UNBALANCED POWER ALLOCATION FOR TURBO CODED PSAM AND PILOT ASSISTED SOVA DECODER

by

Yi Zheng

B.A.Sc, Wuhan University of Science and Technology, 1997

M.A.Sc, Huazhong University of Science and Technology, 2000

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

In the School
of
Engineering Science

SIMON FRASER UNIVERSITY

April, 2004

Approval

Name: Yi Zheng

Degree: Master of Applied Science

Title of Thesis: Unbalanced Power Allocation for Turbo Coded PSAM and Pilot Assisted SOVA Decoder

Examining Committee:

Chair: Dr. John Bird
Professor

Dr. Dong In Kim
Senior Supervisor
Associate Professor

Dr. James Cavers
Supervisor
Professor

Dr. Stephen Hardy
Examiner
Professor
School of Engineering Science
Simon Fraser University

Date Approved: _____

SIMON FRASER UNIVERSITY

Partial Copyright Licence

ABSTRACT

In communication systems, channel coding is widely employed in order to have reliable data transmissions. Turbo codes were first introduced by C.Berrou in 1993 [1] as a powerful channel coding technique, and were shown to achieve performance near Shannon capacity over an additive white Gaussian noise (AWGN) channel. In this thesis, two schemes related to turbo codes are proposed, and are shown to have superior performance over previous work.

In [1], turbo codes were applied to Pilot Symbol Assisted Modulation for iterative channel estimation and decoding over correlated flat-fading channels. The first scheme of this thesis extends the work in [1] by allocating different power values to the pilot symbols and turbo coded symbols. When compared with previous work in this field, simulation results show a significant performance gain.

The second scheme is based on the Soft Output Viterbi Algorithm. Pilot symbols are introduced to detect errors as well as to correct errors. Simulation results show a coding gain of 0.43 dB when using the pilot assisted turbo SOVA decoder over the conventional SOVA decoder

DEDICATION

To my family!

ACKNOWLEDGEMENTS

I would like to express my gratitude to my previous senior supervisor Dr. Jacques Vaisey who, unfortunately, passed away last October. He was an excellent advisor and a great teacher, who devoted much of his time and energy to his students and work.

I also give thanks greatly to Dr. Dong In Kim and Dr. James Cavers! They gave me very important guidance and kind help on my research and thesis, as my senior supervisor and supervisor.

In addition, special thanks go to Dr. Stephen Hardy and Dr. John Bird for their comments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
MAP	Maximum A Posteriori
PCCC	Parallel Concatenated Convolutional Codes
RSC	Recursive Systematic Convolutional
SOVA	Soft Output Viterbi Algorithm
FSK	Frequency-Shift-Keying
DPSK	Differential Phase-Shift-Keying
BPSK	Binary Phase-Shift-Keying
MSDD	Multiple Symbol Differential Detection
PSAM	Pilot Symbol Assisted Modulation
CDMA	Code Division Multiple Access
CSI	Channel Side Information
NSI	No Channel Side Information
OFDM	Orthogonal Frequency Division Multiplexing
LLR	Log likelihood ratio.

CHAPTER ONE: INTRODUCTION

1.1 INTRODUCTION

In wireless communication systems, the physical channel is generally distorted and affected by noise. To help overcome this, error-correcting codes are widely used in order to provide reliable data transmission. An error-correcting code is a common signal processing technique which introduces redundancy on the encoder side, such that the decoder can later reconstruct the transmitted data through the redundant information. Figure 1.1 shows a simplified model of a coded system.

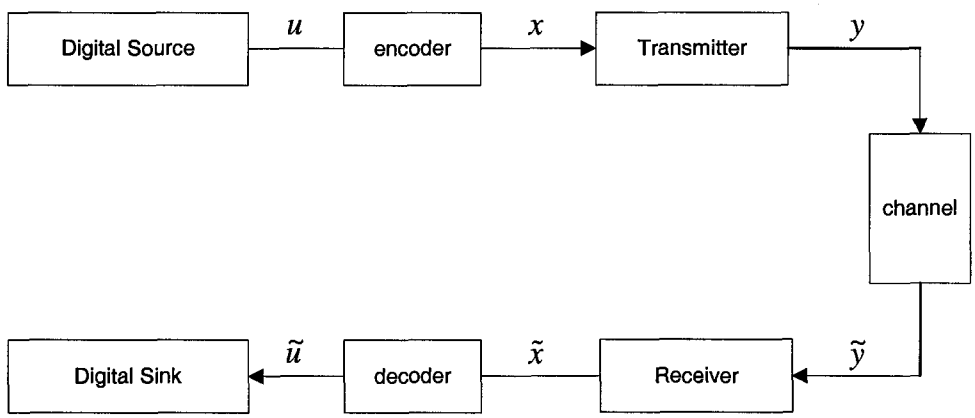

Figure 1.1 Simplified model of a coded communication system

The assumption is made that the information source is digital; if it is not, the analog information can be converted to the digital format. The digital output u is sent to the encoder in order to generate the coded symbol stream x. Next, x is modulated and transmitted over the physical channel. The output of the channel $\tilde{y}$ is then processed by

the receiver. Next, the decoder will make an estimate, $\tilde{u}$, of the original information u based on the receiver output $\tilde{x}$.

The major objective of a coding technique is to control the decision errors so that reliable communications can be obtained; i.e., so that $\tilde{u}$ is as close to u as possible. In information theory [2], it was shown that arbitrarily small error rates can be achieved provided that the rate of transmission is less than the channel capacity. It is well known that by increasing the block length, the channel capacity is approachable through the use of random codes [3]. However, the complexity of the optimal decoder becomes unmanageable as the block length becomes large. Turbo codes are the most exciting and potentially important development in coding theory in recent years. These powerful codes are capable of achieving near Shannon capacity performance [1] on additive white Gaussian noise (AWGN) channels, at a bit error rate (BER) of 10^{-5} for sufficient block length.

1.2 BACKGROUND AND RELATED WORK

Many papers have discussed the decoding algorithms [4], [5] for turbo codes. The optimal decoder uses the Maximum A Posteriori (MAP) decoding algorithm for convolutional codes, proposed in 1974 by Bahl *et al* [4]. Initially, this algorithm received little attention due to its increased complexity when compared with alternative convolutional decoders, for only a slightly reduced BER. However, the MAP decoding algorithm has recently enjoyed renewed and greatly increased attention, due to the fact that it was used in the iterative decoder developed by Berrou *et al.* in 1993 [1]. Turbo codes were first introduced in the form of Parallel Concatenated Convolutional Codes

(PCCC). For PCCC, the encoder consists of two Recursive Systematic Convolutional (RSC) codes concatenated in parallel through an interleaver. The decoder uses two component decoders based on the MAP algorithm, with soft input and soft output. Since the introduction of PCCCs, researchers around the world have investigated the design and performance of Turbo codes, as well as the application of the iterative principle to other areas such as coded modulation, equalization and channel estimation. Interleaver patterns with high performance have been evaluated in [6]-[9]. In iterative decoding, two classes of decoding algorithms have been proposed; one class consists of the MAP algorithm and its simplified versions, such as the log-MAP algorithm and max-log-MAP algorithm, while the other class consists of the Soft Output Viterbi Algorithm (SOVA) [5] and its modified versions [9]-[12].

In a wireless environment, the channel is often non-stationary due to fading; the fading is a consequence of the physical nature of the channel, as the channel gain is a random process. The fading effect imposes a severe penalty on data transmission, and as such various channel coding techniques have been investigated to mitigate the fading effect.

Turbo codes have been used as channel codes for correlated flat fading channels with noncoherent, as well as coherent, schemes. Noncoherent schemes include turbo coded frequency-shift-keying (FSK) and turbo coded differential phase-shift-keying (DPSK) which was proposed by Hall and Wilson [13]. According to [14], DPSK needs no estimation of the channel gain, however noncoherent demodulation leads to a significant performance loss of 6.1dB at BER of 10^{-5} when compared with coherent binary phase-shift-keying (BPSK). Another noncoherent scheme uses multiple symbol differential detection (MSDD), which is a maximum likelihood detection scheme to detect a DPSK

signal [15][16]. Unfortunately, the computational complexity increases exponentially as the window size of MSDD increases.

For coherent demodulation, channel gain estimation is necessary. The most efficient channel gain estimation method for Rayleigh and Rician fading channels is Pilot Symbol Assisted Modulation (PSAM), proposed by Dr. Cavers in 1991 [17]. A coherent scheme using iterative channel estimation has also been studied in [14] and [18] for the flat fading channel, where the coherent scheme was shown to be superior in power efficiency when compared with noncoherent schemes. Iterative channel estimation has also been studied for frequency-selective fading channels in [19]-[21], and channel estimation and multi-user detection in Code Division Multiple Access (CDMA) [22]-[24].

1.3 MOTIVATIONS AND THESIS OVERVIEW

This thesis consists of two parts, both of which are related to turbo codes. In the first part of this thesis, an algorithm based on conventional SOVA [5] for turbo codes is proposed and studied. The conventional SOVA algorithm is a lower complexity decoder which uses *a priori* information for iterative decoding. However, the coding gain achieved by SOVA is generally 0.7 dB less than when using the MAP algorithm. This difference makes the issue of improving the SOVA performance, without seriously affecting the algorithmic complexity, an important problem. An alternative version of the SOVA decoder has been developed based on the Viterbi Algorithm [2]. This technique has been shown to be equivalent in performance to the Max-Log-MAP decoder at high SNR; however, it is much more difficult to implement than the conventional SOVA approach. In [26], a modified version of conventional SOVA was proposed, and in [27] was shown

to be superior to conventional SOVA. However, it has the disadvantage of increased complexity after taking the competitor path metric into account. In [28], a bi-directional SOVA decoding algorithm was proposed which employs both forward and backward SOVA decoding in each decoding stage. The performance of this method is close to the Max-Log-MAP algorithm at high SNR, but the computational complexity is doubled. In addition, this method leads to significant delays in decoding.

In this thesis, a pilot assisted turbo encoding and decoding scheme is proposed based on the conventional SOVA, that offers improved performance and negligible increase in complexity. On the encoder side, pilot symbols are periodically embedded into the bit stream. On the decoder side, two different decoding schemes are proposed: one uses pilots to detect possible errors and scales the extrinsic information accordingly, the other modifies the path metric to improve decoding performance. These two schemes can be combined to achieve synergistic improvements.

In the second part of this thesis, a scheme is proposed to allocate more power to pilot symbols and less power to data symbols, with the overall power expense unchanged, in order to improve the performance of the iterative estimation and decoding.

This thesis is organized as follows. Chapter 1 provided introductory information, and the objectives of this thesis. In Chapter 2, turbo codes and the performance upper bound are discussed in detail. In Chapter 3, first Rayleigh fading channel characteristics and Pilot Symbol Assisted Modulation are discussed, then the iterative channel estimation and decoding scheme is described. In Chapter 4, a Pilot Assisted SOVA decoder for turbo codes is studied. In Chapter 5, the unbalanced power allocation for iterative channel

estimation is studied. Finally, conclusions and future directions are given, with simulation results shown in the Appendix.

CHAPTER TWO:
TURBO CODES

In this chapter, the turbo codes scheme is introduced first, followed by a detailed description of RSC codes, MAP decoders and SOVA decoders. Finally, the performance bound for turbo codes is discussed.

2.1 PRINCIPLES OF TURBO CODES

According to Shannon's theorem, assuming the data rate is less than channel capacity, a random long code requires the minimum signal-to-noise ratio (SNR) to make the probability of error arbitrarily low. However, as block length increases, the optimal decoder becomes unmanageable. Before turbo codes were introduced, highly structured codes, such as block codes and convolutional codes, were used. However, these codes suffer in the low SNR range due to the fact that they have a low degree of randomness, as they are highly structured. That results in an unacceptable performance with respect to Shannon's Limit. Turbo codes attempt to achieve a large degree of randomness by using combinations of multiple interleaved structured codes at the encoder, and using an iterative decoding algorithm at the decoder in order to retrieve the information. In doing so, the turbo codes can perform very close to what is suggested by the Shannon limit.

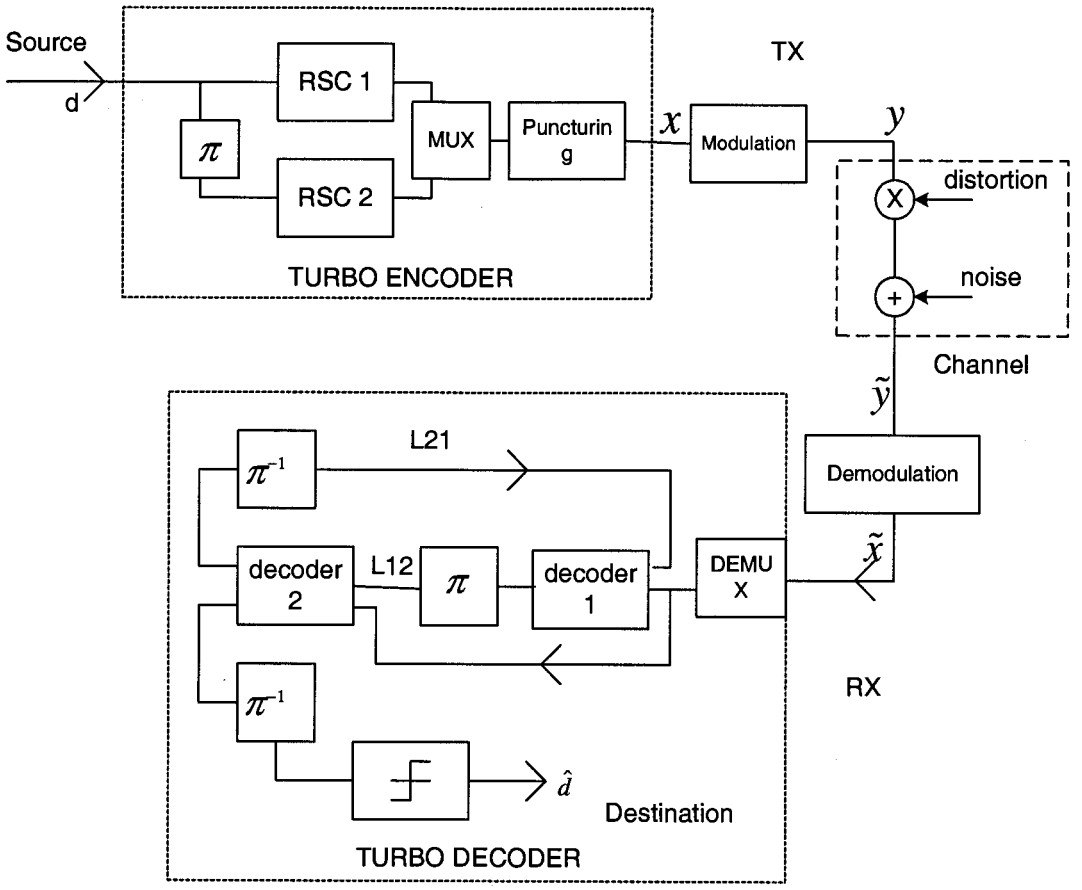

Figure 2.1 Turbo codec scheme

Figure 2.1 shows the diagram of a typical turbo-coded system. The system consists of the turbo encoder on the transmitter side, the channel and the turbo decoder on the receiver side. Denote d as the binary data to be transmitted, x as the turbo encoder output and y as the modulation output. The encoder introduces redundancy, which lowers the bandwidth efficiency but makes the system more robust to channel distortion and noise. The blocks labelled by π and π^{-1} are interleavers and de-interleavers, respectively. To generate the coded symbols, the first RSC encoder is fed with d and the second RSC encoder is fed with the interleaved version of d. The input d combined with the parity bits

from RSC1 and RSC2 are then multiplexed to form a bit stream. Recall that the aim of the interleaver is to make this bit stream more random. After puncturing, the bit stream is modulated and sent out over the channel, which introduces distortion as well as noise. The received signal is demodulated into $\tilde{x}$, after which $\tilde{x}$ is demutiplexed into two code streams. Corresponding to the turbo encoder structure, the turbo decoder consists of two component decoders joined by the corresponding interleaver and de-interleaver. The input of each component decoder is the channel output and *a priori* information from last decoding stage. The output of each component decoder are the log likelihood ratios (LLRs) of the code symbols. The output can be separated into three parts: *a priori* information, channel output and extrinsic information. The extrinsic information is fed to the partner component decoder as *a priori* information. The decoding process of the turbo decoder is as follows. Each decoding iteration consists of two decoding stages which correspond to the two component decoders. In each decoding stage, the component decoder uses channel output and *a priori* information from the last decoding stage as input, and outputs the LLRs of the code symbols and the extrinsic information. The extrinsic information is used as *a priori* information for the next decoding stage. In the first decoding stage of the first decoding iteration, the *a priori* information of each code symbol is initialized to zero; at the end of the last decoding stage of the last decoding iteration, the LLR of each code symbol is used to make the decision on each transmitted symbol.

2.2 TURBO ENCODER

The fundamental turbo code encoder consists of two identical RSC encoders with parallel concatenation [1]. The RSC encoder can be obtained from the non-recursive, nonsystematic convolutional encoder through matrix transformation, and by feeding back one of its encoded outputs to its input. An RSC encoder is typically rate 1/2 and is termed a component encoder. The two component encoders, RSC1 and RSC2, are connected by an interleaver. Only one of the systematic outputs from the two component encoders is used, as the systematic output from the other component encoder is just a permuted version of the chosen systematic output. Figure 2.2 shows the fundamental turbo codes rate 1/3 encoder. RSC1 outputs the systematic output x_1 and parity output x_2, while RSC2 discards its systematic output and only outputs the parity output x_3.

