

Transceiver Design with Iterative Decoding of Capacity-Approaching Codes over Fading Channels

Yuan Haifeng

Department of Electrical and Computer Engineering

National University of Singapore

A thesis submitted for the degree of

Doctor of Philosophy

Aug. 2012

I would like to dedicate this thesis to my beloved parents ...

Acknowledgements

I would like to express my sincere gratitude to my supervisor Professor Kam Pooi Yuen for his valuable guidance and kind supervision throughout the entire duration of my Ph.D course. It is he who introduced me into the exciting world of research and helped me build confidence in doing research. His profound thinking, prudential attitude and integrity will be an inspiring role model for my future career.

I would also like to express my deepest appreciation to my co-supervisor Professor Marc Andre Armand for his thoughtful inspiration and enthusiastic encouragement. I have benefited tremendously from his unique vision, technical insights and practical sensibility.

I want to give my special thanks to Dr. Chew Yong Huat, who has been providing the helpful suggestions and insightful discussions ever since my undergraduate study. I feel so fortunate to have such a mentor.

I would like to thank my senior, Wu Mingwei, who gave me a lot of help during my candidature. I am also grateful to my former and current colleagues in the communication research laboratory for their kindness, friendship, and cheerfulness. These include Wu Tong, Lin Xuzheng, Song Tianyu, Elisa Mo, Cao Le and many others.

ACKNOWLEDGEMENTS

I would like to acknowledge my parents, who always encourage and support me to achieve my goals.

Finally, I would like to thank the Lee Foundation of Singapore for the financial support in the form of a graduate scholarship.

Abstract

Low-density parity-check (LDPC) codes and turbo codes are two classes of capacity-approaching codes. LDPC codes with iterative decoding based on belief propagation (BP) have been shown to achieve an error performance only a fraction of a decibel away from the Shannon limit. In BP decoding, the reliability of each code symbol, measured by its log-likelihood ratio (LLR), is taken as the input and processed iteratively. We consider LDPC coded transmissions with M -ary phase-shift keying modulation and pilot-symbol-assisted (PSA) channel estimation over time correlated Rayleigh fading channels. The correct conceptual approach is presented for deriving the LLR expression for a general q -ary code. Its bit-error probability (BEP) performance is compared with that of the conventional metric which does not take into account the information concerning the channel estimation accuracy. Simulation results show that this LLR metric outperforms the conventional metric in both BEP performances and average number of decoding iterations required for convergence. Following similar ideas, we study turbo coded transmissions and propose generalizations of the BCJR algorithm and the soft-output Viterbi algorithm (SOVA) for turbo decoding over fading channels with PSA channel estimation. We show how the channel estimate and the estimation error variance enter in determining the *a priori* probabilities

and explain why the minimum mean-square error (MMSE) channel estimator should be used in the receiver. Both the works demonstrate the importance of incorporating the knowledge of channel estimation accuracy into the iterative decoding processes.

The knowledge of the channel statistics is crucial for the computation of the MMSE estimates and the estimation error variances. However, it might be difficult or costly to make precise measurement of the statistics at the receiver. To this end, we propose a SOVA based soft-output detector for LDPC coded transmissions over block-wise static fading channels, which is based on joint maximum-likelihood detection of data sequence and channel. This receiver does not require explicit channel estimation or knowledge of channel fading statistics. Computer simulations show that the proposed detector has substantially better BEP performance than the conventional system with PSA channel estimation.

Binary LDPC codes have been extensively studied and widely used. The extension of LDPC codes to q -ary alphabets has been shown to have better performance than binary codes. We consider, in particular, LDPC codes over integer residue rings, and propose a doubly multistage decoder (DMD) for LDPC codes over \mathbb{Z}_{2^m} , $m > 1$, which fuses the multistage decoding approaches of Armand et. al. and Varnica et. al. Two variants of the DMD are considered. The first (resp., second) performs BP (resp., offset min-sum (OMS)) decoding in each decoding stage and is referred to as DMD-BP (resp., DMD-OMS). Computer simulations show the DMD-BP (resp., DMD-OMS) achieving coding gains of up to 0.43 dB (resp., 0.67 dB) over standard BP decoding at a bit error rate of 10^{-6} on an

additive-white-Gaussian-noise channel, while requiring significantly less computational power. Remarkably, DMD-OMS outperforms DMD-BP, yet has lower computational complexity than DMD-BP. Snapshots of the LLR densities of the decoded bits midway through the decoding process explain the superiority of the DMD over standard BP decoding.

Contents

List of Tables	xii
List of Figures	xiii
List of Acronyms	xxii
List of Notations	xxiv
1 Introduction	1
1.1 Overview of Nonbinary LDPC Codes and Decoding	5
1.2 Transceiver Design and LLR Computations over Fading Channels	7
1.3 Main Contributions	12
1.3.1 Doubly Multistage Decoding	12
1.3.2 The LLR Metric for PSAM with Imperfect CSI	13
1.3.3 The LLR Computation via SOVA with Implicit CSI	14
1.3.4 Generalizations of BCJR Algorithm for Turbo Decoding over Fading Channels	16
1.3.5 Our Contributions towards Green ICT	17

1.4	Organization of the Thesis	17
2	Literature Review	19
2.1	History of Capacity-Approaching Codes	19
2.2	LDPC Codes and BP Decoding	22
2.2.1	Code Construction	22
2.2.2	LDPC Decoding	23
2.2.3	Standard BP Decoding Algorithm	24
2.2.3.1	Time Domain Implementation	24
2.2.3.2	Log/LLR Domain Implementation	26
2.3	Turbo Codes and Iterative Turbo Decoding	28
2.3.1	Turbo Encoding	29
2.3.2	Principle of Turbo Decoding	29
2.3.3	APP Decoding Algorithm	32
3	Doubly Multistage Decoding of LDPC Codes Over \mathbb{Z}_{2^m}	36
3.1	Description of DMD Algorithm	38
3.1.1	Preliminaries	38
3.1.2	Flow of the DMD	39
3.1.3	The modified BP/OMS decoder	40
3.1.4	The channel output correction phase	45
3.2	Simulation Results	47
3.3	LLR Density Analysis	54
3.4	Complexity Analysis	64

3.5	Concluding Remarks	66
4	The LLR Metric for q-ary LDPC Codes with MPSK Modulation over Rayleigh Channels with Imperfect CSI	70
4.1	System Model	72
4.2	Metric Derivation	74
4.3	Receiver Design	81
4.4	Simulation Study and Discussion	82
4.4.1	Effects of Interleaver	83
4.4.2	Effects of Pilot Symbol Spacing	84
4.4.3	Standard BP Decoding with BPSK Modulation	87
4.4.4	Standard BP Decoding under QPSK and 8PSK Modulation	90
4.4.5	Effects of SNR Estimation Error	94
4.4.6	Space Diversity with Multiple Receive Antennas	94
4.4.7	Iterative Channel Estimation and Decoding	97
4.4.8	Quaternary Codes with QPSK Modulation	104
4.5	Conclusion	104
5	The LLR Computation via SOVA with Implicit CSI	111
5.1	System Model	113
5.2	Metric Derivation	116
5.3	SOVA-ICSI	121
5.4	Iterative Channel Estimation and Decoding	123
5.5	Simulation Studies	125

5.5.1	Comparison between SOVA-ICSI and SOVA-PSAM	126
5.5.2	Effect of window length in SOVA-ICSI	127
5.5.3	Effect of SNR Mis-estimation	131
5.5.4	Iterative Channel Estimation and Decoding	131
5.6	Conclusion	137
6 Generalizations of the BCJR Algorithm for Turbo Decoding over		
Flat Rayleigh Fading Channels with Imperfect CSI		142
6.1	System Model	143
6.2	PSAM-BCJR Algorithm	147
6.3	Turbo Decoding with PSAM-BCJR/A-PSAM-BCJR algorithm	155
6.4	Receiver Structure	157
6.4.1	Standard Turbo Decoding	157
6.4.2	ICED with Hard Decision Feedback	158
6.4.3	ICED with Soft Decision Feedback	159
6.5	Simulation Study and Discussion	160
6.5.1	Standard Turbo Decoding	161
6.5.2	Iterative Channel Estimation and Decoding	162
6.5.3	Performance Comparison among Various Decoding Schemes	171
6.6	Conclusions	174
7 Summary of Contributions and Suggestions for Future Work 176		
7.1	Summary of Contributions	176
7.2	Proposals for Future Research	178

7.2.1	Implementation of DMD over Fading Channels	178
7.2.2	Channel Estimation with Soft Decision Feedback	179
7.2.3	LP Decoding for Nonbinary Codes	180
	Appendix	181
	List of Publications	183
	Bibliography	184

List of Tables

3.1	Distribution of decoded bit errors for \mathbb{Z}_8 code	57
4.1	Average number of BP iterations required for the PSAM-LLR metric and the A-PSAM-LLR metric over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.005$	90
4.2	Average number of BP iterations required for the PSAM-LLR metric and the A-PSAM-LLR metric over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.02$	91

List of Figures

2.1	Block diagram of turbo encoder	29
2.2	Block diagram of turbo decoder	33
3.1	Flowchart of the DMD	41
3.2	(a) BEP performance of DMD-OMS at different offset values for the \mathbb{Z}_4 code at the SNR of 2.6 dB; (b) BEP performance of DMD-OMS at different offset values for the \mathbb{Z}_8 code at the SNR of 2.6 dB	49
3.3	BEP performance of 1200-bit long regular \mathbb{Z}_4 code under various decoding strategies	50
3.4	BEP performance of 1200-bit long regular \mathbb{Z}_8 code under various decoding strategies	51
3.5	(a) BEP performance of 500-bit long regular \mathbb{Z}_4 code under various decoding strategies; (b) BEP performance of 500-bit long regular \mathbb{Z}_8 code under various decoding strategies	53

3.6	(a) BEP performance of 12000-bit long regular \mathbb{Z}_4 code under various decoding strategies; (b) BEP performance of 12000-bit long regular \mathbb{Z}_8 code under various decoding strategies	55
3.7	BEP performance of 2016-bit long irregular \mathbb{Z}_4 code under various decoding strategies	56
3.8	LLR density of various bit positions for the \mathbb{Z}_8 code: (a) LSB (b) 2nd LSB (c) MSB	60
3.9	LLR density of various bit positions for the \mathbb{Z}_8 code around LLR=0: (a) LSB (b) 2nd LSB (c) MSB	61
3.10	LLR density of various bit positions for the \mathbb{Z}_4 code: (a) LSB (b) MSB	62
3.11	LLR density of various bit positions for the \mathbb{Z}_4 code around LLR=0: (a) LSB (b) MSB	63
3.12	The complexity ratio associated with decoding the \mathbb{Z}_4 code	67
3.13	The complexity ratio associated with decoding the \mathbb{Z}_8 code	68
4.1	System Model	74
4.2	Effects of size of interleavers over Rayleigh fading channels with normalized fade rates $f_d T_s = 0.005$ and $f_d T_s = 0.02$ with perfect CSI	84
4.3	(a) Effects of pilot symbol spacing at normalized fade rates $f_d T_s = 0.005$ and $f_d T_s = 0.02$; (b) Equivalent SNR interpretation of the optimum pilot symbol spacing	86

4.4	Performance comparison of LLR metrics under BPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	88
4.5	Comparison of error floors with irregular codes under BPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	89
4.6	Performance comparison of LLR metrics under QPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	92
4.7	Performance comparison of LLR metrics under 8PSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	93
4.8	Robustness comparison between PSAM-LLR and A-PSAM-LLR subjected to SNR mis-estimation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	95
4.9	Effects of diversity for the PSAM-LLR over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	98
4.10	BEP performance comparison of iterative channel estimation and decoding and standard BP decoding under BPSK modulation over Rayleigh fading channels at normalized fade rate $f_d T_s = 0.02$. . .	99

4.11 Performance comparison of iterative channel estimation and decoding with different LLR metrics under BPSK modulation over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.005$	100
4.12 Performance comparison of iterative channel estimation and decoding with different LLR metrics under BPSK modulation over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.02$	101
4.13 Performance comparison of iterative channel estimation and decoding with different LLR metrics under BPSK modulation over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.05$	102
4.14 Performance comparison of iterative channel estimation and decoding with different LLR metrics under QPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	105
4.15 Performance comparison of iterative channel estimation and decoding with different LLR metrics under 8PSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	106
4.16 Performance comparison between PSAM-LLR and A-PSAM-LLR with \mathbb{GF}_4 code under QPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	107

4.17	Performance comparison between PSAM-LLR and A-PSAM-LLR with \mathbb{Z}_4 code under QPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	108
4.18	Performance comparison between PSAM-LLR and A-PSAM-LLR under QPSK modulation for binary code, \mathbb{GF}_4 code and \mathbb{Z}_4 code over Rayleigh fading channels with various normalized fade rates and pilot symbol spacings: (a) $f_d T_s = 0.005$ with pilot spacing $B = 20$; (b) $f_d T_s = 0.02$ with pilot spacing $B = 10$	109
5.1	System model of the receiver with SOVA-ICSI	114
5.2	System model of the receiver with SOVA-PSAM	114
5.3	Trellis diagrams illustrating the two situations encountered in the SOVA-ICSI	122
5.4	System models of iterative channel estimation and decoding: (a) ICED-SOVA-PSAM; (b) ICED-SOVA-ICSI.	125
5.5	BEP performance comparison of SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.001$ with various parameters: (a) $L = 10, K = 8$; (b) $L = 10, K = 200$	128
5.6	BEP performance comparison of SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.005$ with various parameters: (a) $K = 8$; (b) $K = 200$	129

5.7	Effect of the window length L in the SOVA-ICSI at the normalized fade rate of $f_d T_s = 0.001$: (a) $E_b/N_0 = 6.5$ dB, $K = 8$; (b) $E_b/N_0 = 7.5$ dB, $K = 200$	132
5.8	Effect of the window length L in the SOVA-ICSI at the normalized fade rate of $f_d T_s = 0.005$: (a) $E_b/N_0 = 8$ dB, $K = 8$; (b) $E_b/N_0 = 8.5$ dB, $K = 200$	133
5.9	Robustness comparison between SOVA-ICSI and SOVA-PSAM under SNR mis-estimation at the normalized fade rate of $f_d T_s = 0.001$: (a) $E_b/N_0 = 6.5$ dB, $L = 10$, $K = 8$; (b) $E_b/N_0 = 7.5$ dB, $L = 10$, $K = 200$	134
5.10	Robustness comparison between SOVA-ICSI and SOVA-PSAM under SNR mis-estimation at the normalized fade rate of $f_d T_s = 0.005$: (a) $E_b/N_0 = 8$ dB, $L = 7$, $K = 8$; (b) $E_b/N_0 = 8$ dB, $L = 7$, $K = 200$	135
5.11	BEP performance comparison of ICED-SOVA-ICSI, ICED-SOVA-PSAM, SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.001$ with $K = 8$	138
5.12	BEP performance comparison of ICED-SOVA-ICSI, ICED-SOVA-PSAM, SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.001$ with $K = 200$	139
5.13	BEP performance comparison of ICED-SOVA-ICSI, ICED-SOVA-PSAM, SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.005$ with $K = 8$	140

5.14	BEP performance comparison of ICED-SOVA-ICSI, ICED-SOVA-PSAM, SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.005$ with $K = 200$	141
6.1	System model of decoding turbo codes over fading channels with PSAM channel estimation	144
6.2	Block diagram of turbo encoder	144
6.3	Simplified system model	145
6.4	System model of iterative channel estimation and decoding with hard decision feedback over fading channels with PSAM channel estimation	158
6.5	System model of iterative channel estimation and decoding with soft decision feedback over fading channels with PSAM channel estimation	160
6.6	Performance comparison of standard turbo decoding between the PSAM-BCJR and A-PSAM-BCJR algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	163
6.7	Performance comparison of standard turbo decoding between the PSAM-SOVA and A-PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	164

6.8	Performance comparison of standard turbo decoding between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	165
6.9	Performance comparison of two-stage ICED with hard decision feedback between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	167
6.10	Performance comparison of two-stage ICED with soft decision feedback between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	168
6.11	Performance comparison of ICED with hard decision feedback between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	169
6.12	Performance comparison of two-stage ICED with soft decision feedback between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$	170

6.13 Performance comparison of standard turbo decoding, two-stage ICED and ICED with one internal-iteration over Rayleigh fading channels with normalized fade $f_d T_s = 0.005$ and pilot symbol spacing $B = 20$	172
6.14 Performance comparison of standard turbo decoding, two-stage ICED and ICED with one internal-iteration over Rayleigh fading channels with normalized fade $f_d T_s = 0.02$ and pilot symbol spacing $B = 10$	173

List of Acronyms

APP	<i>a posteriori</i> probability
AWGN	additive white Gaussian noise
BEP	bit-error probability
BF	bit-flipping
BP	belief propagation
CSI	channel state information
DMD	doubly multistage decoder
DPSK	differential phase-shift keying
EXIT	extrinsic information transfer
FFT	fast Fourier transform
GLRT	generalized likelihood ratio test
ICED	iterative channel estimation and decoding
ICT	information and communication technology
KCS	knowledge of channel statistics
LDPC	Low-density parity-check
LLR	log-likelihood ratio

LP	linear programming
LSB	least significant bit
MAP	maximum <i>a posteriori</i> probability
ML	maximum likelihood
MLD	maximum likelihood decoding
MLSD	maximum-likelihood sequence detector
MMSE	minimum mean-square error
MS	min-sum
MSB	most significant bit
MSDD	multiple symbol differential detection
NMS	normalized min-sum
OMS	offset min-sum
PCCC	parallel concatenated convolutional code
PSA	pilot-symbol-assisted
PSAM	pilot-symbol-assisted modulation
SCCC	serial concatenated convolutional code
SISO	soft-input soft-output
SNR	signal-to-noise ratio
SOVA	soft-output Viterbi algorithm
SPA	sum-product algorithm

List of Notations

- $(\cdot)^T$ the transpose of a vector or a matrix
- $(\cdot)^*$ the conjugate only of a scalar or a vector or a matrix
- $(\cdot)^H$ the Hermitian transpose of a vector or a matrix
- $|\cdot|$ the absolute value of a scalar
- $\|\cdot\|$ the Euclidean norm of a vector
- $\mathbb{E}[\cdot]$ the statistical expectation operator
- $\text{Re}(\cdot)$ the real part of the argument
- $\text{Im}(\cdot)$ the imaginary part of the argument
- $p(\cdot)$ the probability density function
- $P(\cdot)$ the probability mass function

Chapter 1

Introduction

In 1948, Shannon [1] demonstrated in a landmark paper that there exist codes that can achieve reliable transmission, i.e., the probability of error at the receiver can be made arbitrarily small, at any rate R less than the channel capacity C . Based on Shannon's theory, for a coded system with code rate R , a theoretical limit on the minimum signal-to-noise ratio (SNR), which is often referred to as the Shannon limit [2], is required to achieve error-free communication.

In Shannon's work, it was pointed out precisely that randomly chosen codes, along with maximum likelihood decoding (MLD), can provide capacity achieving performance. However, he gave no guidance about how to construct such good codes. Since Shannon's work, much effort has been put into constructing codes with good error-correcting capability and developing efficient decoding algorithms for these codes.

In 1993, the first capacity-approaching code – turbo code [3] was invented. It

was shown [3] that turbo codes with iterative decoding over the additive white Gaussian noise (AWGN) channel with binary phase-shift keying (BPSK) modulation can achieve a bit-error probability (BEP) of 10^{-5} at an SNR of 0.7 dB, which is within 1 dB of the Shannon limit. This exceeds the performance of all previously known codes with comparable length and decoding complexity. Following the advent of turbo codes and iterative decoding, low-density parity-check (LDPC) codes, introduced by Gallager in 1960s [4], were rediscovered and the performance of long LDPC codes was shown to be only a fraction of a decibel away from the Shannon limit over the AWGN channel [5–7]. The remarkable capacity-approaching performance of turbo and LDPC codes enable communication systems to operate in a low SNR region very close to the Shannon limit. The energy required for transmission can be significantly reduced. This offers great advantage to modern communication systems, especially for source nodes with limited power supply.

Over the last two decades, the study of turbo and LDPC codes has been extended to wireless channels. A substantial amount of research was conducted into the transceiver design of capacity-approaching codes over fading channels, aiming to achieve reliable communications at low SNRs. For transmissions over fading channels, in addition to AWGN, signals also suffer from various types of amplitude and phase distortions, which are usually characterized by the coherence time (or the Doppler spread), the coherence bandwidth (or the maximum delay spread) and the fading profile. The fading effects can severely degrade the performance of the communication system, unless measures are taken to compen-

sate for them at the receiver. To combat against the fading effect, the channel state information (CSI) is usually required at the receiver. The technique of pilot symbol assisted modulation (PSAM) is one of the most commonly adopted approaches for CSI acquisition, whereby pilot signals with deterministic information are inserted into the data signal sequence and transmitted together with data signals through the channel. In general, the CSI can be more accurately acquired when more pilot signals are used and higher energy is allocated to each of the pilot signals. However, the transceivers involving capacity-approaching codes are usually designed to operate in relatively low SNR regions. The energy allocation to the pilot signals is strictly limited, and it becomes much more difficult to acquire reliable channel estimates at these SNRs. On one hand, we want to use strong codes to save transmission energy, by reducing the required SNR to as close to the Shannon limit as possible. On the other hand, a sufficiently high SNR is required to perform reliable CSI acquisition, which is crucial for the success of the error-free decoding at the receiver. These two conflicting requirements make it a great challenge to design energy-efficient transceivers over fading channels with reliable CSI acquisition.

The strong error correcting capability of turbo and LDPC codes is mainly attributed to their random-like coding structures, as originally envisioned by Shannon in deriving the Shannon limit. However, because of the lack of structure, optimum decoding of these codes is prohibitively complex. In particular, decoding complexity increases exponentially with the length of the code (which is known as an NP-hard problem). In the ensuing years after the invention of turbo codes and

the rediscovery of LDPC codes, a large amount of research was conducted into the development of sub-optimum decoding algorithms with reasonable complexity.

For LDPC codes, the sub-optimum iterative decoding via belief propagation (BP) [8], which is commonly known as the sum-product algorithm (SPA), has been frequently used for decoding, during which, the log-likelihood ratio (LLR), representing the reliability information of the bit to be decoded, is refined iteratively when it is passed back and forth between check nodes and variable nodes. This linear-time algorithm, which was initially designed for binary codes, was shown to be effective with acceptable complexity, especially for binary LDPC decoding. In [9], it was reported that nonbinary LDPC codes can outperform their binary counterparts. For nonbinary codes, although the BP algorithm still provides reasonably good decoding performance, its complexity has increased dramatically. This is because the amount of computations required in each BP iteration increases quadratically with the size of the code alphabet, and also more iterations is usually required for the BP decoder to converge to a reliable solution for nonbinary codes than that for binary codes. When the size of the code alphabet is large, the high decoding complexity precludes the use of the standard BP decoder for real applications. Hence, designing efficient low-complexity decoders for nonbinary codes has become an imperative task for researchers.

In this chapter, we first give an overview of the iterative decoding problem for nonbinary LDPC codes in Section 1.1. A literature review on the transceiver design over fading channels will be given in Section 1.2. In particular, several channel estimation techniques and detector structures, involving the decoding of

turbo and LDPC codes, will be discussed and analyzed. In Section 1.3, we will summarize our main contributions. Finally, we present the organization of the thesis in Section 1.4.

1.1 Overview of Nonbinary LDPC Codes and Decoding

Nonbinary LDPC codes were first considered by Davey and MacKay in 1998 in [9]. In the same paper, they showed that nonbinary LDPC codes can achieve better error performance than the binary counterparts. Motivated by this promising result, nonbinary LDPC codes, especially codes over Galois fields ($\mathbb{GF}(q)$) [9–19], have been extensively studied. References [9–13] discussed the design and analysis of codes over $\mathbb{GF}(q)$. The generalization for the SPA for decoding q -ary LDPC codes was presented in [9]. The SPA based on fast Fourier transforms (FFT) was presented in [14, 15]. In order to reduce decoding complexity, the log-domain SPA was proposed in [16]. With this approach, the multiplications in the SPA are replaced with additions and subtractions, and a look-up table is used to perform the additional exponential and logarithmic computations in the log-domain SPA. In [17], the min-sum (MS) algorithm was generalized to the LLR domain for decoding nonbinary codes. An extended MS algorithm was proposed in [18] and further elaborated in [19], where only critical elements are considered at the check node processing to save computations and correction techniques are applied at the variable node processing to improve performance.

Recently, the study of nonbinary LDPC codes has been extended beyond finite field codes and includes in particular, codes over integer residue rings (\mathbb{Z}_q). See e.g. [20–25]. References [21] and [22] laid the theoretical foundation of LDPC codes over rings, by showing that the asymptotic spectra of LDPC ensembles over \mathbb{Z}_q approaches the spectrum of a random code. The structure and design of LDPC codes over integer rings were discussed in [23,24]. The decoding algorithms developed for codes over $\mathbb{GF}(q)$ can be applied, in general, to codes over \mathbb{Z}_q with some modifications. In [25], the FFT-based SPA is extended to codes over abelian groups and rings.

In [26], a multistage decoding algorithm for LDPC codes over \mathbb{Z}_{2^m} , $m > 1$, was proposed. The algorithm involves the repeated application of BP decoding to exploit the natural ring epimorphism $\mathbb{Z}_{2^m} \rightarrow \mathbb{Z}_{2^k} : r \mapsto \sum_{i=0}^{k-1} r_i 2^i$ with kernel $2^k \mathbb{Z}_{2^m}$ where $\sum_{i=0}^{m-1} r_i 2^i$ is the 2-adic expansion of r . In particular, the standard BP decoder is used to sequentially decode the canonical image of a \mathbb{Z}_{2^m} code over \mathbb{Z}_{2^k} , and the *a priori* probabilities of the code symbols are refined after every stage, based on the decoding outcome. Some coding gains can be achieved from this decoding approach over standard BP decoding on an AWGN channel. However, this comes at the expense of increased decoding complexity. One of our studies in this thesis will focus on exploiting the structures of codes over \mathbb{Z}_{2^m} to develop more efficient decoding algorithms.

1.2 Transceiver Design and LLR Computations over Fading Channels

Capacity-approaching codes can achieve reliable transmission at SNRs extremely close to the Shannon limit over the AWGN channel. During the iterative decoding process, the LLR of each received code bit is taken as the soft information input to the decoder and refined after each iteration. Therefore, using the correct LLR metric is crucial for reliable decoding.

In the literature, several LLR metrics or approximate metrics have been proposed for various channels. Gallager derived the LLR metric for the AWGN channel in [4]. In [27, 28], the LLR metric was derived based on two-symbol-interval observations for the case of differentially encoded phase shift keying over noncoherent channels and it was shown that its bit error performance is much better than that of the approximate metric proposed by Hall and Wilson in [29]. Over the same channel, reference [30] derived the LLR metric for BPSK transmission with PSAM. In [31], the authors presented a more accurate method of computing the initial LLRs for LDPC decoding over Chi-square based optical channels, which enhances the performance of optical transmissions. Reference [32] introduced a measure for the accuracy of the LLRs and studied several linear LLR approximations.

The LLR computation for transmissions over fading channel is more complex, because the transmitted signals are perturbed by an unknown complex fading gain, which will severely degrade the performance of the system. To compensate

for the fading effect, the CSI is usually acquired at the receiver. The accuracy of the acquired CSI plays an important role in determining the overall system performance, especially when the channel varies rapidly with time [33–37]. Hence, there are two main concerns for LDPC or turbo decoding over fading channels. Firstly, how can we acquire the CSI more accurately? Secondly, how should we use the acquired information to correctly compute the LLRs or the *a posteriori* probabilities?

In [38], the authors considered the joint probability density function of two consecutive received signals conditioned on each possible value of the information code bit concerned, i.e., $p(r(k), r(k-1)|\Delta\phi(k))$, and derived the correct LLR metric for the transmission with binary differential phase-shift keying (BDPSK) modulation over slow Rayleigh fading channels. Reference [39] extended the work in [38], by taking into account the information of channel autocorrelation, and showed that the error performance using the derived metric over time-correlated fading channels is better than that of the existing metrics. In differential detection, the information carried in $r(k)$ is retrieved by using $r(k-1)$ as the reference, because the signal $r(k-1)$ contains some information of the channel gain experienced by $r(k)$. In other words, one-symbol channel estimation is implicitly used in the differential encoding and detection scheme, which is not efficient in combating fading. This explains why substantial performance loss is incurred in differential detection, compared with coherent detection, where an accurate reference symbol can readily be estimated. More recent techniques, such as multiple symbol differential detection (MSDD) [40–42], can reduce this performance gap by making

observations over several consecutive differentially encoded symbols. However, the major drawback of the MSDD is its complexity, which grows exponentially with the observation symbol interval.

Alternatively, the receiver could acquire the CSI explicitly through some channel estimation process. The channel estimates can be obtained through blind channel estimation by using only unknown data signals [43, 44]. However, because of the high computational complexity and low estimation accuracy, the blind estimation technique is not commonly used. For practical applications, pilot symbol assisted (PSA) channel estimation appears to be more attractive due to its simplicity and robustness. The conventional PSAM was first introduced and studied in 1991 [45]. Different structures of PSAM detectors were proposed in [46–51] for various types of fading channels. PSAM schemes are also used for channel estimation or synchronization in advanced wireless communication systems, such as multiple-input-multiple-output systems and orthogonal-frequency-division-multiplexing systems [52–55].

In [56–65], iterative decoding of LDPC or turbo codes with PSAM channel estimation is discussed. However, the approaches adopted in [56–65] for the LLR computations are sub-optimum. In particular, in [56–62], the channel estimates are assumed to be perfect and the LLR metric based on that for the AWGN channel derived in [4] is used. The information regarding channel estimation accuracy has been neglected. In [63–65], the receiver is assumed to contain a channel estimator with a certain structure. By processing the received pilot information with the assumed channel estimator, the estimated channel gain and

the estimation error variance are obtained. The problem of this 'structured' approach is that the LLR metric varies with the estimator structure, even for the same received signal sequence.

In PSAM schemes, the initial channel estimates are obtained from only the received pilot signals. The data signals, which also contain a substantial amount of information on the CSI, are not utilized. The accuracy of channel estimation should be improved if both data signals and pilot signals are used. In fact, in the early 1980s, the idea of using data signals for channel estimation had been introduced [66–68], which was even several years earlier than the invention of the PSAM. In the proposed symbol-by-symbol detection scheme [66–68], which will be called PSAM-DF, the past received message signals and their decisions are fed back to estimate the channel gain for the current received message symbol. A stream of pilot signals is used to start up the transmission as a training sequence by providing channel estimates for the initial data signals. Meanwhile, to prevent “run way” due to a burst of decision errors, streams of pilot symbols are periodically inserted into the transmitted data sequence to refresh the memory of the receiver. Besides pilot signals, data signals are also utilized in channel estimation in the PSAM-DF scheme, which is an improvement over the conventional PSAM system. However, one drawback of the PSAM-DF receiver is that the effectiveness of the PSAM-DF relies on the accuracy of the past decisions. The channel estimates may not be reliable when the operating SNR is low and there are plenty of decision errors. Furthermore, a firm symbol decision is required immediately after the corresponding signal is received. The decision only depends on the cur-

rent and past received signals, while the additional information on the current detection contained in the future transmissions is completely neglected.

To improve on the PSAM-DF, the maximum-likelihood sequence detector (MLSD) based on trellis search was proposed in [69] and [70], which is also known as the generalized likelihood ratio test (GLRT) detector [71]. In the MLSD, decisions are made based on the joint maximum-likelihood (ML) detection of the data sequence and the channel gain, which is more robust and reliable than the symbol-by-symbol detection in the PSAM-DF. The firm symbol decision is not required during the time when the corresponding signal is received. Instead, it is decided at a later stage when sufficient information has been collected from the future received signals. Another advantage of the MLSD is that it does not require explicit channel estimation or knowledge of channel statistics (KCS). Here, KCS refers to the channel fading characteristics only, which includes, in particular, the fading model and the parameters associated with it.

For the PSAM-DF and the MLSD, hard decision output is produced, which contains only part of the information from the received signals. A significantly large amount of information is lost, which includes, in particular, the reliabilities of the hard decisions. Moreover, in the MLSD, sequence detection is carried out using path search. Since the transmitted sequence is uncoded, the minimum relative Hamming distance between contending paths during the path search is only one. We expect that the decisions would be more reliable if the relative Hamming distance between contending paths increases. These ideas motivate us to develop new detectors with more reliable soft decision output, which could be

used with iterative decoding of LDPC or turbo codes.

1.3 Main Contributions

1.3.1 Doubly Multistage Decoding

We consider LDPC codes over \mathbb{Z}_{2^m} , $m > 1$ and propose a new decoding algorithm that enables a higher coding gain over standard BP decoding to be achieved, yet with less computational burden. The new algorithm fuses the multistage decoding approach of [26] and the augmented decoding approach of [72], which is a multistage decoding approach for binary codes, as additional iterations are performed following modifications to the input LLRs of the code bits. For this reason, we refer to the proposed decoder as a *doubly multistage decoder* (DMD).

Two variants of the DMD are considered. The first performs BP decoding [8] in each decoding stage and is referred to as DMD-BP. The second performs offset min-sum (OMS) decoding in each stage and is referred to as DMD-OMS. The motivation for studying the DMD-OMS is that the OMS decoder is a good approximation to the BP decoder and can achieve small or negligible performance degradation compared to BP decoding at significantly lower computational cost [73, 74, 126, 127]. For moderate-length codes, computer simulations show the DMD-BP (resp., DMD-OMS) achieving coding gains of up to 0.43 dB (resp., 0.67 dB) over standard BP decoding at a bit error rate of 10^{-6} on an AWGN channel, while requiring significantly less computational power. Remarkably, DMD-OMS outperforms DMD-BP, yet has lower computational complexity than DMD-BP.

For short codes, even larger coding gains over standard BP decoding can be achieved. For long codes however, performance improvements are modest which is not surprising since for large codelengths, the performance of BP decoding is already close to the ML decoding performance. Thus, DMD-BP and DMD-OMS are suited for codes of short to moderate lengths.

1.3.2 The LLR Metric for PSAM with Imperfect CSI

LDPC decoding over time-selective, frequency-flat, Rayleigh fading channels is considered in this thesis. We will present the correct conceptual approach for deriving the LLR metric of a q -ary code with M -ary phase-shift keying (MPSK) modulation and PSAM channel estimation. Unlike the suboptimum approaches in [56–65], which assume either structured channel estimators or perfect channel estimations, our derivation starts from first principles without assuming any receiver structure and demonstrates how the pilot information should be incorporated into the LLR computation. In particular, we show how the channel estimate and the estimation error variance enter in determining the reliability of each received coded symbol. The derivation shows why the minimum mean-square error (MMSE) channel estimator and the estimation error variance should enter in the receiver.

The metric derived will be called the PSAM-LLR. The BEP performance, the convergence speed and the robustness will be compared between the PSAM-LLR metric and the conventional metric which does not take into account the information concerning the channel estimation accuracy, which will be called

approximate PSAM-LLR (A-PSAM-LLR) metric. Through simulations studies, we show that the PSAM-LLR has substantially better error performance and lower error floors than the A-PSAM-LLR. Furthermore, the PSAM-LLR requires, on average, fewer decoding iterations for convergence than the A-PSAM-LLR.

Our unstructured approach explains clearly why it is suboptimum to derive the metrics based on the channel estimates obtained from some predetermined estimators [56–65]. Our work demonstrates the importance of incorporating the knowledge of the channel estimation accuracy in the iterative decoding process.

1.3.3 The LLR Computation via SOVA with Implicit CSI

The computation of the PSAM-LLR metric requires perfect KCS, which includes the exact channel model and the autocorrelation function of the channel gain. However, it could be very complicated or computationally costly to obtain the KCS accurately, especially when the channel statistics varies with time. When a wrong channel model is used or the parameters that define the autocorrelation function are measured wrongly, the receiver will suffer from serious performance degradation. To build a more robust receiver with iterative decoding of LDPC codes for the cases when acquiring accurate KCS is difficult or impossible, extending the work in [70], we propose a more general soft-input soft-output sequence detection scheme using path search on the trellis of convolutional codes, which does not require KCS.

Since the MLSD is an ML detector, it can be combined perfectly with the soft-output Viterbi algorithm (SOVA) [75], which is an ML based decoder. We will

propose an algorithm based on the SOVA which converts the hard decision output of the MLSD to soft decisions. We will present how the soft information, in term of LLR, should be computed based on the MLSD with convolutional codes via SOVA over block-wise static Rayleigh fading channels with unknown channel statistics, and demonstrate that LDPC coded transmissions can be more reliably recovered using iterative decoding with the obtained LLRs. The algorithm is developed based on the GLRT, by maximizing the ML probability density function of the convolutionally encoded data sequence with respect to the channel gain, so no explicit channel estimation is required. For this reason, we call it *SOVA with implicit CSI* (SOVA-ICSI). We will show that the LLR output of the SOVA-ICSI is computed solely based on the received signal sequence, which does not require KCS. Compared with the systems with differential detection [39], [40–42] and the PSAM systems [56–65], all of which require precise KCS, the SOVA-ICSI detector is more robust and much less demanding, and thus it can be used more widely in real applications. Through computer simulations, we demonstrate that iterative decoding of LDPC codes with the SOVA-ICSI detector has substantially better BEP performance and greater robustness against SNR mis-estimations than that with the conventional SOVA and PSA channel estimation.

We emphasize that when the receiver has accurate KCS, it is crucial to consider the channel estimation accuracy in the iterative decoding process. However, when KCS is not available, the channel estimation accuracy cannot be obtained accurately. Therefore, for the SOVA-ICSI detector, the discussion of the channel estimation accuracy is not involved.

1.3.4 Generalizations of BCJR Algorithm for Turbo Decoding over Fading Channels

The BCJR algorithm [76] was invented in 1974. One of the main advantages of the algorithm is its capability of producing reliable soft-decision output. The BCJR algorithm was initially designed for soft-decision MAP decoding of convolutional codes. Following the discovery of turbo code, the BCJR algorithm has been widely considered in turbo decoding [3] and turbo equalization [77]. In recent years, turbo codes were intensively studied over fading channels [58–62, 78, 79].

We will propose a generalization of the BCJR algorithm over time-selective frequency-flat Rayleigh fading channels with PSAM channel estimation, which will be called the PSAM-BCJR. In the PSAM-BCJR, we use the correct conceptual approach to derive the exact probability density functions. Both the channel estimates and the estimation accuracy are taken into consideration in the PSAM-BCJR algorithm. An approximate BCJR algorithm, called A-PSAM-BCJR, is obtained by assuming the estimation error variance is equal to zero. In the simulations, we compare the BEP performance of the turbo decoding of the PSAM-BCJR and A-PSAM-BCJR algorithms, showing that the former achieves noticeable performance gain over the latter. The role of the channel estimation accuracy in iterative decoding will be emphasized.

1.3.5 Our Contributions towards Green ICT

In the light of the developments of global warming and its impact on our environment, the importance of green information and communication technology (ICT) has been widely recognized. Researchers have paid more attention to environmentally sustainable computing, which includes, in particular, the study and design of networking and communications systems with minimum resource usage whenever possible [80].

In this thesis, we focus on the design of transceiver structures and decoding algorithms over fading channels, aiming to reduce the energy consumption from *both* the transmitter and the receiver. Our work contributes towards green ICT by proposing transceivers that enable more reliable communication in relatively low SNR regions, and require on average significantly less computational power to recover the message. The resource utilization in our designs is more efficient.

1.4 Organization of the Thesis

The organization of the thesis is given as follows.

In Chapter 2, we will give a short review on LDPC and turbo codes and the iterative decoding algorithms.

In Chapter 3, we will introduce the doubly multistage decoding algorithm and explain its superiority from both the BEP performance and the computational complexity.

In Chapter 4, we present the derivation of the PSAM-LLR metric over time-

selective, frequency-flat, Rayleigh fading channels and highlight the importance of incorporating the knowledge of the channel estimation accuracy.

In Chapter 5, we consider block-wise static Rayleigh fading channels, and develop the SOVA-ICSI for the computation of the LLRs with implicit CSI. The advantages of the SOVA-ICSI will be demonstrated through simulations.

In Chapter 6, generalizations of the BCJR algorithm over time-selective, frequency-flat, Rayleigh fading channels with PSAM channel estimation are derived. The performance of turbo decoding with the derived algorithms will be investigated.

Finally, we summarize our contributions and make suggestions for future research in Chapter 7.

Chapter 2

Literature Review

In this chapter, we will first introduce the evolution of capacity-approaching codes and give a brief history of turbo and LDPC codes. The iterative decoding algorithm via BP will be reviewed for LDPC codes and the main shortcoming of BP decoding for nonbinary LDPC codes will be pointed out. Finally, we will recapitulate the principle of turbo decoding and discuss the differences between the BCJR algorithm and the SOVA. In this chapter, we will only consider BPSK modulation over the AWGN channel.

2.1 History of Capacity-Approaching Codes

In 1948, Shannon published his paper “A Mathematical Theory of Communication” in the Bell Systems Technical Journal [1]. In this work, Shannon pointed out that every communication channel has a speed limit, measured in binary digits per second, which is known as the channel capacity. Although the fun-

damental limits on communication efficiency is stated clearly and the existence of “good” codes is assured, Shannon’s methodology provides no insight on code constructions to actually achieve these limits, since his derivation is based on the sophisticated average performance of a randomly chosen ensemble of codes which eliminate all detailed system structures.

Following Shannon’s ground-breaking work, various error correcting codes were designed, which mainly include linear block codes, such as Hamming codes (1950) [81], Reed-Muller codes (1954) [82, 83], BCH codes (1959) [84, 85] and Reed-Solomon codes (1960) [86], and convolutional codes [87].

For both block codes and convolutional codes, the code design involves a large amount of structures, either algebraic or topological. These code structures guarantee that the codes have good minimum distance and require simple decoding algorithms. However, the random-like properties, as originally envisioned by Shannon in deriving the Shannon limit, have been ignored. This is why these codes usually fall far short of achieving the performance promised by Shannon.

Motivated by the ideas of Batill and Hagenauer on random-like coding design [88–90], Berrou, Glavieux and Thitimajshima successfully designed a random-like code with just enough structure for decoding. This code was named turbo code. The concept of turbo codes was first introduced during the International Conference on Communications in 1993 [3] and was further elaborated upon in 1996 [91], and in 1998 [92], respectively. The first version of turbo codes was the parallel concatenated convolutional codes (PCCCs). It was shown in [3] that PCCCs with iterative decoding via the BCJR algorithm over the AWGN

2. LITERATURE REVIEW

channel with BPSK modulation can achieve a BEP of 10^{-5} at an SNR of 0.7 dB, which is within 1 dB of the Shannon limit. Besides PCCCs, several variations of turbo codes have also been proposed in the literature, such as serial concatenated convolutional codes (SCCCs) [93–96], and hybrid parallel and serial concatenated turbo codes [97–99]. References [100–106] give theoretical justifications of the superiority of turbo codes and provide insights into iterative turbo decoding.

Following the advent of turbo codes and iterative decoding, another type of capacity-approaching code - the LDPC code, was rediscovered through the work of MacKay and Neal [2,8]. In fact, LDPC codes were first discovered by Gallager in his doctoral dissertation [4] in the 1960s. However, for the next several decades, this remarkable discovery was largely forgotten primarily because computers of the time could not simulate the performance of these codes with meaningful block lengths at low error rates. During this period, Tanner noted Gallager's work and he generalized LDPC codes in 1981 [107], by introducing a graphical representation of LDPC codes, called Tanner graphs. After the invention of turbo codes, it was proved, through computer simulations, that long LDPC codes with iterative decoding based on BP [8] can achieve an error performance only a fraction of a decibel away from the Shannon limit [5–7] and soon after, LDPC codes became a hot research topic.

In this chapter, we will briefly review important concepts regarding turbo and LDPC codes, and the standard decoding techniques.

2.2 LDPC Codes and BP Decoding

2.2.1 Code Construction

An LDPC code is simply a linear block code with a parity-check matrix that is very sparse, i.e., it only contains a small number of nonzero entries. Gallager [4] proposed constructing LDPC codes by randomly selecting the positions of the nonzero entries in the parity check matrix such that the row weights are the same and the column weights are also the same. Codes of this form are referred to as *regular* LDPC codes. On the other hand, if the row/column weight distribution is not uniform, the LDPC code will be called *irregular*.

However, this random code design from Gallager did not guarantee good BEP performance. Early work on LDPC code design can be found in [108, 109]. In recent years, based on different asymptotic analysis tools such as density evolution [5–7, 110] and extrinsic information transfer (EXIT) charts [111–113], numerous LDPC codes with good error performance were designed, such as repeat-accumulate codes [114–116] and quasi-cyclic codes [117–119]. Efficient code construction techniques, such as the bit-filling algorithm [120] and the progressive edge-growth construction [121, 122], were also invented. Through years of experiments, researchers have found some binary LDPC codes with exceptionally good performance, which can be retrieved online from the website maintained by David MacKay [123].

2.2.2 LDPC Decoding

LDPC codes can be decoded in various ways. Iterative decoding based on BP is most commonly used for LDPC decoding. There are other types of decoding algorithms. The majority-logic algorithm [159], the bit-flipping (BF) algorithm [124, 159] and the weighted BF algorithm [124, 159] are computationally simple, but they have relatively poor error performances. The MS algorithm [125] and its variants, such as the offset min-sum (OMS) algorithm and the normalized min-sum (NMS) algorithm [126, 127], are approximations of BP decoding. Some trade-off between error performance and decoding complexity can be achieved using these algorithms. Recently, Feldman et al. [128] proposed linear programming (LP) decoding for LDPC codes. This LP decoder has the desirable ML certificate property, i.e., its failure to find an ML codeword is always detectable. However, the much higher complexity compared to standard BP decoding precludes its use in practice.

In the literature, it has become a common practice to compare the decoding performance of a newly designed decoding algorithm with that of the BP decoder. Hence, the BP decoder has been widely regarded as the “standard” decoder for LDPC codes, and used as a benchmark of measuring decoding performance. For this reason, we will refer to it as the *standard BP decoding* algorithm. A brief review on standard BP decoding is given in the following section.

2.2.3 Standard BP Decoding Algorithm

2.2.3.1 Time Domain Implementation

Using the Tanner graph representation, each LDPC code can be represented by a bipartite graph, which has two disjoint subsets of nodes, called variable nodes and check nodes. There is one variable node for each coded symbol, and one check node corresponding to each parity-check constraint.

Given a q -ary LDPC codes with an $m \times n$ parity-check matrix $H = [h_{ij}]$, there are n variable nodes and m check nodes. Denote the code alphabet by Ω_q and a codeword by $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, where each $x_j \in \Omega_q$. The input to the BP decoder is the *a priori* probability vectors for all coded symbols. Let the *a priori* probability vector of the j -th coded symbol be $\mathbf{p}_j = [p_j^0, p_j^1, \dots, p_j^{q-1}]^T$, where p_j^b is the probability that the j -th coded symbol x_j is equal to b .

Each iteration of standard BP decoding involves two types of operations, row operations and column operations. Let $M(j)$ be the set of check nodes connected to the j th variable node and $L(i)$ the set of variable nodes connected to the i th check node. Denote the message passed from the j th variable node to the i th check node by $\mathbf{q}_{ji} = [q_{ji}^0, q_{ji}^1, \dots, q_{ji}^{q-1}]^T$, where q_{ji}^b is the probability that x_j is equal to b , given the information obtained via the i' check node for all $i' \in M(j) \setminus \{i\}$. Similarly, the message passed from the i th check node to the j th variable node is denoted by $\mathbf{r}_{ji} = [r_{ji}^0, r_{ji}^1, \dots, r_{ji}^{q-1}]^T$, where r_{ji}^b is the probability that the i th parity-check constraint

$$\sum_{j' \in L(i) \setminus \{j\}} h_{ij'} x_{j'} = -h_{ij} x_j \quad (2.1)$$

is satisfied when $x_j = b$, given that the other variable nodes are independent with probabilities given by the elements of $\mathbf{q}_{j'i}$ for all $j' \in L(i) \setminus \{j\}$.

The standard BP decoding algorithm for a general q -ary LDPC code is given as follows:

Step 1: the values of the \mathbf{q}_{ji} are initialized by \mathbf{p}_j , i.e., $\mathbf{q}_{ji} = \mathbf{p}_j$ for all $i \in M(j)$.

