THE PERFORMANCE OF TURBO TRELLIS CODED 8-PSK MODULATION ON FREQUENCY SELECTIVE RAYLEIGH FADING CHANNEL

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of
Dokuz Eylül University
In Partial Fulfillment of the Requirements for
the Degree of Master of Science in Electrical and Electronics Engineering

by

Mümtaz YILMAZ

July, 2002
İZMİR

Ms.Sc. THESIS EXAMINATION RESULT FORM

We certify that we have read the thesis, entitled **"The performance of Turbo Trellis coded 8 PSK modulation on frequency selective Rayleigh fading channel"** completed by **Mümtaz YILMAZ** under supervision of Asst.Prof.Dr. Reyat YILMAZ and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst.Prof.Dr. Reyat YILMAZ

Supervisor

Prof.Dr. Kemal ÖZMEHMET

Committee Member

Yrd. Doç. Dr. Mustafa A. ALTINKAYA

Committee Member

Approved by the

Graduate School of Natural and Applied Sciences

Prof.Dr.Cahit Helvacı

Director

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor Asst.Prof.Dr. Reyat YILMAZ for his continuous support, technical guidance and encouragement throughout this research. He had the kindness to accept me as a research assistant when I started master of science study at Dokuz Eylul University. Also, I would like to thank my family, especially my mother, for their tolerance and never ending support throughout my life.

ABSTRACT

Coded-modulation techniques combine coding and modulation in one process in order to provide both error correcting and spectral efficiency.The first approach in applying turbo codes with multilevel modulation is the natural combination of turbo codes with multilevel modulation and is called the pragmatic approach. Another one is the extension of binary turbo codes by replacing component recursive systematic convolutional codes with recursive systematic trellis coded-modulation (TCM) codes. This approach is referred to as turbo TCM (T-TCM). It takes the advantages of both near Shannon limit low error rate performance of turbo codes and spectral efficiency of TCM.

In this thesis T-TCM technique was investigated together with two decoding algorithm methods, namely maximum a posteriori probability (MAP) and MAP equivalent soft output Viterbi algorithm (SOVA). Ungerboeck type, 8 state, rate 2/3 code and 8 phase shift keying modulation is used in the encoder. The system performance is investigated in both flat and frequency selective Rayleigh fading channels.Decision feedback equalization technique is used for reducing the effects of frequency-selective Rayleigh fading channel. It is shown that T-TCM gives a better performance in both flat and frequency selective Rayleigh fading channels compared to TCM with the same component code.

ÖZET

Kodlanmış modülasyon teknikleri, kodlama ve modülasyon işlemlerini, hata düzeltme ve spektral verimliliği aynı anda sağlayacak şekilde, tek bir işlemde birleştirir. Turbo kodları çok seviyeli modülasyon teknikleri ile birleştirmeyi amaçlayan ilk yaklaşım, turbo kodların direk olarak çok seviyeli modülasyon teknikleri ile birleşimidir ve pragmatik yaklaşım olarak adlandırılır. Bir diğeri ise ikili turbo kodlayıcıdaki, yinelemeli sistematik konvolüsyon kod bileşeninin, yinelemeli sistematik kafes kodlanmış modülasyon (KKM) kodlarıyla değiştirilmesidir. Bu yaklaşım turbo KKM (T-KKM) olarak tanımlanır. Bu sistem, turbo kodların Shannon limitine yakın düşük hata oranı performansı ile KKM sisteminin spektral verimlilik avantajlarına sahiptir.

Bu tezde, T-KKM tekniği iki farklı kod çözme algoritması, maksimum sonsal olasılık (MSO) ve MSO dengi yumuşak çıkış Viterbi algoritması kullanılarak incelenmiştir. Kodlayıcıda Ungerboeck tipi, 8 durumlu, 2/3 oranlı kod ve 8 evre kaydırmalı modülasyon kullanılmıştır. Sistem performansı düz ve frekans seçici Rayleigh sönümlü kanalların her ikisinde birden incelenmiştir. Frekans seçici Rayleigh sönümlü kanalın sinyaller üzerindeki etkisini azaltmak için karar geri beslemeli denkleştirme tekniği kullanılmıştır. T-KKM sisteminin düz ve frekans seçici Rayleigh sönümlü kanallarda aynı bileşen kodlayıcıya sahip KKM sisteminden daha iyi bir performans sağladığı görülmüştür.

CONTENTS

Page

Page

Chapter Four

TURBO TRELLIS CODED MODULATION

Chapter Five

MULTIPATH FADING CHANNELS AND EQUALIZERS

Page

Chapter Six
THE PERFORMANCE OF T-TCM IN A MOBILE RADIO CHANNEL

Chapter Seven
CONCLUSIONS

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

In his work on the subject of information theory, Shannon showed that there exist error correcting codes that can achieve arbitrarily small bit error rate, as long as the signal to noise ratio used is above the capacity of the channel (Shannon,1948). Unfortunately, Shannon's findings gave no indication of the structure of the codes that would give performance near channel capacity. Since this time, coding theorists have attempted to find error correcting codes that will provide performance near capacity. The papers by (Hamming,1950) and (Golay,1949) are the first important works after Shannon. These works are rapidly followed with papers about new codes (Muller,1954) and (Reed,1954). The RS codes by (Reed&Solomon,1960) and BCH codes by (BoseRay-Chaudhuri,1960) are still used widely in many applications. Two major techniques for building codes have been the primary focus of study. These techniques are block (or algebraic) codes and convolutional (or trellis) codes. Although in theory, channel capacity can be approached by these codes, the block size of the block code or memory order of the trellis code must be made very large for channel capacity to be approached. This increase in size necessitates an increase in decoding complexity due to the increased block size or increased number of states.

Trellis coded-modulation (TCM) which was first introduced by (Ungerboeck, 1982) is a means of achieving a coding gain on both bandwidth-constrained and power-constrained channels. This can be achieved by increasing the number of signal points in the constellation over the corresponding uncoded system and designing the trellis codes so that the Euclidean distance in a sequence of transmitted symbol corresponding to paths that merge at any node in the trellis is larger than the Euclidean distance per symbol in an uncoded system. The original work on TCM

was for additive white Gaussian noise (AWGN) channel, further results on TCM for fading channels were developed by (Divsalar & Simon, 1987, 1988a, 1988b).

Recently, the turbo codes introduced by (Berrou et al., 1993) have excited the coding community with the promise of performing near channel capacity. Turbo codes are binary error-correcting codes built from the parallel concatenation of two recursive systematic convolutional (RSC) codes which use an iterative decoding technique. The invention of turbo codes is based on AWGN channel and further investigation of turbo codes for fading channels is necessary as the turbo coding scheme is a relatively new technique. This technique uses an iterative decoding algorithm to achieve performance near capacity with a relatively simple decoder. However, the turbo encoder is typically a very low rate (R=1/2) code which means that the bandwidth of systems using a turbo code will suffer from bandwidth expansion. In order to improve the transmission spectral efficiency of mobile channels, it is interesting to combine turbo codes with a bandwidth-efficient modulation.

Different approaches for turbo-coded modulation systems have been proposed until now. The least complicated is the one suggested in (Goff et.al.,1994). It's based on mapping the encoded bits of a standart turbo code (possibly after puncturing some of the parity bits to obtain a desired spectral efficiency) to a certain constellation. The decoding is performed by first computing the log-likelihood ratios of the transmitted systematic and parity bits, and then using a standard turbo decoding algorithm. Another approach proposed by (Robertson,1996), is straight forward extension of Ungerboeck type TCM to parallel concatenated coding. The decoding algorithm is a modified version of the decoding algorithm used for decoding of the binary turbo codes. In (Benedetto et.al.,1996), the inputs of the convolutional codes are the entire information block and its interleaved version. Half of the information bits are punctured at the output of the first component code, and the other half are punctured at the output of the second component code. Decoding is done by using the MAP algorithm extended to symbols (Benedetto et.al.,1996).

Mobile radio channels exhibit time-varying multipath propagation. A pulse transmitted over one of these channels is received as a train of delayed pulses of different amplitudes. When these multipath signals which arrive via many paths vectorially sum to a small value, the signal is said to be in a deep fade. Such a channel is referred to as a *fading multipath channel*. A more serious problem arises when data is transmitted at high speed. The fading multipath channel causes not only the signal attenuation but also the severe time dispersion of the adjacent pulses. This time-dispersion phenomenon is referred to as intersymbol interference (ISI) and the channel in this situation is called *multipath fading ISI channel* or *frequency-selective fading channel*. Additional ISI could also be caused by transmitting the data over bandlimited channels. Therefore for reliable communication over multipath fading ISI channel, it is essential to eliminate ISI. Conventional ways to reduce ISI include:

a) the zero-forcing filter which eliminates ISI completely
b) equalizers which make an optimum estimation under various criteria

The use of channel coding together with an equalizer to combat multipath fading ISI channel is presented in block diagram of Fig. 1.1

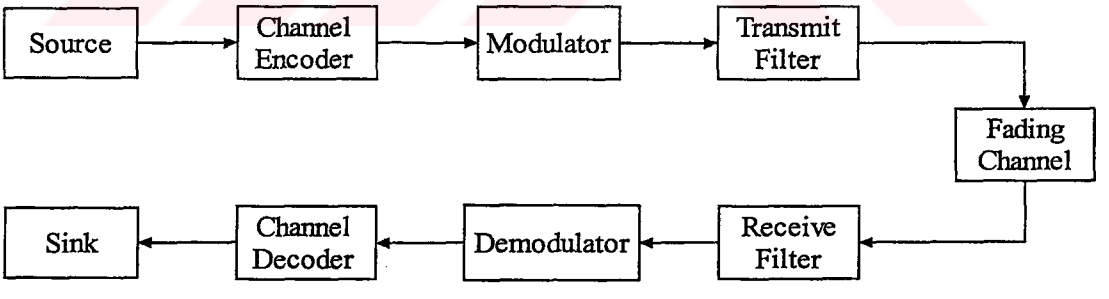


Figure 1.1. Digital Communication over multipath fading ISI channel

In all of the works about T-TCM, the system performance was examined in AWGN channels. In addition to AWGN present in the channel, there exists fading effects, especially in mobile radio channels. A system designed for AWGN channel may not perform well in fading channels like flat or frequency selective fading channels. So it's worthwhile to examine the performance of T-TCM systems for

different environments. Also the complexity of the MAP decoding algorithm, used in the previous works, introduced the need for less complex decoding algorithms with desirable performance degradation or with an increase in performance. The use of soft output Viterbi algorithm (SOVA) for non-binary codes may be a solution. In this work the performance of T-TCM system on flat and frequency selective fading channels is investigated with two decoding algorithms, MAP and MAP equivalent SOVA.

1.1 Outline of thesis

In Chapter 2, the structures of convolutional codes and types of convolutional codes are explained first, and then the basic principles of trellis coded modulation are described. The basic parameters of a TCM system are explained.

Chapter 3 is devoted to fundamentals of turbo codes. Turbo encoding process is explained together with subjects interleaving and trellis termination. Iterative turbo decoding is described by means of a posteriori probability algorithms. Chapter 4 investigates bandwidth efficient turbo coded systems. Turbo trellis coded modulation (T-TCM) which is the main subject of this thesis is introduced. The encoding and decoding mechanisms of T-TCM is given. Various decoding techniques such as MAP decoder and a MAP equivalent SOVA for non-binary Turbo codes is described.

In chapter 5 the characterization of a general multipath fading channel is described and a discrete model is presented. Simulation of Rayleigh fading concept is explained and equalization techniques to combat the effects of frequency selctive fading channels is investigated. In Chapter 6 the model of the system that is used in simulations for finding the performance of (T-TCM) on frequency selective Rayleigh fading channels is described. The computer simulation results are presented.

Finally, Chapter 7 gives conclusions and recommendations for further study.

CHAPTER TWO
TRELLIS CODED MODULATION

2.1 Convolutional Codes:

Convolutional codes are implemented using a linear shift register network and provide continuous encoding of input streams. Consider a rate k/n convolutional code,where k and n are the number of input and output streams of the encoder.The information stream, m, is split into k streams $m^{(p)} = \{m_p, m_{p+k}, m_{p+2k}, ...\}$, where $p \in \{0,...,k-1\}$. Each of these k streams are passed through the convolutional encoder, which produces n output streams, $x^{(q)}$, where $q \in \{0,...,n-1\}$. These n output streams are multiplexed to form the codeword $x = \{x_0^{(0)},...,x_0^{(n-1)},x_1^{(0)},...,x_1^{(n-1)},...\}$. There are, therefore, k shift registers in the encoder, one for each input stream. Let M_p be the length of the shift register associated with the input p. Then, the total memory of the encoder is given by

$$M_c = \sum_{p=0}^{k-1} M_p \tag{2.1}$$

The constraint length K_c of the encoder is most commonly defined to be the maximum number of bits in a single output stream that can be affected by any input bit

$$K_c = 1 + \max_p M_p . \tag{2.2}$$

The encoding process can be described as the convolution of the input with the coefficients of the generator polynomial (tap weights of the shift register), i.e.,

$$v_i^{(p,q)} = \sum_{j=0}^{M_p} m_{i-j}^{(p)} g_j^{(p,q)}, \quad i \in \{0,...,N_b - 1\} \tag{2.3}$$

where $g_j^{(p,q)}$ is the j -th coefficient of the generator polynomial associated with input p and output q ,and N_b is the length of the input sequence. The output stream $x^{(q)}$ is then formed by

$$x^{(q)} = \sum_{p=0}^{k-1} v^{(p,q)} \tag{2.4}$$

An example rate $1/2$ encoder with generator polynomials $g^{(0)} = 1 + D + D^2$ and $g^{(1)} = 1 + D^2$ is shown in Fig. 2.1 For brevity, the coefficients of each generator polynomial are often concatenated into a single binary word, which is then represented as an octal number, which is (7,5) for the encoder shown in Fig. 2.1.

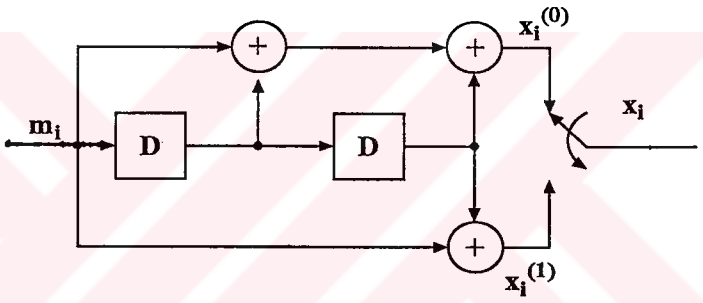


Figure 2.1 $r = 1/2$ convolutional encoder with constraint length $K_c = 3$.

A recursive systematic convolutional (RSC) code can be obtained from the convolutional (non-recursive, non-systematic) code by using a feedback loop, and by setting one (or more) of the output streams equal to one (or more) input streams. For instance, consider a rate 1/n convolutional code with generator matrix (for a rate 1 /n code, p equals 0, and is discarded from all subsequent notations)

$$g(D) = \left[g^{(0)}(D) \quad g^{(1)}(D) \quad \ldots \quad g^{(n-1)}(D) \right] \tag{2.5}$$

To obtain the equivalent RSC code, we compute the remainder polynomial

$$r(D) = \frac{m(D)}{g^{(0)}(D)} \tag{2.6}$$

where m(D) is the information or message polynomial. Consequently, the parity outputs are given by

$$x^{(q)}(D) = r(D)g^{(q)}(D) \qquad q \in \{1,...,n-1\} \tag{2.7}$$

and the systematic output is simply

$$x^{(0)}(D) = m(D). \tag{2.8}$$

Hence, the generator matrix for the RSC code can be written as

$$g_{RSC}(D) = \left[1 \quad \frac{g^{(1)}(D)}{g^{(0)}(D)} \quad \cdots \quad \frac{g^{(n-1)}(D)}{g^{(0)}(D)}\right] \tag{2.9}$$

and the RSC encoding process can be described by

$$r_i = m_i + \sum_{j=1}^{M_c} r_{i-j} g_j^{(0)} \qquad i \in \{0,...,N_b - 1\} \tag{2.10}$$

where r_i is the feedback variable, and the parity outputs $x_i^{(q)}$ are given by

$$x_i^q = \sum_{j=0}^{M_c} r_{i-j} g_j^{(q)} \qquad i \in \{0,...,N_b - 1\}. \tag{2.11}$$

2.2 TCM Principle

In classical digital communication systems, coding and modulation are two separate processes. The modulators and demodulators convert an analog waveform channel into a discrete channel whereas encoders and decoders correct the errors that occur in the discrete channel. When the code rate R is equal to k/n, (n-k) redundant check bits are appended to every k information bits to generate n coded bits. The redundancy allows the receiver to detect and correct errors caused by noise in the channel. However, for the coded system to retain the same information transmission rate as the uncoded system, it is required that the symbol transmission rate R_s (and thus the Nyquist bandwidth) for the coded system must be increased (R^{-1}) times than that of the uncoded system. In a bandwidth limited system this presents a problem since the additional bandwidth required for conventional error control is not

available. One alternative is to use multilevel or multiphase modulation. In this case the use of very powerful codes (large constraint-length convolutional codes or large block-length block codes) are required to overcome the penalty due to increase of power and provide some significant coding gain.

Trellis coded-modulation (TCM) introduced by (Ungerboeck,1982) is a powerful additional alternative. In his paper Ungerboeck treated the modulation as an integral part of the encoding process and designed it in conjunction with the code to increase the minimum Euclidean distance between pairs of coded signals. Then it is showed that the loss of power from the expansion of the signal set is easily overcome and a significant coding gain is achieved with relative simple codes.

The key to the integrated modulation and coding approach is to effectively map the encoded bits into the signal points such that the minimum Euclidean distance is maximized. The method developed by Ungerboeck is based on the principle of *mapping by set partitioning.* The purpose of this partitioning is that each partition should produce subconstellations with increased minimum distance. More detailed explanation on this can be found in (Ungerboeck,1982,1987a,1987b). As an example, the partitioning applied to M-ary Phase Shift Keying (MPSK) modulation with M=8 is reproduced in Fig. 2.2. The general structure of encoder/modulator for trellis coded-modulation is shown in Fig. 2.3.

Out of (m-1) information bits, *k* bits are encoded using rate k/(k+1) convolutional encoder, the resulting (k+1) bits are used to select one of the 2^{k+1} partitions of the 2^m_ary signal constellation. The remaining (m-k-1) uncoded information bits are then used to select a signal point within a particular partition. The name trellis coded-modulation is due to the trellis structure of the convolutional encoder which produces the allowed signal sequence at the output of Ungerboeck encoder.

