NON-ITERATIVE JOINT DECODING AND SIGNAL PROCESSING: UNIVERSAL CODING APPROACH FOR CHANNELS WITH MEMORY

A Dissertation

by

NITIN ASHOK NANGARE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2006

Major Subject: Electrical Engineering

NON-ITERATIVE JOINT DECODING AND SIGNAL PROCESSING: UNIVERSAL CODING APPROACH FOR CHANNELS WITH MEMORY

A Dissertation

by

NITIN ASHOK NANGARE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Krishna R. Narayanan
Committee Members,	Costas N. Georghiades
	Sunil Khatri
	Andreas Klappenecker
Head of Department,	Costas N. Georghiades

May 2006

Major Subject: Electrical Engineering

ABSTRACT

Non-iterative Joint Decoding and Signal Processing:
Universal Coding Approach for Channels with Memory. (May 2006)
Nitin Ashok Nangare, B.E., Pune University, India;
M.Sc., Technical University of Munich, Germany
Chair of Advisory Committee: Dr. Krishna Narayanan

A *non-iterative* receiver is proposed to achieve near capacity performance on inter-symbol interference (ISI) channels. There are two main ingredients in the proposed design. i) The use of a novel BCJR-DFE equalizer which produces optimal soft estimates of the inputs to the ISI channel given all the observations from the channel and L past symbols exactly, where L is the memory of the ISI channel. ii) The use of an encoder structure that ensures that L past symbols can be used in the DFE in an error free manner through the use of a capacity achieving code for a *memoryless* channel. Computational complexity of the proposed receiver structure is less than that of one iteration of the turbo receiver. We also provide the proof showing that the proposed receiver achieves the i.i.d. capacity of any constrained input ISI channel. This DFE-based receiver has several advantages over an iterative (turbo) receiver, such as low complexity, the fact that codes that are optimized for memoryless channels can be used with channels with memory, and finally that the channel does not need to be known at the transmitter. The proposed coding scheme is universal in the sense that a single code of rate r, optimized for a memoryless channel, provides small error probability uniformly across all AWGN-ISI channels of i.i.d. capacity less than r.

This general principle of a proposed non-iterative receiver also applies to other signal processing functions, such as timing recovery, pattern-dependent noise whiten-

ing, joint demodulation and decoding etc. This makes the proposed encoder and receiver structure a viable alternative to iterative signal processing. The results show significant complexity reduction and performance gain for the case of timing recovery and patter-dependent noise whitening for magnetic recording channels.

To my family

ACKNOWLEDGMENTS

This dissertation is the result of my work at Texas A&M University where I have been accompanied and supported by many people. It gives me great pleasure to acknowledge all of them. First of all, I would like to express my deepest gratitude to my advisor, Professor Krishna Narayanan. It was a rewarding and stimulating experience to work under the guidance of Professor Krishna Narayanan. Without his continuous support, guidance and patience this dissertation would never have been completed. He has always been very encouraging and inspiring throughout my studies.

I am also very thankful to Professor Costas Georghiades for providing us excellent research environment at Wireless Communications Laboratory at Texas A&M University.

I would like to thank Dr. Erozan Kurtas for his support and guidance during my internship at Seagate research lab. I am also very thankful to Dr. Xueshi Yang, my mentor at Seagate research lab, who helped me immensely to understand the magnetic recording system. His sharp intellect and willingness to help, always encouraged me to learn more about the magnetic recording system. I sincerely appreciate his guidance throughout my studies.

I would also like to thank Dr. Fredy Neeser for his excellent guidance during my master thesis. My master thesis work with him laid much of the foundation of my work at Texas A&M.

I would like to express my gratitude to my other two Ph.D. committee members, Professor Sunil Khatri and Professor Andreas Klappenecker, for their patience and help in several stages of my doctoral studies. I am also grateful to Professor Scott Miller, for his excellent teaching of many graduate courses at Texas A&M. I would

like to acknowledge the invaluable help of Tammy Carda, Sonny Matous and Linda Currin in dealing with all the administrative procedures.

I am also very grateful to all of my current and former colleagues and my friends who made studying and living in College Station much more pleasant. I would particularly like to thank Zubin Sukheswalla, Vishal Padiyar, Gaurav Shirvastav, Uday Sant, Sagar Sabade, Hari Sankar, Vivek Gulati, Janath Peiris, Angelos Liveris, Abhiram Prabhakar, Jing Jiang, Dung Doan, Kapil Bhattad and Arzu Karaer.

This acknowledgement would be incomplete without mentioning the unconditional love and support of my parents, my brother, my young sister and my wife Trupti. I owe a lot to my parents and my brother for their consistent efforts and sacrifice to provide me the best possible education.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION AND ORGANIZATION OF THE DISSERTATION

A. Introduction

After the introduction of turbo codes [1], iterative signal processing has become a popular paradigm in communications. Several papers have demonstrated the benefit of iterating between a soft output decoder and a front end signal processing block such as an equalizer, demodulator, channel estimator or timing recovery block [2–5]. In this dissertation, we show that turbo or iterative signal processing is not necessary (and demonstrate an alternate approach) in order to achieve near capacity performance. We show a simple receiver structure whose complexity is less than that of one iteration of the turbo receiver and yet achieves i.i.d. capacity [1]. The proposed receiver performs joint decoding and signal processing through a decision feedback mechanism.

Often in wireless communication the channel realization is not known at the transmitter and also it is not constant. In such scenario it is desired to have a coding scheme whose performance is independent of the channel realization. Existence of such codes known as *universal* codes for the case of linear, time invariant Gaussian channels was shown by Root and Varaiya [6] in 1968. Root and Varaiya showed the existence of universal codes but did not provide any construction for them. In this dissertation we show that the proposed encoder and receiver forms a universal coding scheme for the case of constrained input channels with Markov memory.

The general principle of decision feedback mechanism can also be applied to other

---

The journal model is *IEEE Transactions on Automatic Control.*

[1] We mean achieves the information rate corresponding to the input constellation and the independent and identically distributed input.

signal processing functions such as such as timing recovery, pattern dependent noise whitening, joint demodulation and decoding etc., making the proposed encoder and receiver structure a viable alternative to iterative signal processing.

B. Organization of the Dissertation

Organization of the dissertation into different chapters is explained in this section.

Magnetic recording

A model for the magnetic recording channel is explained in chapter II. The magnetic recording channel is a unique channel which consist of inter-symbol interference (ISI) from neighboring magnetized bit-cells, pattern dependent media noise (which is correlated with itself and also with the data pattern) due to unevenness of magnetic media and timing errors due to misalignment of magnetized media and read head. Thus it offers a perfect example of a channel with memory for evaluating the performance of the proposed non-iterative receiver which performs joint decoding and signal processing.

Communication channels and channel coding

Memoryless communication channels and channels with memory are explained in chapter III. Performance of class of codes known as low density parity-check (LDPC) codes over memoryless channels is discussed. Previously published research which shows that LDPC codes are universal codes for the case of many memoryless channels is explained in brief. Three examples of channels with memory which are of interest to this dissertation, namely i) ISI channel, ii) channel with unknown timing offsets and iii) channel with additive correlated Gaussian noise are explained. A

general problem of code design known as Root-Varaiya problem, which shows the existence of universal codes is outlined. In this chapter, we also briefly discuss the result from [7] which shows that for iterative receivers it is impossible to design a universal code.

BCJR-DFE receiver

Chapter IV introduces a novel non-iterative receiver based on a decision feedback scheme with the BCJR (Bahl, Cocke, Jelinek and Raviv) algorithm [8] (hence the name BCJR-DFE receiver). Similarities and main differences of the proposed receiver structure to other known schemes are discussed. Main ingredients of the proposed design, i) an encoding structure and ii) the BCJR-DFE algorithm are developed in this chapter. Main properties of BCJR-DFE receiver are discussed in this chapter, which shows that i) it achieves the i.i.d. capacity of any constrained input ISI channel and ii) it is a universal coding scheme for the class of constrained input Gaussian ISI channels. Finite length performance of BCJR-DFE receiver is evaluated for three ISI channels which shows that this non-iterative receiver outperforms the iterative (turbo) receiver and at the same time it has complexity less than that of one iteration of the iterative receiver. Part of the material in chapter IV has been published in [9]

Low latency techniques for BCJR-DFE receiver

A disadvantage of BCJR-DFE receiver is that it has long latency. Two methods to reduce the latency of BCJR-DFE receiver are proposed and evaluated in chapter V. Simulation results show that latency of the BCJR-DFE receiver can be reduced significantly without much loss in the performance. We also discuss loss in information rate (or loss in signal-to-noise ratio) when the optimum nonlinear BCJR estimator

in BCJR-DFE receiver is replaced with a linear MMSE estimator. Results show that there is little loss in the information rate with linear MMSE filtering for a wide range of SNRs with the proposed MMSE-DFE scheme. Part of the material in chapter V is submitted in [10].

Joint synchronization, ISI equalization and decoding

In chapter VI, we consider a channel with unknown timing errors along with ISI in a communication channel. Performance of BCJR-DFE receiver in such environment is evaluated using per-survivor based processing for timing recovery. Longitudinal and perpendicular magnetic recording channels are used along with high rate codes (suitable for magnetic recording systems) for performance evaluation. Part of the material in chapter VI has been published in [11].

BCJR-DFE receiver for media noise dominated magnetic recording channels

Due to strong pattern dependent correlated media noise in magnetic recording channels, pattern dependent noise prediction is often used for magnetic systems for additional performance improvement. We combine a pattern dependent noise predictive detector along with BCJR-DFE receiver to combat media noise and ISI jointly. Performance of this pattern dependent noise predictive BCJR-DFE receiver is tested on longitudinal and perpendicular magnetic recording channels dominated by media noise. In all chapters, performance of proposed non-iterative scheme is compared with corresponding iterative receiver setup, which shows significant gains for the proposed BCJR-DFE receiver. The material in chapter VI has been published in [12] and [13].

Extension of dissertation work

The BCJR-DFE principle is a general principle which can be applied to many other signal processing problems. In chapter VIII we discuss other scenarios where it can applied. The use of BCJR-DFE receiver to obtain lower bounds on the capacity of channels involving various signal processing functions is discussed.

CHAPTER II

MAGNETIC RECORDING

Magnetic recording is the process of storing digital data in the form of a magnetization of a physical medium. In our day to day life we carry lot of data stored on magnetic medium. Our credit cards, transportation tickets, security badges, computer hard-disks, camera video tapes etc. contain magnetically recorded digital data. Data storage capacity of hard disk has increased enormously since the invention of a first hard disk by IBM in 1957. As the capacity of hard disk increases, so does the complexity of signal processing required to read and convert the magnetic signal back to digital signal.

In this chapter we explain the process of magnetic recording. Channel coding and signal processing for magnetic recording is a challenging problem due to variety of noise sources in the magnetic channel. The magnetic recording channel is unique channel with inter-symbol interference, timing offset and additive correlated noise (which is correlated not only with itself but also with data pattern). At the same time a low complexity solution is often required for magnetic recording due to size limitations of the hard-disk. Thus our proposed non-iterative decoding and signal processing scheme offers a perfect match for the magnetic recording channel. In the following sections we will explain the process of magnetic recording and the magnetic read-back signal. We will also characterize many forms of distortions present in the magnetic read-back signal.

A. Write and Read-back Signal

Data is stored on the magnetic disk drives in binary form, “1” (one) is stored as positive magnetization or magnetization in the left direction and “0” (zero) is stored as

negative magnetization or magnetization in the right direction. Three essential elements of magnetic recording are write head, magnetic medium and read head. Electric current can magnetize ferromagnetic material, and the trajectory of medium magnetization can be shown by a nonlinear hysteresis loop. Magnetic recording channel is linearized artificially by constraining the input write current to two extreme levels such that the magnetic medium is always saturated, known as *saturation-recording*. Thus magnetic recording channel is always a binary input channel as opposed to other

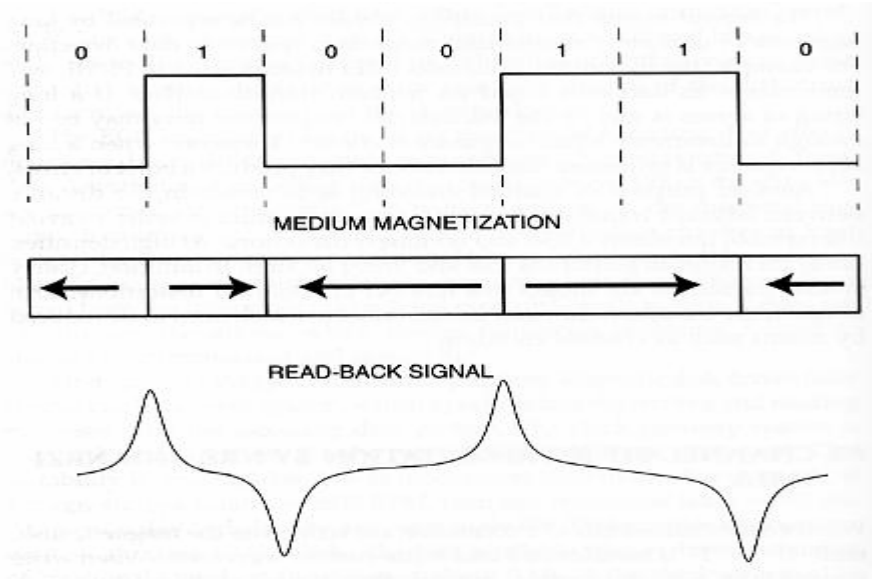

Fig. 1. Magnetic recording process

communication channels. The magnetic recording process is shown in Fig. 1. Ideal write process of magnetic recording consist of magnetization of medium either to the left or to the right direction by two extreme current levels. In ideal read process, read head reads off the data from a magnetic disk by detecting the changes in magnetic flux. Thus read head acts as a differentiator for the recorded magnetic data. The read-back signal can be characterized by the transition response $h(t, w)$, which corre-

sponds to a positive transition from bit 0 to bit 1, (-$h(t,w)$ for a negative transition from bit 1 to bit 0). For longitudinal recording, the read-back transition response is well approximated as Lorentzian pulse determined by a parameter, $w = PW_{50}$, the pulse width at 50% amplitude of peak value V_p. It is given by

$$h(t,w) = \frac{V_p}{1 + (\frac{2t}{PW_{50}})^2}$$

The ratio PW_{50}/T, where T is the data rate (or bit spacing parameter) is termed as *normalized linear density* of a magnetic disk. It is the single most important parameter to characterize the channel in a magnetic recording system. For perpendicular recording, the transition response is approximated by,

$$h(t,w) = V_p \cdot \text{erf}\left(\frac{2\sqrt{\ln 2}}{w}t\right)$$

where erf(x) is the error function defined as

$$\text{erf}(t) = \frac{2}{\sqrt{\pi}}\int_0^t e^{-x^2}dx$$

Typical read-back signal for recorded bits $a_k \in \{+1, -1\}$ can be written as,

$$r(t) = \sum_k b_k h(t - kT, w) + n(t)$$

where $b_k = a_k - a_{k-1}$ denotes the differentiation operation of the read head, $n(t)$ denotes the electronics noise assumed to be additive white Gaussian (AWGN). Due to differentiation, effective impulse response (known as dibit response) of a magnetic channel is given by $p(t,w) = h(t,w) - h(t-T,w)$. Thus read-back signal can also be written as

$$r(t) = \sum_k a_k p(t - kT, w) + n(t).$$

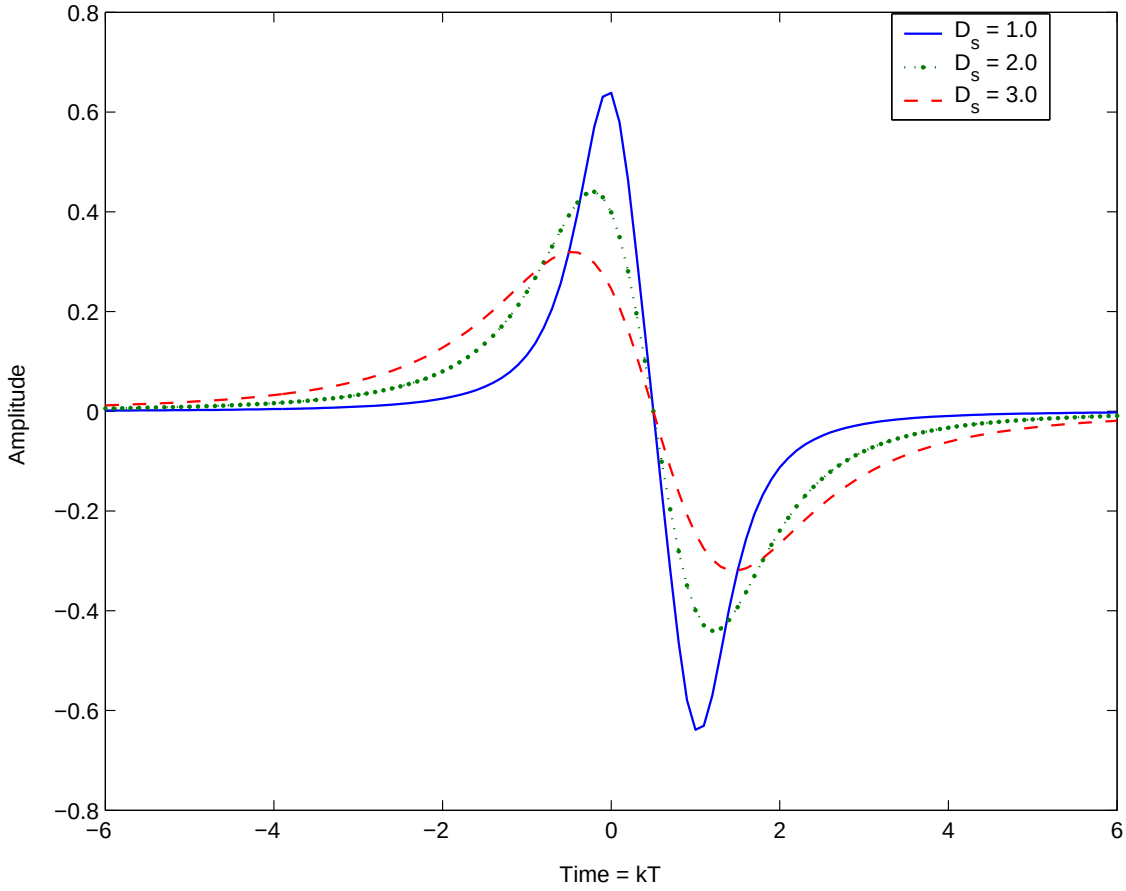

Fig. 2. Change in dispersion of dibit response of a longitudinal magnetic recording with different normalized densities D_s

B. ISI in Magnetic Recording System

Normalized density $D_s = PW_{50}/T$ indicates the number of bits recorded in the interval $w = PW_{50}$. Overlapping of adjacent read-back pulses causes ISI in a magnetic recording system. A larger value of normalized density (D_s) causes more dispersion of the read-back pulse and therefore more ISI in the system. Figures 2 and 3 shows increase in dispersion of dibit response for longitudinal and perpendicular magnetic recording as normalized density increases. We also observe that magnitude of the

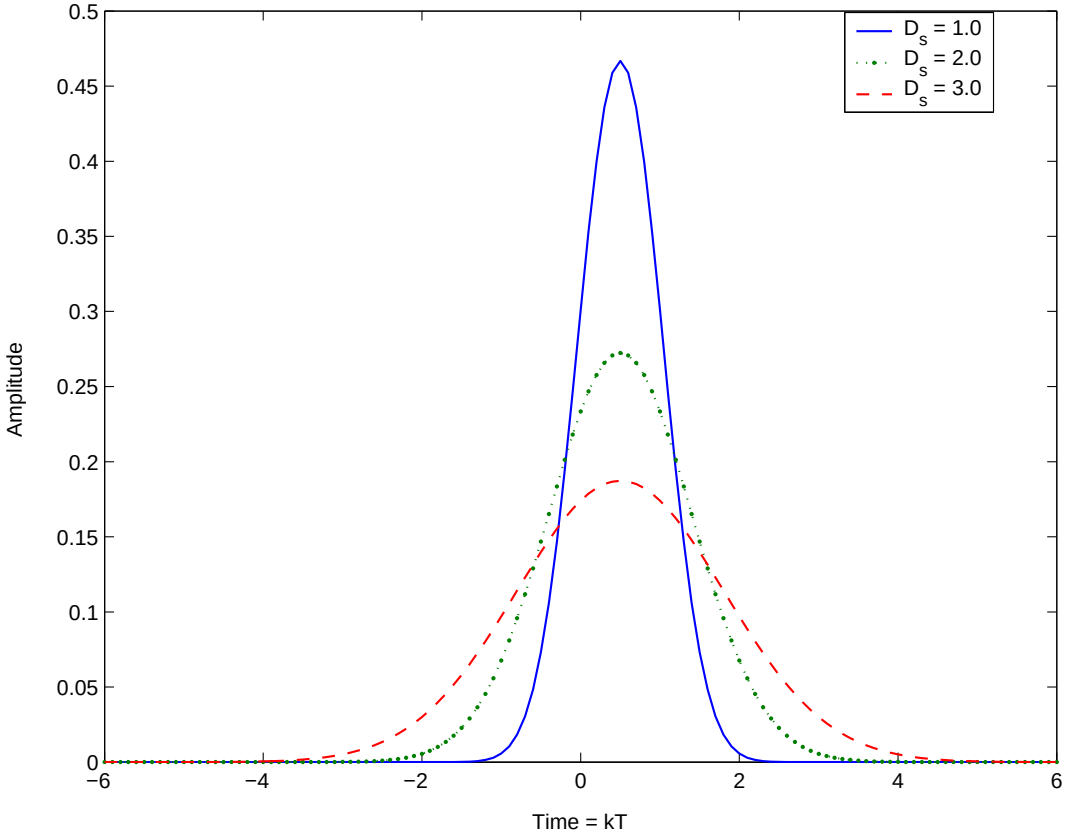

Fig. 3. Change in dispersion of dibit response of a perpendicular magnetic recording with different normalized densities D_s

read-back signal decreases as normalized density increases due to more ISI.

C. Noise in Magnetic Recording System

Noise in a magnetic recording system comes from three sources. First, there is noise due to unevenness of the magnetic medium and random magnetization patterns in the magnetic media. This type of noise is referred as media noise. Second, there is noise produced by the read head and finally, there is the system electronics noise. As normalized density increases, the size of a magnetized grain decreases and media noise

becomes the dominant source of noise in a magnetic recording system, accounting for almost 90% of total noise in future generation high density hard disks. Ideal magnetic transitions are shown in Fig. 1, which are perpendicular to the recording direction. In practice these lines are zig-zag patterns [14] as shown in Fig. 4. Due to zig-zag nature

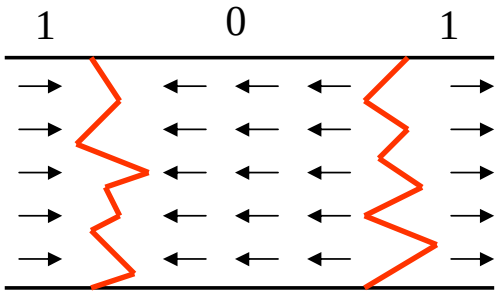

Fig. 4. Realistic zig-zag nature of magnetic transitions: caused due to unevenness of magnetic media

of magnetic transitions, the *transition center*, defined as the average transition center across data track, becomes uncertain. This uncertainty is called as *position jitter* (Δt_k for kth transition). It also causes shortening/broadening of the read-back pulse called as pulse-width variation (Δw_k for kth transition). These random variations form a random transition noise in the read-back pulse. Taking into account pulse position jitter (Δt_k) and pulse width variation (Δw_k), we can write transition response $h_k(t)$ at kth bit-interval as, $h_k(t) = h(t - kT - \Delta t_k, w + \Delta w_k)$. A block diagram of a typical magnetic recording channel is shown in Fig. 5, which consist of differentiation operation of read head and transition jitter and pulse width variation as explained previously. At the receiver end, electronic noise gets added into the read-back signal. Received signal can be expressed as,

$$r(t) = \sum_k b_k h(t - kT - \Delta t_k, w + \Delta w_k) + n(t).$$

This read-back signal $r(t)$ is then filtered with the low pass filter which restricts the signal energy to be further down-sampled at the baud-rate. Unknown timing offset τ_k is introduced in a sampling process due to imperfect knowledge of exact arrival of read-back signal. This further changes the transition response to $h_k(t) = h(t - kT - \Delta t_k - \tau_k, w + \Delta w_k)$.

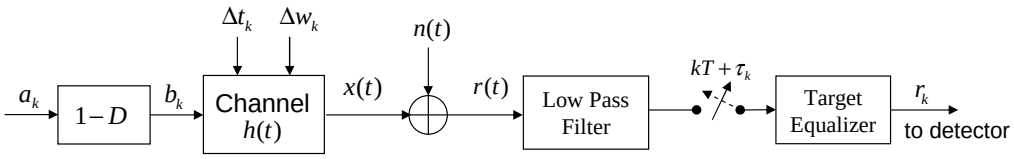

Fig. 5. A typical model of magnetic recording channel

D. Signal to Noise Ratio

After low-pass filtering and sampling, read-back signal is equalized to the target response with an equalizer filter. These filtering operations convert the system electronics noise (AWGN) into a correlated noise. Assuming ideal low-pass filtering and ideal target equalization, we can define the signal-to-noise ratio (SNR) for recording channel with transition noise as

$$SNR = \frac{E_i}{N_0 + M_0}$$

where $E_i = \int_{-\infty}^{\infty} \left[\frac{\partial h(t,w)}{\partial t}\right]^2 dt$, is the energy of the impulse response of the recording channel, N_0 is the power spectral density of AWGN and M_0 is the average transition noise energy associated with an isolated transition. Assuming that the transition jitter Δt_k and pulse width variation Δw_k are independent between transitions, the

average transition noise energy M_0 is defined as

$$M_0 \triangleq \int_{-\infty}^{\infty} E\{[h(t,w) - h(t+\Delta t_k, w+\Delta w_k)]^2\}dt$$

Also, if we assume that the distribution of Δt_k and Δw_k do not change with normalized density then M_0 is completely determined by the transition response and is independent of the normalized density. So M_0 can be considered as analogous to N_0 for AWGN, and it represents the equivalent power spectral density for transition noise. Often Δt_k and Δw_k are assumed to be zero-mean Gaussian distributed. Different methods to determine the value of pulse position jitter variance σ_j and pulse width variation variance σ_w for given percentage of jitter noise (media noise) are shown in [15, 16]. Unlike other communication channels where there is a power constraint, magnetic recording channels are constrained by the transition response $h(t,w)$, which is fixed once the head/media and the associated electronic circuits are defined. Thus once SNR is given, N_0 and M_0 are fixed, which fixes the head/media and electronic circuits. From the above definition of an SNR we see that higher SNR corresponds to lower electronic noise power N_0 and lower transition noise power M_0 which are obtained through better electronic circuits and better magnetic media. Any channel coding gain (in terms of SNR) achieved is used for combating increased ISI and media noise for higher density of magnetic recording. For a more detailed explanation of SNR in magnetic recording system we refer reader to [15, 16].

E. Equalization and Detection

As shown in Fig. 5, sampled read-back signal is equalized with a target equalizer which shapes the read-back signal to a known target response. The magnetic recording channel is inherently a bad channel. Longitudinal channel has a spectral null at DC

and strong attenuation at high frequencies due to Lorentzian pulse response, whereas perpendicular channel has strong attenuation only at high frequencies. Complete removal of ISI is not possible without strong enhancement of noise at DC and/or at the Nyquist frequency. So controlled amount of ISI is introduced in the magnetic recording system by shaping the channel to a target response which has close spectral characteristics of the read-back pulse. For the longitudinal channel, a natural choice is placing zeros at DC and at Nyquist frequency, giving the partial response (PR) target of type $(1-D)(1+D)^n$ whereas for perpendicular channel the partial response (PR) target is given by $(1+D)^n$. For both cases, value of n controls the high frequency attenuation. Above mentioned targets give rise to integer coefficients for the target response, making it suitable for hardware implementation. Non-integer targets known as generalized partial response (GPR) targets [17], which match the high density magnetic read-back signal more closely can be used. For magnetic recording, often equalization refers to equalizing the read-back signal to a suitable partial response target. After equalization, a partial response maximum likelihood (PRML) detector (which operates on known target response) is needed for data detection. The Viterbi detector [18] matched to an equalized PR target can be used for data detection. It is the optimum detector for detection of a signal corrupted with additive white noise. Without any noise whitening, it provides sub-optimum detection for magnetic read-back signal since it is corrupted with correlated media noise. If we assume that the noise correlation is independent of the data pattern then noise whitening filter followed by the Viterbi detector provides optimum detection for magnetic read-back signal. An advanced detector known as noise predictive maximal likelihood (NPML) detector which whitens the noise before detection was proposed in [19]. In practice media noise depends on magnetic transitions which in turn depend on data pattern being written. So noise correlation is data dependent and NPML detector does not

provide optimum detection. More advanced detector known as pattern dependent noise predictive (PDNP) detector which provides optimum detection in the presence of pattern dependent correlated noise has been developed in [20,21]. For an overview of different detection schemes for magnetic recording system we refer reader to [22,23].

CHAPTER III

COMMUNICATION CHANNELS AND CHANNEL CODING

Information is communicated from one point to the other through a physical medium known as a communication channel. The maximum rate at which this information can be communicated with arbitrarily small probability of error through a given channel is known as channel capacity of that channel [24]. Error correction codes [1] (ECC) are used in order to achieve errorfree communication. In following sections, we broadly classify communication channels as channels without memory and channels with memory. Channel coding schemes approaching channel capacities are discussed along with an interesting problem of universal codes for memoryless channels and channels with memory.

A. Memoryless Channels

Consider a discrete time communication channel shown in Fig. 6, where X_k denote transmitted symbol over the channel at time k and Y_k denote received symbol at time k. The channel is said to be a memoryless channel if the conditional probability

Fig. 6. Communication channel

[1] often referred as channel codes

$P(\mathbf{Y}|\mathbf{X})$ can be written as

$$P(\mathbf{Y}|\mathbf{X}) = \prod_k P(Y_k|X_k) \tag{3.1}$$

So for a memoryless channel, the received symbol Y_k at time k depends only on transmitted symbol X_k at time k. Examples of memoryless channels include additive white Gaussian noise (AWGN) channel, binary symmetric channel (BSC) and binary erasure channel (BEC).

