INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600
Iterative Decoding of Coded Continuous Phase Modulation

by

Victor F. Szeto

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science in the Department of Electrical and Computer Engineering University of Toronto

October 1998

Copyright © 1998, Victor F. Szeto
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

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

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni desextraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-40953-8
Iterative Decoding of Coded Continuous Phase Modulation

by

Victor F. Szeto

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

ABSTRACT

In this thesis, the concept of iterative decoding is applied to demodulating and decoding coded *continuous phase modulation* (CPM). The background theory of iterative decoding and CPM is reviewed. Various types of concatenation are studied and simulation results show the performance of different systems in Gaussian and Rayleigh fading channels. It is found that coded CPM system with iterative decoding has a good bit error rate performance; however, the choice of constituent encoders is an important parameter in determining the overall performance.
ACKNOWLEDGMENTS

First I would like to express my deepest appreciation to Professor Pasupathy for his patience and guidance throughout the course of this work, and for his encouragement in the times when I was so frustrated. It was a great pleasure to have worked with him.

My thanks are also due to my officemates for their valuable (and humorous) discussions and insightful comments. I really treasure the days and nights that we spent in the office over the last two years.

I wish to thank my friends for their moral support that made this period an enjoyable and rewarding one.

Lastly but most importantly, I would like to express my gratitude to my parents, to whom I am most indebted.
Contents

List of Tables .................................................................................. iv

List of Figures .................................................................................. vi

1 Introduction .................................................................................. 1
   1.1 Continuous Phase Modulation ................................................... 1
   1.2 Iterative Decoding of Concatenated Codes .................................. 3
   1.3 Coded Continuous Phase Modulation ......................................... 4
   1.4 Goal of Thesis ........................................................................... 4
   1.5 Outline of Thesis ....................................................................... 4

2 Theory of Iterative Decoding and CPM ......................................... 6
   2.1 Notation and Definitions ........................................................... 6
   2.2 Iterative decoding of Concatenated Code .................................... 9
   2.3 Continuous Phase Modulation ................................................... 15
       2.3.1 Signal Description and State Representation ....................... 16
       2.3.2 Description of CPM as Memoryless Modulation Preceded by
             Coding ................................................................................... 20
       2.3.3 Minimum Shift Keying: An Example .................................... 23
   2.4 Iterative Decoding of Concatenated Convolutional Code and CPM .. 24
3 Performance Simulation of Coded CPM

3.1 System Model ........................................ 28
  3.1.1 Channel Model .................................. 29
  3.1.2 Coded CPM Transmitter ....................... 30
  3.1.3 Modulator ...................................... 34
  3.1.4 Interleaver .................................... 34
  3.1.5 Demodulator .................................. 35
  3.1.6 Iterative Decoder ............................. 39

3.2 Performance of Different Coded CPM Systems .......... 40
  3.2.1 Type 1 Coded CPM ............................. 40
  3.2.2 Type 2 Coded CPM ............................. 46
  3.2.3 Type 3 Coded CPM ............................. 55
  3.2.4 The Three Coded CPM Schemes Compared ......... 58

4 Conclusion .............................................. 63
  4.1 Summary of Results .............................. 63
  4.2 Suggestion for Future Research ................... 64

A Fading Channel Simulation .................................. 65
  A.1 Jakes' Simulator .................................. 65

B List of Interleavers ................................... 68
List of Tables

2.1 Definition of some common frequency pulses $g(t)$ ........................................ 18
2.2 Mapping of $(c_1, c_2)$ to channel signal $x(t)$ .................................................. 24
3.1 Rate $1/n$ FF convolutional encoders used in this thesis. ................................. 33
3.2 Rate $2/3$ convolutional encoders used in this thesis. ........................................ 33
3.3 Simulation parameters .................................................................................................. 41
# List of Figures

2.1 A trellis edge at the $k^{th}$ symbol interval. ........................................ 7  
2.2 A (6,3) FF convolutional encoder. ......................................................... 8  
2.3 Example of a rate $R = 2/3$ convolutional encoder. ................................. 8  
2.4 Parallel and serial concatenation of codes. ............................................. 9  
2.5 The APP decoder. ....................................................................................... 10  
2.6 Block diagram of a serial concatenated encoder. ........................................ 14  
2.7 Block diagram of decoder for serial concatenated system. .......................... 15  
2.8 Generation of CPM signal. ......................................................................... 17  
2.9 Phase response pulse $q(t)$ and frequency pulse $g(t)$ for GMSK. ............. 18  
2.10 Decomposition of CPM modulator ............................................................ 20  
2.11 State trellis for the binary 2REC scheme with $h = 1/2$. .......................... 21  
2.12 Coherent CPM receiver of Osborne and Luntz ......................................... 22  
2.13 Coded modulation representation of serial MSK. ...................................... 23  
2.14 A simple coded modulation system. ......................................................... 24  
2.15 A receiver for the simple coded modulation system................................. 24  
2.16 Serial concatenated code with modulation. .............................................. 25  
2.17 Decoder for serial concatenated coded modulation, ................................. 25  
2.18 Viewing coded CPM system as serial concatenation of trellis codes. ....... 26  
2.19 Coded CPM with combined code and CPM trellis. .................................... 27  
3.1 The coded CPM system block diagram. ...................................................... 29
3.2 Transmitter of a Type 1 coded CPM system. ............................................. 31
3.3 Transmitter of a Type 2 coded CPM system. ........................................... 32
3.4 Transmitter of a Type 3 coded CPM system. The memory of the modulation is not used in iterative decoding. ................................................. 32
3.5 The correlator bank demodulator for CPM. .............................................. 37
3.6 Equivalent baseband correlator receiver. ................................................. 39
3.7 Performance bound on coherent demodulation of MSK and GMSK with $BT = 0.3$. ................................................................. 41
3.8 Performance of the Osborne and Luntz type receiver in AWGN with an observation duration of 5 symbol intervals for MSK and GMSK ($BT = 0.3$) 42
3.9 Performance of Type 2 coded MSK with (15,17) outer code and a 4-state rate $R = 2/3$ recursive inner code, $N = 500$ with 1,2,3,5 and 9 iterations. 43
3.10 Type 1 coded MSK and GMSK ($BT = 0.3$) with (25,33,37) outer code in AWGN. $N=500$ 4 iterations ......................................................... 44
3.11 Type 1 coded MSK and GMSK ($BT = 0.5$) with (25,33,37) outer code in a fading channel. $N=500$, 3 iterations, $f_m T = 0.02$ ............................... 45
3.12 Type 1 coded MSK and GMSK ($BT = 0.3$) systems with 64 states in a AWGN channel. $N=500$, 3 iteration ................................................. 45
3.13 Type 1 coded GMSK ($BT = 0.3$) with outer codes (5,7,7), (25,33,37) and (133,145,175) in a AWGN channel. $N=500$, 4 iterations. .................... 46
3.14 Type 1 coded GMSK ($BT = 0.3$) with outer code (25,33,37) in a AWGN channel. $N=200$, $N=500$ and $N=1000$, 3 iterations. ......................... 47
3.15 Type 1 coded MSK with outer code (25,33,37) in a AWGN channel. $N=200$, $N=500$ and $N=1000$, 3 iterations. ................................. 47
3.16 A general configuration of a Type 2 coded MSK. ................................... 48
3.17 A rate $R = k/l$ signal space code for MSK. ......................................... 48
3.18 A rate $R = 1$ signal space code for MSK. ......................................... 48
3.19 Type 2 coded MSK without an inner code. ................................. 49

vii
3.20 Type 2 coded MSK with and without an inner code in AWGN. The result of the recursive code shown here is obtained after 4 iterations. The outer code in both cases is the rate $R = 1/3(25,33,37)$ code. 

3.21 A rate $R = 1$ signal space code that violates the continuous phase constraint. 

3.22 The effect of the continuous phase constraint on the performance of a coded modulation system. 

3.23 4-state recursive and non-recursive inner code in Type 2 coded MSK. 

3.24 Type 2 coded MSK with (15,17) outer code, both 4-state recursive and non-recursive rate $R = 2/3$ inner code in a AWGN channel. $N=500$, 5 iterations. 

3.25 Type 2 coded MSK with (15,17) outer code, both recursive and non-recursive 4-state rate $R = 2/3$ inner code in a $+\text{AWGN}$ channel. $N=500$ and 1000. 5 iterations. 

3.26 Type 2 coded MSK with (15,17) outer code, both recursive and non-recursive 4-state rate $R = 2/3$ inner code in fading channel. $N=500$, 5 iterations. 

3.27 Type 2 coded GMSK ($BT = 0.5$) with (7,5) outer code, both recursive and non-recursive inner code in the AWGN channel. The inner codes are listed in Table 3.2. $N=500$, 5 iterations. 

3.28 Reversing the order of codes in Type 2 coded MSK. 

3.29 Reversing the order of codes in Type 2 coded GMSK. 

3.30 Performance of Type 2 coded MSK in the AWGN channel. 

3.31 Performance of 16-state, rate $R = 2/3$ recursive and non-recursive code in a AWGN channel with BPSK. 

3.32 Type 3 coded MSK and GMSK with (7,5) outer code, 16-state rate $R = 2/3$ recursive and non-recursive inner codes in a AWGN channel. The inner codes are listed in Table 3.2 5 iterations $N=500$. 

3.33 Performances of coded MSK in AWGN.
3.34 Performance of coded GMSK in fading channel. .......................... 61
3.35 Performance of coded MSK in fading channel. ............................ 62
A.1 Rayleigh fading simulator. \( n_1(t) \) and \( n_Q(t) \) are independent white Gaussian noise. .......................................................... 66
A.2 Jakes' simulator used in this thesis. ........................................... 66
Chapter 1

Introduction

In radio frequency (RF) communication, bandwidth is limited and precious and therefore spectral efficiency becomes a very important issue. Advances in RF digital communication technology (such as satellite communications and cellular radios) have put a strong pressure on the demand for RF channels. The never-ending rise in traffic through our fixed RF spectrum motivates the search for bandwidth efficient and reliable communications systems. The challenge is to transmit information through RF channels reliably while keeping the spectrum of the modulated signal as narrow as possible. The solutions to these problems are channel coding and bandwidth efficient modulation.

