ON TRELLIS CODED CONTINUOUS PHASE FREQUENCY SHIFT KEYING

By

HSIN-HSYONG RICHARD YANG
B.S.E.E., National Tsing Hua University, Taiwan
M.S.E.E., New Jersey Institute of Technology, U. S. A.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Ph.D.

McMaster University
June 1994
©Copyright 1994

ON TRELLIS CODED CONTINUOUS PHASE FREQUENCY SHIFT KEYING

PH.D. (1994) MCMASTER UNIVERSITY
(Electrical and Computer Engineering) Hamilton, Ontario

TITLE: **On Trellis Coded Continuous Phase Frequency Shift Keying**

AUTHOR: Hsin-Hsyong Richard Yang
B.S.E.E., National Tsing Hua University, Taiwan
M.S.E.E., New Jersey Institute of Technology, U. S. A.

SUPERVISOR(S): Dr. D. P. Taylor
Professor,
Department of Electrical and Computer Engineering
McMaster University, Ontario, CANADA
Professor,
Department of Electrical and Computer Enginnering
University of Canterbury, Christchurch, New Zealand

NUMBER OF PAGES: xiii, 140

ABSTRACT

This thesis reports my research work in the area of trellis coded continuous phase frequency shift keying (CPFSK). Previous approaches [1, 2, 3, 4, 5] applied binary convolutional codes to CPFSK to achieve power and bandwidth efficiency. However, the work in [6] and part of this thesis show that no single approach among previous approaches can be outperformed by the others if only binary convolutional codes are considered.

A new coding scheme based on *convolutional codes on the ring of integers modulo-P* is shown to be a *natural* way to apply trellis coding to CPFSK [7]. Recent work has decomposed CPFSK into two parts; a linear encoder with memory, called the continuous phase encoder (CPE), and a memoryless modulator (MM), where the CPE often has a code structure defined over the ring of integers modulo-P. The combination of a modulo-P convolutional channel encoder (CE) and the CPE, is a linear modulo-P encoder. Design examples are given for rate 1/2 coded quaternary CPFSK with modulation indices 1/2 and 1/4, and rate-2/3 coded octal CPFSK with modulation index 1/8. Combinations are optimized in the normalized minimum Euclidean distance sense for a given total number of states in the overall maximum likelihood sequence estimation (MLSE) receiver. Numerical results show that this new coding scheme consistently achieves better performance than previous schemes [1, 2, 3, 4, 5].

An upper bound on the bit error probability (BER) for ring convolutionally encoded CPFSK is derived. The bound shows that feedback-free CPFSK usually has

a smaller error coefficient than CPFSK. The minimum Euclidean distance is a good parameter for estimating performance, and the ring convolutionally encoded CPFSK has a good BER for both moderate and practical signal to noise ratio.

Acknowledgement

I want to thank Professor Taylor for his encouragement and generous financial support during my program. I owe him the idea of using nonbinary codes to trellis encode CPFSK. To Drs. Jim Reilly, Patrick Yip, Manfred Kolster and Max Wong, I am grateful for their many helpful discussions. I also want to thank my colleagues in Communications Research Laboratory, especially to Wayne Dam, for his linguistic assistance and many important helps and to Milan Despinic for his enthusiasm. To my three good friends, Zhiqiang Bi, Tom Luo and Fidel Morales-Moreno, I want to say a special thank-you for their families' unfailing friendship and hospitality. I have had many happy times with them. Finally, I want to thank my family for their support and understanding during these years. Without this I could not have been able to finish my work. And finally, thanks to God for giving me this life and my strength when I was weakest.

Contents

List of Figures

List of Tables

Chapter 1

Introduction: Thesis Overview

The topic of this thesis is to study convolutional coding techniques applied to continuous phase frequency shift keying (CPFSK). The noise environment is additive white Gaussian noise (AWGN) and the receiver performs maximum likelihood sequence estimation (MLSE). To optimize the combination of coding and modulation, the best codes are chosen to achieve the largest Euclidean distance between the different code designs while holding the decoding complexity constant.

1.1 Trellis Coded Modulation for Digital Communications Systems

Due to the requirement of of reliable data communications while maintaining a large number of users in a finite transmission, modern coding and digital modulation techniques are necessary. A model of a digital-radio communication system is shown in Figure 1.1. The goal is to achieve power and bandwidth efficient transmission of data. In Figure 1.1, the source is assumed to be digital and memoryless. Some traditional definitions for coding and modulation are as follows: [8]

- Encoding: The introduction of redundant symbols to correct errors.

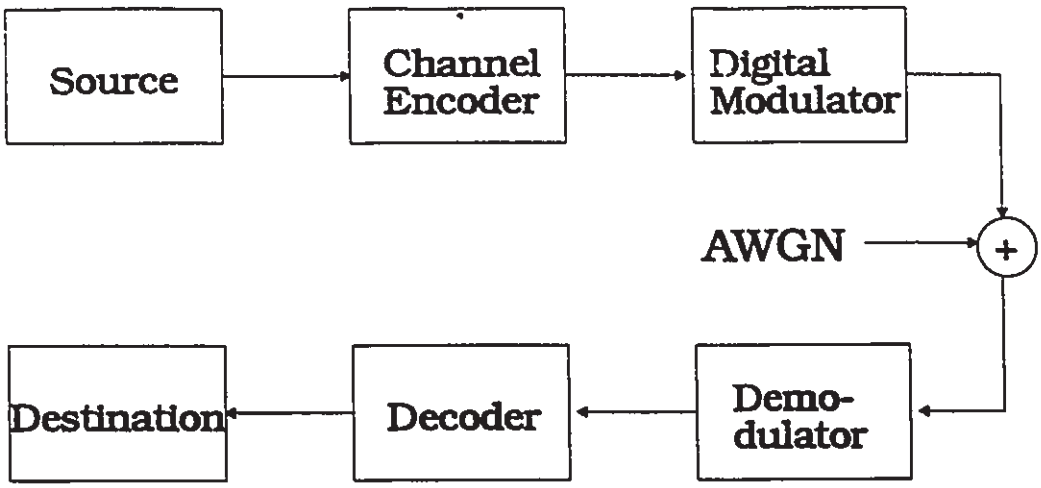


Figure 1.1: Digital radio communications system.

- Modulation: The conversion of symbols to a carrier waveform, usually a sinusoidal function of time.
- Demodulation: The conversion of the carrier waveform back into symbols.
- Decoding: The use of redundant symbols to correct data errors.

The encoder may use block codes, such as Reed-Solomon codes, or trellis codes, such as binary convolutional codes. For block codes, each transmitted codeword depends only on the current input information data. For trellis codes, however, the transmitted codeword not only depends on the current input information data but also on previous input data. The block encoder can be implemented with combinatorial logic, and the trellis encoder can be implemented with sequential logic. The modulator can be *memoryless*, such as phase shift keying (PSK), quadrature amplitude modulation (QAM), and amplitude shift keying (ASK), or with *memory*, such continuous phase modulation (CPM), tamed-FM (TFM), and partial response modulation. In *memoryless* modulations, the transmitted signals in any modulation interval are a function only of the present input. In modulators with *memory*, the transmitted signals in any modulation interval are a function of both the present and some of the past inputs. The statistics of the waveform can be stationary or nonstationary. This

thesis only considers the stationary additive white Gaussian noise (AWGN) channel.

In early coded digital communications systems, the encoding and modulation were designed independently. This approach has two drawbacks [9]. First, for the same source rate and modulation scheme, block or convolutional coding will widen the bandwidth of the signal compared to the uncoded system. Second, coding will decrease the transmitted energy per channel symbol, so more channel symbols will be in error compared to uncoded transmission. These observations seem rather discouraging. Because of these two drawbacks, decreased bandwidth efficiency and an increased number of errors in the received sequence are to be expected.

This problem remained at an impasse, until Massey, in a now well-known seminal paper [10], formally suggested the notion of improving the system performance by looking at the modulation and coding as a combined entity. This concept of seeing coding and modulation as one entity was later implemented by Ungerboeck's breakthrough papers [11] for PSK, QAM and ASK (using *memoryless* modulators). The concept is now called trellis coded modulation (TCM). Using the TCM approach with memoryless modulations, such as M-ary phase shift keying (MPSK) or quadrature amplitude modulation (QAM), significant coding gain can be achieved without bandwidth expansion.

The basic idea of TCM is to cascade a trellis encoder (using a convolutional code) with digital modulation using redundant signal sets. An approach, called set partitioning, is used to assign signals to each branch of the trellis. This coding approach can obtain up to 4 dB of gain with simple coding scheme and to 6 dB of gain with high complexity in the receiver's maximum likelihood sequence estimation (the Viterbi algorithm). An excellent tutorial treatment for TCM for *memoryless* modulations can be found in the encyclopedic paper [12] and text [13]. TCM has also been applied to fading dispersive channels [14], partial response channels [15], and other practical channels [13]. TCM was originally believed to be a nonlinear form of modulation. However, it has been recently found that some TCM schemes are

linear. In fact, some TCM signal space codes are linear codes over a ring or group [16, 17, 18]. New coding schemes and results using this new concepts are expected.

1.2 Trellis Coded Modulation for Continuous Phase Modulation

Continuous phase modulation (CPM) is a real constant envelope modulation. PSK, however, is not a constant envelope modulation since the transmitted signal is bandlimited by a bandpass filter so as to reduce the out-of-band spectral sidelobes and prevent interference with adjacent channels. This effect is highly undesirable when the signal undergoes nonlinear amplification, as in satellite repeaters [9]. CPM has advantages when nonlinear amplifiers are applied, as quoted from [19]: "Saturating amplifiers for constant envelope signals are 2 dB to 3 dB more efficient than purely linear amplifiers, and this 2 dB to 3 dB must be counted as a gain for constant-envelope schemes relative to variable-envelope schemes." Moreover, CPM can also, in principle, perform near channel capacity in an AWGN channel for a given bandwidth and energy constraint [19].

When more energy and bandwidth efficiency are needed for CPM, trellis-encoding techniques can be applied. There are two effective ways which are to employ trellis encoding with CPM. One, is to cyclically vary the modulation index h of the modulator, giving multi-h phase codes [20] (this includes its relative, see [19]). The other cascades external convolutional codes to single-h CPM [1]. However, relatively simple two-h schemes have inferior energy/bandwidth performance compared to comparable convolutional-code/CPM combinations. Summaries of trellis encoding for CPM can be found in [8, 19, 21].

CPM can be divided into two major subsets: full response CPM (continuous phase frequency shift keying), and partial response CPM [22, 23]. In this thesis, we will consider cascading convolutional codes with CPFSK. CPFSK is very useful for

digital mobile-radio transmission systems because of its constant-envelope and small RF bandwidth requirement. Combining coding with CPFSK to achieve coding gain is, therefore, of practical interest.

For a trellis-encoded *memoryless* modulator, the complexity of the receiver depends only on the complexity of the convolutional encoder. However, for encoding a modulator with *memory*, the overall complexity of the receiver is due to the combination of the memory of the convolutional encoder and the memory of the CPM modulation. The *traditional approach* [1, 2, 24, 8] finds the best combination of coding and modulation without considering the interaction between the external encoder and the memory of CPM, which may change the complexity of the receiver. Thus, the codes found by the conventional approach may not be optimal for a fixed complexity receiver. However, the work of Massey [25] suggested that continuous phase modulation (CPM) be decomposed into two parts: one an encoder with memory, the other a memoryless modulator. A general decomposition model of M-level CPM, comprised of a continuous phase encoder (CPE) and a memoryless modulator (MM), has since been derived by Rimoldi [26, 4, 27] (a similar model for MSK has been derived in [28]). He showed that the CPE is a linear (modulo some integer P) time invariant encoder and the MM is a time invariant device. It is then of interest to optimally combine an external channel encoder with the CPE. All approaches [3, 4, 5, 29], namely the *matched encoding approach*, the *decomposition approach*, and the *double trellis encoding approach*, employ this concept to find the optimal combination of coding and modulation. New codes have been found using these concepts which have slightly better coding gains compared to the traditional approaches.

In previous work [1, 3, 4, 30, 5], as shown in Fig. 1.2, a binary convolutional channel encoder (CE), **G**, is connected to the M-ary CPFSK via a binary to M-level mapper, **Q**. Alternatively, to avoid the mapper **Q**, and to combine the binary CE with the CPE in a *natural* [26, 4] way, a scrambler $\mathbf{T}'$ was cascaded with the CPFSK model to allow a binary feedback-free CPE.

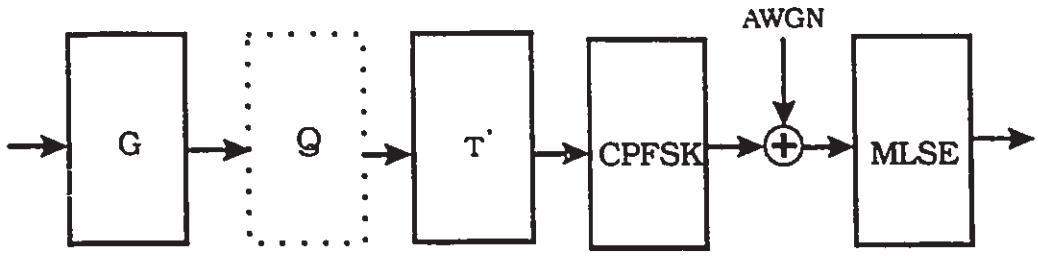


Figure 1.2: Binary convolutional encoded CPFSK system (the dotted box can be omitted for the decomposition approach).

However, it appears [6] that these coded systems produce different performance for feedback-free and feedback CPFSK. In fact, it appears that no single approach to encoding CPFSK will consistently obtain the best performance if only binary convolutional codes are employed. Based on the code structure of CPFSK, this thesis suggests a new convolutional encoded CPFSK system [31]. It shows that a *ring convolutional encoder* [16], called a *modulo-P encoder* [32], is a more *natural* way to combine the CE/CPE pair than a binary convolutional encoder [7], and that the resulting coded modulation consistently obtains better performance than in previous approaches.

An upper bound for the bit error rate of the ring encoded CPFSK system is derived. It is shown that the bit error probability of ring encoded CPFSK system performs well for both practical and moderate signal to noise ratios. Coding gain obtained by ring encoded CPFSK system is very close to the coding gain predicted by the practical channel capacity [10], R_0. For example, it is predicted by R_0 [27] that rate-1/2 coded quaternary CPFSK with $h = 1/4$ has approximately 7 dB gain over MSK, and we have achieved 6 dB for the complexity of 256 states.

In the CPM literature, this work is the first one that employs non-field convolutional codes to CPM. The coding gain is high, for example, rate-1/2 modulo-4 coded quaternary $h = 1/4$ CPFSK has coding gain over minimum shift keying (MSK) from 2 dB to 6 dB, for the number of states of 4 states to 256 states in the MLSE receiver. This is very effective compared to all the previous work. Previous work has

only minor coding gain for relatively simple complexity in the receiver (0.3 dB for the complexity of 4 states [4]). Recently, people have found that using the concepts of ring convolutional codes or group convolutional codes, it is possible to generate new codes for TCM [17, 33, 18]. This work is the first one that applies ring encoders to digital modulations and obtains significantly improvement over the traditional approach. Therefore, it is very interesting to construct new coding scheme to other digital modulations.

1.3 Thesis Overview

In chapter 2, we describe the decomposition model of CPFSK and a useful subset of CPFSK which has modulation index $h = K/P$ and P^{k_M} modulation levels, where k_M is an integer. The resulting CPE is seen to be a linear convolutional encoder over the ring of integers modulo-P [26, 4]. A precoded feedback-free form of CPFSK [31], defined over the same code structure as the CPE, will be introduced to simplify the code search process. We will also modify the incremental squared Euclidean distance (ISED) formula [4] for computing the normalized squared Euclidean distance between codewords of the *overall encoder* (the combination of the CE/CPE pairs.)

In chapter 3, binary convolutional encoded CPFSK systems are compared. In addition to our previous work [6], we compare another approach, called double trellis encoded CPM [5], to the other's previous work. The conclusion is that no single approach in previous work (including the double trellis encoded approach) can be outperformed by the others if only binary convolutional encoders are considered.

In chapter 4, we will define the ring convolutional coded CPFSK system. Here, we consider an external convolutional encoder G defined over the ring of integers modulo-P. Similar encoders have been applied to M-ary PSK [32, 16, 34, 35]. It has been proved that a systematic ring convolutional encoder is minimal and always non-catastrophic [36, 33, 37]. Therefore, we will assume the encoder is systematic. It will

be shown that the combination of the CE/CPE pair is an overall ring convolutional encoder. Because the channel encoder uses the same code structure as the CPE, a mapper is then not necessary. That is, we can input the coded information sequence directly into the CPE. It is not necessary to use a precoder to produce a feedback-free CPE. However, a feedback-free form of CPFSK can simplify the code search process.

The structure of the overall encoder is then studied. Two approaches, one *directly* deciding the number of states of the overall encoder [3, 29] and a second *indirectly* finding the transfer function of the overall encoder [4], are considered. Design results for several modulation indices are given and compared to [6], along with another recent approach [5] for a fixed complexity MLSE receiver. It is concluded that this new coded CPFSK system consistently achieves performance gains at least as large as described in previous work, and with better coding gains being obtained in many cases.

In chapter 5, the average transfer function technique [38, 39, 40, 41, 14, 42] is applied to ring convolutional coded CPFSK. An upper bound on the BER is derived and applied to different forms of coded or uncoded CPFSK. Numerical results show that the feedback-free forms of CPFSK have a better BER than CPFSK and the normalized minimum Euclidean distance is a good parameter to estimate the performance of ring convolutional encoded CPFSK. For some examples, the Euclidean distance alone gives a pessimistic performance prediction (this also happens in binary convolutional encoded systems [43]). It is also important to note that the ring convolutional encoded CPFSK has significant coding gain for both practical and small signal to noise ratios.

Chapter 6 gives out conclusions and discusses possible future work. Appendix A gives a brief derivation of the equivalent CPE with expanded code rate. Appendix B contains a short introduction to convolutional codes. This will give basic definitions and background for the readers.

Chapter 2

Continuous Phase Modulation (CPM) and Continuous Phase Frequency Shift Keying (CPFSK)

2.1 Introduction

In the following we recall the basic features of continuous phase modulated signals. Continuous phase modulation is the only real constant envelope digital modulation [19]. Other digital modulations, for example QPSK, still have envelope variation. In a radio channel with a nonlinear transmitter power amplifier, constant envelope signals are a virtual necessity; the alternatives are expensive compensation for nonlinearity and more DC power to support a less efficient linear amplifier. Continuous phase frequency shift keying is an important subset of CPM. This subset of signals has a narrow bandwidth occupancy and relatively simple complexity, and is therefore very practical for digital-radio transmissions.

Early work in developing CPFSK can be found in [44, 45, 46, 20], where the work of [44] is called FFSK or MSK and the works of Anderson and Taylor [20], Miyakawa, Harashima and Tanaka [46] are called multi-h modulation. This line of work was

continued and summarized by Aulin, Rydbeck and Sundberg in [22, 23]. Recently, combined convolutional coding with CPM became a necessity for bandwidth and power efficient transmission systems. Many works have been studied [1, 2, 8] and the coding gain for relatively simple complexity convolutional coded CPFSK is better than relatively simple multi-h modulation. It is predicted in a review article [19] that new convolutional coding combined with CPM is promised. From the suggestions of Massey's paper in [25], it would be desirable to decompose a CPM into a encoder combined with a memoryless modulator (this will be shown in the sequel). This model was then found by Rimoldi [26]. This model is important for combining coding with CPFSK because then the external encoder and the inner phase encoder can be optimized as a single entity. This chapter will introduce both the traditional and decomposed model. The decomposed model will be applied to the design of convolutional coded CPFSK systems.

In section 2.2, traditional descriptions of CPM are given. We will concentrate on a subclass of CPM, known as CPFSK. With the traditional description, CPFSK has a trellis structure where the trellis transition patterns are different for adjacent channel symbol intervals. Also, this trellis structure is not minimal, i.e, it can be represented as a trellis with fewer states.

In section 2.3, a decomposition approach to CPFSK is described, as in [3, 26, 5], which can be seen as a realization of the idea in [25].

In section 2.4, equivalent forms of the decomposed CPFSK model are described. We are interested in a useful subset of CPFSK which has a modulation index $h = K/P$ with P^{k_M} modulation levels, k_M an integer.

In section 2.5, the Euclidean distance properties of encoded CPFSK signals are discussed. A modified formula to calculate the Euclidean distance, which depends only on the input of the memoryless modulators, is derived.

Section 2.6 then gives some discussion on the preceding.

2.2 Traditional CPM Model and Its Subclass CPFSK

The CPM transmitted signal is a sequence of bandpass waveforms of the form

$$s(t,\alpha) = \sqrt{\frac{2E_s}{T_s}}\cos(2\pi f_0 t + \phi(t,\alpha)), t \geq 0 \tag{2.1}$$

where E_s being the energy transmitted per symbol time, T_s.

The name CPM comes from the information-carrying phase $\phi(t,\alpha)$ being a continuous function of time. This phase function is a function of the information sequence $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_i, \ldots)$, where we assume α_i to take values in the set $\{\pm 1, \pm 3, \ldots, \pm M-1\}$ if M is an even integer [22], and in the set $\{0, \pm 2, \pm 4, \ldots, \pm M-1\}$ if M is an odd integer [47]. The information-carrying phase has the following general expression:

$$\begin{aligned}\phi(t,\alpha) &= 2\pi h \int_0^t \sum_{i=0}^{n} \alpha_i g(\tau - iT_s)d\tau \\ &= 2\pi h \sum_{i=0}^{n} \alpha_i f(t - iT_s), \quad nT_s \leq t < (n+1)T_s,\end{aligned} \tag{2.2}$$

where the definition $f(t) \equiv \int_0^t g(\tau)d\tau$ has been introduced.

The function $g(t)$ is called the baseband frequency pulse. In general, we have

$$g(t) = 0, \quad \text{for } t < 0 \text{ and } t > LT_s; \tag{2.3}$$

where L is the *correlation length* of the pulse. Moreover, it satisfies the normalizing convention:

$$\int_0^\infty g(t)dt = \int_0^{LT_s} g(t)dt = \frac{1}{2}. \tag{2.4}$$

With this choice, the accumulated phase values due to completed $g(t)$ pulses are always multiples of $h\pi$. From (2.3) and (2.4), we have

$$\begin{aligned} f(t) &= 0, \quad t \leq 0, \\ f(t) &= \tfrac{1}{2}, \quad t \geq LT_s. \end{aligned} \tag{2.5}$$

The three parameters in (2.2) which can be varied to control the properties of a specific CPM signaling scheme are:

- the variable h, which is called the *modulation index*;
- the number of possible values of α_i, defined to be M;
- the shape and duration of the baseband frequency pulse $g(t)$, or equivalently of $f(t)$.

It is noted that the information carried by one input symbol can be stretched over more than one symbol interval in the phase trajectory. As a consequence, successive phase changes are correlated. These modulation schemes are also named *correlative phase modulations* for this reason.

We are interested in a subclass of CPM called *continuous phase frequency shift keying* (CPFSK). It is defined as CPM with the following characteristics:

$$\begin{aligned} L &= 1 \\ g(t) &= \frac{1}{2T_s}, \text{ for } 0 \le t \le T_s. \end{aligned} \tag{2.6}$$

Let us rewrite (2.2) for CPFSK

$$\begin{aligned} \phi(t,\alpha) &= 2\pi h \sum_{i=0}^{n} \alpha_i f(t - iT_s), \\ &= 2\pi h \sum_{i=0}^{n-1} \alpha_i f(t - iT_s) + 2\pi h \alpha_n f(t - nT_s), \\ &= \pi h \sum_{i=0}^{n-1} \alpha_i + 2\pi h \alpha_n f(t - nT_s), \\ &= \overline{\phi_n} + 2\pi h \alpha_n f(t - nT_s), \\ &\quad \text{for } nT_s \le t < (n+1)T_s, \end{aligned} \tag{2.7}$$

where the term $\overline{\phi_n}$ is a constant phase within the present symbol interval, and is given by

$$\overline{\phi_n} = [\pi h \sum_{i=0}^{n-1} \alpha_i] \text{ modulo } 2\pi. \tag{2.8}$$

The $\overline{\phi_n}$ is called the *phase state* at time nT_s. A restricted number of such phase states can be achieved only for rational values of the modulation index h. Let us define

$$h = K/P \tag{2.9}$$

with K and P being relatively prime integers. Recalling (2.8), and the set of values assumed by α_i, we have the following two cases:

1. The integer K is even. The possible distinct (mod 2π) values for the phase state $\overline{\phi_n}$ are

$$0, 2\pi\frac{K}{2P}, 2\pi\frac{2K}{2P}, \quad \ldots \quad 2\pi\frac{(P-1)K}{2P}, \tag{2.10}$$

   and the total number of phase states is equal to P.

2. The integer K is odd. Now the phase states are

$$0, 2\pi\frac{K}{2P}, 2\pi\frac{2K}{2P}, \quad \ldots \quad 2\pi\frac{(2P-1)K}{2P}, \tag{2.11}$$

   and the total number of phase states is $2P$.

Example 2.1: Assume binary ($M = 2$) CPFSK with $h = 1/2$ ($P = 2$) CPFSK. This modulator is also called minimum shift keying (MSK) or fast frequency-shift keying (FFSK) [44]. The phase states according to the above definition are:

$$\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}. \tag{2.12}$$

Recalling equation (2.7), the phase state at time nT_s, $\overline{\phi_n}$, can transfer to a phase state at time $(n+1)T_s$, $\overline{\phi_{n+1}}$, with the phase increment

$$2\pi h\alpha_n f(t - nT_s), \quad nT_s < t < (n+1)T_s \tag{2.13}$$

Therefore, there are M possible phase state transitions for each phase state. We can use a trellis diagram to describe the transition structure. The total number of different transitions are $2PM$. Figure 2.1 shows the trellis structure for MSK. In this diagram, each state is labeled with its phase state and each branch between shows the input data α_n which produces the transition. The dotted lines represent the input symbol -1, and the solid lines represent the input symbol 1. The S_1, S_2, S_3, S_4 are the phase states $0, \pi/2, \pi$, and $3\pi/2$ respectively.

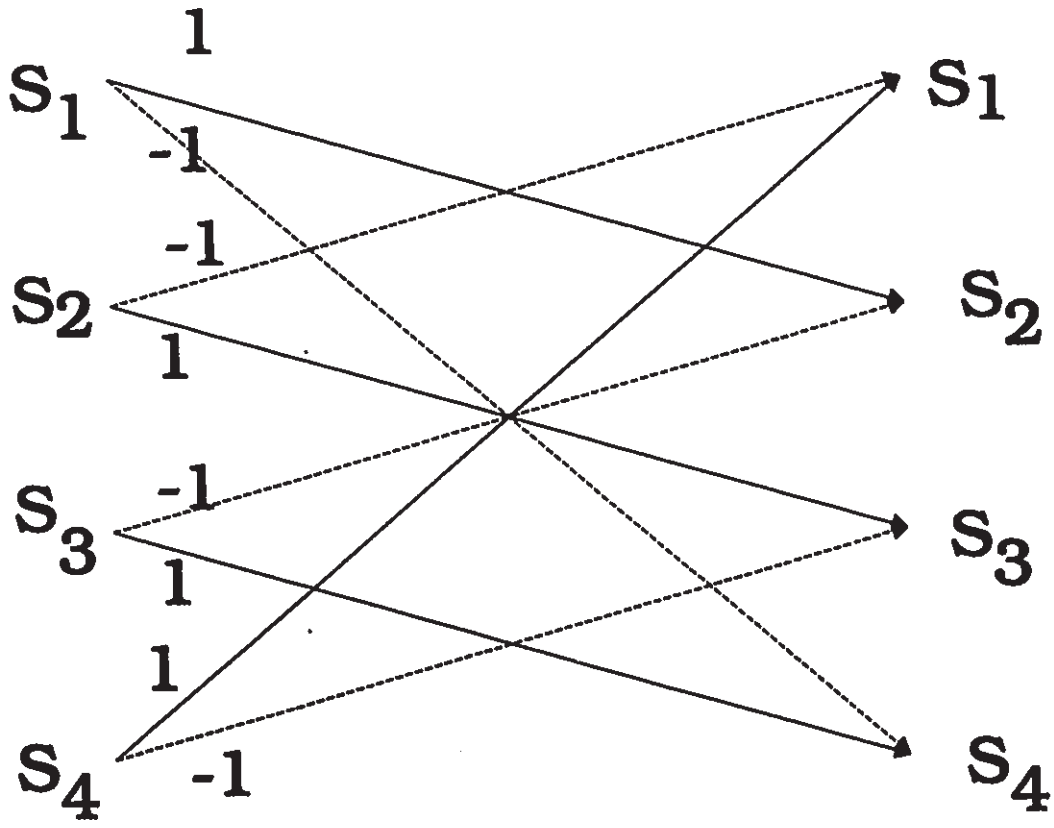


Figure 2.1: Conventional MSK trellis diagram

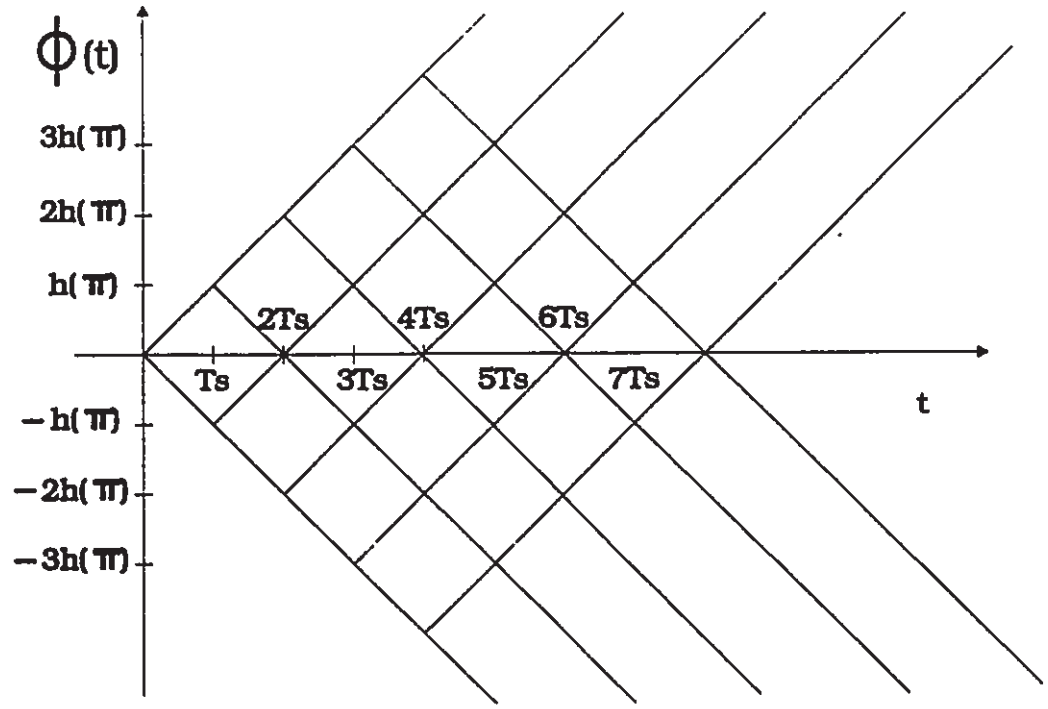


Figure 2.2: Phase tree for binary CPFSK

Another approach describing the phase transition structure is called the *phase tree*, which represents the phase variation with time. Refer to equation (2.7) which describes the real phase $\phi(t, \alpha)$. For binary CPFSK, the phase tree is shown in Figure 2.2. We use *pi* to represent π. If the input is -1, the tree goes downward; if the input is 1 the tree goes upward.

2.3 Decomposition Model of CPFSK

It is known that the conventional implementation of CPFSK is not minimal, i.e., the trellis representation for CPFSK can be reduced [26, 3]. Our goal is to introduce the concept that CPFSK can be decomposed as a time-invariant linear encoder combined with a memoryless modulator.

Let us consider the MSK trellis first. MSK has $M = 2$ modulation levels and modulation index $h = 1/2$. The phase tree for MSK is shown in Figure 2.3. Since

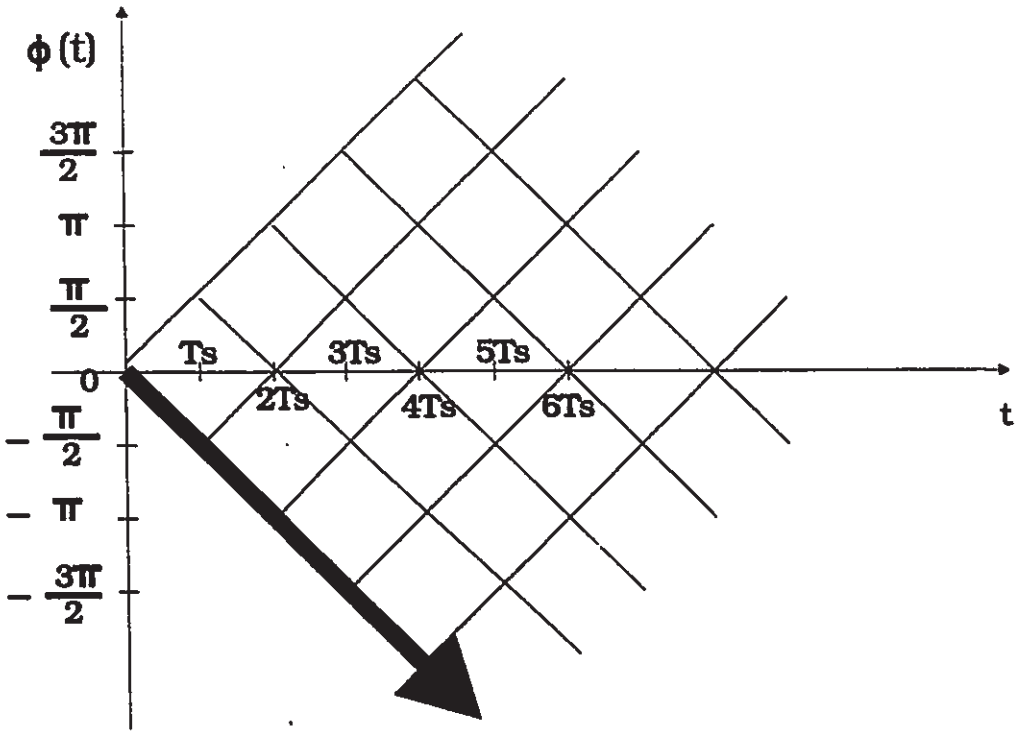


Figure 2.3: MSK phase tree.

phase differences that are an integer multiple of 2π are physically indistinguishable, the phase tree can be represented as a simple trellis structure, as shown in Figure 2.4.

