Serially Concatenated Coded Continuous Phase Modulation for Aeronautical Telemetry

Kanagaraj Damodaran

Submitted to the Department of Electrical Engineering &
Computer Science and the Faculty of the Graduate School
of the University of Kansas in partial fulfillment of
the requirements for the degree of Master of Science

Thesis Committee:

Dr. Erik Perrins: Chairperson

Dr. Victor Frost

Dr. James Roberts

Date Defended

2008/08/21

The Thesis Committee for Kanagaraj Damodaran certifies
that this is the approved version of the following thesis:

Serially Concatenated Coded Continuous Phase Modulation for Aeronautical Telemetry

Committee:

____
Chairperson

____

____

____
Date Approved

To my mom and dad

Acknowledgements

I would like to acknowledge and thank people who have supported me in this thesis. I thank Dr. Perrins, my advisor for his valuable guidance and inputs all through my thesis. I would also like to thank Dr. Frost and Dr. Roberts for being on my thesis committee and reviewing this thesis document. I would like to thank the department of Electrical Engineering and Computer Science, and Information and Telecommunication Technology Center, at The University of Kansas for all its support.

I would like to thank the Test Resource Management Center (TRMC) Test and Evaluation/Science and Technology (T&E/S&T) Program for their support. This work was funded by the T&E/S&T Program through the White Sands Contracting Office, contract number W9124Q-06-P-0337.

I would like to thank my mom and dad for their unconditional love and affection. They have been my source of strength and encouragement in every phase of my life. I would also like to thank my Uncle Shanmugam and Aunt Lalitha for their emotional support to my family. I thank my Aunt Selvi and Uncle Kumaravelu for their love and prayers.

I thank all my friends here in Lawrence, KS, and in Chennai, India, for the fun and support I have had all my life.

Abstract

This thesis treats the development of bandwidth-efficient serially concatenated coded (SCC) continuous phase modulation (CPM) techniques for aeronautical telemetry. The concatenated code consists of an inner and an outer code, separated by an interleaver in most configurations, and is decoded using relatively simple near-optimum iterative decoding algorithms. CPM waveforms such as shaped-offset quadrature phase shift keying (SOQPSK) and pulse code modulation/frequency modulation (PCM/FM), which are currently used in military satellite and aeronautical telemetry standards, can be viewed as inner codes due to their recursive nature. For the outer codes, this thesis applies serially concatenated convolutional codes (SCCC), turbo-product codes (TPC) and repeat-accumulate codes (RAC) because of their large coding gains, high code rates, and because their decoding algorithms are readily implemented. High-rate codes are of special interest in aeronautical telemetry applications due to recent reductions in available spectrum and ever-increasing demands on data rates. This thesis evaluates the proposed coding schemes with a large set of numerical simulation results and makes a number of recommendations based on these results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The primary objective of any digital communication system is to effectively transmit information over a channel, while efficiently utilizing power, bandwidth and complexity. For this to be done, the selected modulation scheme must match the channel characteristics. Moreover, efficiency of data transmission is increased with well-chosen combinations of channel coding and modulation techniques. The introduction of *turbo codes* in 1993 [1] led to a flurry of research effort in parallel concatenated convolutional codes (PCCC) separated by a random interleaver and decoded iteratively. Turbo codes yield bit error rates (BER) around 10^{-5} [1] for even code rates well beyond the channel cutoff rate.

Another equally powerful code configuration with comparable performance to turbo codes is serially concatenated convolutional codes (SCCC) separated by a random interleaver and decoded iteratively [2]. Although the use of channel codes provides protection against errors introduced by the channel and increases the power efficiency of data transmission, their use also reduces the bandwidth efficiency of the overall communication system.

In recent years, bandwidth efficiency has become a major concern in aeronautical telemetry. PCM/FM (pulse code modulation/frequency modulation), which is a rather spectrally *inefficient* modulation, has been the dominant carrier for aeronautical telemetry since the 1970s. Spectrum reallocations of frequency bands in 1997 prompted a migration away from PCM/FM and gave rise to the Advanced Range Telemetry Modulation (ARTM)-CPM program [3]. Size, weight, and power supply constraints forced the use of fully saturated, nonlinear RF power amplifiers. As a consequence, the search for more bandwidth efficient waveforms was limited to constant envelope waveforms, in particular, continuous phase modulations (CPMs). By 2004, a pair of interoperable waveforms were adopted in the IRIG 106 standard as “ARTM Tier I” modulations [4]. The first is a version of Feher-patented QPSK (FQPSK) [5], which is a licensed technology. The second is a version of shaped offset quadrature phase shift keying, known as “SOQPSK-TG” [6], which is an unlicensed technology that has also been used in military satellite communication standards [7]. This waveform uses a custom frequency pulse shape, developed by the telemetry group (TG), and hence the name ”SOQPSK-TG”. These waveforms achieve twice the spectral efficiency of PCM/FM even when nonlinear amplifiers are used [8].

This thesis treats the development of bandwidth-efficient serially concatenated coded (SCC) techniques for PCM/FM and SOQPSK-TG. Forward error correction (FEC) schemes for aeronautical telemetry have received only preliminary attention to date. The only published results on this subject are found in [9], which discussed a combination of turbo-product codes (TPCs) with PCM/FM and SOQPSK-TG using a non-CPM based *ad hoc* approach.

This thesis develops SCC schemes for aeronautical telemetry that take full advantage of the fact that PCM/FM and SOQPSK-TG are *recursive modulations* and can be treated as inner codes in SCC schemes [10–14]. In particular, it develops high-rate

SCCC schemes and TPC schemes. The performance of repeat-accumulate codes (RAC) with SOQPSK-TG and PCM/FM is also studied in this thesis. This thesis also develops *coherent* and *noncoherent* soft-input soft-output (SISO) demodulators for use with these codes in an iterative demodulation and decoding architecture. Finally, a large set of numerical simulation results is presented which compares the resulting SCC schemes based on several important factors such as, 1) SOQPSK-TG vs. PCM/FM, 2) coherent demodulation vs. noncoherent demodulation, 3) SCCC vs. TPC and so on. These numerical results indicate that SOQPSK-TG is an excellent choice due to its spectrum efficiency advantage over PCM/FM and due to its large coding gains. These results also show that SCCCs yield larger coding gains than TPCs and RACs. Additonally these results show that noncoherent demodulation offers attractive performance in light of its simplified synchronization requirements.

This thesis is organized as follows. Chapter 2 explains the FEC schemes considered in this thesis. This chapter also presents an overview of the algorithms used to encode and decode the FEC schemes considered in this thesis. Chapter 2 also provides the BER performance of these FEC schemes over the additive white gaussian noise (AWGN) channel. Chapter 3 explains SCC systems in general and provides a brief discussion on SOQPSK-TG and PCM/FM. In addition to these general discussions, Chapter 3 explains in detail the various serially concatenated coded CPM (SCC-CPM) systems developed in thesis. Chapter 4 provides the BER performance results of the various SCC-CPMs considered in this thesis. In addition to the performance results presented here, Chapter 4 lists several important comparisons and gives recommendations based upon these comparisons. Finally, Chapter 5 gives a few important concluding remarks and gives direction for the future work that is to be carried out based on the results provided in this thesis. This is followed by an appendix where additional BER performance results of SCC-CPMs under noncoherent demodulation are presented.

Chapter 2

Error Control Coding

In the context of digital communication, the history of error control coding dates back to the middle of the twentieth century. However, in recent years there has been a tremendous improvement in performance, with channel codes closely approximating channel capacity. *Error Correction Coding* is instrumental in correcting errors introduced into the transmitted signal whereas *Error Detection Coding* only detects errors based on the received signal. Both of these coding formats have differing advantages in different applications and they are collectively termed as *Error Control Coding*. Coding schemes are omnipresent in the modern information-based era with CD-ROM's, hard-disk, phone calls made over a digital cellular phone, packets transmitted over the Internet, etc. All of these examples employ some form of error control coding to protect data.

In 1948, Shannon [15] demonstrated that the errors introduced by the noisy channel can be avoided by proper encoding of information without sacrificing the rate of transmission. Since then, much work has been carried out to improve encoding and decoding efficiencies and to improve the reliability of modern digital communication systems. A typical digital transmission system can be represented by a block diagram

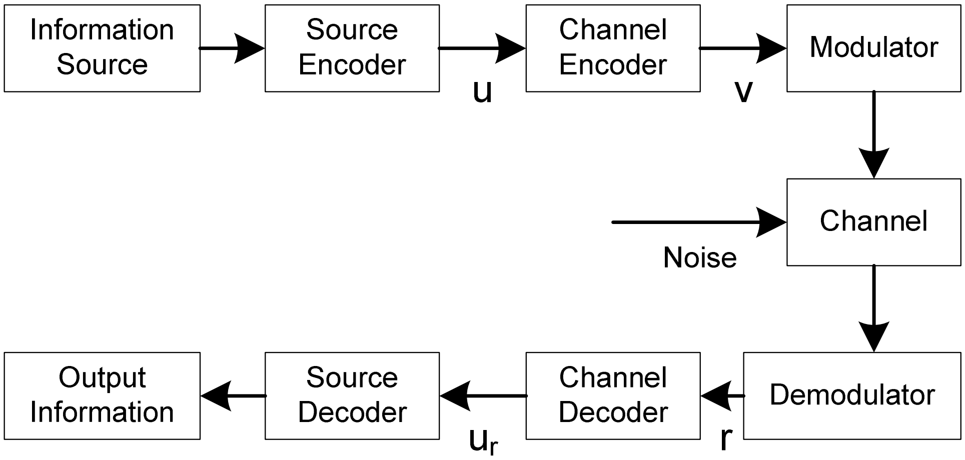

Figure 2.1. Block Diagram of a Typical Digital Transmission System.

such as the one shown in Figure 2.1. The information source can either produce a continuous waveform or a sequence of discrete symbols, which are converted into binary digits by a source encoder. Source encoding by itself is an important concept and is discussed in great depth in [16, 17]. Next, the channel encoder converts the source coded sequence u into a discrete encoded sequence v, which is called a *codeword*. The discrete sequence might not be suitable for transmission over a noisy channel and hence is suitably modulated and transmitted over a channel, which introduces noise into the transmitted signal.

The demodulator demodulates each received waveform and produces a discrete sequence r that corresponds to the encoded sequence v. An appropriate channel decoder then converts the received sequence r into an estimated information sequence u_d. The type of the channel decoder mainly depends upon the type of the channel encoder and the noisy characteristics of the channel. A suitable source decoder transforms u_d into an output sequence which is delivered at the destination.

A basic communications problem that is addressed via channel coding is to efficiently transmit information over a noisy environment. This chapter gives a descriptive overview of certain important channel coding techniques like convolutional codes (CC), turbo-product codes (TPC) and repeat-accumulate codes (RAC), which this thesis utilizes in developing a SCC system.

2.1 Convolutional Codes

Convolutional codes (CC) were first introduced by Elias [18] in 1955 as an alternative to block codes. Shortly thereafter extensive research was carried out involving CCs. However, it was in 1967 that Viterbi proposed a maximum likelihood (ML) decoding algorithm that was a relatively simple soft-decision decoding algorithm for CCs. With the introduction of this decoding algorithm, CCs found widespread applications in deep-space and satellite communication systems. The idea of concatenating CCs began when Gottfried Ungerboeck, in his classic 1982 [19] paper, showed that efficient performance can be obtained by combining modulation and coding. With the introduction of turbo codes in 1993 [1], research interest was kindled towards concatenated coding schemes separated by an interleaver and decoded using a near-optimum iterative decoding algorithm.

A rate $R = u/n$ CC with a memory m takes in u information bits as its input and produces n coded bits at the output. An important difference between CCs and block codes is that the former introduces memory into the encoder unit. CCs also differ from block codes in that they achieve large minimum distances and low error probabilities by increasing the memory m associated with the codes, rather than by increasing u and n. This chapter describes the encoding procedure for CCs and overviews two different decoding algorithms.

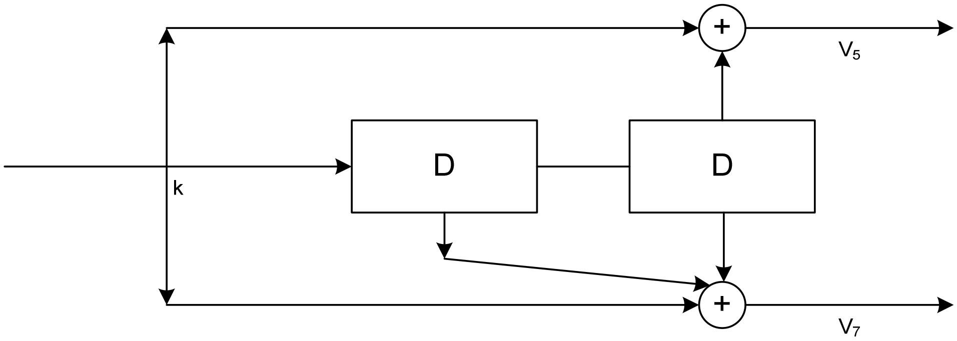

Figure 2.2. A Simple (5,7) Convolutional Encoder.

2.1.1 Encoding

A convolutional encoder may be viewed as nothing more than a set of digital filters whose overall output is an interleaved sequence of the internal filter outputs. In general, a code sequence is generated by passing an information sequence through a linear finite-state shift register. These shift registers have m stages that indicate the memory associated with the code and are usually termed as the constraint length k of the CC. Similar to block codes, a CC can be described by its generator matrix but an alternative representation uses vectors to describe a CC. With n output codeword bits corresponding to u input bits, we specify n vectors where each vector represents a modulo-2 adder. A 1 in the ith position of a vector indicates that the corresponding memory element is connected to a modulo-2 adder, whereas a 0 indicates no connection between the adder and ith memory element. These vectors describe CCs effectively and they are termed as generator polynomials. These generator polynomials with binary 1's and 0's can also be conveniently represented by a simple octal notation [20–22].

To better understand the functionality of convolutional codes let us consider a simple binary convolutional encoder shown in Figure 2.2. For every bit u input into the encoder it produces two output bits, v_5 and v_7, and hence the rate of this encoder is $R = u/n = 1/2$. Since there are two memory elements D, the constraint length of this encoder is given by $k = 2$. Hence every input bit is retained for two bit times and influences next two output bits. While the 2 bit time memory associated with the encoder contributes to the performance of this code, any increase in the constraint length (or bit time memory), increases the associated performance. As seen in Figure 2.2, there are $n = 2$ modulo-2 adders corresponding to v_5 and v_7 output bits. A modulo-2 adder which represents v_5 output takes as its input, a new bit and a bit from the second memory element. Hence a generator polynomial which represents this adder has a corresponding 1 in first and third position, whereas it has a 0 in the second position which corresponds to first memory element which is not connected to the adder. This generator polynomial in binary format can be given as

$$g_1 = [101].$$

Similarly the other generator polynomial which represents modulo-2 adder connected to v_7 output bit is represented by g_2 and it has a 1 in all places since both the memory elements and the new input bit is connected to this adder. Binary representation of g_2 is given as

$$g_2 = [111].$$

These generator polynomials can also be represented conveniently in a simple octal notation corresponding to the binary representation of a polynomial. Hence (g_1, g_2) convolutional encoder can simply be written as (5,7) convolutional encoder.

In addition to the basic (5,7) convolutional encoder, this thesis also uses a encoder which is similar to the (5,7) encoder in the sense that it takes $u = 1$ bit input and produces a $n = 2$ bit output. Hence the rate of this other encoder is also $R = u/n = 1/2$. The constraint length of this other encoder is $k = 4$, which means it has 4 memory elements and so the current input bit influences the next 4 output bits. As specified

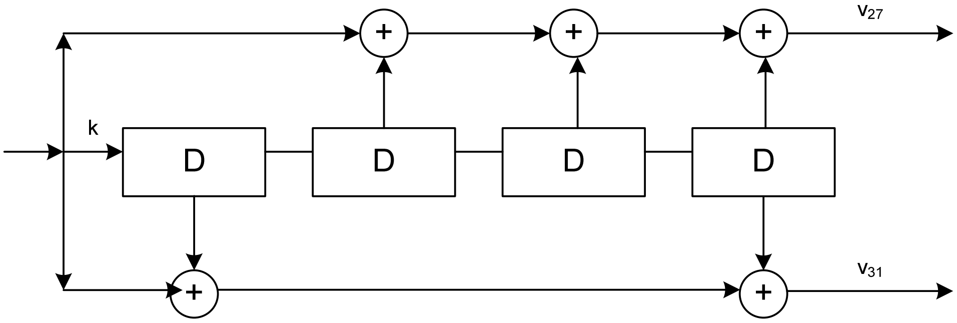

Figure 2.3. A (27,31) Convolutional Encoder.

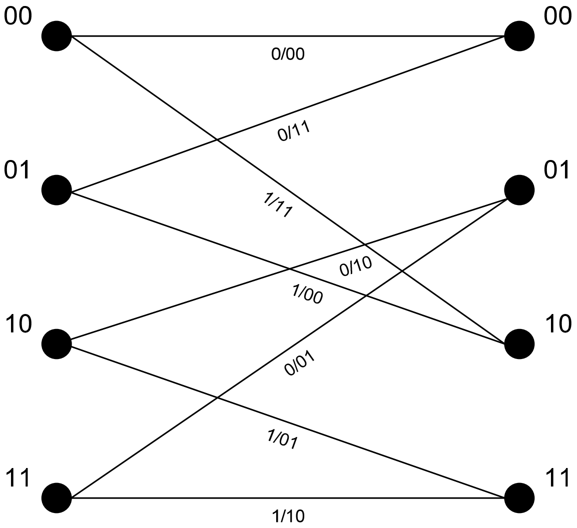

Figure 2.4. Trellis Representation of the (5,7) Convolutional Code.

earlier, due to increased memory this code shows increased performance. The generator polynomials of this encoder are given below with corresponding octal representation being (27,31). Figure 2.3 shows the (27,31) convolutional encoder, which illustrates the modulo-2 adder connections given by the generator polynomials g_1 and g_2,

$$g_1 = [10111]$$
$$g_2 = [11001].$$

A CC can be described by three alternative methods, namely: a tree diagram, a state diagram, and a trellis diagram [21]. Of these various methods, the trellis diagram is widely used to describe CCs since it aids in structuring decoding algorithms for these codes. The trellis representation of the (5,7) CC is shown in Figure 2.4. It can be seen that the (5,7) CC has four states and two branches entering and exiting each state at a given time instant. In general, each trellis diagram has 2^m states with 2^u branches entering each state and 2^u branches exiting each state of the trellis. In drawing a trellis diagram we use the convention that each branch is labeled with the input bit followed by the output codeword. In this case, after the initial transient, the trellis contains four nodes corresponding to four states of this code. From Figure 2.4, for instance, assume that we are at state 01, with an input bit 0 then we go to state 00 producing an output codeword 11. On the other hand, if the input bit was 1 then we would go to state 10 with 00 as the output codeword. A similar trellis description for the (27,31) CC is shown in Figure 2.5.

2.1.2 Optimum Decoding of Convolutional Codes

Several algorithms have been developed for decoding CCs. One of the most commonly used algorithms is the Viterbi algorithm, which is a maximum likelihood sequence estimator. Another commonly used variation of Viterbi algorithm which works with reliabilities of the decoded symbols is known as soft-output Viterbi algorithm (SOVA). The maximum a posteriori (MAP) decoding algorithm, like the Bahl, Cocke, Jelinek, Raviv (BCJR) algorithm, provides performance comparable to the Viterbi algorithm with only a little increase in complexity. This MAP decoding procedure works with probability measures of decoded symbols and hence it is suitable for an iterative turbo decoder. It is to be noted that both Viterbi and BCJR algorithms have a number of fundamental similarities. A variation the of BCJR algorithm, known as the soft-input soft-output (SISO) decoding algorithm, is explained in [23] and is used in this thesis to

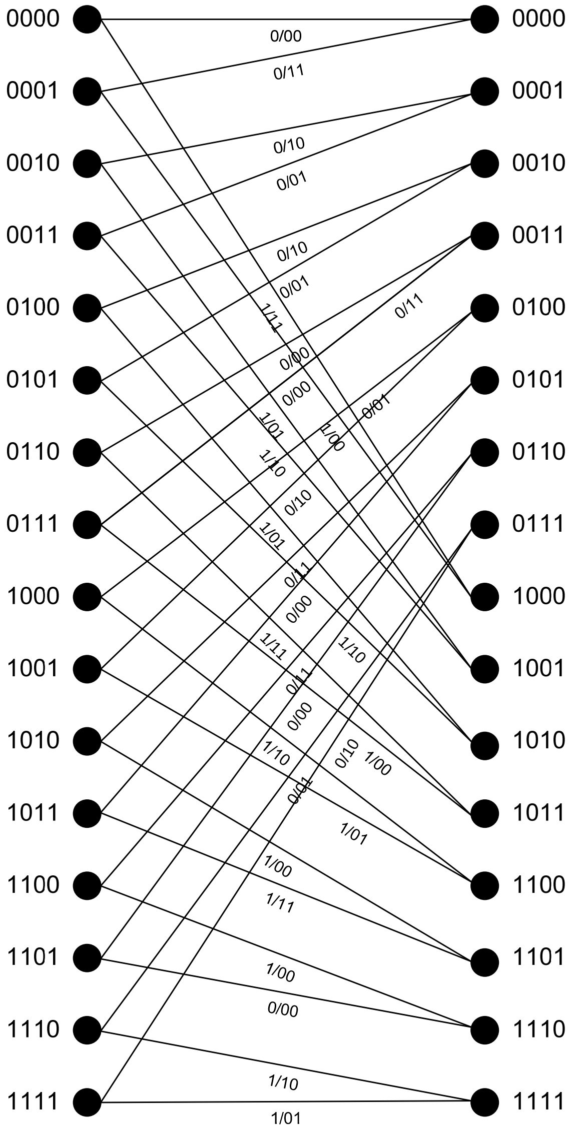

Figure 2.5. Trellis Representation of the (27,31) Convolutional Code.

decode CCs concatenated with CPM. A overview of the SISO algorithm is provided in Section 2.1.4 whereas the Viterbi algorithm is explained below.

The Viterbi algorithm computes the maximum likelihood code sequence given the received data [20]. A coded sequence $\{c_0,\, c_1,\, ...\}$ at the output of the convolutional encoder follows a path through the encoder trellis, while the corresponding received sequence r corrupted by channel noise may not follow the same path through the trellis. Hence the Viterbi decoder tries to find the maximum likelihood path through the trellis, which is closest to the path followed by the coded sequence. With the AWGN channel, the maximum likelihood path corresponds to the path through the trellis which is closest in *Euclidean* distance to r. While in a binary symmetric channel (BSC), the maximum likelihood path corresponds to the path through the trellis which is closest in *Hamming* distance to r. Two important styles of Viterbi decoding of CCs are hard and soft decision decoding. The crux of the algorithm can be explained as follows,

1) For each state s at time $t+1$, find the path metric of each path to state s by adding the path metric of each survivor path at a time t with the branch metric computed at $t+1$

$$\lambda_{t+1}(E_s) = \lambda_t(S_s) + Z_{s,t} \tag{2.1}$$

where $\lambda_t(\cdot)$ is the cumulative metric for a given state at index t, $Z_{s,t}$ is the incrementing branch metric computed at time index t, S_s denotes the starting state, and E_s denotes the ending state.

2) Among the paths to state s, the survivor path is selected to be the path with the smallest path metric.

3) Save the path and its metric at each state.

