COMBATING CHANNELS WITH LONG IMPULSE RESPONSE USING COMBINED TURBO EQUALIZATION AND TURBO DECODING

By

Chan Yiu Tong

A Thesis

submitted in partial fulfillment of the requirements

for the Degree of Master of Philosophy

Division of Information Engineering

The Chinese University of Hong Kong

June 2000

Acknowledgement

I am indebted to my thesis advisor, Professor Victor, K. W. Wei, whose insightful advice has become the foundation of this research. I have benefited a lot from him throughout the past two years. I would like to express my appreciation to his patience and open-mindedness.

Here, I would like to thank my friend, Julie Lam, for her encouragements throughout the past two years. I would also like to thank my colleagues in SHB Engineering Building of CUHK who provide me all the elements I need in my life. In addition, I would like to express gratitude to my schoolmates during my undergraduate studies and secondary school days who gave me a lot of good memories. And of course, many thanks to my family members who gave me the greatest support in all means.

Above all, I must thank researchers around the world who have been interrupted by my emails. All of their replies help me in solidifying my work in the last two years. Finally, my PC, Pentium II 450 MHz, and the faculty supercomputer, Sun Enterprise 6500 Server e6500, are indispensable to this work.

簡介

在大多數通訊系統中, 訊息不單只是受到四週環境的雜訊(noise)干擾, 訊息還會被自己或其他訊息所干擾. 這現象是因爲頻道的頻寬是有限的. 而這現象叫做 Intersymbol Interference (ISI). 若果沒有實施任何措施去底消 ISI, 訊息的準確性一定會很差. 因爲這個原因, 我們一定要取消 ISI 的影響. 而取消 ISI 的方法叫 equalization. 爲了增強訊息的準確性, 糾錯碼亦都會一同使用.

於一九九三年, C. Berrou 等發表了 turbo 碼, 其譯碼性能極之接近申農理論值, 在信噪比爲 0.7dB 時, 比特(bit)誤差概率可達 10^{-5}. 及後, turbo 碼的原理, 亦被稱爲 turbo principle, 被應用在其他方面上. 一九九五年 turbo principle 被應用在 equalization 上, 被稱之爲 turbo equalization. 之後 turbo 碼及 turbo equalization 被結合一起, turbo 解碼器能和 turbo equalizer 互相交換訊息, 來加強其解讀性能. 這結構被証明能夠加強接收器解讀性能及準確性.

但是, 在實際應用環境中, 訊道的 Impulse Response 長度通常都十分大. 用一個合併 turbo 碼和 turbo equalization 接收器來接收的話, 其複雜性卻會大於實際上能夠建立的. 而本論文的主旨, 就是研發出一個方法, 去減低這接收器的複雜性, 而能夠保持其良好的特性. 我們的理論是利用一個濾器, 令合併接收器能夠假設訊道的 Impulse Response 的長度比實際上的短. 結果我們發現, 我們的改進法比分開的 equalizer 和 turbo 碼接收器有大約 2.4 dB 改進, 於誤差概率 10^{-4}

Abstract

In many communication systems, received signals are not only being corrupted by surrounding noise but also the earlier transmitted signals which do not die away as time pasts. This is called Intersymbol Interference (ISI). Effects of ISI channel have to be compensated by equalization. And channel coding is usually added in order to provide error-free communications.

In 1993, Turbo Codes [10] was invented and was identified as a breakthrough in the coding community. The Turbo Principle [6][8][9][10][14] was further being applied into other areas including equalization [15]. However, in practical cases, the number of interfering signals is large. Turbo Equalization is too complex to be implemented without any modifications and would be unfavorable unless complexity problem is solved.

In this work, receiver structures built in combined turbo equalization and turbo decoding with modifications are proposed for combating channels with long impulse responses. We will evaluate their performance by computer simulations and simple performance anticipations based on theoretical limitations of the combinational blocks. Our derived receivers are shown to have 2.4 dB gain from the receivers with separate linear equalization followed by turbo decoding at BER 10^{-4}.

Contents

List of Figures

Chapter 1

Introduction

Information is one of the utmost important issues to any person in any times. In ancient wars, win or loss always depended on the reliability of information about enemies. Spy was ranked the most valuable in the book of Sun Tzu's the Art of War. Even in modern commerce society, knowledge about market trends and strategies of competitors are still important to any enterprise. Information is conveyed through communication. How communications process reliably and securely are always the top concern of businessmen and engineers.

In modern world, communication systems in many forms are widely being used. Users enjoy current provided services but at the same time, wish to have a more reliable system and a system with higher speed. Engineers and researchers work all day and night to accomplish those wishes.

1.1 Communications and Coding Technology

Certain communication systems like mobile phones, magnetic and optical storage devices, satellite communications and modems can be generalized in figure 1.1, where $x(t)$ denotes the transmitted signal, $y(t)$ is the received signal and $n(t)$ is the Additive White Gaussian Noise (AWGN). Different communication systems have different channel characteristics $h(t)$. If the transmitted information is not "properly processed", error rate of the received information will be high. As a result, channel coding is introduced in order to reduce the error rate. A message is encoded with a channel code before it is transmitted over the channel. Channel Coding usually adds in more bits to the data. The added bits will help combating against AWGN. At the receiving end, the received signal is decoded back to the message. But how good a system can be with channel coding added? It is an question worth thinking since it saves efforts spending on design issues.

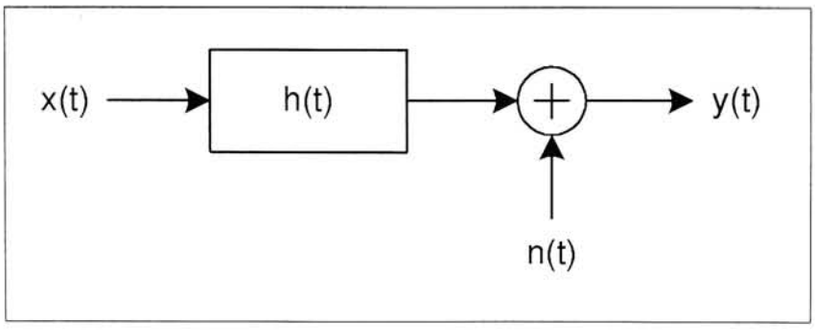

Figure 1.1: A Simplified Communication Model

In 1948, Shannon [1] stated that arbitrarily small error rates are achievable provided that the rate of transmission, as defined by the ratio of input bits to coded bits, is less than the capacity limit, which is approachable by random codes

as the block length increases. However, as block length increases, complexity of the optimal decoder becomes unmanageable. Therefore, it has been a constant quest of coding theorists to design codes which perform near the capacity limit while offering modest encoding and decoding complexity. To date, there are two classes of codes which offer the most promise, block codes and trellis codes.

1.2 The Emerge of Turbo Codes

In 1993, Berrou, Glavieux and Thitimajshima [10] invented a new coding scheme, named Turbo Codes, which uses ideas related to both block codes and trellis codes. Turbo Codes use simple convolutional codes separated by interleaving stage(s) to produce generally low rate ($R < 1/2$) block codes. The decoding of Turbo Codes is done by decoding the convolutional codes separately and sharing bit reliability information in an iterative procedure. The result is a block coding technique which mimics the performance of random codes but whose decoding complexity is only loosely tied to the block size. It has been shown that for sufficiently large block length, performance of Turbo Codes is just 0.5 dB away from the capacity limit on AWGN channel at BER 10^{-5} [10].

1.3 The Extension of Turbo Principle

For many cases, $h(t)$ of many communication systems cannot be modeled as a real-valued time invariant constant. For example, in many wireless and cable communication environments, $h(t)$ is often modeled as a complex-valued time variant random variable due to "multipath fading" of the transmitted signal [21].

Different techniques [16] called equalizations were developed which attempt to compensate the effect of $h(t)$. For particular systems like magnetic and optical disk devices, $h(t)$ is categorized as the channel with InterSymbol Interference (ISI). ISI arises in systems whenever the effects of one transmitted pulse does not die away completely before the transmission of the next. [2][16] show that the output of channels with ISI could be viewed as the output of finite state machines and could be estimated by Maximum Likelihood Sequence Estimation (MLSE) built in Viterbi Algorithm (VA).

Since estimation and decoding could be done in the same way. It is clear that ideas in [6][8][9][10][14], sometimes called "Turbo Principle", can be used in a wide variety of detection and decoding tasks. In 1995, Turbo Equalization [15] was first introduced which used the idea of "Turbo Principle" in the area of equalization with convolutional codes. Later on, the use of more powerful error correcting codes with this scheme [23][24][26] are investigated in order to archive better coding gains. It is believed that "Turbo Principle" could be applied to many different areas other than channel equalization and channel coding.

1.4 Receiver Structures for Practical Situations

In [2][16], MLSE is showed to be the optimal receiver structure for channels with ISI. However, in certain practical situations where the number of interfering component L is large, MLSE equalizer become too complex to be implemented because the complexity of the structure grows exponentially with L. Subsequently, suboptimal but practical receiver structure is proposed in [3]. As time goes by, newer techniques and algorithms are proposed for better receptions for

different cases. But they are still constrained by the limited computing powers and memory sizes. Nowadays, how can we communicate with advanced techniques like Combined Turbo Codes and Turbo Equalization in channels with long impulse responses? It is a question worth answering.

1.5 Thesis Overview

In this work, we propose receiver structures for ISI channels with long impulse responses which employ Combined Turbo Equalization and Turbo Coding techniques like [23][24][26][27][28] with certain modifications. They gives better performance yet comprise in manageable complexities compared with receivers which use traditional linear equalizers. Chapter 2 gives a brief introduction to the ISI channel model and equalization techniques. Chapter 3 gives an overview of Turbo Codes. Receiver structures for ISI channels with long impulse responses will be proposed in Chapter 4. Performance analysis of the proposed structures will be given in Chapter 5. Finally, we will conclude our work in Chapter 6.

Chapter 2

ISI Channel Model and Channel Equalization

This chapter will introduce why and how a channel with ISI can be modeled as a discrete time channel. It can be shown that MLSE is optimal for the estimation process. No information will be lost for subsequent estimation and decoding processes. This chapter will also list out two of the well documented equalizers, namely Minimum Mean Square Error (MMSE) Equalizer and MLSE Equalizer, which are related to this work seriously. Detailed discussions of the materials in this chapter can be found in [2][16].

2.1 A Discrete Time ISI Channel Model

Consider the structure of the optimum demodulator and detector for digital transmission through a nonideal, band-limited channel with additive gaussian noise. The equivalent lowpass transmitted signal for several different types of

digital modulation techniques has the common form

$$v(t) = \sum_{n=0}^{\infty} I_n g(t - nT) \tag{2.1}$$

where $\{I_n\}$ represents the discrete information-bearing sequence of symbols and $g(t)$ is a pulse that is assumed to have a band-limited frequency response characteristic $G(f)$, $G(f) = 0 for |f| > W$. The received signal is expressed as

$$r_l(t) = \sum_{n} I_n h(t - nT) + z(t) \tag{2.2}$$

where $h(t)$ represents the response of the channel to the input signal pulse $g(t)$ and $z(t)$ represents the AWGN.

This section reveals that the optimum demodulator can be constructed as a filter matched to $h(t)$, followed by a sampler operating at the symbol rate $1/T$ and a subsequent processing algorithm for estimating the information sequence $\{I_k\}$ from the sample values. Consequently, the samples at the output of the matched filter are sufficient for the estimation of the sequence I_k.

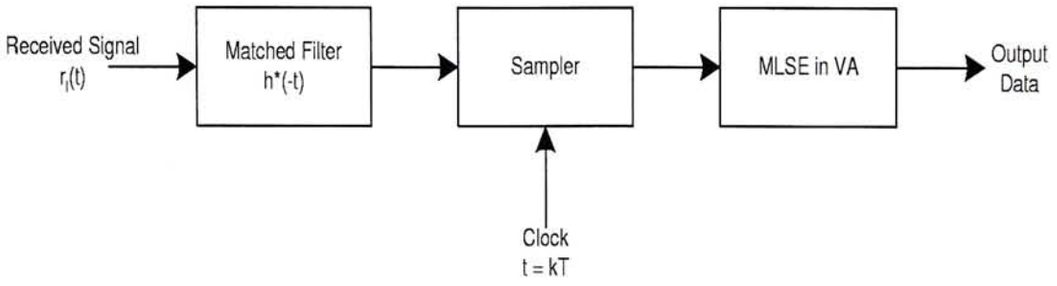

Figure 2.1: Optimum Receiver for AWGN Channel with ISI

2.1.1 Optimum Maximum Likelihood Receiver

Let us write down the received signal $r_l(k)$ in the series

$$r_l(k) = \lim_{N\to\infty} \sum_{k=1}^{N} r_k f_k(t) \tag{2.3}$$

where $\{f_k(t)\}$ is a complete set of orthonormal functions and $\{r_k\}$ are the observable random variables obtained by projecting $r_l(t)$ onto the set $\{f_k(t)\}$. It is shown that

$$r_k = \sum_n I_n h_{kn} + z_k, \qquad k = 1, 2, \ldots \tag{2.4}$$

where h_{kn} is the value obtained from projecting $h(t - nT)$ onto $f_k(t)$, and z_k is the value obtained from projecting $z(t)$ onto $f_k(t)$. The sequence z_k is gaussian with zero mean and covariance

$$\frac{1}{2}E(z_k^* z_m) = N_0 \delta_{km} \tag{2.5}$$

The joint probability density function of the random variables $\mathbf{r}_N = [r_1 \quad r_2 \quad \ldots \quad r_N]$ conditioned on the transmitted sequence $\mathbf{I}_P = [I_1 \quad I_2 \quad \ldots \quad I_P]$, where $p \leq N$, is

$$p(\mathbf{r}_N, \mathbf{I}_P) = \left(\frac{1}{2\pi N_0}\right)^N \exp\left(- \frac{1}{2N_0} \sum_{k=1}^{N} \left| r_k - \sum_n I_n h_{kn} \right|^2 \right) \tag{2.6}$$

In the limit as the number N of observable random variables approaches infinity, the logarithm of $p(\mathbf{r}_N, \mathbf{I}_P)$ is proportional to the metrics $PM(\mathbf{I}_P)$, defined as

$$\begin{aligned} PM(\mathbf{I}_P) &= -\int_{-\infty}^{\infty} \left| r_l(t) - \sum_n I_n h(t - nT) \right|^2 dt \\ &= -\int_{-\infty}^{\infty} \left| r_l(t) \right|^2 dt + 2\mathrm{Re} \sum_n \left[I_n^* \int_{-\infty}^{\infty} r_l(t) h^*(t - nT) dt \right] \\ &\quad - \sum_n \sum_m I_n^* I_m \int_{-\infty}^{\infty} h^*(t - nT) h(t - mT) dt \end{aligned} \tag{2.7}$$

The maximum likelihood estimates of the symbols $I_1, I_2, \ldots, I_p$ are those that maximize this quantity. However, the integral of $|r_l(t)|^2$ is common to all metrics

and it may be discarded. The other integral involving $r(t)$ gives rise to the variables

$$y_n \equiv y(nT) = \int_{-\infty}^{\infty} r_l(t)h^*(t - nT)dt \tag{2.8}$$

The samples $\{y_n\}$ form a set of sufficient statistics for the computation of $PM(\mathbf{I}_P)$ or, equivalently, of the correlation metrics

$$CM(\mathbf{I}_P) = 2\mathrm{Re}\Big(\sum_n I_n^* y_n\Big) - \sum_n \sum_m \mathbf{I}_n^* \mathbf{I}_m x_{n-m} \tag{2.9}$$

where $x(t)$ is the response of the matched filter to $h(t)$ and

$$x_n \equiv x(nT) = \int_{-\infty}^{\infty} h^*(t)h(t + nT)dt \tag{2.10}$$

Thus, $x(t)$ represents the output of a filter having an impulse response $h^*(-t)$ and an excitation $h(t)$. In other words, $x(t)$ represents the autocorrelation function of $h(t)$. Consequently, $\{x_n\}$ represents the samples of the autocorrelation function of $h(t)$, taken periodically at $1/T$.

If we substitute for $r_l(t)$ in (2.8) using (2.2), we obtain

$$y_k = \sum_n I_n x_{k-n} + v_k \tag{2.11}$$

where v_k denotes the additive noise sequence of the output of the matched filter

$$v_k = \int_{-\infty}^{\infty} z(t)h^*(t - kT)dt \tag{2.12}$$

The output of the demodulator (matched filter) at the sampling instants is corrupted by ISI as indicated by (2.11). In any practical system, it is reasonable to assume that $x_n = 0$ for $|n| > L$. Consequently, the ISI observed at the output of the demodulator may be viewed as the output of a finite state machine.

