Performance of Serial Concatenated Convolutional Codes with MSK over ISI Wireless Channels

Le Feng

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

January 2003

0-612-77968-8

ABSTRACT

Performance of Serial Concatenated Convolutional Codes with MSK over ISI Wireless Channels

Le Feng

Serial concatenated convolutional code (SCCC) with minimum shift keying (MSK) modulation technique over AWGN channel and ISI channel in wireless communication systems are investigated and analyzed. Iterative log-MAP (Maximum A Posteriori Probability) decoding algorithm is applied. The design criteria for serial concatenated codes are given. For AWGN channel, the simulation results show SCCC with MSK is an alternative to turbo codes and the performance is superior at high SNR. The types of the interleaver, the interleaver size, the number of iterations and different algorithms that influence the performance are investigated and simulated. For intersymbol interference (ISI) channel, which is caused by multipath, limited bandwidth and motion in wireless communication systems, zero-forcing and least mean square equalization algorithms are used to compensate for the effect of ISI. It is shown that least mean square algorithm has better performance than zero-forcing algorithm.

Dedicated to my dearest parents ......

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere appreciation to my supervisor Dr. M. R. Soleymani for giving me invaluable support, guidance and encouragement. I am grateful for his patience and kindness in answering my questions. Without his continuous encouragement and support, I would not succeed with my research.

I would also like to thank Dr. Xiangming Li who generously provided me a lot of useful suggestions and help in my thesis. Furthermore, I want to thank Yingzi Gao, Yuying Dai, Jianfeng Weng, Qing Zhang, Qing Ma, Bing Qiu and Hui Zhang who helped me a lot with my problems. My thanks go to all my friends and labmates who have provided me all kinds of help during my study.

Finally, I would like to thank my parents for their love and encouragement. I could not have completed my degree without their continuous love and immeasurable support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

APP	A Posteriori Probability
BER	Bit Error Rate
CDMA	Code Division Multiple Access
CMA	Constant Modulus Algorithm
CPE	Continuous Phase Encoder
CPM	Continuous Phase Modulation
FSK	Frequency Shift Keying
FSM	Finite State Machine
ISI	Inter-Symbol Interference
LDPCC	Low Density Parity Check Codes
LMS	Least Mean Square
MAP	Maximum A Posterior (probability)
ML	Maximum Likelihood
MM	Memoryless Modulator
MSK	Minimum Shift Keying
PCCC	Parallel Concatenated Convolutional Codes
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
RSC	Recursive System Convolutional (code)
SCBC	Serial Concatenated Block Codes
SCCC	Serial Concatenated Convolutional Codes
SISO	Soft-In-Soft-Out
SNR	Signal-to-Noise Ratio
SOVA	Soft Output Viterbi Algorithm
TCM	Trellis Coded Modulation
TDMA	Time Division Multiple Access

Chapter 1

Introduction

Digital communication systems play an irreplaceable role in the military, government and our daily life. Digital communication is simply the practice of exchanging information using finite sets of signals that are in the form of electrical waveforms or electro-magnetic fields. Figure 1.1 illustrates the functional diagram and the basic elements of a communication system. The *information source* generates messages (words, code symbols, etc.) bearing information to be transmitted. The *source encoder* is designed to convert the source output sequence into a sequence of binary digits with minimum redundancy. The purpose of the *channel encoder* is to minimize transmission errors caused by the channel by adding redundancy to the information sequence. The *digital modulator* maps the encoded digital sequences into a train of analog waveforms suitable for transmission. *Channel* examples are wire lines, optical fiber cables, microwave radio links, magnetic recording media, etc. At the receiving end, the *digital demodulator* generates a binary or analog sequence at its output as the best estimates of the modulated sequence. The *channel decoder* makes an estimate of the transmitted message based on the encoding rule and the characteristics of the channel. The *source decoder* transforms the estimated sequence from the channel decoder into an estimate of the source output and delivers it to the *information sink*[1][2][3].

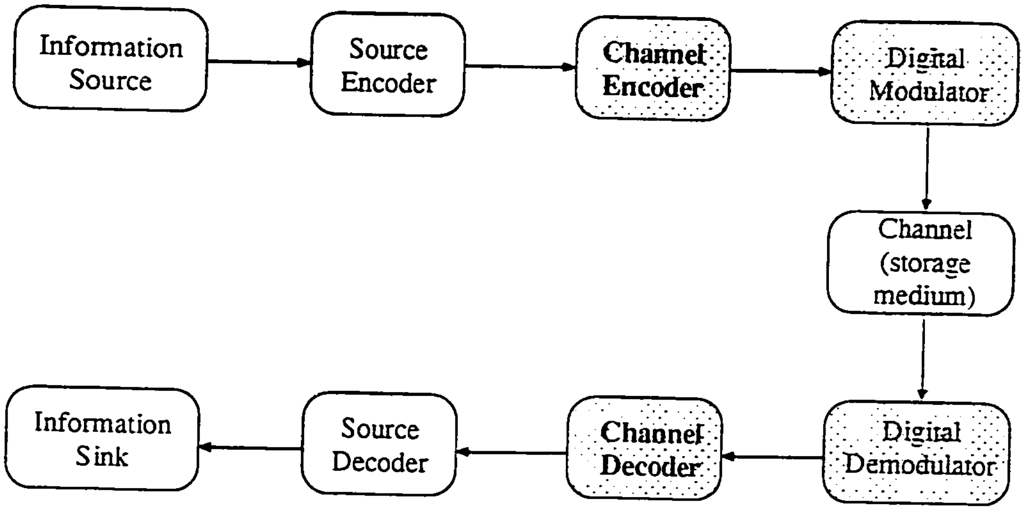


Figure 1.1: General model of a digital communication system

Increasing demand for higher data rates through efficient and reliable digital data transmission and storage systems has motivated the development of novel communication technology and information services. More and more research has been done on every functional block. Among these functional blocks, channel coding and modulation techniques for wireless communication systems are the concern of this thesis.

1.1 Channel Coding Background

The main challenge of communication systems is to recover the original information at the receiver as reliably as possible. In 1948, Shannon demonstrated in his landmark paper [4] that by proper encoding of the information, errors caused by a noisy channel or storage medium can be reduced to any desired level if the data transmission rate r_b (in bits/sec) from the source encoder is smaller than the channel capacity C. The concept of channel capacity C is defined as the maximum rate at which information can be transmitted over a noisy channel. For an Additive White Gaussian Noise (AWGN) channel, the capacity is given by the Shannon-Hartley formula: $C = Blog_2(1 + \frac{S}{N})$ $bits/sec$, where B is the system bandwidth and $\frac{S}{N}$ denotes the *signal-to-noise ratio* (SNR) [1]. This formula shows that noise sets a limit on the data rate not on the error probability. If the data rate $r_b > C$, it is not possible to design a code that can achieve an arbitrarily small error probability. Therefore, the primary objective of error control coding is to maximize the reliability of transmission within the constraints of signal power, system bandwidth and complexity of the implementation. A great deal of effort has been devoted to the problem of devising efficient encoding and decoding methods for error control in a noisy environment [1][3].

Block codes and *convolutional codes* are two different types of codes most commonly used today. The encoder of a block code is memory-less with implementation of a combinational logic circuit, while the encoder of a convolutional code contains memory which can be implemented by a sequential logic circuit. In the class of block codes, Hamming code [5] and Golay code [6] are single-error-correcting and multi-error-correcting perfect codes, respectively and they have played an important role in coding history. Binary BCH codes (discovered by Hocquenghem in 1959 [7] and independently by Bose and Chaudhuri in 1960 [8]) and non-binary BCH codes, Reed-Solomon (RS) codes (introduced by Reed and Solomon in 1960 [9]) are powerful and widely used random error-correcting cyclic codes (subclass of block codes). Block codes are widely used for error control in computer main/control memories and data storage systems, such as magnetic drums, photo-digital storage systems, magnetic bubble memories. Convolutional codes were first introduced by Elias in 1955 [10] as an alternative to the block codes. Convolutional codes with Viterbi decoding have found many applications in space and satellite communication systems, cellular mobile, digital video broadcasting, etc [1][2][3]. In 1982, another pioneering work by Ungerboeck [11] introduced *trellis-coded modulation* (TCM). It has become an effective scheme of combining coding and modulation techniques for bandlimited channels. By using the TCM approach, significant coding gain can be achieved without bandwidth expansion [3].

1.2 Parallel Concatenated Codes

The introduction of digital mobile telephony and wireless data communications has made lower transmission power and higher network capacity necessary. Hence the use of more advanced channel codes has become essential and highly demanded so that the system can provide both improved performance and better bandwidth efficiency. Noise and finite bandwidth are the two major limitations of the real channel which influence the design of channel coding and modulation schemes of the system. A recent landmark development is *Turbo Error Control Coding* invented by Berrou, Glavieux and Thitimajshima in 1993 [12]. The performance of Turbo code, in terms of *Bit Error Rate* (BER), is very close to Shannon's limit [12][13]. (*Low Density Parity Check* (LDPC) Codes invented by R.G.Gallager [14] in 1962, and rediscovered by D.J.C.Mackay have been shown to have a performance better than that of Turbo Codes with lower complexity. However, these codes are not subject of this thesis).

The discovery of turbo codes is one of the most significant breakthroughs in the field of channel coding since they offer near capacity performance in power-limited channels such as deep space, satellite and wireless channels. A large amount of research and effort have been expended on how to further improve the performance of turbo code and to apply turbo codes in current communication systems so as to improve system performance. NASA's next generation deep space transponder will utilize turbo codes [15]. Another application is 3rd generation of wideband *code division multiple access* (W-CDMA) cellular mobile systems [16] where turbo codes have been recommended for both forward and reverse supplemental channels. Moreover turbo codes have been accepted by ETSI as the standard for *digital video broadcasting* (DVB). For 3rd Generation Partnership Project (3GPP) and INMARSAT's new mobile multimedia service, turbo codes are proposed to be used [1].

Turbo codes, originally suggested by Berrou et al. [13], consist of a parallel concatenation of two rate 1/2 *recursive systematic convolutional* (RSC) encoders with an interleaver between them. The first encoder receives an original copy of the input sequence, while the second one receives a permuted version of the input sequence. Thus turbo codes are also referred to as *Parallel Concatenated Convolutional Codes* (PCCC). The interleaver will do the permutation of the input sequence. The turbo decoder consists of the serial concatenation of two decoders separated by an interleaver. The decoding algorithm is based on either a *Maximum A Posterior probability* (MAP) algorithm or a *Soft Output Viterbi Algorithm* (SOVA). Iterative decoding process instead of maximum likelihood (ML) trellis decoding gives near-ML performance with reasonable complexity [1]. It has been shown by Berrou et al. in [3][12] that turbo coding could achieve a BER lower than 10^{-5} at $E_b/N_0 = 0.7dB$ with code rate of 1/2 which means 0.7 dB from Shannon's limit. The concept of turbo codes has been extended to block codes in [17].

1.3 Serial Concatenated Codes

Shortly after the introduction of turbo codes, Serially Concatenated Convolutional (Block) Codes (SCCC or SCBC) were found to be appropriate for iterative decoding.

As an alternative to parallel concatenated codes, Benedetto et al. have demonstrated in that the interleaver gain can be made significantly higher than that of turbo codes and lower changes of slope in the bit error probability curves than turbo codes. They also showed that low-complexity iterative decoding algorithm yields performance close to the Shannon limit (0.76 dB from Shannon capacity limit) [18]. A 4-state encoder with overall code rate 1/3 SCCC is one of the two candidates for the 3GPP. The inner code is a rate 1/2 RSC code with generator matrix $G(D) = \begin{bmatrix} 1 & \frac{1+D^2}{1+D+D^2} \end{bmatrix}$. The outer code is a rate 2/3 code obtained by puncturing every other parity-check bit of a rate 1/2 RSC code. The generator matrix for the outer code is the same as for the inner code [1].

A serial concatenated code consists of the cascade of an outer encoder, an interleaver permuting the outer codewords and an inner encoder whose input words are the permuted outer codewords. Iterative MAP or SOVA algorithm can also be applied to the serial concatenated codes. On the other hand, in many communication systems, such as satellite and mobile communication systems, non-linear power amplifiers that require a low peak-to-average power ratio for the modulated signal are employed. *Continuous Phase Modulation* (CPM) is a constant envelope modulation technique and thus a good choice for such systems. Bandwidth efficiency can be achieved by combining the serial concatenated codes with CPM modulation.

1.4 Thesis Outline

In this thesis, the focus is on investigating the performance of SCCC with different modulation schemes over various types of channels.

In Chapter 2, the fundamentals of serial concatenated codes are explained. Iterative log-MAP and Max-log-MAP decoding algorithms for serial concatenated convolutional codes are studied in detail. Analysis and design guidelines for SCCC are presented. For bandwidth efficient transmission using M-ary modulation technique, a converting algorithm is applied for the first time in the demodulation of SCCC with M-ary PSK modulation. The simulation result shows that the converting algorithm is a good trade-off between the computational complexity and the BER performance.

In Chapter 3, the performance of serial concatenated convolutional code with *Minimum Shift Keying* (MSK) over AWGN channel is investigated and analyzed. Iterative log-MAP decoding for the outer and the inner decoder is studied in detail. The performance comparison of SCCC with MSK modulation and Turbo convolutional codes with BPSK modulation reveals that SCCC with MSK outperforms turbo code at high SNR. Factors that influence the performance such as interleaver type, number of iterations and decoding algorithms are evaluated and analyzed.

The performance of decoding SCCC with MSK over ISI channel is also studied. ISI channel is a commonly encountered channel model in wireline, radio, satellite and mobile communication systems and it degrades the performance significantly. Considering the complexity of decoding SCCC with MSK over ISI channel, equalizers using Least Mean Square Error (LMSE) and Zero-forcing algorithms to compensate for the distortion caused by ISI channel are deployed in Chapter 4.

In Chapter 5, conclusions are made and directions for future work are suggested.

Chapter 2

Serial Concatenated Convolutional Code

In this chapter, we first give a brief introduction of the turbo code, followed by the introduction of the component encoder, interleaver and design considerations of serial concatenated convolutional code. Second, the general iterative decoding algorithm of SCCC is studied in more detail. Finally, the converting algorithm of transforming the probability density of non-binary symbol to the probability density of binary symbol. which is applied in SCCC with M-ary PSK modulation is presented.

2.1 Introduction

It is well known that there always exists a trade-off between the coding gain and system complexity. G.D.Forney proposed the scheme of serial concatenated codes with two levels of coding. an inner and an outer code, linked by an interleaver [18][19]. The initial scheme had a short convolutional code as the inner code and a long high-rate low redundancy nonbinary Reed-Solomon code as the outer code.

The decoding scheme would be simple maximum-likelihood decoding of the inner code and a powerful algebraic error-correction algorithm for the outer code using reliability information from the inner decoder. This approach has evolved as a standard for those applications where very high coding gains are needed such as space and deep-space applications [18]. The primary reason for using a concatenated code is to achieve a low error rate with an overall decoding complexity lower than that required for a single code of the corresponding performance. The relatively low complexity is obtained by decoding each component code separately [20]. Since turbo code is very popular and has been applied in many systems, let us discuss turbo code in some detail first and then introduce serial concatenated code.

2.1.1 Turbo Code

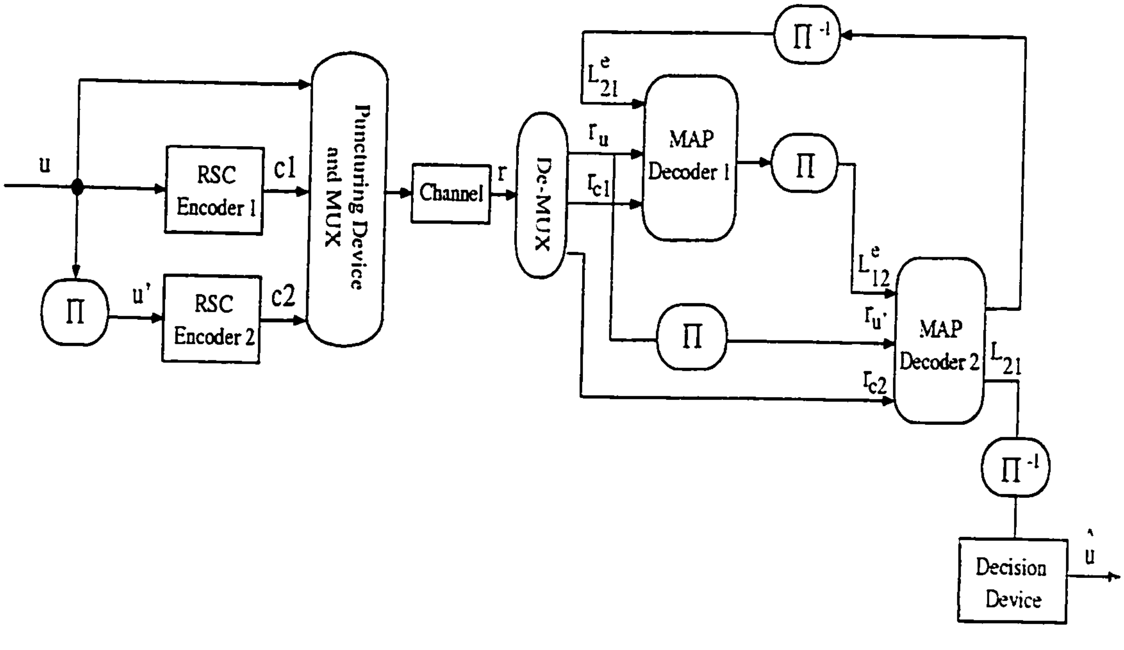


Figure 2.1: Block diagram of encoder and iterative decoder for turbo code

Figure 2.1 illustrates the block diagram of the encoder and the iterative decoder of a parallel concatenated convolutional code, i.e., turbo code. Taking the advantage of the classical concatenated coding scheme, the main ingredients that formed the basis of the invention of turbo codes and serial concatenated codes are two or more constituent codes (CCs) and an interleaver (or interleavers).

For the turbo codes, the same input information will be encoded twice but in a different order. The two constituent encoders are arranged in a parallel way and can be the same or different. The original design scheme [12] was formed by two identical recursive systematic convolutional (RSC) encoders separated by a non-uniform interleaver. The first component encoder operates on the input sequence μ directly. Since the two encoders are systematic ones, the two outputs of the first encoder will be the information μ and the parity check sequence $c1$. Then the information sequence μ goes to the interleaver to generate a "scrambled" information sequence μ' and feeds the second encoder. The output of the second encoder is the parity check sequence $c2$. The information sequence μ, parity check sequences $c1$ and $c2$ are passed to the puncturing device and multiplexer and then are transmitted over the channel.

The puncturing device is to delete selected parity bits to reduce the coding overhead based on the puncturing matrix in order to obtain the desired higher code rate. The code rate of the original turbo code without puncturing is 1/3. To achieve a code rate of 1/2, we might delete all the even parity bits from the first encoder and all the odd parity bits from the second one. In a similar way, it is possible to design code rates of 2/3, 3/4, 6/7, etc. The performance of higher code rate turbo code will degrade compared to the lower rate codes [1].

One of the principal factors that has great impact on the performance of the convolutional codes is the minimum distance of the codeword. The interleaver between the CCs is important in designing the encoder and decoder system to achieve high performance. A good interleaver can break the low weight input sequences so as to increase the minimum distance of the overall turbo code. Thus, the error floor caused by the minimum distance can be lowered and the overall performance of the turbo code can be improved.

The decoding process for the turbo codes is an iterative decoding algorithm based on the maximum a posteriori probability (MAP) algorithm. In Figure 2.1, the decoding part consists of two component decoders serially concatenated via an interleaver. The first MAP decoder takes as input the received information sequence r_u, the received parity sequence r_{c1} generated by the first encoder and the deinterleaved extrinsic information (the soft output information from all the other coded bits in the code sequence) produced by decoder2 L_{21}^e as a priori. Decoder1 then generates the extrinsic estimates of the information and passes the interleaved sequence L_{12}^e to decoder2. Note that one does not send the interleaved information sequence from the second encoder, just the parity check bits. Therefore, for decoder2, the other two inputs are the interleaved received information sequence $r_{u'}$ and the received parity check sequence r_{c2} produced by the second encoder. At the final iteration, the a posteriori probabilities of the information sequence from decoder2 will be deinterleaved and passed to the hard decision device to make final estimates of the information sequence $\hat{u}$. The simulation results in [13] show that with a 16-state RSC encoder, nonuniform interleaver and 18 iterations, the BER is lower than 10^{-5} at $E_b/N_0 = 0.7\,dB$.

2.1.2 Serial Concatenated Convolutional Code (SCCC)

It has been shown in [18] that the serial concatenated convolutional code has significantly lower change of slope in the BER than PCCC with an equal input decoding delay. Figure 2.2 shows the block diagram of a general model of coding/decoding of the serial concatenated codes. The outer encoder and the inner encoder are linked by an interleaver. Iterative max*- log-MAP (call max-log-MAP with correction term as max*-log-MAP) decoding algorithm applied in both the outer decoder and the inner decoder of the serial concatenated codes demonstrates a good trade-off between performance and complexity. An iterative log-MAP algorithm is employed in simulation to get better performance. The decoding algorithm and the factors that influence the design criteria are discussed in detail in the following section.