4) Go to the next time instant and repeat steps 1–3 until the end of the code sequence.

In the event that the path metrics of the merging paths are equal, a random choice among the paths is made with no negative impact on the likelihood performance.

The basic (5,7) CC with a block length of 1024 bits is simulated over 100000 blocks. The BER performance of this CC with binary phase shift keying (BPSK) modulation under hard and soft decision Viterbi decoding is shown in Figure 2.6. By way of reference, uncoded BPSK crosses BER $= 10^{-5}$ at $E_b/N_0 = 9.6$ dB. From the figure it can be seen that soft decision decoding of this CC yields a coding gain of 3.6 dB, which is

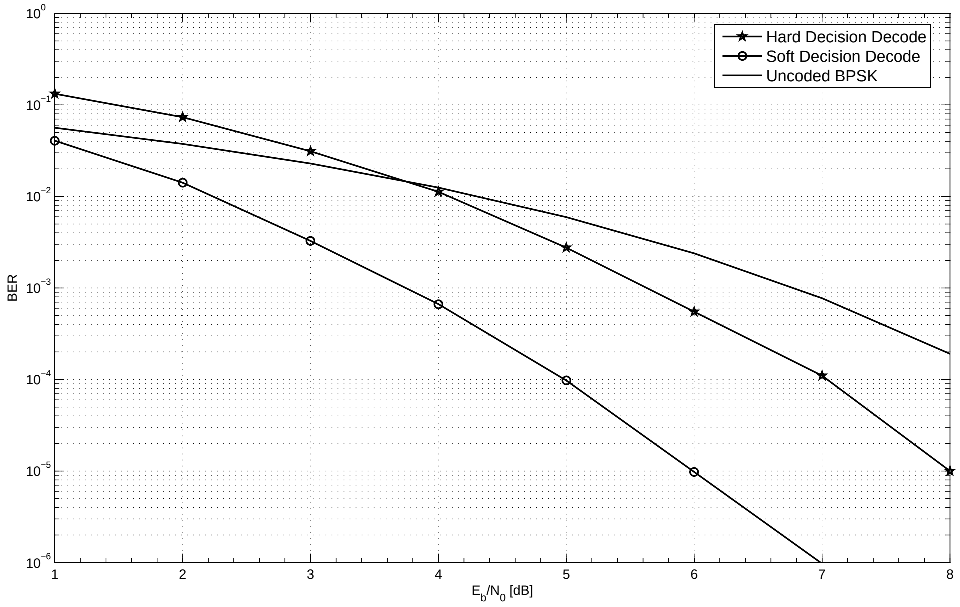

Figure 2.6. BER Performance of the (5,7) Convolutional Code under Viterbi decoding.

2.0 dB more than the gain produced by hard decision decoding.

2.1.3 Punctured Convolutional Codes

In some practical applications, including SCC-CPM developed in this thesis, there is a need to use high-rate CCs. However, the decoder implementation for a high-rate code is very complex [20]. Hence, to reduce the associated complexity we develop high-rate CCs from their low-rate counterparts by deleting a few coded bits before they are transmitted over the AWGN channel. This deletion of coded bits at the output of convolutional encoder is called *puncturing*. Thus high-rate CCs are developed from their low-rate counterparts with the encoder/decoder maintaining the complexity of a low-rate code.

The puncturing process involves periodically deleting selected bits at the output of the encoder which creates a periodically varying trellis code. Deletion of selected bits

Table 2.1. Map of Deleting Bits for High Rate Punctured Convolutional Codes Derived from Basic Rate 1/2 Codes with Constraint Lengths 2 and 4.

Coding Rate	Constraint Length = 2	Constraint Length = 4	N	S

1/2	1(5)
	1(7)
		1(27)
		1(31)
			2048	32

2/3	10 11	11 10	1536	27
3/4	101 110	101 110	1364	26
4/5	1011 1100	1010 1101	1280	25
5/6	10111 11000	10111 11000	1230	24
6/7	101111 110000	101010 110101	1197	24
7/8	1011111 1100000	1010011 1101100	1168	24
8/9	10111111 11000000	10100011 11011100	1152	24
9/10	101111111 110000000	111110011 100001100	1140	23

at the output of convolutional encoder depends upon a puncturing table which indicates the specific bits at the output to be deleted. Puncturing tables for the (5,7) CC and the (27,31) CC are given in Table 2.1. With a specific rate code, a 1 at the output indicates the specific bit is transmitted over the channel, whereas a 0 at the output signifies a deletion of the corresponding bit. In Table 2.1, the parameters N and S stand for the length of the codeword bits transmitted over the channel after puncturing and the spacing between any two bits in an interleaved sequence, respectively. More information on the parameters N and S is given in Chapter 3. A detailed description on puncturing CCs can be found in [24].

2.1.4 Soft-Input Soft-Output Decoding of Convolutional Codes

Though the Viterbi algorithm is an optimum decoding algorithm (in the sense of maximum likelihood sequence decoding) for CCs, the computational burden and storage requirements associated with the Viterbi algorithm make it impractical for CCs with large constraint length. Also, in a concatenated coding application, unlike the Viterbi algorithm, the decoder must be able to accept and output soft values. For the SCC systems developed in this thesis, it is of prime importance to utilize a decoder that takes soft values as its input and produces soft decisions at the output. Such a decoder is built here using the SISO decoding algorithm explained by Benedetto *et al* in [23]. The crux of this SISO algorithm is explained in this section.

In any turbo coding scheme with iterative decoding, the core of the decoder is a SISO *a posteriori* probability module. The SISO module described in [23] is a four port device with two inputs and two outputs. It takes as its input *a priori* probability distributions of the information word $P(u;I)$ and the code word $P(c;I)$ and forms as output an update of the probability distributions based on the code constraints. The updated *a posteriori* probability distributions for the information and code words are represented as $P(u;O)$ and $P(c;O)$, respectively. This algorithm is based on the trellis representation of CCs. With a detailed description of SISO given in [23], a brief description is given in the following two steps.

1) Similar to the branch metrics in the Viterbi algorithm, the SISO module calculates forward and backward recursion metrics represented by A_k and B_k, where k indexes time

$$A_k(s) = \sum_{e:s^E(e)=s} A_{k-1}[s^S(e)]P_k[u(e);I]P_k[c(e);I] \tag{2.2}$$

$$B_k(s) = \sum_{e:s^S(e)=s} B_{k+1}[s^E(e)]P_{k+1}[u(e);I]P_{k+1}[c(e);I] \tag{2.3}$$

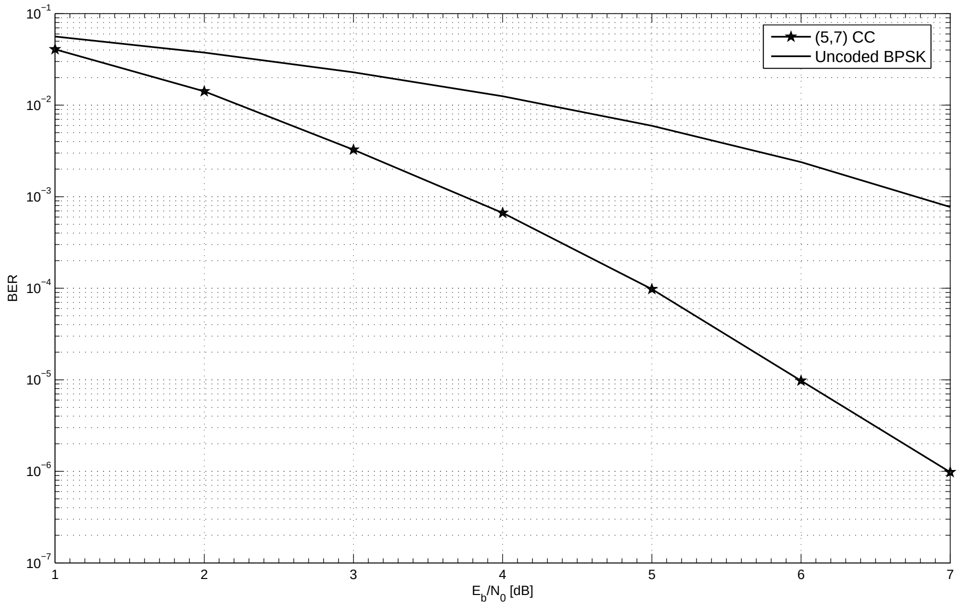

Figure 2.7. BER Performance of the (5,7) Convolutional Code under SISO decoding.

where S and E indicates starting and ending states. The initial values are $A_0(s) = 1$ if $s = s_0$ and $A_0(s) = 0$ otherwise; $B_n(s) = 1$ if $s = s_n$ and $B_n(s) = 0$, otherwise. The notation $e : s^E(e) = s$ indicates that the summation is performed over all edges such that the ending state of the edge is S. It is important to note that, an edge is an alternative name for a branch which connects any two states in the trellis representation.

2) Output probability distributions $(P(u;O)$ and $P(c;O))$ are calculated based upon forward and backward recursive branch metrics and input *a priori* probability distributions

$$\tilde{H}_c^j = \sum_{e:c_k^j(e)=c^j} A_{k-1}[s^S(e)]P_k[u(e);I]P_k[c(e);I]B_k[s^E(e)] \tag{2.4}$$

$$\tilde{H}_u^j = \sum_{e:u_k^j(e)=u^j} A_{k-1}[s^S(e)]P_k[u(e);I]P_k[c(e);I]B_k[s^E(e)] \tag{2.5}$$

where $\tilde{H}_c^j$ and $\tilde{H}_u^j$ represents *a posteriori* probabilities $P(c;O)$ and $P(u;O)$ respectively.

These output probability distributions are the soft-outputs representing the reliability of the decoded sequence. These reliabilities can be hard decoded after a pre-defined number of iterations to produce a decoded sequence. The performance of the (5,7) CC with the SISO decoding algorithm is shown in Figure 2.7. In a modified “max-log” version of the SISO decoding algorithm, known as log-likelihood ratio SISO (LLR-SISO), we replace the “sum” term, in Equations 2.2 and 2.3, with a “max” term and then take the log of the resulting equation. CCs with constraint lengths $k = 2, 3, 4$ were simulated with a block length of 1024 bits over 100000 blocks. The performance of these CCs with BPSK modulation under LLR-SISO decoding is shown in Figure 2.8. From Figures. 2.6, 2.7 and 2.8 it can be seen that the performance of both SISO and LLR-SISO is exactly the same as the performance shown by soft decision Viterbi decoding. The coding gain produced by each of these decoding algorithm is 3.6 dB, but as mentioned earlier LLR-SISO is widely used in turbo decoders because of its computational simplicity (i.e., because of the log operation, the multipications in Equations 2.2 and 2.3 become additions).

2.2 Turbo-Product Codes

A key problem in the field of channel coding is the inherent decoding complexity associated with most powerful codes. An approach to solve this problem would be to use simplifications that reduce the decoding complexity associated with these codes. Another important approach is to construct good codes that also have practical decoding complexity. A solution to this problem is concatenated coding. The idea behind decoding concatenated codes is to decode each constituent code individually so that the overall decoding complexity remains reasonable. An excellent example for concatenated coding is a coding scheme based on CCs concatenated with Reed-Solomon

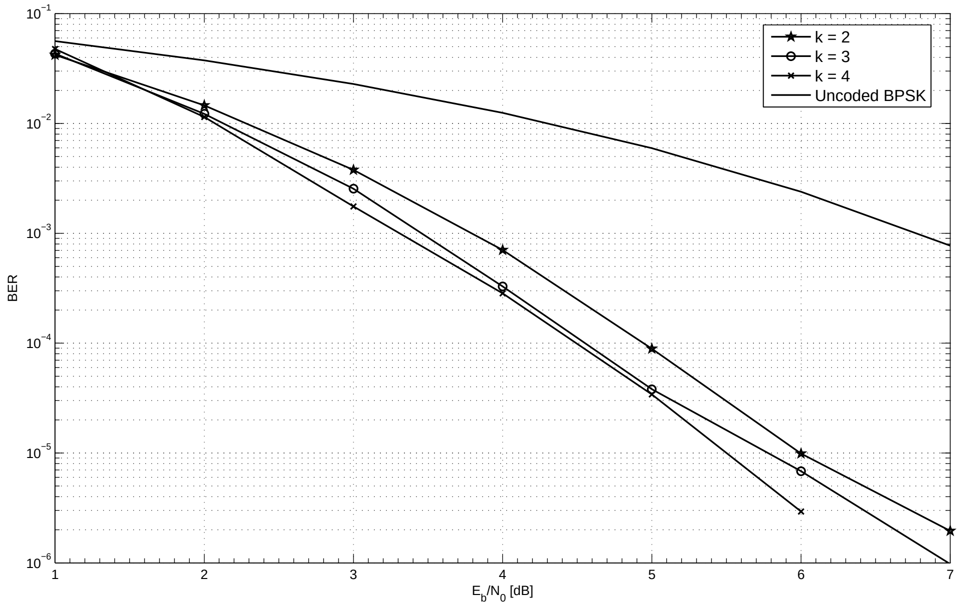

Figure 2.8. BER Performance of Convolutional Codes under LLR-SISO decoding Algorithm.

code. This code achieves a code rate close to or even better than the channel cutoff rate, which was considered to be practical channel capacity until very recently [2].

With the introduction of turbo codes much attention was given to convolutional turbo-codes (CTC) with little attention on block turbo codes (BTC). Even though concatenated coding was first introduced for block codes, the first algorithms introduced to decode these codes produced poor results owing to their hard-input hard-output decoding capabilities. Pyndiah *et al* in 1994 [25] proposed new soft-input soft-output decoders for linear block codes and showed their performance to be comparable to CTCs using near-optimal algorithms. These new BTCs, also known as turbo-product codes (TPCs) provide a good compromise between performance and complexity and are highly suited for practical implementation.

2.2.1 Encoding

Product codes, introduced by Elias [26] in 1954, are relatively simple and efficient BTCs built from two or more shorter block codes. Let us consider two systematic linear block codes C^1 and C^2 with parameters (n_1, k_1, δ_1) and (n_2, k_2, δ_2), where n_i, k_i, and δ_i are the codeword length, information block length, and minimum Hamming distance, respectively. Now product code P, as depicted in Figure 2.9, is obtained by arranging information bits along k_1 rows and k_2 columns and then coding the k_1 rows using code C^2 and the n_2 columns using code C^1. The resulting product code P has dimensions $n = n_1 \times n_2$, $k = k_1 \times k_2$, $\delta = \delta_1 \times \delta_2$ with code rate R given by $R = R_1 \times R_2$, where R_1 and R_2 are code rates of individual systematic linear block codes. Given this procedure to form TPCs, all rows of matrix P are codewords of C^1 and all columns of matrix P are codewords of C^2. Thus long TPCs with large minimum Hamming distance can be produced by simply multiplying short systematic block codes with small minimum Hamming distance. Once encoded, n coded bits are modulated and sent over the AWGN channel.

2.2.2 Near-Optimum Chase Decoding Algorithm

TPCs can be decoded sequentially by alternatively decoding rows and columns of P. However, optimum performance can be realized when soft decoding of the component codes is done. Thus SISO decoders provide optimum decoding performance and allow iterating the sequential decoding of P, which reduces the BER after each iteration. In 1972, Chase proposed a near-optimum algorithm for soft decoding TPCs [27]. The main idea behind this algorithm is to reduce the number of reviewed codewords and use a set of the most probable codewords. The algorithm can be briefly described as follows,

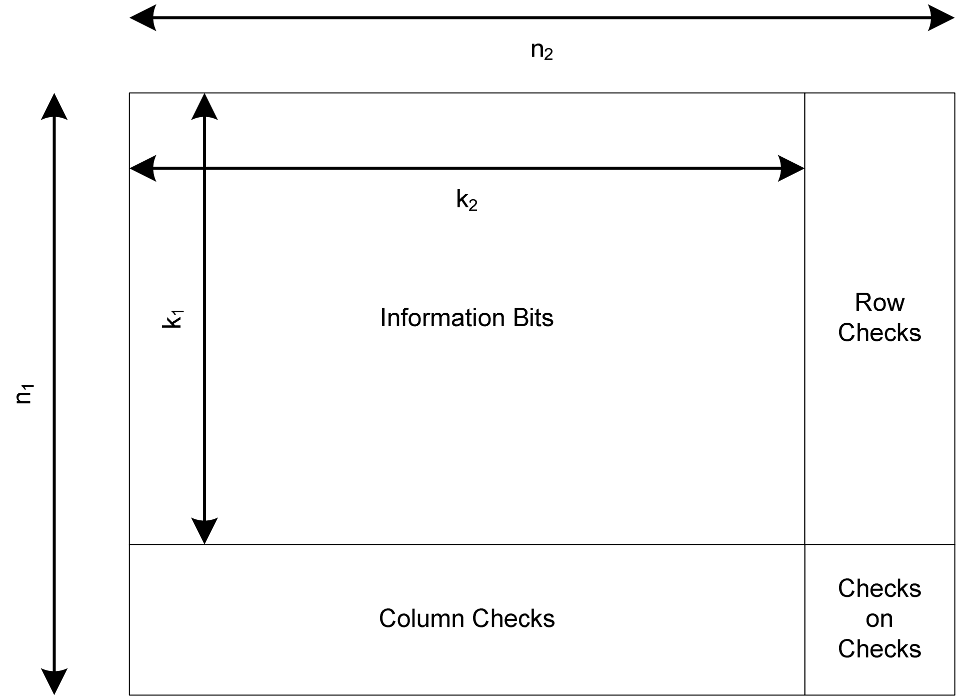

Figure 2.9. A Simple Turbo-Product Code Example.

step 1: Determine the position of $p = \lfloor \delta/2 \rfloor$ least reliable binary elements of Y using R. The reliability of y_j is given by $|r_j|$.

step 2: Form 2^p binary n-tuple test patterns T at p least reliable positions.

step 3: Decode test sequences $Z^q = y \oplus t^q$ using an algebraic decoder and the resulting codeword C^q to the subset ω. Here $\oplus$ denotes modulo-2 addition.

For each codeword found in the above step we compute the Euclidean distance from R and then select a optimum codeword D based upon minimum Euclidean distance. Then we find a competing codeword C which is at a minimum Euclidean distance from R such that $c_j \neq d_j$. With the codewords C and D known we the calculate normalized reliability of decision d_j as

$$\hat{r_j} \approx r_j + w_j \tag{2.6}$$

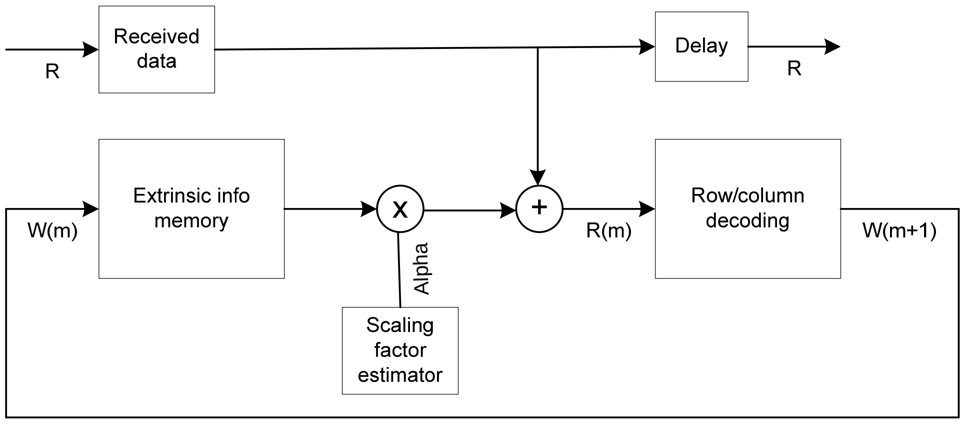

Figure 2.10. Block Diagram of the Turbo-Product Code Decoder.

where the extrinsic information w_j is

$$w_j = \left(\frac{|R - C|^2 - |R - D|^2}{4} \right) d_j - r_j. \tag{2.7}$$

If the competing codeword could not be found due to limited reliable bits p, the extrinsic information used for next decoding step is

$$w_j = \beta \times d_j \; with \; \beta \geq 0. \tag{2.8}$$

The block diagram for decoding TPCs is shown in Figure 2.10. With the extrinsic information W calculated, the soft-input for the next decoding step is

$$[R(m)] = [R] + \alpha(m)[W(m)] \; with \; R(0) = R. \tag{2.9}$$

Different TPCs with $m = 5$ ((32, 26) × (32, 26) TPC), $m = 6$ ((64, 57) × (64, 57) TPC), $m = 7$ ((128, 120) × (128, 120) TPC) were simulated with varying information block lengths. The performance of these TPCs with BPSK modulation over the AWGN channel under near-optimal iterative Chase decoding is shown in Figure 2.11. From Figure 2.11 it is seen that a rate 0.7932 TPC yields a coding gain of 5.9 dB over uncoded

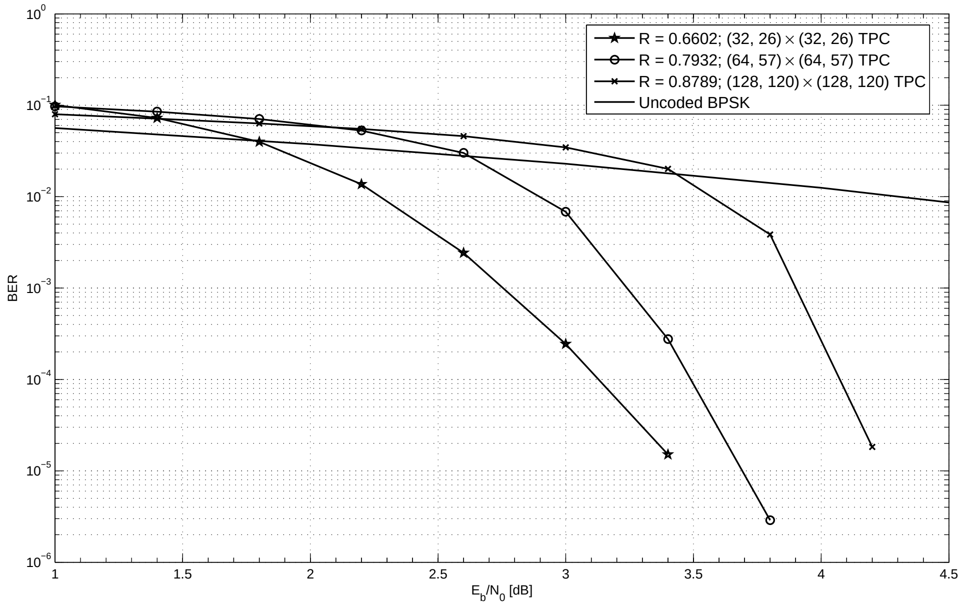

Figure 2.11. BER Performance of the Turbo-Product Codes under a Near-Optimum Chase decoding Algorithm.

BPSK.

2.3 Repeat-Accumulate Codes

Repeat-accumulate codes (RAC) [28] are a simple class of rate $1/q$ serially concatenated codes where the outer code is a rate $1/q$ repetition code and the inner code is a rate 1 CC with a transfer function $1/(1+D)$. The iterative decoding performance of these RACs is seen to be exceptional in spite of the code being simple and the decoding algorithm being only near-optimal. On the AWGN channel, as the code rate tends to zero, these RACs achieve the ultimate Shannon limit, -1.6 dB [28].