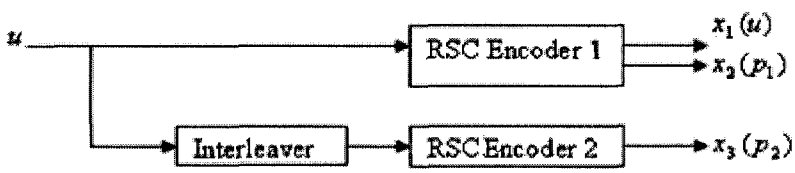

Figure 2.2 Turbo codes encoder

Figure 2.3 shows an example of an RSC encoder with generators G(D) that are [1, 05/07] in octal, and have memory size equal to 2. At time k, u_k is the input bit to the encoder, and $x_{2,k}, x_{3,k}$ are parity code symbols.

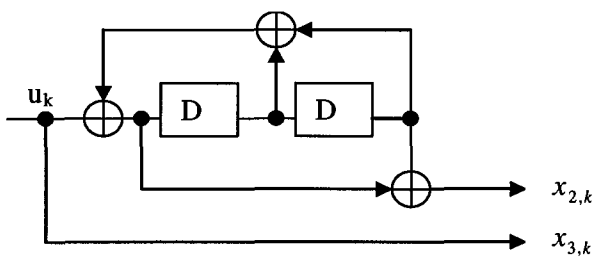

Figure 2.3 An RSC encoder with generators [1, 05/07], memory size 2

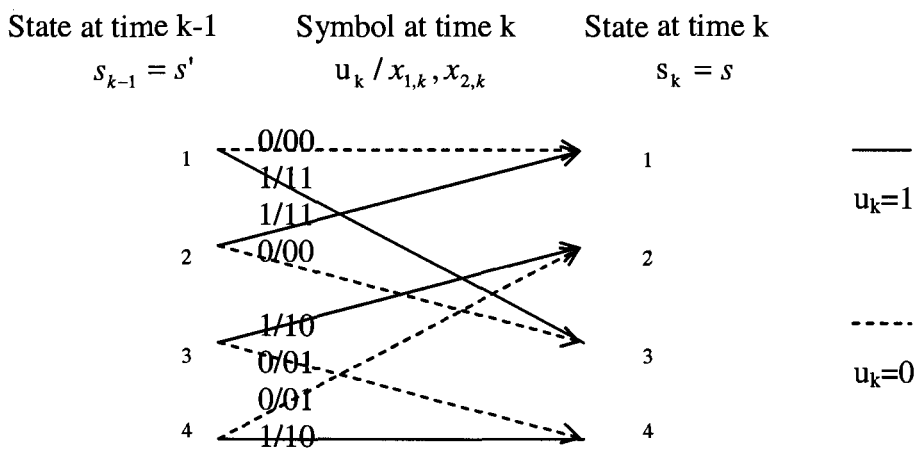

Figure 2.4 A trellis section of an RSC encoder with generator [1, 05/07], memory size 2

The trellis section generated by the RSC encoder with generator [1, 05/07] is illustrated in Figure 2.4. The trellis section is characterized by the four parameters u_k, x_k, s' and s, which are the input, output, start state and end state of the trellis section, respectively. In a systematic convolutional code, given any pair (s', u), (s, u) or (s', x), the trellis section is uniquely identified. In an RSC encoder, i$\in$(1, 2^M) is the trellis state index, S= {s_i, i$\in$ (1, 2^M)}, where M is the constituent encoder memory size, subject to M=K-1, where K is the constraint length. The trellis state at time k-1, or start state of s_k with input u_k, is given by $s'=s_{k-1}$ where $s' \in$ S. Similarly, the trellis state at time k, or the end state s_k with input u_k is given by $s=s_k$ where $s \in$ S.

2.3 TURBO CODES DECODER

As illustrated in Figure 2.5, turbo codes component decoders can be classified into two classes; one class is based on the MAP algorithm and the other class is based on the Viterbi Algorithm. The symbol-by-symbol MAP algorithm [4] was formally presented by Bahl in 1974 as an alternative to the Viterbi algorithm for decoding convolutional codes. In principle, the MAP algorithm can calculate accurate estimates of the *a posteriori* probability of each code symbol. However, the MAP algorithm suffers from two severe practical problems. First, it is computationally intensive, requiring 6×2^{M_c} multiplications and additions per estimated bit, where M_c is the number of states. Second, it is sensitive to round-off errors. To alleviate these two problems, the Log-MAP and Max-Log-MAP algorithms have been developed. Both of these algorithms perform multiplications as additions in the log-domain, the difference lies in how they compute addition in the log-domain. The Max-Log-MAP approximates addition in the log-domain as a maximization operation, while the Log-MAP computes the addition in the log-domain as a maximization followed by a correction term. The Viterbi algorithm was introduced in 1967 as a computationally efficient method for performing maximum likelihood decoding of convolutional codes. Although the Viterbi algorithm [5] accepts soft-inputs in the form of *a priori* information, it does not produce soft-outputs in terms of *a posteriori* probabilities, and is therefore not suitable for turbo decoding. In 1989, Hagebauer and Hoeher proposed a modification to the Viterbi algorithm which produces the a posteriori probabilities of the state transitions, and is known as the Soft Output Viterbi Algorithm (SOVA) [5]. Various improvements based on SOVA have been

proposed [10]-[12]. The MAP and SOVA algorithms will be discussed in details in next sections.

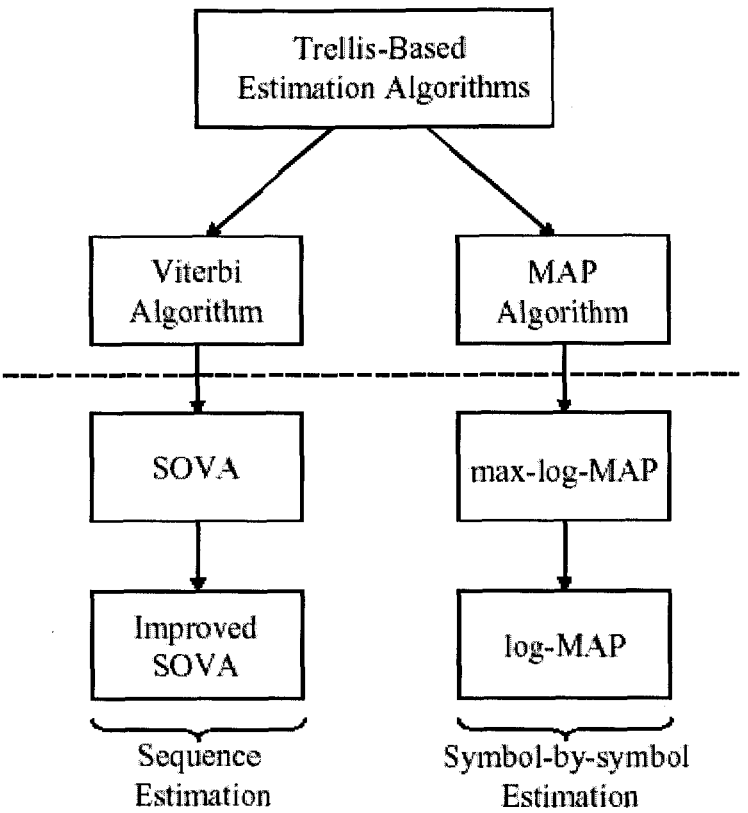

Figure 2.5 Overview of Trellis based estimation algorithms

2.3.1 MAP DECODER

The following review of the symbol-by-symbol MAP decoding algorithm is based on the following definitions:

- N represents the frame size of transmitted symbols.
- k∈ {1, N} is the time index.
- $x_k = \left(x_{1,k}, x_{2,k}, x_{3,k}, \ldots x_{M,k}\right) \in (-1,1)$ is the modulated symbol at time k.
- $y_k = \left(y_{1,k}, y_{2,k}, y_{3,k}, \ldots y_{M,k}\right)$ is received symbol corresponding to x_k.

- $y_1^N = (y_1, y_2, \ldots y_N)$ is one frame of received symbols.

According to [2], the symbol-by-symbol MAP decoding algorithm at the bit level is derived as follows.

The log-likelihood ratio $L(u_k)$ of a binary random variable u_k is defined a

$$L(u_k) \triangleq \log\left(\frac{P(u_k = 1 | y_1^N)}{P(u_k = 0 | y_1^N)}\right). \tag{2-1}$$

The decision of u_k is made based on the sign of $L(u_k)$, i.e.,

$$\tilde{u}_k = sign[L(u_k)]. \tag{2-2}$$

According to probability theory and the trellis section generated by RSC encoder

$$\begin{aligned} L(u_k) &= \log\left(\frac{P(u_k = 1 | y_1^N)}{P(u_k = 0 | y_1^N)}\right) \\ &= \log\left[\frac{\sum_{u^+, s' \to s} P(s_{k-1} = s' \to s_k = s | y_1^N)}{\sum_{u^-, s' \to s} P(s_{k-1} = s' \to s_k = s | y_1^N)}\right] \\ &= \log\left[\frac{\sum_{u^+, s' \to s} P(s_{k-1} = s' \to s_k = s, y_1^N)/P(y_1^N)}{\sum_{u^-, s' \to s} P(s_{k-1} = s' \to s_k = s, y_1^N)/P(y_1^N)}\right] \\ &= \log\left[\frac{\sum_{u+, s' \to s} \alpha_{k-1}(s')\beta_k(s)\gamma_k(s', s)/P(y_1^N)}{\sum_{u-, s' \to s} \alpha_{k-1}(s')\beta_k(s)\gamma_k(s', s)/P(y_1^N)}\right], \end{aligned}$$

where $\alpha_k(s) \triangleq P(s_k = s | y_1^k)$, $\beta_{k-1}(s') \triangleq P(y_k^N | s_{k-1} = s')$, and

$$\begin{aligned} \gamma_k(s', s) &\triangleq P(s_k = s, y_k | s_{k-1} = s') \\ &= P(s_k = s | s_{k-1} = s') P(y_k | s_{k-1} = s' \to s_k = s) \\ &= P(u_k) P(y_k | x_k) \end{aligned}$$

Note that $P(y_k|x_k)=\frac{1}{\sqrt{\pi N_0/E_s}}\exp\left\{\frac{-E_s}{N_0}\sum_{q=2}^{M}\left[y_{q,k}-(2x_{q,k}-1)\right]^2\right\}$ is the branch metric, and $P(u_k)$ is a priori information from last decoding stage.

The forward path metrics $\alpha_k(s)$ can be recursively computed as

$$\alpha_k(s)=\sum_{s'}\alpha_{k-1}(s')\gamma_k(s',s), \tag{2-4}$$

and the backward path metric $\beta_k(s)$ can be recursively computed as

$$\beta_{k-1}(s')=\sum_{s/s_{k-1}=s'}\beta_k(s)\gamma_k(s',s), \tag{2-5}$$

where $\sum_{s'}(\,)$ is the summation over all possible items with starting state $s_{k-1}=s'$, and $\sum_{s'/s_k=s}(\,)$ is the summation over all possible items with starting states $s_{k-1}=s'$ and ending state $s=s_k$. The initial condition is $\alpha_0(s)=\begin{cases}1 & \text{if s = 1}\\ 0 & \textit{otherwise}\end{cases}$ which means that the encoder Trellis states always start at state 1. The $\beta_k(s)$'s initial condition is $\beta_N=\begin{cases}1 & \text{if s = 1}\\ 0 & \text{otherwise}\end{cases}$ which means that the encoder is expected to end in state 1.

Finally the LLR is expressed as

$$L(u_k)=\log\left[\frac{\sum_{u+,s'\to s}\alpha_{k-1}(s')\beta_k(s)\gamma_k(s',s)/P(y_1^N)}{\sum_{u-,s'\to s}\alpha_{k-1}(s')\beta_k(s)\gamma_k(s',s)/P(y_1^N)}\right]. \tag{2-6}$$

2.3.2 SOVA ALGORITHM

The SOVA algorithm proceeds by first executing the standard soft decision Viterbi algorithm, followed by a second stage of LLR calculation. The soft output Viterbi algorithm, along with a reliability updating procedure, can be implemented as follows:

- At time index $k=0$, initialize $M_0^{(0)}=0$ for the zero state in the trellis diagram, and $M_0^{(m)}=-\infty$ for all other states
- Increase time index k-1 to k, and compute the path metrics for state m as

$$M_k^{(m)} = M_{k-1}^{(m')} + u_k L_c y_{1,k} + \sum_{j=2}^{N} x_{j,k}^{(m)} L_c y_{j,k} + u_k L(u_k), \qquad (2\text{-}7)$$

where $M_k^{(m)}$ is the accumulated path at time k for state m, u_k is the systematic bit at time k, $x_{j,k}^{(m)}$ is the j-th bit of M bits at time k on the assumed branch, $y_{t,j}^{(m)}$ is the received value from the channel corresponding to $x_{j,k}^{(m)}$, $L_c = 4\frac{E_b}{N_0}$ is the channel reliability value, and $L(u_k)$ is the *a priori* reliability value at time k. Note that for the first decoding stage, $L(u_k)$ is set to zero.

- Find $\max_m M_k^{(m)}$ for each state; denote the path with the largest path metric as the survivor path, and the path with the smallest path metric as the competitor path. Let $M_k^{(m_1)}$ stands for the survivor path metric and $M_k^{(m_2)}$
- Compute the initial LLR as $\Delta_k^0 = \frac{1}{2}\left|M_k^{(m_1)} - M_k^{(m_2)}\right|$.

- Denote the reliability along the survivor path for a particular node at time k as Δ_k^{MEM} $(MEM = 0,1,\cdots k)$, MEM is denoted as memorization level to index the reliability updating process. Compare the survivor path and the competing path at each state for time k, and store the MEMs where the estimated binary decision of the two paths differ.
- Update $\Delta_t^{MEM} \approx \min_{i=0,\cdots,MEM} \{\Delta_k^i\}$ for all MEMs from the smallest to largest MEM.
- Repeat from the path metric computation step until reaching the end of the trellis.
- Output the estimated sequence $\hat{u}$ and its LLR as $L(\hat{u}) = \hat{u} \bullet \Delta$, where $\bullet$ stands for element by element multiplication and Δ is the final updated reliability sequence. $L(\hat{u})$ is then passed to the next decoding stage as *a priori* information.
- The extrinsic information to feed to the next decoding stage is

$$L_e(\hat{u}_k) = \beta\left(L(\hat{u}_k) - L_c y - L_{in}(u_k)\right). \tag{2-8}$$

2.4 PERFORMANCE BOUND FOR TURBO CODES

Turbo codes can achieve small BER at low SNRs; however, when the SNR becomes large, it is hard to obtain accurate BER performance through simulations. In addition to performance obtained from simulations, upper bounds on the BERs of turbo codes have been highlighted by the works of S. Benedetto [29] [30], D. Divsalar [30] [31] and S. Riedel [32].

Transfer function bounding techniques [30] have been applied to obtain union bounds on the BER for maximum likelihood decoding of turbo codes, constructed with random interleaver permutations. As it is not tractable to obtain analytic results on the BER for a

particular interleaver, the bounds on the BER have been developed as averages over certain ensembles with random coding properties. According to [31], the performance bound analysis is based on taking the turbo code scheme as the concatenation of multiple "code fragments". In Figure 2.6, one of the code fragments is the input frame $\mathbf{u}^s$, while the other fragments are generated by the two component encoders, $\mathbf{u}^{p1}$ and $\mathbf{u}^{p2}$.

