NOVEL APPROACHES FOR ROBUST LINEAR AND NONLINEAR TURBO EQUALIZATION

by

Nargiz Kalantarova

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical & Computer Engineering

Koç University

July, 2011

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Nargiz Kalantarova

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assist. Prof. Süleyman Serdar Kozat

Assoc. Prof. Alper T. Erdoğan

Prof. Serpil Sayın

Date:

To my family

ABSTRACT

In this thesis, we consider two types of turbo equalization methods over the frequency selective channels, robust linear turbo equalization methods and nonlinear turbo equalization methods. We provide novel approaches for both of these methods, a minimax and a competitive approaches for robust linear turbo equalization and a nonlinear adaptive turbo equalization method using context trees.

First, robust linear turbo equalization is studied when there are uncertainties in the channel parameters. The turbo equalization framework investigated in this part is consisted of a linear equalizer to combat ISI and a trellis based decoder. Instead of completely tuning the linear equalizer parameters to the available inaccurate channel information, we develop a minimax and a competitive schemes, which incorporate the uncertainty in channel information to equalizer design in order to improve robustness. For both approaches, the problem of obtaining the linear equalizer coefficients is posed as a semi-definite programming (SDP) problem. Approximate implementations of these methods are presented with reduced computational complexity. The performance improvement obtained by the proposed algorithms are demonstrated through simulations under different scenarios.

In the second part, nonlinear turbo equalization is studied when the underlying communication channel is not known at the receiver. In particular, adaptive nonlinear turbo equalization is investigated in order to model the nonlinear dependency of the linear minimum mean square error (MSE) equalizer on the soft information from the decoder. To achieve this, we introduced piecewise linear models based on context trees. The piecewise linear models introduced can adaptively choose both the piecewise regions as well as the linear equalizer coefficients in each region independently, with computational complexity only in the order of a regular adaptive linear equalizer. Through simulations it is demonstrated that this approach is guaranteed to asymptotically achieve the performance of the best piecewise linear equalizer that can choose both its piecewise regions (from a class of doubly exponential number of partitions) as well as its filter parameters based on observing

the whole data in advance. We also quantify the MSE performance of the resulting algorithm and demonstrate the convergence of its MSE to the MSE of the linear minimum MSE estimator as the depth of the context tree and the data length increase.

ÖZETÇE

Bu tez çalışmasında, frekans seçici kanallar için iki tip turbo dengeleme yöntemi üzerinde durulmaktadır: gürbüz doğrusal turbo dengeleme yöntemleri ve doğrusal olmayan turbo dengeleme yöntemleri. Bu yöntemlerin her ikisi için yeni yöntemler, gürbüz lineer turbo dengeleme yöntemleri icin minimax ve kompetitif yaklaşımlar, doğrusal olmayan turbo dengeleme yöntemleri için ise bağlam ağacı bazlı uyarlanır doğrusal olmayan turbo dengeleme yöntemi sunulmuştur.

İlk olarak, gürbüz doğrusal turbo dengeleme kanal parametrelerinde belirsizlikler varken incelenmiştir. Bu bölümde incelenmiş olan turbo dengeleme çerçevesi, kafes tabanlı dekoder ve semboller arası girişim (ISI) ile mücadele etmek için doğrusal dengeleyiciden oluşmuştur. Doğrusal dengeleyici parametrelerini belirsiz kanal bilgilerine ayarlamak yerine, gürbüzlüğü artırmak için dengeleyici probleminin formülasyonuna kanaldaki belirsizliği katarak minimax ve kompetitif yaklaşımları geliştirilmiştir. Her iki yaklaşımda, kanal parametrelerini elde etme problemi SDP (*Semi Definite Programming*) problemine indirgenmiştir. Bu yöntemlerin daha az hesap karmasıklıgı içeren yaklaşık uygulamaları da sunulmuştur. Farklı senaryolar altında yapılan simülasyonlar ile önerilen algoritmalardaki performans artışı gösterilmiştir.

İkinci bölümde ise alıcıda kanal bilgisi yokken doğrusal olmayan turbo dengeleme yöntemleri ele alındı. Özellikle, uyarlanır doğrusal olmayan turbo dengeleme hatanın karesinin ortalamasının minimum değerinin (*Minimum Mean Square Error (MMSE)*) yumuşak bilgiye olan doğrusal olmayan bağımlılığını modellemek amacıyla incelendi. Bunun için, bağlam ağacı bazlı parçalı doğrusal modeller sunulmuştur. Önerilen parcalı doğrusal modeller uyarlamalı olarak hem parcalı b”olgeleri hem de doğrusal dengeleyici katsayılarını bağımsız olarak her bölgede seçebilir ve bu seçimi sadece doğrusal dengeleyici vektörünün uzunluğu karmaşı klığı seviyesinde yapabilir. Simülasyonlar ile önerilen yaklaşımın performansının en iyi parçalı doğrusal dengeleyiciye (iki misli üssel numaralı parçaların içinden) asimptotik olarak yakınsadığı gösterilmiştir. Buna ilaveten önerilen yöntemin derinlik ve data

uzunluğu arttıkça MSE (*Mean Square Error*) değeri hesaplandı ve bu değerin MSE'sinin doğrusal MMSE dengeleyicisinin MSE'sine yakınsadığı gösterildi.

ACKNOWLEDGMENTS

I first wish to express my sincere gratitude to my supervisor, Dr. Süleyman Serdar Kozat. His valuable guidance, experience and encouragement have had a great impact on this thesis. I consider myself to be truly fortunate to have had the opportunity to work with him. I express special thanks to the two readers on my M.S. thesis committee, Dr. Alper Tunga Erdoğan and Dr. Serpil Sayın.

I would like to thank to Koç University and TUBITAK for providing me with financial support which made this study possible.

On a personal note, I would like to thank to the members of the Competitive Signal Processing Laboratory and to my former and current officemates for their friendship, help and support. Finally, I express my gratitude to my parents and my sisters for their lifetime support, encouragement and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In data transmission, the main aim is to achieve reliable communications when sending a data signal from one point to another over channels that exhibit some form of time dispersion. The signal may be observed in the presence of noise and other interfering signals, or it may be distorted due to propagation of the signal from its source to the receiver. Primarily two problems emerge in transmitting a signal from one point to another point, additive noise and inter-symbol interference (ISI). Channel coding and equalization techniques can be used mitigate noise and ISI.

With the introduction of turbo-codes by Berrou et al [1] a considerable interest has been attracted to the field of digital communications. Modern communication systems are typically consisted of a concatenation of several systems, each optimized to perform a single task. In the turbo-processing, an iterative exchange of information is established between two subsystems of the receiver in order to improve the overall system performance. In 1995, the turbo principle was applied to an equalizer by Douillard et al [6]. Turbo equalization has drew lots of interest due to its impressive performance gains for communication equalization, i.e., those that suffer from inter symbol interference (ISI) [30]. Turbo equalization combines equalization with channel decoding in an efficient manner and offers significant gains with respect to the conventional approach where equalization and decoding are realized independently. Turbo equalization enables to jointly and efficiently perform the required equalization and decoding tasks at the receiver, by making use of the turbo principle [1]. This leads to significant performance improvement with regard to conventional disjoint equalization and decoding approaches.

This thesis is based on two papers [11, 12]. In this thesis, two types of robust turbo equalization methods are studied over frequency selective communication channels with inter-symbol interference (ISI), robust linear turbo equalization and nonlinear turbo equalization. Particular attention is devoted to the design of equalization methods when there are inaccuracies in the channel parameters and when the channel is not known at the receiver.

1.1 Linear Turbo Equalization

Turbo equalization takes advantage of the concatenated code structure of the data path that consists of an error-correcting code (ECC) implemented at the transmitter and the convolutional structure of the communication channel perceived as a rate-1 convolutional code [1]. Turbo equalization mimics the classical turbo decoding procedure for the turbo codes, however, one of the intentional ECCs of the classical turbo coding framework is replaced by the unintentional convolutional channel [30]. In the classical turbo equalization framework, MAP or ML based techniques are used for both equalization as well as decoding. However, in certain applications, using MAP or ML based equalizers for large alphabet sizes or long ISI channel filters may be computationally infeasible, since these methods have exponentially increasing complexity with respect to channel length or alphabet size [24]. The MAP based decoders suffer similar computational problems for long convolutional encoding filters.

Linear turbo equalization framework we investigate here is consisted of a linear equalizer to combat ISI and a trellis based decoder. The framework of the linear turbo equalization studied in this thesis, where the MAP equalizer is replaced by a linear equalizer is initially studied in [8], where an LMS adaptive algorithm is used to train the linear equalizer parameters. Different extensions of this idea are further elaborated in [25, 30]. Note that the methods introduced in here can be used in conjunction with such adaptive algorithms since usually the channel or system parameters cannot be learned perfectly by the adaptive algorithms and the uncertainty in learning can be learned perfectly by the adaptive algorithms and the uncertainty in learning can be incorporated in the equalizer design as in here. Along the lines of [25], in [30], authors replaced the MAP equalizer with a linear equalizer or a DFE, where the parameters of these filters are trained using the MMSE criteria. In [30], the parameters of the system are trained assuming perfect knowledge of the channel impulse response. In this thesis, we assume that the underlying communication channel is not known exactly. We propose robust equalization methods in linear turbo equalization framework and compare compare our results with the classical linear MMSE equalizer which is tuned to the inaccurate channel information and demonstrate through simulations that the introduced algorithms provide better BERs in certain scenarios.

1.2 Nonlinear Turbo Equalization

Nonlinear turbo equalization framework studied in this thesis contains a nonlinear equalizer to combat ISI and a trellis based decoder. A significant amount of research has focused on linear MMSE equalization for known or estimated channel information [29, 30]. However, linear MMSE turbo equalizers still have higher computational complexity (and require the channel information or the estimate of the channel) than adaptive linear turbo equalizers that use adaptive learning algorithms such as the recursive least squares (RLS) or least mean squares (LMS), to train their coefficients [24,26,30]. Such computational load becomes especially important in applications with long channel impulse responses [28].

In the context of linear equalization, several different equalization algorithms based on different adaptation methods without channel estimation have been investigated, e.g. [8, 19] due to their relatively good performance with low computational complexity compared to trellis based turbo equalizers. Even though such adaptive linear turbo equalizers may converge to their "‘optimal, i.e., Wiener, solution, they usually deliver inferior performance compared to a linear MMSE turbo equalizer [2], since the Wiener solution is for the stationarized problem, where the time-varying soft information is replaced by its time average [30]. The performance loss of these adaptive algorithms is due to their implicit use of the log likelihood ratio (LLR) information from the decoder as stationary soft decision data [2], whereas a linear MMSE turbo equalizer considers LLR information as nonstationary *a priori* statistics over the transmitted symbols [30]. Hence, the filter coefficients of the linear MMSE turbo equalizer are time varying even for a time invariant channel since the a priori LLRs at the output of the decoder are themselves time varying. However, the filter coefficients of an adaptive linear turbo equalizer, e.g., trained using the LMS update, converge to one time-invariant steady state filter [2]. In order to mimic such nonlinear dependency of the optimal MMSE turbo equalizer to the *a priori* LLRs and to improve the performance of such direct adaptive equalizers, we propose nonlinear adaptive turbo equalizers that can explicitly follow the time variation of the soft decision data.

1.3 Contributions

The contributions of this thesis are as follows:

- In linear turbo equalization framework, robust equalization methods, namely minimax and competitive approaches are introduced.

- A minimax approach is studied where the inaccuracies in channel estimation are incorporated in the problem formulation. Specific to the turbo equalization framework, this setup, unlike [5], needs an adaptive bias term and convolutive structure, where obtaining equalization parameters are formulated as a SDP problem. Approximate implementation of this method is also presented with reduced computational complexity.

- A competitive approach is studied where the cost function is defined with respect to the performance of the best linear equalizer in MSE sense (which is unavailable). As in the minimax case, unlike [14] and [5], this competitive setup has a bias term and a convolutive structure that needs different formulation specific to the turbo equalization framework. Obtaining equalization parameters that optimizes this competitive setup is formulated as a SDP problem. Approximate implementation of this method is also presented with reduced computational complexity.

- A nonlinear adaptive turbo equalization algorithm is introduced using context trees in order to model the nonlinear dependency of the linear MMSE equalizer on the soft information generated from the decoder.

- A proof is given to show that the introduced nonlinear turbo equalization algorithm asymptotically achieves the performance of the best piecewise model defined on this context tree with a computational complexity in the order of an ordinary linear equalizer.

- The convergence of the MSE of the CTW algorithm to the MSE of the linear MMSE estimator is demonstrated as the depth of the context tree and the data length increase.

1.4 Content

In Chapter 2, robust linear turbo equalization methods are studied [12]. In Section 2.1 the basic setup for turbo equalization is described. We illustrate the proposed robust equalization approaches in Section 2.3. First the linear MMSE equalization tuned to the inaccurate channel filter is studied in order to introduce reduced complexity versions of the proposed algorithms. Then, we investigate the minimax approach and the competitive approach, and demonstrate that both problems can be cast as SDP problems. Simulation results to illustrate the performance of the proposed algorithms are presented in Section 2.5. Finally, the conclusions are given in Section 2.6.

In Chapter 3, nonlinear turbo equalization is investigated [11]. In Section 3.1, we introduce the basic system description. The nonlinear equalizers studied are introduced in Section 3.2. In Section 3.2.1, we first introduce a partitioned linear turbo equalization algorithm, where the piecewise regions are fixed. We continue in Section 3.3 with the turbo equalization framework using context trees. We also provide the MSE performance of all the algorithms introduced and compare them to the MSE performance of the linear MMSE equalizer. The numerical examples demonstrating the performance gains and the learning mechanism of the algorithm are introduced in Section 3.4. The conclusions are given in Section 3.5.

Finally in Chapter 4, we summarize this work and give concluding remarks.

1.5 Notations

Throughout this document, bold lowercase letters will denote vectors and bold uppercase letters will denote matrices. All vectors are column vectors and l^2-norm of a vector $\mathbf{v}$ is defined as $\|\mathbf{v}\| = \sqrt{\mathbf{v}^H\mathbf{v}}$, where $(\cdot)^T$, $(\cdot)^H$ and $(\cdot)^+$ represent transpose, conjugate transpose and conjugation, respectively. The time index is shown in the subscripts. The operator $\mathrm{E}[.]$ denotes the expectation operator. For notational simplicity, the expected value of a random variable x is denoted as $\bar{x} = \mathrm{E}[x]$, and the expected value of a random vector $\mathbf{x}$ is $\bar{\mathbf{x}} = \mathrm{E}[\mathbf{x}]$. The matrix $\mathbf{I}$ denotes the identity matrix of appropriate dimensions. The vector (or matrix) $\mathbf{0}$ represents a vector (or matrix) of zeros, where the dimensions are understood from the context. Here, $\mathcal{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2.

The operator "$*$" is the convolution operator. For a square matrix $\mathbf{M}$, $\mathrm{tr}(\mathbf{M})$ is the trace. The sequences are represented using curly brackets, e.g., $\{x_t\}$.

Chapter 2

ROBUST LINEAR TURBO EQUALIZATION UNDER CHANNEL UNCERTAINTIES

In this chapter, we consider robust turbo equalization over communication channels with inter-symbol interference (ISI) in the presence of channel uncertainties [12]. Turbo equalization mimics the classical turbo decoding procedure for the turbo codes, however, one of the intentional ECCs of the classical turbo coding framework is replaced by the "unintentional" convolutional channel [30]. The parameters of this unintentional code are to be estimated by the receiver. These "code parameters" are prone to estimation errors. The inaccuracies in the channel parameters may be either due to imperfect channel estimation caused by the limited training data, high energy noise, or due to the time variations of the channel parameters outside the training period which may be attributed to the time variations in the channel or timing recovery problems. Our goal in this chapter is to present novel turbo equalization approaches to achieve robustness against such potential uncertainties in the estimated channel parameters. In particular, we show that through use of equalization algorithms based on minimax and competitive [5,16] frameworks we can obtain turbo equalization methods whose performance are less sensitive against the channel estimation errors and better in terms of bit-error-rate (BER) over the plug-in methods in certain scenarios.

The methods introduced are set up by using certain MMSE criterion which incorporate the channel inaccuracies in the problem formulation. In the first approach, we apply a minimax framework where the linear equalizer coefficients are selected by minimizing the MSE with respect to the worst possible channel around the inaccurate channel coefficients [13,23]. Then, we extend this framework and define a relative performance measure between the MSE of a linear equalizer and the MMSE of the linear MMSE equalizer calculated with the correct knowledge of the underlying channel [5, 14, 16]. Note that this relative performance measure defines our regret in using a linear equalizer that is not the correct linear MMSE (which is not available). We then seek for a linear equalizer that minimizes this regret with respect to the worst possible channel around the inaccurate channel coefficients. We show that the linear equalizers for both approaches can be found by solving a semidefininite programming (SDP) problem, which can be efficiently solved [3]. For certain applications, as in the linear MMSE equalization setup, applying these approaches at each time instant may require highly computationally complexity [30]. Hence, we also provide approximate implementations of both methods with lower computational complexity.

The robust minimax approach to equalization problems under channel uncertainties is studied in [5, 13, 23]. In [23], the uncertainty in the channel information is represented in spectral domain as bounds on the phase and amplitude function of the unknown channel. However, unlike in here where we explicitly provide the linear equalizer coefficients that are robust in a minimax sense, no expressions for a linear equalizer satisfying the functional forms of the phase and amplitude response are given in [23]. Although in [5, 13] the minimax method is used to incorporate the uncertainty in the channel into the equalization problem, the framework, the application as well as the cost function definitions are different in here. Furthermore, it is critical to note that, unlike in [5], the equalizers used in here have bias terms specific to the turbo equalization framework, hence the minimax problem needs different formulation and optimization. However, we use the similar SDP approach with [5, 14] to solve the convex constraint convex optimization problems.

The competitive approach as an alternative to the minimax framework studied in here has extensive roots in computational learning theory, information theory and signal processing [15–17]. The competitive approach studied in here with a similar cost function is introduced in [5] for linear estimation. However, the competitive framework of [5] concentrates on data estimation where the uncertainty is in the statistics of the desired and noise signals. In here, because of the nature of the communication problem, the uncertainty is in the communication channel; the statistics of the desired signal and the noise are assumed to be known. Hence the competitive algorithms of [5] cannot be applied in here. Furthermore, unlike [5], we specifically work with a constrained linear mapping, i.e., the convolutive channel, which requires different formulations in this study. We emphasize that unlike in [14] or [5], the equalizers used in here have bias terms specific to the turbo equalization framework, hence the competitive approach needs different formulation and optimization.