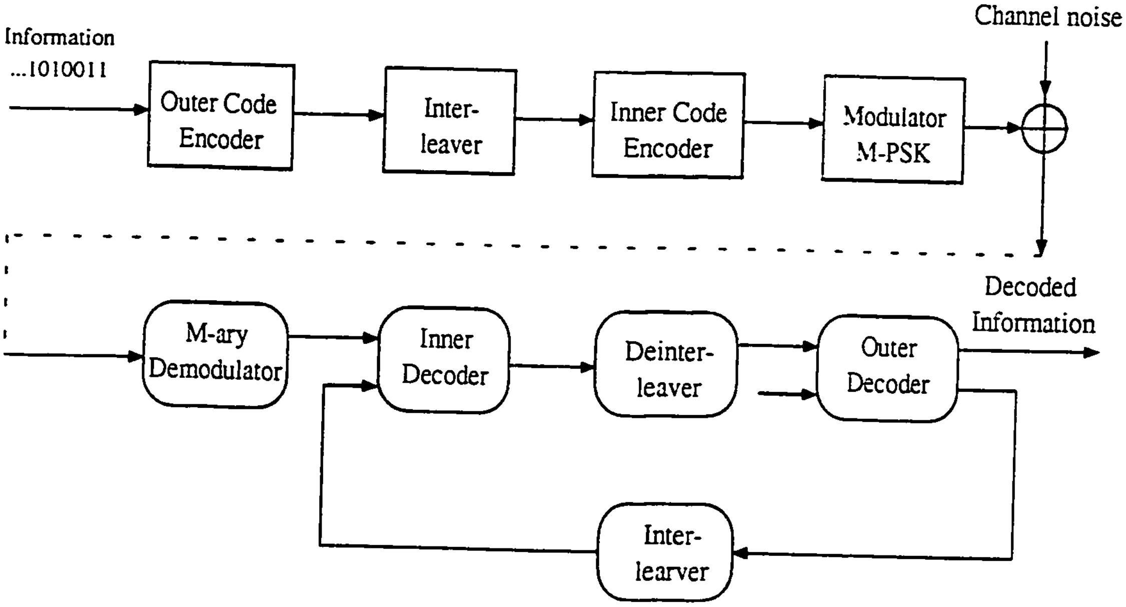


Figure 2.2: General code/decode model of serial concatenated codes

To achieve higher bandwidth efficiency, M-ary (M>2) modulation techniques are widely used in practical systems [16]. To get soft-decision values from the M-ary received symbols at relatively low complexity, namely transforming symbol probability to bit probability, a converting algorithm, which can obtain soft values of the coded bits to be used by the SCCC inner decoder is introduced.

2.2 Design of SCCC

For serial concatenated codes, the CCs can be either block codes resulting in *serial concatenated block codes* (SCBC) or convolutional codes resulting in *serial concatenated convolutional codes* (SCCC). For practical applications, SCCCs are preferred to SCBCs. One reason is that the maximum a posteriori algorithms are less complex for convolutional codes than for block codes. Another reason is that the interleaver gain can be greater for convolutional CCs, provided they are suitably designed [18]. In this thesis, therefore, the focus is on studying the performance and decoding algorithms of SCCC with applications to coded MSK and M-ary PSK modulation schemes.

2.2.1 The Component Encoder

In the original turbo code encoder [13], the two constituent encoders are connected in a parallel manner, while for the serial concatenated codes, the two encoders are combined in a serial manner. The structure of SCCC encoder consists of an outer encoder with rate $R_o = k/p$ and an inner encoder with rate $R_i = p/n$ and an interleaver between them. The overall concatenated code rate is $R = R_o \cdot R_i = k/n$. Figure 2.3 illustrates the structure of the component encoders for a given SCCC. The information frame U_o is encoded by the outer encoder which produces the outer codeword C_o. The codeword C_o maps into U_i after being permuted by the interleaver. U_i is fed to the inner encoder as the input information sequence.

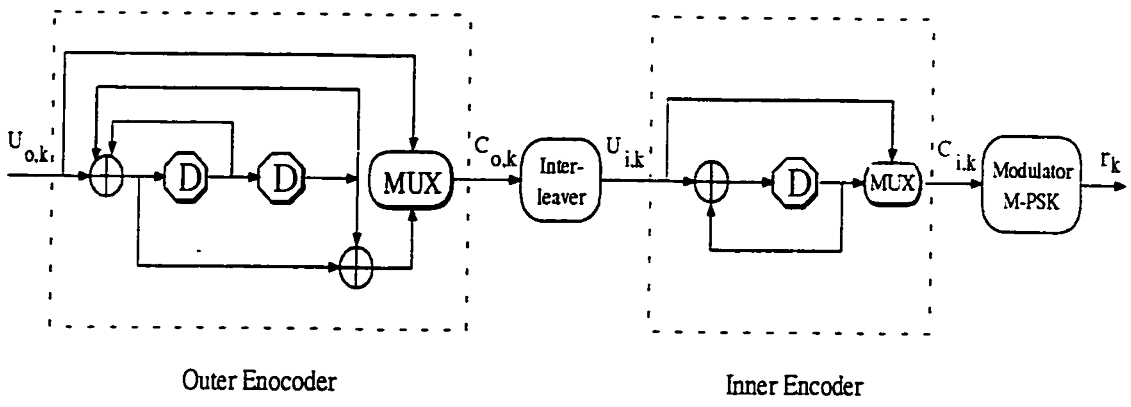


Figure 2.3: Encoder structure of serial concatenated convolutional code

The outer encoder can be either a recursive or a non recursive code with memory v_o and code rate k/p. The inner encoder must be a convolutional recursive encoder with memory v_i and coder rate p/n. In the thesis, for example, we use RSC code of code rate $R_o = 1/2$ with memory $v_o = 2$ as the outer encoder and RSC code of $R_i = 1/2$ with memory $v_i = 1$ as the inner encoder. Figure 2.3 shows an example of the structure of an outer encoder concatenated with an inner encoder.

2.2.2 Interleaver

The interleaver plays an important role in the design of both turbo codes and serial concatenated codes. The choice of interleaver has a great effect on the performance of these codes. Interleaving is the process of rearranging the order of a data sequence in a one-to-one deterministic mapping format. The inverse of this process is called deinterleaving which restores the received sequence to its original order [1].

The performance of any binary code is dominated by its free distance, d_{free} (the minimum Hamming distance between codewords, which coincides with the minimum Hamming weight of a nonzero codeword for linear codes) and its multiplicity [21]. The main purpose of the interleaver is to break low weight input sequences, and hence increase the code's Hamming distance or reduce the number of codewords with small distance in the code distance spectrum. Another function of an interleaver is to spread out burst errors. The interleaver provides new order information data (output codewords of the outer encoder) to the inner encoder. For the decoding process, one decoder decorrelates the input from the other decoder so that an iterative decoding algorithm based on "uncorrelated" information exchange between the two decoders can be applied [1].

In general, the interleaver size N is much larger than the encoder memory v. A long block code can be constructed from small memory convolutional encoder by an interleaver [1]. Usually the performance can be improved by increasing interleaver size. Regarding the interleaving techniques, there are four major types of interleaving schemes: block interleavers, convolutional interleavers, random interleavers and code matched interleavers. Different types of interleaver with different sizes and structures will have considerable effects on the performance of both turbo codes and serial concatenated codes. At low SNR's, the interleaver size is the principle factor while both interleaver size and structure affect the code performance greatly at high SNR's. Code matched interleavers can improve the performance significantly by breaking the low weight input patterns effectively yet with highest complexity. Random interleavers are commonly used in practical systems. A non-uniform interleaver has been used in the original turbo codes [13]. S-random interleavers proposed by Divsalar and Pollara are pseudo-random interleavers [22]. S is defined as the minimum interleaving distance. The principles of S-random interleavers are as follows. For a S-random interleaver of size N, the distance between two integers less than S_1 before interleaving will be spread to the distance greater than or equal to S_2, where S_1 and S_2 are two integers smaller than N. In general, the parameters S_1 and S_2 should be chosen as large as possible in order to break the maximum input pattern lengths. Usually it is appropriate to set $S_1 = S_2 = S$. For $S = 1$, the S-random interleaver becomes a random interleaver.

2.2.3 Design considerations

In [18], Benedetto et al. have proposed an analytical bound on the performance of the SCCCs and used it to derive design criteria for the constituent codes (the conclusions can be extended to SCBCs when appropriate). On the basis of the mathematical analysis, the design guidelines for the SCCCs are as follows:

1. The inner encoder must be a convolutional recursive encoder.
2. The outer code could be either recursive or non-recursive code.
3. The free distance of the outer code d^o_{free} is preferred to be odd and maximized.

2.3 Iterative Decoding of SCCC

Trellis-based iterative decoding of SCCC consists of an outer decoder and an inner decoder separated by a (de)interleaver. The decoding algorithm of SCCC is based on the iterative use of *a posteriori probability* (APP) algorithms [23] applied to each constituent code. *Soft-input-soft-output* (SISO) module is very appropriate for the implementation of the APP algorithm in its basic form for the iterative decoding of a concatenated coding scheme. MAP or log-MAP algorithms are used to estimate the most likely information bit which have been transmitted in a coded sequence.

2.3.1 SISO Module for Iterative Decoding

2.3.1.1 Log-likelihood Algebra

The log-likelihood ratio $L(u_k)$ of a binary random variable U taking value u_k is defined as:

$$L(u_k) = \ln \frac{P(u_k = 1)}{P(u_k = 0)}$$

where u_k is in the field GF(2) with the elements $\{1, 0\}$.

Since

$$P(u_k = 1) = 1 - P(u_k = 0)$$

hence

$$e^{L(u_k)} = \frac{P(u_k = 1)}{1 - P(u_k = 1)}$$

so we can get

$$\begin{aligned} P(u_k = 1) &= \frac{e^{L(u_k)}}{1 + e^{L(u_k)}} \\ &= \left(\frac{e^{-L(u_k)/2}}{1 + e^{-L(u_k)}} \right) \cdot e^{L(u_k)/2} \end{aligned}$$

and

$$\begin{aligned} P(u_k = 0) &= \frac{1}{1 + e^{L(u_k)}} \\ &= \left(\frac{e^{-L(u_k)/2}}{1 + e^{-L(u_k)}} \right) \cdot e^{-L(u_k)/2} \end{aligned}$$

From the two equations above, there is one common item and one different item, therefore the probability of random variable U taking value 0 or 1 can be represented by a general expression as follows:

$$\begin{aligned} P(u_k) &= \left(\frac{e^{-L(u_k)/2}}{1 + e^{-L(u_k)}} \right) \cdot e^{(u_k - \frac{1}{2}) \cdot L(u_k)} \\ &= A_k \cdot e^{(u_k - \frac{1}{2}) \cdot L(u_k)} \qquad (2.1) \end{aligned}$$

where $A_k = \left(\frac{e^{-L(u_k)/2}}{1+e^{-L(u_k)}} \right)$ is a common factor.

2.3.1.2 SISO module for iterative decoding algorithm

The core of the decoding algorithm for SCCC is a four-port device with two inputs and two outputs. It accepts the probability (or corresponding likelihood ratio) of the information and code symbols labeling the edges of the code trellis as its inputs. and forms an update of these probabilities as its outputs. Figure 2.4 shows an edge of the trellis section, where symbol e denotes the trellis edges, i.e, a transition from time $k-1$ to k, and the information and code symbols associated to the edge e are represented by $u(e)$ and $c(e)$, and the starting and ending states of the edge e are $s^S(e)$ and $s^E(e)$, respectively. Assuming a time-invariant convolutional encoder.

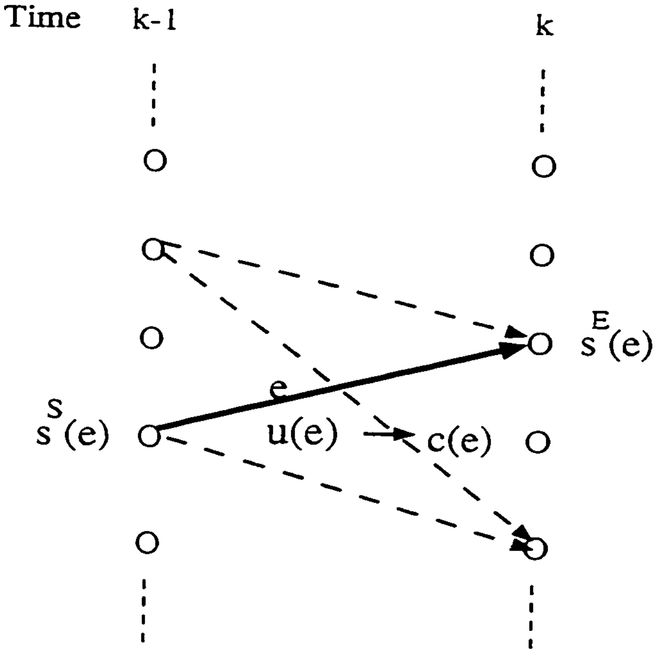


Figure 2.4: Trellis section of a transition caused by an input symbol

Two SISO modules are used in the iterative decoding process as shown in Figure 2.5. Figure 2.1 illustrates the encoder and iterative decoder for turbo codes (PCCC) as a comparison. Let us explain the decoding process in detail.

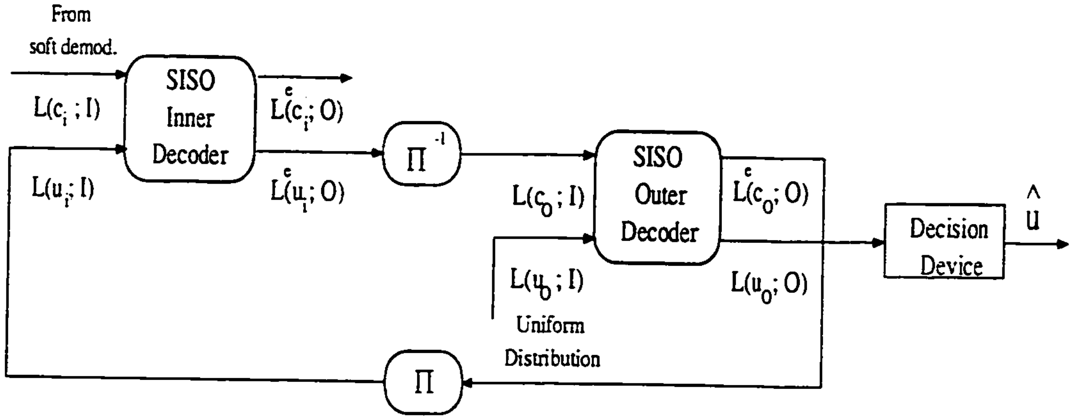


Figure 2.5: Block diagram of iterative decoder based on SISO module for SCCC

The symbols $L(\cdot;\ I)$ and $L(\cdot:\ O)$ at the input and output ports of SISO refer to the logarithmic likelihood ratios (LLR's). The second argument I represents soft values from channel, or a priori values from the other decoder while O represents a posteriori of the information symbols or code symbols. The subscript o refers to the outer encoder and i to the inner encoder. The argument u and c refer to the information symbols and code symbols respectively.

In contrast to the iterative decoding algorithm employed for turbo decoding, first the SISO outer decoder of SCCC iterative decoding algorithm receives a priori probabilities of code symbols, instead of channel observations and, secondly, it must compute and update *a posteriori probabilities* of both information symbols and code symbols.

During the first iteration of the SCCC algorithm, the SISO inner decoder is fed with the demodulator soft outputs, consisting of the symbol values received from the channels, i.e., of the code symbols of the inner encoder. The second input $L(u_i;\ I)$ of the SISO inner decoder is set to zero during the first iteration, since no *a priori* information is available (no output from SISO outer decoder for the first iteration).

The LLR's $L(c_i;\ I)$ are processed by SISO inner decoder block which computes the *extrinsic* $L^e(u_i;\ O)$ of the information symbols of the inner encoder. Then the *extrinsic* LLR's $L^e(u_i;\ O)$ are passed through the inverse interleaver $\pi^{-1}$and the outputs correspond to the LLR's of the code symbols of the outer encoder, i.e., $\pi^{-1}[L^e(u_i;\ O)] = L(c_o;\ I)$.

These $L(c_o;\ I)$ are sent to the block SISO outer decoder through its upper input. The SISO outer decoder starts to process the input $L(c_o;\ I)$ and compute *a posteriori probability* LLR's of both code symbols and information symbols. The input $L(u_o;\ I)$ at the lower input of the SISO outer decoder is always set to zero. This implies assuming equally likely transmitted source information symbols. The *extrinsic* LLR's $L^e(c_o;\ O)$ of the outer code symbols, after interleaving, π, are fed back to the lower input (corresponding to the information symbols of the inner encoder) of the block SISO inner decoder to start the second iteration, i.e., $\pi[L^e(c_o;\ O)] = L(u_i;\ I)$.

In the final iteration, the output LLR's $L(u_o;\ O)$ of the information symbols (which yield the *a posteriori* LLR's of the SCCC information symbols) will be used to recover the information bits [18][20][24][25]. The decision device makes a hard decision based on the $L(u_o;\ O)$.

2.3.2 Optimum and Sub-optimum Decoding Algorithms

The *Maximum Likelihood* (ML) algorithm (e.g., Viterbi Algorithm) is an optimal decoding method which minimizes the probability of *information sequence error* for convolutional codes, while *maximum a posterior* (MAP) algorithm is optimal to find the most probable *information bit* to have been transmitted given the received sequence.

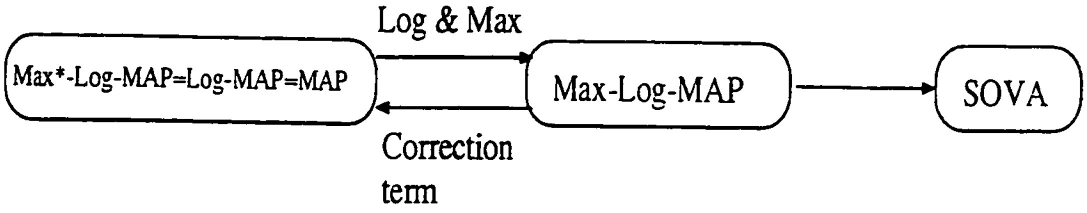


Figure 2.6: Relationship between MAP, log-MAP, Max-log-MAP and SOVA

For trellis-based soft-output decoding algorithms delivering additional reliability information together with hard decisions, the BCJR algorithm [23], known as *symbol-by-symbol* MAP algorithm is an optimal algorithm for estimating the states or outputs of a Markov process. It generates the APP for each decoded bit. However, MAP algorithm is difficult in practice due to its complex numerical representation of probabilities. Therefore by using the logarithm of the actual probabilities, log-MAP algorithm is equivalent to the true MAP in terms of performance but without its problems of implementation. MAP like algorithms, *Soft-Output Viterbi Algorithm* (SOVA) and the Max-log-MAP algorithm, are both suboptimal at low signal-to-noise ratios. The relationship between these algorithms is illustrated in Figure 2.6 [26]. It is worth noting that adding the correction factor to Max-Log-MAP results in an algorithm, Max*-Log-MAP whose performance is close to the Log-MAP algorithm.

2.3.2.1 SISO MAP algorithm

A section of a trellis diagram of a convolutional encoder is shown in Figure 2.4. Figure 2.7 shows the general SISO model.

Assume we have the following quantities:

(1) A finite time index set K, i.e., $K = \{1, 2, ..., n\}$,

(2) $\underline{U} = (U_k)_{k \in K}$, the sequences of input symbols over the time index set K and drawn from the alphabet $\underline{u} = \{u_1, u_2, ..., u_{N_I}\}$. The sequence of *a priori probability*

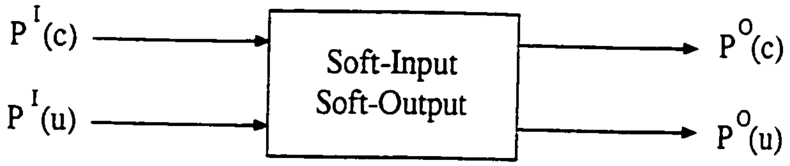


Figure 2.7: General soft-input soft-output (SISO) model

distribution defined as:

$$P^I(u) \underline{\triangle}\, (P[U_k = u_k])_{k \in K}$$

(3) $\underline{C} = (C_k)_{k \in K}$, the sequences of output (coded) symbols over the same time index set K, and drawn from the alphabet $\underline{c} = \{c_1 . c_2, \ldots, c_{N_o}\}$. For the sequence of coded symbols, the associated *a priori probability* distribution defined as:

$$P^I(c) \underline{\triangle}\, (P[C_k = c_k])_{k \in K}$$

(4) A set of N states $\underline{s} = \{s_1, s_2, \ldots, s_N\}$. The state of the trellis at time k is $S_k = s$.

(5) All possible transitions between the trellis states can be obtained by the product:

$$\underline{e} = \underline{s} \times \underline{u} = \{e_1, \ldots, e_{N \times N_I}\}$$

The following quantities are associated with each edge e

- The starting state $s^S(e)$

- The ending state $s^E(e)$

- The input symbol $u(e)$

- The output symbol $c(e)$

The relationship between these functions depends on the particular encoder. Throughout the thesis, we assume that the pair $(s^S(e),\ u(e))$ uniquely identifies the ending state $s^E(e)$ which means that given the initial trellis state, there is a one-to-one correspondence between input sequences and state sequences. (we will drop e for simplicity) From Figure 2.7, the probability distributions of the output sequences $P^O(c)$ and $P^O(u)$ are based on its inputs and on its knowledge of the trellis branch transition section. At time k, the output probability distributions are computed as:

$$\tilde{P}_k^O(c) = \tilde{B}_c \sum_{e: c(e)=c} \alpha_{k-1}[s^S(e)] P_k^I[u(e)] P_k^I[c(e)] \beta_k[s^E(e)] \tag{2.2}$$

$$\tilde{P}_k^O(u) = \tilde{B}_u \sum_{e: u(e)=u} \alpha_{k-1}[s^S(e)] P_k^I[u(e)] P_k^I[c(e)] \beta_k[s^E(e)] \tag{2.3}$$

where $\tilde{B}_c$ and $\tilde{B}_u$ are normalization constants.