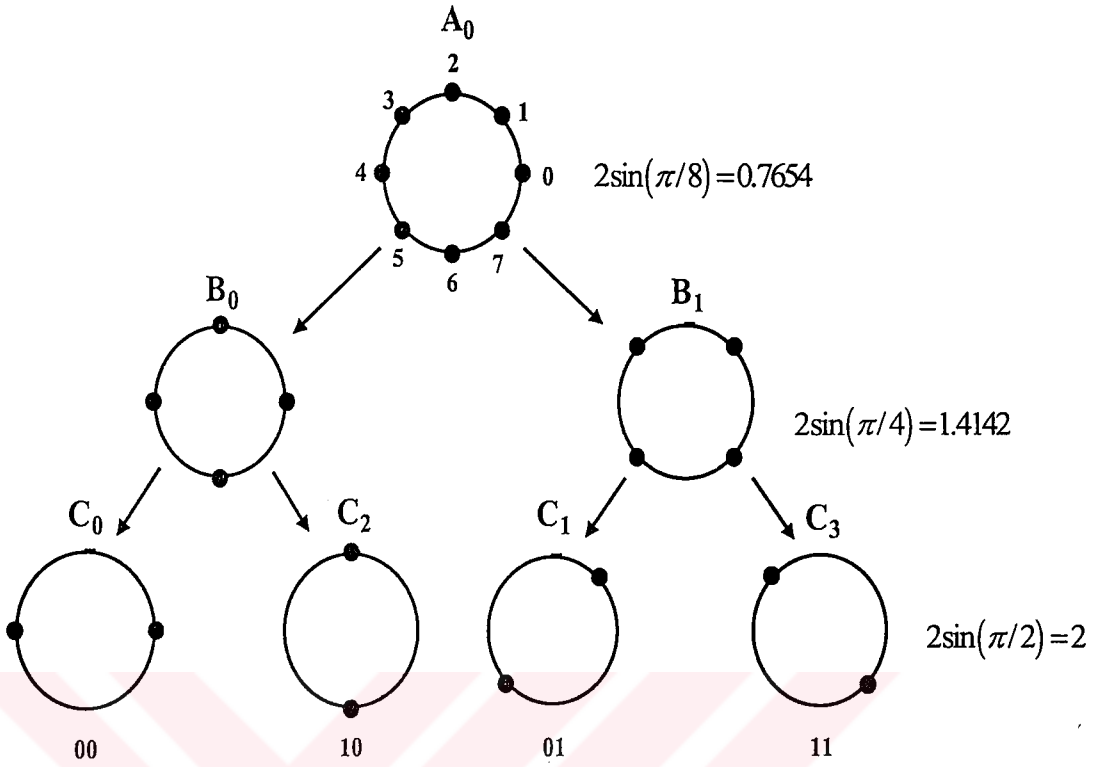


Figure 2.2. Partitioning of 8-PSK channel signals into subsets with increasing minimum subset distances (Ungerboeck,1987a)

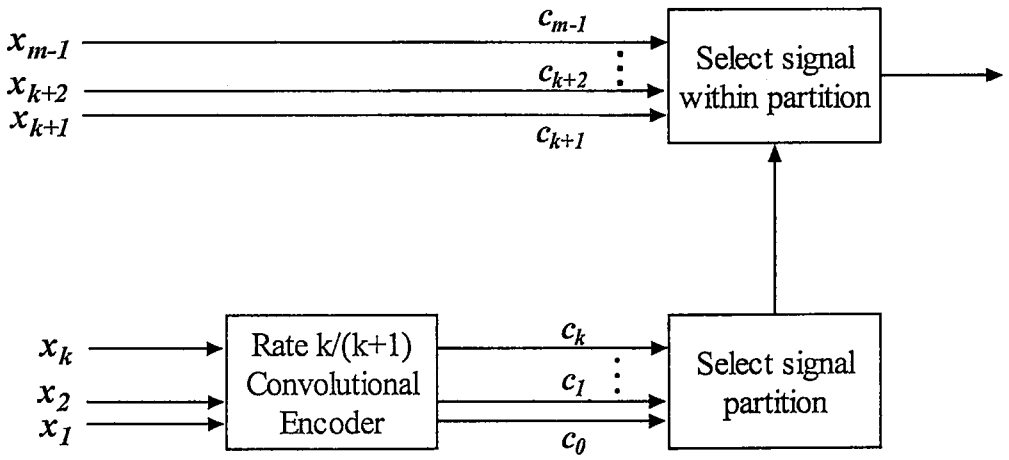


Figure 2.3 Structure of an Ungerboeck encoder (Wicker,1995)

It is well-known that the probability of symbol error for transmission over an AWGN channel is a function of the minimum Euclidean distance between a pair of distinct signal sequences called the minimum free distance. In the uncoded case two valid sequences of symbols may differ in only one symbol and that symbol can take any value in the signal alphabet. As a consequence d_{free} in an uncoded system is limited to the minimum Euclidean distance between any pair of the points in the signal constellation. However in the coded case, due to the trellis structure of the encoder the symbol sequences can be limited to certain patterns so as to make d_{free} larger than the minimum Euclidean distance in the uncoded case.

In general the performance improvement provided by the coded system relative to the uncoded system is measured in term of the *asymptotic coding gain*:

$$\gamma = \frac{S_u / d^2_{free/u}}{S_c / d^2_{free/c}} = \left(\frac{S_u}{S_c}\right) \times \left(\frac{d^2_{free/u}}{d^2_{free/c}}\right) = \gamma_C \times \gamma_D \qquad (2.12)$$

where S_u and S_c are the normalized average received energy for the uncoded and coded case and $d_{free/u}$, $d_{free/c}$ is the free distance of uncoded and coded case respectively. γ_C is called constellation expansion factor representing the additional energy due to expanding signal constellation, γ_D is the increased distance factor representing the increased minimum distance when using TCM (Wicker,1995). Ungerboeck design rules for constructing TCM schemes are summarized below (Ungerboeck,1987a,1987b).

1. Parallel transitions are associated with signals with maximum distance
2. Signals in the preceding partition are assigned to transitions that start or stop in the same state.
3. All signals are used equally often.

In general, the asymptotic coding gain of TCM can be increased by increasing the memory and the rate of convolutional encoder. However the complexity of the trellis

(and thus the decoder) is also increased. Finally it should be noted that the above design rules of TCM are only applicable for AWGN channel. Figures 2.4.(a) and 2.5.(a) present encoders for two Ungerboeck TCM schemes, one is 4-state 8-PSK and the other is 8-state 8-PSK, these TCM schemes are optimized for AWGN channel. The trellis diagram of these encoders are shown in Fig. 2.4.(b) and 2.5.(b) respectively. For 4-state 8-PSK TCM, d_{free} is equal to 4 (assuming normalized energy of 1) corresponding to parallel transitions. The coding gain according to (2.12) is $\gamma = 2 = (3.01 \text{ dB})$

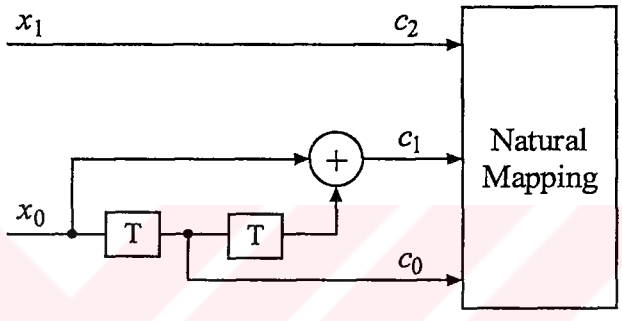


Figure 2.4.(a) 4-state 8-PSK Ungerboeck encoder

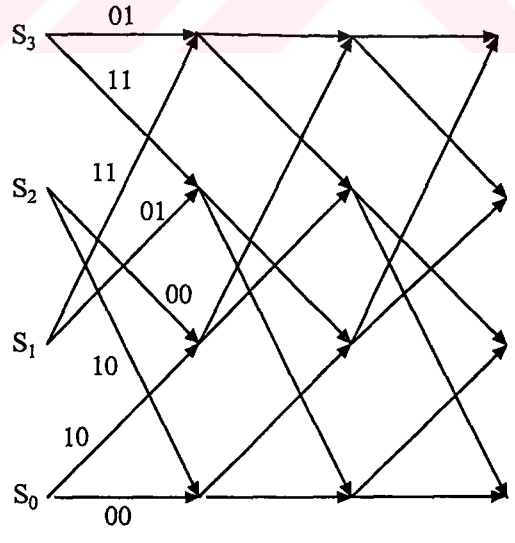


Figure 2.4.(b) Trellis diagram of 4-state 8-PSK Ungerboeck encoder

For 8-state 8-PSK d_{free} is equal to 4.586 (no parallel transitions) and the coding gain is $\gamma = 2.29 = (3.60\ \text{dB})$.

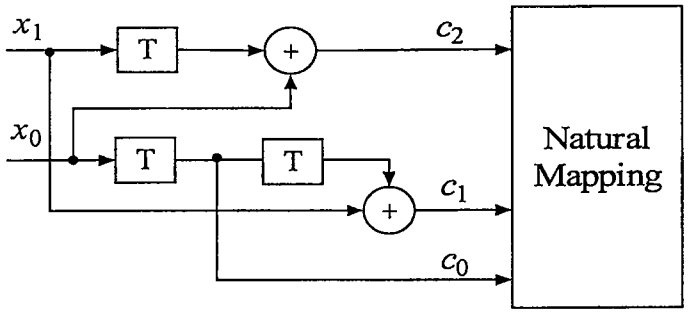


Figure 2.5.(a) 8-state 8-PSK Ungerboeck encoder

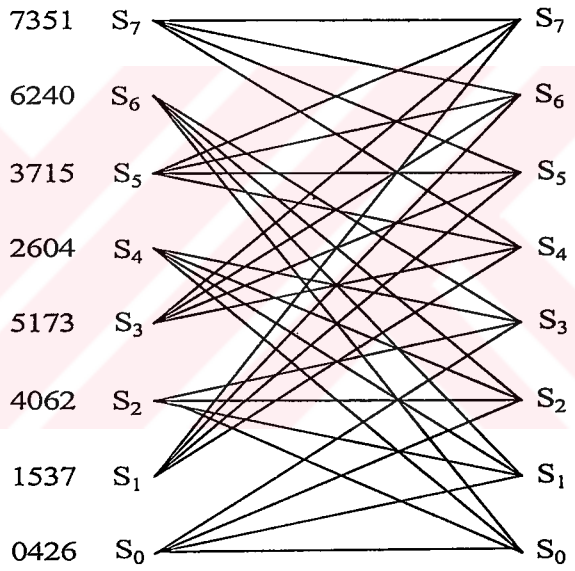


Figure 2.5.(b) Trellis diagram of 8-state 8-PSK Ungerboeck encoder

A further development of TCM technique was made by (Divsalar&Simon,1988a) where the authors demonstrated a trellis coded-modulation technique referred to as multiple trellis coded-modulation (MTCM). It was shown that MTCM can achieve a significant increase in coding gain relative to conventional TCM by increasing the number of signals transmitted per each trellis branch. However, additional computations per branch are needed for MTCM which further increases the complexity of MTCM scheme over uncoded system.

CHAPTER THREE
FUNDAMENTALS OF TURBO CODES

A well known result from information theory is that a randomly chosen code of sufficiently large block length n is capable of approaching channel capacity (Shannon,1948). The maximum-likelihood (ML) decoding of such a code increases exponentially with n up to a point where decoding becomes physically unrealizable. The goal of coding theorists has been to develop codes that have large equivalent block lengths, yet contain enough structure that practical decoding is possible. Recently, Turbo coding, introduced by (Berrou et al.,1993) is a coding technique that uses an iterative decoding algorithm to give BER performance near channel capacity. Turbo codes is one of the most exciting and important development in coding research in recent years, which has given rise to a large interest in the coding community. Turbo codes achieve near-Shannon-limit error performance with relative simple component codes and a large interleaver. Bit error rate as low as 10^{-5} at Eb/N0 = 0.7dB for a rate 1/2, and as low as 10^{-6} at Eb/N0 = -0.6dB for a rate 1/15 turbo codes over AWGN channel have been shown in (Berrou et al.,1993) and (Divsalar&Pollara,1995b) respectively.

Turbo codes are parallel concatenated convolutional codes which have demonstrated near-capacity performance through the use of simple constituent encoders and an iterative, soft-decoding algorithm. The true power of these codes is in their ability to create very powerful code structures while retaining the ability to perform soft-decoding without dramatically increased complexity.

3.1 Turbo Encoding

The classical turbo encoding scheme proposed by (Berrou et al, 1993) consists of a parallel concatenation of two recursive systematic convolutional (RSC) codes. Figure 3.1 shows the block diagram of such a turbo encoder. The two rate 1/2 RSC encoders, C_1 and C_2 , are separated by an interleaver π . An information symbol m_i is fed directly into C_1. Its interleaved (permuted) version, $\overline{m}_i$, is fed into C_2. The systematic output of the turbo encoder, $x_i^{(0)}$, is taken from C_1, while the two parity outputs, $x_i^{(1)}$ and $x_i^{(2)}$, are taken from C_1 and C_2. These three outputs are multiplexed to form the codeword $x = \left\{x_i^{(0)}, x_i^{(1)}, x_i^{(2)}\right\}$. Hence, the overall code rate of the turbo encoder is 1/3. However, other code rates can be derived by using an appropriate puncturing scheme. For instance, a rate 1/3 turbo code can be transformed into a rate 1/2 turbo code by using the puncturing matrix

$$P_M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.1}$$

where the two parity outputs are alternately punctured.

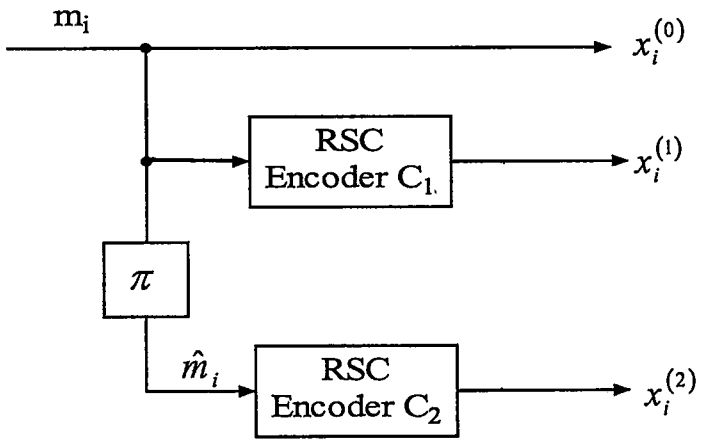


Figure 3.1 :A rate 1 /3 turbo encoder.

3.1.1 Interleaving

Typically, the output codewords of a RSC encoder have a high Hamming weight. However, some input sequences can cause the RSC encoder to produce low weight codewords. The combination of interleaving (permuting) and RSC encoding ensures that the codewords produced by a turbo encoder have a high Hamming weight. For instance,assume that the RSC encoder C_1 receives an input sequence which causes it to generate a low weight output. Then,it is improbable that the other RSC encoder, C_2 , —which receives the interleaved version of the input— will also produce a low weight output. Hence,the interleaver spreads the low weight input sequences, so that the resulting codewords have a high Hamming weight. This interleaving operation can be defined as

$$\bar{m}_{\pi(i)} = m(i) \tag{3.2}$$

where $m(i)$ and $\bar{m}_{\pi(i)}$ are the i-th input and the output of the interleaver, N_b is the size of the interleaver, and π is some interleaving function. The amount of spreading or the efficiency of the interleaver depends on its size, and on the type of the interleaving function used. Moreover, it is this interleaving operation that gives the turbo code its random-like characteristic.

Several different types of interleavers have been used with turbo codes. These include non-random or block, semi-random, and pseudo-random interleavers. Block interleavers are defined by a matrix with v_r rows and v_c columns, such that $N_b = v_r \times v_c$. The interleaver works by writing data row-wise (or column-wise) into the matrix, and by reading data column-wise (or row-wise) out of the matrix. Block interleavers are effective if the low weight input sequence is confined to a row (assuming that data is read out of the matrix in a column-wise manner). If the low weight input sequences are confined to several consecutive rows, then the v_c columns of the interleaver need to be read out in a manner specified in (Dunscombe&Piper, 1989). However, the interleaver may still fail to spread certain sequences. In such cases, the method proposed in (Dunscombe&Piper,1989) can be used to first permute the columns and then the rows.

Semi-random or S-random interleavers work by generating random integers i, $1 \le i \le N_b$, without replacement. Each randomly selected integer is compared to S previously selected integers. If the current selection is equal to any S previous selections within a distance of $\pm S$, then the current selection is rejected. This process is repeated until all the N_b integers have been selected. Typically, S is chosen to be less than $N_b/2$ in order to minimize the search time. However,a given instance of the algorithm is not guaranteed to finish the search process successfully.

Pseudo-random interleavers map a bit position i to some other location j, $j \le N_b$, according to a prescribed, but randomly (pseudo-randomly) generated rule.Pseudo-random interleavers possess little structure, and have, therefore, been shown to perform better than both the block and the S-random interleavers. In general, for a turbo code with two RSC codes separated by a pseudo-random interleaver, the probability that a weight w input sequence will be reproduced somewhere within the turbo encoder is given by (Divsalar&Pollara,1995)

$$1-\left[1-\frac{\beta}{N_b^{w-1}}\right]^{N_b} \tag{3.3}$$

where β is a constant that depends on the weight w. It is evident from (3.3) that the probability of the mentioned event decreases rapidly as the size of the pseudo-random interleaver increases.

3.1.2 Trellis Termination

An important issue that affects the performance of turbo codes is trellis termination. Trellis termination helps impose a known boundary condition upon the trellis of a RSC code. While it is possible to terminate the trellis of a constitent RSC encoder with a tail of M_c bits, the simultaneous termination of both trellises is a non trivial task. There are two complicating factors. First, due to interleaving, the tail bits for the second (bottom) RSC encoder may not be at the end of its input stream.

Second, the process of interleaving and the recursive nature of RSC encoding complicate the computation of the tail bits suitable for both RSC encoders. Several solutions have been proposed to this trellis termination problem. For instance, the simplest solution requires that only the trellis associated with the top RSC encoder be terminated (Valenti,1999). Another solution is based on a circular-shift interleaver that uses a single tail of M_c bits to terminate both trellises (Valenti,1999). However, circular-shift registers add an extra level of structure to the interleaver, and, therefore, degrade the performance of the turbo encoder.

3.2 Iterative Turbo Decoding

The problem of turbo decoding involves estimating the Markov process associated with each of the two constituent RSC codes. Although turbo codes can be modeled as a single Markov process, the presence of an interleaver complicates the structure of the resulting trellis, and yields exorbitantly complex decoding algorithms. Therefore,the two Markov processes are estimated independently using two separate decoders (hence referred to as constituent decoders). However, the two constituent decoders operate in a symbiotic manner, i.e, the output of one feeds the input of the other, and vice versa. This symbiotic relationship is realized by the fact that the two RSC encoders (the two Markov processes) are driven by the same set of input data.Turbo decoding, therefore, proceeds in an iterative manner,where each iteration corresponds to the mentioned symbiotic cycle.

For a rate 1/n RSC code, a constituent decoder accepts the received systematic symbol $y_i^{(0)}$ the received parity symbols $\left\{y_i^{(1)},...,y_i^{(n-1)}\right\}$, and the soft output of the second decoder. It computes the log-likelihood ratio (LLR)

$$\Lambda_i = \ln\left[\frac{P(m_i = 1|y)}{P(m_i = 0|y)}\right] \tag{3.4}$$

where m_i is the systematic output of the RSC encoder (i.e. $m_i = x_i^{(0)}$), **y** is the received sequence, and, $P(m_i = b|y)$ $b \in \{0,1\}$ is the a posteriori probability (APP) of the i-th message symbol.Since the constituent decoders compute the AP probability and operate on soft information, they are commonly referred to as soft-in soft-out (SISO) APP decoders.

Using Baye's rule, (3.4) can be rewritten as

$$\Lambda_i = \ln\left[\frac{P(y|m_i=1)}{P(y|m_i=0)}\right] + \ln\left[\frac{P(m_i=1)}{P(m_i=0)}\right] \tag{3.5}$$

where the second term represents the a priori information about the message symbol m_i. For a rate 1/3 turbo code (two constituent decoders), this a priori information can be expressed as the sum

$$\ln\left[\frac{P(m_i=1)}{P(m_i=0)}\right] = W_i + Z_i \tag{3.6}$$

where W_i is the extrinsic (new information derived during the current decoding iteration) information extracted from the output of the constituent decoder, and Z_i is the soft information derived from the second decoder.