2.1 System Description of The Turbo Equalization

We first present the general turbo equalization scheme, where equalization and decoding are performed in an iterative manner, by exchanging soft information at all stages of the process. In Fig. 2.1, we provide the basic description of the communication system studied in this chapter. Here, $\{a_t\}$, $t = 1, \ldots, n_a$, $a_t \in \{0, 1\}$, is the transmitted signal. To incorporate redundancy in transmission, the input signal $\{a_t\}$ is encoded by a convolutional code producing $\{b_t\}$, $t = 1, \ldots, n_b$. To further decrease the possible transmission errors, the encoded bits $\{b_t\}$ are interleaved using an S-random interleaver [7] to produce the interleaved and coded bits $\{r_t\}$ where each successive bits are separated at least S bits apart. Finally, the interleaved bits are modulated to produce channel symbols $\{x_t\}$. Throughout the thesis we assume BPSK signaling for notational simplicity when we need to specify a modulation method. Note that for BPSK signaling $x_t = (-1)^{r_t+1}$. However, the formulations for the introduced equalization algorithms are applicable for complex modulated data. The modulated sequence, $\{x_t\}$, is transmitted through a discrete-time finite-length impulse response channel $\{f_t\}$, $t = 0, 1, \ldots, M-1$, represented by $\mathbf{f} \overset{\triangle}{=} [f_{M-1}, \ldots, f_0]^T$. Here, the transmitted signal $\{x_t\}$ is assumed to be uncorrelated owing to the function of the interleaver. The received signal y_t is given by

$$y_t \overset{\triangle}{=} x_t * f_t + n_t = \left(\sum_{k=0}^{M-1} f_k x_{t-k} \right) + n_t,$$

where $\{n_t\}$ is the additive complex white Gaussian noise with zero mean and circular symmetric variance σ_n^2. Note that the underlying channel impulse response vector is not accurately known. An estimate of $\mathbf{f}_t$ is provided as $\hat{\mathbf{f}}_t$ (which can be possibly time varying for certain adaptive methods [10]). The uncertainty in the channel impulse response vector is modeled by $\|\mathbf{f} - \tilde{\mathbf{f}}_t\| \leq \delta$, $\delta \in \mathbb{R}^+$, $\delta < \infty$, where δ or a bound on δ is known. Note that although the results provided here hold for time varying f_t and δ_t, we have dropped the time index from $\mathbf{f}$ and δ for notational simplicity. The received signal $\{y_t\}$ is then processed by a turbo equalization system comprised of an equalizer and a decoder as shown in Fig. 2.1. In this framework, the equalizer and decoder are considered as the inner decoder and outer decoder, respectively, and an iterative decoding scheme is used at the receiver of Fig. 2.1. The equalizer computes the a posteriori information using the received signal,

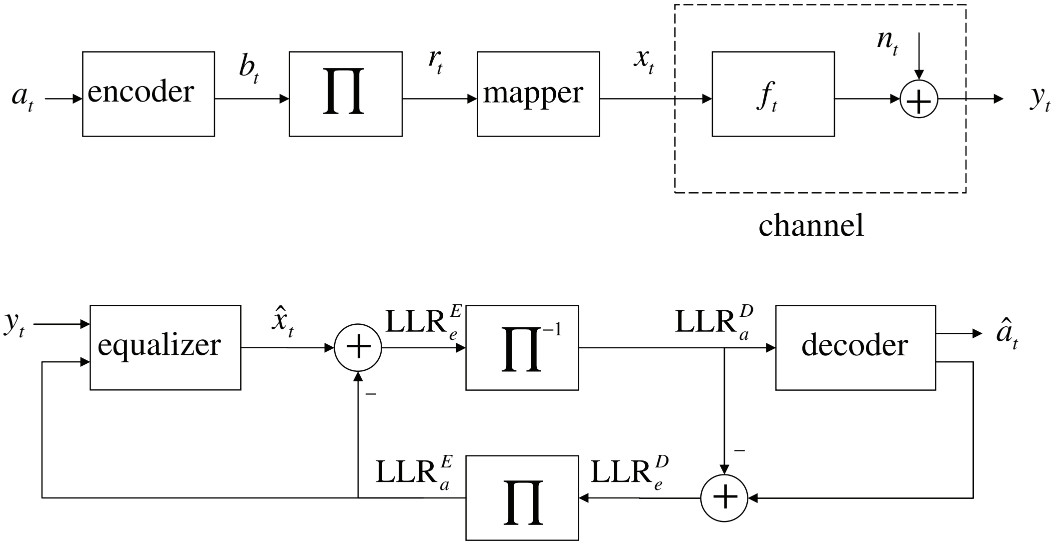


Figure 2.1: A basic turbo equalization framework with the transmitter, the channel and the receiver. The receiver contains both the equalizer and the decoder parts.

transmitted signal estimate, channel convolution matrix (or an estimate of it) and a priori probability of the transmitted signals. After subtracting the a priori information LLR_a^E and LLR_a^E and de-interleaving the extrinsic information LLR_e^E, a soft input soft output (SISO) channel decoder computes the extrinsic information LLR_e^D on coded bits, which are fed back to the linear equalizer as *a priori* information LLR_a^E after interleaving. The *a priori* information from the decoder can be used to compute the mean and variance of the x_t as $\bar{x}_t \overset{\triangle}{=} E[x_t|\{\text{LLR}_{a,t}^E\}]$ and $q_t \overset{\triangle}{=} E[x_t^2|\{\text{LLR}_{a,t}^E\}] - \bar{x}_t^2$, respectively. As an example, for BPSK signaling, the mean and variance are given as $\bar{x}_t = \tanh(\text{LLR}_{a,t}^E/2)$ and $q_t = 1 - |\bar{x}_t|^2$. Hence, the equalizer in turbo equalization system has access to second order statistics of $\{x_t\}$ in addition to $\{y_t\}$. To mitigate ISI we use a linear equalizer. The estimate of the desired data x_t is modeled as follows

$$\hat{x}_t = \mathbf{c}_t^T \mathbf{y}_t + l_t + \bar{x}_t, \tag{2.1}$$

where $\mathbf{c}_t = [c_{t,N_2}, \ldots, c_{t,-N_1}]^T$ is length $N = N_1 + N_2 + 1$ linear equalizer, where N_1 is the length of the anticausal part and N_2 is the length of the causal part and $\mathbf{y}_t \overset{\triangle}{=} [y_{t-N_2}, \ldots, y_{t+N_1}]^T$. It is important to note that in (3.1), the equalizer is "affine", i.e., there is a bias term l_t since the received signal y_t is not zero mean and the mean sequence $\{\bar{y}_t\}$ is not known exactly due to uncertainty in the channel. The received data vector $\mathbf{y}_t$ is given by

$$\mathbf{y}_t = \mathbf{F}\mathbf{x}_t + \mathbf{n}_t,$$

where $\mathbf{x}_t \overset{\triangle}{=} [x_{t-M-N_2+1}, \ldots, x_{t+N_1}]^T$ and $\mathbf{F} \in \mathbb{C}^{N\times(N+M-1)}$

$$\mathbf{F} \overset{\triangle}{=} \begin{bmatrix} f_{M-1} & f_{M-2} & \ldots & f_0 & 0 & \ldots & & 0 \\ 0 & f_{M-1} & f_{M-2} & \ldots & f_0 & 0 & \ldots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ldots & & 0 & f_{M-1} & f_{M-2} & \ldots & f_0 \end{bmatrix}$$

is the convolution matrix constructed by $\mathbf{f} = [f_{M-1}, \ldots, f_0]^T$. The estimate of x_t can be written as

$$\hat{x}_t = \mathbf{c}_t^T\mathbf{F}\mathbf{x}_t + \mathbf{c}_t^T\mathbf{n}_t + l_t + \bar{x}_t \tag{2.2}$$

or

$$\hat{x}_t = \mathbf{f}^T\mathbf{C}_t\mathbf{x}_t + \mathbf{c}_t^T\mathbf{n}_t + l_t + \bar{x}_t,$$

where $\mathbf{C}_t \in \mathbb{C}^{M\times(N+M-1)}$ is the convolution matrix corresponding to $\mathbf{c}_t$ and $\mathbf{c}_t^T\mathbf{F} = \mathbf{f}^T\mathbf{C}_t$.

If the linear MMSE equalizer is used as the linear equalizer, then the corresponding equalizer coefficients can be found as [10]

$$\begin{aligned} \mathbf{c}_t &= [\mathbf{v}^T\mathbf{Q}_t\mathbf{F}^H(\sigma_n^2\mathbf{I} + \mathbf{F}\mathbf{Q}_t\mathbf{F}^H)^{-1}]^T = [\sigma_n^{-2}\mathbf{v}^T(\mathbf{Q}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F})^{-1}\mathbf{F}^H]^T, && (2.3) \\ l_t &= \mathbf{c}_t^T\mathbf{F}\bar{\mathbf{x}}_t, && (2.4) \end{aligned}$$

where $\bar{\mathbf{x}}_t = [\bar{x}_{t-M-N_2+1}, \ldots, \bar{x}_{t+N_1}]^T$, $\mathbf{Q}_t \overset{\triangle}{=} E[(\mathbf{x}_t - \bar{\mathbf{x}}_t)(\mathbf{x}_t - \bar{\mathbf{x}}_t)^H]$ is a diagonal matrix (due to uncorrelatedness assumption on x_t with diagonal entries $\mathbf{Q}_t = \text{diag}\left([q_{t-M-N_2+1}, \ldots, q_t, \ldots, q_{t+N_1}]\right)$ and $\mathbf{v} \in \mathbb{R}^{N+M-1}$ is a vector of all zeros except the $(M + N_2)$th entry is equal to 1. Then, the corresponding linear MMSE is given by

$$\begin{aligned} \min_{\mathbf{c},l} E[\|x_t - \hat{x}_t\|^2] &= \mathbf{v}^T\left[\mathbf{Q}_t - \mathbf{Q}_t\mathbf{F}^H(\sigma_n^2\mathbf{I} + \mathbf{F}\mathbf{Q}_t\mathbf{F}^H)^{-1}\mathbf{F}\mathbf{Q}_t\right]\mathbf{v} \\ &= \mathbf{v}^T(\mathbf{Q}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F})^{-1}\mathbf{v}. \qquad (2.5) \end{aligned}$$

To remove dependency of $\hat{x}_t$ to $\mathrm{LLR}_{a,t}^E$ due to using $\bar{x}_t$ and q_t in (2.3) and (2.2), we set $\mathrm{LLR}_{a,t}^E$ to 0 while computing $\hat{x}_t$, yielding $\bar{x}_t = 0$ and $q_t = 1$ [30]. This changes the covariance matrix to $\mathbf{Q}'_t \triangleq \mathbf{Q} + (1 - q_t)\mathbf{v}\mathbf{v}^T$ and the mean of $\bar{\mathbf{x}}_t$ to $\bar{\mathbf{x}}_t - \bar{x}_t\mathbf{v}$, resulting in (2.2) and (2.3)

$$\mathbf{c}_t = [\sigma_n^{-2}\mathbf{v}^T(\mathbf{Q}'^{-1}_t + \sigma_n^{-2}\mathbf{F}^H\mathbf{F})^{-1}\mathbf{F}^H]^T \text{ and } l_t = \mathbf{c}_t^T\mathbf{F}(\bar{\mathbf{x}}_t - \bar{x}_t\mathbf{v}) \tag{2.6}$$

$$\hat{x}_t = \mathbf{c}_t^T\mathbf{y}_t + l_t. \tag{2.7}$$

Since, the underlying channel vector $\mathbf{f}$ is not accurately known at the receiver, but an estimate $\tilde{\mathbf{f}}_t$, $\|\mathbf{f} - \tilde{\mathbf{f}}_t\| \le \delta$, $\delta \in \mathbb{R}^+$, $\delta < \infty$ is provided, one cannot directly calculate (2.6) and (2.7).

In the next section, we investigate three methods: a method using the inaccurate $\tilde{\mathbf{f}}_t$ to calculate the linear MMSE equalizer in (2.7); a minimax equalizer and a competitive equalizer that incorporate the uncertainty in the problem formulation to mitigate the effect of uncertainty on the equalization performance.

2.2 Linear MMSE Equalization

When the underlying communication channel is not known accurately but an estimated channel vector is provided $\tilde{\mathbf{f}}_t$, a straightforward approach to estimate filter coefficients is to use MMSE estimate corresponding to the estimated channel vector $\tilde{\mathbf{f}}_t$. If one uses a linear MMSE equalizer matched to the estimated channel vector $\tilde{\mathbf{f}}_t$ instead of the channel $\mathbf{f}$ then the equalizer coefficients are obtained as follows

$$\begin{aligned}\tilde{\mathbf{c}}_t &= [\mathbf{v}^T\mathbf{Q}'_t\widetilde{\mathbf{F}}_t^H(\sigma_n^2\mathbf{I} + \widetilde{\mathbf{F}}_t\mathbf{Q}'_t\widetilde{\mathbf{F}}_t^H)^{-1}]^T = [\sigma_n^{-2}\mathbf{v}^T(\mathbf{Q}'^{-1}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\widetilde{\mathbf{F}}_t^H]^T,\\ \tilde{l}_t &= \tilde{\mathbf{c}}_t^T\widetilde{\mathbf{F}}_t\bar{\mathbf{x}}_t,\end{aligned} \tag{2.8}$$

where $\widetilde{\mathbf{F}}_t$ is the convolution matrix of the vector $\tilde{\mathbf{f}}_t$. Calculating $\tilde{\mathbf{c}}_t$ using (2.8) at each time t may be computationally infeasible for certain applications since (2.8) requires $O\left(M^2 + N^2\right)$ operations (as shown in Table 2.1). With the assumption that the channel estimate is time invariant, i.e., $\tilde{\mathbf{f}}_t = \tilde{\mathbf{f}}$, the computational complexity can be reduced by approximating the covariance matrix $\mathbf{Q}'_t$.

As an example, (2.8) can be computed using $\mathbf{Q}'_t = \mathbf{I}$, i.e., if unit variance and zero mean

	number of multiplications	number of additions
MMSE (time variant), $\tilde{\mathbf{c}}_t$	$O(N^2+M^2)$	$O(N^2+M^2)$
MMSE (time invariant), $\tilde{\mathbf{c}}^{\mathrm{APP}}$	$O(N+M)$	$O(N+M)$
Minimax (time variant), $\tilde{\mathbf{c}}_t^{\mathrm{MM}}$	$O(N\,(N+M)^{7/2})$	$O(N\,(N+M)^{7/2})$
Minimax (time invariant), $\tilde{\mathbf{c}}^{\mathrm{MM,APP}}$	$O(N+M)$	$O(N+M)$
Competitive (time variant), $\tilde{\mathbf{c}}_t^{\mathrm{CP}}$	$O(N\,(N+M)^{7/2})$	$O(N\,(N+M)^{7/2})$
Competitive (time invariant), $\tilde{\mathbf{c}}^{\mathrm{CP,APP}}$	$O(N+M)$	$O(N+M)$

Table 2.1: Number of required computations to implement the corresponding algorithms at each time t per received symbol y_t for large packet size. Here, N is the equalizer length, M is the channel length.

is assumed for each x_t, corresponding to a covariance matrix constructed without *a priori* information on x_t [30]. Then, (2.8) can be solved once and the resulting time invariant linear equalizer can be used over the whole block of the received data in (2.3) and (2.4). A better approximation can be achieved by computing (2.8) using $\mathbf{Q}_t = \frac{1}{n_x}(\sum_{i=1}^{n_x} q_i)\mathbf{I}$, i.e., using time averaged variances, yielding $\mathbf{Q}^{'}_{\,t} = \frac{1}{n_x}(\sum_{i=1}^{n_x} q_i)\mathbf{I} - (1-q_t)\mathbf{v}\mathbf{v}^T$, where n_x is the size of the data block. By this approximation (2.8) yields

$$\begin{aligned}\tilde{\mathbf{c}}_t^{\mathrm{APP}} &= [\mathbf{v}^T\widetilde{\mathbf{F}}^H(\beta\widetilde{\mathbf{F}}\widetilde{\mathbf{F}}^H + \sigma_n^2\mathbf{I} + (1-q_t)\widetilde{\mathbf{F}}\mathbf{v}\mathbf{v}^T\widetilde{\mathbf{F}}^H)^{-1}]^T \\ &= \frac{[\mathbf{v}^T\widetilde{\mathbf{F}}^H(\beta\widetilde{\mathbf{F}}\widetilde{\mathbf{F}}^H + \sigma_n^2\mathbf{I})^{-1}]^T}{1+(1-q_t)\mathbf{v}^T\widetilde{\mathbf{F}}^H(\beta\widetilde{\mathbf{F}}\widetilde{\mathbf{F}}^H + \sigma_n^2\mathbf{I})^{-1}\widetilde{\mathbf{F}}\mathbf{v}},\end{aligned} \tag{2.9}$$

where $\beta \overset{\triangle}{=} \frac{1}{n_x}\sum_{i=1}^{n_x} q_i$, (2.9) follows from matrix inversion lemma and $\widetilde{\mathbf{F}}$ is the convolution matrix constructed by the time invariant channel estimate $\tilde{\mathbf{f}}$. Note that to get a time invariant version of (2.9), a time invariant $\mathbf{Q}^{'}_{\,t} = \mathbf{I}$ can be used assuming no *apriori* knowledge on x_t or $\mathbf{Q}^{'}_{\,t} = \beta\mathbf{I}$, i.e., without $(1-q_t)\mathbf{v}\mathbf{v}^T$ term. The linear equalizer with time invariant approximation is given by

$$\begin{aligned}\tilde{\mathbf{c}}^{\mathrm{APP}} &= [\beta\mathbf{v}^T\widetilde{\mathbf{F}}^H(\sigma_n^2\mathbf{I} + \widetilde{\mathbf{F}}\widetilde{\mathbf{F}}^H)^{-1}]^T \\ &= [\sigma_n^{-2}\mathbf{v}^T(\beta\mathbf{I} + \sigma_n^{-2}\widetilde{\mathbf{F}}^H\widetilde{\mathbf{F}})^{-1}\widetilde{\mathbf{F}}^H]^T.\end{aligned} \tag{2.10}$$

The required number of computations at each time t, per received symbol y_t, for $\tilde{\mathbf{c}}_t$ and $\tilde{\mathbf{c}}^{\mathrm{APP}}$ are listed in Table 2.1.

2.3 Linear Equalization with A Minimax Formulation

To improve robustness over (2.8) one can incorporate the uncertainty in the channel estimate. In this minimax framework, the MSE performance is optimized with respect to the worst possible communication channel around the channel estimate $\tilde{\mathbf{f}}_t$ and the equalizer coefficients are obtained by minimizing the worst case MSE, i.e.,

$$\{\tilde{\mathbf{c}}_t^{\mathrm{MM}}, \tilde{l}_t^{\mathrm{MM}}\} = \arg\min_{\mathbf{c},l} \max_{\mathbf{f}=\tilde{\mathbf{f}}_t+\mathbf{df},\|\mathbf{df}\|\leq\delta} \left\{ E[|x_t - \bar{x}_t - \mathbf{c}^T\mathbf{y}_t - l|^2] \right\}. \tag{2.11}$$

Note that

$$\begin{aligned} E[|x_t - \bar{x}_t - \mathbf{c}^T\mathbf{y}_t - l|^2] &= E[|x_t - \bar{x}_t - \mathbf{c}^T\mathbf{F}\mathbf{x}_t - \mathbf{c}^T\mathbf{n}_t - l + \mathbf{c}^T\mathbf{F}\bar{\mathbf{x}}_t - \mathbf{c}^T\mathbf{F}\bar{\mathbf{x}}_t|^2] \qquad (2.12)\\ &= (\mathbf{v} - \mathbf{F}^T\mathbf{c})^H\mathbf{Q}'_t(\mathbf{v} - \mathbf{F}^T\mathbf{c}) + \sigma_n^2\mathbf{c}^H\mathbf{c} + |l + \mathbf{c}^T\mathbf{F}\bar{\mathbf{x}}_t|^2 \\ &= (\mathbf{v} - \mathbf{C}^T\mathbf{f})^H\mathbf{Q}'_t(\mathbf{v} - \mathbf{C}^T\mathbf{f}) + \sigma_n^2\mathbf{c}^H\mathbf{c} + |l + \mathbf{f}^T\mathbf{C}\bar{\mathbf{x}}_t|^2, \end{aligned}$$

where $\mathbf{C}$ is the convolution matrix of $\mathbf{c}$, the second line follows since n_t is i.i.d. and $(\mathbf{x}_t - \bar{\mathbf{x}}_t)$ has zero mean. To find the minimax equalizer coefficients $\{\tilde{\mathbf{c}}_t^{\mathrm{MM}}, \tilde{l}_t^{\mathrm{MM}}\}$ satisfying (2.11),the corresponding problem is formulated in (2.11) as an SDP problem. Note that SDP problems are convex constraint convex optimization problems, where efficient methods exist for their solutions [3]. The following theorem, whose proof is given in Appendix A, provides the corresponding robust linear equalizer while solving the corresponding SDP problem.

Theorem 1: Let $\{x_t\}$, $\{y_t\}$ and $\{n_t\}$ represent the transmitted, received and noise sequences in Fig. 2.1 such that $y_t = f_t * x_t + n_t$, where $\mathbf{f} = [f_{M-1}, \ldots, f_0]^T$ is the unknown, possibly time varying, channel impulse response vector and n_t is zero mean. At each time t, given an estimate $\tilde{\mathbf{f}}_t$ of the underlying communication channel response vector $\mathbf{f}$ satisfying $\mathbf{f} = \tilde{\mathbf{f}}_t + \mathbf{df}$, $\|\mathbf{df}\| \leq \delta$, then the problem

$$\underset{\mathbf{c},l}{\text{minimize}} \underset{\mathbf{f}=\tilde{\mathbf{f}}_t+\mathbf{df},\|\mathbf{df}\|\leq\delta}{\text{maximize}} \left[(\mathbf{v} - \mathbf{C}^T\mathbf{f})^H\mathbf{Q}'_t(\mathbf{v} - \mathbf{C}^T\mathbf{f}) + \sigma_n^2\mathbf{c}^H\mathbf{c} + |l + \mathbf{f}^T\mathbf{C}\bar{\mathbf{x}}_t|^2 \right], \tag{2.13}$$

where $\mathbf{c} = [c_{N_2}, \ldots, c_{-N_1}]^T$ and l are the coefficients of the linear equalizer, $\mathbf{C}$ is the convolution matrix generated from $\mathbf{c}$, $\mathbf{Q}'_t = \mathbf{Q}_t - (1 - q_t)\mathbf{v}\mathbf{v}^T$ and $E[\mathbf{n}_t\mathbf{n}_t^H] = \sigma_n^2\mathbf{I}$ are the covariance matrices of the transmitted and noise sequences, respectively, is equivalent to the SDP problem

$$\underset{\alpha,\mathbf{c},l,\tau}{\text{minimize}}\,\alpha \tag{2.14}$$

such that

$$\begin{bmatrix} \alpha-\tau & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t)^H & (l+\bar{\mathbf{x}}_t^T\mathbf{C}^T\tilde{\mathbf{f}}_t)^H & \mathbf{0} \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t) & \mathbf{0} & \mathbf{Q}'^{-1}_t & \mathbf{0} & -\delta\mathbf{C}^T \\ (l+\bar{\mathbf{x}}_t^T\mathbf{C}^T\tilde{\mathbf{f}}_t) & \mathbf{0} & \mathbf{0} & 1 & \delta\bar{\mathbf{x}}_t^T\mathbf{C}^T \\ \mathbf{0} & \mathbf{0} & -\delta\mathbf{C}^+ & \delta\mathbf{C}^+\bar{\mathbf{x}}_t^+ & \tau\mathbf{I} \end{bmatrix} \geq 0. \tag{2.15}$$

The minimizer $\{\mathbf{c}, l\}$ in (2.14) yields the robust linear equalizer $\{\tilde{\mathbf{c}}_t^{\text{MM}}, \tilde{l}_t^{\text{MM}}\}$ in (2.11).