The forward recursion $\alpha_k[\cdot]$ and backward recursion $\beta_k[\cdot]$ are given by

$$\alpha_k(s) = \sum_{e: s^E(e)=s} \alpha_{k-1}[s^S(e)] \cdot P_k^I[u(e)] \cdot P_k^I[c(e)], \quad k = 1, 2, ...n \tag{2.4}$$

and

$$\beta_k(s) = \sum_{e: s^S(e)=s} \beta_{k+1}[s^E(e)) \cdot P_{k+1}^I[u(e)] \cdot P_{k+1}^I[c(e)], \quad k = n-1, ..., 0 \tag{2.5}$$

with initial values:

$$\alpha_0(s) = \begin{cases} 1 & s = S_0 \\ 0 & otherwise \end{cases}$$

$$\beta_n(s) = \begin{cases} 1 & s = S_n \\ 0 & otherwise \end{cases}$$

The quantities $\tilde{B}_c$, $\tilde{B}_u$ are normalization constants calculated as follows:

$$\tilde{B}_c \rightarrow \sum_c \tilde{P}_k^O(c) = 1$$

$$\tilde{B}_u \rightarrow \sum_u \tilde{P}_k^O(u) = 1$$

It is apparent that the $P_k^I[c(e)]$ in Equation 2.2 and $P_k^I[u(e)]$ in Equation 2.3 do not depend on e, by definition of the summation indices and can be extracted from the summations. Therefore, we define the new quantities

$$P_k^O(c) \underline{\triangle}\, B_c \frac{\tilde{P}_k^O(c)}{P_k^I(c)} \tag{2.6}$$

$$P_k^O(u) \underline{\triangle}\, B_u \frac{\tilde{P}_k^O(u)}{P_k^I(u)} \tag{2.7}$$

where B_c and B_u are normalization constants as

$$B_c \rightarrow \sum_c P_k^O(c) = 1$$

$$B_u \rightarrow \sum_u P_k^O(u) = 1$$

From Equations 2.2, 2.3, 2.6 and 2.7, we can obtain

$$P_k^O(c) = B_c \tilde{B}_c \sum_{e:\, c(e)=c} \alpha_{k-1}[s^S(e)] P_k^I[u(e)] \beta_k[s^E(e)] \tag{2.8}$$

$$P_k^O(u) = B_u \tilde{B}_u \sum_{e:\, u(e)=u} \alpha_{k-1}[s^S(e)] P_k^I[c(e)] \beta_k[s^E(e)] \tag{2.9}$$

The probability distributions $P_k^O(u)$ and $P_k^O(c)$ are extrinsic information that present the added value of the SISO module to the a priori distributions $P_k^I(u)$ and $P_k^I(c)$.

2.3.2.2 Log-MAP algorithm

Although MAP algorithm is the optimal way of decoding for both turbo code and serial concatenated codes. its high computational complexity prevent it from being implemented in practical systems. This is due to the fact that MAP is a multiplicative algorithm. To overcome this drawback, the natural logarithm function with monotonicity is applied to convert multiplications into additions.

The following are some definitions:

$L_k^I(c) = \ln[P_k^I(c)]$ $\qquad$ $L_k^I(u) = \ln[P_k^I(u)]$

$\alpha_k^{LM}(s) = \ln[\alpha_k(s)]$ $\qquad$ $\beta_k^{LM}(s) = \ln[\beta_k(s)]$

Taking the natural logarithm of the output probability distributions of Equations 2.8 and 2.9, we can get

$$\begin{aligned} L_k^O(c) &= \ln[P_k^O(c)] \\ &= \ln\left(\sum_{e:\, c(e)=c} \alpha_{k-1}[s^S(e)]P_k^I[u(e)]\beta_k[s^E(e)]\right) + \ln(B_c\tilde{B}_c) \\ &= \ln\left[\sum_{e:\, c(e)=c} \exp\left(\ln\left\{\alpha_{k-1}[s^S(e)]P_k^I[u(e)]\beta_k[s^E(e)]\right\}\right)\right] + \ln(B_c\tilde{B}_c) \\ &= \ln\left[\sum_{e:\, c(e)=c} \exp\left(\alpha_{k-1}^{LM}[s^S(e)] + L_k^I[u(e)] + \beta_k^{LM}[s^E(e)]\right)\right] \\ &\quad + \ln(B_c\tilde{B}_c) \end{aligned} \tag{2.10}$$

$$\begin{aligned} L_k^O(u) &= \ln[P_k^O(u)] \\ &= \ln\left(\sum_{e:\, u(e)=u} \alpha_{k-1}[s^S(e)]P_k^I[c(e)]\beta_k[s^E(e)]\right) + \ln(B_u\tilde{B}_u) \\ &= \ln\left[\sum_{e:\, u(e)=u} \exp\left(\alpha_{k-1}^{LM}[s^S(e)] + L_k^I[c(e)] + \beta_k^{LM}[s^E(e)]\right)\right] \\ &\quad + \ln(B_u\tilde{B}_u) \end{aligned} \tag{2.11}$$

Similarly, the forward and backward recursions in logarithm domain can be obtained by taking the logarithm of Equations 2.4 and 2.5 respectively:

$$\alpha_k^{LM}(s) = \ln\left[\sum_{e:\, s^E(e)=s} \exp\left\{\alpha_{k-1}^{LM}[s^S(e)] + L_k^I[u(e)] + L_k^I[c(e)]\right\}\right], \quad k = 1, 2, ..., n$$

$$\beta_k^{LM}(s) = \ln\left[\sum_{e:\, s^S(e)=s} \exp\left\{\beta_{k+1}^{LM}[s^E(e)] + L_{k+1}^I[u(e)] + L_{k+1}^I[c(e)]\right\}\right], \quad k = n-1, ..., 1, 0$$

with initial values

$$\alpha_0^{LM}(s) = \begin{cases} 0 & s = S_0 \\ -\infty & otherwise \end{cases}$$

$$\beta_n^{LM}(s) = \begin{cases} 0 & s = S_n \\ -\infty & otherwise \end{cases}$$

2.3.2.3 Max log-MAP and correction term

When calculating $L_k^O(u)$, $L_k^O(c)$, $\alpha_k^{LM}(s)$ and $\beta_k^{LM}(s)$, the sum of exponentials in these equations will still result in the computing complexity and the possibility of excessive growth of the numerical values of α's and β's. A simple solution to this problem is using the max function to yield approximations of these quantities.

1. Max function

Define two variables a_1, a_2, we have

$$\begin{aligned} \ln(e^{a_1} + e^{a_2}) &= \ln e^{a_1} + \ln(1 + e^{(a_2 - a_1)}) \\ &= \ln e^{a_2} + \ln(1 + e^{(a_1 - a_2)}) \\ &= \max(a_1, a_2) + \ln(1 + e^{-|a_2 - a_1|}) \\ &= \max(a_1, a_2) + f_c(|a_2 - a_1|) \end{aligned} \tag{2.12}$$

where f_c is a correction term.

Assume $\max(a_1, a_2) \gg a_i$, $i = 1, 2$, then

$$\ln(e^{a_1} + e^{a_2}) \approx \max(a_1, a_2).$$

Similarly we can get max function for k variables:

$$\begin{aligned} \ln\left[\sum_{i=1}^{k} e^{a_i}\right] &= \ln\left[e^{a_1} + e^{a_2} + \cdots + e^{a_{k-1}} + e^{a_k}\right] \\ &= \max(a_i) + \ln\left[\sum_{i=1}^{k} e^{a_i - \max(a_i)}\right] \\ &\approx \max(a_i) \end{aligned} \qquad (2.13)$$

Therefore by using Max-log-MAP algorithm, equations of $L_k^O(u)$, $L_k^O(c)$, $\alpha_k^{LM}(s)$ and $\beta_k^{LM}(s)$ become:

$$L_k^O(c) \approx \max_{e:\, c(e)=c} \left\{\alpha_{k-1}^{LM}[s^S(e)] + L_k^I[u(e)] + \beta_k^{LM}[s^E(e)]\right\} + \ln(B_c\bar{B}_c)$$

$$L_k^O(u) \approx \max_{e:\, u(e)=u} \left\{\alpha_{k-1}^{LM}[s^S(e)] + L_k^I[c(e)] + \beta_k^{LM}[s^E(e)]\right\} + \ln(B_u\bar{B}_u)$$

$$\alpha_k^{LM}(s) \approx \max_{e:\, s^E(e)=s} \left\{\alpha_{k-1}^{LM}[s^S(e)] + L_k^I[u(e)] + L_k^I[c(e)]\right\}, \quad k = 1, 2, ..., n$$

$$\beta_k^{LM}(s) \approx \max_{e:\, s^S(e)=s} \left\{\beta_{k+1}^{LM}[s^E(e)] + L_{k+1}^I[u(e)] + L_{k+1}^I[c(e)]\right\}, \quad k = n-1, ..., 1, 0$$

2. Correction term

Max-log-MAP algorithm is good for medium to high signal-to-noise ratios but the performance degrades at low signal-to-noise ratios. According to Equation 2.12, a correction term can be added to the max function to increase the accuracy at the expense of sacrificing some of the lowered complexity of the Max-log-MAP algorithm. For k variables in Equation 2.13, the correction term could be $\ln\left[\sum_{i=1}^{k} e^{a_i - \max(a_i)}\right]$ or an approximation of $\ln\left[\sum_{i=1}^{k} e^{a_i - \max(a_i)}\right]$ that can be computed using a one-dimensional look-up table to simplify the calculation while keeping the desired performance [17]. We call max-log-MAP with correction term as max*-log-MAP.

2.4 Demodulation and Decoding of SCCC with M-ary Modulation

In most digital communication systems, bandwidth efficiency and power efficiency are important factors in choosing coding and modulation techniques. Coding improves the performance dramatically and requires lower power, however, it needs more bandwidth for the transmission at the same data rate compared with the uncoded case. On the other hand, higher level modulation schemes decrease bandwidth occupancy but increase the required received power [27]. For bandwidth-efficient transmission, we may use Turbo Trellis Coded Modulation (TCM) but at the cost of a very high decoding complexity caused by the hypertrellis of the combination of coding and modulation. To reduce the decoding complexity of SCCC with bandwidth-efficient M-ary modulation techniques such as M-ASK (amplitude-shift keying), M-PSK (phase-shift keying) or QAM (quadrature amplitude modulation), we apply a converting algorithm to demodulate the received symbols the first time in the system of SCCC with M-ary modulation. The soft-decision bit values obtained from demodulating the received symbol values will be used by the SCCC's inner decoder. We will study the algorithm in more detail in this section. The general system model of serial concatenated convolutional code with M-ary modulation is shown in Figure 2.8.

2.4.1 Converting algorithm

At time k, we assume the following parameters over AWGN channel:

C: a binary sequence output from inner encoder

S: transmitted symbol sequence generated by mapping coded bits C onto signal constellation

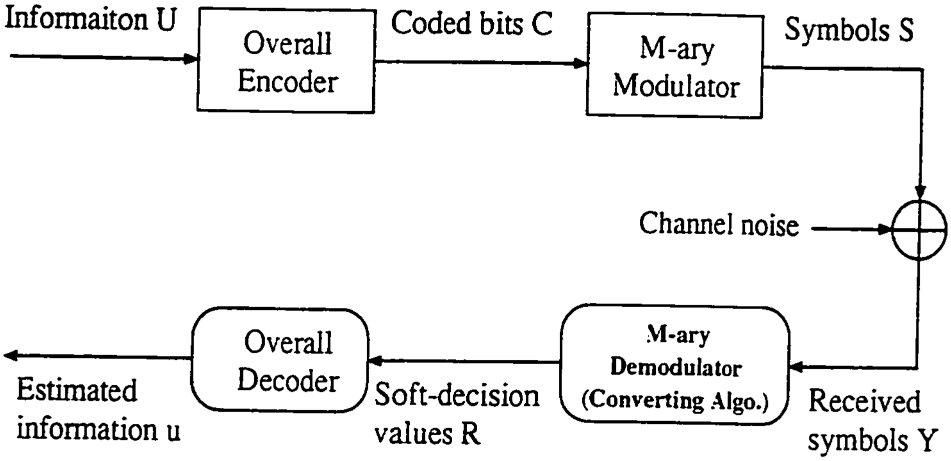


Figure 2.8: General system model of SCCC with M-ary modulation

Y: received symbols from channel

R: soft-decision values of coded bits received from m-ary demodulator

2^m symbol points on the constellation, m is the number of coded bits mapping to one symbol, e.g., $m = 2$ means that 4 possible signal points on the constellation space, $m = 3$ means 8 possibilities, etc.

$b_1, b_2, ...b_m$ represent the m coded bits taking values of 0 or 1.

The modulated signals (symbols) are transmitted through in-phase (I-part) and quadrature (Q-part) channel. The transmitted signals and received signals can be represented by the following, respectively:

$$s^k = s_I^k + js_Q^k$$

and

$$\begin{aligned} y^k &= y_I^k + jy_Q^k \\ &= (s_I^k + \eta_I^k) + j(s_Q^k + \eta_Q^k), \end{aligned}$$

where η_I^k and η_Q^k are Gaussian noise at time k in I-part and Q-part.

Now, we begin to describe the algorithm to find the log-likelihood ratio of every coded bit $b_1, b_2, ...b_m$. Take a constellation with 2^m signal points. The probability of m conditioned on the received signal y is (we drop k for simplicity):

$$\begin{aligned} P(b_i = 0 \mid y) &= \frac{p(y \mid b_i = 0) \cdot P(b_i = 0)}{p(y)} \\ &= \frac{1}{p(y)} \{ \sum_{b_1} \cdots \sum_{b_{i-1}} \sum_{b_{i+1}} \cdots \sum_{b_m} p(y \mid b_i = 0,\ b_1, ...b_{i-1}, b_{i+1}, ...b_m) \\ &\quad \cdot P(b_i = 0,\ b_1, ...b_{i-1}, b_{i+1}, ...b_m) \} \end{aligned} \quad (2.14)$$

Suppose that the signals are transmitted with equal probability. Therefore, in Equation 2.14, the second term in the numerator is a constant, i.e., $1/2^m$ and can be extracted from the summation. Then Equation 2.14 becomes:

$$P(b_i = 0 \mid y) = \frac{1}{2^m \cdot p(y)} \sum_{b_1} \cdots \sum_{b_{i-1}} \sum_{b_{i+1}} \cdots \sum_{b_m} p(y \mid b_i = 0,\ b_1, ...b_{i-1}, b_{i+1}, ...b_m)$$

Similarly we can get the probability of $b_i = 1$ conditioned on y :

$$P(b_i = 1 \mid y) = \frac{1}{2^m \cdot p(y)} \sum_{b_1} \cdots \sum_{b_{i-1}} \sum_{b_{i+1}} \cdots \sum_{b_m} p(y \mid b_i = 1,\ b_1, ...b_{i-1}, b_{i+1}, ...b_m)$$

The log-likelihood ratio of bit b_i is:

$$\begin{aligned} \lambda_i &= \ln \frac{P(b_i = 1 \mid y)}{P(b_i = 0 \mid y)} \\ &= \ln \frac{\sum_{b_1} \cdots \sum_{b_{i-1}} \sum_{b_{i+1}} \cdots \sum_{b_m} p(y \mid b_i = 1,\ b_1, ...b_{i-1}, b_{i+1}, ...b_m)}{\sum_{b_1} \cdots \sum_{b_{i-1}} \sum_{b_{i+1}} \cdots \sum_{b_m} p(y \mid b_i = 0,\ b_1, ...b_{i-1}, b_{i+1}, ...b_m)} \end{aligned} \quad (2.15)$$

The computation of the summation will be very complicated for both the numerator and the denominator in Equation 2.15. We find that there exists a dominant term among the summation in both the numerator and the denominator in Equation 2.15. Using the max function described in 2.3.2.3, we may keep the dominant term and omit the rest of the terms. Though we sacrifice the performance a little bit, it is worth to do the approximation so that the complexity can be reduced significantly.

Now let us do the approximation. The dominant term will be the probability of the received signal conditioned on the nearest signal point with $b_i = 1$ *or* 0.

When $b_i = 1$ $(i = 1, 2, ...m)$, let $A_1(i) = A_{1,I}(i) + jA_{1,Q}(i)$ be the nearest signal point, i.e., $(b_1, b_2, ..., b_{i-1}, 1, b_{i+1}, ..., b_m)$, to the received signal y. When $b_i = 0$ $(i = 1, 2, ...m)$, let $A_0(i) = A_{0,I}(i) + jA_{0,Q}(i)$ be the nearest signal point i.e., $(b_1, b_2, ..., b_{i-1}, 0, b_{i+1}, ..., b_m)$, to the received signal y. Simplify the Equation 2.15, we get:

$$\begin{aligned} \lambda_i &\approx \ln \frac{\frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{\left(y_I(i)-A_{1,I}(i)\right)^2}{2\sigma^2}\right] \cdot \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{\left(y_Q(i)-A_{1.Q}(i)\right)^2}{2\sigma^2}\right]}{\frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{\left(y_I(i)-A_{0,I}(i)\right)^2}{2\sigma^2}\right] \cdot \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{\left(y_Q(i)-A_{0,Q}(i)\right)^2}{2\sigma^2}\right]} \\ &= \frac{1}{2\sigma^2}\{A_{0,I}^2(i) + A_{0,Q}^2(i) - [A_{1,I}^2(i) + A_{1,Q}^2(i)] \\ &\quad +2 \cdot y_I(i) \cdot [A_{1,I}(i) - A_{0,I}(i)] + 2 \cdot y_Q(i) \cdot [A_{1,Q}(i) - A_{0,Q}(i)]\} \quad (2.16) \end{aligned}$$

Equation 2.16 is the general expression of the converting algorithm. From Equation 2.16, the computation of LLR converts to simply calculating the distance between the two signal points. $A_{0,I}^2(i) + A_{0,Q}^2(i)$ and $A_{1,I}^2(i) + A_{1,Q}^2(i)$ are the energy of the signal points $A_0(i)$ and $A_1(i)$ respectively. When using modulation techniques with equal energy of transmitted signals, such as 8PSK, the Equation 2.16 can be further simplified to

$$\lambda_i = \frac{1}{\sigma^2}\left\{y_I(i) \cdot [A_{1,I}(i) - A_{0,I}(i)] + y_Q(i) \cdot [A_{1,Q}(i) - A_{0.Q}(i)]\right\}. \quad (2.17)$$

After all λ_i, $i = 1, 2, ...m$ which are the log-likelihood ratio soft-decision values of the coded bits are obtained, they will be passed as the soft input to the inner decoder.

We now study an example in a bit more detail to illustrate the algorithm. The constellation of 8PSK is shown in Figure 2.9 where three coded bits are mapped to one signal point. First we calculate the distances between the received signal y and every signal point, i.e., the 8 signal points on the constellation, for $b_i = a$, $a = 1\,or\,0$ and $i = 1, 2, 3$, respectively. Secondly, we choose the signal points on the constellation for $b_i = 1$ or 0, respectively that have the minimal distance to the received signal y. For example, if the received signal y is within the decision area of the signal point V_0 with x-y values of $(\sqrt{2}A,\ 0)$, the nearest signal points for $b_1 = 0$ and $b_1 = 1$ are $(\sqrt{2}A,\ 0)$ and $(A, -A)$, respectively. Similarly, we can get the nearest signal points for $b_2 = 0$, $b_2 = 1$ and $b_3 = 0$, $b_3 = 1$ are $(\sqrt{2}A,\ 0)$, $(0,\ \sqrt{2}A)$, and $(\sqrt{2}A,\ 0)$, $(A,\ A)$, respectively. For 8PSK modulation with equal energy of the transmitted signals, we may use the Equation 2.17 to get the log-likelihood ratio soft values for the three coded bits. Therefore, the log-likelihood ratio of the coded bit b_1 is:

$$\lambda_1 = \frac{1}{\sigma^2}\left[y_I \cdot \left(A - \sqrt{2}A\right) + y_Q \cdot (-A)\right].$$

Similarly, we obtain log-likelihood ratio of the coded bits b_2:

$$\lambda_2 = \frac{1}{\sigma^2}\left[y_I \cdot \left(-\sqrt{2}A\right) + y_Q \cdot \left(\sqrt{2}A\right)\right],$$

and b_3:

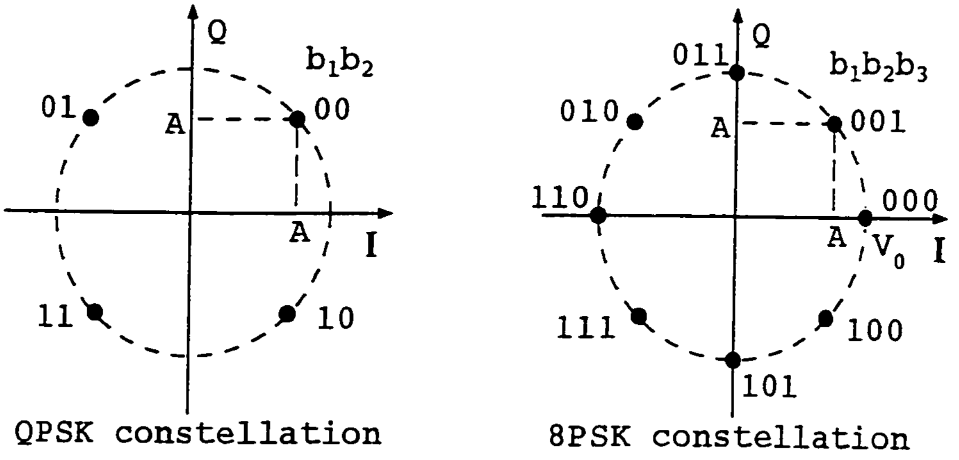
$$\lambda_3 = \frac{1}{\sigma^2}\left[y_I \cdot \left(A - \sqrt{2}A\right) + y_Q \cdot A\right],$$

respectively. In this way, we obtain the soft values of the coded bits from the received symbols with acceptable demodulation complexity. These values will be passed to the SCCC inner decoder.