1. Channel Coding for Memoryless Channels

The goal of any channel coding scheme is to achieve errorfree performance very close to the channel capacity. Berrou *et al.* [1] proposed turbo codes in 1993 which perform close to the capacity on additive white Gaussian noise channel. After turbo codes, many "turbo-like" codes which have a low complexity iterative message passing decoding algorithm were introduced. Among these turbo-like codes are low density parity-check codes (LDPC or Gallager [25, 26] codes), irregular LDPC codes [27, 28], repeat-accumulate codes [29, 30], irregular turbo-like codes [31] etc. All of these tubo-like codes were shown to perform close to capacity on symmetric memoryless channels. There effectiveness even for nonstandard channels is conjectured in [32]. Here we consider binary regular and irregular LDPC codes for memoryless channels and channels with memory.

a. LDPC Codes

LDPC codes were discovered by Gallager [25, 26] and then rediscovered by Spielman *et al.* [33] and MacKay *et al.* [34]. A binary LDPC code of length N is a (N, K) binary linear block code and is defined by a sparse parity-check matrix H of size $N \times (N-K)$. LPDC code can be represented by a bipartite graph (Tanner graph) [35] in which a set

of nodes called *variable nodes* correspond to codeword bits (columns of H) and other set of nodes called *check nodes* correspond to set of parity checks defined by rows of parity-check matrix H. An (j, k)-regular LDPC code has j ones in all columns of H (variable nodes of degree-j) and k ones in all rows of H (check nodes of degree-k). Irregular LDPC code is represented with degree distribution pair $(\lambda(x), \rho(x))$ where,

$$\lambda(x) = \sum_{i=1}^{j_{max}} \lambda_i x^{i-1},$$

$$\rho(x) = \sum_{i=1}^{k_{max}} \rho_i x^{i-1}.$$

In above equations, λ_i and ρ_i denote the fractions of edges corresponding to degree-i variable node and check node respectively [36]. Here j_{max} and k_{max} denote the maximum variable degree (maximum number of ones in any given column of H) and maximum check degree (maximum number of ones in any given row of H) respectively.

LDPC code design for memoryless channel is well understood and there are tools (e.g. density evolution) to analyze and design good ensembles of codes for memoryless channels [37, 38]. Example of rate 1/2 irregular LDPC code for which the iterative decoding threshold on AWGN channel is within 0.0045 dB from the capacity was given in [39]. A procedure for designing LDPC code ensembles which achieves capacity on BEC channel as code length goes to infinity was given by Oswald and Shokrollahi in [40]. Universality of LDPC codes in terms of its performance for memoryless channels was shown in [37, 41], where authors suggested that no matter what is the initial density function of received symbol log-likelihood-ratios (LLRs) is (i.e. no matter which type of memoryless channel is), LDPC codes perform equally good on all memoryless channels. So for memoryless channels, channel capacity is the most important parameter seen by an LDPC code. Table I (taken from [37]) shows channel capacities at the threshold values of various (j, k) regular LDPC codes.

Table I. Channel capacities of different memoryless channels at the threshold values of various (j, k) regular LDPC codes

j	k	Rate	$C_{(BEC)}$	$C_{(BSC)}$	$C_{(Laplace)}$	$C_{(AWGN)}$
3	6	0.5	0.571	0.584	0.577	0.571
4	8	0.5	0.617	0.609	0.606	0.605
3	4	0.25	0.353	0.349	0.348	0.346
4	10	0.6	0.692	0.689	0.684	0.683
3	12	0.75	0.790	0.814	0.801	0.793

That is, in order to provide a bit-error-rate approaching zero as code length goes to infinity with chosen (j, k) regular LDPC code, different memoryless channels need to have capacities as shown in Table I. The author of [37] quotes that the fact that these capacity values for different channels are very close to each other indicates that LDPC codes optimized for one memoryless channel provides good performance for other memoryless channels, as observed in [41]. The same thing was shown to be true for various irregular LDPC codes in [37]. Thus LDPC codes can be considered as universal codes for the class of memoryless channels. Universality of LDPC codes for partial-band jamming and fading channels was shown in [42].

B. Channels with Memory

In practice many times we encounter channels with memory. For a channel with memory, we can write

$$P(\mathbf{Y}|\mathbf{X}) \neq \prod_{k} P(Y_k|X_k) \tag{3.2}$$

In this thesis, we will consider a Markov model with memory $L = L_1 + L_2$ such that

$$P(\mathbf{Y}|\mathbf{X}) = \prod_k P(Y_k|X_{k-L_1} \cdots X_k \cdots X_{k+L_2}) \tag{3.3}$$

Thus for channels with memory, received symbol Y_k at time k depends not only on transmitted symbol X_k at time k, but also depends on few or all transmitted symbols. Examples of channels with memory include ISI channel, channel with timing offsets and channel with additive correlated Gaussian noise.

1. Inter-Symbol Interference Channel

Due to multi-path nature of a communication channel, signal pulses transmitted in adjacent time interval overlap with each other causing inter-symbol interference. For an ISI channel, received signal Y_k can be written as weighted combination of more than one transmitted symbol. Specifically, for an ISI channel of memory L, the received signal Y_k at the output is given by

$$Y_k = \sum_{i=0}^{L} h_i \; X_{k-i} + N_k, \tag{3.4}$$

where h_i's are the channel tap coefficients (assumed to be complex) and N_k is complex, additive white Gaussian noise. The memory of an ISI channel is decided by the total number of resolvable paths between the transmitter and the receiver. Maximum likelihood (ML) sequence estimation can be performed on the received sequence $\{Y_k\}$ using a nonlinear Viterbi algorithm [18, 43] or using the BCJR algorithm [8] if soft output is desired.

2. Channel with Unknown Timing Offsets

Another example of channel with memory is a channel with unknown timing offsets. Consider a transmitter, transmitting an analog signal pulse at every kth time instant

kT. The receiver sampler is uncertain about the correct arrival time of an analog pulse due to timing offsets between the transmitter and the receiver clock. The received pulse is sampled at time $kT + \tau_k$, where τ_k denote the timing offset at time kT. These unknown timing offsets introduce memory in the channel. For example, consider a transmitter, transmitting user information X_k using *sinc* pulse shape $g(t) = \frac{\sin(\pi t)}{\pi t}$. In the absence of any timing offsets, receiver samples the received signal at every integer multiple of time T. In this case, the received signal $Y(kT)$ with receiver noise N_k can be written as,

$$Y(kT) = \sum_{k=-\infty}^{+\infty} X_k \cdot g(kT) + N_k \tag{3.5}$$

$$= X_k + N_k \tag{3.6}$$

which results in zero ISI. Whereas with any non-zero sampling timing offset τ_k, we can write

$$Y(kT) = \sum_{k=-\infty}^{+\infty} X_k \cdot g(kT + \tau_k) + N_k \tag{3.7}$$

which results in infinite ISI in the system.

3. Channel with Additive Correlated Gaussian Noise

Consider a communication channel corrupted with additive Gaussian noise which is correlated with itself. The received signal Y_k corresponding to the transmitted symbol X_k can be written as

$$Y_k = X_k + N_k$$

where N_k denote correlated Gaussian noise samples. A correlated Gaussian noise process with given correlation can be represented as a filtered version of white Gaussian noise process. Hence, noise whitening can be performed on the received signal Y_k

using a noise whitening filter W such that

$$\begin{aligned} \widetilde{Y}_k &= W * Y_k \\ &= W * X_k + \widetilde{N}_k \end{aligned} \tag{3.8}$$

where $*$ denotes convolution operation and $\widetilde{N}_k$ denotes additive white Gaussian noise (AWGN). Thus from (3.8) we can see that, a channel with additive correlated Gaussian noise is channel with memory. The memory is introduced in the channel by the memory in a correlated noise process. The magnetic recording channel is an example of a channel with additive correlated Gaussian noise. In magnetic recording, noise is not only correlated with itself but it is also correlated with the recorded data bits [44], which makes the noise correlation process signal dependent. So for magnetic recording every signal experiences a different channel with memory.

C. Capacity and Achievable Information Rates for an ISI Channel

Let us first consider the case when $\mathbf{X} = \{X_1, X_2, \cdots, X_k\}$ is a sequence of BPSK (binary phase shift keying) modulated symbols and $\mathbf{Y} = \{Y_1, Y_2, \cdots, Y_k\}$ is a corresponding output sequence of an ISI channel. The channel capacity is defined as

$$C = \lim_{k\to\infty} \frac{1}{k} \sup_{P(\mathbf{X})} I(\mathbf{X}; \mathbf{Y}). \tag{3.9}$$

When the distribution on X_k's is uniform and i.i.d., then the achievable information rate is called $C_{i.i.d.}$ and is given by

$$C_{i.i.d} = \lim_{k\to\infty} \frac{1}{k} I(\mathbf{X}; \mathbf{Y}) \quad \text{where } P(\mathbf{X}) = \prod P(X_k) \tag{3.10}$$

We can see that, $C_{i.i.d.} \leq C$. A method to calculate this achievable information rate was devised independently by Arnold and Loeliger [45], Pfister, Soriaga and

Siegel [46], Sharma and Singh [47]. The extension to the case of non i.i.d. inputs is possible if the inputs have a Markov structure. Note that we only consider the case of finite alphabet input as we require a Markov structure with finite number of states. When the constellation size is large and appropriate shaping is used, the true capacity of the channel can be approached fairly closely.

D. Root-Varaiya Problem of Universal Codes

More general problem of code design was proposed by Root and Varaiya [6] in 1968.
Definition 1: Root-Varaiya problem : Root-Varaiya problem states that, a single code of rate r should be able to provide small error probability *uniformly* across any linear, time invariant Gaussian channel (including ISI) as long as the capacity of the channel is greater than r.

Let us assume that we have a class of Gaussian ISI channels $\{H_1, H_2, H_3, \cdots\}$ with corresponding output sequence,

$$\mathbf{Y}_i = H_i * \mathbf{X}_i + \mathbf{N}_i$$

where $*$ denotes convolution operation. Let $C(H_i) > r$, where $C(H_i)$ denote the capacity of channel H_i. Then Root and Varaiya showed that there exists a universal code which can transmit reliably with rate r on all of these channels. For the case of constrained input constellation, we assume that similar universal code exists. Furthermore we consider only i.i.d. capacities of the channels such that $C_{i.i.d.}(H_i) > r$. Certainly these universal codes are of practical importance in wireless communication as often the channel realization is not known at the transmitter. Root and Varaiya showed the existence of universal codes but did not provide any construction for them. In this thesis, we provide an encoding an decoding scheme based on LDPC

codes which is universal for the class of constrained input Gaussian ISI channels. Here we assume the existence of universally good LDPC codes for the memoryless channel. Thus we show that LDPC codes are universal codes for the class of constrained input Gaussian ISI channels under special encoding and decoding scheme.

E. Channel Coding for Channels with Memory

Channel codes need to be designed keeping in mind the memory of a channel. Often when error correction codes are used for channels with memory, we can represent the whole system as a serial concatenation of error correction code used and a channel with memory. Designing capacity achieving coding schemes and corresponding decoding algorithm for channels with memory was a hard problem until the invention of turbo codes in 1993 by Berrou, *et al.* [1]. Soon turbo decoding principle was extended to other communications problems in which two or more data processing units interact with each other for data transmission [48]. Figure 7 shows a typical turbo processing scheme for a serially concatenated system. As shown in Fig. 7, user bits are encoded

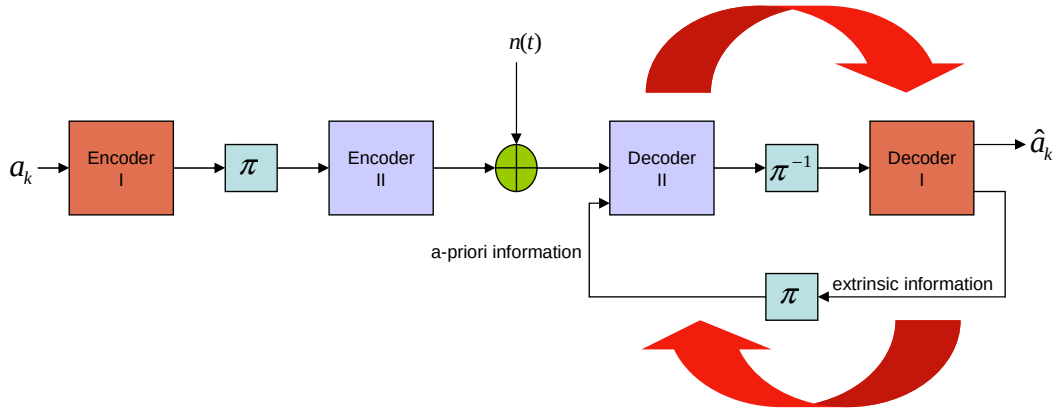

Fig. 7. A typical serially concatenated turbo (iterative) signal processing scheme

with 'Encoder-I' (outer encoder), then interleaved and encoded with 'Encoder-II' (inner encoder). On the receiver side, 'Decoder-II' (inner decoder) provides extrinsic information which is de-interleaved and then given to 'Decoder-I' (outer decoder). Interleaved extrinsic information from outer decoder is given to the inner decoder as a-priori information. Many communications and signal processing schemes for channels with memory can be configured in such serially concatenated scheme as shown in Table II.

Table II. Turbo (iterative) configuration for different serially concatenated systems

Configuration	**En-/Decoder I**	**En-/Decoder II**
Turbo decoding	ECC en-/decoder	ECC en-/decoder
Turbo equalization	ECC en-/decoder	ISI channel/detector (equalizer)
Turbo synchronization	ECC en-/decoder	asynchronous channel/detector with timing recovery
Turbo noise prediction	ECC en-/decoder	correlated channel/detector with noise predicttion

1. Turbo Equalization

Consider a coded data transmission over an ISI channel. For this case, 'Encoder-I' represents an outer code while 'Encoder-II' represents an ISI channel. On the receiver side, 'Decoder-II' represents a channel equalizer and 'Decoder-I' represents an outer decoder. Iterative signal processing can be performed between a channel equalizer and an outer decoder which is often called as turbo equalization [2]. Soft-input-soft-output (SISO) equalizer is required for turbo equalization. A trellis based SISO equalization

using the BCJR algorithm is often used which produces optimum soft-outputs for the bits transmitted over an ISI channel. The BCJR algorithm for ISI channel equalization is explained in detail in Appendix-A. Sub-optimal SISO equalizers based on soft interference cancellation [49] and linear MMSE filtering technique proposed by Tüchler *et. al.* [50, 51] can also be used for complexity reduction. Turbo processing involves iterating extrinsic information between inner decoder and outer decoder till convergence is achieved or the maximum number of iterations are reached. In order to obtain a better estimate of the transmitted codeword with turbo iterations, we need to design the codes carefully. A precoding technique [52] can be used with turbo equalization which results in significant interleaving gains due to the recursive nature of the precoded ISI channels. The code design and precoder design aspects for optimizing the performance of turbo equalization are shown in [53]. LDPC code design for turbo equalization with optimal and sub-optimal soft output equalizers is presented in [54, 55]. Recently a code design technique for ISI channels was presented in [56–58], where the authors provide a method to design LDPC code ensembles for a given ISI channel which achieves performance close to the i.i.d. capacity of an ISI channel. All of these code design techniques are channel specific and assumes the knowledge of communication channel at the transmitter.

F. Non-universality of Turbo Signal Processing

Consider a serially concatenated encoding scheme and corresponding iterative decoding scheme shown in Fig. 7. For the case of coded transmission over an ISI channel, 'Encoder-I' represents an outer code while 'Encoder-II' represents an ISI channel. On the receiver side, 'Decoder-II' represents a channel detector and 'Decoder-I' represents an outer decoder. To show the universality of this iterative decoding scheme, we need

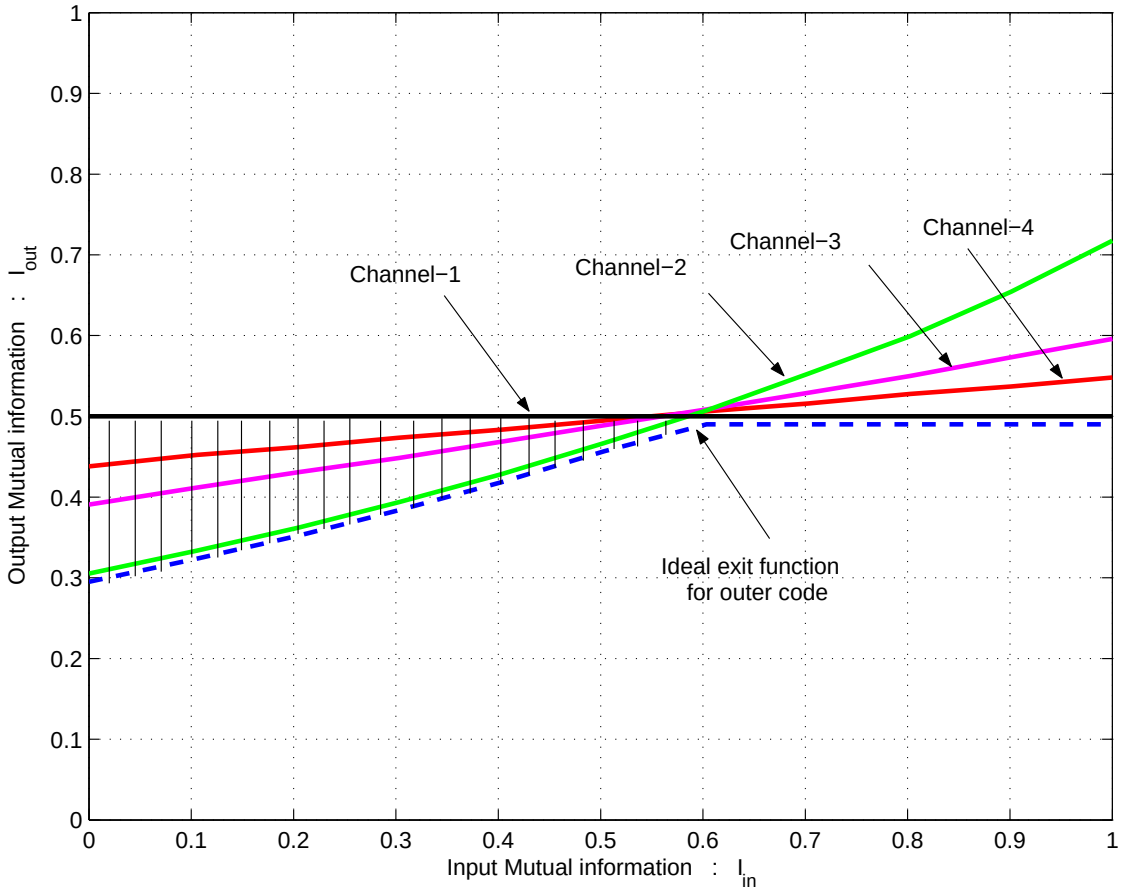

Fig. 8. Exit functions of different ISI channels along with ideal EXIT function for an outer code which maximizes the rate of a concatenated system.

to find a single outer code which performs universally well on all of the ISI channels under consideration. Here we outline the proof from [7], which shows with the help of EXIT (extrinsic information transfer) chart that for iterative receivers it is impossible to design one code that is good for several ISI channels with the same i.i.d. capacity. The EXIT chart, which plot the extrinsic output mutual information versus extrinsic input mutual information for given decoding blocks in a concatenated system was introduced in [59] and is extensively used to track the performance of iterative receivers. Figure 8 shows the EXIT functions for four different channels evaluated at signal to

noise ratio corresponding to the i.i.d. capacity of each channel for an outer code of rate $r = 0.5$. An AWGN channel was used as the *a priori* channel to obtain these curves. The area property of EXIT chart for erasure channel, which relates the area under the EXIT function to the conditional entropy was proved in [60]. In general this property is believed to be true for the case of Gaussian channels [61]. Consequence of this property is that for a serial concatenation system shown in Fig. 7, the area under the EXIT function an ISI channel (inner code) is equal to the maximum code rate of an outer code that the channel can support. This maximum code rate is equal to 0.5 for all four channels shown in Fig. 8. Due to the area property of an EXIT chart, code design for iterative receivers reduces to a curve fitting problem. Curve fitting approach for code design has been used by many researchers [62, 63]. EXIT chart is completed by drawing an inverted EXIT function (EXIT function with X and Y axis interchanged) of an outer decoder along with an EXIT function of inner decoder (owing to the fact that output of one decoder is input to other decoder). Progress of iterative decoding can be tracked by drawing staircase lines between the gap of two EXIT functions [59]. In order to guarantee the convergence to a correct codeword on a given channel, the inverted EXIT function of an outer code should not cross that channel EXIT function. In order to be a universal code, inverted EXIT function of an outer code should not cross EXIT function of any of the ISI channel under consideration. Also in order to maximize the rate of an outer code, its inverted EXIT function should lie just below the EXIT functions of all the channels. Thus EXIT function of a universal code under iterative processing is a convex hull of the EXIT functions of the channels, which is indicated by a dotted line in Fig. 8. We see that this universal code suffers from a code rate loss equal to the area of shaded region shown in Fig. 8. Thus even if all ISI channels can support the code rate equal to r, universal code under such an iterative scheme can not transmit reliably with

rate equal to r.

CHAPTER IV

BCJR-DFE RECEIVER[†]

The use of turbo principle for iterative decoding and signal processing is explained in chapter III. Such an iterative scheme is non-universal [7] and its performance depends on the channel realization which may not be known at the transmitter. In this chapter we develop a non-iterative receiver that performs joint decoding and signal processing through a decision feedback mechanism. With analysis and simulation results we show that the proposed receiver provides a universal solution for the case of constrained input Gaussian ISI channels. In this chapter, we consider coding and equalization for ISI channels.

A. Encoder Structure and Notation

The encoder structure is shown in Fig. 9. The user data is encoded with an LDPC code of rate r and length n. Encoded codewords are arranged in the form of a data-matrix of size $n \times m$ bits. The first L columns are a sequence of zero bits or known bits. Each of the last $m - L$ columns are codewords of an LDPC code. We will assume BPSK as the modulation; extension to other memoryless modulation is straight-forward. During the transmission, bits are transmitted sequentially along the rows, i.e. from 1st row to nth row. Before we proceed further, we explain some notation. We use upper case letters to denote random variables and lower case letters to denote their realizations. Operator $E[\ \cdot\ ]$ denotes the expectation operator and $E_A[\ \cdot\ ]$ denotes expectation w.r.t. random variable A. Let us denote $\boldsymbol{X} = [X_1, X_2, \ldots, X_{nm}]$ as a

[†]©2004 Allerton Conference. Reprinted, with permission, from "A BCJR-DFE based receiver for achieving near capacity performance on ISI channels", N. Nangare and K.R. Narayanan, *Allerton Conference.*, Sept. 2004.

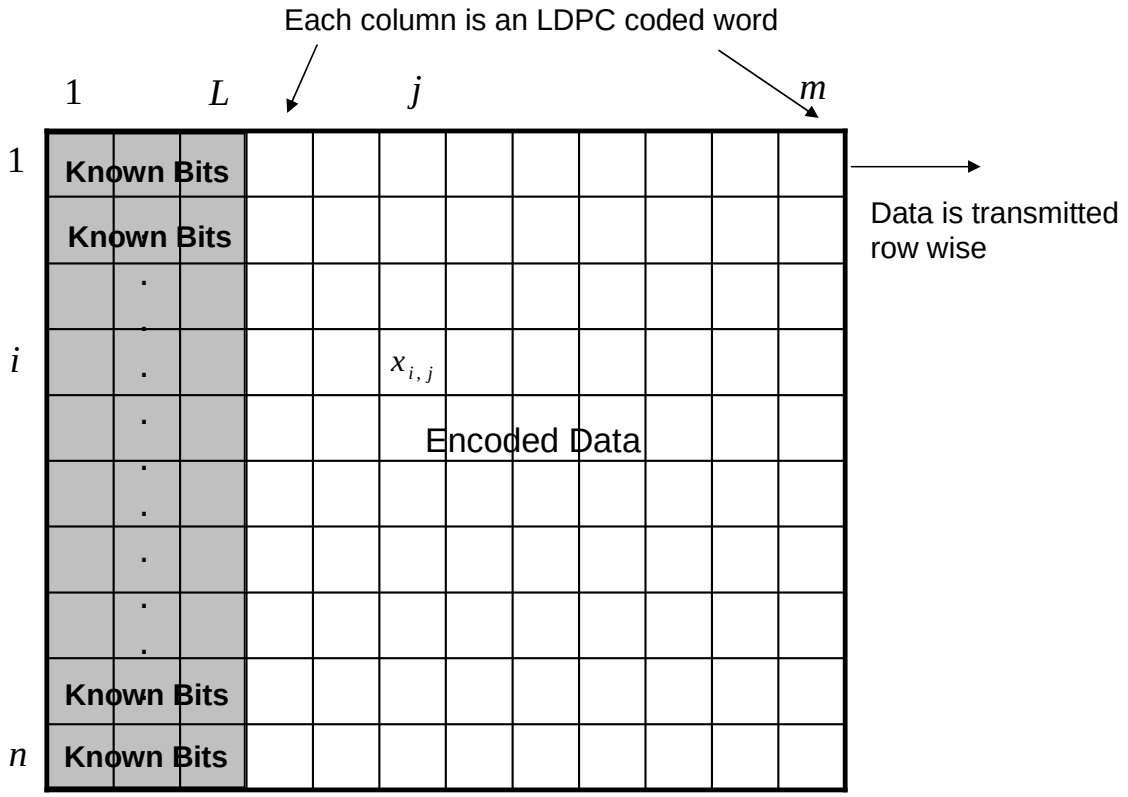

Fig. 9. Encoder structure for the proposed coding scheme

sequence of transmitted bits and $\boldsymbol{Y} = [Y_1, Y_2, \ldots, Y_{nm}]$ as a received sequence. We will use $\boldsymbol{x}$ and $\boldsymbol{y}$ to denote the particular realization of entire sequence of input and received vectors respectively, i.e. $\boldsymbol{x} = [x_1, \ldots, x_{nm}]$ and $\boldsymbol{y} = [y_1, \ldots, y_{nm}]$. We also use $X_{i,j}$ (with corresponding realization $x_{i,j}$) to refer to the transmitted bit in the ith row and the jth column in the data-matrix in Fig. 9. Therefore, $X_{i,j}$ and $X_{(i-1)m+j}$ refer to the same bit. Received sequence $\boldsymbol{Y}$ is also arranged in the same data-matrix structure. We use $Y_{i,j}$ (with corresponding realization $y_{i,j}$) to refer to the received signal in the ith row and the jth column in the received data-matrix. Similarly, $Y_{i,j}$ and $Y_{(i-1)m+j}$ refer to the same received signal. It may seem strange that a two dimensional index and a one dimensional index is used to denote the same signal, but it will simplify our notation later on. The sequence of L bits $[X_{i,1}, X_{i,2}, \ldots, X_{i,L}]$ are known bits $\forall i$. The overall rate is $R = rn(m-L)/(nm) = r(1-L/m)$. For $m \to \infty$, the rate becomes r. The sequence of bits $\{X_k\}$ are transmitted through a $L+1$ tap ISI channel (memory L). The received sequence $\{Y_k\}$ at the output is given by

$$Y_k = \sum_{i=0}^{L} h_i \; X_{k-i} + N_k, \tag{4.1}$$

where h_i's are the tap coefficients (assumed to be complex) and N_k is complex additive white Gaussian noise. We will use superscript to refer to vector of signals in a column, i.e., $\boldsymbol{X}^j$ is the vector of bits in the jth column, e.g $\boldsymbol{X}^j = [X_{1,j}, X_{2,j}, \ldots, X_{n,j}]$. We will use subscript to denote a vector of signals in a row, e.g $\boldsymbol{Y}_i$ denotes the vector of received bits corresponding to the transmitted bits in the ith row, $\boldsymbol{Y}_i = [Y_{i,1}, Y_{i,2}, \ldots, Y_{i,m}]$. We will use $\boldsymbol{Y}_{i,j}^{p,q}$ to denote the vector of bits between the time instants (i,j) and (p,q), including both. The ISI channel can be represented using a trellis with 2^L states, where the state before the transmission of $X_{i,j}$ is denoted by $S_{i,j}$. The state $S_{i,j}$ corresponds to the past L bits, $[X_{(i-1)m+j-L}, X_{(i-1)m+j-L+1}, \ldots, X_{(i-1)m+j-1}]$.

It is assumed that the channel state is reset to zero state at the end of the transmission of last row of a data-matrix.

B. Proposed Receiver Structure

The receiver begins by computing optimal soft outputs for each of the bits in the $(L+1)$th column of the data-matrix shown in Fig. 9. We assume that a trellis based equalizer is used for equalization. Note that for each $X_{i,L+1}$, the previous L bits $(X_{i,1}, X_{i,2}, \ldots, X_{i,L})$ are known bits. Hence, the optimal soft output is the likelihood ratio

$$\Lambda(x_{i,L+1}) = \frac{P(X_{i,L+1} = 1|\boldsymbol{Y}, X_{i,1}, X_{i,2}, \ldots, X_{i,L})}{P(X_{i,L+1} = -1|\boldsymbol{Y}, X_{i,1}, X_{i,2}, \ldots, X_{i,L})} \tag{4.2}$$

We will use $\Lambda_{i,j}$ to denote $\Lambda(x_{i,j})$, the soft output for the bit in the (i,j)th position. The corresponding realization is denoted by $\lambda_{i,j}$. When one dimensional indexing is used then $\Lambda_{(i-1)m+j}$ (with corresponding realization $\lambda_{(i-1)m+j}$) is same as $\Lambda_{i,j}$. Since knowing the past L bits perfectly is equivalent to fixing the state in the trellis of the equalizer, the optimal soft output is given by

$$\Lambda_{i,L+1} = \frac{P(X_{i,L+1} = 1|\boldsymbol{Y}, S_{i,L+1})}{P(X_{i,L+1} = -1|\boldsymbol{Y}, S_{i,L+1})} \tag{4.3}$$

Using (4.3), we can calculate the soft outputs $\Lambda_{i,L+1}$ of an equalizer for $1 \leq i \leq n$, which corresponds to the soft inputs for the LDPC codeword at $(L+1)$th column. The next step is to run a decoder for the code $\boldsymbol{X}^{L+1}$ and decode it. If we assume that $n \to \infty$ and the code $\boldsymbol{X}^{L+1}$ achieves capacity on the equivalent channel characterized by the input $X_{i,L+1}$ and output $\Lambda_{i,L+1}$, then we can perfectly decode the $(L+1)$th column. Now the decoder uses these decisions as perfect decision feedback and proceeds to generate optimal soft output for the bits in the $(L+2)$th column assuming the previous L bits are perfectly known. This process continues and for the jth column, the

equalizer produces optimal soft outputs $\Lambda_{i,j}$ for all the bits in the jth column assuming the previous L bits are perfectly known i.e

$$\Lambda_{i,j} = \frac{P(X_{i,j} = 1|\boldsymbol{Y}, S_{i,j})}{P(X_{i,j} = -1|\boldsymbol{Y}, S_{i,j})} \tag{4.4}$$

We will prove several interesting properties of this receiver but before that, we will show how to generate the optimal soft outputs $\Lambda_{i,j}$ in (4.4) using just one pass of the BCJR algorithm.