Consider a communication system with binary PSK modulation, coherent detection, and a channel model

$$y_i = a_i(2x_i - 1) + n_i, \tag{3.7}$$

where $x_i \in \{0,1\}$ is the transmitted symbol,y_i is the received symbol,a_i is the channel gain, n_i is a zero-mean Gaussian noise sample with variance $N_0/2E_s$, E_s is the energy per symbol, and $N_0/2$ is the double-sided noise power spectral density. Using the mentioned channel model, (3.5) can be rewritten as (Hagenauer,1996)

$$\Lambda_i = \frac{4a_i^{(0)}E_s}{N_0} y_i^{(0)} + Z_i + W_i, \tag{3.8}$$

and the extrinsic information derived from the output of the constituent decoder can be expressed as

$$W_i = \Lambda_i - \frac{4a_i^{(s)}E_s}{N_0} y_i^{(s)} - Z_i \tag{3.9}$$

Fig. 3.2 shows the block diagram for a rate 1/3 turbo decoder. The first constituent decoder, D_1, receives the scaled systematic received sequence $y^{(0)}$, the first scaled parity received sequence $y^{(1)}$, and the a priori information Z_2 from the second decoder D_2. It computes the LLR's Λ_1. The extrinsic information W_1 is extracted from Λ_1 using (3.9). This extrinsic information is interleaved (the interleaver used in the turbo decoder is same as the interleaver used in the turbo encoder), and is used as a priori information Z_1 by the second decoder. The second decoder, D_2, also receives the interleaved and scaled systematic received sequence $\overline{y}^{(0)}$, and the second scaled parity sequence $y^{(2)}$. It generates the LLRs Λ_2. The extrinsic information W_2 is extracted from Λ_2 using (3.9), is de-interleaved, and used as a priori information by D_1 during the next decoding iteration. After η iterations, the final estimate of the message sequence $\hat{m}$ is found by de-interleaving and hard-limiting the output of D_2 i.e.

$$\hat{m}_i = \begin{cases} 1 & if \quad \Lambda_{2_{\pi^{-1}(i)}} \geq 0 \\ 0 & if \quad \Lambda_{2_{\pi^{-1}(i)}} < 0 \end{cases} \tag{3.10}$$

If puncturing is used, then each decoder will not have a complete set of observations of the corresponding encoder's parity bits. In this case, the observed values of the bits that were punctured prior to transmission are simply set to zero.

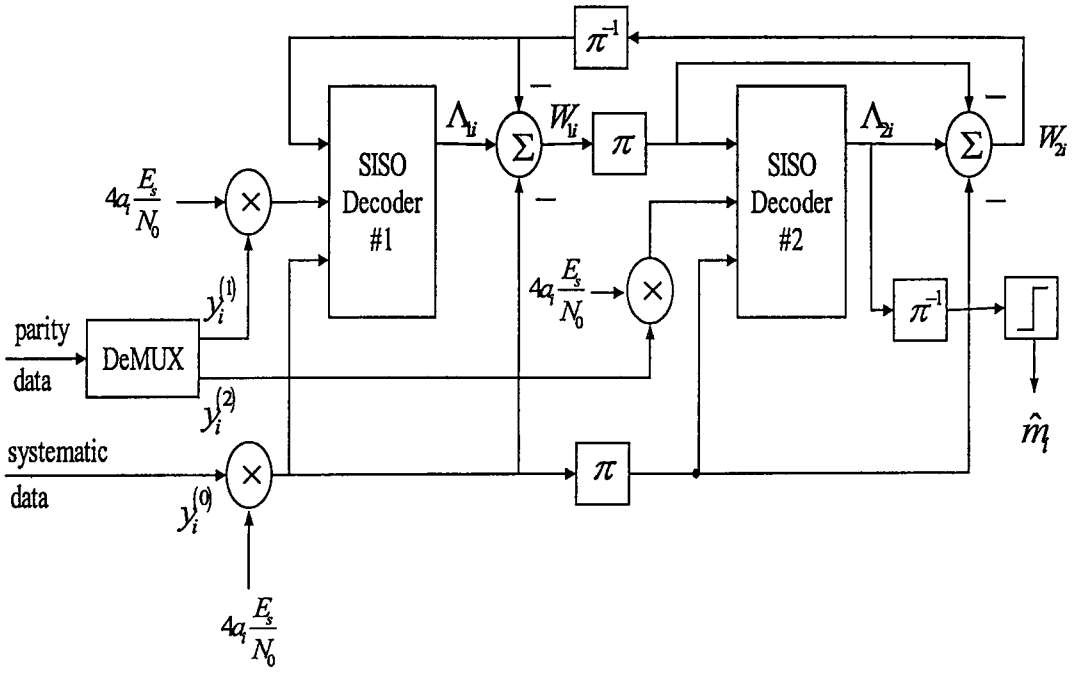


Fig. 3.2. Turbo Decoder Schematic

3.3 A Posteriori Probability Algorithms

Several different decoding strategies have been adopted for the SISO APP decoding of turbo codes. These decoding schemes can be classiffed as either maximum likelihood sequence estimation (MLSE) algorithms, or as symbol-by-symbol maximum a posteriori probability (MAP) algorithms. As their names suggest, the two mentioned algorithms differ in the way they estimate a given Markov process. For instance, MLSE algorithms attempt to estimate the most probable state sequence, $\hat{s}$, of the Markov process, given a received sequence y

$$\hat{s}_{MLSE} = \arg\max_{s}\left[P\left(s|y\right)\right] \tag{3.11}$$

Contrariwise, MAP algorithms find the most likely individual states, $\hat{s}_i$, given y

$$\hat{s}_{MAP} = \arg\max_{s_i}\left[P\left(s|y\right)\right] \tag{3.12}$$

Hence, the sequence $\hat{s}_{MLSE}$ traverses a continuous path through the trellis defined by the Markov process.However,the sequence $\hat{s}_{MAP}$ need not form such a continuous path. Therefore, MLSE algorithms minimize the codeword error probability or the state estimation error rate (SER), while MAP algorithms minimize the symbol error probability or bit error rate (BER).

The MLSE family of APP decoding algorithms include the soft output Viterbi algorithm (SOVA) (Hagenauer&Hoeher,1989), and the Improved SOVA algorithm (Papke et.al.,1996). The MAP family of decoding algorithms include the modified BCJR algorithm (Bahl et.al.,1974) —originally used by Berrou et al. It also includes the log-MAP and the Max-Log-MAP algorithms (Robertson et.al.,1997) (Viterbi,1998).

3.3.1 Modified BCJR Algorithm

The symbol-by-symbol MAP algorithm —also known as the BCJR algorithm— was developed by Bahl et al as an alternative to the Viterbi algorithm (Viterbi,1967) for decoding convolutional codes (Bahl et.al.,1974). When applied to turbo codes, a modified BCJR is used to account for the recursive nature of iterative turbo decoding. The BCJR algorithm computes the APP of the message symbols, and the LLR

$$\Lambda_i = \ln\left[\frac{P(m_i = 1|y)}{P(m_i = 0|y)}\right] \tag{3.13}$$

where mi is a message symbol (systematic output of the RSC enoder), and y is the received sequence of length Nb. By incorporating the RSC code's trellis, (3.13) can be rewritten as

$$\Lambda_i = \ln\left[\frac{\sum_{S^+} P(s_{i-1} = s', s_i = s, y)/P(y)}{\sum_{S^-} P(s_{i-1} = s', s_i = s, y)/P(y)}\right] \tag{3.14}$$

where $s_i \in S = \{0,...,2^{M_c} - 1\}$ is the state of the encoder at time i, S^+ is the set of ordered pairs (s',s) corresponding to all state transitions $(s_{i-1} = s') \rightarrow (s_i = s)$ caused by the message symbol $m_i = 1$, and S^- is likewise defined for $m_i = 0$. The probability $P(s_{i-1} = s', s_i = s, y)$ can be expressed as the product of three probabilities

$$P(s_{i-1} = s', s_i = s, y) = \alpha_{i-1}(s')\gamma_i(s',s)\beta_i(s) \tag{3.15}$$

where the α, β, and γ probabilities are defined as

$$\alpha_i(s) = P\left(s_i = s, y_1^i\right) \tag{3.16}$$

$$\beta_i(s) = P\left(y_{i+1}^N \middle| s_i = s\right) \tag{3.17}$$

$$\gamma_i(s', s) = P\left(s_i = s, y_i \middle| s_{i-1} = s'\right) \tag{3.18}$$

Hence, the α probability represents the probability of being in state s, given the partial received sequence until time i. Likewise, the β probability represents the probability 0of being in state s, given the partial received sequence after time i. The probability γ represents the probability of transition from state s' to state s at time i.

The α probabilities are computed via a forward recursion through the trellis

$$\alpha_i(s) = \sum_{s'} \alpha_{i-1}(s')\gamma_i(s', s) \quad i \in \{1, 2, ..., N-1\} \tag{3.19}$$

For large N, the α probabilities approach zero geometrically fast, and must, therefore, be normalized at each time index to maintain accuracy and stability. Hence,

$$\alpha_i(s) = \frac{\sum_{s'} \alpha_{i-1}(s')\gamma_i(s', s)}{\sum_{s}\sum_{s'} \alpha_{i-1}(s')\gamma_i(s', s)} \qquad i \in \{1, 2, ..., N-1\} \tag{3.20}$$

Furthermore, since the RSC encoder starts in the all-zero state, the α probabilities are initialized as

$$\alpha_0(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0 \end{cases} \tag{3.21}$$

The β probabilities are computed and normalized via a backward recursion through the trellis

$$\beta_{i-1}(s') = \frac{\sum_{s} \beta_i(s)\gamma_i(s', s)}{\sum_{s}\sum_{s'} \alpha_{i-1}(s')\gamma_i(s', s)} \qquad i \in \{N-1, N-2, ..., 0\}, \tag{3.22}$$

If the RSC encoder terminates in the all-zero state, then the β probabilities are initialized as

$$\beta_{N-1}(s)=\begin{cases}1 & s=0\\ 0 & s\neq 0\end{cases} \tag{3.23}$$

Contrariwise, if the RSC encoder is left unterminated, then

$$\beta_{N-1}(s)=\frac{1}{2^{M_c}} \qquad \forall s \tag{3.24}$$

The γ probability defined in (3.18) can be expressed as

$$\gamma_i(s',s)=P(m_i)P(y_i|x_i) \tag{3.25}$$

where x_i and y_i are the transmitted and the received codewords associated with the message symbol m_i. The probability $P(m_i)$ is obtained from the a priori information Z_i at the output of the second APP decoder

$$P(m_i)=\begin{cases}\dfrac{e^{Z_i}}{1+e^{Z_i}} & m_i=1\\ \dfrac{1}{1+e^{Z_i}} & m_i=0\end{cases} \tag{3.26}$$

Alternately,the probability $P(m_i)$ can be written in a more compact form as

$$P(m_i)=\frac{\left(e^{Z_i}\right)^{m_i}}{1+e^{Z_i}} \tag{3.27}$$

Using Bayes' theorem, the a priori probability $P(y_i|x_i)$ in (3.25) can be rewritten as

$$P(y_i|x_i)=\frac{P(x_i|y_i)P(x_i)}{P(y_i)}=P(x_i|y_i).C \tag{3.28}$$

where C is constant for a given codeword, and can, therefore, be ignored from all future computations. Defined the channel reliability variable, $R_i^{(q)}$, as

$$R_i^{(q)}=\ln\left[\frac{P\left(x_i^{(q)}=1|y\right)}{P\left(x_i^{(q)}=0|y\right)}\right] \qquad q\in\{0,1,\ldots,n-1\} \tag{3.29}$$

Using Bayes' theorem the AP probability $P\left(x_i^{(q)}=b\middle|y\right)$, $b\in\{0,1\}$, can be written in terms of the a priori probability $P\left(y_i^{(q)}\middle|x_i^{(q)}=b\right)$. Furthermore, using the channel model defined in (3.7), the probability $P\left(y_i^{(q)}\middle|x_i^{(q)}\right)$ can be expressed as

$$P\left(y_i\middle|x_i\right)=\frac{1}{\sqrt{\pi N_0/E_s}}\exp\left\{\frac{-E_s}{N_0}\left[y_i^{(q)}-a_i^{(q)}\left(2x_i^{(q)}-1\right)\right]^2\right\} \tag{3.30}$$

Consequently the channel reliability variable can be simplified as

$$R_i^{(q)}=4a_i^{(q)}\frac{E_s}{N_0}y_i^{(q)} \tag{3.31}$$

and the AP probabilities $P\left(x_i^{(q)}=b\middle|y_i^{(q)}\right)$, $b\in\{0,1\}$, can either be written as

$$P\left(y_i^{(q)}\middle|x_i^{(q)}\right)=\begin{cases}\dfrac{e^{R_i^{(q)}}}{1+e^{R_i^{(q)}}} & x_i^{(q)}=1\\[2ex] \dfrac{1}{1+e^{R_i^{(q)}}} & x_i^{(q)}=0\end{cases} \tag{3.32}$$

or alternately as

$$P\left(y_i^{(q)}\middle|x_i^{(q)}\right)=\frac{\left(e^{R_i}\right)^{x_i^{(q)}}}{1+e^{R_i^{(q)}}} \tag{3.33}$$

Therefore,the γ probability in (3.25) can be rewritten as

$$\gamma_i\left(s',s\right)=\frac{\left(e^{Z_i}\right)^{m_i}}{1+e^{Z_i}}\prod_{q=0}^{n-1}\frac{\left(e^{R_i}\right)^{x_i^{(q)}}}{1+e^{R_i^{(q)}}} \tag{3.34}$$

Since the denominator in (3.34) is constant for a given codeword, the computation of the γ probability can be simplified as

$$\gamma_i\left(s',s\right)=\left(e^{Z_i}\right)^{m_i}\prod_{q=0}^{n-1}\left(e^{R_i}\right)^{x_i^{(q)}} \tag{3.35}$$

By using the α, β, and γ probabilities, the BCJR algorithm computes the LLR

$$\Lambda_i = \ln\left[\frac{\sum_{s^+}\alpha_{i-1}(s')\gamma_i(s',s)\beta_i(s)}{\sum_{s^-}\alpha_{i-1}(s')\gamma_i(s',s)\beta_i(s)}\right] \tag{3.36}$$

Note that the term P(y) in (3.14) cancels out, and, therefore, need not be computed. Furthermore, the extrinsic information, W, can be computed using (3.9).

3.3.2. Log-MAP and the Max-Log-MAP Algorithms

Although optimal, the BCJR algorithm brings several technical difficulties. These include sensitivity to numeric representation, the necessity of non-linear functions, and a high number of additions and multiplications. These technical difficulties can be alleviated by implementing sub-optimal algorithms such as the Log-MAP algorithm or the Max-Log-MAP algorithm (Robertson et.al,1997), (Viterbi,1998). As their names suggest, these algorithms compute the α, β, and the γ probabilities in the log-domain. Both algorithms perform multiplications in the log-domain as additions. However, they perform additions in the log-domain by using an approximation of the Jacobian logarithm. This approximation is known as the max*( ) function, and is defined as

$$\max^*(x,y) = \begin{cases} \max(x,y) & \text{Max-Log-MAP algorithm} \\ \max(x,y) + f_c(|y-x|) & \text{Log-MAP algorithm} \end{cases} \tag{3.37}$$

where $f_c(\cdot\ )$ is the associated correction function. Therefore, the only difference between the Log-MAP algorithm and the Max-Log-MAP algorithm lies in the implementation of the max*( ) function.

Let $\overline{\alpha}, \overline{\beta}$, and $\overline{\gamma}$ represent the natural logarithm of the α, β, and the γ probabilities.Hence, $\overline{\gamma}$ can be expressed as

$$\overline{\gamma}_i(s',s) = m_i Z_i + \sum_{q=0}^{n-1} x_i^{(q)} R_i^{(q)} \tag{3.38}$$

or alternately as

$$\bar{\gamma}_i(s',s)=\begin{cases}\sum_{q=1}^{n-1} x_i^{(q)} R_i^{(q)} & m_i = x_i^{(0)} = 0 \\ Z_i + R_i^{(0)} + \sum_{q=1}^{n-1} x_i^{(q)} R_i^{(q)} & m_i = x_i^{(0)} = 1\end{cases} \tag{3.39}$$

Likewise, $\bar{\alpha}$ and $\bar{\beta}$ can be computed and normalized as

$$\begin{aligned}\bar{\alpha}_i(s) &= \ln\left\{\sum_{s'} \exp\left[\bar{\alpha}_{i-1}(s')+\bar{\gamma}_i(s',s)\right]\right\} - \max_s\left[\bar{\alpha}_i(s)\right] \\ &= \max_{s'}{}^{*}\left[\bar{\alpha}_{i-1}(s')+\bar{\gamma}_i(s',s)\right] - \max_s\left[\bar{\alpha}_i(s)\right]\end{aligned} \tag{3.40}$$

and

$$\begin{aligned}\bar{\beta}_{i-1}(s) &= \ln\left\{\sum_{s} \exp\left[\bar{\beta}_i(s)+\bar{\gamma}_i(s',s)\right]\right\} - \max_{s'}\left[\bar{\beta}_{i-1}(s')\right] \\ &= \max_{s}{}^{*}\left[\bar{\beta}_i(s')+\bar{\gamma}_i(s',s)\right] - \max_{s'}\left[\bar{\beta}_{i-1}(s')\right]\end{aligned} \tag{3.41}$$

Since the RSC encoder starts in the all-zero state, $\bar{\alpha}$ is initialized as

$$\bar{\alpha}_0(s)=\begin{cases}0 & s=0 \\ -\infty & s\neq 0\end{cases} \tag{3.42}$$

If the RSC encoder terminates in the all-zero state,then $\bar{\beta}$ is initialized as

$$\bar{\beta}_{N_b-1}(s)=\begin{cases}0 & s=0 \\ -\infty & s\neq 0\end{cases} \tag{3.43}$$

CHAPTER FOUR
TURBO TRELLIS CODED MODULATION

4.1. Turbo Codes with High Spectral Efficiency Modulations

The first attempt in combining turbo codes with multilevel modulation was described in (Goff et al.,1994) and is called "pragmatic" approach to TCM. In this approach a Gray mapper is used after binary turbo encoder for multilevel modulation. The coding and modulation are separate processes and hence it is actually not a coded-modulation scheme. Decoding relies on binary turbo decoder, hence the term "pragmatic". In the second approach (Roberson&Woerz,1996) binary RSC component codes in binary turbo code are replaced by Ungerboeck TCM codes to retain the advantages of both classical turbo code and TCM

Figure 4.1 shows the association of a binary turbo code with *M*-level modulation (MPSK, M-level QAM). The standard turbo code uses two rate-1/2 RSC codes as constituent codes. The parity check bits at the output of constituent codes are denoted as c^1 and c^2 respectively. The puncturing function is inserted at the output of the standard turbo code and thus it is possible to obtain a large code family with various rates $R=(m-n)/m$. In addition to the interleaver I_1 inside the turbo encoder, another interleaver I_2 is inserted between the puncturing function and the modulator in order to obtain symbols affected with independent noise at the turbo decoder input. It was mentioned in (Goff et al.,1994) that this second interleaver I_2 is only necessary when the symbols are transmitted over a fading channel. At any time k a set $\{u_{k,i}\}$ (i = 1, ..., m) of m bits is Gray mapped into a complex signal symbol s_k^j (j=1,...,M) to be transmitted over the channel. Each symbol s_k^j is represented by a

couple of real-valued symbols, $\{A_k^j, B_k^j\}$. The redundant bits provided by the turbo encoder are always associated to the highest protected bits $u_{k,i}$.