The proof of the theorem is provided in Appendix A. Note that in Theorem 1, for notational simplicity, the time indices are dropped from $\mathbf{f}$ and δ. The same formulation equally applies to time varying δ. To get the corresponding log-likelihood ratios, $\text{LLR}_{e,t}^E = \ln\frac{p(\hat{x}_t|x_t=+1)}{p(\hat{x}_t|x_t=-1)}$, to fed into the decoder, assume that $p(\hat{x}_t|x_t = x)$ is a Gaussian distribution with $\mathcal{N}\big(E[\hat{x}_t|x_t = x], \text{Cov}(\hat{x}_t, \hat{x}_t|x_t = x)\big)$ [30]. With the formulation $\hat{x}_t = \tilde{\mathbf{c}}_t^{\text{MM},T}\mathbf{y}_t + \tilde{l}_t^{\text{MM}}$, one calculates $E[\hat{x}_t|x_t = x]$ and $\text{Cov}(\hat{x}_t, \hat{x}_t|x_t = x)$ as

$$\begin{aligned} E[\hat{x}_t|x_t = x] &= E[\tilde{\mathbf{c}}_t^{\text{MM},T}\mathbf{y}_t + \tilde{l}_t^{\text{MM}}|x_t = x] \\ &= \tilde{\mathbf{c}}_t^{\text{MM},T}\mathbf{F}(\bar{\mathbf{x}}_t - \bar{x}_t\mathbf{v} + x\mathbf{v}) + \tilde{l}_t^{\text{MM}}, \qquad (2.16) \\ \text{Cov}(\hat{x}_t, \hat{x}_t|x_t = x) &= \tilde{\mathbf{c}}_t^{\text{MM},T}[\sigma_n^2\mathbf{I} + \mathbf{F}(\mathbf{Q}_t - q_t\mathbf{v}\mathbf{v}^T)\mathbf{F}^T]\tilde{\mathbf{c}}_t^{\text{MM}}. \qquad (2.17) \end{aligned}$$

Since $\mathbf{F}$ is unknown, $\widetilde{\mathbf{F}}_t$ in (2.16) and (2.17) are used to calculate $\text{LLR}_{e,t}^E$. As in (2.8) of Section 2.2, the SDP problem in (2.14) should be solved at each time t to compute $\tilde{\mathbf{c}}_t^{\text{MM}}$ since $\mathbf{Q}'_t$ and (possibly) $\tilde{\mathbf{f}}_t$ are time dependent. Although there exists efficient methods to solve the corresponding SDP problem presented in Theorem 1, using these methods for each time instant t may be computationally infeasible in certain applications since these calculations have $O\left(N\,(N+M)^{7/2}\right)$ computational complexity. To reduce computational complexity, assuming a time invariant channel estimate $\tilde{\mathbf{f}}_t = \tilde{\mathbf{f}}$, one can use a time invariant covariance matrix $\mathbf{Q}'_t = \mathbf{I}$ corresponding to a covariance matrix constructed without *a priori* information on x_t or $\mathbf{Q}'_t = \beta\mathbf{I}$ corresponding to time averaged variances. Note that unlike in Section 2.2, one can not directly use ${\mathbf{Q}^{'}}_t = \beta\mathbf{I} - (1-q_t)\mathbf{v}\mathbf{v}^T$, since this formulation is time dependent. Then, the corresponding SDP problem can be solved once to yield a time invariant equalizer, which can be used over the whole block, i.e.,

$$\underset{\alpha,\mathbf{c},l,\tau}{\text{minimize}}\, \alpha \tag{2.18}$$

such that

$$\begin{bmatrix} \alpha-\tau & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}})^H & (l+\bar{\mathbf{x}}_t^T\mathbf{C}^T\tilde{\mathbf{f}})^H & \mathbf{0} \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}) & \mathbf{0} & \beta\mathbf{I} & \mathbf{0} & -\delta\mathbf{C}^T \\ (l+\bar{\mathbf{x}}_t^T\mathbf{C}^T\tilde{\mathbf{f}}) & \mathbf{0} & \mathbf{0} & 1 & \delta\bar{\mathbf{x}}_t^T\mathbf{C}^T \\ \mathbf{0} & \mathbf{0} & -\delta\mathbf{C}^+ & \delta\mathbf{C}^+\bar{\mathbf{x}}_t^+ & \tau\mathbf{I} \end{bmatrix} \geq 0.$$

The required number of computations at each time t, per received symbol y_t, for $\tilde{\mathbf{c}}_t^{\mathrm{MM}}$ and $\tilde{\mathbf{c}}^{\mathrm{MM,APP}}$ are given in Table 2.1.

2.4 Linear Equalization With Competitive Algorithm Formulation

The minimax framework investigated in Section 2.3 to construct robust linear equalizers may result in overly conservative solutions in certain applications, since the linear equalizers are optimized to minimize the MSE corresponding to the worst possible channel. To improve the equalization performance, while trying to preserve robustness, a competitive approach may be used [5,14,16,17]. In this competitive framework, instead of the usual MSE performance, the performance of a linear equalizer is defined with respect to the MMSE linear equalizer tuned to the underlying unknown channel. For any affine equalizer coefficients $\{\mathbf{c}, l\}$, the regret defined as the difference between the MSE using a linear equalizer coefficients $\{\mathbf{c}, l\}$ and the linear MMSE equalizer tuned to $\mathbf{f}$ is

$$\begin{aligned} & E[|x_t - \bar{x}_t - \mathbf{c}^T\mathbf{y}_t - l|^2] - \left(\min_{\mathbf{w},r} E[|x_t - \bar{x}_t - \mathbf{w}^T\mathbf{y}_t - r)|^2]\right) \\ & = \left[(\mathbf{v}-\mathbf{C}^T\mathbf{f})^H{\mathbf{Q}^{'}}_t(\mathbf{v}-\mathbf{C}^T\mathbf{f}) + \sigma_n^2\mathbf{c}^H\mathbf{c} + |l+\mathbf{f}^T\mathbf{C}\bar{\mathbf{x}}_t|^2\right] - \left(\mathbf{v}^T[{\mathbf{Q}^{'}}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F}]^{-1}\mathbf{v}\right), \end{aligned} \tag{2.19}$$

where (2.12) and (2.5) are used in (2.19). However, to make the SDP problem formulation tractable, instead of directly using $(\mathbf{v}^T[{\mathbf{Q}'}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F}]^{-1}\mathbf{v})$ in the regret formulation of (2.19), one can use a first order linear (Taylor) approximation around $\tilde{\mathbf{f}}_t$ [14], given in Appendix A, as

$$\mathbf{v}^T[{\mathbf{Q}'}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F}]^{-1}\mathbf{v} = \eta_t + \mathbf{df}^H{\mathbf{g}_t}^+ + {\mathbf{g}_t}^T\mathbf{df} + O(\|\mathbf{df}\|^2),$$

where $\eta_t \overset{\triangle}{=} \mathbf{v}^T[{\mathbf{Q}'}_t^{-1} + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t]^{-1}\mathbf{v}$ and $\mathbf{g}_t \overset{\triangle}{=} -\widetilde{\mathbf{C}}_t({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\mathbf{v}$ and $\widetilde{\mathbf{C}}_t$ is the convolution matrix constructed using $\tilde{\mathbf{c}}_t$ in (2.8). Using this in (2.19) yields the regret as

$$\left[(\mathbf{v} - \mathbf{C}^T\mathbf{f})^H{\mathbf{Q}'}_t(\mathbf{v} - \mathbf{C}^T\mathbf{f}) + \sigma_n^2\mathbf{c}^H\mathbf{c} + |l + \mathbf{f}^T\mathbf{C}\bar{\mathbf{x}}_t|^2\right] - \left(\eta_t + \mathbf{df}^H{\mathbf{g}_t}^+ + {\mathbf{g}_t}^T\mathbf{df}\right), \quad (2.20)$$

where the $O(\|\mathbf{df}\|^2)$ term is left out. Note that the first order Taylor approximation is introduced in order to make the solution of (2.19) in a minimax setting tractable. Clearly, the effect of this approximation diminishes as $\|\mathbf{df}\|$ gets smaller. For distortions with larger $\|\mathbf{df}\|$, one can use the higher order Taylor approximations instead, however, we have observed through our simulations that the solution using the first order approximation yields satisfactory results even for fairly large $\|\mathbf{df}\|$ (when compared to $\|\mathbf{f}\|$).

To get the competitive linear equalizer, this regret is minimized over all possible communication channels around the channel estimate $\tilde{\mathbf{f}}_t$, i.e.,

$$\{\tilde{\mathbf{c}}_t^{\mathrm{CP}}, \tilde{l}_t^{\mathrm{CP}}\} = \arg\min_{\mathbf{c},l} \max_{\mathbf{f}=\tilde{\mathbf{f}}_t+\mathbf{df}, \|\mathbf{df}\|\leq\delta} \Big[(\mathbf{v} - \mathbf{C}^T\mathbf{f})^H{\mathbf{Q}'}_t(\mathbf{v} - \mathbf{C}^T\mathbf{f}) + \sigma_n^2\mathbf{c}^H\mathbf{c} + |l + \mathbf{f}^T\mathbf{C}\bar{\mathbf{x}}_t|^2$$
$$- (\eta_t + \mathbf{df}^H{\mathbf{g}_t}^+ + {\mathbf{g}_t}^T\mathbf{df})\Big]. \quad (2.21)$$

The problem in (2.21) that will yield the corresponding competitive linear equalizer can be formulated as an SDP problem as follows.

Theorem 2: Let $\{x_t\}$, $\{y_t\}$ and $\{n_t\}$ represent the transmitted, received and noise sequences in Fig. 2.1 such that $y_t = f_t * x_t + n_t$, where $\mathbf{f}$ is the unknown, possibly time varying, channel impulse response vector and n_t is zero mean. At each time t, given an estimate $\tilde{\mathbf{f}}_t$ of the underlying communication channel impulse response vector $\mathbf{f}$ satisfying

$\mathbf{f} = \tilde{\mathbf{f}}_t + \mathbf{df}$, $\|\mathbf{df}\| \le \delta$, then the problem

$$\underset{\mathbf{c},l}{\text{minimize}} \underset{\mathbf{f}=\tilde{\mathbf{f}}_t+\mathbf{df},\|\mathbf{df}\|\le\delta}{\text{maximize}} \Big[(\mathbf{v}-\mathbf{C}^T\mathbf{f})^H\mathbf{Q}^{'}{}_t(\mathbf{v}-\mathbf{C}^T\mathbf{f}) + \sigma_n^2\mathbf{c}^H\mathbf{c} + |l+\mathbf{f}^T\mathbf{C}\bar{\mathbf{x}}_t|^2 \\ -(\eta_t + \mathbf{df}^H\mathbf{g}_t{}^+ + \mathbf{g}_t{}^T\mathbf{df})\Big] \tag{2.22}$$

where $\mathbf{c} = [c_{N_2}, \ldots, c_{-N_1}]^T$ and l are the coefficients of the linear equalizer, $\mathbf{C}$ is the convolution matrix generated from $\mathbf{c}$, $\mathbf{Q}^{'}{}_t = \mathbf{Q}_t - (1-q_t)\mathbf{v}\mathbf{v}^T$ and $E[\mathbf{n}_t\mathbf{n}_t^H] = \sigma_n^2\mathbf{I}$ are the covariance matrices of the transmitted and noise sequences, respectively, $\eta_t = \mathbf{v}^T[\mathbf{Q}^{'}{}_t^{-1} + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t]^{-1}\mathbf{v}$, $\mathbf{g}_t = -\widetilde{\mathbf{C}}_t(\mathbf{Q}^{'}{}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\mathbf{v}$, and > 0, is equivalent to the SDP problem

$$\underset{\alpha,\mathbf{c},l,\tau}{\text{minimize}}\, \alpha \tag{2.23}$$

such that

$$\begin{bmatrix} \alpha+\eta_t-\tau & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t)^H & (l+\bar{\mathbf{x}}_t^T\mathbf{C}^T\tilde{\mathbf{f}}_t)^H & \delta\mathbf{g}_t{}^T \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t) & \mathbf{0} & \mathbf{Q}^{'}{}_t^{-1} & \mathbf{0} & -\delta\mathbf{C}^T \\ (l+\bar{\mathbf{x}}_t^T\mathbf{C}^T\tilde{\mathbf{f}}_t) & \mathbf{0} & \mathbf{0} & 1 & \delta\bar{\mathbf{x}}_t^T\mathbf{C}^T \\ \delta\mathbf{g}_t{}^+ & \mathbf{0} & -\delta\mathbf{C}^+ & \delta\mathbf{C}^+\bar{\mathbf{x}}_t^+ & \tau\mathbf{I} \end{bmatrix} \ge 0. \tag{2.24}$$

The minimizer $\{\mathbf{c}, l\}$ in (2.23) yields the competitive linear equalizer coefficients $\{\tilde{\mathbf{c}}_t^{\text{CP}}, \tilde{l}_t^{\text{CP}}\}$ in (2.21).

The proof of the theorem is provided in Appendix A. Note that in Theorem 2, for notational ease, the time indices from $\mathbf{f}$ and δ are dropped. The same formulation equally applies to time varying δ. Note that to get the corresponding $\text{LLR}_{e,t}^{E}$, one needs to replace $\{\tilde{\mathbf{c}}_t^{\text{MM}}, \tilde{l}_t^{\text{MM}}\}$ with $\{\tilde{\mathbf{c}}_t^{\text{CP}}, \tilde{l}_t^{\text{CP}}\}$ in (2.16) and (2.17).

The proof of the Theorem 2 is provided in Appendix A 5. Instead of solving the SDP problem for all t, one can approximate the time varying correlation matrix $\mathbf{Q}^{'}{}_t$ with a time invariant correlation matrix $\mathbf{Q}^{'}{}_t = \mathbf{I}$ or $\mathbf{Q}^{'}{}_t = \beta\mathbf{I}$ as in Section 2.4. Then, assuming a time invariant channel estimate $\tilde{\mathbf{f}}$, the SDP problem in (2.23) can be solved only once. The linear equalizer $\tilde{\mathbf{c}}^{\text{CP,APP}}$ calculated under this approximation can then be used over the whole block of received data with the corresponding SDP problem formulation

$$\underset{\alpha,\mathbf{c},l,\tau}{\text{minimize}}\,\alpha \tag{2.25}$$

such that

$$\begin{bmatrix} \alpha+\eta-\tau & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}})^H & (l+\bar{\mathbf{x}}_t^T\mathbf{C}^T\tilde{\mathbf{f}})^H & \delta\mathbf{g}^T \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}) & \mathbf{0} & \beta\mathbf{I} & \mathbf{0} & -\delta\mathbf{C}^T \\ (l+\bar{\mathbf{x}}_t^T\mathbf{C}^T\tilde{\mathbf{f}}) & \mathbf{0} & \mathbf{0} & 1 & \delta\bar{\mathbf{x}}_t^T\mathbf{C}^T \\ \delta\mathbf{g}^+ & \mathbf{0} & -\delta\mathbf{C}^+ & \delta\mathbf{C}^+\bar{\mathbf{x}}_t^+ & \tau\mathbf{I} \end{bmatrix} \geq 0.$$

where η and $\mathbf{g}$ are computed using time invariant $\tilde{\mathbf{f}}$ and $\mathbf{Q}^{'}$.

The required number of computations at each time t, per received symbol y_t, for $\tilde{\mathbf{c}}_t^{\mathrm{CP}}$ and $\tilde{\mathbf{c}}^{\mathrm{CP,APP}}$ are given in Table 2.1.

2.5 Simulations

In this section, we demonstrate the performance of the introduced algorithms under different settings. For all examples, bits to be transmitted are encoded using a convolutional encoder with a generator matrix

$$G = \begin{bmatrix} 1 & 0 & D^2; 1 & D & D^2 \end{bmatrix}.$$

An 8-random interleaver is used to shuffle the coded bits such that any consecutive bits will have a minimum distance of 8 bits after interleaving [7]. The coded bits are BPSK modulated. We use linear equalizers introduced in the text and a MAP based algorithm for decoding [24, 30].

For the initial experiments, the modulated bits are transmitted through the ISI channel from [24] (Chapter 10)

$$\mathbf{f} = \begin{bmatrix} 0.227 & 0.46 & 0.688 & 0.46 & 0.227 \end{bmatrix}^T \tag{2.26}$$

with $||\mathbf{f}|| = 1$, $M = 5$ and the noise variance σ_n^2 is determined by

$$\text{SNR} = \frac{E[||x_t||^2]}{N_0} = \frac{1}{2\sigma_n^2}.$$

The channel estimates are constructed using $\tilde{\mathbf{f}} = \mathbf{f} + \mathbf{df}$, where the distortion $\mathbf{df}$ is randomly generated using a zero mean Gaussian distribution. In the first set of experiments, the norm of $\mathbf{df}$ is randomly scaled to give $|\mathbf{df}| \leq 0.3$ for each trial, the length of $\{x_t\}$ is selected as 2048 and SNR $=$ 15dB. For all equalizers $N = 15$, $N_1 = 7$ and $N_2 = 7$. In Fig. 2.2a, we plot the sorted MSEs, i.e., $E[||x_t - \hat{x}_t||^2]$, at the equalizer output for the first iteration of the turbo equalization with respect to 100 randomly selected $\mathbf{df}$'s. Here, we have $\tilde{\mathbf{c}}_t$ from (2.8) labeled "mmse", $\tilde{\mathbf{c}}^{\mathrm{MM}}$ from (2.11) labeled "minimax" and $\tilde{\mathbf{c}}^{\mathrm{CP}}$ from (2.21) labeled "regret". For the same algorithms, we also plot the sorted BERs at the decoder output with respect to randomly selected $\mathbf{df}$'s in Fig. 2.2b. We observe that, as expected, the worst case MSE under channel distortion is minimized for the "minimax" algorithm. The same behavior is observed in BER plot in Fig. 2.2b. However, although the "minimax" algorithm has the best worst case performance, its average performance over randomly selected channel distortions is worse than the "regret" and the "mmse" algorithms: the worst case and the average BERs for the "mmse" algorithm are 0.3542 and 0.0748, respectively; for the "minimax" algorithm are 0.1194 and 0.0847, respectively; for the "regret" algorithm are 0.2798 and 0.0229, respectively. Note that the worst case BER performance of the "regret" approach is worse than the "minimax" method but better than the plug-in MMSE. However, the average BER of the competitive approach is better than the "minimax" method and worse than the "mmse" algorithm. Hence, for these simulations, the competitive approach provides a fair trade-off between the worst case performance and the average case performance. We then plot the corresponding sorted BERs for the second and fourth iterations of turbo equalization. We observe similar results for the second and fourth iterations in Fig. 2.2c and Fig. 2.2d, respectively, such that the robust methods significantly outperform the plug-in "mmse" method for these simulations. We note that the performance improvement due to the robust methods becomes more noticeable as the turbo iteration count increases. We observe that since the "minimax" method is able to minimize the worst case performance, it is able to further minimize the BERs (forcing them to zero) as iteration count increases for all random distortions in these simulations.

In the next set of experiments, we simulate the performance of the introduced algorithms under different SNRs values over the channel in (2.26). However, since the channel estimates usually deteriorate with low SNR [26], we scale the bound for the norm of $\mathbf{df}$ inversely proportional to SNR to give $\|\mathbf{df}\| \leq 0.4$ for SNR $= 0$(dB) and $\|\mathbf{df}\| \leq 0.3$ for SNR $= 6$(dB), i.e., $\|\mathbf{df}\| \leq 0.4 - 0.1$ SNR/6 (based on some empirical values). For these simulations, at each SNR, BERs are averaged over 100 random $\mathbf{df}$ and random $\{x_t\}$ with packet length 2048. In Fig. 2.3a, we present average BERs corresponding to the linear equalizers $\tilde{\mathbf{c}}_t$ from (2.8) labeled "mmse", $\tilde{\mathbf{c}}_t^{\mathrm{MM}}$ from (2.11) labeled "minimax" and $\tilde{\mathbf{c}}_t^{\mathrm{CP}}$ from (2.21) labeled "regret". We present BERs for the first iteration (the straight lines), the second iteration (dashed lines) and the fourth iteration (the dotted lines). We observe that although the robust algorithms are "optimized" with respect to the worst case MSE or to the worst case regret, their average performance is comparable and in certain SNRs much better than the plug-in method. We next repeat the same set of experiments over a different channel from [24]

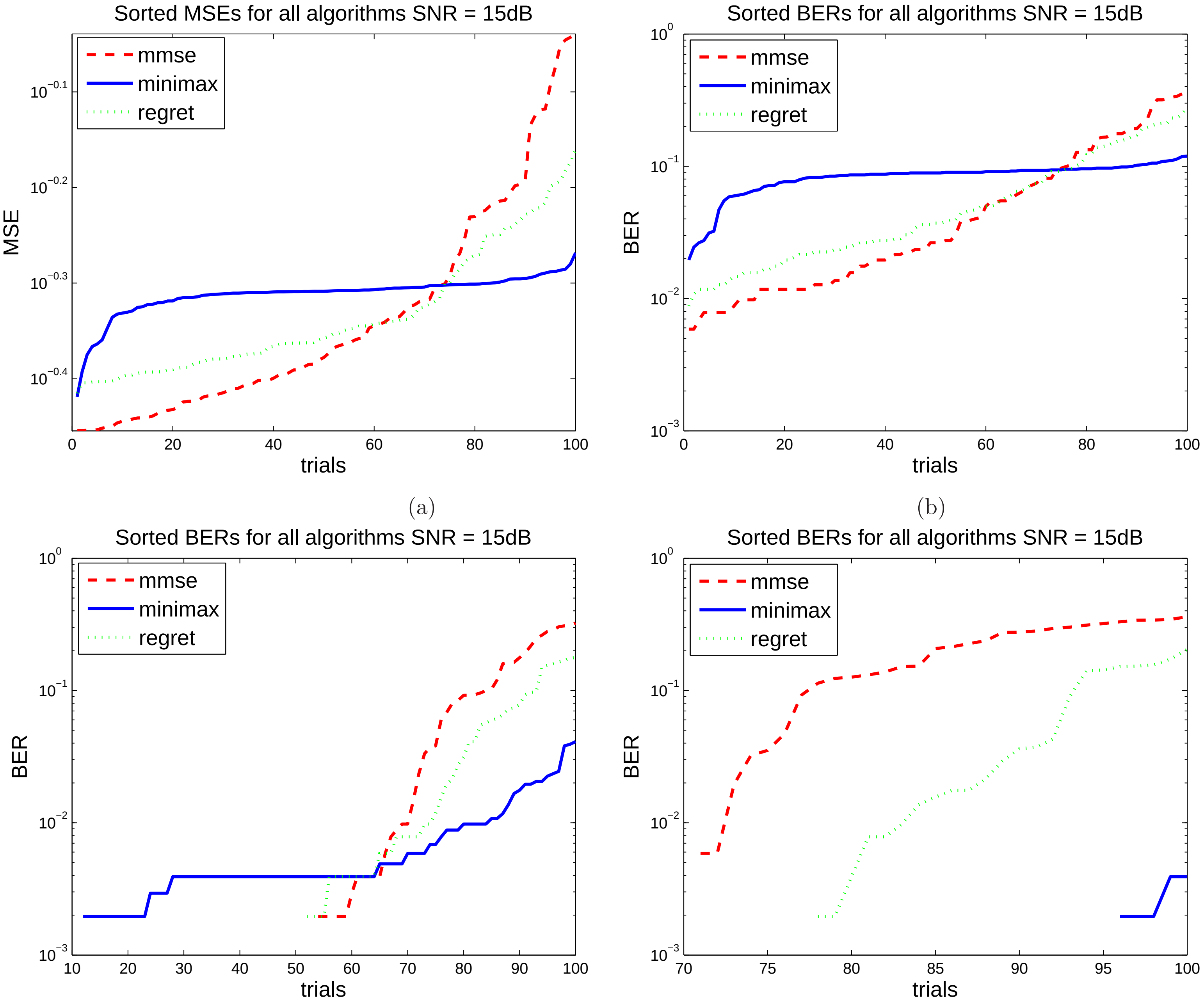
$$\mathbf{f} = \begin{bmatrix} 0.407 & 0.815 & 0.407 \end{bmatrix}^T. \tag{2.27}$$

For these simulations, we run the experiments over 200 randomly selected channel distortions with packet length 4096. For this three tap channel, we choose $N = 7$, $N_1 = 3$ and $N_2 = 3$. The other system parameters are set to the same values as in the first set of experiments. As in the previous example, we scale the norm of randomly generated $\mathbf{df}$ inversely proportional to SNR such that $\|\mathbf{df}\| \leq 0.4$ for SNR $= 0$(dB) and $\|\mathbf{df}\| \leq 0.3$ for SNR $= 6$(dB), i.e., $\|\mathbf{df}\| \leq 0.4 - 0.1$ SNR/6. The BERs with respect to different SNRs are plotted in 2.3b. Note that since this channel introduces less severe ISI than (2.26), the BERs are better than the first channel. We observe similar behavior as in Fig. 2.3a such that the robust methods provide comparable or better BERs with respect to the plug-in method.

