

The detection MSE is then given by [59]

$$E\big((\hat{\mathbf{s}}[i] - \mathbf{s}[i])(\hat{\mathbf{s}}[i] - \mathbf{s}[i])^H\big) = \sigma_v^2(\mathbf{\Theta}^H[i]\mathbf{\Theta}[i])^{-1} = \sigma_v^2\underline{\mathbf{R}}^{-1}\underline{\mathbf{R}}^{-H} \tag{3.5.45}$$

We see that small eigenvalues of $\mathbf{\Theta}^H[i]\mathbf{\Theta}[i]$ will lead to large errors due to noise amplification. The estimation error for the k^{th} stream is then $\sigma_v^2/|r_{kk}|^2$. So we see that the optimal ordering for MMSE-SIC is the same as that for ZF-SIC, i.e. we would like to choose an ordering that would maximize $|r_{kk}|$ in each detection step. Thus the sorted QR decomposition algorithm for ZF-SIC can again be applied here.

3.5.5 Simulation Results

Using $\rho = 0.2$, we simulated a MIMO OFDM system with $M_T = 2$ transmit antennas, $M_R = 3$ receive antennas, and channel length $L = 7$ using 4 QAM modulation. A cyclic prefix order $L_{cp} = 2$ was taken and a shortening equalizer with $L_s = 9$ was used to shorten the channel to a desired length $L_d = L_{cp} = 2$. OFDM symbol length $N = 256$ was used. The channel taps were simulated as independent and identically distributed Rayleigh Fading. The BER and normalized estimation mean squared error simulation results are provided in figures 3.9 and 3.10 respectively for MMSE equalization and MMSE equalization with OSIC.

Figure 3.9 shows that the performance improvement decreases with the iterations. The largest improvement is shown by the first iteration over the initial BER and subsequent iterations give diminishing gains. Further, it can be seen that most of the gains are achieved in the first three iterations so that

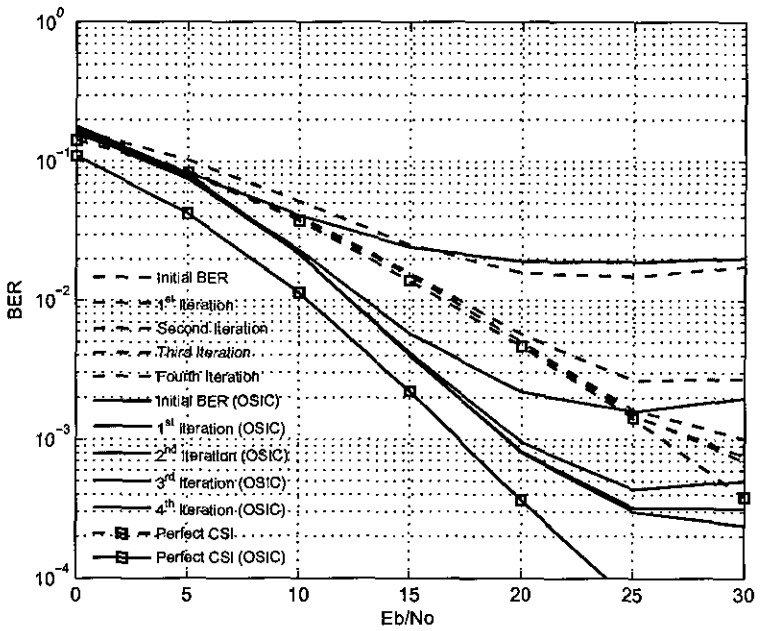

Figure 3.9. Bit error rate is plotted against Eb/No for various iterations with MMSE equalization (dashed lines) and MMSE equalization with ordered successive interference cancellation (solid lines). Results for perfect CSI are marked □.

the performance of our iterative approach is substantially better than that of superimposed training alone. It can be seen that the iterative approach provides BER performance that is nearly two orders of magnitude better than a non iterative approach incase of an MMSE-OSIC receiver at 25-30 dB SNR. For a MMSE receiver, the iterative approach provides an order of magnitude better performance than a non iterative approach at 25-30 dB SNR. The performance gap with the perfect CSI is seen to be about 2 dB in figure 3.9.

It can be seen that the initial estimates exhibit an early error floor. This error floor occurs because of the superimposed training structure where the channel estimate suffers interference from the residual unknown data transmission. Notice that an error floor is also observed for our iterative scheme at $Eb/No > 25$dB. However, this error floor is less than that due to the non iterative approach by orders of magnitude. We remind the reader

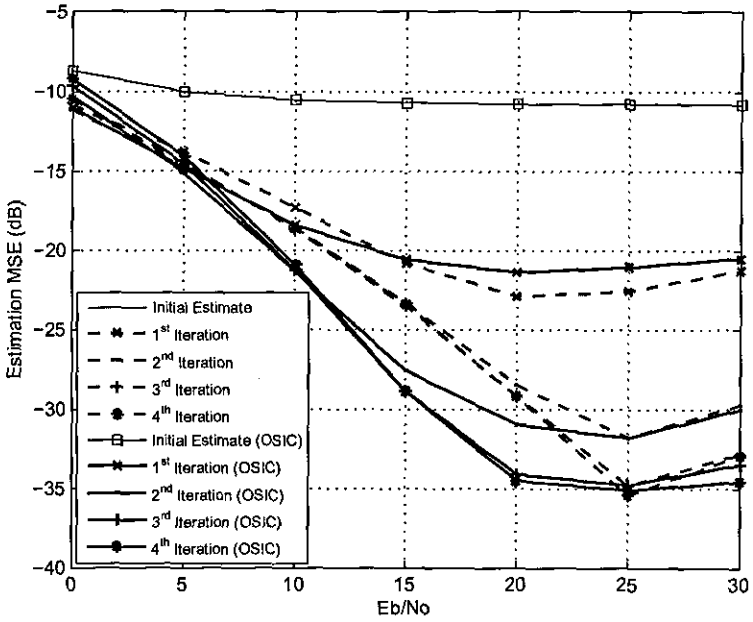

Figure 3.10. The normalized estimation mean square error is plotted against Eb/No for MMSE equalization (dashed lines) and MMSE equalization with ordered successive interference cancellation (solid lines). Results for the initial superimposed training estimates are marked □.

that an OFDM system is under consideration and uncoded OFDM typically suffers from an error floor even with perfect CSI because of carriers that get nulled because of the channel frequency response. This is why OFDM is used in conjunction with a code and an interleaver and the code bits are scattered over multiple coherence bandwidths so that the transmit signal may be recovered even if it sees channel fades.

3.6 Conclusion

Typically a BER of 10^{-3} is sufficient for voice communications with such an error rate imperceptible to the human ear. Data applications may require an even better link. We can see from the simulation results that superimposed training results in an early error floor. It cannot even be used for voice communications, let alone data communications, regardless of the

SNR. This error floor precludes the direct application of superimposed training to communications. Although OFDM is typically employed with forward error correction, we see from the simulation results that with the proposed iterative approach reasonable performance is achieved in the SNR band of interest allowing for voice communication even without introducing codes. For applications requiring better performance, error control coding can always be introduced. A superimposed training based iterative scheme for wireless systems is proposed for multicarrier communications. Our simulation results indicate that an iterative approach to channel estimation equalization and decoding in conjunction with superimposed training can provide overwhelming performance gains including lower estimation MSE and BER.

Chapter 4

SUPERIMPOSED TRAINING FOR SINGLE CARRIER SYSTEMS

4.1 Introduction

The focus of this chapter is on superimposed training based, iterative semiblind channel estimation techniques for single carrier systems with emphasis on doubly selective channels. By now the reader is probably familiar with the concepts behind superimposed training. While a brief overview of existing literature on superimposed training is provided, the contribution of this chapter will be to the study of a semiblind superimposed training based scheme for a coded system in doubly selective channels. An excellent overview of space time coding for wireless communications as well as error correction coding is provided in [61].

Superimposed training exploits the fact that the training and data pass through the same channel. It thus induces cyclostationarity in the transmission to allow for channel estimation through a simple sample averaging operation at the receiver. As a consequence of this cyclostationarity, the training adds coherently at the receiver while the unknown data does not. The channel information is extracted from the first order statistics of the received signal. This makes it possible to estimate the channel without

allocation of any explicit time slots for the transmission of training. Superimposed training thus allows for channel estimation at the expense of power as opposed to time multiplexed training which estimates the channel at the expense of bandwidth. This could have important ramifications when the channel is fast varying and also when operating in the bandwidth limited regime. Meanwhile, it retains the computational simplicity that is a striking feature of time multiplexed training. An overview of pilot based channel estimation is provided in [62].

Superimposed training may be considered as a compromise between blind and time multiplexed training. Blind channel estimation techniques have been motivated because they are known to be bandwidth efficient as they allow estimation of the channel from only the channel output and statistical (or other) information of the input. Blind techniques could be especially suitable for fast varying channels as a very significant portion of the throughput would be consumed in training the receiver otherwise for such channels. Trellis based blind algorithms like the per survivor processing approach in [63], or the expanded trellis approach of [64] are fast to converge but computationally very intensive. On the other hand, inverse filtering and channel estimation algorithms are computationally less complex but slower converging. These blind techniques were initially based on higher order statistics because phase information of the channel could not be extracted from the second order statistics. This made them slow to converge and computationally complex. Later, it was realized that the second order statistics of a cyclostationary sequence contain the phase information and thus faster blind techniques for channel estimation based on second order statistics were developed for fractionally spaced systems [65–67]. Nevertheless, blind algorithms are still computationally demanding. They also suffer from a phase ambiguity in that the channel can be identified up to an unknown phase offset. This requires the use of differential encoding for communications, which incurs a three dB SNR penalty. Like blind techniques, superimposed

training is also bandwidth efficient as it does not require the allocation of any explicit time slots for training. Furthermore, it relies on the first order statistics and could therefore converge to a solution much faster and with far less complexity. In terms of power, only a small fraction of the transmit power is allocated to the training so the loss is less than the three dB incurred by the blind schemes. Thus superimposed training offers both the bandwidth efficiency of blind techniques and the computational elegance of multiplexed training.

4.2 Superimposed Training in Single Carrier

Superimposed training for communication over a multipath wireless environment was first proposed in [68] and investigated by [69] where the second order statistics of the received signal were used to obtain a channel estimate. [70] was the first to realize that a reliable channel estimate could actually be formed from the first order statistics of the received signal. *Superimposed training based on first order statistics was discussed in detail* in [53]. [71] proposed a simple method of dealing with mean value uncertainty in the received sequence. Soon enough, it was realized that superimposed training alone could not provide sufficient performance levels because of an early noise floor resulting from interference from the data. As a solution to this problem, [72] proposed a data dependant superimposed training scheme for cyclic prefixed transmission where the superimposed pilot is actually modified by adding a data dependant component to the periodic pilot. This data dependant component is meant to cancel out the interference of the unknown data to the training. This scheme thus calculates the interference from the data and cancels it out at the transmitter. The training and the data are thus orthogonal at the receiver. Now a better channel estimate can be obtained. Once the channel estimate is used to detect the data, it will not detect the original data stream but instead will detect the data stream

modified by the data dependant component of the training. Detection of the original data stream then proceeds by minimum distance decoding (MDD) over the set of possible transmit symbols and a lower complexity symbol by symbol detection algorithm is also provided. An alternative approach is to use an iterative semiblind superimposed training algorithm. MIMO channel estimation using superimposed training was studied in [73].

Superimposed training for doubly selective channels has been previously studied in [74–77]. A data dependant superimposed training algorithm was proposed for superimposed training in complex exponential basis expansion model doubly selective channels by [75], while a partially data dependant scheme was proposed in [76] for discrete prolate spheroidal basis expansion model doubly selective channels. Training power allocation was discussed in [74] while superimposed training for MIMO doubly selective channel estimation was discussed in [77].

It has become clear that superimposed training suffers from an early error floor due to interference from data transmission into the channel estimation which severely limits its applicability. We study superimposed training based semiblind channel estimation for doubly selective channels where the detected data is also introduced into the channel estimation. We consider a coded system and an iterative equalization and decoding receiver so that the sequence at the decoder output is used for channel re-estimation. In each iteration, the channel is re-estimated and the equalizer is updated for the next iteration based on these estimates.

4.3 Doubly Selective Channels

The demand for higher data rates and mobility has shifted the focus of channel estimation research towards doubly selective channels. These channels result from a combination of multipath, high data rates, mobility, and carrier frequency offsets and they display both time and frequency selectivity. They

can be modeled using basis expansion models (BEM) [12]. The BEM models the channel variations within a block with basis functions which remain unchanged from block to block. The variations between blocks are modeled with the basis coefficients which are held constant in a block but vary from block to block. Various choices for bases have been proposed including the polynomial, wavelet, Fourier and the Slepian [78,79]. The BEM provides a tractable description of the channel by sampling it in both the delay and Doppler dimensions. Because of the time and frequency selectivity, these channels offer both delay and Doppler diversity. But this also makes them difficult to acquire.

The time varying impulse response of a channel including the effects of the transmit receive filter pair and the doubly selective propagation effects of the environment is denoted by $h(t;\tau)$. The Fourier transform of such an impulse response is denoted by $H(f;\tau)$. The delay spread τ_{max} and the Doppler spread f_{max} are defined such that $|H(f;\tau)| \approx 0$ for $\tau > \tau_{max}$ or for $f > f_{max}$. The delay and Doppler spread are assumed to be bounded such that the delay Doppler spread factor $2f_{max}\tau_{max}$ is less than one. Otherwise there would be more unknowns than equations and the estimation problem will not be well posed. The baseband channel input output relation is

$$y(n) = \sum_{l=0}^{L} h(n,l)x(n-l) + v(n) \tag{4.3.1}$$

where $h(n,l)$ is the sampled channel response of the channel at time n to an impulse applied at time $n-l$, $y(n)$ is the channel output, $x(n-l)$ is the channel input at time $n-l$ and $v(n)$ is the additive white Gaussian noise. We note that the channel impulse response depends explicitly on the time index. The parameter l indicates frequency selectivity, while the parameter n indicates time selectivity. If $h(n,l) = 0 \ \ \forall \ \ l \neq 0$, the channel is purely time selective and if $h(n,l) = h(l) \ \ \forall \ \ n$, the channel is purely frequency selective. The response $h(n,l)$ includes any transmit receive filtering along

with the physical channel. This channel is concisely described in the delay Doppler dimensions as a finite set of complex exponential BEM coefficients as

$$h(n,l) = \sum_{q=-Q/2}^{Q/2} h_q(\lfloor n/N \rfloor, l) e^{j\frac{2\pi nq}{N}} \tag{4.3.2}$$

where $\frac{Q}{2}$ is the discrete Doppler spread. The discrete parameters are related to their continuous counterparts as $L = \lfloor \tau_{max}/T_s \rfloor$ and $Q = 2\lceil f_{max} N T_s \rceil$ where T_s is the sampling period at the receiver. The BEM coefficients (h_q) are used to model the channel variations between blocks and they are held constant over a block. The basis vectors are unchanged across blocks and are responsible for modeling the time variations within the blocks. The BEM coefficients may be viewed as two dimensional samples of the channel, i.e. samples along the Doppler and the delay dimensions. We make the assumption that the channel is underspread, i.e. the delay - Doppler spread product is less than $\frac{1}{2}$ for identifiability. We also note that there are $(L+1)(Q+1)$ paths between the input and the output which is the order of the diversity offered by the doubly selective channel. The $N \times (N+L)$ dimensional convolution matrix for a linear time varying channel has the structure

$$\mathbf{H} = \begin{bmatrix} h(0,L) & \cdots & h(0,0) & 0 & \cdots & 0 & 0 \\ 0 & h(1,L) & \cdots & h(1,0) & 0 & \cdots & 0 \\ & & & \ddots & & & \\ 0 & \cdots & 0 & & h(N-1,L) & \cdots & h(N-1,0) \end{bmatrix} \tag{4.3.3}$$

By substituting in equation (4.3.2) this becomes

$$\sum_{q=-\frac{Q}{2}}^{\frac{Q}{2}} \begin{bmatrix} h_q(0)e^{\frac{j2\pi q(0)}{N}} & \cdots & h_q(L)e^{\frac{j2\pi q(0)}{N}} & 0 & \cdots & 0 \\ & & \ddots & & & \\ 0 & \cdots & 0 & h_q(0)e^{\frac{j2\pi q(N-1)}{N}} & \cdots & h_q(L)e^{\frac{j2\pi q(N-1)}{N}} \end{bmatrix} \tag{4.3.4}$$

where $\omega_q = \frac{2\pi q}{N}$. This is the product of two matrices

$$\mathbf{H} = \sum_{q=-Q/2}^{Q/2} \begin{bmatrix} e^{\frac{j2\pi q(0)}{N}} & & \\ & \ddots & \\ & & e^{\frac{j2\pi q(N-1)}{N}} \end{bmatrix} \begin{bmatrix} h_q(0) & \cdots & h_q(L) & & \\ & \ddots & & & \\ & & h_q(0) & \cdots & h_q(L) \end{bmatrix} \tag{4.3.5}$$

This is written as

$$\mathbf{H} = \sum_{q=-Q/2}^{Q/2} \mathbf{D}_q \mathbf{H}_q \tag{4.3.6}$$

where $\mathbf{D}_q = \text{diag}\begin{bmatrix} 1 & e^{j\frac{2\pi q}{N}} & \cdots & e^{j\frac{2\pi q(N-1)}{N}} \end{bmatrix}$ is an $N \times N$ diagonal matrix and $\mathbf{H}_q$ is a $N \times (N+L)$ dimensional Toeplitz matrix. The linear time varying (LTV) convolution matrix $\mathbf{H}$ has a banded structure, but no longer possess the Toeplitz property characteristic of linear time invariant (LTI) convolution matrices. Neither does it have the diagonal structure associated with time selective channels.

A LTV receiver is required to process a signal received over a LTV channel. The design of LTV receivers according to the zero forcing criterion and the MMSE criterion has been studied in [80]. This work was later extended to the design of LTV decision feedback equalizers in [81]. [82] and [83] have also studied LTV equalizer design for LTV channels. The design of LTV time domain channel shorteners has been studied in [84] where LTV time domain equalizers were designed according to the MMSE principle in order to eliminate the time selectivity of the channel and shorten the channel impulse response length. By using a LTV receiver to transform the time varying channel to a time invariant channel, the orthogonality between subcarriers is restored and OFDM may be employed. Similarly, LTV per tone equalizers have been studied in [85] for use in OFDM for doubly selective channels.

It is also possible to to use an array of time invariant receivers to recover the signal transmitted over a time varying channel. This approach considers a time variant SIMO system as a time invariant MIMO system [86] and then

blind equalization techniques that have been developed for time invariant MIMO channels can be employed. The difficulty with this approach is the large antenna array that is required at the receiver to blindly detect the signal. The number of receive antennas required for this scheme is directly related to the number of basis functions which can be large for rapidly fading channels. Such large antenna arrays can be unwieldy. Similarly differential modulation over doubly selective channels has been studied by [87] but there is significant complexity as well as a 3 dB loss. With our iterative semiblind superimposed training based approach we can establish the link with far less complexity.

4.4 SISO Systems

4.4.1 Signal Model

The matrix vector model for block transmission over a doubly selective channel is

$$\mathbf{y} = \mathbf{H}\tilde{\mathbf{x}} + \mathbf{v} \tag{4.4.1}$$

where $\mathbf{v} \in \mathbf{C}^{N\times 1}$ is the vector of additive white Gaussian noise, $\mathbf{y} \in \mathbf{C}^{N\times 1}$ is the vector of received samples, and $\tilde{\mathbf{x}} \in \mathbf{C}^{(N+L)\times 1}$ is the vector of transmit symbols. The doubly selective channel $\mathbf{H}$ is as defined in (4.3.6). For implicit training, the pilot is superimposed on the unknown data as

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{c} \tag{4.4.2}$$

where $\mathbf{x}$ contains the unknown data while $\mathbf{c}$ contains the pilot signal. We can further define the training to information power ratio

$$TIR = \frac{\rho}{(1-\rho)} = \frac{\|\mathbf{c}\|_2^2}{E(\|\mathbf{x}\|_2^2)} \tag{4.4.3}$$

as the fraction of the total power allocated to training. Allocation of more power to the pilot results in better channel estimation but it also leaves very little power and increased interference for data transmission. However, allocation of insufficient power to channel estimation adversely affects the detection of the unknown data through poor channel estimates. We delineated this tradeoff for OFDM in the last chapter with the help of simulation results.

