View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Scholarship at UWindsor

University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

1-1-2006

Iterative decoding with imperfect channel estimation for wireless systems.

Yibo Zhou
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

Zhou, Yibo, "Iterative decoding with imperfect channel estimation for wireless systems." (2006). *Electronic Theses and Dissertations*. 7146.
https://scholar.uwindsor.ca/etd/7146

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Iterative Decoding with Imperfect Channel Estimation for Wireless Systems

by

Yibo Zhou

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through Electrical Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada
2006

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-42337-0
Our file Notre référence
ISBN: 978-0-494-42337-0

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

© 2006 Yibo Zhou

All Rights Reserved. No Part of this document may be reproduced, stored or otherwise retained in a retreival system or transmitted in any form, on any medium by any means without prior written permission of the author.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Abstract

In this thesis, we propose a new iterative algorithm for turbo decoding with integrated channel estimation for Rayleigh flat-fading channels. The design of the proposed algorithm is based on a new turbo decoding metric for binary phase-shift keying (BPSK) signaling on fast fading Rayleigh channels with noisy channel estimates. The algorithm consists of a channel estimator that can reduce the error variance of the channel estimate iteratively by using the soft extrinsic information output from the turbo decoder. The extrinsic information generated from the turbo decoder has some *a priori* information of the transmitted data symbols which can be used to refine the channel estimate after each iteration of decoding. The refined channel estimate is then fed back to the turbo decoder for the next iteration of decoding. The resulting iteration between the channel estimator and the turbo decoder using the new metric, with intermediate exchange of soft channel-symbol information, yields very impressive results. Simulation results verify that the proposed algorithm can outperform conventional algorithms significantly and even its performance can approach that of a turbo decoding algorithm with perfect knowledge of the channel.

In most existing turbo decoding algorithms for fading channels, the received data symbols are multiplied by the complex conjugate of an estimate of the channel gain before turbo decoding without any mathematical justification. However, in this thesis, we prove that the result of this kind of preprocessing (or filtering) operation on the received signal to compensate the effect of fading distortion can be viewed as a method to obtain linear

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

minimum-mean-square-error (MMSE) estimates of the transmitted data symbols.

We also study the effect of signal-to-noise ratio (SNR) mismatch on the bit-error rate (BER) performance of turbo decoding algorithms. According to the previous results obtained by other researchers, turbo-decoding with the Max-Log-MAP decoder is independent of SNR for Rayleigh fading channels when the channel is assumed to be perfectly known to the receiver. Using such an assumption, the Rayleigh fading channel may then be viewed as an additive white Gaussian noise (AWGN) channel conditioned on the fading coefficient. Since in practical communication systems, the channel information is not available at the receiver, the channel must be estimated at the receiver. Because of the low signal to noise ratios typical of turbo coded systems, it is difficult to obtain perfect estimates of the fading coefficients. Thus, we show that the Max-Log-MAP decoder is sensitive to SNR mismatch over fading channels with noisy channel estimates. Also, it is shown that the Max-Log-MAP decoder is less sensitive to SNR mismatch than the MAP or Log-MAP decoder.

Finally, we propose a new SNR mismatch model to examine the sensitivity of turbo decoding algorithms to SNR mismatch for systems with an SNR estimator. This model is more practical than the conventional one. Under this model, we still can verify that the Max-Log-MAP decoder is sensitive to SNR mismatch over fading channels.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Acknowledgement

I would like to thank all who made this work possible: my advisor Dr. Behnam Shahrrava, for thousands of suggestions and guidelines to maintain overall high standards; my internal reader Dr. Kemal Tepe and my external reader Dr. Scott Goodwin. I would also like to thank my parents who have always supported me with my graduate study. Lastly, I thank my dearest girl friend Yanjie Mao, who is always by my side and has helped me to prepare the simulations and thesis.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Contents

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

List of Figures

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 1

Intoduction

1.1 Turbo (iterative) decoding algorithms

Turbo codes with iterative decoding algorithm (MAP algorithm) were introduced by Berrou and Glavieux in 1993 [3]. After introduction of turbo codes, a lot of research has been conducted on the structure of the turbo encoder and turbo decoder.

In [6], the trellis termination of turbo encoder and extrinsic information generated by the turbo decoder have been described thoroughly by Robertson. The same author in [7] introduced two kinds of the *maximum a posteriori* (MAP) algorithms, the Log-MAP and Max-Log-MAP decoding algorithms. The performance of the MAP decoding algorithm has been proved to be remarkably well over additive white Gaussian noise (AWGN) channels [3]. Since the MAP algorithm is very complex, the Log-MAP and Max-Log-MAP decoding algorithms have been introduced to solve this problem. The Log-MAP algorithm operates in the logarithmic domain and reduces the number of additions and multiplications in the decoding process while its performance is the same as that of the MAP decoder. The Max-Log-MAP algorithm further reduces the computational complexity by approximating all the non-linear functions in the Log-MAP algorithm with the linear functions. The performance of the Max-Log-MAP algorithm is suboptimal compared to that of the Log-MAP algorithm.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In [5], Sklar summarized the ideas behind turbo decoding. He discussed the component codes and the interleaver in the turbo encoder. He also derived the channel reliability factor L_c for turbo decoding algorithms over AWGN channels. It was shown that the channel reliability factor depends on SNR in this case.

The performance of turbo decoding algorithms over fading channels has also been studied since turbo decoding algorithms have been introduced. In [16], Barbulescu assumed the fading amplitude and phase are known at the receiver and the channel reliability factor L_c for this case depends on the SNR and the fading amplitude. In [18], Hall and Wilson assumed only the phase of the fading is known at the receiver and studied the decoding process. In [19], the same authors studied the decoding process when the fading amplitude is assumed to be known at the receiver. In [12], Frenger derived a new turbo decoding metric over fading channels by considering the uncertainty of both fading amplitude and phase. Frenger also indicated that if the transmitted bits experience fading distortion with imperfect channel estimation, the channel reliability factor L_c depends on both SNR and the error variance of the channel estimate. In [12], using computer simulations, Frenger proved that turbo decoding based on his new metric outperforms the metric without considering the error variance of the channel estimate. He also showed that Log-MAP decoder can achieve better bit error rate (BER) performance with smaller error variance of channel estimate.

Turbo decoding over AWGN channels needs SNR [5], and turbo decoding over fading channels needs both SNR and channel fading factor [16]. Practically, neither SNR or fading factor are known at the receiver. A lot of research thus has been done on SNR and channel estimation schemes for turbo decoding. In [28] and [35], SNR and channel(fading) estimation schemes for turbo decoding have been discussed. All these estimation schemes can be classified into three categories: pilot-assisted channel estimation, blind channel estimation, and hybrid channel estimation. The pilot-assisted and blind channel estimation methods have been discussed in [36] and [37], respectively. The hybrid channel estimation scheme combines two schemes. Most of the proposed methods perform channel estimation without any feedback from the turbo decoder, [33] and [34]. In [23], an iterative channel estimation

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

scheme is introduced which uses the extrinsic information generated from the turbo decoder. Later, more work was done on the iterative channel estimation in [24].

1.2 A new algorithm for turbo decoding with channel estimates

In this thesis, we propose a new turbo decoding algorithm for BPSK signaling on Rayleigh flat-fading channels with noisy channel estimates.

Conventional channel estimators for turbo decoding have no feedback from the turbo decoder and ignore the extrinsic information generated from the turbo decoder. While, the extrinsic information can be fed back to the channel estimator to refine its previous estimate. In [23], Valenti proposed a turbo decoding algorithm with integrated pilot-assisted channel estimator without using the turbo decoding metric proposed by Frenger in [11]. In [23], Valenti used the channel reliability factor L_c derived by Barbulescu in [16] and tried to re-estimate the fading factor and thus derived a new L_c. However, the reliability factor derived by Barbulescu was based on the assumption that both the amplitude and phase of fading are perfectly known at the receiver. Since in practical communication systems, the channel information is not available at the receiver, the channel must be estimated at the receiver. Because of the low signal to noise ratios typical of turbo coded systems, it is difficult to obtain perfect estimates of the fading coefficients. Hence, for fading channels with noisy channel estimate, we will use the channel reliability factor L_c derived by Frenger in [11].

Then, using a modified version of the channel reliability factor L_c derived by Frenger, we propose a new iterative algorithm for turbo decoding with integrated channel estimation for Rayleigh flat-fading channels. Further, we use the hybrid channel estimation scheme in the proposed algorithm. For the first iteration, a pilot-assisted channel estimator is assumed to provide us with the initial channel estimate and its error variance. After each iteration of decoding, a blind linear MMSE (minimum mean square error) channel estimator refines the channel estimate and its error variance by using the extrinsic information fed

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

back from the turbo decoder. The refined channel estimate is then sent back to the Log-MAP turbo decoder for the next iteration of decoding. This way, in the blind channel estimator, we may reduce the error variance of the channel estimate iteratively by using the *a priori* information of the data bits buried in the extrinsic information. Finally, the resulting iteration between the channel estimator and the turbo decoder, with intermediate exchange of soft channel-symbol information, can improve the performance of both the channel estimator and the channel decoder.

1.3 Preprocessing (fading compensation) for turbo decoding

In practical systems, in order to undo the effect of fading, the received data symbols are multiplied by the complex conjugate of an estimate of the channel gain before turbo decoding, [44]. Since there has been no mathematical justification in the literature for this kind of fading compensation for systems without perfect knowledge of the channel, we try to come up with some justification. First, we derive a linear MMSE estimator to obtain optimal estimates of the transmitted data symbols with noisy channel estimates. Then, it is shown the linear MMSE estimator can be viewed as a normalized version of the preprocessing (fading compensation) operation used in most existing turbo decoding algorithms, [11]. Also, we show that the normalization has no effect on the BER performance of the turbo decoder.

1.4 Effect of SNR mismatch on BER performance

One requirement of the Log-MAP turbo decoder over fading channels is the knowledge of SNR and the error variance of channel estimate. Since the proposed turbo decoding algorithm composed of the Log-MAP decoder, both the channel SNR and the channel fading must be estimated. Adding any estimator to this algorithm increases the complexity of the algorithm. Therefore, any decoding algorithm that does not need perfect knowledge of SNR or fading factor is more desirable. The effect of SNR mismatch on the performance of turbo decoding algorithms over AWGN channels has been discussed in [29] by Jordan

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and Nichols. In [30] and [28], the effect of SNR mismatch over the fully interleaved channels has been discussed. Valenti and Woerner further discussed the effect of SNR mismatch over correlated fading channels in [27]. In [9], Worm indicated that Max-Log-MAP turbo decoder is insensitive to SNR mismatch over AWGN channels and fading channels. However, this result is valid for the Max-Log-MAP turbo decoding over AWGN channels. Whereas, for the Max-Log-MAP turbo decoding over fading channels, it was assumed the fading factors are known at the receiver. Using such an assumption, the Rayleigh fading channel may then be viewed as an AWGN channel conditioned on the fading coefficient. However, for fading channels with imperfect channel estimation, the error variance of channel estimate must be included in the channel reliability factor L_c, [12].

In this thesis, we study the sensitivity of the Log-MAP and Max-Log-MAP algorithms to SNR mismatch over fading channels by using the channel reliability factor L_c derived in [12]. We show that both the Log-MAP and Max-Log-MAP decoder are sensitive to SNR mismatch over fading channels when the channel is not perfectly known. However, we find the Max-Log-MAP decoder is less sensitive to the SNR mismatch than the Log-MAP decoder.

1.5 A New SNR mismatch model

At last, we also propose a new SNR mismatch model. The new SNR mismatch model is more practical than the conventional model used in the literatures. The conventional model considers all the data frames experience the same SNR mismatch offset at a given SNR. In the new model, we consider the SNR mismatch is a random variable and different frames experience different SNR mismatch. Under the new model, we can still verify that both Log-MAP and Max-Log-MAP decoder are sensitive to SNR mismatch over fading channels when the channel is not perfectly known.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1.6 Organization of the thesis

The organization of this thesis is as follows. In Chapter 2, the channel models are introduced. In Chapter 3, the turbo encoder and turbo decoding algorithms are described. In Chapter 4, we propose a new Log-MAP turbo decoding with integrated with channel estimation over fading channels. In Chapter 5, the preprocessing or fading compensation is discussed from the estimation theory point of view. In Chapter 6, the effect of SNR mismatch on the performance of turbo decoding algorithms is investigated. In Chapter 7, simulation results are presented and also we introduce a new model to study the effect of SNR mismatch on the BER performance of turbo decoding algorithms. Finally, conclusions and future research directions are given in Chapter 8.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 2

Channel Models

To design the channel estimator and analyze the effect of SNR mismatch on the performance of turbo decoding algorithms, we have to understand the channel that the transmitted data bits experiences. In this thesis, we need two kinds of channel models to discuss our contributions. In this chapter, we first introduce additive white Gaussian noise (AWGN) channel model, and then fading channel model.

2.1 AWGN channels

The additive white Gaussian noise (AWGN) channel model together with binary phase shift keying (BPSK) modulator is depicted as in figure 2.1. Where $d_k \in (0, 1)$ are the transmitted data bits, $x_k \in (-\sqrt{E_s}, \sqrt{E_s})$ are BPSK symbols, n_k are the additive white Gaussian noise and y_k are received symbols. In AWGN channel models, a AWGN noise n_k is added to the transmitted symbols x_k. Here, n_k are complex-valued and Gaussian distributed random variables. The mean and variance of n_k are

$$\begin{aligned} E[n_k] &= 0, \\ E[|n_k|^2] &= 2\sigma_n^2 = N_0. \end{aligned} \tag{2.1}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

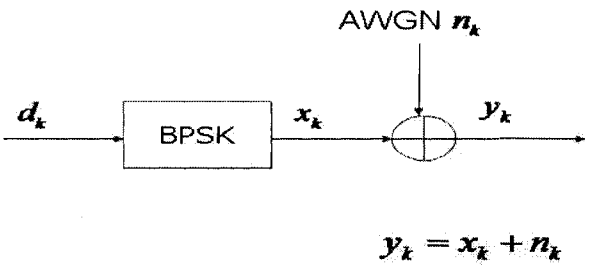

Figure 2.1: AWGN channel model

The signal to noise ratio (SNR) for this case is

$$\frac{E_b}{N_0} = \frac{E_s}{R \times 2\sigma_n^2}, \tag{2.2}$$

where R is the coding rate and $E_s = E_b \times R$. E_s is the transmitted symbol energy and E_b is the source bit energy. The received symbols y_k are

$$y_k = x_k + n_k. \tag{2.3}$$

For communication systems over AWGN channels, at the receiver, we need to know the signal to noise ratio. Normally, the symbol energy E_s is set to 1 and the coding rate R is known. The noise variance N_0 is the only value we need to estimate to get the SNR.

2.2 Fading channels

For wireless communications, the transmitted data bits will experience a much more complicated channel than AWGN channel. Due to the complex environments between the wireless communication terminals (buildings, forests and vehicles) and movement of the terminals, the transmitted data bits will be affected by unpredictable errors. Normally these errors are called channel fading. And the wireless channels are usually modeled as fading channels.

Channel fading can be classified into two categories:

- Large-scale fading
- Small-scale fading

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In [40], it has been mentioned *"Large-scale fading is caused by reflection, diffraction and scatting of the signals when they are obstructed by buildings, forests and vehicles"*. This type of fading usually reflects a long period of channel characteristics.