2.3.1 Encoding

A simple encoder structure for a rate $1/q$ RAC is shown in Figure 2.12. A N-bit input block is repeated q times and scrambled by a $qN \times qN$ interleaver P which repre-

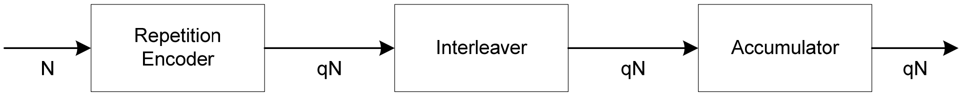

Figure 2.12. Encoder for the (qN, N) Repeat-Accumulate Code.

sents an arbitrary (random) permutation of a qN-bit block. The output of the interleaver is encoded by a rate 1 accumulator which is nothing more than a recursive convolutional encoder with a transfer function $1/(1 + D)$. A simple way to understand this repeat-accumulate code is to think of it as a block code whose input block $[x_1, x_2, ..., x_n]$ and output block $[y_1, y_2, ..., y_n]$ are related by the formula

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_1 + x_2 \\ &. \\ &. \\ &. \\ y_n &= x_1 + x_2 + x_3 + ... + x_n \end{aligned} \tag{2.10}$$

where the addition performed is modulo-2.

2.3.2 Sum-Product Decoding Algorithm

It has been proved in [28] that RACs perform well with maximum likelihood decoding but the associated complexity is prohibitively large. Hence for RACs a simple message passing decoding algorithm which closely approximates the performance of maximum likelihood decoding is used. To better understand message passing decoding of RACs, we represent these codes in the form of a Tanner graph [28]. A Tanner graph is bipartite graph whose vertices are partitioned into variable nodes V_m and check nodes

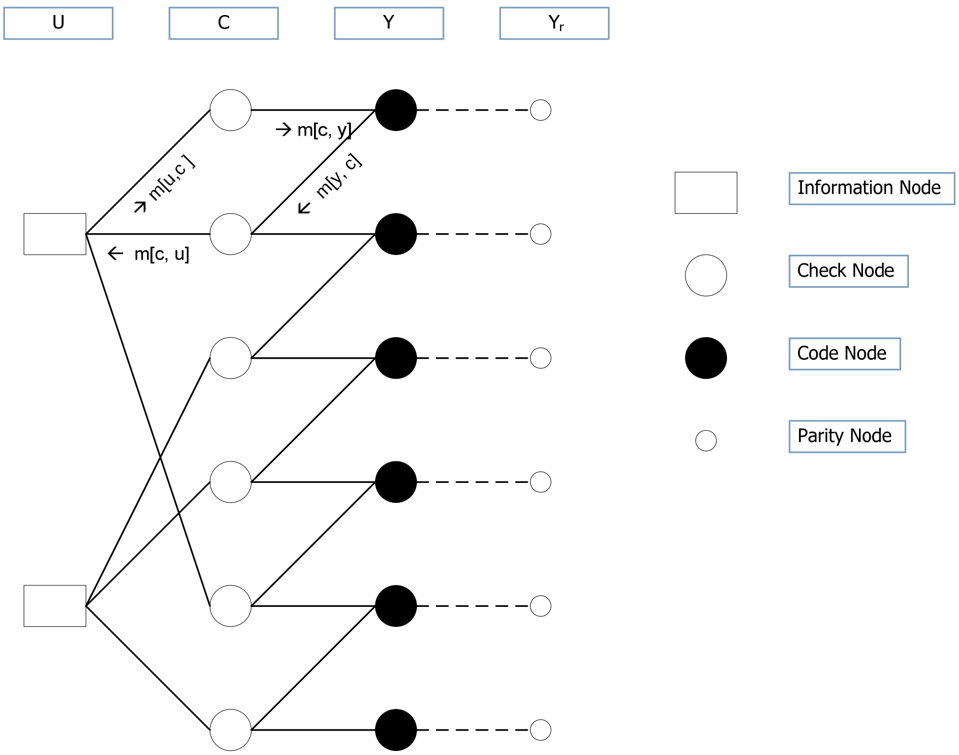

Figure 2.13. Tanner Graph for a Repetition 3, Length 2 Repeat-Accumulate Code.

V_c with edges $E \subseteq V_m \times V_c$. The check nodes represent certain constraints on variable nodes and edges indicates the presence of a variable node in any check constraints.

The Tanner graph realization of a repetition 3 length 2 RAC is shown in Figure 2.13. This graph includes information bits u_i, code bits c_i, parity bits y_i and received code bits y_r. It is to be noted that y_r are not a part of Tanner graph, they are included along with the regular Tanner graph because they provide evidence for decoding the received bits. In the Tanner graph shown in this example, regardless of the block length, each information node u_i is connected to q check nodes and hence every $u \in U$ has degree q. Also, every vertex $c \in C$ has degree 3 except the first vertex c_1 and $y \in Y$ has degree 2 except the last vertex.

The message passing algorithm also known as the belief propagation algorithm, is a

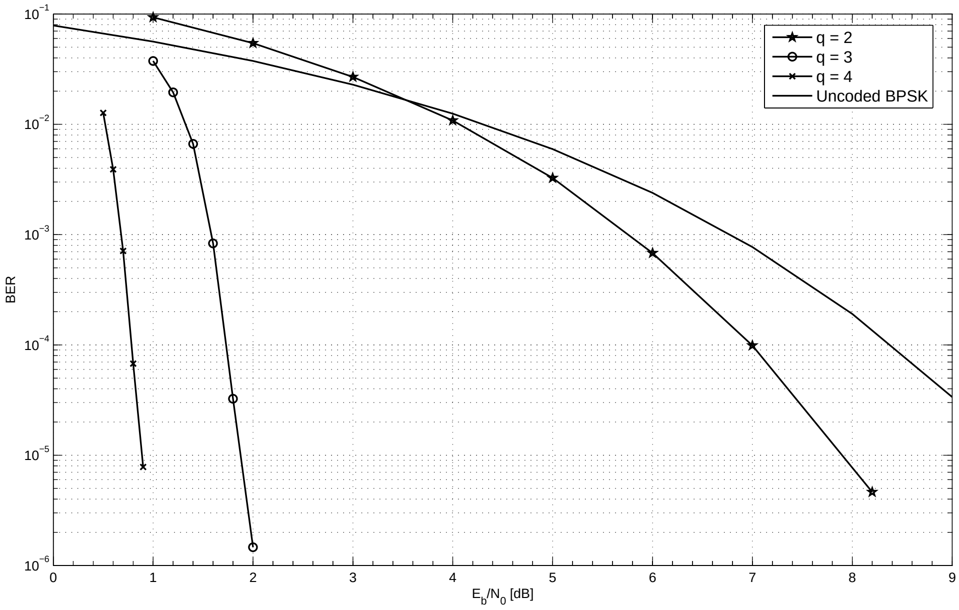

Figure 2.14. BER Performance of the Repeat-Accumulate Codes under Sum-Product decoding Algorithm.

specific instance of the generalized distributive law (GDL) algorithm [29] with specific scheduling. The messages passed over edges in this algorithm are posterior densities of bits associated with variable nodes. There are four types of messages passed over the edges in this belief propagation algorithm for decoding RACs. These messages are of two distinct classes 1) messages sent and received between vertices $u \in U$ and $c \in C$ $(m[u, c], [c, u])$, and 2) messages exchanged between vertices $y \in Y$ and $c \in C$ $(m[y, c], [c, y])$. All these messages passed over the edges of the Tanner graph are shown in Figure 2.13 and have conditional probabilities. The code node y has belief provided by y_r and it is denoted as $B(y)$. A brief description of belief propagation algorithm for decoding RACs, explained in [28], is as follows,

Initialization: Initialize all message $m[u, c], m[c, u], m[c, y], m[y, c]$ to zero and update them in each iteration. Select the maximum number of iterations to be K. Each iteration has three steps, which are executed in the order given below *step 1*: Update $m[y, c]$

step 2: Update $m[u, c]$

step 3: Update $m[c, y]$ and $m[c, u]$

The update procedure associated with steps 1–3 are explained in [28]. After K iterations we calculate $s(u) = \sum_c m[u, c]$, where the summation is over all c such that $[u, c] \in E$. Now the calculated $s(u)$ is hard decoded to produce the decoded sequence, if $s(u) > 0$ the bit is decoded to be a 1 else the bit is decoded to be a 0.

RACs with $q = 2$ (rate 1/2), $q = 3$ (rate 1/3), and $q = 4$ (rate 1/4) were simulated for 100000 blocks with each information block's length being 4096 bits. The performance of RA codes with BPSK is shown in Figure 2.14. As seen from this figure, a rate 1/4 RAC yields a coding gain of 8.7 dB compared to a gain of 7.7 dB produced by a rate 1/3 RAC under BPSK modulation.

Chapter 3

Serially Concatenated Codes

Chapter 2 presented a basic overview of FEC schemes that can be combined along with CPM to build a SCC-CPM system. This chapter deals with serial concatenation of interleaved codes. To start with, an overview of SCC is presented which provides a detailed description of a concatenated coded communication system. Section 3.2 describes such a system and defines certain design guidelines which are to be followed while building a SCC system with iterative decoding. Ideally, any SCC would have an inner and outer encoder separated by an interleaver and decoded via a near-optimum iterative decoder. This decoder iterates *a posteriori* probabilities of decoded symbols between the inner and outer decoder. Section 3.3 provides a brief description on coded modulations like SOQPSK-TG and PCM/FM, which are treated as inner codes of the SCC-CPM systems developed here. With the inner codes fixed, Section 3.4 discusses the different SCC-CPM systems built in this thesis. Additionally this section lists the parameters associated with these SCC-CPM systems and describes the system functionality.

3.1 Overview

Shannon's channel coding theorem suggests strong coding behavior for random-like codes as the code block length increases. However, any increase in block length would imply an exponential increase in the decoding complexity. To overcome this issue a new coding scheme was introduced in 1993 which allowed for a very long concatenated code word with only moderate decoding complexity. This coding technique with concatenated code words was called parallel concatenated codes (PCC) or *turbo codes*. Since the decoding complexity was relatively small for the dimension of this code, very long codes were possible and hence the bounds of the channel coding theorem were practically achievable. Alternative solutions to parallel concatenation have also been studied such as trellis-coded modulation (TCM) or serial concatenation of convolutional codes.

In any classic concatenated coding scheme, the main ingredients that form the basis are *constituent codes* and an interleaver. The novelty of these concatenated codes lies in the way we use the interleaver, which is embedded into the code structure to form an overall concatenated code with very large block length. Concatenated codes, either SCCs or PCCs, can be coupled with CPM. However we prefer SCCs because, with PCCs, the location of the interleaver would destroy the continuous phase of the modulation. Moreover, based on the results provided in [2], it is believed that SCCs can be considered valid and in some cases, superior alternative to PCCs. Hence this thesis considers serial concatenation of interleaved codes or SCCs. These SCCs can either be serially concatenated block codes (SCBCs) or serially concatenated convolutional codes (SCCCs) based on the nature of their constituent codes. The next section in this chapter describes in detail a typical SCC while the following sections justifies the choice of constituent codes and explains different SCCs built in this thesis.

3.2 General System Description

Initially motivated only by theoretical research interests, concatenated codes have since evolved as a standard for applications where coding gains are needed. Such a typical SCC communication system is shown in Figure 3.1. Just like any other serially concatenated system, the system shown in Figure 3.1 has an inner encoder separated from an outer encoder by an interleaver. In most conventional realizations, the codes used are recursive systematic CCs [2]; however, other codes can also be used. Such a system uses separate decoders connected via a deinterleaver to decode received symbols. The information sequence at the input is encoded, interleaved, and modulated before being transmitted over the AWGN channel. As a matter of fact, it is assumed that several blocks of information sequences are continuously transmitted over the channel with encoder states not being reset to zero with each new block of information bits. This continuous time signal, given by $s(t)$, is transmitted over the AWGN channel. This transmitted siganl is corrupted by the AWGN signal $n(t)$ which is a complex baseband representation of noise in the channel with a double-sided power spectral density $N_0/2$. Hence the continuous time received signal is given by $r(t) = s(t) + n(t)$. At the receiver, a demodulator initially demodulates the received signal which is deinterleaved and decoded eventually.

A key feature of concatenated codes is the *iterative* decoding algorithm. In this algorithm, the decoder of each constituent codes takes a turn in operating on the received data. Each decoder produces an estimate of the probabilities of the transmitted symbols and hence these are soft decoders. The probabilities of the symbols passed between decoders are known as *extrinsic* probabilities and the decoders uses these as prior probabilities to update the estimates of the transmitted symbols. After a predefined number of iterations, the soft output of the decoder is hard limited and presented

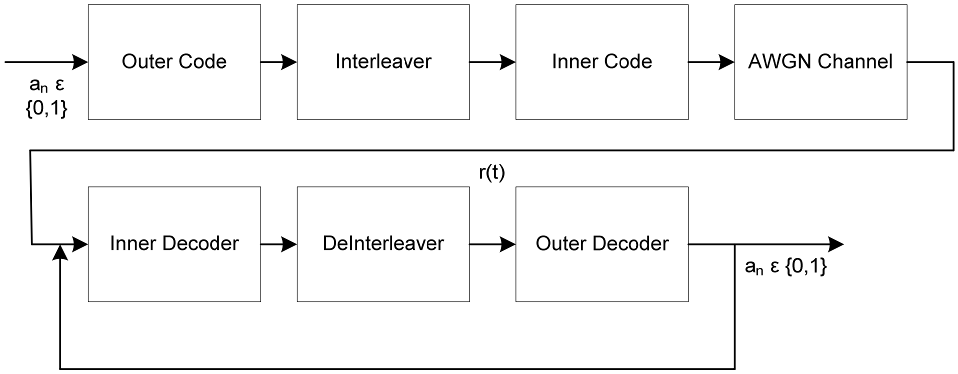

Figure 3.1. A Typical Serially Concatenated Coded Digital Communication System.

as an output.

In a serially concatenated system, like the systems explained here, an interleaver aims at increasing the Hamming weight of a code word between outer and inner encoder, i.e., the interleaver permutes the information word from the input of the outer encoder to that of the inner encoder, thereby associating a low weight outer code word with a large weight inner code word. This helps to improve the maximum likelihood performance, which depends upon the code word weight spectrum [2]. At the receiver, a deinterleaver helps to provide loosely correlated input to the outer decoder from the inner decoder which improves the decoding performance. We prefer a random interleaver instead of a block interleaver, since the latter is not effective with SCC systems. Also, immunity to fading, achieved with block interleavers, is realized to some extent with random interleavers.

It is to be noted that the coding gain of SCC systems increases along with an exponential increase in the complexity of ML decoding, because the size of the interleaver increases. However, when iterative decoding of constituent codes is used, which is not ML and is slightly suboptimal, decoding complexity is independent of the size of the interleaver. The difference in the performance of a SCC system with the variation in the size of the interleaver will be illustrated with BER performance results in Chapter 4. It is of prime importance to realize a good tradeoff between the performance of a SCC system and the size of an interleaver used to build such a system, because a large interleaver is not preferred in aeronautical telemetry as it increases latency in the transmitter.

We use a S-random interleaver which separates the constituent codes and provides good interleaver gain, the order of which is determined by d_{free}. The performance of this interleaver, adopted from [30], depends upon the time separation introduced between any two bits in an information sequence. This time separation, denoted by a parameter S, allows different bits to fade differently, thereby preventing any burst of errors. The parameter S is bounded by $S < \sqrt{N/2}$, where N is the length of the block of coded bits. The value of S with different coding rates is given in Table 2.1. An important advantage of using a S-random interleaver in a SCC system is to provide some form of time diversity to guard against localized corruptions and burst of errors. Other important advantages includes 1) increasing the Hamming weight of the codeword thereby increasing code strength, 2) reducing correlation in information between inner and outer decoder which improves iterative decoding of concatenated codes.

Serially concatenated systems designed to realize high coding gains can either be SCCC or SCBC. For practical applications, SCCC's are to be preferred to SCBC's. One reason is that *a posteriori* probability algorithms are less complex for convolutional codes than for block codes [23] and another is that the interleaver gain can be greater for convolutional codes [20]. The performance of a concatenated code with an interleaver depends on the constituent codes and on the interleaver in a strictly independent manner. Hence to build a serially concatenated code, a *decoupled* design procedure is adopted in which we first design the constituent code and then tailor an interleaver based on the code's characteristics. Benedetto *et al* [2] state a few important design considerations which this thesis utilizes to build SCC-CPM:

$1)$ the inner encoder must be a recursive encoder since it yields an interleaver gain, unlike block codes and nonrecursive codes.

$2)$ the effective free distance (d_{free}) of the inner encoder must be maximized.

$3)$ for values of E_b/N_0 where the performance of the concatenated system is dominated by its d_{free}, increasing interleaver length yields a gain in performance. Also to increase the interleaver gain, an outer code with large and, possibly, odd value of d_{free} should be selected.

$4)$ the outer encoder should be a nonrecursive encoder which has less input errors associated with error events at the d_{free}.

3.3 Inner Codes

The term *continuous phase modulation* (CPM) [21] refers to a large class of constant-envelope waveforms that are characterized by three parameters: the data alphabet size M (e.g. binary, quaternary), the modulation index h, and the shape and duration of the *frequency pulse*. These three parameters are usually selected to satisfy constraints on the limited resources of power, bandwidth, and complexity. CPMs are a natural choice for the inner codes in a SCC system. This is because they can be viewed as *recursive codes*, which are necessary to yield large interleaving gains in such systems [2]. In this thesis, we consider coding schemes using two of the CPMs used in aeronautical telemetry [4]: PCM/FM and SOQPSK-TG.

PCM/FM is a binary CPM with parameters $M = 2$, $h = 7/10$, and a raised cosine frequency pulse shape with duration $L = 2$ symbol times (2RC) [32]. Among the two modulations considered here, PCM/FM has the highest detection efficiency, the lowest spectrum efficiency, and it requires moderate decoding complexity.

The other modulation technique considered here is SOQPSK-TG. SOQPSK-TG is often considered a derivative of offset quadrature phase shift keying (OQPSK) and minimum shift keying (MSK). Although OQPSK has an improved power spectrum compared to quadrature phase shift keying (QPSK) when using nonlinear amplifiers, it still has waveform envelope fluctuations due to instantaneous transitions between adjacent phase states. SOQPSK-TG is a *constant envelope* generalization of OQPSK. It is more spectrally efficient than OQPSK and MSK, in exchange for slightly lower detection efficiency. The SOQPSK transmitter consists of a special binary-to-ternary *precoder*—which converts the binary information symbols to ternary channel symbols that are constrained to follow OQPSK data transitions—followed by a standard CPM modulator. In the case of SOQPSK-TG, the CPM modulator is configured with $h = 1/2$ and uses the custom frequency pulse shape specified in [4]. SOQPSK-TG can be described with a 4 state trellis [31], which requires low decoding complexity. Compared to PCM/FM, SOQPSK-TG has twice the spectral efficiency and has lower detection efficiency.

As mentioned above, we use PCM/FM and SOQPSK-TG as inner codes of a SCC system over an AWGN channel. The inner demodulator/decoders for these codes are based on the soft-input soft-output (SISO) algorithm [11, 23] and were designed and implemented in [32]. We now develop SCC systems that combine a number of different outer codes with these inner modulations/codes.

We use the coherent and noncoherent demodulators from [32]. The following sections describes the system set-up of serially concatenated convolutionally coded CPM (SCCC-CPM), turbo-product coded CPM (TPC-CPM) and repeat-accumulate coded CPM (RAC-CPM).

3.4 Serially Concatenated Coded CPM

In this section we describe the set up of a SCCC-CPM. Based on the results given in [9], TPCs prove to be an attractive candidate to develop a concatenated TPC-CPM. In this section we also summarize the set up for a RAC-CPM.

3.4.1 Serially Concatenated Convolutionally Coded CPM

The first SCC scheme we consider is serially concatenated convolutionally coded CPM (SCCC-CPM), a block diagram of which is shown in Figure 3.2. This scheme consists of an outer convolutional encoder and an inner CPM modulator, which are separated by an S-random interleaver [30]. We select two rate 1/2 convolutional codes as candidates for the outer code:

- CC1: constraint length $k = 2$ and generators $(5, 7)$; and
- CC2: constraint length $k = 4$ and generators $(27, 31)$.

The encoders for CC1 and CC2 are explained in [20] and are non-recursive with a d_{free} of 5 and 7, respectively. These codes completely satisfy the design criteria for outer codes stated in Section 3.2 [2]. A S-random interleaver, explained in Section 3.2, separates the two constituent codes.

In Figure 3.2, the received signal is demodulated using the PCM/FM and SOQPSK-TG SISO algorithms explained in [32]. The two SISO modules are "max-log" versions of the ones in [11, 23] and the soft probabilities are in the form of log-likelihood ratios (LLRs). The demodulator SISO takes as its inputs 1) the received signal $r(t)$ and 2) a soft input on the probability of the coded symbols. The demodulator SISO outputs an updated soft probability of the coded symbols. This output is deinterleaved and used as a soft input to the convolutional SISO decoder. The lower soft input to the convolutional

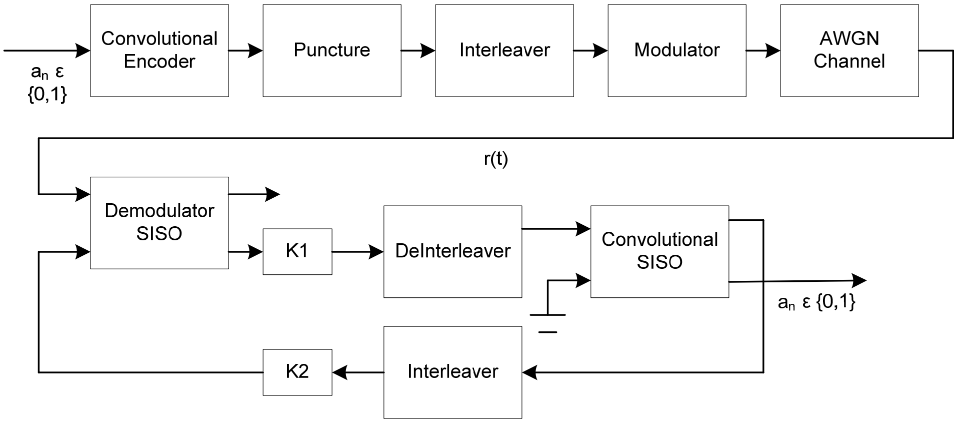

Figure 3.2. Serially Concatenated Convolutionally Coded CPM with Iterative Turbo Decoding.

SISO decoder in Figure 3.2 has a numerical value of zero, which corresponds to the assumption that the information bits (ones and zeros) are equally likely to occur. The outer SISO decoder produces updated versions of its inputs, one of which is interleaved and fed back as an input to the CPM SISO demodulator.

Probabilities are exchanged between the inner demodulator and the outer decoder in this manner for a predefined number of iterations. These probabilities are scaled by constants $K1$ and $K2$ to improve the overall BER performance [33]. In the case of SOQPSK-TG we select $K1 = 0.75$ and $K2 = 0.75$. In the case of PCM/FM we select $K1 = 0.65$ and $K2 = 0.65$; these values were determined by simulation. At the end of predefined number of iterations (in this case five iterations), the lower soft output of the convolutional SISO decoder is hard limited and constitutes the final output of the decoder. The performance of this system under coherent and noncoherent demodulation is presented in Section 4.

