

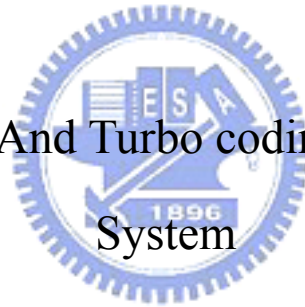
國立交通大學

電信工程學系碩士班

碩士論文

渦輪等化及渦輪碼在 MIMO-OFDM 系統上之實現

Turbo Equalization And Turbo coding for MIMO-OFDM
System



研究生：林育丞

指導教授：吳文榕 博士

中華民國九十六年七月

渦輪等化及渦輪碼在 MIMO-OFDM 系統上之實現

Turbo Equalization and Turbo coding for MIMO-OFDM
System

研究生：林育丞

Student：Yu-Cheng Lin

指導教授：吳文榕 博士

Advisor：Dr. Wen-Rong Wu

國立交通大學

電信工程學系碩士班



Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao-Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

In

Communication Engineering

July 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

渦輪等化及渦輪碼在 MIMO-OFDM 系統上之實現

研究生：林育丞

指導教授：吳文榕 教授

國立交通大學電信工程學系碩士班

摘要

在這篇論文中，我們著重在多輸入多輸出(MIMO)-正交分頻多工(OFDM)系統的外部接收機設計。外部接收機主要包括軟性位元產生器與錯誤更正碼解碼器，一般將這兩部分視為獨立，但依此無法達到最佳效能。在第一部分，我們研究渦輪等化(Turbo Equalization)在多輸入多輸出(MIMO)-正交分頻多工(OFDM)系統中的應用。此方法藉由遞迴的方式來結合軟性位元反對映與錯誤更正碼解碼。然而，位元交錯調變碼(BICM)依舊將錯誤更正碼以及調變視為獨立，並沒有利用到多輸入多輸出(MIMO)架構。而頻帶交錯調變碼(TICM)可用來解決這個問題。在第二部分，我們提出在渦輪碼(Turbo coding)在頻帶交錯調變碼(TICM)上的應用。在 802.11n 系統下，模擬證明渦輪等化以及渦輪碼結合頻帶交錯調變碼的效能均優於傳統的做法。

Turbo Equalization and Turbo coding for MIMO-OFDM System

Student: Yu-Cheng Lin

Advisor: Dr. Wen-Rong Wu

Department of Communication Engineering
National Chiao-Tung University

Abstract

In this thesis, we focus on the outer receiver design for multiple-input-multiple output (MIMO) OFDM systems. The outer receiver mainly consists of a soft-bit demapper and an outer code decoder. The conventional approach considers these two devices separately, and cannot achieve true optimum performance. In the first part of the thesis, we study the application of turbo equalization technique in MIMO-OFDM systems. Using this approach, we can combine demapping and decoding in an iterative way. The conventional communication system often uses the bit-interleaved-coding modulation (BICM) scheme. However, the BICM scheme treats coding and modulation separately, and does not exploit the MIMO structure constraint. Tone-interleaved-coding-modulation (TICM) has been proposed to solve the problem. However, only the convolution code was considered. In the second part of the thesis, we propose to use the turbo code in the TICM scheme. Simulations with the IEEE 802.11n systems show that both the turbo equalization and the TICM schemes can significantly outperform conventional approaches.

誌謝

本篇論文得以順利完成，首先要特別感謝我的指導教授 吳文榕博士，在課業學習與論文研究上不厭其煩的引導我正確的方向。

另外，我要感謝許兆元學長、李俊芳學長與楊華龍學長等在研究上不吝指導，且同時感謝寬頻傳輸與訊號處理實驗室所有同學與學弟妹們的幫忙，以及昭曄和健甫無私的提供資源上的協助。特別感謝沈士琦學長，在工作百忙之餘仍撥冗與我討論，並給我許多建議。最後感謝我家人以及佳盈，給予我在精神上最大的鼓勵與支持，使得我可以順利地完成碩士學位。



Contents

摘要	iii
Abstract.....	iv
誌謝	v
Chapter 1 Introduction.....	1
Chapter 2 Turbo Equalization in MIMO-OFDM Systems	4
2.1 Introduction to IEEE802.11n Proposal	4
2.2 MIMO Channel Model	5
2.3 System Model	7
2.4 Bit-Interleaved Coded Modulation.....	8
2.5 Suboptimal Receiver with the MMSE Equalizer.....	9
2.5.1 Minimum Mean-Square Error (MMSE) Equalizer.....	10
2.5.2 Simplified Soft-Bit Demapper for SISO-OFDM system.....	11
2.6 Optimal Receiver with LSD.....	14
2.6.1 Optimum MIMO Soft-Demapping	14
2.6.2 List Sphere Decoding (LSD).....	16
2.6.3 Real-Valued LSD	18
2.6.4 Radius of the Sphere	18
2.7 Turbo Equalization.....	19
2.7.1 BCJR Algorithm on Convolutional code decoder.....	21
2.7.2 List Sphere Decoding with Priori Information	25
2.7.3 MIMO Detection Using the List Sphere Decoder	26
2.8 Simulation Results	27

Chapter 3 Tone Interleaved Coded Modulation with Turbo Code.....	36
3.1 MIMO-OFDM with TICM-T	37
3.2 Transmitter for TICM-T.....	37
3.2.1 Turbo Encoder.....	38
3.2.2 Tone-Level Interleaver for TICM-T.....	40
3.3 Receiver for TICM-T.....	40
3.3.1 BM Calculation in TICM.....	41
3.3.2 Modified BCJR Algorithm for TICM-T	43
3.3.3 The Modified LSD detector with the Priori Information	46
3.4 Simulation Results	47
Chapter 4 Conclusions.....	51
Reference.....	52



List of Tables

Table 2-1: The primary parameters in the TGn Sync proposal	5
Table 2-2: Model parameters for LOS/NLOS conditions	6
Table 2-3: Path loss model parameters	7



List of Figures

Fig. 2-1: The block diagram of the transmitter in the TGn Sync proposal.....	4
Fig. 2-2: The block diagram of the MIMO-OFDM transmitter for BICM.....	10
Fig. 2-3: The block diagram of the MIMO-OFDM receiver for BICM.....	10
Fig. 2-4: The block diagram of the SISO-OFDM transmitter for BICM.....	13
Fig. 2-5: The block diagram of the SISO-OFDM receiver using the MMSE equalizer for BICM.....	13
Fig. 2-6: Partitions of the 16-QAM constellation.....	13
Fig. 2-7: The block diagram of the MIMO-OFDM receiver using the LSD detector for BICM.....	14
Fig. 2-8: LSD with the given radius of the sphere.....	16
Fig. 2-9: Diagram of turbo processing for the OFDM-MIMO system.....	20
Fig. 2-10: Recursive calculating of α in trellis diagram.....	23
Fig. 2-11: Recursive calculating of β in trellis diagram.....	24
Fig. 2-12: Performance comparison of soft+BCJR and turbo scheme (20MHz, 2x2, 16-QAM, channel-B, perfect-channel).....	29
Fig. 2-13: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4, 16-QAM, channel-B, perfect-channel).....	29
Fig. 2-14: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4, 64-QAM, channel-B, perfect-channel).....	30
Fig. 2-15: Performance comparison of soft+BCJR and turbo scheme (20MHz, 2x2, 16-QAM, channel-D, perfect-channel).....	30
Fig. 2-16: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4,	

16-QAM, channel-D, perfect-channel).....	31
Fig. 2-17: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4, 64-QAM, channel-D, perfect-channel).....	31
Fig. 2-18: Performance comparison of soft+BCJR and turbo scheme (20MHz, 2x2, 16-QAM, channel-B, estimated-channel).....	32
Fig. 2-19: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4, 16-QAM, channel-B, estimated-channel).....	33
Fig. 2-20: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4, 64-QAM, channel-B, estimated-channel).....	33
Fig. 2-21: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4, 16-QAM, channel-D, estimated-channel)	34
Fig. 2-22: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4, 16-QAM, channel-D, estimated-channel)	34
Fig. 2-23: Performance comparison of soft+BCJR and turbo scheme (20MHz, 4x4, 64-QAM, channel-D, estimated-channel)	35
Fig. 3-1: The block diagram of the MIMO-OFDM transmitter for TICM-T	38
Fig. 3-2: The turbo encoder for TICM-T	40
Fig. 3-3: The block diagram of the MIMO-OFDM receiver for TICM-T	41
Fig. 3-4: The iteration loop of the TICM-T	43
Fig. 3-5: The block diagram of the MIMO-OFDM transmitter for BICM-T	48
Fig. 3-6: The block diagram of the MIMO-OFDM receiver using the LSD for BCIM-T	48
Fig. 3-7: Performance comparison of BICM-T, and TICM-T (20MHZ, 4X4 16-QAM, channel B, estimated-channel).....	50

Fig. 3-8: Performance comparison of BICM-T, and TICM-T (20MHZ, 4X4 16-QAM,
channel B, estimated-channel)..... 50



Chapter 1 Introduction

Turbo equalization [1] [2] is a powerful mean to perform joint equalization and decoding, when considering coded data transmission over time dispersive channels. It can significantly improve the equalization result. The association of the code and the discrete-time equivalent channel (separated by an interleaver) is seen as the serial concatenation of two codes. The turbo principle can then be used at the receiver: performances are improved through an iterative exchange of extrinsic information between a soft-in/soft-out (SISO) equalizer and a SISO decoder. Classically, these SISO modules are implemented using conventional a posteriori probability (APP) algorithms based on [3].

Orthogonal frequency division multiplexing (OFDM), proposed by Salzberg in 1967 [6], is known to have high spectrum efficiency and good resistance against multi-path interference. Moreover, duplexing and multiple accesses can be easily implemented with a frequency division manner. One way to further enhance rates on a scattering-rich wireless channel is to use multiple transmit and receive antenna (MIMO) structure. However, the MIMO structure will induce the interference between different transmit antennas. A simple solution for the problem is to use a zero-forcing (ZF) or minimum-mean-squared-error (MMSE) equalizer to suppress the interference, and then convert the MIMO system into multiple SISO systems. Since the noise becomes correlated after the equalization, conventionally used SISO soft-demappers are not optimal. The optimum MIMO soft-demapper is known to require very high computational complexity, even higher than maximum likelihood (ML) estimator. An efficient algorithm named list sphere decoding (LSD) was proposed [5] to solve the problem. It has been shown that the LSD algorithm can significantly reduce the computational complexity, while having near optimum performance.

The bit interleaved coded modulation (BICM) scheme, originally proposed by Zehavi [6], inserts a bit-level interleaver between the encoder and the QAM mapper to reduce the bursty error problem, improving the error correction capability of the forward-error correction (FEC) coder. As a result, the BICM scheme is popularly applied in fast fading channel environments. In orthogonal frequency division multiplexing (OFDM) systems, subcarriers within same coherent bandwidth can experience deep fades even the channel is non-fading. The BICM scheme is then useful in this scenario.

In addition to the LSD algorithm, there is another way improving the performance of MIMO-OFDM systems when the BICM is used. The MIMO structure introduces interference between data transmitted in different antenna. Thus, the effect can be seen as the result of an additional coder. Combining with the original FEC coder, we can then apply the turbo equalization. Using the turbo principle, we can improve the detection performance by exchanging the information between the MIMO soft-demapper and the channel decoder. This problem has been considered in [5] and the result shows significant performance improvement.

To develop the next generation of wireless local area networks (WLAN), IEEE announced an 802.11n Task Group (TGn) in January 2004. The target data rate for the new standard has a theoretical bound of 540Mbit/s. In other words, it will be 100 times faster than 802.11b, or 10 times faster than 802.11a/11g. The modulation of 802.11n uses the MIMO-OFDM and BICM techniques such that the throughput can be significantly increased and the channel fading effect can be minimized. In the first part of this thesis, we will investigate turbo equalization in IEEE 802.11n system. We will use the channel models suggested by the standard to evaluate the performance of turbo equalization. The LSD method, mentioned above, will be used to obtain the MIMO soft-demapping results.

Tone interleaved and coded modulation (TICM), a new coding-modulation scheme proposed

in [11], uses the symbol block as a unit for interleaving, instead of bits. With this structure, the detection symbols in the received symbol vector (after the MIMO channel) can have a trellis relationship. As a result, the detection performance can be improved. However, since the interleaving is conducted in the symbol block level, the resistance to bursty error is reduced. Simulation results show that when the delay spread of the channel is short, TICM can have much better performance, and when the delay spread of the channel is long, TICM does not have significant advantages. Note that only convolutional coders are considered in [11]. In the second part of the thesis, we then propose to apply the turbo code in TICM. Since there is an interleaver between two component decoders, the bursty error problem can then be reduced. It is expected that TICM with a turbo code can outperform BICM with the same turbo code.

The rest of this thesis is organized as follows: In Chapter 2, we review IEEE802.11n systems, and the conventional demodulation method for the MIMO-OFDM system with BICM, including the MMSE equalizer, the soft-bit demapper, and the LSD detector. Also, we will describe how to apply turbo equalization in MIMO-OFDM system. In Chapter 3, we describe the proposed TICM scheme and its optimal demodulation method. Finally, we draw some conclusions and outline future works in Chapter 4.

Chapter 2 Turbo Equalization in MIMO-OFDM Systems

In MIMO-OFDM systems with BICM, we can treat the MIMO channel as an inner code and the FEC coder as an outer code, and use an “iterative decoding” technique [5] to approach the optimum performance of joint equalization and detection. This is commonly called turbo equalization. In this chapter, we will first review IEEE 802.11n systems, and outline the conventional MIMO-OFDM receiver. Then, we will discuss the LSD detector and MIMO soft-demapper. Finally, we will describe turbo equalization in MIMO-OFDM systems, and the corresponding application in IEEE802.11n systems.

2.1 Introduction to IEEE802.11n Proposal

The block diagram of the transmitter in the IEEE 802.11n is shown in Fig. 2-1.

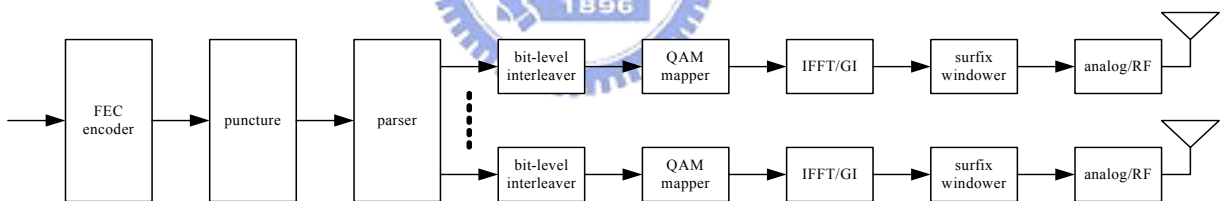


Fig. 2-1: The block diagram of the transmitter in the TGn Sync proposal

The mandatory number of spatial streams is two and the mandatory channelization bandwidth is 20MHz. The proposal also utilizes optional techniques such as four spatial streams, and 40MHz channelization bandwidth. Table 2-1 shows the primary parameters in the proposal.

Parameter	Mandatory	Optional
Number of Spatial Streams	1 and 2	3 and 4
Number of Transmit Antennas	2	Greater than 2
Channelization bandwidth	20MHz	40MHz
Number of Occupied Sub-channels	56 in 20MHz	114 in 40MHz
Number of Data Sub-channels	52	108
Number of Pilot Sub-channels	4	6
Modulation Type	BPSK, QPSK, 16-QAM, 64-QAM	256-QAM
Code Rate	1/2, 2/3, 3/4, 5/6	
Guard Interval	800ns	400ns
Convolutional Coding	$R_c = 1/2, K_{cc} = 7$	
LDPC		TX and RX Optional

Table 2-1: The primary parameters in the TGn Sync proposal

The mandatory channel code is the convolutional code (CC). The encoder is with $K_{cc} = 7$ and $R_c = 1/2$, where K_{cc} represents the constraint length of the CC encoder, and R_c represents the code rate. The encoded output is then punctured, according the required data rate, and then parsed to multiple spatial streams. To have the frequency diversity gain and reduce the spatial correlation of the MIMO channel, the binary data is interleaved by the size of OFDM symbol and then mapped into 2^M -quadrature amplitude modulation (QAM) symbols, which are used to modulate data carriers in subchannels. Each OFDM symbol allocates $N_{sd} = 52$ subchannels for data transmission.

2.2 MIMO Channel Model

There are six channel models, A, B, C, D, E and F, provided 802.11n [15]. The environments

for these channel models can be described as follows.