Small-scale fading, or simply fading, on the other hand reflects a short period of channel characteristics. Small-scale fading is caused by the multipath interference waves and the movement of the wireless terminals. The multipath interference waves are the different versions of the transmitted data bits arriving at the receiver at different times. The sum of the multipath waves at the receiver will result in a received signal which can varies widely in amplitude and phase. The movement of the terminals will also introduce channel errors. In this thesis, we only use the small-scale fading channel model. Below, we will only talk about the small-scale fading.

2.2.1 Type of small-scale fading

The type of small-scale fading depends on the two important factors in the wireless communication channels: multipath delay and Doppler spread. Multipath delay is introduced by multipath interference waves. Doppler spread is introduced by the movement of the wireless terminals. In this thesis, we only briefly introduce the fading types, for detail, see the book by Stüber, [40].

- Fading types due to multipath delay

Multipath delay causes the transmitted signal to experience either flat or frequency selective fading.

If the transmitted data bits period is greater than the multipath delay, the transmitted data bits will experience a flat fading. On the other hand, if the data bits period is shorter than the multipath delay, the data bits will experience a frequency selective fading.

- Fading types due to Doppler spread

Doppler spread causes the transmitted signal to experience either slow or fast fading.

If the signal bandwidth is much greater than the Doppler spread, the transmitted bits will experience a slow fading. If the signal bandwidth is smaller than or close to the Doppler spread, the transmitted bits will experience a fast fading.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

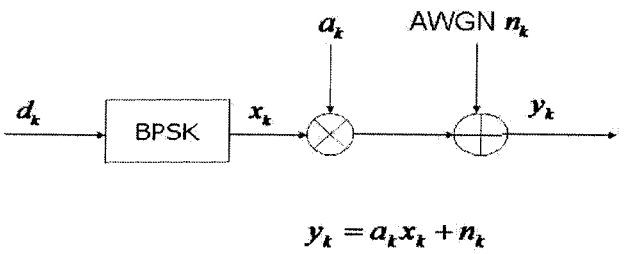

Figure 2.2: Fading channel model

Two kinds of fading classification are independent.

2.2.2 Fading channel model in this thesis

The fading channel model used in this thesis is a small-scale flat fading channel. The Rayleigh fading model is the most common channel model used to describe the flat fading channel. The fading channel model together with the BPSK modulator is depicted as Fig 2.2. Compared with the AWGN channel model, the multiplicative fading factors a_k are added. The fading factors a_k are complex-valued and Gaussian distributed random variables. The mean and variance of a_k are

$$\begin{aligned} E[a_k] &= 0, \\ E[|a_k|^2] &= 2\sigma_a^2. \end{aligned} \tag{2.4}$$

We assume a_k and n_k are independent and the the variance of the received bits y_k are:

$$\sigma_y^2 = \sigma_a^2 + \sigma_n^2. \tag{2.5}$$

The amplitude $|a_k|$ has a Rayleigh distribution with pdf

$$p(x) = \frac{x}{\sigma_a^2} \exp(-\frac{x^2}{2\sigma_a^2}). \tag{2.6}$$

And that is why this fading channel model is called Rayleigh fading channel model.

For communication systems over Rayleigh fading channels, at the receiver, we need both signal to noise ratio and the fading factors. Fading channels are more complex to

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

analysis than AWGN channels because we need to estimate the fading factors. There are many channel estimation schemes. They can be classified into three categories: pilot-assisted channel estimation, blind channel estimation and hybrid channel estimation. In the proposed new metric in this thesis, we use the hybrid channel estimation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 3

Turbo Codes with Iterative Decoding Algorithms

3.1 Channel coding

The fading and AWGN noise induced by channels will affect the transmitted data bits a lot. At the receiver, it is very hard to recover what have been originally sent out. Channel coding is an effective method for deducing the effects introduced by channels.

There are many different types of error control codes, but they can be classified into block codes and convolutional codes. In [40], it has been mentioned that *"Both block codes and convolutional codes are used in the mobile radio systems. Some second generation digital cellular standards (e.g., GSM, IS-54) use convolutional codes, while others (e.g., PDC) use block codes"*.

Block codes can detect and correct a limited number of errors. The detecting and correcting error ability of the block codes depends on the code free distance. Traditional block codes include Hamming codes, BCH codes, Reed-Solomon codes etc. Hard decision block decoders are easy to implement.

Convolutional codes are different from block codes. Instead of grouping the data bits

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

into blocks and then encoding, convolutional codes use linear finite shift registers (LFSR) to encode the data bits sequence. Convolutional codes outperform block codes. The most common decoding algorithm for convolutional codes is Viterbi algorithm.

Turbo codes, introduced by Berrou and Glavieux in 1993 [3], also known as parallel concatenated recursive systematic convolutional codes, can outperform most codes.

3.2 Turbo codes

Turbo codes with iterative decoding algorithm (MAP algorithm) were introduced by Berrou and Glavieux in 1993 [3]. The turbo encoder in [3] consists of two RSC encoders, a random interleaver and a multiplexer. The multiplexer concatenates the outputs from the two RSC encoders and sends them out serially. What makes turbo code different from other channel codes is its special decoding algorithms. The turbo decoder in [3] consists two MAP decoders. Two MAP decoders generate and exchange the extrinsic information. The decoding process runs several times to obtain a better BER performance.

3.3 Turbo encoders

The turbo encoder in [3] consists of two recursive systematic convolutional (RSC) codes encoders. A simple binary code rate 1/2 RSC codes encoder with code memory 4 is depicted in Fig. 3.1, where the bits in the memory u_k is recursively calculated as

$$u_k = d_k + \sum_{i=0}^{3} g_{1i} u_{k-i}. \tag{3.1}$$

The corresponding codeword is the bit pair (d_k^s, d_k^p)

$$\begin{aligned} d_k^s &= d_k, \\ d_k^p &= d_k + \sum_{i=0}^{3} g_{2i} u_{k-i}, \quad mod\,2 \quad g_{2i} = 0,1 \end{aligned} \tag{3.2}$$

where $g_{1i} = (11111)$ and $g_{2i} = (10001)$ are the code generators. They can also be expressed in octal notation as $g_{1i} = 37$ and $g_{2i} = 21$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

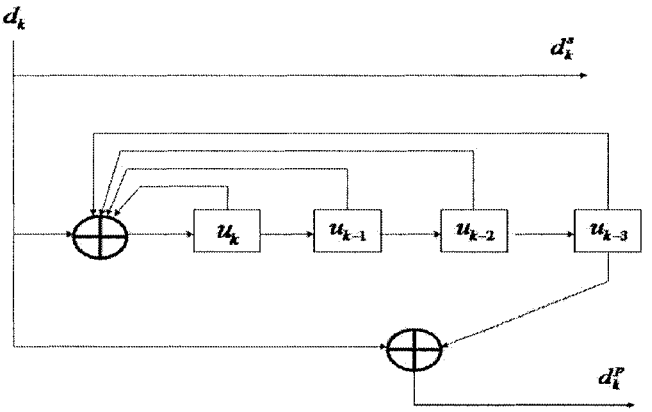

Figure 3.1: RSC codes encoder

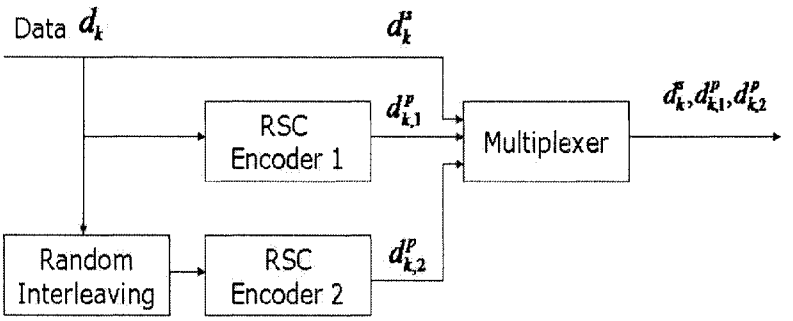

Figure 3.2: Turbo codes encoder

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

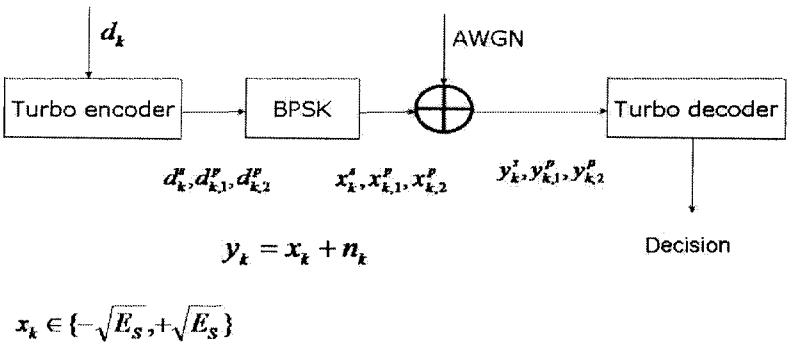

Figure 3.3: System model over AWGN channels

3.3.1 Concatenation of RSC codes encoders

The turbo encoder in [3] is shown in Fig. 3.2. The source data bits are sent to the first RSC codes encoder. The interleaved version of data bits will be sent to the second RSC codes encoder. The codewords are d_k^s, $d_{k,1}^p$ and $d_{k,2}^p$, where d_k^s are source data bits and $d_{k,1}^p$, $d_{k,2}^p$ are the parity bits from the first and second RSC encoder respectively. The code rate for this turbo encoder is $R = 1/3$, the output sequence from the multiplexer is $\{d_k^s, d_{k,1}^p, d_{k,2}^p \cdots\}$. For higher code rate, we can add a switch to puncture the codeword. For example, for $R = 1/2$, the output sequence from the multiplexer is $\{d_k^s, d_{k,1}^p, d_{k+1}^s, d_{k+1,2}^p \cdots\}$.

However, this is not the only construction of the turbo encoder. In [4], the turbo encoder can consist of N RSC encoders, the coding rate for that system will be $1/N$. Also in [14] and [15], it has been indicated that the component codes of turbo encoder can be any block codes and convolutional codes, not only RSC codes. Different design of turbo codes encoder will result in different codes performance and decoding complexity.

3.4 Turbo decoding algorithms over AWGN channels

The turbo codes system over AWGN channels is depicted in Fig. 3.3. Maximum *a posteriori* probability (MAP) decoding algorithms is the optimal algorithm for turbo decoding. However, since its complexity, it is not practical to be implemented. Log-MAP and Max-

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Log-MAP decoding algorithms have been introduced to make the implementation practical. Below, we discuss the detail of these decoding algorithms over AWGN channels and fading channels.

3.4.1 MAP algorithm over AWGN channels

The optimal sequence decoding algorithm for convolutional codes is Viterbi algorithm. Viterbi algorithm is a very simple decoding algorithm and its decoding process depends on the standard of maximizing *a posteriori* probability of a sequence received bits. MAP algorithm, on the other hand, tries to decode the received data bits depending on maximizing *a posteriori* probability of each received bit. Viterbi algorithm is a sequence decoding algorithm and MAP algorithm is a bit-by-bit decoding algorithm. The MAP algorithm is discussed thoroughly in [1] and [3], bellow we try to explain the fundamentals.

In the MAP decoder, we need to calculate the *a posteriori* probability (APP) of each received bit. For binary phase shift keying (BPSK), the transmitted bits can only be 0 or 1, so we need to calculate two APP $Pr\{d_k = i/observations\}$, $i = 0, 1$. The decoder decision depends on:

$$\begin{aligned} \widehat{d_k} &= 1 \quad if \quad Pr\{d_k = 1/observations\} \geq Pr\{d_k = 0/observations\}, \\ \widehat{d_k} &= 0 \quad if \quad Pr\{d_k = 1/observations\} \leq Pr\{d_k = 0/observations\}. \end{aligned} \tag{3.3}$$

The observations are the received bits. We assume the frame size is N and the observations are the received bits from 1 to N, noted as R_1^N, $R_k = \{y_k^s, y_k^p\}$. The APP can be expressed as

$$Pr\{d_k = i/R_1^N\} \quad i = 0, 1 \tag{3.4}$$

The APP can be derived from the joint probability:

$$\begin{aligned} Pr\{d_k = i/R_1^N\} &= \sum_m Pr\{d_k = i, S_k = m/R_1^N\}, \\ &= \sum_m \sum_{m'} Pr\{d_k = i, S_k = m, S_{k-1} = m'/R_1^N\}. \end{aligned} \tag{3.5}$$

where S_k is the encoder trellis state at time k and S_{k-1} is the encoder trellis state at time $k-1$, m and m' are the possible states. Using the Bayes' rule, the APP can be expressed

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

as

$$Pr\{d_k = i/R_1^N\} = \frac{1}{Pr\{R_1^N\}} \sum_m \sum_{m'} Pr\{d_k = i, S_k = m, S_{k-1} = m', R_1^N\}. \quad (3.6)$$

Further, if the trellis state S_k is known, the trellis state and received bits after time k are not affected by observation R_1^k and bit d_k. Using the Bayes' rule, the APP can be expressed as

$$\begin{aligned} Pr\{d_k = i/R_1^N\} &= \frac{1}{Pr\{R_1^N\}} \sum_m \sum_{m'} Pr\{d_k = i, S_k = m, S_{k-1} = m', R_1^{k-1}, R_k, R_{k+1}^N\} \\ &= \frac{1}{Pr\{R_1^N\}} \sum_m \sum_{m'} Pr\{R_{k+1}^N/S_k = m\} \\ &\quad \times Pr\{d_k = i, S_k = m, S_{k-1} = m', R_1^{k-1}, R_k\} \\ &= \frac{1}{Pr\{R_1^N\}} \sum_m \sum_{m'} Pr\{R_{k+1}^N/S_k = m\} \\ &\quad \times Pr\{d_k = i, S_k = m, R_k/S_{k-1} = m'\} Pr\{S_{k-1} = m', R_1^{k-1}\} \quad (3.7) \end{aligned}$$

In [1], three probability functions have been defined to help the calculation of the APP:

$$\begin{aligned} \alpha_k(m) &= Pr\{S_k = m, R_1^k\}, \\ \beta_k(m) &= Pr\{R_{k+1}^N/S_k = m\}, \\ \gamma_i(R_k, m', m) &= Pr\{d_k = i, S_k = m, R_k/S_{k-1} = m'\}. \quad (3.8) \end{aligned}$$

where $\alpha_k(m)$, $\beta_k(m)$, and $\gamma_i(R_k, m', m)$ are called forward probability, backward probability, and transition probability, respectively. By taking (3.8) into (3.7), we obtain

$$Pr\{d_k = i/R_1^N\} = \frac{1}{Pr\{R_1^N\}} \sum_m \sum_{m'} \gamma_i(R_k, m', m) \alpha_{k-1}(m') \beta_k(m) \quad (3.9)$$

As shown in [1], the $\alpha_k(m)$ and $\beta_k(m)$ can be recursively calculated by $\gamma_i(R_k, m', m)$ as

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

follows:

$$\begin{aligned}
\alpha_k(m) &= Pr\{S_k = m, R_1^k\} \\
&= \sum_{m'}\sum_{i=0}^{1} Pr\{d_k = i, S_{k-1} = m', S_k = m, R_1^k\} \\
&= \sum_{m'}\sum_{i=0}^{1} Pr\{d_k = i, S_{k-1} = m', S_k = m, R_k, R_1^{k-1}\} \\
&= \sum_{m'}\sum_{i=0}^{1} Pr\{d_k = i, S_k = m, R_k/S_{k-1} = m'\} Pr\{S_{k-1} = m', R_1^{k-1}\} \\
&= \sum_{m'}\sum_{i=0}^{1} \gamma_i(R_k, m', m)\alpha_{k-1}(m') \qquad (3.10)
\end{aligned}$$

$$\begin{aligned}
\beta_k(m) &= Pr\{R_{k+1}^N/S_k = m\} \\
&= \sum_{m'}\sum_{i=0}^{1} Pr\{d_{k+1} = i, S_{k+1} = m', R_{k+1}^N/S_k = m\} \\
&= \sum_{m'}\sum_{i=0}^{1} Pr\{d_{k+1} = i, S_{k+1} = m', R_{k+1}, R_{k+2}^N/S_k = m\} \\
&= \sum_{m'}\sum_{i=0}^{1} Pr\{d_{k+1} = i, S_{k+1} = m', R_{k+1}/S_k = m\} Pr\{R_{k+2}^N/S_{k+1} = m'\} \\
&= \sum_{m'}\sum_{i=0}^{1} \gamma_i(R_{k+1}, m, m')\beta_{k+1}(m') \qquad (3.11)
\end{aligned}$$