4.4.2 Channel Estimation

The problem we study is the semiblind estimation of doubly selective channels using superimposed training. Again, the material in this section is similar to the that in 3.4.1 with the essential difference that the basis functions for modeling the doubly selective channel are now introduced. While only one sample averaging operation was required to estimate the channel in 3.4.1, $Q+1$ sample averaging operations are now required to estimate all the BEM coefficients. By substituting equation (4.3.6) in (4.4.1), the N received samples in a block are given by

$$\mathbf{y} = \sum_{q=-Q/2}^{Q/2} \mathbf{D}_q \mathbf{H}_q \tilde{\mathbf{x}} + \mathbf{v} \tag{4.4.4}$$

$$\mathbf{y} = \left[\mathbf{D}_{-Q/2}\mathbf{H}_{-Q/2}\tilde{\mathbf{x}} \quad \cdots \quad \mathbf{D}_{Q/2}\mathbf{H}_{Q/2}\tilde{\mathbf{x}} \right] + \mathbf{v} \tag{4.4.5}$$

Because of the commutativity between a Toeplitz matrix and a vector, this may be written as

$$\mathbf{y} = \left[\mathbf{D}_{-Q/2}\tilde{\mathbf{X}}\mathbf{h}_{-Q/2} \quad \cdots \quad \mathbf{D}_{Q/2}\tilde{\mathbf{X}}\mathbf{h}_{Q/2} \right] + \mathbf{v} \tag{4.4.6}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{D}_{-Q/2} & \cdots & \mathbf{D}_{Q/2} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}} & & \\ & \ddots & \\ & & \tilde{\mathbf{X}} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{h}_{-Q/2} \\ \vdots \\ \mathbf{h}_{Q/2} \end{bmatrix}}_{\mathbf{h}} + \mathbf{v} \tag{4.4.7}$$

$$\mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{D}_{-Q/2} & \cdots & \mathbf{D}_{Q/2} \end{bmatrix}}_{\mathbf{D}} (\mathbf{I}_{Q+1} \otimes \tilde{\mathbf{X}})\mathbf{h} + \mathbf{v} \tag{4.4.8}$$

where $\tilde{\mathbf{X}}$ is a Toeplitz matrix constructed from $\tilde{\mathbf{x}}$ and $\mathbf{h}$ is a $(Q+1)(L+1)$ dimensional vector containing the BEM coefficients of the channel. The received signal $y(n)$ is not stationary owing to the deterministic training $\mathbf{c}$. It is, however, cyclostationary because of the periodicity of $\mathbf{c}$, so that a sequence of vector symbols comprising of P received symbols each is stationary and a sample average can be formed. We now consider this sample average

$$\mathbf{z} = \frac{1}{R}\mathbf{\Phi}\mathbf{D}^H\mathbf{y} \tag{4.4.9}$$

where $\mathbf{\Phi} = \mathbf{I}_{(Q+1)} \otimes \check{\mathbf{\Phi}}$ has dimensions $(Q+1)P \times (Q+1)N$, $\check{\mathbf{\Phi}} = \mathbf{1}_R^T \otimes \mathbf{I}_P = \begin{bmatrix} \mathbf{I}_P & \cdots & \mathbf{I}_P \end{bmatrix}$ is a $P \times N$ dimensional matrix that takes a sequence of P sample averages from $R = \frac{N}{P}$ samples, and $\mathbf{z} \in \mathbf{C}^{(Q+1)P \times 1}$. By using the fact that $\mathbf{D}_q = \mathbf{D}_{q,R}^P \otimes \mathbf{D}_{q,P}$ where $\mathbf{D}_{q,R}^P = \text{diag}(\begin{bmatrix} e^{\frac{j2\pi qP(0)}{N}} & e^{\frac{j2\pi qP(1)}{N}} & \cdots & e^{\frac{j2\pi qP(R-1)}{N}} \end{bmatrix})$ and $\mathbf{D}_{q,P} = \text{diag}(\begin{bmatrix} e^{\frac{j2\pi q(0)}{N}} & e^{\frac{j2\pi q1}{N}} & \cdots & e^{\frac{j2\pi q(P-1)}{N}} \end{bmatrix})$ i.e. that the sample averaging operation will add points on the DFT grid and will thus sum to zero suppressing the BEM coefficients for all all q but $q = 0$. To estimate the BEM coefficients for any other q, we must first multiply the received samples with $\mathbf{D}_q^H$ to translate these coefficients to $q = 0$ and then perform the sample averaging operation. Thus we will obtain

$$\mathbf{z} = \underbrace{(\mathbf{I}_{(Q+1)} \otimes \mathbf{\Upsilon})}_{\mathbf{\Delta}}\mathbf{h} + \frac{1}{R}\underbrace{\mathbf{\Phi}\mathbf{D}^H\mathbf{v}}_{\tilde{\mathbf{v}}} \tag{4.4.10}$$

where $\mathbf{\Delta}$ has dimensions $(Q+1)P \times (Q+1)(L+1)$, $\mathbf{\Upsilon} = \check{\mathbf{\Phi}}\mathbf{C} \in \mathbf{C}^{P\times(L+1)}$ (where $\mathbf{C}$ is a Toeplitz matrix constructed from $\mathbf{c}$). This allows us to form a least squares estimate of the channel as

$$\hat{\mathbf{h}} = (\mathbf{\Delta}^H\mathbf{\Delta})^{-1}\mathbf{\Delta}^H\mathbf{z} \tag{4.4.11}$$

The solution of this least squares problem requires that $\mathbf{\Delta}$ be full rank which in turn requires that $\mathbf{\Upsilon}$ be full rank, implying that $P \geq L+1$ for identifiability. As [88] pointed out, an important caveat here is that this result assumes that the additive noise was zero mean - a situation that often arises from linearization about an unknown bias point that is filtered out as it contains no information. However since we wish to exploit the first order statistics for channel estimation, we cannot just filter out the bias point and it is necessary to include the effect of an unknown bias in the noise term. Assuming now that the additive noise is actually $\mathbf{v} + m\mathbf{1}_N$ then equation (4.4.8) becomes

$$\mathbf{y} = \mathbf{D}(\mathbf{I}_{(Q+1)} \otimes \tilde{\mathbf{X}})\mathbf{h} + \mathbf{v} + m\mathbf{1}_N \tag{4.4.12}$$

The BEM coefficient estimate is corrupted by the unknown mean m. As pointed out by [71], the DFT matrix can be used overcome this difficulty in channel estimation by noting that only the first term is affected by the unknown mean. A DFT operation is introduced after the sample averaging and the first DFT term is discarded as it is corrupted by the unknown mean while the remaining terms are used for channel estimation. Let $\check{\mathbf{F}}$ denote a $(P-1) \times P$ matrix obtained from the $P \times P$ DFT matrix by discarding its first row. Then $\mathbf{\Phi} = \mathbf{I}_{Q+1} \otimes \check{\mathbf{\Phi}}$ is replaced by $\mathbf{\Phi} = \mathbf{I}_{Q+1} \otimes \check{\mathbf{F}}\check{\mathbf{\Phi}}$ and

$$\mathbf{z} = \underbrace{(\mathbf{I}_{Q+1} \otimes \check{\mathbf{F}}\mathbf{\Upsilon})}_{\mathbf{\Delta}'}\mathbf{h} + \frac{1}{R}\underbrace{\mathbf{\Phi}'\mathbf{D}^H\mathbf{v}}_{\mathbf{v}'} \tag{4.4.13}$$

and the channel estimate is formed as

$$\hat{\mathbf{h}} = (\boldsymbol{\Delta}'^H \boldsymbol{\Delta}')^{-1} \boldsymbol{\Delta}'^H \mathbf{z} \tag{4.4.14}$$

and the unknown DC offset m is estimated as

$$\hat{m} = \mathbf{1}_N^T (\mathbf{y} - \mathbf{D}(\mathbf{I}_{(Q+1)} \otimes \mathbf{C})\hat{\mathbf{h}})/N \tag{4.4.15}$$

where $\mathbf{C}$ is a Toeplitz matrix formed from the training sequence. The condition for unique identifiability of the channel is now $P \geq L+2$. From equation (4.4.14), it is evident that the estimator performance depends on the choice of the training sequence through $\boldsymbol{\Upsilon} = \check{\boldsymbol{\Phi}}\mathbf{C}$. This least squares estimate depends on the condition number of $\boldsymbol{\Upsilon}^H\boldsymbol{\Upsilon}$. For a condition number of unity, $\boldsymbol{\Upsilon}^H\boldsymbol{\Upsilon} = e\mathbf{I}$ where e is a scalar. Thus the training sequences should be chosen such that the normalized circulant training matrix is unitary [53]. It was shown in [53] that for any period P, such training sequences can always be constructed along with the desirable property of a peak to average power ratio of unity.

The initial channel estimate thus formed is used to design an equalizer and begin turbo equalization. In every iteration of the turbo equalization, the channel is re-estimated when soft estimates of all transmit symbols have been generated by the decoder. In doing so, we incorporate the unknown data into the channel estimation process as well. Let $\hat{\mathbf{x}}$ denote the soft estimates of the transmitted signal generated by the decoder, then we have

$$\mathbf{y} = \sum_{q=-Q/2}^{Q/2} \mathbf{D}_q \hat{\mathbf{X}} \mathbf{h}_q + \mathbf{v} \tag{4.4.16}$$

where $\hat{\mathbf{X}}$ is a L^{th} order Toeplitz matrix. We thus have

$$\mathbf{x} = \boldsymbol{\Phi}\mathbf{h} + \mathbf{v} \tag{4.4.17}$$

where $\Phi = \left[\mathbf{D}_{-Q/2}\hat{\mathbf{X}} \quad \cdots \quad \mathbf{D}_{Q/2}\hat{\mathbf{X}}\right]$ and $\mathbf{h} = \left[\mathbf{h}_{-Q/2}^T \quad \cdots \quad \mathbf{h}_{Q/2}^T\right]^T$. The least squares channel estimate using the detected data as training is then

$$\hat{\mathbf{h}} = (\Phi^H\Phi)^{-1}\Phi^H\mathbf{y} \tag{4.4.18}$$

4.4.3 Turbo Equalization

Although discovered four decades later, turbo equalization and turbo coding are directly inspired by the first of three lessons from Shannon's ground breaking paper, i.e. to never prematurely discard information that may be useful in making a decision until after all decisions related to that information have been made. It is for this reason that their decoding algorithms do not take any hard decisions till the decoding is complete. These probabilistic decoding algorithms operate directly on the Bayesian principle whereby the extrinsic information produced by one component serves as a priori information to the other. This a priori is weighted by the constituent block against its observations. If the observations are unreliable, the algorithm will rely more on the a priori and vice versa. Incase of zero a priori information, the algorithm relies entirely on the observations. Turbo equalization is such an iterative equalization and decoding algorithm used in place of the computationally infeasible, albeit optimal joint maximum likelihood equalization and decoding. It involves a detector that uses a priori symbol estimates to generate a soft replica of the interference. The detector mitigates intersymbol interference by subtracting soft replicas of the interference from the received signal and then filters the signal to further minimize the effect of any residual interference. We employ a MMSE detector and a rate 1/3 convolutional code with generator polynomial $(145, 133, 7)_8$ for forward error correction. A sequential MMSE equalizer of order L_{eq} and a MAP decoder exchange soft information iteratively and the turbo principle is used. An estimate of the training contribution to the received signal is removed to form the effective

signal for processing in each iteration.

$$\dot{\mathbf{y}} = \mathbf{y} - \hat{\mathbf{H}}\mathbf{c} - \hat{m}\mathbf{1}_N \tag{4.4.19}$$

The linear MMSE estimate of x_n is then obtained as [9]

$$\hat{x}_n = \mathfrak{h}_n^H \mathrm{Cov}(\dot{\mathbf{y}}_n, \dot{\mathbf{y}}_n)^{-1}(\dot{\mathbf{y}}_n - \hat{\mathbf{H}}_n\bar{\mathbf{x}}_n + \bar{x}_n\mathfrak{h}_n) \tag{4.4.20}$$

where $\dot{\mathbf{y}}_n = \left[\dot{y}(n - \lfloor Leq/2 \rfloor) \quad \cdots \quad \dot{y}(n + \lfloor Leq/2 \rfloor)\right]$, $\hat{\mathbf{H}}_n$ is the $(Leq + 1) \times (Leq + L + 1)$ time variant convolution matrix reconstructed from the estimates of the BEM coefficients and

$$\mathrm{Cov}(\dot{\mathbf{y}}_n, \dot{\mathbf{y}}_n) = (\sigma_v^2\mathbf{I}_N + \hat{\mathbf{H}}_n\mathbf{Q}_n\hat{\mathbf{H}}_n^H + (1 - q_n)\mathfrak{h}_n\mathfrak{h}_n^H) \tag{4.4.21}$$

where $q_n = \mathrm{Cov}(x_n, x_n)$ and

$$\bar{\mathbf{x}}_n = [\bar{x}^*_{n-\lfloor (L+Leq)/2 \rfloor} \quad \cdots \quad \bar{x}^*_{n+\lceil (L+Leq)/2 \rceil}]^H \tag{4.4.22}$$

$$\mathbf{Q}_n = \mathrm{Diag}([q_{n-\lfloor (L+Leq)/2 \rfloor} \quad \cdots \quad q_{n+\lceil (L+Leq)/2 \rceil}]) \tag{4.4.23}$$

$$\mathfrak{h}_n = \hat{\mathbf{H}}_n[\mathbf{0}_{1\times(\lfloor (L+Leq)/2 \rfloor)} \quad 1 \quad \mathbf{0}_{1\times(\lceil (L+Leq)/2 \rceil)}]^H \tag{4.4.24}$$

Note that since we use only extrinsic information in equations (4.4.20) and (4.4.21), the prior $\bar{x}_n$ is not involved in the computation of $\hat{x}_n$. Once the estimates have been obtained, they are converted to log likelihood ratios for exchange with the decoder [9]

$$L_E(x_n) = 2(\hat{x}_n\mu_{n,+1})/\sigma^2_{n,+1} \tag{4.4.25}$$

The statistics used in (4.4.25) are

$$\mu_{n,u} = \mathbf{g}_n^H(E(\dot{\mathbf{y}}_n^H | x_n = x) - \hat{\mathbf{H}}_n\bar{\mathbf{x}}_n + \bar{x}_n\mathfrak{h}_n) = x\mathbf{g}_n^H\mathfrak{h}_n \tag{4.4.26}$$

$$\sigma_{n,u}^2 = \mathbf{g}_n^H \mathrm{Cov}(\dot{\mathbf{y}}_n, \dot{\mathbf{y}}_n | x_n = x)\mathbf{g}_n = \mathbf{g}_n^H \mathfrak{h}_n (1 - \mathfrak{h}_n^H \mathbf{g}_n) \qquad (4.4.27)$$

and $\mathbf{g}_n \triangleq \mathrm{Cov}(\dot{\mathbf{y}}_n, \dot{\mathbf{y}}_n)^{-1}\mathfrak{h}_n$ is the equalizer. The effect of the equalizer should be to convert the channel into an equivalent AWGN channel on which the decoder can operate.

Turbo equalization and space time turbo equalization for MIMO systems can provide significant performance improvements over a non iterative receiver if coding is employed. However, exploiting these gains requires the receiver to have good CSI. The channel has to be estimated and is characterized by estimation errors which reduce the ability of the equalizer to remove the correlation introduced in the received signal by the channel. These correlations cause the turbo equalization algorithm to stall and it fails to make any significant progress over the iterations. As a solution to this problem, the channel estimation is introduced in the iterative equalization and decoding loop so that the channel is re-estimated after the completion of each iteration. This allows the channel estimate to be refined along with the iterations.

MAP Decoder

To minimize the probability of symbol error, the decoder has to maximize the a posteriori probabilities for all bits.

$$x_k = \arg\max_{x_k} p(x_k|\mathbf{y}) \qquad (4.4.28)$$

The *a posteriori probability* is

$$p(x_k = x|\mathbf{y}) = \sum_{\mathbf{x}:x_k=x} p(\mathbf{x}|\mathbf{y}) = \sum_{\mathbf{x}:x_k=x} \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \qquad (4.4.29)$$

The maximum a posteriori principle can be used for equalization as well as decoding. MAP equalization will minimize the probability of symbol error at the equalizer output. Although we use a MMSE equalizer and a MAP

decoder, we use the the channel input and output variables x and y in this section to highlight the applicability of this algorithm to the equalization task. Furthermore, turbo equalization was initially envisaged with a MAP equalizer. At this point we can give a reason why separate equalization and decoding fare so poorly compared to iterative equalization and decoding. Assuming a block length N, a code rate r and BPSK modulation, we see that there would be 2^N possible input and output sequences of the encoder but $2^{N/r}$ possible input and output sequences of the equalizer. When the equalizer processes the received signal according to (4.4.29), it assumes that all of the $2^{N/r}$ sequences are equally likely whereas the the code in series with the channel has imposed the constraint that only 2^N of these $2^{N/r}$ sequences are actually possible. Thus the ideal approach should be to modify (4.4.29) so that only those sequences are considered that are possible. By only considering the sequences that are actually possible, much of the performance loss would be recovered by a separate equalization and decoding scheme. *However, this would require exhaustive searches and be infeasible* as we explain shortly. Another reason for the performance gain is the use of soft information. The performance of the turbo equalization with MMSE and MAP equalizers converge for large block lengths and the performance approaches that of the code in an equivalent AWGN channel. The performance can be further improved beyond this by the use of a precoder.

In concept, decoding a sequence by maximizing the a posteriori probabilities is quite simple. However, unlike the Viterbi algorithm which only retains the survivors and discards the other paths, the summations here are over all possible allowed paths. This means that the complexity of a direct approach to the problem using equations (4.4.28) and (4.4.29) is exponential in the length of the sequence to decode and hence infeasible. However, many of the operations performed in such a direct approach are redundant and simplification is possible by noting that these operations are repeated time and again. We thus explain an efficient algorithm for computing a pos-

teriori probabilities known as the BCJR algorithm known after its inventors, Bahl, Cocke, Jelenik and Raviv [89].

The APP decoder exploits the structure of the channel / encoder and the Markov property, i.e. its trellis. The Markov property states that random variables E, F, G form a Markov chain, denoted by $E \leftrightarrow F \leftrightarrow G$, if we can write

$$p(E, G|F) = p(E|F)p(G|, F) \tag{4.4.30}$$

Two equivalent conditions are $p(E|F, G) = p(E|F)$ and $p(G|E, F) = p(G|F)$. For the equalization or decoding problem, we denote the states in the trellis at time k by a variable s_k, then it is possible to write

$$p(x_k = x|\mathbf{y}) = \sum_{(l',l)\in\mathfrak{B}^x} p(s_{k-1} = l', s_k = l|\mathbf{y}) \tag{4.4.31}$$

where $\mathfrak{B}^x$ denotes the set of all state transitions $s_{k-1} = l' \Rightarrow s_k = l$ which correspond to a symbol $x_k = x$.

$$p(s_{k-1} = l', s_k = l|\mathbf{y}_1^\tau) = p(\mathbf{y}_{k+1}^\tau|s_k = l)p(s_{k-1} = l', s_k = l, \mathbf{y}_1^{k-1}, y_k)/p(\mathbf{y}) \tag{4.4.32}$$

Again by application of conditional probabilities

$$p(s_{k-1} = l', s_k = l, \mathbf{y}_1^{k-1}, y_k) = p(s_k = l, y_k|s_{k-1} = l', \mathbf{y}_1^{k-1})p(s_{k-1} = l', \mathbf{y}_1^{k-1}) \tag{4.4.33}$$

Thus by substituting equation (4.4.33) in (4.4.32) and by the Markov property, we have

$$p(s_{k-1} = l', s_k = l|\mathbf{y}_1^\tau) = \tag{4.4.34}$$

$$\underbrace{p(\mathbf{y}_{k+1}^\tau|s_k = l)}_{\beta_k(l)}\underbrace{p(s_k = l, y_k|s_{k-1} = l')}_{\gamma_k(l,l')}\underbrace{p(s_{k-1} = l', \mathbf{y}_1^{k-1})}_{\alpha_{k-1}(l')}/p(\mathbf{y}) \tag{4.4.35}$$

Where the α's and β's are forward and backward messages, the reason for which will become clear shortly. Computation of the these proceeds itera-

tively

$$\alpha_{k-1}(l') = p(\mathbf{y}_1^{k-1}, s_{k-1} = l') = \sum_{l \in \mathfrak{B}^x} p(\mathbf{y}_1^{k-2}, y_{k-1}, s_{k-1} = l', s_{k-2} = l) \tag{4.4.36}$$

$$\alpha_{k-1}(l') = \sum_{(l,l') \in \mathfrak{B}^x} \underbrace{p(y_{k-1}, s_{k-1} = l' | s_{k-2} = l)}_{\gamma_{k-1}(l,l')} \underbrace{p(\mathbf{y}_1^{k-2}, s_{k-2} = l)}_{\alpha_{k-2}(l)} \tag{4.4.37}$$

Thus α_k is computed from α_{k-1} and so on. The computation of α's begins at the left of the trellis and proceeds in the forward direction toward the right. For computation of the the β's,

$$\beta_k(l) = \sum_{(l,l') \in \mathfrak{B}^x} p(\mathbf{y}_{k+1}^{\tau}, s_k = l, s_{k+1} = l')/p(s_k = l) \tag{4.4.38}$$

$$\beta_k(l) = \sum_{(l,l') \in \mathfrak{B}^x} \frac{p(\mathbf{y}_{k+2}^{\tau} | y_{k+1}, s_k = l, s_{k+1} = l') p(y_{k+1}, s_k = l, s_{k+1} = l')}{p(s_k = l)} \tag{4.4.39}$$

which can be simplified by applying the Markov property as

$$\beta_k(l) = \sum_{(l,l') \in \mathfrak{B}^x} \underbrace{p(\mathbf{y}_{k+2}^{\tau} | s_{k+1} = l')}_{\beta_{k+1}(l')} \underbrace{p(y_{k+1}, s_{k+1} = l' | s_k = l)}_{\gamma_{k+1}(l,l')} \tag{4.4.40}$$

Thus β_k is calculated from β_{k+1} and so on. The computation of the β's begins at the right end of the trellis and proceeds in the backward direction towards the left hence the name backward messages. The APP decoder is also called a forward backward algorithm for obvious reasons. This leaves the computation of the γ's.

$$\gamma_k(l,l') = p(y_{k+1}, s_{k+1} = l' | s_k = l) \tag{4.4.41}$$

$$= p(y_{k+1} | s_{k+1} = l', s_k = l) p(s_{k+1} = l' | s_k = l) \tag{4.4.42}$$

where $p(s_{k+1} = l' | s_k = l)$ is the state transition probability determined by the structure of the trellis and $p(y_{k+1} | s_{k+1} = l', s_k = l)$ is the probability of the output given a state transition. The variables associated with a state

transition are known from the trellis, hence

$$p(y_{k+1}|s_{k+1}=l', s_k=l) = p(y_{k+1}|x_{k+1}) = \frac{1}{(2\pi\sigma_v^2)^{n/2}} e^{\frac{-\|y_{k+1}-x_{k+1}\|^2}{2\sigma_v^2}} \tag{4.4.43}$$

It is more convenient to work with log likelihood ratios rather than probabilities. For binary constellations, while there are two probabilities, they are constrained to add up to one so that there is only one degree of freedom. Thus only one variable is needed to convey the reliability information, i.e. the log likelihood ratio

$$L(x_k|\mathbf{y}) = \frac{p(x_k=0|\mathbf{y})}{p(x_k=1|\mathbf{y})} = \frac{\sum_{(l,l')\in\mathcal{B}^0} \alpha_{k-1}(l')\gamma_k(l,l')\beta_k(l)}{\sum_{(l,l')\in\mathcal{B}^1} \alpha_{k-1}(l')\gamma_k(l,l')\beta_k(l)} \tag{4.4.44}$$

The decision rule on the variable thus becomes

$$\hat{x}_k = \begin{cases} 0, & L(x_k|\mathbf{y}) \geq 0; \\ 1, & L(x_k|\mathbf{y}) < 0. \end{cases} \tag{4.4.45}$$

The APP algorithm just described is actually the message passing algorithm when applied to a trellis. The message passing algorithm for probabilistic decoding is applied to codes on graphs for decoding. On cycle free graphs, i.e. trees, the message passing algorithm is finite and exact while it is iterative and approximate on graphs with cycles. The performance loss in this case depends on the length of the smallest cycle and is smaller for larger cycles. The message passing algorithm essentially solves global constraints by factoring them into small local constraints and then proceeds to solve the problem by passing messages between these local constraints. It thus solves a computationally difficult problem by solving a multitude of easier subproblems. In the example of our convolutional code or the communications channel, the branches of the trellis represent the variables and the

nodes represent constraints. The message passing algorithm is also known as the sum product algorithm because it involves taking the sum of products[1].