Step 2: For each $1 \leq i \leq m$ and $j \in L(i)$, update \mathbf{r}_{ij} element-wise as

$$r_{ji}^b = \sum_{\mathbf{x} \in \Omega_q^n: x_j = b} P((2.1) \text{ is satisfied} | \mathbf{x}) \prod_{j' \in L(i) \setminus \{j\}} q_{j'i}^{x_{j'}},$$

for all $b \in \Omega_q$.

Step 3: For each $1 \leq j \leq n$ and $i \in M(j)$, update \mathbf{q}_{ij} element-wise as

$$q_{ji}^b = \alpha p_j^b \prod_{i' \in M(j) \setminus \{i\}} r_{ji'}^b$$

for all $b \in \Omega_q$, where α is a suitable normalization factor such that $\sum_{b \in \Omega_q} q_{ji}^b = 1$.

Step 4: Find the codeword estimate $\hat{\mathbf{x}}$ with each entry \hat{x}_j given by $\hat{x}_j := \arg \max_{b \in \Omega_q} \{q_j^b\}$ where

$$q_j^b = \alpha p_j^b \prod_{i' \in M(j)} r_{ji'}^b,$$

and α is chosen to ensure that $\sum_{b \in \Omega_q} q_j^b = 1$.

Step 5: If $\mathbf{H}\hat{\mathbf{x}} = \mathbf{0}$, then output $\hat{\mathbf{x}}$ and exit. Otherwise, repeat Step 2 to Step 4, until a codeword estimate is found satisfying $\mathbf{H}\hat{\mathbf{x}} = \mathbf{0}$ or a predefined number of iterations is reached, whichever is earlier.

2.2.3.2 Log/LLR Domain Implementation

The standard BP decoding algorithm can also be implemented in the log-domain or LLR-domain. For the log-domain standard BP decoder, the inputs to the decoder become the *a priori* LLR vectors, defined by $\Lambda_j = [\lambda_j^0, \lambda_j^1, \dots, \lambda_j^{q-1}]^T$, where $\lambda_j^b = \log p_j^0/p_j^b$. Note that the first entry of Λ_j is always equal to zero and it is introduced only for ease of presentation. The messages passed between variable nodes and check nodes, \mathbf{r}_{ji} and \mathbf{q}_{ji} , take the same forms as those for the time domain implementation, but they are interpreted as LLR vectors now. With these notations, the log-domain standard BP decoding algorithm is given as follows:

Step 1: the values of the \mathbf{q}_{ji} are initialized by Λ_j , for all $i \in M(j)$.

Step 2: For each $1 \leq i \leq m$ and $j \in L(i)$, update \mathbf{r}_{ij} element-wise as

$$\begin{aligned} r_{ji}^b &= \bigoplus_{\mathbf{x} \in \Omega_q^n : x_j = b} P((2.1) \text{ is satisfied} | \mathbf{x}) \sum_{j' \in L(i) \setminus \{j\}} q_{j'i}^{x_{j'}} \\ &- \bigoplus_{\mathbf{x} \in \Omega_q^n : x_j = 0} P((2.1) \text{ is satisfied} | \mathbf{x}) \sum_{j' \in L(i) \setminus \{j\}} q_{j'i}^{x_{j'}} \end{aligned}$$

for all $b \in \Omega_q$, where the operator \oplus is defined as

$$a \oplus b := -\log(\exp(-a) + \exp(-b)) = \min(a, b) - \log(1 + \exp(-|a - b|))$$

Step 3: For each $1 \leq j \leq n$ and $i \in M(j)$, update \mathbf{q}_{ij} as

$$\mathbf{q}_{ij} = \Lambda_j + \sum_{i' \in M(j) \setminus \{i\}} \mathbf{r}_{ji'}$$

Step 4: Find the codeword estimate $\hat{\mathbf{x}}$ with \hat{x}_j given by $\hat{x}_j := \arg \max_{b \in \Omega_q} \{q_j^b\}$

where

$$q_j^b = \lambda_j^b \sum_{i' \in M(j)} r_{ji'}^b,$$

Step 5: If $\mathbf{H}\hat{\mathbf{x}} = \mathbf{0}$, then output $\hat{\mathbf{x}}$ and exit. Otherwise, repeat Step 2 to Step 4, until a codeword estimate is found satisfying $\mathbf{H}\hat{\mathbf{x}} = \mathbf{0}$ or a predefined number of iterations is reached, whichever is earlier.

- *Remark*

1. The computational load of the time-domain BP algorithm mainly comes from the multiplication operations, which is usually much more costly than additions. In the log-domain BP algorithm, multiplications are replaced by additions, but evaluation of exponential and logarithm functions are required, which can be computed to any arbitrary accuracy using a truncated Taylor series expansion or roughly estimated using a look-up table.

For binary LDPC codes, the LLR inputs Λ_j and message variables \mathbf{r}_{ji} and \mathbf{q}_{ji} can be taken as scalars. Thus, the second step of the log-domain BP algorithm can be simplified further as:

Step 2*: For each $1 \leq i \leq m$ and $j \in L(i)$, update \mathbf{r}_{ij} as

$$r_{ji} = 2 \tanh^{-1} \left(\prod_{j' \in L(i) \setminus \{j\}} \tanh \frac{q_{j'i}}{2} \right)$$

The computation of \mathbf{r}_{ij} can be done without examining the parity-check constraints and the complexity of the log-domain standard BP decoder is further reduced. In general, the log-domain algorithm is used more frequently than the time-domain algorithm, especially for decoding binary LDPC codes.

2. For both the time-domain and log-domain BP algorithms, the computa-

tional complexity is linearly proportional to the size of the parity check matrix H , and is quadratic with respect to the size of the code alphabet. When LDPC codes with large code alphabet are used, the standard BP algorithm is not computationally efficient. In Chapter 3, we will consider a specific class of nonbinary codes, and propose a decoding algorithm that can achieve better error performance, while requiring significantly less computational power, than the standard BP algorithm.

3. The standard BP decoder only requires the *a priori* LLRs (or equivalently the *a priori* probabilities) as input. The reliability of the decoding outcome is highly dependent on the accuracy of the LLRs. Therefore, how to compute the LLRs from the channel output is crucial for reliable decoding. For transmission over the AWGN channel, the exact LLR metric has been derived in [4]. In Chapters 4 and 5, we will discuss how to compute the LLRs more reliably for fading channels.

2.3 Turbo Codes and Iterative Turbo Decoding

In this section, we will present the most fundamental form of the turbo code, i.e., the PCCCs. The turbo decoding principle will be reviewed with theoretical justifications and discussions. Finally, the BCJR algorithm and the SOVA will be introduced briefly.

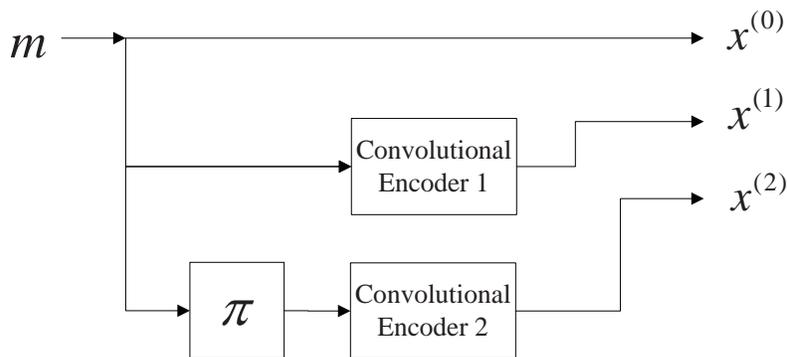


Figure 2.1: Block diagram of turbo encoder

2.3.1 Turbo Encoding

An encoder for a classical turbo code formed by PCCCs is shown in Fig. 2.1. The two convolutional encoders are usually identical and both of them are chosen to be recursive systematic codes of rate half. An interleaver is used to permute the input bits such that the two encoders operate on the same block of input message bits, but in a different order. Since both encoders are systematic, it is only necessary to transmit the input bits once, and thus the overall code has rate $R = 1/3$. (Note that the code rate can be increased to $1/2$ through puncturing by alternately deleting the encoded bits from the nonsystematic branches of the two convolutional encoders. In this thesis, we will not consider this feature.)

2.3.2 Principle of Turbo Decoding

The message sequence is represented as $\mathbf{m} = [m(1) m(2) \dots m(K)]$. Let the encoded sequences from the three output branches be $\mathbf{x}^{(0)}$, $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, as shown

in Fig. 2.1. Obviously, $\mathbf{x}^{(0)} = \mathbf{m}$. Assume that the encoded sequence is modulated using BPSK and transmitted through an AWGN channel.

The principle of iterative turbo decoding at the receiver is developed from the maximum *a posteriori* probability (MAP) decision rule, which would select

$$\hat{m}(k) = \max_{t \in \{0,1\}} P(m(k) = t | \mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}),$$

where $\mathbf{r}^{(0)}$ is the received systematic bit sequence, corresponding to $\mathbf{x}^{(0)}$, and $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ are the received parity bit sequences corresponding to $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, respectively.

From Bayes' rule, the formula can be developed as

$$\begin{aligned} & \max_{t \in \{0,1\}} P(m(k) = t | \mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}) \\ &= \max_{t \in \{0,1\}} p(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)} | m(k) = t) P(m(k) = t) \\ &= \max_{t \in \{0,1\}} \sum_{\mathbf{m}: m(k)=t} p(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)} | \mathbf{m}) P(\mathbf{m}) \end{aligned}$$

Conditioned on the message sequence \mathbf{m} , the received signals $\mathbf{r}^{(0)}$, $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$ are statistically independent, i.e.,

$$p(\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)} | \mathbf{m}) = p(\mathbf{r}^{(2)} | \mathbf{m}) p(\mathbf{r}^{(0)}, \mathbf{r}^{(1)} | \mathbf{m}) \quad (2.2)$$

Using (2.2) and applying Bayes' rule again, we obtain

$$\begin{aligned} & \max_{t \in \{0,1\}} P(m(k) = t | \mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}) \\ &= \max_{t \in \{0,1\}} \sum_{\mathbf{m}: m(k)=t} p(\mathbf{r}^{(2)} | \mathbf{m}) p(\mathbf{r}^{(0)}, \mathbf{r}^{(1)} | \mathbf{m}) P(\mathbf{m}) \\ &= \max_{t \in \{0,1\}} \sum_{\mathbf{m}: m(k)=t} p(\mathbf{r}^{(2)} | \mathbf{m}) P(\mathbf{m} | \mathbf{r}^{(0)}, \mathbf{r}^{(1)}) \end{aligned} \quad (2.3)$$

The exact evaluation of (2.3) leads to the MAP decision. However, it is far too complex. In the iterative turbo decoding algorithm, the distribution separation is assumed, i.e.,

$$P(\mathbf{m}|\mathbf{r}^{(0)}, \mathbf{r}^{(1)}) \approx \prod_{l=1}^K P(m(l)|\mathbf{r}^{(0)}, \mathbf{r}^{(1)}) \quad (2.4)$$

The term $P(m(l)|\mathbf{r}^{(0)}, \mathbf{r}^{(1)})$ represents the soft information produced by the first decoder which has access only to $\mathbf{r}^{(0)}$ and $\mathbf{r}^{(1)}$. It can be interpreted as the *a priori* information to the second convolutional decoder. Therefore, we have

$$\begin{aligned} & \max_{t=\{0,1\}} P(m(k) = t|\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)}) \\ & \approx \max_{t=\{0,1\}} \sum_{\mathbf{m}:m(k)=t} p(\mathbf{m}, \mathbf{r}^{(2)}) \\ & = \max_{t=\{0,1\}} P(m(k) = t|\mathbf{r}^{(2)}) \end{aligned} \quad (2.5)$$

Note that the *a posteriori* probability (APP) decoding algorithm can be used to compute the probability $P(m(k) = t|\mathbf{r}^{(2)})$, with the *a priori* probability given by $P(m(k)|\mathbf{r}^{(0)}, \mathbf{r}^{(1)})$.

Now we consider the LLR approach. Define the *a posteriori* LLR as

$$\lambda(m(k)) = \log \frac{P(m(k) = 0|\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)})}{P(m(k) = 1|\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \mathbf{r}^{(2)})}$$

Using the soft-output APP decoding algorithm on the second decoder, we have

$$\lambda(m(k)) \approx \lambda_{2,e}(m(k)) + \log \frac{P(m(k) = 0|\mathbf{r}^{(0)}, \mathbf{r}^{(1)})}{P(m(k) = 1|\mathbf{r}^{(0)}, \mathbf{r}^{(1)})}$$

where $\lambda_{2,e}(m(k))$ is the extrinsic information contributed by the second decoder.

Applying the same argument to the *a posteriori* LLR

$$\log \frac{P(m(k) = 0 | \mathbf{r}^{(0)}, \mathbf{r}^{(1)})}{P(m(k) = 1 | \mathbf{r}^{(0)}, \mathbf{r}^{(1)})},$$

we obtain

$$\lambda(m(k)) \approx \lambda_{2,e}(m(k)) + \lambda_{1,e}(m(k)) + \lambda_0(m(k)), \quad (2.6)$$

where $\lambda_{1,e}(m(k))$ is the extrinsic information contributed by the first decoder and

$$\lambda_0(m(k)) = \log \frac{P(m(k) = 0 | \mathbf{r}^{(0)})}{P(m(k) = 1 | \mathbf{r}^{(0)})}$$

is the *a posteriori* LLR of the systematic bits.

Equation (2.6) describes the key principle of iterative turbo decoding. The block diagram of the iterative turbo decoder is shown in Fig. 2.2, where the soft-output APP decoders correspond to the constituent convolutional encoders in Fig. 2.1, and the interleaver is the same as that used in the turbo encoder. Note that to avoid the accumulation of repeated ‘old’ information, only the extrinsic information should be exchanged between the APP decoders.

2.3.3 APP Decoding Algorithm

The APP algorithm for convolutional decoding with unequal *a priori* probabilities of the information bits was originally invented by Bahl, Cocke, Jelinek, and Raviv [76] in 1972, which is named BCJR algorithm. Since the BCJR algorithm only leads to minor improvements in term of BEP, compared to the Viterbi algorithm,

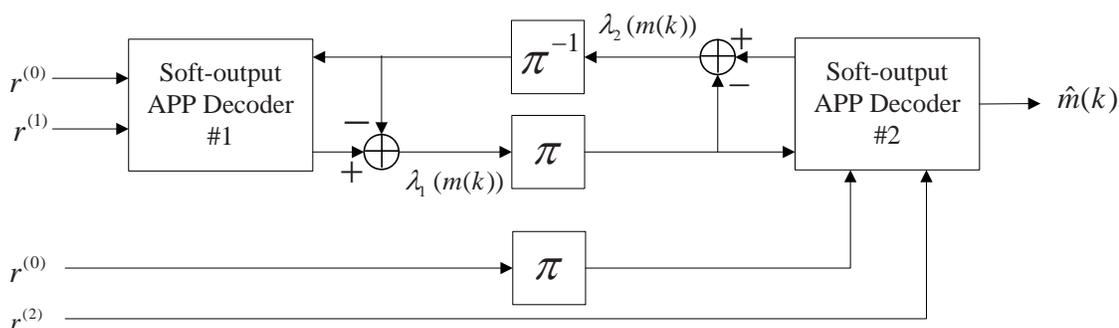


Figure 2.2: Block diagram of turbo decoder

while requiring much higher computational power, it was not widely used for decoding of convolution codes. With the invention of turbo codes in 1993, the BCJR algorithm became the major representative of the soft-input soft-output (SISO) algorithms that can be used for turbo decoding.

For a convolutional code, the LLR can be expressed as

$$\begin{aligned} \lambda(m(k)) &= \log \frac{P(m(k) = 0 | \mathbf{r})}{P(m(k) = 1 | \mathbf{r})} \\ &= \ln \frac{\sum_{(s, s') \in \Sigma_k^0} p(s(k) = s', s(k+1) = s, \mathbf{r})}{\sum_{(s, s') \in \Sigma_k^1} p(s(k) = s', s(k+1) = s, \mathbf{r})}, \end{aligned}$$

where \mathbf{r} is the received sequence observed at the channel output, and Σ_k^0 (resp., Σ_k^1) is the set of all state pairs (s', s) such that the state transition from s' to s corresponds to the k th input message bit being the 0-bit (resp., 1-bit).

In the BCJR algorithm, the probability density function $p(s(k) = s', s(k+1) = s, \mathbf{r})$ is expressed the products of three density functions as

$$p(s', s, \mathbf{r}) = \beta_{k+1}(s) \gamma_k(s', s) \alpha_k(s')$$

The values of $\gamma_k(s', s)$ can be easily computed for BPSK transmissions over the

AWGN channel. The values of $\alpha_k(s')$ are computed by the forward recursion as

$$\alpha_{k+1}(s) = \sum_{s' \in \Omega_k} \gamma_k(s', s) \alpha_k(s')$$

The values of the $\beta_{k+1}(s)$ can be calculated by an analogous procedure, called the backward recursion, as

$$\beta_k(s') = \sum_{s \in \Omega_k} \gamma_k(s', s) \beta_{k+1}(s)$$

where Ω_k is the set of all possible states at time k .

Unlike the Viterbi algorithm, the BCJR algorithm needs to go through the trellis twice, once in the forward direction and once in the backward direction. Moreover, the values of $\alpha_k(s')$ or $\beta_{k+1}(s)$ must be stored and the storage grows exponentially with the constraint length of the convolutional code and linearly with the length of the code. Due to these limitations, another type of SISO algorithm, i.e., the SOVA, is used when there are strict requirements on computational complexity or memory space.

The SOVA was introduced by Hagenauer and Hoeher [75] in 1989. It is an extension of the Viterbi algorithm [129]. The basic operation of the SOVA is identical to that of the Viterbi algorithm. The only difference is that a reliability indicator is attached to the hard-decision output for each bit decision in the SOVA. The combination of the hard-decision output and the reliability indicator forms a estimate of the *a posteriori* LLR. The SOVA is computationally simpler than the BCJR algorithm, but its performance with iterative turbo decoding is not as good. More details of the BCJR algorithm and the SOVA can be found in [159].

There are other SISO algorithms available, such as the Max-log-MAP algorithm [130], but they are not very commonly used. In the thesis, we focus on the BCJR algorithm and the SOVA for turbo decoding. In Chapter 5, we will consider the LLR computation based on the SOVA with implicit CSI over block-wise static Rayleigh fading channels, where iterative channel estimation and decoding will be investigated for a turbo-like coding structure. In Chapter 6, we will propose generalizations of the BCJR algorithm and the SOVA for transmissions over time-selective Rayleigh flat fading channels with PSAM channel estimation, and study their performance in turbo decoding.

Chapter 3

Doubly Multistage Decoding of LDPC Codes Over \mathbb{Z}_{2^m}

It is now well known that nonbinary LDPC codes can outperform their binary counterparts, a fact first reported in [9]. The study of nonbinary LDPC codes extends beyond finite field codes and includes in particular, codes over integer residue rings. See e.g. [20–25]. In [26], a multistage decoding algorithm for LDPC codes over \mathbb{Z}_{2^m} , $m > 1$, was proposed. This algorithm repeatedly invokes the standard BP decoder to sequentially decode the canonical image of a \mathbb{Z}_{2^m} code over \mathbb{Z}_{2^k} , first for $k = 1$, then $k = 2$, and so on – thus exploiting the natural ring epimorphism $\mathbb{Z}_{2^m} \rightarrow \mathbb{Z}_{2^k} : r \mapsto \sum_{i=0}^{k-1} r_i 2^i$ with kernel $2^k \mathbb{Z}_{2^m}$ where $\sum_{i=0}^{m-1} r_i 2^i$ is the 2-adic expansion of r . Unfortunately, the coding gains that this decoding approach achieves over standard BP decoding on an AWGN channel are modest and in fact outweighed by the increase in computational burden.

We present a new decoding algorithm for \mathbb{Z}_{2^m} codes that enables a higher coding gain over standard BP decoding to be achieved, yet with less computational burden. The new algorithm fuses the multistage decoding approach of [26] and the augmented decoding approach of [72], which is a multistage decoding approach for binary codes, as additional iterations are performed following modifications to the input LLRs of the code bits. For this reason, we refer to the proposed decoder as a *doubly multistage decoder*. Two variants of the DMD are considered. The first performs BP decoding [8] in each decoding stage and is referred to as DMD-BP. The second performs OMS decoding in each stage and is referred to as DMD-OMS. The motivation for studying the DMD-OMS is that the OMS decoder is a good approximation to the BP decoder and can achieve small or negligible performance degradation compared to BP decoding at significantly lower computational cost [73, 126, 127]. For moderate-length codes, computer simulations show the DMD-BP (resp., DMD-OMS) achieving coding gains of up to 0.43 dB (resp., 0.67 dB) over standard BP decoding at a bit error rate of 10^{-6} on an AWGN channel, while requiring significantly less computational power. Remarkably, DMD-OMS outperforms DMD-BP, yet has lower computational complexity than DMD-BP. For short codes, even larger coding gains over standard BP decoding can be achieved. For long codes however, performance improvements are modest which is not surprising since for large codelengths, the performance of BP decoding is already close to ML decoding performance. Thus, DMD-BP and DMD-OMS are suited for codes of short to moderate lengths. A description of the DMD follows.

3.1 Description of DMD Algorithm

3.1.1 Preliminaries

Let \mathcal{C}_m be a length- N linear code over \mathbb{Z}_{2^m} with parity-check matrix $\mathbf{H}_m := [h_{ij}^m]$. Further, let the canonical image of \mathcal{C}_m over \mathbb{Z}_{2^k} , $1 \leq k \leq m$, be denoted by \mathcal{C}_k and its parity-check matrix by $\mathbf{H}_k := [h_{ij}^k]$ which is equal to $\mathbf{H}_m \bmod 2^k$. In [24], it is shown that the presence of zero divisors amongst the nonzero entries of the parity-check matrix of a \mathbb{Z}_{2^m} code may inhibit the convergence of the decoder. Moreover, the *optimal* set of edge weights (i.e., the edge weights that result in convergence at the lowest channel SNR) of the Tanner graph representation of the codes considered there, do not contain zero divisors. For these reasons, we will take all the nonzero entries in \mathbf{H}_m to be units of \mathbb{Z}_{2^m} . Consequently, the sets $\{j : h_{ij}^k \neq 0\}$ (respectively, $\{i : h_{ij}^k \neq 0\}$) are identical for all k and so we denote them by $L(i)$ (respectively, $M(j)$). Moreover, by [131, Cor. 4.7(i)], given a code over \mathbb{Z}_{2^m} that is not *free*, there exists a free code over \mathbb{Z}_{2^m} of the same length and minimum distance but of higher rate. (A free code over \mathbb{Z}_{2^m} is a \mathbb{Z}_{2^m} -submodule with a basis.) Therefore, we further assume that \mathcal{C}_m is free.

For a codeword $\mathbf{x} := (x_1, \dots, x_N) \in \mathbb{Z}_{2^m}^N$ of \mathcal{C}_m , let

$$\mathbf{x}_{\text{bin}} := (x_{1,0}, \dots, x_{1,m-1}, \dots, x_{N,0}, \dots, x_{N,m-1}) \in \mathbb{Z}_2^{mN}$$

be its natural binary representation, i.e., $\sum_{k=0}^{m-1} x_{j,k} 2^k$ is the 2-adic expansion of x_j . Thus, $x_{j,0}$ is the (first) least significant bit (LSB) of x_j , $x_{j,1}$ is the second LSB, and so on. In particular, the m th LSB $x_{j,m-1}$ is also the most

significant bit (MSB) of x_j . Suppose \mathbf{x}_{bin} is transmitted through a binary-input-continuous-output AWGN channel using binary signaling and received as $\mathbf{y} := (y_{1,0}, \dots, y_{1,m-1}, \dots, y_{N,0}, \dots, y_{N,m-1}) \in \mathfrak{R}^{mN}$. From \mathbf{y} , one obtains for each $x_{j,k}$, the pair of *a priori* probabilities $p_{x_{j,k}}^0 := P\{x_{j,k} = 0|y_{j,k}\}$ and $p_{x_{j,k}}^1 := P\{x_{j,k} = 1|y_{j,k}\}$, and the *a priori* LLR of $x_{j,k}$, i.e., $\Lambda_{x_{j,k}} = \log \frac{p_{x_{j,k}}^0}{p_{x_{j,k}}^1}$.

3.1.2 Flow of the DMD

Unlike the multistage decoder in [26] which attempts to recover \mathbf{x} from \mathbf{y} by decoding the canonical images of \mathcal{C}_m sequentially over m stages starting with \mathcal{C}_1 , the DMD does so by decoding a coset of \mathcal{C}_1 in \mathbb{Z}_2^N in each stage starting with \mathcal{C}_1 itself – a strategy similar in spirit to the multistage *algebraic* decoding approach of [132] for free, linear \mathbb{Z}_2^m codes. This key difference between the DMD and the multistage decoder in [26] is necessary for an LLR modification strategy similar to that in [72], to be incorporated.

In the k th decoding stage, the DMD computes a codeword estimate $\hat{\mathbf{x}}_k$ of \mathcal{C}_k which, in its natural binary representation, has the form

$$(\hat{x}_{1,0}, \dots, \hat{x}_{1,k-2}, \hat{x}_{1,k-1}, \dots, \hat{x}_{N,0}, \dots, \hat{x}_{N,k-2}, \hat{x}_{N,k-1}).$$

If the codeword estimate $\hat{\mathbf{x}}_{k-1}$ found at the end of the $(k-1)$ th decoding stage is not a codeword of \mathcal{C}_{k-1} , then in the k th stage, a standard BP decoder \mathcal{SD}_k for \mathcal{C}_k is used to obtain $\hat{\mathbf{x}}_k$, before advancing to the $(k+1)$ th decoding stage. Thus, the k th decoding stage does not make use of the decoding decision made in the preceding stage in this case.

If, on the other hand, $\hat{\mathbf{x}}_{k-1}$ is a codeword of \mathcal{C}_{k-1} , then in the k th stage, a modified BP or OMS decoder \mathcal{MD}_k is used to compute $\hat{\mathbf{x}}_k$ by leveraging on $\hat{\mathbf{x}}_{k-1}$. For DMD-BP, \mathcal{MD}_k is a modified BP decoder while for DMD-OMS, \mathcal{MD}_k is a modified OMS decoder. If $\hat{\mathbf{x}}_k$ is a codeword of \mathcal{C}_k , we advance to the $(k+1)$ th decoding stage, otherwise, we proceed to a channel output correction phase which involves, in part, further iterations of \mathcal{MD}_k in the hope of eventually obtaining a valid codeword of \mathcal{C}_k . We refer to the concatenation of \mathcal{MD}_k and the channel output correction phase as the *augmented modified BP/OMS decoder* for \mathcal{C}_k , or \mathcal{AMD}_k in short. Regardless of whether a valid codeword is eventually obtained when the channel output correction phase terminates, we will subsequently proceed to the $(k+1)$ th decoding stage.

Obviously, if $k = m$, there is no $(k+1)$ th stage to proceed to. Fig. 3.1 summarizes the flow of our doubly multistage decoding approach. Note that for $k = 1$, \mathcal{MD}_k and \mathcal{SD}_k are actually identical. (This will be apparent from the description of \mathcal{MD}_k below.) For this reason, Fig. 3.1 shows the first decoding stage containing only \mathcal{AMD}_k while subsequent stages contain both \mathcal{AMD}_k and \mathcal{SD}_k . We proceed to describe \mathcal{MD}_k followed by the channel output correction phase.

3.1.3 The modified BP/OMS decoder

Recall, the modified BP/OMS decoder \mathcal{MD}_k computes a codeword estimate $\hat{\mathbf{x}}_k$ of \mathcal{C}_k by leveraging on the valid codeword $\hat{\mathbf{x}}_{k-1}$ of \mathcal{C}_{k-1} obtained in the $(k-1)$ th

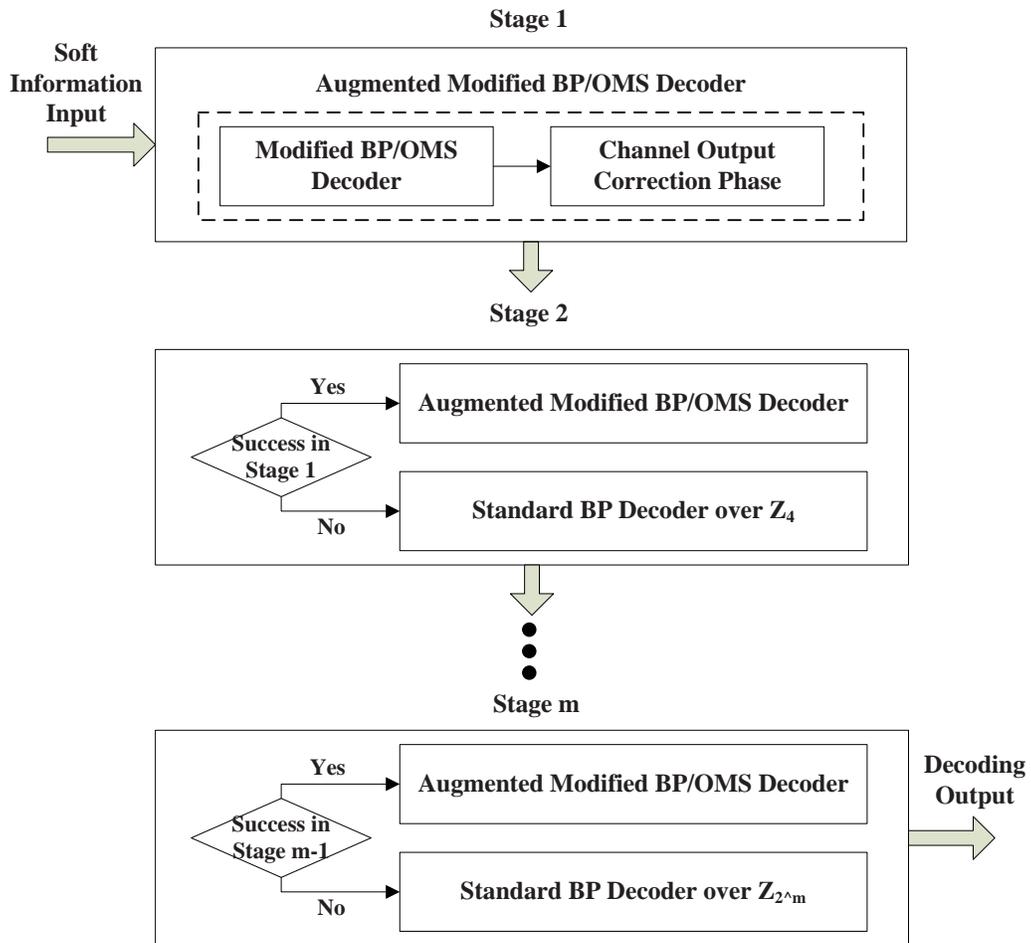


Figure 3.1: Flowchart of the DMD

decoding stage which, in its natural binary representation, has the form

$$(\hat{x}_{1,0}, \dots, \hat{x}_{1,k-2}, \dots, \hat{x}_{N,0}, \dots, \hat{x}_{N,k-2}).$$

Thus, the goal of \mathcal{MD}_k is to determine $\hat{x}_{1,k-1}, \dots, \hat{x}_{N,k-1}$. Consequently, when applying \mathcal{MD}_k to the Tanner graph associated to \mathbf{H}_k , the variable nodes will represent the $x_{j,k-1}$ whose values we wish to estimate. For convenience, the variable node representing $x_{j,k-1}$ will be referred to as $x_{j,k-1}$ as well.

The inputs to \mathcal{MD}_k are the *a priori* LLRs of the $x_{j,k-1}$. Each iteration of \mathcal{MD}_k involves two types of operations, row operations and column operations. A row (resp., column) operation involves the computation of check-to-variable (resp., variable-to-check) node messages. The message \mathbf{q}_{ji}^{k-1} passed from the j th variable node to the i th check node is the LLR $\log \frac{q_{ji}^0}{q_{ji}^1}$, where q_{ji}^b is the probability that $x_{j,k-1} = b$, given information obtained via the i' check node for all $i' \in M(j) \setminus \{i\}$. The values of the \mathbf{q}_{ji}^{k-1} are initialized by $\Lambda_{x_{j,k-1}}$, i.e., $\mathbf{q}_{ji}^{k-1} = \Lambda_{x_{j,k-1}}$ for all $i \in M(j)$.

For DMD-BP, the message \mathbf{r}_{ji}^{k-1} passed from the i th check node to the j th variable node is the LLR $\log \frac{r_{ji}^0}{r_{ji}^1}$, where r_{ji}^b is the probability that the i th parity-check constraint

$$\begin{aligned} \sum_{j' \in L(i) \setminus \{j\}} h_{ij'}^k \left(\sum_{k'=0}^{k-2} \hat{x}_{j',k'} 2^{k'} + x_{j',k-1} 2^{k-1} \right) \\ = -h_{ij}^k \left(\sum_{k'=0}^{k-2} \hat{x}_{j,k'} 2^{k'} + x_{j,k-1} 2^{k-1} \right) \end{aligned} \quad (3.1)$$

is satisfied when $x_{j,k-1} = b$ and the other variable nodes are independent with probabilities given by the elements of $\mathbf{q}_{j'i}^{k-1}$ for $j' \in L(i) \setminus \{j\}$. Following [17,

Section III], \mathbf{r}_{ji}^{k-1} is computed as

$$\begin{aligned} \mathbf{r}_{ji}^{k-1} &= \bigoplus_{\mathbf{x}' \in \mathbb{Z}_2^{k|L(i)|}: x_{j,k-1}=1} P((3.1) \text{ is satisfied} | \mathbf{x}') \sum_{j' \in L(i) \setminus \{j\}} x_{j',k-1} \mathbf{q}_{j'i}^{k-1} \\ &- \bigoplus_{\mathbf{x}' \in \mathbb{Z}_2^{k|L(i)|}: x_{j,k-1}=0} P((3.1) \text{ is satisfied} | \mathbf{x}') \sum_{j' \in L(i) \setminus \{j\}} x_{j',k-1} \mathbf{q}_{j'i}^{k-1} \end{aligned} \quad (3.2)$$

where $\mathbf{x}' := \{\tilde{\mathbf{x}}_j\}_{j \in L(i)}$, $\tilde{\mathbf{x}}_j := (\hat{x}_{j,0}, \dots, \hat{x}_{j,k-2}, x_{j,k-1})$ and $a \oplus b := -\log(\exp(-a) + \exp(-b)) = \min(a, b) - \log(1 + \exp(-|a - b|))$.

Further, \mathbf{q}_{ji}^{k-1} is computed as

$$\mathbf{q}_{ji}^{k-1} = \Lambda_{x_{j,k-1}} + \sum_{i' \in M(j) \setminus \{i\}} \mathbf{r}_{ji'}^{k-1} \quad (3.3)$$

An estimate $\hat{x}_{j,k-1}$ of $x_{j,k-1}$ is given by $\hat{x}_{j,k-1} := \text{sgn}(\mathbf{q}_j^{k-1})$, where

$$\text{sgn}(a) := \begin{cases} 0 & : a \geq 0 \\ 1 & : a < 0 \end{cases}$$

and

$$\mathbf{q}_j^{k-1} := \log \frac{q_j^0}{q_j^1} = \Lambda_{x_{j,k-1}} + \sum_{i \in M(j)} \mathbf{r}_{ji}^{k-1}$$

with q_j^b denoting the probability that $x_{j,k-1} = b$, given the information obtained via the i th check node for all $i \in M(j)$. Finally, if $\hat{\mathbf{x}}_k \mathbf{H}_k^T = \mathbf{0}$ where

$$\hat{\mathbf{x}}_k = \left(\sum_{k'=0}^{k-1} \hat{x}_{1,k'} 2^{k'}, \dots, \sum_{k'=0}^{k-1} \hat{x}_{N,k'} 2^{k'} \right)$$

then a valid codeword of \mathcal{C}_k has been found. Otherwise, further iterations are performed until a valid codeword is obtained or the prescribed maximum number of iterations is reached, whichever is earlier. This completes the description of \mathcal{MD}_k for DMD-BP.

For DMD-OMS, the message \mathbf{r}_{ji}^{k-1} passed from the i th check node to the j th variable node is obtained by first computing an approximation $\hat{\mathbf{r}}_{ji}^{k-1}$ of the LLR of r_{ji}^1 and r_{ji}^0 , and then reducing the absolute value of $\hat{\mathbf{r}}_{ji}^{k-1}$ by a predetermined positive constant f_c called the *offset* of the OMS algorithm. Following [125, Section 3.4 & Appendix A.3], $\hat{\mathbf{r}}_{ji}^{k-1}$ is computed as

$$\begin{aligned} \hat{\mathbf{r}}_{ji}^{k-1} = & \min_{\mathbf{x}' \in \mathbb{Z}_2^{k|L(i)|}: x_{j,k-1}=1} P((3.1) \text{ is satisfied} | \mathbf{x}') \sum_{j' \in L(i) \setminus \{j\}} x_{j',k-1} \mathbf{q}_{j'i}^{k-1} \\ & - \min_{\mathbf{x}' \in \mathbb{Z}_2^{k|L(i)|}: x_{j,k-1}=0} P((3.1) \text{ is satisfied} | \mathbf{x}') \sum_{j' \in L(i) \setminus \{j\}} x_{j',k-1} \mathbf{q}_{j'i}^{k-1}. \end{aligned} \quad (3.4)$$

Then, following [126, Section III-B], \mathbf{r}_{ji}^{k-1} is obtained from $\hat{\mathbf{r}}_{ji}^{k-1}$ using the rule

$$\mathbf{r}_{ji}^{k-1} = \begin{cases} \hat{\mathbf{r}}_{ji}^{k-1} - f_c & : \hat{\mathbf{r}}_{ji}^{k-1} > f_c \\ \hat{\mathbf{r}}_{ji}^{k-1} + f_c & : \hat{\mathbf{r}}_{ji}^{k-1} < -f_c \\ 0 & : \text{otherwise.} \end{cases}$$

For simplicity, we keep f_c fixed over all iterations executed by \mathcal{MD}_k , as in [126]. For the same reason, we also keep f_c fixed over all m stages of DMD-OMS. (*Remarks:* (i) In fact, for the codes considered in [126], the analysis therein suggests that varying f_c in each iteration achieves little additional improvements in performance; (ii) For the case where f_c is unchanged for all iterations executed by \mathcal{MD}_k , We conjecture that the optimal value of f_c for each of the m stages is the same. The premise for this conjecture is that for each k , \mathcal{MD}_k decodes some coset of \mathcal{C}_1 and all cosets of \mathcal{C}_1 have the same weight distribution. If this conjecture is true, then the optimal offset may be found by determining the optimal offset for the code \mathcal{C}_1 under OMS decoding.)

The variable-to-check node messages and the estimates of $x_{j,k-1}$ are computed in the exact same manner as described above. Thus, \mathcal{MD}_k in DMD-BP and DMD-OMS only differ in the way the check-to-variable node messages are computed. This completes the description of \mathcal{MD}_k for DMD-OMS.

3.1.4 The channel output correction phase

Recall, we enter the channel output correction phase while in the k th decoding stage if \mathcal{MD}_k fails to deliver a valid codeword of \mathcal{C}_k . The first task of this correction phase is to identify the variable nodes that participate in at least one unsatisfied parity-check constraint to form a set, V_k . For $x_{j,k-1} \in V_k$, we denote by $d(x_{j,k-1})$ the number of unsatisfied parity-check constraints that $x_{j,k-1}$ participates in and refer to this quantity as the d -value of $x_{j,k-1}$. The nodes in V_k are then sorted in descending order of their d -values with ties broken by further sorting the nodes that are tied in ascending order of the absolute value of their initial LLRs obtained from the channel output.

Denoting the first node in the sorted set by $\tilde{x}_{1,k-1}$, second node by $\tilde{x}_{2,k-1}$ and so on, we then set the *a priori* LLR $\Lambda_{\tilde{x}_{1,k-1}}$ of $\tilde{x}_{1,k-1}$ to $-\infty$ as in [72] and run a maximum of I_1 iterations of \mathcal{MD}_k . If the resulting codeword estimate is a codeword of \mathcal{C}_k , we exit the correction phase immediately and proceed to the $(k + 1)$ th decoding stage. On the other hand, if the resulting codeword estimate is not a codeword of \mathcal{C}_k , we set $\Lambda_{\tilde{x}_{1,k-1}}$ to $+\infty$ and run a maximum of I_1 iterations of \mathcal{MD}_k again, after restoring the *a priori* LLRs obtained from the channel output. If a codeword of \mathcal{C}_k is found, we proceed to the $(k + 1)$ th decoding

stage. Otherwise, we restore the *a priori* LLRs obtained from the channel output again, and repeat the above procedure to modify them. This time however, we modify $\Lambda_{\tilde{x}_{1,k-1}}$ and $\Lambda_{\tilde{x}_{2,k-1}}$ to yield the following four pairs of modified *a priori* LLRs:

$$\begin{aligned}
 \Lambda_{\tilde{x}_{1,k-1}} &= -\infty, & \Lambda_{\tilde{x}_{2,k-1}} &= -\infty \\
 \Lambda_{\tilde{x}_{1,k-1}} &= -\infty, & \Lambda_{\tilde{x}_{2,k-1}} &= +\infty \\
 \Lambda_{\tilde{x}_{1,k-1}} &= +\infty, & \Lambda_{\tilde{x}_{2,k-1}} &= -\infty \\
 \Lambda_{\tilde{x}_{1,k-1}} &= +\infty, & \Lambda_{\tilde{x}_{2,k-1}} &= +\infty
 \end{aligned}$$

In like manner, starting with the first pair, we run a maximum of I_2 iterations of \mathcal{MD}_k . If a codeword of \mathcal{C}_k is found, we proceed to the $(k+1)$ th decoding stage, otherwise we go on to the next pair. If a codeword is still not obtained after exhausting all four pairs, we start over, this time by modifying $\Lambda_{\tilde{x}_{1,k-1}}$, $\Lambda_{\tilde{x}_{2,k-1}}$ and $\Lambda_{\tilde{x}_{3,k-1}}$ to yield eight triplets of modified *a priori* LLRs. This process continues until we have exhausted a prescribed maximum number L of modified *a priori* LLRs or a valid codeword of \mathcal{C}_k is found, whichever is earlier. We name L the *correction limit* of the \mathcal{AMD} . Note that one can choose different values for L at different stages of the DMD. For simplicity however, we only consider the case when all the \mathcal{AMD} 's have the same value of L .

3.2 Simulation Results

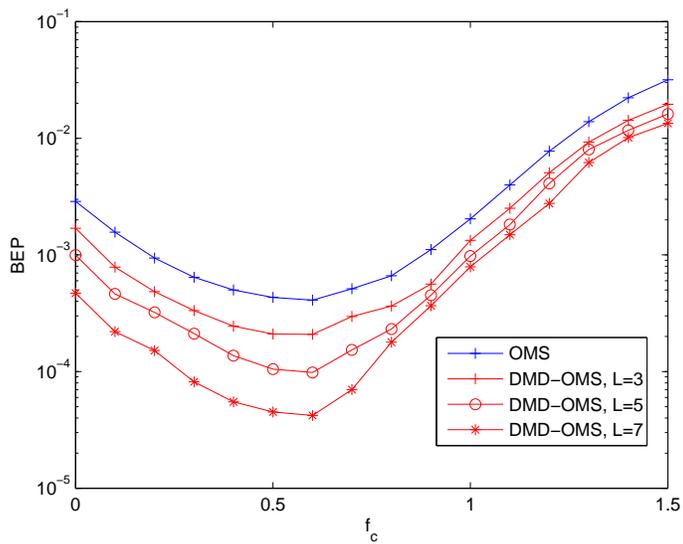
In this section, we demonstrate via computer simulations, the superiority of the DMD over standard BP decoding as well as the multistage BP decoder of [26] on an AWGN channel with BPSK signaling. Each BEP data point is obtained from collecting at least 1000 bit errors.

To this end, we first consider two rate-1/2, randomly generated, regular codes of column weight 3 and row weight 6. One code is a \mathbb{Z}_4 code, the other, a \mathbb{Z}_8 code. Both codes have an equivalent binary length of 1200 bits. The nonzero entries of their parity-check matrices are drawn uniformly from the units of their respective code alphabets. Moreover, both codes are free. In our simulations, I_i is set to 20 for $i = 1, \dots, L$. The maximum number of iterations of the component decoders \mathcal{SD}_k and \mathcal{MD}_k of the DMD is also set to 20 for all k . For each code, we varied the DMD-OMS's offset value f_c from 0.1 to 2.0 in steps of 0.1, simulated the DMD-OMS at a fixed SNR of 2.6 dB, as shown in In Figs. 3.2(a) and (b), and picked the f_c that yielded the best BEP performance. We obtained $f_c = 0.6$ for the \mathbb{Z}_4 code and $f_c = 0.5$ for the \mathbb{Z}_8 code. These values are near optimal as the optimal offset for the OMS decoder has been shown to be rather insensitive to SNR and is more a function of the code [126]. In comparison, the BEP performance for the canonical image \mathcal{C}_1 of the \mathbb{Z}_4 and \mathbb{Z}_8 code under OMS decoding with the offset value f_c ranging from 0.1 to 2.0 is also shown in Figs. 3.2(a) and (b). The offset values that yield the best BEP performance matched the offset values obtained by simulating the DMD-OMS, thus providing some numerical evidence to support

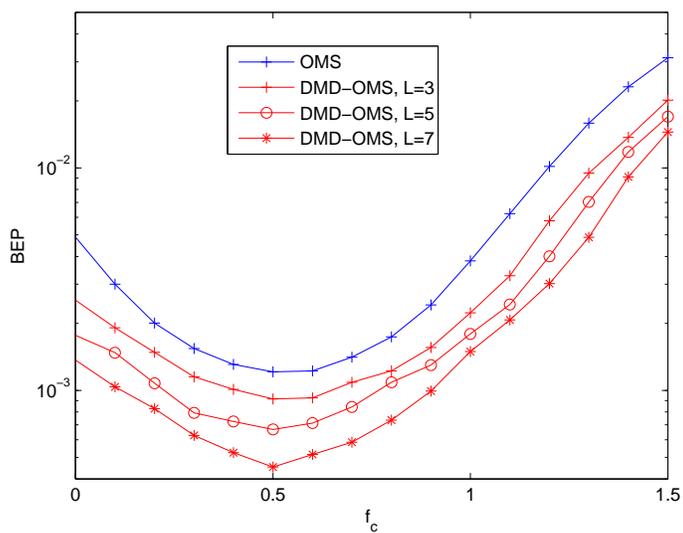
our conjecture in Section 3.1.3.

Fig. 3.3 and Fig. 3.4 show the BEP performance of the \mathbb{Z}_4 , respectively, \mathbb{Z}_8 code, when decoded using DMD-BP and DMD-OMS for $L = 3, 5, 7$. The BEP performance of standard BP decoding and the multistage decoder of [26] are also shown for comparison. The maximum number of iterations for the former decoder, which is \mathcal{SD}_2 for the \mathbb{Z}_4 code and \mathcal{SD}_3 for the \mathbb{Z}_8 code, is also set to 20. Similarly, the maximum number of iterations performed in each stage of the latter decoder is 20.

For the \mathbb{Z}_4 code, Fig. 3.3 shows that with $L = 3$, DMD-BP achieves a coding gain of 0.14 dB, 0.19 dB and 0.26 dB over standard BP decoding at a BEP of 10^{-4} , 10^{-5} and 10^{-6} , respectively. Not surprisingly, larger gains are observed when $L = 5$ and $L = 7$. In particular, at a BEP of 10^{-6} , DMD-BP achieves a coding gain of 0.32 dB and 0.37 dB over standard BP decoding when L is equal to 5 and 7, respectively. Remarkably, DMD-OMS provides even larger coding gains despite the OMS decoder being an approximation of the BP decoder. At a BEP of 10^{-6} , DMD-OMS achieves a coding gain of 0.32 dB, 0.39 dB and 0.45 dB over standard BP decoding when L is equal to 3, 5 and 7, respectively. That DMD-OMS outperforms standard BP decoding is not too surprising as it has already been reported elsewhere that for binary codes, OMS decoding can in some cases outperform BP decoding with properly chosen offsets. For example, in [74], the authors report the OMS decoder achieving a coding gain of about 0.2 dB over BP decoding for a moderate-length code at a BEP of 10^{-7} . The margin by which DMD-OMS outperforms DMD-BP is consistent with this result.