The transition probability can be further separated into three terms:

$$\begin{aligned}
\gamma_i(R_k, m', m) &= Pr\{R_k/d_k = i, S_k = m, S_{k-1} = m'\} Pr\{d_k = i, S_k = m/S_{k-1} = m'\} \\
&= Pr\{R_k/d_k = i, S_k = m, S_{k-1} = m'\} Pr\{d_k = i/S_k = m, S_{k-1} = m'\} \\
&\quad \times Pr\{S_k = m/S_{k-1} = m'\}. \qquad (3.12)
\end{aligned}$$

The first term in (3.12) depends on the channel. In an AWGN channel, since the systematic

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

data bits y_k^s are independent of the trellis states:

$$\begin{aligned} Pr\{R_k/d_k = i, S_k = m, S_{k-1} = m'\} &= Pr\{y_k^s/d_k = i, S_k = m, S_{k-1} = m'\} \\ &\quad \times Pr\{y_k^p/d_k = i, S_k = m, S_{k-1} = m'\} \\ &= Pr\{y_k^s/d_k = i\} \\ &\quad \times Pr\{y_k^p/d_k = i, S_k = m, S_{k-1} = m'\} \quad (3.13) \end{aligned}$$

$$\begin{aligned} Pr\{y_k^s/d_k = i\} &= \frac{1}{\sqrt{2\pi}\sigma_n} \cdot e^{-\frac{(y_k^s - x_k^s)^2}{2\sigma_n^2}} \\ Pr\{y_k^p/d_k = i, S_k = m, S_{k-1} = m'\} &= \frac{1}{\sqrt{2\pi}\sigma_n} \cdot e^{-\frac{(y_k^p - x_k^p)^2}{2\sigma_n^2}} \end{aligned} \quad (3.14)$$

The second term of (3.12), $Pr\{d_k = i/S_k = m, S_{k-1} = m'\}$, is 0 or 1 depending on the trellis. The third term of (3.12), $Pr\{S_k = m/S_{k-1} = m'\}$, is 1/2 depending on trellis.

3.4.2 The modified MAP algorithm for turbo decoding

In [3], in order to use the MAP algorithm in turbo decoding, Berrou modified the original MAP algorithm. Since BPSK signal only have two possible values, for simplicity, instead of calculating each *a posteriori* probability, we calculate the log-likelihood ratio (LLR) of the *a posteriori* probabilities. The LLR can be expressed as

$$\Lambda(d_k) = \log \frac{Pr\{d_k = 1/observations\}}{Pr\{d_k = 0/observations\}}. \quad (3.15)$$

At the decoder, after iteratively decoding, we can make a decision $\widehat{d_k}$ by comparing $\Lambda(d_k)$ to a threshold equal to zero

$$\begin{aligned} \widehat{d_k} &= 1 \quad if \quad \Lambda(d_k) \geq 0, \\ \widehat{d_k} &= 0 \quad if \quad \Lambda(d_k) \leq 0. \end{aligned} \quad (3.16)$$

The LLR $\Lambda(d_k)$ can be expressed as the function of three probabilities:

$$\Lambda(d_k) = \log \frac{\sum_m \sum_{m'} \gamma_1(R_k, m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma_0(R_k, m', m)\alpha_{k-1}(m')\beta_k(m)}. \quad (3.17)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

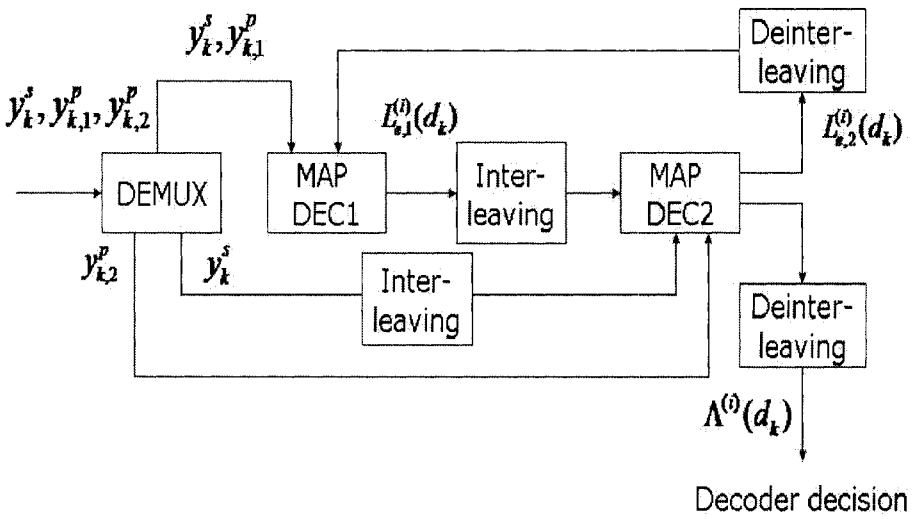

Figure 3.4: Turbo codes decoder

As indicated in [6], the LLR $\Lambda(d_k)$ can be further separated into three terms

$$\begin{aligned}\Lambda(d_k) &= \log \frac{\sum_m \sum_{m'} \gamma_1(y_k^p, m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma_0(y_k^p, m', m)\alpha_{k-1}(m')\beta_k(m)}, \\ &\quad + \log \frac{Pr\{y_k^s/d_k = 1\}}{Pr\{y_k^s/d_k = 0\}} + \log \frac{Pr\{d_k = 1\}}{Pr\{d_k = 0\}}, \\ &= L_e(d_k) + L_c \cdot y_k^s + L_a(d_k). \end{aligned} \tag{3.18}$$

where $L_e(d_k)$ is the extrinsic information generated from the decoder, $L_c \cdot y_k^s$ represents the reliability value of the channel, L_c is called the channel reliability factor and $L_a(d_k)$ is the *a priori* information of the transmitted bits. From equation (3.14), the channel reliability factor L_c for discrete memoryless Gaussian channels [5] is

$$L_c = \frac{4\sqrt{E_s}}{N_0}. \tag{3.19}$$

The extrinsic information $L_e(d_k)$ is independent of the channel reliability value $L_c \cdot y_k^s$ and the *a priori* information $L_a(d_k)$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4.3 Turbo decoder

The turbo decoder in [3] is depicted in Fig. 3.4. In the decoder, a de-multiplexer first accepts the serial received bits and parallel sends them out. The received data bits y_k^s and parity bits $y_{k,1}^p$ are sent to the first MAP decoder. The received parity bits $y_{k,2}^p$ and interleaved version of the received data bits y_k^s are sent to the second MAP decoder.

In the first MAP decoder, for the first iteration of decoding, the *a priori* probability $L_{a,1}(d_k)$ is set to 0. The extrinsic information is derived as:

$$L_{e,1}^{(1)}(d_k) = \Lambda_1^{(1)}(d_k) - L_c \cdot y_k^s - L_{a,1}(d_k). \tag{3.20}$$

The subscript 1 in $L_{e,1}^{(1)}(d_k)$ and $\Lambda_1^{(1)}(d_k)$ represents the first decoder and the superscript $(i = 1)$ represents the first iteration of decoding. The subscript 1 in $L_{a,1}(d_k)$ represents the *a priori* probability of the first turbo decoder. The channel reliability factor L_c is independent of the number of iterations and decoder. This extrinsic information $L_{e,1}^{(1)}(d_k)$, after interleaved, is sent to the second MAP decoder. The second MAP decoder will take the extrinsic information $L_{e,1}^{(1)}(d_k)$ as the *a priori* information of the transmitted data bits. In the second MAP decoder, after decoding, a new extrinsic information is $L_{e,2}^{(1)}(d_k)$ generated:

$$\begin{aligned} L_{e,2}^{(1)}(d_k) &= \Lambda_2^{(1)}(d_k) - L_c \cdot y_k^s - L_{a,2}(d_k), \\ &= \Lambda_2^{(1)}(d_k) - L_c, \cdot y_k^s - L_{e,1}^{(1)}(d_k) \end{aligned} \tag{3.21}$$

The $L_{e,2}^{(1)}(d_k)$, after de-interleaved, is fed back to the first MAP decoder.

For the next iterations, the first MAP decoder will take the de-interleaved version of $L_{e,2}^{(i-1)}(d_k)$ as the *a priori* information of the transmitted data bits; the second MAP decoder will take the interleaved version of $L_{e,1}^{(i-1)}(d_k)$ as the *a priori* information of the transmitted data bits. The general formula for generating extrinsic information when the number of iterations $i > 1$ are:

$$\begin{aligned} L_{e,1}^{(i)}(d_k) &= \Lambda_1^{(i)}(d_k) - L_c \cdot y_k^s - L_{e,2}^{(i-1)}(d_k), \\ L_{e,2}^{(i)}(d_k) &= \Lambda_2^{(i)}(d_k) - L_c \cdot y_k^s - L_{e,1}^{(i-1)}(d_k). \end{aligned} \tag{3.22}$$

The whole process runs iteratively for several times to improve the performance.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4.4 Log-MAP algorithm over AWGN channels

Log-MAP decoder does not change the way to obtain the extrinsic information. Log-MAP decoder changes the way to calculate three probability functions $\alpha_k(m)$, $\beta_k(m)$ and $\gamma_i(R_k, m', m)$. Instead of calculating three probabilities directly, Log-MAP decoder calculates them in the logarithmic domain.

$$\begin{aligned} \overline{\gamma}_i[(y_k^s, y_k^p), m', m] &\triangleq \log(\gamma_i(R_k, m', m)), \\ \overline{\alpha}_k(m) &\triangleq \log(\alpha_k(m)), \\ \overline{\beta}_k(m) &\triangleq \log(\beta_k(m)). \end{aligned} \tag{3.23}$$

Jacobian logarithm is used in [7] to derive the Log-MAP algorithm. The Jacobian logarithm for two parameters is expressed as

$$\ln(e^{\delta_1} + e^{\delta_2}) \triangleq \widehat{max}(\delta_1, \delta_2) = max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}). \tag{3.24}$$

The Jacobian logarithm for more than two parameters can be derived recursively from (3.24):

$$\begin{aligned} \ln(e^{\delta_1} + \cdots + e^{\delta_n}) &\triangleq \widehat{max}(\delta_1, \cdots, \delta_n) \\ &= \ln(\triangle + e^{\delta_n}) \quad with \quad \triangle = e^{\delta_1} + \cdots + e^{\delta_n - 1} = e^{\delta} \\ &= max(\delta, \delta_n) + \ln(1 + e^{-|\delta_n - \delta|}) \end{aligned} \tag{3.25}$$

The δ can derived from $\delta' = \ln(e^{\delta_1} + \cdots + e^{\delta_n - 2})$ and so on.

The way to derive the probabilities in logarithmic domain is out of range of this thesis. For detail, please see in [7]. Here, we use the same notation as in [7] and simply present the way to calculate the three probabilities:

$$\begin{aligned} \overline{\gamma}_i[(y_k^s, y_k^p), m', m] &= \frac{1}{2} L_a(d_k) x_k^s(i) + \frac{1}{2} L_c y_k^s x_k^s(i) + \frac{1}{2} L_c y_k^p x_k^p(i) \\ \overline{\alpha}_k(m) &= \widehat{\max_{(m',i)}} \{\overline{\gamma}_i[(y_k^s, y_k^p), m', m] + \overline{\alpha}_{k-1}(m')\} \\ \overline{\beta}_k(m) &= \widehat{\max_{(m',i)}} \{\overline{\gamma}_i[(y_{k+1}^s, y_{k+1}^p), m, m'] + \overline{\beta}_{k+1}(m')\} \end{aligned} \tag{3.26}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We see $\overline{\gamma}_i[(y_k^s, y_k^p), m', m]$ depends on the channel reliability factor L_c, the *a priori* information $L_a(d_k)$ and received bits. $\overline{\alpha}_k(m)$ and $\overline{\beta}_k(m)$ can be calculated recursively by $\overline{\gamma}_i[(y_k^s, y_k^p), m', m]$ with non-linear $\widehat{\max}$ functions. These results are very important for the discussion of the contributions later.

The way to derive LLR $\Lambda(d_k)$ [7] is:

$$\begin{aligned}\Lambda(d_k) &= \widehat{\max_{(m,m')}}\{\overline{\gamma}_1[(y_k^s, y_k^p), m', m] + \overline{\alpha}_{k-1}(m') \\ &\quad +\overline{\beta}_k(m)\} - \widehat{\max_{(m,m')}}\{\overline{\gamma}_0[(y_k^s, y_k^p), m', m] \\ &\quad +\overline{\alpha}_{k-1}(m') + \overline{\beta}_k(m)\} \end{aligned} \tag{3.27}$$

The calculation of LLR also needs the non-linear functions $\widehat{\max}$.

3.4.5 Max-Log-MAP algorithm over AWGN channels

The Log-MAP algorithm reduces computational complexity, but it still has many nonlinear functions $\widehat{\max}$. The Max-Log-MAP algorithm solves this problem by using approximations. The Max-Log-MAP decoder is deduced from the Log-MAP decoder by substituting the non-linear functions $\widehat{\max}$ with the linear functions max. The LLR $\Lambda(d_k)$ and probabilities can be expressed as

$$\begin{aligned}\Lambda(d_k) &= \max_{(m,m')}\{\overline{\gamma}_1[(y_k^s, y_k^p), m', m] + \overline{\alpha}_{k-1}(m') \\ &\quad +\overline{\beta}_k(m)\} - \max_{(m,m')}\{\overline{\gamma}_0[(y_k^s, y_k^p), m', m] \\ &\quad +\overline{\alpha}_{k-1}(m') + \overline{\beta}_k(m)\}, \end{aligned} \tag{3.28}$$

$$\begin{aligned}\overline{\alpha}_k(m) &= \max_{(m',i)}\{\overline{\gamma}_i[(y_k^s, y_k^p), m', m] + \overline{\alpha}_{k-1}(m')\}, \\ \overline{\beta}_k(m) &= \max_{(m',i)}\{\overline{\gamma}_i[(y_{k+1}^s, y_{k+1}^p), m, m'] + \overline{\beta}_{k+1}(m')\}. \end{aligned} \tag{3.29}$$

The $\overline{\gamma}_i[(y_k^s, y_k^p), m', m]$ remains unchanged.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

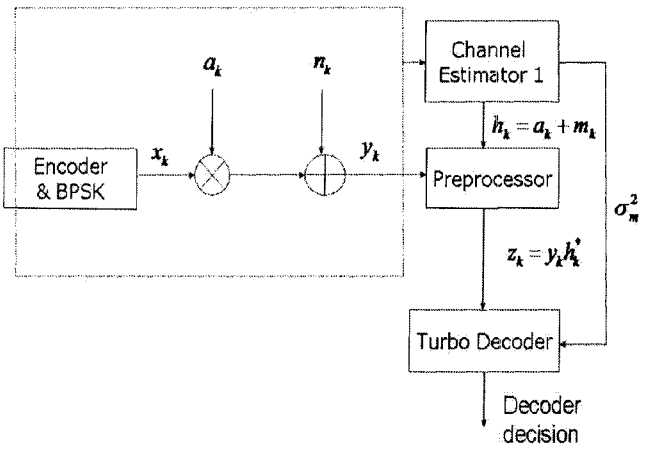

Figure 3.5: System model over fading channels

3.5 Turbo decoding algorithms over fading channels

The turbo decoding algorithms over fading channels are similar with those over AWGN channels. In [12], it has been indicated that *"when SNR is known, the decoding algorithms that are used for AWGN channels remain unchanged and only the channel reliability factor L_c needs to be redefined"*. By including the turbo decoder, fading channel model and turbo decoder, the system model over fading channels is depicted in Fig. 3.5.