1.1 Continuous Phase Modulation

Continuous Phase Modulation (CPM) is the class of digital phase modulation techniques in which the phase of the signal is constrained to be continuous. It has become a popular transmission technique, primarily owing to the fact that the CPM signal possesses a constant envelope and good spectral properties intrinsic to the modulation
process. Unlike quadrature phase shift keying (QPSK) or offset QPSK (O-QPSK), where the baseband signal linearly modulates the carrier, CPM is a family of nonlinear modulations. Power efficient communication systems often employ nonlinear amplifiers for signal amplification. However, linearly modulated signals cannot be amplified by nonlinear amplifiers because the nonlinearity of the amplifier will introduce undesired signal amplitude variations. When a band limited carrier is non-linearly amplified, the filtered side lobes re-appear, which causes adjacent channel interference; and therefore, linear modulation requires the less power-efficient linear amplifiers. The constant envelope property of CPM schemes avoids the linearity requirements which reduces the cost of amplification. This is essential for a power efficient communication system. Some examples of CPM schemes are Minimum Shift Keying (MSK) [1], Gaussian Minimum Shift Keying (GMSK) [2], Tamed Frequency Modulation (TFM) [3] and Generalized Tamed Frequency Modulation (GTFM) [4], to name just a few.

CPM has been studied extensively during the past two decades. It is not only a topic of theoretical research, but is also widely applied in many communications systems. The most well known application of CPM is the GMSK scheme in Groupe Spécial Mobile (GSM), the Pan-European cellular radio standard. Another cellular radio standard, the DCS1800, which is essentially a derivative of GSM, also employs GMSK. CPM is also very popular in radio modem applications. Recent standards such as Cellular Digital Packet Data (CDPD) (which is a standard for Wide Area Networks) and Mobitex uses GMSK for their modulation method.

CPM is a modulation scheme with memory. In fact, a CPM scheme can be viewed as a trellis code combined with a memoryless modulator with a rather large modulator alphabet. This viewpoint was first advanced by Rimoldi [5]. It leads us to think that the optimum decoding algorithm for trellis codes can also be applied to demodulate CPM. This "trellis" view of CPM opens the way to joint decoding/demodulation of

\footnote{Mobitex is a trademark owned by the Telia Corporation.}
coded CPM with iterative algorithms, which is the theme of this thesis.

1.2 Iterative Decoding of Concatenated Codes

In 1948, Claude Shannon in Bell Laboratories published the remarkable paper entitled “A Mathematical Theory of Communication” in the Bell System Technical Journal [6], showing that there exists an error control code which allows transmission of information at any rate less than the channel capacity with arbitrary low bit error rate. This result profoundly changed people’s view on the limits of communications and stimulated coding theorists to search for good error-control codes that perform close to this theoretical limit with tolerable complexity.

In 1993, Berrou, Glavieux and Thitimajshima [7] introduced turbo codes, a new coding scheme with near Shannon limit error correction capabilities. The key elements in turbo codes are parallel concatenated recursive systematic convolutional codes separated by an interleaver and iterative decoding. The remarkable performance of turbo codes (with iterative decoding) motivates researchers to investigate other concatenated codes with iterative decoding algorithms, such as serially concatenated codes [8], double serially concatenated codes [9] and hybrid concatenated codes [10].

The basic idea of iterative decoding is to break up decoding of a fairly complex and long code into steps; each step decodes the data independently and produces “soft-information” (i.e., the probability information or the reliability of the data). The “soft-information” is exchanged between decoding steps that ensures almost no information loss. Ideally, the “soft-information” is computed by the a posteriori probability (APP) algorithm. However, the complexity of APP algorithm for decoding concatenated code makes it impossible for practical implementation. Instead of using the APP algorithm, some algorithms that approximate the APP algorithm are used in each decoding step.
1.3 Coded Continuous Phase Modulation

Viewing CPM as a trellis coded modulation scheme, we may think that the iterative decoding algorithm for convolutional codes can also be applied to demodulating CPM. In fact, Gertsman and Lodge [11] have proposed an iterative decoding algorithm for demodulating CPM signals in Rayleigh fading channels. If a convolutional code is concatenated with CPM, the resulting system can be thought of as a concatenated code system and the iterative decoding strategy for concatenated codes can be used to jointly demodulate/decode data. This means that the information provided by the demodulator is used by the decoder and the demodulator utilizes the "soft-information" computed by the decoder as well, so that the resulting system has a good bit error rate performance.

1.4 Goal of Thesis

The goal of this thesis is to find the parameters (such as constituent codes and trellis combination) for a good coded CPM system with iterative decoding. The performance of the system will be evaluated by Monte Carlo simulation.

1.5 Outline of Thesis

The development of the thesis is covered in Chapters 2-3 and conclusions are given in Chapter 4. Some mathematical detail and channel modeling methodology is covered in the appendices. Chapter 2 and Chapter 3 are organized as follows.

In Chapter 2, most of the background concept of this thesis will be discussed, including the principle of iterative decoding, decomposition of CPM and the motivation of iterative decoding/demodulating coded CPM. Various ways of concatenating CPM with convolutional codes will be introduced. The focus of this chapter is to
introduce the tools necessary to model and simulate the coded CPM system.

In Chapter 3, coded CPM is applied to additive white Gaussian noise (AWGN) channels and slow frequency non-selective Rayleigh fading channels. The simulation methodology will be described in the chapter. Simulation results for different coded CPM systems will be presented.

Conclusions are given in chapter 4. The derivation of the a posteriori probability (APP) algorithm and the fading channel simulator used in this thesis are given in the appendices.
Chapter 2

Theory of Iterative Decoding and CPM

In this chapter, the principle of iterative decoding of concatenated codes will be presented. The decomposition of CPM into a trellis code and memoryless modulation scheme will be discussed. Finally, the idea of iterative decoding/demodulation of concatenated CPM and a convolutional codes will be introduced.

2.1 Notation and Definitions

The following notation will be used in the thesis. Consider a trellis code with a trellis edge at the $k^{th}$ symbol interval as shown in Fig. 2.1. Let $\mathcal{U}$ be the set of valid input symbols and $C$ be the set of valid subsequence of this trellis code. Let $u_k \in \mathcal{U}$ be the input symbol at the $k^{th}$ symbol interval and let $c_k \in C$ be the output subsequence at the $k^{th}$ symbol interval. The state of the encoder at time $k$ is denoted by $s_k$. The trellis edge is defined as a 4-tuple $(s_{k-1}, u_k, c_k, s_k)$, and let $T_k$ be the set of edges in
Figure 2.1: A trellis edge at the \(k^{th}\) symbol interval.

The \(k^{th}\) trellis section. Then

\[ u_k(m', m) = \{ u : (m', u, c_k, m) \in T_k \} \]

and

\[ c_k(m', m) = \{ c : (m', u_k, c, m) \in T_k \} \].

Given the state at time \((k - 1)\) and the input \(u_k\), the output \(c_k\) and the state at time \(k\) is uniquely determined. When \(c_k \in C\) is transmitted through the channel, the received noisy symbol is denoted by \(Y_k\). The notation \(Y_1^N\) represents the sequence \(\{Y_1, Y_2, \ldots, Y_N\}\).

A binary feedforward (FF) convolutional code is a convolutional code without feedback. In this thesis, a FF convolutional encoder will also be referred to as a non-recursive convolutional encoder. A rate \(R = 1/n\) FF convolutional code is denoted by the \(n\)-tuple \((g_1, g_2, \ldots, g_n)\) where \(g_i\) is the encoder generator polynomial of the \(i^{th}\) output bit expressed in octal form. For example, for the rate \(R = 1/2\) feedforward encoder shown in Fig. 2.2, the \(D\)-transform [12] of the impulse responses are

\[
X^{(0)}(D) = 1 + D
\]

\[
X^{(1)}(D) = D + D^2.
\]
The encoder will be denoted as a (6,3) feedforward encoder.

A rate $R = k/n$ convolutional code is represented by a $k \times n$ matrix, called the encoder transfer function matrix. The $ij^{th}$ element (the $i^{th}$ row, $j^{th}$ column element) is the $D$-transform of $g^{(j)}_i$, which is the impulse response of the $j^{th}$ output bit with respect to the $i^{th}$ input bit. For example, the encoder shown in Fig. 2.3 is described by the encoder transfer function matrix

\[
\begin{bmatrix}
1 + D & D & 1 + D \\
D & 1 & 1
\end{bmatrix}.
\]

In the case of feedback convolutional encoder
2.2 Iterative decoding of Concatenated Code

A popular method to achieve a long powerful code at a moderate decoding complexity is the concatenation of codes. A simple concatenation scheme consists of a combination of two simple constituent encoders and an interleaver. Concatenation can be done in serial or parallel (see Fig. 2.4) such that the constituent codes can be decoded independently. It had been shown that iterative decoding of concatenated code yields remarkable coding gains close to the theoretical limit [7]. The essence of iterative decoding of concatenated codes (serial or parallel) are:

- soft decision decoding,
- exchange of information between constituent decoders,
- improving the reliability of data by repeated decoding using results of previous decoding steps, and
- the presence of an interleaver between constituent codes.

Decoding concatenated codes consists of two steps: decoding the constituent codes independently and passing the decision along with the probability information about the data to other constituent decoders in the system. In “conventional” decoding (as opposed to iterative decoding or decoding with feedback) of concatenated codes, only hard-decision is passed from one decoder to another decoder. However, it is well known that soft-decision decoding is better than hard-decision decoding; it is
Figure 2.3: The APP decoder.

because, unlike hard-decision decoding where the decoder only outputs the decision, soft-decision decoding involves passing the confidence level along with the decision to other decoders. The information loss between decoders is thus minimized, which is an essential requirement in iterative decoding.

In order to perform soft decision decoding and allow information exchange between decoders, a special decoding module is needed. The decoding module should take the confidence level of data and codeword symbols as input and produce a "smoothed" version of them using information from the code structure and input \textit{a priori} probabilities. We will call such a decoder an "APP decoder". The APP decoder implements the APP decoding algorithm which is optimal for decoding individual constituent code. Different implementation of the APP algorithm had been proposed in the literature, such as \cite{13} the \textit{soft-input-soft-output} (SISO) decoder in \cite{13} and the MAP "filter" in \cite{14}, for any codes admitting a trellis representation. A "black box" representation of the APP decoder is shown in Fig. 2.5.

In Fig. 2.5, \( \Pr\{u_k; I\} \) and \( \Pr\{c_k; I\} \) are the probability mass functions of the input symbols \( u_k \) and codeword symbols \( c_k \) respectively at the input of the decoder; and \( \Pr\{u_k; O\} \) and \( \Pr\{c_k; O\} \) are the probability mass functions at the output of the decoder.