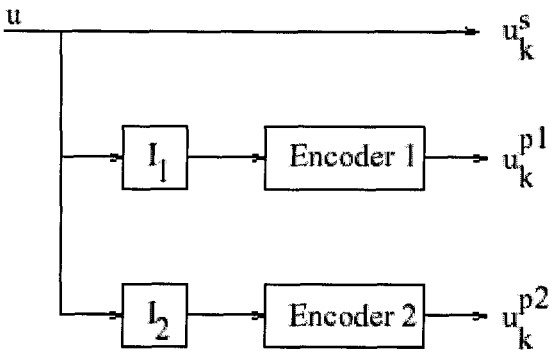

Figure 2.6 Turbo codes diagram

Without loss of generality, consider sending an all zero codeword for performance analysis. Assuming that the interleavers I_1 and I_2 are known, the traditional union upper bound for maximum likelihood decoding of a (n, k) block code over a memoryless channel can be given as:

$$P_W(I_1,I_2) \le \sum_{d=d_{\min}}^{n} A(d|I_1,I_2)P_2(d), \qquad (2\text{-}9)$$

where $P_2(d)$ is the probability of choosing a specific incorrect codeword with weight d and $A(d|I_1,I_2)$ is the number of codewords with Hamming weight d. For a specific interleaver pair (I_1,I_2), $A(d|I_1,I_2)$ is obtained by encoding all input sequences and calculating the number with the same output weight. Consider forming the word error probability averaged over all interleaver pairs:

$$\begin{aligned}\overline{P_w} &= \sum_{I_1}\sum_{I_2} P(I_1,I_2)P_w(I_1,I_2) \\ &= \sum_{I_1}\sum_{I_2} P(I_1,I_2)\sum_{d=d_{\min}}^{n} A(d|I_1,I_2)P_2(d) \\ &= \sum_{d=d_{\min}}^{n} P_2(d)\sum_{I_1}\sum_{I_2} P(I_1,I_2)A(d|I_1,I_2) \\ &= \sum_{d=d_{\min}}^{n} P_2(d)\overline{A(d)}\end{aligned} \tag{2-10}$$

where $P(I_1,I_2)$ is the probability of a specific interleaver pair (I_1,I_2) and $\overline{A(d)}$ is the average weight distribution function $\overline{A(d)} = \sum_{I_1}\sum_{I_2} P(I_1,I_2)A(d|I_1,I_2)$. $A(d|I_1,I_2)$ can be expressed as $A(d|I_1,I_2) = \sum_i \binom{k}{i} P(d|i,I_1,I_2)$. $P(d|i,I_1,I_2)$ is the conditional probability of producing a codeword of weight d, from input weight i and interleaver pair (I_1,I_2). As turbo codes are the concatenation of three code fragments, the total output weight should be the sum of the three codeword weights $d = i + d_1 + d_2$, where i is the input codeword weight, and d_1 and d_2 are the Hamming weights of sequences $\mathbf{u}^{p^1}$ and $\mathbf{u}^{p^2}$ respectively. $A(d|I_1,I_2)$ can be further expressed as

$$A(d|I_1,I_2) = \underbrace{\sum_i \sum_{d_1} \sum_{d_2}}_{d=i+d_1+d_2} \binom{k}{i} P(i,d_1,d_2|i,I_1,I_2), \tag{2-11}$$

where $P(i,d_1,d_2|i,I_1,I_2) = P(i|i)P(d_1|i,I_1)P(d_2|i,I_2)$. (2-12)

Substituting (2-11) and (2-12) into (2-10), the average word error probability is union bounded as:

$$\begin{aligned}\overline{P_w} &\le \sum_{d=d_{\min}}^{n} \overline{A(d)} P_2(d) \\ &= \sum_{d=d_{\min}}^{n} \underbrace{\sum_{i} \sum_{d_1} \sum_{d_2}}_{d=i+d_1+d_2} \binom{k}{i} P(i|i) P(d_1|i,I_1) P(d_2|i,I_2) P_2(d)\end{aligned} \tag{2-13}$$

and the bit error probability is bounded as

$$P_b \le \sum_{d=d_{\min}}^{n} \underbrace{\sum_{i} \sum_{d_1} \sum_{d_2}}_{d=i+d_1+d_2} \frac{i}{k} \binom{k}{i} P(d_1|i,I_1) P(d_2|i,I_2) P_2(d). \tag{2-14}$$

According to [31], the distribution of the parity fragments, d_1 and d_2, may be expressed as

$$P(d_p|i) = \frac{t(k,i,d_p)}{\sum_{d_p} t(k,i,d_p)} = \frac{t(k,i,d_p)}{\binom{k}{i}}, \tag{2-15}$$

where $t(k,i,d_p)$ is defined as the number of trellis paths of length k, input weight i and parity weight d_p, that start and end in the zero state. In [31], it was shown that for an arbitrary block length ($k = l$), $t(l,i,d)$ can be computed recursively by examining the transfer function of the convolutional encoder $T(L,I,D)$. The transfer function is defined as

$$\begin{aligned}T(L,I,D) &= \sum_{l\ge 0} \sum_{i\ge 0} \sum_{d\ge 0} L^l I^i D^d t(l,i,d) \\ &= \left[\left(\mathbf{I} - \mathbf{A}(L,I,D) \right)^{-1} \right]_{(0,0)}\end{aligned}. \tag{2-16}$$

In (2-16), $\mathbf{I}$ stands for identity matrix and $\mathbf{A}(L,I,D)$ is the state transition matrix for the convolutional encoder. For an RSC with memory γ, both $\mathbf{I}$ and $\mathbf{A}(L,I,D)$ are $(2^\gamma \times 2^\gamma)$

full rank matrices. Once $T(L,I,D)$ is determined, a recursion is formed to compute $t(l,i,d)$, which is in turn necessary to compute $P(d|i)$ for the average upper bound.

For the 4-state RSC with generator (7/5), the BER is calculated as follows. Consider bounding turbo codes with input frame length k; in order to compute the average upper bound, $t_{7/5}(k,i,d)$ must first be formed. Following the procedures above, in [31] it was shown that the transfer function can be expressed as

$$T_{7/5}(L,I,D) \approx \frac{1-LD-L^2D+L^3D^2-L^3I^2}{1-L(1+D)+L^3\left(D+D^2-I^2-I^2D^3\right)-L^4\left(D^2-I^2-I^2D^4+I^4D^2\right)}. \qquad (2\text{-}17)$$

By multiplying both sides of (2-17) by the right-hand side denominator, the recursive expression for $t_{7/5}(l,i,d)$ for $l \geq 0, i \geq 0, d \geq 0$ is as follows:

$$\begin{aligned} t_{7/5}(l,i,d) = {} & t_{7/5}(l-1,i,d-1)+t_{7/5}(l-1,i,d)+t_{7/5}(l-3,i-2,d-3) \\ & +t_{7/5}(l-3,i-2,d)-t_{7/5}(l-3,i,d-2)-t_{7/5}(l-3,i,d-3) \\ & +t_{7/5}(l-4,i-4,d-2)+t_{7/5}(l-4,i-2,d-4)-t_{7/5}(l-4,i-2,d) \\ & +t_{7/5}(l-4,i,d-2)+\delta(l,i,d)-\delta(l-1,i,d-1)-\delta(l-2,i,d-1) \end{aligned} \qquad (2\text{-}18)$$

where $\delta(l,i,d)=1$ if $l=i=d=0$ and 0 otherwise, with the constraint that $t_{7/5}(l,i,d)=0$ for any negative index. Next, the probability of producing a parity sequence of weight d_p given a randomly selected input sequence with Hamming distance i, is computed as

$$P_{7/5}(d_p|i) = \frac{t_{7/5}(k,i,d_p)}{\sum_d t_{7/5}(k,i,d_p)} \approx \frac{t_{7/5}(k,i,d_p)}{\binom{k}{i}}, \qquad (2\text{-}19)$$

which gives the average upper bound on bit error probability as

$$P_b \le \sum_{d=d_{\min}}^{3k} \underbrace{\sum_{i} \sum_{d_1} \sum_{d_2}}_{d=i+d_1+d_2} \frac{i}{k}\binom{k}{i} P_{7/5}(d_1|i) P_{7/5}(d_2|i) P_2(d). \qquad (2\text{-}20)$$

2.5 SUMMARY

In this chapter, various aspects of turbo codes were discussed. First, Section 2.1 introduced the turbo code scheme, followed by a description of turbo codes encoders In Section 2.2. In Section 2.3, the MAP and SOVA algorithms were discussed in detail. Finally in Section 2.4, the union upper bound to measure the performance of turbo codes was presented.

CHAPTER THREE:
FLAT FADING CHANNEL AND ITERATIVE CHANNEL ESTIMATION AND DECODING

In this chapter, wireless channel (specifically, narrowband multipath fading channel) characteristics will be described, followed by a detailed discussion of channel estimation using PSAM. Finally, the iterative channel estimation and decoding scheme over flat fading channel is described.

3.1 FLAT RAYLEIGH FADING CHANNEL

The impulse response of the multipath fading channel is given as

$$c(\tau;t)=\sum_{n=0}^{N(t)} g_n(t)e^{-j\varphi_n(t)}\delta\left(\tau-\tau_n(t)\right), \tag{3-1}$$

where g_n,ϕ_n,τ_n denotes the amplitude, phase and delay of the nth path respectively. Denoting the multipath delay spread as T_m, if $T_m \ll B_u^{-1}$ (where B_u is the baseband modulation bandwidth), then the channel is a narrowband multipath flat fading channel. In this case, the channel gain (3-1) can be simplified as

$$c(t)=\sum_n \alpha_n(t)e^{-j\varphi_n(t)} = r_I(t)+jr_Q(t). \tag{3-2}$$

For the narrowband multipath flat-fading channel, the real and imagnary parts of the channel gain are independent Gaussian processes. The magnitude of the channel gain follows the Rayleigh distribution, and the phase follows the uniform distribution in $[0,2\pi]$. The autocorrelation of the complex channel gain is

$$A_{r_I}(\tau) = A_{r_Q}(\tau) = \left(\sum_n E\left[g_n^2\right]/2\right) J_0(2\pi f_d \tau), \tag{3-3}$$

where g_n is the magnitude of the channel gain, $f_d = \frac{v}{c} f_c$ is the maximum doppler frequency, v is vehicle speed, c is speed of light, f_c is carrier frequency and $J(\cdot)$ is the zeroth order bessel function. The cross-correlation of the complex channel gain is

$$E\left(r_I(t) r_Q(t)\right) = 0.$$

For isotropic channel and vertically polarized antenna, the power spectrum of the narrowband multipath channel gain is given as

$$S_g(f) = \sigma^2 / \pi \sqrt{f_d^2 - f^2},$$

where $\sigma^2 = \sum_n E\left[\alpha_n^2\right]/2$, $f_d = \frac{v}{c} f_c$. The autocorrelation of the complex channel gain is shown in Figure 3.1, and its power spectrum is shown in Figure 3.2.

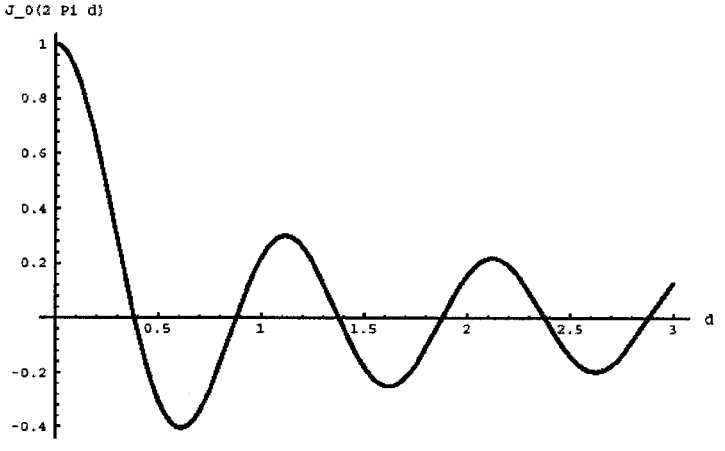

Figure 3.1 Autocorrelation of complex channel gain

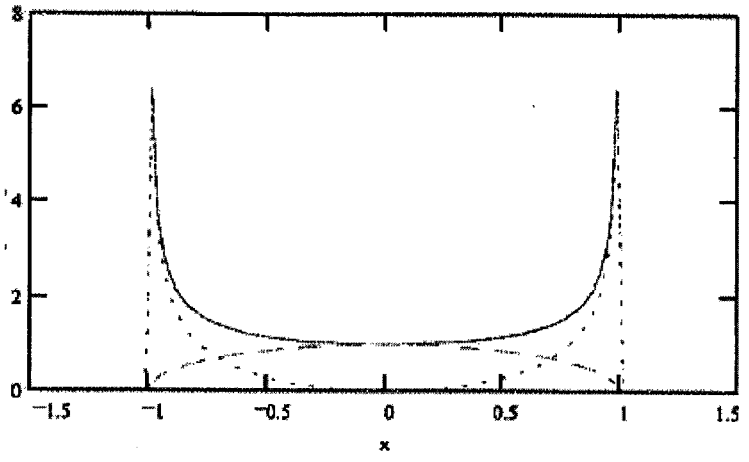

Figure 3.2 Power spectrum of Rayleigh fading channel gain

The Rayleigh fading process can be modeled as in Figure 3.3. First, generate N zero mean Gaussian distributed samples for the real and imaginary parts respectively. Then, pass these two streams through the root square power spectrum filter in order to generate the real and imaginary parts of the channel gain. The square root of the real and imaginary parts is the magnitude of channel gain, which follows Rayleigh distribution.

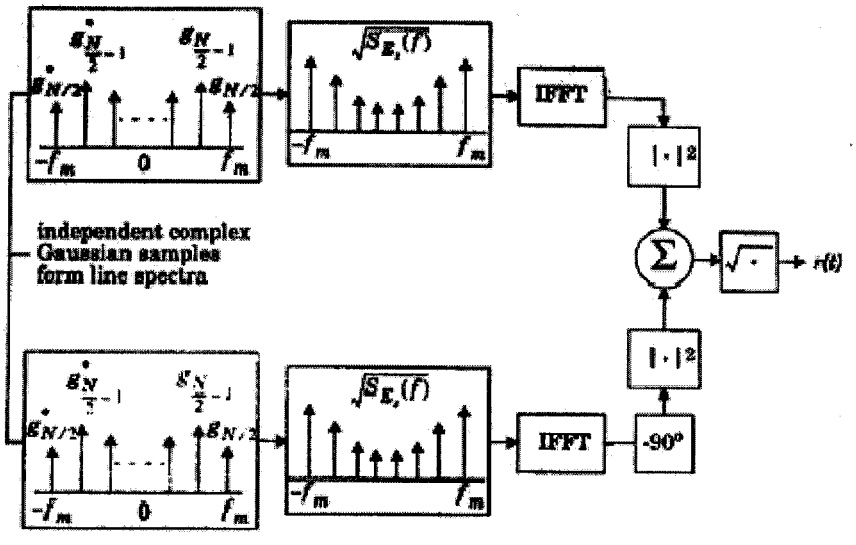

Figure 3.3 Rayleigh fading channel gain modeling [33]

3.2 CHANNEL CAPACITY

According to Shannon's theorem, a given communication system has a maximum rate of information C, known as the channel capacity. If the information rate R is less than C, then one can approach arbitrarily small error probabilities by using intelligent coding techniques. To get lower error probabilities, the encoder has to work on longer blocks of signal data, which corresponds to longer delay and higher computational complexity requirements. Thus, capacity can be regarded as a criterion to evaluate how "good" a code is. Channel capacity can be derived for the case when channel side information is available (CSI), and for the case when it is not (NSI).

The discrete channel model can be defined as $y = gx + n$, where x is the input BPSK with energy constraint $E < E_s$, g is the channel gain. Note that x and n are independent.

3.2.1 CHANNEL CAPACITY WITH CSI

The channel capacity is defined as the maximum of the mutual information between channel input and output *I(x,y),* over the input distribution, $P_X(x)$. When CSI is known, the mutual information is formed conditioned on the knowledge of the channel gain g as follows:

$$
\begin{aligned}
C_{SI}^{BPSK} &= \max_{P_{X(x)}} \left\{ I\left(x, y \middle| g\right) \right\} \\
&= \max_{P_{X(x)}} \left\{ E_{p(x,y,g)} \left[\log_2 \left(\frac{p\left(x, y \middle| g\right)}{p\left(x \middle| g\right) p\left(y \middle| g\right)} \right) \right] \right\} \\
&= \max_{P_{X(x)}} \left\{ E_{p(x,y,g)} \left[\log_2 \left(\frac{p\left(x, y \middle| g\right)}{\sum_z P_X(z) p\left(y \middle| z, g\right)} \right) \right] \right\}
\end{aligned}
,
$$

where $p(x,y,g)=p(y|x,g)P_X(x)p_g(g)$, as x and g are independent. For symmetric channels with a finite input alphabet, the maximization in capacity is achieved with an equiprobable input distribution $P_X\left(x=\sqrt{E_s}\right)=P_X\left(x=-\sqrt{E_s}\right)=1/2$. Thus, the capacity can be further expressed as

$$C_{SI}^{BPSK}=\sum_x \int_g \int_y 1/2 p_g(g) p(y|x,g)\log_2\left[\frac{p(y|x,g)}{\sum_z 1/2 p(y|z,g)}\right]dydg.$$

By symmetry, C_{SI}^{BPSK} becomes

$$C_{SI}^{BPSK}=\sum_x \int_g \int_y p_g(g) p\left(y|x=\sqrt{E_s},g\right)\log_2\left[\frac{p\left(y|x=\sqrt{E_s},g\right)}{\sum_z 1/2 p(y|z,g)}\right]dydg,$$

and can be further simplified

$$C_{SI}^{BPSK}=\sum_x \int_g \int_y p_g(g) p\left(y|x=\sqrt{E_s},g\right)\log_2\left[\frac{p\left(y|x=\sqrt{E_s},g\right)}{\sum_z 1/2 p(y|z,g)}\right]dydg, \qquad (3\text{-}4)$$

where g is Rayleigh distributed with average power equal to 1, and y is Gaussian distributed with mean $g\sqrt{E_s}$ and variance $N_0/2$. In addition, let

$$\varphi(y,g)=\frac{p\left(y|x=-\sqrt{E_s},g\right)}{p\left(y|x=\sqrt{E_s},g\right)}=e^{-(1/N_0)\cdot y\cdot g\cdot\sqrt{E_s}}.$$

3.2.2 CHANNEL CAPACITY WITHOUT CSI

In the case where CSI is not available, the channel capacity is formed as

$$\begin{aligned} C_{NSI}^{BPSK} &= \max_{P(X)} \{I(x,y)\} \\ &= \max_{P(X)} \left\{ E_{p(x,y)} \left[\log_2 \left(\frac{p(x,y)}{P_X(x)\,p(y)} \right) \right] \right\}, \\ &= \max_{P(X)} \left\{ E_{p(x,y)} \left[\log_2 \left(\frac{p(y|x)}{p(y)} \right) \right] \right\} \end{aligned}$$

where $p(x,y) = \int_c p(x,y,c)\,dc = \int_c p_c(c) \cdot P_X(x) \cdot p(y|x,c)\,dc$. After substitution and simplification, C_{NSI}^{BPSK} can be expressed as

$$C_{NSI}^{BPSK} = -\int_g \int_y p_g(g) \cdot p\left(y \middle| x = \sqrt{E_s}, g\right) \log_2 \left[\frac{1}{2} \left(1 + \phi(y)\right) \right] dy\,dg\,, \qquad (3\text{-}5)$$

where $\phi(y) = \dfrac{\int_g p_g(g)\, p\left(y \middle| x = -\sqrt{E_s}, g\right) dg}{\int_g p_g(g)\, p\left(y \middle| x = \sqrt{E_s}, g\right) dg}$.