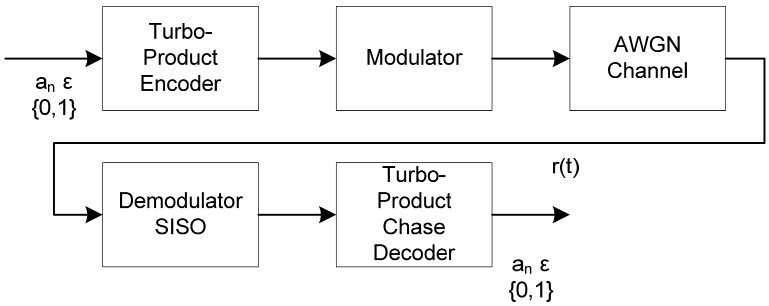

Figure 3.3. Turbo Product Coded CPM with Iterative Chase Decoding.

3.4.2 Turbo-Product Coded CPM

The second SCC scheme we consider here is TPC serially concatenated with CPM (TPC-CPM), a block diagram of which is shown in Figure 3.3. An initial study of TPC scheme for aeronautical telemetry was conducted in [9]. The primary difference between our approach and the one in [9] is that we use a SISO module to generate the soft output of the demodulator that is CPM-based and has general applicability, as opposed to the *ad hoc* soft output schemes developed in [9] that are applicable to the telemetry modulations only.

In Figure 3.3, the information bits are turbo product encoded using the methods explained in [20, 27]. We select the $(64, 57) \times (64, 57)$ TPC, the $(32, 26) \times (32, 26)$ TPC, and the $(128, 120) \times (128, 120)$ TPC as candidates for the outer code. An interleaver between the TPC and the modulator is not considered here because there is no noticeable difference in the performance of TPC-CPM with or without an interleaver.

The received signal is CPM demodulated using a SISO demodulator, which is identical to the SISO demodulator used in the SCCC-CPM scheme. The soft output of the demodulator is fed to an iterative Chase decoder [27]. Before proceeding to the simulation results, we shall now give the parameters used in the simulation of the Chase decoder which are exactly the same as the parameters specified in [27].

- The number of test patterns is 16 and are generated by the four least reliable bits;
- $Weighting\ Factor\ \alpha = [0.0, 0.2, 0.3, 0.5, 0.7, 0.9, 1.0, 1.0]$;
- $Reliability\ Factor\ \beta = [0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0]$;
- The maximum iteration number is 4, which is equivalent to eight decoding steps.

Because TPCs are effectively concatenated block codes, the Chase decoder iteratively decodes its own constituent codes without involving the demodulator SISO in the iteration loop. At the end of predefined number of iterations (in this case four iterations), the output from the Chase decoder is hard limited and constitutes the final output of the TPC-CPM system.

The performance of this system under coherent and noncoherent demodulation is presented in Section 4. In particular, the performance results show that our approach is 0.8 dB better than the approach in [9] due to our use of the SISO algorithm for CPM demodulation.

Figure 3.4 shows the various styles of decoding TPC-CPM. As seen from the figure the best coding gain performance is realized when the received signal is subjected to 2 receiver iterations. Each receiver iteration consists of a single demodulation with 5 decoding iterations. At the end of the first receiver iteration the soft-output from the chase decoder is fed back to the CPM SISO demodulator in a way similar to the exchange of soft information done in SCCC-CPM. However the complexity due to 2 receiver iterations is double the complexity of a single receiver iteration. Moreover with a single receiver iteration the coding gain realized is just 0.2 dB less than the gain realized with an additional receiver iteration. Hence as a trade off between complexity and BER performance, the TPC-CPM developed here employs a single receiver iteration. It is noted that of all the styles of decoding TPC-CPM, a single receiver iteration delivers the best trade off between complexity and performance.

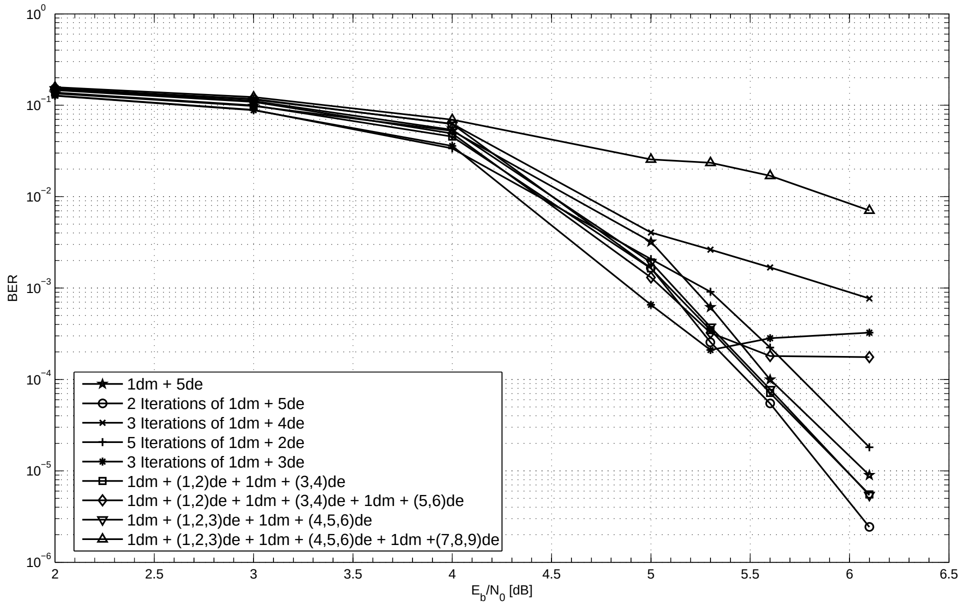

Figure 3.4. Different Styles for Decoding Turbo Product Coded CPM: “dm” means demodulation, “de” means decode.

3.4.3 Repeat-Accumulate Coded CPM

The final SCC scheme considered here is RAC serially concatenated with CPM (RAC-CPM). As shown in the Figure 3.5 information bits are repeat-accumulate encoded using the procedure explained in 2.3. We select a length 2 RAC with its repetition factor q varying between $q = 3$ and $q = 4$. Based on simulation results, an interleaver between the RAC and the modulator is not considered here because there is no noticeable difference in the BER performance of RAC-CPM with or without an interleaver.

The received signal is CPM demodulated using a SISO demodulator, which is identical to the SISO demodulator used in the SCCC-CPM scheme. The soft output of the demodulator is fed to an iterative sum-product decoder [28]. Because RACs are effectively serially concatenated codes, the sum-product decoder iteratively decodes its own constituent codes without involving the demodulator SISO in the iteration loop. At the

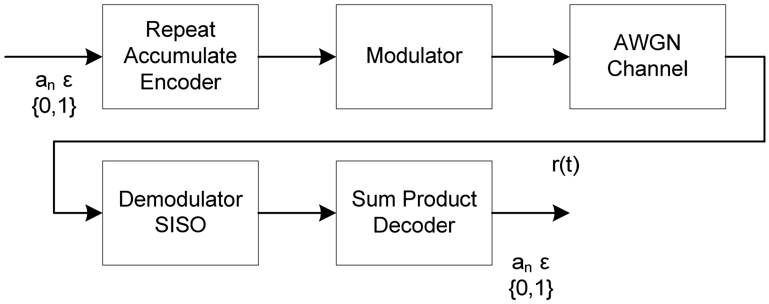

Figure 3.5. Repeat-Accumulate Coded CPM with Iterative Sum-Product Decoding.

end of predefined number of iterations (in this case 100 iterations), the output from the sum-product decoder is hard limited and constitutes the final output of the RAC-CPM system.

Chapter 4

Simulation Results

In this chapter, we present the BER performance of SCC-CPM over the AWGN channel. In order to compare and contrast the performance differences between the systems built here, we do the following:

- compare the performance variations of SCC-CPM due to SOQPSK-TG and PCM/FM.
- study the effects of coherent and noncoherent demodulation of SCC-CPM.
- compare the performances of $(5, 7)$ convolutional code (CC1) and $(27, 31)$ convolutional code (CC2) with CPM.
- describe BER performances due to CC, TPC, and RAC with CPM.
- compare our TPC-CPM against a similar system developed in [9].
- describe the performance of RAC-CPM.
- document the variations in performance due to the variations in system parameters like the input block size, and the number of decoding iterations.

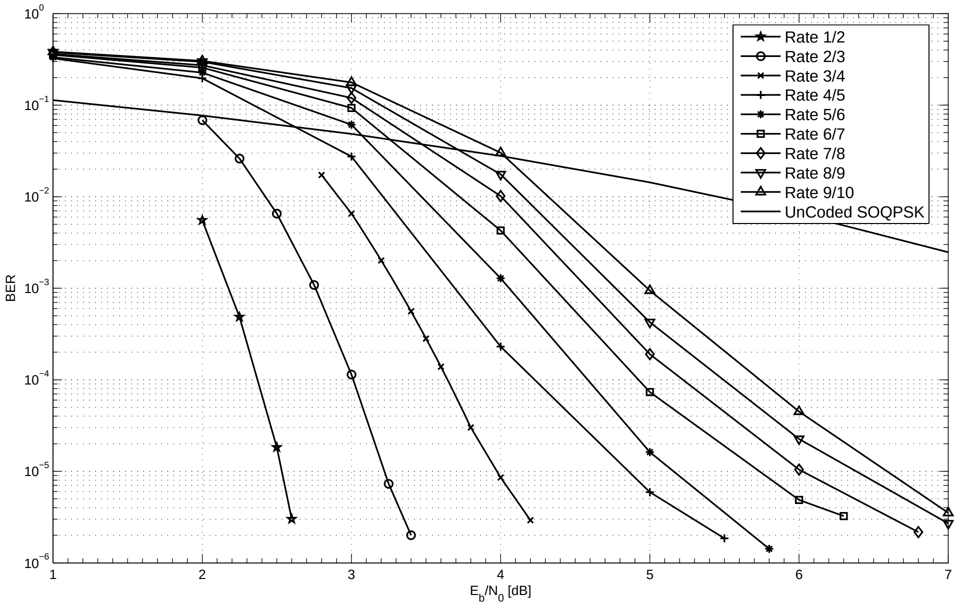

Figure 4.1. BER Performance of Convolutional Code 1 with Coherent SOQPSK-TG.

4.1 SOQPSK-TG vs. PCM/FM

The BER performance of coded SOQPSK-TG and PCM/FM with CC1 and various coding rates is shown in Figures 4.1 and 4.2, respectively. We measure the coding gains of these two schemes at the BER $= 10^{-5}$ crossing point. By way of reference, uncoded SOQPSK-TG crosses BER $= 10^{-5}$ at $E_b/N_0 = 10.56$ dB and uncoded PCM/FM crosses BER $= 10^{-5}$ at $E_b/N_0 = 8.44$ dB [32] (E_b/N_0 denotes the bit energy to noise power spectral density ratio). From Figure 4.1, we can see that a rate 1/2 CC1 with SOQPSK-TG attains a BER $= 10^{-5}$ at $E_b/N_0 = 2.6$ dB. In comparison, a rate 1/2 CC1 with PCM/FM reaches a similar BER at a $E_b/N_0 = 1.8$ dB. Similarly at a higher rate 7/8, CC1 with SOQPSK-TG and PCM/FM attains a BER $= 10^{-5}$ at $E_b/N_0 = 6.0$ dB and $E_b/N_0 = 3.8$ dB, respectively. So, based upon BER performances, it would be justified to say that coded PCM/FM is a power efficient coded modulation compared to coded SOQPSK-TG. However, there is more to system design than just BER

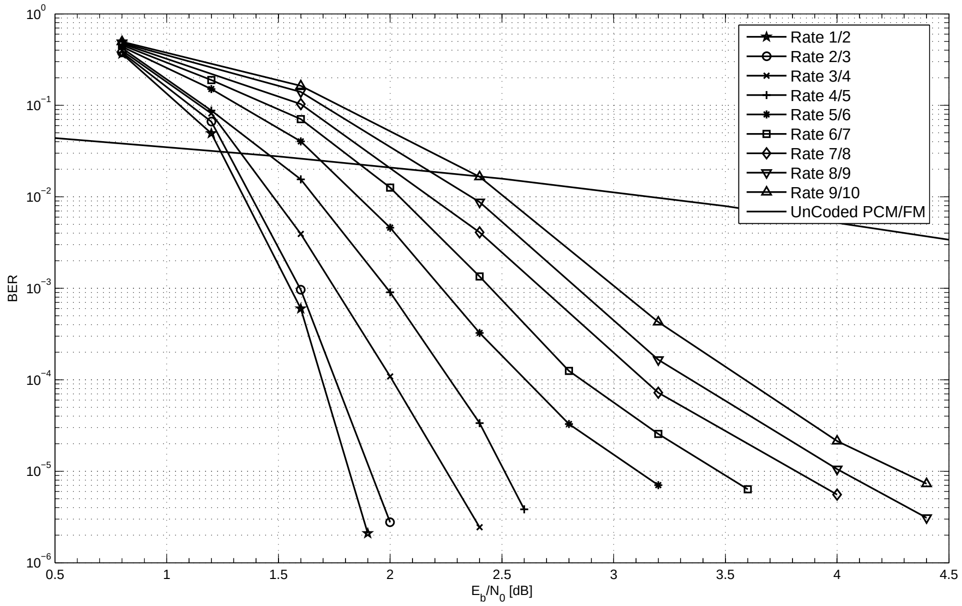

Figure 4.2. BER Performance of Convolutional Code 1 with Coherent PCM/FM.

performance. So, a better design criteria would be to realize a trade-off between BER performance and spectral efficiency. Among the two modulation techniques considered here, SOQPSK-TG is 2 times more bandwidth efficient[1] than PCM/FM. Hence at a rate 1/2, coded SOQPSK-TG, with its inherent bandwidth efficiency, and because, its BER performance at a BER $= 10^{-5}$ is only 0.8 dB worse than that of coded PCM/FM, is a good coded modulation technique for aeronautical telemetry.

Extensive simulations confirm similar performances shown by SOQPSK-TG and PCM/FM when they were coupled with CC2 and TPC. The performance of coded SOQPSK-TG and PCM/FM with CC2 is shown in Figures 4.3 and 4.4, respectively. As seen from these figures, a rate 1/2 SCCC-SOQPSK-TG attains a BER $= 10^{-5}$ at $E_b/N_0 = 2.7$ dB. Likewise, a rate 1/2 SCCC-PCM/FM reaches a BER $= 10^{-5}$ at a $E_b/N_0 = 2.1$ dB. Similar to CC1 with CPM, CC2s with SOQPSK-TG and PCM/FM,

[1] Bandwidth efficiency is measured at 60 dB down from an unmodulated carrier, as is the custom in aeronautical telemetry.

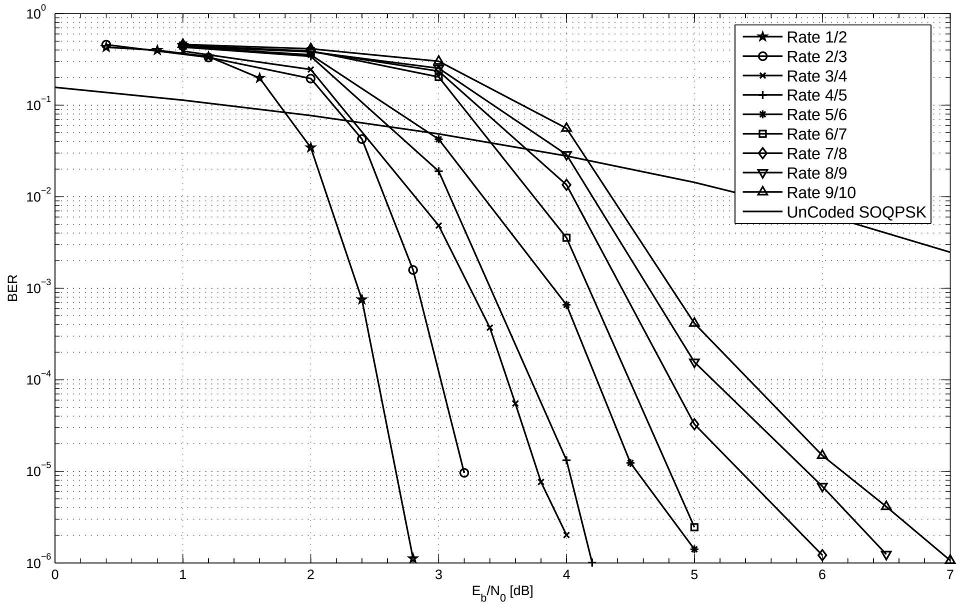

Figure 4.3. BER Performance of Convolutional Code 2 with Coherent SOQPSK-TG.

at a higher rate 7/8 attains a BER $= 10^{-5}$ at a $E_b/N_0 = 5.4$ dB and $E_b/N_0 = 3.8$ dB, respectively. Similarly, at a rate 0.7932, TPCs with SOQPSK-TG and PCM/FM attain a BER $= 10^{-5}$ at a $E_b/N_0 = 6.1$ dB and $E_b/N_0 = 4.4$ dB, respectively.

From these BER performance results, it becomes clear that coded PCM/FM is a better power efficient scheme compared to coded SOQPSK-TG. However, because of its bandwidth efficiency, and because its BER performance is close to the BER performance of coded PCM/FM, coded SOQPSK-TG gains significance in the field of aeronautical telemetry.

From the coding gains listed in Table 4.1, we see that coded PCM/FM has more power efficiency, while coded SOQPSK-TG, with its bandwidth efficiency and BER performance, offers a good compromise between power and spectral efficiency.

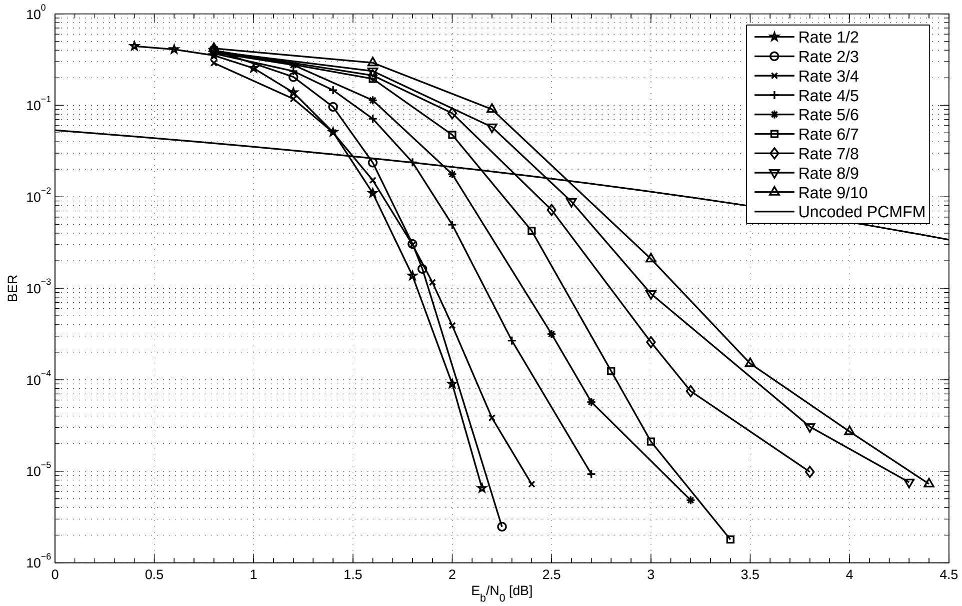

Figure 4.4. BER Performance of Convolutional Code 2 with Coherent PCM/FM.

Table 4.1. BER Performances of coded CPM.

Modulation	Coding	Code Rate	BER = 10^{-5}	Gain in dB	Reference
SOQPSK-TG	CC1	1/2	2.6	8.0 dB	Figure 4.1
SOQPSK-TG	CC1	7/8	6.0	4.6 dB	Figure 4.1
PCM/FM	CC1	1/2	1.8	6.6 dB	Figure 4.2
PCM/FM	CC1	7/8	3.8	4.6 dB	Figure 4.2
SOQPSK-TG	CC2	1/2	2.7	7.9 dB	Figure 4.3
SOQPSK-TG	CC2	7/8	5.4	5.2 dB	Figure 4.3
PCM/FM	CC2	1/2	2.1	6.3 dB	Figure 4.4
PCM/FM	CC2	7/8	3.8	4.6 dB	Figure 4.4
SOQPSK-TG	TPC	0.7932	6.1	4.5 dB	Figure 4.5
PCM/FM	TPC	0.7932	4.4	4.0 dB	Figure 4.6

4.2 Coherent Demodulation vs. Noncoherent Demodulation

The BER performances of coherent and noncoherent demodulation of SCCC-SOQPSK-TG with CC1 are shown in Figures 4.1 and 4.7, respectively. The *forgetting factor* parameter of the noncoherent demodulator is selected as 0.875 and the standard deviation of phase noise was set at 2° [32]. From these figures, we see that noncoherent

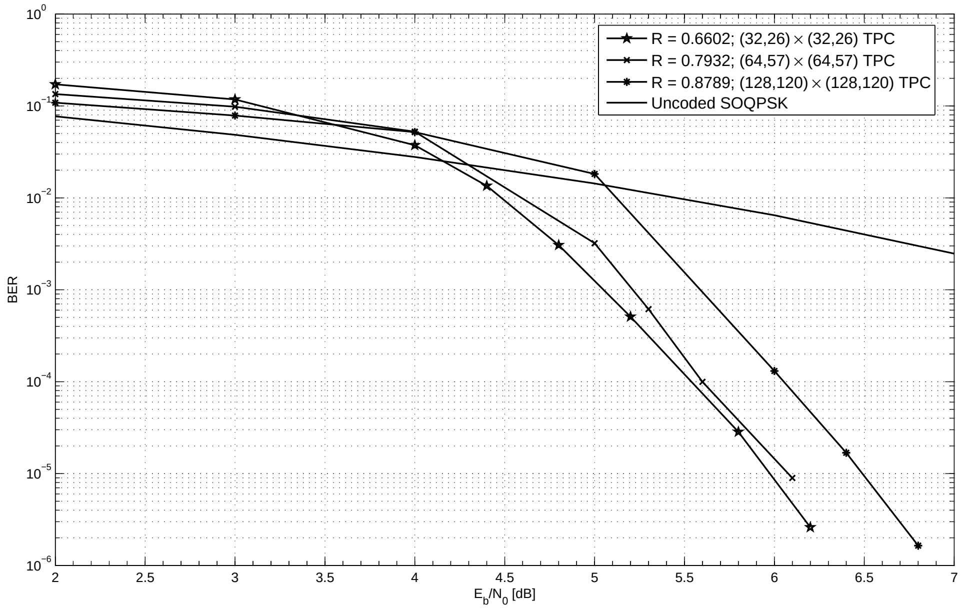

Figure 4.5. BER Performance of Turbo-Product Code with Coherent SOQPSK-TG.

demodulation of SCCC-SOQPSK-TG is about 1 dB worse than coherent demodulation of SCCC-SOQPSK-TG at BER $= 10^{-5}$. For instance, rate 1/2 SCCC-SOQPSK-TG with coherent demodulation yields a coding gain of 8.0 dB, which is 1.0 dB more than the gain produced by a similar system with noncoherent demodulation. Similar differences in performance are also evident at higher code rates. This is a good tradeoff between complexity and performance since noncoherent demodulation reduces the synchronization complexity of the receiver. The performance of SCC-CPM systems under noncoherent demodulation with varing *forgetting factors* is shown in the appendix.