This trellis is obtained by taking the real phase $\phi(t)$ in (2.2) modulo-2π, , i.e.,

$$\overline{\phi(t)} = R_{2\pi}[\phi(t)] \tag{2.14}$$

where $R_x[\cdot]$ denotes the "modulo x operator," defined by

$$R_x[y] = y - \lfloor \frac{y}{x} \rfloor x. \tag{2.15}$$

The operator $\lfloor \cdot \rfloor$ represents the largest integer not exceeding the enclosed number. As described in (2.11), the $\overline{\phi(t)}$ at $t = nT_s$ has four possibilities: $\{0, \pi/2, \pi, -\pi/2\}$. However, if we rewrite the MSK signal in the following form:

$$s(t, \alpha) = \sqrt{\frac{2E_s}{T_s}} \cos(2\pi f_1 t + \theta(t, \alpha)), \tag{2.16}$$

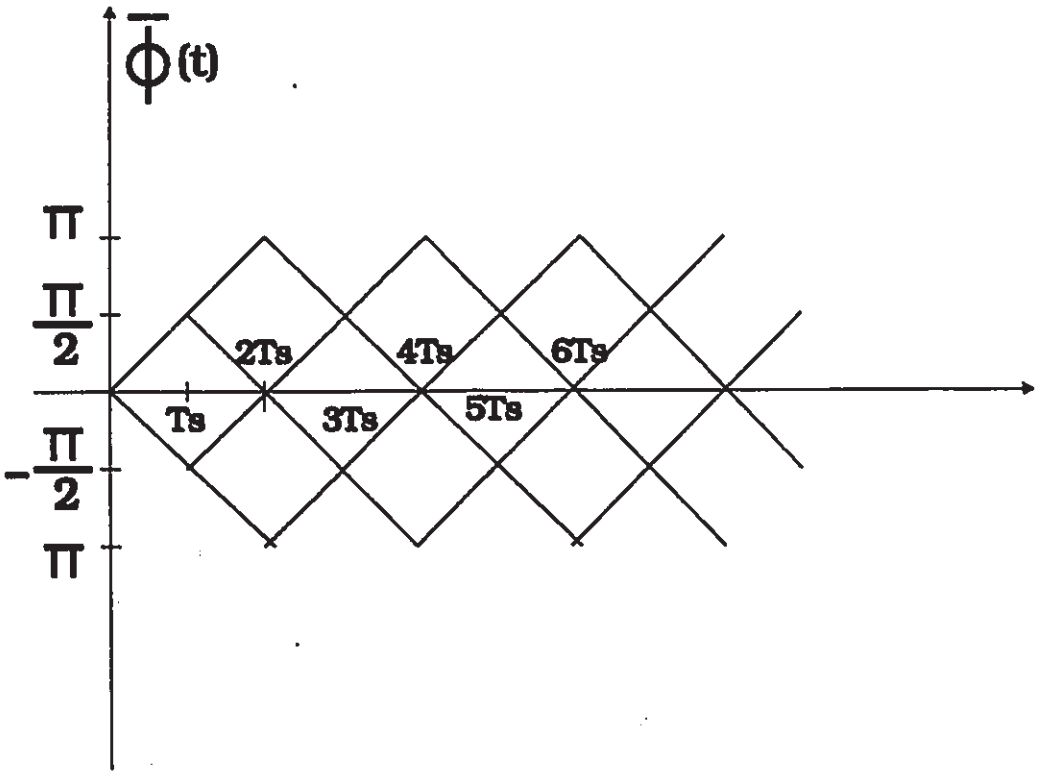


Figure 2.4: MSK physical trellis diagram.

where the asymmetrical carrier frequency is defined as

$$f_1 = f_c - \frac{1}{4T_s}, \tag{2.17}$$

and is equivalent to measuring the phase relative to the lowest phase trajectory in Figure 2.3 (the lower boldface line). This new phase, $\theta(t, \alpha)$, called the *tilted phase*, is defined by

$$\theta(t, \alpha) = \phi(t, \alpha) + \frac{\pi t}{2T_s} \tag{2.18}$$

In a similar way to (2.14), we can define the *tilted physical phase*, $\overline{\theta}$,

$$\overline{\theta} = R_{2\pi}[\theta] \tag{2.19}$$

The new phase tree and physical phase are depicted in Figure 2.5 and Figure 2.6 respectively. It is very interesting that this physical phase tree is a time-invariant trellis. This time-invariant form of MSK trellis was introduced in [48, 28]. It has only two tilted phases. In [3] it is shown that this time-invariant trellis can be implemented by a rate-1/2 convolutional encoder cascaded with a memoryless modulator.

Massey suggested that any CPM can be considered as a encoder cascaded with a memoryless modulator [25], as shown in Figure 2.7. The input information symbols are M-ary positive integers just like a traditional channel encoder cascaded with a memoryless modulator. An implementation of this idea was discovered by Rimoldi [26]. It is proved [26] that CPM, in general, can be decomposed as a linear encoder (modulo some integer P), called continuous phase encoder (CPE), cascaded with a memoryless modulator (MM). We will follow [26, 13] to give a simplified derivation for the decomposed model of CPFSK.

In order to represent CPFSK as a time-invariant trellis, the CPFSK signal can be generated by relating the phase to a asymmetrical carrier, f_1, which is defined as [26, 3, 5]

$$f_1 = f_c - (M-1)\frac{h}{2T_s}, \tag{2.20}$$

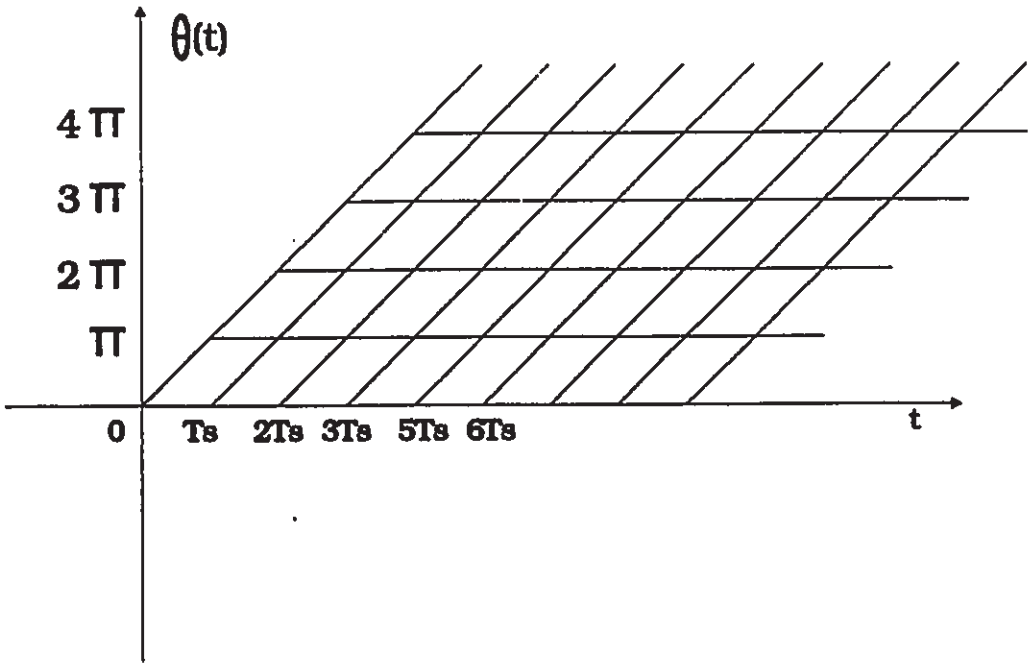


Figure 2.5: New phase tree related to the asymmetrical carrier.

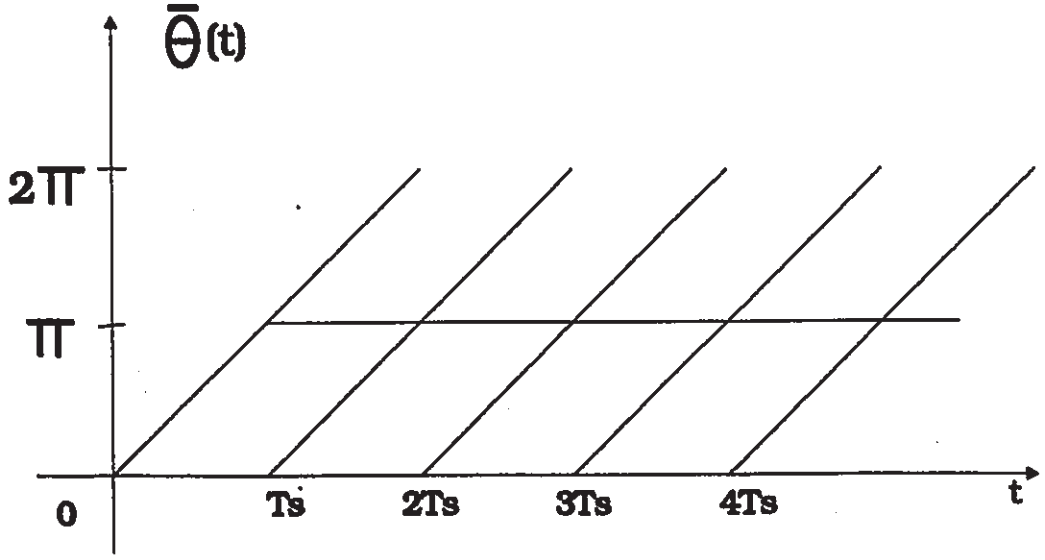


Figure 2.6: Two-state trellis for MSK.

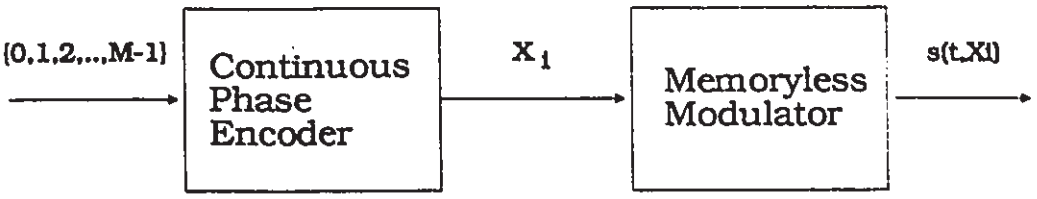


Figure 2.7: Massey's suggestion for a decomposition model of CPM.

to produce a time invariant trellis. The CPFSK signal can then be written as

$$s(t,\alpha) = \sqrt{\frac{2E_s}{T_s}}\cos(2\pi f_1 t + \phi(t,\alpha) + \pi h(M-1)t/T_s),\ t > 0. \tag{2.21}$$

In order to employ positive integers to represent the data sequence, α, we can define a *modified data sequence* $\mathbf{U} = (U_0, U_1, \ldots)$, where

$$U_i = \frac{\alpha_i + (M-1)}{2}. \tag{2.22}$$

By writing (2.2) in terms of f_1 and $\mathbf{U}$, the CPFSK signal can be represented as

$$s(t,\mathbf{U}) = \sqrt{\frac{2E_s}{T_s}}\cos(2\pi f_1 t + \theta(t,\mathbf{U})),\ t > 0, \tag{2.23}$$

where $\theta(t,\mathbf{U})$ is called the *tilted phase*. Let $t = \tau + nT_s$, $0 \le \tau < T_s$. We then obtain

$$\theta(\tau + nT_s, \mathbf{U}) = 2\pi h \sum_{i=0}^{n-1} U_i + 4\pi h U_n f(\tau + nT_s) + W(\tau), \tag{2.24}$$

where

$$W(\tau) = -\pi h(M-1)\tau/T_s + 2\pi h(M-1)f(\tau). \tag{2.25}$$

The *physical tilted phase* is, for $0 \le \tau < T_s$,

$$\overline{\theta}(\tau + nT_s, \mathbf{U}) = R_{2\pi}[2\pi h R_P[\sum_{i=0}^{n-1} U_i] + 4\pi h U_n f(\tau) + W(\tau)], \tag{2.26}$$

where we let $t = \tau + nT_s$. In the above equation we have used the fact that $R_{2\pi}[2\pi h x] = R_{2\pi}[2\pi h R_P[x]]$, when $h = K/P$. Note that all time-dependent terms on the right side of (2.26) depend only on the translated time variable $\tau = t - nT_s$. For

any symbol interval, the trellis transition is a function of the vector $X_n = [U_n, V_n]$, where the time-independent, data-dependent term,

$$V_n = R_P[\sum_{i=0}^{n-1} U_i], \tag{2.27}$$

has only P possible values. Also, all P values are possible when n satisfies

$$n(M-1) \geq P-1, \tag{2.28}$$

since this ensures that the sum of n M-ary digits can reach the largest required value $P-1$.

We have thus shown that the physical tilted-phase, $\overline{\theta}(\tau + nT, \mathbf{U})$, for CPFSK always has a time-invariant trellis. To complete the decomposition of a CPFSK modulator into a CPE and MM, we need only to determine the MM input and to find the recursion by which the CPE can update this MM input.

2.3.1 The Memoryless Modulator

For convenience, we write $\overline{\theta}(\tau, \mathbf{X_n})$ instead of $\overline{\theta}(\tau + nT_s, \mathbf{U})$, and $s(\tau, \mathbf{X_n})$ instead of $s(\tau + nT_s, \mathbf{U})$. The CPFSK signal is now denoted

$$s(\tau, X_n) = \sqrt{\frac{2E_s}{T_s}} \cos[2\pi f_1(\tau + nT_s) + \overline{\theta}(\tau, X_n)], \quad 0 \leq t < T. \tag{2.29}$$

For realization purposes, the above equation is decomposed into in-phase and quadrature components. We then obtain the set of equations that describe the operation of the memoryless modulator in a symbol interval,

$$\begin{aligned} s(\tau, X_n) &= I(\tau, X_n) \cdot \sqrt{2} \cdot \cos[2\pi f_1(\tau + nT_s)] + Q(\tau, X_n) \cdot \sqrt{2} \cdot \sin[2\pi f_1(\tau + nT_s)] \\ &= I(\tau, X_n) \cdot \Phi_I + Q(\tau, X_n) \cdot \Phi_Q \end{aligned} \tag{2.30}$$

where

$$\begin{aligned} I(\tau, X_n) &= \sqrt{\frac{E_s}{T_s}} \cos(\overline{\theta}(\tau, X_n)), \\ Q(\tau, X_n) &= \sqrt{\frac{E_s}{T_s}} \sin(\overline{\theta}(\tau, X_n)), \end{aligned} \tag{2.31}$$

Figure 2.8 shows the memoryless modulator in block diagram form.

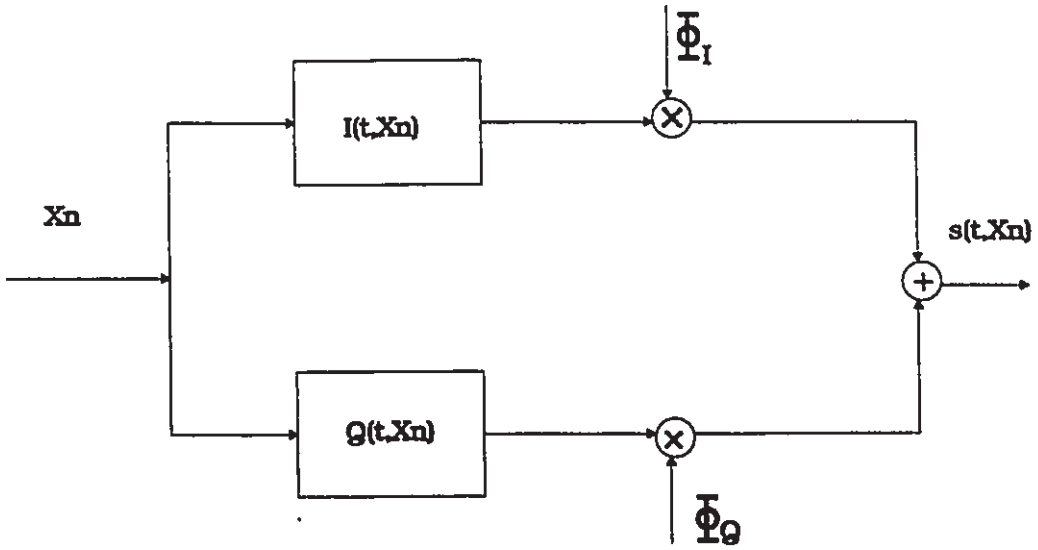


Figure 2.8: Memoryless modulator.

2.3.2 The Continuous Phase Encoder

The purpose of the CPE is to update the memoryless modulator input, X_n, to produce the next input, X_{n+1}. Making use of equation (2.27), and the fact that $R_x[y_1 + y_2] = R_x[R_x[y_1] + y_2]$, we have

$$\begin{aligned} X_{1,n+1} &= U_{n+1} \\ X_{2,n+1} &= R_P[\sum_{i=0}^{n} U_i] \\ &= R_P[\sum_{i=0}^{n-1} U_i + U_n] \\ &= R_P[R_P[\sum_{i=0}^{n-1} U_i] + U_n] \\ &= R_P[V_n + U_n] \end{aligned} \tag{2.32}$$

A realization of the CPE is shown in Figure 2.9, where the sum is taken modulo-P. A similar realization of the CPE can be found in [49].

We have shown that the transmitted signal (2.23) can be generated from a system composed of a CPE and an MM, as in Fig. 2.10 [26], where $X_{1,n} = U_n$, $X_{2,n} = V_n$ and the shift register is represented as a delay cell, D. $V_n = R_P[\sum_{i=0}^{n-1} U_i]$ is the *state*

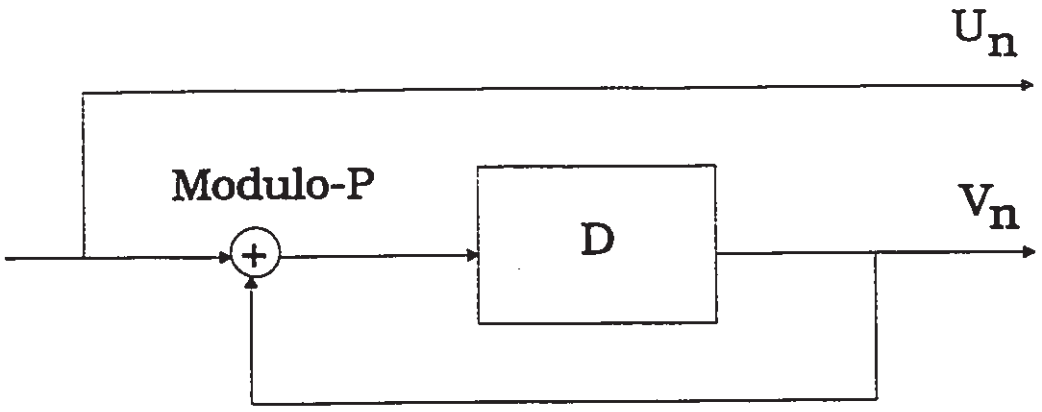


Figure 2.9: Continuous phase encoder

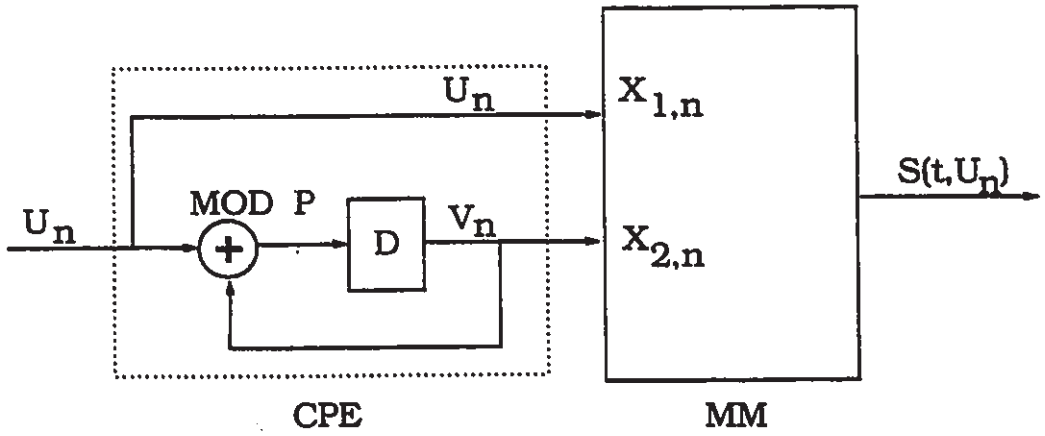


Figure 2.10: Decomposed CPFSK model.

of the encoder. The number of possible states at time nT_s is P, and the number of possible signals in the MM is MP.

In [4], it is pointed out that many of the coded M-ary CPM systems with $h = K/P$ and $M = P^{k_M}$, for some integer k_M, have maximum Euclidean distance. If U_n is represented in radix-P form, i.e.

$$U_n = U_n^1 P^{k_M-1} + U_n^2 P^{k_M-2} + \cdots + U_n^{k_M} = \sum_{j=1}^{k_M} U_n^j P^{k_M-j}, \tag{2.33}$$

then the resulting CPE has a well-defined structure, as shown in Fig. 2.11 [26]. Because of the modulo-P operation (see equation (2.27)), the state variable, V_n, is

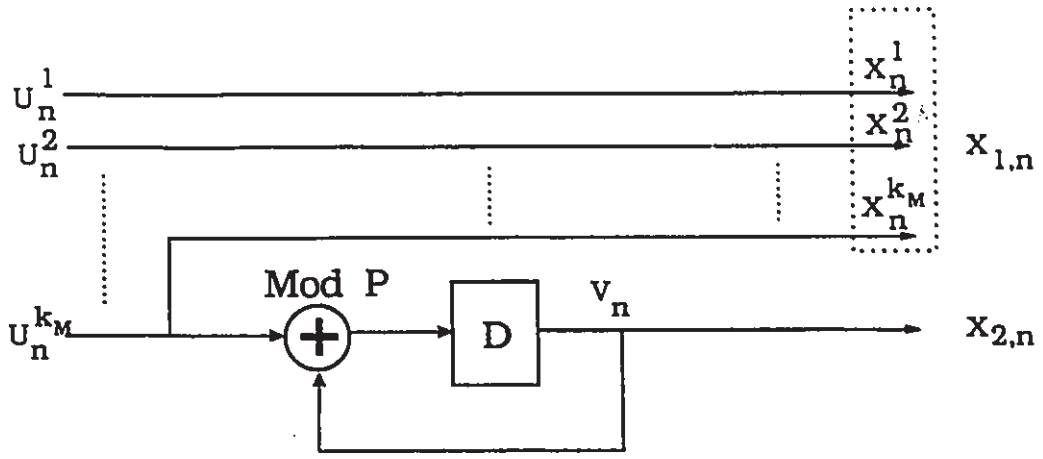


Figure 2.11: P^{k_M}-ary CPFSK with modulation index $h = K/P$.

represented by a single P-ary digit with the value

$$V_n = R_P[\sum_{i=0}^{n-1} U_i^{k_M}]. \tag{2.34}$$

It is noted that this form of CPE is a rate k_M/k_M+1 systematic convolutional encoder over the ring of integers modulo-P. It belongs to a new class of linear encoders recently developed by Massey and Mittelholzer [16] for phase modulation. We will discuss ring encoders in Chapter 4. The code generating matrix for the CPE can be denoted as C(D), where

$$\mathrm{C}(D) = \begin{bmatrix} 1 & \dots & 0 & 0 & 0 \\ \vdots & \ddots & \dots & \dots & \dots \\ 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{D}{1-D} \end{bmatrix}. \tag{2.35}$$

The relation between the outputs of the CPE and the inputs to the MM is given by

$$\begin{aligned} X_{1,n} &= \sum_{j=1}^{k_M} X_n^j P^{k_M-j} = \sum_{j=1}^{k_M} U_n^j P^{k_M-j} = U_n \\ X_{2,n} &= V_n \end{aligned} \tag{2.36}$$

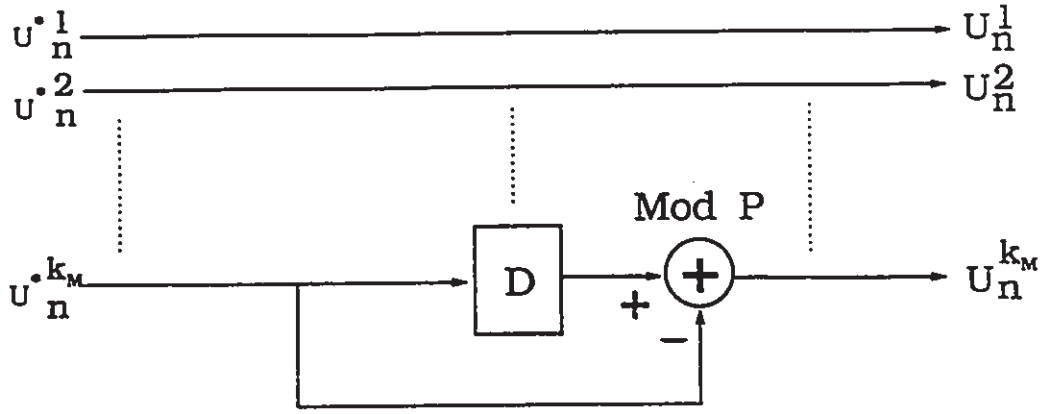


Figure 2.12: A scrambler for the CPE

2.4 Feedback-Free Decomposition Model of CPFSK

In the last section, it is shown that the CPE is a systematic encoder with feedback. Two different scramblers will be introduced to remove the feedback from the CPE. The reason for introducing a feedback-free model of CPFSK is because it is more convenient to cascade it with an external channel encoder compared to using the original model [3, 4]. Moreover, it has been found that the bit error probability of feedback free MSK is better than MSK [26]. This interesting fact will be shown in examples in Chapter 5.

A. Scrambler defined over the same code structure as the CPE.

In [26, 4], a precoder or scrambler is suggested to remove feedback in the CPE. Here we also define a precoder which has the same code structure (modulo-P) as the CPE. The reason for using such a precoder will become obvious after we specify our new convolutional coded CPFSK system in the following chapters. Consider a $k_M \times k_M$ precoder, as shown in Fig. 2.12, and defined by

$$\mathbf{T}(D) = \begin{bmatrix} \mathbf{I}_{k_M-1} & 0 \\ 0 & 1-D \end{bmatrix}. \tag{2.37}$$

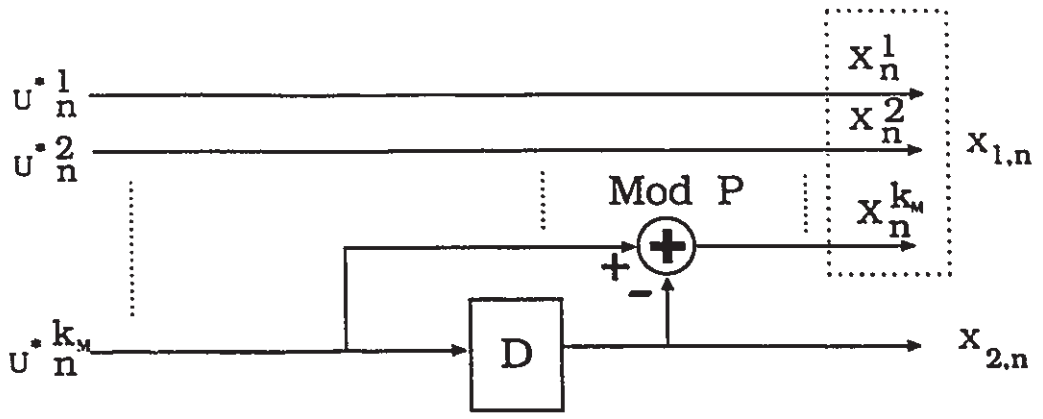


Figure 2.13: Feedback-free CPE.

$\mathbf{I}_{k_M-1}$ is an $(k_M-1)\times(k_M-1)$ identity matrix. This is cascaded with the encoder

$$\mathbf{C}(D)=\begin{bmatrix}\mathbf{I}_{k_M-1} & 0 & 0\\ 0 & 1 & \frac{D}{1-D}\end{bmatrix} \tag{2.38}$$

to produce an overall feedback-free and non-systematic encoder

$$\mathbf{C}'(D)=\mathbf{T}(D)\mathbf{C}(D), \tag{2.39}$$

as shown in Fig. 2.13.

B. Precoder defined over different code structure as the CPE

In [4], another feedback-free form of CPE has been suggested if the CPFSK satisfies the following condition:

$$M \text{ is an integer multiple of } P. \tag{2.40}$$

A scrambler, given in Figure 2.14, where

$$U_n = R_M[U_n^* - R_P[U_{n-1}^*]] \tag{2.41}$$

can be employed to remove the feedback of the CPE. Such a device is called a scrambler because it maps the list of all input sequences into a scrambled list of the same sequence [50]. Substituting U_n into (2.36), we obtain the feedback free form of CPFSK,

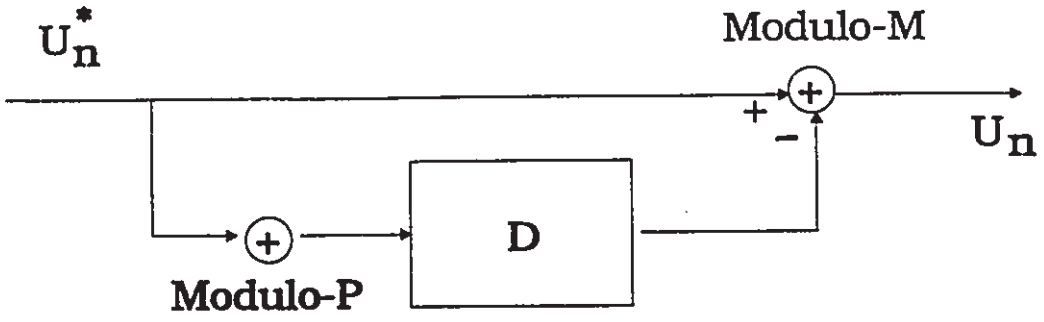


Figure 2.14: Rimoldi's scrambler for binary encoded CPFSK system.

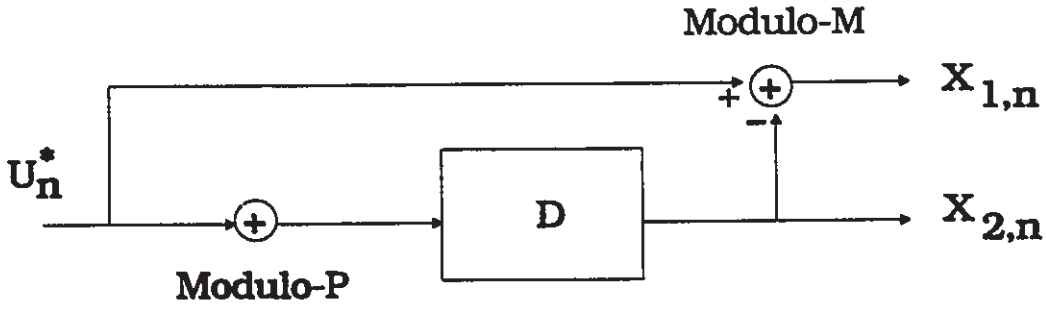


Figure 2.15: Feedback-free form of CPFSK using Rimoldi's scrambler.

where

$$\begin{aligned} X_{1,n} &= U_n = R_M[U_n^* - R_P[U_{n-1}^*]] \\ X_{2,n} &= V_n = R_P[U_{n-1}^*]. \end{aligned} \tag{2.42}$$

The feedback free encoder is shown in Figure 2.15.

If M is a power of 2, i.e. $M = 2^{k_M}$, and $P = 2^{k'_M}$, then the input data is $U_n^* = \sum_{i=1}^{k_M} U_n^{*i} 2^{k_M - i}$. U_n^* is the encoded data sequence. The encoder can be represented as a binary encoder, shown in Figure 2.16. It is noted that the overall encoder is not a linear binary convolutional encoder (except for MSK). For example, for $M = 4$ CPFSK with $h = 1/2$, the encoder is given as in Figure 2.17. The encoder is nonlinear (in binary sense) since it contains a modulo-4 operator.

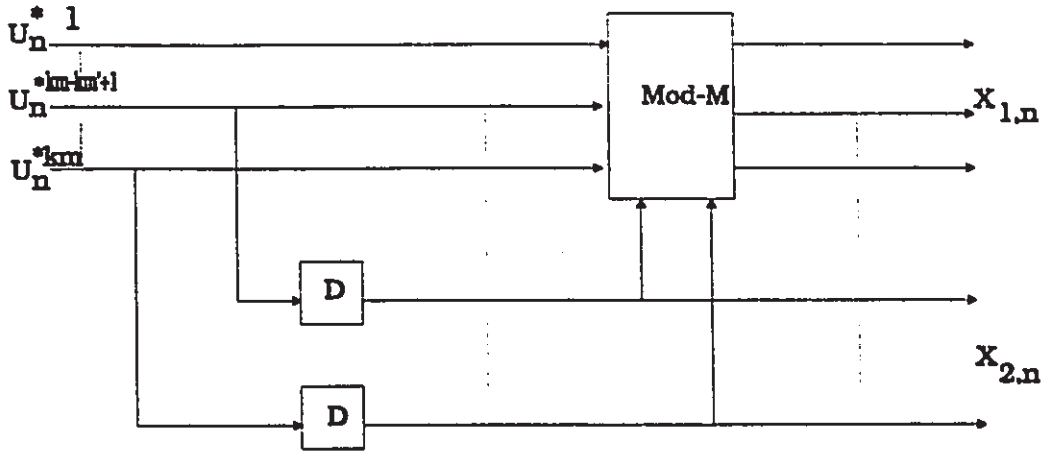


Figure 2.16: Binary equivalent CPFSK with Rimoldi's scrambler

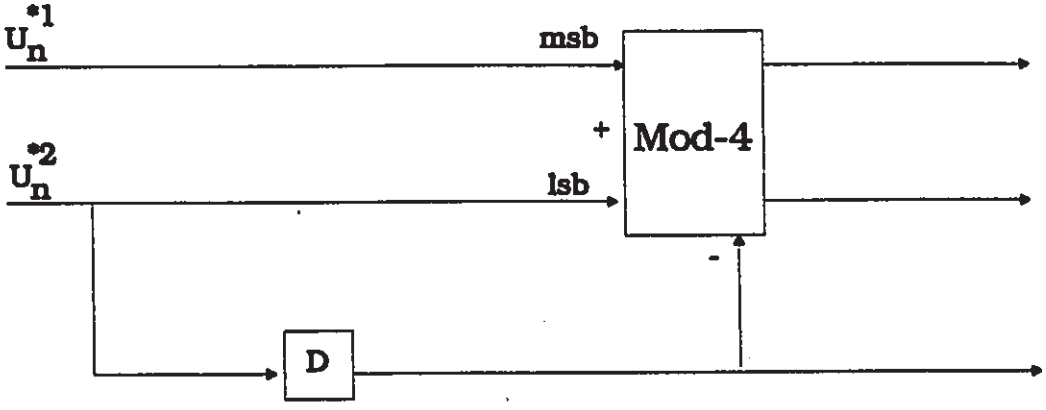


Figure 2.17: The feedback-free CPE for $h = 1/2$ quaternary CPFSK (the scrambler is nonlinear in binary sense.)

2.5 Euclidean Distance Properties of CPFSK Signals

In an AWGN environment, it is well known that the normalized minimum squared Euclidean distance (NMSED),

$$d^2_{min} = \frac{D^2_{min}}{2E_b}, \tag{2.43}$$

is a good figure of merit for the probability of error, P_e, of the coded or uncoded system [8, 1], i.e.

$$P_e \approx Q[(\frac{d^2_{min}E_b}{N_0})^{\frac{1}{2}}], \tag{2.44}$$

provided E_b/N_0 is not too small. $Q[\cdot]$ is the error function and N_0 is the single-sided power spectrum density of AWGN. In (2.43),

$$D^2_{min} = \min_{\mathbf{U}\neq\mathbf{U}'} D^2(\mathbf{U},\mathbf{U}') \tag{2.45}$$

is the minimum SED between any two different signal sequences generated by the coded system. $E_b = E_s/R$ is the transmitted energy per bit and R (bits/symbol) is the information rate of the coded or uncoded CPFSK system.