(Channel-A). a typical office environment, non-line-of-sight (NLOS) conditions, and 50 ns rms delay spread

(Channel-B). a typical large open space and office environments, NLOS conditions, and 100 ns rms delay spread

(Channel-C). a large open space (indoor and outdoor), NLOS conditions, and 150 ns rms delay spread

(Channel-D). the same as model C, line-of-sight (LOS) conditions, and 140 ns rms delay spread (10 dB Ricean K -factor at the first delay)

(Channel-E). a typical large open space (indoor and outdoor), NLOS conditions, and 250 ns rms delay spread

Properties of these models are shown in Table 2-2 and Table 2-3. Here, K -factor for the LOS condition applies only to the first tap, and K -factor = $-\infty$ (dB) applies to all other taps. There is another factor d_{BP} called the breakpoint distance. If the transmit-receive separation distance is set less than d_{BP} , the channel will be considered as the LOS condition. If the transmit-receive separation distance is greater than d_{BP} , the channel will be considered as the NLOS condition.

Model	Condition	K-factor (dB)	RMS delay spread (ns)	# of clusters
A (optional)	LOS/NLOS	0 / $-\infty$	0	1 tap
B	LOS/NLOS	0 / $-\infty$	15	2
C	LOS/NLOS	0 / $-\infty$	30	2
D	LOS/NLOS	3 / $-\infty$	50	3
E	LOS/NLOS	6 / $-\infty$	100	4
F	LOS/NLOS	6 / $-\infty$	150	6

Table 2-2: Model parameters for LOS/NLOS conditions

New Model	d_{BP} (m)	Slope before d_{BP}	Slope after d_{BP}	Shadow fading std. dev. (dB) before d_{BP}	Shadow fading std. dev. (dB) after d_{BP}

				(LOS)	(NLOS)
A (optional)	5	2	3.5	3	4
B	5	2	3.5	3	4
C	5	2	3.5	3	5
D	10	2	3.5	3	5
E	20	2	3.5	3	6
F	30	2	3.5	3	6

Table 2-3: Path loss model parameters

We choose the channel-B (NLOS) with distance 6m, channel-D (NLOS) with distance 11m, and channel-E (NLOS) with distance 21m as our simulation environments in this thesis. Besides, the time resolution of the channel model is 10ns, which is one-fifth of the sampling period (3.2ms/64=50ns). We oversample the transmitted signal by a factor of 5 and interpolate it linearly in order to convolve with the response generated by the channel model.

2.3 System Model

Before our formal development, we define notations will be used in the sequel. Scalars are denoted in lower case letters, vectors in lower case italic letters, and metrics in upper case bold letters. Also, $[\cdot]^T$ and $[\cdot]^H$ indicate the transpose and conjugate transpose of a vector or matrix inside the bracket respectively. Since 802.11n is designed to be used in the indoor environment, we can assume that the MIMO channel is a multi-path quasi-static Rayleigh fading channel. The frequency response of the MIMO channel at the k^{th} sub-channel is defined as

$$\mathbf{H}^k = \begin{bmatrix} h_{1,1}^k & \cdots & h_{1,N_T}^k \\ \vdots & h_{q,p}^k & \vdots \\ h_{N_R,1}^k & \cdots & h_{N_R,N_T}^k \end{bmatrix} \quad (2.1)$$

where p represents the index of the transmit antenna, q the index of the receive antenna, N_T the number of transmit antennas, and N_R the number of receive antennas. The transmitted symbol vector at the k^{th} sub-channel and the l^{th} OFDM symbol before IFFT/GI is defined as

$$\mathbf{s}^{l,k} = [s_1^{l,k}, \dots, s_{N_T}^{l,k}]^T \quad (2.2)$$

The received symbol vector at the k^{th} sub-channel and the l^{th} OFDM symbol after GI-Remover/FFT is defined as

$$\mathbf{r}^{l,k} = [r_1^{l,k}, \dots, r_{N_R}^{l,k}]^T \quad (2.3)$$

Assume that there are no inter symbol interference (ISI) and inter carrier interference (ICI). Then, $\mathbf{r}^{l,k}$ can be represented as

$$\mathbf{r}^{l,k} = \mathbf{H}^k \cdot \mathbf{s}^{l,k} + \mathbf{n}^{l,k} \quad (2.4)$$

where

$$\mathbf{n}^{l,k} = [n_1^{l,k}, \dots, n_{N_R}^{l,k}]^T \quad (2.5)$$

is the received noise vector, and each element in $\mathbf{n}_{l,k}$ is statistically independent and identically distributed (i.i.d.) complex Gaussian random variable with zero mean and variance $\sigma^2 = \sigma_I^2 + \sigma_Q^2 = N_0$. Signal to noise ration (SNR) is defined as the average received power per receive antenna divided by the average noise power.

$$SNR = \frac{E\{|r'_q(t)|^2\}}{\sigma^2} \quad (2.6)$$

where $r'_q(t)$ represents the received signal at time t at the q^{th} transmit antenna. For simplicity, we ignore the index k and l in the rest of this thesis. Then, we have

$$\mathbf{r} = \mathbf{H} \cdot \mathbf{s} + \mathbf{n} \quad (2.7)$$

2.4 Bit-Interleaved Coded Modulation

First of all, we would review two existing demodulation schemes on BICM structure. The first one “decouples” the MIMO symbol vector to single-input-single-output (SISO) scalar

signals by a ZF/MMSE equalizer. And then it calculates the soft coded bits information with a SISO soft-demapper, and feed the result to the FEC decoder. The scheme is simple and the required computational complexity is low, but it is not optimal. In many cases, its performance becomes not acceptable. The other one calculates the soft “coded bits” information using a MIMO soft-demapper directly. This is an optimum approach; however, its computational complexity can be very large and it is not feasible in real-world systems in general. In [10], a low-complexity algorithm overcoming the problem, named LSD, was proposed. Note that to apply the “iterative decoding” technique, we have to use the MIMO soft-demapper. This is because soft-bit information calculated with the equalization-and-SISO-soft-demapping approach is not accurate enough. As a result, the error will be propagated during iteration, yielding poor performance. The iterative decoding (Turbo equalization) proposed by [5] exchanging the coded bits extrinsic information between the equalizer and the channel decoder. As will be shown later, turbo equalization can greatly enhance the performance of MIMO-OFDM systems.

2.5 Suboptimal Receiver with the MMSE Equalizer

Here, let the FEC coder be the CC coder. Since the performance of the MMSE equalizer is better than the ZF equalizer, we only consider the MMSE equalizer. This receiver scheme includes a MMSE equalizer, a SISO soft-demapper, and an optimum decoder, called the BCJR decoder [7]. First, the received symbol vector is processed by the MMSE equalizer, which decouples the MIMO symbol vector to multiple SISO signals. For each SISO signal, we apply a SISO soft-demapper to obtain soft coded bit information. Finally, the whole soft-bit stream is fed into the BCJR decoder to detect information bits. The block diagram of the MIMO-OFDM transceiver is shown in Fig. 2-2 and Fig. 2-3.

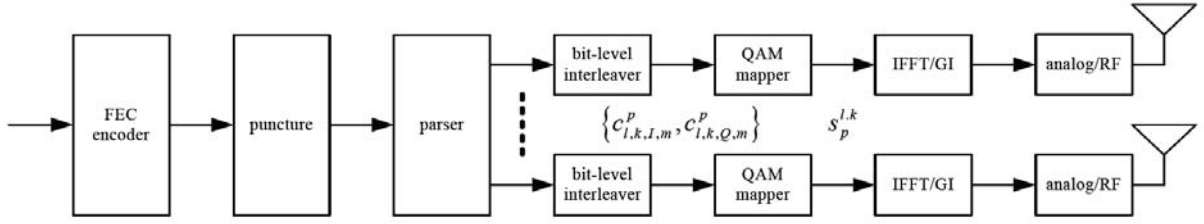


Fig. 2-2: The block diagram of the MIMO-OFDM transmitter for BICM

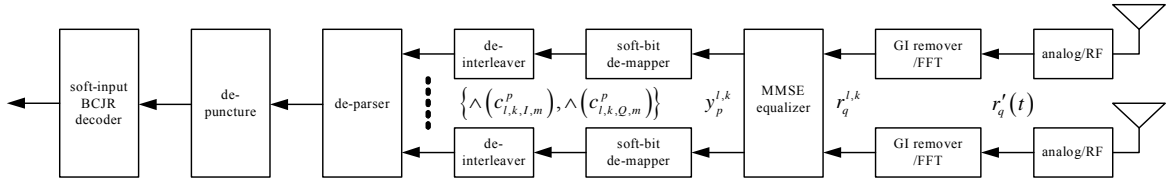


Fig. 2-3: The block diagram of the MIMO-OFDM receiver for BICM

2.5.1 Minimum Mean-Square Error (MMSE) Equalizer

Assume that the received symbol vector : $\mathbf{r} = \mathbf{H} \cdot \mathbf{s} + \mathbf{n}$. We define an error vector e_k , which is the error between the received signal and the MMSE equalizer output \mathbf{y} . That is

$$G_{MMSE} = \arg \min_{G_{MMSE}} \{|\mathbf{y} - \mathbf{s}|^2\} = \arg \min_{G_{MMSE}} \{G_{MMSE}^H \mathbf{r} - \mathbf{s}\}^2 \quad (2.8)$$

With the Wiener-Hopf equation, we can have

$$G_{MMSE} = [\mathbf{H}\mathbf{H}^H + \alpha I_{N_r * N_r}]^{-1} \mathbf{H} \quad (2.9)$$

where $\alpha = \frac{\sigma_{noise}^2}{\sigma_{per-antenna}^2}$, σ_{noise}^2 is the additive complex Gaussian noise variance, $\sigma_{per-antenna}^2$ is

the average transmission power over per antenna. Finally, the MMSE equalizer output can be obtained as

$$\mathbf{y}_k = [y_1 \cdots y_{N_r}]^T = G_{MMSE}^H \cdot \mathbf{r}_k \quad (2.10)$$

2.5.2 Simplified Soft-Bit Demapper for SISO-OFDM system

After equalization, the MIMO signal is decoupled into SISO signals. We can then use SISO soft-demappers [8] calculate coded bits information. Let M be the size of constellation in Gray-mapping, where $N_{bpsc} = \log_2(M)$ is the number of coded bits in one QAM symbol. That is, $N_{bpsc}/2$ coded bits are mapping to the in-phase or quadrature-phase part of QAM symbol. Soft bit information is generally represented by LLR, defined as

$$LLR(c_{I,i}) \triangleq \log\left(\frac{p(c_{I,i} = 1|r)}{p(c_{I,i} = 0|r)}\right) = \log\left(\frac{\sum_{s_j \in \mathcal{S}_{I,i}^{(1)}} p(s_j|r)}{\sum_{s_j \in \mathcal{S}_{I,i}^{(0)}} p(s_j|r)}\right) \quad (2.11)$$

where the $\mathcal{S}_{I,i}^{(1)}$ comprises the set of symbols on certain transmitter antenna with a '0' in position (I, i) .

The LLR can be simplified by using the log-sum approximation as:

$$\log\left(\sum_j p_j\right) \approx \max\{\log(p_j)\} \quad (2.12)$$

The approximation error will be small if the terms in the left-hand side of (2.13) is dominated by the largest one. This is true when SNR is high. Then, we have

$$\log\left(\frac{\max_{s_j \in \mathcal{S}_{I,i}^{(1)}}\{p(s_j|r)\}}{\max_{s_j \in \mathcal{S}_{I,i}^{(0)}}\{p(s_j|r)\}}\right) = \log \max_{s_j \in \mathcal{S}_{I,i}^{(1)}}\{p(s_j|r)\} - \log \max_{s_j \in \mathcal{S}_{I,i}^{(0)}}\{p(s_j|r)\} \quad (2.14)$$

The SISO system can be modeled as $r = h \cdot s + n$ (we omit the index p and the index q for simplicity). Since $N_T = 1$ and $N_R = 1$, and $p(s_j|r) = \frac{1}{\pi\sigma^2} \exp\left\{-\frac{1}{2} \frac{|r - h \cdot s_j|^2}{\sigma^2}\right\}$, we can rewrite (2.14) as

$$\begin{aligned}
LLR(c_{I,i}) &= \frac{1}{\sigma^2} \left\{ -\min_{s_j \in \mathcal{S}_{I,i}^{(1)}} \{|r - h \cdot s_j|^2\} + \min_{s_j \in \mathcal{S}_{I,i}^{(0)}} \{|r - h \cdot s_j|^2\} \right\} \\
&= \frac{|h|^2}{\sigma^2} \left\{ \min_{s_j \in \mathcal{S}_{I,i}^{(0)}} \{|y - s_j|^2\} - \min_{s_j \in \mathcal{S}_{I,i}^{(1)}} \{|y - s_j|^2\} \right\} \\
&= \frac{|h|^2}{\sigma^2} D_{I,i}
\end{aligned} \tag{2.15}$$

where $\mathbf{y}_k = [y_1 \cdots y_{N_T}]^T$ is the equalizer output. From (2.15), we can see that only the distance between the two nearest QAM symbol on set 0 or 1 is required to calculate the LLR. Note that for the Gray mapping scheme, some simple rules exist such that the distance can be found with a low computational complexity. It turns out that we only have to partition the constellation plane, horizontally or vertically (corresponding to in-phase or quadrature-phase part), and find the nearest QAM mapping points of the two set, which are always laid on a same vertical or horizontal or straight line. The observation holds for 16 and 64-QAM constellation. For 16 QAM constellations, we can derive the distance in the in-phase part as [8]

$$\begin{aligned}
D_{I,1} &= \begin{cases} |y_I|, & |y_I| \leq 2 \\ 2(y_I - 1), & y_I > 2 \\ 2(y_I + 1), & y_I < -2 \end{cases} \\
D_{I,2} &= -|y_I| + 2
\end{aligned} \tag{2.16}$$

It can be easily verified that $D_{Q,i}$ for the two quadrature bits are the same as that calculated in (2.16) with $y_{I,i}$ replaced by $y_{Q,i}$.

The block diagram of the SISO-OFDM transceiver (for BICM) is shown in Fig. 2-4 and Fig.

2-5

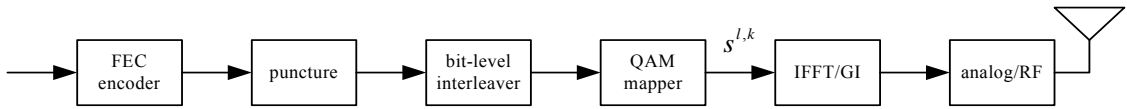


Fig. 2-4: The block diagram of the SISO-OFDM transmitter for BICM

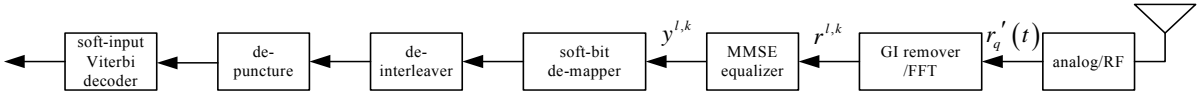


Fig. 2-5: The block diagram of the SISO-OFDM receiver using the MMSE equalizer for BICM

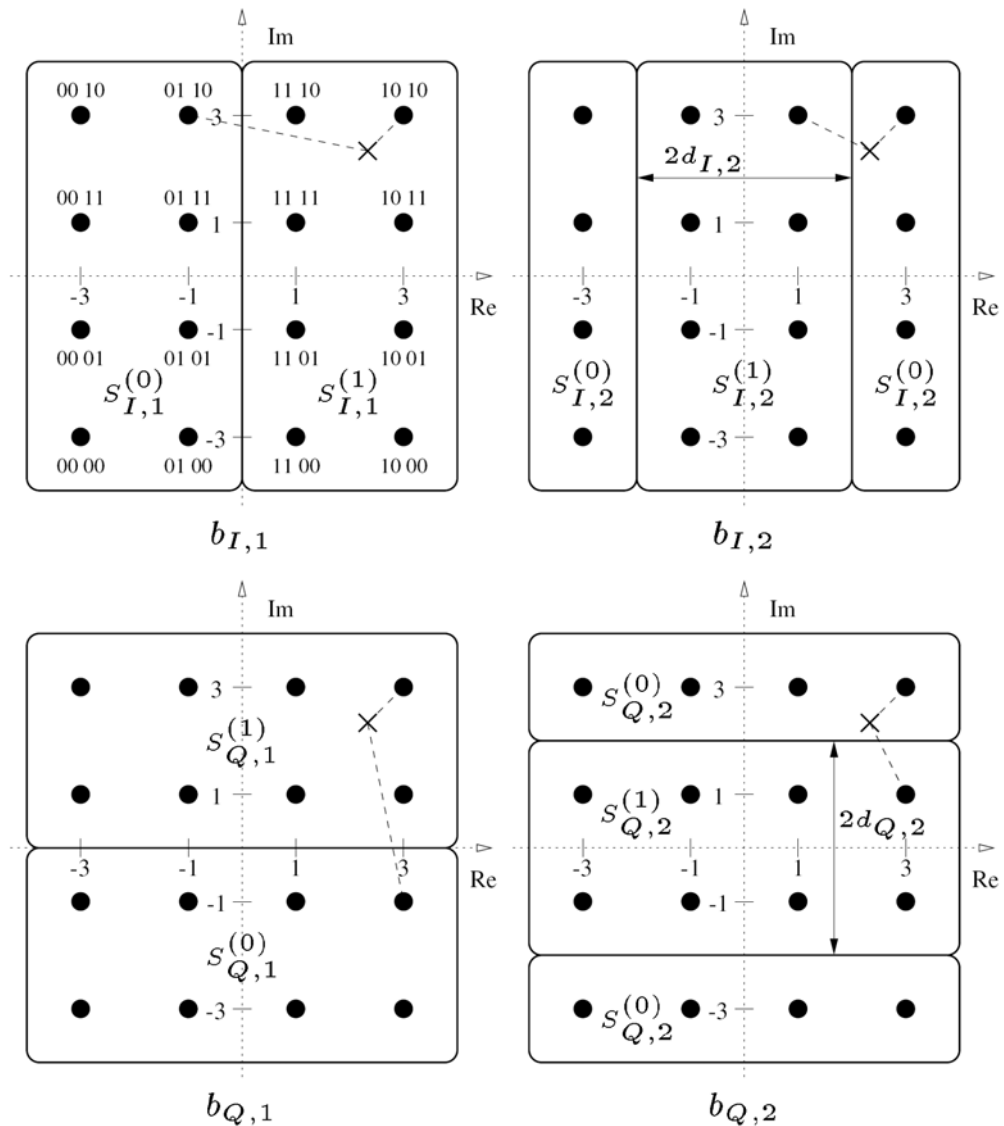


Fig. 2-6: Partitions of the 16-QAM constellation

2.6 Optimal Receiver with LSD

In the previous section, we can see that the MIMO symbol vector can be decoupled by a MMSE equalizer, and coded bit information can be obtained by SISO soft demappers. The receiver structure is simple and the required computational complexity is low. However, the noise after equalization becomes colored. Also, signals from different antennas may become correlated. The optimality of the BCJR decoder is based on the assumption that the noise is white and the signals are independent to each other. Thus, the conventional approach is not optimal. To solve this problem, we use a MIMO soft-demapper, instead of an equalizer and SISO soft-demappers. It has been known that the computational complexity of the optimum MIMO soft-demapping can be very large. Thus, we use the LSD algorithm to solve the problem. The block diagram of the MIMO-OFDM receiver with the LSD demapping is shown in Fig. 2-7