In short, since the transmitter sends discrete time symbols at a rate $1/T$ symbols/s and the sampled output of the matched filter at the receiver is also a

discrete time signal with samples occurring at a rate $1/T$ per second, it follows that the cascade of the analog filter at the transmitter with impulse response $g(t)$, the channel with impulse response $c(t)$, the matched filter at the receiver with impulse response $h^*(-t)$, and the sampler can be represented by an equivalent discrete time transversal filter having tap gain coefficients $\{x_k\}$. Consequently, we have an equivalent discrete time transversal filter that spans a time interval of $2LT$ seconds. Its input is the sequence of information symbols $\{I_k\}$ and its output is the discrete time sequence $\{y_k\}$ given by (2.11).

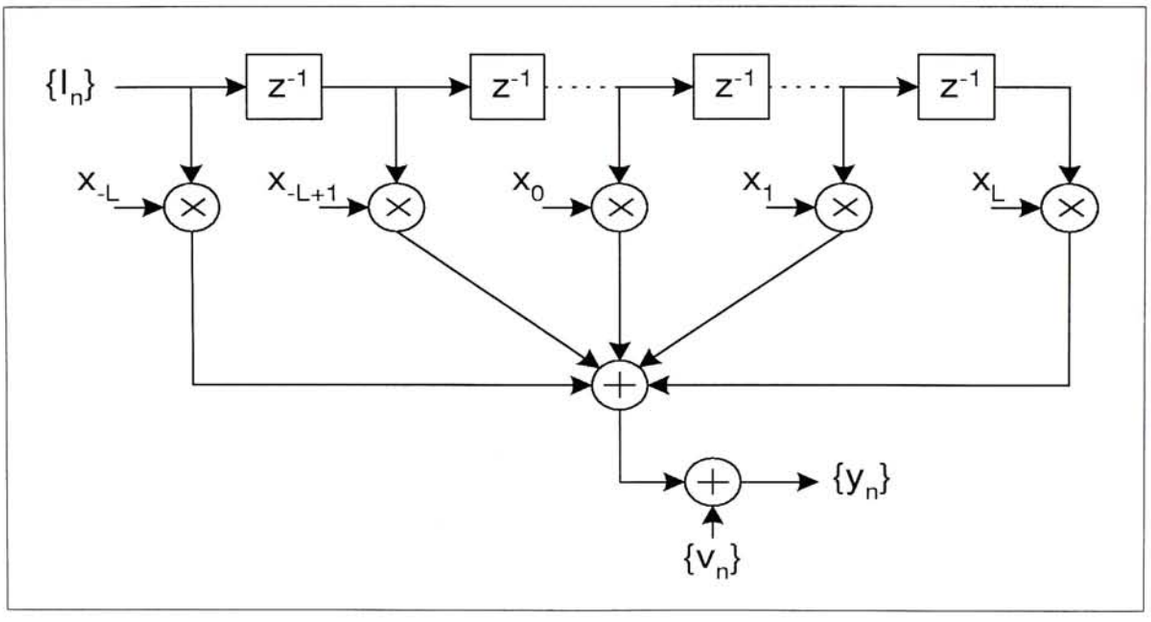

Figure 2.2: Equivalent Discrete Time Model with ISI

2.1.2 The Whitened Matched Filter

Unfortunately, the noise sequence $\{v_k\}$ at the output of the equivalent discrete time model mentioned above is correlated which will cause difficulties in evaluating the performance of the various equalization and estimation techniques. The set of noise variables $\{v_k\}$ is a gaussian distributed sequence with zero mean and autocorrelation function

$$\frac{1}{2}E(v_k^* v_j) = \begin{cases} N_0 x_{k-j} & (|k-j|) \leq L) \\ 0 & (otherwise) \end{cases} \tag{2.13}$$

where

$$x_k = \sum_{n=0}^{L-k} f_n^* f_{n+k}, \qquad k = 0, 1, 2, \ldots, L \tag{2.14}$$

Let $X(z)$ denote the (two side) z transform of the sampled autocorrelation function $\{x_k\}$

$$X(z) = \sum_{k=-L}^{L} x_k z^{-k} \tag{2.15}$$

Since $x_k = x_{-k}^*$, it follows that $X(z) = X^*(z^{-1})$ and the $2L$ roots of $X(z)$ have symmetry that if ρ is a root, $1/\rho^*$ is also a root. Hence, $X(z)$ can be factored and expressed as

$$X(z) = F(z)F^*(z^{-1}) \tag{2.16}$$

where $F(z)$ is a polynomial of degree L having the roots ρ_1, ρ_2, ..., ρ_L and $F^*(z^{-1})$ is a polynomial of degree L having the roots $1/\rho_1^*$, $1/\rho_2^*$, ..., $1/\rho_L^*(z^{-1})$. Then an appropriate noise whitening filter has a z transform $1/F^*(z^{-1})$. Consequently, passage of the sequence $\{y_k\}$ through the digital filter $1/F^*(z^{-1})$ results in an output sequence $\{v_k\}$ that can be expressed as

$$v_k = \sum_{n=0}^{L} f_n I_{k-n} + \eta_k \tag{2.17}$$

where $\{\eta_k\}$ is a white gaussian noise sequence and $\{f_k\}$ is a set of tap coefficients of an equivalent discrete time transversal filter having a transfer function $F(z)$.

In summary, the cascade of the transmitting filter $g(t)$, the channel $c(t)$, the matched filter $h^*(-t)$, the sampler, and the discrete time noise whitening filter $1/F^*(z^{-1})$ can be represented as an equivalent discrete time transversal filter having the set $\{f_k\}$ as its tap coefficients. The additive noise sequence $\{\eta_k\}$ corrupting the output of the discrete time transversal filter is a white guassian noise sequence having zero mean and variance N_0. We call such a cascade The Whitened Matched Filter.

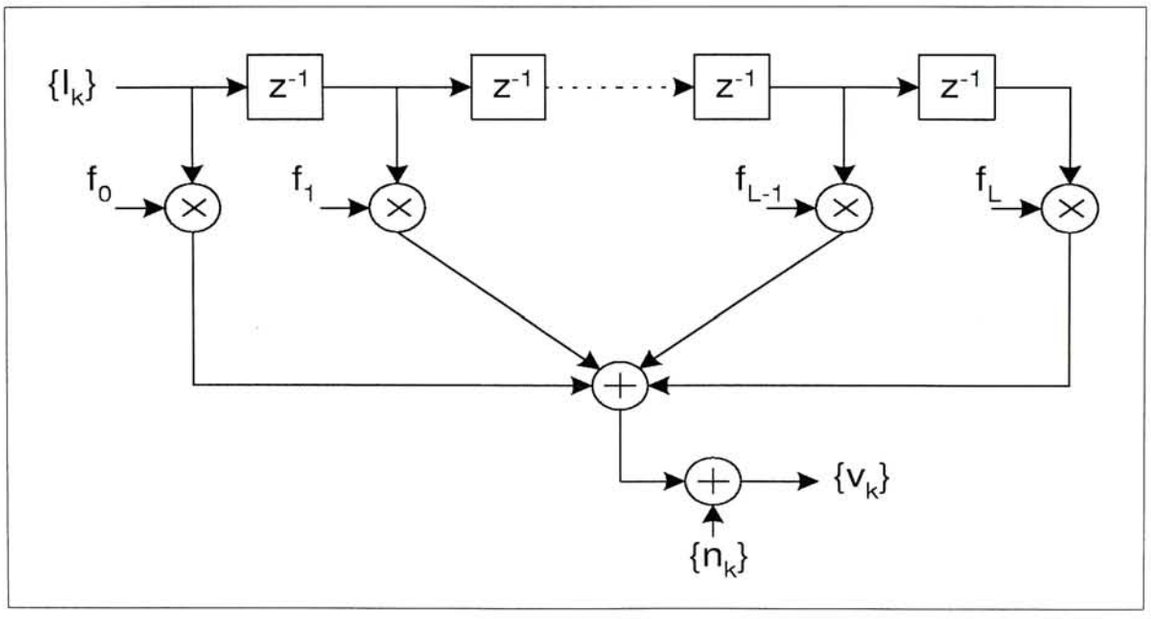

Figure 2.3: Equivalent Discrete Time Model of ISI Channel with AWGN

2.2 Equalization Techniques for Combating ISI

When communication takes place in the presence of ISI, high error rates would be resulted because of the channel distortion. The solution to the ISI problem is to employ a receiver which is capable of compensating or reducing the ISI in the received signal. The compensator for the ISI is called an equalizer.

Only MMSE equalizer and MLSE equalizer will be discussed here. MMSE equalizer is known for its simplicity in practical implementation. However, MMSE equalizers only works in channels which are well behaved and do not exhibit spectral nulls. They are inadequate as a compensator for ISI channels with spectral nulls. MLSE equalizers should be used in combating such bad channels but its complexity is high.

2.2.1 Linear MMSE Equalizer

MMSE equalizer is a linear transversal filter shown in figure 2.4, where $\{c_j\}$ is the tap weight coefficients of the equalizer, where I_k is the information symbol transmitted in the kth signaling interval and its estimated information sequence $\hat{I}_k$ is expressed as

$$\hat{I}_k = \sum_{j=-K}^{K} c_j v_{k-j} \tag{2.18}$$

MMSE equalizer is so called because its tap weight coefficients $\{c_j\}$ is adjusted to minimize the Mean Square Error (MSE) J

$$\begin{aligned} J &= E|\epsilon_k|^2 \\ &= E|I_k - \hat{I}_k|^2 \end{aligned} \tag{2.19}$$

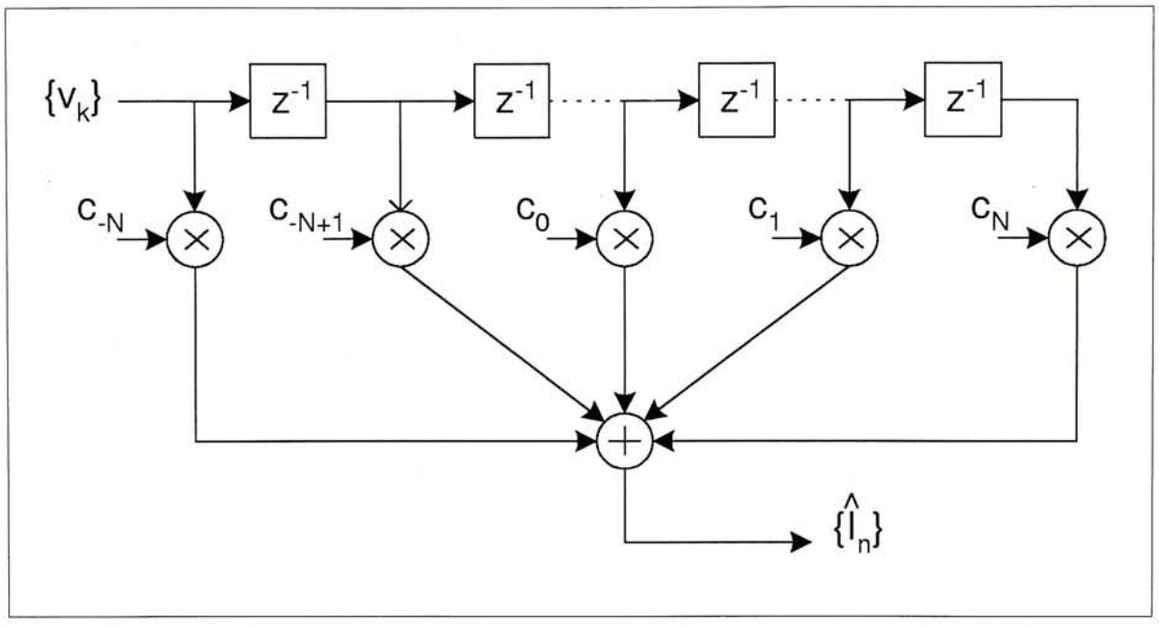

Figure 2.4: Linear Transversal Filter

The MSE for the equalizer having $2K+1$ taps, denoted by $J(K)$, is

$$J(K) = E|I_k - \hat{I}_k|^2 = E\Big|I_k - \sum_{j=-K}^{K} c_j v_{k-j}\Big|^2 \tag{2.20}$$

Minimization of $J(K)$ with respect to the tap weight $\{c_j\}$ or, equivalently, forcing the error $\epsilon_k = I_k - \hat{I}_k$ to be orthogonal to the signal samples v_{j-l}^*, $|l| \leq K$, yields the following set of simultaneous equations

$$\sum_{j=-K}^{K} c_j \Gamma_{lj} = \xi_l \qquad l = -K, \ldots, -1, 0, 1, \ldots, K \tag{2.21}$$

where

$$\Gamma_{lj} = \begin{cases} x_{l-j} + N_0 \delta_{lj} & (-L \leq l \leq L) \\ 0 & (otherwise) \end{cases} \tag{2.22}$$

and

$$\xi_l = \begin{cases} f_{-l}^* & (-L \leq l \leq 0) \\ 0 & (otherwise) \end{cases} \tag{2.23}$$

It is convenient to express the set of linear equations in matrix form. Thus,

$$\mathbf{\Gamma C} = \xi \tag{2.24}$$

where $\mathbf{C}$ denotes the column vector of $2K+1$ tap weight coefficients, $\mathbf{\Gamma}$ denotes the $(2K+1) \times (2K+1)$ Hermitian covariance matrix with elements Γ_{ij}, and ξ is a $(2K+1)$ dimensional column vector with elements ξ_l. the solution of (2.24) is

$$\mathbf{C}_{\text{opt}} = \mathbf{\Gamma}^{-1}\xi \tag{2.25}$$

The performance index $J(K)$ is minimized with the optimum tap weight coefficients $\mathbf{C}_{\text{opt}}$ given in (2.26).

$$\begin{aligned} J_{min}(K) &= 1 - \sum_{j=-K}^{0} c_j f_{-j} \\ &= 1 - \xi^{t*}\mathbf{\Gamma}^{-1}\xi \end{aligned} \tag{2.26}$$

where ξ^t represents the transpose of the column vector ξ.

2.2.2 MLSE Equalizer in Viterbi Algorithm

As shown in previous sections, MLSE equalizer is optimal in searching the most probable information sequence which is transmitted across the ISI channel. MLSE is equivalent to the problem of estimating the state of a discrete time finite state machine. The finite state machine in this case is the equivalent discrete time channel with coefficients $\{f_k\}$ and its state at any instant in time is given by the L most recent inputs. That means the state at time k is

$$S_k = (I_{k-1}, I_{k-1}, \ldots, I_{k-L}) \tag{2.27}$$

where $I_k = 0$ for $k \leq 0$. Hence, if the information symbols are M-ary, the channel filter has M^L states. Consequently, the channel is described by an M^L-state trellis and Viterbi Algorithm can be used to determined the most probable path through the trellis.

The metrics used in the trellis search are akin to the metrics used in soft-decision decoding of convolutional codes. In brief, we begin with the samples $v_1, v_2, \ldots, v_{L+1}$, from which we compute the M^{L+1} metrics

$$\sum_{k=1}^{L+1} \ln p(v_k | I_k, I_{k-1}, \ldots, I_{k-L}) \tag{2.28}$$

The M^{L+1} possible sequences of $I_{L+1}, I_L, \ldots, I_2, I_1$ are subdivided into M^L groups corresponding to the M^L states $(I_{L+1}, I_L, \ldots, I_2)$. Note that the M sequences in each group (state) differ in I_1 and correspond to the paths through the trellis that merge at a single node. From the M sequences in each of the M^L states, we select the sequence with the largest probability (with respect to I_1) and assign to the surviving sequence the metric

$$\begin{aligned} PM_1(\mathbf{I}_{L+1}) &= PM_1(I_{L+1}, I_L, \ldots, I_2) \\ &= \max_{I_1} \sum_{k=1}^{L+1} \ln p(v_k | I_k, I_{k-1}, \ldots, I_{k-L}) \end{aligned} \tag{2.29}$$

The $M - 1$ remaining sequences from each of the M^L groups are discarded. Thus, we are left with M^L surviving sequences and their metrics.

Upon reception of v_{L+2}, the M^L surviving sequences are extended by one stage, and the corresponding M^{L+1} probabilities for the extended sequences are computed using the previous metrics and the new increment, which is $\ln p(v_{L+2} | I_{L+2}, I_{L+1}, \ldots, I_2)$. Again, the M^{L+1} sequences are subdivided into M^L groups corresponding to the M^L possible states $(I_{L+2}, \ldots, I_3)$ and the most

probable sequence from each group is selected, while the other $M-1$ sequences are discarded.