Figure 3.4 gives the numerical results for channel capacity, in both the CSI and NSI cases. It can be seen that the lack of CSI amounts to a loss of approximately 1dB in E_s / N_0. According to the channel coding theorem, a code exists that will give arbitrarily small errors if the code rate is less than the channel capacity. The smallest E_b / N_0 can be determined by equating the code rate and channel capacity. Through puncturing, an arbitrary code rate can be achieved. Using the widely applied rate ½ turbo codes, equations (3-4) and (3-5) will yield the following relation

$$\left(\frac{E_b}{N_0} \right)_{C=r} = \frac{1}{r} \left(\frac{E_s}{N_0} \right)_{C=r}, \qquad (3\text{-}6)$$

from which it can be found that the minimum SNR for rate ½ codes is 1.8 dB when pefect CSI is available, and s 2.7 dB when no CSI is available.

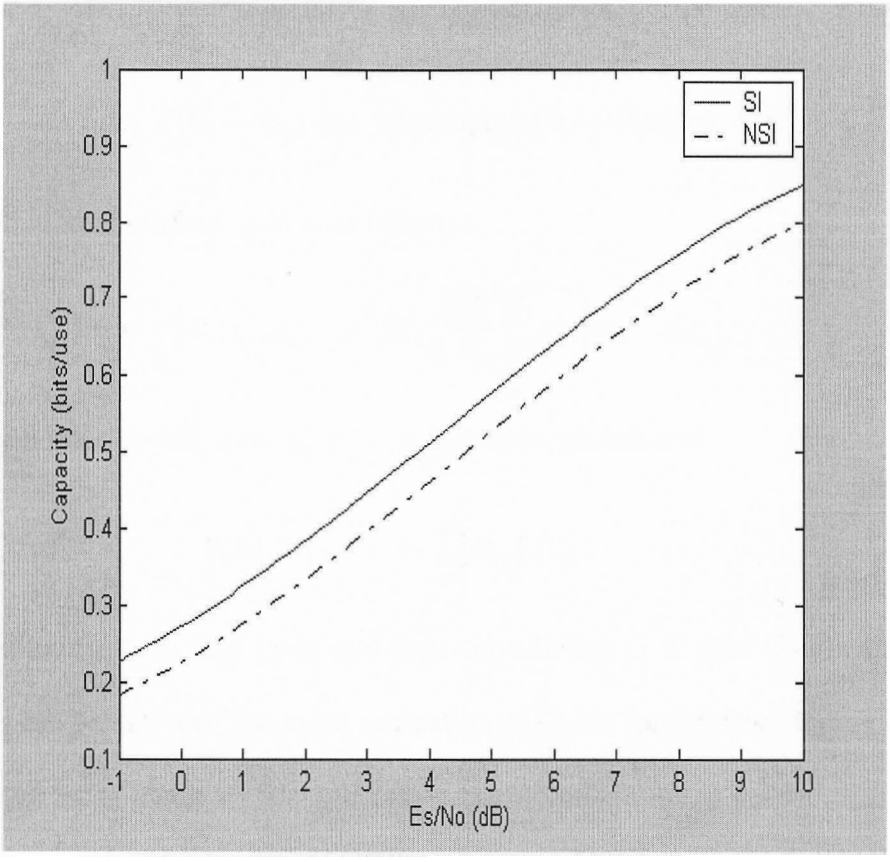

Figure 3.4 Capacity on fully-interleaved Rayleigh fading channel with BPSK signaling

3.3 PERFORMANCE BOUND FOR TURBO CODES OVER RAYLEIGH FADING CHANNEL

According to Section 2.4, to compute the bit error probability P_b for the Rayleigh fading channel, the two codeword probability $P_2(d)$ is necessary. Assuming a Rayleigh fading channel with CSI available, the probability of erroneously selecting codeword $\mathbf{C}_j$ when codeword $\mathbf{C}_i$ is transmitted, is

$$P\left(\mathbf{C}_i \to \mathbf{C}_j \middle| g\right) = Q\left(\sqrt{\frac{2E_s}{N_0}\sum_{k=1}^{d} g_{i_k}^2}\right), \tag{3-7}$$

where $(i_1, i_2, \cdots i_d)$ are the indexes in bit positions where $\mathbf{C}_j$ differs from $\mathbf{C}_i$, and $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-z^2} dz$. $P\left(\mathbf{C}_i \to \mathbf{C}_j\right)$ can be computed by averaging $P\left(\mathbf{C}_i \to \mathbf{C}_j \middle| a\right)$ over the joint distribution of channel gain **a** as follows:

$$P\left(\mathbf{C}_i \to \mathbf{C}_j\right) = \int_{g_{i_1}} \cdots \int_{g_{i_d}} P_{\mathbf{g}}\left(g_{i_1}, g_{i_2}, \cdots g_{i_d}\right) Q\left(\sqrt{\frac{2E_s}{N_0}\sum_{k=1}^{d} g_{i_k}^2}\right) dg_{i_1} \cdots dg_{i_d}. \tag{3-8}$$

Consider full interleaving, then $g_{i_1}, g_{i_2}, \cdots g_{i_d}$ are independent and

$$\mathbf{P}_g\left(g_{i_1}, g_{i_2}, \cdots g_{i_d}\right) = \prod_{k=1}^{d} P_{g_{i_k}}\left(g_{i_k}\right). \tag{3-9}$$

After substituting (3-9) into (3-8) and then substituting (3-8) into (2-20), the bit error probability can be derived. The exact evaluation of (2-20) for Rayleigh fading is difficult, especially for large block length, and thus numeric integration is needed. The following simplification for $Q(x)$ [32] can be used for (3-9):

$$Q(x) = \frac{1}{\pi}\int_0^{\pi/2} e^{-x^2/\left(2\sin^2\phi\right)} d\phi. \tag{3-10}$$

By substitution using (3-10), $P_2\left(d \middle| \mathbf{g}\right)$ is simplified as

$$P_2\left(d \middle| g\right) = \frac{1}{\pi}\int_0^{\pi/2} e^{\frac{-\left(\frac{2E_s}{N_0}\sum_{k=1}^{d} g_{i_k}^2\right)}{2\sin^2\phi}} d\phi. \tag{3-11}$$

As the channel gains are independent, the d-dimensional integral of $P_2(d)$ has the following closed form expression:

$$P_2(d) = \frac{1}{\pi}\int_0^{\pi/2}\left[\frac{\sin^2\phi}{E_s/N_0 + \sin^2\phi}\right]^d d\phi. \tag{3-12}$$

While (3-12) has no closed form solution, it is numerically integrable.

3.4 PILOT SYMBOL ASSISTED MODULATION (PSAM)

Pilot assisted modulation was first proposed in 1984, and later in 1987 as pilot tone modulation. Data and pilot tone can be separated by low pass filtering, if the spectral notch is wide enough to accommodate the Doppler-spread tone and data. However, pilot tone assisted modulation has the disadvantages of requiring the precoding of data, leading to bandwidth expansion and an increase in the dynamic range of the signal; it therefore requires amplifer linearization.

The pilot symbol assisted modulation scheme was first proposed by Dr. Cavers in 1991 [17]. The block diagram is shown in Figure 3.5.

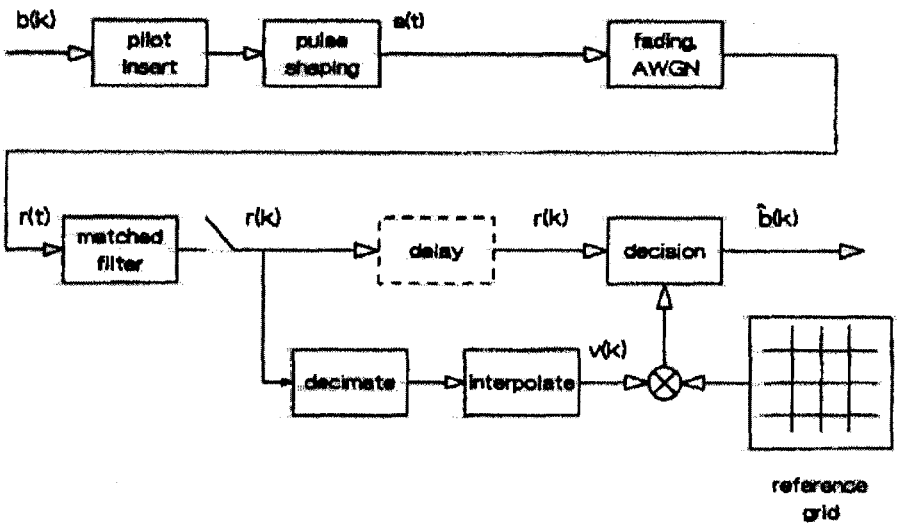

Figure 3.5 Pilot Symbol Assited Modulation scheme [17]

The pilot symbols are periodically inserted into the data stream prior to pulse shaping, resulting in a frame structure as shown in Figure 3.6

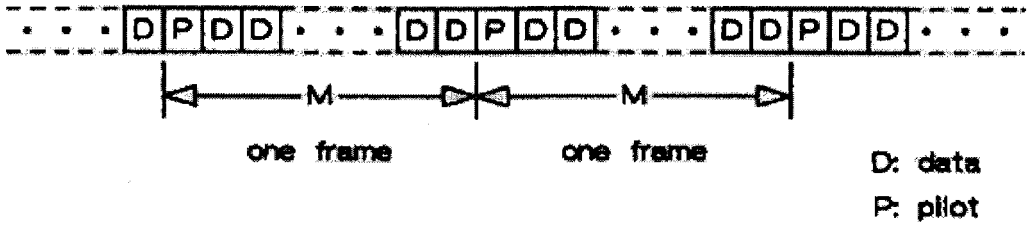

Figure 3.6 Frame structure of PSAM [17]

The frame size M must be chosen such that the Nyquist sampling criterion is satisfied:

$$f_s = \frac{1}{MT} \geq 2f_d . \tag{3-13}$$

According to [17], the transmitted signal with a complex envelope is given by

$$s(t) = A\sum_{k=-\infty}^{\infty} b(k)p(t-kT), \tag{3-14}$$

where T is the symbol duration, A is the amplitude factor and $p(t)$ is a unit energy pulse. Pilot symbols are inserted into data stream at time index $i = kM$, $k = 0,1,2,\cdots$, with a known value $\tilde{b}$. The fading channel output is given by

$$r_c(t) = g(t)s(t) + n_c(t). \tag{3-15}$$

Assuming there is no frequency offset, the autocorrelation of the complex channel gain is given by

$$R_c(\tau) = \sigma_g^2 J_0(2\pi f_D \tau). \tag{3-16}$$

After matched filtering with impulse response $p^*(t)/\sqrt{N_0}$, the symbol-spaced samples $r(kT)$ are given by

$$r(k) = \frac{Ac(kT)}{\sqrt{N_0}} b(k) + n(k) = u(k)b(k) + n(k). \tag{3-17}$$

The Gaussian noise samples $n(k)$ are white with unit variance. Defining q as the pilot power to data power, then q is given by

$$q = \left|\tilde{b}^2\right| / (M-1) E\left[|b|^2\right]. \tag{3-18}$$

The total energy E_f and bit energy E_b are given as

$$E_f = \sigma_g^2 A^2 \left(\left|\tilde{b}\right|^2 + (M-1) E\left[|b|^2\right]\right). \tag{3-19}$$

and

$$E_b = \frac{\sigma_g^2 A^2}{n(M-1)} \left|\tilde{b}\right|^2 \left(\frac{1+q}{q}\right). \tag{3-20}$$

respectively. The channel gain $u(k)$ has variance

$$\sigma_u^2 = \frac{1}{2} E\left[|u|^2\right] = \gamma_b \frac{n(M-1)}{\left|\tilde{b}\right|^2} \frac{q}{1+q}. \tag{3-21}$$

where $\gamma_b = E_b / N_0$. As indicated in [17], letting $b(0)$ designate a pilot symbol and considering the detection of $b(k), -\lfloor M/2 \rfloor \le k \le \lfloor (M-1)/2 \rfloor$, the channel gain estimator uses the K nearest pilot samples to do the estimation

$$v(k) = \sum_{i=-\lfloor K/2 \rfloor}^{\lfloor K/2 \rfloor} h^*(i,k) r(iM). \tag{3-22}$$

The estimation error is denoted as $e(k)$ and $u(k) = v(k) + e(k)$. The optimal filter to minimize the variance of estimation error *e*, is a Weiner filter. Define the length *K* column vector *r* as the set of pilot samples $r(iM)$, $-\lfloor K/2 \rfloor \le i \le \lfloor K/2 \rfloor$, and $h(k)$ as the corresponding set of coefficients for the *k*th position in the frame. The estimated channel gain is given by $v(k) = \mathbf{h}^\dagger \mathbf{r}$, and the estimation error can be denoted by

$$e(k) = u(k) - \mathbf{h}^\dagger \mathbf{r}. \tag{3-23}$$

Then the optimal channel gain estimates results when $\mathbf{h}(k)$ satisfies $\mathbf{Rh}(k)=\mathbf{w}(k)$, as the optimization is in the minimum mean squared error sense with respect $e^2(k)$, where $\mathbf{R}$ is the $K\times K$ autocorrelation matrix and $\mathbf{w}(k)$ is the $M-1$ length k covariance vector, which are given as÷

$$\mathbf{R}=\frac{1}{2}E\left[\mathbf{r}\mathbf{r}^{\dagger}\right] \tag{3-24}$$

and

$$\mathbf{w}(k)=\frac{1}{2}E\left[u^{*}(k)r\right] \tag{3-25}$$

respectively. The specific components in $\mathbf{R}$ and $\mathbf{w}$ are given by

$$\mathbf{R}_{ik}=\gamma_b n(M-1)\frac{q}{1+q}\tilde{R}_c\left((i-k)MT\right)+\delta_{ik} \tag{3-26}$$

and

$$\mathbf{w}_i(k)=\gamma_b\frac{n(M-1)}{\tilde{b}^{*}}\frac{q}{1+q}\tilde{R}_c\left((iM-k)T\right). \tag{3-27}$$

The corresponding minimum estimation error variance is given as

$$\sigma_e^2(k)=\sigma_u^2-\sigma_v^2, \tag{3-28}$$

where $\sigma_v^2(k)=\sigma_{uv}^2(k)=\mathbf{w}^{\dagger}(k)\mathbf{R}^{-1}\mathbf{w}(k)$. The BER for optimal $h(k)$ for BPSK is given by

$$P_b(k)=\left(1-\rho(k)\right)/2, \tag{3-29}$$

where

$$\rho(k)=\frac{\mathbf{w}^{\dagger}(k)\mathbf{R}^{-1}\mathbf{w}(k)}{\sqrt{\left(\sigma_u^2+1\right)\mathbf{w}^{\dagger}(k)\mathbf{R}^{-1}\mathbf{w}(k)}}. \tag{3-30}$$

3.5 ITERATIVE CHANNEL ESTIMATION AND DECODING

When Turbo codes are used as the channel coding for a Rayleigh fading channel, the decoded symbol can be further used to re-estimate the channel, thus improving the channel estimation and decoding performance. The scheme is illustrated in Figure. 3.7.

