TURBO-EQUALIZATION OF PULSE SHAPED MULTI-CARRIER MODULATION IN DOUBLY SELECTIVE CHANNELS

A Thesis

Presented in Partial Fulfillment of the Requirements for

the Degree Master of Science in the

Graduate School of The Ohio State University

By

Sibasish Das, B. Tech.

* * * * *

The Ohio State University

2004

Master's Examination Committee:

Dr. Philip Schniter, Adviser

Dr. Oscar Y. Takeshita

Approved by

Adviser

Department of Electrical and Computer Engineering

To my parents Anusree and Achintya Das, and my sister Sohini.

ACKNOWLEDGMENTS

I have spent two inspiring years at The Ohio State University pursuing a master's degree. First and foremost, this thesis would not have taken shape without the constant motivation and guidance of my advisor, Prof. Philip Schniter. I am thankful to him for the innumerable things that I have learned from him. I would like to thank all the people I have interacted with at The Ohio State University, specifically everyone associated with the Information Processing Systems Laboratory in the Department of Electrical and Computer Engineering. Finally, I would like to acknowledge NSF and Motorola, Inc. the sponsors of the work presented in this thesis. This material is based upon work supported by the National Science Foundation under Grant No. 0237037.

ABSTRACT

There is a growing demand for higher data rate systems that can function in a highly mobile environment. This mandates the design of communication systems that can function in doubly selective channels. Pulse-shaped multi-carrier modulation schemes prove to be an attractive option for transmission over doubly-dispersive channels. The pulse shapes are designed to yield an inter-symbol/inter-carrier interference profile matching a given target response. The receiver relies on high-performance/low-complexity equalizers that can reliably extract the transmitted symbols from the observations in the presence of controlled amounts of interference in the target response.This thesis presents two such high-performance/low-complexity iterative equalizers. Specifically, the two iterative equalization algorithms are based on minimum mean squared error and maximum likelihood criteria, respectively. In order to protect the transmitted information against sub-carrier nulls, an error control code is used at the transmitter. The equalizers exchange soft information with a maximum *a-posteriori* probability (MAP) optimal decoder in a turbo-like fashion at the receiver. Simulations suggest that turbo-equalization with linear complexity iterative equalizers offer significant performance enhancements over standard techniques.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

Introduction

Broadband wireless communication has proved to be a cornerstone of today's global information infrastructure. Successful communication over a wireless link entails overcoming two main hurdles.

The first is posed by the effects of multi-path propagation [1]. In simple terms, reflections from physical objects produce multiple versions of the same signal at the receiver. The cumulative effect at the receiver is a signal composed of various time and phase delayed echoes. When the delay spread of this received signal exceeds the duration of transmitted signals, energy from each transmitted symbol spills over and contaminates neighboring symbols. This effect is commonly known as inter-symbol interference (ISI). Such channels are referred to as frequency selective channels, since the channel response is not uniform over the bandwidth of the transmitted signal.

The second hurdle to be overcome in a wireless communication system is induced by mobility of the communicating devices. As a result of motion, time variations are introduced in the wireless communication channel. Motion results in Doppler shifts in frequency of the transmitted signal at the receiver. For instance, when a pure tone is transmitted through a time varying channel, the observed signal at the receiver is composed of a

band of frequencies. Such a channel is termed time selective, since its characteristics vary with time.

Channels that are frequency as well as time selective are referred to as “doubly selective”.

There is a growing demand for higher data rate (smaller symbol duration) systems that can function in a highly mobile environment. This mandates the design of communication systems that can function in doubly selective channels.

Leveraging computationally efficient FFT operations at both receiver and transmitter, orthogonal frequency division multiplexing (OFDM) [2–4] is an attractive alternative for frequency selective channels. The key idea in OFDM to transmit in parallel over the set of orthogonal discrete Fourier basis vectors as sub-carriers, effectively increasing the symbol interval. By the insertion of a guard interval(proportional to the maximum delay spread of the channel) between symbols, ISI is completely avoided. However, when channels in question are doubly selective, the OFDM sub-carriers are no longer orthogonal due to Doppler spreading. The energy from symbols transmitted on a particular sub-carrier corrupt neighboring sub-carriers [5]. This phenomena is termed inter-carrier interference (ICI). A number of different approaches to make OFDM robust to ICI exist [5–8]. An alternative approach is to consider the more general set of modulation techniques (of which OFDM is a subset) known as Multi-carrier Modulation (MCM). One such approach to use MCM over doubly selective channels, is using pulse shaped MCM (PS-MCM) [9, 10]. PS-MCM is based on the realization that trying to mitigate both ISI and ICI in a doubly selective channel might be over ambitious. Instead, time domain pulses are used to *shape* the channel profile so that it meets a target specification. It relies on a high performance

low complexity equalizer to reliably extract the transmitted symbols in the presence of this controlled amount of interference in the target specification.

This design philosophy serves as a premise for this thesis. Specifically, this thesis attempts to design suitable equalizers for a PS-MCM system.

Classical equalizers can be classified into two broad classes. The first class is based on linear combining of the observation, *viz.*, linear equalizers (LE). LE have low computational complexity, but suffer from poor performance due to noise enhancement. The other class of equalizers are those that exploit the trellis structure introduced by the channel, *viz.*, trellis based equalizers (TBE). Commonly used TBE such as a Viterbi Algorithm based equalizer [11] outperform LE but suffer from enormous computational complexity for long channels. The aim is to design equalizers that outperform linear equalizers, and have affordable low complexity.

Iterative approaches to interference suppression have been suggested for various systems. Most of these methods use soft decisions to cancel interference [12]. The advantage lies in the fact that soft decisions quantify the reliability of the decision in addition to the decision itself. Thus the interference canceled corresponds to symbols with highly reliable estimates only. Interference cancellation using feed back of soft decisions, such as in [12], are popularly called soft interference cancellation (SIC). SIC is more robust to error propagation than decision feedback equalizers [13]. Motivated by this, both an iterative structure and use of SIC is incorporated in the designs presented.

Finally, the bit error rate (BER) performance of an MCM system is severely hampered by sub-carrier nulls. Thus coding is employed at the transmitter. Decoding and interference-cancellation are coupled at the receiver.

Specifically, we propose two receiver schemes, both of which pass soft bit estimates between an iterative equalizer and soft-input soft-output (SISO) decoder in a turbo-like fashion [14, 15]. The first scheme, based on a LE using the minimum mean-squared error (MMSE) criterion, builds on the work of [15, 16].

The second scheme is based on a maximum likelihood (ML) equalization stage which is related to the probabilistic data association (PDA) algorithms in [17, 18]. However, in contrast to [15] and [17, 18], our algorithms are specifically tailored to the structure of the ICI/ISI-shaped channel.

1.1 Organization of Thesis

This thesis has been organized into 6 chapters. The ensuing chapter introduces the system model used and the key components of the iterative equalizers. The following two chapters present detailed discussions on the two designed equalizers. The next chapter presents a short discussion on the coding aspect of the problem. The final chapter presents experimental results and conclusions along with possible future research directions.

1.2 Notation Used

We use $(\cdot)^t$ to denote transpose, $(\cdot)^*$ conjugate, and $(\cdot)^H$ conjugate transpose. $\mathcal{C}(\boldsymbol{b})$ denotes the circulant matrix with first column $\boldsymbol{b}$, $\mathcal{D}(\boldsymbol{b})$ the diagonal matrix created from vector $\boldsymbol{b}$, and $\boldsymbol{I}_K$ the $K \times K$ identity matrix. We use $[\boldsymbol{B}]_{m,n}$ to denote the element in the m^{th} row and n^{th} column of $\boldsymbol{B}$, where row/column indices begin with zero. Expectation is denoted by $\mathrm{E}\{\cdot\}$, cross-covariance by $\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{c}} := \mathrm{E}\{\boldsymbol{b}\boldsymbol{c}^H\} - \mathrm{E}\{\boldsymbol{b}\}\,\mathrm{E}\{\boldsymbol{c}^H\}$ and auto-covariance by $\boldsymbol{\Sigma}_{\boldsymbol{b}} := \mathrm{E}\{\boldsymbol{b}\boldsymbol{b}^H\} - \mathrm{E}\{\boldsymbol{b}\}\,\mathrm{E}\{\boldsymbol{b}^H\}$. $\oplus$ denotes modulo-2 addition over $\{0, 1\}$. Finally, $\mathbb{R}$ the field of reals, and $\mathbb{Z}$ the set of integers.

CHAPTER 2

Iterative Equalization

This chapter introduces the iterative equalization algorithms that form the core of this thesis. First, an appropriate model for a generic MCM scheme incorporating pulse shapes is explained. Then a suitable decision metric is introduced. Finally, the key ideas behind the iterative equalization algorithms are elucidated based on the given system model and decision metric. This chapter also presents two receiver configurations incorporating these iterative equalizers, addressing the issue of coupling the equalizers with decoders for error control coding.

2.1 System Model

In a MCM system, at each multi carrier symbol index $i \in \mathbb{Z}$, a vector of uncorrelated [1] bits $\boldsymbol{c}^{(i)} = [\mathbf{c}_0^{(i)^t}, \mathbf{c}_1^{(i)^t}, \cdots, \mathbf{c}_{N-1}^{(i)^t}]^t$, where $\mathbf{c}_k^{(i)} = \left[c_{k,0}^{(i)}, c_{k,1}^{(i)}, \cdots, c_{k,M-1}^{(i)}\right]^t$ and $c_{k,m}^{(i)} \in \{0,1\}$, is mapped to a vector of symbols, $\boldsymbol{s}^{(i)} = [s_0^{(i)}, s_1^{(i)}, \cdots, s_{N-1}^{(i)}]^t$, $s_k^{(i)} \in \mathbb{S}$ by the symbol mapping $\psi : \{0,1\}^M \rightarrow \mathbb{S}$, where $\mathbb{S}$ is the *constellation* of size $|\mathbb{S}| = 2^M$. In this thesis, $\mathbb{S}$ is restricted to a Gray-coded PSK constellation, for simplicity. This set of N coded PSK symbols $\{s_k^{(i)}\}$ is collected to form a multi-carrier symbol $\boldsymbol{s}^{(i)} = [s_0^{(i)}, \ldots, s_{N-1}^{(i)}]^t$. These

[1] If coding is employed, then $\boldsymbol{c}^{(i)}$ is an interleaved vector of coded bits.

symbols are used to modulate pulsed carriers as follows:

$$t_n \;=\; \sum_{i=-\infty}^{\infty} a_{n-iN_s} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} s_k^{(i)} e^{j\frac{2\pi}{N}(n-iN_s-N_o)k} \tag{2.1}$$

In (2.1), $\{a_n\}$ is the transmit pulse sequence, N_s is the multi-carrier symbol interval, and $N_o \in \{0, \ldots, N-1\}$ delays the carrier origin relative to the pulse origin. The multi-path channel is described by its time-variant discrete impulse response $h_{\mathsf{tl}}(n,l)$, defined as the time-n response to an impulse applied at time $n-l$. We assume a causal impulse response of length N_h. The signal observed by the receiver is then

$$r_n \;=\; \nu_n + \sum_{l=0}^{N_h-1} h_{\mathsf{tl}}(n,l) t_{n-l} \tag{2.2}$$

where ν_n denotes samples of circular white Gaussian noise (CWGN) with variance σ^2. Defining $r_n^{(i)} := r_{iN_s+n}$, $\nu_n^{(i)} := \nu_{iN_s+n}$, and $h_{\mathsf{tl}}^{(i)}(n,l) := h_{\mathsf{tl}}(iN_s+n,l)$, it can be shown that

$$r_n^{(i)} \;=\; \nu_n^{(i)} + \sum_{l=0}^{N_h-1} h_{\mathsf{tl}}^{(i)}(n,l) \sum_{\ell=-\infty}^{\infty} a_{\ell N_s+n-l} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} s_k^{(i-\ell)} e^{j\frac{2\pi}{N}(n-l+\ell N_s-N_o)k} \tag{2.3}$$

To estimate the multi-carrier symbol $\boldsymbol{s}^{(i)}$, the receiver employs the pulse $\{b_n\}$ as follows:

$$x_d^{(i)} \;=\; \frac{1}{\sqrt{N}} \sum_n r_n^{(i)} b_n e^{-j\frac{2\pi}{N}d(n-N_o)} \tag{2.4}$$

Here again N_o delays the carrier origin relative to the pulse origin. Note that this system reduces to CP-OFDM with $N_o = N_s - N$, $\{a_n\}_{n=0}^{N_s-1} = 1$, and $\{b_n\}_{n=N_o}^{N_s-1} = 1$ (else $a_n = b_n = 0$). Note also that $N_g := N_s - N$ is analogous to CP-OFDM guard interval.

Plugging (2.3) into (2.4), we find

$$x_d^{(i)} \;=\; w_d^{(i)} + \sum_{\ell} \sum_{k=0}^{N-1} h_{\mathsf{df}}^{(i,\ell)}(d-k,k)\, s_k^{(i-\ell)} \tag{2.5}$$

where

$$w_d^{(i)} := \frac{1}{\sqrt{N}} \sum_n b_n \nu_n^{(i)} e^{-j\frac{2\pi}{N} d(n-N_o)} \tag{2.6}$$

$$h_{\mathsf{df}}^{(i,\ell)}(d,k) := \frac{1}{N} \sum_n \sum_{l=0}^{N_h-1} h_{\mathsf{tl}}^{(i)}(n,l) b_n a_{\ell N_s+n-l}\, e^{-j\frac{2\pi}{N} d(n-N_o)} e^{-j\frac{2\pi}{N} k(l-\ell N_s)} \tag{2.7}$$

Equation (2.5) indicates that $h_{\mathsf{df}}^{(i,\ell)}(d,k)$ can be interpreted as the response, at time i and sub-carrier $k+d$, to a frequency-domain impulse applied at time $i-\ell$ and sub-carrier k.

In practice we implement finite-duration causal pulses $\{a_n\}$ and $\{b_n\}$ of length N_a and N_b, respectively, implying that only a finite number of terms in the set $\{h_{\mathsf{df}}^{(i,\ell)}(d,k), \ell \in \mathbb{Z}\}$ will be non-zero. Specifically, (2.7) implies that non-zero terms result from indices ℓ which satisfy $0 \le \ell N_s + n - l \le N_a - 1$ for some $n \in \{0,\ldots,N_b-1\}$ and some $l \in \{0,\ldots,N_h-1\}$. It is straightforward to show that $h_{\mathsf{df}}^{(i,\ell)}(d,k)$ is non-zero for $\ell \in \{-L_{\mathsf{pre}},\ldots,L_{\mathsf{pst}}\}$ where $L_{\mathsf{pre}} = -\lfloor \frac{N_b-1}{N_s} \rfloor$ and $L_{\mathsf{pst}} = \lfloor \frac{N_a+N_h-2}{N_s} \rfloor$.

With the definitions $\boldsymbol{x}^{(i)} := [x_0^{(i)},\ldots,x_{N-1}^{(i)}]^t$, $\boldsymbol{w}^{(i)} := [w_0^{(i)},\ldots,w_{N-1}^{(i)}]^t$, and $[\boldsymbol{H}^{(i,\ell)}]_{d,k} := h_{\mathsf{df}}^{(i,\ell)}(d-k,k)$, (2.5) implies the linear time-varying (LTV) multiple-input multiple-output (MIMO) system

$$\boldsymbol{x}^{(i)} = \boldsymbol{w}^{(i)} + \sum_{\ell=-L_{\mathsf{pre}}}^{L_{\mathsf{pst}}} \boldsymbol{H}^{(i,\ell)} \boldsymbol{s}^{(i-\ell)}. \tag{2.8}$$

In the sequel we assume wide-sense stationary uncorrelated scattering (WSSUS) [13] so that $\mathrm{E}\{h_{\mathsf{tl}}(n,l) h_{\mathsf{tl}}^*(n-q,l-m)\} = r_{\mathsf{t}}(q)\sigma_l^2\delta(m)$. Here, $r_{\mathsf{t}}(q)$ denotes the normalized autocorrelation (i.e., $r_{\mathsf{t}}(0)=1$) and σ_l^2 the variance of the l^{th} lag.