In this thesis, we assume that an initial channel estimation is available from a pilot-assisted channel estimator (channel estimator 1). The initial channel estimate h_k is modeled as in [12]

$$h_k = a_k + m_k, \tag{3.30}$$

where a_k is the actual channel fading factor, m_k is the channel estimation error. The m_k is a complex-valued Gaussian distributed random variable with mean and variance

$$\begin{aligned} E[m_k] &= 0, \\ E[|m_k|^2] &= 2\sigma_m^2. \end{aligned} \tag{3.31}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We assume a_k and m_k are independent, the mean and variance of channel estimate h_k are

$$\begin{aligned} E[h_k] &= 0, \\ E[|h_k|^2] &= 2\sigma_h^2 = 2(\sigma_a^2 + \sigma_m^2). \end{aligned} \tag{3.32}$$

By comparing the system model over fading channels with that over AWGN channels, besides the fading factor a_k and pilot-assisted channel estimator, a new preprocessor is also added. This processor will be discussed in detail in chapter 5. In the preprocessor, with the received bits y_k and initial channel estimate h_k available, we calculate the decision variables

$$z_k = y_k h_k^* = z_{k,r} + j z_{k,i}, \tag{3.33}$$

where j is the imaginary unit and $z_{k,r}$ and $z_{k,i}$ are the real and imaginary part of z_k. In [12], the cross correlation coefficient of y_k and h_k is defined as:

$$\begin{aligned} \mu_k &\triangleq \frac{E[y_k h_k^*]}{\sqrt{E[|y_k|^2]E[|h_k|^2]}}, \\ &= x_k \frac{2\sigma_a^2}{\sqrt{(|x_k|^2 2\sigma_a^2 + 2\sigma_n^2)(2\sigma_a^2 + 2\sigma_m^2)}}, \\ &= |\mu| e^{-j\varepsilon_k}. \end{aligned} \tag{3.34}$$

Further, Frenger in [12] derived the probability of decision variable conditioned on the transmitted bits as follows:

$$Pr\{z_k/d_k\} = \frac{1}{2\pi\sigma_h^2\sigma_y^2(1-|\mu|^2)} \exp\left[\frac{\Re[z_k\mu_k]}{\sigma_y\sigma_h(1-|\mu|^2)}\right] K_0\left(\frac{|z_k|}{\sigma_y\sigma_h(1-|\mu|^2)}\right), \tag{3.35}$$

where $K_0(x)$ is the zeroth-order Hankel function of x, and $\Re[x]$ is the real component of x.

Over AWGN channels, when we derive the channel reliability factor L_c, we use the probability of received data bits on the transmitted bits:

$$Pr\{y_k^s/d_k\} = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(y_k^s - x_k^s)^2}{2\sigma_n^2}}. \tag{3.36}$$

and the channel reliability factor L_c is derived as:

$$\begin{aligned} \log\frac{Pr\{y_k^s/d_k=1\}}{Pr\{y_k^s/d_k=0\}} &= L_c y_k^s, \\ L_c &= \frac{4\sqrt{E_s}}{N_0}. \end{aligned} \tag{3.37}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Over fading channels, based on the analysis of conditional probability of decision variable z_k, the channel reliability factor for turbo decoding algorithms over fading channels can be redefined as follows:

$$\log \frac{Pr\{z_k/d_k = 1\}}{Pr\{z_k/d_k = 0\}} = L_c \Re[z_k] \tag{3.38}$$

where

$$L_c = \frac{4\sqrt{E_s}}{N_0}\sigma_a^2 \left[\sigma_m^2 \left(\frac{2E_s}{N_0}\sigma_a^2 + 1\right) + \sigma_a^2\right]^{-1}. \tag{3.39}$$

As seen, by assuming that the variance of fading factor σ_a^2 is known, the channel reliability factor L_c over fading channels depends on both SNR and the error variance of channel estimate σ_m^2.

The way of the Log-MAP and Max-Log-MAP decoding algorithms over fading channels will stay the same as over AWGN channels. There are only two modifications. The original received bits y_k is replaced by the decision variables z_k and the channel reliability factor L_c is redefined as in (3.39).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 4

Turbo Decoding with Integrated Channel Estimation

Log-MAP turbo decoding algorithm over fading channels needs the information of SNR and the error variance of channel estimate σ_m^2 to obtain the channel reliability factor L_c. It has been proved in [12] that smaller error variance results in better BER performance. Many estimation schemes have been introduced to reduce the error variance of channel estimate and thus improve the BER performance. Most of the schemes are processed before the start of the turbo decoding. However, the extrinsic information generated and exchanged between the turbo decoders has some *a priori* information of the transmitted data bits. This information can help us to refine the channel estimate and reduce the σ_m^2 after each iteration of decoding.

In this chapter, we propose a new algorithm of turbo decoding integrated with channel estimation over fading channels. We use the Log-MAP turbo decoder in our new algorithm. And the hybrid channel estimation scheme is used to refine the channel estimation. We first review the estimation theory in the mean-square-error sense. Then the working fundamentals of the channel estimators are discussed.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.1 Estimation theory

Using the theory of estimation, we can derive an estimator to estimate an unknown parameter from our observations. Usually, in the estimation process, we try to minimize a cost function. In the mean-square-error sense, the cost function is

$$C[x, \widehat{x}] = E[(x - \widehat{x})^2], \tag{4.1}$$

where x is the unknown signal and $\widehat{x}$ is the estimate. Given a collection of observations y, the minimum-mean-square-error (MMSE) estimator of x given y is given by, [42],

$$\widehat{x} = E[x|y] = \int_{S_x} x f_{x|y}(x|y), \tag{4.2}$$

where S_x is the domain of x. The mean value of the estimate and the resulting minimum cost is given by

$$\begin{aligned} E[\widehat{x}] &= \overline{x}, \\ E[(x - \widehat{x})^2] &= E[x^2] - E[\widehat{x}^2]. \end{aligned} \tag{4.3}$$

However, it is not always easy to obtain a closed form expression for the MMSE estimator. This difficulty limits many engineers to the linear estimators. The linear MMSE estimator and its minimum cost function or error variance can be expressed as

$$\begin{aligned} \widehat{x} &= \overline{x} + K_0(y - \overline{y}), \\ \min(E[(x - \widehat{x})^2]) &= R_x - R_{xy} R_y^{-1} R_{yx}, \end{aligned} \tag{4.4}$$

where K_0 is any solutions to the equation $K_0 R_y = R_{xy}$, and

$$\begin{aligned} R_x &= E[(x - \overline{x})(x - \overline{x})^*], \\ R_{xy} &= E[(x - \overline{x})(y - \overline{y})^*], \\ R_{yx} &= E[(y - \overline{y})(x - \overline{x})^*], \\ R_y &= E[(y - \overline{y})(y - \overline{y})^*]. \end{aligned} \tag{4.5}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

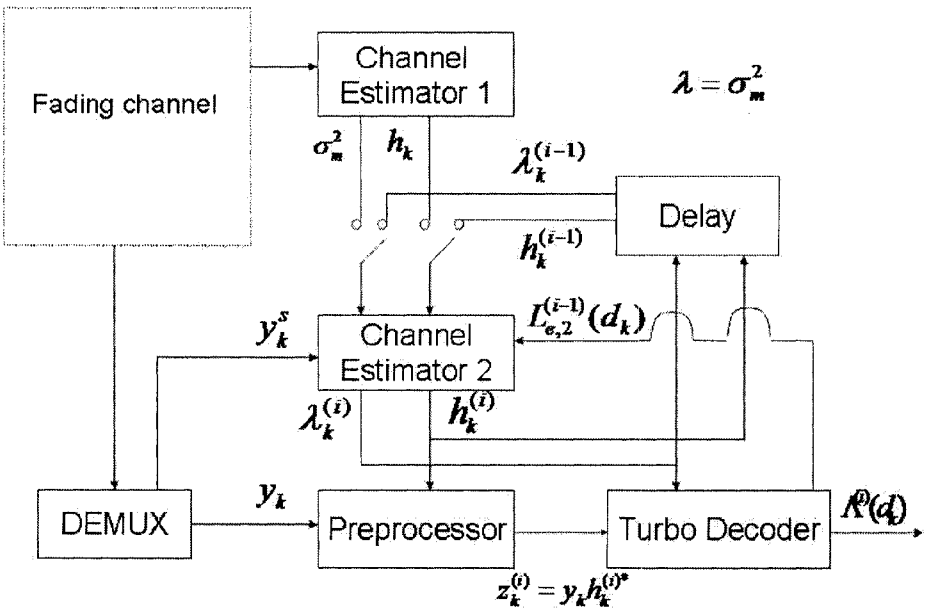

Figure 4.1: Turbo decoding integrated with channel estimation

4.2 A new algorithm

In this thesis, we propose a new turbo decoding algorithm with integrated channel estimation over Rayleigh fading channels as depicted in Fig. 4.1. The turbo decoder used in this algorithm is a Log-MAP decoder. Comparing with the original Log-MAP decoder over fading channels, we have added the following blocks: a blind channel estimator (channel estimator 2), a de-multiplexer, and a delay module.

4.2.1 The first iteration of decoding

For the first iteration, we assume there is a pilot-assisted channel estimator (channel estimator 1) which provides us with the initial channel estimate h_k and their error variance σ_m^2. This initial channel estimate h_k go through the channel estimator 2 which is a blind linear MMSE channel estimator without any processing. After that, they are sent to both the preprocessor and a delay module. The error variance σ_m^2 also goes through the channel

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

estimator 2 and is sent to both the turbo decoder and the delay module. The delay module reserves h_k and σ_m^2 for the next iteration of channel estimation. In the turbo decoder, after the first iteration of decoding, we have the first version of the extrinsic information $L_{e,2}^{(1)}(d_k)$ of the transmitted data bits. The de-interleaved version of the extrinsic information $L_{e,2}^{(1)}(d_k)$ from the second Log-MAP decoder, instead of being fed back to the first Log-MAP decoder, is sent to the channel estimator 2.

4.2.2 The next iterations of decoding

For the next iterations, a de-multiplexer takes out the received data bits y_k^s and sends them to the channel estimator 2, the proposed estimator. Depending on the reserved channel estimate $h_k^{(i-1)}$ and its error variance, the received data bits y_k^s and extrinsic information $L_{e,2}^{(i-1)}(d_k)$, we refine the channel estimate for data bits $h_k^{(i)}$ and its error variance in the linear MMSE estimator (channel estimator 2). For simplicity, we use the following notation:

$$\lambda \triangleq \sigma_m^2. \tag{4.6}$$

The refined error variance for iteration i and bit d_k is $\lambda_k^{(i)}$. The refined channel estimate $h_k^{(i)}$ and their error variance $\lambda_k^{(i)}$ are again saved in the delay module for the next iteration of channel estimation. They are also sent to the turbo decoder for this iteration of decoding. After decoding, we have an updated version of the extrinsic information $L_{e,2}^{(i)}(d_k)$. This information will be fed back to the channel estimator 2, the proposed estimator, for the next iteration of channel estimation. The whole process will run iteratively. In following, we go into detail to explain the process of exchanging information between the channel estimator 2 and turbo decoding.

The channel estimator 2, depending on the received data bits y_k^s, the reserved channel estimate for data bits $h_k^{(i-1)}$ together with its error variance $\lambda_k^{(i-1)}$ and the de-interleaved extrinsic information $L_{e,2}^{(i-1)}(d_k)$, derives the new channel estimate $h_k^{(i)}$ of a_k and its error variance $\lambda_k^{(i)}$:

$$\widehat{h_k^{(i)}} = K_0 \begin{bmatrix} h_k^{(i-1)} \\ y_k^s \end{bmatrix} \tag{4.7}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Here we assign a_k as matrix 1 and $\begin{bmatrix} h_k^{(i-1)} \\ y_k^s \end{bmatrix}$ as matrix 2, then K_0 can be expressed as

$$K_0 = R_{12}R_2^{-1}, \tag{4.8}$$

where subscripts 1 and 2 are for matrix 1 and 2, respectively. The error variance is

$$E[|m_k|^2] = 2\widehat{\lambda_k^{(i)}} = R_1 - R_{12}R_2^{-1}R_{21}. \tag{4.9}$$

The correlation functions are

$$\begin{aligned}
R_1 &= E[a_k a_k^*] = 2\sigma_a^2 \\
R_{12} &= E\left\{ a_k \begin{bmatrix} h_k^{(i-1)} \\ y_k^s \end{bmatrix}^* \right\} = [2\sigma_a^2 \quad 2\sigma_a^2(x_k^s)^*] \\
R_{21} &= E\left\{ \begin{bmatrix} h_k^{(i-1)} \\ y_k^s \end{bmatrix} a_k^* \right\} = \begin{bmatrix} 2\sigma_a^2 \\ 2\sigma_a^2(x_k^s)^* \end{bmatrix} \\
R_2 &= E\left\{ \begin{bmatrix} h_k^{(i-1)} \\ y_k^s \end{bmatrix} \begin{bmatrix} h_k^{(i-1)} \\ y_k^s \end{bmatrix}^* \right\} = \begin{bmatrix} 2(\sigma_a^2 + \lambda_k^{(i-1)}) & 2\sigma_a^2(x_k^s)^* \\ 2\sigma_a^2 x_k^s & 2(\sigma_a^2|x_k^s|^2 + \sigma_n^2) \end{bmatrix}
\end{aligned} \tag{4.10}$$

The new channel estimate and its variance can then be expressed as

$$\widehat{h_k^{(i)}} = \frac{1}{1 + \lambda_k^{(i-1)}(\frac{2E_s}{N_0} + \frac{1}{\sigma_a^2})} \left(h_k^{(i-1)} + \frac{2\lambda_k^{(i-1)}}{N_0} x_k^{s*} y_k^s \right) \tag{4.11}$$

$$\widehat{\lambda_k^{(i)}} = 2\sigma_a^2 \left(1 - \frac{1 + \frac{2\lambda_k^{(i-1)}}{N_0} x_k^{s*}}{1 + \lambda_k^{(i-1)}(\frac{2E_s}{N_0} + \frac{1}{\sigma_a^2})} \right) \tag{4.12}$$

In (4.11) and (4.12), we see x_k^s is in the equation. However, they are the unknown transmitted data bits. What we have is the extrinsic information $L_{e,2}^{(i-1)}(d_k)$ for the data bits from the last iteration of turbo decoding. We assign $Pr(d_k = 1) = Pr(x_k^s = +1) = p$, $Pr(d_k = 0) = Pr(x_k^s = -1) = 1 - p$, the extrinsic information $L_{e,2}^{(i-1)}(d_k)$ from the last iteration of decoding will be used as the *a priori* information

$$L_{e,2}^{(i-1)}(d_k) = \log \frac{p}{1-p} \tag{4.13}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

so we have

$$p = \frac{\exp[L_{e,2}^{(i-1)}(d_k)]}{1+\exp[L_{e,2}^{(i-1)}(d_k)]} \tag{4.14}$$