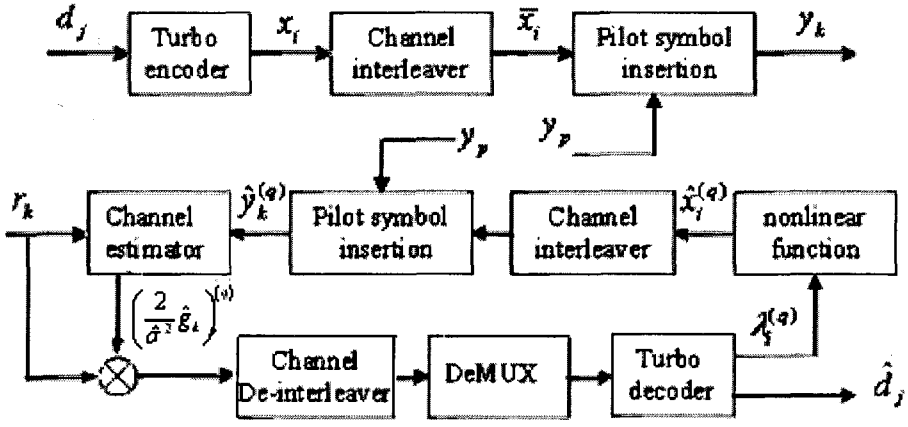

Figure 3.7 Iterative channel estimation and decoding scheme [14]

In Figure. 3.7, the data stream $\{d_j\}, 1 \le j \le L$, or in polar form $d_j \in \{0,1\}$, is first encoded by the rate ½ turbo encoder and then bipolar mapped. Fading channels produce burst errors, however, turbo codes are effective for random errors. Thus, the encoded and mapped bits $\{x_i\}$, $x_i \in \{-1,1\}$, $1 \le i \le L/r$ (r=1/2) are then passed through a block channel interleaver, which scrambles the burst errors into random errors. The interleaved sequence $\{\bar{x}_i\}$ is parsed into groups of (M-1) contiguous bits. A pilot symbol is inserted every (M-1) bits to form M-bit data groups. To maintain the effective code rate, r equal to ½ for every group of M encoded bits, one parity bit is punctured, so that the size of each group is still (M-1). The data groups of size (M-1) are then reassembled into a sequence of symbols $\{y_k\}$, $1 \le k \le L/r$. After pulse shaping, the transmitted sequence $\{y_k\}$ passes

through a flat-fading channel with AWGN. The signal is then matched filtered, and the received signal is

$$r_k = g_k y_k + n_k ,$$

where $\{n_k\}$ are statistically independent complex-valued Gaussian random variables, with zero mean and variance $\sigma^2 = N_0 / 2E_s$ in each dimension. The received sequence $\{r_k\}$ is first applied to the channel estimation algorithm described in Section 3.4 in order to perform channel estimation. After channel compensation, the data sequence goes through the channel de-interleaving and demultiplexing process, and is then fed to the turbo decoder. The receiver has a feedback path, and after the qth ($q \geq 1$) iteration of turbo decoding using the MAP decoder, the combination of each received signal y_k and its decoded LLR becomes a pilot symbol. In this way, the channel can be re-estimated to improve channel estimation and decoding performance.

3.6 SUMMARY

In this chapter, the multipath fading channel characteristics, the channel capacity and the BER of turbo codes when applied to the fading channel, were first discussed. Next, the channel estimation method PSAM was introduced, and the iterative channel estimation and decoding scheme was presented.

CHAPTER FOUR:
PILOT ASSISTED SOVA DECODER

In this chapter, a pilot assisted turbo codec scheme based on the conventional SOVA decoder is proposed. First, the motivation and idea of pilot assisted turbo codec is discussed, followed by the pilot assisted encoder and SOVA decoder, which are described in detail. Finally, simulation results for this scheme are presented.

4.1 MOTIVATION FOR PILOT ASSISTED SOVA DECODER

Currently, turbo decoding is a type of blind decoding. The LLR can be used to judge the reliability of a decision; however, little can be done if the reliability is found to be low, or when LLR is high but it is an erroneous decision, which is frequently the case when burst errors happen. In order to detect errors after each stage of turbo decoding and to further help to correct these errors, pilot bits (known bits) can be used to assist the decoder. In order to maintain bandwidth efficiency, both the pilots and a corresponding number of parity bits are punctured at the encoder.

In the scheme proposed in this thesis, pilot bits are inserted regularly, say every m bits. Since RSC is employed, these pilot bits would appear directly in the coded data stream and can be reinserted by the decoder. However, this cannot be done with the parity bits, since they are an algebraic combination of the elements in the memory, which are dependent on previous, unknown data. Turbo codes are known to be very robust to uncorrelated random errors, however burst errors cause problems. Heuristically, errors in pilot symbols are thus an indication of the presence of burst errors.

Hence, a method is proposed that locates the erroneous path, and then scales the extrinsic information associated with the potentially problematic bits with a factor less than 1. The idea is that their extrinsic reliabilities should be given less weight when fed into the next decoding stage. Another method related to de-correlation of the *a priori* and extrinsic information is also proposed. Another advantage of pilot insertion, is that it increases the interleaver gain somewhat due to the larger frame-size after the pilot bits are inserted.

4.2 PILOT ASSISTED TURBO ENCODER

On the encoder side, the length of the original bit stream, *N*, can be written as $N = nm + r$, where $r < m$. When a pilot bit is inserted every *m* information bits, this length of the bit stream to be coded will grow by *n* bits. The turbo encoder interleaver for this scheme is illustrated in Fig. 4.1, where the notation ζ_j stands for a pilot bit and u_i stands for an information bit. The interleaver is designed to keep the positions of pilot bits invariant. After interleaving, information bits will still occupy the positions of information bits and are re-indexed as $\{b_1, b_2 ..., b_{nm+r}\}$.

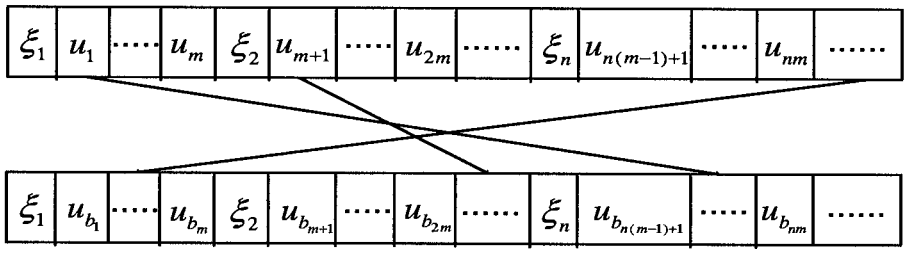

Figure 4.1 Interleaver design for pilots and data bits

As with the conventional turbo encoder, both the bit stream with inserted pilots and its interleaved version are fed into the two component RSC encoders. Different code rates

can be achieved through different puncturing patterns. The conventional puncturing pattern for rate ½ encoder is odd-even puncturing. In this scheme, to maintain the effective code rate of ½, the puncturing pattern is designed as shown in Figures 4.2 and 4.3, for when m is odd and when m is even, respectively. The design principle of this puncturing pattern is the same whether m is odd or even which is as follows.

Consider 2*m {replace with $2 \cdot m$ }information symbols, then 2 pilots are inserted, and the locations of these two pilots are m+1 and 2m+2. To maintain the effective code rate of ½, the parity bits following the pilots in both encoders are punctured. If for the first m information bits, the puncture pattern is odd-even (even-odd) puncturing, then for the next m information bits, the puncture pattern is reversed to even-odd (odd-even) puncturing. In this way, the number of parity bits punctured on both parity bit streams are the same and are evenly spaced, which leads to the balanced decoding performance at both decoders on the receiver side.

u_1	u_2	u_3	ξ_1	u_4	u_5	u_6	ξ_2	u_7	u_8	u_9	ξ_3	u_{10}	u_{11}	u_{12}	ξ_4	u_{13}	u_{14}	u_{15}	ξ_5	u_{16}	u_{17}	u_{18}	ξ_6
p_1		p_3			p_5			p_7		p_9			p_{11}			p_{13}		p_{15}			p_{17}		
	p_2			p_4		p_6			p_8			p_{10}		p_{12}			p_{14}			p_{16}		p_{18}	

Figure 4.2 Puncture pattern for pilot assisted turbo encoder (m odd)

u_1	u_2	u_3	u_4	ξ_1	u_5	u_6	u_7	u_8	ξ_2	u_9	u_{10}	u_{11}	u_{12}	ξ_3	u_{13}	u_{14}	u_{15}	u_{16}	ξ_4
p_1		p_3			p_5		p_7			p_9		p_{11}			p_{13}		p_{15}		
	p_2		p_4			p_6		p_8			p_{10}		p_{12}			p_{14}		p_{16}	

Figure 4.3 Puncture pattern for pilot assisted turbo encoder (m even)

Heuristically, the pilots should also appear like random data without periodicity; a pseudo-random sequence is a good choice for the pilots.

4.3 PILOT ASSISTED SOVA DECODER

The decoder algorithm is shown in Figure 4.4.

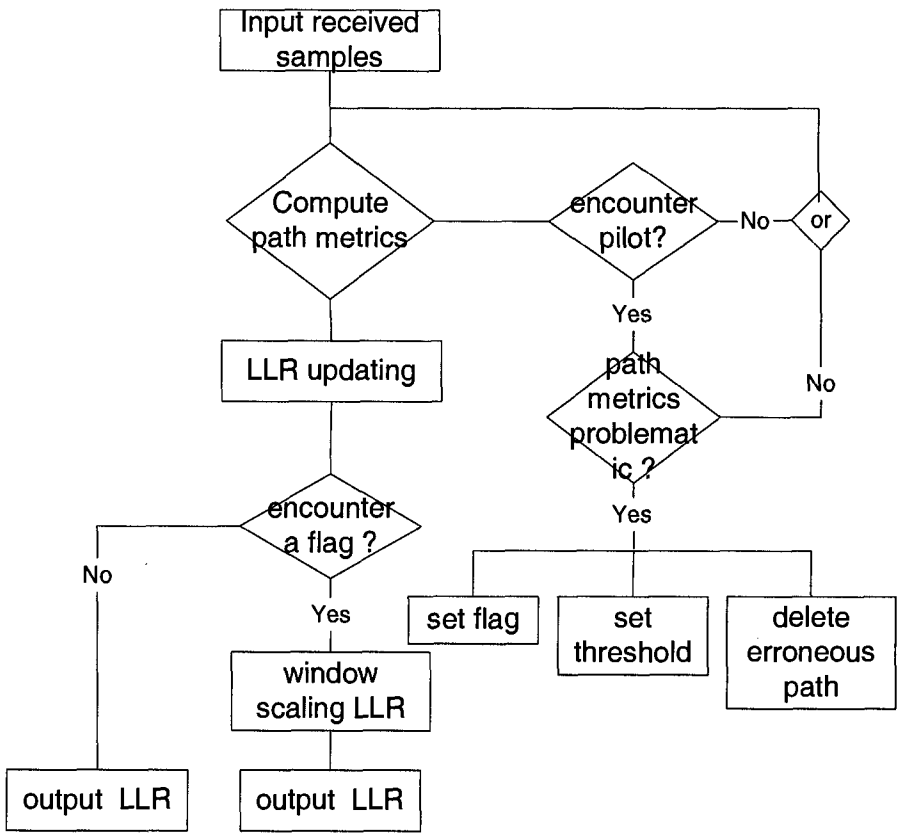

Figure 4.4 Pilot assisted SOVA decoder

On the receiver side, the received signal is first BPSK demodulated, and the pilots are then periodically re-inserted into the bit stream in the places that they are punctured from, with +1 corresponding to bit 1 and -1 corresponding to bit 0. The reliability values of the re-inserted pilots are set to 0 for all decoding stages. The parity bits corresponding to the pilots are left as zeros, as the blanks indicated in the puncturing pattern in Figure 4.2 and Figure 4.3.

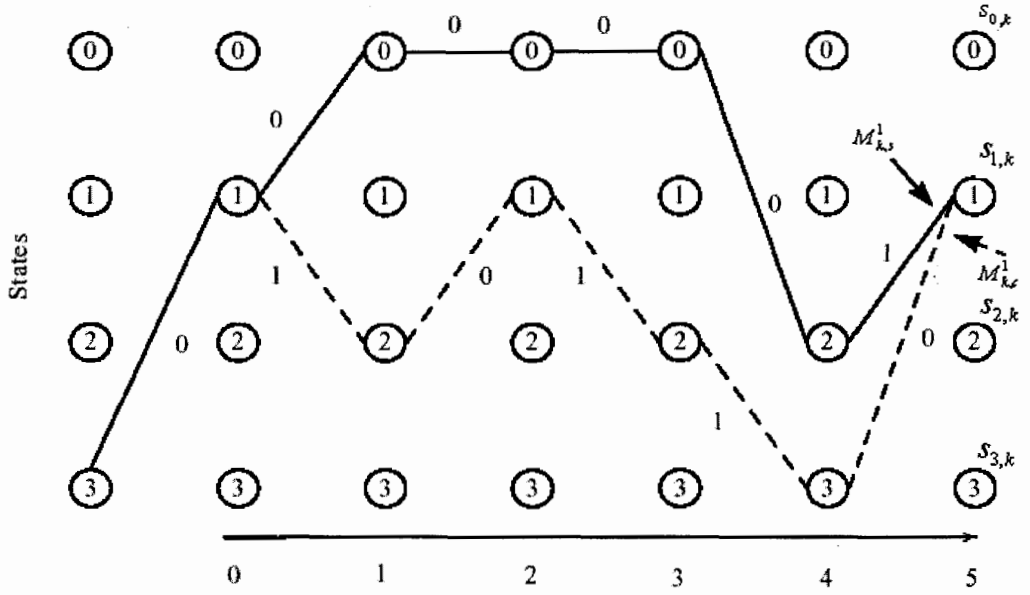

Figure 4.5 Erroneous path detection

Consider the path metric computation and path choice at the pilot bits. Assume all the pilot bits are 1s for convenience. As shown in Figure 4.5, the path entering a state with information bit "1" is represented by the solid line, and the path entering a state with information bit "0" is represented by the dashed line. The path metric associated with information bit "1" is denoted as $M_{k,s}^{(m)}$ where the subscript 's' stands for survivor, and the path metric associated with information bit '0' is denoted as $M_{k,c}^{(m)}$ where the subscript 'c' stands for competitor. The path metric is computed according to (2-7). If $M_{k,s}^{(m)} \le M_{k,c}^{(m)}$, i.e. the path metric associated with information bit 1 is less than the path metric associated with information bit 0, then the path through the pilot bit is regarded as problematic; otherwise, if $M_{k,s}^{(m)} > M_{k,c}^{(m)}$, then it is regarded as correct and no corrective measure is taken.

Next, if there is a problematic path passing through the pilot bit, then the following measures are taken. A flag is set to indicate a "problematic" path associated with the pilot bit. The path metric associated with information bit "0" is set to the path metric associated with information bit "1" minus a small positive value, l , which is sufficient to eliminate the erroneous path. Next, a small positive value is used for the initial LLR. This small initial LLR for the pilot bit will serve as a threshold to suppress the over-estimation of the LLR. Experimental evidence has shown that setting $l = 2$ gives good results.

After running a SOVA decoding stage, checks are made to see if the final decoded path has passed through a flagged state, in which case the extrinsic information values neighbouring the pilots are scaled by a factor $\beta(\beta < 1)$. A scaling window of width $2T$ is defined, and centred on the pilot bit in question. The window is set in this way as the bits in the range of the truncation length will be affected by the threshold associated with the pilot, when undergoing reliability updating. The value $b = 0.5$ has been experimentally found to be effective.

Reliability updating is skipped for the pilot bits, since these bits are known. The increase in the complexity of the pilot assisted scheme is only due to encoding a small portion of bits on the encoder side and a scaling of the extrinsic information; this increase is negligible when compared with the overall complexity of the conventional turbo SOVA codec.

4.4 MODIFIED PATH METRIC METHOD

As indicated in [9], the SOVA decoder suffers from two serious problems: an over-estimation of extrinsic information, and excessive correlation between the *a priori* information and the extrinsic information. In this thesis, a scheme is proposed to modify the basic path metric given in (2-7) as follows

$$M_k\left(s^{(i)}\right) = M_{k-1}\left(s'^{(i)}\right) + \beta\left[\frac{1}{2}\left(L_c y_{k,1} u_k + L(u_k) u_k\right)\right] + \frac{1}{2}\sum_{v=2}^{N} L_c y_{k,v} x_{k,v}^{(i)} \qquad (4\text{-}1)$$

The extrinsic information is computed as

$$L_e(\hat{u}_k) = \beta\left(L(\hat{u}_k) - L_c y - L_{in}(u_k)\right) \qquad (4\text{-}2)$$

where the *a priori* information is weighted by a factor of β $(\beta < 1)$ in order to reduce the correlation between the *a priori* information and the extrinsic information.

4.5 SIMULATION RESULTS

To verify the effectiveness of our algorithms, simulations have been performed for an AWGN channel using the rate ½, constraint-length 4 component code described by the following generators: $g_0(D) = (1 + D^3 + D^4)$ and $g_1(D) = (1 + D^2 + D^3 + D^4)$; this code has the largest free distance for constraint-length 4 RSC codes. The interleaver block-size is set to 400 and the uniform interleaving strategy is used.