The procedure described continues with the reception of subsequent signal samples. In general, upon reception of v_{L+k}, the metrics

$$PM_k(\mathbf{I}_{L+k}) = \max_{I_k}[\ln p(v_{L+k}|I_{L+k},\ldots,I_k)] + PM_{k-1}(\mathbf{I}_{L+k-1}) \tag{2.30}$$

that are computed give the probabilities of the M^L surviving sequences. Thus, as each signal sample is received, the Viterbi Algorithm involves first the computation of the M^{L+1} probabilities

$$\ln p(v_{L+k}|I_{L+k},\ldots,I_k) + PM_{k-1}(\mathbf{I}_{L+k-1}) \tag{2.31}$$

corresponding to the M^{L+1} sequences that form the continuations of the M^L surviving sequences from the previous stage of the process. Then the M^{L+1} sequences are subdivided into M^L groups, with each group containing M sequences that terminate in the same set of symbols $I_{L+k},\ldots,I_{k+1}$ and differ in the symbol I_k. From each group of M sequences, we select the one having the largest probability as indicated by (2.30), while the remaining $M-1$ sequences are discarded. Thus, we are left again with M^L sequences having the metrics $PM_k(\mathbf{I}_{L+k})$.

Chapter 3

An Overview of Turbo Codes

Turbo Codes are Parallel Concatenated Convolutional Codes (PCCC) which have demonstrated near-capacity performance shown in [10]. The encoder side of a PCCC system is comprised of two (or more) constituent Recursive Systematic Convolutional (RSC) encoders connected through an interleaver. On the other hand, its decoder side is built up with an iterative soft decoding algorithm. While the invention of Turbo Codes involved reviving some dormant concepts and algorithms with combining some clever new ideas, its discovery was still a breakthrough for its performance with limited decoding complexities.

Since 1993, researchers around the world have been investigating design parameters of Turbo Codes. As a result, much is now known regarding the design of interleavers, nature of the decoding algorithms and relationships between performance and block size, number of iterations and constituent encoder choice. Turbo Code research results are numerous and well documented. In this chapter, apart from giving a detailed analysis of Turbo Codes, only several important issues will be listed, which will be helpful for understanding our proposed models.

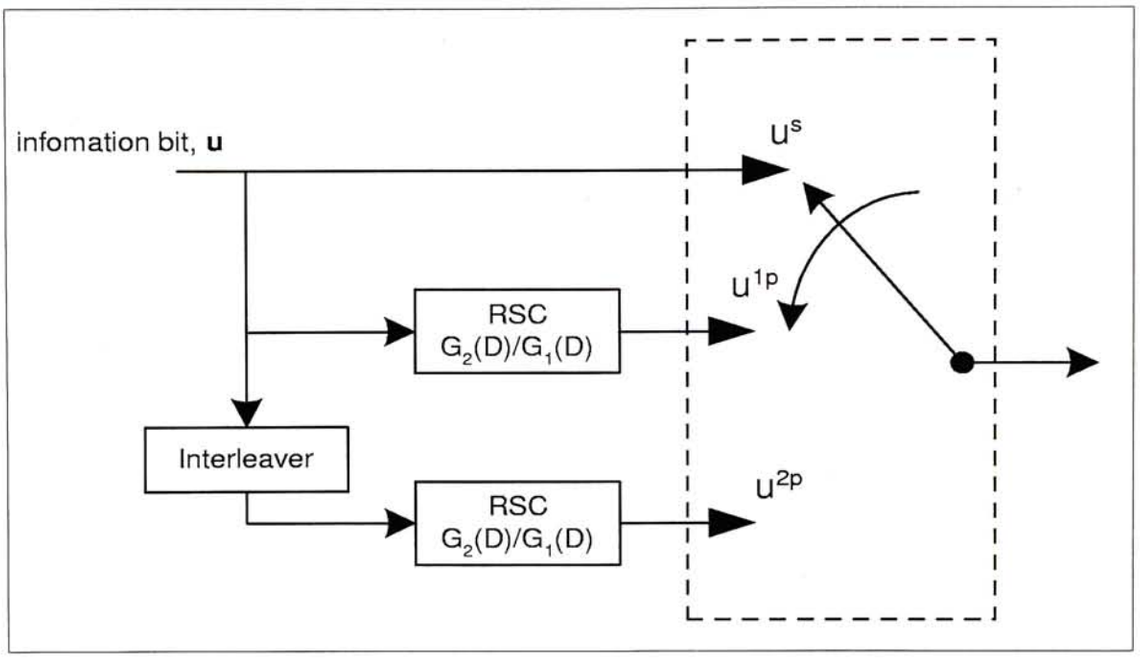

Figure 3.1: A Turbo Encoder

3.1 The Turbo Encoder

A Turbo Encoder structure is shown in figure 3.1. RSC encoders are chosen as the constitute encoders since Turbo Coding System was first appeared in the literature, rather than the traditional Non-recursive Systematic Convolutional (NSC) encoders. Generally speaking, any NSC or RSC encoders can be used as the constituent encoder of a turbo coding system since their trellis structures are identical to each other.

A RSC encoder is often described by a ratio of generator polynomials $G_2(D)/G_1(D)$. This ratio is often represented in octal form. For example, consider the 16-state RSC shown in the figure 3.2, $G_1(D) = 1 + D + D^2 + D^3 + D^4 = 37_8$ and $G_2(D) = 1 + D^4 = 21_8$. In fact, RSC encoder is essential for the ultimate

performance of Turbo Codes. Two of its key properties is as follows [13][29]

1. Due to its Infinite Impulse Response (IIR) property, A weight-one input will produce an infinite weight output. That is, input $u = [\cdots 010 \cdots]$ will not produce a low-weight codeword as the RSC encoder will generate a codeword with an infinite number of ones in the parity bits.

2. There exists a family of weight-two inputs (with $2 + 3t$ zeros between two ones, where t is a non-negative integer) which produce finite weight outputs. These are called *self-terminating* weight-two input sequence. For instance, a weight-two input $u = [\cdots 010010 \cdots]$ produce a low-weight output with $u^p = [\cdots 011110 \cdots]$.

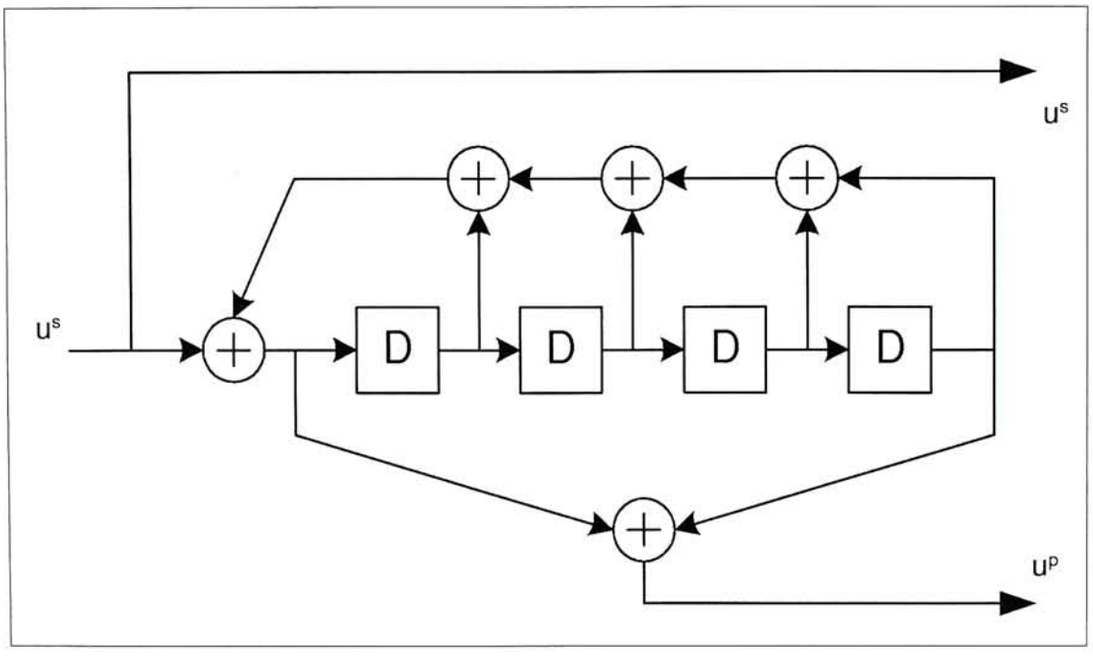

Figure 3.2: (1,21/37) RSC

An example will be given in the next section which explains how the combination of RSC encoders and interleavers could prevent pairing low weight codewords from the constituent encoders. For a more thorough discussion about RSC encoders, readers are advised to reach [18][19]

3.2 The Turbo Interleaver

The performance of Turbo Codes depends on how effectively data sequences which produces low-weight outputs are permuted to other sequences which yield outputs with higher weights in other RSC encoders. The following example shows the importance of RSC encoders and the interleavers [13][29].

> RSC encoders only generate low-weight outputs for particular weight-two input sequences mentioned in the last section. Consider the case of a rate 1/3 Turbo Code with interleaver size $N = 100$. For a weight-two input sequence $u = [\cdots 010010 \cdots]$ is to be inserted to the first RSC encoder, a low-weight output will only be resulted at the output of the second RSC encoder if the permuted version of the weight-two input sequence u is also a *self-terminating* weight-two input sequence. For a interleaver size $N = 100$, there are totally ${}_{100}C_2 = 4095$ possible permutations for weight-two sequences. And among these permutations, only 98 permutations are sequences with two ones separated by two zeros. Therefore, the probability for the outputs of the two separate RSC encoders which produce low-weight outputs at the same time is greatly reduced. Note that larger interleaver sizes perform better based on the above reasonings.

Another design issue for turbo interleavers is that for any input sequence which causes the first encoder to terminate should also cause the second encoder to terminate as well. Knowledge of the final state of the encoder is exploitable in the decoder. For small frames, performance benefits from this knowledge and extra bits are often added to the input frame to ensure this termination. Various literatures like [11][12] explore the design issues of turbo interleavers in more depth.

3.3 The Iterative Decoder

In turbo decoding, two MAP decoders or any other decoders based on different algorithms concatenated serially are required. There are three parts in the received code vectors, namely $\mathbf{y}^s$, $\mathbf{y}^{1p}$ and $\mathbf{y}^{2p}$. The bits in $\mathbf{y}^s$ are the received bits corresponding to the information bits. $\mathbf{y}^{1p}$ and $\mathbf{y}^{2p}$ are the received bits corresponding to the 1st and 2nd parity bits from the first and second constituent encoder respectively. The received information bits will be passed on to the first decoder and decoded with the parity bits from the first RSC encoder. Extrinsic information calculated by the first decoder will be passed to the second decoder which serves as a priori information of the information bits. Likewise, the second decoder will also calculate the extrinsic information based on the permuted version of information bits, parity bits of the permuted version of information bits and the a priori information from the first decoder. Similarly, extrinsic information from the second decoder will then pass back to the first decoder and the decoding iteration proceeds as described above. The idea behind extrinsic information is Decoder Two provides soft information to Decoder One for each

information symbol, using only the information not available to Decoder One. Decoder One does likewise for Decoder Two. We will then give a brief discussion on MAP, Max-Log MAP and Log MAP algorithms which could be employed as the iterative decoders of turbo coding systems.

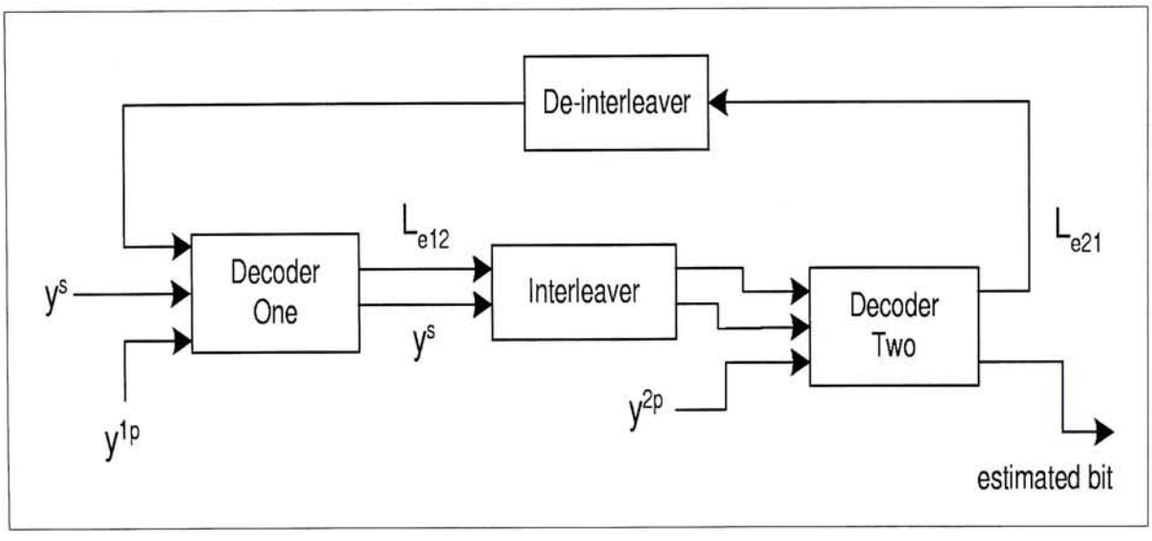

Figure 3.3: An Iterative Turbo Decoder

3.3.1 The MAP Algorithm

In a MAP decoder, it decides $u_k = +1$ if $P(u_k = +1|\mathbf{y}) > P(u_k = -1|\mathbf{y})$, and it decides $u_k = -1$ otherwise. More succinctly, the decision $\hat{u_k}$ is given by

$$\hat{u_k} = \begin{cases} +1 & \text{if } L(u_k) \geq 0 \\ -1 & \text{if } L(u_k) < 0 \end{cases} \tag{3.1}$$

where $L(u_k)$ is the Log Likelihood Ratio (LLR) defined as

$$L(u_k) \equiv \log\left(\frac{P(u_k = +1|\mathbf{y})}{P(u_k = -1|\mathbf{y})}\right) \tag{3.2}$$

Incorporating the code's trellis, this may be written as

$$L(u_k) = \log\left(\frac{\sum_{S^+} p(s_{k-1} = s', s_k = s, \mathbf{y})/p(\mathbf{y})}{\sum_{S^-} p(s_{k-1} = s', s_k = s, \mathbf{y})/p(\mathbf{y})}\right) \tag{3.3}$$

where $s_k \in S$ is the state of the encoder at time k, S^+ is the set of ordered pairs (s', s) corresponding to all state transitions $(s_{k-1} = s') \rightarrow (s_k = s)$ caused by data input $u_k = +1$, and S^- is similarly defined for $u_k = -1$ Based on the derivations by [5][10][22]

$$p(s', s, \mathbf{y}) = \alpha_{k-1}(s') \cdot \gamma_k(s', s) \cdot \beta_k(s) \tag{3.4}$$

where $\alpha_k(s) \equiv p(s_k = s, \mathbf{y}_1^k)/p(\mathbf{y}_1^k)$ is computed recursively as

$$\alpha_k(s) = \frac{\sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s)}{\sum_s \sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s)} \tag{3.5}$$

with initial conditions

$$\alpha_0(0) = 1 \quad \text{and} \quad \alpha_0(s \neq 0) = 0 \tag{3.6}$$

These conditions state that the encoder is expected to start in state 0. The probability $\beta_k(s) \equiv p(\mathbf{y}_{k+1}^N | s_k = s)/p(\mathbf{y}_{k+1}^N | \mathbf{y}_1^k)$ is computed recursively as

$$\beta_{k-1}(s') = \frac{\sum_s \beta_k(s)\gamma_k(s', s)}{\sum_s \sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s)} \tag{3.7}$$

with boundary conditions

$$\beta_N(0) = 1 \quad \text{and} \quad \beta_N(s \neq 0) = 0 \tag{3.8}$$

This condition is expected in the case of the encoder is end in state 0 caused by the last m termination bits.