(a)



(b)

Figure 3.2: (a) BEP performance of DMD-OMS at different offset values for the \mathbb{Z}_4 code at the SNR of 2.6 dB; (b) BEP performance of DMD-OMS at different offset values for the \mathbb{Z}_8 code at the SNR of 2.6 dB

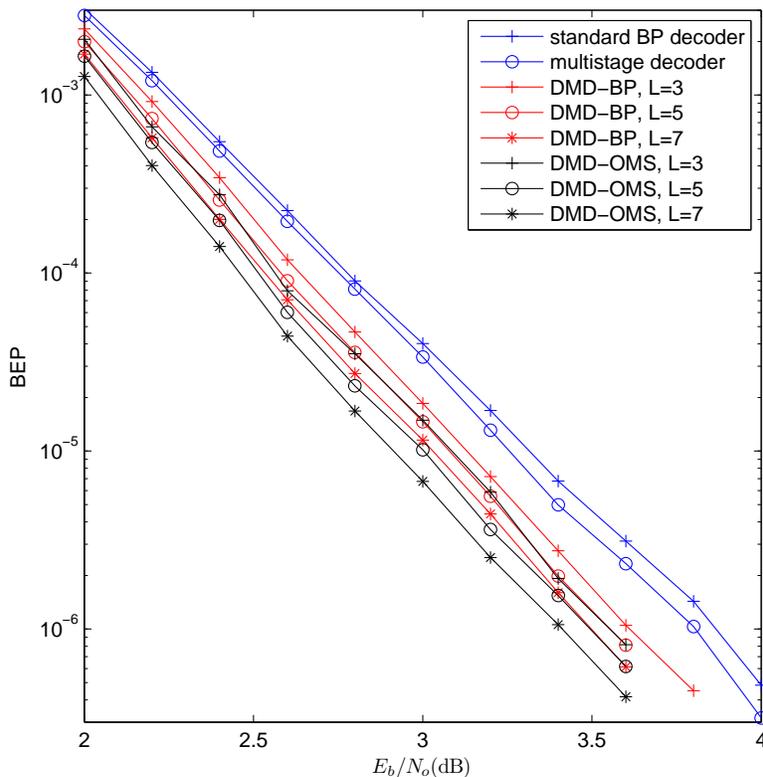


Figure 3.3: BEP performance of 1200-bit long regular \mathbb{Z}_4 code under various decoding strategies

For the \mathbb{Z}_8 code, Fig. 3.4 shows that with $L = 3$, DMD-BP achieves a coding gain of 0.16 dB, 0.25 dB and 0.31 dB over standard BP decoding at a BEP of 10^{-4} , 10^{-5} and 10^{-6} , respectively. Further, at a BEP of 10^{-6} , DMD-BP achieves a coding gain of 0.36 dB and 0.43 dB over standard BP decoding when L is 5 and 7, respectively. In contrast, at a BEP of 10^{-6} , DMD-OMS achieves a coding gain of 0.53 dB, 0.62 dB and 0.67 dB over standard BP decoding when L is equal to 3, 5 and 7, respectively, so we again see DMD-OMS outperforming DMD-BP.

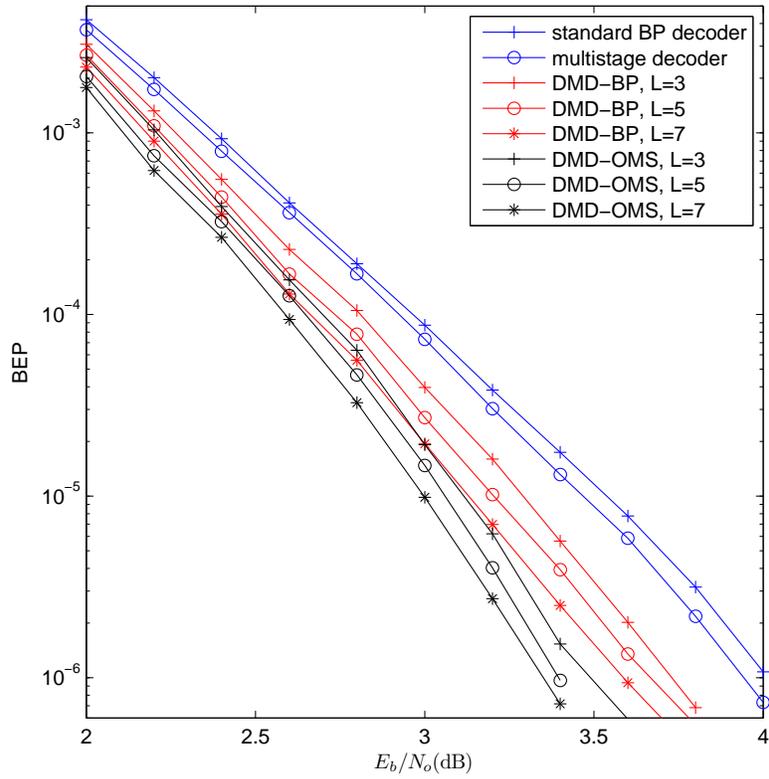
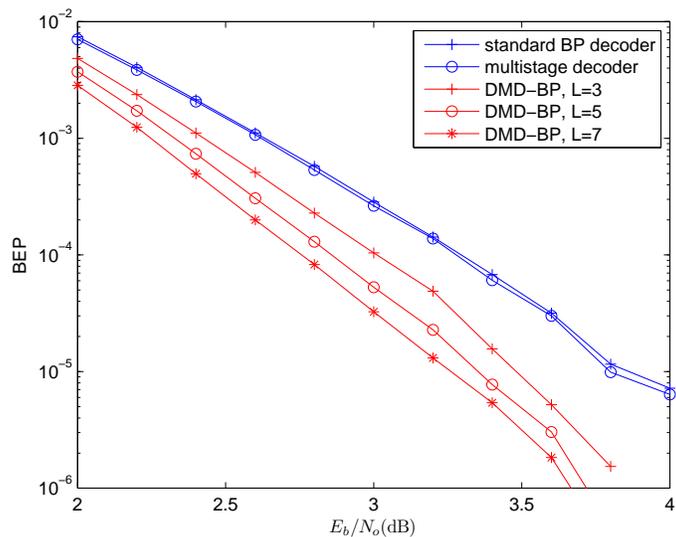


Figure 3.4: BEP performance of 1200-bit long regular \mathbb{Z}_8 code under various decoding strategies

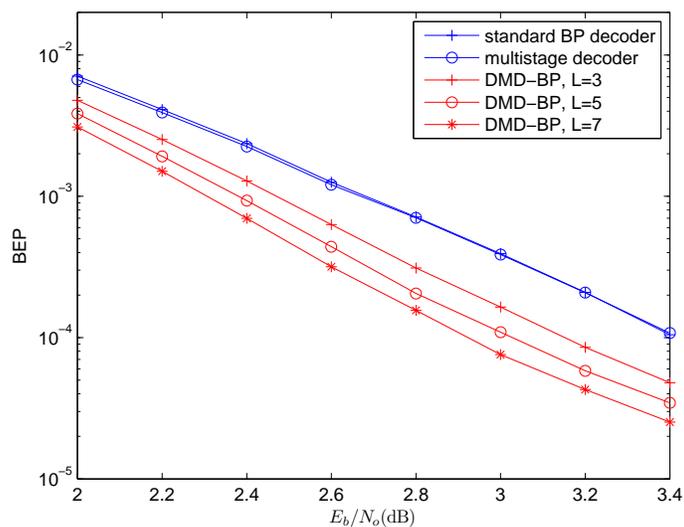
Having showed that both DMD-BP and DMD-OMS outperform standard BP decoding for moderate-length codes, we focus henceforth on demonstrating the performance of DMD-BP for short regular codes, moderate-length irregular codes, and long regular codes.

In Figs. 3.5(a) and (b), we consider short regular code of rate $1/2$, column weight 3, row weight 6 and an equivalent binary length of 500 bits over \mathbb{Z}_4 and \mathbb{Z}_8 , respectively. It can be observed that the additional coding gain over standard BP decoding that DMD-BP achieves is larger for this short code. In particular, for the \mathbb{Z}_4 code, at a BEP of 10^{-5} , DMD-BP achieves a coding gain of 0.39 dB, 0.5 dB and 0.59 dB over standard BP decoding when L is equal to 3, 5 and 7, respectively. These larger coding gains are due in part to the fact that the codelength N is now very small so that the normalized correction limit L/N is not so insignificant for the values of L considered. Consequently, the correction phase is more effective in enabling a valid codeword estimate to be found when \mathcal{MD}_k initially fails to do so. Similar observations can be made for the \mathbb{Z}_8 code.

In Figs. 3.6(a) and (b), we consider rate- $1/2$, regular code of equivalent binary length 12000 over \mathbb{Z}_4 and \mathbb{Z}_8 , respectively. It can be observed that there is not much difference in performance between DMD-BP and the multistage decoder of [26]. Further, the additional coding gain over standard BP decoding afforded by DMD-BP is only about 0.1 dB, with increasing values of L yielding little additional improvements. As stated earlier, this is not surprising since for long codes, the performance of standard BP decoding is already close to ML decoding performance. Hence, the performance gains of DMD-BP over standard BP decoding



(a)



(b)

Figure 3.5: (a) BEP performance of 500-bit long regular \mathbb{Z}_4 code under various decoding strategies; (b) BEP performance of 500-bit long regular \mathbb{Z}_8 code under various decoding strategies

will understandably be small.

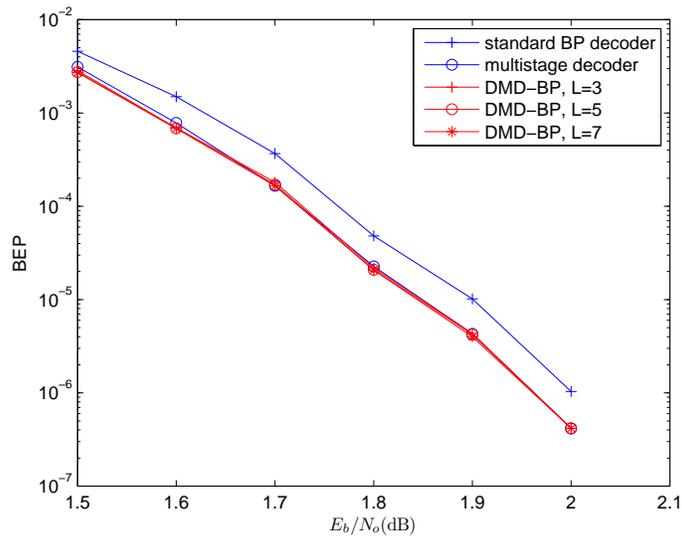
In Fig. 3.7, we consider a rate-1/2, irregular \mathbb{Z}_4 code whose canonical image over \mathbb{Z}_2 , i.e., \mathcal{C}_1 , is the “PEGirReg504x1008” code in [123]. Thus this \mathbb{Z}_4 code has an equivalent binary length of 2016 bits. Clearly, DMD-BP significantly outperforms standard BP decoding although error floors are observed at a BEP of around 10^{-6} . This is expected since (i) \mathcal{C}_1 exhibits an error floor at the BEP of around 10^{-6} , (ii) DMD-BP decodes \mathcal{C}_1 in the first decoding stage and a decoding error in this stage will propagate to the next.

3.3 LLR Density Analysis

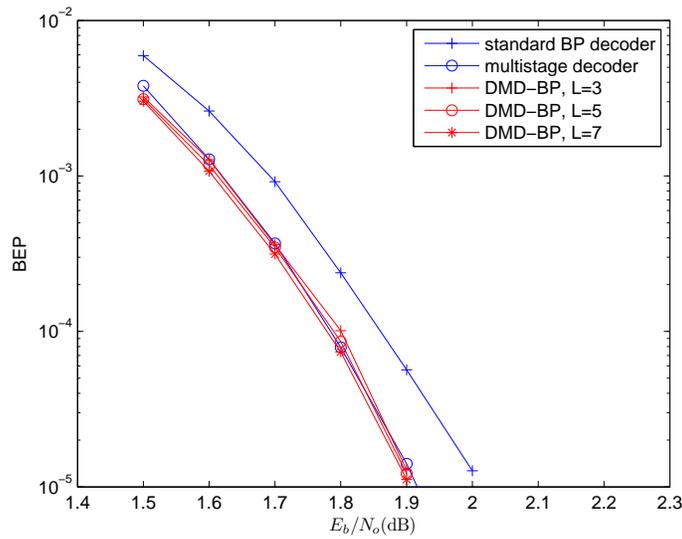
Next, we will show the effect of the channel output correction phase on the LLR densities of the decoded bits in the vicinity of the decision boundaries, to provide insight into why DMD-BP performs better than standard BP decoding. To this end, we will continue to use the moderate-length, regular \mathbb{Z}_8 code considered in Section 3.2.

We begin by noting that, regardless of which decoder is used, decoded bit errors occur most frequently in the position of the MSBs and least frequently in the position of the LSBs. The distribution of decoded bit errors arising from the independent transmission of 10^5 codewords at an SNR of 3 dB is summarized in Table 3.1. As we shall see in the next section, at this SNR value, the average computational complexity of DMD-BP will be lower than that of the standard BP decoder. We will therefore compare the (expected) LLR densities of the decoded

3. DMD ALGORITHM



(a)



(b)

Figure 3.6: (a) BEP performance of 12000-bit long regular \mathbb{Z}_4 code under various decoding strategies; (b) BEP performance of 12000-bit long regular \mathbb{Z}_8 code under various decoding strategies

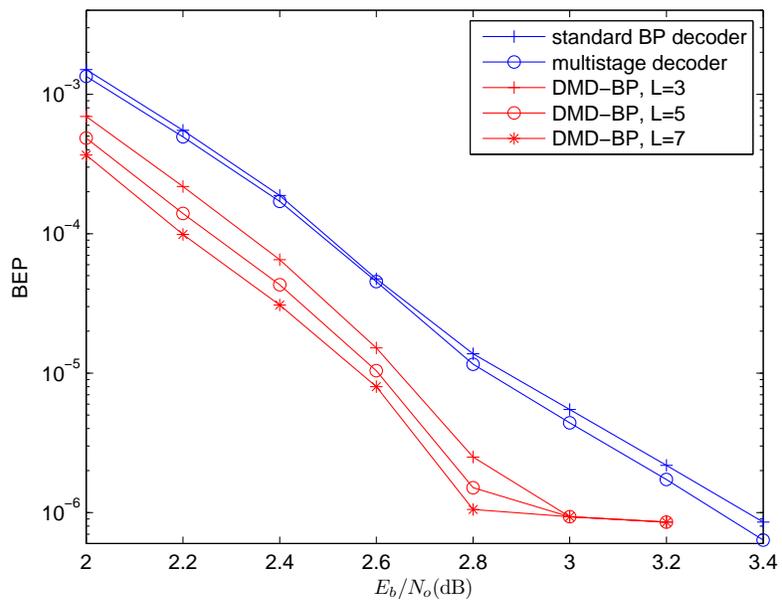


Figure 3.7: BEP performance of 2016-bit long irregular \mathbb{Z}_4 code under various decoding strategies

	Number of errors ($\times 10^3$)			
	LSB	2nd LSB	MSB	overall
standard BP decoder	0.776	14.762	144.028	159.566
DMD-BP, $L = 3$	0.657	8.284	30.621	39.562
DMD-BP, $L = 5$	0.593	6.703	24.452	31.748
DMD-BP, $L = 7$	0.522	5.520	19.680	25.722

Table 3.1: Distribution of decoded bit errors for \mathbb{Z}_8 code

LSBs through to the decoded MSBs resulting from the two decoding approaches at this SNR value.

To that end, we compute the LLR densities corresponding to the standard BP decoder after it has performed 5 iterations. The LLR $\Lambda(x_{j,k-1})$ of the k th LSB $x_{j,k-1}$ of the j th code symbol x_j is given by

$$\Lambda(x_{j,k-1}) = \log \frac{\sum_{\xi \in \mathbb{Z}_{2^m}: (\xi)_{k-1}=0} q_j^\xi}{\sum_{\xi \in \mathbb{Z}_{2^m}: (\xi)_{k-1}=1} q_j^\xi} \quad (3.5)$$

where q_j^ξ is the pseudo-posteriori probability that $x_j = \xi$ given the channel output, computed in the final iteration, and $(\xi)_{k-1}$ is the $(k-1)$ th coefficient in the 2-adic expansion of ξ .

To compute the LLR densities of the k th LSBs under doubly multistage decoding, we set each call to the component decoders $\mathcal{SD}_{k'}$ and $\mathcal{MD}_{k'}$ in stage k' , $1 \leq k' < k$, to incur a maximum of 20 iterations; for the remaining $m - k + 1$ stages, we set each call to $\mathcal{SD}_{k'}$ and $\mathcal{MD}_{k'}$, $k \leq k' \leq m$, to incur 5 iterations. The desired densities will then correspond to the completion of 5 iterations of

either \mathcal{MD}_k or $\mathcal{SD}_{k'}$ for some $k' \geq k$. If $\mathcal{SD}_{k'}$ is not invoked in stage k' for all $k' \geq k$ – which means that a valid codeword of \mathcal{C}_k is generated in stage k by \mathcal{MD}_k and the pseudo-posteriori probabilities q_j^0 and q_j^1 computed in the final iteration of \mathcal{MD}_k , are not superceded by the pseudo-posteriori probabilities computed by $\mathcal{SD}_{k'}$ in stage k' for all $k' > k$ – then the LLR of $x_{j,k-1}$ is given by

$$\Lambda(x_{j,k-1}) = \log \frac{q_j^0}{q_j^1}.$$

On the other hand, if \mathcal{SD}_k is invoked in stage k – which means that a codeword estimate of \mathcal{C}_k is generated in stage k by \mathcal{SD}_k rather than by \mathcal{MD}_k – then the LLR of $x_{j,k-1}$ is given by (3.5) with m replaced by k' and q_j^r reinterpreted as the pseudo-posteriori probability that $x_j = r$ given the channel output, computed in the final iteration of $\mathcal{SD}_{k'}$, where $k' \geq k$ such that $\mathcal{SD}_{k'}$ is the last instance of standard BP decoding invoked by DMD-BP. Note that we are therefore taking into account the possibility that the pseudo-posteriori probabilities computed by \mathcal{SD}_k in stage k , are superceded by the pseudo-posteriori probabilities computed by $\mathcal{SD}_{k'}$ in stage k' for some $k' > k$.

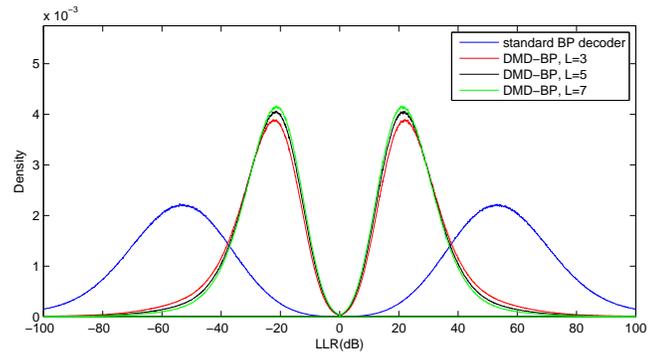
Figs. 3.8(a), (b) and (c) show the LLR densities of the decoded LSBs through to the MSBs generated from our computer simulations. At a glance, Figs. 3.8(a) and (b) do not appear to be congruent to Table 3.1. To reconcile them, we magnify Figs. 3.8(a) and (b) around the critical value of $\text{LLR} = 0$ in Figs. 3.9(a) and (b). For clarity, we only show the densities for bit 0. These magnifications show that for both the first and second LSBs, the LLR density function corresponding to DMD-BP experiences a sharp vertical drop, crossing below the LLR density

function corresponding to the standard BP decoder, as the LLR crosses below zero from above. Moreover, the larger the correction limit L , the further the former density function drops below the latter density function in the negative LLR region. Fig. 3.9(c), which magnifies Fig. 3.8(c) around $\text{LLR} = 0$, shows a similar but less dramatic drop in the LLR density function corresponding to DMD-BP as the LLR crosses below zero from above.

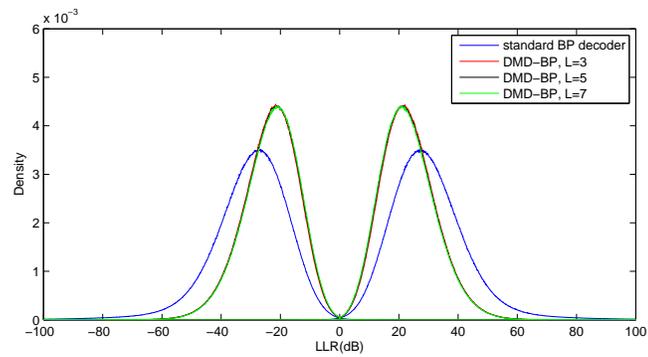
The mechanism of the channel output correction phase accounts for the discontinuities in the expected density functions under doubly multistage decoding. Recall, when \mathcal{MD}_k fails to deliver a valid codeword of \mathcal{C}_k , \mathcal{AMD}_k enters the correction phase. There, the *a priori* LLRs of one or more selected nodes will be modified to $-\infty$ or $+\infty$. The Gaussian distribution of the initial LLRs is consequently broken, leading to the observed discontinuities. These discontinuities are nevertheless desirable since the area under the graph of the LLR density function for negative LLR values, determines the bit error rate of the corresponding decoder. Since the LLR density functions corresponding to DMD-BP fall below the LLR density functions corresponding to standard BP decoding in the negative LLR region, the superiority of the former decoder is immediate.

Similar observations can be made for the other codes considered in Section 3.2 as shown in Figs. 3.10 and Figs. 3.11. Further, the same analysis can be applied to explain the superiority of DMD-OMS over standard BP decoding.

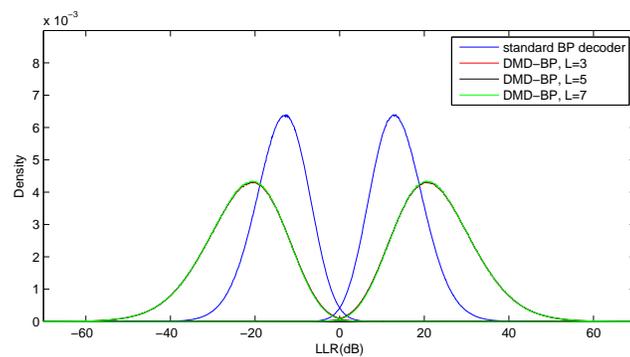
3. DMD ALGORITHM



(a)



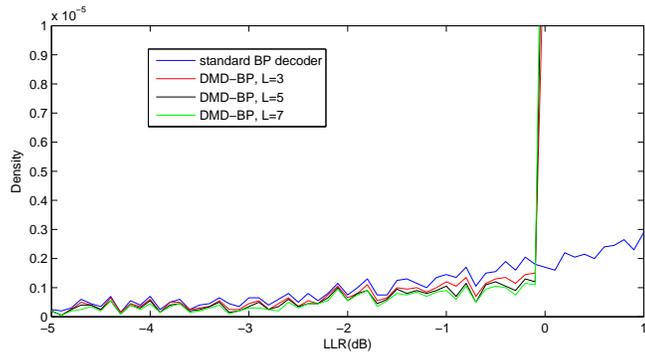
(b)



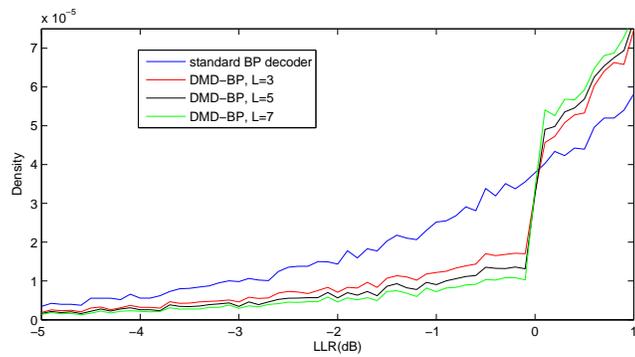
(c)

Figure 3.8: LLR density of various bit positions for the \mathbb{Z}_8 code: (a) LSB (b) 2nd LSB (c) MSB

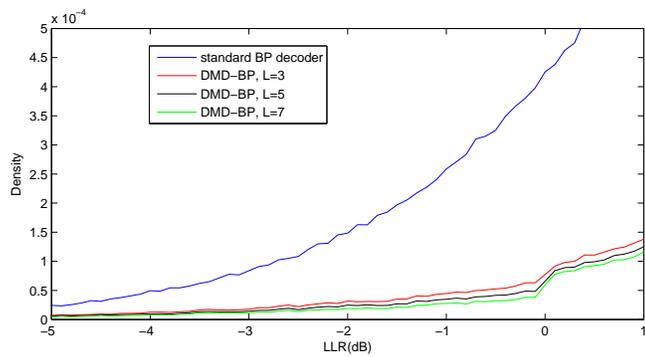
3. DMD ALGORITHM



(a)



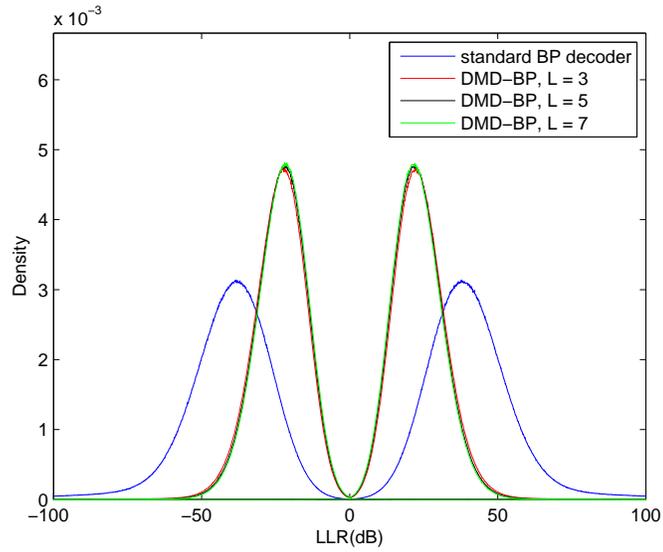
(b)



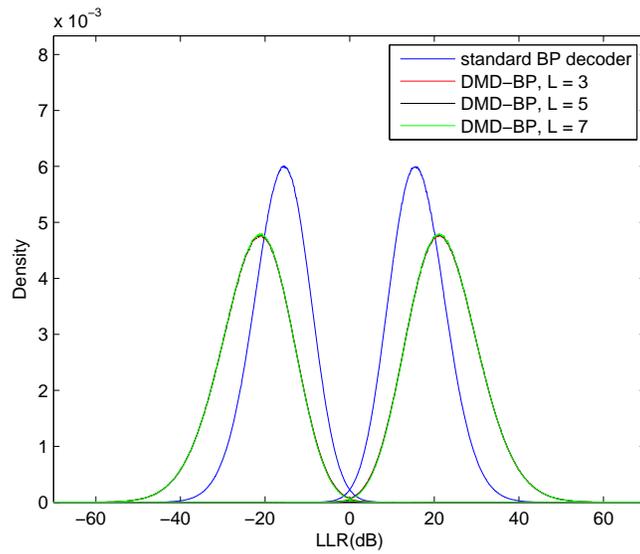
(c)

Figure 3.9: LLR density of various bit positions for the \mathbb{Z}_8 code around LLR=0:

(a) LSB (b) 2nd LSB (c) MSB

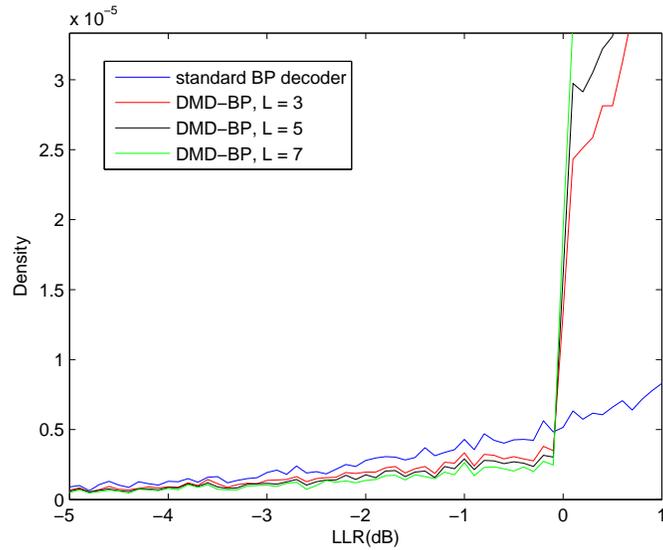


(a)

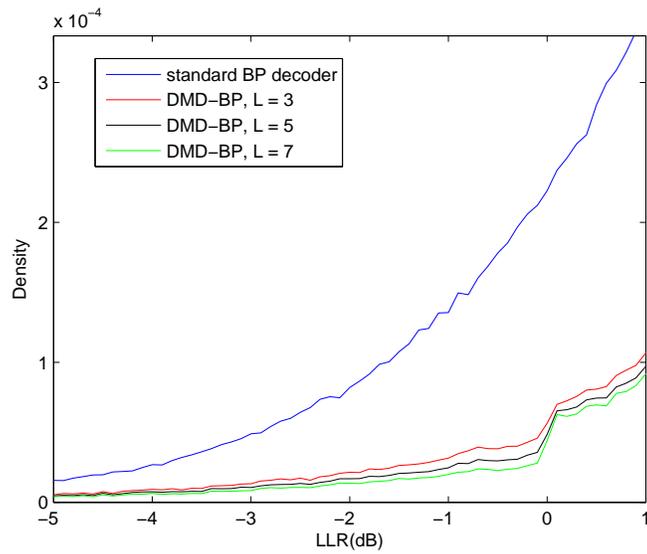


(b)

Figure 3.10: LLR density of various bit positions for the \mathbb{Z}_4 code: (a) LSB (b) MSB



(a)



(b)

Figure 3.11: LLR density of various bit positions for the \mathbb{Z}_4 code around LLR=0:

(a) LSB (b) MSB

3.4 Complexity Analysis

We next consider the price to pay in terms of the increase in decoding complexity for the above improvements in coding gain reported in Section 3.2 for our moderate-length, regular \mathbb{Z}_4 and \mathbb{Z}_8 codes. Since the majority of the operations performed in log-domain BP and OMS decoding are (real) additions and subtractions, we will count the number of additions/subtractions incurred by the DMD as a measure of its complexity. In the following analysis, we assume that the cost of one (real) multiplication/division is equivalent to the cost of two additions, and the cost of a subtraction is equal to the cost of an addition. (In a hardware implementation, the cost of these arithmetic operations will be the number of clock cycles needed to complete a given operation.) We further assume that the exponential and logarithm functions are both computed using the first 9 terms of their Taylor series expansions. Thus, evaluating an exponential or logarithm function requires 40 additions.

Recall that in the first of the m stages of the DMD, only the augmented modified decoder \mathcal{AMD}_k is deployed while in each subsequent stage, either the standard BP decoder \mathcal{SD}_k or \mathcal{AMD}_k is used. Thus, the average number of additions needed by the DMD to decode \mathcal{C}_m is given by

$$\Theta := \sum_{i=1}^m (P_k I_{\mathcal{MD}_k} \delta_{\mathcal{MD}_k} + (1 - P_k) I_{\mathcal{SD}_k} \delta_{\mathcal{SD}_k}) \quad (3.6)$$

where P_k is the probability that \mathcal{AMD}_k is used in the k th stage, $\delta_{\mathcal{MD}_k}$ and $\delta_{\mathcal{SD}_k}$ are the number of additions per iteration required by \mathcal{MD}_k and \mathcal{SD}_k , respectively, while $I_{\mathcal{MD}_k}$ and $I_{\mathcal{SD}_k}$ are the average number of iterations required by \mathcal{AMD}_k

and \mathcal{SD}_k , respectively. Clearly, P_k is always equal to 1 for $k = 1$.

For DMD-BP, the number of additions per iteration required by \mathcal{MD}_k is given by

$$\delta_{\mathcal{MD}_k} := \rho N \times (2\gamma + \rho + 167), \quad \text{for all } k$$

where ρ and γ denote the column and row weight of \mathcal{C}_m , respectively. For DMD-OMS, the number of additions per iteration required by \mathcal{MD}_k is given by

$$\delta_{\mathcal{MD}_k} := \rho N \times (2\gamma + \rho - 4), \quad \text{for all } k$$

The number of additions per iteration required by \mathcal{SD}_k is given by

$$\delta_{\mathcal{SD}_k} := \begin{cases} \rho N \times (2\gamma + \rho + 167) & : k = 1 \\ \rho N \times (12\gamma + 3\rho + 945) & : k = 2 \\ \rho N \times (56\gamma + 7\rho + 4417) & : k = 3 \end{cases}$$

In contrast, the average number of additions needed by \mathcal{SD}_m to decode \mathcal{C}_m is given by

$$\theta := I'_{\mathcal{SD}_m} \delta_{\mathcal{SD}_m}$$

where $I'_{\mathcal{SD}_m}$ is the average number of iterations needed by the standard BP decoder to decode \mathcal{C}_m .

By collecting all relevant statistics from our computer simulations, the ratio $\eta := \Theta/\theta$ of the (average) complexity of the DMD to that of the standard BP decoder when applied to our \mathbb{Z}_4 (respectively, \mathbb{Z}_8) code is computed for the different values of L considered and plotted against E_b/N_0 in Fig. 3.12 (respectively, Fig. 3.13). Note that the vertical axes in both figures are in logarithmic scale, i.e., what is actually plotted is $20 \log_{10} \eta$ against E_b/N_0 .

From Fig. 3.12, it is observed that at lower SNR's, DMD-BP has higher complexity than the standard BP decoder and its complexity increases exponentially with L , as the number of additional iterations required in the channel output correction phase increases exponentially with L when the noise power is large. However, as SNR increases, the complexity of DMD-BP decreases dramatically and drops below that of the standard BP decoder at an SNR of 2.2 dB when $L = 3$, 2.6 dB when $L = 5$, and 3 dB when $L = 7$. In particular, at an SNR of 3.6 dB which corresponds to a BEP of about 10^{-6} , the complexity of DMD-BP is 32% of that of the standard BP decoder. In contrast, DMD-OMS is only more complex than the standard BP decoder for $L = 7$ at lower SNR's. As with DMD-BP, its complexity decreases rapidly as SNR increases. At an SNR of 2.2 dB, its complexity when $L = 7$ is already less than that of the standard BP decoder. Remarkably, DMD-OMS requires less than 3% of the number of additions required by the standard BP decoder for SNR values exceeding 3.2 dB.

Similar observations can be made from Fig. 3.13. In particular, at an SNR of 3.4dB, the complexity of DMD-BP (resp., DMD-OMS) is only 10% and 1% of that of the standard BP decoder.

3.5 Concluding Remarks

Unlike the multistage decoder in [26] which decodes the m canonical images of \mathcal{C}_m sequentially over m stages, the DMD decodes cosets of \mathcal{C}_1 in \mathbb{Z}_2^N repeatedly over m stages. This modification enables the channel output correction phase to

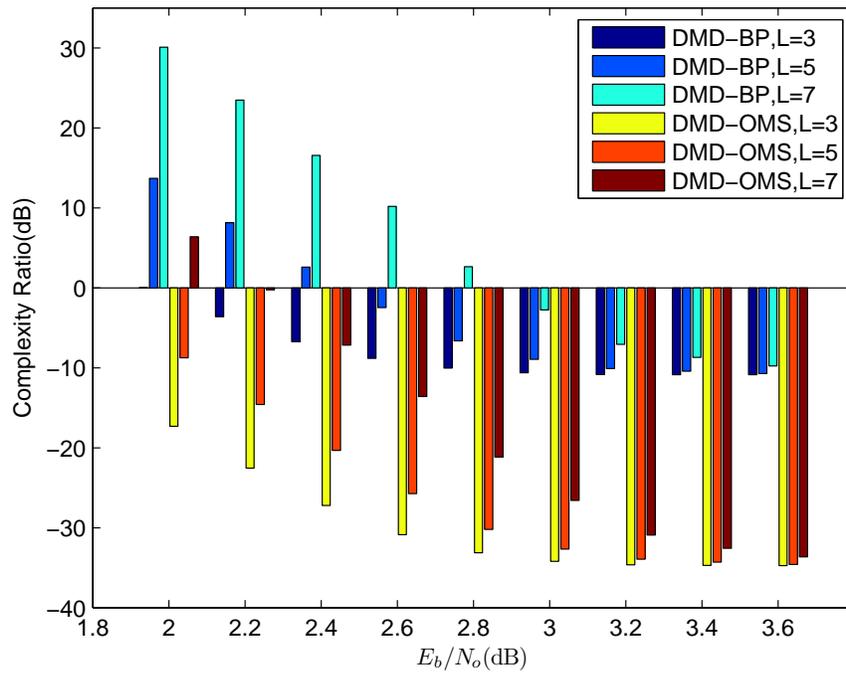


Figure 3.12: The complexity ratio associated with decoding the \mathbb{Z}_4 code

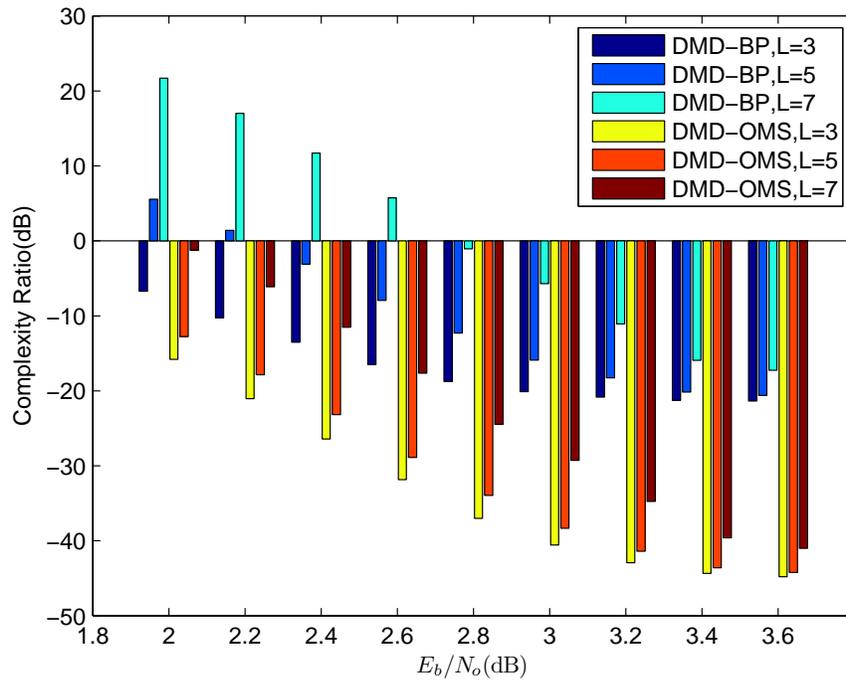


Figure 3.13: The complexity ratio associated with decoding the \mathbb{Z}_8 code

be incorporated in each decoding stage. We have demonstrated that the DMD can outperform standard BP decoding in terms of *both* BEP performance and complexity for moderate-length codes. Moreover, we have revealed the effect of the channel output correction phase on the LLR densities of the decoded bits in the vicinity of the decision boundaries, thus providing insight into why the DMD performs better than standard BP decoding. We have also showed that the DMD yields larger performance improvements for short codes. On the other hand, very little performance improvements are achieved for long codes. We emphasize that this is due to the fact that for long codes, the performance of BP decoding is already close to ML decoding performance. Thus, DMD is suitable for short to moderate-length codes.

Finally, we point out that while we have only considered two variants of the DMD, other variants are possible. For example, we could employ a modified normalized min-sum decoder in place of the modified OMS decoder in each decoding stage of the DMD. (See [126] for details of the NMS decoder.) We could also use the extended min-sum decoder of [19] in place of the standard BP decoder in each stage of the DMD. We have nevertheless not considered these other possibilities due to space constraints.

Chapter 4

The LLR Metric for q -ary LDPC Codes with MPSK Modulation over Rayleigh Channels with Imperfect CSI

The capacity-approaching codes, i.e., LDPC and turbo codes, can achieve reliable transmission at SNRs extremely close to the Shannon limit over the AWGN channel. During the iterative decoding process, the LLR of each received code bit is taken as the soft information input to the decoder and refined after each iteration. Therefore, using the correct LLR metric is crucial for reliable decoding.

Gallager derived the LLR metric for the AWGN channel in [4]. We focus on the time-selective, frequency-flat, Rayleigh fading channels, for which acquiring

the CSI is essential and the PSAM is frequently used for channel estimation. Iterative decoding of LDPC and turbo codes with PSAM channel estimation is discussed in [56–65]. In [56–62], the channel estimates are assumed to be perfect and the LLR metric based on that for the AWGN channel derived in [4] is used. In [63–65], the channel estimator is assumed to have a certain structure. By processing the received pilot information with the assumed channel estimator, the estimated channel gain and the estimation error variance are obtained. The problem of this ‘structured’ approach is that the LLR metric varies with the estimator structure, even for the same received signal sequence.

In this chapter, we present the correct conceptual approach for deriving the LLR metric of a q -ary code over time-selective, frequency-flat, Rayleigh fading channels with MPSK modulation and PSAM channel estimation. The derivation starts from first principles without assuming any receiver structure and demonstrates how the pilot information should be incorporated into the LLR computation. In particular, we demonstrate how the channel estimate and the estimation error variance enter in determining the reliability of each received coded symbol. The derivation shows why the MMSE channel estimator and the estimation error variance enter in the receiver. The decoding metric derived will be called the PSAM-LLR. As our simulation studies will show, the accuracy of the channel estimates, as measured by the channel estimation error variance, plays a crucial role in determining the error performance of the iterative decoder as well as the average number of iterations it takes to converge to a decision. The conventional LLR as used in [56–62] can be seen to be a special case of the PSAM-LLR when

the channel estimation error variance is assumed to be zero. This conventional LLR ignores the accuracy of the CSI, and is an approximate metric. We will call it the approximate PSAM-LLR (A-PSAM-LLR).

In transmitting an LDPC code, a very low SNR per symbol is usually used. The SNR per symbol of the pilot symbols is also very small. The channel estimation error variance is therefore usually high, and this should naturally be reflected in the reliability of each received code symbol when the CSI is used in its computation. In a time-selective fading channel, the channel estimation error variance varies with the position with respect to the pilot symbols, because of the channel decorrelation. Thus, the reliabilities of the code symbols also vary due to the different degrees of accuracy of the CSI, and this must be clearly reflected in the soft information input to the iterative decoder. Hence, performance degradation is expected for the A-PSAM-LLR, because the information of the estimation error variance is neglected. The simulation results verify that the PSAM-LLR has substantially better error performance, faster convergence speed and lower error floors than the A-PSAM-LLR. Our unstructured approach explains clearly why it is suboptimum to derive the metrics based on the channel estimates obtained from some predetermined estimators [56–65].

4.1 System Model

Let \mathcal{C}_q be a q -ary linear block code. Assume that we have $q = 2^{V_c}$ and the number of constellation points of the MPSK modulation is $M = 2^{V_m}$, where V_c and V_m are

positive integers and V_m is an integer multiple of V_c . We define $\kappa = V_m/V_c$. The system model is shown in Fig. 4.1. A q -ary message sequence \mathbf{m} is fed into the encoder of the code \mathcal{C}_q . The coded symbol sequence is sequentially divided into sub-sequences of length κ symbols and each sub-sequence is mapped to a MPSK signal with phase $\phi = \Gamma(x_1, x_2, \dots, x_\kappa)$, where $\Gamma(\cdot)$ is a one-to-one function that uniquely maps every sequence of κ variables, each of which belongs to the set $\{0, 1, \dots, q-1\}$, onto a phase $\phi \in \{\frac{2m\pi}{M}\}_{m=0}^{M-1}$. The signal sequence generated is then shuffled by a block interleaver to protect the transmission against burst errors in the fading channel. After that, the signal sequence is partitioned into sub-blocks of length B . Let $s(l, b) = \sqrt{E_s}e^{j\phi(l, b)}$ denote the b -th complex baseband signal in the l -th sub-block, where $\phi(l, b)$ is the information content in $s(l, b)$. Note that each MPSK signal contains κ coded symbols. We use $x(l, b, v)$ to denote the v -th coded symbol in the b -th complex baseband signal in the l -th sub-block. A known pilot symbol $p(l)$ is placed at the beginning of each sub-block, denoted by $s(l, 0)$. The pilot symbols are spaced B symbol intervals apart from one another. B is called the pilot symbol spacing. Each received signal can be expressed as

$$r(l, b) = c(l, b)s(l, b) + n(l, b), \quad (4.1)$$

where $r(l, b)$ denotes the received signal over the b -th interval in the l -th sub-block, and $\{n(l, b)\}$ is a set of statistically independent, complex, Gaussian random variables, each with mean zero and variance N_0 . The channel gains $\{h(k)|h(k) = c(l, b), k = (l-1)(B+1)+b+1\}$ are modeled as a correlated, zero-mean, complex, Gaussian process with autocorrelation function $\mathbb{E}[h(k)h^*(k-i)] = 2R_c(i)$.

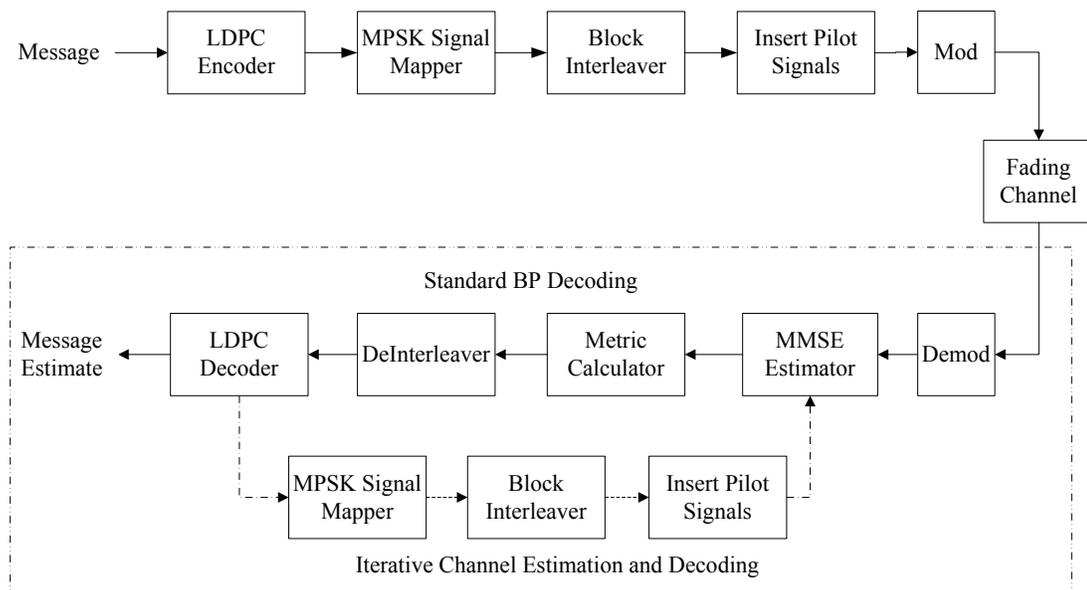


Figure 4.1: System Model

Note that the receiver structure shown in the dotted rectangle in Fig. 4.1 is proposed based on our derivation result and it will be discussed later in section 4.3.

4.2 Metric Derivation

We assume that the channel SNR and the CSI statistics are fixed and known by the receiver. Let $r_p(l)$ denote the received signal corresponding to the pilot signal $p(l)$. For the b -th signal in the l -th sub-block, define its observation window of length $W = 2Q + 1$ as the received signals within a distance of Q signals from it. Note that each observation window contains $2K$ pilot signals (K preceding and K succeeding), where $K = \lfloor \frac{Q}{B+1} \rfloor$, except near the beginning or end of

the transmitted sequence. Denote the vector formed by the pilot signals within the observation window of the b -th interval in the l -th sub-block, by $\mathbf{S}(l) = [r_p(l - K + 1), r_p(l - K + 2) \dots, r_p(l + K)]^T$.

