BLIND, ADAPTIVE EQUALIZATION FOR MULTICARRIER RECEIVERS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Richard K. Martin

May 2004

BLIND, ADAPTIVE EQUALIZATION

FOR MULTICARRIER RECEIVERS

Richard K. Martin, Ph.D.

Cornell University 2004

Multicarrier modulation has been gaining in popularity in recent years. It has been implemented in systems such as Digital Subscriber Loops, broadcast High Definition Television in Europe, wireless local area network standards such as IEEE 802.11a, power line communications, and satellite radio. In a multicarrier receiver, a time-domain equalizer (TEQ) is needed to mitigate the distortion due to the transmission channel. This typically takes the form of a filter that is designed such that the delay spread of the channel-TEQ combination has a much shorter delay spread than that of the channel alone.

This thesis has two thrusts: the primary goal is to propose, analyze, and simulate several blind, adaptive algorithms for designing the TEQ. The secondary goal is characterization and complexity reduction of both adaptive and non-adaptive TEQ designs. This will include examining symmetry of the impulse response and the locations of its zeros, as well as techniques to reuse calculations in laborious matrix computations.

BIOGRAPHICAL SKETCH

Rick Martin was born in Great Falls, Montana on May 20, 1976. He spent his youth in Ohio, Colorado, North Carolina, New Mexico, Alabama, and Maryland. His undergraduate years were spent at the University of Maryland, wherein he received degrees in Physics and Electrical Engineering (*Summa Cum Laude*, or as his eight-year-old sister said at the time, *Sigma Less Cloudy*). From 1999 to 2004 he was a graduate student at Cornell University, earning a Masters in Electrical Engineering in May 2001. He has authored eight journal papers (three of which are still under review), fifteen conference papers, three patent applications, and a solutions manual for a textbook. When he is not designing algorithms or ranting about the peer review process, he can often be found riding horses or swing dancing.

To Mom, Dad, Helen, Kirk, Andy, Elise, Erik, Gretchen,
Ladyhawke, Bulldog, Dutchman, Pooh Bear, and Doc.

ACKNOWLEDGEMENTS

Above all, I thank my family for their support. This thesis could not have been completed without the invaluable support of my advisor, Rick Johnson. My coauthors on several associated research papers, Bill Sethares, Brian Evans, Ming Ding, Geert Ysebaert, Koen Vanbleu, and Marc Moonen, contributed greatly to my understanding of the material in this thesis. My fellow students at Cornell, Jaiganesh Balakrishnan, Wonzoo Chung, Andy Klein, and John Walsh, also provided many helpful comments. This work was supported financially by NSF Grants EEC-98 72436 and CCR-0310023, by grants from Applied Signal Technology and Texas Instruments, and by the Olin Fellowship from Cornell University. Ernie Davis and the staff at Cornell Research foundation have earned many thanks for filing patent applications on several of the algorithms proposed in this thesis.

I would like to thank Kathy, Jeanelle, and Christie for teaching me how to jump; Dutchman and Molly for teaching me how to fall; John and Ronda for teaching me how to lolly kick; and Lucy, Skye, Andy, Sarah, and Cindy for teaching me how to hop.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

> "When we mean to build, we first survey the plot, then draw the model."
>
> – William Shakespeare, *Henry IV Part II*, Act I, Scene iii.

Loosely speaking, the goal of equalization for single carrier transmission systems is to design an equalizer such that the convolution of the channel and equalizer is a Kronecker delta, i.e. an impulse at some delay Δ and zero otherwise. In multicarrier transmission systems, the problem is more general. The delay spread of the transmission channel must be within a predefined length, and the equalizer is designed such that the convolution of the channel and equalizer produces an effective channel that has been shortened to this length. This design problem is referred to as "channel shortening."

Although the task of blind adaptive equalization is well understood, its generalization, blind adaptive channel shortening, has not hitherto received nearly as much attention from academics. Essentially all of the published literature on channel shortening has considered bit rate maximization for an ADSL system with a known, time-invariant channel and known, time-invariant noise statistics. However, multicarrier systems have been proposed for and deployed in increasing numbers of applications in which time variations are expected. This chapter first motivates the need for channel shortening, then motivates why blind, adaptive designs are of interest. In addition, the notation to be used throughout the thesis will be established in this chapter.

1.1 Channel shortening system model

Channel shortening first became an issue for reduced-state sequence estimation (RSSE) in the 1970's, and then reappeared in the 1990's in the context of multicarrier modulation. Channel shortening has also recently been proposed for use in multiuser detection. This section begins by reviewing these three applications as a means of motivating the channel shortening problem. Particular attention will be paid to channel shortening for multicarrier systems.

Maximum likelihood sequence estimation (MLSE) [32] is the optimal estimation method in terms of minimizing the error probability of a sequence. However, for an alphabet of size $\mathcal{A}$ and an effective channel length of $L_c + 1$, the complexity of MLSE grows as $\mathcal{A}^{L_c}$. For many practical transmission schemes, such as Enhanced Data rates for GSM Evolution (EDGE), the complexity of the MLSE is too high to be implemented [33], [114]. There are several methods available for reducing the complexity. Delayed Decision Feedback Sequence Estimation (DDFSE) [27], and its generalization, Reduced-state Sequence Estimation (RSSE) [28], [29], reduce the number of states considered in the trellis. DDFSE only considers the first K taps of the channel for use in the trellis, and hard decisions are used to cancel the remaining taps; whereas RSSE divides the constellation into subsets, effectively reducing the constellation size.

An alternate approach is to use the full MLSE algorithm, but to employ a prefilter to shorten the effective channel to a manageable length [4], [30], [48], [50], [68], [82]. One approach is to design both the prefilter and the (shortened) target impulse response to minimize the mean squared error (MSE) between the outputs of the target and the convolution of the channel and prefilter [4], [30]. Other approaches use a decision feedback equalizer (DFE) to shorten the channel, and then apply the MLSE [48], [68]. However, the use of a DFE assumes that the transmitted data has a finite alphabet, which may not be the case for other applications (e.g. multicarrier systems).

More recently, channel shortening has been proposed for use in multiuser detection [66]. Consider a DS-CDMA system with L users, with a flat fading channel for each user. The optimum multiuser detector in this case is the MLSE, yet complexity grows exponentially with the number of users. "Channel shortening" can be implemented to suppress $L-K$ of the scalar channels and retain the other K channels, effectively reducing the number of users from L to K. Then the MLSE can be implemented to recover the signals of the remaining K users [66]. In this context, "channel shortening" means reducing the number of scalar channels rather than reducing the number of channel taps, and the mathematical structure is similar to channel shortening for MLSE applications.

Channel shortening has recently seen a revival due to its use in multicarrier modulation (MCM) [11]. MCM techniques like orthogonal frequency division multiplexing (OFDM) and discrete multi-tone (DMT) have been deployed in applications such as the wireless LAN standards IEEE 802.11a [76] and HIPERLAN/2 [39], Digital Audio Broadcast (DAB) [41] and Digital Video Broadcast (DVB) [40] in Europe, asymmetric and very-high-speed digital subscriber loops (ADSL, VDSL), power line communications (PLC), and satellite radio. MCM is attractive due to the ease with which it can combat channel dispersion, provided the channel delay spread is not greater than the length of the cyclic prefix (CP). However, if the cyclic prefix is not long enough, the orthogonality of the sub-carriers is lost, causing inter-carrier interference (ICI) and inter-symbol interference (ISI). A review of MCM is provided in [1].

A well-known technique to combat the ICI/ISI caused by the inadequate CP length is the use of a time-domain equalizer (TEQ)[1] at the receiver front-end. The TEQ is a filter that shortens the channel so that the delay spread of the combined channel-equalizer response is no larger than the length of the CP.

In many cases, the receiver may need to jointly shorten multiple channels using a single TEQ. In a multicarrier code division multiple access (MC-CDMA) system, multiple users each spread their signals using a spreading code before multicarrier modulation takes place [46]. To enhance performance, the receiver can jointly shorten all of the users' channels to mitigate ISI before de-spreading takes place. In DSL, each modem receives the desired signal as well as crosstalk from other signals in the same cable bundle. In this case, joint channel shortening can be combined with multiuser detection to improve the receiver's performance. If a DSL system is operating in echo cancelling mode [95], then the channel and the echo path must be jointly shortened [67], [69]. As another example, multiple receive antennas or oversampling of the received data may be employed, leading to multiple outputs for each input. This motivates a multiple input, multiple output (MIMO) system model, in which multiple channels need to be shortened simultaneously. Joint channel shortening has been studied in [2], [49], [67], [69], [94], [96], [114]. However, these works involved extending the training-based, non-adaptive single-input, single output (SISO) TEQs proposed in [4], [30], and [67] to the MIMO case; whereas this thesis considers the blind, adaptive case.

Single-input, multiple-output (SIMO) channel shortening, which often arises due to oversampling the signal from one user, is of particular interest. Conditions (either necessary or sufficient) on the ability to achieve perfect shortening have been considered in [72], [73], [79], [80], [88].

For clarity, we first discuss the traditional single-input single-output (SISO)

[1]The TEQ has also been called a channel shortening equalizer (CSE) [2].

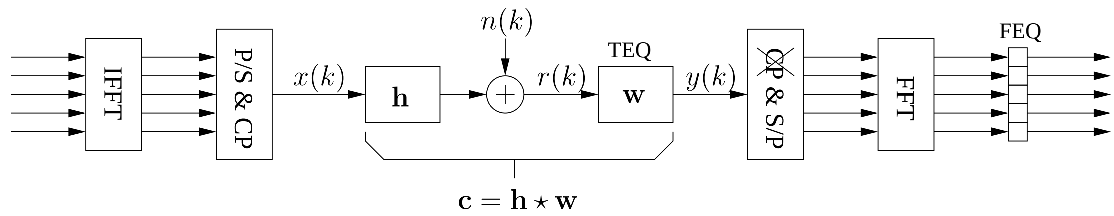

Figure 1.1: Traditional SISO multicarrier system model. (I)FFT: (inverse) fast Fourier transform, P/S: parallel to serial, S/P: serial to parallel, CP: add cyclic prefix, xCP: remove cyclic prefix.

system model, then generalize to the MIMO case. The SISO multicarrier system model is shown in Figure 1.1, and the time-domain portion of the MIMO model is shown in Figure 1.2. The notation is summarized in Tables 1.1, 1.2, 1.3, and 1.4. In the SISO case, the input stream is first divided into blocks of N QAM symbols. Each block of symbols is mapped into N bins, and each bin is viewed as a QAM signal that will be modulated by a different carrier. An efficient means of implementing the modulation in discrete time is to use an inverse fast Fourier transform (IFFT), with each bin acting as one of the frequency components that will be converted into a time signal by the IFFT. Then, after transmission, an FFT can be used to restore the data to its original format.

In order for the subcarriers to be independent, the convolution of the signal and the channel must be a circular convolution. It is actually a linear convolution, so it is made to appear circular by adding a cyclic prefix to the start of each data block, which is obtained by prepending the last ν samples of each block to the beginning of the block. If the CP is at least as long as the channel, then the output of each subchannel is equal to the input times a scalar gain factor. The signals in the bins can then be equalized by a bank of complex scalars, referred to as a frequency

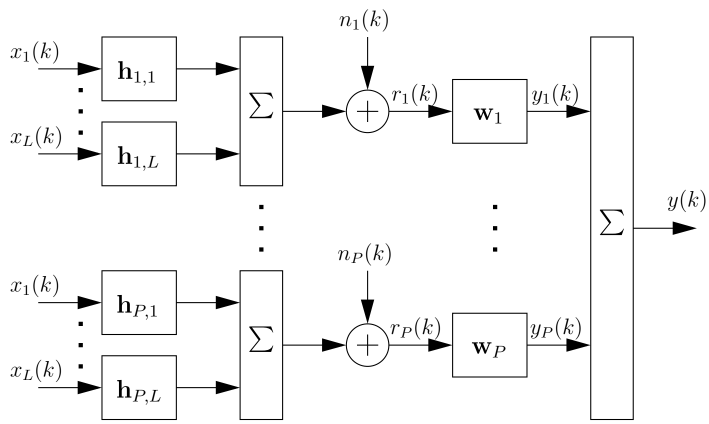

Figure 1.2: MIMO TEQ model, for L transmitters and P receive antennas (or oversampling by a factor of P). Channel $\mathbf{h}_{p,l}$ connects the l^{th} transmitter to the p^{th} receive antenna.

Table 1.1: Notation for signals used in multicarrier systems

Notation	**Meaning**
$X_{i,l}(k)$	source signal for tone i, user l (IFFT input)
$x_l(k)$	transmitted signal for user l (IFFT output)
$n_p(k)$	additive noise on p^{th} received sequence
$r_p(k)$	p^{th} sequence of received data
$y_p(k)$	output of TEQ p
$y(k)$	recombined output $= \sum_p y_p(k)$
$\hat{z}_{i,l}(k)$	signal at input of FEQ for tone i, user l
$z_{i,l}(k)$	signal at output of FEQ for tone i, user l

domain equalizer (FEQ).

The above discussion assumes that the CP length + 1 is greater than or equal to the channel length. However, transmitting the cyclic prefix wastes time slots that could be used to transmit data. Thus, the CP is usually set to a reasonably

Table 1.2: Notation for filters used in multicarrier systems

Notation	**Meaning**
$D_i(k)$	FEQ for tone i at time k
$\mathbf{h}_{p,l} = [h_{p,l,0}, \cdots, h_{p,l,L_h}]^T$	channel from user l to receiver p
$\mathbf{w}_p = [w_{p,0}, \cdots, w_{p,L_w}]^T$	p^{th} TEQ impulse response
$\mathbf{c}_{p,l} = [c_{p,l,0}, \cdots, c_{p,l,L_c}]^T$	effective channel $= \mathbf{h}_{p,l} \star \mathbf{w}_p$
$\mathbf{b}_{p,l} = [b_{p,l,0}, \cdots, b_{p,l,\nu}]^T$	target impulse response from user l to receiver p
$\mathbf{v}_{i,p} = [v_{i,p,0}, \cdots, v_{i,p,L_v}]^T$	per tone equalizer (PTEQ) for tone i, receiver p
$\overline{\mathbf{w}}_p = [w_{p,L_w}, \cdots, w_{p,0}]^T$	time-reversal (of $\mathbf{w}_p$, e.g.)

small value, and a TEQ is employed to shorten the channel to this length. The TEQ design problem has been extensively studied in the literature [5], [7], [17], [30], [31], [47], [53], [67], as will be discussed in Chapter 2.

For the MIMO case, we will consider L transmitters, indexed by $1 \leq l \leq L$; and P receivers (or oversampling by P), indexed by $1 \leq p \leq P$. The received signal $r_p(k)$ from antenna p (or the sequence of each p^{th} sample per baud interval), $p \in \{1, \cdots, P\}$, is obtained by passing each signal from user $l \in \{1, \cdots, L\}$ through channel $h_{p,l}$, adding the L channel outputs, and adding noise sequence $n_p(k)$. In an MC-CDMA system, each user's signal $x_l(k)$ is obtained by spreading one or more symbols in the frequency domain, taking an IFFT, and adding a cyclic prefix; see [46], e.g. If the multiple users arise due to cross-talk in a wireline system, then each signal $x_i(k)$ is generated in the manner of Figure 1.1.

After the CP is added, the last ν samples are identical to the first ν samples in each transmitted symbol, i.e.

$$x_l\left(Mk+i\right) = x_l\left(Mk+i+N\right), \quad i \in \{1, \ldots, \nu\}, \quad l \in \{1, \ldots, L\}, \tag{1.1}$$

Table 1.3: Notation for parameters and indices

Notation	**Meaning**
i	tone index
k	symbol index
j	sample index within a symbol, or generic index
l	user (transmitter) index
p	receiver (or polyphase) index
L	number of transmitters
P	number of receivers
N	FFT size
ν	cyclic prefix (CP) length and TIR order
$M = N + \nu$	total time-domain symbol size
N_c	number of used carriers (containing data, not pilots or zeros)
N_z	number of null carriers (zeros transmitted)
T_{CP}	time span of the cyclic prefix
Δ	delay of effective channel
$\tilde{L}_h = L_h + 1$	channel length
$\tilde{L}_w = L_w + 1$	TEQ length
$\tilde{L}_c = L_c + 1$	length of effective channel

where $M = N + \nu$ is the total symbol duration and k is the symbol index. The received data r_p is obtained from $\{x_l : l = 1, \cdots, L\}$ by

$$r_p(Mk+i) = \sum_{l=1}^{L} \sum_{j=0}^{L_h} h_{p,l}(j) \cdot x_l(Mk+i-j) + n_p(Mk+i), \tag{1.2}$$

and y_p, the output of TEQ p, is obtained from r_p by

$$y_p(Mk+i) = \sum_{j=0}^{L_w} w_p(j) \cdot r_p(Mk+i-j). \tag{1.3}$$

Table 1.4: Notation for matrices used in multicarrier systems

Notation	**Meaning**
$\mathbf{H}_{p,l}$	$\tilde{L}_c \times \tilde{L}_w$ tall channel convolution matrix
$\mathbf{H}_{p,l,win}$	middle $\nu + 1$ rows of $\mathbf{H}$
$\mathbf{H}_{p,l,wall}$	$\mathbf{H}$ with middle $\nu + 1$ rows removed
$\mathbf{R}_{n,p}$	autocorrelation matrix of $n_p(k)$
$\mathbf{I}_N$	$N \times N$ identity matrix
$\mathbf{J}_N$	$N \times N$ matrix with ones on the cross diagonal and zeros elsewhere
$\mathcal{F}_N$	FFT matrix of size $N \times N$
$\mathcal{I}_N = \mathcal{F}_N^H$	IFFT matrix of size $N \times N$
$\mathbf{f}_i^T$	i^{th} row of $\mathcal{F}_N$
$\mathbf{P}$	$M \times N$ matrix that adds the cyclic prefix
$\hat{\mathbf{P}}$	$N \times M$ matrix that removes the cyclic prefix
$\mathbf{A}^*$, $\mathbf{A}^T$, $\mathbf{A}^H$	conjugate, transpose, and Hermitian (of $\mathbf{A}$, e.g.)

Then the final, recombined output is obtained by

$$y(Mk + i) = \sum_{p=1}^{P} y_p(Mk + i). \tag{1.4}$$

One could either work with the collection of P sequences $\{y_p(k)\}$ or the single output sequence $y(k)$. The weights for the linear combination in (1.4) have implicitly been absorbed into the P TEQs. Each of the $P \cdot L$ channels is modeled as an FIR filter of length $L_h + 1$, each of the P TEQs is an FIR filter of length $L_w + 1$, and each effective channel $\mathbf{c}_{p,l} = \mathbf{h}_{p,l} \star \mathbf{w}_p$ has length $L_c + 1$, where $L_c = L_h + L_w$. The symbol $\star$ denotes linear convolution.

In a wireline system such as DSL, the transmitted signal is at baseband and must be real. This is accomplished by only using $\frac{N}{2}$ inputs, and then conjugating and mirroring the first half of the tones such that the resulting N IFFT inputs obey conjugate symmetry. The transform of such a sequence will be real [77]. Specifically, given $\frac{N}{2}$ complex symbols X_i', the N IFFT inputs X_i are obtained by

$$\begin{aligned}
X_1 &= \mathcal{R}\left\{X_1'\right\} \\
X_2 &= X_N^* = X_2' \\
X_3 &= X_{N-1}^* = X_3' \\
&\vdots \\
X_{N/2-1} &= X_{N/2+3}^* = X_{N/2-1}' \\
X_{N/2} &= X_{N/2+2}^* = X_{N/2}' \\
X_{N/2+1} &= \mathcal{I}\left\{X_1'\right\}
\end{aligned} \tag{1.5}$$

where $\mathcal{R}\left\{\cdot\right\}$ and $\mathcal{I}\left\{\cdot\right\}$ denote the taking of the real or imaginary part of a complex number, respectively. This distinction between baseband and passband channels is important, since most TEQ algorithms were designed for DSL and assume real (baseband) data and channels, yet here we consider the more general case of complex (passband) channels and complex time-domain data.

1.2 The need for blind, adaptive algorithms

> "There is no reason but I shall be blind. If I can check my erring love, I will."
>
> – William Shakespeare, *Two Gentlemen of Verona*, Act II, Scene iv.

Most channel shortening schemes in the literature have been designed in the context of ADSL, which runs over twisted pair telephone lines [5], [7], [31], [67]. As a consequence, most of the TEQ designs in the literature are trained and non-adaptive. However, other current and emerging multicarrier standards have less training and are used in environments with more rapid time-variations, so new adaptive TEQ design methods are needed. This section motivates the need for blind, adaptive algorithms.

There are several factors which can give rise to time variations. In wireline systems such as ADSL and VDSL, circuits heat up over time, changing the characteristics of the channel. In addition, the number of other users may change, affecting the noise (crosstalk) statistics. The current solution to this problem is to design the TEQ with a large performance margin, and then keep the same static TEQ despite the time-variations. The hope is that the time-variations do not change the channel enough to push the TEQ out of its performance margin. However, eventually the TEQ will need to be retrained. The current standard does not have a provision for retransmitting a training sequence after the initial handshake period, so designs that are non-adaptive or require training can do nothing but request a complete reinitialization of the DSL modem. This leads to an interruption in service while the 17000 training symbols are retransmitted and the new TEQ is computed. As stated in *The Ericsson Review* [42],

> During data transmission, [the TEQ] remains fixed. Overall performance would improve, however, if the TEQ continued its adaptation relative to time-variant loop[2] and noise conditions.

The current ADSL standard transmits one synchronization symbol per 69 symbols, which, as the name implies, is intended to be used for synchronization rather than training. However, it can be used to train the TEQ adaptively. A blind, adaptive algorithm could allow the TEQ to adapt 69 times as often as a trained, adaptive TEQ. Alternatively, one could repeatedly perform a blind channel estimation[3] and

---

[2] In the context of DSL, "loop" refers to the transmission channel.

[3] Blind channel estimation methods for OFDM are readily available [37], al-

then design the TEQ based on the channel estimate, using the methods in [4], [7], [67], e.g. However, it is computationally expensive to repeatedly perform non-adaptive TEQ designs, and the validity of the channel estimates decrease if they are performed infrequently.

The channels for wireless systems are prone to more rapid time variations than wireline systems. Examples of such systems are the European Digital Audio Broadcast (DAB) [41] and Digital Video Broadcast (DVB) [40] standards, the High Performance Radio Local Area Network (HIPERLAN/2) [39] standard, the local area network (LAN) standard IEEE 802.11a [76], and the proprietary modulation techniques employed by both companies currently providing satellite radio to North America. As multicarrier modulation becomes more popular, we can expect more wireless standards to arise which use OFDM.

We can also expect existing standards to be pushed beyond their initial design intentions. IEEE 802.11a is intended for indoor, non-mobile use. However, these systems will eventually be pushed into other uses (such as outdoor and/or mobile use), and even during the intended use the design assumptions may not be correct. The IEEE 802.11a standard specifies a CP of duration $0.8\mu s$, yet indoor channel measurement studies [24], [25], [83], [89] have shown that the worst case delay spreads can be as large as $1\mu s$ to $3\mu s$ depending on the topology of the indoor environment and the surrounding buildings. Current IEEE 802.11a receivers do not typically employ TEQs, but they would have better performance (in some situations) if they did. Future wireless LAN standards may be designed for outdoor use, which would lead to even longer delay spreads and faster time variations.

Broadcast systems, such as DAB and DVB, do not have the long initial training

---

though in some cases they assume the channel is already short [74], [108], and thus will not help in designing a TEQ.

sequence used in ADSL, but rather have sporadic training sequences that only occur in some frequency bins and/or some symbols. This can lead to severe problems for trained TEQ algorithms. For example, the adaptive MMSE TEQ of [30] requires that the time-domain signal be known in order to adapt the TEQ. However, in DVB, the training is implemented during every symbol, but only on some frequency bins. That is, a known subset of carriers contain known data, but the time-domain signal is the IFFT of all the frequency data (both known and unknown) for that symbol, so no samples of the time-domain symbol are known. This means that the receiver cannot use the trained algorithm of [30], or in fact any other TEQ that uses time-domain training. Furthermore, even in a broadcast system that transmits known time-domain symbols on occasion, a blind TEQ will be able to adapt much more often, and hence will have superior tracking capabilities.

Another advantage of blind equalizers is that they sometimes facilitate the design of the rest of the receiver. The standard example for a single carrier system is the Constant Modulus Algorithm (CMA) [34], which allows the equalizer to be designed before carrier frequency offset (CFO) estimation is performed [101]. Then the CFO estimation can be performed with greater ease and accuracy on the equalized signal. This thesis discusses a TEQ design algorithm with similar properties, called SAM [8]. The SAM cost function is also invariant to carrier frequency offsets, so SAM can also be applied before CFO estimation is performed. Moreover, like CMA, SAM allows the equalizer to be designed without specifiying the equalization delay Δ. In contrast, existing multicarrier equalization techniques require a costly global search over the delay parameter [7], [30], [67], [70]. Such approaches require a complete TEQ design for each possible value of Δ, increasing the complexity by one to two orders of magnitude.

1.3 Examples of multicarrier systems

Table 1.5 shows typical parameter values for several multicarrier applications. Applications for which other FFT sizes or CP lengths are allowed are marked with an asterisk. The systems considered are the wireless LAN standards IEEE 802.11a, IEEE 802.11g, and HIPERLAN/2; the standard for upstream and downstream ADSL; the terrestrial broadcast standards for digital audio and video broadcast; power line communications; and a representative model for satellite radio. The notation is as in Table 1.3, and "$\mathcal{R}$ or $\mathcal{C}$?" refers to whether or not the transmitted time domain signal is real or complex.

Table 1.5: Parameters of various multicarrier systems. The "*" denotes systems for which multiple FFT sizes and/or CP lengths are allowed. "TBD" (to be determined) means that the parameters have not been standardized yet.

System	N	ν	M	N_c	N_z	T_{CP}	$\mathcal{R}$ or $\mathcal{C}$?
IEEE 802.11a [76]	64	16	80	48	12	0.8 μs	$\mathcal{C}$
IEEE 802.11g	64	16	80	48	12	0.8 μs	$\mathcal{C}$
HIPERLAN/2* [39]	64	16	80	48	12	0.8 μs	$\mathcal{C}$
Upstream ADSL [95]	64	4	68	28	4	14.5 μs	$\mathcal{R}$
Downstream ADSL [95]	512	32	544	224	32	14.5 μs	$\mathcal{R}$
DAB mode I [41]	2048	504	2552	1536	512	250 μs	$\mathcal{C}$
DAB mode IV [41]	1024	252	1276	768	256	500 μs	$\mathcal{C}$
DAB mode II [41]	512	126	638	384	128	1 ms	$\mathcal{C}$
DAB mode III [41]	256	63	319	192	64	2 ms	$\mathcal{C}$
DVB, 2K mode* [40]	2048	64	2112	1512	343	0.7 μs	$\mathcal{C}$
DVB, 8K mode* [40]	8192	256	8448	6048	1375	0.28 μs	$\mathcal{C}$
PLC	TBD	TBD	TBD	TBD	TBD	TBD	$\mathcal{R}$
Satellite Radio	2048	200	2248	1000	1048	25 μs	$\mathcal{C}$

Chapter 2

Design Methods: Literature Survey and Unification

"Come, and take choice of all my library, and so beguile thy sorrow."

– William Shakespeare, *Titus Andronicus*, Act IV, Scene i.

This chapter[1] will discuss the multitude of multicarrier equalizer designs in the literature. The focus will be on SISO non-adaptive designs related to the adaptive designs discussed in this thesis, as well as on alternate adaptive algorithms. Section 2.1 sets up a common design formulation that encapsulates the vast majority of TEQ designs in the literature. Sections 2.2 and 2.3 deal with special cases of this formulation, and Section 2.4 deals with TEQ designs that do not fall into this formulation. Section 2.5 discusses alternate equalizer structures, as opposed to the TEQ structure of the first four sections. Much of the work in this chapter was performed jointly with researchers at the University of Texas at Austin and Catholic University, Leuven [62], [63]. Figure 2.1 categorizes some of the more popular equalizer structures and designs.

2.1 A common TEQ design formulation

There are many ways of designing the DMT equalizer, depending on how the optimization problem is posed. However, almost all of the algorithms fit into the same formulation: the maximization of a generalized Rayleigh quotient or a

[1]Some material in this chapter is contained in [62]. © 2004 IEEE. Reprinted, with permission, from [62].

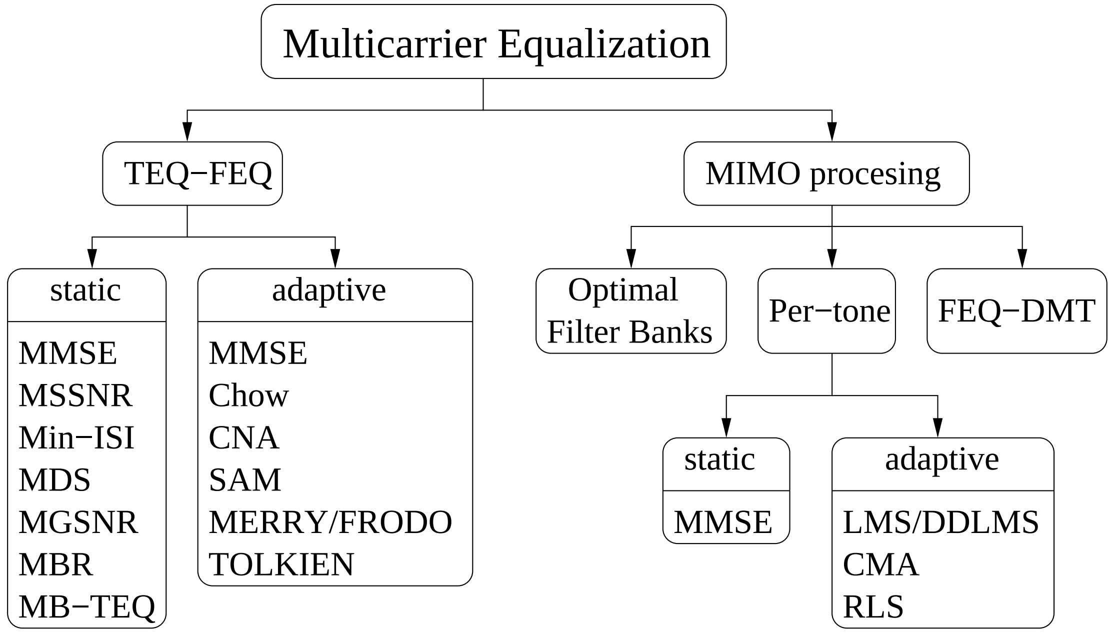

Figure 2.1: Approaches to multicarrier equalization.

product of generalized Rayleigh quotients [62], [63]. Consider the optimization problem

$$\boxed{\hat{\mathbf{w}}^{opt} = \arg\max_{\hat{\mathbf{w}}} \prod_{j=1}^{M} \frac{\hat{\mathbf{w}}^H \mathbf{B}_j \hat{\mathbf{w}}}{\hat{\mathbf{w}}^H \mathbf{A}_j \hat{\mathbf{w}}}} \tag{2.1}$$

In general, the solution to (2.1) is not well-understood when $M > 1$. However, for $M = 1$,

$$\boxed{\hat{\mathbf{w}}^{opt} = \arg\max_{\hat{\mathbf{w}}} \frac{\hat{\mathbf{w}}^H \mathbf{B} \hat{\mathbf{w}}}{\hat{\mathbf{w}}^H \mathbf{A} \hat{\mathbf{w}}}}, \tag{2.2}$$

the solution is the generalized eigenvector of the matrix pair $(\mathbf{B}, \mathbf{A})$ corresponding to the largest generalized eigenvalue [109]. Equivalently, the inverse of the ratio in (2.2) is minimized by the eigenvector of $(\mathbf{A}, \mathbf{B})$ corresponding to the smallest generalized eigenvalue. Most TEQ designs fall into the category of (2.2), although several have $M \gg 1$ as in (2.1). The vector $\hat{\mathbf{w}}$ to be optimized is usually the TEQ, but it may also be e.g. the (shortened) target impulse response (TIR) [30], the per-tone equalizer [104], or half of a symmetric TEQ [56].

TEQ designs of the form of (2.2) include the Minimum Mean Squared Error (MMSE) design [30], [4], the Maximum Shortening SNR (MSSNR) design [67], the MSSNR design with a unit norm TEQ constraint (MSSNR-UNT) or a symmetric TEQ constraint (Sym-MSSNR) [60], the Minimum Inter-symbol Interference (Min-ISI) design [7], the Minimum Delay Spread (MDS) design [92], and the Carrier Nulling Algorithm (CNA) [23]. When a separate filter is designed for each tone, as in the per-tone equalizer (PTEQ) [104] or the TEQ filter bank (TEQFB) [70], [71], each filter can be designed by solving (2.2) separately for each tone.

The generalized eigenvector problem requires computation of the $\hat{\mathbf{w}}$ that satisfies [35], [109]

$$\mathbf{B}\ \hat{\mathbf{w}} = \lambda\ \mathbf{A}\ \hat{\mathbf{w}}, \tag{2.3}$$

where $\hat{\mathbf{w}}$ corresponds to the largest generalized eigenvalue λ. If $\mathbf{A}$ is invertible, the problem can be reduced to finding an eigenvector of $\mathbf{A}^{-1}\mathbf{B}$ [109]. When $\mathbf{A}$ is real and symmetric, another approach is to form the Cholesky decomposition $\mathbf{A} = \sqrt{\mathbf{A}}\sqrt{\mathbf{A}}^T$, and define $\hat{\mathbf{v}} = \sqrt{\mathbf{A}}^T\hat{\mathbf{w}}$, as in [67]. Then

$$\hat{\mathbf{v}}^{opt} = \arg\max_{\hat{\mathbf{v}}} \frac{\hat{\mathbf{v}}^T \overbrace{\left(\sqrt{\mathbf{A}}^{-1}\ \mathbf{B}\ \sqrt{\mathbf{A}}^{-T}\right)}^{\mathbf{C}} \hat{\mathbf{v}}}{\hat{\mathbf{v}}^T\hat{\mathbf{v}}}. \tag{2.4}$$

The solution for $\hat{\mathbf{v}}$ is the eigenvector of $\mathbf{C}$ associated with the largest eigenvalue, and $\hat{\mathbf{w}} = \sqrt{\mathbf{A}}^{-T}\hat{\mathbf{v}}$, assuming that $\mathbf{A}$ is invertible. If $\mathbf{A}$ is not invertible, then it has a non-zero null space, so the ratio is maximized (to infinity!) by choosing $\hat{\mathbf{w}}$ to be a vector in the null space of $\mathbf{A}$.

In some cases, $\mathbf{A}$ or $\mathbf{B}$ is the identity matrix, in which case (2.3) reduces to a traditional eigenvalue problem. Examples include the computation of the MMSE target impulse response (TIR) [30], the MSSNR TEQ with a unit norm constraint on the TEQ [60], the MDS algorithm [92], and CNA [23]. There is a variety of all-purpose methods available for finding extreme eigenvectors, such as the power method [35] and conjugate gradient methods [112]. More specific iterative eigensolvers may be designed for specific problems, such as the MERRY (Multicarrier Equalization by Restoration of RedundancY) algorithm [53] and Nafie and Gatherer's method [75], which adaptively/iteratively compute the MSSNR TEQ.

If neither $\mathbf{A}$ or $\mathbf{B}$ is the identity matrix, efficient iterative techniques can still be used. The generalized power method [35] can be used to solve (2.2) by iterating

$$\begin{aligned} \mathbf{B}\ \hat{\mathbf{w}}_{k+1} &= \mathbf{A}\ \mathbf{w}_k \\ \mathbf{w}_{k+1} &= \frac{\hat{\mathbf{w}}_{k+1}}{\|\hat{\mathbf{w}}_{k+1}\|}, \end{aligned} \tag{2.5}$$

requiring a square root and division at each step.

The expensive renormalization in (2.5) can be avoided through the use of a Lagrangian constraint, as in [15], [16], leading to an iterative eigensolver for (2.2) of the form

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \left(\mathbf{B}\mathbf{w} - \mathbf{A}\mathbf{w} \left(\mathbf{w}^H \mathbf{B}\mathbf{w} \right) \right), \tag{2.6}$$

where μ is a small user-defined step size. If stochastic rank-one approximations of $\mathbf{B}$ and $\mathbf{A}$ are available, as in [53], then the generalized eigensolver in (2.6) requires $\mathcal{O}(L_w)$ multiply-adds per update. If the matrices $\mathbf{A}$ and $\mathbf{B}$ are used explicitly, (2.6) requires $\mathcal{O}(L_w^2)$ multiply-adds per update. For comparison, each Cholesky decomposition requires $\mathcal{O}(L_w^3)$ floating point operations, including many divisions.

The much more difficult case when $M > 1$ in (2.1) is not well-understood. There may be many solutions that are locally optimal but not necessarily globally optimal, so gradient-descent strategies only ensure convergence to a local optimum. Some TEQ algorithms of this form are the maximum geometric SNR (MGSNR) [5], Maximum Bit Rate (MBR) [7], Maximum Data Rate (MDR) [70], and Bitrate

Maximum TEQ (BM-TEQ) [105], [106] methods. One approach is to compute several reasonable initial guesses, apply gradient descent (or a Newton-like algorithm) to each initialization, and then pick the best solution. This is not guaranteed to be optimal. The initial guesses can be made by computing the closed-form solutions for various $M = 1$ cases, such as the MSSNR TEQ or TEQs that optimize the bit rate on individual tones [70].

The next three sections review the literature for the cases of a single Rayleigh quotient, multiple Rayleigh quotients, and exceptions to the rule.

2.2 Single Rayleigh quotient cases

Several common TEQ designs that are designed by maximizing a generalized Raleyeigh quotient are the minimum mean squared error (MMSE) design [30], [4], the maximum shortening SNR (MSSNR) design [67], [113], the minimum inter-symbol interference (Min-ISI) design [7], the minimum delay spread (MDS) design [92], the minimum inter-block interference (Min-IBI) design [15] and their variants. This section summarizes these designs.

The design that deals with the concept of "channel shortening" most directly is the MSSNR design [67], since it operates on the channel with the goal of shortening it without regard to any other performance metrics. As depicted in Figure 2.2, the MSSNR TEQ attempts to maximize the ratio of the energy in a window of the effective channel over the energy in the remainder of the effective channel. The MSSNR design was reformulated for numerical stability in [113], and iterative and adaptive implementations have been proposed in [75] and [53]. Following [67], we

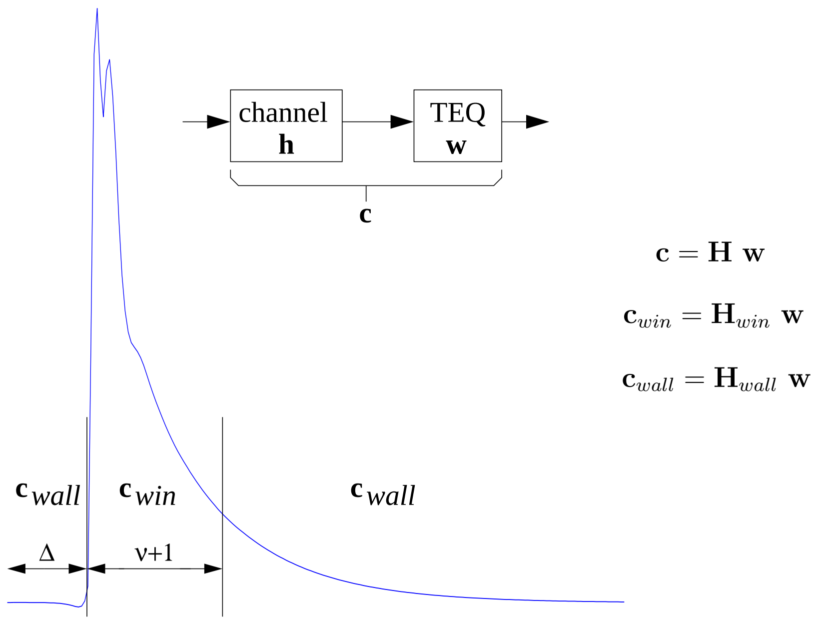

Figure 2.2: Maximum Shortening SNR channel shortening.

define

$$\mathbf{H}_{win} = \begin{bmatrix} h(\Delta) & h(\Delta-1) & \cdots & h(\Delta-\tilde{L}_w+1) \\ \vdots & & \ddots & \vdots \\ h(\Delta+\nu) & h(\Delta+\nu-1) & \cdots & h(\Delta+\nu-\tilde{L}_w+1) \end{bmatrix} \tag{2.7}$$

as the middle $\nu+1$ rows of the (tall) channel convolution matrix $\mathbf{H}$, and $\mathbf{H}_{wall}$ as the remaining rows of $\mathbf{H}$:

$$\mathbf{H}_{wall} = \begin{bmatrix} h(0) & 0 & \cdots & 0 \\ \vdots & \ddots & & \\ h(\Delta-1) & h(\Delta-2) & \cdots & h(\Delta-\tilde{L}_w) \\ h(\Delta+\nu+1) & h(\Delta+\nu) & \cdots & h(\Delta+\nu-\tilde{L}_w+2) \\ \vdots & \ddots & & \\ 0 & \cdots & 0 & h(L_h) \end{bmatrix} \tag{2.8}$$

Thus, $\mathbf{c}_{win} = \mathbf{H}_{win}\mathbf{w}$ yields a length $\nu+1$ window of the effective channel, and $\mathbf{c}_{wall} = \mathbf{H}_{wall}\mathbf{w}$ yields the remainder of the effective channel. The MSSNR design problem can be stated as "maximize $\|\mathbf{c}_{win}\|$ subject to the constraint $\|\mathbf{c}_{wall}\| = 1$," [67], [113] which reduces to (2.2) with

$$\begin{aligned} \mathbf{A} &= \mathbf{H}_{wall}^H \mathbf{H}_{wall} \\ \mathbf{B} &= \mathbf{H}_{win}^H \mathbf{H}_{win}. \end{aligned} \tag{2.9}$$

Since $\mathbf{H}^H\mathbf{H} = \mathbf{H}_{win}^H\mathbf{H}_{win} + \mathbf{H}_{wall}^H\mathbf{H}_{wall}$, it is mathematically equivalent to minimize the wall energy over the total channel energy [98], with

$$\begin{aligned} \mathbf{A} &= \mathbf{H}_{wall}^H \mathbf{H}_{wall} \\ \mathbf{B} &= \mathbf{H}^H \mathbf{H}; \end{aligned} \tag{2.10}$$

or to maximize the energy inside the window over the total channel energy [38], with

$$\begin{aligned} \mathbf{A} &= \mathbf{H}^H \mathbf{H} \\ \mathbf{B} &= \mathbf{H}_{win}^H \mathbf{H}_{win}, \end{aligned} \tag{2.11}$$

which results in a reduced-complexity implementation.

The MMSE design [30], originally intended for complexity reduction in maximum likelihood sequence estimation (MLSE), is similar to the MSSNR design, although it takes noise into account. Indeed, for a white input and no noise, the MMSE and MSSNR designs are identical [20]. The system model for the MMSE solution is shown in Figure 2.3. It creates a virtual target impulse response (TIR) $\mathbf{b}$ of length $\nu + 1$ such that the MSE, which is measured between the output of the effective channel and the output of the TIR, is minimized.

The MMSE TEQ and TIR must satisfy [4], [30]

$$\mathbf{R}_{rx}\mathbf{b} = \mathbf{R}_r\mathbf{w}, \tag{2.12}$$

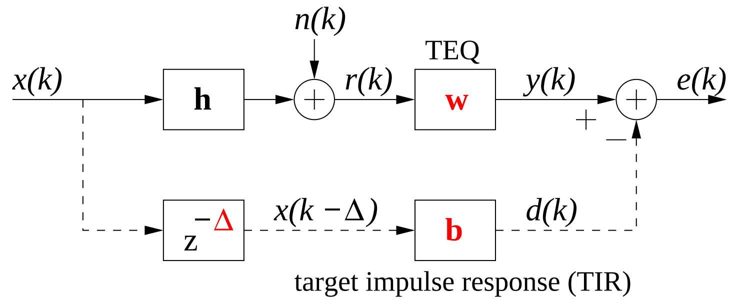

Figure 2.3: MMSE system model: $\mathbf{h}$, $\mathbf{w}$, and $\mathbf{b}$ are the impulse responses of the channel, TEQ, and target, respectively. Here, Δ represents transmission delay. The dashed lines indicate a virtual path, which is used only for analysis.

where

$$\mathbf{R}_{rx}(\Delta) = \mathrm{E}\left[\begin{bmatrix} r^*(k) \\ \vdots \\ r^*(k-L_w) \end{bmatrix} \begin{bmatrix} x(k-\Delta) & \cdots & x(k-\Delta-\nu) \end{bmatrix}\right] \tag{2.13}$$

is the channel input-output cross-correlation matrix and

$$\mathbf{R}_r = \mathrm{E}\left[\begin{bmatrix} r^*(k) \\ \vdots \\ r^*(k-L_w) \end{bmatrix} \begin{bmatrix} r(k) & \cdots & r(k-L_w) \end{bmatrix}\right] \tag{2.14}$$

is the channel output autocorrelation matrix. Typically, $\mathbf{b}$ is computed first, and then (2.12) is used to determine $\mathbf{w}$. The goal is that $\mathbf{h} \star \mathbf{w}$ approximates a delayed version of $\mathbf{b}$. The target impulse response is the eigenvector corresponding to the minimum eigenvalue of [5], [30], [31]

$$\mathbf{R}\left(\Delta\right) = \mathbf{R}_x - \mathbf{R}_{xr}\mathbf{R}_r^{-1}\mathbf{R}_{rx}. \tag{2.15}$$

where $\mathbf{R}_{xr} = \mathbf{R}_{rx}^H$ and

$$\mathbf{R}_x = \mathrm{E}\left[\begin{bmatrix} x^*(k) \\ \vdots \\ x^*(k-\nu) \end{bmatrix} \begin{bmatrix} x(k) & \cdots & x(k-\nu) \end{bmatrix}\right] \tag{2.16}$$

is the input autocorrelation matrix. Thus, the TIR can be designed using (2.2) with $\hat{\mathbf{w}} = \mathbf{b}$ as the vector to be solved for and

$$\begin{aligned} \mathbf{A} &= \mathbf{R}\left(\Delta\right) \\ \mathbf{B} &= \mathbf{I}_{\nu+1}. \end{aligned} \tag{2.17}$$

It is also possible to solve for the TEQ directly, without first computing the TIR. For a white input signal, the MMSE TEQ can be designed directly using (2.2) with [56]

$$\begin{aligned} \mathbf{A} &= \mathbf{H}_{wall}^H \mathbf{H}_{wall} + \mathbf{R}_n \\ \mathbf{B} &= \mathbf{H}_{win}^H \mathbf{H}_{win}, \end{aligned} \tag{2.18}$$

where $\mathbf{R}_n$ is the $\tilde{L}_w \times \tilde{L}_w$ noise autocorrelation matrix.

