Design and Implementation of Belief Propagation Symbol Detectors for Wireless Intersymbol Interference Channels

by
Yanjie Peng

A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Electrical and Computer Engineering

December 2012

APPROVED:

Prof. Xinming Huang, Major Advisor, ECE Department, WPI

Prof. Andrew G. Klein, Co-Advisor, ECE Department, WPI

Prof. Ada Poon, Dissertation Committee, EE Department, Stanford University

Abstract

In modern wireless communication systems, intersymbol interference (ISI) introduced by frequency selective fading is one of the major impairments to reliable data communication. In ISI channels, the receiver observes the superposition of multiple delayed reflections of the transmitted signal, which will result errors in the decision device. As the data rate increases, the effect of ISI becomes severe. To combat ISI, equalization is usually required for symbol detectors. The optimal maximum-likelihood sequence estimation (MLSE) based on the Viterbi algorithm (VA) may be used to estimate the transmitted sequence in the presence of the ISI. However, the computational complexity of the MLSE increases exponentially with the length of the channel impulse response (CIR). Even in channels which do not exhibit significant time dispersion, the length of the CIR will effectively increase as the sampling rate goes higher. Thus the optimal MLSE is impractical to implement in the majority of practical wireless applications.

This dissertation is devoted to exploring practically implementable symbol detectors with near-optimal performance in wireless ISI channels. Particularly, we focus on the design and implementation of an iterative detector based on the belief propagation (BP) algorithm. The advantage of the BP detector is that its complexity is solely dependent on the number of nonzero coefficients in the CIR, instead of the length of the CIR. We also extend the work of BP detector design for various wireless applications.

Firstly, we present a partial response BP (PRBP) symbol detector with near-optimal performance for channels which have long spanning durations but sparse multipath structure. We implement the architecture by cascading an adaptive linear

equalizer (LE) with a BP detector. The channel is first partially equalized by the LE to a target impulse response (TIR) with only a few nonzero coefficients remaining. The residual ISI is then canceled by a more sophisticated BP detector. With the cascaded LE-BP structure, the symbol detector is capable to achieve a near-optimal error rate performance with acceptable implementation complexity. Moreover, we present a pipeline high-throughput implementation of the detector for channel length 30 with quadrature phase-shift keying (QPSK) modulation. The detector can achieve a maximum throughput of 206 Mb/s with an estimated core area of 3.162 mm^2 using 90-nm technology node. At a target frequency of 515 MHz, the dynamic power is about 1.096 W.

Secondly, we investigate the performance of aforementioned PRBP detector under a more generic 3G channel rather than the sparse channel. Another suboptimal partial response maximum-likelihood (PRML) detector is considered for comparison. Similar to the PRBP detector, the PRML detector also employs a hybrid two-stage scheme, in order to allow a tradeoff between performance and complexity. In simulations, we consider a slow fading environment and use the ITU-R 3G channel models. From the numerical results, it is shown that in frequency-selective fading wireless channels, the PRBP detector provides superior performance over both the traditional minimum mean squared error linear equalizer (MMSE-LE) and the PRML detector. Due to the effect of colored noise, the PRML detector in fading wireless channels is not as effective as it is in magnetic recording applications.

Thirdly, we extend our work to accommodate the application of Advanced Television Systems Committee (ATSC) digital television (DTV) systems. In order to reduce error propagation caused by the traditional decision feedback equalizer (DFE) in DTV receiver, we present an adaptive decision feedback sparsening filter BP (DFSF-BP)

detector, which is another form of PRBP detector. Different from the aforementioned LE-BP structure, in the DFSF-BP scheme, the BP detector is followed by a nonlinear filter called DFSF as the partial response equalizer. In the first stage, the DFSF employs a modified feedback filter which leaves the strongest post-cursor ISI taps uncorrected. As a result, a long ISI channel is equalized to a sparse channel having only a small number of nonzero taps. In the second stage, the BP detector is applied to mitigate the residual ISI. Since the channel is typically time-varying and suffers from Doppler fading, the DFSF is adapted using the least mean square (LMS) algorithm, such that the amplitude and the locations of the nonzero taps of the equalized sparse channel appear to be fixed. As such, the channel appears to be static during the second stage of equalization which consists of the BP detector. Simulation results demonstrate that the proposed scheme outperforms the traditional DFE in symbol error rate, under both static channels and dynamic ATSC channels.

Finally, we study the symbol detector design for cooperative communications, which have attracted a lot of attention recently for its ability to exploit increased spatial diversity available at distributed antennas on other nodes. A system framework employing non-orthogonal amplify-and-forward half-duplex relays through ISI channels is developed. Based on the system model, we first design and implement an optimal maximum-likelihood detector based on the Viterbi algorithm. As the relay period increases, the effective CIR between the source and the destination becomes long and sparse, which makes the optimal detector impractical to implement. In order to achieve a balance between the computational complexity and performance, several sub-optimal detectors are proposed. We first present a multitrellis Viterbi algorithm (MVA) based detector which decomposes the original trellis into multiple parallel irregular sub-trellises by investigating the dependencies between the received

symbols. Although MVA provides near-optimal performance, it is not straightforward to decompose the trellis for arbitrary ISI channels. Next, the decision feedback sequence estimation (DFSE) based detector and BP-based detector are proposed for cooperative ISI channels. Traditionally these two detectors are used with fixed, static channels. In our model, however, the effective channel is periodically time-varying, even when the component channels themselves are static. Consequently, we modify these two detector to account for cooperative ISI channels. Through simulations in frequency selective fading channels, we demonstrate the uncoded performance of the DFSE detector and the BP detector when compared to the optimal MLSE detector. In addition to quantifying the performance of these detectors, we also include an analysis of the implementation complexity as well as a discussion on complexity/performance tradeoffs.

Acknowledgements

First of all, I would like to express my deepest gratitude for my advisor Professor Xinming Huang, and co-advisor Professor Andrew G. Klein for their guidance through all stages of my Ph.D. studies and research at the Worcester Polytechnic Institute. I am very grateful for their great inspiration, patience, recognition and support that have helped me overcome difficulties and finally made this dissertation possible.

I would like to thank Professor Ada Poon for her insightful comments as my dissertation committee, and many other professors in the ECE Department from whom I have learned a lot.

Also, I deeply appreciate the support from The MathWorks fellowship to WPI. Thanks for the help from my manager Bharath Venkataraman, Don Orofino, Mike McLernon and HDL group members.

At last, I wish to thank my WPI colleagues Dr. Cao Liang, Dr. Wenxuan Guo, Dr Kai Zhang, Dr Qingxiong Deng, Chen Shen, and Wei Wang, for their friendship and support.

This dissertation is dedicated to my parents, who always give me their love, support and encouragement during all my life.

Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

8-VSB 8-vestigial sideband

ACS Add-compare-select

AF Amplify-and-forward

ATSC Advanced Television Systems Committee

AWGN Additive white Gaussian noise

BER Bit error rate

BMU Branch metric unit

BP Belief propagation

CIR Channel impulse response

CSF Channel shortening filter

CTV Check-to-variable

DD-LMS Decision-directed LMS

DFE Decision feedback equalizer

DFSE Decision feedback sequence estimation

DFSF Decision feedback sparsening filter

DTV Digital television

FDE Frequency domain equalization

FIR Finite impulse response

HDTV High-definition television

ISI Intersymbol interference

LDPC Low-density parity-check

LE Linear equalizer

LLR Log-likelihood ratio

LMS Least mean square

LPU Layer processing unit

LUT Look-up table

MAC Multiplier-accumulator

MLSE Maximum-likelihood sequence estimation

MMSE Minimum mean squared error

MVA Multitrellis Viterbi algorithm

NAF Non-orthogonal amplify-forward

OFDM Orthogonal Frequency Division Multiplexing

PRE Partial response equalizer

PRML Partial response maximum-likelihood

PTVA Parallel trellis Viterbi algorithm

QPSK Quadrature phase-shift keying

RLS Recursive least squares

SER Symbol error rate

SMU Survivor-path memory unit

SNR Signal-to-noise ratio

SRRC Squared root raised cosine

TDE Time domain equalization

TIR Target impulse response

TOV Threshold of visibility

VA Viterbi algorithm

VTC Variable-to-check

WF Whitening filter

Chapter 1

Introduction

In this chapter, we first introduce background and discuss motivations of our work in Section 1.1. The major contributions of our work are summarized in Section 1.2. Finally the organization of this dissertation is presented in Section 1.3.

1.1 Motivations

The demand for high speed wireless applications has been increasing rapidly recently. A major challenge for reliable high speed data transmission is the intersymbol interference (ISI) introduced by frequency selective fading. For wideband communication systems, different frequency components of the transmitted signal experience different attenuation, leading to distorted signals at the receiver. The ISI can span hundreds of symbols over a typical wireless channel for high data rate application. In order to mitigate the ISI, effective channel equalization is required for the physical layer of symbol detectors.

Historically multi-carrier (MC) modulation [1], particularly Orthogonal Frequency

Division Multiplexing (OFDM) scheme, has been applied to a large variety of wideband communication systems to efficiently combat frequency selective fading. However, MC transmissions undergo several drawbacks, such as large peak-to-average-power ratio (PAPR), intolerance to amplifier nonlinearities, and high sensitivity to carrier frequency offsets (CFOs) [2], which result in an increased cost of front-end design for strict demand on power amplification and highly accurate frequency synchronization.

Meanwhile, single-carrier (SC) transmissions, as a conventional scheme, is still attractive for its low cost in various applications, like high-definition television (HDTV) [3], underwater acoustic communication [4]. Equalization techniques for SC transmissions, either in time domain or in frequency domain, have been studied intensively. As a promising candidate to mitigate ISI, single-carrier frequency domain equalization (SC-FDE) [5] inherits similar merits of OFDM systems without the high PAPR and CFOs issues. Like other FDE schemes, in order to prevent contamination of a block by ISI from the previous block, redundant data called cyclic prefix, is inserted at the beginning of each block at the transmitter, and is discarded from each block at the receiver. The length of the cyclic prefix must be at least the impulse response span of the multipath channel, which imposes extra overhead to the bandwidth usage. Single-carrier time domain equalization (SC-TDE) schemes are favorable due to the high bandwidth efficiency. Among the well-known SC-TDE schemes, maximum-likelihood sequence estimator (MLSE) using the Viterbi algorithm [6] provides optimal performance to combat ISI in the presence of additive white Gaussian noise (AWGN). However, the computational complexity of MLSE increases exponentially with the channel length, making it impractical to implement in the majority of practical wireless applications. Suboptimal approaches with reasonable complexity, in-

cluding linear equalizers (LEs) and nonlinear decision feedback equalizers (DFEs) [7], have been widely used in the past. However, the performance of LEs is unsatisfactory due to the noise enhancement issue, while the DFEs suffer the error propagation problem. Therefore, high-performance low-complexity SC-TDEs for severe ISI channels are highly desirable.

1.2 Summary of Contributions

We are motivated to design and implement a group of symbol detectors in time domain for SC transmissions over wireless ISI channels. Our contributions are summarized as follows.

- Design and implementation of a low-complexity symbol detector for sparse channels.

  Sparse channels are characterized as having long multipath delay spreads but with few nonzero coefficients. Such channels arise in a number of modern wireless communication applications such as underwater acoustic communications, hilly terrain broadband wireless communications, and high-definition television (HDTV). We present a partial response belief propagation (PRBP) based symbol detector with near-optimal performance for sparse channels. The architecture is implemented by cascading an adaptive LE with the BP detector, which enables the detector to achieve a near-optimal error rate performance with acceptable implementation complexity. Moreover, the architecture of the BP detector is based on a pipelined layer processing scheme, which allows high throughput with low complexity. We implement the proposed LE-BP detector for channel length 30 with quadrature phase-shift keying (QPSK) modulation.

The detector can achieve a maximum throughput of 206 Mb/s with an estimated core area of 3.162 mm^2 using 90-nm technology node. At a target frequency of 515 MHz, the dynamic power is about 1.096 W.

We emphasize our contributions on the algorithm and hardware design in the following aspects:

- We consider the use of Least Mean Square (LMS) algorithm in the target impulse response (TIR) calculation, as the LMS algorithm is preferred for practical hardware implementation.
- We extend the BP algorithm in the existing literature [8] from BPSK to M-ary modulation system. If the complexity of the same algorithm in binary representation is x, then the computation complexity for QPSK is x^2. It is analogous to the non-binary LDPC decoder complexity problem.
- We simplify the calculation of the check-to-variable (CTV) message using Jacobian logarithm to facilitate hardware implementation.
- We conduct extensive experimental study to evaluate the impact of TIR structure, the number of iterations, the number of non-zero coefficients, and the fixed-point quantization.
- To the best of our knowledge, this is the first implementation of belief propagation symbol detector. Existing works do not consider hardware implementation issues.
- The use of layer processing here is different than conventional LDPC decoding due to the fact that the channel matrix is Tœplitz and contains floating point numbers (as opposed to LDPC decoding where the parity check matrix is binary, and not Tœplitz). We modify the BP algorithm

in a novel way that exploits this structure. The architecture is different from LDPC decoder because the channel matrix is time varying and the actual computation is different. Also, most of the LDPC work are focused on binary and this detector is focused on QPSK (or non-binary). The computation is so complex such that we have to use serialized architecture to make the area/power under the budget.

  - To balance the tradeoff between the hardware efficiency and error performance, we study the choosing of the implementation parameters, e.g. the length of the prefilter, the number of nonzero taps, the number of iterations, by simulations.

- Performance study of the PRBP detector over frequency-selective fading channels.

  We study the performance of the aforementioned PRBP detector under a more generic 3G channel rather than the sparse channel. Another suboptimal partial response maximum-likelihood (PRML) detector, which has been shown to be effective in high-density magnetic recording applications, is considered for comparison. Similar to the PRBP detector, the PRML detector also employs a hybrid two-stage scheme, in order to allow a tradeoff between performance and complexity. The first stage of the PRML detector is a linear filter which transforms the original channel to a very short channel. The residual ISI is then cancelled in the second stage using the MLSE. In simulations, we consider a slow fading environment and use the ITU-R 3G channel models. From the numerical results, it is shown that in frequency-selective fading wireless channels, the PRBP detector provides superior performance over both the traditional mini-

mum mean squared error linear equalizer (MMSE-LE) and the PRML detector. We emphasize our contributions in the following aspects:

  - The performance of the PRBP detector in realistic wireless channels is illustrated for the first time, while in previous works, the performance simulation is limited to either a simple 3-tap/5-tap equipower slow fading channel or underwater acoustic channels.
  - In order to provide reference for the PRBP detector design for 3G wireless applications, we also discuss several implementation issues on the PRBP detector.
  - Through simulations, we show that the PRML detector is not as useful as it is in high density magnetic recording applications, since the spectrum of the TIR does not always well match that of the CIR, which introduces increased colored noise for the MLSE at the second stage.

- Hybrid DFSF-BP detector for ATSC DTV receivers.

  We extend our work to accommodate the application of ATSC DTV systems. In order to reduce error propagation caused by the traditional decision feedback equalizer (DFE) in DTV receiver, we present a hybrid BP based detector, consisting of an adaptive decision feedback sparsening filter (DFSF) as the partial response equalizer, and the BP detector. In the first stage, the DFSF employs a modified feedback filter which leaves the strongest post-cursor ISI taps uncorrected. As a result, a long ISI channel is equalized to a sparse channel having only a small number of nonzero taps. In the second stage, the BP detector is applied to mitigate the residual ISI.

  We emphasize our contributions in the following aspects:

- Our idea is conceived from the scheme in [9], though the major difference from previous work is that our scheme targets 8-VSB-based ATSC DTV systems and is suitable for channels that experience Doppler fading; [9] only considers static channels and BPSK modulation.
- Besides the multipath fading, Doppler fading is another distortion effects of the ATSC channel. To tackle the Doppler fading, the DFSF is adapted using the LMS algorithm, such that the amplitude and the locations of the nonzero taps of the equalized sparse channel appear to be fixed. As such, the channel appears to be static during the second stage of equalization which consists of the BP detector.
- We assess the error performance of the proposed scheme by simulation under static/dynamic ATSC DTV channel models. Simulation results show that the proposed detector outperforms the traditional DFE in both static and dynamic environments.

- Symbol detector design for cooperative communications.

  Cooperative communications have attracted a lot of attention recently for its ability to exploit increased spatial diversity available at distributed antennas on other nodes. However, limited research has yet been conducted into the implementation issues of relaying and cooperation. We investigate design and implementation of optimal and sub-optimal detectors for half-duplex relays in frequency selective fading channels encountered in practice.

  We emphasize our contributions in the following aspects:

  - A system framework employing non-orthogonal amplify-and-forward half-duplex relays through ISI channels is developed.

– Based on the system model, we design and implement an optimal MLSE detector based on the Viterbi algorithm.

– In order to achieve a balance between the computational complexity and performance, several sub-optimal detectors are proposed. We first present a multitrellis Viterbi algorithm (MVA) based detector which decomposes the original trellis into multiple parallel irregular sub-trellises by investigating the dependencies between the received symbols.

– The decision feedback sequence estimation (DFSE) based detector and BP detector are proposed for cooperative ISI channels. Traditionally these two detectors are used with fixed, static channels. In our model, however, the effective channel is periodically time-varying, even when the component channels themselves are static. Consequently, we modify these two detector to account for cooperative ISI channels.

– Through simulations in frequency selective fading channels, we demonstrate the uncoded performance of the DFSE detector and the BP detector when compared to the optimal MLSE detector.

– In addition to quantifying the performance of these detectors, we also include an analysis of the implementation complexity as well as a discussion on complexity/performance tradeoffs.

1.3 Outline

This dissertation is organized as follows.

Chapter 2 presents the design and implementation of the PRBP detector for sparse channels. We first introduce related literatures on sparse channel symbol detectors,

and give the communication system model. The algorithms for the PRE and BP detector are described, and then the overall architecture and individual blocks are discussed in details. Finally we show the simulation performance and implementation results.

Chapter 3 compares the performance of the PRBP detector with the PRML detector under ITU-R 3G channel models. The PRML detector is briefly introduced first. Numerical results are given afterwards. The performance of the traditional MMSE-LE is also given as the benchmark. We conclude with discussions on several implementation issues.

Chapter 4 presents the DFSF-BP detector for ATSC DTV receivers. We first introduce the background of the ATSC DTV systems, and present the the system model for time-varying channels. The algorithm of DFSF-BP detector is given, and the performance of the proposed detector is evaluated in both static and dynamic ATSC channels.

Chapter 5 shows several symbol detectors for cooperative communications in ISI channels. The background of cooperative communications is introduced. After developing a system model for the case of AF relays in ISI channels, we first present the optimal MLSE detector. Next we propose the MVA detector, the DFSE detector, and the BP detector to deal with the periodically time-varying and sparse cooperative ISI channels. At last, we demonstrate the uncoded performance of the DFSE detector and the BP detector in frequency selective fading channels, and we also include an analysis of the implementation complexity.

Chapter 6 draws the conclusions and discusses open issues.

Chapter 2

Belief Propagation Based Detector for Sparse Channels

In this chapter, we present the design and implementation of a partial response belief propagation (PRBP) detector for communication channels which have long spanning durations but sparse multipath structure. The architecture of the PRBP detector is implemented by cascading a partial response equalizer (an adaptive LE) with a BP detector, which enables the detector to achieve a near-optimal error rate performance with acceptable implementation complexity. The chapter is organized as follows. Related work on detectors for sparse channels is introduced in Section 2.1. In Section 2.2, we gives the communication system model in consideration. The design of the partial response equalizer is described in Section 2.3.1. The factor graph representation and BP algorithm are described in Section 2.3.2 and the implementation of the symbol detector is given in Section 2.4. We show the simulation performance and implementation results in Section 2.5 and Section 2.6, followed by conclusions in Section 2.7.

2.1 Introduction

Sparse channels are characterized as having long multipath delay spreads but with few nonzero coefficients. Such channels arise in a number of modern wireless communication applications. For instance, in underwater acoustic communications, the intersymbol interference (ISI) can span several hundreds of symbols but the multipath structure is usually very sparse [4]. In hilly terrain broadband wireless communications [10] and high-definition television (HDTV) [3], the delay profile also exhibits a sparse channel impulse response (CIR).

In these applications, ISI is one of the major impairments to reliable symbol detection. The optimal maximum-likelihood (ML) detector based on the Viterbi algorithm (VA) may be used to estimate the transmitted sequence in the presence of the ISI [7]. However, the computational complexity of the ML detector increases exponentially with the delay spread of the channel. Thus, the optimal ML detector is impractical for sparse channels whose delay spread is usually very long. Near-optimal detectors for sparse channels have been investigated by several researchers. The parallel trellis Viterbi algorithm (PTVA) [11] reformulated the original single trellis into a set of independent trellises which could operate in parallel and have much less complexity. The PTVA requires that the sparse channel have equi-spaced coefficients, however, which often cannot be satisfied in practice. Although a generalized PTVA is proposed in [11] to address general sparse channels, optimal performance can be guaranteed only if the CIR is well matched to an equi-spaced structure. A near-optimal detector based on a multitrellis Viterbi algorithm (MVA) [12] was proposed which decomposes the trellis into multiple irregular sub-trellises by investigating the dependencies between the received symbols. It was shown that the complexity does

not depend on the channel length but only on the number of nonzero coefficients. However, the trellis decomposition is not straightforward for a general sparse channel. Decision feedback sequence estimation (DFSE) [13, 14], which is a popular scheme for long ISI channels, can also be an alternative for sparse channels. However, it only yields high performance if the CIR is minimum-phase. The whole system could be unstable, if a prefilter is used in front of the DFSE to transform the CIR to its minimum-phase equivalent. In a recent work, iterative belief propagation (BP) detectors have been proposed for ISI channels [8, 15, 16]. It has been shown in [15] that, for an uncoded system, the BP detector achieves near-optimal performance over frequency selective ISI channels. Furthermore, the BP detector has a complexity which only depends on the number of nonzero channel coefficients, and thus is very suited for sparse channels.

The BP algorithm, also known as the message passing algorithm or sum-product algorithm, has been widely used for iterative decoding of low-density parity-check (LDPC) codes [17], and its implementation has been well studied in [18–22]. However, the implementation of a BP detector for ISI channels is remarkably different from the LDPC decoder in the following aspects. Firstly, LDPC codes are usually designed to have a structure that facilitates node processing in parallel and reduces the complexity of message passing during the iterative decoding process; an example of such structured codes is quasi-cyclic (QC) LDPC codes. In the case of LDPC decoding, the structured parity check matrix consists of purely binary values and it is effectively a constant matrix. For BP-based symbol detection in ISI channels, the analogous Tœplitz channel matrix is not constrained to the binary numbers, is not a fixed constant, and can even be time-varying. Consequently, it is difficult to exploit parallelism in the BP detector, and one must resort to serial processing. Secondly,

the calculation of messages passing between nodes is significantly more complex for symbol detection than for LDPC decoding. In fact, the complexity of the BP detector increases exponentially with the number of nonzero channel coefficients which makes it impractical to implement BP detectors even for sparse channels where the number of nonzero coefficients is on the order of 10. One solution to permit the use of near-optimal BP detectors is to first pass the received signal through a partial response equalizer (PRE) [8] which equalizes the original sparse channel to an even more sparse channel with fewer nonzero coefficients. This permits the design effort of the BP detector to be greatly mitigated while maintaining near-optimal performance [8]. Thirdly, the channel matrix for the BP detector is Tœplitz and therefore has a banded structure which simplifies the wire routing for the message passing.

By investigating the aforementioned differences, we present an implementation of an efficient symbol detector for sparse channels as in Fig. 2.1. The symbol detector consists of two blocks: the PRE in the form of an adaptive LE, and the BP detector. In the PRE, the channel is first partially equalized to a target impulse response (TIR) which is designed to have only a small, fixed number of nonzero coefficients. The coefficients of PRE and TIR are obtained based on the minimum mean squared error (MMSE) criterion, which is designed adaptively using the Least Mean Square (LMS) algorithm. Once the coefficients of the PRE and TIR are computed, the partial equalization is carried out by a finite impulse response (FIR) filter, and the residual ISI is eliminated by the BP detector.

The architecture of the BP detector is based on a pipelined layer processing scheme, which allows high throughput with low complexity. The BP detector is also designed to be reconfigurable so that it can adapt to time-varying ISI channels. Since the complexity of the BP detector merely scales with the number of nonzero

channel coefficients, we limit the nonzero coefficients of the TIR to keep a reasonable complexity of the BP block. Note that a system designer can specify the TIR to be sparse even when the original CIR is not sparse; as pointed out in [8], this implies that the symbol detector is general enough to work even for non-sparse channels. We implement the proposed symbol detector for quadrature phase-shift keying (QPSK) modulation. We consider a sparse channel with 30 coefficients, a PRE with 100 coefficients, and a TIR designed to have 3 nonzero coefficients. The BP detector operates on 1024 signal samples for 5 iterations per frame. The target frequency for synthesis is set as 515 MHz, for an equivalent throughput of 206 Mb/s. The synthesized result shows an estimated area of 3.162 mm^2 in TSMC 90-nm technology, and a total dynamic power of 1.096 W.

2.2 System Model

We first introduce some notations for the description of the communication system. Assume that a length N sequence of M-ary symbols $x[k] \in \{a_0, a_1, ..., a_{M-1}\}$ is transmitted through a complex ISI channel whose equivalent discrete-time CIR [23] is described as $\mathbf{h} = [h[0], h[1], ..., h[L_h - 1]]^T$, where L_h is the length of the CIR. Letting D_h be the number of nonzero coefficients of $\mathbf{h}$, then $D_h \ll L_h$ for sparse channels. The received signal sample at time instant k can be expressed as

$$y[k] = \sum_{i=0}^{L_h-1} h[i]x[k-i] + w[k],$$

where $w[k]$ is complex additive white Gaussian noise (AWGN) with noise power σ^2.