4.4.4 Simulation Results

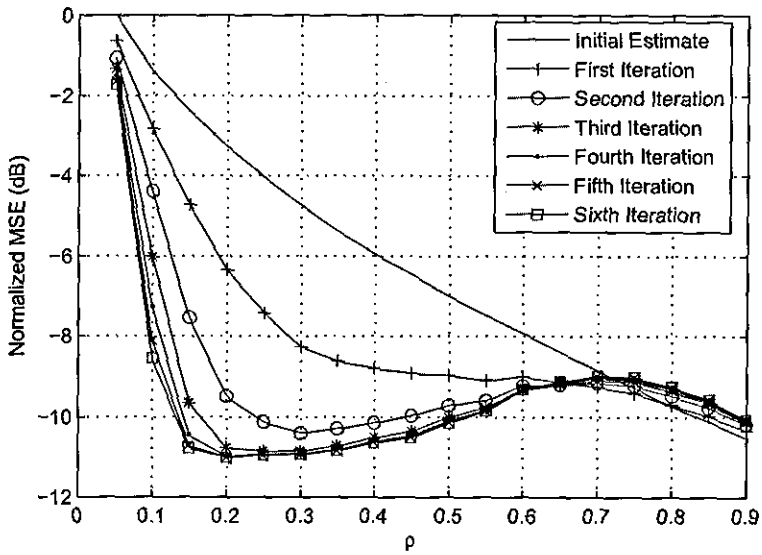

Figure 4.1. The normalized estimation mean square error as a function of the power allocation ρ at Eb/No = 5 dB for various iterations.

The joint channel estimation and symbol detection algorithm was simulated for doubly selective channels with four taps and a Doppler spread of 0.01 normalized to the symbol rate. Block lengths of $N = 2400$ were used, i.e. $Q/2 = 24$. Apart from the Doppler spread, no prior channel knowledge was assumed. The simulation results in figure 4.4 indicate that superimposed training used in conjunction with turbo equalization has the potential to considerably improve performance over a simple superimposed training based system. In all our simulation results, the bit energy to noise power spectral density definition used includes the training power in the bit energy.

The normalized mean squared error (channel estimation) is plotted against the power allocation ρ in figure 4.1 at an Eb/No of 5 dB. We note that for the initial estimate (and similarly a non iterative scheme), estimation

[1]In statistical inference, the message passing algorithm is called belief propagation

MSE decreases monotonically with increasing allocation of power to training. However, for our iterative approach, the estimation MSE decreases

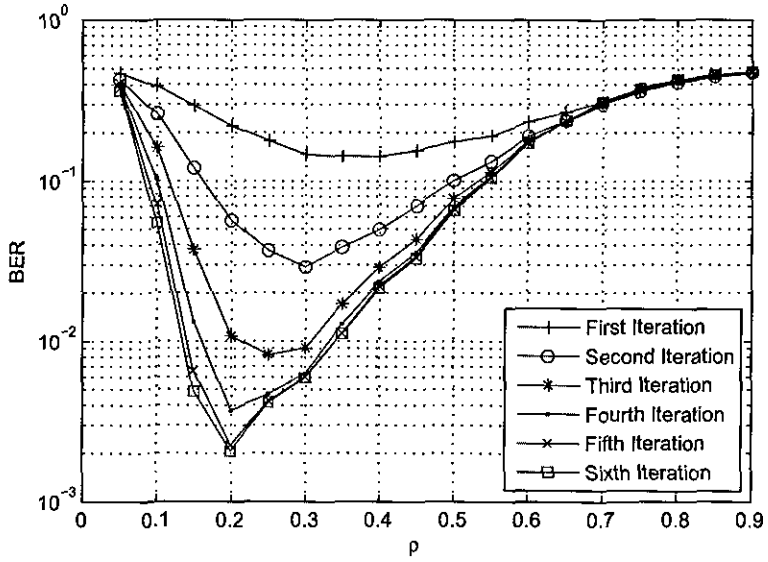

Figure 4.2. The BER as a function of ρ at Eb/No = 5 dB for various iterations.

steeply till $\rho = 0.15$ and remains steady till $\rho = 0.45$. This steep improvement in the estimation MSE at $\rho \leq 0.15$ can be attributed to an improving initial estimate for beginning the turbo equalization. While the share of power to the unknown data (which is also used in estimation in subsequent iterations) is decreasing, it is easily compensated for by the more accurate initialization of equalizer in the turbo equalization. The steady region from $0.15 \leq \rho \leq 0.45$ is where the any gains due to an improved initial estimate are offset by the decrease in power allocation to the unknown data. It is evident from these results that the iterative scheme proposed here provides the comparable estimation performance at $\rho = 0.2$ to the estimation performance of a simple superimposed scheme at $\rho = 0.9$. This improvement can translate to a large gain in the BER as 80% of the transmit power may now be allocated to data transmission as opposed to 10% while achieving comparable channel estimation performance. This improvement in BER is

evident in figure 4.2 where the BER is plotted against ρ. The BER is minimum when the estimation MSE is at its lowest and the power allocated to the unknown data is maximum.

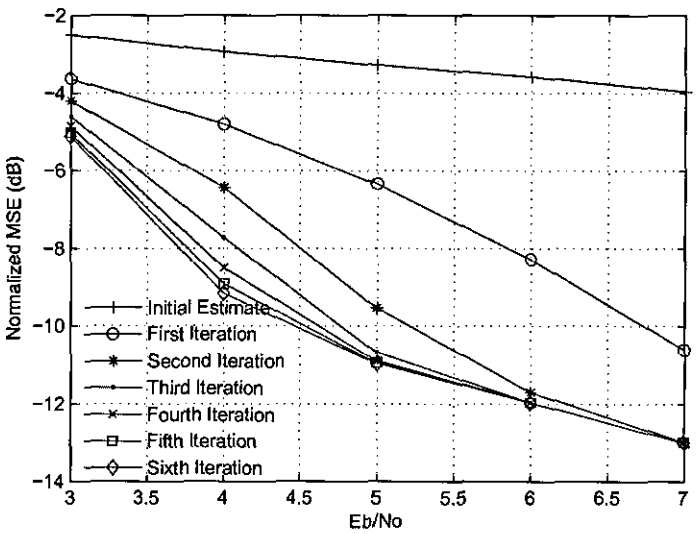

Figure 4.3. The normalized MSE performance plotted against Eb/No for various iterations ($\rho = 0.2$).

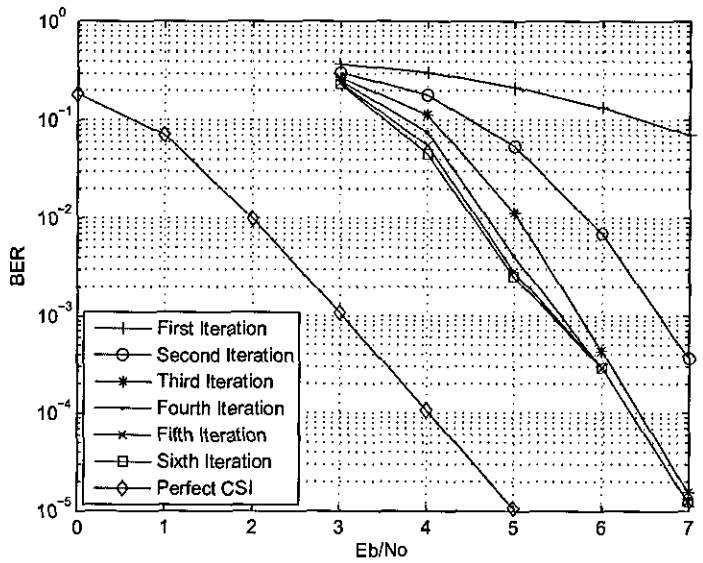

Figure 4.4. The bit error rate curves for turbo equalization with perfect CSI and the iterations of turbo equalization with iterative channel estimation.

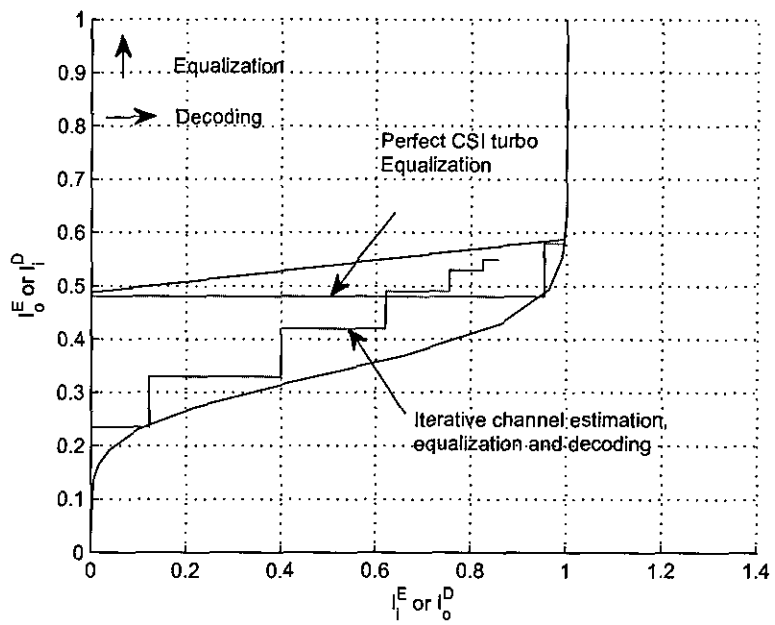

Figure 4.5. The EXIT chart for the iterative channel estimation and turbo equalization process at 4 dB signal to noise ratio

The normalized channel estimation mean squared error is plotted for various iterations against Eb/No with $\rho = 0.2$ in figure 4.3. We see that the normalized mean squared error is reduced by approx. 8 dB (at 6 dB Eb/No) compared to a simple non iterative scheme. The BER results in figure 4.4 indicate approx. 3 dB gain at a BER of 10^{-1}. The results in figure 4.4 indicate that there is a 2 dB penalty as compared to the situation with perfect CSI. An EXIT chart [8] is shown in figure 4.5 for turbo equalization with perfect CSI and our iterative scheme at Eb/No = 4 dB. Note that while the turbo equalization converges in one iteration due to the wide open gap between the transfer characteristics, our algorithm requires several iterations to make its way. Because of the unavailability of true CSI, the trajectory is more confined than the equalizer transfer characteristics and cannot achieve the mutual information bounds set by it in the equalization steps. Nevertheless, with the channel updates in every iteration, the mutual information at the equalizer output increases along with the iterations.

4.4.5 Conclusion

A superimposed training based iterative scheme for wireless systems with doubly selective channels has been proposed. Our simulation results indicate that by including channel estimation in the iterative equalization and decoding loop, the turbo equalization algorithm can provide significant performance gains including a lower estimation MSE and BER. Also, no error floor is observed in our simulation results.

4.5 MIMO Systems

The capacity of a practical communications system is given by

$$C = W\log_2(1 + \frac{\text{SNR}}{\Gamma}) \quad \text{bits/s} \tag{4.5.1}$$

where SNR $= \frac{P}{WN_0}$, P is the transmit power, W is the bandwidth, N_0 is the one sided noise power spectral density and Γ is called the gap[2] and is a function of the desired BER performance, system margin and coding gain [13]. It is obvious from this equation that increase in capacity requires simultaneous increase in both power and bandwidth. Failure to increase the bandwidth will only push the system into the bandwidth limited regime where a three dB power increase will only yield one additional bit per dimension increase in spectral efficiency. Similarly only increasing the bandwidth will push the system into the power limited regime. The capacity will saturate at $\frac{P}{N_0}\log_2 e$ despite the availability of infinite bandwidth.

High data rates could theoretically be achieved with SISO systems employing sufficient bandwidth and power, but such a system cannot be practically feasible [90]. For one thing, bandwidth is a scarce resource in terrestrial applications. The use of a large bandwidth also entails a range penalty[3]. On the other hand, the transmission power is limited by a host of factors including the battery life, biohazard issues, difficulty in building cheap linear amplifiers over the larger output range, etc. The capacity of SISO systems is therefore capped. MISO and SIMO systems do not fare much better either. MISO systems can provide a diversity gain so their capacity is bounded by the AWGN channel capacity. SIMO systems offer an array gain which boosts the receiver SNR to provide a further improvement over MISO systems and their capacity increases logarithmically in the number of receive antennas employed - hardly a feasible method of increasing capacity given that there will be a diminishing return because of the concavity of the logarithm function.

MIMO provides a practical means of boosting the system capacity. Like SIMO and MISO, MIMO provides a diversity gain. Like SIMO, MIMO also offers an array gain. However, there is a spatial multiplexing gain [91]

[2] also referred to as SNR_{norm} sometimes

[3] With the total transmit power capped, the signal to noise ratio per hertz will decrease, reducing the range.

which is unique to MIMO systems. Spatial multiplexing makes it possible to transmit multiple streams through the channel so that the capacity increases linearly in the minimum number of antennas at either end. The diversity gain allows any SNR improvement in the link to be transformed into an improvement in the link reliability while the spatial multiplexing gain exploits any SNR improvement to enhance the data rate supported by the link. While both these gains may be exploited simultaneously, there is a tradeoff between the diversity gain and the spatial multiplexing gain that a space time coding scheme can exploit. This tradeoff has been studied in [92]. MIMO systems can also pave the way for a large gains at the network level including trunking gain where the static sector frequency assignment can be relaxed providing a larger pool of channels where needed. With a little coordination between base stations, it can help reduce co-channel interference (CCI). It also makes resource reuse within cell[4] (RWC) possible at the cost of introducing multiple access interference.

We consider a simple MIMO system where M_T data streams are encoded, interleaved, and transmitted once the pilots have been superimposed onto them at low power. The channels between the transmit and receive antennas are doubly selective complex exponential BEM channels and no channel knowledge is assumed at the receiver. The signal received at an array of M_R antennas is then processed by the channel estimator and space time turbo equalizer to recover the M_T data streams.

4.5.1 Channel Estimation

The material in this section is similar to that in section 4.4.2 with the essential difference that multiple transmit and receive antennas are now included in the mathematical framework. We can describe the channel between the

[4]also known as space division multiple access (SDMA)

κ^{th} transmitter and the ν^{th} receiver using a BEM as follows [78]

$$h^{(\nu,\kappa)}(k;l) = \sum_{q=-Q/2}^{Q/2} h_{ql}^{(\nu,\kappa)} e^{j2\pi qk/N} \tag{4.5.2}$$

In block form, we can write [78]

$$\mathbf{H}^{(\nu,\kappa)} = \sum_{q=-Q/2}^{Q/2} \mathbf{D}_q \mathbf{H}_q^{(\nu,\kappa)} \tag{4.5.3}$$

where $\mathbf{D}_q := \text{diag}(\begin{bmatrix} 1 & e^{j2\pi q/N} & \dots & e^{j2\pi q(N-1)/N} \end{bmatrix})$ and $\mathbf{H}_q^{(\nu,\kappa)}$ is a as described previously. The convolution matrix $\mathbf{H}^{(\nu,\kappa)}$ no longer has the Toeplitz structure characteristic of an LTI system. $Q/2$ is the discrete Doppler spread. We assume that the channel is underspread, i.e. the delay Doppler spread product is less than $\frac{1}{2}$ for identifiability.

We assume a zero prefixed block based transmission scheme so the observations may be expressed using a matrix vector model where the zero prefix of length L equal to the maximum order of the channel ensures IBI free transmission. We assume the length of the block is $N+L$. The observations at the ν^{th} receive antenna are given by

$$\mathbf{y}_\nu = \sum_{\kappa=1}^{M_T} \mathbf{H}^{(\nu,\kappa)} \tilde{\mathbf{x}}_\kappa + \mathbf{v}_\nu \tag{4.5.4}$$

By substituting equation (4.5.3) into (4.5.4) and by commutativity of the product of a Toeplitz matrix and a vector, we may write

$$\mathbf{y}_\nu = \sum_{q=-Q/2}^{Q/2} \mathbf{D}_q \sum_{\kappa=1}^{M_T} \tilde{\mathbf{X}}_\kappa \mathbf{h}_q^{(\nu,\kappa)} + \mathbf{v}_\nu \tag{4.5.5}$$

where $\tilde{\mathbf{X}}_\kappa$ is a Toeplitz matrix constructed from the κ^{th} users transmit sequence $\tilde{\mathbf{x}}_\kappa$ and $\mathbf{h}_q^{(\nu,\kappa)} = \begin{bmatrix} h_q^{(\nu,\kappa)}(0) & \dots & h_q^{(\nu,\kappa)}(L) \end{bmatrix}$. Similarly let $\mathbf{h}_q^{\nu} =$

$\left[\mathbf{h}_q^{(\nu,1)T} \quad \cdots \quad \mathbf{h}_q^{(\nu,M_T)T}\right]^T$, then we have

$$\mathbf{y}_\nu = \left[\mathbf{D}_{-Q/2} \quad \cdots \quad \mathbf{D}_{Q/2}\right](\mathbf{I}_{Q+1} \otimes \tilde{\mathbf{X}})\mathbf{h}_\nu + \mathbf{v}_\nu \tag{4.5.6}$$

where $\tilde{\mathbf{X}} = \left[\tilde{\mathbf{X}}_1 \quad \cdots \quad \tilde{\mathbf{X}}_{M_T}\right]$ and $\mathrm{h}_\nu = \left[\mathbf{h}_{-Q/2}^\nu \quad \cdots \quad \mathbf{h}_{Q/2}^\nu\right]$. The transmit signal $\tilde{\mathbf{X}}$ comprises the known periodic training $\mathbf{C}$ added at low power to the unknown data $\mathbf{X}$. Thus we have $\tilde{\mathbf{X}} = \mathbf{C} + \mathbf{X}$ where

$$\mathbf{C} = \left[\mathbf{C}_1 \quad \cdots \quad \mathbf{C}_{M_T}\right] \tag{4.5.7}$$

and $\mathbf{X}$ is defined similarly. The training sequence $\mathbf{c}_\kappa$ is known and periodic to allow for estimation of the channel through first order statistics. We assume the training period P is chosen such that $N/P = R$ is an integer. Denote the $P \times L+1$ toeplitz matrix formed from a period of the training sequence as $\boldsymbol{\Upsilon}_\kappa$, then we have

$$\mathbf{C} = \mathbf{1}_R \otimes \underbrace{\left[\boldsymbol{\Upsilon}_1 \quad \cdots \quad \boldsymbol{\Upsilon}_{M_T}\right]}_{\Upsilon} \tag{4.5.8}$$

where $\mathbf{1}_R$ is a $R \times 1$ vector of all ones. Stacking the received vectors at all receive antennas, we can write

$$\mathbf{y} = (\mathbf{I}_{M_R} \otimes \left[\mathbf{D}_{-Q/2} \quad \cdots \quad \mathbf{D}_{Q/2}\right])(\mathbf{I}_{(Q+1)M_R} \otimes \tilde{\mathbf{X}})\mathbf{h} + \mathbf{v} \tag{4.5.9}$$

where $\mathbf{h} = \left[\mathbf{h}_1^T \quad \cdots \quad \mathbf{h}_{M_R}^T\right]^T$ and $\mathbf{v}$ and $\mathbf{y}$ are defined similarly. Now let us define

$$\mathbf{z} = \frac{1}{R}\boldsymbol{\Omega}(\mathbf{I}_{M_R} \otimes \begin{bmatrix} \mathbf{D}_{-Q/2}^H \\ \vdots \\ \mathbf{D}_{Q/2}^H \end{bmatrix})\mathbf{y} \tag{4.5.10}$$

where $\mathbf{\Omega} = \mathbf{I}_{M_R(Q+1)} \otimes \mathbf{1}_R^T \otimes \mathbf{I}_P$ simply takes a sample average of the received sequence over a period P. Simplifying the right hand side gives equation (4.5.11).

$$\mathbf{z} = \frac{1}{R}\mathbf{\Omega}(\mathbf{I}_{M_R} \otimes \mathfrak{D}_C)\mathbf{h} + \underbrace{\frac{1}{R}(\mathbf{\Omega}(\mathbf{I}_{M_R} \otimes \mathfrak{D}_X)\mathbf{h} + \mathbf{\Omega}(\mathbf{I}_{M_R} \otimes \begin{bmatrix} \mathbf{D}_{-Q/2}^H \\ \vdots \\ \mathbf{D}_{Q/2}^H \end{bmatrix})\mathbf{v})}_{\tilde{\mathbf{v}}} \tag{4.5.11}$$

where

$$\mathfrak{D}_C = \begin{bmatrix} \mathbf{C} & \mathbf{D}_1\mathbf{C} & \cdots & \mathbf{D}_Q\mathbf{C} \\ \mathbf{D}_{-1}\mathbf{C} & \mathbf{C} & \cdots & \mathbf{D}_{Q-1}\mathbf{C} \\ \vdots & & \ddots & \\ \mathbf{D}_{-Q}\mathbf{C} & \cdots & \mathbf{D}_{-1}\mathbf{C} & \mathbf{C} \end{bmatrix} \tag{4.5.12}$$

$$\mathfrak{D}_X = \begin{bmatrix} \mathbf{X} & \mathbf{D}_1\mathbf{X} & \cdots & \mathbf{D}_Q\mathbf{X} \\ \mathbf{D}_{-1}\mathbf{X} & \mathbf{X} & \cdots & \mathbf{D}_{Q-1}\mathbf{X} \\ \vdots & & \ddots & \\ \mathbf{D}_{-Q}\mathbf{X} & \cdots & \mathbf{D}_{-1}\mathbf{X} & \mathbf{X} \end{bmatrix} \tag{4.5.13}$$