The MSSNR design has spawned many extensions and variations. The minimum inter-symbol interference (Min-ISI) method shapes the residual energy in the tails of the channel in the frequency domain, with the goal of placing the excess energy in unused frequency bins [7]. The $\mathbf{A}$ and $\mathbf{B}$ matrices are more complicated in this case; see [7] or [62], [63] for details. Efficient implementation of the Min-ISI algorithm was considered in [111].

The minimum inter-block interference (Min-IBI) method [14], [15] models the inter-block interference power as increasing linearly with the distance of the channel taps from the boundaries of the length $\nu + 1$ desired non-zero window. Thus, the Min-IBI penalty function is similar to the MSSNR penalty function, except with a linear weighting matrix, leading to a design of the form of (2.2) with

$$\begin{aligned} \mathbf{A} &= \mathbf{H}_{wall}^T \mathbf{Q}_{ibi} \mathbf{H}_{wall} \\ \mathbf{B} &= \mathbf{H}_{win}^T \mathbf{H}_{win}, \end{aligned} \tag{2.19}$$

where $\mathbf{Q}_{ibi}$ is a diagonal matrix of the form

$$\mathbf{Q}_{ibi} = \text{diag}\left[\Delta,\ \Delta - 1,\ \cdots,\ 1,\ \mathbf{0}_{1\times(\nu+1)},\ 1,\ 2, \cdots,\ L_c - \nu - \Delta\right]. \tag{2.20}$$

The minimum delay spread (MDS) algorithm [92] attempts to minimize the RMS delay spread of the effective channel, with a unit norm constraint on the effective channel. The design again has the form of (2.2) with

$$\begin{aligned} \mathbf{A} &= \mathbf{H}_{wall}^T \mathbf{Q}_{mds} \mathbf{H}_{wall} \\ \mathbf{B} &= \mathbf{H}^T \mathbf{H}, \end{aligned} \tag{2.21}$$

where $\mathbf{Q}_{mds}$ is a diagonal matrix of the form

$$\mathbf{Q}_{mds} = \text{diag}\left[\eta^2,\ (\eta - 1)^2,\ \cdots,\ 1,\ 0,\ 1,\ \cdots,\ (L_c - \eta)^2\right], \tag{2.22}$$

and where η is a design parameter indicating the desired “center of mass,” or centroid, of the channel impulse response.

A generalization of the MSSNR cost function was proposed by Tkacenko and Vaidyanathan [97], [98] which involves a more flexible “window” of the channel impulse response as well as an attention to noise gain. The MSSNR, Min-IBI, and MDS designs are special cases. Additionally, it was proposed (for the sake of completeness) to consider linear weighting of the delay spread instead of quadratic weighting,

$$\mathbf{Q}_{mds,linear} = \text{diag}\left[\eta,\ \eta - 1,\ \cdots,\ 1,\ 0,\ 1,\ \cdots,\ L_c - \eta\right], \tag{2.23}$$

leading to yet another possible TEQ design.

Arslan, Evans, and Kiaei [7] proposed another generalization of the MSSNR design, called the minimum intersymbol interference (Min-ISI) design. The idea is to penalize the residual interference according to which subchannel it is in. Subchannels with higher signal power have higher weights, hence the residual interference is penalized more. The design is a minimization of a generalized Rayleigh quotient with [7], [26]

$$\begin{aligned} \mathbf{A} &= \widehat{\mathbf{H}}_{wall}^T \sum_{i \in S_u} \left(\mathbf{f}_i^* \frac{S_{x,i}}{S_{n,i}} \mathbf{f}^T \right) \widehat{\mathbf{H}}_{wall} \\ \mathbf{B} &= \mathbf{H}^T \mathbf{H}, \end{aligned} \tag{2.24}$$

where $\widehat{\mathbf{H}}_{wall}$ is equal to $\mathbf{H}_{wall}$ padded with zeros in the middle, $S_{x,i}$ is the transmitted signal power in tone i, $S_{n,i}$ is the received noise power in tone i, S_u is the set of used tones, and $\mathbf{f}_i^T$ produces the i^{th} FFT coefficient of a vector. If all the tones were used and if $S_{x,i}/S_{n,i}$ were constant across all the tones, then the Min-ISI matrices in (2.24) would reduce to the MSSNR matrices in (2.10). Thus, the MSSNR design is a special case of the Min-ISI design. The Min-ISI design is also related to a variant of the MMSE design [21].

De Courville, *et al.* [23], and later Romano and Barbarossa [88], proposed an adaptive TEQ which relies on the presence of "null-carriers" in an MCM system. Often the edge carriers transmit zeros in order to reduce interference, and the TEQ can be designed to minimize the output energy on these channels. However, this leads to a TEQ which shortens the effective channel to a single spike, rather than to a window [23]. This "Carrier Nulling Algorithm" (CNA) design problem has the form of (2.2) with

$$\begin{aligned} \mathbf{A} &= \mathbf{P}_{cna} + \mathbf{Q}_{cna} \\ \mathbf{B} &= \mathbf{I}_{Lw+1}, \end{aligned} \tag{2.25}$$

where $\mathbf{P}_{cna}$ and $\mathbf{Q}_{cna}$ are signal- and noise-dependent matrices [23].

2.3 Multiple Rayleigh quotients

In a point-to-point system such as DSL, the true performance metric to optimize is the maximum bit allocation that does not cause the error probability to exceed a threshold (usually 10^{-7}). In broadcast systems, the true performance metric is the bit error rate (BER) for a fixed bit allocation, although a TEQ has not yet been designed to explicitly minimize BER. Optimizing the MSE or SSNR does not necessarily optimize the bit rate or bit error rate.

Recent work [5], [7], [31], [70], [105] has addressed bit rate maximization. Al-Dhahir and Cioffi were the first to attempt bit rate maximization, though the use of several approximations (generally considered dubious [7]), led to an algorithm which does not truly maximize the bit rate. Their algorithm is called the maximum geometric SNR (MGSNR) algorithm. Lashkarian and Kiaei proposed an efficient iterative implementation of the MGSNR algorithm in [47]. Arslan, *et al.* proposed the maximum bit rate (MBR) design which is very nearly optimal in terms of bit rate. However, the use of several small approximations make the MBR algorithm not quite optimal [70], [105]. These approximations were addressed in [70] and [105], leading to an accurate model of the bit rate cost function. However, the bit rate cost function is not unimodal, so the global minimum cannot always be found by standard gradient descent techniques.

All of the algorithms mentioned in this section can be cast into the framework of (2.1), with $\mathbf{A}_j$ and $\mathbf{B}_j$ matrices as given in [62], [63]. However, the adaptive algorithms considered in this thesis generally do not attempt to maximize the bit rate, but rather attempt to optimize a simpler proxy cost function in the manner of the algorithms in Section 2.2. Hence, the details of algorithms that attempt to maximize the bit rate are ommitted here. The details can be found in [62], [63].

2.4 Exceptions to the common formulation

A TEQ design that does not have an explicit cost function was proposed by Chow, Cioffi, and Bingham [17] in 1991, when ADSL was becoming popular. Chow's algorithm involves iteratively adapting and truncating the TEQ and the target impulse response until a short effective channel is achieved. Despite a total absense of theory on the convergence behavior of Chow's algorithm, it is popular due to its ease of use and generally acceptable (if not stellar) performance. However, it makes explicit use of the initial training sequence in ADSL systems. This sequence consists of a single time-domain symbol without a prefix, transmitted over and over. Thus, Chow's algorithm cannot be used in other multicarrier applications.

2.5 Other equalizer structures

Instead of having a single filter, a multicarrier system can be equalized by a bank of filters. Van Acker, *et al.* proposed the use of a bank of filters after the FFT, called a per tone equalizer (PTEQ) [104]; and Milosevic, *et al.* proposed the use of a time-domain equalizer filter bank (TEQ-FB) [69]. The two structures are mathematically equivalent, and both are generalizations of the TEQ/FEQ structure. The PTEQ or TEQ-FB filters are separately optimized for each tone, leading to better overall performance at the cost of increased complexity. In terms of the common formulation of Section 2.1, each of the filters can be designed by maximizing a generalized Rayleigh quotient.

Trautmann and Fliege proposed another post-FFT equalizer which they call FEQ-DMT [99]. They show that if the transmitted signal contains N_z null tones and if the channel order is at most N_z, then even if a cyclic prefix is not present, the signal can be recovered perfectly. This is accomplished by forming a liner combination for each tone i whose inputs are the i^{th} FFT output and the N_z FFT outputs corresponding to the null input tones. Memory and complexity requirements during data transmission are similar to the per tone case.

Yet another approach involves linear precoding [90], [91], in which the transmitted signal is precoded by a not neccessarily square matrix, which can be more general than precoding by an FFT and CP. The precoding matrix and the decoding matrix used at the receiver can be optimized based on the transmission channel. This brings up a distinction between point-to-point (generally wireline) multicarrier systems and broadcast (generally wireless) multicarrier systems. In point-to-point systems, the receiver can enlist the help of the transmitter in equalizing the channel. In such a situation, the receiver estimates the channel, then feeds the estimate back to the transmitter. Then the transmitter and receiver can implement optimal transmit and receive filter banks to recover the data (hence the term "linear precoding"). Since the FFT structure is only one specific type of precoding, it is not necessarily optimal, and the optimal precoder may improve performance. However, the optimal precoder usually destroys the FFT structure, and the computational complexity increases significantly for the transmitter and/or the receiver.

Chapter 3

Cyclic Prefix-based Adaptive Equalizers

> "This is the hour of the Shire-folk, when they arise from their quiet fields to shake the towers and the counsels of the Great."
>
> – Elrond, in *The Fellowship of the Ring*, by J. R. R. Tolkien.

When training is unavailable, the next best thing is the presence of known properties in the transmitted signal. An equalizer (or channel shortener) can then be designed to restore the presence of such properties in the equalized data. The "property restoral" philosophy of adaptive algorithm design is discussed in [100, Chapter 6]. For example, decision-directed algorithms make use of the finite-alphabet nature of most transmitted signals in digital communication systems, and the constant modulus algorithm (CMA) makes use of the fact that often the collection of possible transmitted symbols all have the same modulus.

The question, then, is "what properties are present in the transmitted data in a multicarrier communication system?" There are in fact numerous properties[1]:

1. A cyclic prefix is present, so each symbol has redundancy in the data.
2. The channel is desired to be shorter than the cyclic prefix length. If it is, then the auto-correlation of the output data should be short as well.
3. Often zeros are transmitted on the band edges, which is somewhat like frequency-domain training.
4. The frequency domain data is generally chosen from QAM constellations.

[1]Some material in this chapter has been published or submitted for publication.

Algorithms making use of Property 1 are the focus of this chapter. Properties 2 and 4 are discussed in Chapters 4 and 5, respectively. Property 3 was discussed in [23], [88], leading to the CNA equalization algorithm.

Throughout this chapter, we assume that:

A1. The L input sequences of the IFFTs are each zero-mean, white, and wide sense stationary (implying that the output bins of each IFFT are uncorrelated), with variance $\sigma_{x,l}^2$.

A2. $L_c + 1 \leq N$ (the length of the effective channel is no larger than the FFT size).

A3. For each p, the noise autocorrelation function $\mathbf{R}_{n,p}(\delta) = 0$ for $\delta \geq N - L_w$.

A4. All data sequences x_l and noise sequences n_p are pairwise uncorrelated.

We may alter assumption A4 in some cases to set $x_1 = x_2 = \cdots = x_L$, i.e. for the case of multiple transmit antennas for a single user. If there is one user with L antennas, assumption 4 may still hold if space-time coding is applied.

3.1 MERRY: a low cost, blind, adaptive channel shortener

This section derives the basic MERRY algorithm. The following sections will discuss various generalizations and performance-enhancing extensions. For the moment, consider a SISO multicarrier system. Once the cyclic prefix (CP) has been added, the transmitted data obeys the relation

$$x\left(Mk+i\right) = x\left(Mk+i+N\right), \quad i \in \{1, 2, \ldots, \nu\}, \tag{3.1}$$

where k is the symbol (block) index. The received data $\mathbf{r}$ is obtained from $\mathbf{x}$ by

$$r(Mk+i) = \sum_{l=0}^{L_h} h(l) \cdot x(Mk+i-l) + n(Mk+i), \tag{3.2}$$

and the equalized data $\mathbf{y}$ is similarly obtained from $\mathbf{r}$ by

$$y(Mk+i) = \sum_{j=0}^{L_w} w(j) \cdot r(Mk+i-j), \tag{3.3}$$

where the notation is as in Tables 1.1, 1.2, 1.3, and 1.4, with $L = P = 1$.

The channel destroys the relationship in (3.1), because the ICI & ISI that affect the CP are different from the ICI & ISI that affect the last ν samples in the symbol. Consider the motivating example in the top half of Figure 3.1. The transmitted samples 2 and 10 are identical. However, at the output of the TEQ in the receiver, the interfering samples before sample 2 are not all equal to their counterparts before sample 10. Observe that if $c(2)$, $c(3)$, and $c(4)$ were zero, then we would have $y(2) = y(10)$. Thus, if we try to force $y(2) = y(10)$ in the mean squared error sense, we may force the channel to be as short as the CP. This can be viewed as a form of property restoral [100, Chapter 6].

The astute reader will note that the preceding example shortens the channel to a particular window of ν taps: the first ν taps in the effective channel. The location of the window, and thus the transmission delay, can be adjusted by forming a different comparison. For example, as shown in the bottom half of Figure 3.1, if we force $y(3) = y(11)$ rather than $y(2) = y(10)$, then the non-zero window of the effective channel becomes $[c_1, c_2]$ rather than $[c_0, c_1]$.

The astute reader will also note that we have shortened the channel to ν taps, yet a multicarrier system only requires shortening to $\nu + 1$ taps. However, when ν is large (e.g. 32 in ADSL), shortening the channel by an extra tap should have a minimal effect on the performance.

In general, if the effective channel has been shortened, then the last sample in the Δ-delayed CP should match the last sample in the symbol. One cost function

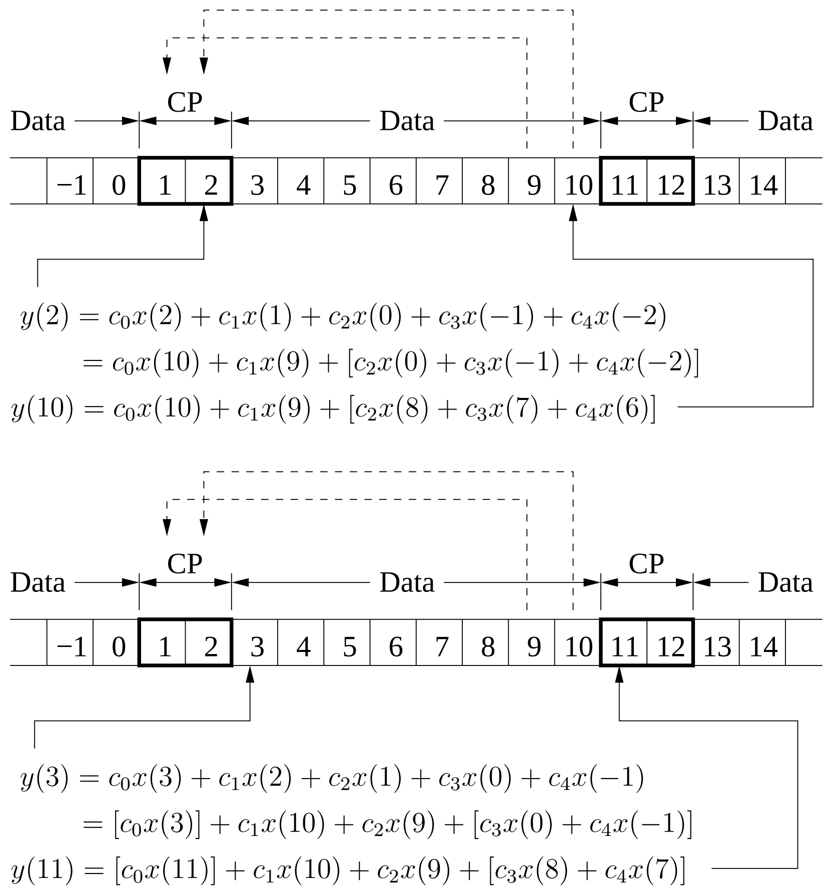

Figure 3.1: Illustration of the difference in the ISI at the received CP and at the end of the received symbol. Top: delay of $\Delta = 0$. Bottom: delay of $\Delta = 1$. $x(i)$, c_i, and $y(i)$ are the transmitted data, effective channel, and TEQ output, respectively, and the bracketed terms are intended to be suppressed.

that reflects this is

$$J_{merry}(\Delta) = \mathrm{E}\left[\left|y(Mk+\nu+\Delta) - y(Mk+\nu+N+\Delta)\right|^2\right],$$
$$\Delta \in \{0,\ \ldots,\ M-1\}, \tag{3.4}$$

where Δ is the symbol synchronization parameter, which represents the desired delay of the effective channel. The choice of Δ affects the cost function. Eq. (3.4) can apply to MIMO models as well as SISO models, if the output sequence $\{y\}$ is generated as in (1.4).

A stochastic gradient descent of (3.4) leads to a blind, adaptive TEQ, since the transmitted data need not be known. The algorithm, "Multicarrier Equalization by Restoration of RedundancY" (MERRY), performs a stochastic gradient descent of (3.4), with a constraint to avoid the trivial solution $\mathbf{w} = \mathbf{0}$ [53], [54]. For a SISO system, the basic MERRY algorithm is:

$$\text{MERRY:}\quad \boxed{\begin{aligned} &\text{Given } \Delta, \text{ for symbol } k = 0, 1, 2, \ldots, \\ &\quad \tilde{\mathbf{r}}(k) = \mathbf{r}(Mk + \nu + \Delta) - \mathbf{r}(Mk + \nu + N + \Delta) \\ &\quad e(k) = \mathbf{w}^T(k)\ \tilde{\mathbf{r}}(k) \\ &\hat{\mathbf{w}}(k+1) = \mathbf{w}(k) - \mu\ e(k)\ \tilde{\mathbf{r}}^*(k) \\ &\mathbf{w}(k+1) = \frac{\hat{\mathbf{w}}(k+1)}{\|\hat{\mathbf{w}}(k+1)\|} \end{aligned}} \tag{3.5}$$

where $\mathbf{r}(i) = [r(i), r(i-1), \ldots, r(i-L_w)]^T$, and * denotes complex conjugation. The norm can be the common L_2 norm, the L_p norm for p an integer, the norm with respect to a matrix, or any other conceivable norm.

Observe that MERRY is a simple vector update rule, with the added complexity of a renormalization. Due to the fact that MERRY compares the CP to the end of the symbol, only one update is possible per symbol. Alternate implementations of the constraint include fixing one tap to unity, maintaining a channel estimate and renormalizing to enforce $\|\mathbf{c}\| = 1$ instead of $\|\mathbf{w}\| = 1$, or including a penalty term in the cost function to enforce the norm constraint.

MERRY can also be implemented in transmitter-zero OFDM (TZ-OFDM) systems [90], as opposed to cyclic prefix OFDM (CP-OFDM) systems. TZ-OFDM systems transmit zeros during the guard period that is used for the cyclic prefix in CP-OFDM. This is equivalent to assuming that the samples in the CP ($x(1)$ and $x(2)$ in Figure 3.1) are zero, rather than copies of the data. The MERRY cost function then becomes

$$J_{merry,TZ}(\Delta) = 2\ \mathrm{E}\left[|y(Mk+\nu+\Delta)|^2\right], \qquad \Delta \in \{0,\ \ldots,\ M-1\}. \tag{3.6}$$

The update equation is a stochastic gradient descent of (3.6) with a periodic renormalization.

3.2 FRODO: MERRY's cousin

This section proposes a generalization to the MERRY cost function. This generalization allows for the use of more than one sample in the update rule and allows for channel shortening to variable window lengths (this last feature is incidental rather than specifically sought after). The generalization of MERRY will be referred to as Forced Redundancy with Optional Data Omission (FRODO) [64], for reasons that will become apparent.

Since there are ν samples in the cyclic prefix, a natural generalization is to compare more than one of these samples to their counterparts at the end of the symbol. Thus, for a CP-OFDM system, a more general cost function than the MERRY cost of (3.4) is

$$J_{frodo}(\Delta) = \sum_{i \in S_f} \mathrm{E}\left[|y(Mk+i+\Delta) - y(Mk+i+N+\Delta)|^2\right],$$
$$\Delta \in \{0,\ \ldots,\ M-1\}, \tag{3.7}$$

where $S_f \subset \{1, \cdots, \nu\}$ is an index set. Similarly, for a TZ-OFDM system, the FRODO cost function can be modifed to

$$J_{frodo,TZ}(\Delta) = 2 \sum_{i \in S_f} \mathrm{E}\left[|y(Mk+i+\Delta)|^2\right], \qquad \Delta \in \{0,\ \ldots,\ M-1\}. \tag{3.8}$$

These cost functions can be considered for either the SISO or the MIMO model.

For MERRY, $S_f = \{\nu\}$. Different sets allow for the use of more or less data, as well as for shortening to different channel lengths, which will be shown momentarily. Since the modified cost function allows the option of using all of the data in the CP ($S_f = \{1, \cdots, \nu\}$), or a single sample ($S_f = \{\nu\}$), or anything in between, we use the name Forced Redundancy with Optional Data Omission (FRODO) to refer to a stochastic gradient descent algorithm using this cost function. An equalization (not channel shortening) algorithm equivalent to using FRODO with the set $S_f = \{1, \cdots, \nu\}$ was proposed in [45]. The general FRODO algorithm includes both [45] and MERRY [53] as special cases.

3.3 Cost function analysis

This section analyzes the MERRY and FRODO cost functions, with the goal of showing that minimizing the cost function will suppress the unwanted energy in the impulse response of the effective channel.

The following theorem relates the MERRY and FRODO cost functions to the non-adaptive maximum shortening SNR (MSSNR) TEQ design [67]. It says that MERRY attempts to produce a "don't care" region with a width of ν taps, and FRODO uses a "don't care" region that is the intersection of multiple such windows.

Theorem 3.3.1 *For CP-OFDM systems and TZ-OFDM systems, the FRODO cost functions (3.7) and (3.8) simplify to*

$$J_{frodo} = 2 \sum_{i \in S_f} \sum_{l=1}^{L} \sigma_{x,l}^2 \|\mathbf{c}_{l,wall}^{i+\Delta}\|^2 + 2\,|S_f| \sum_{p=1}^{P} \mathbf{w}_p^H \mathbf{R}_{n,p} \mathbf{w}_p, \tag{3.9}$$

where

$$\|\mathbf{c}_{l,wall}^{i+\Delta}\|^2 = \sum_{j=0}^{\Delta+i-\nu-1} |c_l(j)|^2 + \sum_{j=\Delta+i}^{L_c} |c_l(j)|^2, \quad l \in \{1, \ldots, L\}, \tag{3.10}$$

and where

$$c_l(j) = \sum_{p=1}^{P} c_{p,l}(j), \quad j \in \{0, \ldots, L_c\}, \quad l \in \{1, \ldots, L\}. \tag{3.11}$$

Remarks on Theorem 3.3.1: A proof is given in Appendix A. For the case $L = P = 1$, and for $S_f = \{\nu\}$, we have the term $i = \nu$ only, which leads to shortening to a ν-length window (i.e. the basic MERRY algorithm). MERRY minimizes the energy outside of a length ν window plus the energy of the filtered noise, subject to a constraint (e.g. $\|\mathbf{w}\| = 1$ or $\|\mathbf{c}\| = 1$). In contrast, the MSSNR design [67] minimizes the energy of the combined impulse response outside of a window of length $\nu + 1$ (rather than ν), subject to some constraint (usually $\|\mathbf{c}_{win}\| = 1$). For the case $L = 1$, $P > 1$, and $S_f = \{\nu\}$, the cost function suppresses the tails of the averaged channel, $\mathbf{c} = \sum_p \mathbf{c}_p$, allowing for diversity gain. For the case $L > 1$, $P = 1$, and $S_f = \{\nu\}$, the cost function suppresses the average of the tail energies of the L channels (rather than the tail energy of the average channel in the previous case), effectively shortening all L channels at once. In this case, the demodulated signal will be the sum of the transmitted signals of the L users, but the signal will be free of ISI and ICI. Thus, in an MC-CDMA scenario, the L signals can now be separated using the spreading codes, which would not have been possible if the channels were not shortened first. The case $L = P = 1$ with $S_f \supsetneq \{\nu\}$ will be discussed in the pedagogical example after Theorem 3.3.2.

Theorem 3.3.2 *If we relax assumption A4 so that* $x_l(k) = x(k)\ \forall l \in \{1, \cdots, L\}$*, i.e. multiple transmit antennas for a single user, then the FRODO cost functions (3.7) and (3.8) simplify to*

$$J_{frodo} = 2\sigma_x^2 \sum_{i \in S_f} \|\mathbf{c}_{wall}^{i+\Delta}\|^2 + 2\,|S_f| \sum_{p=1}^{P} \mathbf{w}_p^H \mathbf{R}_{n,p} \mathbf{w}_p, \tag{3.12}$$

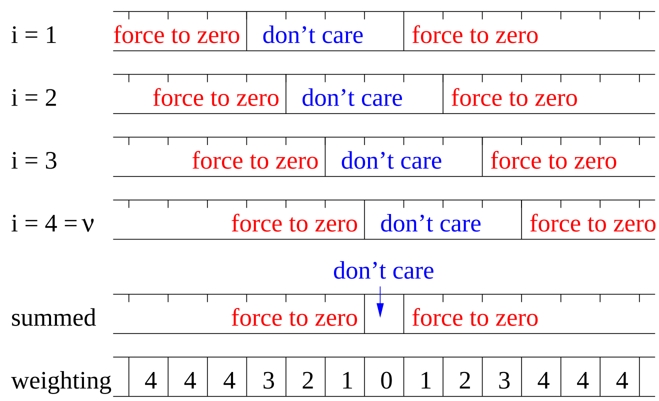

Figure 3.2: The relation of the "don't care" windows in the different terms of the FRODO cost function, for $\nu = 4$. The line "summed" indicates the effect of considering all four terms at once, and the line "weighting" indicates how much emphasis the total cost function places on forcing each tap to zero.

where

$$\|\mathbf{c}_{wall}^{i+\Delta}\|^2 = \sum_{j=0}^{\Delta+i-\nu-1} |c(j)|^2 + \sum_{j=\Delta+i}^{L_c} |c(j)|^2 , \tag{3.13}$$

and where

$$c(j) = \sum_{p=1}^{P} \sum_{l=1}^{L} c_{p,l}(j), \quad j \in \{0, \ldots, L_c\} . \tag{3.14}$$

Remarks on Theorem 3.3.2: The proof of Theorem 3.3.2 follows along the same lines as the proof of Theorem 3.3.1, except with the definition of (3.14) instead of (3.11), hence the details of the proof are omitted. In this case, FRODO will try to minimize the tail energy of the average channel, which is averaged over both p and l.

The effect of using more than one comparison [more than one value of i in (3.7)] can be illustrated as follows. For simplicity, let $L = P = 1$ and drop the subscripts l and p. Consider using the "full" index set, $i \in \{1, \cdots, \nu\}$. The total cost function is the sum of (A.7) over these values of i:

$$J = 2\sigma_x^2 \sum_{i=1}^{\nu} \|\mathbf{c}_{wall}^{i+\Delta}\|^2 + 2 \sum_{i=1}^{\nu} \mathbf{w}^H \mathbf{R}_n \mathbf{w}. \tag{3.15}$$

Figure 3.2 shows a pictorial example for $\nu = 4$. Each time i is incremented, the window location shifts over by one sample. The summed cost function takes the form

$$\begin{aligned} J_{frodo} = 2\ \sigma_x^2 \big[&\nu\ c^2(0) + \nu\ c^2(1) + \cdots + \nu\ c^2(\Delta - \nu) + \\ &(\nu - 1)\ c^2(\Delta - \nu + 1) + \cdots + 2\ c^2(\Delta - 2) + c^2(\Delta - 1) + \\ &c^2(\Delta + 1) + 2\ c^2(\Delta + 2) + \cdots + (\nu - 1)\ c^2(\Delta + \nu - 1) + \\ &\nu\ c^2(\Delta + \nu) + \nu\ c^2(\Delta + \nu + 1) + \cdots + \nu\ c^2(L_c) \big] \\ &+ 2\ \nu\ \mathbf{w}^H \mathbf{R}_n \mathbf{w}. \end{aligned} \tag{3.16}$$

Thus, the "full" frodo cost function tries to suppress all of the taps of the effective channel except for the Δ^{th} tap, and the noise gain is limited as well. Taps farther from the center are more heavily weighted, and hence should be smaller. This makes the cost function very similar to the minimum delay spread (MDS) algorithm [92] which minimizes

$$J_{MDS} = \sum_{j=0}^{L_c} |j - \Delta|^2\, c^2(j), \tag{3.17}$$

subject to a unit norm constraint on the effective channel, $\|\mathbf{c}\| = 1$. A variant of the MDS algorithm proposed in [97] uses linear weights, rather than quadratic:

$$\hat{J}_{MDS} = \sum_{j=0}^{L_c} |j - \Delta|\, c^2(j). \tag{3.18}$$

Notice that this alternate MDS penalty increases linearly with the distance from tap Δ, and the FRODO penalty [in this example which uses all possible values of i in the summation in (3.7)] increases linearly for a distance of ν on each side of tap Δ, and then remains fixed at that penalty level for larger distances. As a consequence of suppressing all taps save the Δ^{th} tap, FRODO will attempt to shorten the channel to an impulse function with delay Δ, but with a tendency to minimize delay spread rather than simply to equalize. Since the ISI and ICI caused by channel taps increase with their distance from the Δ^{th} tap [92], the delay spread minimizing nature of FRODO is more advantageous for a TEQ than a traditional zero-forcing equalizer, yet less advantageous than an algorithm designed for channel shortening (such as the MSSNR design [67]). The linearly increasing penalty function of (3.18) may have the effect of amplifying the effects of channel estimation errors [15]. The fact that the FRODO penalty function levels off after a certain distance prevents this from happening.

As a compromise, if a window size between 1 and ν is desired, then the index set for FRODO can be changed accordingly. If two comparisons are made in (3.7) rather than one as in (3.4), then the algorithm has access to more data per update, hence convergence should be faster and smoother. The penalty is that the window size will be smaller by one sample.

3.4 Equivalent problem statements

In this section, the FRODO design problem is re-cast into several formulations that are mathematically equivalent but quite different in appearance. This equivalence will be used in the next section to pick a "good" problem statement which allows for an update rule that has no divisions or square roots, and allows for the use of alternate constraints.

Defining "stacked" amalgamations of various vectors as

$$\mathbf{r}_p(j) = \left[r_p(j), r_p(j-1), \cdots, r_p(j-L_w)\right]^T, \quad p \in \{1, \cdots, P\}, \tag{3.19}$$

$$
\begin{aligned}
\mathbf{r}(j) &= \left[\mathbf{r}_1^T(j), \mathbf{r}_2^T(j), \cdots, \mathbf{r}_P^T(j)\right]^T, && (3.20)\\
\tilde{\mathbf{r}}_i(k) &= \mathbf{r}(Mk+i+\Delta) - \mathbf{r}(Mk+i+N+\Delta), \quad \forall i \in S_f && (3.21)\\
\mathbf{w} &= \left[\mathbf{w}_1^T, \mathbf{w}_2^T, \cdots, \mathbf{w}_P^T\right]^T, && (3.22)
\end{aligned}
$$

the FRODO cost function (3.7) can be rewritten as

$$
\begin{aligned}
J_{frodo} &= \sum_{i \in S_f} \mathrm{E}\left[\left|\tilde{\mathbf{r}}_i^T(k)\mathbf{w}\right|^2\right] && (3.23)\\
&= \mathbf{w}^H \underbrace{\sum_{i \in S_f} \underbrace{\mathrm{E}\left[\tilde{\mathbf{r}}_i^*(k)\tilde{\mathbf{r}}_i^T(k)\right]}_{\mathbf{A}_i}}_{\mathbf{A}} \mathbf{w}. && (3.24)
\end{aligned}
$$

Under our four assumptions, it can be shown that

$$
\mathbf{A}_i = 2\sum_{l=1}^{L} \sigma_{x,l}^2 \begin{bmatrix} \mathbf{H}_{1,l,wall}^H \mathbf{H}_{1,l,wall}, & \cdots, & \mathbf{H}_{1,l,wall}^H \mathbf{H}_{P,l,wall} \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{P,l,wall}^H \mathbf{H}_{1,l,wall}, & \cdots, & \mathbf{H}_{P,l,wall}^H \mathbf{H}_{P,l,wall} \end{bmatrix} + 2 \begin{bmatrix} \mathbf{R}_{n,1}, & \cdots, & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0}, & \cdots, & \mathbf{R}_{n,P} \end{bmatrix} \qquad (3.25)
$$

where $\mathbf{H}_{p,l,wall}$ is obtained by forming the convolution matrix $\mathbf{H}_{p,l}$ for channel p, l and removing rows $\Delta + i - \nu$ through $\Delta + i - 1$, similar to $\mathbf{H}_{wall}$ in [67]. (The proof is quite similar to the proof of Theorem 3.3.1, and hence is omitted.) We wish to minimize (3.24), with some constraint to avoid the trivial solution $\mathbf{w} = \mathbf{0}$.

The FRODO cost function is a measure of the energy in the "wall" portion of the effective channel. Also of interest are the energy in the "window" portion of the effective channel, and the total energy of the effective channel. To this end, we define

$$
\begin{aligned}
J_{win} &= 2\sum_{i \in S_f} \mathrm{E}\left[y^*(Mk+i+\Delta)\; y(Mk+i+N+\Delta)\right]\\
&= \mathbf{w}^H \underbrace{\left(\sum_{i \in S_f} \underbrace{2\;\mathrm{E}\left[\mathbf{r}^*(Mk+i+\Delta)\; \mathbf{r}^T(Mk+i+N+\Delta)\right]}_{\mathbf{B}_i}\right)}_{\mathbf{B}} \mathbf{w}, \qquad (3.26)
\end{aligned}
$$

and

$$
\begin{aligned}
J_{total} &= 2\,|S_f|\ \mathrm{E}\left[|y(Mk+i_o+\Delta)|^2\right], \quad i_o \in \{0,\cdots,M-1\} \\
&= \mathbf{w}^H \underbrace{\left(2\,|S_f|\ \mathrm{E}\left[\mathbf{r}^*(Mk+i_o+\Delta)\ \mathbf{r}^T(Mk+i_o+\Delta)\right]\right)}_{\mathbf{C}} \mathbf{w}.
\end{aligned} \tag{3.27}
$$

It can be shown that the $\mathbf{B}_i$ matrices have the same form as the $\mathbf{A}_i$ matrices in (3.25), except with $\mathbf{H}_{p,l,wall}$ replaced by $\mathbf{H}_{p,l,win}$ (which equals rows $\Delta + i - \nu$ through $\Delta + i - 1$ of the channel convolution matrix $\mathbf{H}_{p,l}$), and the $\mathbf{C}$ matrix has the same form as the $\mathbf{A}_i$ matrices, except with $\mathbf{H}_{p,l,wall}$ replaced by $\mathbf{H}_{p,l}$.

Theorem 3.4.1 *Under the assumtions in Section 3.2, the following optimization problems all produce the same solution* $\mathbf{w}_{opt}$*, up to a scale factor:*

$$
\begin{aligned}
\mathbf{w}_{opt}^1 &= \arg\min_{\mathbf{w}} J_{frodo} \ \textit{such that}\ J_{win} = 1 & (3.28)\\
\mathbf{w}_{opt}^2 &= \arg\max_{\mathbf{w}} J_{win} \ \textit{such that}\ J_{frodo} = 1 & (3.29)\\
\mathbf{w}_{opt}^3 &= \arg\min_{\mathbf{w}} J_{frodo} \ \textit{such that}\ J_{total} = 1 & (3.30)\\
\mathbf{w}_{opt}^4 &= \arg\max_{\mathbf{w}} J_{total} \ \textit{such that}\ J_{frodo} = 1 & (3.31)\\
\mathbf{w}_{opt}^5 &= \arg\min_{\mathbf{w}} J_{total} \ \textit{such that}\ J_{win} = 1 & (3.32)\\
\mathbf{w}_{opt}^6 &= \arg\max_{\mathbf{w}} J_{win} \ \textit{such that}\ J_{total} = 1. & (3.33)
\end{aligned}
$$

Proof: For simplicity of notation, we consider the case $S_f = \{\nu\}$. The general case is straitforward but more tedious. Let $\mathbf{u}_1 = \mathbf{r}(Mk + \nu + \Delta)$, $\mathbf{u}_2 = \mathbf{r}(Mk + \nu + N + \Delta)$, and $\mathbf{u}_3 = \mathbf{u}_1 - \mathbf{u}_2$. Note that $\mathrm{E}\left[\mathbf{u}_1^*\mathbf{u}_1^T\right] = \mathrm{E}\left[\mathbf{u}_2^*\mathbf{u}_2^T\right]$ and $\mathrm{E}\left[\mathbf{u}_1^*\mathbf{u}_2^T\right] = \mathrm{E}\left[\mathbf{u}_2^*\mathbf{u}_1^T\right]$ for uncorrelated source sequences. Then

$$
\underbrace{\mathrm{E}\left[\mathbf{u}_3^*\mathbf{u}_3^T\right]}_{\mathbf{A}} = \mathrm{E}\left[\mathbf{u}_1^*\mathbf{u}_1^T + \mathbf{u}_2^*\mathbf{u}_2^T - \mathbf{u}_1^*\mathbf{u}_2^T - \mathbf{u}_2^*\mathbf{u}_1^T\right] = \underbrace{\mathrm{E}\left[2\mathbf{u}_1^*\mathbf{u}_1^T\right]}_{\mathbf{C}} - \underbrace{\mathrm{E}\left[2\mathbf{u}_1^*\mathbf{u}_2^T\right]}_{\mathbf{B}}, \tag{3.34}
$$

i.e. $\mathbf{A}+\mathbf{B}=\mathbf{C}$. Thus,

$$\begin{aligned}
\mathbf{w}_{opt}^{1} &= \arg\min_{\mathbf{w}} \frac{\mathbf{w}^H\mathbf{A}\mathbf{w}}{\mathbf{w}^H\mathbf{B}\mathbf{w}} && (3.35)\\
&= \arg\max_{\mathbf{w}} \frac{\mathbf{w}^H\mathbf{B}\mathbf{w}}{\mathbf{w}^H\mathbf{A}\mathbf{w}} = \mathbf{w}_{opt}^{2} && (3.36)\\
&= \arg\max_{\mathbf{w}} \left(\frac{\mathbf{w}^H\mathbf{B}\mathbf{w}}{\mathbf{w}^H\mathbf{A}\mathbf{w}} + \underbrace{\frac{\mathbf{w}^H\mathbf{A}\mathbf{w}}{\mathbf{w}^H\mathbf{A}\mathbf{w}}}_{1} \right) && (3.37)\\
&= \arg\max_{\mathbf{w}} \frac{\mathbf{w}^H\mathbf{C}\mathbf{w}}{\mathbf{w}^H\mathbf{A}\mathbf{w}} = \mathbf{w}_{opt}^{4}. && (3.38)
\end{aligned}$$

The remaining equivalence relations are proven in a similar fashion. ■

Thus, we can transform our original constrained minimization problem into various constrained maximization problems. Chatterjee, *et al.* [16], have proposed an iterative algorithm which can solve optimization problems of the form of (3.29), (3.31), (3.33), rather than the form of (3.28), (3.30), (3.32). We will combine Chatterjee's algorithm with the results of Theorem 3.4.1 in the next section.

It is of interest to note that in the SISO case, the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ used in the preceeding proof can be simplifed to

$$\mathbf{A} = \mathbf{H}_{wall}^H\mathbf{H}_{wall} + \mathbf{R}_n \tag{3.39}$$

$$\mathbf{B} = \mathbf{H}_{win}^H\mathbf{H}_{win} \tag{3.40}$$

$$\mathbf{C} = \mathbf{H}^H\mathbf{H} + \mathbf{R}_n = \mathbf{A}+\mathbf{B}, \tag{3.41}$$

where $\mathbf{H}\mathbf{w}$, $\mathbf{H}_{win}\mathbf{w}$, and $\mathbf{H}_{wall}\mathbf{w}$ form the effective channel, the windowed effective channel, and the effective channel outside the window, respectively, as in [67]. Thus, from Theorem 3.4.1, the minimization of the MERRY cost (3.4) under the constraint $\mathrm{E}\left[y^2(k)\right]=1$ (which is $\|\mathbf{c}\|_2^2=1$ in the noiseless case) yields the same solution (up to a scale factor) as the minimization of (3.4) under the constraint $\|\mathbf{c}_{win}\|_2^2=1$.

If we enforce the constraint by monitoring the TEQ output energy $\mathrm{E}\left[y^2(k)\right]$ and forcing it to unity, then MERRY will converge to the MSSNR solution in the noiseless case. In the noise case, MERRY converges to the MMSE solution (assuming a white input signal), since the only difference between the MSSNR solution and the MMSE solution is that the MMSE solution includes the noise correlation $\mathbf{R}_n$ in the penalty function [56], and MERRY includes this term as well.

3.5 Division-free update rule

This section derives the FRODO update algorithm. The periodic renormalization (with square root and division) of MERRY [53] and of other adaptive and iterative TEQ designs [8], [17], [23], [30], is avoided by using a Lagrangian constraint in the manner of [16]. The adaptive generalized eigen-decomposition algorithm of [16] was proposed in the context of neural networks; and it has been applied to trained, iterative (not adaptive) TEQ design in [15]. The update algorithm is a gradient ascent of a cost function ($\mathbf{w}^H\mathbf{C}\mathbf{w}$) with a Lagrangian constraint ($\mathbf{w}^H\mathbf{A}\mathbf{w} = 1$), and it is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\left(\mathbf{C}\mathbf{w} - \mathbf{A}\mathbf{w}\left(\mathbf{w}^H\mathbf{C}\mathbf{w}\right)\right). \tag{3.42}$$

A proof of the global convergence of this algorithm to the maximum generalized eigenvalue of the matrix pencil $(\mathbf{C}, \mathbf{A})$ in the case of real parameters was given in [16]. In the case of FRODO, we have blind, stochastic approximations of $\mathbf{A}$ and $\mathbf{C}$ available at the receiver, which are obtained by removing the expectations in (3.24) and (3.27). Making use of these estimates in (3.42), the FRODO update rule is

$$
\boxed{
\begin{aligned}
&\text{Given } \Delta \text{ and } i_o, \text{ for symbol } k = 0, 1, 2, \ldots, \\
&\quad \tilde{\mathbf{r}}_i(k) = \mathbf{r}(Mk + i + \Delta) - \mathbf{r}(Mk + i + N + \Delta), \quad \forall i \in S_f \\
&\quad e_i(k) = \mathbf{w}^T(k)\, \tilde{\mathbf{r}}_i(k), \quad \forall i \in S_f \\
&\quad y_{i_o}(k) = y(Mk + i_o + \Delta) = \mathbf{w}^T \mathbf{r}(Mk + i_o + \Delta) \\
&\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\, y_{i_o}(k) \left(\mathbf{r}^*(Mk + i_o + \Delta) - y_{i_o}^*(k) \sum_{i \in S_f} e_i(k) \tilde{\mathbf{r}}_i^*(k) \right)
\end{aligned}
}
\tag{3.43}
$$

where the constant $2\,|S_f|$ has been absorbed into the step size μ. When $S_f = \{\nu\}$ and $P = L = 1$, we obtain an algorithm that is similar to the MERRY algorithm (3.5) without the renormalization. The algorithm of (3.43) requires approximately $(2\,|S_f| + 3)P\tilde{L}_w$ multiplications and $(3\,|S_f| + 2)P\tilde{L}_w$ additions, where $|S_f|$, the size of the set S_f, is usually 1.