C. The BCJR-DFE Algorithm

First let us begin by noting that due to the Markov nature of the trellis,

$$P(X_{i,j}|\boldsymbol{Y}, S_{i,j}) = P(X_{i,j}|\boldsymbol{Y}_{i,j}^{n,m}, S_{i,j})$$

That is, given the state $S_{i,j}$, the past received values do not affect the probabilities at time i, j. Given that $S_{i+1,L+1}$ is also fixed, the received values in the $(i+1)$th row do not affect the soft outputs for the bits in the ith row. Hence,

$$P(X_{i,j}|\boldsymbol{Y}_{i,j}^{n,m}, S_{i,j}) = P(X_{i,j}|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j}) \tag{4.5}$$

Therefore, we can replace $\boldsymbol{Y}$ in (4.4) by $\boldsymbol{Y}_{i,j}^{i,m}$ for the jth bit in the ith row. Therefore,

$$\Lambda_{i,j} = \frac{P(X_{i,j} = 1|\boldsymbol{Y}, S_{i,j})}{P(X_{i,j} = -1|\boldsymbol{Y}, S_{i,j})} = \frac{P(X_{i,j} = 1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})}{P(X_{i,j} = -1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})} \tag{4.6}$$

That is, we can run n BCJR equalizers in parallel, one for each row. So without loss of generality let us consider the ith row.

$$\frac{P(X_{i,j} = 1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})}{P(X_{i,j} = -1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})} = \frac{P(\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j}|X_{i,j} = 1)}{P(\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j}|X_{i,j} = -1)} \frac{P(X_{i,j} = 1)}{P(X_{i,j} = -1)} \tag{4.7}$$

Equation (4.7) can be further simplified as below where $P(X_{i,j} = 1) = P(X_{i,j} = -1)$ is assumed.

$$\frac{P(X_{i,j} = 1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})}{P(X_{i,j} = -1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})} = \frac{P(\boldsymbol{Y}_{i,j}^{i,m}|S_{i,j}, X_{i,j} = 1)\ P(S_{i,j}, X_{i,j} = 1)}{P(\boldsymbol{Y}_{i,j}^{i,m}|S_{i,j}, X_{i,j} = -1)\ P(S_{i,j}, X_{i,j} = -1)} \tag{4.8}$$

For a sequence of uniform i.i.d. bits, the second term in the numerator and the denominator in the RHS are the same and, hence, can be dropped. The first term can be split into two as follows

$$\frac{P(X_{i,j} = 1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})}{P(X_{i,j} = -1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})} = \frac{P(Y_{i,j}|S_{i,j}, X_{i,j} = 1)\ P(\boldsymbol{Y}_{i,j+1}^{i,m}|S_{i,j}, X_{i,j} = 1)}{P(Y_{i,j}|S_{i,j}, X_{i,j} = -1)\ P(\boldsymbol{Y}_{i,j+1}^{i,m}|S_{i,j}, X_{i,j} = -1)} \tag{4.9}$$

Let S' be the future state when $X_{i,j} = 1$ and current state is equal to $S_{i,j}$. Similarly let S'' be the future state when $X_{i,j} = -1$ and current state is equal to $S_{i,j}$. Then, the above equations reduce to

$$\Lambda_{i,j} = \frac{P(X_{i,j} = 1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})}{P(X_{i,j} = -1|\boldsymbol{Y}_{i,j}^{i,m}, S_{i,j})} = \frac{\gamma_{i,j}(+1, S_{i,j}, S')\ \beta_{i,j+1}(S')}{\gamma_{i,j}(-1, S_{i,j}, S'')\ \beta_{i,j+1}(S'')} \tag{4.10}$$

where $\beta_{i,j+1}(S') = P(\boldsymbol{Y}_{i,j+1}^{i,m}|S_{i,j+1} = S')$ and $\gamma_{i,j}(a, S_{i,j}, S') = P(Y_{i,j}|S_{i,j}, S_{i,j+1} = S', X_{i,j} = a)$ are the standard definitions of backward state probability and state transition probability respectively in a BCJR algorithm [8]. This can be realized as follows. We first perform the backward recursion in the BCJR algorithm to generate the β's for each stage. Then, in the forward pass, for each j, we only need to know the state $S_{i,j} = s_{i,j}$ in order to compute (4.10). This is provided by the DFE, that is when the past L bits are known, $s_{i,j}$ is fixed. Hence, we effectively use only the backward recursion of a BCJR algorithm and simply set the known states during the forward recursion as we decode each column in our codeword data-matrix. For every coded bit, we have only one pass of the BCJR algorithm to produce $\Lambda_{i,j}$. Hence the equalization complexity is nearly half that of a BCJR algorithm. Both MMSE-DFE and BCJR-DFE use past decisions to cancel ISI. The feed forward section in the

MMSE-DFE is a linear MMSE filter, whereas in the BCJR-DFE, the feed forward filter is a non-linear estimator that produces optimal soft decisions. This is the main difference between the MMSE-DFE and the BCJR-DFE.

D. Properties of BCJR-DFE Receiver

We will prove some interesting properties of the proposed receiver in this section. First consider the output of the BCJR-DFE, $\Lambda_{i,j}$ for a fixed column j and $\forall i$. The $\Lambda_{i,j}$'s are random variables and, hence, we can think of them as the output of some equivalent channel $\mathcal{C}_j$ that has input $X_{i,j}$ and output $\Lambda_{i,j}$.

Proposition 1: The equivalent channel along each column $\mathcal{C}_j : L < j \leq m$ is a memoryless channel.

Proof: To show that the channel is memoryless, we need to show that for a fixed column j,

$$P((\Lambda_{i,j}, \Lambda_{k,j})|(X_{i,j}, S_{i,j}, X_{k,j}, S_{k,j})) = P(\Lambda_{i,j}|X_{i,j}, S_{i,j})\ P(\Lambda_{k,j}|X_{k,j}, S_{k,j}).$$

Note that $\Lambda_{i,j}$ is a function only of $\boldsymbol{Y}_{i,j}^{i,m}$ and $S_{i,j}$, and $\Lambda_{k,j}$ is a function only of $\boldsymbol{Y}_{k,j}^{k,m}$ and $S_{k,j}$. Note that $\boldsymbol{Y}_{i,j}^{i,m}$ and $\boldsymbol{Y}_{k,j}^{k,m}$ are conditionally independent given $S_{i,j}, X_{i,j}, S_{k,j}, X_{k,j}$ This is because of the known bits at the end of each row, the trellis is brought back to the known state. Hence, $\Lambda_{i,j}$ and $\Lambda_{k,j}$ are conditionally independent given $S_{i,j}, S_{k,j}, X_{i,j}$ and $X_{k,j}$.

Proposition 2: The equivalent channel $\mathcal{C}_j$ is statistically identical to that of $\mathcal{C}_l,\ L < j \leq m,\ L < l \leq m$.

Proof: This is true since all processes considered are stationary. The presence of known bits ensures that for all the channels, the past L decisions are perfectly known. Since the channels are all the same, we will drop the subscript and call it as $\mathcal{C}$. Propo-

sition 1 and 2 in effect mean that, one code of rate $r = I(X_{i,j}, \Lambda_{i,j})$ suffices to be used along each column in the codeword matrix. As $n \to \infty$, by choosing this code to achieve capacity on the memoryless channel $\mathcal{C}$ we can generate perfect decisions in the proposed receiver structure.

Theorem 1: The BCJR-DFE receiver is an optimal receiver that achieves the i.i.d information rate of the ISI channel in (3.10) as $n, m \to \infty$. That is, $r = C_{i.i.d}$ of the ISI channel.

Proof: We will use one dimensional indices for X, Y and Λ in this section.

$$C_{i.i.d} = \frac{1}{nm} I(\boldsymbol{X}; \boldsymbol{Y}) = \frac{1}{nm} I(X_1, X_2, \ldots, X_{mn}; \boldsymbol{Y}) \tag{4.11}$$

Using the chain rule of mutual information

$$C_{i.i.d} = \frac{1}{nm} \sum_{i=1}^{nm} I(X_i; \boldsymbol{Y} | X_{i-1}, X_{i-2}, \ldots, X_1) \tag{4.12}$$

Since knowing the previous L bits fixes the state of the encoder trellis, we have

$$P(X_i | \boldsymbol{Y}, X_{i-1}, \ldots, X_1) = P(X_i | \boldsymbol{Y}, X_{i-1}, \ldots, X_{i-L}) \tag{4.13}$$

and, hence,

$$I(X_i; \boldsymbol{Y} | X_{i-1}, \ldots, X_1) = I(X_i; \boldsymbol{Y} | X_{i-1}, \ldots, X_{i-L}). \tag{4.14}$$

The above information rate is identical for all i (using proposition 2). Hence,

$$\begin{aligned}
C_{i.i.d} &= I(X_i; \boldsymbol{Y} | X_{i-1}, \ldots, X_{i-L}) \\
&= H(X_i | X_{i-1}, \ldots, X_{i-L}) - H(X_i | \boldsymbol{Y}, X_{i-1}, \ldots, X_{i-L}) \\
&= 1 + E_{\boldsymbol{Y}, X_i, X_{i-1}, \ldots, X_{i-L}} \big[\log(P(x_i | \boldsymbol{y}, x_{i-1}, \ldots, x_{i-L})) \big] \\
&= 1 + E_{\boldsymbol{Y}, X_{i-1}, \ldots, X_{i-L}} \big[P(x_i = +1 | \boldsymbol{y}, x_{i-1}, \ldots, x_{i-L}) \log(P(x_i = +1 | \boldsymbol{y}, x_{i-1}, \ldots, x_{i-L})) \\
&+ P(x_i = -1 | \boldsymbol{y}, x_{i-1}, \ldots, x_{i-L}) \log(P(x_i = -1 | \boldsymbol{y}, x_{i-1}, \ldots, x_{i-L})) \big]
\end{aligned} \tag{4.15}$$

Using the definition of likelihood ratio given in (4.2) we can write the conditional probabilities in (4.15) as,

$$P(x_i = +1|\boldsymbol{y}, x_{i-1}, \ldots, x_{i-L}) = \frac{\lambda_i}{1+\lambda_i} \tag{4.16}$$

$$P(x_i = -1|\boldsymbol{y}, x_{i-1}, \ldots, x_{i-L}) = \frac{1}{1+\lambda_i} \tag{4.17}$$

Substituting (4.16) and (4.17) in (4.15) we get,

$$C_{i.i.d} = 1 + E_{\Lambda_i}\left[\frac{\lambda_i}{1+\lambda_i}\log\left(\frac{\lambda_i}{1+\lambda_i}\right) + \frac{1}{1+\lambda_i}\log\left(\frac{1}{1+\lambda_i}\right)\right] \tag{4.18}$$

The proposed receiver produces $\Lambda_i = \frac{P(X_i=1|\boldsymbol{Y},X_{i-1},\ldots,X_{i-L})}{P(X_i=-1|\boldsymbol{Y},X_{i-1},\ldots,X_{i-L})}$ and hence we can write the information rate for the equivalent channel $\mathcal{C}$ between X_i and Λ_i as,

$$\begin{aligned}
r &= I(X_i;\Lambda_i) \\
&= H(X_i) - H(X_i|\Lambda_i) \\
&= 1 + E_{X_i,\Lambda_i}[\log(P(x_i|\lambda_i)] \\
&= 1 + E_{\Lambda_i}\big[P(x_i=+1|\lambda_i)\log(P(x_i=+1|\lambda_i)) + P(x_i=-1|\lambda_i)\log(P(x_i=-1|\lambda_i))\big] \\
&= 1 + E_{\Lambda_i}\left[\frac{\lambda_i}{1+\lambda_i}\log\left(\frac{\lambda_i}{1+\lambda_i}\right) + \frac{1}{1+\lambda_i}\log\left(\frac{1}{1+\lambda_i}\right)\right] \qquad (4.19)
\end{aligned}$$

From (4.18) and (4.19) we can say that the achievable information rate $r = C_{i.i.d.}$. There is a small capacity loss due to the known bits. Therefore the achievable information rate with the proposed scheme is $r = C_{i.i.d}(1 - \frac{L}{m})$ and, hence,

$$\lim_{m\to\infty} r = C_{i.i.d} \tag{4.20}$$

This proves that $C_{i.i.d}$ is achievable with an ideal BCJR-DFE receiver, if each column code achieves capacity on a memoryless channel.

Proposition 4: For any Markov input (non i.i.d inputs), the information rate is achievable with the BCJR-DFE receiver.

Proof: Consider a Markov input which is generated by passing the sequence $\{X_i\}$'s through a rate b/d trellis encoder. Then we can express the combination of the trellis encoder and the channel as one channel which is driven by an i.i.d sequence. In this case, the equivalent channel taken for a group of b bits together will be the same for all columns and, hence, codes that achieve capacity on that channel will also achieve the information rate with the Markov input. The channel for each of the b bits may be different hence, b different codes for single input memoryless channels will be required.

E. Universality of BCJR-DFE Receiver

Strictly speaking, the equivalent channel $\mathcal{C}$ is a function of the channel taps and, hence, in order to achieve capacity on this channel, the ISI channel must be known at the transmitter. For the MMSE-DFE, the assumption that the equivalent channel is an AWGN channel has been used widely. So, we believe that this would hold for the BCJR-DFE also. Even if this does not hold true, it is possible to find good codes that provide good performance on a variety of memoryless single input channels. For example, it was observed in [37, 41] that LDPC code ensemble optimized for the AWGN channel provides good performance for several other memoryless channels also (see chapter III, section A-1). Thus, it is reasonable to expect that we can find one code of rate r, capable of providing good performance on any ISI channel as long as $C_{i.i.d} > r$. Threshold computations in section H will corroborate this. This essentially solves the Root-Varaiya problem [6] for ISI channels with constrained input (see chapter III, section D). Let us assume that we have a class of Gaussian ISI channels $\{H_1, H_2, \cdots\}$, such that $C_{i.i.d.}(H_i) > r$. Then BCJR-DFE receiver along with the proposed data-matrix encoding scheme with rate equal to r provides small error probability uniformly across any of the channel $\{H_1, H_2, \cdots\}$. This result is

quite useful in wireless communications, where the most likely scenario is that the transmitter does not know the channel but the receiver does.

F. Comparison to Iterative Receivers

Compared to using turbo equalization, the proposed receiver has the following advantages - the complexity is significantly smaller since only one iteration of BCJR equalizer is required. Also it is not required to calculate and store the forward state probabilities (α's) [8] in order to calculate the likelihood ratios (see (4.10)). With the proposed receiver, we do not need to specially design codes for ISI channels such as in [56]. Through simulation results we show that this code can even be chosen to be a code optimized for an AWGN channel. The BCJR-DFE receiver with an AWGN optimized code of rate r performs universally well across the class of several ISI channels as long as the i.i.d. capacity of the channel is greater than r. Whereas in chapter III, we outlined the proof from [7] which shows that for iterative receivers it is impossible to design one code that is good for several ISI channels with the same $C_{i.i.d}$. The disadvantage of the proposed approach is that for a given delay of nm, the codewords can be only of length n, whereas in the case of turbo equalization, the codewords can be length nm. With codes like LDPC codes whose performance improves with length, it may be a disadvantage of the proposed structure. When $n \geq 1000$, usually the sphere packing bound is only less than a dB away from the capacity limit, meaning that we may at most loose 1 dB or so compared to infinite length codewords. As we can see from the following sections that even if n can be large, BCJR-DFE receiver outperforms iterative receiver. Secondly, even if nm is very large, in practice it is quite hard to implement LDPC decoders for such long block lengths due to the hardware constraints. For BCJR-DFE implementation, n parallel equalizers

are not necessary. Each row can be processed serially with a single equalizer and backward state probabilities of whole data-matrix can be stored in memory. Since the BCJR-DFE algorithm is symmetric, instead of performing backward pass first and doing column decoding in forward direction (column-$L+1$ through column-m), we can perform forward pass first and do the column decoding in backward direction (column-m through column-$L+1$). This allows us to start calculating the forward state probabilities as soon as samples are received.

G. Discussion

The proposed receiver structure is not entirely new. There are at least four papers that are closely related to this work. However, there are differences between these structures, which are briefly summarized below. The proposed receiver is motivated from the classic two part paper by Cioffi *et al.* [64, 65] where they showed that minimum mean squared error (MMSE) decision feedback equalization (DFE) with error free decision feedback is a canonical structure and predicts the performance of coding schemes accurately for any ISI channel. Here, we show by replacing the MMSE filter by a BCJR algorithm [8], with ideal feedback we can exactly achieve capacity for any SNR and any constrained input constellation. We propose a computationally efficient BCJR-DFE algorithm, whose complexity is almost half that of the conventional BCJR algorithm. In order to achieve perfect decision feedback equalization, Cioffi *et al.* proposed to cancel the ISI at the transmitter through precoding which requires channel to be known at the transmitter. In the BCJR-DFE receiver, the DFE is at the receiver and, hence, the channel knowledge is not required at the transmitter. It must be noted that Varanasi and Guess [66] have also shown the optimality of ideal decision feedback multi-user detection. Although this encoder structure was derived

to facilitate a DFE in the receiver, in effect this is nearly equivalent to the concept of interleaved multiplexed codewords that Pfister, Soriaga and Siegel used to derive information rates for ISI channels in their landmark paper [46]. In [46], authors used the idea of m interleaved codes each of a different rate (the rates are usually dependent on the channel) and a multi-stage decoder which involves m uses of an *a posteriori* probability (APP) decoder. The encoder structure here can be thought of as m interleaved codewords but with some known symbols added between the codewords. The presence of these known symbols makes the channel completely memoryless. Hence, the receiver structure proposed here does not require m uses of an APP decoder as in [46], but it requires only one APP decoder. Further, m different codes are not required. Only one code of a fixed rate (almost independent on the channel taps but dependent only on the achievable information rate) is required. The receiver structure is also nearly the same that proposed in Fechtel and Meyr [67] and in Eyuboglu [68], where trellis codes are used along with DFE. The difference is in the use of the novel BCJR-DFE and capacity achieving LDPC code for a memoryless channel, both of which are required to achieve the capacity of an ISI channel. Further, we proved the optimality and utility of this scheme, which seems to have been forgotten in favor of turbo equalization.

H. Simulation Results

The performance was simulated on three ISI channels - the dicode channel with tap coefficients $[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$, the three-tap ISI channel with tap coefficients $[0.407,\ 0.815,\ 0.407]$ and the EPR4 channel with tap coefficients $[0.5,\ 0.5,\ -0.5,\ -0.5]$. The three tap channel is the channel with severe ISI taken from [69]. An LDPC code with rate 0.5 which was optimized for the AWGN channel, taken from [41] with maximum variable

Table III. Degree distribution of AWGN optimized LDPC code of rate $r = 0.5$

λ_2	0.21991
λ_3	0.23328
λ_4	0.02058
λ_6	0.08543
λ_7	0.06540
λ_8	0.04767
λ_9	0.01912
λ_{19}	0.08064
λ_{20}	0.22798
ρ_8	0.64854
ρ_9	0.34747
ρ_{10}	0.00399

node degree of 20 was used. Degree distribution of the chosen LDPC code is given in Table III. We used bit-filling algorithm [70] for the construction of LDPC graph for all LDPC codes simulated in this dissertation, unless otherwise stated. Sum-product algorithm was used for LDPC decoding with maximum number of iterations in LDPC decoder set to 100. In Figs. 10, 11 and 12, we plot the following curves for each of these channels, respectively - the i.i.d capacity, threshold for the LDPC code used with the BCJR-DFE receiver computed using density evolution, threshold for the same code used with turbo equalization, simulation results for $n = 10000$ for the BCJR-DFE receiver and simulation results for $n = 10000$ and for $n = 10^6$ for turbo equalization. In computing the threshold with the BCJR-DFE receiver,

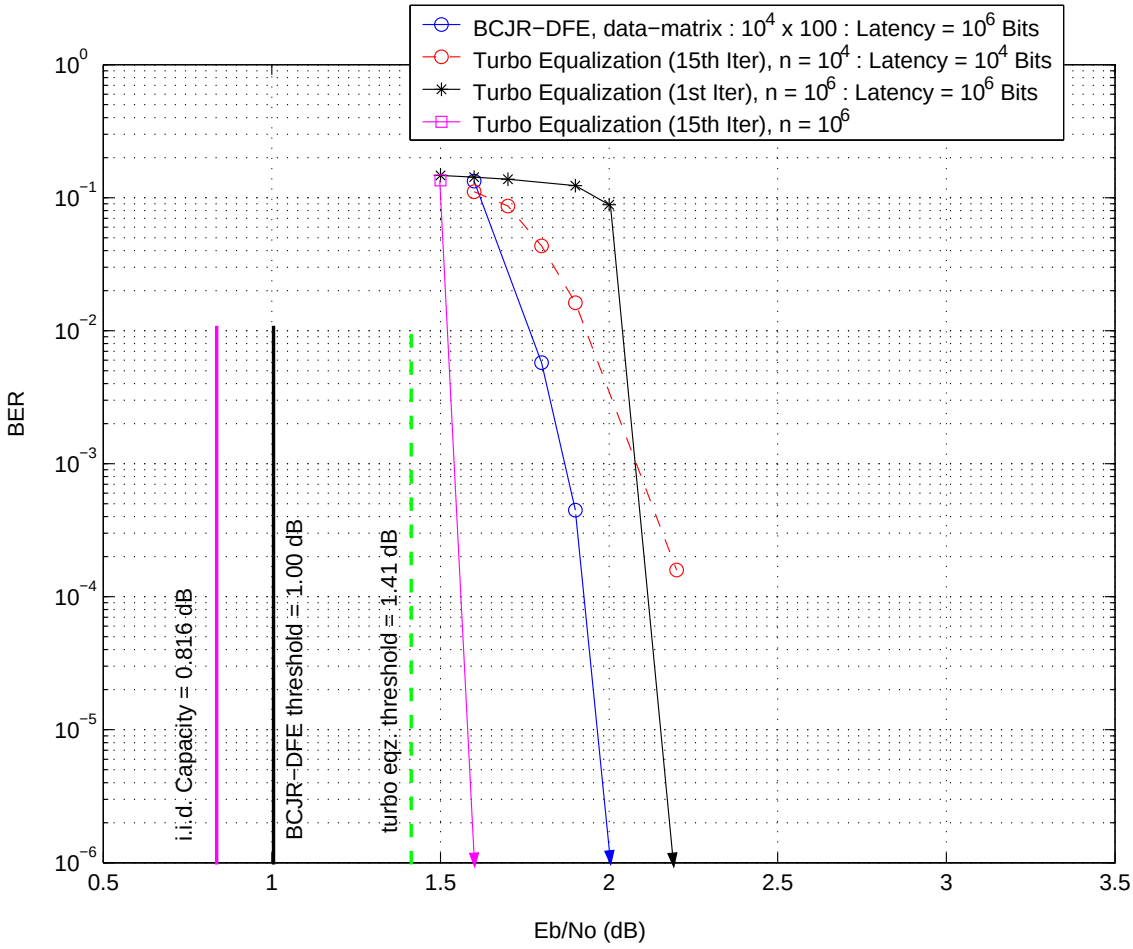

Fig. 10. Simulation results for the dicode channel $H = [\frac{1}{\sqrt{2}} \ -\frac{1}{\sqrt{2}}]$ with BCJR-DFE

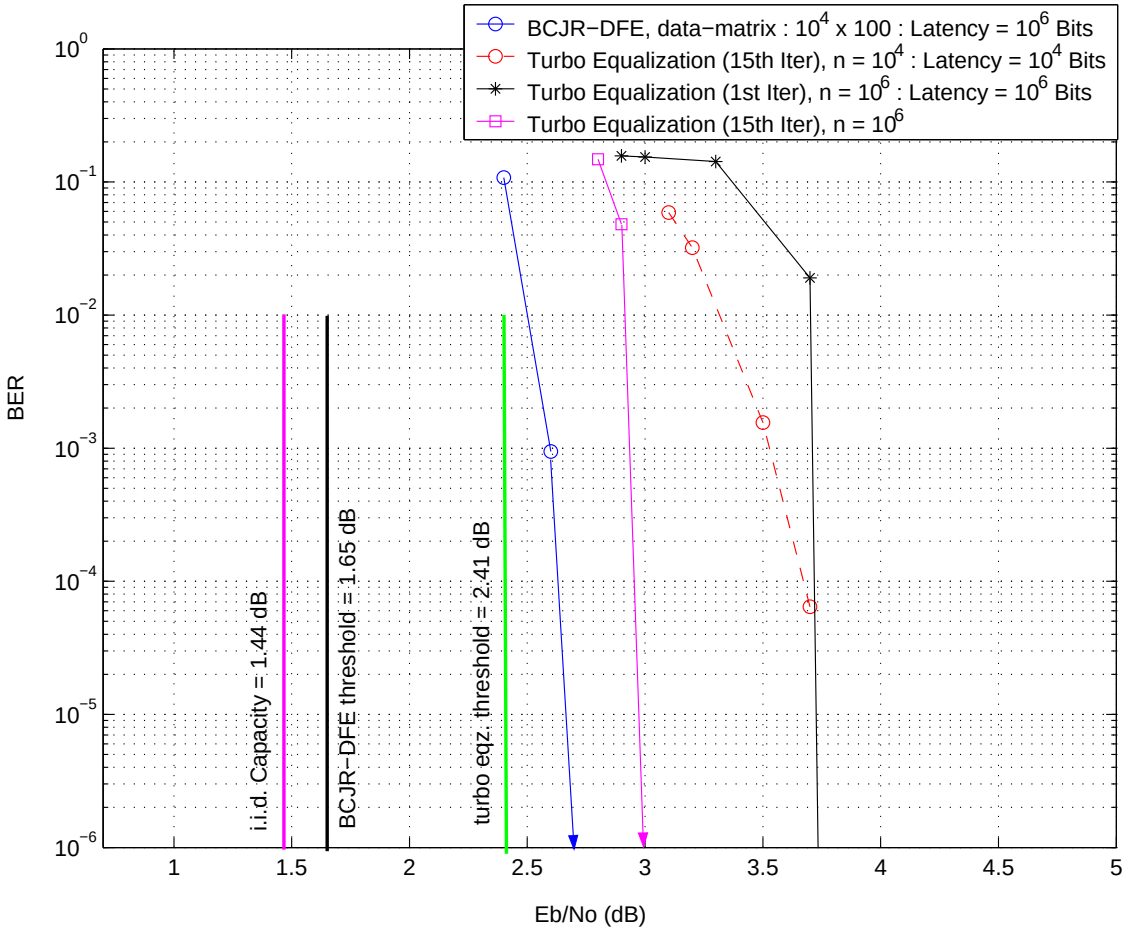

Fig. 11. Simulation results for the three tap channel $H = [0.407\ 0.815\ 0.407]$ with BCJR-DFE

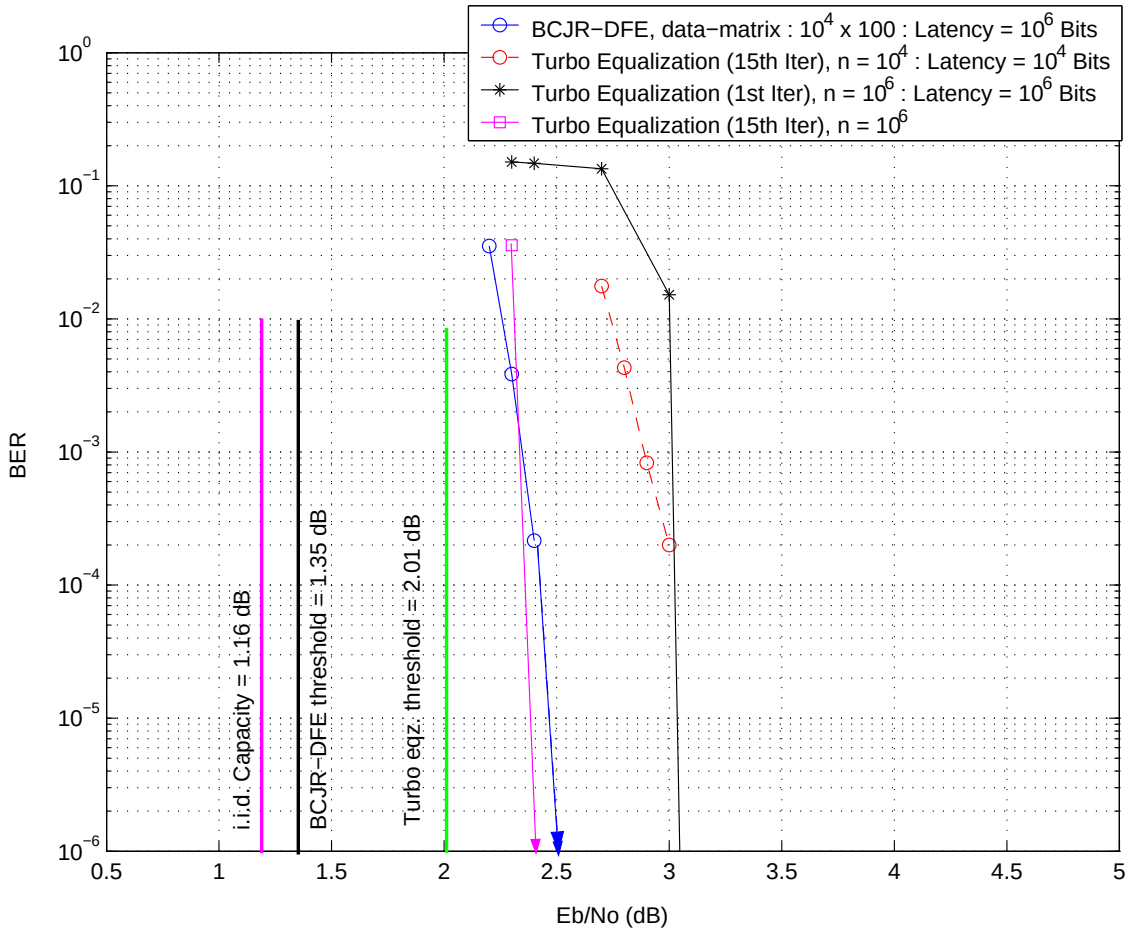

Fig. 12. Simulation results for the EPR4 channel $H = (0.5\ 0.5\ -0.5\ -0.5)$ with BCJR-DFE