2.2 Pulse Shape Design: An Aside

Even though, the design of suitable pulse shapes $\{a_n\}$ and $\{b_n\}$ is not the focus of this thesis, they are a key component of a MCM system. Hence, this short section presents a

terse qualitative discussion on pulse shape design. For details on pulse shape design, the reader is referred to [9, 10, 16].

It is well known that unlike the class of LTI systems, there is no common set of eigen vectors for LTV systems. Therefore, use of the discrete Fourier basis to modulate frequency domain symbols is not justified. In the approach adopted in [9, 10], the symbols are modulated by complex exponentials that are shaped by time domain pulses $\{a_n\}$ and $\{b_n\}$ at the transmitter and receiver, respectively as in (2.3) and (2.4).

The problem can be viewed from another perspective. There are two classes of interference to be countered, *viz.*, inter-symbol interference (ISI) and inter-carrier interference (ICI). Use of narrow time domain pulses allows for the suppression of ISI at the expense of introduced ICI. On the other hand, use of wide time domain pulses helps suppress ICI at the expense of the ISI introduced in the absence of guard intervals. Even if this were possible using linear processing, the noise enhancement would impair the performance significantly. Instead, the attempt is to shape the channel profile to allow for a "tolerable" amount of ISI and/or ICI that has to be countered by a suitably designed equalizer.

In [9,16], the pulse shapes are designed to allow for D taps of ICI on either side of a subcarrier, where D depends on the maximum Doppler frequency f_d for the system. Fig. 2.1 shows a typical desired channel profile. Three types of pulse shapes can be designed using this approach, *viz.*,

- *Transmitter Pulses*: Design optimal $\{a_n\}$ for a fixed (rectangular) pulse shape $\{b_n\}$ at the receiver.
- *Jointly Optimized Pulses*: Design optimal pair of transmitter and receiver pulses $\{a_n\}$ and $\{b_n\}$.

- *Receiver Pulses*: Design optimal receiver pulse shape $\{b_n\}$ for a fixed CP-OFDM (rectangular) transmitter pulse shape.

2.3 Decision Metric

The bit error rate optimal MAP detection strategy [19] for bit $c_{k,m}^{(i)}$ of a given multi-carrier symbol $\boldsymbol{x}^{(i)}$, is based on a comparison of the *a posteriori* probabilities $\{P(c_{k,m}^{(i)} = \gamma|\boldsymbol{x}^{(i)})\}_{\gamma=0}^{1}$. Our equalizer uses the observation $\boldsymbol{x}^{(i)}$ and knowledge of $\boldsymbol{H}^{(i,0)}$ to update the i^{th} multi-carrier symbol's bit reliability metric $\{L^{(i)}(k,m), \forall m\}_{k=0}^{N-1}$, also referred to as L-values (LVs).

$$L^{(i)}(k,m) \;:=\; \ln \frac{P(c_{k,m}^{(i)} = 0|\boldsymbol{x}^{(i)})}{P(c_{k,m}^{(i)} = 1|\boldsymbol{x}^{(i)})} \tag{2.9}$$

Since all quantities pertain to the i^{th} multi-carrier symbol, superscript indices are omitted for the remainder of the thesis without loss of generality. They are only re-introduced when multiple multicarrier symbols have to be distinguished. Here, it is pointed out that when subscripts are omitted, $\boldsymbol{H}$ refers to $\boldsymbol{H}^{(i,0)}$. Note that the sign of $L(k,m)$ is the uncoded MAP bit decision, the magnitude of $L(k,m)$ indicates the reliability of this decision and that the *a posteriori* probabilities for each bit can be written as (2.10).

$$P(c_{k,m} = \gamma|\boldsymbol{x}) \;=\; \begin{cases} \frac{e^{L(k,m)}}{1+e^{L(k,m)}} & \gamma = 0 \\ \\ \frac{e^{-L(k,m)}}{1+e^{-L(k,m)}} & \gamma = 1. \end{cases} \tag{2.10}$$

In fact, this decision statistic is very popular and often seen in algorithms for both equalization [17, 20] and decoding [21, 22]. Using Bayes' rule and *assuming independent bits*,[2] $L(k,m)$ can be rewritten as the sum of the prior LV, $L_{\text{old}}(k,m)$, and the extrinsic LV,

[2] As a consequence of, e.g., interleaving, in the presence of error control coding.

$\Delta L(k,m)$, defined in (2.11).

$$L(k,m) = \underbrace{\ln \frac{\sum_{\boldsymbol{\gamma}\in\mathcal{G}_{kN+m,0}^{NM}} p(\boldsymbol{x}|\boldsymbol{c}=\boldsymbol{\gamma}) \prod_{(k',m')\neq(k,m)} P(c_{k',m'}=\gamma_{k'N+m'})}{\sum_{\boldsymbol{\gamma}\in\mathcal{G}_{kN+m,1}^{NM}} p(\boldsymbol{x}|\boldsymbol{c}=\boldsymbol{\gamma}) \prod_{(k',m')\neq(k,m)} P(c_{k',m'}=\gamma_{k'N+m'})}}_{\Delta L(k,m)} + \underbrace{\ln \frac{P(c_{k,m}=0)}{P(c_{k,m}=1)}}_{L_{\text{old}}(k,m)}. \tag{2.11}$$

Here, $\mathcal{G}_{k_2,\alpha}^{k_1}$ denotes the set of all length-k_1 bit vectors in which the k_2^{th} bit has been set to $\alpha \in \{0,1\}$. Intuitively, 2.11 says that the LV can be calculated by taking the ratio of the sum of *a posteriori* probabilities of all length-NM bit sequences in which the $(kM+m)^{th}$ bit is fixed as $\gamma \in \{0,1\}$. Another important issue is the decoupling of $\Delta L(k,m)$ and $L_{\text{old}}(k,m)$. The algorithms assume that the extrinsic LV $\Delta L(k,m)$ is *independent* of the prior LV $L_{\text{old}}(k,m)$. Even though this is not completely true of the presented algorithms, we ensure that $L_{\text{old}}(k,m)$ is not directly used in the calculation of $\Delta L(k,m)$. However, from the algorithms, it can be deduced that, through $L_{\text{old}}(k,m')$ where $m' \neq m$, $L_{\text{old}}(k,m)$ does influence the value of $\Delta L(k,m)$ to a small extent.

2.4 Iterative Equalization: Key Ideas

Given the observation $\boldsymbol{x}$, the channel information $\boldsymbol{H}$, and the statistics of the additive zero-mean complex Gaussian noise $\boldsymbol{\Sigma_w}$, classical equalizers use various algorithms to compute an estimate $\hat{\boldsymbol{x}}$. In addition to these available quantities, by virtue of its iterative nature, an iterative equalizer has prior information L_{old} available to it. The task of the iterative equalizer is to update the LV for each bit sequentially using (2.11). In essence, this entails computing the extrinsic LV, ΔL. We see in (2.11) that the numerator and denominator for ΔL are summations of 2^{MN-1} terms each. Hence, the computation of ΔL is $\mathcal{O}\left(2^{MN}\right)$ per

LV update and its exact computation is generally infeasible. The algorithms proposed in this thesis are sub-optimal strategies based on approximations of $\Delta L(k,m)$ by

$$\Delta L(k,m) \approx \ln \frac{\sum_{\boldsymbol{\gamma}\in\mathcal{G}_{m,0}^{M}} p(\boldsymbol{x}|\boldsymbol{c}_k=\boldsymbol{\gamma}) \prod_{m'\neq m} P(c_{k,m'}=\gamma_{m'})}{\sum_{\boldsymbol{\gamma}\in\mathcal{G}_{m,1}^{M}} p(\boldsymbol{x}|\boldsymbol{c}_k=\boldsymbol{\gamma}) \prod_{m'\neq m} P(c_{k,m'}=\gamma_{m'})}. \tag{2.12}$$

In (2.12), the extrinsic LV is approximately computed by considering all possible values of symbol x_k for which $c_{k,m} = \gamma \in \{0,1\}$. This brings down the complexity of the algorithm to $\mathcal{O}\left(2^M\right)$ per LV update. For simple constellations, *viz.*, BPSK and QPSK, it will be shown that this update can be greatly simplified. The next two chapters provide alternative approaches to compute $\Delta L(k,m)$ via (2.12).

The information present in the LVs can be further exploited in equalization. Using these *a posteriori* probabilities and the symbol mapping function ψ, the symbol means $\{\mu_k\}$ and variances $\{v_k\}$ are defined in (2.13)-(2.14).

$$\mu_k = \sum_{\beta\in\mathbb{S}} \beta P(s_k=\beta|\boldsymbol{x}) \tag{2.13}$$

$$v_k = \sum_{\beta\in\mathbb{S}} |\beta|^2 P(s_k=\beta|\boldsymbol{x}) - |\mu_k|^2 \tag{2.14}$$

Recall that, for β and $\boldsymbol{\gamma}$ related by $\psi(\boldsymbol{\gamma})=\beta$, we have

$$\begin{aligned} P(s_k=\beta|\boldsymbol{x}) &= P(\boldsymbol{c}_k=\boldsymbol{\gamma}|\boldsymbol{x}) \\ &= \Pi_{i=1}^{M} P(c_{k,m}=\gamma_m|\boldsymbol{x}). \end{aligned}$$

Also note that, since Gray-mapping is assumed, the real and imaginary components of each symbol are independent and thus the variances $v_{k,R} := \mathrm{var}(\mathrm{Re}\, s_k)$ and $v_{k,I} := \mathrm{var}(\mathrm{Im}\, s_k)$ can be written

$$v_{k,R} = \sum_{\beta\in\mathbb{S}} (\mathrm{Re}\,\beta)^2 P(s_k=\beta|\boldsymbol{x}) - (\mathrm{Re}\,\mu_k)^2 \tag{2.15}$$

$$v_{k,I} = \sum_{\beta\in\mathbb{S}} (\mathrm{Im}\,\beta)^2 P(s_k=\beta|\boldsymbol{x}) - (\mathrm{Im}\,\mu_k)^2. \tag{2.16}$$

Now, these estimated symbol means μ_k can be used for interference cancellation, in place of quantized hard decisions. Note that this kind of interference cancellation is commonly termed in literature as soft interference cancellation (SIC). SIC is less prone to error propagation as compared to feeding back hard decisions. This is because symbol estimates that are unreliable will have small LVs and small symbol means as a consequence. Thus the interference due to an unreliable symbol remains almost unaltered and only the interference arising from reliably estimated symbols is canceled.

These are the pivotal ideas behind the iterative equalizers detailed in the subsequent chapters. The equalization algorithms discussed are sequential in nature. We begin with symbol index $k = 0$ and sequentially perform the three steps mentioned below for each symbol.

- Use current symbol means $\{\mu_k\}$ for SIC.
- Update the LV using (2.12) and $\{v_k\}$.
- Re-estimate $\{\mu_k\}$ and $\{v_k\}$ using updated LVs.

The exact iteration structure for each algorithm is explained after a description of the algorithms in subsequent chapters.

2.5 Receiver Configurations

In order to compare the efficacy of turbo equalization, we consider two receiver configurations: serial equalization/decoding, shown in Fig. 2.2, and turbo equalization/decoding, shown in Fig. 2.3. The equalizer block is same in either configuration; it uses the observation $\boldsymbol{x}^{(i)}$ to update LVs of bits mapping to the i^{th} multi-carrier symbol. In serial equalization/decoding, the equalizer's input LVs are initialized to zero and its output LVs

are passed to a soft-input decoder which generates final bit decisions. Turbo equalization/decoding starts out this way, but, instead of generating final bit decisions, the decoder updates the LVs and passes them back to the equalizer. The equalizer then updates the LVs and passes them back to the decoder, and this process repeats several times before the decoder generates final bit decisions. The equalizer block itself is iterative in that its LVs are updated internally several times before being passed to the decoder. Note that the equalizer does not use the code structure directly; it assumes the bits are independent and calculates their prior probabilities from the input LVs.

As for the notation in Figs. 2.2 and 2.3, L_{eq}, L_{dc}, and L_{c} denote equalizer, decoder, and coder LVs, respectively, on which the superscripts i, e, and o label input, extrinsic, and output versions, respectively. The switch (s_a, s'_a) is set to position s'_a to initialize the decoder.

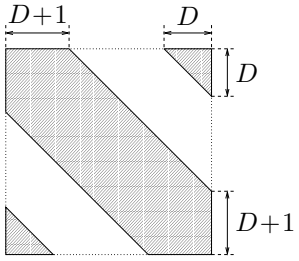

Figure 2.1: Desired structure of MIMO cursor coefficient $\boldsymbol{H}^{(i,0)}$.

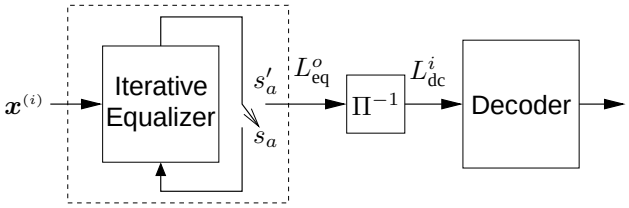

Figure 2.2: Serial receiver configuration.

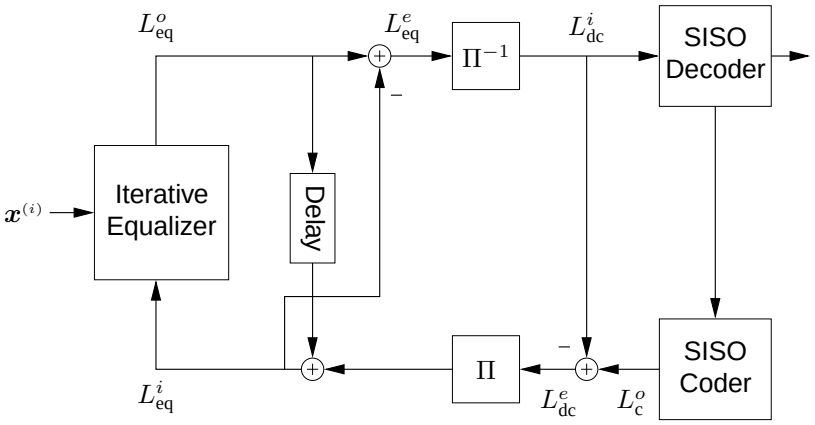

Figure 2.3: Turbo receiver configuration.

CHAPTER 3

Iterative Maximum Likelihood Equalizer (IMLE)

This technique derives its name from the fact that $\Delta L(k,m)$ is the maximum likelihood (ML) decision statistic for bit $c_{k,m}$. However, as discussed earlier, computing the exact ML decision metric is computationally infeasible. Here $\Delta L(k,m)$ is approximated to reduce computational complexity. In this chapter, first a generic expression for the extrinsic LV is derived. Then a simplified system model, exploiting the structure of the channel matrices, is presented and the extrinsic LV are computed on the basis of this simplified system model. This results in substantial reduction in complexity. Finally, the extrinsic LV is simplified for BPSK and Gray mapped QPSK constellations.

3.1 Derivation of the Extrinsic LV

The key idea is to first perform a soft interference cancellation (SIC) using the symbol means $\{\mu_k\}$, then to apply a Gaussian model to the residual interference and noise.The resulting $\Delta L(k,m)$ approximation, denoted by $\Delta L_{\text{IMLE}}(k,m)$, is much easier to compute.

We define $\boldsymbol{\mu}_k$ as

$$\boldsymbol{\mu}_k \;:=\; [\mu_0,\ldots,\mu_{k-1},0,\mu_{k+1},\ldots,\mu_{N-1}]^t. \tag{3.1}$$

and use it for SIC. Notice that the mean for the k^{th} symbol has been set to zero. This ensures that the contribution of the k^{th} symbol is not suppressed from $\boldsymbol{y}_k$. Specifically, the

observation after SIC is written

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{x} - \boldsymbol{H}\boldsymbol{\mu}_k & (3.2)\\ &= \boldsymbol{h}_k s_k + \underbrace{\sum_{\substack{j=0\\ j\neq k}}^{N-1} \boldsymbol{h}_j (s_j - \mu_j) + \boldsymbol{w}}_{\boldsymbol{q}_k} & (3.3) \end{aligned}$$

where $\boldsymbol{h}_k$ denotes the k^{th} column of $\boldsymbol{H}$.