An alternate approach for creating a division-free update rule is to reparametertize the TEQ in generalized spherical coordinates, as in [84]. However, this leads to the constraint $\|\mathbf{w}\|_2 = 1$, without the option of the more appropriate $\|\mathbf{c}\|_2 = 1$ or $\|\mathbf{c}_{win}\|_2 = 1$, both of which are possible for the FRODO algorithm (3.43).

3.6 Global convergence

Since MERRY is essentially an iterative eigensolver, a proof of its global convergence is readily derivable.

Theorem 3.6.1 *The expectation of the trajectory of the MERRY algorithm of (3.5) has $L_w + 1$ stationary points. All of the stationary points corresponding to local minima have the same value of the cost function, and thus are global minima. I.e., the MERRY algorithm is globally convergent.*

Proof: Define

$$\begin{aligned}\mathbf{r}_j &= [r(j), r(j-1), \ldots, r(j-L_w)]^T \\ \tilde{\mathbf{r}}_i &= \mathbf{r}_j - \mathbf{r}_{j+N}\end{aligned} \tag{3.44}$$

Adding a Lagrangian constraint, the cost function becomes

$$\begin{aligned}J_{merry}(\Delta) &= \mathrm{E}\left[|y(\nu+\Delta) - y(\nu+\Delta+N)|^2\right] + \lambda\left(1 - \mathbf{w}^H\mathbf{w}\right) \\ &= \mathrm{E}\left[\left|\mathbf{w}^T\mathbf{r}_{\nu+\Delta} - \mathbf{w}^T\mathbf{r}_{\nu+\Delta+N}\right|^2\right] + \lambda\left(1 - \mathbf{w}^H\mathbf{w}\right) \\ &= \mathbf{w}^H \underbrace{\mathrm{E}\left[\tilde{\mathbf{r}}^*_{\nu+\Delta}\tilde{\mathbf{r}}^T_{\nu+\Delta}\right]}_{\mathbf{A}} \mathbf{w} + \lambda\left(1 - \mathbf{w}^H\mathbf{w}\right),\end{aligned}$$

with gradient and Hessian

$$\bigtriangledown_{\mathbf{w}} J_{merry}(\Delta) = 2\left(\mathbf{A}\mathbf{w} - \lambda\mathbf{w}\right), \tag{3.45}$$

$$\mathcal{H}_{\mathbf{w}} J_{merry}(\Delta) = 2\left(\mathbf{A} - \lambda\mathbf{I}\right). \tag{3.46}$$

The gradient is zero if and only if $(\mathbf{w}, \lambda)$ are an eigenpair of $\mathbf{A}$, hence there are exactly $L_w + 1$ stationary points. The Hessian is positive definite (corresponding to a local minima) if and only if we choose λ to be the smallest eigenvalue. If the smallest eigenvalue is repeated, then there will be multiple minima but all will have the same cost (equal to the repeated eigenvalue). This completes the proof.

■

The proof is similar for the MERRY algorithm using the constraints $\mathbf{w}^T\mathbf{B}\mathbf{w} = 1$, $\|\mathbf{c}\| = 1$, and $\mathrm{E}\left[y^2(k)\right] = 1$. The division-free FRODO algorithm of (3.43) is based on the generalized eigenvector computation algorithm of [16], which is globally convergent. However, the FRODO algorithm replaces the product of two expectations with the instantaneous values of their arguments. Since the expectation of a product is not the product of the expecatations, we cannot gaurantee that the stochastic version of FRODO will converge.

3.7 Initialization

This section proposes a blind, non-adaptive approach to solving the FRODO cost function. If the matrices **A** and **C** in (3.24) and (3.27) are ill-conditioned, then the FRODO algorithm will have slow modes of convergence, as with the LMS algorithm [110]. One way to avoid this is to accumulate the estimates of **A** and **C** from the data,

$$\hat{\mathbf{A}} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i \in S_f} \tilde{\mathbf{r}}_i^*(k)\ \tilde{\mathbf{r}}_i^T(k), \tag{3.47}$$

$$\hat{\mathbf{C}} = \frac{2\,|S_f|}{K} \sum_{k=1}^{K} \mathbf{r}^*(Mk + i_o + \Delta)\ \mathbf{r}^T(Mk + i_o + \Delta), \tag{3.48}$$

and then find the generalized eigenvector corresponding to the maximum generalized eigenvalue of $(\hat{\mathbf{C}}, \hat{\mathbf{A}})$. However, if P is large, this approach is computationally intensive. A computationally cheaper solution would be to individually solve the $P = 1$ eigendecomposition for each $\mathbf{w}_p$, $p \in \{1, \cdots, P\}$, and then let the adaptive FRODO algorithm refine these estimates.

Simulation results have shown that this non-adaptive approach blindly finds a near-optimum solution for the TEQ, and under ergodicity assumptions, this approach will find the optimal (in terms of the FRODO cost) solution if an infinite amount of data is used in the time-averaging. The simulation results will be presented in Chapter 8.

3.8 Symbol synchronization

One of the difficulties with the MERRY and FRODO algorithms is that the user must choose a value for the symbol synchronization parameter Δ before the algorithm is run. This section shows how the basic idea in MERRY/FRODO can be used to obtain a reasonable heuristic choice for the symbol synchronization parameter.

We propose the following heuristic: given the delay Δ_{peak} in which maximizes the energy of the average (unshortened) channel in a window of taps Δ through $\Delta+\nu-1$, a near-optimum delay (in the shortening SNR sense [67]) can be obtained by choosing

$$\Delta = \Delta_{peak} + \left\lfloor \frac{L_w}{2} \right\rfloor . \tag{3.49}$$

There are two issues to be addressed: (1) the means of obtaining Δ_{peak}, and (2) the validity of this heuristic. These will be addressed in order.

In the absence of a TEQ (i.e. $\mathbf{w} = 1$), $\mathbf{c}_{p,l} = \mathbf{h}_{p,l}$. From Theorem 3.3.1, if we only make one comparison using $i = \nu$,

$$J_{frodo}\left(\Delta\right) = 2 \sum_{l=1}^{L} \sigma_{x,l}^2 \|\mathbf{h}_{l,wall}^{\nu+\Delta}\|^2 + 2 \sum_{p=1}^{P} \sigma_{n,p}^2, \tag{3.50}$$

with an analogous definition for $\mathbf{h}_{l,wall}^{\nu+\Delta}$ as for $\mathbf{c}_{l,wall}^{\nu+\Delta}$. Since

$$\|\mathbf{h}_l\|^2 = \|\mathbf{h}_{l,win}^{\nu+\Delta}\|^2 + \|\mathbf{h}_{l,wall}^{\nu+\Delta}\|^2, \tag{3.51}$$

the index Δ_{peak} in which the average windowed channel energy is highest is the index in which the average walled channel energy is smallest. Thus, Δ_{peak} can be estimated by minimizing an estimate of $J_{frodo}\left(\Delta\right)$ over Δ,

$$\hat{\Delta}_{peak} = \arg \min_{0 \le \Delta \le M-1} \sum_{k=1}^{K} \left| r(Mk+\nu+\Delta) - r(Mk+\nu+N+\Delta) \right|^2 \tag{3.52}$$

for some number of symbols K. This approach only requires $M \cdot K$ multiplications and $M \cdot (2K-1)$ additions to compute the function to be minimized, and then $M-1$ comparisons to find the minimum. Thus, a sufficiently large value of K can be used for an accurate estimate with low computational complexity. Moreover,

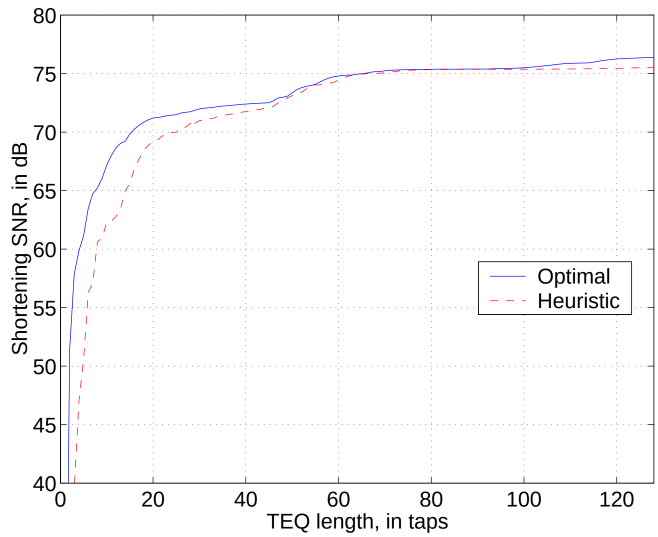

Figure 3.3: Shortening SNR versus TEQ length for FRODO using the optimal and a heuristic delay.

this heuristic can be applied to other design methods (besides FRODO) to avoid a global search over the delay parameter.

The second question is whether or not this heuristic is valid. Figure 3.3 shows a plot of the shortening SNR achieved by the delay-optimized FRODO design and by the FRODO design using the heuristic delay of (3.49). For simplicity, $P = L = |S_f| = 1$. The performance was averaged over carrier serving area (CSA) loops 1 through 8 [7] (standard synthetic ADSL test channels), and the window size was 32 taps. The heuristic delay provides reasonable performance relative to the optimal delay for TEQs with at least 8 taps, and very nearly optimal performance for TEQs with at least 32 taps. For TEQs shorter than 8 taps, the range of "good" delay choices will be small, so a heuristic approach may not be adequate. For ADSL, typical TEQ lengths are 16 or 32 taps. Other heuristics may be used; the proposed approach is merely one method which generally works and is blind.

Chapter 4

Correlation-based Adaptive Equalizers

"By the discovery we shall be shorten'd in our aim."

– William Shakespeare, *Coriolanus*, Act I, Scene ii.

This chapter[1] is devoted to adaptive TEQs that rely on correlation estimates. The Sum-squared Auto-correlation Minimization (SAM) algorithm [8], [9], attempts to blindly shorten the autocorrelation of the data at the output of the channel shortener. The Trained OFDM L_2-norm Correlation-based Iterative Equalization with Normalization (TOLKIEN) algorithm assumes that training can be used to shorten the cross-correlation between the effective channel's input and output. Both algorithms have much higher complexity than the MERRY and FRODO algorithms, but have the advantage that they can update every sample rather than once per symbol. SAM behaves much like the CMA equalization algorithm, in that it does not require the user to specify the desired delay and can adapt before carrier frequency offset (CFO) recovery is performed.

4.1 SAM: a rapidly converging, blind, adaptive channel shortener

Since SAM does not rely on the presence of a cyclic prefix, it need not be used exclusively for multicarrier systems. Thus, a more general system model is shown in Figure 4.1. As before, $x(k)$ is the source sequence to be transmitted through a linear finite-impulse-response (FIR) channel $\mathbf{h}$; and $r(k)$ is the received signal,

[1]Some material in this chapter has been previously published. © 2004 IEEE. Reprinted, with permission, from [8] and [9].

which will be filtered through a channel shortener (i.e. TEQ) with an impulse response vector $\mathbf{w}$ to obtain the output sequence $y(k)$. Let $\mathbf{c} = \mathbf{h} \star \mathbf{w}$ denote the effective channel-equalizer impulse response vector. The TEQ will be adapted with the goal of shortening the effective channel $\mathbf{c}$ such that it possesses significant coefficients only within a contiguous window of size $\nu + 1$ taps. In multicarrier systems, ν is the CP length, but more generally it is the desired window length minus one. The received sequence $r(k)$ is

$$r(k) = \sum_{j=0}^{L_h} h(j)x(k-j) + n(k), \tag{4.1}$$

and the output of the TEQ is

$$y(k) = \sum_{j=0}^{L_w} w(j)r(k-j) = \mathbf{w}^T \mathbf{r}(k), \tag{4.2}$$

where $\mathbf{r}(k) = [r(k), r(k-1), \cdots, r(k-L_w)]^T$. Throughout this chapter, we make the following assumptions.

A1. The source sequence $x(k)$ is white, zero-mean and wide-sense stationary (W.S.S).

A2. The relation $2L_c < N$ holds for multicarrier (or block-based[2]) systems, i.e. the effective channel has length less than half the FFT (or block) size.

A3. The source sequence $x(k)$ has unit variance: $\sigma_x^2 = 1$.

A4. The noise sequence $n(k)$ is zero-mean, i.i.d., uncorrelated to the source sequence, and has variance σ_n^2.

[2] Vaidyanathan and Vrcelj [102] have proposed the use of a block structure and a cyclic prefix for single-carrier systems, in which case channel shortening may be needed.

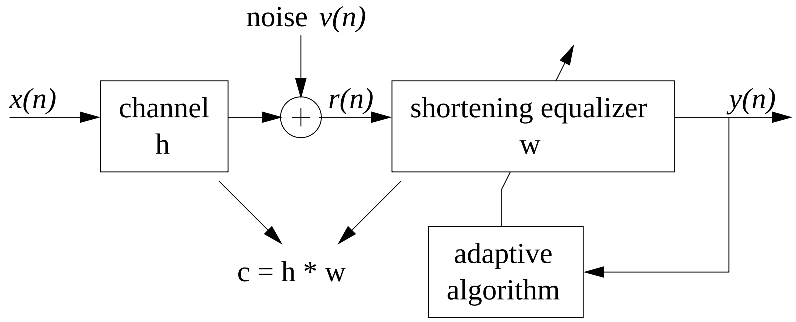

Figure 4.1: System model for SAM.

Assumption A1 is critical for the proposed channel shortening algorithm. In a multicarrier system, since the IFFT matrix is an unitary transformation, the transmitted sequence $x(k)$ will be white, zero-mean and W.S.S if the message vector $X(k)$ is white, zero-mean and W.S.S. Assumption A2 is important for analytical reasons, but if it is modestly violated the performance degradation should be minor. This assumption is irrelevant for the application of SAM to equalization of (non-CP-based) single carrier systems. Assumptions A3 and A4 are for notational simplicity. In [9], the parameters were assumed to be real, and in [78], they were generalized to the complex case. This thesis assumes complex parameters for generality.

Sum-squared Auto-correlation Minimization

This section motivates the use of the SAM cost function, and shows how to blindly measure it from the data. Consider the auto-correlation sequence of the combined channel-equalizer impulse response, i.e.,

$$R_{cc}(l) = \sum_{j=0}^{L_c} c^*(j)\ c(j-l). \tag{4.3}$$

For the effective channel $\mathbf{c}$ to have zero-valued taps outside a window of size $\nu + 1$, the auto-correlation values $R_{cc}(l)$ must be zero outside a window of length $2\nu + 1$,

$$R_{cc}(l) = 0, \qquad \forall\, |l| > \nu\ . \tag{4.4}$$

Hence, one possible way of performing channel shortening is by ensuring that (4.4) is satisfied by the auto-correlation function of the combined response. However, this has a trivial solution when $\mathbf{c} = \mathbf{0}$ or equivalently $\mathbf{w} = \mathbf{0}$. This trivial solution can be avoided by imposing a norm constraint on the effective channel, for instance $\|\mathbf{c}\|_2^2 = 1$, or equivalently $R_{cc}(0) = 1$.

It should be noted that perfect nulling of the auto-correlation values outside the window of interest is not possible, since perfect channel shortening is not possible when a finite length baud-spaced TEQ is used. This is because if the channel has L_h zeros, then the effective response will always have $L_h + L_w$ zeros. If we had decreased the length of the channel to, say, $L_s < L_h$ taps, then the combined response would only have L_s zeros, which contradicts our previous statement.

Hence, we define a cost function $J_{\nu+1}$ in an attempt to minimize (instead of nulling) the sum-squared auto-correlation terms, i.e.,

$$J_{SAM}(\nu) = \sum_{l=\nu+1}^{L_c} |R_{cc}(l)|^2. \tag{4.5}$$

The TEQ optimization problem can then be stated as

$$\mathbf{w}_{opt}^{sam} = \arg \min_{\mathbf{w}:\|\mathbf{c}\|_2^2=1} J_{SAM}(\nu). \tag{4.6}$$

Consider the auto-correlation function of the sequence $y(k)$,

$$\begin{aligned} R_{yy}(l) &= \mathrm{E}\left[y^*(k)y(k-l)\right] \\ &= \mathrm{E}\left[\left(\mathbf{c}^H\mathbf{x}^*(k) + \mathbf{w}^H\mathbf{n}^*(k)\right)\left(\mathbf{x}^T(k-l)\mathbf{c} + \mathbf{n}^T(k-l)\mathbf{w}\right)\right], \end{aligned} \tag{4.7}$$

where $\mathbf{x}(k) = [x(k), x(k-1), \cdots, x(k-L_c)]^T$, and $\mathbf{n}(k) = [n(k), n(k-1), \cdots, n(k-L_w)]^T$. To simplify,

$$\mathrm{E}\left[\mathbf{n}^*(k)\mathbf{n}^T(k-l)\right] = \begin{bmatrix} R_{nn}(l) & \cdots & R_{nn}(l+L) \\ \vdots & \ddots & \vdots \\ R_{nn}(l-L) & \cdots & R_{nn}(l) \end{bmatrix} \tag{4.8}$$

where $R_{nn}(l) = \mathrm{E}\left[n^*(k)n(k-l)\right]$. Since $n(k)$ is i.i.d., this matrix will be Toeplitz, with only one diagonal of nonzero entries. It becomes a shifting matrix, i.e. its affect on a vector is to shift the elements of the vector up or down (depending on l). Since the signal and noise are uncorrelated, $\mathrm{E}\left[\mathbf{x}^*(k)\mathbf{n}^T(k-l)\right] = \mathbf{0}$ and $\mathrm{E}\left[\mathbf{n}^*(k)\mathbf{x}^T(k-l)\right] = \mathbf{0}$. Finally, $\mathrm{E}\left[\mathbf{x}^*(k)\mathbf{x}^T(k-l)\right]$ becomes another shifting matrix, provided that the assumption $2\left(L_h + L_w\right) < N$ holds. If this is violated, then the matrix is still Toeplitz, but for some values of l there will be another diagonal of nonzero entries, corresponding to the correlation between samples in the transmitted symbol end and samples in the transmitted cyclic prefix. Fortunately, assumption A2 is a reasonable one, as can be seen by considering the CSA test loop channels [93] for the case of DSL: $L_h \cong 200$, $L_w \cong 32$, and $N = 512$, so $2\left(200 + 32\right) < 512$.

Now (4.7) can be simplified to

$$\begin{aligned} R_{yy}(l) &= \sum_{j=0}^{L_c} c^*(j)c(j-l) + \sigma_n^2 \sum_{j=0}^{L_w} w^*(j)w(j-l) \\ &= R_{cc}(l) + \sigma_n^2\, R_{ww}(l). \end{aligned} \tag{4.9}$$

Under the noiseless scenario, $R_{yy}(l) = R_{cc}(l)$ and hence equation (4.5) can be rewritten as

$$J_{SAM}(\nu) = \sum_{l=\nu+1}^{L_c} \left|R_{cc}(l)\right|^2 = \sum_{l=\nu+1}^{L_c} \left|R_{yy}(l)\right|^2. \tag{4.10}$$

In the presence of noise, (4.10) is only approximately true. This suggests approximating the cost function of (4.5) by

$$\begin{aligned}\hat{J}_{SAM}(\nu) &= \sum_{l=\nu+1}^{L_c} |R_{yy}(l)|^2 \\ &= \sum_{l=\nu+1}^{L_c} |R_{cc}(l)|^2 + 2\sigma_n^2 \sum_{l=\nu+1}^{L_w} \mathcal{R}\left\{R_{cc}(l)R_{ww}(l)\right\} + \sigma_n^4 \sum_{l=\nu+1}^{L_w} |R_{ww}(l)|^2\,,\end{aligned} \tag{4.11}$$

where $\mathcal{R}\{\cdot\}$ indicates the real part of the argument. In many cases, the equalizer length $L_w + 1$ is comparable to or shorter than the cyclic prefix length ν. (This is true, for example, in [7] and [67].) In such situations, both noise terms in (4.11) vanish entirely, due to the empty summations. Even if L_w is significantly longer than ν, for typical SNR values σ_n^4 will be very small (compared to the unit variance source signal), so we can neglect the last term in (4.11). Furthermore, the summands in the second term will be both positive and negative, so they will often add to a small value. Combining this with the fact that the second summation is multiplied by the (small) noise variance, we are justified in ignoring the second term in (4.11) as well. This leaves us with $\hat{J}_{SAM}(\nu) \cong J_{SAM}(\nu)$ (and $\hat{J}_{SAM}(\nu) = J_{SAM}(\nu)$ exactly if $L_w \leq \nu$). Accordingly, we will henceforth drop the hat on $J_{SAM}(\nu)$ and ignore the noise terms. The effect of noise on the performance of SAM is investigated in Chapter 8.

Note that the cost function $J_{SAM}(\nu)$ depends only on the output of the TEQ and the choice of ν. Hence, a gradient-descent algorithm over this cost function, with an additional norm constraint on $\mathbf{c}$ or $\mathbf{w}$, requires no knowledge of the source sequence. Such an algorithm will be derived momentarily. Also note that the channel length $L_h + 1$ must be known in order to determine L_c. In ADSL systems, the channel is typically modeled as a length N FIR filter, where $N = 512$ is the

FFT size. The CSA test loops [93] typically have almost all of their energy in 200 consecutive taps, so the FFT size is a very conservative choice for $L_h + 1$ in this application. For other applications, the user must choose a reasonable estimate (or overestimate) for L_h based on typical delay spread measurements for that application. Alternatively, one could maximize the auto-correlation of the received data within the first $\nu + 1$ delays of the auto-correlation function, while maintaining some constraint. This method would not require an estimate of the channel length.

Adaptive Algorithm

The steepest gradient-descent algorithm over the cost surface $J_{SAM}(\nu)$ is

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \mu \nabla_{\mathbf{w}} \left(\sum_{l=\nu+1}^{L_c} \left| \mathrm{E}\left[y^*(k) y(k-l) \right] \right|^2 \right), \tag{4.12}$$

where μ denotes the step size and $\nabla_{\mathbf{w}}$ denotes the gradient with respect to $\mathbf{w}$. To implement this algorithm, an instantaneous cost function can be defined, where the expectation operation is replaced by a moving average over a user-defined window of length N_{av}.

$$J_{SAM}^{inst}(\nu, k) = \sum_{l=\nu+1}^{L_c} \left| \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \frac{y^*(n) y(n-l)}{N_{av}} \right|^2 . \tag{4.13}$$

The value of N_{av} is a design parameter. It should be large enough to give a reliable estimate of the expectation, but no larger, as the algorithm complexity is proportional to N_{av}. (One possible choice for block-based systems is $N_{av} = M$, where M is the total block size. This allows for one update per block, like the

MERRY algorithm.) The "stochastic" gradient-descent algorithm is then given by

$$\begin{aligned}\mathbf{w}(k{+}1) = \; & \mathbf{w}(k) \\ & -\mu \sum_{l=\nu+1}^{L_c} \left[\left\{ \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \frac{y^*(n)y(n-l)}{N_{av}} \right\} \left\{ \nabla_{\mathbf{w}} \left(\sum_{n=kN_{av}}^{(k+1)N_{av}-1} \frac{y(n)y^*(n-l)}{N_{av}} \right) \right\} \right. \\ & \qquad \left. + \left\{ \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \frac{y(n)y^*(n-l)}{N_{av}} \right\} \left\{ \nabla_{\mathbf{w}} \left(\sum_{n=kN_{av}}^{(k+1)N_{av}-1} \frac{y^*(n)y(n-l)}{N_{av}} \right) \right\} \right]\end{aligned} \tag{4.14}$$

which simplifies to

$$\boxed{\begin{aligned}\mathbf{w}(k+1) = \; & \mathbf{w}(k) \\ & -\mu \sum_{l=\nu+1}^{L_c} \left[\left\{ \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \frac{y^*(n)y(n-l)}{N_{av}} \right\} \left\{ \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \left(\frac{y(n)\mathbf{r}^*(n-l)}{N_{av}} \right) \right\} \right. \\ & \qquad \left. + \left\{ \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \frac{y(n)y^*(n-l)}{N_{av}} \right\} \left\{ \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \left(\frac{y(n-l)\mathbf{r}^*(n)}{N_{av}} \right) \right\} \right]\end{aligned}} \tag{4.15}$$

The TEQ update algorithm described in (4.15) will be referred to as the *Sum-squared Auto-correlation Minimization* (SAM) algorithm, as it attempts to minimize the cost function described in (4.5). The algorithm here with complex parameters appeared in [78]. The original SAM algorithm with real parameters [9] is computationally simpler.

An alternate method of implementing the algorithm comes from using auto-regressive (AR) estimates instead of moving average (MA) estimates. Let

$$\begin{aligned}
\mathbf{A}(k) &= (1-\alpha)\mathbf{A}(k-1) + \alpha\; y(k) \begin{bmatrix} r^*(k-\nu-1) \\ \vdots \\ r^*(k-L_c-L_w) \end{bmatrix} \\
\mathbf{B}(k) &= \mathbf{W}^*\mathbf{A}(k) \qquad (4.16) \\
\mathbf{C}(k) &= (1-\alpha)\mathbf{C}(k-1) + \alpha \begin{bmatrix} r^*(k) \\ \vdots \\ r^*(k-L_w) \end{bmatrix} \begin{bmatrix} y(k-\nu-1) \\ \vdots \\ y(k-L_c) \end{bmatrix}^T
\end{aligned}$$

where the "forgetting factor" $0 < \alpha < 1$ is a design parameter and $\mathbf{W}$ is the $(L_c - \nu) \times (L_c + L_w - \nu)$ convolution matrix of the equalizer,

$$\mathbf{W} = \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & 0 & 0 \\ 0 & w_0 & w_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & w_{L_w-1} & w_{L_w} \end{bmatrix}. \qquad (4.17)$$

The exact SAM update rule is

$$\begin{aligned}
\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \sum_{l=\nu+1}^{L_c} \big(&\{\mathrm{E}\,[y^*(k)y(k-l)]\} \cdot \{\mathrm{E}\,[y(k)\mathbf{r}^*(k-l)]\} + \\
&\{\mathrm{E}\,[y(k)y^*(k-l)]\} \cdot \{\mathrm{E}\,[y(k-l)\mathbf{r}^*(k)]\}\big)\,.
\end{aligned} \qquad (4.18)$$

Using the AR estimates, the stochastic update rule can be written as

$$\boxed{\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \sum_{l=\nu+1}^{L_c} \left(\mathbf{B}_{l-\nu}(k) \begin{bmatrix} \mathbf{C}_{1,l-\nu}(k) \\ \vdots \\ \mathbf{C}_{L_w+1,l-\nu}(k) \end{bmatrix} + \mathbf{B}^*_{l-\nu}(k) \begin{bmatrix} \mathbf{A}_{l-\nu}(k) \\ \vdots \\ \mathbf{A}_{l-\nu+L_w}(k) \end{bmatrix} \right)} \qquad (4.19)$$

Again, the algorithm here with complex parameters appeared in [78], and the original SAM algorithm with real parameters [9] is computationally simpler. With both the MA and AR implementations, $\mathbf{w}$ must be periodically renormalized (or else the constraint may be implemented in some other fashion, such as by adding a penalty term onto the cost function). The advantage of the AR implementation is that it allows us to form an update at each time instant, rather than once every N_{av} samples, where N_{av} is the number of samples used in the block averaging of the expectation estimates. The disadvantage is that the estimates now depend more on previous settings of $\mathbf{w}$ rather than the current setting, but if the time variations are reasonably slow, this should not matter. In terms of complexity, the auto-regressive implementation of (4.19) requires approximately $5L_w (L_c - \nu)$ complex multiplications and additions (each) per update, plus a division for renormalization; whereas the moving average implementation of (4.15) requires approximately $4N_{av}L_w (L_c - \nu)$ complex multiplications and additions (each) per update, plus a division for renormalization. Hence the complexity per unit time is approximately the same for the two if (4.15) is implemented only once every N_{av} samples. However, the moving average implementation is useful for analytic purposes.

The choice of α in the AR implementation is analogous to the choice of N_{av} in the MA implementation of (4.15). Both the MA and the AR estimates are unbiased:

$$\begin{aligned}
\mathrm{E}\left[\hat{R}_{yy}^{MA}(l)\right] &= \frac{1}{N_{av}} \sum_{k} \mathrm{E}\left[y^*(k)y(k-l)\right] \\
&= \frac{1}{N_{av}} \cdot N_{av} \cdot R_{yy}(l) = R_{yy}(l), \\
\mathrm{E}\left[\hat{R}_{yy}^{AR}(l)\right] &= \sum_{j=0}^{\infty} \alpha\,(1-\alpha)^j\, \mathrm{E}\left[y^*(k-j)y(k-j-l)\right] \\
&= \alpha \cdot \frac{1}{1-(1-\alpha)} \cdot R_{yy}(l) = R_{yy}(l).
\end{aligned} \tag{4.20}$$

A "fair" comparison of the two approaches should set N and α such that the variances of the two estimates are equal, yet closed-form expressions for the variances of the two estimates are difficult to obtain. An examination of (4.20) suggests that $\alpha = 1/N_{av}$ is a reasonable choice.

As stated earlier, to prevent the algorithm from collapsing the TEQ to an all-zero solution, the equalizer parameters can be normalized after each update to ensure that the norm of the effective response is unity, i.e., $\|\mathbf{c}\|_2^2 = 1$. As the source sequence is assumed to be white, from (4.9) we have

$$E[|y(k)|^2] = \|\mathbf{c}\|_2^2 + \sigma_n^2 \|\mathbf{w}\|_2^2 \approx \|\mathbf{c}\|_2^2 \tag{4.21}$$

so the norm of $\mathbf{c}$ can be approximately determined by monitoring the energy of the output sequence $y(k)$. The approximation does not matter much as it is only used to keep $\|\mathbf{c}\|_2^2$ non-zero, and the actual value of $\|\mathbf{c}\|_2^2$ does not matter. Similarly, if the source is non-white, (4.21) does not hold exactly, but maintaining $E[|y(k)|^2] = 1$ will still keep $\|\mathbf{c}\|_2^2 \neq 0$. A more easily implementable constraint is the unit norm constraint on $\mathbf{w}$, i.e. $\|\mathbf{w}\|_2^2 = 1$. This is easier to implement because we have direct access to $\mathbf{w}$, but not to $\mathbf{c}$.

4.2 Properties of the SAM Cost Function

As is typical of blind equalization algorithms, for instance the constant modulus algorithm (CMA) [44], SAM's cost surface can be expected to be multi-modal. If it has bad local minima, then initialization to ensure convergence to the global minimum becomes important. In general, the SAM cost surface will have local minima. This is a direct result of the following theorem.

Theorem 4.2.1 *The SAM cost function is invariant to the operation* $\mathbf{w} \rightarrow \overline{\mathbf{w}}^*$*, where* $\overline{\mathbf{w}}$ *denotes* $\mathbf{w}$ *with the order of its elements reversed and* * *denotes complex conjugation.*

Proof: Consider the autocorrelation sequences of the combined channels $\mathbf{c}_1 = \mathbf{h} \star \mathbf{w}$ and $\mathbf{c}_2 = \mathbf{h} \star \overline{\mathbf{w}}^*$.

$$\begin{aligned} R_{c_1c_1} &= \mathbf{c}_1^* \star \overline{\mathbf{c}}_1 = (\mathbf{h}^* \star \mathbf{w}^*) \star \overline{(\mathbf{h} \star \mathbf{w})} \\ &= \mathbf{h}^* \star \mathbf{w}^* \star \overline{\mathbf{h}} \star \overline{\mathbf{w}} \\ &= (\mathbf{h} \star \overline{\mathbf{w}}^*)^* \star \left(\overline{\mathbf{h}} \star \mathbf{w}^*\right) \\ &= \mathbf{c}_2^* \star \overline{\mathbf{c}}_2 = R_{c_2c_2}. \end{aligned} \tag{4.22}$$

Since the autocorrelation is invariant to reversing the order of the elements of $\mathbf{w}$ and conjugating them, the SAM cost is also invariant to such a transformation. ■

The upshot of Theorem 4.2.1 is that whenever there is a good minimum of the SAM cost surface, say at $\mathbf{w}_o$, there will also be another minimum at $\overline{\mathbf{w}}_o^*$. There is no reason to expect that the flipped and conjugated $\mathbf{w}_o$ is as good an equalizer as $\mathbf{w}_o$ (in terms of achievable bit rate or bit error rate, for example), so each good minimum may give rise to a bad minimum. Here, "good" and "bad" mean that even though the SAM cost is the same, the ultimate performance metric (achievable bit rate or bit error rate) will not be the same for the two settings. Another consequence is that the SAM cost surface is symmetric with respect to $\mathbf{w} \Leftrightarrow \overline{\mathbf{w}}^*$, so there will be minima, maxima, or saddle points along the subspace $\mathbf{w} = \overline{\mathbf{w}}^*$.

More generally, for a TEQ of order L_w, there will be as many as 2^{L_w} global minima of the SAM cost, corresponding to any combination of flipping a subset of the L_w zeros of the TEQ over the unit circle. Inverting the zero locations of a filter will not change its auto-correlation, so this will not change the value of the SAM cost. With regrards to Theorem 4.2.1, flipping all of a filter's zeros over the unit circle is the same as time-reversing the filter. Again, we can expect minima, maxima, or saddle points along the subspace $\mathbf{w} = \mathcal{T}\left\{\mathbf{w}\right\}$, where $\mathcal{T}\left\{\cdot\right\}$ flips one or more zeros of its argument over the unit circle.

To visualize Theorem 4.2.1, consider the following example. The channel is $\mathbf{h} = [1, 0.3, 0.2]$, the cyclic prefix length is 1 (so we want a 2-tap channel), there is no noise, the equalizer $\mathbf{w}$ has 3 taps, and we use the unit norm constraint $\|\mathbf{w}\| = 1$. With this constraint, the equalizer must lie on a unit sphere, so we can represent the equalizer in spherical coordinates: $w_0 \triangleq w_x = \cos(\theta)\sin(\phi)$, $w_1 \triangleq w_z = \cos(\phi)$, $w_2 \triangleq w_y = \sin(\theta)\sin(\phi)$. In this case, $\mathbf{w} \to \overline{\mathbf{w}}$ is equivalent to switching w_x and w_y (the first and third taps), which is equivalent to reflecting θ over $\frac{\pi}{4}$ or $\frac{5\pi}{4}$; and $\mathbf{w} \to -\mathbf{w}$ is equivalent to the combination of reflecting ϕ over $\frac{\pi}{2}$ and adding π to θ (mod 2π). Since the channel is real, conjugation is ignored.

A contour plot of the SAM cost function is shown in Figure 4.2. The axes represent normalized values of the spherical coordinates θ and ϕ. The contours are logarithmically spaced to show detail in the valleys. There are four minima, but they all have equivalent values of the SAM cost, due to the symmetry relations $\mathbf{w} \Leftrightarrow -\mathbf{w}$ and $\mathbf{w} \Leftrightarrow \overline{\mathbf{w}}$. Note the presence of maxima and saddle points along the subspace $\mathbf{w} = \overline{\mathbf{w}}$, indicated by the dashed line.

We compare the locations of these minima to those of a traditional channel shortening cost function: the shortening SNR (SSNR) [67]. The SSNR is defined as

$$\mathrm{SSNR} = \frac{\mathbf{c}_{win}^H \mathbf{c}_{win}}{\mathbf{c}_{wall}^H \mathbf{c}_{wall}}, \tag{4.23}$$

where $\mathbf{c}_{win}$ is the effective channel impulse response inside the window of interest (of width $\nu + 1$), and $\mathbf{c}_{wall}$ is the effective channel impulse response outside this window. Thus, for our 5-tap effective channel, we pick a 2-tap window and compute the energy of these taps, then divide by the energy of the remaining 3 taps. For each equalizer setting, we will compute the combined channel, pick the 2-tap window

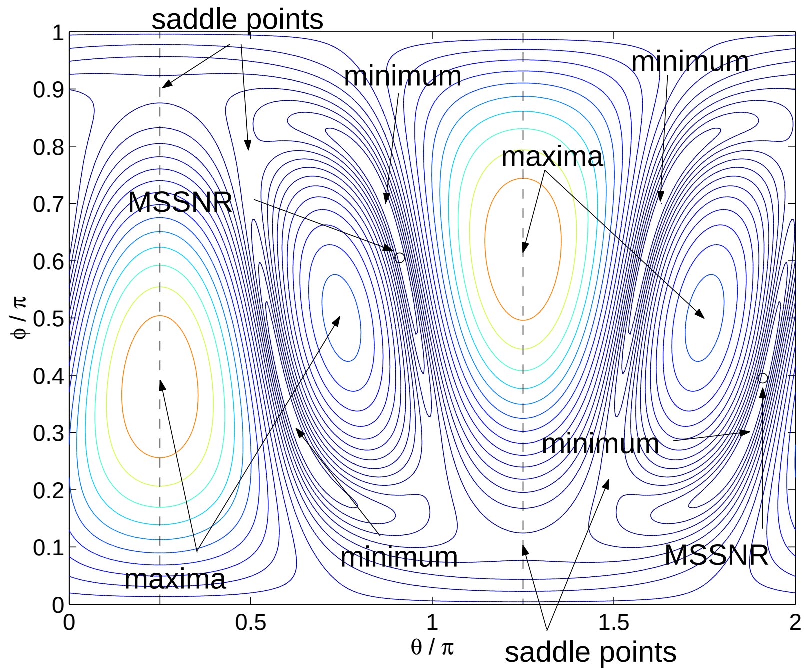

Figure 4.2: Logarithmically spaced contours of the SAM cost function. The two circles are the global minima of the 1/SSNR cost function. The cost function is symmetric about the dashed line.

with the highest SSNR, and then plot the inverse of that value (so that we are looking for minima rather than maxima). Contours of this cost function are shown in Figure 4.3. The four triangles represent the global minima of the SAM cost function. The pair of global minima of 1/SSNR match up nicely with two of the global minima of the SAM cost. Thus, if we find a pair of global minima of the SAM cost, and they have a high value of 1/SSNR, we can fix this by switching to the other global minima of SAM simply by reversing the order of taps in $\mathbf{w}$, or more generally by inverting the locations of the zeros.

For comparison, the two global minima of the MERRY cost function are shown as squares in Figure 4.3. Note that MERRY seeks an effective channel impulse response with $\nu = 1$ tap rather than $\nu + 1 = 2$ taps, hence the MERRY solution

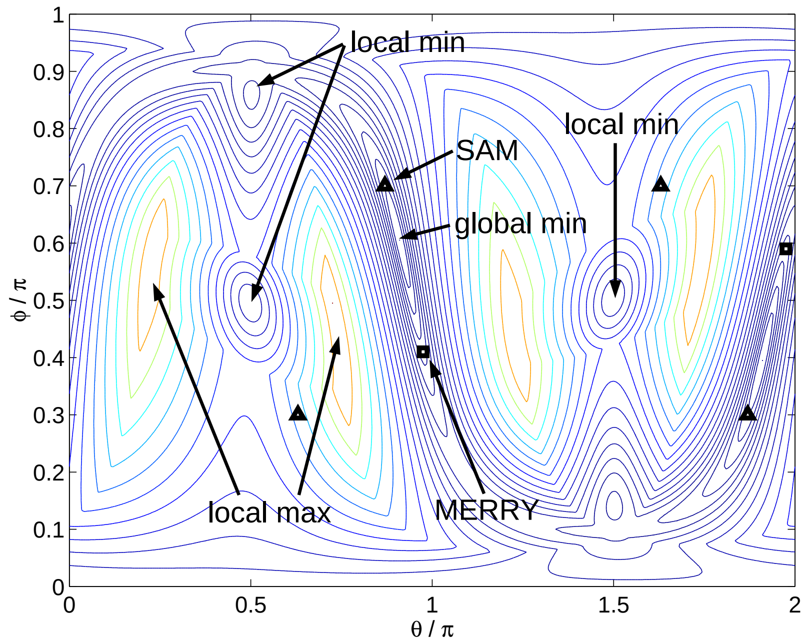

Figure 4.3: Logarithmically spaced contours of the 1/SSNR cost function. The four triangled are the global minima of the SAM cost function, and the two squares are the global minima of the MERRY cost function.

is not expected to be near optimal for such a small CP length. Even so, it lies in the valley of the optimal shortening SNR solution.

4.3 TOLKIEN: a trained version of SAM

SAM attempts to blindly shorten a channel by shortening the auto-correlation of the data at the output of the channel shortener. If training is available, one could instead shorten the cross-correlation of the training data and TEQ output data. This leads to the Trained OFDM L_2-norm Correlation-based Iterative Equalization with Normalization (TOLKIEN) algorithm, which is the subject of this section.

The goal of the maximum shortening SNR TEQ design [67] is to minimize the energy of taps in the effective channel outside of a window of length $\nu + 1$ while maintaining a fixed energy in the remaining taps. As such, we define the following sets for convenience:

$$S_{\Delta}^{win} = \{\Delta, \cdots, \Delta + \nu\} \tag{4.24}$$

$$S_{\Delta}^{wall} = \{0, \cdots, \Delta - 1, \Delta + \nu + 1, \cdots, L_c\} \tag{4.25}$$

$$S = \{0, \cdots, L_c\} = S_{\Delta}^{win} \cup S_{\Delta}^{wall} \tag{4.26}$$

The delay Δ is a design choice that represents the estimate of the symbol placement within the data stream, or equivalently, the placement of the desired window of non-zero taps within the effective channel impulse response.

Observe that an estimate of the channel can be obtained via the cross-correlation

$$c_l = \mathrm{E}\left[y(k)x(k-l)\right], \qquad l \in S. \tag{4.27}$$

Suppressing the out-of-window energy in the effective channel is equivalent to minimizing the cross-correlation squared, summed over the taps of interest, yielding the cost function

$$J_{TOLKIEN}(\Delta) = \sum_{l \in S_{\Delta}^{wall}} \left|\mathrm{E}\left[y(k)x(k-l)\right]\right|^2. \tag{4.28}$$

As the undesirable solution $\mathbf{w} = \mathbf{0}$ yields $J_{TOLKIEN}(\Delta) = 0$, we must impose a constraint. Possible constraints include

A.) $\|\mathbf{w}\| = 1$,

B.) $\|\mathbf{c}\| = 1$,

C.) $w_l = 1$ for some $l \in \{0, \cdots, L_w\}$,

D.) $\|\left[c_{\Delta}, \cdots, c_{\Delta+\nu}\right]^T\| = 1$ (or equivalently, $\|\mathbf{H}_{win}\mathbf{w}\| = 1$).

Any type of norm can be used, although here we use the L_2 norm. The constraint used in the MSSNR solution [67] is constraint D, so it should be used if one desires to converge asymptotically to that solution. However, constraint A can be easily implemented via a periodic renormalization of the filter $\mathbf{w}$ as it adapts.

The gradient of (4.28) is

$$\nabla_{\mathbf{w}} J_{\Delta} = 2 \sum_{l \in S_{\Delta}^{wall}} \mathrm{E}\left[y(k)\ x(k-l)\right] \cdot \mathrm{E}\left[x^*(k-l)\ \mathbf{r}^*(k)\right], \tag{4.29}$$

where $\mathbf{r}(k) = \left[r(k), r(k-1), \cdots, r(k-L_w)\right]^T$. A stochastic gradient descent update can be implemented by using estimates of the expectation terms, for example

A.) $\mathrm{E}\left[y(k)\ x(k-l)\right] \approx y(k)\ x(k-l)$,

B.) $\mathrm{E}\left[y(k)\ x(k-l)\right] \approx \frac{1}{N_{av}} \sum_{j=k-N_{av}+1}^{k} y(j)\ x(j-l)$,

C.) $\mathrm{E}\left[y(k)\ x(k-l)\right] \approx (1-\alpha)\,(\text{previous estimate}) + \alpha\ y(k)\ x(k-l)$.

Estimate A is an instantaneous estimate, B is a moving average (MA), and C is an auto-regressive (AR) estimate. This thesis uses C, since it is cheap to implement and provides a better estimate than estimate A. This leads to the TOLKIEN algorithm:

$$\boxed{\begin{aligned} \mathbf{B}(k) &= (1-\alpha)\mathbf{B}(k-1) + \alpha\left[r(k), \cdots, r(k-L_w)\right]^T \left[x(k), \cdots, x(k-L_c)\right] \\ \mathbf{a}(k) &= \mathbf{w}^T \mathbf{B}(k) \\ \hat{\mathbf{w}}(k+1) &= \mathbf{w}(k) - \mu \sum_{l \in S_{\Delta}^{wall}} \mathbf{a}(l) \mathbf{B}^*(:, l) \\ \mathbf{w}(k+1) &= \frac{\hat{\mathbf{w}}(k+1)}{\|\hat{\mathbf{w}}(k+1)\|} \end{aligned}} \tag{4.30}$$

The TOLKIEN algorithm has a computational complexity similar to the SAM algorithm.

Chapter 5

Frequency-domain Adaptive Equalizers

> "Put away these dispositions which of late transform you from what you rightly are."
>
> – William Shakespeare, *King Lear*, Act I, Scene iv.

Traditional approaches to creating blind, adaptive algorithms rely on the finite alphabet nature of messages sent in digital communication systems. In particular, the constant modulus algorithm (CMA) [44] was designed to equalize signals drawn from symbols all having the same modulus; and decision-directed algorithms rely on explicit knowledge of the constellation from which the transmitted sequence is drawn. In a multicarrier system, the frequency-domain data is generally drawn from a finite alphabet, but the time-domain data is the Fourier transform of the frequency-domain data, and thus no longer has a finite alphabet. It is natural, then, to consider forming frequency-domain cost functions for creating blind, adaptive algorithms for multicarrier receivers.