Similar to the BER performance shown by SCCC-SOQPSK-TG under noncoherent demodulation, the performance of other SCC-CPM systems under noncoherent demodulation is listed in Table 4.2. For comparison, the performance of these systems under coherent demodulation is tabulated. From this table, it can be seen that the performance difference between coherent and non-coherent demodulation for any SCC-CPM system

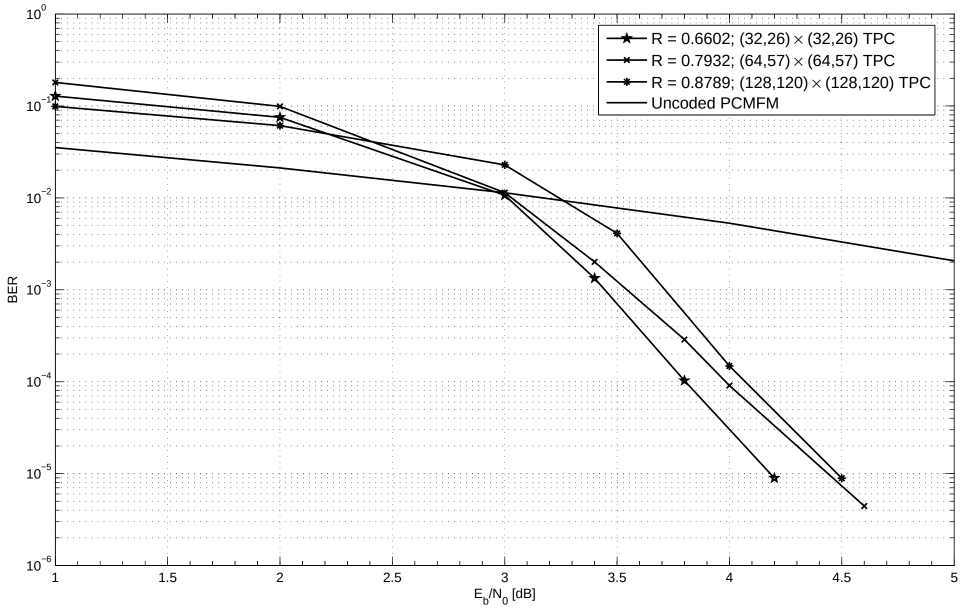

Figure 4.6. BER Performance of Turbo-Product Code with Coherent PCM/FM.

Table 4.2. BER Performances of coded CPMs under Coherent and Non-Coherent Demodulation.

Modulation	Coding	Gain in CD	Reference	Gain in NCD	Reference	Overall Gain
SOQPSK-TG	CC1	8.0 dB	Figure 4.1	7.0 dB	Figure 4.7	1.0 dB
PCM/FM	CC1	6.6 dB	Figure 4.2	5.9 dB	Figure 4.8	0.7 dB
SOQPSK-TG	CC2	7.9 dB	Figure 4.3	6.7 dB	Figure 4.9	1.2 dB
PCM/FM	CC2	6.3 dB	Figure 4.4	5.7 dB	Figure 4.10	0.5 dB
SOQPSK-TG	TPC	4.5 dB	Figure 4.5	3.2 dB	Figure 4.11	1.3 dB
PCM/FM	TPC	4.0 dB	Figure 4.6	3.2 dB	Figure 4.12	0.8 dB

is around 1.0 dB. This difference in performance shown by noncoherent demodulators can be tolerated considering the receiver complexity reductions provided by them.

4.3 CC1 vs. CC2

Figures 4.1 and 4.3 show the BER performances of SOQPSK-TG with CC1 and CC2, respectively. At a rate 1/2, the performances shown by CC1 and CC2 with SOQPSK-TG are approximately the same, as they produce a coding gain of 8.0 dB and 7.9 dB, respectively. But at a higher code rate, the gains produced by CC2 with

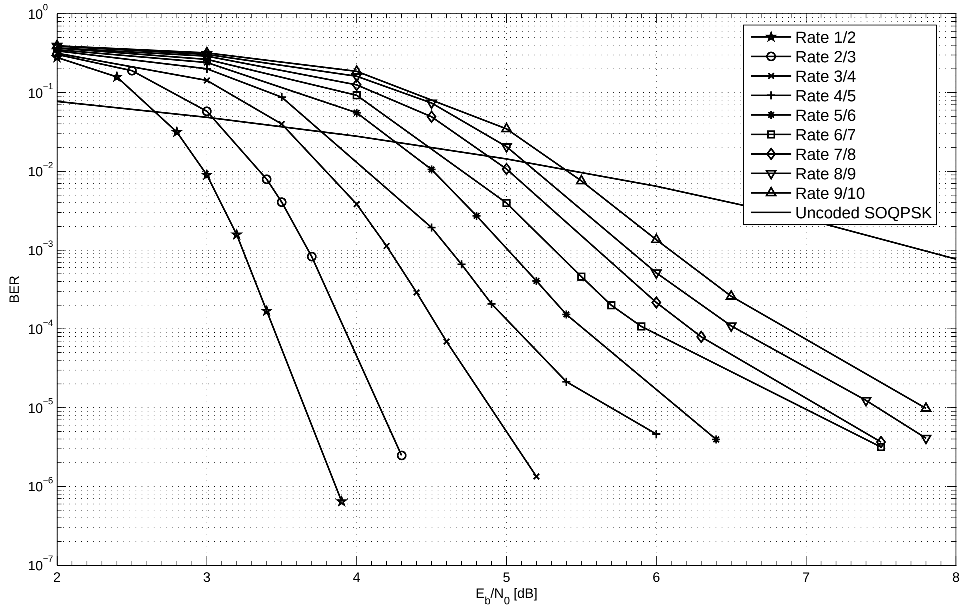

Figure 4.7. BER Performance of Convolutional Code 1 with Non-Coherent SOQPSK-TG.

CPM are more than the gains provided by CC1 with CPM. For instance, a rate 4/5 CC2 with SOQPSK-TG yields a gain of 6.6 dB, this is 0.9 dB better than the gain produced by CC1 with SOQPSK-TG.

Similarly, the BER performances of PCM/FM with CC1 and CC2 are shown in Figures 4.2 and 4.4, respectively. From these figures, we can see that, at lower code rates CC1 with PCM/FM performs better than CC2 with PCM/FM. A rate 1/2 CC1 with PCM/FM yields a gain of 6.6 dB, this is 0.3 dB better than the gain produced by CC2 with PCM/FM. However, at a higher code rate, the performances of CC1 and CC2 with PCM/FM are approximately same. For easier reference, the coding gains provided by CC1 and CC2 with CPM at a rate 1/2 are listed in Table 4.1.

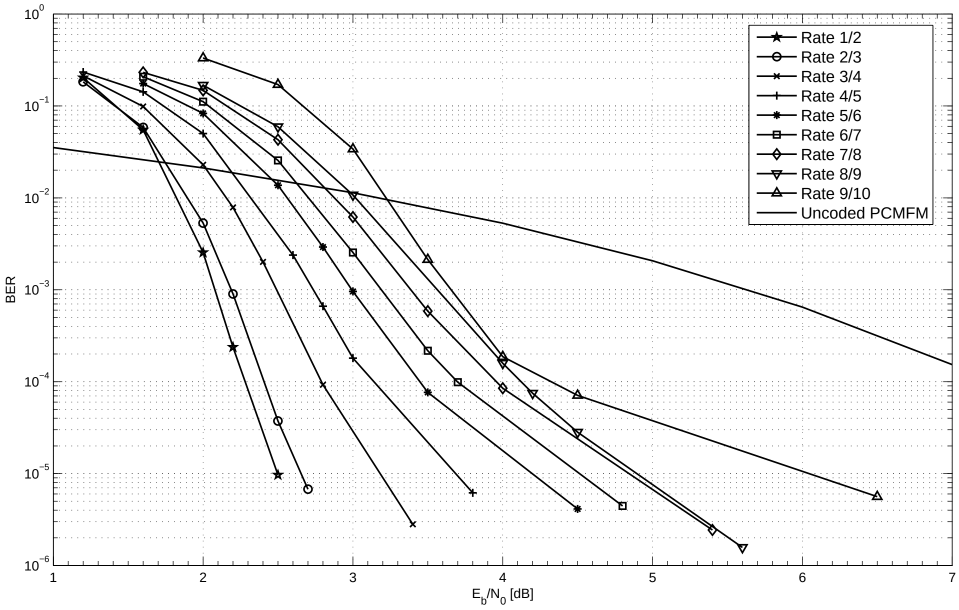

Figure 4.8. BER Performance of Convolutional Code 1 with Non-Coherent PCM/FM.

4.4 Performance of TPC-CPM

The performances of TPC with SOQPSK-TG and PCM/FM are shown in Figures 4.5 and 4.6, respectively. As we can see from these figures, TPC-PCM/FM is a power efficient scheme than TPC-SOQPSK-TG. However, TPC-SOQPSK-TG gains importance because of SOQPSK-TG's inherent bandwidth efficiency and also because its BER performance is only a little worse (at a higher rate 0.7932) than the BER performance of TPC-PCM/FM. A rate 0.7932 TPC-SOQPSK-TG yields a coding gain of 4.5 dB, this is 0.5 dB better than the gain provided by a similar code rate TPC-PCM/FM. This fact was earlier stated in Section 4.1. The performances of TPC-CPM under non-coherent demodulation are shown in Figures 4.11 and 4.12, respectively. This performance closely approximates the performance shown by these systems under coherent demodulation. This was earlier illustrated in Table 4.2.

As mentioned in Chapter 1, similar TPC-CPM systems using a non-CPM based

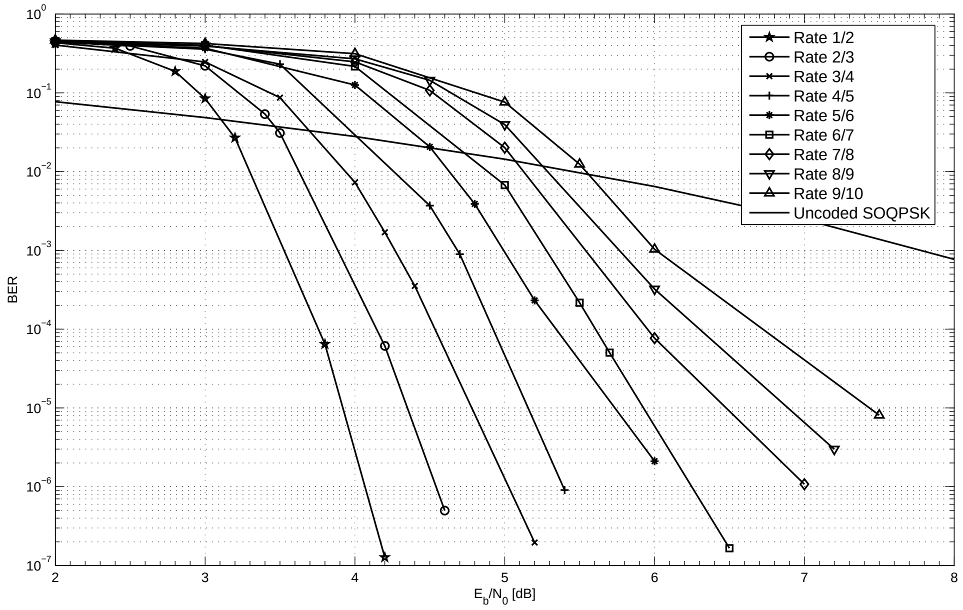

Figure 4.9. BER Performance of Convolutional Code 2 with Non-Coherent SOQPSK-TG.

Table 4.3. BER Performances of Similar TPC-CPM Systems.

Modulation	Coding	TPC-CPM built here	Reference	TPC-CPM built in [9]
SOQPSK-TG	TPC	4.5 dB	Figure 4.5	3.7 dB
PCM/FM	TPC	4.0 dB	Figure 4.6	3.4 dB

ad hoc approach were built earlier in [9]. Comparing the BER performance results of TPC-CPM systems developed in this thesis with the performance results shown in [9], we can see that the TPC-CPM systems developed here show better coding gain performances. For instance, at a rate 0.7932, the TPC-SOQPSK-TG system built here shows an improvement of 0.8 dB at BER $= 10^{-5}$ over a similar system developed in [9]. In the case of TPC-PCM/FM, the performance improvement of the system built here over the system built in [9] is 0.6 dB. This performance improvement shown by the TPC-CPMs developed here is attributed to the use of a near-optimal SISO algorithm for CPM demodulation instead of the ad hoc soft demodulation techniques used in [9]. The respective gains realized by the similar TPC-CPM systems at a rate 0.7932 is shown in

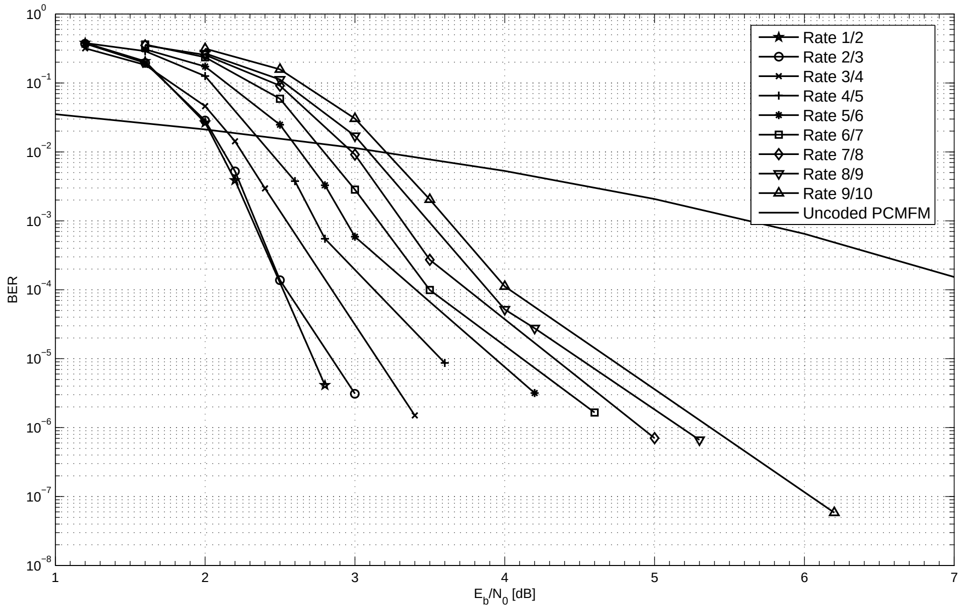

Figure 4.10. BER Performance of Convolutional Code 2 with Non-Coherent PCM/FM.

Table 4.3.

4.5 Performance of RAC-CPM

In addition to coupling CCs and TPCs with CPM, this thesis also combines RACs with CPM. The performance of RAC-SOQPSK-TG and RAC-PCM/FM is shown in Figures 4.13 and 4.14, respectively. From these figures, we see that a repetition $q = 3$ RAC-SOQPSK-TG yields a coding gain of 5.5 dB and a repetition $q = 4$ RAC-SOQPSK-TG provides a coding gain of 5.9 dB. Similarly with PCM/FM, a repetition $q = 3$ RAC yields a gain of 3.2 dB and a repetition $q = 4$ RAC provides a gain of 3.4 dB. Unlike CCs and TPCs with CPM, the BER performance of RAC-CPMs show that, at a rate 1/3, coded SOQPSK-TG is slightly more power efficient than coded PCM/FM, also coded SOQPSK-TG is better suited to the design criteria set for aeronautical telemetry.

The code rate of RAC-CPMs with $q = 3$ is 1/3 and with $q = 4$ is 1/4. It is because

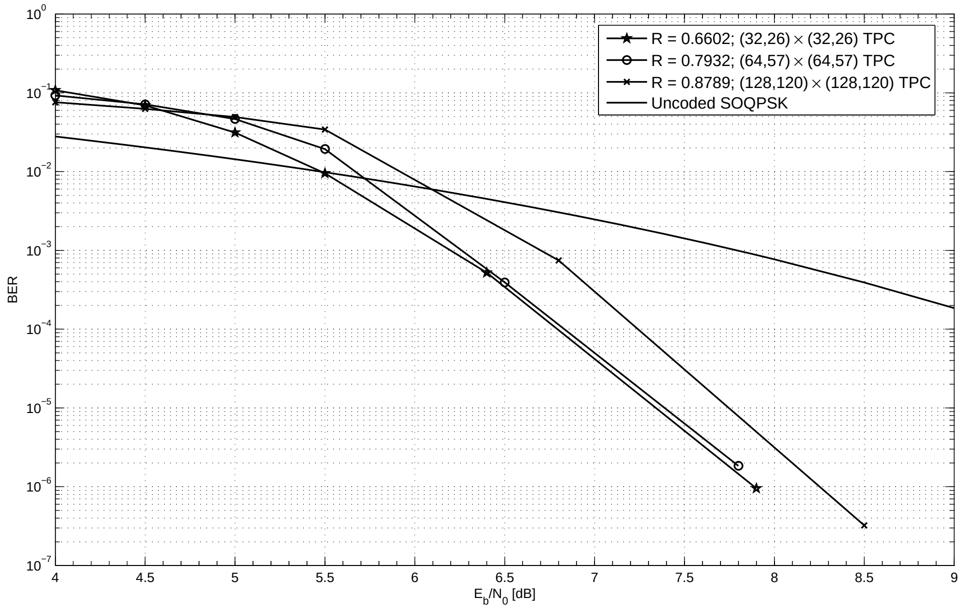

Figure 4.11. BER Performance of Turbo-Product Codes with Non-Coherent SOQPSK-TG.

Table 4.4. BER Performances of RAC-CPMs with SCCC-CPMs and TPC-CPMs.

Modulation	RAC	Reference	TPC	Reference	SCCC	Reference
SOQPSK-TG	5.5 dB	Figure 4.13	4.5 dB	Figure 4.5	5.8 dB	Figure 4.1
PCM/FM	3.2 dB	Figure 4.14	4.0 dB	Figure 4.6	5.9 dB	Figure 4.2

of these lower code rates, RAC-CPMs lose their significance compared to TPC-CPMs and SCCC-CPMs. The coding gain realized by a rate 1/3 RAC-SOQPSK-TG is higher than the gain reported by a rate 0.7932 TPC-SOQPSK-TG. However, due to their lower code rate their transmission bandwidth efficiency is reduced and hence they become less significant in aeronautical telemetry. Whereas with PCM/FMs, the gain produced by a rate 0.7932 TPC is 0.8 dB better than the gain produced by a lower code rate RAC. The BER performance of a rate 1/3 RAC-CPM, rate 4/5 SCCC-CPM and rate 0.7932 TPC-CPM is shown in Table 4.4.

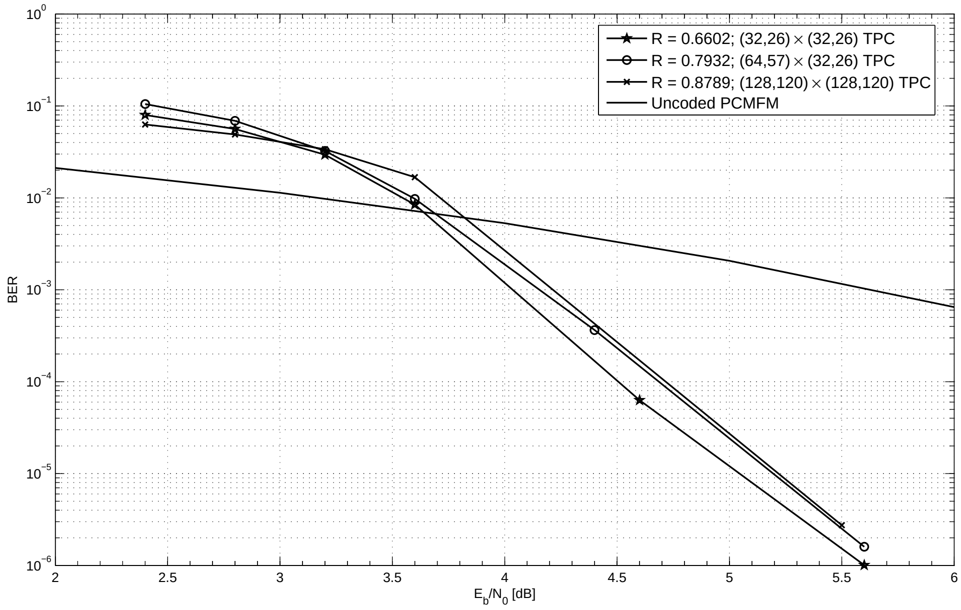

Figure 4.12. BER Performance of Turbo-Product Codes with Non-Coherent PCM/FM.

4.6 CCs vs. TPCs & RACs

In general, the performances of CC1 and CC2 with CPM are better than the performance of TPC with CPMs. As shown in Figures 4.1 and 4.3, a rate 4/5 CC1 and CC2 with SOQPSK-TG provide coding gains of 5.7 dB and 6.6 dB, respectively. In contrast, a rate 0.7932 TPC with SOQPSK-TG provides a coding gain of only 4.5 dB, as shown in Figure 4.5. From this we can see that, CCs with SOQPSK-TG performs better than TPCs with SOQPSK-TG.

Similarly, the performances of CC1 and CC2 with PCM/FM are shown in Figures 4.2 and 4.4, respectively. A rate 4/5 CC1 and CC2 with PCM/FM provide a coding gain of 5.9 dB and 5.7 dB, respectively. This is a better performance than a rate 0.7932 TPC-PCM/FM which yields a gain of only 4.0 dB. This is shown in Figure 4.6. Based on the coding gains reported above, we can say that CCs with CPM are a better choice for aeronautical telemetry than TPCs with CPM. For quick reference, this is shown in

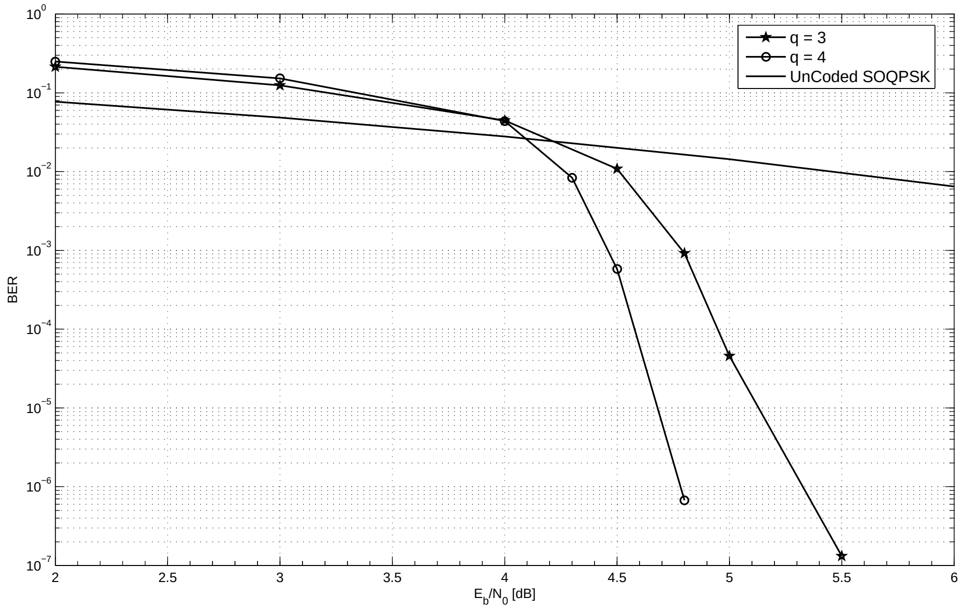

Figure 4.13. BER Performance of Repeat-Accumulate Codes with Coherent SOQPSK-TG.

Table 4.1. Similarly, Table 4.4 shows the coding gain superiority of SCCC-CPMs over RAC-CPMs.