From [20, 51], it has been shown that the squared Euclidean distance (SED) between two CPFSK coded or uncoded symbol sequences is defined as

$$D^2(\mathbf{U},\mathbf{U}') = \int_{-\infty}^{\infty}[s(t,\mathbf{U}) - s(t,\mathbf{U}')]^2 dt, \tag{2.46}$$

where $s(t,\mathbf{U})$ and $s(t,\mathbf{U}')$ are the signals produced by two different data sequences $\mathbf{U}$ and $\mathbf{U}'$. Replacing $s(t,\mathbf{U})$ and $s(t,\mathbf{U}')$ with equation (2.23), and assuming these two symbol sequences start from the same state at $t = 0$ and merge to this state at $t = NT_s$, the above equation can be rewritten as

$$\begin{aligned} D^2(\mathbf{U},\mathbf{U}') &= \int_{-\infty}^{\infty}[s(t,\mathbf{U}) - s(t,\mathbf{U}')]^2 dt \\ &= \int_{-\infty}^{\infty} s^2(t,\mathbf{U})dt + \int_{-\infty}^{\infty} s^2(t,\mathbf{U}')dt \\ &\quad -2\int_{-\infty}^{\infty} s(t,\mathbf{U})\cdot s(t,\mathbf{U}')dt \end{aligned}$$

$$
\begin{aligned}
&= 2E_s(N+1) - 2\cdot\int_{-\infty}^{\infty} s(t,\mathbf{U})\cdot s(t,\mathbf{U}')dt \\
&= 2E_s(N+1) - 2\cdot\frac{2E_s}{T_s}\cdot\int_{-\infty}^{\infty}\cos(2\pi f_1 t+\theta(t,\mathbf{U}))\cos(2\pi f_1 t+\theta(t,\mathbf{U}'))dt \\
&= 2E_s(N+1) - 2\cdot\frac{E_s}{T_s}\cdot\int_{-\infty}^{\infty}\cos(4\pi f_1 t+\theta(t,\mathbf{U})+\theta(t,\mathbf{U}'))dt \\
&\quad -2\cdot\frac{E_s}{T_s}\cdot\int_{-\infty}^{\infty}\cos(\theta(t,\mathbf{U})-\theta(t,\mathbf{U}'))dt \qquad (2.47)
\end{aligned}
$$

Assuming $f_1 \gg 1/T_s$, the above equation reduces to

$$
\begin{aligned}
D^2(\mathbf{U},\mathbf{U}') &= 2E_s(N+1) - \frac{2E_s}{T_s}\int_{-\infty}^{\infty}\cos(\theta(t,\mathbf{U})-\theta(t,\mathbf{U}'))dt \\
&= \sum_{i=0}^{N} 2E_s[1-\frac{1}{T_s}\int_{iT_s}^{(i+1)T_s}\cos(\theta(t,\mathbf{U})-\theta(t,\mathbf{U}'))dt] \\
&= \sum_{i=0}^{N} D_i^2(\mathbf{U},\mathbf{U}'), \qquad (2.48)
\end{aligned}
$$

where

$$
D_i^2(\mathbf{U},\mathbf{U}') = 2E_s[1-\frac{1}{T_s}\int_{iT_s}^{(i+1)T_s}\cos(\theta(t,\mathbf{U})-\theta(t,\mathbf{U}'))dt]. \qquad (2.49)
$$

To further simplify the equation, we first note that the argument of the cosine function can be denoted for CPFSK for the $i=n$ interval as

$$
\theta(t,\mathbf{U})-\theta(t,\mathbf{U}') = 2\pi h\sum_{i=0}^{n-1}(U_i-U_i') + 4\pi h(U_n-U_n')\frac{t-nT_s}{2T_s}, \qquad (2.50)
$$

where $nT_s \le t < (n+1)T_s$. Let

$$
\epsilon_n \equiv U_n - U_n', \quad 0 \le n \le N. \qquad (2.51)
$$

Solving equation (2.49), we have

$$
D_n^2(\mathbf{U},\mathbf{U}') = \begin{cases} 2E_s[1-\frac{\sin(2\pi h\sum_{i=0}^{n}\epsilon_i)-\sin(2\pi h\sum_{i=0}^{n-1}\epsilon_i)}{2\pi h\epsilon_n}], & \epsilon_n \neq 0 \\ 2E_s[1-\cos(2\pi h\sum_{i=0}^{n-1}\epsilon_i)], & \epsilon_n = 0 \end{cases} \qquad (2.52)
$$

D_i^2 is called the incremental squared Euclidean distance (ISED) [4, 51]. For any real number ξ, and $h = K/P$, $\sin(2\pi h\xi) = \sin(2\pi h R_P[\xi])$ and $\cos(2\pi h\xi) = \cos(2\pi h R_P[\xi])$.

Because $V_n = R_P[\sum_{i=0}^{n-1} U_i]$, the above equation can also be written in terms of V_n and U_n as

$$D_n^2(\mathbf{U}, \mathbf{U}') = \begin{cases} 2E_s[1 - \frac{\sin 2\pi h(U_n+V_n-U_n'-V_n')-\sin 2\pi h(V_n-V_n')}{2\pi h(U_n-U_n')}], & U_n \neq U_n' \\ 2E_s[1 - \cos 2\pi h(V_n - V_n')], & U_n = U_n' \end{cases} \tag{2.53}$$

or in terms of $\mathbf{X} = [X_{1,n}, X_{2,n}]$ as

$$D_n^2(\mathbf{X}, \mathbf{X}') = \begin{cases} 2E_s[1 - \frac{\sin 2\pi h(X_{1,n}+X_{2,n}-X_{1,n}'-X_{2,n}')-\sin 2\pi h(X_{2,n}-X_{2,n}')}{2\pi h(X_{1,n}-X_{1,n}')}], & X_{1,n} \neq X_{1,n}' \\ 2E_s[1 - \cos 2\pi h(X_{2,n} - X_{2,n}')], & X_{1,n} = X_{1,n}' \end{cases} \tag{2.54}$$

Equation (2.54) specifies the relation between the output of the modulo-P CPE and the squared Euclidean distance of the outputs of the MM. When the cascaded CE and CPE are to be considered as one entity, we can find the transfer function of the overall encoder and then use this modified ISED formula to compute the SED.

Chapter 3

Binary Convolutional Coded CPFSK Systems

3.1 Introduction

This chapter studies previous binary convolutional coding techniques applied to continuous phase frequency shift keying (CPFSK). Previous work in trellis encoding of CPFSK includes the traditional approach [1, 2, 24], the matched encoding approach [3], the decomposition approach [4] and the double trellis encoding approach [5].

CPFSK is a modulation with *memory*. To cascade the external channel encoder with CPFSK we must consider the combination of the memory of the external channel encoder (CE) with the memory of the internal continuous phase encoder (CPE). It is important to note that the complexity of the receiver should depend on the combination of these two encoders and not depend on the complexity of the channel encoder alone. With this concept in mind, we will give comparisons between the codes found in [1, 2, 24], which are the *traditional approach*, and the other three approaches, which studied the interaction of the memory of the external channel encoder and the inner continuous phase encoder.

Since the decomposition model of CPFSK was found, new approaches have been

studied for the interaction between the code structure of the external channel encoders and the inner continuous phase encoder. One of these approaches is called the *matched encoding* approach [3] and the other is called the *decomposition approach* [4]. Coding gains with these new approaches are no more than 0.6 dB better than the traditional approach with the same complexity in the MLSE receiver. A different approach, called *double-trellis coded* CPM [5, 30], applies the multiple trellis coding technique [14] to CPFSK, and has improvement over some of the other approaches for certain cases. Either the traditional approach or the above approaches consider only a binary convolutional encoder cascaded with M-ary CPFSK. In previous work [6], it is found that there is no single approach which can be outperformed by the others for all cases. Although the double trellis encoding approach can achieve better coding gain in some examples of CPFSK [30], however, whether it can outperform in all cases is not known and will be discussed in this chapter.

In section 3.2, a survey of previous work in trellis encoded CPFSK is presented. The major concepts and systems models of these approaches are described and their performances are compared for the achievable maximum normalized squared Euclidean distance for the same complexity in the MLSE receiver.

In section 3.3, the double trellis encoding approach is studied. We use this approach to design trellis codes for quaternary CPFSK with modulation index 1/2. The codes so obtained are worse than those found in the traditional approach.

In the 3.4, comparisons between the best codes found for the different approaches are given. It is concluded that no single previous approach achieves consistently good codes if only binary convolutional codes are considered.

3.2 Previous Binary Coding Techniques

In previous work [1, 3, 4], a binary convolutional channel encoder (CE), **G**, is connected to the M-ary CPFSK with a binary to M-level mapper, **Q**, as shown in Figure

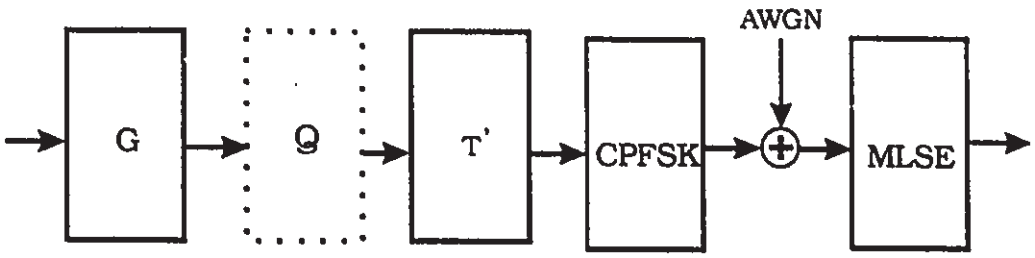


Figure 3.1: System model for binary convolutional encoded CPFSK.

3.1. Alternatively, to avoid the mapper **Q** and to combine the binary CE and the CPE in a natural way [4, 27], a scrambler $\mathbf{T}'$ was cascaded with the CPFSK model to allow a binary feedback-free CPE. Three approaches are introduced, namely the traditional approach and the matched encoding approach, which have a mapper between the channel encoder and the CPE, and the decomposition approach, which employs a scrambler to produce a binary equivalent CPE.

3.2.1 Binary Convolutional Coded CPFSK System Model with Signal Mapper

Figure 3.1 shows a convolutional encoder cascaded with M-level CPFSK (M is a power of two). In Figure 3.1, **G** is the generator matrix of a rate k/l binary convolutional encoder. **G** has entries $G_{ij}(D)$ $(i = 1, 2, \ldots, k; j = 1, 2, \ldots, l)$, the generator polynomials of the code. At time t, the input vector **a** is a k-tuple represented by $(a^1(t), a^2(t), \ldots, a^k(t))$, which produces a coded l-tuple, $\mathbf{b} = (b^1(t), \ldots, b^l(t))$, where $\mathbf{b} = \mathbf{aG}$. The constraint length (it is defined in [50] or can be found in Appendix B) of this encoder is defined as $\nu = \sum_i \max_j(deg\{G_{ij}(D)\})$. The number of states in the channel encoder, S_G, is defined as

$$S_G = 2^{\nu} \tag{3.1}$$

In order to cascade a rate-k/l binary convolutional encoder with M-ary CPFSK, a mapper **Q** is introduced between the encoder and the modulator. Here, we denote the input symbols of CPFSK as positive integers. Even though in [22] the input

Table 3.1: An example of a quaternary mapper

MSB	LSB	Output
b^1	b^2	U
0	0	0
0	1	1
1	0	2
1	1	3

symbols of CPFSK are not represented as positive integers, these two representations of CPFSK are equivalent (see Chapter 2). In order to simplify the description, we use only positive input symbols. For binary CPFSK, **Q** is a mapper which transforms the l-tuple **b** sequentially into a binary sequence $\mathbf{U} = \{U_0, U_1, \ldots, U_i, \ldots\}$, where $U_i \in \{0, 1\}$. For M-level CPFSK, **Q** is a mapper which transforms **b** into an M-level pulse-amplitude modulated signal **U**, where $\mathrm{M} = 2^l$ and $U_i \in \{0, 1, 2, \ldots, M-1\}$. A block of coded symbols of length l enters the mapper, which outputs M-ary symbols to CPFSK. One channel symbol is then transmitted. The mapping rule we use here is the so called natural mapping rule , i.e.

$$U_i(t) = \sum_{i=1}^{l} b^i(t) 2^{l-i}. \tag{3.2}$$

Table 3.1 shows the mapping between input binary data and output M-ary symbol, where we assume $l = 2$ and $M = 4$.

The modulator uses M-ary CPFSK with $h = K/P$, which can be considered as a CPE and a MM. The number of states, S_M, in the time-invariant trellis of CPFSK is the contents of the CPE, i.e.,

$$S_M = P \tag{3.3}$$

For example: when $h = 1/4$ and $M = 4$, the *state*, σ_M, of the CPE has four possibilities, i.e. $\sigma_M \in \{0, 1, 2, 3\}$. For every M-ary symbol entering the modulator, a CPFSK signal is transmitted. The channel environment, $n(t)$, is AWGN, and the

decoder, V, is an MLSE receiver using the Viterbi algorithm (refer to Appendix B). The complexity of the receiver is S_V.

Since both the channel encoder and the CPE of CPFSK are finite-state machines, the combined state $\sigma_V = [\sigma_G, \sigma_M]$, σ_G and σ_M are the states in the CE and the CPE respectively, should have a complexity, S_V, equal to or less than the multiplication of S_G and S_M [3, 52], i.e.,

$$S_V \leq S_G \cdot S_M \tag{3.4}$$

Two examples of rate-1/2 convolutional encoders combined with MSK ($h = 1/2, M = 2$) are given to show that changes in the structure of the channel encoder will affect the complexity in the receiver.

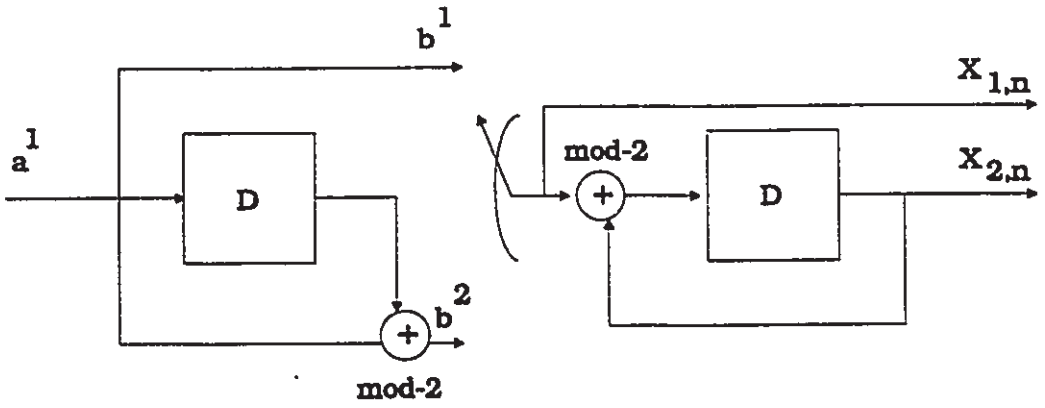
Example 2.1: Consider the encoder $\mathbf{G}(D) = [1, 1 + D]$ connected with MSK. The CE and CPE are shown in Figure 3.2 (a). Each encoder contains only one delay cell, i.e., $S_G = 2$ and $S_M = 2$. The combined trellis is shown in Figure 3.2(b). A combined state is denoted by a 2-tuple vector, $[\sigma_G, \sigma_M]$. Each branch of the combined trellis is labeled with the input data. There are four states in the combined trellis, i.e., $S_V = S_G \cdot S_M$.

Example 2.2: Let us consider another encoder, $\mathbf{G} = [1, D]$, connected with MSK. This encoder also has $S_G = 2$ and $S_M = 2$, but the combined trellis is only $S_V = 2$. The system model and combined trellis diagram are shown in Figure 3.3, i.e., $S_V = 1/2 S_G S_M$.

Now we are ready to describe the difference between the traditional approach and the matched encoding approach.

Traditional Approach

This approach obtains codes with maximum d^2_{min} by fixing the CPFSK modulation, and then searching the convolutional encoders with the same constraint length for the same code rate. That is, this approach searches codes with a encoder with $S_G = X$ states and a modulator with $S_M = Y$ states, and assumes the codes so obtained are the



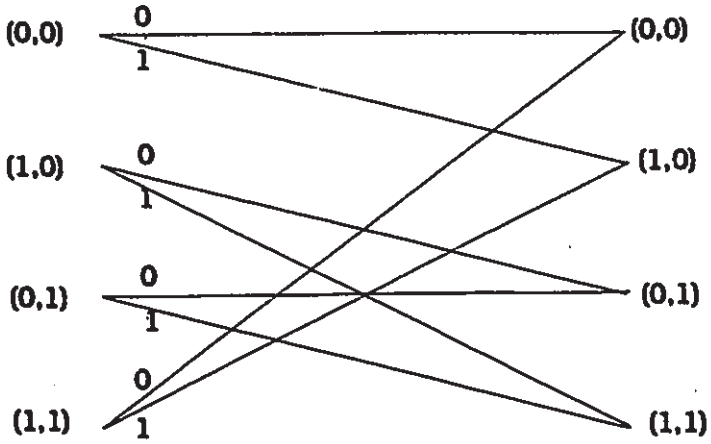
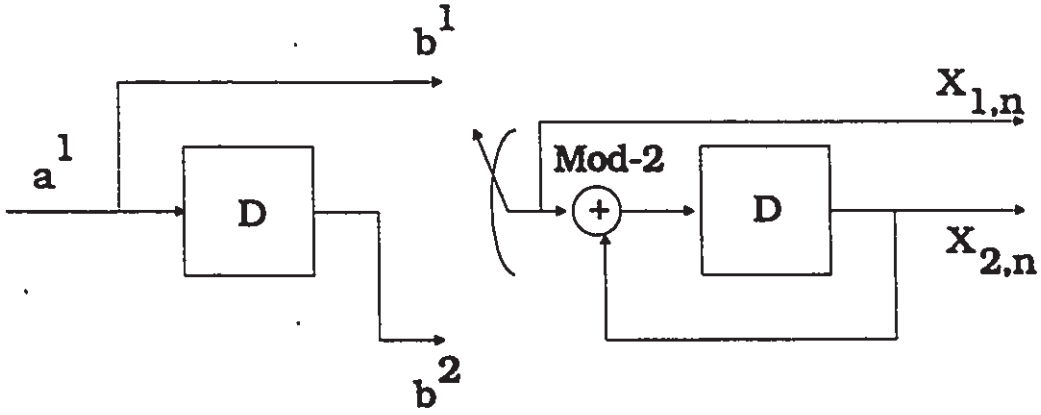


Figure 3.2: $G(D) = [1, 1 + D]$ cascaded with MSK.



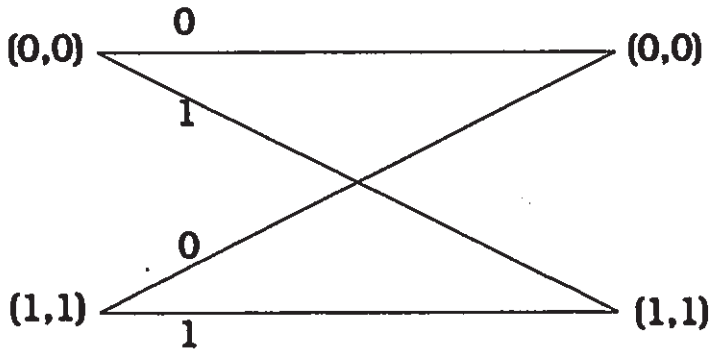


Figure 3.3: $G(D) = [1, D]$ cascaded with MSK.

best codes for $S_V = XY$. Since the complexity of the MLSE receiver is determined by the complexity of the overall encoder and not by the complexity of the convolutional encoders alone, there may exist convolutional codes with higher $S_G = Z, Z > X$, when combined with the same CPFSK, $S_M = Y$, to produce a receiver complexity $S_V = XY$. If this encoder has a larger d^2_{min}, then we will not find this code since the traditional approach does not consider the state reduction characteristics of the CE/CPE pairs. Hence, the codes so obtained with the traditional approach may not achieve the best d^2_{min} for a given number of states in the receiver.

The Matched Encoding Approach

It is found that when combining the CE with the CPE [3], some states in the CE may be coincident with memory in the CPE. The total number of states in the overall encoder can be reduced compared to the traditional approach (as shown in example 2.2).

For the same number of states in the MLSE, the matched encoding approach suggests that the best codes can be found by only searching the encoders that are *matched*. Here, *matched* encoders are the subset of available encoders which have a highest S_G for a given S_V in the MLSE receiver. In other words, if there are two subsets of encoders which, when connected with a modulator, produce the same fixed S_V in the receiver, then we need only to search the subset of encoders with a higher S_G.

In [6], we showed that this approach is at least as good as the traditional approach. However, the codes obtained are not optimal for equivalent forms of CPFSK [6].

3.2.2 Binary Convolutional Encoded CPFSK Model Employing Decomposed CPFSK

This model is suggested by Rimoldi [26, 27, 4], where a CPM system is divided into two parts : a subsystem with memory and a component without. The subsystem with

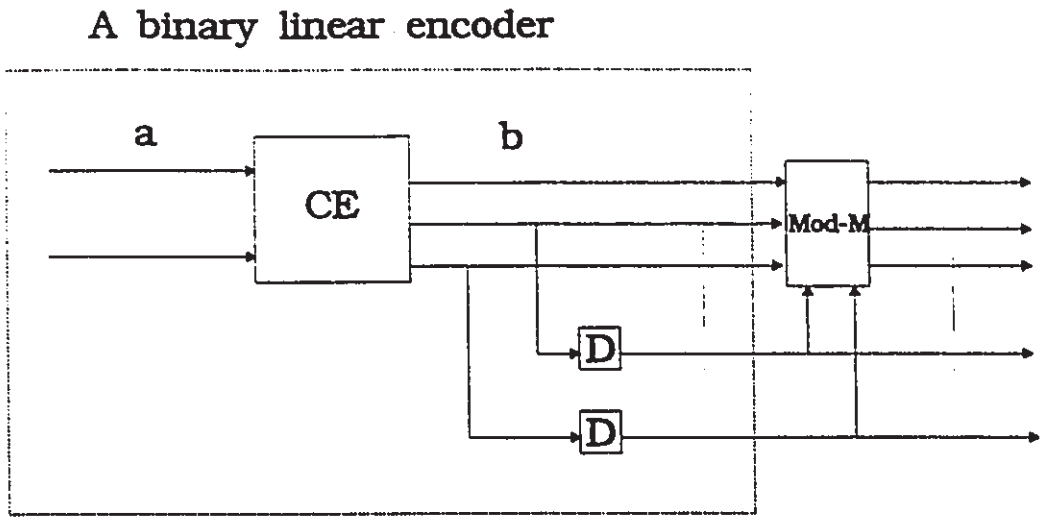


Figure 3.4: Binary encoded decomposed CPFSK system.

memory is a linear encoder and the memoryless component is a signal mapper. Figure 3.4 illustrates the convolutional encoded M-ary CPFSK model employed in Rimoldi's papers [4, 27]. It is shown that if the modulation index satisfies $h = J/M$, where $J = 1$ or 2, and the code rate equals k/l, where $l = log_2 M$, then we can combine the convolutional encoder with the modulator without a mapper and all notation is represented in binary (this subset of CPFSK has a better cutoff rate [4]).

In contrast to the matched encoding approach, the decomposition approach combines CE with the binary feedback free model of CPE in a natural way, i.e. both the CE and the CPE are binary encoders (see the dotted box in Figure 3.4). It is found that one of the delay cells can be removed if we use the set partitioning concept [11] for the feedback-free model. This approach is actually similar to the matched encoding approach in the sense of reducing the same number of states compared to the traditional approach. However, in our previous work [6], we showed that this approach does not always achieve the best performance for all examples.

Even though the dotted box in Figure 3.4 is a binary linear encoder, the overall encoder includes a modulo-M operator that will produce an overall encoder which is a *nonlinear* binary encoder. Actually, the code structure of the CPE is often a

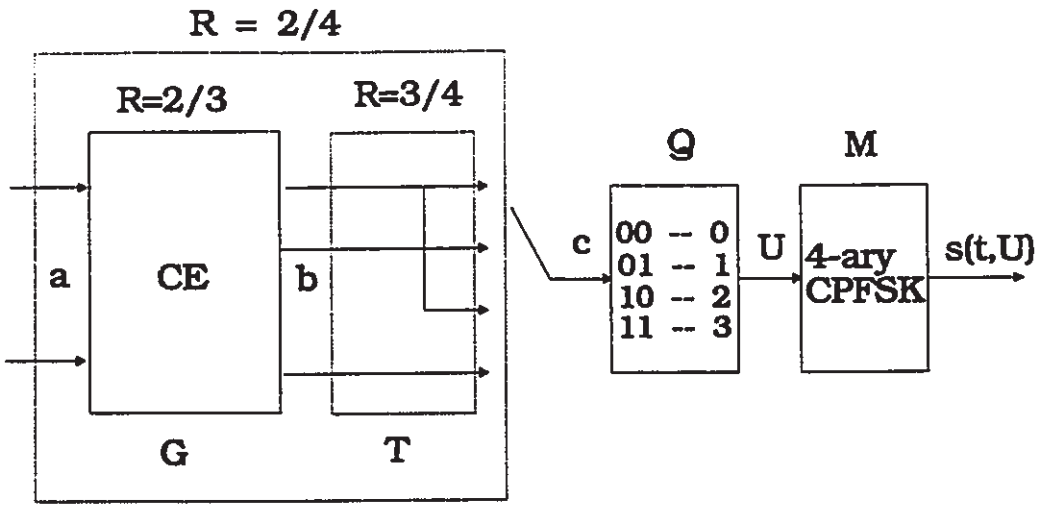


Figure 3.5: System model of double trellis encoded CPM.

ring convolutional encoder. Using an external encoder with the same code structure as the inner continuous phase encoder is called *natural* in [26]. Rimoldi derived this decomposition model but employed a *non-natural* combination of coding and modulation in [4].

3.3 Double Trellis Encoding Approach for CPFSK

This approach applies the *multiple trellis coding technique* [14] to CPFSK. The multiple trellis coding technique expands code rate from k/l to $\kappa k/(\kappa l)$, where κ is an integer. In [5, 30], a rate-2/4 convolutional encoder (κ equals 2) connects with 4-ary CPFSK. The system model is shown in Figure 3.5.

The channel encoder **G** is a rate-2/3 binary convolutional encoder with input sequence $\mathbf{a}(D)$ and output sequence $\mathbf{b}(D)$. The convolutional encoder is followed by a rate-3/4 linear block encoder **T** with output $\mathbf{c}(D)$. The combination of **G** and **T** is a rate-2/4 binary convolutional encoder. Again, **Q** is a natural mapper which maps the binary sequence into 4-ary symbols. The modulator **M** is CPFSK. The receiver is a MLSE receiver and the channel environment is AWGN.

The rate-3/4 block encoder **T** has an input/output relation as shown in Table

Table 3.2: Rate-3/4 mapper for Double Trellis Coded CPFSK

b	(c_n^1, c_n^2) (lsb1,msb1)	(c_n^3, c_n^4) (lsb2,msb2)	U_{2n}, U_{2n+1}
0 0 0	0 0	0 0	0 0
0 0 1	0 0	0 1	0 2
0 1 0	0 1	0 0	2 0
0 1 1	0 1	0 1	2 2
1 0 0	1 0	1 0	1 1
1 0 1	1 0	1 1	1 3
1 1 0	1 1	1 0	3 1
1 1 1	1 1	1 1	3 3

3.2. Two binary information symbols produce three binary encoded symbols from the rate-2/3 convolutional encoder, and these three encoded binary symbols are sent to the rate-3/4 block encoder **T** to produce four binary symbols. The four binary encoded symbols are then mapped to two quaternary symbols, U_{2n} and U_{2n+1}, with the mapper **Q**. Each combined trellis interval, n, consists of two channel symbols generated by U_{2n} and U_{2n+1}.

Table 3.2 lists all possible binary inputs to **T** and the related output quaternary symbols, U_{2n} and U_{n+1}. It is noted that the addition of U_{2n} and U_{2n+1} is always an even number weight. Therefore, all coded sequences $\mathbf{U} = (U_0, U_1, \ldots, U_n, \ldots)$ generated by **G** are an even number in weight. From [6, 53], it has been shown that when a rate-k/l channel encoder, which has an all-even-weight codewords property, is cascaded with an M-ary CPFSK, where M is a power of two and $h = K/P$, P is even, $M \geq 4$, then the number of states in the combined trellis will reduce to one half of the traditional combined trellis, i.e. $S_V = (1/2)S_G S_M$. Hence, this approach has a state reduction characteristic. This is again similar to the decomposition approach and the matched encoding approach.

The double trellis encoding approach can be compared to rate-1/2 binary convolutional encoded CPFSK systems, as explained in [5, 30]. The criterion used to

Table 3.3: Normalized minimum squared Euclidean distance for rate-1/2 binary convolutional encoded $h = 1/4$ quaternary CPFSK

S_V	(a) [30]	(b) [6]	(c) [4]	(d) [1]
4	2.73	2.42	2.15	2.00
8	4.00	3.00	3.57	3.00
16	4.88	4.42	4.51	4.30
32	5.45*	5.24	5.82	5.24
64	N.A.	6.15	6.18	6.15
128	N.A.	N.A.	7.09	N.A.
256	N.A.	N.A.	7.24	N.A.

*: search not complete, N.A.: not available, (a)double trellis encoding (b)The matched encoding approach, (c)The decomposition approach, (d)The traditional approach

compare the double trellis encoding approach with the other three approaches is the maximum d^2_{min} which can be achieved among the different approaches, for the same number of states in the receiver [5]. In [30], double trellis encoding approach provides better coding gain than rate-1/2 binary convolutional encoded, $h = 1/4$, quaternary CPFSK, for relatively simple complexity in the receiver, i.e. for S_V from 4 to 16. For higher complexity in the receiver, it becomes very hard to find the best codes with this approach, namely $S_V > 32$. The best codes found in [49] for $S_V = 64$ are worse compared to using the decomposition approach. A comparison of all the rate-1/2 binary convolutional encoded $h = 1/4$ quaternary CPFSK systems found to date is shown in Table 3.3.

However, in Table 3.4 we show the simulation results for a double trellis encoded quaternary CPFSK with modulation index $h = 1/2$. The convolutional encoder we considered is a rate-2/3 systematic convolutional encoder. Definitions for the canonical form of the rate-2/3 encoder can be found in the Appendix B. The rational function of the systematic encoder is represented in binary form, i.e. $H(D) = 1 + D + D^2$ is represented as 111. We compare this approach with the traditional approach.

Table 3.4: Normalized minimum squared Euclidean distance for rate-1/2 binary convolutional encoded quaternary CPFSK with $h = 1/2$ (double trellis encoding)

S_V	$H^0(D)$ $H^1(D)$ $H^2(D)$	d^2_{min}	d^2_{min} in [1]
2	1 1 1 0 0 0	2	2
4	1 1 1 1 0 1 1 0 0	4	3
8	1 0 0 1 1 1 0 0 1 0 0 0	4	5
16	1 0 0 1 1 0 1 0 1 0 1 0 0 0 0	5	6
32	1 0 1 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0	6	7

It is very interesting to note that it is no longer superior to the traditional approach when the number of combined states is greater than 4. It should be mentioned that in order to reduce the code searching complexity, this approach actually considers a subset of rate-2/4 binary convolutional encoders, i.e., it is actually a rate-2/3 binary convolutional encoder combined with a rate-3/4 block encoder.

3.4 Conclusions

3.4.1 A Comparison of Previous Binary Encoding CPFSK Results

In Table 3.3 and Table 3.5, we summarize all previous work for rate-1/2 convolutional encoded quaternary CPFSK with modulation indices $h = 1/4$ and $h = 1/2$.

Table 3.5: Normalized minimum squared Euclidean distance for rate-1/2 binary convolutional encoded $h = 1/2$ quaternary CPFSK (previous work)

S_V	(a) [30]	(b) [6]	(c) [4]	(d) [1]
4	2	2	2	2
8	4	4	3	3
16	4	5	4	5
32	5	6	5	6
64	6	7	6	7
128	N.A.	8	N.A.	N.A.

*: search not complete; N.A.: not available, (a)double trellis encoding (b)The matched encoding approach, (c)The decomposition approach, (d)The traditional approach

From these two tables, we see that no previous binary encoding structure consistently outperforms the others. This extends our previous results in [6]. The next chapter gives a solution to this problem where we obtain consistently good results compared to previous coding schemes.

3.4.2 Discussion

We have studied previous work on binary convolutional encoding CPFSK. The characteristics of each approach that produce improvement over the traditional approach can now be summarized as follows:

- The matched encoding approach suggests that better coding gains are possible if we use the encoder to produce a combined trellis that has less states than the traditional approach. The greater the number of states that can be reduced by the channel encoder, the greater the coding gain that is possible.

- The decomposition approach gives a *natural* combination of the CE and the CPE in the binary sense. It seems that we can use a coding scheme that is

more *natural* than using binary convolutional encoders, as suggested by the structure of the CPE.

- The double trellis encoding approach suggests that a quaternary encoder may be helpful. That is because considering a rate-2/4 binary encoder results in better coding gains for some cases.

Chapter 4

Trellis Coded CPFSK with Ring Convolutional Codes

4.1 Introduction

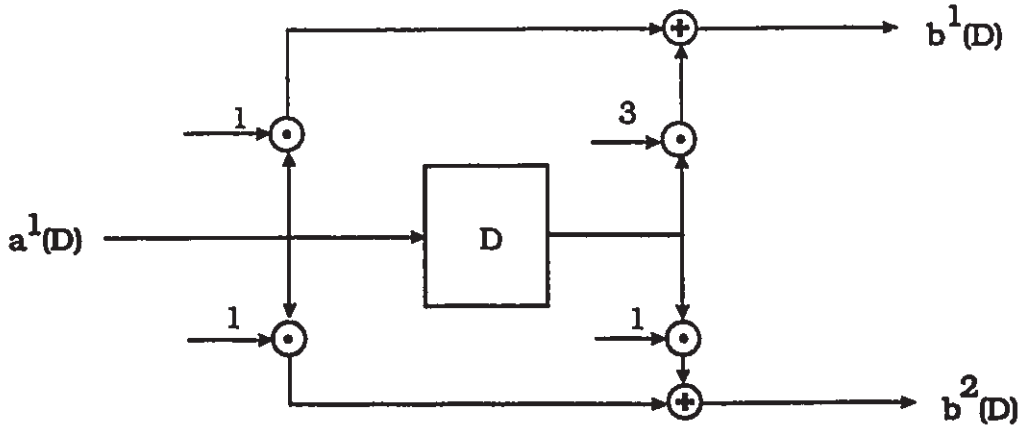
Several trellis coding techniques have been applied to phase modulation [11]. Most of the channel encoders for trellis encoding are binary convolutional codes on the Galois field GF(2). However, the work of [32, 34, 36, 33, 35] suggests that using convolutional codes over the ring of integers modulo-P has advantages compared to the binary-field case. This is because the combination of ring encoders and phase modulation generates an overall linear system. The extension of this work has been generalized by Forney [17]. Surprisingly, many famous TCM schemes are actually linear codes defined over groups or rings. However, in the CPM literature, the cascading of a ring encoder with CPM has not been examined. We are motivated to investigate this by the fact that the code structure of CPM is a ring convolutional encoder and to employ a modulo-P ring encoder to perform trellis encoding of the CPFSK. It is *natural* to consider a modulo-P encoder because it results in an encoder in the same regime as the CPE. We find that this encoding scheme has many characteristics that cannot be achieved by traditional binary convolutional encoded schemes. Numerical results

show that this approach leads to large coding gains. One example where we compare this approach with MSK, shows that we can achieve coding gains of 2 to 6 dB with $S_V = 4$ to $S_V = 256$ for the same bit rate per bandwidth (bit/sec/Hz).