Section 5.1 discusses several finite-alphabet-based frequency-domain cost functions that can be used to adapt a channel shortener that operates in the time-domain. Section 5.2 investigates these same cost functions, but for use in a per tone equalizer, which is a bank of filters operating in the frequency domain. Section 5.3 examines the unusual topograhy of the per-tone CMA cost function. Section 5.4 compares the cost function of the per-tone LMS algorithm to the MERRY cost function. Sections 5.1–5.3 have been revised from [57].

5.1 TEQ algorithms with frequency-domain cost functions

Since the frequency domain data is expected to be drawn from a QAM constellation, one might consider directly adapting the TEQ to minimize the frequency-domain decisision-directed mean squared error cost,

$$J_{FDD} = \sum_{i=1}^{N} \beta_i \text{ E} \left[\left| Q\left[z_i(k)\right] - z_i(k) \right|^2 \right], \tag{5.1}$$

where the β_i's are weighting coefficients[1], $z_i(k)$ is the output of D_i (the FEQ for tone i) at time k, and $Q[\cdot]$ is the quantizer (nearest constellation point detector). If training is available, then $Q[z_i(k)]$ can be replaced by $X_i(k)$, the transmitted signal on tone i.

Alternatively, one might consider directly adapting the TEQ to minimize the frequency-domain constant modulus cost,

$$J_{FCM} = \sum_{i=1}^{N} \beta_i \text{ E} \left[\left(|z_i(k)|^2 - \gamma_i \right)^2 \right], \tag{5.2}$$

where γ_i, the dispersion constant, can be selected individually for each tone. In [19], the CM cost and related cost functions are considered for adapting the carrier frequency offset (CFO) estimate. This section discusses the merits and pitfalls of using the J_{FCM} and J_{FDD} cost functions for TEQ adaptation.

As an aside, it is of note that if a multicarrier system employs null tones, then $Q[z_i(k)] = 0$ for those tones. If only the set S_{null} of null tones are included (with equal weighting) in the summation in (5.1), then the cost function becomes

$$\begin{aligned} J_{CNA} &= \sum_{i \in S_{null}} \text{ E} \left[|0 - z_i(k)|^2 \right] \\ &= \sum_{i \in S_{null}} \text{ E} \left[|z_i(k)|^2 \right], \end{aligned} \tag{5.3}$$

[1] If there are different numbers of bits on each subcarrier, then the cost can be weighted in favor of the bins with more bits, e.g.

which is the cost function used by the Carrier Nulling Algorithm (CNA) [23], [88].

Using notation similar to [104], the received data $\mathbf{r}$ is obtained from the transmitted data $\mathbf{X}$ via

$$\overbrace{\begin{bmatrix} r(kM+\nu-L_w+1+\Delta) \\ \vdots \\ r((k+1)M+\Delta) \end{bmatrix}}^{\mathbf{r}} = \left[\begin{array}{c|ccc|c} & \overline{\mathbf{h}} & \cdots & 0 & \\ \mathbf{0}_{(1)} & \ddots & & \ddots & \mathbf{0}_{(2)} \\ & 0 & \cdots & \overline{\mathbf{h}} & \end{array}\right] \cdot$$
$$\begin{bmatrix} \mathbf{P}\mathcal{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}\mathcal{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}\mathcal{I}_N \end{bmatrix} \overbrace{\begin{bmatrix} X_{1:N}^{(k-1)} \\ X_{1:N}^{(k)} \\ X_{1:N}^{(k+1)} \end{bmatrix}}^{\mathbf{X}} + \overbrace{\begin{bmatrix} n(kM+\nu-L_w+1+\Delta) \\ \vdots \\ n((k+1)M+\Delta) \end{bmatrix}}^{\mathbf{n}} \tag{5.4}$$

where $\mathbf{h}$ is a row matrix containing the physical channel, $\overline{\mathbf{h}}$ is the time-reversal of $\mathbf{h}$, and $\mathbf{n}$ is additive noise or interference. The effective channel $\mathbf{H}$ includes the physical channel $\mathbf{h}$, the addition of the cyclic prefix (inserted by $\mathbf{P}$), and the IFFT (implemented by $\mathcal{I}_N$); and $\mathbf{X}$ contains the symbol of interest as well as the preceding and succeeding symbols. The matrices $\mathbf{0}_{(1,2)}$ are large zero matrices, the sizes of which are determined by the symbol synchronization parameter Δ.

Assume for the moment that no FEQ is used, so that we may replace the FEQ output for tone i, $z_i(k)$, by the FFT output for tone i, $\hat{z}_i(k)$. Consider a stochastic gradient descent of the CM cost (5.2),

$$\frac{\partial J_{PTCM}}{\partial w_l} = 2\sum_{i=1}^{N} \beta_i \left(|\hat{z}_i(k)|^2 - \gamma_i\right) \cdot \frac{\partial \left(\hat{z}_i(k)\hat{z}_i^*(k)\right)}{\partial w_l}. \tag{5.5}$$

The gradient with respect to a complex vector is defined as

$$\frac{\partial \left(\hat{z}_i(k)\hat{z}_i^*(k)\right)}{\partial w_l} = \frac{\partial \left(\hat{z}_i(k)\hat{z}_i^*(k)\right)}{\partial w_{l,R}} + j\frac{\partial \left(\hat{z}_i(k)\hat{z}_i^*(k)\right)}{\partial w_{l,I}} \tag{5.6}$$

where the subscripts R and I refer to the real and imaginary components, and *

denotes complex conjugation. After a modest amount of algebra, this yields

$$\frac{\partial\left(\hat{z}_i(k)\hat{z}_i^*(k)\right)}{\partial w_l} = 2\ \hat{z}_i \left(\sum_{j=1}^{N} \mathcal{F}_N(i,j)\mathbf{r}(l+j-1)\right)^*, \tag{5.7}$$

where $\mathbf{r}(j)$ means the j^{th} element of the vector $\mathbf{r}$, as defined in (5.4). The resulting stochastic gradient descent algorithm is

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \sum_{i=1}^{N} \beta_i \hat{z}_i \left(|\hat{z}_i(k)|^2 - \gamma_i\right) \Delta_i^*, \tag{5.8}$$

where the N different $\tilde{L}_w$-vectors Δ_i are obtained by a "sliding FFT,"

$$\begin{bmatrix} \Delta_1^T \\ \vdots \\ \Delta_N^T \end{bmatrix} = \mathcal{F}_N \cdot \begin{bmatrix} \mathbf{r}(0) & \cdots & \mathbf{r}(L_w) \\ \vdots & & \vdots \\ \mathbf{r}(N-1) & \cdots & \mathbf{r}(N-1+L_w) \end{bmatrix}. \tag{5.9}$$

This "sliding FFT" can be implemented efficiently, using little more computational power than a single FFT [104].

There are several problems with this approach. First of all, the sliding FFT adds enough complexity to bring this approach on par with a per tone structure (which ultimately has a higher performance bound), so it makes more sense to pursue a per tone equalizer adapting via CMA. More importantly, once the FEQ is added into the mix, there will be a TEQ and an FEQ in series. The FEQ coefficients will be adapting on the CM (or DD) cost function for their respective tones, and the TEQ will be adapting based on the sum of these cost functions. The interaction between the adapting TEQ and the adapting FEQ is phenomenally difficult to analyze. Also, the cost (5.2) will be very large unless all of the FEQs are properly initialized and can successfully track the variation of the channel and the adaptation of the TEQ. Instead, we will favor adaptive per tone equalizers, in which the equalizer for each tone absorbs the FEQ for that tone.

5.2 Per tone LMS, DDLMS, and CMA

Van Acker *et al.* [104] have proposed an alternate equalization structure, called a per tone equalizer (PTEQ), which accomplishes the same task as the TEQ/FEQ structure in Figure 1.1, but with improved performance and comparable complexity. A comparison of the block diagrams for the TEQ/FEQ structure and the per tone equalizer structure is given in Figure 5.2, and the full details of the per tone structure can be found in [104]. In a PTEQ structure, the single FIR filter (the TEQ) is moved after the FFT and replaced by a bank of linear combiners, one for each tone. For any given TEQ, there is a PTEQ that results in the same output, but the converse is not true in general. Thus, the PTEQ is a generalization of the TEQ.

Per tone equalization of bin i is accomplished by forming a linear combination of the i^{th} FFT output and L_v difference terms of the pre-FFT signal, $\mathbf{r}$:

$$z_i = \overline{\mathbf{v}}_i^T \cdot \underbrace{\left[\begin{array}{c|c|c} \mathbf{I}_{L_v} & \mathbf{0} & -\mathbf{I}_{L_v} \\ \hline \mathbf{0} & \multicolumn{2}{c}{\mathcal{F}_N(i,:)} \end{array}\right]}_{\mathbf{F}_i} \cdot \mathbf{r}. \tag{5.10}$$

The linear combiner (not a tapped delay line) $\mathbf{v}_i$ is the equalizer for tone i; $\overline{\mathbf{v}}_i$ is the time-reversal of $\mathbf{v}_i$, defined for convenience; and z_i is the equalized data for tone i. The notation in (5.4) and (5.10) was introduced in [104].

Determination of the per tone equalizer coefficients has been explored in [49], [104], [103], [116], and [117]. In [104], the optimal coefficients (in terms of bit rate) are calculated in a least-squares manner, based on knowledge of the transmission channel, and the signal and noise statistics. In [103], the coefficients are determined in a less computationally-intensive fashion through the use of recursive least-squares (RLS), which requires training throughout the adaptation. In

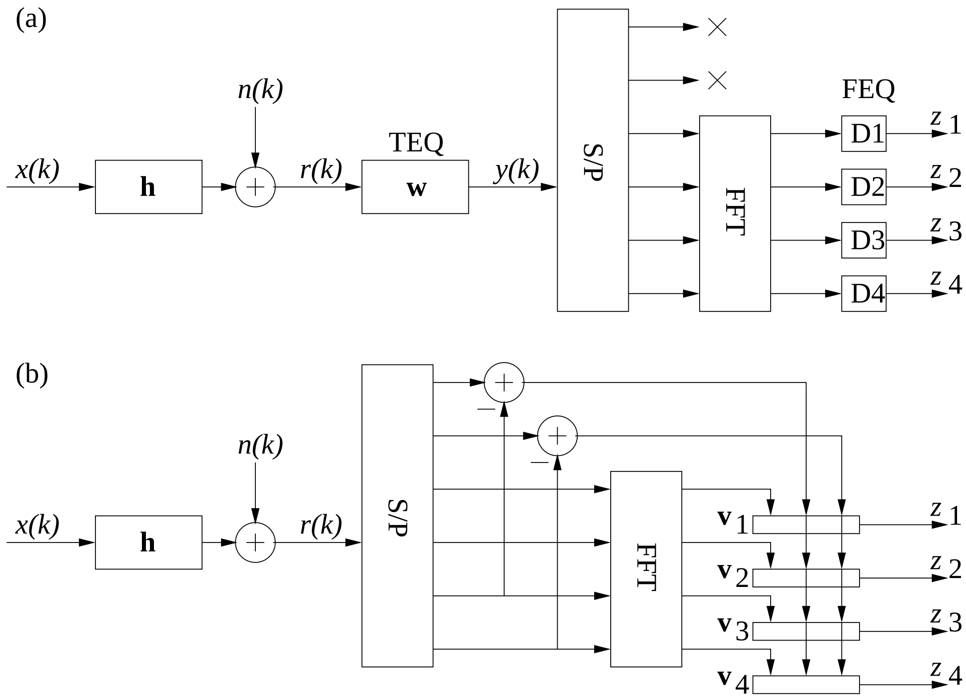

Figure 5.1: Channel model and receiver diagrams for (a) the TEQ/FEQ structure, and (b) the per tone equalizer structure. The transmitted data is $x(k)$ and the received data is $r(k)$.

[117], an RLS-LMS combination was used for adapting the PTEQ. In [116], a time-domain window was designed for use with a PTEQ. The PTEQ was generalized to the MIMO case in [49]. These approaches are well-suited to a system that has plentiful training and computational power. The algorithms discussed in this thesis are designed for situations in which neither condition is true, and in which the environment is modestly time-varying.

Per Tone Decision-Directed LMS (PT-DDLMS)

The PT-DDLMS algorithm is obtained by performing a stochastic gradient descent of the decision-directed cost function for tone i,

$$J_{PTDD,i} = \mathrm{E}\left[\left| Q\left[z_i(k) \right] - z_i(k) \right|^2 \right], \tag{5.11}$$

where $Q[\cdot]$ is the quantization operator (decision device). The resulting algorithm is:

$$\text{PT-DDLMS:} \boxed{\begin{aligned} &\text{For } i = 1, \ldots, N \text{ and } k = 1, 2, 3, \ldots \\ z_i(k) &= \overline{\mathbf{v}}_i^T(k)\ \mathbf{F}_i\ \mathbf{r}(k) \\ e_i(k) &= Q\left[z_i(k) \right] - z_i(k) \\ \overline{\mathbf{v}}_i(k+1) &= \overline{\mathbf{v}}_i(k) + \mu\ e_i(k)\ \mathbf{F}_i^*\ \mathbf{r}^*(k) \end{aligned}} \tag{5.12}$$

Per Tone CMA (PT-CMA)

The constant modulus algorithm (CMA) is a popular alternative to decision-directed algorithms. A detailed review of its convergence behavior in single-carrier systems can be found in [44]. CMA attempts to minimize the dispersion of the equalized symbols by performing a stochastic gradient descent of

$$J_{PTCM,i} = \mathrm{E}\left[\left(\left| z_i(k) \right|^2 - \gamma_i \right)^2 \right] \tag{5.13}$$

for each bin i. The resulting algorithm is:

$$\text{PT-CMA:} \boxed{\begin{aligned} &\text{For } i = 1, \ldots, N \text{ and } k = 1, 2, 3, \ldots \\ z_i(k) &= \overline{\mathbf{v}}_i^T(k)\ \mathbf{F}_i\ \mathbf{r}(k) \\ e_i(k) &= -z_i(k) \cdot \left(\left| z_i(k) \right|^2 - \gamma_i \right) \\ \overline{\mathbf{v}}_i(k+1) &= \overline{\mathbf{v}}_i(k) + \mu\ e_i(k)\ \mathbf{F}_i^*\ \mathbf{r}^*(k) \end{aligned}} \tag{5.14}$$

Structurally, the only difference between CMA and the DD-LMS algorithm in this application and single-carrier equalization is the presence of the $\mathbf{F}_i^*$ matrices. The computation of $\mathbf{F}_i \cdot \mathbf{r}(k)$ is not actually implemented as a matrix-vector multiply. Rather, the last element is obtained directly from the output of the FFT, and the other L_v elements are computed via a single subtraction each (c.f. equation (5.10)).

A point to emphasize is that these single-carrier techniques and others are readily applicable to the per tone structure, but not as easily to the TEQ/FEQ structure, due to the coupling between the TEQ and FEQs. This is because for per tone equalization, there are no separate FEQs.

5.3 Topography of the PT-CM cost surface

This section derives the CM cost function (5.13) as a function of the equalizer parameters $\overline{\mathbf{v}}_i$ and the symbol synchronization parameter Δ, in a fashion similar to that in [44]. Then a low-dimensional example is used to build intuition.

The first step is to decide on appropriate values for the dispersion constants γ_i. Our approach is analogous to the approach taken by Godard [34], i.e. the dispersion constant for tone i will be chosen such that when equalization is achieved, the gradient of the cost function for tone i with respect to the equalizer will be zero. This will lead to dispersion constants that can be independently chosen for each subchannel. The gradient of (5.13) is

$$\nabla_{\overline{\mathbf{v}}_i} J = \mathrm{E}\left[4 z_i \left(|z_i|^2 - \gamma_i\right) \mathbf{F}_i^* \mathbf{r}^*\right]. \tag{5.15}$$

To make this zero, we require that

$$\gamma_i \,\mathrm{E}\left[z_i \mathbf{F}_i^* \mathbf{r}^*\right] = \mathrm{E}\left[z_i \,|z_i|^2\, \mathbf{F}_i^* \mathbf{r}^*\right]. \tag{5.16}$$

When equalization has been achieved, we will have $z_i \cong X_i$. We will use this assumption and (5.4) to get

$$\gamma_i \ \mathbf{F}_i^* \mathbf{H}^* \mathrm{E}\left[X_i \mathbf{X}^*\right] = \mathbf{F}_i^* \mathbf{H}^* \mathrm{E}\left[X_i \left|X_i\right|^2 \mathbf{X}^*\right]. \tag{5.17}$$

Denote $\mathbf{e}_j$ as the vector of all zeros except for a 1 in position j. Assuming that the input data is uncorrelated between symbols and between tones, (5.17) becomes

$$\gamma_i \ \mathbf{F}_i^* \mathbf{H}^* \mathrm{E}\left[\left|X_i\right|^2\right] \mathbf{e}_{N+i} = \mathbf{F}_i^* \mathbf{H}^* \mathrm{E}\left[\left|X_i\right|^4\right] \mathbf{e}_{N+i}. \tag{5.18}$$

Thus, an appropriate choice for γ_i is

$$\boxed{\gamma_i = \frac{\mathrm{E}\left[\left|X_i\right|^4\right]}{\mathrm{E}\left[\left|X_i\right|^2\right]}} \tag{5.19}$$

This is the same as Godard's choice for single carrier CMA, except that we are now free to assign different statistics to each subchannel. This is necessary for transmission schemes that use bit loading (such as DSL), since $\mathrm{E}\left[\left|X_i\right|^4\right]$ will generally vary with the bin number i, even if the power $\mathrm{E}\left[\left|X_i\right|^2\right]$ is held constant.

Now we can discuss the CM cost function. Recall that $\mathbf{r} = \mathbf{HX} + \mathbf{n}$ and $z_i = \overline{\mathbf{v}}_i^T \mathbf{F}_i \mathbf{r}$. Thus,

$$\begin{aligned} z_i &= \left(\mathbf{H}^T \mathbf{F}_i^T \overline{\mathbf{v}}_i\right)^T \mathbf{X} + \left(\mathbf{F}_i^T \overline{\mathbf{v}}_i\right)^T \mathbf{n} \\ &\triangleq \sum_j c_j X_j + \sum_j f_j n_j. \end{aligned} \tag{5.20}$$

In the definitions of $\mathbf{c}$ and $\mathbf{f}$, the subscript i has been dropped for simplicity. We will make use of these definitions momentarily.

We can expand (5.13) to

$$J_{PTCM,i} = \mathrm{E}\left(\left|z_i\right|^4\right) - 2\gamma_i \mathrm{E}\left(\left|z_i\right|^2\right) + \gamma_i^2, \tag{5.21}$$

which requires a calculation of $\mathrm{E}\left(\left|z_i\right|^4\right)$ and $\mathrm{E}\left(\left|z_i\right|^2\right)$. Assuming that the noise and the data are uncorrelated and the noise is stationary,

$$\mathrm{E}\left(\left|z_i\right|^2\right) = \sum_k \mathrm{E}\left[\left|X_k\right|^2\right] \cdot \left|c_k\right|^2 + \sigma_n^2 \sum_k \left|f_k\right|^2. \tag{5.22}$$

Determining $\mathrm{E}\left(|z_i|^4\right)$ is more complicated.

$$\begin{aligned}\mathrm{E}\left(|z_i|^4\right) = \mathrm{E} &\left(\sum_k c_k X_k + \sum_a f_a n_a\right)\left(\sum_l c_l X_l + \sum_b f_b n_b\right) \\ &\left(\sum_{\hat{i}} c_{\hat{i}}^* X_{\hat{i}}^* + \sum_c f_c^* n_c^*\right)\left(\sum_j c_j^* X_j^* + \sum_d f_d^* n_d^*\right).\end{aligned} \tag{5.23}$$

(The hat on the i is used to distinguish the summation index $\hat{i}$ from the tone index i.) This will produce 16 cross-terms. The first cross-term corresponds to the noiseless case, and the last term will have a similar structure. The 8 cross-terms with an odd number of noise factors drop out. Of the remaining 6 terms, two are such that the signal and noise are paired with their unconjugated counterparts, so they also drop out (assuming that the source is QAM); and the remaining four of these terms are identical. After an extensive amount of algebra, we arrive at the following general form of the CM cost function,

$$\begin{aligned}J_{PTCM,i} = &\sum_k \left[\mathrm{E}\left(|X_k|^4\right) - 2\mathrm{E}^2\left(|X_k|^2\right)\right] \cdot |c_k|^4 \\ &+ 2\left[\sum_k \mathrm{E}\left(|X_k|^2\right) \cdot |c_k|^2\right]^2 - 2\frac{\mathrm{E}\left[|X_i|^4\right]}{\mathrm{E}\left[|X_i|^2\right]} \sum_k \mathrm{E}\left(|X_k|^2\right) \cdot |c_k|^2 \\ &+ \left(\frac{\mathrm{E}\left[|X_i|^4\right]}{\mathrm{E}\left[|X_i|^2\right]}\right)^2 + 4\left(\sum_k \mathrm{E}\left[|X_k|^2\right] \cdot |c_k|^2\right)\left(\sigma_n^2 \sum_k \mathrm{E}\,|f_k|^2\right) \\ &+ \left[\mathrm{E}\left(|n_k|^4\right) - 2\sigma_n^4\right] \sum_k |f_k|^4 + 2\left[\sigma_n^2 \sum_k \mathrm{E}\,|f_k|^2\right]^2 \\ &- 2\frac{\mathrm{E}\left[|X_i|^4\right]}{\mathrm{E}\left[|X_i|^2\right]} \sigma_n^2 \sum_k \mathrm{E}\,|f_k|^2\end{aligned} \tag{5.24}$$

The subscripts on X are all tone indices, and the subscripts on c and f are tap indices. If we were to remove these tone subscripts on X, this would simplify (5.24) to the result in equation (61) in the appendix of [44]. Alternatively, (5.24) could be simplified by assuming that at least the power $\mathrm{E}\left[|X_k|^2\right]$ does not vary with the bin index k. That is valid in most applications. However, [3] and [18] have shown that if the transmit power is optimized across the bins, then slight performance gains can be achieved, so that assumption does entail a loss of generality.

The most important conclusion we can reach from our analysis of the CMA cost function is that its similarity with the traditional CM cost function in [44] suggests a similarity of behavior. Furthermore, the inputs to the IFFT are generally white (across both symbols and frequency bins), which is analogous to the assumption made in most single-carrier CMA papers, in which the source symbols are assumed to be white. Assuming that the appropriate assumptions are verified for the multicarrier case (corresponding to the assumptions made in the single carrier case), the rich literature for single-carrier CMA can be applied here. In particular, we cannot obtain a closed form solution for the locations of the global minima.

In order to view this cost function, we consider low-order examples. Figure 5.3 shows the CM cost function (and the amalgamated MSE, which is a composition of the MSE's for different delays [44]) for tone 2. The plots for tone 1 are similar. The variables were: a 3-tap channel, a cyclic prefix length of $\nu = 1$, and a 2-tap equalizer on each of the 2 tones. The three plots represent different values of the symbol synchronization, and the axes on each of the plots represent the equalizer taps for tone 2. The CM cost function is periodic in Δ, with period $N + \nu = 3$ in this case (hence only three plots are needed).

Figure 5.3 provides intuition regarding the effects of the symbol synchronization. It is clear that the parameter Δ drastically changes the shape of the cost surface, the depth of the minima, and even the number of minima. For this reason, it is expected that the performance of per tone CMA will vary significantly with Δ, so symbol synchronization must be done with care. However, if $N = 8196$, as in the European HDTV standard [40], there might be a more gradual transition

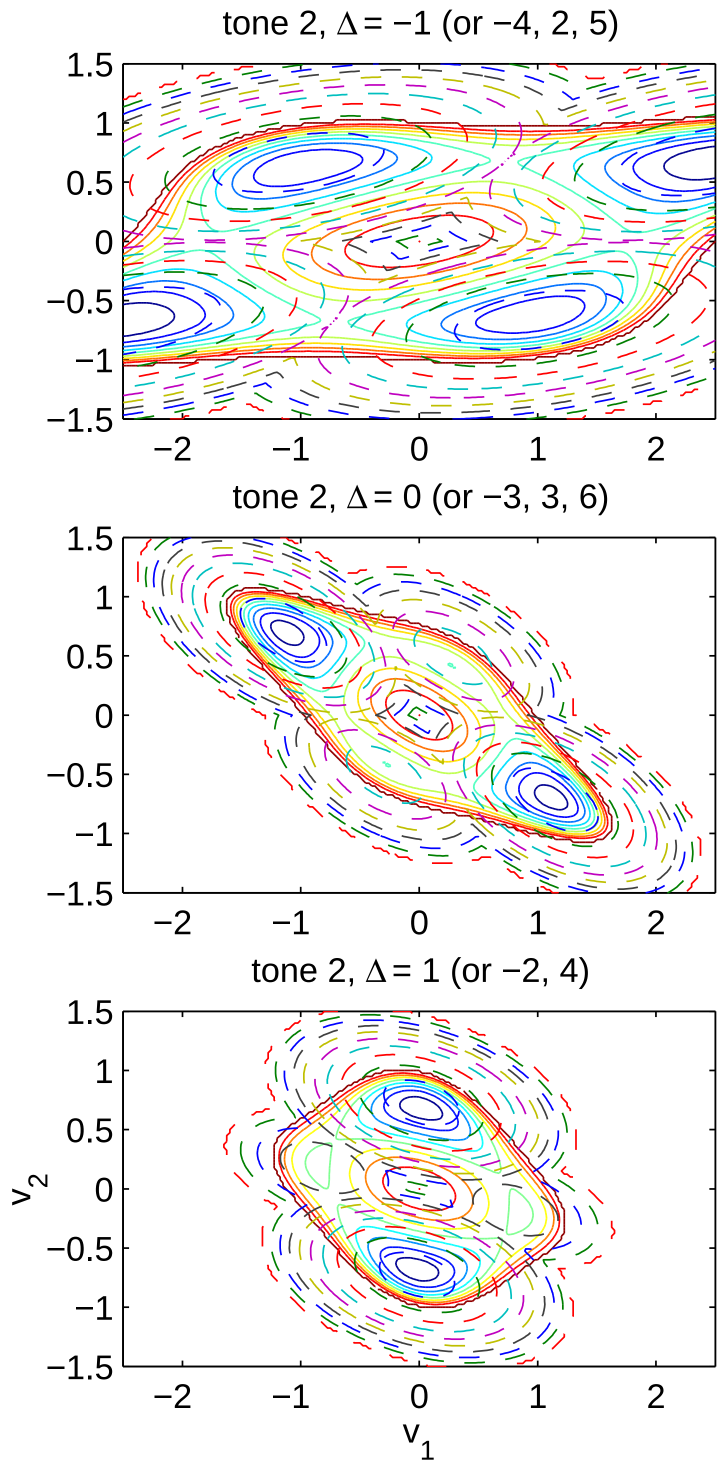

Figure 5.2: The CM (solid) and amalgamated MSE (dashed) cost functions as a function of the equalizer taps for tone 2 (v_1, v_2) and the delay (Δ).

between the cost surfaces as Δ varies, and we must be cautious when generalizing from such a low order example.

5.4 Comparing PT-LMS and MERRY

There are interesting parallels between the cost function of an LMS-based per tone equalizer for tone i with $L_v + 1$ taps and the combined cost functions of a MERRY-based TEQ with L_v taps and the i^{th} 1-tap FEQ, updated via LMS[2]. Consider updating the PTEQ with LMS. Recall that the output of $\mathbf{v}_i$, the PTEQ for tone i, is

$$z_i = \overline{\mathbf{v}}_i^T \cdot \underbrace{\left[\begin{array}{c|c|c} \mathbf{I}_{L_v} & \mathbf{0}^T & -\mathbf{I}_{L_v} \\ \hline \mathbf{0}^T & \multicolumn{2}{c}{\mathcal{F}_N(i,:)} \end{array}\right]}_{\mathbf{F}_i} \cdot \mathbf{r}(k) \tag{5.25}$$

where $\mathcal{F}_N(i,:)$ is the i^{th} row of the N-point DFT matrix, $\mathbf{r}(k)$ is a vector of received data samples, and $\overline{\mathbf{v}}_i$ is the time-reversed version of $\mathbf{v}_i$ [104]. The error for the LMS update is

$$\begin{aligned} e_i^{PTEQ}(k) &= X_i(k) - z_i(k) \\ &= X_i(k) - \overline{\mathbf{v}}_i^T \, \mathbf{F}_i \, \mathbf{r}(k). \end{aligned} \tag{5.26}$$

To compare to the MERRY cost function, first partition the i^{th} PTEQ into two segments,

$$\overline{\mathbf{v}}_i(k) = \left[\overline{\mathbf{w}}^T(k),\ D_i(k)\right]^T, \tag{5.27}$$

where $\overline{\mathbf{w}}(k)$ has L_v taps and $D_i(k)$ is a scalar. Defining

$$\mathbf{G} = \left[\begin{array}{c|c|c} \mathbf{I}_{L_v} & \mathbf{0}_{L_v\times(N-L_v)} & -\mathbf{I}_{L_v}, \end{array}\right] \tag{5.28}$$

[2]The updates can be made blind via decision direction without significantly changing the dynamics of the updates, but for simplicity of notation we consider LMS updates.

then the error becomes

$$
\begin{aligned}
e_i^{PTEQ}(k) &= X_i(k) - \left[\overline{\mathbf{w}}^T(k), D_i(k)\right] \left[\begin{array}{c|c} \multicolumn{2}{c}{\mathbf{G}} \\ \hline \mathbf{0}^T, & \mathcal{F}_N(i,:) \end{array}\right] \cdot \mathbf{r}(k) \\
&= X_i(k) - \overline{\mathbf{w}}^T \mathbf{G}\ \mathbf{r}(k) - D_i(k)\left[\mathbf{0}^T,\ \mathcal{F}_N(i,:)\right]\mathbf{r}(k) \\
&= \left(X_i(k) - D_i(k)\hat{z}_i(k)\right) - \overline{\mathbf{w}}^T \mathbf{G}\ \mathbf{r}(k),
\end{aligned} \tag{5.29}
$$

where $\hat{z}_i(k)$ is the i^{th} FFT output for symbol k. Now note that the error in the MERRY update equation can be written as

$$
\begin{aligned}
e^{MERRY}(k) &= y(kM + \nu + \Delta) - y((k+1)M + \Delta) \\
&= \overline{\mathbf{w}}^T \left[\mathbf{r}(kM + \nu + \Delta) - \mathbf{r}(kM + \nu + N + \Delta)\right] \\
&= \overline{\mathbf{w}}^T\ \mathbf{G}\ \mathbf{r}(kM + \nu + N + \Delta).
\end{aligned} \tag{5.30}
$$

Also note that in the absence of a TEQ, the error for the LMS update of the FEQ for tone i is the input minus the FEQ times the FFT output,

$$
e_i^{FEQ} = X_i(k) - D_i(k)\hat{z}_i(k). \tag{5.31}
$$

(The absence of a TEQ simply means that $\hat{z}_i(k)$ is the i^{th} FFT coefficient of the received data rather than of the TEQ output.) Thus, from (5.29),

$$
\boxed{e_i^{PTEQ}(k) = e_i^{FEQ}(k) - e^{MERRY}(k)} \tag{5.32}
$$

(The odd-looking minus sign can be negated by a redefinition of the MERRY error, if desired.) In other words, if we adapt the TEQ and the FEQ for tone i over the mean square of the PTEQ error of (5.32), then the amalgamation of the L_v-tap TEQ and the 1-tap FEQ will be identical to the $(L_v + 1)$-tap PTEQ adapting over the mean square PTEQ error for tone i. This is not to suggest actually modifying the MERRY update in practice, but rather this suggests a similarity in the behavior of a MERRY-based TEQ and an LMS-based PTEQ.

Since the errors and corresponding updates are related, then when one algorithm suffers from slow convergence, the other may as well. However, the actual MERRY update evolves over the square of the second error term in (5.32) rather than the square of the total error function of (5.32), so the analogy is only a rough approximation.

Chapter 6

Symmetric Filters and Frequency Nulls

"What immortal hand or eye dare frame thy fearful symmetry?"

– William Blake, *The Tiger.*

This chapter[1] discusses several properties of MMSE and MSSNR TEQ designs that can prove useful in parameter selection and complexity reduction. Specifically, we consider symmetry of the TEQ and TIR impulse responses, as well as the locations of the roots of the TIR impulse response. These properties lead to design guidelines and methods for reducing the computational complexity of the MMSE and MSSNR TEQ design families, which include the MERRY algorithm. Since this chapter deals heavily with the MMSE and MSSNR designs, it is assumed that the reader is familiar with the relevant background material in Chapter 2.

6.1 Symmetry in eigenvectors

This section reviews well-known properties of the eigenvectors of symmetric centrosymmetric matrices, and extends these properties to the case of generalized eigenvectors of two such matrices. The following sections show the implications of these properties on TEQ design.

Let $\mathbf{J}$ be the square matrix with ones on the cross diagonal, and zeros elsewhere. Left-multiplication by $\mathbf{J}$ reverses the order of the rows of a matrix and right-multiplication by $\mathbf{J}$ reverses the order of the columns. Symmetric centrosymmetric

[1]Some material in this chapter has been previously published. © 2004 IEEE. Reprinted, with permission, from [56], [60], and [61]. © 2004 Hindawi Publishing Corporation. Reprinted, with permission, from [55].

matrices[2] are defined as matrices in the set

$$\mathcal{C} = \{\mathbf{C}: \quad \mathbf{C}^T = \mathbf{C}, \quad \mathbf{JCJ} = \mathbf{C}\}. \tag{6.1}$$

Symmetric centrosymmetric matrices of size $L \times L$ have exactly $\lceil L/2 \rceil$ symmetric eigenvectors and $\lfloor L/2 \rfloor$ skew-symmetric eigenvectors [12]. This property can be applied directly to the MSSNR solution for $\mathbf{w}$ with a unit norm TEQ (UNT) constraint [55], and we will extend it to the generalized eigenvector MSSNR solution.

The results in [12] were developed for real matrices. To generalize their results, we define the set of *Hermitian centro-Hermitian matrices*

$$\mathcal{C}^* = \{\mathbf{C}: \quad \mathbf{C}^H = \mathbf{C}, \quad \mathbf{JCJ} = \mathbf{C}^*\}. \tag{6.2}$$

These matrices have the properties that (i) they are Hermitian, (ii) they are persymmetric (symmetric about the cross diagonal), and (iii) rotating the matrix by 180^* is equivalent to conjugating the matrix. Any two of these properties together imply the third.

Consider the (tall) Toeplitz channel convolution matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} h_0, & h_1, & h_2, & \cdots, & 0, & 0, & 0 \\ 0, & h_0, & h_1, & \cdots, & 0, & 0, & 0 \\ \ddots, & \ddots, & \ddots, & \ddots, & \ddots, & \ddots, & \ddots \\ 0, & 0, & 0, & \cdots, & h_{L_h-2}, & h_{L_h-1}, & h_{L_h} \end{bmatrix}^T. \tag{6.3}$$

Note that $\mathbf{H}^H\mathbf{H}$ is Hermitian and Toeplitz. Since Toeplitz implies persymmetric, we have $\mathbf{H}^H\mathbf{H} \in \mathcal{C}^*$. The MSSNR design is a generalized eigenvector of the matrices $\mathbf{A} = \mathbf{H}_{wall}^H\mathbf{H}_{wall}$ and $\mathbf{B} = \mathbf{H}_{win}^H\mathbf{H}_{win}$, where $\mathbf{H}_{win}$ and $\mathbf{H}_{wall}$ partition $\mathbf{H}$. This

[2] a.k.a. symmetric persymmetric matrices or doubly symmetric matrices, which are symmetric about the main diagonal and the cross diagonal, and are unchanged by rotation by 180^o.

partitioning and similarity of structure suggests that $\mathbf{A}$ and $\mathbf{B}$ may also be in $\mathcal{C}^*$. Unfortunately, $\mathbf{A}$ and $\mathbf{B}$ are not perfectly Hermitian centro-Hermitian, but they are nearly so. Specifically, they are Hermitian and nearly Toeplitz, facts exploited in [111]. For the moment, assume that $\mathbf{A}$ and $\mathbf{B}$ are approximately centro-Hermitian. Numerical results that quantify this are presented later in this section.

The TEQ obtained by the MSSNR-UNT solution is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}$. Since $\mathbf{A}$ has been observed to be nearly centrosymmetric, its eigenvectors should be approximately symmetric or skew-symmetric, and indeed that is the case. For the MSSNR and MMSE solutions, we must consider the generalized eigenvectors of $(\mathbf{B}, \mathbf{A})$, where $\mathbf{B} = \mathbf{H}_{win}^H \mathbf{H}_{win}$, and where $\mathbf{A} = \mathbf{H}_{wall}^H \mathbf{H}_{wall}$ for the MSSNR design and $\mathbf{A} = \mathbf{H}_{wall}^H \mathbf{H}_{wall} + \mathbf{R}_n$ for the MMSE design. However, if $\mathbf{A}$ or $\mathbf{B}$ is invertible, then the generalized eigenvalue problem can be reduced to a traditional eigenvalue problem [109]. In some practical cases, $\mathbf{B}$ has been observed to be invertible [67]. However, when $L_w > \nu$, the matrix $\mathbf{H}_{win}$ cannot have full column rank, and hence $\mathbf{B}$ will not be invertible [113]. Furthermore, even when $L_w \leq \nu$, $\mathbf{B}$ may be singular. Fortunately, as will be proved in Theorem 6.1.1, $\mathbf{A}$ is invertible for all channels longer than the CP.

Theorem 6.1.1 *For an FIR channel* $\mathbf{h}$*, if* $L_h > \nu$*, then the matrices* $\mathbf{A}_1 = \mathbf{H}_{wall}^H \mathbf{H}_{wall}$ *and* $\mathbf{A}_2 = \mathbf{H}_{wall}^H \mathbf{H}_{wall} + \mathbf{R}_n$ *will be invertible.*

Proof: For ease of notation, we assume that the first and last taps in the channel $\mathbf{h}$ are non-zero. This entails no loss of generality, since leading zeros can be separated into a bulk delay, and trailing zeros can be omitted. Thus, if the FIR channel has length $L_h + 1$, then its transfer function will have exactly L_h zeros. Convolving the channel with an FIR TEQ of length $L_w + 1$ will add L_w zeros. Thus, the effective channel impulse response will have $L_h + L_w > \nu$ zeros, since $L_h > \nu$ and $L_w \geq 0$.

Assume that $\mathbf{A}_1$ is singular. Then there exists $\mathbf{w}_{null} \neq \mathbf{0}$ such that $\mathbf{A}_1\mathbf{w}_{null} = \mathbf{0}$. Then

$$\|\mathbf{c}_{wall}\|_2^2 = \mathbf{c}_{wall}^H \mathbf{c}_{wall} = \mathbf{w}_{null}^H \mathbf{A}_1 \mathbf{w}_{null} = 0. \tag{6.4}$$

This implies that there exists a TEQ $\mathbf{w}_{null}$ that forces $\mathbf{c}_{wall} = \mathbf{0}$, which in turn implies that the channel has been shortened to $\nu+1$ taps or less. This requires that the convolution of $\mathbf{h}$ and $\mathbf{w}_{null}$ has no more than ν zeros. However, we have already shown that the effective channel will have more than ν zeros. This contradiction invalidates the assumption that $\mathbf{A}_1$ is singular. Hence, $\mathbf{A}_1$ is invertible when $L_h > \nu$.

Since $\mathbf{A}_1 = \mathbf{H}_{wall}^H \mathbf{H}_{wall}$ is invertible for $L_h > \nu$, we have $\mathbf{w}^H \mathbf{A}_1 \mathbf{w} > 0$ for all $\mathbf{w} \neq \mathbf{0}$. Hence

$$\begin{aligned} \mathbf{w}^H \left(\mathbf{A}_1 + \mathbf{R}_n\right) \mathbf{w} &= \mathbf{w}^H \mathbf{A}_1 \mathbf{w} + \mathbf{w}^H \mathbf{R}_n \mathbf{w} \\ &\geq \mathbf{w}^H \mathbf{A}_1 \mathbf{w} > 0, \end{aligned} \tag{6.5}$$

where we have made use of the fact that $\mathbf{R}_n$ is positive semi-definite. By (6.5), $\mathbf{A}_2 = \mathbf{A}_1 + \mathbf{R}_n$ is invertible when $L_h > \nu$. ■

(A more visual proof is possible by looking at the number of pivots in the $\mathbf{H}_{wall}$ matrix, but that proof involves a large number of special cases regarding the relative values of ν, L_h, L_w, and Δ.)

The MSSNR and MMSE TEQs must satisfy the generalized eigenvalue equation

$$\mathbf{B}\mathbf{w} = \lambda \mathbf{A}\mathbf{w}. \tag{6.6}$$

Since $\mathbf{A}$ is invertible by Theorem 6.1.1, we can convert this into a traditional eigenvalue problem,

$$\left(\mathbf{A}^{-1}\mathbf{B}\right)\mathbf{w} = \lambda \mathbf{w}. \tag{6.7}$$

$\mathbf{A}$ and $\mathbf{B}$ are symmetric and approximately centrosymmetric, so $\mathbf{JAJ} \approx \mathbf{A}^*$ and $\mathbf{JBJ} \approx \mathbf{B}^*$. The inverse of a centro-Hermitian matrix is also centro-Hermitian (see [36] for the real case), and the product of centro-Hermitian matrices is centro-Hermitian (see [10] for the real case), so $(\mathbf{A}^{-1}\mathbf{B})$ is approximately centro-Hermitian. Unfortunately, $(\mathbf{A}^{-1}\mathbf{B})$ may not be *Hermitian*, even though $\mathbf{A}^{-1}$ and $\mathbf{B}$ are, so the full range of results in [12] cannot be immediately applied. Even so, as will be shown in Theorem 6.1.2, the eigenvectors of a centro-Hermitian matrix $(\mathbf{A}^{-1}\mathbf{B})$ can always be chosen to be conjugate symmetric or conjugate skew-symmetric.

Theorem 6.1.2 *If $\mathbf{A}, \mathbf{B} \in \mathcal{C}^*$ (so they are Hermitian centro-Hermitian) and $\mathbf{A}$ is invertible, then the eigenvectors of $(\mathbf{A}^{-1}\mathbf{B})$ can always be chosen to be conjugate symmetric or conjugate skew-symmetric. Furthermore, if the eigenvalues of $(\mathbf{A}^{-1}\mathbf{B})$ are distinct, then the eigenvectors will all be conjugate symmetric or conjugate skew-symmetric.*

Proof: Since $(\mathbf{A}^{-1}\mathbf{B})$ is centro-Hermitian, $\mathbf{J}\left(\mathbf{A}^{-1}\mathbf{B}\right)\mathbf{J} = \left(\mathbf{A}^{-1}\mathbf{B}\right)^*$. Let $\mathbf{w}$ be an eigenvector of $\mathbf{A}^{-1}\mathbf{B}$, or equivalently a generalized eigenvector of $(\mathbf{B}, \mathbf{A})$. Since $\mathbf{A}$ and $\mathbf{B}$ are Hermitian, the associated eigenvalue λ will be real. Thus, $\mathbf{w}$ satisfies

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{B}\,\mathbf{w} &= \lambda\,\mathbf{w}, \\ \left(\mathbf{A}^{-1}\mathbf{B}\right)^*\ \mathbf{w}^* &= \lambda\,\mathbf{w}^*, \\ \left(\mathbf{J}\mathbf{A}^{-1}\mathbf{B}\mathbf{J}\right)\ \mathbf{w}^* &= \lambda\,\mathbf{w}^*, \\ \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{J}\mathbf{w}^*\right) &= \lambda\ \left(\mathbf{J}\mathbf{w}^*\right), \\ \mathbf{A}^{-1}\mathbf{B}\left(-\mathbf{J}\mathbf{w}^*\right) &= \lambda\ \left(-\mathbf{J}\mathbf{w}^*\right) \end{aligned} \tag{6.8}$$

where we have made use of $\mathbf{JJ} = \mathbf{I}$. Thus, if $\mathbf{w}$ is an eigenvector of $(\mathbf{A}^{-1}\mathbf{B})$ with eigenvalue λ, then $\mathbf{J}\mathbf{w}^*$ and $-\mathbf{J}\mathbf{w}^*$ are also eigenvectors with the same eigenvalue λ. As a consequence, for a given eigenpair $(\lambda, \mathbf{w})$, we can always force the eigenvector to be conjugate symmetric, $\mathbf{w}_{sym} = \frac{1}{2}(\mathbf{w} + \mathbf{J}\mathbf{w}^*)$, or conjugate skew-symmetric, $\mathbf{w}_{skew} = \frac{1}{2}(\mathbf{w} - \mathbf{J}\mathbf{w}^*)$, without changing the eigenvalue. Note that if $\mathbf{w}$ itself is already conjugate symmetric (skew-symmetric), then we cannot make it into a conjugate skew-symmetric (symmetric) vector by this procedure because the result yields $\mathbf{w}_{skew} = \mathbf{0}$ ($\mathbf{w}_{sym} = \mathbf{0}$).

If all of the eigenvalues of $\mathbf{A}^{-1}\mathbf{B}$ are distinct, then its eigenvectors are unique. Thus, $\mathbf{w}$, $\mathbf{J}\mathbf{w}^*$, and $-\mathbf{J}\mathbf{w}^*$ must all be identical (up to a scalar, e.g. ± 1). The only way for this to be satisfied is if each $\mathbf{w}$ is either conjugate symmetric or conjugate skew-symmetric. ■

Theorem 6.1.2 has the condition that $\mathbf{A} \in \mathcal{C}^*$ and $\mathbf{B} \in \mathcal{C}^*$. In general, this is only approximately true. Thus, the eigenvectors of $\mathbf{A}$ and of $\mathbf{A}^{-1}\mathbf{B}$ will all be *approximately* conjugate symmetric or conjugate skew-symmetric. Oddly enough, the MSSNR and MSSNR-UNT TEQs always seem to be nearly symmetric rather than nearly skew-symmetric, and the point of symmetry need not be the center of the TEQ, as can be seen by the example MSSNR TEQ in Figure 6.1. One possible reason for this is that in some special cases (e.g. tridiagonal matrices), the eigenvectors alternate between symmetric and skew-symmetric as the eigenvalues decrease, and the eigenvectors corresponding to extreme eigenvalues will be symmetric [12]. It is difficult to prove that this is always the case here, but something of this sort may be occuring.