4.7 BER Performance due to Increased Input Block Size

The BER performance of SCCC-CPM systems for a input (I/P) block of 1024 bits was described earlier in Section 4.1. This section describes the performance of these systems for a I/P block of 4096 bits. Figures 4.15 and 4.16, show the BER performances of CC1 with SOQPSK-TG and PCM/FM, respectively. As seen from these figures, a rate 1/2 SCCC-SOQPSK-TG yields a coding gain of 8.3 dB, this is 0.3 dB better than the gain realized by a similar system with an I/P block of 1024 bits. Likewise, a rate 1/2 SCCC-PCM/FM, shown in Figure 4.16, yields a coding gain of 6.9 dB, this again is 0.3 dB better than the gain realized by a similar system with a 1024 bit I/P block. Similar performances are seen with higher code rate SCC-CPMs when the I/P block

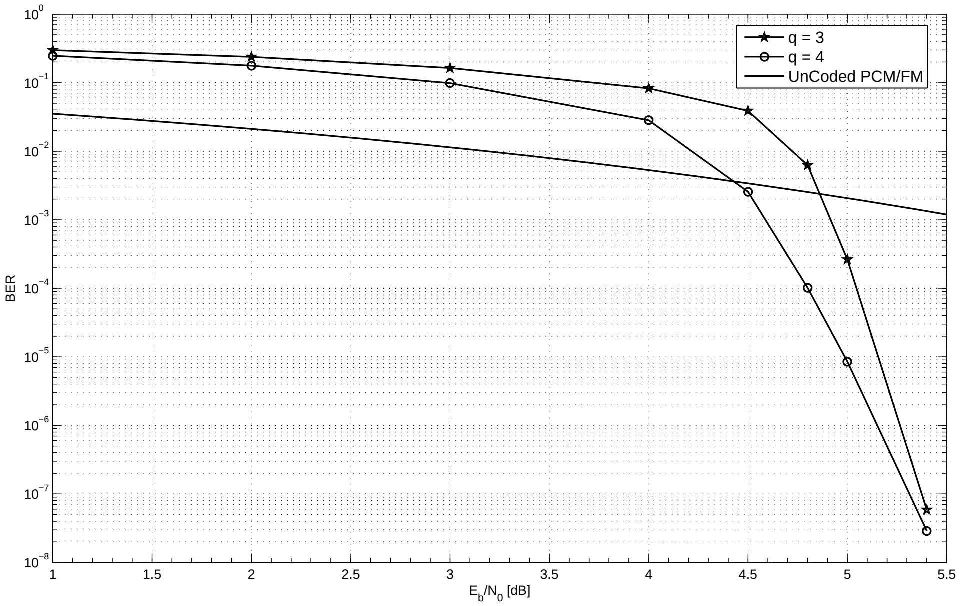

Figure 4.14. BER Performance of Repeat-Accumulate Codes with Non-Coherent PCM/FM.

Table 4.5. BER Performances of SCC-CPMs under Varying Input Block Size.

Modulation	Coding	1024 bit I/P	Reference	4096 bit I/P	Reference	Additional Gain
SOQPSK-TG	CC1	8.0 dB	Figure 4.1	8.3 dB	Figure 4.15	0.3 dB
PCM/FM	CC1	6.6 dB	Figure 4.2	6.9 dB	Figure 4.16	0.3 dB
SOQPSK-TG	CC2	7.9 dB	Figure 4.3	8.3 dB	Figure 4.17	0.4 dB
PCM/FM	CC2	6.3 dB	Figure 4.4	6.8 dB	Figure 4.18	0.5 dB

size is varied between 1024 bits and 4096 bits.

This increase in performance shown by SCCC-CPM systems, with an increase in block size, is justified as Benedetto in his 1998 paper [2] clearly stated that the I/P block size of an outer code in a SCC system should be large. Similar performances were seen when a rate 1/2 CC2 was combined with SOQPSK-TG and PCM/FM. As seen in Figure 4.17, the gain produced by this SCCC-SOQPSK-TG system is 8.3 dB, this is 0.4 dB more than the gain reported by a similar system described in Section 4.1. Also, a SCCC-PCM/FM shown in Figure 4.18 produces a gain of 6.8 dB. This again is 0.5 dB better than the gain reported by a similar system with a 1024 bit I/P block. For

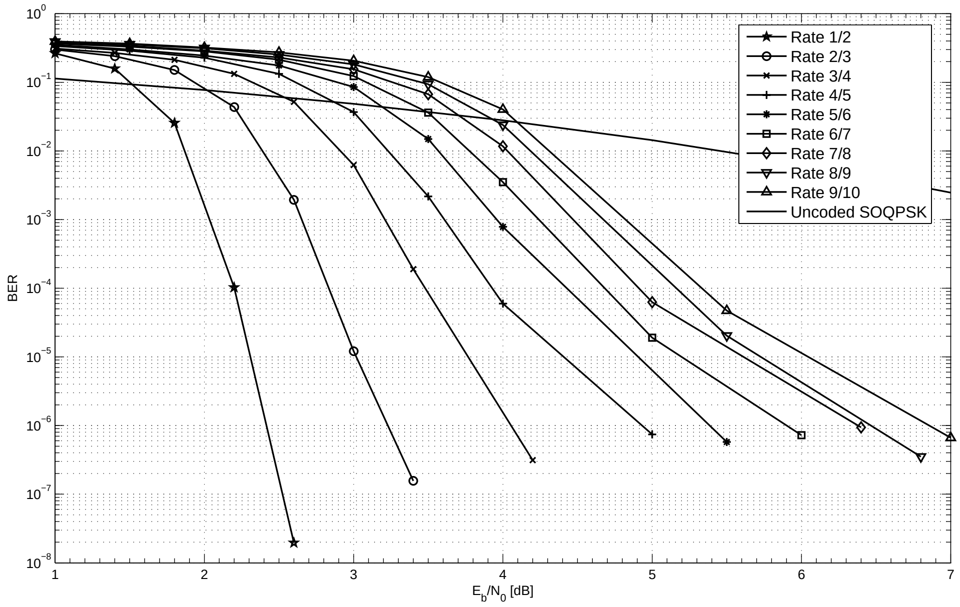

Figure 4.15. BER Performance of Convolutional Code 1 with Coherent SOQPSK-TG for a Input Block of 4096 Bits.

convenience, the coding gains reported by a rate 1/2 SCCC-CPM system with a 1024 bit I/P block and a 4096 bit I/P block is shown in Table 4.5.

4.8 BER Performance due to Increased Number of Decoding Iterations

This section describes the BER performance of SCCC-CPM systems under 10 decoding iterations. This means that the *posteriori* probabilities between the CPM SISO demodulator and the convolutional SISO decoder are exchanged 10 times before the decoded output is hard limited and given as a final decoded information. Figures 4.19 and 4.20, show the performance of CC1 with SOQPSK-TG and PCM/FM, respectively. As shown in these figures, a rate 1/2 SCCC-SOQPSK-TG provides a gain of 8.7 dB, this is 0.4 dB better than the gain produced by a similar system with 5 decoding iterations. Likewise, a rate 1/2 SCCC-PCM/FM produces a gain of 7.2 dB, this is 0.3 dB

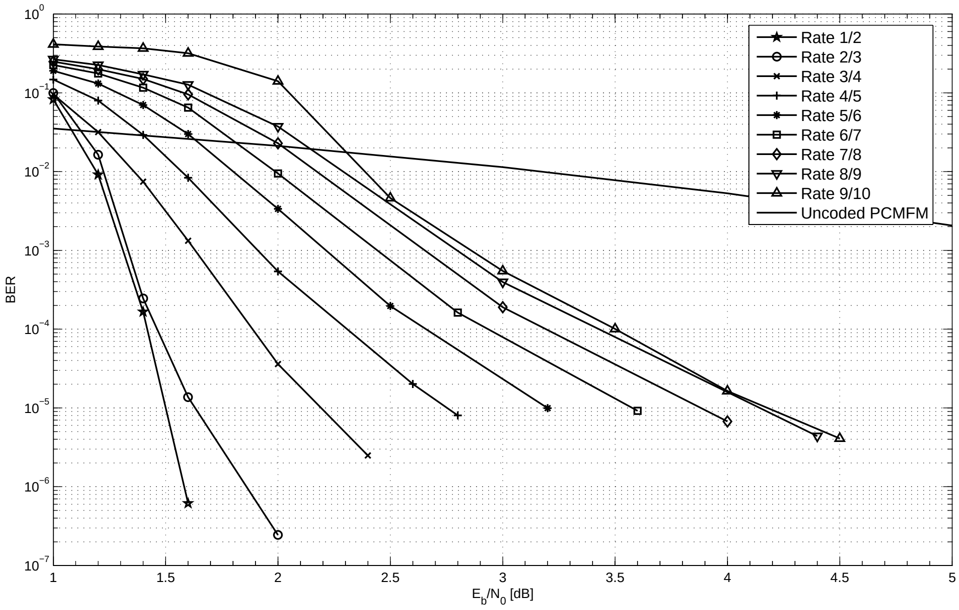

Figure 4.16. BER Performance of Convolutional Code 1 with Coherent PCM/FM for a Input Block of 4096 Bits.

Table 4.6. BER Performances of SCC-CPMs under Varying Decode Iterations.

Modulation	Coding	10 Decode	Reference	5 Decode	Reference	Additional Gain
SOQPSK-TG	CC1	8.7 dB	Figure 4.19	8.3 dB	Figure 4.15	0.4 dB
PCM/FM	CC1	7.2 dB	Figure 4.20	6.9 dB	Figure 4.16	0.3 dB
SOQPSK-TG	CC2	8.4 dB	Figure 4.21	8.3 dB	Figure 4.17	0.1 dB
PCM/FM	CC2	7.0 dB	Figure 4.22	6.8 dB	Figure 4.18	0.2 dB

better than the gain produced by a similar SCCC-PCM/FM with 5 decoding iterations. Similar performances can also be seen at higher code rate SCC-CPMs.

A similar performance improvement was realized when CC2 was coupled with CPM. The performance of this SCCC-SOQPSK-TG and SCCC-PCM/FM under 10 decoding iterations is shown in Figures 4.21 and 4.22, respectively. As seen from the figures, a rate 1/2 CC2 with SOQPSK-TG and PCM/FM produces a gain of 8.4 dB and 7.0 dB respectively. This is 0.1 dB and 0.2 dB better performance than their corresponding systems under 5 decoding iterations. Also similar performance raise can be realized at higher code rates. This is shown in the figures considered above.

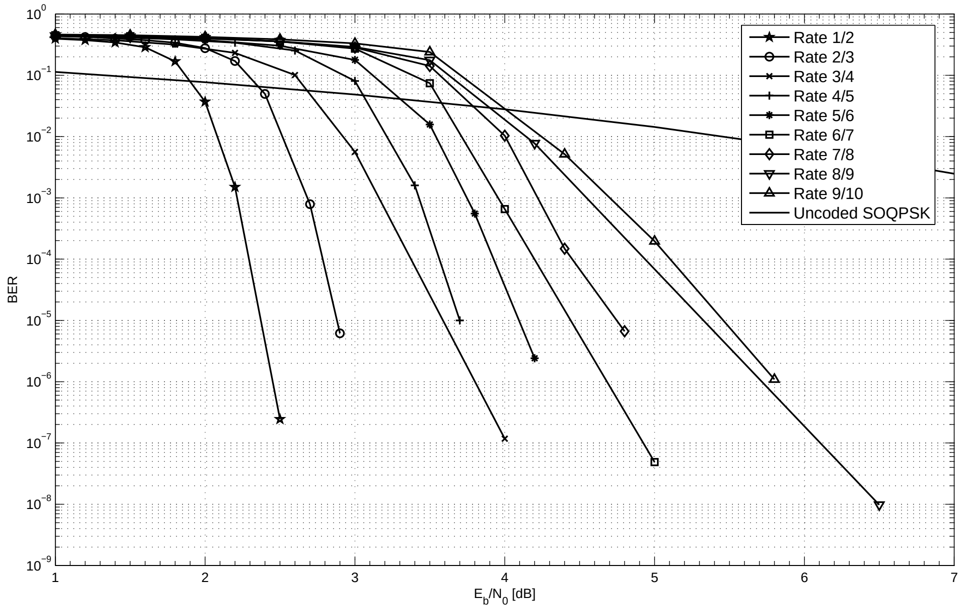

Figure 4.17. BER Performance of Convolutional Code 2 with Coherent SOQPSK-TG for a Input Block of 4096 Bits.

4.9 Theoretical vs. Practical Performance of coded CPM

From Figure 4.23, we see theoretical coding gain limits for any channel code with variable code rates over an AWGN channel. The maximum capacity (code rate) of this channel under hard and soft decision decoding, derived in [21], is given in Equations 4.1 and 4.2, respectively.

$$C = 1 + plog_2p + (1-p)log_2(1-p) \tag{4.1}$$

$$C \approx \frac{\gamma_b R_c}{ln2} \tag{4.2}$$

From this figure, it is easily seen that, as the code rate tends to zero the coding gain difference between soft and hard decision decoding increases. In limit, as the code rate approaches 0, this difference attains a maximum of 2 dB [21]. In this thesis, as we are

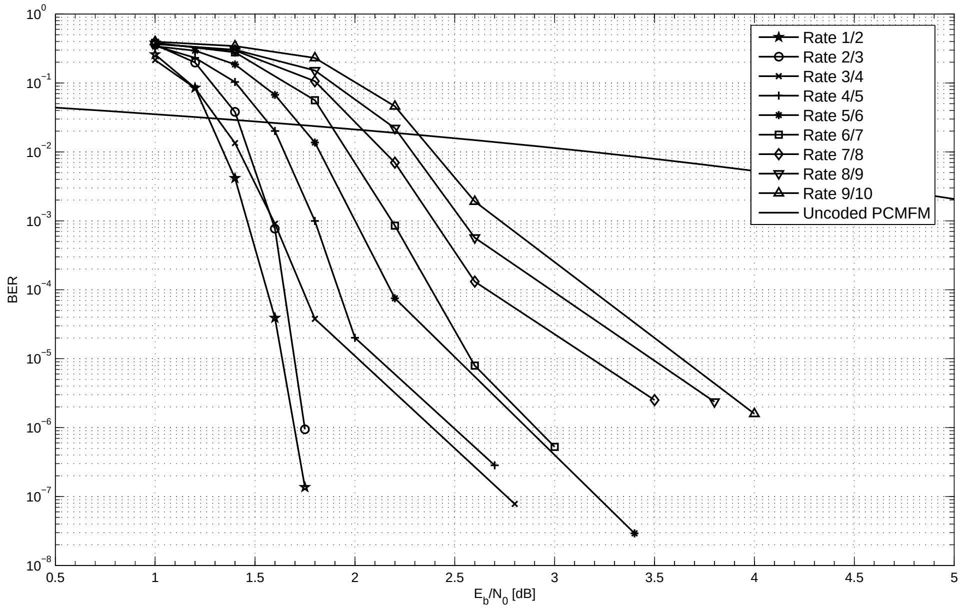

Figure 4.18. BER Performance of Convolutional Code 2 with Coherent PCM/FM for a Input Block of 4096 Bits.

concerned only about soft-input soft-output decoders, we compare the coding gains realized by these decoders at variable code rates with the theoretical coding gains reported for soft decision decoding in Figure 4.23. This comparison for coded SOQPSK-TG and coded PCM/FM is shown in Figures 4.24 and 4.25.

Figure 4.24 shows the coding gains realized by variable code rate CCs and TPCs with SOQPSK-TG. This figure compares the coding gains realized by these coded SOQPSK-TGs against the theoretical coding gains under soft decision decoding. As seen from this figure, at lower code rates (like rate 1/2 and 2/3) both CC1 and CC2 provide equal coding gains. As the code rate increases, the gains produced by CC2 significantly increase, thereby taking us closer to the limit. This improved performance has been reported earlier in Sec 4.3. This performance improvement achieved by CC2 over CC1 is primarily because of the increase in constraint length of the code. Also it is seen from this figure that both CC1 and CC2 outscore the performance of TPCs with

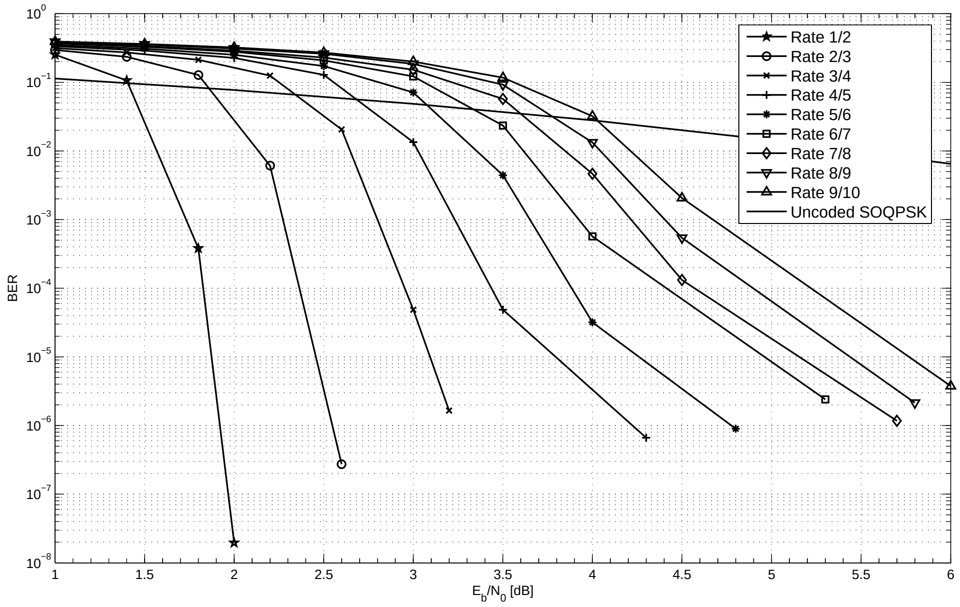

Figure 4.19. BER Performance of Convolutional Code 1 with Coherent SOQPSK-TG for a Input Block of 4096 Bits and 10 Decoding Iterations.

SOQPSK-TG.

Similarly the coding gains realized by CC1, CC2, and TPCs with PCM/FM are shown in Figure 4.25. As seen from this figure, at lower code rates, CC1 outperforms CC2 in terms of coding gains realized. However at higher rates (like rate 7/8, 8/9, 9/10), CC2 performs slightly better than CC1. This is evident from the figure considered here. Also it is easily seen that both CCs outscore TPCs with PCM/FM. Thus it can be concluded that, except at very high code rates, CC1 with PCM/FM takes us closer to the theoretical coding gain limit.

4.10 Key Observations and Recommendations

Based upon the simulation results presented in this chapter, we can infer that

• Coded PCM/FM is a power efficient modulation technique (at the cost of spectral efficiency), whereas coded SOQPSK-TG offers a good trade-off between power and

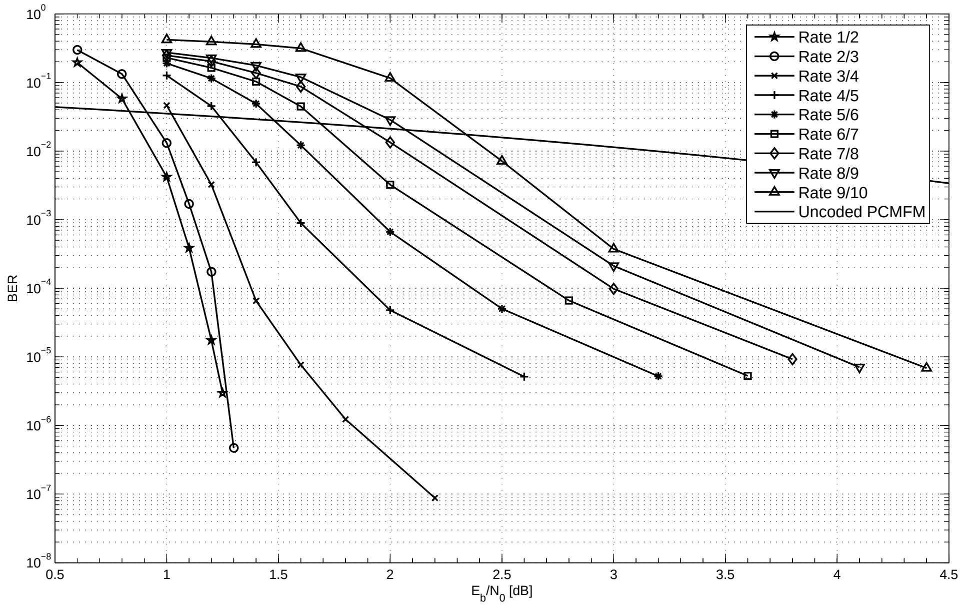

Figure 4.20. BER Performance of Convolutional Code 1 with Coherent PCM/FM for a Input Block of 4096 Bits and 10 Decoding Iterations.

spectral efficiencies. (Table 4.1).

- noncoherent demodulation of SCC-CPM performs within 1 dB of coherent demodulation (Table 4.2).
- With SOQPSK-TG, CC2 outperforms CC1 at higher code rates; with PCM/FM, CC1 outperforms CC2 at lower code rates. (Table 4.1).
- CCs with CPM, in general outperform TPCs and RACs with CPM (Table 4.1 and Table 4.4).
- TPC-CPM developed in this thesis outperforms the system developed in [9] (Table 4.3).
- with the increase in the input block size and with the increase in the number of decoding iterations, the performance of SCC-CPMs increase (Table 4.5 and Table 4.6).

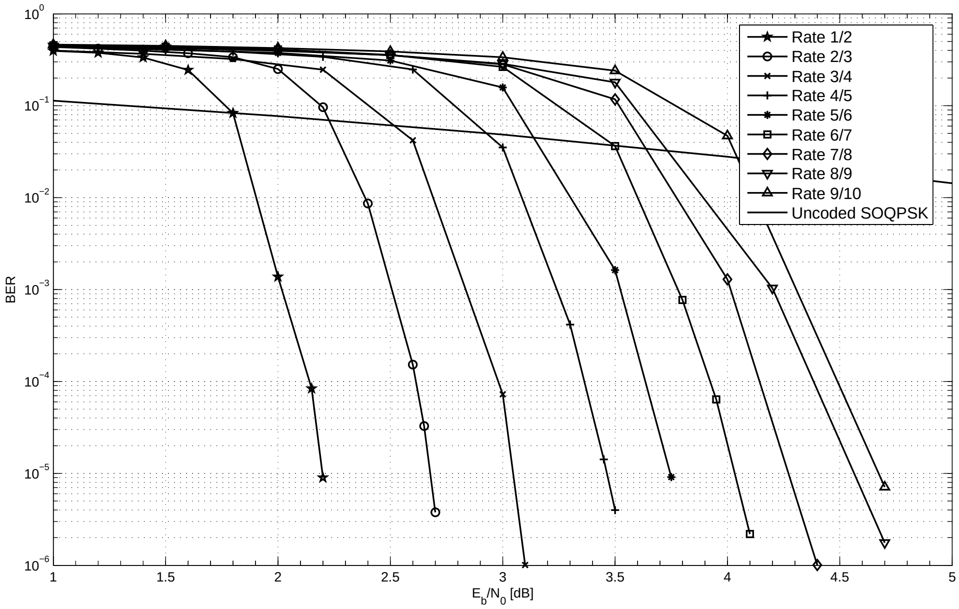

Figure 4.21. BER Performance of Convolutional Code 2 with Coherent SOQPSK-TG for a Input Block of 4096 Bits and 10 Decoding Iterations.