The residual interference vector $\boldsymbol{q}_k$ is modeled as Gaussian, independent of s_k. The mean of $\boldsymbol{q}_k$ is

$$\begin{aligned} E\{\boldsymbol{q}_k\} &= \sum_{\substack{j=0\\ j\neq k}}^{N-1} \boldsymbol{h}_j (E\{s_j\} - \mu_j) + E\{\boldsymbol{w}\} \\ &= \boldsymbol{0}. \end{aligned}$$

The covariance $\boldsymbol{\Sigma}_{\boldsymbol{q}_k}$ is

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{q}_k} &= E\{\boldsymbol{q}_k \boldsymbol{q}_k^H\} \\ &= \sum_{\substack{j=0\\ j\neq k}}^{N-1} \boldsymbol{h}_j E\left\{(s_j - \mu_j)(s_j - \mu_j)^*\right\} \boldsymbol{h}_j^H + \boldsymbol{\Sigma}_{\boldsymbol{w}} \\ &= \sum_{\substack{j=0\\ j\neq k}}^{N-1} v_j \boldsymbol{h}_j \boldsymbol{h}_j^H + \boldsymbol{\Sigma}_{\boldsymbol{w}}. & (3.4)\\ &= \boldsymbol{H}\,\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}^H + \boldsymbol{\Sigma}_{\boldsymbol{w}} & (3.5) \end{aligned}$$

where

$$\boldsymbol{v}_k := [v_0, \ldots, v_{k-1}, 0, v_{k+1}, \ldots, v_{N-1}]^t. \quad (3.6)$$

Replacing $p(\boldsymbol{x}|\boldsymbol{c}=\boldsymbol{\gamma})$ in (2.12) with $p(\boldsymbol{y}_k|s_k = \psi(\boldsymbol{\gamma}))$, the extrinsic LV becomes

$$\Delta L_{\text{IMLE}}(k,m) = \ln \frac{\sum_{\boldsymbol{\gamma}\in\mathcal{G}_{m,0}^M} p\left(\boldsymbol{y}_k|s_k=\psi(\boldsymbol{\gamma})\right) \prod_{m'\neq m} P(c_{k,m'}=\gamma_{m'})}{\sum_{\boldsymbol{\gamma}\in\mathcal{G}_{m,1}^M} p\left(\boldsymbol{y}_k|s_k=\psi(\boldsymbol{\gamma})\right) \prod_{m'\neq m} P(c_{k,m'}=\gamma_{m'})} \quad (3.7)$$

Expanding $p(\boldsymbol{y}_k|s_k=\psi(\boldsymbol{\gamma}))$ using (3.3):

$$p(\boldsymbol{y}_k|s_k=\psi(\boldsymbol{\gamma})) = C\exp\left[-\frac{1}{2}(\boldsymbol{y}_k-\psi(\boldsymbol{\gamma})\boldsymbol{h}_k)^H\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1}(\boldsymbol{y}_k-\psi(\boldsymbol{\gamma})\boldsymbol{h}_k)\right] \quad (3.8)$$

$$= C\exp\left[-\frac{1}{2}\boldsymbol{y}_k^H\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1}\boldsymbol{y}_k-\frac{1}{2}|\psi(\boldsymbol{\gamma})|^2\boldsymbol{h}_k^H\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1}\boldsymbol{h}_k)\right]\exp\left[\mathrm{Re}\left(\psi(\boldsymbol{\gamma})g_k\right)\right] \quad (3.9)$$

Since the first exponential in (3.9) is invariant to $\boldsymbol{\gamma}$ (recalling the PSK assumption), its contributions to the numerator and denominator of (3.7) cancel. From (2.10), we find

$$P(c_{k,m'}=0) = \frac{e^{L_{\text{old}}(k,m')/2}}{2\cosh(L_{\text{old}}(k,m')/2)} \quad (3.10)$$

$$P(c_{k,m'}=1) = \frac{e^{-L_{\text{old}}(k,m')/2}}{2\cosh(L_{\text{old}}(k,m')/2)}. \quad (3.11)$$

Together,

$$P(c_{k,m'}=\gamma_{m'}) = \frac{\exp\left[(-1)^{\gamma_{m'}}L_{\text{old}}(k,m')/2\right]}{2\cosh(L_{\text{old}}(k,m')/2)} \quad (3.12)$$

Since $\cosh(L_{\text{old}}(k,m')/2)$ is invariant to $\gamma_{m'}$, its contributions to the numerator and denominator of (3.7) cancel. The remaining terms yield

$$\Delta L_{\text{IMLE}}(k,m) = \ln\frac{\sum_{\boldsymbol{\gamma}\in\mathcal{G}_{m,0}^M}\exp\left[\mathrm{Re}\big(\psi(\boldsymbol{\gamma})g_k\big)+\frac{1}{2}\sum_{m'\neq m}(-1)^{\gamma_{m'}}L_{\text{old}}(k,m')\right]}{\sum_{\boldsymbol{\gamma}\in\mathcal{G}_{m,1}^M}\exp\left[\mathrm{Re}\big(\psi(\boldsymbol{\gamma})g_k\big)+\frac{1}{2}\sum_{m'\neq m}(-1)^{\gamma_{m'}}L_{\text{old}}(k,m')\right]}, \quad (3.13)$$

where $g_k := \boldsymbol{y}_k^H\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1}\boldsymbol{h}_k$.

Calculation of (3.13) can be implemented in two stages. First we produce g_k from $\boldsymbol{y}_k$ using the linear combiner $\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1}\boldsymbol{h}_k$. Next we compute the extrinsic LVs using g_k and $\{L_{\text{old}}(k,m')\}_{m'\neq m}$. Also note that, as emphasized, $L_{\text{old}}(k,m)$ is not used when computing $\Delta L_{\text{IMLE}}(k,m)$.

This completes the derivation of $L_{\text{IMLE}}(k,m)$. In Sec. 3.5, expressions for $L_{\text{IMLE}}(k,m)$, μ_k and v_k for specific constellations, *viz.*, BPSK and QPSK, are derived.

3.2 Structure of Iterations

The IMLE algorithm proceeds as follows. Prior to the first iteration, $\{L_{\text{old}}(k,m)\ \forall k,m\}$ are obtained from the output of a soft decoder, if available, or otherwise set to zero. These LVs are then used to initialize $\{\mu_k\}_{k=0}^{N-1}$ and $\{v_k\}_{k=0}^{N-1}$. We begin the first iteration by working on symbol index $k=0$. The means $\boldsymbol{\mu}_0$ and variances $\boldsymbol{v}_0$ are used to calculate $\boldsymbol{y}_0$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}_0}$, which in turn are used to compute g_0. From g_0 and $\{L_{\text{old}}(0,m)\}_{m=0}^{M-1}$, $\{\Delta L_{\text{IMLE}}(0,m)\}_{m=0}^{M-1}$ are calculated and used to compute $\{L(0,m)\}_{m=0}^{M-1}$. Finally, $\{L(0,m)\}_{m=0}^{M-1}$ are used to update μ_0 and v_0. Moving on to $k=1$, the vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{v}_1$ are used to calculate $\boldsymbol{y}_1$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}_1}$, and later g_1. This allows the computation of $\{\Delta L_{\text{IMLE}}(1,m)\}_{m=0}^{M-1}$, the computation of $\{L(1,m)\}_{m=0}^{M-1}$, and the update of μ_1 and v_1. The $k=2$ case is tackled next, then $k=3$, and so on, until $k=N-1$. Finally, $\{L(k,m)\ \forall k,m\}$ are copied to $\{L_{\text{old}}(k,m)\ \forall k,m\}$. This concludes the first iteration. The next (e.g., second) iteration begins again at symbol index $k=0$ and proceeds through $k=N-1$. The algorithm terminates after a specified number of iterations.

The computationally expensive step of the algorithm is the computation of $\boldsymbol{\Sigma}_{\boldsymbol{q}_k}$ for each k. In order to reduce the complexity, $\boldsymbol{\Sigma}_{\boldsymbol{q}_k}$ is computed as an update of $\boldsymbol{\Sigma}_{\boldsymbol{q}_{k-1}}$. First, $\boldsymbol{\Sigma}_{\boldsymbol{Q}_k}$ is defined as:

$$\boldsymbol{\Sigma}_{\boldsymbol{Q}_k} \;=\; \boldsymbol{\Sigma}_{\boldsymbol{q}_k} + v_k \boldsymbol{h}_k \boldsymbol{h}_k^H \qquad \forall k \tag{3.14}$$

Once v_{k-1} is recomputed using updated LVs, $\boldsymbol{\Sigma}_{\boldsymbol{Q}_{k-1}}^{-1}$ is updated to $\boldsymbol{\Sigma}_{\boldsymbol{Q}_k}^{-1}$ using the matrix inversion lemma as:

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{Q}_k}^{-1} &= \boldsymbol{\Sigma}_{\boldsymbol{Q}_{k-1}}^{-1} - \frac{\Delta v_{k-1} \boldsymbol{\Sigma}_{\boldsymbol{Q}_{k-1}}^{-1} \boldsymbol{h}_{k-1} \boldsymbol{h}_{k-1}^H \boldsymbol{\Sigma}_{\boldsymbol{Q}_{k-1}}^{-1}}{1 + \Delta v_{k-1} \boldsymbol{h}_{k-1}^H \boldsymbol{\Sigma}_{\boldsymbol{Q}_{k-1}}^{-1} \boldsymbol{h}_{k-1}} \\ \text{where, } \Delta v_{k-1} &= v_{k-1}^{updated} - v_{k-1}. \end{aligned} \tag{3.15}$$

Finally, the matrix inversion lemma is used again to compute

$$\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} = \boldsymbol{\Sigma}_{\boldsymbol{Q}_k}^{-1} + \frac{v_k \boldsymbol{\Sigma}_{\boldsymbol{Q}_k}^{-1} \boldsymbol{h}_k \boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{Q}_k}^{-1}}{1 - v_k \boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{Q}_k}^{-1} \boldsymbol{h}_k} \tag{3.16}$$

This reduces the complexity of computing $\boldsymbol{\Sigma}_{\boldsymbol{q}_k}$ from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$. In spite of this update process, the $\mathcal{O}(N^2)$ update can prove to be very expensive for large N. Thus with the aim of substantially reducing the complexity, a simplified system model is presented in Sec. 3.3. This simplified system model exploits the banded structure of the channel matrices $\boldsymbol{H}^{(i,0)}$. Equalization based on this system model helps reduce the complexity to approximately $\mathcal{O}(N)$ without a substantial performance degradation.

3.3 Simplified System Model

With proper application of the pulse shapes described in [9, 16], the MIMO channel $\{\boldsymbol{H}^{(i,\ell)}\}_{\ell=-L_{\text{pre}}}^{L_{\text{pst}}}$ has negligible[3] pre- and post-cursor ISI and a cursor coefficient $\boldsymbol{H}$ with the banded structure shown in Fig. 2.1[4]. This implies that s_k will contribute primarily to the observation elements $\{x_d\}_{d=k-D}^{k+D}$, *where all indexing in this section is taken modulo-*N. Thus, good "local" estimates of s_k can be generated using $\boldsymbol{x}_k := [x_{k-D}, \cdots, x_{k+D}]^t$. If we define $\boldsymbol{s}_k := [s_{k-2D}, \cdots, s_{k+2D}]^t$, $\boldsymbol{w}_k := [w_{k-D}, \cdots, w_{k+D}]^t$, and $\boldsymbol{\varepsilon}_k := [\varepsilon_{k-D}, \cdots, \varepsilon_{k+D}]^t$, then

[3]When the delay spread is long relative to the multi-carrier symbol interval, it may be necessary to suppress post-cursor ISI using block decision feedback, as discussed in [9]. We assume that this is not the case here.

[4]As stated earlier, $\boldsymbol{H}$ refers to $\boldsymbol{H}^{(i,0)}$.

we can write

$$\boldsymbol{x}_k \;=\; \boldsymbol{H}_k \boldsymbol{s}_k + \boldsymbol{\varepsilon}_k, \tag{3.17}$$

where $\boldsymbol{H}_k$ is the sub-matrix of $\boldsymbol{H}$ built from rows $\{k-D,\ldots,k+D\}$ and columns $\{k-2D,\ldots,k+2D\}$, and where $\boldsymbol{\varepsilon}_k$ denotes noise plus residual ICI and ISI. Numerical studies have shown that, when pulse shapes are designed as in [9, 16], $\boldsymbol{\varepsilon}_k^{(i)}$ is dominated by the noise component and hence can be modeled as zero-mean Gaussian with covariance $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}_k}$. Note that, as a consequence of modulo-N indexing, the elements of $\boldsymbol{H}$ from the top-right and bottom-left shaded triangles in Fig. 2.1 will be included in $\{\boldsymbol{H}_k\}_{k=0}^{N-1}$.

Now, we use $\boldsymbol{x}_k$ and $\boldsymbol{H}_k$ to compute $\Delta L_{\text{IMLE}}(k,m)$. We use the process described in Sec. 3.1. For SIC, we define $\boldsymbol{\mu}_k$ as:

$$\boldsymbol{\mu}_k \;:=\; [\mu_{k-2D},\ldots,\mu_{k-1},0,\mu_{k+1},\ldots,\mu_{k+2D}]^t. \tag{3.18}$$

and compute a partial observation after SIC as

$$\begin{aligned}\boldsymbol{y}_k &= \boldsymbol{x}_k - \boldsymbol{H}_k\boldsymbol{\mu}_k \\ &= \boldsymbol{h}_{k,0}s_k + \underbrace{\sum_{\substack{j=k-2D \\ j\neq k}}^{k+2D} \boldsymbol{h}_{k,j}(s_j-\mu_j) + \boldsymbol{\varepsilon}_k}_{\boldsymbol{q}_k}\end{aligned} \tag{3.19}$$

where $\boldsymbol{h}_{k,j}$ denotes the $(j+2D)^{th}$ column of $\boldsymbol{H}_k$.

The covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{q}_k}$ then has the expression

$$\boldsymbol{\Sigma}_{\boldsymbol{q}_k} \;=\; \boldsymbol{H}_k\,\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}_k^H + \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}_k}$$

where,

$$\boldsymbol{v}_k \;:=\; [v_{k-2D},\ldots,v_{k-1},0,v_{k+1},\ldots,v_{k+2D}]^t. \tag{3.20}$$

Finally, the extrinsic LV, $\Delta L_{\text{IMLE}}(k,m)$ is given by (3.13), where $g_k := \boldsymbol{y}_k^H\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1}\boldsymbol{h}_{k,0}$.

3.4 Computational Complexity and Other Similar Algorithms

The computational complexity for IMLE is dominated by the inversion of the $(2D+1)\times(2D+1)$ matrix $\mathbf{\Sigma}_{q_k}$, yielding a per-iteration complexity order of $\mathcal{O}\left(ND^3\right)$. It is interesting to note that IMLE-BPSK is similar to the probabilistic data association (PDA)-based multi-user detection (MUD) schemes proposed in [17]. However, in [17], the iterative symbol detection strategy is applied *after* a zero-forcing (ZF) transformation is applied to $\boldsymbol{x}$, i.e., after the channel has been trivialized. Since the ZF transformation has a complexity order of $\mathcal{O}\left(N^3\right)$, it is much more costly than IMLE and IMSE since, typically, $D \ll N$. Essentially, IMLE leverages the banded structure of $\boldsymbol{H}$, while PDA does not.

IMLE is also similar to [18]. However, IMLE updates LVs for individual bits and passes bit LVs to the decoder, whereas soft symbol estimates are updated in [18] and bits are detected via hard decisions on the final symbol estimates. Also note that whereas [18] works in the time-lag domain, our scheme operates in the frequency-doppler domain. This is advantageous since the number of interfering symbols is smaller for our scheme as $D \ll N_h$.

3.5 IMLE Algorithm for BPSK and QPSK

In this section, we consider the BPSK and Gray Mapped QPSK symbol constellations and simplify the expressions for the extrinsic LV, the symbol mean and variance. It is assumed in this section that the simplified system model is used.