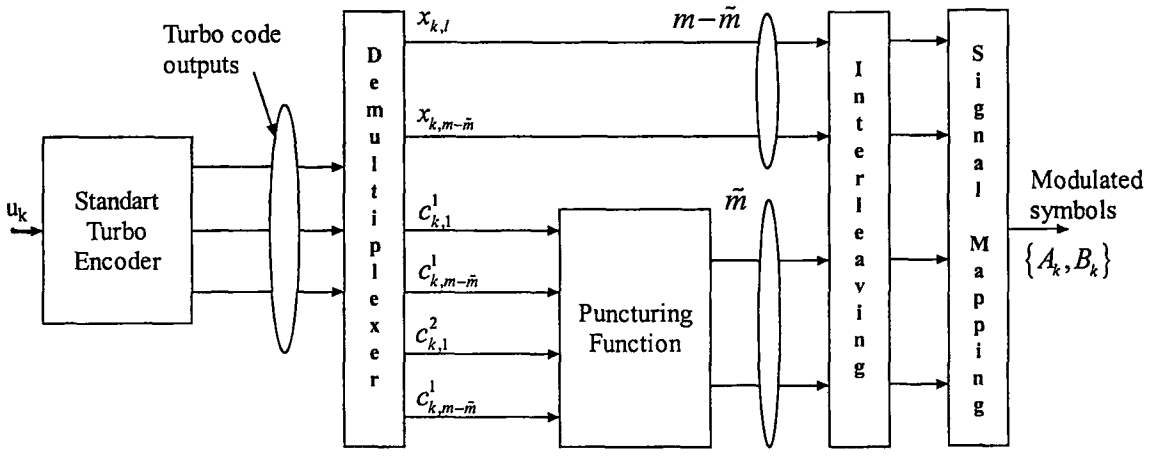


Figure 4.1. Association of turbo codes with multilevel modulations

The receiver for the above association scheme is shown in Fig. 4.2. The correlation demodulator (or matched filter demodulator) produces the output of noisy symbol r_k . Each noisy symbol r_k consists of in-phase and quadrature components X_k and Y_k respectively and contains all the sufficient information in the received signal waveform. Based on the observation of r_k the log-likelihood value associated with each bit $u_{k\,i}$, i=1, ..., m can be determined and then used as a relevant soft information by the binary turbo decoder:

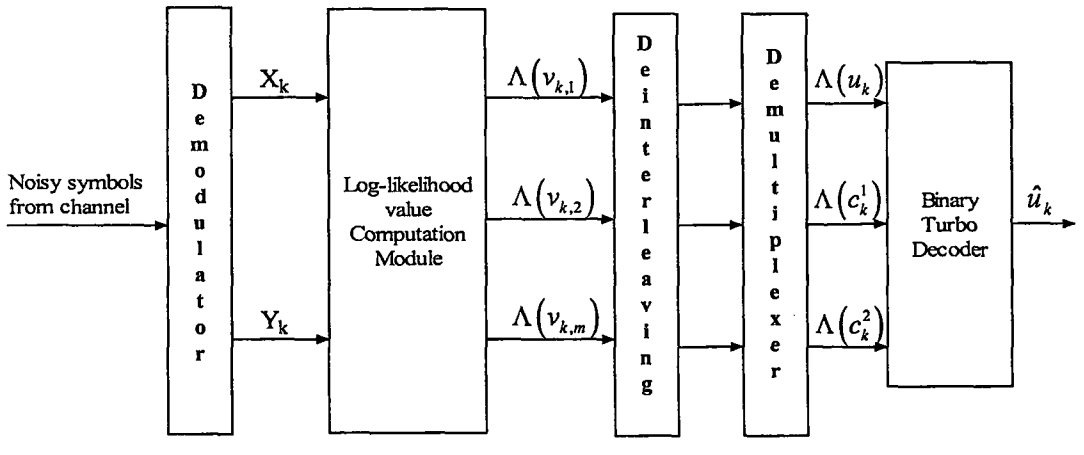


Figure 4.2. Decoder structure in pragmatic approach

4.2. Turbo Trellis Coded-Modulation

In this second approach, two Ungerboeck TCM codes in their recursive systematic feedback form are used as component codes in a similar way as in binary turbo codes. The recursive systematic TCM code is essentially a recursive systematic convolutional (RSC) encoder followed by symbol mapper. Usually the rate of RSC code is (m-1)/m. It means that one redundant bit is created for every (m-1) information bits and then m bits are mapped into one of $M=2^m$ symbols following Ungerboeck's mapping by set partitioning rule. The general structure of RSC code that constructs recursive systematic TCM code is illustrated in Fig.4.3.

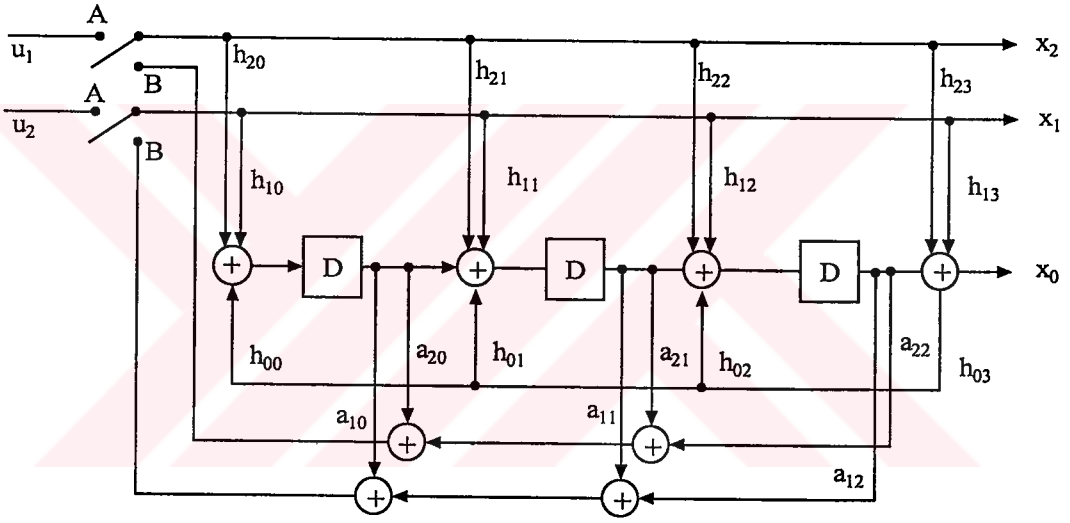


Figure 4.3. Canonical structure of rate $k/(k+1)$ encoder. (k=2,m=3)

There are 3 memories in Fig. 4.3 hence 8 states of the trellis diagram, k information bits are encoded by a rate-k/(k+1) RSC code. Trellis termination can be realized by moving the switch from position A to B with appropriate a coefficients.

Figure 4.4 shows the encoder structure of T-TCM when the component recursive systematic TCM is 8-PSK TCM.

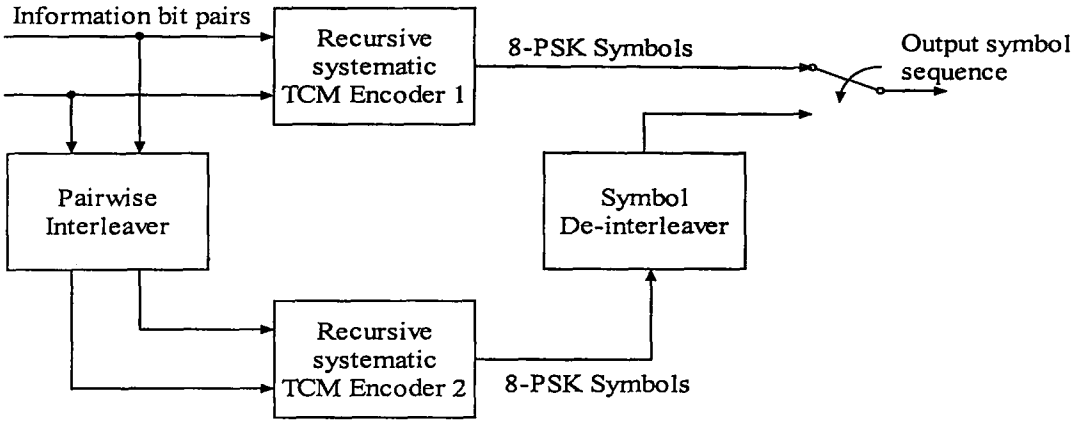


Figure 4.4 Turbo-TCM encoder

The following are the major differences when comparing the above encoder with binary turbo encoder (Roberson&Woerz 1996).

-Interleaving operates on group of *k* bits instead of single bits.
-In order to achieve a systematic overall code, the pairwise interleaver must map even positions to even positions and odd positions to odd positions (or even-odd, odd-even).
-For the component code, the corresponding trellis diagram should have no parallel transitions and the information bits at time *k* do not affect the parity bits at time k.

In Fig. 4.4 the sequence of information bit pairs is TCM encoded to yield the 8-PSK symbol sequence. The information bits are then interleaved on a pairwise basis and TCM encoded again into the second 8-PSK symbol sequence. The second 8-PSK symbol sequence is deinterleaved to ensure that the ordering of the two information bits partly defining each symbol corresponds to that of the first encoder. Finally the transmitted symbol sequence is alternatively selected from two output symbol sequences. In this way, each information bit pair is contained in only one 8-PSK symbol and the parity bit is alternately chosen from the first and second encoder.

4.2.1. T-TCM Decoder

Figure 4.5 illustrates structure of the complete T-TCM decoder. Basically iterative decoding in T-TCM is similar to that in binary turbo code. However there is a difference in the nature of the information passed between component decoders and in the first decoding step. In the binary turbo code, the output of component decoder can be split into three additive parts for each information bit k in the logarithmic domain: the *systematic* component, the *a priori* component and the *extrinsic* component and only the last component can be passed to the next decoder. In T-TCM each decoder alternatively sees its corresponding encoder's noisy output symbol and then the other encoder's noisy output symbol. The noisy output symbol corresponding to the other encoder are referred to "punctured" symbols and denoted by "*" in Fig. 4.5

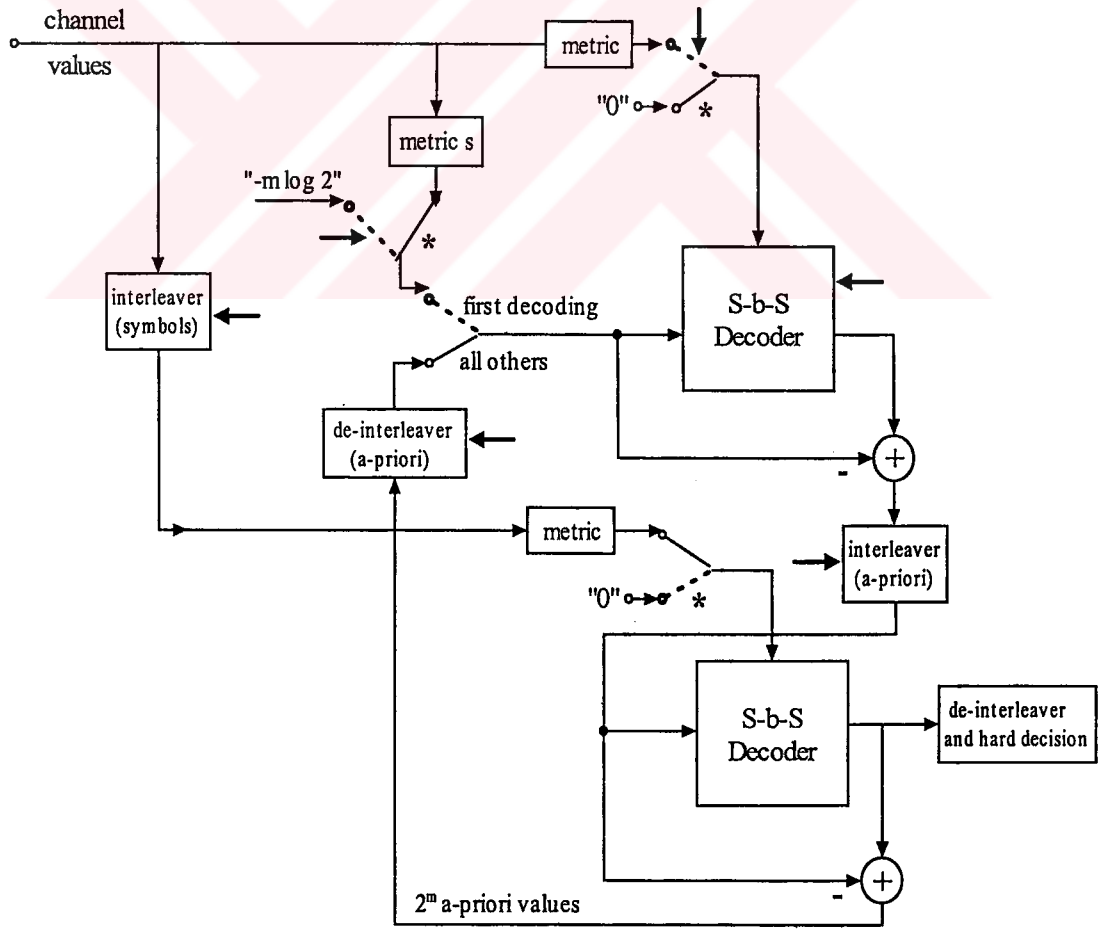


Fig. 4.5. Complete Decoder

4.2.2. MAP decoder for non-binary trellises

In (Robertson&Woerz,1996), MAP algorithm is adapted for non-binary codes and used for T-TCM. $s' = s_{k-1}$ and $s = s_k$, being the states at time *k-1* and *k* respectively, and u_k being the input symbol for state transition $s' \to s$, the output of MAP algorithm for each symbol is

$$P\{u_k = i \mid y\} = const \cdot \sum_{s} \sum_{s'} \gamma_i(y_k, s', s) \cdot \alpha_{k-1}(s') \beta_k(s), \quad \forall i \in \{0,1,...,2^m - 1\}. \quad (4.1)$$

The branch transition probability for step *k*, $P_r\{u_k = i, y_k, S_k = s \mid S_{k-1} = s'\}$, is denoted by, and calculated as

$$\gamma_i(y_k, s', s) = p(y_k \mid u_k = i, S_k = s, S_{k-1} = s') \cdot q(u_k = i \mid S_k = s, S_{k-1} = s') \cdot P\{S_k = s \mid S_{k-1} = s'\}. \quad (4.2)$$

$q(u_k = i \mid S_k = s, S_{k-1} = s')$ is either zero or one depending on whether encoder input $i \in \{0,1,...,2^{m-1} - 1\}$ is associated with the transition from state $S_{k-1} = s'$ to $S_k = s$ or not. The received vector $\vec{y} = (y_1, y_2, ..., y_n)$ is the TCM encoder output sequence that has been disturbed by AWGN with one-sided noise power spectral density N_0. In the last component we use the a-priori information:

$$P_r\{S_k = s \mid S_{k-1} = s'\} = \begin{cases} P_r\{u_k = 0\} & if \quad q(u_k = 0 \mid S_k = s, S_{k-1} = s') = 1 \\ P_r\{u_k = 1\} & if \quad q(u_k = 1 \mid S_k = s, S_{k-1} = s') = 1 \\ ... & .... \\ P_r\{u_k = 2^{m-1} - 1\} & if \quad q(u_k = 2^{m-1} - 1 \mid S_k = s, S_{k-1} = s') = 1 \end{cases}$$

$$= P_r\{u_k = j\} \quad \text{where} \quad q(u_k = j \mid S_k = s, S_{k-1} = s') = 1. \quad (4.3)$$

If there does not exist a j such that $q(u_k = i \mid S_k = s, S_{k-1} = s') = 1$, then $P_r\{S_k = s \mid S_{k-1} = s'\}$ is set to zero. It's important to note that since (4.3) will only be non-zero if $q(u_k = i \mid S_k = s, S_{k-1} = s')$ is one, it suffices to set $P_r\{S_k = s \mid S_{k-1} = s'\} = P_r\{u_k = i\}$ when calculating $\gamma_i(y_k, s', s)$. The forward and backward probabilities $\alpha_k(s)$ and $\beta_k(s)$ are:

$$\alpha_k(s) = \frac{P_r\{S_k = s, \vec{y}_1^k\}}{P_r\{\vec{y}_1^k\}} = \frac{\sum_{s'}\sum_{i=0}^{2^m-1}\gamma_i(y_k, s', s)\cdot\alpha_{k-1}(s')}{\sum_{s}\sum_{s'}\sum_{i=0}^{2^m-1}\gamma_i(y_k, s', s)\cdot\alpha_{k-1}(s')} \tag{4.4}$$

$$\beta_k(s) = \frac{P_r\{\vec{y}_{k+1}^N \mid S_k = s\}}{P_r\{\vec{y}_{k+1}^N \mid \vec{y}_1^k\}} = \frac{\sum_{s''}\sum_{i=0}^{2^m-1}\gamma_i(y_{k+1}, s, s'')\cdot\beta_{k-1}(s')}{\sum_{s''}\sum_{s}\sum_{i=0}^{2^m-1}\gamma_i(y_k, s, s'')\cdot\alpha_k(s)} \tag{4.5}$$

where $\gamma_i(y_k, s, s'') = P_r\{d_{k+1} = i, y_{k+1}, S_{k+1} = s'' \mid S_{k-1} = s\}$. The output of the MAP is:

$$P_r\{u_k = i \mid \vec{y}\} = const\cdot\sum_{s}\sum_{s'}\gamma_i(y_k, s', s)\cdot\alpha_{k-1}(s')\beta_k(s), \quad \forall i \in \{0,1,...,2^m - 1\}. \tag{4.6}$$

4.2.3 A MAP equivalent SOVA for Non-binary Turbo Codes

The complexity of MAP algorithm requires the implementation in logarithm domain, where the MAP algorithm becomes Log-MAP algorithm. Further simplification to Log-MAP have been proposed, such as a look-up table approximation, and Max-Log-MAP (Robertson et.al.,1995). There are two recursion processes in the MAP algorithm, the forward and the backward recursion. In (Viterbi,1998), the forward and backward recursions are simplified as forward and backward VA.

The MAP and Log-MAP algorithms are optimal maximum a posteriori sequence probability algorithms. The simplified versions of MAP or Log-MAP, such as Max-

Log-MAP, SOVA, are suboptimal. It is found in (Robertson et.al.,1995) that SOVA is roughly half as complex as the Log-MAP with some performance degradation. As for non-binary codes, the complexity of MAP algorithm requires the simplified suboptimal algorithms, with desirable small performance degradation. In this thesis, a different implementation of MAP algorithm which was proposed in (Tan&Stuber,2000), together with a suboptimal algorithm is used.

The Viterbi Algorithm (VA) (Forney,1973) is a maximum a posteriori probability sequence estimator, which finds the maximum a posteriori probability path over the trellis diagram given the received sequence $y=\{y_k\}_{k=1}^{N}$. Here the symbol block length or symbol frame length is N, while the binary block length is mN. A path in the trellis can be represented by a state sequence $\mathbf{s}=\{s_1,s_2,...,s_N\}$, which indicates the trellis path starting at state s_1, passing through every state s_k at each k, and terminating at state s_N. The VA finds the trellis path or state sequence $\boldsymbol{s}$ so that the a posteriori probability $P(s|y)$ is maximized. If the state sequence $\boldsymbol{s}$ is assumed to be a Markov sequence, and since the received sequence $y=\{y_k\}_{k=1}^{N}$ does not dependent on the selection of trellis path $\boldsymbol{s}$, and $P(s|y)=P(s,y)/P(y)$, at each k, we can equivalently maximize

$$p(s_k,y_k)=p(s_{k-1},y_{k-1})P(u_k)p(y_k|s',s) \tag{4.7}$$

where $s_k=\{s_1,s_2,...,s_k\}$, $y_k=\{y_1,y_2,...,y_k\}$, $s'=s_{k-1}$, $s=s_k$, and u_k is the source symbol for the state transition $s'\rightarrow s$ of trellis path s_k. The path metric $M_k(s_k)$ associated with the trellis path s_k is defined as

$$M_k(s_k)=\log\left(p(s_k,y_k)\right). \tag{4.8}$$

Substituting (4.7) into (4.8), and using the notation $\lambda=\log P$ gives,

$$M_k(s_k)=M_{k-1}(s_{k-1})+\lambda(u_k)+\lambda(y_k|s',s). \tag{4.9}$$

In (4.9), $\lambda(u_k)$ is the a priori information of the source symbol u_k, and $\lambda(y_k|s',s)$ is the branch metric for the state transition $s'\rightarrow s$ given the received signal y_k. At

each k, for each states, the path metrics for all possible paths terminating at state s are calculated. Only the maximal path metric is saved and the corresponding path is the survivor path.