For a q -ary code, the LLR metric Λ_x for a coded symbol x is a vector of length $q - 1$, which is represented as $\Lambda_x = [\lambda_x^1, \lambda_x^2, \dots, \lambda_x^{q-1}]^T$, where $\lambda_x^i = \ln \frac{P(x=0)}{P(x=i)}$. By definition, the log-likelihood ratio $\lambda^i(l, b, v)$ for the v -th coded symbol in the b -th complex baseband signal in the l -th sub-block is computed based on the received signal $r(l, b)$ and the pilot signal vector $\mathbf{S}(l)$ as

$$\lambda^i(l, b, v) = \ln \frac{P(x(l, b, v) = 0 | r(l, b), \mathbf{S}(l))}{P(x(l, b, v) = i | r(l, b), \mathbf{S}(l))}, \quad (4.2)$$

Assuming that the *a priori* probabilities of the information symbols are equal, by Bayes' rule, we get

$$\lambda^i(l, b, v) = \ln \frac{p(r(l, b) | x(l, b, v) = 0, \mathbf{S}(l))}{p(r(l, b) | x(l, b, v) = i, \mathbf{S}(l))}.$$

By introducing the instantaneous channel gain $c(l, b)$, the numerator and the denominator of the likelihood ratio in $\lambda^i(l, b, v)$ can be rewritten in the form

$$\begin{aligned} & p(r(l, b) | x(l, b, v) = i, \mathbf{S}(l)) \\ &= \int p(r(l, b), c(l, b) | x(l, b, v) = i, \mathbf{S}(l)) dc(l, b) \\ &= \int p(r(l, b) | x(l, b, v) = i, \mathbf{S}(l), c(l, b)) \\ & \quad p(c(l, b) | x(l, b, v) = i, \mathbf{S}(l)) dc(l, b), \end{aligned} \quad (4.3)$$

Since $x(l, b, v)$ takes on the values in $\{0, 1, \dots, q - 1\}$ with equal probability, the phase $\phi(l, b)$ conditioned on $x(l, b, v) = i$ is uniformly distributed over the set of

all possible phases corresponding to the event $x(l, b, v) = i$, which is denoted by $\Phi^i(v)$. It is not difficult to see that $|\Phi^i(v)| = \frac{M}{q}$.

Conditioned on $c(l, b)$ and any particular phase corresponding to $x(l, b, v) = i$, the only randomness in $r(l, b)$ is due to the AWGN $n(l, b)$, which is independent of the pilot signal set $\mathbf{S}(l)$. Hence, the first term in the integrand in the last line of (4.3) can be expressed as

$$\begin{aligned} & p(r(l, b)|x(l, b, v) = i, c(l, b)) \\ &= \sum_{\phi \in \Phi^i(v)} \frac{q}{M} \frac{1}{\pi N_0} \exp\left[-\frac{|r(l, b) - e^{j\phi} E_s^{\frac{1}{2}} c(l, b)|^2}{N_0}\right]. \end{aligned} \quad (4.4)$$

Next, we consider the term $p(c(l, b)|x(l, b, v) = i, \mathbf{S}(l))$. The channel gain $c(l, b)$ is independent of the transmitted information $\phi(l, b)$ or $x(l, b, v)$. Since the channel gains are jointly Gaussian, $c(l, b)$ and $\mathbf{S}(l)$ are also jointly Gaussian. Thus, $c(l, b)$ conditioned on $\mathbf{S}(l)$ is a Gaussian random variable with mean $\mathbb{E}[c(l, b)|\mathbf{S}(l)] = \hat{c}(l, b)$ and variance $\text{var}[c(l, b)|\mathbf{S}(l)] = 2\tilde{\sigma}^2(l, b)$ [156]. We will show later how to compute $\hat{c}(l, b)$ and $2\tilde{\sigma}^2(l, b)$ based on the knowledge of the autocorrelation function of the channel gain. Now, we have

$$\begin{aligned} & p(c(l, b)|x(l, b, v) = i, \mathbf{S}(l)) \\ &= \frac{1}{\pi 2\tilde{\sigma}^2(l, b)} \exp\left[-\frac{|c(l, b) - \hat{c}(l, b)|^2}{2\tilde{\sigma}^2(l, b)}\right]. \end{aligned} \quad (4.5)$$

Substituting (4.4) and (4.5) into (4.3), exchanging the order of integration and

summation, combining the exponents, and performing the integration, we obtain

$$\begin{aligned} p(r(l, b)|x(l, b, v) = i, \mathbf{S}(l)) \\ = \frac{q\alpha}{M\pi\beta} \sum_{\phi \in \Phi^i(v)} \exp\left[\frac{\gamma_\phi}{\beta 2\tilde{\sigma}^2(l, b)N_0}\right], \end{aligned} \quad (4.6)$$

where

$$\alpha = \exp\left[-\frac{|r(l, b)|^2}{N_0} - \frac{|\hat{c}(l, b)|^2}{2\tilde{\sigma}^2(l, b)}\right], \quad (4.7)$$

$$\beta = 2\tilde{\sigma}^2(l, b)E_s + N_0, \quad (4.8)$$

$$\gamma_\phi = |2\tilde{\sigma}^2(l, b)\sqrt{E_s}r(l, b)e^{-j\phi} + N_0\hat{c}(l, b)|^2. \quad (4.9)$$

Substituting (4.6)–(4.9) back into the expression of $\lambda^i(l, b, v)$ and after simplification, we obtain

$$\lambda^i(l, b, v) = \ln \frac{\sum_{\phi \in \Phi^0(v)} \exp(\psi_\phi)}{\sum_{\phi \in \Phi^i(v)} \exp(\psi_\phi)}, \quad (4.10)$$

where

$$\psi_\phi = \frac{2\sqrt{E_s}}{2\tilde{\sigma}^2(l, b)E_s + N_0} \operatorname{Re}\{r(l, b)\hat{c}^*(l, b)e^{-j\phi}\}.$$

We refer to this LLR metric in (4.10) as the PSAM-LLR.

Observe that the numerator and the denominator of the likelihood ratio in $\lambda(l, b, v)$ are summation of exponential functions. By assuming that $\sum_i \exp(x_i) \approx \exp(\max_i x_i)$, an approximation of the metric, after simplification, can be expressed as

$$\begin{aligned} \lambda^i(l, b, v) = \frac{2\sqrt{E_s}}{2\tilde{\sigma}^2(l, b)E_s + N_0} \\ \operatorname{Re}\{r(l, b)\hat{c}^*(l, b)(e^{-j\phi^0(v)} - e^{-j\phi^i(v)})\}, \end{aligned} \quad (4.11)$$

where

$$\phi^i(v) = \arg \max_{\phi \in \Phi^i(v)} \psi_\phi \quad (4.12)$$

We refer to the LLR metric in (4.11) as the simplified-PSAM-LLR (S-PSAM-LLR). Note that if the size of the code alphabet is equal to the number of constellation points of the MPSK modulation, i.e., $V_c = V_m$, the PSAM-LLR and the S-PSAM-LLR are trivially the same. Through simulations, we will show in general that these two metrics yield similar performance and can be treated as an alternative to each other.

If it is assumed that the channel estimation is perfect, it is easy to obtain

$$\lambda^i(l, b, v) = \ln \frac{\sum_{\phi \in \Phi^0(v)} \exp(\psi_\phi^{\text{app}})}{\sum_{\phi \in \Phi^i(v)} \exp(\psi_\phi^{\text{app}})}, \quad (4.13)$$

where

$$\psi_\phi^{\text{app}} = \frac{2\sqrt{E_s}}{N_0} \text{Re}\{r(l, b)\hat{c}^*(l, b)e^{-j\phi}\}.$$

We refer to this metric in (4.13) as the A-PSAM-LLR.

Similarly, assuming $\sum_i \exp(x_i) \approx \exp(\max_i x_i)$, we simplify (4.13) and get

$$\lambda^i(l, b, v) = \frac{2\sqrt{E_s}}{N_0} \text{Re}\{r(l, b)\hat{c}^*(l, b)(e^{-j\phi^0(v)} - e^{-j\phi^i(v)})\}, \quad (4.14)$$

where $\phi^i(v)$ is defined in (4.12), and we refer to this metric in (4.14) as the Simplified-Approximate-PSAM-LLR (SA-PSAM-LLR) metric, which is an alternative to the A-PSAM-LLR metric.

Next, we show how to compute $\hat{c}(l, b)$ and $2\tilde{\sigma}^2(l, b)$. By [156], the conditional mean is the MMSE estimate of $c(l, b)$ and the conditional variance is the MSE.

We get

$$\hat{c}(l, b) = \boldsymbol{\omega}_o^H(b) \mathbf{S}(l). \quad (4.15)$$

Here, $\boldsymbol{\omega}_o(b)$ is the optimum weight vector of the MMSE estimator and is given by $\boldsymbol{\omega}_o(b) = \mathbf{R}^{-1} \mathbf{p}(b)$. $\mathbf{R} = \mathbb{E}[\mathbf{S}(l) \mathbf{S}(l)^H] = E_s 2\sigma^2 \mathbf{G} + N_0 \mathbf{I}$ is the auto-correlation matrix, and $\mathbf{p}(b) = \mathbb{E}[\mathbf{S}(l) c^*(l, b)] = \sqrt{E_s} 2\sigma^2 \mathbf{v}(b)$ is the cross-correlation vector, where $\mathbf{G} = [G_{ij}]$ is a $2K \times 2K$ matrix whose ij -th element is $G_{ij} = R_c((i-j)(B+1))$. \mathbf{I} is the identity matrix, and $\mathbf{v}(b)$ is a $2K \times 1$ column vector with the i -th element given by $R_c((i-K)(B+1) - b)$. The conditional variance $2\tilde{\sigma}^2(l, b)$ can be expressed as

$$2\tilde{\sigma}^2(l, b) = 2R_c(0) - \mathbf{p}^T(b) (\mathbf{R}^{-1})^T \mathbf{p}(b) \quad (4.16)$$

Note that when the SNR goes to infinity, the term $\frac{N_0}{E_s}$ approaches zero, and the MSE tends to

$$2\tilde{\sigma}^2(l, b)_\infty = 2R_c(0) (1 - \mathbf{v}(b)^H (\mathbf{G}^{-1})^H \mathbf{v}(b)). \quad (4.17)$$

If the normalized fade rate $f_d T_s > 0$, this MSE $2\tilde{\sigma}^2(l, b)_\infty$ is not equal to zero. In other words, the CSI at the receiver cannot be perfectly acquired even when the SNR is very large because of channel decorrelation.

- *Remark*

Consider the important case of BPSK-modulated binary code. Simplifying (4.10) and (4.13), we get the PSAM-LLR metric (4.18) and the A-PSAM-LLR metric

(4.19).

$$\lambda(l, b) = \frac{1}{\frac{E_s}{N_0} 2\tilde{\sigma}^2(l, b) + 1} \frac{4\sqrt{E_s}}{N_0} \text{Re}\{r(l, b)\hat{c}^*(l, b)\} \quad (4.18)$$

$$\lambda(l, b) = \frac{4\sqrt{E_s}}{N_0} \text{Re}\{r(l, b)\hat{c}^*(l, b)\} \quad (4.19)$$

In (4.18), the term $\frac{4\sqrt{E_s}}{N_0} \text{Re}\{r(l, b)\hat{c}^*(l, b)\}$, which represents the component of the LLR calculated based on the channel output and the channel estimate, is scaled down by the factor of $\frac{1}{\frac{E_s}{N_0} 2\tilde{\sigma}^2(l, b) + 1}$ to account for the channel estimation errors. Intuitively, one can expect that the reliability of each received code bit should depend on the channel output, the channel estimate and the accuracy of the channel estimate. It is convenient to neglect the estimation errors and use the A-PSAM-LLR metric (4.19) as an approximation to the PSAM-LLR, only when the estimation errors are very small. However, this A-PSAM-LLR metric was widely used, while the channel estimation accuracy was ignored [56–62].

Error floors have been observed in PSAM systems, as reported in [59] and [60], in which the A-PSAM-LLR metric is used as the reliability function and the reason can be seen from the structure of the PSAM-LLR metric. Because of the channel decorrelation, the CSI cannot be estimated perfectly even as the SNR goes to infinity, as shown in (4.17). Compared to the PSAM-LLR metric, the term $\frac{1}{\frac{E_s}{N_0} 2\tilde{\sigma}^2(l, b) + 1}$, which accounts for the channel estimation inaccuracy, is missing in the A-PSAM-LLR expression. This term is strictly less than one and thus the A-PSAM-LLR metric always over-estimates the reliability of the channel output. Because of the non-zero irreducible MSE $2\tilde{\sigma}^2(l, b)$, the term $\frac{E_s}{N_0} 2\tilde{\sigma}^2(l, b)$ dominates the denominator when the SNR is high. As a result, the scaling factor

$\frac{1}{\frac{E_s}{N_0} 2\tilde{\sigma}^2(l,b)+1}$ becomes much smaller than one. Hence, the A-PSAM-LLR metric will exceedingly over-estimate the reliability information and become less accurate. This explains why error floors are observed in high SNR regions in [59, 60].

4.3 Receiver Design

To compute the LLR information using the PSAM-LLR metric in (4.10), we need the knowledge of the MMSE estimate $\hat{c}(l, b)$ and the MSE $2\tilde{\sigma}^2(l, b)$. Hence, the MMSE channel estimator is required at the receiver and the MSE should be computed for each channel estimate. Dictated by the PSAM-LLR metric, the receiver structure is shown in the dotted rectangle in Fig. 4.1. The MMSE estimator provides the channel estimate $\hat{c}(l, b)$ and MSE $2\tilde{\sigma}^2(l, b)$. The LLR for each coded symbol is then computed using the proposed LLR metrics. After deinterleaving, the LLR soft information is then passed to the BP decoder to obtain the estimated message $\hat{\mathbf{m}}$. This process is referred to as *standard BP decoding*.

Another decoding process, called *iterative channel estimation and decoding*, is shown in Fig. 4.1 with the dotted path included. Initially, the received pilot signals are used by the MMSE estimator to estimate the channel gain and the MSE, based on which, the LLRs of the coded symbols are computed and passed to the BP decoder. After several BP iterations, tentative coded symbol decisions, which are either in the form of soft information or hard information, are obtained. These decisions are mapped to MPSK signals, reinterleaved with

pilot signals reinserted, and then fed back to the MMSE channel estimator. New channel estimates and MSEs are computed based on these tentative decisions and the received signal sequence. The LLR information is updated using the new estimates and then passed to the BP decoder for further decoding. This process can be repeated several times before making the final decision. Note that this process of iterative channel estimation and decoding can provide better BEP performance than the standard BP decoding, which will be demonstrated in section 4.4, but it is much more computationally involved.

4.4 Simulation Study and Discussion

In our simulations, the binary code used is a rate-half, regular, (1008, 504) LDPC code of column weight three, adopted from [123]. Two types of the commonly used nonbinary codes, namely codes over Galois field $\mathbb{GF}(q)$ [9] and codes over integer residue ring \mathbb{Z}_q [24] are considered. The nonbinary codes used in our simulations are randomly generated, rate-half, regular codes of column weight three, having an equivalent binary length of 1008 bits. The sum-product algorithm is used with a maximum of 50 iterations. Each BEP data point is obtained from collecting at least 1000 bit errors.

In all the simulations, we choose the block interleaver of size 1008×10 . The observation window length is fixed at $W = 801$. The SNR refers to the energy per message bit E_b over the power N_0 of the AWGN. By assuming that a pilot signal takes up the same amount of energy as a data signal, the energy per transmitted

signal is $E_s = \frac{RV_m B}{B+1} E_b$.

In our simulations, we follow Jakes's isotropic scattering model [152] by assuming that the real and imaginary parts of $h(k)$ are independent, each with autocorrelation

$$R_c(i) = \sigma^2 J_0(2\pi f_d T_s i)$$

where f_d is the relative Doppler shift between the transmitter and the receiver, T_s is the symbol period and $J_0(\cdot)$ is the Bessel function of the first kind of order zero.

4.4.1 Effects of Interleaver

We first study the effect of size of block interleavers using binary codes with BPSK modulation, under the assumption of perfect CSI. For simplicity, the length of the interleaver is chosen to be equal to the length of the LDPC code, i.e., $n = 1008$. The BEP performance of interleavers of different widths is compared and shown in Fig. 4.2. It is observed that as the size of the interleaver increases, better BEP performance can be achieved. The marginal BEP improvement becomes insignificant when the interleaved BEP performance is close to that of the uncorrelated fading channel. To achieve a targeted BEP, interleavers with large sizes are required for channels with lower fade rates.

On the other hand, although increasing the size of interleaver can help to break the memory of the correlated fading channel, it also increases the latency of the system, because the receiver has to obtain all the data signals in an interleaved

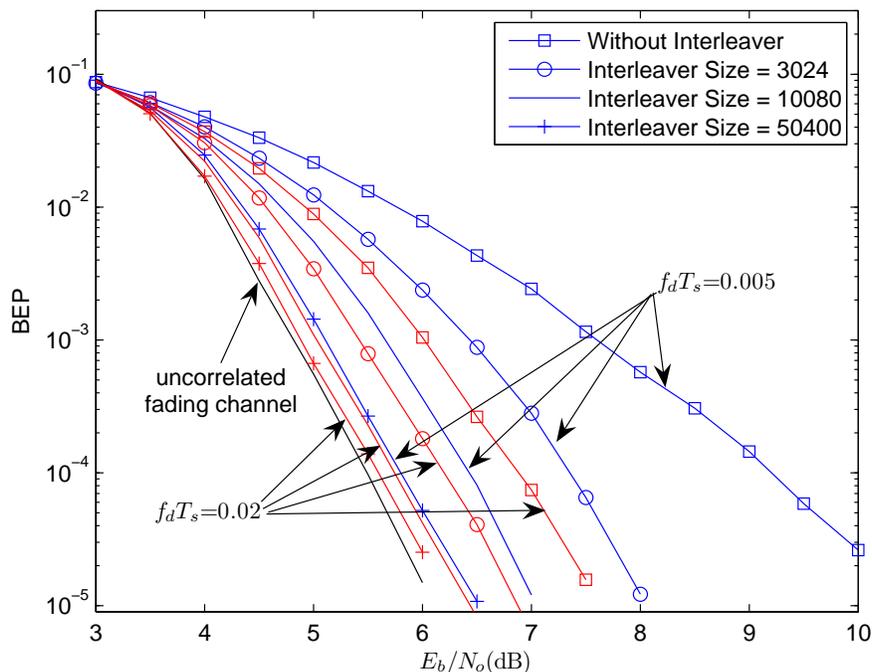


Figure 4.2: Effects of size of interleavers over Rayleigh fading channels with normalized fade rates $f_d T_s = 0.005$ and $f_d T_s = 0.02$ with perfect CSI

block before it starts to decode the message.

For all the simulations in this section, the uniform interleaver of size 1008×10 is chosen, unless otherwise specified.

4.4.2 Effects of Pilot Symbol Spacing

Consider binary codes with BPSK modulation. The effects of pilot symbol spacing B on the BEP performance are investigated at two SNRs at the normalized fade rates of $f_d T_s = 0.005$ and $f_d T_s = 0.02$, as shown Fig. 4.3(a). It is observed that as pilot spacing B increases, the BEP performance improves at low values of B .

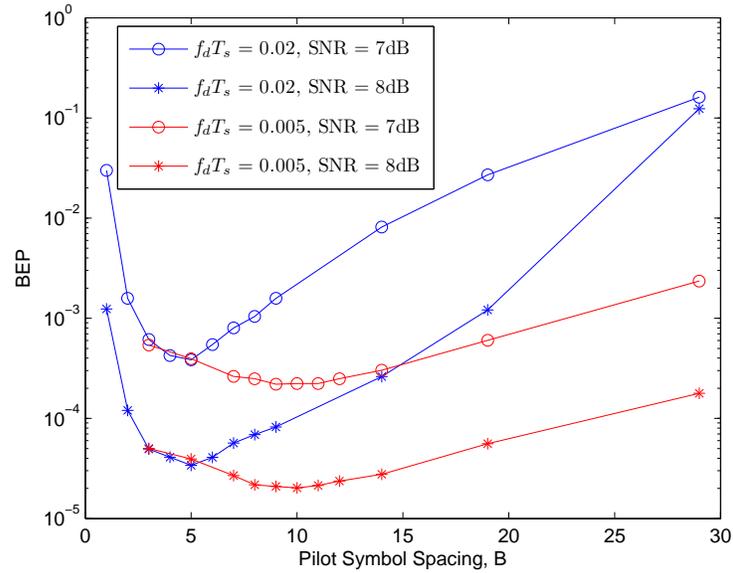
This is because the proportion of energy devoted to the pilot symbols decreases as B increase, and thus, the energy per data signal E_s increases, giving rise to the improvement in the BEP.

As B increases further, the pilot signals are further away from the interval where the channel gain is to be estimated. The correlation between them becomes weaker. Moreover, the number of pilot signals within the observation window also decreases. For these two reasons, the channel estimates become less accurate when B increases. Since the pilot spacing B is large, the increase in energy per data symbol E_s when B increases further becomes negligibly small. The decrease in channel estimation accuracy becomes the dominating factor and causes degradation in the BEP performance. The optimal performance is achieved at $B = 5$ at $f_d T_s = 0.02$, and in the range between 9 and 11 at $f_d T_s = 0.005$.

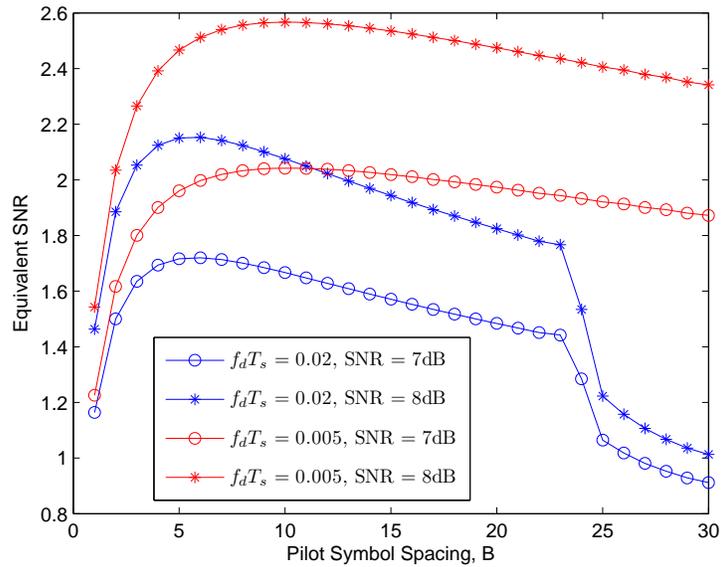
Alternative, the optimal value of the pilot symbol spacing can be approximately determined by examining the PSAM-LLR metric in (4.18). We can roughly treat the multiplication factor $\frac{1}{\frac{E_s}{N_0} 2\tilde{\sigma}^2(l,b)+1}$ as the scaling down factor on the SNR. Hence, by considering the equivalent SNR,

$$\begin{aligned} \text{SNR}_{\text{equ}} &= \frac{1}{\frac{E_s}{N_0} 2\tilde{\sigma}^2(l,b) + 1} \frac{E_s}{N_0} \\ &= \frac{E_s}{2\tilde{\sigma}^2(l,b)E_s + N_0} \\ &= \frac{RBE_b}{2\tilde{\sigma}^2(l,b)RBE_b + (B+1)N_0} \end{aligned}$$

we plot the graph of the equivalent SNR against the pilot symbol spacing B , as shown in Fig. 4.3(b). It can be observed that the optimal pilot symbol spacing obtained in this way approximately agrees with that from simulations.



(a)



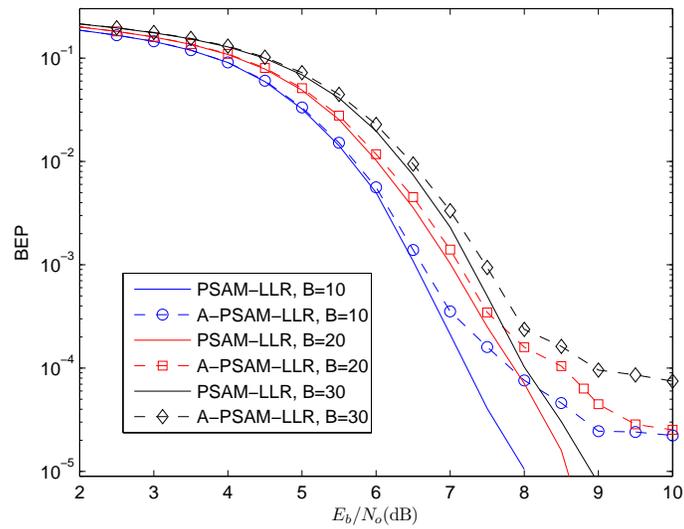
(b)

Figure 4.3: (a) Effects of pilot symbol spacing at normalized fade rates $f_d T_s = 0.005$ and $f_d T_s = 0.02$; (b) Equivalent SNR interpretation of the optimum pilot symbol spacing

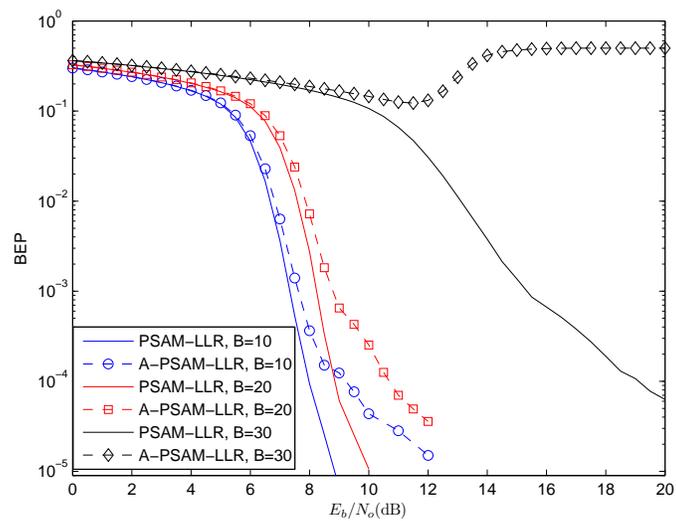
4.4.3 Standard BP Decoding with BPSK Modulation

In this section, we compare the performance the PSAM-LLR metric (4.18) and the A-PSAM-LLR metric (4.19) with BPSK modulated binary LDPC code transmitted at the normalized fade rates of $f_d T_s = 0.005$ and $f_d T_s = 0.02$, as shown in Figs. 4.4(a) and (b). It is observed that the A-PSAM-LLR is a good approximation to the PSAM-LLR at low SNRs. As the SNR increases, the PSAM-LLR achieves more significant performance gain over the A-PSAM-LLR. Noticeable error floors can be observed for the performance curves of the A-PSAM-LLR when BEP reaches 10^{-5} , while no error floor is observed in this region for the PSAM-LLR metric. At the higher fade rate, when the pilot symbols are placed far apart ($B = 30$), the channel estimates are very poor and the MSEs are quite large. The A-PSAM-LLR exceedingly over-estimates the channel LLR information when the SNR is high and the soft input to the BP decoder computed using A-PSAM-LLR metric is inaccurate. As a result, the BEP of the A-PSAM-LLR approaches 0.5, while the PSAM-LLR still gives reasonable performance. To illustrate how the PSAM-LLR improves the error floor performance, we choose an irregular (1008, 504) binary code, which has an error floor at a BEP of around 10^{-5} . It can be observed, from Figs. 4.5(a) and (b), that the error floor of the PSAM-LLR metric remains at the BEP of 10^{-5} , while that of the A-PSAM-LLR is approximately one decade higher at the fade rate $f_d T_s = 0.02$ and half decade higher at the fade rate $f_d T_s = 0.005$.

We also compare, in TABLE 4.1 and TABLE 4.2, the time complexity of

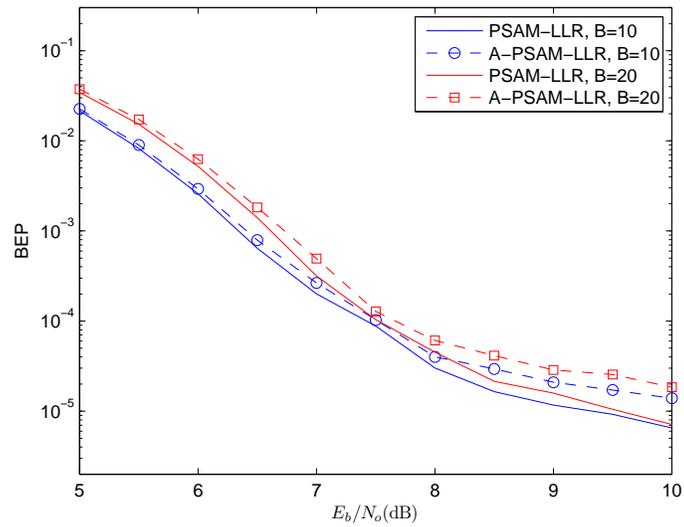


(a)

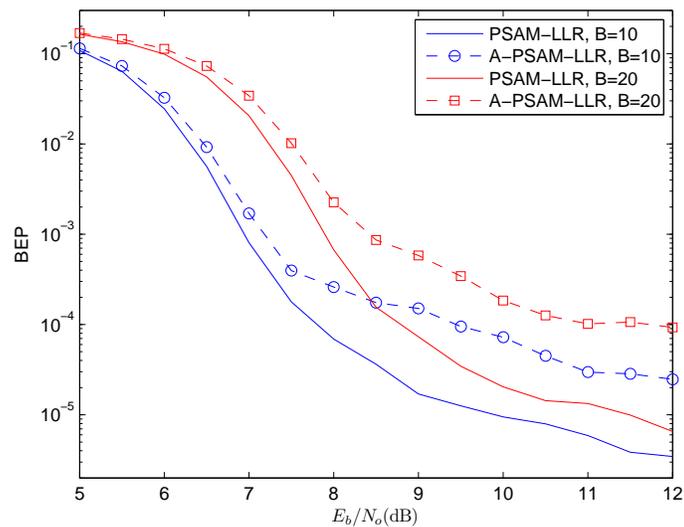


(b)

Figure 4.4: Performance comparison of LLR metrics under BPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.



(a)



(b)

Figure 4.5: Comparison of error floors with irregular codes under BPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

$\frac{E_b}{N_0}$	Average Number of BP Iterations					
	B=10		B=20		B=30	
	PSAM-LLR	A-PSAM-LLR	PSAM-LLR	A-PSAM-LLR	PSAM-LLR	A-PSAM-LLR
5.5	17.208	17.646	22.244	23.020	28.388	29.323
6.0	11.242	11.552	14.554	15.257	19.498	20.637
6.5	7.573	7.766	9.637	10.099	12.677	13.684
7.0	5.771	5.859	6.844	7.111	8.478	9.073

Table 4.1: Average number of BP iterations required for the PSAM-LLR metric and the A-PSAM-LLR metric over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.005$

the two metrics in term of the average number of BP iterations required for convergence. It is observed that the PSAM-LLR achieves faster convergence than the A-PSAM-LLR. At the fade rate $f_d T_s = 0.02$, when the SNR is 7.0dB and B is 20, the PSAM-LLR saves as many as 3.8 BP iterations, compared to the A-PSAM-LLR. Note that the MSE can be computed offline, and thus the PSAM-LLR metric is more efficient in processing long data streams.

4.4.4 Standard BP Decoding under QPSK and 8PSK Modulation

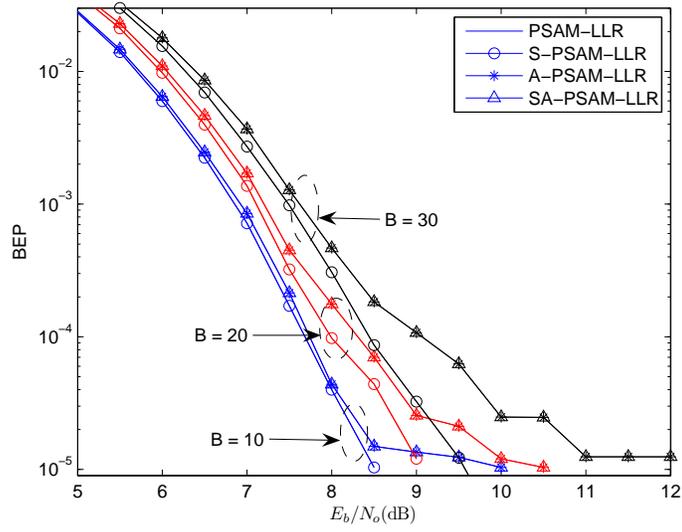
Figs. 4.6(a) and (b) show the performance comparison of the various LLR metrics using the binary LDPC code under QPSK modulation with Gray coding. Observe that the PSAM-LLR and the S-PSAM-LLR yield exactly the same performance.

$\frac{E_b}{N_0}$	Average Number of BP Iterations			
	B=10		B=20	
	PSAM-LLR	A-PSAM-LLR	PSAM-LLR	A-PSAM-LLR
7.0	12.123	13.834	30.980	34.837
7.5	7.903	8.582	18.904	23.218
8.0	6.060	6.282	11.053	13.532
8.5	5.024	5.107	7.400	8.305

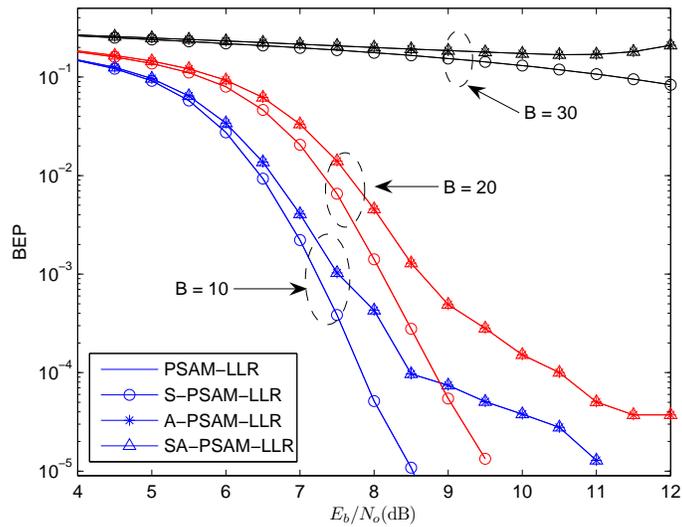
Table 4.2: Average number of BP iterations required for the PSAM-LLR metric and the A-PSAM-LLR metric over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.02$

In the Appendix, we prove that, for binary codes, the PSAM-LLR (resp., the A-PSAM-LLR) equals to the S-PSAM-LLR (resp., the SA-PSAM-LLR) under Gray coded QPSK. For 8PSK modulation, as shown in Figs. 4.7(a) and (b), the PSAM-LLR and the S-PSAM-LLR yield similar BEP performance, so the S-PSAM-LLR can be used as a simplified alternative to the PSAM-LLR metric. Likewise, the A-PSAM-LLR and the SA-PSAM-LLR are alternatives of each other.

Similar to the BPSK case, for both QPSK and 8PSK, the A-PSAM-LLR metric, which ignores the channel estimation accuracy, has high error floors and performs significantly worse than the PSAM-LLR metric, especially in high SNR regions.

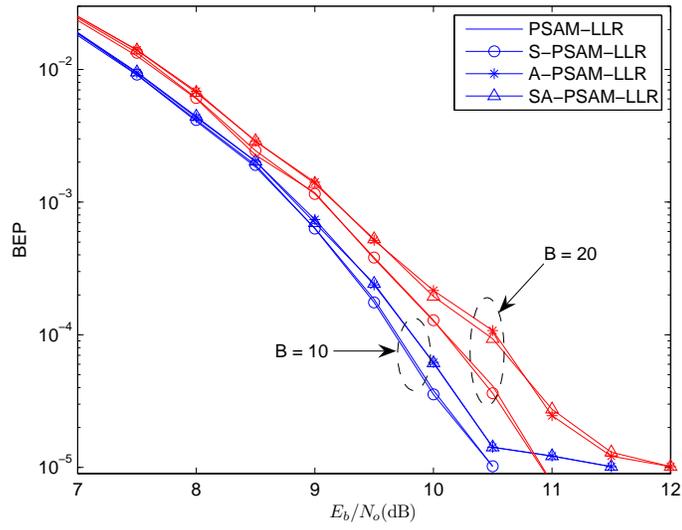


(a)

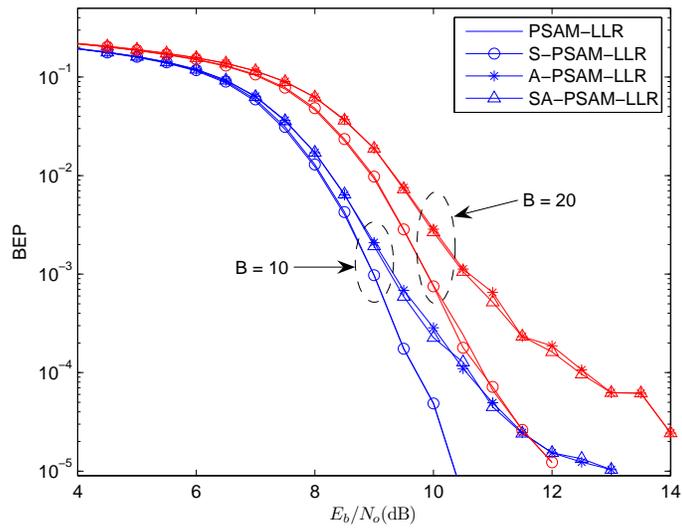


(b)

Figure 4.6: Performance comparison of LLR metrics under QPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.



(a)



(b)

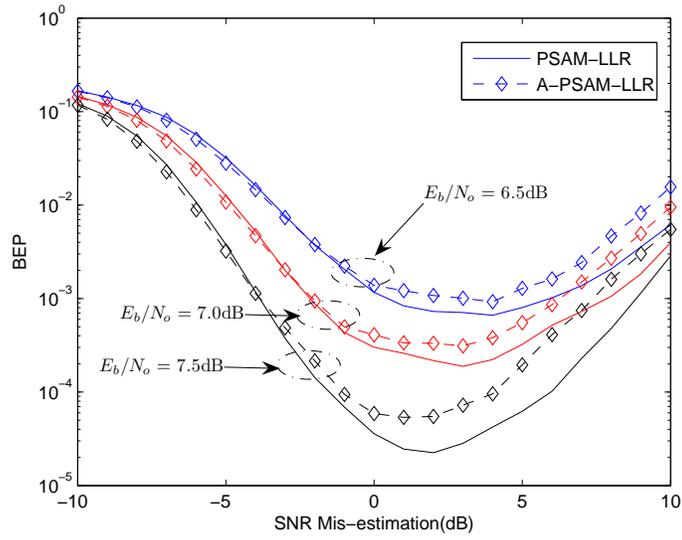
Figure 4.7: Performance comparison of LLR metrics under 8PSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

4.4.5 Effects of SNR Estimation Error

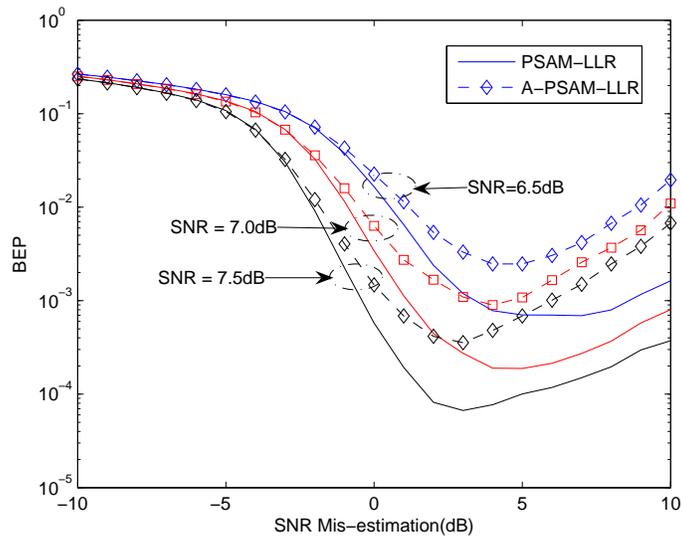
The computation of the PSAM-LLR metric requires the knowledge of the SNR at the receiver. Figs. 4.8(a) and (b) demonstrate the effect of SNR estimation errors for BPSK modulated binary LDPC code transmitted at the normalized fade rates of $f_d T_s = 0.005$ and $f_d T_s = 0.02$, respectively. The horizontal axis indicates the estimation errors (in dB), with positive values representing over-estimation, and negative values, under-estimation. It can be observed that the PSAM-LLR metric performs better than the A-PSAM-LLR metric under SNR estimation errors. For both metrics, under-estimation results in detrimental degradation in the BEP performance and an over-estimation of 2 – 5 dB tends to improve the performance, but the performance drops very quickly if the SNR is over-estimated further, with the A-PSAM-LLR deteriorating much faster than the PSAM-LLR.

4.4.6 Space Diversity with Multiple Receive Antennas

Antenna diversity technique is effective in mitigating the fading effect to improve the quality and reliability of a wireless link. When a message signal is transmitted through one transmit antenna and received by M receive antennas, M diversity channels are formed, all of which carry the same information-bearing signal. If it is assumed that the receive antennas are spaced sufficiently far apart such that the multipath components in the signal have significantly different propagation delays at the receive antennas, the M channels can be assumed to be mutually



(a)



(b)

Figure 4.8: Robustness comparison between PSAM-LLR and A-PSAM-LLR subjected to SNR mis-estimation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

statistically independent. Consequently, it can be derived that the LLR metric for this single-input multiple-output system is equal to the summation of the LLRs from all the M diversity channels, each of which is computed independently based on the received signal sequence from the corresponding diversity channel. As the number of receive antennas or the order of diversity M increases, the probability that a transmitted symbol is received in deep fade, decreases and thus the detection is more reliable.

In this section, we consider the PSAM-LLR metric and study the diversity effect by assuming that all the diversity channels have the same SNR and experience independent fading with identical spectrum. The error performance for diversity reception of different orders is plotted against the total mean received SNR per message bit, $\bar{\gamma}_b$, given by

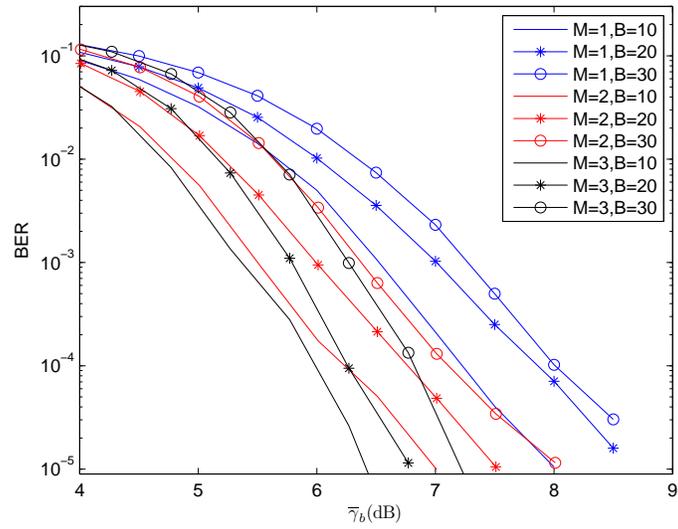
$$\bar{\gamma}_b = \frac{E_b}{N_0} \sum_{m=1}^M \mathbb{E}[c_m^2],$$

as shown in Figs. 4.9(a) and (b), where $\mathbb{E}[c_m^2]$ is the average power gain of the m -th channel. The plots clearly illustrate the advantage of diversity reception in overcoming the performance degradation caused by fading. However, when the order of diversity exceeds a certain limit, increasing the diversity order may have detrimental effect on the performance. This is because at a given total mean received SNR $\bar{\gamma}_b$, the received SNR per branch, i.e., $\frac{\bar{\gamma}_b}{M}$, decreases as the number of receive antennas increases. As a result, the energy devoted to the pilot signals drops and the accuracy of channel estimation in each branch decreases. When the diversity order is so high that the pilot signals do not have sufficient energy for

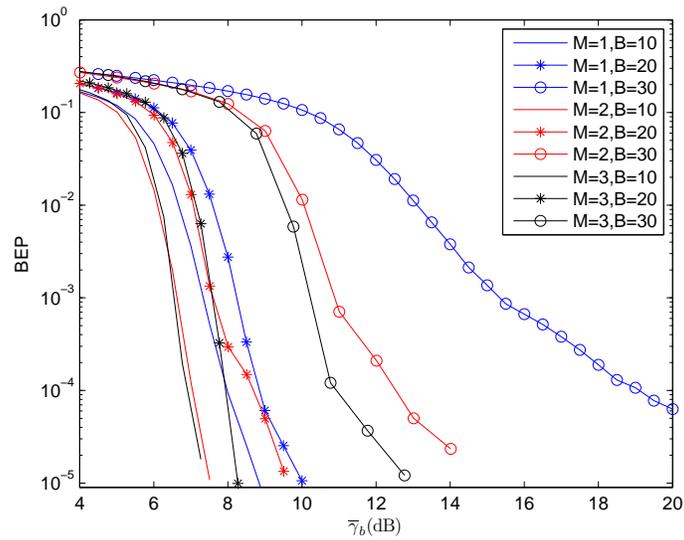
channel estimation, the performance will get worse. The incapability of obtaining reliable channel estimates can even outweigh the advantage of diversity reception in minimizing the effect of deep fade. It can be observed, in Fig. 4.9(b), that when the $\bar{\gamma}_b$ is less than 6.5 dB, the error performance when $M = 2$ is better than that when $M = 3$ for the case $B = 10$ at normalized fade rate $f_d T_s = 0.02$.

4.4.7 Iterative Channel Estimation and Decoding

In this section, we consider a two-stage process of iterative channel estimation and decoding with hard decision feedback for the binary LDPC code under BPSK modulation. The decoding process is described as follows. Suppose that the A-PSAM-LLR metric is used. In the first stage, the LLR values are computed using this metric from only the pilot signals within an observation window of length $W = 801$. The hard-decision tentative code-bit estimates are taken after 10 BP iterations. These hard-decision estimates are reinterleaved and pilot symbols are reinserted. Assuming this resulting symbol sequence is equal to the actual transmitted sequence, which means that all signals can be interpreted as pilot signals, at the second stage, the channel gains are re-estimated using the received signals from a shorter observation window of length $\bar{W} = 81$. New LLR values are obtained using the same metric. A maximum of 50 BP iterations are performed to get the final decision. If the PSAM-LLR metric is used, the process is the same as described above except that at each stage, the channel estimation MSE is calculated in addition, and the LLRs are computed using the PSAM-LLR metric.