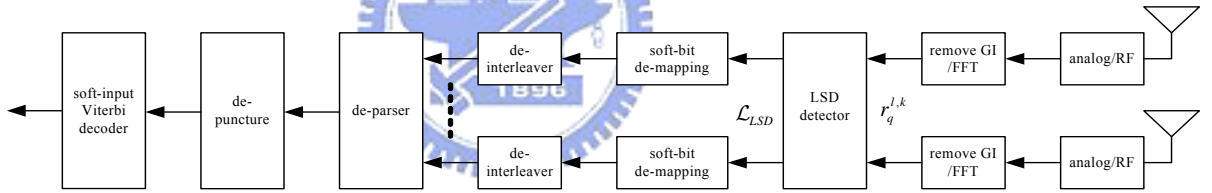


Fig. 2-7: The block diagram of the MIMO-OFDM receiver using the LSD detector for BICM

2.6.1 Optimum MIMO Soft-Demapping

The received symbol vector of a MIMO-OFDM system can be modeled as:

$$\mathbf{r} = \mathbf{H} \cdot \mathbf{s} + \mathbf{n}, \quad \mathbf{r} = [r_1 \cdots r_{N_T}]^T \quad (2.17)$$

If we use 2^M -QAM transmission, and the number of transmit antenna is N_T , there will be a total of $M \cdot N_T$ bits in the received symbol vector \mathbf{r} . With the MIMO soft-demapping, we calculate the coded bit LLR directly without equalization. Now, we find the nearest symbol vectors belonging to two sets (0 or 1) based on the received symbol vector, instead of an

individual scalar signal. That is:

$$LLR(c_{k,i}) \triangleq \log\left(\frac{\Pr[c_{k,i} = 1|\mathbf{r}]}{\Pr[c_{k,i} = 0|\mathbf{r}]}\right) = \log\left(\frac{\sum_{\varphi_j \in \mathbf{s}_{k,i}^{(1)}} p(\varphi_j|\mathbf{r})}{\sum_{\varphi_j \in \mathbf{s}_{k,i}^{(0)}} p(\varphi_j|\mathbf{r})}\right) \quad (2.18)$$

where $c_{k,i}$ is the i -th bit on k -th transmitter antenna, $\mathbf{s}_{k,i}^{(l)}$ is the set of symbol vectors with ‘0’ on position (k, i) . $\varphi_j = [\varphi_1, \dots, \varphi_{N_T}]$ is the possible symbol vector belonging to $\mathbf{s}_{k,i}^{(l)}$.

According to Baye’s rule, we can derive the probability density function of $p(\varphi_j|\mathbf{r})$ as:

$$p(\varphi_j|\mathbf{r}) = \frac{p(\varphi_j, \mathbf{r})}{p(\mathbf{r})} = \frac{p(\mathbf{r}|\varphi_j) \cdot p(\varphi_j)}{p(\mathbf{r})} \quad (2.19)$$

Assuming that the appearing probability of each symbol $p(\varphi_j)$, is equally probable. So, $p(\varphi_j)$ and $p(\mathbf{r})$ can be canceled in the LLR calculation. We then have

$$p(\mathbf{r}|\varphi_j) = \frac{1}{(\pi\sigma^2)^{N_r}} \exp\left\{-\frac{1}{2\sigma^2} \|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\right\} \quad (2.20)$$

Equation (2.18) can now be rewritten as:

$$\begin{aligned} LLR(c_{k,i}) &= \log\left(\frac{\sum_{\varphi_j \in \mathbf{s}_{k,i}^{(1)}} p(\varphi_j|\mathbf{r})}{\sum_{\varphi_j \in \mathbf{s}_{k,i}^{(0)}} p(\varphi_j|\mathbf{r})}\right) \\ &\approx \log \max_{\varphi_j \in \mathbf{s}_{k,i}^{(1)}} (p(\mathbf{r}|\varphi_j)) - \log \max_{\varphi_j \in \mathbf{s}_{k,i}^{(0)}} (p(\mathbf{r}|\varphi_j)) \\ &\approx \frac{1}{2\sigma^2} \left\{ \min_{\varphi_j \in \mathbf{s}_{k,i}^{(0)}} \{\|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\} - \min_{\varphi_j \in \mathbf{s}_{k,i}^{(1)}} \{\|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\} \right\} \end{aligned} \quad (2.21)$$

Though the max-log theorem simplifies the calculation, the required computational complexity is still very large when the number of transmit antennas and the QAM size become large. For example, when the number of transmitter antenna is 4 ($N_T=4$) and the modulation is 64-QAM ($M=8$), there are $2^{M \cdot N_T - 1} \approx 8.4 \times 10^6$ symbol vectors in the ‘0’ and ‘1’ sets, required to

be tested. In order to reduce the complexity, we use the LSD algorithm.

2.6.2 List Sphere Decoding (LSD)

A simply way to approximate (2.21) is to consider only the symbol vectors whose distance $\|\mathbf{r} - \mathbf{H} \cdot \boldsymbol{\varphi}\|^2$ is small. As a result, we can have a “list” of candidate symbol vectors, and find the minimum in the list instead of the exhausted search. Searching the list generally provides a good approximation of (2.21), if the distance is carefully chosen. Observe that:

$$\|\mathbf{r} - \mathbf{H} \cdot \boldsymbol{\varphi}\|^2 = (\boldsymbol{\varphi} - \mathbf{y})^H \cdot \mathbf{H}^H \cdot \mathbf{H} \cdot (\boldsymbol{\varphi} - \mathbf{y}) + \mathbf{r}^H \cdot (\mathbf{I} - \mathbf{H} \cdot (\mathbf{H}^H \cdot \mathbf{H})^{-1} \cdot \mathbf{H}^H) \cdot \mathbf{r} \quad (2.22)$$

where \mathbf{y} is the equalizer output in (2.10). The LSD algorithm find those symbol vectors with distance small than r_{LSD} , where is:

$$\begin{aligned} \mathcal{L}_{LSD} &= \arg \min_{\boldsymbol{\varphi} \in \Psi} \|\mathbf{r} - \mathbf{H} \cdot \boldsymbol{\varphi}\|^2 \leq r_{LSD}^2 \\ &\approx \arg \min_{\boldsymbol{\varphi} \in \Psi} (\boldsymbol{\varphi} - \mathbf{y})^H \cdot \mathbf{H}^H \cdot \mathbf{H} \cdot (\boldsymbol{\varphi} - \mathbf{y}) \leq r_{LSD}^2 \end{aligned} \quad (2.23)$$

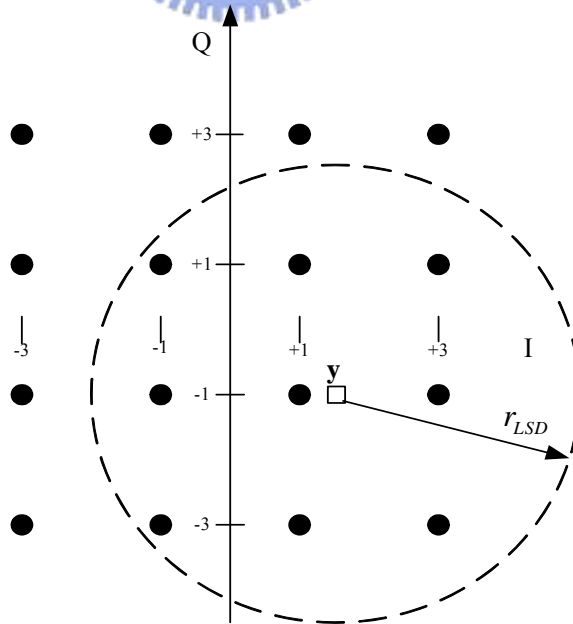


Fig. 2-8: LSD with the given radius of the sphere

where $\Psi = [\psi_1 \cdots \psi_{N_T}]$ represents the subset of all possible symbol vectors from the constellation map. The LSD detector only checks those symbol vectors lying inside a sphere as shown in Fig. 2-8.

We now show how to implement the idea. Applying Cholesky factorization, we factorize \mathbf{H} as upper triangular \mathbf{U}

$$\mathbf{U}^H \cdot \mathbf{U} = \mathbf{H}^H \cdot \mathbf{H} \quad (2.24)$$

where

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & \cdots & u_{1,N_T} \\ & u_{i,i} & \vdots \\ 0 & & u_{N_T,N_T} \end{bmatrix} \quad (2.25)$$

It can be proved that \mathbf{U} is positive and real. Then, (2.23) can be modified as

$$\begin{aligned} \arg_{\varphi \in \Psi} (\varphi - \mathbf{y})^H \cdot \mathbf{U}^H \cdot \mathbf{U} \cdot (\varphi - \mathbf{y}) &\leq r_{LSD}^2 \\ \Rightarrow \sum_{i=1}^{N_T} \left(u_{i,i}^2 |\varphi_i - y_i|^2 + \sum_{j=i+1}^{N_T} u_{i,j}^2 |\varphi_j - y_j|^2 \right) &\leq r_{LSD}^2 \end{aligned} \quad (2.26)$$

With the transformation, we can search the candidates in a tree structure. We start from $i = N_T$ and accumulate the distance, layer by layer. If the distance of one branch is smaller than r_{LSD}^2 , we continue to calculate the branch distance extending from this branch. If the distance is larger than r_{LSD}^2 , we prune this branch.

After obtaining the list \mathcal{L}_{LSD} , we can find the one with the minimum distance in the sets with the indicated bit is '0' and '1'. We can rewrite (2.21) to:

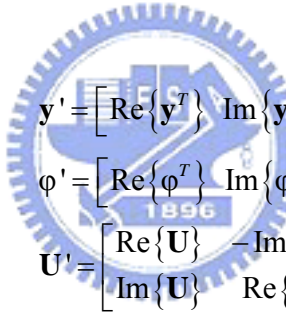
$$\begin{aligned} LLR(c_{k,i}) &\approx \frac{1}{2\sigma^2} \left\{ \min_{\varphi_j \in \mathcal{S}_{k,i}^{(0)}} \{\|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\} - \min_{\varphi_j \in \mathcal{S}_{k,i}^{(1)}} \{\|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\} \right\} \\ &\approx \frac{1}{2\sigma^2} \left\{ \min_{\varphi_j \in \mathcal{S}_{k,i}^{(0)} \cap \mathcal{L}_{LSD}} \{\|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\} - \min_{\varphi_j \in \mathcal{S}_{k,i}^{(1)} \cap \mathcal{L}_{LSD}} \{\|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\} \right\} \end{aligned} \quad (2.27)$$

2.6.3 Real-Valued LSD

In the QAM transmission, the real and the imaginary parts of the received symbol vector \mathbf{r} can be considered to be independent. The channel matrix \mathbf{H} and the symbol vector \mathbf{s} can also be decomposed into real and imaginary parts. That is, we can transform the original complex-valued LSD to the real-valued LSD with $2 \cdot N_T$ dimensions. Then, (2.26) can be revised to

$$\begin{aligned} \arg_{\varphi \in \Psi} (\varphi - \mathbf{y})^H \cdot \mathbf{U}^H \cdot \mathbf{U} \cdot (\varphi - \mathbf{y}) &\leq r_{LSD}^2 \\ \Rightarrow \arg_{\varphi \in \Psi} (\varphi' - \mathbf{y}')^H \cdot \mathbf{U}'^H \cdot \mathbf{U}' \cdot (\varphi' - \mathbf{y}') &\leq r_{LSD}^2 \end{aligned} \quad (2.28)$$

where



$$\begin{aligned} \mathbf{y}' &= [\text{Re}\{\mathbf{y}^T\} \quad \text{Im}\{\mathbf{y}^T\}]^T \\ \varphi' &= [\text{Re}\{\varphi^T\} \quad \text{Im}\{\varphi^T\}]^T \\ \mathbf{U}' &= \begin{bmatrix} \text{Re}\{\mathbf{U}\} & -\text{Im}\{\mathbf{U}\} \\ \text{Im}\{\mathbf{U}\} & \text{Re}\{\mathbf{U}\} \end{bmatrix} \end{aligned} \quad (2.29)$$

With this approach, the computational complexity of the LSD can be further reduced since more symbol vectors will be pruned in early stages. However, the latency of the real-valued LSD becomes larger due to the higher dimensions.

2.6.4 Radius of the Sphere

The size of the list \mathcal{L}_{LSD} determines how well (2.27) can approximate (2.21). If the radius is too small, no symbol vectors will be found. If the radius is too large, the number of symbol vectors in the list will be too large, and the complexity will be high (though it can approach the optimum solution better). How to determine the radius becomes a critical problem. In [5], a

simple method for the determination of the radius is proposed. Assuming that the true transmitter symbol vector is \mathbf{s}_{true} , we then have

$$|\mathbf{r} - \mathbf{H} \cdot \mathbf{s}_{true}|^2 = |\mathbf{n}|^2 \sim \sigma^2 \cdot \chi_{N_T}^2 \quad (2.30)$$

where $\chi_{N_T}^2$ is a random variable having a chi-square distribution with N_T degrees of freedom. However, note that the vector used for distance calculation is transformed by the matrix \mathbf{U} . The distribution of the squared-distance will be highly depended on \mathbf{H} , In [16], the radius is given by

$$r_{LSD}^2 = C_{LSD} \cdot \det(\mathbf{H})^{1/N_R} \quad (2.31)$$

where C_{LSD} is a real constant greater than 1. Once a suitable C_{LSD} is chosen, we can determine the radius of LSD. We will use the method in our later development. However, even for a good C_{LSD} , there will be no guarantee that at least one candidate will be in the sphere. If no candidates are found, the radius needs to be enlarged and repeat the search until $\mathcal{L}_{LSD} \neq \emptyset$. In case of that one of the two sets ($\mathbf{s}_{k,i}^{(1)}$ and $\mathbf{s}_{k,i}^{(0)}$) is empty, we can simply assign the LLR as an extreme value representing that the probability of the bit being 1 or 0 is high.

2.7 Turbo Equalization

With the optimum MIMO soft-demapping, we have greatly enhanced the performance of the MIMO-OFDM receiver. However, demapping and decoding are still conducted separately. In other words, the true optimum performance for the receiver is not achieved. To solve the problem, demapping and decoding should be considered jointly. However, this will results in a prohibited computational complexity. Turbo equalization, an iterative detection method, can effectively solve the problem. We can treat the MIMO channel as an inner encoder, and the CC as an outer encoder. Since an interleaver has been equipped between these two encoders, the combination of

these devices resembles a turbo code structure. The iterative process can be described as follows. The outer decoder processes the soft information from the demapper, and generates its own soft information indicating the relative likelihood of each of the coded bits. This soft information from the outer decoder is then fed back to the demapper, creating a feedback loop between the demapper and decoder. The operation is repeated until a stopping criterion is met. This process is often termed “belief propagation” or “message passing” and has a number of important connections to methods in artificial intelligence, statistical inference, and graphical learning theory. Fig. 2-9 gives a flowchart of turbo algorithm that we use.