The function of the APP decoder is to estimate the probability distribution of input and output symbols. The underlying decoding algorithm of the APP decoder for
trellis codes is called the APP algorithm (it is also known as the two-way algorithm, forward-backward algorithm; and, in coding literature, it is frequently called the BCJR algorithm, after [15]). The APP algorithm used in this thesis will be briefly described below.

Suppose the decoder receives a block of $N$ noisy codewords ($Y_i^N$) from the channel. After observing the entire sequence $Y_i^N$, the conditional probability distributions of the $k^{th}$ data ($u_k$) and codeword ($c_k$) symbols are computed as

$$\Pr\{u_k = d|Y_i^N\} = K_u \sum_{c \in (m', d, c_k, m) \in \mathcal{T}_k} \alpha_{k-1}(m') \beta_k(m) \Pr\{Y_k|c_k = c(m', m)\} \quad d \in \mathcal{U}$$

$$\Pr\{c_k = x|Y_i^N\} = K_c \sum_{c \in (m', u_k, x, m) \in \mathcal{T}_k} \alpha_{k-1}(m') \Pr\{u_k = u(m', m)\} \beta_k(m) \quad x \in \mathcal{C}$$

where $K_u$ and $K_c$ are normalization constants such that

$$K_u \sum_d \Pr\{u_k = d|Y_i^N\} = 1$$

$$K_c \sum_x \Pr\{c_k = x|Y_i^N\} = 1.$$

The quantities $\alpha_k(m)$ and $\beta_k(m)$ are defined as

$$\alpha_k(m) = \Pr\{s_k, Y_i^k\}$$

$$\beta_k(m) = \Pr\{Y_i^N|s_k = m\}.$$

They are computed by forward and backward recursions:

$$\alpha_k(m) = \sum_{m'} \alpha_{k-1}(m') \Pr\{u_k = u(m', m)\} \Pr\{Y_k|c_k = c(m', m)\}$$

for $k = 1, \ldots, N$

$$\beta_k(m) = \sum_{m'} \beta_{k+1}(m') \Pr\{u_{k+1} = u(m, m')\} \Pr\{Y_{k+1}|c_{k+1} = c(m, m')\}$$
for \( k = (N - 1), \ldots, 0 \).

The initialization of \( \alpha_0(m) \) and \( \beta_N(m) \) can be done in various ways, depending on the encoding process. If the starting state and the ending state of the encoder are known, the initial conditions are

\[
\alpha_0(m) = \begin{cases} 
1 & \text{if } m = \text{initial state;} \\
0 & \text{otherwise.}
\end{cases} \tag{2.1}
\]

\[
\beta_N(m) = \begin{cases} 
1 & \text{if } m = \text{ending state;} \\
0 & \text{otherwise.}
\end{cases} \tag{2.2}
\]

This implies that the encoder trellis is properly terminated at a known state. In the case where the initial state of the encoder is known but the final state is not known (which is the case when the trellis is truncated), \( \alpha_0(m) \) can be initialized using Eq. (2.1) but \( \beta_N(m) \) should be initialized as

\[
\beta_N(m) = \frac{1}{N_{state}} \quad \forall m = 1, \ldots, N_{state} \tag{2.3}
\]

where \( N_{state} \) is the number of states in the trellis. If data is being encoded continuously so that both the starting state and ending state of the encoder are not known to the decoder, the initialization of \( \alpha_0(m) \) and \( \beta_N(m) \) should be done by utilizing information from previously decoded data. That is,

\[
\alpha_i^0(m) = \begin{cases} 
1 & \text{if } m = \text{initial state;} \\
0 & \text{otherwise;}
\end{cases} \tag{2.4}
\]

\[
\alpha_i^1(m) = \alpha_{N-1}^i(m) \quad \forall m = 1, \ldots, N_{state} \quad \text{and } i > 1 \tag{2.5}
\]

\[
\beta_N^i(m) = \frac{1}{N_{state}} \quad \forall m = 1, \ldots, N_{state} \tag{2.6}
\]

12
where $\alpha^k_i(m)$ is the value of $\alpha_k(m)$ computed in the $i^{th}$ data block.

After computing the soft information ($\Pr\{u_k|Y^N_1\}$ and $\Pr\{c_k|Y^N_1\}$), each constituent decoder should pass the soft information to other decoders. In “conventional” decoding of concatenated codes, although the later decoding stages can benefit from using information from earlier stages, the reverse is not true. For example, when decoding a serial concatenated code in the “conventional” way, first the decoding is done by the inner decoder, the decoded symbols are then decoded again by the decoder of the outer code. In this case, only the outer code can use the information gleaned from the inner code.

In iterative decoding, the constituent decoders should be configured in such a way that the APP information output by one constituent decoder can be used as a priori information for other decoders, so that every constituent decoder in the system can refine its decision based on information from other constituent decoders. For a serial concatenated system shown in Fig. 2.6, the corresponding decoder configuration is shown in Fig. 2.7. In the figure, the block labeled “π” is the interleaver; “DEC 1” and “DEC 2” are the constituent APP decoders corresponding to the constituent codes “CODE 1” and “CODE 2” respectively. From Fig. 2.7 it can be seen that the iterative decoding algorithm consists of a number of decoding stages (iterations). At each stage the APP information is being refined; the refined APP information propagates to the next decoding stage where it will be used as a priori information. Therefore, iterative decoding can be thought of as a process of adding and refining information at each stage to produce the final result.

Another crucial component in iterative decoding is the interleaver. The interleaver is present for two basic reasons. First, the interleaver improves the performance of the decoder in a bursty channel, since error bursts introduced by the channel are broken up and spread across several codewords by the interleaver. Secondly, although the APP algorithm is optimal for decoding individual constituent code, the global algorithm for iterative decoding of concatenated code is sub-optimal. It is only
an approximation to the APP algorithm. This approximation is good only if the inputs to the decoder are independent. As a result, interleaving must be present to randomize the inputs to the decoder.

Let us explain how the iterative decoding process works according to Fig. 2.7. In the first iteration, "DEC 2" is fed with the demodulator soft output, consisting of the probability distributions of the symbols that are output symbols of "CODE 2" received from the channel. The probability information is then processed by "DEC 2" that refines the information conditioned on the code structure. This refined information (often referred to as the extrinsic information in coding literature) is deinterleaved (by passing through the block labeled "\(\pi^{-1}\)". Since the input symbol of "CODE 2" after deinterleaving corresponds to the output symbols of "CODE 1", the deinterleaved information becomes the probability distribution of the output symbols of "CODE 1". It is then sent to the "DEC 1" block that processes the output symbol probability distribution and computes the probability distributions for both input and output symbols based on the code constraints. Since "DEC 1" does not have any a priori information about the input symbols, it assumes that the input symbols are uniformly distributed. The probability distribution of the input symbol computed by "DEC 1" will be used in the last iteration to recover the input symbols; whereas the probability distribution of the output symbols will be fed back to "DEC 2" to start the next iteration.
Figure 2.7: Block diagram of decoder for serial concatenated system.

2.3 Continuous Phase Modulation

CPM is a suitable candidate for transmission of digital data in RF channels with narrow bandwidth due to its excellent spectral property. Unlike the case in QPSK or OQPSK where there are sudden phase changes in successive symbol intervals, there are no abrupt changes (and hence high frequency components) in CPM signals. This gives CPM signals good spectral properties and makes it attractive for wireless applications. There are several variations of CPM schemes: multi-amplitude CPM, where both the amplitude and the phase (or frequency) of the signal change with time; multi-$h$ CPM, where the modulation index is cyclically changed for successive symbol intervals; and constant envelope CPM where the transmitted signal has a constant envelope. In this thesis, we will focus on the more popular CPM scheme: the constant envelope CPM with a fixed modulation index $h = \frac{1}{2}$. The modulation index is defined in the next section.
2.3.1 Signal Description and State Representation

A CPM signal is a phase modulated sinusoid defined by:

\[ s(t) = \sqrt{\frac{2E}{T}} \cos(\omega_c t + \phi(t, u)). \]  

(2.7)

Here \( E \) is the energy of the signal expanded per symbol interval \( T \) and \( \omega_c \) is the carrier frequency. An \( M \)-ary data sequence \( u = (u_0, \ldots, u_j, \ldots) \) with components \( u_j \) selected from the set

\[ \{ \pm 1, \pm 3, \ldots, \pm (M - 1) \} \quad \text{where} \quad M = 2^k, \]

drives the carrier phase \( \phi(t, u) \). The phase at time \( t \) is obtained as

\[ \phi(t, u) = 2\pi h \sum_{i=0}^{\infty} u_i q(t - iT), \]

where \( h \) is the modulation index and \( q(t) = \int_{-\infty}^{t} g(\tau) \, d\tau \) is the phase shaping function, which is the heart of a CPM signal. The modulation index \( h \) is defined as

\[ h = 2f_d T \]

where \( f_d \) is the peak frequency deviation. The function \( g(t) \) is a smooth pulse shape (called the frequency pulse) over a finite time interval \( 0 \leq t \leq LT \) and zero outside. The pulse shape \( g(t) \) is normalized such that

\[ \int_0^{LT} g(t) \, dt = \frac{1}{2}. \]
The shape of \( g(t) \) determines the smoothness of the transmitted information bearing phase. Memory is introduced into the CPM signal by means of the continuous phase and \( L \) is the number of symbol intervals over which an input symbol exerts a changing influence on the cumulative phase. CPM schemes with \( L = 1 \) are referred to as full response schemes and those with \( L > 1 \) are called partial response schemes. A general CPM signal generator is shown in Fig. 2.8.

A CPM signal can be described by its phase response pulse \( q(t) \) or by its frequency pulse \( g(t) \). Some important frequency shaping pulses are listed in Table 2.1. The meaning of the abbreviations in the table are

- **LRC** – raised cosine, pulse length \( L \),
- **LREC** – rectangular frequency pulse, length \( L \), and
- **GMSK** – Gaussian minimum shift keying.

Note that 1REC CPM is more commonly known as MSK. The two names will be used synonymously in this thesis. The phase response pulse \( q(t) \) and the frequency pulse \( g(t) \) for GMSK is illustrated in Fig. 2.9.