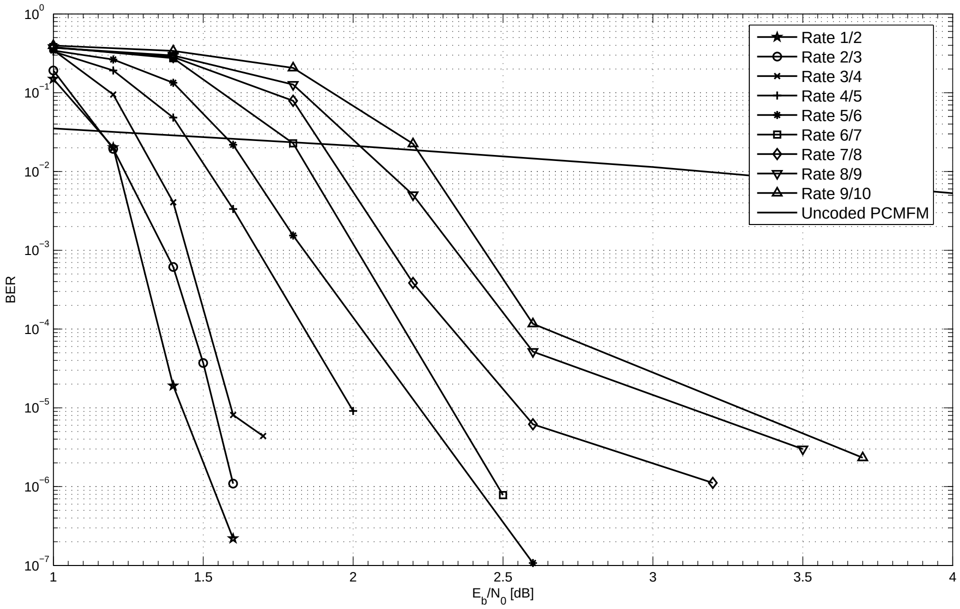

Figure 4.22. BER Performance of Convolutional Code 2 with Coherent PCM/FM for a Input Block of 4096 Bits and 10 Decoding Iterations.

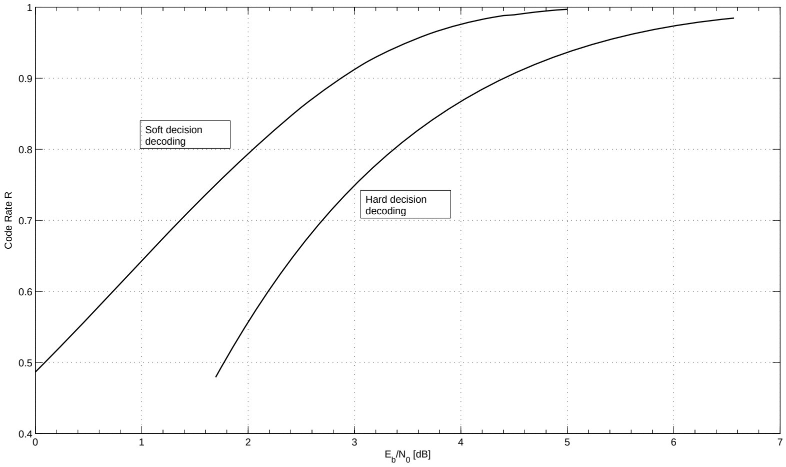

Figure 4.23. Shannon’s Soft Vs. Hard Decision Decoding.

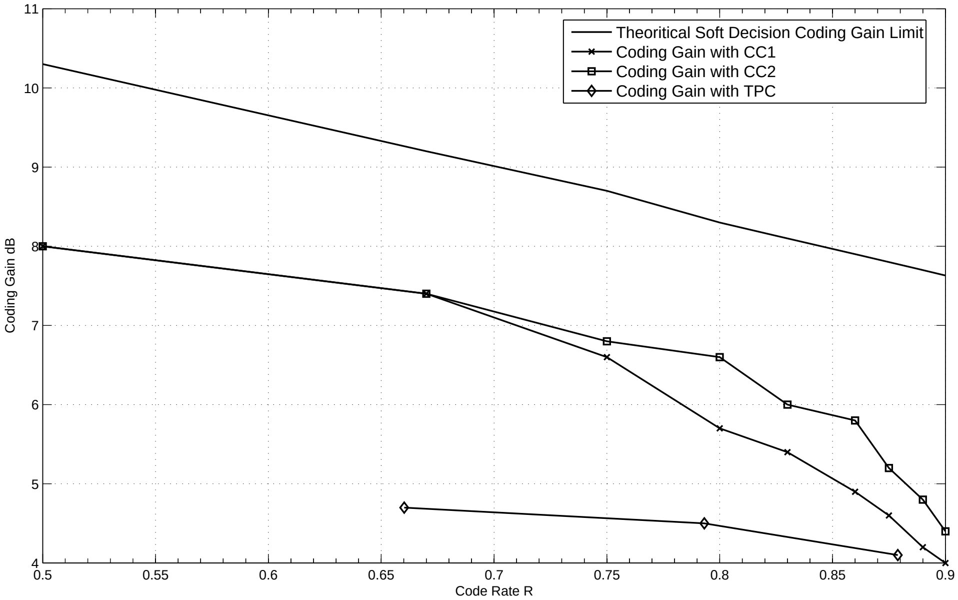

Figure 4.24. Comparison of Coding Gain Performance of Coded SOQPSK-TG against Shannon’s Soft Decision Decoding.

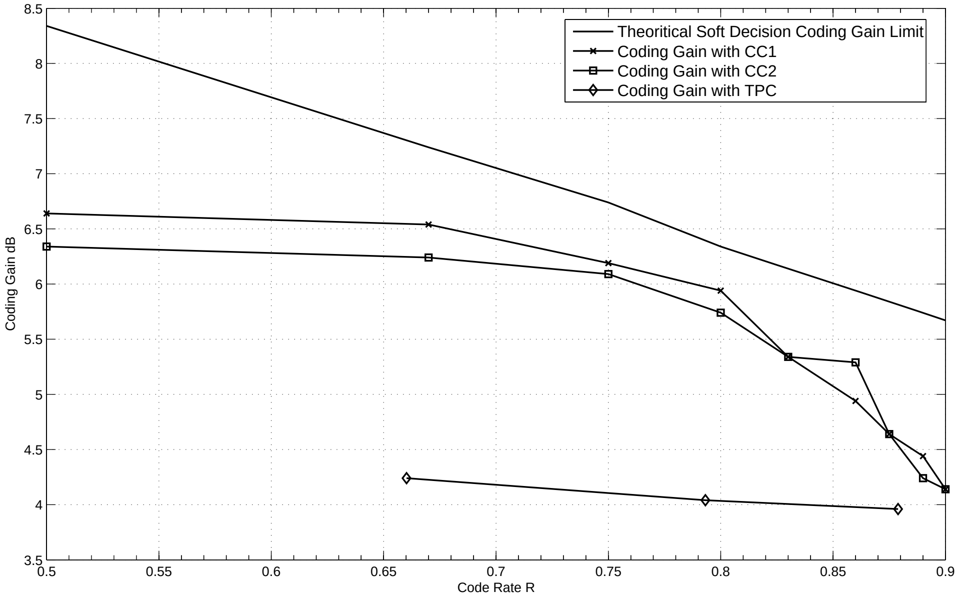

Figure 4.25. Comparison of Coding Gain Performance of Coded PCM/FM against Shannon's Soft Decision Decoding.

Chapter 5

Conclusion and Future Work

This thesis considered SCC-CPM systems with iterative demodulation and decoding, where the inner modulation was SOQPSK-TG or PCM/FM. These systems consist of SISO algorithms each for the inner modulation and the outer code, which pass soft probabilities to each other in an iterative manner.

Based on the simulation results given in Section 4.1, it is concluded that SOQPSK-TG yields larger coding gains than PCM/FM. This advantage comes in addition to the fact that SOQPSK-TG has twice the spectrum efficiency of PCM/FM. The simulation results shown in Section 4.2 also demonstrated that noncoherent demodulation results in a small loss on the order of 1 dB, which is an attractive trade for simplified synchronization requirements at the receiver. The relative performance of the two SCCCs was compared with each other and also with a turbo-product code and a repeat-accumulate code. It was also found that the convolutional codes resulted in larger coding gains than the turbo-product code and the repeat-accumulate code. Further it was also found that the relative ranking between the convolutional codes was a function of code rate. In this thesis, based on the numerical results shown in Section 4.7 and 4.8, it is shown that the performance of the SCCC-CPM system improves with an increase in the size of the input information block and also with an increase in the number of decoding iterations. In general, this thesis establishes the fact that SCC-CPM using the modulations currently used in aeronautical telemetry resulted in large coding gains and could be implemented with practical levels of complexity.

Future work will include an exhaustive search among a huge number of SCC systems to come up with an optimal combination of coding and CPM for a given bandwidth efficiency and decoder complexity. Similar to the SCC-CPM systems developed in this thesis, which combines CCs, TPCs, and RACs with SOQPSK-TG and PCM/FM, low-density parity-check (LDPC) codes can be optimally combined with CPM to develop a serially concatenated LDPC-CPM system. Also as mentioned in Chapter 1, spectrum reallocations of frequency bands prompted a migration away from PCM/FM and gave rise to the ARTM-CPM program [3]. Hence SCC-CPM systems, similar to the systems built here, can be built with ARTM-CPM as an inner code thereby developing a SCC-ARTM-CPM system.

Appendix A

Non-Coherent Demodulation

A.1 Convolutional Codes with CPM

This appendix describes the performance of CCs and TPCs with CPM under non-coherent demodulation. In this section, the performance of CCs with CPM is explained followed by a section on the performance of TPCs with CPM. Figures A.1 and A.2 show the BER performance of CC1 with SOQPSK-TG and PCM/FM, respectively. The *forgetting factor* parameter is set at 0.9375. It is to be noted that the *forgetting factor* parameter for coherent demodulation is 1. Phase noise with a standard deviation of 2° is introduced into the system. A rate 1/2 SCCC-SOQPSK-TG, shown in Figure A.1, yields a coding gain of 7.2 dB, this is 0.8 dB less than the gain realized by a similar coded SOQPSK system under coherent demodulation. Likewise, a coded PCM/FM under noncoherent demodulation produces a gain of 6.2 dB, this is 0.4 dB less than the gain produced by a similar coherent SCCC-PCM/FM. This amount of performance degradation is also seen at higher code rates. Even when CC2 is coupled with CPM similar performance degradation is realized. The coding gains realized by a rate 1/2 CC with CPM is tabulated in Table A.1. The performance of CC2 with CPM at higher

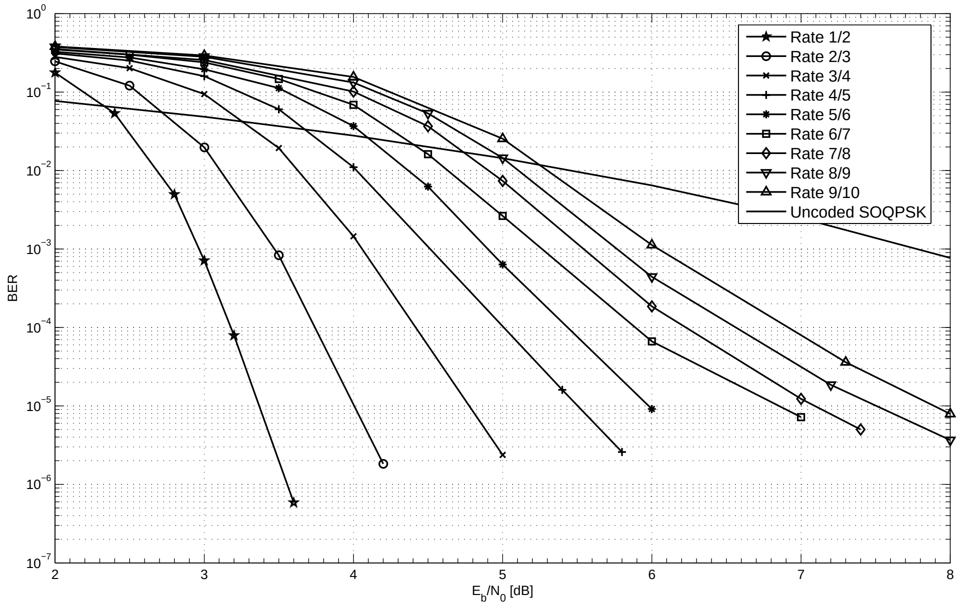

Figure A.1. BER Performance of Convolutional Code 1 with Non-Coherent SOQPSK-TG with a Forgetting Factor of 0.9375 and a 2° Standard Deviation of Phase Noise.

Table A.1. BER Performance of SCCC-CPM under Non-Coherent Demodulation with a Forgetting Factor of 0.9375 and a 2° Standard Deviation of Phase Noise.

Modulation	Coding	Coding Gain	Reference
SOQPSK-TG	CC1	7.2 dB	Figure A.1
PCM/FM	CC1	6.2 dB	Figure A.2
SOQPSK-TG	CC2	6.9 dB	Figure A.3
PCM/FM	CC2	5.9 dB	Figure A.4

code rates can be seen in Figures A.3 and A.4.

Again the performance of SCCC-CPM under noncoherent demodulation is considered and the performance of a rate 1/2 SCCC-CPM is tabulated in Table A.2. This time the *forgetting factor* is set at 0.875 and phase noise with a higher standard deviation of $5°$ is introduced into the system. Naturally, with the *forgetting factor* parameter going down from its ideal coherent value and with an increased phase noise, the performance of the system is bound to detoriate. This is evident from Figures A.5 and A.6 which show the BER performance of CC1 with CPM under noncoherent demodulation. A rate

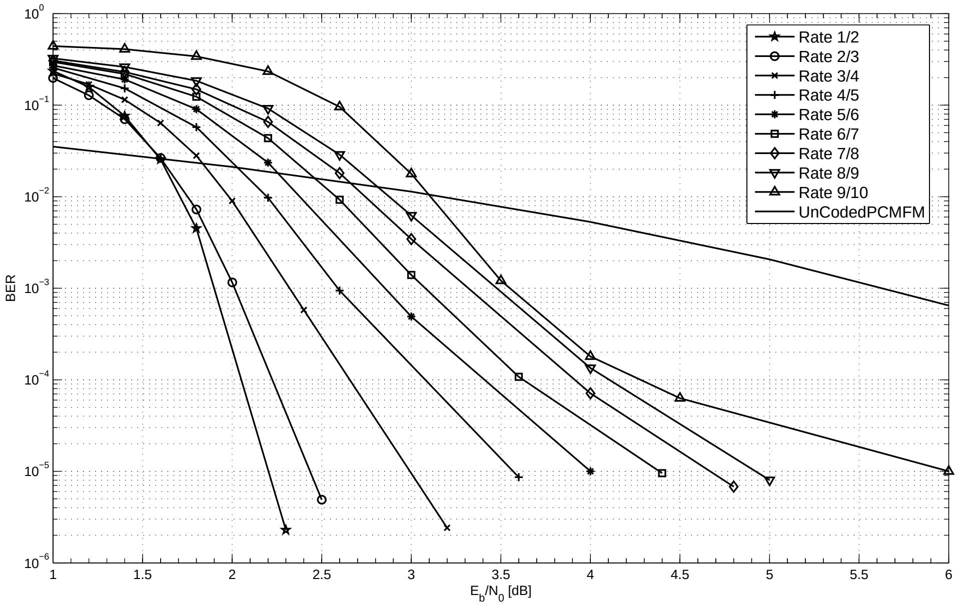

Figure A.2. BER Performance of Convolutional Code 1 with Non-Coherent PCM/FM with a Forgetting Factor of 0.9375 and a 2° Standard Deviation of Phase Noise.

1/2 SCCC-SOQPSK-TG yields a coding gain of 7 dB, this is only 1.0 dB less than the performance gain realized by a similar system under coherent demodulation. However at a higher rate, like rate 7/8, no coding gain is realized which is mainly due to the increased phase noise present in the system. The performance of SCCC-PCM/FM is less affected by the increase in phase noise. This can be seen from Figure A.6, which shows a coding gain of 4.0 dB for a rate 1/2 system and a gain of 2.4 dB for a higher rate 7/8 system.

Also the performance of CC2 with CPM is shown in Figures A.7 and A.8. Similar to the performance shown by CC1 with CPM, a rate 1/2 CC2 with SOQPSK-TG yields a gain of 5.6 dB, this is 2.3 dB less than the gain produced by a similar coherent system. Also at a higher rate 7/8, as seen with CC1, SCCC-SOQPSK-TG yields no coding gain which again is mainly due to the increased phase noise and reduced forgetting factor. However, SCCC-PCM/FM systems are less affected due to this increased phase noise.

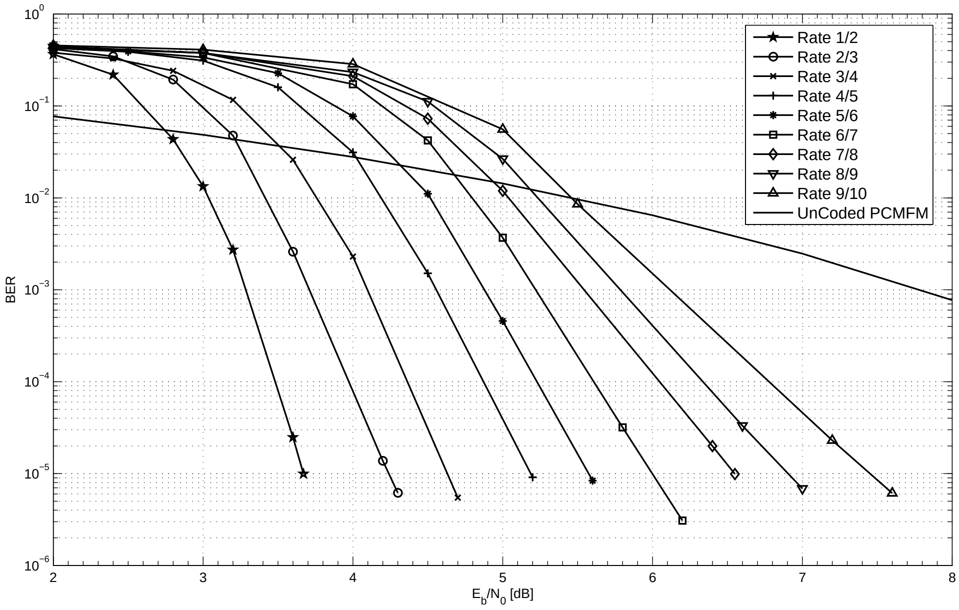

Figure A.3. BER Performance of Convolutional Code 2 with Non-Coherent SOQPSK-TG with a Forgetting Factor of 0.9375 and a 2° Standard Deviation of Phase Noise.

Table A.2. BER Performance of SCCC-CPM under Non-Coherent Demodulation with a Forgetting Factor of 0.875 and a 5° Standard Deviation of Phase Noise.

Modulation	Coding	Coding Gain	Reference
SOQPSK-TG	CC1	7.0 dB	Figure A.5
PCM/FM	CC1	4.0 dB	Figure A.6
SOQPSK-TG	CC2	5.6 dB	Figure A.7
PCM/FM	CC2	4.6 dB	Figure A.8

This is seen in Figure A.8, where a rate 1/2 system yields a gain of 4.6 dB, this is 1.7 dB less than the corresponding coherent system. Likewise, a rate 7/8 system yields a gain of 2.2 dB, this again is 2.4 dB less than its coherent counterpart.

A.2 Turbo-Product Codes with CPM

Figures A.9 and A.10 show the BER performance of TPCs with SOQPSK-TG and PCM/FM under non-coherent demodulation with a *forgetting factor* of 0.9375 and a 2°

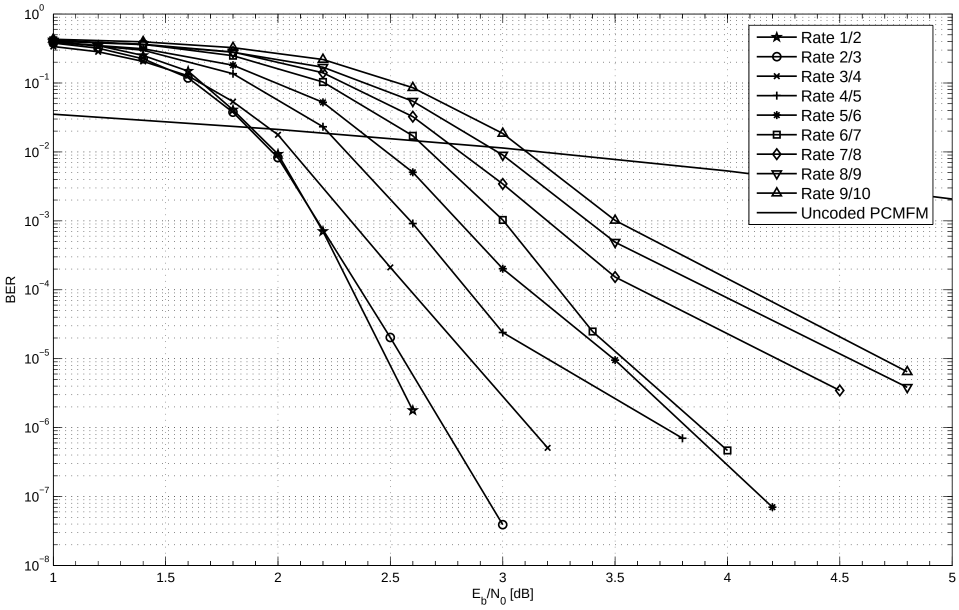

Figure A.4. BER Performance of Convolutional Code 2 with Non-Coherent PCM/FM with a Forgetting Factor of 0.9375 and a 2° Standard Deviation of Phase Noise.

standard deviation of phase noise. The coding gains realized with this system is tabulated in Table A.3. For instance, consider a rate 0.7932 TPC-SOQPSK-TG shown in Figure A.9, this system yields a coding gain of 3.2 dB, this is 1.3 dB less compared to the coding gain produced by the same system under coherent demodulation. Similarly, a rate 0.7932 TPC-PCM/FM system produces a gain of 3.2 dB, this is 0.8 dB less than the gain produced under coherent demodulation. As shown in the figures, similar performance difference between coherent and non-coherent demodulation is evident with other code rate TPC-CPMs. For reference, the coding gains realized here is tabulated in Table A.3

Now we consider the performance of TPCs with SOQPSK-TG and PCM/FM under non-coherent demodulation with a *forgetting factor* of 0.875 and a 5° standard deviation of phase noise. This is shown in Figures A.11 and A.12, respectively. Table A.4 tabulates the coding gains produced by this TPC-CPM. As shown in Figure A.12, a

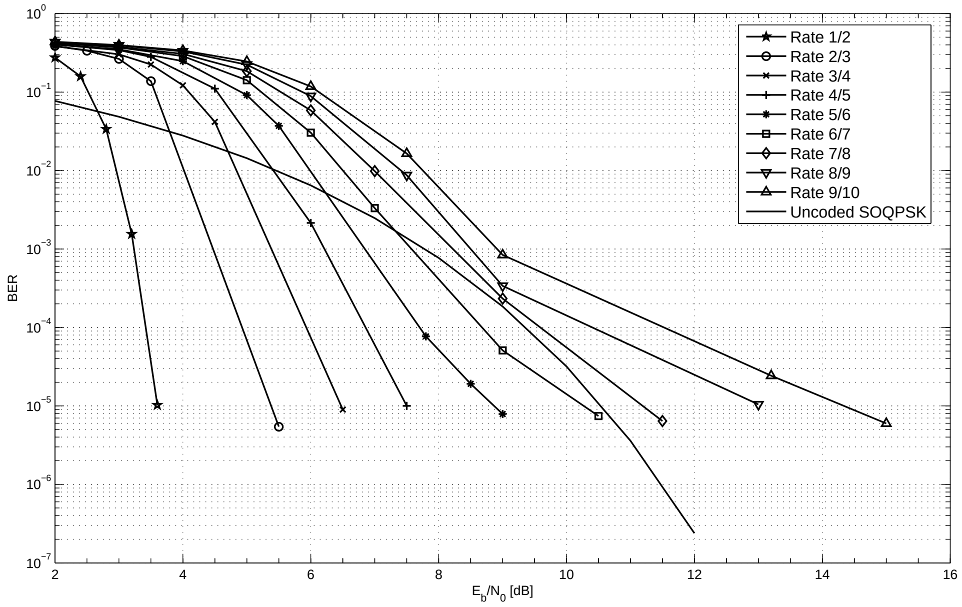

Figure A.5. BER Performance of Convolutional Code 1 with Non-Coherent SOQPSK-TG with a Forgetting Factor of 0.875 and a 5° Standard Deviation of Phase Noise.