2.4.2 Simulation results

Now, some examples of SCCC with QPSK and 8PSK modulation techniques are given. The outer encoder and inner encoder are shown in Figure 2.3. The constellation of QPSK and 8PSK are illustrated in Figure 2.9.

In Figure 2.10, the two solid lines and the two dash-dotted lines represent the BER performance of SCCC using QPSK and 8PSK modulation techniques with two different block sizes respectively. The overall code rate of the SCCC encoder is 1/4,



QPSK constellation

8PSK constellation

Figure 2.9: QPSK and 8PSK constellation diagram with Gray mapping

for QPSK modulation, the bandwidth efficiency is $1/2\,bps/Hz$ while for 8PSK is $3/4\,bps/Hz$. The number of iteration is 8 for all simulations. A random interleaver links the outer and the inner encoder. From the simulation results, we can conclude that (1) the performance of 8PSK modulation is about $1.0\,dB$ worse than QPSK at BER of 10^{-5} under the same block size but with higher bandwidth efficiency; (2) the performance improves with the size of the block, e.g., the BER of SCCC with 8PSK modulation of $blocksize = 2047$ reaches 10^{-6} at $2.5\,dB$ and is about $0.3\,dB$ better than $blocksize = 1021$.

2.5 Summary

In this chapter, the fundamental principles of designing serial concatenated convolutional codes have been introduced, including encoder structure, functions of the interleaver, soft-input soft-output optimal and sub-optimal decoding algorithm and design criteria. The inner encoder must be a convolutional recursive encoder which yields the interleaver gain. The iterative log-MAP decoding algorithm is based on SISO module which generates maximum a posteriori log-likelihood ratios of the

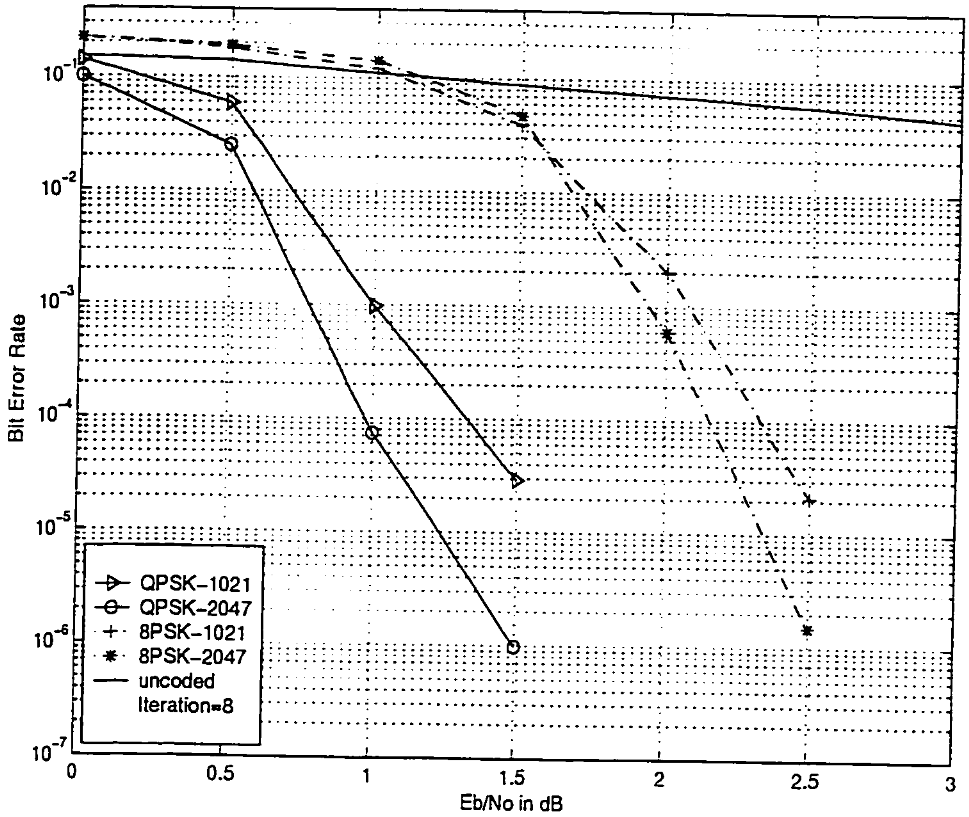


Figure 2.10: BER Performance of SCCC with QPSK and 8PSK modulation techniques using the converting algorithm

information symbols.

For bandwidth-efficient transmission, multi-ary modulation techniques are applied with SCCC. To reduce the complexity of the decoding algorithm, a converting algorithm is introduced to yield soft-decision values of the coded bits utilized by SCCC inner decoder. The simulation result shows that the performance is acceptable with low decoding complexity.

Chapter 3

SCCC with Coded MSK over AWGN Channel

In reference [18], Benedetto et al. showed that serial concatenated convolutional codes have good performance, especially at high signal to noise ratios. The analytical results showed that serial concatenated codes can yield significantly higher interleaver gains and steeper asymptotic slopes for the error probability curves. In addition, for SCCC, the error floor presents at very low bit error rate. The choice of coding and modulation techniques is important in communication systems for improving performance and saving bandwidth. TCM is one method to be utilized in SCCC to obtain bandwidth efficiency. In TCM systems, the inner code is a recursive trellis code designed for the chosen signal set [28][29]. On the other hand, a more advanced modulation technique is Continuous Phase Modulation (CPM). The fact that CPM is a modulation technique makes it a good candidate for the inner code of a SCCC. We will investigate the performance of a SCCC with CPM as an inner code over AWGN channel and ISI channel in the next two chapters. The basic principles of the CPM, the decoding algorithm for the convolutional outer decoder and MSK inner decoder/demodulator and the simulation of different factors which impact SCCC with MSK over AWGN channel will be discussed in this chapter.

3.1 Continuous Phase Modulation (CPM)

In the transmission of digital information over a communications channel, the modulator is the interface device that maps the digital information into analog waveforms that match the characteristics of the channel. There are two main categories of modulation techniques, memoryless modulation such as PAM, PSK, QAM, etc., and modulation with memory [30]. We focus on the nonlinear modulation method with memory called continuous phase modulation (CPM) in which the phase of the signal is constrained to be continuous .

3.1.1 Brief introduction

The conventional memoryless modulation techniques, such as FSK are generated by shifting the carrier by a number of frequencies $f_n = \frac{1}{2}\Delta f \cdot u_n$, $u_n = \pm 1, \pm 3, ..., \pm(M-1)$, to reflect the digital information being transmitted, where $\Delta f = 1/2T$ represents the minimum frequency separation between adjacent signals for orthogonality of the M signals, where T is the symbol interval. The switching from one frequency to another may be accomplished by having $M = 2^m$ separate oscillators tuned to the desired frequencies and selecting one frequency according to the m-bit symbol transmitted in the signal interval. However, such abrupt switching from one oscillator output to another in successive signalling intervals may result in dramatic phase-changing and relatively large spectral side lobes outside of the main spectral band, consequently, requiring a large frequency band for transmission of the signal.

M-PSK modulation techniques also generate phase shifts and make undesired sidebands [2]. The solution to this problem is to make the phase of the modulated signal constrained to be continuous. To this end, Continuous Phase Modulation (CPM) was proposed and research interest grew rapidly in the late 1970's and early 1980's. References [31], [32] and [33] present a comprehensive summary of much of this research work. The constant envelope of CPM scheme and their excellent spectral properties make them attractive in many digital transmission systems, such as satellite and mobile communication systems. The constant envelope property is useful for non-linear power amplifiers which require a low peak-to-average power ratio and is desirable for digital data transmission over nonlinear and/or fading channels. In real communication systems, CPM has a narrower spectral occupancy in terms of fractional out-of-band power. The bandwidth efficiency accounts for the popularity of CPM in many digital communications systems.

3.1.2 Decomposition of CPM

CPM systems are memory modulation schemes thus exhibit a coding gain when compared to memoryless modulation techniques. The manner of introducing memory to a CPM system resembles a convolutional encoding system in many ways. In both cases, trellis can be used to display the possible output signals and describe how the states change with time. As described in [34], CPM can be decomposed into a *continuous-phase encoder (CPE)* and a *memoryless modulator (MM)*. Such a decomposition would have two obvious advantages. First, it allows the encoding operation to be studied independently of the modulation. If the CPE is time invariant and linear, then it can be studied using the same techniques that have been developed for convolutional codes. Second, the isolation of the MM allows us to model the system as a cascade of the MM, the waveform channel and the demodulator that operates over a discrete memoryless channel [34].

Consider a CPM system with M-ary information symbols $U \in \{0, 1, ..., M-1\}$ transmitted every symbol interval T, where $M = 2^m$. The symbols are phase-modulated using a positive normalized frequency pulse $g(t)$ containing no impulses and being non-zero for L symbol intervals. L is a positive integer called the *memory* of the CPM scheme. If $g(t) = 0$ for $t > T$, $L = 1$ and the CPM signal is called *full response CPM.* If $g(t) \neq 0$ for $t > T$, $L > 1$ and it is called *partial response CPM.* The phase pulse $q(t)$ is the integral of the frequency pulse [34][35]. For example, if $g(t)$ is a rectangular pulse with amplitude $1/2LT$

$$g(t) = \begin{cases} 1/(2LT) & 0 \leq t \leq LT \\ 0 & otherwise \end{cases}$$

then $q(t)$ can be represented as

$$\begin{aligned} q(t) &= \int_{-\infty}^{t} g(\tau)\, d\tau \\ &= \begin{cases} 0 & t < 0 \\ t/(2LT) & 0 \leq t \leq LT \\ 1/2 & t > LT \end{cases} \end{aligned}$$

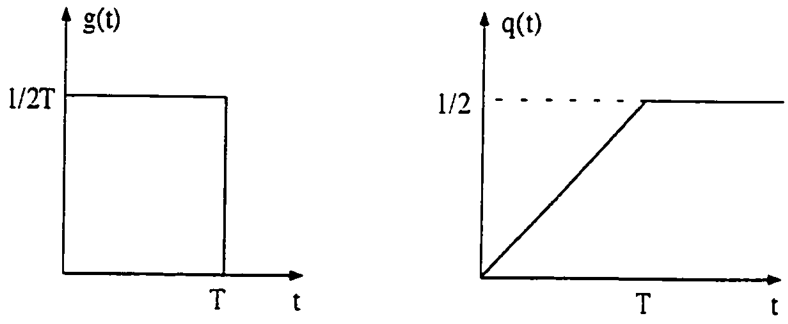


Figure 3.1: Pulse shape for full response CPM ($L = 1$)

Figure 3.1 shows the pulse shape of the full response CPM. The modulation index is assumed to be rational and irreducible, $h = Q/P$. An infinite variety of CPM signals can be generated by choosing different pulse shapes $g(t)$ and by varying the modulation index h and the alphabet size M. The Minimum-shift keying (MSK) which is a binary continuous phase FSK with $h = \frac{1}{2}$ is widely used in many digital communications systems and is our main research concern.

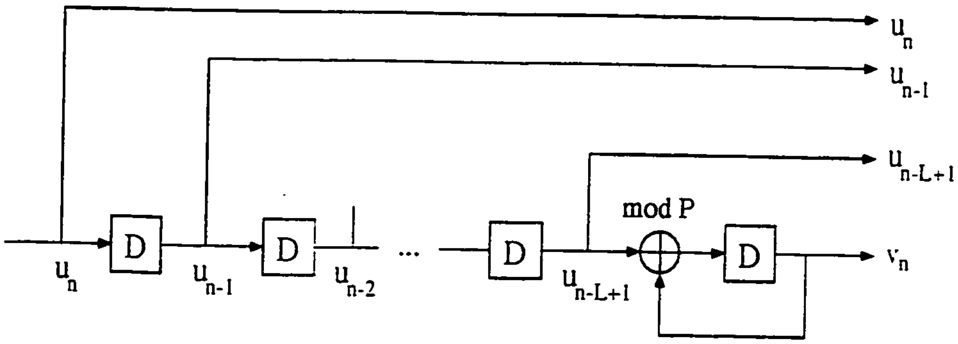


Figure 3.2: General diagram of continuous phase encoder (CPE)

It is well known that the CPM system can be described by a trellis [31]. The general diagram of continuous phase encoder (CPE) is shown in Figure 3.2. The transmitted signal during symbol interval n is specified by the current symbol u_n, the $L - 1$ previous data symbols $u_{n-L+1}, ..., u_{n-1}$, and the accumulated value of $\{u_i\}$ is denoted by v_n, i.e., $v_n = \sum_{i=0}^{n-L} u_i \ mod\ P$ which can take only P values. The output vector from the CPE $x_n = [u_n, ..., u_{n-L+1}, v_n]$ is fed to the memoryless modulator, which generates one of the $P \cdot M^L$ signals.

3.2 Decoding of SCCC with MSK Modulation

Minimum Shift Keying (MSK) is a special form of binary continuous phase FSK with modulation index $h = \frac{1}{2}$ $(P = 2)$. The two possible frequencies in the interval $nT \leq$

$t \leq (n+1)T$ are $f_1 = f_c - \frac{1}{4T}$ and $f_2 = f_c + \frac{1}{4T}$, where f_c is the central carrier frequency [2]. The frequency separation $\Delta f = f_2 - f_1 = 1/2T$ is the minimum frequency spacing to allow coherent demodulation. Compared with QPSK and OQPSK (offset QPSK), MSK has smoother pulse and lower sidelobes. It is a spectrally efficient modulation scheme which is particularly attractive for use in bandwidth and power efficient communication systems, including mobile satellite systems [36] [37] and high-rate microwave systems [38][39]. A baseband filtered version of MSK, called Gaussian MSK is used in the GSM and DECT systems [40]. From Figure 3.2, we can see that the CPE behaves as a recursive systematic convolutional (RSC) code and thus can be incorporated as the inner code in an SCCC system.

3.2.1 System model

Figure 3.3 illustrates the system model and decoding algorithm of serial concatenated convolutional code with MSK modulation as the inner code. The outer encoder structure is the same as the outer encoder in Figure 2.3 which is a systematic recursive code with code rate $R_o = \frac{1}{2}$. The generator matrix for the outer encoder is $G_{outer}(D) = [1, \frac{1+D^2}{1+D+D^2}]$. MSK is coded modulation with full response taken as both inner encoder and modulator.

As mentioned in Section 3.1.2, CPM can be decomposed into CPE and MM. Thus MSK can also be decomposed into a continuous phase encoder followed by a one-to-one memoryless mapper. The main function of the CPE is to ensure carrier phase continuity. MM will map the coded bits encoded by the CPE to the signal space. According to Figure 3.2, for full response of MSK, i.e., $L = 1$, the encoder will be binary recursive systematic one (i.e., $M = 2$) with one memory. Figure 3.4 depicts the decomposition of MSK, the structure of CPE and memoryless mapping. The

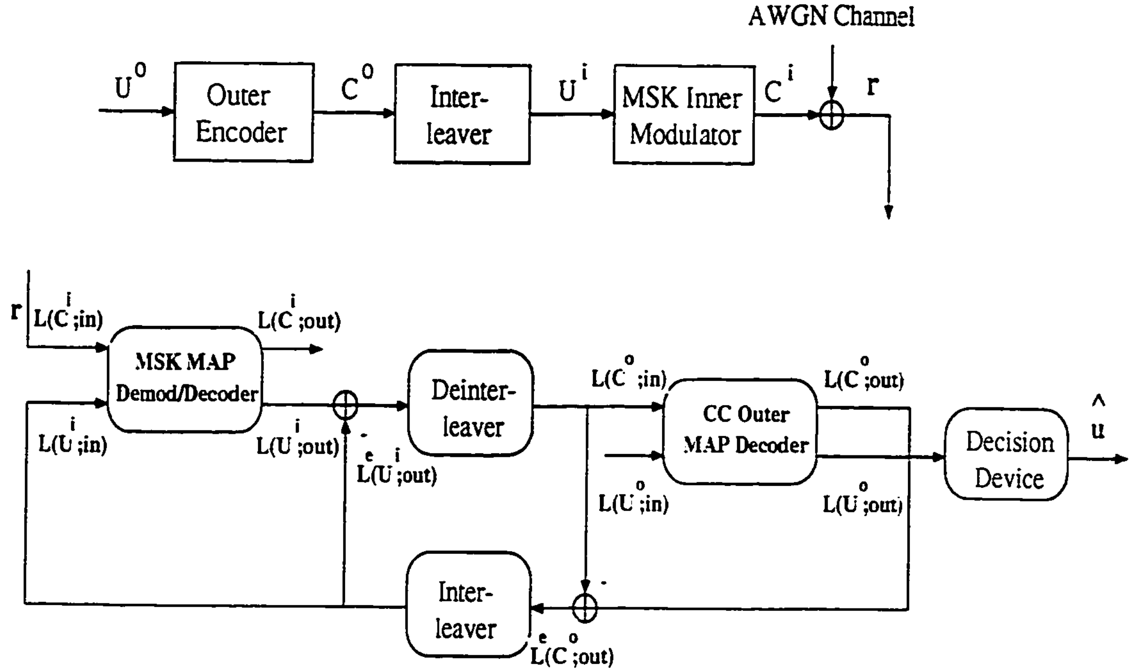


Figure 3.3: System model of SCCC with MSK

CPE generates a RSC code with code rate $R_{cpe} = 1/2$ and the generator matrix $G_{inner}(D) = [1, \frac{D}{1+D}]$, where $b_1(k) = U_k^I$, $b_2(k) = [\sum_{i=0}^{k-1} U_i^I]_{mod 2}$ and U_k^I is the permuted coded bits output from the outer encoder. The two coded bits will map to one of the four transmitted waveforms.

$b_1 b_2$	$x(t)$		signal point
1 1	$-\cos[2\pi(f_c + \frac{1}{4T})(t - kT)]$	$-s^+(t)$	$(0, -A)$
0 1	$-\cos[2\pi(f_c - \frac{1}{4T})(t - kT)]$	$-s^-(t)$	$(-A, 0)$
1 0	$\cos[2\pi(f_c + \frac{1}{4T})(t - kT)]$	$s^+(t)$	$(0, A)$
0 0	$\cos[2\pi(f_c - \frac{1}{4T})(t - kT)]$	$s^-(t)$	$(A, 0)$

Table 3.1: Mapping relationship between the coded bits of CPE and signal points

It has been shown in [36] that the phase of transmitted waveforms in each symbol time interval can be represented by a single time invariant memoryless one-to-one

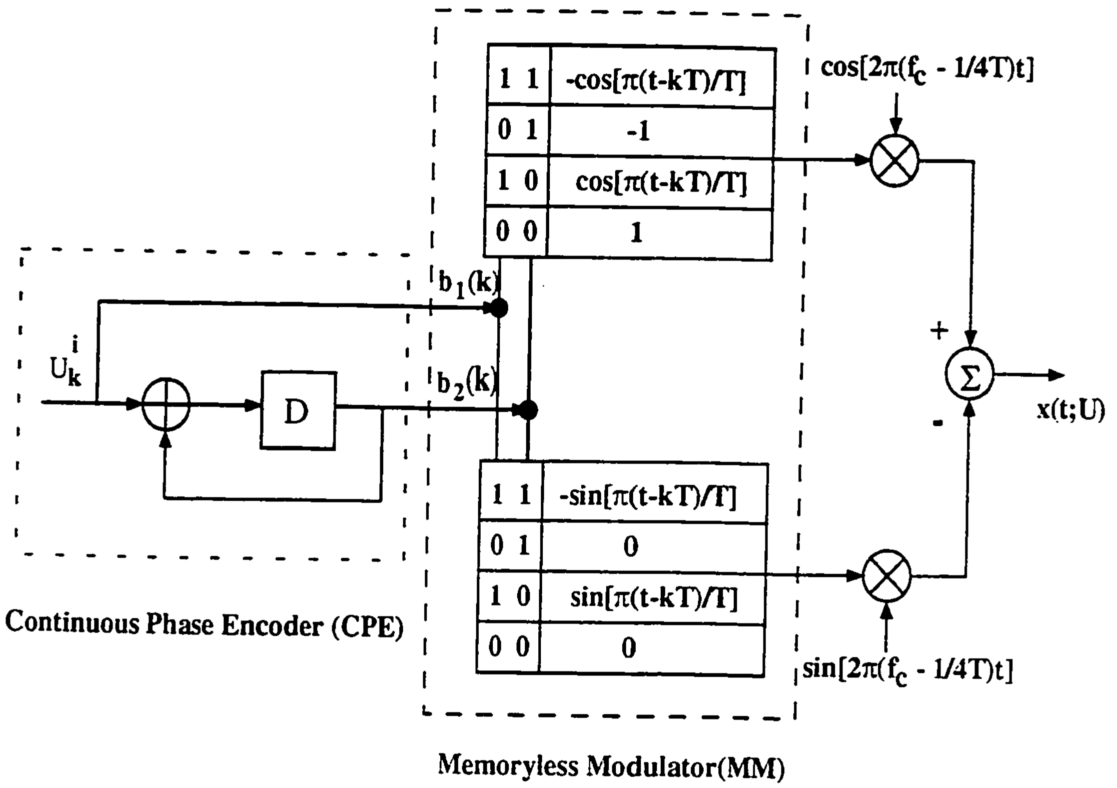
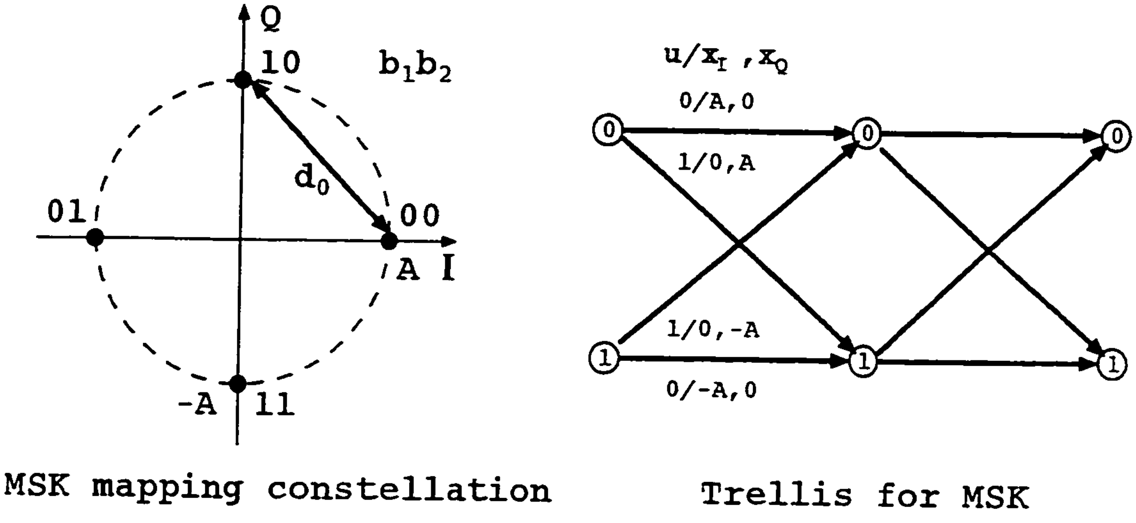


Figure 3.4: Block diagram of decomposition structure of MSK

mapping, so that the time varying phase output $x(t;U)$ in Figure 3.4 can be transformed into $x(t)$ as shown in Table 3.1. From Table 3.1, it is seen that in each symbol interval, the information symbol U_k, equivalent to $b_1(k)$ determines the frequency, for example, $b_1 = 1$ means the signal waveform of higher frequency $f_c + \frac{1}{4T}$ is selected, and $b_2(k)$ determines the phase of the transmitted waveform, e.g., $b_2 = 1$ or 0 causes the starting phase of the signal waveform to be $\pi\, rad$ or $0\, rad$ respectively [36]. The signal constellation and trellis structure for MSK are given in Figure 3.5. The minimum squared Euclidean distance is given by $d_0^2 = 2A^2$. There are two states 0 and 1 in the trellis. The trellis branch is labelled by *input information symbol / transmitted waveform (signal point)* where we map the input symbols to the signal points represented by in-phase (I) and quadrature (Q) components on the constellation directly.