The information carrying phase \( \phi(t, u) \) during the interval \( nT \leq t < (n + 1)T \) can

\[ \text{If } g(t) \text{ is viewed as the impulse response of a Gaussian filter, } B \text{ is the } -3\text{dB bandwidth of the filter; and } BT \text{ is the bandwidth-bit-duration product.} \]
Table 2.1: Definition of some common frequency pulses $g(t)$

<table>
<thead>
<tr>
<th>$LREC$</th>
<th>$g(t) = \begin{cases} \frac{1}{2LT} &amp; 0 \leq t \leq LT \ 0 &amp; \text{otherwise} \end{cases}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LRC$</td>
<td>$g(t) = \begin{cases} \frac{1}{2LT}(1 - \cos \frac{2\pi t}{LT}) &amp; 0 \leq t \leq LT \ 0 &amp; \text{otherwise} \end{cases}$</td>
</tr>
<tr>
<td>GMSK$^2$</td>
<td>$g(t) = \frac{1}{2T} \left[ Q \left( 2\pi B \frac{t-\frac{T}{2}}{\sqrt{\ln 2}} \right) - Q \left( 2\pi B \frac{t+\frac{T}{2}}{\sqrt{\ln 2}} \right) \right] \quad 0 \leq BT &lt; \infty$ $t \in [0, \infty)$</td>
</tr>
<tr>
<td>$Q(t) = \int_t^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{\tau^2}{2}} d\tau$</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.9: Phase response pulse $q(t)$ and frequency pulse $g(t)$ for GMSK.
be written as:

\[ \phi(t, u) = \pi h \sum_{i=0}^{n-L} u_i + 2\pi h \sum_{i=n-L+1}^{n} u_i q(t - iT) \]

\[ = \theta_n + 2\pi h \sum_{i=n-L+1}^{n} u_i q(t - iT) . \] (2.8)

It is obvious that \( \phi(t, u) \) exhibits infinite memory, even though the frequency pulse lasts only \( LT \). During the \( n^{th} \) symbol interval, the \( n^{th} \) input symbol \( u_n \) as well as those symbols indexed back to \( (n - L - 1) \) will exert a changing influence on the phase \( \phi(t, u) \). This is represented by the second term of Eq. (2.8). The first term in Eq. (2.8) represents the final phase influence produced by all symbols prior to \( t = n - L + 1 \):

\[ \theta_n = \left[ h\pi \sum_{i=0}^{n-L} u_i \right] \text{mod} 2\pi . \]

From Eq. (2.8), it can be seen that if the cumulative phase term \( \theta_n \) and the input vector \( (u_{n-L+1}, \ldots, u_n) \) are known, the CPM signal \( s(t) \) over the interval \( nT \leq t < (n + 1)T \) is completely specified. This suggests a state-variable description of the CPM signal. The state of the CPM modulator at the \( n^{th} \) symbol interval can be described by the state vector

\[ S(n) = (\theta_n, u_{n-1}, u_{n-2}, \ldots, u_{n-L+1}) . \]

The state vector consists of the phase state \( \theta_n \) and \( M^{(L-1)} \) correlative states for partial response systems. Although in theory the modulation index \( h \) can take on any real value, we observe that the state description of CPM will not be useful unless \( h \) is a rational number. If \( h \) is not rational, the CPM scheme will have an infinite number
of states. Suppose $h = \frac{m}{p}$, where $m$ and $p$ are relatively prime, then $\theta_n$ will have $2p$ different values if $m$ is odd, or $p$ different values if $m$ is even. For a CPM system with modulation index $h = \frac{m}{p}$, $M$-ary input symbols and pulse duration $LT$, the number of states, $N_{\text{state}}$, that (at most) is needed to describe the modulation scheme is given by:

$$N_{\text{state}} = \begin{cases} 
2pM^{(L-1)} & \text{if } m \text{ is odd} \\
pM^{(L-1)} & \text{if } m \text{ is even} 
\end{cases}$$

2.3.2 Description of CPM as Memoryless Modulation Preceded by Coding

The output signal $s(t)$ (Eq. (2.7)) at time $nT \leq t < (n+1)T$ is clearly only dependent on the state vector, $S(n)$, and the $n^{th}$ input symbol, $u_n$. This leads to a realization of the CPM modulator in the form shown in Figure 2.10, where the continuous phase encoder (CPE) is a rate $R = 1$ trellis code. Rimoldi [5] referred to this representation of CPM as a cascade of a part having memory (the CPE) and a part that is memoryless (a memoryless modulator). The modulator takes one of the $M$ possible inputs at each symbol interval, the finite state machine enforces the continuous phase constraint and chooses one of the $2pM^{L}$ (or $pM^{L}$ if $m$ is even) possible output waveforms from the memoryless modulator. As in the case of convolutional encoders, the continuous phase encoder can be described by a trellis diagram. Figure 2.11 shows the trellis diagram for $2REC$, $h = 1/2$, $M = 2$. This is a trellis with 8 states. The trellis representation
suggests that the optimum coherent detection of CPM can be performed by means of maximum likelihood sequence detection (MLSD) (for example, the Viterbi algorithm) or MAP algorithm. In fact, Osborne and Luntz [16] proposed a symbol-by-symbol coherent receiver for CPM, which is shown in Fig. 2.12, where $s(t, u_1, D_k)$ is the CPM signal with $u_1$ as the first symbol and $D_k$ is the $(N - 1)$-tuple $D_k = (u_2, \ldots, u_N)$. $D_k$ can have $q = M^{N-1}$ different possibilities. The symbol-by-symbol receiver computes $M$ likelihood parameters

$$
\begin{align*}
   l_1 &= \sum_{j=1}^{q} \exp \left[ \frac{2}{N_0} \int_0^{NT} r(t)s(t, 1, D_j) \, dt \right] \\
   l_2 &= \sum_{j=1}^{q} \exp \left[ \frac{2}{N_0} \int_0^{NT} r(t)s(t, -1, D_j) \, dt \right] \\
   &\vdots \\
   l_{M-1} &= \sum_{j=1}^{q} \exp \left[ \frac{2}{N_0} \int_0^{NT} r(t)s(t, M-1, D_j) \, dt \right] \\
   l_M &= \sum_{j=1}^{q} \exp \left[ \frac{2}{N_0} \int_0^{NT} r(t)s(t, -(M-1), D_j) \, dt \right]
\end{align*}
$$
Figure 2.12: Coherent CPM receiver of Osborne and Luntz [16].
for the \( M \) possible input symbols over an observation interval of \( N \) symbol duration and decides the data symbol \( u_1 \) that corresponds to the largest of \( l_1 \) to \( l_M \). \( N_0/2 \) is the noise power.

In this thesis, binary CPM scheme with modulation index \( h = \frac{1}{2} \) will be investigated. For \( h = \frac{1}{2} \), \( M = 2 \), \( \theta_n \) has only 4 different values. Thus the total number of states is \( 4 \times 2^{L-1} \).

### 2.3.3 Minimum Shift Keying: An Example

MSK is a full response \((L=1)\) CPM scheme with a rectangular frequency pulse \( g(t) = \frac{1}{2LT} \). It has been shown in [17] that the coded-modulation representations of MSK is not unique. One of the coded-modulation representation of MSK is shown in Fig. 2.13. The memoryless part in the realization in Fig. 2.13 is a mapper of 2-tuples \( \mathbf{c} = (c_1, c_2) \) to sinusoids \( s_n(t) \). The mapping rule is shown in Table 2.2.

Morales-Moreno [18] had shown that with the Gray Code mapping shown in Table 2.2, the squared Euclidean distance between a pair of signal \( s_i(t) \) and \( s_j(t) \) and the Hamming distance between the corresponding 2-tuples \( \mathbf{c}_i \) and \( \mathbf{c}_j \) are linearly related.
Table 2.2: Mapping of \((c_1, c_2)\) to channel signal \(x(t)\)

<table>
<thead>
<tr>
<th>((c_1, c_2))</th>
<th>(s_n(t))</th>
<th>(x(t))</th>
</tr>
</thead>
<tbody>
<tr>
<td>((0, 0))</td>
<td>(s_1(t))</td>
<td>(A \cos[2\pi(f_c - \frac{1}{4T})t])</td>
</tr>
<tr>
<td>((0, 1))</td>
<td>(s_2(t))</td>
<td>(A \cos[2\pi(f_c + \frac{1}{4T})t])</td>
</tr>
<tr>
<td>((1, 0))</td>
<td>(-s_2(t))</td>
<td>(-A \cos[2\pi(f_c + \frac{1}{4T})t])</td>
</tr>
<tr>
<td>((1, 1))</td>
<td>(-s_1(t))</td>
<td>(-A \cos[2\pi(f_c - \frac{1}{4T})t])</td>
</tr>
</tbody>
</table>

Figure 2.14: A simple coded modulation system.

2.4 Iterative Decoding of Concatenated Convolutional Code and CPM

Communications systems today usually consists of a cascade of forward error correction code and a modulator. The simplest example is the concatenation of a convolutional code and a modulator, as shown in Fig. 2.14, the receiver of this simple system is shown in Fig. 2.15.

Figure 2.15: A receiver for the simple coded modulation system.
Another non-trivial example of coded modulation is shown in Fig. 2.16. In this example, the data is precoded using a serial concatenated code, which is a serial concatenation of two convolutional encoders separated by an interleaver, before modulation. It is obvious that the serial concatenated code in this system can be decoded iteratively, as explained in Section 2.2. The receiver for such system consists of a demodulator followed by an iterative decoder, as shown in Fig. 2.16. Notice that in this example, the communication between the demodulator and the decoder is \textit{"one-way"} (i.e., the decoder can use information from the demodulator, but the reverse is not true). For systems that employ modulation schemes with memory (such as CPM), performance can be improved by allowing \textit{"two-way"} information exchange between demodulator and decoder, such that the information added by the decoder can be used in the demodulation process.

Given the discussion in Section 2.3.2, it becomes clear that demodulating a CPM signal is similar to decoding a trellis code. A simple implementation of coded CPM that allows joint decoding/demodulation is a direct cascade of a convolutional encoder, an interleaver and a CPM modulator. This is shown in Fig. 2.18. This configuration is almost identical to the configuration shown in Fig. 2.6 and it is obvious that the iterative decoder shown in Fig. 2.7 can also be used to jointly demodulate and decode the coded CPM system. The decoding process is identical to that being described in Section 2.2. Here the probability information computed by the CPM
decoder/demodulator and the APP decoder of the outer code are used jointly in the detection process. The demodulator is able to refine its decision by utilizing the information provided by the outer APP decoder.