The branch transition probability $\gamma_k(s', s)$ is given by

$$\begin{aligned} \gamma_k(s', s) &= p(y_k^s | u_k = i) \cdot p(y_k^p | u_k = i, s', s) \cdot \\ &\quad q(u_k = i | s', s) \cdot Pr\{s | s'\}; \qquad i = \pm 1 \end{aligned} \tag{3.9}$$

The value of $q(u_k = i|s', s)$ is either one or zero depending on whether bit i is associated with the transition from state s' to s or not. In the case of no parallel transitions, $Pr\{s'|s\} = Pr\{u_k = 1\}$ if $q(u_k = 1|s', s) = 1$; and $Pr\{s'|s\} = Pr\{u_k = -1\}$ if $q(u_k = -1|s', s) = 1$. Therefore, $\gamma_k(s', s)$ can be rewritten as

$$\gamma_k(s', s) = p(y_k^s|u_k = i) \cdot Pr\{u_k = i\} \cdot \gamma_k^e(s', s) \tag{3.10}$$

Note that the function $\gamma_k^e(s', s)$ is not a function of the systematic symbol at time k or the a priori probability for u_k. Substituting and simplifying $L(u_k)$, we have

$$\begin{aligned} L(u_k) &= \log\left(\frac{p(y_k^s|u_k = +1)}{p(y_k^s|u_k = -1)}\right) \\ &\quad + \log\left(\frac{Pr(u_k = +1)}{Pr(u_k = -1)}\right) \\ &\quad + \log\left(\frac{\sum_{s^+} \alpha_{k-1}(s') \cdot \gamma_k^e(s', s) \cdot \beta_k(s)}{\sum_{s^-} \alpha_{k-1}(s') \cdot \gamma_k^e(s', s) \cdot \beta_k(s)}\right) \end{aligned} \tag{3.11}$$

It is notable that the LLR has now decomposed into three components which can be represented as

$$L(u_k) = L_{systematic} + L_{apriori} + L_{extrinsic} \tag{3.12}$$

The first term $L_{systematic}$ is based only on the received systematic symbol at time k. The second term $L_{apriori}$ represents any A priori information about u_k provided by any previous decoder. And the third term $L_{extrinsic}$ represents extrinsic information that can be passed on to any subsequent decoder.

3.3.2 The Max-Log MAP Algorithm

However, the MAP algorithm is likely to be considered too complex for implementation in a real system. To avoid the number of complicated operations and

also number representation problems in the MAP algorithm, we should compute the logarithms of $\alpha_k(s)$, $\beta_k(s)$ and $\gamma_k(s', s)$ instead of $\alpha_k(s)$, $\beta_k(s)$ and $\gamma_k(s', s)$ in computer simulations.

By taking the logarithm of $\gamma_k(s', s)$ in (3.9) and inserting

$$p(y_k^s|u_k = i) = \frac{1}{\sqrt{\pi N_0}}\exp\Big(- \frac{1}{N_0}\big(y_k^s - u_k\big)^2\Big) \tag{3.13}$$

and

$$p(y_k^p|u_k = i, s', s) = \frac{1}{\sqrt{\pi N_0}}\exp\Big(- \frac{1}{N_0}\big(y_k^p - x_k^p\big)^2\Big) \tag{3.14}$$

we obtain the following expression for $q(.) = 1$

$$\log\big(\gamma_k(s', s)\big) = \frac{2y_k^s u_k}{N_0} + \frac{2y_k^p x_k^p}{N_0} + \log\big(Pr\{s|s'\}\big) + K \tag{3.15}$$

where K is a constant which will be canceled out in the calculation of $\log(\alpha_k(s))$ and $\log(\beta_k(s))$. Let us define $\bar{\gamma}_k(s', s) = \log(\gamma_k(s', s))$, $\bar{\alpha}_k(s) = \log(\alpha_k(s))$ and $\bar{\beta}_k(s) = \log(\beta_k(s))$. $\bar{\alpha}_k(s)$ equals

$$\begin{aligned} \bar{\alpha_k}(s) &= \log\Big(\sum_{s'} e^{\bar{\gamma}_k(s',s)+\bar{\alpha}_{k-1}(s')}\Big) \\ &\quad - \log\Big(\sum_{s}\sum_{s'} e^{\bar{\gamma}_k(s',s)+\bar{\alpha}_{k-1}(s')}\Big) \end{aligned} \tag{3.16}$$

In order to simply numerical operations, we use the following approximation

$$\log(e^{\delta_1} + \cdots + e^{\delta_n}) \approx \max_i \delta_i \tag{3.17}$$

Therefore, we obtain

$$\begin{aligned} \bar{\alpha_k}(s) &\approx \max_{s'}(\bar{\gamma}_k(s', s) + \bar{\alpha}_{k-1}(s')) \\ &\quad - \max_{s}(\max_{s'}(\bar{\gamma}_k(s', s) + \bar{\alpha}_{k-1}(s'))) \end{aligned} \tag{3.18}$$

and similarly

$$\begin{aligned}\bar{\beta}_{k-1}(s') &\approx \max_{s'}(\bar{\gamma}_k(s',s)+\bar{\beta}_k(s)) \\ &\quad -\max_{s}(\max_{s'}(\bar{\gamma}_k(s',s)+\bar{\alpha}_{k-1}(s'))) \end{aligned} \tag{3.19}$$

$L(u_k)$ now becomes

$$\begin{aligned}L(u_k) &\approx \max_{s^+}(\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k(s',s)+\bar{\beta}_k(s)) \\ &\quad -\max_{s^-}(\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k(s',s)+\bar{\beta}_k(s)) \end{aligned} \tag{3.20}$$

To write (3.20) in the form of (3.12), we write

$$\bar{\gamma}_k^e = \log(p(y_k^p|u_k,s',s)) + \log(q(u_k|s',s)) \tag{3.21}$$

Next, we insert this to (3.20)

$$\begin{aligned}L(u_k) &\approx \max_{s^+}(\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k^e(s',s)+\bar{\beta}_k(s)) \\ &\quad +\log(p(y_k^s|u_k=+1))+\log(Pr\{u_k=+1\}) \\ &\quad -\max_{s^-}(\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k^e(s',s)+\bar{\beta}_k(s)) \\ &\quad +\log(p(y_k^s|u_k=-1))+\log(Pr\{u_k=-1\}) \end{aligned} \tag{3.22}$$

which can be written as

$$\begin{aligned}L(u_k) &\approx \max_{s^+}(\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k^e(s',s)+\bar{\beta}_k(s)) \\ &\quad -\max_{s^-}(\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k^e(s',s)+\bar{\beta}_k(s)) - \frac{4y_k^s}{N_0} - L^e(u_k) \end{aligned} \tag{3.23}$$

The first two terms comprise the *extrinsic* component, the third term is the systematic information and the last term is the a priori component. Note that this decomposition is only possible for systematic codes. $L^e(u_k)$ can be written as

$$L^e(u_k) = \log\left(\frac{Pr\{u_k=+1\}}{Pr\{u_k=-1\}}\right) = \log\left(\frac{Pr\{s|s'\}}{1-Pr\{s|s'\}}\right) \tag{3.24}$$

if $q(u_k = +1|s, s') = 1$. And hence $\log(Pr\{s|s'\}) = L^e(u_k) - \log(1 + e^{L^e(u_k)})$. Using the approximation in (3.17)

$$\log(Pr\{s|s'\}) \approx L^e(u_k) - \max(0, L^e(u_k)) \tag{3.25}$$

On the other hand

$$L^e(u_k) = \log\left(\frac{Pr\{u_k = +1\}}{Pr\{u_k = -1\}}\right) = \log\left(\frac{1 - Pr\{s|s'\}}{Pr\{s|s'\}}\right) \tag{3.26}$$

if $q(u_k = -1|s, s') = 1$. And hence $\log(Pr\{s|s'\}) = -\log(1+e^{L^e(u_k)})$ which could be written as

$$\log(Pr\{s|s'\}) \approx -\max(0, L^e(u_k)) \tag{3.27}$$

3.3.3 The Log-MAP Algorithm

Because of the approximation (3.12) is used, the Max-Log MAP algorithm is suboptimal and yields an inferior soft output than the MAP algorithm. To calculate $\log(e^{\delta_1} + \cdots + e^{\delta_n})$ exactly, we write

$$\begin{aligned} \log(e^{\delta_1} + e^{\delta_2}) &= \max(\delta_1, \delta_2) + \log(1 + e^{-|\delta_1 - \delta_2|}) \\ &= \max(\delta_1, \delta_2) + f_c(|\delta_1 - \delta_2|) \end{aligned} \tag{3.28}$$

where $f_c(.)$ is a correction function. As a consequence, by correcting at each step the approximation made by the Max-Log MAP algorithm with the correction function, it will be equivalent to the original MAP algorithm[25]. Various treatments for $f_c(.)$ can be found in [17][20]

Chapter 4

Receivers for Channels with Long Impulse Responses

The Turbo Principle was further being applied to channel equalization in [15]. This is known as Turbo Equalization. Later on, Turbo Equalization is combined with Turbo Codes for combating channels with ISI [23][24][26]. These structures perform well in most of the communication channels.

In this chapter, we will discuss the motivations and detailed structures of our new receiver structures for combating channels with long impulse responses. In short, our proposed receiver structures are constituted with a channel shortening filter (CSF) mentioned in [2][3][4] and subsequent combined turbo equalization and turbo decoding blocks like [23][24][26]. Modifications are necessary for receivers like [23][24][26] in the case where the length of the communication channel impulse response is long .

4.1 Shortcomings for the Existing Models

Generally speaking, equalizer structures used in combined turbo equalization and turbo decoding are similar to the MAP decoders used in turbo decoders because of its ability in producing a priori information for subsequent decoding and equalization. It is also known for its ability in combating channels with spectral deep notches. However, the complexity of MAP Equalizers, like the MLSE equalizers, grows exponentially with the length of channel impulse responses. In some of the real applications, like indoor wireless communication systems or certain mobile communication systems in urban areas with numerous high-rise buildings, the length of actual channel impulse response is long and MAP equalizers would be too complex to be implemented. In this view point, suboptimal but simpler structures are derived and explored.

4.2 Proposed System Architecture

As the complexity of MAP equalizer structure increases exponentially with the length of channel impulse responses, one quick solution to compensate this problem is to add a channel shortening filter right before the combined equalizer and decoder so that the Desired Impulse Response (DIR) seen by the receiver would be shorter than the actual channel length.

In figure 4.1, $r'(t)$ could be modeled as output of the communication model in figure 1.1 with channel response $h'(t)$ and noise $n'(t)$. The length of channel impulse response $h'(t)$ is shorter than $h(t)$ so that receivers for $h'(t)$ could be less complex than that for $h(t)$. However, inefficiencies of the channel shortening filter probably produce noticeable amount of error rates and the coloured noise

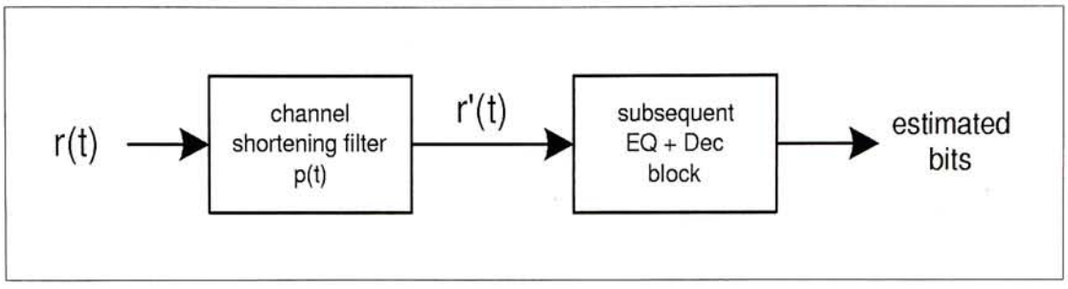

Figure 4.1: Overview of Proposed Receiver Structure

$n'(t)$ will case problems too.

As shown in figure 4.1, our proposed receiver is comprised with a channel shortening filter followed by subsequent equalizer and decoding blocks. Section 4.2.1 discuss the channel shortening filter used in our receiver. Sections 4.2.2 and 4.2.3 reveal two possible block structures for manipulating outputs of the CSF. Method One talked in section 4.2.2 is built in combined turbo equalization and turbo decoding with separate trellises. Method Two mentioned in section 4.2.3 is similar to method one but equalization and decoding are done together in a supertrellis. While the first method requires more but simpler equalizers and decoders, the second method requires less but more complicated blocks.

4.2.1 Optimized Model for Channel Shortening Filter

The necessities of utilizing channel shortening filters as the prefilter of MLSE equalizers in VA was already identified since MLSE in VA for combating ISI channels was first proposed [2]. Such a receiver combination was proposed and demonstrated in [3]. [4] attempted to adaptively optimized both the desired impulse response and the linear prefilter parameters in order to minimize the

mean square error at the output of the prefilter. We will employ results of [4] as the part of channel shortening filter. We will illustrate the model we employed here. Readers should refer to [4] for more discussions about this model.

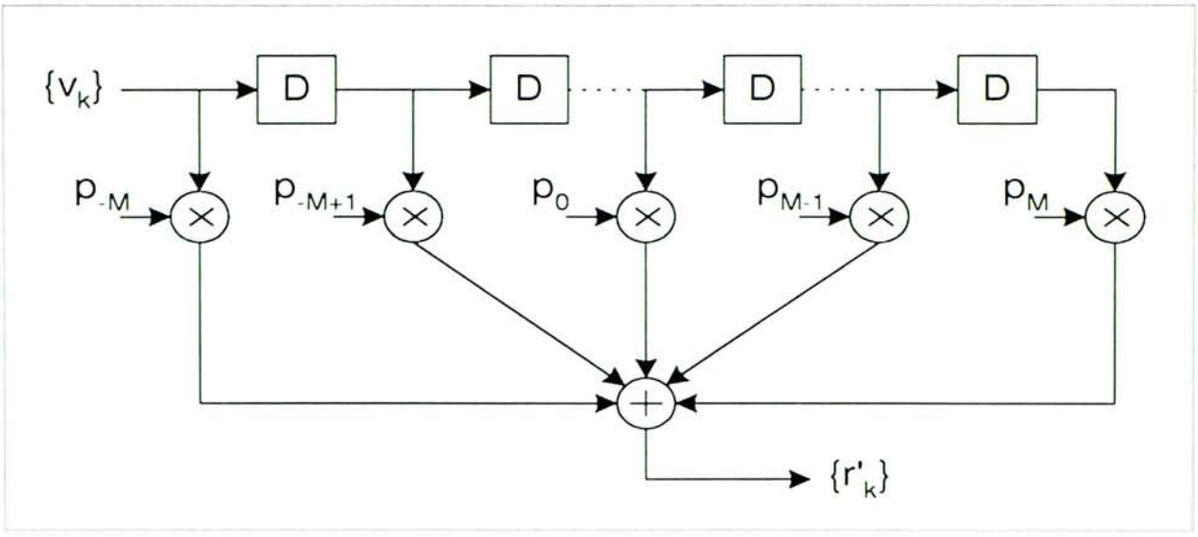

Figure 4.2: The Employed Channel Shortening Filter

The prefilter structure is shown in the figure 4.2. The received sequence feeds a linear tapped delay line filter whose function is to shorten the overall impulse response length and this filter has $L(=2M+1)$ taps . The output of this filter feeds other blocks for detecting and decoding information sequence. Blocks after the prefilter would make decisions on the assumption that the DIR $\{q_k\}_{k=0}^{LL}$ is the actual overall channel response, illustrated in figure 4.3. The value of LL (the length of the DIR) is much less than $2N+1$ (the length of the actual channel response). LL is chosen to make acceptable the complexity of the detector and the decoder while taking a small noise penalty in the linear preprocessing.

Recall indications in last chapters, (2.17) indicates the channel output, if the vectors $\mathbf{P}^t \equiv [p_{-M} \ \ldots \ p_0 \ \ldots \ p_M]$ and $\mathbf{Q}^t \equiv [q_0 \ \ldots \ q_{LL}]$ represent the

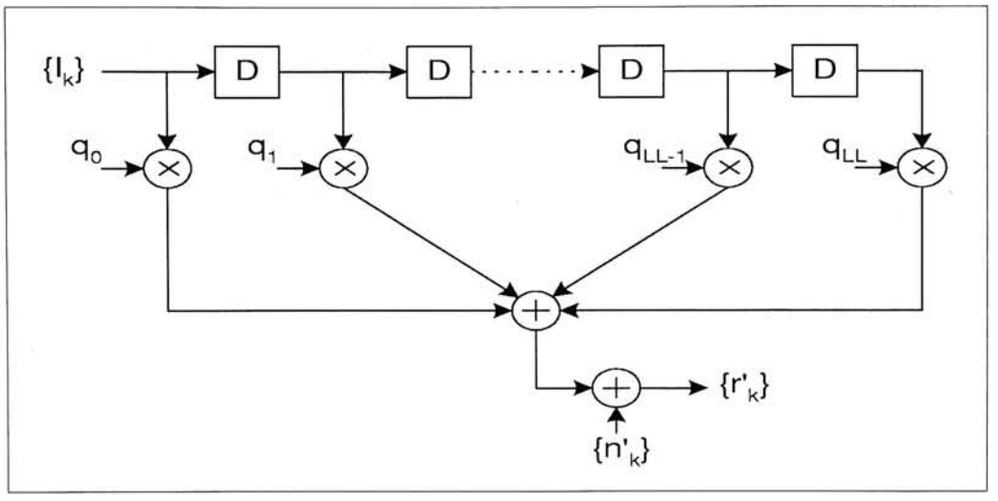

Figure 4.3: Resultant Model of Desired Impulse Response

transposes of the tap coefficients of the prefilter and the DIR respectively, the error in the kth interval, analogous to (2.18), is

$$\epsilon_k = \sum_{j=-M}^{M} p_j v_{k-j} - \sum_{j=0}^{LL} q_j I_{k-j} \tag{4.1}$$

In order to simplify the equations, it is assumed that the information sequence is uncorrelated and the information sequence estimate equals the information sequence. Substituting (2.17) into (4.1) and averaging ϵ_k^2, we get

$$E|\epsilon_k|^2 = \mathbf{P}^t \Gamma \mathbf{P} + \mathbf{Q}^t \mathbf{Q} - 2\mathbf{P}^t F \mathbf{Q} \tag{4.2}$$

where

$$F = \begin{pmatrix} f_M & \cdots & f_{M+LL} \\ \vdots & & \vdots \\ f_0 & \cdots & f_{LL} \\ \vdots & & \vdots \\ f_{-M} & \cdots & f_{-M+LL} \end{pmatrix} \tag{4.3}$$

is a $L \times (LL+1)$ matrix and Γ is a $(L \times L)$ channel covariance matrix the same as (2.23).