At the receiver, the symbol detector consists of two cascaded components: an

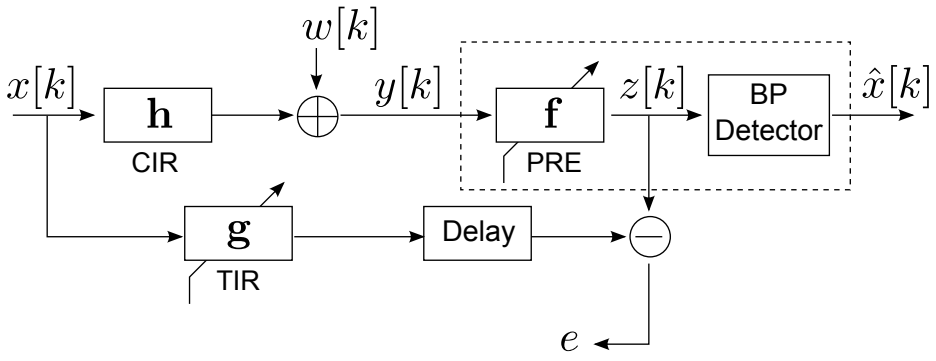

Figure 2.1: System model of the partial response belief propagation detector.

adaptive PRE denoted by $\mathbf{f}$, and a BP detector. The system model is shown in Fig. 2.1. The goal of the adaptive PRE is to equalize the original CIR $\mathbf{h}$ to a TIR $\mathbf{g}$ which has fewer nonzero coefficients, thus making second stage detection via belief propagation computationally efficient. The adaptive PRE and BP detector are discussed in detail in the next section.

2.3 Design of the Symbol Detector

2.3.1 Partial Response Equalizer

The PRE equalizes the channel to a sparse TIR, and is designed by jointly finding the set of PRE and TIR coefficients which minimize the mean squared error (MSE). The PRE is implemented as a finite impulse response (FIR) filter whose coefficients are $\mathbf{f} = [f[0], f[1], ..., f[L_f - 1]]^T$ where L_f is the chosen length of the PRE. Conventional wisdom for classical equalizer design [7] dictates that the equalizer length should be approximately 5 times the channel length L_h. Here, however, the PRE has more degrees of freedom since it does not need to equalize the channel to a single spike. As we will show empirically in Section 2.5, an equalizer length of about 3 times the

channel length (i.e. choosing $L_f \approx 3L_h$) seems sufficient in practice.

We denote the TIR by $\mathbf{g} = [g[0], g[1], ..., g[L_g - 1]]^T$ where L_g is the length of the TIR, and denote the nonzero version of the TIR by $\mathbf{g}' = [g[l_1], g[l_2], \ldots, g[l_{D_g}]]^T$ where D_g is the number of nonzero coefficients. Choosing D_g in the sparse TIR $\mathbf{g}$ is a tradeoff between error-rate performance and detector complexity. When D_g is smaller, the complexity of the BP detector is reduced but the overall error-rate performance is degraded. A larger value of D_g adds complexity to the BP detector but also leads to better error-rate performance. In general, the nonzero coefficients of $\mathbf{g}$ are chosen so that $D_g \leq D_h \ll L_h$. We note that the case of $D_g = 1$ corresponds to classical equalization (in which case the BP detector can be reduced to a memoryless slicer), while the choice of $D_g = D_h$ corresponds to a full-blown BP detector. The locations of these nonzero coefficients in the TIR must be also placed properly. There are L_g-choose-D_g possible sets of locations, and several approaches have been proposed [24,25] for choosing the locations to optimize various criteria. For simplicity we adopt the approach in [8] which chooses the locations of the nonzero coefficients of the TIR to coincide with the dominant arrivals in the CIR. We note that this approach assumes that the receiver has some knowledge of the channel $\mathbf{h}$, though it need not know the CIR precisely.

The nonzero coefficients in the TIR as well as the PRE coefficients are then calculated jointly using the MMSE criterion [26], which causes the combined response of the CIR and PRE to approximate the TIR. Direct computation of the MMSE solution of the PRE and TIR involves solving a generalized eigen-decomposition and requires full knowledge of the channel $\mathbf{h}$. In order to make the implementation feasible, the PRE can be designed adaptively. In [8], the recursive least squares (RLS) algorithm was employed. Here, we follow the approach of [26] by employing two least

Algorithm 2.1 LMS algorithm for PRE design .

Parameters:
TIR at time n: $\mathbf{g}_n = [g_n[0], g_n[1], \ldots, g_n[L_g - 1]]^T$
Indices of nonzero coefficients of the TIR: $l_1, l_2, ...l_{D_g}$
Nonzero version of the TIR at time n:
$\mathbf{g}'_n = [g_n[l_1], g_n[l_2], \ldots, g_n[l_{D_g}]]^T$:
PRE at time n: $\mathbf{f}_n = [f_n[0], f_n[1], \ldots, f_n[L_f - 1]]^T$
Step size for TIR updating: μ_g
Step size for PRE updating: μ_f
Received symbols vector at time n:
$\mathbf{y}(n) = [y[n], y[n-1], ..., y[n - L_f + 1]]^T$
Transmitted symbols vector at time n:
$\mathbf{x}(n) = [x[n - l_1], x[n - l_2], ..., x[n - l_{D_g}]]^T$

Initialization:
Set the middle element of $\mathbf{f}_0$ to 1, and set the first element of $\mathbf{g}'_0$ to 1.

Computation:
For $n = 0, 1, ...,$ compute

1. $e_n = {\mathbf{f}_n}^T \mathbf{y}(n) - {\mathbf{g}'_n}^T \mathbf{x}(n)$
2. $\mathbf{f}_{n+1} = \mathbf{f}_n - \mu_f \mathbf{y}(n)^H e_n$
3. $\mathbf{g}'_{n+1} = \mathbf{g}'_n + \mu_g \mathbf{x}(n)^H e_n$
4. Normalize $\mathbf{g}'_{n+1}$

mean squares (LMS) algorithms in tandem to jointly calculate the PRE and TIR. While this algorithm does not require knowledge of $\mathbf{h}$, it does require the availability of known training data. The dual LMS algorithm is summarized in Algorithm 2.1. Since only the nonzero coefficients are updated in each iteration, the TIR can be represented by the nonzero values in $\mathbf{g}'$ and their locations $l_1, l_2, ..., l_{D_g}$. After the convergence of the LMS algorithm, the PRE output signal at time n is $z[n] = \mathbf{f}^T \mathbf{y}(n)$ which in the absence of noise is ideally equal to $\mathbf{g}'^T \mathbf{x}(n)$.

$$R_{m\to n}(i) = \log \frac{\sum_{\forall \mathbf{x}(m):x[n]=a_i} \exp\left\{ \frac{-\left|z[m]-\mathbf{g}'^T\mathbf{x}(m)\right|^2}{2\sigma^2} + \sum_{j\in\mathcal{N}(m)\backslash n} Q_{j\to m}(\psi^{-1}(x[j])) \right\}}{\sum_{\forall \mathbf{x}(m):x[n]=a_0} \exp\left\{ \frac{-|z[m]-\mathbf{g}'^T\mathbf{x}(m)|^2}{2\sigma^2} + \sum_{j\in\mathcal{N}(m)\backslash n} Q_{j\to m}(\psi^{-1}(x[j])) \right\}} \tag{2.1}$$

$$\begin{aligned} &\approx \max_{\forall \mathbf{x}(m):x[n]=a_i} \left\{ \frac{-\left|z[m]-\mathbf{g}'^T\mathbf{x}(m)\right|^2}{2\sigma^2} + \sum_{j\in\mathcal{N}(m)\backslash n} Q_{j\to m}(\psi^{-1}(x[j])) \right\} \\ &\quad - \max_{\forall \mathbf{x}(m):x[n]=a_0} \left\{ \frac{-\left|z[m]-\mathbf{g}'^T\mathbf{x}(m)\right|^2}{2\sigma^2} + \sum_{j\in\mathcal{N}(m)\backslash n} Q_{j\to m}(\psi^{-1}(x[j])) \right\}, \\ &\quad i = 0, 1, ..., M-1. \end{aligned} \tag{2.2}$$

2.3.2 Belief Propagation Detection Algorithm

The residual ISI of PRE output is further mitigated by a near-optimal BP detector, based on the factor graph [17] which represents the input-output relationship of an ISI channel. Fig. 2.2 shows the factor graph of a simple example with channel impulse response of $[h[0], 0, h[2]]^T$. In the factor graph, channel input symbols $x[0], x[1], ...$ are represented by the circles on the top (known as variable nodes or bit nodes), and the channel output signals $y[0], y[1], ...$ are denoted by the squares at the bottom (known as check nodes or function nodes). The connections (edges) between variable nodes and check nodes represent dependencies between input and output. For example, $y[3]$ is connected to $x[1]$ and $x[3]$, since $y[3] = h[0]x[3] + h[2]x[1] + w[3]$ in the example.

The BP algorithm is also referred to as a two-phase message passing or sum-product algorithm, in which check-to-variable (CTV) and variable-to-check (VTC) messages are transmitted along the edges to update each other iteratively. During the first phase, the CTV message is computed at the check nodes based on the known

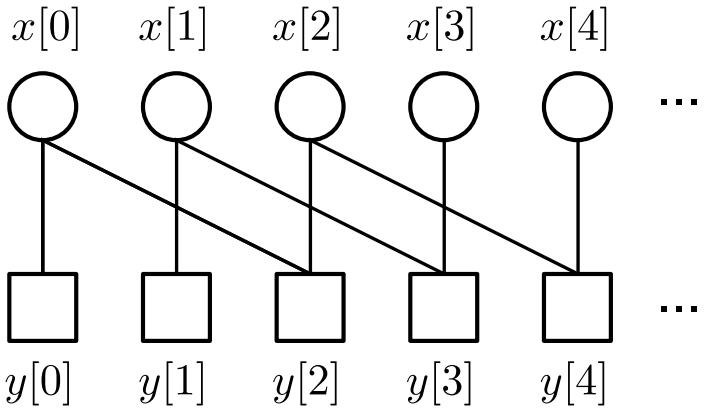

Figure 2.2: Factor graph of an example channel $[h[0], 0, h[2]]^T$.

channel coefficients and the *a priori* probability information from the variable nodes. The updated message is then passed from each check node to its connected variable nodes. During the second phase, the variable nodes update their *a priori* information and send it back to their connected check nodes. The same procedure is repeated iteratively. After several iterations, the variable nodes are accumulated with sufficient likelihood information and a hard decision can be made for each input symbol. The operations of the BP algorithm in log-likelihood ratio (LLR) domain are summarized as follows. Note that since the PRE has been applied, we assume that the sparse channel is the TIR $\mathbf{g}$ (whose nonzero version is $\mathbf{g}'$), and channel input and output at time n are $x[n]$ and $z[n]$, respectively.

2.3.2.1 Calculating CTV messages

The extrinsic message from check node m to variable node n is computed as (2.1) if the two nodes are connected, where $\mathbf{x}(m) = [x[m - l_1], x[m - l_2], \ldots, x[m - l_{D_g}]]^T$ is the data vector and $Q_{j\to m}$ is the *a priori* information from variable node j for check node m, and $\mathcal{N}(m)$ is the set of variable nodes connected with check node m. We use the mapping function $\psi : \{0, 1, ...M - 1\} \to \{a_0, a_1, ..., a_{M-1}\}$ to denote

the modulation on the M-ary constellation, and the demapping function is expressed as $\psi^{-1} : \{a_0, a_1, ..., a_{M-1}\} \rightarrow \{0, 1, ...M-1\}$. The nonlinear log-sum-exponential function in (2.1) could be implemented with look-up tables (LUTs), but it is not desirable because the LUTs would require a large size memory. As a good approximation, the calculation in (2.1) can be simplified as (2.2) by Jacobian logarithm with negligible performance loss [27]. Note that for M-ary modulation, there are $M-1$ LLRs $R_{m\rightarrow n}(1), R_{m\rightarrow n}(2), ...R_{m\rightarrow n}(M-1)$ need to be calculated from check node m to variable node n, and $R_{m\rightarrow n}(0) = 0$ by (2.2).

2.3.2.2 Calculating VTC messages

After receiving the updated extrinsic message from the check nodes, the variable nodes update the *a priori* information for the next iteration. The *a priori* information at variable node n for check node m is calculated as

$$Q_{n\rightarrow m}(i) = \sum_{j\in\mathcal{M}(n)\backslash m} R_{j\rightarrow n}(i), i = 0, 1, ..., M-1, \tag{2.3}$$

where $\mathcal{M}(n)$ is the set of check nodes connected with variable node n. Note that for M-ary modulation, there are $M-1$ LLRs $Q_{n\rightarrow m}(1), Q_{n\rightarrow m}(2), ..., Q_{n\rightarrow m}(M-1)$ need to be obtained from variable node n to check node m, and $Q_{n\rightarrow m}(0) = 0$ by (2.1) and (2.3). Then we go back to the first step for the next iteration.

2.3.2.3 Summing up and decision

After iterating the above two steps for several times, the accumulated LLRs for variable node n are

$$\Lambda_n(i) = \sum_{j \in \mathcal{M}(n)} R_{j \to n}(i), i = 0, 1, ..., M-1. \tag{2.4}$$

By comparing these LLR sums, an estimate of the transmitted symbol sequence $x[n]$ can be made by

$$\hat{x}[n] = \psi(\arg\max_i \Lambda_n[i]), i = 0, 1, ..., M-1.$$

2.4 Architecture of the Symbol Detector

2.4.1 Overall Architecture of the PRBP Detector

In this section, we present a low-complexity high-throughput architecture of the proposed symbol detector. The overall system architecture of the proposed sparse channel detector is shown in Fig. 2.3. The LMS block runs the adaptive algorithm to obtain the PRE coefficients $\mathbf{f}$ and TIR coefficients $\mathbf{g}$. Since the BP algorithm processes symbol detection frame by frame, a serial-to-parallel (S/P) converter is placed before the BP block to buffer N symbols where N is the frame length.

We further assume that the coherence time of the channel is sufficiently large so that the channel can be considered static during each frame of N symbol periods. Consequently, the LMS algorithm only needs to be carried out once per frame. We use a digital signal processor (DSP) to process the LMS update calculation for several reasons. Since the PRE length L_f can be very long, an ASIC implementation would consume excessive circuit area. A DSP is also more accurate for recursive algorithms if floating-point is used. In addition, the LMS adaptation is only performed once per block of training sequence and a regular DSP or embedded processor is sufficient to handle the computation. Thus, a DSP provides an efficient solution for LMS block

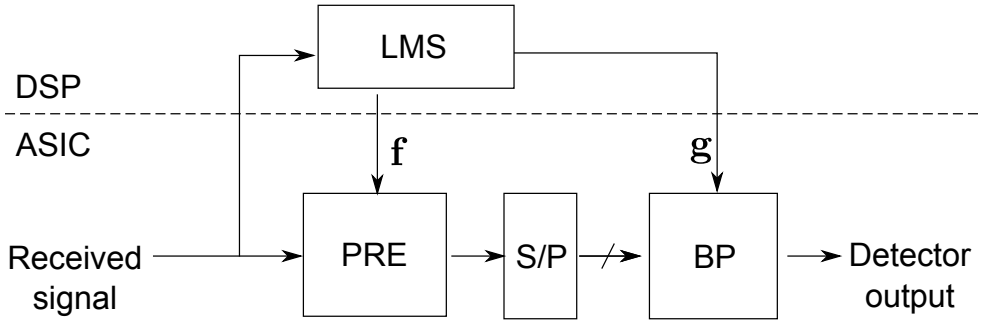

Figure 2.3: Architecture of the PRBP symbol detector.

as in Figure 3.

For high data rate application, we suggest the implementation of both PRE and BP algorithms in an FPGA or a customized ASIC. Since the length of the PRE is usually large, a circuit implementation can exploit pipelining and parallelism for the large filter. As the BP detection algorithm is very complex, a customized circuit implementation is the best solution to achieve high throughput. The architectures of the PRE and BP blocks are discussed in the following subsections.

2.4.2 PRE as a Folded FIR Filter

The folded FIR filter architecture is selected as an efficient implementation of the PRE block. Since the BP block is the most complex and limits the overall detector throughput, the PRE is designed such that the filter output matches the input rate requirement of the BP block. For each frame, the BP detector takes N samples and processes for n_{it} iterations, before outputs the detected symbols. Considering the pipelined layer processing architecture, which will be discussed in Section 2.4.3, applied to the BP detector, it takes $N \times n_{it}$ cycles to process one frame of received symbols. The function of the S/P block is to buffer the next frame while the current frame is being processed, thus the PRE block only needs to output one sample per

n_{it} clock cycles. For area efficiency, we implement a partly serial FIR filter using the folding technique [28]. The folding factor of the FIR filter is n_{it}, that is, one multiplier-accumulator (MAC) is allocated for n_{it} filter coefficients. Hence, the number of MACs used in the PRE is $\lceil L_f/n_{it} \rceil$. The block diagram of the folded filter design is given in Fig. 2.4.

Given the symbol detector running at frequency f and the number of BP iterations n_{it}, we can estimate a throughput of f/n_{it} symbol per second. For M-ary modulation, the bit rate is

$$\text{Throughput}(bps) = (f/n_{it}) \log_2 M \tag{2.5}$$

2.4.3 Pipelined Architecture for Layer Processing

Prior to the discussion of the hardware architecture for the BP detector, we first give the Tœplitz channel matrix which shows dependency between the input and output. As an example, Fig. 2.5 show the matrix for a channel with only three nonzero coefficients. The channel matrix is similar to parity-check matrix in quasi-cyclic LDPC codes. Each check node corresponds to a column index, and its related variable nodes corresponds to the row indices. The architecture design for a BP detector can be referenced to the existing LDPC decoder design [18–22]. Here we adopt a sequential BP algorithm, which is performed in a way that each check node and its connected variable nodes are treated as a layer. Within each layer processing, we update the LLRs from the check node (e.g. check node m) to its neighboring variable nodes $\mathcal{N}(m)$, and then update the accumulated LLRs associated to these variable nodes. These updated messages are then used in the next layer processing. As an illustration, the sequential BP processing is shown in Fig. 2.6. Previous studies [29] also showed that layered BP algorithm improves the convergence speed by reducing

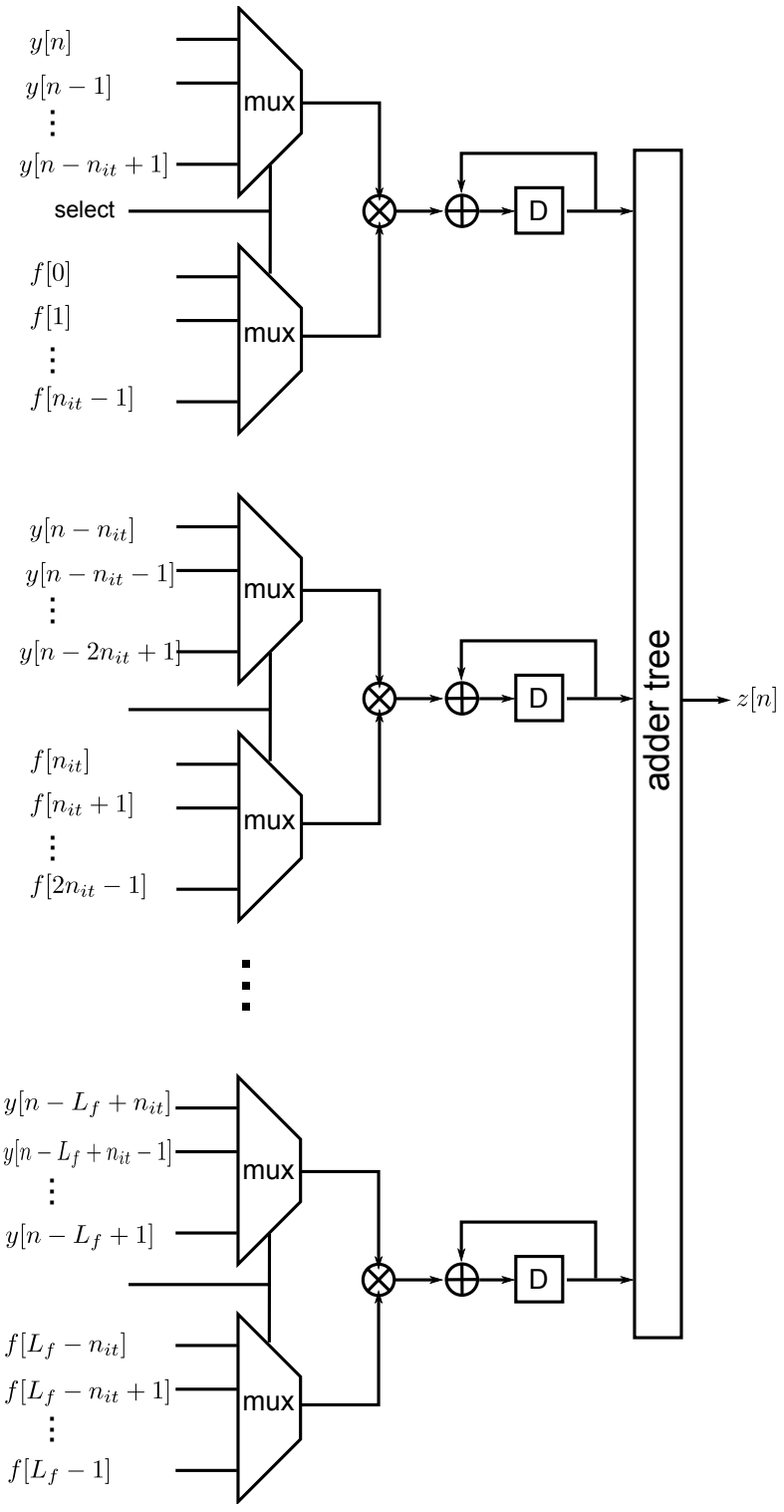

Figure 2.4: Block diagram of the PRE folded filter design.

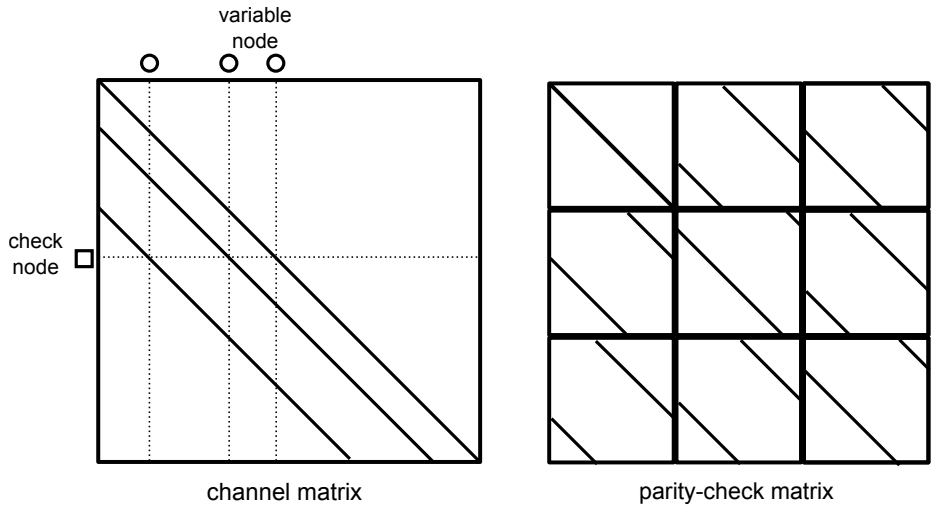

Figure 2.5: Tœplitz channel matrix vs. parity-check matrix.

the number of iterations.

However, there are major differences between the BP detection algorithm and LDPC decoding algorithm. The structure of detector channel matrix is time-varying, while LDPC parity-check matrix is usually fixed. In LDPC decoders, the structured property of the parity-check matrix can be decomposed into cyclic permutation submatrices, which allows for tradeoff between hardware complexity and decoding throughput using the partially parallel architecture [20, 22]. In contrast, the parallelism is difficult to exploit for the BP detector since the channel matrices are different frame by frame. If multiple layers are processed simultaneously, it is possible that one variable node will receive the updated messages from different check nodes, then the memory writing conflict will occur.

Another major difference between the BP detection algorithm and LDPC decoding algorithm is in the calculation of CTV messages. The calculation for the BP detector operates in the Euclidean space. While for LDPC decoding, it operates in GF(2). Therefore, CTV calculation are much more complicated in a BP

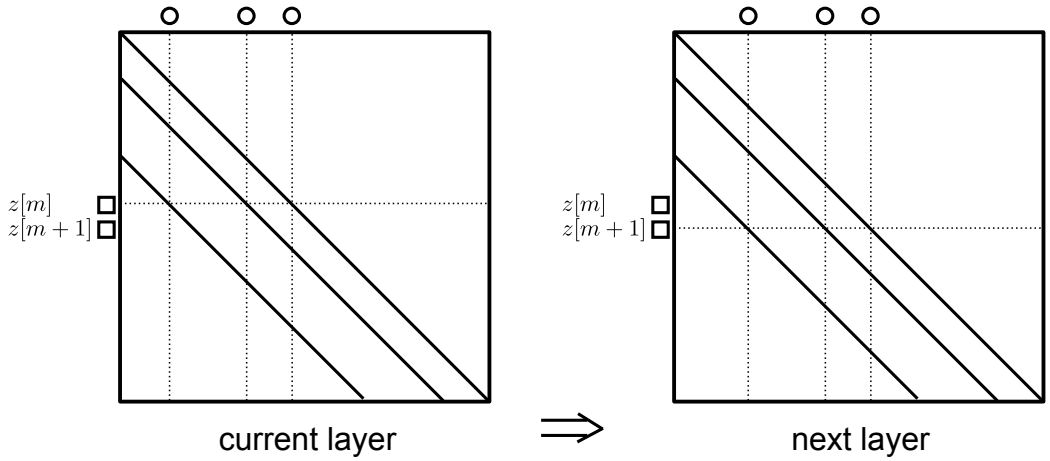

Figure 2.6: An illustration of sequential BP processing.

detector. Recalling that the number of nonzero coefficients is D_g and the modulation order is M, there are D_g elements in $\mathbf{x}(m)$ and hence M^{D_g} instance of $\left\{ \frac{-\left|z[m]-\mathbf{g}'^T\mathbf{x}(m)\right|^2}{2\sigma^2} + \sum_{j\in\mathcal{N}(m)\backslash n} Q_{j\to m}(\psi^{-1}(x[j])) \right\}$ need to be calculated in order to obtain the LLRs passing from a check node to its connected variable nodes. If each instance requires 3 multiplications, there are $3M^{D_g}$ multiplications required for 1 CTV LLR calculation. Given that each layer updates the messages sequentially, computing resource can be reused such that the total number of multipliers is

$$n_{\text{mult}} = 3M^{D_g}. \tag{2.6}$$

In a typical case, for example, $D_g = 3$, and $M = 4$ (QPSK), we need to allocate $3\times 4^3 = 192$ multipliers to calculate the CTV LLRs. For the practical implementation of a BP detector, the parameters M and D_g should be reasonably small. Since the number of multipliers is exponential with respect to the constellation size M, the BP detector approach is typically suited for lower-order QAM systems, such as BPSK and QPSK modulations.