Table IV. Gap of BCJR-DFE threshold and turbo equalization threshold from $C_{i.i.d}$ for different ISI channels

Channel	BCJR-DFE Threshold (dB)	Turbo Equalization Threshold (dB)
AWGN	0.1234	0.1234
$[\frac{1}{\sqrt{2}} \ -\frac{1}{\sqrt{2}}]$	0.184	0.594
[0.407 0.815 0.407]	0.21	0.97
[0.5 0.5 -0.5 -0.5]	0.19	0.85
200 Complex random channels with 3 rayleigh fading taps	0.21	≥ 0.97

we have ignored the E_b/N_0 loss due to pilots (as we assume $m, n \to \infty$). For the finite length simulations, the loss is accounted for. To compute the threshold with turbo equalization we used Gaussian approximation for input and output messages for both equalizer and LDPC decoder. Table IV compares the threshold of BCJR-DFE receiver with the threshold of turbo equalization for different channels. For the dicode channel, for the three tap channel and the EPR4 channel, the threshold with BCJR-DFE is only about 0.184 dB, 0.21 dB, and 0.19 dB away from that of the i.i.d capacity respectively. It can be seen that the thresholds with BCJR-DFE are significantly better than the thresholds with turbo equalization for the same code. In order to show the universality of the proposed receiver, we selected 200 complex ISI channels of length 3, with real and imaginary part drawn from Gaussian distribution with zero mean and unit variance i.e. $\mathcal{N}(0, 1)$. For each of these 200 complex random channels, we calculated the i.i.d capacity and the threshold for the LDPC code used

with the BCJR-DFE receiver using density evolution. The maximum gap between the i.i.d. capacity and the corresponding threshold with BCJR-DFE was found to be only 0.21 dB. This shows that the proposed receiver can achieve near capacity performance with a code that is optimized for the AWGN channel. The universality of the proposed receiver can be seen from the fact that the gap between i.i.d capacity and threshold is same for all the channels considered and is only slightly higher (0.0866 dB) than that for the AWGN channel. That is, one code is able to provide good performance on the AWGN channel and all of these ISI channels suggesting that this can be a good solution to the Root-Varaiya problem for ISI channels with constrained inputs. However in the case of turbo equalization, the gap between i.i.d capacity and threshold is nearly 0.844 dB, 0.97 dB and 0.85 dB for the dicode channel, the three tap channel and the EPR4 channel respectively, clearly showing that the performance of turbo equalization is strongly dependent on the channel realization. It is possible to obtain better performance from turbo equalization by optimizing the LDPC code for the specific channel. However, this will require that the channel be known at the transmitter. Since we consider the case when the channel is not known, we have used LDPC codes optimized for the AWGN channel. Simulation results with BCJR-DFE receiver for finite length LDPC codes with $n = 10000$ and $m = 100$ are also shown in Figs. 10, 11 and 12. It can be seen that the proposed scheme outperforms the turbo equalization case with $n = 10000$ and 15 turbo iterations for both the channels, the difference being nearly 1 dB for the three tap case. Since the overall latency is 10^6 bits, whereas the LDPC code is only of length 10^4 bits, in the turbo equalization case, an LDPC code of length 10^6 can be used. We have shown the performance for turbo equalization case with $n = 10^6$ for 1st iteration and 15th iteration, where the complexity and latency is same for the 1st iteration of turbo equalization and BCJR-DFE receiver and the complexity of turbo equalization at 15th iteration is more than 15

times higher than the complexity of BCJR-DFE receiver. It can be seen that for the three tap channel case, the proposed receiver with $n = 10000$ and latency equal to 10^6 bits outperforms the same latency turbo equalization result with 15 turbo iterations. For the EPR4 channel case the performance of proposed receiver with $n = 10000$ and latency equal to 10^6 bits is almost same as the same latency turbo equalization result. This clearly shows that even for comparison based on the same latency, the proposed BCJR-DFE receiver achieves better or same performance as turbo equalization over some channels with significant reduction in the complexity.

I. Conclusion

Cioffi *et al.* showed that an ideal MMSE-DFE equalizer is a canonical receiver. We have extended this to show that a BCJR-DFE equalizer is optimal (achieves the information rate) for any SNR and any finite sized input constellation. The DFE can be used at the receiver (instead of being used as a precoder at the transmitter) and LDPC codes that achieve capacity on an equivalent memoryless channel are optimal for the channel with memory. The complexity of this receiver is only that of one iteration of a turbo receiver. The solution is an universal solution in the sense that a single code of rate-r can provide small error probability uniformly across any linear, time invariant constrained input Gaussian channel with ISI, as long as the capacity of the channel is greater than r. We have shown through simulations that this code can even be chosen to be an LDPC code optimized for the AWGN channel.

CHAPTER V

LOW LATENCY TECHNIQUES FOR BCJR-DFE RECEIVER

Encoder structure proposed in chapter IV uses a data-matrix of size $n \times m$, which gives total latency of nm bits for the BCJR-DFE receiver. Large latency of BCJR-DFE receiver poses practical limitations on its applications. In this section we present two methods to reduce the latency of BCJR-DFE receiver. In the the first method we reduce the number of columns m, while in the second method we reduce the column length n. We also discuss the low complexity MMSE-DFE equalization in which optimal BCJR equalizer is replaced with sub-optimal linear MMSE equalizer.

A. Multi-rate BCJR-DFE

In the this section we show that by selecting proper rates for each column, we can reduce the latency of BCJR-DFE receiver significantly. The proposed data-matrix encoder structure (Fig. 9) for BCJR-DFE receiver uses a sequence of zero bits or known bits at the start of each row to reset the channel state memory. These L known columns formed by the known bits can be considered as a code with rate $r = 0$. These known columns account for the loss of $10 \log_{10}(1 - \frac{L}{m})$ dB in SNR, which becomes significant for small values of m and thus prevents us from reducing the number of columns m, to lower the latency of BCJR-DFE. For the design of multi-rate BCJR-DFE, we assume that channel is known at the transmitter. We can completely avoid the loss in SNR due to known bits if we use low rate codes (instead of rate zero code) for first L columns of the data-matrix. Rates for each columns of the data-matrix should be set by noting that previous state is not known completely but channel is known at the transmitter. In the absence of known bits, none of the previous bits are known for the bits in the first column of the data-matrix. So we can set the code rate

r_1 for the first column equal to the information rate for the given channel assuming none of the previous bits are known i.e. $r_1 = I(X_i; \boldsymbol{Y})$. For the second column we assume that we have decoded the first column successfully and hence the code rate r_2 can be set to the information rate for the given channel assuming previous 1 bit is known i.e $r_2 = I(X_i; \boldsymbol{Y}|X_{i-1})$. Thus we can set the rate for jth column assuming that previous $j-1$ bits are known i.e. $r_j = I(X_i; \boldsymbol{Y}|X_{i-1}, X_{i-2}, \cdots, X_{i-j+1})$ for $1 \leq j \leq L$. We also note that BCJR-DFE produces high values of LLRs for few last columns in the data-matrix due to end effect of a BCJR trellis. For example for all the bits in the last column of the data-matrix, assuming we have decoded previous columns perfectly, we know the previous L bits and also we know the future L bits. So we can set $r_m = I(X_i; \boldsymbol{Y}|X_{i+L}, \cdots, X_{i+1}, X_{i-1}, \cdots, X_{i-L})$. This allows us to increase the code rate for last few columns in the data-matrix, which can compensate for the low code rate in first L columns of the data-matrix. Thus we have a new encoder structure without any known bits, where code rate gradually increases from the first column to the last column in the data-matrix. Transmission of the data over an ISI channel is done as before, that is, the data is transmitted row wise from first row to the last row of the data-matrix. For decoding, we first perform a single LARGE backward pass from the last received bit of last row to the first received bit of the first row of the data-matrix. Backward state probabilities (β's) corresponding to all the bits are stored during this iteration. In the forward direction we use n parallel equalizers corresponding to each row of the data-matrix. For the first column we assign equal forward probabilities α's to all the starting states of n trellises, as none of the previous bits are known. We can then calculate the LLRs for the first column

$(j = 1)$ using standard BCJR equation [8] as

$$\Lambda_{i,j} = \frac{\sum_{\substack{S_{i,j}\to S_{i,j+1}: \\ X_{i,j}=+1}} \alpha_{i,j}(S_{i,j})\gamma_{i,j}(+1, S_{i,j}, S_{i,j+1})\beta_{i,j+1}(S_{i,j+1})}{\sum_{\substack{S_{i,j}\to S_{i,j+1}: \\ X_{i,j}=-1}} \alpha_{i,j}(S_{i,j})\gamma_{i,j}(-1, S_{i,j}, S_{i,j+1})\beta_{i,j+1}(S_{i,j+1})} \tag{5.1}$$

These LLRs are given as input to the LDPC decoder for the first column. Hard decision outputs from the LDPC decoder are given as decision feedback to n parallel equalizers. Assuming that we successfully decode the first column with LDPC decoder, we can calculate the LLRs of the second column using the fact that previous one bit is known for all the bits in second column. So for second column only those forward states of n equalizers are active for which previous bit is equal to the decoded result of first column. We proceed in the same way until the first L columns are decoded. Once the first L columns are decoded successfully, we can treat these L columns as known columns and switch back to our original BCJR-DFE receiver. Before proceeding with the original BCJR-DFE, we need to run the backward pass again with n equalizers corresponding to each row of the data-matrix with known starting states due to L decoded columns. Thus the complexity of this new decoding algorithm is slightly more than the complexity of two backward passes or equivalently one iteration of a turbo equalization. Note that we can use a single equalizer (instead of using n parallel equalizers) and process n rows serially. The equivalent channel seen by an LDPC decoder for columns 1 to L is a channel with memory. In practice if we use AWGN optimized codes for these columns then we need to set the code rate sufficiently low ($r_j < I(X_i; \boldsymbol{Y}|X_{i-1}, X_{i-2}, \cdots, X_{i-j+1})$) so that the LDPC decoder converges for the first L columns. Also in the case when channel is not known at the transmitter, we need to make sure that the code rates for the first L columns are such that $r_j < \min I(X_i; \boldsymbol{Y}|X_{i-1}, X_{i-2}, \cdots, X_{i-j+1}, H = H_k)$ over the class of channels

$H_k \in \{H_1, H_2, \cdots\}$.

B. Step BCJR-DFE

Data-matrix encoder (Fig. 9) uses a code of length n over one column. To reduce the latency of BCJR-DFE receiver we can reduce the size n of a column code, but this will lead to error propagation due to a shorter length code. In order to keep the length of a codeword same and reduce the size of a column, we can define a single codeword over multiple columns of the data-matrix. Let us define a single codeword over D consecutive columns of the data-matrix. Assuming that we use a code of length n, we now need to define a column of size n/D only. Thus for step BCJR-DFE, we define the encoder data-matrix of size $\frac{n}{D} \times m$, where again first L columns are zero bits or known bits and remaining $(m - L)$ columns contain $(\frac{m-L}{D})$ LDPC codewords. We will need to calculate the LLRs of D columns before decoding

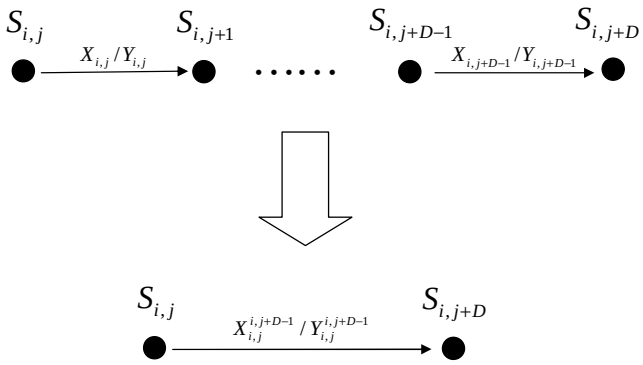

Fig. 13. Combining D consecutive transitions as one transition in step BCJR-DFE

the column code of length n. If we use n/D parallel equalizers corresponding to each row of data-matrix, then D consecutive columns corresponds to D time steps of these binary equalizer trellises. In order to calculate the LLRs on D consecutive columns, we can combine D time steps of a binary trellis as shown in Fig. 13, such

that sequence of D state transitions are now defined by a single state transition. Each new state transition is defined by D input bits ($\boldsymbol{X}_{i,j}^{i,j+D-1}$) and each new state has 2^D incoming and 2^D outgoing transitions. To keep the Markov structure of a trellis we need to have $1 \leq D \leq L$, where L is the channel memory. Length of this newly formed trellis is D times smaller than the original trellis length. Now we can have a BCJR-DFE receiver with this newly formed D-step BCJR trellis. Thus overall latency of step BCJR-DFE is $\frac{n}{D} \times m$ bits, which is D times lower than original BCJR-DFE receiver. Note that the proposition-1 does not hold true for step BCJR-DFE receiver. The equivalent channel along each column is not a memoryless channel, but it is a composite ISI channel where first n/D bits appear to come from a memoryless channel, the next n/D bits appear to come from an equivalent channel of memory 1, next n/D bits appear to come from an equivalent channel of memory 2, and so on. Optimum performance with step BCJR-DFE is obtained when column code is optimized for the composite ISI channel. Unequal error protection LDPC codes can be used in this case. Note that we achieve maximum reduction in latency when $D = L$, while for $L = 1$ (e.g. dicode channel, $H = [\frac{1}{\sqrt{2}} \ \frac{-1}{\sqrt{2}}]$) no latency reduction can be achieved with step BCJR-DFE receiver. In the following section we will compare the performance of multi-rate BCJR-DFE and step BCJR-DFE receivers with the performance of BCJR-DFE receiver and turbo equalization.

C. Simulation Results

1. Multi-rate BCJR-DFE Results

In this section we present simulation results for multi-rate BCJR-DFE where we used low rate codes instead of known bits at the beginning of the data-matrix. As explained in section A, we gradually increased the rates of column codes ($r_1 < r_2 < \cdots < r_m$)

Table V. Code rates for different columns of the data-matrix in multi-rate BCJR-DFE

Channel	m	r_1	r_2	r_3	r_4	r_5	r_6
$[\frac{1}{\sqrt{2}} \ -\frac{1}{\sqrt{2}}]$	4	0.40	0.50	0.52	0.58	-	-
[0.407 0.815 0.407]	5	0.37	0.42	0.54	0.55	0.62	-
[0.5 0.5 -0.5 -0.5]	6	0.38	0.40	0.42	0.55	0.61	0.64

in the data-matrix such that overall rate is equal to the desired rate r. Table V shows the number of columns used (m) along with their rates (r_m) for the dicode channel, the 3-tap channel and EPR4 channel. For all these channels, overall rate is $r = \frac{1}{m}\sum_{i=1}^{m} r_i = 0.5$. We used LDPC codes of length $n = 10000$ and rates shown in Table V, where codes were optimized for AWGN channel. Degree distributions of the AWGN optimized codes used are given in Tables VI, VII and VIII.

Hundred iterations were used in the LDPC decoder. Figs. 14, 15 and 16 show the performance results for the dicode channel, the 3-tap channel and EPR4 channel respectively. Latency of the multi-rate BCJR-DFE receiver is 4×10^4 bits for the dicode channel, 5×10^4 bits for the 3-tap channel, and 6×10^4 bits for the EPR4 channel. We also plot the performance of equal latency turbo equalization with 15 turbo iterations, where LDPC codes optimized for AWGN channel, of length 4×10^4, 5×10^4, and 6×10^4 were used for the dicode channel, the 3-tap channel, and EPR4 channel respectively. For dicode channel we get almost the same performance for BCJR-DFE, low latency BCJR-DFE, and turbo equalization with LDPC code of length 4×10^4. For the 3-tap channel, and EPR4 channel we get a gain of about 0.5 dB and 0.2 dB respectively over the same latency turbo equalization result. At the same time the complexity of multi-rate BCJR-DFE receiver is approximately 15 times smaller than the complexity of turbo equalization. It can be seen from the BER

Table VI. Degree distribution of AWGN optimized LDPC codes of different rates used with multi-rate BCJR-DFE for dicode ($[\frac{1}{\sqrt{2}} \ -\frac{1}{\sqrt{2}}]$) channel

$r = 0.40$		$r = 0.50$		$r = 0.52$		$r = 0.58$	
λ_2	0.29433	λ_2	0.21991	λ_2	0.21986	λ_2	0.21866
λ_3	0.25733	λ_3	0.23328	λ_3	0.21041	λ_3	0.22335
λ_{10}	0.44833	λ_4	0.02058	λ_6	0.03569	λ_6	0.06414
		λ_6	0.08543	λ_7	0.21664	λ_7	0.20383
		λ_7	0.06540	λ_{13}	0.06710	λ_{20}	0.29000
		λ_8	0.04767	λ_{23}	0.00053		
		λ_9	0.01912	λ_{25}	0.07060		
		λ_{19}	0.08064	λ_{27}	0.11625		
		λ_{20}	0.22798	λ_{28}	0.02592		
				λ_{29}	0.03557		
				λ_{30}	0.00138		
ρ_6	1.0	ρ_8	0.64854	ρ_9	1.0	ρ_{10}	1.0
		ρ_9	0.34747				
		ρ_{10}	0.00399				

Table VII. Degree distribution of AWGN optimized LDPC codes of different rates used with multi-rate BCJR-DFE for 3-tap ([0.407 0.815 0.407]) channel

$r = 0.37$		$r = 0.42$		$r = 0.54$		$r = 0.55$		$r = 0.62$	
λ_2	0.23537	λ_2	0.23205	λ_2	0.20178	λ_2	0.21847	λ_2	0.16958
λ_3	0.18115	λ_3	0.18993	λ_3	0.19793	λ_3	0.21244	λ_3	0.17925
λ_5	0.04342	λ_5	0.06984	λ_5	0.00217	λ_5	0.05359	λ_4	0.01631
λ_6	0.08761	λ_6	0.01177	λ_6	0.05556	λ_6	0.01688	λ_5	0.03113
λ_9	0.11938	λ_7	0.04945	λ_7	0.17697	λ_7	0.05418	λ_6	0.01018
λ_{13}	0.02741	λ_8	0.11193	λ_{15}	0.10989	λ_8	0.16438	λ_8	0.18442
λ_{14}	0.05460	λ_{13}	0.08794	λ_{17}	0.02734	λ_{12}	0.01519	λ_9	0.02090
λ_{41}	0.25102	λ_{15}	0.00103	λ_{19}	0.02189	λ_{18}	0.03673	λ_{13}	0.02968
		λ_{20}	0.02539	λ_{35}	0.02282	λ_{19}	0.01604	λ_{14}	0.03060
		λ_{24}	0.00720	λ_{38}	0.16934	λ_{21}	0.01477	λ_{15}	0.01259
		λ_{25}	0.01616	λ_{41}	0.01426	λ_{22}	0.00825	λ_{18}	0.00878
		λ_{26}	0.02701			λ_{23}	0.12138	λ_{41}	0.30651
		λ_{38}	0.17021			λ_{31}	0.06762		
ρ_7	1.0	ρ_7	0.5	ρ_{10}	1.0	ρ_9	0.5	ρ_{13}	0.5
		ρ_8	0.5			ρ_{10}	0.5	ρ_{14}	0.5

Table VIII. Degree distribution of AWGN optimized LDPC codes of different rates used with multi-rate BCJR-DFE for EPR4 ($[0.5\ 0.5\ -0.5\ -0.5]$) channel

$r = 0.38$		$r = 0.40$		$r = 0.42$		$r = 0.55$		$r = 0.61$		$r = 0.64$	
λ_2	0.23838	λ_2	0.29433	λ_2	0.23205	λ_2	0.21847	λ_2	0.18063	λ_2	0.17055
λ_3	0.18034	λ_3	0.25733	λ_3	0.18993	λ_3	0.21244	λ_3	0.18829	λ_3	0.19023
λ_5	0.10161	λ_{10}	0.44833	λ_5	0.06984	λ_5	0.05359	λ_5	0.07365	λ_6	0.07154
λ_7	0.02951			λ_6	0.01177	λ_6	0.01688	λ_6	0.01861	λ_7	0.13861
λ_9	0.04665			λ_7	0.04945	λ_7	0.05418	λ_9	0.21726	λ_{14}	0.15833
λ_{10}	0.12603			λ_8	0.11193	λ_8	0.16438	λ_{20}	0.07889	λ_{16}	0.00158
λ_{13}	0.01168			λ_{13}	0.08794	λ_{12}	0.01519	λ_{23}	0.00094	λ_{17}	0.00067
λ_{16}	0.00221			λ_{15}	0.00103	λ_{18}	0.03673	λ_{27}	0.03217	λ_{41}	0.26845
λ_{19}	0.00831			λ_{20}	0.02539	λ_{19}	0.01604	λ_{29}	0.01072		
λ_{21}	0.04645			λ_{24}	0.00720	λ_{21}	0.01477	λ_{41}	0.19879		
λ_{41}	0.20877			λ_{25}	0.01616	λ_{22}	0.00825				
				λ_{26}	0.02701	λ_{23}	0.12138				
				λ_{38}	0.17021	λ_{31}	0.06762				
ρ_7	1.0	ρ_6	1.0	ρ_7	0.5	ρ_9	0.5	ρ_{12}	0.5	ρ_{14}	1.0
				ρ_8	0.5	ρ_{10}	0.5	ρ_{13}	0.5		

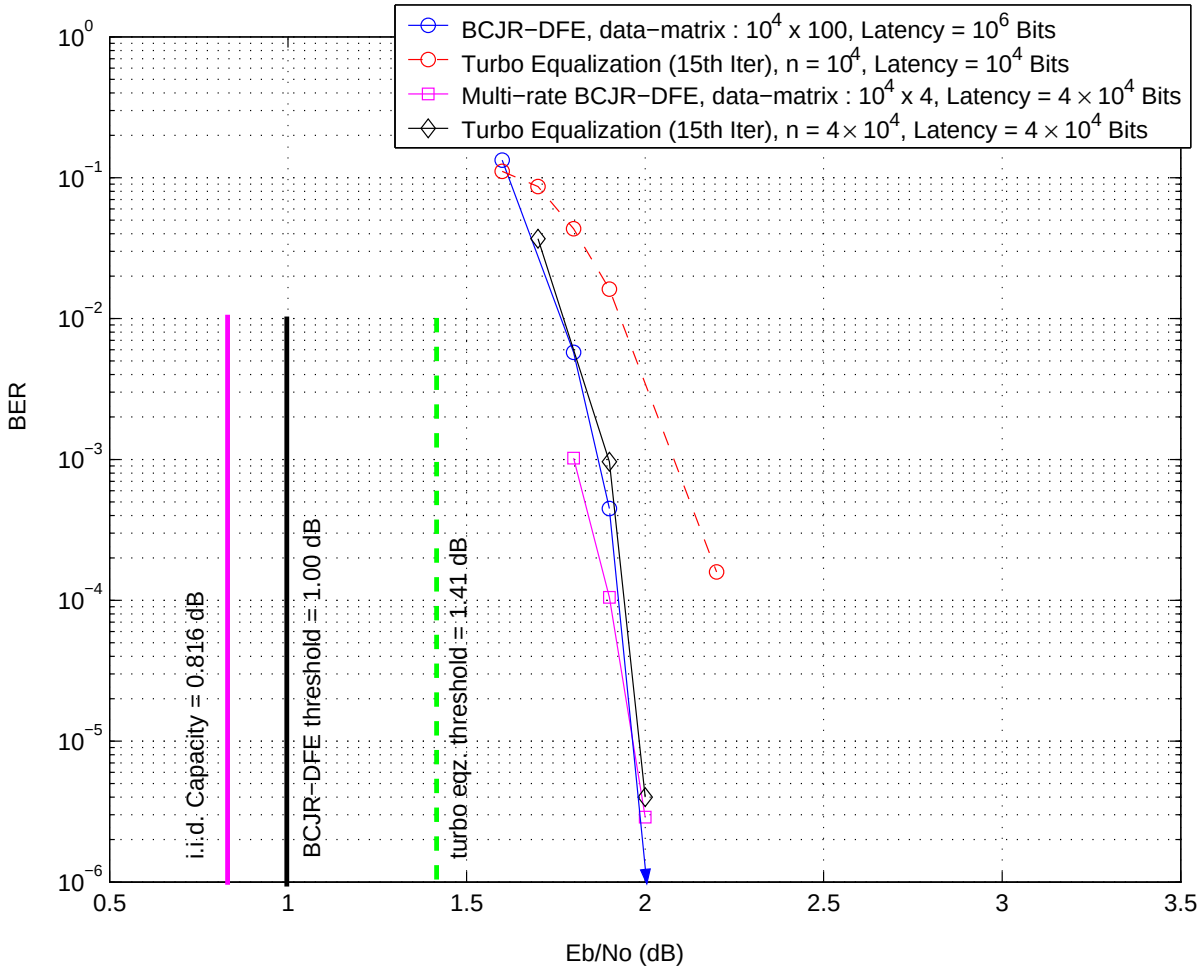

Fig. 14. Simulation results for the dicode channel $H = [\frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}}]$ with multi-rate BCJR-DFE

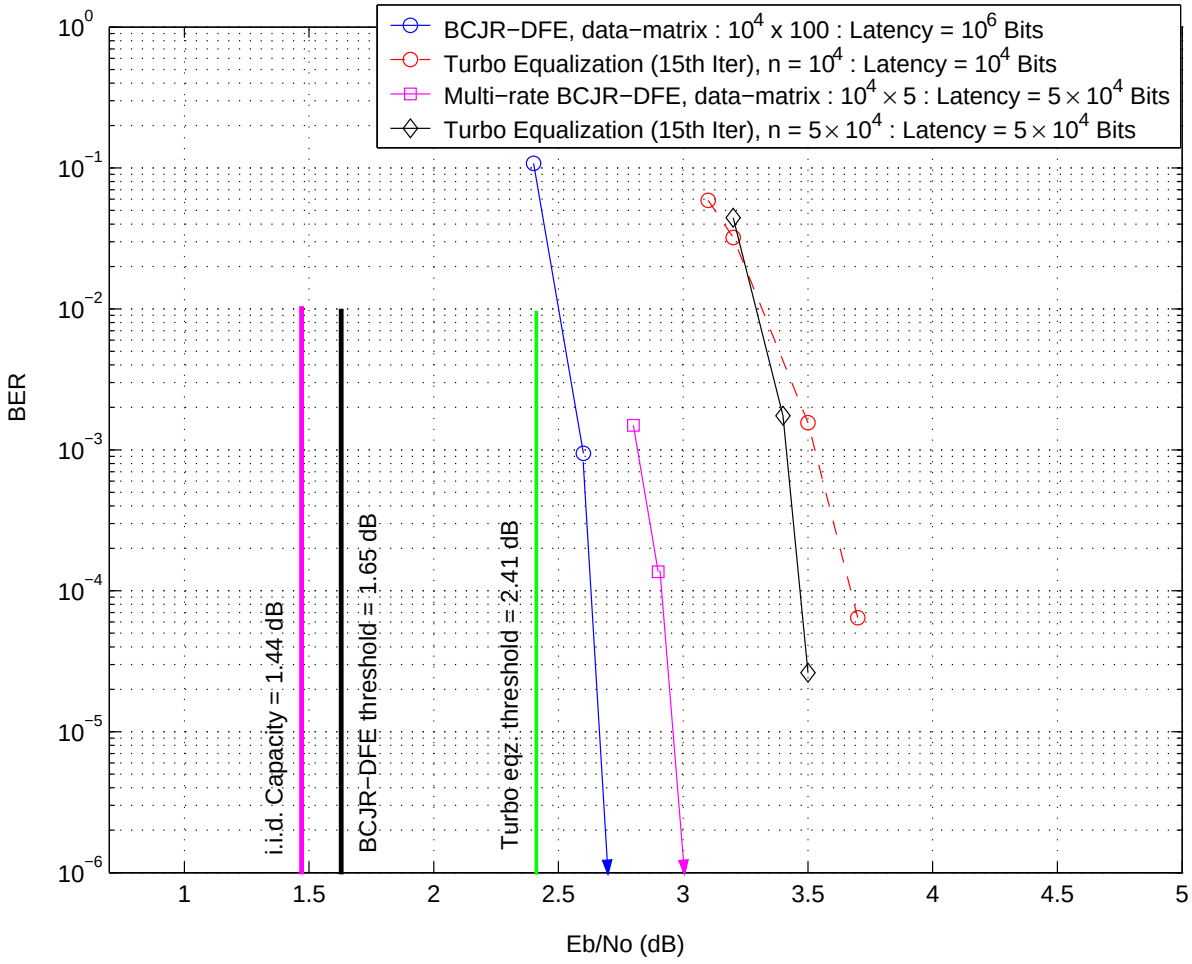

Fig. 15. Simulation results for the three tap channel $H = [0.407\ 0.815\ 0.407]$ with multi-rate BCJR-DFE

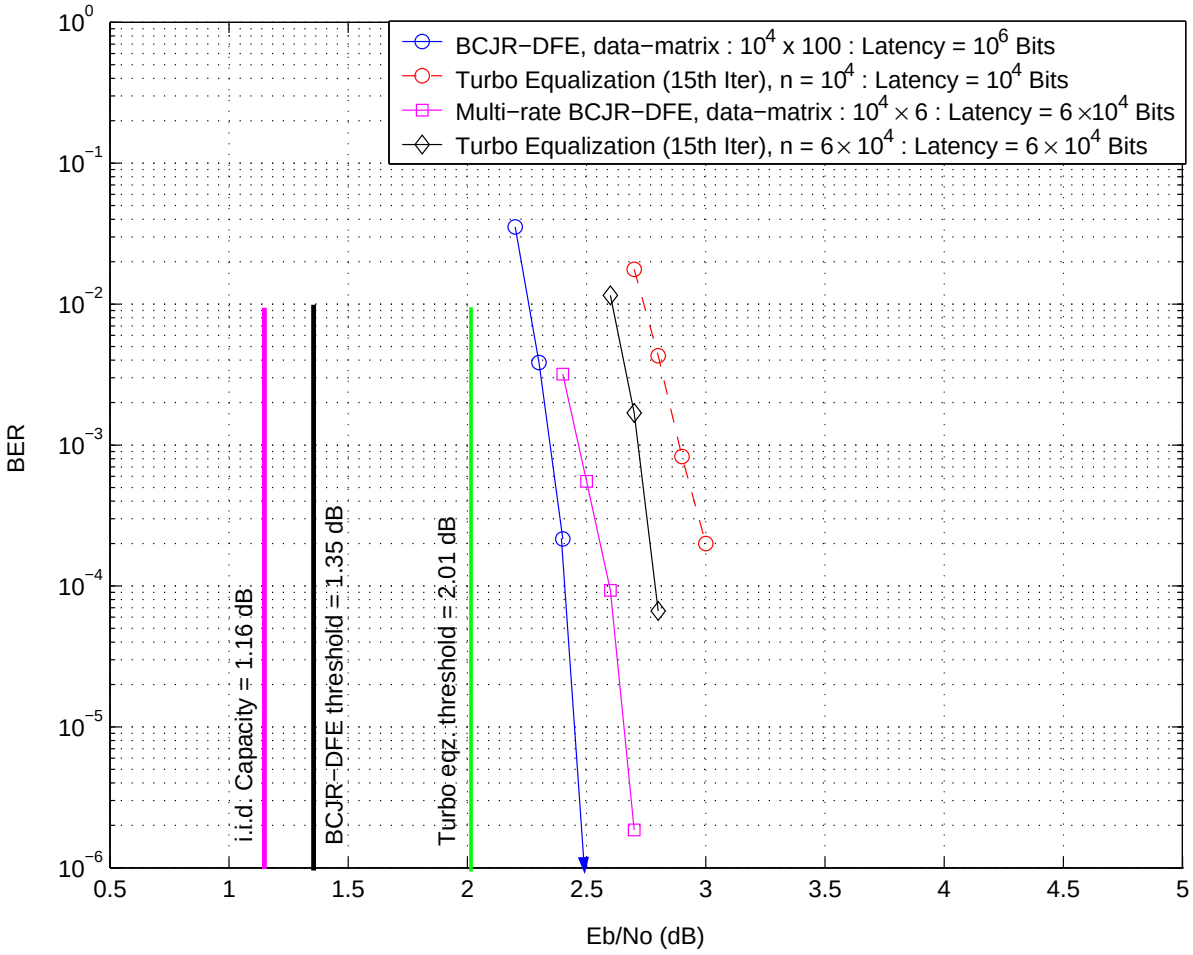

Fig. 16. Simulation results for the EPR4 channel $H = (0.5\ 0.5\ -0.5\ -0.5)$ with multi-rate BCJR-DFE

curves that there is very little loss in performance if we use low rate codes instead of known bits.