The error is minimized with respect to the prefilter by taking the gradient with respect to the taps $\{p_i\}$ and setting it equal to zero. The taps $\{q_i\}$ are constrained to be nonzero.

$$\frac{\partial E|\epsilon_k|^2}{\partial \mathbf{P}} = 2\Gamma\mathbf{P} - 2F\mathbf{Q} = 0 \tag{4.4}$$

and therefore

$$\mathbf{P}_{\text{opt}} = \Gamma^{-1}F\mathbf{Q} \tag{4.5}$$

If a fixed length is assumed for this desired response, the desired response can be optimized in the sense of minimizing MSE. Substituting (4.5) into (4.2), we obtain

$$E|\epsilon_k|^2 = \mathbf{Q}^t[I - F^t\Gamma^{-1}F]\mathbf{Q} \tag{4.6}$$

where I is the identity matrix.

Since $E|\epsilon_k|^2 \geq 0$, this is a positive definite quadratic form in $\mathbf{Q}$ which depends only upon $\mathbf{Q}$ and channel characteristics. This can be minimized by choosing $\mathbf{Q}$ to be the eigenvector with the minimum eigenvalue of the matrix $[I - F^t\Gamma^{-1}F]$. Therefore, to minimize the MSE, choose

$$\mathbf{P}_{\text{opt}} = \Gamma^{-1}F\mathbf{Q}_{\text{opt}} \tag{4.7}$$

where $\mathbf{Q}_{\text{opt}}$ is the eigenvector of $[I - F^t\Gamma^{-1}F]$ corresponding to its minimum eigenvalue. And minimum MSE J_{min} = minimum eigenvalue of $[I - F^t\Gamma^{-1}F]$.

4.2.2 Method One - Separate Trellises for EQ and DEC

The encoder side of this method is similar to figure 3.1 but encoded bits are further interleaved by a channel interleaver before transmission. There are totally two interleavers used, namely the turbo interleaver π and the channel interleaver τ. Consequently, the received sequences are equalized and decoded in separate blocks of MAP equalizers and MAP decoders iteratively, as illustrated in figure 4.4. All equalizer and decoder blocks can be built in any algorithm which could

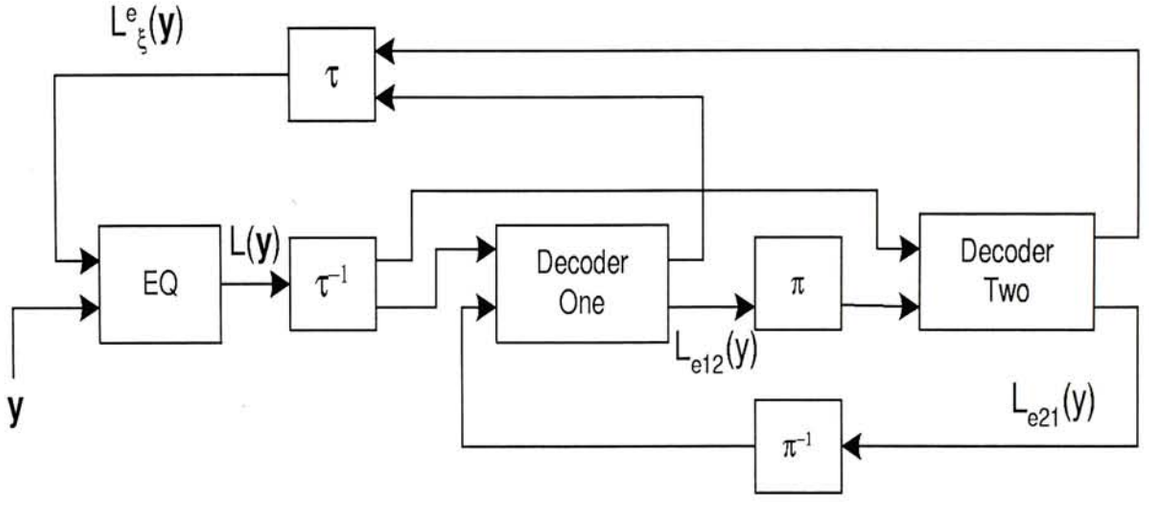

Figure 4.4: Receiver Structure for the Separate Trellises Scheme

produce soft information for subsequent equalization and decoding processes. However, it has to make sure that only extrinsic information is passed towards subsequent blocks to prevent positive feedback.

MAP Equalizer Block

MAP Equalizer Block is used based on [5] instead of ML Equalizers. This algorithm could be described as an algorithm which accepts LLRs at the input and

delivers LLRs at the output. These LLR values are calculated by equations similar to (3.3). The only difference between the metrics of the equalizer and of the channel decoder appears in the calculation of the branch transition probability γ_k. For the MAP equalizer γ_ξ can be calculated according to

$$\gamma_\xi(s', s) = \gamma_\xi^e(s', s) \cdot \exp\left(\frac{1}{2} \cdot x_\xi \cdot L(x_\xi)\right) \tag{4.8}$$

where

$$\gamma_\xi^e(s', s) = \exp\left(-\frac{1}{2\sigma^2} \cdot \left(y_\xi - \sum_{i=0}^{L} f_i x_{\xi-i}\right)^2\right) \tag{4.9}$$

y_ξ denotes the symbol received at time ξ and the equalizer uses an a priori information $L(x_\xi)$. As the ISI channel is non-systematic, we cannot separate channel information and extrinsic information in the output of the equalizer. Therefore, the overall LLR calculated by the MAP equalizer is

$$L(\tilde{x_\xi}) = L(x_\xi) + \log\left(\frac{\sum_{s^+} \alpha_{\xi-1}(s') \cdot \gamma_\xi^e(s', s) \cdot \beta_\xi(s)}{\sum_{s^-} \alpha_{\xi-1}(s') \cdot \gamma_\xi^e(s', s) \cdot \beta_\xi(s)}\right) \tag{4.10}$$

Constituent Decoder Blocks

For the decoder block, γ_k is obtain by

$$\gamma_k(s', s) = \exp\left(\frac{1}{2}\left(L(\tilde{x_k^s}) \cdot x_k^s + L(\tilde{x_k^p}) \cdot x_k^p + u_k \cdot L(u_k)\right)\right) \tag{4.11}$$

where $L(u_k)$ denotes the a priori information for the information bit u_k. $L(\tilde{x_k^s})$ and $L(\tilde{x_k^p})$ are the LLRs

$$L(\tilde{x_k^s}) = L(x_k^s|\mathbf{y}) = \log\left(\frac{p(x_k^s = +1|\mathbf{y})}{p(x_k^s = -1|\mathbf{y})}\right) \tag{4.12}$$

and

$$L(\tilde{x_k^p}) = L(x_k^p|\mathbf{y}) = \log\left(\frac{p(x_k^p = +1|\mathbf{y})}{p(x_k^p = -1|\mathbf{y})}\right) \tag{4.13}$$

In the serially concatenated scheme we use the output LLRs of the equalizer as an estimation of $L(\tilde{x_k^s})$ and $L(\tilde{x_k^p})$. In transmission over an AWGN channel without ISI, $L(\tilde{x_k^s})$ and $L(\tilde{x_k^p})$ would be in the form $L_c y_k^s$ and $L_c y_k^p$ respectively, where $L_c \equiv 2E_b/N_0$ as we assume code rate is 1/2.

For a systematic code, output of the decoder $L(\hat{u_k})$ can be expressed as a sum of a priori information $L(u_k)$, channel information $L(\tilde{x_k^s})$ and extrinsic information $L^e(\hat{u_k})$

$$L(\hat{u_k}) = L(u_k) + L(\tilde{x_k^s}) + \underbrace{\log\left(\frac{\sum_{s+} \alpha_{k-1}(s') \cdot \gamma_k^e(s', s) \cdot \beta_k(s)}{\sum_{s-} \alpha_{k-1}(s') \cdot \gamma_k^e(s', s) \cdot \beta_k(s)}\right)}_{L^e(\hat{u_k})} \tag{4.14}$$

where

$$\gamma_k^e(s', s) = \exp\left(\frac{1}{2} \cdot L(\tilde{x_k^p}) \cdot x_k^p\right) \tag{4.15}$$

Analogously the LLR of the parity bits would be

$$L(\hat{x_k^p}) = L(\tilde{x_k^p}) + \underbrace{\log\left(\frac{\sum_{s+} \alpha_{k-1}(s') \cdot \gamma_{k-1}^{e_p}(s', s) \cdot \beta_k(s)}{\sum_{s-} \alpha_{k-1}(s') \cdot \gamma_{k-1}^{e_p}(s', s) \cdot \beta_k(s)}\right)}_{L^{e_p}(\hat{u_k})} \tag{4.16}$$

where

$$\gamma_k^{e_p}(s', s) = \exp\left(\frac{1}{2}\left(L(\tilde{x_k^p}) \cdot x_k^p + u_k \cdot L(u_k)\right)\right) \tag{4.17}$$

$L^e(\hat{u_k})$ and $L^{e_p}(\hat{u_k})$ are the extrinsic information of the information bits and its corresponding parity bits respectively that are to be passed back to the MAP equalizers.

4.2.3 Method Two - Combined Trellises for EQ and DEC

Since the channel model can be represented in the form of finite state machine, equalization and decoding can be done jointly in a single process. This is done

by searching the best path in the supertrellis formed by combining trellises of the decoder and the equalizer. Therefore, in order to build the supertrellis, channel interleaver is not used. Note that since decoding and equalization are

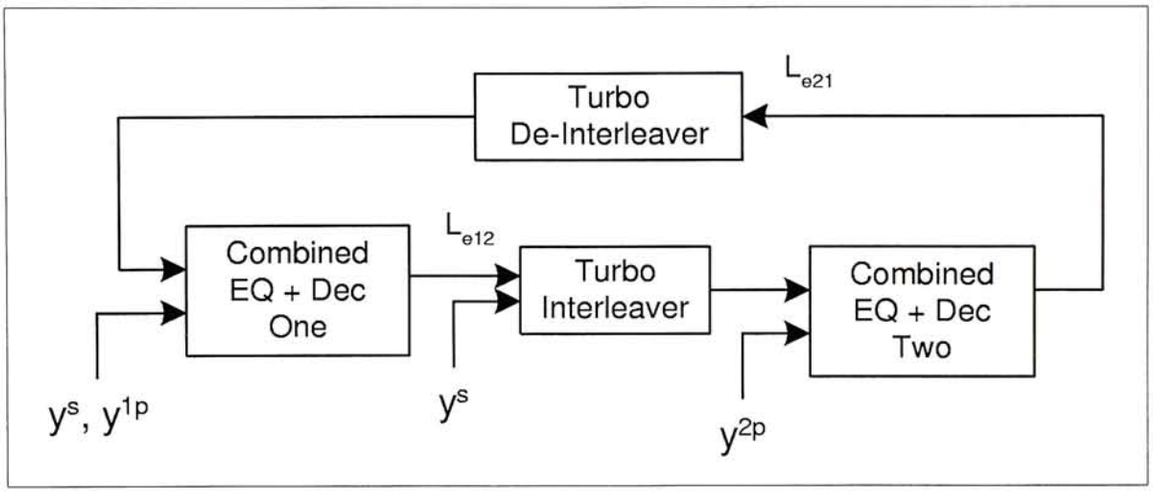

Figure 4.5: Receiver Structure for Combined Trellises Scheme

done jointly, the same class of bit should be grouped before transmission. In order words, the block to be transmitted should be in the form

$$[\cdots \quad y_0^s \quad y_1^s \quad \cdots \quad y_0^{1p} \quad y_1^{1p} \quad \cdots \quad y_0^{2p} \quad y_1^{2p} \quad \cdots] \tag{4.18}$$

where the same kind of bits is grouped first before transmission. The LLR values are calculated in the same way to the previous cases. Transitional probability γ_k, in the case of combined trellises, is calculated as

$$\begin{aligned} \gamma_k(s', s) &= \exp\left(-\frac{1}{4} L_c \left(\left(y_k^s - \sum_{i=0}^{L} f_i u_{k-i} \right)^2 + \left(y_k^p - \sum_{i=0}^{L} f_i x_{k-i}^p \right)^2 \right) \right) \\ &\quad \cdot \exp\left(\frac{1}{2} \cdot u_k \cdot L(u_k) \right) \end{aligned} \tag{4.19}$$

In this way, $L(\hat{u_k})$ is expressed as

$$L(\hat{u_k}) = L(u_k) + L^i(u_k) + L^e(u_k) \tag{4.20}$$

We express $L^i(u_k)$ and $L^e(u_k)$ as

$$L^i(u_k) = \log\left(\frac{\sum_{s+} \alpha_{k-1}(s') \cdot \gamma_k^i(s', s) \cdot \beta_k(s)}{\sum_{s-} \alpha_{k-1}(s') \cdot \gamma_k^i(s', s) \cdot \beta_k(s)}\right) \tag{4.21}$$

and

$$L^e(u_k) = \log\left(\frac{\sum_{s+} \alpha_{k-1}(s') \cdot \gamma_k^e(s', s) \cdot \beta_k(s)}{\sum_{s-} \alpha_{k-1}(s') \cdot \gamma_k^e(s', s) \cdot \beta_k(s)}\right) \tag{4.22}$$

where $\gamma_k^i(s', s)$ and $\gamma_k^e(s', s)$ in this case equal to

$$\gamma_k^i(s', s) = \exp\left(-\frac{1}{4}L_c\left(\left(y_k^s - \sum_{i=0}^{L} f_i u_{k-i}\right)^2\right)\right) \tag{4.23}$$

and

$$\gamma_k^e(s', s) = \exp\left(-\frac{1}{4}L_c\left(\left(y_k^p - \sum_{i=0}^{L} f_i x_{k-i}^p\right)^2\right)\right) \tag{4.24}$$

We name $L^i(u_k)$ the intrinsic information and $L^e(u_k)$ the extrinsic information. Note that only the extrinsic information should be passed onto subsequent decoders. Worth noting that the design and performance of this receiver structure is greatly constrained by the absence of channel interleaver and properties of the supertrellis structure. Two of the reasons are

1. Puncturing is unfavorable because supertrellis structure would be more complex since equalization and decoding are done at the same time.

2. Parity bits cannot be refined iteratively due to the absence of the channel interleaver and the properties of combined supertrellis structure.

Chapter 5

Performance Analysis

This chapter evaluates the performance of the proposed receiver structures. In section 5.1, we will list the platform and settings we used in our simulations. Since theoretical understandings and analysis are important in performance evaluation, we will give some limitations and expectations on the performance of the receivers in section 5.2. Finally, simulation results and discussions will be presented in section 5.3.

5.1 Simulation Model and Settings

It is expected that our newly proposed receiver structures perform better than separate equalization and turbo decoding while the complexity is lower than structures appeared in [23][24][26]. Based on these expectations, we would examine our receiver structures in several ways.

The communication system model employed is similar to figure 1.1. We

choose an $h(t)$ which is long (13 taps) and contain a deep notch. Its characteristics are illustrated in figure 5.1 and figure 5.2. Such a channel would be too complex to be implemented based on [23][24][26] while separate equalization (with linear equalizers) and turbo decoding would not perform well since of the spectral deep notch.