(a)



(b)

Figure 4.9: Effects of diversity for the PSAM-LLR over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

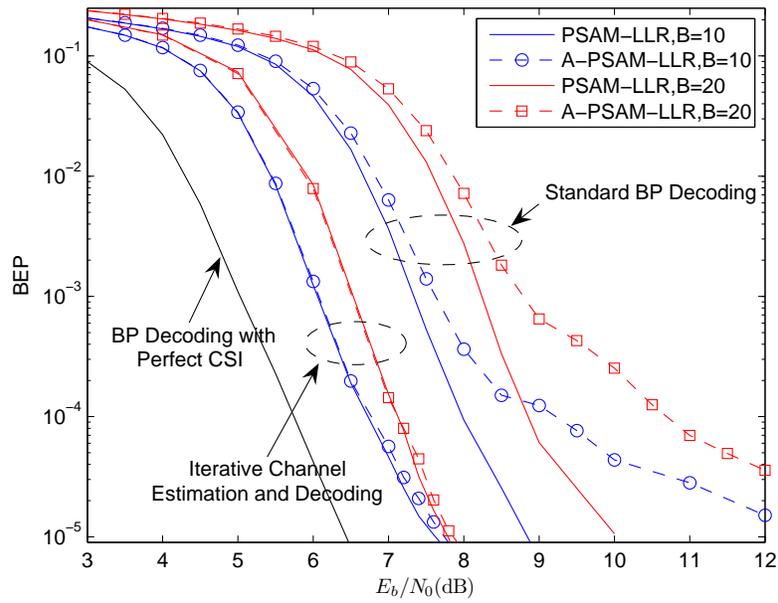


Figure 4.10: BEP performance comparison of iterative channel estimation and decoding and standard BP decoding under BPSK modulation over Rayleigh fading channels at normalized fade rate $f_d T_s = 0.02$

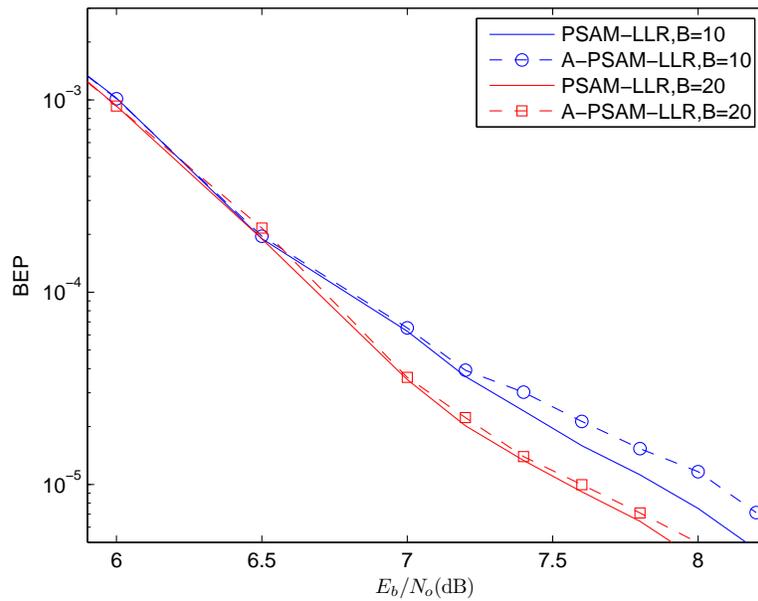


Figure 4.11: Performance comparison of iterative channel estimation and decoding with different LLR metrics under BPSK modulation over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.005$

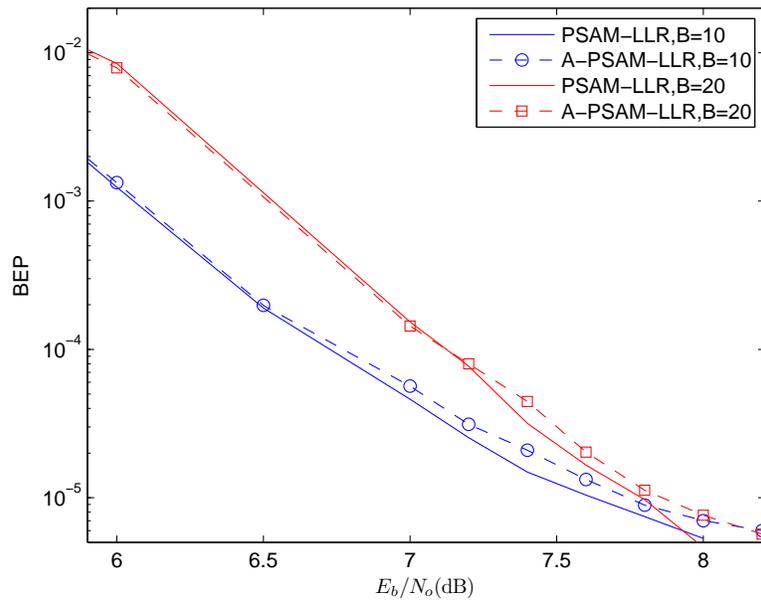


Figure 4.12: Performance comparison of iterative channel estimation and decoding with different LLR metrics under BPSK modulation over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.02$

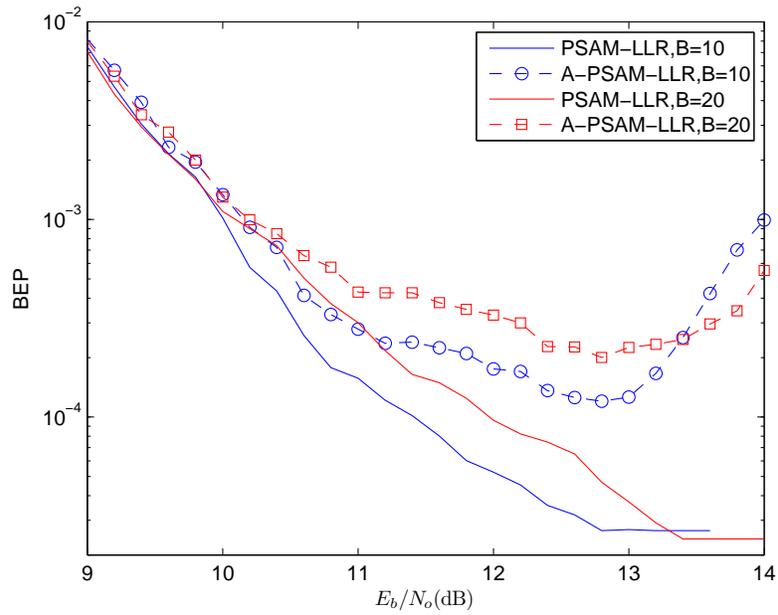


Figure 4.13: Performance comparison of iterative channel estimation and decoding with different LLR metrics under BPSK modulation over Rayleigh fading channels with normalized fade rate $f_d T_s = 0.05$

Fig. 4.10 shows that this two-stage process of iterative channel estimation and decoding achieves a BEP performance gain of around 1.5–2 dB over the standard BP decoding at $f_d T_s = 0.02$. The performance comparisons of the iterative channel estimation and decoding between the PSAM-LLR and A-PSAM-LLR with BPSK modulation are shown in Fig. 4.11, Fig. 4.12 and Fig. 4.13 at normalized fade rate of $f_d T_s = 0.005$, $f_d T_s = 0.02$ and $f_d T_s = 0.05$, respectively. It is observed that the PSAM-LLR outperforms the A-PSAM-LLR in the high SNR region, especially when the fade rate is high. The reasons are as follows. The tentative decisions after the initial ten iterations using the PSAM-LLR metric are more accurate than those using the A-PSAM-LLR metric. When these tentative decisions are fed back, more accurate channel re-estimates and more reliable soft information are obtained for the PSAM-LLR metric, and thus the BEP performance is better. As the normalized fade rate increases, the channel estimates are less accurate and the estimation error variance $2\tilde{\sigma}_m^2(l, b)$ increases. The term $\frac{1}{\frac{E_s}{N_0} 2\tilde{\sigma}^2(l, b) + 1}$ becomes much less than unity, especially when the SNR is high. The LLRs obtained from the A-PSAM-LLR metric are thus less accurate. As a result, the difference in the accuracies of the tentative decisions between the PSAM-LLR metric and the A-PSAM-LLR metric at the first stage of the iterative channel estimation decoder becomes larger as the fade rate increases. This explains why the performance gap between the two metrics increases with the increase of the fade rate.

Similar observations can be made from the two-stage process of iterative channel estimation and decoding for QPSK modulation, as shown in Figs. 4.14(a) and

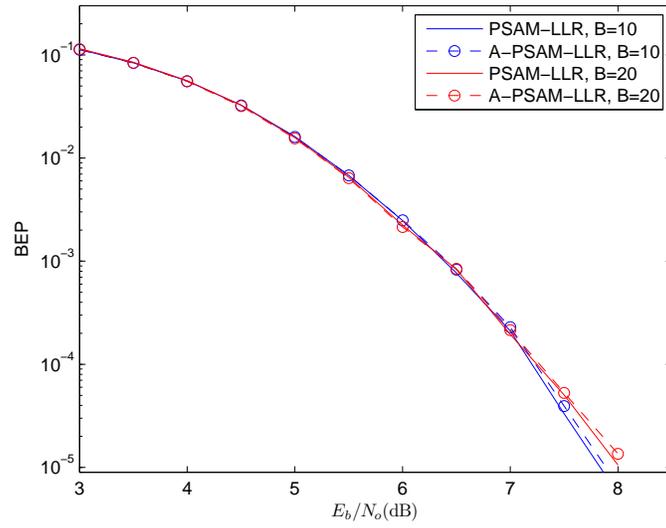
(b), and for 8PSK modulation, as shown in Figs. 4.15(a) and (b).

4.4.8 Quaternary Codes with QPSK Modulation

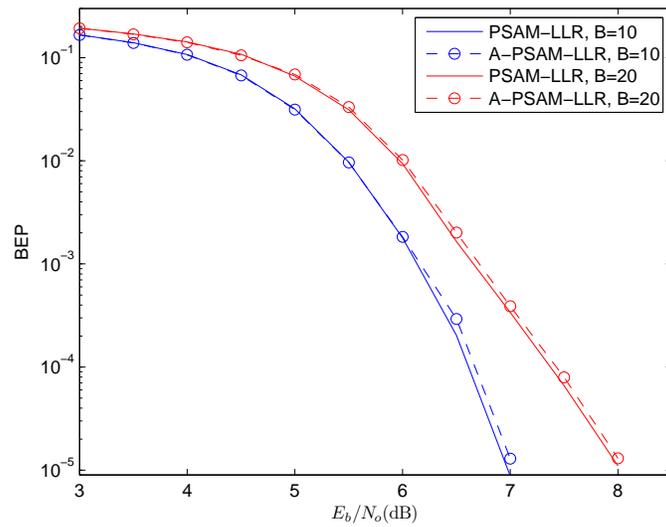
In this section, the standard BP decoding of the quaternary codes with non-Gray coded QPSK modulation is considered. Figs. 4.16(a) and (b) show the comparison between the PSAM-LLR and the A-PSAM-LLR for the \mathbb{GF}_4 code. It is observed that significant performance gain is achieved by the PSAM-LLR metric over the A-PSAM-LLR metric. Error floors are observed at the BEP of around 10^{-5} when the A-PSAM-LLR metric is used, while they are not observed for the PSAM-LLR metric. Similar observations can be made for the \mathbb{Z}_4 code, as shown in Figs. 4.17(a) and (b). The performance of binary code, \mathbb{GF}_4 code and \mathbb{Z}_4 code under QPSK modulation is compared in Figs. 4.18(a) and (b). It is observed that the BEP performance of the \mathbb{Z}_4 code is better than that of the \mathbb{GF}_4 , both of which exceed that of the binary code. For each of the three codes, the PSAM-LLR metric always outperforms the A-PSAM-LLR metric. In general, nonbinary codes can be used with MPSK modulations to improve the BEP performance, especially when our PSAM-LLR metric is used.

4.5 Conclusion

In this chapter, we have derived the PSAM-LLR metric and several approximations for an MPSK-modulated PSAM system over time-selective, frequency-flat, Rayleigh fading channels. We have also shown that the PSAM-LLR metric, which

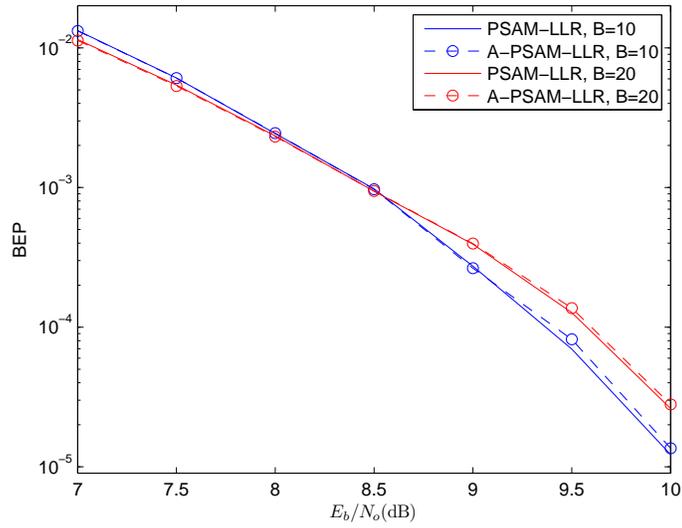


(a)

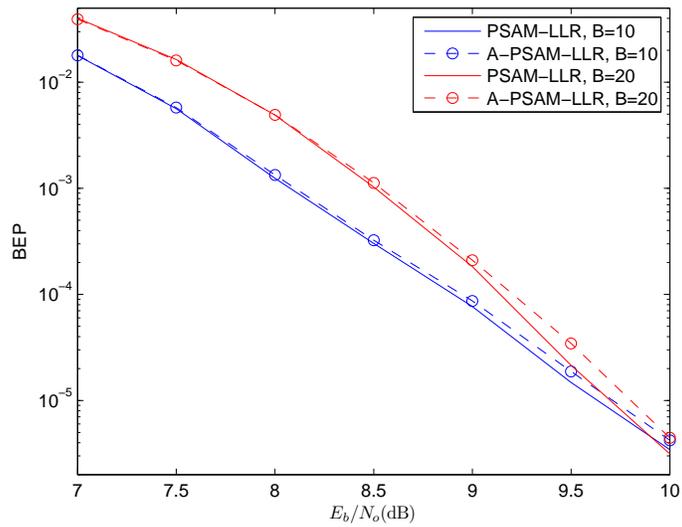


(b)

Figure 4.14: Performance comparison of iterative channel estimation and decoding with different LLR metrics under QPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

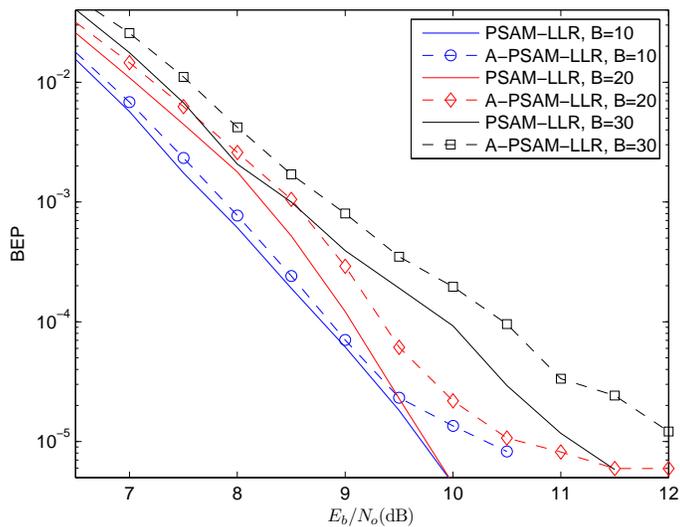


(a)

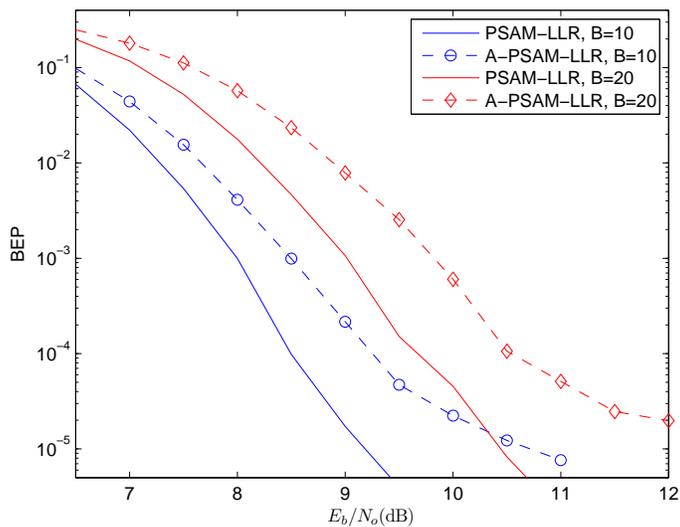


(b)

Figure 4.15: Performance comparison of iterative channel estimation and decoding with different LLR metrics under 8PSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

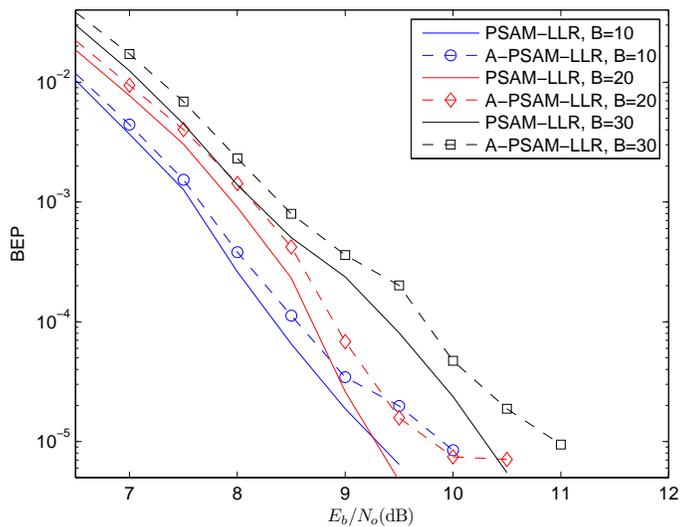


(a)

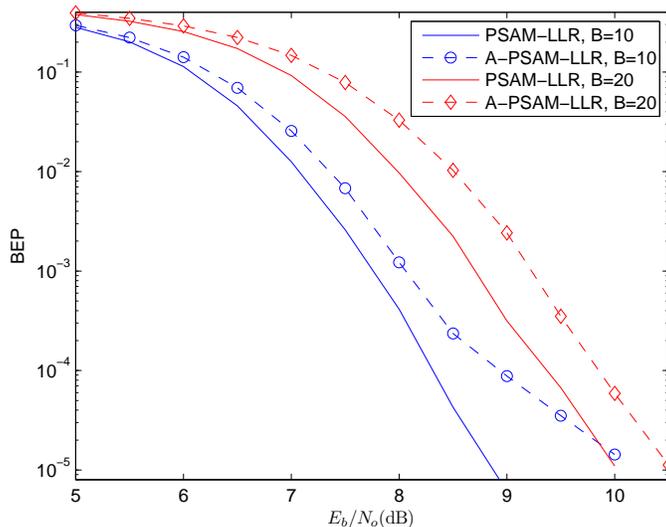


(b)

Figure 4.16: Performance comparison between PSAM-LLR and A-PSAM-LLR with \mathbb{GF}_4 code under QPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

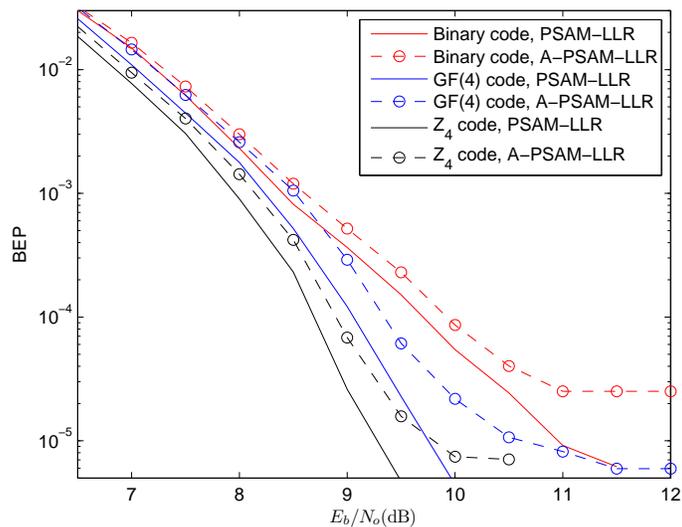


(a)

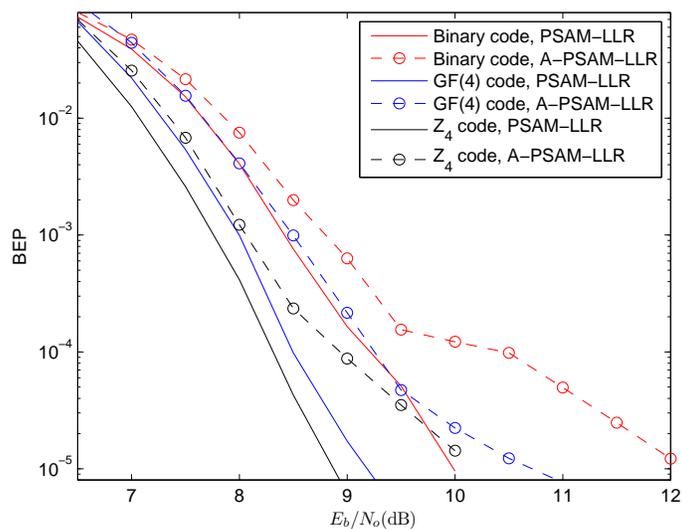


(b)

Figure 4.17: Performance comparison between PSAM-LLR and A-PSAM-LLR with \mathbb{Z}_4 code under QPSK modulation over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.



(a)



(b)

Figure 4.18: Performance comparison between PSAM-LLR and A-PSAM-LLR under QPSK modulation for binary code, \mathbb{GF}_4 code and \mathbb{Z}_4 code over Rayleigh fading channels with various normalized fade rates and pilot symbol spacings: (a) $f_d T_s = 0.005$ with pilot spacing $B = 20$; (b) $f_d T_s = 0.02$ with pilot spacing $B = 10$.

takes into account the information concerning the channel estimation accuracy, achieves better BEP performance and faster convergence speed. We note that the PSAM-LLR metric relies on the accuracy of the computation of the estimation MSE, which is quite complicated in some cases (e.g., when soft decision feedback is used). These problems will be investigated in our future studies.

It is important to note also that the PSAM-LLR is derived based on the assumption that perfect KCS is available at the receiver. When this information is not available, the estimation MSE cannot be computed and the PSAM-LLR metric cannot be used. In this case, the receiver with PSAM channel estimation may not have good performance.

Chapter 5

The LLR Computation via SOVA with Implicit CSI

In the previous chapter, we discussed the importance of incorporating the knowledge of the channel estimation accuracy into the iterative decoding process and pointed out that significant performance loss will be incurred if this information is neglected. The discussions on the PSAM-LLR metric and the role of the channel estimation accuracy rely on the assumption of perfect KCS at the receiver. However, in practice, it could be very complicated and computationally costly to obtain it accurately, especially when the channel statistics varies with time. When the wrong channel model is used or the parameters that define the autocorrelation function are measured wrongly, the PSAM receiver will suffer from severe performance degradation.

In this chapter, we will consider the case when KCS is not available at the

receiver. Here, KCS refers to the channel fading characteristics only, which includes, in particular, the fading model and the parameters associated with it. The spectrum density of the AWGN is known. Since the explicit channel estimates and the channel estimation accuracy cannot be obtained accurately, we consider the MLSD [70] approach via trellis search, which does not require KCS. We will incorporate convolutional codes into the MLSD to increase the relative Hamming distance between contending paths and thus improve the reliability of the decisions.

Since the MLSD can be combined perfectly with the SOVA [75], which is an ML based decoder, we propose an algorithm based on SOVA, which produces a reliability indicator together with the hard-decision output for each bit decision in the MLSD. We will present how the soft information, in term of LLR, should be computed based on MLSD with convolutional codes via SOVA over block-wise static Rayleigh fading channels with unknown channel statistics, and demonstrate that the LDPC coded transmissions can be more reliably recovered using iterative decoding with the obtained LLRs. The algorithm is developed based on the maximum-likelihood sequence detection, which does not require explicit channel estimation. For each survivor path, an implicit channel estimate is obtained from the received signals and the tentative decisions of the path. For these reasons, we name the algorithm as *SOVA with implicit CSI* (SOVA-ICSI). We will show that the LLR output of the SOVA-ICSI can be computed solely based on the received signal sequence, which does not require KCS. Compared with the systems with differential detection [39–42] and the PSAM systems [56–65], all of which

require the precise KCS, the SOVA-ICSI detector is more robust and much less demanding of knowledge of the channel, and thus it can be used more widely in real applications.

Through computer simulations, we show that the iterative decoding of LDPC codes with the SOVA-ICSI has substantially better BEP performance and stronger robustness against signal-to-noise ratio mis-estimation than that with conventional SOVA with PSAM channel estimation. Moreover, the SOVA-ICSI is more conveniently used for iterative channel estimation and decoding (ICED). Since the channel refinement is automatically carried out when the updated soft information is fed back to the SOVA-ICSI detector as the input, external channel estimation structures are not required and thus no extra computational power is required for the refinement of the channel estimates. Simulation studies show that the BEP performance of ICED via SOVA-ICSI is much better than that of the conventional ICED with hard-decision feedback. We will start this chapter by introducing the system model.

5.1 System Model

The system model is shown in Fig. 5.1. A binary message sequence \mathbf{m} is first encoded by the outer LDPC encoder. The encoded sequence is then shuffled by a block interleaver. The interleaved sequence $\mathbf{u}' = [u'(1), u'(2), \dots, u'(qK)]$ is sequentially divided into q blocks of length K , each of which is individually encoded by a convolutional encoder of rate $R = 1/n$ and constraint length v .

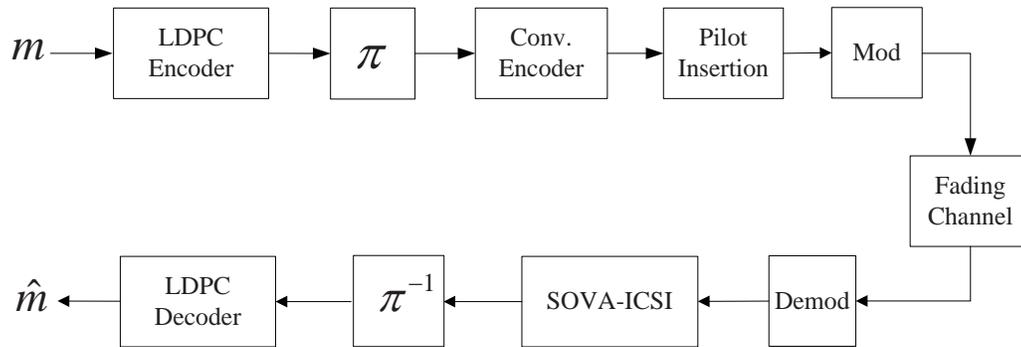


Figure 5.1: System model of the receiver with SOVA-ICSI

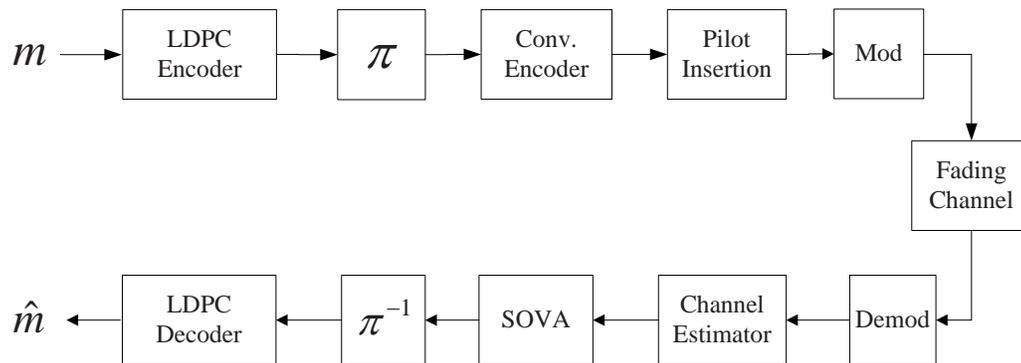


Figure 5.2: System model of the receiver with SOVA-PSAM

Note that for the convolutional encoding of each block, additional $v - 1$ input bits are required to flush the memory. It will be more convenient if we consider the sequence $\mathbf{u} = [u(1), u(2), \dots, u(q(K + v - 1))]$, which is formed inserting all the flushing bits into the message sequence \mathbf{u}' , as the input to the convolutional encoder, where $u(i(K + v - 1) + j) = u'(iK + j)$, for $0 \leq i \leq q, 0 \leq j \leq K$ and $\{u(i(K + v - 1) + j)\}_{0 \leq i \leq q, K < j \leq K + v - 1}$ are the flushing bits. The encoded sequence is denoted by $\mathbf{x} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(k), \dots, \mathbf{x}(q(K + v - 1))]$, where $\mathbf{x}(k) = [x^{(1)}(k), x^{(2)}(k), \dots, x^{(n)}(k)]$ and $x^{(i)}(k)$ is the output from the i -th output branch of the convolutional encoder. We further assume that the *a priori* LLR of the bit $u(k)$ is given by $L_a(k) = \log \frac{P(u(k)=0)}{P(u(k)=1)}$.

The coded sequence \mathbf{x} is modulated using BPSK. The baseband data signal sequence is denoted by $\mathbf{s} = [s^{(i)}(k)]_{1 \leq i \leq n, 1 \leq k \leq q(K + v - 1)}$, and $s^{(i)}(k) = \sqrt{E_s} \exp(j\pi x^{(i)}(k))$, where E_s denotes the energy of each modulated signal. Pilot symbols (signals) are inserted periodically into the data signal sequence with pilot symbol spacing B . For simplicity, we assume each pilot symbol takes bit value 0 and energy E_s , i.e., the baseband pilot signal has the form $\sqrt{E_s} \exp(j\pi 0)$.

The signal sequence is transmitted through a frequency-flat quasi-static or block-wise static fading channel. Assume the channel gains remain constant within every interval consisting of W consecutive transmitted signals, which is referred to as a *fading block*, and W is called the *fading block length*.

Denote the received data signal sequence by $\mathbf{r} = [r^{(i)}(k)]_{1 \leq i \leq n, 1 \leq k \leq q(K + v - 1)}$, and each of the received signals can be expressed as

$$r^{(i)}(k) = h^{(i)}(k)s^{(i)}(k) + n^{(i)}(k), \quad (5.1)$$

where $h^{(i)}(k)$ is the channel gain experienced during the transmission of $x^{(i)}(k)$, and $n^{(i)}(k)$ is the complex AWGN with mean zero and variance N_0 .

The SOVA-ICSI detector is used to decode the convolutional code and generate soft output for each LDPC encoded bit. After deinterleaving, the soft information is passed to the BP decoder to obtain the estimated message $\hat{\mathbf{m}}$.

For comparison, the system model for the conventional SOVA with PSAM channel estimation, which is referred to as the *SOVA-PSAM*, is shown in Fig. 5.2. Note that the same transmitter structure is used and the same fading channel is assumed. At the receiver, the channel gain for each received data signal is estimated from the pilot symbols, using a moving average channel estimator [59]. Assuming that the estimated channel gain is perfect, the conventional SOVA is used to decode the inner convolutional code. The soft output, after deinterleaving, is fed to the LDPC decoder to obtain message estimates. Note that since KCS is not available, the MMSE estimates cannot be computed at the receiver. Based on the information available, we can only use the moving average estimator for channel estimation in the PSAM scheme and assume that the channel estimates are equal to the real channel gains.

5.2 Metric Derivation

At the receiver, the SOVA-ICSI detector is used to generate soft decisions from the received signals, as shown in Fig. 5.1. The SOVA-ICSI is based on the SOVA, whereby a survivor is selected by comparing the reliability metrics of the

contending paths when they merge into the same state on the trellis. We will first compute the reliability metric for the SOVA-ICSI detector.

Suppose that the tentative decisions on a hypothesized path are given by $\mathbf{u}_{[1,t]} = [u(1), u(2), \dots, u(t)]$, where $t \leq qK$, and the transmitted signal sequence corresponding to $\mathbf{u}_{[1,t]}$ is denoted by $\mathbf{s}_{[1,t]}$. We will compute the ML probability density function for the path segment containing L tentative decisions from $t' = t - L + 1$ to t , and use it as a measure of the reliability of this hypothesized path. Here, the reliability of a path segment is used to represent that of the entire path and we will justify this later. We call L the *window length* of the SOVA-ICSI.

For the ease of representation, let $\mathbf{v} = \mathbf{u}_{[t',t]} = [u(t'), u(t' + 1), \dots, u(t)]$, where $t' = \max\{t - L + 1, 1\}$, and denote the transmitted signal vector corresponding to \mathbf{v} by $\mathbf{s}(\mathbf{v})$. Note that \mathbf{v} is one segment of the path $\mathbf{u}_{[1,t]}$. Given $\mathbf{u}_{[1,t'-1]}$, the segment \mathbf{v} and the transmitted signal vector $\mathbf{s}(\mathbf{v})$ are of one-to-one correspondence. We assume that \mathbf{v} is only a short segment, such that the transmitted data signals in $\mathbf{s}(\mathbf{v})$ are all inside the fading block centered at the transmission interval of $s^{(1)}(t)$, which is the transmitted signal corresponding to the code bit from the first branch of the convolutional encoder at time t . Let $\mathbf{r}(\mathbf{v})$ the vector of the received data signals corresponding to $\mathbf{s}(\mathbf{v})$, and let $\mathbf{r}_p(t)$ denote the vector formed by all the received pilot signals within this fading block. For the ease of presentation, we will simply use \mathbf{s} , \mathbf{r} and \mathbf{r}_p to denote $\mathbf{s}(\mathbf{v})$, $\mathbf{r}(t)$ and $\mathbf{r}_p(\mathbf{v})$, respectively, in this section.

Since the received signals \mathbf{r} and \mathbf{r}_p are inside the same fading block, they experience the same fading, denoted by h . Since KCS is not available at the

receiver, h is assumed to be an unknown constant. Following [70], the probability density function is maximized with respect to the channel gain h , as

$$\max_h p(\mathbf{v}, \mathbf{r}, \mathbf{r}_p | h),$$

Using Bayes' rule, we get

$$p(\mathbf{v}, \mathbf{r}, \mathbf{r}_p | h) = p(\mathbf{r}, \mathbf{r}_p | \mathbf{v}, h) P(\mathbf{v} | h). \quad (5.2)$$

Since the *a priori* probability of the message \mathbf{v} is independent of the parameter h , we have

$$P(\mathbf{v} | h) = P(\mathbf{v})$$

Conditioned on the channel gain and the transmitted signal sequence, each of the received signals is Gaussian distributed and statistically independent of one another. Substituting the probability density function of $p(\mathbf{r}, \mathbf{r}_p | h, \mathbf{v})$ into (5.2), we obtain

$$p(\mathbf{v}, \mathbf{r}, \mathbf{r}_p | h) = \frac{1}{(\pi N_0)^J} \exp\left(-\frac{\|\mathbf{r} - h\mathbf{s}\|^2 + \|\mathbf{r}_p - h\sqrt{E_s}\mathbf{I}\|^2}{N_0}\right) P(\mathbf{v}), \quad (5.3)$$

where \mathbf{I} is a vector of the same length as \mathbf{r}_p with each entry equal to 1, and J is the total number of received signals in \mathbf{r} and \mathbf{r}_p .

From [70], for a given hypothesis \mathbf{v} (or equivalently \mathbf{s}), the optimum value of the channel gain, obtained by maximizing (5.3), is given by

$$\hat{h}(\mathbf{s}) = \frac{\mathbf{s}^H \mathbf{r} + \sqrt{E_s} \mathbf{I}^H \mathbf{r}_p}{\|\mathbf{s}\|^2 + E_s \|\mathbf{I}\|^2} \quad (5.4)$$

Substitute (5.4) into (5.3), and after simplification, the likelihood function is written as a function of \mathbf{v} , as

$$\begin{aligned} \max_h p(\mathbf{v}, \mathbf{r}, \mathbf{r}_p | h) &= p(\mathbf{v}, \mathbf{r}, \mathbf{r}_p | \hat{h}(\mathbf{s})) \\ &= \frac{1}{(\pi N_0)^J} \exp\left(-\frac{1}{N_0}(\|\mathbf{r}\|^2 + \|\mathbf{r}_p\|^2 - \frac{|\mathbf{s}^H \mathbf{r} + \sqrt{E_s} \mathbf{I}^H \mathbf{r}_p|^2}{\|\mathbf{s}\|^2 + E_s \|\mathbf{I}\|^2})\right) P(\mathbf{v}) \end{aligned}$$

Note that for a 0-1 random variable y ,

$$\log P(y = i) = \frac{1}{2} \cos(\pi i) \underbrace{\log \frac{P(y = 0)}{P(y = 1)}}_{L_a(k)} + C_y, \quad \text{for } i = 0, 1$$

where $C_y = \frac{1}{2}(\log P(y = 0) + \log P(y = 1))$ is independent of the realization of y . Hence, the term $P(\mathbf{v})$ can be written as the summation of the *a priori* LLR values as

$$P(\mathbf{v}) = \exp\left(\sum_{k=t'}^t \frac{1}{2} \cos(\pi u(k)) L_a(k) + C_{u(k)}\right)$$

Now, we define the reliability metric for the path segment \mathbf{v} as

$$\begin{aligned} M^*(\mathbf{v}) &= \log p(\mathbf{v}, \mathbf{r}, \mathbf{r}_p | \hat{h}(\mathbf{s})) \\ &= \underbrace{\left(\log \frac{1}{(\pi N_0)^J} - \frac{\|\mathbf{r}\|^2 + \|\mathbf{r}_p\|^2}{N_0} + \sum_{k=t'}^t C_{u(k)}\right)}_{\text{constant}} + \frac{|\mathbf{s}^H \mathbf{r} + \sqrt{E_s} \mathbf{I}^H \mathbf{r}_p|^2}{N_0(\|\mathbf{s}\|^2 + E_s \|\mathbf{I}\|^2)} \\ &\quad + \sum_{k=t'}^t \frac{1}{2} \cos(\pi u(k)) L_a(k). \end{aligned}$$

By cancelling the additive constant, we obtain the simplified metric as

$$M(\mathbf{v}) = \frac{|\mathbf{s}^H \mathbf{r} + \sqrt{E_s} \mathbf{I}^H \mathbf{r}_p|^2}{N_0(\|\mathbf{s}\|^2 + E_s \|\mathbf{I}\|^2)} + \sum_{k=t'}^t \frac{1}{2} \cos(\pi u(k)) L_a(k), \quad (5.5)$$

We refer to this metric in (5.5) as the *SOVA-ICSI metric*.

Next, we explain why the reliability metric of the selected segment provides a reasonably good measure of the reliability of the whole path. Consider two contending paths, denoted by $\mathbf{u}_{[1,t]} = [u(1), u(2), \dots, u(t)]$ and $\mathbf{u}'_{[1,t]} = [u'(1), u'(2), \dots, u'(t)]$, entering the same state, as shown in Fig. 5.3. If the two paths have converged before time t' , i.e., $u(k) = u'(k)$ for $k \leq t'$, as shown in Fig. 5.3(a), the difference between the two hypothesized input sequences $\mathbf{u}_{[1,t]} = [u(1), u(2), \dots, u(t)]$ and $\mathbf{u}'_{[1,t]} = [u'(1), u'(2), \dots, u'(t)]$ can be fully characterized by the segments $\mathbf{u}_{[t',t]}$ and $\mathbf{u}'_{[t',t]}$. Hence, the survivor path can be reliably determined by comparing the metrics $M(\mathbf{u}_{[t',t]})$ and $M(\mathbf{u}'_{[t',t]})$.

On the other hand, if the two paths have not converged at time t' , as shown in Fig. 5.3(b), the difference in the metrics $M(\mathbf{m}_{[t',t]})$ and $M(\mathbf{m}'_{[t',t]})$ can only provide a partial reliability measure in selecting the survivor path. We refer to this case as the *path divergence problem*. It can be shown that the probability of encountering the path divergence problem is small when the window length L is sufficiently large, and it decreases very rapidly with the increase of L [133, 134]. However, it is not desirable to choose large values of L . The reasons are as follows.

Firstly, the choice of L is restricted by the fading characteristics of the channel, since the transmitted signals $\mathbf{r}_{[t',t]}$ are required to be within the same fading block. Thus, the value of L is restricted by the fading block length of the channel. Secondly, it has been well known that the joint maximization of the channel and the data suffers from phase and divisor ambiguities [135]. In the SOVA-ICSI, the pilot signals are shared by the contending hypothesized sequences, and help

to minimize these ambiguities. If more data signals are considered in the joint estimation, more pilot signals should be used to minimize performance degradation caused by these ambiguities. Since pilots are the transmission overhead, which increase the transmission bandwidth, the number of pilot signals used in transmissions is usually limited. Hence, the value of L should be chosen based on the frequency of pilot insertion, in order to minimize performance loss due to phase and divisor ambiguities.

To minimize the occurrence of the path divergence problem, large window lengths are preferred. However, small values of window length L help resolve the phase and divisor ambiguities. With a deliberate choice of the window length L , we can strike a good balance and obtain more reliable soft decisions from the SOVA-ICSI. We will show, in section 5.5, that optimum value of L exists, and the SOVA-ICSI with optimum L always outperforms the SOVA-PSAM.

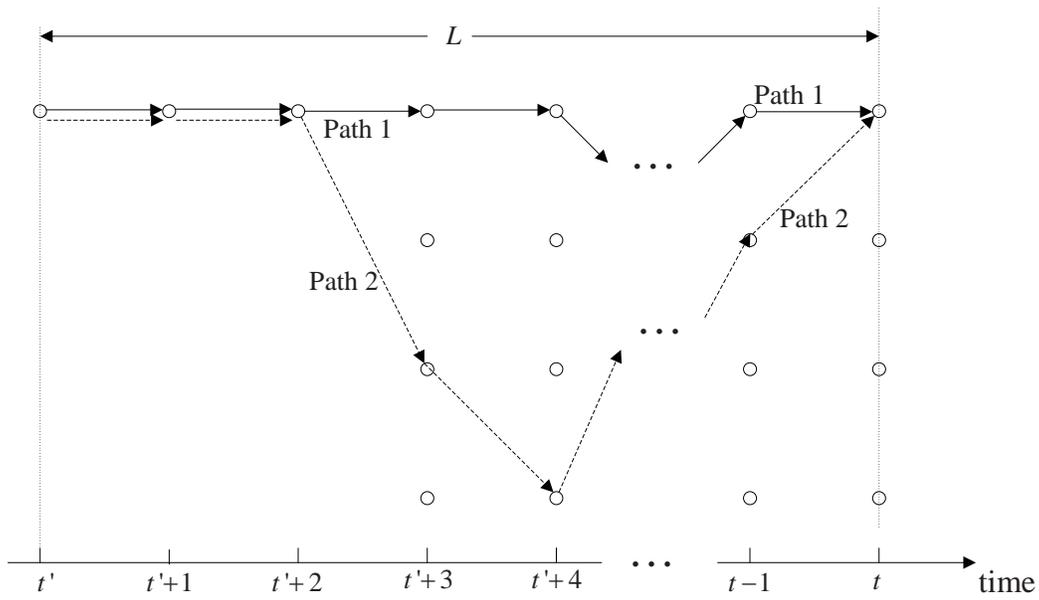
5.3 SOVA-ICSI

With the SOVA-ICSI metric in (5.5), the SOVA-ICSI is described as follows:

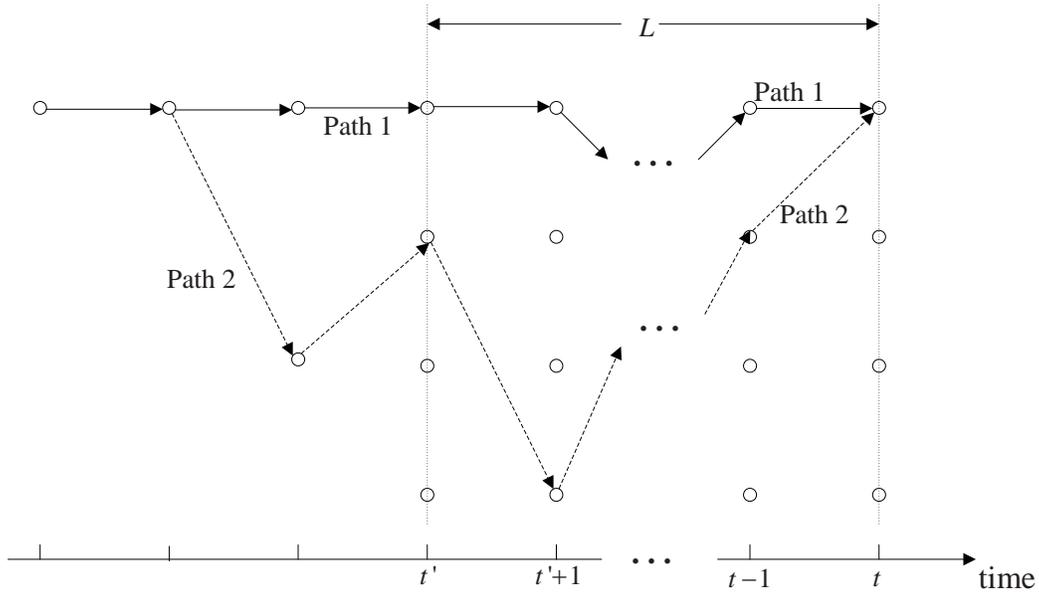
Step 1. Initialize the set of available states, $\mathbb{S} = \{S_0\}$, where S_0 is the all zero state. Initialize the survivor for state S_0 as $\mathbf{d}_0(S_0) = \emptyset$, and the reliability vector as $\mathbf{\Lambda}_0(S_0) = \emptyset$.

Step 2. Beginning from time $t = 1$, compute the set of current available states \mathbb{S}' from the previous states \mathbb{S} . For each state S_i in \mathbb{S}' , do the followings:

 Compute the path metric using (5.5) for each path enter the state S_i . If



(a)



(b)

Figure 5.3: Trellis diagrams illustrating the two situations encountered in the SOVA-ICSI

only one path enters S_i from the previous state S_j , store this path as the survivor and the reliability vector is updated as $\mathbf{\Lambda}_t(S_i) = [\mathbf{\Lambda}_{t-1}(S_j), \infty]$. If two paths enter the state S_i from the previous state S_j and $S_{j'}$, respectively, denote the tentative decisions for this two path as $\mathbf{u}_t = [u(1), u(2), \dots, u(t)]$ and $\mathbf{u}'_t = [u'(1), u'(2), \dots, u'(t)]$, and the corresponding path metrics as $M(\mathbf{u}_t)$ and $M(\mathbf{u}'_t)$. Select the path with the larger metric as the survivor. Without loss of generality, assume $\Delta(t) = M(\mathbf{u}_t) - M(\mathbf{u}'_t) > 0$ and hence \mathbf{u}_t is selected as the survivor. Suppose the reliability vector $\mathbf{\Lambda}_{t-1}(S_j) = [\lambda_{t-1}(1), \lambda_{t-1}(2), \dots, \lambda_{t-1}(t-1)]$, the reliability vector $\mathbf{\Lambda}_t(S_i) = [\lambda_t(1), \lambda_t(2), \dots, \lambda_t(t)]$ is updated as

$$\lambda_t(k) = \begin{cases} \min\{\Delta(t), \lambda_{t-1}(k)\}, & \text{if } u(k) \neq u'(k) \\ \lambda_{t-1}(k), & \text{if } u(k) = u'(k) \end{cases} \quad \text{for } k = 1, 2, \dots, t-1$$

$$\lambda_t(t) = \begin{cases} \Delta(t), & \text{if } u(t) \neq u'(t) \\ +\infty, & \text{if } u(t) = u'(t) \end{cases}$$

(The details of the reliability update can be found in [159, Section 12.5].)

Step 3 Increase t by 1 and update \mathbb{S} by \mathbb{S}' . If $t \leq q(K + v - 1)$, repeat step 2; otherwise, stop.

5.4 Iterative Channel Estimation and Decoding

The receiver structure for the conventional iterative channel estimation and decoding with hard decision feedback is shown in Fig. 5.4(a). After demodulation, the initial channel estimates are obtained from the pilot signals. Assuming that

the estimated channel gains are perfect and the *a priori* LLR λ_1^{in} is zero, the SOVA is used to decode the inner convolutional code. The soft output λ_1^{out} , is deinterleaved and then passed to the LDPC decoder. After a fixed number of iterations, which is referred to as the *internal-iteration*, the extrinsic information, which is obtained by subtracting the input LLR λ_0^{in} from the *a posteriori* LLR λ_0^{out} , is reinterleaved and fed back to the SOVA decoder as the new *a priori* information. At the same time, tentative hard decisions are determined based on λ_0^{out} , which are reinterleaved and convolutionally encoded. Assuming this encoded sequence is equal to the actual transmitted data sequence, which means that all the data signals can be interpreted as pilot signals, the channel gains are re-estimated. With new channel estimates and new *a priori* information, SOVA produces new LLR output λ_1^{out} . Again, the extrinsic information, obtained by $\lambda_1^{\text{out}} - \lambda_1^{\text{in}}$, is deinterleaved and passed to the LDPC decoder. This process is repeated for several *super-iterations*, before making the final decision. We refer to this decoding process as the *ICED-SOVA-PSAM*.

The receiver for the iterative channel estimation and decoding with the SOVA-ICSI detector is much simpler. As shown in Fig. 5.4(b), no feedback path is used since explicit channel estimation is not required. Similar to the ICED-SOVA-PSAM, the extrinsic information is exchanged between the LDPC decoder and the SOVA-ICSI decoder. We refer to this decoding process as the *ICED-SOVA-ICSI*.