Figure 6.1 shows a typical MSSNR TEQ with real coefficients, as well as its symmetric part and a skew-symmetric perturbation. The symmetric part was obtained by considering all possible points of symmetry, and choosing the one for which the norm of the symmetric part divided by the norm of the perturbation was maximized. For example, if the TEQ were $\mathbf{w} = [1, 2, 4, 2.2]$, then $\mathbf{w}_{sym} = [0, 2.1, 4, 2.1]$ and $\mathbf{w}_{skew} = [1, -0.1, 0, 0.1]$. In Figure 6.1, most of the energy is

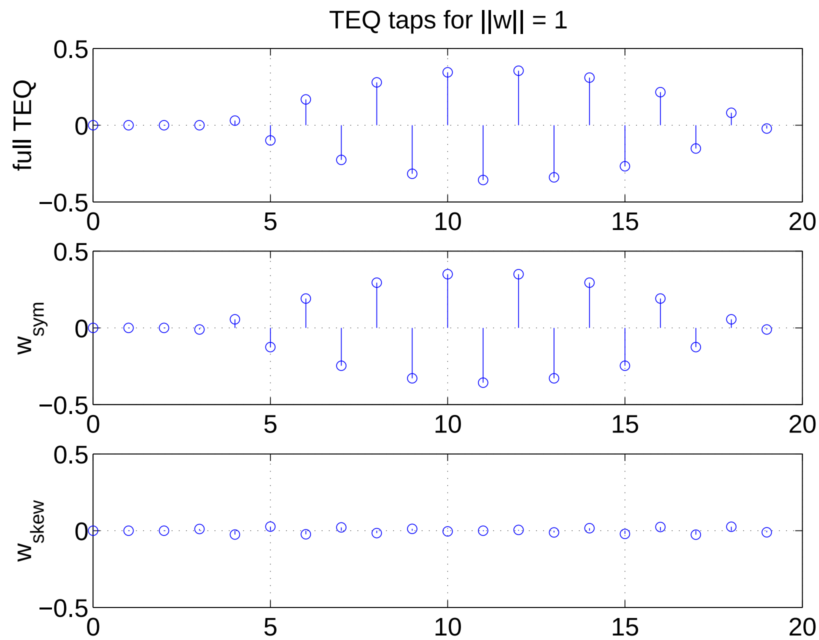

Figure 6.1: Top: MSSNR TEQ, middle: symmetric part, bottom: skew-symmetric part. The channel was CSA test loop 1, the CP length was $\nu = 32$, and the TEQ had 20 taps.

in the symmetric part (middle plot), and there is a small skew-symmetric part (bottom plot).

To quantify the symmetry of the MSSNR and MSSNR-UNT TEQ designs for various parameter values, we computed both TEQs for $1 \leq \nu \leq 120$ and $3 \leq \tilde{L}_w \leq 40$, for the eight carrier serving area (CSA) test loops [7], available at [6]. The CSA loops are real channel models that are commonly used in ADSL simulations. For each TEQ, we decomposed $\mathbf{w}$ into $\mathbf{w}_{sym}$ and $\mathbf{w}_{skew}$, then computed the measure of asymmetry

$$\mu_{asym}(\mathbf{w}) = \frac{||\mathbf{w}_{skew}||^2}{||\mathbf{w}_{sym}||^2}. \tag{6.9}$$

A mesh plot of this ratio for the MSSNR TEQ is shown in Figure 6.2, and a similar plot for the MSSNR-UNT TEQ is shown in Figure 6.3. A cross-section of these

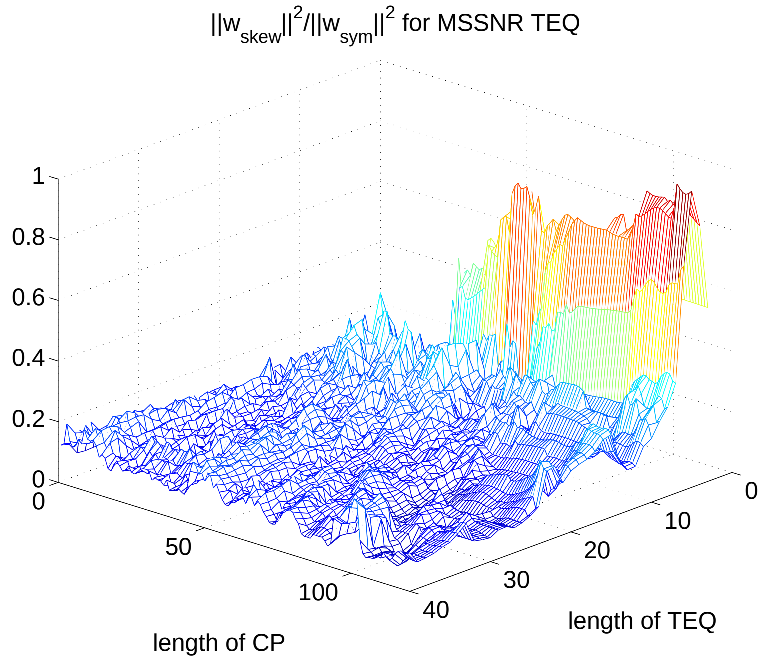

Figure 6.2: Energy in the skew-symmetric part of the TEQ over the energy in the symmetric part of the TEQ, for the MSSNR solution, [67]. The data was optimized for transmission delay Δ and averaged over CSA test loops 1-8.

plots for $\nu = 32$ (the value used in downstream ADSL) is shown in Figure 6.4. The value of Δ was determined via a global search for the MSSNR solution, and the same Δ was used for each corresponding MSSNR-UNT solution. The ratios were computed for CSA test loops 1 through 8 and then averaged.

The MSSNR-UNT TEQ (Figure 6.3) becomes increasingly symmetric for large CP and TEQ lengths, whereas the MSSNR TEQ (Figure 6.2) is highly symmetric for all parameter values, but does not display any significant trends. For parameter values that lead to highly symmetric TEQs, the TEQ can be initialized by only computing half of the TEQ coefficients. For MSSNR, MSSNR-UNT, and MMSE solutions, this effectively reduces the problem from finding an eigenvector (or generalized eigenvector) of an $\tilde{L}_w \times \tilde{L}_w$ matrix to finding an eigenvector (or gen-

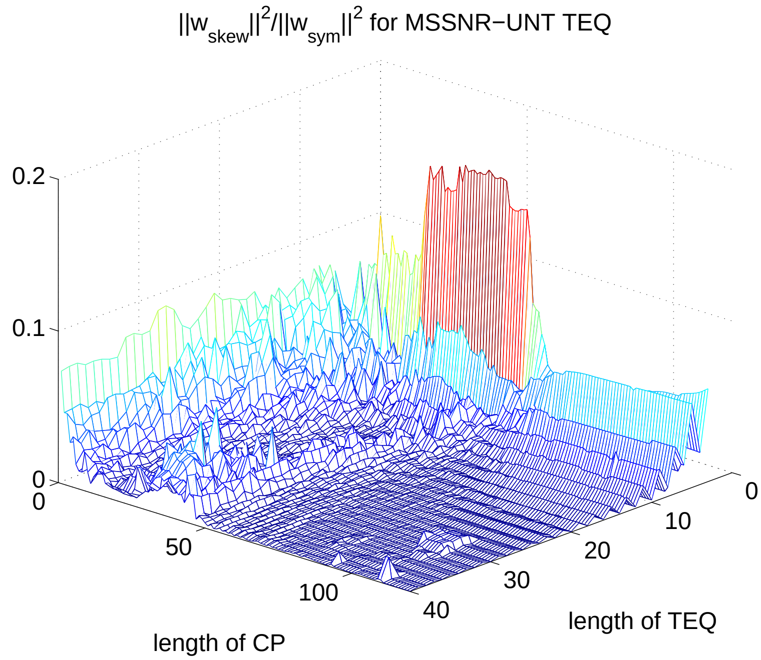

Figure 6.3: Energy in the skew-symmetric part of the TEQ over the energy in the symmetric part of the TEQ, for the MSSNR solution with unit norm TEQ constraint [55]. The data was optimized for transmission delay Δ and averaged over CSA test loops 1-8.

eralized eigenvector) of a $\left\lceil \tilde{L}_w/2 \right\rceil \times \left\lceil \tilde{L}_w/2 \right\rceil$ matrix, as shown in [12]. The MERRY algorithm, which converges to the MSSNR TEQ in the absence of noise, can also be made to compute a symmetric TEQ. This halves the number of multiplies in the MERRY update equation. Such approaches lead to a significant reduction in complexity, but with the cost of a possible performance loss. Reduced complexity algorithms are discussed in Section 6.5. Performance and complexity of such algorithms will be considered in Chapter 8.

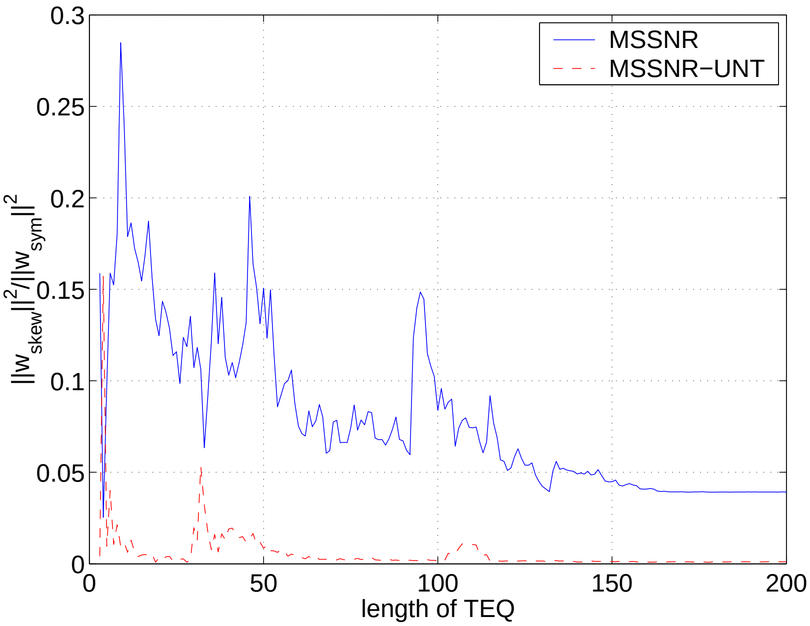

Figure 6.4: Energy in the skew-symmetric part of the TEQ over the energy in the symmetric part of the TEQ, for $\nu = 32$. The data was delay-optimized and averaged over CSA test loops 1 - 8.

6.2 Infinite length MSSNR results

This section examines the limiting behavior of $\mathbf{A}$ and $\mathbf{B}$, and the resulting limiting behavior of their eigenvectors (i.e. the MSSNR and MSSNR-UNT TEQs). We will show that

$$\lim_{L_w \to \infty} \frac{\|\mathbf{H}^H\mathbf{H} - \mathbf{A}\|_F}{\|\mathbf{A}\|_F} = 0, \tag{6.10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm [35]. Since $\mathbf{H}^H\mathbf{H} \in \mathcal{C}^*$, its eigenvectors are symmetric or skew-symmetric. Thus, as $L_w \to \infty$, we can expect the eigenvectors of $\mathbf{A}$ to become conjugate symmetric or conjugate skew-symmetric. Although this is a heuristic argument, the more rigorous $\sin(\theta)$ theorem[3] [22] is difficult to apply.

[3] The $\sin(\theta)$ theorem is a commonly used bound on the angle between the eigenvector of a matrix and the corresponding eigenvector of the perturbed matrix.

First, consider a TEQ that is finite, but very long. Specifically, we make the following assumptions:

A1: $\Delta > L_h > \nu$,

A2: $L_w > \Delta + \nu$.

Such a large Δ in *A1* is reasonable when the TEQ length is large. Now we can partition $\mathbf{H}$ as

$$\mathbf{H} = \begin{array}{l} \left[\begin{array}{ccccc} \mathbf{H}_1 & \mathbf{H}_{L2} & \mathbf{H}_{L1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{U3} & \mathbf{H}_M & \mathbf{H}_{L3} & \mathbf{0} \\ \underbrace{\mathbf{0}}_{(\Delta-L_h)} & \underbrace{\mathbf{0}}_{\nu} & \underbrace{\mathbf{H}_{U1}}_{(L_h-\nu)} & \underbrace{\mathbf{H}_{U2}}_{\nu} & \underbrace{\mathbf{H}_2}_{(L_w-\nu-\Delta)} \end{array} \right] \end{array} \begin{array}{l} \}\Delta \\ \}(\nu+1) \\ \}(L_h+L_w-\nu-\Delta) \end{array} \tag{6.11}$$

where $[\mathbf{H}_{L2}, \mathbf{H}_{L1}]$ and $\mathbf{H}_{L3}$ are both lower triangular and contain the "head" of the channel, $[\mathbf{H}_{U1}, \mathbf{H}_{U2}]$ and $\mathbf{H}_{U3}$ are both upper triangular and contain the "tail" of the channel, $\mathbf{H}_1$ and $\mathbf{H}_2$ are tall channel convolution matrices, and $\mathbf{H}_M$ is Toeplitz. Then $\mathbf{H}_{win}$ is simply the middle row (of blocks) of $\mathbf{H}$, and $\mathbf{H}_{wall}$ is the concatenation of the top and bottom rows.

Under the two assumptions above, $\mathbf{H}_{U3}$, $\mathbf{H}_M$, and $\mathbf{H}_{L3}$ will be constant for all values of Δ and L_w. As such, the limiting behavior of $\mathbf{B} = \mathbf{H}_{win}^H \mathbf{H}_{win}$ is

$$\begin{aligned} \mathbf{B} &= [\mathbf{0}, \mathbf{H}_{U3}, \mathbf{H}_M, \mathbf{H}_{L3}, \mathbf{0}]^H [\mathbf{0}, \mathbf{H}_{U3}, \mathbf{H}_M, \mathbf{H}_{L3}, \mathbf{0}] \\ &\triangleq \left[\mathbf{0}, \overline{\mathbf{H}}_3^T, \mathbf{0}\right]^H \left[\mathbf{0}, \overline{\mathbf{H}}_3^T, \mathbf{0}\right]. \end{aligned} \tag{6.12}$$

This bound is a function of the eigenvalue separation of the matrix, which is not explicitly known in our problem; hence, the theorem cannot be directly applied.

Note that $\overline{\mathbf{H}}_3$ is a size $(\nu + \tilde{L}_h) \times (\nu + 1)$ channel convolution matrix formed from $\overline{\mathbf{h}}$, the time-reversed channel. Since $\mathbf{B}$ is a zero-padded version of $\overline{\mathbf{H}}_3^* \overline{\mathbf{H}}_3^T$, it has the same Frobenius norm, which is a constant. Therefore,

$$\|\mathbf{B}\|_F^2 = \|\overline{\mathbf{H}}_3^* \overline{\mathbf{H}}_3^T\|_F^2 \stackrel{\triangle}{=} \beta, \tag{6.13}$$

whenever our two initial assumptions *A1* and *A2* are met.

Now consider the limiting behavior for $\mathbf{A}$, which equals

$$\mathbf{A} = \begin{bmatrix} \mathbf{H}_1^H \mathbf{H}_1 & \mathbf{H}_1^H \mathbf{H}_{L2} & \mathbf{H}_1^H \mathbf{H}_{L1} & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_{L2}^H \mathbf{H}_1 & \mathbf{H}_{L2}^H \mathbf{H}_{L2} & \mathbf{H}_{L2}^H \mathbf{H}_{L1} & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_{L1}^H \mathbf{H}_1 & \mathbf{H}_{L1}^H \mathbf{H}_{L2} & \mathbf{H}_{L1}^H \mathbf{H}_{L1} + \mathbf{H}_{U1}^H \mathbf{H}_{U1} & \mathbf{H}_{U1}^H \mathbf{H}_{U2} & \mathbf{H}_{U1}^H \mathbf{H}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_{U2}^H \mathbf{H}_{U1} & \mathbf{H}_{U2}^H \mathbf{H}_{U2} & \mathbf{H}_{U2}^H \mathbf{H}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_2^H \mathbf{H}_{U1} & \mathbf{H}_2^H \mathbf{H}_{U2} & \mathbf{H}_2^H \mathbf{H}_2 \end{bmatrix} \tag{6.14}$$

Thus, a lower bound on the Frobenius norm of $\mathbf{A}$ can be found as follows:

$$\begin{aligned} \|\mathbf{A}\|_F^2 &\geq \|\mathbf{H}_1^H \mathbf{H}_1\|_F^2 + \|\mathbf{H}_2^H \mathbf{H}_2\|_F^2 \\ &\geq \|\mathbf{h}\|_2^4 \cdot ((\Delta - L_h) + (L_w - \nu - \Delta)) \\ &= \|\mathbf{h}\|_2^4 \cdot (L_w - L_h - \nu), \end{aligned} \tag{6.15}$$

which goes to infinity as $L_w \to \infty$. In the second inequality, we have dropped all of the terms in the Frobenius norms except for those due to the diagonal elements of $\mathbf{H}_1^H \mathbf{H}_1$ and $\mathbf{H}_2^H \mathbf{H}_2$.

Now let $\mathbf{C} \stackrel{\triangle}{=} \mathbf{H}^H \mathbf{H}$, and note from (6.11) that $\mathbf{C} = \mathbf{A} + \mathbf{B}$. Thus,

$$\frac{\|\mathbf{C} - \mathbf{A}\|_F^2}{\|\mathbf{A}\|_F^2} = \frac{\|\mathbf{B}\|_F^2}{\|\mathbf{A}\|_F^2} \leq \frac{\beta}{\|\mathbf{h}\|_2^4 \cdot (L_w - L_h - \nu)}, \tag{6.16}$$

which goes to zero as $L_w \to \infty$. Thus, in the limit, $\mathbf{A}$ approaches $\mathbf{C}$, which is a Hermitian Toeplitz matrix, and hence is in $\mathcal{C}^*$. Heuristically, this suggests that in the limit, the eigenvectors of $\mathbf{A}$ (including the MSSNR-UNT solution) will be conjugate symmetric or conjugate skew-symmetric.

Note that in the limit, $\mathbf{B}$ does not become Hermitian centro-Hermitian (refer to (6.12)), although it is approximately so. Thus, we cannot make as strong of a limiting argument for the MSSNR solution as for the MSSNR-UNT solution.

6.3 Infinite length MMSE results

Now we consider the effects of an infinite length MMSE TEQ. Specifically, the goal is to show that in the limit, the TIR becomes symmetric, although the TEQ may or may not. In addition, a useful corollary arises concerning the zero locations of the MMSE TIR. For simplicity, this section assumes a real TEQ and TIR.

Theorem 6.3.1 *Assume the input signal is white, and the noise is non-zero (and possibly non-white). If the TEQ* $\mathbf{w}$ *is an infinite length real discrete-time filter, then the finite length real TIR* $\mathbf{b}$ *will become symmetric or skew-symmetric.*

Remarks: The proof is given in Appendix B. An outline of the proof of a special case of this theorem was given in [30], which required that the noise was white, that $\Delta = 0$, and that $\mathbf{w}$ was a continuous-time filter. The fact that the TIR in that case was either symmetric or skew-symmetric was pointed out separately in [13].

The implications are that for long TEQs, the TIR design can be implemented more efficiently when the symmetry is exploited. As shown by Theorem 6.3.1, the TIR for an infinite length TEQ is an eigenvector of a symmetric centrosymmetric matrix. The TIR for a finite length TEQ is thus expected to be an eigenvector of a matrix that is approximately centrosymmetric. This is analogous to the approximate symmetry of a finite length MSSNR TEQ shown in Figs. 6.2 and 6.3. Thus, using results in [12], the computation of $\mathbf{b}$ can be reduced from an eigendecomposition of a matrix of size $(\nu+1)\times(\nu+1)$ to an eigendecomposition of a matrix of size $\lceil(\nu+1)/2\rceil \times \lceil(\nu+1)/2\rceil$.

A useful corollary of Theorem 6.3.1 explains why the MMSE and MSSNR based TEQ design methods have nulls in their magnitude responses, a fact observed in [31]. Recall that the MMSE TEQ design uses a target impulse response (TIR) $\mathbf{b}$ that must satisfy the relation [4]

$$\mathbf{R}_{rx}\ \mathbf{b} = \mathbf{R}_r\ \mathbf{w}, \tag{6.17}$$

where $\mathbf{R}_{xr}$ is the channel input-output cross-correlation matrix and $\mathbf{R}_r$ is the channel output autocorrelation matrix. Typically, $\mathbf{b}$ is computed first, and then (6.17) is used to determine $\mathbf{w}$. The goal is that $\mathbf{h} \star \mathbf{w}$ approximates a delayed version of $\mathbf{b}$. As such, if $\mathbf{b}$ has nulls in its magnitude response, then $\mathbf{w}$ (and/or $\mathbf{h}$) will have them as well.

Recall that for an i.i.d. channel input sequence, the target impulse response is the eigenvector corresponding to the minimum eigenvalue of [5], [30], [31]

$$\begin{aligned} \mathbf{R}(\Delta) &= \mathbf{R}_x - \mathbf{R}_{xr}\mathbf{R}_r^{-1}\mathbf{R}_{rx} \\ &= \mathbf{I}_{(\nu+1)} - \Omega\mathbf{H}\left(\mathbf{H}^T\mathbf{H} + \mathbf{R}_n\right)^{-1}\mathbf{H}^T\Omega^T, \end{aligned} \tag{6.18}$$

where $\mathbf{H}$ is the channel convolution matrix of size $\tilde{L}_c \times \tilde{L}_w$, and

$$\Omega = \left[\mathbf{0}_{(\nu+1)\times\Delta},\ \mathbf{I}_{\nu+1},\ \mathbf{0}_{(\nu+1)\times(L_c-\nu-\Delta)}\right]. \tag{6.19}$$

Robinson ([87], pp. 269–272) has shown that the eigenvector corresponding to the largest eigenvalue of a symmetric Toeplitz matrix will have all of its zeros on the unit circle, and Makhoul [51] has generalized this to show that the zeros of the eigenvector corresponding to the smallest eigenvalue has all of its zeros on the unit circle. (The proofs rely on the assumption that the corresponding eigenvalue has multiplicity 1.) Thus, we would like to show that $\mathbf{R}(\Delta)$ is approximately Toeplitz (and it is clearly symmetric), which would imply that the MMSE TIR has its zeros near the unit circle. Furthermore, if we remove the term $\mathbf{R}_n$ from (6.18) and compute $\mathbf{b}$, then we obtain a windowed version of the MSSNR TEQ. So, the preceding approach could also be used to show that the MSSNR TEQ has its zeros near the unit circle.

First consider a limiting case: an infinite length TEQ with a finite length TIR.

Corollary 6.3.1 *Assume the input signal is white, and the noise is non-zero (and possibly non-white). If the TEQ* $\mathbf{w}$ *is allowed to be any infinite length discrete-time filter, and if the minimum eigenvalue of* $\mathbf{R}(\Delta)$ *has multiplicity* 1*, then the finite length MMSE TIR* $\mathbf{b}$ *will have all* ν *of its zeros on the unit circle. As a consequence, the effective channel impulse response will have* ν *zeros approximately on the unit circle.*

Proof: The proof of Theorem 6.3.1 has shown that for the conditions stated above, the matrix $\mathbf{R}(\Delta)$ becomes a symmetric Toeplitz matrix. As discussed above, Robinson [87] and Makhoul [51] have shown that the eigenvector corresponding to the minimum eigenvalue of a symmetric Toeplitz matrix has all of its zeros on the unit circle, so long as the eigenvalue has multiplicity 1. (In fact, any of the eigenvectors will have its zeros on the unit circle, so long as the corresponding eigenvalue has multiplicity 1.) This implies that the TIR has ν zeros on the unit circle. Since the TEQ has infinite length, the effective channel will be approximately a zero-padded version of the TIR. Hence, the channel-TEQ frequency response will have ν zeros approximately on the unit circle. ■

Remarks: Daly, Heneghan, and Fagan [20] have shown that in the noiseless case with a white input, the MMSE and MSSNR design methods produce identical TEQs. Thus, we can infer that for a finite length MSSNR TEQ, the TEQ transfer function will have ν zeros approximately on the unit circle.

For multicarrier systems, if a null lies at one of the subchannel carrier frequencies, then no data can be transmitted in that subchannel. This is a severe problem [31]. This result helps explain why the MMSE TEQ exhibits poor performance for large TEQ lengths [7].

A practical design consideration is the question of how quickly the infinite length results are approached as the TEQ length is increased. In this case, $\mathbf{R}(\Delta)$ is symmetric, but not quite Toeplitz. Thus, the zeros of its eigenvector $\mathbf{b}$ will not be precisely on the unit circle. Analytic results for this case are rather intractable, so we will give empirical results. Figure 6.5 plots the average distance of the zeros of the TIR to the unit circle. The distances were sorted (least to greatest) and then averaged over CSA loops 1 through 8. There are 32 curves, one for each zero. Observe that most of the zeros start out at a distance of about 0.2. For a length 32 TEQ, the zeros are clustered around a distance of 0.01 from the unit circle; and for a length 100 TEQ, the zeros are clustered around a distance of 10^{-4} from the unit circle. The asymptotic results agree with Corollary 6.3.1.

6.4 Linear phase and the FEQ

A symmetric TEQ can be classified as either a Type I or Type II FIR Linear Phase System [77, pp. 298–299]. These forms generalize for complex filters that are conjugate symmetric. A conjugate symmetric TEQ $\mathbf{w}$ with $L_w + 1$ taps obeys

$$w(k) = w^*\left(L_w - k\right). \tag{6.20}$$

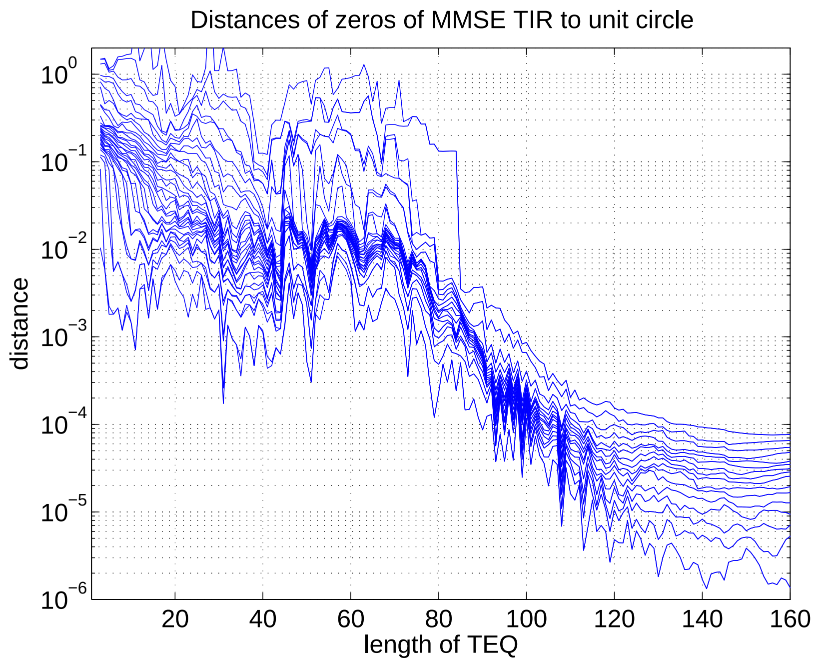

Figure 6.5: Distance of the zeros of the MMSE TIR to the unit circle. The values are averaged over CSA loops 1 through 8. Each curve represents the distance for a single zero.

Let $L_w + 1$ be even for simplicity. Generally, $L_w + 1$ is a power of 2 in practice, but in any case the results from (6.24) onwards also hold for the odd length case. Applying (6.20),

$$W\left(e^{j\omega}\right) = \sum_{k=0}^{(L_w-1)/2} \left(w(k)e^{-j\omega k} + w^*(k)e^{-j\omega(L_w-k)}\right) \tag{6.21}$$

$$= e^{-j\frac{1}{2}\omega L_w} \sum_{k=0}^{(L_w-1)/2} \left(w(k)e^{-j\omega k}e^{j\frac{1}{2}\omega L_w} + w^*(k)e^{-j\omega k}e^{-j\frac{1}{2}\omega L_w}\right) \tag{6.22}$$

$$= e^{-j\frac{1}{2}\omega L_w} \underbrace{\sum_{k=0}^{(L_w-1)/2} 2\,\mathcal{R}\left\{w(k)e^{-j\omega k}e^{j\frac{1}{2}\omega L_w}\right\}}_{\widehat{M}(\omega)}. \tag{6.23}$$

The term $\widehat{M}(\omega)$ is real, but not necessarily positive. To force a positive magnitude response, we use the form

$$W\left(e^{j\omega}\right) = M(\omega) \exp\left(-j\frac{L_w}{2}\omega + j\beta\right), \tag{6.24}$$

where $M(\omega) = \left|\widehat{M}(\omega)\right|$ is the magnitude response and $\beta \in \{0, \pi\}$. We determine β as follows. The DC response can be obtained by setting $\omega = 0$ in (6.24) or by setting $\omega = 0$ in the standard form of the Fourier transform of $\mathbf{w}$; equating these two, we get

$$M(0)\ e^{j\beta} = \sum_{k=0}^{L_w} w(k). \tag{6.25}$$

Then β is given by

$$\beta = \begin{cases} 0, & \sum_k w(k) > 0, \\ \pi, & \sum_k w(k) < 0. \end{cases} \tag{6.26}$$

If $\sum_k w(k) = 0$, the DC response does not reveal the value of β. In this case, one must determine the phase response at another frequency, which is more complicated to compute. The response at $\omega = \pi$ is fairly easy to compute, and will also reveal the value of β.

From (6.24) – (6.26), given the TEQ length, the phase response of a conjugate symmetric TEQ is known up to the factor $e^{j\beta}$, even before the TEQ is designed. The phases of the FEQs are then determined entirely by the channel phase response. Thus, if a channel estimate is available, the two possible FEQ phase responses could be determined in parallel with the TEQ design. If differential encoding is used, then the value of β can arbitrarily be set to either 0 or π, since a rotation of exactly 180 degrees does not affect the output of a differential detector. Furthermore, if 2-PAM or 4-QAM signaling is used on a subcarrier, the magnitude of the FEQ does not matter, and the entire FEQ for that tone can be designed without knowledge of the TEQ.

For an ADSL system, 4-QAM signaling is used on all of the subcarriers during training. Thus, the FEQ can be designed for the training phase by only setting its phase response. The magnitude response can be set after the TEQ is designed. The benefit here is that if the FEQ is designed all at once (both magnitude and phase), then a division of complex numbers is required for each tone. However, if the phase response is already known, determining the FEQ magnitude only requires a division of real numbers for each tone. This can allow for a more efficient implementation.

6.5 Implications for receiver design

This section presents practical algorithms for designing a TEQ that is perfectly symmetric or that arises from a TIR that is perfectly symmetric. For simplicity, this section considers a real TEQ and a real TIR. The designs we will consider are the Sym-MSSNR TEQ design, the Sym-MMSE TIR design, and the Sym-MERRY adaptive TEQ design. The Sym-Min-ISI design, though not discussed here, is discussed in [26]. Such designs reduce the complexity of computing the TEQ, and they have the added benefit of allowing the exploitation of the phase properties discussed in Section 6.4.

First, consider designing a perfectly symmetric MSSNR TEQ. In [85] and [86], simulations were presented for forcing the MSSNR TEQ to be perfectly symmetric or skew-symmetric, though no justification was provided for this approach. If the TEQ length $\tilde{L}_w$ were even, then we could enforce the symmetry by

$$\mathbf{w}^T = \left[\mathbf{v}^T, (\mathbf{J}\mathbf{v})^T\right], \tag{6.27}$$

and if $\tilde{L}_w$ were odd, we could enforce the symmetry by

$$\mathbf{w}^T = \left[\mathbf{v}^T, \gamma, (\mathbf{J}\mathbf{v})^T\right], \tag{6.28}$$

where in each case $\mathbf{v}$ has dimensions $\left\lfloor \tilde{L}_w/2 \right\rfloor \times 1$. For the even-length case, the generalized eigenvalue problem of (2.2) can be simplified via

$$\left[\mathbf{v}^T, \mathbf{v}^T\mathbf{J}\right] \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{J}\mathbf{v} \end{bmatrix} = \mathbf{v}^T \underbrace{\left[\mathbf{A}_{11} + \mathbf{J}\mathbf{A}_{21} + \mathbf{A}_{12}\mathbf{J} + \mathbf{J}\mathbf{A}_{22}\mathbf{J}\right]}_{\widehat{\mathbf{A}}} \mathbf{v}, \quad (6.29)$$

$$\left[\mathbf{v}^T, \mathbf{v}^T\mathbf{J}\right] \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{J}\mathbf{v} \end{bmatrix} = \mathbf{v}^T \underbrace{\left[\mathbf{B}_{11} + \mathbf{J}\mathbf{B}_{21} + \mathbf{B}_{12}\mathbf{J} + \mathbf{J}\mathbf{B}_{22}\mathbf{J}\right]}_{\widehat{\mathbf{B}}} \mathbf{v}. \quad (6.30)$$

Then the MSSNR design problem becomes

$$\min_{\mathbf{v}} \left(\mathbf{v}^T \widehat{\mathbf{A}} \mathbf{v}\right) \quad \text{subject to} \quad \mathbf{v}^T \widehat{\mathbf{B}} \mathbf{v} = 1. \quad (6.31)$$

The solution for $\mathbf{v}$ is the generalized eigenvector of the matrix pair $(\widehat{\mathbf{A}}, \widehat{\mathbf{B}})$ corresponding to the minimum eigenvalue. Thus, an efficient implementation consists of:

1. Compute $\mathbf{A}$ and $\mathbf{B}$ using the fast algorithms in [111] and Chapter 7.

2. Form $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{B}}$ as in (6.29) and (6.30), requiring approximately $\frac{3}{2}\left(\frac{\tilde{L}_w}{2}\right)^2$ additions/subtractions each (but no multiplications).

3. Find the generalized eigenvector of $(\widehat{\mathbf{A}}, \widehat{\mathbf{B}})$ corresponding to the minimum eigenvalue using the methods surveyed in Chapter 2.

4. Repeat steps 1 – 3 for each value of Δ (which changes $\mathbf{A}$ and $\mathbf{B}$), then pick the value of Δ and corresponding solution that yields the smallest generalized eigenvalue.

Remarks: First, step 2 requires only $\frac{3}{2}\left(\frac{\tilde{L}_w}{2}\right)^2$ additions each rather than $3\left(\frac{\tilde{L}_w}{2}\right)^2$ additions each, since $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{B}}$ are symmetric. Second, the extra additions in step 2 are negligible. Third, step 3 requires a generalized eigendecomposition of symmetric matrices of size $\frac{1}{2}\tilde{L}_w \times \frac{1}{2}\tilde{L}_w$ rather than a generalized eigendecomposition of symmetric matrices of size $\tilde{L}_w \times \tilde{L}_w$. This reducing the number of multiply-adds by a factor of 4, since an eigendecomposition of an $\tilde{L}_w \times \tilde{L}_w$ symmetric matrix requires $\mathcal{O}\left(\tilde{L}_w^2\right)$ computations.

Similar results hold for the odd TEQ length case of (6.28), except now we have

$$\begin{aligned}
&\left[\mathbf{v}^T, \gamma, \mathbf{v}^T\mathbf{J}\right] \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \gamma \\ \mathbf{J}\mathbf{v} \end{bmatrix} \\
&\qquad = \left[\mathbf{v}^T, \gamma\right] \underbrace{\begin{bmatrix} \left(\mathbf{A}_{11} + \mathbf{A}_{13}\mathbf{J} + \mathbf{J}\mathbf{A}_{31} + \mathbf{J}\mathbf{A}_{33}\mathbf{J}\right) & \left(\mathbf{A}_{12} + \mathbf{J}\mathbf{A}_{32}\right) \\ \left(\mathbf{A}_{21} + \mathbf{A}_{23}\mathbf{J}\right) & \mathbf{A}_{22} \end{bmatrix}}_{\widehat{\mathbf{A}}} \begin{bmatrix} \mathbf{v} \\ \gamma \end{bmatrix},
\end{aligned} \tag{6.32}$$

with an analogous definition of $\widehat{\mathbf{B}}$ (replace each $\mathbf{A}_{ij}$ with $\mathbf{B}_{ij}$). In this case, we have reduced $\mathbf{A}$ (size $\tilde{L}_w \times \tilde{L}_w$) to $\widehat{\mathbf{A}}$ (size $\left\lceil \tilde{L}_w/2 \right\rceil \times \left\lceil \tilde{L}_w/2 \right\rceil$). Note that this is a generalization of the results in [12], in which $\mathbf{A}$ and $\mathbf{B}$ are exactly centrosymmetric.

For the MSSNR-UNT approach, we set $\widehat{\mathbf{B}} = \mathbf{I}$. This leaves the norm constraint on $\mathbf{v}$ rather than on $\mathbf{w}$, which can be dealt with by renormalizing $\mathbf{w}$ after we construct it from $\mathbf{v}$. The MSSNR-UNT TEQ arises, for example, in the MERRY algorithm [53], which is a blind, adaptive algorithm for computing the TEQ; or in Nafie and Gatherer's algorithm [75] (if the constraint used is a unit norm TEQ), which is a trained, iterative algorithm for computing the TEQ. We focus next on extending the MERRY algorithm to the symmetric case.

In practical applications, the TEQ length is often even (a power of two, in fact), due to a desired efficient use of memory. A symmetric TEQ has the form (6.27), though an odd-length symmetric TEQ could be accommodated using the form in

(6.28). The TEQ output is

$$y(Mk+i) = \sum_{j=0}^{L_w} w(j) \cdot r(Mk+i-j), \tag{6.33}$$

which can be rewritten for a symmetric TEQ as

$$\begin{aligned} y(Mk+i) &= \sum_{j=0}^{\tilde{L}_w/2-1} v(j) \cdot [r(Mk+i-j) + r(Mk+i-L_w+j)] && (6.34) \\ &= \mathbf{v}^T\ \mathbf{u}(Mk+i), && (6.35) \end{aligned}$$

where

$$\mathbf{u}(i) = \left[r(i) + r(i-L_w),\ \cdots,\ r\left(i - \frac{\tilde{L}_w}{2} + 1\right) + r\left(i - \frac{\tilde{L}_w}{2}\right) \right]^T. \tag{6.36}$$

The Sym-MERRY update is a stochastic gradient descent of the MERRY cost function (3.4), but now with respect to the half-TEQ coefficients $\mathbf{v}$. Again, a renormalization can be used to avoid the trivial solution $\mathbf{v} = \mathbf{0}$. Making use of (6.34), the Sym-MERRY algorithm is

$$\text{Sym-MERRY:} \quad \boxed{\begin{aligned} &\text{Given } \Delta, \text{ for symbol } k = 0, 1, 2, \ldots, \\ &\quad \tilde{\mathbf{u}}(k) = \mathbf{u}(Mk+\nu+\Delta) - \mathbf{u}(Mk+\nu+N+\Delta) \\ &\quad e(k) = \mathbf{v}^T(k)\ \tilde{\mathbf{u}}(k) \\ &\hat{\mathbf{v}}(k+1) = \mathbf{v}(k) - \mu\ e(k)\ \tilde{\mathbf{u}}^*(k) \\ &\mathbf{v}(k+1) = \frac{\hat{\mathbf{v}}(k+1)}{\|\hat{\mathbf{v}}(k+1)\|} \end{aligned}} \tag{6.37}$$

Compared to the regular MERRY algorithm in (3.5), the number of multiplications has been cut in half for Sym-MERRY, though a few additional additions are needed to compute $\tilde{\mathbf{u}}$. A symmetric FRODO algorithm could be developed as well to avoid the expensive renormalization. Simulations of Sym-MERRY are presented in Chapter 8.

Now consider designing the MMSE TEQ to have a perfectly symmetric TIR, rather than a perfectly symmetric TEQ. This is useful because in the MMSE design, far more computational power is devoted to computing the TIR than the TEQ. The MSE (which we wish to minimize) is given by

$$\mathrm{E}\left[e^2\right] = \mathbf{b}^T\ \mathbf{R}\left(\Delta\right)\ \mathbf{b}. \tag{6.38}$$

Typically, the CP length ν is a power of 2 (see Table 1.5), so the TIR length $\nu+1$ is odd. To force a symmetric odd-length TIR, partition the TIR as

$$\mathbf{b}^T = \left[\mathbf{v}^T, \gamma, \left(\mathbf{J}\mathbf{v}\right)^T\right], \tag{6.39}$$

where γ is a scalar and $\mathbf{v}$ is a real $\frac{\nu}{2} \times 1$ vector. Now rewrite the MSE as

$$\left[\mathbf{v}^T, \gamma, \mathbf{v}^T\mathbf{J}\right] \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{R}_{13} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \gamma \\ \mathbf{J}\mathbf{v} \end{bmatrix} = \left[\sqrt{2}\mathbf{v}^T, \gamma\right] \hat{\mathbf{R}} \begin{bmatrix} \sqrt{2}\mathbf{v} \\ \gamma \end{bmatrix} \tag{6.40}$$

where

$$\hat{\mathbf{R}} = \begin{bmatrix} \frac{1}{2}\left(\mathbf{R}_{11} + \mathbf{R}_{13}\mathbf{J} + \mathbf{J}\mathbf{R}_{31} + \mathbf{J}\mathbf{R}_{33}\mathbf{J}\right) & \frac{1}{\sqrt{2}}\left(\mathbf{R}_{12} + \mathbf{J}\mathbf{R}_{32}\right) \\ \frac{1}{\sqrt{2}}\left(\mathbf{R}_{21} + \mathbf{R}_{23}\mathbf{J}\right) & \mathbf{R}_{22} \end{bmatrix} \tag{6.41}$$

For simplicity, let $\hat{\mathbf{v}}^T = \left[\sqrt{2}\mathbf{v}^T, \gamma\right]$. In order to prevent the all-zero solution, the non-symmetric TIR design uses the constraint $\|\mathbf{b}\| = 1$. This is equivalent to the constraint $\|\hat{\mathbf{v}}\| = 1$. Under this constraint, the TIR that minimizes the MSE must satisfy

$$\hat{\mathbf{R}}\ \hat{\mathbf{v}} = \lambda\ \hat{\mathbf{v}}, \tag{6.42}$$

where λ is the smallest eigenvalue of $\hat{\mathbf{R}}$. Since both $\mathbf{R}$ and $\hat{\mathbf{R}}$ are symmetric, solving (6.42) requires $\frac{1}{4}$ as many computations as solving the initial eigenvector problem. However, the forced symmetry could, in principle, degrade the performance of the associated TEQ. Simulations of the Sym-MMSE algorithm are presented in Chapter 8.

Chapter 7

Efficient Matrix Computation

"Be not ta'en tardy by unwise delay."

– William Shakespeare, *Richard III*, Act IV, Scene i.

This chapter[1] discusses efficient methods of computing the matrices needed for MMSE, MSSNR, Min-IBI, and MDS channel shorteners. The focus is the re-use of computations from one delay to the next. This chapter assumes a rough familiarity with the material in Section 2.2.

The TEQs that will be discussed in Sections 7.1–7.4 can be considered to be special cases of the general form proposed in [98]. Consider minimization of the cost function [98]

$$J = \alpha J_{short} + (1-\alpha) J_{noise} \tag{7.1}$$

$$= \frac{\alpha \sum_{n=0}^{L_c} f(n-\Delta)\left|c(n)\right|^2 + (1-\alpha)\frac{\sigma_q^2}{\sigma_x^2}}{\sum_{n=0}^{L_c}\left|c(n)\right|^2} \tag{7.2}$$

where $\sigma_q^2 = \mathbf{w}^H \mathbf{R}_n \mathbf{w}$ is the power of the filtered noise. The function $f(\cdot)$ in the numerator is chosen to penalize channel taps in undesired locations, which are then minimized with respect to the entire channel energy (via the denominator). The α and $1-\alpha$ weights allow for variable suppression of the noise as well. Throughout this chapter, we assume unit signal power, $\sigma_x^2 = 1$.

Eq. (7.2) can be rewritten as a generalized Rayleigh quotient, leading to

$$\mathbf{w}_{opt} = \arg\min_{\mathbf{w}} \frac{\mathbf{w}^H \mathbf{A} \mathbf{w}}{\mathbf{w}^H \mathbf{C} \mathbf{w}}, \tag{7.3}$$

[1]Some material in this chapter has been previously published. © 2004 Hindawi Publishing Corporation. Reprinted, with permission, from [55].

where

$$\mathbf{A} = \alpha\mathbf{H}^H\mathbf{Q}\mathbf{H} + (1-\alpha)\mathbf{R}_n, \tag{7.4}$$

$$\mathbf{C} = \mathbf{H}^H\mathbf{H}, \tag{7.5}$$

and where $\mathbf{Q}(\Delta)$ is a diagonal matrix with n^{th} diagonal element equal to $f(n-\Delta)$. Designs that fit into this framework include the maximum shortening SNR (MSSNR) design [67], the minimum MSE (MMSE) design with a white input [4], the Min-IBI design [15], and the MDS design [92]. This chapter addresses complexity reduction of these designs.