Substituting the expression for $\mathbf{C}$ and that for $\mathbf{\Omega}$, we have equation

$$\mathfrak{D}_C = \begin{bmatrix} (\mathbf{1}_R^T \otimes \mathbf{I}_P)(\mathbf{1}_R \otimes \mathbf{\Upsilon}) & \cdots & (\mathbf{1}_R^T \otimes \mathbf{I}_P)\mathbf{D}_Q(\mathbf{1}_R \otimes \mathbf{\Upsilon}) \\ (\mathbf{1}_R^T \otimes \mathbf{I}_P)\mathbf{D}_{-1}(\mathbf{1}_R \otimes \mathbf{\Upsilon}) & \cdots & (\mathbf{1}_R^T \otimes \mathbf{I}_P)\mathbf{D}_{Q-1}(\mathbf{1}_R \otimes \mathbf{\Upsilon}) \\ \vdots & \ddots & \vdots \\ (\mathbf{1}_R^T \otimes \mathbf{I}_P)\mathbf{D}_{-Q}(\mathbf{1}_R \otimes \mathbf{\Upsilon}) & \cdots & (\mathbf{1}_R^T \otimes \mathbf{I}_P)(\mathbf{1}_R \otimes \mathbf{\Upsilon}) \end{bmatrix} \tag{4.5.14}$$

But $(\mathbf{1}_R^T \otimes \mathbf{I}_P)(\mathbf{1}_R \otimes \mathbf{\Upsilon}) = R\mathbf{\Upsilon}$ and we may write $\mathbf{D}_q = \mathbf{D}_{q,R}^P \otimes \mathbf{D}_{q,P}$ where $\mathbf{D}_{q,R}^P = \text{diag}(\begin{bmatrix} e^{\frac{-j2\pi qP(0)}{N}} & e^{\frac{-j2\pi qP(1)}{N}} & \cdots & e^{\frac{-j2\pi qP(R-1)}{N}} \end{bmatrix})$ and $\mathbf{D}_{q,P} = \text{diag}(\begin{bmatrix} e^{\frac{-j2\pi q(0)}{N}} & e^{\frac{-j2\pi q1}{N}} & \cdots & e^{\frac{-j2\pi q(P-1)}{N}} \end{bmatrix})$ so that $(\mathbf{1}_R^T \otimes \mathbf{I}_P)\mathbf{D}_q(\mathbf{1}_R \otimes \mathbf{\Upsilon}) = \mathbf{1}_R^T\mathbf{D}_{q,R}^P\mathbf{1}_R \otimes \mathbf{D}_{q,P}\mathbf{\Upsilon} = 0$ as $\mathbf{1}_R^T\mathbf{D}_{q,R}^P\mathbf{1}_R = 0$ as $\mathbf{D}_{q,R}^P$ just contains the points

on the R^{th} order DFT grid. We thus have

$$\mathbf{z} = \underbrace{(\mathbf{I}_{M_R} \otimes \mathbf{I}_{Q+1} \otimes \boldsymbol{\Upsilon})}_{\boldsymbol{\Delta}} \mathbf{h} + \frac{1}{R}\tilde{\mathbf{v}} \tag{4.5.15}$$

where $\tilde{\mathbf{v}}$ is a random vector that contains the contribution of the noise and the unknown data. The sample averaging operation performed by $\boldsymbol{\Omega}$ on the noise and unknown data will increase the power by R but the scaling term $1/R$ will reduce it by $1/R^2$ so that there will be a linear decrease in the power of the effective noise term with increasing R. At large enough R, this noise term may become negligible. A least squares estimate of the channel BEM coefficients can then be formed as

$$\hat{\mathbf{h}} = (\boldsymbol{\Delta}^H \boldsymbol{\Delta})^{-1} \boldsymbol{\Delta}^H \mathbf{z} \tag{4.5.16}$$

This solution requires $\boldsymbol{\Delta}$ to be full column rank implying the condition $P \geq (L+1)M_T$ for identifiability. As [88] pointed out, an important caveat here is that this result assumes that the additive noise was zero mean, a situation that often arises from linearization about an unknown bias point which is filtered out as it contains no information. However since we wish to exploit the first order statistics for channel estimation, we cannot just filter out the bias point and it is necessary to include the effect of an unknown bias in the noise term [88]. Assuming now that the additive noise at the ν^{th} receive antenna has mean m_ν then

$$\mathbf{y} = (\mathbf{I}_{M_R} \otimes \mathbf{D})(\mathbf{I}_{(Q+1)M_R} \otimes \tilde{\mathbf{X}})\mathbf{h} + \mathbf{m} \otimes \mathbf{1}_N + \mathbf{v} \tag{4.5.17}$$

where $\mathbf{m} = \begin{bmatrix} m_1 & \cdots & m_{M_R} \end{bmatrix}^T$, and m_i is the mean of the noise at the i^{th} receive antenna. When the sample averaging operation is applied at the

receiver, this mean term gives

$$\frac{1}{R}\mathbf{\Omega}(\mathbf{I}_{M_R}\otimes\mathbf{D}^H)(\mathbf{m}\otimes\mathbf{1}_N)=\mathbf{\Omega}\begin{bmatrix}\mathbf{D}^H & & \\ & \ddots & \\ & & \mathbf{D}^H\end{bmatrix}\begin{bmatrix}m_1\mathbf{1}_N\\ \vdots\\ m_{M_R}\mathbf{1}_N\end{bmatrix} \tag{4.5.18}$$

$$\frac{1}{R}\mathbf{\Omega}(\mathbf{I}_{M_R}\otimes\mathbf{D}^H)(\mathbf{m}\otimes\mathbf{1}_N) = \mathbf{\Omega}\begin{bmatrix}\mathbf{D}^H & & \\ & \ddots & \\ & & \mathbf{D}^H\end{bmatrix}\begin{bmatrix}m_1\mathbf{1}_N\\ \vdots\\ m_{M_R}\mathbf{1}_N\end{bmatrix} \tag{4.5.19}$$

$$= \begin{bmatrix}\check{\mathbf{\Omega}}\mathbf{d}_{-Q/2}m_1\\ \vdots\\ \check{\mathbf{\Omega}}\mathbf{d}_{Q/2}m_1\\ \vdots\\ \check{\mathbf{\Omega}}\mathbf{d}_{-Q/2}m_{M_R}\\ \vdots\\ \check{\mathbf{\Omega}}\mathbf{d}_{Q/2}m_{M_R}\end{bmatrix} \tag{4.5.20}$$

But $\check{\mathbf{\Omega}}\mathbf{d}_q$ is just the sum of uniformly separated points on the DFT grid and is zero. Thus the estimates of the BEM coefficients for $q \neq 0$ are unaffected by the unknown DC offset. However, the estimates for $q = 0$ suffer from an amplitude ambiguity because of the DC offset. Again this is solved by a DFT operation after the sample averaging. The first term of the DFT is discarded because it is affected by the unknown mean. Let $\check{\mathbf{F}}$ denote the $(P-1)\times P$ matrix obtained by discarding the first row of a $P\times P$ DFT matrix. Then, $\mathbf{\Omega} = \mathbf{I}_{M_R(Q+1)}\otimes\check{\mathbf{F}}\check{\mathbf{\Omega}}$ and

$$\mathbf{z} = \underbrace{(\mathbf{I}_{M_R(Q+1)}\otimes\check{\mathbf{F}}\mathbf{\Upsilon})}_{\mathbf{\Delta}'}\mathbf{h} + \frac{1}{R}\tilde{\mathbf{v}} \tag{4.5.21}$$

Then the channel estimate may be formed as

$$\hat{\mathbf{h}} = (\boldsymbol{\Delta}'^H \boldsymbol{\Delta}')^{-1} \boldsymbol{\Delta}'^H \mathbf{z} \tag{4.5.22}$$

The unknown DC offset at the ν^{th} antenna can be estimated as

$$\hat{m}_\nu = \mathbf{1}_N^T (\mathbf{y}_\nu - \mathbf{D}(\mathbf{I}_{Q+1} \otimes \mathbf{C})\hat{\mathbf{h}}_\nu)/N \tag{4.5.23}$$

where $\mathbf{C}$ is the training. The condition for unique identifiability of the channels is now $P \geq (L+1)M_T + 1$. Note that the block diagonal structure of $\boldsymbol{\Delta}$ allows the various estimations to be separated. For example if the BEM coefficients from all transmitters to the ν^{th} receive antenna for the q^{th} basis were required, they could be obtained as

$$\hat{\mathbf{h}}_q^\nu = \frac{1}{R}(\boldsymbol{\Upsilon}^H \check{\mathbf{F}}^H \check{\mathbf{F}} \boldsymbol{\Upsilon})^{-1} (\mathbf{1}_R^T \otimes \mathbf{I}_P) \mathbf{D}_q^H \mathbf{y} \tag{4.5.24}$$

The estimator performance depends on the choice of the training sequence through $\boldsymbol{\Upsilon}$. This least squares estimate depends on the condition number of $\boldsymbol{\Upsilon}^H \boldsymbol{\Upsilon}$ and which implies that in addition to satisfying the discussed periodicity constraints, the training sequences for the transmitters must be chosen such that $\boldsymbol{\Upsilon}^H \boldsymbol{\Upsilon}$ is close to a scaled identity. A possible choice of training sequence for the κ^{th} transmitter could be $\mathbf{e}_{\kappa,\tilde{P}}^T \otimes \mathbf{e}_{1,M_T}^T$ where $\mathbf{e}_{\kappa,\tilde{P}}^T$ is the κ^{th} row of the $\tilde{P} \times \tilde{P}$ matrix and $\tilde{P} = P/M_T$.

4.5.2 Space Time Turbo Equalization

The idea behind space time turbo equalization is the same as that behind turbo codes or turbo equalization, i.e. probabilistic processing between the constituent blocks at the receiver so that no information is prematurely discarded. Turbo equalization can eliminate ISI, which is self interference lying in the temporal dimension in a SISO or SIMO system. With the addition of multiple transmitters, it becomes necessary to exploit the spatial dimension

to separate these streams with the help of their distinct spatial signatures. In the MIMO-SU system, the multiple antennas are possessed by one user at each end of the link. The source signal can be processed in different ways depending on the conditions and requirements. The source stream may be demultiplexed into M_T streams which are then encoded independently. This scheme, called horizontal encoding, is the basis for V-BLAST and it considerably simplifies receiver design [10]. It can achieve the capacity in fast fading environments but it can only provide a diversity order of at most M_R in slow fading environments because any given symbol is transmitted from only one antenna. It is depicted in figure 4.6. Alternatively, the source stream may be

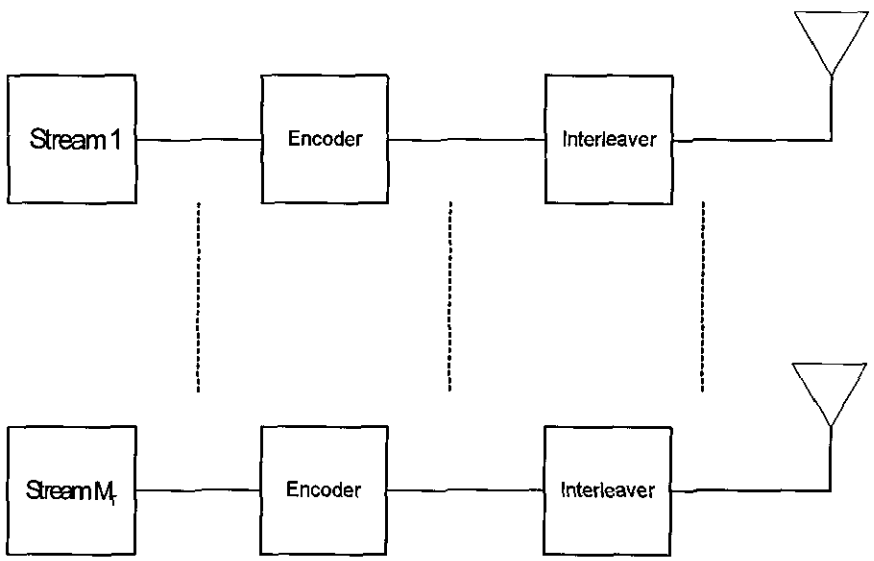

Figure 4.6. A block diagram of the transmitter for space time turbo equalization using horizontal encoding.

first encoded and interleaved and then demultiplexed for transmission from the M_T antennas. This scheme is called vertical encoding and it can achieve the capacity in a slow fading environment. However, the problem with this scheme is that it can be computationally infeasible because it requires joint equalization and decoding. It can be employed with much less complexity in an iterative equalization and decoding architecture like space time turbo equalization. A compromise between the two is the diagonal encoding used

by D-BLAST which uses horizontal encoding but employs a stream rotator immediately before transmission so that all antennas are used in the transmission of any stream [10]. The diagonal encoding structure then allows for a relatively simple receiver that can achieve the capacity of a slow fading channel. However, it does result in an initial wasted triangular block during startup. The space time turbo equalization architecture can employ any of these encoding structures for spatial multiplexing[5]. In the following, we use horizontal encoding.

To facilitate space time turbo equalization [93] the streams are independently encoded and interleaved before transmission. The signals at the receiver are iteratively processed by a single MIMO MMSE detector and a SISO decoder corresponding to each stream with the operations of de-interleaving and interleaving performed as these blocks exchange soft information as depicted in figure 4.7. The MIMO MMSE detector mitigates ISI and multiple stream interference (MSI) by replicating them from the interferer's log likelihood ratios (LLR) that have been fed back from the decoders. It then tries to remove the residual interference by filtering in order to minimize the mean square error between the filter output and the desired user's coded signal. The detector thus generates estimates of the user's coded data. The decoding for each of the users is performed independently by individual decoders operating on the extrinsic information output from the MIMO MMSE detector after it is de-interleaved. The extrinsic information from the decoder's output is then interleaved and used by the MIMO MMSE detector to repeat iterations of the process.

We define the vector $\dot{\mathbf{y}}$ by rearranging the received symbols as

$$\dot{\mathbf{y}} = \mathbf{P}(\mathbf{y} - \hat{\mathbf{H}}\mathbf{c} - \hat{\mathbf{m}} \otimes \mathbf{1}_N) \tag{4.5.25}$$

[5] Just as it can use different transmitter architectures for space time diversity coding.

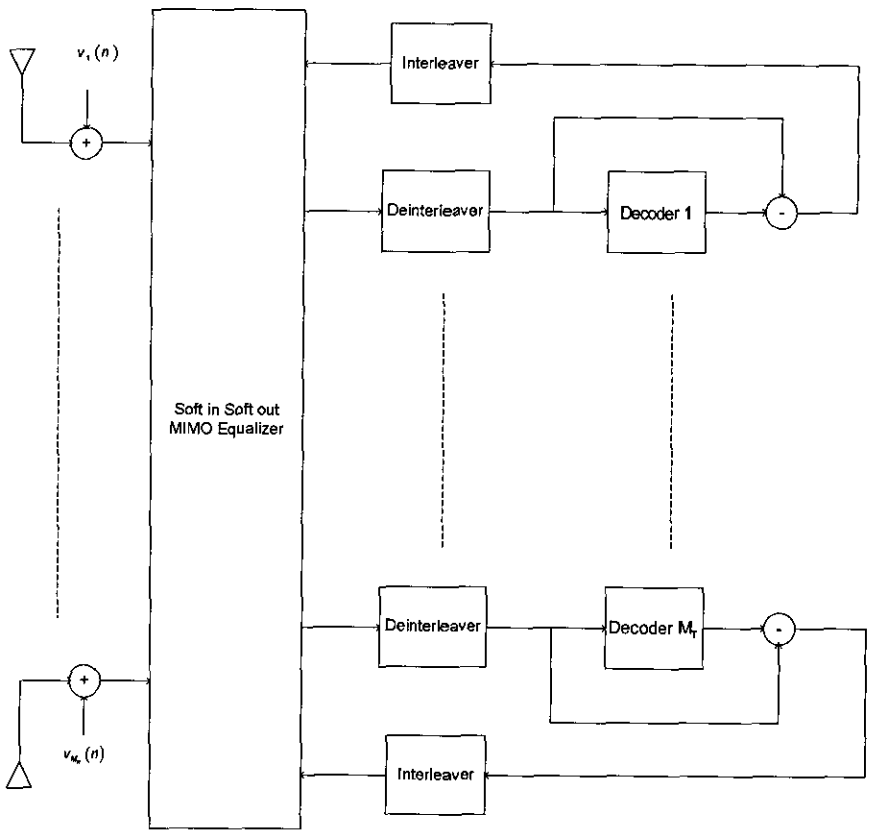

Figure 4.7. A block diagram of the receiver for space time turbo equalization with horizontal encoding.

where $\mathbf{c} = \begin{bmatrix} \mathbf{c}_1^T & \cdots & \mathbf{c}_{M_T}^T \end{bmatrix}^T$ and $\hat{\mathbf{H}}$ is generated from the estimates of the BEM coefficients. The DC offset estimate is subtracted from the received signal before processing. The matrix $\mathbf{P}$ is the following permutation matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_{M_R} \otimes \mathbf{e}_1 & \cdots & \mathbf{I}_{M_R} \otimes \mathbf{e}_N \end{bmatrix}^T \tag{4.5.26}$$

where $\mathbf{e}_m$ is the m^{th} column of the $N \times N$ identity matrix. Also defining $\dot{\mathbf{x}}$ and $\dot{\mathbf{v}}$ accordingly, we have

$$\dot{\mathbf{x}} \triangleq \underbrace{\begin{bmatrix} \mathbf{I}_{M_T} \otimes \mathbf{e}_1 & \cdots & \mathbf{I}_{M_T} \otimes \mathbf{e}_N \end{bmatrix}^T}_{\mathbf{T}} \mathbf{x} \tag{4.5.27}$$

We may write

$$\dot{\mathbf{y}} = \mathbf{P}(\hat{\mathbf{H}}\mathbf{x} + \tilde{\mathbf{H}}\mathbf{x} + \tilde{\mathbf{H}}\mathbf{c} + \tilde{\mathbf{m}} \otimes \mathbf{1}_N + \mathbf{v}) = \underbrace{\mathbf{P}\hat{\mathbf{H}}\mathbf{T}^H}_{\dot{\mathbf{H}}}\dot{\mathbf{x}} + \dot{\mathbf{v}} \tag{4.5.28}$$

where $\dot{\mathbf{v}}$ is the effective noise. The space time turbo equalizer (MIMO soft input soft output equalizer) operates on the the residual interference by subtracting a soft replica of the interference (combination of the MSI from undesired streams and the ISI from the stream of interest) from the received signals at the M_R receive antennas over L_{eq} delays as

$$\tilde{\mathbf{y}}_k^n = \dot{\mathbf{y}}_k - \dot{\mathbf{H}}_k \hat{\mathbf{x}}_k^n \tag{4.5.29}$$

where

$$\dot{\mathbf{y}}_k = (\begin{bmatrix} \mathbf{e}_{k-\lfloor L_{eq}/2 \rfloor} & \cdots & \mathbf{e}_{k+\lceil L_{eq}/2 \rceil} \end{bmatrix}^T \otimes \mathbf{I}_{M_R})\dot{\mathbf{y}} \tag{4.5.30}$$

and the matrix $\dot{\mathbf{H}}_k$ is a sub block of $\dot{\mathbf{H}}$ given by

$$\dot{\mathbf{H}}_k = \mathbf{\Psi}\dot{\mathbf{H}}\mathbf{\Phi} \tag{4.5.31}$$

where $\mathbf{\Phi} = \begin{bmatrix} \mathbf{e}_{k-\lfloor (L_{eq}+L)/2 \rfloor} & \cdots & \mathbf{e}_{k+\lceil (L_{eq}+L)/2 \rceil} \end{bmatrix} \otimes \mathbf{I}_{M_T}$ is a $NM_T \times (L_{eq} + L + 1)M_T$ matrix and $\mathbf{\Psi} = \begin{bmatrix} \mathbf{e}_{k-\lfloor L_{eq}/2 \rfloor} & \cdots & \mathbf{e}_{k+\lceil L_{eq}/2 \rceil} \end{bmatrix}^T \otimes \mathbf{I}_{M_R}$ is a $(L_{eq} + 1)M_R \times NM_R$ matrix. The vector $\hat{\mathbf{x}}_k^n$ contains the estimates of $\mathbf{\Phi}\dot{\mathbf{x}} = \mathbf{\Phi}\mathbf{T}\mathbf{x}$ used to construct residual soft interference

$$\hat{\mathbf{x}}_k^n = \begin{bmatrix} \hat{\mathbf{x}}^T[k - \lfloor L_{eq}/2 \rfloor] & \cdots & \hat{\mathbf{x}}^T[k + \lceil L_{eq}/2 \rceil] \end{bmatrix}^T \tag{4.5.32}$$

where $\hat{\mathbf{x}}^T[m] = \begin{bmatrix} \hat{x}_1[m] & \cdots & \hat{x}_{M_T}[m] \end{bmatrix}$ but care is taken to set $\hat{x}_n[k]$ equal to zero. The space time soft input soft output equalizer for the n^{th} user is then constructed by minimizing the cost function

$$\mathbf{g}_k^n = \arg\min_{\mathbf{g}_k^n} \|\mathbf{g}_k^{nH}\bar{\mathbf{y}}_k^n - \dot{x}_n[k]\|^2 \tag{4.5.33}$$

and the equalizer is obtained as [93]

$$\mathbf{g}_k^n = (\dot{\mathbf{H}}_k \mathbf{Q}_k^n \mathbf{H}_k^H + \sigma^2 \mathbf{I}_{(L_{eq}+1)M_R})^{-1} \dot{\mathbf{H}}_k \mathbf{1}_n \tag{4.5.34}$$

where $\mathbf{1}_n$ is a $(L + L_{eq} + 1)M_R \times 1$ vector of zeros with a one in position $\lfloor (L + L_{eq})/2 \rfloor M_R + n$. The matrix $\mathbf{Q}_k^n$ is

$$\mathbf{Q}_k^n = \text{diag}(\begin{bmatrix} \mathbf{Q}[k - \lfloor (L + L_{eq})/2 \rfloor] & \cdots & \mathbf{Q}[k + \lceil (L + L_{eq})/2) \rceil] \end{bmatrix}) \tag{4.5.35}$$

and $\mathbf{Q}[m] = \text{diag}(\begin{bmatrix} 1 - \hat{u}_1^2[m] & \cdots & 1 - \hat{u}_{M_T}^2[m] \end{bmatrix})$ but again care is taken to set $\hat{x}_n^2[k] = 0$. The log likelihood ratios produced by the equalizer are [93]

$$\lambda(\dot{x}_n[k]) = \frac{4Re(z_n[k])}{1 - \mu_n[k]} \tag{4.5.36}$$

where $z_n[k] = \mathbf{w}_k^{nH} \tilde{\mathbf{y}}_k^n$ is the equalizer output and

$$\mu_n[k] = \mathbf{1}_n^H \dot{\mathbf{H}}_k^H (\dot{\mathbf{H}}_k \mathbf{Q}_k^n \mathbf{H}_k^H + \sigma^2 \mathbf{I}_{(L_{eq}+1)M_R})^{-1} \dot{\mathbf{H}}_k \mathbf{1}_n \tag{4.5.37}$$

A rate 1/3 convolutional code with generator polynomial $(145, 133, 7)_8$ was employed for each stream.