MSK mapping constellation

Trellis for MSK

Figure 3.5: MSK constellation and trellis structure

3.2.2 Decoding algorithm

The decoding algorithm of SCCC with MSK is a log-MAP algorithm with iterative decoding. The decoding principles are the same as what we have explained in Section 2.3 which is the general description of the algorithm. We will derive the decoding algorithm for SCCC with MSK in detail in the following section.

3.2.2.1 The decoding algorithm for inner decoder

Let $r_{I,k}$ and $r_{Q,k}$ be the received signal values from AWGN channel in the I part and Q part, respectively at time k. Let u_k^I be the input bit at the inner encoder taking values 0 or 1 that is associated with the transition from time $k-1$ to time k. x_I and x_Q are the corresponding I part and Q part values in the signal space as shown in Figure 3.5. The trellis states at time $k-1$ and k are indexed by the integers s' and s. For binary convolutional encoder, the MAP algorithm can be derived by:

$$L(\hat{u}_k) = \ln \frac{P(u_k = 1 \mid r)}{P(u_k = 0 \mid r)} = \ln \frac{\sum_{(s',s,u_k=1} p(s',\, s,\, r)}{\sum_{(s',s\ \ u_k=0} p(s',\, s,\, r)} \tag{3.1}$$

where

$$\begin{aligned} p(s',s,r) &= p(s',r_{j<k}) \cdot p(s,r_k \mid s') \cdot p(r_{j>k} \mid s) \\ &= \underbrace{p(s',r_{j<k})}_{} \cdot \underbrace{P(s \mid s') \cdot p(r_k \mid s',s)}_{} \cdot \underbrace{p(r_{j>k} \mid s)}_{} \\ &= \alpha_{k-1}(s') \cdot \gamma_k(s',s) \cdot \beta_k(s) \end{aligned}$$

Here $r_{j<k}$ denotes the sequence of received symbols r_j from the beginning of the trellis up to time $k-1$ and $r_{j>k}$ is the corresponding sequence from time $k+1$ up to the end of the trellis. The forward recursion and backward recursion as mentioned before are represented by

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

and

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

respectively.

The branch transition probability between s' and s is given by (we drop k for simplicity)

$$\gamma(s',s) = p(r \mid u^I) \cdot P(u^I),$$

where r is the received signal consisting of the two orthogonal component r_I and r_Q. Hence, we can get the branch transition of the inner decoder:

$$\begin{aligned} \gamma_{in}(s',s) &= p(r_I \mid u^I) \cdot p(r_Q \mid u^I) \cdot P(u^I) \\ &= \frac{1}{\sqrt{2\pi}\,\sigma} \exp\{-\frac{(r_I - x_I)^2}{2\sigma^2}\} \cdot \frac{1}{\sqrt{2\pi}\,\sigma} \exp\{-\frac{(r_Q - x_Q)^2}{2\sigma^2}\} \cdot P(u^I) \\ &= \frac{1}{2\pi\sigma^2}\{\exp\left[-\frac{r_I^2 + r_Q^2}{2\sigma^2}\right] \exp\left[-\frac{x_I^2 + x_Q^2}{2\sigma^2}\right] \qquad (3.2) \\ &\quad \exp\left[\frac{r_I x_I + r_Q x_Q}{\sigma^2}\right]\} \cdot P(u^I) \end{aligned}$$

In Equation 3.2, $r_I^2 + r_Q^2$, which is independent of branch transitions, is the energy of the received symbol from time $k-1$ to time k. $x_I^2 + x_Q^2$ is the energy of the branch output symbol from state s' to s in the trellis. Then $x_I^2 + x_Q^2$ is equal to the constant A^2 since the signal space is an equal-energy constellation. So, Equation 3.2 becomes:

$$\gamma_{in}(s', s) = A_k \cdot B_k \cdot \exp\left[\frac{r_{I,k}x_{I,k} + r_{Q,k}x_{Q,k}}{\sigma^2}\right] \cdot \exp\left[(u_k^I - \frac{1}{2}) \cdot L(u_k^I)\right].$$

Here, we have used the expression for $P(u^I)$ given by the Equation 2.1.

$L(u_k^I)$ is the *a priori probability* derived from the outer decoder which is equal to zero in the first iteration. The terms A_k and B_k are equal for all transitions and hence will be cancelled out in the ratio of Equation 3.1. Taking the logarithm of $\gamma_{in}(s', s)$, we obtain:

$$\begin{aligned} \gamma_{in}^{LM}(s', s) &= \ln \gamma_{in}(s', s) \\ &= \frac{r_{I,k}x_{I,k} + r_{Q,k}x_{Q,k}}{\sigma^2} + (u_k^I - \frac{1}{2}) \cdot L(u_k^I) \end{aligned}$$

Now, we denote max*-log-MAP as Max-log-MAP with correction term. We briefly define the following,

$$\begin{aligned} max*(a, b) &= \ln(e^a + e^b) \\ &= \max(a,\, b) + \ln(1 + e^{-|b-a|}) \\ &= \max(a,\, b) + \delta(a,\, b) \end{aligned} \tag{3.3}$$

and $\delta(a,\, b)$ is a correction term approximating $\ln(1 + e^{-|b-a|})$. If the correction term is $\ln(1 + e^{-|b-a|})$, then max*-log-MAP is exactly log-MAP. We use Max-log-MAP with correction term, i.e., max^*, to compute α and β.

Hence,

$$\begin{aligned} \alpha_{in,k}^{LM}(s) &= \ln\{\exp[\ln(\sum_{s'} \gamma_{in,k}(s', s) \cdot \alpha_{in,k-1}(s'))]\} \\ &= max_{s'}^*\{[\gamma_{in,u_k=1}^{LM}(s', s) + \alpha_{in,k-1}^{LM}(s')], [\gamma_{in,u_k=0}^{LM}(s', s) + \alpha_{in,k-1}^{LM}(s')]\} \end{aligned}$$

$$\beta_{in,k}^{LM}(s') = max_s^*\{[\gamma_{in,u_k=1}^{LM}(s',s) + \beta_{in,k+1}^{LM}(s)], [\gamma_{in,u_k=0}^{LM}(s',s) + \beta_{in,k+1}^{LM}(s)]\}$$

and the log likelihood ratio of the *a posteriori* of the input information symbols at the inner encoder becomes:

$$\begin{aligned} L(\hat{u}_k^I) &= \ln \frac{\sum_{(s',s),u_k=1} e^{\ln[\alpha_{k-1}(s')\cdot\gamma_k(s',s)\cdot\beta_k(s)]}}{\sum_{(s',s),u_k=0} e^{\ln[\alpha_{k-1}(s')\cdot\gamma_k(s'.s)\cdot\beta_k(s)]}} \\ &= max_{u_k=1}^*\{\gamma_{in}^{LM}(s',s) + \alpha_{in,k-1}^{LM}(s) + \beta_{in,k}^{LM}(s')\} \\ &\quad -max_{u_k=0}^*\{\gamma_{in}^{LM}(s',s) + \alpha_{in,k-1}^{LM}(s) + \beta_{in,k}^{LM}(s')\} \end{aligned}$$

Subtracting the *a priori probability* $L(u_k^I)$ from $L(\hat{u}_k^I)$, we get the extrinsic information $L^e(\hat{u}_k^I)$ which will be passed to the de-interleaver and then to the outer decoder.

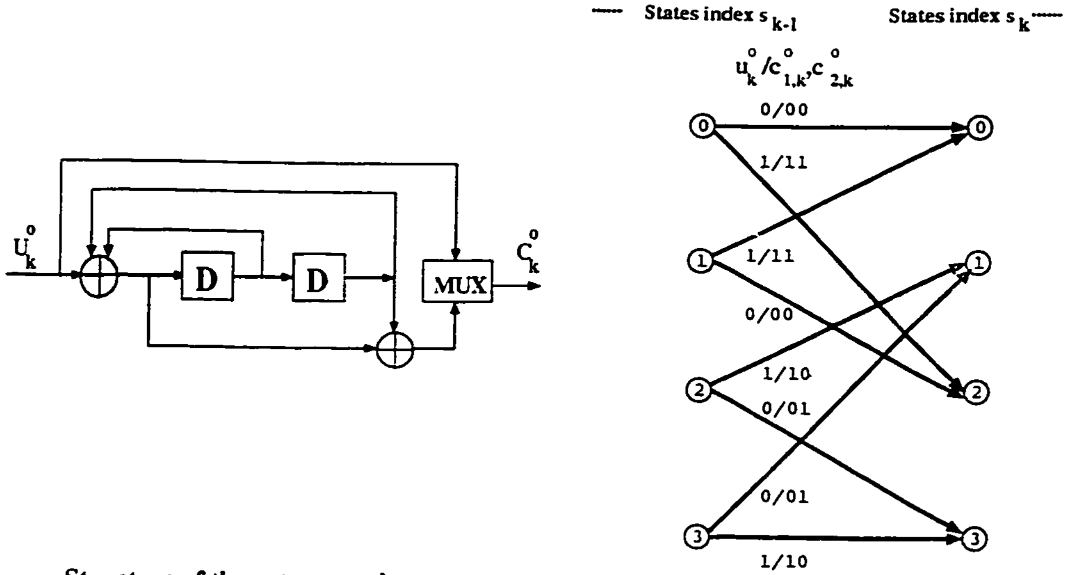
$$L^e(\hat{u}_k^I) = L(\hat{u}_k^I) - L(u_k^I)$$

3.2.2.2 The decoding algorithm for the outer decoder

In Figure 3.3, the outer decoder receives the permuted extrinsic information from the inner decoder instead of channel values. The decoding algorithm is similar to the one employed in the inner decoder. The structure of the outer encoder and the corresponding four-state trellis are shown in Figure 3.6. The branch transitions are denoted by *input bit* u_k^o / *coded bits* $c_{1,k}^o$, $c_{2,k}^o$. For the recursive systematic outer codes, $c_{1,k}^o = u_k^o$. Different from the inner decoder, the outer decoder will produce the a posteriori probability of both the information bits and the coded bits.

The log-likelihood ratio of the information u, coded bits c_1, c_2 for the outer codes are:

$$\begin{aligned} L(\hat{u}_k^o) &= max_{u_k=1}^*\{\gamma_{out}^{LM}(s',s) + \alpha_{out,k-1}^{LM}(s) + \beta_{out,k}^{LM}(s')\} \\ &\quad -max_{u_k=0}^*\{\gamma_{out}^{LM}(s',s) + \alpha_{out,k-1}^{LM}(s) + \beta_{out.k}^{LM}(s')\} \end{aligned}$$



Structure of the outer encoder

Trellis for the outer encoder

Figure 3.6: Structure of the outer encoder and corresponding trellis

$$
\begin{aligned}
L(c^o_{1,k}) \;=\;& max^*_{c_{1,k}=1}\{\gamma^{LM}_{out}(s',s)+\alpha^{LM}_{out,k-1}(s)+\beta^{LM}_{out,k}(s')\} \\
& -max^*_{c_{1,k}=0}\{\gamma^{LM}_{out}(s',s)+\alpha^{LM}_{out,k-1}(s)+\beta^{LM}_{out,k}(s')\}
\end{aligned} \tag{3.4}
$$

$$
\begin{aligned}
L(c^o_{2,k}) \;=\;& max^*_{c_{2,k}=1}\{\gamma^{LM}_{out}(s',s)+\alpha^{LM}_{out,k-1}(s)+\beta^{LM}_{out,k}(s')\} \\
& -max^*_{c_{2,k}=0}\{\gamma^{LM}_{out}(s',s)+\alpha^{LM}_{out,k-1}(s)+\beta^{LM}_{out,k}(s')\}
\end{aligned} \tag{3.5}
$$

The extrinsic information of the coded bits c_1 and c_2 will be obtained by subtracting the a priori probability which is the corresponding permuted a posteriori probability from the inner decoder from Equations 3.4 and 3.5. This extrinsic information will be employed as the a priori probabilities to feed the inner decoder to start the next iteration. $L(\hat{u}^o_k)$ will be used in the final iteration to make estimation of the input information sequence. If $L(\hat{u}^o_k) \geq 0$, we decide $\hat{u}^o_k = 1$, and we let $\hat{u}^o_k = 0$ if $L(\hat{u}^o_k) < 0$.

3.3 Simulation Results

In this section we present some simulation results and analysis based on the above mentioned encoding structure and decoding algorithm. The simulation results comply with the theoretical analysis.

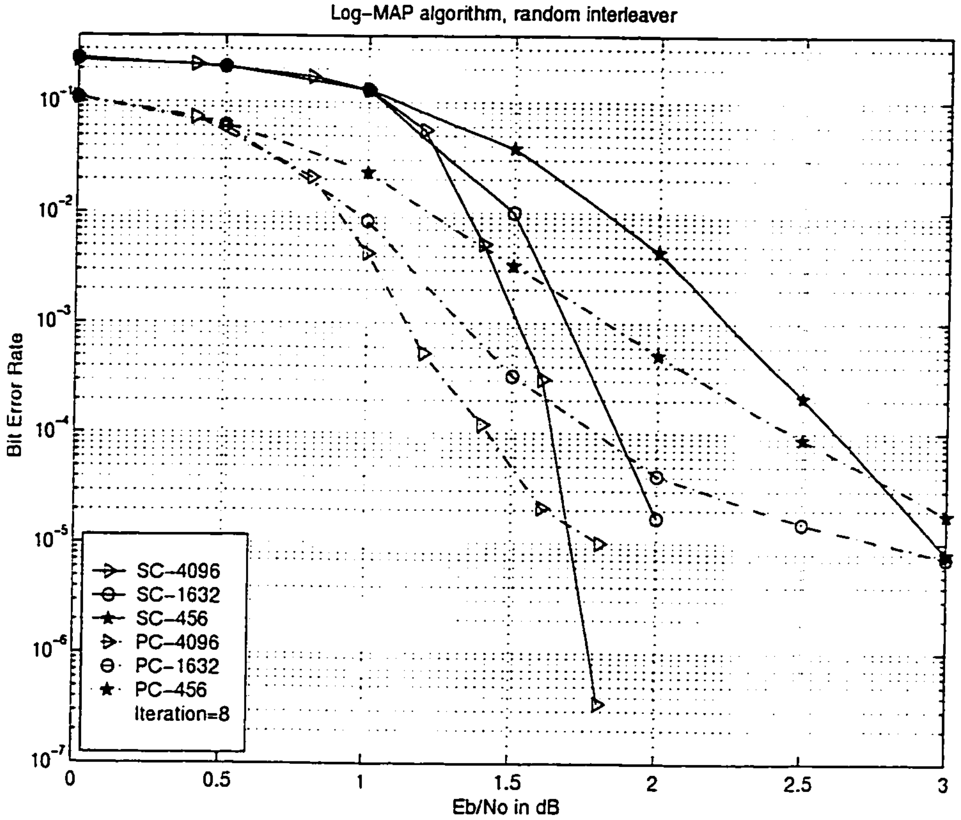


Figure 3.7: Comparison of SCCC with MSK and PCCC with BPSK under the same bandwidth efficiency

Figure 3.7 shows the simulation result of comparing SCCC with MSK and PCCC with BPSK modulation over AWGN channel at three different block lengths. The structure of the two component encoders of PCCC are the same as the outer encoder of SCCC with MSK. The decoding algorithm is the iterative log-MAP algorithm as described in [13]. The coded bits for PCCC will be punctured to code rate 1/2. The bandwidth efficiency of PCCC with BPSK modulation is the same as SCCC with MSK, i.e.. 1/2 bps/Hz. Random interleaver is used for both schemes. The solid lines and dashed lines represent the BER curves of SCCC with MSK and PCCC with BPSK, respectively. We can see that SCCC with MSK modulation outperforms PCCC with BPSK at high signal-to-noise ratio (E_b/N_0) for all three block lengths. The cross points for each pair of corresponding lines, i.e., same block length, are around 10^{-5}. The error floor is absent, or at least, much lower in the case of SCCC with MSK than PCCC with BPSK.

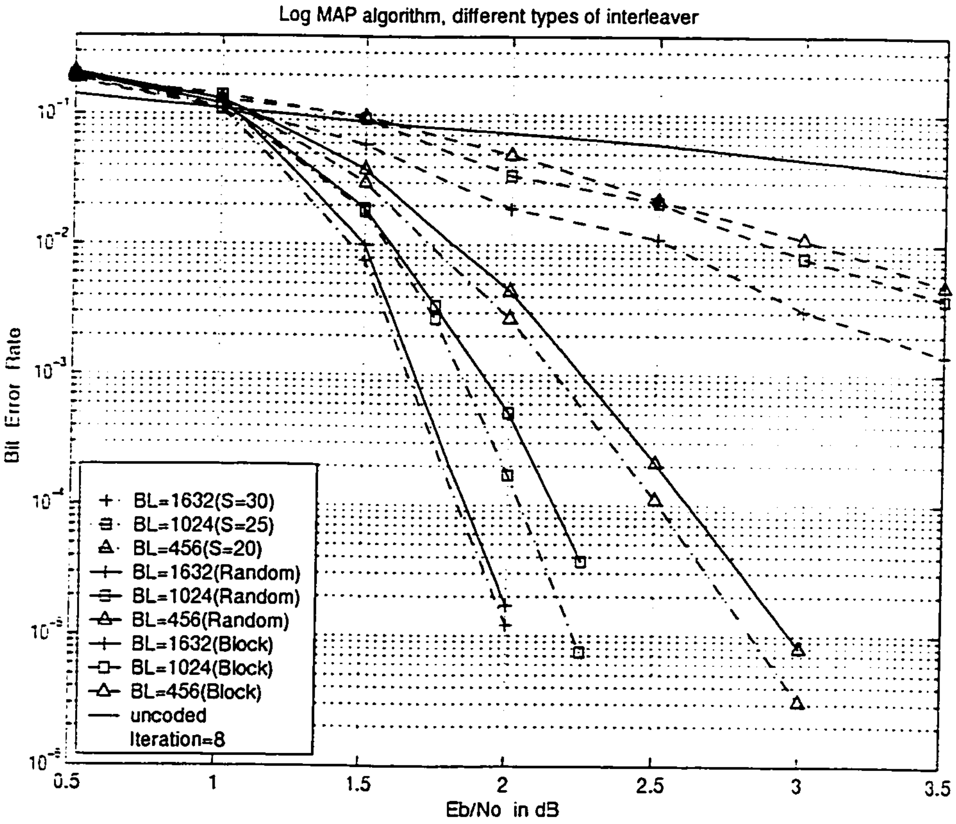


Figure 3.8: Performance of SCCC with MSK using different types of interleaver

Figure 3.8 illustrates the performance of SCCC with MSK under three block lengths using different interleavers. The dashed lines show the bit error probabilities using block interleaver. The performance does not change noticeably with the increase of the block length. The solid lines and dash-dotted lines represent the performance with pseudo-random and S-random interleavers. We can see that by using S-random interleaver the performance can be improved by about $0.15\,dB$ under block lengths of 1024 and 456 but no noticeable improvement is observed for block length 1632. The possible reason is that the minimum interleaving distance $S = 30$ is not big enough for size of 1632 to break low weight input sequences [1]. In general, the larger the value of S is, the better the performance will be. However, the search time for S-random interleavers will become fairly long for large values of S compared to random interleavers. for a given interleaver size S, the value of S is bounded by $S < \sqrt{N/2}$ [22]. Both types of interleaver can achieve much better performance than block interleaver.

In Figure 3.9, it is obvious that the performance is affected by the interleaver size. By increasing the interleaver size N, the bit error probability is reduced by a factor. This is called the interleaver gain. For interleaver size of N and outer code free distance d_f^o, it has been shown in [18] that in theory the interleaver gain is $N^{-d_f^o/2}$ for even values of d_f^o and $N^{-(d_f^o+1)/2}$ for odd values of d_f^o. In our case of SCCC with MSK, the outer code free distance $d_f^o = 5$ so that the interleaver gain is N^{-3}. The error probability decreases sharply when increasing the length of the input information block. It is about $1.2\,dB$ interleaving gain between block length 4096 and 456.