The problem in the sequential BP algorithm is that each layer processing takes a long time to execute due to the computational complexity according to (2.2). The long processing latency for each layer results in a very low throughput rate. The bottleneck can be overcome by pipelining the layer processing unit. For layered processing scheme, the messages on the m-th layer are updated first, and messages on the $(m+1)$-th layer are processed next. The two successive layer processings can be pipelined without problem as there is no dependency. Note that the independence between layers can be easily satisfied for sparse channels since the nonzero coefficients are often widely separated. It is also observed from the channel matrix that the maximum depth of the pipeline is equal to the shortest distance between the locations of any two adjacent nonzero coefficients.

2.4.4 Cache-Memory Architecture for the BP Detector

The overall architecture of the BP detector is shown in Fig. 2.7. The BP detector mainly consists of a layer processing unit (LPU) responsible for layer message update, the LLR sum memory and CTV message memory which are used to store the belief propagation messages, a cache which is a temporary storage for the LLRs of the nodes currently being processed, and a decision unit for estimating the input symbols. Note that control signals on each blocks are omitted for clearness.

The LPU is the core computing unit executing message updating expressed in (2.2), (2.3) and (2.4). We note that the VTC messages Q used in (2.2) can be obtained by LLR sum Λ and CTV messages R, i.e.

$$Q_{n\to m}(i) = \Lambda_n(i) - R_{m\to n}(i), i = 0, 1, ..., M-1$$

according to (2.3) and (2.4). In order to minimize the hardware cost, we only store and update Λ instead of Q since the memory for Q is about D_g times as large as the memory for Λ. In each layer processing, the LPU first calculates the VTC messages Q needed in (2.2), and then update the CTV messages R and LLR sums Λ. The architecture of LPU is given in Fig. 2.8. The calculation of

$$\max\left\{\frac{-\left|z[m]-\mathbf{g}'^{T}\mathbf{x}(m)\right|^{2}}{2\sigma^{2}}+\sum_{j\in\mathcal{N}(m)\backslash n}Q_{j\to m}(\psi^{-1}(x[j]))\right\},\forall\mathbf{x}(m):x[n]=a_{i}$$

is carried out in M square blocks for $i=0,1,...,M-1$ respectively. The architecture for each square block is shown in the dashed box. Since there are M^{D_g-1} different cases of $\mathbf{x}(m)$, M^{D_g-1} branches are implemented for each case. A pipeline architecture is implemented in the LPU design such that it can take a new signal input every clock cycle.

The updated R from LPU are stored back into the CTV message memory, and they will be fetched from the memory in next iteration. There are $(M-1)$ LLRs for each connection, and thus $(M-1)ND_g$ LLRs for the entire factor graph. The size of CTV message memory is $(M-1)ND_g \cdot q_R$ where q_R is the word length of the CTV LLR.

The cache is a register bank only containing the accumulated LLRs of L_h variable nodes being processed. This is similar to a "sliding window" that only needs to fetch one new LLR from the Λ memory when LPU moves to process the next layer. The updated Λ from LPU consists of D_g sets of LLRs corresponding to D_g connected variable nodes, which is illustrated in Fig 2.6. The updated Λ may be used for the following layer processing in a short time. For efficient memory operations, they are written back to the cache to replace the old values. Only the Λ related to the leftmost

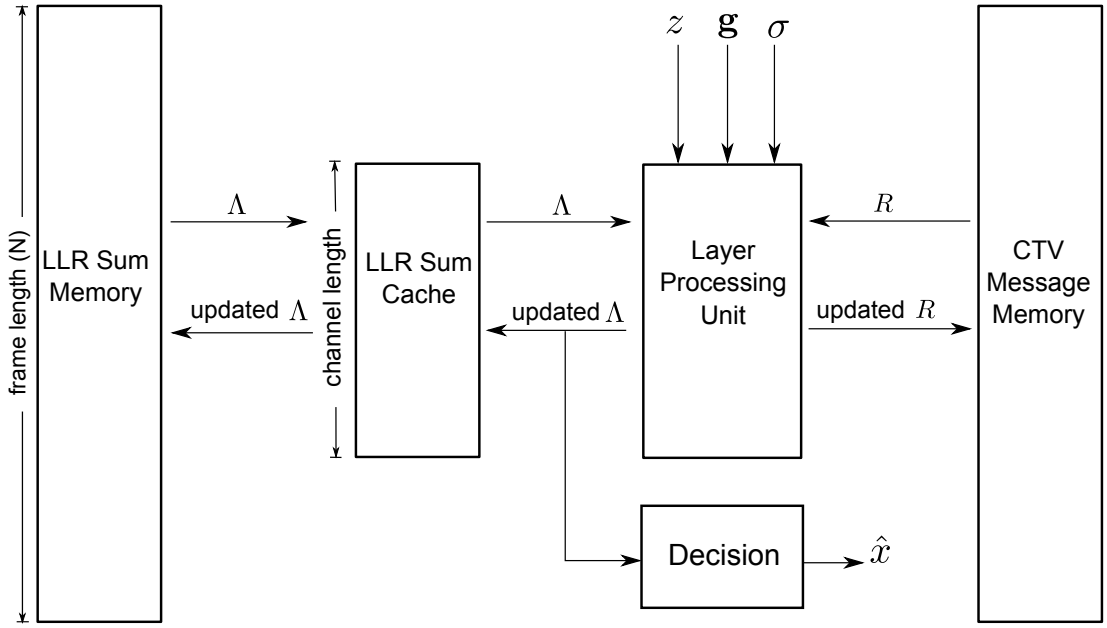

Figure 2.7: Overall architecture of the BP detector.

variable node will be used in the next iteration, and they are stored back to the Λ memory.

The LLR sum memory is used to store the content of Λ. There are $(M-1)$ LLRs for each variable nodes. Thus the size of the LLR sum memory is $(M-1)N \cdot q_\Lambda$ where q_Λ is the word length of the LLR sum.

Particularly if the number of iterations is reached, the updated Λ from the LPU do not need to store back to the cache or memory. They will be delivered directly to the decision block for symbol decision and output.

2.4.5 Interconnect Network

The interconnect network between the memory units and the LPU is determined by the channel matrix. Since the structure of channel matrix becomes trivial after the PRE, memory access control and interconnection network in the BP detector are

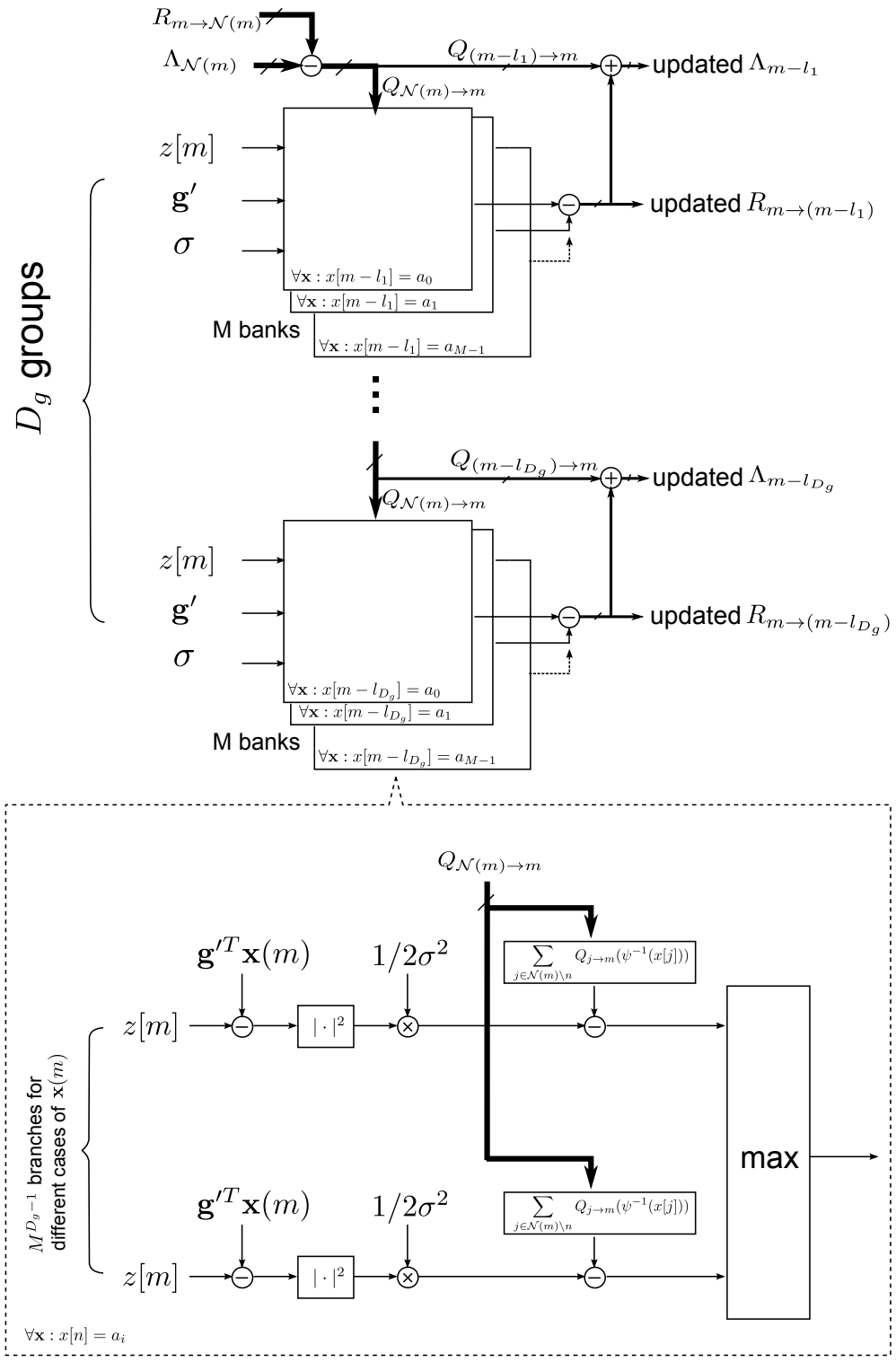

Figure 2.8: Detail architecture of the layer processing unit.

much simpler than LDPC decoder. Given the TIR $\mathbf{g}' = [g[l_1], g[l_2], \ldots, g[l_{D_g}]]^T$, the m-th layer is associated with the $(m - l_1)$-th, $(m - l_2)$-th, $\ldots$, and $(m - l_{D_g})$-th variable nodes, i.e. $\mathcal{N}(m) = \{m - l_1, m - l_2, \ldots, m - l_{D_g}\}$. When the sample $z[m]$ is in process in the LPU, $R_{m\to(m-l_1)}, R_{m\to(m-l_2)}, \ldots, R_{m\to(m-l_{D_g})}$ are loaded from the CTV memory, and $\Lambda_{m-l_1}, \Lambda_{m-l_2}, \ldots, \Lambda_{m-l_{D_g}}$ are loaded from the LLR sum cache. For the next layer process, the read/write addresses for the memories are updated automatically by an increment of 1.

In practice, the sparse channel $\mathbf{h}$ may vary from time to time, and so do its corresponding TIR $\mathbf{g}$ in both values and locations of coefficients. The BP detector can easily adapt to time-varying channels just by changing the coefficients in LPUs and write/read addresses of the memories accordingly.

2.5 Simulation Results and Error Performance

The performance of the proposed symbol detector is evaluated by simulations in terms of symbol error rate (SER) versus signal-to-noise ratio per bit E_b/N_0. In particular, we consider a sparse channel with $L_h = 30$ taps, which is shown in Fig. 2.9. The effect of the transmitter pulse shaping and the receiver matched filter has been included in the channel model. QPSK modulation is selected, so that $M = 4$, and the transmitted symbol has unit power. Each time the BP detector processes one frame with 1024 symbols.

Fig. 2.10 shows the performance for different length of PRE L_f, where $n_{it} = 5$ iterations of message passing are applied and the number of nonzero coefficients in the TIR is $D_g = 3$. As expected, the performance improves with longer L_f, though the performance improvement is minimal once the length reaches $L_f = 3L_h$. Recall, the

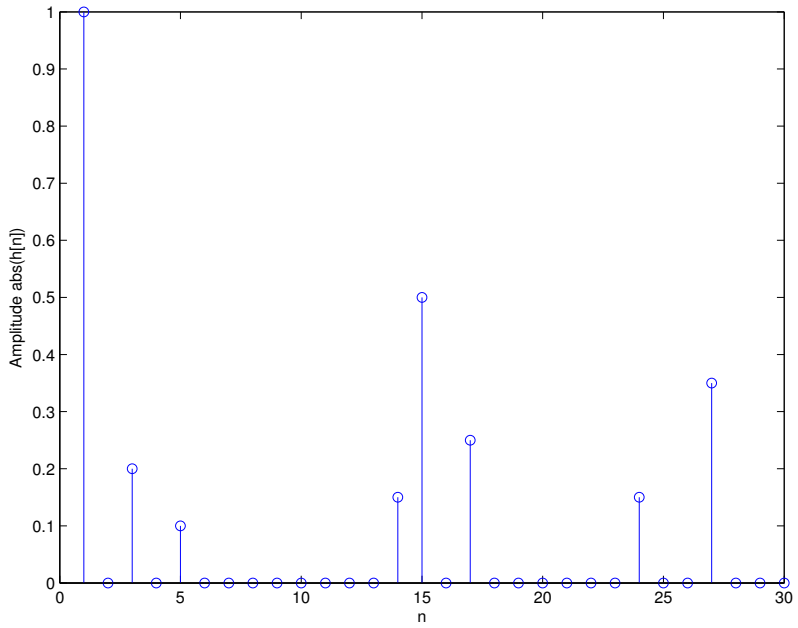

Figure 2.9: Sparse channel coefficients for the experimental setup.

complexity of LMS and PRE will also increase with L_f. In our implementation, the length of the PRE is chosen as $L_f = 100 \approx 3L_h$, and the number of MACs consumed by the PRE is $\lceil L_f/n_{it} \rceil = 20$. Considering the complex filter for QPSK modulation, the number of multipliers is $20 \times 4 = 80$.

Fig. 2.11 shows the performance for different number of nonzero coefficients in TIR, where $n_{it} = 5$ iterations of message passing are applied. To evaluate the effectiveness of the symbol detector, we provide the performance of full BP detector with $D_g = 8$ and the linear equalizer with memoryless slicer for comparison. At a SER of 10^{-5}, the detector exhibits performance 5.5 dB better than the linear equalizer when $D_g = 8$, 4 dB better when $D_g = 4$, around 3.5 dB better when $D_g = 3$, and 1 dB better when $D_g = 2$. As discussed in Section 2.4.3, the complexity of BP detector increases exponentially with D_g. By (2.6) the multiplier utilization is 48, 192, 768, and 196608 for $D_g = 2$, $D_g = 3$, $D_g = 4$, and $D_g = 8$ respectively. Thus in our

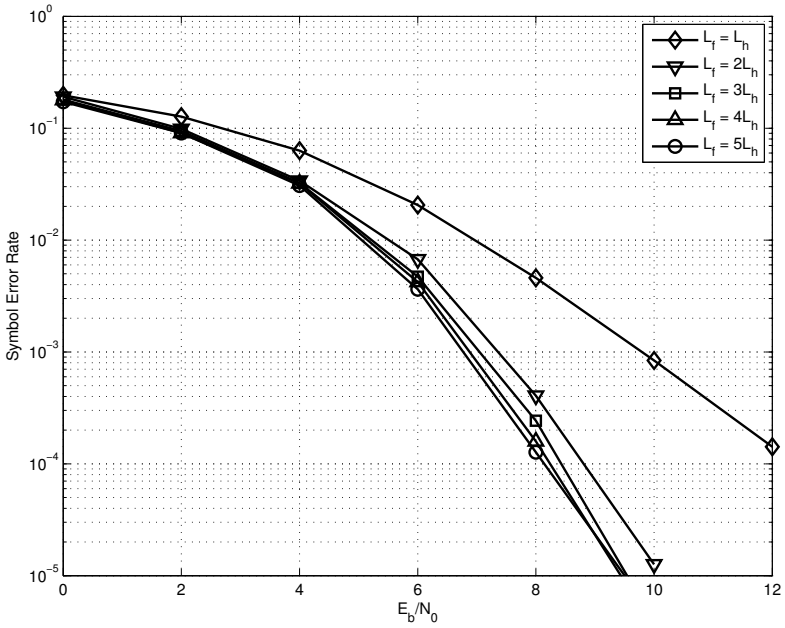

Figure 2.10: SER performance of the PRBP detector for different length of PRE L_f.

implementation example, we use $D_g = 3$ to ensure a good complexity/performance tradeoff.

Fig. 2.12 shows the effect of different number of BP iterations, where the TIR has $D_g = 3$ nonzero coefficients. We can see that the performance improvement is marginal when n_{it} is larger than 5. Since more iterations means more computing time and thus lower throughput, we suggest using $n_{it} = 5$ in practice.

2.6 Detector Implementation Results

The detector implementation in ASIC includes two main modules - the PRE block and BP block. The same set of parameters as in Section 2.5 are applied for hardware implementation: the channel length is 30; the number of nonzero taps in TIR is 3; the number of iterations in BP detection is 5. The pipelined layered processing

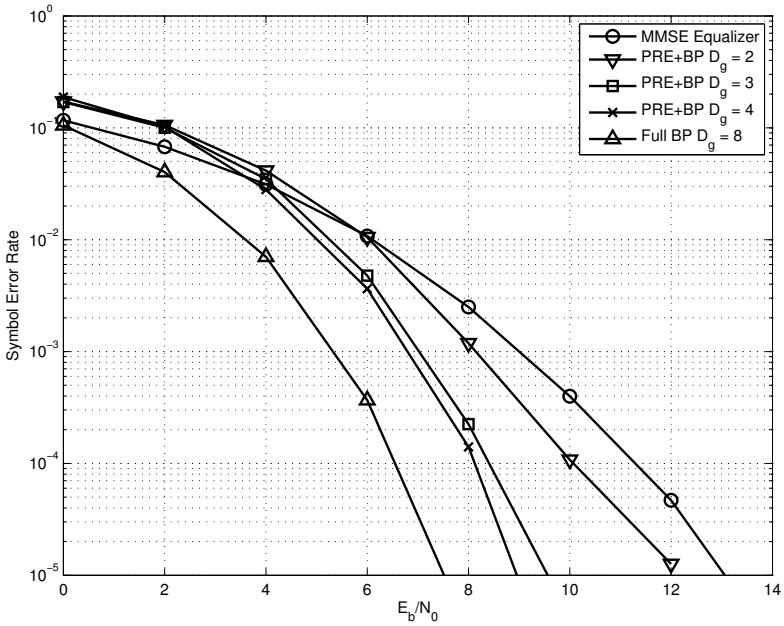

Figure 2.11: SER performance of the PRBP detector for different number of nonzero coefficients in TIR.

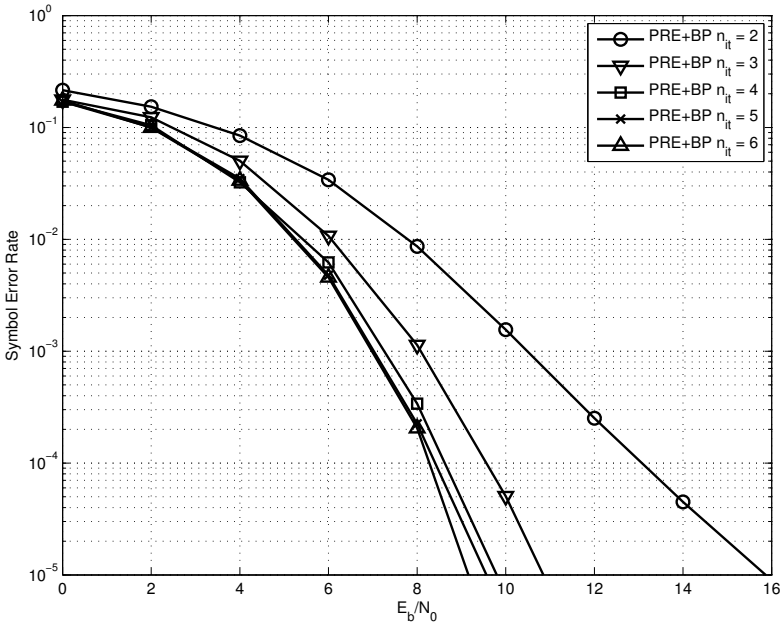

Figure 2.12: SER performance of the PRBP detector for different number of BP iterations.

Detector scheme	PRBP detector	ML detector	MMSE equalizer
Multiplier	$4\lceil L_f/n_{it}\rceil+3M^{D_g}$	$2M^{L_h}$	$4L_f$
Adder	$4\lceil L_f/n_{it}\rceil+(3+D_g)M^{D_g}+3(M-1)D_g$	$4M^{L_h}$	$4L_f$
Comparator	$(M^{D_g-1}-1)MD_g$	$(M-1)M^{L_h-1}$	—
Memory (bits)	$(M-1)ND_g\cdot q_R+(M-1)N\cdot q_\Lambda$	$20L_h\log_2 M\cdot M^{L_h-1}$	—
Throughput (bit/s)	$f\log_2 M/n_{it}$	$f\log_2 M$	$f\log_2 M$
Latency (cycles)	$((L_f+L_h)/2+N)n_{it}$	$10L_h$	$(L_f+L_h)/2$

Table 2.1: Comparison of hardware resource utilization by different detectors.

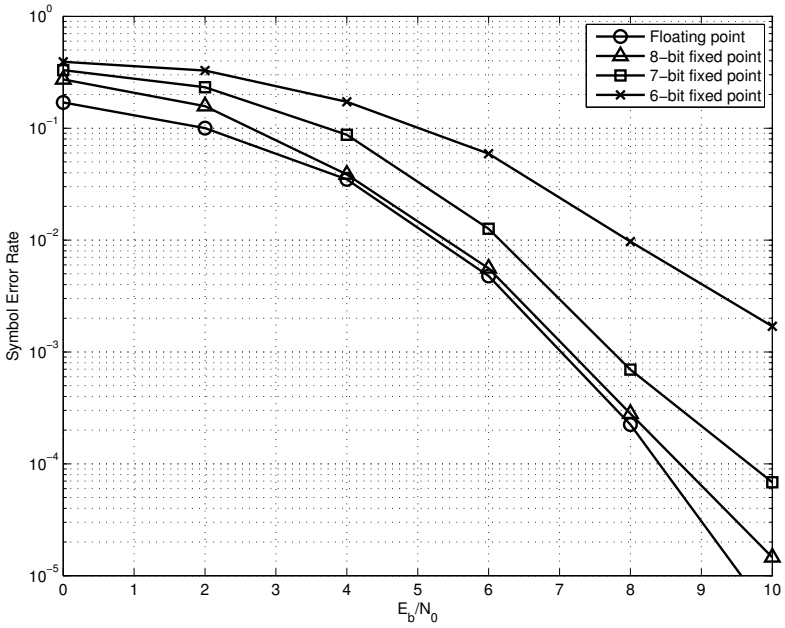

Figure 2.13: SER performance of the PRBP detector for different fixed-point word lengths.

architecture as described in Section 2.4.3 is adopted. The cache-memory architecture presented in Section 2.4.4 is also implemented to store and transfer messages efficiently in the BP detector.

Fixed point quantization is evaluated for the received signals and the belief propagation messages. Fig. 2.13 shows the symbol error rates for different fixed point quantization. For 8-bit word length, the fixed point performance is 0.2 dB away from the floating point result at a SER of 10^{-4}. The SER performance for 7-bit word length is about 1 dB worse than the floating point result. Hereby, we apply 8 bits word length quantization in this implementation.

The LMS block is implemented on a TI TMS320C6748 DSP, which operates at 456 MHz. The result shows that 10,779 clock cycles are needed for each calculation in Algorithm 2.1. Assuming that 300 received samples are required for PRE calculation,

the PRE and TIR information can be obtained within 7.09 ms.

The proposed symbol detector is implemented with TSMC 90-nm technology and synthesized by Synopsys Design Compiler. The synthesis result shows that the maximum clock frequency is up to 515 MHz, the core area is 3.162 mm^2, and the estimated dynamic power is 1.096 W. The overall throughput is about 206 Mb/s calculated by (2.5).