A natural extension to the coded CPM system described above is to add another encoder between the interleaver and the CPM modulator. This is shown in Fig. 2.19. The key to implement the iterative decoder/demodulator for this system is to combine the trellises of the inner code and CPM. The combination of the inner convolutional encoder and the continuous phase encoder of CPM is viewed as one entity. The combined trellis code is called a signal space code [18], whose output symbols preserves the continuous phase property of the transmitted signal. One of the constituent decoders in the iterative decoder for this system is the APP decoder for this signal space code.
Figure 2.19: Coded CPM with combined code and CPM trellis.
Chapter 3

Performance Simulation of Coded CPM

The performance of iterative decoding of coded CPM is evaluated using Monte Carlo simulation. In this chapter, the system model, channel model and simulation parameters are described; and simulation results are presented. Two types of channels are studied in the simulation: the AWGN channel and the frequency non-selective Rayleigh fading channel.

3.1 System Model

The communications system model used in this thesis is shown in Fig. 3.1. The simulation methodology will be outlined below; but first it is important to list the critical assumptions that underlie the simulation approach.

- It is assumed that the receiver down-converts the signal to complex baseband, passes it through an ideal anti-aliasing filter, then samples it. The bandwidth of
the anti-aliasing filter is wide enough so that practically all of the signal energy is passed, including the signal energy that is spread by the Doppler spread in the fading channel. In this case, all modeling and analysis can be done in complex baseband.

- It is assumed that the fading process is slow such that the random phase offset caused by fading can be taken care of by the frequency synchronization stage in the receiver front end. We shall henceforth assume coherent reception.

- It is also assumed that in fading channels, no channel state information (CSI) is available.

### 3.1.1 Channel Model

The simulated channel impairments are AWGN and slow flat Rayleigh fading. In Fig. 3.1, the additive distortion is a white Gaussian noise and the multiplicative distortion simulates the effect of Rayleigh fading. A fading process filter can be used
to simulate the frequency non-selective Rayleigh fading process. The transfer function of the filter is \( H(f) \), which is given by:

\[
S(f) = |H(f)|^2
\]

where \( S(f) \) is the Doppler power spectrum of the fading channel. The input of the fading process filter is an independent complex-valued white Gaussian process with zero mean; the output, \( a(t) \), is a complex-valued Gaussian process with a Rayleigh faded envelope. In this thesis, the filtered complex-valued Gaussian process \( a(t) \) is simulated using Jakes' method \([19]\). The detailed implementation of the fading channel simulator is described in Appendix B. The AWGN channel (i.e., no fading) is simulated by setting \( a(t) = 1 \).

### 3.1.2 Coded CPM Transmitter

Three different types of coded CPM schemes will be studied in this thesis. In each case, part of the system can be identified as a serial concatenated code, similar to the one shown in Fig. 2.6, and therefore, the structure of the iterative decoder for all system will be identical to the decoder shown in Fig. 2.7. The only difference between different coded CPM schemes are the constituent decoders (the blocks labeled “DEC 1” and “DEC 2” in Fig. 2.7). The overall rate for all systems is \( R = 1/3 \).

**Type 1 Coded CPM**

The first type of coded CPM system, which will be referred to as *Type 1* coded CPM, consists of a cascade of a convolutional encoder as an outer code, an interleaver and a CPM modulator. The continuous phase encoder of the CPM modulator is viewed as the inner code of a concatenated system. Iterative decoding algorithm is used to decode the outer code and CPM trellis code as a serial concatenated code. This idea
Figure 3.2: Transmitter of a Type 1 coded CPM system.

is illustrated in Fig. 3.2. The decoder structure is shown in Fig. 2.7. Referring to Fig. 2.7, the block "DEC 1" will be the constituent decoder for the outer code and "DEC 2" will be the constituent decoder for the continuous phase code of CPM.

**Type 2 Coded CPM**

In the special case where the modulation scheme is MSK, a convolutional encoder of rate $R = k/l$ is inserted between the interleaver and the CPM modulator. The trellis of the rate $R = k/l$ inner encoder and that of the continuous phase encoder are combined to form a new inner code, called a signal space code. The resulting system will be referred to as *Type 2* coded CPM, as shown in Fig. 3.3. One important constraint in designing a signal space code is that the continuous phase property of the output signal has to be preserved. It will be seen later that this restriction limits the performance gain by coding in a concatenated system. The constituent decoders for the iterative decoding process will be the APP decoder for the outer code and the APP decoder for the signal space code. In this thesis, the outer code in Type 2 coded CPM is a rate $R = 1/m$ convolutional code and the inner code is a rate $R = m/3$ signal space code, so that the overall rate of the system is $R = 1/3$.

**Type 3 Coded CPM**

The structure of the transmitter of *Type 3* coded CPM is very similar to that of Type 2 system. However, in decoding Type 3 coded CPM systems, the memory of the
Figure 3.3: Transmitter of a Type 2 coded CPM system.

Figure 3.4: Transmitter of a Type 3 coded CPM system. The memory of the modulation is not used in iterative decoding.

Modulation is not used in the iterative decoding process, as shown in Fig. 3.4. This type of coded CPM can be used as a benchmark to study the effect of utilizing the memory of the modulation in joint decoding/demodulation.

Convolutional Encoders

All convolutional encoders used in this thesis are listed in Table 3.1 and Table 3.2. The minimum free distance\(^3\) \(d_{\text{min.f}}\) of the convolutional codes are also shown in the table.

\(^3\)The minimum free distance is the minimum Hamming distance between all pairs of distinct complete convolutional code words.
Table 3.1: Rate 1/n FF convolutional encoders used in this thesis.

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Rate</th>
<th>Number of states</th>
<th>$d_{min,f}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(7, 5)</td>
<td>1/2</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(15, 17)</td>
<td>1/2</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>(23, 35)</td>
<td>1/2</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>(5, 7, 7)</td>
<td>1/3</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>(25, 33, 37)</td>
<td>1/3</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>(133, 145, 175)</td>
<td>1/3</td>
<td>64</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 3.2: Rate 2/3 convolutional encoders used in this thesis.

<table>
<thead>
<tr>
<th>Encoder transfer function matrix</th>
<th>Number of states</th>
<th>Type</th>
<th>$d_{min,f}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\begin{bmatrix} 1 + D &amp; D &amp; 0 \ 1 &amp; 1 + D &amp; 1 \end{bmatrix}$</td>
<td>4</td>
<td>non-recursive</td>
<td>3</td>
</tr>
<tr>
<td>$\begin{bmatrix} 1 &amp; 0 &amp; \frac{1+D}{1+D} \ 0 &amp; 1 &amp; \frac{D}{1+D} \end{bmatrix}$</td>
<td>4</td>
<td>recursive</td>
<td>2</td>
</tr>
<tr>
<td>$\begin{bmatrix} 1 + D + D^2 &amp; D^2 &amp; 1 \ D &amp; 1 + D^2 &amp; 1 + D + D^2 \end{bmatrix}$</td>
<td>16</td>
<td>non-recursive</td>
<td>5</td>
</tr>
<tr>
<td>$\begin{bmatrix} 1 &amp; 0 &amp; \frac{1+D}{1+D+D^2} \ 0 &amp; 1 &amp; \frac{D^2}{1+D+D^2} \end{bmatrix}$</td>
<td>16</td>
<td>recursive</td>
<td>3</td>
</tr>
</tbody>
</table>

33
3.1.3 Modulator

The system is simulated entirely in the complex baseband equivalent domain. Therefore, the modulator model is simply

\[ s_n(t) = \cos[\phi(t, u)] + j \sin[\phi(t, u)] \]

where \( \phi(t, u) \) is the information carrying phase defined in Eq. (2.8), \( s_n(t) \) is the complex baseband signal in the \( n^{th} \) symbol interval. Two CPM schemes were studied in this thesis: MSK and GMSK with \( BT = 0.3 \). The definition of the frequency pulse, \( g(t) \) for each modulation scheme is listed in Table 2.1. In the case of GMSK with \( BT = 0.3 \), the frequency pulse \( g(t) \) is truncated to a 3 symbol period and for GMSK with \( BT = 0.5 \), \( g(t) \) is truncated to 2 symbol period. A sampling rate of 8 samples per symbol is used in the simulation. This sampling rate was determined experimentally by running simulations with different sampling rates. It was found that no further significant gain in performance results for any sampling rate greater than 8 samples per symbol.

3.1.4 Interleaver

The interleavers used in this simulation are pseudo random interleavers. Pseudo random interleavers are computer generated interleavers that permutes the bits in a block arbitrarily. For a given block length, the same interleaver is used for all systems. No specific design rule was applied in choosing interleavers. The interleavers used in this thesis are listed in Appendix B.
3.1.5 Demodulator

Consider the CPM modulator in Fig. 2.10. Let $N_{\text{output}}$ denote the number of possible output CPM signal. In each symbol interval, the continuous phase encoder chooses one of the $N_{\text{output}}$ possible output waveforms from the memoryless modulator and transmits the signal through the channel. Suppose the $N_{\text{output}}$ possible output waveforms are

$$\{s_1(t), \ldots, s_{N_{\text{output}}}(t)\}.$$

In order to perform iterative decoding, the demodulator needs to estimate the APP of the received signal. If the CPM signal $s_i(t)$ is transmitted through an AWGN channel during the $n^{th}$ symbol interval, the receiver will observe the signal

$$r(t) = s_i(t) + n(t)$$

$$= \sqrt{\frac{2E}{T}} \cos[\omega_c t + \Phi_i(t)] + n(t)$$

for $nT \leq t < (n+1)T$ and the noise $n(t)$ is Gaussian and white. To estimate the APP, $\Pr[s_j(t)|r(t)]$, for all $j \in [1, N_{\text{output}}]$, we notice that the log likelihood function with respect to the signal $s_j(t)$ can be computed by [20]

$$\ln[\Pr(r(t)|s_j(t))] \approx - \int_{nT}^{(n+1)T} [r(t) - s_j(t)]^2 dt = l_j.$$

$l_j$ can also be written as

$$l_j = -E_r - E_{s_j} + 2 \int_{nT}^{(n+1)T} r(t)s_j(t) dt$$

$$= K + 2 \int_{nT}^{(n+1)T} r(t)s_j(t) dt$$
where \( E_r \) and \( E_s \) represent the energy of \( r(t) \) and \( s_j(t) \) respectively over one symbol interval. For constant envelope CPM signal with very large carrier frequency, \( E_s \) is constant for all \( j \); also, \( E_r \), the energy of \( r(t) \), is independent of \( j \). Therefore, the sum of the energy terms can be lumped into a constant, \( K \), that is independent of \( s_j(t) \). Moreover, notice that for a given \( r(t) \) and equiprobable \( s_j(t) \), the likelihood probability is proportional to the APP, that is

\[
\Pr[r(t)|s_j(t)] \propto \Pr[s_j(t)|r(t)].
\]

The metric for the detection process is the correlation

\[
Z_n(j) = \int_{nT}^{(n+1)T} r(t) s_j(t) \, dt = \int_{nT}^{(n+1)T} r(t) \sqrt{\frac{2E}{T}} \cos(\omega_c t + \Phi_j(t))
\]

(3.1)

where \( s_j(t) = \sqrt{\frac{2E}{T}} \cos(\omega_c t + \Phi_j(t)) \). Eq. (3.1) is the basis for a correlation receiver, in which all possible transmitted signals, \( s_j(t) \), in one symbol interval are correlated with the received signal \( r(t) \). The correlation in Eq. (3.1) gives an estimation of the APP:

\[
\exp[Z_n(j)] \propto \Pr[s_j(t)|r(t)].
\]

The demodulator can implemented using a bank of correlators as shown in Fig. 3.5. The probability information \( \Pr[s_j(t)|r(t)] \) is used as input for the iterative decoder. Since simulations are carried out in the complex baseband, \( Z_n(i) \) can be computed using the complex baseband signal as well.