If we define $P(\hat{u}_k = a|y)$ and $p(s',s,y)$ as

$$P(\hat{u}_k = a|y) = \sum_{\substack{(s',s) \\ u_k=a}} p(s,s,y) \tag{4.10}$$

$$\begin{aligned} p(s',s,y) &= P(s',s|y)p(y) \\ &= P(s',s|y)P(s|y)p(y) \end{aligned} \tag{4.11}$$

where p(y) is similar for all state transitions. The probability of s' given s and the received y is

$$P(s'|s,y) = \frac{p\left(s_k^{(s'\to s)}, y_{j<k}\right)}{\sum_{s''} p\left(s_k^{(s''\to s)}, y_{j<k}\right)} \tag{4.12}$$

where $s_k^{(s'\to s)}$ is the trellis path containing branch $(s'\to s)$, and $s_k^{(s''\to s)}$ is the trellis path containing branch $(s''\to s)$ at each k. For each states at k, the number of trellis branches terminated at s is equal to M for M-ary codes. There are M possible states s'' for state transitions $s''\to s$. The ratio of probability of a trellis path containing branch $(s'\to s)$ to the sum of probabilities of all possible trellis paths terminated at s is the probability of state s' given s and y, i.e., $P(s'|s,y)$. Since $p(s_k, y_k) = \exp\left(M_k(s_k)\right)$,

$$P(s'|s,y) = \frac{\exp\left(M_k\left(s_k^{(s'\to s)}\right)\right)}{\sum_{s''} \exp\left(M_k\left(s_k^{(s''\to s)}\right)\right)} \tag{4.13}$$

where the received symbol sequence $y_{j>k}$ is independent of the state transition at each k. The conditional probability P(s|y) can be yielded through backward recursion, as

$$P(s|y) = \sum_{s} P(s'|s,y)P(s|y) \tag{4.14}$$

The initial value of the backward recursion is P(s|y) =1 for terminal state s at each k=N, and P(s|y) = 0 otherwise.

From the soft output definition in (4.10), the soft output should have M possible values for M-ary source symbols. The soft output value of the proposed algorithm in probability form is obtained as,

$$P(\hat{u}_k = a \mid y) = \sum_{\substack{(s' \to s) \\ u_k = a}} P(s \mid y) \frac{\exp\left(M_k\left(s_k^{(s' \to s)}\right)\right)}{\sum_{s''} \exp\left(M_k\left(s_k^{(s'' \to s)}\right)\right)} \tag{4.15}$$

Using the joint probability in (4.11) with the a posteriori probability definition in (4.10), we derived the implementation of MAP algorithm. The path metrics are computed with VA. (4.11) is the a posteriori probability, or the soft output definition of VA. So we still can call the proposed MAP implementation as a SOVA. However, the proposed SOVA is different from the SOVA in literature (Hagenauer&Hoeher,1989). The proposed SOVA is an optimal MAP algorithm, while the conventional SOVA is a Max-Log-MAP equivalent, which is sub-optimal.

The SOVA soft output in (4.15) can be simplified by passing to the log-domain of probability. This leads to the problem of evaluating the logarithm of a sum of exponentials $\log\left\{\sum_i e^{Ai}\right\}$. Define $A_{\max} = \max_i \{A_i\}$, then we have,

$$\log\left(\sum_i e^{Ai}\right) = A_{\max} + \log\left(1 + \sum_{A_i \neq A_{\max}} e^{A_i - A_{\max}}\right) \approx A_{\max}. \tag{4.16}$$

If we neglect the second term, and use the log-domain denotation $\lambda = \log P$, and applying the approximation $\log\left(\sum_i e^{Ai}\right) \approx A_{\max}$ to (4.10), yields

$$\lambda(u_k = a \mid y) \approx \max_{\substack{(s' \to s) \\ u_k = a}} \{\lambda(s', s \mid y)\} = \max_{\substack{(s' \to s) \\ u_k = a}} \{\lambda(s' \mid s, y) + \lambda(s, y)\}. \tag{4.17}$$

The approximations to $\lambda(s'|s,y)$ and $\lambda(s'|y)$ can be obtained through equations (4.13) and (4.14), respectively, as

$$\lambda(s'|s,y) \approx M_k\left(s_k^{(s'\to s)}\right) - \max_{s^*}\left\{M_k\left(s_k^{(s^*\to s)}\right)\right\}, \tag{4.18}$$

$$\lambda(s'|y) \approx \max_{s}\left\{\lambda(s'|s,y) + \lambda(s|y)\right\}. \tag{4.19}$$

The path metric $M_k(s_k)$ can be obtained by (4.9). For M-ary symbols, each symbol has M soft output values, corresponding to a = 0,1,...,M-1. Here, the a priori information is $\lambda(u_k)$. The soft output is $\lambda(\hat{u}_k) = \lambda(\hat{u}_k = a|y)$. The extrinsic information, which is the information passed to the next iterative decoder, is $\lambda_e(\hat{u}_k) = \lambda(\hat{u}_k) - \lambda(u_k)$.

Unlike the conventional SOVA, the proposed SOVA does not need the storage of reliability measures for all states. The soft output values are calculated by forward and backward process. This is different from the updating process for the reliability measures in the conventional SOVA. The simplified proposed SOVA needs the storage of metric difference $\Delta_k^{(s)}$ at every state. This is similiar to the Max-Log-MAP, which requires the entire state metric history be stored. However, in the backward recursion, the proposed SOVA updates the backward value $\lambda(s'|y)$ with the stored metric difference $\Delta_k^{(s)}$, while the Max-Log-MAP needs a full backward "VA" operation. The simplified proposed SOVA only needs one "VA" operation for the calculation of metrics. It is less complex than the Max-Log-MAP algorithm.

4.2.4. Extrinsic, A Priori, and Systematic Components

The component (extrinsic and systematic) , abbreviated (*e&s*) should be passed to the next decoder in which it is used as *a priori* information. We shall define the component (extrinsic and systematic) as that part of the decoder output that does not depend on the *a priori* information $P_r\{u_k = i\}$. In other words, we must subtract the *a*

priori term $\log P_r\{u_k = i\}$ from $P_r\{u_k = i \mid \vec{y}\}$ to obtain a term independent of the *a priori* information $P_r\{u_k = i\}$:

$$L_{e\&s}(u_k = i) = \log P_r\{u_k = i \mid \vec{y}\} - \log P_r\{u_k = i\}; \qquad \forall i \in \{0,1,...,2^m - 1\}. \quad (4.20)$$

The decoder must be formulated in such a way that it correctly uses the channel observation y_k and the *a priori* information $P_r\{u_k = i\}$ at each step *k*.

Since the *a priori* information for the upper decoder is not available, in the first decoding stage, a metric calculation is necessary. If the upper decoder is not at a * transition, then we simply set $P_r\{u_k = i\}$ to $1/2^m$.

CHAPTER FIVE
MULTIPATH FADING CHANNELS AND EQUALIZERS

Small-scale fading, or simply fading, is used to describe the rapid fluctuation of the amplitude of a radio signal over a short period of time or travel distance, so large scale path loss effects may be ignored. Fading is caused by interference between two or more versions of the transmitted signal which arrive at the receiver at slightly different times. These waves combine at the receiver antenna to give a resultant signal which can vary widely in amplitude and phase, depending on the distribution of the intensity and relative propagation time of the waves and the bandwidth of the transmitted signal. The three most important effects of multipath propagation are: rapid changes in signal strength over a small travel distance or time interval, random frequency modulation due to varying Doppler shifts on different mutipath signals and time dispersion (echoes) caused by multipath propagation delays.

Doppler Shift: A typical wave component incident on a mobile station is shown in Fig. 5.1 below.

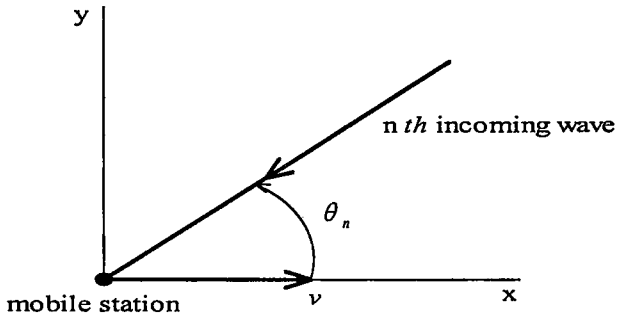


Figure 5.1 A typical wave component incident on a mobile station

The doppler shift is $f_{D,n} = f_m \cos\theta_n$, where $f_m = v/\lambda_c$ (λ_c is the carrier wavelength, v is the mobile station velocity).

5.1. Multipath Propagation

Consider the transmission of the band-pass signal $s(t) = \mathrm{Re}\{u(t)e^{j2\pi f_c t}\}$. At the receiver antenna, the n*th* plane wave arrives at angle θ_n and experiences Doppler shift $f_{D,n} = f_m \cos\theta_n$ and propagation delay τ_n. If there are N propagation paths, the received bandpass signal is

$$x(t) = \mathrm{Re}\left[\sum_{n=1}^{N} C_n e^{j2\pi[(f_c + f_{D,n})(t-\tau_n)]} u(t-\tau_n)\right]. \tag{5.1}$$

The received bandpass signal $x(t)$ has the form $x(t) = \mathrm{Re}\left[r(t)e^{j2\pi f_c t}\right]$ where the received envelope is

$$r(t) = \sum_{n=}^{N} C_n e^{-j\phi_n(t)} u(t-\tau_n) \text{ and } \phi_n(t) = 2\pi\left\{(f_c + f_{D,n})\tau_n - f_{D,n}t\right\}. \tag{5.2}$$

Flat fading: The channel can be modeled by a linear time-variant filter having the complex low-pass impulse response $g(t,\tau) = \sum_{n=1}^{N} C_n e^{-j\phi_n(t)} \delta(\tau - \tau_n)$. If the differential path delays $\tau_i - \tau_j$ are small compared to the duration of a modulated symbol, then the τ_n are all approximately equal to $\hat{\tau}$. The channel impulse response has the form

$$g(t,\tau) = \sum_{n=1}^{N} C_n e^{-j\phi_n(t)} \delta(\tau - \hat{\tau}) = g(t)\delta(\tau - \hat{\tau}). \tag{5.3}$$

The corresponding channel transfer function is obtained by taking the Fourier transform

$$T(t,f) = g(t)e^{-j2\pi f\hat{\tau}}. \tag{5.4}$$

Since the amplitude response is $|T(t,f)| = g(t)$, all frequency components in the received signal are subject to the same complex gain g(t).

Rayleigh Fading: Consider the transmission of an unmodulated carrier, $u(t) = 1$, such that $s(t) = \cos(2\pi f_c t)$. For flat fading channels the received band-pass signal has the quadrature representation

$$x(t) = g_I(t)\cos 2\pi f_c t - g_Q(t)\sin 2\pi f_c t \tag{5.5}$$

where $g_I(t) = \sum_n C_n \cos\phi_n(t)$ and $g_Q(t) = \sum_n C_n \sin\phi_n(t)$. By invoking the central limit theorem, $g_I(t)$ and $g_Q(t)$ are Gaussian random processes, i.e., at any time t, $g_I(t)$ and $g_Q(t)$ are Gaussian random variables. The complex enveope is $g_I(t) + jg_Q(t)$.

Rayleigh fading occurs when there is no line-of-sight or strong specular component in the received signal. That is there is no dominant C_n. In this case $g_I(t)$ and $g_Q(t)$ are zero-mean Gaussian random variables with variance b_0. The envelope of the received signal $\alpha(t) = |g(t)| = \sqrt{g_I^2(t) + g_Q^2(t)}$ is Rayleigh distributed at any time t, i.e.,

$$p_\alpha(x) = \frac{x}{b_0}\exp\left\{-\frac{x^2}{2b_0}\right\} = \frac{2x}{\Omega_p}\exp\left\{-\frac{x^2}{\Omega_p}\right\} \quad x \geq 0, \tag{5.6}$$

where $\Omega_p = E[\alpha^2] = 2b_0$ is the average envelope power.

5.2 Jakes' Sum of Sinusoids Fading Simulator

With N equal strength (C_n=1) arriving plane waves

$$g(t) = g_I(t) + jg_Q(t) = \sum_{n=1}^{N}\cos\left(2\pi f_m t\cos\theta_n + \hat{\phi}_n\right) + j\sum_{n=1}^{N}\sin\left(2\pi f_m t\cos\theta_n + \hat{\phi}_n\right) \tag{5.7}$$

where $\phi_n = -2\pi(f_c + f_m)\tau_n$. To approximate an isotropic scattering channel, assume that the N arriving plane waves are uniformly distributed in angle of incidence,

i.e., $\theta_n = 2\pi n / N$, n=1,2,...,N . Jakes' (Jakes,1974) imposes the constraints $-\hat{\phi}_i = \hat{\phi}_{N/2-1-i}$ and $\hat{\phi}_n = -\hat{\phi}_{-n}$ to give

$$g(t) = \sqrt{2}\left\{\left[2\sum_{n=1}^{M}\cos\beta_n \cos 2\pi f_n t + \sqrt{2}\cos\alpha\cos 2\pi f_m t\right]\right.$$
$$\left. + j\left[2\sum_{n=1}^{M}\sin\beta_n \cos 2\pi f_n t + \sqrt{2}\sin\alpha\cos 2\pi f_m t\right]\right\} \qquad (5.8)$$

where $\alpha = \hat{\phi}_N = -\hat{\phi}_{-N}$ $\beta_n = \hat{\phi}_n = -\hat{\phi}_{-n}$ $M = \frac{1}{2}\left(\frac{N}{2}-1\right)$

We choose β_n and α so that $g_I(t)$ and $g_Q(t)$ have zero-mean, equal variance, and zero cross-correlation. The choices $\alpha = 0$ and $\beta_n = \pi n / M$ will yield $\langle g_Q^2(t)\rangle = M$, $\langle g_I^2(t)\rangle = M+1$ and $\langle g_I(t) g_Q(t)\rangle = 0$.

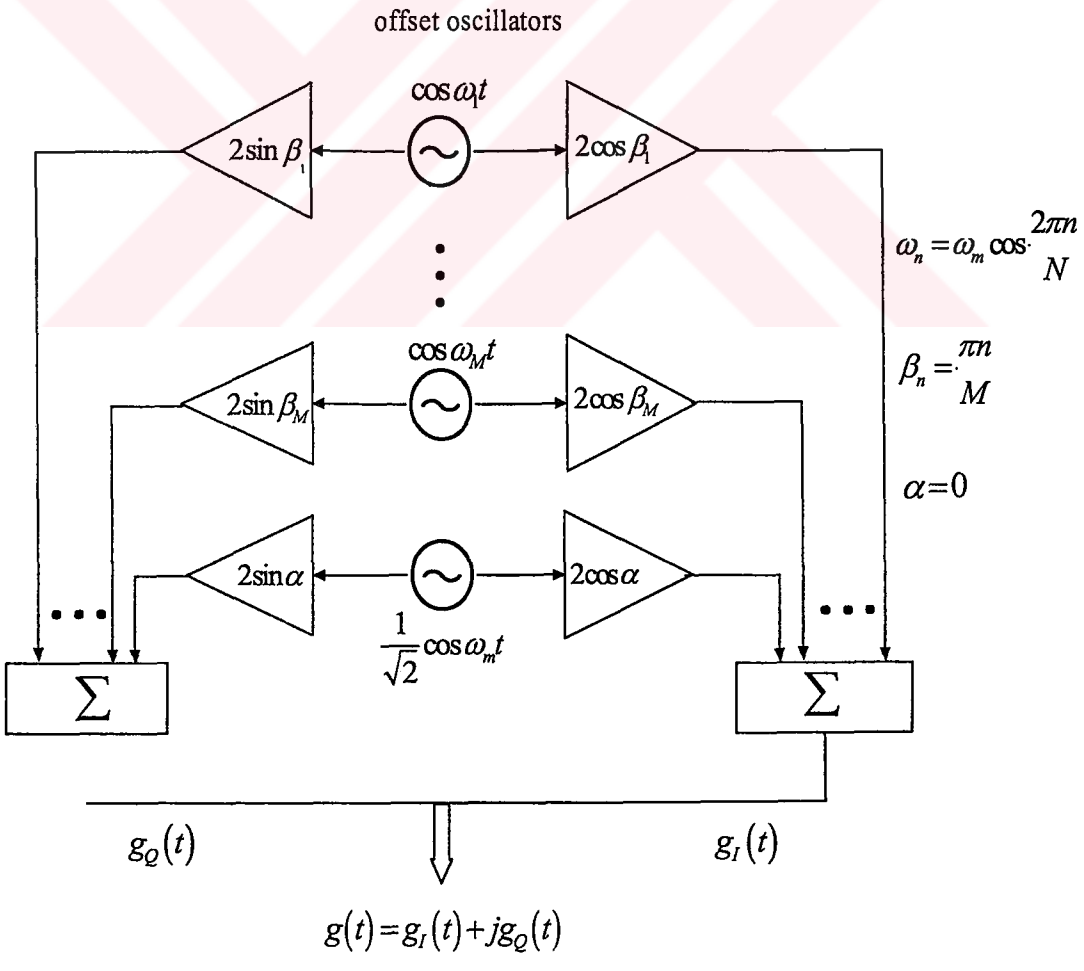
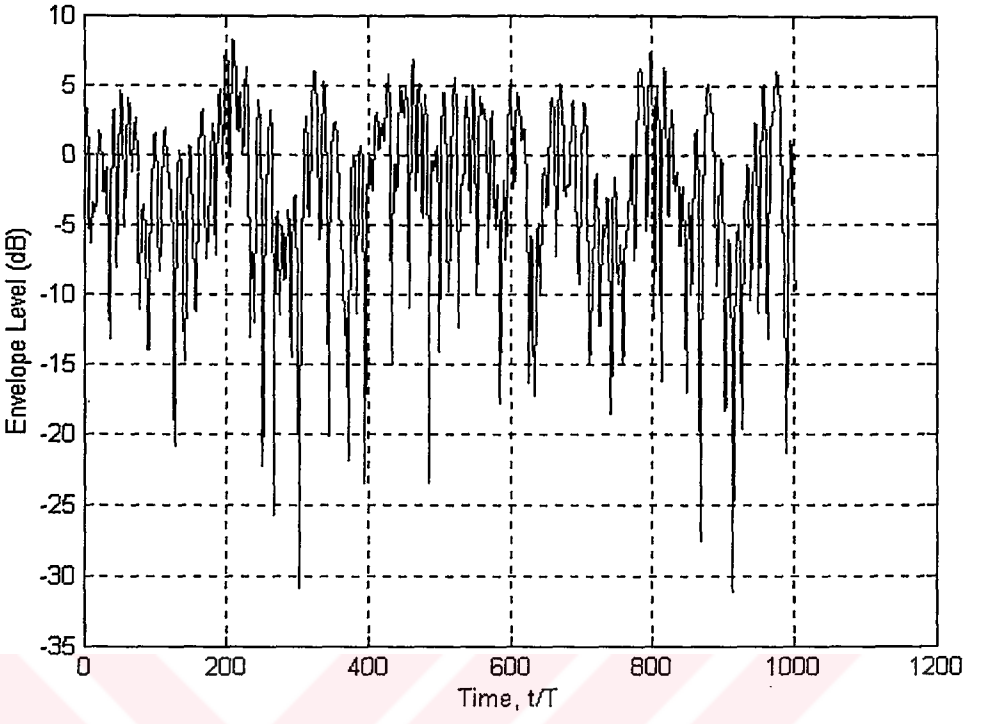
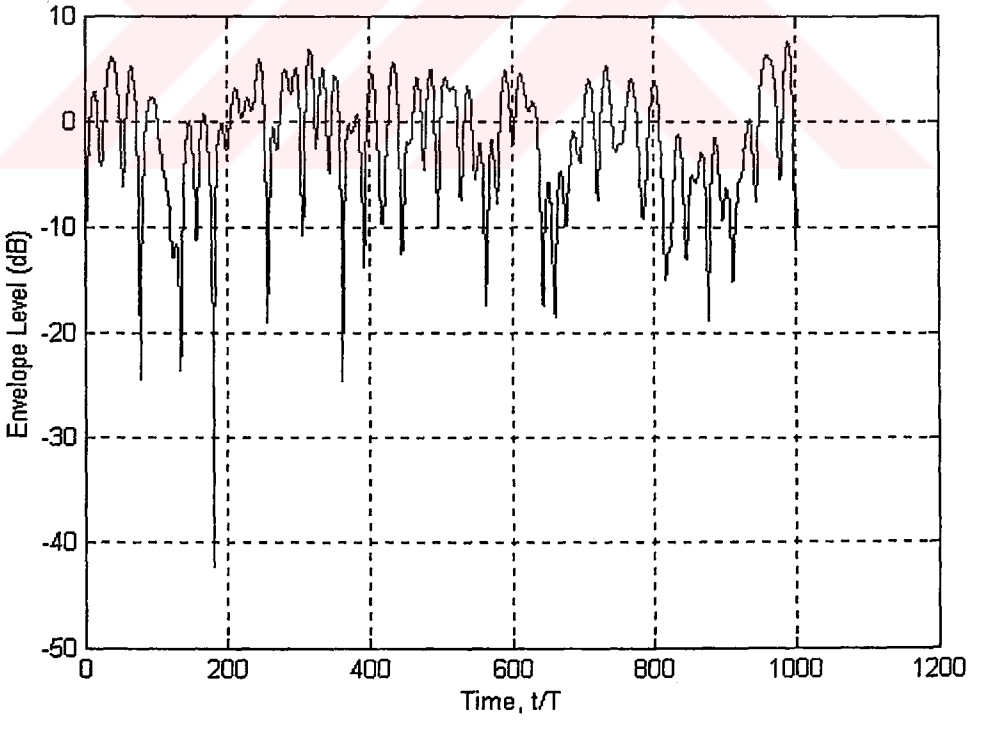