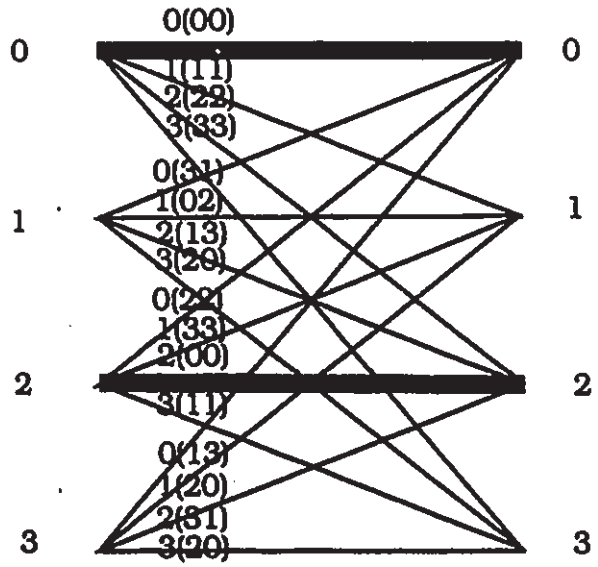
4.2 Ring Convolutional Codes

Let $R = \mathbf{Z}_P$ denote a commutative ring [16, 36, 54] where $\mathbf{Z}_P$ represents the ring of integers modulo-P. $R(D)$ is the ring of rational functions over R, whose numerators are polynomials with coefficients in $\mathbf{Z}_P$, as are the denominators with the restriction that their trailing coefficient is 1. A rate k/l convolutional code C over R is a free rank-k submodule of the free module $R(D)^l$. Every rate k/l convolutional code over R can be generated by some encoding matrix $\mathbf{G}(D) = g_{ij}(D)$, $1 \leq i \leq k$ and $1 \leq j \leq l$, $g_{ij}(D) \in R(D)$ with kernel, $\mathrm{Ker}(\mathbf{G}(D)) = 0$. Every realizable generator matrix (i.e., a generator matrix all of whose components are causal) can be realized with a finite number of memory cells capable of storing ring elements and a finite number of scalars and adders that allow multiplication by constants and additions, respectively, within the ring R. A k-symbol input sequence, $a^i(D)$, produces an l-symbol output sequence, $b^j(D)$, where each is a sequence of integers modulo-P. The input/output relations are $b^j(D) = \sum_{i=1}^{k} a^i(D) g_{ij}(D)$. As in the field case, a systematic ring convolutional encoder is always minimal and noncatastrophic [54, 33].

Example 4.1: A rate-1/2 ring convolutional encoder with transfer function $\mathbf{G}(D) = [1 + 3D, 1 + D]$ over $\mathbf{Z}_4$ can use the obvious realization in Fig. 4.1(a).
The symbols $\oplus$ and $\odot$ represent modulo-4 addition and multiplication respectively. It has a four-state trellis, shown in Fig. 4.1(b). Each branch is labeled with $a^1(b^1b^2)$, where a is the input symbol and b is an output symbol. Each node is labeled with the content of the delay cell. It can been seen from the figure that the branches denoted by the two bold parallel lines have the same outputs for distinct inputs. Hence, this is a catastrophic encoder. However, this code can also be implemented as in Fig. 4.2(a)



(a)



(b)

Figure 4.1: (a) $\mathbf{G}(D) = [1+3D, 1+D]$ (b) Four-state trellis generated by $[1+3D, 1+D]$

(The transfer function is $[1, (1+D)/(1+3D)]$).

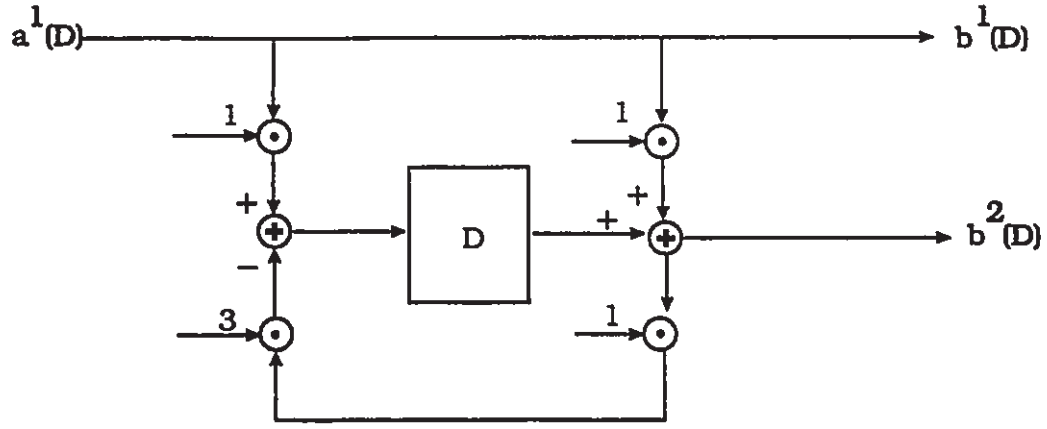
It is a systematic encoder with feedback and the resulting trellis has only two states, i.e. 0 and 2. These two states are obtained by enumerating the different contents in the delay cell, starting from the zero state and for different inputs. Fig. 4.2(b) shows the trellis diagram. We note that in this case each trellis branch actually consists of two parallel branches and the minimum distance of the code will be determined by the distance between parallel branches.

References [37, 35] consider a feedback realization of a ring convolutional encoder. This chapter will apply the feedback realization of a rate-$(l-1)/l$ systematic convolutional encoder, as shown in Fig. 4.3.

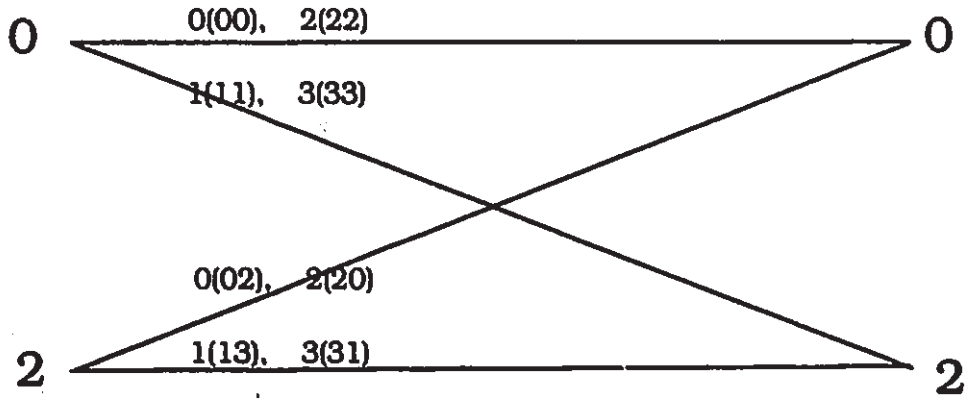
The transfer function for this encoder is in general

$$\mathbf{G}(D) = \begin{bmatrix} 1 & \cdots & 0 & \frac{H^1(D)}{H^0(D)} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & \frac{H^{l-1}(D)}{H^0(D)} \end{bmatrix} \tag{4.1}$$

where $H^i(D) = h_0^i + h_1^i D + \cdots + h_\nu^i D^\nu$ and ν is the number of delay cells in the encoder. In general, permutations between the redundancy carrying output line and the information-carrying output lines are necessary [27]. A permutation between a redundancy carrying output line and an information-carrying line corresponds to a permutation of the last column of $\mathbf{G}(D)$ and the column related to the information output line. For example, consider a rate-1/2 modulo-4 encoder $\mathbf{G}(D) = [1, (1+D)/(1+3D)]$. Permuting the information symbol output line with the redundancy-carrying output line will produce another encoder with transfer function $[(1+D)/(1+3D), 1]$. This corresponds to a permutation of the columns of $[1, (1+D)/(1+3D)]$. Cascading these two encoders with CPFSK may produce codes with different minimum squared Euclidean distances. Permutation of information lines is not necessary in the present case because we have assumed that the input data are random.



(a)



(b)

Figure 4.2: (a) $\mathbf{G}(D) = [1, (1+D)/(1+3D)]$ (b) Two-state trellis generated by $[1, (1+D)/(1+3D)]$

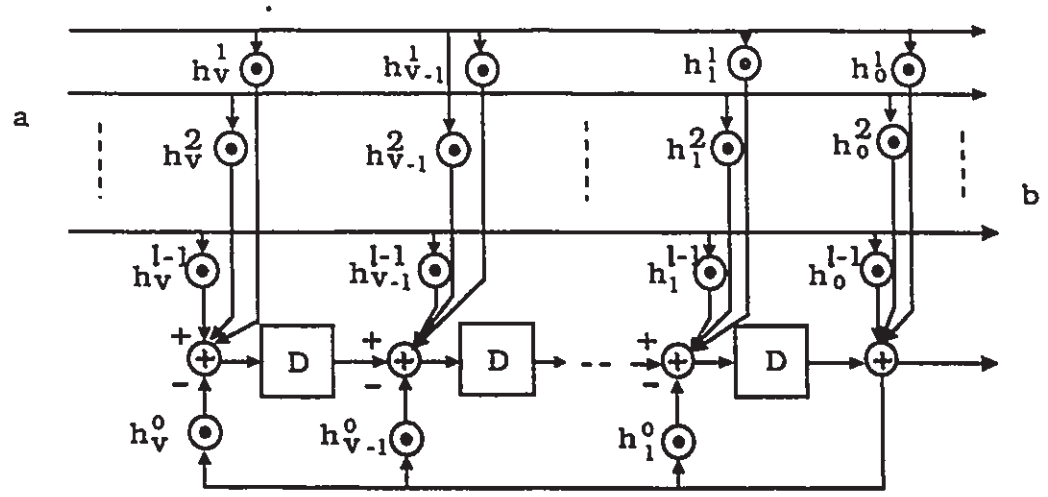


Figure 4.3: Rate-$(l-1)/l$ systematic ring convolutional encoder

4.3 System Model of Ring Convolutional Coded CPFSK

The system model of a ring-convolutional-encoded CPFSK system is shown in Fig. 4.4, where G is a rate-k/l ring convolutional encoder over $\mathbf{Z}_P$. The vectors of polynomials $\mathbf{a} = (a^1(D), \ldots, a^i(D), \ldots, a^k(D))$ and $\mathbf{b} = (b^1(D), \ldots, b^j(D), \ldots, b^l(D))$ are input/output of G. Each $a^i(D)$ or $b^j(D)$ is a sequence of integers modulo-P. A scrambler T, as in Fig. 4.5, is connected to the feedback P^{k_M}-ary CPFSK with $h = K/P$ to produce a feedback-free CPE (the reason to consider this class of CPFSK can be found in Chapter 2). The output data sequences b are partitioned into blocks of length k_M modulo-P symbol sequences, $\mathbf{U}^*$. Each block of modulo-P symbols produces a CPFSK channel symbol. The reason for using the feedback-free CPE is because catastrophic convolutional codes may combine with the feedback CPE to produce, overall, noncatastrophic codes [3]. To find the best codes for CPFSK, then, we should also consider catastrophic encoders. The conditions for catastrophic codes to produce noncatastrophic overall encoders must be found before code searching can take place. Also, with feedback-free CPE, noncatastrophic CE's always generate non-

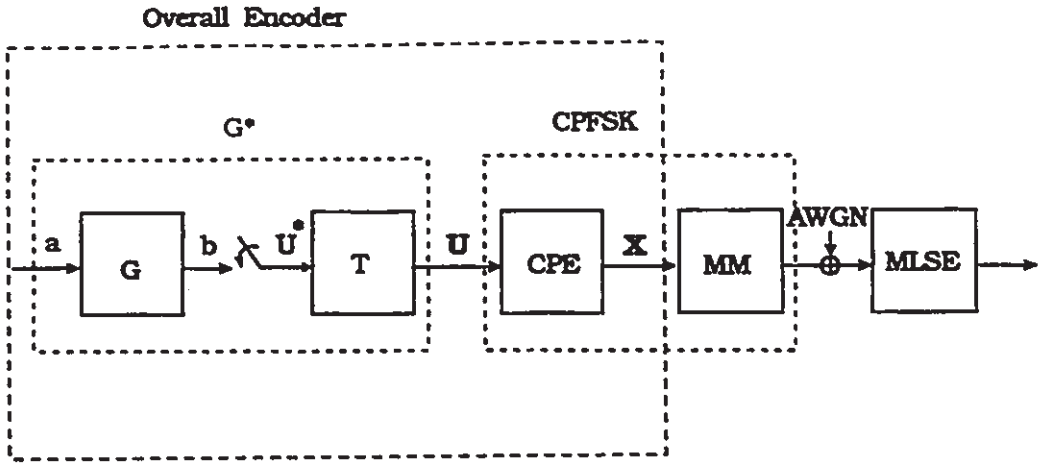


Figure 4.4: Modulo-P coded CPFSK system.

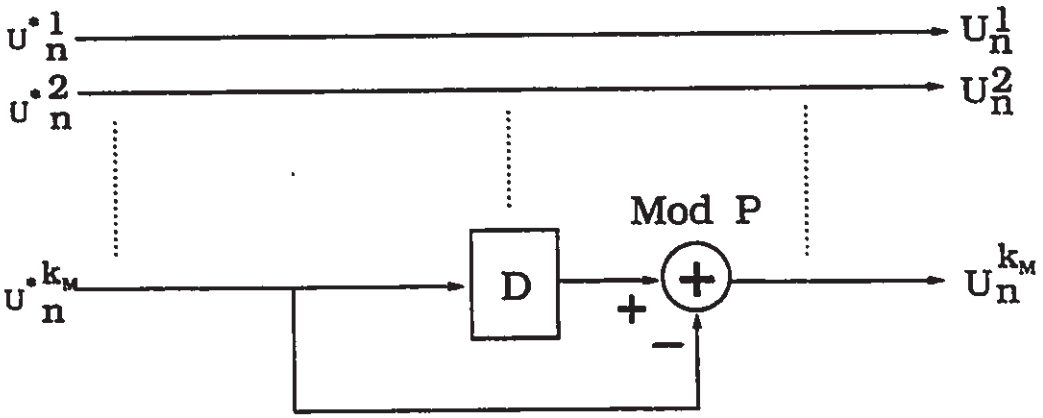


Figure 4.5: $k_M \times k_M$ scrambler $T(D)$.

catastrophic overall encoders, while catastrophic CE's always generate catastrophic overall encoders. Hence, code searching is simplified by considering the feedback free model [29]. The overall complexity of the MLSE receiver depends on the number of states in the combined trellis of the CE and CPE. A state of the combined trellis can be represented as $\sigma_V = [\sigma_G, \sigma_M]$, where σ_G and σ_M are the states in the CE and CPE respectively.

From knowledge of the equivalence between the feedback-free CPE and the original CPE, the corresponding best codes $\mathbf{G}^*$ for the feedback CPE model can be found by scrambling the codewords of the best codes $\mathbf{G}$ for the feedback-free CPE (c.f. Fig. 4.4).

Because T is a scrambler over the ring of integers modulo P, G* is also a modulo-P convolutional encoder. One advantage of using modulo-P encoders is that there is no performance difference between the feedback and feedback-free CPFSK models.

The information rate (bits/symbol) of the overall encoded system is given by

$$R = R_{CE}R_{CPE}R_{MM} = \frac{k\log_2 P}{l\log_2 P}\cdot\frac{k_M\log_2 P}{(k_M+1)\log_2 P}\cdot\frac{(k_M+1)\log_2 P}{1} = \frac{k}{l}\cdot k_M\cdot\log_2 P \tag{4.2}$$

where R_{CE} is the code rate of the CE, R_{CPE} is the code rate of the CPE and R_{MM} (bits/symbol) is the information rate of the MM. For example: for a rate-1/2 modulo-4 convolutional coded CPFSK with $h = 1/4$, $R_{CE} = 1/2$, $R_{CPE} = 1/2$ and $R_{MM} = 2\log_2 4$ (bits/symbol). So the total coded information rate $R = 1$ (bit/symbol).

4.4 Cascade of Modulo-P Encoder with Feedback-Free CPFSK

Previously [3, 26, 4], two equivalent approaches have been used to find the complexity of encoded CPFSK; one, called the match encoding approach, *indirectly* [3] finds the combined trellis of the channel encoder and CPFSK [3, 29]; the second, called the decomposition approach [4], *directly* specifies the overall encoder of the CE/CPE pairs. We will apply both to find the complexity of modulo-P encoded CPFSK.

4.4.1 Combination of Ring Convolutional Encoder with Feedback-Free CPFSK

It is clear that changes in the structure of the channel encoder can change the number of states in the overall encoder. The complexity in the MLSE receiver depends on the number of states in the overall encoder. We will give a criterion to decide the overall number of states when modulo-P encoders are connected with the feedback-free CPFSK model.

First, two definitions from ring theory are given [55]:

Definition: If a subset S of a ring R is itself a ring under the addition and multiplication in R, then we say that S is a *subring* of R.

Definition: A subring I of a ring R is an *ideal* provided: Whenever $a \in R$ and $b \in I$, then $ab \in I$ and $ba \in I$.

From the above two definitions, an interesting characteristic of ring convolutional codes is obtained : the set I_s formed by all the last modulo-P symbols of any codewords before merging to the zero state (include the zero symbol) is an *ideal* in the ring $\mathbf{Z_P}$. This can be seen from the fact that ring convolutional codes are linear, i.e. the addition of any two codewords and the multiplication of any codewords with any elements in the ring must be a codeword. Therefore, the set I_s satisfies the definition of an *ideal*.

Lemma 1: Consider a rate k/l modulo-P convolutional encoder connected to feedback-free P^{k_M}-ary CPFSK having modulation index $h = K/P$ with l a multiple of k_M. The total number of states, S_V, in the combined trellis of CE and CPFSK is then

$$S_V = S_G \cdot n_s \tag{4.3}$$

In (4.3), S_G is the number of states in G (the channel encoder). Let b_i be the last modulo-P symbol of any codeword before it merges to the zero state, then n_s is the number of elements in the ideal, I_s, formed by all b_i's.

Proof: Consider the feedback-free CPE of Fig. 4.6.

Let C_G be the set of codewords generated by a modulo-P convolutional encoder. A state in the CE/CPE trellis can occur if and only if there exist two codewords of C_G that produce a state merge in the CE and the CPE simultaneously. First, we consider the possible states of the combined trellis generated by all the nonzero codewords that merge to the zero encoder state. These states of the combined trellis are $\sigma_V = [\sigma_G, \sigma_M] = [0, \sigma_M]$, where the CE state, σ_G, is zero and the CPE state, σ_M, is chosen from one of the P possible CPE states. Depending on the structure of the

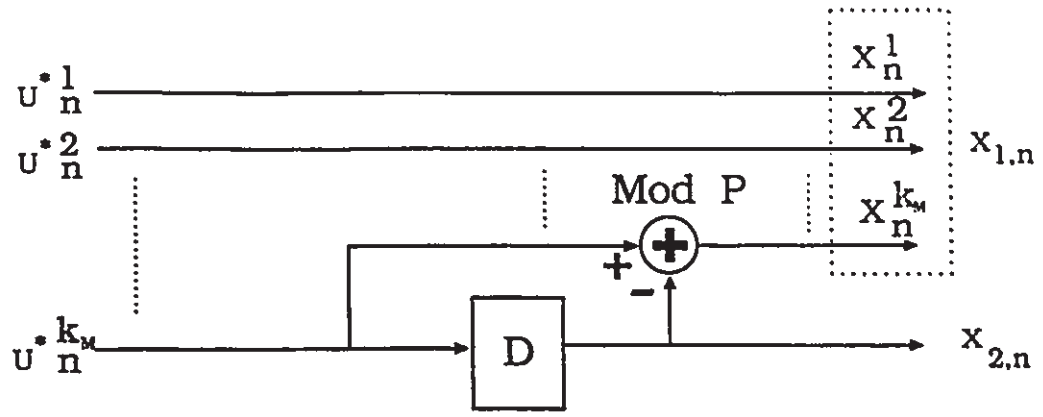


Figure 4.6: Feedback free continuous phase encoder.

encoders, the combined trellis states, with zero encoder state, will have no more than P possibilities. Due to the linearity of the convolutional codes, the total number of combined trellis states is the product of the number of encoder states and the number of possible combined trellis states with encoder state equal to zero.

Let $\mathbf{b}_1$ and $\mathbf{b}_2$ be any two codewords generated by G that diverge at some time m and merge at time $n-1$ $(n > m+1)$ in zero state $\sigma_G = [\mathbf{0}]$, where

$$\begin{aligned} \mathbf{b}_1 &= (\ldots, b_{1,n-4}, b_{1,n-3}, b_{1,n-2}, b_{1,n-1}, 0, 0\ldots) \\ \mathbf{b}_2 &= (\ldots, b_{2,n-4}, b_{2,n-3}, b_{2,n-2}, b_{2,n-1}, 0, 0\ldots). \end{aligned} \tag{4.4}$$

In (4.4), $b_{1,n-1} = b_{2,n-1} = b_i$. From the fact that the content of the CPE is only a delay of b, it will merge in a CPE state $\sigma_M = b_i$ at time n. Therefore it produces a combined trellis state $[\mathbf{0}, b_i]$ in the overall trellis. Let n_s be the number of elements in the ideal formed by the last modulo-P symbols of all the codewords before merging to the zero state. Hence, the states in CPE have n_s possibilities to combine with the zero state in the CE. Therefore, $S_V = S_G \cdot n_s$. This ends the proof.

Example 4.2: Consider a rate-1/2 systematic encoder $\mathbf{G}(D) = [1, \frac{1}{1+2D}]$ over $\mathbf{Z}_4$ connected to quaternary CPFSK with modulation index $h = 1/4$, as in Fig. 4.7(a). The encoder state trellis diagram is in Fig. 4.7(b). From the trellis diagram, it can be seen that every codeword has the last modulo-P symbol equal to zero or two before it merges to the zero state. The number of different elements in the ideal formed

by the last modulo-P symbols is only two (i.e. 0 and 2). The combined trellis has, therefore, only four states as shown in Fig. 4.7(c).

Example 4.3: Consider a rate-1/2 systematic encoder $\mathrm{G}(D) = [2 + D, 1]$ over $\mathbb{Z}_4$ connected with quaternary CPFSK with modulation index $h = 1/4$, as in Fig. 4.8(a). The encoder state trellis diagram is in Fig. 4.8(b). From the trellis diagram, it can be seen that every codeword has the last modulo-P symbol equal to zero before it merges to the zero state. The number of elements in the ideal of the last modulo-P symbols is only one (i.e. 0). The combined trellis has, therefore, only four states as shown in Fig. 4.8(c).

Example 4.3 says that we can reduce the number of combination states to one-fourth of S_G times P (the number of states of modulator). Previous approaches [3, 4, 30, 5] can only reduce the number of states to one-half of S_G times P compared to the traditional approach. This is an important feature of modulo-P systems. If every codeword has its last modulo-P symbol equal to zero before it merges to the zero state, Lemma 1 tells us that we can reduce the number of states to $1/P$ of S_G times P for a modulo-P system compared to the traditional approach. This does not happen in binary convolutional coded systems.

4.4.2 Transfer Function of the Overall Encoder

The channel encoders we consider are systematic encoders and have the advantages of being both minimal and noncatastrophic [54]. From [37, 35], the number of states of a systematic encoder can be found by enumerating the possible reachable states, starting with the zero state, for every possible input. We will follow this approach to count the minimum number of states in the CE and the number of states in the overall encoder can be found by enumerating all the possible combinations of CE/CPE pairs.

The procedure to find the transfer function of the overall encoder will be developed for different examples. It is obvious that this procedure can be applied to other code

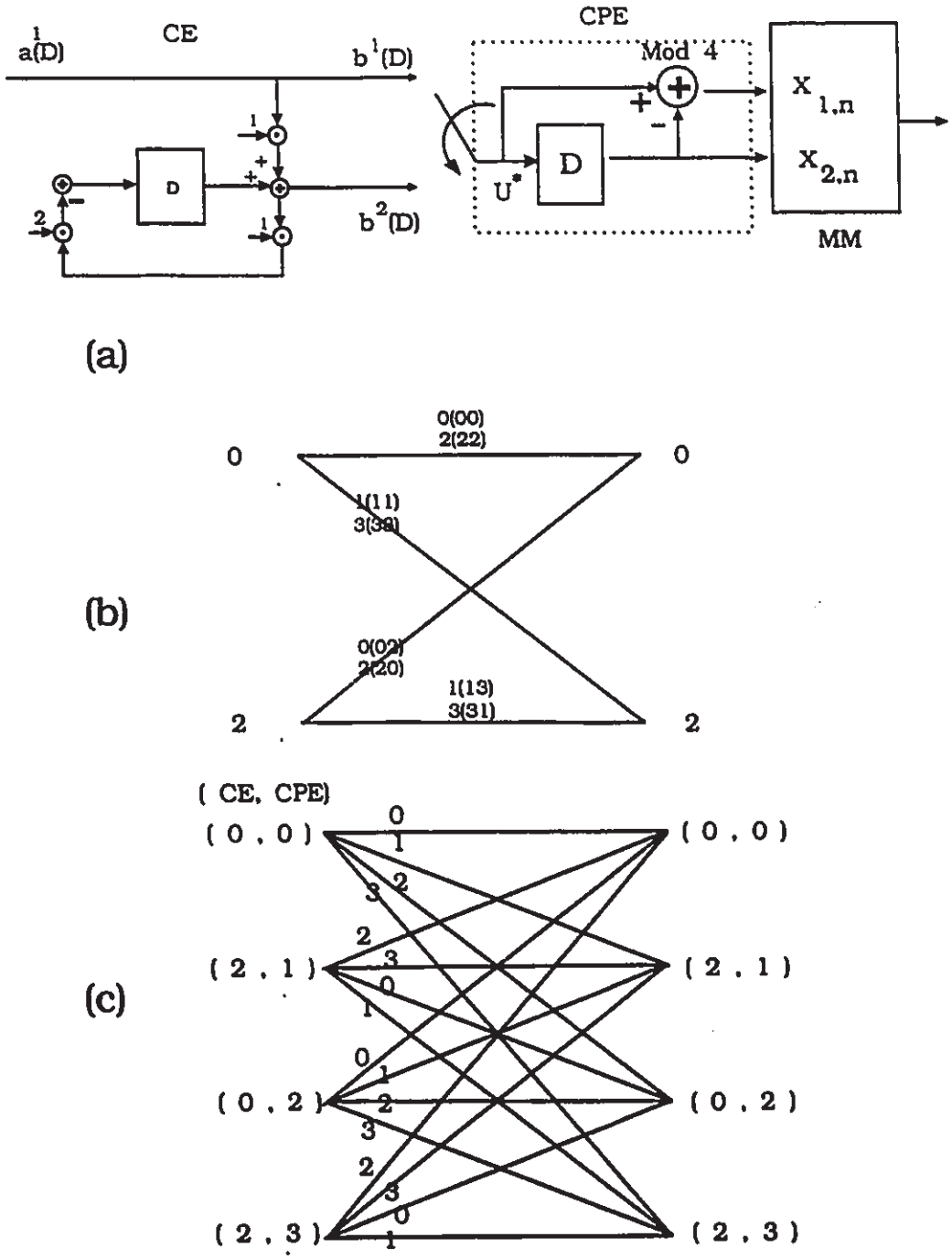
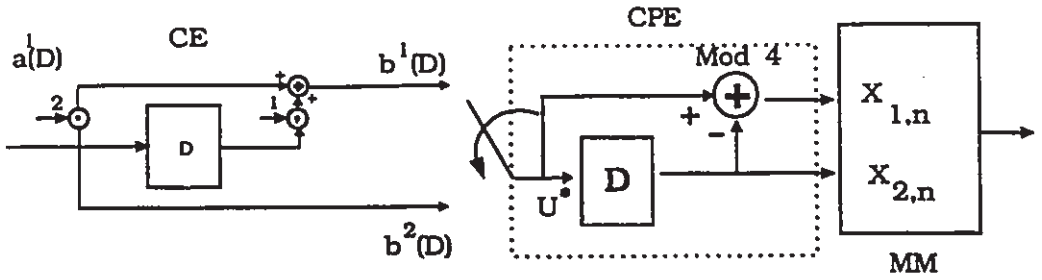
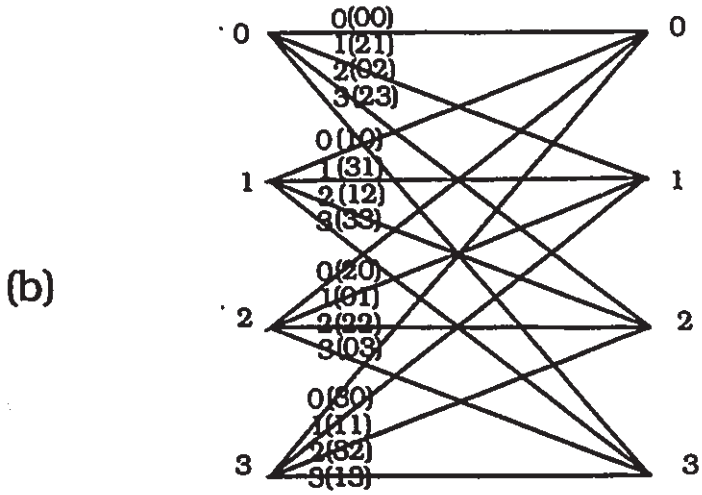


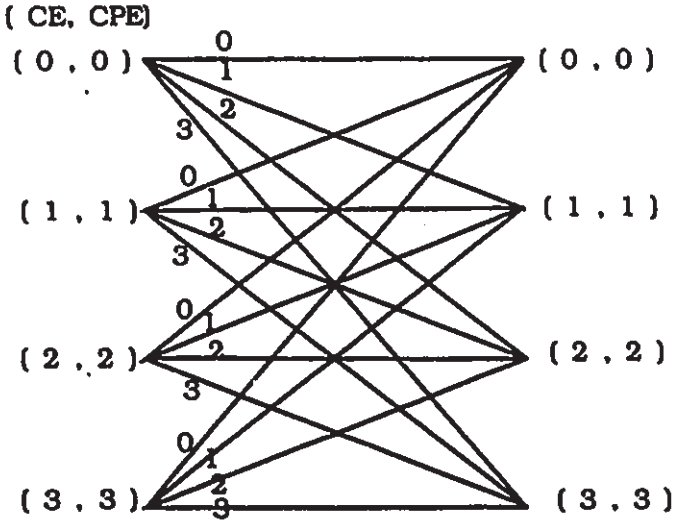
Figure 4.7: (a) $G(D) = [1, 1/(1+2D)]$ connects with $h = 1/4$ quaternary CPFSK (b) Trellis diagram for $G(D)$ (c) The combined trellis of the CE/CPE pairs.



(a)



(b)



(c)

Figure 4.8: (a) $\mathbf{G}(D) = [1, 1/(1+2D)]$ connects with $h = 1/$ quaternary CPFSK (b) Trellis diagram for $\mathbf{G}(D)$ (c) The combined trellis of the CE/CPE pairs.

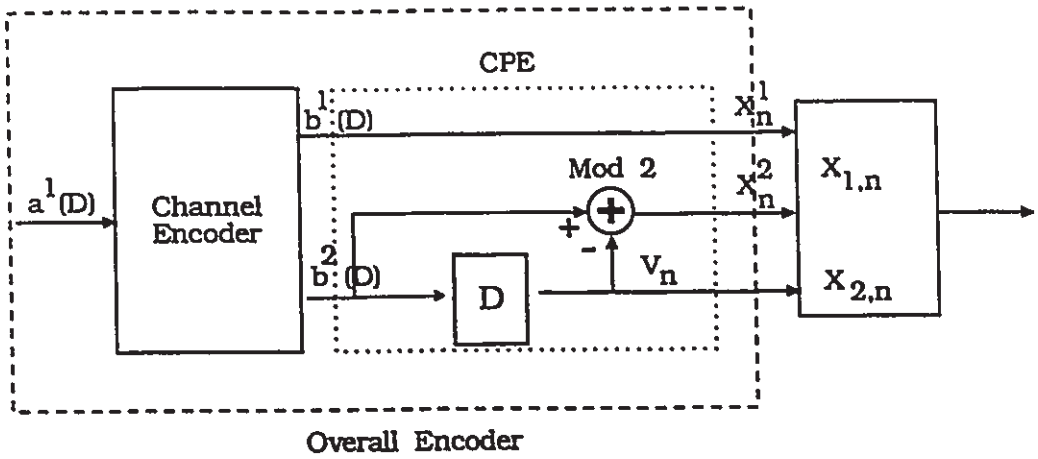


Figure 4.9: Rate-1/2 modulo-2 coded $h = 1/2$ quaternary CPFSK.

rates and different modulation indices.

(a) **Rate-1/2 modulo-2 CE connected with $h = 1/2$ quaternary CPFSK**

For $h = 1/2$ and $M = P^{k_M} = 4$ CPFSK, P is 2 and $k_M = 2$. The resulting transmitted information rate $R = (1/2) \cdot 2 \cdot 1 = 1$ (bit/symbol) (see equation (4.2)). Referring to Fig. 4.6, the CPE of this modulator is a rate-2/3 encoder with transfer function $\mathbf{C}'(D)$. The encoded system is shown in Fig. 4.9.

The CE is binary ($P = 2$) with rate-1/2 and transfer function $\mathbf{G}(D) = [1, g(D)]$, $g(D) \in R(D)$. The columns of $\mathbf{G}(D)$ can be permuted, i.e. $\mathbf{G}(D)$ can also be $[g(D), 1]$. The overall encoder is a rate-1/3 $(1/2 \cdot 2/3)$ binary convolutional encoder. The overall transfer function is

$$\mathbf{G}(D) \cdot \mathbf{C}'(D) = \mathbf{G}(D) \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1-D & D \end{bmatrix} \tag{4.5}$$

(b) **Rate-1/2 modulo-4 CE connected with $h = 1/4$ quaternary CPFSK**

For $h = 1/4$ quaternary CPFSK, P is 4 and $k_M = 1$. The resulting transmitted information rate $R = (1/2) \cdot 1 \cdot \log_2 4 = 1$ (bit/symbol) (see equation (4.2)).