With respect to the implementation complexity, we compare the hardware resource utilization for three schemes in Table 2.1: the proposed detector, the ML detector, and the MMSE equalizer. Complex received signal and complex channel coefficients are considered. For comparison purpose, the ML detector is implemented based on the classic Viterbi algorithm. As the implementation of Viterbi-based decoder has been well studied [30–33], we simply extend the classic architecture to the Viterbi-based detector by calculating the Euclidean distance instead of Hamming distance for branch metric calculation. Four banks of memory are used for pipelined output, and the depth for each bank is equal to the traceback length, which is chosen as $5L_h$ in this table. For the MMSE equalizer, we consider a fully parallel FIR filter implementation using multipliers and adders. The latencies for different schemes are compared. The decision delay is estimated as $(L_f + L_h)/2$ for the MMSE equalizer and the PRE. The latency of the ML detector depends on the traceback scheme. We estimate the latency as twice of the traceback length, i.e. $10L_h$. Table 2.2 shows the actual number of multipliers, adders, comparators and memory use in our implementation of the proposed detector. Since direct implementation of a Viterbi-based ML detector for channel length of 30 is astonishedly expensive in computations and not practical, we omit the listing of resource usage of the ML detector in Table 2.2.

Due to the iterative nature of the BP algorithm, the PRBP detector has the draw-

Detector scheme	PRBP detector	MMSE equalizer
Multiplier	272	400
Adder	491	400
Comparator	180	—
Memory (KBytes)	10	—
Throughput (bit/s)	$0.4f$	$2f$
Latency (cycles)	5445	65

Table 2.2: Comparison of hardware complexity of different detector implementations.

backs of longer processing latency and reduced throughput. However, it demonstrates much improved error performance with affordable complexity. For the parallel implementation, the complexity of an MMSE equalizer is about the same as the BP detector, but its performance is about 3.5 dB worse at SER of 10^{-5} than that of the BP detector ($D_g = 3$, $n_{it} = 5$) which is significant in practical communication systems.

For comparison purpose, the PRBP detector is also implemented on the C6748 DSP processor. For one symbol output, it cost an average of 1,442 cycles, which leads to a throughput of 632.5 Kb/s only. The comparison of the ASIC and DSP implementations are given in Table 2.3. The technology, core voltage, and power consumption of the DSP are provided in [34, 35]. It is shown that the ASIC implementation of the PRBP detector has significantly higher throughput than the DSP implementation.

2.7 Conclusions

For symbol detection under sparse channels, the optimal maximum-likelihood sequence estimator is impractical due to its prohibitively high complexity, while the

	ASIC	DSP
Technology	90 nm	65 nm
Core Voltage	1 V	1.3 V
Power	1.096 W	0.66 W
Throughput	206 Mb/s	632.5 Kb/s

Table 2.3: Comparison of the ASIC and DSP implementations.

MMSE linear equalizer is incapable to provide a superior performance. To trade off between the performance and complexity, we investigated the design of a near-optimal PRBP detector for sparse channels. This two-stage detector consists of the PRE and BP detector. The PRE first equalize the sparse channel to a target impulse response which has a limited number of nonzero channel coefficients. Next the residual ISI is eliminated by the BP detector whose complexity depends merely on the number of nonzero channel coefficients. By exploiting the characteristic of sparse channels, a pipelined layer processing scheme is adopted in the BP detector to achieve high throughput. The PRBP detector was implemented with TSMC 90-nm technology for the first time. The implementation result showed a maximum throughput of 206 Mb/s, with an area of 3.162 mm^2, and a total dynamic power of 1.096 W.

Chapter 3

Performance of the BP-Based Detector for 3G Channels

In Chapter 2, we present an implementable PRBP detector for sparse channels. Note that a system designer can specify the TIR to be sparse even when the original CIR is not sparse, which implies that the PRBP detector is general enough to work even for non-sparse channels. In this chapter, we study the performance of the PRBP detector under more generic ISI channels. The performance of another two-stage detector, the partial response maximum-likelihood (PRML) detector, is also assessed for comparison. The chapter is organized as follows. In Section 3.1, we first introduce related background of symbol detectors that we will investigate in this chapter. The PRML detector is briefly described in Section 3.2. Section 3.3 shows detailed simulation results, along with several remarks on the implementation issues. Finally, conclusions are presented in Section 3.4.

3.1 Introduction

In this chapter, we compare the performance of the PRBP detector with an alternative implementable PRML detector [26,36–40]. Both detectors employ a hybrid two-stage scheme, and allow a tradeoff between performance and complexity. PRML detectors have demonstrated superior error-rate performance over classical equalization methods. In particular, these PRML techniques have shown to be effective in high-density magnetic recording applications. Instead of removing all the ISI components from the received signal, the PRML detector first conditions the long channel using a channel shortening filter (CSF) which shortens the effective CIR to a TIR. A Viterbi-based MLSE detector with reduced states is applied after the channel shortening. Note that, since the CSF does not approximate an all-pass filter, the prefiltering will lead to colored noise depending on the given CIR. This will subsequently degrade the error performance of the second stage of the PRML detector, since the performance of the MLSE detector is optimal only under white noise [39].

We study the performance of the two aforementioned two-stage symbol detectors: the PRML detector and the PRBP detector, under frequency-selective fading wireless channels. The performance of the traditional minimum mean squared error based linear equalizer (MMSE-LE) is also given as the benchmark. ITU-R 3G channel models [41] with slow fading are considered in our simulation. Numerical results show that in the experimental 3G channels, the PRBP detector provides superior performance over the PRML detector and the MMSE-LE. The performance of the PRBP in realistic wireless channels is illustrated for the first time, while in previous works [8], the performance simulation is limited to either a simple 3-tap/5-tap equipower slow fading channel or underwater acoustic channels. In order to provide

reference for the PRBP detector design for 3G wireless applications, we also discuss several implementation issues on the PRBP detector. Moreover, through simulations, we show that the PRML detector is not as useful in wireless fading channels as it is in high density magnetic recording channels since the spectrum of the TIR is not always well-matched to that of the CIR, and this introduces increased colored noise for the MLSE in the second stage.

3.2 Partial Response Maximum-Likelihood (PRML) Detector

Fig. 3.1 shows the system model of the PRML detector. The received signal $y[k]$ is first processed by a prefilter $\mathbf{f}$ which performs partial equalization. In the second stage, a symbol detector is applied to mitigate the residual ISI components. Note that this system model of the PRML detector is very similar to that of the PRBP detector as shown in Fig. 2.1. For the PRML detector, the TIR $\mathbf{g}$ is a shortened version of the CIR, while for the PRBP detector, the TIR $\mathbf{g}$ is a sparse version of the CIR; in both cases, the TIR has a small number of nonzero entries. When the TIR is chosen to have a single nonzero tap, the prefilter $\mathbf{f}$ is equivalent to a linear equalizer. In that case, a simple memoryless slicer can optimally be employed as the symbol detector to output the estimated symbol $\hat{x}[k]$.

For PRML detectors, the prefilter $\mathbf{f}$ serves as a CSF which shortens the CIR to a reasonably short TIR, so that a conventional Viterbi-based MLSE detector can be applied to eliminate the residual ISI. We adopt a widely used approach to design the TIR and the CSF, which is to jointly minimize the MSE between the CSF output and the TIR output under a unit-energy constraint [26]. In order to facilitate a practical

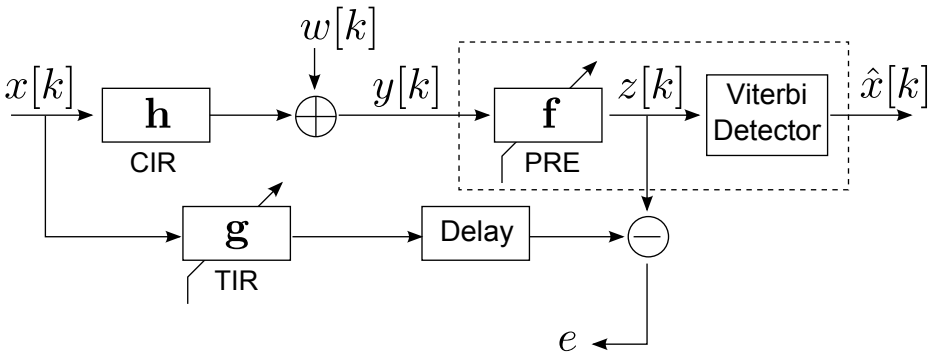

Figure 3.1: System model of the partial response maximum-likelihood detector.

implementation, the TIR and the CSF can be optimized using adaptive algorithms such as the least mean square (LMS) algorithm. After the convergence of the adaptive algorithm, the combined response of the CIR and the CSF approximates the TIR which contains $L_g \ll L_h$ taps. Therefore, the complexity of the Viterbi detector can be significantly reduced.

Since the CSF is not an all-pass filter, it can lead to colored noise, and the effect of the colored noise can become severe when the spectra of the CIR and the TIR are not well matched [39]. In certain wireless environments, the CIR appears to be very long and sparse (see Fig. 3.2). Since the TIR taps are required to be contiguous for the PRML detector, the corresponding CSF can lead to severe colored noise for the second stage MLSE. As a result, the performance of the whole detector may be affected, since the classical MLSE is optimum only for white noise.

3.3 Numerical Results and Remarks

3.3.1 Simulation Parameters

We evaluate the performance of the discussed detectors for frequency-selective fading channels in terms of SER versus SNR per bit E_b/N_0. We assume that the channel information is perfectly known at the receiver, and the channel coding is not taken into account. The modulation scheme is quadrature phase-shift keying (QPSK), so that $M = 4$. A sequence of $N = 1024$ symbols are processed each time. We assume that the SRRC pulse shaping filter is employed at both the transmitting and receiving ends. The duration time of pulse waveform is truncated to $[-2/f, 2/f]$ with a roll-off factor 0.25, where $f = 50\,\text{MHz}$ is the symbol rate. We simulate over 200 independent fading realizations using the ITU-R 3G channel model. Four channel profiles (Table 3.1) are considered. Given the simulation parameters above, the example CIR for each channel profile is shown in Fig. 3.2.

3.3.2 Simulation Results

The performance of the PRML detector and the PRBP detector is given in Figs. 3.3 through 3.6 for different channel profiles. In all cases, the length of the TIR is 3 for the PRML detector. Moreover, the number of nonzero taps of the TIR is 3 for the PRBP detector, where 5 iterations of message passing are applied. The MMSE-LE performance curve is also given as a reference. The prefilter length L_f is chosen as $5L_h$, which is sufficiently large according to the conventional wisdom for classical equalizer design [42].

As can be seen from the simulation results, the PRBP detector universally outperforms the other two candidates. Particularly, with a very long CIR, for example,

Channel Profile	Path Delays (μs)	Path Gain (dB)
Indoor office, channel A	[0 0.05 0.11 0.17 0.29 0.31]	[0 -3 -10 -18 -26 -32]
Indoor office, channel B	[0 0.1 0.2 0.3 0.5 0.7]	[0 -3.6 -7.2 -10.8 -18 -25.2]
Outdoor to indoor and pedestrian, channel A	[0 0.11 0.19 0.41]	[0 -9.7 -19.2 -22.8]
Outdoor to indoor and pedestrian, channel B	[0 0.2 0.8 1.2 2.3 3.7]	[0 -0.9 -4.9 -8 -7.8 -23.9]

Table 3.1: ITU-R 3G Channel profiles used in simulations.

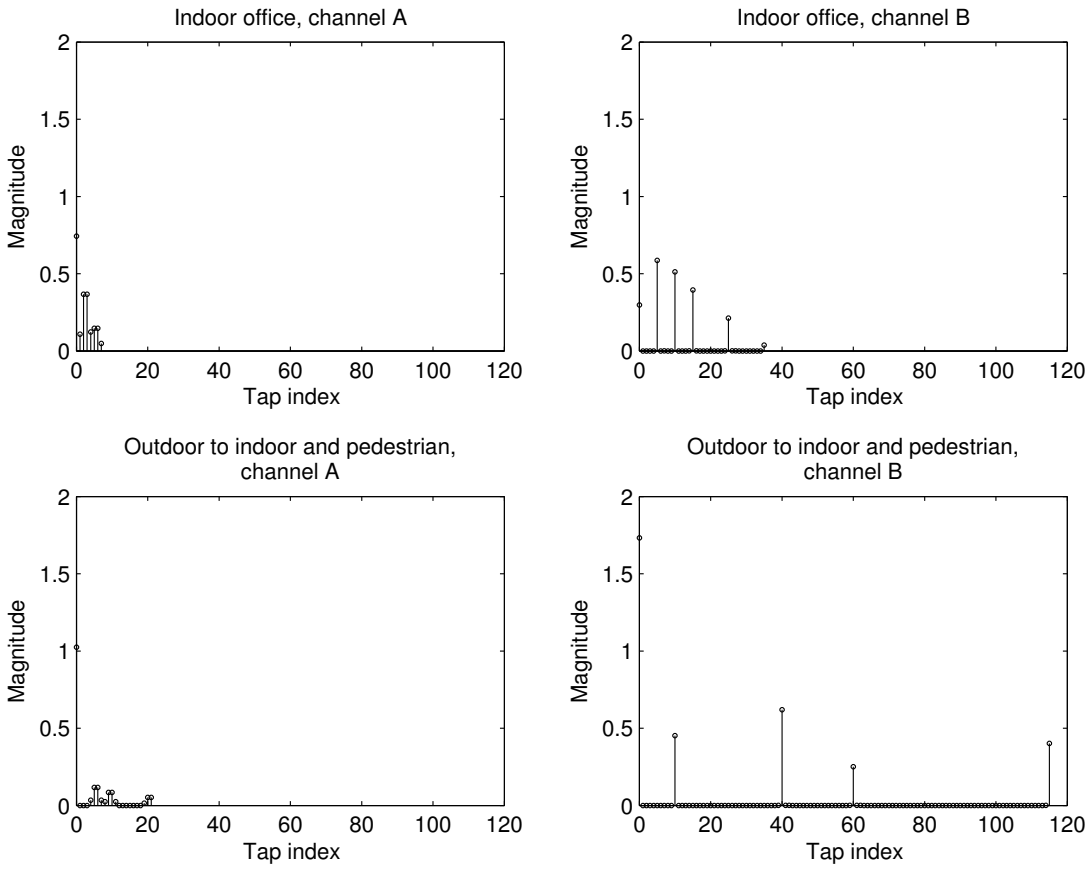

Figure 3.2: Example CIR for ITU-R 3G channel profiles.

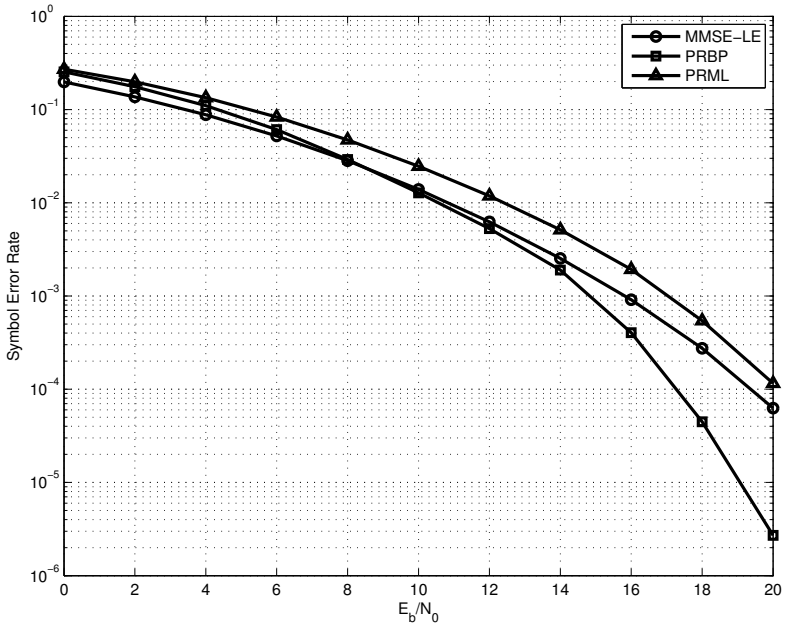

Figure 3.3: SER performance of detectors under consideration under ITU-R 3G indoor office, channel A.

indoor/outdoor channel B, the performance gain is significant. At a SER of 10^{-4}, the PRBP detector exhibits performance 3.5 dB better than the MMSE-LE under indoor office channel B, and 2 dB better under outdoor to indoor and pedestrian channel B. It is also worthwhile to mention that, although the PRML detector is effective in high-density magnetic recording, it does not show any superiority over the MMSE-LE scheme in 3G fading channels, due to the effect of colored noise.

3.3.3 Remarks

According to Section 2.3.2, the overall complexity of the PRBP detector depends on the length of the prefilter and the number of nonzero taps of the TIR. Larger filter lengths generally give better performance, but also increase complexity as measured by the number of multiplications and additions. Fig. 3.7 shows the performance of the

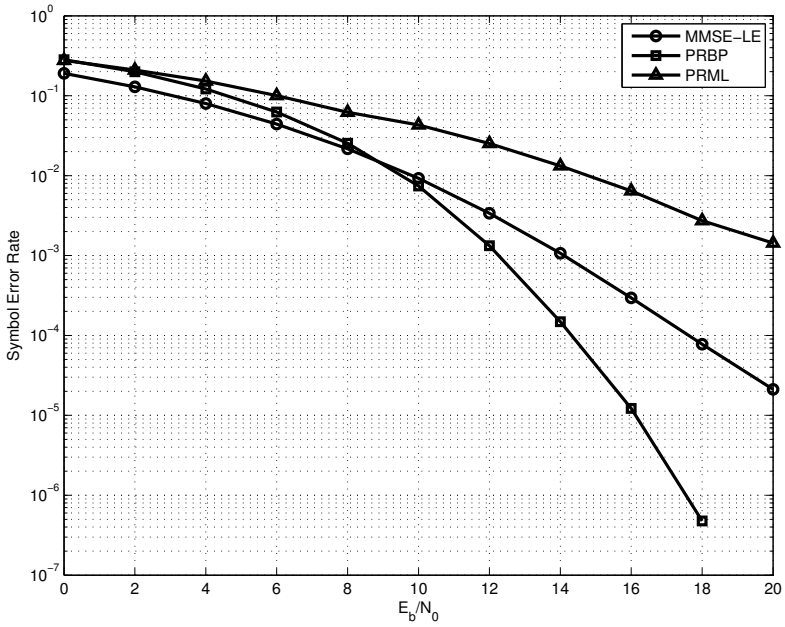

Figure 3.4: SER performance of detectors under consideration under ITU-R 3G indoor office, channel B.

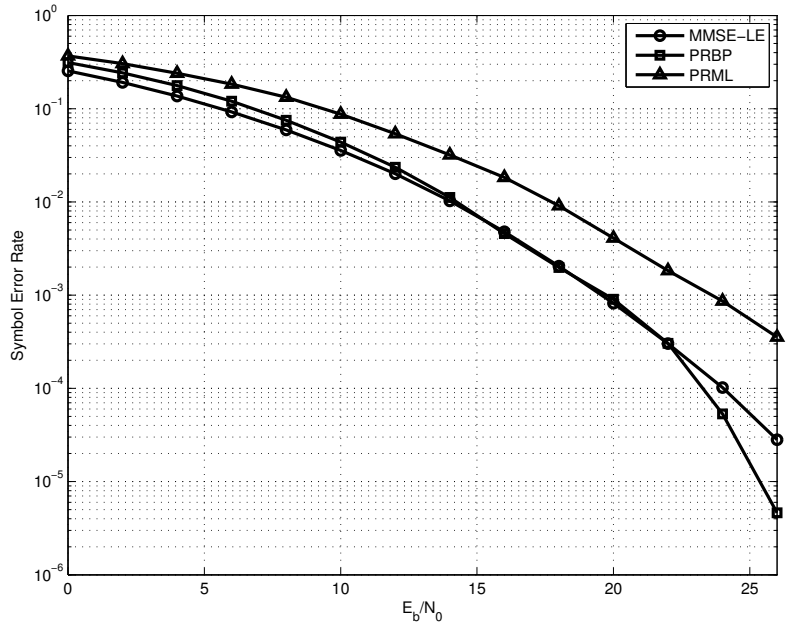

Figure 3.5: SER performance of detectors under consideration under ITU-R 3G outdoor to indoor and pedestrian, channel A.

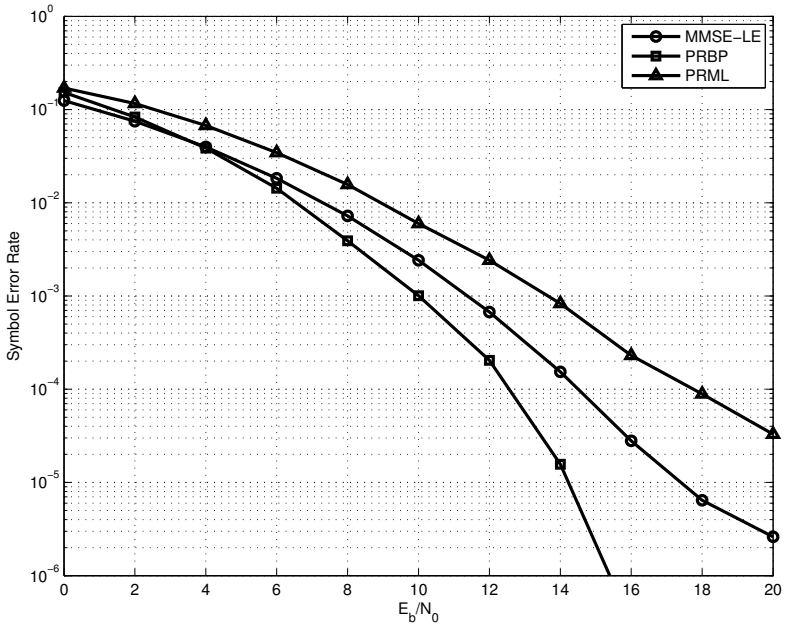

Figure 3.6: SER performance of detectors under consideration under ITU-R 3G outdoor to indoor and pedestrian, channel B.

PRBP detector with different prefilter lengths under ITU-R 3G indoor office, channel A. It can be seen that a filter length of about 3 times the channel length (i.e. choosing $L_f \approx 3L_h$) is sufficient in practice for this case. Further increase of the filter length only gives marginal performance improvement.

Moreover, the performance of the PRBP detector improves as the number of nonzero taps in the TIR grows. Fig. 3.8 illustrates the performance of the PRBP detector with different amounts of nonzero taps in the TIR for ITU-R 3G indoor office, channel A. We suggest a minimum of 3 nonzero taps in the TIR for the PRBP detector in order to achieve performance that begins to outperform a conventional MMSE-LE. On the other hand, it is advisable to limit the number of nonzero taps in the TIR in order to make the BP detector implementable.

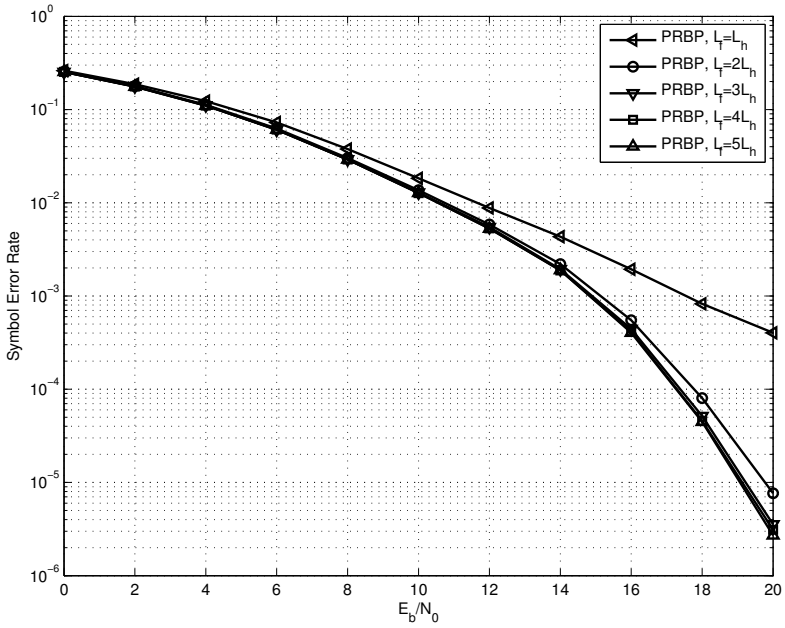

Figure 3.7: SER performance of the PRBP detector with different length of the prefilter under ITU-R 3G indoor office, channel A.

3.4 Conclusions

We reviewed two feasible two-stage hybrid detectors, including the partial response maximum-likelihood detector and the partial response belief propagation detector. ITU-R 3G channel models with slow fading were considered in our simulation. From the numerical results, the PRBP detector showed an advantage in error rate performance over the PRML detector and the traditional minimum mean squared error based linear equalizer. Useful remarks on implementation issues were given to provide guidance for the PRBP detector design in 3G wireless applications. Moreover, we showed that in the experimental 3G channels, the PRML detector is not as effective as it is in high density magnetic recording applications, due to the spectrum mismatch between the channel impulse response and the target impulse response, which leads to performance degradation of the second stage equalization.

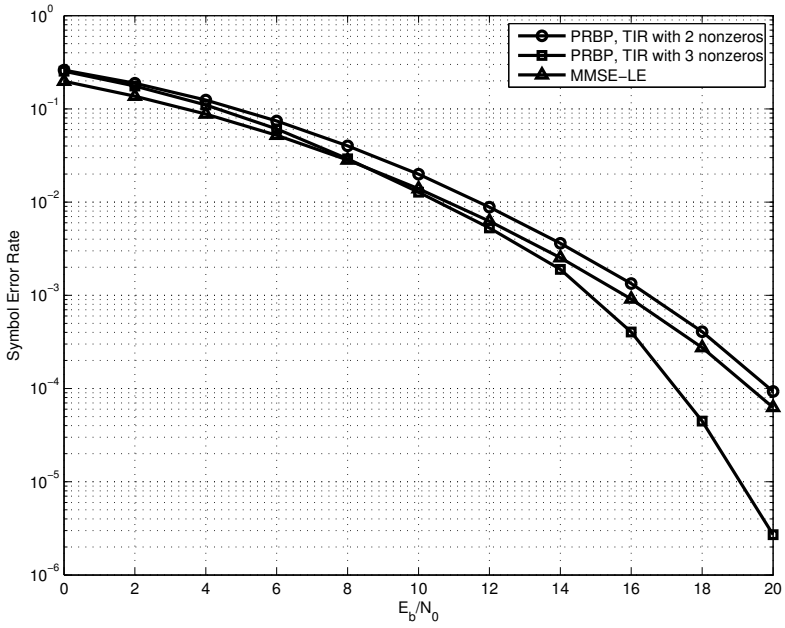

Figure 3.8: SER performance of the PRBP detector with different TIR under ITU-R 3G indoor office, channel A.