4.5.3 Simulation Results

We simulated a simple MIMO system with $M_T = 2$ transmit antennas and $M_R = 3$ receive antennas. The channels between the transmitter and receiver are doubly selective with order $L = 3$ and doppler spread normalized to the symbol rate $f_d = 0.005$. We took a block size of $N = 2400$ $(Q/2 = 12)$. No prior channel knowledge was assumed and joint channel estimation equalization and decoding was performed over each block. Mean square estimation error and bit error rate results are presented in figures 4.8 and 4.9 where the known training is included in the bit energy definition. It can be observed from the simulation results that the space time turbo equalization provides

Algorithm 2 Iterative Channel Estimation and Equalization and Decoding algorithm

1. At the transmitter, choose a periodic training sequence according to section 4.5.1, and add it onto the interleaver shuffled coded data stream
2. Form an initial BEM coefficient estimate $\hat{\mathbf{h}}$ according to (4.5.22) and an initial estimate of the dc offset according to (4.5.23).
3. Carry out an iteration of the space time turbo equalization algorithm using $\hat{\mathbf{h}}$ and $\hat{\mathbf{m}}$.
4. Convert the LLRs generated by the decoder into soft estimates, interleave them and add the known training to form a virtual pilot $\mathbf{p}_\kappa$ for all transmitters.
5. Form Ξ_κ as the Toeplitz matrix of $\mathbf{p}_\kappa$, and then obtain the BEM coefficient estimate as $\hat{\mathbf{h}} = (\mathbf{I}_{M_R} \otimes \Xi^H \Xi)^{-1}(\mathbf{I}_{M_R} \otimes \Xi^H)\mathbf{y}$ where $\Xi = \begin{bmatrix} \Xi_1 & \cdots & \Xi_{M_T} \end{bmatrix}$.
6. Iterate steps 3 to 6.

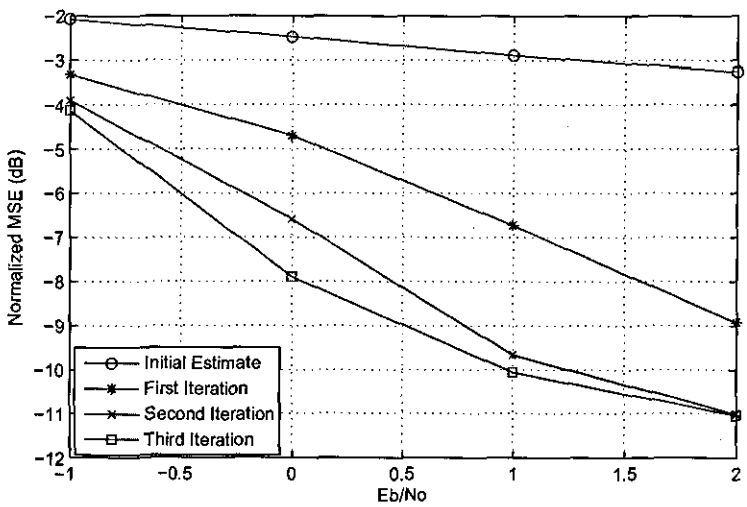

Figure 4.8. The normalized estimation mean square error is plotted against the Eb/No for the iterations.

significant performance improvements over three iterations. These simulation results show that the normalized estimation mean square error improves by 7 dB at Eb/No = 1 dB while the BER is reduced by approximately two orders of magnitude as compared to a non iterative scheme.

Although the space time turbo equalization algorithm can provide perfor-

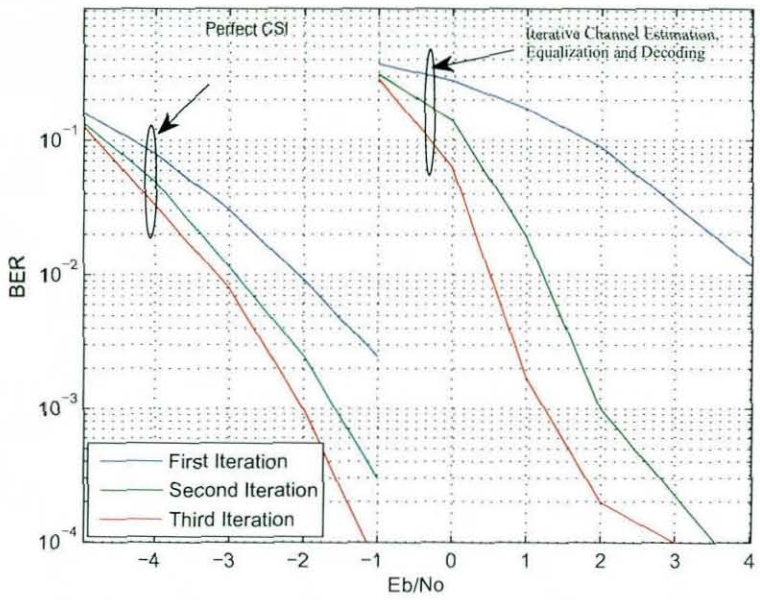

Figure 4.9. The BER is plotted against Eb/No for various iterations of the algorithm.

mance improvements over many iterations, the algorithm requires the use of large block lengths to suppress the correlations introduced by the constituent forward backward algorithms. But because of the time varying nature of the doubly selective channel, the block length would usually be kept small to moderate sized. The BEM is a linear approximation of the doubly selective channel that may only be valid locally. Furthermore, larger block lengths will require more parameters to be estimated and increase estimation complexity. Furthermore, the equalization delay for turbo equalization of time invariant channels is often optimized[6]. The taps of doubly selective channels for the various transmitter receiver pairs are time varying and so is the optimal equalization delay. In our simulations, we fix the equalization delay and do not track the optimal delay with the time variations. Because of these imperfections, the turbo equalization with iterative channel estimation only provides performance gains over three iterations. Nevertheless, these

[6]The equalization delay was optimized for the results provided in Chapter 2.

performance gains provided are significant.

4.6 Conclusion

Superimposed training provides a simple channel estimation algorithm. Similarly the turbo equalization algorithm provides remarkable performance when the channel is perfectly known at the receiver. However complications arise in the turbo equalization procedure with imperfect CSI. Similarly, superimposed training performance improvement requires an increase in transmission power (or block length). Combining the two algorithms by including channel estimation in the iterative process, significant improvements in the performance can still be obtained without the need for any extra power transmission. This can have very significant effects both at the physical and network layers. The SNR required for a target BER is greatly reduced which may translate into a reduction in the transmit power, an increase in the range, or even an increase in throughput. It can also mean reduced interference in the cells.

Chapter 5

TIME MULTIPLEXED TRAINING FOR DOUBLY SELECTIVE CHANNELS

5.1 Introduction

An alternative training technique is to multiplex the pilot into the unknown data. This technique, called explicit training, has been around for much longer than superimposed training and has been widely studied for time and frequency selective channels. For time selective channels, it is termed as pilot symbol assisted modulation (PSAM) [94]. It has been applied to both single carrier and multicarrier communications for frequency selective channels.

These techniques are generally used to decouple the channel estimation and the data detection so that the data transmission does not interfere with the channel estimation. This allows accurate channel estimation while reducing complexity and relaxing the identifiability conditions.

Initially, the Cramer-Rao lower bound (CRLB) and the estimator mean square error were used for training design and optimization. But it was quickly realized that channel estimation is just one facet of the problem and the design of training affects not only the channel estimate but also the detection of the unknown symbols. While increasing power allocation to pilots will always result in better channel estimates with smaller error, it will also

reduce the power available for data transmission to combat noise. Similarly, an increasing number of pilots will also provide better channel estimation but will leave little time for data transmission resulting in diminishing data rates. The signal processing performance criterion of estimation MSE was replaced by the information theoretic criterion of training based capacity as the appropriate criterion for optimal design of training. Training based capacity [95] makes a global assessment of the communications system. It is different from capacity in the traditional sense in that it takes into consideration the estimation of the unknown channel.

The contribution of this work is the application of iterative estimation and detection for time multiplexed training in doubly selective channels. We use existing optimal training designs for training and apply iterative techniques to improve on the performance.

5.2 Frequency Selective Channel

For single carrier communications, the pilot symbols are grouped together and transmitted at the beginning of a block in what is called a training preamble. The baseband vector matrix block transmission model for frequency selective channels is

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} = \mathbf{X}\mathbf{h} + \mathbf{v} \tag{5.2.1}$$

where $\mathbf{h} = \begin{bmatrix} h[0] & \cdots & h[L] \end{bmatrix}^T$ where L is the channel order and $\mathbf{H}$ is a convolution matrix constructed from $\mathbf{h}$. The transmit block is assumed to span $P + N$ symbols, where the first P symbols constitute the training preamble and the remaining N compose unknown data. We can write

$$\mathbf{x} = \begin{bmatrix} c_1 & \cdots & c_P & s_1 & \cdots & s_N \end{bmatrix} \tag{5.2.2}$$

and

$$\mathbf{X} = \begin{bmatrix} \mathbf{C} \\ \mathbf{S} \end{bmatrix} = \left[\begin{array}{cccc} c_{L+1} & 0 & \cdots & 0 \\ \vdots & c_{L+1} & & \vdots \\ c_P & & \ddots & 0 \\ \hline s_1 & c_P & & c_{L+1} \\ \vdots & s_1 & \ddots & \vdots \\ s_N & & \ddots & c_P \\ 0 & s_N & & s_1 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & s_N \end{array}\right] \tag{5.2.3}$$

We can partition the received signal into the first P received samples which only depend on the training and the remaining N samples which are determined by the data and have interference from the training

$$\begin{bmatrix} \mathbf{y}_c \\ \mathbf{y}_s \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{S} \end{bmatrix} \mathbf{h} + \begin{bmatrix} \mathbf{v}_c \\ \mathbf{v}_s \end{bmatrix} \tag{5.2.4}$$

Since we know $\mathbf{c}$, we can form the maximum likelihood channel estimate as

$$\hat{\mathbf{h}} = (\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H\mathbf{y}_c \tag{5.2.5}$$

To ensure that there is no interblock interference, the transmitter is silent in the first L transmissions, i.e. $c_i = 0, i \leq L$. There are $L+1$ parameters to be estimated, so we require at least $L+1$ observations. Therefore, the training preamble must contain $P \geq 2L+1$ pilots. This channel estimate may then be used for detecting the received signal. However, a word of caution is necessary because any receiver that assumes this estimate to be the true channel will be suboptimal. The process of using such a channel estimate in the receiver as if it were the true channel is information lossy, i.e.

$$I(\mathbf{y}_s; \mathbf{s}|\mathbf{h}) \geq I(\mathbf{y}_s; \mathbf{s}|\hat{\mathbf{h}}) \tag{5.2.6}$$

where $I(x;y)$ denotes the mutual information between random variables x and y [96].

For $P = 2L+1$, this preamble placement is optimal from the training based capacity perspective. However, in case of significantly more training, $P \gg 2L+1$, it has been shown by [97] that the training based capacity is actually maximized by the use of a placement scheme called the quasi periodic placement (QPP-α) where the pilots are divided into as many clusters of length $\alpha \geq 2L+1$ as possible and these training clusters are placed as far apart as possible with the separation between them as uniform as possible. Assuming a block consisting of N_s data symbols and N_c pilot symbols, an α and let $P = \lfloor N_c/\alpha \rfloor$, then a QPP-$\alpha$ scheme can be defined by any two tuple $(\mathbf{n}_s, \mathbf{n}_c)$ that satisfies the following two conditions [97]

1. $\mathbf{n}_c \in \mathcal{N}_c$ *where*
$$\mathcal{N}_c = \{(N_{c,1}, \ldots, N_{c,P}) \mid \sum_{p=1}^{P} N_{c,p} = N_c, \min(\{N_{c,1}, \ldots, N_{c,P}\}) \geq \alpha\}$$
2. $\mathbf{n}_s \in \mathcal{N}_s$ *where*
$$\mathcal{N}_s = \{(N_{s,1}, \ldots, N_{s,P}) \mid \sum_{p=1}^{P} N_{s,p} = N_s, N_{s,p} \in \{\lfloor N_s/P \rfloor, \lfloor N_s/P \rfloor + 1\}$$

Simply put, the training clusters in QPP-α should be as uniform as possible and scattered as uniformly as possible. A transmit block with QPP-α placement is depicted in figure 5.1.

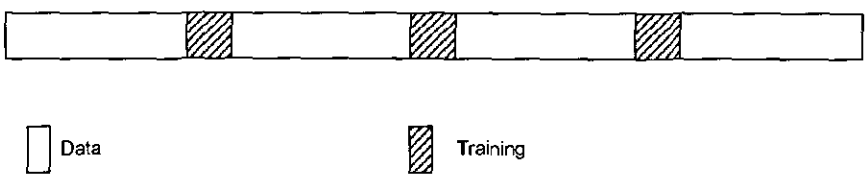

Figure 5.1. Pilots embedded in a transmit block by time multiplexing them with the data according to QPP-α.

The optimality of the QPP-α scheme stems from the concavity of the logarithm. Assuming a received signal to noise ratio ρ and that all training

clusters contain at least L symbols so that successive data clusters do not interfere with one another, the capacity of such a training scheme can be expressed as

$$C = \frac{1}{N_s + N_c} \log \det(\mathbf{I}_{N_s+N_c} + \rho \mathbf{H} \mathbf{\Theta} \mathbf{H}^H) \tag{5.2.7}$$

where $\mathbf{\Theta}$ is a diagonal matrix with ones in the positions corresponding to data symbols and zeros for training. This may be expressed as [98]

$$C = \frac{1}{N_s + N_c} \sum_{p=1}^{P} \log \det(\mathbf{I}_{N_s,p} + \rho \mathbf{H}_p \mathbf{H}_p^H) \tag{5.2.8}$$

where $\mathbf{H}_p$ is the segment of the channel matrix corresponding to the p^{th} data block. Jensen's inequality states that for a convex function f of a random variable X,

$$E\big(f(X)\big) \geq f\big(E(X)\big) \tag{5.2.9}$$

It can now be seen by Jensen's inequality that because of the concave nature of the logarithm, this summation is maximized by having all P terms as equal as possible. The optimal placement is thus QPP-α. Intuitively, in addition to estimating the channel, the training helps to remove ISI. The more the training symbols are scattered, the more ISI that is known apriori and can simply be subtracted out once the channel estimate has been obtained. This leaves less ISI for the system to contend with, improving performance. The gain offered by the QPP-α scheme over a preamble based or another scheme is greater with more severe ISI in the channel. Similarly the gap between the use of QPP-α placement and any other arbitrary placement is larger at higher SNR because the link performance is dominated by ISI at higher SNRs.

The problem of placement of known symbols for training users in broadcast channels was studied in [98] where it was again shown that the QPP-α placement is optimal for the outage capacity. In broadcast applications, periodic transmission of training is necessary as the users may tune in at any

time. The surprising aspect of this result is that the optimal placement is independent of the channel statistics. While different users may experience different channels, the QPP-α placement is optimal for all users irrespective of their channels. It has also been shown in [99] that the QPP-α scheme is optimal for training in slowly fading wireless channels, i.e. channels over which it is not possible to code over multiple coherence intervals.

5.3 Time Selective Channel

For a memoryless[1] SISO time selective channel, PSAM design consists of inserting lone pilot symbols periodically in the stream of the unknown data. The time selective channel could be obtained through the use of a BEM. The fading waveform exhibits a great deal of correlation. Although the channel realization at each symbol is different, it is highly correlated with the adjacent channel realizations thus underscoring the fact that the varying channel is generated by a small number of fixed underlying parameters which are responsible for these variations. The channel variation is caused by Doppler spreads and carrier frequency offsets, so it is reasonable to expect the Doppler domain to provide a finite parameter representation of the channel. The channel fading waveform can be expressed as the weighted sum of exponentials

$$\mathbf{h} = \sum_{q=-Q/2}^{Q/2} \mathbf{d}_q h_q = \underbrace{\begin{bmatrix} \mathbf{d}_{-Q/2} & \cdots & \mathbf{d}_{Q/2} \end{bmatrix}}_{\mathbf{D}} \mathfrak{h} \tag{5.3.1}$$

where $\mathfrak{h} = \begin{bmatrix} h_{-Q/2} & \cdots & h_{Q/2} \end{bmatrix}^T$ are the unknown parameters and $\mathbf{d}_q = \begin{bmatrix} 1 & e^{-j\frac{2\pi q}{N}} & \cdots & e^{-j\frac{2\pi q(N-1)}{N}} \end{bmatrix}^T$. This allows modeling of any fading wave-

[1]The use of memoryless here implies that the channel has no delay spread, i.e. a flat frequency response. In general a time selective channel does have memory in the sense that the fading waveform is correlated and so is the channel behavior. When the channel is in a fade and more errors occur, it remembers that it is in a fade and continues to produce more errors for the duration of the fade.

form resulting from a Doppler spread of f_d by choosing $Q = 2\lceil f_d N \rceil$. The input output relation is

$$\mathbf{y} = \mathbf{h} \odot \mathbf{x} + \mathbf{v} = \mathbf{D}\mathfrak{h} \odot \mathbf{x} + \mathbf{v} \tag{5.3.2}$$

where $\odot$ is the Hadamard product. We require estimates of $Q+1$ parameters, so we need at least $Q+1$ observations. The transmit signal is of the form

$$\mathbf{x} = \begin{bmatrix} \mathbf{s}_1^T & c_1 & \mathbf{s}_2^T & c_2 & \cdots & \mathbf{s}_{Q+1}^T & c_{Q+1} \end{bmatrix}^T \tag{5.3.3}$$

where the c_i are training symbols. The $Q+1$ observations thus obtained by inserting $Q+1$ pilots are given by

$$\mathbf{y}_{pilot} = \mathbf{D}_{pilot}\mathfrak{h} \odot \mathbf{c} + \mathbf{v}_{pilot} \tag{5.3.4}$$

These can then be used to get the channel estimate as

$$\hat{\mathfrak{h}} = \mathbf{D}_{pilot}^{-1}\mathbf{z}_{pilot} \tag{5.3.5}$$

where $\mathbf{z}_{pilot}(k) = \mathbf{y}_{pilot}(k)/\mathbf{c}(k)$. As an alternative to the BEM, the first order Gauss Markov process has also been used to model the channel variations in time where a recursive least squares (RLS) or Kalman filter may be used to estimate the channel [100].

5.4 SISO Doubly Selective Channels

Explicit training involves time division multiplexing of the training with the data. The design of explicit training has been studied by various authors in the literature. Periodic insertion of pilots, termed as pilot symbol assisted modulation (PSAM), was proposed in [94] for time selective channels. As the problem of channel estimation became clearer, different optimization criteria for the design of training were proposed.

[95] studied the optimization of training for multiple input multiple output flat fading channels. The degrees of freedom to be optimized in such a channel were the power allocation between training and data, the number of training symbols and the choice of these training symbols. Some important insights of their work were that training based schemes are strictly suboptimal when the coherence time is small or when the SNR is low but they are capacity approaching at large coherence times and large SNR. Optimal design of training for frequency selective channels was considered soon after by [97] and [101]. While [101] considered only the single carrier system, both researchers came to the same optimal design where the training is concentrated in a single cluster at one end of the block. An interesting result in [97] was that the same placement which maximized the training based capacity also minimized the channel estimation mean square error. Further [97] considered the multicarrier scenario where he proposed a pilot tone assisted modulation scheme where the pilots are inserted in uniformly separated tones for maximizing the training based capacity. Interestingly, this scheme had previously been proposed by [51] as an optimal placement in terms of the estimation mean square error. The problem was then considered for doubly selective channels by [102] where optimal training sequences were designed for doubly selective channels. The first section of this chapter concerning explicit training will try to build on the work by [102] and much of the rest of this section will therefore follow [102].

5.4.1 Channel Estimation

We consider a block based transmission scheme where the channel estimation and data detection is performed on a block by block basis. The transmit vector is partitioned into P information and pilot sub-blocks

$$\mathbf{x} = \begin{bmatrix} \mathbf{s}_1^T(k) & \mathbf{c}_1^T(k) & \dots & \mathbf{s}_P^T(k) & \mathbf{c}_P^T(k) \end{bmatrix}^T \tag{5.4.1}$$

where the $\mathbf{s}_p(k)$ and $\mathbf{c}_p(k)$ represent information and training sub-blocks of length $N_{s,p}$ and $N_{c,p}$. Let the total number of training symbols be N_c and the total number of information symbols be N_s so that $N = N_s + N_c$. Also, $\sum_{p=1}^{P} N_{s,p} = N_s$ and $\sum_{p=1}^{P} N_{c,p} = N_c$.