Figure 5.2. Jakes' Sum of Sinusoids Fading Simulator



(a)



(b)

Figure 5.3 Rayleigh faded envelope at f_c=900 MHz and T_s=1msn for different velocities a) v=70 km/hr b)35 km/hr

Delay Spread: Because the transmitted signal follows several paths with different length, there are multiple signals that arrive with additional delay at the receiver. In mobile radio environment a single symbol transmitted from one end and the receiver at the other end receives not only its own symbol but also echoes of its symbol. Therefore signals smear or spread out. The delayed waves are caused by local scatterers in near end region , the reflection of high-rise buildings and reflection from mountains. In a digital system that operates at high bit rate, because of the delay spread each symbol overlaps with preceeding and following symbols, and intersymbol interference occurs. Mean delay spread data is listed in Table 5.1

Type of Enironment	**Delay Spread**
In-building	< 0.1 μs
Open area	< 0.1 μs
Sub-urban area	0.5 μs
Urban area	3 μs

Table 5.1 Mean Delay Spread (Lee,1993)

Frequency Selective Fading: In mobile radio environment, as a result of high delay spread and data rate frequency selective fading occurs.As a result time varying intersymbol interference occurs. Frequency selectivity means, signals with different frequency will differently be affected by this type of channel.

Given the bandwidth B_s of the transmitted signal, the fading channels are classified into four categories depending on the two independent parameters, Doppler spread B_d which is proportional to Doppler frequency and coherence bandwidth B_c which is inversely proportional to delay spread,

1. Frequency-flat $(B_s << B_c)$; slow fading $(B_s >> B_d)$,
2. Frequency-flat $(B_s << B_c)$; fast fading $(B_s < B_d)$,
3. Frequency-selective $(B_s > B_c)$; slow fading $(B_s >> B_d)$,
4. Frequency-selective $(B_s > B_c)$; fast fading $(B_s < B_d)$.

5.3 Equalizers

Equalization techniques for combatting intersymbol interference (ISI) on bandlimited time dispersive channels may be subdivided into two general types: linear and nonlinear equalization. There is one or more structures associated with each type of equalizer. Furthermore, for each structure there is a class of algorithms that may be employed to adaptively adjust the equalizer parameters according to some specified performance criterion.

5.3.1 Linear Equalization Techniques

A linear equalizer may be implemented as a finite-duration impulse response (FIR) filter (also called a transversal filter) with adjustable coefficients. The adjustment of the equalizer coefficients is usually performed adaptively during the transmission of information by using the decisions at the output of the detector in forming the error signal for the adaptation, as shown in Fig 5.4. For symbol error rates below 10^{-2}, the occasional errors made by the detector have a negligible effect on the performance of the equalizer. During the start up period, a short known sequence of symbols is transmitted for the purpose of initial adjustment of the equalizer coefficients.

The criterion most commonly used in the optimization of the equalizer coefficients is the minimization of the mean square error (MSE) between the desired equalizer output and the actual equalizer output. The minimization of the MSE results in the optimum Wiener filter solution for the coefficient vector, which may be expressed

$$C_{opt} = \Gamma^{-1}\xi \tag{5.11}$$

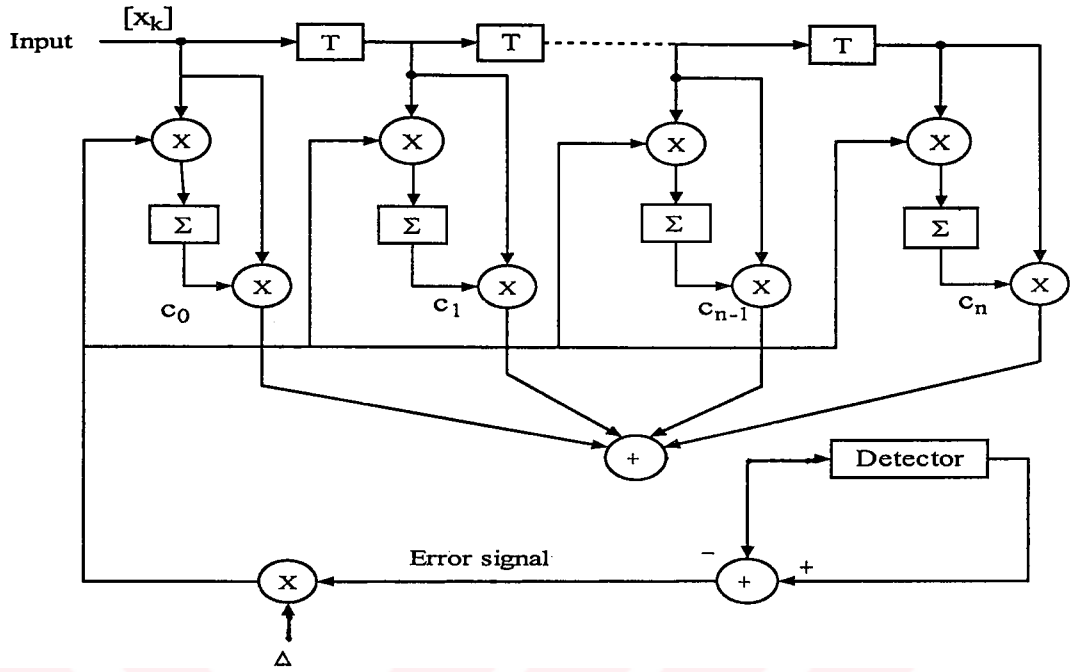


Figure 5.4. Adaptive linear FIR equalizer with LMS algorithm

where Γ is the autocorrelation matrix of the vector of the signal samples in the equalizer at any given time instant and ξ is the vector of cross correlations between the desired data symbol and the signal samples in the equalizer.

Alternatively, the minimization of the MSE may be accomplished recursively by the use of stochastic gradient algorithm introduced by Widrow, called the LMS algorithm. This algorithm is described by the coefficient update equation

$$C_{k+1} = C_k + \Delta e_k X_k^*, \quad k=0,1,... \tag{5.12}$$

where C_k is the vector of the equalizer coefficients at the kth iteration, X_k represents the signal vector for the signal samples stored in the FIR equalizer at the kth iteration, e_k is the error signal, which is defined as the difference between the kth transmitted symbol I_k and its corresponding estimate $\hat{I}_k$ at the output of the equalizer, and is the step size parameter that controls the rate of adjustment. The asterisk on X_k^* signifies the complex conjugate of X_k.

It's well known that the step size parameter Δ controls the rate of adaptation of the equalizer and the stability of the LMS algorihm. For stability, $0 < \Delta < 2/\lambda_{max}$, where λ_{max} is the largest eigenvalue of the signal covariance matrix. A choice of Δ just below the upper limit provides rapid convergence, but it also intoduces large fluctuations in the equalizer coefficients during steady-state operation. These fluctuations constitute a form of self-noise whose variance inceases with an increae in Δ. Consequently, the choice of Δ involves a trade-off between rapid convergence and the desire to keep the variance of the self-noise small.

5.3.2 Nonlinear Equalization Techniques

Nonlinear equalizers find use in applications where the channel distortion is too severe for a linear equalizer to handle. In particular, linear equalizer does not perform well on channels with spectral nulls in their frequency response characteristics. In an attempt to compensate for the channel distortion, the linear equalizer places a large gain in the vicinity of the spectral null and, as a consequence, significantly enhances the additive noise present in the received signal.

Three very effective nonlinear equalization methods have been developed over the past three decades. One is decision feedback equalization. The second is a symbol-by-symbol detection algorithm based on maximum a posteriori probabilty (MAP) criterion proposed by Bahl et. al. The third is a sequence detection algorithm, based on maximum-likelihood sequence estimation (MLSE) criterion, which is efficiently implemented by means of the Viterbi algorithm (VA).

5.3.2.1 Decision Feedback Equalizer (DFE)

The basic idea in the DFE is that once an information symbol has been detected, the ISI that it causes on future symbols may be estimated and subtracted out prior to symbol detection. The DFE may be realized either in the direct form or as a lattice. The direct form structure of the DFE is illustrated in Fig.5.5. It consists of a feedforward filter (FFF) and a feedback filter (FBF). The latter is driven by the

decisions of the output of the detector and its coefficients are adjusted to cancel the ISI on the current symbol that results from past detected symbols.

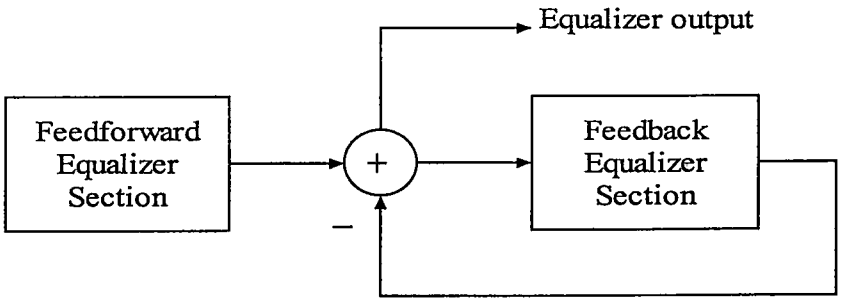


Figure 5.5 Generalized DFB equalizer

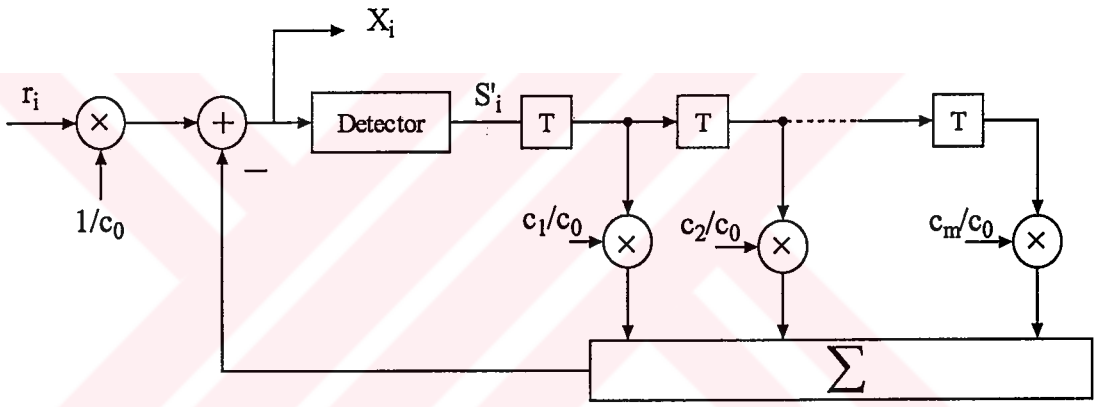


Figure 5.6 Decision feedback equalizer

Suppose that the sampled channel impulse response vector is

$$C = [c_0, c_1, \ldots, c_m]. \tag{5.13}$$

The received signal at time instant $t = iT$ is

$$r_i = \sum_{j=0}^{m} s_{i-j} c_j + n_i \qquad = s_i c_0 + \sum_{j=1}^{m} s_{i-j} c_j + n_i \tag{5.14}$$

where s_i is the transmitted symbol at time $t=i$, and n_i is the additive noise for the symbol s_i. If both sides of the equation are divided by c_0, which is the first impulse response of the channel, it becomes

$$r_i / c_0 = s_i + \sum_{j=1}^{m} s_{i-j} c_j / c_0 + n_i / c_0 \tag{5.15}$$

The output of the equalizer shown in Fig. 5.6. is

$$x_i = r_i / c_0 - \sum_{j=1}^{m} s'_{i-j} c_j / c_0 \tag{5.16}$$

Substituting equation 5.15 into 5.16, we have

$$x_i = s_i + \sum_{j=1}^{m} s_{i-j} c_j / c_0 + n_i / c_0 - \sum_{j=1}^{m} s'_{i-j} c_j / c_0 \tag{5.17}$$

If the decision is correct, which means $s_{i-j} = s'_{i-j}$, the equation (5.17) becomes

$$x_i = s_i + n_i / c_0 \ . \tag{5.18}$$

As a result, as long as correct decision is made, the DFE removes the ISI. In addition to correct decision the DFE requires the impulse responses of the channel. So a channel estimator must be used for identification of the channel impulse responses. Channel estimation can be accomplished adaptively as illustrated in Fig. 5.7. The channel estimator is usually an FIR tranversal filter with adjustable coefficients. The gradient LMS algorithm may be used to adjust the coefficients of the channel estimator. In transmission part a training sequence which is also known by the estimator should be added to every frame for adjustment of coefficients.

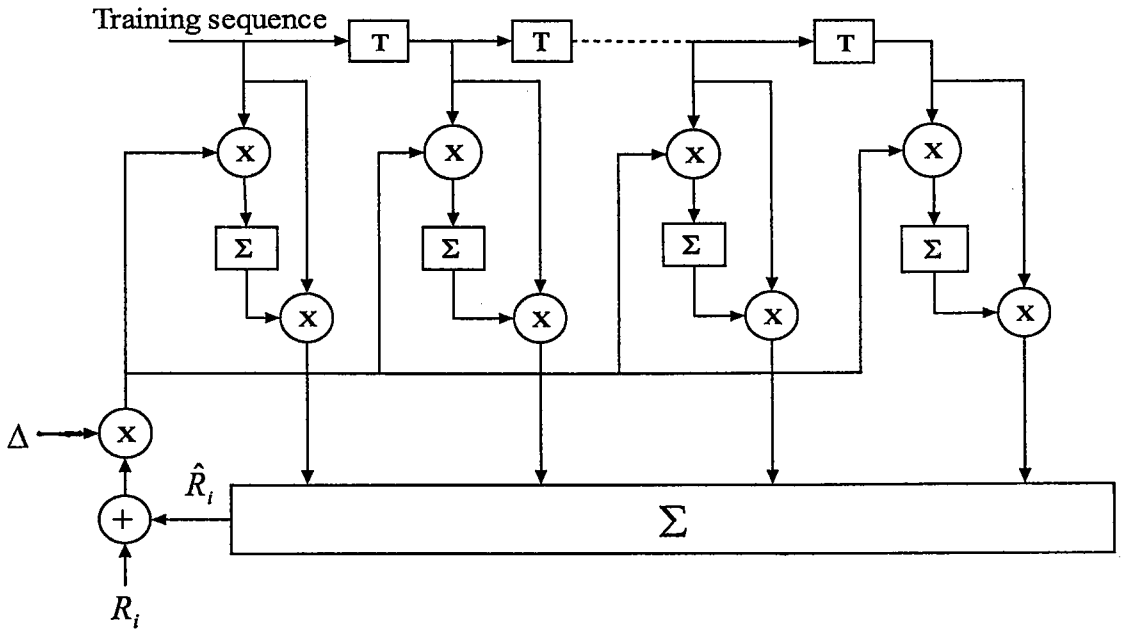


Figure 5.7. Adaptive channel estimation

CHAPTER 6
THE PERFORMANCE OF T-TCM IN A MOBILE RADIO CHANNEL

6.1 System Description

The diagram of the system that was used in order to find the performance of T-TCM 8 PSK in frequency selective Rayleigh fading channel is given in Fig. 6.1

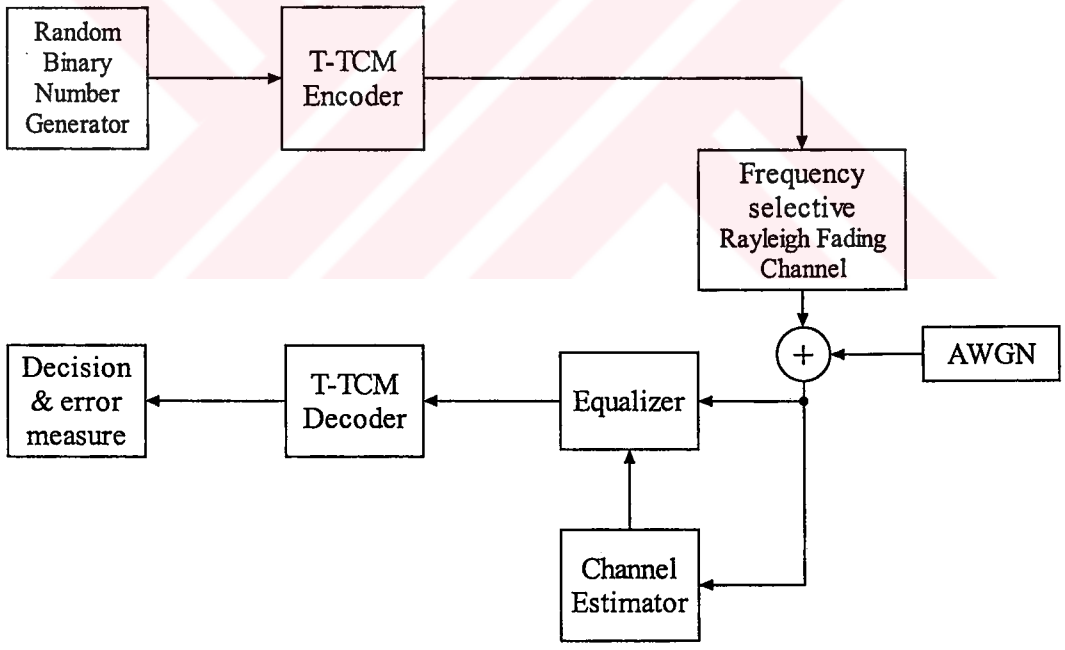


Figure 6.1 The complete system diagram

Although the main aim of this thesis was to investigate the performance of T-TCM in frequency selective Rayleigh fading channel the performance of the system is found for both AWGN and Rayleigh flat fading channels, too.