Chapter 4

DFSF-BP Detector for ATSC Channels

In this chapter, we present a hybrid DFSF-BP detector for ATSC DTV receivers. The DFSF-BP detector is in a category of PRBP detector. Different from the BP-based detector introduced in Chapter 2 and Chapter 3, the DFSF-BP detector employs a non-linear filter, the decision feedback sparsening filter (DFSF), as the partial response equalizer. By combining the non-linear filter with the BP detector, the error propagation encountered in the traditional DFE based detector can be effectively reduced. The chapter is organized as follows. The background of the ATSC DTV systems is introduced in Section 4.1. Section 4.2 gives the system model for time-varying channels. The proposed detector is described in Section 4.3. We show the simulation performance in Section 4.4, followed by conclusions in Section 4.5.

4.1 Introduction

In Advanced Television Systems Committee (ATSC) digital television (DTV) systems [43], terrestrial channels often suffer from strong multipath distortion. The duration of the CIR can span hundreds of symbol periods. The long and large pre-cursors and post-cursors of the channel impose great challenges to performing reliable equalization of the 8-vestigial sideband (VSB) signals. To combat the severe ISI by the multipath channel, decision feedback equalizers [44–46] are commonly used in DTV receivers. It is well known that DFEs undergo error propagation which results in bursty errors. When the symbol decisions from the slicer output are incorrect, the feedback filter fails to subtract off accurate residual ISI from the feedforward filter output. The incorrect decisions exacerbate the ISI when large post-cursors are present in the combined response of the channel and feedforward equalizer since the largest taps in the feedback path will contribute the most unintended ISI.

In order to reduce error propagation, a DFSF combined with a BP detector was proposed in [9]. First, the DFSF conditions the channel to a sparse channel with only a few nonzero taps, the number of which is specified by the system designer. By setting the largest taps in the feedback path to zero, residual ISI is intentionally present at the output of the DFSF. By specifically zeroing the largest taps in the feedback filter (corresponding to the taps that contribute the most ISI), there is less chance of introducing unintended ISI in the event of symbol decision errors. In the next stage of the receiver, the residual sparse ISI is compensated by the BP detector, which provides near-optimal error performance, with the complexity depending on the number of system designer-specified nonzero taps in the effective channel. This BP detector has been shown to be effective and feasible for a sparse channel with

only a few nonzero taps [47, 48].

In this chapter, we present a hybrid DFSF-BP equalization scheme for DTV receivers. Our idea is conceived from the scheme in [9], though the major difference from previous work is that our scheme targets 8-VSB-based ATSC DTV systems and is suitable for channels that experience Doppler fading; [9] only considers static channels and BPSK modulation. In our proposed scheme, the DFSF is designed to adapt to the time-varying channel using the least mean square (LMS) algorithm, such that the channel can be assumed to be fixed after the DFSF processing. Thus the second-stage BP detector can benefit from the fixed channel by reducing the implementation complexity. We assess the error performance of the proposed scheme by simulation under static/dynamic ATSC DTV channel models. Simulation results show that the proposed scheme outperforms the traditional DFE in both static and dynamic environments.

4.2 System Model

The system model is shown in Fig. 4.1. Since the focus of this chapter is the equalization scheme, channel coding/decoding blocks are omitted for simplicity. We assume that the data symbols $x[k]$ are Reed-Solomon (RS)-encoded, interleaved, and trellis-encoded symbols, drawn i.i.d. from the 8 level pulse amplitude modulation constellation $(\pm 1, \pm 3, \pm 5, \pm 7)$ [43]. The data symbols $x[k]$, with a variance of σ_x^2, are transmitted through an ISI channel at a symbol rate of 10.76 MHz.

We assume that a SRRC filter is applied as the transmitter filter. The received signal is processed by a matched filter, and then sampled at the symbol rate. The equivalent discrete-time CIR at time k which includes the effects of pulse shaping is

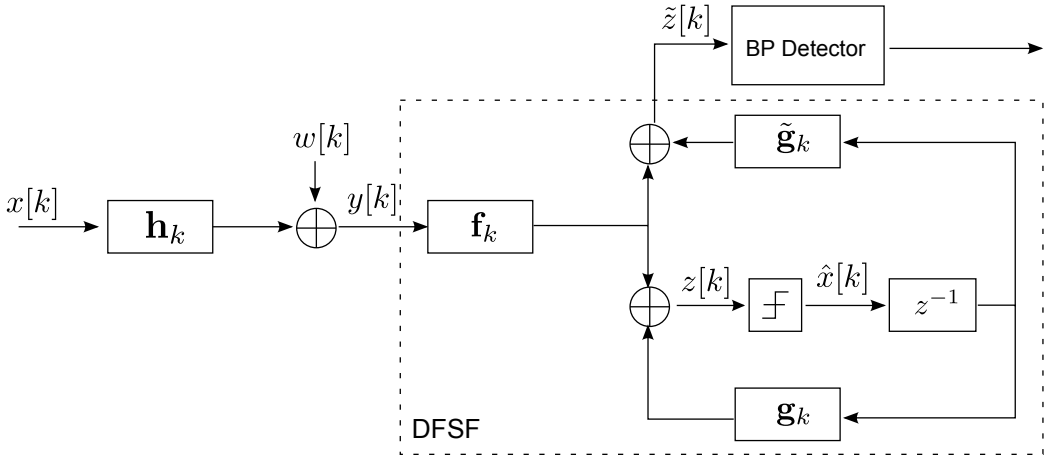

Figure 4.1: System model of the DFSF-BP detector.

described as $\mathbf{h}_k = [h_k[0], h_k[1], ..., h_k[L_h - 1]]^T$, where L_h is the channel length. The received signal at time k can be expressed as

$$y[k] = \sum_{i=0}^{L_h-1} h_k[i]x[k-i] + w[k],$$

where $w[k]$ is additive white Gaussian noise with variance σ_n^2.

The DFSF consists of two parts: a regular DFE and a modified feedback filter $\tilde{\mathbf{g}}_k$. In the DFE, the feedforward filter and the feedback filter are denoted as $\mathbf{f}_k$ and $\mathbf{g}_k$, respectively. Note that since the channel $\mathbf{h}_k$ is time-varying, the filters are also time dependent. The filters $\mathbf{f}_k$, $\mathbf{g}_k$, and $\tilde{\mathbf{g}}_k$ have discrete finite impulse responses, which are expressed as $\mathbf{f}_k = [f_k[0], f_k[1], ..., f_k[L_f - 1]]^T$, $\mathbf{g}_k = [g_k[0], g_k[1], ..., g_k[L_g - 1]]^T$, and $\tilde{\mathbf{g}}_k = [\tilde{g}_k[0], \tilde{g}_k[1], ..., \tilde{g}_k[L_{\tilde{g}} - 1]]^T$. The input signal for the slicer is given by

$$z[k] = \sum_{i=0}^{L_f-1} f_k[i]y[k-i] + \sum_{i=0}^{L_g-1} g_k[i]\hat{x}[k-i-1],$$

where $\hat{x}[k]$ is the tentative decisions from the slicer. The modified feedback filter only

suppresses partial ISI using the tentative decisions. The output signal of the DFSF is

$$\tilde{z}[k] = \sum_{i=0}^{L_f-1} f_k[i]y[k-i] + \sum_{i=0}^{L_g-1} \tilde{g}_k[i]\hat{x}[k-i-1]. \tag{4.1}$$

Assuming that the tentative decisions are correct, $\hat{x}[k]$ is the delayed version of symbol input $x[k]$. Then (4.1) can be simplified as

$$\tilde{z}[k] = \sum_{i=0}^{L_c-1} c_k[i]x[k-i] + n[k],$$

where $c_k[i]$ contains the coefficients of the combined response of the CIR and DFSF at time k, and $n[k]$ is the DFSF output noise. We denote the combined response of the CIR and DFSF as $\mathbf{c}_k$ with the channel length L_c. The system designer chooses the desired number of nonzero taps D in $\mathbf{c}_k$, and the modified feedback filter $\tilde{\mathbf{g}}_k$ is obtained from $\mathbf{g}_k$ by setting its $(D-1)$ largest taps to zero.

In the second stage, the BP detector will compensate for the D nonzero taps. Since the complexity of the BP detector depends the number of nonzero taps of the CIR, we can trade performance for complexity of the overall equalization scheme by choosing different values of D. With small values of D, most ISI is canceled by the DFSF, and thus the burden on the complexity of the BP detector is reduced. However, it is more likely that the unintended ISI is introduced in the feedback loop, which will cause error propagation. A extreme case is to choose $D = 1$ in which case the DFSF is equivalent to the regular DFE, and the BP detector is reduced to a slicer. To minimize the incorrect feedback of ISI components, the system designer can increase the value of D so that more ISI components are processed by the near-optimal BP detector. On the other hand, since the complexity of the BP detector increases exponentially with the nonzero taps of the CIR, D should be constrained

to a small enough value that permits practical implementation.

4.3 DFSF-BP Detector

In DTV systems, the channel typically experiences Doppler fading [46]. In [9], the channel is assumed to be fixed and known perfectly to the receiver, and without modification the DFSF-BP scheme of [9] cannot be directly applied here. We now extend the DFSF-BP equalization scheme to an adaptive implementation suitable for use in time-varying channels. Recall that the receiver under consideration is a two-stage receiver, where in the first stage an equalizer partially equalizes the channel, and the residual ISI is compensated by a belief propagation detector in the second stage.

The equalizer operates in two modes: (A) a startup mode, where the DFSF taps are initialized based on training data or other channel sounding techniques, and (B) a tracking mode where the DFSF adapts to the time-varying channel using a decision-directed approach. We make the assumption that the channel is approximately static during the startup mode, and as such we can directly apply the results of [9] to initialize the DFSF taps.

4.3.1 Startup Mode

We adopt the minimum mean-squared error (MMSE) criterion to design the feedforward filter $\mathbf{f}_k$, the feedback filter $\mathbf{g}_k$, the modified feedback filter $\tilde{\mathbf{g}}_k$. Here we assume that the channel is approximately static over a block of length L_f. The filters $\mathbf{f}_k$ and $\mathbf{g}_k$ are designed to have coefficients given by the classical MMSE-DFE which has

solution

$$\mathbf{f}_k = \sigma_x^2 \left(\sigma_x^2 \mathbf{H}_k(\mathbf{I}_{L_c} - \mathbf{\Sigma}^T\mathbf{\Sigma})\mathbf{H}_k^T + \sigma_n^2 \mathbf{I}_{L_f}\right)^{-1} \mathbf{H}_k \mathbf{e}, \tag{4.2}$$

$$\mathbf{g}_k = -\mathbf{\Sigma}\mathbf{H}_k^T \mathbf{f}_k, \tag{4.3}$$

where $\mathbf{H}_k$ is the channel convolution matrix denoted as

$$\mathbf{H}_k = \begin{bmatrix} h_k[0] & h_k[1] & \dots & h_k[L_h-1] & 0 & 0 & \dots \\ 0 & h_k[0] & h_k[1] & \dots & h_k[L_h-1] & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ 0 & & & & & & \\ 0 & \dots & 0 & h_k[0] & h_k[1] & \dots & h_k[L_h-1] \end{bmatrix},$$

and

$$\mathbf{\Sigma} = [\mathbf{0}_{L_g \times (\Delta+1)} \quad \mathbf{I}_{L_g} \quad \mathbf{0}_{L_g \times (L_h+L_f-L_g-\Delta-2)}],$$

and

$$\mathbf{e} = \left[\underbrace{0, \dots, 0}_{\Delta}, \; 1, \; \underbrace{0, \dots, 0}_{L_f+L_h-\Delta-2}\right]^T,$$

where Δ is the decision delay of the DFE slicer. Next, $\tilde{\mathbf{g}}_k$ is chosen to be equal to $\mathbf{g}_k$ but with the $(D-1)$ largest taps set to zero to mitigate the effects of errors out of the symbol slicer. Consequently, the coefficients of the combined response of the CIR and DFSF $\mathbf{c}_k$ can be calculated as

$$c_k[i] = h_k[i] \star f_k[i] + \tilde{g}_k[i - \Delta - 1], \tag{4.4}$$

where $(\star)$ denotes the convolution operation. We note that, under the assumption of correct decisions at the output of the slicer, $\mathbf{c}_k$ corresponds to an impulse response with D nonzero coefficients.

To compute the initial MMSE equalizer settings at the start of transmission, we make the assumption that the receiver either has knowledge of the initial channel coefficients and can compute the initial equalizer setting via (4.2) and (4.3); alternatively, the receiver may use training data and a trained algorithm like LMS to directly adapt the equalizer coefficients $\mathbf{f}_k$, $\mathbf{g}_k$, and $\tilde{\mathbf{g}}_k$ to the initial MMSE solution.

Finally, we make the assumption that the startup mode is completed at time $k = 0$, so $\mathbf{f}_0$ and $\mathbf{g}_0$ are the MMSE-DFE coefficients. Furthermore, $\tilde{\mathbf{g}}_0$ is a zeroed version of $\mathbf{g}_0$, and the initial combined response $\mathbf{c}_0$ can be computed via (4.4).

4.3.2 Tracking Mode

We now develop an approach for using the DFSF with time-varying channels that experience Doppler. One approach we could employ is to continuously adapt (using, for example, decision-directed LMS) the coefficients $\mathbf{f}_k$, $\mathbf{g}_k$, and $\tilde{\mathbf{g}}_k$ to track the MMSE solution at time k for any channel $\mathbf{h}_k$. Then, however, the effective channel $\mathbf{c}_k$ observed by the BP detector would also be time-varying since the intentional residual ISI terms (i.e. those taps which get zeroed in $\tilde{\mathbf{g}}_k$) change in both amplitude and location as the channel changes. Such a time-varying effective channel would lead to a BP detector which is significantly more complicated than a more conventional static BP detector.

Consequently, instead of adapting $\mathbf{f}_k$, $\mathbf{g}_k$, and $\tilde{\mathbf{g}}_k$ to track the MMSE solution, we adapt them so the combined effective response $\mathbf{c}_k$ appears static to the BP detector. That is, we design an algorithm to adapt $\mathbf{f}_k$, $\mathbf{g}_k$, and $\tilde{\mathbf{g}}_k$ so that the combined response is equal to the initial combined response $\mathbf{c}_0$ in the MMSE sense. As such, the equalizer coefficients will coincide with the MMSE-DFE setting at startup, and will then gradually drift away from that setting as the channel changes and the equalizer adjusts to maintain a static effective channel.

As the channel changes significantly, the feedback filter $\tilde{\mathbf{g}}_k$ may have taps that grow large in attempt to keep the combined response equal to $\mathbf{c}_0$. Recall that our motivation for using the hybrid DFSF-BP scheme is to mitigate error propagation by keeping taps in the feedback path small, and relying on the more sophisticated BP algorithm to compensate for the significant ISI terms. Consequently, it may prove beneficial to periodically reset the DFSF if the channel drifts to a situation resulting in large taps in $\tilde{\mathbf{g}}_k$, which can be accomplished by repeating the startup procedure.

Once the initial $\mathbf{f}_k$, $\mathbf{g}_k$, and $\tilde{\mathbf{g}}_k$ are obtained, we adapt the DFSF to the channel using the LMS algorithm in decision-directed mode. The decision-directed LMS (DD-LMS) algorithm for DFSF is listed in Algorithm 4.1. As mentioned above, the adaptive DFSF tracks the time-varying channel and keeps the combined response of the CIR and DFSF $\mathbf{c}_k$ fixed from the perspective of the BP detector, which explains the presence of $\mathbf{c}_0$ in the error term $\tilde{e}[k]$ used in updating $\tilde{\mathbf{g}}_k$. By adapting the DFSF to maintain a static combined response, the implementation of the BP detector is drastically simplified.

In the second stage, we adopt an iterative equalizer based on the BP algorithm, which has been widely used for iterative decoding of low-density parity-check (LDPC) codes [17]. Although the complexity of the BP detector increases only with the number of nonzero channel coefficients, the direct implementation of the BP detector is still impractical due to the prohibitively high complexity even for sparse channels where the number of nonzero coefficients is on the order of 10 [47]. To make use of the near-optimal BP detector, the number of nonzero taps D must be limited to a small number. This permits the design effort of the BP detector to be greatly mitigated while maintaining near-optimal performance. Since the algorithm and the implementation of the BP detector have been elaborated in Chapter 2, we will not

Algorithm 4.1 DD-LMS algorithm for DFSF design.

Parameters:
Feedforward filter coefficients at time k: $\mathbf{f}_k$
Feedback filter coefficients at time k: $\mathbf{g}_k$
Modified feedback filter coefficients at time k: $\tilde{\mathbf{g}}_k$
Combined response of the CIR and DFSF at time k: $\mathbf{c}_k$
Received data vectors at time k:
$\mathbf{y}_k = [y[k], y[k-1], \ldots, y[k-L_f+1]]^T$
Tentative decisions at time k:
$\hat{\mathbf{x}}_{k,L_g} = [\hat{x}[k], \hat{x}[k-1], \ldots, \hat{x}[k-L_g+1]]^T$
$\hat{\mathbf{x}}_{k,L_c} = [\hat{x}[k], \hat{x}[k-1], \ldots, \hat{x}[k-L_c+1]]^T$
Step size for updating $\mathbf{f}_k$: μ_f
Step size for updating $\mathbf{g}_k$: μ_g
Step size for updating $\tilde{\mathbf{g}}_k$: $\mu_{\tilde{g}}$
Decision-directed error at time k: $e[k]$
Error between the actual DFSF output and the expected DFSF output at time k: $\tilde{e}[k]$

Startup mode:
Set the initial coefficients $\mathbf{f}_0$ and $\mathbf{g}_0$ using (4.2) and (4.3).
Set the initial coefficients $\tilde{\mathbf{g}}_0$ to $\mathbf{g}_0$ but with the $(D-1)$ largest taps set to zero.
Set the initial coefficients $\mathbf{c}_0$ using (4.4).

Tracking mode:
Update the coefficients at each time instant:

$$\begin{aligned} \mathbf{f}_{k+1} &= \mathbf{f}_k - \mu_f e[k]\mathbf{y}_k \\ \mathbf{g}_{k+1} &= \mathbf{g}_k - \mu_g e[k]\hat{\mathbf{x}}_{k-1,L_g} \\ \tilde{\mathbf{g}}_{k+1} &= \tilde{\mathbf{g}}_k - \mu_{\tilde{g}} \tilde{e}[k]\hat{\mathbf{x}}_{k-1,L_g}, \end{aligned}$$

where

$$\begin{aligned} e[k] &= \mathbf{f}_k^T \mathbf{y}_k + \mathbf{g}_k^T \hat{\mathbf{x}}_{k-1,L_g} - \hat{x}[k] \\ \tilde{e}[k] &= \mathbf{f}_k^T \mathbf{y}_k + \tilde{\mathbf{g}}_k^T \hat{\mathbf{x}}_{k-1,L_g} - \mathbf{c}_0^T \hat{\mathbf{x}}_{k,L_c}. \end{aligned}$$

discuss the details in this chapter.

4.4 Numerical Results and Remarks

The performance of the proposed detector is evaluated by simulations in terms of SER versus signal-to-noise ratio per bit E_b/N_0. The static channel profile used for the simulation is Brazil A channel and Brazil C channel which are the severe indoor channel used for the Laboratory Test in Brazil, and the dynamic channel profile is CRC channel #4. The channel profile details [46] are listed in Table 4.1. Note that the attenuation of Path 5 of the CRC Dynamic #4 channel is denoted as *threshold of visibility* (TOV). We set the attenuation of Path 5 to 3 dB for different Doppler shifts [49]. According to the ATSC DTV standard, the symbol rate is 10.76 MHz and the roll-off factor of the pulse shaping filter is 11.5%. The performance of a classical MMSE linear equalizer (LE) and classical DFE is also simulated to compare with the proposed detector. The LE employs an FIR with 800 taps. The DFE consists of 400 feedforward taps and 400 feedback taps. The filter lengths for the DFSF are $L_f = L_g = L_{\tilde{g}} = 400$, and the updating steps are $\mu_f = \mu_g = \mu_{\tilde{g}} = 10^{-5}$. The combined response of the CIR and DFSF $\mathbf{c}_k$ contains $D = 3$ nonzero taps. The number of iterations in BP detection is 5.

We first assess the proposed detector in static ATSC channels. Fig. 4.2 and Fig. 4.3 show the SER performance for Brazil A and Brazil C channel respectively. It is shown that at a SER of 10^{-5}, the proposed detector exhibits performance about 1 dB better than the traditional DFE for both channels.

The SER performance for dynamic channel is shown in Fig. 4.4 and Fig. 4.5, for Doppler shift $f_d = 1$ Hz and $f_d = 5$ Hz, respectively. We can see that the

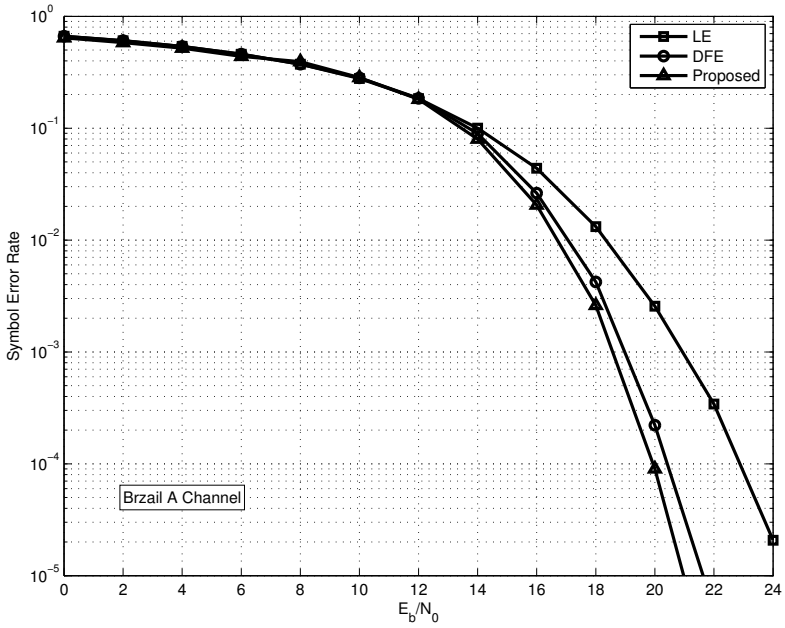

Figure 4.2: SER performance of the DFSF-BP detector in Brazil A channel

proposed detector can successfully track the channel changes in both cases. It is also demonstrated that the proposed detector provides around 1 dB performance gain over the DFE at a SER of 10^{-5}.

4.5 Conclusions

We presented a hybrid detector for ATSC DTV systems in order to reduce the error propagation effect encountered in traditional DFE based receivers, where a large amount of undesirable ISI can be introduced from the feedback filter by incorrect tentative decisions in the DFE. In our scheme we first used a decision feedback sparsening filter (DFSF) to equalize the time-varying channel to a static sparse channel with only a few nonzero taps. Then a near-optimal BP detector was adopted to further compensate the residual ISI. Since the DFSF uses a modified feedback filter

Table 4.1: ATSC Channel profile details.