Suppose the signal \( s(t) = \sqrt{2E/T} \cos(\omega_c t + \psi(t)) \) is sent through an AWGN channel, the received signal is \( r(t) = s(t) + n(t) \) where \( n(t) \) is bandpass Gaussian noise that can be expressed in bandpass form

\[
n(t) = n_I(t) \cos(\omega_c t) - n_Q(t) \sin(\omega_c t)
\]
Figure 3.5: The correlator bank demodulator for CPM.

where \( n_1(t) \) and \( n_Q(t) \) are independent Gaussian processes with means and variances

\[
\begin{align*}
E[n(t)] &= E[n_1(t)] = E[n_Q(t)] = 0 \\
E[n^2(t)] &= E[n_1^2(t)] = E[n_Q^2(t)] = N_0 B
\end{align*}
\]

where \( B \) is the bandwidth and \( E[\cdot] \) denotes the expectation operation. \( s(t) \) can be expressed in quadrature components as

\[
s(t) = \sqrt{\frac{2E}{T}} \cos(\omega_c t + \psi(t)) = \sqrt{\frac{2E}{T}} \cos(\omega_c t) \cos[\psi(t)] - \sqrt{\frac{2E}{T}} \sin(\omega_c t) \sin[\psi(t)].
\]
Let

\[ I(t) = \cos[\psi(t)] \]
\[ Q(t) = \sin[\psi(t)] , \]

the baseband equivalent quadrature components of the received signal are

\[ \hat{I}(t) = \sqrt{\frac{2E}{T}} I(t) + n_f(t) \]
\[ \hat{Q}(t) = \sqrt{\frac{2E}{T}} Q(t) + n_Q(t) . \]

Therefore, the received signal \( r(t) \) can be written as

\[ r(t) = \hat{I}(t) \cos(\omega_c t) - \hat{Q}(t) \sin(\omega_c t) . \]

Substituting Eq. (3.2) into Eq. (3.1) and omitting all the double frequency terms, the correlation, \( Z_n(i) \), can be written as

\[ Z_n(i) = \int_{nT}^{(n+1)T} \hat{I}(t) \cos[\Phi_i(t)] \, dt + \int_{nT}^{(n+1)T} \hat{Q}(t) \sin[\Phi_i(t)] \, dt \]
\[ = Z_{nI}(i) + Z_{nQ}(i) . \]

where

\[ Z_{nI}(i) = \int_{nT}^{(n+1)T} \hat{I}(t) \cos[\Phi_i(t)] \, dt \]
\[ Z_{nQ}(i) = \int_{nT}^{(n+1)T} \hat{Q}(t) \sin[\Phi_i(t)] \, dt . \]

For each symbol duration, the demodulator calculates \( Z_{nI}(i) \) and \( Z_{nQ}(i) \), which are
the correlations over the $n^{th}$ interval of a certain signal and the received signal plus noise, denoted $\hat{I}(t)$ and $\hat{Q}(t)$. The value of $Z_n(i)$ is the sum of $Z_{nI}(i)$ and $Z_{nQ}(i)$. The operation is shown in Fig. 3.6.

Although the demodulator is designed for the AWGN channel, the same demodulator will be used to demodulate in the flat Rayleigh fading channel as a sub-optimal demodulator, assuming no CSI is available.

The demodulator described above is implemented in Type 1 and Type 2 coded CPM systems. For Type 3 system, the Osborne and Luntz type receiver described in Section 2.3.2 is used.

### 3.1.6 Iterative Decoder

The iterative decoder used in the simulation has the same configuration as the one shown in Fig. 2.7 except that now “DEC 2” is the APP decoder for CPM trellis code (in Type 1 coded CPM), combined CPM/convolutional trellis code (in Type 2 coded CPM) or inner convolutional code (in Type 3 systems). The trellis of the CPM continuous phase encoder cannot be truncated (or the continuous phase constraint
will be violated), and therefore, the decoder of the CPM trellis needs to be initialized by Eq. (2.4) and (2.6). The outer encoder is reset to the “0” state before a block is transmitted, hence, the decoder “DEC 1” is initialized by Eq. (2.1) and (2.3).

3.2 Performance of Different Coded CPM Systems

Before looking at the performance of various coded CPM systems, the performance of the uncoded CPM used in this thesis will be discussed here.

It is well known that optimum coherent demodulation of GMSK is inferior to MSK [2]. The performance bound for optimum coherent reception of MSK and GMSK is shown in Fig. 3.7 [2]. Using the Osborne type symbol-by-symbol receiver described in Section 2.3.2 with 5 symbol observation duration, the performance is shown in Fig. 3.8. It is obvious that using the bandwidth efficient GMSK modulation incurs a penalty in bit error rate performance.

The performance of the coded systems in both AWGN and fading channel will be studied. The fading channel considered in this thesis is a slow, frequency non-selective Rayleigh fading channel, which is commonly used to model land-mobile channel with narrow-band transmission.

The simulation parameters for the coded CPM systems are shown in Table 3.3. The parameter $f_mT$ is the product of maximum Doppler shift ($f_m$) and symbol duration ($T$). In the simulation, only 5 iterations were performed because no significant performance gain was achieved after 5 iteration. This can be shown in Fig. 3.9

3.2.1 Type 1 Coded CPM

Fig. 3.10 shows the performance of a Type 1 coded MSK and GMSK in AWGN with a rate $R = 1/3$, 16-state outer code (25,33,37). It can be seen that although uncoded GMSK performs worse than uncoded MSK, by exploiting the memory of the
Figure 3.7: Performance bound on coherent demodulation of MSK and GMSK with $BT = 0.3$.

Table 3.3: Simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Rate</td>
<td>8 samples/symbol</td>
</tr>
<tr>
<td>$f_mT$</td>
<td>0.02 (no CSI)</td>
</tr>
<tr>
<td>Block length ($N$)</td>
<td>200, 500, 1000 information bits</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>5</td>
</tr>
<tr>
<td>Overall rate</td>
<td>$1/3$</td>
</tr>
<tr>
<td>Modulation method</td>
<td>MSK, GMSK with $BT = 0.3$ and $BT = 0.5$</td>
</tr>
</tbody>
</table>
Figure 3.8: Performance of the Osborne and Luntz type receiver in AWGN with an observation duration of 5 symbol intervals for MSK and GMSK ($BT = 0.3$)
Figure 3.9: Performance of Type 2 coded MSK with (15,17) outer code and a 4-state rate $R = 2/3$ recursive inner code, $N = 500$ with 1,2,3,5 and 9 iterations.
modulation and iterative decoding, coded GMSK performs almost as well as coded MSK schemes for a block length of 500 information bits. The performance of the same system in a fading channel is shown in Fig. 3.11. The performance of both coded CPM schemes are about the same in a fading channel. However, the decoding complexity for the two systems are different because the memory in GMSK is longer than that in MSK, which implies a larger number of states in the continuous phase trellis code. The total number of states in the coded MSK system is 64 while the number of states in coded GMSK system is 256. Consider the Type 1 coded MSK system described above and a rate $R = 1/3$ coded GMSK system with a 4-state outer code $(5,7,7)$, the total number of states in this coded GMSK system is 64, which is equal to the total number of states in the coded MSK system. The performance of these two systems are shown in Fig. 3.12. A performance degradation of about 0.5 dB is observed at a bit error rate of $10^{-5}$.

Performance of the coded CPM system can be improved by using more powerful
Figure 3.11: Type 1 coded MSK and GMSK ($BT = 0.5$) with (25,33,37) outer code in a fading channel. $N=500$, 3 iterations, $f_mT = 0.02$

Figure 3.12: Type 1 coded MSK and GMSK ($BT = 0.3$) systems with 64 states in an AWGN channel. $N=500$, 3 iteration
Figure 3.13: Type 1 coded GMSK ($BT = 0.3$) with outer codes (5,7,7), (25,33,37) and (133,145,175) in a AWGN channel. $N=500$, 4 iterations.

outer codes (see Fig. 3.13) or longer block lengths (see Fig. 3.14 and Fig. 3.15). Although there is a significant performance improvement by changing the outer code from a 4-state code to a 16-state code, the gain by going from a 16-state code to a 64-state code is not substantial.

3.2.2 Type 2 Coded CPM

As presented in Section 3.1.2, in Type 2 coded CPM, a rate $R = k/l$ convolutional code is combined with the trellis of CPM to form a new rate $R = k/l$ trellis code, called a signal space code. The general configuration of a Type 2 coded MSK is shown in Fig. 3.16. Similar configuration can be implemented for GMSK with $BT = 0.5$. The connection between the signal space code and the memory mapper is illustrated in Fig. 3.17. The simplest case is to consider a 4-state, rate $R = 1$ signal space code and a rate $R = 1/3$, (25,33,37) outer code. A number of rate $R = 1$, 2-state signal
Figure 3.14: Type 1 coded GMSK ($BT = 0.3$) with outer code $(25,33,37)$ in a AWGN channel. $N=200$, $N=500$ and $N=1000$, 3 iterations.

Figure 3.15: Type 1 coded MSK with outer code $(25,33,37)$ in a AWGN channel. $N=200$, $N=500$ and $N=1000$, 3 iterations.
space codes had been investigated, both recursive and non-recursive, but only the configuration shown in Fig. 3.18 is effective. Consider a Type 2 coded MSK without an inner code as shown in Fig. 3.19. Fig. 3.20 shows the performance of the system with the rate $R = 1$ inner code and the system without an inner code.