For BPSK ($M = 1$), we have

$$\psi(\boldsymbol{\gamma}) \;=\; (-1)^{\gamma_0}$$

The LV update $\Delta L_{\text{IMLE}}(k,m)$ for the IMLE algorithm simplifies to

$$\begin{aligned}\Delta L_{\text{IMLE}}(k,0) &= \ln \frac{\exp(\operatorname{Re} g_k)}{\exp(-\operatorname{Re} g_k)} \\ &= 2 \operatorname{Re} g_k,\end{aligned}$$

The symbol mean can be calculated as

$$\mu_k = \sum_{\gamma_0=0}^{1} (-1)^{\gamma_0} P(c_{k,0} = \gamma_0) \tag{3.21}$$

$$= \frac{e^{\frac{1}{2}L_{\text{old}}(k,0)} - e^{-\frac{1}{2}L_{\text{old}}(k,0)}}{2\cosh(L_{\text{old}}(k,0)/2)} \tag{3.22}$$

$$= \tanh(L_{\text{old}}(k,0)/2), \tag{3.23}$$

where (3.22) follows from (3.12).

To calculate variance v_k, we note in (2.14) that $|\beta| = 1$ for all symbols β in the BPSK and QPSK constellations. Thus $v_k = 1 - |\mu_k|^2$ for BPSK and QPSK.

On the other hand, for QPSK (M=2), we have

$$\psi(\boldsymbol{\gamma}) = \frac{1}{\sqrt{2}} \left[(-1)^{\gamma_0} + j(-1)^{\gamma_1}\right]$$

The LV updates $\Delta L_{\text{IMLE}}(k,m)$ for the IMLE algorithm can be written as

$$\begin{aligned}\Delta L_{\text{IMLE}}(k,0) &= \ln \frac{\sum_{\gamma_1=0}^{1} \exp\left[\operatorname{Re}(\psi([0,\gamma_1])g_k) + \frac{1}{2}(-1)^{\gamma_1} L_{\text{old}}(k,1)\right]}{\sum_{\gamma_1=0}^{1} \exp\left[\operatorname{Re}(\psi([1,\gamma_1])g_k) + \frac{1}{2}(-1)^{\gamma_1} L_{\text{old}}(k,1)\right]} \\ &= \ln \frac{\exp[\frac{1}{\sqrt{2}}\operatorname{Re} g_k] \sum_{\gamma_1=0}^{1} \exp\left[-\frac{1}{\sqrt{2}}\operatorname{Im}((-1)^{\gamma_1} g_k) + \frac{1}{2}(-1)^{\gamma_1} L_{\text{old}}(k,1)\right]}{\exp[-\frac{1}{\sqrt{2}}\operatorname{Re} g_k] \sum_{\gamma_1=0}^{1} \exp\left[-\frac{1}{\sqrt{2}}\operatorname{Im}((-1)^{\gamma_1} g_k) + \frac{1}{2}(-1)^{\gamma_1} L_{\text{old}}(k,1)\right]} \\ &= \sqrt{2}\operatorname{Re} g_k \\ \Delta L_{\text{IMLE}}(k,1) &= \ln \frac{\sum_{\gamma_0=0}^{1} \exp\left[\operatorname{Re}(\psi([\gamma_0,1])g_k) + \frac{1}{2}(-1)^{\gamma_0} L_{\text{old}}(k,0)\right]}{\sum_{\gamma_0=0}^{1} \exp\left[\operatorname{Re}(\psi([\gamma_1,0])g_k) + \frac{1}{2}(-1)^{\gamma_0} L_{\text{old}}(k,0)\right]} \\ &= \ln \frac{\exp[-\frac{1}{\sqrt{2}}\operatorname{Im} g_k] \sum_{\gamma_0=0}^{1} \exp\left[\frac{1}{\sqrt{2}}\operatorname{Re}((-1)^{\gamma_0} g_k) + \frac{1}{2}(-1)^{\gamma_0} L_{\text{old}}(k,0)\right]}{\exp[\frac{1}{\sqrt{2}}\operatorname{Im} g_k] \sum_{\gamma_0=0}^{1} \exp\left[\frac{1}{\sqrt{2}}\operatorname{Re}((-1)^{\gamma_0} g_k) + \frac{1}{2}(-1)^{\gamma_0} L_{\text{old}}(k,0)\right]} \\ &= -\sqrt{2}\operatorname{Im} g_k.\end{aligned}$$

For the symbol mean, a similar derivation to the one for BPSK leads to

$$\begin{aligned}\mu_k &= \frac{1}{\sqrt{2}}\sum_{\gamma_0=0}^{1}(-1)^{\gamma_0}P(c_{k,0}=\gamma_0)\sum_{\gamma_1=0}^{1}P(c_{k,1}=\gamma_1) \\ &\quad + \frac{j}{\sqrt{2}}\sum_{\gamma_1=0}^{1}(-1)^{\gamma_1}P(c_{k,1}=\gamma_1)\sum_{\gamma_0=0}^{1}P(c_{k,0}=\gamma_0) \qquad (3.24) \\ &= \frac{1}{\sqrt{2}}\tanh(L_{\text{old}}(k,0)/2)+\frac{j}{\sqrt{2}}\tanh(L_{\text{old}}(k,1)/2). \qquad (3.25)\end{aligned}$$

Table 3.1 shows the steps in one iteration of the IMLE algorithm for BPSK and QPSK constellations.

This concludes the discussion on the IMLE algorithm.

BPSK	QPSK
for $k = 0 : N-1$	for $k = 0 : N-1$
$\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} = \left(\boldsymbol{H}_k \mathcal{D}(\boldsymbol{v}_k) \boldsymbol{H}_k^H + \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}_k}\right)^{-1}$	$\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} = \left(\boldsymbol{H}_k \mathcal{D}(\boldsymbol{v}_k) \boldsymbol{H}_k^H + \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}_k}\right)^{-1}$
$\boldsymbol{y}_k = \boldsymbol{x}_k - \boldsymbol{H}_k \boldsymbol{\mu}_k$	$\boldsymbol{y}_k = \boldsymbol{x}_k - \boldsymbol{H}_k \boldsymbol{\mu}_k$
$g_k = \boldsymbol{y}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_{k,0}$	$g_k = \boldsymbol{y}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_{k,0}$
$L(k,0) = L_{\text{old}}(k,0) + 2 \operatorname{Re} g_k$	$L(k,0) = L_{\text{old}}(k,0) + \sqrt{2} \operatorname{Re} g_k$
	$L(k,1) = L_{\text{old}}(k,1) - \sqrt{2} \operatorname{Im} g_k$
$\mu_k = \tanh\left(\frac{L(k,0)}{2}\right)$	$\mu_k = \frac{1}{\sqrt{2}} \tanh\left(\frac{L(k,0)}{2}\right) + \frac{j}{\sqrt{2}} \tanh\left(\frac{L(k,1)}{2}\right)$
$v_k = 1 - \mu_k^2$	$v_k = 1 - \lvert\mu_k\rvert^2$
$L_{\text{old}}(k,0) = L(k,0)$	$L_{\text{old}}(k,0) = L(k,0), \;\; L_{\text{old}}(k,1) = L(k,1)$
end	end

Table 3.1: Steps in one iteration of IMLE.

CHAPTER 4

Iterative Minimum Mean Squared Error Equalizer (IMSE)

The main idea of the IMSE is to enhance the performance of a minimum mean squared error (MMSE) equalizer exploiting the prior LVs and the finite alphabet of the constellation. Whereas in IMLE we performed SIC before computing extrinsic LVs, here we perform linear MMSE estimation, and compute the extrinsic LVs by applying a Gaussian approximation to the estimation error. The chapter begins with a derivation of the extrinsic LV update equation for the IMSE algorithm. Next, the iteration structure is discussed. The simplified system model, introduced in Sec. 3.3, can be used to substantially reduce the complexity of the IMSE algorithm, too. The required modifications are summarized. A discussion on the complexity follows. As with the IMLE, simplified expressions for the extrinsic LVs, the symbol mean and variances are derived for the BPSK and Gray mapped QPSK constellations in the final section.

4.1 Derivation of Extrinsic LV for IMSE

The MMSE estimate of s_k given the observation $\boldsymbol{x}$ is [19]

$$\begin{aligned} \hat{s}_k\big|_{\text{MMSE}} &= E\{s_k\} + E\{s_k\boldsymbol{x}^H\}E\{\boldsymbol{x}\boldsymbol{x}^H\}^{-1}\left(\boldsymbol{x} - E\{\boldsymbol{x}\}\right) \\ &= E\{s_k\} + E\{s_k s_k^*\}\boldsymbol{h}_k^H\left(\boldsymbol{\Sigma_w} + \boldsymbol{H}\boldsymbol{\Sigma_s}\boldsymbol{H}^H\right)^{-1}\left(\boldsymbol{x} - \mathrm{E}\{\boldsymbol{x}\}\right). \end{aligned} \tag{4.1}$$

Now, the bit independence assumption implies that the covariance matrix $\boldsymbol{\Sigma_s}$ is diagonal. Using the prior LVs, we wish to compute $E\{\boldsymbol{x}\}$ and $\boldsymbol{\Sigma_s}$. However, to ensure $\Delta L(k,m)$ invariant to $L_{\text{old}}(k,m)$, we set $L_{\text{old}}(k,m) = 0,\ \forall m$. This results in $\mu_k = 0$ and $v_k = 1$ when estimating s_k.

Using these conditions, $E\{\boldsymbol{x}\}$ is computed as

$$\begin{aligned} E\{\boldsymbol{x}\} &= \boldsymbol{H}E\{\boldsymbol{s}\} + E\{\boldsymbol{w}\} \\ &= \boldsymbol{H}\boldsymbol{\mu}_k \end{aligned}$$

and

$$\boldsymbol{\Sigma_s} = \mathcal{D}\,\boldsymbol{v}_k,$$

where, as before, we define

$$\begin{aligned} \boldsymbol{v}_k &:= [v_0, \ldots, v_{k-1}, 0, v_{k+1}, \ldots, v_{N-1}]^t \\ \text{and, } \boldsymbol{\mu}_k &:= [\mu_0, \ldots, \mu_{k-1}, 0, \mu_{k+1}, \ldots, \mu_{N-1}]^t \end{aligned}$$

Using these definitions, the "extrinsic" MMSE estimate is calculated as

$$\hat{s}_k = \underbrace{\boldsymbol{h}_k^H \left(\boldsymbol{\Sigma_w} + \boldsymbol{H}\,\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}^H + \boldsymbol{h}_k\boldsymbol{h}_k^H\right)^{-1}}_{\boldsymbol{f}_k^H} (\boldsymbol{x}_k - \boldsymbol{H}_k\boldsymbol{\mu}_k)\,. \tag{4.2}$$

The extrinsic estimate $\hat{s}_k$ can be viewed from an alternate perspective. The idea is to consider SIC on the observation $\boldsymbol{x}$ and then using the resulting vector to find a linear MMSE estimate of s_k. The linear MMSE solution $\boldsymbol{f}_k$ is modified to reflect the suppressed interference, accordingly.

Next, it is assumed that the estimation error, $\hat{s}_k - s_k$, is complex Gaussian with uncorrelated real and imaginary components.

Equivalently, $\hat{s}_k$ is conditionally Gaussian with the means and variances below.

$$\bar{s}_{k,\boldsymbol{\gamma}} := \mathrm{E}\left\{\hat{s}_k | s_k = \psi(\boldsymbol{\gamma})\right\} \tag{4.3}$$

$$\sigma^2_{k,R} := \mathrm{E}\left\{\mathrm{Re}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}}) | s_k = \psi(\boldsymbol{\gamma})\right\} \tag{4.4}$$

$$\sigma^2_{k,I} := \mathrm{E}\left\{\mathrm{Im}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}}) | s_k = \psi(\boldsymbol{\gamma})\right\}. \tag{4.5}$$

Simulation studies have shown that IMSE's provision for $\sigma^2_{k,I} \neq \sigma^2_{k,R}$ is advantageous, especially when complex alphabets along with Gray mapping is employed. Using complex alphabets with Gray mapping means that the real and imaginary parts of each symbol depend on independent bits. In this case, the variance of the real and imaginary parts of the symbol are dependent on the LVs of independent bits.

The quantity $\bar{s}_{k,\boldsymbol{\gamma}}$ can be computed as:

$$\begin{aligned} \bar{s}_{k,\boldsymbol{\gamma}} &= E\left\{\boldsymbol{f}_k^H (\boldsymbol{x} - \boldsymbol{H}\boldsymbol{\mu}_k) | s_k = \psi(\boldsymbol{\gamma})\right\} \\ &= \boldsymbol{f}_k^H \left(E\left\{\boldsymbol{x} | s_k = \psi(\boldsymbol{\gamma})\right\} - \boldsymbol{H}\boldsymbol{\mu}_k\right) \\ &= \psi(\boldsymbol{\gamma}) \boldsymbol{f}_k^H \boldsymbol{h}_k. \end{aligned} \tag{4.6}$$

In order to compute $\sigma^2_{k,R}$, define

$$\begin{aligned} \bar{\boldsymbol{x}}_k &:= \boldsymbol{x} - \boldsymbol{H}\boldsymbol{\mu}_k \\ \text{so that, } \hat{s}_k &= \boldsymbol{f}_k^H \bar{\boldsymbol{x}}_k. \end{aligned}$$

The real part of $\hat{s}_k$ can be expressed as

$$Re\left(\hat{s}_k\right) = \frac{1}{2} \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{f}_k^* \end{bmatrix}^H \begin{bmatrix} \bar{\boldsymbol{x}}_k \\ \bar{\boldsymbol{x}}_k^* \end{bmatrix} \tag{4.7}$$

Then it follows that

$$\begin{aligned} \sigma^2_{k,R} &= \frac{1}{4} \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{f}_k^* \end{bmatrix}^H \mathrm{Cov}\left(\begin{bmatrix} \bar{\boldsymbol{x}}_k \\ \bar{\boldsymbol{x}}_k^* \end{bmatrix} \middle| s_k = \psi(\boldsymbol{\gamma}) \right) \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{f}_k^* \end{bmatrix} \\ &= \frac{1}{4} \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{f}_k^* \end{bmatrix}^H \begin{bmatrix} \boldsymbol{\Sigma}_{\bar{\boldsymbol{x}}_k | s_k} & \tilde{\boldsymbol{\Sigma}}_{\bar{\boldsymbol{x}}_k | s_k} \\ \tilde{\boldsymbol{\Sigma}}^*_{\bar{\boldsymbol{x}}_k | s_k} & \boldsymbol{\Sigma}_{\bar{\boldsymbol{x}}_k | s_k} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{f}_k^* \end{bmatrix}, \end{aligned}$$

where, assuming CWGN and symbols with uncorrelated real and imaginary components,

$$\begin{aligned}
\boldsymbol{\Sigma}_{\bar{\boldsymbol{x}}_k|s_k} &= \boldsymbol{H}\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}^H + \boldsymbol{\Sigma}_{\boldsymbol{w}} \\
\tilde{\boldsymbol{\Sigma}}_{\bar{\boldsymbol{x}}_k|s_k} &:= \mathrm{E}\big\{(\bar{\boldsymbol{x}}_k - \mathrm{E}\{\bar{\boldsymbol{x}}_k|s_k = \psi(\boldsymbol{\gamma})\})(\bar{\boldsymbol{x}}_k - \mathrm{E}\{\bar{\boldsymbol{x}}_k|s_k = \psi(\boldsymbol{\gamma})\})^t \,\big|\, s_k = \psi(\boldsymbol{\gamma})\big\} \\
&= \boldsymbol{H}\mathcal{D}(\boldsymbol{v}_{k,R} - \boldsymbol{v}_{k,I})\boldsymbol{H}^t,
\end{aligned}$$

for $\boldsymbol{v}_k$ from (3.6), and $\boldsymbol{v}_{k,R}$ defined as (4.8), and $\boldsymbol{v}_{k,I}$ defined as (4.9).

$$\boldsymbol{v}_{k,R} = [v_{0,R}, \ldots, v_{k-1,R}, 0, v_{k+1,R}, \ldots, v_{N-1,R}]^t \tag{4.8}$$

$$\boldsymbol{v}_{k,I} = [v_{0,I}, \ldots, v_{k-1,I}, 0, v_{k+1,I}, \ldots, v_{N-1,I}]^t \tag{4.9}$$

Putting the above equations together gives

$$\sigma^2_{k,R} = \frac{1}{4}\begin{bmatrix}\boldsymbol{f}_k \\ \boldsymbol{f}_k^*\end{bmatrix}^H \begin{bmatrix}\boldsymbol{H}\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}^H + \boldsymbol{\Sigma}_{\boldsymbol{w}} & \boldsymbol{H}\mathcal{D}(\boldsymbol{v}_{k,R} - \boldsymbol{v}_{k,I})\boldsymbol{H}^t \\ \boldsymbol{H}^*\mathcal{D}(\boldsymbol{v}_{k,R} - \boldsymbol{v}_{k,I})\boldsymbol{H}^H & \boldsymbol{H}^*\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}^t + \boldsymbol{\Sigma}^*_{\boldsymbol{w}}\end{bmatrix}\begin{bmatrix}\boldsymbol{f}_k \\ \boldsymbol{f}_k^*\end{bmatrix} \tag{4.10}$$

Using

$$Im\,(\hat{s}_k) = \frac{1}{2}\begin{bmatrix}\boldsymbol{f}_k \\ -\boldsymbol{f}_k^*\end{bmatrix}^H \begin{bmatrix}\bar{\boldsymbol{x}}_k \\ \bar{\boldsymbol{x}}_k^*\end{bmatrix}$$

and similar methods, we arrive at

$$\sigma^2_{k,I} = \frac{1}{4}\begin{bmatrix}\boldsymbol{f}_k \\ -\boldsymbol{f}_k^*\end{bmatrix}^H \begin{bmatrix}\boldsymbol{H}\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}^H + \boldsymbol{\Sigma}_{\boldsymbol{w}} & \boldsymbol{H}\mathcal{D}(\boldsymbol{v}_{k,R} - \boldsymbol{v}_{k,I})\boldsymbol{H}^t \\ \boldsymbol{H}^*\mathcal{D}(\boldsymbol{v}_{k,R} - \boldsymbol{v}_{k,I})\boldsymbol{H}^H & \boldsymbol{H}^*\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}^t + \boldsymbol{\Sigma}^*_{\boldsymbol{w}}\end{bmatrix}\begin{bmatrix}\boldsymbol{f}_k \\ -\boldsymbol{f}_k^*\end{bmatrix} \tag{4.11}$$

Note that the conditional variances do not actually depend on $\boldsymbol{\gamma}$. Thus it is not required to compute the conditional variances for each realization of $\psi(\boldsymbol{\gamma})$ individually.