The optimized Channel Shortening Filter P plays an important role in the receiver structures. It helps in adjusting the length of the DIR and it is in charge of the performance versus complexity tradeoff. In the simulations we performed, we set LL for Q equals to 2, 3 and 4 respectively. And we choose method one and method two as the subsequent processor after the prefilter. We expect these LL values would fully reveal the characteristics of the prefilter used. Reasons for choices of LL would be mentioned in the next section.

Since design parameters of combined turbo equalization and turbo decoding with combined supertrellis are heavily constrained and are predicted to have inferior performance and complexity gains, we would only test this structure with LL equals to 2. In most of the cases, receivers with separate trellises for equalization and decoding are more preferable than the one with combined supertrellises.

In all simulations we performed, (1,5/7,5/7) Turbo Code is used with code rate 1/3. Max-Log MAP algorithm is used in all equalizer blocks and turbo decoder blocks since it requires less computer resources and avoids getting troubles from complicated operations and number representation problems. Information block size is 1024 including termination bits. All channel interleavers and turbo interleavers are generated randomly.

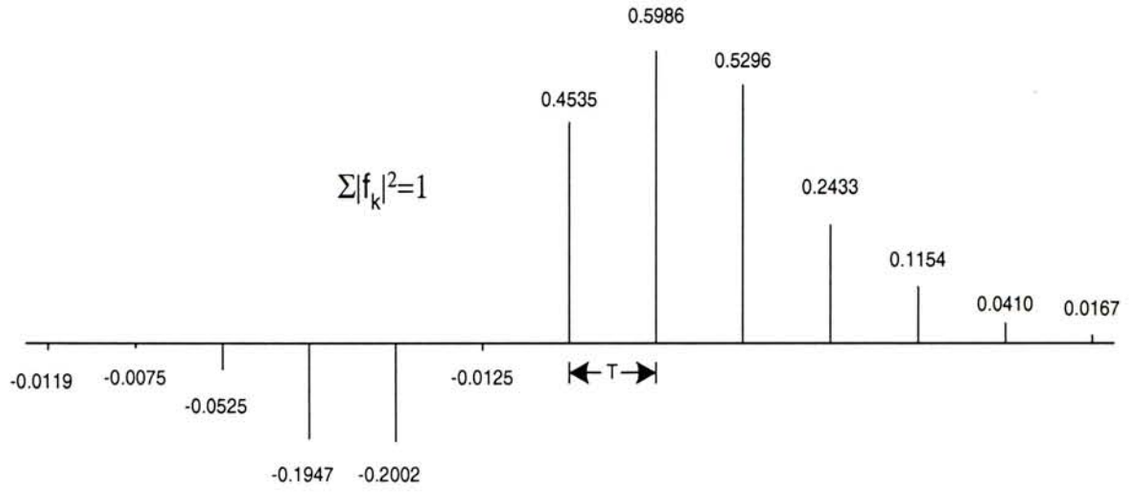

Figure 5.1: Discrete Time Channel used in Performance Analysis

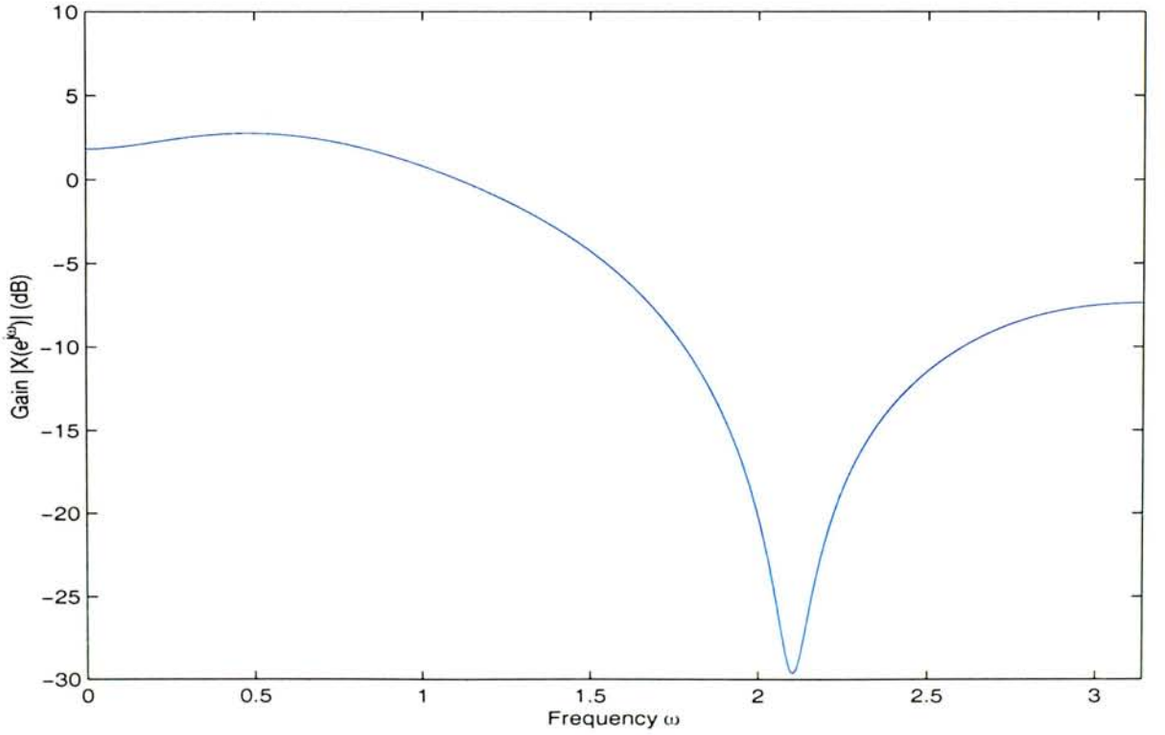

Figure 5.2: Frequency Response of the Corresponding channel

5.2 Performance Expectations

In the case of limited computing power available, theoretical analysis of the proposed system architecture is advantageous and time saving. However, exact numerical analysis is complicated because the existence of turbo inteleaver and channel interleaver. Here, we would try to anticipate the performance of the newly proposed receiver structures with simple reasonings. Based on performance prediction made here, we could foresee the results made by computer simulations and setup simulation strategies accordingly.

Separated Equalization and Turbo Decoding

The receiver in this case equalizes the received sequences and then decodes estimated sequences by the decoder. Equalization and decoding are done separately and no information exchange is performed between equalizer blocks and decoder blocks. Any kind of equalizers can be chosen as the equalizer blocks, however due to our target channel is a channel with long impulse response ($L = 12$), complex equalizers like MLSE cannot be used. Linear MMSE equalizer is chosen in our case. Since our simulation channel has got a spectral deep notch, this scheme, we expect, would perform the worst among all our simulation setups based on the understandings of the performance of linear equalizers[16][24].

Combined Equalization and Turbo Decoding with Separated Trellises

In our case, a channel shortening filter is employed in order to reduce the complexity of the equalizer in our proposed receivers. Surely suboptimal results would be obtained due to the inefficiency of the CSF. We can look at the relation between mean square error introduced by the CSF and the complexity of the receivers in different settings of the CSF based on the following table

LL	J_{min} @ SNR = 8 dB	J_{min} @ SNR = 9 dB	J_{min} @ SNR = 10 dB
1	0.3042	0.2831	0.2641
2	0.1667	0.1466	0.1289
3	0.1129	0.0952	0.0803
4	0.0925	0.0762	0.0627
5	0.0845	0.0691	0.0565
≀	≀	≀	≀
12	0.0789	0.0638	0.0514

As noticed from the above table, minimium mean square error J_{min}, and hence error rate, decreases faster if LL increases from 2 to 3 while it does not drop too much when LL become large. Note that the complexity of the equalizer grows exponentially with LL. As a result, we predict that performance will be saturated if LL approaches to the channel length and our proposed receiver structures are suboptimal even computational power is unlimited to allow any complex structure.

Frequency responses of the channel shortening filter P_{opt} and the corresponding desired impulse response Q_{opt} with different parameter settings are illustrated in figures 5.3-5.8. From these figures we know how DIR changes as length LL changes. [3] suggests many design criteria of the channel shortening filter. These figures help in realizing characteristics of the filter and they can be used to anticipate the performance of the whole system.

Combined Equalization and Turbo Decoding with Supertrellises

As the same as in method one, the value of LL used with method two is limited because of the growth of the complexity. This structure is more sensitive to the

value of LL because the number of states of the supertrellis grows in proportion to 2^{M+LL} where M is the number of memory blocks in the turbo encoder. Note that while the total number of states in the separate trellises scheme is in proportion to $2^M + 2^{LL}$.

In addition, several limitations in this model exist. The most frustrating one is that extrinsic information of the parity bits could not being re-used due to the absence of the channel interleaver. To prevent positive feedback, only extrinsic information of the information bits should be passed to subsequent decoder for processing. In addition, decoders built in a supertrellis are more complicated than decoders with two separated trellises. Thus, in most cases, as suggested in [27][28], method one is better than method two.