The pdf of the transmitted data bits x_k^s can be expressed as

$$f_x(x) = p\delta(x_k^s - 1) + (1-p)\delta(x_k^s + 1). \tag{4.15}$$

The mean value of the transmitted data bits x_k^s can be derived as

$$E[x_k^s] = \int x f_x(x)\, dx = 2p - 1 = tanh(\frac{L_{e,2}^{(i-1)}(d_k)}{2}). \tag{4.16}$$

We try to approximate the new channel estimate and its error variance by taking the expected value to (4.11) and (4.12) on x_k^s as follows:

$$\begin{aligned} h_k^{(i)} &= E_{x_k^s}[\widehat{h_k^{(i)}}] \\ &= \frac{1}{1+\lambda_k^{(i-1)}(\frac{2E_s}{N_0}+\frac{1}{\sigma_a^2})}\left(h_k^{(i-1)} + \frac{2\lambda_k^{(i-1)}}{N_0}\tanh(\frac{L_{e,2}^{(i-1)}(d_k)}{2})y_k^s\right) \end{aligned} \tag{4.17}$$

and

$$\lambda_k^{(i)} = E_{x_k^s}[\widehat{\lambda_k^{(i)}}] = 2\sigma_a^2\left(1 - \frac{1+\frac{2\lambda_k^{(i-1)}}{N_0}\tanh(\frac{L_{e,2}^{(i-1)}(d_k)}{2})}{1+\lambda_k^{(i-1)}(\frac{2E_s}{N_0}+\frac{1}{\sigma_a^2})}\right). \tag{4.18}$$

Now, we save both information in the delay module for the next iteration of channel estimation. At the same time, the refined the channel estimate $h_k^{(i)}$ (only the channel estimate for data bits have been refined) are sent to the preprocessor to derive the new decision variables $z_k^{(i)}$:

$$z_k^{(i)} = y_k(h_k^{(i)})^*. \tag{4.19}$$

The refined error variance $\lambda_k^{(i)}$ are sent to the turbo decoder to update the channel reliability factor:

$$L_c^{(i)}(d_k) = \frac{4\sqrt{E_s}}{N_0}\sigma_a^2\left[\lambda_k^{(i)}\left(\frac{2E_s}{N_0}\sigma_a^2+1\right)+\sigma_a^2\right]^{-1}. \tag{4.20}$$

For the sake of comparison, we also show the channel reliability factor L_c derived by Frenger, [11]:

$$L_c = \frac{4\sqrt{E_s}}{N_0}\sigma_a^2\left[\sigma_m^2\left(\frac{2E_s}{N_0}\sigma_a^2+1\right)+\sigma_a^2\right]^{-1}. \tag{4.21}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

We see in (4.21), the error variance σ_m^2 is the same for all the transmitted bits. Also, the error variance σ_m^2 is estimated before the start of turbo decoding. In the new algorithm, however, each transmitted bit has its own error variance $\lambda_k^{(i)}$, also the error variance are re-estimated after each iteration of turbo decoding. The channel reliability factor $L_c^{(i)}(d_k)$ is then updated after each iteration of decoding for each transmitted bit. In the new algorithm, the Log-MAP turbo decoder is integrated with a blind channel estimator to improve the performance.

The inputs to our proposed channel estimator are the channel estimate from the last iteration $h_k^{(i-1)}$ and its error variance $\lambda_k^{(i-1)}$, the received data bits y_k^s and extrinsic information from the last turbo decoding $L_{e,2}^{(i-1)}(d_k)$. The outputs of the proposed channel estimator are the refined channel estimate $h_k^{(i)}$ and its error variance $\lambda_k^{(i)}$. This two information will be reserved for the next iteration of channel estimate. Further, the refined channel estimate $h_k^{(i)}$ will first be preprocessed to derive the new decision variable $z_k^{(i)}$. The refined error variance $\lambda_k^{(i)}$ and the new decision variable $z_k^{(i)}$ will then be sent to the turbo decoder to derive the new channel reliability factor $L_c^{(i)}(d_k)$. The turbo decoder then generates the new extrinsic information $L_{e,2}^{(i)}(d_k)$ and sends it to the channel estimator 2 for the next iteration of channel estimation.

The proposed blind channel estimator uses the *a priori* information of the transmitted data bits buried under the extrinsic information from the turbo decoder. This information can help us to refine the channel estimate and its error variance for data bits. However, the proposed channel estimator will only refine the channel estimate for the transmitted data bits, but not for the transmitted parity bits, because we only have the extrinsic information of the data bits. Also, since the fading factors in our fading channel model are independent from each other, the proposed channel estimator is working in a bit-by-bit way. The refined channel estimate and its error variance are then fed back to the turbo decoder for the next iteration of decoding. The resulting iteration between the channel estimator and the turbo decoder using the new algorithm, with intermediate exchange of soft channel-symbol information, can improve the performance of both the channel estimator and the channel decoder, as shown by the simulation results in Chapter 7.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 5

Fading Compensation

One of the differences between the structure of turbo decoding over fading channels and AWGN channels is a preprocessor or a fading compensator that can undo the effect of fading. In this chapter, we try to explain why the received data symbols are multiplied by the complex conjugate of an estimate of the channel gain before turbo decoding, [44]. Since there has been no mathematical justification in the literature for this kind of fading compensation for systems without perfect knowledge of the channel, we try to come up with some justification. First, we derive a linear MMSE estimator to obtain optimal estimates of the transmitted data symbols with noisy channel estimates. Then, it is shown the linear MMSE estimator can be viewed as a normalized version of the preprocessing (fading compensation) operation used in most existing turbo decoding algorithms, [11]. Also, we show that the normalization has no effect on the BER performance of the turbo decoder.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

5.1 Linear MMSE estimator of the transmitted bits

The linear MMSE estimator tries to estimate the transmitted bits x_k from the received bits y_k and the channel estimate h_k. We know the received bits and channel estimate are

$$\begin{aligned} y_k &= a_k x_k + n_k, \\ h_k &= a_k + m_k. \end{aligned} \tag{5.1}$$

From these two equations, we have

$$y_k = (h_k - m_k)x_k + n_k. \tag{5.2}$$

The linear MMSE estimator is given by

$$\mathcal{Z}_k = \widehat{x_k} = K_0 y_k, \tag{5.3}$$

where $K_0 = R_{x_k y_k} R_{y_k}^{-1}$, and $R_{x_k y_k}$, R_{y_k} are given by

$$\begin{aligned} R_{x_k y_k} &= E[x_k y_k^*] = h_k^*, \\ R_{y_k} &= E[y_k y_k^*] = |h_k|^2 + 2\sigma_m^2 + 2\sigma_n^2. \end{aligned} \tag{5.4}$$

Then, the linear MMSE estimator in (5.3) can be written as follows:

$$\mathcal{Z}_k = \widehat{x_k} = \frac{1}{|h_k|^2 + 2\sigma_m^2 + 2\sigma_n^2} y_k h_k^*. \tag{5.5}$$

5.2 The preprocessing before turbo decoding

In Frenger's paper [12], the preprocessing operation before the turbo decoder is performed as follows:

$$z_k = y_k h_k^*. \tag{5.6}$$

The relation between the linear MMSE estimator in (5.5) and the fading compensator used by Frenger in (5.6) can be shown as follows:

$$\mathcal{Z}_k = \frac{1}{R_{y_k}} z_k. \tag{5.7}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Using the results obtained in [12], we know the pdf of z_k conditioned on the transmitted code symbol x_k is:

$$p_{z_k|x_k}(z_k) = \frac{1}{2\pi\sigma_h^2\sigma_y^2(1-|\mu|^2)} \exp\left[\frac{\Re[z_k\mu_k]}{\sigma_y\sigma_h(1-|\mu|^2)}\right] K_0\left(\frac{|z_k|}{\sigma_y\sigma_h(1-|\mu|^2)}\right) \tag{5.8}$$

The pdf of $\mathcal{Z}_k$ conditioned on the transmitted code symbol x_k can be derived as

$$\begin{aligned} p_{\mathcal{Z}_k|x_k}(\mathcal{Z}_k) &= |R_{y_k}| p_{z_k|x_k}(R_{y_k}\mathcal{Z}_k) \\ &= \frac{|R_{y_k}|}{2\pi\sigma_h^2\sigma_y^2(1-|\mu|^2)} \exp\left[\frac{\Re[R_{y_k}\mathcal{Z}_k\mu_k]}{\sigma_y\sigma_h(1-|\mu|^2)}\right] K_0\left(\frac{|R_{y_k}\mathcal{Z}_k|}{\sigma_y\sigma_h(1-|\mu|^2)}\right) \end{aligned} \tag{5.9}$$

We can derive the channel reliability factor from the pdf in (5.9) as follows:

$$\log\left(\frac{p_{\mathcal{Z}_k|x_k}(\mathcal{Z}_k|x_k=+1)}{p_{\mathcal{Z}_k|x_k}(\mathcal{Z}_k|x_k=-1)}\right) = \mathcal{L}_c \times \Re[\mathcal{Z}_k] \tag{5.10}$$

where $\mathcal{L}_c$ is the channel reliability factor,

$$\mathcal{L}_c = \frac{2|\mu|R_{y_k}}{\sigma_y\sigma_h(1-|\mu|^2)}. \tag{5.11}$$

For the Log-MAP turbo decoder, in order to find the LLR $\Lambda(d_k)$, we need to calculate probabilities $\overline{\gamma}_i[\mathcal{Z}_k^s, \mathcal{Z}_k^p, m', m]$, $\overline{\alpha}_k(m)$ and $\overline{\beta}_k(m)$. $\overline{\alpha}_k(m)$ and $\overline{\beta}_k(m)$ can be recursively calculated by $\overline{\gamma}_i[\mathcal{Z}_k^s, \mathcal{Z}_k^p, m', m]$, and $\overline{\gamma}_i[\mathcal{Z}_k^s, \mathcal{Z}_k^p, m', m]$ in this case is

$$\overline{\gamma}_i[\mathcal{Z}_k^s, \mathcal{Z}_k^p, m', m] = \frac{1}{2}L_a(d_k)x_k^s(i) + \frac{1}{2}\mathcal{L}_c\mathcal{Z}_{k,r}^s x_k^s(i) + \frac{1}{2}\mathcal{L}_c\mathcal{Z}_{k,r}^p x_k^p(i) \tag{5.12}$$

we find that the R_{y_k} in the numerator of $\mathcal{L}_c$ and the R_{y_k} in the denominator of $\mathcal{Z}_{k,r}^s$ and $\mathcal{Z}_{k,r}^p$ are canceled out by each other, as since:

$$\overline{\gamma}_i[\mathcal{Z}_k^s, \mathcal{Z}_k^p, m', m] = \overline{\gamma}_i[z_k^s, z_k^p, m', m] \tag{5.13}$$

As a result, the normalization factor does not affect the performance of turbo decoding. In other words, employing the fading compensator (5.6), used by Frenger, or the linear MMSE estimator derived in (5.5) as a fading compensator does not change the BER performance of the turbo decoder.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 6

Effect of SNR Mismatch on the Performance of Turbo Decoding Algorithms

One requirement of the Log-MAP turbo decoder over fading channels is the knowledge of SNR and the error variance of channel estimate σ_m^2 to derive the channel reliability factor L_c. Since the proposed turbo decoding algorithm composed of the Log-MAP decoder, both the channel SNR and the channel fading must be estimated. Adding any estimator to this algorithm increases the complexity of the algorithm. Therefore, any decoding algorithm that does not need perfect knowledge of SNR or fading factor is more desirable. The effect of SNR mismatch on the performance of turbo decoding algorithms over AWGN channels has been discussed in [29] by Jordan and Nichols. In [30] and [28], the effect of SNR mismatch over the fully interleaved channels has been discussed. Valenti and Woerner further discussed the effect of SNR mismatch over correlated fading channels in [27]. In 2000, Alexander indicated in [9] that *"An estimation of SNR is not necessary when implementing the turbo decoder with the Max-Log-MAP algorithm"*. However, for fading channels, he assumed the channel is perfectly known at the receiver. This is not a good assumption. Practically, for fading

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

channels, neither SNR nor channel is perfectly known. In this chapter, we first analyze the effects of SNR mismatch on the performance of the Log-MAP and Max-Log-MAP decoding algorithms over AWGN channels, then we move to fading channels.

6.1 Effect of an SNR mismatch (AWGN Channels)

6.1.1 The Max-Log-MAP algorithm

We first follow Alexander's work and analyze the effect of SNR mismatch on the performance of Max-Log-MAP algorithm over AWGN channels.

The estimated SNR at the input of receiver $(\frac{E_b}{N_0})_r$ in decibels equals to the actual value of the channel SNR $(\frac{E_b}{N_0})_{ch}$ added by an SNR offset $\triangle(\frac{E_b}{N_0})$:

$$(\frac{E_b}{N_0'})_r = (\frac{E_b}{N_0})_{ch} + \triangle(\frac{E_b}{N_0}). \tag{6.1}$$

The variance of the noise at the input of the receiver N_0' can be expressed as:

$$N_0' = K_1 N_0, \tag{6.2}$$

where

$$K_1 = 10^{-\frac{1}{10}\triangle(\frac{E_b}{N_0})}. \tag{6.3}$$

The channel reliability factor L_c over AWGN channels at the receiver is:

$$\mathcal{L}_c = \frac{4\sqrt{E_s}}{N_0'} = \frac{1}{K_1} L_c. \tag{6.4}$$

By considering the SNR mismatch indicated in (6.4), the branch transition probability at the receiver can be expressed as

$$\begin{aligned} \widetilde{\gamma}_i[(y_k^s, y_k^p), m', m] &= \frac{1}{2} L_a(d_k) x_k^s(i) + \frac{1}{2} \mathcal{L}_c y_k^s x_k^s(i) + \frac{1}{2} \mathcal{L}_c y_k^p x_k^p(i), \\ &= \frac{1}{2} L_a(d_k) x_k^s(i) + \frac{1}{2}\frac{1}{K_1} L_c y_k^s x_k^s(i) + \frac{1}{2}\frac{1}{K_1} L_c y_k^p x_k^p(i). \end{aligned} \tag{6.5}$$

The hat on the $\widetilde{\gamma}_i[(y_k^s, y_k^p), m', m]$ denotes that this parameter considers the SNR mismatch at the receiver. The parameters without this hat denote there are not SNR mismatch.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

For the first iteration at the first turbo decoder, we always assume the *a priori* information $L_{a,1}(d_k) = 0$, so the first term in (6.5) is 0 and the transition probability at the receiver can be expressed as

$$\widetilde{\gamma}_i[(y_k^s, y_k^p), m', m] = \frac{1}{K_1}\overline{\gamma}_i[(y_k^s, y_k^p), m', m]. \tag{6.6}$$

Since the calculation of $\widetilde{\alpha}_k(m)$, $\widetilde{\beta}_k(m)$ and $\widetilde{\Lambda}_1^{(1)}(d_k)$ only need linear *max* functions, by taking (6.6) into the calculation of these probabilities, we have

$$\begin{aligned} \widetilde{\alpha}_k(m) &= \frac{1}{K_1}\overline{\alpha}_k(m), \\ \widetilde{\beta}_k(m) &= \frac{1}{K_1}\overline{\beta}_k(m), \\ \widetilde{\Lambda}_1^{(1)}(d_k) &= \frac{1}{K_1}\Lambda_1^{(1)}(d_k). \end{aligned} \tag{6.7}$$

For the first turbo decoder, since K_1 is always a positive number, the hard decision which depends on the sign of the LLR $\widetilde{\Lambda}_1^{(1)}(d_k)$ will not be affected by the SNR mismatch $\triangle(\frac{E_b}{N_0})$.