2. Step BCJR-DFE Results

In this section we present simulation results for step BCJR-DFE with three tap channel and EPR4 channel. For 2-step BCJR-DFE ($D = 2$), we used an LDPC code of length $n = 10000$ while for 3-step BCJR-DFE ($D = 3$) we used an LDPC code of length $n = 9999$. Both the codes were optimized for AWGN channel and have same degree distribution as given in chapter IV, Table III. Hundred iterations were used in the LDPC decoder. Data-matrix size was set to 5000×100 for 2-step BCJR-DFE and 3333×99 for 3-step BCJR-DFE receiver. As shown in Fig. 17 for the 3-tap channel, the performance with 2-step BCJR-DFE receiver is nearly the same as turbo equalization performance and there is about 0.5 dB loss as compared to BCJR-DFE receiver. For EPR4 channel (Fig. 18), the performance with 2-step BCJR-DFE receiver is nearly the same as the performance of BCJR-DFE receiver, while performance with 3-step BCJR-DFE receiver is nearly the same as turbo equalization performance. Better performance can be obtained with step BCJR-DFE by optimizing the LDPC code for respective composite ISI channel.

D. Turbo Equalization with Data-matrix Encoding

As we decrease the length of a column code, decoder for the column code fails to converge to the correct codeword at low SNR. This causes severe error propagation in the BCJR-DFE receiver. We can overcome this error propagation by performing a few turbo iterations between column codes. Figure 19 shows the frame error rate

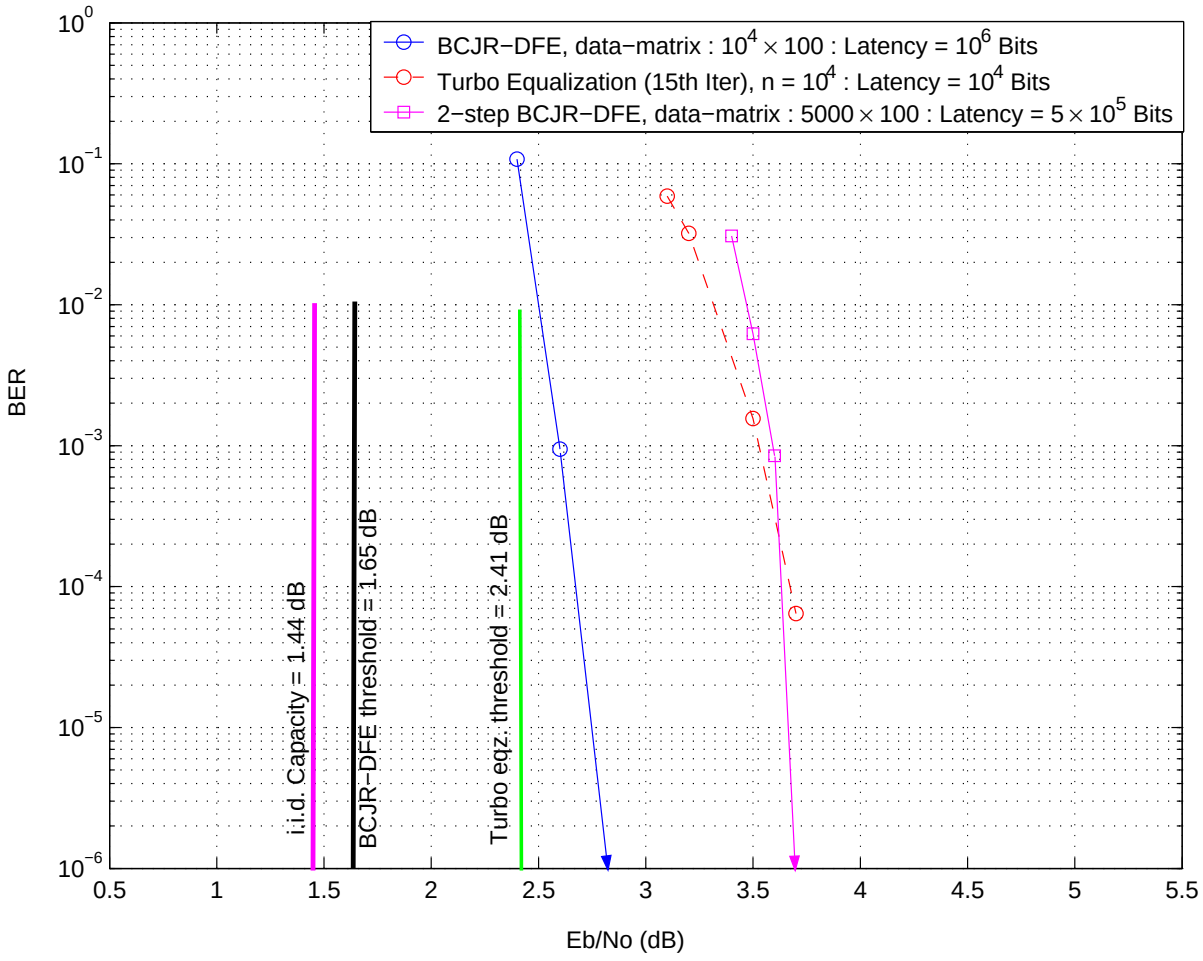

Fig. 17. Simulation results for the 3-tap channel $H = [0.407\ 0.815\ 0.407]$ with step BCJR-DFE

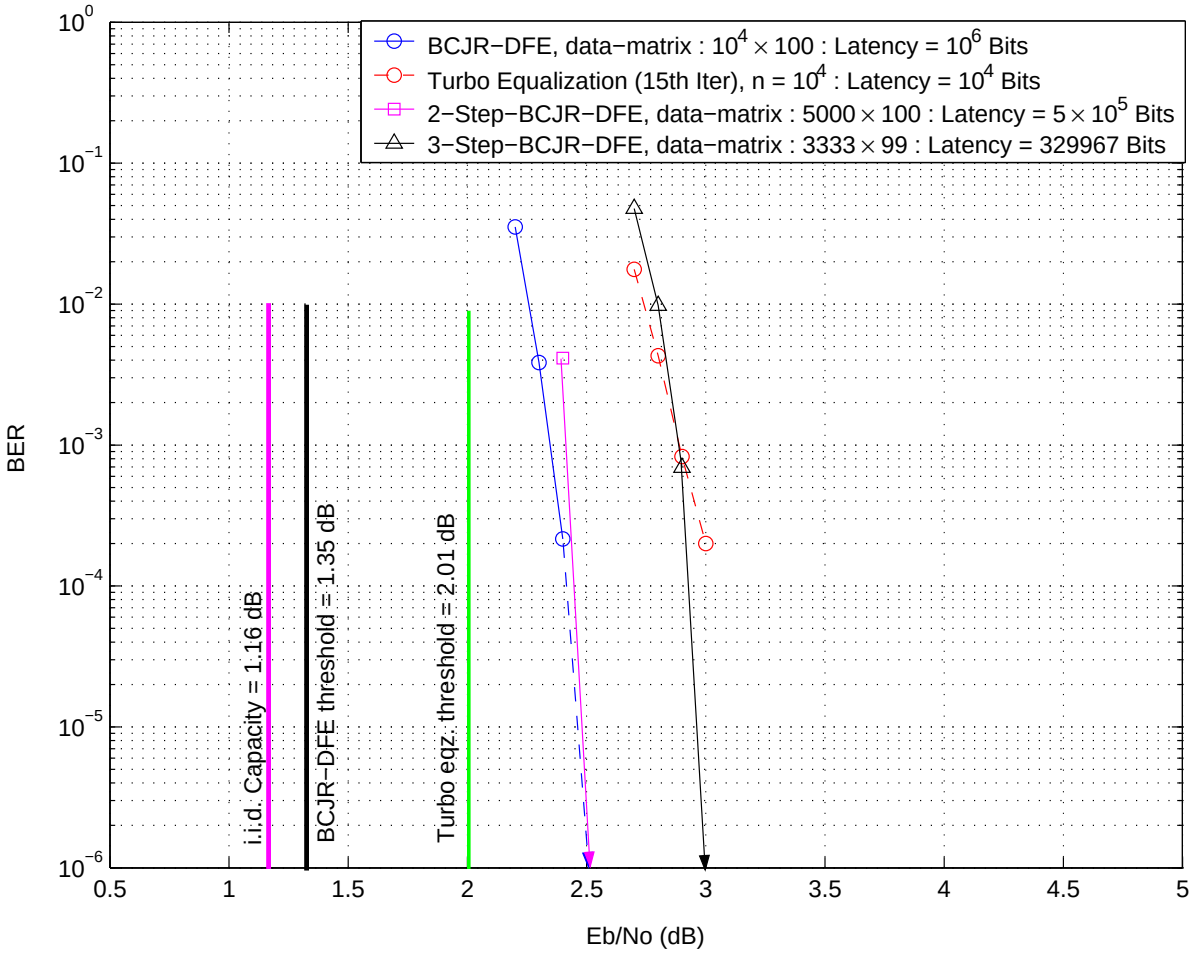

Fig. 18. Simulation results for the EPR4 channel $H = (0.5\ 0.5\ -0.5\ -0.5)$ with step BCJR-DFE

(assuming one frame equal to one LDPC column codeword) with different receiver structures for 3-tap ISI channel. Data is encoded with LDPC code of length 500 bits and arranged in the data-matrix structure as shown in Fig. 9. Number of columns m is set to 20. Data is transmitted along the rows from first row to the last row. As

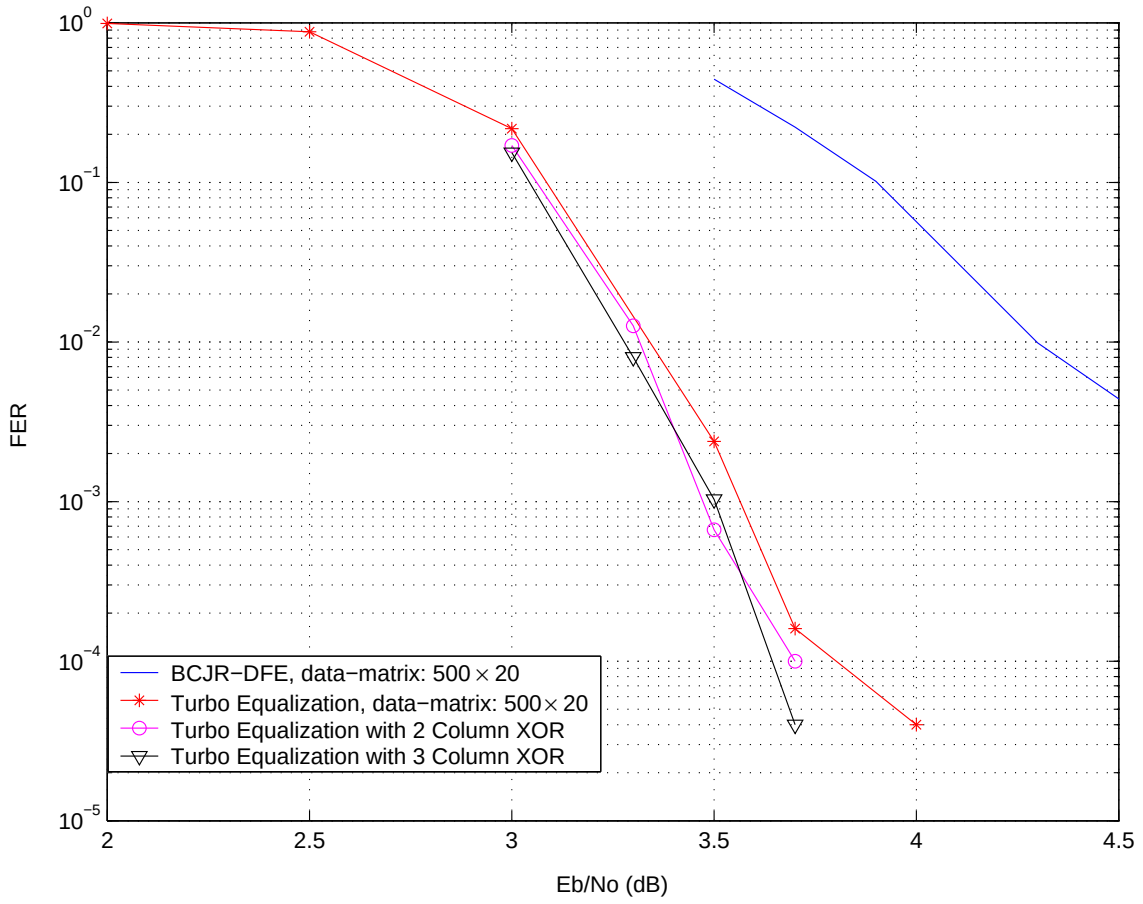

Fig. 19. Simulation results for the 3-tap channel $H = [0.407\ 0.815\ 0.407]$ for turbo equalization with data-matrix encoding

shown in Fig. 19, BCJR-DFE suffers from error propagation. We can also perform turbo equalization considering the whole data-matrix as one data-block. Performance with 20 turbo iterations is shown in Fig. 19, which shows significant frame error rate

improvement over BCJR-DFE albeit with higher complexity. It was observed that this turbo equalization has a bootstrap effect in which many column codewords at the start and at the end of the data-matrix quickly converge due to columns of known bits. Codeword columns in the data-matrix are correlated with each other through channel memory. The correlation between columns can be increased if we do bitwise XOR operation on adjacent columns. For example for '2 column XOR', we obtain a new data-matrix from original encoded data-matrix, where ith column of new data-matrix is bitwise XOR of $(i-1)$th and ith column of original data-matrix (first column is copied as it is). Similarly '3 column XOR' denotes the operation where ith column of new data-matrix is bitwise XOR of $(i-2)$th, $(i-1)$th, and ith column of original data-matrix (first column is copied as it is and 2nd column is obtained by '2 column XOR' operation). The frame error rate of turbo equalization in with '2 column XOR' and '3 column XOR' operation with 20 turbo iterations is also shown in Fig. 19. We can see that a little gain can be obtained with 'column XOR' operation. Figure 20 compares the frame error rate of different receivers for the 3-tap channel, where one frame consists of 10^4 bits. The frame error rate of standard turbo equalization with LDPC code of length $n = 10^4$ is shown in Fig. 20. We also plot the frame error rate for turbo equalization with data-matrix (of size 500×20) encoding, with '2 column XOR' operation and '3 column XOR' operation where one frame is assumed to be equal to the whole data-matrix of size $500 \times 20 = 10^4$ bits. We can see that '2 column XOR' and '3 column XOR' operation gives comparable results with conventional turbo equalization. Here we need LDPC decoder of size $n = 10^4$ for conventional turbo equalization, while for other receivers we need LDPC decoder of size $n = 500$ only. Thus significant hardware complexity reduction can be obtained for turbo equalization with data-matrix encoding.

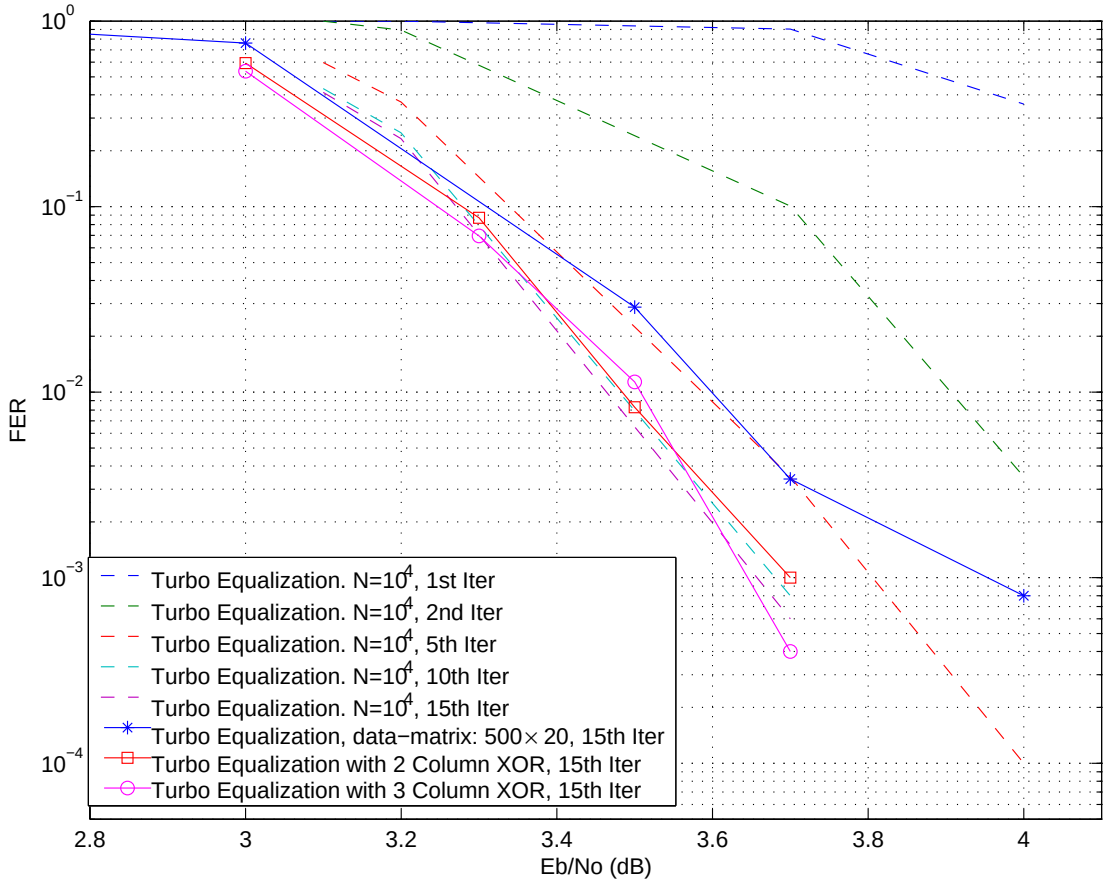

Fig. 20. Simulation results for the 3-tap channel $H = [0.407\ 0.815\ 0.407]$ for turbo equalization with $n = 10^4$ and turbo equalization with data-matrix encoding

E. Replacing BCJR with MMSE

In this section we consider the new receiver structure called as MMSE-DFE, which is similar to BCJR-DFE, except that non-linear BCJR equalizer is replaced with linear MMSE equalizer. Certainly for constrained input constellation, MMSE-DFE receiver is suboptimum and we want to find the loss in achievable information rates with MMSE-DFE as compared to a BCJR-DFE receiver for constrained input constellation. Consider a causal ISI channel of memory L with input X_k and output Y_k such

that

$$Y_k = \sum_{i=0}^{L} h_i X_{k-i} + N_k \tag{5.2}$$

where h_i's are the channel coefficients and N_k is additive white Gaussian noise. We can rewrite (5.2) in the matrix form as

$$\underbrace{\begin{bmatrix} Y_k \\ \vdots \\ Y_{k+L} \end{bmatrix}}_{\mathbf{Y}_k} = \underbrace{\begin{bmatrix} h_L & h_{L-1} & \cdots & h_0 & 0 & \cdots & 0 & 0 \\ 0 & h_L & h_{L-1} & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_L & h_{L-1} & \cdots & h_1 & h_0 \end{bmatrix}}_{\mathbf{H}_k} \underbrace{\begin{bmatrix} X_{k-L} \\ \vdots \\ X_k \\ \vdots \\ X_{k+L} \end{bmatrix}}_{\mathbf{X}_k} + \underbrace{\begin{bmatrix} N_k \\ \vdots \\ N_{k+L} \end{bmatrix}}_{\mathbf{N}_k} \tag{5.3}$$

$$\mathbf{Y}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{N}_k \qquad \text{with} \qquad \mathbf{N}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \tag{5.4}$$

Any a-priori information $\Lambda^{(a)}(x_k) = \Lambda_k^{(a)}$ on X_k can be converted into the soft estimate $\widetilde{X}_k$ using

$$\widetilde{X}_k \triangleq \tanh\left(\frac{\Lambda_k^{(a)}}{2}\right).$$

We define vector $\widetilde{\mathbf{X}}_k$ as

$$\widetilde{\mathbf{X}}_k \triangleq [\widetilde{X}_{k-L}, \widetilde{X}_{k-L+1} \cdots, 0, \cdots, \widetilde{X}_{k+L-1}, \widetilde{X}_{k+L}]^T \tag{5.5}$$

Using (5.5) we can perform soft interference cancellation on $\mathbf{Y}_k$ to obtain

$$\widetilde{\mathbf{Y}}_k \triangleq \mathbf{Y}_k - \mathbf{H}_k \widetilde{\mathbf{X}}_k = \mathbf{H}_k(\mathbf{X}_k - \widetilde{\mathbf{X}}_k) + \mathbf{N}_k$$

Instantaneous linear MMSE filtering can be applied to $\widetilde{\mathbf{Y}}_k$ to obtain

$$Z_k = \mathbf{W}_k^H \widetilde{\mathbf{Y}}_k, \tag{5.6}$$

where filter $\mathbf{W}_k$ minimizes the mean square error between X_k and Z_k. It can be shown that

$$\mathbf{W}_k = (\mathbf{H}_k \mathbf{\Delta}_k \mathbf{H}_k^H + \mathbf{\Sigma})^{-1} \mathbf{H}_k \mathbf{e} \tag{5.7}$$

$$\text{where} \quad \mathbf{\Delta}_k \triangleq \text{diag}\{1 - \widetilde{X}_{k-L}^2, \cdots, 1, \cdots, 1 - \widetilde{X}_{k+L}^2\} \tag{5.8}$$

and $\mathbf{e}$ is a $[2L+1]$ vector with all zero entries, except for the $(L+1)$th entry which is 1. In order to form the LLR of $X_k, (\Lambda(x_k) = \Lambda_k)$ we can assume that $p(Z_k|X_k)$ is Gaussian distributed with mean $\mu_k X_k$ and variance σ_k^2. It can be shown that for real valued channels [55],

$$\Lambda_k = \frac{2Z_k}{1 - \mu_k} \tag{5.9}$$

$$\text{where} \quad \mu_k \triangleq E\{Z_k X_k\} = \mathbf{e}^T (\mathbf{H}_k \mathbf{\Delta}_k \mathbf{H}_k^H + \mathbf{\Sigma})^{-1} \mathbf{H}_k \mathbf{e}. \tag{5.10}$$

For MMSE-DFE receiver, we use the same data-matrix encoder structure with known bits and same column code decoding sequence as used for BCJR-DFE receiver (see Fig. 9). In order to calculate the achievable information rates, we can assume that column LDPC code always converges to the correct codeword and thus for every bit X_k we know past L bits perfectly. With this vector $\widetilde{\mathbf{X}}_k$ becomes

$$\widetilde{\mathbf{X}}_k = [X_{k-L}, X_{k-L+1} \cdots, X_{k-1}, 0, \cdots, 0]^T.$$

So we can write $\mathbf{\Delta}_k$ in (5.8) as

$$\mathbf{\Delta}_k = \text{diag}\{0, \cdots, 0, 1, \cdots, 1\} \tag{5.11}$$

i.e. first L diagonal entries are equal to zero (perfect knowledge of past L bits) and remaining $L+1$ entries are equal to one (no knowledge of future bits). Using (5.11) to calculate the LLR of X_k (5.9), we can find out the achievable information rate with

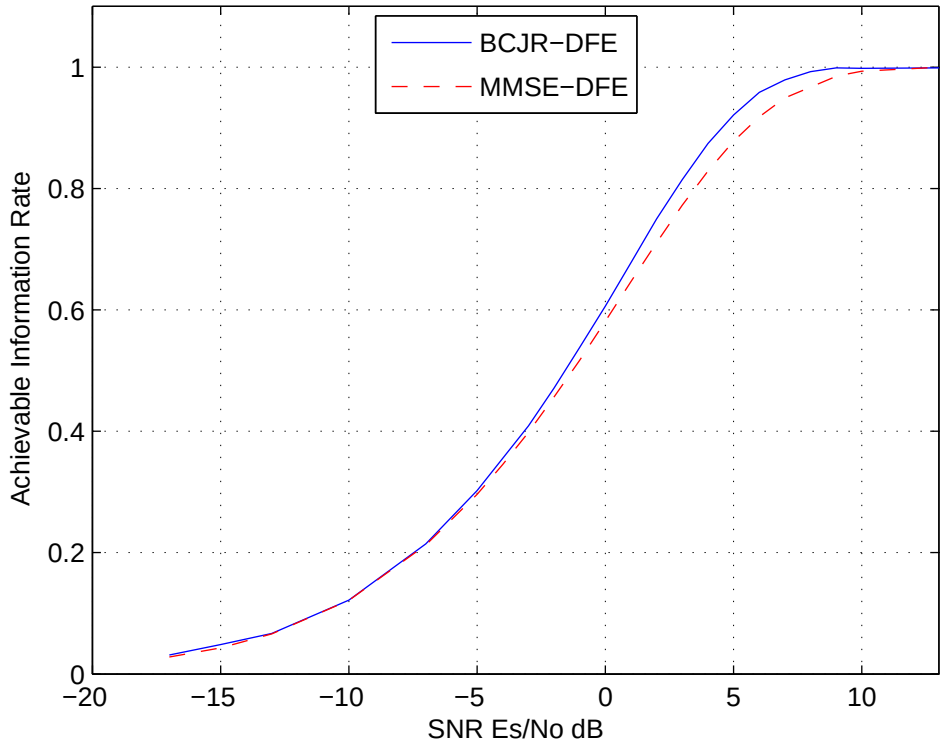

Fig. 21. Comparison of achievable information rates with MMSE-DFE and BCJR-DFE scheme for 3-tap channel

MMSE-DFE as

$$I_{MMSE-DFE} = \lim_{k\to\infty} \frac{1}{k} I(X_1, \cdots, X_k; \Lambda_1, \cdots, \Lambda_k)$$

For BCJR-DFE receiver, achievable information rate is calculated by using BCJR-DFE algorithm to calculate the LLR of bit X_k, where perfect knowledge of previous state in BCJR trellis is assumed. Due to optimality of BCJR-DFE receiver, this information rate is equal to the $C_{i.i.d.}$ of the channel. Figures 21, 22 and 23 show the plot of achievable information rate with BCJR-DFE and MMSE-DFE receiver as a function of signal-to-noise ratio (Es/No) for 3-tap channel, EPR4 channel and