Note that in each super-iteration, the extrinsic information is exchanged between the LDPC decoder and the SOVA-PSAM or the SOVA-ICSI decoder. This

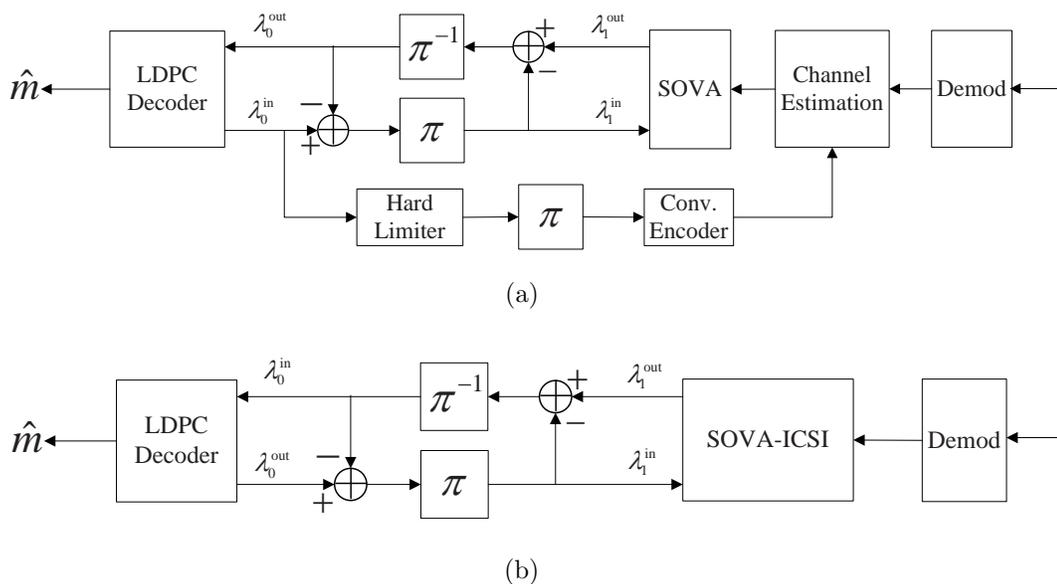


Figure 5.4: System models of iterative channel estimation and decoding: (a) ICED-SOVA-PSAM; (b) ICED-SOVA-ICSI.

process resembles the turbo decoding process, which has been discussed in Chapter 2. In the literature, this is known as the ‘turbo-like’ decoding [136–138]. In ICED-SOVA-PSAM and ICED-SOVA-ICSI, besides the performance gain due to more accurate channel estimates, additional coding gain is also achieved from the turbo-like coding structure.

5.5 Simulation Studies

In this section, we present the simulation results which illustrate the superiority of the SOVA-ICSI. We choose the simplest maximum-free-distance convolutional code of rate half with the generator given by [5 7] in octal. The rate-half, regular,

(1008, 504) LDPC code of column weight three, adopted from [123], is chosen and BP decoding or the sum-product algorithm is used in the LDPC decoder, with a maximum of 50 iterations. Each BEP data point is obtained from collecting at least 1000 bit errors.

A block interleaver of size 1008×100 is chosen. The time-selective, frequency-flat, slow, Rayleigh fading channel is assumed, which follows Jakes's isotropic scattering model [152]: the real and imaginary parts of the channel gain are assumed to be independent, each with autocorrelation

$$R_c(k) = \sigma^2 J_0(2\pi f_d T_s k)$$

where f_d is the relative Doppler shift between the transmitter and the receiver, T_s is the symbol period and $J_0(\cdot)$ is the Bessel function of the first kind of order zero. Two normalized fade rates, $f_d T_s = 0.001$ and $f_d T_s = 0.005$, are considered. We assume that this fading channel can be approximated to be the block-wise fading with fading block length $W = \frac{1}{4f_d T_s}$.

5.5.1 Comparison between SOVA-ICSI and SOVA-PSAM

In Figs. 5.5(a) and (b), we compare the performance of BP decoding between the SOVA-ICSI and SOVA-PSAM for convolutional codes with different length K with various pilot symbol spacings at the normalized fade rate of $f_d T_s = 0.001$. It can be observed that the SOVA-ICSI outperforms the SOVA-PSAM by 0.6 dB, 0.35 dB and 0.17 dB at the BEP of 10^{-5} when B is equal to 50, 30 and 10, respectively when $K = 8$, and the performance gain of the SOVA-ICSI is

0.6 dB, 0.55 dB and 0.45 dB over the SOVA-PSAM when B is equal to 50, 30 and 10, respectively when $K = 200$. Moreover, the performance with the short convolutional code is better than that with the long convolutional code.

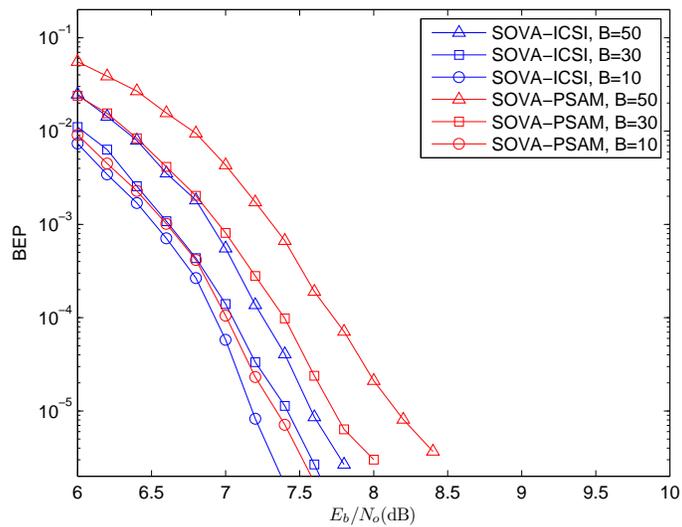
Similar observations can be made at the normalized fade rate of $f_d T_s = 0.005$, as shown in In Figs. 5.6(a) and (b).

5.5.2 Effect of window length in SOVA-ICSI

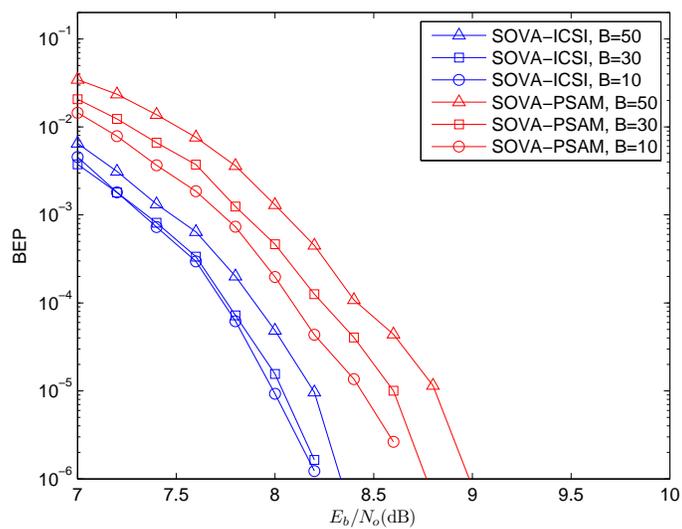
In this section, we study the effect of the window length L for the SOVA-ICSI. Figs. 5.7(a) and (b) show the BEP of the BP decoding with SOVA-ICSI at different window lengths at the fade rate $f_d T_s = 0.001$. It is observed that when L is less than 10, the BEP performance degrades with the decrease of L . This is because when L is small, the reliability metric is computed based on a very short segment. The probability of encountering the path divergence problem is thus quite significant. Hence, the BEP increases with the decrease of L .

When L is sufficiently large, the occurrence of the path divergence problem is almost negligible and the minimum BEPs are achieved when the window lengths are around 10 to 12.

When L increases further, more data signals are considered in the joint optimum decision in SOVA-ICSI. The number of pilot signals is not sufficient to overcome the phase and divisor ambiguities. This explains why the BEP shoots up when the window length is further increased. Moreover, when the pilot spacing is short, there would be more pilot signals within each fading block, which can be used to resolve the phase and divisor ambiguities. Hence, larger window

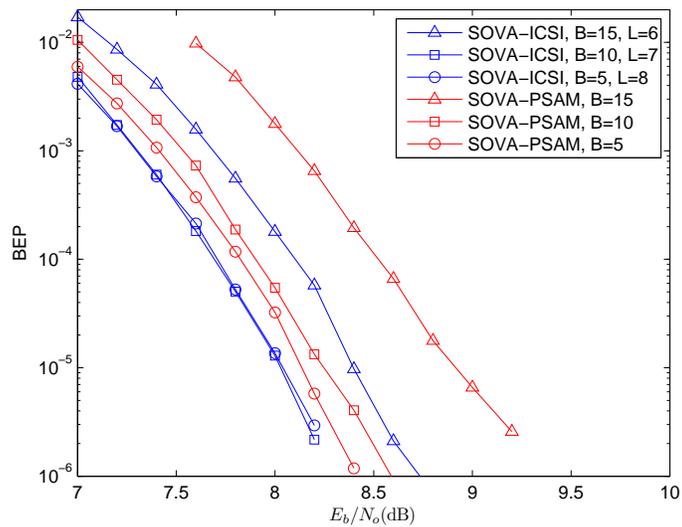


(a)

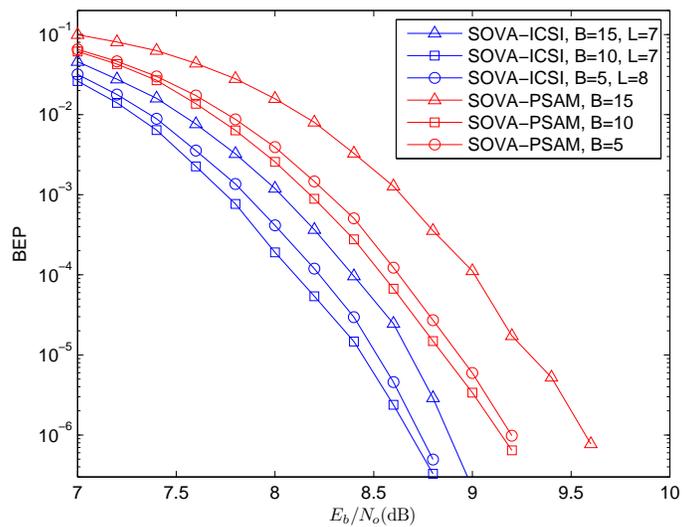


(b)

Figure 5.5: BEP performance comparison of SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.001$ with various parameters: (a) $L = 10$, $K = 8$; (b) $L = 10$, $K = 200$.



(a)



(b)

Figure 5.6: BEP performance comparison of SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.005$ with various parameters: (a) $K = 8$; (b) $K = 200$.

length can be used for the SOVA-ICSI. It can be observed in Fig. 5.7(a) that for $B = 50$, the BEP increases immediately after hitting the optimum value at $L = 10$, while for $B = 10$, the BEP remains almost the same at the minimum value from $L = 10$ to $L = 18$, and rises only when L goes beyond 19.

Similar trends are observed at the fade rate of $f_d T_s = 0.005$, as shown in Figs. 5.8(a) and (b). The optimum values of L , for the short convolutional encoder with $K = 8$, are obtained at 6, 7 and 8 for $B = 15$, $B = 10$ and $B = 5$, respectively. For $K = 200$, the optimum values are obtained at 7, 7 and 8 for $B = 15$, $B = 10$ and $B = 5$, respectively. Compared with Figs. 5.7(a) and (b), the u-shape curves have much narrower bases, and the BEP rises very quickly when L deviates away from the optimum value, suggesting that the performance of the SOVA-ICSI is more sensitive to the window length. This can be explained as follows. Since the fading block length W is approximated by $\frac{1}{4f_d T_s}$, the fading block at the fade rate of $f_d T_s = 0.005$ is only one-fifth of the length of the fading block at $f_d T_s = 0.001$. There are much fewer pilot signals in the fading block, which can only effectively resolve the phase and divisor ambiguities for the SOVA-ICSI with short window lengths. For these window lengths, the probability of occurrence of the path divergence problem in the SOVA-ICSI is still considerably high. Thus, if a smaller value of L is chosen, the SOVA-ICSI will suffer significantly from the path divergence problem and the performance degrades rapidly. On the other hand, when L increases further, the phase and divisor ambiguities could no longer be resolved effectively and the BEP also increases very fast. Therefore, sharp u-shape curves are observed.

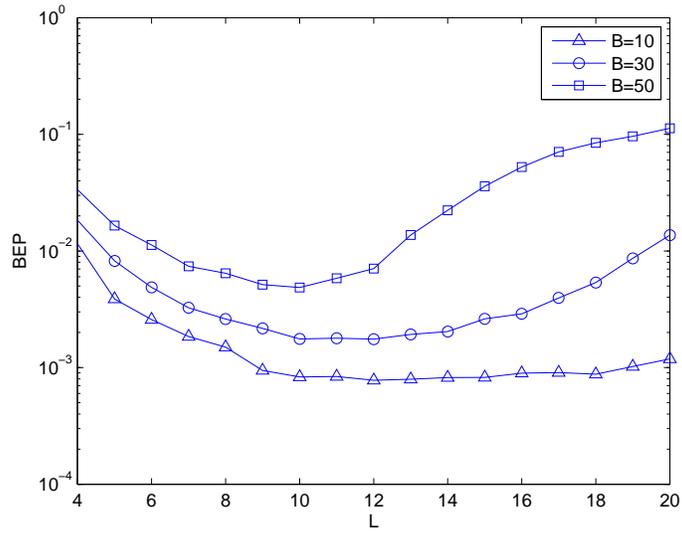
Based on Figs. 5.7(a) and (b), we choose the window length L to be 10 for all the simulations at the normalized fade rate of 0.001 in section 5.5.1. For the normalized fade rate of 0.005, since the performance is very sensitive to the window length, we choose the optimum L for each case, as indicated in Figs. 5.6(a) and (b).

5.5.3 Effect of SNR Mis-estimation

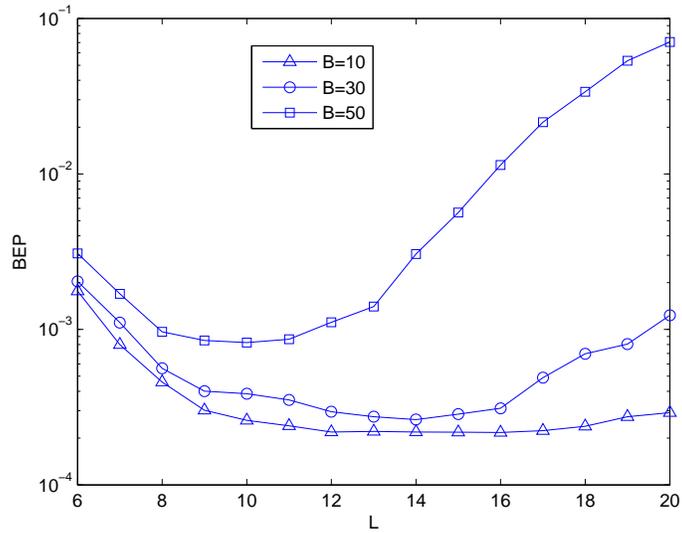
Figs. 5.9(a) and (b) show the performance of the SOVA-ICSI and the SOVA-PSAM subjected to SNR estimation errors at the normalized fade rate of $f_d T_s = 0.001$. The horizontal axis indicates the estimation errors (in dB), with positive values representing over-estimation and negative values denoting under-estimation. It can be observed that over-estimation results in detrimental BEP performance degradation and an under-estimation of around 3dB tends to improve the performance for both the algorithms. Furthermore, it can be observed that the curves of the SOVA-ICSI have broader u-shaped bases, suggesting that the SOVA-ICSI is more robust than the SOVA-PSAM, subjected to SNR mis-estimation errors. Similar observations can be made at the normalized fade rate of $f_d T_s = 0.005$, as shown in In Figs. 5.10(a) and (b).

5.5.4 Iterative Channel Estimation and Decoding

In this section, the two receiver structures with iterative channel estimation and decoding are considered, namely, the ICED-SOVA-PSAM and the ICED-SOVA-

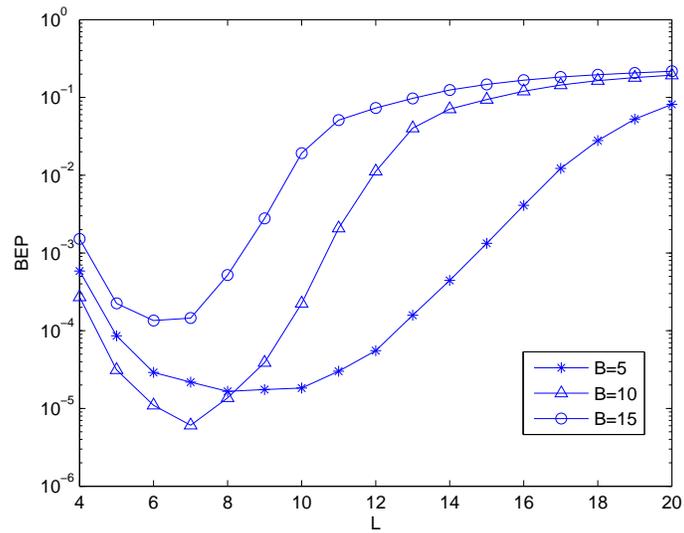


(a)

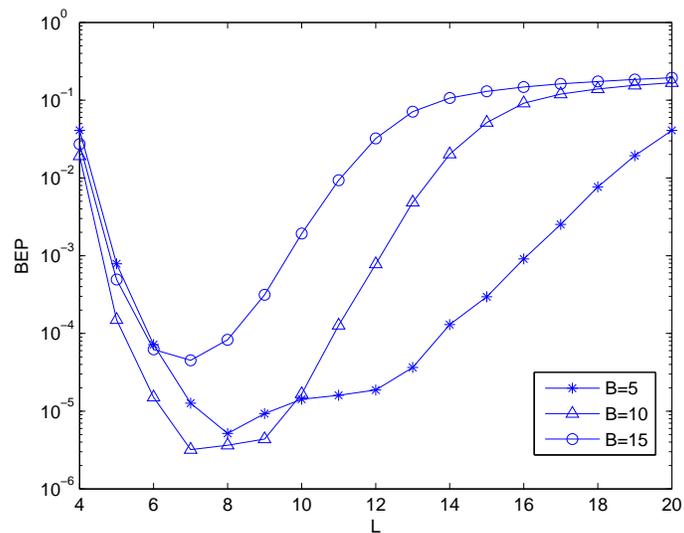


(b)

Figure 5.7: Effect of the window length L in the SOVA-ICSI at the normalized fade rate of $f_d T_s = 0.001$: (a) $E_b/N_0 = 6.5$ dB, $K = 8$; (b) $E_b/N_0 = 7.5$ dB, $K = 200$.

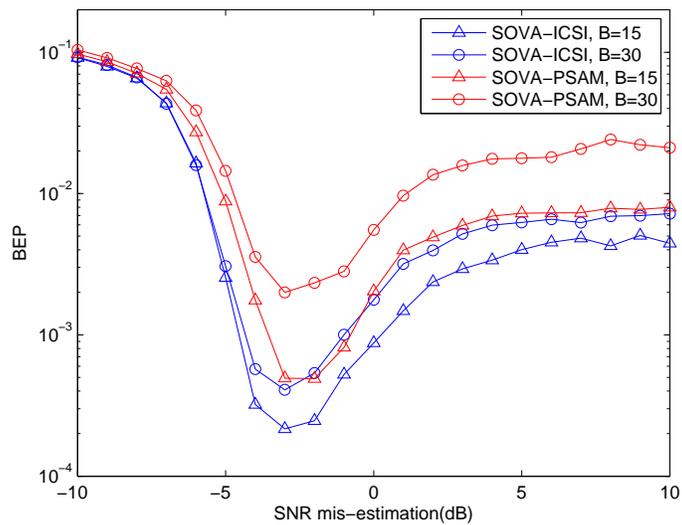


(a)

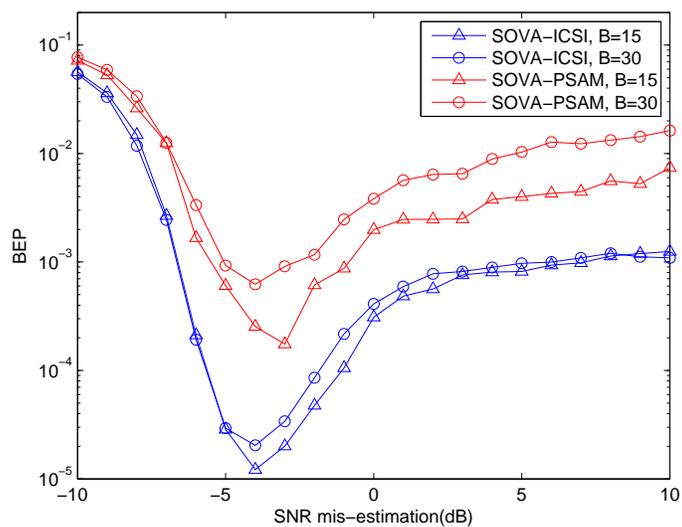


(b)

Figure 5.8: Effect of the window length L in the SOVA-ICSI at the normalized fade rate of $f_d T_s = 0.005$: (a) $E_b/N_0 = 8$ dB, $K = 8$; (b) $E_b/N_0 = 8.5$ dB, $K = 200$.

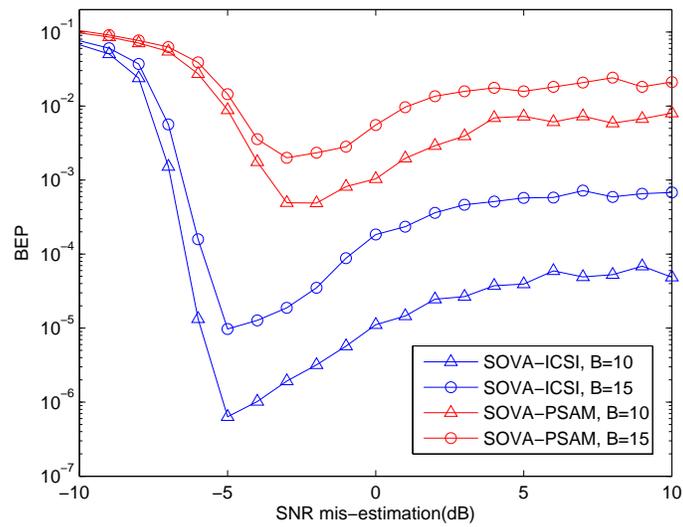


(a)

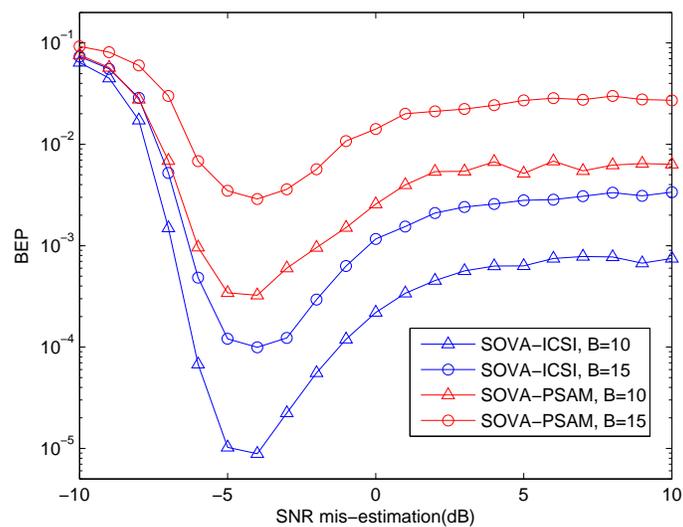


(b)

Figure 5.9: Robustness comparison between SOVA-ICSI and SOVA-PSAM under SNR mis-estimation at the normalized fade rate of $f_d T_s = 0.001$: (a) $E_b/N_0 = 6.5$ dB, $L = 10$, $K = 8$; (b) $E_b/N_0 = 7.5$ dB, $L = 10$, $K = 200$.



(a)



(b)

Figure 5.10: Robustness comparison between SOVA-ICSI and SOVA-PSAM under SNR mis-estimation at the normalized fade rate of $f_d T_s = 0.005$: (a) $E_b/N_0 = 8$ dB, $L = 7$, $K = 8$; (b) $E_b/N_0 = 8$ dB, $L = 7$, $K = 200$.

ICSI. For each receiver, 10 super-iterations are performed, with a maximum of 5 internal-iterations in each super-iteration.

In Figs. 5.11, 5.12, 5.13 and 5.14, the BEP performances of SOVA-ICSI, SOVA-PSAM, ICED-SOVA-PSAM and ICED-SOVA-ICSI are compared for the convolutional code with different lengths at the two normalized fade rates. The performance of the SOVA-PSAM with perfect CSI is obtained using the SOVA-PSAM receiver structure assuming the CSI is perfectly known at the receiver. Similarly, the ICED-SOVA-PSAM with perfect CSI assumes the ICED-SOVA-PSAM receiver structure with perfect knowledge of CSI.

It can be observed that the ICED-SOVA-ICSI (resp., the ICED-SOVA-PSAM) has much better performance than the SOVA-ICSI (resp., the SOVA-PSAM), and the performance gain of using iterative channel estimation and decoding is around 1 – 1.5 dB at the BEP of 10^{-5} . Not surprisingly, the ICED-SOVA-ICSI outperforms the ICED-SOVA-PSAM by approximately 0.5 dB. In Figs. 5.11 and 5.12, it is observed that the performance of the ICED-SOVA-ICSI is better than that of the SOVA-PSAM with perfect CSI. This is because the implicit channel estimation in the ICED-SOVA-ICSI has been quite accurate. The performance loss due to estimation errors is smaller compared to the performance gain from the additional coding gain of the turbo-like coding structure. On the other hand, when the fade rate is higher, the channel varies more rapidly and it is expected that accurate channel estimation becomes much more difficult. As shown in Fig. 5.13, the performance of the ICED-SOVA-ICSI is worse than that of the SOVA-PSAM with perfect CSI. In this case, the channel estimates are less accurate and

the loss in the BEP performance due to estimation errors is huge. The additional coding gain of the ICED-SOVA-ICSI is not sufficiently large to compensate the performance loss due to the decreasing estimation accuracy. However, it can be observed that the performance gap between the ICED-SOVA-ICSI and the SOVA-PSAM with perfect CSI decreases as the BEP drops. In Fig. 5.14, this performance gap decreases with the decrease of the BEP, and eventually when the BEP is less than 10^{-4} , the ICED-SOVA-ICSI outperforms the SOVA-PSAM with perfect CSI. This is expected because when the BEP is low, the channel estimates from the data signals are more accurate and consequently the performance degradation caused by channel estimation errors becomes smaller.

5.6 Conclusion

In this chapter, the SOVA-ICSI detector is proposed for the transmission over a flat quasi-static fading channel, which produces soft-decision output. The SOVA-ICSI does not require explicit channel estimation or the channel fading statistics. We show that the SOVA-ICSI has substantially better BEP performance than the conventional receiver with PSA channel estimation. Since the channel statistics is not available, the channel estimation accuracy cannot be obtained. Note that if the channel statistics is available, the SOVA-ICSI can be generalized to take into account the information of the channel estimation accuracy. This extension will be investigated in our future studies.

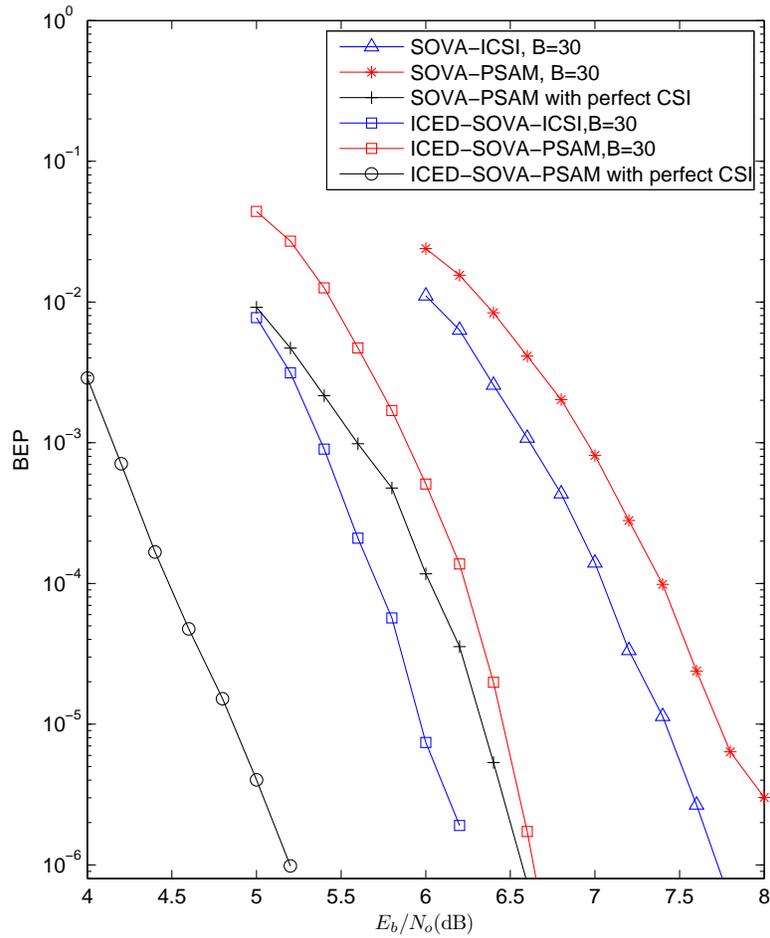


Figure 5.11: BER performance comparison of ICED-SOVA-ICSI, ICED-SOVA-PSAM, SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.001$ with $K = 8$.

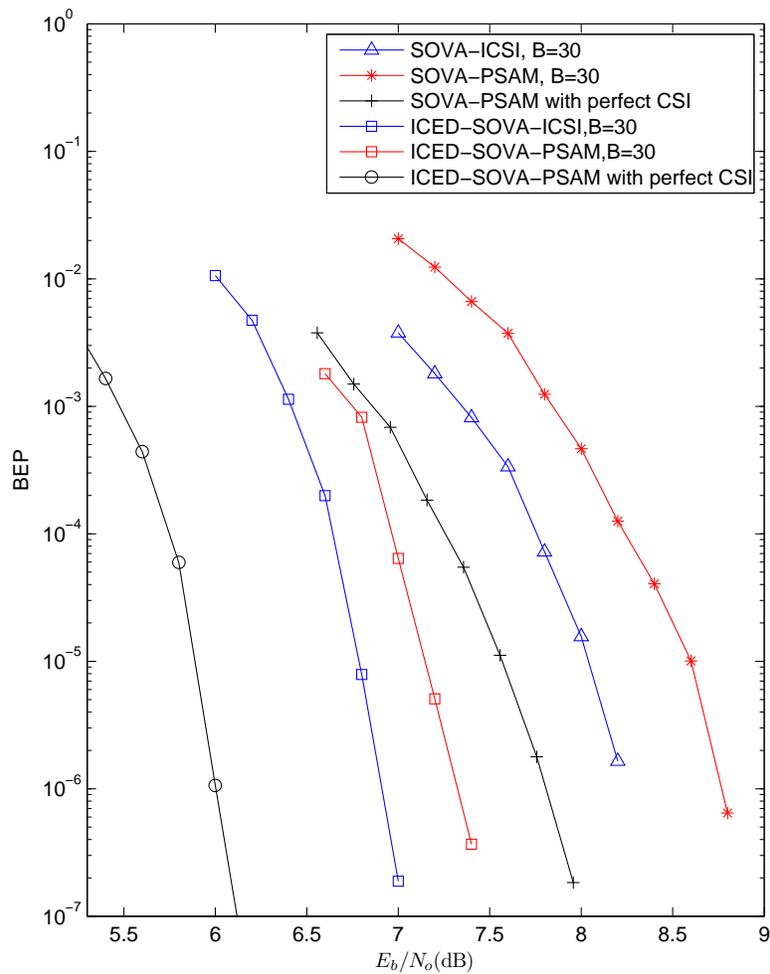


Figure 5.12: BEP performance comparison of ICED-SOVA-ICSI, ICED-SOVA-PSAM, SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.001$ with $K = 200$.

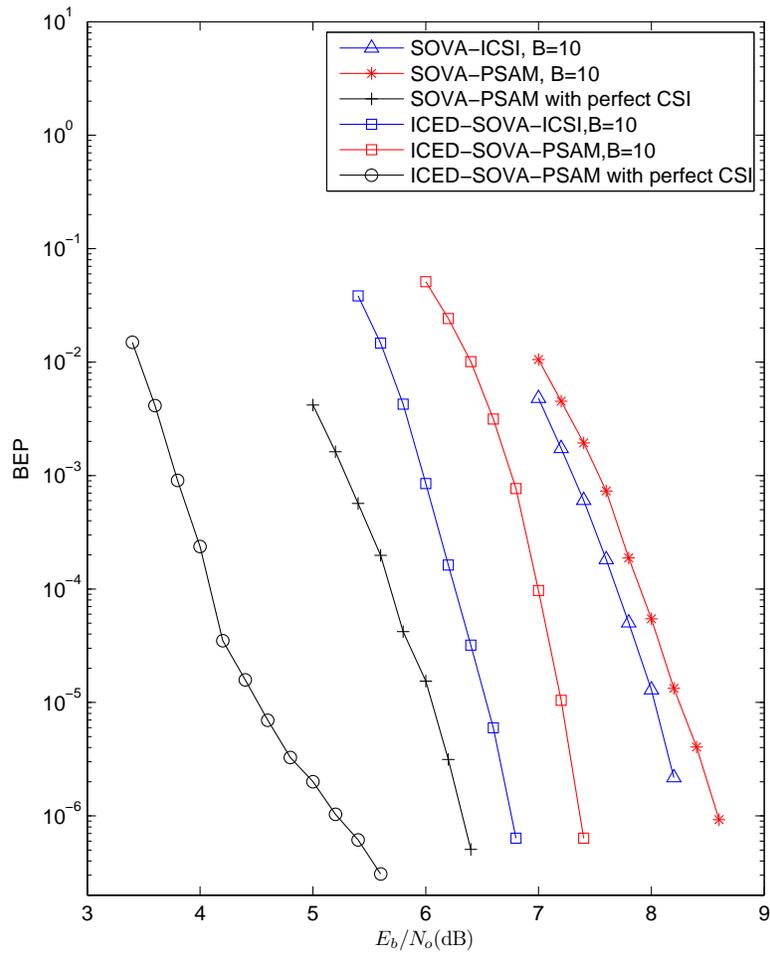


Figure 5.13: BEP performance comparison of ICED-SOVA-ICSI, ICED-SOVA-PSAM, SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.005$ with $K = 8$.

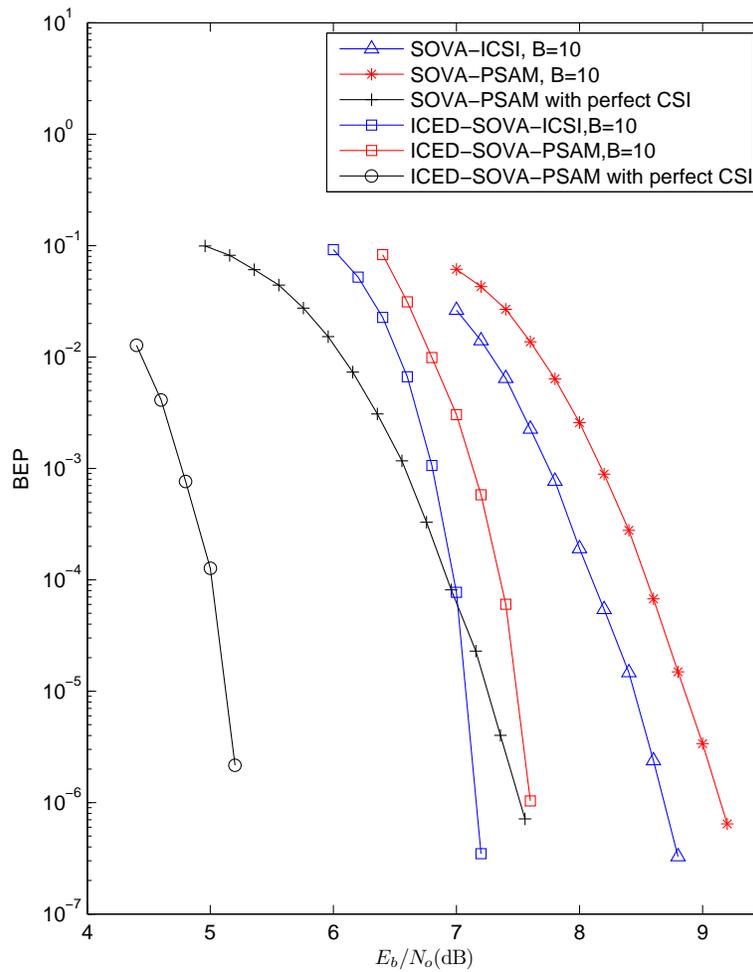


Figure 5.14: BEP performance comparison of ICED-SOVA-ICSI, ICED-SOVA-PSAM, SOVA-ICSI and SOVA-PSAM at the normalized fade rate of $f_d T_s = 0.005$ with $K = 200$.

Chapter 6

Generalizations of the BCJR

Algorithm for Turbo Decoding

over Flat Rayleigh Fading

Channels with Imperfect CSI

In Chapter 4, we derived the PSAM-LLR metric from the first principle of probability theory for the transmissions over time-selective, frequency-flat, Rayleigh fading channels with PSAM channel estimation. The main difference between the PSAM-LLR metric and the conventional LLR metric, the A-PSAM-LLR metric, is that the information concerning the channel estimation accuracy is taken into the computation of the LLR for the former and neglected for the latter. The decoding performance of LDPC codes with these metrics is studied, demonstrating

the superiority of the PSAM-LLR metric and the importance of incorporating the knowledge of the channel estimation accuracy in the iterative decoding process.

In this chapter, we consider the transmission of turbo codes over the same channel. For turbo decoding, besides the *a posteriori* LLRs, the iterative decoder also requires the APP decoding algorithm for the constituent convolutional codes, as discussed in Chapter 2. The conventional BCJR algorithm and the SOVA are only applicable for the AWGN channel. In this chapter, we will propose generalizations of the BCJR algorithm and the SOVA for the transmissions over time-selective, frequency-flat, Rayleigh fading channels with PSAM channel estimation. These algorithms will be applied to decode turbo codes formed by PCCCs, together with the LLR metrics derived in Chapter 4, and their BEP performances will be evaluated and compared. Furthermore, iterative channel estimation and decoding technique will be considered for both hard-decision feedback and soft-decision feedback.

Our work here reinforces the importance of incorporating the knowledge of the channel estimation accuracy in the iterative decoding process when precise KCS is available at the receiver.

6.1 System Model

The system model is shown in Fig. 6.1. A binary information bit sequence $\mathbf{m} = [m(1) m(2) \dots m(K)]$ is encoded by a turbo encoder of PCCCs, as shown in Fig. 6.2, where $m(k)$ takes on a value of $\{0, 1\}$ with *a priori* LLR given by

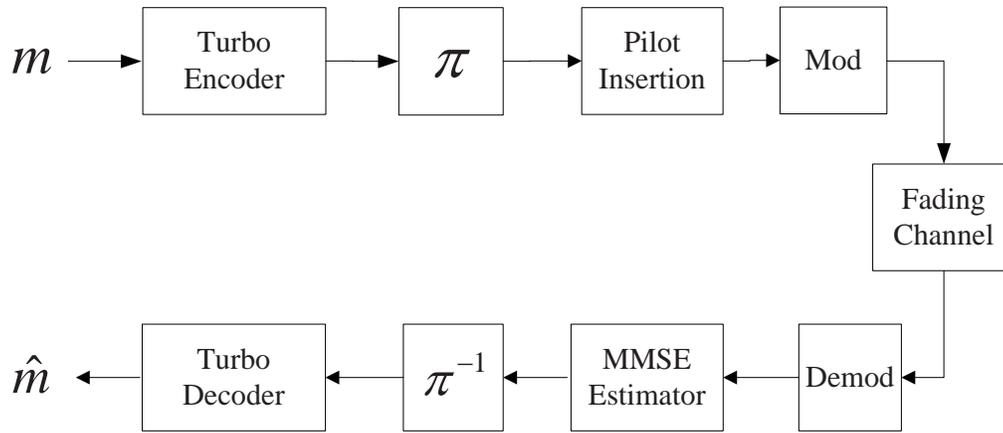


Figure 6.1: System model of decoding turbo codes over fading channels with PSAM channel estimation

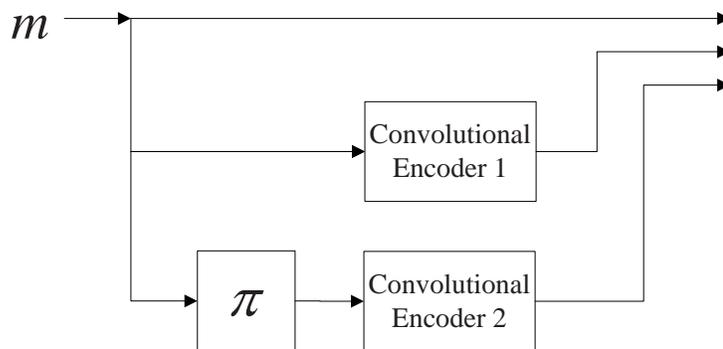


Figure 6.2: Block diagram of turbo encoder

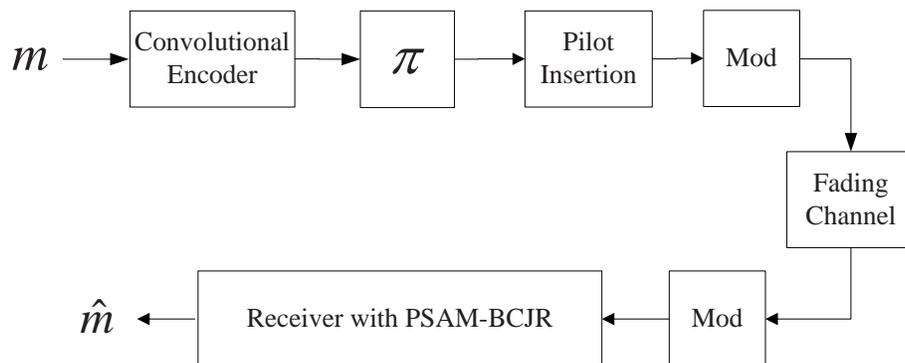


Figure 6.3: Simplified system model

$L_a(m(k)) = \ln \frac{P(m(k)=0)}{P(m(k)=1)}$. The turbo coded sequence is then shuffled by a block channel interleaver. (Note that we refer to the interleaver in Fig. 6.2 as the code interleaver.) The interleaved sequence is sequentially mapped to BPSK signals with energy E_s . Pilot signals are periodically inserted with pilot spacing B . Without loss of generality, we can assume that every pilot signal has the same energy as a data signal and takes the form $p = \sqrt{E_s}e^{j\pi^0}$. After that the signal sequence is transmitted through a time-selective, frequency-flat, Rayleigh fading channel.

As we discussed in Chapter 2, to decode the PCCCs, the extrinsic information is exchanged iteratively between the two constituent convolutional codes using the turbo principle and the SISO APP decoder is required to compute the extrinsic information. Hence, we will focus on the APP decoding algorithm or the BCJR algorithm for convolutional codes over the time-selective, frequency-flat, Rayleigh fading channels. To this end, it is sufficient to consider the simplified system mode as shown in Fig. 6.3.

Assume that the message sequence $\mathbf{m} = [m(1) \ m(2) \ \dots \ m(K)]$ is convolutionally encoded by the convolutional encoder rate $R = 1/n$. The n output streams of convolutionally encoded message are denoted by $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^n$, and for each $i = 1, 2, \dots, n$, $\mathbf{x}^i = [x^i(1) \ x^i(2) \ \dots \ x^i(K)]$.

Note that the convolutional encoder under consideration may not return to the all-zero state after encoding the message \mathbf{m} . The reasons are as follows. For the turbo encoder as shown Fig. 6.2, we usually deliberately append $v-1$ flushing bits to drive the first convolutional encoder to the all-zero state at the end of encoding. However, after the message sequence with the appended flushing bits is permuted by the code interleaver, it is not guaranteed that the second convolutional encoder will end up in the all-zero state. Hence, it is frequently assumed that the first encoder is driven to the all-zero state, and the second encoder can end up in any possible state. Our generalization of the BCJR algorithm is derived to be capable of handling both the cases.

After encoding, a block interleaver is used. After BPSK modulation and pilot insertion, the signal sequence is transmitted through a time-selective, frequency-flat, Rayleigh fading channel. Each received signal can be expressed as

$$r^i(k) = c^i(k) \sqrt{E_s} e^{j\pi x^i(k)} + n^i(k), \quad (6.1)$$

where $r^i(k)$ denotes the received signal corresponding to the transmission of $x^i(k)$, and $\{n(l, b)\}$ is a set of statistically independent, complex, Gaussian random variables, each with mean zero and variance N_0 . The channel gains $\{c^i(k)\}$ are modeled as a correlated, zero-mean, complex, Gaussian process with known au-

to correlation function.

We will first derive the PSAM-BCJR algorithm, based on which, the receiver is designed.

For ease of representation, we use $\mathbf{x}(k)$ to denote the vector $[x^1(k), x^2(k), \dots, x^n(k)]^T$, $\mathbf{r}(k)$ to denote the vector $[r^1(k), r^2(k), \dots, r^n(k)]^T$ and $\mathbf{c}(k)$ to denote the vector $[c^1(k), c^2(k), \dots, c^n(k)]^T$. Furthermore, let \mathbf{r} denote the row vector $[\mathbf{r}(1), \mathbf{r}(2), \dots, \mathbf{r}(K)]$.

6.2 PSAM-BCJR Algorithm

Let \mathbb{S} denote the set of all the received pilot signals. From the definition, the *a posteriori* LLR for the message bit $m(k)$, where $1 \leq k \leq K$, can be expressed as

$$\lambda(m(k)) = \ln \frac{P(m(k) = 0 | \mathbf{r}, \mathbb{S})}{P(m(k) = 1 | \mathbf{r}, \mathbb{S})}, \quad (6.2)$$

Making use of the trellis structure of the convolutional code, the *a posteriori* probability $P(m(k) = 0 | \mathbf{r}, \mathbb{S})$ can be rewritten as

$$\begin{aligned} P(m(k) = 0 | \mathbf{r}, \mathbb{S}) &= \frac{p(m(k) = 0, \mathbf{r} | \mathbb{S})}{p(\mathbf{r} | \mathbb{S})} \\ &= \frac{\sum_{(s, s') \in \Sigma_k^0} p(s(k) = s', s(k+1) = s, \mathbf{r} | \mathbb{S})}{p(\mathbf{r} | \mathbb{S})} \end{aligned}$$

where Σ_k^0 denotes the set of all state pairs (s', s) such that if the current state $s(k)$ is s' and the incoming message input is 0 bit, the next state $s(k+1)$ will be s . Rewrite $P(m(k) = 1 | \mathbf{r}, \mathbb{S})$ in the same way, (6.2) can be expressed as

$$\lambda(m(k)) = \ln \frac{\sum_{(s, s') \in \Sigma_k^0} p(s(k) = s', s(k+1) = s, \mathbf{r} | \mathbb{S})}{\sum_{(s, s') \in \Sigma_k^1} p(s(k) = s', s(k+1) = s, \mathbf{r} | \mathbb{S})}, \quad (6.3)$$

Note that the convolutional code has 2^v possible states and for each state, if the incoming message is fixed, there is only one possible next state. Thus the summation extends over a set of 2^v state pairs.

Now we consider the joint probability density function $p(s(k) = s', s(k+1) = s, \mathbf{r}|\mathbb{S})$. For simplicity, we denote $p(s(k) = s', s(k+1) = s, \mathbf{r}|\mathbb{S})$ in a shorter form as $p(s', s, \mathbf{r}|\mathbb{S})$. Now, the probability density function $p(s', s, \mathbf{r}|\mathbb{S})$ is rewritten as

$$p(s', s, \mathbf{r}|\mathbb{S}) = p(s', s, \mathbf{r}_{t < k}, \mathbf{r}(k), \mathbf{r}_{t > k}|\mathbb{S}), \quad (6.4)$$

where $\mathbf{r}_{t < k} = [\mathbf{r}(1), \mathbf{r}(2), \dots, \mathbf{r}(k-1)]$ and $\mathbf{r}_{t > k} = [\mathbf{r}(k+1), \mathbf{r}(k+2), \dots, \mathbf{r}(K)]$.