We next repeat the previous experiments with the same channels and the same system parameters to test the performance of the approximate implementations. In Fig. 2.4a and 2.4b, we plot the BERs with respect to different SNRs over the channels from (2.26) and from (2.27), respectively, for algorithms with low computational complexity: $\tilde{\mathbf{c}}^{\mathrm{APP}}$ from (2.10) labeled "mmse", $\tilde{\mathbf{c}}^{\mathrm{MM,APP}}$ from (2.18) labeled "minimax" and $\tilde{\mathbf{c}}^{\mathrm{CP,APP}}$ from (2.25) labeled "regret". For all equalizers, we use $\mathbf{Q}'_t = \mathbf{I}$. We present BERs corresponding to the first iteration (the straight lines), the second iteration (dashed lines) and the fourth iteration (the dotted lines). Although, as expected, the performance of the approximate implementations are inferior to exact implementations, we observe that the "regret" and "minimax" algorithms provide similar or better BERs with respect to the time invariant plug-in MMSE equalization algorithm for these simulations.

2.6 Conclusions

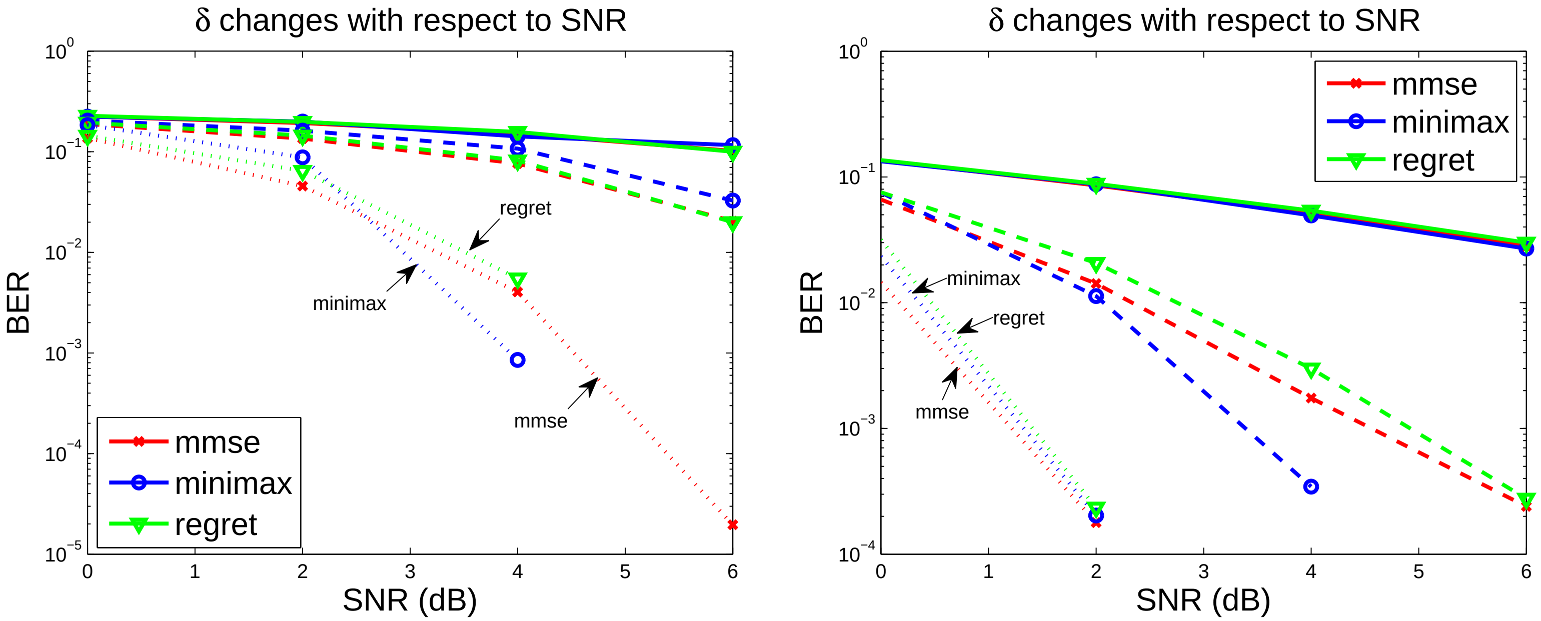
In this chapter, robust linear Turbo equalization problem is investigated when the coefficients of the underlying discrete time communication channel are not accurately known. A minimax approach and a competitive approach are studied in the design of the equalizer where the uncertainty in the channel coefficients are incorporated and a certain MSE optimality criterion is used in the problem formulation. For both approaches, the linear equalizer coefficients are found by solving the corresponding SDP problems. We observed through simulations that the introduced methods improve over the plug-in MMSE estimators for our examples under different distortions and SNRs.



(a) (b)

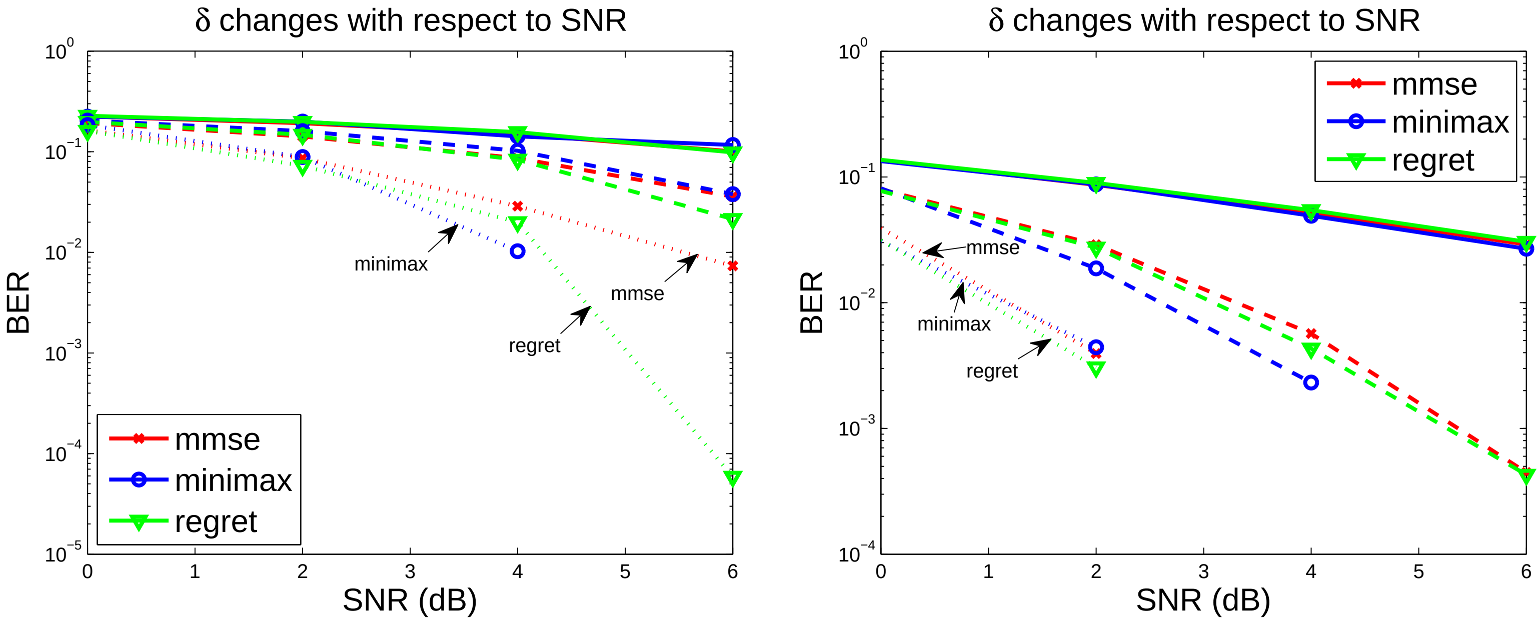
(a) (b)

Figure 2.2: Equalization results for the channel in (2.26) for 100 randomly introduced distortions with $|\mathbf{df}| \leq 0.3$. Here, SNR=15dB. Included algorithms are $\tilde{\mathbf{c}}_t$ from (2.8) labeled "mmse", $\tilde{\mathbf{c}}_t^{\mathrm{MM}}$ from (2.11) labeled "minimax" and $\tilde{\mathbf{c}}_t^{\mathrm{CP}}$ from (2.21) labeled "regret". (a) Sorted MSEs for the 1st iteration. (b) Sorted BERs for the 1st iteration. (c) Sorted BERs for the 2nd iteration. (d) Sorted BERs for the 3rd iteration.



(a) (b)

Figure 2.3: Equalization results and average BERs for BPSK signaling under different SNRs. Included algorithms are $\tilde{\mathbf{c}}_t$ from (2.8) labeled "mmse", $\tilde{\mathbf{c}}_t^{\mathrm{MM}}$ from (2.11) labeled "minimax" and $\tilde{\mathbf{c}}_t^{\mathrm{CP}}$ from (2.21) labeled "regret". Here, the first iteration (the straight lines), the second iteration (dashed lines) and the fourth iteration (the dotted lines). (a) Results for the channel in (2.26) (b) Results for the channel in (2.27).



(a)

(b)

Figure 2.4: Approximate equalization algorithms and average BERs. Included algorithms are $\tilde{\mathbf{c}}^{\mathrm{APP}}$ from (2.8) labeled “mmse”, $\tilde{\mathbf{c}}^{\mathrm{MM,APP}}$ from (2.18) labeled “minimax” and $\tilde{\mathbf{c}}^{\mathrm{CP,APP}}$ from (2.25) labeled “regret”. Here, the first iteration (the straight lines), the second iteration (dashed lines) and the fourth iteration (the dotted lines). (a) Results for the channel in (2.26) (b) Results for the channel in (2.27).

Chapter 3

NONLINEAR TURBO EQUALIZATION

In this chapter, we consider turbo equalization [6] when the underlying communication channel is not known at the receiver. In particular, we investigate adaptive nonlinear turbo equalization in order to model the nonlinear dependency of the optimal linear MMSE equalizer with priors on the soft information from the decoder [11, 30]. To achieve this we introduce piecewise linear models based on context trees [31] that partition the space of variances calculated from the soft information. The nonlinear algorithm we introduce can adaptively choose the piecewise regions as well as the linear equalizer coefficients in each region with computational complexity only in the order of a regular adaptive linear equalizer [24]. We demonstrate that the resulting nonlinear equalizer is guaranteed to asymptotically (and uniformly) achieve the performance of the best piecewise linear equalizer that can choose both its piecewise regions (from a class of doubly exponential number of partitions) as well as the filter parameters in these regions based on the underlying signal. We also quantify the MSE of this equalizer and demonstrate the convergence of its MSE to the MSE of the linear MMSE estimator as the depth of the context tree and data length increase.

We propose a novel adaptive turbo equalization framework where the nonlinear dependence of the linear MMSE turbo equalization on the soft information from the decoder are modeled using piecewise linear models by partitioning the space of variances calculated from the soft information into disjoint regions. By this approach, we view the time-varying nature of the MMSE-optimal filter as a time-invariant, but spatially-varying filter, over the space of a priori LLRs from the decoder. However, instead of directly learning (or approximating) the nonlinear relation through fixed piecewise partitioning of the space of soft information, we use context trees to adaptively learn both the optimal partitioning of the space as well as the filter coefficients for each region. In the context of nonlinear prediction, context trees are used to represent piecewise linear models by partitioning the space of past regressors [16], specifically for labeling the past observations. Note that although we use

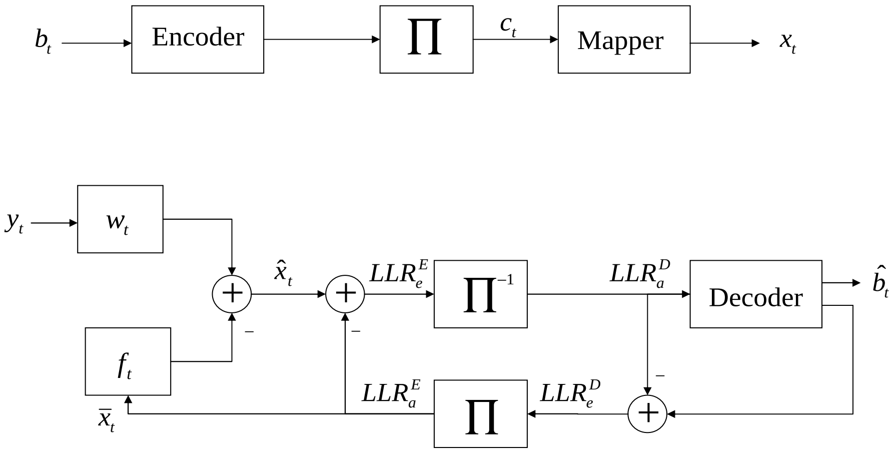


Figure 3.1: The block diagram for a bit interleaved coded modulation transmitter and receiver with a linear turbo equalizer.

the notion of context trees for nonlinear modeling as in [9,17,31], our results and usage of context trees differ from [9,17,31], in a number of important ways. We emphasize that the context trees used here are specifically used to represent the nonlinear dependence of equalizer coefficients on the soft information. In this sense, as an example, the time adaptation is mainly (in addition to learning) due to the time variation of the soft information coming from the decoder. Hence, in here, we explicitly calculate the MSE performance and quantify the difference between the MSE of the context tree algorithm and the MSE of the linear MMSE equalizer, which is the main objective. We also quantify the difference between the MSE performances of the introduced adaptive piecewise modeling and the optimal linear MMSE turbo equalization.

3.1 System Description

The basic communication system studied in this chapter is illustrated in Fig. 3.1. The transmitted information bits $\{b_t\}$ are first encoded using an error correcting code (ECC) and then encoded bits are interleaved producing $\{c_t\}$. The interleaved coded bits $\{c_t\}$ are transmitted after modulation. For notational simplicity we assume BPSK signaling. The symbols $x(t) = (-1)^{c(t)}$ are transmitted through a baseband discrete-time channel with finitle length impulse response $\{h(t)\}$, $t = 0, 1, \ldots, M-1$, represented by $\mathbf{h} \overset{\triangle}{=} [h_{M-1}, \ldots, h_0]^T$. Note that the underlying communication channel $\mathbf{h}$ is unknown. The transmitted signal is assumed as uncorrelated due to interleaver. The received signal $y(t)$ is given in the following formula:

$$y(t) \overset{\triangle}{=} \left(\sum_{k=0}^{M-1} h(k)x(t-k) \right) + n(t),$$

where $\{n(t)\}$ is the additive complex white Gaussian noise with zero mean and circular symmetric variance σ_n^2.

If a linear equalizer is used to mitigate the ISI, then the estimate of the desired data $x(t)$ using the received data $y(t)$ is

$$\hat{x}_t = \mathbf{w}_t^T[\mathbf{y}(t) - \bar{\mathbf{y}}(t)] + \bar{x}(t), \tag{3.1}$$

where $\mathbf{w}(t) = [w(t, N_2), \ldots, w(t, -N_1)]^T$ with length $N = N_1 + N_2 + 1$ linear equalizer, where N_1 is the length of the anticausal part and N_2 is the length of the causal part and the received data vector$\mathbf{y}(t) \overset{\triangle}{=} [y(t-N_2), \ldots, y(t+N_1)]^T$ is given by

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{n}(t),$$

where $\mathbf{x}(t) \overset{\triangle}{=} [x(t-M-N_2+1), \ldots, x(t+N_1)]^T$ and $\mathbf{H} \in \mathbb{C}^{N \times (N+M-1)}$

$$\mathbf{F} \overset{\triangle}{=} \begin{bmatrix} h(M-1) & h(M-2) & \ldots & h(0) & 0 & \ldots & & 0 \\ 0 & h(M-1) & h(M-2) & \ldots & h(0) & 0 & \ldots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ldots & & 0 & h(M-1) & h(M-2) & \ldots & h(0) \end{bmatrix}$$

is the convolution matrix corresponding to $\mathbf{h} = [h(M-1), \ldots, h(0)]^T$. The estimate of $x(t)$ can be written as

$$\hat{x}(t) = \mathbf{w}^T(t)[\mathbf{y}(t) - \mathbf{H}\bar{\mathbf{x}}(t)]\bar{x}(t),$$

given that the mean of the transmitted data is known.

If one uses the linear MMSE equalizer as the linear equalizer, the mean and the variance of $x(t)$ are required to calculate $\mathbf{w}(t)$ and $\hat{x}(t)$. The mean and the variance of the transmitted signal are computed as $\bar{x}(t) = E[x(t)|LLR_a^E(t)]$, and $q(t) = E[x^2(t)|LLR_a^E(t)] - \bar{x}^2(t)$ respectively using the *a priori* information from the decoder. For BPSK modulation $\bar{x}(t) = \tanh(LLR_a^E(t)/2)$ and $q(t) = 1 - |\bar{x}(t)|^2$. Then, the linear MMSE equalizer is given by

$$\mathbf{w}(t) = [\mathbf{v}^H(\sigma_n^2\mathbf{I} + \mathbf{H}_r\mathbf{Q}(t)\mathbf{H}_r^H + \mathbf{v}\mathbf{v}^H)^{-1}]^T, \tag{3.2}$$

where $\mathbf{Q}(t) \in \mathbb{C}^{N+M-2}$ is a diagonal matrix (due to uncorrelatedness assumption on x_t) with diagonal entries $\mathbf{Q}(t) = \text{diag}\left([q(t-M-N_2+1), \ldots, q(t), \ldots, q(t+N_1)]\right)$ and $\mathbf{v} \in \mathbb{C}^N$ is the $(M+N_2)$th column of $\mathbf{H}$, $\mathbf{H}_r$ is the reduced form of $\mathbf{H}$ where $(M+N_2)$th column is removed.

The linear MMSE equalizer is a nonlinear function of the variances of the transmitted signal, i.e.,

$$\mathbf{w}(t) = W(\mathbf{q}) \overset{\triangle}{=} [\mathbf{v}^H(\sigma_n^2\mathbf{I} + \mathbf{H}_r\text{diag}(\mathbf{q})\mathbf{H}_r^H + \mathbf{v}\mathbf{v}^H)^{-1}]^T, \tag{3.3}$$

$W(.) : \mathbb{C}^{N+M-2} \to \mathbb{C}^N$ and time variation in 3.3 is due to the time variation in the vector of variances $\mathbf{q}$, i.e., $\mathbf{w}(t) = W(\mathbf{q}(t))$. Here we assume $\mathbf{h}$ is time invariant.

The linear MMSE equalizer in (3.1) yields

$$\begin{aligned}\hat{x}_t &= \mathbf{w}^T(t)[\mathbf{y}(t) - \mathbf{H}\bar{\mathbf{x}}(t)] \\ &= \mathbf{w}^T(t)\mathbf{y}(t) - \mathbf{f}^T\bar{\mathbf{x}}(t)\end{aligned} \tag{3.4}$$

where $\mathbf{f}(t) \overset{\triangle}{=} \mathbf{H}^T\mathbf{w}(t)$. Equation 3.4 shows that linear MMSE equalizer can be decomposed into a feedforward filter $\mathbf{w}(t)$ on $\mathbf{y}(t)$ and a feedback filter $\mathbf{f}(t)$ on $\mathbf{x}\bar{(}t)$. Both of these filters are nonlinear functions of $\mathbf{q}$, i.e., $\mathbf{w} = W(\mathbf{q})$ and $\mathbf{f} = F(\mathbf{q}) \overset{\triangle}{=} \mathbf{H}^T W(\mathbf{q})$, $F(.) : C^{N+M-2} \to C^{N+M-1}$. To learn feedforward and feedback filters we use piecewise linear models based on vector quantization and context trees.