Table A.3. BER Performance of TPC-CPM under Non-Coherent Demodulation with a Forgetting Factor of 0.9375 and a 2° Standard Deviation of Phase Noise.

Modulation	TPC	Coding Rate	Coding Gain
SOQPSK-TG	$(32, 26) \times (32, 26)$	0.6602	3.5 dB
SOQPSK-TG	$(64, 57) \times (64, 57)$	0.7932	3.2 dB
SOQPSK-TG	$(128, 120) \times (128, 120)$	0.8789	2.8 dB
PCM/FM	$(32, 26) \times (32, 26)$	0.6602	3.4 dB
PCM/FM	$(64, 57) \times (64, 57)$	0.7932	3.2 dB
PCM/FM	$(128, 120) \times (128, 120)$	0.8789	3.2 dB

rate 0.7932 TPC-PCM/FM yields a coding gain of 2.8 dB, this is 1.2 dB less than the gain produced a similar rate system under coherent demodulation. From Figure A.11, we can see the performance of TPC-SOQPSK-TG at a 5° standard deviation of phase noise. As seen from this figure, the performance of TPC-SOQPSK-TG is rather very poor. This shows the impact of increased phase noise on TPC-CPM systems. However, as seen in Figures 4.11 and A.9, TPC-SOQPSK-TG performs close to coherent

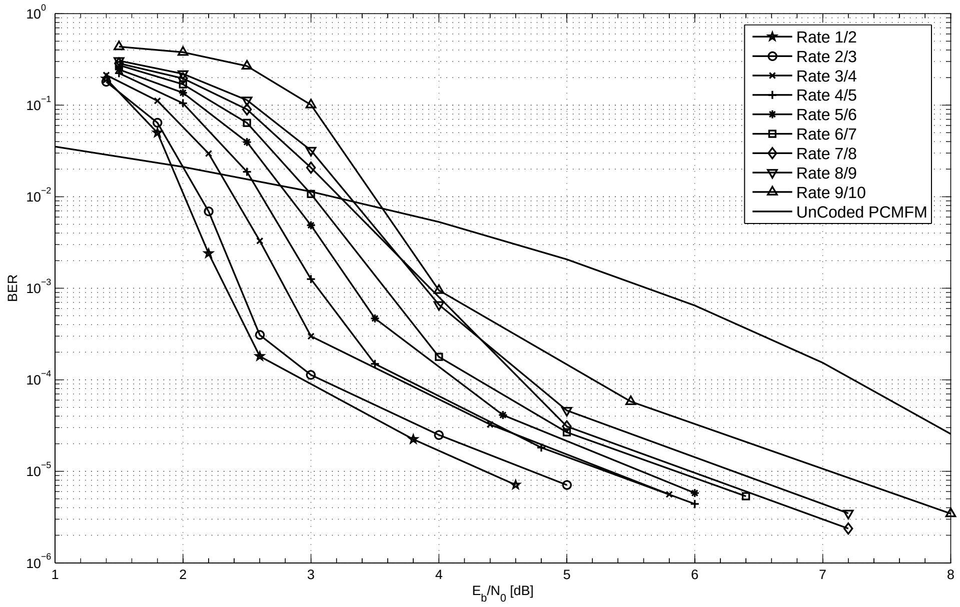

Figure A.6. BER Performance of Convolutional Code 1 with Non-Coherent PCM/FM with a Forgetting Factor of 0.875 and a 5° Standard Deviation of Phase Noise.

Table A.4. BER Performance of TPC-CPM under Non-Coherent Demodulation with a Forgetting Factor of 0.875 and a 5° Standard Deviation of Phase Noise.

Modulation	TPC	Coding Rate	Coding Gain
SOQPSK-TG	$(32, 26) \times (32, 26)$	0.6602	$-$
SOQPSK-TG	$(64, 57) \times (64, 57)$	0.7932	$-$
SOQPSK-TG	$(128, 120) \times (128, 120)$	0.8789	$-$
PCM/FM	$(32, 26) \times (32, 26)$	0.6602	2.9 dB
PCM/FM	$(64, 57) \times (64, 57)$	0.7932	2.8 dB
PCM/FM	$(128, 120) \times (128, 120)$	0.8789	2.7 dB

demodulation at a moderate 2° standard deviation of phase noise. From Figures 4.11 and A.11, we can also see that, given a 2° standard deviation of phase noise, the performance of TPC-CPM systems does not vary much due to varying forgetting factors (0.875 and 0.9375). This is evident from almost equal coding gains reported by these systems under consideration.

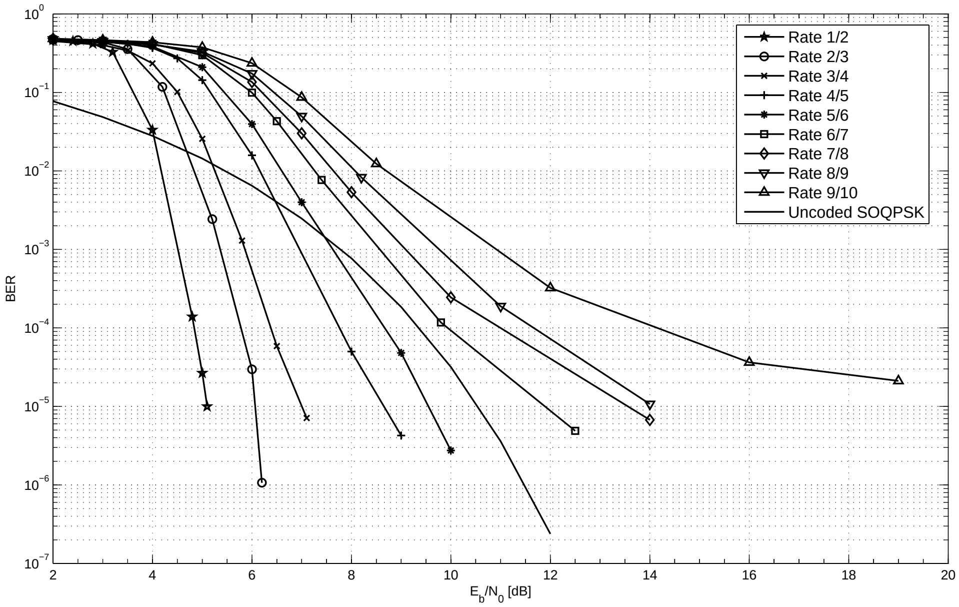

Figure A.7. BER Performance of Convolutional Code 2 with Non-Coherent SOQPSK-TG with a Forgetting Factor of 0.875 and a 5° Standard Deviation of Phase Noise.

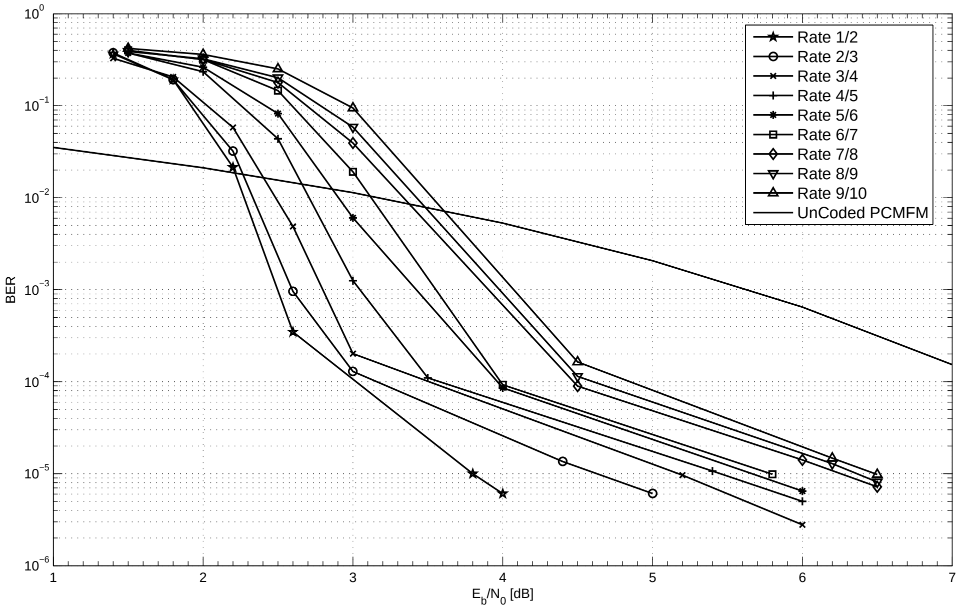

Figure A.8. BER Performance of Convolutional Code 2 with Non-Coherent PCM/FM with a Forgetting Factor of 0.875 and a 5° Standard Deviation of Phase Noise.

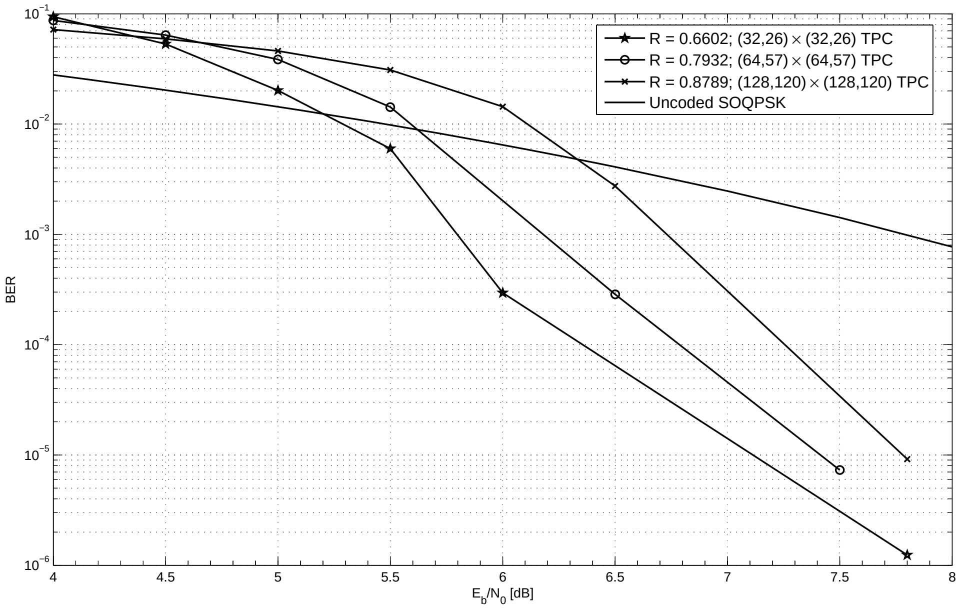

Figure A.9. BER Performance of Turbo-Product Code with Non-Coherent SOQPSK-TG with a Forgetting Factor of 0.9375 and a 2° Standard Deviation of Phase Noise.

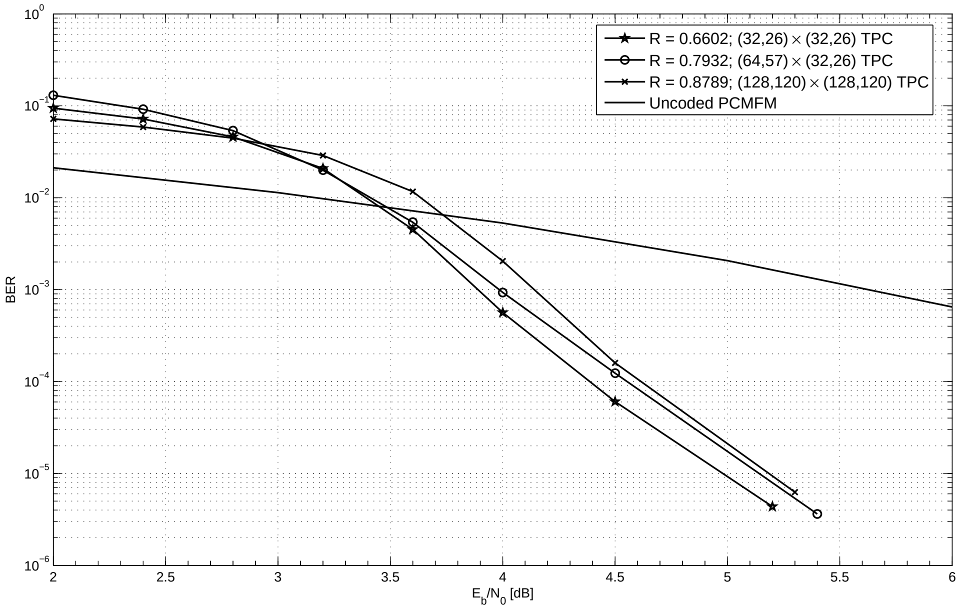

Figure A.10. BER Performance of Turbo-Product Code with Non-Coherent PCM/FM with a Forgetting Factor of 0.9375 and a 2° Standard Deviation of Phase Noise.

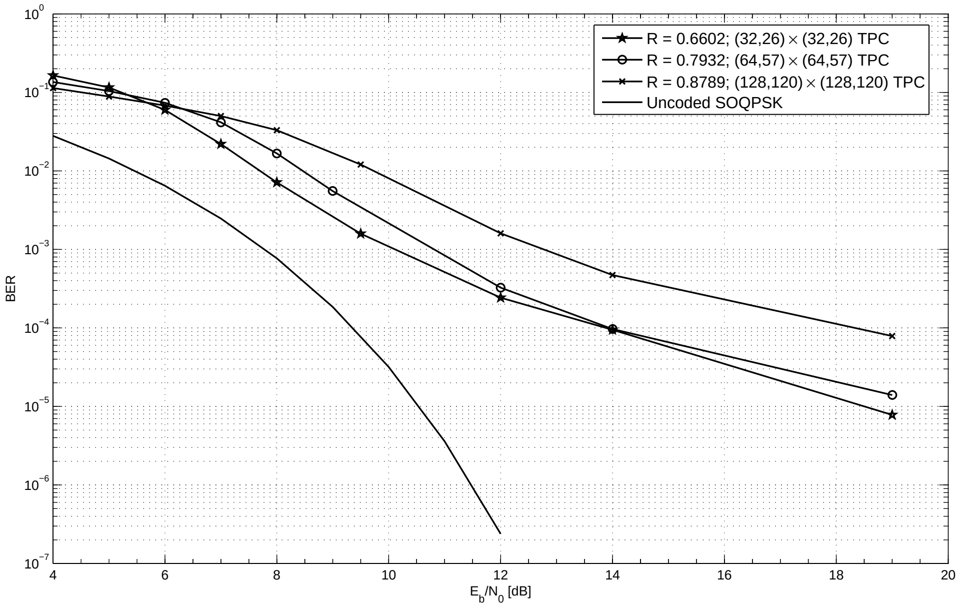

Figure A.11. BER Performance of Turbo-Product Code with Non-Coherent SOQPSK-TG with a Forgetting Factor of 0.875 and a 5° Standard Deviation of Phase Noise.

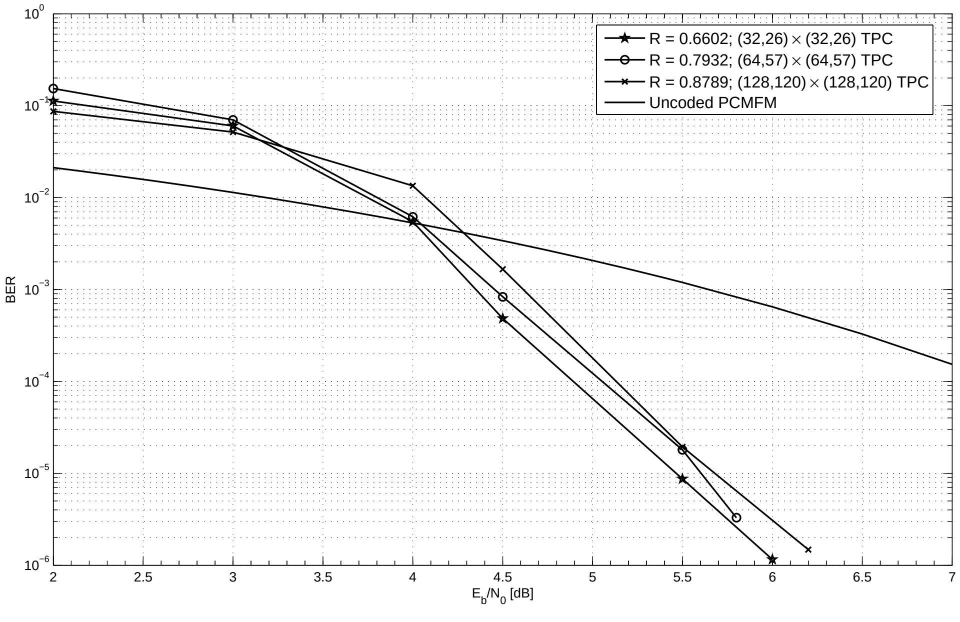

Figure A.12. BER Performance of Turbo-Product Code with Non-Coherent PCM/FM with a Forgetting Factor of 0.875 and a 5° Standard Deviation of Phase Noise.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, pp. 1064–1070, May. 1993.

[2] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 909–926, May 1998.

[3] T. Chalfant and C. Irving, "Range telemetry improvement and modernization," in *Proc. Int. Telemetering Conf.*, (Las Vegas, NV), pp. 294–303, Oct. 1997.

[4] Range Commanders Council Telemetry Group, Range Commanders Council, White Sands Missile Range, New Mexico, *IRIG Standard 106-04: Telemetry Standards*, 2004. (Available on-line at http://www.ntia.doc.gov/osmhome/106.pdf).

[5] W. Gao and K. Feher, "FQPSK: A bandwidth and RF power efficient technology for telemetry applications," in *Proc. Int. Telemetering Conf.*, (Las Vegas, NV), pp. 480–488, Oct. 1997.

[6] T. Hill, “A non-proprietary, constant envelope, variant of shaped offset QPSK (SOQPSK) for improved spectral containment and detection efficiency,” in *Proc. IEEE Military Commun. Conf.*, Oct. 2000.

[7] D. I. S. Agency, “Department of Defense interface standard, interoperability standard for single-access 5-kHz and 25-kHz UHF satellite communications channels.” Tech. Rep. MIL-STD-188-181B, Department of Defense, Mar. 1999.

[8] E. Law and K. Feher, “FQPSK versus PCM/FM for aeronautical telemetry applications; spectral occupancy and bit error probability comparisons,” in *Proc. Int. Telemetering Conf.*, (Las Vegas, NV), pp. 489–496, Oct. 1997.

[9] M. Geoghegan, “Exprimental results for PCM/FM, Tier 1 SOQPSK and Tier 2 multi-h CPM with turbo-product codes,” in *Proc. Int. Telemetering Conf.*, (Las Vegas, NV), Oct. 2003.

[10] P. Moqvist, *Serially Concatenated Systems: An Iterative Decoding Approach with Application to Continuous Phase Modulation*. PhD thesis, Lic. Engg. Thesis., Chalmers Univ. of Technology., Goteborg, Sweden, 1999. (Available: http://www.ce.chalmers.se.).

[11] P. Moqvist and T. Aulin, “Serially concatenated continuous phase modulation with iterative decoding,” *IEEE Trans. Commun.*, vol. 49, pp. 1901–1915, Nov. 2001.

[12] C. Brutel and J. Boutros, “Serial concatenation of interleaved convolutional codes and M-ary continuous phase modulations,” *Ann. Tele-Comm*, vol. 54, Mar/Apr. 1999.

[13] V. Szeto and S.Pasupathy, “Iterative decoding of serially concatenated convolutional codes and msk,” *IEEE Commun. Lett.*, vol. 3, pp. 272–274, Sept. 1999.

[14] K. Narayanan and G. Stüber, “Performance of trellis-coded CPM with iterative demodulation and decoding,” *IEEE Trans. Commun.*, vol. 49, pp. 676–687, Apr. 2001.

[15] C. E. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. Part 1, pp. 379–423, Jul. 1948.

[16] R. G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley, 1968.

[17] R. Gray, *Source Coding Theory*. Boston, Mass.: Kluwer Academic, 1990.

[18] P. Elias, “Coding for noisy channels,” *IRE Conv. Rec.*, pp. 4:37–47, 1955.

[19] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inform. Theory*, vol. 28, pp. 55–67, Jan 1982.

[20] S. Lin and D. Costello, *Error Control Coding*. New York: Prentice Hall, 2004.

[21] J. Proakis, *Digital Communications*. New York: McGraw-Hill, 2001.

[22] T. K. Moon, *Error Correction Coding*. New Jersey: Wiley-Interscience, 2005.

[23] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “A soft-input soft-output APP module for iterative decoding of concatenated codes,” *IEEE Commun. Lett.*, vol. 1, pp. 22–24, Jan. 1997.

[24] Y. Yasuda, K. Kashiki, and Y. Hirata, “High-rate punctured convolutional codes for soft decision Viterbi decoding,” *IEEE Trans. Commun.*, vol. 32, pp. 315–319, Mar. 1984.

[25] R. Pyndiah, “Near optimum decoding of product codes,” in *Proc. IEEE Global Telecommun. Conf.*, (San Francisco, California), pp. 339–343, Nov./Dec. 1994.

[26] P. Elias, “Error free coding,” *IEEE Trans. Inform. Theory*, pp. 29–37, Sept. 1954.

[27] R. Pyndiah, “Near-optimum decoding of product codes: Block turbo codes,” *IEEE Trans. Commun.*, vol. 46, pp. 1003–1010, Aug. 1998.

[28] H. Jin, *Analysis and Design of Turbo-like Codes*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, May 2001.

[29] S. Aji and R. J. McEliece, “The generalized distributed law,” *IEEE Trans. Inform. Theory*, vol. 32, pp. 325–343, Mar. 2000.

[30] D. Divsalar and F. Pollara, (1995, May), “Multiple turbo codes for deep-space communications,” *Telecommunications and Data Acquisition Progress Report*, vol. [Online]. Available: http://tmo.jpl.nasa.gov/tmo/progress_report/42-121/121T.pdf.

[31] E. Perrins and M. Rice, “Reduced-complexity approach to iterative detection of SOQPSK,” *IEEE Trans. Commun.*, vol. 55, pp. 1354–1362, Jul. 2007.

[32] D. Kumaraswamy, “Simplified detection techniques for serially concatenated coded continuous phase modulations,” Master’s thesis, Dept. Elect. Eng. Comp. Sci., Univ. Kansas, Lawrence, KS, July. 2007.

[33] D. Kim, T. Kwon, J. Choi, and J. Kong, “A modified two-step SOVA-based turbo decoder with a fixed scaling factor,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2000.