For the second turbo decoder, the extrinsic information generated from the first turbo decoder can be expressed as

$$\mathcal{L}_{e,1}^{(1)}(d_k) = \widetilde{\Lambda}_1^{(1)}(d_k) - L_{a,1}(d_k) - \mathcal{L}_c y_k^s. \tag{6.8}$$

In (6.8), the second term on the right of the equation is 0 , the LLR $\widetilde{\Lambda}_1^{(1)}$ and channel reliability factor $\mathcal{L}_c$ have been proved to be proportional to the factor $\frac{1}{K_1}$, so the extrinsic information $\mathcal{L}_{e,1}^{(1)}(d_k)$ will be proportional to the factor $\frac{1}{K_1}$ too. This extrinsic information, after interleaving, will be used as the *a priori* information for the second turbo decoder. We substitute the $L_{a,2}(d_k)$ in the branch transition probability calculation with $\mathcal{L}_{e,1}^{(1)}(d_k)$ and get

$$\widetilde{\gamma}_i[(y_k^s, y_k^p), m', m] = \frac{1}{2}\mathcal{L}_{e,1}^{(1)}(d_k)x_k^s(i) + \frac{1}{2}\mathcal{L}_c y_k^s x_k^s(i) + \frac{1}{2}\mathcal{L}_c y_k^p x_k^p(i). \tag{6.9}$$

Similar to the proof for the first turbo decoder, the first term in (6.9) now is proportional to the $\frac{1}{K_1}$, we have

$$\widetilde{\gamma}_i[(y_k^s, y_k^p), m', m] = \frac{1}{K_1}\overline{\gamma}_i[(y_k^s, y_k^p), m', m], \tag{6.10}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and get the same results as (6.7):

$$\begin{aligned}\widetilde{\alpha}_k(m) &= \frac{1}{K_1}\overline{\alpha}_k(m),\\ \widetilde{\beta}_k(m) &= \frac{1}{K_1}\overline{\beta}_k(m),\\ \widetilde{\Lambda}_2^{(1)}(d_k) &= \frac{1}{K_1}\Lambda_2^{(1)}(d_k).\end{aligned} \tag{6.11}$$

For the next iterations, the whole process repeats. The second turbo decoder will also generate the extrinsic information $\mathcal{L}_{e,2}^{(1)}(d_k)$,

$$\begin{aligned}\mathcal{L}_{e,2}^{(1)}(d_k) &= \widetilde{\Lambda}_2^{(1)}(d_k) - L_{a,2}(d_k) - \mathcal{L}_c y_k^s,\\ &= \widetilde{\Lambda}_2^{(1)}(d_k) - \mathcal{L}_{e,1}^{(1)}(d_k) - \mathcal{L}_c y_k^s.\end{aligned} \tag{6.12}$$

This extrinsic information, after de-interleaving, will be used by the first turbo decoder as the *a priori* information. $\mathcal{L}_{e,2}^{(1)}(d_k)$ is also proportional to the $\frac{1}{K_1}$. Similarly, for the successive iterations of decoding, we can get the same results with (6.7) and (6.11).

So far, for the Max-Log-MAP decoder over AWGN channels, we have mathematically proved the hard decision which depends on the sign of the LLR $\widetilde{\Lambda}_1^{(i)}(d_k)$ and $\widetilde{\Lambda}_2^{(i)}(d_k)$ will not be affected by the SNR mismatch at the receiver. As a result, the Max-Log-MAP algorithm is insensitive to SNR mismatch over AWGN channels.

6.1.2 The Log-MAP algorithm

For the Max-Log-MAP algorithm, we only need to calculate the linear max function and we get the result in (6.7). For the Log-MAP algorithm, when trying to obtain the probabilities $\overline{\alpha}_k(m)$, $\overline{\beta}_k(m)$ and LLR $\Lambda(d_k)$, we need to calculate the non-linear $\widehat{\max}$ function:

$$\widehat{max}(\delta_1, \delta_2) \triangleq max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}). \tag{6.13}$$

The $\widehat{max}$ function has nonlinear part $\ln(1+e^{-|\delta_2-\delta_1|})$, thus we cannot get the similar results with (6.7), or we can say

$$\begin{aligned}\widetilde{\alpha}_k(m) &\neq \frac{1}{K_1}\overline{\alpha}_k(m),\\ \widetilde{\beta}_k(m) &\neq \frac{1}{K_1}\overline{\beta}_k(m),\\ \widetilde{\Lambda}^{(i)}(d_k) &\neq \frac{1}{K_1}\Lambda^{(i)}(d_k).\end{aligned} \tag{6.14}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The sign of the $\widetilde{\Lambda}^{(i)}(d_k)$ is not guaranteed to be the same with $\Lambda^{(i)}(d_k)$. The performance will be affected and the Log-MAP turbo decoder is sensitive to the SNR mismatch over AWGN channels.

6.2 Effect of an SNR mismatch (Fading Channels)

6.2.1 The Max-Log-MAP algorithm

For fading channels with noisy channel estimation, we will use the channel reliability factor derived in [12],

$$L_c = \frac{4\sqrt{E_s}}{N_0}\sigma_a^2\left[\sigma_m^2\left(\frac{2E_s}{N_0}\sigma_a^2+1\right)+\sigma_a^2\right]^{-1}. \tag{6.15}$$

If we consider the SNR mismatch in (6.2), the channel reliability factor at the receiver can be derived as follows:

$$\begin{aligned}\mathcal{L}_c &= \frac{4\sqrt{E_s}}{N_0'}\sigma_a^2\left[\sigma_m^2\left(\frac{2E_s}{N_0'}\sigma_a^2+1\right)+\sigma_a^2\right]^{-1},\\ &= \frac{4\sqrt{E_s}}{K_1N_0}\sigma_a^2\left[\sigma_m^2\left(\frac{2E_s}{K_1N_0}\sigma_a^2+1\right)+\sigma_a^2\right]^{-1},\\ &\neq \frac{1}{K_1}L_c.\end{aligned} \tag{6.16}$$

For the first iteration of decoding, at the first turbo decoder, by considering the *a priori* information is 0, the branch transition probability can be expressed as follows:

$$\begin{aligned}\widetilde{\gamma}_i[(z_k^s,z_k^p),m',m] &= \frac{2\sqrt{E_s}}{K_1N_0}\sigma_a^2\left[\sigma_m^2\left(\frac{2E_s}{K_1N_0}\sigma_a^2+1\right)+\sigma_a^2\right]^{-1}\\ &\quad\times\left[z_{k,r}^s x_k^s(i)+z_{k,r}^p x_k^p(i)\right].\end{aligned} \tag{6.17}$$

We cannot take the second K_1 on the right side of the equation out of the brackets without affecting other parameters. We cannot obtain the result as we did for the Max-Log-MAP algorithm over AWGN channels:

$$\widetilde{\gamma}_i[(z_k^s,z_k^p),m',m] \neq \frac{1}{K_1}\overline{\gamma}_i[(z_k^s,z_k^p),m',m]. \tag{6.18}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Therefore, we cannot get the similar result as (6.7) either,

$$\begin{aligned} \widetilde{\alpha}_k(m) &\neq \frac{1}{K_1}\overline{\alpha}_k(m), \\ \widetilde{\beta}_k(m) &\neq \frac{1}{K_1}\overline{\beta}_k(m), \\ \widetilde{\Lambda}_1^{(1)}(d_k) &\neq \frac{1}{K_1}\Lambda_1^{(1)}(d_k). \end{aligned} \tag{6.19}$$

In this case, The sign of the $\widetilde{\Lambda}_1^{(1)}(d_k)$ is not guaranteed to be the same with $\Lambda_1^{(1)}(d_k)$.

For the second turbo decoder, the extrinsic information generated from the first turbo decoder can be expressed as

$$\mathcal{L}_{e,1}^{(1)}(d_k) = \widetilde{\Lambda}_1^{(1)}(d_k) - L_{a,1}(d_k) - \mathcal{L}_c y_k^s. \tag{6.20}$$

The second term on the right side of equation $L_{a,1}(d_k) = 0$, from (6.16) and (6.19), we get

$$\mathcal{L}_{e,1}^{(1)}(d_k) \neq \frac{1}{K_1} L_{e,1}^{(1)}(d_k). \tag{6.21}$$

This extrinsic information will be used in the second decoder as the *a priori* information. In the second decoder, when calculating the transition probability:

$$\begin{aligned} \widetilde{\gamma}_i[(z_k^s, z_k^p), m', m] &= \frac{1}{2}\mathcal{L}_{e,1}^{(1)}(d_k)x_k^s(i) + \frac{1}{2}\mathcal{L}_c\Re[z_k^s]x_k^s(i) \\ &\quad + \frac{1}{2}\mathcal{L}_c\Re[z_k^p]x_k^p(i) \end{aligned} \tag{6.22}$$

From (6.16), (6.19) and (6.22), we get

$$\widetilde{\gamma}_i[(z_k^s, z_k^p), m', m] \neq \frac{1}{K_1}\overline{\gamma}_i[(z_k^s, z_k^p), m', m]. \tag{6.23}$$

Finally, we get

$$\begin{aligned} \widetilde{\alpha}_k(m) &\neq \frac{1}{K_1}\overline{\alpha}_k(m), \\ \widetilde{\beta}_k(m) &\neq \frac{1}{K_1}\overline{\beta}_k(m), \\ \widetilde{\Lambda}_2^{(1)}(d_k) &\neq \frac{1}{K_1}\Lambda_2^{(1)}(d_k). \end{aligned} \tag{6.24}$$

The sign of the $\widetilde{\Lambda}_2^{(1)}(d_k)$ is not guaranteed to be the same as that of $\Lambda_2^{(1)}(d_k)$. The performance of decoding will be affected by SNR mismatch. From the second iteration, the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

process repeats. When the number of iteration $i > 1$, we can further prove that

$$\begin{aligned} \widetilde{\Lambda}_1^{(i)}(d_k) &\neq \frac{1}{K_1}\Lambda_1^{(i)}(d_k), \\ \widetilde{\Lambda}_2^{(i)}(d_k) &\neq \frac{1}{K_1}\Lambda_2^{(i)}(d_k). \end{aligned} \tag{6.25}$$

In [9], it has been concluded that the Max-Log-MAP algorithm is insensitive to the SNR mismatch at the receiver over both AWGN and fading channels. The authors assumed the channel is known at the receiver when discussing the sensitivity of the Max-Log-MAP algorithm over fading channels. Such a result is misleading, since for fading channels, if the channel is perfectly known or $\sigma_m^2 = 0$, the channel reliability factor will be the same as that of AWGN channel,

$$L_c = \frac{4\sqrt{E_s}}{N_0}. \tag{6.26}$$

In this case, the performance of the Max-Log-MAP decoder can be insensitive to the SNR mismatch. However, if $\sigma_m^2 \neq 0$, the Max-Log-MAP algorithm is sensitive to SNR mismatch. In other words, the Max-Log-MAP turbo decoding algorithm is sensitive to SNR mismatch over fading channels with noisy channel estimates.

6.2.2 The Log-MAP algorithm

Two factors will make the Log-MAP decoder sensitive to SNR mismatch over fading channels. The first factor is that we have to consider the error variance of channel estimate σ_m^2 when deriving the channel reliability factor L_c. For both turbo decoders, similar to (6.18),

$$\widetilde{\gamma}_i[(z_k^s, z_k^p), m', m] \neq \frac{1}{K_1}\overline{\gamma}_i[(z_k^s, z_k^p), m', m]. \tag{6.27}$$

The second factor is that we need to calculate the non-linear $\widehat{\max}$ function when trying to obtain the probabilities $\overline{\alpha}_k(m)$, $\overline{\beta}_k(m)$ and LLR $\Lambda(d_k)$, the nonlinear $\widehat{\max}$ is defined as follows:

$$\widehat{max}(\delta_1, \delta_2) \triangleq max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}). \tag{6.28}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Because of the nonlinear term $\ln(1+e^{-|\delta_2-\delta_1|})$ on the right side of the equation, we have

$$\begin{aligned} \widetilde{\alpha}_k(m) &\neq \frac{1}{K_1}\overline{\alpha}_k(m), \\ \widetilde{\beta}_k(m) &\neq \frac{1}{K_1}\overline{\beta}_k(m), \\ \widetilde{\Lambda}^{(i)}(d_k) &\neq \frac{1}{K_1}\Lambda^{(i)}. \end{aligned} \tag{6.29}$$

The sign of the $\widetilde{\Lambda}^{(i)}(d_k)$ is not guaranteed to be the same as that of $\Lambda^{(i)}(d_k)$. Therefore, the Log-MAP decoder is sensitive to SNR mismatch over fading channels.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 7

Simulation Results

In this chapter, simulation results are presented to demonstrate the performance of the proposed iterative algorithm for turbo decoding with integrated channel estimation for BPSK signaling over Rayleigh fading channels. Also, simulation results are given to show the effect of SNR mismatch on the performance of the Log-MAP and Max-Log-MAP decoding algorithms using both the proposed SNR mismatch model and the conventional one.

7.1 Channel estimator performance

In all the simulated results presented in this section, it is assumed that the Rayleigh fading channel is fully interleaved. Also, in these simulations, we use a turbo code produced by the generator polynomials $(37, 21)$ (in octal form) and a code rate of $1/3$. The data frame size is $N = 420$ and the turbo decoding algorithm uses 8 iterations. In Fig. 7.1, the performance of the blind channel estimator 2 using the new metric is examined when the error variance of the initial channel estimation is $\sigma_m^2 = 0.1$. The performance of the estimator are shown at SNRs 2, 3, 4 and 5 dB with diamond, triangle (up), triangle (down) and circle lines, respectively. The horizontal axis is the number of iterations, and the vertical axis is the mean value of the error variance for one frame of data bits. We see that the error variance of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

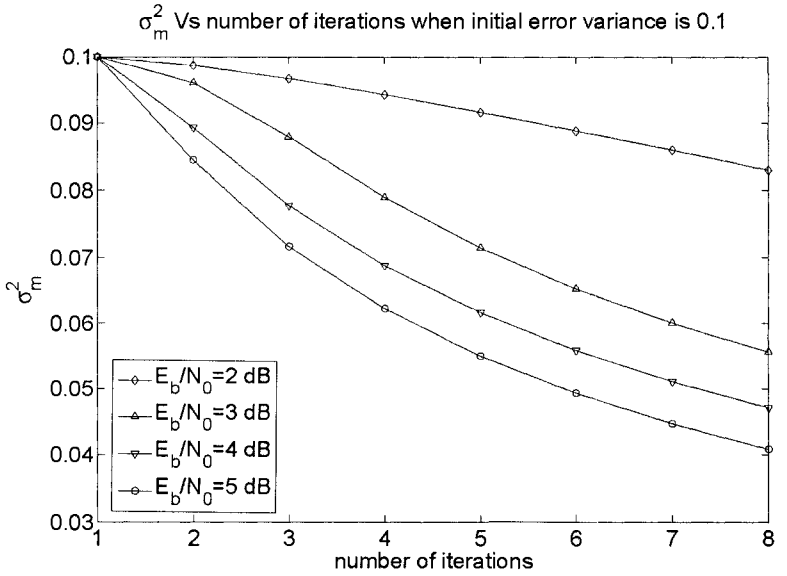

Figure 7.1: Performance of Channel estimator 2

the channel estimation is decreasing with the number of iteration of decoding. The extrinsic information provide by the channel decoder from each iteration of decoding can help the channel estimator 2 to refine its channel estimate. At high SNR, the error variance drops faster. For example, when $E_b/N_0 = 2$ dB, after 8 iterations, the error variance σ_m^2 drops to 0.08, while for $E_b/N_0 = 5$ dB, after the same number of iterations, the error variance σ_m^2 drops to 0.04. In Fig. 7.2, the performance of the blind channel estimator 2 is examined when the error variance of the initial channel estimation is 0.2. At high SNR, the channel estimator 2 needs only 5 to 6 iterations for updating its estimates to obtain estimates of the fading factors of data bits that are very close to their actual values. For example, when $E_b/N_0 = 7$ dB, after 6 iterations of re-estimation, the error variance already drops to around 0.05. In Fig. 7.3, the performance of the channel estimator 2 is examined when the error variance of initial channel estimation is 0.4. We see when the error variance of the initial channel estimation is high, the channel estimator performs even better. For example, by comparing Figures 7.2 and 7.3 with each other, after 8 iterations at $E_b/N_0 = 7$ dB,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