7.1 Efficient MSSNR computation

There is a tremendous amount of redundancy involved in the brute force calculation of the MSSNR design. Reusing some of this redundancy has been addressed in [111]. This section discusses methods of reusing even more of the computations to dramatically decrease the required complexity. Specifically, for a given delay Δ, $\mathbf{A}(\Delta)$ and $\mathbf{B}(\Delta+1)$ can both be computed from $\mathbf{B}(\Delta)$ almost for free, where $\mathbf{B} = \mathbf{H}_{win}^H\mathbf{H}_{win}$ is maintained solely for the purpose of updating $\mathbf{A}$ efficiently.

Computing $\mathbf{A}(\Delta)$ from $\mathbf{B}(\Delta)$

Let $\alpha = 1$ in (7.4) and let $\mathbf{Q}$ be a "wall" matrix,

$$\mathbf{Q}_{ssnr}(\Delta) = \mathrm{diag}[\underbrace{1,\cdots,1}_{\Delta},\underbrace{0,\cdots,0}_{\nu+1},\underbrace{1,\cdots,1}_{L_c-\nu-\Delta}], \tag{7.6}$$

where "diag$[\cdot]$" is a diagonal matrix with the elements of the argument along the main diagonal. Then we have $\mathbf{A} = \mathbf{H}_{wall}^H\mathbf{H}_{wall}$. For convenience, also define

$\mathbf{B} = \mathbf{H}_{win}^H \mathbf{H}_{win}$. From (2.7) and (2.8) we can write

$$\mathbf{H} = \left[\begin{array}{c} \mathbf{H}_1 \\ \hline \mathbf{H}_{win} \\ \hline \mathbf{H}_2 \end{array} \right], \quad \mathbf{H}_{wall} = \left[\begin{array}{c} \mathbf{H}_1 \\ \hline \mathbf{H}_2 \end{array} \right]. \tag{7.7}$$

Thus,

$$\mathbf{C} = \mathbf{H}_1^H \mathbf{H}_1 + \mathbf{H}_{win}^H \mathbf{H}_{win} + \mathbf{H}_2^H \mathbf{H}_2 \tag{7.8}$$

$$= \underbrace{\left(\mathbf{H}_1^H \mathbf{H}_1 + \mathbf{H}_2^H \mathbf{H}_2\right)}_{\mathbf{A}} + \underbrace{\left(\mathbf{H}_{win}^H \mathbf{H}_{win}\right)}_{\mathbf{B}}. \tag{7.9}$$

To emphasize dependence on and independence of the delay Δ, we write

$$\boxed{\mathbf{C} = \mathbf{A}(\Delta) + \mathbf{B}(\Delta)} \tag{7.10}$$

Since $\mathbf{C}$ is Hermitian and Toeplitz, it is fully determined by its first row or column:

$$\mathbf{C}_{(0:L_w,0)} = \mathbf{H}^H \left[\mathbf{h}^T, \mathbf{0}_{(1\times L_w)}\right]^T = \left(\mathbf{H}_{(0:L_h,0:L_w)}\right)^H \mathbf{h}. \tag{7.11}$$

$\mathbf{C}$ can be computed using less than $\tilde{L}_h^2$ multiply adds and its first column can be stored using $\tilde{L}_w$ memory words[2]. Since $\mathbf{C}$ is independent of Δ, we only need to compute it once. Then each time Δ is incremented and the new $\mathbf{B}(\Delta)$ is computed, $\mathbf{A}(\Delta)$ can be computed from $\mathbf{A}(\Delta) = \mathbf{C} - \mathbf{B}(\Delta)$ using only $\tilde{L}_w^2$ additions and no multiplications. In constrast, the "brute force" method requires $\tilde{L}_w^2 (L_h - \nu)$ multiply-adds per delay to compute $\mathbf{A}(\Delta)$, and Wu's method [111] requires about $\tilde{L}_w (L_w + L_h - \nu)$ multiply-adds per delay.

[2] A memory word is defined as the amount of space needed to store one complex number.

Computing $\mathbf{B}(\Delta+1)$ from $\mathbf{B}(\Delta)$

Recall that $\mathbf{B}(\Delta) = \mathbf{H}_{win}^H(\Delta)\mathbf{H}_{win}(\Delta)$, where

$$\mathbf{H}_{win}(\Delta) = \begin{bmatrix} h(\Delta) & h(\Delta-1) & \cdots & h(\Delta-\tilde{L}_w+1) \\ \vdots & & \ddots & \vdots \\ h(\Delta+\nu) & h(\Delta+\nu-1) & \cdots & h(\Delta+\nu-\tilde{L}_w+1) \end{bmatrix} \tag{7.12}$$

The key observation is that

$$[\mathbf{H}_{win}(\Delta+1)]_{(0:\nu,1:L_w)} = [\mathbf{H}_{win}(\Delta)]_{(0:\nu,0:L_w-1)}. \tag{7.13}$$

This means that

$$\boxed{[\mathbf{B}(\Delta+1)]_{(1:L_w,1:L_w)} = [\mathbf{B}(\Delta)]_{(0:L_w-1,0:L_w-1)}} \tag{7.14}$$

so most of $\mathbf{B}(\Delta+1)$ can be obtained without requiring any computations. Now partition $\mathbf{B}(\Delta+1)$ as

$$\mathbf{B}(\Delta+1) = \left[\begin{array}{c|c} \alpha & \mathbf{g}^H \\ \hline \mathbf{g} & \hat{\mathbf{B}} \end{array}\right], \tag{7.15}$$

where $\hat{\mathbf{B}}$ is obtained from (7.14). Since $\mathbf{B}(\Delta+1)$ is almost Toeplitz, α and all of the elements of $\mathbf{g}$ save the last can be efficiently determined from the first column of $\hat{\mathbf{B}}$ via [111]

$$\begin{aligned}[\mathbf{B}(\Delta+1)]_{(i,j)} = [\mathbf{B}(\Delta+1)]_{(i+1,j+1)} &+ h^*((\Delta+1)-i+\nu)\ h((\Delta+1)-j+\nu) \\ &- h^*((\Delta+1)-i-1)\ h((\Delta+1)-j-1).\end{aligned} \tag{7.16}$$

Computing each of these L_w elements requires two multiply-adds. Finally, to compute the last element of $\mathbf{g}$,

$$\mathbf{g}_{(\nu-1)} = \left([\mathbf{H}_{win}]_{(0:\nu,L_w)}\right)^H\ [\mathbf{H}_{win}]_{(0:\nu,0)}, \tag{7.17}$$

1. Compute $\mathbf{B} = \mathbf{H}_{win}^{H}\left(\Delta_{min}\right)\mathbf{H}_{win}\left(\Delta_{min}\right)$.

2. $\mathbf{C}_{(0:L_w,0)} = \left(\mathbf{H}_{(0:L_h,0:L_w)}\right)^H \ \mathbf{h}$.

3. Fill in the rest of the Hermitian, Toeplitz matrix $\mathbf{C}$.

4. $\mathbf{A} = \mathbf{C} - \mathbf{B}$.

5. Solve $\mathbf{Aw} = \lambda\mathbf{Cw}$ for the generalized eigenvector corresponding to the smallest eigenvalue, as in [67].

6. For $\Delta = \Delta_{min} + 1 : \Delta_{max}$, do the following:

   (a) $[\mathbf{B}]_{(1:L_w,1:L_w)} = [\mathbf{B}]_{(0:L_w-1,0:L_w-1)}$

   (b) $\begin{cases} [\mathbf{B}]_{(0:L_w-1,0)} = [\mathbf{B}]_{(1:L_w,1)} \\ \qquad\qquad +h(\Delta+\nu)\cdot h^*(\Delta+\nu-[0:L_w-1]) \\ \qquad\qquad -h(\Delta-1)\cdot h^*(\Delta-1-[0:L_w-1]) \end{cases}$

   (c) $[\mathbf{B}]_{(L_w,0)} = \left([\mathbf{H}_{win}]_{(0:\nu,L_w)}\right)^H \ [\mathbf{H}_{win}]_{(0:\nu,0)}$

   (d) $[\mathbf{B}]_{(0,1:L_w)} = [\mathbf{B}]_{(1:L_w,0)}^H$

   (e) $\mathbf{A} = \mathbf{C} - \mathbf{B}$.

   (f) Solve $\mathbf{Aw} = \lambda\mathbf{Cw}$ for the generalized eigenvector corresponding to the smallest eigenvalue.

   (g) If this delay produces a smaller λ than the previous delay, save $\mathbf{w}$.

Figure 7.1: Fast MSSNR TEQ design algorithm.

requiring $\nu + 1$ multiply-adds.

The resulting fast MSSNR design algorithm is shown in Figure 7.1. Note that in step (b), the indices may become negative, in which case the corresponding elements are zero. It should be stressed that the gains will not be as apparent in an environment such as Matlab, since the brute force method is matrix based, and the proposed approach is an element-by-element approach. Matlab is optimized for the former, but embedded DSPs may not be.

Complexity comparison

Table 7.1 shows the (approximate) number of computations for each step of the MSSNR method, using the "brute force" approach, the method in [111], and the proposed approach. Note that N_Δ refers to the number of values of the delay that are possible (usually equal to the length of the effective channel minus the CP length). For a typical downstream ADSL system, the parameters are $\tilde{L}_w = L_w + 1 = 32$, $\tilde{L}_h = L_h + 1 = 512$, $L_c = L_w + L_h = 542$, $\nu = 32$, and $N_\Delta = \tilde{L}_c - \nu = 511$. The "example" lines in Table 7.1 show the required complexity for computing all of the $\mathbf{A}$'s and $\mathbf{B}$'s for these parameters using each approach. Observe that [111] beats the brute force method by a factor of 29, the proposed method beats [111] by a factor of 140, and the proposed method beats the brute force method by a factor of 4008.

7.2 Efficient Min-IBI computation

The Min-IBI design [14], [15] minimizes the IBI power subject to the constraint that the desired signal energy is held constant. We briefly review the design, then demonstrate how to simplify its computation.

Define the IBI weighting matrices as [15]

$$\mathbf{Q}_{ibi}(\Delta) = \mathrm{diag}[\Delta, \cdots, 2, 1, \underbrace{0, \cdots, 0}_{\nu+1}, 1, 2, \cdots, L_c - \nu - \Delta], \tag{7.18}$$

Table 7.1: Computational complexity of various MSSNR implementations. MACs are real multiply-and-accumulates and adds are real additions (or subtractions).

	brute force	**Wu, et al. [111]**
step	MACs	MACs
$\mathbf{C}$	0	0
$\mathbf{B}(\Delta_{min})$	$\tilde{L}_w^2(\nu+1)$	$\tilde{L}_w(L_w+\nu)$
$\mathbf{A}(\Delta_{min})$	$\tilde{L}_w^2(L_h-\nu)$	$\tilde{L}_w(L_c-\nu)$
Each $\mathbf{B}(\Delta)$	$\tilde{L}_w^2(\nu+1)$	$\tilde{L}_w(L_w+\nu)$
Each $\mathbf{A}(\Delta)$	$\tilde{L}_w^2(L_h-\nu)$	$\tilde{L}_w(L_c-\nu)$
Total:	$\tilde{L}_w^2\tilde{L}_h N_\Delta$	$\tilde{L}_w(L_w+L_c)N_\Delta$
Example:	267,911,168	9,369,696

	proposed	
step	MACs	adds
$\mathbf{C}$	$\tilde{L}_h\tilde{L}_w$	0
$\mathbf{B}(\Delta_{min})$	$\tilde{L}_w(L_w+\nu)$	0
$\mathbf{A}(\Delta_{min})$	0	$\tilde{L}_w^2$
Each $\mathbf{B}(\Delta)$	$2L_w+\nu+1$	0
Each $\mathbf{A}(\Delta)$	0	$\tilde{L}_w^2$
Total:	$\left(2\tilde{L}_w+\nu\right)(N_\Delta-1)+\tilde{L}_h\tilde{L}_w$	$\tilde{L}_w^2 N_\Delta$
Example:	66,850	523,264

$$\widetilde{\mathbf{Q}}_{ibi}(\Delta) = \mathrm{diag}\left[\Delta,\cdots,2,1,1,2,\cdots,L_c-\nu-\Delta\right] \quad (7.19)$$

The Min-IBI design uses $\mathbf{Q}$ to suppress the taps of the effective channel outside of the desired window, with linearly increasing weights at further distances from the edge of the window. Thus, in the formulation of (7.4),

$$\begin{aligned} \mathbf{A} &= \alpha \mathbf{H}^H \mathbf{Q}_{ibi} \mathbf{H} + (1-\alpha)\mathbf{R}_n \\ &= \alpha \mathbf{H}_{wall}^H \widetilde{\mathbf{Q}}_{ibi} \mathbf{H}_{wall} + (1-\alpha)\mathbf{R}_n. \end{aligned} \tag{7.20}$$

The optimization problem is then given by[3] (7.3), (7.5), (7.18), and (7.20). The solution is the generalized eigenvector of $(\mathbf{A}, \mathbf{C})$ corresponding to the smallest generalized eigenvalue [35].

The only differences between the MSSNR design [67] and the Min-IBI design are that the MSSNR design sets $\widetilde{\mathbf{Q}} = \mathbf{I}$, the identity matrix, and the MSSNR design assumes $\alpha = 1$. The MSSNR design places equal weight on all taps in the channel tails, even though the more distant taps contribute more to the IBI. However, with $\widetilde{\mathbf{Q}} = \mathbf{I}$, the $\mathbf{A}$ matrix for delay $\Delta + 1$ can be obtained almost entirely from the $\mathbf{A}$ matrix for delay Δ [55], as discussed in Section 7.1. The method of performing this feat for the Min-IBI matrix is not so apparent, and this is the focus of the remainder of this section.

Define the error matrices

$$\widehat{\mathbf{E}}(\Delta) = \operatorname{diag}\left[\mathbf{1}_{1\times\Delta}, \mathbf{0}_{1\times\nu}, -\mathbf{1}_{1\times(L_c+1-\nu-\Delta)}\right] \tag{7.21}$$

$$\begin{aligned} \mathbf{E}(\Delta) &= \mathbf{H}^H \; \widehat{\mathbf{E}}(\Delta) \; \mathbf{H} \\ &= \mathbf{H}_{wall,\nu}^H(\Delta) \; \widetilde{\mathbf{H}}_{wall,\nu}(\Delta) \end{aligned} \tag{7.22}$$

where $\widetilde{\mathbf{H}}_{wall,\nu} = \left[\mathbf{H}_1^T, -\mathbf{H}_2^T\right]^T$, and the subscript ν denotes the fact that these particular matrices only eliminate ν rows from $\mathbf{H}$ rather than $\nu + 1$. With this

[3] Celebi's Min-IBI design [15] uses $\alpha = 1$ and $\mathbf{C} = \mathbf{H}_{win}^H \mathbf{H}_{win}$ rather than $\mathbf{C} = \mathbf{H}^H \mathbf{H}$, but we prefer Tkacenko's formulation [98].

definition of the error matrix $\mathbf{E}$, we have

$$\mathbf{Q}_{ibi}(\Delta+1) = \mathbf{Q}_{ibi}(\Delta) + \widehat{\mathbf{E}}(\Delta+1), \tag{7.23}$$

$$\mathbf{A}(\Delta+1) = \mathbf{A}(\Delta) + \mathbf{E}(\Delta+1). \tag{7.24}$$

where (7.24) follows from (7.23) by left- and right-multipliying (7.23) by $\mathbf{H}^H$ and $\mathbf{H}$, respectively. The intuition behind (7.24) is that $\mathbf{A}(\Delta)$ linearly weights the channel tails outside of a length-$(\nu+1)$ window, and $\mathbf{A}(\Delta)$ does the same thing for a window that is shifted over by one sample. Thus, the effect of incrementing the delay is to increment all of the weights on the channel "head" (up to tap $\Delta+1$) by one and to decrement the weights on the channel "tail" (starting at tap $\Delta+\nu$) by one.

The objective is to form an efficient update rule for $\mathbf{E}(\Delta)$, then use (7.24) to update $\mathbf{A}$. Since $\mathbf{E}(\Delta)$ is very similar to $\mathbf{H}_{wall}^H\mathbf{H}_{wall}$, we can use techniques similar to those used for the MSSNR design [55], [111] in Section 7.1. For $i \geq j$, element (i,j) of $\mathbf{E}(\Delta)$ is given by

$$[\mathbf{E}(\Delta)]_{(i,j)} = \sum_{l=0}^{\Delta-1-i} h_l^* h_{(l+i-j)} - \sum_{l=\Delta+\nu-j}^{L_h} h_{(l+j-i)}^* h_l. \tag{7.25}$$

Throughout, matrix and vector indexing starts at zero, rather than at one. By incrementing Δ,

$$\begin{aligned}[\mathbf{E}(\Delta+1)]_{(i,j)} &= \sum_{l=0}^{\Delta-1-(i-1)} h_l^* h_{(l+(i-1)-(j-1))} - \sum_{l=\Delta+\nu-(j-1)}^{L_h} h_{(l+(j-1)-(i-1))}^* h_l \\ &= [\mathbf{E}(\Delta)]_{(i-1,j-1)}\,. \end{aligned} \tag{7.26}$$

In block form,

$$\boxed{[\mathbf{E}(\Delta+1)]_{(1:L_w,1:L_w)} = [\mathbf{E}(\Delta)]_{(0:L_w-1,0:L_w-1)}} \tag{7.27}$$

By keeping Δ fixed and incrementing i and j instead,

$$\begin{aligned}
[\mathbf{E}(\Delta)]_{(i+1,j+1)} &= \sum_{l=0}^{\Delta-1-i-1} h_l^* h_{(l+i-j)} - \sum_{l=\Delta+\nu-j-1}^{L_h} h_{(l+j-i)}^* h_l \\
&= [\mathbf{E}(\Delta)]_{(i,j)} - h_{(\Delta-1-i)}^* h_{(\Delta-1-j)} - h_{(\Delta+\nu-1-i)}^* h_{(\Delta+\nu-1-j)} \qquad (7.28)
\end{aligned}$$

Or, equivalently,

$$\boxed{[\mathbf{E}(\Delta)]_{(i,j)} = [\mathbf{E}(\Delta)]_{(i+1,j+1)} + h_{(\Delta-1-i)}^*\ h_{(\Delta-1-j)} + h_{(\Delta+\nu-1-i)}^*\ h_{(\Delta+\nu-1-j)}} \qquad (7.29)$$

By using (7.27) we can obtain all of $\mathbf{E}(\Delta+1)$ except the first row and column from $\mathbf{E}(\Delta)$. The first column of $\mathbf{E}(\Delta+1)$ can be efficiently obtained via (7.29), and its first row can be obtained by symmetry. Finally, $\mathbf{E}(\Delta_{min})$ must be computed explicitly, but this can also be done efficiently using (7.29). An outline of an efficient Min-IBI algorithm is given in Figure 7.2.

Note that $\mathbf{Q}_{ibi}$ may assign very large weights to the extreme edges of the effective channel impulse response. If the channel estimate is imperfect, these large weights will amplify the errors. The solution proposed in [15] is to limit the maximum weight value by redefinining the weighting matrix as

$$\begin{aligned}
\overline{\mathbf{Q}}_{ibi}(\Delta) &= \min\{\mathbf{Q}_{ibi}(\Delta), \gamma\} \\
&= \mathrm{diag}[\underbrace{\gamma, \gamma, \cdots, \gamma}_{\Delta-\gamma+1}, \gamma-1, \cdots, 2, 1, \underbrace{0, \cdots, 0}_{\nu}, 1, 2, \cdots, \gamma-1, \underbrace{\gamma, \gamma, \cdots, \gamma}_{L_c-\nu-\Delta-\gamma+1}]
\end{aligned} \qquad (7.30)$$

Then the error weighting matrix of (7.21) becomes

$$\widehat{\mathbf{E}}(\Delta) = \mathrm{diag}\left[\mathbf{0}_{1\times(\Delta-\gamma)}, \mathbf{1}_{1\times\gamma}, \mathbf{0}_{1\times\nu}, -\mathbf{1}_{1\times\gamma}, \mathbf{0}_{1\times(L_c-\nu-\Delta-\gamma+1)}\right]. \qquad (7.31)$$

In this case, (7.24) and (7.27) still hold, but (7.29) requires four update terms rather than two.

1. For Δ_{min}, compute $\mathbf{A}$, $\mathbf{C}$ (using the efficient methods of [111]), and $\mathbf{E}$ (using (7.29)).

2. Solve $\mathbf{Aw} = \lambda\mathbf{Cw}$ for the generalized eigenvector corresponding to the smallest eigenvalue, as in [67].

3. For $\Delta = \Delta_{min} + 1 : \Delta_{max}$, do the following:

   (a) $[\mathbf{E}]_{(1:L_w,1:L_w)} = [\mathbf{E}]_{(0:L_w-1,0:L_w-1)}$

   (b) $[\mathbf{E}]_{(0:L_w-1,0)} = [\mathbf{E}]_{(1:L_w,1)}$
   $+ h(\Delta - 1) \cdot h(\Delta - 1 - [0 : L_w - 1])^*$
   $+ h(\Delta + \nu - 1) \cdot h(\Delta + \nu - 1 - [0 : L_w - 1])^*$

   (c) Compute $[\mathbf{E}]_{(L_w,0)}$ from (7.22)

   (d) $[\mathbf{E}]_{(0,1:L_w)} = [\mathbf{E}]^H_{(1:L_w,0)}$

   (e) $\mathbf{A} = \mathbf{A} + \mathbf{E}$

   (f) Solve $\mathbf{Aw} = \lambda\mathbf{Cw}$ for the generalized eigenvector corresponding to the smallest eigenvalue.

   (g) If this delay produces a smaller λ than the previous delay, save $\mathbf{w}$.

Figure 7.2: Fast Min-IBI TEQ design algorithm.

7.3 Efficient MDS computation

The MDS design [92] is similar to the Min-IBI design in that is also uses a diagonal weighting matrix. However, the main diagonal is quadratic rather than piecewise linear. The result is that the delay spread of the effective channel is minimized, though the size of the cyclic prefix is not explicitly taken into account.

To formally describe the MDS design, first define the weighting matrix $\mathbf{Q}_{mds}$

as

$$\mathbf{Q}_{mds}(\eta) = \left[\eta^2, (\eta-1)^2, \cdots, 9, 4, 1, 0, 1, 4, 9, \cdots, (L_c-\eta)^2\right]. \tag{7.32}$$

Then the MDS design is given by (7.3), (7.4), (7.5), and (7.32). The parameter η is the desired centroid of the effective channel. It replaces the delay parameter Δ as the parameter to be searched over.

Since $\mathbf{C}$ is not a function of η, it only has to be computed once. Moreover, it is Hermitian and Toeplitz, and thus is easily computed. On the other hand, $\mathbf{A}(\eta)$ must be computed once per value of η if a full search is made. We now propose an efficient recursive method for computing $\mathbf{A}(\eta+1)$ from $\mathbf{A}(\eta)$. Define the first-order error matrices

$$\widehat{\mathbf{E}}_1(\eta) = \mathrm{diag}[(2\eta-1), (2\eta-3), \cdots, 3, 1, -1, -3, \cdots, (-2(L_c-\eta)-1)], \tag{7.33}$$

$$\mathbf{E}_1(\eta) = \mathbf{H}^H\ \widehat{\mathbf{E}}_1(\eta)\ \mathbf{H}. \tag{7.34}$$

The difference of two monic quadratic polynomials is a linear polynomial, so we have

$$\mathbf{Q}(\eta+1) = \mathbf{Q}(\eta) + \widehat{\mathbf{E}}_1(\eta+1), \tag{7.35}$$

$$\mathbf{A}(\eta+1) = \mathbf{A}(\eta) + \mathbf{E}_1(\eta+1), \tag{7.36}$$

where (7.36) follows from (7.35) by left- and right-multiplying by $\mathbf{H}^H$ and $\mathbf{H}$, respectively. As with the Min-IBI design, we would like to effciently update $\mathbf{E}_1$ and use it to update $\mathbf{A}$. However, $\mathbf{E}_1$ is now linear rather than peicewise constant. The procedure can be iterated by implicitly defining the second-order error matrices $\widehat{\mathbf{E}}_2$ and $\mathbf{E}_2$ such that we will have

$$\widehat{\mathbf{E}}_1(\eta+1) = \widehat{\mathbf{E}}_1(\eta) + \widehat{\mathbf{E}}_2(\eta+1), \tag{7.37}$$

$$\mathbf{E}_1(\eta+1) = \mathbf{E}_1(\eta) + \mathbf{E}_2(\eta+1). \tag{7.38}$$

1. For η_{min}, compute $\mathbf{A}$, $\mathbf{C}$ (exploiting the Hermitian Toeplitz structure), and $\mathbf{E}_1$.
2. Solve $\mathbf{Aw} = \lambda\mathbf{Cw}$ for the generalized eigenvector corresponding to the smallest eigenvalue, as in [67].
3. For $\eta = \eta_{min} + 1 : \eta_{max}$, do the following:
   (a) $\mathbf{E}_1 = \mathbf{E}_1 + 2\mathbf{C}$ (where multiplication by 2 is performed by a shift in binary representation)
   (b) $\mathbf{A} = \mathbf{A} + \mathbf{E}_1$
   (c) Solve $\mathbf{Aw} = \lambda\mathbf{Cw}$ for the generalized eigenvector corresponding to the smallest eigenvalue.
   (d) If this delay produces a smaller λ than the previous delay, save $\mathbf{w}$.

Figure 7.3: Fast MDS TEQ design algorithm.

Inspection of (7.33) reveals that $\widehat{\mathbf{E}}_2 = 2\mathbf{I}_{L_c+1}$ for all η, and thus

$$\mathbf{E}_2 = \mathbf{H}^H \left(2\mathbf{I}_{L_c+1}\right) \mathbf{H} = 2\mathbf{C}. \tag{7.39}$$

Note that $\mathbf{C}$ has already been computed, and multiplication by two is simply a shift in binary representation. In summary, (7.38) is used to update $\mathbf{E}_1$, then (7.36) is used to update $\mathbf{A}$. This procedure only requires $(L_w + 1)^2$ extra memory locations and $(L_w + 1)^2$ extra additions, with the savings of not having to recompute $\mathbf{A}$ at all. For comparison, computing $\mathbf{A}$ normally takes $\mathcal{O}\left(L_w^2(L_h - \nu)\right)$ multiply-adds for *each* of the L_c+1 possible values of η. An outline of an efficient MDS algorithm is given in Figure 7.3.

The MDS penalty function leading to (7.32) is $f(n) = n^2$. Tkacenko and

Vaidyanathan [97] also considered a linear MDS penalty function, $f(n) = |n|$. This leads to a Min-IBI design that uses $\nu = 0$. In Chapter 8 we will refer to the original MDS design with a quadratic penalty function as MDS-Q, and the alternate MDS design with a linear penalty function as MDS-L.

7.4 Methods for arbitrary polynomial weighting functions

To generalize to arbitrary polynomial weighting functions, let the diagonal elements of $\mathbf{Q}$ be defined by an M^{th} order polynomial. The coefficients may be defined peicewise over the regions $[0, \Delta - 1]$, $[\Delta, \Delta + \nu]$, and $[\Delta + \nu + 1, L_c]$, or a single polynomial may be used. For example, the MSSNR TEQ uses three zeroth order (constant) polynomials, the Min-IBI TEQ uses three first order (linear) polynomials, and the MDS design uses a single second order (quadratic) polynomial.

To generalize (7.24), (7.36), and (7.38), up to M error matrices $\mathbf{E}_m$, $1 \leq m \leq M$, may be needed for efficient updating of the $\mathbf{A}$ matrix. Each error matrix will be of size $(L_w + 1) \times (L_w + 1)$, but it may turn out that some error matrices are independent of the delay and/or already computed, as with $\mathbf{E}_2$ in the MDS technique. Moreover, since $\mathbf{A}$ is Hermitian, so is each $\mathbf{E}_m$, and only half of the coefficients must be stored. Thus, the additional memory requirements are at most $\frac{M}{2}(L_w + 1)(L_w + 2)$ storage words. The computational savings of not recomputing $\mathbf{A}$ are $\mathcal{O}\left(L_w^2(L_h - \nu)\right)$ per delay for up to $L_c + 1$ delays, which more than balances out the extra memory use for reasonably small values of M.

7.5 Efficient MMSE computation

When the input signal is white, the MMSE design can be written as the MSSNR design, with the sole difference of the use of $\alpha = \frac{1}{2}$ rather than $\alpha = 1$ in (7.4); c.f.

(2.18). For general input signal statistics, the complexity reduction techniques are somewhat different. However, the end result is similar: $\mathbf{R}(\Delta+1)$ can be computed from $\mathbf{R}(\Delta)$ almost for free.

Computing $\mathbf{R}(\Delta+1)$ from $\mathbf{R}(\Delta)$

Recall that for the MMSE design, we must compute

$$\mathbf{R}(\Delta) = \mathbf{R}_x - \mathbf{R}_{xr}\mathbf{R}_r^{-1}\mathbf{R}_{rx}, \tag{7.40}$$

where

$$\begin{aligned}
\mathbf{R}_x &= \mathrm{E}\left[\mathbf{x}_k^* \, \mathbf{x}_k^T\right], & (7.41)\\
\mathbf{R}_{rx} &= \mathrm{E}\left[\mathbf{r}_k^* \, \mathbf{x}_k^T\right], & (7.42)\\
\mathbf{x}_k &= \Big[x(k-\Delta), \quad \cdots, \quad x(k-\Delta-\nu)\Big]^T, & (7.43)\\
\mathbf{r}_k &= \Big[r(k), \quad \cdots, \quad x(k-L_w)\Big]^T. & (7.44)
\end{aligned}$$

Note that $\mathbf{R}_x$ does not depend on Δ, and that it is Toeplitz. Thus,

$$\begin{aligned}
[\mathbf{R}_x(\Delta+1)]_{(0:\nu-1,0:\nu-1)} &= [\mathbf{R}_x(\Delta)]_{(0:\nu-1,0:\nu-1)}\\
&= [\mathbf{R}_x(\Delta)]_{(1:\nu,1:\nu)}. \qquad (7.45)
\end{aligned}$$

Let $\mathbf{G}(\Delta) = \mathbf{R}_{xr}\mathbf{R}_r^{-1}\mathbf{R}_{rx}$. Observing that

$$[\mathbf{R}_{rx}(\Delta+1)]_{(0:L_w,0:\nu-1)} = [\mathbf{R}_{rx}(\Delta)]_{(0:L_w,1:\nu)}, \tag{7.46}$$

and that $\mathbf{R}_{xr} = \mathbf{R}_{rx}^H$, we see that

$$[\mathbf{G}(\Delta+1)]_{(0:\nu-1,0:\nu-1)} = [\mathbf{G}(\Delta)]_{(1:\nu,1:\nu)}. \tag{7.47}$$

Combining (7.40), (7.45), and (7.47),

$$\boxed{[\mathbf{R}(\Delta+1)]_{(0:\nu-1,0:\nu-1)} = [\mathbf{R}(\Delta)]_{(1:\nu,1:\nu)}} \tag{7.48}$$

The matrix $\mathbf{R}_r$ is Hermitian and Toeplitz. However, the inverse of a Toeplitz matrix is, in general, not Toeplitz [36]. This means that $\mathbf{R}(\Delta)$ has no further structure that can be easily exploited, so the first row and column of $\mathbf{R}(\Delta+1)$ cannot be obtained from the rest of $\mathbf{R}(\Delta+1)$ using the method in (7.16) [111]. Even so, (7.48) allows us to obtain most of the elements of each $\mathbf{R}(\Delta)$ for free, so only $\nu+1$ elements must be computed rather than $(\nu+1)(\nu+2)/2$ elements. In ADSL, $\nu=32$; in VDSL, ν can range up to 512; and in DVB (2K mode), ν can range up to 2048. Thus, the proposed method reduces the complexity of calculating $\mathbf{R}(\Delta)$ by factors of 17, 257, and 1025 (respectively) for these standards.

Complexity comparison

Table 7.2 shows the (approximate) computational requirements of the "brute force" approach and the proposed approach for computing the matrices $\mathbf{R}(\Delta), \Delta \in \{\Delta_{min}, \cdots, \Delta_{max}\}$. The "example" line shows the required complexity for computing the $\mathbf{R}(\Delta)$ matrices using each method for the same parameter values as the example in Table 7.1. The proposed method yields a decrease in complexity by a factor of the channel shortener length over two, which in this case is a factor of 16.

7.6 Intelligent eigensolver initialization

Let $\mathbf{w}(\Delta)$ be the TEQ for a given delay. If we were to increase the allowable filter length by 1, then it follows that

$$\hat{\mathbf{w}}(\Delta+1) = z^{-1}\mathbf{w}(\Delta) = \left[0, \mathbf{w}^T(\Delta)\right]^T \tag{7.49}$$

Table 7.2: Computational complexity of various MMSE implementations. MACs are real multiply-and-accumulates.

	brute force	**proposed**
step	MACs	MACs
$\mathbf{R}(\Delta_{min})$	$\tilde{L}_w^3$	$\tilde{L}_w^3$
Each $\mathbf{R}(\Delta)$	$\tilde{L}_w^3$	$2\tilde{L}_w^2$
Total:	$N_\Delta \tilde{L}_w^3$	$\tilde{L}_w^2 \left(2(N_\Delta - 1) + \tilde{L}_w\right)$
Example:	16,744,448	1,077,248

should be a near-optimum solution, since it produces the same value of the cost function (SSNR, MSE, IBI, or delay spread) as for the previous delay. Experience suggests that the TEQ coefficients are small near the edges, so the last tap can be removed without drastically affecting the performance. Therefore,

$$\boxed{\hat{\mathbf{w}}(\Delta+1) = \left[0, \left[\mathbf{w}^T(\Delta)\right]_{(0:L_w-1)}\right]^T} \tag{7.50}$$

is a fairly good solution for the delay $\Delta + 1$, so this should be the initialization for the generalized eigenvector solver for the next delay. Similarly, for the MMSE TIR,

$$\boxed{\hat{\mathbf{b}}(\Delta+1) = \left[\left[\mathbf{b}^T(\Delta)\right]_{(1:\nu)}, 0\right]^T} \tag{7.51}$$

should be the initialization for the eigenvector solver for the next delay. These initializations should decrease the number of iterations of the eigensolver that are required for it to converge, though the amount of complexity reduction that is achieved will depend greatly on the channel.

Chapter 8

Algorithm Comparisons and Simulations

"What! hast thou been long blind, and now restor'd?"

– William Shakespeare, *Henry VI part 2*, Act II, Scene i.

The goals of this chapter[1] are twofold: first, to compare the complexity of the different algorithms discussed in this thesis (both adaptive and non-adaptive) to the algorithms in the literature; and second, to compare the convergence speeds and asymptotic performance of these algorithms.

8.1 Complexity comparison

Complexity is in general difficult to measure exactly. Factors influencing the complexity include how clever one is at reusing computations, how much extra memory is available for storing previous computations rather than reusing them, whether the processing is performed in fixed-point or floating-point arithmetic, and whether or not certain operations (e.g. FFTs) can be done in dedicated hardware. Thus, the complexity results shown here are at best approximate.

We assume that the channel shortening algorithms will usually be implemented on a fixed-point DSP, and as such, the basic unit of complexity is the multiply and accumulate ("MAC"). Divisions and square roots are far more costly and will be listed separately. If the term "flops" (floating-point operations) is used, this means a mix of divisions and multiplications. If extra FFTs are required (other than the

[1]Some material in this chapter has been previously published. © 2004 IEEE. Reprinted, with permission, from [8], [9], [54], [53], [60], [61], [56], [62], [63], [64], and [65]. © 2004 Hindawi Publishing Corporation. Reprinted, with permission, from [55].

Table 8.1: Adaptive TEQ/PTEQ algorithm complexity comparison.

algorithm	**MACs/ update**	$(\div, \sqrt{\ })$ / **update**	**FFTs/ update**	**updates/ symbol**
MERRY [53]	$2\tilde{L}_w$	1	0	1
FRODO [64]	$2 \lvert S_f \rvert \tilde{L}_w$	0	0	1
SAM [9]	$4\tilde{L}_w (L_c - \nu)$	1	0	$N + \nu$
TOLKIEN	$4\tilde{L}_w \nu$	1	0	1
PT-LMS [57]	$2N_c\tilde{L}_w$	0	0	1
PT-DDLMS [57]	$2N_c\tilde{L}_w$	0	0	1
PT-CMA [57]	$2N_c\tilde{L}_w$	0	0	1
MMSE [30]	$3\left(\tilde{L}_w + \nu\right)$	1	0	$N + \nu$
Chow [17]	$6N$	1	5	1
CNA [23], [88]	$N\left(N_z + \tilde{L}_w\right)$	1	0	1

one used for demodulation), then they will be listed separately. The memory use will be given in terms of "words," with each filter tap or data sample counting as one "word."

Table 8.1 compares the computational complexity of various adaptive TEQ and PTEQ algorithms, and Table 8.2 lists several qualitative properties of the same algorithms. "Window size" indicates the length of the window of non-zero taps that is sought by the algorithm, the property "symbol synch" refers to whether or not symbol (block) synchronization must be performed before equalizer adaptation, and "false minima" indicates whether or not there are local minima in the cost function that the given algorithm descends.

The computational complexity for several MMSE and MSSNR designs is sum-

Table 8.2: Adaptive TEQ/PTEQ algorithm properties.

algorithm	**window size**	**trained or blind**	**symbol synch?**	**false minima?**
MERRY [53]	ν	blind	yes	no
FRODO [64]	$\in \{1, \cdots, \nu\}$	blind	yes	no
SAM [9]	$\nu + 1$	blind	no	yes
TOLKIEN	$\nu + 1$	trained	yes	no
PT-LMS [57]	$\nu + 1$	trained	yes	no
PT-DDLMS [57]	$\nu + 1$	blind	yes	no
PT-CMA [57]	$\nu + 1$	blind	yes	yes
MMSE [30]	$\nu + 1$	trained	yes	no
Chow [17]	$\nu + 1$	trained	no	yes
CNA [23], [88]	1	blind	yes	no

marized in Table 8.3. It is assumed that the most eficient techniques are used in all cases, such as applying (7.48) during matrix computations and not repeating matrix inversions at each delay unless the matrix is delay-dependent. The MMSE, MSSNR, Min-IBI, and MDS methods assume that the methods in Chapter 7 are used. The Min-ISI design has a slightly different structure that cannot make use of such tehniques, hence it has a higher complexity. Note that the complexity if the iterative and adaptive designs is given *per update*, and the total complexity must consider the number of iterations needed.

Table 8.3: Complexity of MMSE and MSSNR implementations.

Optimal design	Complexity per delay
MMSE, UTC on $\mathbf{b}$ [4]	$\mathcal{O}\left(\frac{1}{3}\nu^3 + \nu^2 + 2\nu L_w + L_w^2\right)$ MACs
MMSE, UTC on $\mathbf{w}$ [115]	$\mathcal{O}\left(\frac{1}{3}\nu^3 + \nu^2 + \nu L_w\right)$ MACs
MMSE, UNC on $\mathbf{b}$ [30]	$\mathcal{O}\left(\nu^2 + 2\nu L_w + 2L_w^2\right)$ MACs
MMSE, UNC on $\mathbf{w}$ [115]	$\mathcal{O}\left(\nu^2 + \nu L_w + L_w^2\right)$ MACs
Sym-MMSE, UNC on $\mathbf{b}$ [55]	$\mathcal{O}\left(\nu^2 + \frac{3}{2}\nu L_w + \frac{5}{4}L_w^2\right)$ MACs
MSSNR [67]	$\mathcal{O}\left(L_w^3\right)$ flops
Noise-limited MSSNR [98]	$\mathcal{O}\left(L_w^3\right)$ flops
Sym-MSSNR [60]	$\mathcal{O}\left(\frac{1}{4}L_w^3\right)$ flops
Min-ISI [7]	$\mathcal{O}\left((5NL_w + L_w^3)\right)$ flops
Min-IBI [15]	$\mathcal{O}\left((L_w^3)\right)$ flops
MDS [92]	$\mathcal{O}\left((L_w^3)\right)$ flops
Iterative/adaptive design	**Complexity per update per delay**
Adaptive MMSE [30]	$\mathcal{O}\left(4\nu + 2L_w\right)$ MACs + renormalization
MSSNR via power method (2.5) [35]	$\mathcal{O}\left(3L_w^2\right)$ MACs
MSSNR via iteration of (2.6) [16]	$\mathcal{O}\left(2L_w^2\right)$ MACs
MERRY/FRODO	$\mathcal{O}\left(4L_w\right)$ MACs
Nafie & Gatherer [75]	$\mathcal{O}\left(2L_w\right)$ MACs + renormalization

8.2 Simulations

This section simulates the blind, adaptive algorithms derived in this thesis, and compares their asymptotic performance to the performance of popular trained, non-adaptive algorithms. The simulations will be broken down into six self-contained examples.

- Example 1: a comparison of the effective channels after being shortened using various TEQ algorithms.
- Example 2: a study of the convergence rates of MERRY, Sym-MERRY, and FRODO.
- Example 3: a study of the convergence rate of SAM.
- Example 4: a study of the convergence rates of PT-CMA and PT-LMS.
- Example 5: a performance assessment of MIMO channel shortening.
- Example 6: a performance comparison of the reduced complexity symmetry-based algorithms of Chapter 6 and their non-symmetric counterparts.

The simulation parameters will be drawn from two settings: (i) a downstream ADSL setting, which is a pont-to-point baseband link; and (ii) a wireless broadcast system with parameters similar to the IEEE 802.11a and HIPERLAN/2 wireless LAN standards. Setting (i) will use an FFT size of $N = 512$, a CP of length $\nu = 32$, and the transmission channels will be the CSA test loops [7], available at [6]. The CSA loops are synthetic models of twisted pair telephone lines. They are real, with energy in about 200 taps. The DSL performance metric is the achievable bit rate for a fixed probability of error,

$$R = \frac{1}{T} \sum_i \log_2 \left(1 + \frac{SNR_i}{\Gamma} \right), \tag{8.1}$$

where SNR_i is the signal to interference and noise ratio in frequency bin i, Γ is the SNR gap, and $T = 246.4\mu s$ is the symbol duration. We assume a 6 dB margin and 4.2 dB coding gain. For more details on calculating the bit rate, refer to [7].

Setting (ii) will use an FFT size of $N = 64$ and a CP of length $\nu = 16$. The transmission channels will be modelled as consisting of three parts [81]: $\mathbf{h}_{local,1}$,

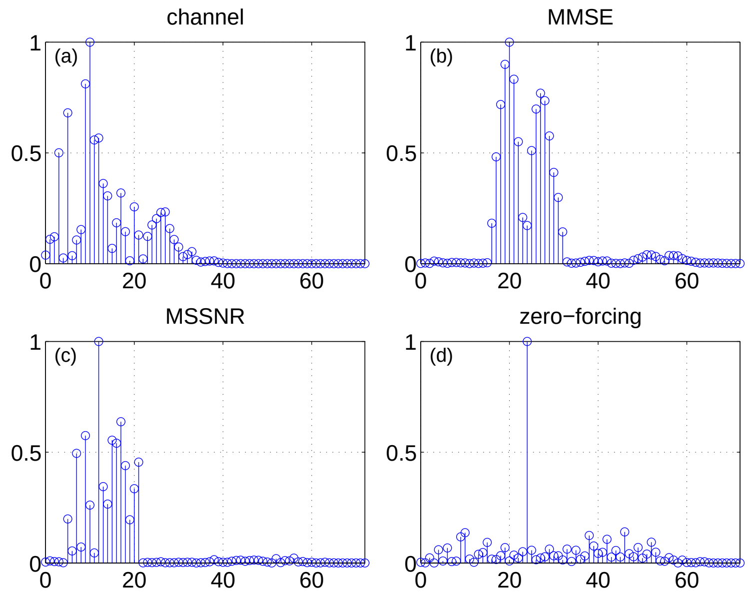

Figure 8.1: Example 1: the plots show the shortened channel impulse response magnitudes using (a) no TEQ, (b) an MMSE TEQ, (c) an MSSNR TEQ, and (d) a (zero-forcing) MSSNR TEQ with a window of size 1. Here, $\nu = 16$.

scatterers near the transmitter; $\mathbf{h}_{mid}$, remote scatterers; and $\mathbf{h}_{local,2}$, scatterers near the receiver. The channel is then

$$\mathbf{h} = \mathbf{h}_{local,1} \star \mathbf{h}_{mid} \star \mathbf{h}_{local,2}, \tag{8.2}$$

where $\star$ denotes convolution. $\mathbf{h}_{mid}$ consists of 32 uncorrelated Rayleigh fading taps with an exponential delay profile, and $\mathbf{h}_{local,1}$ and $\mathbf{h}_{local,2}$ each consist of 6 uncorrelated Rayleigh fading taps with a uniform delay profile.