Thus the output of the k^{th} block of N samples may be expressed using the matrix vector notation as

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{H}^{ibi}(k)\mathbf{x}(k-1) + \mathbf{v}(k) \tag{5.4.2}$$

where $\mathbf{y}(k) = \begin{bmatrix} y(kN) & \dots & y(kN+N-1) \end{bmatrix}^T$ and the noise is assumed to be additive white Gaussian. The matrices $\mathbf{H}(k)$ and $\mathbf{H}^{ibi}(k)$ are $N \times N$ lower and upper triangular matrices defined by $[\mathbf{H}(k)]_{n,m} = h(kN+n, n-m)$ and $[\mathbf{H}^{ibi}(k)]_{m,n} = h(kN+n, N+n-m)$ for $n, m = 1, \dots, N$. For IBI free transmission, either a cyclic prefix or a zero prefix may be inserted at the transmitter and the interference corrupted samples are discarded at the receiver. We use zeros prefixed transmission [12]. With IBI free transmission, we get $\mathbf{H}^{ibi}(k)\mathbf{x}(k-1) = \mathbf{0}$, so

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \tag{5.4.3}$$

Because of the zero prefix, the transmitter has to be silent for the last L symbol periods in a block where L is the channel order. Because the zero prefix is not used for data transmission so these L symbols are thus considered as a part of the pilot. The BEM allows the channel matrix to be written as (see 4.3.6)

$$\mathbf{H} = \sum_{q=0}^{Q} \mathbf{D}_q \mathbf{H}_q \tag{5.4.4}$$

where $\mathbf{D}_q = \text{diag}\begin{bmatrix} 1 & e^{jw_q} & \dots & e^{jw_q(N-1)} \end{bmatrix}$ and $\mathbf{H}_q$ is a lower triangular Toeplitz matrix with first column $\begin{bmatrix} h_q(0) & \dots & h_q(L) & 0 & \dots & 0 \end{bmatrix}^T$. The received vector $\mathbf{y}$ is now partitioned into two vectors, $\mathbf{y_s}$ and $\mathbf{y_c}$. $\mathbf{y_c}$ contains

the received symbols due to training that are free of any interference from the unknown data while $\mathbf{y}_s$ contains all other received symbols. These received symbols, $\mathbf{y}_c$, will be used for channel estimation. The channel matrix $\mathbf{H}$ can also be decomposed according to this partition of the received samples into $\mathbf{H}_c$ which relates the input training samples with $\mathbf{y}_c$, $\mathbf{H}_s$ which relates the unknown data with $\mathbf{y}_s$ and $\bar{\mathbf{H}}_c$ which describes the contribution of the training to $\mathbf{y}_s$ [102]

$$\mathbf{y}_s = \mathbf{H}_s\mathbf{s} + \bar{\mathbf{H}}_c\bar{\mathbf{c}} + \mathbf{v}_s \tag{5.4.5}$$

$$\mathbf{y}_c = \mathbf{H}_c\mathbf{c} + \mathbf{v}_c \tag{5.4.6}$$

where $\mathbf{s} = \begin{bmatrix}\mathbf{s}_1^T & \dots & \mathbf{s}_P^T\end{bmatrix}^T$ and $\mathbf{c} = \begin{bmatrix}\mathbf{c}_1^T & \dots & \mathbf{c}_P^T\end{bmatrix}^T$ and $\bar{\mathbf{c}}$ is formed from the elements of $\mathbf{c}_p$ that interfere with $\mathbf{s}$. Similarly $\mathbf{v}_c$ and $\mathbf{v}_s$ denote the noise vectors. This partition of the convolution matrix for one block is depicted in figure 5.2. The shaded region indicates the interference of training into the data observations $\mathbf{y}_s$. The structure of $\mathbf{H}_s$, $\bar{\mathbf{H}}_c$ and $\mathbf{H}_c$ is depicted in figure 5.3. The output vector that solely depends on the pilots may be written as [102]

$$\mathbf{y}_c = \begin{bmatrix}\mathbf{y}_1^c \\ \vdots \\ \mathbf{y}_P^c\end{bmatrix} = \begin{bmatrix}\mathbf{H}_1^c\mathbf{c}_1 \\ \vdots \\ \mathbf{H}_P^c\mathbf{c}_P\end{bmatrix} + \mathbf{v}_c \tag{5.4.7}$$

If any of the training sub-blocks $\mathbf{c}_p$ has less than L symbols, the matrix $\mathbf{H}_p^c$ vanishes altogether and the training symbols in the p^{th} cluster do not contribute to an estimate of the channel. Thus each of the P clusters must have at least L training symbols each. Notice that there are N_c training symbols in P clusters, thus providing $N_c - PL$ pilot dependant observations because the first L symbols in each cluster are affected by interference from the data transmission. As there are $(Q+1)(L+1)$ unknown parameters to

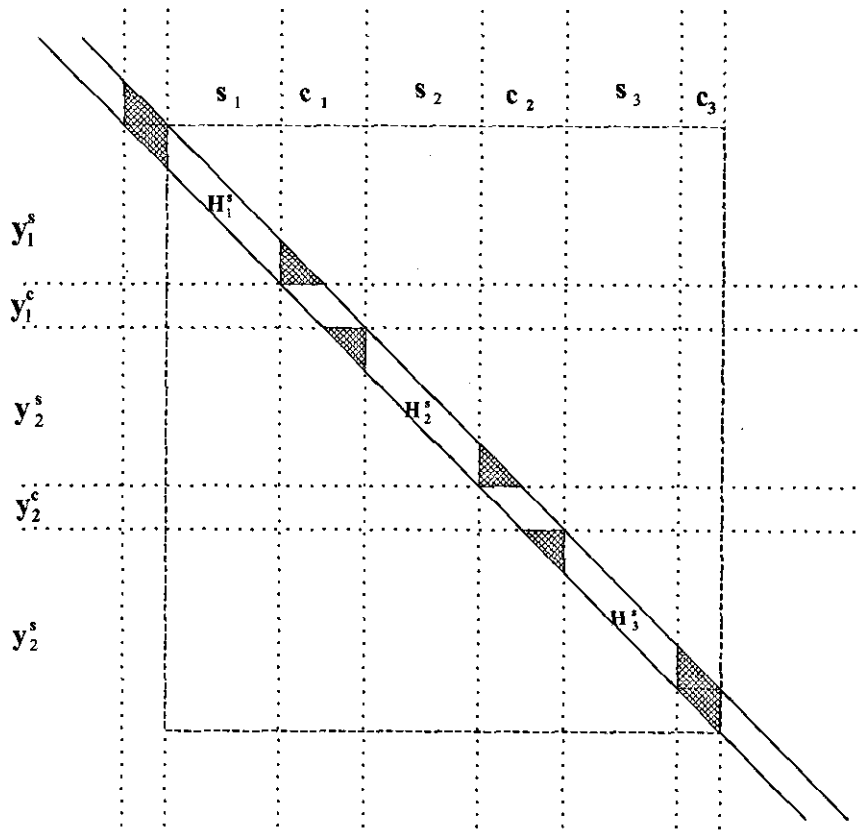

Figure 5.2. Convolution matrix.

be estimated, the number of training symbols must satisfy [102]

$$N_c \geq PL + (Q+1)(L+1) \tag{5.4.8}$$

to ensure a unique solution. We see that the number of training symbols depends on the number of training clusters P and the minimum number would be achieved if only one cluster were to be used, similar to the preamble based approach for frequency selective channels. However, an optimal training is

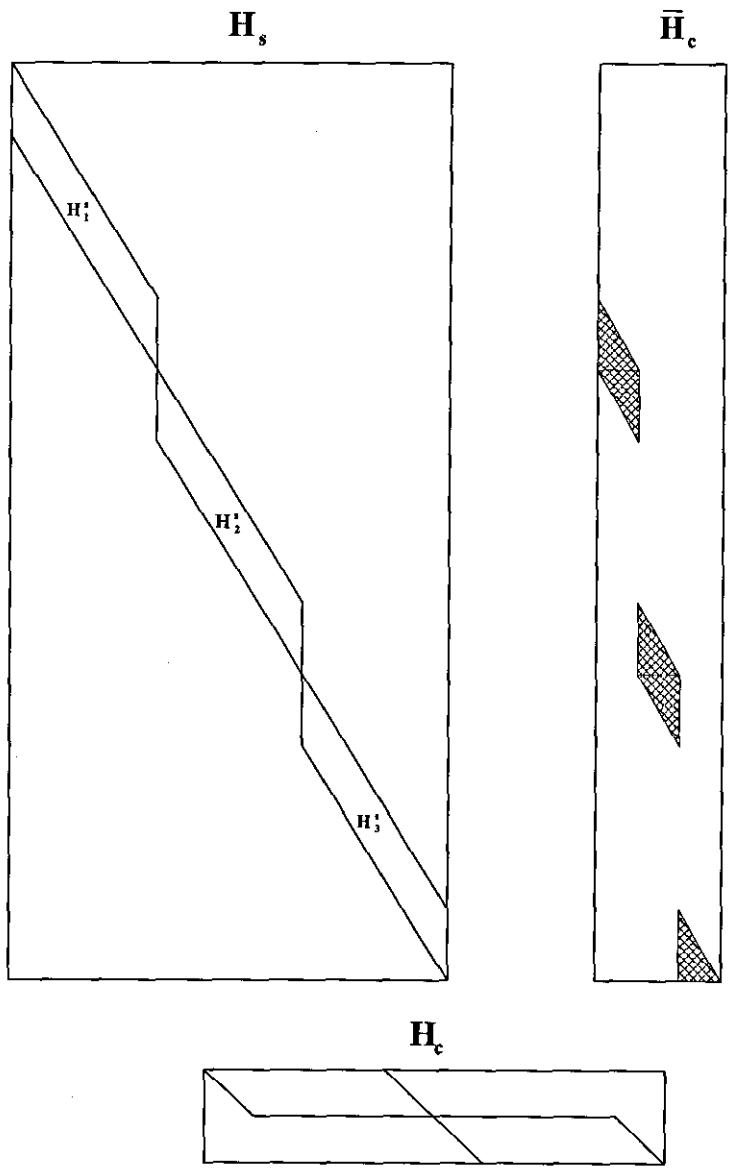

Figure 5.3. Partition of the convolution matrix.

sought we will see that $P = Q + 1$ is optimal [102]. Equation (5.4.4) may now be used to write [102]

$$\mathbf{y}_c = \sum_{q=0}^{Q} \begin{bmatrix} \mathbf{D}_{q,1}^c \mathbf{H}_{q,1}^c \mathbf{c}_1 \\ \vdots \\ \mathbf{D}_{q,P}^c \mathbf{H}_{q,P}^c \mathbf{c}_P \end{bmatrix} + \mathbf{v}_c \tag{5.4.9}$$

Because the channel coefficients are to be estimated using the known pilot symbols, the convolution should be expressed using Toeplitz data matrices, i.e. $\mathbf{H}^c_{q,p}\mathbf{c}_p = \mathbf{C}_p\mathbf{h}_q$ where $\mathbf{C}_p$ is a l^{th} order Toeplitz matrix

$$\mathbf{C}_p = \begin{bmatrix} c_{p,L} & \cdots & c_{p,0} \\ \vdots & \ddots & \vdots \\ c_{p,N_{b,p}-1} & \cdots & c_{p,N_{b,p}-L-1} \end{bmatrix} \tag{5.4.10}$$

and

$$\mathbf{h}_q = \begin{bmatrix} h_q(0) \\ \vdots \\ h_q(L) \end{bmatrix} \tag{5.4.11}$$

where $\mathbf{c}_{p,n}$ is the $(n+1)st$ entry of $\mathbf{c}_p$. The input output relation in (5.4.9) may be written as

$$\mathbf{y}_c = \mathbf{\Phi}_c\mathbf{h} + \mathbf{v}_c \tag{5.4.12}$$

where

$$\mathbf{\Phi}_c = \begin{bmatrix} \mathbf{D}^c_{0,1}\mathbf{C}_1 & \cdots & \mathbf{D}^c_{Q,1}\mathbf{C}_1 \\ \vdots & \ddots & \vdots \\ \mathbf{D}^c_{0,P}\mathbf{C}_P & \cdots & \mathbf{D}^c_{Q,P}\mathbf{C}_P \end{bmatrix} \tag{5.4.13}$$

and

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_0^T & \cdots & \mathbf{h}_Q^T \end{bmatrix}^T \tag{5.4.14}$$

The Wiener solution of (5.4.12) is the conditional mean of $\mathbf{h}$ given $\mathbf{c}, \mathbf{y}_c$ and it can be obtained as [102]

$$\hat{\mathbf{h}} = E(\mathbf{h}|\mathbf{c}, \mathbf{y}_c) = \frac{1}{\sigma_v^2}(\mathbf{R}_h^{-1} + \frac{1}{\sigma_v^2}\mathbf{\Phi}_c^H\mathbf{\Phi}_c)^{-1}\mathbf{\Phi}_c^H\mathbf{y}_c \tag{5.4.15}$$

The covariance of the estimation error $\tilde{\mathbf{h}} = \mathbf{h} - \hat{\mathbf{h}}$ is given by [102]

$$\mathbf{R}_{\tilde{\mathbf{h}}} = E(\hat{\mathbf{h}}\hat{\mathbf{h}}) = (\mathbf{R}_h^{-1} + \frac{1}{\sigma_v^2}\mathbf{\Phi}_c^H\mathbf{\Phi}_c)^{-1} \tag{5.4.16}$$

and the estimation mean square error is $\sigma_{\tilde{h}}^2 = \mathrm{tr}(\mathbf{R}_{\tilde{h}})$. The statistics $\mathbf{R}_h$ of the channel are assumed to be known at the receiver. In a rich scattering environment with no line of sight, the central limit theorem assures that these statistics will approximate the identity matrix [102]. The estimation mean squared error is affected by the placement through $\mathbf{\Phi_c}$.

The design of optimal training has been obtained by [102]. According to this design, the transmitter should be silent during the first and last L samples of a training cluster. Intuitively, the first L received samples in a training cluster are corrupted by interference from the unknown data and cannot be used for channel estimation. Thus the energy allocated to these samples is not completely useful for channel estimation. In fact, only the energy that is received after a delay of the initial L samples can be used while the rest is simply wasted. Similarly, because of the delay profile of the channel, all the energy transmitted during the last L samples of the cluster is also not used. Once unknown data paths begin to arrive, the multipaths associated with the training become useless wastage as they cannot be decoupled from the unknown data and used. In fact, the training now constitutes interference to the data transmission. This interference may be peeled off using the channel estimate, but there will still be some residual interference due to channel estimation error. Thus to ensure that all energy allocated to the pilot is used for estimation, the transmitter should be silent during the first and last L samples of any training cluster and to concentrate its energy in the window between these samples. This is depicted in figure 5.4 where we see that if the training is not appropriately designed, it interferes with the data transmission as depicted in part a of the figure. However, if the transmitter is silent during the transmission of the first and last L samples of a training cluster the data transients die out before training and the training transients are allowed to die out before data transmission is resumed as depicted in part b of the figure. This allows the pilot energy to be concentrated in only the samples which will be used to measure the

channel. The transmitter thus avoids interference between the training and the unknown data altogether facilitating a better training based channel capacity and a better channel estimate.

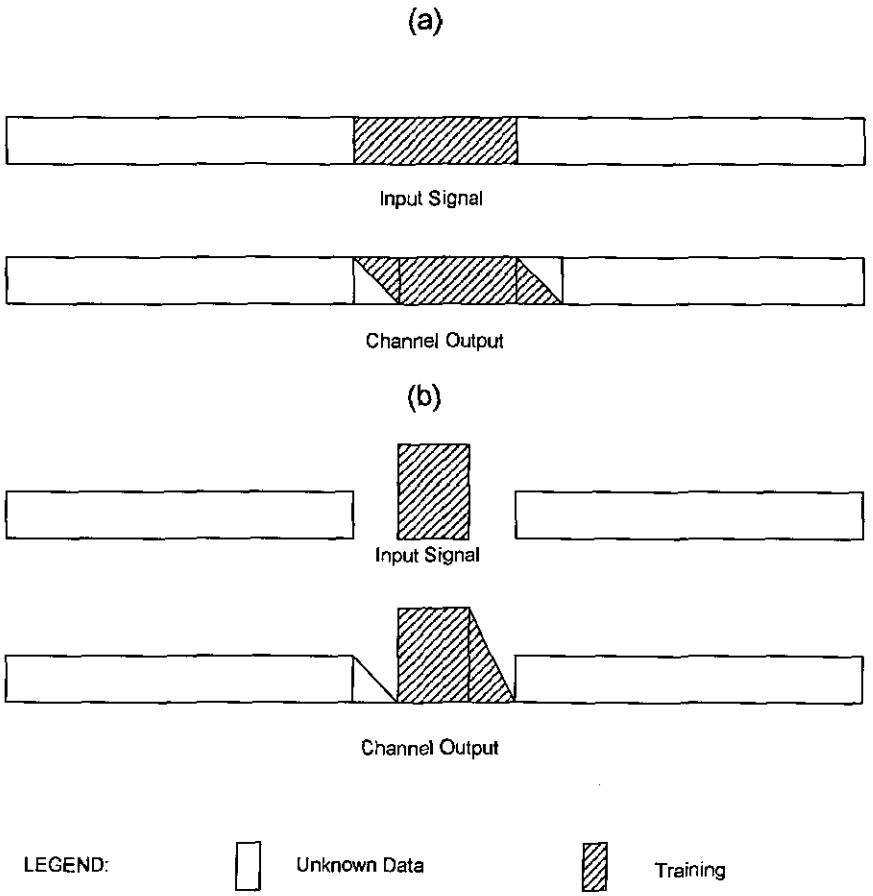

Figure 5.4. (a) Training and data transmission interfere with one another (b) No interference between training and data transmission.

Another result of [102] is that each training cluster should have $2L+1$ symbols with all the cluster energy concentrated in the $(L+1)^{th}$ symbol. Intuitively, estimation of the channel requires more than or equal to $2L+1$ symbols because of the frequency selective nature of the channel. Any increase in the number of pilot symbols in the training cluster will not help the channel estimate because the channel can be estimated with $2L+1$ symbols, but it will reduce the number of data transmissions before the training must

be transmitted again. The training cluster is thus simply an impulse. During each training cluster, the transmitter becomes silent to allow the channel transients due to the data transmission to die out before it sends an impulse to sound the channel. The transmitter remains silent till all transients due to the this impulse have died out at the receiver before resuming transmission of the data. It has also been shown by [102] that the optimal number of training clusters in a block is $P = Q + 1$. Again, more training clusters will only reduce the throughput and there will not be any improvement in the estimator performance.

Intuitively, the time selective channel is varying in time and therefore must be sampled periodically in time. Because the channel has no memory, this sampling is achieved by simply placing a single pilot at each sampling instant. The frequency selective channel varies with frequency and must therefore be sampled uniformly in frequency. For frequency domain transmissions, this can be achieved by employing uniformly spaced pilot tones for sampling the frequency response. For time domain transmissions, the channel may be measured once as it does not change. However, because of the channel's memory, the channel transients should be allowed to die out before the channel is sounded for time domain transmission and data transmission cannot resume until all transients have died out. The doubly selective channel is a frequency selective channel with time selective characteristics as well. Therefore it must be sampled periodically in time and each sampling must be preceded and succeeded by silent periods.

5.4.2 Iterative Estimation and Equalization

The doubly selective channel is estimated according to the optimal pilot design strategy discussed above. As opposed to superimposed training where the channel estimate is affected by interference from the unknown data as well as noise, the channel estimate is now perfectly decoupled from data transmission and is only affected by the noise. In the high SNR region,

this may result in sufficiently good channel estimates. However, in the low to moderate SNR region of operation of the turbo equalization algorithm, the channel estimate is insufficient for any significant performance gains. With these channel estimates, the iterative equalization and decoding scheme simply does not converge because the equalizer fails to undo the effects of the channel. The equalizer cannot even take advantage of the correct a priori information provided by the decoder because it generates and tries to cancel out an incorrect soft replica of the interference, thereby inadvertently passing and even introducing new interference while also providing fake reliability information to the decoder. The significant performance gains promised by turbo equalization thus fizzle out.

By introducing channel estimation into the iterative equalization and decoding loop, these gains can be recovered. In each iteration, the channel is re-estimated at the end of the decoding operation before proceeding to the next iteration. The channel estimate is thus updated in accordance with the information gleaned by the decoder. Now that the channel estimate has been updated to reflect the observations of the decoder in the iterative equalization and decoding loop, this allows the equalizer to exploit the a priori information provided by the decoder. The equalizer was unable to do so without the channel re-estimation because the channel estimate used by the equalizer was not consistent with the observations of the decoder. The channel estimate is thus updated with a new estimate formed from the soft information of the decoder and the received signal in each iteration before proceeding to the next iteration. A least squares estimator is used to form the new estimates. This allows the equalizer to exploit the a priori information from the decoder and generate reliable new extrinsic information for the decoder and the process iterates.

5.4.3 Simulation Results

We simulated a SISO system with a doubly selective channel with four taps and a Doppler spread of $f_d = 0.01$ normalized to the symbol rate. We used

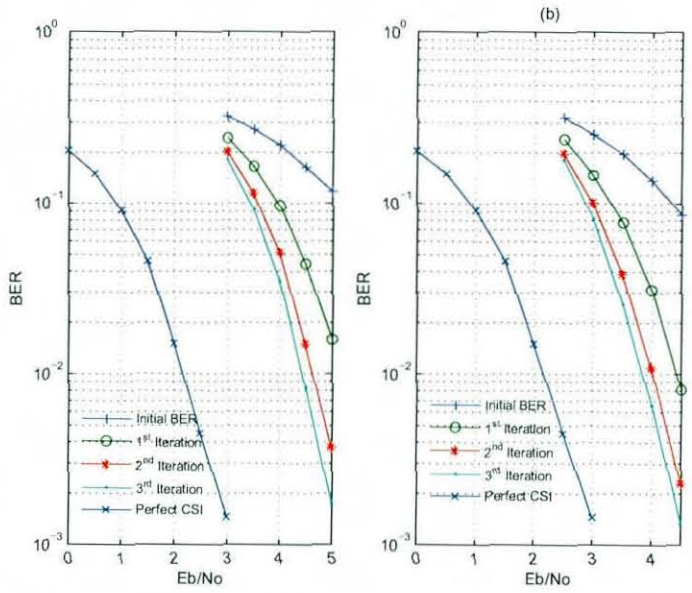

Figure 5.5. BER performance for various iterations of the algorithm when channel is generated using (a) Jakes fading model (b) the basis expansion model

block lengths of $N = 300$ so that $Q/2 = 3$ and used an equalizer of order $L_{eq} = 3$. Simulation results are provided in figures 5.5 and 5.6. Simulation results in figure 5.5 indicate that the penalty for lack of CSI is approximately 1.5 dB for channels generated with the complex exponential BEM. For channels generated by Jakes fading model, the penalty is approximately 2 dB. We note that the performance improvements seen are less than the performance improvements produced by iterative semiblind channel estimation based on superimposed training. This can be attributed to the better initial estimates formed from time multiplexed training as there is no interference from the unknown data.