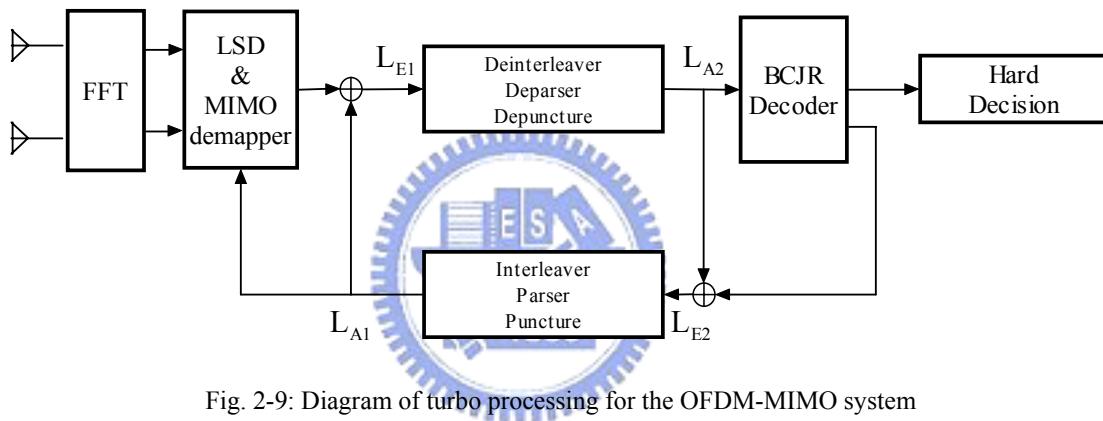


Fig. 2-9: Diagram of turbo processing for the OFDM-MIMO system

The LSD and the MIMO demapper takes observations and a priori coded bits information L_{A_1} to compute new (also referred to as “extrinsic”) coded bits information L_{E_1} for each of the received symbol vector. Then L_{E_1} is de-interleaved and rearranged to become the a priori input L_{A_2} to the outer soft-in/soft-out maximum a posteriori probability (MAP) decoder, which calculates extrinsic coded bits information L_{E_2} and the information bits LLR. Then L_{E_2} is re-interleaved and feedback as a priori information to the MIMO soft-demapper. This completes one iteration. Each iteration can reduce the bit-error rate by exchanging information between the inner and the outer decoder.

2.7.1 BCJR Algorithm on Convolutional code decoder

In order to minimize the probability of error for each detected symbol and calculate the soft coded bits information, we use the MAP detector instead the Viterbi decoder [9]. The algorithm for the MAP detection is termed the BCJR algorithm [7] in the literature. The original BCJR algorithm was expressed in the probabilistic form. For example, to make a decision about the information bit of k -th stage b_k , this decoder calculates the a posteriori probability $\Pr[b_k|\mathbf{r}]$ for each possible $b_k \in \{0,1\}$, and decides on the b_k that maximizes $\Pr[b_k|\mathbf{r}]$. In the convolutional code trellis diagram, these probabilities are easily computed once the *a posteriori state transition probability* $\Pr[\Psi_k = p; \Psi_{k+1} = q|\mathbf{c}]$ is known for each state transition in the trellis. The BCJR (Bahl, Cocke, Jelinek and Raviv) algorithm [7] provides a computationally efficient method for finding these state transition probabilities.

The key to the BCJR algorithm is to decompose *a posteriori state transition probability* into three separable parts, and use a recursive technique to calculate the transition probability. We start from the a posteriori probability in convolutional trellis structure.

$$\Pr[b_k|\mathbf{c}] = \sum_{(p,q) \in S_a} \Pr[\Psi_k = p; \Psi_{k+1} = q|\mathbf{c}] \quad (2.32)$$

where $p, q \in \{1 \dots Q\}$ is the state of convolutional code, the \mathbf{c} is the coded bits probability. We can calculate the a posteriori transition probability of certain information bit by summarizing a set of transition probabilities S_a whose certain information bit is 0 or 1.

We now can further decompose the transition probabilities of such stage into three parts: the first $\mathbf{c}_{l < k} = \{c_l : l < k\}$ only depends on the “past” observations, the second one only depends on the “present” observations c_k , and the third one depends on the “future” observations

$\mathbf{c}_{l>k} = \{c_l : l > k\}$. We can accomplish this through the following series of equalities.

$$\begin{aligned}
\Pr[\Psi_k = p; \Psi_{k+1} = q | \mathbf{c}] &= p(\Psi_k = p; \Psi_{k+1} = q; \mathbf{c}) / p(\mathbf{c}) \\
&= p(\Psi_k = p; \Psi_{k+1} = q; \mathbf{c}_{l<k}; \mathbf{c}_k; \mathbf{c}_{l>k}) / p(\mathbf{c}) \\
&= p(\mathbf{c}_{l>k} | \Psi_k = p; \Psi_{k+1} = q; \mathbf{c}_{l<k}; \mathbf{c}_k) \cdot p(\Psi_k = p; \Psi_{k+1} = q; \mathbf{c}_{l<k}; \mathbf{c}_k) / p(\mathbf{c})
\end{aligned}
\tag{2.33}$$

Due to the Markov property of the finite-state machine model for the channel, knowledge of the state at time $k+1$ supersedes the knowledge of the state at time k , and it also supersedes knowledge of $\mathbf{c}_{l<k}$ and \mathbf{c}_k , so (2.33) can be reduced to

$$\begin{aligned}
\Pr[\Psi_k = p; \Psi_{k+1} = q | \mathbf{c}] &= p(\mathbf{c}_{l>k} | \Psi_{k+1} = q) \cdot p(\Psi_k = p; \Psi_{k+1} = q; \mathbf{c}_{l<k}; \mathbf{c}_k) / p(\mathbf{c}) \\
&= p(\mathbf{c}_{l>k} | \Psi_{k+1} = q) \cdot p(\Psi_{k+1} = q; \mathbf{c}_k | \Psi_k = p; \mathbf{c}_{l<k}) \cdot p(\Psi_k = p; \mathbf{c}_{l<k}) / p(\mathbf{c})
\end{aligned}
\tag{2.34}$$

Again, exploiting the Markov property, we can simplify (2.34) to:

$$\Pr[\Psi_k = p; \Psi_{k+1} = q | \mathbf{c}] = \underbrace{p(\Psi_k = p; \mathbf{c}_{l<k})}_{\alpha_k(p)} \cdot \underbrace{p(\Psi_{k+1} = q; \mathbf{c}_k | \Psi_k = p)}_{\gamma_k(p,q)} \cdot \underbrace{p(\mathbf{c}_{l>k} | \Psi_{k+1} = q)}_{\beta_{k+1}(q)} / p(\mathbf{c})
\tag{2.35}$$

Observe that $\alpha_k(p)$ is the probability measure for state p at stage k that depends only on the past observations $\mathbf{c}_{l<k}$. $\beta_{k+1}(q)$, on the other hand, is a probability measure for state q at stage $k+1$ that depends only on the future observations $\mathbf{c}_{l>k}$. And $\gamma_k(p, q)$ is a probability measure connecting state p at stage k and the state q at stage $k+1$, and also, it depends only on the present observation \mathbf{c}_k .

Now we can derive (2.32) by using (2.35)

$$\begin{aligned}
\Pr[b_k | \mathbf{c}] &= \sum_{(p,q) \in S_a} \Pr[\Psi_k = p; \Psi_{k+1} = q | \mathbf{c}] \\
&= \frac{1}{p(\mathbf{c})} \sum_{(p,q) \in S_a} \alpha_k(p) \cdot \gamma_k(p, q) \cdot \beta_{k+1}(q)
\end{aligned}
\tag{2.36}$$

How to calculate the three parts of transition probability measure recursively is stated in the

following. We start from the “present” branch metric $\gamma_k(p, q)$. According to (2.35)

$$\begin{aligned}\gamma_k(p, q) &= p(\Psi_{k+1} = q; \mathbf{r}_k | \Psi_k = p) \\ &= p(\mathbf{c}_k | \Psi_k = p; \Psi_{k+1} = q) \cdot \Pr[\Psi_{k+1} = q | \Psi_k = p]\end{aligned}\quad (2.37)$$

where the first part is the product of certain coded bits probability, according to the transition codeword.

$$p(\mathbf{c}_k | \Psi_k = p; \Psi_{k+1} = q) = \prod_{j=1}^n p(c_j) \quad (2.38)$$

And the second part of the equality is the priori information of transition codeword. Generally we assume all input sequences with equal probability, so the priori probability of each codeword is a const. We now derive a recursive technique for calculating α and β

$$\begin{aligned}\alpha_{k+1}(q) &= p(\Psi_{k+1} = q; \mathbf{c}_{l < k+1}) \\ &= \sum_{p=1}^{p=Q} p(\Psi_{k+1} = q; \mathbf{c}_k; \Psi_k = p; \mathbf{c}_{l < k}) \\ &= \sum_{p=1}^{p=Q} p(\Psi_{k+1} = q; \mathbf{c}_k | \Psi_k = p) \cdot p(\Psi_k = p; \mathbf{c}_{l < k}) \\ &= \sum_{p=1}^{p=Q} \gamma_k(p, q) \cdot \alpha_k(p)\end{aligned}\quad (2.39)$$

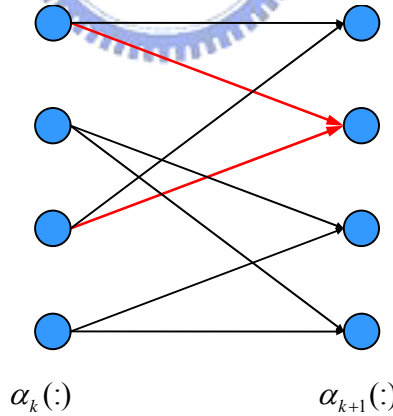


Fig. 2-10: Recursive calculating of α in trellis diagram

then

$$\begin{aligned}
\beta_k(p) &= p(\mathbf{c}_{l>k-1} | \Psi_k = p) \\
&= \sum_{q=1}^{p=Q} p(\mathbf{c}_{l>k-1}; \mathbf{c}_k; \Psi_{k+1} = q | \Psi_k = p) \\
&= \sum_{q=1}^{p=Q} p(\mathbf{c}_{l>k-1} | \Psi_{k+1} = q) \cdot p(\mathbf{c}_k; \Psi_{k+1} = q | \Psi_k = p) \\
&= \sum_{q=1}^{p=Q} \beta_{k+1}(q) \cdot \gamma_k(p, q)
\end{aligned} \tag{2.40}$$

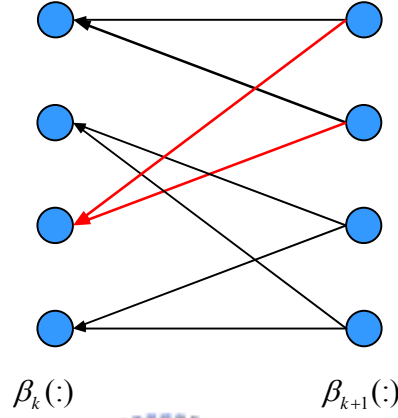


Fig. 2-11: Recursive calculating of β in trellis diagram

We now make two passes, α forward pass from beginning to the end and β backward pass from finish to start. After we calculate all state probabilities in the trellis, we can calculate the information bits LLR by

$$\begin{aligned}
LLR(b_k) &= \log\left(\frac{\Pr(b_k = 1 | \mathbf{c})}{\Pr(b_k = 0 | \mathbf{c})}\right) \\
&= \log\left(\frac{\sum_{(p,q) \in B_k^{(1)}} \alpha_k(p) \cdot \gamma_k(p, q) \cdot \beta_{k+1}(q)}{\sum_{(p,q) \in B_k^{(0)}} \alpha_k(p) \cdot \gamma_k(p, q) \cdot \beta_{k+1}(q)}\right)
\end{aligned} \tag{2.41}$$

and we also re-calculate the coded bits LLR by

$$\begin{aligned}
LLR(\tilde{c}_{k,i}) &= \log\left(\frac{\Pr(c_{k,i} = 1|\mathbf{c})}{\Pr(c_{k,i} = 0|\mathbf{c})}\right) \\
&= \log\left(\frac{\sum_{(p,q) \in C_{k,i}^{(1)}} \alpha_k(p) \cdot \gamma_k(p,q) \cdot \beta_{k+1}(q)}{\sum_{(p,q) \in C_{k,i}^{(0)}} \alpha_k(p) \cdot \gamma_k(p,q) \cdot \beta_{k+1}(q)}\right) \\
&= \underbrace{\log\left(\frac{p(c_{k,i} = 1)}{p(c_{k,i} = 0)}\right)}_{L_{A_2}} + \underbrace{\log\left(\frac{\sum_{(p,q) \in C_{k,i}^{(1)}} \alpha_k(p) \cdot \gamma_k(p,q) \cdot \beta_{k+1}(q)}{\sum_{(p,q) \in C_{k,i}^{(0)}} \alpha_k(p) \cdot \gamma_k(p,q) \cdot \beta_{k+1}(q)}\right)}_{L_{E_2}} - LLR(c_{k,i})
\end{aligned} \tag{2.42}$$

2.7.2 List Sphere Decoding with Priori Information

With the coded bits priori information calculated by the outer BCJR decoder, we can modify the original LSD to have another form. Meanwhile, we assume that we are working on a block of bits corresponding to one symbol vector φ_j . The a posteriori LLR of the coded bit on certain antenna, conditioned on the received symbol vector \mathbf{r} , is:

$$LLR(c_{i,k}) \triangleq \log\left(\frac{p(c_{i,k} = 1|\mathbf{r})}{p(c_{i,k} = 0|\mathbf{r})}\right) \tag{2.43}$$

We now assume the bits on φ_j have been encoded with channel, and the interleaver is used to scramble the bits from other antennas. So, the bits within φ_j are approximately statistically independent of one another. Using the Bayes' theorem, and splitting the joint probability, we can have the soft output value as

$$L_D(c_{k,i}) = L_A(c_{k,i}) + \log\left(\frac{\sum_{\varphi_j \in \mathcal{S}_{k,i}^{(1)}} p(\mathbf{r}|\varphi_j) \cdot \exp\left(\sum_{l \in \Omega_k} L_A(c_{k,i})\right)}{\sum_{\varphi_j \in \mathcal{S}_{k,i}^{(0)}} p(\mathbf{r}|\varphi_j) \cdot \exp\left(\sum_{l \in \Omega_k} L_A(c_{k,i})\right)}\right) \tag{2.44}$$

$L_E(b_{k,i})$

where $\mathcal{S}_{k,i}^{(1)}$ is the set of $2^{N_T \cdot M - 1}$ symbol vectors having i -th bits on k -th transmit antenna

$b_{k,i} = 1$. Ω_k is the set of indices l with

$$\Omega_k = \{l | l = 0, \dots, N_T \cdot M - 1, l \neq (k, i)\} \quad (2.45)$$

$$L_A(c_{k,i}) = \log\left(\frac{P(c_{k,i} = 1)}{P(c_{k,i} = 0)}\right) \quad (2.46)$$

Multiplying the numerator and the denominator with $\exp(-1/2 \cdot \sum_{l=0}^{N_T \cdot M - 1} L_{A_l}(c_{k,i}))$, we can rewrite (2.44) as

$$\begin{aligned} L_D(c_{k,i}) &= L_{A_i}(c_{k,i}) + \log\left(\frac{\sum_{\varphi_j \in \mathbf{s}_{k,i}^{(1)}} p(\mathbf{r} | \varphi_j) \cdot \exp\left(\frac{1}{2} \mathbf{b}_{[j]}^T \cdot \mathbf{L}_{A,[j]}\right)}{\sum_{\varphi_j \in \mathbf{s}_{k,i}^{(0)}} p(\mathbf{r} | \varphi_j) \cdot \exp\left(\frac{1}{2} \mathbf{b}_{[j]}^T \cdot \mathbf{L}_{A,[j]}\right)}\right) \end{aligned} \quad (2.47)$$

$L_{E_1}(c_{i,k})$

where $\mathbf{b}_{[j]}$ denotes the vector which omitting its j -th element, which is the position of $c_{k,i}$.

$$\begin{aligned} \mathbf{b} &= [b_1 \ b_2 \ \dots \ b_{N_T \cdot M}] \\ b_i &= \begin{cases} 1 & \text{if } \varphi_j \in \mathbf{s}_{k,i}^{(1)} \\ -1 & \text{if } \varphi_j \in \mathbf{s}_{k,i}^{(0)} \end{cases} \end{aligned} \quad (2.48)$$

and $\mathbf{L}_{A,[j]}$ denotes the vector form of all L_{A_l} except the same term. Further, we can use the max-log approximation to simplify (2.47), that is

$$\begin{aligned} LLR(c_{i,j}) &\approx L_{A_i}(c_{k,i}) + \frac{1}{2} \left\{ \frac{1}{\sigma^2} \min_{\varphi_j \in \mathbf{s}_{k,i}^{(0)}} \{\|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\} + \mathbf{b}_{[j]}^T \cdot \mathbf{L}_{A,[j]} \right\} \\ &\quad - \frac{1}{2} \left\{ \frac{1}{\sigma^2} \min_{\varphi_j \in \mathbf{s}_{k,i}^{(1)}} \{\|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2\} + \mathbf{b}_{[j]}^T \cdot \mathbf{L}_{A,[j]} \right\} \end{aligned} \quad (2.49)$$

L_{E_1}

2.7.3 MIMO Detection Using the List Sphere Decoder

As mentioned, we use the LSD to help find the candidates that can be used to compute soft bit information. If the candidate list generated by LSD is \mathcal{L}_{LSD} , (2.49) can be approximated using

the candidates in \mathcal{L}_{LSD} as

$$LLR(c_{i,j}) \approx L_{A_2}(c_{k,i}) + \frac{1}{2} \left\{ \frac{1}{\sigma^2} \min_{\varphi_j \in \mathcal{L}_{LSD} \cap \mathcal{S}_{k,i}^{(0)}} \{ \|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2 \} + \mathbf{b}_{[k]}^T \cdot \mathbf{L}_{A,[k]} \right\} - \underbrace{\frac{1}{2} \left\{ \frac{1}{\sigma^2} \min_{\varphi_j \in \mathcal{L}_{LSD} \cap \mathcal{S}_{k,i}^{(1)}} \{ \|\mathbf{r} - \mathbf{H} \cdot \varphi_j\|^2 \} + \mathbf{b}_{[k]}^T \cdot \mathbf{L}_{A,[k]} \right\}}_{L_{E_2}} \quad (2.50)$$

There is also tradeoff between the accuracy of (2.50) and the computational complexity of the LSD. When the size of the list is too small, the soft bit information is not accurate enough such that the “turbo” scheme may not work. The reason is that \mathcal{L}_{LSD} may not contain the true transmitter symbol vector, the coded bits LLR generated by the outer coder will not contain the soft information of the true transmitter symbol vector. As a result, the LLR will not be increased toward the right direction. On the other hand, if the size of the list is too large, the computational complexity of the LSD will become too high, adversely affecting the applicability of the turbo equalization scheme in real-world systems.