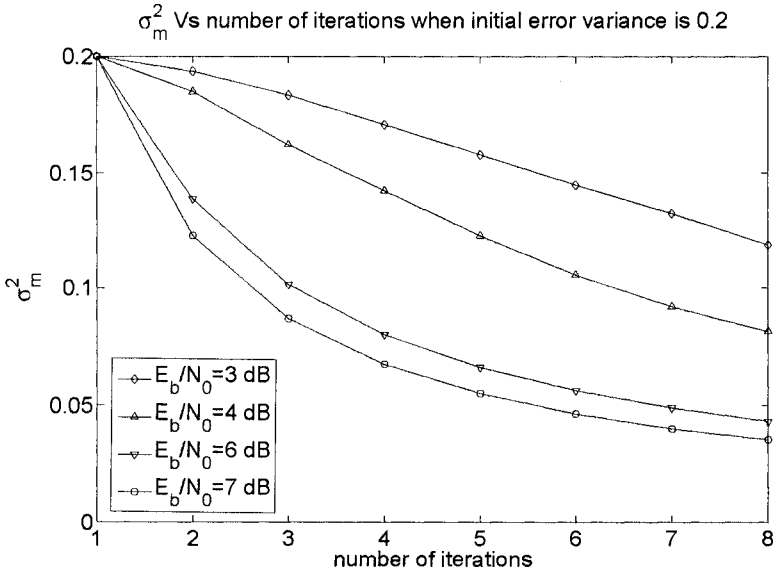

Figure 7.2: Performance of Channel estimator 2

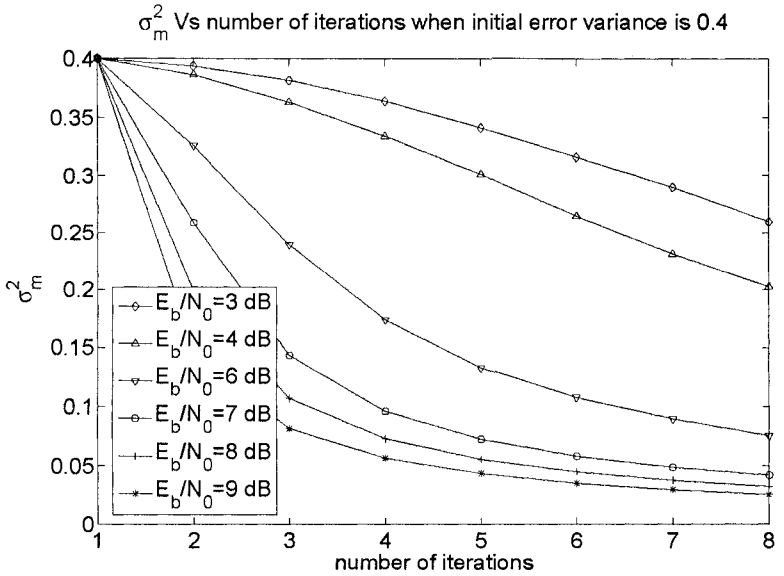

Figure 7.3: Performance of Channel estimator 2

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

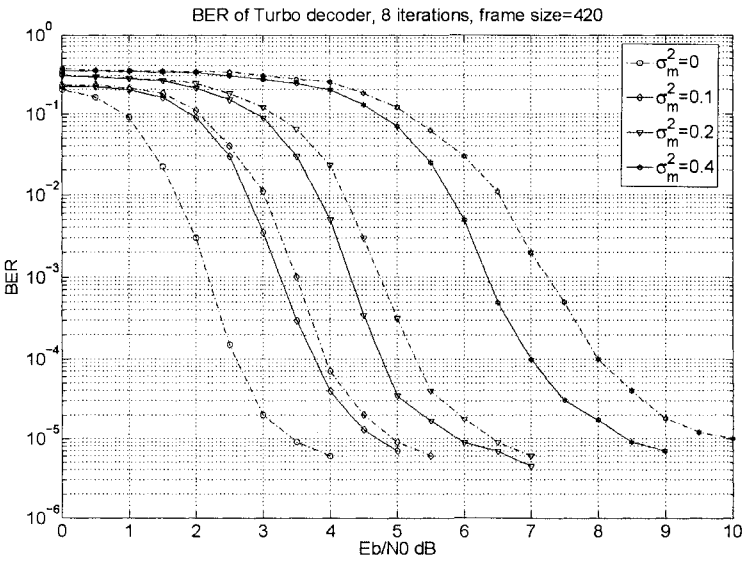

Figure 7.4: BER performance of the proposed algorithm

the error variance σ_m^2 drops from 0.2 to about 0.04 in the first graph, while in the second graph, σ_m^2 drops from 0.4 to about 0.05. At last, in Fig. 7.4, the bit-error rate (BER) performance of the new metric is examined. The horizontal axis is $SNR = E_b/N_0$ in dB, and the vertical axis is the BER performance. The dashed lines is the performance of the Log-MAP turbo decoding algorithm over fading channels without the channel estimator 2, the solid lines is the performance of the proposed turbo decoding algorithm based on the new metric and integrated with channel estimation. The diamond, triangle and hexagram shapes represent the error variance of the initial channel estimate at $\sigma_m^2 = 0.1, 0.2, 0.4$, respectively. As a reference, we also show the performance of the algorithm with perfect channel estimation using dashed circle lines. From the simulation results, it is observed that the BER performance has been improved by integrating the Log-MAP decoder with the channel estimator. Especially, when the initial estimation is very farther from the actual fading factor (i.e. $\sigma_m^2 = 0.4$), the SNR gain is higher. For example, when $\sigma_m^2 = 0.4$, at BER 10^{-4}, the gain that can be obtained is about 1 dB.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

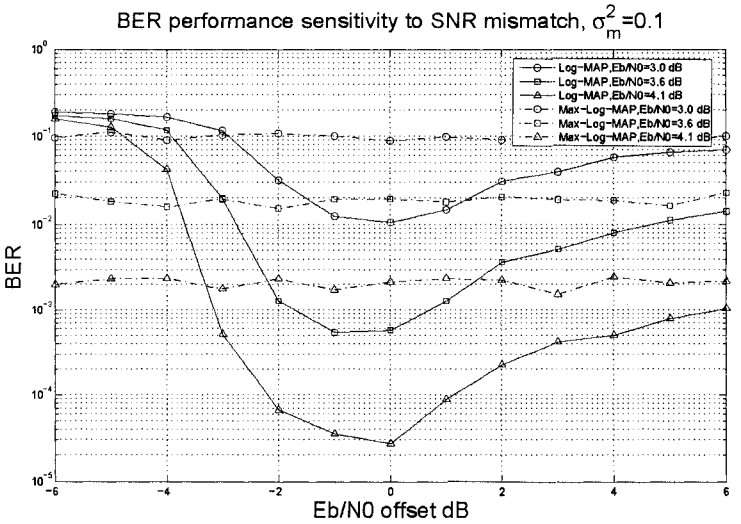

Figure 7.5: Effect of SNR mismatch (fading channels)

7.2 BER performance of turbo decoding algorithms: Effect of SNR mismatch

In the following simulations, we will consider the fading channels with imperfect channel estimation. In Fig. 7.5, the effect of SNR mismatch on the Log-MAP and Max-Log-MAP decoding algorithms over fading channels are examined when the error variance of the initial channel estimation is $\sigma_m^2 = 0.1$. The dashed lines represent the performance of the Max-Log-MAP decoder and the solid lines represent the performance of the Log-MAP decoder. The shapes of circle, square and triangle represent different SNRs. The E_b/N_0 are chosen for the case that there is no SNR mismatch. As hown in the figure, the BER performance of the Log-MAP decoder is approximately $10^{-2}, 10^{-3}, 10^{-4}$. The horizontal axis is the SNR offset from -6 dB to 6 dB. The vetical axis is the BER performance. From the simulations, it can be seen that the Max-Log-MAP decoder is sensitive to the SNR mismatch because the performance is not independent of SNR mismatch. However, the Max-Log-MAP decoder is less sensitive to the SNR mismatch than Log-MAP decoder. In Figures 7.6 and 7.7,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

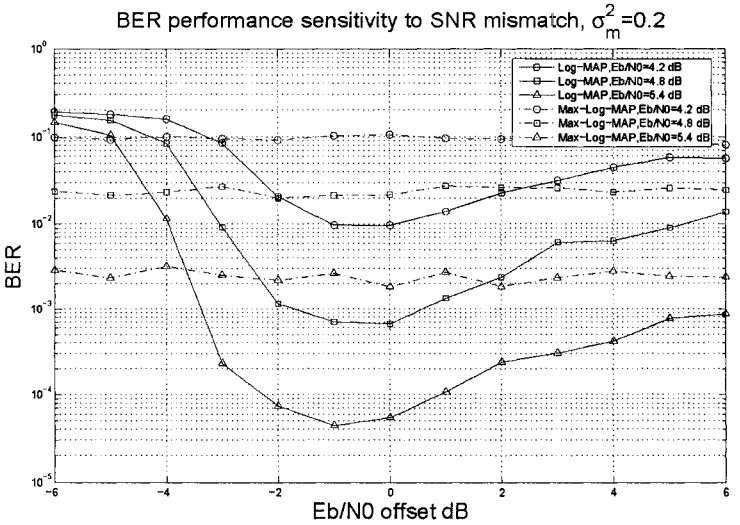

Figure 7.6: Effect of SNR mismatch (fading channels)

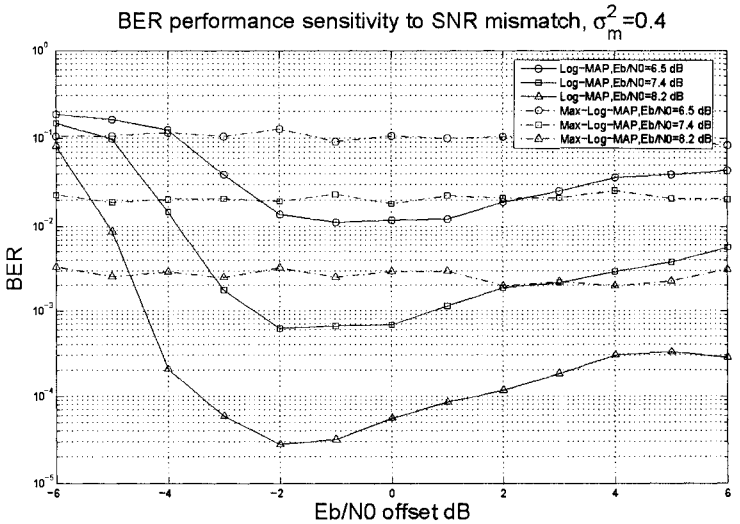

Figure 7.7: Effect of SNR mismatch (fading channels)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the effect of SNR mismatch has also been examined when the error variance of channel estimation is $\sigma_m^2 = 0.2$ and $\sigma_m^2 = 0.4$.

All the results reported in the literature by other researchers, are based on the following SNR mismatch model:

$$\frac{E_b}{N_0'} = \frac{E_b}{N_0} + offset \tag{7.1}$$

where the SNR mismatch (offset) is viewed as a constant for all the transmitted frames for a given SNR point. Therefore, each frame is assumed to experience the same SNR mismatch.

In this thesis, we propose a new SNR mismatch model as

$$\frac{E_b}{N_0'} = \frac{E_b}{N_0} + (\frac{E_b}{N_0})_{error} \tag{7.2}$$

where the $(\frac{E_b}{N_0})_{error}$ is a random variable with zero mean and variance σ_r^2. Using this new SNR mismatch model, the SNR mismatch is considered as a random variable and each frame experiences different SNR mismatch. This is a more realistic assumption, since the channel estimates can be different for each frame.

In Figures 7.8, 7.9 and 7.10 the effect of SNR mismatch on the performance of the Log-MAP and Max-Log-MAP decoding algorithms over fading channels has been examined. The error variance of the initial channel estimation are $\sigma_m^2 = 0.1$, $\sigma_m^2 = 0.2$ and $\sigma_m^2 = 0.4$. The variance of the SNR mismatch offset σ_r^2 is from 0 dB to 8 dB. The horizontal axis is the variance of the SNR mismatch σ_r^2 and the vertical axis is the BER performance. All the simulation results still verify that the Max-Log-MAP decoder is sensitive to the SNR mismatch over fading channels when the channel is not perfectly known. However, the Max-Log-MAP decoder is less sensitive to the SNR mismatch than the MAP or Log-MAP decoder. There is another scenario that it has been observed in the simulations. When the SNR mismatch is not that high, the BER performance of Log-MAP decoder is much better than that of the Max-Log-MAP decoder. However, when the SNR mismatch is high, the BER performance of the two algorithms are approximately the same. For example, in Fig. 7.10, when $\sigma_r^2 = 3$ dB, the BER performance of the Log-MAP decoder is 10^{-4} and the Max-Log-MAP is greater than 10^{-3}. When $\sigma_r^2 = 8$ dB, the performance of the Max-Log-MAP decoder is even a little better than that of the Log-MAP decoder. This result can