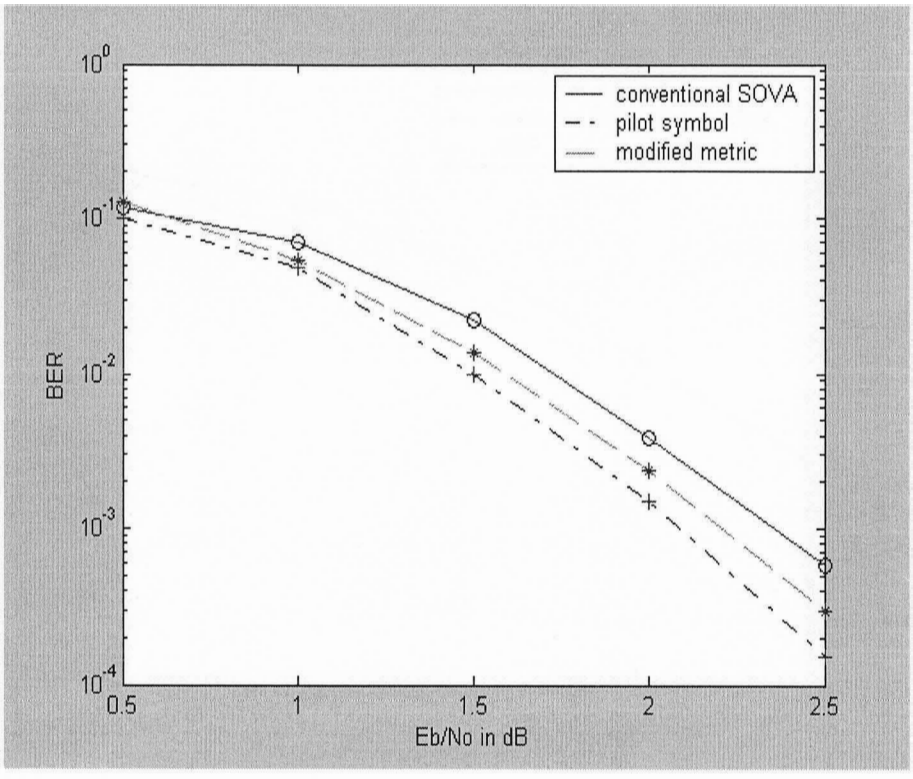

Figure 4.6 The benefit of pilot bits and metric modification in SOVA

The decoder is based on the commonly used SOVA decoder in [3], and the basic results have been verified against those published. For the first scheme in this paper, a pilot bit is inserted every 8 bits and the results are shown in Figure 4.6. From the figure, we can see that the use of pilot bits results in a decoder that outperforms the conventional SOVA decoding, reducing the BER by factor of 4 at an SNR of 2.5 dB. The metric modification method, on the other hand, only lowers the BER by a factor of 2 at the same SNR. The corresponding coding gains are 0.38 dB and 0.12 dB respectively.

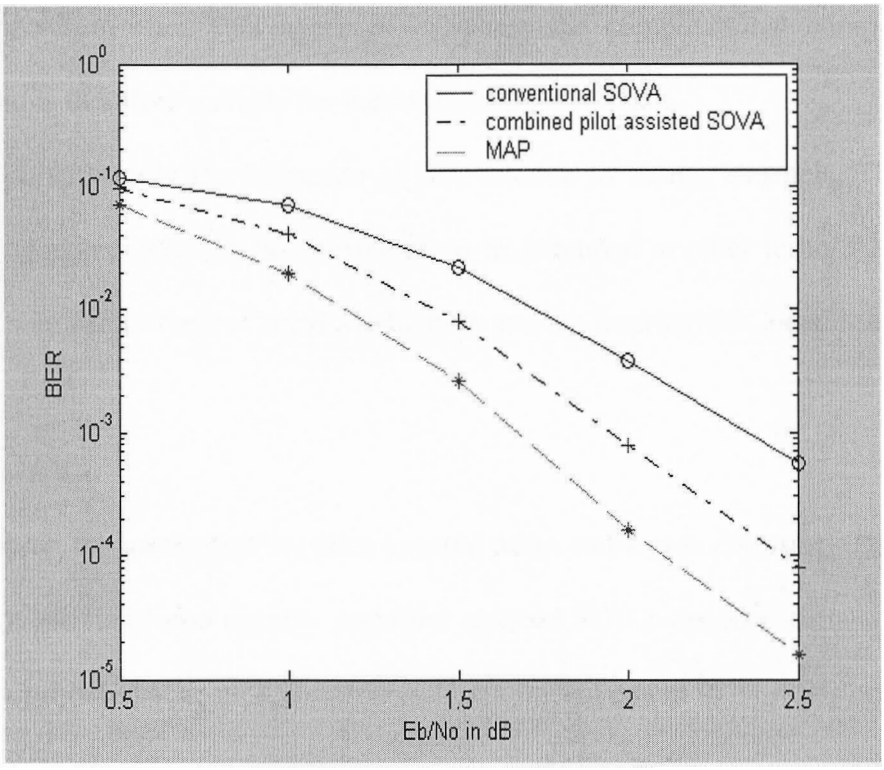

Figure 4.7 The combined benefit

The two schemes, however, work on roughly independent mechanisms and can be combined; Figure 4.7 illustrates the improvement associated with the combined approach. The performance of the combined scheme is compared with the conventional SOVA decoder and the MAP decoder. It is observed that the BER of the combined scheme has dropped by an order of 6.5 at 2.5 dB. This corresponds to a coding gain of about 0.45 dB; a significant amount given the negligible increase in complexity. The performance has still not reached that of the MAP decoder; however, this combined scheme has effectively "split the gap" between the conventional SOVA and MAP approaches. Simulations for other encoders have also been performed, and show similar results.

In this chapter, a new scheme is developed that employs pilot bits and metric modification, to improve the performance of turbo codes by about 0.45 dB without

changing the data rate. This approach increases the computational complexity only slightly, and is therefore suitable for hardware implementation.

Future work will study the extension of this scheme to fading channels. The pilot bit approach presented here can be considered to be extended to other turbo/SOVA related topics, such as Turbo Trellis Coded Modulation and Bit Interleaved Coded Modulation.

4.5 SUMMARY

In this chapter, the motivation for pilot assisted turbo codec was discussed first, followed by the pilot assisted turbo encoder and pilot assisted SOVA decoder. Simulation results showed the proposed scheme achieves significant improvement in BER performance with very slightly increased complexity.

CHAPTER FIVE:
UNBALANCED POWER ALLOCATION FOR ITERATIVE CHANNEL ESTIMATION AND DECODING

In this chapter, the unbalanced power allocation for PSAM is first revisited, followed by a study of unbalanced power allocation for iterative channel estimation and decoding, through simulations.

5.1 UNBALANCED POWER ALLOCATION FOR PSAM

In [15], various aspects of PSAM have been studied in detail. The analysis in [15] is conducted for the general case, where the pilot symbol power may differ from the data symbol power. The simplest and most common case for PSAM is when the pilot symbol power and data symbol power are the same, however, this case may not give the best performance in terms of BER. In this section, the case for different pilot symbol power and data symbol power is studied, with a review of the relevant part of PSAM analysis in [15] conducted first.

Recall from Section 3.4, that according to [15], if the pilot power to data power is defined as q, then q is given by

$$q = \left|\tilde{b}^2\right| / (M-1) E\left[|b|^2\right]. \tag{5-1}$$

Then the optimal channel gain estimate results when $\mathbf{h}(k)$ satisfies $\mathbf{R}\mathbf{h}(k) = \mathbf{w}(k)$, as the optimization is in the minimum mean squared error sense with respect to $e^2(k)$, where $\mathbf{R}$

is the $K \times K$ autocorrelation matrix and $\mathbf{w}(k)$ is the $M-1$ length k covariance vector, which are given as

$$\mathbf{R} = \frac{1}{2} E\left[\mathbf{r}\mathbf{r}^{\dagger} \right] \tag{5-2}$$

and

$$\mathbf{w}(k) = \frac{1}{2} E\left[u^{*}(k) r \right] \tag{5-3}$$

respectively. Also recall that $\mathbf{R}_{ik}$ and $\mathbf{w}_i(k)$ are given by (3-16) and (3-17).

The corresponding minimum estimation error variance is given as

$$\sigma_e^2(k) = \sigma_u^2 - \sigma_v^2,$$

where $\sigma_u^2(k), \sigma_v^2(k), \sigma_{uv}^2(k)$ are given by (3-21) and (3-28). The BER for optimal $h(k)$ for BPSK is given by

$$P_b(k) = \left(1 - \rho(k)\right)/2, \tag{5-4}$$

where

$$\rho(k) = \frac{\mathbf{w}^{\dagger}(k)\mathbf{R}^{-1}\mathbf{w}(k)}{\sqrt{\left(\sigma_u^2 + 1\right)\mathbf{w}^{\dagger}(k)\mathbf{R}^{-1}\mathbf{w}(k)}}. \tag{5-5}$$

In order to optimize the power allocation in terms of minimizing P_b, $P_b(k)$ should be minimized; this is the same as maximizing $\rho(k)$, as $P_b(k)$ is a linear function of $\rho(k)$. Thus, in order to maximize $\rho(k)$, the following expressions are necessary

$$\begin{aligned} &\frac{\partial \rho(k)}{\partial q} = 0 \\ &\frac{\partial^2 \rho(k)}{\partial q^2} < 0 \end{aligned}. \tag{5-7}$$

However, $\rho(k)$ is expressed in terms of $\mathbf{w}(k)$, $\mathbf{R}$ and σ_u^2, all of which are functions of q. This makes it very difficult to find a closed form solution for factor q, for a specific k. In addition, $P_b(k)$ is dependent on k, and P_b is the mean value of the sum of $P_b(k)$ for different k in an estimation block. From the above discussion, it can be seen that a closed form expression to optimize q in terms of minimal P_b is hard to get.

The following subsection details the performance of unbalanced power allocation between the pilot symbol and data symbol by simulations, with basic results having been verified in accordance with [15].

5.1.1 UNBALANCED POWER ALLOCATION PERFORMANCE

Defining E_b as the bit energy, E_d as the energy for data symbol, E_p as the energy for pilot symbol and E_{avg} as the average energy, then

$$E_{avg} = E_b(M-1)/M\,. \tag{5-8}$$

The ratio of data symbol energy to average energy is defined as

$$\alpha = E_d / E_{avg}\,, \tag{5-9}$$

with α subject to the constraint $\alpha \le 1$. Accordingly, the data symbol and pilot symbol energies can be defined as

$$E_d = \alpha E_b(M-1)/M \tag{5-10}$$

and

$$E_p = E_b(M-1) - E_b\alpha(M-1)^2/M\,, \tag{5-11}$$

where E_p and E_d satisfy

$$E_p + (M-1)E_d = (M-1)E_b\,. \tag{5-12}$$

For the balanced power allocation for PSAM, $E_d = E_p = E_{avg}$. From (5-8) to (5-12), it is easy to derive the factor q in [17] as

$$
\begin{aligned}
q &= \frac{r}{(M-1)E_d} \\
&= \frac{E_b(M-1) - E_b\alpha(M-1)^2/M}{\alpha E_b(M-1)^2/M} \\
&= \frac{M-\alpha(M-1)}{\alpha(M-1)}
\end{aligned}
\tag{5-13}
$$

Then, using (5-13) and (3-14) to (3-30), $P_b(k)$ and P_b can be easily calculated.

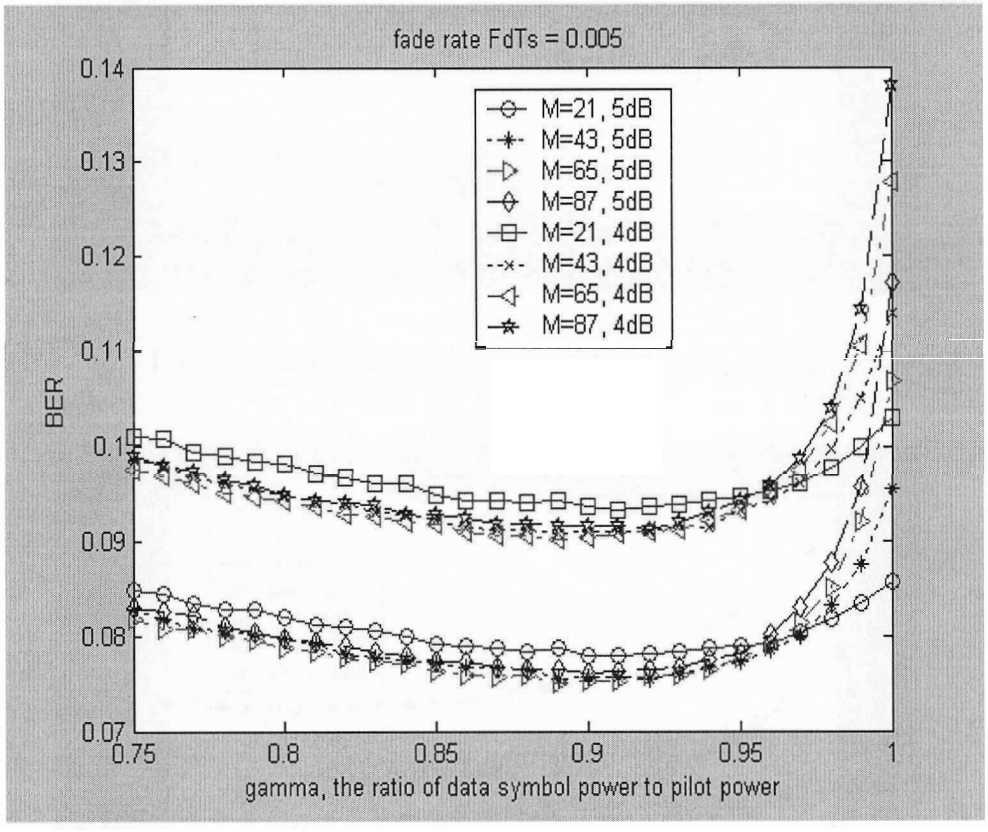

Figure 5.1 Modified PSAM for fade rate $f_d T_s = 0.005$

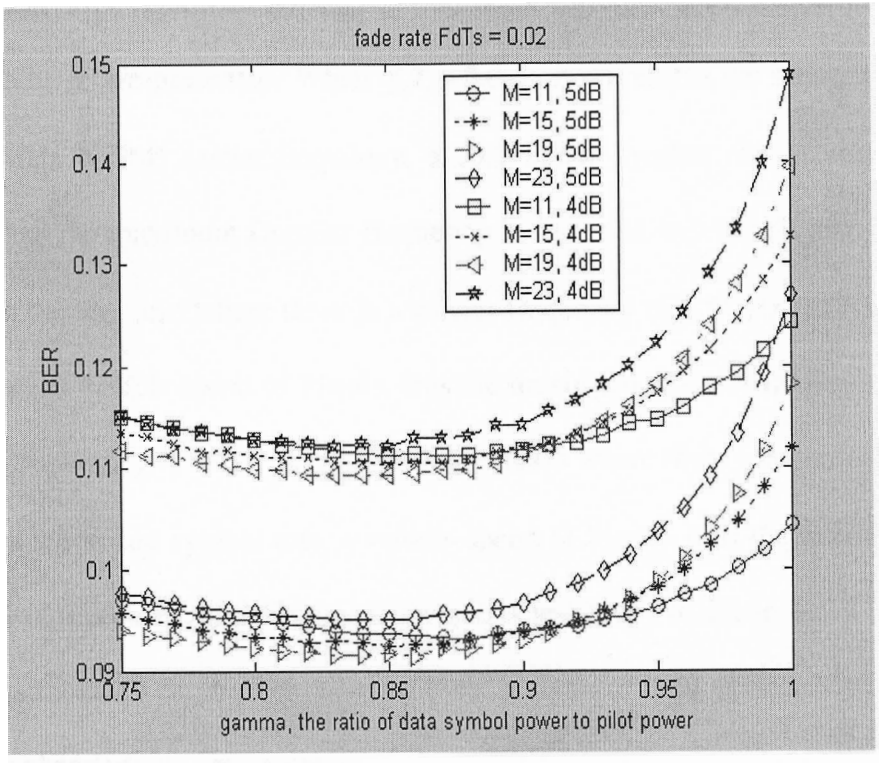

Figure 5.2 Modified PSAM for fade rate $f_d T_s = 0.02$

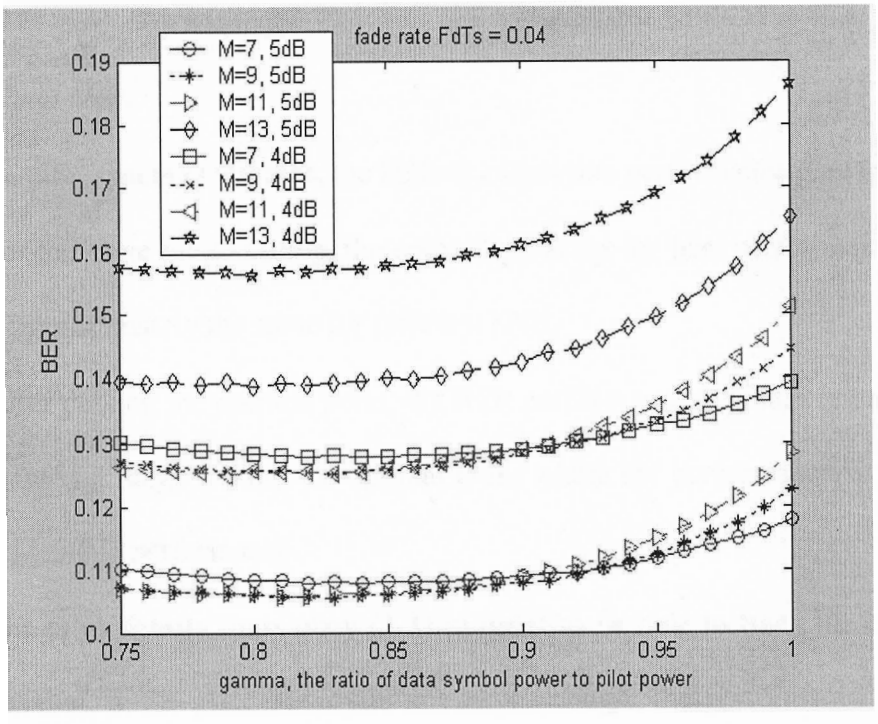

Figure 5.3 Modified PSAM for fade rate $f_d T_s = 0.04$

In the Figure 5.1-Figure 5.3, simulation results for different SNR, different pilot spacing and different α are presented. When $f_d T_s = 0.005$, it represents the scenario where there is a 900MHz (GSM) carrier frequency, a 19.2 kbaud symbol rate, a vehicle speed of 71mph; thus the maximum Doppler frequency in this case is 95Hz. When $f_d T_s = 0.02$, it represents the scenario where there is a carrier frequency of 1.9GHz (PCS), a 9.6 kbaud symbol rate, a vehicle speed of 71m/h; thus the maximum Doppler frequency in this case is 200Hz. Finally, $f_d T_s = 0.04$ represents the scenario where there is a carrier frequency of 3.8GHz, a 9.6 kbaud symbol rate, a vehicle speed of 71m/h; thus the maximum Doppler frequency in this case is 401 Hz, a value which is less likely to occur.