In Fig. 4.10, both the CE and CPE are rate-1/2 modulo-4 encoders. One modulo-4 information symbol to the CE produces two encoded modulo-4 symbols, and each

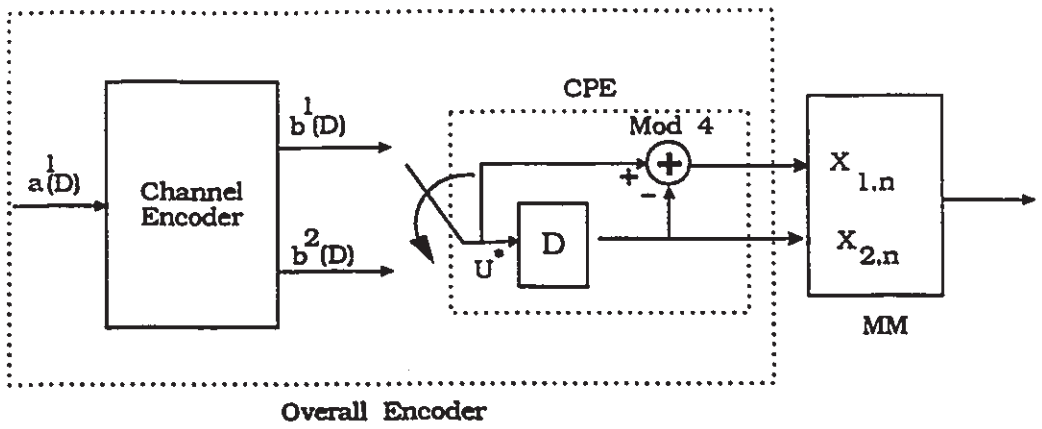


Figure 4.10: Rate-1/2 modulo-4 coded $h = 1/4$ quaternary CPFSK.

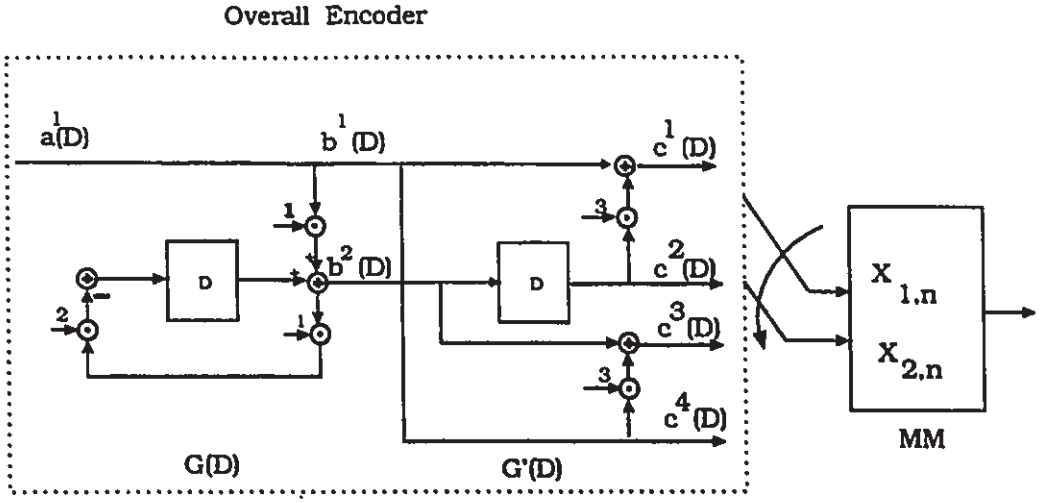
encoded modulo-4 symbol to the CPE produces two modulo-P symbols. Two modulo-P symbols from the CPE input to the MM generate a channel symbol (see equation (1), (2.8) and (2.36)). To merge the CPE to the rate-1/2 CE, the CPE can be transformed to a equivalent rate-2/4 encoder, $\mathbf{G}'(D)$, (the derivation is in Appendix A). The overall encoder is a rate-1/4 ($1/2 \cdot 2/4$) encoder with transfer function

$$\mathbf{G}(D) \cdot \mathbf{G}'(D) = \mathbf{G}(D) \cdot \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3D & D & 1 & 0 \end{bmatrix}, \tag{4.6}$$

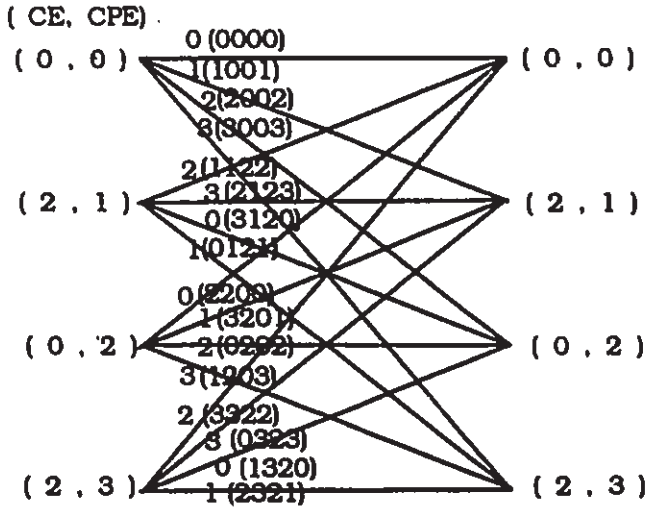
where $\mathbf{G}(D) = [1, g(D)]$ or $[g(D), 1]$, $\mathrm{g(D)} \in R(D)$. The inputs of the overall encoder are $a^1(D)$ and the outputs are represented as $\mathbf{c} = [c^1(D), c^2(D), c^3(D), c^4(D)]$. This is consistent with the original model. One information symbol will generate four encoded modulo-P data, and every two encoded modulo-P symbols to the MM produce a channel symbol. The information rate $R = 1/2$ (modulo-4 symbol/channel symbol) $= 1$ (bit/channel symbol).

Example 4.4: Consider a rate-1/2 modulo-4 encoder $\mathbf{G}(D) = [1, \frac{1}{1+2D}]$ combined with $h = 1/4$ quaternary CPFSK. In example 2 the same encoder is cascaded with the modulator and the complexity of the combination is obtained by using Lemma 1.

Fig. 4.11 shows the overall encoder obtained from the transfer function (4.6). It is a rate-1/4 modulo-4 encoder. Every branch is labeled with the input/output relation,



(a)



(b)

Figure 4.11: (a) $\mathbf{G}(D) = [1, \frac{1}{1+2D}]$ merges with the CPE (b) Trellis diagram for the overall encoder, $\mathbf{G}(D) \cdot \mathbf{G}'(D)$

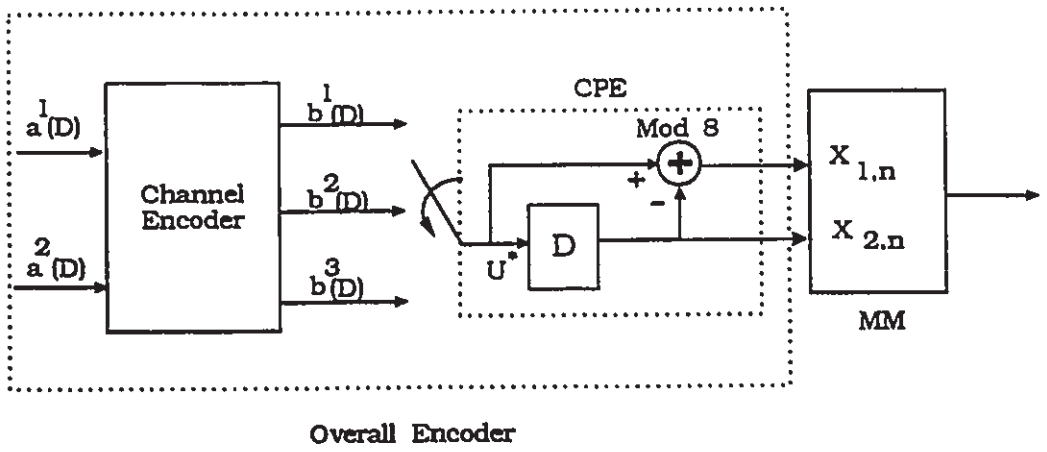


Figure 4.12: Rate-2/3 modulo-8 coded $h = 1/8$ octal CPFSK.

$a^1(c^1c^2c^3c^4)$, where a is the input modulo-P symbol and c's are the encoded modulo-4 symbols. The number of states is obtained by enumerating the *reachable states* [35, 37], starting from the all zero state and different inputs. It is the same as the four-state trellis in Fig. 4.7(c) (in Fig. 4.7(c) only input symbol is labeled on each branch.) The NMSED can be computed by using the Viterbi algorithm and the incremental squared Euclidean distance equation (2.54). The minimum distance d^2_{min} equals 3.15 (1.97 dB gain over MSK).

(c) Rate-2/3 modulo-8 CE connected with $h = 1/8$ octal

For $h = 1/8$ and $P^{k_M} = 8$ CPFSK, $P = 8$ and $k_M = 1$. The resulting transmitted information rate $R = (2/3) \cdot 1 \cdot \log_2 8 = 2$ (bits/symbol) (see equation (4.2)).
In Fig. 4.12, the CE is a rate-2/3 modulo-8 encoder and the CPE is a rate-1/2 modulo-8 encoder. In order to merge the CPE to the CE, an equivalent rate 3/6 modulo-8 CPE , $\mathbf{G}'(D)$, is found (see Appendix A). The overall encoder is a rate-2/6 (2/3·3/6) modulo-8 encoder with transfer function

$$\mathbf{G}(D) \cdot \mathbf{G}'(D) = \begin{bmatrix} 1 & 0 & g_1(D) \\ 0 & 1 & g_2(D) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 7 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 7 & 1 \\ 7D & D & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.7}$$

$g_1(D), g_2(D) \in R(D)$. The last column of $\mathbf{G}(D)$ can be permuted with the first two

Table 4.1: Normalized minimum squared Euclidean distance for rate-1/2 modulo-2 encoded $h = 1/2$ quaternary CPFSK

S_V	ENCODER G	d^2_{min}	d^2_{min} of the best solution in [6]
2	[D,1]	2.00	2.00
4	$[\frac{D^2}{1+D},1]$	4.00	4.00
8	$[\frac{D^2+D^3}{1+D+D^2},1]$	5.00	5.00
16	$[\frac{D^2+D^4}{1+D+D^3},1]$	6.00	6.00

columns. The inputs of the overall encoder are $\mathbf{a} = [a^1(D), a^2(D)]$ and the outputs are represented as $\mathbf{c} = [c^1(D), c^2(D), \ldots, c^6(D)]$.

4.5 Numerical Results

In this section, code search results are presented for rate-1/2 binary encoded quaternary CPFSK with $h = 1/2$, rate-1/2 modulo-4 encoded quaternary CPFSK with $h = 1/4$ and rate-2/3 modulo-8 encoded octal CPFSK with $h = 1/8$. These cases are of interest in coded CPFSK systems [4]. The searching procedure is based on the Viterbi algorithm to find the d^2_{min} [8]. The best codes are found by searching the channel encoders achieving the best minimum normalized squared Euclidean distance for a fixed complexity, S_V, in the overall MLSE receiver.

(a) rate-1/2 binary encoded $h = 1/2$ quaternary feedback-free CPFSK ($R = 1$ bit/symbol)

Table 4.1 provides codes for $h = 1/2$ quaternary feedback-free CPFSK up to $S_V = 16$ states. The resulting codes have minimum distance equal to the best previously published results [6]. The best codes $\mathbf{G}^*$ for CPFSK can be obtained by scrambling

Table 4.2: Normalized minimum squared Euclidean distance for rate-1/2 modulo-4 encoded $h = 1/4$ quaternary CPFSK

ν	S_V	G	d^2_{min}	(a) [6]	(b) [4]	(c) [30]	(d) [1]	Gain(db) over MSK
1	4	$[1, \frac{1}{1+2D}]$	3.15	2.42	2.15	2.73	2.00	1.97
1	8	$[1, \frac{1+2D}{1+D}]$	4.09	3.00	3.57	4.00	3.00	3.11
2	16	$[\frac{2+3D+2D^2}{1+D}, 1]$	5.15	4.42	4.51	4.88	4.30	4.11
3	32	$[\frac{2+D}{1+D+2D^3}, 1]$	6.00	5.24	5.82	5.45*	5.24	4.68
3	64	$[1, \frac{3+D^2+2D^3}{1+2D+D^3}]$	6.42	6.15	6.18	N.A.	6.15	4.94
4	128	$[1, \frac{1+2D+D^3+2D^4}{1+D^2+2D^3+D^4}]$	7.60*	N.A.	7.09	N.A.	N.A.	5.80
4	256	$[\frac{2+3D+D^3+2D^4}{1+2D+D^2+D^3}, 1]$	7.90*	N.A.	7.24	N.A.	N.A.	5.97

*: search not complete, N.A.: not available,(a)The matched encoding approach, (b)The decomposition approach, (c)Double trellis encoding approach, (d)The traditional approach

Table 4.3: Normalized minimum squared Euclidean distance for rate-2/3 modulo-8 encoded $h = 1/8$ octal CPFSK.

ν	S_V	ENCODER G	d^2_{min}	d^2_{min} of the best solution in [8]	Gain(dB) over $h = 1/4$ quaternary CPFSK
1	8	$\begin{bmatrix} 1 & 0 & 6+4D \\ 0 & 1 & 2 \end{bmatrix}$	2.18	1.45	1.77
1	16	$\begin{bmatrix} 1 & 0 & \frac{3+4D}{1+2D} \\ 0 & 1 & \frac{5}{1+2D} \end{bmatrix}$	2.38	1.47	2.13
2	32	$\begin{bmatrix} 1 & 0 & \frac{4+2D}{1+4D+3D^2} \\ 0 & 1 & \frac{1+D}{1+4D+3D^2} \end{bmatrix}$	2.93	2.55	3.05
2	64	$\begin{bmatrix} 1 & 0 & \frac{3+2D}{1+D+4D^2} \\ 0 & 1 & \frac{1+D}{1+D+4D^2} \end{bmatrix}$	3.43	3.28	3.74

the best codes G for feedback-free CPFSK with the scrambler

$$T(D) = \begin{bmatrix} 1 & 0 \\ 0 & 1-D \end{bmatrix}. \tag{4.8}$$

(b) rate-1/2 modulo-4 encoded $h = 1/4$ quaternary feedback-free CPFSK ($R = 1$ bit/symbol)

In Table 4.2, the best rate-1/2 modulo-4 codes for $h = 1/4$ quaternary CPFSK are given. The comparison assumes the convolutional encoders are of the same rate, measured in bits of information per modulation symbol as in previous work [6, 4, 30]. The variable ν is the number of delay cells in the modulo-P encoder and S_V is the number of states in the overall encoder. Since rate-1/2 coded $h = 1/4$ quaternary CPFSK has approximately the same spectral efficiency as MSK, performance is compared to MSK. Coding gain is defined as

$$\text{Coding gain} = 10 \cdot \log_{10} \frac{d^2_{min}}{d^2_{MSK}} \tag{4.9}$$

where $d^2_{MSK} = 2$ is the normalized minimum squared Euclidean distance of MSK. It can be seen that the new coding scheme yields superior d^2_{min} to all previous research results. The increase in coding gain for S_V equal to four is almost 2 dB, so Rimoldi's model becomes more useful than it appears in [4], where no more than 0.3 dB improvement was found. The best codes $\mathbf{G}^*$ are obtained by scrambling G with

$$\begin{bmatrix} 1 & 3 \\ 3D & 1 \end{bmatrix}. \tag{4.10}$$

This is derived from the fact that the codewords of G are scrambled serially by $1-D$, and $1 - D \equiv 1 + 3D$ modulo 4 (the procedure is similar to Appendix A).

(c) rate-2/3 modulo-8 encoded $h = 1/8$ octal CPFSK ($R = 2$ bits/symbol)

In Table 4.3, the best rate-2/3 modulo-8 codes for $h = 1/8$, octal CPFSK are given. Since rate-2/3 coded $h = 1/8$ octal CPFSK has approximately the same spectral

efficiency as $h = 1/4$ quaternary CPFSK ($d^2_{min} = 1.45$) [8, 4], coding gain is compared to that modulator (replace $d^2_{MSK} = 2$ with $d^2_{min} = 1.45$ for $h = 1/4$ quaternary CPFSK in (4.9)). It can be seen that the new coding scheme yields performance superior to the results in [8]. The best codes G* are obtained by scrambling G with

$$\begin{bmatrix} 1 & 7 & 0 \\ 0 & 1 & 7 \\ 7D & 0 & 1 \end{bmatrix}. \tag{4.11}$$

This is derived from the fact that the codewords of G are scrambled serially by $1 - D$, and $1 - D \equiv 1 + 7D$ modulo 8. The reported code searches are not complete in this table because of the long search times involved.

4.6 Discussion

The ring encoded CPFSK system has the following characteristics that have combined the ideas of previous binary convolutionally coded CPFSK.

- The combined states of the overall encoder can be reduced to $(1/S_M) \cdot S_V S_M$, where $S_M = P$, as predicted by Lemma 1. This cannot be achieved by any previous coding schemes.
- The combination of the CE/CPE pairs is *natural* in the sense of using the same code structure as the CPE. The overall encoder is a linear ring convolutional encoder. There is no performance difference between encoded CPFSK and encoded feedback-free CPFSK.
- This coding scheme considered can use both binary and quaternary encoding, depending on the code structure of CPE.

These desirable characteristics have been previously suggested for achieving coding gain (see the discussion in the Chapter 3), and the coding scheme presented here has encompassed these ideas for coded CPFSK.

Chapter 5

Performance Analysis of Ring Convolutional Coded CPFSK System

This chapter studies the bit error probability (BER) of modulo-P encoded CPFSK systems. In general, the BER of digital communications systems can be determined by the normalized minimum squared Euclidean distance among all the possible transmitted channel symbols, provided the signal to noise ratio is large enough [56, 39]. However, the number of pairs of channel symbol sequences separated by the normalized minimum squared Euclidean distance, called the error coefficient, may increase the BER significantly if it is a large number. It is therefore an important consideration when studying the upper bound on the BER for ring convolutional encoded CPFSK. An upper bound on the bit error probability, based on the averaged transfer function techniques [42, 57], is derived for modulo-P encoded systems. Numerical results are given and show that the Euclidean distance is a good parameter to predict the performance of ring encoded CPFSK. Ring encoded CPFSK does not have a large error coefficient, and it performs well for practical and moderate signal to noise ratios.

5.1 Introduction

Previous chapters have introduced a new encoding scheme for CPFSK based on convolutional codes over the ring of integers modulo-P. Modulo-P encoded CPFSK systems consistently obtain good coding gain compared to previous work. However, the comparisons are based on the assumption that the signal to noise ratio is sufficiently large. This chapter studies the BER bound for these systems and investigates the system performance for different signal to noise ratios.

For maximum likelihood decoding of convolutional codes, the Viterbi algorithm is a well known and effective approach. The performance bound on the error probability for systems using the Viterbi algorithm was introduced in [38]. The approach used to derive the performance bound is called the *transfer function approach.* In addition to decoding convolutional codes that have linear trellis structures, the Viterbi algorithm can also be applied to maximum likelihood demodulation of nonlinear trellis codes. A generalized bounding technique to find the performance bound for nonlinear trellis has been derived in [42, 57]. This bounding technique is called the *average transfer function approach.*

Most useful digital modulations for bandlimited channel are nonlinear trellis codes, for example, trellis coded modulation (TCM) and continuous phase modulation (CPM) are all nonlinear trellis codes and have applied the Viterbi algorithm for their maximum likelihood demodulation. Average transfer function bounds for trellis coded modulation can be found in [13], and for traditional binary coded or uncoded CPM in [43].

From previous chapters, it is known that the CPM modulator can be seen as a linear ring convolutional encoder cascaded with a memoryless modulator. Therefore, it is no surprise that the average transfer function approach, applied to a channel encoder connected with memoryless modulator [13, 58, 14], can also be applied to uncoded CPM. Moreover, the combination of a ring channel encoder with CPE is also a modulo-P linear encoder, so it is very convenient to view coded CPM in the

same way as trellis encoded memoryless modulation.

Based on the *average transfer function technique* developed in [42, 57, 43, 40], an upper bound on the BER is derived for the decomposition model of CPFSK (with or without the modulo-P channel encoder). The derivation follows the steps in [43] and [13] by viewing the encoded system as a linear ring encoder, with the CE combined with the CPE, connected to a memoryless modulator. The bound obtained is similar to the bound for combining a channel encoder with a memoryless modulator [13].

Numerical results are given in figures (5.16)-(5.26) for several examples of coded and uncoded CPFSK systems discussed in the last chapter. The conclusions are that ring convolutional encoded CPFSK has a small error coefficient, feedback-free CPFSK has a better BER than CPFSK and ring convolutional encoded CPFSK is robust for high and moderate signal to noise ratios.

5.1.1 System Model

The system model is shown in Fig. 4.4. The CE is a rate-k/l modulo-P encoder and the CPE is a rate-k_M/k_M+1 systematic modulo-P encoder. A scrambler, $\mathbf{T}$, can be connected to the CPE to produce a feedback-free form of CPFSK.

A *state* in the combined trellis of the CE/CPE is characterized by $\sigma_V = [\sigma_G, \sigma_M]$, where σ_G is the state in the CE and σ_M is the state in the CPE. The number of states in the combined trellis is S_V. The state of the combined trellis is labeled as s, where $s \in \{1, 2, \ldots, S_V\}$. An interval of the combined trellis consists of $\hat{k}$ modulo-P information symbols and $\hat{N}$ output channel symbols. The variables $\hat{k}$ and $\hat{N}$ have the relation

$$\frac{\hat{k}}{k} \cdot l = k_M \cdot \hat{N} \tag{5.1}$$

The input information polynomial vector a is a modulo-P symbol sequence. We can separate the input polynomial vector a into blocks of modulo-P symbols

$$[(a_0, a_1, \ldots, a_{\hat{k}-1}), (a_{\hat{k}}, a_{\hat{k}+1}, \ldots), \ldots)] = [\mathbf{a}'_0, \mathbf{a}'_1, \ldots, \mathbf{a}'_j, \ldots]. \tag{5.2}$$

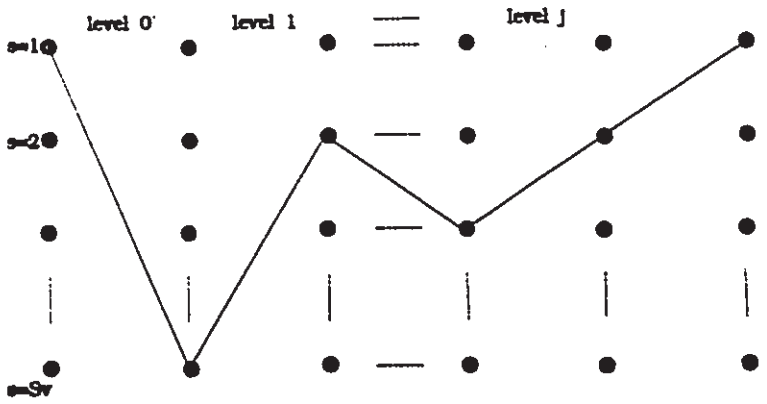


Figure 5.1: A transmitted path in the overall trellis

The variable $\mathbf{a}_j'$ represents a block of transmitted data at *level* j. Each block $\mathbf{a}_j'$ consists of $\hat{k}$ modulo-P symbols, equivalent to k' binary data, where

$$k' = \hat{k} \cdot \log_2 P. \tag{5.3}$$

The decoded modulo-P sequence is represented as $\hat{\mathbf{a}}$. The transmitted polynomial state transition vector, s, is defined as the sequence of transition state numbers corresponding to each block of data $\mathbf{a}_j'$, i.e., $\mathbf{s} = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_j, \ldots)$ and $\mathbf{s}_j \in \{1, 2, \ldots, Sv\}$. For time level j, the state number is $\mathbf{s}_j$ and the block of source symbols is $\mathbf{a}_j$. Figure 5.1 represent one possible transmitted path in the combined trellis. The receiver performs maximum likelihood demodulation in an AWGN channel.

5.1.2 Normalized Squared Euclidean Distance

It is well known that the error probability of digital transmission systems can be estimated by normalized squared Euclidean distance between transmitted signals [39, 56]. This estimation is good for systems in an AWGN environment and with practical signal to noise ratios. From [20, 51], it has been shown that the squared Euclidean distance (SED) between two CPFSK symbol sequences can be written as

$$D^2(\mathbf{U}, \mathbf{U}') = \int_{-\infty}^{\infty} [s(t, \mathbf{U}) - s(t, \mathbf{U}')]^2 dt, \tag{5.4}$$

where $s(t, \mathrm{U})$ and $s(t, \mathrm{U}')$ are the signals produced by two different encoded or uncoded data sequences, U and U'. At t = NT_s, this is readily simplified [20] as

$$\sum_{i=0}^{N} D_i^2(\mathrm{U}, \mathrm{U}'), \tag{5.5}$$

where, in terms of $\mathbf{X} = [X_{1,n}, X_{2,n}]$, (this has been shown in Chapter 2 and [31]) can be written

$$D_n^2(\mathbf{X}, \mathbf{X}') = \begin{cases} 2E_s[1 - \frac{\sin 2\pi h(X_{1,n}+X_{2,n}-X'_{1,n}-X'_{2,n})-\sin 2\pi h(X_{2,n}-X'_{2,n})}{2\pi h(X_{1,n}-X'_{1,n})}], & X_{1,n} \neq X'_{1,n} \\ 2E_s[1 - \cos 2\pi h(X_{2,n} - X'_{2,n})], & X_{1,n} = X'_{1,n} \end{cases} \tag{5.6}$$

Equation (5.6) specifies the relation between the codewords of the modulo-P CPE and the squared Euclidean distance of the outputs of the MM. When the cascading of the CE and CPE is to be considered as one entity, we can compute the SED between codewords with this modified incremental SED formula to compute the SED.

In an AWGN environment, the normalized squared Euclidean distance (NSED), is defined as

$$d^2 = \frac{D^2}{2E_b}. \tag{5.7}$$

The transmitted energy per bit is $E_b = E_s/R$ and R, in bits/symbol, is the information rate of the coded system. The minimum SED, D^2_{min}, between any two different signal sequences generated by the coded system is defined as $D^2_{min} = \min_{\mathrm{U}\neq\mathrm{U}'} D^2(\mathrm{U}, \mathrm{U}')$. The corresponding normalized SED, d^2_{min}, is called the minimum normalized squared Euclidean distance (NMSED).

5.2 An Upper Bound on the Bit Error Probability of Ring Convolutional Coded CPFSK

To evaluate the performance of linear convolutional codes, an approach called the transfer function bound was first introduced by Viterbi in [39]. This approach has been generalized to nonlinear trellises [42, 57, 13], such as trellis coded PSK or QAM

and ISI channels. It has also been used to find the symbol error rate for traditional CPM and the bit error rate for binary convolutional encoded CPM systems [57, 41, 43, 40]. The decomposition model of CPM [26] decomposes the portion of CPM with memory into a continuous phase encoder (CPE) and a memoryless modulator (MM). In fact, when considering modulo-P encoded CPFSK, the whole of the CE/CPE combination is a ring convolutional encoder. The modulo-P system is a time invariant linear channel encoder combined with a memoryless modulator. Therefore, it is more likely that we can obtain an upper bound which is similar to the system consisting of a channel encoder connected with a memoryless modulator (see [14, 58]). The following derivations are based on this decomposed model to derive an upper bound on the bit error probability for ring convolutional encoded CPFSK systems.

Assume L_P modulo-P information symbols in $\mathbf{a} = (a_0, a_1, \ldots, a_{\hat{k}}, a_{\hat{k}+1}, \ldots)$ are transmitted and the receiver decodes a corresponding sequence $\hat{\mathbf{a}}$. Let $m_b(\mathbf{a})$ and $m_b(\hat{\mathbf{a}})$ be the equivalent binary sequences with respect to the modulo-P sequences $\mathbf{a}$ and $\hat{\mathbf{a}}$ respectively. Here, we use a natural mapping rule, i.e., if P is a power of two, then a modulo-P symbol a_j and its binary equivalent symbols $\mathbf{u} = (u_1, u_2, \ldots, u_l)$, $l = \log_2 P$, have the relation

$$a_j = \sum_{i=1}^{i=l} u_i \cdot 2^{l-i}. \tag{5.8}$$

For example, if P is four, then **0** can be represented as (0,0), **1** as (0,1), **2** as (1,0), and **3** as (1,1), respectively. The bit error probability, P_b, is defined as the expected number of information bit errors per decoded information bit,

$$P_b = \frac{E[w_H(m_b(\mathbf{a}), m_b(\hat{\mathbf{a}}))]}{L_P \cdot \log_2 P}, \tag{5.9}$$

where $E[\cdot]$ denotes the expected value and w_H denotes the Hamming weight between $m_b(\mathbf{a})$ and $m_b(\hat{\mathbf{a}})$. Hamming weight is defined as the number of different bits between two binary data sequences. The expectation is over sequence pairs $(m_b(\mathbf{a}), m_b(\hat{\mathbf{a}}))$. The sequence $\mathbf{a}$ can be represented in terms of a block of modulo-P symbol sequence, $\mathbf{a} = [\mathbf{a}'_0\mathbf{a}'_1, \ldots, \mathbf{a}'_j, \ldots]$, where each block, $\mathbf{a}'_j$, consists of $\hat{k}$ modulo-P symbols, i.e., k'

binary digits. The total number of transmitted blocks is L_B. In the derivation of the upper bound on P_b, it is assumed that the information symbols a_i, $i = 0, 1, \ldots, L_P - 1$, are independent and identically distributed. The upper bound on P_b is then obtained by letting L_B (or L_P) become very large ($L_B \to \infty$). Strictly speaking, unless the transmission is assumed to start at time $-\infty$ and end at time $+\infty$, this probability is a function of the discrete time j at which the error event starts. We assume that the transmission is long enough that we may disregard this difficulty [13].

The transmitted path through the trellis (the correct path) is described by the pairs $(\mathrm{a}'_j, \mathrm{s}_j), j = 0, 1, 2, \ldots, L_B - 1$, where a'_j and s_j are the information symbol and state number at level j respectively. Similarly, the decoders path through the trellis is described by the pairs $(\hat{\mathrm{a}}'_j, \hat{\mathrm{s}}_j)$.

Definition 5.1 *An error event that starts at level j and ends at level l with length L is defined as*

1. $\hat{\mathrm{a}}'_j \neq \mathrm{a}'_j$ *and* $\hat{\mathrm{s}}_j = \mathrm{s}_j$.

2. $\mathrm{s}_i \neq \hat{\mathrm{s}}_i$ *for* $j < i < l$.

3. $\mathrm{s}_l = \hat{\mathrm{s}}_l$. *The number of intervals between level j and l are L.*

Whenever a decoding error occurs, an error event must be in progress or starting. An example where one error event occurs is given in Figure 5.2. Assume the sequences a and â generate a sequence of error events. This is shown in Figure 5.3. The random variables W_j, $j = 0, 1, \ldots, L_B - 1$, are defined in the following way. If a and â are such that an error event starts in level j, then let W_j denote the number of information bit errors given by this error event. If a and â are such that an error event does not start in level j, then let $W_j = 0$.

The total number of information bit errors, N_e, given by a and â, is

$$N_e = \sum_{j=0}^{L_B - 1} W_j \tag{5.10}$$

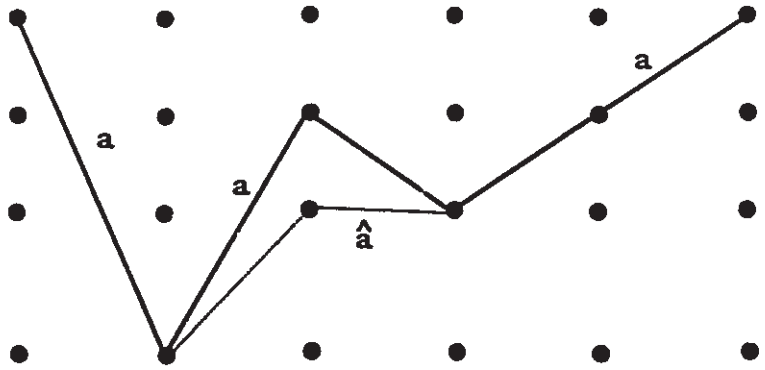


Figure 5.2: An example of error events

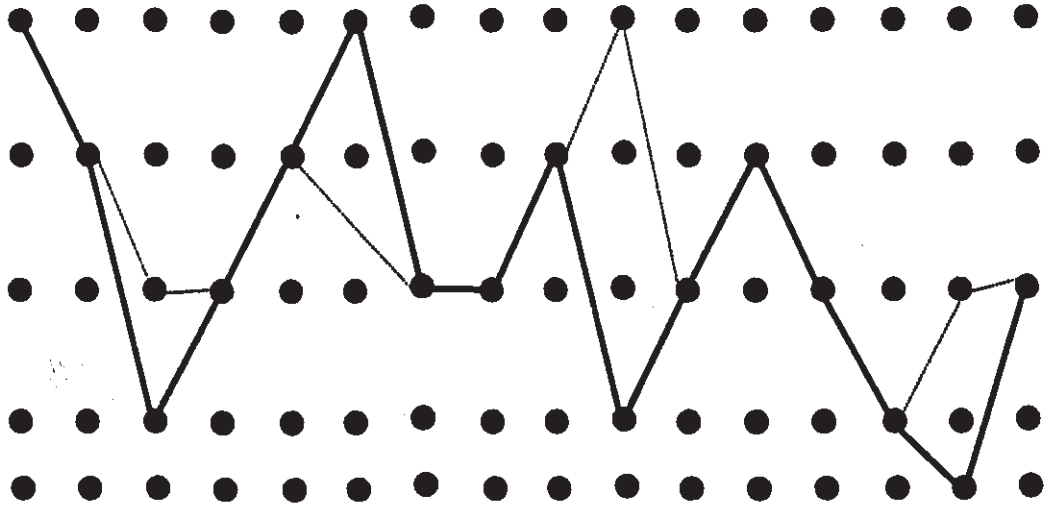


Figure 5.3: A sequence of error events

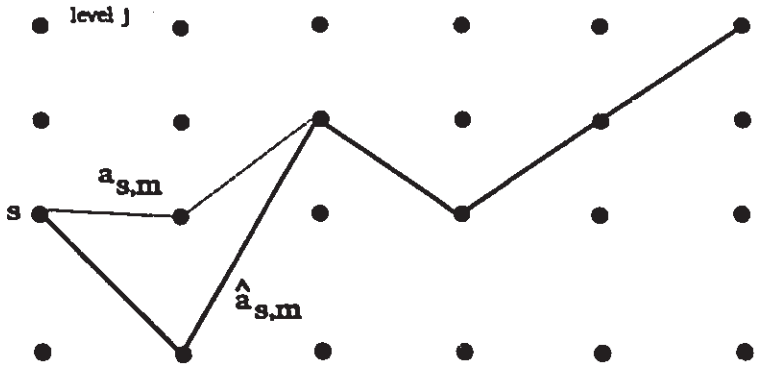


Figure 5.4: A specific error event starts at state s

and the bit error probability can be written as

$$P_b = \frac{E[N_e]}{L_P \log_2 P} = \frac{E[N_e]}{k' L_B} = \frac{1}{k' L_B} E[\sum_{j=0}^{L_B-1} W_j] = \frac{1}{k' L_B} \sum_{j=0}^{L_B-1} E[W_j]. \qquad (5.11)$$

To find $E[W_j]$, let us study the set of all error events starting in level j for which $W_j > 0$. Take an arbitrary state, s, $s \in \{1, 2, 3, \ldots, S_V\}$. All error events which start from this state are numbered $F_{s,1}, F_{s,2}, F_{s,3}, \ldots$ A specific error event, $F_{s,m}$, is completely described by its start state s and by the pair of sequences $(\mathbf{a}_{s,m}, \hat{\mathbf{a}}_{s,m})$ which generate $F_{s,m}$. The set of all error events starting in level j is then obtained by constructing $S_V - 1$ additional such lists, one list for each state. Furthermore, define $i_{s,m}$ as the number of information bit errors associated with $F_{s,m}$. Define, also, $l_{s,m}$ as the length (in intervals) of $F_{s,m}$. This specific error event is visualized in Figure 5.4. The normalized squared Euclidean distance associated with the specific error event $F_{s,m}$ is $d^2_{s,m}$. So far, we have studied the set of all error events starting in level j for which $W_j > 0$. Each specific member in this set, say $F_{s,m}$, ($s = 1, 2, \ldots, S_V; m = 1, 2, \ldots$,(m is finite)) is (partially) characterized by the three parameters $i_{s,m}, l_{s,m}$ and $d^2_{s,m}$ respectively.