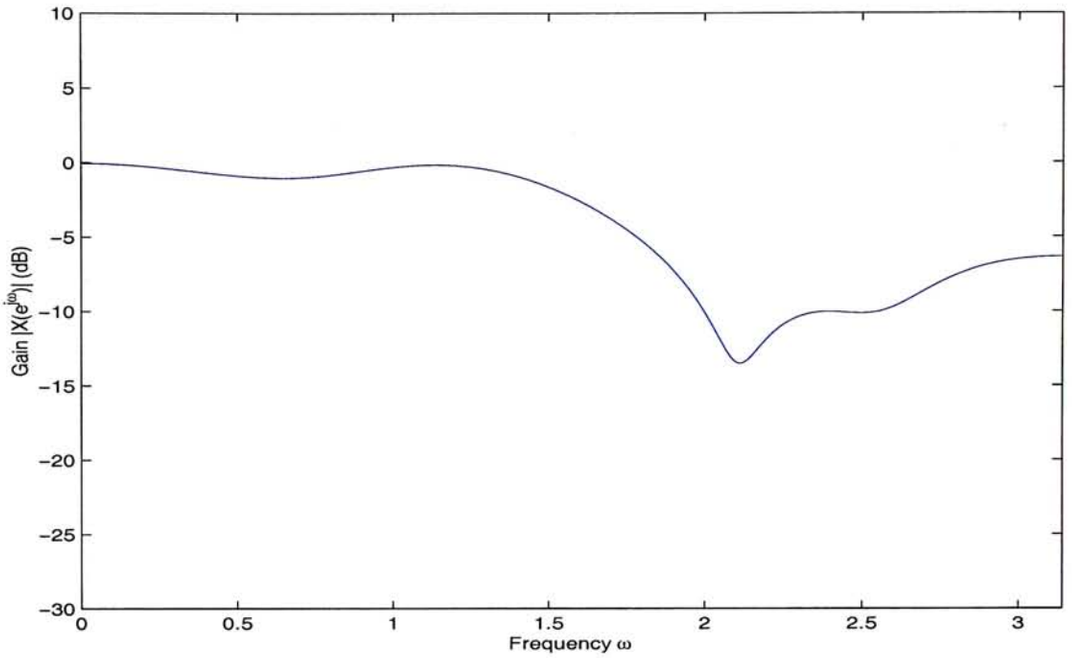

Figure 5.3: Frequency Response of P_{opt} @ 8 dB, $LL = 2$

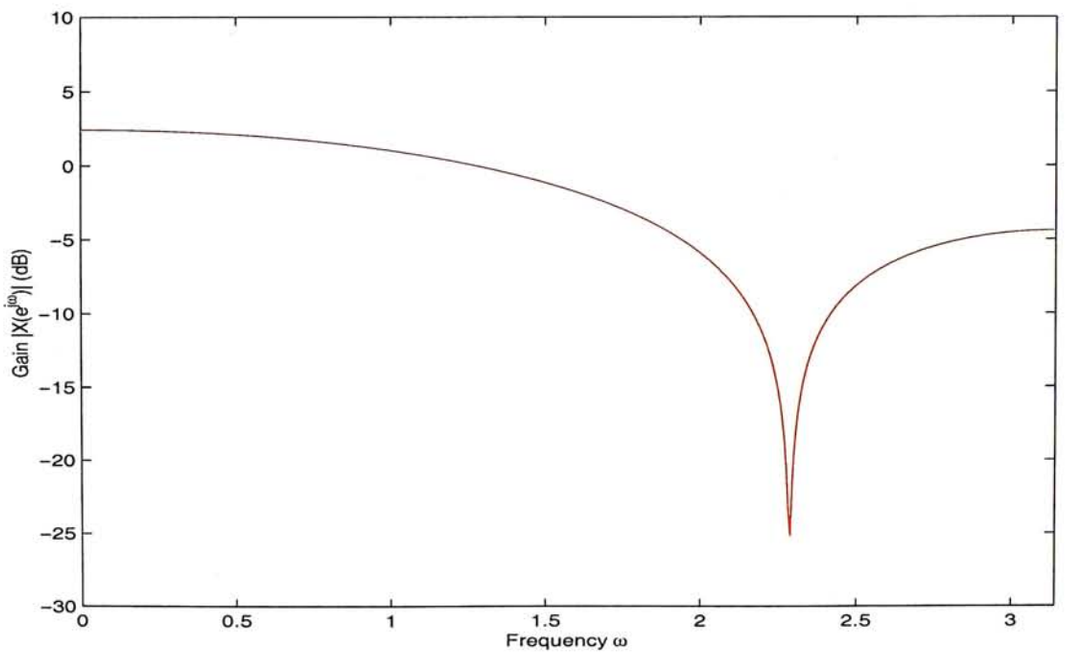

Figure 5.4: Frequency Response of Q_{opt} @ 8 dB, $LL = 2$

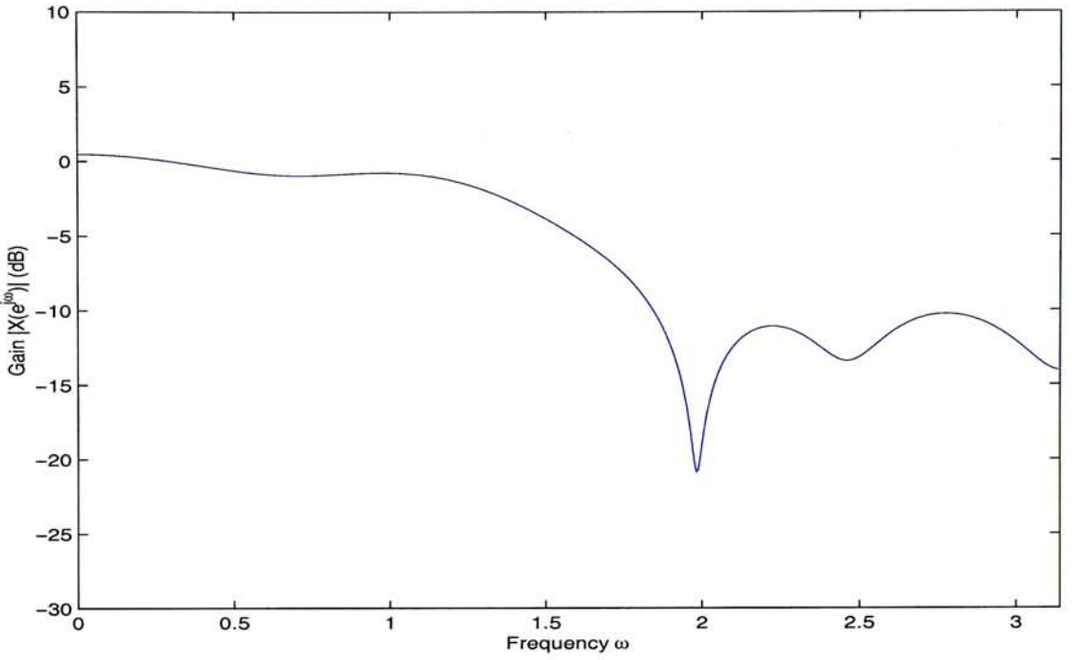

Figure 5.5: Frequency Response of P_{opt} @ 8 dB, $LL = 3$

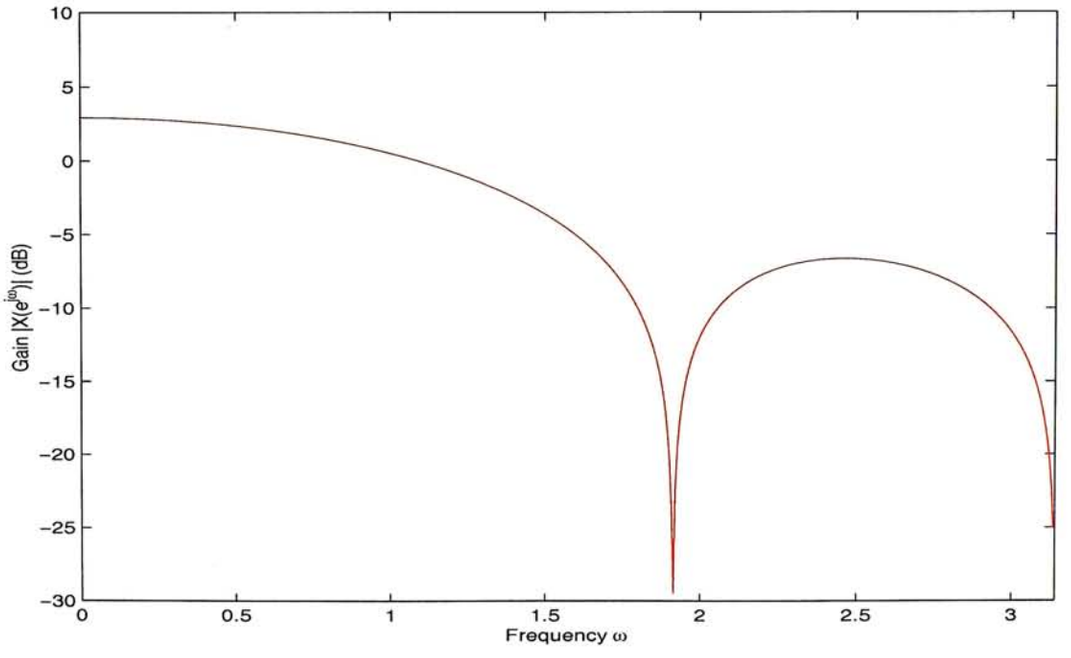

Figure 5.6: Frequency Response of Q_{opt} @ 8 dB, $LL = 3$

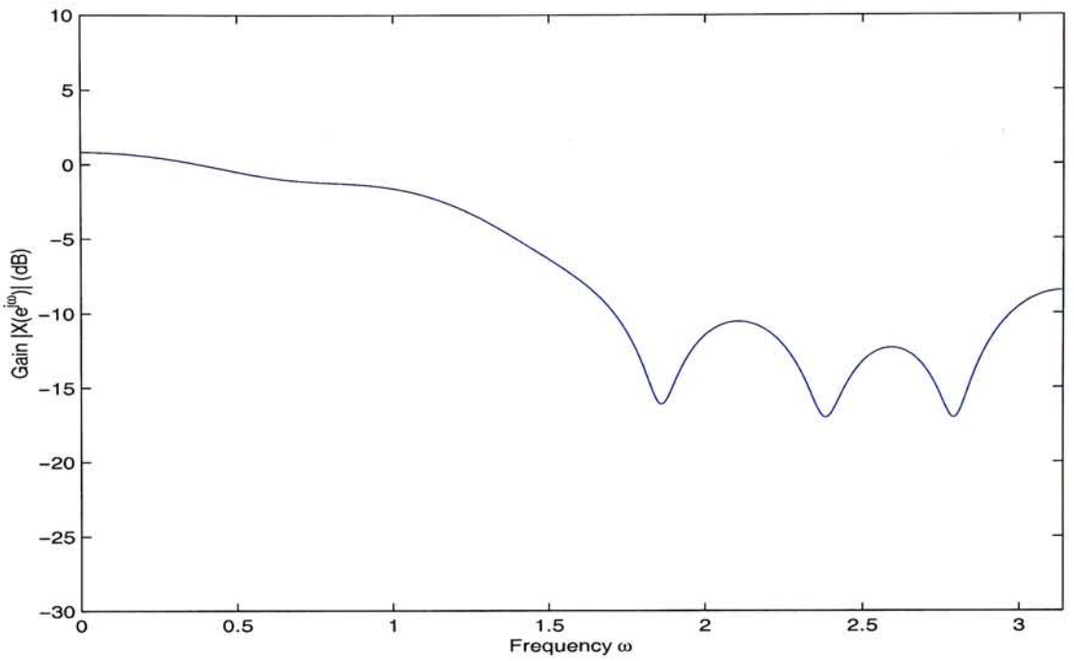

Figure 5.7: Frequency Response of P_{opt} @ 8 dB, $LL = 4$

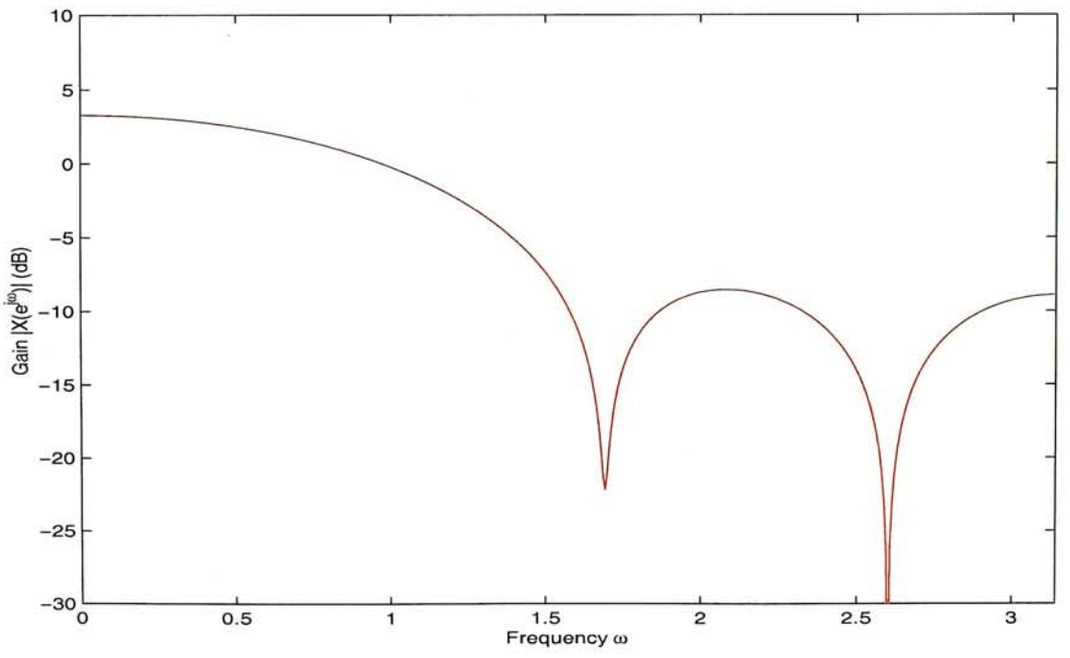

Figure 5.8: Frequency Response of Q_{opt} @ 8 dB, $LL = 4$

5.3 Simulation Results and Discussions

Simulation results are plotted in figures 5.9-5.16. Figure 5.9 illustrates the performance of our turbo decoder in AWGN channel. Figure 5.10 illustrates the performance of a system which employs a linear (MMSE) equalizer followed by a (1,5/7,5/7) turbo code decoder. Figures 5.11 and 5.12 demonstrate the performance of systems using the two methods we proposed with the length of DIR = 3. Figures 5.13 and 5.14 show the performance of systems employing method one with length of DIR equals 4 and 5 respectively. Finally, figure 5.15 makes a comparison between different lengths of DIR (3, 4 and 5 respectively) using method one (number of iteration = 12). And figure 5.16 makes a comparison between method one (number of iteration = 12), method two (number of iteration = 6) and the benchmark system (a linear equalizer followed by a turbo decoder, number of iteration = 12). It is shown that in figure 5.16 there is 2.4 dB gain at BER 10^{-4} by the newly proposed receivers over the benchmark system. Several characteristics about the receivers can be concluded based on the simulation results.

First, we can see in figures 5.11, 5.13 and 5.14 that the code gain obtained by method one in increasing the number of iterations decreases as the length of the desired impulse response increases. In addition, as shown in figure 5.15, the performance of method one decreases as the length of desired impulse response increases from 3 to 5. It turns out that while increasing the length of the desired impulse response reduces mean square error, decoder instability is also introduced and performance of method one does not increase as expected. To explain such phenomenon, in addition to minimizing the mean square error,

there exists some other design criteria to be obeyed in addition to minimizing the MSE.

Second, as illustrated in figure 5.12, there is not much gain in performance as to increase the number of iterations beyond 7 in method two. Simulations also show that method two is more stable and does not have any error re-growth during turbo iterations. In method two, only the extrinsic information of the information sequence is available for subsequent decoding. It may be one of the reasons why it is more stable than method one and its performance still approaches to method one which extrinsic information of information bits and parity bits are available for subsequent decoding.

Finally, from figure 5.16, both methods have similar code gains (2.4 dB) compared with the benchmark system. Method one needs 12 iterations while method two needs only 6 in order to reach the same BER with the same SNR. However, from figure 5.11 and 5.12, method one needs only about 0.4 dB more to reach BER 10^{-5} but method two needs about 1.3 dB more to reach BER 10^{-5}. Based on this reason, and other points mentioned in previous sections, method one is superior than method two.

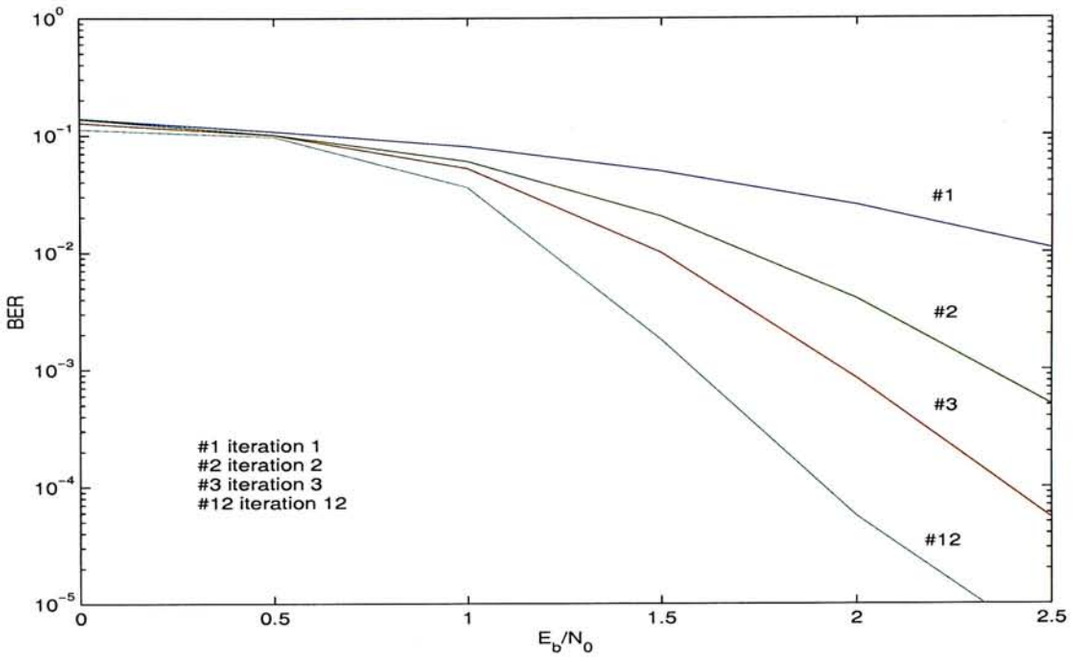

Figure 5.9: (1,5/7,5/7) Turbo Code in AWGN Channel

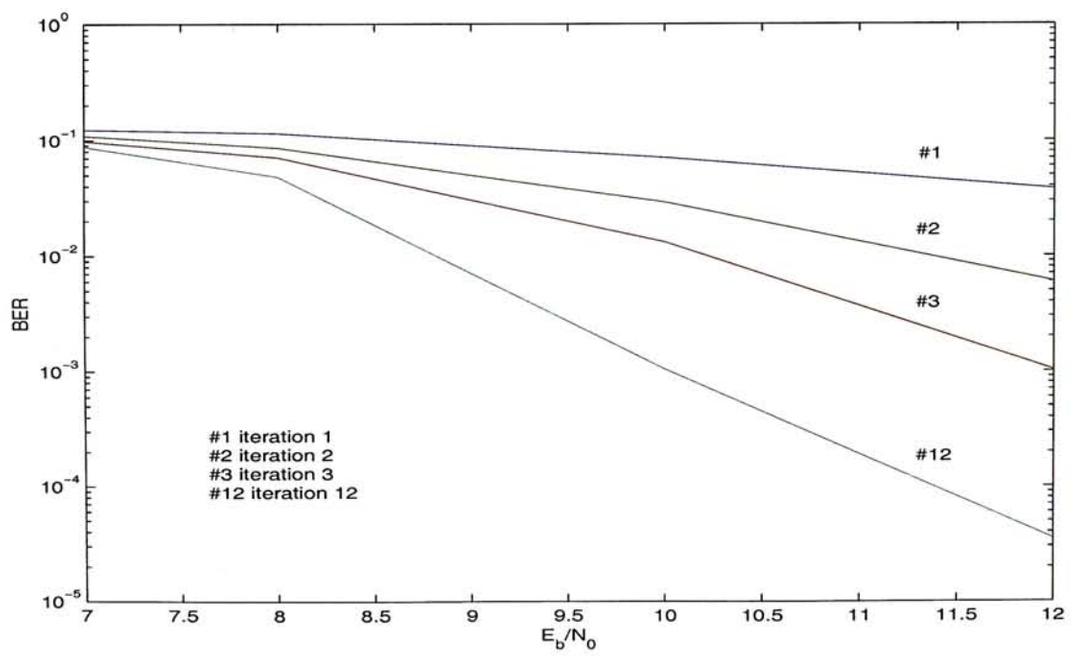

Figure 5.10: Equalization (MMSE) followed by Turbo Decoding

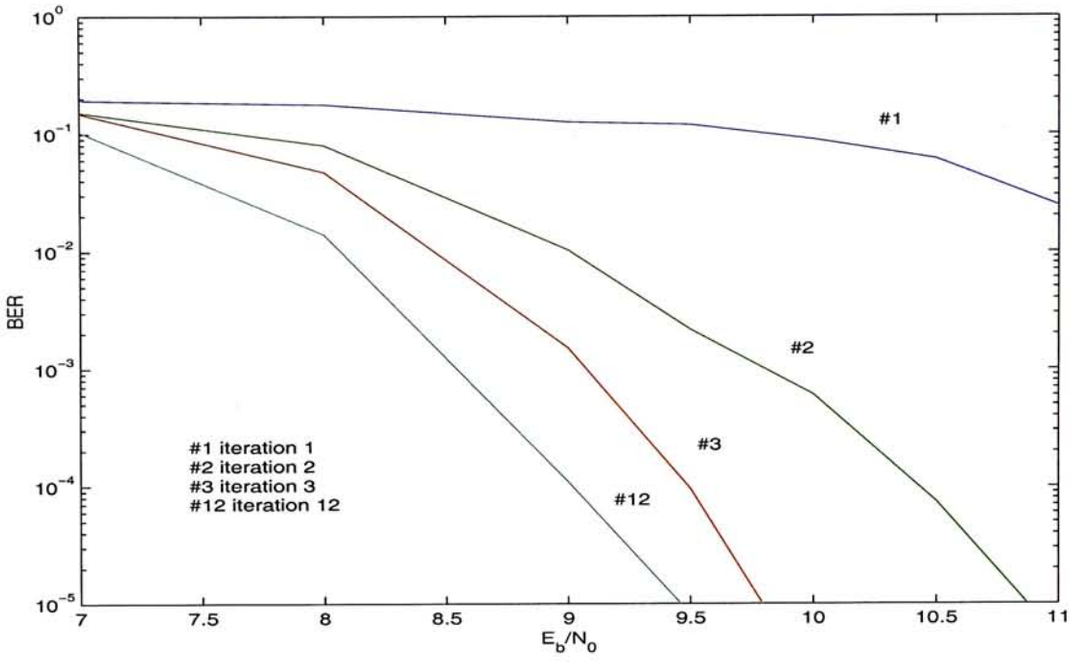

Figure 5.11: Method One, $LL = 2$ (3 taps)