8.2.1 Example 1: comparison of optimal (non-adaptive) designs

The purpose of this example is to compare the asymptotic (i.e. non-adaptive) performance of MERRY and FRODO to other TEQ designs. The TEQs each

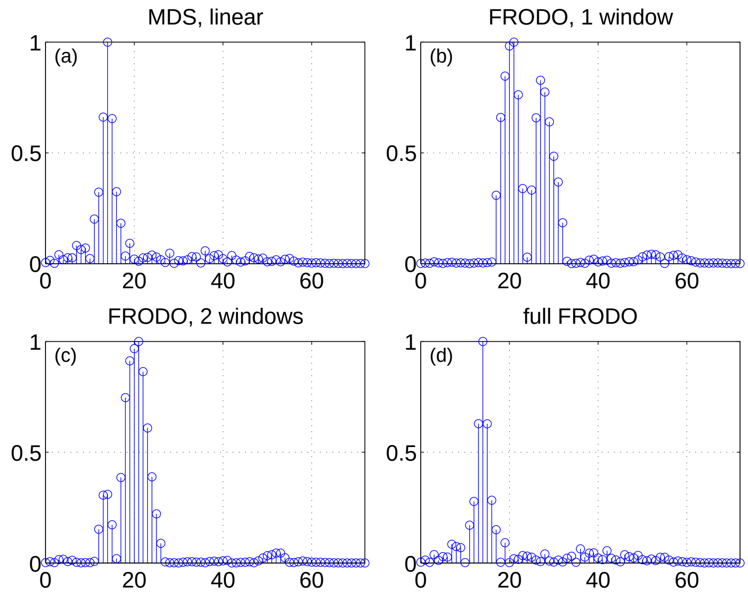

Figure 8.2: Example 1: the plots show the shortened channel impulse response magnitudes using (a) an MDS TEQ with linear weights, (b) a MERRY TEQ, (c) a FRODO TEQ with two windows, and (d) a "full" FRODO TEQ with $\nu = 16$ windows.

have 32 taps, the SNR is 20 dB (AWGN), the environment is setting (ii) described above, and the channel is as in (8.2). Figs. 8.1 and 8.2 show the shortened channel impulse responses magnitudes using various designs, for the channel realization shown in Figure 8.1(a). The "full" FRODO impulse response is quite similar to the linear MDS impulse response (rather than the zero-forcing impulse response), FRODO with one window has characteristics like the MSSNR design, and FRODO with two windows has similar characteristics but is slightly narrower. Figure 8.3 shows the shortening SNR [67], the inverse of the MSE [30], and the inverse of the inter-block interference (IBI) [14], averaged over 10000 channel realizations and normalized relative to the largest (i.e. best) value obtained from the 8 designs. The FRODO cost function with 1 window (i.e. MERRY) performs much like the

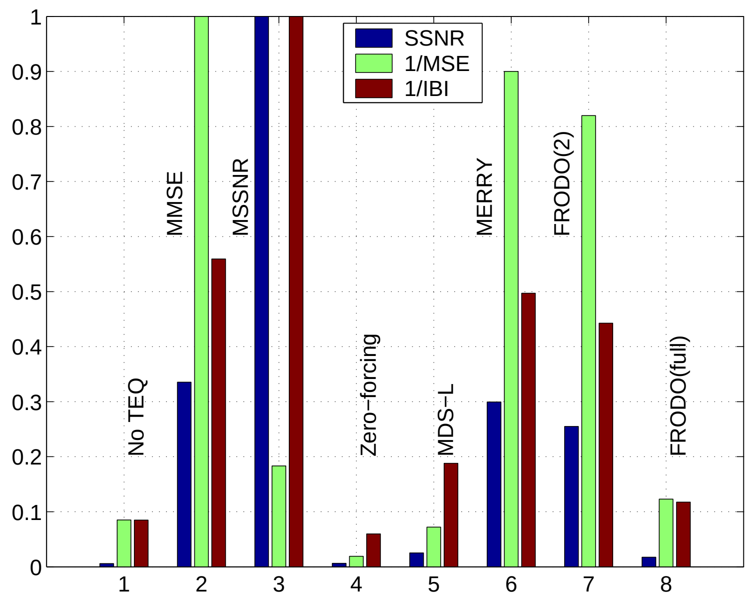

Figure 8.3: Performance of various shortened channels for example 1. The shortening SNR [67], the inverse of the MSE [30], and the inverse of the inter-block interference [14] were averaged over 10000 channel realizations and normalized relative to the largest (i.e. best) value obtained from the 8 TEQ designs.

MMSE design. The use of two windows for FRODO only slightly degrades the performance. Hence, we may use two windows in the adaptive version of FRODO without fear of significantly affecting the asymptotic performance. However, the use of all 16 windows causes the FRODO TEQ to achieve performance that is slightly worse than the (linear) MDS TEQ. Thus, the number of windows for FRODO should be relatively small, since we want to improve the convergence speed without adversely affecting the asymptotic solution.

Another insight gained from this example is that if the blind, non-adaptive FRODO initialization is used, then a performance comparable to the MMSE and MSSNR designs can be achieved without the need for training.

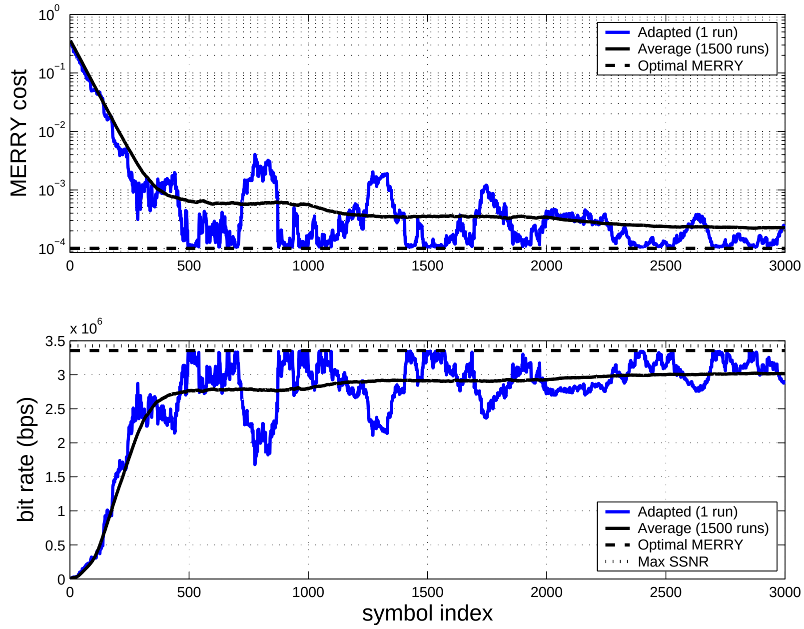

Figure 8.4: Example 2: performance vs. time for MERRY on CSA loop 4.

8.2.2 Example 2: convergence of the MERRY family

In this example, we simulate the MERRY and FRODO algorithms. First, we simulate MERRY using CSA loops 1 and 4, for an ADSL setting. The TEQ had 16 taps, the SNR was 40 dB, and initialization was a single spike. For an ADSL system, 40 dB is actually a low SNR. Figure 8.4 shows the MERRY cost and the bit rate versus symbol number for CSA loop 4. MERRY rapidly approaches the maximum SSNR solution and the optimal (closed-form, non-adaptive) MERRY solution. The convergence time (to within 75% of the optimal bit rate) is about 800 symbols, or 50 updates per tap. The step size μ was reduced by a factor of 2 after each 1000 iterations, leading to a decrease in misadjustment and an improvement in both performance metrics. This suggests that the remaining gap between the performance after 3000 iterations and the optimal MERRY performance is due in part to the large step size and in part to slow modes of convergence.

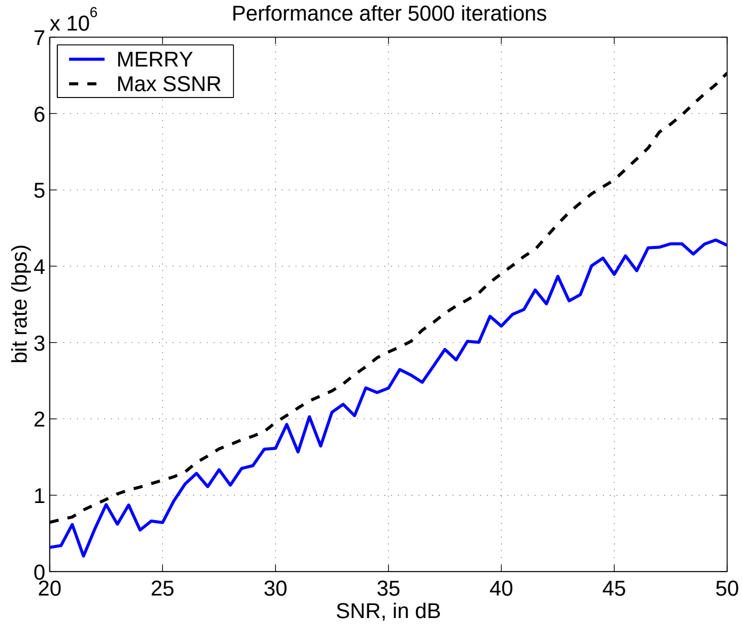

Figure 8.5: Example 2: achievable bit rate vs. SNR for MERRY on CSA loop 1.

Figure 8.5 shows a plot of the bit rate vs. SNR for CSA loop 1. The bit rate for this plot was computed by running for 5000 symbols and halving the stepsize after every 1000 symbols. At higher SNR values, midadjustment becomes more significant than noise, hence the adaptive MERRY algorithm does not achieve the optimal bit rate.

We now examine the convergence rate of FRODO using various numbers of windows. Specifically, we will compare the use of one window (i.e. MERRY) to the use of two windows. The environment is setting (ii) described above, hence the channel model is as in (8.2). The TEQ has 16 taps and the SNR is 25 dB (AWGN). For a fair comparison, both algorithms used the same step size, normalized by the number of windows $|S_f|$. The synchronization was performed blindly using the method of Section 3.8.

The performance of FRODO versus time is shown in Figure 8.6, in terms of the

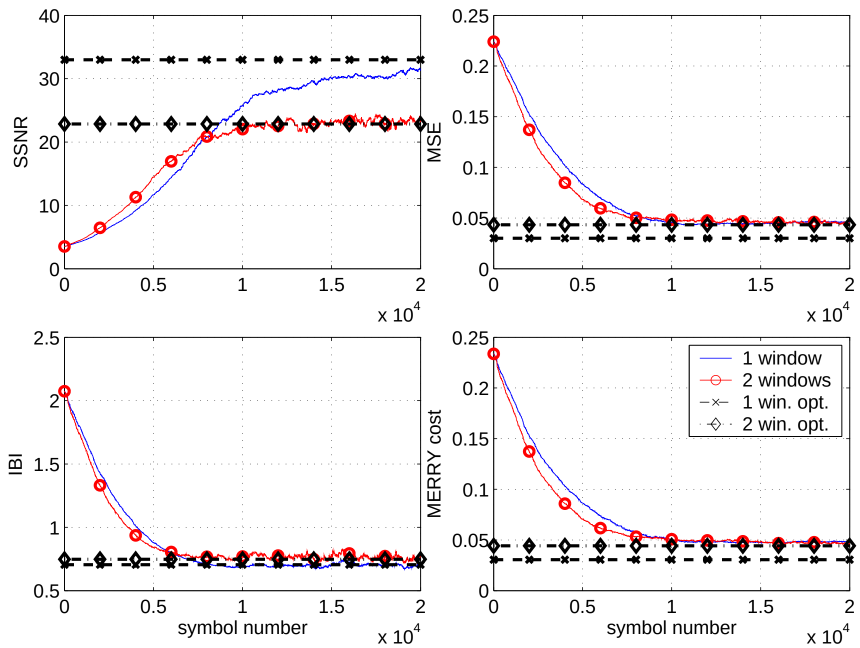

Figure 8.6: Performance metrics vs. time: SSNR (top left), MSE (top right), IBI (bottom left), and MERRY cost (bottom right).

shortening SNR [67], the MSE [30], the IBI [15], and the MERRY cost (3.4). For this example, FRODO takes about 5000 iterations to converge. This corresponds to about 300 iterations per tap. By adding additional comparisons, the algorithm converges slightly more quickly, but the quality of the final solution is not as good. Ideally, one would choose the parameters such that the final performance of the two algorithms were equal and then compare convergence rates, but that is not possible here since the use of more comparisons changes the asymptotic performance. The moral is that multiple comparisons should only be used to speed convergence or tracking, but near convergence, the algorithm should drop down to only one comparison.

8.2.3 Example 3: convergence of SAM

This section provides a numerical performance assessment of SAM in an ADSL environment. Parameters were chosen to match the standard for ADSL downstream transmission. The TEQ had 16 taps, and the channel was CSA test loop 1. The SNR was 40 dB. 75 symbols were used (of 544 samples each), and SAM used the auto-regressive implementation of (4.19) with $\alpha = 1/100$ and with the unit norm TEQ constraint. The initialization was a single spike, and the step size was 5 (such a large step size can be used because the SAM cost is very small, so the update size is still small). SAM is compared to the maximum shortening SNR solution [67], obtained using the code at [6]; and the matched filter bound (MFB) on capacity, which assumes no ICI.

Two types of noise are considered: AWGN and near-end cross-talk (NEXT) [95], which is highly colored. The NEXT was generated by exciting a coupling filter with spectrum $|H_{next}(f)|^2 = H_o \; H_{mask}(f) \; f^{(3/2)}$ with white noise [47]. The constant H_o was chosen so that the variance of the NEXT was σ_v^2, with σ_v^2 chosen to achieve the desired SNR. The filter H_{mask} is an ADSL upstream spectral mask that passes frequencies between 28 kHz and 138 kHz, since the upstream signal is the source of the NEXT for the downstream signal. The code to generate the NEXT was obtained from [6].

Figures 8.7 and 8.8 show the SAM cost and achievable bit rate versus the iteration number. Since there are 544 iterations of SAM per sample, the time scale here is different than in Figure 8.4, for example. The fact that the SAM cost is not monotonically decreasing in the first few hundred samples is because of the renormalization. After each iteration, the equalizer is divided by its norm, and this projection causes the algorithm to no longer be a gradient descent algorithm

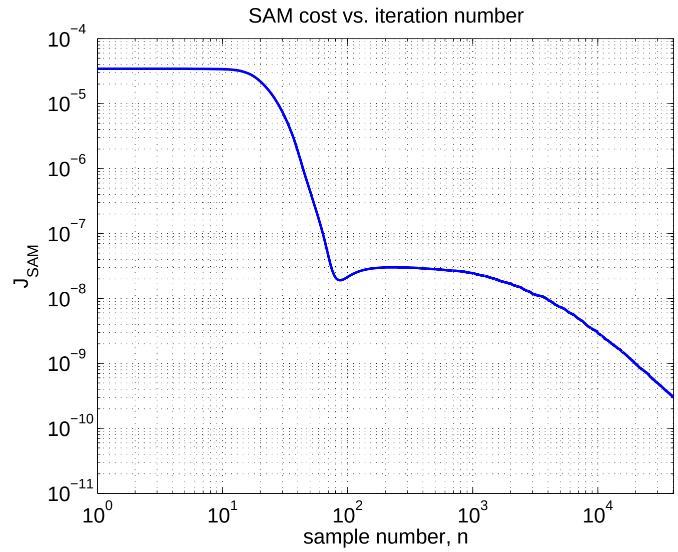

Figure 8.7: Example 3: SAM cost vs. iteration (not symbol) number, for 40 dB SNR.

(though it is approximately so). The bit rate is not monotonically increasing because the SAM cost bears no direct relation to the bit rate. At 340 iterations, SAM achieves 96% of the MFB, but then drops, and eventually rises again to 74% of the bound. The fact that the SAM cost is steadily decreasing when the bit rate decreases and then increases again indicates that the SAM minima and the bit rate maxima are not in exactly the same location. Note that SAM performs similarly for white noise and colored noise.

Figure 8.9 shows the achievable bit rate versus SNR for SAM and for the maximum shortening SNR algorithm of [67], for white noise and for NEXT. The bit rate was determined for the settings SAM arrived at after 75 DMT symbols (40800 samples). Observe that for low SNR, the performance of SAM and the MSSNR method are comparable, and the performance of SAM degrades (relatively) for high SNR. This is because when the noise is high, SAM only needs to reduce the inter-channel interference (ICI) below the noise floor, but when the SNR is 60 dB, the

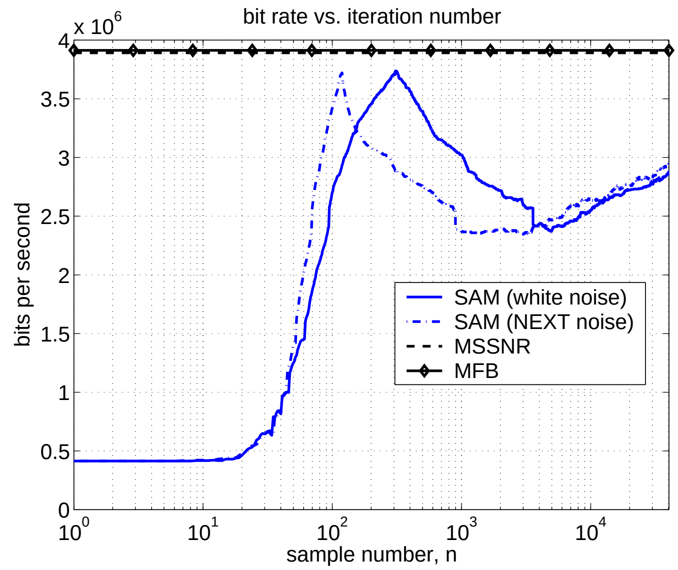

Figure 8.8: Example 3: achievable bit rate vs. iteration number (not symbol number), for 40 dB SNR. The dashed line and the diamonds correspond to the maximum SSNR solution and the matched filter bound.

excess ICI becomes more noticeable. For very low SNR (less than 15 dB for white noise, less than 25 dB for NEXT), the performance of SAM degrades, presumably due to the noiseless assumption in the derivations. However, typical SNR values for ADSL are 40 dB to 60 dB, and an SNR less than 25 dB is very unusual. Bit error rate (BER) curves are not included because for ADSL, the bit allocation on each tone is increased until the BER becomes 10^{-7}, so a BER curve would be flat as a function of SNR.

8.2.4 Example 4: convergence of PT-CMA & PT-DDLMS

This example compares the converges rates of PT-CMA and PT-DDLMS, and also considers the effect of the synchronization parameter Δ on their asymptotic performance. The modulation parameters were as in setting (ii) for a wireless LAN

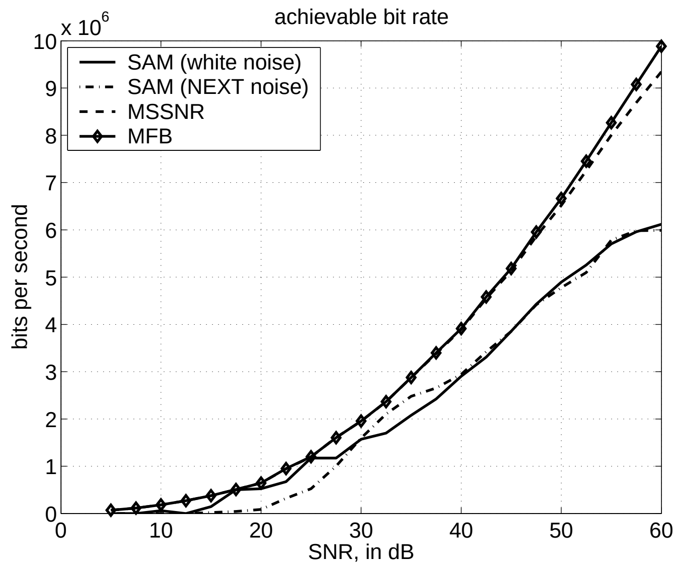

Figure 8.9: Example 3: achievable bit rate vs. SNR for SAM and the maximum SSNR algorithm for white noise and for colored noise (NEXT).

system. Since per tone algorithms do not require a white source signal, we changed 12 of the 64 carriers to be null carriers. The equalizer on each tone had 16 taps, and the noise was AWGN with an SNR of 40 dB.

Figure 8.10 shows the SNR obtained at the output of the receiver for tone 2 (the first tone that is not a null carrier), with the parameter Δ on the horizontal axis. The dashed line is the SNR of the equalizer settings determined in Section V.A in [104], the dotted line is the SNR at initialization (corresponding to no equalizer), and the solid line is the SNR of the CMA settings after convergence. The poor performance for $-40 \leq \Delta \leq 0$ is not of importance because those values of Δ corresponding to picking a symbol synchronization such that each received block is a significant mixture of two trasnmitted blocks, so such values of Δ would not be used in any practical system. Figure 8.11 shows similar plots, but for the CM cost rather than the SNR, and the same comments apply.

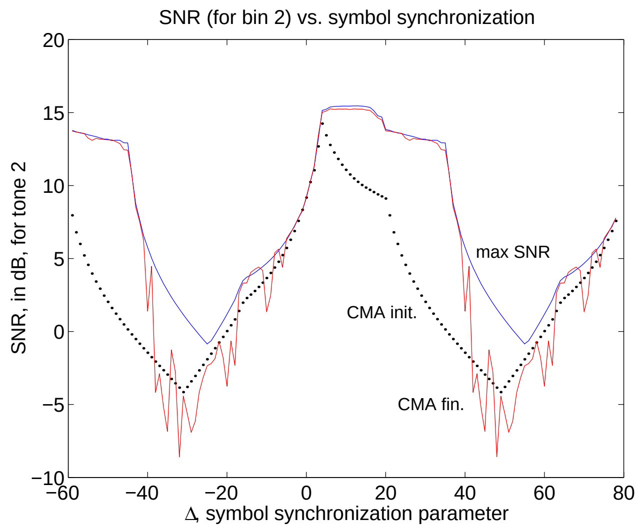

Figure 8.10: Exmaple 4: SNR for PT-CMA and the optimal MMSE solution as a function of delay.

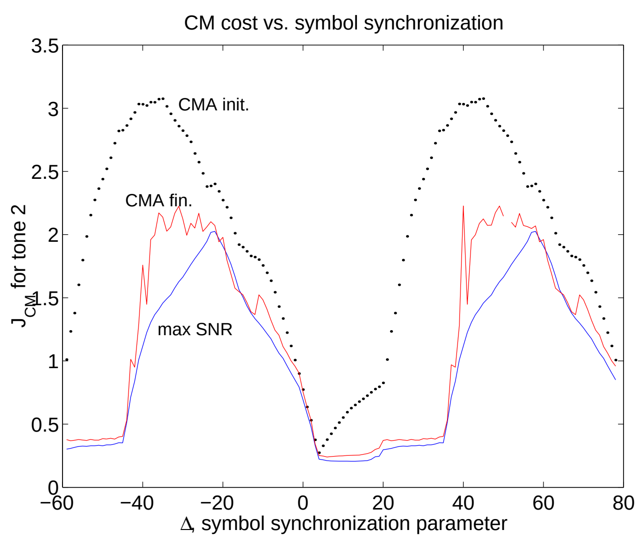

Figure 8.11: Example 4: PT-CM cost for PT-CMA and the optimal MMSE solution as a function of delay.

There are several features to note from these results. First, for a reasonable range of values of Δ around the optimal value, the performance of per tone equalizers is relatively insensitive to the choice of Δ (in our example, this is the region

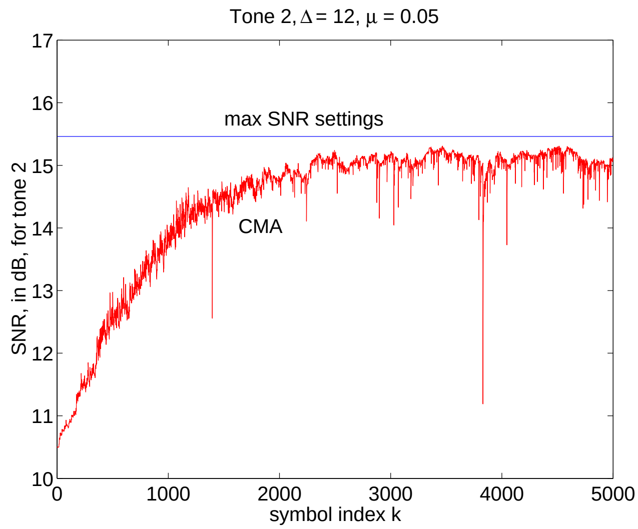

Figure 8.12: Example 4: SNR (for tone 2) over time, using PT-CMA.

$4 \leq \Delta \leq 16$ or $4 \leq \Delta \leq 35$, depending on how much variation in the SNR is considered acceptable). Second, note that the performance of CMA (after convergence) closely matches the performance of the optimal settings [104], especially for the "good" choices of Δ. Also note that the performance is periodic in Δ, with period equal to the symbol size $M = N + \nu = 64 + 16 = 80$.

Figure 8.12 shows the SNR for tone 2 as a function of time. The symbol synchronization parameter Δ was chosen to be within the range discussed above, i.e. $\Delta = 12$. Observe that most of the convergence takes place within 1000 symbols, and optimal performance is achieved asymptotically.

Figure 8.13 and Figure 8.14 show the same simulation as above, but performed using DD-LMS as the adaptive algorithm. A different step size was used for DD-LMS than for CMA under the same conditions. In each case, the stepsize was chosen to be as large as possible without significant asymptotic performance degradation. Observe that DD-LMS exhibits better convergence speed.

One might ask why PT-DDLMS would not always be favored over PT-CMA.

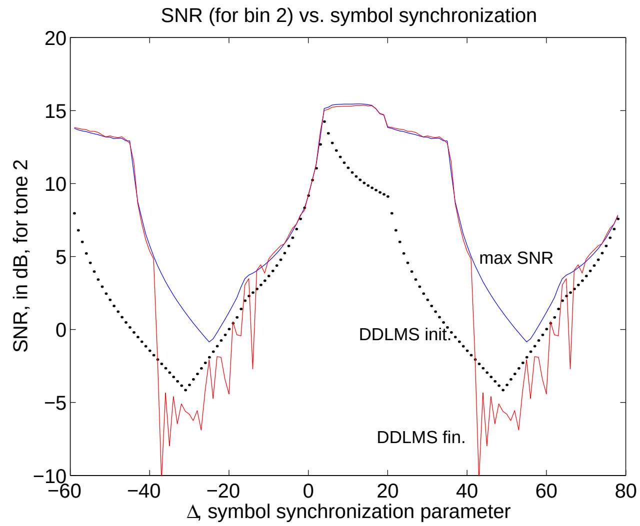

Figure 8.13: Example 4: SNR for PT-DDLMS and the optimal MMSE solution as a function of delay.

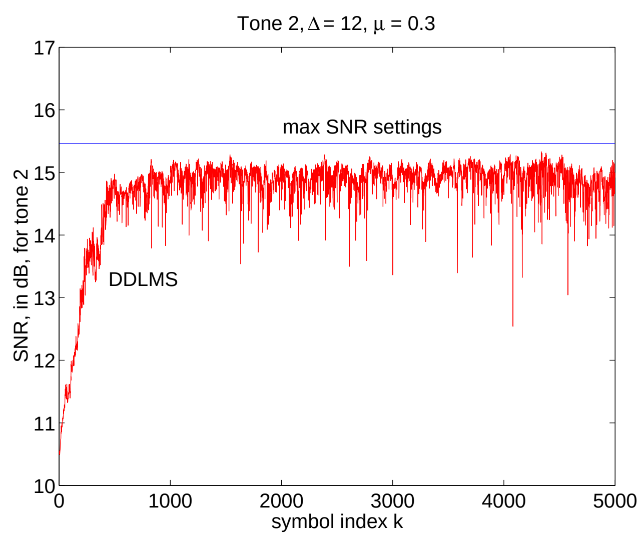

Figure 8.14: Example 4: SNR (for tone 2) over time, using PT-DDLMS.

Conventional wisdom states that DDLMS can only converge to a good answer if the initial decisions are fairly accurate. If the channel is severe enough, this may not be the case. In such a situation, CMA should be used for the first stage of adaptation, then a switch should be made to DDLMS [101].

8.2.5 Example 5: MIMO channel shortening

We now consider the use of multiple transmit and/or receive antennas (or fractional sampling at the receiver). First, we consider a MISO case, which demonstrates MERRY's ability to jointly shorten multiple channels blindly and adaptively using a single filter. There are $L = 2$ transmitters and $P = 1$ receivers. The TEQ has 64 taps, the SNR is 20 dB (AWGN), and the channel is as in (8.2). The two input sequences and the noise sequence were independent. We assume that the transmitted sequences are coarsely synchronized, i.e. that the two cyclic prefixes arrive very roughly at the same time as each other, otherwise no joint channel shortening algorithm will succeed.

The implementation of MERRY in this case is no different than for a single channel. Figure 8.15 shows the two channel impulse responses and the two effective channels shortened by MERRY after convergence. Figure 8.16 shows the joint SSNR [67], [69] versus time. The joint SSNR, as defined in [67], is the total window power in *both* channels divided by the total wall power in both channels. The synchronization was performed blindly as in Section 3.8. The fact that the joint SSNR increased from 6 to 32 is evidence that MERRY can jointly shorten multiple channels, blindly and adaptively.

We now consider bit error rate (BER) as a performance metric. Figs. 8.17 and 8.18 show BER curves for the SISO and SIMO cases, respectively, using the wireless channel model of (8.2), with $L = 1$ and $P = 2$. The BER values were averaged over 200 independent runs for each SNR value. The frequency domain signal was differentially encoded BPSK, so that no FEQ was needed. The blind, non-adaptive MERRY TEQ was compared to the non-adaptive MMSE [4], [2] and MSSNR [67] designs, all using delay optimization. The MIMO MSSNR de-

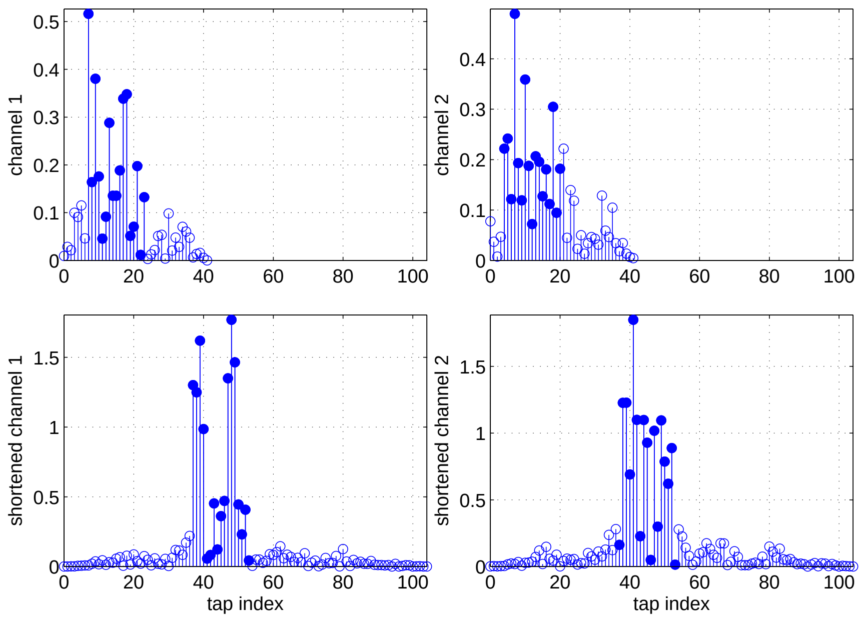

Figure 8.15: Example 5: joint shortening of two Rayleigh fading channels. Top: channel impulse responses magnitudes; bottom: impulse responses magnitudes of the shortened channels. The "filled" stems in the channels indicate the window of $\nu + 1$ taps with largest energy (for the channel) or the window of $\nu + 1$ taps starting with the desired delay (for the shortened channel).

sign is simply the MMSE design with assumptions of white input and no noise. The performance of MERRY with the heuristic delay choice of (3.49) is denoted "MERRY-H." For low SNRs, all TEQs have very little effect on the BER. For larger SNRs, the three delay-optimized methods perform similarly, and MERRY with a heuristic delay performs almost as well. The SISO curves level off for high SNR because the channel cannot be perfectly shortened. The SIMO BER values are much lower because the effective channel can be almost perfectly shortened. Conditions on the feasibility of perfect channel shortening are given in [107].

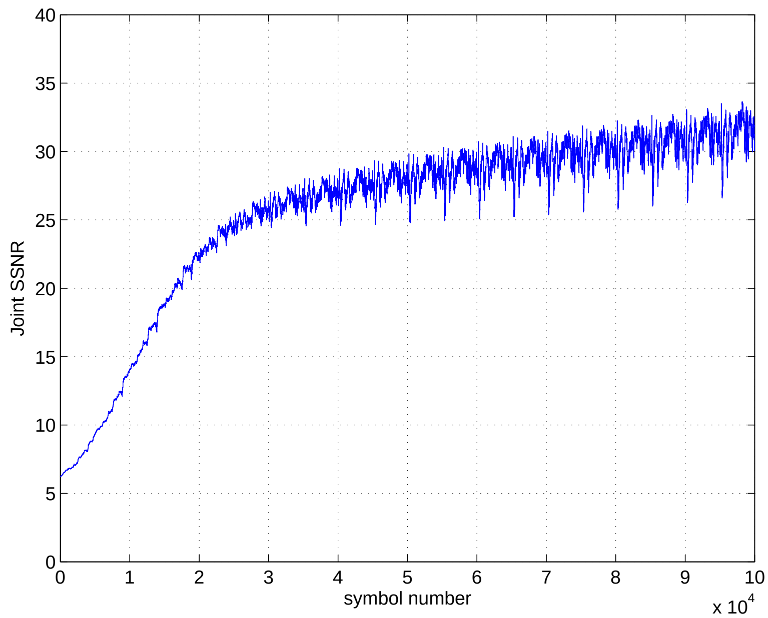

Figure 8.16: Example 5: the joint shortening SNR versus time as FRODO adapts to jointly shorten two Rayleigh fading channels.

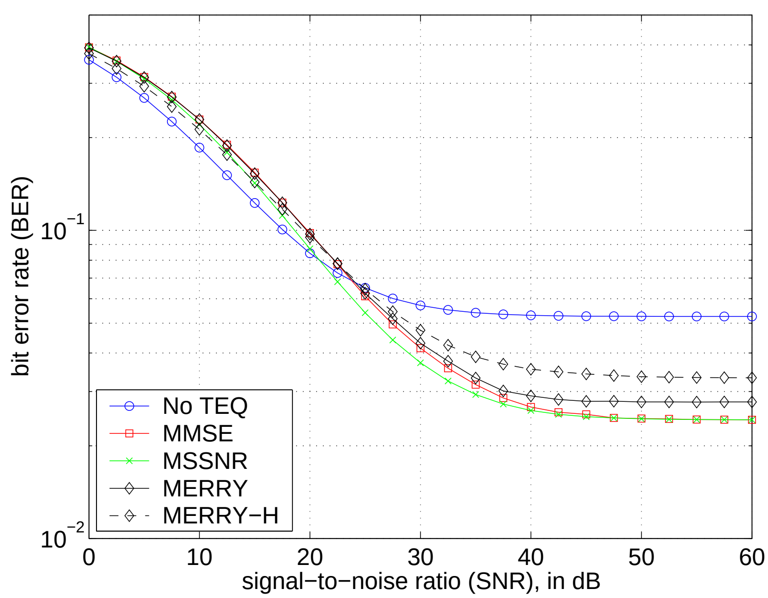

Figure 8.17: Example 5: BER vs. SNR for the SISO case.

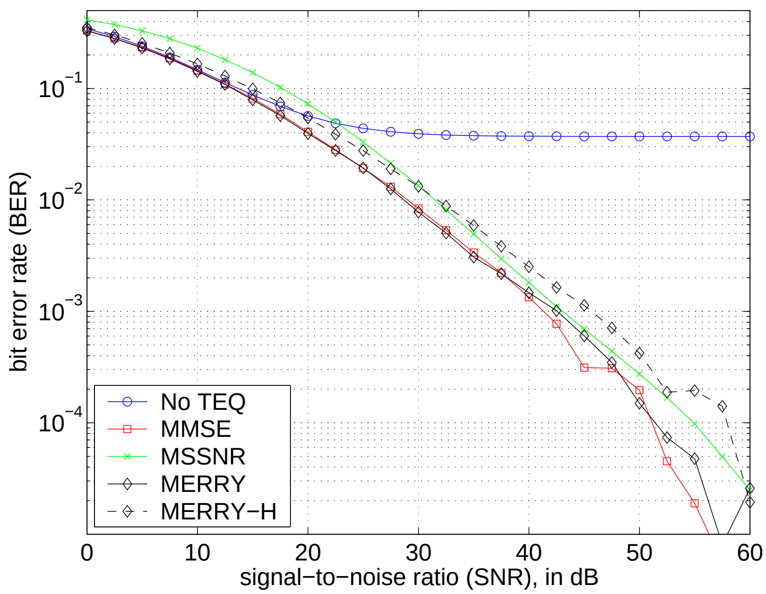

Figure 8.18: Example 5: BER vs. SNR for the SIMO case.

8.2.6 Example 6: symmetric designs

We now simulate the algorithms with symmetric TEQ or TIR impulse responses, as proposed in Chapter 6. The algorithms are the MSSNR design with symmetric TEQ (Sym-MSSNR), the MMSE design with symmetric TIR (Sym-MMSE), and the adaptive MERRY algorithm with a symmetric TEQ (Sym-MERRY).

A comparison of the basic MSSNR TEQ and the Sym-MSSNR TEQ is given in Table 8.4. The TEQ had 32 taps. The channels were the eight standard CSA test loops [93]. The noise was AWGN, with no crosstalk. For a 32-tap TEQ, the performance loss for the symmetric algorithm ranges from 0.1% (loop 3) to 10% (loop 1), with an average loss of 3%. For some channels and TEQ lengths, the symmetric design actually has a higher bit rate. This is because maximizing the SSNR does not necessarily maximize the bit rate. Even though the unconstrained design will have a higher SSNR, it is not guaranteed to have a higher bit rate.

A comparison of the MMSE design and the MMSE design with symmetric TIR

Table 8.4: Achievable bit rate (Mbps) for MSSNR and Sym-MSSNR, using 32-tap TEQs. The last column is the performance of the Sym-MSSNR method relative to the MSSNR method. The channel has AWGN but no crosstalk.

Loop #	MSSNR	Sym-MSSNR	Relative
CSA1	12.187	10.921	89.61%
CSA2	13.016	12.493	95.98%
CSA3	11.543	11.529	99.88%
CSA4	11.696	11.431	97.73%
CSA5	12.120	11.800	97.36%
CSA6	10.995	10.798	98.21%
CSA7	10.978	10.880	99.11%
CSA8	10.294	9.956	96.72%

is given in Figure 8.19 and Table 8.5. The FFT size was $N = 512$, and the CP length was $\nu = 32$. The TEQ had $\tilde{L}_w = 20$ taps in Table 8.5, and $3 \leq \tilde{L}_w \leq 128$ in Figure 8.19. The channels were the eight CSA test loops [93], available at [6].

In Figure 8.19, TEQs were designed for CSA loops 1–8, then the bit rates were averaged. For TEQs with fewer than 20 taps, the bit rate performance of the symmetric MMSE method is not as good as that of the unconstrained MMSE method. However, asymptotically, the results of the two methods agree; and for some parameters, the symmetric method achieves a higher bit rate. Table 8.5 shows the individual bit rates achieved on the 8 channels using 20 tap TEQs, which is roughly the boundary between good and bad performance of the Sym-MMSE design in Figure 8.19. On average, for a 20-tap TEQ, the Sym-MMSE method achieves 89.5% of the bit rate of the MMSE method, with a significantly lower computational cost, but the performance (at this filter length) varies significantly

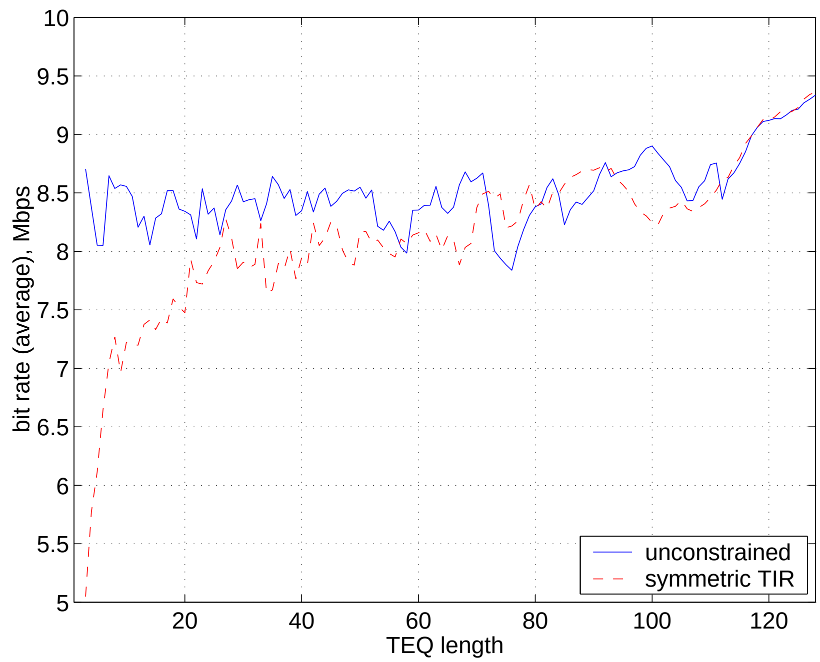

Figure 8.19: Achievable bit rate in Mbps of MMSE (solid) and Sym-MMSE (dashed) designs vs. TEQ length, averaged over eight CSA test loops.

depending on the channel. Thus, it is suggested that the symmetric MMSE design only be used for TEQs with at least 20 taps, and preferably more.

Figure 8.20 shows performance vs. time as the symmetric MERRY TEQ adapts. Again, $N = 512$, and $\nu = 32$. The TEQ had $\tilde{L}_w = 16$ taps (8 taps get updated, then mirrored), and the SNR was 40 dB (AWGN). The channel was CSA loop 4. The dashed line in Figure 8.20 represents the solution obtained by a non-adaptive solution to the MERRY cost (3.4), without imposing symmetry; and the dotted line represents the performance of the MSSNR solution [67]. Sym-MERRY rapidly obtains a near-optimal performance. The jittering around the asymptotic portion of the curve is due to the choice of a large stepsize, as in Figure 8.4.

Table 8.5: Achievable bit rate (Mbps) for MMSE and Sym-MMSE, using 20-tap TEQs and 33-tap TIRs. The last column is the performance of the Sym-MMSE method relative to the MMSE method. The channel has AWGN but no crosstalk.

Loop #	MMSE	Sym-MMSE	Relative
CSA1	8.6323	7.9343	91.91%
CSA2	9.1396	9.1721	100.36%
CSA3	8.5877	8.3360	97.07%
CSA4	8.3157	5.6940	68.47%
CSA5	8.4821	6.3433	74.78%
CSA6	8.8515	9.0016	101.70%
CSA7	7.5244	5.8360	77.56%
CSA8	7.2037	7.4878	103.94%

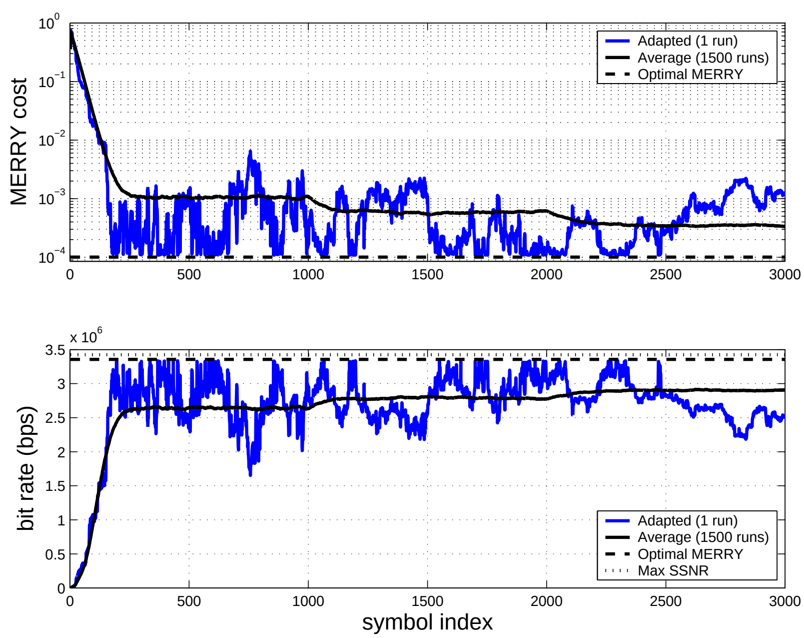

Figure 8.20: Performance of Sym-MERRY vs. time for CSA loop 4. Top: MERRY cost. Bottom: achievable bit rate.

Chapter 9

Conclusions

"I charge thee, waft me safely cross the Channel."

– William Shakespeare, *Henry IV Part 2*, Act IV, Scene i.

This thesis has presented several adaptive channel shortening algorithms based on the "propery restoral" concept. The MERRY and FRODO algorithms blindly and adaptively shorten the channel by attempting to restore the redundancy created by the addition of the cyclic prefix. The SAM algorithm blindly and adaptively shortens the channel by shortening the auto-correlation of the received data, and the TOLKIEN algorithm uses training to perform a similar function by shortening the cross-correlation of the transmitted and received data. The use of frequency domain cost functions was also considered, leading to blind, adaptive per tone equalization algorithms.

In addition, the impulses responses and frequency responses of MMSE and MSSNR channel shortening designs were characterized. It was shown that the impulse responses of MMSE and MSSNR TIRs and TEQs become increasingly symmetric as the TEQ length goes to infinity, and the roots of the TIR approach the unit circle as the TEQ length goes to infinity. The former property leads to reduced-complexity implementations, and the latter property allows for a better understanding of the poor performance of MMSE and MSSNR designs when the TEQ length is unneccessarily long. Since the closed form optimal MERRY channel shortener is related to these non-adaptive designs, it can be similarly understood, and similar complexity reduction techniques were applied to the MERRY update equation.