Finally, in order to use the extrinsic estimate $\hat{s}_k$ and its conditional density to calculate the extrinsic LVs $\Delta L_{\text{IMSE}}(k,m)$, $p(\boldsymbol{x}|\boldsymbol{c} = \boldsymbol{\gamma})$ in (2.11) is replaced with $p(\hat{s}_k|s_k = \psi(\boldsymbol{\gamma}))$. This implies that

$$\Delta L_{\text{IMSE}}(k,m) = \ln \frac{\sum_{\boldsymbol{\gamma}\in\mathcal{G}^M_{m,0}} p\left(\hat{s}_k|s_k = \psi(\boldsymbol{\gamma})\right)\prod_{m'\neq m} P(c_{k,m'} = \gamma_{m'})}{\sum_{\boldsymbol{\gamma}\in\mathcal{G}^M_{m,1}} p\left(\hat{s}_k|s_k = \psi(\boldsymbol{\gamma})\right)\prod_{m'\neq m} P(c_{k,m'} = \gamma_{m'})}. \tag{4.12}$$

Using the conditionally Gaussian assumption described,

$$p(\hat{s}_k|s_k = \psi(\boldsymbol{\gamma})) \;\; = \;\; C \exp\left[-\frac{\mathrm{Re}^2(\hat{s}_k - \bar{s}_k)}{2\sigma_{k,R}^2} - \frac{\mathrm{Im}^2(\hat{s}_k - \bar{s}_k)}{2\sigma_{k,I}^2} \right].$$

Next, recall that $P(c_{k,m'} = \gamma_{m'})$ can be written as in (3.12). Substituting (3.12) into (4.12), the $\cosh(L_{\text{old}}(k,m')/2)$ contributions from numerator and denominator cancel (since they are invariant to $\gamma_{m'}$), and due to the remaining terms, $\Delta L_{\text{IMSE}}(k,m)$ becomes

$$\Delta L_{\text{IMSE}}(k,m) \;\; = \;\; \ln \frac{\sum_{\boldsymbol{\gamma} \in \mathcal{G}_{m,0}^M} \exp\left[-\frac{\mathrm{Re}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,R}^2} - \frac{\mathrm{Im}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,I}^2} + \frac{1}{2}\sum_{m' \neq m} (-1)^{\gamma_{m'}} L_{\text{old}}(k,m') \right]}{\sum_{\boldsymbol{\gamma} \in \mathcal{G}_{m,1}^M} \exp\left[-\frac{\mathrm{Re}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,R}^2} - \frac{\mathrm{Im}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,I}^2} + \frac{1}{2}\sum_{m' \neq m} (-1)^{\gamma_{m'}} L_{\text{old}}(k,m') \right]} \tag{4.13}$$

This completes the derivation of the extrinsic LV(s) $\Delta L_{\text{IMSE}}(k,m)$.

4.2 Structure of Iterations

Similar to the IMLE, the IMSE algorithm proceeds as follows. Prior to the first iteration, $\{L_{\text{old}}(k,m)\ \forall k,m\}$ are obtained from the output of a soft decoder, if available, or otherwise set to zero. These LVs are then used to initialize $\{\mu_k\}_{k=0}^{N-1}$, $\{v_{k,R}\}_{k=0}^{N-1}$, $\{v_{k,I}\}_{k=0}^{N-1}$ and $\{v_k\}_{k=0}^{N-1}$. We begin the first iteration by working on symbol index $k = 0$. The means $\{\mu_k\}_{k=0}^{N-1}$ are used for SIC, and the variances $\{v_k\}_{k=0}^{N-1}$ are used to calculate $\boldsymbol{f}_0$, which in turn are used to compute $\hat{s}_0$. Next, the statistics of the conditional densities for the extrinsic estimate, *viz.*, $\bar{s}_{0,\boldsymbol{\gamma}}$ for every $\boldsymbol{\gamma} \in \{0,1\}^M$, $\sigma_{0,R}^2$ and $\sigma_{0,I}^2$ are computed. Using these, $\hat{s}_0$ and $\{L_{\text{old}}(0,m)\}_{m=0}^{M-1}$, $\{\Delta L_{\text{IMLE}}(0,m)\}_{m=0}^{M-1}$ are calculated and used to compute $\{L(0,m)\}_{m=0}^{M-1}$. Finally, $\{L(0,m)\}_{m=0}^{M-1}$ are used to update μ_0, $v_{0,R}$, $v_{0,I}$ and v_0. Moving on to $k = 1$, the entire process listed above is repeated. The $k = 2$ case is tackled next, then $k = 3$, and so on, until $k = N-1$. Finally, $\{L(k,m)\ \forall k,m\}$ are copied to $\{L_{\text{old}}(k,m)\ \forall k,m\}$. This concludes the first iteration. The next (e.g., second) iteration begins again at symbol index $k = 0$ and proceeds through $k = N-1$. The algorithm terminates after a specified number of iterations.

The complexity of the algorithm is dominated by the $\mathcal{O}(N^3)$ computation of $\boldsymbol{f}_k$ for each symbol. A technique similar to the one mentioned for IMLE using the matrix inversion lemma can be used to obtain $\boldsymbol{f}_k$ as an update of $\boldsymbol{f}_{k-1}$. However, the cumbersome update only helps reduce the complexity of the algorithm to $\mathcal{O}(N^2)$. A much more effective complexity reduction technique, via the simplified system model in Sec. 3.3, is suggested in the following section.

4.3 Using the Simplified System Model

The simplified system model is used to reduce the complexity of the IMSE, too without sacrificing performance significantly. The idea is to use only the partial observation $\boldsymbol{x}_k$ and partial channel information $\boldsymbol{H}_k$ as defined in Sec. 3.3, respectively, to generate good local extrinsic estimates $\hat{s}_k$. Similar to the IMLE, μ_k is redefined as in Sec. 3.3, $\boldsymbol{v}_{k,R}$ and $\boldsymbol{v}_{k,I}$ as

$$\begin{aligned} \boldsymbol{v}_{k,R} &:= \left[v_{(k-2D,R)}, \ldots, v_{(k-1,R)}, 0, v_{(k+1,R)}, \ldots, v_{(k+2D,R)}\right]^t \\ \text{and, } \boldsymbol{v}_{k,I} &:= \left[v_{(k-2D,I)}, \ldots, v_{(k-1,I)}, 0, v_{(k+1,I)}, \ldots, v_{(k+2D,I)}\right]^t, \end{aligned}$$

respectively. Then $\boldsymbol{v}_k$ is defined as the sum of $\boldsymbol{v}_{k,R}$ and $\boldsymbol{v}_{k,I}$. Proceeding with the IMSE algorithm in this case results in extrinsic LV expressions similar to (4.13).

4.4 Computational Complexity and Other Similar Algorithms

IMSE bears similarity to the algorithms proposed in [15] and [16], though there are significant differences. First, in both [15] and [16], the estimation error is modeled as complex *circular* Gaussian. Simulation studies have shown that IMSE's provision for $\sigma_{k,I}^2 \neq \sigma_{k,R}^2$ is advantageous, especially when complex alphabets are used. Second, the algorithms in [15] update the symbol means and variances once per iteration (i.e., after updating $\{\Delta L(k,m), \forall m\}_{k=0}^{N-1}$), whereas IMSE updates the symbol means and variances

at every symbol index k. It was demonstrated in [16] that symbol-rate updating is advantageous. Finally, [15] uses the full observation vector $\boldsymbol{x}$, mandating the computation of an $N \times N$ inverse autocorrelation matrix, whereas IMSE (and [16]) use the partial observation $\boldsymbol{x}_k$, requiring only the computation of a $(2D+1)\times(2D+1)$ inverse autocorrelation matrix. Since, typically, $D \ll N$, this leads to significant computational savings.

4.5 IMSE Algorithm for BPSK and QPSK

In this section, we consider the BPSK and Gray Mapped QPSK constellations and simplify the expressions for extrinsic LV(s), the symbol mean and variances for the IMSE algorithm. It is assumed in this section that the simplified system model is used.

Recall that the symbol mapping function for BPSK ($M = 1$) was defined as

$$\psi(\boldsymbol{\gamma}) \;=\; (-1)^{\gamma_0}$$

Then, $\bar{s}_{k,\gamma} = (-1)^{\gamma} \boldsymbol{f}_k^H \boldsymbol{h}_{k,0} \in \mathbb{R}$, so that (4.13) simplifies to

$$\begin{aligned} \Delta L_{\text{IMSE}}(k,0) &= \ln \frac{\exp[-\operatorname{Re}^2(\hat{s}_k - \bar{s}_{k,0})/\sigma_{k,R}^2]}{\exp[-\operatorname{Re}^2(\hat{s}_k - \bar{s}_{k,1})/\sigma_{k,R}^2]} \\ &= \frac{2\operatorname{Re}(\hat{s}_k)\boldsymbol{f}_k^H \boldsymbol{h}_{k,0}}{\sigma_{k,R}^2}. \end{aligned}$$

On the other hand, the symbol mapping function for Gray Mapped QPSK ($M = 2$) was defined as

$$\psi(\boldsymbol{\gamma}) \;=\; \frac{1}{\sqrt{2}}\left[(-1)^{\gamma_0} + j(-1)^{\gamma_1}\right]$$

Therefore, in this case, (4.13) simplifies to

$$\Delta L_{\text{IMSE}}(k,0) \;=\; \ln \frac{\sum_{\gamma_1=0}^{1} \exp\left[-\frac{\text{Re}^2(\hat{s}_k-\bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,R}^2} - \frac{\text{Im}^2(\hat{s}_k-\bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,I}^2} + \frac{1}{2}(-1)^{\gamma_1} L_{\text{old}}(k,1)\right]}{\sum_{\gamma_1=0}^{1} \exp\left[-\frac{\text{Re}^2(\hat{s}_k-\bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,R}^2} - \frac{\text{Im}^2(\hat{s}_k-\bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,I}^2} + \frac{1}{2}(-1)^{\gamma_1} L_{\text{old}}(k,1)\right]} \tag{4.14}$$

$$\Delta L_{\text{IMSE}}(k,1) \;=\; \ln \frac{\sum_{\gamma_0=0}^{1} \exp\left[-\frac{\text{Re}^2(\hat{s}_k-\bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,R}^2} - \frac{\text{Im}^2(\hat{s}_k-\bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,I}^2} + \frac{1}{2}(-1)^{\gamma_0} L_{\text{old}}(k,0)\right]}{\sum_{\gamma_0=0}^{1} \exp\left[-\frac{\text{Re}^2(\hat{s}_k-\bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,R}^2} - \frac{\text{Im}^2(\hat{s}_k-\bar{s}_{k,\boldsymbol{\gamma}})}{2\sigma_{k,I}^2} + \frac{1}{2}(-1)^{\gamma_0} L_{\text{old}}(k,0)\right]} \tag{4.15}$$

The numerator of (4.14) uses $\bar{s}_{k,\boldsymbol{\gamma}} = \psi([0,\gamma_1])\boldsymbol{f}_k^H \boldsymbol{h}_{k,0}$ in forming

$$\text{Re}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}}) \;=\; \text{Re}^2(\hat{s}_k) - \sqrt{2}\,\text{Re}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0} + \frac{1}{2}(\boldsymbol{f}_k^H\boldsymbol{h}_{k,0})^2,$$

while the denominator of (4.14) uses $\bar{s}_{k,\boldsymbol{\gamma}} = \psi([1,\gamma_1])\boldsymbol{f}_k^H \boldsymbol{h}_{k,0}$ in forming

$$\text{Re}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}}) \;=\; \text{Re}^2(\hat{s}_k) + \sqrt{2}\,\text{Re}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0} + \frac{1}{2}(\boldsymbol{f}_k^H\boldsymbol{h}_{k,0})^2.$$

Since the numerator and denominator of (4.14) have identical $\text{Im}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}})$, we pull the $\exp[\frac{\sqrt{2}}{\sigma_{k,R}^2}\text{Re}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0}]$ terms out from the γ_1-summations, cancel common terms, and find

$$\Delta L_{\text{IMSE}}(k,0) \;=\; \frac{\sqrt{2}\,\text{Re}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0}}{\sigma_{k,R}^2}.$$

Similarly, the numerator of (4.15) uses $\bar{s}_{k,\boldsymbol{\gamma}} = \psi([\gamma_0,0])\boldsymbol{f}_k^H \boldsymbol{h}_{k,0}$ in forming

$$\text{Im}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}}) \;=\; \text{Im}^2(\hat{s}_k) - \sqrt{2}\,\text{Im}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0} + \frac{1}{2}(\boldsymbol{f}_k^H\boldsymbol{h}_{k,0})^2,$$

while the denominator of (4.15) uses $\bar{s}_{k,\boldsymbol{\gamma}} = \psi([\gamma_0,1])\boldsymbol{f}_k^H \boldsymbol{h}_{k,0}$ in forming

$$\text{Im}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}}) \;=\; \text{Im}^2(\hat{s}_k) + \sqrt{2}\,\text{Im}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0} + \frac{1}{2}(\boldsymbol{f}_k^H\boldsymbol{h}_{k,0})^2,$$

Since the numerator and denominator of (4.15) have identical $\text{Re}^2(\hat{s}_k - \bar{s}_{k,\boldsymbol{\gamma}})$, we pull the $\exp[\frac{\sqrt{2}}{\sigma_{k,I}^2}\text{Im}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0}]$ terms out from the γ_0-summations, cancel common terms, and find

$$\Delta L_{\text{IMSE}}(k,1) \;=\; \frac{\sqrt{2}\,\text{Im}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0}}{\sigma_{k,I}^2}.$$

The expressions for the mean symbol depend only on the updated LV and not on the algorithm that was used to obtain it. Hence they are identical to the ones derived in Sec. 3.5.