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

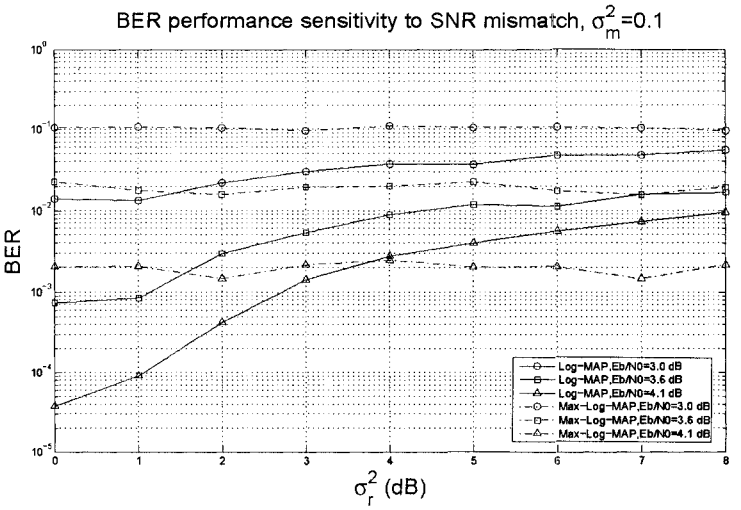

Figure 7.8: Effect of SNR mismatch with new model

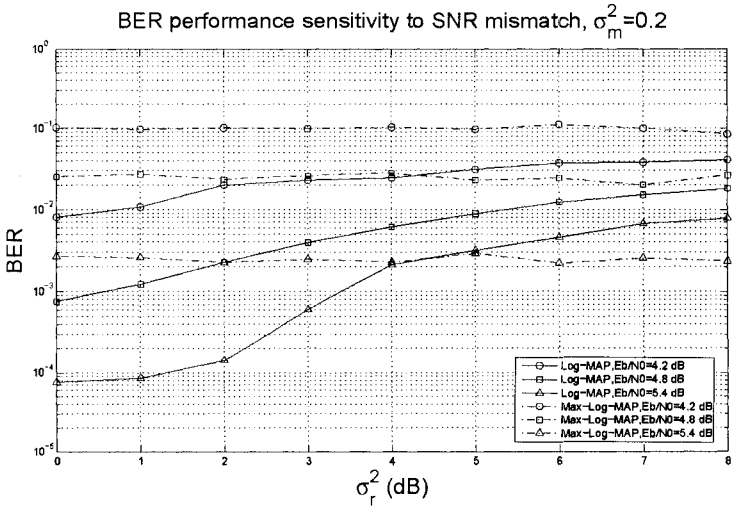

Figure 7.9: Effect of SNR mismatch with new model

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

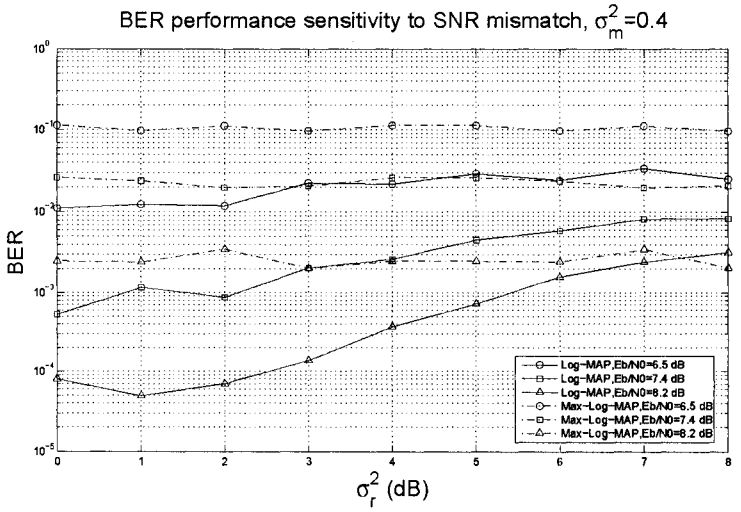

Figure 7.10: Effect of SNR mismatch with new model

only be seen when we use the new SNR mismatch model. The new SNR mismatch model is more practical than the conventional one for systems having knowledge of noisy channel estimates (instead of having perfect knowledge of the channel).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 8

Conclusion and Future Research Directions

8.1 Overview

In this thesis, a number of open problems in turbo decoding for BPSK signaling on Rayleigh flat-fading channels with noisy channel estimates have been solved.

First, we have proposed an iterative algorithm for joint turbo decoding and channel estimation over Rayleigh flat-fading channels. The proposed algorithm is based on a new turbo decoding metric which includes the uncertainty of the channel estimate. The channel decoder in this iterative algorithm provides soft extrinsic information of the transmitted data symbols which are used by the channel estimator to refine its estimate for the next iteration of decoding. The resulting iteration between the channel estimator and the turbo decoder using the new metric, with intermediate exchange of soft channel-symbol information, can improve the performance of both the channel estimator and the channel decoder. Simulation results show that the proposed algorithm can outperform conventional algorithms significantly.

As our second contribution in this thesis, we have proposed a new preprocessor (or

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

fading compensator) to compensate the effect of the fading distortion on the turbo-coded signals before feeding the turbo decoder. First, we have derived the linear MMSE estimates of the transmitted data symbols with noisy channel estimates. Then, it has been shown the linear MMSE estimator can be viewed as a normalized version of the preprocessing (fading compensation) operation used in most existing turbo decoding algorithms. Also, we have shown that the normalization has no effect on the BER performance of the turbo decoder.

We also have studied the effect of SNR mismatch on the performance of the Log-MAP and Max-Log-MAP decoding algorithms over fading channels. Based on the previous results reported by other researchers, turbo-decoding with the Max-Log-MAP decoder is independent of SNR for Rayleigh fading channels. However, we prove that the Max-Log-MAP decoder is sensitive to SNR mismatch over fading channels when the channel is not perfectly known. The result has been verified by the simulations. Also, it has been shown that the Max-Log-MAP decoder is less sensitive to SNR mismatch than the Log-MAP decoder. Using simulation results, it is also observed that for low SNR mismatch the BER performance of the Log-MAP decoder is much better than that of Max-Log-MAP decoder. Whereas, for high SNR mismatch, the BER performance of the Max-Log-MAP decoder is closer to or even better than that of Log-MAP decoder. Thus, when SNR mismatch is low and complexity is not a problem, the Log-MAP algorithm for turbo decoding in fading channels is the best choice. On the other hand, when SNR mismatch is very high, the Max-Log-MAP algorithm for turbo decoding is a better choice for simplicity.

In the end, we propose a new SNR mismatch model to examine the sensitivity of the BER performance of turbo decoding to SNR mismatch for systems having knowledge of noisy channel estimates (instead of having perfect knowledge of the channel). In the conventional SNR mismatch model, the SNR mismatch (offset) is viewed as a constant for all the transmitted frames for a given SNR point. Then, each frame experiences the same SNR mismatch. Whereas, in the new SNR mismatch model, the SNR mismatch is considered as a random variable and each frame experiences different SNR mismatch. This is a more realistic assumption, since the channel estimates can be different for each frame. Using the new SNR mismatch model, simulation results still verify that the Max-Log-MAP decoder

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

is sensitive to SNR mismatch over fading channels when the channel is not perfectly known at the receiver.

8.2 Suggestions for future studies

The results obtained in this thesis can be used as the basis for the following further research:

- *Higher-order modulation schemes*: All the results reported in this thesis are for turbo decoding for binary phase-shift keying (BPSK) signaling. It would be of theoretical and practical interest importance to consider higher-order modulation schemes (any type of digital modulation with an order of 4 or higher), for example, quadriphase shift keying (QPSK) or M-ary quadrature-amplitude modulation (M-ary QAM).

- *A new turbo decoding metric*: Moreover, in this thesis, the channel reliability factor is derived based on the assumption that the SNR is perfectly known at the receiver. In practice, since this assumption is not realistic, the parameter SNR have to be estimated. Thus, it is very important to derive a new turbo decoding metric that can take into account not only the uncertainty (or error variance) of the channel estimate but also the error variance of the SNR estimate.

- *Turbo equalization*: Finally, the concept of turbo decoding, or iterative decoding algorithm can be applied to channel equalization. In this thesis, we have considered a fully interleaved Rayleigh flat-fading channel which is a memoryless channel. It is be very important to design an iterative equalizer for turbo decoding over fading intersymbol interference (ISI) channels without having perfect knowledge of the channel.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

References

[1] L. Bahl, J. Cocke, F, Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, Vol. 20, Issue 2, pp. 284-287, Mar 1974.

[2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, Vol. 44, Issue 10, pp. 1261-1271, Oct. 1996.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," In *Proc. IEEE International Conference on Communications (ICC'93)*, Vol. 2, pp. 1064-1070, May 1993.

[4] Christian Schlegel and Lance Perez, "*Trellis and Turbo Coding* ," Wiley-IEEE Press, First Edition, 2003.

[5] B. Sklar, "A Primer on Turbo Code Concepts," *IEEE Communications Magazine*, Vol. 35, Issue 12, pp. 94-102, Dec. 1997.

[6] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," In *Proc. 1994 IEEE Global Telecommunication Conference (Globecom'94)*, Vol. 3, pp. 1298-1303, 1994.

[7] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," In *IEEE Proc. International Conference on Communications (ICC'95)*, Vol. 2, pp. 1009-1013, June 1995.

[8] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, Vol. 42, Issue 2, pp. 429-445, March 1996.

[9] A. Worm, P. Hoeher, and N. Wehn, "Turbo-decoding without SNR estimation" *IEEE Communications Letters*, Vol. 4, Issue 6, pp. 193-195, June 2000.

[10] S. Talakoub and B. Shahrrava, "Turbo equalization with iterative online SNR estimation," In *Proc. IEEE Wireless Communications and Networking Conference (WCNC'05)*, Vol. 2, pp. 1097-1102, March 2005.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[11] P. Frenger, "Turbo decoding for wireless systems with imperfect channel estimates," *IEEE Transactions on Communications*, Vol. 48, Issue 9, pp. 1437-1440, Sept. 2000 .

[12] P. Frenger, "Turbo decoding on Rayleigh fading channels with noisy channel estimates," In *Proc. 49th IEEE Vehicular Technology Conference (VTC'99)*, Vol. 2, pp. 884-888, May 1999.

[13] Hyundong Shin and Jae Hong Lee., "Channel reliability estimation for turbo decoding in rayleigh fading channels with imperfect channel estimates," *IEEE Communications Letters*, Volume 6, Issue 11, pp. 503-505, Nov. 2002.

[14] J. Hagenauer, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Info. Theory*, Vol. 42, No. 2, pp. 429-445, March 1996,.

[15] J. Hagenauer, "The turbo principle: Tutorial and state of the art," In *Proc. Int. symposium on Turbo Codes and Related Topics*, pp. 1-11, Brest, France, Sept. 1997.

[16] S.A. Barbulescu, *"Iterative decoding of turbo codes and other concatenated codes,"* PH.D. dissertation, School of Elect. Eng., Faculty of Engineering., Univ. South Australia, Feb. 1996.

[17] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Info. Theory*, Vol. 42, No. 2, pp. 409-428, March 1996.

[18] E.K. Hall and S.G. Wilson, "Design and analysis of turbo codes on Rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 2, pp. 160-174, Feb. 1998.

[19] E.K. Hall and S.G. Wilson, "Turbo codes for noncoherent channels," *IEEE Communicaiton Theory, Mini-Conf.*, pp.66-70, Phoenix, AZ, Nov. 1997.

[20] D. Divsalar and F. Pollara, "On the design of Turbo Codes," TDA progress rep. 42-123, Jet Propulsion Lab., pp. 99-121, Pasadena, CA, Nov 1995.

[21] Hyundong Shin and Jae Hong Lee, "Effective Free Distance of Turbo Codes, *Electronics Letters*, Vol. 32, No. 5, p 445-446, Feb. 1996.

[22] M.C. Valenti and B.D. Woerner, "Refined channel estimation for coherent detection of turbo codes over flat-fading channels," *Electronics Letters*, Vol. 34, pp 1648-1649, Aug. 1998.

[23] M.C. Valenti and B.D. Woerner, "Iterative channel estimation and decoding of pilot symbol assisted turbo codes over flat-fading channels," *IEEE Journal on Selected Areas in Communications*, Vol. 19, Issue 9, pp. 1697-1705, Sept. 2001.

[24] B. Mielczarek and A. Svensson,"Improved iterative channel estimation and turbo decoding over flat-fading channels," In *Proc. 2002 IEEE Vehicular Technology Conference (VTC'02)*, Vol. 2, pp. 975-980, Sept. 2002.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[25] M.C. Valenti and B.D. Woerner, "A bandwidth efficient pilot symbol technique for coherent detection of turbo codes over fading channels," In *Proc. 1999 IEEE Military Communications Conference (MILCOM'99)*, Vol. 1, pp. 81-85, Atlantic City, NJ, Nov. 1999.

[26] M.C. Valenti and B.D. Woerner, "SNR mismatch and online estimation in turbo decoding," In *Proc. IEEE Vehic. Technol. Conf.*, Ottawa, Canada, May 1998.

[27] M.C. Valenti and B.D. Woerner, "Performance of turbo-codes in interleaved flat fading channels with estimated channel state information," In *Proc. IEEE Vehic. Technol. Conf. (VTC'98)*, pp. 66-70, Ottawa, Canada, May 1998.

[28] T. A. Summers and S. G. Wilson, "SNR mismatch and online estimation in turbo decoding," *IEEE Trans. Commun.*, Vol. 46, pp. 421-423, April 1998.

[29] M. Jordan and R. Nichols, "The effects of channel characteristics on turbo code performance," In *Proc. 1996 IEEE Military Communications Conference (MILCOM'96)*, pp. 17-21, McLean, VA, Oct. 1996.

[30] M. C. Reed and J. Asenstorfer, "A novel variance estimator for turbo code decoding," In *Proc. Int. Conf. on Telecommun.*, pp. 173-178, Melbourne, Australia, Apr. 1997.

[31] J. K. Cavers,"An analysis of pilot symbol assisted modulation for Rayleigh fading channels, *IEEE Trans. Veh. Technol.*, Vo.l 40, pp. 686-693, Nov. 1991.

[32] J. K. Cavers and M. Liao, "A comparison of pilot tone and pilot symbol techniques for digital mobile communications, In *Proc. 1992 IEEE Global Telecommunication Conference (Globecom'92)*, pp. 915-921, Orlando, FL, Dec. 1992.

[33] L.-D. Jeng, Y. T. Su, and J.-T. Chiang, "Performance of turbo codes in multipath fading channels," In *Proc IEEE Vehicular Technology Conf.*, pp. 61-65, Ottawa, Canada, May 1998.

[34] K.L.Li and S.W.Cheung, "Modified MAP algorithm incorporated with PSA technique for turbo codes in Rayleigh fading channels," *Electronics Letters*, Vol. 35, pp. 537-539, April 1999.

[35] A. Ramesh, A. Chockalingam, and L.B.Milstein, "SNR estiamtion in generalized fading channels and its application to turbo decoding," In *Proc. International Conference on Communications (ICC'01)*, pp. 1094-1098, Helsinki, Finland, June 2001.

[36] P. Ho and Jae Hyung Kim, "Pilot symbol-assisted detection of CPM schemes operating in fast fading channels, *IEEE Transactions on Communications*, Vol. 44, Issue 3, pp. 337 - 347, March 1996.

[37] J.K. Tugnait, "Blind channel estimation and adaptive blind equalizer initialization," In *Proc. International Conference on Communications (ICC'91)*, pp. 1388-1392, Vol. 3, June 1991.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[38] H. Shin, S. Kim, and J. H. Lee, "Turbo decoding in a Rayleigh fading channel with estimated channel state information," In *Proc. 2000 IEEE Vehicular Technology Conference (VTC'00 Fall)*, pp. 1358-1363, Boston, MA, Sept. 2000, .

[39] Yu Fei, "*Turbo Code Demo*," http://www.ee.vt.edu/ yufei/turbo.html.

[40] Gordon L. Stüber., "*Principles of Mobile Communication*," Kluwer Academic Publishers, 2001.

[41] Theodore S. Rappaport, "*Wireless Communicaitons: Principles and Practice*," Prentice Hall PTR, 2nd Edition, 2001.

[42] Ali H. Sayed, "*Fundamentals of Adaptive Filtering*," Wiley-IEEE Press, 2003.

[43] G.Ungerboeck, "Channel coding with multilevel phase signals," *IEEE Trans. Inform. Theory*, Vol. 28, pp.55-67, January 1982.

[44] J. G. Proakis, "*Digital Communications*," McGraw-Hill, 3rd Edition, 1995.

[45] J. G. Proakis, "Probabilities of error for addaptive reception of M-phase signals," *IEEE Trans. Communications*, Vol. COM-16, No. 1, pp. 71-81, Feb. 1968.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

VITA AUCTORIS

Yibo Zhou was born in Shenyang, China, in 1980. He received his Bachelor of Engineering degree with honors from the Radio Engineering Department of Southeast University in 2002. He is currently a candidate for the Master of Applied Science in Electrical and Computer Engineering at the University of Windsor and hopes to graduate in Fall 2006.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.