Channel Ensemble	Channel Parameter	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6
Brazil A	Delay (μs)	0	0.15	2.22	3.05	5.86	5.93
	Attenuation (dB)	0	13.8	16.2	14.9	13.6	16.4
	Phase (degree)	0	0	0	0	0	0
Brazil C	Delay (μs)	0	0.089	0.419	1.506	2.322	2.799
	Attenuation (dB)	2.8	0	3.8	0.1	2.5	1.3
	Phase (degree)	0	0	0	0	0	0
CRC Dynamic #4	Delay (μs)	0	-1.8	0.15	1.8	5.7	35
	Attenuation (dB)	0	11	11	1	TOV	9
	Phase or Doppler	0	125°	80°	45°	1/5 Hz	90°

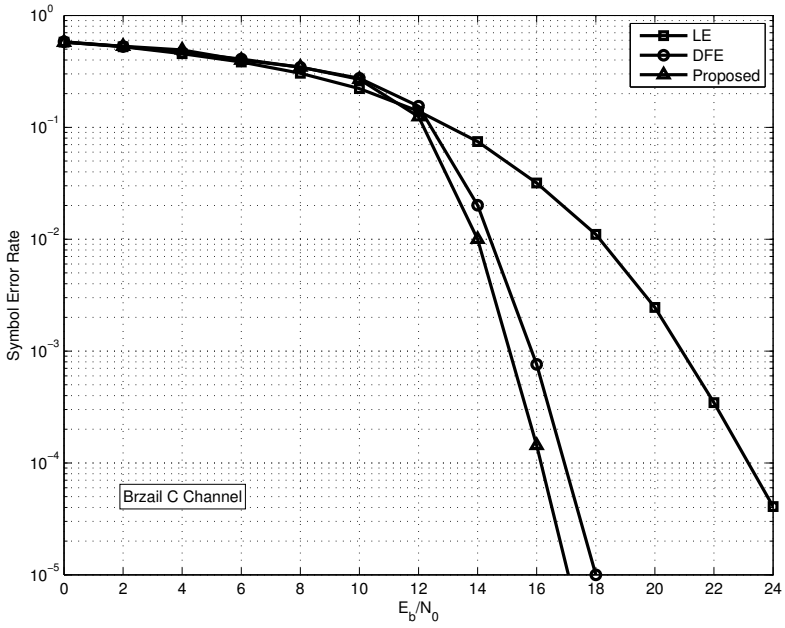

Figure 4.3: SER performance of the DFSF-BP detector in Brazil C channel

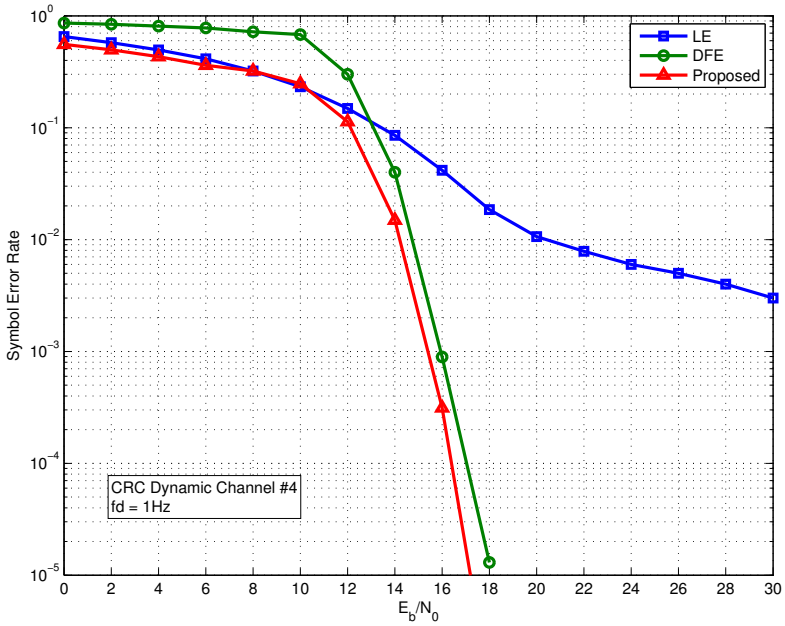

Figure 4.4: SER performance of the DFSF-BP detector in CRC #4 channel with 1Hz Doppler shift

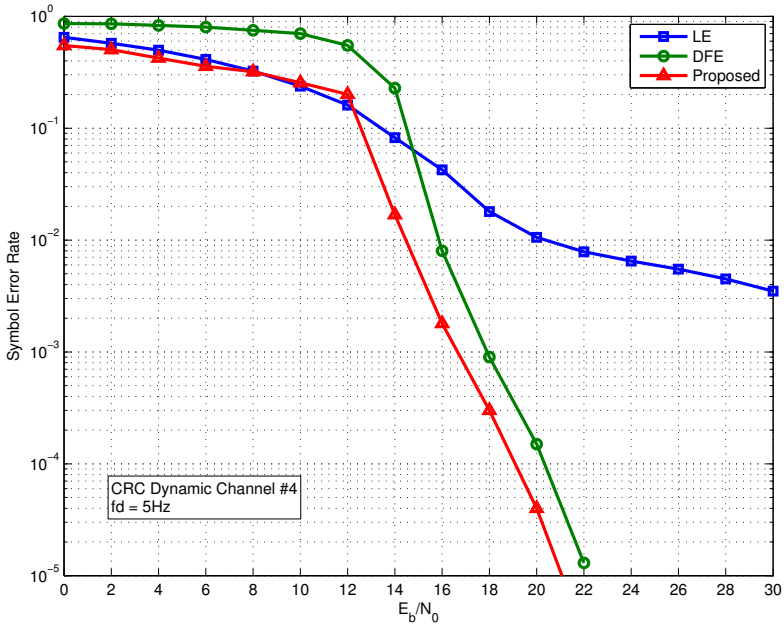

Figure 4.5: SER performance of the DFSF-BP detector in CRC #4 channel with 5Hz Doppler shift

which only cancels the ISI from less significant taps, and leaves the ISI from the dominant taps for the BP detector, there is less chance to introduce unintended ISI in the feedback loop, and thus the impact of error propagation to the overall system is reduced. To address the Doppler shift effect in the practical ATSC channels, the DFSF was designed adaptively using the LMS algorithm to track the time-varying channel. The simulation results under both static and dynamic channels demonstrated that the proposed scheme outperforms the traditional DFE in terms of symbol error rate.

Chapter 5

Symbol Detectors for Cooperative Communications

In this chapter, we study the detector design for cooperative communications in ISI channels. The background of cooperative communications is introduced in Section 5.1. A novel system framework employing non-orthogonal amplify-and-forward half-duplex relays through ISI channels is described in Section 5.2. We first consider an optimal detector that consists of a whitening filter and a MLSE in Section 5.3, which, however, has practical issues of high complexity if the relay period is long. Next we study three other feasible detectors to deal with long relay period. Section 5.4 presents a multitrellis Viterbi algorithm (MVA) based detector that reduces the complexity significantly but still achieves near optimal performance. Two alternatives which are more practically implementable, decision feedback sequence estimation (DFSE) based detector and BP-based detector, are shown in Section 5.5 and Section 5.6 respectively. Simulation results and complexity/performance analysis are given in Section 5.7, followed by the conclusions in Section 5.8.

5.1 Introduction

Historically, multipath fading has been combated by using time and frequency diversity techniques. In the last fifteen years, results in information theory have shown that spatial diversity can yield significant gains in the spectral efficiency and power efficiency of point-to-point multiple-antenna communication (MIMO) systems [50,51]. The transition from theory to practice has largely taken place with MIMO technology, as many modern consumer wireless standards exploit MIMO technology. To realize such gains, however, it is necessary that each of the paths between transmit and receive antennas is uncorrelated. For such an assumption to be valid, it is typically required that the antenna elements are spaced at least a half carrier wavelength apart [52], and perhaps even more in environments with minimal scattering. In many scenarios, however, it may not be practical for size-constrained nodes to have even two antennas with sufficient spacing between them. Furthermore, for each antenna that is added to the node, a complete RF front end must be added. There may be cost and power constraints that preclude the inclusion of multiple antennas, as the RF portion of a communications system often accounts for the majority of the cost and power.

More recently, cooperative diversity [53,54] and relay networks [55] have attracted a lot of attention for their ability to exploit increased spatial diversity available at distributed antennas on other nodes in the system. By intelligent cooperation among nodes in the network which may only have a single antenna, a virtual multiple antenna system can be formed. Indeed, information theoretic results demonstrate that some of the loss associated with using only a single antenna can be recuperated by using intelligent cooperation among distributed nodes [56, 57].

While communication via cooperative relays has seen a lot of active research interest in recent years, most of the existing work has largely come from the information theory and coding communities. While there are a few exceptions, e.g. [58, 59], little research has yet been conducted into the implementation issues of relaying and cooperation. As such, the majority of works in the field of cooperative diversity assume that receivers employ optimal detectors.

In this chapter, we set out to investigate detector design for half-duplex relays in frequency selective fading channels encountered in practice. While a variety of forwarding protocols have been previously proposed, we will consider amplify-and-forward (AF) for its simplicity and reduced implementation costs. Frequency selective fading channels are an inevitable impairment in wideband communication systems, and such channels cause the receiver to observe the superposition of multiple delayed reflections of the transmitted signal, resulting in ISI. Even in channels which do not exhibit significant time dispersion, the non-orthogonal AF relay itself effectively introduces ISI since the destination observes a superposition of the source and relayed signals. As our focus is on the complexity of the detector itself, we do not treat the performance gains possible with relays as this has been demonstrated elsewhere [60]. Similarly, with our focus on the complexity of detector implementation, we do not address the problem of channel estimation, and thus consider the optimistic case where the detector has perfect channel knowledge.

5.2 Cooperative Communication System Model

A basic model of a three-node relay system model is shown in Fig. 5.1. Both source and relay can be considered as mobile users, and each has only one antenna. The

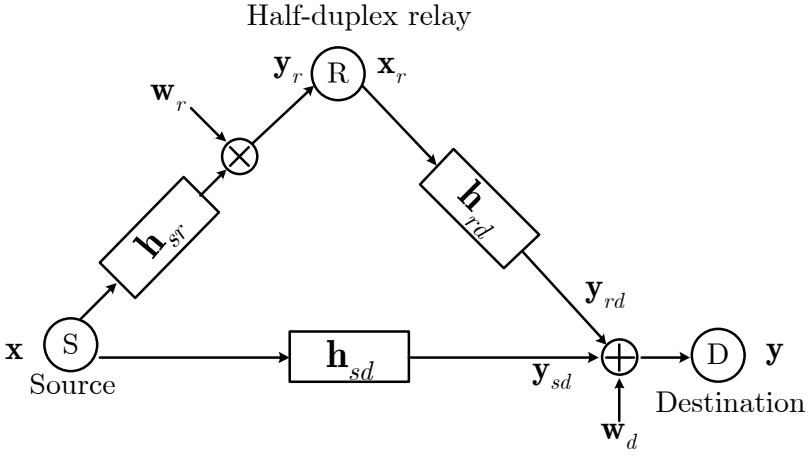

Figure 5.1: System model with one half-duplex relay.

source is attempting to send a message to the destination. Due to the broadcast nature of wireless communications, however, the relay receives transmissions from the source that are intended for the destination; therefore, the relay can assist by forwarding additional copies of these transmissions to the destination. Since the channels from source and relay to destination are statistically independent, the three-node cooperative communication scheme effectively forms spatial diversity. We consider a system where a source transmits a continuous stream of data to a destination, and a simplistic AF relay assists the source by amplifying and forwarding the data to the destination. We do not assume the relay has performed any synchronization with the destination, and so the relay forwards information to the destination in an open-loop fashion.

Relays have mainly two types: full duplex relays that can transmit and receive simultaneously, and half-duplex relays that can either transmit or receive in any time slot. Since full duplex relays are difficult to implement due to self interference

which occurs when both transmit and receive operations are in the same band, half-duplex is considered more practical for cooperative communication systems. In our system model, the half-duplex relay period T is a parameter which defines the frame structure where the relay receives for T symbol periods, and then transmits for T symbol periods. The relay repeats these two tasks alternately. The source and relay are assumed to transmit on the same channel, employing the so-called non-orthogonal amplify-forward protocol (NAF) [60].

The source sends the symbols $\mathbf{x} = [x[0], x[1], \ldots x[N-1]]^T \in \mathbb{C}^N$ at a symbol rate of f, where N is the number of transmitted symbols. We assume that a SRRC filter with the impulse response $h_{Tx}(t)$ is applied as the transmitter filter. The received signal is processed by a matched filter with the impulse response $h_{Rx}(t) = h_{Tx}(-t)$, and then sampled at the symbol rate of f. The equivalent discrete-time channel impulse responses [23] which include the effect of pulse shaping are denoted by $\mathbf{h}_{sd}$, $\mathbf{h}_{sr}$, $\mathbf{h}_{rd}$ for the source-destination, source-relay, and relay-destination channels, respectively, and they have corresponding channel lengths L_{sd}, L_{sr} and L_{rd} (e.g. $\mathbf{h}_{sd} = [h_{sd}[0], h_{sd}[1], \ldots, h_{sd}[L_{sd}-1]]^T$). The signals $\mathbf{w}_r$ and $\mathbf{w}_d$ are complex AWGN at the relay and the destination with variances σ_r^2 and σ_d^2, respectively.

The destination receives the superposition of the two signals from the source and the relay, and the received signal can be expressed as

$$\mathbf{y} = \mathbf{y}_{sd} + \mathbf{y}_{rd} + \mathbf{w}_d$$

where $\mathbf{y}_{sd} \in \mathbb{C}^{N+L_{sd}-1}$ is the contribution from the source and $\mathbf{y}_{rd} \in \mathbb{C}^{N+L_{sr}+L_{rd}-2}$ is the contribution from the relay.

We first consider the source-destination link. The contribution from source to

destination is written as

$$\mathbf{y}_{sd} = \mathbf{H}_{sd}\mathbf{x} \tag{5.1}$$

where $\mathbf{H}_{sd} \in \mathbb{C}^{(N+L_{sd}-1)\times N}$ is the complex Tplitz channel convolution matrix whose entries are defined by

$$[\mathbf{H}_{sd}]_{i,j} = \begin{cases} h_{sd}[i-j] & 0 \leq i-j \leq L_{sd}-1 \\ 0 & \text{otherwise} \end{cases}$$

where $1 \leq i \leq N + L_{sd} - 1$ and $1 \leq j \leq N$, i.e.

$$\mathbf{H}_{sd} = \begin{bmatrix} h_{sd}[0] & 0 & 0 & 0 & 0 & 0 & \\ h_{sd}[1] & h_{sd}[0] & 0 & 0 & 0 & 0 & \\ h_{sd}[2] & h_{sd}[1] & h_{sd}[0] & 0 & 0 & 0 & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ h_{sd}[L_{sd}-1] & \cdots & & h_{sd}[1] & h_{sd}[0] & 0 & \\ 0 & h_{sd}[L_{sd}-1] & \cdots & & h_{sd}[1] & h_{sd}[0] & \cdots \\ \vdots & \vdots & & & & \vdots & \ddots \end{bmatrix}.$$

For the source-relay-destination link, the corresponding contribution is given by

$$\begin{aligned} \mathbf{y}_{rd} &= \mathbf{H}_{rd}\mathbf{x}_r \\ &= \mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{y}_r \\ &= \mathbf{H}_{rd}\mathbf{\Gamma}(\mathbf{H}_{sr}\mathbf{x} + \mathbf{w}_r). \end{aligned} \tag{5.2}$$

The Tplitz channel matrices $\mathbf{H}_{rd} \in \mathbb{C}^{(N+L_{sr}+L_{rd}-2)\times(N+L_{sr}-1)}$ and $\mathbf{H}_{sr} \in \mathbb{C}^{(N+L_{sr}-1)\times N}$ are defined in the same way as $\mathbf{H}_{sd}$, $\mathbf{y}_r \in \mathbb{C}^{N+L_{sr}-1}$ is the signal received by the relay, $\mathbf{x}_r \in \mathbb{C}^{N+L_{sr}-1}$ is the signal transmitted from the relay, and $\mathbf{\Gamma} \in \mathbb{C}^{(N+L_{sr}-1)\times(N+L_{sr}-1)}$ is a fixed matrix described below. Note that for the matrix dimensions to be compatible, we require that $L_{sd} = L_{sr} + L_{rd} - 1$; if this is not satisfied, we can append

zeros to the appropriate matrix without loss of generality. The function of $\mathbf{\Gamma}$ is to impose the half-duplex constraint by selecting groups of T symbols from $\mathbf{y}_r$ (receiving), scaling these symbols by a factor β (amplifying), and then delaying the scaled symbols of $\mathbf{y}_r$ for transmission in the next T symbol block (forwarding). The value of β is typically chosen to satisfy an average power constraint at the relay by choosing $\beta = \sqrt{\frac{P_r}{\|\mathbf{h}_{sr}\|^2 P_s + \sigma_r^2}}$ where P_s and P_r are the source power and relay power respectively. The constant matrix $\mathbf{\Gamma}$ is given by

$$\mathbf{\Gamma} \triangleq \beta \, \mathbf{I}_{\left(\frac{N+L_{sr}-1}{2T}\right)} \otimes \begin{bmatrix} \mathbf{0}_{T\times T} & \mathbf{0}_{T\times T} \\ \mathbf{I}_{T\times T} & \mathbf{0}_{T\times T} \end{bmatrix} \tag{5.3}$$

where $\otimes$ denotes Kronecker product. Here we implicitly require that $N + L_{sr} - 1$ be divisible by $2T$. As an example, when $T = 2$, the signals received and transmitted by the relay are shown in Fig. 5.2, where the first eight time periods are considered, and the shadow indicates time period where the relay cannot receive because it is transmitting, or vice versa. For this example,

$$\mathbf{\Gamma} = \beta \, \mathbf{I}_2 \otimes \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \beta \begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \end{bmatrix} \tag{5.4}$$

such that $\mathbf{x}_r = \Gamma \mathbf{y}_r$ as in Fig. 5.2.

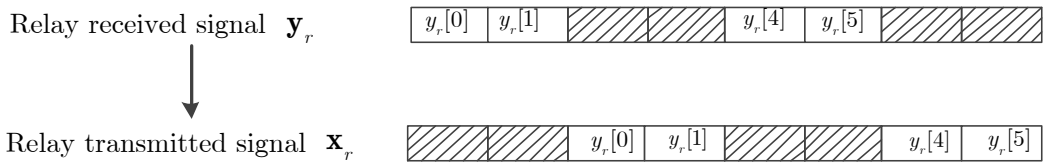

Figure 5.2: Signals received and transmitted by the relay when $T = 2$

From (5.5), (5.1) and (5.2), the received signal at the destination is expressed as

$$\begin{aligned} \mathbf{y} &= \overbrace{\mathbf{H}_{sd}\mathbf{x}}^{\text{from source}} + \overbrace{\mathbf{H}_{rd}\mathbf{\Gamma}(\mathbf{H}_{sr}\mathbf{x}+\mathbf{w}_r)}^{\text{from relay}} + \mathbf{w}_d \\ &= \underbrace{(\mathbf{H}_{sd}+\mathbf{H}_{rd}\Gamma\mathbf{H}_{sr})}_{\triangleq\tilde{\mathbf{H}}}\mathbf{x} + \underbrace{(\mathbf{H}_{rd}\Gamma\mathbf{w}_r+\mathbf{w}_d)}_{\triangleq\tilde{\mathbf{w}}} \end{aligned} \tag{5.5}$$

where $\tilde{\mathbf{w}} \sim \mathcal{CN}(\mathbf{0}, \sigma_d^2\mathbf{I} + \sigma_r^2\mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{\Gamma}^H\mathbf{H}_{rd}^H)$. Note that $\tilde{\mathbf{w}}$ is colored, not white, since the AWGN on the source-relay link is amplified-and-forwarded over the relay-destination ISI channel which colors the noise. Additionally, from (5.3) and (5.4), we see that the relay matrix $\mathbf{\Gamma}$ has a repetitive structure with a period of $2T$. Accordingly, the channel matrix $\tilde{\mathbf{H}}$ shows the same structure as $\mathbf{\Gamma}$. Consequently $\tilde{\mathbf{H}}$ can be interpreted as a periodically time-varying FIR channel which consists of $2T$ sets of different channel coefficients. In summary, equation (5.5) allows us to describe the input-output behavior of the system with a linear equation. While the constituent channels themselves are not time-varying, the effective impulse response of the overall system is indeed time-varying due to the on/off behavior of the relay.

5.3 Maximum-Likelihood Detector

Assuming that receivers can acquire perfect channel knowledge, MLSE can be employed to combat the ISI by searching for the minimum Euclidean distance between

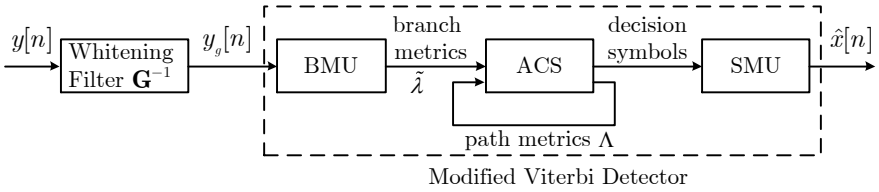

Figure 5.3: Block diagram of the proposed ML detector.

observed signal and any given transmitted signals [6]. The Viterbi algorithm [61] is an efficient technique for solving the minimum distance problem, and its implementation has been investigated extensively [30–33, 62]. The traditional Viterbi algorithm as proposed in [61] is directly applicable only to time-invariant channels. A modified Viterbi detector is proposed to address the periodically time-varying *effective* channel induced by the half-duplex relay. Furthermore, since the minimum Euclidean distance is not optimal in the presence of colored Gaussian noise, we employ a whitening filter before detection, which is optimal as shown in [6]. The block diagram of our design is given in Fig. 5.3.

To whiten the noise, spectral factorization of the composite noise covariance must be performed. We factor the noise covariance matrix as:

$$\mathbf{G}\mathbf{G}^H = \sigma_d^2\mathbf{I} + \sigma_r^2\mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{\Gamma}^H\mathbf{H}_{rd}^H \tag{5.6}$$

which can be accomplished by taking $\mathbf{G}$ to be the Cholesky factorization of the covariance. We note that the Cholesky factorization is not the only such factorization $\mathbf{G}$, as the factorization in (5.6) is not unique. By filtering the received signal with $\mathbf{G}^{-1}$ (i.e. by forming $\mathbf{G}^{-1}\mathbf{y}$), the noise becomes whitened since the covariance of the

filtered noise $\mathbf{G}^{-1}\tilde{\mathbf{w}}$ is given by

$$\begin{aligned} E\left[(\mathbf{G}^{-1}\tilde{\mathbf{w}})(\mathbf{G}^{-1}\tilde{\mathbf{w}})^H\right] &= (\mathbf{G}^{-1})E[\tilde{\mathbf{w}}\tilde{\mathbf{w}}^H](\mathbf{G}^{-1})^H \\ &= (\mathbf{G}^{-1}\mathbf{G})(\mathbf{G}^{-1}\mathbf{G})^H \\ &= \mathbf{I}. \end{aligned}$$

We note that noise covariance matrix in (5.6) is positive definite, so the inverse of $\mathbf{G}$ always exists. Ignoring end effects (or, equivalently, taking the block length $N \to \infty$), $\mathbf{G}^{-1}$ follows the same repetitive structure as $\mathbf{\Gamma}$, and thus also exhibits the periodically time-varying property.

After applying the whitening filter to the received signal (5.5), the whitening filter output becomes

$$\begin{aligned} \mathbf{y}_g &= \overbrace{\mathbf{G}^{-1}\tilde{\mathbf{H}}}^{\mathbf{H}_{eff}}\mathbf{x} + \overbrace{\mathbf{G}^{-1}\tilde{\mathbf{w}}}^{\mathbf{w}_{eff}} \\ &= \mathbf{H}_{eff}\mathbf{x} + \mathbf{w}_{eff} \end{aligned} \tag{5.7}$$

where $\mathbf{w}_{eff}$ is now *white* Gaussian noise. Note that the effective whitened channel matrix $\mathbf{H}_{eff}$ maintains the periodically time-varying property due to the similar structures of $\mathbf{G}^{-1}$ and $\tilde{\mathbf{H}}$.

The structure of the effective channel matrix $\mathbf{H}_{eff}$ is given in Fig. 5.4 where we see that the matrix has a block Tplitz structure with rows repeating every multiple of $2T$. It defines $2T$ sets of effective channel coefficients as $\mathbf{h}_0$, $\mathbf{h}_1$, ..., $\mathbf{h}_{2T-1}$, where $\mathbf{h}_0 = [h_0[0],\ h_0[1],\ ...,\ h_0[L-1]]^T$, $\mathbf{h}_1 = [h_1[0],\ h_1[1],\ ...,\ h_1[L-1]]^T$, ..., and L is the effective channel length. The effective length L may be significantly extended by the delay introduced by the relay, as well as the group delay introduced by the whitening

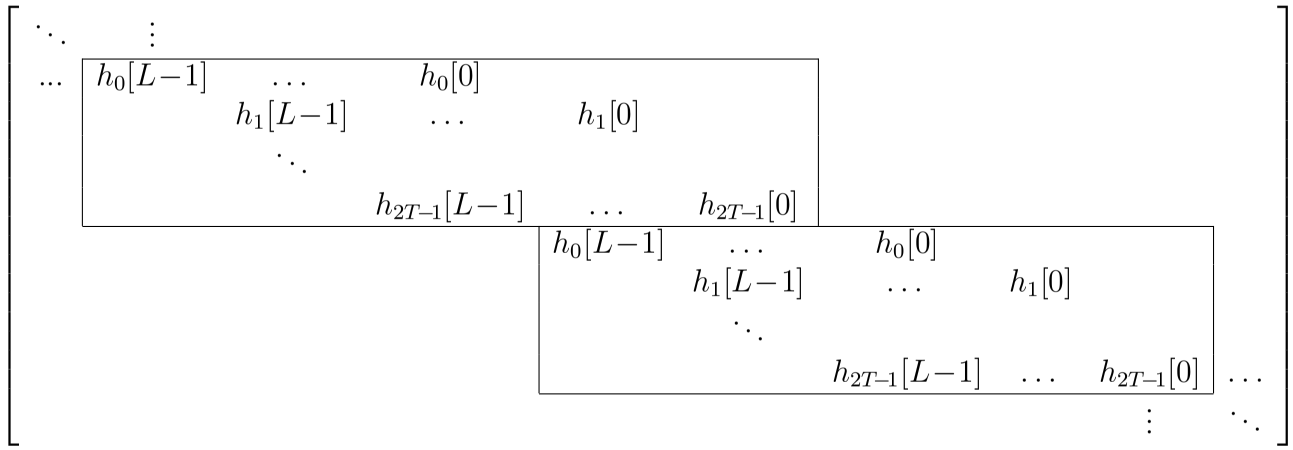

Figure 5.4: Structure of the effective channel matrix

filter. We can determine the lower bound of the effective channel length in terms of the constituent channel lengths and the relay period as

$$L \geq \max(L_{sd},\, L_{sr} + L_{rd} + T - 1). \tag{5.8}$$

At time n, the corresponding coefficients of the periodically time-varying effective channel are

$$\mathbf{h}_n = \mathbf{h}_{\mathrm{mod}(n,2T)} \in \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{2T-1}\}, \quad n = 0, 1, \ldots$$

where $\mathrm{mod}(\cdot)$ is the modulus. The ideal output of the whitening filter at time n is then given by

$$s[n] = \sum_{i=0}^{L-1} h_n[i]x[n-i], \quad n = 0, 1, \ldots$$

which is simply the convolution of the source symbols with the periodically time-varying channel coefficients. Thus, the system model for relay-aided transmission through ISI channels reduces to a classical MLSE problem, with the additional twist that the effective channel is periodically time-varying. As the true output is of course corrupted by AWGN, the maximum likelihood detector for estimating the source symbols $\mathbf{x}$ from (5.7) can be accomplished most efficiently with the Viterbi algorithm.

For the branch metric unit (BMU) in Fig. 5.3, the branch metrics along the trellis path are not only related to state transitions but also the current time instant. The branch metric calculation is modified as

$$\tilde{\lambda}[n] = |y_g[n] - \tilde{s}[n]|^2, \quad n = 0, 1, \ldots$$

where $y_g[n]$ is the signal from the whitening filter at the time instant n, $\tilde{s}[n]$ and $\tilde{\lambda}[n]$ are the estimated output signal and the corresponding branch metric the time instant

n, respectively.