The space spanned by $\mathbf{q}$ is partitioned into piecewise disjoint regions and a separate linear model is trained for each region to approximate functions $W(\mathbf{q})$ and $F(\mathbf{q})$ using piecewise linear models.

If the channel is not known, the corresponding equalizers in 3.4 can be directly trained by using adaptive algorithms such as in [8], [2] without channel estimation or piecewise constant partitioning which is studied in this chapter. In this case, one directly applies the adaptive algorithms to feedforward and feedback filters using the received data $\{y_t\}$ and the mean vector $\bar{x}(t)$ as feedback without considering the soft decisions as a priori probabilities. Assuming stationarity of $\bar{x}(t)$, such an adaptive feedforward and feedback filters have Wiener solutions [2]

$$\begin{aligned} \mathbf{w}(t) &= [\mathbf{v}^H(\sigma_n^2\mathbf{I} + \mathbf{H}_r\text{diag}(\mathbf{q})\mathbf{H}_r^H + \mathbf{v}\mathbf{v}^H)^{-1}]^T, \\ \mathbf{f} &= \mathbf{H}^T\mathbf{w}. \end{aligned} \tag{3.5}$$

If PSK modulation is adopted such that $E[|x(t)|^2|\{LLR_a^E(t)\}] = 1$, the filter coefficients in (3.5) is equal the the MMSE equalizer in [30] with time averaged soft information.

In the next section, piecewise linear equalizers are introduced in order to approximate $W(.)$ and $F(.)$. First, adaptive piecewise linear equalizers with a fixed partition of $\mathbb{C}^{N+M-2}$ (where $\mathbf{q} \in \mathbb{C}^{N+M-2}$) are presented. Then, adaptive piecewise linear equalizers using context trees are introduced that can also learn the best partition from a large class of possible partitions of $\mathbb{C}^{N+M-2}$.

3.2 Nonlinear Turbo Equalization Using Piecewise Linear Models

In this section, our aim is to construct piecewise linear equalizer to approximate $\mathbf{w} = W(\mathbf{q})$ and $\mathbf{f} = F(\mathbf{q})$. In order to achieve this, first, we divide the space spanned by $\mathbf{q} \in [0,1]^{N+M-2}$ into disjoint regions V_k, e.g., $[0,\ 1]^{N+M-2} = \cup_{k=1}^K V_k$ for some K and train an independent linear equalizer in each region V_k to obtain a final piecewise linear equalizer. For each region V_k, $k = 1, \ldots, K$, a time varying linear equalizer is assigned to each region as $\mathbf{w}_k(t)$, $\mathbf{f}_k(t)$. At each time t, if $\mathbf{q}(t) \in V_k$, the estimate of the transmitted signal is computed as

$$\begin{aligned} \hat{x}_k(t) &\overset{\triangle}{=} \mathbf{w}_k^T(t)[\mathbf{y}(t) - \mathbf{f}_k^T(t)\bar{\mathbf{x}}(t)] \\ \hat{x}(t) &= \hat{x}_k(t). \end{aligned} \tag{3.6}$$

If K is large and the regions are dense such that $W(\mathbf{q})$ and $F(\mathbf{q})$ can be considered as constant in V_k, say equal to $W(\mathbf{q}_k)$ and $F(\mathbf{q}_k)$ respectively, for some $\mathbf{q}_k \in V_k$ then if the adaptive method used in each region converges successfully, this implies $\mathbf{w}_k(t) \to W(\mathbf{q}_k)$

and $\mathbf{f}_k(t) \to F(\mathbf{q}_k)$ as $t \to \infty$. If these regions are dense and there is enough data to learn the corresponding filter coefficients in each region, then the piecewise model can approximate any smoothly varying $W(\mathbf{q})$ and $F(\mathbf{q})$ [22]. In the following, we first introduce a method to choose the corresponding piecewise regions and then provide adaptive algorithms to train the corresponding piecewise linear models in each region to approximate $W()$ and $F()$. We also quantify the corresponding approximation error in convergence due to using such adaptive piecewise linear filters to learn the corresponding highly nonlinear functions instead of directly using the linear MMSE filter (which is not available).

3.2.1 Piecewise Region Choosing and Linear Modeling

In this section, we introduce a method to choose the corresponding piecewise regions $V_1, \ldots, V_K$. We apply a vector quantization (VQ) algorithm to the sequence of $\{\mathbf{q}(t)\}$, such as the LBG VQ algorithm [**?**]. For VQ algorithm the centroids of the clusters are calculated as

$$\tilde{\mathbf{q}}_k \triangleq \frac{\sum_{t,\mathbf{q}(t)\in V_k} \mathbf{q}(t)}{\sum_{t,\mathbf{q}(t)\in V_k} 1}, \tag{3.7}$$

$$V_k \triangleq \{\mathbf{q} : ||\mathbf{q} - \tilde{\mathbf{q}}_k|| \leq ||\mathbf{q} - \tilde{\mathbf{q}}_j||, j = 1, \ldots, K, j \neq k\}. \tag{3.8}$$

We emphasize that we use a VQ algorithm on $\{\mathbf{q}(t)\}$ to construct the corresponding piecewise regions in order to concentrate on $\mathbf{q}$ vectors that are in $\{\mathbf{q}(t)\}$ since $W(.)$ and $F(.)$ should only be learned around $\mathbf{q} \in \{\mathbf{q}(t)\}$, not for all $\mathbb{C}^{N+M-2}$. After the regions are selected using VQ algorithm, one can calculate the estimate of $x(t)$ at each time t as follows

$$\hat{x}(t) = \hat{x}_i(t) \text{ if } i = \arg\min_k ||\mathbf{q} - \tilde{\mathbf{q}}_k||. \tag{3.9}$$

Then, the corresponding filters in each region can be trained by an adaptive method.

In Table 3.1, we introduce a sequential piecewise linear equalizer that uses the LMS update to train its equalizer filters. One can also use different update methods instead of the LMS update, such as RLS or NLMS [26].

In the pseudocode, the iteration numbers are displayed as superscripts, e.g., $\mathbf{w}_i^{(m)}$, $\mathbf{f}_i^{(m)}$ are the feedforward filter and feedback filter for the mth iteration corresponding to the ith

Table 3.1: A piecewise linear equalizer for turbo equalization. This algorithm requires $O(M+N)$ computations.

A Pseudo-code of Piecewise Linear Turbo Equalizer:

```
%1st iteration:
for $t = 1, \ldots, n$:
     $\hat{x}(t) = w^{(1)T}(t) y(t)$,
     if $t \leq T : e(t) = x(t) - \hat{x}(t)$, elseif $t > T$: $e(t) = Q(\hat{x}(t)) - \hat{x}(t)$, $Q()$ is a quantizer.
    $w^{(1)}(t+1) = w^{(1)}(t) + \mu e(t) y(t)$.
calculate $q(t)$ using the SISO decoder for $t > T$.
% 2nd iteration:
apply LBG VQ [21] algorithm to $q(t)_{t>T}$ to generate $\tilde{q}_k^{(2)}$ , $k = 1, \ldots, K$.
for $k = 1, \ldots, K$: $w_k^{(2)}(0) = w^(1)(n)$, where $w^{(1)}(n)$ is from 1st iteration. (line A)
for $t = 1, \ldots, T$:
    for $k = 1, \ldots, K$:
        $e_k(t) = x(t) - w_k^{(2)T}(t) y(t) - f_k^{(2)T}(t)[\mathbf{I} - diag(\tilde{q}_k^{(2)})]^{1/2} x(t)$, (line B)
        $w_k^{(2)}(t+1) = w_k^{(2)}(t) + \mu e_k(t) y(t)$,
$f_k^{(2)}(t+1) = f_k^{(2)}(t) + \mu e_k(t)[I - diag(\tilde{q}_k^{(2)})]^{1/2} x(t)$.
for $t = T+1, \ldots, n$:
     $i = argmin_k ||q(t) - \tilde{q}^(2)||$,
     $\hat{x}(t) = w_i^{(2)T}(t) y(t) - f_i^{(2)T}(t) \bar{x}(t)$,
     $e(t) = Q(\hat{x}(t)) - \hat{x}(t)$,
     $w^(2)_i(t+1) = w_i^{(2)}(t) + \mu e(t) y(t)$, $f_i^{(2)}(t+1) = f_i^{(2)}(t) + \mu e(t) \bar{x}(t)$.
calculate $q(t)_{t>T}$ using the SISO decoder.
% mth iteration:
apply LBG VQ algorithm to $q(t)_{t>T}$ to generate $\tilde{q}_k^{(m)}, k = 1, \ldots, K$.
for $k = 1, \ldots, K$:
$w_k^{(m)}(0) = w_j^{(m-1)}(n), f_k^{(m)}(0) = f_j^{(m-1)}(n)$, where $j = argmin_i ||\tilde{q}_k^{(m)} - \tilde{q}_i^{(m-1)}||$. (line C)
for $t = 1, \ldots, T$:
     for $k = 1, \ldots, K$:
         $e_k(t) = x(t) - w_k^{(m)T}(t) y(t) - f_k^{(m)T}(t)[I - diag(\tilde{q}_k)]^{1/2} x(t)$,
         $w_k^{(m)}(t+1) = w_k^{(m)}(t) + \mu e_k(t) y(t)$, $f_k^{(m)}(t+1) = f_k^{(m)}(t) + \mu e_k(t)[I - diag(\tilde{q}_k^{(m)})]^{1/2} x(t)$.
for $t = T+1, \ldots, n$,
     $i = argmin_k ||q(t) - \tilde{q}_k^{(m)}||$,
     $\hat{x}(t) = w_i^{(m)T}(t) y(t) - f_i^{(m)T}(t) \bar{x}(t)$,
     $e(t) = Q(\hat{x}(t)) - \hat{x}(t)$,
$w_i^{(m)}(t+1) = w_i^{(m)}(t) + \mu e(t) y(t)$, $f_i^{(m)}(t+1) = f_i^{(m)}(t) + \mu e(t) \bar{x}(t)$.
calculate $q(t)_{t>T}$ using the SISO decoder.
```

region, respectively. μ is the step size of the LMS updates. The size of the training data is denoted as T. After the training data is used, the adaptive methods work in decision directed (DD) mode [26]. In the first iteration, since there is no *a priori* probabilities only the feedforward filter $\mathbf{w}(t)$ is trained, i.e., $\hat{x}(t) = \mathbf{w}^T(t)\mathbf{y}(t)$ without any piecewise regions or mean vectors. The feedforward filter $\mathbf{w}(t)$ trained on $\mathbf{y}(t)$ without a priori probabilities converges to [2] (assuming zero variance in convergence)

$$\mathbf{w}(t) \to \mathbf{w}_0 \overset{\triangle}{=} [\mathbf{v}^H(\sigma_n^2\mathbf{I} + \mathbf{H}_r\mathbf{H}_r^H + \mathbf{v}\mathbf{v}^H)^{-1}]^T \;\; ast \to \infty, \tag{3.10}$$

which is linear MMSE feedforward filter in (3.2) with $\mathbf{Q}(t) = \mathbf{I}$.

After the first iteration, $\{\mathbf{q}(t)\}$ becomes available. To get the centroids of the K regions we apply VQ algorithm. Then, for each region k, we train feedforward and feedback filters by LMS algorithm and construct the estimated data as in (3.6). In the beginning of the second iteration, that is, in line A, each feedforward filter is initialized by the feedforward filter trained in the first iteration. Furthermore, although the linear equalizers should have the form $\mathbf{w}_k^T(t)\mathbf{y}(t) - \mathbf{f}_k^T\bar{\mathbf{x}}(t)$, since we have the correct $x(t)$ in the training mode for $t = 1, \ldots, T$, the algorithms are trained using $\mathbf{w}_k^T(t)\mathbf{y}(t) - \mathbf{f}_k^T[\mathbf{I} - \mathrm{diag}(\tilde{\mathbf{q}}_k)]^1/2\bar{\mathbf{x}}(t)$ in line B, to incorporate the uncertainty during training [18]. After the second iteration, in the beginning of each iteration, in line C, the linear equalizers in each region, say k, are initialized using the filters trained in the previous iteration that are closest to the kth region, i.e.,

$$\begin{aligned} &j = \mathrm{argmin}_i ||\tilde{\mathbf{q}}_k^m - \tilde{\mathbf{q}}_i^{(m-1)}||, \\ &\mathbf{w}_k^{(m)}(0) = \mathbf{w}_j^{(m-1)}(n) \\ &\mathbf{f}_k^{(m)}(0) = \mathbf{f}_j^{(m-1)}(n). \end{aligned} \tag{3.11}$$

For each region k, a separate linear equalizer is trained using the part of the data $\{x(t)\}$ which belongs to V_k, i.e., using $x(t)$s with $\mathbf{q}(t)$ as the closest vector to $\tilde{\mathbf{q}}_k(t)$. Hence, assuming large K with dense regions we have $\mathbf{q}(t) \approx \tilde{\mathbf{q}}_k$ if $\mathbf{q}(t) \in V_k$.

For each region k we want to find the limiting vector for the linear filters as a result of

LMS training as $t \to \infty$. Since

$$
\begin{aligned}
& E\{[\mathbf{x}(t)-\bar{\mathbf{x}}(t)][\mathbf{x}(t)-\bar{\mathbf{x}}(t)]^H|\mathbf{q}(t)=\tilde{\mathbf{q}}_k\} \\
&= E\{[\mathbf{x}(t)-\bar{\mathbf{x}}(t)][\mathbf{x}(t)-\bar{\mathbf{x}}(t)]^H|\{LLR_a^E(t)\},\mathbf{q}(t)=\tilde{\mathbf{q}}_k\}|\mathbf{q}(t)=\tilde{\mathbf{q}}_k\} \\
&= \operatorname{diag}\{[\tilde{\mathbf{q}}_k(1),\ldots,\tilde{\mathbf{q}}_k(M+N_2-1),1,\tilde{\mathbf{q}}_k(M+N_2),\ldots,\tilde{\mathbf{q}}_k(M+N-2)]\}
\end{aligned}
$$

assuming stationarity on $\bar{x}(t)$. Then for each region k

$$
\begin{aligned}
\mathbf{w}_k(t) \to \mathbf{w}_{k,0} &\overset{\triangle}{=} [\mathbf{v}^H(\sigma_n^2\mathbf{I}+\mathbf{H}_r\tilde{\mathbf{Q}}_k\mathbf{H}_r^H+\mathbf{v}\mathbf{v}^H)^{-1}]^T,\ t\to\infty \\
\mathbf{f}_k(t) \to \mathbf{f}_{k,0} &\overset{\triangle}{=} \mathbf{H}^T\mathbf{w}_{k,0},\ \ t\to\infty,
\end{aligned} \tag{3.12}
$$

where $\tilde{\mathbf{Q}}_k \overset{\triangle}{=} \operatorname{diag}(\tilde{\mathbf{q}}_k)$, assuming zero variance in convergence. Hence, at each time t, the difference between the MSE of the equalizer in (3.12) and the MSE of the linear MMSE equalizer in (3.2) is given by

$$
||\mathbf{w}_{k,0}^T\mathbf{H}_r\mathbf{Q}(t)\mathbf{H}_r^H\mathbf{w}_{k,0}^*+\sigma_n^2\mathbf{w}_{k,0}^T\mathbf{w}_{k,0}^*-[1-\mathbf{v}^H(\sigma_n^2\mathbf{I}+\mathbf{H}_r\mathbf{Q}(t)\mathbf{H}_r^H+\mathbf{v}\mathbf{v}^H)^{-1}\mathbf{v}]|| \leq O(||\mathbf{q}(t)-\tilde{\mathbf{q}}_k||), \tag{3.13}
$$

as shown in Appendix B. As the number of regions K increases $||\mathbf{q}(t)-\tilde{\mathbf{q}}_k|| \to 0$, the MSE of the converged adaptive filter accurately approximates the MSE of the linear MMSE equalizer.

Note that in the pseudocode, the partition of the space $\mathbf{q} \in [0,1]^{N+M-2}$ is fixed i.e., piecewise regions are fixed in the start of the equalization after the VQ algorithm, and we sequentially learn a different linear equalizer for each region. Since the equalizers are sequentially learned with a limited amount of data, this may cause training problems if there is not enough data in each region to train on. Even if K is increased to increase the approximation power, if there is not enough data to learn the linear equalizers for each region, this may penalize the performance. To mitigate this, one can use a piecewise model with smaller K in the beginning of the learning and gradually increase the number of regions K if enough data is available. In the next section, the context tree weighting method is introduced which intrinsically applies such weighting among different piecewise models based on the performance of each region, hence, allowing the boundaries of the piecewise regions to be design parameters.

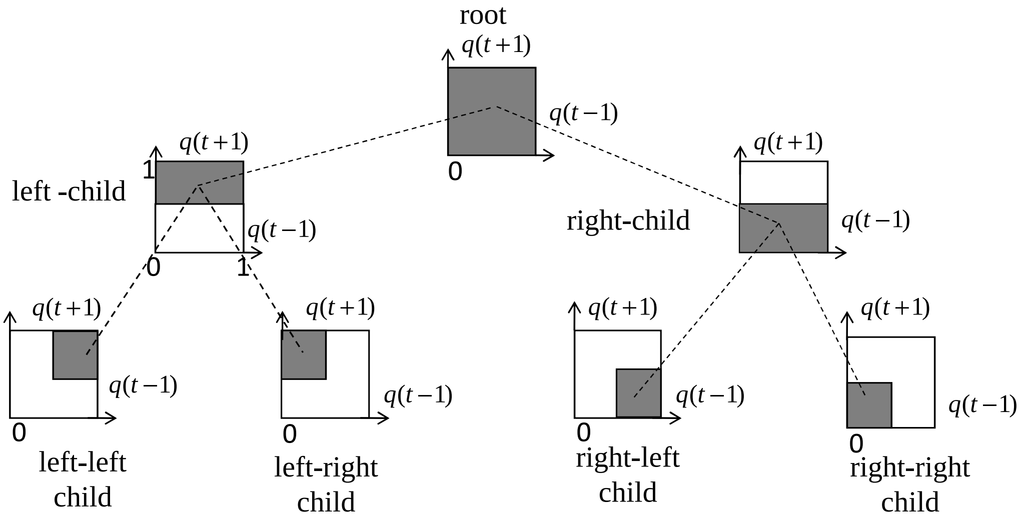


Figure 3.2: A full binary context tree with depth, $D = 2$, with 4 leaves. The leaves of this binary tree partitions $[0, 1]^2$, i.e., $[q(t-1)\ q(t+1)] \in [0, 1]^2$, into 4 disjoint regions.

3.3 Piecewise Linear Turbo Equalization Using Context Trees

In this section, we present a binary context tree to partition the space $[0, 1]^{N+M-2}$ into disjoint regions. Our aim is to construct piecewise linear equalizer which can choose both the piecewise regions as well as the equalizer coefficients in these regions based on the equalization performance. Fig. 3.2 shows a binary context tree of depth $D = 2$ with 4 leaves. Starting from the root node which is shown with dark square, we have left and a right child. Each left and right hand child have their own left and right hand children. Each internal node on this tree represents a region which is the union of regions assigned to its children. At the end, this branching yields a binary tree of depth D with a total of $2^{D+1} - 1$ nodes including the leaves at depth D.

On a binary tree of depth D, a doubly exponential number $m \approx 1.5^{2^D}$ of "‘complete” subtrees can be defined as in Fig 3.3. We call a subtree complete if the union of regions assigned to its leaves results $[0, 1]^{N+M-2}$. Note that a leaf of a subtree is either an internal node or the leaf of the original tree. For example, for a subtree i, if the regions assigned to its leaves are labeled as $V_{1,i}, \ldots, V_{K_i,i}$, where K_i is the number of leaves of the subtree i, then $[0,\ 1]^{N+M-2} = \cup_{k=1}^{K_i} V_{k,i}$. Each $V_{k,i}$ of the subtree corresponds to a node in the original tree. Thus a complete subtree with the regions assigned to its leaves defines a complete "partition" of $[0,\ 1]^{N+M-2}$.

Consider Fig. 3.3, on a binary context tree of depth $D = 2$, there are 5 different partitions of $[0, 1]^2$ as $\Gamma_1, \ldots, \Gamma_5$ labeled as $\Gamma_1, \ldots, \Gamma_5$. For example, the partition Γ_1 corresponds to the root node with only one region covering the space $[0, 1]^2$ and Γ_5 corresponds to all leaf nodes of the original tree. For each partition, $\Gamma_i = \{V_{1,i}, \ldots, V_{K_i,i}\}$ and $\cup_l = 1^{K_i} V_{k,i} = [0, 1^{N+M-2}$. We next describe the construction of such a binary context tree and the selection of the regions to partition $[0, 1^{N+M-2}$ for the proposed nonlinear modeling.