Applying Bayes' rule, we obtain

$$\begin{aligned} p(s', s, \mathbf{r}|\mathbb{S}) &= p(\mathbf{r}_{t > k}|s', s, \mathbf{r}_{t < k}, \mathbf{r}(k), \mathbb{S})p(s', s, \mathbf{r}_{t < k}, \mathbf{r}(k)|\mathbb{S}) \\ &= p(\mathbf{r}_{t > k}|s', s, \mathbf{r}_{t < k}, \mathbf{r}(k), \mathbb{S})p(s, \mathbf{r}(k)|s', \mathbf{r}_{t < k}, \mathbb{S})p(s', \mathbf{r}_{t < k}|\mathbb{S}) \end{aligned}$$

Conditioned on $s(k+1) = s$, the future received signal vector $\mathbf{r}_{t > k}$ does not depend $s(k) = s'$, or the previous received signals $\mathbf{r}_{t < k}$, or the current received signal $\mathbf{r}(k)$. Thus, we get

$$p(\mathbf{r}_{t > k}|s', s, \mathbf{r}_{t < k}, \mathbf{r}(k), \mathbb{S}) = p(\mathbf{r}_{t > k}|s, \mathbb{S}) \quad (6.5)$$

Conditioned on the current state s' , the joint pdf of next state s and the received signal $\mathbf{r}(k)$ is independent of the previous received signals, which leads to

$$p(s, \mathbf{r}(k)|s', \mathbf{r}_{t < k}, \mathbb{S}) = p(s, \mathbf{r}(k)|s', \mathbb{S}) \quad (6.6)$$

Now we define

$$\begin{aligned}\alpha_k(s') &= p(s', \mathbf{r}_{t < k} | \mathbb{S}) \\ \beta_{k+1}(s) &= p(\mathbf{r}_{t > k} | s, \mathbb{S}) \\ \gamma_k(s', s) &= p(s, \mathbf{r}(k) | s', \mathbb{S})\end{aligned}$$

From (6.5) and (6.6), we can write (6.4) as

$$p(s', s, \mathbf{r} | \mathbb{S}) = \beta_{k+1}(s) \gamma_k(s', s) \alpha_k(s') \quad (6.7)$$

In the BCJR algorithm, the values of $\alpha_k(s')$ and $\beta_{k+1}(s)$ can be computed recursively from the forward and backward recursions as, (we refer reader to reference [159] for the details.)

Forward recursion:

$$\alpha_{k+1}(s) = \sum_{s' \in \Omega_k} \gamma_k(s', s) \alpha_k(s') \quad (6.8)$$

where Ω_k is the set of all possible states at time k . Since the encoder always starts from the all-zero state, this forward recursion begins with the initial condition $\alpha_0(s') = 1$ if s' is the all-zero state and $\alpha_0(s') = 0$ otherwise.

Backward recursion:

$$\beta_k(s') = \sum_{s \in \Omega_k} \gamma_k(s', s) \beta_{k+1}(s) \quad (6.9)$$

where Ω_k is the set of all possible states at time $k + 1$. If it is known that the ending state is the all-zero state, the backward recursion begins with the initial

condition $\beta_K(s') = 1$ if s' is the all-zero state and $\beta_K(s') = 0$ otherwise. On the other hand, if we have no information about the ending state, we may assume that all states are possible. In this case, the backward recursion begins with the initial condition $\beta_K(s') = 1/2^v$ for all possible states.

Note that both the forward and the backward recursion can be computed if $\gamma_k(s', s)$ is available. Now we will derive the expression for $\gamma_k(s', s)$. Using Bayes' rule, we start by rewriting $\gamma_k(s', s)$ as

$$\begin{aligned} \gamma_k(s', s) &= p(s, \mathbf{r}(k) | s', \mathbb{S}) \\ &= \frac{p(s', s, \mathbf{r}(k) | \mathbb{S})}{P(s' | \mathbb{S})} \\ &= \frac{P(s', s | \mathbb{S}) p(s', s, \mathbf{r}(k) | \mathbb{S})}{P(s' | \mathbb{S}) P(s', s | \mathbb{S})} \end{aligned}$$

Since both the events $s(k) = s'$ and $s(k+1) = s$ are independent of the pilot signals, we have

$$\begin{aligned} \gamma_k(s', s) &= \frac{P(s', s)}{P(s')} \frac{p(s', s, \mathbf{r}(k) | \mathbb{S})}{P(s', s | \mathbb{S})} \\ &= P(s(k+1) = s | s(k) = s') \cdot p(\mathbf{r}(k) | s(k) = s', s(k+1) = s, \mathbb{S}) \quad (6.10) \end{aligned}$$

The first term of the product in (6.10) is the *a priori* probability of $m(k)$, i.e.,

$$P(s(k+1) = s | s(k) = s') = \begin{cases} P(m(k) = 0) & \text{if } (s', s) \in \Sigma_k^0 \\ P(m(k) = 1) & \text{if } (s', s) \in \Sigma_k^1 \\ 0 & \text{otherwise} \end{cases}$$

Note that *a priori* LLR $L_a(m(k)) = \ln \frac{P(m(k)=0)}{P(m(k)=1)}$ is a constant, which is independent of the realization of the random variable $m(k)$. The *a priori* probability

$P(m(k) = i)$ for $i = 0$ or 1 can be expressed as

$$P(m(k) = i) = G_k e^{(1-i)L_a(m(k))},$$

where $G_k = (1 + e^{L_a(m(k))})^{-1}$ is a constant, independent of $m(k)$.

Now we consider the second term in (6.10). It is reasonable to assume that $P(s(k+1) = s | s(k) = s') \neq 0$, i.e., $(s', s) \in \Sigma_k^0$ or $(s', s) \in \Sigma_k^1$, otherwise (6.10) simply equals to zero. Let $\mathbf{x}(k) = [x^1(k), x^2(k), \dots, x^n(k)]^T$ be the corresponding convolutional encoder output when the state transition is from s' to s . By introducing the instantaneous channel gain vector $\mathbf{c}(k) = [c^1(k), c^2(k), \dots, c^n(k)]^T$, the second term in (6.10) can be rewritten in the form of

$$\begin{aligned} & p(\mathbf{r}(k) | s(k) = s', s(k+1) = s, \mathbb{S}) \\ &= p(\mathbf{r}(k) | \mathbf{x}(k), \mathbb{S}) \\ &= \underbrace{\int \dots \int}_n p(\mathbf{r}(k), \mathbf{c}(k) | \mathbf{x}(k), \mathbb{S}) d\mathbf{c}(k) \\ &= \underbrace{\int \dots \int}_n p(\mathbf{r}(k) | \mathbf{c}(k), \mathbf{x}(k), \mathbb{S}) \cdot p(\mathbf{c}(k) | \mathbf{x}(k), \mathbb{S}) d\mathbf{c}(k) \end{aligned} \quad (6.11)$$

Conditioned on $\mathbf{c}(k)$ and $\mathbf{x}(k)$, the only randomness in $\mathbf{r}(k)$ is due to the AWGN, which is independent of the pilot signal set \mathbb{S} . Since the AWGN is independent for each received signal, the first term in the integrand in the last

line of (6.11) can be expressed as

$$\begin{aligned}
p(\mathbf{r}(k)|\mathbf{c}(k), \mathbf{x}(k), \mathbb{S}) &= p(\mathbf{r}(k)|\mathbf{c}(k), \mathbf{x}(k)) \\
&= \prod_{i=1}^n p(r^i(k)|c^i(k), x^i(k)) \\
&= \prod_{i=1}^n \frac{1}{\pi N_0} \exp\left[-\frac{|r^i(k) - e^{j\pi x^i(k)} E_s^{\frac{1}{2}} c^i(k)|^2}{N_0}\right] \quad (6.12)
\end{aligned}$$

Next, we consider the term $p(\mathbf{c}(k)|\mathbf{x}(k), \mathbb{S})$. Since $\mathbf{c}(k)$ is independent of the transmitted information $\mathbf{x}(k)$, we have

$$p(\mathbf{c}(k)|\mathbf{x}(k), \mathbb{S}) = p(\mathbf{c}(k)|\mathbb{S}).$$

Assuming that the channel interleaver is perfect, the memory of the channel is perfectly broken and thus the channel gains $c^1(k), c^2(k), \dots, c^n(k)$ are statistically independent. We have

$$p(\mathbf{c}(k)|\mathbf{x}(k), \mathbb{S}) = \prod_{i=1}^n p(c^i(k)|\mathbb{S}).$$

As the channel gains are jointly Gaussian, $c^i(k)$ and \mathbb{S} are also jointly Gaussian. Thus, $c^i(k)$ conditioned on \mathbb{S} is a conditional Gaussian random variable with conditional mean $\mathbb{E}[c^i(k)|\mathbb{S}]$ and conditional variance $\text{var}[c^i(k)|\mathbb{S}]$. From [156], for jointly Gaussian random variables, we have

$$\begin{aligned}
\mathbb{E}[c^i(k)|\mathbb{S}] &= \hat{c}^i(k) \\
\text{var}[c^i(k)|\mathbb{S}] &= (2\tilde{\sigma}^2)^i(k),
\end{aligned}$$

where $\hat{c}^i(k)$ and $(2\tilde{\sigma}^2)^i(k)$ can be computed by (4.15) and (4.16), respectively, as in Chapter 3. Hence, the probability density function $p(c^i(k)|\mathbf{S}^i(k))$ can be

expressed as

$$p(c^i(k)|\mathbb{S}) = \frac{1}{\pi(2\tilde{\sigma}^2)^i(k)} \exp\left[-\frac{|c^i(k) - \hat{c}^i(k)|^2}{(2\tilde{\sigma}^2)^i(k)}\right]$$

and subsequently, we have

$$p(\mathbf{c}(k)|\mathbf{x}(k), \mathbb{S}) = \prod_{i=1}^n \frac{1}{\pi(2\tilde{\sigma}^2)^i(k)} \exp\left[-\frac{|c^i(k) - \hat{c}^i(k)|^2}{(2\tilde{\sigma}^2)^i(k)}\right]. \quad (6.13)$$

Substituting (6.12) and (6.13) into (6.11), and then exchanging the order of integration and product leads to

$$\begin{aligned} p(\mathbf{r}(k)|s(k) = s', s(k+1) = s, \mathbb{S}) \\ &= \prod_{i=1}^n \int p(r^i(k)|c^i(k), x^i(k)) \cdot p(c^i(k)|\mathbf{S}^i(k)) dc^i(k) \\ &= \prod_{i=1}^n \int \frac{1}{\pi N_0} \exp\left[-\frac{|r^i(k) - e^{j\pi x^i(k)} E_s^{\frac{1}{2}} c^i(k)|^2}{N_0}\right] \frac{1}{\pi(2\tilde{\sigma}^2)^i(k)} \exp\left[-\frac{|c^i(k) - \hat{c}^i(k)|^2}{(2\tilde{\sigma}^2)^i(k)}\right] dc^i(k) \end{aligned} \quad (6.14)$$

Combining the exponents, doing a completion of squares in the combined exponent and performing the integration, each integral in (6.14) can be evaluated as

$$\begin{aligned} &\int p(r^i(k)|c^i(k), x^i(k)) \cdot p(c^i(k)|\mathbf{S}^i(k)) dc^i(k) \\ &= \frac{A^i(k)}{\pi B^i(k)} \exp\left[\frac{C^i(k)}{B^i(k)N_0(2\tilde{\sigma}^2)^i(k)}\right], \end{aligned} \quad (6.15)$$

where

$$A^i(k) = \exp\left[-\frac{|r^i(k)|^2}{N_0} - \frac{|\hat{c}^i(k)|^2}{(2\tilde{\sigma}^2)^i(k)}\right], \quad (6.16)$$

$$B^i(k) = (2\tilde{\sigma}^2)^i(k)E_s + N_0, \quad (6.17)$$

$$C^i(k) = |(2\tilde{\sigma}^2)^i(k)\sqrt{E_s}r^i(k)e^{-j\pi x^i(k)} + N_0\hat{c}^i(k)|^2. \quad (6.18)$$

Substituting (6.15)–(6.18) back into the expression of $p(\mathbf{r}(k)|s', s, \mathbb{S})$ and after simplification, we obtain

$$p(\mathbf{r}(k)|s', s, \mathbb{S}) = \prod_{i=1}^n \frac{D^i(k)}{\pi B^i(k)} \exp\left(\frac{2\sqrt{E_s}}{B^i(k)} \operatorname{Re}[r^i(k)\hat{c}^i(k)^*(1 - 2x^i(k))]\right) \quad (6.19)$$

where

$$D^i(k) = \exp\left(\frac{((2\tilde{\sigma}^2)^i(k) - B^i(k))|r^i(k)|^2}{B^i(k)N_0} + \frac{(N_0 - B^i(k))|\hat{c}^i(k)|^2}{(2\tilde{\sigma}^2)^i(k)B^i(k)}\right)$$

Finally, we have obtained the expression for $\gamma_k(s', s)$ as

$$\gamma_k(s', s) = G_k \underbrace{\prod_{i=1}^n \frac{D^i(k)}{\pi B^i(k)}}_{\text{constant}} e^{(1-m(k))L_a(m(k))} \prod_{i=1}^n \exp\left(\frac{2\sqrt{E_s}}{B^i(k)} \operatorname{Re}[r^i(k)\hat{c}^i(k)^*(1 - 2x^i(k))]\right)$$

Note that the term $G_k \prod_{i=1}^n \frac{D^i(k)}{\pi B^i(k)}$ is a function of k , which is independent of the current state s' or the next state s of the decoding trellis. Note also that both the denominator and the numerator of the LLR expression in (6.4) contain the common term $\prod_{k=1}^K G_k \prod_{i=1}^n \frac{D^i(k)}{\pi B^i(k)}$, and thus the term $G_k \prod_{i=1}^n \frac{D^i(k)}{\pi B^i(k)}$ can be cancelled. We can reformulate the expression for $\gamma_k(s', s)$ as

$$\gamma_k(s', s) = e^{(1-m(k))L_a(m(k))} \prod_{i=1}^n \exp\left(\frac{2\sqrt{E_s}}{(2\tilde{\sigma}^2)^i(k)E_s + N_0} \operatorname{Re}[r^i(k)\hat{c}^i(k)^*(1 - 2x^i(k))]\right) \quad (6.20)$$

We refer to the BCJR algorithm with forward recursion in (6.8), backward recursion in (6.9), and $\gamma_k(s', s)$ in (6.20) as the PSAM-BCJR Algorithm.

If all the channel estimates $\hat{c}^i(k)$ are assumed to be perfect, $\gamma_k(s', s)$ can be computed as

$$\gamma_k(s', s) = e^{(1-m(k))L_a(m(k))} \prod_{i=1}^n \exp\left(\frac{2\sqrt{E_s}}{N_0} \operatorname{Re}[r^i(k)c^i(k)^*(1 - 2x^i(k))]\right) \quad (6.21)$$

We refer to this algorithm with $\gamma_k(s', s)$ in (6.21) as the Approximate PSAM-BCJR (A-PSAM-BCJR) Algorithm.

- *Remark*

For SOVA, the reliability metric of a path \mathbf{m} is defined as

$$M(\mathbf{v}) = \log p(\mathbf{r}|\mathbf{m}, \mathbb{S})P(\mathbf{v})$$

Following a similar derivation procedure, we can obtain

$$M(\mathbf{v}) = \sum_{k=1}^{k=K} \left(\log p(\mathbf{r}(k)|\mathbf{x}(k), \mathbb{S}) + L_a(m(k)) \right)$$

where $p(\mathbf{r}(k)|\mathbf{x}(k), \mathbb{S})$ is computed using (6.14). We will call this generalization of the SOVA as the PSAM-SOVA. Similarly, if the channel estimation is assumed to be perfect, $p(\mathbf{r}(k)|\mathbf{x}(k), \mathbb{S})$ can be computed using (6.14) with $B^i(k)$ replaced by N_0 , and the SOVA based on this reliability metric is called A-PSAM-SOVA.

6.3 Turbo Decoding with PSAM-BCJR/A-PSAM-BCJR algorithm

In this section, we will reformulate the turbo decoding algorithm with the newly developed PSAM-BCJR/A-PSAM-BCJR algorithm, based on the turbo principle in Chapter 2. We denote the *a priori* input to the constituent decoder 1 by $\lambda_1^{\text{in}}(m(k))$ and its output by $\lambda_1^{\text{out}}(m(k))$. The *a priori* input to the constituent decoder 2 is denoted by $\lambda_2^{\text{in}}(m(k))$ and its output by $\lambda_2^{\text{out}}(m(k))$. The turbo decoding with the PSAM-BCJR algorithm is described as follows:

Step 1. Compute the LLR for each message bit $m(k)$ using the PSAM-LLR in (4.18), denoted by $\lambda_0(m(k))$.

Step 2. Set the *a priori* input $\lambda_1^{\text{in}}(m(k))$ to be $L_a(m(k))$. Set the iteration number to be zero.

Step 3. Run the PSAM-BCJR algorithm on constituent decoder 1, with the *a priori* input $\lambda_1^{\text{in}}(m(k))$, and obtain the output $\lambda_1^{\text{out}}(m(k))$. Obtain the extrinsic information by $\lambda_1^{\text{out}}(m(k)) - \lambda_1^{\text{in}}(m(k)) - \lambda_0(m(k))$, which, after interleaving (using the code interleaver), is passed to constituent decoder 2 as the *a priori* input $\lambda_2^{\text{in}}(m(k))$.

Step 4. Run the PSAM-BCJR algorithm on constituent decoder 2, with the *a priori* input $\lambda_2^{\text{in}}(m(k))$, and obtain the output $\lambda_2^{\text{out}}(m(k))$. Obtain the extrinsic information by $\lambda_2^{\text{out}}(m(k)) - \lambda_2^{\text{in}}(m(k)) - \lambda_0(m(k))$, which, after deinterleaving (using the code interleaver), is passed to constituent decoder 2 as the *a priori* input $\lambda_1^{\text{in}}(m(k))$.

Step 5. Increase the iteration number by one. If the iteration number is less than or equal to the predefined maximum number of iterations, repeat step 3 and step 4. Otherwise, stop and output the soft decision as $\lambda_1^{\text{out}}(m(k))$ (the deinterleaved version of $\lambda_2^{\text{out}}(m(k))$).

The turbo decoding with the A-PSAM-BCJR algorithm is obtained by replacing the PSAM-BCJR algorithm in steps 3 and 4 by the A-PSAM-BCJR algorithm, and using the A-PSAM-LLR metric (4.19) to compute the initial LLR values $\lambda_0(m(k))$. In this scheme, the channel estimation accuracy is ignored and the channel estimates are assumed to be perfect.

6.4 Receiver Structure

Governed by the fact that the knowledge of the MMSE estimate and the MSE is required in order to implement the PSAM-BCJR or the A-PSAM-BCJR algorithm, the MMSE channel estimator is used at the receiver.

In this section, we will consider three types of receiver structures with MMSE channel estimator for turbo decoding via the PSAM-BCJR or A-PSAM-BCJR algorithm, namely, standard turbo decoding, ICED with hard decision feedback, and ICED with soft decision feedback.

6.4.1 Standard Turbo Decoding

The standard turbo decoder is shown in Fig. 6.1. After the signal sequence is received at the receiver, the data signals and the pilot signals are separated and demodulated. We assume that the channel gain over a particular transmission interval is correlated with the channel gains which are within the observation window of length $W = 2Q + 1$, and statistically independent to all the other channel gains. Therefore, to estimate the channel gain for a received signal, we only need to consider the nearest $2L$ pilot signals (L preceding and L succeeding), where $L = \lfloor \frac{Q}{B+1} \rfloor$. The MMSE estimate and the MSE can be obtained for each coded bit through the MMSE channel estimator using $2L$ received pilot signals. The received data signals, together with their corresponding channel estimates and the MSEs, are deinterleaved, and then passed to the turbo decoder. Turbo decoding with the PSAM-BCJR or A-PSAM-BCJR algorithm will be performed.

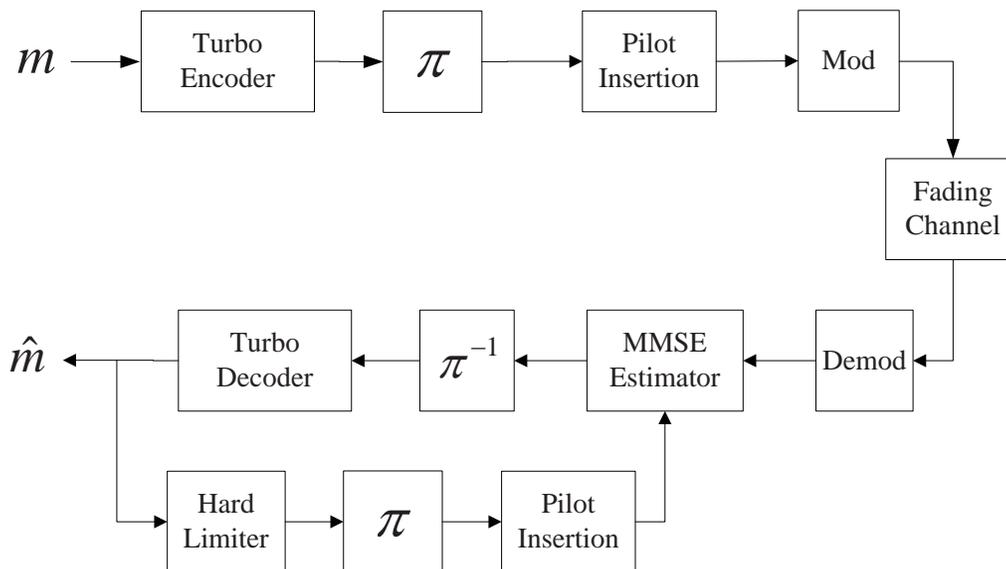


Figure 6.4: System model of iterative channel estimation and decoding with hard decision feedback over fading channels with PSAM channel estimation

After a predefined number of iterations is reached, the final decision is made. This process is referred to as the *standard turbo decoding*.

6.4.2 ICED with Hard Decision Feedback

The ICED with hard decision feedback is shown in Fig. 6.4. Initially, the standard turbo decoding, as described in section 6.4.1, is performed for several iterations. In the last iteration, we also compute the soft LLR decisions for each code bit $x^i(k)$ using the PSAM-BCJR algorithm with the following equation,

$$\lambda(x^i(k)) = \ln \frac{\sum_{(s,s') \in \Sigma_k^0(x^i(k))} p(s(k) = s', s(k+1) = s, \mathbf{r}|\mathbb{S})}{\sum_{(s,s') \in \Sigma_k^1(x^i(k))} p(s(k) = s', s(k+1) = s, \mathbf{r}|\mathbb{S})}, \quad (6.22)$$

where $\Sigma_k^0(x^i(k))$ (resp., $\Sigma_k^1(x^i(k))$) denotes the set of all state pairs (s', s) such that if the current state $s(k)$ is s' and the next state $s(k+1)$ is s , the i th branch

of the convolutional code will output bit-0 (resp., bit-1).

The soft tentative code-bit decisions will be passed through a hard limiter, where

$$\widehat{e^{j\pi x^i(k)}} = \begin{cases} 1 & : \text{ if } \lambda(x^i(k)) \geq 0 \\ -1 & : \text{ if } \lambda(x^i(k)) < 0 \end{cases}$$

Assume that the actual transmitted data signal for the code bit $x^i(k)$ is equal to $\sqrt{E_s} \widehat{e^{j\pi x^i(k)}}$. Now every data signal can also be treated as a pilot signal. After interleaving, pilot signals are reinserted. For the code bit $x^i(k)$, a new channel estimate and the corresponding MSE is computed using the MMSE estimator from the set of received signals, consisting of the received signal $r^i(k)$, the nearest \bar{L} received signal preceding $r^i(k)$ and the nearest \bar{L} received signals succeeding $r^i(k)$. With the updated channel estimates and MSEs, standard turbo decoding will be used for further processing. This process can be repeated several times before the receiver makes the final decision. We refer to this process as the *ICED with hard decision feedback*. Borrowing the terminology in Chapter 5.3, we will call the iteration within the turbo decoding as internal-iteration. The cycle from using standard turbo decoding to obtain tentative decisions to the moment when updated channel estimates are fed back to the turbo decoder will be called super-iteration.

6.4.3 ICED with Soft Decision Feedback

In the ICED with soft decision feedback, as shown in Fig. 6.5, a nonlinear limiter is used in the soft decision feedback, instead of a hard limiter. Following [146],

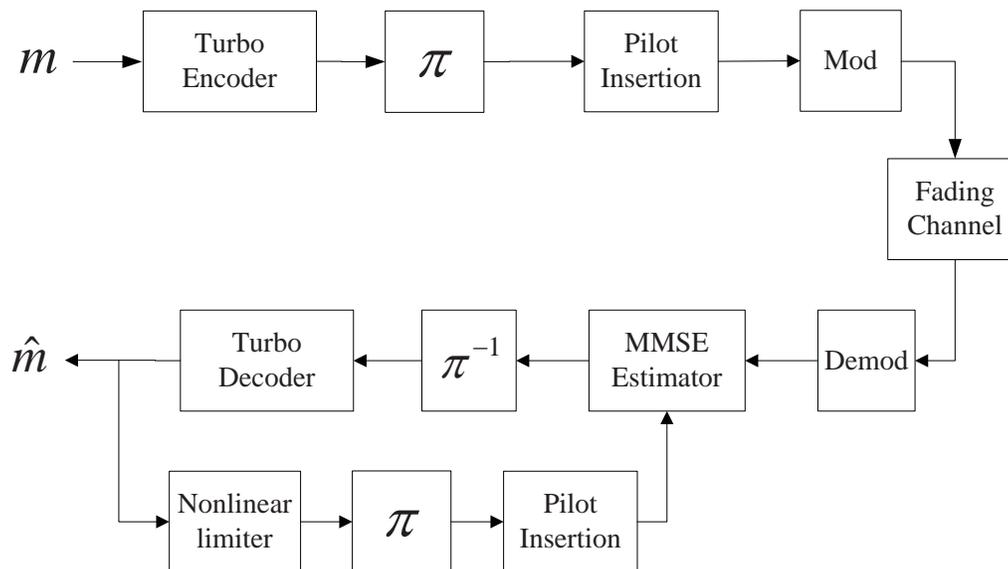


Figure 6.5: System model of iterative channel estimation and decoding with soft decision feedback over fading channels with PSAM channel estimation

the nonlinear function in the limiter is chosen as

$$\widehat{e^{j\pi x^i(k)}} = \tanh\left(\frac{\lambda(x^i(k))}{2}\right).$$

The other procedures in decoding are the same as those of the ICED with hard decision feedback.

6.5 Simulation Study and Discussion

In the simulation studies, the turbo code under consideration has the same structure as shown in Fig. 6.2, which is composed of two rate half recursive convolutional encoders, each of which has the generator polynomial given by $1 + D + D^3/1 + D + D^2 + D^3$. The trellis of the upper encoder is terminated

with 3 flushing bits, while the trellis of the lower encoder is left open. The data frame consists of 1021 data bits and 3 flushing bits. A pseudo-random interleaver is chosen as the code interleaver. In all the simulations, a block interleaver of size 128×120 is chosen as the channel interleaver. Thus, each interleaver block contains exactly five codewords. The standard turbo decoding is performed with a total of 12 iterations. In all simulations, we choose $L = 400$ and $\bar{L} = 40$. Each BEP data point is obtained from collecting at least 1000 bit errors.

In our simulations, we follow Jakes's isotropic scattering model [152] by assuming that the real and imaginary parts of $h(k)$ are independent, each with autocorrelation

$$R_c(i) = \sigma^2 J_0(2\pi f_d T_s i)$$

where f_d is the relative Doppler shift between the transmitter and the receiver, T_s is the symbol period and $J_0(\cdot)$ is the Bessel function of the first kind of order zero. The SNR in all the graphs refers to the energy per message bit E_b over the power N_0 of the AWGN of the fading channel.

6.5.1 Standard Turbo Decoding

In this section, we will study the performance of standard turbo decoding over flat Rayleigh fading channels with two normalized fade rates. The standard turbo decoding is carried out with a total of 12 iterations.

The BEP performance of standard turbo decoding using the PSAM-BCJR and the A-PSAM-BCJR algorithms is shown in Figs. 6.6(a) and (b). It can be

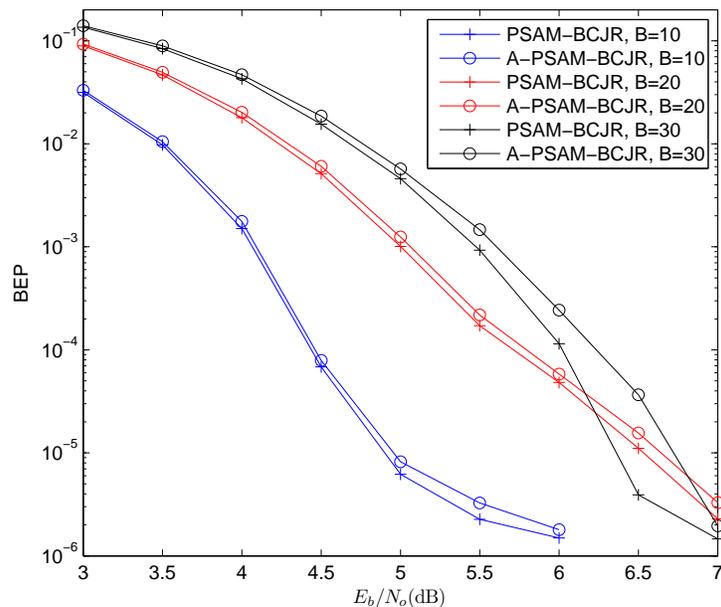
observed that the PSAM-BCJR outperforms the A-PSAM-BCJR, and the performance gain increases with the increase of the SNR. When the BEP is between 10^{-5} and 10^{-6} , error floors, which are inherited from the turbo code, are observed for both the PSAM-BCJR and the A-PSAM-BCJR. As a result, the two algorithms have similar performance at the error floor region.

In comparison, the BEP performance of standard turbo decoding using the PSAM-SOVA and the A-PSAM-SOVA algorithms is shown in Figs. 6.7(a) and (b). It is observed that, unlike the BCJR algorithms, the PSAM-SOVA and the A-PSAM-SOVA give similar error performance, which suggests that the channel estimation accuracy only has a minor effect. Moreover, as a sub-optimum MAP algorithm, the performance of the PSAM-SOVA is much worse than that of the PSAM-BCJR for standard turbo decoding over time-selective flat Rayleigh fading channels, as shown in Figs. 6.8(a) and (b).

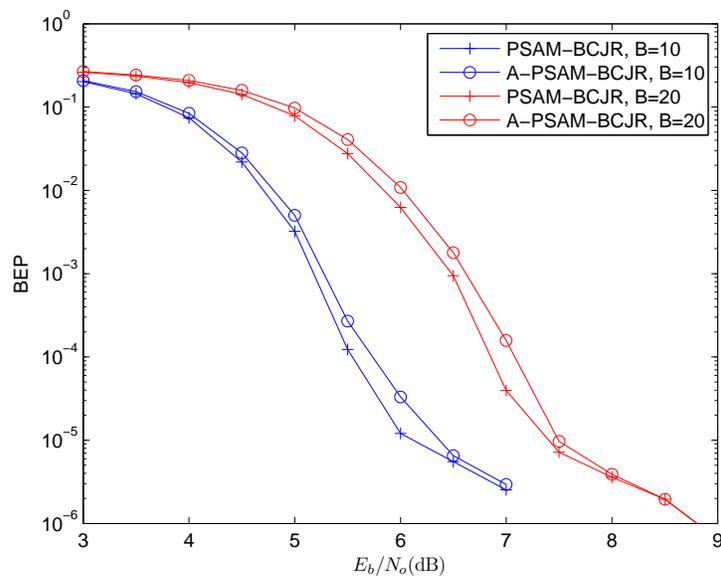
6.5.2 Iterative Channel Estimation and Decoding

In this section, we study and compare two types of receiver structures with iterative channel estimation and decoding. The first type is the two-stage decoding, similar to what we have considered in Section 4.4.7, Chapter 3, where the tentative decisions obtained after 5 initial turbo iterations are fed back for channel re-estimation, and with the updated channel estimated, another 7 turbo iterations are performed before making the final decision. The BEP performance for the two-stage iterative channel estimation and decoding of turbo codes with the PSAM-BCJR and the A-PSAM-BCJR algorithms is shown in Figs. 6.9(a) and

6. PSAM-BCJR ALGORITHM



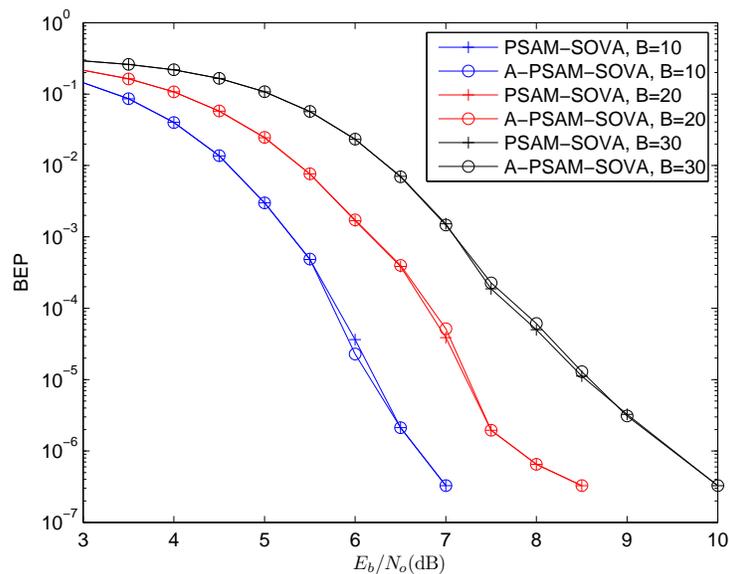
(a)



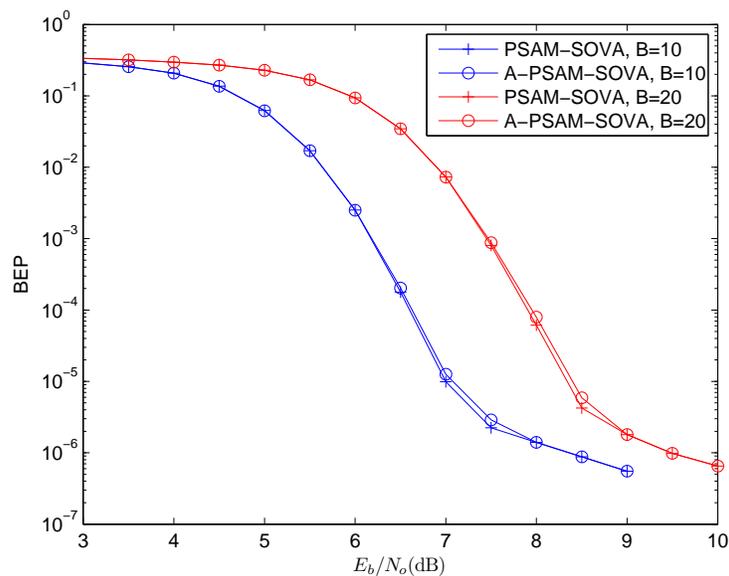
(b)

Figure 6.6: Performance comparison of standard turbo decoding between the PSAM-BCJR and A-PSAM-BCJR algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

6. PSAM-BCJR ALGORITHM



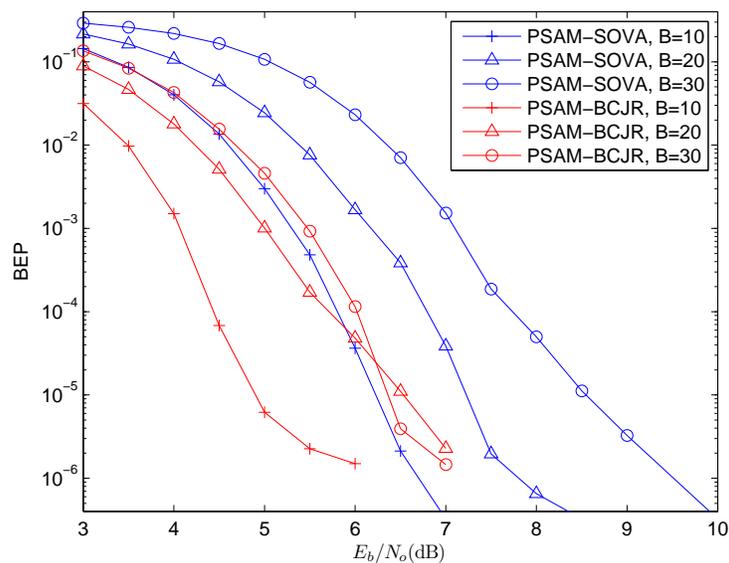
(a)



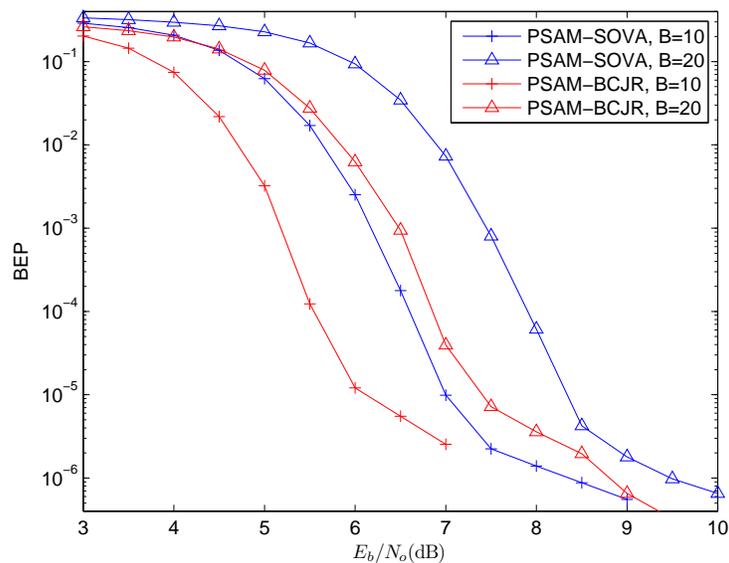
(b)

Figure 6.7: Performance comparison of standard turbo decoding between the PSAM-SOVA and A-PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

6. PSAM-BCJR ALGORITHM



(a)



(b)

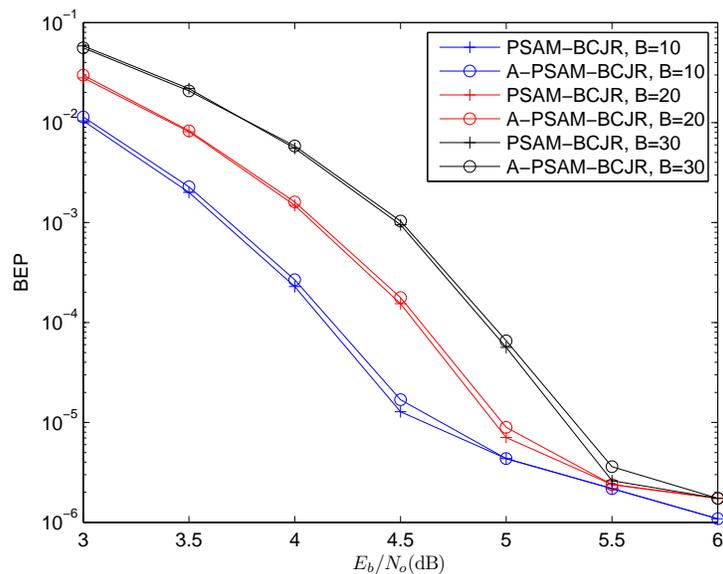
Figure 6.8: Performance comparison of standard turbo decoding between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

(b). It is observed that the PSAM-BCJR still gives better performance than the A-PSAM-BCJR, but the performance gain is only marginal. Error floors are observed for both algorithms. Similar observations are observed when the soft decision feedback is used, as shown in Figs. 6.10(a) and (b).

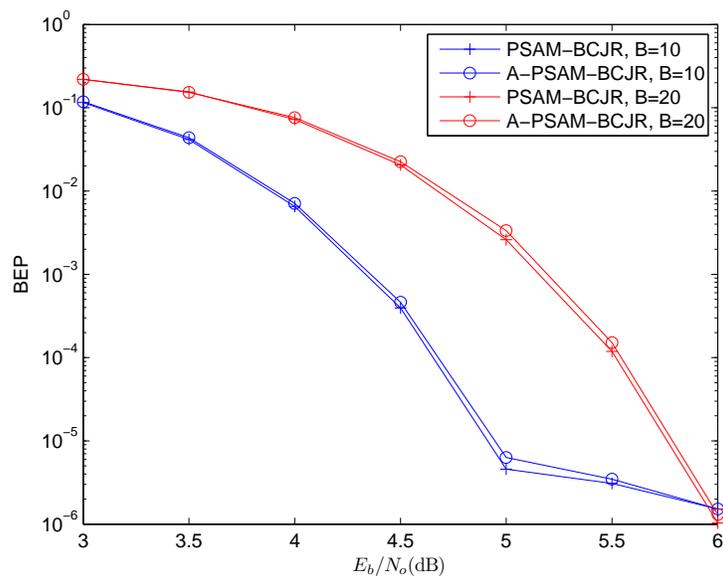
Next we will consider another type of iterative channel estimation and decoding structure, where the tentative decisions are fed back after every turbo iteration. Since only one internal-iteration is used in each super-iteration, we will name it as the *iterative channel estimation and decoding with one internal-iteration*. In our simulations, a total of 12 super-iterations are performed. As shown in Figs. 6.11(a) and (b) and Figs. 6.12(a) and (b), the PSAM-BCJR only has a marginal performance gain over the A-PSAM-BCJR. In the error floor region, the two algorithms have similar performance.

Note that compared to the standard turbo decoding scheme, the performance gain of the PSAM-BCJR over the A-PSAM-BCJR has decreased significantly when the iterative channel estimation and decoding technique is used. To explain this, we need to look into the feedback structures. When the tentative decisions are fed back, they are passed through a nonlinear limiter. It is assumed that the output sequence from the limiter is equal to the actual BPSK modulated signal sequence and all data signals are treated as pilot signals. New channel estimates and MSEs are computed based on the MMSE criterion from the received signals, which include both the pilot signals and the data signals. Since the tentative decisions are not perfect, data signals with erroneous decisions will be used. As a result, the channel estimates are less accurate than those obtained from trans-

6. PSAM-BCJR ALGORITHM



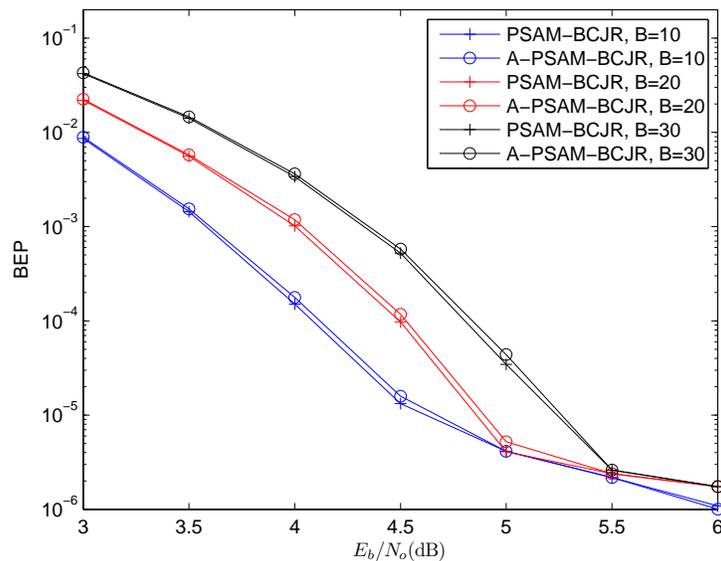
(a)



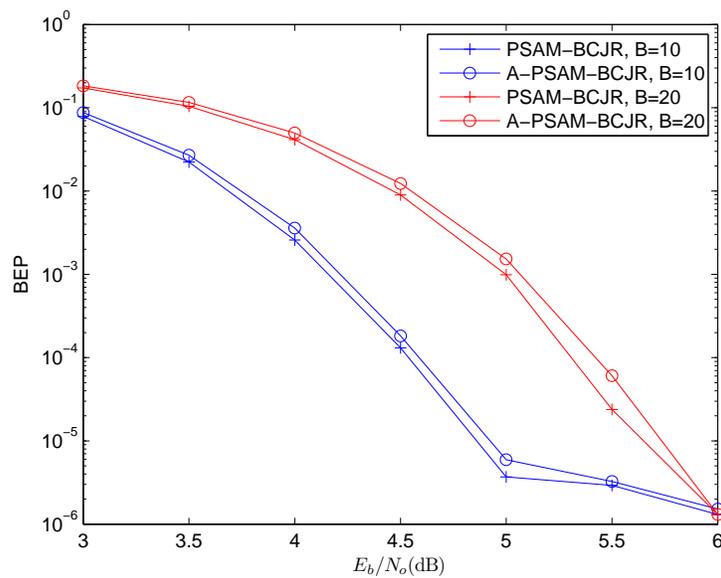
(b)

Figure 6.9: Performance comparison of two-stage ICED with hard decision feedback between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

6. PSAM-BCJR ALGORITHM



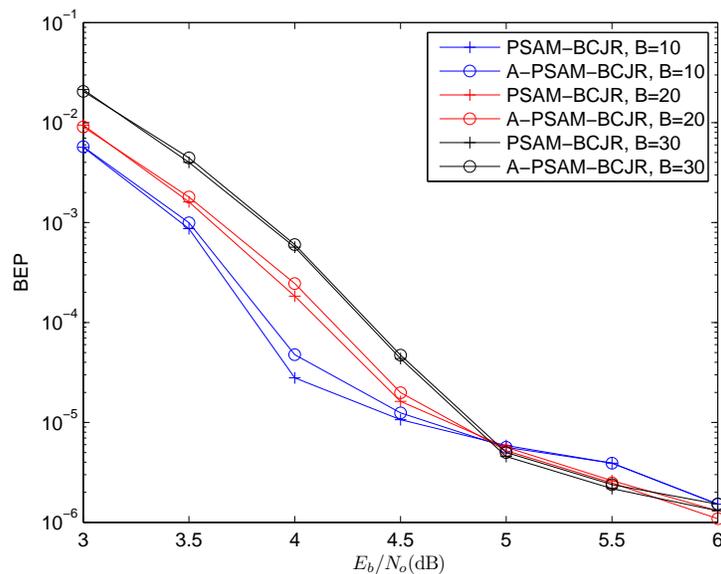
(a)



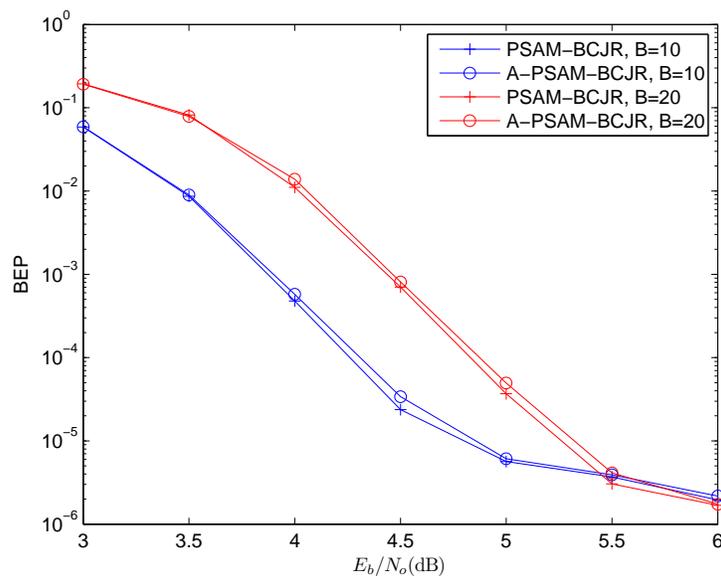
(b)

Figure 6.10: Performance comparison of two-stage ICED with soft decision feedback between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

6. PSAM-BCJR ALGORITHM



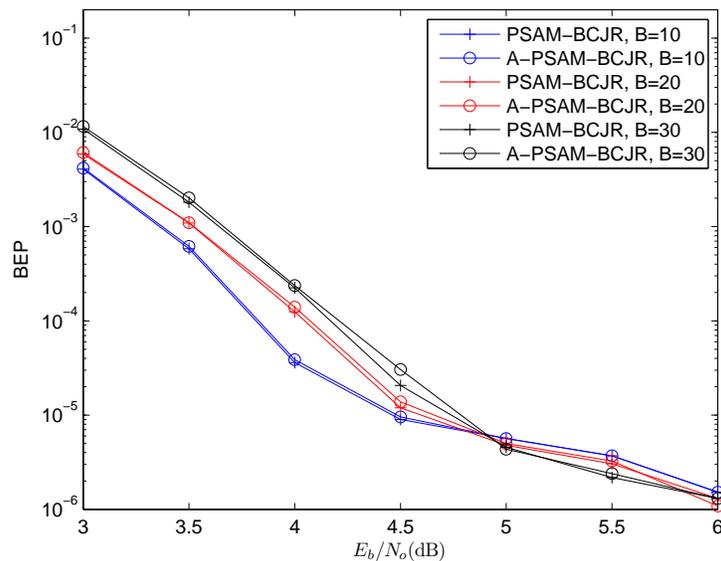
(a)



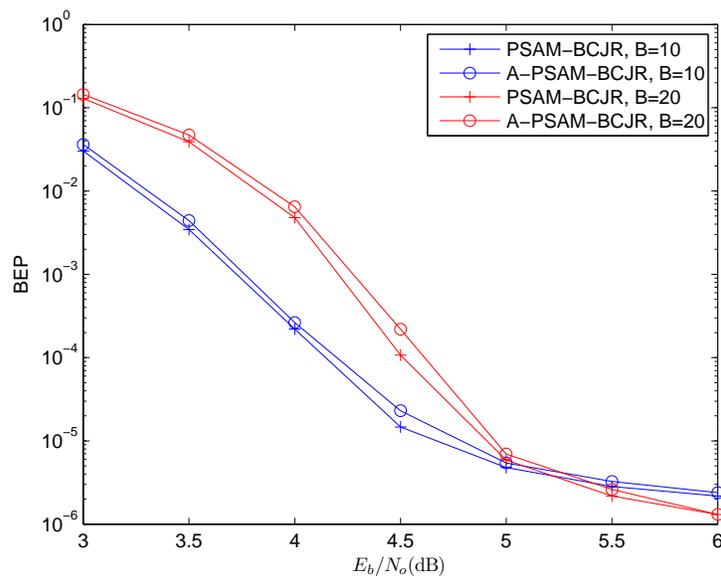
(b)

Figure 6.11: Performance comparison of ICED with hard decision feedback between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

6. PSAM-BCJR ALGORITHM



(a)



(b)

Figure 6.12: Performance comparison of two-stage ICED with soft decision feedback between the PSAM-BCJR and PSAM-SOVA algorithms over Rayleigh fading channels with various normalized fade rates: (a) $f_d T_s = 0.005$; (b) $f_d T_s = 0.02$.

mitting a stream of real pilot signals. More importantly, the estimation accuracy is over-estimated and the MSE is under-estimated. Performance degradation is expected from the PSAM-BCJR algorithm, which requires the precise knowledge of the MSE of the channel estimation. Therefore, the PSAM-BCJR only gives a marginal performance improvement over the A-PSAM-BCJR when iterative channel estimation and decoding with decision feedback is used.