2.8 Simulation Results

Our simulation platform is developed based on the IEEE802.11n proposal. We use the mandatory mode in which the constraint length of the convolutional code is 7, the channelization bandwidth is 20MHz (56 occupied sub-channels), and the number of spatial streams is two $N_r=2$ and $N_t=4$. The number of receive antennas is the same as that of transmit antennas. Two different systems with the same throughput are considered.

- (a) Conventional equalization and SISO demapping: An MMSE equalizer, a SISO soft-bit demapper, and a BCJR decoder are used at the receiver. In the figures shown later, we use “SB” to denote this approach.
- (b) Turbo Equalization: A LSD detector, MIMO soft-demapper, a BCJR decoder, and the iterative procedure describe above are used at the receiver. We call the system

“TE” in the figures shown below. The dash “TE” indicates the iteration number.

Assume that the frequency offset and timing offset are perfectly compensated at the receiver. We use the HT-LTF of the preamble and the standard per-tone channel estimation method (no smoothing) to estimate MIMO channels. The PPDU length is set as 1000 bytes; so there are 8000 information bits in each packet. The packet error rate (PER) is used as the performance measure. We choose the channel-B (non-line-of-sight) [15], and the channel-D (non-line-of-sight) as simulation environments. Fig. 2-12 to Fig. 2-17 show the simulation results for the perfect-channel cases. Fig. 2-18 to Fig. 2-23 show the results for the estimated-channel cases. For all case, the turbo scheme performs much better than the conventional schemes. From Fig. 2-12 to Fig. 2-17, we observe that when PER is 0.1 and channel-B is considered, the turbo scheme with zero iteration outperforms the conventional scheme about 3dB in 2X2 16-QAM, 10dB in the 4X4 16-QAM, 12dB in 4X4 64-QAM scenarios. With iterations, the performance is further improved. The performance improves about additional 2dB with one iteration, and further 0.5dB with four iteration. In channel-D environment, the improvement of the turbo scheme with zero iteration is less than channel-B. But it still has 3dB, 7dB, and 9dB gain in the 2X2 16-QAM, 4X4 16-QAM, and 4X4 64-QAM scenarios. From Fig. 2-15 to Fig. 2-17, we can see that the gain due to iterations is almost the same as channel-B. From the result, we can conclude that the MIMO soft-demapping is the major factor for performance improvement. From implementation point of view, we can see that the turbo scheme only requires one iteration.

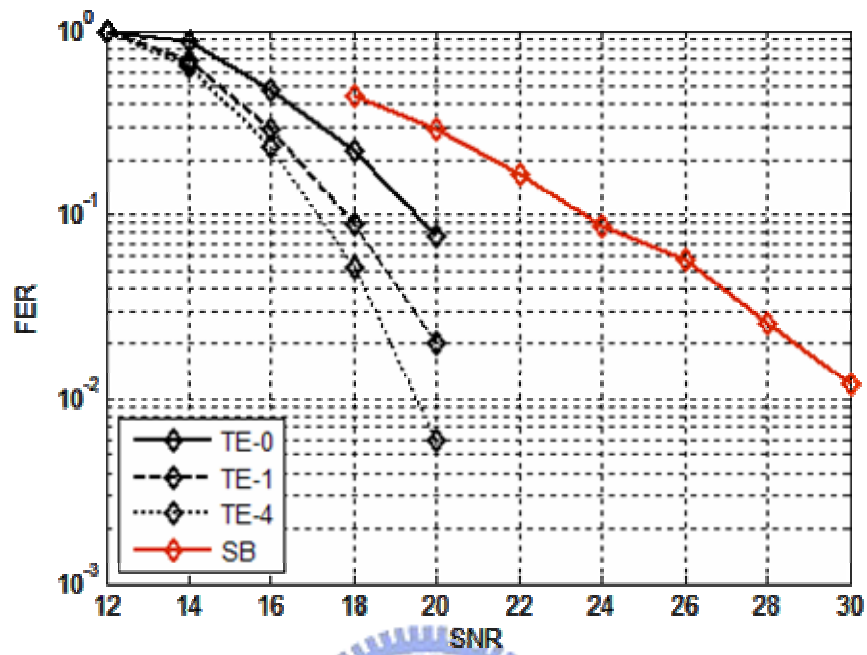


Fig. 2-12: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 2x2, 16-QAM, channel-B, perfect-channel)

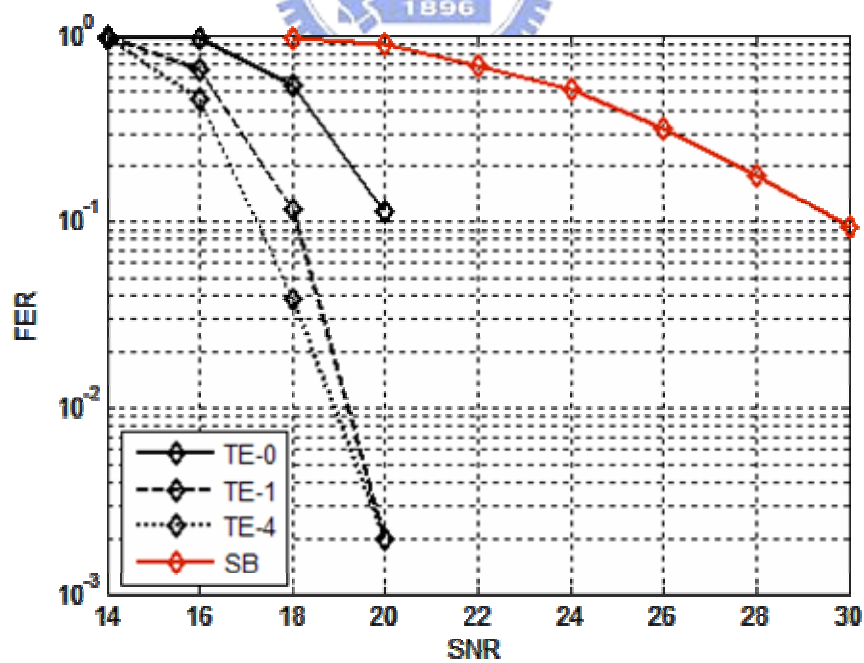


Fig. 2-13: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 16-QAM, channel-B, perfect-channel)

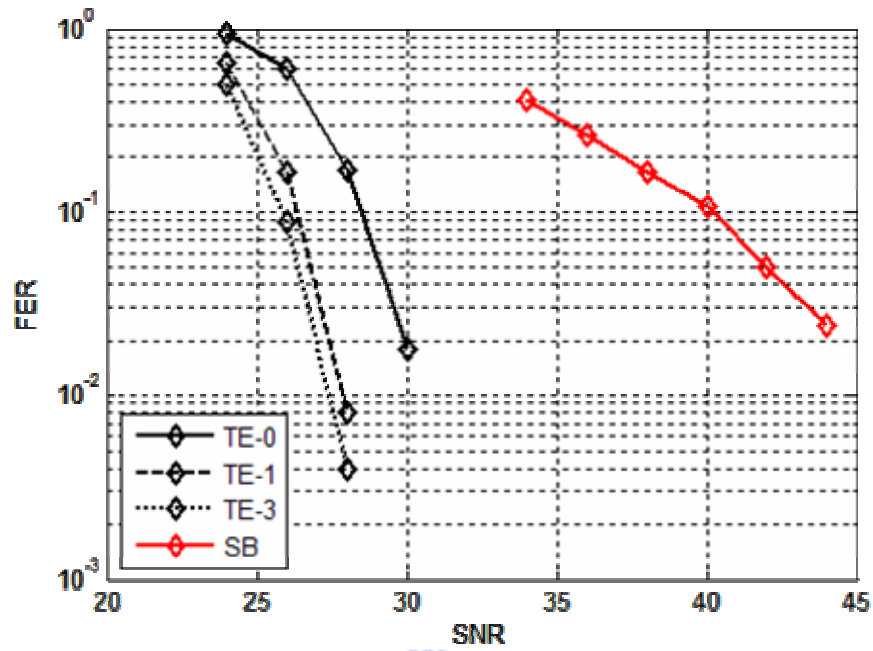


Fig. 2-14: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 64-QAM, channel-B, perfect-channel)

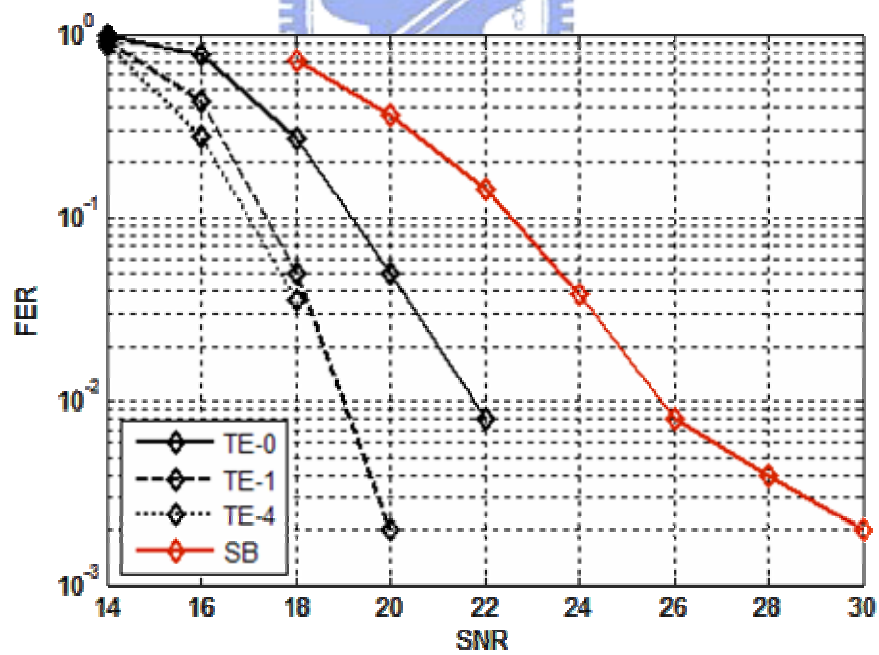


Fig. 2-15: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 2x2, 16-QAM, channel-D, perfect-channel)

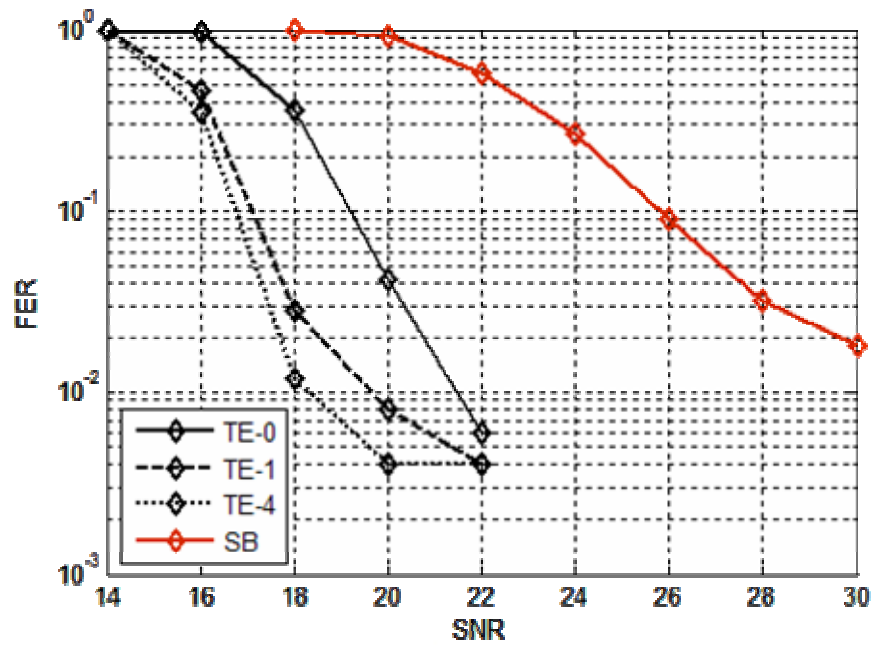


Fig. 2-16: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 16-QAM, channel-D, perfect-channel)

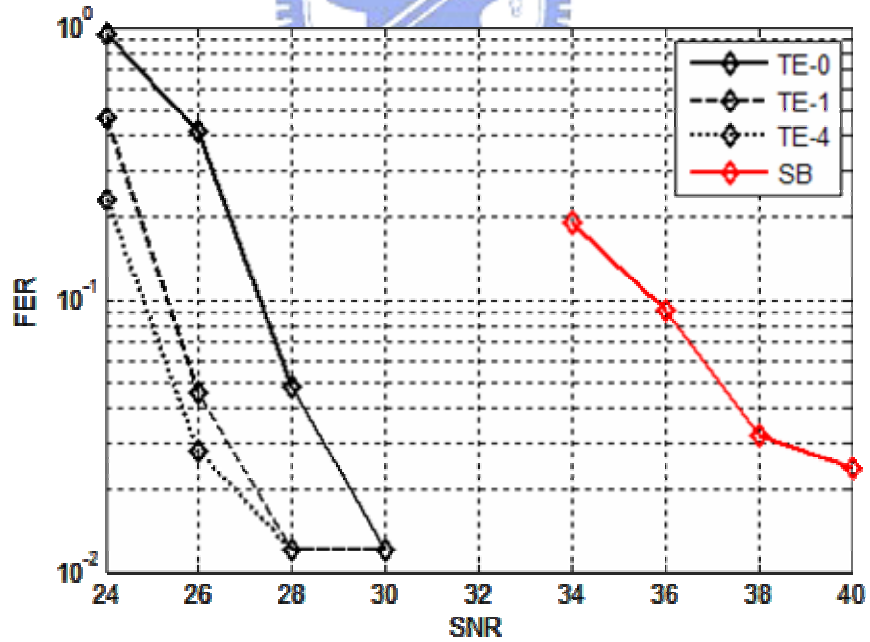


Fig. 2-17: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 64-QAM, channel-D, perfect-channel)

From Fig. 2-18 to Fig. 2-23 show the simulation results for estimated channels. The standard per-tone channel estimation method (no smoothing) is used to estimate the MIMO channel. In channel-B, the improvement of the turbo scheme with zero iteration is 4dB, 10dB, and 11dB in the 2X2 16-QAM, 4X4 16-QAM and 4X4 64-QAM scenarios, respectively. The gain, due to iterations, is about 2 dB with one iteration, and additional 0.5dB with four iterations. In channel-D, the improvement with zero iteration is 3dB, 6dB, and 8dB in the 2X2 16-QAM, 4X4 16-QAM and 4X4 64-QAM scenarios, respectively. The gain due to iterations is similar to that in channel-B.