The add-compare-select (ACS) unit in Fig. 5.3 recursively computes path metrics and decision bits,

$$\Lambda^{(j)}[n] = \min_i \left(\Lambda^{(i)}[n-1] + \tilde{\lambda}^{(i,j)}[n]\right), \quad n = 0, 1, \ldots$$

where $\Lambda^{(j)}[n]$ denotes the path metric at state j at the time instant n, and $\tilde{\lambda}^{(i,j)}[n]$ is the branch metric from state i to state j at instant n. The path metrics for each state are updated for the next iteration, and the decision indicating the survivor path for state j is recorded and retrieved from the survivor-path memory unit (SMU) in order to estimate the transmitted symbols along the final survivor path.

Similar to the traditional MLSE, the implementation cost of the ML detector for relay networks increases exponentially with respect to the effective channel length. Furthermore, the overhead of the proposed detector when compared with the traditional MLSE comprises the whitening filter, the multiplexers in the BMU, and additional control logic to account for the periodically time-varying effective channel [63]. As indicated in (5.8), the effective channel length increases with the relay period T. In cooperative relay systems, the relay period T is likely chosen to be very long, possibly spanning hundreds of symbols, so that the relay is not required to switch frequently between transmit and receive modes. When the relay period T is large, however, an implementation of the Viterbi algorithm based optimal detector becomes not practical.

We simulate the bit error rate (BER) performance of the ML detector with the following parameters: the transmitted BPSK signal consists of i.i.d. unit-power symbols $x[n] = \{\pm 1\}$, the relay transmitting power is $P_r = 1$. The SNR values $E_b/N_o^{(d)} = 1/\sigma_d^2$

and $E_b/N_o^{(r)} = 1/\sigma_r^2$ denote bit-energy-to-noise ratio for the destination and the relay, respectively. Unless otherwise specified, we assumed $E_b/N_o^{(r)} = E_b/N_o^{(d)} + 10dB$, which represents a scenario where the source-relay link is better, on average, than the source-destination link.

We assume the SRRC pulse shaping filter is employed at both the transmitting and receiving ends. The SRRC filter is truncated to $[-2/f, 2/f]$ with a roll-off factor 0.5, where $f = 5$ MHz is the symbol rate. We simulate over 200 fading realizations in the ITU-R 3G indoor office test environment [41] with 6 independent channel paths. The time delay relative to the first path is $[0, 50, 110, 170, 290, 310](ns)$, and the average power relative to the strongest path is $[0, -3, -10, -18, -26, -32](dB)$. As a practical matter, it is possible for the whitening filter to be quite long, depending on the noise covariance matrix. In these simulations, we truncate the whitening filter to have $L = \max(L_{sd},\, L_{sr} + L_{rd} + T - 1)$ taps.

We consider the effect of the number of independent paths on system performance. While additional ISI is often viewed as an impairment to reliable communication, the additional fading paths result in increased diversity, and hence increased BER performance through the cooperative relay. This effect is observed in Fig. 5.5, where we consider three cases. Specifically, we truncate the ITU-R 3G indoor office channel so that it only uses either the first two paths, the first three paths, or all the six paths. We note that the BER performance with respect to $E_b/N_o^{(d)}$ improves as the number of independent paths increases.

We next consider the effect of the relay cooperation period T. Recall that the relay receives T symbols, and then amplifies and retransmits those T symbols. As shown in Fig. 5.6 for an uncoded system, better performance is obtained when choosing a smaller relay period. If the relay period T is much larger than the channel delay

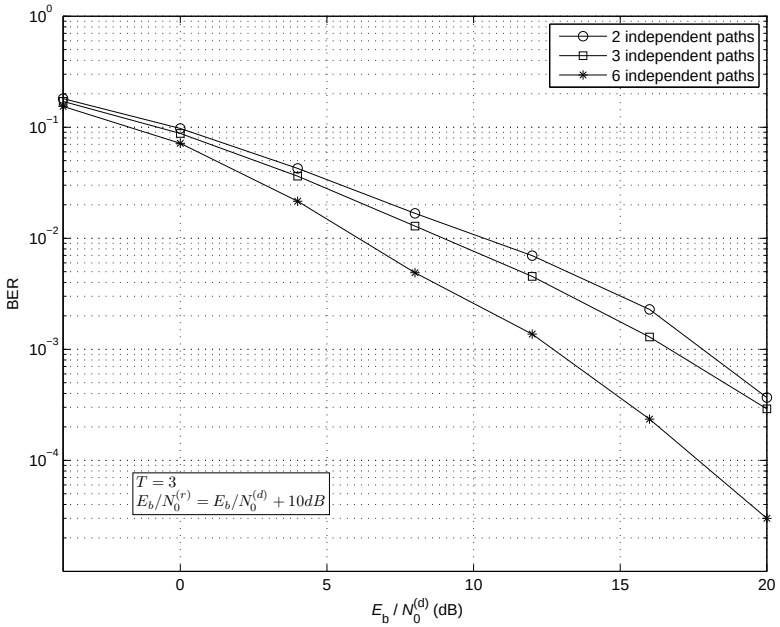

Figure 5.5: BER performance of the ML detector for different channel length.

spread, the diversity technique of using cooperative relays becomes less effective since there is almost no overlap between the symbols directly received from the source and forwarded by the relay.

The effect of the whitening filter is illustrated in Fig. 5.7 where $T = 3$. As we expect, the BER performance with the whitening filter is better than the performance without it. It is also observed that if the noise on the source-relay link is small relative to the noise on the source-destination link so that $\sigma_d \gg \sigma_r$, the whitening filter does little to help. The reason for this is that the noise looks approximately white when the noise on the source-destination link dominates since $\sigma_d^2\mathbf{I} + \sigma_r^2\mathbf{H}_{rd}\mathbf{\Gamma}\mathbf{\Gamma}^H\mathbf{H}_{rd}^H \approx \sigma_d^2\mathbf{I}$. Thus, in situations where the source-relay link is particularly good, it may be possible to reduce complexity without much performance penalty by removing the whitening filter.

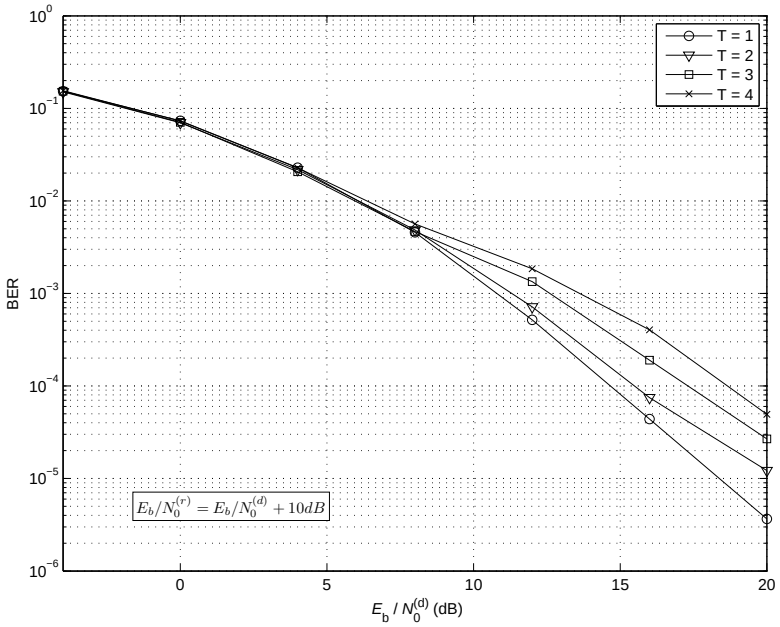

Figure 5.6: BER performance of the ML detector for different relay period T.

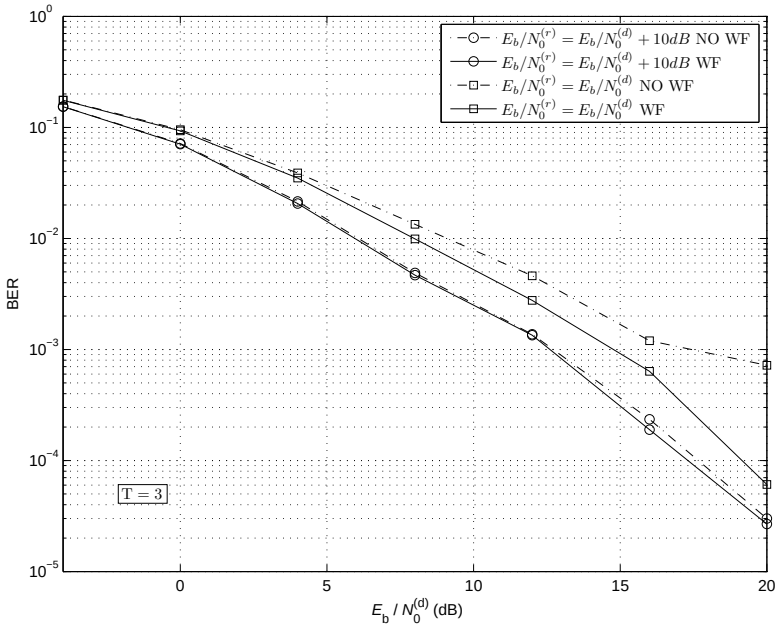

Figure 5.7: BER performance of the ML detector with and without whitening filter (WF).

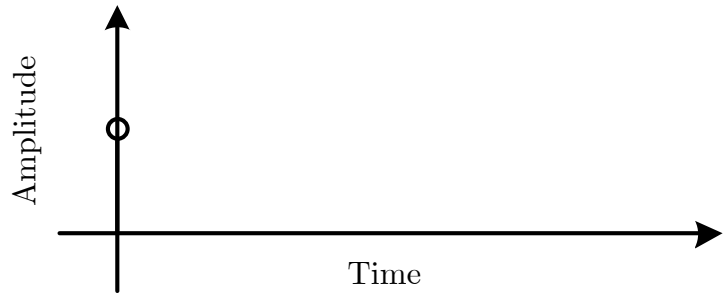

Figure 5.8: Effective channel impulse response in non-ISI channels during relay reception.

5.4 MVA Detector for Long Relay Periods

Consider the practical effect of the relay period T in a simplified relay network with only AWGN channels and no ISI so that $L_{\{sd,sr,rd\}} = 1$. When the relay is in the receiving period, the destination only receives the signal from the source. The channel impulse response at the destination is shown in Fig. 5.8. After T time instants, the relay begins to forward the copy of the signal to the destination. The effective channel impulse response as seen at the destination during relay transmission is depicted in Fig. 5.9. We observe that the relay actually introduces ISI even in the non-ISI channels, and the effective channel impulse response alternates between that shown in Fig. 5.8 and Fig. 5.9. Also, large values of T will increase the number of zero coefficients in the channel impulse response, which makes the effective channels become sparse.

Viterbi algorithms for sparse channels have been investigated by several independent researchers. The parallel trellis Viterbi algorithm (PTVA) proposed in [11]

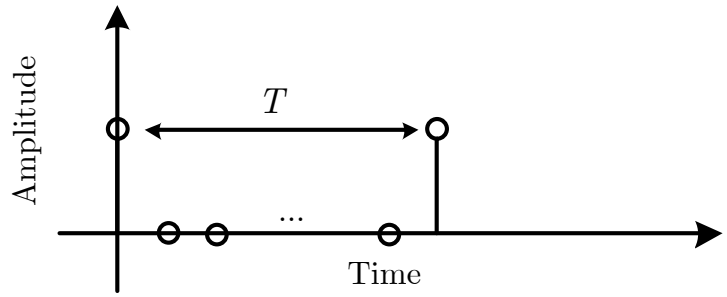

Figure 5.9: Effective channel impulse response in non-ISI channels during relay transmission.

reformulated the original single trellis into a set of independent trellises. These independent trellises operate in parallel and have less overall complexity than a single trellis. The PTVA requires that the sparse channel have equi-spaced coefficients, however, which usually cannot be satisfied in practice. Although a generalized PTVA is given to deal with general sparse channels, its performance loss is remarkable if the channel taps are not well-approximately by an equi-spaced structure.

A multitrellis Viterbi algorithm (MVA) was proposed in [12] for near-optimal detection in sparse ISI channels. For the MVA, the complexity does not depend on the channel impulse response length but only on the number of non-zero coefficients. In order to process the sparse time-varying channels for relay networks, the MVA is modified and incorporated in our MVA-based ML detector.

To illustrate the operation of the MVA, we begin by considering an example. Assume $\mathbf{h}_n \in \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{2T-1}\}$ has only a few non-zero coefficients, e.g. $h_n[i] \neq 0$ for $i = 0, K, L-1$, $(0 < K < L-1)$. The ideal (noiseless) output signal at the time

Table 5.1: The dependencies between $x[0]$ and related output signal.

Related output signal	Symbols to be recorded
$s[0] = f(\mathbf{h}_0, x[0], x[-K], x[-L+1])$	$x[0]$
$s[K] = f(\mathbf{h}_K, x[K], x[0], x[K-L+1])$	$x[0], x[K]$
$s[L-1] = f(\mathbf{h}_{L-1}, x[L-1], x[L-1-K], x[0])$	$x[0], x[K], x[L-1]$
$s[2K] = f(\mathbf{h}_{2K}, x[2K], x[K], x[2K-L+1])$	$x[K], x[L-1], x[2K]$
$s[K+L-1] = f(\mathbf{h}_{K+L-1}, x[K+L-1], x[L-1], x[K])$	$x[K], x[L-1], x[2K], x[K+L-1]$
$s[2L-2] = f(\mathbf{h}_{2L-2}, x[2L-2], x[2L-2-K], x[L-1])$	$x[L-1], x[2K], x[K+L-1], x[2L-2]$
$s[3K] = f(\mathbf{h}_{3K}, x[3K], x[2K], x[3K-L+1])$	$x[2K], x[K+L-1], x[2L-2]$
$s[2K+L-1] = f(\mathbf{h}_{2K+L-1}, x[2K+L-1], x[K+L-1], x[2K])$	$x[2K], x[K+L-1], x[2L-2]$
$s[K+2L-2] = f(\mathbf{h}_{K+2L-2}, x[K+2L-2], x[2L-2], x[K+L-1])$	$x[K+L-1], x[2L-2]$

n is given by

$$s[n] = h_n[0]x[n] + h_n[K]x[n-K] + h_n[L-1]x[n-L+1].$$

When $x[0]$ is to be estimated, we can see that $x[0]$ is needed in $s[0]$, $s[K]$, $s[L-1]$. Next, in $s[K]$, a new symbol $x[K]$ appears and it is also needed in $s[L-1]$ and $s[2K]$. In this way, we record all the output signals and symbols related with $x[0]$ in Table 5.1, and we use the notation of $f(\cdot)$ to indicate the dependency of outputs on input symbols and channels. Note that some output signals and symbols are not needed when $x[0]$ is under detection; for example, if $K \neq 1$, there is no need to record $s[1]$, since $x[1]$, $x[1-K]$, $x[2-L]$ in $s[1]$ do not affect the estimation of $x[0]$. With the traceback length $L_{tb} = 3(L-1)$, for example, the estimation of $x[0]$ depends on $x[K]$, $x[L-1]$, $x[2K]$, $x[K+L-1]$, $x[2L-2]$ by a non-instantaneous relationship , assuming that $x[n]$, $n < 0$ are known.

We note that some related symbols appear only once in the first column, e.g. $x[L-1-K]$, and do not need to be recorded. Its value can be determined by an instant decision given by

$$\hat{x}[L-1-K] = \underset{\tilde{x}[L-1-K]}{\arg\min} |y_g[L-1] - \tilde{s}[L-1]|$$

where $\tilde{s}[L-1] = f(\mathbf{h}_{L-1}, x[L-1], \tilde{x}[L-1-K], x[0])$, and $x[L-1]$ and $x[0]$ are known for a given state.

When two or more symbols are determined by the instant decision, e.g. $x[3K]$ and $x[3K-L+1]$ in $s[3K]$, the estimation is given by

Table 5.2: State definition ($x[0]$ under estimation).

Related time instant	State definition
0	$[x[0]]$
K	$[x[K], x[0]]$
$L-1$	$[x[L-1], x[K]]$
$2K$	$[x[2K], x[L-1], x[K]]$
$K+L-1$	$[x[K+L-1], x[2K], x[L-1]]$
$2L-2$	$[x[2L-2], x[L-1+K], x[2K]]$
$3K$	$[x[2L-2], x[L-1+K], x[2K]]$
$2K+L-1$	$[x[2L-2], x[L-1+K], x[2K]]$
$K+2L-2$	$[x[2L-2], x[L-1+K], x[2K]]$

$$\{\hat{x}[3K], \hat{x}[3K-L{+}1]\} = \underset{\tilde{x}[3K], \tilde{x}[3K-L+1]}{\arg\min} |y_g[3K] - \tilde{s}[3K]|$$

where $\tilde{s}[3K] = f(\mathbf{h}_{3K}, \tilde{x}[3K], x[2K], \tilde{x}[3K-L{+}1])$, and $x[2K]$ is known for a given state.

The definition of state depends only on the related time instant. By the list of related symbols, the state definition is derived and given in Table 5.2. Note that the state definition excludes the symbols assumed to be known, i.e. $x[-K]$, $x[-L+1]$, $x[K-L+1]$, and the symbols that can be determined by the instant decision.

From Table 5.1 and 5.2, it is observed that the corresponding trellis shrinks in two dimensions, which leads to a significant reduction in computational complexity. Furthermore, the process of traceback is faster, since for some instant given the current state, the previous state can be obtained immediately without survivor path decisions. There are two categories for these instant tracebacks. First, the previous state definition is a subset of the current state definition. For example, the state at instant $2K$ is defined as $[x[2K], x[L-1], x[K]]$, while the previous state $[x[L-1], x[K]]$ at the instant $L-1$ can be obtained from the current state without the help of the

survivor path decision. Second, the state definition is the same for the current and previous state. For example, considering the instants $K+2L-1$, $2K+L-1$, $3K$ and $2L-2$, we can bypass the traceback from $K+2L-2$ to $2L-2$. Once the start state at the instant $K+2L-2$ is available, we can begin to traceback from the instant $2L-2$ at the same state.

When subsequent symbols are under detection, the structure of the trellis remains the same, except that the branch metric calculation for the first several instants are slightly different, since the initial symbols (e.g. $x[-K]$, $x[-L+1]$, $x[K-L+1]$ for $x[0]$) have been estimated. The available estimated symbols will be used in the calculation of output signal $s[n]$ when needed.

Due to the reduced-size trellis, the detector can be realized by utilizing L_{tb} trellises working in parallel to increase the throughput. The received signals $y_g[n]$ are filled in the L_{tb} trellises sequentially. At the instant $L_{tb}-1$, the first trellis is full and $x[0]$ is estimated. At the instant L_{tb}, the received signal $y_g[L_{tb}]$ is ready to fill in the first trellis, and also, $x[1]$ is available from the second trellis. Notice that these trellis are similar in structure, however, the channel coefficients used in the branch metric calculation are not the same at different instants. For example, in the first trellis, $\mathbf{h}_0, \mathbf{h}_K, \mathbf{h}_{L-1}, \ldots, \mathbf{h}_{K+2L-2}$ are used in sequence for each step, while in the second trellis, $\mathbf{h}_1, \mathbf{h}_{K+1}, \mathbf{h}_L, \ldots, \mathbf{h}_{K+2L-1}$ are used for each step. The corresponding channel coefficients for the received signals $y_g[n]$ are $\mathbf{h}_{\mathrm{mod}(n,2T)} \in \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{2T-1}\}$.

For general Viterbi detectors in M-ary modulation systems, complex multiplications dominate the computational cost. There are M^{L-1} states and each state corresponds to M complex multiplications for branch metric calculation. Then the estimated total computational cost is M^L. For the proposed MVA detector in the sparse channel with 3 nonzero coefficients, there are M^3 states at most, and the

corresponding computational cost for each trellis is M^4. Thus the estimated total computational cost for MVA detector is $L_{tb}M^4$. Furthermore, survivor path decisions are recorded in the SMU which requires a significant amount of memory. The memory cost in bits for the general Viterbi detector is $L_{tb}M^{L-1}\log_2 M$. For the proposed MVA detector, we do not need to record the survivor path decisions for all L_{tb} instants. Assuming that only L'_{tb} ($< L_{tb}$) instants are considered in each trellis, the memory size for each trellis is $L'_{tb}M^3\log_2 M$.

In general, the complexity of Viterbi detector is in $\mathcal{O}(M^L)$, where L is the channel length. In comparison, the complexity of the MVA detector does not depend on the channel length but on the number of nonzero coefficients. When instantaneous decisions are made appropriately so that the symbol dependency table simplifies [12], the computing and memory resource required for the MVA detector is in $\mathcal{O}(M^{L'})$, where L' is the number of non-zeros coefficients. Therefore, the MVA detector is a better solution for sparse channels.

To simulate the BER performance of the MVA-based detector, we set the channel length $L_{\{sd,rd\}} = 1$, $L_{sr} = 2$, so that each effective channel $\mathbf{h}_n$ has 3 non-zero coefficients and Table 5.1 can be applied directly. Without loss of generality, we assume that each coefficient is i.i.d. as $\mathcal{CN}(0,1)$. The performance is given in Fig. 5.10 with $T = 3$ and Fig. 5.11 with $T = 5$. It is shown that performance loss from the MVA-based detector is negligible, as was claimed previously in [12].

For the MVA-based detector, if the dependencies are appropriately simplified, it was shown that the complexity does not depend on the length of the effective CIR but only on the number of non-zero coefficients. However, the simplification of the dependencies is not straightforward for an effective CIR with an arbitrary structure of non-zero tap coefficients. Also, irregular sub-trellises increase the implementation

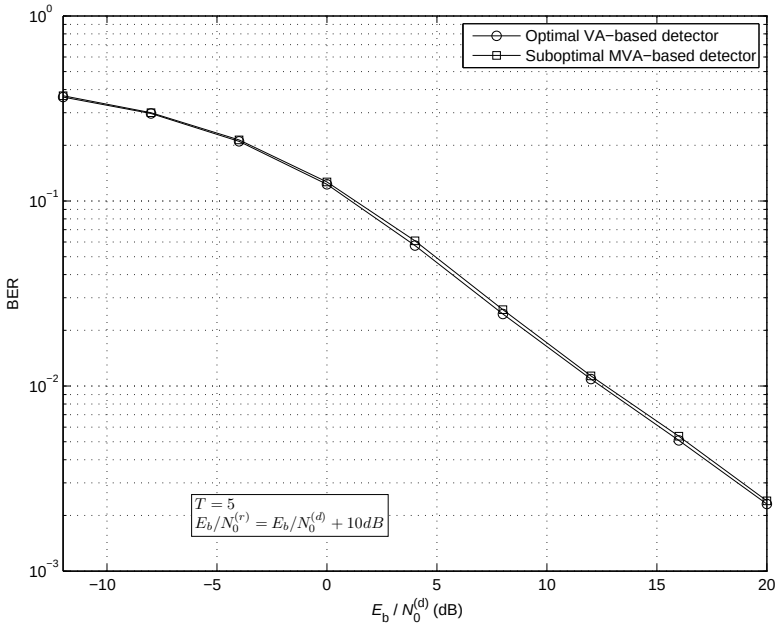

Figure 5.10: BER performance of the MVA-based detector $T = 3$.

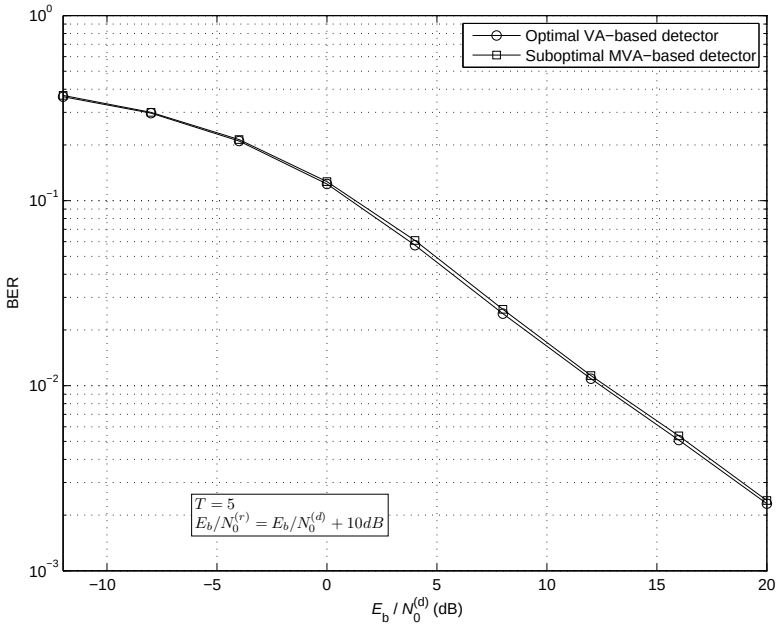

Figure 5.11: BER performance of the MVA-based detector $T = 5$.

overhead compared to a regular trellis as in traditional MLSE. Therefore, the MVA-based detector is not a strong candidate for cooperative ISI channels.

5.5 DFSE-Based Detector

In [13, 14], DFSE is proposed to incorporate the decision feedback mechanism within the calculation of branch metric directly in the Viterbi detector to enhance the reliability of the feedback decision.