As in the previous section, piecewise regions are selected with the LBG VQ algorithm. The context tree is built over these piecewise regions as follows. Suppose the LBG VQ algorithm is applied to $\{\mathbf{q}(t)\}$ with $K = 2^D$ to generate K regions [21]. The LBG VQ algorithm is an iterative algorithm that alternatively solves (3.7) and (3.8). The LBG VQ algorithm uses a tree notion similar to the context tree introduced in Fig. 3.3 such that the algorithm starts from a root node which calculates the mean of all the vectors in $\{\mathbf{q}(t)\}$ as the root codeword, and binary splits the data as well as the root codeword into two segments. Then, these newly constructed codewords are iteratively used as the initial codebook of the split segments. These two codewords are then split in four and the process is repeated until the desired number of regions are obtained. At the end, this binary splitting yield 2^D regions with the corresponding centroids $\tilde{\mathbf{q}}_i, i = 1, \ldots, 2^D$, which are assigned to the leaves of the context tree. Note that since each couple of the leaves (or nodes) come from a parent node after a binary splitting, codewords assigned to these parent nodes are stored as the internal nodes of the context tree. Hence, in this sense, the LBG VQ algorithm intrinsically constructs the context tree. However, note that, at each turn, even though the initial centroids at each splitting directly come from the parent node in the original LBG VQ algorithm, the final regions of the leaf nodes while minimizing distortion by iterating (3.7) and (3.8) may invade the regions of the other parents nodes, i.e., the union of regions assigned to the children of the split parent node can be different than region assigned to the parent node. Note that one can modify the LBG algorithm with the constraint that the region of the children nodes should be optimized within the regions of the their parent nodes. However, this constraint may deteriorate the performance due to more quantization error.

For a context tree of depth D, one can define $m \approx 1.5^{2^D}$ different partitions of the

Table 3.2: A context tree based turbo equalization. This algorithm requires O(D(M + N)) computations

A Pseudo-code of Piecewise Linear Turbo Equalizer Using Context Trees:

```
% 1st iteration:
for $t = 1, \ldots, n$:
    $\hat{x}_{ctw}(t) = w^{(1)T}(t)y(t)$,
    if $t = T$: $e(t) = x(t) - \hat{x}(t)$.
    elseif $t > T$: $e(t) = Q(\hat{x}(t)) - \hat{x}(t)$, $Q()$ is a quantizer.
   $w^{(1)}(t+1) = w^(1)(t) + \mu e(t)y(t)$.
calculate $q(t)$ using the SISO decoder for $t > T$. % mth iteration:
apply LBG VQ algorithm to $q(t)_{t>T}$
    to generate $\tilde{q}_k^{(2)}, k = 1, \ldots, 2^D$ and $\tilde{q}_k^{(2)}, k = 2^D + 1, \ldots, 2^{D+1} - 1$.
if $m == 2$:
    for $k = 1, \ldots, 2^{D+1} - 1$:
$w_k^{(2)}(1) = w^{(1)}(n)$, where $w^{(1)}(n)$ is from 1st iteration.
if $m \geq 3$:
    for $i = 1, ..., 2^{D+1} - 1$:
$w_k^{(m)}(1) = w_j^{(m-1)}(n), f_k^{(m)}(1) = f_j^{(m-1)}(n)$, where $j = \text{argmin}_i ||\tilde{q}_k^{(m)} - \tilde{q}_i^{(m-1)}||$.
for $k = 1, \ldots, 2^{D+1} - 1$: $A_k(0) = 1, B_k(0) = 1$. (line A)
for $t = 1, \ldots, T$:
    for $i = 1, \ldots, 2^D$:
        $\bar{x}(t) \overset{\triangle}{=} [\mathbf{I} - diag(\tilde{q}_k^{(m)})]^{1/2} x(t)$ %to consider uncertainty during training
         $l_{(1)} = i$, where $l$ corresponds dark nodes starting from the leaf node $i$
         $\eta_1(t) = 1/2$.
         for $l = 2, \ldots, D+1$:
             $\eta_l(t) = \frac{1}{2} A_s(t-1)\eta_{l-1}(t)$, where $s$ is a sibling node of $l(l)$, i.e., $V_s \cup V_{l(l)} = V_{l(l-1)}$,
              $\beta_l(t) = \eta_l(t) B_{l(l)}(t-1) A_{l_{(1)}}(t-1)$. (line B)
         for $l = D+1, \ldots, 1$:          $e_{l(l)}(t) = x(t) - [w_{l(l)}^{(m)T}(t)y(t) - f_{l(l)}^{(m)T}(t)\bar{x}(t)]$,
              $B_{l(l)}(t) = B_{l(l)}(t-1)exp(-c||e_{l(l)}(t)||^2)$, % where $c$ is a positive constant, (line C)
              if $l = D+1 : A_{l(l)}(t) = B_{l(l)}(t)$, (line D)
              else: $A_{l(l)}(t) = \frac{1}{2} A_{l(l),l}(t-1) A_{l(l),r(t-1)} + \frac{1}{2} B_{l(l)}(t), (lineE)$
             % where $(l(l), l)$ and $(l(l), r)$ are the left and right hand children of $l(l)$, respectively,
              $w_{l(l)}^{(m)}(t+1) = w_{l(l)}^{(m)}(t) + \mu e_{l(l)}(t)y(t)$, $f_{l(l)}^{(m)}(t+1) = f_{l(l)}^{(m)}(t) + \mu e_{l(l)}(t)\bar{x}(t)$.
for $t = T+1, \ldots, n$:      $i = \text{argmin}_k ||q(t) - \tilde{q}_k^{(m)}||, k = 1, \ldots, 2^D$.
    find nodes that $i$ belongs to and store them in $l$ starting from the leaf node $i$, i.e., $l(1) = i$.
   $\eta_1(t) = 1/2$,
   for $l = 2, \ldots, D+1$:
       $\eta_l(t) = \frac{1}{2} A_s(t)\eta_{l-1}(t)$.
       $\beta_l(t) = \eta_l(t) B_{l(l)}(t-1) A_{l(1)}(t-1)$.
   $\hat{x}_{ctw}(t) = \sum_{k=1}^{D+1} \beta_k(t)[w_{l(k)}^{(m)T}(t)y(t) - f_{l(k)}^{(m)T}(t)\bar{x}(t)]$, (line F)
    for $l = D+1, \ldots, 1$:          $e_{l(l)}(t) = Q(\hat{x}_{ctw}(t)) - [w_{l(l)}^{(m)T}(t)y(t) - f_{l(l)}^{(m)T}(t)\bar{x}(t)]$, (line G)
        $B_{l(l)}(t+1) = B_{l(l)}(t-1)exp(-c||e_{l(l)}(t)||^2)$,
        if $l = D+1$: $A_{l(l)}(t) = B_{l(l)}(t)$,
        else: $A_{l(l)}(t) = \frac{1}{2} A_{l(l),l}(t-1) A_{l(l),r}(t-1) + \frac{1}{2} B_{l(l)}(t)$,
        $w_{l(l)}^{(m)}(t+1) = w_{l(l)}^{(m)}(t) + \mu e_{l(l)}(t)y(t)$, $f_{l(l)}^{(m)}(t+1) = f_{l(l)}^{(m)}(t) + \mu e_{l(l)}(t)\bar{x}(t)$.
calculate $q(t)_{t>T}$ using the SISO decoder.
```

space of $\mathbf{q}$ and construct a piecewise linear equalizer as in Table 3.1 for each partition. However, there exist a double exponential number of different piecewise linear models that one can choose from. For each such partition, one can train and use a piecewise linear model. However, the best piecewise model with the best partition is not known *a priori*.

Although one can construct a doubly exponential number m of piecewise linear equalizers as in Table 3.1, note that all these piecewise linear equalizers are constructed using subsets of nodes $\rho \in \{1, \ldots, 2^{D+1} - 1\}$. Now, suppose we number each node on this context tree and assign a linear equalizer to each node as

$$\hat{x}_\rho(t) = \mathbf{w}_\rho^T \mathbf{y}(t) - \mathbf{f}_\rho^T \bar{\mathbf{x}}(t) \tag{3.14}$$

The linear equalizer coefficients w_ρ, f_ρ that are assigned to node ρ, train only on the data assigned to that node as in Table 3.1. That is, if $\mathbf{q}(t) \in V_\rho$ then w_ρ and f_ρ are updated. Then, the estimate of the desired data using the piecewise linear equalizer corresponding to the partition Γ_i, is defined as follows. If $\mathbf{q}(t) \in V_{k,i}$ and ρ is the node that is assigned to $V_{k,i}$ then

$$\hat{x}_{\Gamma_i}(t) = \hat{x}_\rho(t) = \mathbf{w}_\rho^T \mathbf{y}(t) - \mathbf{f}_\rho^T \bar{\mathbf{x}}(t). \tag{3.15}$$

One of these partitions, with the given piecewise adaptive linear model $\hat{x}_{\Gamma_i}$ achieves the minimal loss. However, we want to emphasize that one needs to try a doubly exponential number of different partitions to find that partition with the best fit to the data.

We next introduce an algorithm that achieves the performance of the best partition with the best linear model with complexity only linear in depth of the context tree per sample, i.e., complexity $O(D(2N + M))$ instead of $O((1.5)^{2^D} D(2N + M)$. We emphasize that the piecewise model that corresponds to the union of the leaves, i.e., the finest partition, has the finest partition of the space of variances. Hence, it has the highest number of regions and parameters to model the nonlinear dependency. However, note that at each such region, the finest partition needs to train the corresponding linear equalizer that belongs to that region. As an example, the piecewise equalizer with the finest partition may not yield satisfactory results in the beginning of the adaptation if there are not enough data to train all the model parameters. In this sense the context tree algorithm adaptive weights coarser and finer models based on their performances.

The pseudo-code of the introduced algorithm is given in Table 3.2. The context tree based equalization algorithm hypothetically builds all $\hat{x}_{\Gamma_i}(t)$, $i = 1, \ldots, m$, piecewise linear equalizers and run these equalizers on parallel on the received data. At each time t, the final estimation $\hat{x}_{ctw}(t)$ is constructed as a weighted combination of all the outputs $\hat{x}_{\Gamma_i}(t)$ of these piecewise linear equalizers, where the combination weights are calculated proportional to the performance of each equalizer $\hat{x}_{\Gamma_i}(t)$ on the past data. However, as shown in (3.15), although there are m different piecewise linear algorithms, at each time t, each $\hat{x}_{\Gamma_i}(t)$ is equal to one of the D node estimations that $\mathbf{q}(t)$ belongs to. In Table 3.2, the update of each m piecewise linear models and their performance based combination weights with computational complexity only linear in the depth of the context tree is described.

For the context tree algorithm, since there are no a priori probabilities in the first iteration, the first iteration of Table 3.2 is the same as the first iteration of Table 3.1. After the first iteration, to incorporate the uncertainty during training as in Table 3.1, the context tree algorithm is implemented by using weighted training data corresponding to 2^D leaf nodes for not only updating filter coefficients but also updating context tree weights. At each time $t > T$, $\hat{x}_{ctw}(t)$ constructs its nonlinear estimation of $x(t)$ as follows. First the regions (or nodes) that $\mathbf{q}(t)$ belongs to are found. Note that due to the tree structure of regions, one needs to only find the leaf node that $\mathbf{q}(t)$ belongs to and collect all the parent nodes towards the root node. The nodes that $\mathbf{q}(t)$ belongs to are stored in $\boldsymbol{l}$. The final estimate $\hat{x}_{ctw}(t)$ is constructed as a weighted combination of the estimates generated in these nodes, i.e., $\hat{x}_\rho(t)$, $\rho \in \boldsymbol{l}$, where the weights are functions of the performance of the node estimates in previous samples.

At each time t, $\hat{x}_{ctw}(t)$ requires $O(\ln(D))$ calculations to find the leaf that $\mathbf{q}(t)$ belongs to. Then, $D + 1$ node estimations, $\hat{x}_\rho(t)$, $\rho \in \boldsymbol{l}$, are calculated and the equalizers at these nodes should be updated with $O(2N + M)$ computations. The final weighted combination is produced with $O(D)$ computations. Thus the overall computaional complexity of the context tree algorithm is $O(D(2N + M))$ at each time t. For this algorithm, we have the following result.

Theorem 1: Let $\{x_t\}$, $\{n_t\}$, and $\{y_t\}$represent the transmitted, noise and received signals and $\{\mathbf{q}(t)\}$ represent the sequence of variances constructed using the a priori probabilities for each constellation produced by the SISO decoder. Let $\hat{x}_\rho(t)$, $\rho = 1, \ldots, 2^{D+1} - 1$, are estimates of $x(t)$ produced by the equalizers assigned to each node on the context tree. The algorithm $\hat{x}_{ctw}(t)$, when applied to $\{y_t\}$, for all n achieves

$$\sum_{t=1}^{n} (x(t) - \hat{x}_{ctw}(t))^2 \leq \min_{\Gamma_i} \{ \sum_{t=1}^{n} [x(t) - \hat{x}_{\Gamma_i}(t)]^2 + 2K_i - 1 \},$$

for all i, $i = 1, \ldots, m \approx (1.5)^{2^D}$, assuming perfect feedback in decision directed mode i.e., $Q(\hat{x}(t)) = x(t)$ when $t \geq T$, where $\hat{x}_{\Gamma_i}(t)$ is the equalizer constructed as $\hat{x}_{\Gamma_i}(t) = \hat{x}_\rho(t)$ and ρ is the node which is assigned to the volume in $\Gamma_i = \{V_{1,i}, \ldots, V_{K_i,i}\}$ such that $\mathbf{q}(t)$ belongs.

If RLS update is used instead of the LMS update, (3.3) yields

$$\begin{aligned} \sum_{t=1}^{n} (x(t) - \hat{x}_{ctw}(t))^2 \leq \min_{\Gamma_i} \{ \min_{\mathbf{w}_{k,i} \in \mathbb{C}^N, \mathbf{f}_{k,i} \in \mathbb{C}^{N+M-1}} \sum_{t=1}^{n} E\{[x(t) - \mathbf{w}_{s_i(t),i}^T \mathbf{y}(t) - \mathbf{f}_{s_i(t-1),i}^T \bar{\mathbf{x}}(t)]^2\} \\ + O((2N + M)ln(n)) + 2K_i - 1\}. \end{aligned} \tag{3.16}$$

where $s_i(t)$ is an indicator variable for Γ_i such that if $\mathbf{q}(t) \in V_{k,i}$ then $s_i(t) = k$

The outline of the proof is given in Appendix B.

Remark: We observe from (3.3) that the context tree algorithm achieves the performance of the best sequential algorithm among a doubly exponential number of possible algorithms. Note that the bound in (3.3) holds uniformly for all i, however the bound is the largest for the finest partition corresponding to all leaves (however still $O(1)$). Also, we observe from (3.16) that the context tree algorithm achieves the performance of even the best piecewise linear model, independently optimized in each region, for all i, if the node estimators are selected as certain adaptive algorithms.

3.3.1 MSE Performance of the Context Tree Equalizer

To get the MSE performance of the context tree equalizer, we observe that the result (3.16) in the theorem is uniformly true for any sequence $\{x(t)\}$. Hence, as a corollary to the theorem, taking the expectation of both sides of (3.16) with respect to any distribution on $\{x(t)\}$ yields the following corollary:

Corollary:

$$\sum_{t=1}^{n} E\{(x(t)-\hat{x}_{ctw}(t))^2\} \leq \min_{\Gamma_i}\{\min_{\mathbf{w}_{k,i}\in\mathbb{C}^N,\mathbf{f}_{k,i}\in\mathbb{C}^{N+M-1}} \sum_{t=1}^{n} E\{[x(t)-\mathbf{w}^T_{s_i(t),i}\mathbf{y}(t)-\mathbf{f}^T_{s_i(t-1),i}\bar{\mathbf{x}}(t)]^2\} + O((N+M)ln(n)) + 2K_i - 1\}. \tag{3.17}$$

Note that the minimizer vectors $\mathbf{w}_{k,i}$ and $\mathbf{f}_{k,i}$ at the right hand side of (3.17) minimize the sum of all the MSEs. Hence, the corollary does not relate the MSE performance of the CTW equalizer to the MSE performance of the linear MMSE equalizer given in (3.2). However, if we assume that the adaptive filters trained at each node converge to their optimal coefficient vectors with zero variance and if D and n are sufficiently large, we have for piecewise models such as for the finest partition

$$\sum_{t=1}^{n}(x(t)-\hat{x}_{\Gamma_{|K|}}(t))^2 \approx \sum_{t=1}^{n}\{[x(t)-\mathbf{w}^T_{s_{|K|}(t),|K|,o}\mathbf{y}(t)-\mathbf{f}^T_{s_{|K|}(t),|K|,o}\bar{\mathbf{x}}(t)]^2\}, \tag{3.18}$$

where we assumed that, for notational simplicity, the $|K|$th partition is the finest partition, $\mathbf{w}_{s_{|K|}(t),|K|,o}$ and $\mathbf{f}_{s_{|K|}(t),|K|,o}$ are the MSE optimal filters (if defined) corresponding to the regions assigned to the leaves of the context tree. Note that we require D to be large so that we can assume $\mathbf{q}(t)$ to be constant at each region such that these MSE optimal filters are well-defined. Since (3.16) is correct for all partitions and for the minimizer $\mathbf{w}$,$\mathbf{f}$ vectors, (3.16) holds for any $\mathbf{w}$ and $\mathbf{f}$ pairs including $\mathbf{w}_{s_{|K|}(t),|K|,o}$ and $\mathbf{f}_{s_{|K|}(t),|K|,o}$ pair.

Then by taking the expectation of the equation (3.16) and by using (3.18) we obtain

$$\frac{1}{n}\sum_{t=1}^{n}E\{[x(t)-\hat{x}_{ctw}(t)]^2\} \leq \frac{1}{n}\sum_{t=1}^{n}E\{[x(t)-\mathbf{w}^T_{s_|K|(t),|K|,o}\mathbf{y}(t)-\mathbf{f}^T_{s_|K|(t),|K|,o\bar{\mathbf{x}}(t)}]^2+O(\frac{2^{D+1}}{n})\}. \tag{3.19}$$

Using the definition of the MSE of each node

$$\begin{aligned}
&\frac{1}{n}\sum_{t=1}^{n} E\{[x(t)-\hat{x}_{ctw}(t)]^2\} \\
&\leq \frac{1}{n}\sum_{t=1}^{n}\{\mathbf{w}^T_{s_{|K|}(t),|K|,o}\mathbf{H}_r\mathbf{Q}(t)\mathbf{H}_r^H\mathbf{w}^*_{s_{|K|}(t),|K|,o} \\
&+\sigma_n^2\mathbf{w}^T_{s_{|K|}(t),|K|,o}\mathbf{w}^*_{s_{|K|}(t),|K|,o}+O(\frac{2^{D+1}}{n})\} \\
&\leq \frac{1}{n}\sum_{t=1}^{n}\{min_{\mathbf{w},\mathbf{f}}E\{[x(t)-\mathbf{w}^T\mathbf{y}(t)-\mathbf{f}^T\bar{\mathbf{x}}(t)]^2|\mathbf{q}(t)+O(\frac{1}{2^D})\}\}+O(\frac{2^{D+1}}{n}), \qquad (3.20)
\end{aligned}$$

where the last equation line follows since for large D, the MSE in each node is bounded as in equation 3.13. Furthermore, $O(||\mathbf{q}(t)-\tilde{\mathbf{q}}_l||)$ can be upper bounded by $O(\frac{1}{2^D})$ assuming large enough D. Hence, as $D\rightarrow\infty$, the context tree algorithm asymptotically achieves the performance of the linear MMSE equalizer.

3.4 Simulations

For this chapter, we demonstrate the performance of the introduced algorithms through numerical examples under different scenarios. The set of experiments involve channel examples from [24] (Chapter 10). Rate 1/2 with constraint length 3 convolutional code is used. The coded bits are shuffled by random interleaving.

In the first set of experiments, we use the time invariant channel

$$\mathbf{h}=[0.227, 0.46, 0.688, 0.46, 0.227]^T$$

with the training size $T=1024$ and data length 5120 (excluding the training part). The BERs and MSE curves are calculated over 20 independent trials. The decision directed (DD) mode is used for all the LMS algorithms, e.g., for the ordinary LMS turbo equalizer we compete against and for all the node filters on the context tree. Our calculation of the extrinsic LLR at the output of the ordinary LMS algorithm is based on [4]. For all LMS filters, we use $N1=9$, $N2=5$, length $N+M-1=19$ feedback filter. The learning rates for the LMS algorithms are set to $\mu=0.001$. This learning rate is selected to guarantee the convergence of the ordinary LMS filter in the training part. The same learning rate is directly used on the context tree without tuning. In Fig. 3.4 (a), we demonstrate the time evaluation of the weight vector for the ordinary LMS turbo equalization algorithm in the first turbo iteration. Also the plot of the convolution of the h and the converged weight vector of the LMS algorithm at the end of the first iteration is given in Fig. 3.4 (b). In Fig. 3.4 (a), BERs for an ordinary LMS algorithm, a context-tree equalization algorithm with $D = 2$ is given in Table 3.2 and the piecewise equalization algorithm with the finest partition, i.e., $x_{T_{|K|}}(t)$, on the same tree. Note that the piecewise equalizer with the finest partition, i.e., Γ_5, in Fig. 3.3, has the finest partition with the highest number of linear models, i.e., 2^D independent filters, for equalization. However, we emphasize that all the linear filters in the leaves should be sequentially trained for the finest partition. Hence, as explained in Section 3.3, the piecewise model with the finest partition may yield inferior performance compared to the CTW algorithm that adaptively weights all the piecewise models based on their performance. We observe that the context tree equalizer outperforms the ordinary LMS equalizer and the equalizer corresponding to the finest partition for these simulations. In Fig. 3.4 (b), we plot the weight evaluation of the context tree algorithm, i.e., the combined weight in line F of Table 3.2 to show the convergence of the CTW algorithm. The context tree algorithm, unlike the finest partition piecewise model, adaptively weights different partitions in each level. To see this, in Fig. 3.4 (a), we plot weights assigned to each level in depth $D = 2$ context tree. We also plot the time evaluation of the performance measures $A_\rho(t)$ in Fig. **??** (b). We observe that the context tree algorithm, as expected, at the start of the equalization divides the weights fairly uniform among the partitions or node equalizers. However, naturally, as the training size increases, when there is enough data to train all the node filters, than the context tree algorithm favors the models with better performance. To see the effect of depth on the performance of the context tree equalizer, we plot the for the same channel, BERs corresponding to context tree equalizers of depth, $D = 1$, $D = 2$ and $D = 3$ in Fig. (a). We observe that as the depth of the tree increases the performance of the tree equalizer gets better for these depth range. However, note that the computational complexity of the CTW equalizer is directly proportional to the depth. As the last set of experiments, we perform the same set of experiments on a randomly generated channel of length 7 and plot the BERs in Fig. (b). We observe the similar improvement in BER for this randomly generated channel for these simulations.