For the BPSK ($M = 1$) case,

$$\mu_k \;=\; \tanh(L_{\text{old}}(k,0)/2)$$

and for Gray Mapped QPSK ($M = 2$) case,

$$\mu_k \;=\; \frac{1}{\sqrt{2}}\tanh(L_{\text{old}}(k,0)/2) + \frac{j}{\sqrt{2}}\tanh(L_{\text{old}}(k,1)/2).$$

In order to calculate the symbol variances, it can be seen that $v_{k,R} = 1 - (\operatorname{Re}\mu_k)^2$ and $v_{k,I} = 0$ in the BPSK case, and that $v_{k,R} = \frac{1}{2} - (\operatorname{Re}\mu_k)^2$ and $v_{k,I} = \frac{1}{2} - (\operatorname{Im}\mu_k)^2$ in the QPSK case. Notice that, with uncorrelated real and imaginary symbol components, $v_k = v_{k,R} + v_{k,I}$.

Table 4.1 and Table 4.2 shows the steps in one iteration of the IMSE algorithm for BPSK and Gray Mapped QPSK constellations, respectively.

BPSK
`for` $k = 0 : N-1,$
$\boldsymbol{f}_k = \left(\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}_k} + \boldsymbol{H}_k\,\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}_k^H + \boldsymbol{h}_{k,0}\boldsymbol{h}_{k,0}^H\right)^{-1}\boldsymbol{h}_{k,0}$
$\hat{s}_k = \boldsymbol{f}_k^H\left(\boldsymbol{x}_k - \boldsymbol{H}_k\boldsymbol{\mu}_k\right)$
`compute` $\sigma_{k,R}^2$ `from (4.10)`
$L(k,0) = L_{\text{old}}(k,0) + 2\sigma_{k,R}^{-2}\operatorname{Re}(\hat{s}_k)\boldsymbol{f}_k^H\boldsymbol{h}_{k,0}$
$\mu_k = \tanh\left(\frac{L(k,0)}{2}\right)$
$v_k = 1 - \mu_k^2$
$L_{\text{old}}(k,0) = L(k,0)$
`end`

Table 4.1: Steps in one iteration of IMSE for BPSK.

QPSK
for $k = 0 : N-1,$
$\boldsymbol{f}_k = \left(\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}_k} + \boldsymbol{H}_k \mathcal{D}(\boldsymbol{v}_k) \boldsymbol{H}_k^H + \boldsymbol{h}_{k,0} \boldsymbol{h}_{k,0}^H\right)^{-1} \boldsymbol{h}_{k,0}$
$\hat{s}_k = \boldsymbol{f}_k^H \left(\boldsymbol{x}_k - \boldsymbol{H}_k \boldsymbol{\mu}_k\right)$
compute $\sigma_{k,R}^2$ and $\sigma_{k,I}^2$ from (4.10)-(4.11)
$L(k,0) = L_{\text{old}}(k,0) + \sqrt{2}\sigma_{k,R}^{-2} \operatorname{Re}(\hat{s}_k) \boldsymbol{f}_k^H \boldsymbol{h}_{k,0}$
$L(k,1) = L_{\text{old}}(k,1) + \sqrt{2}\sigma_{k,I}^{-2} \operatorname{Im}(\hat{s}_k) \boldsymbol{f}_k^H \boldsymbol{h}_{k,0}$
$\mu_k = \frac{1}{\sqrt{2}} \tanh\left(\frac{L(k,0)}{2}\right) + \frac{j}{\sqrt{2}} \tanh\left(\frac{L(k,1)}{2}\right)$
$v_{k,R} = \frac{1}{2} - (\operatorname{Re} \mu_k)^2,\ v_{k,I} = \frac{1}{2} - (\operatorname{Im} \mu_k)^2$
$L_{\text{old}}(k,0) = L(k,0),\ \ L_{\text{old}}(k,1) = L(k,1)$
end

Table 4.2: Steps in one iteration of IMSE for QPSK.

4.6 Relation between IMSE and IMLE

Starting from the optimal detection strategy for transmitted symbols and trading off performance for affordable complexity, two algorithms, *viz.*, IMLE and IMSE, are developed in this thesis. In the light of this common starting point, it seems imperative to probe if the IMSE and IMLE are related in some way. In this context, this section shows that the two algorithms developed are similar. In particular, for specific choices, the two algorithms are identical. This section shows that for the BPSK constellation, the IMLE is is identical to the IMSE, when the estimation error for the extrinsic estimates is assumed to be circular-Gaussian distributed. Note that here, we use the general expressions for IMLE and IMSE and *not* ones using the simplified system model. However, this equivalence can be shown for the algorithms for the simplified system model following the same procedure.

Consider that at some iteration, all the LVs $\{L_{\text{old}}(k,m) : k = 1, 2, \cdots, N\ , m = 1, 2, \cdots, M\}$ obtained by IMLE are identical to those obtained by the IMSE. Consider

also that the most current symbol means $\{\mu_k\}$ and variances $\{v_k\}$ are identical for IMLE and IMSE. This can be assumed since the symbol means and variances are computed solely using the updated LVs.

The extrinsic LV for the BPSK constellation using IMLE is given by

$$\begin{aligned} L_{\text{IMLE}}(k,0) &= 2\,\mathrm{Re}\, g_k \\ &= 2\,\mathrm{Re}\left(\boldsymbol{y}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_k\right) \end{aligned} \tag{4.16}$$

as shown in Sec. 3.5.

Now, for IMSE under the aforesaid assumption, the extrinsic LV is given by

$$L_{\text{IMSE}}(k,0) = \ln \frac{\exp\left(\frac{-|\hat{s}_k - \boldsymbol{f}_k^H \boldsymbol{h}_k|^2}{2\sigma_k^2}\right)}{\exp\left(\frac{-|\hat{s}_k + \boldsymbol{f}_k^H \boldsymbol{h}_k|^2}{2\sigma_k^2}\right)}. \tag{4.17}$$

In (4.17), σ_k^2 is defined and calculated in (4.18).

$$\begin{aligned} \sigma_k^2 &= E\left[\hat{s}_k | s_k = \beta\right], \qquad \beta \in \mathbb{S} \\ &= \boldsymbol{f}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k} \boldsymbol{f}_k \end{aligned} \tag{4.18}$$

Note that σ_k^2 does not depend on β.

Again, in (4.17), the extrinsic estimate $\hat{s_k}$ is calculated as

$$\hat{s}_k = \boldsymbol{f}_k^H \left(\boldsymbol{x} - \boldsymbol{H}\boldsymbol{\mu}_k\right),$$

where using (3.2) , we obtain

$$\hat{s}_k = \boldsymbol{f}_k^H \boldsymbol{y}_k. \tag{4.19}$$

Now, using the matrix inversion lemma, $\boldsymbol{f}_k$ can be written as (4.20).

$$\begin{aligned} \boldsymbol{f}_k &= \left(\boldsymbol{\Sigma}_{\boldsymbol{w}} + \boldsymbol{H}\,\mathcal{D}(\boldsymbol{v}_k)\boldsymbol{H}^H + \boldsymbol{h}_k \boldsymbol{h}_k^H\right)^{-1} \boldsymbol{h}_k \\ &= \left(\boldsymbol{\Sigma}_{\boldsymbol{q}_k} + \boldsymbol{h}_k \boldsymbol{h}_k^H\right)^{-1} \boldsymbol{h}_k \\ &= \frac{\boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_k}{1 + \boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_k} \end{aligned} \tag{4.20}$$

Using (4.18) and (4.19) in (4.17), the extrinsic LV can be simplified to

$$L_{\text{IMSE}}(k,0) \;=\; \frac{2\,\text{Re}\left(\boldsymbol{h}_k^H \boldsymbol{f}_k \boldsymbol{f}_k^H \boldsymbol{y}_k\right)}{\boldsymbol{f}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k} \boldsymbol{f}_k}. \tag{4.21}$$

Now substituting (4.20) in (4.21), we write

$$L_{\text{IMSE}}(k,0) \;=\; \frac{2\,\text{Re}\left(\boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{q}_k} \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_k \boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{y}_k\right) \left(1 + \boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_k\right)^2}{\left(\boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{q}_k} \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_k\right) \left(1 + \boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_k\right)^2}. \tag{4.22}$$

Finally, noting that $\boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{h}_k \in \mathbb{R}$ in the expression above, we obtain

$$L_{\text{IMSE}}(k,0) \;=\; 2\,\text{Re}\left(\boldsymbol{h}_k^H \boldsymbol{\Sigma}_{\boldsymbol{q}_k}^{-1} \boldsymbol{y}_k\right). \tag{4.23}$$

Then $L_{\text{IMLE}}(k,0) = L_{\text{IMSE}}(k,0)$ from (4.16) and (4.23). The calculation of the extrinsic LVs is the only step that differs in the two algorithms. Also, the values of $\{L_{\text{old}}(k,m)\}$, $\{\mu_k\}$ and $\{v_k\}$ are initialized to the same values for both algorithms. Then it can be seen that the algorithms are identical.

This concludes the discussion on the equalizer algorithms.

CHAPTER 5

Error Control Coding

The aim of this thesis is to evaluate the performance of the designed equalizers in a coded system with a receiver configured as a turbo-equalizer (or as an iterative equalizer followed by a SISO decoder). Hence, the design of the error control code and the corresponding decoder is not the primary focus of this thesis. However, the error control coder and decoder are an integral part of the turbo-equalization setup. Thus a short discussion on this aspect of the problem is presented. The selected code is described. The choice of the decoder is also discussed. The final section of this chapter focuses on the choice of the interleaver.

5.1 Selected Coding Scheme

The turbo equalization idea has been used in a number of applications. In keeping with most of these, we choose to use convolutional codes in our system. We choose to use a code with relatively small memory in order to have a decoder with relatively low complexity. More specifically, we wish to use the convolutional code with memory $\nu = 2$ with the generator matrix

$$g(D) = \begin{bmatrix} 1 + D^2 & 1 + D + D^2 \end{bmatrix} \tag{5.1}$$

In order to feed back soft information (in the form of LVs) to the equalizer as shown in Fig. 2.3, we require a soft-input soft-output (SISO) coder. The input to this block is a stream of LVs for input bits. As its output, it generates the LVs for the bits that are the result of convolutionally encoding the input bits using the convolutional code specified by (5.1). In order to achieve this, we use results from [22].

The first result states that given statistically independent $a_1, a_2 \in \{0, 1\}$, the LV of $a_1 \oplus a_2$, $L(a_1 \oplus a_2)$, is given by

$$\begin{aligned} L(a_1 \oplus a_2) &= \ln \frac{1 + e^{L(a_1)} e^{L(a_2)}}{e^{L(a_1)} + e^{L(a_2)}} \\ &\approx \operatorname{sgn}(L(a_1)) \operatorname{sgn}(L(a_2)) \min\{|L(a_1)|, |L(a_2)|\}, \end{aligned} \tag{5.2}$$

where $L(a_i)$ is the LV for bit a_i, $i = 1, 2$. Building on this, it can be shown that for statistically independent $a_i \in \{0, 1\}$ where, $i = 1, 2, \cdots, M$,

$$\begin{aligned} L(a_1 \oplus a_2 \oplus \cdots \oplus a_M) &= 2 \tanh^{-1} \left(\Pi_{i=1}^{M} \tanh(L(a_i)/2) \right) \\ &\approx \left(\Pi_{i=1}^{M} \operatorname{sgn}(L(a_i)) \right) \min\{|L(a_1)|, |L(a_2)|, \cdots, |L(a_M)|\} \end{aligned} \tag{5.3}$$

5.2 Non-Systematic Codes versus Recursive Systematic Codes

For our turbo-equalizers, we prefer to use the Non-Systematic generator (NSG) matrix (5.1) over the Recursive Systematic generator (RSG) matrix given in (5.4).

$$g_1(D) = \begin{bmatrix} 1 & \dfrac{1 + D^2}{1 + D + D^2} \end{bmatrix} \tag{5.4}$$

This choice seems counter-intuitive from a Turbo-Codes perspective which suggests that codes defined by RSGs outperform those defined by the corresponding NSGs. But,

codes defined by RSGs run into problems with the SISO-coder. It is clear that for a RSG, the parity bit at any point has the influence of all previous inputs. Equation (5.3) shows that the magnitude of the LV at the output of the SISO coder is related to the LV with the minimum magnitude encountered at the input. Thus even if a single bit is unreliable in the input, all subsequent outputs have very low reliability. This adversely affects the equalizer when these LVs are fed back. In contrast, the output LV of from a SISO coder using a NSG depends only on a maximum of $\nu + 1$ input LVs and performs better.

5.3 Choice of Decoder

A number of decoders are available for decoding convolutional codes. Again, we stress that we wish to evaluate the performance of the equalizers in a system with coding. Thus we choose the bit error rate optimal MAP decoder, hoping that no performance loss is incurred by the decoder. Specifically, we use the well-known MAP optimal BCJR algorithm in [23]. In order to keep the decoding complexity low, we use codes with small memory (small nu).

5.4 Interleaver

As seen from Fig. 2.3, an interleaver is used in the transmitter and a corresponding de-interleaver is used in the receiver. In our experiments, we use a block interleaver. This interleaver stores the values into a matrix row-wise and generates the output by reading out the stored values column-wise. This works well, since the function of the interleaver is merely to scramble bits at the output of the coder to destroy the correlation introduced as far as possible. The interleaver makes our assumption of independent bits for the IMLE and IMSE seem more realistic. The error events at the output of the equalizers are highly

bursty in nature. The interleaver helps distribute these error events evenly in the stream input to the decoder.

This concludes the short discussion on the error control coding scheme used. All the relevant blocks of the proposed system have now been discussed. In the following chapter, we present simulation results and draw conclusions on the efficacy of the algorithms developed.

CHAPTER 6

Experimental Results and Conclusion

In this chapter, the performance of the designed equalizers is characterized through simulation results. First, the setup for the simulations is described. Next, the pulse shapes and their success in shaping the channel into the desired banded structure of Fig. 2.1 is validated. Then benchmark/reference configurations are introduced against which the performance of the equalizers is measured. The performance of the equalizers is presented. Important conclusions that are drawn are listed. Finally, the last section presents pointers for possible future research directions.

6.1 Experimental Setup

The experiments performed employed BPSK and (Gray-mapped) QPSK constellations. SNR^{-1}-variance CWGN noise is used along with a WSSUS Rayleigh-fading channel with uniform power profile (i.e., $\sigma_l^2 = N_h^{-1}$ for $0 \leq l < N_h$). For the channel and pulse shapes, the design choices made are $N_a = 1.5N_s$, $N_b = N_a + N_h/2$, and $D = \lceil f_{\mathsf{d}} N \rceil + 1$ (from [9, 16]). We note here that the pulse shape designs only use channel statistics. Even though pulse shaping gives $\boldsymbol{H}$ the desired structure there are small leakages for practical channel realizations. Thus, the equalization algorithms use a slightly larger value of D while using

the simplified system model than that used for the pulse design. Usually the value of D_{eq} for the equalizer is set to $D_{eq} = D + \Delta D$. We choose $\Delta D = 1$.

For the MCM, $N = 64$, $N_h = 32$, and $N_s = N$ (i.e., no guard interval) are chosen and the system studied for maximum Doppler frequencies $f_{\mathsf{d}} = 0.03$ and $f_{\mathsf{d}} = 0.01$. Recall that f_d is normalized to the chip rate, *i.e.*, $r_{\mathsf{t}}(q) = J_0(2\pi f_d q)$. Perfect channel knowledge is assumed. Hence, pilots are not employed. Note that since N_h is not larger than N, the block DFE configuration suggested [9] is not used for the simulation results presented.

At the transmitter, i.i.d. information bits are coded, interleaved, and mapped to symbols. Coding and interleaving is performed on blocks of 40 multi-carrier symbols. The block interleaver Π ensures that bits being mapped to symbols have very low correlation. These symbols are then modulated by the PS-FDM transmitter described in Sec. 2.1 and transmitted through the doubly selective channel, whose realizations were generated using Jakes method. At the receiver, the windowed frequency domain observation $\boldsymbol{x}^{(i)}$ is computed according to (2.4) and passed to the equalization/decoding stage. Both serial and turbo equalization/decoding configurations (i.e., Fig. 2.2 and Fig. 2.3, respectively) are simulated, each with IMLE and IMSE algorithms. Note that only the low-complexity versions of the IMSE and IMLE resulting from the simplified system model of Sec. 3.3 are simulated, since the algorithms in their original form have very high computational complexity. For turbo equalization/decoding, one equalizer iteration is inserted between each decoding iteration. Sixteen turbo iterations are used for both BPSK and QPSK constellations. Increasing the number of turbo iterations further did not enhance performance. For each BER data point, we average over 3000 multi-carrier symbols.