Figure 3.10 shows the BER curves of SCCC with MSK at different number of iterations under a fixed block length of 1632. The performance can be improved by increasing the number of iterations. There is about $0.5\,dB$ gain when the number of iteration is increased from 4 to 8. However, the improvement is very limited when

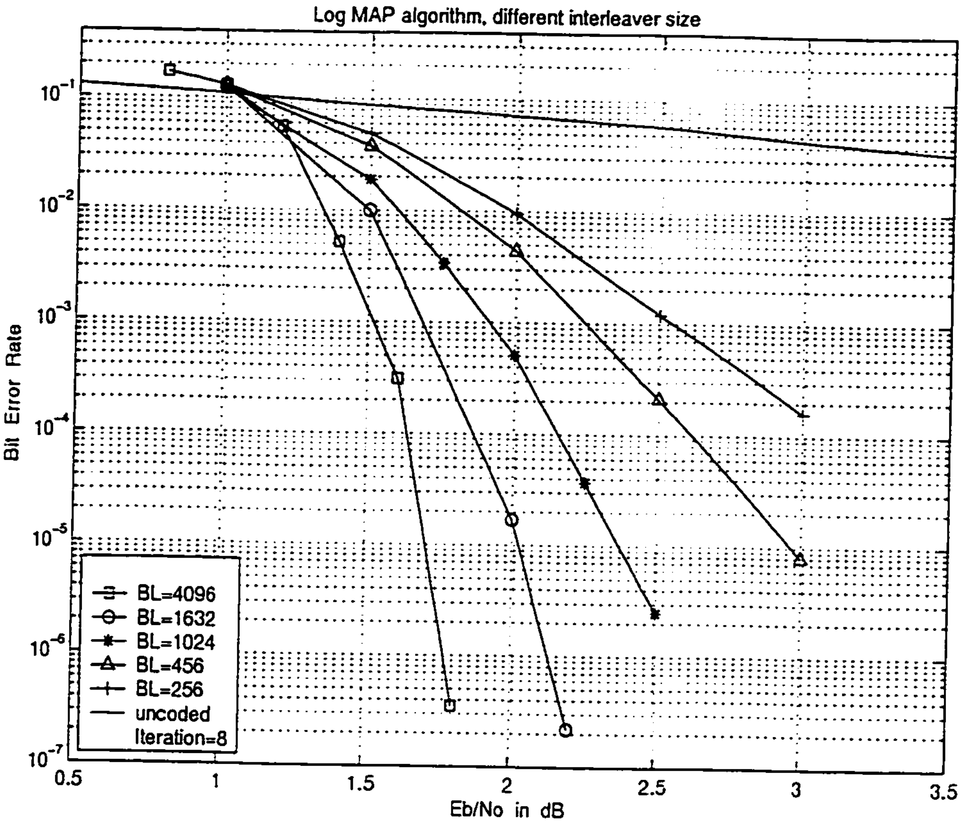


Figure 3.9: Performance of SCCC with MSK with different interleaver sizes

the number of iteration is over 8, for example, less than $0.2dB$ gain between 18 iterations and 8 iterations. In addition, more iterations mean higher complexity. Thus in terms of the acceptable performance, we choose the number of iterations to be 8 in most of our simulations to get a good trade-off between performance and complexity.

In previous simulations, we use the log-MAP algorithm in all decoding processes. Figure 3.11 illustrates the performance comparison of employing log-MAP, Max log-MAP (without correction term) and max*-log-MAP (with correction term) decoding algorithms. Two block lengths are simulated for comparison. The solid lines and dashed lines represent SCCC with MSK at block size 1632 and 456 with 8 iterations,

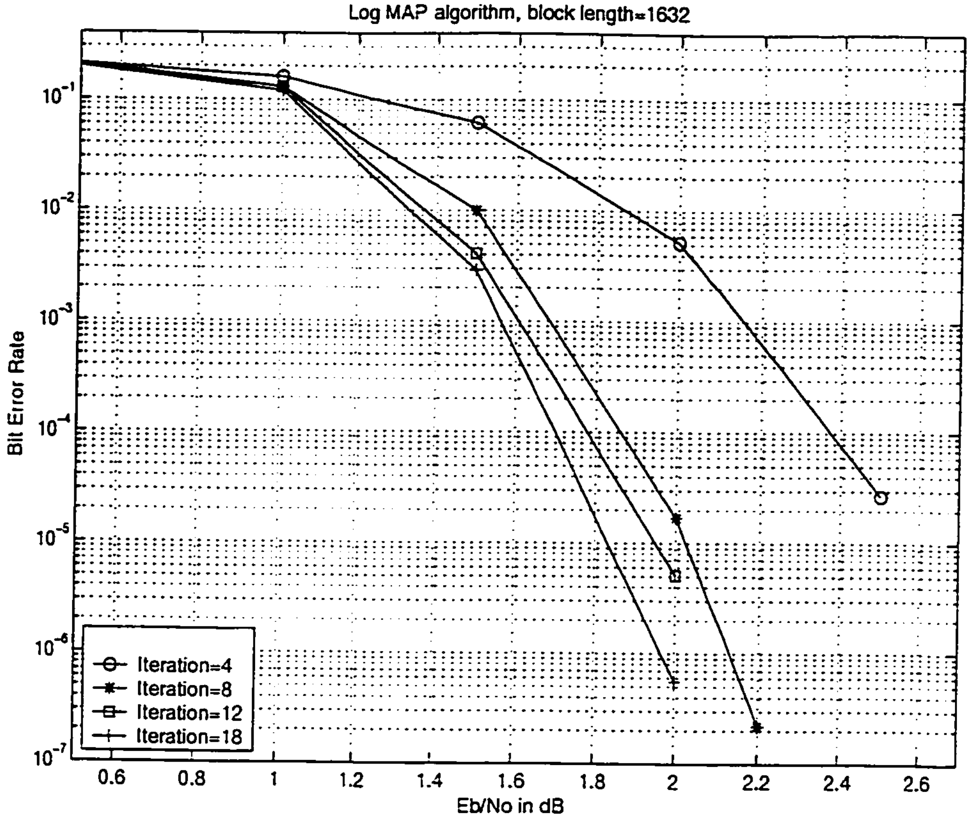


Figure 3.10: Performance curves of SCCC with MSK with different iterations

respectively. For max*-log-MAP algorithm, we use look-up table for the correction term $\delta(a,\ b)$ in Equation 3.3. An 8-value look-up table with a step-size of 1/8 of the correction term is used. The range of the term| $b - a$ |is between 0 and 5. We may see from the figure that the performance of using log-MAP algorithm is about 0.5 dB better than Max-log-MAP (without correction term) algorithm. The performance of max*-log-MAP algorithm is between the two with medium complexity. We use log-MAP as our main decoding algorithm. However, the complexity is higher for log-MAP algorithm than the other two algorithms. In practical systems, max*-log-MAP is more suitable.

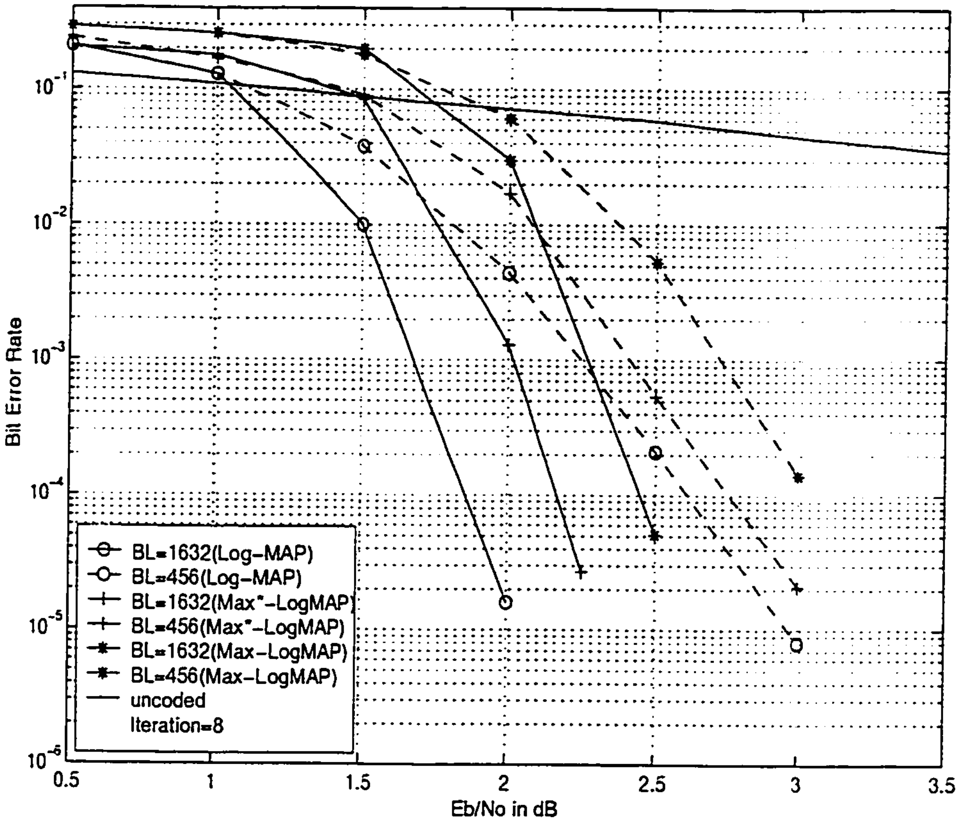


Figure 3.11: Performance comparison of log-MAP and Max log-MAP algorithm

3.4 Summary

In this chapter, the design of serial concatenated convolutional code with the application of coded modulation technique minimum shift keying (MSK) over AWGN channel was investigated. The properties of constant envelope and spectral efficiency for continuous phase modulation (CPM) techniques, such as MSK make them attractive in wireless communication systems. The decomposition structure of CPM (MSK) introduces memories so that it can be combined in SCCC as the inner encoder and modulator. Iterative MAP decoding algorithms for both the outer decoder and the inner decoder are derived and simulation results are given based on the decoding algorithms.

From the simulation results, we may conclude:

- Serial concatenated convolutional codes using MSK as the inner code can be considered as a valid alternative to turbo codes. SCCC with MSK shows excellent performance at medium to high signal-to-noise ratios and has significantly lower change of slope in the bit error probability curves than PCCC, i.e., no error floor found at 10^{-7} for different block sizes.

- Interleaver type will affect the performance greatly as we can see from the simulation results. Random or S-random interleaver can be chosen as the interleaver between the outer and the inner encoder to achieve better performance. Block interleaver should not be used in the case of SCCC with MSK.

- Since the interleaver gain, which is the factor of decreasing the bit error probability, is a function of the interleaver size and free distance of the outer codes, the performance improves if we increase the interleaver size or choose outer code with a larger free distance.

- The number of iterations is another factor that influences the performance. Taking the decoding complexity into account, we use 8 iterations to achieve good performance while keeping relatively low complexity.

AWGN channel is the most used channel model in wireless and satellite communication systems. We will discuss SCCC with coded MSK over more complicated channels in the next chapter.

Chapter 4

SCCC with Coded MSK over ISI Channel

In previous chapters, we discussed the design, performance analysis and iterative decoding of SCCC with MSK over Gaussian channels. It is observed that SCCC with MSK can achieve remarkably low bit error rate with iterative log-MAP decoding algorithm over AWGN channel. However, in real communication systems, such as radio, satellite and mobile communication systems, besides Gaussian noise, inter-symbol interference (ISI) caused by limited bandwidth, multipath propagation and motion can distort the transmitted signals at the receiver and generate serious degradation in performance. It is useful to study and evaluate the performance of SCCC with MSK modulation over ISI channel.

The transmitted pulses through the non-ideal channel at rates comparable to the channel bandwidth W are smeared to the point that they are no longer distinguishable as well-defined pulses at the receiving terminal [2] [41]. Let us take the example of systems with a data rate of 150 $kbit/s$ (close to ISDN data rate of 144 $kbit/s$), the required transmission rate would be approximately 75 $ksymbol/s$ assuming rate 1/2 coded 16 QAM signalling with a symbol period of 13.3 μs or approximately 6.9 μs for 150 $ksymbol/s$ system with rate 1/2 coded QPSK signalling. It is obvious that the multipath delay spread in the order of tens of microseconds will cause significant frequency selective fading, i.e., inter-symbol interference (ISI) in these situations and the system performance will degrade significantly over such fading channel [2][42]. In digital cellular radio systems such as GSM, some environment multipath components may be delayed by up to 30 μs which cause ISI and thus affect the received signal quality greatly [43]. Therefore, ISI has been recognized as the major obstacle to high speed data transmission over mobile radio channels. It is important to find a way to eliminate or reduce the effect of ISI to improve the performance.

For communication channels where the channel frequency response characteristics are unknown a priori or are time-varying, optimum modulation filter at the transmitter and demodulation filter at the receiver can't be designed directly. In order to eliminate or reduce the effect of ISI, one solution is to combine the inner encoder and the ISI channel which can be treated as a Finite State Machine (FSM) to create a super trellis and use ML or symbol-by-symbol MAP algorithms to decode. Better performance can be obtained through this method. However, the total number of states in the resulting super trellis will become large due to the combination of the inner encoder and the channel memories and the trellis will be fairly complicated. Thus the decoding complexity increases dramatically. Another solution is to use an equalizer to compensate or reduce the ISI of the received signal. Though the performance may not be as good as the previous solution, the decoding complexity decreases significantly and the performance is acceptable. The ISI channel model, the criteria of the equalization algorithm and equalization algorithms will be discussed in this chapter.

4.1 Design of Equalizer to Compensate ISI Channel

4.1.1 System model

As is well known, the equivalent discrete-time ISI channel can be regarded as a nonrecursive nonsystematic convolutional code with memory L and rate-1 [44], whose channel coefficients h_i may vary in time. Figure 4.1 shows the general model of serial concatenated convolutional code with MSK over an ISI channel with the additive Gaussian noise and the general transversal structure of the equalizer. The nonrecursive nonsystematic form of channel can also be considered as a tapped delay line with tap spacing at symbol duration time T. The output y_n from the ISI channel will be the convolution sum of the modulator output s_n and the channel impulse response h_n. Suppose the coefficients h_i, $i = 0, 1, ..., L$ of the impulse response are invariant during the period of one block length. The signal is assumed to be corrupted by the addition of white Gaussian noise. Thus the received signal at time n can be expressed as:

$$\begin{aligned} r_n &= y_n + \eta_n \\ &= \sum_{i=0}^{L} h_i \cdot s_{n-i} + \eta_n \\ &= h_n \otimes s_n + \eta_n \end{aligned} \qquad (4.1)$$

where η_n is Gaussian noise with zero mean and variance σ^2, $\otimes$ represents the convolution operation. Then the corrupted signals will enter the equalizer and the output from the equalizer q_n is fed to the iterative log-MAP Inner/Outer decoder described in the previous chapters to obtain the estimate value $\hat{u}$.

As we mentioned earlier, equalization is an effective way of compensating for intersymbol interference in terms of complexity. In general, most of the channel characteristics are unknown a priori and are time varying. Equalizer must track the

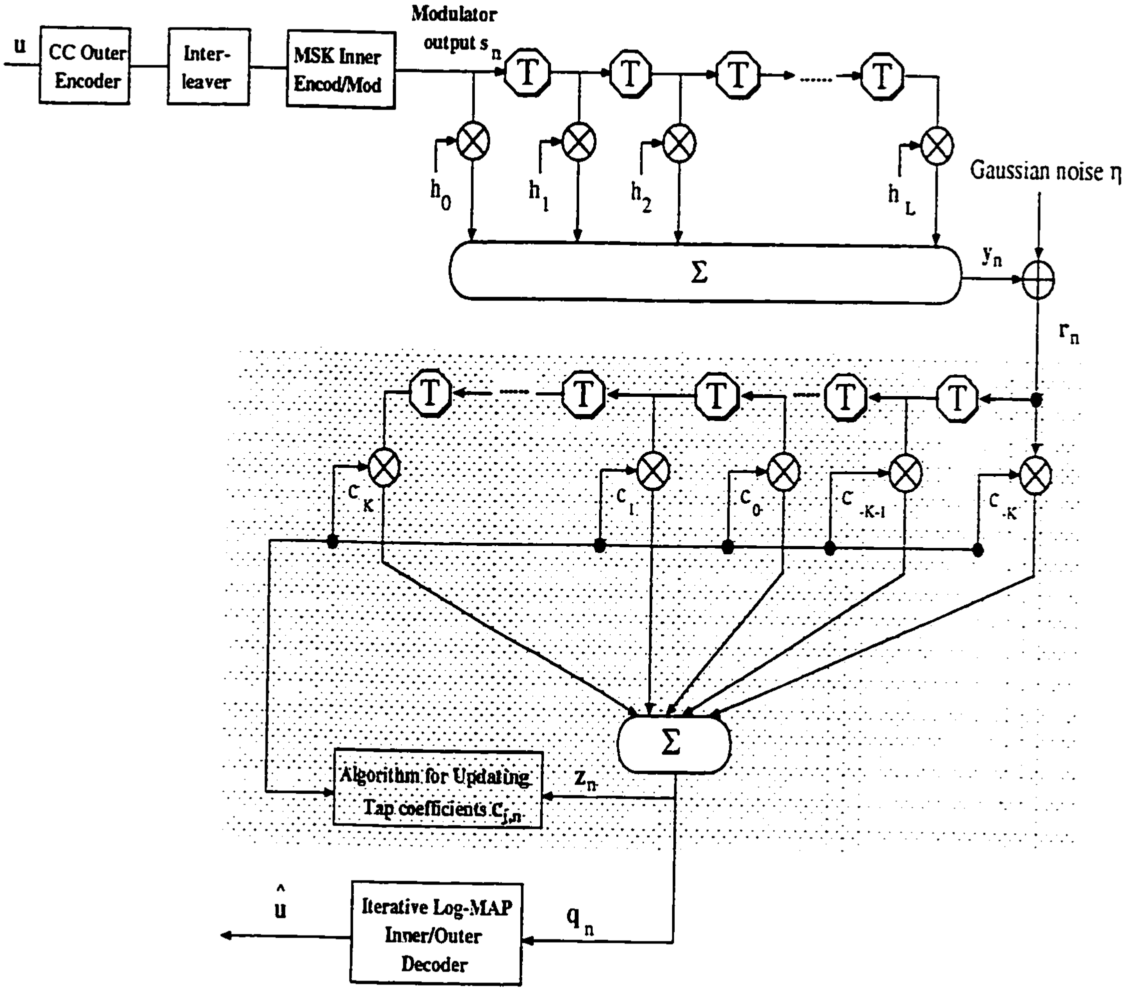


Figure 4.1: General model of SCCC with MSK over ISI channel

time varying characteristics of the channel and adjust the coefficients c_j^n accordingly. These equalizers are called adaptive equalizers and are widely used in real systems. Equalization techniques can be divided into two general categories - linear and nonlinear equalization. These categories are determined from how the output of an adaptive equalizer is used for subsequent control (feedback) of the equalizer [38]. In general, the equalizer output sequence q_n is processed by the decision making device at the receiver. The decision maker determines the value of the data sequence being received and applies a slicing or thresholding operation (a nonlinear operation) in order to obtain the estimated sequence. If the estimated sequence is not used in the feedback path to adapt the equalizer, the equalization is linear. On the other hand, if the estimated sequence is fed back to change the subsequent outputs of the equalizer, the equalization is nonlinear. In terms of structures, there are two main classes, transversal and lattice. For each structure, there are numerous algorithms used to adapt the equalizer. Figure 4.2 provides a general categorization of the equalization techniques according to the types, structures and algorithms used [38].

The most common equalizer structure is a linear transversal equalizer (LTE). A linear transversal filter is made up of tapped delay lines, with the tappings spaced a symbol period apart. We use LTE as our equalizer structure, as shown in Figure 4.1, where there are $2K$ delay elements and $2K+1$ equalizer taps with coefficients c_j and T is the symbol duration. Among the algorithms, the zero-forcing algorithm and the Least Mean Square (LMS) algorithm are discussed in the following paragraph. The Recursive Least-Squares (RLS) algorithm is not covered in this thesis.