The simulation results indicates that the performance improvements offered by turbo equalization can be achieved over a doubly selective channel

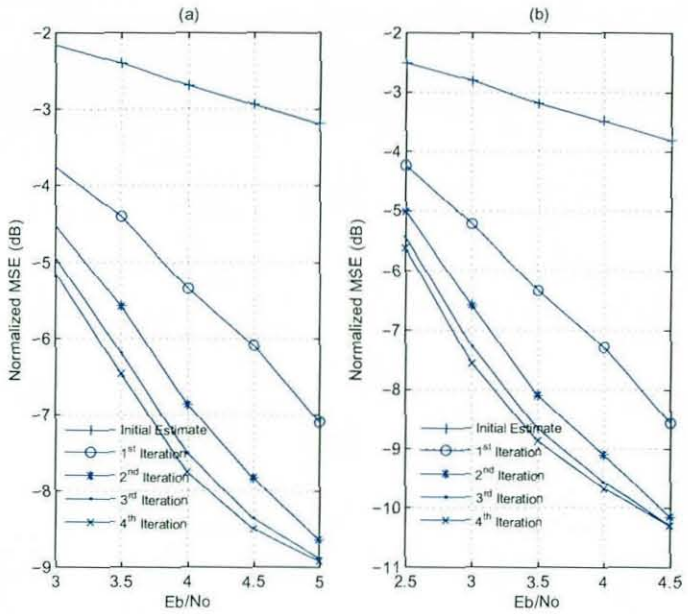

Figure 5.6. Normalized channel estimation mean square error for various iterations of the algorithm when channel is generated using (a) Jakes fading model (b) the basis expansion model

by introducing channel re-estimation in the iterative equalization and decoding loop. It is seen from figure 5.5 that most of the performance improvement is achieved in the first two or three iterations. This is because of the short block size of $N = 300$. For rapidly changing channels, channel estimation and turbo equalization have somewhat conflicting requirements. Turbo equalization favors long blocks so that the interleaver can suppress the correlations introduced by the forward backward algorithm in the samples. However, channel estimation favors small block lengths with fewer parameters to estimate. Another factor is that the BEM is a linear approximation that can only be locally valid for modeling physical time varying channels in a small window. Further, the equalization delay is not optimized for these channels because the channel taps are time varying and the optimal delay will also be time varying but we have chosen a fixed equalization delay throughout the block.

5.5 MIMO Doubly Selective Channels

We assume a zero prefixed block based transmission scheme so the observations may be expressed using a matrix vector model where the zero prefix of length L equal to the maximum order of the channel ensures IBI free transmission. The observations at the ν^{th} receive antenna are given by

$$\mathbf{y}_\nu = \sum_{\kappa=1}^{M_T} \mathbf{H}^{(\nu,\kappa)} \mathbf{x}_\kappa + \mathbf{v}_\nu \tag{5.5.1}$$

where we assume M_T transmit antennas. Stacking the received vectors at all the receive antennas we can write $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$. Because we process the received signal block by block, the block and serial indices are dropped. Each transmit block is partitioned into P sub blocks of training symbols b_κ and unknown data s_κ as

$$\mathbf{x}_\kappa = \begin{bmatrix} \mathbf{s}_{\kappa,1}^T & \mathbf{c}_{\kappa,1}^T & \cdots & \mathbf{s}_{\kappa,P}^T & \mathbf{c}_{\kappa,P}^T \end{bmatrix}^T \tag{5.5.2}$$

to decouple channel estimation and data detection. The sub blocks are of identical lengths with $\bar{N}_c = N_c/P$ and $\bar{N}_s = N_s/P$ representing the number of symbols in a training and data sub block so that $N_c + N_s = N$. The total energy in the block is divided as $\mathcal{P} = \mathcal{P}_c + \mathcal{P}_s$ where $\mathcal{P}_c$ is the energy allocated to training and $\mathcal{P}_s$ is the energy allocated to data transmission. Because of this structure of the transmit vector, it is possible to rearrange the observations at the ν^{th} receive antenna so that the observations that depend on training only are separated because these will be employed for channel estimation [103]

$$\begin{bmatrix} \mathbf{y}_{\nu,s} \\ \mathbf{y}_{\nu,c} \end{bmatrix} = \sum_{\kappa=1}^{M_T} \begin{bmatrix} \mathbf{H}_s^{(\nu,\kappa)} & \mathbf{H}_{c,s}^{(\nu,\kappa)} \\ \mathbf{0} & \mathbf{H}_c^{(\nu,\kappa)} \end{bmatrix} \begin{bmatrix} \mathbf{s}_\kappa \\ \mathbf{c}_\kappa \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{\kappa,s} \\ \mathbf{v}_{\kappa,c} \end{bmatrix} \tag{5.5.3}$$

where $\mathbf{y}_{\nu,s}$ captures the contribution of the unknown data and the interference due to the unknown data in the training while $\mathbf{y}_{\nu,c}$ contains the output

samples due to training that are free of any interference from the data. We can write

$$\mathbf{y}_{\nu,c} = \sum_{\kappa=1}^{M_T} \mathbf{\Xi}_\kappa \mathbf{h}^{\nu,\kappa} + \mathbf{v}_{\nu,c} = \mathbf{\Xi}\mathbf{h}_\nu + \mathbf{v}_{\nu,c} \tag{5.5.4}$$

where

$$\mathbf{\Xi}_\kappa = \begin{bmatrix} \mathbf{D}_{b,1,0}\mathbf{B}_{\kappa,1} & \cdots & \mathbf{D}_{b,1,Q}\mathbf{B}_{\kappa,1} \\ \vdots & \ddots & \vdots \\ \mathbf{D}_{b,P,0}\mathbf{B}_{\kappa,P} & \cdots & \mathbf{D}_{b,P,Q}\mathbf{B}_{\kappa,P} \end{bmatrix} \tag{5.5.5}$$

We may then collect the terms for all receive antennas as

$$\mathbf{y}_c = (\mathbf{I}_{M_R} \otimes \mathbf{\Xi})\mathbf{h} + \mathbf{v}_c \tag{5.5.6}$$

where $\otimes$ denotes the Kronecker product. This is identical in form to the relation for the SISO case. The Wiener solution for the channel estimate is given by [103]

$$\hat{\mathbf{h}} = (\sigma_v^2 \mathbf{I}_{M_R M_T (L+1)(Q+1)} + (\mathbf{I}_{M_R} \otimes \mathbf{\Xi}^H \mathbf{\Xi}))^{-1} (\mathbf{I}_{M_R} \otimes \mathbf{\Xi}^H)\mathbf{y}_c \tag{5.5.7}$$

and the mean square error of the estimator is

$$\sigma_{\tilde{h}}^2 = M_R \mathrm{tr}\big((\mathbf{I}_{M_T(L+1)(Q+1)} + \frac{1}{\sigma_v^2}\mathbf{\Xi}^H\mathbf{\Xi})^{-1}\big) \tag{5.5.8}$$

where tr() is the trace operator. This mean square error is lower bounded by

$$\sigma_{\tilde{h}}^2 \geq M_R \sum_m \frac{1}{[\mathbf{I}_{M_T(L+1)(Q+1)} + \frac{1}{\sigma_v^2}\mathbf{\Xi}^H\mathbf{\Xi}]_{m,m}} \tag{5.5.9}$$

with equality only if $\mathbf{\Xi}^H\mathbf{\Xi}$ is diagonal. The mean square error depends on the choice of training sequence through $\mathbf{\Xi}$. Similar to the SISO case, this mean square error is minimized when $\mathbf{\Xi}^H\mathbf{\Xi} = \bar{\mathcal{P}}_c\mathbf{I}$. By following a similar line of argument as for the SISO case, we have P training clusters transmitted for the estimation of $M_T(L+1)(Q+1)$ parameters. The first L samples in any training cluster are useless as they contain interference from data

transmission, so the number of pilots required is now

$$N_c \geq PL + M_T(Q+1)(L+1) \tag{5.5.10}$$

Under this condition and the assumption of a Gaussian source, it was shown in [102, 103] that the minimization of the mean square error also maximizes a tight lower bound on the training based capacity. Similar to the SISO case, the effective noise term will include the additive noise, interference from the data transmission due to channel errors and interference due to training. The effect of the noise can be minimized by choosing training so that it does not interfere with the data. So optimal training sequences are again an impulse with all the training sub block energy $\bar{\mathcal{P}}_c$ concentrated in a single position. Intuitively, the use of an impulse is advantageous as it ensures that all the energy in the training sequence is used for training as the observations corresponding to the first L pilots are corrupted by the unknown data and are rendered unusable, hence no energy is allocated to those pilots. Further, the last L pilots in the training sub block will interfere with the first L data samples in the next unknown data sub block, so they should also be silent. However, each of the P training clusters will consist of $M_T(L+1)+L$ samples. The first L zeros in the pilot will be for the channel transients to die out. Then the pilot transmitted by the κ^{th} transmitter will be

$$\begin{bmatrix} \mathbf{0}_{\kappa(L+1)-1} & \bar{\mathcal{P}}_c & \mathbf{0}_{(M_T-\kappa)(L+1)+L} \end{bmatrix} \tag{5.5.11}$$

This will ensure that the structure $\Xi^H\Xi = \bar{\mathcal{P}}_c\mathbf{I}$ is satisfied. It can be interpreted as sampling [102] of the channel in both the delay and Doppler dimensions to obtain the BEM coefficients. Only $P = Q+1$ training clusters will be required to form an estimate of the channel and any more will only decrease the throughput while providing negligible improvement in the channel estimate. This structure of the training sequence is depicted in fig-

ure 5.7 for two transmitters.

The MIMO system will train as M_T SIMO systems by virtue of the structure of the optimal training sequence. The transmitters will become silent for all channels and wait for the ISI to die out. Then the M_T transmitters will take turns training the receivers one by one. Each transmitter will remain silent and wait for its turn to train the receivers. A transmitter will train the receivers by transmitting an impulse once the transients from the previous transmitter have died out. Meanwhile, each receiver will be able to form an estimate of the channel from that transmitter.

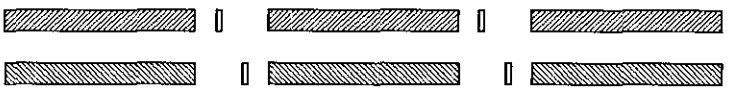

Figure 5.7. Structure of training sequence for two transmitters.

Apart from the choice of optimal training design, the power allocation between training and data also affects the estimator performance through $\mathcal{P}_c$. Although increasing $\mathcal{P}_c$ will result in less estimation MSE, it will also reduce the energy available for data transmission. This tradeoff was delineated in [102, 103] where the optimal power allocation for training based capacity was found in terms of the SNR and asymptotic power allocations were found for high SNR. While these choices of the training sequence, placement, and power allocation are indeed optimal, better MSE and BER performance can still be achieved if the unknown data can be incorporated into the channel estimation process. The data is detected and this detected data can be used to further improve on the channel estimate because of the finite alphabet property of practical data transmission systems. Once a linear or decision feedback equalizer is applied to the received signal, the signal is passed through a nonlinear decision device to quantize the equalized signal to the transmit constellation. This detected data can be treated as a virtual pilot, i.e. we assume that this was the sequence transmitted and use

Algorithm 3 Iterative Channel Estimation and Equalization and Decoding algorithm

1. Design training according to [103], multiplex it into the data and transmit.
2. Form an initial channel estimate $\hat{\mathbf{h}}$ according to equation (5.5.7).
3. Carry out an iteration of the space time turbo equalization algorithm using $\hat{\mathbf{h}}$.
4. Convert the LLRs generated by the decoder into soft estimates, interleave them and multiplex the known training to form a virtual pilot $\mathbf{v}$.
5. Form an estimate of $\boldsymbol{\Upsilon}$ using $\mathbf{v}$ and then solve for an estimate of the channel using the least squares approximation $\hat{\mathbf{h}} = (\mathbf{I}_{M_R} \otimes \boldsymbol{\Upsilon}^H\boldsymbol{\Upsilon})^{-1}(\mathbf{I}_{M_R} \otimes \boldsymbol{\Upsilon}^H)\mathbf{y}$.
6. Repeat steps 3 to 6.

it along with the received sequence to form a new channel estimate using a least squares estimator. In a coded system, the correction capability of a code can be used to iteratively improve on the channel estimation by using an estimate of the transmit signal generated by the decoder. Improvements on the channel estimate will result in correction of more errors in the equalization and decoding in the next iteration so the channel estimation and equalization iteratively propel one another until the process converges in a few iterations. We have considered such a coded system. The process is summarized in algorithm 3.

5.5.1 Simulation Results

We simulated a MIMO system with $M_T = 2$ transmit antennas, $M_R = 3$ receive antennas and a channel order of $L = 2$. The space time equalizer at the receiver had an order $L_{eq} = 2$. We assumed a block length of $N = 300$ and a Doppler spread of $f_d = 0.01$ normalized to the symbol rate so that $Q/2 = 3$. The simulation results for the BER and normalized estimation MSE are provided in figures 5.8 and 5.9 respectively. We see that similar

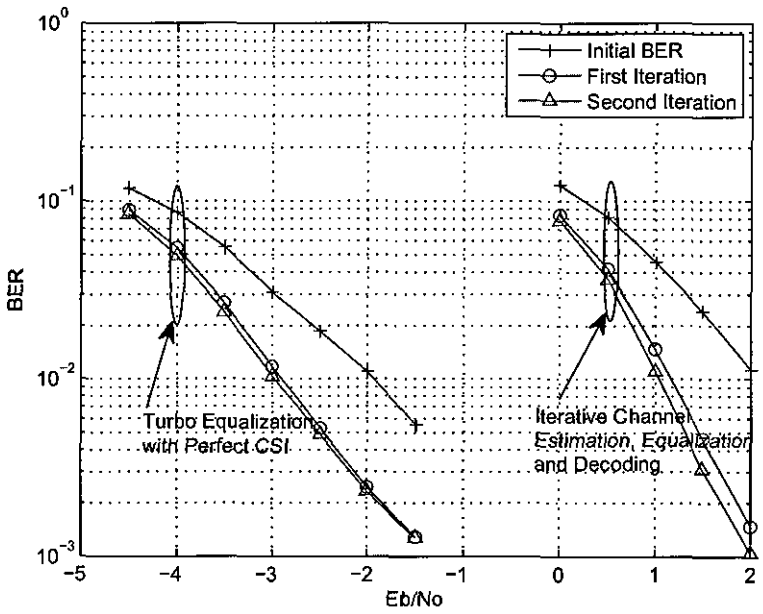

Figure 5.8. The BER is plotted against Eb/No for various iterations of the algorithm and also for turbo equalization with perfect CSI

to space time turbo equalization with perfect CSI, performance improvements are observed with the use of iterative equalization and decoding in a trained system that uses time multiplexed training if the channel estimation is included in the iterative equalization and decoding loop. However, the performance losses over a system with perfect CSI are larger. There is approximately a 4 dB loss at a BER of 10^{-2} in addition to the loss in throughput incurred by the transmission of the time multiplexed pilots. A performance improvement of approximately 1 dB is observed at a BER of 10^{-2} in figure 5.8 for both turbo equalization and iterative channel estimation, equalization and decoding. It is seen that iterative channel estimation, equalization and decoding only provides performance improvement over a couple of iterations because of the small block size used.

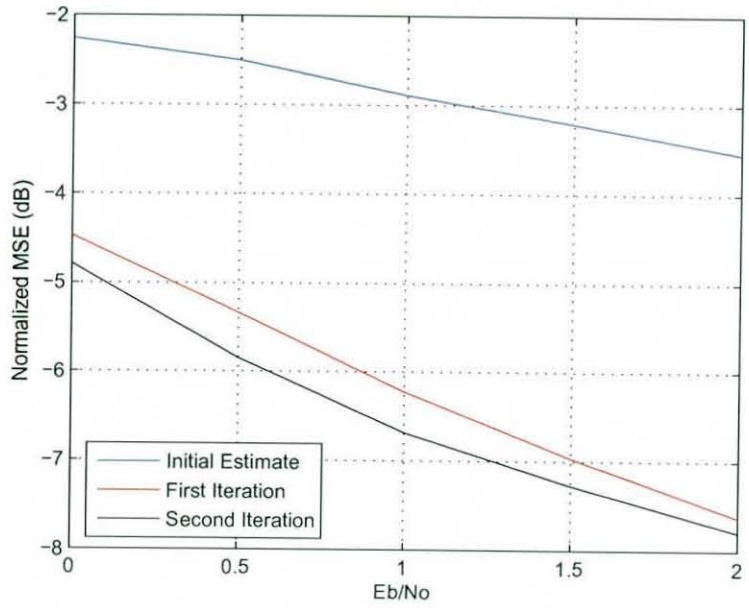

Figure 5.9. The estimation mean square error is plotted against the Eb/No for the iterations

5.6 Conclusion

Simulation results indicate that in the low SNR region of operation of the turbo equalization algorithm, the quality of the channel estimates for doubly selective channels is insufficient to provide the gains expected from the turbo equalization algorithm even with optimally designed time multiplexed training where the estimates are not affected by interference from the data transmission. However with introduction of channel estimation in the iterative equalization and decoding loop, the turbo equalization iterations can provide significant performance gains.

Chapter 6

SUMMARY, CONCLUSION AND FUTURE WORK

When a message is transmitted over a communications channel, the received signal has undergone distortion because of several unwanted effects of the channel including interference, time variations, and thermal noise. The received signal requires signal processing to deal with these distortive effects of the channel before the transmit information can be extracted from it. This receiver processing cannot increase the information about the transmit message in the received signal, it only brings the received signal in format that is recognizable by the receiver. Although the received signal contains the transmit information, it is not in a form that can be used to detect this information until such processing has been performed. This signal processing needed to mitigate the unwanted effects of the channel requires knowledge of how the channel distorts the signal, i.e. it needs to know and track the channel. It is pertinent to mention here that even if the receiver is perfectly armed with the CSI, such receiver processing does not guarantee that the transmit signal can always be recovered.

Most literature assumes that the receiver has perfect CSI and thus overlooks this problem. The channel can be estimated through the transmission of a pilot signal or blindly. Blind schemes use some statistical or other information about the transmit signal that is known a priori. For trained schemes, the pilot signal is known a priori to the receiver and it is used to estimate the

channel from the received signal. The pilots may be multiplexed with the unknown data or they may be superimposed on it, i.e. added onto it with a low power, or any combination of these schemes may be used. However, it is often desirable in engineering applications to make the channel estimation process as computationally simple as possible. These training based channel estimation schemes have been studied in this thesis.

Superimposed training was introduced first and applied to OFDM in chapter 3 for frequency selective channels. The general case where the channel delay spread exceeds the cyclic prefix has been considered and the channel shortening equalizers are explained. Superimposed training in multicarrier communications is studied for both SIMO OFDM and MIMO OFDM. For the MIMO OFDM system, we have explained MIMO channel shortening and studied ST coding for spatial multiplexing. The novelty of our work is that we have proposed the use of semiblind iterative channel estimation based on superimposed training for OFDM.

The focus of channel estimation research has recently shifted to doubly selective channels in which the transmit signal experiences fading in both the time and frequency dimensions. Because of the fast varying nature of these channels, they are difficult to track and require continual training. The superimposed training scheme was studied for single carrier communications in chapter 4 for doubly selective channels. A coded transmission system was considered where channel estimation was introduced in the iterative equalization and decoding loop. The cases of SISO and MIMO for spatial multiplexing were considered and it was shown that an iterative approach that allows the inclusion of detected data in channel estimation significantly enhances the performance over a non iterative scheme. The channel estimate formed by superimposed training suffers from interference from the unknown data in addition to the noise effects and exhibits poor performance because of an error floor. By inclusion of the detected data in channel re-estimation in each iteration, the performance is considerably enhanced over a few it-

erations. Turbo equalization along with superimposed training for doubly selective channels has not been studied previously in the literature.

Time multiplexed training was introduced for doubly selective channels in chapter 5 for SISO and MIMO spatial multiplexing. It was observed that the channel estimate formed from an optimal design of pilots in the low to moderate SNR region is insufficient for the turbo equalization process that operates in this SNR region. The turbo equalization fails to make any significant performance improvements over the iterations when this initial channel estimate is employed to design the equalizer. However, if the detected data is reused in conjunction with the received signal to re-estimate the channel BEM coefficients in each iteration and the equalizer is updated with the new estimates, the turbo equalization algorithm shows similar performance improvements as it does for perfect CSI. The novelty of this work is in the use of iterative equalization and decoding for BEM parameterized doubly selective channel estimation with time multiplexed training.

As a possible extension, it would be interesting to study the application of robust estimation techniques to the channel estimation problem. For time multiplexed training, the least squares estimate of the channel is the maximum likelihood estimate because the noise is independent and identically distributed Gaussian and the communications model is linear. However, it has been stressed before that when superimposed training is employed, the effective noise term includes an interference term along with the noise which may not Gaussian. It would be an interesting problem to study the effective noise and possibly characterize its statistics which may lead to a better estimator than the least squares estimator. For example it is well known that the least squares estimator is very sensitive to outliers. This is of course not a problem when the noise is Gaussian because the Gaussian distribution is mostly clustered within three standard deviations and outliers are unlikely to occur anyway. However, incase the noise distribution is less clustered like the Laplacian distribution, outliers are very likely to occur and it is more

sensible to apply the l_1 norm minimization instead

$$\text{minimize} \quad \|\mathbf{Xh} - \mathbf{y}\|_1 \tag{6.0.1}$$

This is minimized as a linear program by introducing a slack variable t [104]

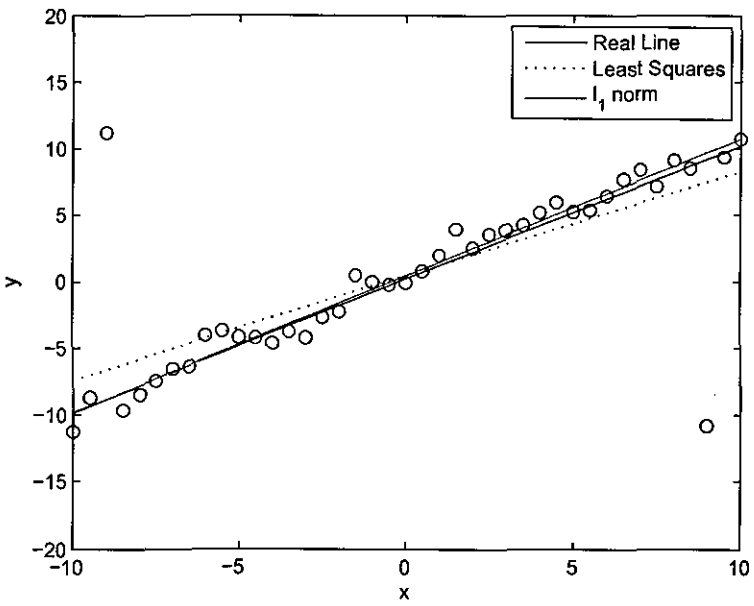

Figure 6.1. Comparison of l_1 and l_2 norm approximation.

$$\text{minimize} \quad t$$

$$\text{subject to} \quad -t\mathbf{1} \leq \mathbf{Xh} - \mathbf{y} \leq t\mathbf{1}$$

In fact the l_1 norm approximation is known to be robust estimation because it is relatively insensitive to outliers. It is also the maximum likelihood estimator if the noise distribution is Laplacian instead of Gaussian[1]. In figure 6.1, we have taken observations on a straight line $y = mx + c$ and added

[1] In practical applications noise is typically Gaussian so that the l_2 norm approximation is more accurate. If we know that the noise is Gaussian and we want to be careful about outliers which are introduced by some other mechanism, then we could also use the Huber penalty function which approximates l_2 norm approximation for small residuals and l_1 norm approximation for large residuals [104].

some noise. Two outliers are also introduced in the observations and then we perform l_1 and l_2 norm regressions to estimate the parameters of the line. It is clearly seen that l_1 norm is robust to the outliers.