It should be noted that the continuous phase constraint severely limits the performance of the system. Consider the mapping rule described in Table 2.2 and a signal space code shown in Fig. 3.21; it can be shown that the signal space code does not preserve the continuous phase property of the signal, thus the resulting system is not a
Figure 3.19: Type 2 coded MSK without an inner code.

Figure 3.20: Type 2 coded MSK with and without an inner code in AWGN. The result of the recursive code shown here is obtained after 4 iterations. The outer code in both cases is the rate $R = 1/3(25,33,37)$ code.
Figure 3.21: A rate $R = 1$ signal space code that violates the continuous phase constraint.

Figure 3.22: The effect of the continuous phase constraint on the performance of a coded modulation system.

coded CPM system. However, Fig. 3.22 shows that the performance of this system is better than a coded MSK system with the same total number of states. This demonstrates that the continuous phase restriction limits the performance improvement that can be obtained by coding.

Keeping the total number of states and the overall rate of the coded MSK system unchanged, one can implement a concatenated system shown in Fig. 3.23. The system consists of a rate $R = 1/2$ outer code and a rate $R = 2/3$ signal space code, which is a trellis code of the combined inner code and continuous phase code of MSK.
The topic of optimal combining of convolutional code and MSK had been studied by Morales-Moreno [18], and the idea of matched codes was introduced. Consider combining a convolutional code with MSK. For a given code rate and number of states, i.e., for a given complexity in the decoder, a matched code is a convolutional code that maximizes the free Euclidean distance of the system when combined with the continuous phase code of MSK. The 4-state non-recursive code listed in Table 3.2 is a matched code, which will be used in the simulation of the Type 2 coded MSK system. Another rate $R = 2/3$ code that will be used in the simulation is a 4-state recursive code, which is shown in the second row in Table 3.2. An 8-state rate $R = 1/2$ $(15, 17)$ encoder is used as an outer in both systems, so that the total number of states in both systems are 64.

The performance of the two systems in AWGN is shown in Fig. 3.24. It is surprising that although the combined non-recursive inner code (matched code) and MSK has a larger minimum free distance$^4$ ($d_{\text{min.f}} = 6$ for combined MSK/non-recursive inner code and $d_{\text{min.f}} = 3$ for combined MSK/recursive inner code, see Fig. 3.23), the

---

$^4$Since the mapping rule is chosen such that the Hamming distance of the code is linearly related to the Euclidean distance of the signals, a large minimum free distance implies a large minimum free Euclidean distance.
Figure 3.24: Type 2 coded MSK with (15,17) outer code, both 4-state recursive and non-recursive rate $R = 2/3$ inner code in a AWGN channel. $N=500$, 5 iterations.

Figure 3.25: Type 2 coded MSK with (15,17) outer code, both recursive and non-recursive 4-state rate $R = 2/3$ inner code in a AWGN channel. $N=500$ and 1000. 5 iterations.
Figure 3.26: Type 2 coded MSK with (15,17) outer code, both recursive and non-recursive 4-state rate $R = 2/3$ inner code in fading channel. $N = 500$, 5 iterations.

Recursive system performs better than a non-recursive system. It had been suggested in [8] that a recursive inner code is crucial and simulation result confirms this statement. The effect of a recursive inner code is more apparent when the block length is increased. Fig. 3.25 shows that significant performance improvement can be achieved by increasing block length only when the inner code is recursive. Similar result is shown for GMSK with $BT = 0.5$ in Fig. 3.27.

The performance of these two schemes in fading channel is shown in Fig. 3.26. The importance of a recursive inner code is also observed here.

The order of the inner and outer code can be exchanged, in that case, a rate $R = 2/3$ code is used as an outer code and $R = 1/2$ code is used as inner code. The overall rate is still $R = 1/3$. The effect of reversing the order of convolutional codes can is shown in Fig. 3.28 and 3.29. It is found that while reversing the order of convolutional code in coded MSK systems will not improve the performance, switching the order of code in GMSK systems (i.e., using a rate $R = 2/3$ outer code and a rate
Figure 3.27: Type 2 coded GMSK ($BT = 0.5$) with (7,5) outer code, both recursive and non-recursive inner code in the AWGN channel. The inner codes are listed in Table 3.2. $N=500$, 5 iterations
Figure 3.28: Type 2 coded MSK with $R = 2/3$ outer code (4-state non-recursive encoder shown in Table 3.2) and rate $R = 1/2$ recursive systematic (17/13) inner code combined with the MSK trellis. The channel is a AWGN channel. $N=500$, 5 iterations

$R = 1/2$ inner code) results in performance improvement.

When the performance of the different Type 2 coded MSK systems discussed above is plotted in the same figure (see Fig. 3.30), a number of points can be noted. The coded MSK systems that employs a 4-state matched code as an inner code (which maximizes the minimum free distance of the combined inner code and MSK code) performs worse than a coded system with recursive inner code, which has a smaller minimum free distance. This implies that maximizing the free Euclidean distance of the inner code may not be a good criterion for optimizing a coded MSK scheme.

3.2.3 Type 3 Coded CPM

The performance of a Type 3 coded MSK and GMSK system in AWGN is shown in Fig. 3.32. The outer code is a rate $R = 1/2$ (7,5) code and the inner codes are rate
Figure 3.29: Type 2 coded GMSK $BT = 0.5$ with $R = 2/3$ outer code (4-state non-recursive encoder shown in Table 3.2 and rate $R = 1/2$ recursive systematic (5/7) inner code combined with the GMSK trellis. The channel is a AWGN channel. $N=500$, 5 iterations
Figure 3.30: Performance of the different Type 2 coded MSK schemes in AWGN with $N=500$ and 5 iterations. Line A represents the system with (15,17) outer code and 4-state rate $R = 2/3$ non-recursive inner code; line B represents the system with (15,17) outer code and 4-state rate $R = 2/3$ recursive inner code; line C represents the system with a rate $R = 2/3$ 4-state non-recursive outer code and (17/13) recursive inner code; line D represents the system with (25,33,37) outer code and an inner code shown in Fig. 3.18.
Figure 3.31: Performance of 16-state, rate $R = 2/3$ recursive and non-recursive code in a AWGN channel with BPSK.

$R = 2/3$ recursive and non-recursive codes. The minimum free distance of the non-recursive code is larger than that of the recursive code, the performance of the two inner code with binary phase shift keying (BPSK) in AWGN is shown in Fig. 3.31.

A few points are noted. First, it is noticed that a recursive inner code outperforms a non-recursive inner code, which agrees with the result obtained in Type 2 coded MSK. Secondly, similar to the case of Type 1 coded CPM, the performance of MSK and GMSK are almost the same for the same outer code and inner code.

3.2.4 The Three Coded CPM Schemes Compared

The performance of the 3 types of coded MSK in AWGN is shown in Fig. 3.33. For Type 2 and Type 3 systems, only results for recursive inner codes are shown. The total number of states in each systems is the same. It is observed that while both recursive and non-recursive inner codes improve the system performance, a higher signal-to-noise ratio is required for the non-recursive system to attain a given bit
Figure 3.32: Type 3 coded MSK and GMSK with (7,5) outer code, 16-state rate $R = 2/3$ recursive and non-recursive inner codes in a AWGN channel. The inner codes are listed in Table 3.2 5 iterations $N=500$
Figure 3.33: The three coded MSK systems in AWGN with 4 iterations and \( N=500 \). The Type 1 coded MSK shown has a rate \( R = 1/3 \) (25,33,37) outer code; the Type 2 system with \( R = 2/3 \) inner code employs a 8-state recursive signal space code and a (15,17) outer code; the Type 2 system with \( R = 1 \) inner code uses a 4-state signal space code and (25,33,37) outer code; the Type 3 system uses a (7,5) outer code and a 16-state rate \( R = 2/3 \) 16-state recursive inner code.

error rate. Comparing the performance of Type 3 and Type 2 systems, it is obvious that combining the trellis of the inner code and modulation is necessary for a good coded CPM system.

Simulation results for coded MSK in fading channel is shown in Fig. 3.35. In the case of Type 2 and Type 3 coded MSK, only the results for recursive systems are shown. For the same complexity and block length, the best system is the Type 1 coded MSK system which does not show any flattening effect at high signal-to-noise ratio. The simulation results for coded GMSK in fading channel is shown in Fig. 3.34. Similar to the case of coded MSK, Type 1 coded GMSK performs the best in a fading
Figure 3.34: The performance of coded GMSK ($BT = 0.3$) in fading channel with 3 iterations and $N=500$. The Type 1 system shown uses a rate $R = 1/3$ (25,33,37) outer code and the Type 3 system uses rate $R = 1/2$ (7,5) code as outer code and 16-state, rate $R = 2/3$ code (both recursive and non-recursive) as inner code.
Figure 3.35: Performance of coded MSK in fading channel \( (f_m T = 0.02) \) with \( N=500 \) and 4 iterations. The system configuration is the same as those described in Fig. 3.33 environment.
Chapter 4

Conclusion

4.1 Summary of Results

In this thesis, we have shown that the inherent intersymbol interference in CPM system can be viewed as a kind of correlative coding, which can be utilized to jointly demodulate and decode data. By exploiting the memory of CPM, the performance of coded GMSK approaches that of coded MSK schemes but with higher spectral efficiency, despite the fact that uncoded GMSK is inferior to uncoded MSK.

It has been shown in [8] that a recursive inner code is crucial in a serial concatenated system. This thesis also confirms that the recursiveness of inner code is very important. Significant performance improvement can be achieved with increasing block length only when the inner code is recursive. Maximizing the minimum free Euclidean distance of the inner code does not seem to be a good criterion for optimizing coded MSK schemes.

It is also found that combining the trellis of the inner code and modulation is important in a coded CPM system. The importance of joint decoding/demodulation can be seen by comparing the performance of Type 3 system with the other two type
of coded systems.

4.2 Suggestion for Future Research

We believe that a lot of areas in iterative decoding of coded CPM are still unexplored and the potential to improve the system performance is large. We have shown that a recursive inner code is important for a coded CPM system; however, what are the criteria for choosing a good recursive code such that the performance of the system is the best?

Interleaver design is also an important concept in designing a good concatenated code. How should the interleaver in coded CPM be designed such that the overall system is optimized?