The general operating modes of an adaptive equalizer include training and tracking. First, a known, fixed-length training sequence is sent by the transmitter so that the equalizer at the receiver may initialize to a proper setting. The training sequence is typically a pseudo-random binary signal or a fixed, prescribed bit pattern that is known by both the transmitter and the receiver. Following the training sequence, the user data is sent. The equalizer utilizes the training sequence and a certain kind of equalization algorithm to evaluate the channel and estimate filter coefficients to compensate for the channel. Therefore, the filter coefficients are close the optimal values for reception of user data after the training sequence is finished [45]. Since the channel characteristics are changing over time, equalizers require periodic retraining to adjust the coefficients in order to maintain effective ISI cancellation. These

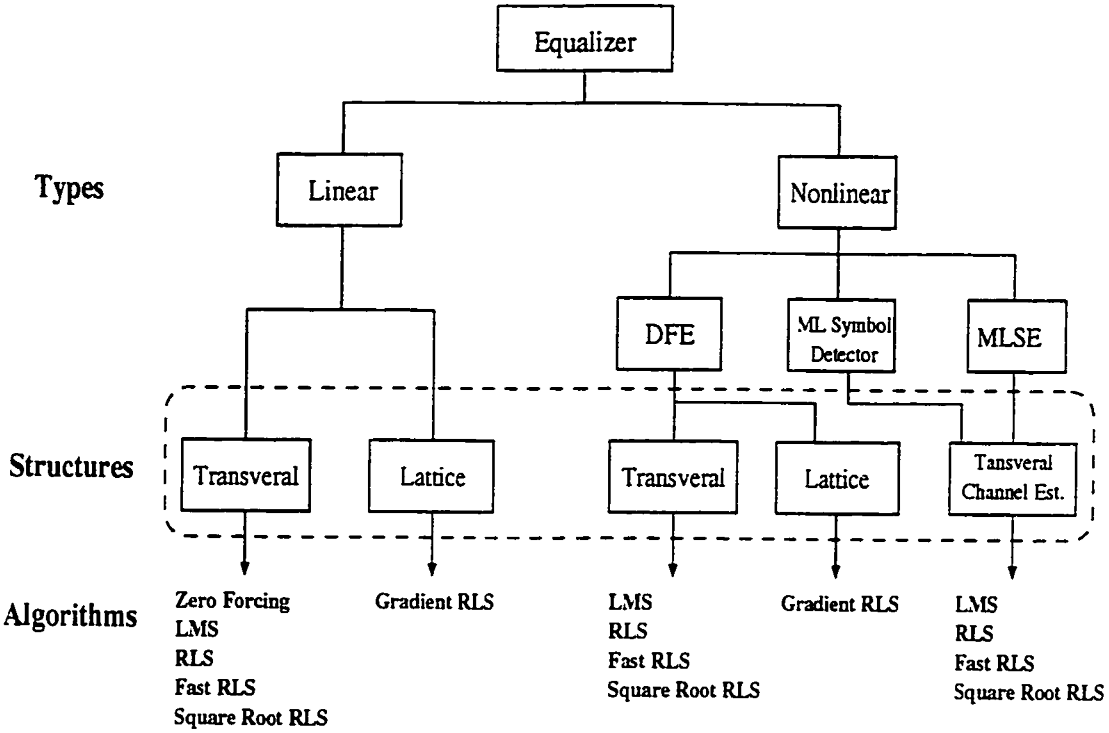


Figure 4.2: Classification of Equalizer

coefficients (or weights) are updated continuously by the adaptive algorithm, either on a sample by sample basis or on a block by block basis, i.e., a specified number of samples input to the equalizer. They are commonly used in digital communication systems where user data is segmented into short time blocks. Time division multiple access (TDMA) wireless systems are particularly well suited for these types of equalizers. For example, in GSM systems, equalization is performed at the receiver with 26 *bits* training sequence out of 156.25 *bits* transmitted in the midamble of every time slot [46].

However, the effective data rate of the communication link will be lowered by sending the training sequence periodically to optimize the coefficients of the equalizer. In many high data rate, bandlimited digital communication systems, the transmission of a training sequence will be very costly or impractical. In some applications, for instance, no training signal is available in trying to intercept enemy communications. Another example is in a multi-user broadcast system, it is highly undesirable for the transmitter to engage in a training session for a single user by temporarily terminating its normal transmission to a number of other users [47]. Thus another class of adaptive algorithms called blind equalization algorithms which do not rely on training signals are developed. The self-recovering ability of blind equalizer is very attractive to short burst formats used in existing wireless communication applications using Time Division Multiple Access (TDMA) technique such as IS-136, GSM and packet data systems [48]. The objective of blind equalization is to recover the unknown input sequence based only on its probabilistic and statistical properties. The cost function of the blind equalization algorithm, which implicitly involves higher order statistics of the channel output, is different from that of the non-blind one [47]. The most popular and effective blind equalization algorithm is the constant modulus algorithm (CMA) which is used for constant envelope modulation and forces the equalizer weights to maintain a constant envelope on the received signal. Blind equalization algorithms are not covered in this thesis.

4.1.2 Criteria for equalization algorithms

The objective of the equalizer is to optimize the filter coefficients $\{c_j\}$ so as to remove the ISI as much as possible and thus reduce the probability of error. Since the probability of error is a highly nonlinear function of $\{c_j\}$, so the probability of error as a performance index for optimizing the tap weights of the equalizer is impractical. The solution is to find a certain kind of criterion as a measure to adjust the coefficients to their optimal values. Two criteria are widely used in optimizing the equalizer coefficients $\{c_j\}$. One is the peak distortion criterion and the other is the mean square error criterion.

4.1.2.1 Peak distortion criterion

The peak distortion is defined as the worst-case intersymbol interference at the output of the equalizer. Suppose the discrete-time linear filter model having an impulse response $\{h_n\}$ and an equalizer with an infinite number of taps having an impulse response $\{c_n\}$, a single equivalent filter is the convolution of $\{c_n\}$ and $\{h_n\}$

$$\begin{aligned} d_n &= c_n \otimes h_n \\ &= \sum_{j=-\infty}^{\infty} c_j\, h_{n-j} \end{aligned} \tag{4.2}$$

The estimate q_k output of the equalizer will be the convolution of the received sequence r_k and the equalizer impulse response c_j:

$$\begin{aligned} q_k &= r_k \otimes c_k \\ &= (y_k + \eta_k) \otimes c_k \\ &= (s_k \otimes h_k + \eta_k) \otimes c_k \\ &= s_k \otimes h_k \otimes c_k + \eta_k \otimes c_k \\ &= s_k \otimes d_k + \eta_k \otimes c_k \end{aligned} \tag{4.3}$$

Combining Equations 4.2 and 4.3, we get

$$\begin{aligned} q_k &= \sum_{n=-\infty}^{\infty} c_n \cdot r_{k-n} \\ &= \sum_{n=-\infty}^{\infty} s_n d_{k-n} + \sum_{j=-\infty}^{\infty} c_j \eta_{k-j} \\ &= s_k d_0 + \sum_{n \neq k} s_n d_{k-n} + \sum_{j=-\infty}^{\infty} c_j \eta_{k-j} \end{aligned} \tag{4.4}$$

The first term in Equation 4.4 represents a scaled version of the desired symbol. For convenience, we normalize d_0 to unity. The second term is the inter-symbol interference. Then the peak distortion is defined as the peak value of the interference, which can be represented as

$$\begin{aligned} \Lambda(c) &= \sum_{n=-\infty, n\neq 0}^{\infty} |d_n| \\ &= \sum_{n=-\infty, n\neq 0}^{\infty} \left| \sum_{j=-\infty}^{\infty} c_j \, h_{n-j} \right| \end{aligned} \tag{4.5}$$

With an equalizer having an infinite number of taps, it is possible to select the tap weights so that $\Lambda(c) = 0$, i.e., $d_n = 0$ for all n except $n = 0$. From Equations 4.4 and 4.5, we infer that the inter-symbol interference can be completely eliminated. The value of the coefficients are determined from the condition:

$$\begin{aligned} d_n &= \sum_{j=-\infty}^{\infty} c_j \, h_{n-j} \\ &= \begin{cases} 1 & (n = 0) \\ 0 & (n \neq 0) \end{cases} \end{aligned}$$

However, infinite length equalizers are not available in real systems. Since $c_j = 0$ for $|j| > K$, the convolution of $\{h_n\}$ with $\{c_n\}$ is zero outside the range $-K \leq n \leq K + L$. The peak distortion for equalizer with $2K + 1$ coefficients and channel with $L + 1$ tap weights is:

$$\begin{aligned} \Lambda(c) &= \sum_{n=-K, n=0}^{K+L} |d_n| \\ &= \sum_{n=-K, n\neq 0}^{K+L} \left| \sum_{j=-K}^{K} c_j \, h_{n-j} \right| . \end{aligned}$$

Although the equalizer has $2K + 1$ adjustable parameters, there are $2K + L + 1$ nonzero values in the response $\{d_n\}$. Therefore it is generally impossible to completely eliminate the inter-symbol interference at the output of the equalizer. There is always some residual interference when the optimum coefficients are used. Zero-forcing equalization algorithm is based on the minimization of peak distortion at the output of the equalizer.

4.1.2.2 Mean Square Error (MSE) Criterion

In the MSE criterion, the tap weight coefficients $\{c_j\}$ of the equalizer are adjusted to minimize the mean square value of the error

$$\varepsilon_k = s_k - q_k$$

where s_k is the signal output from the modulator transmitted in the kth signalling interval and q_k is the estimate of the corresponding symbol at the output of the equalizer. When the signal sequence $\{s_k\}$ is complex-valued, the performance index for the MSE criterion is defined as:

$$\begin{aligned} \Gamma &= E\left|\varepsilon_k\right|^2 \\ &= E\left|s_k - q_k\right|^2 \end{aligned}$$

If the information symbols are real-values, the performance index is simply the square of the real part of ε_k. In either case, Γ is a quadratic function of the equalizer coefficients $\{c_j\}$. In our case, the input signals $\{s_k\}$ are complex-valued. The Least Mean Square algorithm is based on the minimization of the mean square error at the output of the equalizer.

4.1.3 Zero-forcing algorithm

In the peak distortion criterion, the peak distortion Λ_c is minimized by selecting the equalizer coefficients $\{c_k\}$, assuming $\{c_k\}$ is real. In general, there is no simple computational algorithm for performing this optimization. Here we discuss a special case where the peak distortion at the input to the equalizer, defined as

$$\Lambda_0 = \frac{1}{|h_0|} \sum_{i=1}^{L} |h_i|$$

is less than unity. Under this condition, the distortion Λ_c at the output of the equalizer is minimized by forcing the equalizer response $d_n = 0$ for $1 \leq |n| \leq K$, and $d_0 = 1$. In this case, the computational algorithm leads to a zero-forcing algorithm.

Assume that we have an equalizer with $2K + 1$ taps and $L + 1$ channel coefficients. The solution for the zero-forcing algorithm is achieved by forcing the cross-correlation equation function between the error sequence $\varepsilon_k = s_k - q_k$ and the desired information sequence $\{s_k\}$ to zero for shifts in the range $0 \leq |n| \leq K$. Since the signal sequence at the input of the equalizer is complex-valued, we take the expectation of ε_k and the conjugate of s_k and get

$$\begin{aligned} E\left(\varepsilon_k \cdot s^*_{k-m}\right) &= E\left[(s_k - q_k) \cdot s^*_{k-m}\right] \\ &= E\left[s_k \cdot s^*_{k-m} - q_k \cdot s^*_{k-m}\right] \qquad m = -K, ..., K \qquad (4.6) \end{aligned}$$

where $*$ denotes the conjugate operator. From the Equations 4.1 and 4.4 in Section 4.1.1, we have

$$r_k = \sum_{i=0}^{L} h_i \cdot s_{k-i} + \eta_k$$

$$q_k = \sum_{j=-K}^{K} c_j \cdot r_{k-j}$$

In addition, we assume that the symbols $\{s_k\}$ are uncorrelated and normalized to unity, i.e.,

$$\begin{aligned} E\left(s_k \cdot s^*_j\right) &= \delta_{k-j} \\ &= \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} \end{aligned}$$

and also the sequence $\{s_k\}$ is uncorrelated with the additive noise sequence $\{\eta_k\}$, i.e., $E(s_k^* \cdot \eta_j) = 0$. Then we can derive

$$\begin{aligned}
E\left(\varepsilon_k \cdot s_{k-m}^*\right) &= E\left[s_k \cdot s_{k-m}^* - \left(\sum_{j=-K}^{K} c_j \cdot r_{k-j}\right) \cdot s_{k-m}^*\right] \\
&= E\left(s_k \cdot s_{k-m}^*\right) - E\left\{\sum_{j=-K}^{K} c_j \cdot \left[\sum_{i=0}^{L} h_i \cdot s_{k-j-i} + \eta_{k-j}\right] \cdot s_{k-m}^*\right\} \\
&= \delta_m - E\left\{\sum_{j=-K}^{K} c_j \cdot \left[\sum_{i=0}^{L} h_i \cdot s_{k-j-i} \cdot s_{k-m}^* + \eta_{k-j} \cdot s_{k-m}^*\right]\right\} \\
&= \delta_m - \sum_{j=-K}^{K} c_j \cdot \left\{\sum_{i=0}^{L} h_i \cdot E\left[s_{k-j-i} \cdot s_{k-m}^*\right] + E\left[\eta_{k-j} \cdot s_{k-m}^*\right]\right\} \\
&= \delta_m - \sum_{j=-K}^{K} c_j \cdot \sum_{i=0}^{L} h_i \cdot E\left[s_{k-j-i} \cdot s_{k-m}^*\right] \qquad (4.7)
\end{aligned}$$

Only when $j + i = m$, or $i = m - j$, there is an impulse response. Then Equation 4.7 becomes

$$\begin{aligned}
E\left(\varepsilon_k \cdot s_{k-m}^*\right) &= \delta_m - \sum_{j=-K}^{K} c_j \cdot h_{m-j} \\
&= \delta_m - d_m \qquad m = -K, \dots K \qquad (4.8)
\end{aligned}$$

From Equation 4.8, we may derive $d_0 = 1$ when $m = 0$ and $d_m = 0$ for $1 \leq |m| \leq K$, which means the required condition $E\left(\varepsilon_k \cdot s_{k-m}^*\right)$ can be satisfied.

4.1.4 LMS algorithm

In the minimization of the mean square error (MSE) described in subsection 4.1.2.2, we found that the optimum equalizer coefficients can be determined from the solution of a set of linear equations. This will be discussed in the following paragraph.

Assuming that we have an equalizer with $2K + 1$ taps $\{c_k\}$ and $L + 1$ channel coefficients and the input signal sequence at the equalizer is complex-valued, suppose

$\{c_k\}$ is real. Then the mean square error is

$$\begin{aligned} \Gamma &= E\left|s_k - q_k\right|^2 \\ &= E\left[(s_k - q_k)\cdot(s_k^* - q_k^*)\right] \\ &= E\left(s_k \cdot s_k^*\right) + E\left(q_k \cdot q_k^*\right) - E\left(s_k \cdot q_k^* + s_k^* \cdot q_k\right) \end{aligned} \tag{4.9}$$

which is the function of the equalizer coefficients. To minimize Γ, it is required to set the derivative of Equation 4.9 with respect to the coefficients $\{c_i\}$ to zero, i.e.,

$$\begin{aligned} \frac{\partial\Gamma}{\partial c_i} &= \frac{\partial\left[E\left(s_k \cdot s_k^*\right)\right]}{\partial c_i} + \frac{\partial\left[E\left(q_k \cdot q_k^*\right)\right]}{\partial c_i} - \frac{\partial\left[E\left(s_k \cdot q_k^* + s_k^* \cdot q_k\right)\right]}{\partial c_i} \\ &= 0 \qquad\qquad i = -K, ..., K \end{aligned} \tag{4.10}$$

In Equation 4.10, signal sequence $\{s_k\}$ is not a function of c_i so the derivative of $(s_k \cdot s_k^*)$ in terms of c_i will be zero. We may exchange the expectation operation and the derivative operation in Equation 4.10. Thus we obtain

$$E\left[q_k^* \cdot \frac{\partial q_k}{\partial c_i} + q_k \cdot \frac{\partial q_k^*}{\partial c_i}\right] - E\left[s_k \cdot \frac{\partial q_k^*}{\partial c_i} + s_k^* \cdot \frac{\partial q_k}{\partial c_i}\right] = 0 \tag{4.11}$$

Since

$$q_k = \sum_{j=-K}^{K} c_j \cdot r_{k-j}$$

and

$$q_k^* = \sum_{j=-K}^{K} c_j \cdot r_{k-j}^*,$$

we get

$$\frac{\partial q_k}{\partial c_i} = r_{k-i} \qquad\qquad \frac{\partial q_k^*}{\partial c_i} = r_{k-i}^*. \tag{4.12}$$

Then Equation 4.11 becomes

$$E\left[\sum_{j=-K}^{K} c_j \cdot \left(r_{k-j}^* \cdot r_{k-i} + r_{k-j} \cdot r_{k-i}^*\right)\right] = E\left[s_k \cdot r_{k-i}^* + s_k^* \cdot r_{k-i}\right]$$

i.e.,

$$\sum_{j=-K}^{K} c_j \cdot E\left[r^*_{k-j} \cdot r_{k-i} + r_{k-j} \cdot r^*_{k-i}\right] = E\left[s_k \cdot r^*_{k-i} + s^*_k \cdot r_{k-i}\right] \quad i, j = -K, ..., K \tag{4.13}$$

Writing the above equation in detail, we get

$$\left\{ \begin{array}{lcl} c_{-K} \cdot 2ReE\left[r^*_{k+K} r_{k+K}\right] + \cdots + c_K \cdot 2ReE\left[r^*_{k-K} r_{k+K}\right] & = & 2ReE\left[s_k \cdot r^*_{k+K}\right] \\ c_{-K} \cdot 2ReE\left[r^*_{k+K} r_{k+K-1}\right] + \cdots + c_K \cdot 2ReE\left[r^*_{k-K} r_{k+K-1}\right] & = & 2ReE\left[s_k \cdot r^*_{k+K-1}\right] \\ \vdots & \vdots & \vdots \\ c_{-K} \cdot 2ReE\left[r^*_{k+K} r_{k-K+1}\right] + \cdots + c_K \cdot 2ReE\left[r^*_{k-K} r_{k-K+1}\right] & = & 2ReE\left[s_k \cdot r^*_{k-K+1}\right] \\ c_{-K} \cdot 2ReE\left[r^*_{k+K} r_{k-K}\right] + \cdots + c_K \cdot 2ReE\left[r^*_{k-K} r_{k-K}\right] & = & 2ReE\left[s_k \cdot r^*_{k-K}\right] \end{array} \right\} \tag{4.14}$$

where r_k and s_k are the received sequence and the training sequence at time instant k in the training mode. The optimum equalizer coefficients can be determined from the solution of the set of linear equations. Let A and B represent the matrix of the left-side and right-side of the Equation 4.14, respectively. Thus Equation 4.14 may be expressed in matrix form as

$$AC = B \tag{4.15}$$

where A is the $(2K+1) \times (2K+1)$ matrix of the autocorrelation of the received signal samples $\{r_k\}$, C is the column vector of $(2K+1)$ equalizer coefficients, and B is a $(2K+1)$-dimensional column vector of cross-correlation of the received signal samples and the training sequence samples through the channel. Therefore, we may get C_{opt} by inverting the matrix A

$$C_{opt} = A^{-1}B$$

To avoid the high computational complexity of the matrix inversion, an iterative procedure may be used in real systems. Steepest descent is one method to be used in the iterative procedure. The purpose of the steepest descent method is that each tap weight will change in the direction opposite to its corresponding gradient component expressed as

$$c_j^{(n+1)} = c_j^n - \mu \nabla_{c_j^n} \qquad n = 0, 1, 2, ...$$
$$j = -K, ..., K$$

where the gradient component is

$$\begin{aligned} \nabla_{c_j^n} &= \frac{\partial \Gamma}{\partial c_j} \\ &= \frac{\partial \left[E \left| s_k - q_k \right|^2 \right]}{\partial c_j} \\ &= \frac{\partial [E (s_k - q_k) \cdot (s_k^* - q_k^*)]}{\partial c_j}. \end{aligned} \tag{4.16}$$

From Equation 4.12, the gradient component becomes

$$\begin{aligned} \nabla_{c_j^n} &= E\left[q_k \cdot r_{k-j}^* + q_k^* \cdot r_{k-j} - \left(s_k \cdot r_{k-j}^* + s_k^* \cdot r_{k-j}\right)\right] \\ &= 2Re\left[E\left(q_k \cdot r_{k-j}^* - s_k \cdot r_{k-j}^*\right)\right] \end{aligned}$$

μ is a positive number chosen small enough to ensure the convergence of the iterative procedure.

The second derivative of the mean square error Γ is

$$\begin{aligned} \frac{\partial^2 \Gamma}{\partial c_j^2} &= \frac{\partial \left\{ E\left[q_k \cdot r_{k-j}^* + q_k^* \cdot r_{k-j} - \left(s_k \cdot r_{k-j}^* + s_k^* \cdot r_{k-j}\right)\right] \right\}}{\partial c_j} \\ &= E\left(r_{k-j} \cdot r_{k-j}^* + r_{k-j}^* \cdot r_{k-j}\right) \\ &= E\left|r_{k-j}\right|^2 > 0 \qquad j = -K, ..., K \end{aligned}$$

So we can use steepest descent method since the objective function, i.e., mean square error is convex.

In real systems, the choice of μ should satisfy the inequality

$$0 < \mu < \frac{1}{\lambda_{max}}$$

to ensure the convergence of the recursive procedure, where λ_{max} is the largest eigenvalue of A in Equation 4.15 [2].

In practice, the channel response is unknown, so the cross-correlations given by Equations 4.6 and 4.13 are also unknown. The problem can be solved by transmitting a known training sequence periodically to the receiver, which can be used to estimate the cross-correlation by utilizing time averages instead of the ensemble averages. In many applications, the channel is not only unknown, but also time-varying. Therefore the coefficients of the equalizer should be adaptively adjusted to track the change of the channel. The equalization algorithms described above are adaptive algorithms and suitable for the case of a time-varying channel.

4.2 Simulation results

In this section, we will give some simulation results and analysis of SCCC with MSK over ISI channel. The system model is shown in Figure 4.1. The outer encoder is a recursive systematic code with code rate $R_o = 1/2$ shown in Figure 3.6. The inner encoder is coded minimum shift keying modulation with code rate $R_i = 1/2$. The bandwidth efficiency is 1/2 bps/Hz. The signal sequence output from the inner MSK modulator will go through the ISI channel with 3 tap weights and will be corrupted by additive Gaussian noise. We choose a 3-tap channel model of Stanford University Interim (SUI-5) for broadband wireless applications described in [49]. We calculate the attenuation factors according to the power parameters given in [49] and obtain the tap weights as

$$h = \left[\sqrt{0.7060}\ \sqrt{0.2210}\ \sqrt{0.0729}\right].$$

The total mean power is normalized to 1. We assume that the channel impulse response does not change during a data block. The corrupted signal sequence will enter the equalizer first. Then the estimated output from the equalizer will be passed to the inner and the outer decoder to obtain the estimates of the original information. Zero-forcing equalizer and least mean square (LMS) equalizer are used in our simulations. Iterative log-MAP decoding algorithm is employed as described in Chapter 3. The block length is 1024 and the number of iterations is 8 in all simulations. Random interleaver is used between the outer and the inner encoders. The training sequence will be sent every block and the coefficients of the equalizer will be adjusted accordingly.