From the simulation results and the close form analysis of BER performance in [17], the following observations can be made:

- As α decreases from 1, the BER decreases. After reaching an optimal point, the BER gradually increases, however the value is still better than that for the equal power case.
- As pilot spacing increases, the highest achievable performance gain increases.
- For the same pilot spacing, the optimal point for the best performance achievable is approximately the same for different SNR.
- After passing the optimal point, the BER performance degrades rather slowly and there is a range around the optimal point where the performance is near the best achievable performance.
- The pilot density must obey (3-3) in order to be able to track the change in the channel. If the pilot density is not large enough, the performance will degrade

significantly, as shown in Figure 5.3. However, in this case the performance gain with the unbalanced scheme still persists.

As α decreases from 1, the data symbol power decreases while the pilot symbol power increases. Hence, there is a performance gain resulting from improved channel estimation due to the increase in pilot symbol power. Note that this gain is in despite of a performance loss from decreased data symbol power. As α decreases, if the performance gain is larger than the performance loss, the BER continues to decrease until it reaches an optimal point where the performance gain is equal to the performance loss. If α decreases further, the performance gain will be smaller than the performance loss, resulting in a gradual increase in the BER.

As shown in Figures 5.1 and 5.2, a gain of more than 1 dB can be achieved by optimally allocating different power levels to pilot symbols and data symbols. Note that turbo coding is robust to random errors, however burst errors will significantly degrade the performance. In this case, errors that occur after the initial decoding may play an important role in the BER for later iterations in the turbo coded system, as they may contain burst errors. Since unbalanced power allocation for pilot symbols and data symbols is effective in reducing the initial BER, this scheme can also be applied to the iterative channel estimation and decoding scheme.

5.2 UNBALANCED POWER ALLOCATION FOR ITERATIVE ESTIMATION AND DECODING

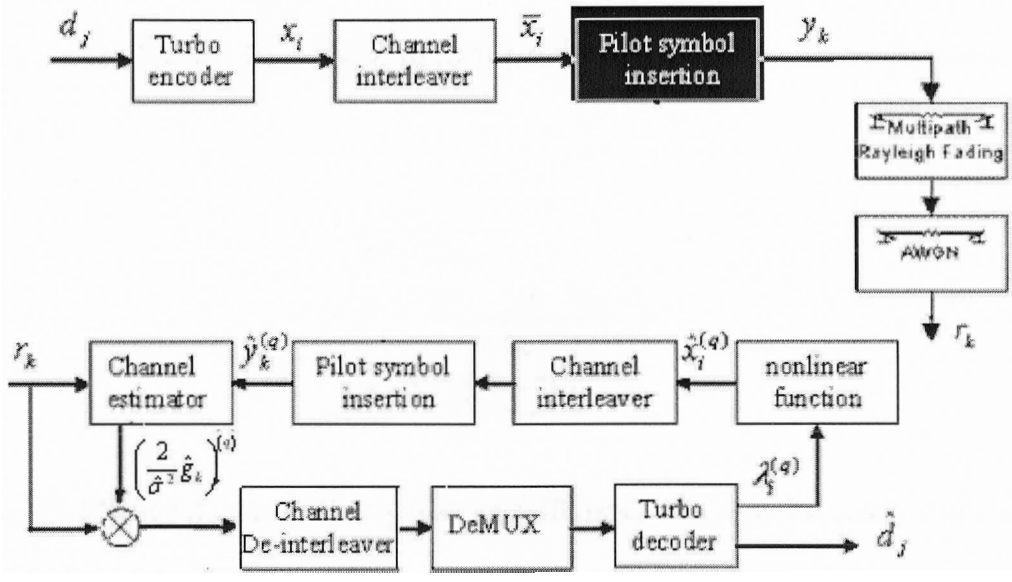

Figure 5.4 Unbalanced power allocation for iterative estimation and decoding [14]

The unbalanced power allocation scheme for iterative estimation and decoding is shown in Figure 5.4. The block that is different from the previous work is highlighted, where the pilot spacing and pilot power will be increased.

Since different power is allocated to pilot symbols and data symbols, define E_b as the bit energy, E_p as the energy for pilot symbol, E_d as the energy for data symbol, r is the effective code rate and E_{avg} as the energy when each pilot symbol and data symbol have the same power, which is given as $E_{avg} = rE_b\ E_P$. Then E_d and E_p can be expressed as

$$E_d = \alpha r E_b \tag{5-14}$$

and

$$E_p = (M-1)rE_b - (M-2)\alpha rE_b \ , \tag{5-15}$$

where E_d and E_p satisfy

$$E_p + (M-2)E_s = (M-1)rE_b \, . \tag{5-16}$$

Similar to Section 5.1, from (5-14) to (5-16), the ratio of pilot power to data power can be derived as

$$\begin{aligned} q &= \frac{E_p}{(M-2)E_d} \\ &= \frac{(M-1)rE_b - (M-2)\alpha rE_b}{\alpha rE_b} \\ &= \frac{(M-1)-(M-2)\alpha}{\alpha} \end{aligned} \ . \tag{5-17}$$

Using (5-13) and (3-4) to (3-19), filter coefficients $\mathbf{h}(k)$ can be easily calculated, as follows.

The receiver has a feedback path, where the sequence $\{r_k\}$ is first applied to the channel estimation algorithm described in Section 3.4. After the qth ($q \geq 1$) iteration of turbo decoding using the MAP decoder [2], the product of each received signal r_k and $\hat{x}_k^{(q)}$, which is the transformation of the decoded data symbol vector $\hat{d}_k^{(q)}$, becomes essentially a pilot symbol. The algorithm computes the estimate of fading gain $\{\hat{g}_k^{(q)}\}$ according to $\hat{g}_k^{(q)} = \mathbf{h}(k)\hat{\mathbf{y}}^{(q)}$.

The optimal coefficients $\mathbf{h}(k)$ satisfy $\mathbf{R}\mathbf{h}(k) = \mathbf{w}(k)$, and the optimization is in the minimum mean-squared error sense, with respect to $\left(g_k^{(q)} - g_k\right)^2$. Recall $\mathbf{R}$ and $\mathbf{w}(k)$ are defined in eqs. (3-24) and (3-35). Similarly, the specific components in R and W were

given by eqs. (3-26) and (3-27), the coefficient q in (3-26) and (3-27) is defined in (5-17). Each element $\hat{y}_k^{(q)}$ in vector $\hat{\mathbf{y}}_k^{(q)}$ is expressed as [33]

$$\hat{y}_k^{(q)} = r_k \hat{x}_k^{(q)}, \quad \hat{x}_k^{(q)} = \tanh(\hat{d}_k^{(q)}/2). \tag{5-19}$$

The noise variance $\hat{\sigma}^2$ is calculated as the sample variance of $z_k^{(q)}$ in the following formula:

$$z_k^{(q)} = r_k - \hat{g}_k^{(q)} \hat{d}_k^{(q)}, \tag{5-20}$$

where r_k is the received signal, $\hat{g}_k^{(q)}$ is the estimated channel response, and $\hat{d}_k^{(q)}$ is the tentative decision of data symbols in the qth iteration.

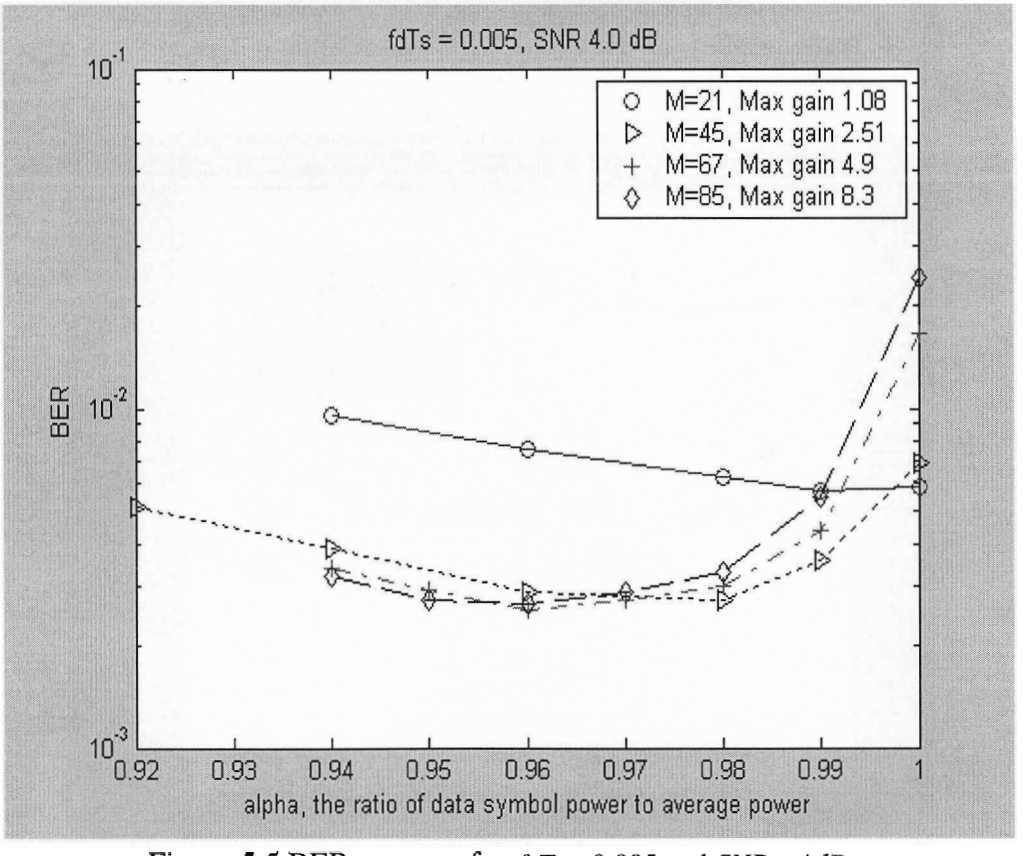

Figure 5.5 BER versus α for $f_d T_s = 0.005$ and $SNR = 4dB$

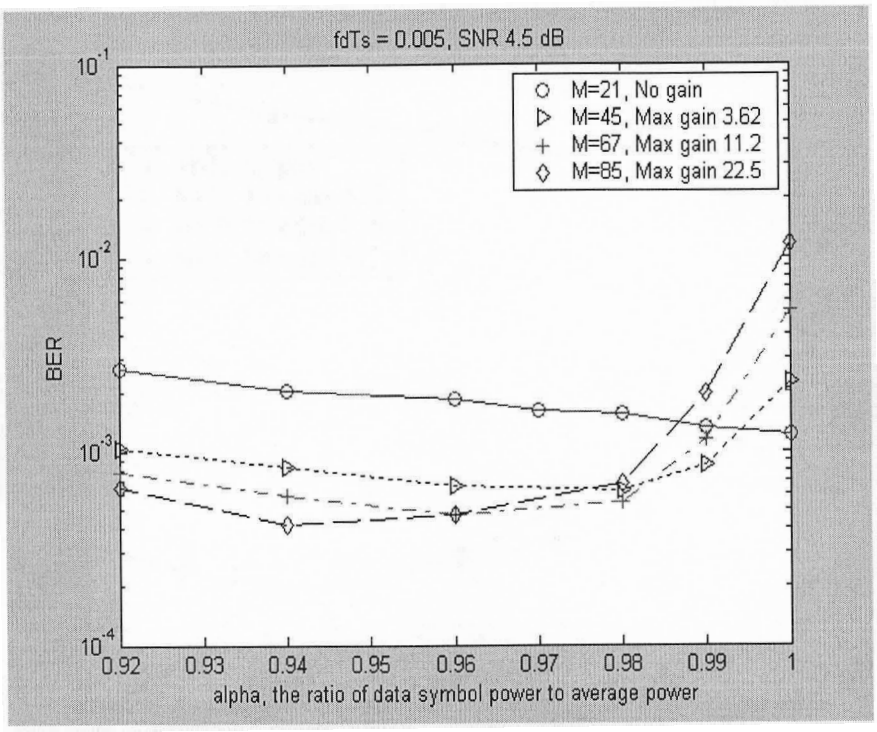

Figure 5.6 BER versus α for $f_d T_s = 0.005$ and $SNR = 4.5dB$

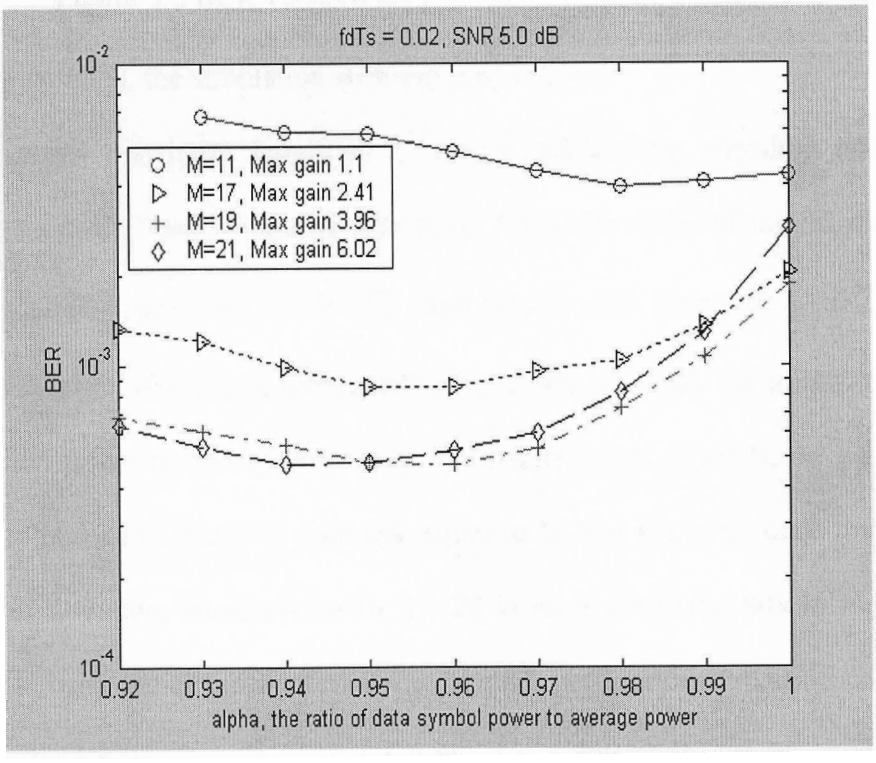

Figure 5.7 BER versus α for $f_d T_s = 0.02$ and $SNR = 5dB$

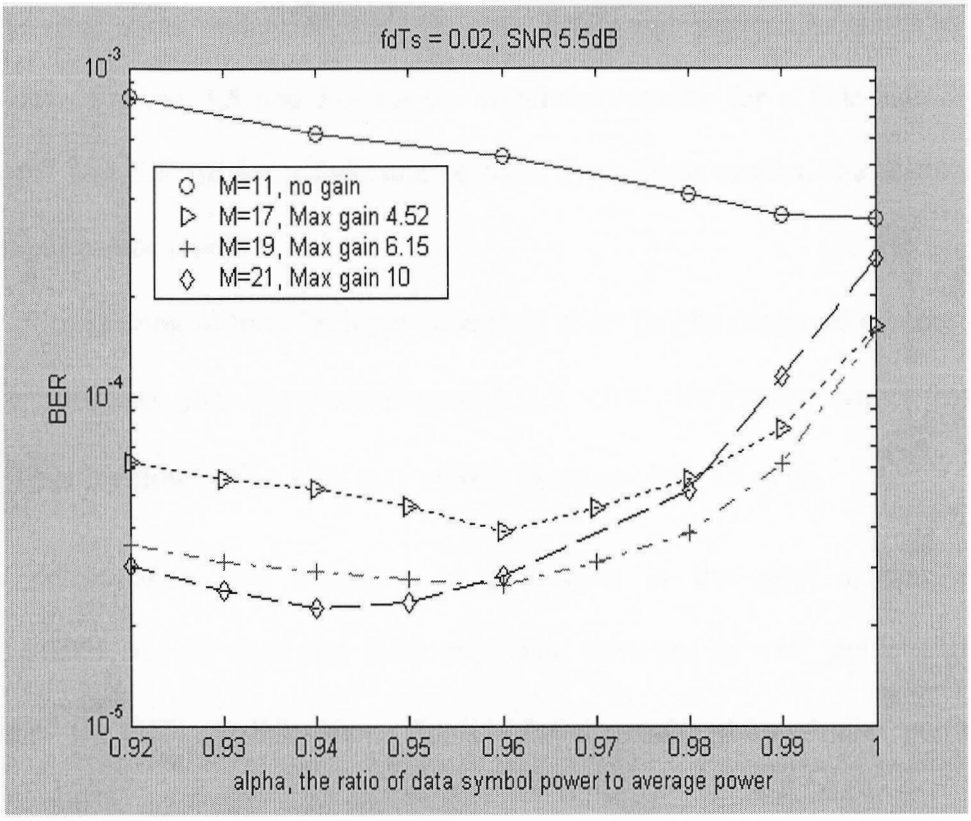

Figure 5.8 BER versus α for $f_d T_s = 0.02$ and $SNR = 5.5dB$

To compare with [5], the simulation scenario is as follows.