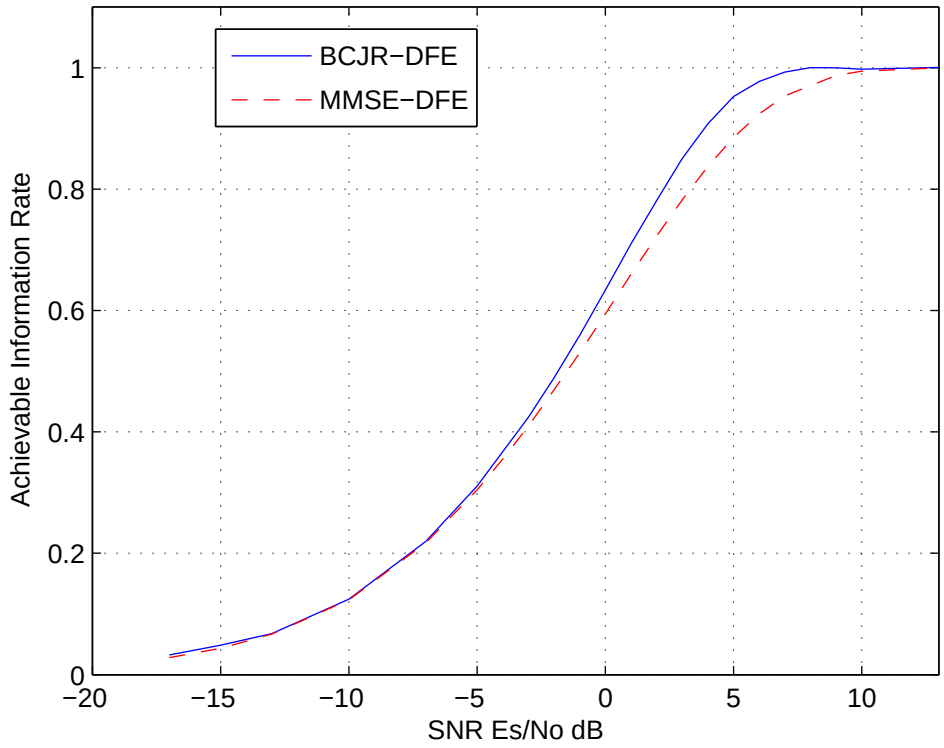

Fig. 22. Comparison of achievable information rates with MMSE-DFE and BCJR-DFE scheme for EPR4 channel

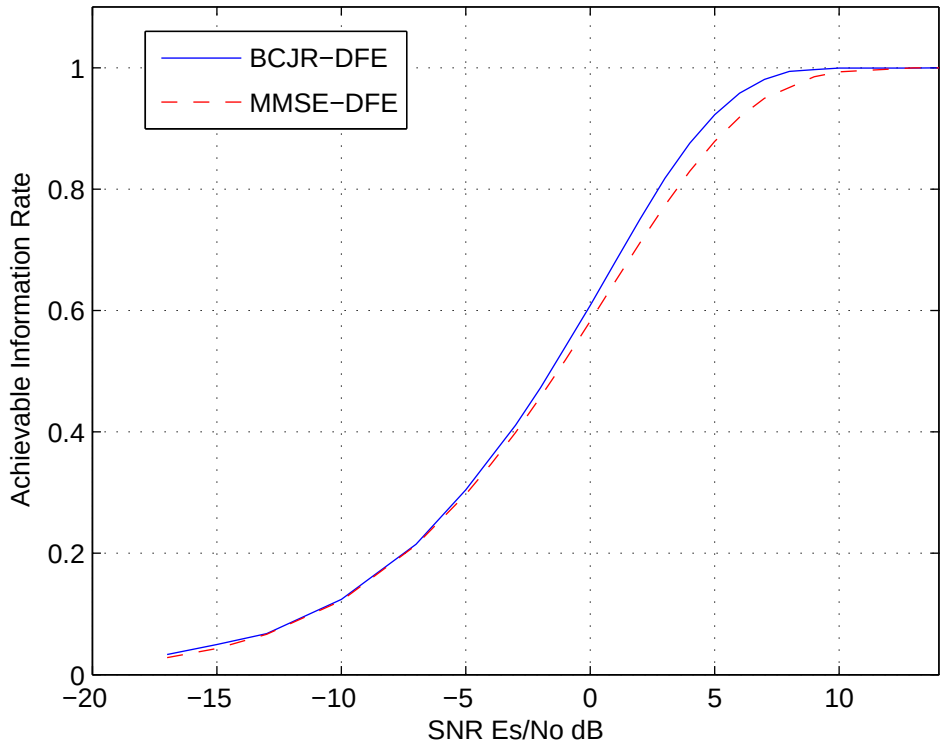

Fig. 23. Comparison of achievable information rates with MMSE-DFE and BCJR-DFE scheme for PR2 channel

Table IX. Loss in SNR with MMSE-DFE over BCJR-DFE at different rates

Channel	Loss in SNR (Es/No) in dB		
	$r = 0.5$	$r = 0.9$	$r = 0.95$
[0.407 0.815 0.407]	0.285	1.00	1.254
[0.5 0.5 -0.5 -0.5]	0.35	1.53	1.932
$[\frac{1}{\sqrt{6}}\ \frac{2}{\sqrt{6}}\ \frac{1}{\sqrt{6}}]$	0.295	1.015	1.24

PR2 channel. These plots clearly show that at lower rate, the loss in SNR with MMSE-DFE over BCJR-DFE is negligible. As the rate increases this loss increases. Table IX summarizes this loss in SNR for rate $r = 0.5$, $r = 0.9$ and $r = 0.95$. Thus lower complexity can be achieved with MMSE-DFE for little loss in performance.

F. Conclusion

Large latency of BCJR-DFE limits its practical implementation. We proposed two solutions for reducing latency of BCJR-DFE receiver. With multi-rate BCJR-DFE receiver, we have shown that by properly selecting the rates for each column in the data-matrix, latency of BCJR-DFE can be reduced significantly without loosing much of the performance. To reduce the complexity of the receiver further, we showed that nonlinear BCJR equalizer can be replaced with linear MMSE equalizer. We quantified the loss in information rate with MMSE-DFE receiver for different channels. It is usually believed that linear MMSE or even DFE (decision feedback equalizer) performs poorly on channels with spectral nulls. Here we show that for wide range of SNRs this loss is negligible even for channels with spectral nulls.

CHAPTER VI

JOINT SYNCHRONIZATION, EQUALIZATION AND DECODING USING PER-SURVIVOR BCJR-DFE RECEIVER[†]

Along with ISI, many communication channels also suffer from timing errors. Traditionally, timing errors are dealt with dedicated timing recovery circuitry. In this chapter, we extend the BCJR-DFE receiver introduced in chapter IV to accomplish timing recovery, ISI equalization and decoding concurrently using per-survivor processing technique. Similar to the case of equalization using BCJR-DFE receiver, we show that the proposed per-survivor BCJR-DFE receiver outperforms other receiver structures, such as iterative timing recovery and detection.

A. Introduction

In a typical communication system, the receiver-sampler is uncertain about the correct arrival time of an analog pulse due to the presence of timing offset between the transmitter and the receiver clock. Sampling of the received signal at an incorrect timing instant results in degradation of the sampled signal quality and, hence, bit error rate performance. These timing offsets are estimated through a process known as timing recovery and used for synchronizing a sampler with the received analog signal.

Conventional timing recovery does not exploit the error correction code (ECC) employed in the channel. It has been shown that by utilizing the ECC code during timing recovery, potential performance gain can be achieved, in particular at very

[†]©2005 IEEE. Reprinted, with permission, from "Joint timing recovery, ISI equalization and decoding using per-survivor BCJR-DFE", N. Nangare, K.R. Narayanan, X. Yang, and E. Kurtas, *Proc. IEEE Globecom.*, Nov./Dec. 2005, pp. 1620-1624.

low SNR regions. For example, iterative timing recovery with turbo equalization was used by Nayak, Barry and McLaughlin [71], where the timing offset estimates were refined during each iteration between equalizer and decoder. Iterative timing recovery based on per-survivor processing [72] with BCJR [8] algorithm was proposed by Kovintavewat, Barry, Erden and Kurtas [73], where timing recovery was embedded inside the equalization trellis. In order to obtain good bit error rate performance, iterative timing recovery techniques often require many iterations which prohibits its use in practical applications. In this chapter, we explore a low complexity and non-iterative joint timing recovery, ISI equalization and decoding using per-survivor processing BCJR-DFE (PSP BCJR-DFE) algorithm. We show that using per-survivor processing for timing recovery along with the BCJR-DFE algorithm can be a low complexity solution for timing recovery over a coded system with ISI. As shown in chapter IV,

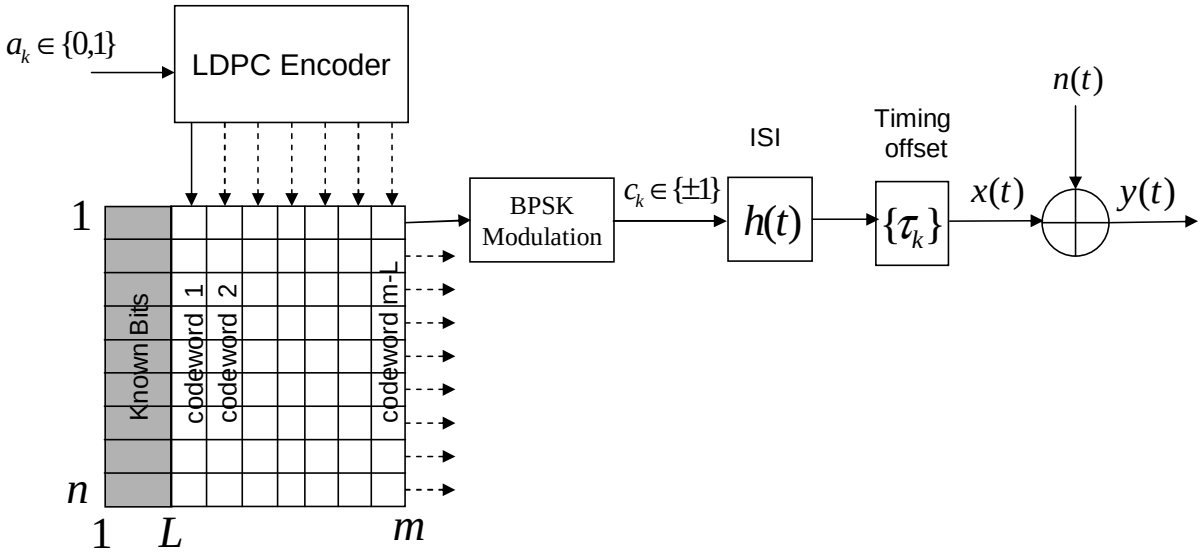

Fig. 24. Transmitter structure incorporating data-matrix encoder for the case of data transmission over a channel with unknown timing offsets

code optimization specific to ISI channel is not required and an AWGN optimized code performs equally well on various ISI channels.

B. System Model

Figures 24 and 25 show the transmitter and receiver structure for the system under consideration. Let L denote the memory of the ISI channel and n be the length of the

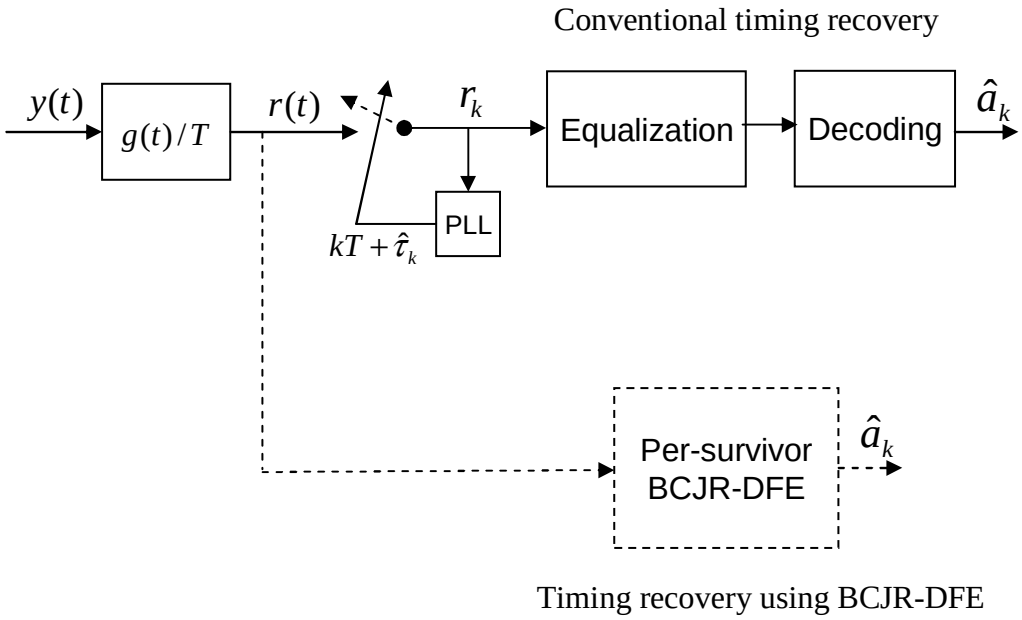

Fig. 25. Receiver structure showing timing recovery using conventional PLL based receiver and using per-survivor BCJR-DFE receiver

LDPC code used. As shown in Fig. 24, information bits $a_k \in \{0, 1\}$ are encoded using an LDPC encoder and arranged in the data-matrix structure similar to Fig. 9. The first L columns in the data-matrix are sequences of known bits. LDPC codewords from 'codeword 1' to 'codeword $(m - L)$' are arranged along the columns in the rest of the data-matrix. Encoded data is transmitted row-wise over an ISI channel with random timing errors. ISI channel is denoted by $h(t)$ where, $h(t) = \sum_k h_k g(t - kT)$

and $g(t) = sin(\pi t/T)/(\pi t/T)$ is an ideal zero-excess-bandwidth Nyquist pulse. We denote channel impulse response sampled at T-clock as $H(D) = \sum_{l=0}^{L} h_l D^l$. Timing offset of transmitter clock at time kT is modelled by delay τ_k. With this timing offset, the received signal $y(t)$ can be represented as

$$y(t) = \sum_k c_k h(t - kT - \tau_k) + n(t), \tag{6.1}$$

where c_k's are BPSK modulated symbols and $n(t)$ is AWGN process with zero mean and $E[n(t)n(t)^*] = \sigma_n^2 \delta(t)$. We assume a random walk model [74] for timing offset τ_k such that

$$\tau_{k+1} = \tau_k + \mathcal{N}(0, \sigma_w^2). \tag{6.2}$$

Here σ_w determines the severity of the timing jitter. A variety of timing jitter channels can be represented with this timing offset model by simply changing σ_w. Receiver structure with conventional timing recovery is shown in Fig. 25. A phase locked loop (PLL) is used to keep track of timing offsets. The received signal $y(t)$ is filtered with the receiver filter $g(t)/T$ and then sampled with a PLL to get a sampled sequence $\{r_k\}$. PLL uses a timing error detector (TED) to estimate the residual timing error $\epsilon_k = \tau_k - \hat{\tau}_k$, where $\hat{\tau}_k$ denote the timing offset estimate at time kT. Commonly used Mueller and Müller TED [75] generates timing error estimates $\hat{\epsilon}_k$ according to

$$\hat{\epsilon}_k = r_k \hat{d}_{k-1} - r_{k-1} \hat{d}_k, \tag{6.3}$$

where $\hat{d}_k$ is an estimate of the ISI channel output $d_k = \sum_{l=0}^{L} c_l h_{k-l}$, which can be obtained by three-level quantization of signal r_k. Performance improvement can be obtained by using the soft estimate $\tilde{d}_k$ in place of $\hat{d}_k$ in (6.3) according to [71]

$$\tilde{d}_k = E[d_k | r_k] = \frac{2 \sinh(2r_k/\sigma_n^2)}{\cosh(2r_k/\sigma_n^2) + e^{2/\sigma_n^2}} \tag{6.4}$$

PLL then updates the timing offset estimates by using a first order model given by

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \mu\hat{\epsilon}_k, \tag{6.5}$$

where μ denotes the PLL gain. In a conventional timing recovery, equalization and decoding is performed in succession on the PLL output to obtain the estimate $\{\hat{a}_k\}$ of the information sequence $\{a_k\}$. Using per-survivor timing recovery [73] with BCJR algorithm, equalization and timing offset estimation can be performed together. Turbo decoding [1] can be performed between decoder and per-survivor-processing BCJR (PSP BCJR) to obtain the better estimate of the information sequence $\{a_k\}$. Computational complexity of such iterative processing is very high, especially when a large number of iterations are required to achieve better performance. Alternatively we can use a non-iterative PSP BCJR-DFE receiver which has significantly less computational complexity than iterative processing. In the next section we describe the PSP BCJR-DFE receiver (Fig. 25) which performs joint timing recovery, ISI equalization and decoding.

C. Per-survivor Processing BCJR-DFE Receiver

Operation of the per-survivor processing BCJR-DFE is similar to the operation of BCJR-DFE with the addition of per-survivor processing for timing offset estimation and performing the forward pass first and then the backward pass. Input to the PSP BCJR-DFE is the low pass filtered signal $r(t)$. Signal $r(t)$ is sampled at baud-rate and arranged in the data-matrix similar to the transmitted data-matrix. Sampled signal has sufficient statistics to represent the continuous and bandlimited signal $r(t)$. Hence, given a sampled sequence $\{r_k\}$ associated with timing offset sequence $\{\tau_k\}$, we can obtain a new set of sample sequence $\{r_k^{no-offset}\}$ with zero timing offsets by

using interpolation of $\{r_k\}$ according to

$$r_k^{no-offset} = \sum_{l=l_{\min}}^{l_{\max}} r_l g(kT - lT - \tau_l). \tag{6.6}$$

1. Forward Pass

In order to describe the algorithm, we will assume an ISI channel with memory $L = 2$. Let $s_k = [c_{k-1}, c_{k-2}]$ denote the channel state in a state trellis diagram at time k. We label the four different states of the trellis as $s_k \in \{0, 1, 2, 3\}, \ \forall \ k$. At first, the PSP BCJR-DFE performs a single LARGE forward pass over the trellis with all the received data bits and calculates the forward log probability $\alpha_k^f(s_k)$ [8] for each state. This is similar to the forward pass of PSP BCJR algorithm explained in [73]. We assume perfect acquisition at the start of the data-matrix block by setting $\hat{\tau}_0^f(s_0) = 0$. Figure 26 shows a section of the forward trellis. Each state in the forward trellis

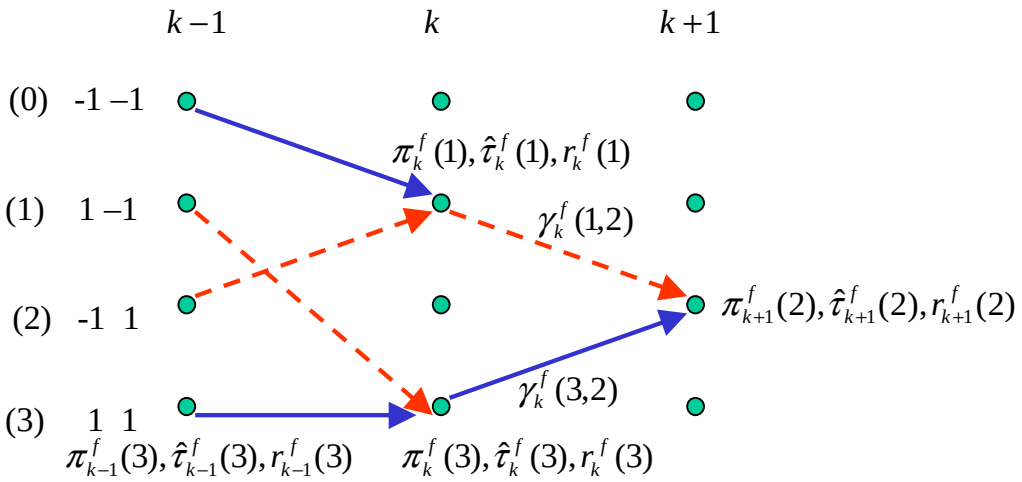

Fig. 26. Forward pass of BCJR-DFE receiver employing per-survivor timing recovery

maintains its own survivor path. Survivor paths are shown by solid lines in Fig. 26. Four different PLLs operate on these survivor paths to produce the forward timing

offset estimate $\hat{\tau}_k^f(s_k)$ at time k. The forward timing offset estimate $\hat{\tau}_k^f(s_k)$ at state s_k is used to calculate the forward sample corresponding to the branches originating from state s_k in the forward direction as, $r_k^f(s_k) = r(kT + \hat{\tau}_k^f(s_k))$. At each time instant, k, each state s_k in the forward trellis maintains three quantities: 1) a survivor state $\pi_k^f(s_k)$ which corresponds to the starting state of the survivor transition that ends in state s_k at time k, 2) the estimated timing offset $\hat{\tau}_k^f(s_k)$ and 3) the corresponding sample $r_k^f(s_k) = r(kT + \hat{\tau}_k^f(s_k))$. In order to derive the forward timing offset update procedure, consider the two forward transitions approaching state $s_{k+1} = 2$ at time $k+1$ as shown in Fig. 26. The two forward state transition probabilities in the log domain are calculated as [8]

$$\gamma_k^f(1,2) = -\frac{1}{2\sigma_n^2}\|r_k^f(1) - d(1,2)\|^2, \tag{6.7}$$

$$\gamma_k^f(3,2) = -\frac{1}{2\sigma_n^2}\|r_k^f(3) - d(3,2)\|^2, \tag{6.8}$$

where $d(s_k, s_{k+1})$ represents the ideal channel output for a state transition from the state s_k to the state s_{k+1}. Let $\alpha_k(1)$ and $\alpha_k(3)$ be the forward log-probabilities at time k for state $s_k = 1$ and $s_k = 3$ respectively. Assuming that $\alpha_k(3) + \gamma_k^f(3,2) > \alpha_k(1) + \gamma_k^f(1,2)$, we update the quantities at state $s_{k+1} = 2$ at time $k+1$ as,

$$\pi_{k+1}^f(2) = 3 \tag{6.9}$$

$$\hat{\tau}_{k+1}^f(2) = \hat{\tau}_k^f(3) + \mu^f(r_k^f(3)d(3,3) - r_{k-1}^f(3)d(3,2)) \tag{6.10}$$

$$r_{k+1}^f(2) = r((k+1)T + \hat{\tau}_{k+1}^f(2)) \tag{6.11}$$

In (6.10), μ^f denotes the PLL gain for the forward pass. In general we can write the update rules as,

$$\pi^f_{k+1}(s_{k+1}) = \arg\max_{s_k}\{\alpha_k(s_k) + \gamma^f_k(s_k, s_{k+1})\} \tag{6.12}$$

$$\begin{aligned}\hat{\tau}^f_{k+1}(s_{k+1}) &= \hat{\tau}^f_k(\pi^f_{k+1}(s_{k+1})) \\ &+\mu^f[r^f_k(\pi^f_{k+1}(s_{k+1}))d(\pi^f_k(\pi^f_{k+1}(s_{k+1})), \pi^f_{k+1}(s_{k+1})) \\ &-r^f_{k-1}(\pi^f_k(\pi^f_{k+1}(s_{k+1})))d(\pi^f_{k+1}(s_{k+1}), s_{k+1})]\end{aligned} \tag{6.13}$$

$$r^f_{k+1}(s_{k+1}) = r((k+1)T + \hat{\tau}^f_{k+1}(s_{k+1})) \tag{6.14}$$

Forward log-probabilities $\alpha_{k+1}(s_{k+1})$ of all the states at time $k+1$ are updated using the standard forward recursion of BCJR algorithm [8] as

$$\alpha_{k+1}(s_{k+1}) = \ln \sum_{\forall s_k \to s_{k+1}} \exp\{\alpha_k(s_k) + \gamma^f_k(s_k, s_{k+1})\} \tag{6.15}$$

With the update rules mentioned above, all the rows of the received data-matrix are processed sequentially from 1 to n in the forward direction. So at the end of a LARGE forward pass, we have forward log-probabilities $\alpha_k(s_k)$, corresponding to all the bits in the data-matrix.

2. Backward Pass

In the backward direction each row is processed separately by running n parallel equalizers in the backward direction. Consider the example of a backward trellis shown in Fig. 27 for the p^{th} row. The starting states of n backward trellises are perfectly known due to L columns of known bits in the data-matrix. The initial backward timing offsets for the bits in the m^{th} column in the data-matrix are set equal to the corresponding forward timing offsets obtained during the forward pass. As we move ONE step in the backward direction with n equalizers, we calculate

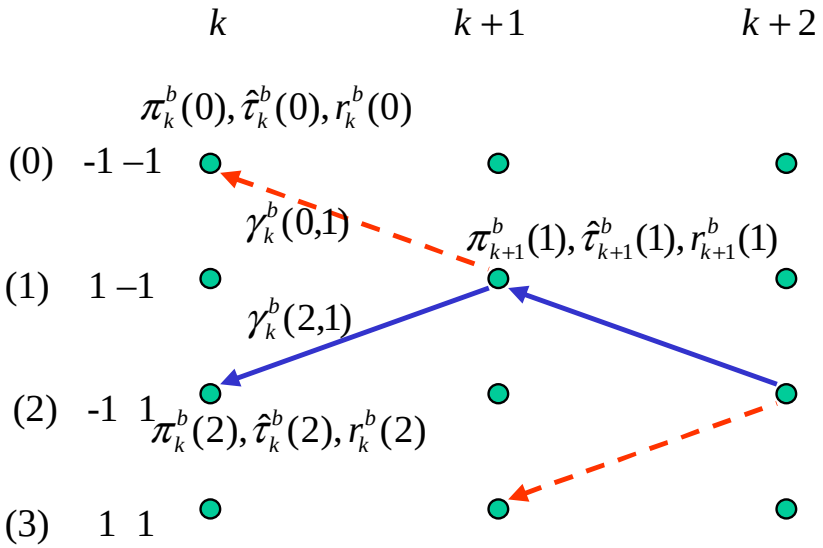

Fig. 27. Backward pass of BCJR-DFE receiver employing per-survivor timing recovery

the LLRs of all the bits of column$-m$ and feed them to the LDPC decoder. The hard decision output of LDPC decoder decides the next backward states of the n backward trellises. So at any given time instant, there are only two possible transitions in the backward direction, originating from the known previous state as shown in Fig. 27. The transitions ruled out by the LDPC decoder are shown by dotted lines in Fig. 27. Similar to the forward state, each backward state s_k at time k maintains three quantities : 1) perfectly known backward state at time $k+1$ as $\pi_k^b(s_k)$, 2) the estimated backward timing offset $\hat{\tau}_k^b(s_k)$ and 3) the corresponding received sample $r_k^b(s_k) = r((k-1)T + \hat{\tau}_k^b(s_k))$. Let $\beta_k(s_k)$ denote the backward log-probability of state s_k and $\gamma_k^b(s_k, s_{k+1})$ denote the backward state transition log-probability for a backward transition from s_{k+1} to s_k. Since the definition of the state $s_k = [c_{k-1}, c_{k-2}]$ includes the bit c_{k-2}, we can calculate the LLR of bit c_{k-2} by marginalizing the forward and backward log-probabilities at all states s_k w.r.t. remaining bits in the

state definition. Thus we can write

$$LLR(c_{k-2}) = \ln \sum_{\substack{\forall s_k: \\ c_{k-2}=+1}} \exp\{\alpha_k(s_k) + \beta_k(s_k)\} - \ln \sum_{\substack{\forall s_k: \\ c_{k-2}=-1}} \exp\{\alpha_k(s_k) + \beta_k(s_k)\} \tag{6.16}$$

$$= \ln \sum_{\substack{\forall s_k \to s_{k+1}: \\ c_{k-2}=+1}} \exp\{\alpha_k(s_k) + \gamma_k^b(s_k, s_{k+1}) + \beta_{k+1}(s_{k+1})\} - \ln \sum_{\substack{\forall s_k \to s_{k+1}: \\ c_{k-2}=-1}} \exp\{\alpha_k(s_k) + \gamma_k^b(s_k, s_{k+1}) + \beta_{k+1}(s_{k+1})\} \tag{6.17}$$

Let $s_{k+1} = s'$ be the known backward state at time $k+1$. Let the two backward transitions be $s' \to s''$ such that $c_{k-2} = +1$ and $s' \to s'''$ such that $c_{k-2} = -1$. As there is only ONE known state at time $k+1$, and only TWO possible backward transitions we can write (6.17) as

$$LLR(c_{k-2}) = [\alpha_k(s'') + \gamma_k^b(s'', s')] - [\alpha_k(s''') + \gamma_k^b(s''', s')] \tag{6.18}$$

In general, at time k we can calculate the $LLR(c_{k-L})$ of all the bits of 'codeword $k-L$'. It is evident from (6.18) that we do not need to calculate and store backward log-probabilities in order to obtain log-likelihood ratios. Thus the total computational complexity of PSP BCJR-DFE forward-backward pass is less than the computational complexity of PSP BCJR forward-backward pass.

In order to explain the backward timing offset update, let us consider Fig. 27. Let state $s_{k+1} = 1$ be the perfectly known state after LDPC decoding at time $k+1$, as shown in Fig. 27. There are two possible backward transitions from state $s_{k+1} = 1$

at time $k+1$ to state $s_k = 0$ and state $s_k = 2$ at time k. Using the backward timing offset $\hat{\tau}^b_{k+1}(1)$, we sample the signal $r(t)$ to get the sample $r^b_{k+1}(1) = r(kT + \hat{\tau}^b_{k+1}(1))$. This sample is used to calculate the state transition probabilities of all the backward transitions originating from state $s_{k+1} = 1$ at time $k+1$. We calculate the two state transition probabilities in the log domain shown in Fig. 27 as

$$\gamma^b_k(0,1) = -\frac{1}{2\sigma_n^2}\|r^b_{k+1}(1) - d(0,1)\|^2, \tag{6.19}$$

$$\gamma^b_k(2,1) = -\frac{1}{2\sigma_n^2}\|r^b_{k+1}(1) - d(2,1)\|^2. \tag{6.20}$$

So by knowing only two possible branch transition metrics $\gamma^b_k(0,1)$ and $\gamma^b_k(2,1)$, we can calculate $LLR(c_{k-L}) = LLR(c_{k-2})$ of bit c_{k-2} of the p^{th} row as

$$LLR(c_{k-2}) = [\alpha_k(2) + \gamma^b_k(2,1)] - [\alpha_k(0) + \gamma^b_k(0,1)], \tag{6.21}$$

where $\alpha_k(2)$ and $\alpha_k(0)$ are the forward log-probabilities at time k of the p^{th} row. We obtain n LLRs from n parallel equalizers and feed them to the LDPC decoder. Assuming that the LDPC decoder sets the next forward state at time k to $s_k = 2$, we update the quantities at that state as

$$\pi^b_k(2) = 1 \tag{6.22}$$

$$\hat{\tau}^b_k(2) = \hat{\tau}^b_{k+1}(1) - \mu^b[r^b_{k+1}(1)d(1,2) - r^b_{k+2}(2)d(2,1)] \tag{6.23}$$

$$r^b_k(2) = r((k-1)T + \hat{\tau}^b_k(2)) \tag{6.24}$$

In (6.23), μ^b denotes the PLL gain for the backward pass. The general rules for the

backward update can be written as,

$$\begin{aligned}
\pi_k^b(s_k) &= \text{previous known state} && (6.25)\\
\hat{\tau}_k^b(s_k) &= \hat{\tau}_{k+1}^b(\pi_k^b(s_k)) - \\
&\quad \mu^b[r_{k+1}^b(\pi_k^b(s_k))d(\pi_k^b(s_k), \pi_{k+1}^b(\pi_k^b(s_k))) \\
&\quad -r_{k+2}^b(\pi_{k+1}^b(\pi_k^b(s_k)))d(s_k, \pi_k^b(s_k))] && (6.26)\\
r_k^b(s_k) &= r((k-1)T + \hat{\tau}_k^b(s_k)) && (6.27)
\end{aligned}$$

Thus as we go in backward direction, we perform joint timing recovery, ISI equalization and decoding, without any global iterations between these three tasks.