3.5 Conclusions

In this chapter, an adaptive nonlinear turbo equalization algorithm using context trees is introduced in order to model the nonlinear dependency of the linear MMSE equalizer on the soft information generated from the decoder. The CTW algorithm is used to partition the space of variances, which are time dependent and generated from the soft information, and train a separate linear model for each region. It is demonstrated that the introduced algorithm asymptotically achieves the performance of the best piecewise model defined on this context tree with a computational complexity only in the order of an ordinary linear equalizer. Also, the convergence of the MSE of the CTW algorithm to the MSE of the linear minimum MSE estimator is shown as the depth of the context tree and the data length increase.

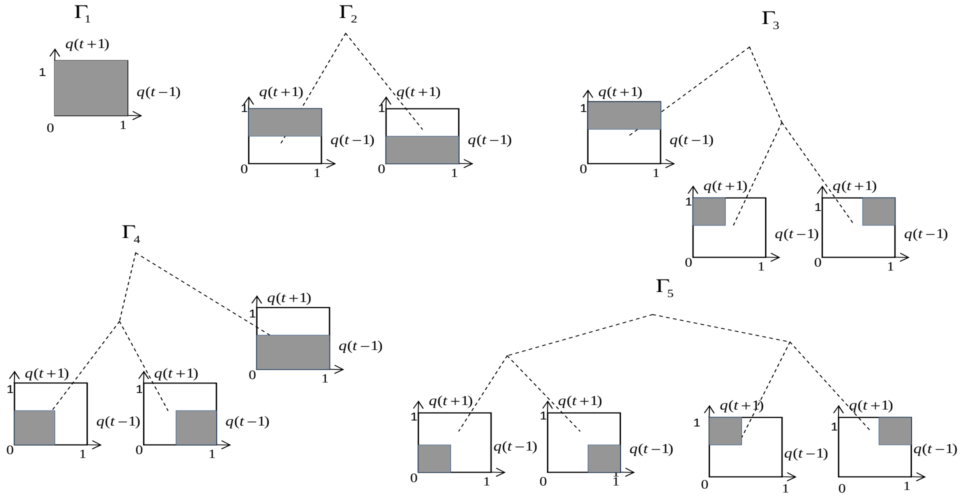
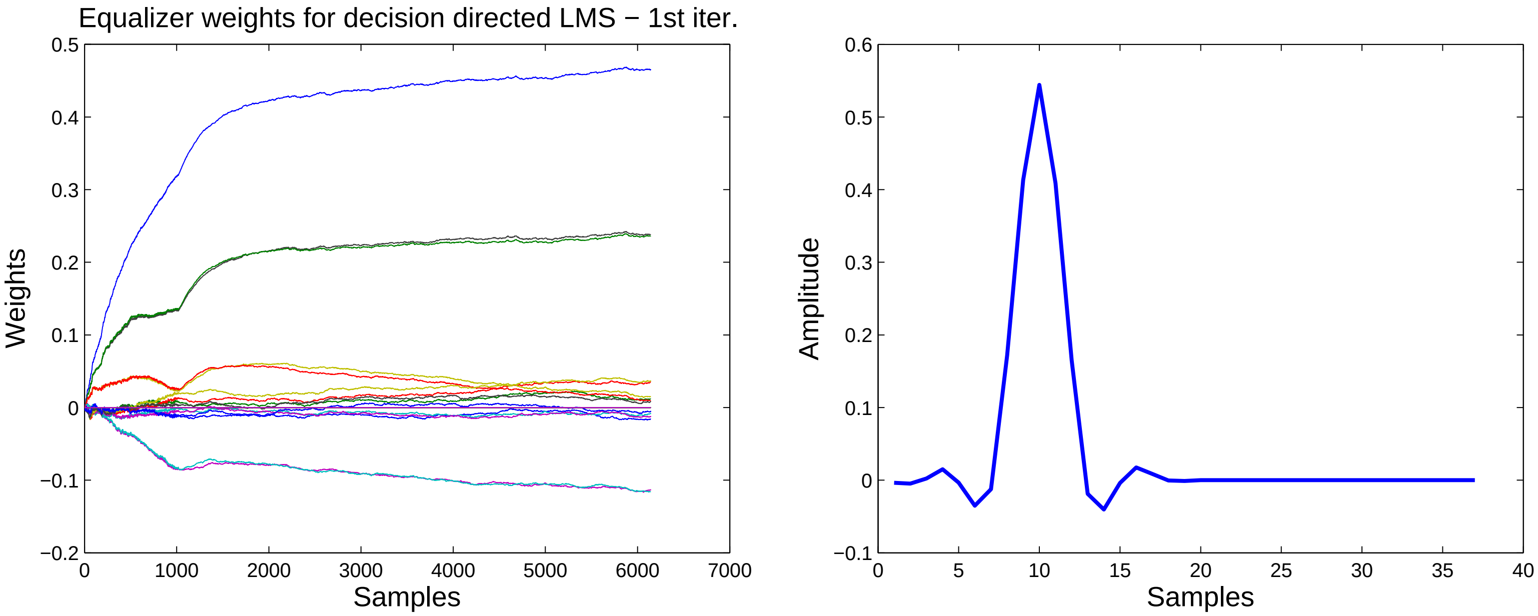
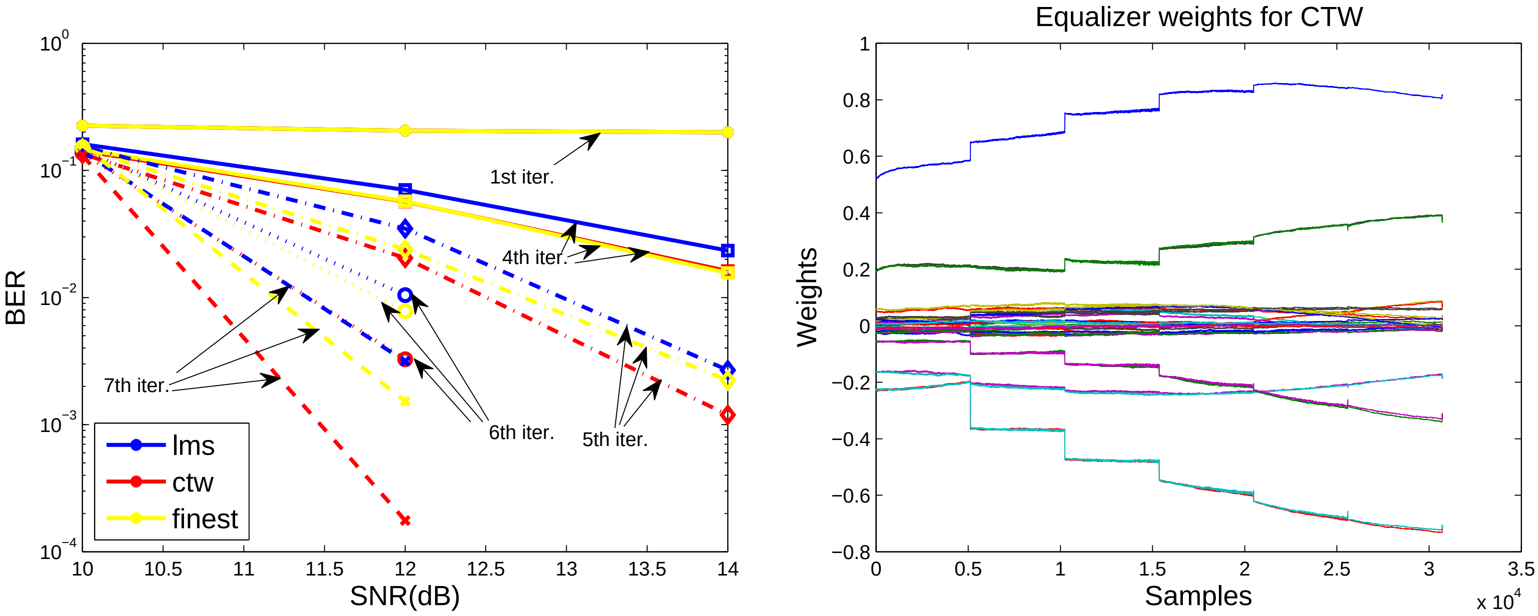


Figure 3.3: All partitions of $[0,1]^2$ using binary context tree with $D = 2$. Given any partition, the union of the regions represented by the leaves of each partition is equal to $[0,1]^2$.



(a) (b)

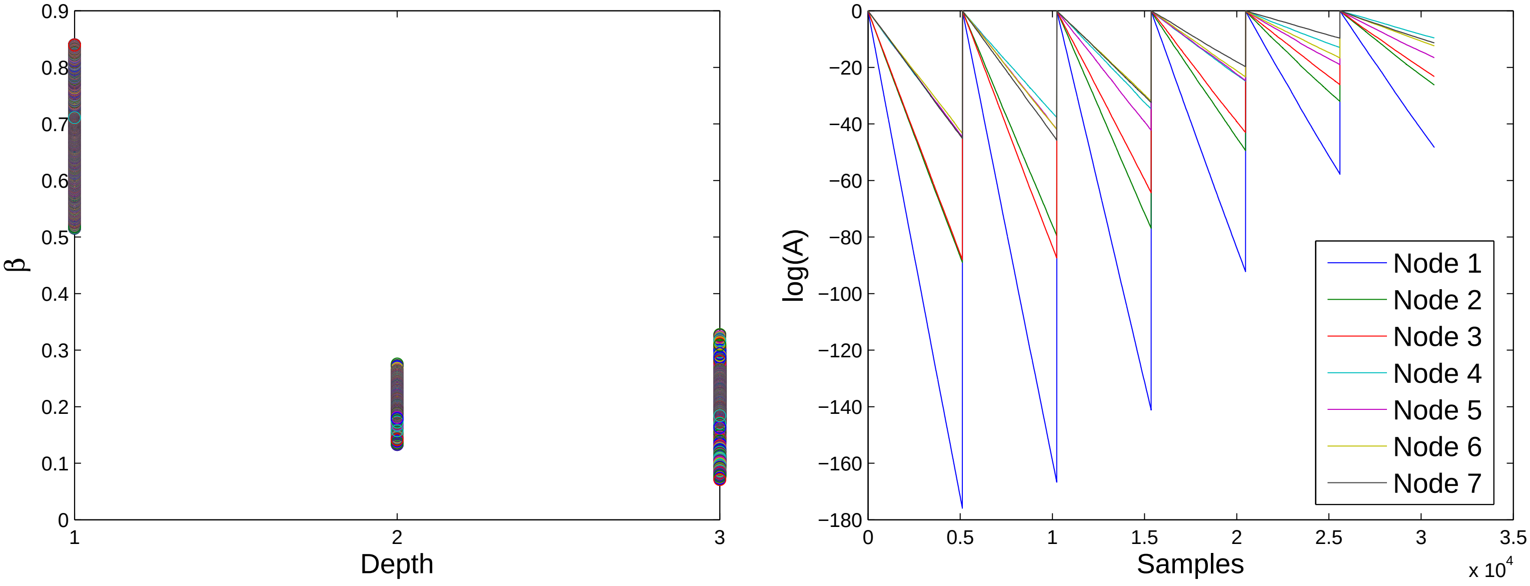
Figure 3.4: (a)Ensemble averaged weight vector for the DD LMS algorithm in the first turbo iteration, where $= 0.001, T = 1024$ and data length 5120. (b) Convolution of the trained weight vector of the DD LMS algorithm at sample 5120 and the channel h.



(a)

(b)

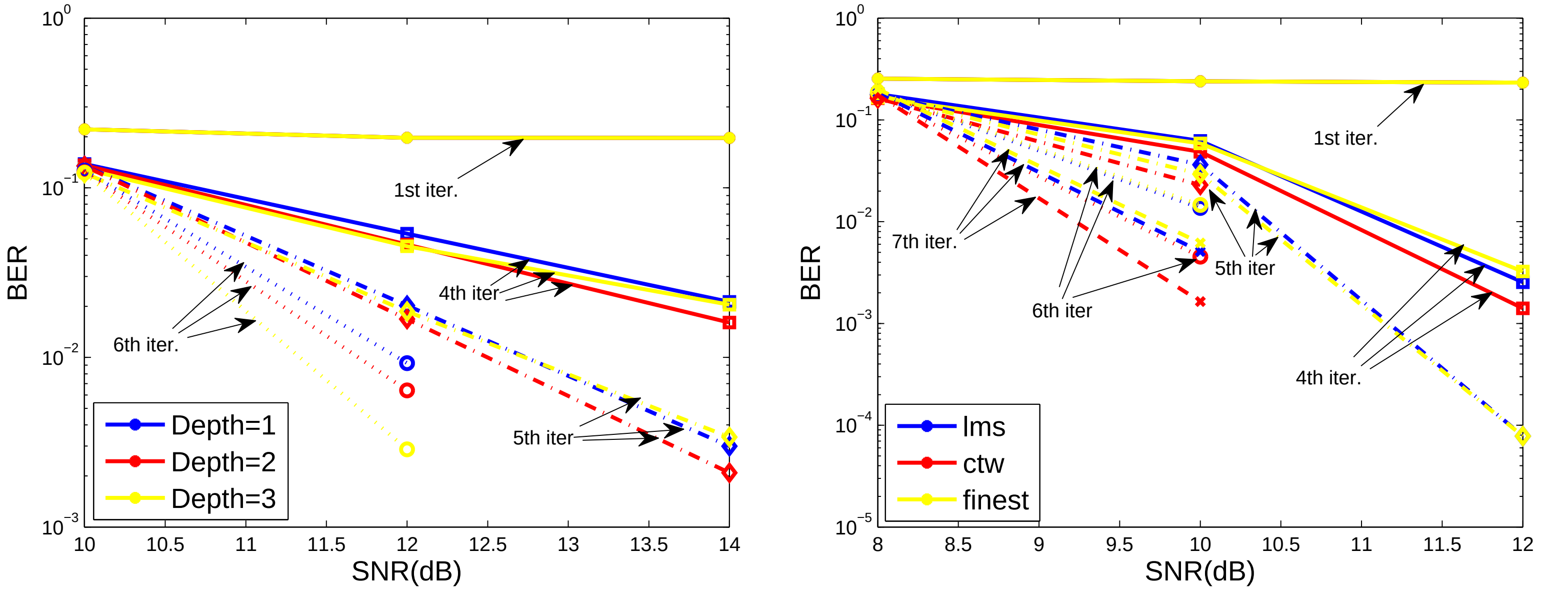
Figure 3.5: (a)BERs for an ordinary DD LMS algorithm, a CTW equalizer with $D = 2$ and tree given in Fig. 3.2, the piecewise equalizer with the finest partition, i.e., $\hat{x}(t)_{\Gamma_5}$, where $\mu = 0.001$, $N_1 = 9$, $N_2 = 5$, $N + M - 1 = 19$. (b)Ensemble averaged combined weight vector for the CTW equalizer over 7 turbo iterations. Here, we have $\mu = 0.001$, $T = 1024$, data length 5120 and 7 turbo iterations. Note that the combined weight vector for the CTW algorithm is only defined over the data length period 5120 at each turbo iteration.



(a)

(b)

Figure 3.6: (a) The distribution of the weights, i.e., values assigned to $\beta_i(t)$, $i = 1, 2, 3$, such that $\beta_i(t)$ belongs to ith level. (b)Time evaluation of $A_\rho(t)$ which represents the performance of the linear equalizer assigned to node ρ.



(a) (b)

Figure 3.7: (a)BERs corresponding to CTW equalizers of depth $D = 1$, $D = 2$ and $D = 3$. (b)BERs for an ordinary DD LMS algorithm, a CTW equalizer with $D = 2$ and tree given in Fig. 3.2, the piecewise equalizer with the finest partition, i.e., $\hat{x}(t)_{\Gamma_5}$, where $\mu = 0.001$, $N_1 = 9$, $N_2 = 5$, $N + M - 1 = 19$.

Chapter 4

CONCLUSIONS

In this thesis, we consider two types of turbo equalization methods over the frequency selective channels, robust linear turbo equalization methods and nonlinear turbo equalization methods. Chapter 2 deals with robust linear turbo equalization methods over frequency selective channels under channel uncertainties and Chapter 3 is dedicated to nonlinear turbo equalization methods when the channel information is not known.

In Chapter 2, we have investigated robust linear equalization methods when there are uncertainties in channel parameters [12]. Robust turbo equalization methods are introduced, minimax approach and the competitive approach. In both approaches, we have shown that obtaining linear equalizer coefficients for both problems can formulated as SDP problems. Furthermore, implementations of the proposed methods with reduced computational complexity are introduced. With simulation results, we have seen that there are significant improvements in the performance of the proposed methods over the plug-in MMSE estimators.

Although in Chapter 2 we have focused on the linear turbo equalizers, they have still have higher computational complexity than adaptive linear turbo equalizers that use adaptive learning algorithms such as the RLS or LMS to train their coefficients and require the channel information or the estimate of the channel. For a linear MMSE turbo equalizer computational complexity is $O(N^2 + M^2)$. However, a turbo equalizer using LMS algorithm only needs $O(N + M)$ computational complexity. In Chapter 3, we introduced a novel nonlinear turbo equalization method when the underlying communication channel is not known at the receiver [11]. In particular, adaptive nonlinear turbo equalization is investigated in order to model the nonlinear dependency of the linear MMSE equalizer on the soft information from the decoder. This is accomplished by the introduction of the piecewise linear models based on context trees. The piecewise linear models introduced can adaptively choose both the piecewise regions as well as the linear equalizer coefficients in each region independently, with computational complexity only in the order of a regular adaptive linear equalizer. Through simulations it is demonstrated that this approach is guaranteed to asymptotically achieve the performance of the best piecewise linear equalizer that can choose both its piecewise regions (from a class of doubly exponential number of partitions) as well as its filter parameters based on observing the whole data in advance. Also, the MSE performance of the resulting algorithm is shown to converge to the MSE of the linear minimum MSE estimator as the depth of the context tree and the data length increase.