Our goal is to upper bound E[W_j] in (5.11). To be able to do this we define the events $f_{j,s,m}$ and $f'_{j,s,m}$, $j = 0, 1, \ldots, L_B - 1$, $s = 1, 2, \ldots, S_V$, $m = 1, 2, \ldots$, (m finite), by

- $f_{j,s,m}$ = the event that a is such that s equals state number s, and $(\mathbf{a}'_j, \mathbf{a}'_{j+1}, \mathbf{a}'_{j+l_{s,m}-1})$

$= \mathbf{a}_{s,m}$.

- $f'_{j,s,m}$ = the event that $\hat{\mathbf{a}}'$ is such that $\hat{\mathbf{s}}$ equals state number s, and $(\hat{\mathbf{a}}'_j, \hat{\mathbf{a}}'_{j+1}, \hat{\mathbf{a}}'_{j+l_{s,m}-1})$ $= \hat{\mathbf{a}}_{s,m}$.

Since the information bits are assumed to be independent and equally likely, we have

$$Pr\{f_{j,s,m}\} = Pr\{\mathbf{s}_j = s\} \cdot (\frac{1}{2})^{k' l_{s,m}} \tag{5.12}$$

The joint event $(f_{j,s,m}, f'_{j,s,m})$ exactly describes the situation when the specific error events $F_{s,m}$ actually occurs, starting at level j. We can apply the union bound to find $E[W_j]$ [39], giving

$$\begin{aligned} E[W_j] &= \sum_i i Pr\{W_j = i\} \\ &\leq \sum_s \sum_m i_{s,m} Pr\{f_{j,s,m}, f'_{j,s,m}\} \\ &= \sum_s \sum_m i_{s,m} Pr\{f_{j,s,m}\} \cdot Pr\{f'_{j,s,m} | f_{j,s,m}\}. \end{aligned} \tag{5.13}$$

Using the fact that [56, 39]

$$Pr\{f'_{j,s,m} | f_{j,s,m}\} = Q(\sqrt{\frac{d^2_{s,m} E_b}{N_0}}), \tag{5.14}$$

we substitute (5.14) into (5.13), and have

$$E[W_j] \leq \sum_s Pr\{\mathbf{s}_j = s\} \sum_m i_{s,m} \cdot (\frac{1}{2})^{k' l_{s,m}} \cdot Q(\sqrt{\frac{d^2_{s,m} E_b}{N_0}}). \tag{5.15}$$

$Q(x)$, called error function, is defined by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2} dy. \tag{5.16}$$

We can rewrite (5.15) in terms of the variables i, l, d as

$$E[W_j] \leq \sum_s Pr\{\mathbf{s}_j = s\} \cdot \sum_i \sum_l \sum_d a(s,i,l,d) \cdot i \cdot (\frac{1}{2})^{k' l} \cdot Q(\sqrt{\frac{d^2 E_b}{N_0}}), \tag{5.17}$$

where $a(s,i,l,d)$ is the number of error events starting from state s, assuming the trellis is infinite, having i information bit errors, length $k'l$ information bits and normalized squared Euclidean distance d^2.

Assume the input symbols are independent and identically distributed. Then the probability for each state is $1/S_V$ and we can rewrite $E[W_j]$ as

$$E[W_j] \leq \frac{1}{S_V} \sum_s \sum_i \sum_l \sum_d a(s,i,l,d) \cdot i \cdot (\frac{1}{2})^{k'l} \cdot Q(\sqrt{\frac{d^2 E_b}{N_0}}). \tag{5.18}$$

Since we have assumed the sequence is sufficient long ($L_B \rightarrow \infty$), we can let (see [38])

$$E[W_j] = E[W], \ \forall j. \tag{5.19}$$

Therefore,

$$P_b = \frac{1}{k'} E[W_j] \tag{5.20}$$

and substituting $E[W_j]$ into (5.20),

$$P_b \leq \frac{1}{k' S_V} \sum_s \sum_i \sum_l \sum_d a(s,i,l,d) \cdot i \cdot (\frac{1}{2})^{k'l} \cdot Q(\sqrt{\frac{d^2 E_b}{N_0}}), \tag{5.21}$$

we can rewrite the above equation as

$$P_b \leq \sum_d C_d \cdot Q(\sqrt{\frac{d^2 E_b}{N_0}}) \tag{5.22}$$

where

$$C_d = \frac{1}{k' S_V} \sum_s \sum_i \sum_l a(s,i,l,d) \cdot i \cdot (\frac{1}{2})^{k'l}. \tag{5.23}$$

From the above equation, it can be seen that, for large SNR, the upper bound on the bit error probability is dominated by the first term in the sum and

$$P_b \leq \sum_{d \geq d_{min}} C_d Q(\sqrt{\frac{d^2 E_b}{N_0}}) = C_{d_{min}} Q(\sqrt{\frac{d_{min}^2 E_b}{N_0}}) + \text{other terms}, \tag{5.24}$$

where the summation is over all Euclidean distances in the set of all error events. d_{min}^2 is the smallest of these distances and is referred to as the normalized minimum squared Euclidean distance. $C_{d_{min}}$ is called the *error coefficient*. When E_b/N_0 is large, the other terms are negligible and we can approximate P_b as

$$P_b \approx C_{d_{min}} Q(\sqrt{\frac{d_{min}^2 E_b}{N_0}}) \tag{5.25}$$

If $C_{d_{min}}$ is not too large, then

$$P_b \approx Q(\sqrt{\frac{d_{min}^2 E_b}{N_0}}) \tag{5.26}$$

We use (5.26) in Chapter 4 to compare coded CPFSK results.

By using the inequality [39]

$$Q(\sqrt{x+y}) \leq Q(\sqrt{x}) \cdot e^{-y/2} \quad x \geq 0 \, y \geq 0, \tag{5.27}$$

we can rewrite (5.21) by letting $x = d_{min}^2 E_b/N_0$ and $y = (d^2 - d_{min}^2)E_b/N_0$. This yields

$$P_b \leq \frac{1}{k' S_V} Q(\sqrt{\frac{d_{min}^2 E_b}{N_0}}) \cdot e^{\frac{d_{min}^2 E_b}{2N_0}} \cdot \sum_s \sum_i \sum_l \sum_d a(s,i,l,d) \cdot i \cdot (\frac{1}{2})^{k'l} \cdot e^{-\frac{d^2 E_b}{N_0}}. \tag{5.28}$$

We define

$$T(D,L,I) \equiv \sum_s \sum_i \sum_l \sum_d a(s,i,l,d) I^i L^l D^{d^2}, \tag{5.29}$$

where we use the dummy variables $I = 1$, $L = (1/2)^{k'}$ and $D = e^{-E_b/N_0}$. P_b can now be upper bounded by

$$P_b \leq P_u = \frac{1}{k'} Q(\sqrt{d_{min}^2 E_b/N_0}) e^{d_{min}^2 E_b/2N_0} \cdot \frac{\partial T(D,L,I)}{\partial I} \Bigg|_{\begin{array}{l} D = e^{-E_b/2N_0} \\ L = (1/2)^{k'} \\ I = 1 \end{array}} \tag{5.30}$$

where the $T(D,L,I)$, called average transfer function, is

$$T(D,L,I) = \frac{1}{S_V} \sum_{s=1}^{S_V} T_s(D,L,I) = \frac{1}{S_V} \sum_{s=1}^{S_V} \sum_i \sum_{l'} \sum_d a(S_s,i,l',d) D^{d^2} L^{l'} I^i. \tag{5.31}$$

$T_s(D,L,I)$ represents the error events starting from any of the states $s, s \in 1,2,\ldots,S_V$. To find $T_s(D,L,I)$, we use a method that generates the parameters d^2,l', and i for all error events starting from a given state. The method we use is a so-called *super state* description [13, 42, 43, 59]. Thus, the BER for coded CPFSK systems with Viterbi detection can be upper-bounded by using the trellis description for the combined coding and modulation scheme. The resulting bound is obtained from an averaged transfer function. Convolutional codes yield a linear system and no averaging is required for the conventional BER upper bounds.

5.3 Average Transfer Function Techniques

To find $T(D, L, I)$ in order to generate the parameters d^2, l and i for all error events, given a specific start state, we use a method called the *super-state trellis* description. A specific pair of sequences, $(\mathbf{a}, \hat{\mathbf{a}})$, is then represented as a specific path in the super-state trellis. Furthermore, each transition in the pair-state trellis carries the information $D^{\Delta d^2} L I^{\Delta i}$, where Δd^2 and Δi are the contribution to d^2 and i respectively, given by that specific transition.

The pair-state at level j, Ω_j, is defined by

$$\Omega_j = (\mathbf{s}_j, \hat{\mathbf{s}}_j). \tag{5.32}$$

Let N_Ω denote the number of pair-states. Then

$$N_\Omega = S_V^2. \tag{5.33}$$

The input at level j for the pair state Ω_j is $(\mathbf{a}'_j, \hat{\mathbf{a}}'_j)$. Therefore, the number of transitions leaving Ω_j equals $2^{2k'}$.

Definition 5.2 Start states: *An error event can start from the pair-state Ω_0 if and only if Ω_0 has the structure*

$$\Omega_0 = (\mathbf{s}_0, \mathbf{s}_0) \tag{5.34}$$

There are, therefore, $N_S = S_V$ such states. These N_S pair-states are referred to as start states and denoted by α_i, $i = 1, 2, \ldots, N_S$.

Definition 5.3 Merge states: *An error event can end in the pair-state Ω_j $(j \geq 1)$ if and only if Ω_j has the structure*

$$\Omega_j = (\mathbf{s}_j, \mathbf{s}_j) \tag{5.35}$$

There are, therefore, $N_M = S_V$ such states. These N_M pair-states are referred to as merge states and denoted by γ_i, $i = 1, 2, \ldots, N_M$.

Definition 5.4 Intermediate states: *Those states Ω_j $(j \geq 1)$ which are neither start states nor merge states are referred to as intermediate states and denoted by β_i. $i = 1, 2, \ldots, N_I$; $N_I = N_\Omega - N_M$.*

An error event of length l intervals is equivalent to a path in the pair-state trellis with the following properties:

1. The path starts from a start state in level 0.
2. $a_0' \neq \hat{a}_0'$.
3. The path is in a merge state for the first time in level l. We say that the path ends in level l.

The state sequence for an error event $(\Omega_0, \Omega_1, \ldots, \Omega_l)$, consequently, has the properties that $\Omega_0 \in \{\alpha_i\}$, $\Omega_j \in \{\beta_j\}$ $\forall j$ satisfying $0 < j < l$, and $\Omega_l \in \{\gamma_i\}$. Below we will study the set of all such paths. Let $B^j_{m,z}$ denote the generating function for all paths starting in start state α_z, and are in the intermediate state β_m in level j. Thus

$$B^j_{m,z} = \sum_d \sum_i f_m(\alpha_z, i, j, d) D^{d^2} L^j I^i, \tag{5.36}$$

where $f_m(\alpha_z, i, j, d)$ = the number of paths, starting in start state α_z and which are in the intermediate state β_m in level j, having (1) the normalized squared Euclidean distance d^2, and (2) i information bit errors.

Now assume that it is possible to go from the intermediate state β_l to the intermediate state β_m in one step, as in Figure 5.5, where we use *beta* to represent β in the trellis of the overall encoder. In general, there may be more than one transition from β_l to β_m in the single step. Each such transition, with corresponding Δd^2 and Δi, contributes $D^{\Delta d^2} L I^{\Delta i} \cdot B^{j-1}_{m,z}$ to the transfer function. The total contribution to $B^j_{m,z}$, from the intermediate state β_l, can therefore be written as $A_{m,l} \cdot B^{j-1}_{l,z}$, where the $A_{m,l}$ is the sum of all the terms $D^{\Delta d^2} L I^{\Delta i}$, one term for each transition from β_l to β_m in single step. Summing over all intermediate states, for $j \geq 2$, yields

$$B^j_{m,z} = \sum_{l=1}^{N_I} A_{m,l} B^{j-1}_{l,z}. \tag{5.37}$$

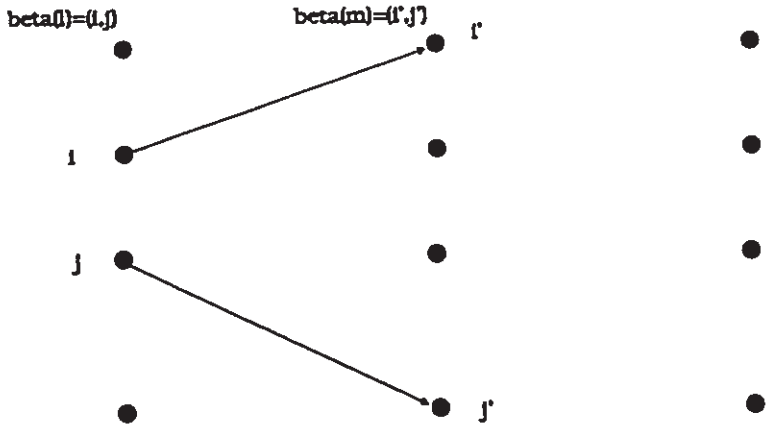


Figure 5.5: Error event in one step

By using matrix notation, (5.37) becomes

$$\mathbf{B}_z^j = \mathbf{A} \cdot \mathbf{B}_z^{j-1}; j \geq 2, \tag{5.38}$$

where $\mathbf{B}_z^j$ is a column vector with N_I elements. and $\mathbf{A}$ is an $N_I \times N_I$ matrix with elements $A_{m,l}$. It is seen that row number m of $\mathbf{A}$ represents all transitions to β_m from the intermediate state in a single step. Furthermore, column number l of $\mathbf{A}$ represents all transitions from β_l that goes to intermediate states in a single step. It is also seen that the matrix $\mathbf{A}$ does not depend on the start state α_z.

From (5.38), it follows that

$$\mathbf{B}_z^j = \mathbf{A}^{j-1} \cdot \mathbf{B}_z; j \geq 1, \tag{5.39}$$

where the column vector $\mathbf{B}_z$ represents all transitions from start state α_z to the intermediate states in one step. That is, $B_{m,z}$ is the sum of all terms $D^{\Delta d^2} LI^{\Delta i}$, one term for each transition from start state α_z to the intermediate state β_m in a single step.

Denote the transfer function for all error events starting in start state α_z and which ends in merge state γ_m in level j by $X_{m,z}^j$ $(m = 1, 2, \ldots, N_M)$. In analogy with equation (5.38) we have

$$X_{m,z}^j = \sum_{l=1}^{N_I} C_{m,l} B_{l,z}^{j-1}; j \geq 2, \tag{5.40}$$

where $C_{m,l}$ equals the sum of all terms $D^{\Delta d^2} LI^{\Delta i}$, one term for each transition from the intermediate state β_l to the merge state γ_m in a single step. Using matrix notation, (5.40) becomes

$$\mathbf{X}_z^j = \mathbf{C} \cdot \mathbf{B}_z^{j-1}; j \geq 2, \tag{5.41}$$

where $\mathbf{X}_z^j$ is a column vector with N_M elements. The matrix $\mathbf{C}$ is an $N_M \times N_M$ matrix with the elements $C_{m,l}$. It is seen that row number m in $\mathbf{C}$ represents all transitions to merge state γ_m from the intermediate states in a single step. Furthermore, column number l in $\mathbf{C}$ represents all transitions from the intermediate state β_l that goes to merge states in a single step. It is also seen that the matrix $\mathbf{C}$ does not depend on the start state α_z. Substituting (5.39) in equation (5.41) yields

$$\mathbf{X}_z^j = \begin{cases} \mathbf{C} \cdot \mathbf{B}_z^{j-1} = \mathbf{C} \cdot \mathbf{A}^{j-2} \cdot \mathbf{B}_z & ; \ j \geq 2 \\ \mathbf{D}_z & ; \ j = 1 \end{cases} \tag{5.42}$$

The column vector $\mathbf{D}_z$ represents all error events of length 1 interval, starting from start state α_z. Element number m in $\mathbf{D}_z$ is $D_{m,z}$, and it equals the sum of all terms $D^{\Delta d^2} LI^{\Delta i}$, one term for each transition from start state α_z to merge state γ_m in a single step. As we know, $\mathbf{X}_z^j$ represents all error events of length j intervals, starting from start state α_z. Since we are interested in all error events, starting from start state α_z, we define the column vector

$$\begin{aligned} \bar{\mathbf{X}}_z &= \lim_{l \to \infty} \sum_{j=1}^{l} \mathbf{X}_z^j \\ &= \mathbf{D}_z + \mathbf{C}\mathbf{B}_z + \mathbf{C}\mathbf{A}\mathbf{B}_z + \mathbf{C}\mathbf{A}^2\mathbf{B}_z + \mathbf{C}\mathbf{A}^3\mathbf{B}_z + \ldots \\ &= \mathbf{D}_z + \mathbf{C}(\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \ldots)\mathbf{B}_z \\ &= \mathbf{D}_z + \mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}_z, \end{aligned} \tag{5.43}$$

where $\mathbf{I}$ is the identity matrix. The average generating function, $T(D, L, I)$, is then given by adding together terms from $z = 1$ to $z = S_V$,

$$T(D, L, I) = \frac{1}{S_V} \cdot \sum_{z=1}^{N_S} \bar{\mathbf{X}}_z = \frac{1}{S_V} \cdot \sum_{z=1}^{S_V} \bar{\mathbf{X}}_z \tag{5.44}$$

Representing this in matrix form

$$T(D,L,I) = \frac{1}{S_V} \cdot \mathbf{1} \cdot (\mathbf{C}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{B} + \mathbf{D}) \tag{5.45}$$

where

$$\mathbf{1} = [1, 1, \ldots, 1], \tag{5.46}$$

$$\mathbf{B} = \sum_z \mathbf{B}_z, \tag{5.47}$$

and

$$\mathbf{D} = \sum_z \mathbf{D}_z. \tag{5.48}$$

5.4 Two Examples: Binary and Quaternary CPFSK with $h = 1/2$

5.4.1 MSK (minimum shift keying) and DMSK (differential minimum shift keying)

Binary CPFSK with modulation index $h = 1/2$ is called minimum shift keying (MSK) [8]. MSK has a simple transmitter structure and can be decoded with MLSE in two decoding intervals. It has a narrow bandwidth requirement and is constant envelope. These properties make it a very useful modulation scheme.

Minimum shift keying can be decomposed into a rate-1/2 systematic convolutional encoder and a memoryless mapper, shown in Figure 5.6. The MM consists of four different channel symbols. DMSK is the feedback-free form of MSK, i.e., a precoded MSK (see Chapter 2). In [26], it is shown that the bit error probability of MSK is twice that of DMSK if it employes the MLSE receiver. To apply the BER upper bound in evaluating the error coefficients of MSK and DMSK, we will see that the $C_{d_{min}}$ for MSK is two and for DMSK it is only one. This is consistent with the previous work of [26].

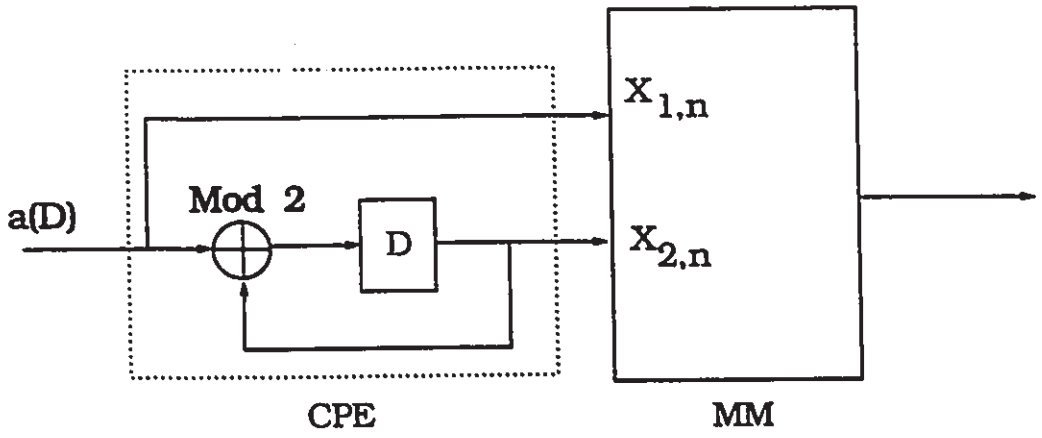


Figure 5.6: Minimum Shift Keying

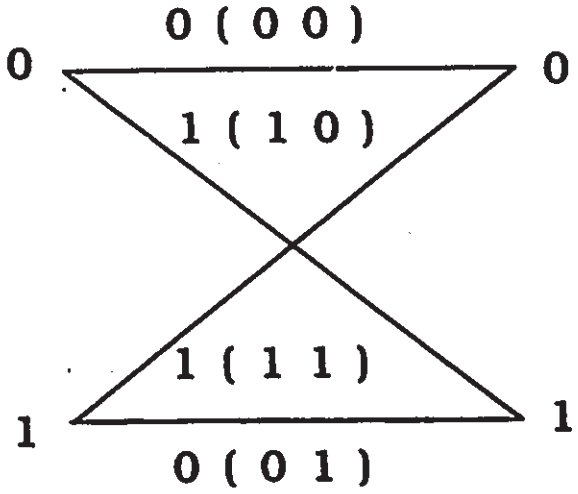


Figure 5.7: MSK trellis diagram

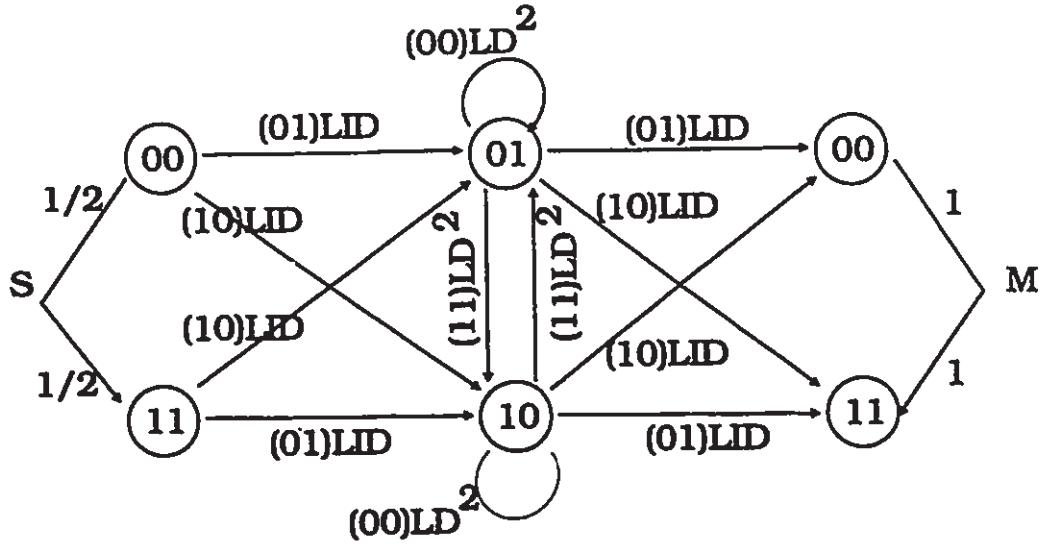


Figure 5.8: MSK super-trellis state digram

Now we will derive P_u for MSK. To draw the super state diagram for MSK, we first study the trellis diagram for MSK, shown in Figure 5.7. It has two states, namely 0 and 1, $S_V = 2$. From the definitions in the last section, there are two start states ($N_S = 2$),

$$\begin{aligned} \alpha_0 &= [0,0] \\ \alpha_1 &= [1,1], \end{aligned} \tag{5.49}$$

two intermediate states ($N_I = 2$)

$$\begin{aligned} \beta_0 &= [0,1] \\ \beta_1 &= [1,0], \end{aligned} \tag{5.50}$$

and two merge states ($N_M = 2$)

$$\begin{aligned} \gamma_0 &= [0,0] \\ \gamma_1 &= [1,1]. \end{aligned} \tag{5.51}$$

The super-state trellis diagram is shown in Figure 5.8. To give an example of labeling the super branch, we study the branch from α_1 to β_0, i.e. the element $C_{0,1}$, so we

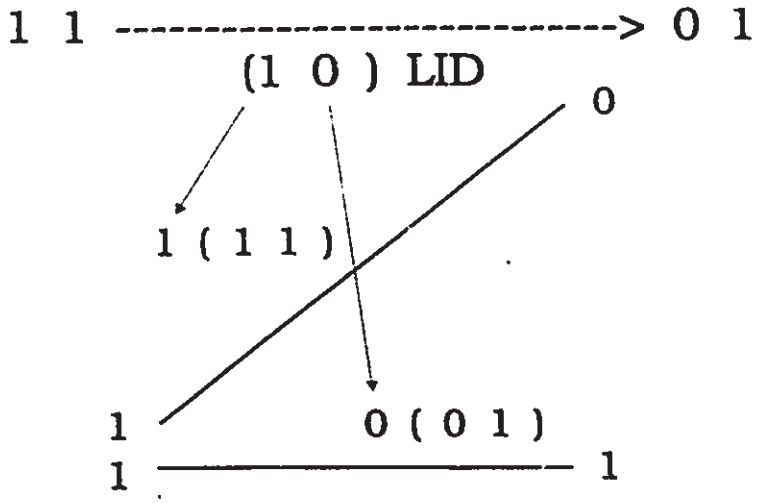


Figure 5.9: An example of super-trellis branch

have the transition from super-state [1, 1] to super-state [0, 1]. The transition related to the MSK trellis is shown in Figure 5.9. There is one bit error, I^1, in one symbol interval, L^1, and $d^2_{min} = 1$, D^1. Therefore, we label this branch as LID, where the notation (01) denotes the input symbols for the super-trellis.

Following the same step, we can find the matrix **A**, **B** and **C** respectively. We have

$$\mathbf{A} = \begin{bmatrix} LI^0D^2 & LI^0D^2 \\ LI^0D^2 & LI^0D^2 \end{bmatrix}, \tag{5.52}$$

$$\mathbf{B} = \begin{bmatrix} 2LID \\ 2LID \end{bmatrix}, \tag{5.53}$$

and

$$\mathbf{C} = \begin{bmatrix} LID & LID \\ LID & LID \end{bmatrix}. \tag{5.54}$$

There is no pair of error events with only one interval, i.e $\mathbf{D} = \mathbf{0}$. Using (5.45), we have

$$\begin{aligned} T(D, L, I) &= \frac{1}{S_V} \cdot \mathbf{1} \cdot \mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}) \\ &= \frac{1}{2} \cdot \mathbf{1} \cdot \mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}) \end{aligned} \tag{5.55}$$

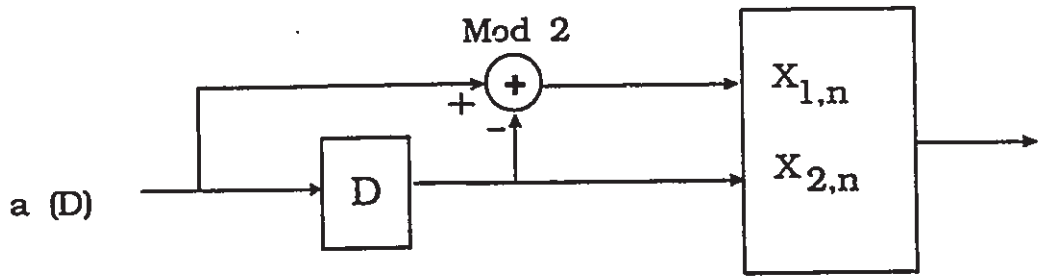


Figure 5.10: Differential MSK

By matrix multiplication,

$$T(D,L,I) = \frac{4L^2I^2D^2}{1-2LID^2} = 4L^2I^2D^2 + \text{other terms with } d_{min}^2 > 2 \tag{5.56}$$

This result is the same as the one derived in another paper [41], which describes CPFSK in a traditional way. To find the upper bound on the bit error rate one can take the partial derivative of $T(D,L,I)$ with respect to I ($k' = 1$ and $d_{min}^2 = 2$),

$$P_b \leq P_u = \frac{1}{1} \cdot Q(\sqrt{2E_b/N_0}) \cdot e^{2E_b/2N_0} \cdot \frac{\partial T(D,L,I)}{\partial I}\Bigg|_{\begin{array}{l} D = e^{-E_b/2N_0} \\ L = (1/2)^1 \\ I = 1 \end{array}} \tag{5.57}$$

We have,

$$\begin{aligned} P_b &\leq 8 \cdot L^2 \cdot Q(\sqrt{\frac{2E_b}{N_0}}) + \text{other terms} \\ &= 2 \cdot Q(\sqrt{\frac{2E_b}{N_0}}) + \text{other terms,} \end{aligned} \tag{5.58}$$

where $L = (1/2)$, and we have obtained $C_{d_{min}}$,

$$C_{d_{min}} = 2 \text{ for MSK} \tag{5.59}$$

We now study the super-state trellis for DMSK. The DMSK and its trellis diagram are shown in Figure 5.10 and Figure 5.11 respectively. In a similar way to the derivation for MSK, we obtain the matrices

$$\mathbf{A} = \begin{bmatrix} LI^1D^2 & LI^1D^2 \\ LI^1D^2 & LI^1D^2 \end{bmatrix}, \tag{5.60}$$

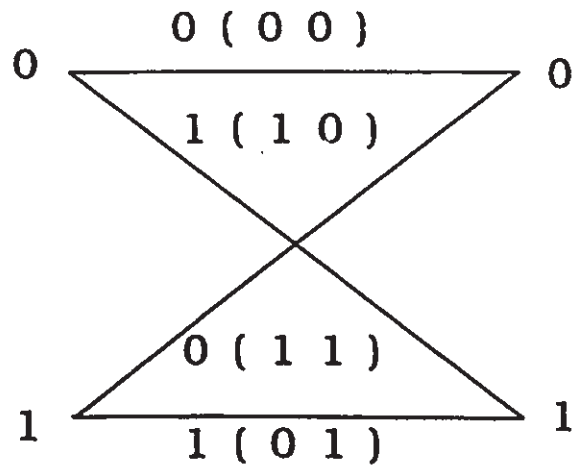


Figure 5.11: Trellis diagram for differential MSK

$$\mathbf{B} = \begin{bmatrix} 2LID \\ 2LID \end{bmatrix}, \tag{5.61}$$

and

$$\mathbf{C} = \begin{bmatrix} LD & LD \\ LD & LD \end{bmatrix}. \tag{5.62}$$

We obtain $T(D, L, I)$ as

$$\begin{aligned} T(D, L, I) &= \frac{4L^2ID^2}{1 - 2LID^2} \\ &= 4L^2ID^2 + \text{other terms with } d_{min}^2 > 2 \end{aligned} \tag{5.63}$$

It should be noted that, when $d_{min}^2 = 2$, the exponent of I is only one. The upper bound P_u can be obtained the same way as for MSK.

$$\begin{aligned} P_b &\leq 4 \cdot L^2 \cdot Q(\sqrt{\frac{2E_b}{N_0}}) + \text{other terms} \\ &= 1 \cdot Q(\sqrt{\frac{2E_b}{N_0}}) + \text{other terms} \end{aligned} \tag{5.64}$$

where $L = (1/2)$, and we have obtained C_{dmin} as

$$C_{d_{min}} = 1 \text{ for DMSK} \tag{5.65}$$

Therefore, the bit error probability of MSK is two times larger than that of DMSK for practical SNR.

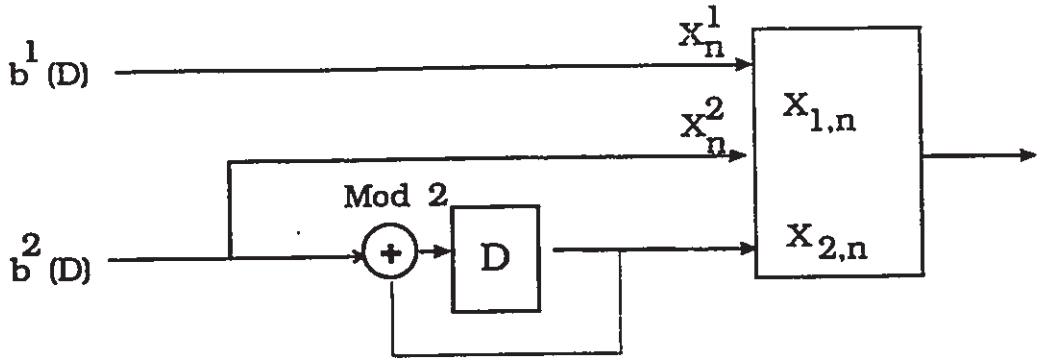


Figure 5.12: Decomposition model of quaternary CPFSK with $h = 1/2$.

5.4.2 Quaternary CPFSK and Feedback-Free CPFSK with $h = 1/2$

Quaternary CPFSK with modulation index equal to 1/2 has a normalized squared distance equal to 2 and transmits two bits per symbol interval. Quaternary CPFSK has the largest normalized squared distance [8]. It can be viewed as a rate-2/3 systematic convolutional encoder connected to a memoryless mapper. The MM consists of eight different channel symbols (as shown in Figure 5.12). The feedback-free form of CPFSK is a precoder connected with CPFSK (refer to Chapter 2). From [26] and the last paragraph, it is shown that the bit error probability of DMSK is better than that of MSK. We may ask whether this also happens between feedback-free CPFSK and CPFSK. In [27] and [49], it is argued that it may be like the differential encoded PSK and PSK. In this subsection, we apply the BER upper bound to find the error coefficient of quaternary CPFSK and feedback-free CPFSK with modulation index 1/2. We derive for this case that $C_{d_{min}}$ for CPFSK and feedback-free CPFSK have the same error coefficient. In the next section we apply simulation techniques to other examples and will see that, for the case considered in this thesis, the $C_{d_{min}}$ for CPFSK is always greater than, but in some cases may *equal*, feedback-free CPFSK.