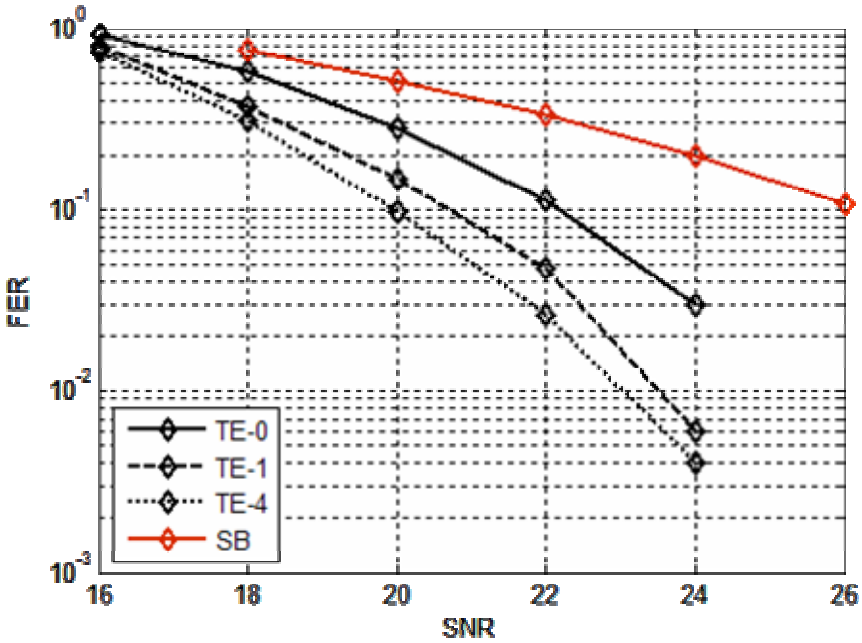


Fig. 2-18: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 2x2, 16-QAM, channel-B, estimated-channel)

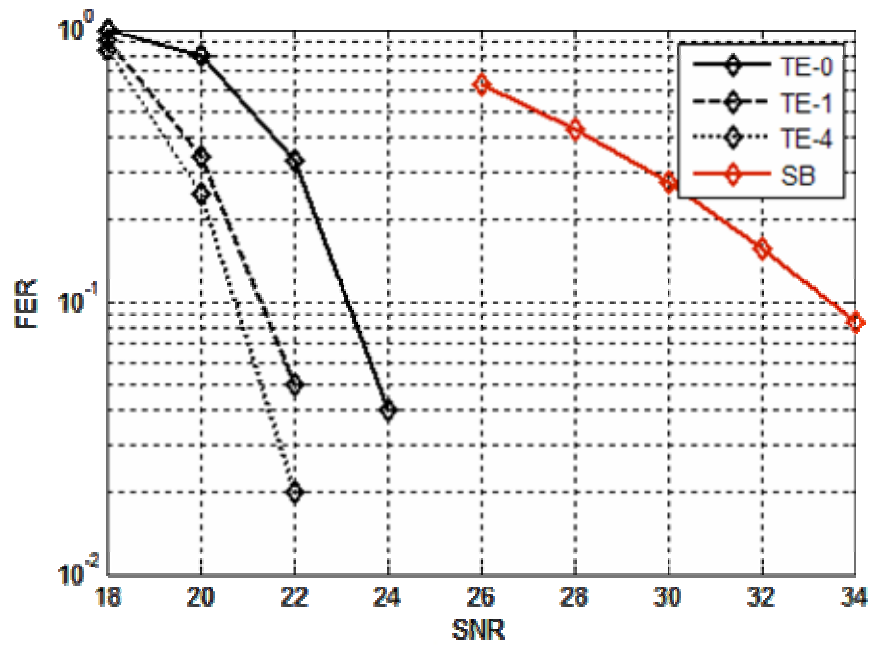


Fig. 2-19: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 16-QAM, channel-B, estimated-channel)

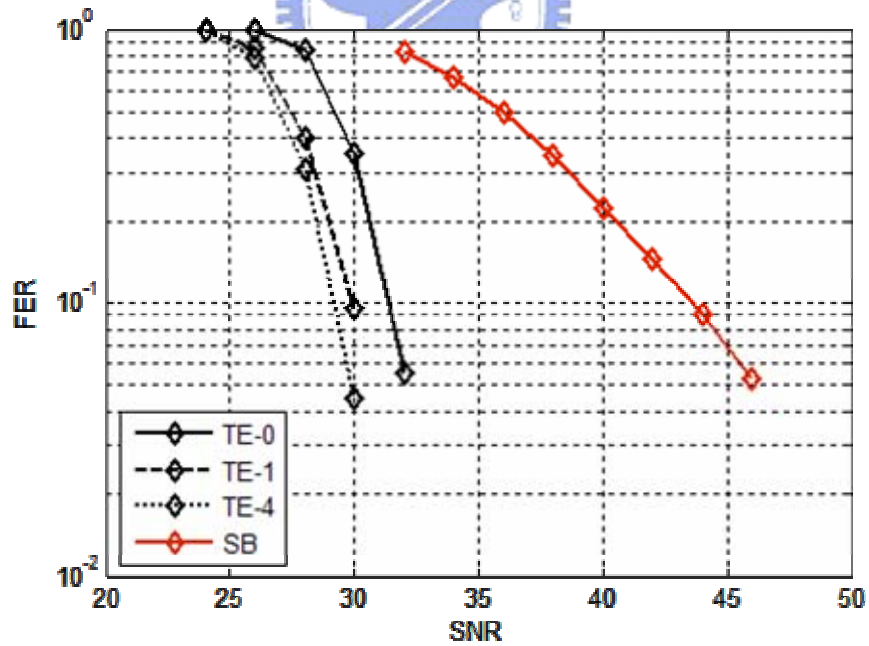


Fig. 2-20: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 64-QAM, channel-B, estimated-channel)

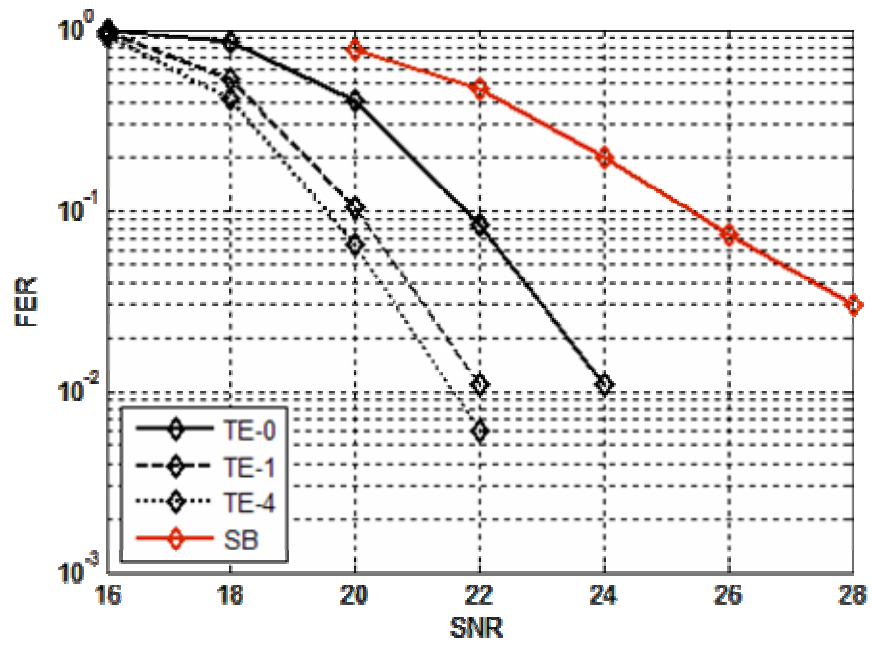


Fig. 2-21: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 16-QAM, channel-D, estimated-channel)

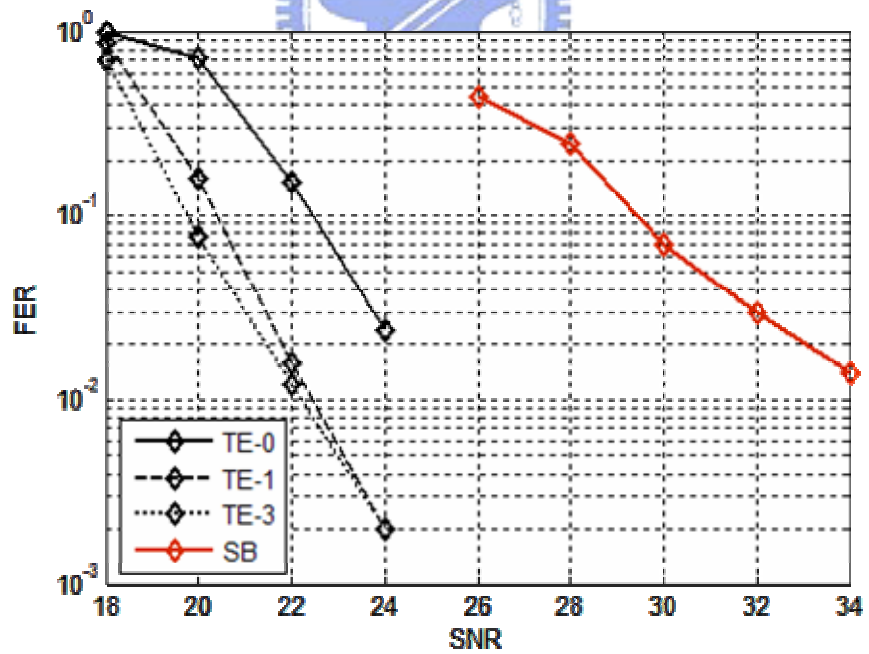


Fig. 2-22: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 16-QAM, channel-D, estimated-channel)

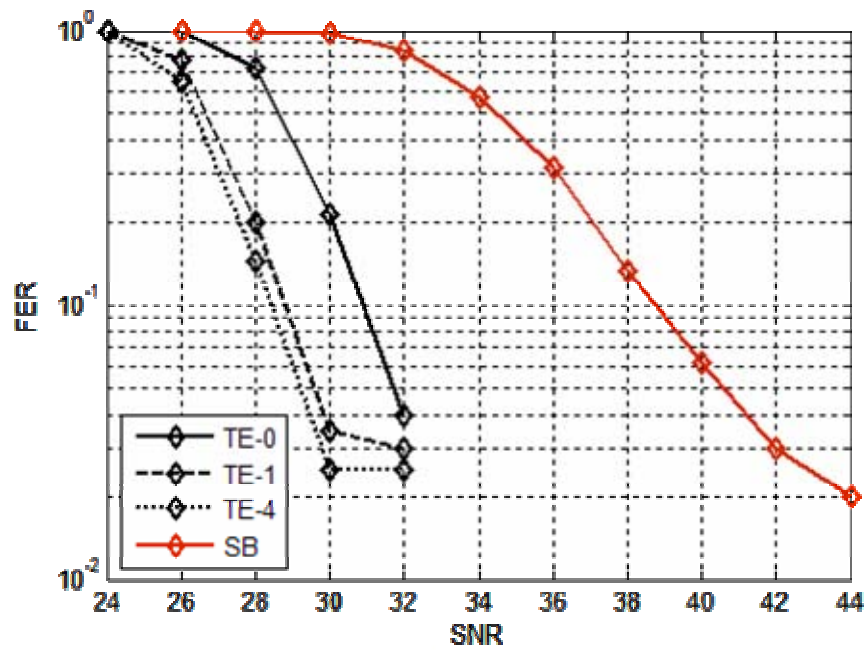
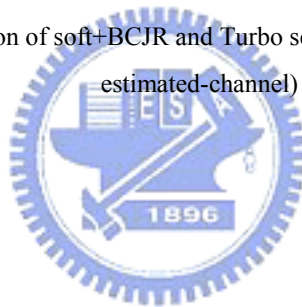


Fig. 2-23: Performance comparison of soft+BCJR and Turbo scheme (20MHz, 4x4, 64-QAM, channel-D, estimated-channel)



Chapter 3 Tone Interleaved Coded Modulation with Turbo Code

MIMO-OFDM with BICM is widely used as a baseband transmission scheme in recently years. With a LSD detector, and a MIMO soft-demapper at the receiver, we have shown that the performance of a MIMO-OFDM system can be greatly improved. We have also shown that the turbo equalization technique mentioned in Chapter 2 can further enhance the performance. Due to the interleaving operations in BICM, the mapped symbols transmitted in each antenna do not have dependency. Now, consider another scenario that if interleaving is conducted with a block base, where the block size is the number of symbols transmitted in one short (via. all antennas). The coded bits are sequentially mapped to the QAM symbols and transmitted via all antennas. In this case, there is strong dependency between the QAM transmitted symbols. In other words, the transmitted symbol posses a trellis structure and bits can be directly solved through the structure without soft-demapping. In this case, the observations of the trellis are the received symbols themselves. Note that in BICM, the observations are the demapped soft-bits. With the structure, we can take the advantage of the dependency between symbols in a data block, and the performance of the BCJR algorithm can be improved. This the main idea of tone-interleaved coded modulation (TICM) proposed in [11].

TICM is to use a symbol block as a unit for interleaving; a symbol block contains all the symbols transmitted (by all available antennas) at a single tone. As mentioned, the algorithm does not require soft-bit demapping, which is absorbed into the BM calculator. In [11], only the CC encoder is considered, and the performance can be improved only when the channel response is short. The performance gain is not apparent when the channel response is long. This is due to the

fact that more fades tend to exist in the channel frequency response, and the block interleaver does not have good resistance to bursty bit errors. In this Chapter, we proposed to apply the turbo code in the TICM systems. Since a bit-level interleaver is introduced between two component encoders, the problem of the bursty error can be effectively reduced.

3.1 MIMO-OFDM with TICM-T

The main advantage of TICM-T is to merge soft-bit demapping into the branch metric calculation. Taking the advantage of the trellis structure inherent in the BCJR algorithm, the TICM-T can achieve the performance that BICM-T cannot. Note that TICM applies only for MIMO systems. In MIMO-OFDM systems, coding is usually conducted along the direction of tone index. This makes the receiver look like experiencing a fast fading channel (though it is in the frequency domain). Due to the interleaver inserted between two component encoders, bursty errors produced by the equivalent fast fading channel can be distributed and corrected. Note that not all kinds of the turbo codes can be applied here. We use the parallel-concatenated-trellis-coded-modulation turbo code proposed by Benedetto, Divsalar, Montorsi and Pollara in 1996 [12]. The reason why this type of turbo code can be used in TICM-T is explained in the following section. Still, the BCJR algorithm is chosen to exchange the extrinsic information between two component decoders. We name this scheme as a TICM-T scheme, and its counterpart scheme a BICM-T scheme. The BICM-T uses the same turbo code as that of the TICM-T except for the BICM is used. Simulation results show that the performance of this TICM-T is better than that BICM-T.

3.2 Transmitter for TICM-T

Fig. 3-1 shows the block diagram of the MIMO-OFDM transmitter for TICM-T system. First,

information bits are encoded by the encoder, and the coded bits are mapped to QAM symbols, sequentially. Since the bit-level interleaver between encoder and QAM mapper is removed, consecutive coded bits are mapped to symbols consecutively. Thus, symbols have strong dependency and possess a trellis structure. After tone-level interleaving, \tilde{s}^j is parsed to each antenna. The remained transmission process is the same as the BICM.

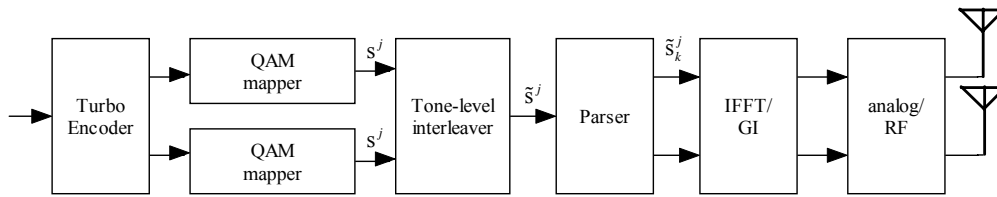


Fig. 3-1: The block diagram of the MIMO-OFDM transmitter for TICM-turbo

3.2.1 Turbo Encoder

The original turbo encoder, which is formed by parallel concatenation of two recursive systematic convolutional (RSC) encoders separated by a random interleaver [13], is not proper for TICM-T. This is because the TICM scheme requires that the consecutive coded bits (systematic and parity check bits) must be transmitted in the same symbol. The original turbo code has a code rate of 1/3. If one systematic bit and one parity check bit are mapped to a symbol, the other parity check bit cannot be properly mapped. Thus, to apply TICM-T, one extra systematic bit stream will be needed. However, this will make systematic bits being transmitted repeatedly. Here, we use a structure designed for turbo trellis coded modulation (TCM) [14]. The basic idea of this scheme is to transmit two systematic and two parity check bits simultaneously. The code rate is then 2/4 (= 1/2). The outputs of the coded bits can then be mapped into two symbol sequences (at a time), and transmitted in different antennas. Figure 3-2 shows the encoder structure of the TICM-T.