6.2 Pulse Shapes and Channel Profile

Out of the three pulse shape designs mentioned in Sec. 2.2, we consider only the first two, *viz.*, the scheme employing optimized pulses at the transmitter with CP-OFDM like pulses at the receiver and the scheme employing jointly optimized pulses. The third scheme mentioned in Sec. 2.2 employs guard intervals, and does not have full rate. However, the interested reader is referred to [16] for results on this scheme.

Fig. 6.1-Fig. 6.4 show the SINR maximizing pulse shapes and corresponding shaped power profile for the channel. Note that our design choices imply $L_{\mathsf{pre}} = 1$ and $L_{\mathsf{pst}} = 1$. Thus, for each pair of pulses $\{a_n\}$ and $\{b_n\}$, the three channel power profiles (in dB) correspond to (b) the pre-cursor $\boldsymbol{H}^{(i,-1)}$, (c) the cursor $\boldsymbol{H}^{(i,0)}$ and (d) the post-cursor $\boldsymbol{H}^{(i,1)}$, respectively(from left to right). Note that all the plots are at SNR $= 2$dB. From the plots, it is clear that the power of the pre- and post-cursor channels are negligibly small. This supports the choice of not using a block DFE as suggested in [9]. It is also clear that the obtained channel profile agrees with the target response, barring small amounts of leakage.

6.3 Performance References Chosen

In order to characterize the performance of the IMLE and IMSE, the following performance references/benchmarks are used:

- **Perfect Global Interference Canceler** (PGIC): This scheme has one equalization and one decoding iteration, in which, the equalization step is MMSE estimation of $s_k^{(i)}$ assuming all interference $\{s_d^{(j)}\}_{(j,d)\neq(i,k)}$ is known perfectly. This can be viewed as the best performance achievable by the IMSE and IMLE using the entire cursor channel matrix $\boldsymbol{H}^{(i,0)}$.

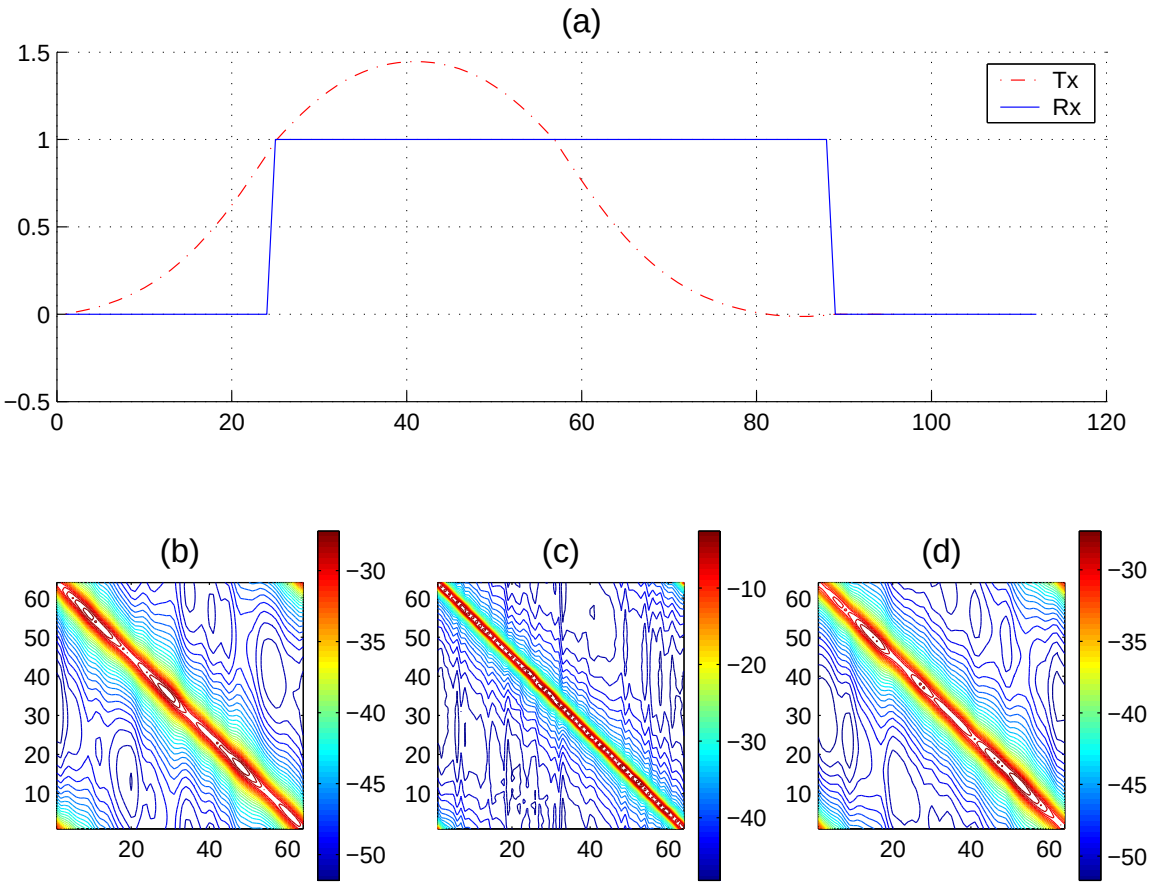

Figure 6.1: Optimized transmitter pulse shapes and channel profiles for $f_d = 0.01$, (a) pulse shapes, (b) pre-cursor, (c)cursor and (d) post-cursor MIMO channels.

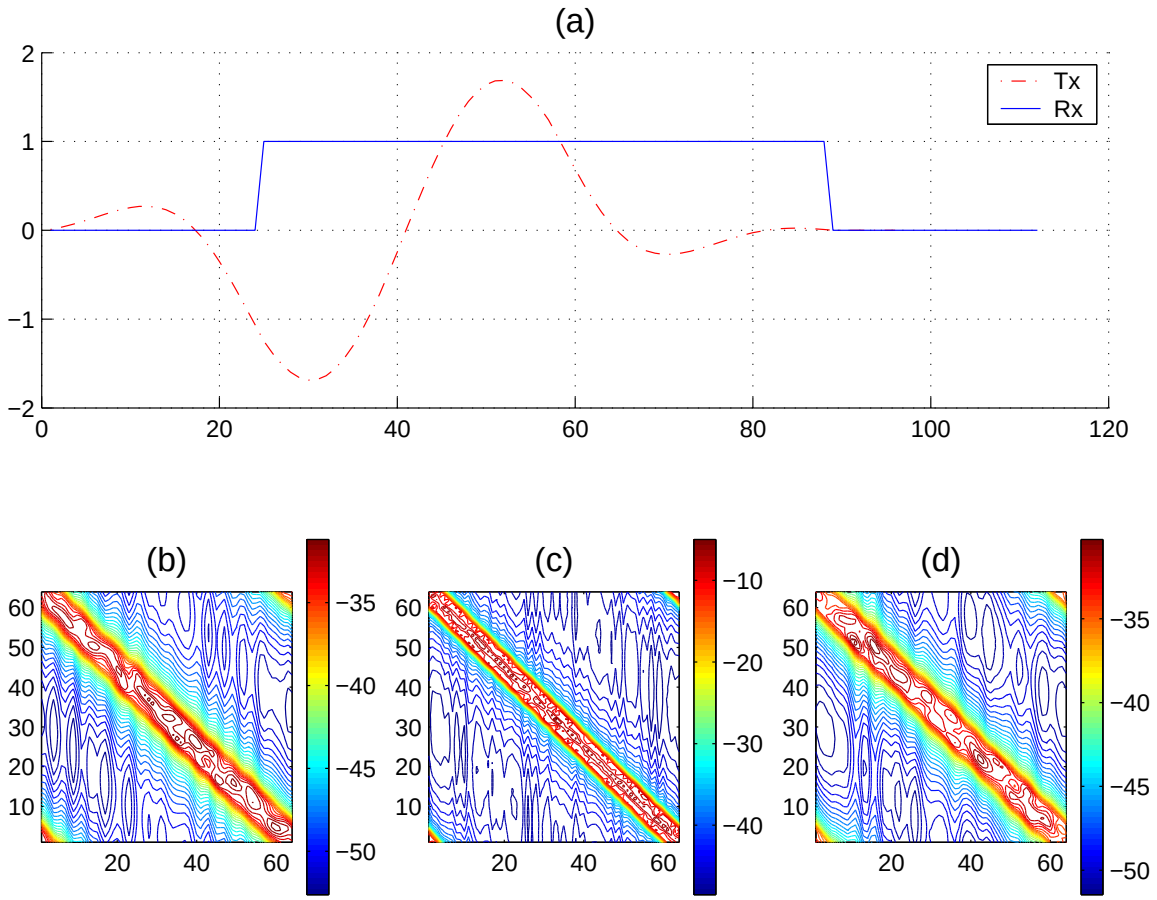

Figure 6.2: Optimized transmitter pulse shapes and channel profiles for $f_d = 0.03$, (a) pulse shapes, (b) pre-cursor, (c) cursor and (d) post-cursor MIMO channels.

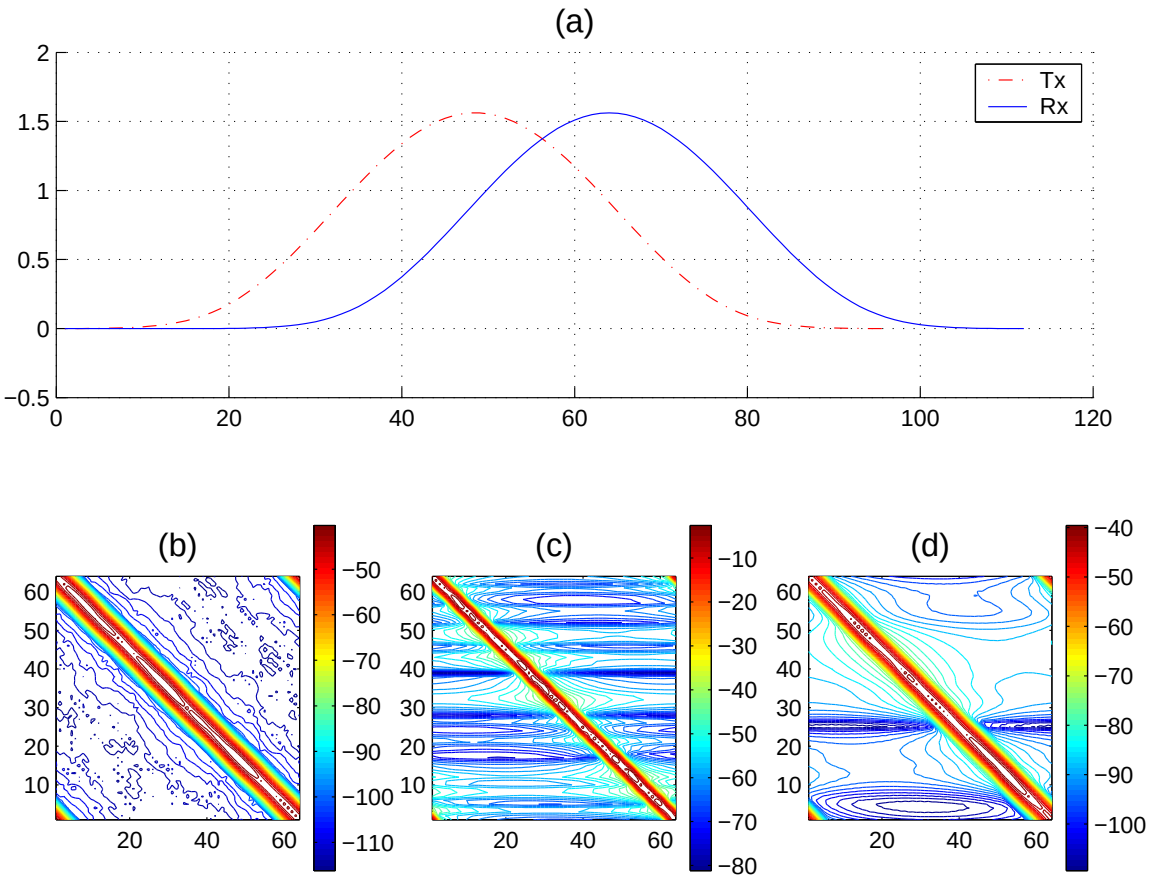

Figure 6.3: Jointly optimized pulse shapes and channel profiles for $f_d = 0.01$, (a) pulse shapes, (b) pre-cursor, (c) cursor and (d) post-cursor MIMO channels.

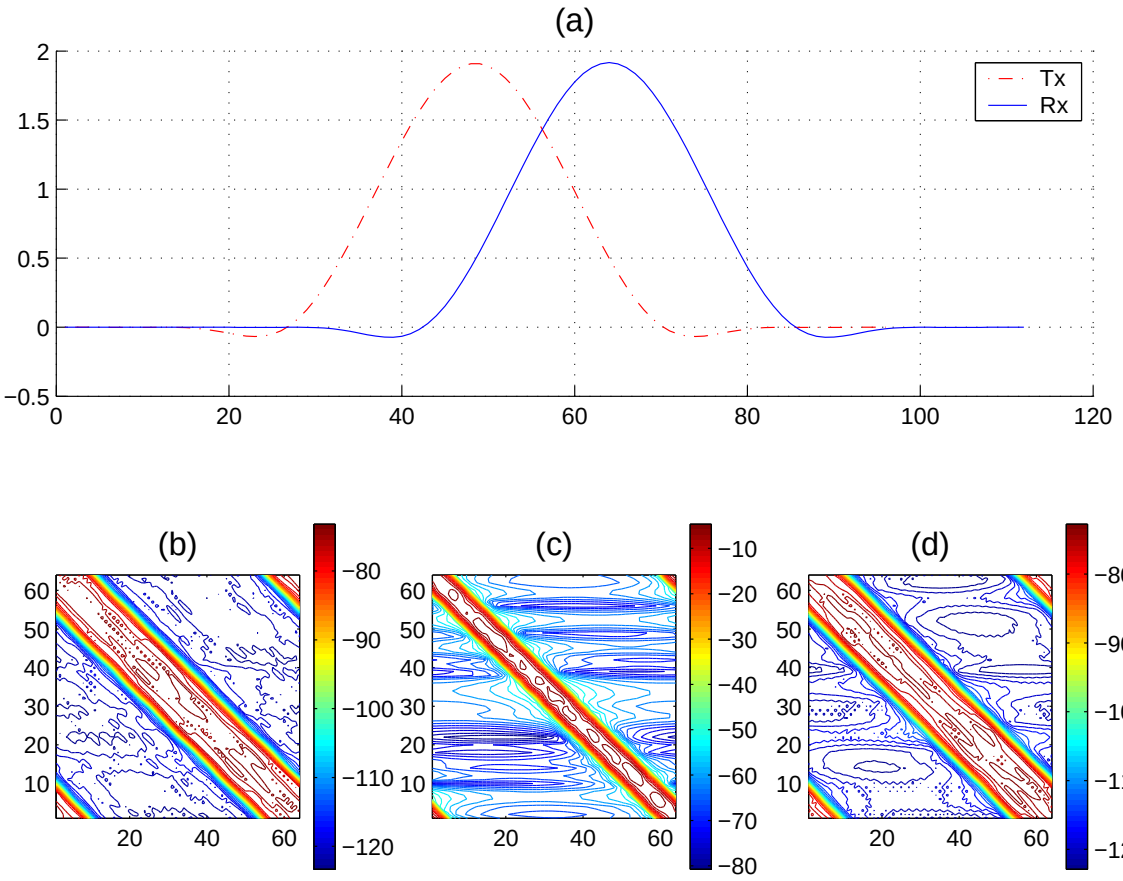

Figure 6.4: Jointly optimized pulse shapes and channel profiles for $f_d = 0.03$, (a) pulse shapes, (b) pre-cursor, (c) cursor and (d) post-cursor MIMO channels.