The complexity of the decoder in this system is high compared to conventional Viterbi detector. Another area of interest is to investigate the performance of the system with low complexity sub-optimal decoding algorithms, such as the Max-Log-MAP algorithm [21, 22] or the soft-output-Viterbi-algorithm (SOVA) [23].

The effect of changing the outer code to recursive convolutional code or block code is an interesting area of research.

A suboptimal demodulator for fading channels was used in this thesis. The performance improvement gained by using a better demodulator (such as that described in [24]) can be investigated.
Appendix A

Fading Channel Simulation

A.1 Jakes’ Simulator

In this thesis, the slow flat Rayleigh fading channel will be used as the fading channel model. Since the time varying channel impulse response is a complex Gaussian random process, a straightforward method of constructing a fading simulator is to multiply the radio signal by a complex-valued Gaussian random function. The spectrum of this function is determined by the Doppler spread of the channel. This method is shown in Fig. A.1, where $|H(f)|^2$ is given by the Doppler power spectrum. The limitation of this approach is that, in order to approximate the spectrum, a high order pole-zero filter is required. This filter will have a long impulse response and this will significantly increase the run time for software simulation. Another effective channel simulator was proposed by Jakes [19]. In Jakes’ simulator, the two filter noise components are generated by adding six or more sinusoidal signals. The frequencies are chosen to approximate the Doppler spectrum.

The simulator used in this thesis is shown in Fig. A.2. where $(M + 1)$ is the total
Figure A.1: Rayleigh fading simulator. $n_I(t)$ and $n_Q(t)$ are independent white Gaussian noise.

Figure A.2: Jakes' simulator used in this thesis.
number of oscillators, $\omega_m$ is the maximum Doppler shift. The other parameters are

\[
\begin{align*}
\omega_n &= \omega_m \cos \frac{2n\pi}{N} \\
\beta_n &= \frac{\pi n}{M+1} \\
\alpha &= 0 \\
M &= \frac{1}{2} \left( \frac{N}{2} - 1 \right).
\end{align*}
\]

In this thesis, $M+1 = 9$ oscillators are used to simulate the fading channel. The generated Gaussian random process $a_I(t)$ and $a_Q(t)$ are independent zero mean Gaussian processes with variances $\text{VAR}[a_I(t)] = M$ and $\text{VAR}[a_Q(t)] = M + 1$. The resulting complex-valued Gaussian process $a(t) = a_I(t) + ja_Q(t)$ will have a Rayleigh fading envelope.
Appendix B

List of Interleavers

The interleaver used in this thesis are computer generated pseudo random interleavers. The bits in the block are permuted according the the interleaving rule specified by the interleaver. All interleavers used in this thesis are listed below. The interleaver is specified by a interleaving table. For example, for a interleaver of size 5 bits, an interleaving table will be

$$3 \ 4 \ 0 \ 2 \ 1$$

which means that the bit at position 0 will be mapped to position 3, the bit at position 1 will be mapped to position 4, the bit at position 2 will be mapped to position 0, etc. If the input of the interleaver is \{a,b,c,d,e\}, the output will be \{c,e,d,a,b\}.

400-bit Interleaver

277  87  0  49  188  141  123  164  98  202  267  50  351  183  275  264  119  219  344  58  99  327  399
88  296  281  4  143  210  126  372  297  21  38  97  317  299  70  320  203  205  195  67  276  367  364
334  18  243  285  106  139  57  95  214  268  173  186  117  251  159  155  71  256  358  388  108  380
260  311  137  56  330  13  375  294  393  138  136  100  229  255  157  25  302  319  250  373  175  160
750-bit Interleaver

77 387 150 199 238 441 623 114 348 352 517 550 401 283 625 614 19 419 344 258 649 27
299 388 46 31 654 43 510 726 472 147 126 671 738 169 102 499 620 670 503 505 545 367
567 314 384 68 693 285 456 739 669 357 245 118 573 36 617 301 309 605 471 708 558 180
311 287 306 280 413 475 594 399 543 538 286 250 129 205 607 727 121 725 509 195 69 50
525 360 685 142 174 246 429 82 527 444 557 723 93 350 113 380 202 243 187 60 220 661
372 420 184 474 91 235 219 197 313 154 340 89 371 171 530 66 330 159 282 141 485 462
681 284 592 136 321 733 706 556 234 175 355 149 695 684 11 464 735 570 496 226 39 638
711 501 569 460 13 596 645 627 167 230 583 325 465 644 228 455 578 132 12 231 504 119
551 290 613 303 459 491 532 229 722 718 554 328 87 255 574 316 710 403 65 29 298 393
365 233 502 34 42 647 697 267 479 591 447 630 378 431 488 349 448 207 17 79 107 200
1000-bit Interleaver

77 887 400 449 988 941 123 364 98 602 267 50 151 783 875 864 519 419 344 258 899 527
799 888 296 281 404 543 10 726 972 897 126 421 238 517 499 870 920 3 5 795 867 876 567
564 134 818 443 285 706 739 919 857 495 814 868 573 786 117 51 559 355 471 456 958 308
180 260 311 537 56 530 413 975 94 793 538 536 500 629 455 357 477 621 225 759 695 102
319 775 360 913 435 892 174 746 429 582 277 194 557 723 93 350 613 630 202 243 437 560
720 411 137 122 170 184 224 841 235 149 969 947 63 840 589 171 780 816 80 503 909 782
891 985 712 681 784 92 386 614 571 483 806 984 425 105 399 181 945 684 511 464 735 943
320 996 789 711 501 569 960 637 13 346 895 627 917 730 83 75 715 860 114 394 478 78
397 318 132 262 215 481 4 369 290 363 846 303 709 741 282 979 472 718 54 539 578 587
324 66 460 403 815 779 298 300 393 865 781 733 221 502 792 147 924 447 208 17 729 187
594 380 378 738 702 599 198 457 767 329 107 67 722 200 545 591 623 90 645 299 269 374
838 110 843 904 966 581 606 131 416 36 813 523 848 655 286 907 620 889 279 218 352 805
547 321 480 667 590 963 776 959 434 664 807 442 359 997 921 542 315 106 946 677 487
103 680 164 353 866 832 295 830 274 226 974 33 685 586 654 683 507 417 409 401 85 383
24 839 577 835 811 852 770 561 744 64 430 653 70 465 512 914 451 660 991 827 690 679
145 801 898 894 961 59 575 371 325 128 513 161 82 546 219 636 766 910 768 798 407 752
951 234 649 849 962 529 27 698 674 439 273 398 764 216 245 377 553 707 927 448 570 189
596 671 183 970 521 402 691 335 130 608 246 86 395 491 124 396 785 936 384 253 952 842
201 244 48 881 141 869 824 115 263 144 340 268 556 541 940 29 794 223 518 89 475 428
995 616 563 388 740 287 504 431 343 52 522 821 714 901 163 509 146 387 476 370 272 810
278 450 301 823 957 368 121 229 688 675 736 615 905 778 687 228 461 704 304 747 879
993 837 326 788 310 555 81 38 858 20 127 418 342 169 57 692 433 414 791 312 372 459
981 950 191 971 745 389 217 266 446 565 772 454 796 488 934 330 632 55 748 693 207 415
365 732 336 829 95 12 647 160 808 724 633 462 49 334 265 155 862 903 22 193 99 65 628
2000-bit Interleaver

1077 887 400 449 988 941 123 1364 98 602 267 50 1151 1783 1875 1864 519 1419 344 1258
899 1527 799 888 296 281 404 543 1010 1726 1972 1897 126 421 1238 897 1517 1499 870
1920 1003 1005 1795 867 1876 1567 1564 1134 818 1443 285 1706 1739 919 857 495 1814
1868 673 1786 117 1051 559 1355 471 1456 1958 1988 1308 1180 260 311 537 56 1530 413
1975 1094 1899 793 538 536 500 629 1455 1357 1477 1621 1225 1759 1695 1102 319 1050
1573 1775 1360 1913 1435 892 174 1746 1429 582 1277 194 557 723 93 1350 1613 630 202
243 437 1560 1720 411 1137 122 170 1184 1224 841 235 1149 969 1947 1063 840 1589 151
621 171 1780 816 1080 503 909 782 1891 985 712 681 784 1092 386 1614 1571 483 456 1806
1984 1425 105 1399 181 945 684 1919 1511 464 735 943 1320 1996 1789 1888 711 501 569
1960 1637 13 346 895 1627 1917 1730 83 75 1715 1860 1114 394 1478 455 1078 1397 318
1895 132 1262 1215 481 1004 369 51 1857 1290 363 1846 1303 709 741 1282 979 472 1718
1054 539 1578 1587 5 324 1066 460 1403 815 779 1298 1300 393 865 1781 1733 221 502 1784
78 1792 147 1924 1447 1208 1017 729 187 1841 447 1594 380 378 1181 738 702 599 1198
457 1767 329 107 1067 722 972 1200 1545 1702 591 1623 1090 1645 299 1269 374 838 1110
1174 1843 1904 1966 581 606 917 131 1416 1036 813 1523 1887 848 1655 695 1709 1286 1907
620 1380 1889 279 218 1352 1805 1547 1321 480 667 590 963 1776 1959 434 1664 807 442
359 1997 1921 726 542 315 106 783 946 677 487 1103 1680 1164 1353 866 1832 295 1830
274 226 1974 1033 685 586 1654 683 507 417 1409 1401 1085 383 545 1024 1839 577 1411
282 835 811 852 770 1681 1561 744 715 1064 1430 1653 1070 465 512 1840 1914 94 451 1660
991 1827 1690 780 679 1145 1801 827 898 1894 961 680 59 575 1371 325 1852 1818 1128
1513 1161 1082 546 219 1636 766 1910 768 1798 1407 752 1951 234 649 849 128 1962 1529
27 698 674 439 1273 398 764 1126 216 245 1377 1553 707 927 448 1570 1189 1216 596 1969
1671 1183 970 521 1311 1285 402 1691 335 130 1608 1246 1086 395 1491 77 124 1396 785
936 200 64 1384 1253 952 1842 201 244 1048 1881 1141 739 570 869 1824 397 115 1263 1144
1340 1268 1707 477 556 541 1940 29 1794 1245 223 518 89 475 428 995 1616 1519 1563 388
1740 1287 504 431 343 52 522 821 608 1714 901 1163 1509 146 1387 664 1476 1779 303 370
272 690 810 278 430 450 1785 881 301 823 1957 1368 795 1121 1229 860 688 675 1736 1343
1615 1905 778 687 1228 451 1704 1304 1747 879 958 1993 837 326 788 310 1555 81 1170

73
References