If n is the number of information bits per frame, the encoder structure consists of two RSC encoders linked in parallel. Each encoder receives two information bits, generating one parity check bit and one systematic bit per time slot. The first $\frac{n}{2}$ information bits are fed to the upper component encoder as one stream of systematic bits, and the second $\frac{n}{2}$ information bits are interleaved and fed to the lower component encoder as another stream of systematic bits.

However, the states of both component encoders are influenced by all n information bits. In this way, the total number of the coded bit becomes $2 \cdot 2 \cdot \left(\frac{n}{2} + 2\right)$. Fig. 3-2 shows an example of the component encoder with $n=2$, $k=2$, and $K_{RSC} = 3$, where K_{RSC} represents the constraint length of the RSC encoder. Note that there are two different bit-level S-random interleavers (π_1 and π_2) between two component encoders. In our simulations, we let the size be 4000 and S is equal to 35. Finally, a consecutive $N_T \cdot N_{bpsc}$ coded bits (systematic and parity check bits alternately) encoded from each component encoder are mapped into $N_T \times 1$ transmitted symbol vector s^j , respectively. Here, N_{bpsc} represents the number of bits transmitted per QAM symbol at a time instant. Note that with our formulation, M-QAM symbols can be mapped, where $M=2^L$ and L is an integer.

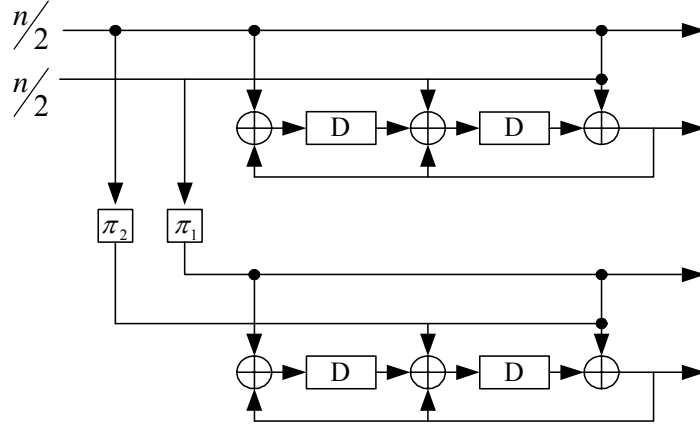


Fig. 3-2: The turbo encoder for TICM-T

3.2.2 Tone-Level Interleaver for TICM-T

We use the simplest interleaver, i.e., the row-in column-out tone level interleaver. Let $s^j = [s_1^j, \dots, s_{N_T}^j]$, where j denotes the index of the transmission data block. Thus, s^j contains the symbols transmitted at one tone and one certain time. We can the N_T symbols as a block. The interleaver uses a block of QAM symbols, mapped from $N_{bpt} = N_T \cdot N_{bpsc}$ continuous coded bits, as its basic unit to perform interleaving. The index j starts from 1 to " $L_s \cdot N_{tone}$ ", where L_s represents the number of OFDM symbols per frame, and N_{tone} is the number of OFDM tone. Thus, there are total $L_s \cdot N_{tone}$ transmitted symbol vectors at each transmission frame.

3.3 Receiver for TICM-T

Let the received symbol vector in a block be denoted by $\tilde{\mathbf{r}}^j$ and the correspondingly estimated MIMO-channel by $\tilde{\mathbf{H}}^j$. Properly combining all $\tilde{\mathbf{r}}^j$, we can reconstruct the block sequence formed in each component coder in the transmitter. We can see that all symbols are related with a trellis structure. However, due to the MIMO channel, the trellis inside a block is

transformed by $\tilde{\mathbf{H}}^j$. As a result, we cannot use the regular BCJR algorithm to obtain the soft bits information. To solve the problem, we can transit from the beginning of a block to its end, yielding a parallel transition. Note that the parallel transition can also be interpreted as a radix processing structure. The problem becomes how to evaluate the branch metric (BM) of the parallel transition. Since the coded bits are directly mapped without further bit interleaving, we can then evaluate its soft information of information bits, rather than the coded bits. In addition, the radix processing structure is employed. Thus, the original BCJR has to be modified. Once all BMs are calculated, they can be fed into the modified BCJR decoder to calculate soft information bits and LLR's of coded bits. After de-interleaving, soft information bits are fed to the other component decoder as the priori information, and vice versa. Note that we can use the LSD method to reduce the number of candidate symbol vectors in the BM calculation. Also, LLR's of coded bits can then be used to conduct detection such that the candidates in the LSD detector can be reduced. The block diagram of the MIMO-OFDM receiver for TICM-turbo is shown in Fig.

3-3

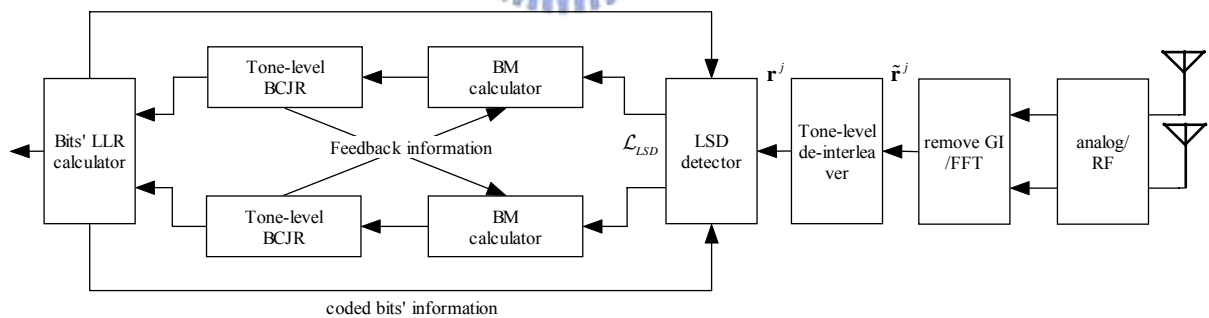


Fig. 3-3: The block diagram of the MIMO-OFDM receiver for TICM-turbo

3.3.1 BM Calculation in TICM

Assume that all output symbols are transmitted with equal probabilities. The channel is memoryless (this is true in OFDM), and \mathbf{r}^j obeys complex Gaussian distribution. The

probability of a candidate symbol vector φ_k with observation \mathbf{r}^j is

$$p(\mathbf{r}^j | \varphi_k) = \frac{1}{(\pi\sigma^2)^{N_T}} \exp\left\{-\frac{1}{2\sigma^2} \|\mathbf{r}^j - \mathbf{H}^j \cdot \varphi_k\|^2\right\} \quad (3.1)$$

where \mathbf{r}^j represents the de-interleaved received signal block, \mathbf{H}^j is the corresponding estimated MIMO-channel, $\varphi_k = [\varphi_1, \dots, \varphi_{N_T}]$ is the possible candidate symbol vector. From 3.2.2, we find that there are total $2^{N_{bpt}}$ possible symbol vectors at stage j . Note that j is also the index of transmission data block. Thus, the required computational complexity for evaluating (3.1) can be high. In order to reduce the computational complexity, we use the LSD algorithm, mentioned before in 2.6.2, to alleviate the problem.

The priori probability of a candidate symbol vector can be calculated by its LLR of information bits. With the extrinsic information calculated by other component decoder, we can weight (3.1) with the priori information;

$$\begin{aligned} p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k \in \Psi^{(p,q)}) &= p(\mathbf{r}^j | \varphi_k) \cdot p(\mathbf{c}_k \in \Psi^{(p,q)}) \\ &= p(\mathbf{r}^j | \varphi_k) \cdot p(\mathbf{b}_k^{(p,q)}) \\ &= \frac{1}{(\pi\sigma^2)^{N_T}} \exp\left\{-\frac{1}{2\sigma^2} \|\mathbf{r}^j - \mathbf{H}^j \cdot \varphi_k\|^2\right\} \cdot \exp\left\{\sum_{b_k \in \varphi_k} L_A(b_k)\right\} \end{aligned} \quad (3.2)$$

where \mathbf{c}_k is the set of coded bits for φ_k , $\mathbf{b}_k^{(p,q)}$ is the set of information bits for φ_k under the transition from state p to state q at stage j . We call $p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k \in \Psi^{(p,q)})$ as the BM from state p to state q for φ_k .

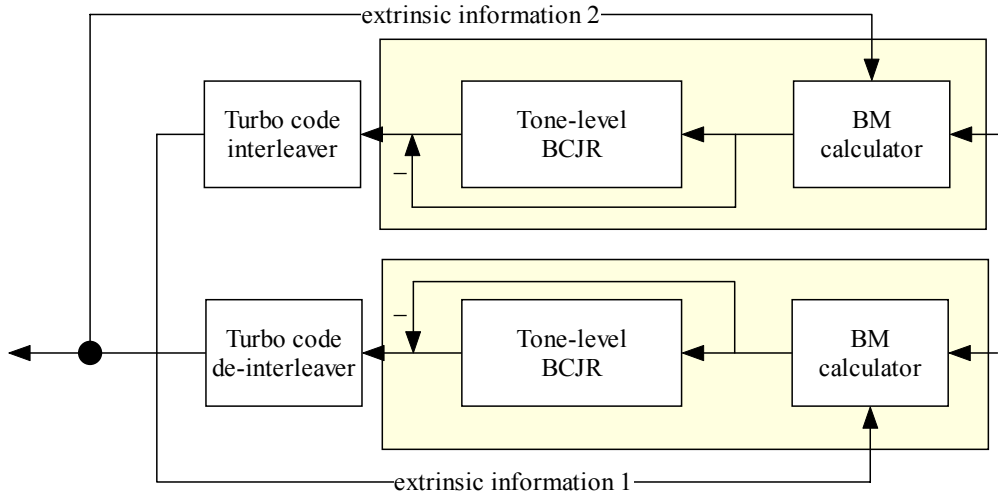


Fig. 3-4: The iteration loop of TICM-T

3.3.2 Modified BCJR Algorithm for TICM-T

After calculating the BM, we use MAP (BCJR) algorithm to calculate probabilities of information bits. The original BCJR algorithm shown in 2.4.1 uses the joint probability of two coded bits as the transition probability in one stage. In the TICM-T scheme, since we use contiguous blocks of N_{bpt} RSC coded bits to conduct symbol-vector mapping without bit-level interleaver, the trellis structure in the RSC code can also be constructed for contiguous symbol vectors. That is similar to a radix- $2^{N_{ibps}}$ structure, where N_{ibps} represents the number of information bits in one symbol vector. Thus, we modify the original bit-level BCJR algorithm, which uses the joint probability of two coded bits as the state transition probability, to a tone-level BCJR algorithm, which uses N_{bpt} the joint probability of contiguous coded bits as the state transition probability.

According to (2.32), we can change the k -th information bit probability in j -th symbol vector

b_k to:

$$\Pr[b_k^{(1)} | \mathbf{r}^j] = \sum_{\varphi_k \in \Psi^{(1)}} p(\varphi_k | \mathbf{r}^j) = \sum_{\varphi_k \in \Psi^{(1)}} \frac{p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k)}{p(\mathbf{r}^j)} \quad (3.3)$$

where $\Psi^{(1)}$ indicates the subset of all possible symbol vectors, whose k -th information bit is 1, \mathbf{r}^j is the received symbol vector. From (2.35), we can decompose the posterior probability as three parts. Starting from γ_j :

$$\gamma_j(p, q) = \Pr[\mathbf{r}^j | \Phi_k = p; \Phi_{k+1} = q] \cdot \Pr[\Phi_{k+1} = q | \Phi_k = p] \quad (3.4)$$

With the radix-2 ^{N_{ibps}} structure, the BM (the jointly probability of possible and received symbol vector from state p to state q) is not unique. In other words, γ_k indicates the summation of all BMs transiting from state p to state q at stage j . This is what we called the parallel transition.

We can rewrite (3.4) as:

$$\begin{aligned} \gamma_j(p, q) &= \left(\sum_{\varphi_k \in \Psi^{(p,q)}} p(\mathbf{r}^j | \varphi_k) \right) \cdot \Pr[\varphi_k \in \Psi^{(p,q)}; \Phi_k = p] / \Pr[\Phi_k = p] \\ &= \left(\sum_{\varphi_k \in \Psi^{(p,q)}} p(\mathbf{r}^j | \varphi_k) \right) \cdot \Pr[\varphi_k \in \Psi^{(p,q)}] \cdot \Pr[\Phi_k = p] / \Pr[\Phi_k = p] \\ &= \sum_{\varphi_k \in \Psi^{(p,q)}} p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k) \end{aligned} \quad (3.5)$$

where $\Psi^{(p,q)}$ is the subset of the symbol vectors transiting from state p to state q at stage j , $p(\varphi_k)$ is the priori probability of the symbol vector, and $p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k)$ is the BM output.

Once we derive γ_j , α and β can be calculated recursively from stage 1 to stage $L_s \cdot N_{tone}$ by the method outlined at 2.7.1. Finally, we can calculate LLRs of information bits by the probability measure α , β , and γ .

$$\begin{aligned}
LLR(b_k) &= \log\left(\frac{\Pr(b_k = 1|\mathbf{r})}{\Pr(b_k = 0|\mathbf{r})}\right) \\
&= \log\left(\frac{\sum_{(p,q) \in \Psi_k^{(1)}} \alpha_k(p) \cdot \left(\sum_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q)} \cap B_k^{(1)}} p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k) \right) \cdot \beta_{k+1}(q)}{\sum_{(p,q) \in \Psi_k^{(0)}} \alpha_k(p) \cdot \left(\sum_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q)} \cap B_k^{(0)}} p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k) \right) \cdot \beta_{k+1}(q)}\right) \quad (3.6)
\end{aligned}$$

where $\Psi_k^{(1)}$ is the pairs of state transition which contains the symbol vectors with $b_k = 1$, $B_k^{(1)}$ is the subset of symbol vectors with $b_k = 1$, and \mathcal{L}_{LSD} is the list of possible symbol vectors within the LSD radius. And we can further simplify (3.6) by using the max-log theorem twice:

$$\begin{aligned}
LLR(b_k) &= \log\left(\frac{\max_{(p,q) \in B_k^{(1)}} \left\{ \alpha_k(p) \cdot \max_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q)} \cap B_k^{(1)}} \left\{ p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k) \right\} \cdot \beta_{k+1}(q) \right\}}{\max_{(p,q) \in B_k^{(0)}} \left\{ \alpha_k(p) \cdot \max_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q)} \cap B_k^{(0)}} \left\{ p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k) \right\} \cdot \beta_{k+1}(q) \right\}}\right) \\
&= \max_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q) \in B_k^{(1)}} \cap B_k^{(1)}} \left\{ \log(\alpha_k(p)) + \frac{1}{2} \left\{ -\frac{1}{\sigma^2} \|\mathbf{r}^j - \mathbf{H} \cdot \varphi_k\|^2 + \mathbf{b}_{[k]}^T \cdot \mathbf{L}_{A,[k]} \right\} + \log(\beta_{k+1}(p)) \right\} \\
&\quad - \underbrace{\max_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q) \in B_k^{(0)}} \cap B_k^{(0)}} \left\{ \log(\alpha_k(p)) + \frac{1}{2} \left\{ -\frac{1}{\sigma^2} \|\mathbf{r}^j - \mathbf{H} \cdot \varphi_k\|^2 + \mathbf{b}_{[k]}^T \cdot \mathbf{L}_{A,[k]} \right\} + \log(\beta_{k+1}(p)) \right\}}_{L_E(b_k)} \\
&\quad + L_A(b_k) \quad (3.7)
\end{aligned}$$

where $\mathbf{b}_{[k]}$ denotes the vector omitting its k -th element,

$$\begin{aligned}
\mathbf{b} &= [b_1 \ b_2 \ \dots \ b_{N_{ibps}}] \\
b_i &= \begin{cases} 1 & \text{if } \varphi_k \in B_k^{(1)} \\ -1 & \text{if } \varphi_k \in B_k^{(0)} \end{cases} \quad (3.8)
\end{aligned}$$

and $\mathbf{L}_{A,[k]}$ denotes the vector form of extrinsic information generated by the other component decoder (excluding k -th term also).