D. Simulation Results

We simulated the performance of PSP BCJR-DFE with perpendicular and longitudinal partial response channels with PR2 ($H(D) = 1 + 2D + D^2$) and PR4 ($H(D) = 1 - D^2$) channel impulse response. Timing offsets were generated according to random walk model (6.2). An LDPC code of length $n = 5000$ and rate $R = 0.89$, optimized for the AWGN channel was used. Number of iterations within the LDPC decoder was set to 100. Encoded data was arranged in the data-matrix structure as shown in Fig. 24. Rows of the data-matrix were grouped together as one sector such that number of coded bits in a sector are equal to 5000. Data was transmitted sector by sector and perfect acquisition was assumed at the start of each sector by setting $\tau_k = 0$. At the receiver, to obtain samples at different timing offsets, we used a 21 tap *sinc* interpolation filter given by $g(t) = sin(\pi t/T)/(\pi t/T)$ with $l_{\min} = -10$ and $l_{\max} = 10$ (6.6). In performing LARGE forward pass of the PSP BCJR-DFE algorithm we also set $\hat{\tau}_k = 0$ after each sector. Figure 28 compares the timing recovery performance of PSP BCJR-DFE over PR2 channel with timing jit-

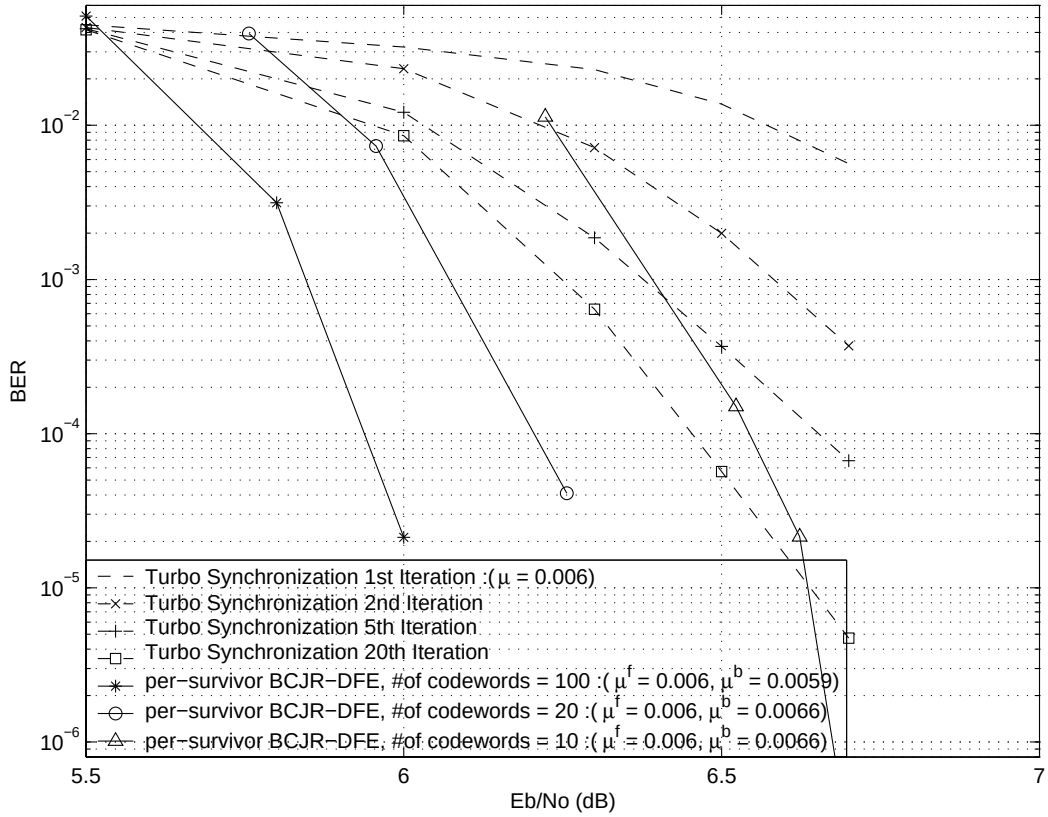

Fig. 28. PSP BCJR-DFE performance over PR2 channel with $\sigma_w/T = 0.3\%$

ter $\sigma_w/T = 0.3\%$ and different number of codewords in the data-matrix. Loss in SNR (equal to $10\log_{10}(\frac{\#ofcodewords}{\#ofcodewords+L})$ dB) due to known bits was accounted for in all simulations. For comparison we also plot the results with turbo synchronization (iterative timing recovery) using PSP BCJR and the same LDPC code of length 5000. Single PLL gain parameter μ was used for forward and backward pass of turbo synchronization, whereas two different PLL gain parameters, μ^f for forward pass and μ^b for backward pass were used for PSP BCJR-DFE. The PLL gain parameters were optimized based on minimizing the RMS error between actual and estimated timing offsets at several SNRs. For backward pass of PSP BCJR-DFE, backward transitions

occur only from known states. So different PLL gain μ^b was used for backward pass of PSP BCJR-DFE where μ^b was optimized with the assumption that all previous decisions are correct. As shown in Fig. 28, PSP BCJR-DFE with number of codewords equal to 100 outperforms turbo synchronization with 20 iterations. At the same time its computational complexity is less than computational complexity of *one* iteration of turbo synchronization. But the overall latency in PSP BCJR-DFE is higher as decoding is delayed until the whole data-matrix is received. Latency can be reduced by decreasing number of codewords in the data-matrix. As we decrease the number

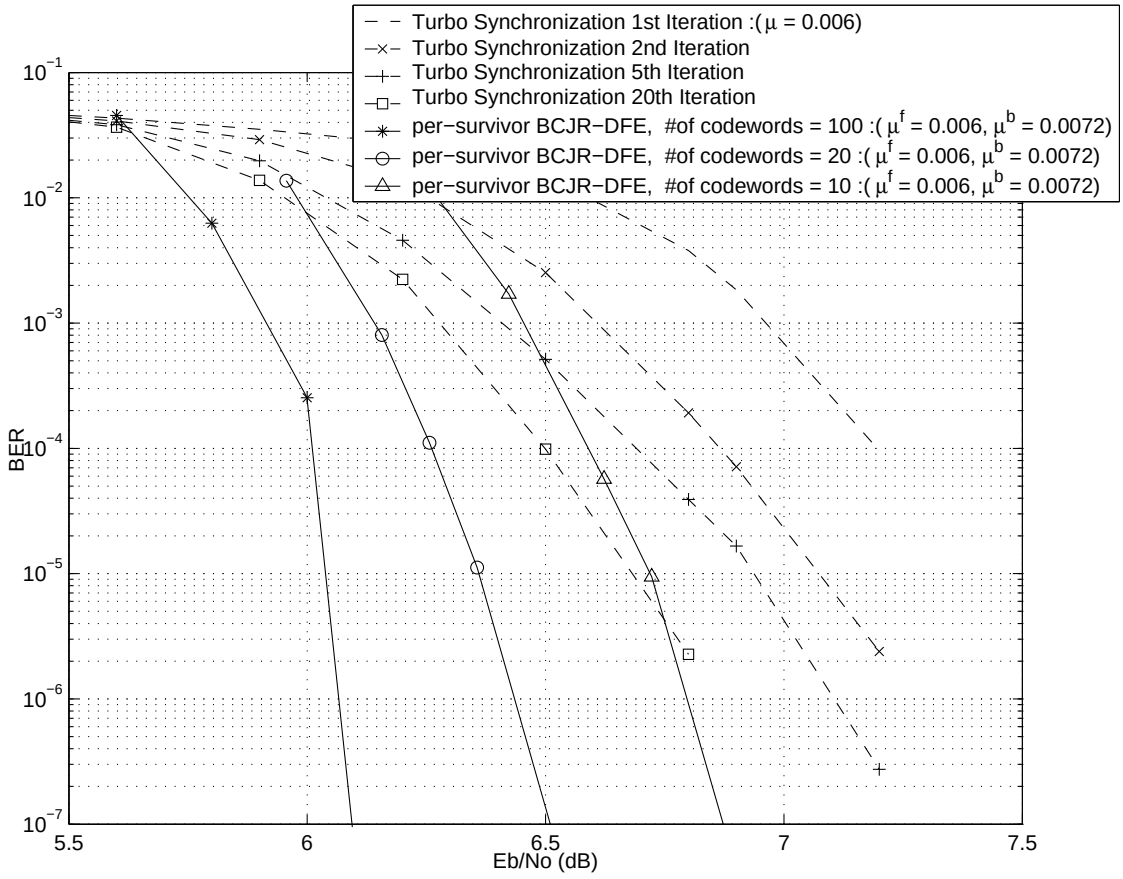

Fig. 29. PSP BCJR-DFE performance over PR2 channel with $\sigma_w/T = 0.5\%$

of codewords in the data-matrix, loss in SNR due to known bits increases and we get different performance with PSP BCJR-DFE as shown in Fig. 28. Thus according to the requirement, one can compromise between latency and performance by selecting appropriate number of codewords in the data-matrix. Similar results are obtained for timing jitter $\sigma_w/T = 0.5\%$ with PR2 channel (Fig. 29). Latency of PSP BCJR-DFE receiver can be reduced by using the latency reduction techniques shown in chapter V. As we increase the timing jitter further to $\sigma_w/T = 0.8\%$, error propagation appears

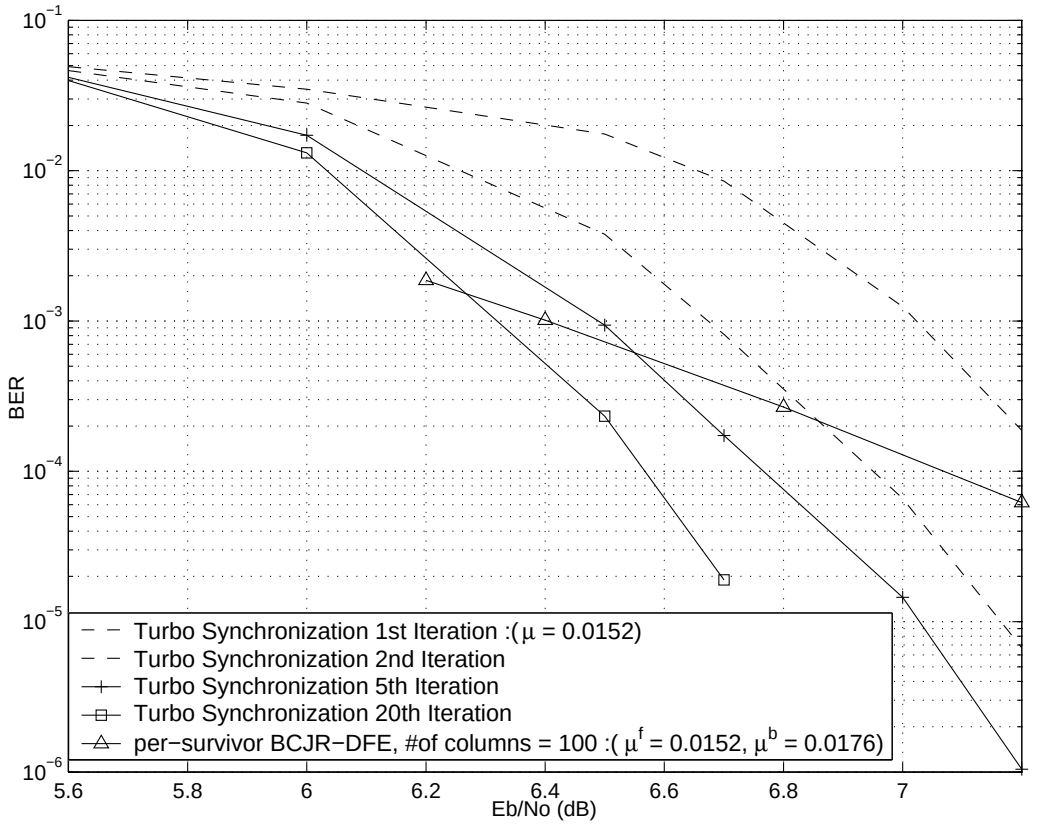

Fig. 30. PSP BCJR-DFE performance over PR2 channel with $\sigma_w/T = 0.8\%$

in PSP BCJR-DFE due to finite length of column code (see Fig. 30). Performance of PSP BCJR-DFE for PR4 channel with $\sigma_w/T = 0.3\%$, and $\sigma_w/T = 0.5\%$ is shown

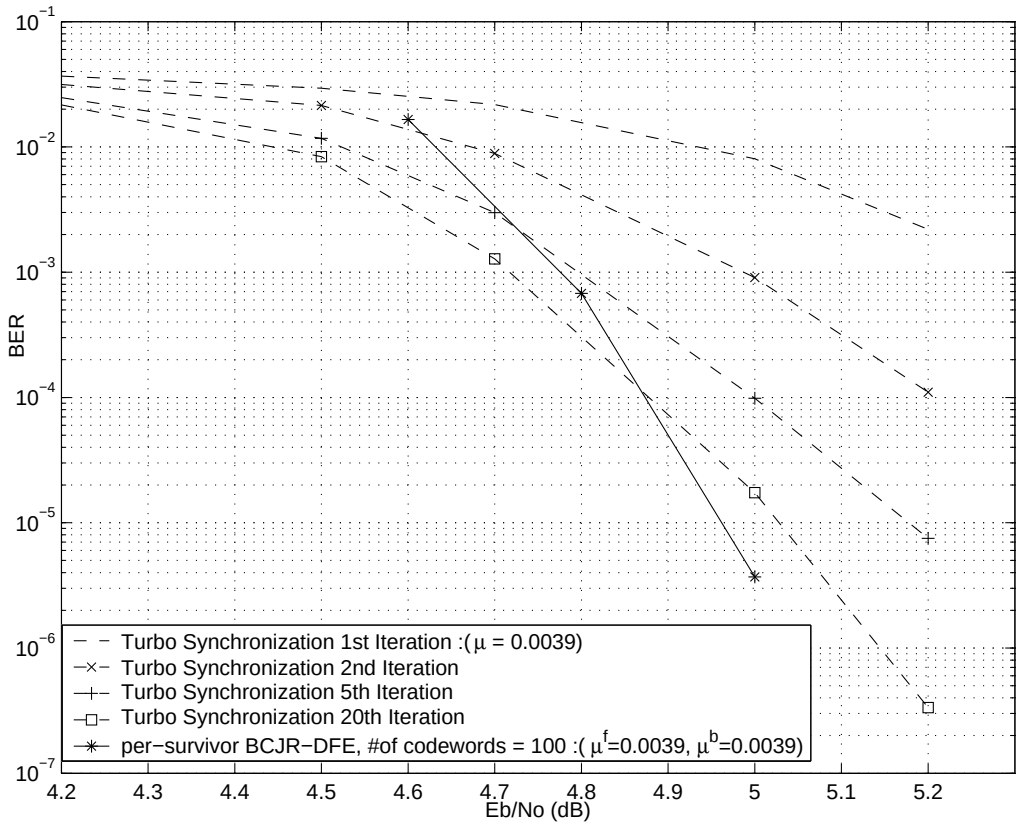

Fig. 31. PSP BCJR-DFE performance over PR4 channel $\sigma_w/T = 0.3\%$

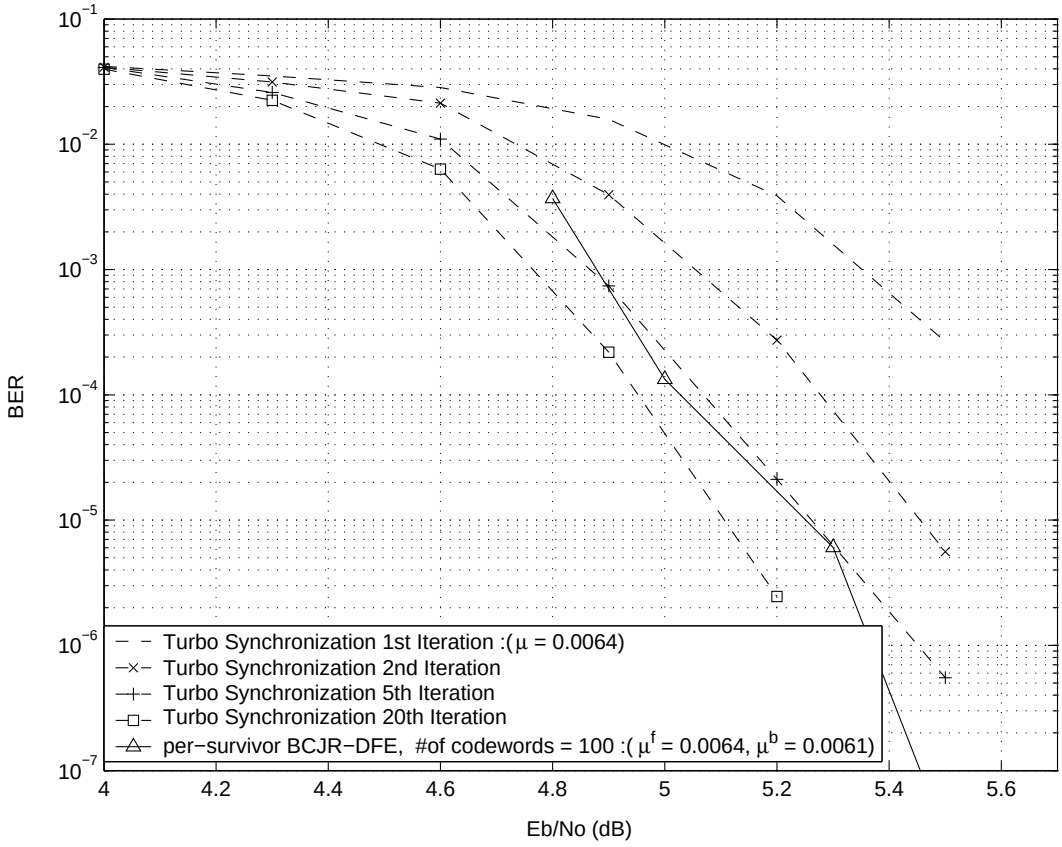

Fig. 32. PSP BCJR-DFE performance over PR4 channel $\sigma_w/T = 0.5\%$

in Fig. 31 and Fig. 32 respectively. We see that the gain of PSP BCJR-DFE over turbo synchronization is less for the PR4 channel as compared to the PR2 channel, nevertheless favorable performance is obtained with PSP BCJR-DFE receiver with significant reduction in computational complexity.

E. Conclusion

We have shown that joint timing recovery, ISI equalization and decoding can be performed using per-survivor processing BCJR-DFE receiver. In many cases, in particular at very low SNR regions, the proposed receiver provides performance advantages over other iterative timing recovery techniques with significant computational savings. Moreover, *no* code optimization tailored to a specific ISI channel is required for the proposed receiver structure.

CHAPTER VII

PERFORMANCE OF BCJR-DFE RECEIVER OVER RECORDING CHANNELS USING PATTERN DEPENDENT NOISE PREDICTION[†]

In this chapter, we investigate the performance of BCJR-DFE receivers for recording channels with pattern-dependent media noise. To account for the media noise, we integrated pattern dependent noise prediction (PDNP) into the BCJR-DFE receiver and evaluated its performance by comparing to conventional iterative (turbo) coded recording channels. Numerical simulations suggest that PDNP BCJR-DFE receivers provide superior or comparable performance as the turbo receiver, albeit, with lower computation complexity and longer processing delay.

A. Introduction

Magnetic recording channels are corrupted by pattern-dependent media noise, in particular transition jitter noise. As linear density of recording increases, media noise becomes the primary system impairment in place of electronics noise. From the detectors' perspective, media noise is correlated as well as pattern-dependent. To account for the correlation and data-dependence of media noise, pattern-dependent noise predictive (PDNP) detection was proposed to improve the performance of traditional partial response maximum likelihood (PRML) detectors [20, 21]. Detectors based on PDNP technique employ pattern-dependent noise whitening filters and are able to provide significant signal-to-noise-ratio (SNR) gains for media noise dominated chan-

[†]©2005 IEEE. Reprinted, with permission, from "Performance of BCJR-DFE based detectors over recording channels using pattern-dependent noise prediction, " N. Nangare, X. Yang, E. Kurtas, and K.R. Narayanan, *Proc. IEEE Intermag.*, Apr. 2005, pp. 983-984, and "Performance of BCJR-DFE based detectors over recording channels using pattern-dependent noise prediction," N. Nangare, X. Yang, E. Kurtas, and K.R. Narayanan, *IEEE Trans. Magn.*, Oct. 2005, pp. 2971-2973.

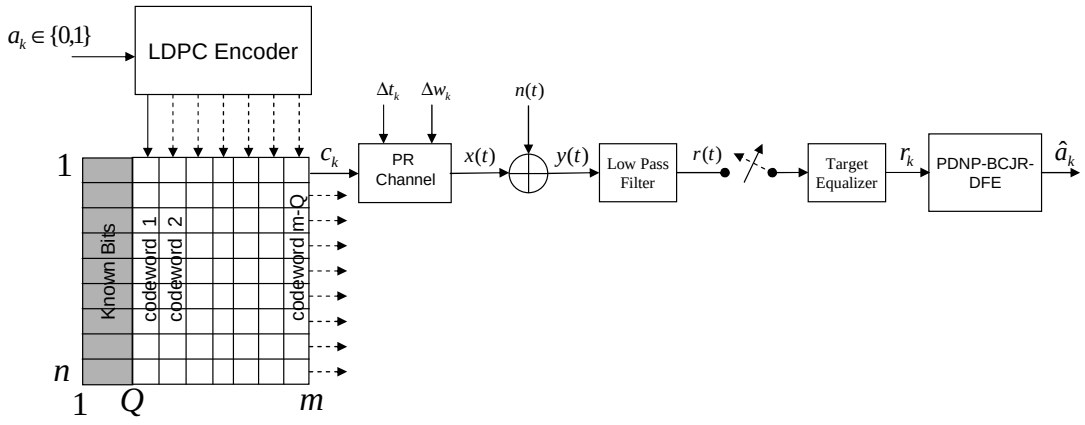

Fig. 33. System model for PDNP BCJR-DFE receiver

.

nel. PDNP detection can also be incorporated into turbo coded channels [1] with iterative signal processing, which furnish additional performance gain.

In this chapter, we investigate the performance of BCJR-DFE receiver in the presence of pattern-dependent media noise over recording channels. To account for the pattern-dependence of noise, we incorporate pattern-dependent noise prediction into the BCJR-DFE receiver, giving rise to the so-called PDNP BCJR-DFE receiver. The performance of the receiver is then tested and benchmarked with conventional turbo decoding under various operating conditions.

B. System Model

Figure 33 shows coded magnetic recording system incorporating PDNP BCJR-DFE receiver. Encoder structure of the PDNP BCJR-DFE receiver is same as that of BCJR-DFE except that more known bits per row are used to accommodate the embedded noise prediction. Let $c_{k-M}^{k} = \{c_{k-M}, \cdots, c_k\}$ with nonnegative integer M denote the window of recorded data pattern used to identify one of the 2^{M+1} noise

prediction filters. If L denotes the memory of the channel and I denotes the prediction filter length, then we need to set a trellis of memory equal to $Q = \max(I + L, M)$ for noise prediction [21]. Consequently, to reset the trellis memory, the number of known bits required becomes Q bits for each row. As shown in Fig. 33, Q columns of known bits are arranged at the beginning of the data-matrix. User data bits are encoded with an LDPC encoder into codewords of length n. Codewords 1 to $m - Q$ are arranged along the columns of the data-matrix as shown. Encoded data is recorded onto a magnetic medium (using saturation recording) row-wise, from row-1 to row-n. Read-head senses the magnetic flux from the media and converts it to an electrical signal. This read-back signal is corrupted by white electronic noise and pattern dependent media noise due to random transition jitter variation Δt_k and random pulse width variation Δw_k present in the channel transition response (see chapter II). This read-back signal $y(t)$ is then filtered with the low pass filter, which restricts the signal energy to be further down sampled at the baud-rate. Baud-rate-sampled-signal is then passed to a target-equalizer, which equalizes the effective signal to known partial response (PR) target. Output of the target-equalizer is given to the detector.

C. PDNP BCJR-DFE Receiver

PDNP BCJR-DFE receiver uses BCJR-DFE algorithm along with path dependent noise prediction which is embedded inside BCJR trellis. Let s_k denote the trellis state at time k, and $\zeta \triangleq s_k \rightarrow s_{k+1}$ denote the state transition at time k. Let $\{r_k\}$ denote the received noisy read-back signal. For each state transition ζ, we can calculate the state transition probability $\gamma_k(s_k, s_{k+1})$ as

$$\gamma_k(s_k, s_{k+1}) = Pr(r_k|s_k, s_{k+1})$$
$$= \frac{1}{\sqrt{2\pi\sigma^2(\zeta)}} \exp\left[-\frac{(r_k - o_k(\zeta) - \hat{n}_k(\zeta))^2}{2\sigma^2(\zeta)}\right] \tag{7.1}$$

where $o_k(\zeta)$ is the noiseless signal, $\hat{n}_k(\zeta)$ is the predicted noise sample and $\sigma^2(\zeta)$ is residual noise variance, all corresponding to the transition ζ. The predicted noise sample $\hat{n}_k(\zeta)$ is obtained using the prediction filter, specific to the pattern which is completely determined by the transition ζ. With a slight abuse of the notation, the prediction filter is denoted by $\mathbf{p}(\zeta) = [p_1(\zeta), \cdots, p_I(\zeta)]$ and

$$\hat{n}_k(\zeta) = \sum_{i=1}^{I} p_i(\zeta)[r_{k-i} - o_{k-i}(\zeta)]. \tag{7.2}$$

As described in [21], the noise prediction filters, $\mathbf{p}$, and corresponding residual variances σ's are obtained via training. The first step in PDNP BCJR-DFE decoding is to execute the backward recursion of BCJR employing pattern-dependent noise prediction for each row of data during which backward state-probabilities β's [8] are calculated for all the bits in the data-matrix as

$$\beta_k(s_k) = \sum_{s_{k+1}} \beta_{k+1}(s_{k+1})\gamma_k(s_k, s_{k+1}). \tag{7.3}$$

This is similar to the backward recursion in a regular BCJR with embedded PDNP. The second step involves forward recursion for n rows of the data-matrix which can be performed in parallel using n equalizers. In this step *a-posteriori* probabilities (APPs) are calculated for all the bits in a single column at a time, given that previous states are known for n equalizers. Let $s_k = s$ be the known state for a particular equalizer at time k, with $s_{k+1} = s'$ (when $c_k = 1$) and $s_{k+1} = s''$ (when $c_k = 0$) being the *only* two possible states at time $(k+1)$. We can calculate the APP of bit c_k of a particular

row as (see chapter IV, section C)

$$APP(c_k) = \frac{\gamma_k(s, s')\beta_{k+1}(s')}{\gamma_k(s, s'')\beta_{k+1}(s'')}. \tag{7.4}$$

Note that calculation of forward state-probabilities α's [8] is not required to obtain APP's. These soft outputs form as the input to the column-code decoder. In turn, the output from the column-code decoder are used for deciding the states in the trellis at time $(k+1)$. Thus for the first column, the states are pinned by using the pilot bits, while for the other columns, the states are determined through the decoded results from the previous column. This process continues until the complete data-matrix is decoded.

D. Simulation Results

In order to examine the performance of PDNP BCJR-DFE receiver over media noise dominated recording channel, we simulated both longitudinal and perpendicular recording channels with normalized linear density equal to 2.5. For longitudinal recording, Lorenzian pulse is used as transition response and the channel is equalized to PR4 ([1 0 -1]) target. For perpendicular recording, error function is used as the transition response and the channel is equalized to PR2 ([1 2 1]) target [16]. For both channels, 8 different noise prediction filters of length 3 were used. Specifically, the data patterns for the filters are defined by $\{c_{k-2}, ..., c_k\}$ (with $M = 2$), where c_k's are the channel input bits (see Fig. 9). We set $m - Q = 100$ (i.e. used 100 codewords in the data-matrix) and LDPC code of length $n = 5000$, optimized for AWGN channel with rate $R = 0.89$ was used. Rate loss due to known bits was accounted for in the simulation. Figures 34 and 35 show the performance for PR4 equalized longitudinal channel with different jitter noise and electronics noise mixtures. For

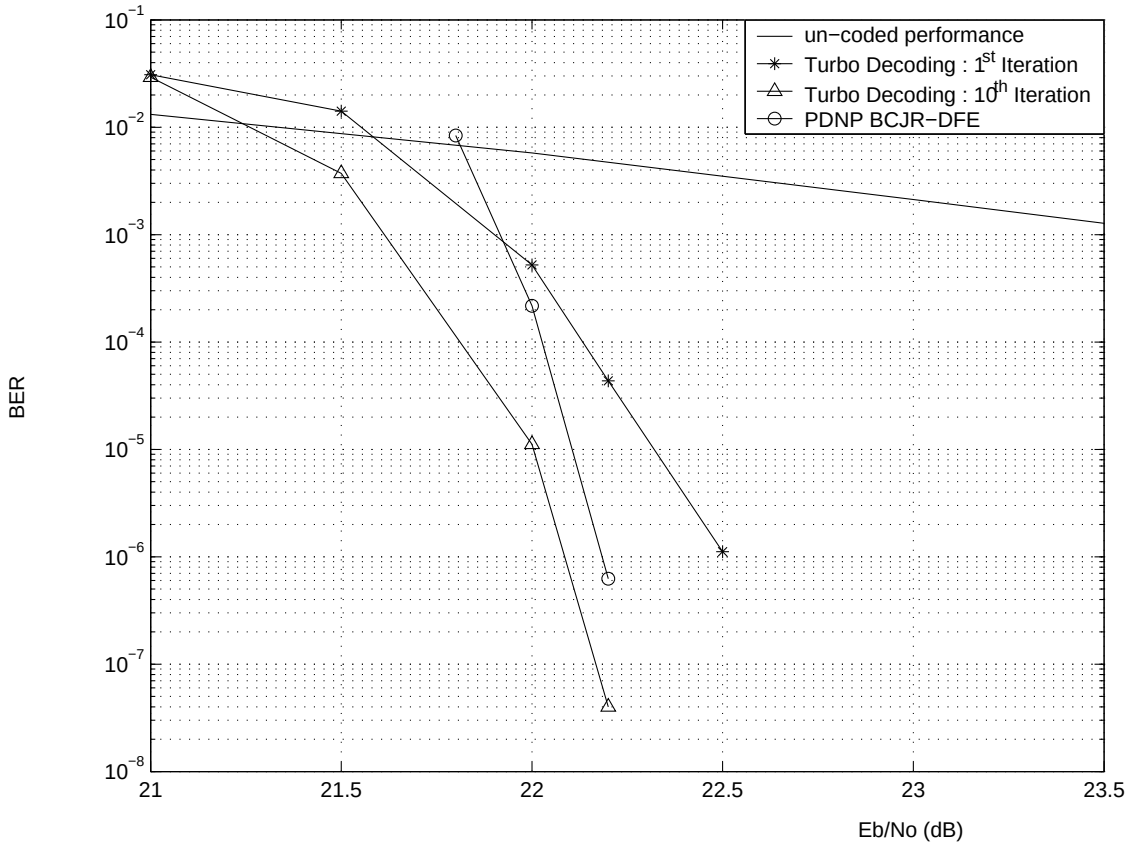

Fig. 34. PDNP BCJR-DFE performance over PR4 channel with 60% jitter noise.