Random binary integers represent the source for the system. These binary integers are grouped in pairs to form the 2-bit symbols. Both the encoder and the decoder works in frame based manner and the length of the frames should be chosen as a multiple of the size of the interleavers used in the system. In simulations the size of the frames is the same as the size of the interleavers and chosen as 1000 and 500. The detailed structure of the encoder is given in Fig. 6.2

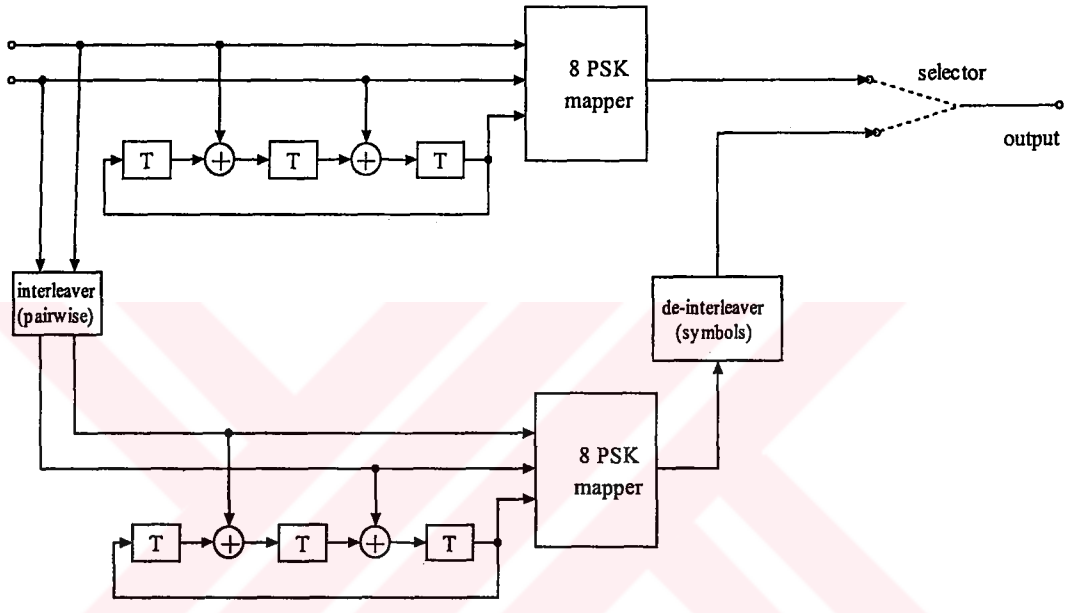


Figure 6.2 Encoder for 8-PSK with two dimensional comp. codes with memory 3.

The encoder used is Ungerboeck type, 8 state, 2/3 rated in control canonical structure, and given with the h polynomials as $h^{(0)} = (11)_8, h^{(1)} = (02)_8$, and $h^{(2)} = (04)_8$ with $d^2_{free} = 4.59$ and a asymptotic coding gain of 3.6 dB. In fact this encoder is equivalent to the 8 state TCM encoder given in chapter 4 (Proakis,1995). The trellis diagram of this encoder was given in that chapter. As in the binary turbo coding scheme, it's difficult to terminate the trellis of both of the encoders at the same time, so only the trellis of first encoder is terminated. The termination is made with the method described in (Divsalar&Pollara,1995).

Another important part in the encoder is the interleaver. The interleaver used in the decoder must have the same structure with the one used in the encoder. Pseudo-

random type interleaver is used. In (Robertson,1996), a necessity of odd-even constraint for the interleaver is mentioned with the reason as a spesific information bit pair to appear once and only once at the encoder output; this constraint isn't necessary since by the effect of the deinterleaver in the encoder, each information symbol corresponds to only one transmission symbol after puncturing. At the end of the encoder each coded symbol is mapped to one of the eight PSK symbols and transmitted. Natural mapping given in Table 6.1 is used as the mapping rule.

Signal levels	0	1	2	3	4	5	6	7
Natural mapping	000	001	010	011	100	101	110	111

Table 6.1 Natural mapping for 8-PSK

For a flat fading channel the channel used is a single path slow Rayleigh fading type and the coefficients of the channel response were found by using Jakes' Rayleigh fading simulator with 8 oscillators. The mobile receiver is supposed to have a velocity of 35 km/hr and a Doppler frequency, B of 25 Hz. A transmission rate T_s^{-1} of 1 ksymbols/sec is choosen so the Doppler bandwidth, symbol duration product, BT_s is 0.025. In receiver part the channel coefficients are required for cancelling the fading effect on the received symbols and in simulations it's assumed that these coefficients are known by the receiver. When the received symbol is multiplied by the complex conjugate of the channel coefficient, the fading effect can be eliminated.

For frequency selective Rayleigh fading channel the delay spread of the channel is assumed to be 3μ sec. with a bit rate of 500 kbit/sec. the channel is simulated as 4 delayed paths as shown in Fig. 6.3. In this model, each of four delayed paths have an independent Rayleigh distribution. The power ratio of the successive delayed paths is assumed to be 10 dB and the total power of four paths are made equal to one. Again the coefficients of the paths were found by using Jakes' Rayleigh fading simulator with 8 oscillators with a mobile velocity of 35 km/hr.

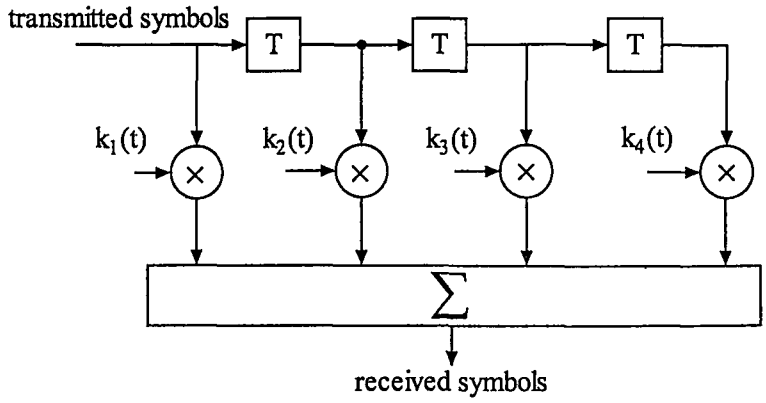


Figure 6.3 Discrete time model of frequency selective Rayleigh fading channel

6.2 Receiver Structure

The system is designed to have a frame-based TDMA structure and training symbols that are known by the receiver are added to each frame and these symbols are used for channel estimation. Each frame consists of 100 information symbols and 25 training symbols. These training symbols are separated in the receiver part used by the channel estimator. The estimator has the same structure given in Chapter 5. The algorithm used is least-mean-square which tries to minimize the mean square error. In order to equalize the ISI caused by the multipath channel a DFE is designed that has the same structure given in Chapter 5. The detector part in the equalizer makes a decision in favour of the one of the eight possible transmitted symbols that the received symbol is closest to. This can be done by finding the angle of received symbol in the signal space diagram and finding to which of the eight parts of the signal space it belongs to or by finding the metrics of received symbol with each of eight possible transmitted symbols and choosing the minimum one.

The decoder in the receiver part uses the output of the equalizer. The upper decoder is adjusted not to be at a * transition in the first decoding stage, so $P_r\{u_k = i\}$ is set to $1/2^2$. Both the first and second decoders work frame-based, with frame lengths 1000 and 500.

6.3 Simulation Results

6.3.1 AWGN Channel

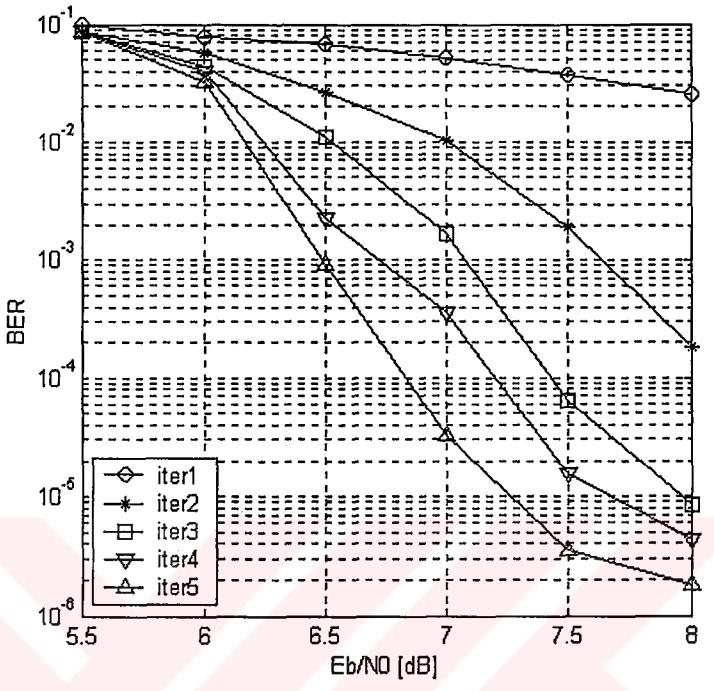


Figure 6.4 BER performance of T-TCM, N=1000, MAP, AWGN, 5 iterations

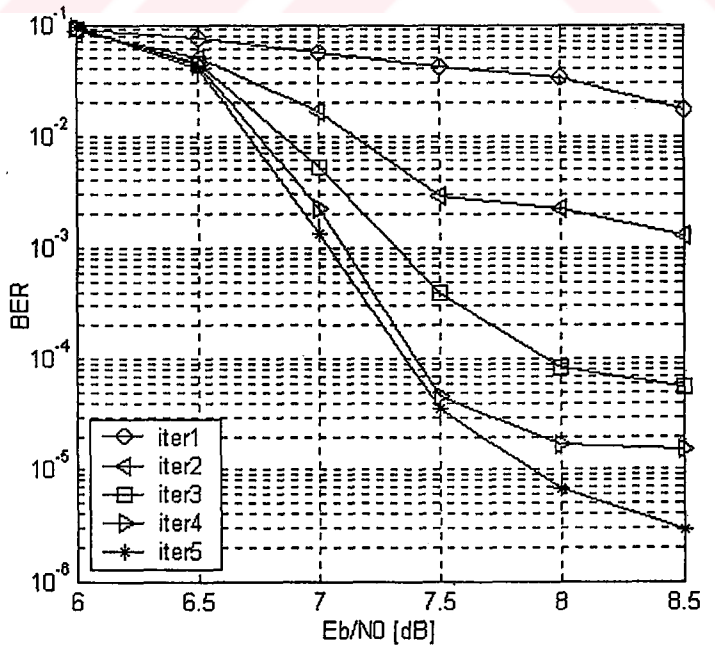


Figure 6.5 BER performance of T-TCM, N=1000, SOVA, AWGN, 5 iterations

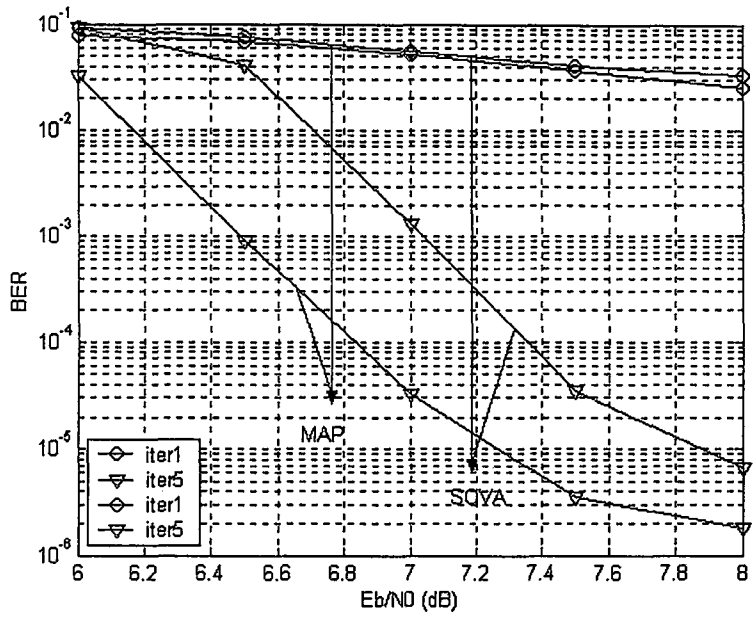


Figure 6.6 BER performance of T-TCM, N=1000, 5 iterations

From Fig.6.6 it can be seen that, at a BER level of 10^{-5} the performance of the MAP equivalent SOVA algorithm is, only 0,6 dB worse than the MAP algorithm for the AWGN channel. Since the SOVA algorithm is almost half less complex than MAP algorithm, this performance degradation is desirable.

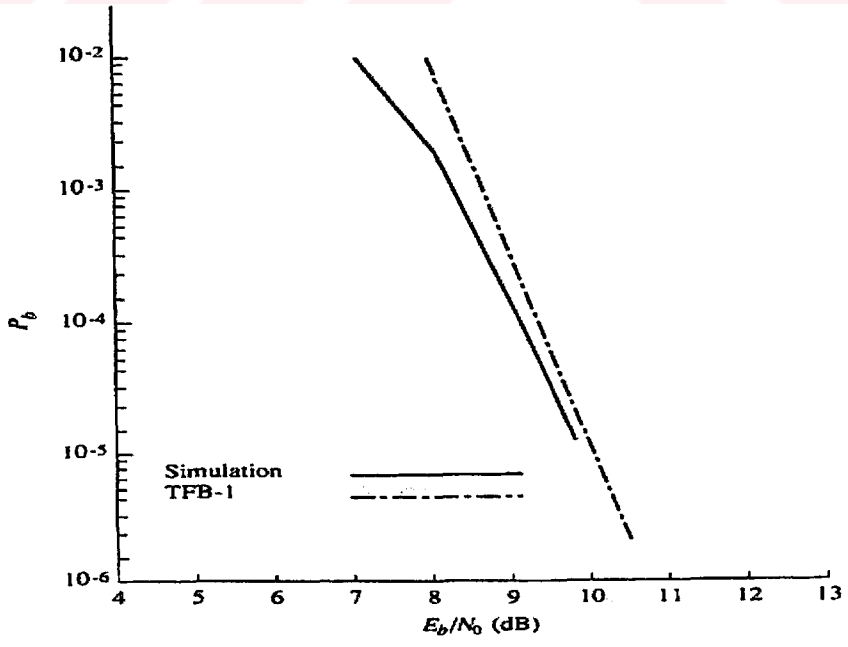


Figure 6.7 Bit error rate of 4 state, 8 PSK TCM scheme (Biglieri et.al.,1991)

When the performance of T-TCM is compared with the TCM using the figures Fig.6.6 and Fig.6.7, performance of T-TCM with MAP algorithm is approximately 2,5 dB and T-TCM with MAP equivalent SOVA algorithm is 1,9 dB better than the TCM at a BER level of 10^{-5}

6.3.2 Rayleigh flat fading (RFF) channel

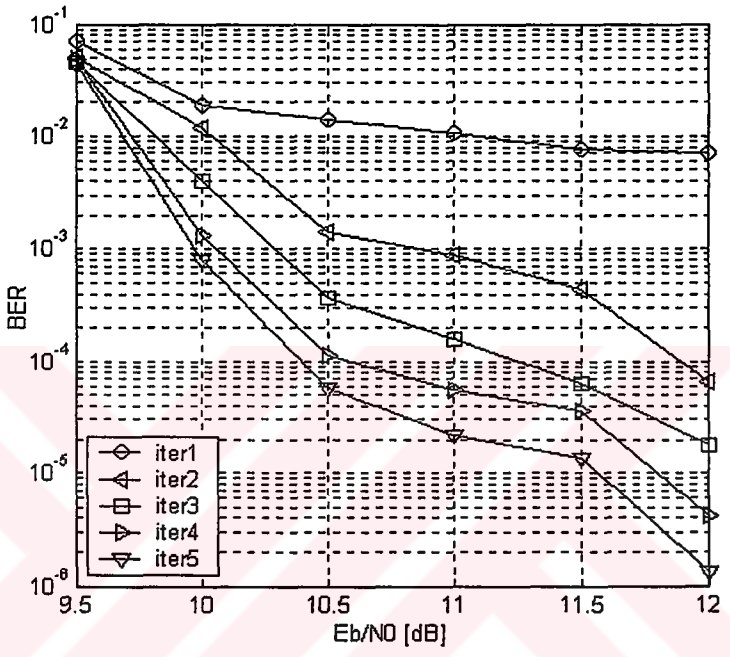


Figure 6.8 BER performance of T-TCM, N=1000, MAP, RFF, 5 iterations

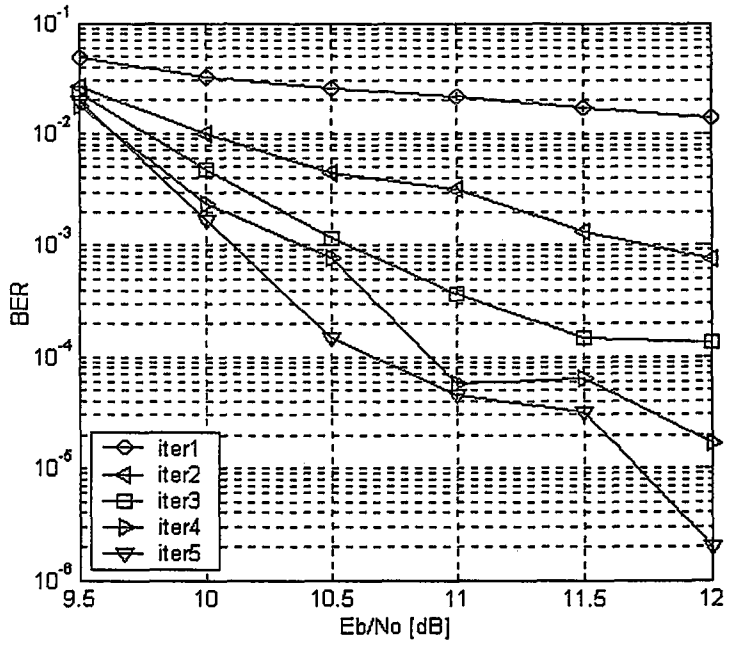


Figure 6.9 BER performance of T-TCM, N=1000, SOVA, RFF, 5 iterations

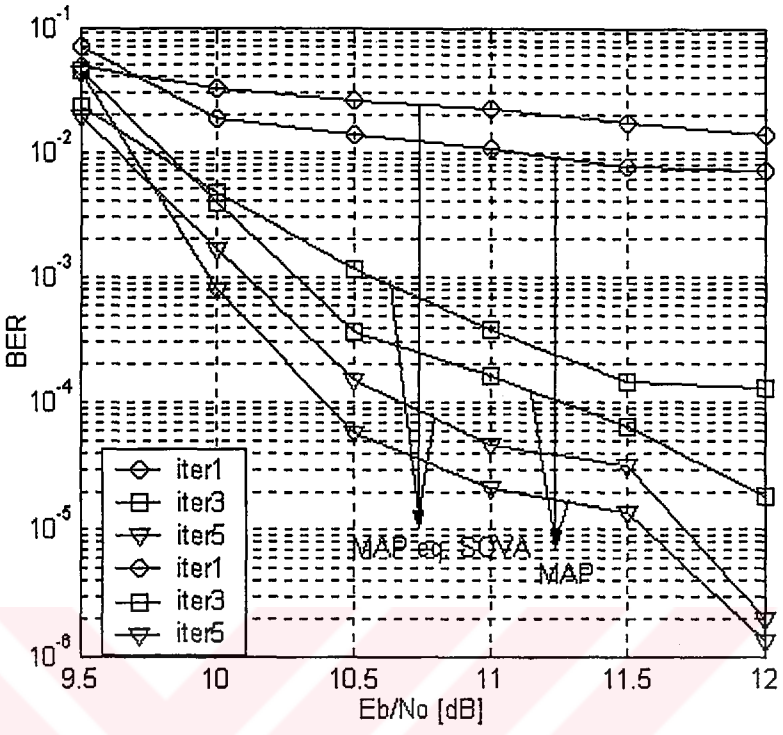


Figure 6.10 BER performance of T-TCM, N=1000, RFF, 5 iterations

From Fig.6.10 it can be seen that the performances of two decoding algorithms are much closer than in AWGN, such that, SOVA is within 0.15 dB of MAP at a BER of 10^{-5}. As mentioned before, for the T-TCM system in RFF channel, the CSI is assumed to be known by the receiver. These simulation results are also presented at (Yılmaz&Yılmaz,2002).