We are now ready to derive P_u for quaternary CPFSK with $h = 1/2$. We can draw the super state diagram for this modulator from the trellis diagram of quaternary

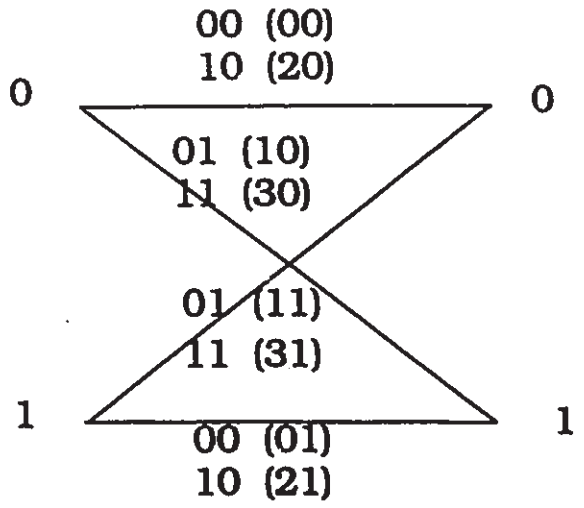


Figure 5.13: Trellis diagram of quaternary CPFSK with $h = 1/2$ trellis diagram

CPFSK with $h = 1/2$, shown in Figure 5.13. It has two states, namely 0 and 1 ($S_V = 2$). Four branches are emitted for each state. We label each branch with its input bits and output symbols (one quaternary and the other is binary). From the definitions of the previous sections, it can be seen that there are two start states ($N_S = 2$),

$$\begin{aligned} \alpha_0 &= [0,0] \\ \alpha_1 &= [1,1], \end{aligned} \tag{5.66}$$

two intermediate states ($N_I = 2$),

$$\begin{aligned} \beta_0 &= [0,1] \\ \beta_1 &= [1,0], \end{aligned} \tag{5.67}$$

and two merge states ($N_M = 2$),

$$\begin{aligned} \gamma_0 &= [0,0] \\ \gamma_1 &= [1,1]. \end{aligned} \tag{5.68}$$

The super-state trellis diagram is shown in Figure 5.14. As an example for labeling the

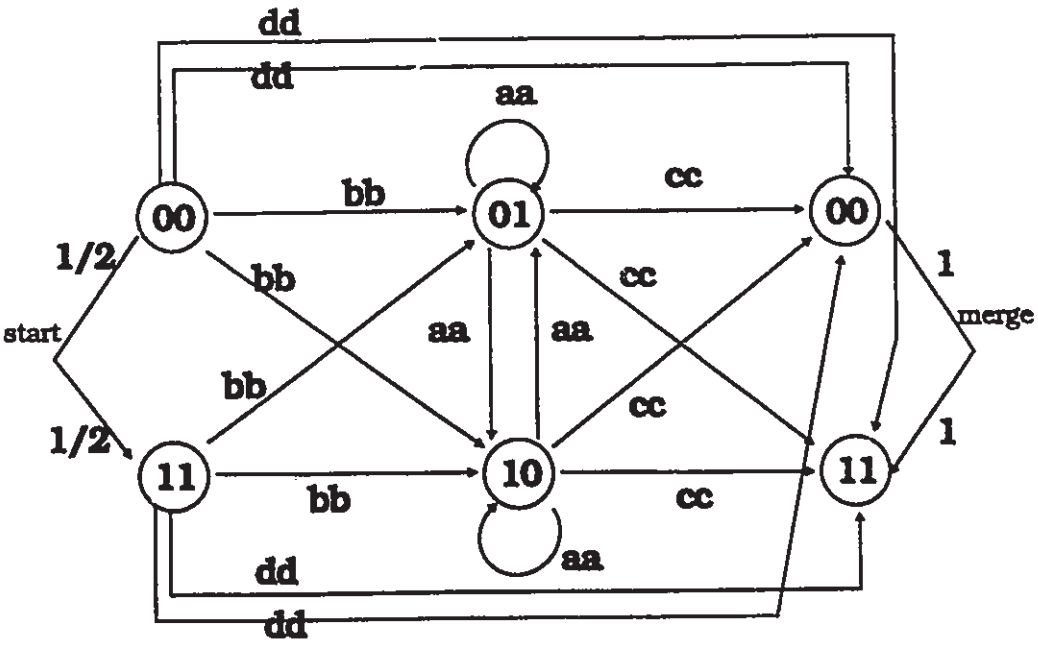


Figure 5.14: Super-trellis state diagram

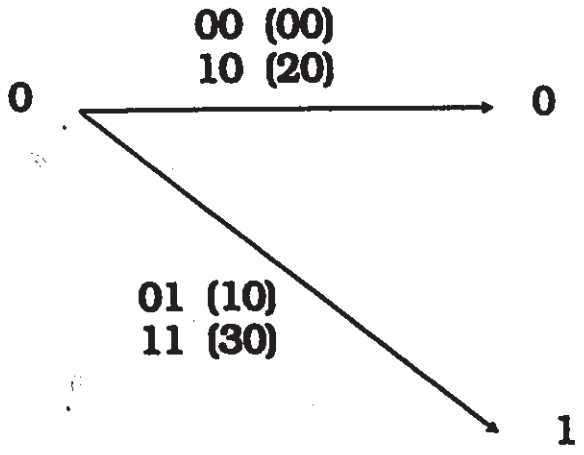


Figure 5.15: Super branch calculation

super branch, we study the branch from α_0 to β_1, i.e. the element $C_{0,1}$, and we have the transition from super-state $[0,0]$ to $[1,0]$. The transition is shown in Figure 5.15. Four transitions can occur in this branch. That is, $B_{0,1} = LID + LI^2D + LI^2D + LID$. In Figure 5.14, aa is $2LID^2 + 2LD^4$, bb is $2LID^2 + 2LI^2D^2$, cc is $LID^2 + LI^2D^2$, and dd is LID^2.

Following the same steps as derivations for MSK, we can generate the matrix **A**, **B** and **C** respectively. We obtain

$$\mathbf{A} = \begin{bmatrix} 2LID^2 + 2LD^4 & 2LID^2 + 2LD^4 \\ 2LID^2 + 2LD^4 & 2LID^2 + 2LD^4 \end{bmatrix}, \tag{5.69}$$

$$\mathbf{B} = \begin{bmatrix} 4LID^2 + 4LI^2D^2 \\ 4LID^2 + 4LI^2D^2 \end{bmatrix}, \tag{5.70}$$

$$\mathbf{C} = \begin{bmatrix} 2LID^2 + 2LI^2D^2 & 2LID^2 + 2LI^2D^2 \\ 2LID^2 + 2LI^2D^2 & 2LID^2 + 2LI^2D^2 \end{bmatrix}, \tag{5.71}$$

and

$$\mathbf{D} = \begin{bmatrix} 2LID^2 \\ 2LID^2 \end{bmatrix}. \tag{5.72}$$

Using (5.45), we have

$$\begin{aligned} T(D,L,I) &= \frac{1}{S_V} \cdot \mathbf{1} \cdot (\mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}) \\ &= \frac{1}{2} \cdot \mathbf{1} \cdot (\mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}) \end{aligned} \tag{5.73}$$

By matrix multiplication, we have

$$T(D,L,I) = 2LID^2 + \frac{2}{\Delta_1} \cdot \Delta_2 \cdot (4LID^2 + 4LI^2D^2) \tag{5.74}$$

where

$$\Delta_1 = 1 - 2LID^2 + 2LD^4 - 4L^2ID^6 \tag{5.75}$$

and

$$\Delta_2 = 2LID - 2LI^2D^2 - 4L^2I^2D^3 + 4L^2I^2D^4 - 4L^2ID^5 + 4L^2ID^6 \tag{5.76}$$

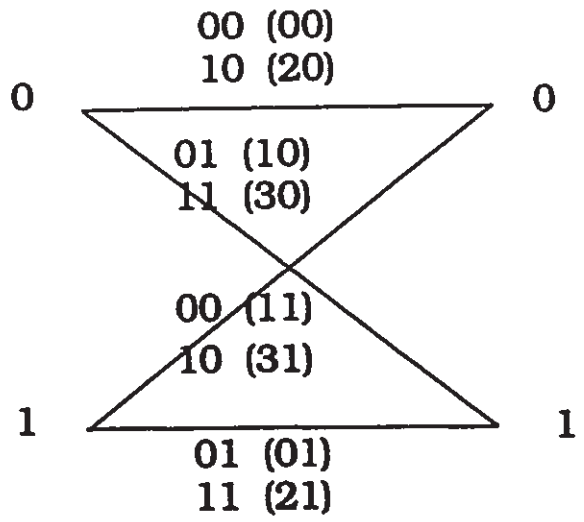


Figure 5.16: Trellis diagram for feedback-free quaternary CPFSK with $h = 1/2$

The second part of (5.74) has $d^2_{min} > 2$. Hence, to find the upper bound on the bit error rate, one can take the partial derivative of $T(D, L, I)$ with respect to I ($k' = 2$ and $d^2_{min} = 2$), giving

$$P_b \leq P_u = \frac{1}{2}Q(\sqrt{2E_b/N_0})e^{2E_b/2N_0} \cdot \left.\frac{\partial T(D,L,I)}{\partial I}\right| \begin{array}{rcl} D & = & e^{-E_b/2N_0} \\ L & = & (1/2)^2 \\ I & = & 1 \end{array} \tag{5.77}$$

We can have

$$\begin{aligned} P_b &\leq \frac{1}{k'} \cdot 2 \cdot LQ(\sqrt{\frac{2E_b}{N_0}}) + \text{other terms} \\ &= (\frac{1}{2})^2 \cdot Q(\sqrt{\frac{2E_b}{N_0}}) + \text{other terms,} \end{aligned} \tag{5.78}$$

where $L = (1/2)^2$, and we obtain $C_{d_{min}}$,

$$C_{d_{min}} = \frac{1}{4} \text{ for } h = 1/2 \text{ quaternary CPFSK} \tag{5.79}$$

Now we study the super-state trellis for feedback-free CPFSK. The feedback-free CPFSK has a trellis diagram as shown in Figure 5.16. In a similar way, we obtain the matrix

$$\mathbf{D} = \begin{bmatrix} 2LID^2 \\ 2LID^2 \end{bmatrix} \tag{5.80}$$

For computing $C_{d_{min}}$, we only need to know **D**. This is because, in this example, the distance properties of feedback-free CPFSK and CPFSK are the same, except for the coefficient LI in each term. From the derivation of the last paragraph, we know that the d^2_{min} originates from the error events of **D**. We obtain $T(D, L, I)$ as,

$$T(D, L, I) = 2LID^2 + \text{other terms with } d^2_{min} > 2. \tag{5.81}$$

It can be noted that, for $d^2_{min} = 2$, the exponent of I is only one. The upper bound P_u can be obtained analogously as for CPFSK.

$$\begin{aligned} P_b &\leq \frac{1}{2} \cdot 2 \cdot L \cdot Q(\sqrt{\frac{2E_b}{N_0}}) + \text{other terms} \\ &= \frac{1}{4} \cdot Q(\sqrt{\frac{2E_b}{N_0}}) + \text{other terms}, \end{aligned} \tag{5.82}$$

where $L = (1/2)^2$, and we have $C_{d_{min}}$ as,

$$C_{d_{min}} = \frac{1}{4} \text{ for quaternary feedback-free. CPFSK with } h = 1/2 \tag{5.83}$$

From the above calculations ,i.e., MSK vs. DMSK, feedback vs. feedback-free quaternary CPFSK with $h = 1/2$, we can see that the bit error probability of feedback-free CPFSK is better than or equals CPFSK for practical SNRs. For quaternary CPFSK with $h = 1/2$, the error coefficient is smaller than one, i.e., the bit error probability of the actual system is better than predicted by the normalized squared Euclidean distance. These are very interesting properties of CPM systems.

5.5 Numerical Computation of the Transfer Function

To compute the transfer function (5.45) is not an easy task. This is because to find the inverse of $(\mathbf{I} - \mathbf{A})$, in general, cannot calculated analytically. To calculate the transfer function numerically we can apply the following steps

$$T(D, L, I) = \mathbf{1} \cdot [\mathbf{C} \cdot (\mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{B} + \mathbf{D}]$$

$$= 1 \cdot \mathbf{D} + 1 \cdot \mathbf{C} \cdot (\mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{B}$$
$$= \mathbf{D}_p + \mathbf{C}_p \cdot (\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdot) \cdot \mathbf{B} \quad (5.84)$$

In (5.84) $\mathbf{D}_p = \mathbf{1} \cdot \mathbf{D}$ and $\mathbf{C}_p = \mathbf{1} \cdot \mathbf{C}$. We can use a finite number of $\mathbf{A}^n$ to calculate an approximation for $T(D, L, I)$. However, to directly compute this summation and then take the partial derivative with respect to I, would be prohibitively costly in memory and CPU time. Instead, we use an efficient computational approach described in [13]. By noting that the above equation can be rewritten as

$$T(D, L, I) = \mathbf{D}_p + \sum_{n=0}^{\infty} \mathbf{C}_p \cdot \mathbf{A}^n \cdot \mathbf{B}$$
$$= \mathbf{D}_p + \sum_{n=0}^{\infty} \mathbf{Z}_n \cdot \mathbf{B}, \quad (5.85)$$

where we have let

$$\mathbf{Z}_0 = \mathbf{C}_p \text{ and} \quad (5.86)$$
$$\mathbf{Z}_n = \mathbf{C}_p \cdot \mathbf{A}^n, \quad (5.87)$$

the vector $\mathbf{Z}_n$ can be found recursively from

$$\mathbf{Z}_{n+1} = \mathbf{Z}_n \cdot \mathbf{A}. \quad (5.88)$$

Taking the derivative of (5.85) with respect to I yields

$$\frac{\partial}{\partial I} T(D, L, I) = \frac{\partial \mathbf{D}_p}{\partial I} + \sum_{n=0}^{\infty} \frac{\partial \mathbf{Z}_n}{\partial I} \cdot \mathbf{B} + \sum_{n=0}^{\infty} \mathbf{Z}_n \cdot \frac{\partial \mathbf{B}}{\partial I}$$
$$= \frac{\partial \mathbf{D}_p}{\partial I} + \sum_{n=0}^{\infty} \mathbf{Y}_n \cdot \mathbf{B} + \sum_{n=0}^{\infty} \mathbf{Z}_n \cdot \frac{\partial \mathbf{B}}{\partial I}, \quad (5.89)$$

where

$$\mathbf{Y}_n = \frac{\partial \mathbf{Z}_n}{\partial I}. \quad (5.90)$$

The vector $\mathbf{Y}_n$ can be found recursively by,

$$\mathbf{Y}_{n+1} = \frac{\partial \mathbf{Z}_{n+1}}{\partial I}$$

$$
\begin{aligned}
&= \frac{\partial}{\partial I}(\mathbf{Z}_n \cdot \mathbf{A}) \\
&= (\frac{\partial}{\partial I}\mathbf{Z}_n) \cdot \mathbf{A} + \mathbf{Z}_n \cdot \frac{\partial}{\partial I}\mathbf{A} \\
&= \mathbf{Y}_n \cdot \mathbf{A} + \mathbf{Z}_n \cdot \frac{\partial}{\partial I}\mathbf{A} \qquad (5.91)
\end{aligned}
$$

with

$$\mathbf{Y}_0 = \frac{\partial}{\partial I}\mathbf{C}_p \qquad (5.92)$$

Replacing $I = 1$ and $L = (1/2)^{k'}$ we can find $T(D, L, I)$ in terms of D. After the expansion of $T(D, L, I)$, we can substitute D to find the P_u. The author has used both the expansion of $(\mathbf{I} - \mathbf{A})^{-1}$ and the approach presented in this section, and the later one is much more efficient in terms of CPU time and the memory requirements.

5.6 Numerical Results

References [26, 5] show that the bit error rate of MSK is twice that of DMSK. Here, we extend this concept with our upper bound to several feedback-free CPFSK systems. The bound is then applied to ring convolutional encoded CPFSK. Some examples are compared to the BER simulations of the systems and it shows that the bound is very tight at practical signal to noise ratios. It is concluded that the upper bound is tight at practical SNRs, and the normalized minimum Euclidean distance is an appropriate single parameter to compare the systems. For some cases, d^2_{min} is pessimistic in its prediction of the performance of ring convolutionally encoded CPFSK.

In the last section, we apply the upper bound to two modulators and their feedback-free forms. The modulators have simple trellis structures and can be analyzed with simple matrix multiplication. However, for other modulators or coded schemes, computer simulation is necessary. In this section we apply numerical techniques for some interesting examples. This results are obtained by using the expansion of $(\mathbf{I} - \mathbf{A})^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \ldots$. We choose the order of the expansion to be the point where the BER curve has sufficiently converged. Due to computer memory

restrictions, we have also restricted the number of different d^2_{min} and the coefficient LI. This will affect the BER at low signal to noise ratios. In this thesis, we keep the number of different d^2_{min} and LI sufficiently large enough to give good results for moderate and practical signal to noise ratios.

Numerical results are given in figures. Each figure gives $10\log_{10}(P_u)$ as a function of signal to noise ratio in dB. Each figure also plots $10\log_{10}(Q(\sqrt{d^2_{min}E_b/N_0}))$, since in Chapter 4 we use this as our performance criteria. This can be seen in (5.26). From [43], we can find the approximate point where the minimum Euclidean distance events dominate the upper bound [43]. Also, we use the MSK, which has $d^2_{min} = 2$, as a reference in some figures, so the function $10\log_{10}(Q(\sqrt{2E_b/N_0}))$ is also plotted.

We use the following plotting conventions.

- curves marked with "–·–·–·–·" represent the function $10\log_{10}(Q(\sqrt{2E_b/N_0}))$.
- curves marked with solid line represents the upper bound.
- curves marked with "- - - -" represents the function $10\log_{10}(Q(\sqrt{d^2_{min}E_b/N_0}))$
- curves marked with "x" represents the simulated bit error probability.

Usually, the simulated bit error probability will fall between the upper bound and curve $10\log_{10}(Q(\sqrt{d^2_{min}E_b/N_0}))$ when the SNR is moderate and small, as has been found in [43]. The numerical results show that the feedback-free CPFSK generally has an error coefficient equal to or smaller than that of CPFSK.

5.6.1 Feedback-Free CPFSK vs. CPFSK

(a) Quaternary CPFSK with h=1/4

In Figure 5.17 we show the upper bound and the line for $Q(\sqrt{d^2_{min}E_b/N_0})$, where d^2_{min} is the normalized squared Euclidean distance for the modulator under consideration. In this case $d^2_{min} = 1.45$ for quaternary CPFSK with $h = 1/4$. A similar curve can be found in [43]. We also plot $Q(\sqrt{2E_b/N_0})$ (2 is the d^2_{min} for MSK) as a

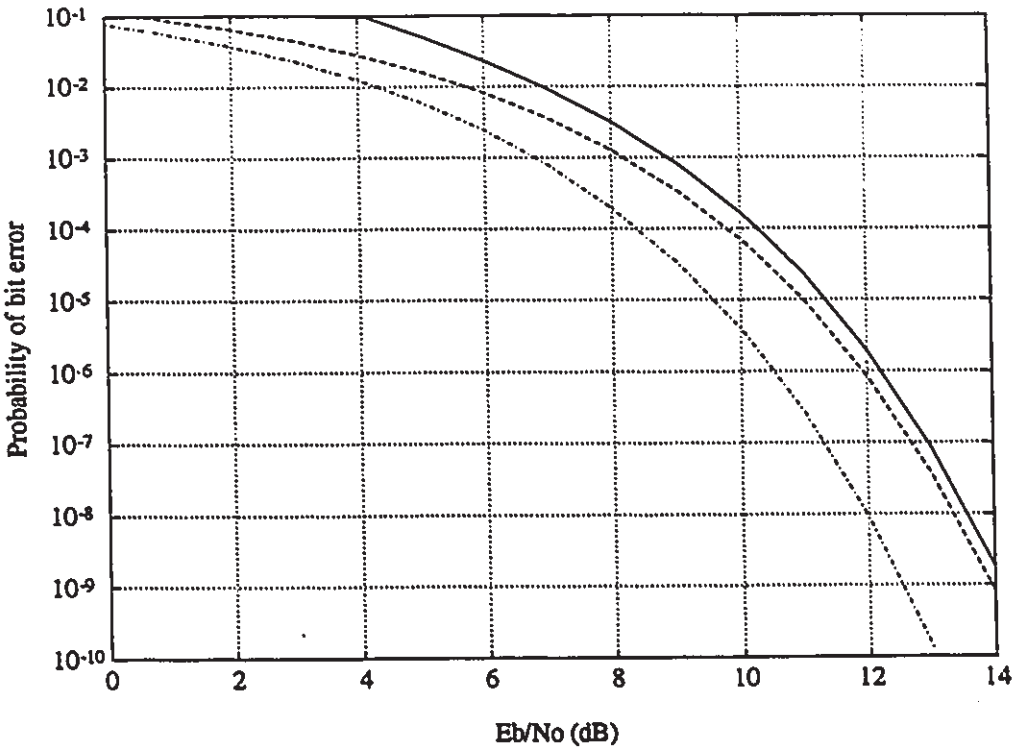


Figure 5.17: Quaternary CPFSK with $h = 1/4$. The solid line is the upper bound. The "– – –" line is the function $Q((d^2_{min}E_b/N_0)^{1/2})$. The "– · – · –" line is the function $Q((2E_b/N_0)^{1/2})$.

comparison. Using the simulation technique, we found $C_{d_{min}}$ for this modulator to be 1.5. In Figure 5.18 we show the upper bound for the feedback-free form of this modulator and the line for $Q(\sqrt{1.45E_b/N_0})$, where 1.45 is the normalized squared Euclidean distance for quaternary CPFSK with $h = 1/4$. Again, $Q(\sqrt{2E_b/N_0})$ (2 is the d^2_{min} is given for MSK) is plotted as a comparison. The error coefficient $C_{d_{min}}$ for this modulator is 0.844. Therefore, the $C_{d_{min}}$ for CPFSK is larger than that of feedback-free CPFSK.

(b) Octal CPFSK with h=1/8

Figure 5.19 shows the upper bound and the line for $Q(\sqrt{0.598E_b/N_0})$, where 0.598 is the normalized squared Euclidean distance for octal CPFSK with $h = 1/8$. The error coefficient, $C_{d_{min}}$, for this modulator is 1.604. In Figure 5.20 we show the upper

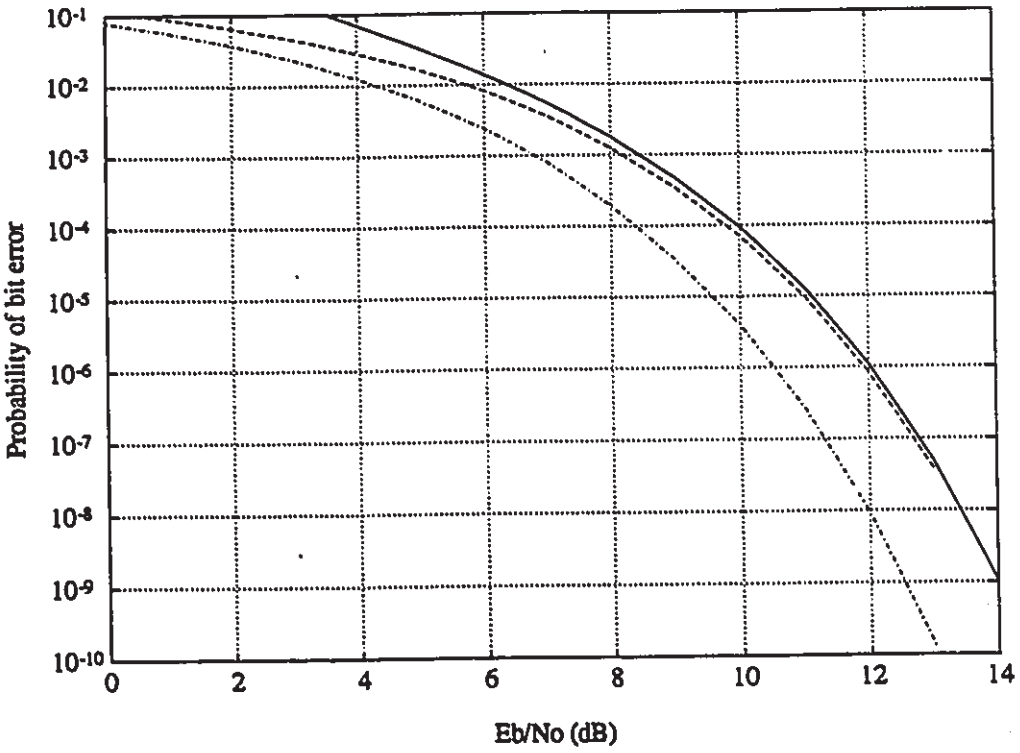


Figure 5.18: Feedback-free quaternary CPFSK with $h = 1/4$. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min}E_b/N_0)^{1/2})$. The "— · — · —" line is the function $Q((2E_b/N_0)^{1/2})$.

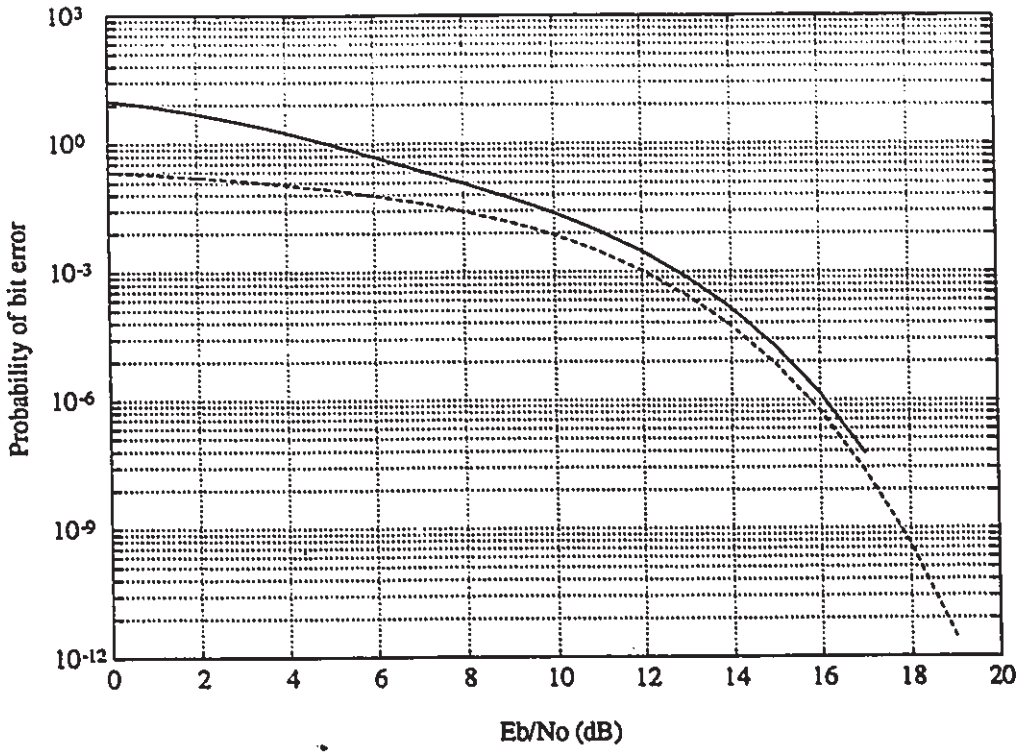


Figure 5.19: Octal CPFSK with $h = 1/8$. The solid line is the upper bound. The "- -" line is the function $Q((d^2_{min}E_b/N_0)^{1/2})$.

bound for feedback-free form of this modulator and the line for $Q(\sqrt{0.598E_b/N_0})$, where 0.598 is the normalized squared Euclidean distance for the feedback-free form of octal CPFSK with $h = 1/8$. The error coefficient, $C_{d_{min}}$, for this modulator is 0.893. The $C_{d_{min}}$ for CPFSK is larger than that of feedback-free CPFSK.

5.6.2 Rate-1/2 Encoded Quaternary CPFSK

(a) Rate 1/2 modulo-4 encoded quaternary CPFSK with $h = 1/2$

Figure 5.21 shows the BER curve for rate-1/2 modulo-4 encoded $h = 1/2$ quaternary CPFSK with $S_V = 2$. Since $d^2_{min} = 2$ and $C_{d_{min}} = 1$ for this case, the lines for $Q(\sqrt{d^2_{min}E_b/N_0})$ and MSK are the same. The upper bound is very close to the dotted line for high signal to noise ratios.

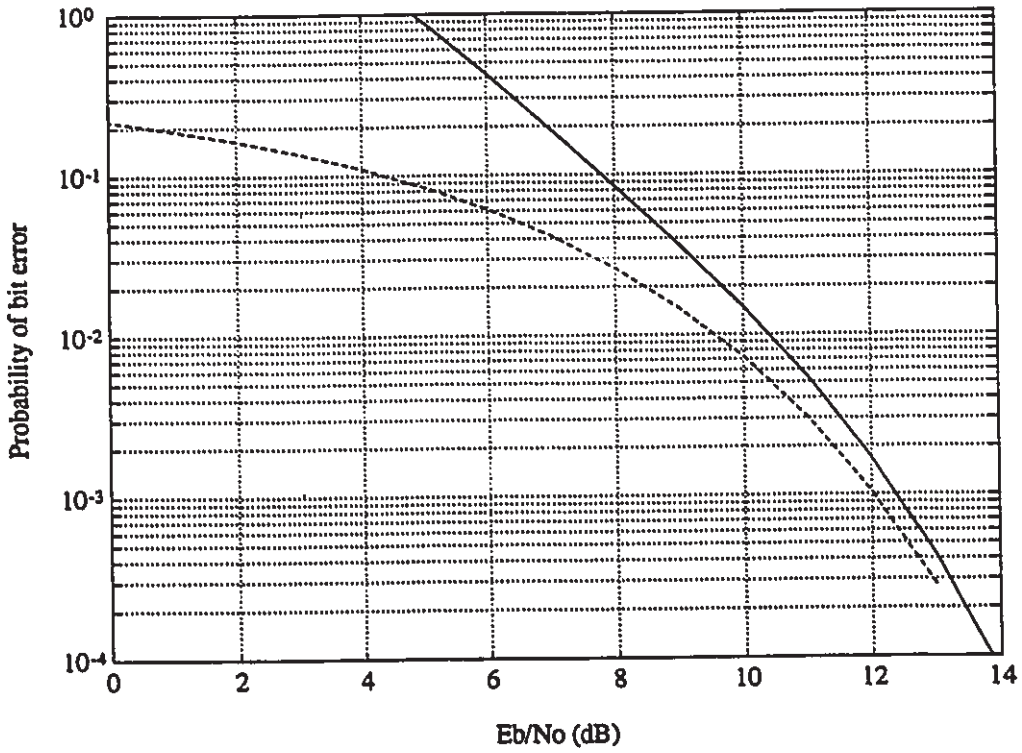


Figure 5.20: Octal feedback-free CPFSK with $h = 1/8$. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min} E_b/N_0)^{1/2})$.

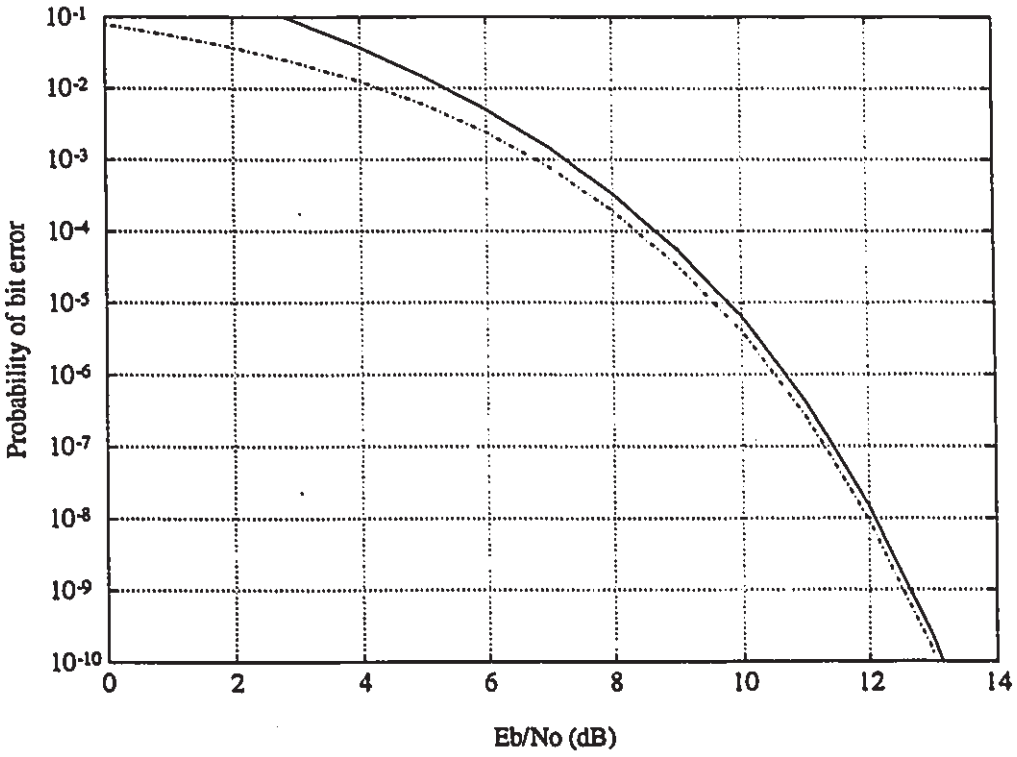


Figure 5.21: Rate-1/2 binary encoded quaternary CPFSK with $h = 1/2$, $S_V = 2$. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min} E_b/N_0)^{1/2})$. The "— · — · —" line is the function $Q((2E_b/N_0)^{1/2})$.

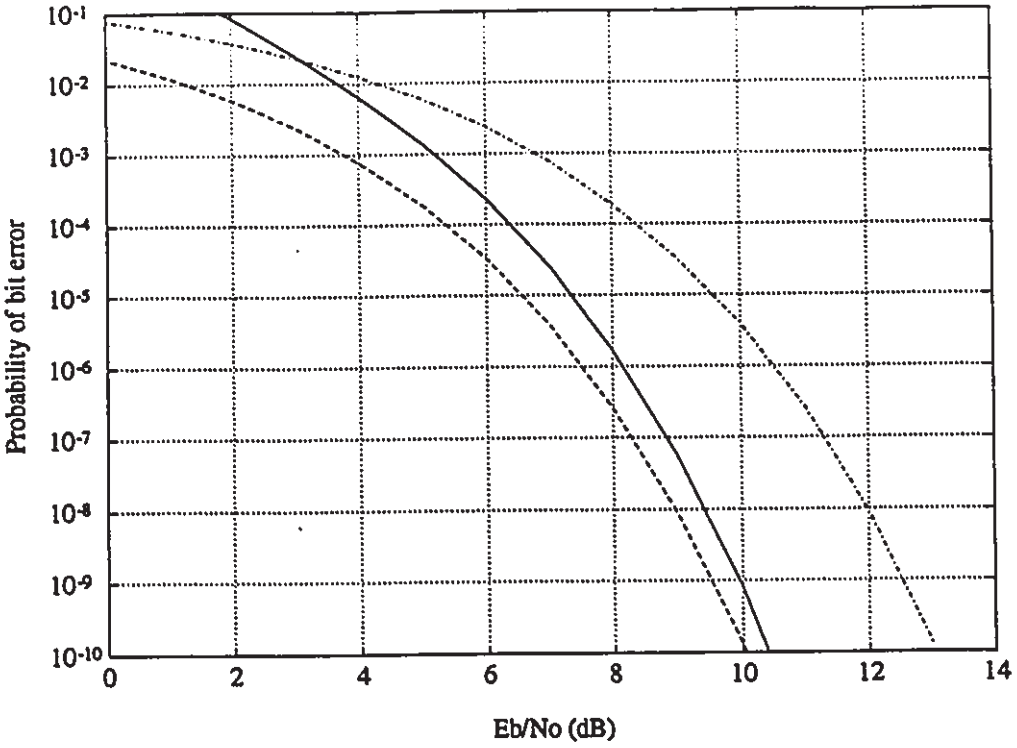


Figure 5.22: Rate-1/2 binary encoded quaternary CPFSK with $h = 1/2$, $S_V = 4$. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min}E_b/N_0)^{1/2})$. The "— · — · —" line is the function $Q((2E_b/N_0)^{1/2})$.