Table 9.1: Correspondence between thesis chapters and journal paper submissions.

Chapter	**Journal paper submissions**
2	"Multicarrier Unification and Evaluation. Part I: Optimal Designs" [62] "Multicarrier Unification and Evaluation. Part II: Implementation Issues and Performance Comparisons" [63]
3	"A Blind, Adaptive TEQ for Multicarrier Systems" [53], "Low Complexity MIMO Blind, Adaptive Channel Shortening" [64]
4	"Blind, Adaptive Channel Shortening by Sum-squared Auto-correlation Minimization (SAM)" [9]
6, 7	"Infinite Length Results and Design Implications for Time-Domain Equalizers" [56], "Efficient Channel Shortening Equalizer Design" [55]

The material in this thesis has appeared in the journal papers [9], [53], [55], and [56]; the journal paper submissions (currently under review) [62], [63], and [64]; and the conference papers [8], [43], [54], [57], [58], [59], [60], [61], and [65]. The correspondence between the chapters of this thesis and journal paper submissions is given in Table 9.1. Material is © 2004 IEEE and © 2004 Hindawi Publishing Corporation, reprinted with permission. Matlab code to reproduce the figures corresponding to these papers is available at [52].

Appendix A

Proof of Theorem 3.3.1

"Well, sir, for want of other idleness, I'll bide your proof."

– William Shakespeare, *Twelfth Night*, Act I, Scene v.

This appendix[1] proves Theorem 3.3.1, which gives a closed-form representation of the FRODO cost function for a MIMO system, in terms of the channel, the TEQ, and the input and noise statistics.

Proof: Consider the i^{th} term in the cost function (3.7) for CP-OFDM,

$$J_i = \mathrm{E}\left[\left|\sum_{p,l} \mathbf{c}_{p,l}^T \tilde{\mathbf{x}}_{l,i+\Delta}(k) + \sum_{p=1}^{P} \mathbf{w}_p^T \tilde{\mathbf{n}}_{p,i+\Delta}(k)\right|^2\right], \tag{A.1}$$

where

$$\tilde{\mathbf{x}}_{l,i}(k) = \mathbf{x}_l(Mk+i) - \mathbf{x}_l(Mk+i+N), \tag{A.2}$$

$$\mathbf{x}_l(j) = \left[x_l(j), x_l(j-1), \cdots, x_l(j-L_c)\right]^T, \tag{A.3}$$

and similarly for $\tilde{\mathbf{n}}$, but with length $\tilde{L}_w$ rather than $\tilde{L}_c$. Using the definition of $\mathbf{c}_l$ from (3.11), making use of the fact that the noise and data are uncorrelated, and then simplifying,

$$J_i = \mathrm{E}\left[\left|\sum_{l=1}^{L} \mathbf{c}_l^T \tilde{\mathbf{x}}_{l,i+\Delta}(k)\right|^2\right] + \mathrm{E}\left[\left|\sum_{p=1}^{P} \mathbf{w}_p^T \tilde{\mathbf{n}}_{p,i+\Delta}(k)\right|^2\right]. \tag{A.4}$$

Now making use of assumptions A3 and A4,

$$J_i = \sum_{l=1}^{L} \mathbf{c}_l^H \underbrace{\mathrm{E}\left[\tilde{\mathbf{x}}_{l,i+\Delta}^* \tilde{\mathbf{x}}_{l,i+\Delta}^T\right]}_{\mathbf{A}_{x,l}^{i+\Delta}} \mathbf{c}_l + 2\sum_{p=1}^{P} \mathbf{w}_p^H \mathbf{R}_{n,p} \mathbf{w}_p, \tag{A.5}$$

[1]Some material in this appendix has been previously published.

From (1.1), for a CP-OFDM system, $\tilde{\mathbf{x}}$ has ν zeros in the middle, starting at position $i-\nu+\Delta$ (with indexing starting at 0). Using assumptions A1 and A2, this leads to

$$\mathbf{A}_{x,l}^{i+\Delta} = 2\ \sigma_{x,l}^2\ \mathrm{diag}\left[\mathbf{1}_{1\times(i-\nu+\Delta)}, \mathbf{0}_{1\times\nu}, \mathbf{1}_{1\times(L_c+1-i-\Delta)}\right]. \tag{A.6}$$

Thus,

$$J_i = 2\sum_{l=1}^{L} \sigma_{x,l}^2 \|\mathbf{c}_{l,wall}^{i+\Delta}\|^2 + 2\sum_{p=1}^{P} \mathbf{w}_p^H \mathbf{R}_{n,p}\mathbf{w}_p, \tag{A.7}$$

and the result follows by summing (A.7) over $i \in S_f$.

For TZ-OFDM systems, the i^{th} term in the cost function (3.8) simplifies to

$$\begin{aligned} J_{i,TZ}(\Delta) &= 2\ \mathrm{E}\left[\left|\sum_{p,l} \mathbf{c}_{p,l}^T \mathbf{x}_l(Mk+i+\Delta) + \sum_{p=1}^{P} \mathbf{w}_p^T \mathbf{n}_p(Mk+i+\Delta)\right|^2\right] \\ &= 2\ \mathrm{E}\left[\left|\sum_{p,l} \mathbf{c}_{p,l}^T \mathbf{x}_l(Mk+i+\Delta)\right|^2\right] + 2\ \mathrm{E}\left[\left|\sum_{p=1}^{P} \mathbf{w}_p^T \mathbf{n}_p(Mk+i+\Delta)\right|^2\right] \\ &= 2\ \mathrm{E}\left[\left|\sum_{l=1}^{L} \mathbf{c}_p^T \mathbf{x}_l(Mk+i+\Delta)\right|^2\right] + 2\ \mathrm{E}\left[\left|\sum_{p=1}^{P} \mathbf{w}_p^T \mathbf{n}_p(Mk+i+\Delta)\right|^2\right] \end{aligned} \tag{A.8}$$

In going to the second line, we have assumed that the noise and the data are uncorrelated (assumption A4); and in going to the third line, we have made use of the definition of $\mathbf{c}_l$ from (3.11). Now making use of assumption A4,

$$J_{i,TZ} = 2\sum_{l=1}^{L} \mathbf{c}_l^H \underbrace{\mathrm{E}\left[\mathbf{x}_l^*(Mk+i+\Delta)\mathbf{x}_l^T(Mk+i+\Delta)\right]}_{\mathbf{A}_{x,l,TZ}^{i+\Delta}} \mathbf{c}_l + 2\sum_{p=1}^{P} \mathbf{w}_p^H \mathbf{R}_{n,p}\mathbf{w}_p, \tag{A.9}$$

Note that for TZ-OFDM, assumption A3 is not needed.

To simplify $\mathbf{A}_{x,l,TZ}^{i+\Delta}$, observe that

$$\mathbf{x}_l(Mk+i+\Delta) = [x_l(Mk+i+\Delta), x_l(Mk+i+\Delta-1), \cdots, x_l(Mk+i+\Delta-L_c)]^T. \tag{A.10}$$

If assumption A1 holds, then $\mathbf{x}_l(Mk + i + \Delta)$ contains less than one FFT length of data, and $\mathbf{A}_{x,TZ}^{i+\Delta}$ will be diagonal. Moreover, the middle ν elements in (A.10) are all zero, due to the zero-padding of the guard interval which is characteristic of TZ-OFDM, i.e.

$$\mathbf{A}_{x,l,TZ}^{i+\Delta} = \sigma_{x,l}^2 \left[\mathrm{diag}\left(\mathbf{1}_{1\times\Delta},\ \mathbf{0}_{1\times\nu},\ \mathbf{1}_{1\times(L_c+1-\nu-\Delta)}\right)\right]. \tag{A.11}$$

Substituting into (A.9) and summing over i, for TZ-OFDM we also get (3.9). This completes the proof. ■

Appendix B

Proof of Theorem 6.3.1

"And this may help to thicken other proofs that do demonstrate thinly."

– William Shakespeare, *Othello*, Act III, Scene iii.

This appendix[1] proves Theorem 6.3.1, by loosely following the less general proof in [30].

Proof: The MMSE solution requires that $\mathbf{R}_{rr}\mathbf{w} = \mathbf{R}_{rx}\mathbf{b}$ [4]. For an uncorrelated input signal, this simplifies to

$$\mathbf{R}_n\mathbf{w} = \mathbf{H}_{win}^T\mathbf{b} - \mathbf{H}^T\mathbf{H}\mathbf{w}. \tag{B.1}$$

Allowing $-\infty < i < \infty$, the i^{th} component becomes

$$\begin{aligned} \sum_j R_n(i,j)w(j) &= \sum_{j=0}^{\nu} h(\Delta + j - i)b(j) - \sum_j \sum_l h(l-i)h(l-j)w(j) \\ &= \sum_{j=0}^{\nu} h(\Delta + j - i)b(j) - \sum_j \phi(i-j)w(j), \end{aligned} \tag{B.2}$$

where $\phi(m) = \sum_l h(l)h(l+m) = h(m) \star h(-m)$ is the channel covariance function. In convolution notation,

$$w(i) \star R_n(i) = b(i) \star h(\Delta - i) - w(i) \star \phi(i), \tag{B.3}$$

where $R_n(m)$ is the noise autocorrelation function with z-transform $S_n(z)$. Taking z-transforms,

$$W(z)S_n(z) = B(z)z^{-\Delta}H\left(z^{-1}\right) - W(z)\Phi(z). \tag{B.4}$$

[1]Some material in this appendix has been previously published. © 2004 IEEE. Reprinted, with permission, from [60], [61], and [56].

Solving for $W(z)$,

$$\boxed{W(z) = \frac{z^{-\Delta} B(z) H\left(z^{-1}\right)}{S_n(z) + \Phi(z)}.} \tag{B.5}$$

We will make use of (B.5) shortly. The error sequence, measured between the TIR and TEQ outputs, is

$$e(k) = \sum_{l} b(l) x(k - \Delta - l) - \sum_{l} w(l) r(k - l). \tag{B.6}$$

Assuming $x(k)$ is white with unit variance, the error covariance is

$$\begin{aligned} E_m \triangleq \mathrm{E}\left[e(k)e(k+m)\right] = & \sum_{l} b(l) b(l+m) - \sum_{l_1, l_2} w(l_1) b(l_2) h(\Delta - m + l_2 - l_1) \\ & - \sum_{l_1, l_2} w(l_1) b(l_2) h(\Delta + m + l_2 - l_1) \\ & + \sum_{l_1, l_2} w(l_1) w(l_2) \left[\phi(m + l_1 - l_2) + R_n(m + l_1 - l_2)\right]. \end{aligned} \tag{B.7}$$

In convolution notation,

$$\begin{aligned} E_m = & \, b(m) \star b(-m) - b(m - \Delta) \star w(\Delta - m) \star h(\Delta - m) \\ & - b(-m - \Delta) \star w(\Delta + m) \star h(\Delta + m) + w(m) \star w(-m) \star \left[\phi(m) + R_n(m)\right]. \end{aligned} \tag{B.8}$$

Taking z-transforms,

$$\begin{aligned} E(z) = & \, B(z) B(z^{-1}) - z^{-\Delta} B(z) W(z^{-1}) H(z^{-1}) \\ & - z^{\Delta} B(z^{-1}) W(z) H(z) + W(z) W(z^{-1}) \left[\Phi(z) + S_n(z)\right]. \end{aligned} \tag{B.9}$$

Now insert (B.5) into (B.9). Noting that $\Phi(z) = H(z) H(z^{-1})$, and simplifying considerably,

$$E(z) = B(z) B(z^{-1}) \underbrace{\left[\frac{S_n(z) - \Phi(z)}{S_n(z) + \Phi(z)}\right]}_{G(z)}. \tag{B.10}$$

To minimize the MSE, we must minimize E_0. By setting $z = e^{j\omega}$, taking the inverse discrete time Fourier transform, and setting $m = 0$, we find that

$$\begin{aligned} E_0 &= \frac{1}{2\pi}\int_{-\pi}^{\pi} E\left(e^{j\omega}\right)\ d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} B\left(e^{j\omega}\right) B\left(e^{-j\omega}\right) G\left(e^{j\omega}\right)\ d\omega && \text{(B.11)} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \|b\left(e^{j\omega}\right)\|^2 G\left(e^{j\omega}\right)\ d\omega, && \text{(B.12)} \end{aligned}$$

where $b\left(e^{j\omega}\right) = \mathbf{b}^T\left[1, e^{j\omega}, \ldots, e^{j\omega\nu}\right]^T$. This can be rewritten as

$$E_0 = \mathbf{b}^T\mathbf{R}_\Delta\mathbf{b}, \qquad [\mathbf{R}_\Delta]_{m,n} = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\omega(m-n)} G\left(e^{j\omega}\right)\ d\omega = g(m-n). \qquad \text{(B.13)}$$

Since $S_n\left(e^{j\omega}\right)$ and $\Phi\left(e^{j\omega}\right)$ are even functions in ω, $G\left(e^{j\omega}\right)$ is as well. Thus, $[\mathbf{R}_\Delta]_{m,n} = [\mathbf{R}_\Delta]_{n,m}$, so $\mathbf{R}_\Delta$ is a symmetric Toeplitz matrix, and the optimal $\mathbf{b}$ is the eigenvector corresponding to the minimum eigenvalue of $\mathbf{R}_\Delta$. By the results in [12], $\mathbf{b}$ will be symmetric or skew-symmetric. ■

Appendix C

Glossary of Acronyms

"Of this my letters before did satisfy you."

– William Shakespeare, *Antony and Cleopatra*, Act II, Scene ii.

ADSL Asymmetric DSL, in which the downstream (phone company to user) direction carries far more data than the upstream (user to phone company) direction.

AWGN Additive white Gaussian noise.

CFO Carrier Frequency Offset. A mismatch between the actual and the estimated carrier frequency.

CMA The Constant Modulus Algorithm [100], a commonly used blind, adaptive equalization algorithm.

CNA The Carrier Nulling Algorithm [23]. A blind, adaptive equalization algorithm for multicarrier systems.

CP Cylic Prefix. The guard interval between multicarrier symbols, consisting of a copy of the end of the symbol which it precedes.

CSA loops Carrier Serving Area test loops, analytic models of transmission channels in telephone networks [7].

DAB Digital Audio Broadcast. A European multicarrier-based radio broadcast standard [41].

DD Decision-directed. For example, DDLMS is the decision-directed form of LMS in which training is replaced by the quantization of the filter output.

DFT Discrete Fourier Transform.

DMT Discrete Multitone. A form of multicarrier modulation used in wireline systems such as DSL [95].

DS-CDMA Direct Sequence Code Division Multiple Access, a commonly used single-carrier modulation standard for cell phones.

DSL Digital Subscriber Loop. An internet access technique deployed over twisted-pair telephone lines [95].

DVB Digital Video Broadcast. A European multicarrier digital television standard [40].

FEQ Frequency-domain Equalizer. A bank of complex scalars that invert the flat fades on the set of subchannels [95].

FEXT Far-end crosstalk. Interference from one copper wire to the next in a cable bundle in a DSL system [95].

FFT Fast Fourier Transform. An efficient means of computing the DFT.

FIR Finite Impulse Response. A filter or system characterized only by zeros (i.e. no poles), whose response to an impulse is finite in time.

FRODO Forced Redundancy with Optional Data Omission. A blind, adaptive channel shortening algorithm. See Section 3.2.

IBI Inter-block Interference, caused by a channel with a delay spread longer than the cyclic prefix. It is qualitatively similar to ISI. It is defined explicitly in [15].

ICI Inter-carrier Interference, caused by a channel with a delay spread longer than the cyclic prefix.

IDFT Inverse Discrete Fourier Transform.

IFFT Inverse Fast Fourier Transform. An efficient means of computing the IDFT.

IID Independent and Identically Distributed. A set of random variables is i.i.d. if any two subsets of the group have statistically indpendent distributions and any two random variables in the set have the same distribution.

ISI Inter-symbol Interference, caused by a channel with a delay spread longer than the cyclic prefix.

LMS The Least Mean Squares adaptive algorithm [100]. A commonly used trained, adaptive equalization algorithm.

MBR The Maximum Bit Rate channel shortener design [7].

MC-CDMA Multicarrier Code Division Multiple Access, a multiuser form of multicarrier communications in which spreading codes are applied to the input signal before multicarrier modulation.

MERRY Multicarrier Equalization by Restoration of Redundancy. A blind, adaptive channel shortening algorithm. See Section 3.1.

MGSNR The Maximum Geometric Signal-to-Noise Ratio channel shortener design [5].

MIMO A Multiple Input, Multiple Output system.

MISO A Multiple Input, Single Output system.

Min-IBI The Minimum Inter-block Interference channel shortener design [15].

Min-ISI The Minimum Inter-symbol Interference channel shortener design [7].

MMSE Minimum Mean Squared Error channel shortener design [30].

MSSNR Maximum Shortening Signal-to-Noise Ratio channel shortener design [67].

MSSNR-UNT Maximum Shortening Signal-to-Noise Ratio channel shortener design with a unit norm TEQ constraint rather than a unit energy constraint on the window of the effective channel.

MDS Minimum Delay Spread channel shortener design [92]. MDS-L and MDS-Q refer to the use of a linear or quadratic weighting function, respectively. Refer to Section 2.2.

NEXT Near-end crosstalk. Interference from one copper wire to the next in a cable bundle in a DSL system [95].

OFDM Orthogonal Frequency Division Multiplexing. A form of multicarrier modulation used in wireless systems such as wireless local area networks [76].

PLC Power Line Communications. A system employing multicarrier modulation to transmit data over power lines, usually within a building.

PT-xxx Per Tone algorithms or cost functions, e.g. PT-LMS, the per tone LMS algorithm.

PTEQ Per Tone Equalizer. An equalizer structure for multicarrier systems which suppresses the interference separately for each tone [104].

RLS The Recursive Least Squares adaptive algorithm [100].

SAM Sum-squared Auto-correlation Minimization. A blind, adaptive channel shortening algorithm. See Section 4.1.

SISO A Single Input, Single Output system.

SIMO A Single Input, Multiple Output system.

SSNR Shortening Signal-to-Noise Ratio. The energy in a desired window of an impulse response over the energy in the remainder of the impulse response.

TEQ A time-domain equalizer, sometimes called a channel shortening equalizer (CSE).

TIR A short target impulse response, which the convolution of the channel and TEQ is supposed to match. Generally used in the MMSE design [4], [30].

TOLKIEN Trained OFDM L_2-norm Correlation-based Iterative Equalization with Normalization. A trained, adaptive channel shortening algorithm. See Section 4.3.

WSS Wide Sense Stationary. A random process is W.S.S. if its first and second order statistics are time-invariant.

VDSL Very-high-speed DSL, an extension of ADSL allowing for higher data rates over shorter distances.

Alternate Usage

BDOA British Domesticated Ostrich Association. [Credit goes to Jai for this one.]

BSE Bovine Spongiform Encephalopathy, more commonly known as mad cow disease. See also FSE.

CMA Calcium Magnesium Acetate. "A low corrosion, environmental alternative to road salt." (www.cryotech.com/cma.htm) May also refer to the Christian Motorcyclists' Association or the Country Music Association.

DMT Dimethyltryptamine. A molecule that has strong effects on the pineal gland. Reference: "DMT: The Spirit Molecule: A Doctor's Revolutionary Research into the Biology of Near-Death and Mystical Experiences," by Rick Strassman, MD.

DSL "DSL is the conceptual name of an advanced supersonic staging space transportation system, which dates back to the year 1992. Today the name is without any official meaning, but is still in use for historic reasons." (www.kp.dlr.de/DSL/DSL-WWW1.HTML)

FSE Feline Spongiform Encephalopathy. One variety of transmissible spongiform encephalopathies, a category of degenerative brain diseases. See also BSE.

MBR Methyl Bromide. "A toxic pesticide that is injected into soil before planting. Because of its ability to cause poisonings, neurological damage and reproductive harm, EPA classifies methyl bromide as a Toxicity Category I compound, the most deadly category of substances. Methyl bromide is also a powerful ozone depleter. Ozone depletion is linked to raising rates of skin cancer, eye cataracts and damage to key ecosystems." (www.panna.org/resources/documents/mbUseInCA.dv.html)

MMSE Mini Mental State Examination, a widely used test for assessing cognitive mental status, developed by Dr. Marshal F. Folstein of the New England Medical Center. (www.nemc.org/psych/mmse.asp)

RLS Restless Legs Syndrome. Can "cause creepy-crawly sensations in the limbs..." (www.rls.org)

TEQ Toxicity Equivalent. "Contribution of a specified component (or components) to the toxicity of a mixture of related substances. Toxicity equivalent is most commonly used in relation to the reference toxicant 2,3,7,8-tetrachlorodibenzo-p-dioxin, 2,3,7,8-TCDD." (www.iupac.org/reports/1993/6509duffus/t.html)

BIBLIOGRAPHY

[1] A. N. Akansu, P. Duhamel, X. Lin, and M. de Courville. Orthogonal Transmultiplexers in Communication: A Review. *IEEE Transactions on Signal processing*, 46(4):979–995, April 1998.

[2] N. Al-Dhahir. FIR Channel-Shortening Equalizers for MIMO ISI Channels. *IEEE Trans. on Comm.*, 49(2):213–218, February 2001.

[3] N. Al-Dhahir and J. M. Cioffi. Bandwidth Optimization for Combined Equalization and Coding Tranceivers with Emphasis on the MMSE-DFE. *IEEE Military Comm. Conf.*, 2:471–475, October 1993.

[4] N. Al-Dhahir and J. M. Cioffi. Efficiently Computed Reduced-Parameter Input-Aided MMSE Equalizers for ML Detection: A Unified Approach. *IEEE Trans. on Info. Theory*, 42(3):903–915, May 1996.

[5] N. Al-Dhahir and J. M. Cioffi. Optimum Finite-Length Equalization for Multicarrier Transceivers. *IEEE Trans. on Comm.*, 44(1):56–64, January 1996.

[6] G. Arslan, M. Ding, B. Lu, M. Milosevic, Z. Shen, and B. L. Evans. MATLAB DMTTEQ Toolbox 3.1 Beta 3 Release. The University of Texas at Austin, July 27, 2003. [Online.] Available: http://www.ece.utexas.edu/∼bevans/projects/adsl/dmtteq/dmtteq.html.

[7] G. Arslan, B. L. Evans, and S. Kiaei. Equalization for Discrete Multitone Receivers To Maximize Bit Rate. *IEEE Trans. on Signal Processing*, 49(12):3123–3135, December 2001.

[8] J. Balakrishnan, R. K. Martin, and C. R. Johnson, Jr. Blind, Adaptive Channel Shortening by Sum-squared Auto-correlation Minimization (SAM). In *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, November 2002.

[9] J. Balakrishnan, R. K. Martin, and C. R. Johnson, Jr. Blind, Adaptive Channel Shortening by Sum-squared Auto-correlation Minimization (SAM). *IEEE Trans. on Signal Processing*, 51(12):3086–3093, December 2003.

[10] S. Bharitkar and C. Kyriakakis. Eigenfilters for Audio Signal Cancellation. In *Proc. IEEE Int. Sym. Intelligent Signal Proc. and Comm. Sys.*, Honolulu, Hawaii, November 2000.

[11] J. A. C. Bingham. Multicarrier Modulation for Data Transmission: An Idea Whose Time Has Come. *IEEE Communications Magazine*, 28(5):5–14, May 1990.

[12] A. Cantoni and P. Butler. Eigenvalues and Eigenvectors of Symmetric Centrosymmetric Matrices. *Linear Algebra and Its Applications*, 13:275–288, 1976.

[13] A. Cantoni and P. Butler. Properties of the Eigenvectors of Persymmetric Matrices with Applications to Communication Theory. *IEEE Trans. on Comm.*, 24:804–809, August 1976.

[14] S. Celebi. Interblock Interference (IBI) and Time of Reference (TOR) Computation in OFDM Systems. *IEEE Trans. on Comm.*, 49(11):1895–1900, November 2001.

[15] S. Celebi. Interblock Interference (IBI) Minimizing Time-Domain Equalizer (TEQ) for OFDM. *IEEE Signal Processing Letters*, 10(8):232–234, August 2003.

[16] C. Chatterjee, V. P. Roychowdhury, J. Ramos, and M. D. Zoltowski. Self-Organizing Algorithms for Generalized Eigen-Decomposition. *IEEE Trans. on Neural Networks*, 8(6):1518–1530, November 1997.

[17] J. S. Chow, J. M. Cioffi, and J. A. C. Bingham. Equalizer Training Algorithms for Multicarrier Modulation Systems. In *Proc. IEEE Int. Conf. on Comm.*, pages 761–765, Geneva, Switzerland, May 1993.

[18] P. Chow. *Bandwidth Optimized Digital Transmission Techniques for Spectrally Shaped Channels with Impulse Noise.* Ph.D. dissertation, Stanford University, 1993.

[19] W. Chung and C. R. Johnson, Jr. Blind, Adaptive Carrier Frequency Offset Correction for Multicarrier Systems. In *Proc. Conf. on Information Sciences and Systems*, Princeton, NJ, March 2002.

[20] D. Daly, C. Heneghan, and A. D. Fagan. A Minimum Mean-Squared Error Interpretation of Residual ISI Channel Shortening for Discrete Multitone Transceivers. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 4, pages 2065–2068, May 2001.

[21] D. Daly, C. Heneghan, and A. D. Fagan. Minimum Mean-Squared Error Impulse Response Shortening for Discrete Multitone Transcievers. *IEEE Trans. Signal Processing*, 52(1):301–306, 2004.

[22] C. Davis and W. Kahan. The Rotation of Eigenvectors by a Perturbation. *SIAM Journal of Numerical Analysis*, 7:1–46, 1970.

[23] M. de Courville, P. Duhamel, P. Madec, and J. Palicot. Blind equalization of OFDM systems based on the minimization of a quadratic criterion. In *Proc. IEEE Int. Conf. on Comm.*, pages 1318–1321, Dallas, TX, June 1996.

[24] D. M. J. Devasirvatham. A comparison of the time delay spread measurements within two dissimilar office buildings. In *IEEE Int. Conf. on Comm.*, pages 852–856, Toronto, Canada, June 1986.

[25] D. M. J. Devasirvatham. Time Delay Spread and Signal Level Measurements of 850 MHz Radio Waves in Building Environments. *IEEE Trans. on Antennas and Propagation*, 34(11):1300–1305, November 1986.

[26] M. Ding, B. L. Evans, R. K. Martin, and C. R. Johnson, Jr. Minimum Intersymbol Interference Methods for Time Domain Equalizer Design. In *Proc. IEEE Global Comm. Conf.*, San Francisco, CA, December 2003.

[27] A. Duel-Hallen and C. Heegard. Delayed Decision-Feedback Sequence Estimation. *IEEE Trans. on Comm.*, 37:428–436, May 1989.

[28] M. V. Eyuboglu and S. U. Qureshi. Reduced-state Sequence Estimation with Set Partitioning and Decision Feedback. *IEEE Trans. on Comm.*, 36:13–20, January 1988.

[29] M. V. Eyuboglu and S. U. Qureshi. Reduced-state Sequence Estimation for Coded Modulation on Intersymbol Interference Channels. *IEEE Journal on Selected Areas in Comm.*, 7:989–995, August 1989.

[30] D. D. Falconer and F. R. Magee. Adaptive Channel Memory Truncation for Maximum Likelihood Sequence Estimation. *Bell Sys. Tech. Journal*, pages 1541–1562, November 1973.

[31] B. Farhang-Boroujeny and M. Ding. Design Methods for Time-Domain Equalizers in DMT Transceivers. *IEEE Trans. on Comm.*, 49(3):554–562, March 2001.

[32] G. D. Forney. Maximum-likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference. *IEEE Trans. on Info. Theory*, 18:363–378, May 1972.

[33] W. H. Gerstacker and R. Schober. Equalizer Concepts for EDGE. *IEEE Trans. on Wireless Comm.*, 1:190–199, January 2002.

[34] D. N. Godard. Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems. *IEEE Trans. on Comm.*, COM-28:1867–1875, November 1980.

[35] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 1996.

[36] M. H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, Inc., New York, NY, 1996.

[37] R. W. Heath, Jr. and G. B. Giannakis. Exploiting Input Cyclostationarity for Blind Channel Estimation in OFDM Systems. *IEEE Trans. on Signal Processing*, 47(3):848–856, March 1999.

[38] I. Ilani. Time Domain Equalizer for DMT Transceivers – A Geometric Approach. US patent no. 6341298, Jan. 2002.

[39] The European Telecomm. Standards Inst. Broadband Radio Access Networks (BRAN); High Performance Radio Local Area Networks (HIPERLAN) Type 2; System Overview. ETR101 683 114, 1999.

[40] The European Telecomm. Standards Inst. Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television. ETSI EN 300 744 V1.4.1, 2001 Edition.

[41] The European Telecomm. Standards Inst. Radio Broadcasting System, Digital Audio Broadcasting (DAB) to Mobile, Portible, and Fixed Receivers. ETS 300 401, 1995–1997.

[42] A. Johansson. ADSL Lite-The Broadband Enabler for the Mass Market. *Ericsson Review*, no. 4, pp. 154–161, 1998.

[43] C. R. Johnson, Jr., R. K. Martin, J. M. Walsh, A. G. Klein, C. E. Orlicki, and T. Lin. Blind Channel Shorteners. In *Proc. 13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, August 2003.

[44] C. R. Johnson, Jr., P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas. Blind Equalization Using the Constant Modulus Criterion: A Review. *Proceedings of the IEEE*, 86:1927–1950, October 1998.

[45] D. L. Jones. Property-Restoral Algorithms for Blind Equalization of OFDM. In *Proc. IEEE Asilomar Conf. on Signals, Systems and Computers*, pages 619–622, Pacific Grove, CA, November 2003.

[46] D. N. Kalofonos, M. Stojanovic, and J. G. Proakis. Performance of Adaptive MC-CDMA Detectors in Rapidly Fading Rayleigh Channels. *IEEE Trans. on Wireless Comm.*, 2(2):229–239, March 2003.

[47] N. Lashkarian and S. Kiaei. Optimum Equalization of Multicarrier Systems: A Unified Geometric Approach. *IEEE Trans. on Comm.*, 49:1762–1769, October 2001.

[48] W. Lee and F. Hill. A Maximum-Likelihood Sequence Estimator with Decision-Feedback Equalization. *IEEE Trans. on Comm.*, 25(9):971–979, September 1977.

[49] G. Leus and M. Moonen. Per-Tone Equalization for MIMO OFDM Systems. *IEEE Trans. on Signal Processing, Special Issue on Signal Processing for MIMO Wireless Comm. Systems*, 51(11):2965–2975, November 2003.

[50] F. Magee. A Comparison of Compromise Viterbi Algorithm and Standard Equalization Techniques Over Band-limited Channels. *IEEE Trans. on Comm.*, 23:361–367, March 1975.

[51] J. Makhoul. On the Eigenvectors of Symmetric Toeplitz Matrices. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 29:868–872, August 1981.

[52] R. K. Martin. Matlab code for papers by R. K. Martin. [Online]. Available: http://bard.ece.cornell.edu/matlab/martin/index.html.

[53] R. K. Martin, J. Balakrishnan, W. A. Sethares, and C. R. Johnson, Jr. A Blind, Adaptive TEQ for Multicarrier Systems. *IEEE Signal Processing Letters*, 9(11):341–343, November 2002.

[54] R. K. Martin, J. Balakrishnan, W. A. Sethares, and C. R. Johnson, Jr. Blind, Adaptive Channel Shortening for Multicarrier Systems. In *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, November 2002.

[55] R. K. Martin, M. Ding, B. L. Evans, and C. R. Johnson, Jr. Efficient Channel Shortening Equalizer Design. *EURASIP Journal on Applied Signal Processing*, 2003(13):1279–1290, December 2003.

[56] R. K. Martin, M. Ding, B. L. Evans, and C. R. Johnson, Jr. Infinite Length Results and Design Implications for Time-Domain Equalizers. *IEEE Trans. on Signal Processing*, 52(1):297–301, January 2004.

[57] R. K. Martin and C. R. Johnson, Jr. Blind, Adaptive Per Tone Equalization for Multicarrier Receivers. In *Proc. Conf. on Information Sciences and Systems*, Princeton, NJ, March 2002.

[58] R. K. Martin and C. R. Johnson, Jr. Efficient Channel Shortening Equalizer Computation. In *Proc. Conf. on Information Sciences and Systems*, Baltimore, MD, March 2003.

[59] R. K. Martin and C. R. Johnson, Jr. Fast Channel Shortening with Polynomial Weighting Functions. In *Proc. Conf. on Information Sciences and Systems*, Princeton, NJ, March 2004.

[60] R. K. Martin, C. R. Johnson, Jr., M. Ding, and B. L. Evans. Exploiting Symmetry in Channel Shortening Equalizers. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc.*, volume 5, pages V–97–V–100, Hong Kong SAR, China, April 2003.

[61] R. K. Martin, C. R. Johnson, Jr., M. Ding, and B. L. Evans. Infinite Length Results for Channel Shortening Equalizers. In *Proc. IV IEEE Sig. Proc. Workshop on Sig. Proc. Advances in Wireless Comm.*, Rome, Italy, June 2003.

[62] R. K. Martin, K. Vanbleu, M. Ding, G. Ysebaert, M. Milosevic, B. L. Evans, M. Moonen, and C. R. Johnson, Jr. Multicarrier Equalization: Unification and Evaluation. Part I: Optimal Designs. Submitted to *IEEE Transactions on Signal Processing*, Dec. 14, 2003; under revision.

[63] R. K. Martin, K. Vanbleu, M. Ding, G. Ysebaert, M. Milosevic, B. L. Evans, M. Moonen, and C. R. Johnson, Jr. Multicarrier Equalization: Unification and Evaluation. Part II: Complexity Reduction, Adaptation, and Performance Evaluation. Under revision.

[64] R. K. Martin, J. M. Walsh, and C. R. Johnson, Jr. Low Complexity MIMO Blind, Adaptive Channel Shortening. Submitted to *IEEE Trans. on Signal Processing*, Oct. 3, 2003; revised Feb. 6, 2004.

[65] R. K. Martin, J. M. Walsh, and C. R. Johnson, Jr. Low Complexity MIMO Blind Adaptive Channel Shortening. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc.*, Montreal, Quebec, Canada, May 2004.

[66] I. Medvedev and V. Tarokh. A Channel-Shortening Multiuser Detector for DS-CDMA Systems. In *Proceeding of the 53rd Vehicular Technology Conference*, volume 3, pages 1834–1838, Rhodes, Greece, May 2001.

[67] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs. Impulse Response Shortening for Discrete Multitone Transceivers. *IEEE Trans. on Comm.*, 44:1662–1672, December 1996.

[68] D. Messerschmitt. Design of a finite impulse response for the viterbi algorithm and decision-feedback equalizer. In *Proceedings of the Int. Conf. on Comm.*, pages 37D.1–37D.5, Minneapolis, MN, June 1974.

[69] M. Milosevic, L. F. C. Pessoa, and B. L. Evans. Simultaneous Multichannel Time Domain Equalizer Design Based On The Maximum Composite Shortening SNR. In *Proc. IEEE Asilomar Conf.*, volume 2, pages 1895–1899, Pacific Grove, CA, November 2002.

[70] M. Milosevic, L. F. C. Pessoa, B. L. Evans, and R. Baldick. Optimal Time Domain Equalization Design for Maximizing Data Rate of Discrete Multitone Systems. *IEEE Trans. on Signal Processing*, in revision.

[71] M. Milosevic, L. F. C. Pessoa, B. L. Evans, and R. Baldick. DMT bit rate maximization with optimal time domain equalizer filter bank architecture. In *Proc. IEEE Asilomar Conf. on Signals, Systems and Computers*, pages 377–382, Pacific Grove, CA, November 2002.

[72] T. Miyajima and Z. Ding. Second-Order Statistical Approaches to Channel Shortening in Multicarrier Systems. to appear in *IEEE Trans. on Signal Processing*, 2004.

[73] T. Miyajima and Z. Ding. Multicarrier Channel Shortening Based on Second-Order Output Statistics. In *Proc. IEEE Workshop on Signal Proc. Advances in Wireless Comm.*, Rome, Italy, June 2003.

[74] B. Muquet, M. de Courville, and P. Duhamel. Subspace-Based Blind and Semi-Blind Channel Estimation for OFDM Systems. *IEEE Trans. on Signal Processing*, 50:1699–1712, July 2002.

[75] M. Nafie and A. Gatherer. Time-Domain Equalizer Training for ADSL. In *Proc. IEEE Int. Conf. on Comm.*, volume 2, pages 1085–1089, Montreal, Canada, June 1997.

[76] The Inst. of Electrical and Electronics Engineers. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 802.11a. 1999 Edition.

[77] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing, Second Edition.* Prentice Hall, Upper Saddle River, NJ, 1999.

[78] C. E. Orlicki. A Comparison Study of Two Blind, Adaptive Channel Shortening Equalizers in a Multicarrier OFDM System. Masters of Engineering Design Project Report, Cornell University, May 2003.

[79] D. Pal, G. N. Iyengar, and J. M. Cioffi. A New Method of Channel Shortening With Applications to Discrete Multi Tone (DMT) Systems. Part I: Theory. submitted to *IEEE Trans. on Comm.*

[80] D. Pal, G. N. Iyengar, and J. M. Cioffi. A New Method of Channel Shortening With Applications to Discrete Multi Tone (DMT) Systems. In *Proc. IEEE Int. Conf. on Comm.*, volume 2, pages 763–768, June 1998.

[81] A. J. Paulraj and B. C. Ng. Space-time Modems for Wireless Personal Communications. *IEEE Personal Communications*, 5:36–48, February 1998.

[82] S. Qureshi and E. Newhall. An Adaptive Receiver for Data Transmission Over Time-Dispersive Channels. *IEEE Trans. on Info. Theory*, 19(4):448–457, July 1973.

[83] T. S. Rappaport and C. D. McGillem. Characterising the UHF Factory Radio Channel. *Electron. Lett.*, 23(19):1015–1017, 1987.

[84] P. A. Regalia. An Adaptive Unit Norm Filter with Applications to Signal Analysis and Karhunen-Loève Transformations. *IEEE Trans. on Circuits and Systems*, 37(5):646–649, May 1990.

[85] C. Ribeiro, V. Silva, and P. S. R. Diniz. Impulse Response Shortening for xDSL Discrete Multitone Modems with Linear Phase Filters. In *II Conferencia Internacional de Telecomunicaciones y Electrónica (TELEC)*, Cuba, July 2002.

[86] C. Ribeiro, V. Silva, and P. S. R. Diniz. Linear Phase Impulse Response Shortening for xDSL DMT Modems. In *IEEE Int. Telecommunications Symp.*, pages 368–371, Brasil, September 2002.

[87] E. Robinson. *Statistical Communication and Detection.* Griffin, London, 1967.

[88] F. Romano and S. Barbarossa. Non-Data Aided Adaptive Channel Shortening for Efficient Multi-Carrier Systems. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, volume 4, pages 233–236, Hong Kong SAR, China, April 2003.

[89] M. J. Feuerstein S. Y. Seidel, T. S. Rappaport and K. L. Blackard. The impact of surrounding buildings on propagation of wireless inbuilding personal communications system design. In *IEEE Vehicular Technology Conf.*, pages 814–818, Denver, CO, May 1992.

[90] A. Scaglione, G. B. Giannakis, and S. Barbarossa. Redundant Filterbank Precoders and Equalizers, Part I: Unification and Optimal Designs. *IEEE Trans. on Signal Processing*, 47:1988–2006, July 1999.

[91] A. Scaglione, G. B. Giannakis, and S. Barbarossa. Redundant Filterbank Precoders and Equalizers, Part II: Blind Channel Estimation, Synchronization, and Direct Equalization. *IEEE Trans. on Signal Processing*, 47:2007–2022, July 1999.

[92] R. Schur and J. Speidel. An Efficient Equalization Method to Minimize Delay Spread in OFDM/DMT Systems. In *Proc. IEEE Int. Conf. on Comm.*, volume 5, pages 1481–1485, Helsinki, Finland, June 2001.

[93] K. Sistanizadeh. Loss characteristics of the proposed canonical ADSL loops with 100-Ohm termination at 70, 90, and 120 F. ANSI T1E1.4 Committee Contribution, no. 161, Nov. 1991.

[94] A. Stamoulis, S. N. Diggavi, and N. Al-Dhahir. Intercarrier Interference in MIMO OFDM. *IEEE Trans. on Signal Processing*, 50(10):2451–2464, October 2002.

[95] T. Starr, J.M. Cioffi, and P.J. Silvermann. *Understanding Digital Subscriber Line Technology*. Prentice Hall PTR, Upper Saddle River, NJ, 1999.

[96] A. Tkacenko and P. P. Vaidyanathan. Eigenfilter Design of MIMO Equalizers for Channel Shortening. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 3, pages 2361–2364, Orlando, FL, May 2002.

[97] A. Tkacenko and P. P. Vaidyanathan. Noise Optimized Eigenfilter Design of Time-domain Equalizers for DMT Systems. In *Proc. IEEE Int. Conf. on Comm.*, volume 1, New York, NY, Apr.–May 2002.

[98] A. Tkacenko and P. P. Vaidyanathan. A Low-Complexity Eigenfilter Design Method for Channel Shortening Equalizers for DMT Systems. *IEEE Trans. on Comm.*, 51(7), July 2003.

[99] S. Trautmann and N. J. Fliege. Perfect Equalization for DMT Systems Without Guard Interval. *IEEE Journal on Selected Areas in Comm.*, 20(5):987–996, June 2002.

[100] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore. *Theory and Design of Adaptive Filters*. Prentice Hall, New York, 2001.

[101] J. R. Treichler, M. G. Larimore, and J. C. Harp. Practical Blind Demodulators for High-order QAM Signals. *Proceedings of the IEEE*, 86(10):1907–1926, October 1998.

[102] P. P. Vaidyanathan and B. Vrcelj. Fast and Robust Blind-Equalization Based on Cyclic Prefix. In *Proc. IEEE Int. Conf. on Comm.*, volume 1, pages 1–5, New York, NY, Apr.–May 2002.

[103] K. Van Acker, G. Leus, M. Moonen, and T. Pollet. RLS-Based Initialization for Per-Tone Equalizers in DMT Receivers. *IEEE Trans. on Comm.*, 51(6):885–889, June 2003.

[104] K. Van Acker, G. Leus, M. Moonen, O. van de Wiel, and T. Pollet. Per Tone Equalization for DMT-Based Systems. *IEEE Trans. on Comm.*, 49(1):109–119, January 2001.

[105] K. Vanbleu, G. Ysebaert, G. Cuypers, M. Moonen, and K. Van Acker. Bitrate Maximizing Time-Domain Equalizer Design for DMT-based Systems. to appear in *IEEE Trans. on Comm.*, 2004.

[106] K. Vanbleu, G. Ysebaert, G. Cuypers, M. Moonen, and K. Van Acker. Bitrate Maximizing Time-Domain Equalizer Design for DMT-based Systems. In *Proc. IEEE Int. Conf. on Comm.*, pages 2360–2364, May 2003.

[107] J. M. Walsh, R. K. Martin, A. G. Klein, N. S. Xenias, J. A. Pagnotta, and C. R. Johnson, Jr. Necessary and Sufficient Conditions for Perfect Channel Shortening and Implications for Interference Mitigation. In *Proc. V IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, Lisbon, Portugal, July 2004.

[108] X. Wang and K. J. R. Liu. Adaptive Channel Estimation Using Cyclic Prefix in Multicarrier Modulation System. *IEEE Comm. Letters*, 3(10):291–293, October 1999.

[109] D. S. Watkins. *Fundamentals of Matrix Computations.* John Wiley & Sons, 1991.

[110] B. Widrow, J. McCool, M. G. Larimore, and C. R. Johnson, Jr. Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter. *Proceedings of the IEEE*, 64(8):1151–1162, August 1976.

[111] J. Wu, G. Arslan, and B. L. Evans. Efficient Matrix Multiplication Methods to Implement a Near-Optimum Channel Shortening Method for Discrete Multitone Transceivers. In *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, volume 1, pages 152–157, Pacific Grove, CA, November 2000.

[112] X. Yang, T. K. Sarkar, and E. Arvas. A Survey of Conjugate Gradient Algorithms for Solution of Extreme Eigen-Problems of a Matrix. *IEEE Trans. on Accoustics, Speech, and Signal Procesing*, 37(10):1550–1556, October 1989.

[113] C. Yin and G. Yue. Optimal Impulse Response Shortening for Discrete Multitone Transceivers. *Electronics Letters*, 34:35–36, January 1998.

[114] W. Younis and N. Al-Dhahir. Joint Prefiltering and MLSE Equalization of Space-Time Coded Transmissions Over Frequency-Selective Channels. *IEEE Trans. on Vehicular Tech.*, 51:144–154, January 2002.

[115] G. Ysebaert, K. Van Acker, M. Moonen, and B. De Moor. Constraints in Channel Shortening Equalizer Design for DMT-based Systems. *Signal Processing*, 83:641–648, March 2003.

[116] G. Ysebaert, K. Vanbleu, G. Cuypers, and M. Moonen. ADSL Per-tone/Per-group Optimal Equalizer and Windowing Design. In *Proc. Thirty-Sixth Asilomar Conf. on Signals, Systems, and Computers*, November 2002.

[117] G. Ysebaert, K. Vanbleu, G. Cuypers, M. Moonen, and T. Pollet. Combined RLS-LMS Initialization for Per Tone Equalizers in DMT-Receivers. *IEEE Trans. on Signal Processing*, 51(7):1916–1927, July 2003.