The turbo codes constraint length is 3, which reduces the encoding and decoding complexity by half, when compared with [14]. The turbo codes of rate ½ are generated with code feedback generator $1+D+D^2$, and feedforward generator $1+D^2$, where the trellis of the upper encoder is terminated with 2 tail bits, and the trellis of the lower encoder is left un-terminated. Considering the inserted pilot symbols, the parity bits are punctured to make the effective code rate equal to ½. For the turbo code interleaver, an $L = 1250$ bit *S*-random interleaver with $S = 20$ is used, while the 50×50 block channel interleaver is used for channel interleaving. As the performance of turbo coded system converges after 8 iterations, we use 8 iterations in our simulation.

Extensive simulations under different fade rates, SNRs and power allocations have been carried out. Figures 5.5 and 5.6 are the simulation results for a fade rate of 0.005. Figures 5.7 and 5.8 are for a fade rate of 0.02. From these results, the following key observations can be made.

- The pilot spacing M must be large enough in order for the proposed scheme to offer the performance gain. For fast fading with $f_d T_s = 0.02$, the spacing ranges from 13 to 23, while for slow fading with $f_d T_s = 0.005$, it ranges from 23 to 96.

- The performance gain increases in proportion to the pilot spacing M. For $f_d T_s = 0.005$, $M = 21$ does not offer any gain, however $M = 85$ yields a gain that reduces the BER by 8.3 times. The simulation results show similar performance behaviour for other fade rates and SNRs.

- With appropriate pilot spacing, the BER first decreases gradually as α decreases, reaching an optimal (lowest BER) point; after this point, the BER increases.

- With pilot spacing fixed, the optimal power ratio α remains almost the same even for different SNRs.

- The BER performance for different fade rates behaves similarly as α decreases.

From the simulation results, we see that for unbalanced power allocation, both the pilot spacing and pilot symbol power can be increased without much effect on the SNR of the data symbols. For example, when the pilot spacing is 19, if α decreases from 1 to 0.99, the data symbol SNR is decreased by 10*log(1/0.99) dB. This is a value of only 0.0436 dB, however the pilot symbol SNR is increased to 10*log(1.18/1) dB, or 0.7188 dB. When pilot spacing is 65, the pilot symbol SNR is further increased to 10*log10(1.64/1)

dB, or 2.1484 dB, which is significant. This means that a small decrease in the data symbol SNR is traded for a large change in the pilot symbol SNR. Thus, increased pilot spacing and pilot power lead to a reduced turbo code rate and improved initial channel estimation, which in turn means increased parity protection, increased diversity and reduced raw data errors. When the pilot spacing is large enough, the gain from the increased parity protection, increased diversity and decreased raw data errors is larger than the loss caused by increased pilot spacing. There is a certain gain in the best achievable BER performance with unbalanced power allocation, when compared with the conventional scheme that uses dense pilot symbols and equal power for pilot symbols and data symbols. We also notice that there is no performance gain if the pilot spacing is not large enough; however, the performance degrades gradually rather than significantly.

The proposed scheme is applicable to both slow and fast fading channels. If the pilot symbol rate is greater than twice the highest frequency of the channel gain, then the best achievable BER performance improves as the pilot spacing increases. However, if the pilot symbol rate is not fast enough to track a change in the channel gain, the best achievable performance degrades significantly due to burst errors. In this case though, the performance gain still persists with unbalanced power allocation.

Figure 5.9 shows the performance comparison of the unbalanced scheme proposed in this thesis, the balanced scheme and the scheme proposed in [14]. It is obvious that the BER performance of the proposed unbalanced power allocation with increased pilot spacing is superior to that of the balanced power allocation with increased pilot spacing. In addition, the balanced power allocation with increased pilot spacing outperforms the scheme proposed in [14].

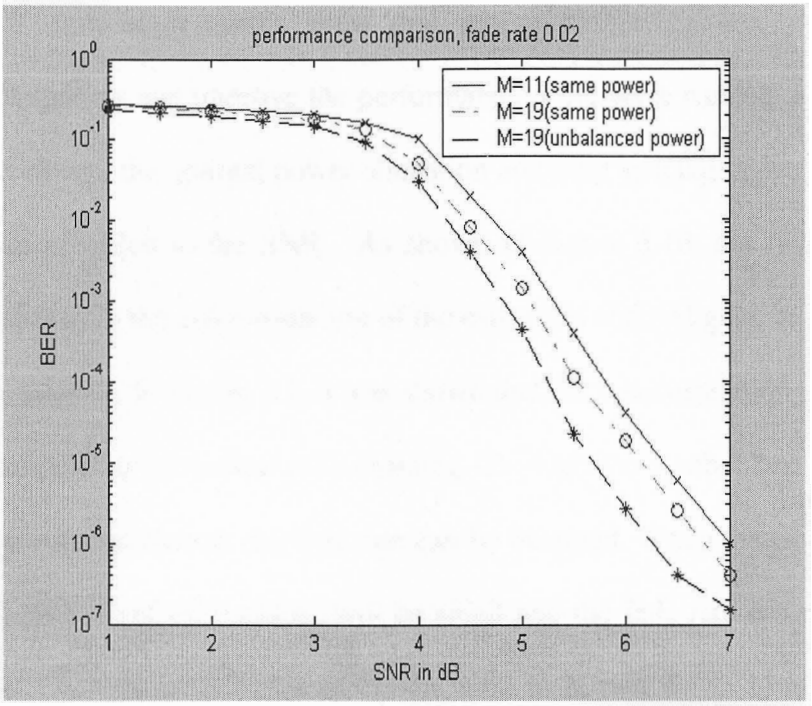

Figure 5.9 Performance comparison for $f_d T_s = 0.02$

5.3 ADAPTIVE SCHEME FOR UNBALANCED POWER ALLOCATION FOR ITERATIVE CHANNEL ESTIMATION

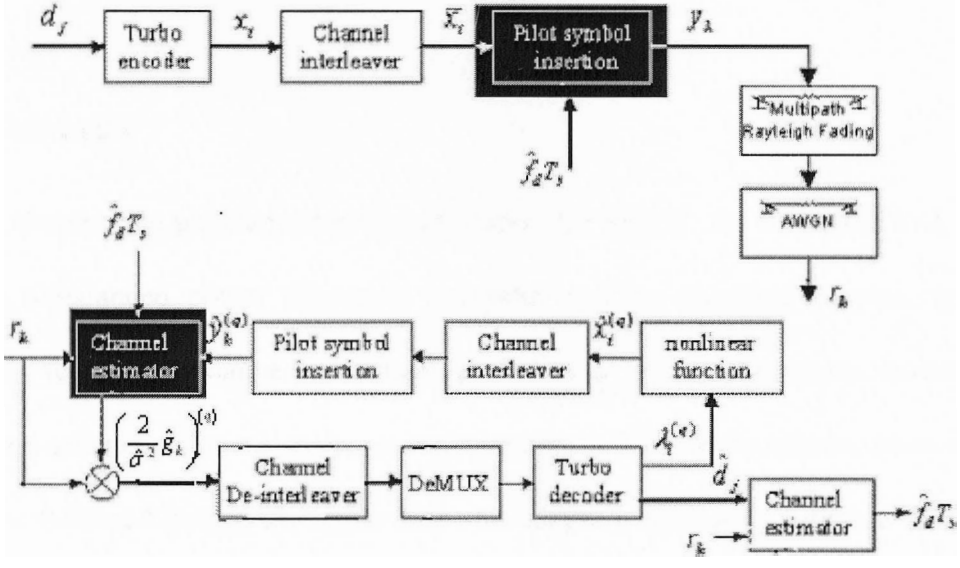

Figure 5.10 Adaptive unbalanced power allocation scheme

In section 5.2, simulation results show that the unbalanced power allocation with increased pilot spacing can improve the performance of iterative channel estimation and decoding. In addition, the optimal power allocation and pilot spacing depends on the fade rate, and is not sensitive to the SNR. As shown in Figure 5.10, the fade rate can be estimated according to the autocorrelation of the estimated channel gain, at the end of the last decoding iteration. In Figure 3.1, it was shown that the autocorrelation of the channel gain has the property of periodical zero-crossing. Through the symbol rate and the zero-crossing of the autocorrelation, the fade rate can be obtained. When the SNR is not very low, the BER after iterative decoding will be small and the fade rate estimation will be fairly accurate. The estimated fade rate can be used to in two ways: 1) to transmit this information to the transmitter in order to decide the optimal power allocation and pilot spacing, assuming a reliable backward signalling channel, and 2) to update the filter coefficients for channel gain estimation according to (3-16) and (3-17), in order to minimize coefficients mismatch. In this way, the system performance can be improved.

5.3 SUMMARY

In this chapter, the unbalanced power allocation for PSAM was discussed first, followed by the unbalanced power allocation and pilot spacing adjusting scheme, which was extended to iterative channel estimation and decoding. Simulation results showed that the proposed scheme offers a significant improvement in BER performance, when compared with the balanced power allocation case. An adaptive scheme to estimate the fade rate, and to adjust power allocation and pilot spacing was also discussed.

CONCLUSION

In this thesis, two schemes related to turbo codes have been proposed. The first scheme employs pilot symbols and metric modification in order to improve the decoding performance of turbo codes, without changing the data rate. This approach increases the computational complexity only slightly, and is therefore suitable for hardware implementation. The second scheme proposed increases the pilot spacing and allocates different power levels to the pilot symbols and data symbols, while keeping the overall power expense unchanged. This scheme reduces the effective turbo code rate, and improves the initial channel estimation with reduced raw data errors. The reduction in raw data errors benefits the iterative channel estimation and decoding through reduced initial burst errors. The resulting performance gain was shown to be significant when compared to the conventional scheme, which uses dense pilot symbols and equal power for pilot symbols and data symbols to perform iterative channel estimation and decoding.

FUTURE WORK

Future work related to the first scheme, is to study the extension of pilot assisted codec to bursty and fading channels. The pilot symbol approach presented here can also be extended to other turbo/SOVA related topics, such as trellis-coded modulation. For the second scheme, future work may be two parts: 1) to theoretically analyze the optimization of power allocation and pilot spacing in terms of maximizing the average channel capacity over flat-fading channel, and 2) to apply the scheme to a turbo-coded Orthogonal Frequency Division Multiplexing (OFDM) system, over frequency-selective fading channels or two-dimensional (time-and frequency-selective) fading channels.

BIBLIOGRAPHY

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, Oct. 1996 On page(s): 1261 - 1271 Volume: 44 , Issue: 10

[2] J. G. Proakis, Digital Communications, 4th edition. New York: *McGraw-Hill*, 2000

[3] M. Bossert, Channel coding for telecommunications, New York : *Wiley*, 1999

[4] 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. 248-287, Mar. 1974

[5] J. Hagenauer and P. Hoehe, "A Viterbi algorithm with soft decision outputs and its applications," in Proc. *GLOBECOM 89*, Nov. 1989, pp. 1680-1686

[6] L. C. Perez; J. Seghers and D. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Tran. Inform Theory*, vol. 42, pp. 1698-1709, Nov. 1996

[7] A. S. Barbulescu and S. S. Pietrobon, "Interleaver design for turbo codes," *Electron. Lett.*, vol. 30, no. 25, pp. 2107-2108, Dec. 1994

[8] A. K. Khandani, "Design of the turbo-code interleaver using Hungarian method," *Electron. Lett.*, vol. 34, no. 1, pp. 63-65, Jan. 1998

[9] K. S. Andrews; C. Heegard, and D. Kozen, "Interleaver design methods for turbo codes," in *Proc. 1988 IEEE Int. Symp. Information Theory* Cambridge, MA: MIT, Aug. 1998, p. 420

[10] G. Battail, "Ponderation des symboles decodes par l'algorithme de Viterbi," *Ann. Telecommun.*, vol. 42, pp. 31-38, Jan. 1987

[11] L. Papke and P. Robertson, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc. ICC96*, pp. 102-106.

[12] L. Lin and R. S. Cheng, "Improvements in SOVA-based decoding for turbo codes," in *Proc. ICC97*, June 1997, pp. 1473-1478

[13] E. K. Hall and S. G. Wilson, "Design and analysis of turbo codes on Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, Volume: 16 , Issue: 2, Feb.1998 Pages:160 – 174

[14] M. C. Valenti and B. D. Woerner, "Iterative channel estimation and decoding of pilot symbol assisted turbo codes over flat-fading channels," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1697-1705, Sept. 2001

[15] D. Divsalar and M. K.Simon, "Multiple-symbol differential detection of MPSK," *IEEE Trans. Commun.*, vol. 38, pp. 300-308, Mar. 1990

[16] P. Hoeher and J. Lodge, "Turbo DPSK: Iterative differential PSK demodulation and channel decoding," *IEEE Trans. Commun.*, vol. 47, pp. 837-843, June 1999

[17] J. K. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *[mobile radio], IEEE Trans. Vehicular Technology*, Volume: 40, Issue: 4, Nov. 1991 Pages: 686 – 693

[18] C. Komninakis and R. D. Wesel, "Joint iterative channel estimation and decoding in flat correlated Rayleigh fading," *IEEE J. Select. Areas Commun.*, Volume:19, Issue: 9, Sept. 2001 Pages: 1706 – 1717

[19] E. Gamal and H. Geraniotis, "Iterative channel estimation and decoding for convolutionally coded anti-jam FH signals," *IEEE Trans. Commun.*, Volume: 50, Issue: 2, Feb. 2002 Pages: 321 – 331

[20] J. H. Kang and W. E. Stark, "Iterative estimation and decoding for FH-SS with slow Rayleigh fading," *IEEE Trans. Commun.*, Volume: 48, Issue: 12, Dec.2000 Pages: 2014 – 2023

[21] T. Abe; S. Tomisato and T. Matsumoto; "A MIMO turbo equalizer for frequency-selective channels with unknown interference," *IEEE Trans. Vehicular Technology,* Volume: 52, Issue: 3, May2003, Pages: 476 – 482

[22] S. Bhashyam and B. Aazhang," Multiuser channel estimation and tracking for long-code CDMA systems," *IEEE Trans. Commun.*, Volume:50, Issue: 7, July 2002, Pages: 1081 – 1090

[23] A. Lampe, "Iterative multiuser detection with integrated channel estimation for coded DS-CDMA," *IEEE Trans. Commun.*, Volume: 50 , Issue: 8, Aug. 2002 Pages: 1217 – 1223

[24] K. Higuchi; A. Fujiwara and M. Sawahashi, "Multipath interference canceller for high-speed packet transmission with adaptive modulation and coding scheme in W-CDMA forward link," *IEEE J. Select. Areas Commun.*, Volume: 20, Issue: 2, Feb. 2002 Pages: 419 – 432

[25] F. Fossorier; S. Burkert; S. Lin and J. Hagenauer, "On the equivalence between SOVA and max-log-MAP decoding," *IEEE Commun. Letters*, vol.2, pp.137-139, May 1998

[26] G. Battail, "Ponderation des symboles decodes par l'algorithme de Viterbi," *Ann. Telecommun.*, vol. 42, pp. 31-38, Jan. 1987

[27] L. Lin and R. S. Cheng, "Improvements in SOVA-based decoding for turbo codes," in *Proc. ICC97*, June 1997, pp. 1473-1478

[28] J. Chen; M. P. C. Fossorier; S. Lin; C. Xu; "Bi-directional SOVA decoding for turbo-codes," *IEEE Commun. Letters*, Volume:4, Issue:12, Dec.2000 Pages: 405 – 407

[29] S. Benedetto and G. Montorsi; "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform Theory*, Volume: 42, Issue: 2, March 1996 Pages: 409 – 428

[30] S. Benedetto; D. Divsalar; G. Montorsi; F. Pollara; "Analysis, design, and iterative decoding of double serially concatenated codes with interleavers," *IEEE J. Select. Areas Commun*, Volume: 16, Issue: 2, Feb. 1998 Pages: 231 - 244

[31] D. Divsalar; S. Dolinar and F. Pollara, "Transfer Function Bounds on the Performance of Turbo Codes," TDA Progress Report 42-122, August 15, 1995

[32] S. Riedel, "Symbol-by-symbol MAP decoding algorithm for high-rate convolutional codes that use reciprocal dual codes," *IEEE J. Select. Areas Commun,* Volume: 16, Issue: 2, Feb. 1998 Pages: 175 – 185

[33] D. J. Young,; N. C. Beaulieu, "The Generation of Correlated Rayleigh Random Variates by Inverse Discrete Fourier Transform," *IEEE Trans. Commun.*, Volume: 48, Issue: 7, July 2000, Pages: 1114 – 1127

[34] J. Craig, "A new, simple and exact result for result for calculating probability for two-dimensional signal constellations," in *Proc. Milcom*, p. 25.5.1, 1991

[35] M. Sandelll; C. Luschi; P.Strauch and R. Yan, "Iterative channel estimation using soft decision feedback," *Proc. IEEE GLOBECOM,* Sydney, Australia, Nov. 1998