6.3.3 Frequency selective Rayleigh Fading (FSRF) channel

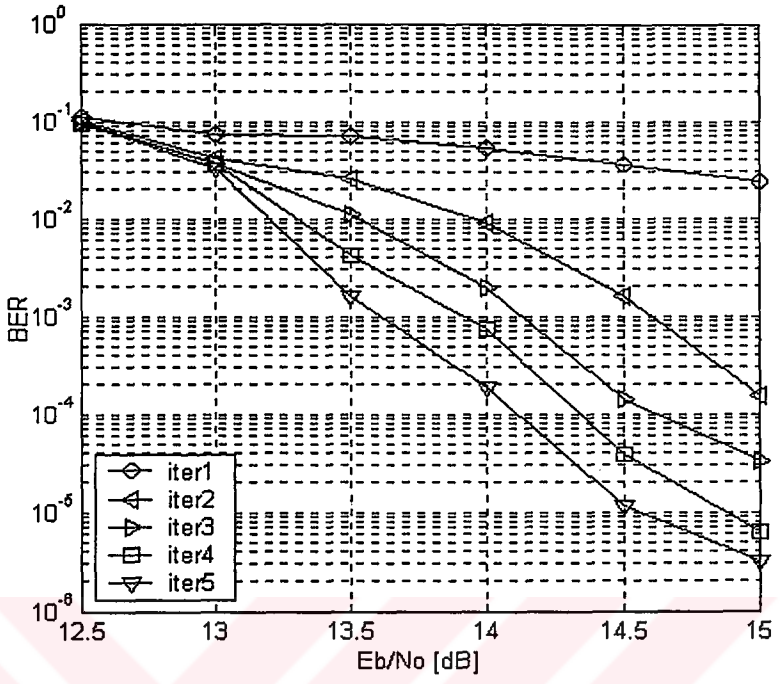


Figure 6.11 BER performance of T-TCM, N=1000, MAP, FSRF, 5 iterations

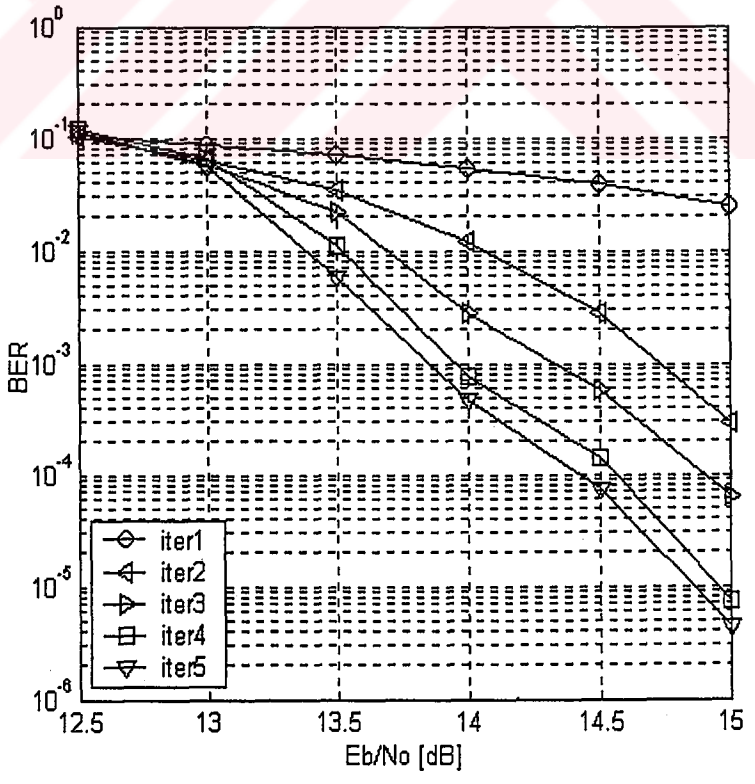


Figure 6.12 BER performance of T-TCM, N=1000, SOVA, FSRF, 5 iterations

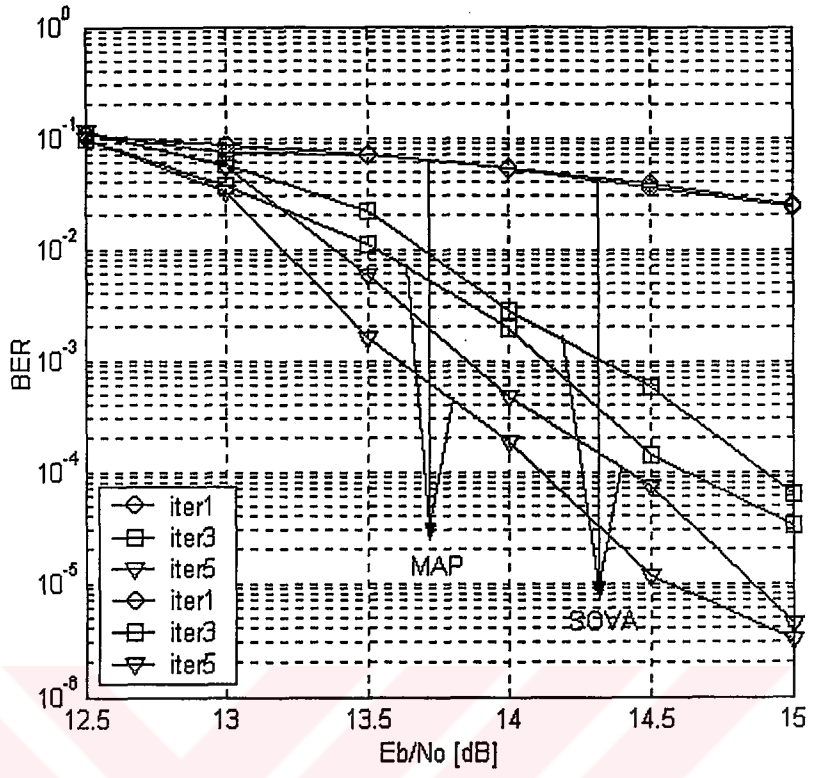


Figure 6.13 BER performance of T-TCM, N=1000, FSRF, 5 iterations

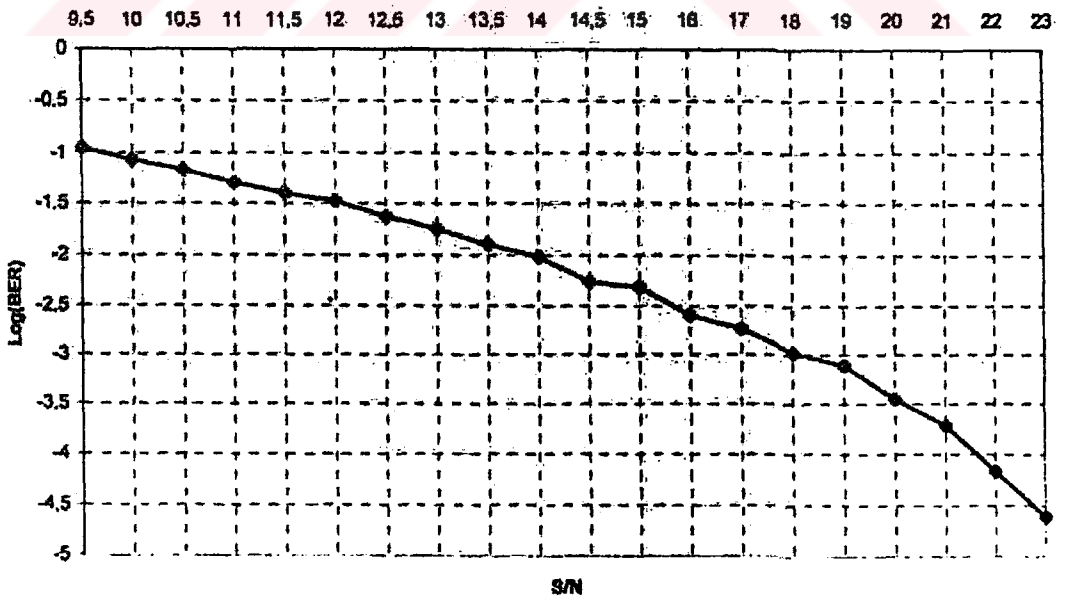


Figure 6.14 BER performance of trellis coded 8-PSK signal in a FSRF channel using DFE (Akgüner,1998)

Finally, the performance of T-TCM on a frequency-selective Rayleigh fading channel was obtained in Fig. 6.13. From simulation results, the difference in signal to noise ratio, at a BER of 10^{-5}, is about 0,3 dB. In order to compare the performance of the T-TCM system with that of TCM, the simulation results in (Akgüner,1998) are used. In that work the performance of trellis coded 8-PSK signal in a FSRF channel using DFE was investigated. From Fig.6.13 and Fig.6.14, , the T-TCM with MAP algorithm achieves approximately 7 dB and T-TCM with MAP equivalent SOVA algorithm achieves about 6,5 dB performance improvement in FSRF channel.

Also in order to see the effect of the frame length on the performance of the system, the frame size is decreased to 500. the simulation results are shown in Fig. 6.15

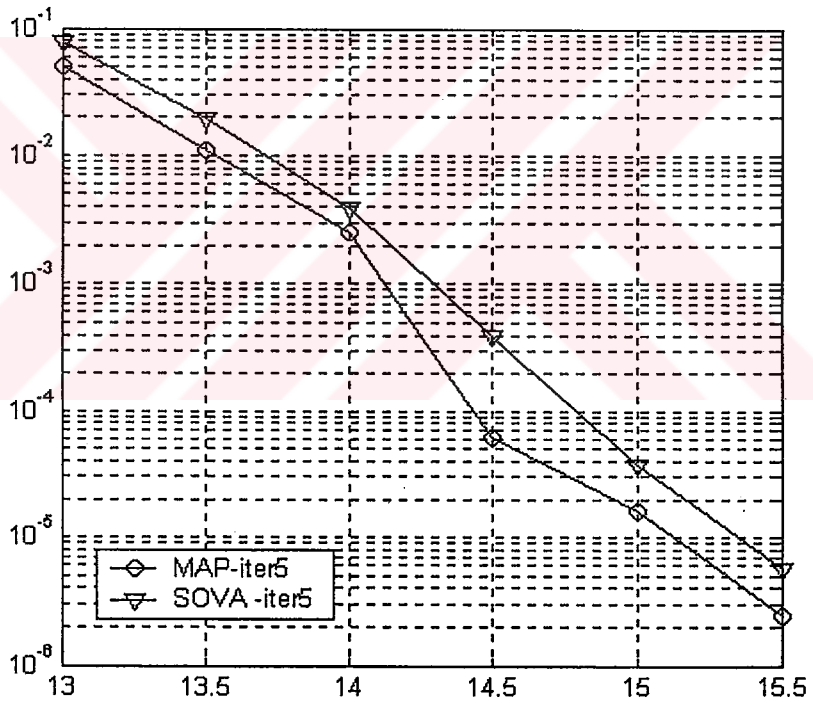


Figure 6.15 BER performance of T-TCM, N=500, FSRF, 5 iterations

When the frame size is reduced to 500 from 1000 in FSRF channel a 0,5 dB of performance degradation occurs. So it can be said that BER performance is proportional to frame size.

CHAPTER SEVEN
CONCLUSIONS

7.1 Conclusions

In this thesis, a coded modulation approach, Turbo Trellis Coded modulation, that can provide low error rate performance without bandwith expansion, was proposed. System performance was firstly investigated with AWGN present at the environment, a common situation in most communication systems, later in two different mobile radio environments, flat and frequency-selective Rayleigh fading channels, in order. Simulation results are compared with that of TCM, in the same environment conditions. Also two different decoding algorithms, MAP for non-binary codes, and a MAP equivalent SOVA algorithm are used. As the component code in the encoder Ungerboeck type, 8 state, rate 2/3 code is used, and 8 phase shift keying modulation is selected as modulation type. For reducing the effects of frequency-selective Rayleigh fading channel decision feedback equalization technique is used.

Simulation results showed that, at a BER level of 10^{-5}, the performance of the MAP equivalent SOVA algorithm is in AWGN, within 0,6 dB, in RFF channel within 0,2 dB, and in FSRF channel within 0,4 dB of the MAP algorithm. When the decoding complexities of the two algorithms are compared, almost half less complex SOVA algorithm is more suitable as a decoding algorithm with desirable small performance degradation. The simulation results under additive white Gaussian noise showed that performance of T-TCM is approximately 1,9~2,5 dB better than the TCM with MAP and MAP equivalent SOVA algorithm at a BER level of 10^{-5}. The

performance improvement in FSRF channel, is about 6,5~7 dB when compared with the TCM scheme with the same encoder type.

When the frame size is reduced to 500 from 1000 in FSRF channel a 0,5 dB of performance degradation occurs. So it can be said that BER performance is proportional to frame size. Finally, it can be said that T-TCM provides both low BER performance and bandwith efficiency in mobile radio environments like FSRF and RFF channels in the presence of AWGN.

7.2 Further Study

To design and select appropriate TCM codes is an important factor in the performance of T-TCM systems. Maximizing the effective free distance, using different types of mapping rules and trying different code structures are the key issues. TCM codes having parallel transitions in their trellis diagrams can be used as component codes. Also the effects of interleaver type on the T-TCM performance can be investigated. The odd-even constraint can be applied. The performance of other approaches for T-TCM other than the one used in this thesis should also be investigated.

Although the effect of frame length in T-TCM system is investigated for FSRF channel in this thesis, it can be done for AWGN and RFF channels, too. Perfect channel side information is assumed to be available in RFF channel, the use of channel estimators should also be considered. The type of the equalizer used in FSRF channel system will probably be a key factor in the system performance.

REFERENCES

Akgüner, F. (1998). The performance of trellis coded modulation in Rayleigh fading environment. M.Sc. Thesis, Dokuz Eylül University.

Bahl, F., Cocke, J. Jelinek, F. & Raviv, J.. (1974). Optimal decoding of linear codes for minimising symbol error rate. IEEE Transactions on Information Theory, IT-20, 284-287.

Benedetto S., Divsalar D., Montorsi G., &Pollara F.,(1996). Parallel concatenated trellis coded modulation. Proc.ICC, 974-978.

Berrou C.,& Glavieux A. (1993). Near Shannon Limit Error Correcting Coding and Decoding, Proc. ICC, 1064-1070.

Biglieri, E., Divsalar, D., McLane, P. J., & Simon. M. K., (1991). Introduction to Trellis-Coded Modulation with Applications, McMillan.

Bose, R. C., & Ray-Chaudhuri, D. K. (1960). On a Class of Error Correcting Binary Group Codes. Inform. Control, 3 68-79.

Divsalar, D., & Simon, M. K. (1987). Trellis Coded Modulation for 4800-9600 bits/sTransmission over a Fading Mobile Satellite Channel. IEEE Transactions on Communications, SAC-5, 162-175.

Divsalar, D., & Simon, M. K. (1988a). Multiple Trellis Coded Modulation (MTCM), IEEE Transactions on Communications, 36, 410-419.

Divsalar, D., and Simon, M. K., (1988b). The Design of Trellis Coded MPSK for Fading Channels: Performance Criteria. IEEE Transactions on Communications, 36, 1004-1012.

D.Divsalar,and F.Pollara. (1995). On the design of turbo codes. JPL TDA Progress Report, 42, 99-120.

Dunscombe E., & F.C.Piper. (1989). Optimal interleaving scheme for convolutional codes. Electronics Letters, 25, 1517-1518.

Forney, G. D. (1973). The Viterbi Algorithm," Proc. of the IEEE, 61, 268-278.

Fossorier, M. P. C., Burkert, F., Lin, S., & Hagenauer, J. (1998). On the Equivalence Between SOVA and Max-Log–MAP Decodings. IEEE Comm. Letters, 2, 137-139.

Golay, M. J. E. (1949). Note on Digital Coding. Proc. IRE, 37, 657.

Goff, L., Berrou, C., & Glavieux, A., (1994). Turbo Codes and High Spectral Efficiency Modulation, Proc. ICC'94, 645-649.

Hagenauer, J., & Hoeher, P., (1989). A Viterbi with Soft-Decision Outputs and its Applications, IEEE Globecom'89 Conf. Rec, 3, 1680-1686.

Hagenauer, J., Robertson, P., & Papke, L., (1994). Iterative (Turbo) Decoding of Systematic Convolutional Codes with the MAP and SOVA Algorithms, in Proc. ITG Conf., 130, 21-29.

Hagenauer, J. (1996). Iterative decoding of binary block and convolutional codes. IEEE Transactions on Information Theory, 42, 429-445.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes. Bell Syst. Tech. J. 29, 147-160

Jakes, W. C. (1974). Microwave mobile communications. New York, John Wiley & Sons.

Muller, D. E. (1954). Application of Boolean Algebra to Switching Circuit Design and to Error Detection. IRE Trans. Electronic Comput., EC-3, 6-12

Papke L., Robertson P., & Villebrum, E. (1996). Improved decoding with the SOVA in parallel concatenated (turbo code)scheme. Proceedings,International Conference on Communications, 102-106.

Proakis, G., (1995). Digital Communications, Third edition, McGraw-Hill Inc.

Robertson, P., Villebrun, E., & Hoeher, P. (1995). A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," Proc. ICC, 1009-1013.

Robertson, P., Woerz T. (1996). A novel bandwith efficient coding scheme employing turbo code. Proc. ICC, 962-967.

Robertson, P., Hoeher, P.,& Villebrum, E. (1997). Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding. European Transactions on Communications, 8, 119-125.

Robertson, P., & Woerz, T. (1998). Bandwidth efficient turbo trellis-coded modulation using punctured component codes. IEEE Journal on Selected Areas of Communications, 16, 206-218.

Reed, I. S. (1954). A Class of Multiple Error Correcting Codes and the Decoding Scheme. IRE Trans. Inform., IT-4, 38-49

Reed, I. S., & Solomon, G. (1960). Polynomial Codes over Certain Finite Fields. SIAM J., 8, 300-304

Shannon, C.E. (1948). A mathematical theory of communication. Bell Systems Technical Journal, 27, 379-423, 623-656.

Tan, J., Stüber G. L. (2000). A MAP equivalent SOVA for Non-binary Turbo Codes. IEEE ICC, 602-606.

Ungerboeck, G., (1982). "Channel Coding with Multilevel/Phase Signals". IEEE Transactions on Information Theory, IT-28, 55-67,

Ungerboeck, G., (1987a). Trellis coded modulation with redundant signal sets part I: Introduction, IEEE Communication. Magazine, 25, 5-11.

Ungerboeck, G., (1987b). "Trellis-Coded Modulation with Redundant Signal Sets; Part II: State of the Art" , IEEE Communication. Magazine, 25, 13-21.

Valenti, M.C. (1999). Iterative detection and decoding for wireless communications. PhD thesis, Virginia Polytechnic Institute and State University.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymtotically optimum decoding algorithm. IEEE Transactions on Information Theory , IT-13, 260-269.

Viterbi, A. (1998). An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes. IEEE Journal on Selected Areas of Communications, 260-264.

Wicker, B., (1995). Error Control Systems for Digital Communication and Storage, Prentice-Hall,Inc.

Yılmaz, M. & Yılmaz, R. (2002) Turbo kafes kodlanmış evre kaydırmalı kiplenim sisteminin Rayleigh düz sönümlü kanalda başarımı. SİU 2002, 605-610