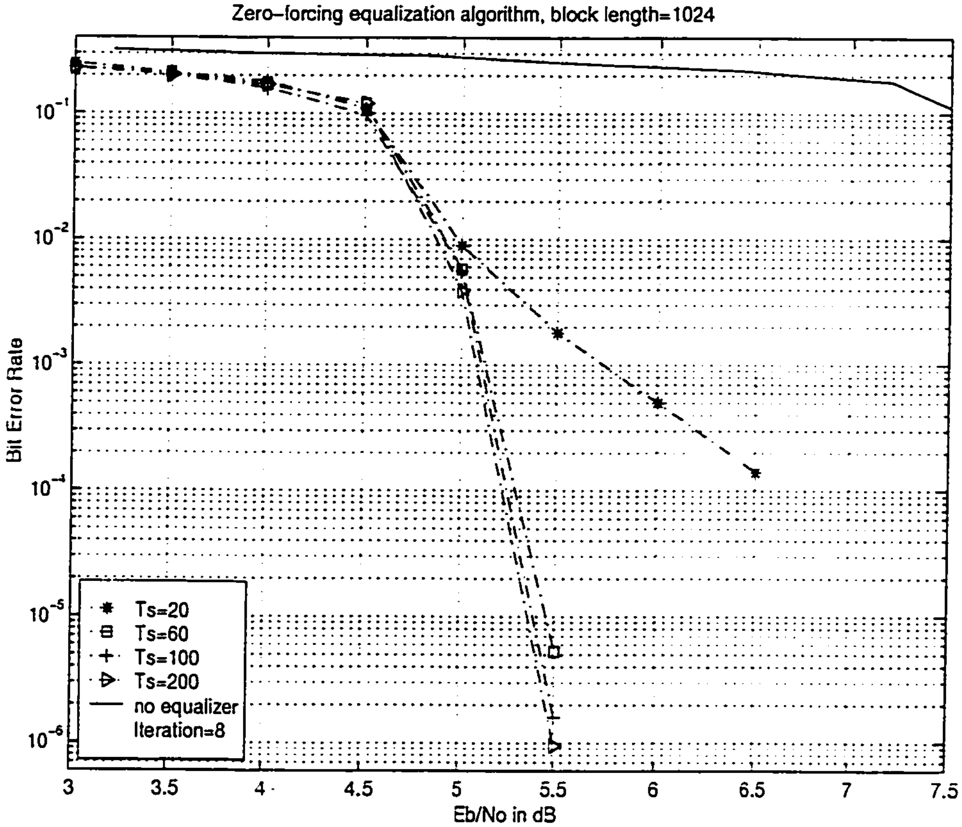


Figure 4.3: Performance of zero-forcing equalization algorithm over ISI channel

Figure 4.3 and 4.4 show the performance using zero-forcing equalization algorithm and least mean square algorithm for different training sequence lengths. The performance improves as the training sequence length increases from 20 to 200 for both algorithms. Since we use time averages instead of ensemble averages, longer length of the training sequence means more samples can be utilized by the receiver to acquire more accurate estimates of the optimal value of the equalizer coefficients. In our simulations, the performance improves about 1.5 dB at BER of 10^{-5} as the training sequence length increases from 20 to 60. However, there is no significant improvement when increasing the length of training sequence over 60, e.g., less than 0.1 dB improvement for the length of 100 training symbols compared to 60 training symbols. Since there are three tap weights in the ISI channel model, the number of the equalizer coefficients we use in our simulations is 3. Only a small margin of improvement can be obtained by excessive samples and it will decrease the data transmission efficiency. Table 4.1 shows the data efficiency under different number of training symbols with the same block length 1024. From our simulations, 60 training symbols is a good compromise between the performance and the data transmission efficiency for both algorithms.

Length of TS	Percentage of TS over block length	Data efficiency
20	1.95%	98.05%
60	5.86%	94.14%
100	9.77%	90.23%
200	19.5%	80.5%

Table 4.1: Data efficiency with different number of the training symbols
TS- Training Symbols

Figure 4.5 shows the performance comparison of zero-forcing algorithm and least mean square algorithm. The solid lines and dashed lines represent the performance of LMS algorithm and zero-forcing algorithm, respectively. For training symbols of

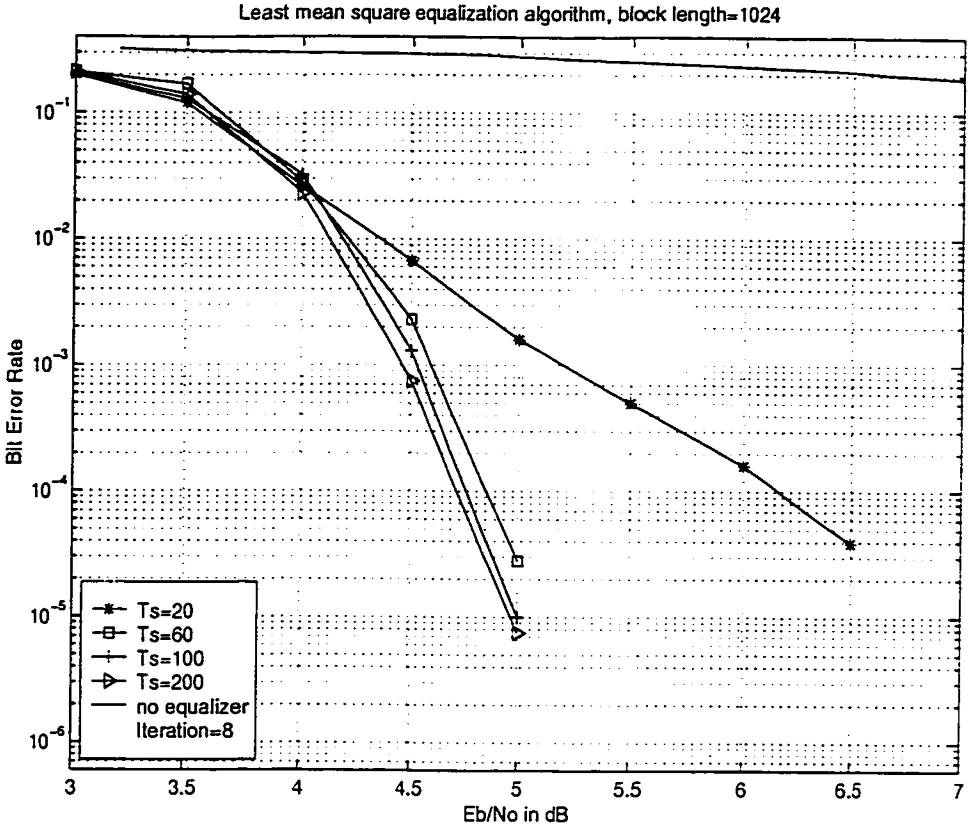


Figure 4.4: Performance of lease mean square equalization algorithm over ISI channel

60, the performance of a system using LMS algorithm is about 0.3 dB better than one with zero-forcing algorithm at BER of 10^{-5} under the same conditions. Since zero-forcing equalizer design does not take into account the effect of additive noise, it will enhance the noise power while eliminating the inter-symbol interference. In contrast, least mean square algorithm based on mean square error criterion overcomes the drawback of zero-forcing algorithm by minimizing the combined power of both the ISI and the additive noise at the equalizer output. Least mean square algorithm has better performance than zero-forcing algorithm.

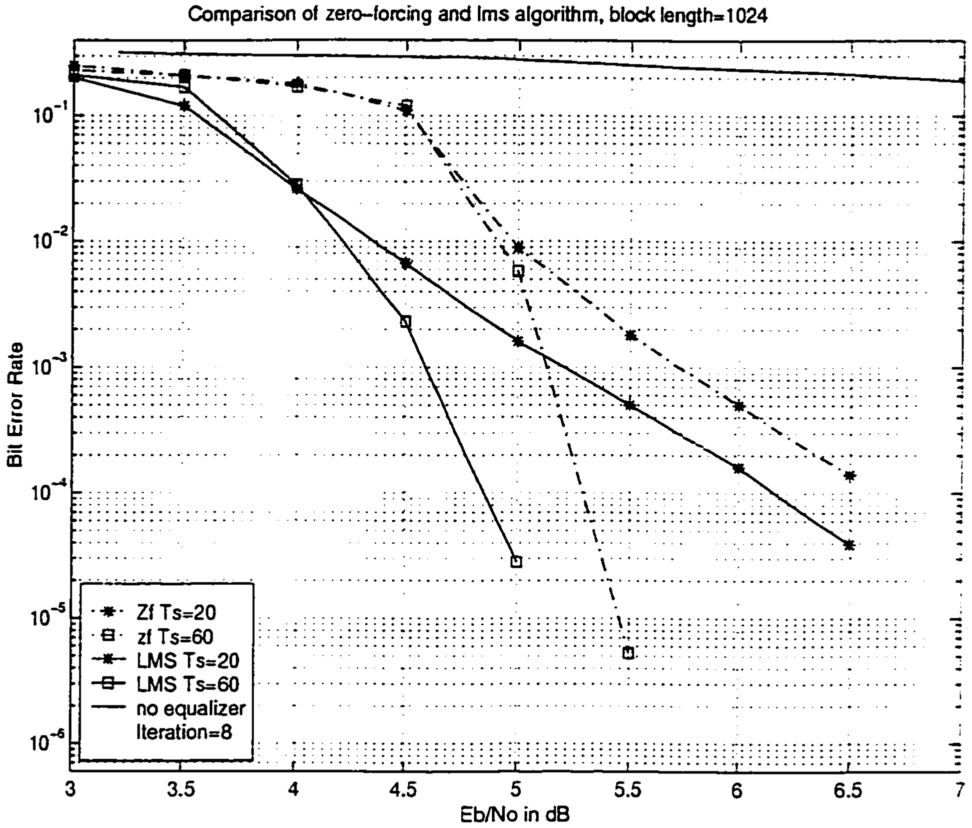


Figure 4.5: Performance comparison of zero-forcing and least mean square algorithm

4.3 Summary

In this chapter, the equalizer design for serial concatenated convolutional code with minimum shift keying modulation over inter-symbol interference channel is investigated. The ISI channel model is important for wireless communication systems since the performance will degrade significantly if there is no compensation for the ISI channel. Equalization has been identified to be an effective method to reduce the influence of the ISI.

The two commonly used criteria, peak distortion criterion and mean square error criterion were discussed. Two adaptive equalization algorithms, zero-forcing algorithm based on the peak distortion criterion and least mean square algorithm based on the mean square error criterion were studied. Simulation results show that (1) the number of training symbols has impact on the performance and the number can be chosen to have a good trade-off between the performance and the data transmission efficiency; (2) least mean square algorithm has better performance than zero-forcing algorithm since both the effects of Gaussian noise and ISI are minimized at the equalizer output.

Chapter 5

Conclusions

In this thesis, the performance of serial concatenated convolutional code with minimum shift keying modulation over AWGN channel and ISI channel have been investigated and analyzed. To reduce the decoding complexity of SCCC with bandwidth efficient M-ary modulation techniques, a converting algorithm which generates soft-decision values of the coded bits is proposed.

The principles of designing SCCC, such as encoder structure, interleaver types, iterative optimal and sub-optimal decoding algorithms have been introduced. In wireless communication systems, minimum shift keying modulation technique has been widely used. Based on these design criteria, combining SCCC with MSK modulation as an inner code over AWGN channel has been identified to be valid and it has comparable performance to the turbo code.

For ISI channel caused by limited bandwidth, multipath propagation and motion, which can distort the transmitted signals severely, it is important to find a feasible way to reduce the effect of ISI. Two popular equalization algorithms, zero-forcing and least mean square algorithms are employed in the linear equalizer at the receiver to do the compensation. Simulation results show that the performance can be improved dramatically by using equalization algorithms, and that the least mean square algorithm results in better performance.

For future work, instead of using equalization, combining the inner encoder and ISI channel to form a super trellis and decoding it as the inner code may be further studied. By using good estimation algorithms of channel state information, better performance may be achieved. In addition, in some applications, such as high data rate bandlimited systems, where training sequence is either costly or impractical, it is desired to use blind equalization algorithms that do not rely on the training sequence. Blind equalization algorithms may be further investigated.

The contributions of this thesis include:

1. To reduce the decoding complexity, we have applied a low complexity converting algorithm to transform the probability distribution of non-binary symbol to the probability distribution of binary one in serial concatenated convolutional code with bandwidth efficient transmission M-ary PSK modulation.

2. For the first time the performance of SCCC with minimum shift keying over intersymbol interference (ISI) channel is investigated and studied. To minimize the BER performance, the optimal decoding algorithm is to combine the inner code and the ISI channel to form a supertrellis and decode it using MAP algorithm. In practice, sometimes it is desirable to sacrifice optimality for computational efficiency. We use equalization algorithm to compensate for the effect of ISI to achieve a good trade-off between complexity and performance.

Bibliography

[1] B. Vucetic and J. Yuan, *Turbo Codes Principles and Applications.* Kluwer Academic Publishers, Boston/Dordrecht/London, 2000.

[2] J. G. Proakis, *Digital Communications.* New York: McGraw-Hill, Inc., Third edition, 1995.

[3] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications,* Englewood Cliffs, NJ: Prentice Hall, Inc., 1983.

[4] C. E. Shannon, *"A mathematical Theory of Communication,"* Bell system technical journal, Vol. 27, pp. 379-423 (Part one), pp. 623-656 (Part Two), Oct. 1948, reprinted in book form, University of Illinois Press, Urbana, 1949.

[5] R. W. Hamming, *"Error detecting and error correcting codes,"* Bell Syst. Tech. J., 29, pp. 147-160, April 1950.

[6] M. J. E. Golay, *"Notes on digital coding,"* Proc. IRE, 37, pp. 657, June 1949.

[7] A. Hocquenghem, *"Codes corecteurs d'erreurs,"* Chiffres, 2, pp. 147-156, 1959.

[8] R. C. Bose and D. K. Ray-Chaudhure, *"On a class of error correcting binary group codes,"* Inf. Control, 3, pp. 68-79, March 1960.

[9] I. S. Reed and G. Solomon, *"Polynomial codes over certain finite fields,"* J. Soc. Ind. Appl. Math., 8, pp. 300-304, June 1960.

[10] P. Elias, *"coding for noisy channels,"* IRE Conv. Rec., Part 4, pp. 37-47, 1955.

[11] G. Ungerboeck, "*Channel coding with multilevel/phase signals*," IEEE Trans. on Information Theory, Vol. IT-28, pp. 55-67, Jan. 1982.

[12] C. Berrou, A. Glavieux and P. Thitimajshima, "*Near Shannon limit error-correcting coding and decoding: Turbo-codes*," ICC'93, Geneva, Switzerland, pp. 1064-1070, May 93.

[13] C. Berrou, A. Glavieux and P. Thitimajshima, "*Near Optimum error-correcting coding and decoding: Turbo-codes*," IEEE Trans. on Comm., Vol. 44, No.10, pp. 1261-1271, October 1996.

[14] D. J. C.Mackay and R. M. Neal, "*Near shannon limit performance of low density parity check codes*," Electron. Lett., Vol. 33, pp. 457-458, Mar. 1997.

[15] C. D. Edwards, C. T. Steilzvied, L. J. Deutsch and L. Swanson, "*NASA's deep-space telecommunications road MAP*," TMO Progress Report 42-126, Feb. 1999.

[16] 3GPP2 C.S0024, "*cdma2000 High Data Packet Data Air Interface Specification*," Version 2.2, Aug. 23, 2001.

[17] R. Pyndiah, A. Glavieux, A. Picart and S. Jacq, "*Near optimum decoding of products codes*," GLOBECOM'94, San Francisco, CA, pp. 339-343, Dec. 1994.

[18] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "*Serial Concatenation of Interleaved Codes: Performance Analysis. Design and Iterative Decoding*," IEEE Trans. on Information Theory, Vol. 44, No.3, May 1998.

[19] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.

[20] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "*A soft-input soft-output Maximum A Posterior (MAP) module to decode Parallel and serial concatenated codes,*" TDA Progress Report 42-127, pp. 1-20, Nov. 15, 1996.

[21] R. Garelle, P. Pierleoni and S. Benedetto, "*Computing the free distance of Turbo codes and serially concatenated codes with interleavers: algorithms and applications,*" IEEE JSAC in Comm. Vol. 19, No.5, pp. 800-812, May 2001.

[22] D. Divsalar and F. Pollara, "*Turbo codes for PCS applications,*" in Proc. ICC'95, Seattle, WA, pp. 54-59, June 1995.

[23] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "*Optimal decoding of linear codes for minimizing symbol error rate,*" IEEE Trans. on Information Theory, Vol. IT-20, pp. 284-297, Mar. 1974.

[24] J. Hagenauer, E. Offer and L. Papke, "*Iterative decoding of binary block and convolutional codes,*" IEEE Trans. on Inform.Theory, Vol. 42, No. 2, pp. 429-445, Mar. 1996.

[25] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "*A soft-input soft-output APP module for iterative decoding of concatenated codes,*" IEEE Comm. Letter, Vol. 1, No. 1, pp. 22-24, Jan. 1997.

[26] P. Robertson and P. Hoeher, "*Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding,*" European Transactions on Telecomm., pp. 119-125, March - April 1997.

[27] T. S. Rappaport, *Wireless Communication Principle and Practice*, Prentice Hall Inc., 1996.

[28] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "*Serial concatenated trellis coded modulation with iterative decoding,*" Proc. Int. Symp. on Inf. Theory (ISIT'97), Ulm, Germay, pp. 8, June-July 1997.

[29] P. Robertson and T. Worz, "*Bandwidth-Efficient Turbo Trellis-Coded Modulation Using Punctured Component Codes,*" Electron.Lett., Vol. 31, pp. 1546-1547, Aug. 1995.

[30] M. R. Shane and R. D. Wesel, "*Parallel concatenated turbo codes for continuous phase modulation,*" IEEE Wireless Comm. and Networking Conference, WCNC, Vol. 1, pp. 147-152, 2000.

[31] J. B. Anderson, T. Aulin and C.-E. W. Sundberg, *Digital Phase Modulation*, Plenum Press, New York, NY, USA, 1986.

[32] T. Aulin and C.-E.W. Sundberg, "*Continuous phase modulation - Part I: Full response signalling,*" IEEE Trans. Comm., Vol. COM-29, pp. 196 - 209, March 1981.

[33] T. Aulin, N. Rydbeck and C.-E. W. Sundberg, "*Continuous phase modulation - Part II: Partial response signaling,*" IEEE Trans. Comm., Vol. COM-29, pp. 210 -225, March 1981.

[34] B. Rimoldi, "*A decomposition approach to CPM,*" IEEE Trans. Inf. Theory, Vol. IT-34, pp. 260 -270, March 1988.

[35] P. Moqvist and T. M. Aulin, "*Concatenated continuous phase modulation with iterative decoding,*" IEEE Trans on Comm., Vol. 49, No.11, pp. 1901-1915, November 2001.

[36] H. Leib and S. Pasupathy, "*Error-control properties of minimum shift keying,*" IEEE Comm. Magazine, pp. 52-61. January 1993.

[37] J. H. Lodge, M. L. Moher and S. N. Crozier, "*A comparison of data modulation techniques for land mobile satellite channels.*" IEEE Trans, Veh. Tech.. Vol. VT-36, No.1, pp. 28-34, Feb. 1987.

[38] C. R. Ryan, A. R. Hambley and D. E. Vogt, "*760 Mbit/sec serial MSK microwave modem,*" IEEE Trans. Comm., Vol. COM-28, No 5, pp. 771-777, May 1980.

[39] R. E. Ziemer and C. R. Ryan, "*Minimum-shift keyed modem implementation for high data rates,*" IEEE Comm. Mag., Vol. 21, No. 7, pp. 28-37, Oct. 1983.

[40] D. J. Goodman, "*Trends in cellular and cordless communications,*" IEEE Comm. Mag., pp. 31-40, June 1991.

[41] C. Laot, A. Glavieux and J. Labat, "*Turbo equalization:adaptive equalization and channel decoding jointly optimized,*" IEEE Trans. on JSAC, Vol. 19, No. 9, pp. 1744-1752, September 2001.

[42] M. S. Akhter, "*Performance of Channel Interleaved Turbo Code System for High Rate Mobile Satellite Communications,*" Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on, Volume: 2, 2000, pp. 908-912.

[43] R. M. Joyce, L. J. Ibbetson and L. B. Lopes, "*Prediction of GSM performance using measured propagation data,*" IEEE 46th on VTC, Mobile technology for the Human Race, Vol. 1, pp. 326-330, 1996.

[44] A. O. Berthet, B. S. Unal and R. Visoz, "*Iterative Decoding of Convolutionally Encoded Signals Over Multipath Rayleigh Fading Channels,*" IEEE JSAC in Comm., Vol. 19, No.9, pp. 1729-1743, Sept. 2001.

[45] S. Haykin, "*Adaptive Filter Theory,*" Prentice Hall, Englewood Cliffs, NJ, 1986.

[46] EIA/TIA Interim Standard, *"Cellular system dual mode mobile station- land station compatibility specification,"* IS-54, Electronic Industries Association, May 1990.

[47] Z. Ding and Y. Li, *"Blind Equalization and Identification"*, Marcel Dekker, Inc. New York. Basel.

[48] B. J. Kim and D. C. Cox, *"Blind equalization for short burst wireless communications,"* IEEE Trans. on Vehicular Technology, Vol. 49, No. 4, pp. 1235-1247, July 2000.

[49] IEEE 802.16.3c-01/29r4, "*Channel models for fixed wireless applications,*" July 17, 2001.