As we already know, the least squares estimate of this problem is $\hat{\mathbf{h}} = (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{y}$. In both superimposed training and time multiplexed training, when an iterative channel estimate is formed based on the detected data, there is always an uncertainty in the matrix $\mathbf{X}$ because of detection errors, particularly at low signal to noise ratios. The estimation problem where we wish to minimize $\|\mathbf{X}\mathbf{h} - \mathbf{y}\|$ thus suffers from an incorrect model because $\mathbf{X}$ is not known perfectly and it may vary somewhat around our estimate. It would be interesting to study the effects this model uncertainty, especially the performance losses incurred because of these detection errors and how much of these losses can be salvaged through a more robust scheme. The application of robust approximation techniques to this problem could be studied. Robust approximation problems are divided into two categories, i.e. stochastic robust approximation and worst case robust approximation. Stochastic robust approximation assumes that X is a random variable around a mean $\bar{\mathbf{X}}$ so that [104]

$$\mathbf{X} = \bar{\mathbf{X}} + \mathbf{U}$$

where $\mathbf{U}$ is a zero mean random matrix that describes the statistical variation of $\mathbf{X}$ around $\bar{\mathbf{X}}$. Stochastic robust approximation then tries to

$$\text{minimize} \quad E\|\mathbf{X}\mathbf{h} - \mathbf{y}\| \tag{6.0.2}$$

In case the system model matrix $\mathbf{X}$ assumes a finite set of values as in our case where the elements of $\mathbf{X}$ are constellation points, this can be written as

$$\text{minimize} \quad p_1\|\mathbf{X}_1\mathbf{h} - \mathbf{y}\| + \cdots + p_k\|\mathbf{X}_k\mathbf{h} - \mathbf{y}\| \tag{6.0.3}$$

where $\mathbf{X}$ assumes value $\mathbf{X}_i$ with probability p_i. Then this problem can be written as

$$\text{minimize} \quad \mathbf{p}^T\mathbf{t} \tag{6.0.4}$$

$$\|\mathbf{X}_i\mathbf{h} - \mathbf{y}\| \leq t_i \tag{6.0.5}$$

which may be solved as a linear program or a second order cone program depending on the choice of the norm [104]. Similarly, worst case robust approximation tries to minimize the worst case error. Assuming that $\mathbf{X} \in \mathcal{X}$, then the associated worst case error is

$$e_{wc}(\mathbf{h}) = \sup\{\|\mathbf{X}\mathbf{h} - \mathbf{y}\| \mid \mathbf{X} \in \mathcal{X}\} \tag{6.0.6}$$

In a study of the application of robust estimation and approximation techniques to iterative channel estimation, an important issue that needs consideration is a comparison of the performance gains achieved and the computational complexity involved, in order to assess the feasibility of such schemes in the channel estimation problem.

A related future research direction is the study and design of robust equalizers for communications. Regardless of the channel estimation algorithm employed, there will always be a channel estimation error. Typical equalizer designs including the equalizers we used assume a perfect channel estimate which is clearly unavailable. In a practical communications scheme where the channel estimate will have errors, it is wiser to design a robust equalizer which is relatively insensitive to small variations of the channel around the estimate. Some work has already been started in this direction by [105].

Bibliography

[1] A. J. Paulraj and C. B. Papadias, "Space time processing for wireless communications," *IEEE Signal Processing Magazine*, vol. 14, pp. 49–82, Nov. 1997.

[2] J. M. Cioffi, "Mining Copper and Ether," *IEEE Communications Magazine*, pp. 18–20, June 2007.

[3] C. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 623–656, 1948.

[4] T. K. Moon, *Error Correction Coding.* John Wiley, 2005.

[5] S. T. Brink, "Coding Over Space and Time for Wireless Systems," *IEEE Wireless Communications*, vol. 13, pp. 18–30, Aug. 2006.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon-Limit error correction coding and decoding: Turbo Codes," *IEEE Conference on Communications*, pp. 1064–1070, May 1993.

[7] B. Vucetic and J. Yuan, *Turbo Codes.* Kluwer Academic Publishers, 2000.

[8] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727–1737, Oct. 2001.

[9] M. Tuchler, R. Koetter, and A. Singer, "Turbo equalization principles and new results," *IEEE Transactions on Communications*, vol. 50, pp. 754–767, May 2002.

[10] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space Time Wireless Communications.* Cambridge University Press, 2003.

[11] R. W. Lucky, "Generalized automatic equalization for communication channels," *Proceedings of IEEE,* vol. 54, pp. 439–440, March 1966.

[12] Z. Wang and G. B. Giannakis, "Wireless multicarrier communications," *IEEE Signal Processing Magazine,* vol. 17, pp. 29–48, May 2000.

[13] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications.* Cambridge University Press, 2005.

[14] J. G. Proakis, *Digital Communications.* Mc Graw Hill, 1995.

[15] J. R. Treichler, "Fractionally spaced equalizers," *IEEE Signal Processing Magazine,* vol. 13, pp. 65–81, May 1996.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete Time Signal Processing.* Prentice Hall, 2003.

[17] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimim decoding algorithm," *IEEE Transactions on Information Theory,* vol. 13, pp. 260–269, April 1967.

[18] D. Falconer and F. R. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell System Technical Journal,* pp. 1541–1562, Nov 1973.

[19] N. Al-Dhahir and J. M. Cioffi, "Efficiently computed reduced parameter input aided MMSE equalizers for ML detection," *IEEE Transactions on Information Theory,* vol. 42, pp. 903–915, May 1996.

[20] J. Cioffi, G. P. Dudevoir, M. Eyuboglu, and G. D. Forney, "MMSE decision-feedback equalizers and coding. I. Equalization results," *IEEE Transactions on Communications,* vol. 43, pp. 2582–2594, Oct. 1995.

[21] J. Cioffi, G. P. Dudevoir, M. Eyuboglu, and G. D. Forney, "MMSE decision-feedback equalizers and coding. II. Coding results," *IEEE Transactions on Communications*, vol. 43, pp. 2595–2604, Oct. 1995.

[22] N. Al-Dhahir and J. M. Cioffi, "MMSE Decision Feedback Equalizers: Finite Length Results," *IEEE Transactions on Information Theory*, vol. 41, pp. 961–975, July 1995.

[23] M. Tuchler, A. Singer, and R. Koetter, "Minimum mean square error equalization using a priori information," *IEEE Transactions on Signal Processing*, vol. 50, pp. 673–683, March 2002.

[24] R. Koetter, A. Singer, and M. Tuchler, "Turbo equalization," *IEEE Signal Processing Magazine*, vol. 21, pp. 67–80, Jan. 2004.

[25] S. ten Brink, "Iterative demapping for QPSK modulation," *Electronics Letters*, vol. 34, pp. 1459–1460, July 1998.

[26] M. L. Moher, "Turbo based multiuser detection," *Proc. IEEE International Symposium on Information Theory*, June 1997.

[27] H. V. Poor, "Iterative multiuser detection," *IEEE Signal Processing Magazine*, vol. 21, pp. 81–88, Jan. 2004.

[28] D. Raphaeli and Y. Zarai, "Combined turbo equalization and turbo decoding," *IEEE Communications Letters*, vol. 2, pp. 107–109, Apr. 1998.

[29] S. Haykin, ed., *Unsupervised Adaptive Filtering: Blind Deconvolution.* Englewood Cliffs.

[30] C. R. Johnson, "Blind equalization using the constant modulus criterion: A review," *Proceedings of IEEE*, vol. 86, pp. 1927–1950, Oct. 1998.

[31] A. Benveniste, M. Ghoursat, and G. Ruget, "Robust identification of a non minimum phase system: Blind adjustment of a linear equalizer in

data communication," *IEEE Transactions Automated Control*, vol. AC-25, pp. 385–399, June 1980.

[32] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems," *IEEE Transactions on Information Theory*, vol. 36, pp. 312–321, March 1990.

[33] C. B. Papadias, "Globally convergent blind source separation based on a multiuser kurtosis maximization criterion," *IEEE Transactions on Signal Processing*, vol. 48, pp. 3508–3519, December 2000.

[34] R. K. Martin and C. R. Johnson, "Adaptive equalization: Transitioning from single carrier to multicarrier systems," *IEEE Signal Processing Magazine*, vol. 22, pp. 108–122, Nov. 2005.

[35] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single carrier broadband wireless systems," *IEEE Communications Magazine*, vol. 40, pp. 58–66, Apr. 2002.

[36] L. M. S. C. of the IEEE, "Wireless LAN Physical (PHY) Layer Specifications," *IEEE Std. 802.11.*

[37] L. M. S. C. of the IEEE, "Wireless MAN Physical (PHY) Layer Specifications," *IEEE Std. 802.16.*

[38] L. M. S. C. of the IEEE, "Mobile Broadband Wireless Access Physical (PHY) Layer Specifications," *IEEE Std. 802.20.*

[39] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Transactions on Communications*, vol. 44, pp. 1662–1672, Dec. 1996.

[40] N. Al-Dhahir and J. M. Cioffi, "Efficiently computed reduced parameter input aided MSE equalizers for ML detection: A unified approach," *IEEE Transactions on Information Theory*, vol. 42, pp. 903–915, May 1996.

[41] N. Al-Dhahir and J. M. Cioffi, "Optimum finite length equalization for multicarrier transceivers," *IEEE Transactions on Communications*, vol. 44, pp. 56–64, Jan. 1996.

[42] G. Arslan, B. L. Evans, and S. Kiaei, "Equalization for discrete multitone transceivers to maximize bit rate," *IEEE Transactions on Signal Processing*, vol. 49, pp. 3123–3135, Dec. 2001.

[43] D. Daly, C. Hanegan, and A. D. Fagan, "Minimum mean squared error impulse respose shortening for discrete multitone transceivers," *IEEE Transactions on Signal Processing*, vol. 52, pp. 301–306, Jan. 2004.

[44] H. Zamiri-Jafarian, H. Khoshbin, and S. Pasupathy, "Time Domain Equalizer for OFDM Systems Based on SINR Maximization," *IEEE Transactions on Communications*, vol. 53, pp. 924–929, June 2005.

[45] N. Al-Dhahir and J. M. Cioffi, "FIR Channel Shortening Equalizers for MIMO ISI Channels," *IEEE Transactions on Communications*, vol. 49, pp. 213–218, Feb. 2001.

[46] K. V. Acker, G. Leus, M. Moonen, O. van de Viel, and T. Pollet, "Per tone equalization for DMT-based systems," *IEEE Transactions on Communications*, vol. 49, pp. 109–119, Jan. 2001.

[47] G. Leus and M. Moonen, "Per Tone Equalization for MIMO OFDM Systems," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2965–2975, Nov. 2003.

[48] R. K. Martin, J. Balakrishnan, W. Sethares, and C. R. Johnson, "A Blind Adaptive TEQ for Multicarrier Systems," *IEEE Signal Processing Letters*, vol. 9, pp. 341–343, Nov. 2002.

[49] R. K. Martin, "Fast converging blind adaptive channel shortening and frequency domain equalization," *IEEE Transactions on Signal Processing*, vol. 55, pp. 102–110, Jan. 2007.

[50] J. Balakrishnan, R. K. Martin, and C. R. Johnson, "Blind Adaptive Channel Shortening by Sum Squared Auto-Correlation Minimization (SAM)," *IEEE Transactions on Signal Processing*, vol. 51, pp. 3086–3093, Dec. 2003.

[51] R. Negi and J. Cioffi, "Pilot tone selection for channel estimation in mobile OFDM system," *IEEE Transactions on Consumer Electronics*, vol. 44, pp. 1122–1128, Aug. 1998.

[52] X. Ma, R. J. Baxley, J. Kleider, and G. T. Zhou, "Superimposed training for channel shortening equalization in ofdm," *Military Communications Conference (MILCOM)*, pp. 1–7, Oct. 2006.

[53] A. G. Orozco-Lugo, M. M. Lara, and D. C. McLernon, "Channel estimation using implicit training," *IEEE Transactions on Signal Processing*, vol. 52, pp. 240–254, Jan. 2004.

[54] S. M. Kay, *Fundamentals of Statistical Signal Processing - Estimation Theory*, vol. 1 of *Prentice Hall Signal Processing Series*. Prentice Hall, 1993.

[55] G. Strang, *Linear Algebra and its Applications*. Saunders College Publishing, 3 ed., 1986.

[56] X. Ma, L. Yang, and G. B. Giannakis, "Optimal Training for MIMO Frequency Selective Fading Channels," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 453–466, March 2005.

[57] G. Foschini and M. Gans, "On limits of wireless communications in a fading environment when using multiple antennnas," *Wireless Personal Communications*, pp. 311–335, March 1998.

[58] B. Hassibi, "An efficient square-root algorithm for BLAST," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 737–740, June 2000.

[59] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer, "Efficient algorithm for decoding layered space time codes," *Electronics Letters*, vol. 37, pp. 1348–1350, Oct. 2001.

[60] R. Bohnke, D. Wubben, V. Kuhn, and K. D. Kammeyer, "Reduced Complexity MMSE Detection for BLAST Architectures," *Global Telecommunications Conference*, pp. 2258–2262, Dec. 2003.

[61] S. ten Brink, "Coding over space and time for wireless systems," *IEEE Wireless Communications*, vol. 13, pp. 18–30, Aug. 2006.

[62] L. Tong, B. M. Sadler, and M. Dong, "Pilot assisted wireless transmissions," *IEEE Signal Processing Magazine*, vol. 21, pp. 12–25, Nov. 2004.

[63] N. Seshadri, "Joint data and channel estimation using blind trellis search techniques," *IEEE Transactions on Communications*, vol. 42, pp. 1000–1011, Feb./Mar./Apr. 1994.

[64] L. M. Davis, I. Collings, and P. Hoeher, "Joint MAP Equalization and Channel Estimation for Frequency Selective and Frequency Flat Fast Fading Channels," *IEEE Transactions on Communications*, vol. 49, pp. 2106–2114, Dec. 2001.

[65] L. Tong, G. Xu, and T. Kailath, "Blind identification and equalization based on second order statistics: A time domain approach," *IEEE Transactions on Information Theory*, vol. 40, pp. 340–349, March 1994.

[66] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least squares aproach to blind channel identification," *IEEE Transactions on Signal Processing*, vol. 43, pp. 2982–2993, Dec. 1995.

[67] L. Tong, G. Xu, B. Hassibi, and T. Kailath, "Blind channel identification based on second order statistics: A frequency domain aproach," *IEEE Transactions on Information Theory*, vol. 41, pp. 329–334, Jan. 1995.

[68] B. Farhang-Boroujeny, "Pilot based channel identification: Proposal for semi-blind identification of communications channels," *Electronics Letters*, vol. 31, pp. 1044–1046, June 1995.

[69] F. Mazzenga, "Channel Estimation and Equalization for M-QAM transmission with a hidden pilot sequence," *IEEE Transactions on Broadcasting*, vol. 46, pp. 170–176, June 2000.

[70] G. T. Zhou, M. Viberg, and T. McKelvey, "A first-order statistical method for channel estimation," *IEEE Signal Processing Letters*, vol. 10, pp. 57–60, March 2003.

[71] J. K. Tugnait and W. Luo, "On channel estimation using superimposed training and first order statistics," *IEEE Communications Letters*, vol. 7, pp. 413–415, Sept 2003.

[72] M. Ghogho, D. McLernon, E. A. Hernandez, and A. Swami, "Channel estimation and symbol detection for block transmission using data dependant superimposed training," *IEEE Signal Processing Letters*, vol. 12, pp. 226–229, March 2005.

[73] S. He, J. K. Tugnait, and X. Meng, "On Superimposed Training for MIMO Channel Estimation and Symbol Detection," *IEEE Trans. on Signal Processing*, vol. 55, pp. 3007–3021, June 2007.

[74] J. Tugnait, S. He, and X. Meng, "On superimposed training power allocation for time varying channel estimation," *2005 IEEE /SP 13th Workshop on*, pp. 17–20, July 2005.

[75] J. Tugnait and S. He, "Doubly selective channel estimation using data dependant superimposed training and exponential basis models," *IEEE Trans. on Wireless Communications*, vol. 6, pp. 3877–3883, Nov. 2007.

[76] S. He and J. Tugnait, "Self interference supression in doubly selective

channel estimation using superimposed training," *IEEE International Conference on Communications*, pp. 3028–3033, June 2007.

[77] X. Meng and J. K. Tugnait, "Doubly Selective MIMO Channel Estimation Using Superimposed Training," *2004 IEEE Sensor Array and Multichannel Signal Processing Workshop*, pp. 407–411, July 2004.

[78] C. Tepedelenlioglu and G. B. Giannakis, "Basis expansion models and diversity techniques for blind identification and equalization of time varying channels," *Proceedings of IEEE*, vol. 86, pp. 1969–1986, Nov. 1998.

[79] T. Zemen and C. Mecklenbauker, "Time variant cannel estimation using discrete prolate spheriodal sequences," *IEEE Transactions on Signal Processing*, vol. 53, pp. 3597–3606, Sept. 2005.

[80] I. Barhumi, G. Leus, and M. Moonen, "Time Varying FIR Equalization for Doubly Selective Channels," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 202–214, Jan. 2005.

[81] I. Barhumi, G. Leus, and M. Moonen, "Time Varying FIR Decision Feedback Equalization of Doubly Selective Channels," *Global Telecommunications Conference*, vol. 4, pp. 2263–2268, Dec. 2003.

[82] J. Tugnait and S. He, "Direct FIR linear equalization of doubly selective channels based on superimposed training," *International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. IV589–592, May 2006.

[83] G. Leus, I. Barhumi, and M. Moonen, "Direct Semi-Blind Design of Serial Linear Equalizers for Doubly Selective Channels," *IEEE International Conference on Communications*, vol. 5, pp. 2626–2630, June 2004.

[84] I. Barhumi, G. Leus, and M. Moonen, "Time Domain Channel Shortening and Equalization of OFDM over Doubly Selective Channels," *International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 17–21, May 2004.

[85] I. Barhumi, G. Leus, and M. Moonen, "Equalization of OFDM Over Doubly Selective Channnels," *IEEE Transactions on Signal Processing*, vol. 54, pp. 1445–1458, Apr. 2006.

[86] H. Liu and G. B. Giannakis, "Deterministic approaches for blind equalization of time varying channels with antenna arrays," *IEEE Transactions on Signal Processing*, vol. 46, pp. 3003–3013, Nov. 1998.

[87] A. Cano, X. Ma, and G. B. Giannakis, "Block-differential modulation over doubly selective wireless fading channels," *IEEE Transactions on Communications*, vol. 53, pp. 2157–2166, Dec. 2005.

[88] X. Meng, J. Tugnait, and S. He, "Iterative joint channel estimation and data detection using superimposed training: Algorithms and performance analysis," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 1873–1880, July 2007.

[89] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.

[90] A. Paulraj, D. Gore, R. Nabar, and H. Bolcskei, "An Overview of MIMO Communications - A key to Gigabit Wireless," *Proceedings of the IEEE*, vol. 92, pp. 198–218, Feb. 2004.

[91] G. J. Foschini, "Layered space time architecture for wireless communication in a fading environment when using multi-element antennas," *BELL Labs Technical Journal*, pp. 41–59, Oct. 1996.

[92] L. Zheng and D. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple antenna channels," *IEEE Transactions on Information Theory*, vol. 49, pp. 1073–1096, May 2003.

[93] T. Abe and T. Matsumoto, "Space Time Turbo Equalization in Fre-

quency Selective MIMO Channels," *IEEE Trans. on Veh. Tech.*, vol. 52, pp. 469–475, May 2003.

[94] J. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 686–693, July 1995.

[95] B. Hassibi and B. Hochwald, "How much training is needed in multiple antenna wireless links," *IEEE Transactions on Information Theory*, vol. 49, pp. 951–963, April 2003.

[96] R. Gallagher, *Information Theory and Reliable Communications.* John Wiley, 1968.

[97] S. Adireddy, L. Tong, and H. Viswanathan, "Optimal placement of training for frequency selective block fading channels," *IEEE Transactions on Information Theory*, vol. 48, pp. 2338–2353, August 2002.

[98] S. Adireddy and L. Tong, "Optimal placement of known symbols for nonergodic broadcast channels," *Conference on Information Sciences and Systems, Princeton University*, March 2002.

[99] S. Adireddy and L. Tong, "Optimal Placement of Known Symbols for Slowly Varying Frequency Selective Channels," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 1292–1296, July 2005.

[100] M. Dong, L. Tong, and B. M. Sadler, "Optimal Insertion of Pilot Symbols for Transmissions Over Time Varying Flat Fading Channels," *IEEE Transactions on Signal Processing*, vol. 52, pp. 1403–1418, May 2004.

[101] H. Vikalo, B. Hassibi, and B. Hochwald, "On the capacity of frequency selctive channels in training based transmission schemes," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2572–2583, Sept. 2004.

[102] X. Ma, G. B. Giannakis, and S. Ohno, "Optimal training for block transmissions over doubly selective wireless fading channels," *IEEE Transactions on Signal Processing*, vol. 51, pp. 1351–1366, May 2003.

[103] L. Yang, X. Ma, and G. B. Giannakis, "Optimal Training for MIMO Fading Channels with Time and Frequency Selectivity," *International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 821–824, May 2004.

[104] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge Univeristy Press, 2004.

[105] Y. Guo and B. Levy, "Robust MSE Equalizer Design for MIMO Communications Systems in the Presence of Model Uncertainties," *IEEE Transactions on Signal Processing*, vol. 54, pp. 1840–1852, May 2006.