3.3.3 The Modified LSD detector with the Priori Information

We use symbol vector instead of coded bits as a basic unit in the BCJR Algorithm. With this approach, we have added the dependency between contiguous symbol vectors. As mentioned, we use the LSD algorithm to reduce the required computational complexity. One problem with the LSD algorithm is that transmitted symbol vectors may not be included in the list \mathcal{L}_{LSD} . If this does happen, the error rate of the TICM-T scheme will be dominated by those bits belonging to the symbol vectors. This kind of error cannot be corrected through iteration. We can enlarge the radius in the LSD algorithm to include more candidates. However, the computational complexity of the LSD will become higher. Here, we propose a method to alleviate this problem. From the modified BCJR algorithm in (2.42), we can calculate LLR of coded bits by

$$\begin{aligned}
 LLR(c_k) &= \log\left(\frac{\max_{(p,q) \in C_k^{(1)}} \left\{ \alpha_k(p) \cdot \max_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q)} \cap C_k^{(1)}} \left\{ p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k) \right\} \cdot \beta_{k+1}(q) \right\}}{\max_{(p,q) \in C_k^{(0)}} \left\{ \alpha_k(p) \cdot \max_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q)} \cap C_k^{(0)}} \left\{ p(\mathbf{r}^j | \varphi_k) \cdot p(\varphi_k) \right\} \cdot \beta_{k+1}(q) \right\}}\right) \\
 &= \max_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q) \in C_k^{(1)}} \cap C_k^{(1)}} \left\{ \log(\alpha_k(p)) + \frac{1}{2} \left[-\frac{1}{\sigma^2} \|\mathbf{r}^j - \mathbf{H} \cdot \varphi_k\|^2 + \mathbf{b}^T \cdot \mathbf{L}_A \right] + \log(\beta_{k+1}(p)) \right\} \\
 &\quad - \max_{\varphi_k \in \mathcal{L}_{LSD} \cap \Psi^{(p,q) \in C_k^{(0)}} \cap C_k^{(0)}} \left\{ \log(\alpha_k(p)) + \frac{1}{2} \left[-\frac{1}{\sigma^2} \|\mathbf{r}^j - \mathbf{H} \cdot \varphi_k\|^2 + \mathbf{b}^T \cdot \mathbf{L}_A \right] + \log(\beta_{k+1}(p)) \right\} \\
 &\quad (3.9)
 \end{aligned}$$

It is known that the LLR of a bit representing the probability of the bit being to one or zero. Thus, if the absolute value of the LLR is large, we can make decision on the bit. The larger the absolute value of the LLR, the more confident we can have for the decision. On the other hands, the more decisions we make, the less candidates we have to test in the LSD algorithm. As a result, we can enlarge the radius of the LSD without increasing computational complexity. For example, the LSD calculates the distance between the possible symbol vector and the received

vector.

$$\begin{aligned}\mathcal{L}_{LSD} &= \arg \min_{s_j \in \Psi} \|\mathbf{r} - \mathbf{H} \cdot \boldsymbol{\varphi}_k\|^2 \leq r_{LSD}^2 \\ &\approx \arg \min_{s_j \in \Psi} (\boldsymbol{\varphi}_k - \mathbf{y})^H \cdot \mathbf{H}^H \cdot \mathbf{H} \cdot (\boldsymbol{\varphi}_k - \mathbf{y}) \leq r_{LSD}^2\end{aligned}\quad (3.10)$$

with the Cholesky factorization, the distance can be calculated as

$$\begin{aligned}\arg \min_{\boldsymbol{\varphi}_k \in \Psi} (\boldsymbol{\varphi}_k - \mathbf{y})^H \cdot \mathbf{U}^H \cdot \mathbf{U} \cdot (\boldsymbol{\varphi}_k - \mathbf{y}) &\leq r_{LSD}^2 \\ \Rightarrow \sum_{i=1}^{N_T} \left(u_{i,i}^2 |\varphi_{j,i} - y_i|^2 + \sum_{j=i+1}^{N_T} u_{i,j}^2 |\varphi_{j,j} - y_j|^2 \right) &\leq r_{LSD}^2\end{aligned}\quad (3.11)$$

This will form a tree structure and the distance can be calculated level-by-level. Now, we can rank LLRs and select corresponding top C_{APP} bits for detection, where $1 \leq C_{APP} \leq N_{bpt}$. Using this method, the searching domain will be decreased from $2^{N_{bpt}}$ to $2^{N_{bpt}-C_{APP}}$. Then, the radius of LSD, r_{LSD} , can be increased at the rest iterations without adding too many symbol vectors in the list. At the beginning of a decoding iteration, the LSD detector will find a new \mathcal{L}_{LSD} with its increased r_{LSD} .

3.4 Simulation Results

In this section, we report simulation results to evaluate the performance of the proposed scheme. We consider two different systems with the same throughput, one with the conventional BICM scheme, and the other with the proposed TICM. The signal to noise ratio (SNR) is defined as the average received power per receive antenna divided by the average noise power. We use an IEEE 802.11n proposal released by TGn Sync in July 2005 [15] as our basic simulation platform. An optional mode with $K=56$ (the channelization bandwidth 20 MHz), and $N_T=4$. The number of receive antennas is assumed to be the same as that of transmit antennas.

The first system is built based on the IEEE802.11n proposal except for that the original CC encoder is replaced by the turbo encoder shown in Fig. 3-2. The two coded bit streams from two component encoders are interleaved and mapped to QAM symbols alternately. The code rate of the turbo encoder is $1/2$ and $K_{RSC}=3$. We use the LSD detector and MIMO soft-bit demapper outlined in 2.7.2 to calculate LLRs of coded bits. The BCJR algorithm is used to exchange the extrinsic information between two component encoders during the iterative decoding process. We can view this as an inner iteration, since we can conduct turbo equalization and yield an outer iteration. In the outer iteration, the extrinsic information generated at the decoder is fed back to the MIMO demapper to update soft information of coded bits. Since we have two feedback loops, there are many ways to conduct iterations. For simplicity, we let a complete cycle of iteration consist of one for turbo decoding, and one for turbo equalization. We call the system BICM-T. The block diagram of the MIMO-OFDM transceiver with BICM-T is shown in Fig. 3-5 and Fig. 3-6.

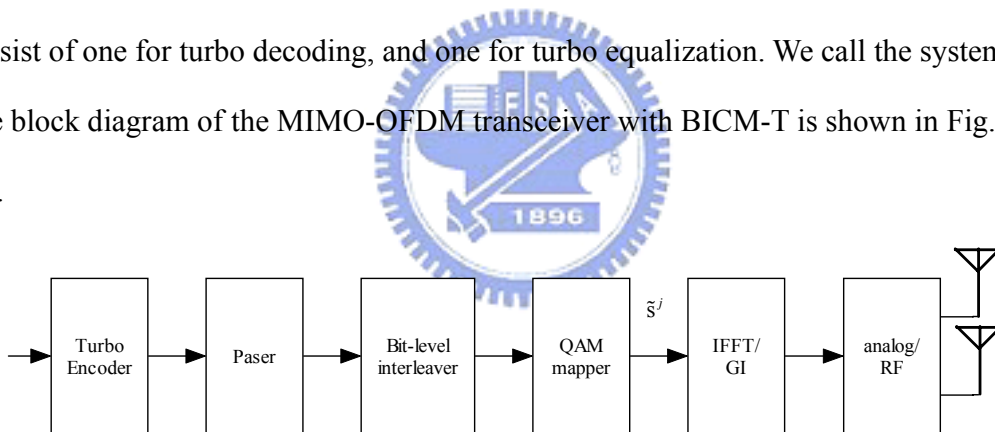


Fig. 3-5: The block diagram of the MIMO-OFDM transmitter for BICM-T

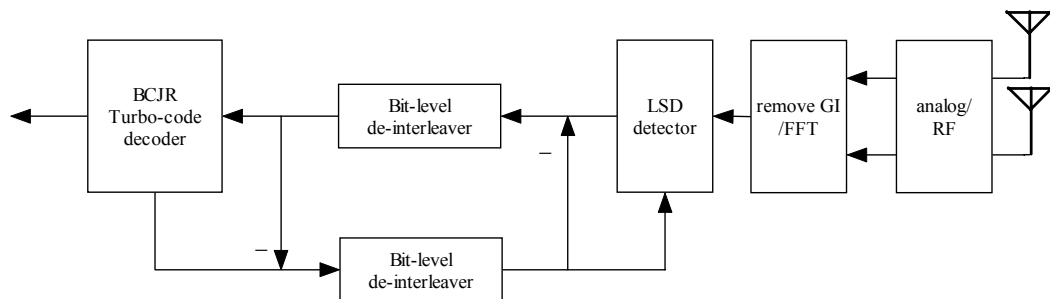


Fig. 3-6: The block diagram of the MIMO-OFDM receiver using the LSD for BICM-T

The second system is also built based on the IEEE802.11n proposal. In addition to that the original CC encoder is replaced by the turbo encoder (shown in Fig. 3-2), but also the bit-interleaving scheme is also replaced by the tone-interleaving scheme proposed in Chapter 3. Then, the tone-level BCJR method is used to exchange extrinsic information between two component decoders, and calculate information bits probability. Here, C_{APP} is set to be 2 (and 4) at the third (fifth) iteration, and enlarge the radius of the LSD twice. As before, we call the system TICM-T.

We choose the channel-B (NLOS) with distance 6m, and the channel-D (NLOS) as our channel models, and a 4x4 system with 16-QAM transmission for comparison. Assume that the frequency offset and timing offset are perfectly compensated at the receiver. The PPDU length is set as 1000bytes; so there are 8000 information bits per frame. The preamble format for TICM-T and BICM-T is assumed to be the same as the original system. The standard per-tone channel estimation method (no smoothing) is used to estimate the MIMO channel. If the number of symbol vectors in \mathcal{L}_{LSD} is less than a threshold (<300), the radius of the sphere is doubled ensuring that a proper iterative decoding.

Fig. 3-7 and Fig. 3-8 show simulation results for these two systems; one for channel-B and the other for channel-D. Note that we only use the estimated channel in simulations. As we can see, TICM-T outperforms BICM-T. If we let the target FER is 0.1, from Fig. 3-7, we can observe that TICM-T outperforms BICM-turbo about 1 dB in the first iteration. In the second and third iterations, TICM-T outperforms BICM-T about 2.5 dB. Fig. 3-8 shows the results for channel-D. The first iteration gap between TICM-T and BICM-T is still about 1dB, also. The second and third iteration gap is about 3 dB, slightly higher than that in channel-B. From the results, we can conclude that the TICM-T scheme is significantly better than the BICM-T scheme.

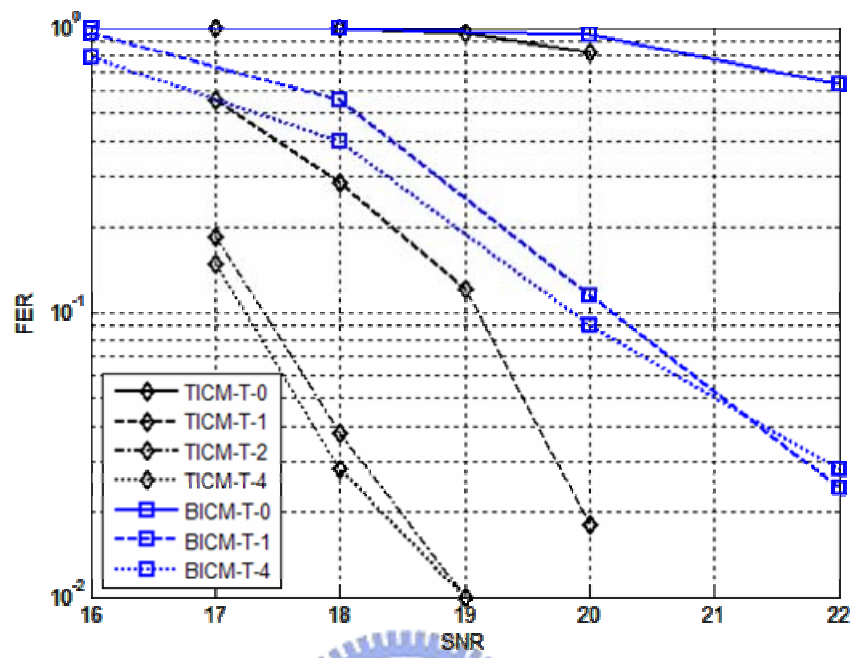


Fig. 3-7: Performance comparison of BICM-T, and TICM-T (20MHz, 4x4 16-QAM, channel B, estimated-channel)

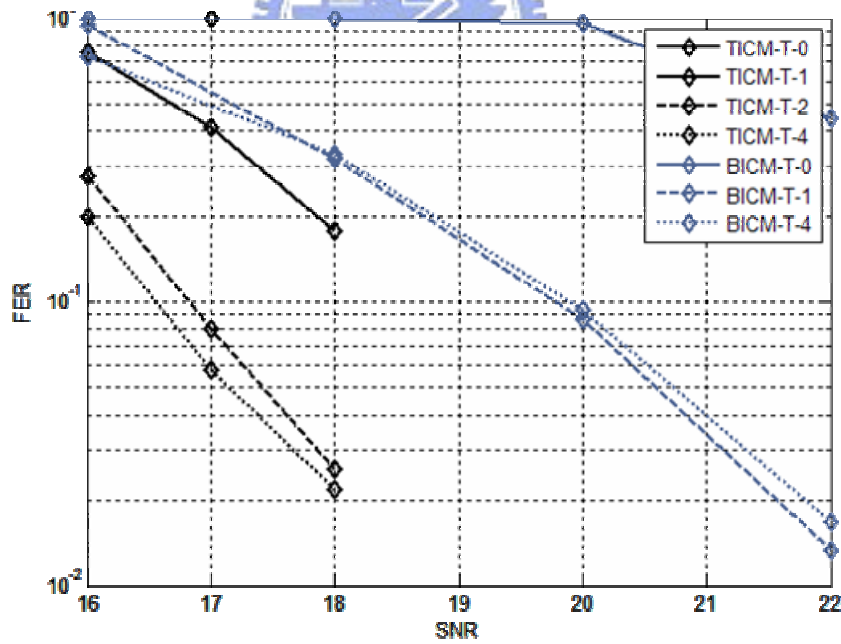


Fig. 3-8: Performance comparison of BICM-T, and TICM-T (20MHz, 4x4 16-QAM, channel B, estimated-channel)

Chapter 4 Conclusions

BICM is a well-known scheme in digital communication. It is widely adopted in many real-world systems. For SISO systems, the calculation of soft-bit information is simple and straightforward. For MIMO systems, calculation of MIMO soft-bit information often requires a high computational complexity. An alternative is to apply an equalizer and decouple the MIMO channels to multiple SISO channels, and then use a SISO soft-demapper for each channel. Combining with the decoding algorithm, this approach provides a simple and straightforward detection scheme. However, this approach is not optimal, and its performance is not satisfactory in many cases. In the first part of this thesis, we investigate the MIMO soft-demapping algorithm and suggest to use the LSD algorithm to reduce the required computational complexity. To further improve the performance, we then study the turbo equalization system. With the IEEE802.11n system, we show that the turbo equalization algorithm can greatly enhance the detection performance.

Recently, a TICM scheme was proposed to enhance the performance of the BICM scheme. Unfortunately, due to the lack of a bit-level interleaver, the performance improvement is not always possible. In the second part of the thesis, we propose to apply the turbo code in the TICM system. Since there is an interleaver between two component decoders, the bursty error problem can then be reduced. Simulations with IEEE802.11n system show that the TICM can significantly outperform the BICM-T scheme. The improvement can be as large as 3 dB.

Reference

- [1]. C. Douillard et al., "Iterative correction of intersymbol interference: Turbo-equalization", ETT, Vol.6, No. 5, pp. 507–511, Sep.-Oct. 1995.
- [2]. G. Bauch, H. Khorram and J. Hagenauer, "Iterative equalization and decoding pp. 307–312, Bonn, Germany, Sep.-Oct. 1997.
- [3]. L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", IEEE Trans. on Inform. Theory, Vol. 20, pp. 284–287, Mar. 1974.
- [4]. B. R. Salzberg, "Performance of an efficient parallel data transmission system," *IEEE Trans. on Commun.*, vol. COM-15, pp. 805-811, Dec. 1967.
- [5]. Bertrand M. Hochwald and Stephan ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Commun.*, vol. 51, no. 3, pp. 389-399, Mar. 2003.
- [6]. E. Zehavi. "8-PSK trellis codes for a rayleigh channel," *IEEE Trans. on Commun.*, vol. 40, pp.873-884, May 1992.
- [7]. L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287, Mar. 1974.
- [8]. F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2," *IEEE Int. Conf. on Commun.*, vol. 2, pp. 664-668, 2002.
- [9]. A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, IT-13, pp. 260-269, 1967.

- [10].U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Computation*, vol. 44, pp. 463-471, Apr. 1985.
- [11].S. C. Shen and W. R. Wu, "A tone-interleaved coded modulation scheme for MIMO OFDM systems," *ISPACS*, Dec 2006.
- [12].B. R. Salzberg, "Performance of an efficient parallel data transmission system," *IEEE Trans. on Commun.*, vol. COM-15, pp. 805-811, Dec. 1967.
- [13].C. Berrou and A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes(1)," *IEEE ICC'93 in Proc.*, pp. 1261-1271, Oct. 1996.
- [14].S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Parallel concatenated trellis coded modulation," *IEEE ICC'96 in Proc.*, pp. 974-978, 1996.
- [15].V. Erceg et al., IEEE 802.11 document 03/940r4 "TGn channel models," May 2004.
- [16].W Zhao, GB Giannakis, "Sphere decoding algorithms with improved radius search", *IEEE Trans. on Commun.*, vol. 53, no. 7, pp. 1104- 1109, Jul. 2005.