In the trellis of DFSE, each state provides only partial information about the actual state of the channel. The states describe all possible values taken on by a small number $\mu < L$ of previous inputs. The required residual information is provided by a built-in decision feedback in branch metric computations. The traditional DFSE is modified and incorporated in the scenario of cooperative communications. In the branch metric calculation, the ideal received signal at time n is given by

$$s[n] = \sum_{i=0}^{\mu-1} h_n[i]x[n-i] + \underbrace{\sum_{i=\mu}^{L-1} h_n[i]\tilde{x}[n-i]}_{\text{Residual ISI}} \tag{5.9}$$

where $(x[n], \ldots, x[n-\mu+1])$ in the first sum are determined by the state in the trellis, and $(\tilde{x}[n-\mu], \ldots, \tilde{x}[n-L+1])$ in the second sum are the estimate of the partial state extracted from the survivor path leading to that state. This estimate is used to cancel ISI from symbols greater than μ symbol periods in the past. The residual ISI from the remaining $L-\mu$ symbols is canceled by the per-survivor decision feedback [64].

Note that (5.9) is modified to tackle the time-varying channel. The coefficients adopted by the decision feedback filter also change with a period of $2T$. The complexity can be managed by varying μ, the number of ISI symbol used in the trellis.

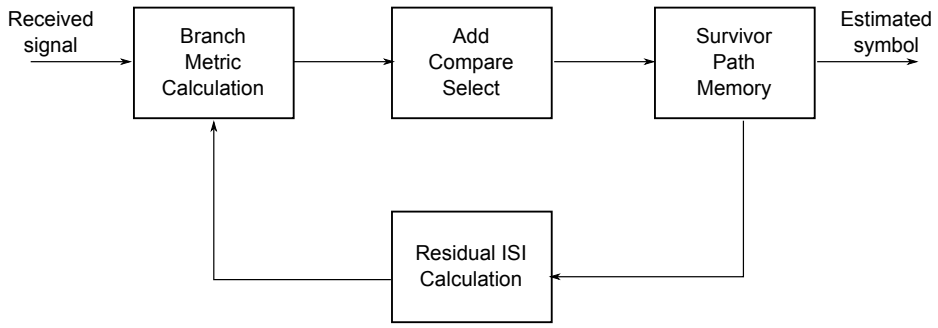

Figure 5.12: Block diagram of the proposed DFSE detector.

If $\mu = L$, the algorithm is equivalent to Viterbi algorithm; if $\mu = 1$, the algorithm reduces to the zero-forcing DFE. Choosing appropriate μ enables the DFSE to achieve a good tradeoff between performance and complexity.

The block diagram is shown in Fig. 5.12. The branch metric calculation unit calculates the Euclidean distance between the received signal and estimated signal along the trellis path. Then the add-compare-select unit recursively computes path metrics and decision bits. The path metric for each state is updated for the next iteration, and the decision indicating the survivor path are recorded and retrieved from the the survivor-path memory unit in order to estimate the transmitted symbols. In the meanwhile, the survivor path information leading to each state is extracted and feed back to the branch metric calculation unit.

The complex multipliers used in branch metric calculation and residual ISI calculation dominate the whole computational complexity. Consider a cooperative communication system using BPSK modulation. In the branch metric calculation, the number of states is $2^{\mu-1}$, and there are 2 branch metric calculations for each state. Thus the number of complex multipliers used in branch metric calculation is 2^{μ}. For the residual ISI calculation, there is a $(L-\mu)$-length decision feedback filter for each possible state transition, requiring $(L-\mu)2^{\mu}$ complex multipliers. The to-

tal computational complexity with respect to the number of complex multipliers is $(L-\mu+1)2^{\mu} \sim \mathcal{O}(2^{\mu})$.

5.6 Belief Propagation-Based Detector

DFSE can be applied as a suboptimal detector scheme for a wide class of long ISI channel in cooperative systems. However, it is interesting to point out that a large relay period T makes the effective CIR become sparse, which motivates us to consider using the BP detector as an alternative solution .

Unlike the factor graph of fixed channels, since the effective CIR in cooperative communication is periodically time-varying, the connections along the factor graph do not maintain the same pattern. Accordingly, the calculation of extrinsic information must adopt different coefficients depending on the node index n, i.e. the channel coefficients used in (2.2) are $\mathbf{h}_n = \mathbf{h}_{\mathrm{mod}(n,2T)} \in \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{2T-1}\}$, which are periodically time-varying. The implementation of the BP detector for the fixed channel coefficients has been elaborated in Section 2.4. We only need to send $\mathbf{h}_n$ instead of $\mathbf{g}'$ to the LPU (Fig. 2.8) to adapt for the cooperative ISI channels.

5.7 Simulation Results

We simulate the BER performance of the DFSE detector and BP detector with the following parameters: the transmitted BPSK signal consists of i.i.d. unit-power symbols $x[n] = \{\pm 1\}$, the relay transmission power is $P_r = 1$, and the symbol frame length $N = 500$. The SNR values $E_b/N_o^{(d)} = 1/\sigma_d^2$ and $E_b/N_o^{(r)} = 1/\sigma_r^2$ denote bit-energy-to-noise ratio for the destination and the relay, respectively. Unless otherwise

specified, we assume $E_b/N_o^{(r)} = E_b/N_o^{(d)} + 10\,\text{dB}$, which represents a scenario where the source-relay link is of higher quality, on average, than the source-destination link. We consider Rayleigh fading channels with $L_{\{sd,sr,rd\}} = 2$, and each coefficient is a zero-mean complex Gaussian random variable with unit average power. The individual channel coefficients are assumed to be statistically independent.

For the DFSE detector, $\mu = 2$ and $\mu = 4$ are considered. For the BP detector, we choose the number of iterations as 2 and 5. We also consider the MLSE as a reference curve. The BER performance for different detectors when the relay period $T = 5$ and $T = 10$ is shown in Fig. 5.13 and Fig. 5.14, respectively. We note that with these parameter choices, the effective CIR lengths for these two simulations are then $L = 8$ and $L = 13$, respectively, according to (5.8).

At a BER of 10^{-3}, when $T = 5$, the DFSE detector exhibits performance approximately 1.3 dB away from the optimal performance when $\mu = 2$, and a performance penalty of 1.2 dB when $\mu = 4$. Recall that when $\mu = L = 8$, the DFSE becomes the MLSE. The BP detector, on the other hand, performs only 0.5 dB away from optimal when 2 iterations are used, and only 0.1 dB away when 5 iterations are used. When $T = 10$, i.e. $L = 13$, the DFSE detector shows approximately 1.2 dB and 1dB away from the optimal performance for $\mu = 2$ and $\mu = 4$, respectively. While the BP detector performs 0.5 dB and 0.05 dB aways from optimal for 2 and 5 iterations, respectively.

Having presented the computational complexity and performance, we now consider practical implementation issues with respect to computing resources, latency, and performance tradeoffs. The parameters used in simulations in Section 5.7 are assumed here, i.e. $L_{\{sd,sr,rd\}} = 2$ and $T = 10$ so that $L = 13$.

When an optimal MLSE detector is considered as a reference, the number of

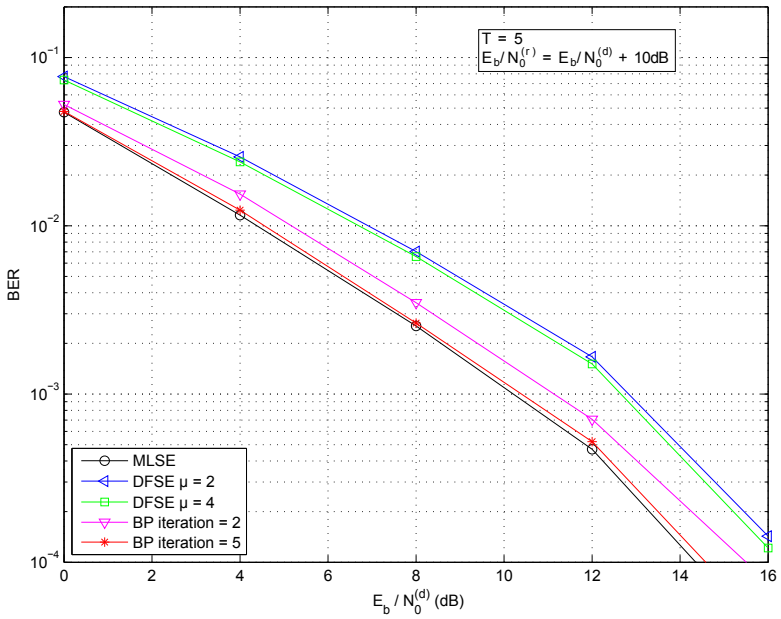

Figure 5.13: BER performance for different detectors when $T = 5$.

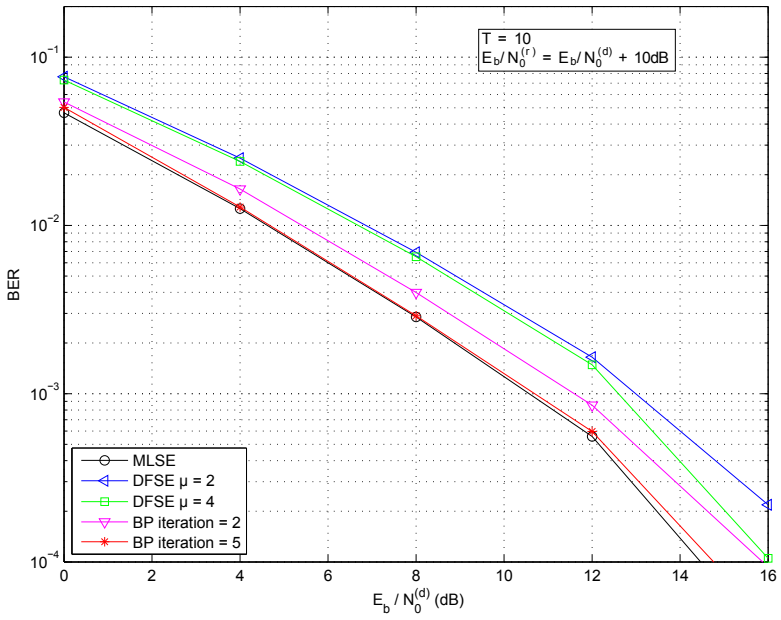

Figure 5.14: BER performance for different detectors when $T = 10$.

dominant computing elements (complex multipliers) is $2^L = 2^{13}$. Meanwhile, a DFSE-based detector consumes $(L - \mu + 1)2^\mu = 48$ elements when $\mu = 2$, and 160 elements when $\mu = 4$. For a BP-based detector, the number of non-zero coefficients is $L' = 6$, so it uses up to $2^{L'} = 64$ elements. The DFSE detector is the most cost-saving solution among the three when μ is small. The DFSE detector has a MLSE-like structure; thus the decoding latency, which is equal to the traceback length, is similar to MLSE. On the other hand, the latency of the BP detector is determined by the iteration number. A large number of iterations results in low throughput for the BP detector, rendering it unsuitable for use in high-rate applications. In some situations, however, the additional performance offered by increasing the number of iterations is often minimal. We also note that if the individual channel lengths are large, there will be a significant number of non-zero taps in the effective channel, and the BP detector may be too complex. In such circumstances, the effective channel can be first equalized to a sparse TIR by a partial response equalization before BP detection, which has been discussed in Section 2.3.1.

5.8 Conclusions

After developing a system model for the case of AF relays in ISI channels, we first presented an optimal ML detector realization based on the Viterbi algorithm; however, the implementation was shown to be limited by high computational complexity which increases exponentially with the length of the effective CIR. Since the duration of the effective CIR is extended as the relay period increases, the use of optimal ML detector becomes impractical even when the relay period has just a few symbol durations. By exploiting the sparse characteristic of the effective CIR, a MVA based

detector was proposed which decomposes the original trellis into multiple parallel irregular sub-trellises by investigating the dependencies between the received symbols. Although MVA provides near-optimal performance through simulations, it is not easy to implement for arbitrary ISI channels since the trellis decomposition is not straightforward in general. In order to achieve a balance between the computational complexity and performance, the DFSE detector and the BP detector were proposed for cooperative ISI channels. Traditionally these two detectors are used with fixed, static channels. In our model, however, the effective channel in the cooperative communications is shown be to periodically time-varying, even when the component channels themselves are static. Consequently, the DFSE detector and the BP detector were modified, as appropriate, to account for this. Through simulations in frequency selective fading channels, we demonstrated the uncoded performance of these detectors when compared to the optimal MLSE detector. In addition to quantifying the performance of these detectors, we also included an analysis of the implementation complexity as well as a discussion on complexity/performance tradeoffs.

Chapter 6

Conclusions

6.1 Summary of Results

This dissertation is devoted to exploring near-optimal and practically implementable symbol detectors in time domain for single-carrier transmissions over wireless ISI channels.

- Firstly, we present a partial response belief propagation (PRBP) symbol detector for sparse channels. The architecture is implemented by cascading a linear equalizer with an iterative BP detector. The channel is first partially equalized by the LE to a even more sparse target impulse response (TIR) with only a few nonzero coefficients remaining. The residual ISI is then canceled by the BP detector. With the cascaded LE-BP structure, the symbol detector is capable to achieve a near-optimal error rate performance with acceptable implementation complexity. Moreover, we present a pipeline high-throughput implementation of the detector for channel length 30 with QPSK modulation.

- Secondly, we investigate the performance of the PRBP detector under a more

generic ISI channel rather than the sparse channel. We compare the performance of the PRBP detector with the partial response maximum-likelihood (PRML) detector, which also employs a hybrid two-stage scheme to allow better performance/complexity tradeoff. Via simulations under the ITU-R 3G channels, it is shown that the PRBP detector provides superior performance over both the traditional minimum mean squared error linear equalizer and the PRML detector. Due to the effect of colored noise, the PRML detector in fading wireless channels is not as effective as it is in magnetic recording applications.

- Thirdly, an adaptive decision feedback sparsening filter BP (DFSF-BP) detector, which is also in the category of the PRBP detector, is proposed. Different from the aforementioned PRBP detector based on LE-BP structure, the BP detector is followed by a nonlinear DFSF as the partial response equalizer aiming to reduce the error propagation. The DFSF employs a modified feedback filter which leaves the strongest post-cursor ISI taps uncorrected. In order to deal with the Doppler fading of the ATSC channel, and also to facilitate the BP implementation, the DFSF is adapted using the LMS algorithm, such that the equalized sparse channel appear to be static during the second stage of BP detector. Simulation results show that the DFSF-BP detector outperforms the traditional DFE in symbol error rate, under both static channels and dynamic ATSC channels.

- Finally, we study the symbol detector design for cooperative communications, which employs a non-orthogonal amplify-and-forward half-duplex relay through ISI channels. We first design and implement an optimal ML detector, which is implementable only for short relay period. Next three feasible sub-optimal

detectors are presented for the cooperative system with long relay period. A multitrellis Viterbi algorithm (MVA) based detector with near-optimal performance is proposed. However, the use of MVA detector is limited since it is not straightforward to extend to arbitrary ISI channels. The DFSE detector and the BP detector, which are traditionally used with static channels, are modified to accommodate to cooperative ISI channels. Also, we demonstrate the uncoded performance of these detectors in frequency selective fading channels, analyze the implementation complexity, and discuss complexity/performance tradeoffs.

6.2 Open Issues

The following is a list of open issues that can be further studied as an extension of this dissertation:

- As discussed in Chapter 2, the complexity of the BP detector is $\mathcal{O}(M^{D_g})$, where M is the order of modulations and D_g is the number of nonzero coefficients of the CIR. Many efforts have been made to reduce the implementation complexity, such as using PRE and layer processing scheme. However, the implementation result showed that the complexity is still very high even for a QPSK system for a channel with only three nonzero coefficients. Thus using BP detector with higher-order modulations is not practically implementable. Possible work can be done by modifying and simplifying the BP algorithm.

- The BP algorithm operates on the assumption that the noise is AWGN. However, in the PRBP detector, the PRE colors the noise. We can seek to quantify the performance penalty of the BP detector in the presence of colored noise.

- In Chapter 4, the performance advantage of the DFSF-BP detector over the traditional DFE is not significant for ATSC channels. It was shown that at a SER of 10^{-5}, the DFSF-BP detector exhibits performance about 1 dB better than the traditional DFE for both static and dynamic channels. However, the complexity of the DFSF-BP detector is much higher than the DFE, which makes the DFSF-BP detector less competitive. Our simulation results (which are not presented in this dissertation) indicates that the DFSF-BP detector still suffer the error propagation. Appropriate schemes like error correction code and interleave/deinterleave should be considered in the system.

- In our system model, perfect channel estimation has been assumed, which is not a usual case in reality. The performance robustness of BP detectors in the presence of channel estimation errors needs to be investigated.

Bibliography

[1] R. Chang, "Synthesis of band-limited orthogonal signals for multichannel data transmission," *Bell Sys. Tech. J.*, vol. 45, pp. 1775–1796, 1966.

[2] H. Sari, G. Karam, and I. Jeanclaude, "Transmission techniques for digital terrestrial TV broadcasting," *Communications Magazine, IEEE*, vol. 33, no. 2, pp. 100–109, 1995.

[3] W. Schreiber, "Advanced television systems for terrestrial broadcasting: Some problems and some proposed solutions," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 958–981, 1995.

[4] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," *The State of Technology in 2008*, vol. 42, no. 1, pp. 103–114, 2008.

[5] D. Falconer, S. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *Communications Magazine, IEEE*, vol. 40, no. 4, pp. 58–66, 2002.

[6] G. Forney Jr, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. 18, no. 3, pp. 363–378, May 1972.

[7] J. Proakis, *Digital Communications*, 4th ed. New York, USA: McGraw-Hill, 2000.

[8] S. Roy, T. M. Duman, and V. K. McDonald, "Error rate improvement in underwater MIMO communications using sparse partial response equalization," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 2, pp. 181–201, Apr. 2009.

[9] R. Machado, A. Klein, and R. Martin, "Decision feedback sparsening filter design for belief propagation detectors," in *Proc. IEEE Conf. on Information Sciences and Systems (CISS'12)*, Mar. 2012.

[10] S. Ariyavisitakul, N. Sollenberger, and L. Greenstein, "Tap-selectable decision-feedback equalization," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1497–1500, 1997.

[11] N. C. McGinty, R. A. Kennedy, and P. Hocher, "Parallel trellis Viterbi algorithm for sparse channels," *IEEE Communications Letters*, vol. 2, no. 5, pp. 143–145, May 1998.

[12] N. Benvenuto and R. Marchesani, "The Viterbi algorithm for sparse channels," *IEEE Transactions on Communications*, vol. 44, no. 3, pp. 287–289, Mar. 1996.

[13] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Transactions on Communications*, vol. 37, no. 5, pp. 428–436, 1989.

[14] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Transactions on Communications*, vol. 36, no. 1, pp. 13–20, 1988.

[15] M. N. Kaynak, T. M. Duman, and E. M. Kurtas, "Belief propagation over frequency selective fading channels," in *Proc. IEEE Veh. Technol. Conf.*, vol. 2, Sep. 26–29, 2004, pp. 1367–1371.

[16] G. Colavolpe and G. Germi, "On the application of factor graphs and the sum-product algorithm to ISI channels," *IEEE Transactions on Communications*, vol. 53, no. 5, pp. 818–825, May 2005.

[17] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[18] X. Hu, E. Eleftheriou, D. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE Global Telecommunications Conference*, vol. 2, 2001, pp. 1036–1036E.

[19] M. Mansour and N. Shanbhag, "High-throughput LDPC decoders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 6, pp. 976–996, 2003.

[20] T. Zhang and K. Parhi, "VLSI implementation-oriented (3, k)-regular low-density parity-check codes," in *Proc. IEEE Workshop on Signal Processing Systems*, 2001, pp. 25–36.

[21] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, "High throughput low-density parity-check decoder architectures," in *Proc. IEEE Global Telecommunications Conference*, vol. 5, 2001, pp. 3019–3024.

[22] Z. Wang and Z. Cui, "Low-complexity high-speed decoder design for quasi-cyclic LDPC codes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 1, pp. 104–114, 2007.

[23] P. Hoeher, "A statistical discrete-time model for the WSSUS multipath channel," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 4, pp. 461–468, 1992.

[24] A. Gomaa and N. Al-Dhahir, "A new design framework for sparse FIR MIMO equalizers," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2132–2140, Aug. 2011.

[25] R. Machado, A. Klein, and R. Martin, "Sparsening filter design for iterative soft-input soft-output detectors," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 72, 2012.

[26] D. Falconer and F. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell Syst. Tech. J*, vol. 52, no. 9, pp. 1541–1562, 1973.

[27] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain," in *Proc. IEEE International Conference on Communications*, vol. 2, 1995, pp. 1009–1013.

[28] K. Parhi, *VLSI digital signal processing systems: design and implementation*. John Wiley & Sons, Inc, 1999.

[29] D. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Workshop Signal Processing and Systems*, 2004, pp. 107–112.

[30] R. Cypher and C. Shung, "Generalized trace-back techniques for survivor memory management in the Viterbi algorithm," *The Journal of VLSI Signal Processing*, vol. 5, no. 1, pp. 85–94, Jan. 1993.

[31] P. J. Black and T. H. Meng, "A 140-Mb/s, 32-state, radix-4 Viterbi decoder," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 12, pp. 1877–1885, Dec. 1992.

[32] G. Feygin and P. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," *IEEE Transactions on Communications*, vol. 41, no. 3, pp. 425–429, Mar. 1993.

[33] Y. Gang, A. T. Erdogan, and T. Arslan, "An efficient pre-traceback architecture for the Viterbi decoder targeting wireless communication applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 9, pp. 1918–1927, Sep. 2006.

[34] Texas Instruments Incorporated, *TMS320C6748 Fixed/Floating-Point DSP Data Sheet*, 2011.

[35] ——, *TMS320C6748/46/42 Power Consumption Summary*, 2010.

[36] S. Qureshi and E. Newhall, "Adaptive receiver for data transmission over time-dispersive channels," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 448–457, 1973.

[37] D. G. Messerschmitt, "Design of a finite impulse response for the Viterbi algorithm and decision-feedback equalizer," in *International Conference on Communications*, 1974, p. 37.

[38] F. Magee, "A comparison of compromise Viterbi algorithm and standard equalization techniques over band-limited channels," *IEEE Transactions on Communications*, vol. 23, no. 3, pp. 361–367, 1975.

[39] C. Beare, "The choice of the desired impulse response in combined linear-Viterbi algorithm equalizers," *IEEE Transactions on Communications*, vol. 26, no. 8, pp. 1301–1307, 1978.

[40] N. Al-Dhahir and J. M. Cioffi, "Efficiently computed reduced-parameter input-aided MMSE equalizers for ML detection: a unified approach," *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 903–915, May 1996.

[41] *Guidelines for evaluation of radio transmission technologies for IMT-2000*, Recommendation ITU-R M.1225 Std., 1997.

[42] R. Martin and C. Johnson Jr, "Adaptive equalization: transitioning from single-carrier to multicarrier systems," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 108–122, 2005.

[43] *ATSC Digital Television Standard*, Advanced Television Systems Committee, Inc. Std., Rev. A/53, 2007.

[44] M. Ghosh, "Blind decision feedback equalization for terrestrial television receivers," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 2070–2081, 1998.

[45] H.-N. Kim, S. I. Park, and S. W. Kim, "Performance analysis of error propagation effects in the dfe for atsc dtv receivers," *IEEE Transactions on Broadcasting*, vol. 49, no. 3, pp. 249–257, 2003.

[46] Y. Wu, X. Wang, R. Citta, B. Ledoux, S. Lafleche, and B. Caron, "An atsc dtv receiver with improved robustness to multipath and distributed transmission environments," *IEEE Transactions on Broadcasting*, vol. 50, no. 1, pp. 32–41, 2004.

[47] Y. Peng, K. Zhang, A. G. Klein, and X. Huang, "Design and implementation of a belief propagation detector for sparse channels," in *Proc. IEEE Int Application-Specific Systems, Architectures and Processors (ASAP) Conf*, 2011, pp. 259–262.

[48] Y. Peng, A. G. Klein, and X. Huang, "Design and implementation of a low-complexity symbol detector for sparse channels," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, accepted.

[49] C. Wu, D. Bao, X. Zeng, and B. Shen, "An efficient iterative frequency domain equalization for ATSC DTV receiver," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2148–2154, 2010.

[50] I. E. Telatar, "Capacity of multi-antenna gaussian channels," AT&T Bell Laboratories, Tech. Rep., Oct. 1995. [Online]. Available: http://mars.bell-labs.com/papers/proof/

[51] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–59, Oct. 1996.

[52] W. Jakes, Jr., *Microwave Mobile Communications*. New York: John Wiley & Sons, Inc., 1974, ch. 1 and 5.

[53] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity, Parts I & II," *IEEE Transactions on Communications*, vol. 51, Nov. 2003.

[54] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[55] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, pp. 3037–3063, Sep. 2005.

[56] A. Sendonaris, E. Erkip, and B. Aazhang, "Increasing uplink capacity via user cooperation diversity," Aug. 1998, p. 156.

[57] J. N. Laneman, G. W. Wornell, and D. N. C. Tse, "An efficient protocol for realizing cooperative diversity in wireless networks," Jun. 2001, p. 294.

[58] P. Murphy, A. Sabharwal, and B. Aazhang, "Building a cooperative communications system," *IEEE Transactions on Information Theory*, submitted for publication.

[59] S. Wei, D. L. Goeckel, and M. C. Valenti, "Asynchronous cooperative diversity," *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1547–1557, 2006.

[60] K. Azarian, H. El Gamal, and P. Schniter, "On the achievable diversity-multiplexing tradeoff in half-duplex cooperative channels," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4152–4172, Dec. 2005.

[61] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[62] M. Kamuf, V. Owall, and J. B. Anderson, "Optimization and implementation of a Viterbi decoder under flexibility constraints," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 8, pp. 2411–2422, Sep. 2008.

[63] Y. Peng, A. G. Klein, and X. Huang, "Design of a maximum-likelihood detector for cooperative communications in intersymbol interference channels," in *Proceedings of the 19th ACM Great Lakes symposium on VLSI*. ACM New York, NY, USA, May 2009, pp. 429–432.

[64] R. Raheli, A. Polydoros, and C.-K. Tzou, "Per-survivor processing: a general approach to mlse in uncertain environments," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 354–364, Feb. 1995.