- **Perfect Local Interference Canceler** (PLIC): This scheme has one equalization and one decoding iteration, in which, the equalization step is similar to PGIC except that, when estimating $s_k^{(i)}$, only neighboring co-cursor ICI $\{s_d^{(i)}\}_{d=k-2D}^{k+2D}$ is known; ISI and non-neighboring ICI are unknown. This PLIC lower bounds the BER of the simulated reduced complexity iterative equalizers, since, at best, they too cancel only local interference. (It is interesting to note that the proximity of PLIC and PGIC performance curves measures the success of the pulse design's out-of-target interference suppression ability.)

  It is interesting to note that the PGIC and PLIC are developed from MMSE estimation. However, it is trivial to show that under the assumptions for PGIC and PLIC, IMLE acheives the same bounds. Hence the PGIC and PLIC are performance bounds for the IMLE, too.

- **Complex White Gaussian Noise Channel** (WN): This refers to the performance of the convolutional code over a CWGN channel, representing BER performance achieved in the absence of *interference* and *fading* using a BER optimal MAP decoder. This represents bounds the performance attainable using the selected code over any CWGN channel with or without interference.

- **Linear MMSE Equalizer** (LIN): This refers to the performance of the MCM system with standard linear MMSE based equalization and one decoding iteration, where the linear MMSE estimates are generated using (6.1).

$$\boldsymbol{s}^{(i)} \;=\; \boldsymbol{H}^{(i,0)H}\left(\boldsymbol{H}^{(i,0)}\boldsymbol{H}^{(i,0)H}+\boldsymbol{\Sigma}_{\boldsymbol{w}}\right)^{-1}\boldsymbol{x}^{(i)} \tag{6.1}$$

  The benefits of exploiting available LVs and the finite alphabet of constellations is highlighted by comparing the equalizers designed with linear MMSE equalization.

6.4 Performance Characteristics

Here we characterize the performance of the proposed pulse-shaped multi-carrier modulation system employing iterative equalization and coding. End-to-end coded BER is used as a performance measure for the proposed iterative equalizers. Note that, in order to provide a fair comparison across constellations, BER is plotted against the ratio of information-bit energy to noise spectral density. In these BER plots, MLk and MSk refer to IMLE and IMSE, respectively, with k decoding iterations. In particular,ML1 and MS1 represent the performance of the IMSE and IMLE using one-shot equalization and decoding. This is the performance of the receiver when used in the configuration shown in Fig. 2.2.

Fig. 6.5 depicts the performance of the equalizers in a system using optimized pulses at transmitter only, whereas, Fig. 6.6 depicts the performance of the equalizers in a system using jointly optimized pulses when a BPSK constellation is used. Fig. 6.7 and Fig. 6.8 depict the performance of optimal transmitter pulses and jointly optimal pulses, respectively using the QPSK constellation.

6.5 Conclusions

From the performance characteristics presented, the following conclusions are drawn.

- **Bounding Performance**:

  In Figs. 6.5-6.8, the gap between the PLIC and PGIC curves is less than 1dB for all cases, implying that, with the pulse designs, the out-of-target ICI/ISI is negligible, even at the high Doppler frequency of $f_d = 0.03$. This also means that a very small sacrifice in performance was made, by choosing to use the simplified system model, in spite of the large computational savings it provides. Also notice that, for either constellation, PGIC/PLIC for $f_d = 0.03$ is lower than $f_d = 0.01$. Recall that D

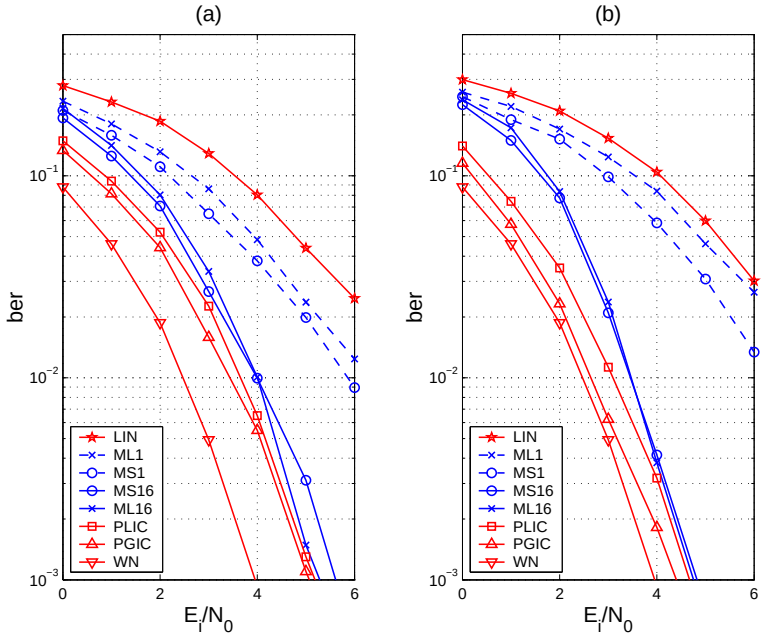

Figure 6.5: BPSK with transmitter pulses and CP-OFDM receiver window for (a) $f_d = 0.01$ and (b) $f_d = 0.03$.

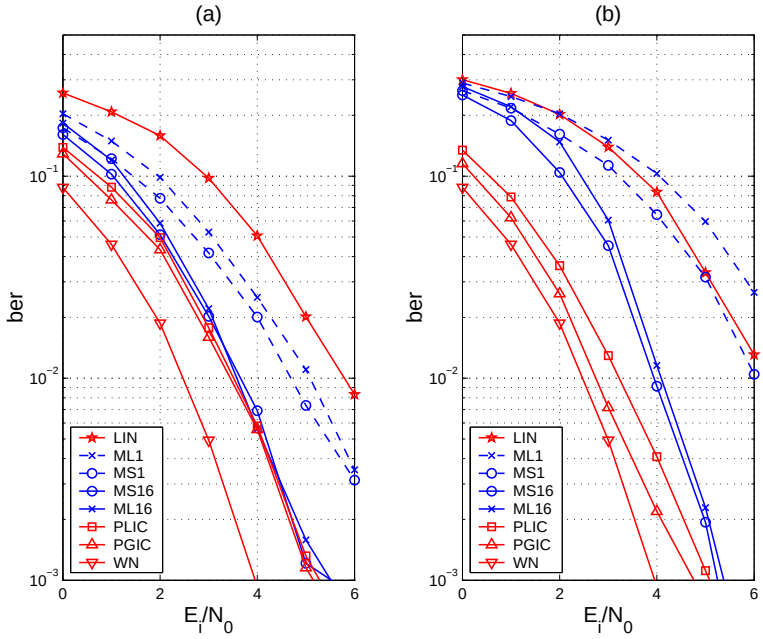

Figure 6.6: BPSK with jointly optimized pulses for (a) $f_d = 0.01$ and (b) $f_d = 0.03$.

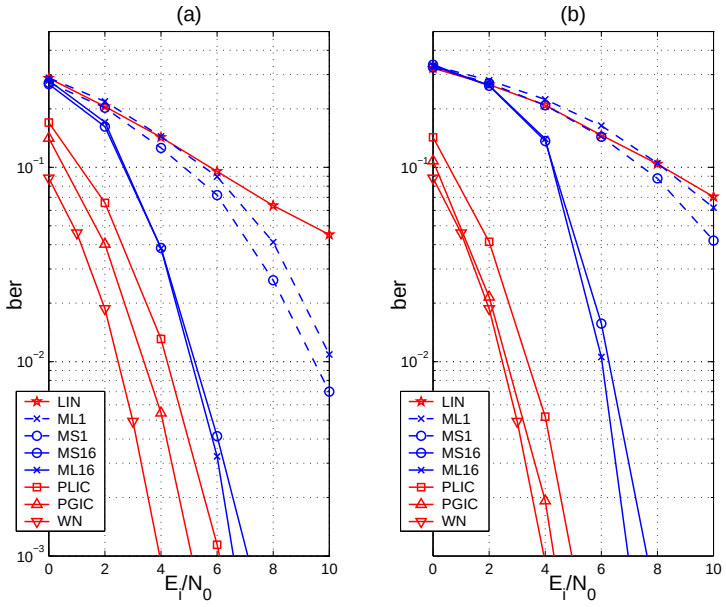

Figure 6.7: QPSK with transmitter pulses and CP-OFDM receiver window for (a) $f_d = 0.01$ and (b) $f_d = 0.03$

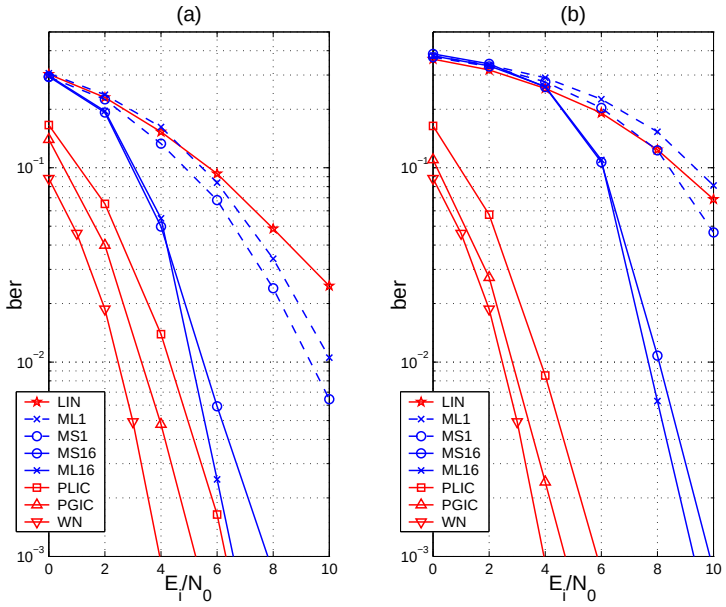

Figure 6.8: QPSK with jointly optimized pulses for (a) $f_d = 0.01$ and (b) $f_d = 0.03$

increases with f_d and provides diversity gain that could be exploited at the cost of complexity.

- **Performance of IMLE versus IMSE**:

  The BER plots for BPSK constellations show that the performance of the equalizers in all cases considered approach the PLIC at about an SNR of 5dB. This tells us that for BPSK, both IMSE and IMLE are near optimal among all equalizers that use the simplified system model.

  The BER plots for QPSK show that IMLE slightly outperforms IMSE at high SNR. We conjecture that IMSE is less robust to violations in the Gaussian interference assumption that are more likely to occur in high-SNR scenarios.

- **Comparison across Pulse Shaping Schemes**:

  Figure 6.7-6.8 suggest that the transmitter-only and joint transmitter/receiver pulse designs give similar performance at $f_{\mathsf{d}} = 0.01$ and the transmitter-only scheme is superior at $f_{\mathsf{d}} = 0.03$. While the joint design is expected to yield less out-of-target ISI/ICI, non-rectangular receiver windowing colors the frequency-domain noise, rendering the simplified model of Sec. 3.3 sub-optimal. Hence, we conjecture that the advantages of the lack of noise coloration outweighs the burden of increase in out-of-target ISI/ICI at higher Doppler for the designed equalizers.

- **Comparison across changing f_{d}**:

  We note that, as f_{d} increases, PGIC/PLIC performance improves (as a consequence of diversity) while IMLE/IMSE performance degrades. We attribute this degradation to error propagation during iterative equalization. Even so, we find that IMLE performance is within 1.5dB the PLIC bound at $f_d = 0.03$ when transmitter-only scheme

is used with QPSK. In all cases, it is clear that the algorithms perform significantly better than an $\mathcal{O}(N^3)$ standard non-iterative linear MMSE (LIN) based equalization strategy. This is remarkable considering that all the equalizers considered have $\mathcal{O}(ND^3)$ complexity.

6.6 Final Remarks

We presented a new approach to equalization for PS-FDM in the presence of doubly-dispersive fading. For suitably constructed pulse sequences to shape ICI/ISI, it is demonstrated that the affordable complexity equalization algorithms described come close to mitigating a substantial part of the interference present.

6.7 Future Work

A few of the interesting ideas that still need investigation include:

- **Information-Theoretic Analysis of Pulse Shaping**: Communication over doubly dispersive channels has received a lot of attention. Pulse shaped multi-carrier modulation is one such scheme. Even though some recent work has touched on it, there is almost no literature on the information theoretic analysis of pulse shapes, thier optimality, and their limiting performance over doubly-selective channels. For instance, the pulses are designed using a SINR maximizing criterion. However, whether this criterion is optimal or not is not well understood.
- **Extension of Pulse Shaping Schemes to Multi-Antenna Systems**: Multi-antenna systems are known to provide large spatial diversity gains. Extension of the pulse shaping schemes to multi-antenna systems, along with suitable equalization and/or coding algorithms will be valuable.

- **Analysis of Iterative Equalizers**: The iterative equalizers proposed in my Master's Thesis work very well with the pulse shaped multi-carrier modulation system. However, the choice seems rather ad-hoc. In particular, there are no well defined grounds on which we can compare iterative equalizers (among themselves and with other high performance equalization algorithms). It would be valuable to find methods of comparing iterative equalization algorithms.

BIBLIOGRAPHY

[1] G. L. Stüber, *Principles of Mobile Communication*. Kluwer Academic Publishers, 2nd ed., 2001.

[2] S. B. Weinstein and P. M. Ebert, "Data transmission by frequency division multiplexing using the discrete Fourier transform," *IEEE Trans. Commun.*, vol. 19, pp. 628–634, Oct. 1971.

[3] L. J. Cimini, Jr., "Analysis and simulation of a digital mobile radio channel using orthogonal frequency division multiplexing," *IEEE Trans. Commun.*, vol. 33, pp. 665–765, July 1985.

[4] J. A. C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Commun. Mag.*, vol. 28, pp. 5–14, May 1990.

[5] A. Stamoulis, S. N. Diggavi, and N. Al-Dhahir, "Intercarrier interference in MIMO OFDM," *IEEE Trans. Signal Processing*, vol. 50, pp. 2451–2464, Oct. 2002.

[6] Y.-S. Choi, P. J. Voltz, and F. A. Cassara, "On channel estimation and detection for multicarrier signals in fast and selective Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 49, pp. 1375–1387, Aug. 2001.

[7] X. Cai and G. B. Giannakis, "Low-complexity ICI suppression for OFDM over time- and frequency-selective Rayleigh fading channels," in *Proc. Asilomar Conf. Signals, Systems and Computers*, Nov. 2002.

[8] A. Gorokhov and J.-P. Linnartz, "Robust OFDM receivers for dispersive time-varying channels: Equalization and channel acquisition," *IEEE Trans. Commun.*, vol. 52, pp. 572–583, Apr. 2004.

[9] P. Schniter, "A new approach to multicarrier pulse design for doubly-dispersive channels," in *Proc. Allerton Conf. Commun., Control, and Computing*, Oct. 2003.

[10] S. Das and P. Schniter, "A new pulse shaped frequency division multiplexing technique for doubly dispersive channels," in *Proc. Asilomar Conf. Signals, Systems and Computers*, 2004.

[11] G. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of inter symbol interference," *IEEE Trans. Inform. Theory*, vol. 18, pp. 363–378, May 1972.

[12] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, pp. 1046–1061, July 1999.

[13] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 4th ed., 2001.

[14] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *European Trans. Telecommun.*, vol. 6, pp. 507–511, Sept.-Oct. 1995.

[15] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean square error equalization using *a priori* information," *IEEE Trans. Signal Processing*, vol. 50, pp. 673–683, Mar. 2002.

[16] P. Schniter, "Low-complexity equalization of OFDM in doubly-selective channels," *IEEE Trans. Signal Processing*, vol. 52, pp. 1002–1011, Apr. 2004.

[17] D. Pham, J. Luo, K. R. Pattipati, and P. K. Willett, "A PDA-Kalman approach to multiuser detection in asynchronous CDMA," *IEEE Commun. Letters*, vol. 6, pp. 475–477, Nov. 2002.

[18] S. Liu and Z. Tian, "Near-optimal soft decision equalization for frequency selective MIMO channels," *IEEE Trans. Signal Processing*, vol. 52, pp. 721–733, March 2004.

[19] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer, 2nd ed., 1994.

[20] M. Tüchler, R. Koetter, and A. C. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, pp. 754–767, May 2002.

[21] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[22] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar 1996.

[23] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.