Chapter 5

APPENDIX A

Lemma 1: The first order linear approximation of $\mathbf{v}^T[{\mathbf{Q}'}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F}]^{-1}\mathbf{v}$ around the channel estimate $\tilde{\mathbf{f}}_t$ is given by

$$\mathbf{v}^T[{\mathbf{Q}'}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F}]^{-1}\mathbf{v} = \eta_t + \mathbf{df}^H{\mathbf{g}_t}^+ + {\mathbf{g}_t}^T\mathbf{df} + O(\|\mathbf{df}\|^2).$$

Proof: The derivation of Lemma 1 is similar to the derivation of Lemma 1 of [14]. The first order linear approximation is given by

$$\mathbf{v}^T[{\mathbf{Q}'}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F}]^{-1}\mathbf{v} \approx \underbrace{\mathbf{v}^T[{\mathbf{Q}'}_t^{-1} + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t]^{-1}\mathbf{v}}_{\eta_t} + 2\Re e\{\mathrm{tr}[\mathbf{T}_t^H\mathbf{dF}]\}, \tag{5.1}$$

where $\mathbf{T}_t \overset{\triangle}{=} \nabla_{\mathbf{F}}\left(\mathbf{v}^T[{\mathbf{Q}'}_t^{-1} + \sigma_n^{-2}\mathbf{F}^H\mathbf{F}]^{-1}\mathbf{v}\right)|_{\mathbf{F}=\widetilde{\mathbf{F}}_t}$ and $\mathbf{dF}$ is the convolution matrix generated from $\mathbf{df}$. The gradient term $\mathbf{T}_t$ is derived in Lemma 1 of [14] as

$$\mathbf{T}_t = -\sigma_n^{-2}\widetilde{\mathbf{F}}_t({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\mathbf{v}\mathbf{v}^T({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}. \tag{5.2}$$

Using (5.2), we have

$$\begin{aligned}
\mathrm{tr}[\mathbf{T}_t^H\mathbf{dF}] &= \mathrm{tr}[-\sigma_n^{-2}({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\mathbf{v}\mathbf{v}^T({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\widetilde{\mathbf{F}}_t^H\ \mathbf{dF}] \\
&= \mathrm{tr}[-\sigma_n^{-2}\mathbf{v}^T({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\widetilde{\mathbf{F}}_t^H\ \mathbf{dF}\ ({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\mathbf{v}] \\
&= \mathrm{tr}[-\tilde{\mathbf{c}}_t^T\ \mathbf{dF}\ ({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\mathbf{v}] \\
&= [\mathbf{df}^T\underbrace{(-\widetilde{\mathbf{C}}_t({\mathbf{Q}'}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t)^{-1}\mathbf{v})}_{\mathbf{g}_t}]
\end{aligned} \tag{5.3}$$

where the second line is due to the properties of the trace operation, the third line follows from (2.8), the fourth line follows since $\mathbf{dF}$ is a convolution matrix and $\widetilde{\mathbf{C}}_t$ is the convolution matrix constructed using $\tilde{\mathbf{c}}_t$. Then, using (5.3) in (5.1) we get

$$\mathbf{v}^T[\mathbf{Q}'^{-1}_t + \sigma_n^{-2}\mathbf{F}^H\mathbf{F}]^{-1}\mathbf{v} \approx \underbrace{\mathbf{v}^T[\mathbf{Q}'^{-1}_t + \sigma_n^{-2}\widetilde{\mathbf{F}}_t^H\widetilde{\mathbf{F}}_t]^{-1}\mathbf{v}}_{\eta_t} + 2\Re e[\mathbf{df}^T\mathbf{g}_t].$$

This completes the proof of Lemma 1. □

Lemma 2: The inequality

$$\begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^H & \mathbf{R} \end{bmatrix} \geq 0, \tag{5.4}$$

where $\mathbf{Q} = \mathbf{Q}^H$, $\mathbf{R} = \mathbf{R}^H$ and $\mathbf{R} > 0$ is equivalent to

$$\mathbf{R} > 0, \quad \mathbf{Q} - \mathbf{S}\mathbf{R}^{-1}\mathbf{S}^H \geq 0. \tag{5.5}$$

Proof of Lemma 2: Assume $\mathbf{R} > 0$ and $\mathbf{M}_1 = \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^H & \mathbf{R} \end{bmatrix} \geq 0$. Then, using the nonsingular matrix $\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{R}^{-1}\mathbf{S}^H & \mathbf{I} \end{bmatrix}$, one can establish the congruence transformation

$$\mathbf{M}_2 = \mathbf{T}^H\mathbf{M}_1\mathbf{T} = \begin{bmatrix} \mathbf{Q} - \mathbf{S}\mathbf{R}^{-1}\mathbf{S}^H & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}. \tag{5.6}$$

Assuming $\mathbf{M}_1 \geq 0$ yields $\mathbf{M}_1$ and $\mathbf{M}_2$ to have the same inertia. Since by assumption $\mathbf{R} > 0$, we conclude that $\mathbf{Q} - \mathbf{S}\mathbf{R}^{-1}\mathbf{S}^H \geq 0$. □

Lemma 3: Given matrices $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{A}$ with $\mathbf{A} = \mathbf{A}^H$,

$$\mathbf{A} \geq \mathbf{P}^H\mathbf{Z}\mathbf{Q} + \mathbf{Q}^H\mathbf{Z}^H\mathbf{P}, \;\; \forall \|\mathbf{Z}\| \leq \alpha,$$

if and only if there exists a $\lambda \geq 0$ such that

$$\begin{bmatrix} \mathbf{A} - \lambda\mathbf{Q}^H\mathbf{Q} & -\alpha\mathbf{P}^H \\ -\alpha\mathbf{P} & \lambda\mathbf{I} \end{bmatrix} \geq 0.$$

This lemma is from Proposition 2 of [5]. □

Proofs of Theorem 1 and Theorem 2

Proof of Theorem 1: We first observe that the MSE expression in (2.13) can be written as

$$\min_{\mathbf{c}} \max_{\mathbf{f}=\tilde{\mathbf{f}}_t+\mathbf{df},\|\mathbf{df}\|\leq\delta} \left[(\mathbf{v}-\mathbf{C}^T\mathbf{f})^H{\mathbf{Q}'}_t(\mathbf{v}-\mathbf{C}^T\mathbf{f})+\sigma_n^2\mathbf{c}^H\mathbf{c}\right] = \min_{\mathbf{df},\alpha} \alpha$$

such that

$$(\mathbf{v}-\mathbf{C}^T\mathbf{f})^H{\mathbf{Q}'}_t(\mathbf{v}-\mathbf{C}^T\mathbf{f})+\sigma_n^2\mathbf{c}^H\mathbf{c} \leq \alpha \tag{5.7}$$

and $\|\mathbf{df}\| \leq \delta$. Using Lemma 2 in (5.7) yields

$$\begin{bmatrix} \alpha-(\mathbf{v}-\mathbf{C}^T\mathbf{f})^H{\mathbf{Q}'}_t(\mathbf{v}-\mathbf{C}^T\mathbf{f}) & \mathbf{c}^H \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} \end{bmatrix} \geq 0. \tag{5.8}$$

Applying Lemma 2 to (5.8) yields

$$\begin{bmatrix} \alpha & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\mathbf{f})^H \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\mathbf{f}) & \mathbf{0} & {\mathbf{Q}'}_t^{-1} \end{bmatrix} \geq 0. \tag{5.9}$$

However, (5.9) can be written as

$$\begin{bmatrix} \alpha & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t)^H \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t) & \mathbf{0} & {\mathbf{Q}'}_t^{-1} \end{bmatrix} \geq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{C}^T \end{bmatrix} \mathbf{df} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{df}^H \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{C}^+ \end{bmatrix}. \tag{5.10}$$

Applying Lemma 3 to (5.10) yields

$$\begin{bmatrix} \alpha-\tau & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t)^H & \mathbf{0} \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t) & \mathbf{0} & {\mathbf{Q}'}_t^{-1} & -\delta\mathbf{C}^T \\ \mathbf{0} & \mathbf{0} & -\delta\mathbf{C}^+ & \tau\mathbf{I} \end{bmatrix} \geq 0, \tag{5.11}$$

with the constraint $\|\mathbf{df}\| \leq \delta$. Hence, using (5.11) in (5.7) results Theorem 1. This completes the proof of Theorem 1. □

Proof of Theorem 2: The proof of Theorem 2 closely follows the proof of Theorem 1. The

MSE expression in (2.22) can be written as

$$\min_{\mathbf{c}} \max_{\mathbf{f}=\tilde{\mathbf{f}}_t+\mathbf{df},\|\mathbf{df}\|\leq\delta} \left[(\mathbf{v}-\mathbf{C}^T\mathbf{f})^H\mathbf{Q}'_t(\mathbf{v}-\mathbf{C}^T\mathbf{f})+\sigma_n^2\mathbf{c}^H\mathbf{c}-(\eta_t+\mathbf{df}^H{\mathbf{g}_t}^{+}+{\mathbf{g}_t}^T\mathbf{df})\right]=\min_{\mathbf{df},\alpha}\alpha$$

such that

$$(\mathbf{v}-\mathbf{C}^T\mathbf{f})^H\mathbf{Q}'_t(\mathbf{v}-\mathbf{C}^T\mathbf{f})+\sigma_n^2\mathbf{c}^H\mathbf{c}-(\eta_t+\mathbf{df}^H{\mathbf{g}_t}^{+}+{\mathbf{g}_t}^T\mathbf{df})\leq\alpha \tag{5.12}$$

and $\|\mathbf{df}\|\leq\delta$. Applying Lemma 2 to (5.12) successively two times yields

$$\begin{bmatrix} \alpha-(\eta_t+\mathbf{df}^H{\mathbf{g}_t}^{+}+{\mathbf{g}_t}^T\mathbf{df}) & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\mathbf{f})^H \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\mathbf{f}) & \mathbf{0} & \mathbf{Q}'^{-1}_t \end{bmatrix}\geq 0,$$

which can be written as

$$\begin{bmatrix} \alpha-\eta_t & \mathbf{c}^H & (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t)^H \\ \mathbf{c} & \sigma_n^{-2}\mathbf{I} & \mathbf{0} \\ (\mathbf{v}-\mathbf{C}^T\tilde{\mathbf{f}}_t) & \mathbf{0} & \mathbf{Q}'^{-1}_t \end{bmatrix}\geq\begin{bmatrix} {\mathbf{g}_t}^T \\ \mathbf{0} \\ \mathbf{C}^T \end{bmatrix}\mathbf{df}\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}+\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\mathbf{df}^H\begin{bmatrix} {\mathbf{g}_t}^{+} & \mathbf{0} & \mathbf{C}^{+} \end{bmatrix}, \tag{5.13}$$

with $\|\mathbf{df}\|\leq\delta$. Using Lemma 3 in (5.13) yields the constraint (2.24) given in Theorem 2. This completes the proof of Theorem 2. □

Chapter 6

APPENDIX B

Proof of equation 3.13

$$
\begin{aligned}
&||\mathbf{w}_{k,0}^{T}\mathbf{H}_r\mathbf{Q}(t)\mathbf{H}_r^{H}\mathbf{w}_{k,0}^{*}+\sigma_n^2\mathbf{w}_{k,0}^{T}\mathbf{w}_{k,0}^{*}-[1-\mathbf{v}^{H}(\sigma_n^2\mathbf{I}+\mathbf{H}_r\mathbf{Q}(t)\mathbf{H}_r^{H}+\mathbf{v}\mathbf{v}^{H})^{-1}\mathbf{v}]|| \\
&\quad=\mathbf{w}_{k,0}^{T}\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H}\mathbf{w}_{k,0}^{*}+[(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})^{-1}-M^{-1}]\mathbf{v},
\end{aligned}
\tag{6.1}
$$

where $\mathbf{M}\overset{\triangle}{=}\sigma_n^2\mathbf{I}+\mathbf{H}_r\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H}+\mathbf{v}\mathbf{v}^{H}$ and $\Delta\tilde{\mathbf{Q}}_k=\mathbf{Q}(t)-\tilde{\mathbf{Q}}_k$. Note that

$$
\begin{aligned}
\mathbf{v}^{H}(\mathbf{M}+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})^{-1}\mathbf{v}&=\mathbf{v}^{H}\mathbf{M}^{-1}\mathbf{v}+\mathrm{tr}\{\nabla^{H}_{\Delta\tilde{\mathbf{Q}}_k}[\mathbf{v}^{H}(\mathbf{M}+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})^{-1}\mathbf{v}]|_{\Delta\tilde{\mathbf{Q}}_k=0}\Delta\tilde{\mathbf{Q}}_k\} \\
&\quad+O(\mathrm{tr}[\Delta\tilde{\mathbf{Q}}_k^{H}\Delta\tilde{\mathbf{Q}}_k]) \\
&=\mathbf{v}^{H}\mathbf{M}^{-1}\mathbf{v}+\mathrm{tr}\{\mathbf{M}^{-1}\mathbf{v}\mathbf{v}^{H}\mathbf{M}^{-1}\Delta\tilde{\mathbf{Q}}_k\}+O(\mathrm{tr}[\Delta\tilde{\mathbf{Q}}_k^{H}\Delta\tilde{\mathbf{Q}}_k])
\end{aligned}
\tag{6.2}
$$

around $\Delta\tilde{\mathbf{Q}}_k=0$. Substituting the above equation into (6.1) yields

$$
\begin{aligned}
&\mathbf{w}_{k,0}^{T}\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H}\mathbf{w}_{k,0}^{*}+[(\mathbf{M}+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})^{-1}-M^{-1}]\mathbf{v} \\
&\quad=\mathbf{w}_{k,0}^{T}\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H}\mathbf{w}_{k,0}^{*}+\mathrm{tr}\{\mathbf{M}^{-1}\mathbf{v}\mathbf{v}^{H}\mathbf{M}^{-1}\Delta\tilde{\mathbf{Q}}_k\}+O(||\mathbf{q}(t)-\tilde{\mathbf{q}}_k||^2)\le O(||\mathbf{q}(t)-\tilde{\mathbf{q}}_k||)
\end{aligned}
\tag{6.3}
$$

by Schwarz inequality.

Lemma:

$$
\nabla_{\Delta\tilde{\mathbf{Q}}_k}\mathbf{v}^{H}(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})^{-1}\mathbf{v}=(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})^{-1}\mathbf{v}\mathbf{v}^{-1}(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})^{-1}. \tag{6.4}
$$

Proof of Lemma: Differentiating the identity $(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})^{-1}(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^{H})=\mathbf{I}$

with respect to $(\Delta\tilde{\mathbf{Q}}_k)_{a,b}$ yields

$$\frac{\partial(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^H)^{-1}}{\partial(\Delta\tilde{\mathbf{Q}}_k)_{a,b}}(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^H)+(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^H)^{-1}(\mathbf{H}_r e_a e_b^T\mathbf{H}_r^T)=0, \quad (6.5)$$

where e_a is a vector of all zeros except ath entry with 1. This yields

$$\begin{aligned}\frac{\partial\mathbf{v}^H(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^H)^{-1}\mathbf{v}}{\partial(\Delta\tilde{\mathbf{Q}}_k)_{a,b}} &= \mathbf{v}^H(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^H)^{-1}\mathbf{H}_r e_a e_b^T\mathbf{H}_r^H(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^H)^{-1}\mathbf{v}\\ &= e_b\mathbf{H}_r^H\mathbf{v}^H(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^H)^{-1}\mathbf{v}\mathbf{v}^H(M+\mathbf{H}_r\Delta\tilde{\mathbf{Q}}_k\mathbf{H}_r^H)^{-1}\mathbf{H}_r e_a\end{aligned} \quad (6.6)$$

which yields the result.

Outline of the proof of the Theorem 1 in Section 3.3: The proof of the theorem follows the proof of the Theorem 2 of [17] and Theorem 1 of [27]. Hence, we mainly focus on differences.

Suppose we construct $\hat{x}_{\Gamma_i}(t), i=1,\ldots,m$ and compute weights

$$c_i(t) \triangleq \frac{2^{-C(\Gamma_i)}exp\{-a\sum_{r=1}^{t-1}\}[x(r)-\hat{x}_{\Gamma_i}(r)]^2}{\sum_{j=1}^{m}2^{-C(\Gamma_j)}exp\{-a\sum_{r=1}^{t-1}\}[x(r)-\hat{x}_{\Gamma_j}(r)]^2}, \quad (6.7)$$

where $0 < C(\Gamma_j) \leq 2K_j - 1$ are certain constants that are used only for proof purposes such that $\sum_{j=1}^{m} 2^{-C(\Gamma_j)} = 1$ [31] and a is a positive constant set to $a = 1/2$ [27]. If we define

$$\hat{x}(t) = \sum_{k=1}^{m} = c_k(t)\hat{x}_{\Gamma_k}(t) \quad (6.8)$$

then it follows from Theorem 1 of [27] that

$$\sum_{t=1}^{n}[x(t)-\hat{x}(t)]^2 \leq \sum_{t=1}^{n}[x(t)-\hat{x}_{\Gamma_i}(t)]^2 + O(K_i)$$

for all $i = 1,\ldots,m$. Hence, $\hat{x}(t)$ is the desired $\hat{x}_{ctw}(t)$ However, note that $\hat{x}(t)$ requires output of m algorithms and computes m performance based weights in (6.7). However, in $\hat{x}(t)$ there are only D distinct node predictions $\hat{x}_\rho(t)$ that $\mathbf{q}(t)$ belongs to such that all the weights with the same node predictions can be merged to construct the performance weighting. It is shown in [17] that if one defines certain functions of performance for each node as $A_\rho(t)$,

$B_{\rho}(t)$ that are initialized in (line A) and updated in (line C), (line D), (line E) of Table 3.2, then the corresponding $\hat{x}(t)$ can be written as $\hat{x}(t) = \hat{x}(t) = \sum_{l=1}^{D+1} = \beta_l(t)\hat{x}_{\boldsymbol{l}(l)}(t)$, where $\boldsymbol{l}$ contains the nodes that $\mathbf{q}(t)$ belongs to and $\beta_l(t)$ are calculated as shown in (line B) of Table 3.2. Hence, the desired equalizer is given by

$$\hat{x}_{ctw}(t) = \hat{x}(t) = \sum_{l=1}^{D+1} = \beta_l(t)\hat{x}_{\boldsymbol{l}(l)}(t), \tag{6.9}$$

which requires to combine $D+1$ node estimations and update only D+1 node equalizers at each time t and store $2^{D+1}-1$ node weights. This completes the outline of the proof. To get the corresponding result in (3.16), we define the node predictors as the LS predictors such that

$$[\mathbf{w}_{\rho}(t)\ \ \mathbf{f}_{\rho}(t)]^T = \mathbf{M}^{-1}(t-1)\mathbf{p}(t-1), \tag{6.10}$$

$$\mathbf{M}(t-1) \overset{\triangle}{=} (\sum_{r=1}^{t} \mathbf{d}(r-1)\mathbf{d}(r-1)^T s_{\rho}(r) + \delta\mathbf{I}) \tag{6.11}$$

where $\mathbf{p}(t-1) \overset{\triangle}{=} \sum_{r=1}^{t-1} Q(\hat{y}(r))\mathbf{d}(r-1)s_{\rho}(r)$, and $\mathbf{d}(r) \overset{\triangle}{=} [\mathbf{y}(r)\ \bar{x}(r)]^T$, $s_{\rho}(r)$ is the indicator variable for node ρ, i.e., $s_{\rho}(r) = 1$ if $\mathbf{q}(r) \in V_{\rho}$ otherwise $s_{\rho}(r) = 0$. The affine predictor in (6.11) is a least squares predictor that trains only on the observed data $\{y(t)\}$ and $\{\bar{x}(t)\}$ that belongs to that node. The RLS algorithm is shown to achieve the excess loss given in (3.16) as shown in [20].

BIBLIOGRAPHY

[1] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes. In *Proc. Int. Conf. on Communications, Geneva, Switzerland*, 1993.

[2] R. Bidan. Turbo-equalization for bandwith-efficient digital communications over frequency-selective channels. *Ph.D. Thesis, Institut TELECOM/TELECOM Bretagne*, 2003.

[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* Studies in Applied Mathematics, 1994.

[4] J. W. Choi, R. Drost, A. Singer, and J. Preisig. Iterative multichannel equalization and decoding for high frequency underwater acoustic communications. *IEEE Sensor Array and Multichannel Sig. Proc. Workshop*, page 127130, 2008.

[5] Y.C. Eldar and N. Merhav. A competitive minimax approach to robust estimation and random parameters. *IEEE Transactions on Signal Process.*, 52:1931–1946, 2004.

[6] C. Douillard et al. Iterative correction of intersymbol interference: Turbo equalization. *Eur. Trans. Telecommun.*, 6:507–511, 1995.

[7] K. Ganti. Interleaver design for modified circular simplex turbo block coded modulator. *Master Thesis,Russ College of Engineering and Technology of Ohio University, Athens, OH*, Nov 2004.

[8] A. Glavieux, C. Laot, and J. Labat. Turbo equalization over a frequency selective channel. In *Proc. Int. Symp. Turbo Codes, Brest, France*, pages 96–102, 1997.

[9] D. P. Helmbold and R. E. Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27:5168, 1997.

[10] Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation.* Prentice-Hall, 2000.

[11] N. Kalantarova, Kyeongyeon Kim, S.S. Kozat, and A.C. Singer. Nonlinear turbo equalization using context trees. In *Information Theory and Applications Workshop (ITA), 2011*, pages 1 –5, feb. 2011.

[12] N. Kalantarova, S.S. Kozat, and A.T. Erdogan. Robust turbo equalization under channel uncertainties. *Signal Processing, IEEE Transactions on*, (Accepted in) July 2011.

[13] S. A. Kassam and H. V. Poor. Robust signal processing for communication systems. *IEEE Commun. Mag.*, 21:20–28, 1983.

[14] S. S. Kozat and A. T. Erdogan. Competitive linear estimation under model uncertainties. *IEEE Trans. on Signal Processing*, 58(4):2388–2393, April 2009.

[15] S. S. Kozat and A. C. Singer. Competitive prediction under additive noise. *IEEE Trans. on Signal Processing*, 57(9):3698–3703, 2009.

[16] S. S Kozat and A. C. Singer. Universal switching linear least squares prediction. *IEEE Transactions on Signal Processing*, 56:189–204, Jan. 2008.

[17] S. S Kozat, A. C. Singer, and G. Zeitler. Universal piecewise linear prediction via context trees. *IEEE Transactions on Signal Processing*, 55:3730–3745, 2007.

[18] K. Kyeongyeon, J. W. Choi, A. C. Singer, and K. Kim. A new adaptive turbo equalizer with soft information classification. *ICASSP*, page 32063209, 2010.

[19] C. Laot, R. Le Bidan, and D D. Leroux. Low-complexity mmse turbo equalization: a possible solution for edge. In *IEEE Transactions on Wireless Communications*, volume 4, pages 96–102, May 2005.

[20] N. Merhav and M. Feder. Universal schemes for sequential decision from individual sequences. *IEEE Transactions on Information Theory*, 39:12801292, 1993.

[21] O. J. J. Michel, A. O. Hero, and A. E. Badel. Vector quantization and signal compression. *The Springer International Series in Engineering and Computer Science*, 1992.

[22] O. J. J. Michel, A. O. Hero, and A. E. Badel. Tree structured non-linear signal modeling and prediction. *IEEE Transactions on Signal Processing*, page 30273041, 1999.

[23] G. Moustakides and S. A. Kassam. Minimax robust equalization for random signals through uncertain channels. In *Allerton Conf. Commun., Contr., Comput.*, pages 945–954, 1982.

[24] J. Proakis. *Digital Communications.* New York: McGraw-Hill, 2000.

[25] D. Raphaeli and A. Saguy. Linear equalizers for turbo equalization: a new optimizaiton criterion for determining the equalizer taps. In *Proc. Int. Symp. Turbo Codes, Brest, France*, pages 371–374, 2000.

[26] Ali H. Sayed. *Fundamentals of Adaptive Filtering.* John Wiley & Sons, 2003.

[27] A. C. Singer and M. Feder. Universal linear prediction by model order weighting. *IEEE Transs on Sig. Proc.*, 47(10), 1999.

[28] A.C. Singer, J.K. Nelson, and S.S. Kozat. Signal processing for underwater acoustic communications. *IEEE Communications Magazine*, 47:90–96, January 2009.

[29] S. Song, A.C. Singer, and K. M. Sung. Soft input channel estimation for turbo equalization. *IEEE Transactions on Signal Processing*, 52:2885 2894, October 2004.

[30] Michael Tuchler, Ralf Koetter, and Andrew C. Singer. Turbo equalization: Principles and new results. *IEEE Transactions on Communications*, 50(5):754–767, May 2002.

[31] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens. The context-tree weighting method: Basic properties. *IEEE Transactions Information Theory*, 41(3):653664, 1995.