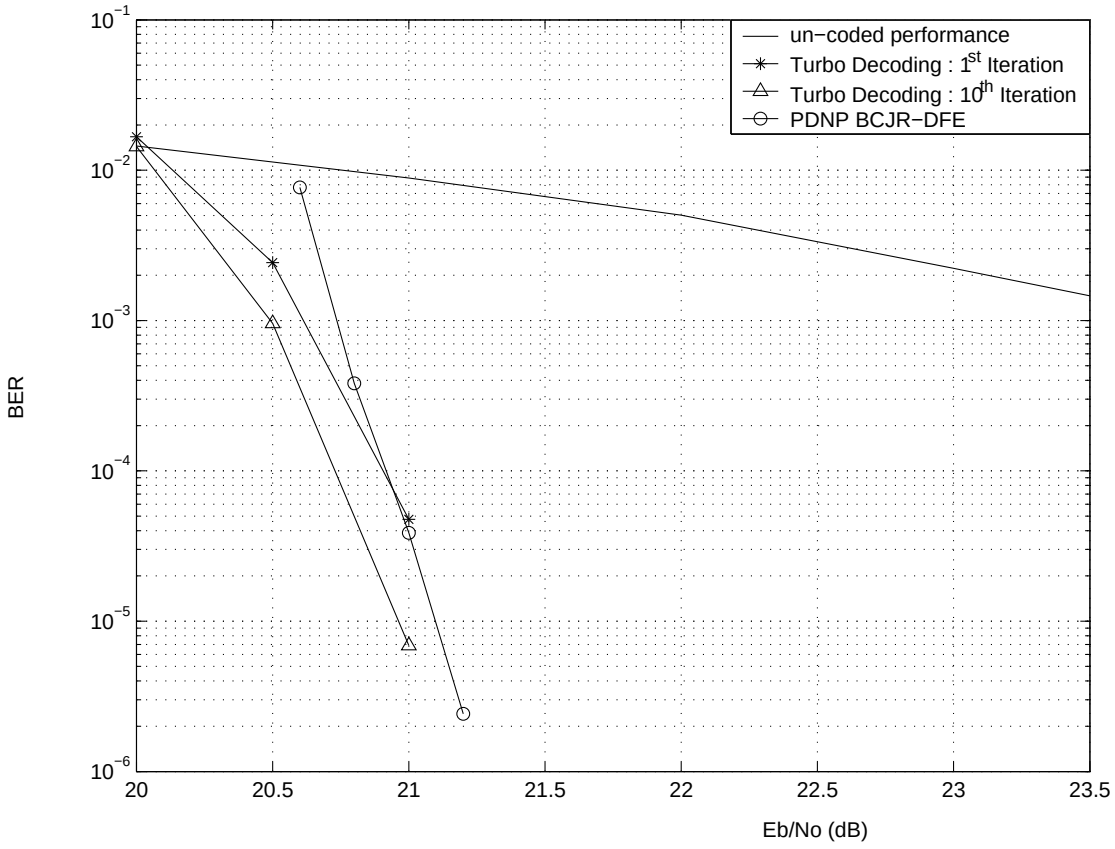

Fig. 35. PDNP BCJR-DFE performance over PR4 channel with 90% jitter noise.

comparison, we also plot the results of turbo decoding using PDNP-BCJR and the same LDPC code with 10 turbo iterations as well as the results of an uncoded channel with PDNP-BCJR receiver. We observe that PDNP BCJR-DFE has comparable performance to that of turbo decoding, although no global iterations are used for the PDNP BCJR-DFE receiver. Figures 36 and 37 show the performance for PR2 equalized perpendicular channel. For a perpendicular channel, PDNP BCJR-DFE

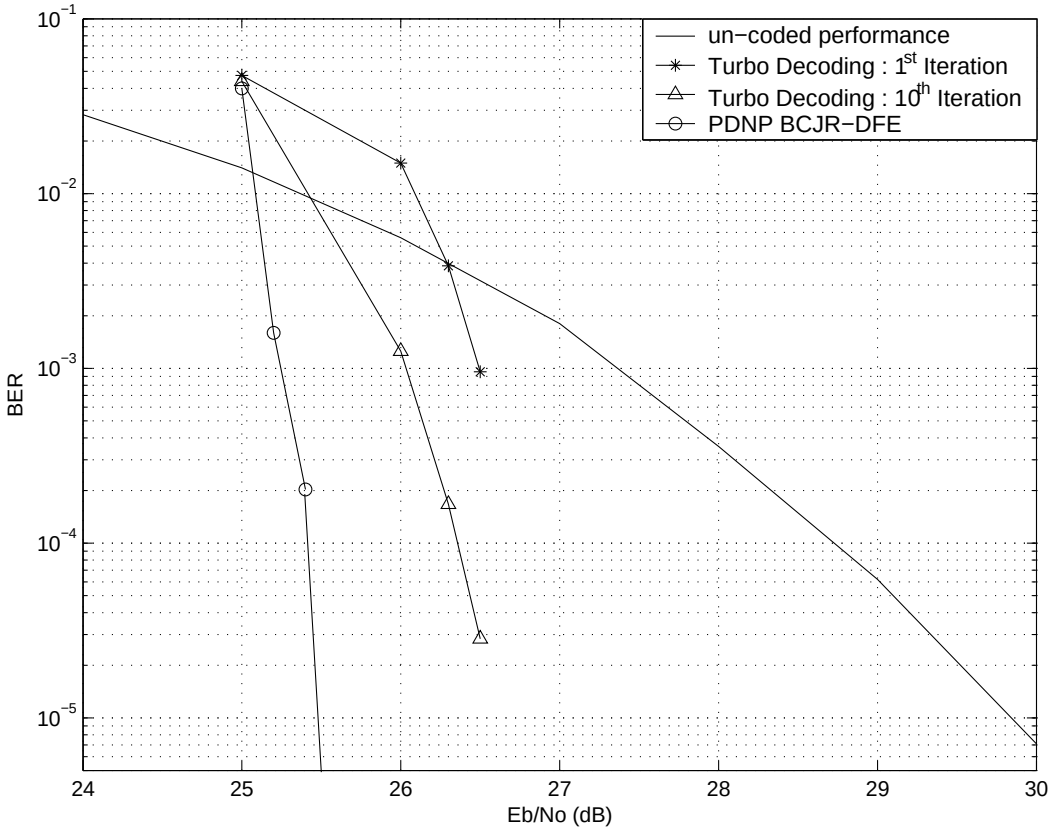

Fig. 36. PDNP BCJR-DFE performance over PR2 channel with 60% jitter noise.

outperforms turbo decoding with 10 iterations. The loss in performance of turbo decoding over PDNP BCJR-DFE is different for PR4 and PR2 channel due to the fact that performance of turbo decoding is strongly dependent on the channel realization.

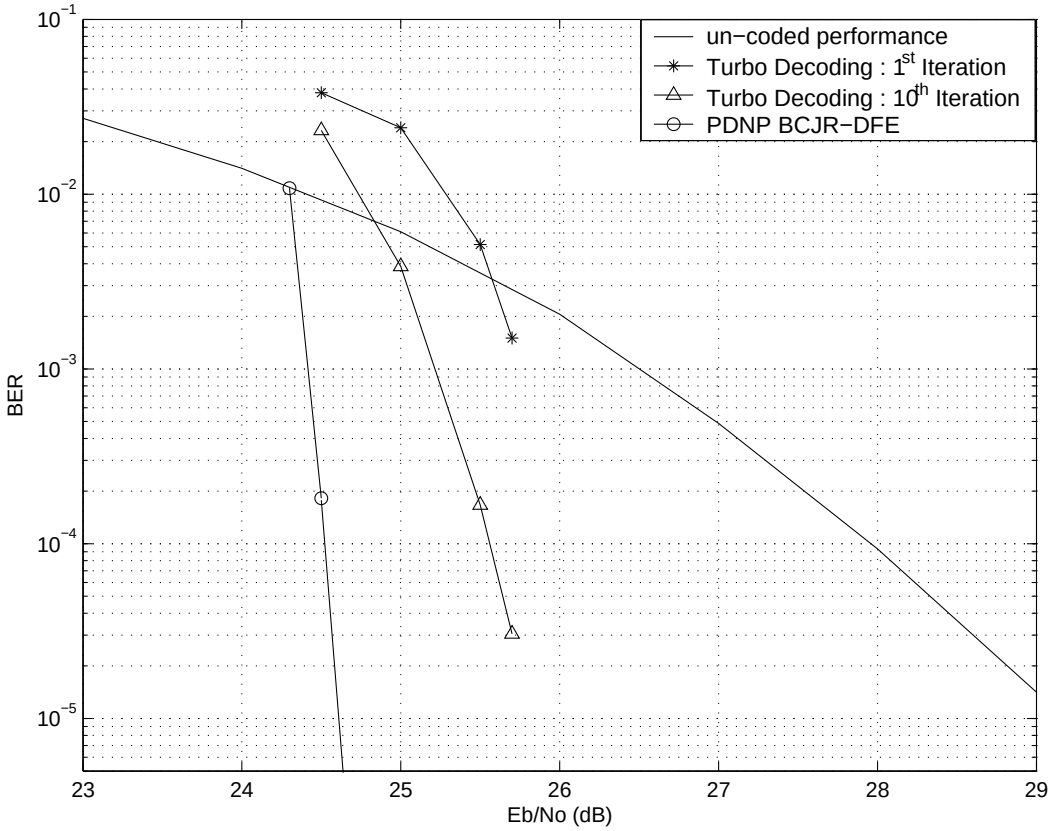

Fig. 37. PDNP BCJR-DFE performance over PR2 channel with 90% jitter noise.

It should be emphasized that, in the simulations above, the PDNP BCJR-DFE receiver, which is free of global iterations, has much less computational complexity (more than 10 times less) as compared to the conventional turbo-decoding algorithm. Furthermore, the recording channel is indeed perceived as a memoryless channel at the receiver and consequently, no specific code optimization is required for the column-code encoder for achieving optimal performance. The latency of PDNP BCJR-DFE receiver is higher than the latency of turbo-decoding algorithm. The latency reduction techniques shown in chapter V can be used along with PDNP BCJR-DFE receiver to obtain a low latency solution.

E. Conclusion

We have shown the applicability of BCJR-DFE receiver with embedded pattern dependent noise prediction for magnetic recording channels. This *non-iterative* receiver was demonstrated to perform superior or comparable to conventional turbo decoding, albeit with significantly less computational complexity and requires no *ad-hoc* code optimization.

CHAPTER VIII

SUMMARY AND FUTURE RESEARCH

After the invention of turbo codes in 1993 [1], turbo principle was applied to many other signal processing problems. BCJR-DFE receiver is a general receiver which operates on the Markov trellis structure. Similar to the turbo principle, the BCJR-DFE principle can also be applied to many other signal processing algorithms which can be implemented using a Markov trellis structure.

A. Timing Synchronization with Markov Model

Recently timing errors were characterized using a Markov model by Zeng and Kavcic [76]. The authors used a joint trellis to represent ISI and timing errors. The Markov model parameters for the timing error process can be obtained using the read-back waveform for the case of the magnetic recording channel [77]. Under the assumption of Markov nature of timing errors, it can be shown that the BCJR-DFE receiver (when implemented on this joint trellis) achieves the i.i.d. capacity of any ISI channel with timing errors (see chapter IV, section D).

B. Joint Demodulation and Decoding

Iterative demodulation and decoding of convolutionally encoded data with different modulation schemes was studied in [3, 78]. An iterative demodulation and decoding scheme using LDPC codes and bit-interleaved coded modulation (BICM) was studied in [79]. Non-coherent iterative demodulation and decoding was considered in [80]. Many modulation schemes with memory can be represented using a Markov trellis structure. Thus, as opposed to iterative demodulation and decoding, one can use

BCJR-DFE principle to perform coherent/non-coherent non-iterative joint demodulation and decoding.

C. Computing Lower Bounds on the Capacity

In chapter IV, we proved that the BCJR-DFE receiver achieves the i.i.d. capacity of an ISI channel when the column code used achieves the i.i.d. capacity of an equivalent memoryless channel as seen by the column code. We can simulate a hypothetical column code which always gives the correct decisions, i.e. we assume perfect decision feedback for BCJR-DFE receiver and calculate the likelihood ratio Λ_i of input bit X_i. Thus for any given ISI channel we can obtain the i.i.d. capacity as $C_{i.i.d.} = I(X_i; \Lambda_i)$, which is a lower bound on the true capacity, $(C \leq C_{i.i.d.})$. In chapter VI and VII we effectively used the per-survivor signal processing along with BCJR-DFE receiver. When perfect past decisions are assumed, we can obtain a lower bound on the true capacity for the case of an ISI channel with timing errors or pattern dependent correlated noise. Here perfect past decisions help in accurate estimation of the unknown parameters like timing errors or pattern dependent correlated noise. This can be easily extended to other signal processing functions.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE International Conference on Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *European Trans. on Telecommunications*, vol. 6, no. 5, pp. 507–511, Sept./Oct. 1995.

[3] K. R. Narayanan and G. L. Stüber, "A serial concatenation approach to iterative demodulation and decoding," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 956–961, July 1999.

[4] M. Tüchler, R. Otnes, and A. Schmidbauer, "Performance of soft iterative channel estimation in turbo equalization," in *IEEE Proc. Int. Conf. Commun. ICC*, Apr./May 2002, pp. 1858–1862.

[5] J. Barry, A. Kavčić, S. W. McLaughlin, A. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Processing Mag.*, vol. 21, no. 1, pp. 89–102, Jan. 2004.

[6] W. Root and P. Varaiya, "Capacity of classes of Gaussian channels," *SIAM J. Appl. Math.*, vol. 16, no. 6, pp. 1350–1393, Sept. 1968.

[7] V. Gulati, "Concatenated codes for the multiple-input multiple-output quasi-static fading channel," Ph.D. dissertation, Texas A&M University, College Station, TX, 2004.

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

[9] K. R. Narayanan and N. Nangare, "A BCJR-DFE based receiver for achieving near capacity performance on ISI channels," in *42nd Allerton Conference on Communication, Control and Computing*, Monticello, IL, Sept. 2004.

[10] K. R. Narayanan and N. Nangare, "A BCJR-DFE based receiver for achieving near capacity performance on ISI channels," *IEEE Trans. Commun.*, 2006, submitted.

[11] N. Nangare, K. R. Narayanan, X. Yang, and E. Kurtas, "Joint timing recovery, ISI equalization and decoding using per-survivor BCJR-DFE," in *Proc. IEEE Globecom*, Nov./Dec. 2005, pp. 1620–1624.

[12] N. Nangare, X. Yang, E. Kurtas, and K. R. Narayanan, "Performance of BCJR-DFE based detectors over recording channels using pattern-dependent noise prediction," in *Proc. IEEE Intermag*, Nagoya, Japan, Apr. 2005, pp. 983–984.

[13] N. Nangare, X. Yang, E. Kurtas, and K. R. Narayanan, "Performance of BCJR-DFE based detectors over recording channels using pattern-dependent noise prediction," *IEEE Trans. Magn.*, vol. 41, no. 10, pp. 2971–2973, Oct. 2005.

[14] A. Kavčić and J. M. F. Moura, "Statistical study of zig -zag transition boundaries in longitudinal digital magnetic recording," *IEEE Trans. Magn.*, vol. 33, no. 6, pp. 4482–4492, Nov. 1997.

[15] J. Moon, "Signal-to-noise ratio definition for magnetic recording channels with transition noise," *IEEE Trans. Magn.*, vol. 36, no. 5, pp. 3881–3883, Sept. 2000.

[16] X. Yang and E.M. Kurtas, *Signals and Noise Generation for Magnetic Recording Channel Simulations*, pp. 6.1–6.20, New York: CRC Press, 2004.

[17] J. Moon and W. Zeng, "Equalization for maximum likelihood detectors," *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083–1088, Mar. 1995.

[18] G.D. Forney Jr., "The Viterbi algorithm," in *Proc. of the IEEE*, Mar. 1973, vol. 61, pp. 268–278.

[19] E. Eleftheriou and W. Hirt, "Noise-predictive maximum-likelihood (NPML) detection for the magnetic recording channel," in *IEEE Proc. Int. Conf. Commun. ICC*, June 1996, pp. 556–560.

[20] A. Kavčić and J.M. Moura, "The Viterbi algorithm and markov noise memory," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 291–301, Jan. 2000.

[21] J. Moon and J. Park, "Pattern-dependent noise prediction in signal-dependent noise," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 730–743, Apr. 2001.

[22] A. Kavčić, "Detection in high density magnetic recording: A matched model approach," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 1998.

[23] J. Moon and L. R. Carley, "Performance comparison of detection methods in magnetic recording," *IEEE Trans. Magn.*, vol. 26, no. 6, pp. 3155–3172, Nov. 1990.

[24] C.E. Shannon, "A mathematical theory of communication," in *Bell System Technical Journal*, July/Oct. 1948, vol. 27, pp. 611–644 and 623–656.

[25] R. G. Gallager, "Low-density parity check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[26] R. G. Gallager, *Low-Density Parity Check Codes*, Cambridge, MA: MIT press, 1963.

[27] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[28] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[29] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for 'turbo-like' codes," in *36th Allerton Conference on Communication, Control and Computing*, Monticello, IL, Sept. 1998, pp. 201–210.

[30] H. Jin, A. Khandekar, and R. McEliece, "Irregural repeat-accumulate codes," in *Proc. IEEE International Symposium on Turbo Codes*, Brest, France, Sept. 2000, pp. 1–8.

[31] B. J. Frey and D. J. C. MacKay, "Irregular turbo-like codes," in *Proc. IEEE International Symposium on Turbo Codes*, Brest, France, Sept. 2000, pp. 67–72.

[32] R. J. McEliece, "Are turbo-like codes effective on nonstandard channels?," in *Proc. IEEE Sym. on Information Theory (ISIT)*, Washington, DC, Dec. 2001, pp. 1–8.

[33] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710–1722, Nov. 1996.

[34] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.

[35] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981.

[36] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. of 29th Annual ACM Symp. on the Theory of Computing*, 1997, pp. 150–159.

[37] S. Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[38] S. Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-desity parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[39] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[40] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 12, pp. 3017–3028, Dec. 2002.

[41] T. Richardson, A. Shokrollahi, and R. L. Urbanke, "Design of capacity approaching irregular low density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[42] C. Jones, A. Matache, T. Tian, J. Villasenor, and R. Wesel, "The universality of LDPC codes on wireless channels," in *Proc. IEEE Conf. on Military Communications MILCOM*, Oct. 2003, vol. 1, pp. 440–445.

[43] G.D. Forney Jr., "Maximum-likelihood seauence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. 18, no. 3, pp. 363–378, May 1972.

[44] J. G. Zhu and N. H. Bertram, "Recording and transition noise simulations in thin film media," *IEEE Trans. Magn.*, vol. 24, pp. 2706–2708, Nov. 1988.

[45] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channels with memory," in *Proc. IEEE Intl. Conf. on Commun. ICC*, June 2001, pp. 2692–2695.

[46] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite state ISI channels," in *Proc. IEEE Globecom '01*, Nov. 2001, vol. 5, pp. 2992–2996.

[47] V. Sharma and S. K. Singh, "Entropy and channel capacity in the regenerative setup with applications to markov channels," in *Proc. IEEE Sym. on Information Theory (ISIT)*, Washington, DC, June 2001, p. 283.

[48] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," in *Proc. IEEE International Symposium on Turbo Codes*, Brest, France, Sept. 1997, pp. 1–11.

[49] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, July 1999.

[50] M. Tüchler, A. Singer, and R. Koetter, "Minimum mean square error equalization with priors," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 673–683, Mar. 2002.

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

[52] K. R. Narayanan, "Effect of precoding on the convergence of turbo equalization for partial response channels," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 686–698, Apr. 2001.

[53] D. N. Doan and K. R. Narayanan, "Some new results on code design for inter-symbol interference ISI channels with turbo equalization," *IEEE Trans. Commun.*, accepted subject to revision.

[54] K. R. Narayanan, D. N. Doan, and R. V. Tamma, "Design and analysis of LDPC codes for turbo equalization with optimal and sub-optimal soft output equalizers," in *40th Allerton Conference on Communication, Control and Computing*, Monticello, IL, Oct. 2002.

[55] K. R. Narayanan, X. Wang, and G. Yue, "Estimating the PDF of the SIC-MMSE equalizer output and its applications in designing LDPC codes with turbo equalization," *IEEE Trans. Wireless Commun.*, vol. 4, no. 1, pp. 278–287, Jan. 2005.

[56] A. Kavčić, M. Xiao, and N. Varnica, "Matched information rate codes for partial response channels," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 973–989, Mar. 2005.

[57] N. Varnica and A. Kavčić, "Optimized low-density parity-check codes for partial response channels," *IEEE Communications Letters*, vol. 7, no. 4, pp. 168–170, Apr. 2003.

[58] N. Varnica, "Iteratively decodable codes for memoryless and intersymbol interference channels," Ph.D. dissertation, Harvard University, Cambridge, MA, 2005.

[59] S. ten Brink, "Convergence behaviour of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[60] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[61] K. Bhattad and K. R. Narayanan, "An MSE based transfer chart to analyze iterative decoding schemes," *IEEE Trans. Inform. Theory*, 2006, submitted.

[62] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Signal Processing*, vol. 51, no. 11, pp. 2764–2772, Nov. 2003.

[63] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[64] J. M. Cioffi, G. P. Dudevoir, M. V. Eyuboglu, and Jr. G. D. Forney, "MMSE decision-feedback equalizers and coding–part I: Equalization results," *IEEE Trans. Commun.*, vol. 43, no. 10, pp. 2582–2594, Oct. 1995.

[65] J. M. Cioffi, G. P. Dudevoir, M. V. Eyuboglu, and Jr. G. D. Forney, "MMSE decision-feedback equalizers and coding–part II: Coding results," *IEEE Trans. Commun.*, vol. 43, no. 10, pp. 2595–2604, Oct. 1995.

[66] M. Varanasi and T. Guess, "Optimum decision feedback multiuser equalization with successive decoding achieves the total capacity of the gaussian multiple-access channel," in *Proc. of Asilomar Conf. on Signals, Systems & Computers*, 1997, pp. 1405–1409.

[67] S. Fechtel and M. Meyr, "Combined equalization, decoding and antenna diversity combining for mobile/personal digital radio transmission using feed forward synchronization," in *Proc. Vehicular Technology Conference*, May 1993, pp. 633–636.

[68] M.V. Eyuboglu, "Detection of coded modulation signals on linear, severely distorted channels using decision-feedback noise prediction with interleaving," *IEEE Trans. Commun.*, pp. 401–409, Apr. 1998.

[69] J. Proakis, *Digital Communications*, Singapore: McGraw-Hill, third edition, 1995.

[70] J. Campello, D. S. Modha, and S. Rajagopalan, "Designing LDPC codes using bit-filling," in *Proc. IEEE Intl. Conf. on Commun. ICC*, June 2001, pp. 55–59.

[71] A. Nayak, J. Barry, and S.W. McLaughlin, "Joint timing recovery and turbo equalization for coded partial response channels," *IEEE Trans. Magn.*, vol. 38, no. 5, pp. 2295–2297, Sept. 2003.

[72] R. Raheli, A. Polydoros, and C.-K. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. 43, no. 2, pp. 354–364, Feb./Mar./Apr. 1995.

[73] P. Kovintavewat, J. Barry, M. Erden, and E. Kurtas, "Per-survivor iterative timing recovery for coded partial response channels," in *Proc. IEEE Globecom*

'04, Dallas, Texas, Nov. 2004, pp. 2604–2608.

[74] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.

[75] K. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. 24, no. 5, pp. 516–531, May 1976.

[76] W. Zeng and A. Kavčić, "Optimal soft-output detector for channels with intersymbol interference and timing errors," *IEEE Trans. Magn.*, vol. 39, no. 5, pp. 2555–2557, Sept. 2003.

[77] W. Zeng, A. Kavčić, and R. Motwani, "Extraction of timing error parameters from readback waveforms," in *IEEE Magnetic Recording Conference*, Stanford, CA, Aug. 2005.

[78] P. Moqvist and T. Aulin, "Power and bandwidth efficient serially concatenated CPM with iterative decoding," in *Proc. IEEE Globecom*, Nov./Dec. 2000, vol. 2, pp. 790–794.

[79] R. Narayanaswami, "Coded modulation with low density parity check codes," M.S. thesis, Texas A&M University, College Station, TX, 2001.

[80] G. Colavolpe, G. Ferrari, and R. Raheli, "Noncoherent iterative (turbo) decoding," *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1488–1498, Sept. 2000.

APPENDIX A

THE BCJR ALGORITHM FOR ISI CHANNEL EQUALIZATION

An ISI channel can be represented as a finite state machine which has a trellis structure. The BCJR algorithm can be used on this trellis structure to find the a posteriori probabilities (APPs) of the symbols transmitted over an ISI channel. Consider the example of an ISI channel of memory L. Let $X_1^N = \{X_1, \cdots, X_N\}$ be the input sequence of N BPSK symbols and $Y_1^N = \{Y_1, \cdots, Y_N\}$ be the corresponding output sequence such that

$$Y_k = \sum_{i=0}^{L} h_i \; X_{k-i} + N_k, \tag{A.1}$$

Here h_i's are the channel tap coefficients (assumed to be complex) and N_k is complex, additive white Gaussian noise with variance $\mathcal{N}_0$. The APP of symbol X_k transmitted at time k can be written as

$$\text{APP}(x_k) = P(X_k = x_k | Y_1^N) \tag{A.2}$$

The optimum soft output is the likelihood ratio which is defined as

$$\Lambda(x_k) = \frac{P(X_k = +1 | Y_1^N)}{P(X_k = -1 | Y_1^N)} \tag{A.3}$$

For any given channel trellis state S_k at time k, the forward state probability $\alpha_k(S_k)$ and the backward state probability $\beta_k(S_k)$ of the BCJR algorithm are defined as,

$$\alpha_k(S_k) = P(S_k = s_k, Y_1^k) \tag{A.4}$$

$$\beta_k(S_k) = P(Y_{k+1}^N | S_k = s_k) \tag{A.5}$$

These state probabilities can be obtained using *forward recursion* and *backward recursion* as follows

$$\alpha_k(S_k) = \sum_{S_k} \alpha_{k-1}(S_{k-1})\gamma_{k-1}(X_{k-1}, S_{k-1}, S_k) \tag{A.6}$$

$$\beta_k(S_k) = \sum_{S_k} \beta_{k+1}(S_{k+1})\gamma_k(X_k, S_k, S_{k+1}) \tag{A.7}$$

Here $\gamma_k(X_k, S_{k-1}, S_k)$ is the state transition probability which can be expressed as

$$\begin{aligned}\gamma_k(X_k, S_k, S_{k+1}) &= P(Y_k|S_k, S_{k+1}, X_k = x_k) \\ &= P(X_k = x_k)\frac{1}{\sqrt{\pi\mathcal{N}_0}} \exp\left(-\frac{||Y_k - \sum_{i=0}^{L} h_i\ X_{k-i}||^2}{\mathcal{N}_0}\right)\end{aligned} \tag{A.8}$$

Finally, the BCJR algorithm computes the optimum likelihood ratio (A.3) using

$$\Lambda(x_k) = \frac{\sum_{S_k \to S_{k+1}: X_k = +1} \alpha_k(S_k)\gamma_k(X_k, S_k, S_{k+1})\beta_{k+1}(S_{k+1})}{\sum_{S_k \to S_{k+1}: X_k = -1} \alpha_k(S_k)\gamma_k(X_k, S_k, S_{k+1})\beta_{k+1}(S_{k+1})} \tag{A.9}$$

VITA

Nitin Ashok Nangare received the Bachelor of Engineering (B.Engg.) degree in instrumentation and control engineering from the University of Pune, Pune, India, in 1999. From August 1999 to July 2001, he worked at NitAl computer systems, Pune, India, as a member of the technical staff. He received the Master of Science (M.Sc.) degree in communications engineering from the Technical University of Munich, Munich, Germany in 2002. In 2002, he began working toward his Ph.D. degree in electrical engineering at Texas A&M University, College Station, TX, and he graduated in 2006. His research interests include iterative decoding and signal processing, equalization and coding for magnetic recording and ISI channels, non-iterative joint decoding and signal processing, reduced complexity equalization. Nitin Nangare can be contacted at E-404, Gagan-Vihar, Pune-411037, India. His email address is nitinn@gmail.com.