6.5.3 Performance Comparison among Various Decoding Schemes

The performance comparison of the standard turbo decoding, the two-stage iterative channel estimation and decoding, and the iterative channel estimation and decoding with one internal-iteration is demonstrated in Fig 6.13 and 6.14. It is observed that the standard turbo decoding, which is the simplest scheme in complexity, performs the worst. A performance gain of 1 – 1.5 dB over the standard turbo coding is obtained when the two-stage iterative channel estimation and decoding is used. The iterative channel estimation and decoding with one internal-iteration requires the highest computational power, but it has the best BEP performance. In addition, the BEP performance with soft decision feedback is always better than that with hard decision feedback, albeit the performance gain is not significant. For all the schemes, error floors exist. In the error floor region, the four feedback receivers considered have similar performance.

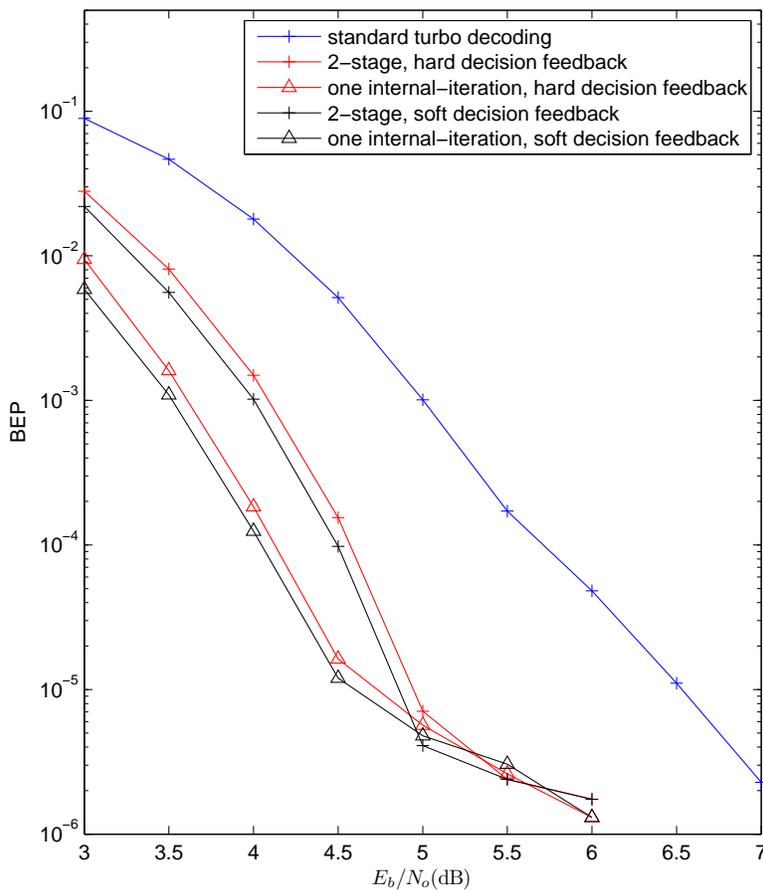


Figure 6.13: Performance comparison of standard turbo decoding, two-stage ICED and ICED with one internal-iteration over Rayleigh fading channels with normalized fade $f_d T_s = 0.005$ and pilot symbol spacing $B = 20$

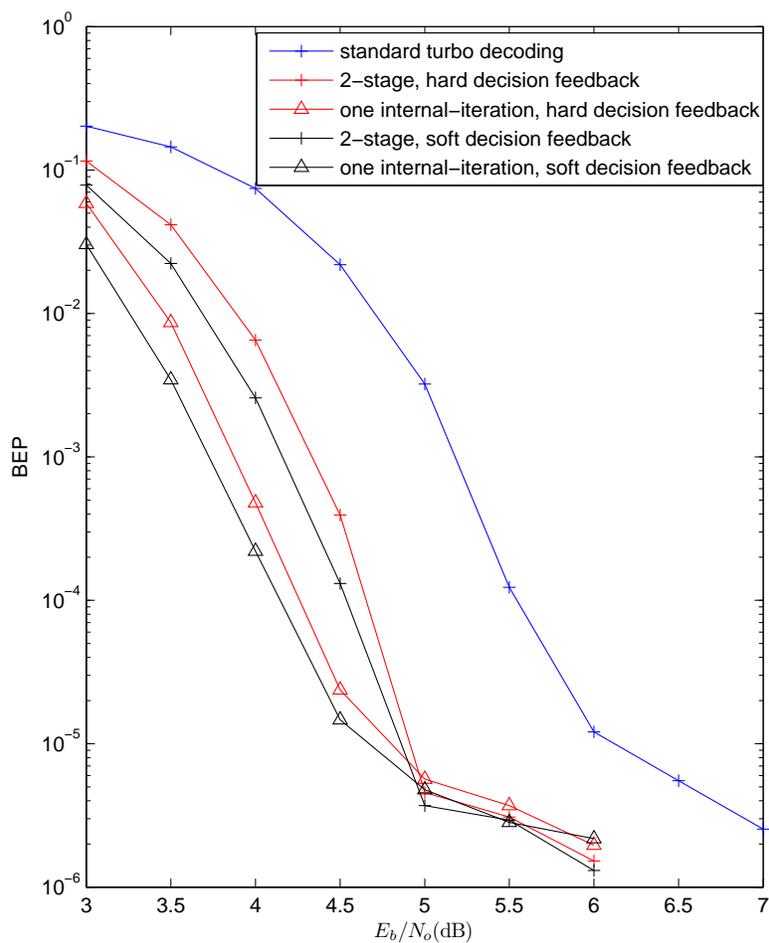


Figure 6.14: Performance comparison of standard turbo decoding, two-stage ICED and ICED with one internal-iteration over Rayleigh fading channels with normalized fade $f_d T_s = 0.02$ and pilot symbol spacing $B = 10$

6.6 Conclusions

We have proposed generalizations of the BCJR algorithm and the SOVA for the transmissions over time-selective, frequency-flat, Rayleigh fading channels with PSAM channel estimation, which could be used in turbo decoding as the SISO algorithms for the constituent convolutional codes. We show that the performance of the PSAM-BCJR is better than the A-PSAM-BCJR, which neglects the information regarding the channel estimation accuracy, the performance gain is not as significant as that for LDPC decoding, due to the existence of error floors of the turbo codes in relatively high BEP regions. However, we emphasize that this work demonstrates the correct conceptual approach of extending the BCJR algorithm and the SOVA to fading channels with PSAM channel estimation.

We also verify that the PSAM-BCJR algorithm achieves better BEP performance than the PSAM-SOVA for turbo decoding over time-selective, frequency-flat, Rayleigh fading channels with PSAM channel estimation, which agrees with the fact for the AWGN channel.

The technique of iterative channel estimation and decoding is studied in detail. We note that the performance gain of the PSAM-BCJR over the A-PSAM-BCJR decreases significantly when the iterative channel estimation and decoding technique is used, compared to that when the standard turbo decoding scheme is employed. The reason lies in the inaccuracy of the computation of the estimation MSE when the data signals are used as pilots in the feedback system. In fact, how to estimate the channel gain and obtain the exact MSE with decision feedback is

6. PSAM-BCJR ALGORITHM

a difficult and challenging task, which is still under active research. We will look into these problems in our future papers.

Chapter 7

Summary of Contributions and Suggestions for Future Work

7.1 Summary of Contributions

Over fading channels, the transmitted signals suffer from power attenuation and phase shift. To compensate for these distortions, the CSI is usually required at the receiver through PSAM. Accurate CSI acquisition requires sufficient number of pilot signals with significantly high transmission energy. However, in the SNR regions close to the Shannon limit, the energy available for the pilots is strictly limited. Hence, it is a challenging task to design energy-efficient transceivers over fading channels with reliable CSI acquisition.

In this thesis, we have considered several transceiver designs for the transmission of LDPC or turbo codes over fading channels. For the time-selective,

frequency-flat, Rayleigh fading channels, we present the correct conceptual approach for deriving the PSAM-LLR metric of a q -ary code with MPSK modulation and PSAM channel estimation. Unlike the suboptimum approaches in [56–65], which assume either structured channel estimators or perfect channel estimation, our derivation starts from first principles without assuming any receiver structure and demonstrates how the pilot information should be incorporated into the LLR computation. In particular, we show how the channel estimate and the estimation error variance enter in determining the reliability of each received coded symbol. Using a similar conceptual approach, we propose a generalization of the BCJR algorithm, which we call PSAM-BCJR. Through simulation studies, we show that the PSAM-LLR metric (resp., the PSAM-BCJR algorithm) can achieve better BEP performance than the conventional A-PSAM-LLR metric (resp., the A-PSAM-BCJR algorithm), which ignores the information concerning the channel estimation accuracy. Our work demonstrates the importance of incorporating the knowledge of the channel estimation accuracy in the iterative decoding process when KCS is available.

On the other hand, if KCS is not available or cannot be reliably acquired at the receiver, it will be difficult to obtain the channel estimate and estimation variance accurately. The transceiver with PSAM channel estimation is not robust. In this case, we propose the SOVA-ICSI detector, which combines the MLSD with the SOVA decoder. The SOVA-ICSI detector is based on maximum-likelihood sequence detection, which does not require explicit channel estimation. Moreover, the SOVA-ICSI detector does not require KCS, because the soft LLR output can

be computed solely based on the received signal sequence. Compared to the PSAM system in Chapter 4, the SOVA-ICSI detector is more robust and much less demanding, and thus it can be used more widely in real applications.

We have also proposed the DMD for LDPC codes over integer residue rings. We show through computer simulations that the DMD can achieve significant coding gains over standard BP decoding, while requiring significantly less computational power. Snapshots of the LLR densities are used to explain the superiority of the DMD over standard BP decoding.

7.2 Proposals for Future Research

7.2.1 Implementation of DMD over Fading Channels

For the nonbinary LDPC codes over \mathbb{Z}_{2^m} , we have designed the DMD for the transmission with BPSK modulation over the AWGN channel and shown that better BEP performance can be achieved with less computational complexity, compare to standard BP decoding.

In our future research, we will apply LDPC codes over \mathbb{Z}_{2^m} and the DMD to various types of fading channels. Firstly, we will study the case when LDPC codes over \mathbb{Z}_{2^m} are transmitted over the time-selective, frequency-flat, Rayleigh fading channels with PSAM channel estimation, whereby the *a posteriori* LLRs are computed using either the PSAM-LLR metric or the A-PSAM-LLR metric, which are derived in Chapter 4. The decoding performance of the DMD can be compared with that of the standard BP decoder. Also, we can investigate the

effect of estimation accuracy on the DMD, by comparing the BEP performances of the PSAM-LLR and the A-PSAM-LLR.

Likewise, we are also going to consider the case when LDPC codes over \mathbb{Z}_{2^m} and the DMD are applied with the SOVA-ICSI detector over block-wise static Rayleigh fading channels with unknown channel statistics.

7.2.2 Channel Estimation with Soft Decision Feedback

In Chapters 4 and 6, we discussed iterative channel estimation and decoding. We note that the performance gain of the PSAM-LLR (resp., the PSAM-BCJR) over the A-PSAM-LLR (resp., the A-PSAM-BCJR) decreases significantly when the iterative channel estimation and decoding technique is used, compared to that when the standard BP (resp., turbo) decoding scheme is adopted. The reason is that when the tentative decisions are fed back, it is assumed that these decisions are all correct and thus the data signals are treated as pilots. New channel estimates and MSEs are computed based on this assumption. As a result, the estimation accuracy is over-estimated and the MSE is significantly under-estimated. The PSAM-LLR and the probability density function in the PSAM-BCJR cannot be computed accurately, so only a marginal performance improvement is obtained when iterative channel estimation and decoding is used. This is leading us to investigate further on the correct methodology to compute the channel estimate and the MSE when decision feedback is used.

7.2.3 LP Decoding for Nonbinary Codes

Feldman et al. [128] proposed the LP approach to decode arbitrary binary linear codes and showed the close relationship between the LP decoding and the iterative BP decoding for LDPC codes. One desirable feature of the LP decoder is that it is more amenable to analysis. Its performance can be completely described in terms of pseudocodewords and some exact combinatorial characterizations of the conditions can be found to guarantee LP decoding success. More attractively, the LP decoder has the ML certificate property, i.e., if LP outputs an integral solution, it must be the ML codeword.

However, the complexity of LP decoding is usually very high. This disadvantage is exacerbated when LP is used to decode non-binary codes. References [147, 148] proposed some LP formulations for general non-binary codes, but they contain too many variables and constraints, which makes the decoding very costly. In our future studies, we will look into the problem of LP formulations for nonbinary codes over $\mathbb{GF}(q)$ and \mathbb{Z}_q and develop more efficient algorithms for LP decoding.

Appendix

In a QPSK-modulated SIMO system with Gray coding, we will show that the PSAM-LLR in (4.10) is equal to the S-PSAM-LLR in (4.11). This is equivalent to showing

$$\frac{\sum_{\phi(l,b) \in \Phi(l,b)|_{x(l,b,v)=0}} \exp(\psi_m)}{\sum_{\phi(l,b) \in \Phi(l,b)|_{x(l,b,v)=1}} \exp(\psi_m)} = \frac{\max_{\phi(l,b) \in \Phi(l,b)|_{x(l,b,v)=0}} \exp(\psi_m)}{\max_{\phi(l,b) \in \Phi(l,b)|_{x(l,b,v)=1}} \exp(\psi_m)} \quad (7.1)$$

The phase of constellation signals of QPSK with Gray coding can be expressed as

$$\phi(l, b) = \begin{cases} \pi/4, & \text{for } x(l, b, 1) = 0, x(l, b, 2) = 0 \\ 3\pi/4, & \text{for } x(l, b, 1) = 1, x(l, b, 2) = 0 \\ -\pi/4, & \text{for } x(l, b, 1) = 0, x(l, b, 2) = 1 \\ -3\pi/4, & \text{for } x(l, b, 1) = 1, x(l, b, 2) = 1 \end{cases}$$

For $v = 1$, the left hand side (LHS) of (7.1) is

$$\begin{aligned} \text{LHS} &= \frac{\exp(\text{Re}\{Ae^{j(\theta-\pi/4)}\}) + \exp(\text{Re}\{Ae^{j(\theta+\pi/4)}\})}{\exp(\text{Re}\{Ae^{j(\theta-3\pi/4)}\}) + \exp(\text{Re}\{Ae^{j(\theta+3\pi/4)}\})} \\ &= \frac{\exp(A \cos(\theta - \pi/4)) + \exp(A \cos(\theta + \pi/4))}{\exp(A \cos(\theta - 3\pi/4)) + \exp(A \cos(\theta + 3\pi/4))} \end{aligned}$$

where $A = \frac{2\sqrt{E_s}}{2\hat{\sigma}_m^2(l,b)E_s+N_{o,m}}|r_m(l,b)\hat{c}_m^*(l,b)|$ and $\theta = \angle r_m(l,b)\hat{c}_m^*(l,b)$. Using the trigonometric identity: $\cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta$, we get

$$\text{LHS} = \frac{\exp(A \cos \theta \cos \pi/4)[\exp(A \sin \theta \sin \pi/4) + \exp(-A \sin \theta \sin \pi/4)]}{\exp(A \cos \theta \cos 3\pi/4)[\exp(A \sin \theta \sin 3\pi/4) + \exp(-A \sin \theta \sin 3\pi/4)]}$$

Since $\sin \alpha = \sin(\pi - \alpha)$, we have

$$\text{LHS} = \frac{\exp(A \cos \theta \cos \pi/4)}{\exp(A \cos \theta \cos 3\pi/4)}$$

In a similar way, the right hand side (RHS) of (7.1) leads to

$$\begin{aligned} \text{RHS} &= \frac{\exp(A \cos \theta \cos \pi/4) \max\{\exp(A \sin \theta \sin \pi/4), \exp(-A \sin \theta \sin \pi/4)\}}{\exp(A \cos \theta \cos 3\pi/4) \max\{\exp(A \sin \theta \sin 3\pi/4), \exp(-A \sin \theta \sin 3\pi/4)\}} \\ &= \frac{\exp(A \cos \theta \cos \pi/4)}{\exp(A \cos \theta \cos 3\pi/4)} \end{aligned}$$

Therefore, equation (7.1) holds.

List of Publications

- [1]. H. Yuan and P.Y. Kam, “The LLR metric for q -ary LDPC codes with MPSK modulation over Rayleigh channels with imperfect CSI,” *IEEE Trans. on Commun.*, vol. 60, no. 7, pp. 1793–1799, Jul. 2012.

- [2]. H. Yuan and M.A. Armand, “Doubly multistage belief-propagation decoding of LDPC codes over \mathbb{Z}_{2^m} ,” to appear in *IET Commun.*

- [3]. H. Yuan and P.Y. Kam, “Log-likelihood ratios for LDPC codes with pilot-symbol-assisted BPSK transmission over flat Rayleigh fading channels,” in *Proc. IEEE VTC*, Anchorage, Alaska, USA, pp. 1–5, Sep. 2009.

- [4]. H. Yuan and P.Y. Kam, “The LLR computation via SOVA with implicit CSI,” to be submitted.

Bibliography

- [1] E. E. Shannon, “A mathematical theory of communication, ” *Bell Syst. Tech. J.*, pp. 379–423 (Part 1); pp. 623–656 (Part 2), Jun. 1948.
- [2] D. MacKay and R. Neal, “Near Shannon limit performance of low-density parity-check codes, ” *Elect. Lett.*, vol. 32, no. 8, pp. 1645–1646, Aug 1996.
- [3] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: turbo codes,” in *Proc. IEEE ICC*, pp. 1064–1070, May 1993.
- [4] R. G. Gallager, *Low Density Parity Check Codes. Research Monograph series*, Cambridge, MA: MIT Press, 1963.
- [5] T. J. Richardson, M. 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.

- [6] T. J. Richardson and R. Urbanker, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Info. Theory*, vol. 47, pp. 599–618, Feb. 2001.
- [7] S. Y. Chung, G. D. Forney, T.J. Richardson and R. Urbanker, “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.
- [8] D. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Info. Theory*, vol. 45, pp. 399–431, Mar. 1999.
- [9] M. C. Davey and D. J. C. MacKay, “Low density parity check codes over $GF(q)$,” *IEEE Commun. Lett.*, vol. 2, pp. 165–167, Jun. 1998.
- [10] X. Y. Hu and E. Eleftheriou, “Binary representation of cycle Tanner-graph $GF(2^q)$ codes,” in *Proc. IEEE ICC*, Paris, France, pp. 528–532, June 2004.
- [11] C. Poulliat, M. Fossorier and D. Declercq, “Design of non binary LDPC codes using their binary image: algebraic properties,” in *IEEE ISIT*, Seattle, pp. 93–97, USA, July 2006.
- [12] A. Bennatan and D. Burshtein, “Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels,” *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 549–583, Feb. 2006.
- [13] R. H. Peng and R. R. Chen, “Design of nonbinary LDPC codes over $GF(q)$ for multiple-antenna transmission,” in *Proc. MilCom*, pp. 1–7, Oct. 2006.

- [14] D. J. C. MacKay and M. C. Davey, "Evaluation of Gallager codes of short block length and high rate applications," in *Proc. IMA Int. Conf. Math. Its Appl.: Codes, Syst., Graphical Models*, New York, 2000, pp. 113–130.
- [15] L. Barnault and D. Declercq, "Fast decoding algorithm for LDPC over $GF(2^q)$," in *Proc. IEEE ITW*, Paris, France, pp. 70–73, Apr. 2003.
- [16] H. Song and J. R. Cruz, "Reduced complexity decoding of Q-ary LDPC codes for magnetic recording," *IEEE Trans. Magn.*, vol. 39, no. 2, pp. 1081–1087, Mar. 2003.
- [17] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-domain decoding of LDPC codes over $GF(q)$," in *Proc. IEEE ICC*, Paris, France, pp. 772–776, Jun. 2004.
- [18] D. Declercq and M. Fossorier, "Extended minsum algorithm for decoding LDPC codes over $GF(q)$," in *IEEE ISIT 2005*, pp. 464–468. Sep. 2005.
- [19] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over $GF(q)$," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.
- [20] G. Como and F. Fagnani, "Average spectra and minimum distances of low-density parity-check codes over abelian groups," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 1, pp. 19–53, 2008.
- [21] U. Erez and G. Millar, "The ML decoding performance of LDPC ensembles over Z_q ," in *Proc. IEEE ISIT*, Yokohama, Japan, Jul. 2003.

- [22] U. Erez and G. Millar, "The ML decoding performance of LDPC ensembles over Z_q ," *IEEE Trans. Inf. Theory*, vol. 51, pp. 1871–1879, May 2005.
- [23] D. Sridhara and T. E. Fuja, "Low density parity check codes over groups and rings," in *Proc. IEEE ITW*, pp. 163–166, Oct. 2002.
- [24] D. Sridhara and T. E. Fuja, "LDPC codes over rings for PSK modulation," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3209–3220, Sep. 2005.
- [25] A. Goupil, M. Colas, G. Gelle and D. Declercq, "FFT-based BP decoding of general LDPC codes over abelian groups," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 644–649, Apr. 2007.
- [26] M. A. Armand and K. S. Ng, "Decoding LDPC codes over integer residue rings," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4680–4686, Oct. 2006.
- [27] E. Mo and P. Y. Kam, "Log-likelihood metrics based on two-symbol-interval observations for LDPC codes with BDPSK transmission," in *Proc. IEEE VTC*, Calgary, Alberta, pp. 1–5, Sep. 2008.
- [28] E. Mo, P. Y. Kam and M. A. Armand, "LLR metrics for LDPC Codes with quadrature differential PSK transmission, and their performances," in *Proc. IEEE ISITA*, pp. 1–6, Auckland, New Zealand, Dec. 2008.
- [29] E. K. Hall and S. G. Wilson, "Turbo codes on noncoherent channels," in *Commun. Theory Mini-Conf. Rec. IEEE Global Telecommun. Conf.*, Phoenix, Arizona, pp. 66–70, 1997.

- [30] E. Mo and P. Y. Kam, “Log-likelihood ratios for LDPC codes with pilot-symbol-assisted BPSK transmission over the noncoherent channel,” in *Proc. IEEE WCNC*, Budapest, Hungary, pp. 1–6, Apr. 2009.
- [31] S. Sahuguede, A. Julien-Vergonjanne, J.P. Cances, “Soft Decision LDPC Decoding Over Chi-Square Based Optical Channel,” *Journal of Lightwave Technology*, vol. 27, no. 16, pp. 3540–3545, Aug. 2009.
- [32] R. Yazdani and M. Ardakani, “Linear LLR approximation for iterative decoding on wireless channels,” *IEEE Trans. Commun.*, vol. 57, no. 11, pp. 3278–3287, Nov. 2009.
- [33] Y. Li, “Simplified channel estimation for OFDM systems with multiple transmit antennas,” *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 67–75, Jan. 2002.
- [34] C. Fragouli, N. Al-Dhahir and W. Turin, “Training-based channel estimation for multiple-antenna broadband transmissions,” *IEEE Trans. Wireless Commun.*, vol. 2, no. 2, pp. 384–391, Mar. 2003.
- [35] B. Hassibi and B. M. Hochwald, “How much training is needed in multiple-antenna wireless links?,” *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, Apr. 2003.
- [36] S. Ohno and G. B. Giannakis, “Capacity maximizing MMSE Optimal pilots for wireless OFDM over frequency-selective block Rayleigh-fading channels,” *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 2138–2145, Sep. 2004.

- [37] X. Ma, G. B. Giannakis, and S. Ohno, "Optimal training for block transmissions over doubly-selective wireless fading channels," *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 1351–1366, May 2003.
- [38] V. T. Nam, P. Y. Kam and Y. Xin, "LDPC codes with BDPSK and differential detection over flat Rayleigh fading channels," in *Proc. IEEE GlobeCom*, Washington D.C. USA, pp. 1–5, Nov. 2007.
- [39] V. T. Nam, P. Y. Kam and Y. Xin, "Iterative decoding of BDPSK modulated LDPC codes using two-symbol-interval observations," in *Proc. IEEE ISIT*, Auckland, New Zealand, pp. 1–5, Dec. 2008.
- [40] S. G. Wilson, J. Freebersyser and C. Marshall, "Multi-symbol differential detection of MPSK," in *Proc. IEEE GLOBECOM*, pp. 1692–1697, Dallas, TX, Nov. 1989.
- [41] D. Divsalar and M. K. Simon, "Multiple-symbol differential detection of MPSK," *IEEE Trans. Commun.*, vol. 38, no. 3, pp. 300–308, Mar. 1990.
- [42] M. K. Simon and M. S. Alouini, "Multiple symbol differential detection with diversity reception," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1312–1319, Aug. 2001.
- [43] J. K. Tugnait, L. Tong and Z. Ding, "Single-user channel estimation and equalization," *IEEE Signal Processing Magazine*, vol. 17, no. 1, pp. 17–28, May. 2000.

- [44] C. Tepedelenlioglu and G. B. Giannakis, "Transmitter redundancy for blind estimation and equalization of time- and frequency-selective channels," *IEEE Trans. Signal Process.*, vol. 48, no. 7, pp. 2029–2043, Jul. 2000.
- [45] J. K. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 686–963, Nov. 1991.
- [46] S. Sampei and T. Sunaga, "Rayleigh fading compensation for QAM in land mobile radio communications," *IEEE Trans. Veh. Technol.*, vol. 42, no. 2, pp. 137–147, May 1993.
- [47] H. Li, Y. Iwanami and T. Ikeda, "Symbol error rate analysis for MPSK under Rician fading channels with fading compensation based on time correlation," *IEEE Trans. Veh. Technol.*, vol. 44, no. 3, pp. 535–542, Aug. 1995.
- [48] J. P. Seymour and M. P. Fitz, "Near-optimal symbol-by-symbol detection schemes for flat Rayleigh fading," *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 1525–1533, Feb. 1995.
- [49] E. Okamoto, H. Li and T. Ikegami, "Rayleigh fading compensation for 16QAM using FFT," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1626–1633, Sep. 1999.
- [50] M. Dong, L. Tong, and B. M. Sadler, "Optimal insertion of pilot symbols for transmissions over time-varying flat fading channels," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1403–1418, May 2004.

- [51] Y. Chen and N. C. Beaulieu, "Optimum pilot symbol assisted modulation," *IEEE Trans. Commun.*, vol. 55, no. 8, pp. 1536–1546, Aug. 2007.
- [52] A. Maaref and S. Aissa, "Optimized rate-adaptive PSAM for MIMO MRC systems with transmit and receive CSI imperfections," *IEEE Trans. Commun.*, vol. 57, pp. 821–830, Mar 2009.
- [53] G. Taricco and E. Biglieri, "Space-time decoding with imperfect channel estimation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1874–1888, Jul. 2005.
- [54] Y. Li, "Pilot-symbol-aided channel estimation for OFDM in wireless systems," *IEEE Trans. Veh. Technol.*, vol. 49, no. 4, pp. 1207–1215, Jul. 2000.
- [55] R. Baxley, J. Kleider and G. Zhou, "Pilot design for OFDM with null edge subcarriers," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 396–405, Jan. 2009.
- [56] K. Fu and A. Anastasopoulos, "Analysis and design of LDPC codes for time-selective complex-fading channels," *IEEE Trans. Wireless Commun.*, pp. 1175–1185, May 2005.
- [57] H. Niu, M. Shen, J. A. Ritcey and H. Liu "A factor graph approach to iterative channel estimation and LDPC decoding over fading channels," *IEEE Trans. Wireless Commun.*, pp. 1345–1350, Jul. 2005.

- [58] C. Komminakis and R. Wesel, “Joint iterative channel estimation and decoding in flat correlated Rayleigh fading,” *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1706–1717, Sep 2001.
- [59] M. Valenti and B. Woerner, “Iterative channel estimation and decoding of pilot symbol assisted turbo codes over flat-fading channels,” *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1697–1705, Sep 2001.
- [60] H. J. Su and E. Geraniotis, “Low-complexity joint channel estimation and decoding for pilot symbol-assisted modulation and multiple differential detection systems with correlated Rayleigh fading,” *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 249–261, Feb. 2002.
- [61] R. Otnes and M. Tüchler, “Iterative channel estimation for turbo equalization of time-varying frequency selective channels,” *IEEE Trans. Wireless Commun.*, pp. 146–156, Nov. 2004.
- [62] M. F. Flanagan and A. D. Fagan, “Iterative channel estimation, equalization, and decoding for pilot-symbol assisted modulation over frequency selective fast fading channels,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 4, pp. 1661–1670, Jul. 2007.
- [63] F. Ling, “Optimal turbo decoding metric generation in a pilot assisted coherent wireless communication system,” *VTC 2000*, Boston, pp. 298–302, Sept. 2000.

- [64] M. Wang, W. Xiao, and T. Brown, "Soft decision metric generation for QAM with channel estimation error," *IEEE Trans. Commun.*, vol. 50, no. 7, pp. 1058–1061, Jul. 2002.
- [65] M. Benjillali, and L. Szczecinski, "On Reliability Metrics for Soft-Input Decoding in Presence of Channel Estimation Errors," *VTC 2006-Fall*, Montreal, pp. 2548–2552, Sept. 2006.
- [66] P. Y. Kam and C. H. Teh, "Reception of PSK signals over fading channels via quadrature amplitude estimation," *IEEE Trans. Commun.*, vol. 31, pp. 1024–1027, Aug. 1983.
- [67] P. Y. Kam and C. H. Teh, "An adaptive receiver with memory for slowly fading channels," *IEEE Trans. Commun.*, vol. 32, pp. 654–659, Jun. 1984.
- [68] P. Y. Kam, "Adaptive diversity reception over a slow nonselective fading channel," *IEEE Trans. Commun.*, vol. 35, pp. 572–574, May 1987.
- [69] P. Y. Kam, "Maximum-likelihood digital data sequence estimation over the Gaussian channel with unknown carrier phase," *IEEE Trans Commun.*, vol. 35, no. 7, pp. 764–767, Jul. 1987.
- [70] M. Wu and P. Y. Kam, "Performance analysis and computational complexity comparison of sequence detection receivers with no explicit channel estimation," in *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp.2625–2631, Jun. 2010.
- [71] D. Warrior and U. Madhow, "Spectrally efficient noncoherent communication," *IEEE Trans. Inf. Theory*, vol. 48, no. 3, pp. 651–668, Mar. 2002.

- [72] N. Varnica, M. P. C. Fossorier, K. Aleksandar, “Augmented belief propagation decoding of low-density parity-check codes,” *IEEE Trans. Commun.*, vol. 55, pp. 1308–1317, Jul. 2007.
- [73] M. Fossorier, M. Mihaljevic, and H. Imai, “Reduced complexity iterative decoding of low density parity check codes based on belief propagation,” *IEEE Trans. on Commun.*, vol. 47, pp. 673–680, May 1999.
- [74] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, “On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes,” *IEEE Trans. on Commun.*, vol. 53, pp. 549–554, Apr. 2005.
- [75] J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” in *Proc. IEEE Globecom*, pp. 1680–1686, Nov. 1989
- [76] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Info. Theory*, vol. 20(2), pp. 284–287, Mar. 1974.
- [77] R. Koetter, A. C. Singer, and M. Töhler, “Turbo equalization,” *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp.67–80, Jan. 2004.
- [78] H. Shi, J. Wu and Z. Wu, “Carrier phase recovery for Turbo codes in Rayleigh fading channel,” *WiCom 2007*, Shanghai, pp. 1180–1183, Sept. 2007.
- [79] M. Nicoli, S. Ferrara and U. Spagnolini, “Soft-iterative channel estimation: methods and performance analysis,” *IEEE Trans. Signal Proc.*, vol. 55, no. 6, pp. 2993–3006, Jun. 2007.

- [80] S. S. Sandhu, A. Rawal, P. Kaur and N. Gupta, “Major components associated with green networking in Information Communication Technology systems,” in *Proc. IEEE ICCCA*, Dindigul, Tamilnadu, India, pp. 1–6, Feb. 2012.
- [81] R. W. Hamming, “Error detecting and error correcting codes,” *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.
- [82] D. E. Muller, “Applications of boolean algebra to switching circuits design and to error detection,” *IRE Trans.*, no. 3, pp. 6–12, Sep. 1954.
- [83] I. S. Reed, “A class of multiple-error-correcting codes and the decoding scheme,” *IRE Trans.*, no. 4, pp. 38–49, Sep. 1954.
- [84] A. Hocquenghem, “Codes correcteurs d’erreurs,” *Chiffres*, no. 2, pp. 147–156, Sep. 1959
- [85] R. C. Bose, and D. K. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Inform. Control*, no. 3, pp. 68–79, Mar. 1960
- [86] I. S. Reed and G. Solomon, “Polynomial codes over certain fields,” *J. Soc. Ind. Appl. Math.*, no. 8, pp. 300–304, Jun. 1960
- [87] P. Elias, “Coding for noisy channels,” *IRE Conv. Rec.*, no. 4, pp. 37–47, 1955.
- [88] G. Battail, M. Decouvelaere and P. Godlewski, “Replication decoding,” *IEEE Trans. Info. Theory*, vol. IT-25, pp. 332–345, May 1979.

- [89] G. Battail, “Building long codes by combination of simple ones, thanks to weighted-output decoding,” in *Proc. ISSSE 1989*, pp. 634–637, Sep. 1989.
- [90] H. h. Lodge, P. Hoeher and J. Hagenauer, “The decoding of multidimensional codes using separable MAP filters,” in *Proc. 16th Biennial Symp. Commun.*, pp. 343–346, May 1992.
- [91] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo-codes,” *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.
- [92] C. Berrou and A. Glavieux, “Reflections on the prize paper: near optimum error correcting coding and decoding: turbo-codes,” *IEEE Inform. Theory Soc. Newsletter*, vol. 48(2), pp. 24–31, Jun. 1998.
- [93] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, “Analysis, design and iterative decoding of doubly serially concatenated codes with interleavers,” *IEEE J. Select. Areas Commun.*, vol. 42, pp. 231–244, Feb. 1998.
- [94] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, “Serial concatenation of interleaved codes: performance analysis, design and iterative decoding,” *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.
- [95] F. Garin and F. Fagnani, “Analysis of serial turbo codes over Abelian groups for symmetric channels,” *SIAM J. Discr. Math.*, vol. 22, no. 4, pp. 1488–1526, Oct. 2008.

- [96] F. Garin, G. Como and F. Fagnani, "The performance of serial turbo codes does not concentrate," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2570–2588, May 2012.
- [97] D. Divsalar and F. Pollara, "Serial and hybrid concatenated codes with applications," in *Proc. Int. Symp. Turbo Codes Related Topics*, Brest, France, pp. 80–87, Sep. 1997.
- [98] A. Graell i Amat and E. Rosnes, "Good concatenated code ensembles for the binary erasure channel," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 928–943, Jul. 2009.
- [99] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta, K. Sh. Zigangirov and D. J. Costello Jr., "Analysis and design of tuned turbo codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4796–4813, Jul. 2012.
- [100] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Info. Theory*, vol. 42, pp. 409–428, Mar. 1996.
- [101] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, pp. 591–600, May 1996.
- [102] L. C. Perez, J. Seghers and D. J. Costello, Jr., "A distance spectrum interpretation of turbo codes," *IEEE Trans. Info. Theory*, vol. 42, pp. 1698–1709, Nov. 1996.

- [103] G. Battail, “A conceptual framework for understanding turbo codes,” *IEEE J. Select. Areas Commun.*, vol. 16, pp. 245–254, Feb. 1998.
- [104] J. Hagenauer, E. Offer and L. Papke, “Iterative decoding of binary block and convolutional codes” *IEEE Trans. Info. Theory*, vol. 42, pp. 429–445, Mar. 1996.
- [105] R. J. McEliece, D. J. C. MacKay and J. F. Cheng, “Turbo decoding as an instance of Pearl’s belief propagation algorithm” *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, Feb. 1998.
- [106] A.J. Viterbi, “An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes,” *IEEE J. Select. Areas Commun.*, vol. 16, pp. 260–264, Feb. 1998.
- [107] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sep. 1981.
- [108] T. Richardson, A. Shokrollahi and R. Urbanke, “Design of provably good low-density parity check codes,” in Proc. IEEE ISIT, Sorrento, Italy, p. 199, Jun. 2000,
- [109] Y. Kou, S Lin and M. Fossorier, “Low density parity check codes based on finite geometries: a rediscovery and more,” *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2711–2736, Nov. 2001.

- [110] S. Y. Chung, T. J. Richardson and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.
- [111] S. Brink, “Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes,” *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.
- [112] S. Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection,” *IEEE Trans. Commun.*, vol. 52, pp. 670–678, Apr. 2004.
- [113] A. Ashikhmin, G. Kramer, and S. Brink, “Extrinsic information transfer functions: Model and erasure channel properties,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 2657–2673, Nov. 2004.
- [114] H. Jin, A. Khandekar, and R. McEliece, “Irregular repeat-accumulate codes,” in *Proc. International Symposium on Turbo codes and Related Topics*, Sept. 2000, pp. 1-8.
- [115] M. Yang, W. E. Ryan, and Y. Li, “Design of efficiently encodable moderate-length high-rate irregular LDPC codes,” *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 564–571, Apr. 2004.
- [116] Y. Zhang and W. E. Ryan, “Structured IRA codes: performance analysis and construction,” *IEEE Trans. Commun.*, vol. 55, no. 5, pp. 837–844, May 2007.

- [117] L. Chen, J. Xu, I. Djurdjevic and S. Lin, “Near Shannon limit quasi-cyclic low-density parity-check codes,” *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1038–1042, July 2004.
- [118] J. Kang, Q. Huang, L. Zhang, B. Zhou and S. Lin, “Quasi-cyclic LDPC codes: An algebraic construction,” *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1383–1396, May 2010.
- [119] L. Zhang, S. Lin, K. Abdel-Ghaffar, Z. Ding and B. Zhou, “Quasi-cyclic LDPC codes on cyclic subgroups of finite fields,” *IEEE Trans. Commun.*, vol. 59, no. 9, pp. 2330–2336, Sep. 2011.
- [120] J. Campello, D. S. Modha and S. Rajagopalan, “Designing LDPC codes using bit-filling,” in Proc. IEEE ICC, Helsinki, Finland, pp. 55–59, Jun. 2001.
- [121] H. Xiao and A. H. Banihashemi, “Improved progressive-edge-growth (PEG) construction of irregular LDPC codes,” *IEEE Commun. Lett.*, vol. 8, no. 12, pp. 715–718, Dec. 2004.
- [122] Y. Hu, E. Eleftheriou, and D. M. Arnold, “Regular and irregular progressive edge growth Tanner graphs,” *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [123] D. MacKay’s Database, [online]. Available: <http://www.inference.phy.cam.ac.uk/mackay/codes/data.html>.

- [124] Y. Kou, S. Lin and M. Fossorier, “Low density parity check codes based on finite geometries: a rediscovery and new results,” *IEEE Trans Inform. Theory*, vol. 47, no.11, pp. 2711–2736, Nov. 2001.
- [125] N. Wiberg, “Codes and decoding on general graphs,” *Ph.D. dissertation*, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1996.
- [126] J. Chen and M. Fossorier, “Density evolution for two improved BP-based decoding algorithms of LDPC codes,” *IEEE Commun. Lett.*, vol. 6, pp. 208–210, May 2002.
- [127] J. Chen and M. Fossorier, “Density evolution for BP-based decoding algorithms of LDPC codes and their quantized versions,” in *Proc. IEEE Globecom*, pp. 1378–1382, Nov. 2002.
- [128] J. Feldman, M.J. Wainwright, and D.R. Karger, “Using linear programming to decode binary linear codes,” *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.
- [129] A.J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Info. Theory*, vol. 13, pp. 260–269, Apr. 1967.
- [130] P. Jung, “Novel low complexity decoder for turbo-codes,” *emphElectron. Lett.*, vol. 31, pp. 86–87, Jan. 1995.

- [131] G. H. Norton and A. Salagean, “On the Hamming distance of linear codes over a finite chain ring,” *IEEE Trans. Inform. Theory*, vol. 56, no. 3, pp. 1060–1067, May 2000.
- [132] M. Greferath and U. Vellbinger, “Efficient decoding of \mathbf{Z}_p^k -linear codes,” *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 1288–1291, May 1998.
- [133] F. Hemmati and D. J. Costello, “Truncation error probability in Viterbi decoding,” *IEEE Trans. Commun.*, vol. 25, pp. 530–532, May 1977.
- [134] I. M. Onyszchuk, “Truncation length for Viterbi decoding,” *IEEE Trans. Commun.*, vol. 39, pp. 1023–1026, Jul. 1991.
- [135] D. J. Ryan, I. V. L. Clarkson and I. B. Collings, “Detection error probabilities in noncoherent channels ,” *ISIT 2005.*, pp. 617–621.
- [136] Y. Tan, J. Guo, Y. Ai, W. Liu and Y. Fei, “A coded modulation scheme for deep-space optical communications,” *IEEE Photonics Technology Lett.*, vol. 20, no. 5, pp. 372–374, Mar. 2008.
- [137] J. T. Seong and H. N. Lee, “Concatenation of LDPC codes with Golden space-time block codes over the block fading MIMO channels: system design and performance analysis,” *CISS 2011*, pp. 23–25, Mar. 2011. pp. 1697–1705, Sep 2001.
- [138] M. Franceschini, G. Ferrari, R. Raheli and A. Curtoni, “Serial concatenation of LDPC codes and differential modulations,” *IEEE J. Select. Areas Commun.*, vol. 23, no. 9, pp. 1758–1768, Sept. 2005.

- [139] J. Gunther, D. Keller, and T. Moon, "A generalized BCJR algorithm and its use in turbo synchronization," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Mar. 2005, vol. III, pp. 837–840.
- [140] J. Gunther, D. Keller, and T. Moon, "A generalized BCJR algorithm and its use in iterative blind channel identification," *IEEE Signal Processing Lett.* vol. 14, no. 10, pp.661–664, Oct. 2007.
- [141] V. Franz and J. B. Anderson, "Concatenated decoding with a reduced search BCJR algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 186–195, Feb. 1998.
- [142] C. Fragouli, N. Seshadri and W. Turin, "Reduced-trellis equalization using the BCJR algorithm," *Wireless Commun. Mobile Computing*, vol. 1, pp. 397–406, 2001.
- [143] G. Colavolpe, G. Ferrari and R. Raheli, "Reduced-state BCJR type algorithms," *IEEE J. Sel. Areas Commun.*, vol. 19, pp. 848–859, May 2001.
- [144] D. Fertonani, A. Barbieri and G. Colavolpe, "Reduced-complexity BCJR algorithm for turbo equalization," *IEEE Trans. Commun.*, vol. 55, pp. 2279–2287, Dec. 2007.
- [145] C. Vithanage, C. Andrieu, and R.J. Piechocki, "Reduced-State BCJR Algorithms," *IEEE Trans. Commun.*, vol. 55, pp.1133–1152, Jun. 2007.

- [146] M. Sandell, C. Luschi, P. Strauch and R. Yan, “Iterative channel estimation using soft decision feedback,” in *Proc. IEEE GLOBECOM*, Sydney, Australia, Nov. 1998.
- [147] M.F. Flanagan, V. Skachek, E. Byrne and M. Greferath, “Linear programming decoding of non-binary linear codes”, *Proc. 7th Int. ITG Confer. on Source and Channel Coding (SCC)*, Jan. 2008, Ulm, Germany.
- [148] M.F. Flanagan, V. Skachek, E. Byrne and M. Greferath, “Polytope representations for linear-programming decoding of non-binary linear codes”, *Proc. IEEE Int. Symp. Information Theory and Its Applications, ISITA 2008*, Toronto, Canada, Jul. 2008.
- [149] X. Tang, M. S. Alouini and A. J. Goldsmith, “Effect of channel estimation error on M-QAM BER performance in Rayleigh fading,” *IEEE Trans. Commun.*, vol. 47, no. 12, pp. 1856–1864, Dec. 1999.
- [150] B. Xia and J. Wang, “Effect of channel-estimation error on QAM systems with antenna diversity,” *IEEE Trans. Commun.*, vol. 53, no. 3, pp. 481–488, Mar. 2005.
- [151] X. Dong and L. Xiao, “Symbol error probability of two-dimensional signaling in Ricean fading with imperfect channel estimation,” *IEEE Trans. Veh. Technol.*, vol. 54, no. 2, pp. 538–549, Mar. 2005.
- [152] W. C. Jakes, *Mobile Microwave Communication*. New York: Wiley, 1974.
- [153] J. G. Proakis, *Digital Communications*, fourth ed. McGraw-Hill, 2000.

- [154] M. K. Simon and M. S. Alouini, *Digital Communication over Fading Channels*, second ed., John Wiley & Sons, 2005.
- [155] A Goldsmith, *Wireless Communications*, first ed., Cambridge University Press, 2005.
- [156] A. Papoulis and S. U. Pillai, *Probability, Random Variable and Stochastic Processes*, fourth ed, McGraw-Hill, 2002.
- [157] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications*, first ed., Cambridge University Press, 2005.
- [158] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, second ed., John Wiley & Sons, 2006.
- [159] S. Lin and D. J. Costello Jr., *Error Control Coding*. second ed., Pearson Prentice Hall, 2004.
- [160] C. B. Schlegel and L. C. Perez, *Trellis and Turbo Coding*, first ed., John Wiley & Sons, 2004.