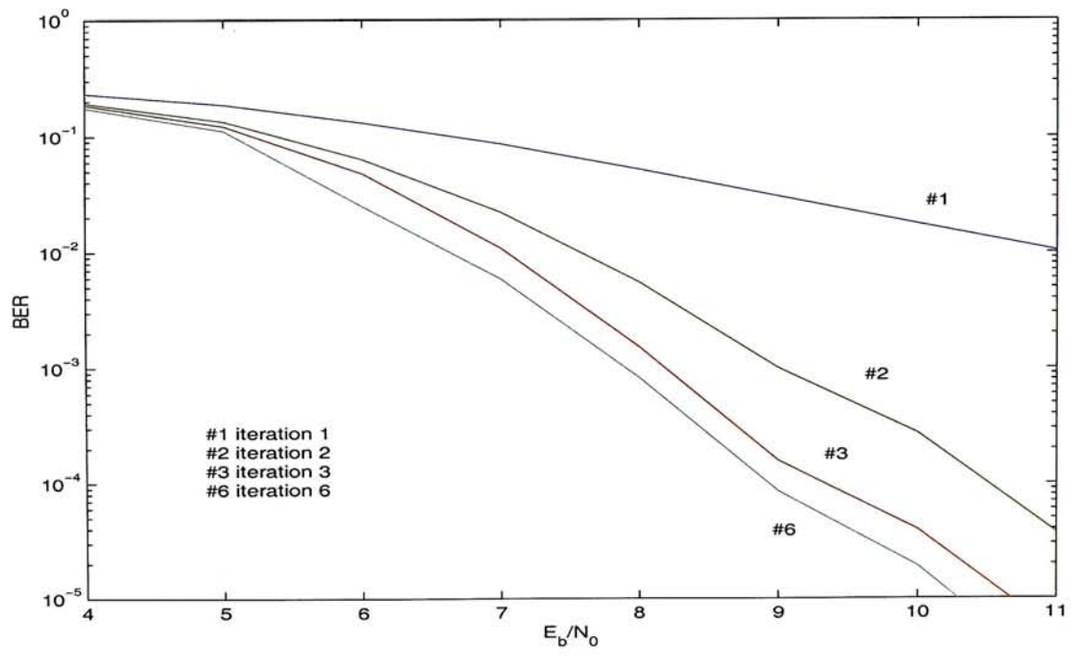

Figure 5.12: Method Two, $LL = 2$ (3 taps)

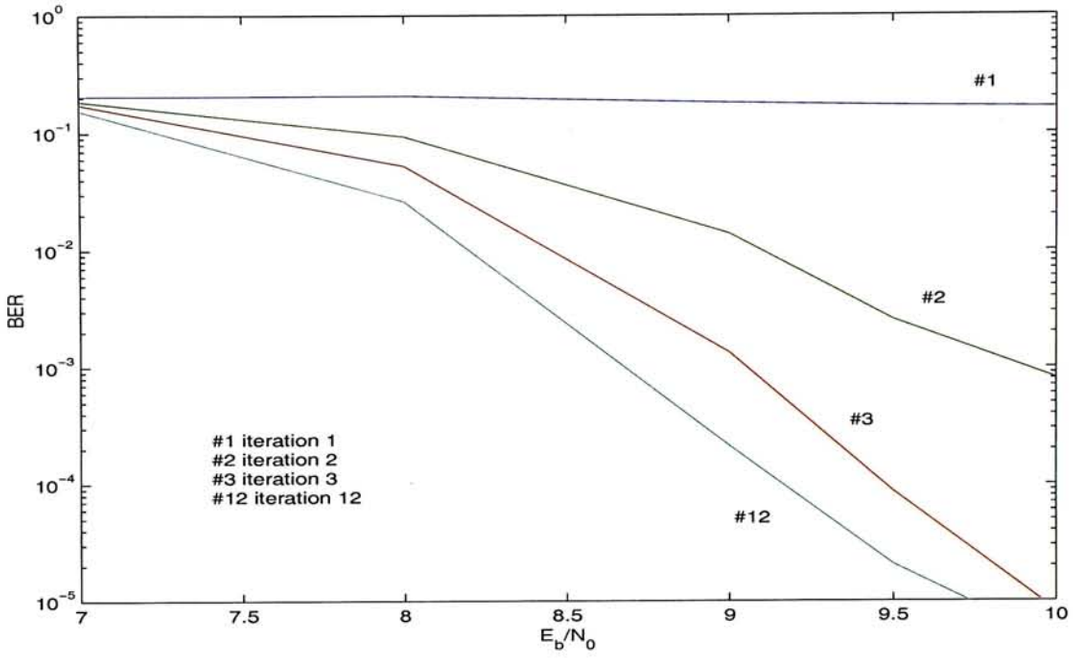

Figure 5.13: Method One, $LL = 3$ (4 taps)

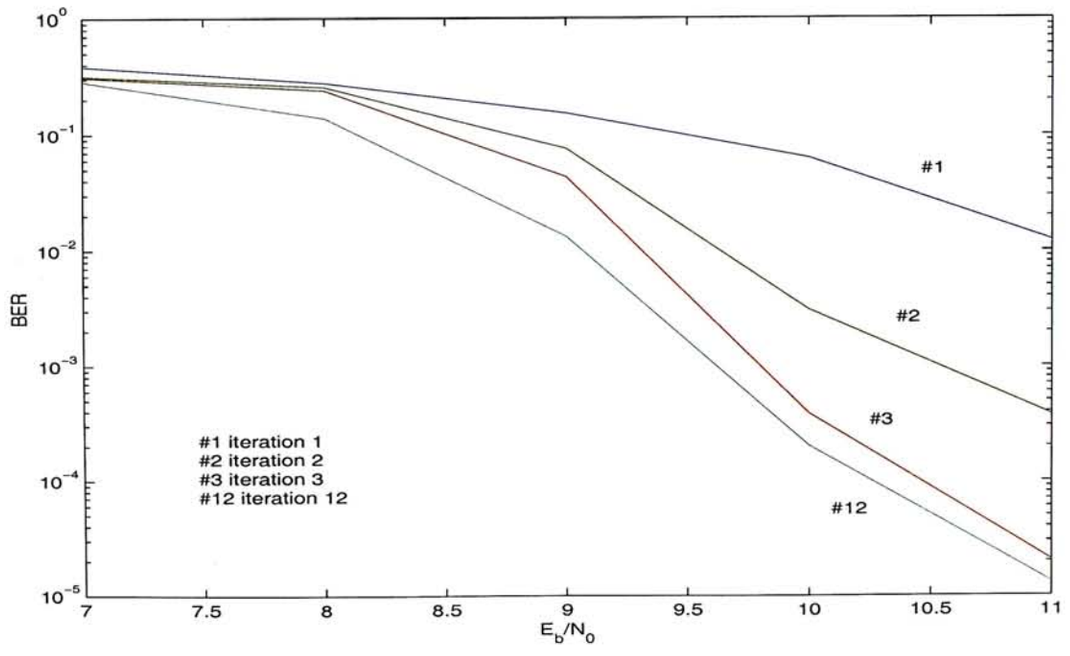

Figure 5.14: Method One, $LL = 4$ (5 taps)

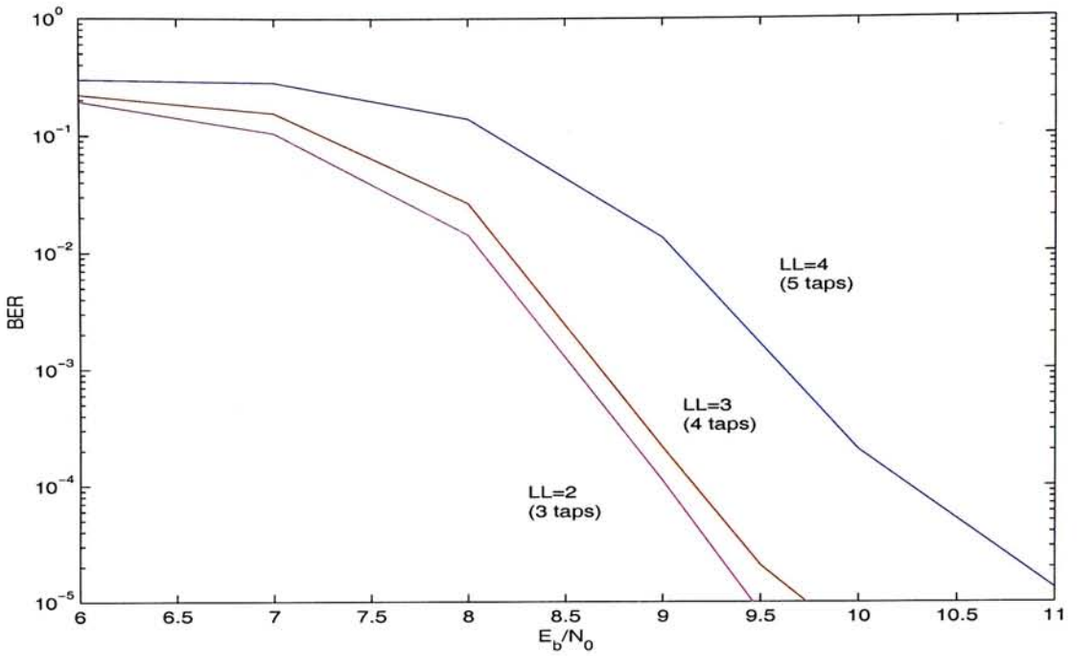

Figure 5.15: Method One, $LL = 2$, 3 and 4 (iteration = 12)

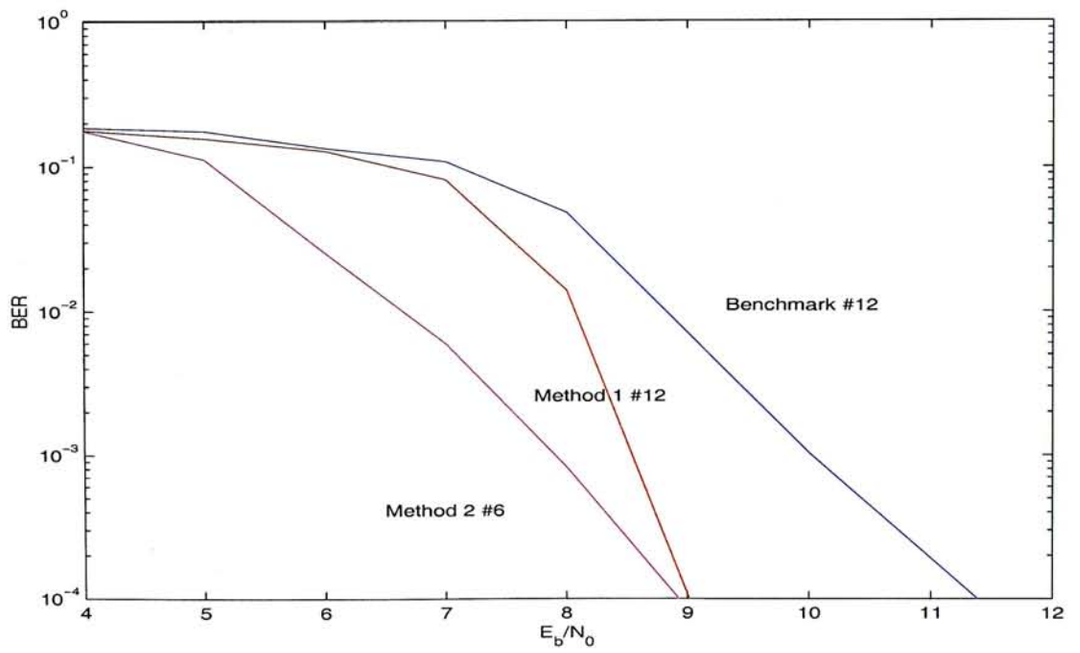

Figure 5.16: Method One, Two ($LL = 2$) and Benchmark System

Chapter 6

Concluding Remarks

Receiver structures comprising with a channel shortening filter and subsequent processing blocks by using combined turbo equalization and turbo decoding algorithms are proposed. 2.4 dB gain in BER 10^{-4} is obtained compared with receivers with separate linear (MMSE) equalization followed by turbo decoding.

In conclusion, method one is better to be used. The channel shortening filter also plays an important role in the performance of the system. However, it is found that the performance does not decrease by simply reduce the mean square error generated by the prefilter. There exists some other design criteria to be obeyed in addition to minizing the MSE.

While the key idea of our proposed methods is to shorten the original channel impulse response, another possible approach is to reduce the complexity of the equalizer by simplifying the equalizer algorithm [7]. This approach surely can escape from the negative influence of the channel shortening filter. Besides, as P_{opt} and Q_{opt} can be refined adaptively [4], our proposed receiver could be modified as an adaptive combined equalizer and decoder.

Bibliography

[1] C. E. Shannon, A Mathematical Theory of Communication, *The Bell System Technical Journal,* Vol. 27 (1948), pp. 379-423, 623-656.

[2] G. D. Forney, "Maximum-Likelihood Sequence Estimation of digitial sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory,* vol. IT-18, pp. 363-378, May 1972.

[3] Qureshi, S. U. H., and Newhall, E. E., "An Adaptive Receiver for Data Transmission over Time-Dispersive Channels," *IEEE Trans. Inform. Theory,* Vol. IT-19, No. 4, July 1973, pp. 448-457.

[4] Falconer, D. D., and Magee, F. R., Jr., "Adaptive Channel Memory Truncation for Maximum-Likelihood Sequence Estimation," *The Bell System Technical Journal,* Vol. 52, No. 9, July 1973, pp. 1541-1562

[5] L.R. Bahl, J. Cocke, F. Jeinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 248-287, March 1974.

[6] R. A. Wagstaff, A. Ronald, C. Berrou, L. Jean, "Robust Iterative Technique for High-Resolution Spatial Processing and Spectral Estimator," *United States Patent*, Patent Number: 4648057, Date of Patent: Mar. 3, 1987.

[7] M. V. Eyuboglu and S. U. Qureshi, "Reduced state sequence estimation with set partitioning and decision feedback," *IEEE Transactions on Communications,* 36:12-20, January 1988.

[8] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc., IEEE Globecom Conf.* (Dallas, TX, Nov. 1989), pp. 1680-1686.

[9] J. Lodge, R. Young, P. Hoeher and J. Hagenauer, "Separable MAP 'filters' for the decoding of product and concatenated codes," in *Proc. ICC '93*, pp. 1740-1745, May 1993.

[10] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC '93*, May 1993.

[11] A. S. Barbulescu and S. S. Pietrobon, "Interleaver design for turbo codes," *Electronics Letters,* vol. 30, pp. 2107-2108, Dec. 1994.

[12] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in *Globecom Conference,* pp. 1298-1303, 1994.

[13] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," *Proc. ICC '95,* Seattle, WA, June 18-22, 1995.

[14] C. Berrou, "Error-correction coding method with at least two systematic convolutional coding in parallel, corresponding iterative decoding method, decoding module and decoder," *United States Patent*, Patent Number: 5446747, Date of Patent: Aug. 29, 1995.

[15] C. Douillard, M. Jézéquel and C. Berrou, "Iterative correction of intersymbol interference: Turbo-Equalization," *European Trans. Telecomm.*, vol. 6, no. 5, pp. 507-511, Sept./Oct. 1995

[16] J. G. Proakis, *Digital Communications.* McGraw-Hill, 3rd ed., 1995

[17] S. S. Pietrobon, "Implementation and performance of a serial MAP decoder for use in an iterative turbo decoder," in *Proc. IEEE Int. Symp. Inform. Theory,* Whistler, B.C., Canada, 1995, p. 471

[18] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. on Info. Theory,* vol. 42, no.2, pp. 409-429, Mar. 1996.

[19] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. on Communications,* vol. 44, no. 5, pp. 591-600, May 1996.

[20] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Soft input soft output MAP module to decode parallel and serial concatenated codes," in *TDA Progr. Rep.* 42-127, Jet Propulsion Lab., Pasadena, CA, pp. 1-20, 1996

[21] T. S. Rappaport, *Wireless Communications.* Prentice Hall, 1996

[22] P. Robertson and P. Hoeher, Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding, *European Trans. on Telecommun.*, Vol. 8, No. 2, March-April 1997, p. 119-125.

[23] G. Bauch, H. Khorram, J. Hagenauer, "Iterative Equalization and Decoding in Mobile Communications Systems," *2nd EPMCC'97* in together with *3rd ITG-Fachtagung "Mobile Kommunikation"*, Bonn, Germany, October 1997.

[24] D. Raphaeli, Y. Zarai, "Combined turbo equalization and turbo decoding," *Global Telecommunications Conference, 1997. GLOBECOM '97., IEEE* Vol. 2 , 1997, pp. 639-643 vol.

[25] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE JSAC,* pp. 260-264, Feb. 1998

[26] D. Raphaeli, Y. Zarai, "Combined turbo equalization and turbo decoding," *IEEE Communications Letters* Vol. 24, April 1998 , pp. 107 -109.

[27] J. Garcia-Frias, J. D. Villasenor, "Blind turbo decoding and equalization," *Vehicular Technology Conference,* 1999 IEEE 49th Vol. 3, 1999, pp. 1881-1885 Vol. 3

[28] J. Garcia-Frias, J. D. Villasenor, "Combined blind equalization and turbo decoding," *Communication Theory Mini-Conference,* 1999, pp. 52-57.

[29] C. Heegard and S. B. Wicker, *Turbo Coding.* Kluwer Academic Publishers, 1999.