Figure 5.22 shows the BER curve for $S_V = 4$. The solid line is the upper bound and the line with "- - -" is the line predicted by using the $d^2_{min} = 4$. Also, we plot the line for $d^2_{min} = 2$ (the upper dotted line). The upper bound is tight and the error coefficient equals 4. It performs close to the coding gain predicted using $d^2_{min} = 4$ for practical signal to noise ratios, i.e, a 3 dB gain over MSK for a BER equal to 10^{-5}, and about 1 dB gain at a BER equal to 10^{-3}.

Figure 5.23 shows the BER curve for $S_V = 8$. The upper bound is shown as the solid line. It is tight for high signal to noise ratios. The error coefficient equals to 5. It performs as predicted for practical signal to noise ratio, i.e, close to 3.98 dB gain over MSK for a BER equal to 10^{-5}, and about 2 dB for a BER equal to 10^{-3}.

Figure 5.24 shows the BER curve for $S_V = 16$. The upper bound is tight for high

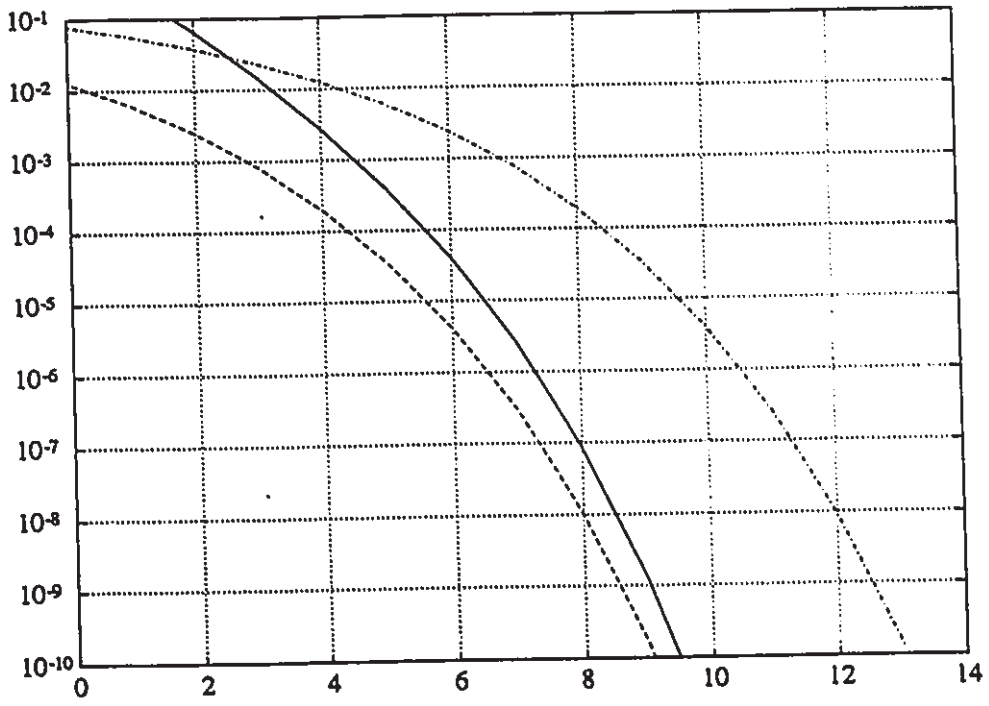


Figure 5.23: Rate-1/2 binary encoded quaternary CPFSK with $h = 1/2$, $S_V = 8$. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min} E_b/N_0)^{1/2})$. The "— · — · —" line is the function $Q((2E_b/N_0)^{1/2})$.

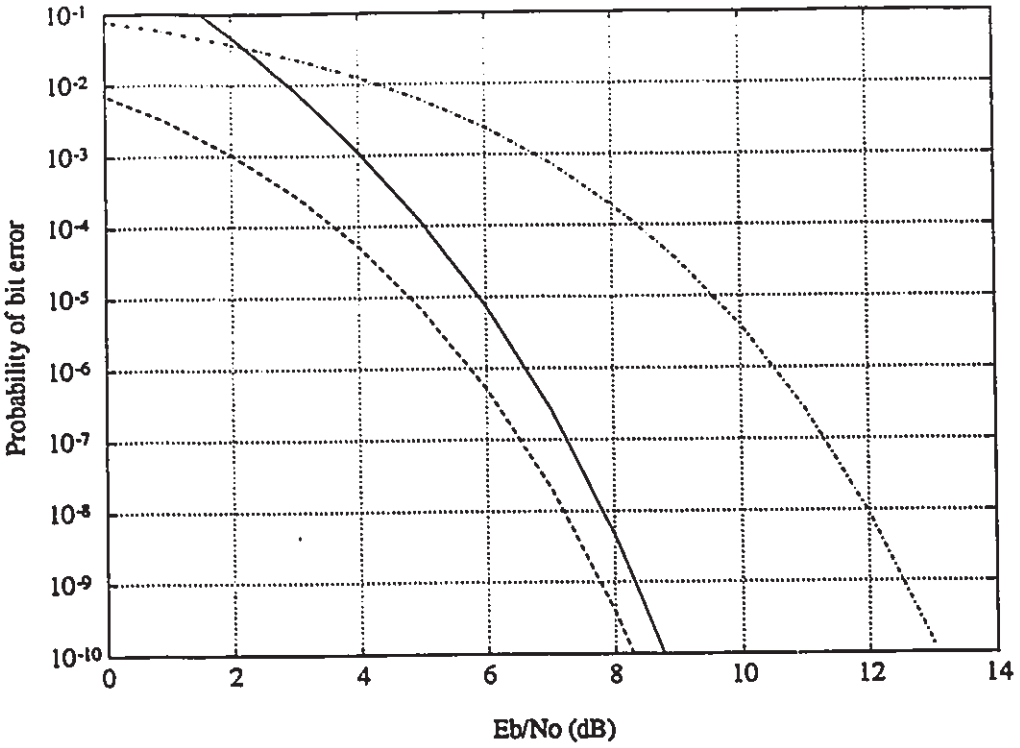


Figure 5.24: Rate-1/2 binary encoded quaternary CPFSK with $h = 1/2$, $S_V = 16$. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min}E_b/N_0)^{1/2})$. The "— · — · —" line is the function $Q((2E_b/N_0)^{1/2})$.

SNR. The error coefficient equals to 7. It has about 4.77 dB gain over MSK for a BER equals to 10^{-10}. However, since the error coefficient is 7, the coding gain over MSK are about 3.6 dB a at BER equal to 10^{-5} and only 2 dB for a BER equal to 10^{-3}.

(b) Rate-1/2 modulo-4 encoded quaternary CPFSK with $h = 1/4$

In this subsection, we show not only the upper bound, but also the simulation results of the systems with $S_V = 4$ and $S_V = 8$. In Figure 5.25 shows the BER curve for rate-1/2 modulo-4 encoded $h = 1/4$ quaternary CPFSK with $S_V = 4$. The upper dotted line is the curve for $Q(\sqrt{2E_b/N_0})$, where we obtain our coding gains by comparing this with the corresponding curve in the last section. The solid line is the upper bound. The lower dotted line is the curve predicted by the Euclidean distance,

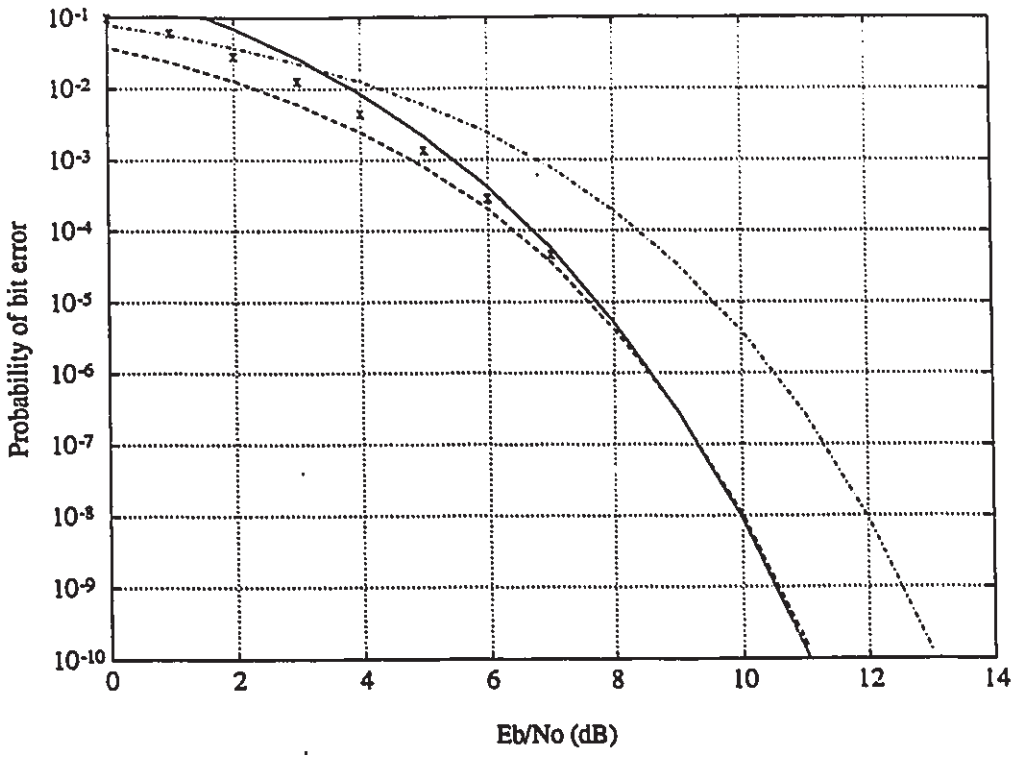


Figure 5.25: Rate-1/2 modulo-4 encoded quaternary CPFSK with $h = 1/4$, $S_V = 4$. The line marked with "x" is the simulated BER. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min}E_b/N_0)^{1/2})$. The "— · — · —" line is the function $Q((2E_b/N_0)^{1/2})$.

i.e. $Q(\sqrt{3.15E_b/N_0})$. In this case, the Euclidean distance alone is pessimistic, because the error coefficient is smaller than 1, i.e. $C_{d_{min}} = 0.468$. Theoretically, d^2_{min} predicts a 3.15 dB gain over MSK. The system is still very good at a BER equal to 10^{-5}, and it has about a 2 dB gain over MSK at a BER equal to 10^{-3}. This is still very good.

Figure 5.26 shows the BER curve for $S_V = 8$. The upper bound is tight and the error coefficient equals 1.25. It performs very well for both practical and small SNRs. This figure shows that the upper bound is reasonably tight.

Figure 5.27 shows the BER curve for $S_V = 16$. The upper bound is tight and the error coefficient equals 0.375. It performs very well for practical and small SNRs. The gain predicted by the normalized squared Euclidean distance is reasonably good

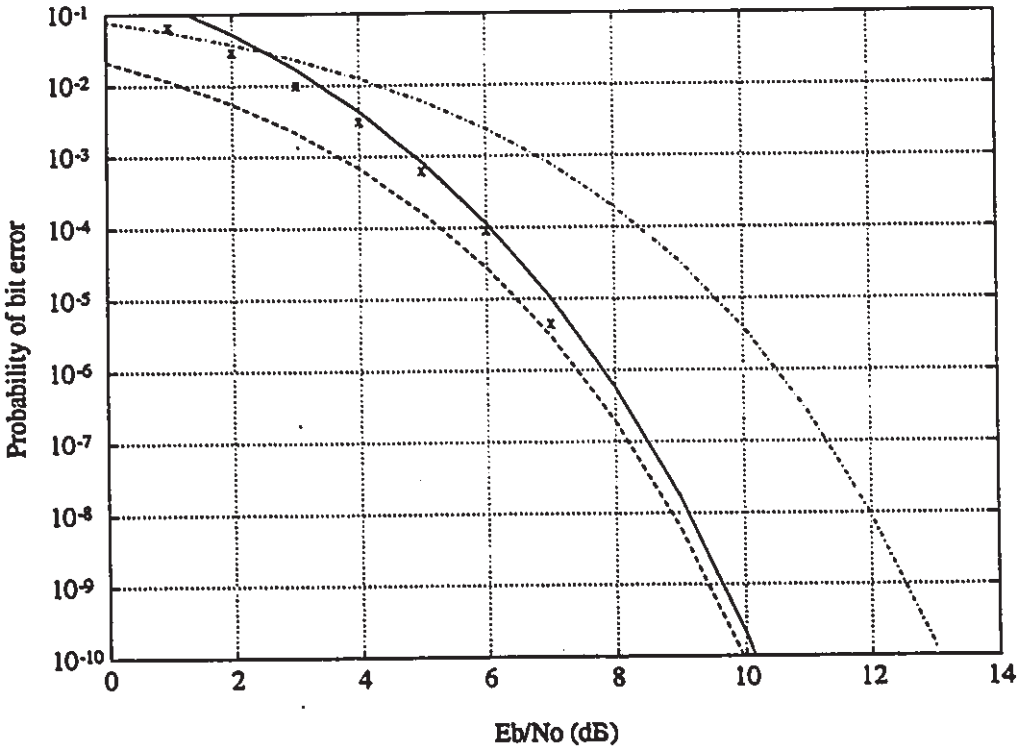


Figure 5.26: Rate-1/2 modulo-4 encoded quaternary CPFSK with $h = 1/4$, $S_V = 8$. The line marked with "x" is the simulated BER. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min} E_b/N_0)^{1/2})$. The "— · — · —" line is the function $Q((2E_b/N_0)^{1/2})$.

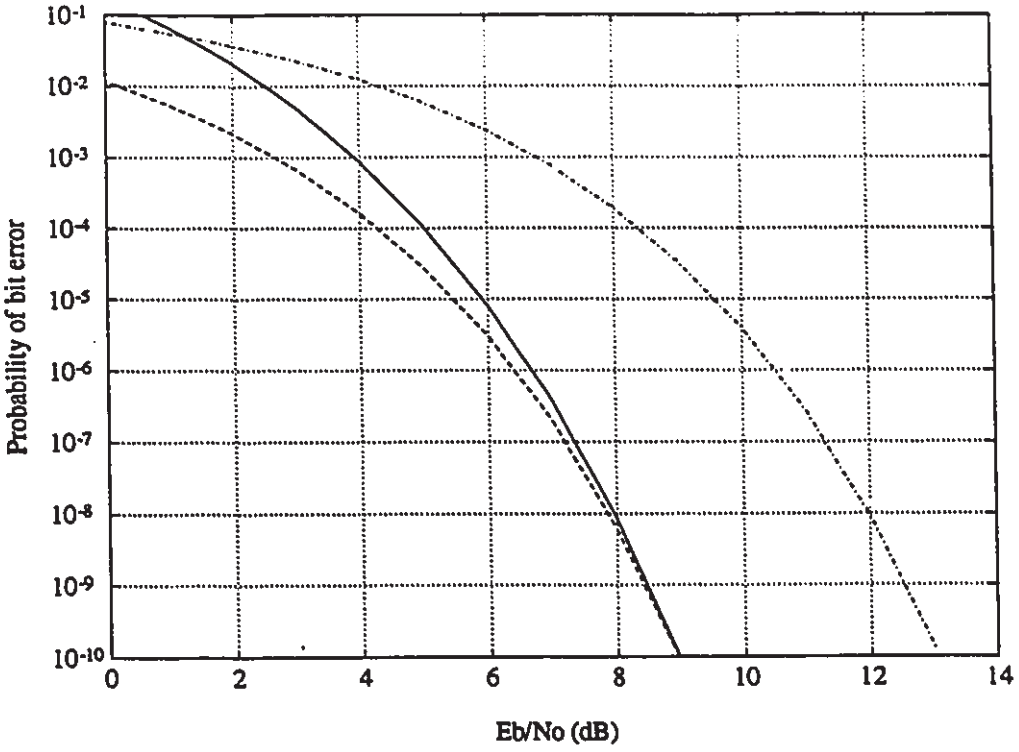


Figure 5.27: Rate-1/2 modulo-4 encoded quaternary CPFSK with $h = 1/4$, $S_V = 16$. The solid line is the upper bound. The "- - -" line is the function $Q((d^2_{min} E_b/N_0)^{1/2})$. The "— · — · —" line is the function $Q((2E_b/N_0)^{1/2})$.

for moderate and practical signal to noise ratios.

To simulate the BER of the systems, we have used a random number generator in [60]. The generator generates binary input sequence and every two binary input are mapped into a quaternary number. We have employed the MLSE receiver given in [8, 43], which can be simulated using in-phase and quadrature phase components. The AWGN can be generated by a subroutine in [60]. About the digital simulation techniques, execellent references can be found in [61, 8, 62]. In order to measure an average probability of bit errors as low as 10^{-x}, for example, the number of input bit sequences will have to be at least as large as 10^{x+2}; that is, 100 times the reciprocal of the error rate. This results in a standard deviation or root mean-square measurement error of no more than 10 percent [61].

5.7 Discussion and Conclusions

In this chapter, we investigate the bit error probability performance of ring convolutional encoded CPFSK. We view the CPFSK in a decomposed manner, i.e. a ring convolutional encoder combined with a memoryless modulator. Therefore, it is not surprising that we obtained the same bound as that of the well-known trellis coded modulation for a memoryless modulator [13]. New numerical results for ring encoded and uncoded CPFSK and feedback-free CPFSK are presented. The BER bound shows that ring convolutional encoded CPFSK has a low error coefficient and achieves good coding gains for moderate and practical SNRs. The feedback-free CPFSK usually has a smaller error coefficient than that of CPFSK.

Chapter 6

Conclusions, Discussion and Future Work

In this thesis a new coding scheme based upon convolutional codes on the ring of integers modulo-P has been applied to obtain trellis-encoded CPFSK. Three important characteristics of modulo-P encoded CPFSK are found:

1. The modulo-P convolutional encoder is a natural coding scheme for CPFSK in the sense of using a similar code structure to that of the CPE.
2. The modulo-P encoding approach has a larger state reduction property than that of other approaches.
3. The modulo-P encoding approach generates the same signal space codes for both the feedback-free and feedback forms of CPFSK.

These three characteristics are not found in previous work [1, 52, 5, 4, 6].

6.1 Conclusions

Our previous work [6] showed that no single approach to encoding CPFSK will consistently obtain the best performance if only binary convolutional codes are employed.

Based on the code structure of CPFSK, this thesis suggests a new convolutional encoded CPFSK system [31]. It shows that a *ring convolutional encoder* [16], called a *modulo-P encoder* [32], is a more *natural* way to combine the CE/CPE pair than a binary convolutional encoder [7], and that the resulting coded modulation consistently obtains better performance than in previous approaches.

This thesis also derives the upper bound for the bit error rate of the ring encoded CPFSK systems. It is shown that the bit error probability performs well for both practical and moderate signal to noise ratios. This is very useful for CPFSK to be applied to other channels, namely fading dispersive channels. Also, the coding gain using ring encoded CPFSK is very close to the coding gain predicted by the practical channel capacity [10], R_0. For example, it is predicted by R_0 [27] that rate-1/2 coded quaternary CPFSK with $h = 1/4$ has approximately 7 dB gain over MSK, and we have achieved 6 dB for the complexity of 256 states.

This thesis is the first work in the CPM literature that employs non-field convolutional codes to CPM. The coding gain is high, for example, rate-1/2 modulo-4 coded quaternary $h = 1/4$ CPFSK has coding gain over minimum shift keying (MSK) from 2 dB to 6 dB, for the complexity of 4 states to 256 states in the receiver. This is very effective compared to all the previous work. Previous work has only minor coding gain for relatively simple complexity in the receiver (0.3 dB for the complexity of 4 states [4]). Also, to the author's knowledge, people have recently found that using the concepts of ring convolutional codes or group convolutional codes to TCM, new encoding structures for TCM are possible [17, 33]. However, there has been no evidence that better coding scheme can be found with this new point of view for TCM yet. This work is the first one that applies ring encoders to digital modulations that obtains significant improvement over the traditional approaches. Therefore, it seems very encouraging that these new classes of convolutional codes may become useful for other digital modulations in the future.

Three conclusions are summarized:

1. A new coding scheme has been found, which can obtain consistently good codes compared to previous work. Numerical results confirm the superiority of this approach.

2. The BER of ring convolutional encoded CPFSK is evaluated. It shows that this new coding scheme performs well at both practical and moderate signal to noise ratios.

3. This new coding scheme combines the idea of using a ring encoder, the idea of using a scrambler to remove the feedback of CPFSK, and the idea of matched codes.

6.2 Discussion

Since the decomposition model of CPFSK is equivalent to traditional CPFSK, it is not necessary to implement a *new* CPFSK that physically consists of a CPE and a MM. Instead, we can connect a ring encoder with a traditional CPFSK. To implement the ring encoded CPFSK system, we can use a natural mapper to map the binary information sequences into ring symbol sequences. The quaternary symbol sequences are then connected to a ring convolutional encoder. The encoded symbol sequences are directly connected to traditional CPFSK (CPFSK with all positive symbols input) without using a mapper. The receiver can be implemented as the traditional MLSE receiver as shown in [8, 43]. Therefore, we can implement the ring encoded CPFSK systems without using any new devices except the ring encoder.

6.3 Future Work

Future work will be included:

1. Since the CPE of partial response CPM is also often a ring encoder with more states, good ring convolutional codes for CPM are promised.

2. Because we have found that the BER of ring encoded CPFSK has good performance at moderate SNRs, ring encoded CPFSK may have applications in channels other than AWGN.

3. Ring convolutional codes can be generalized as convolutional codes over groups [17]. When the CPE of CPM is not a ring encoder, it will be interesting to determine what is the algebraic structure of this CPE. It may be lead to new code structures to implement with this CPE.

Appendix A

Derivation of The Equivalent CPE For Different Examples

(a) $h = 1/4$ quaternary CPFSK: In Fig. A.1, the equivalent rate-2/4 CPE has the following input/output relations:

$$\begin{aligned} c^1(D) &= b^1(D) - D \cdot b^2(D) \\ c^2(D) &= D \cdot b^2(D) \\ c^3(D) &= -1 \cdot b^1(D) + b^2(D) \\ c^4(D) &= b^1(D) \end{aligned} \tag{A.1}$$

Where $-1 \equiv 3$ (modulo 4) and the transfer function in (4.6) is obtained.

(b) $h = 1/8$ octal CPFSK: In Fig. A.2, the equivalent rate-3/6 CPE has the following input/output relations:

$$\begin{aligned} c^1(D) &= b^1(D) - D \cdot b^3(D) \\ c^2(D) &= D \cdot b^3(D) \\ c^3(D) &= -1 \cdot b^1(D) + b^2(D) \\ c^4(D) &= b^1(D) \\ c^5(D) &= -1 \cdot b^2(D) + b^3(D) \end{aligned}$$

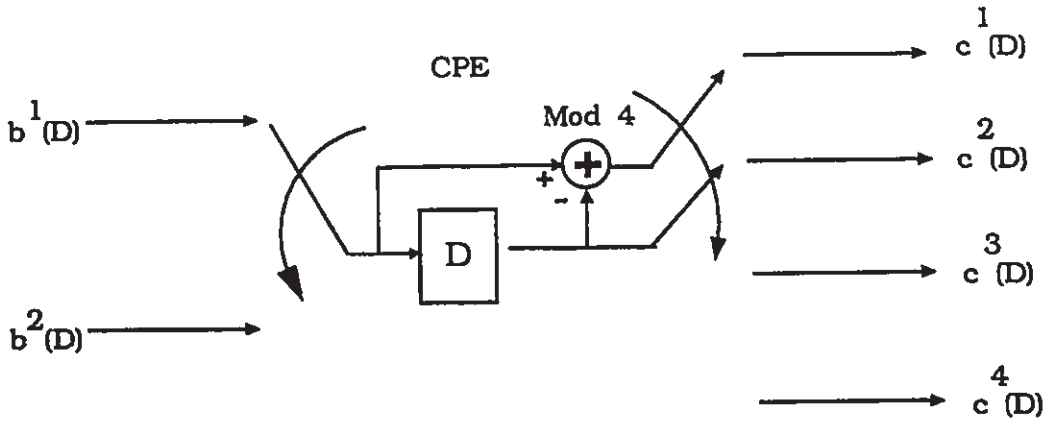


Figure A.1: Rate-2/4 modulo-4 equivalent encoder.

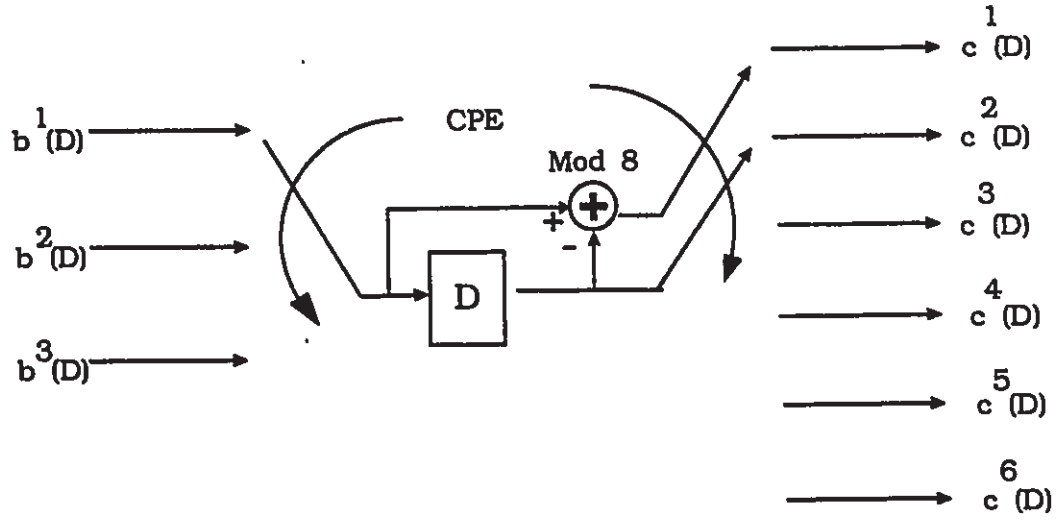


Figure A.2: Rate-3/6 modulo-8 equivalent encoder.

$$c^6(D) = b^2(D) \tag{A.2}$$

Where $-1 \equiv 7$ (modulo 8) and the transfer function in (4.7) is obtained.

Appendix B

Review of Convolutional Coding Techniques

In this appendix we give a brief description of convolutional codes. Some important facts from convolutional codes will be summarized. Excellent references on convolutional codes can be found in [63, 9, 13, 50, 64].

B.1 Convolutional Codes Defined over Galois Field

For simplicity, we first assume that our codes are binary and defined over the field GF(2). Convolutional codes can be studied from many points of view. We start with a simple example.

Example 1: Figure B.1 shows a linear sequential circuit. The delay cell, D, is a shift register, the input symbol at time i is a_i, The two D delay cells are actually memory cells. Previous two input symbols a_{i-1} and a_{i-2} are stored in this two delay cells. At each time interval, T_a, one input symbol will produce two output symbols. Each output is a function of the input symbol and previous two input symbols. The function is performed by an modulo-2 adder $\oplus$. Since modulo-2 addition is a linear operation, the encoder is a linear feed-forward shift register, called convolutional encoder. To

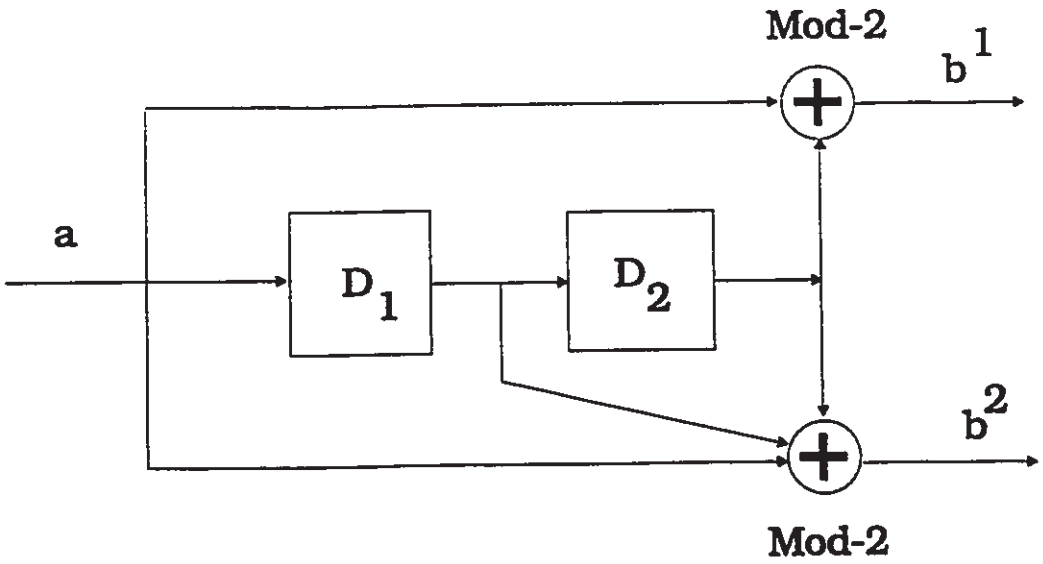


Figure B.1: An example of convolutional encoder.

describe a convolutional codes in this way is called the *shift register approach.* All convolutional encoders can be implemented using a linear feed-forward shift register of this type [50]. The output of encoded sequences are represented in a time sequence

$$\begin{aligned} \mathbf{b}^1 &= (b_0^1, b_1^1, b_2^1, \ldots), \\ \mathbf{b}^2 &= (b_0^2, b_1^2, b_2^2, \ldots). \end{aligned} \tag{B.1}$$

All the possible output, $\mathbf{b} = (\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_i, \ldots)$ with $\mathbf{b}_i = (\mathbf{b}_i^1, \mathbf{b}_i^2)$, $0 \leq i \leq \infty$, are called *codewords* of that encoder.

Since this sequential circuit is a finite state machine, it can be represented as a state transition diagram [63], shown in Figure B.2. There are four states in this diagram and each state is the contents of the two delay cells. Each branch is labeled with the input/output relations. There are total four states. This is way of description is called the *state diagram approach.*

In Figure B.3 shows a *trellis* diagram which equivalently described the above state diagram. From this diagram, we see that the set of possible encoded sequences are just the paths through the trellis. In which we have labeled the nodes with the encoder state and the input/output relation are labeled to each branch. This description is

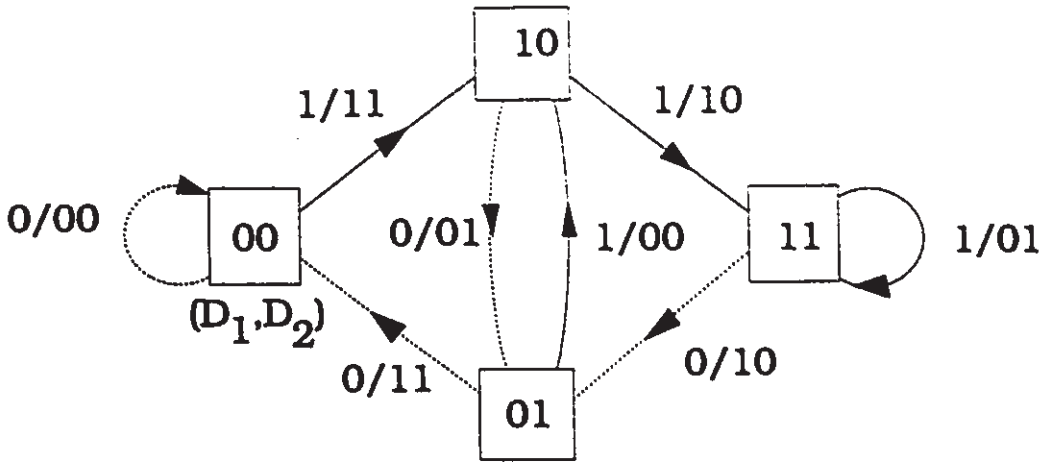


Figure B.2: State diagram

called the *trellis diagram approach.*

Another approach of description, called the *scalar matrix approach*, is now introduced. The information sequence $\mathbf{a} = (a_0, a_1, a_2, \ldots)$ enters the encoder one bit at a time. Since the encoder is a linear system, the two encoder output sequences $\mathbf{b}^1$ and $\mathbf{b}^2$ can be obtained as the convolution of the input sequence with the "input response" of the two subencoder, see Figure B.4.

The impulse response for each subencoder are

$$\begin{aligned} \mathbf{g}_1 &= (g_{10}, g_{11}, g_{12}) = (101) \\ \mathbf{g}_2 &= (g_{20}, g_{21}, g_{22}) = (111) \end{aligned} \tag{B.2}$$

The encoding equation can now be written as

$$\begin{aligned} b_i^1 &= \sum_{j=0}^{2} a_{i-j} g_{1j} \\ b_i^2 &= \sum_{j=0}^{2} a_{i-j} g_{2j} \end{aligned} \tag{B.3}$$

i.e.

$$\begin{aligned} b_i^1 &= a_i g_{10} \oplus a_{i-1} g_{11} \oplus a_{i-2} g_{12} \\ b_i^2 &= u_i g_{20} \oplus u_{i-1} g_{21} \oplus u_{i-2} g_{22} \end{aligned} \tag{B.4}$$

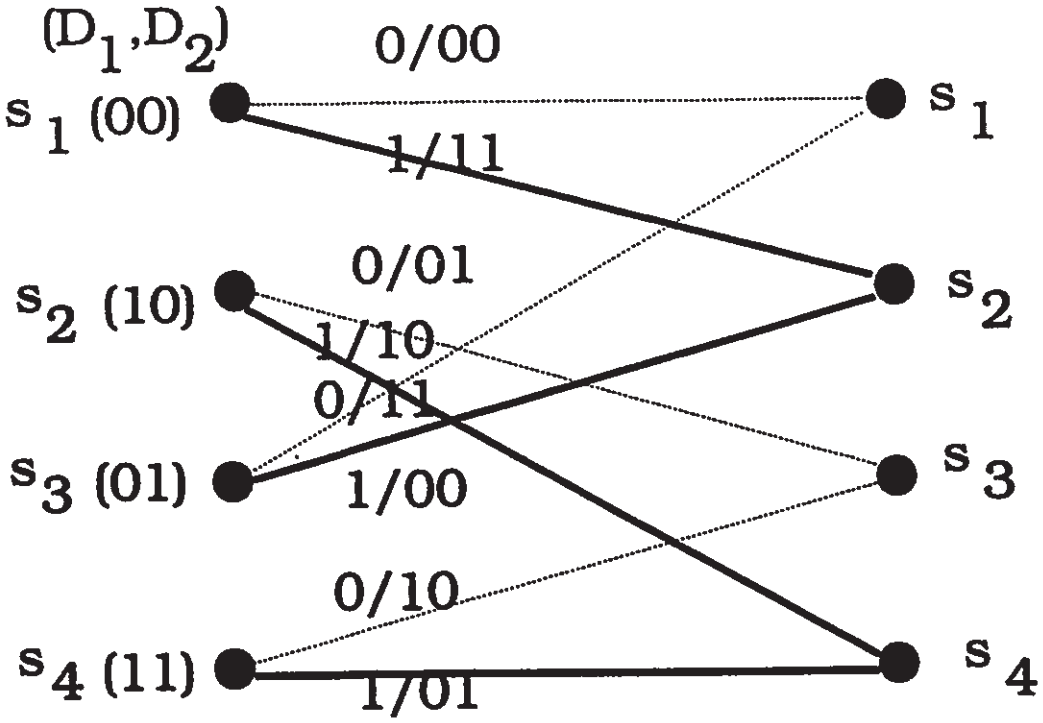


Figure B.3: Trellis diagram.

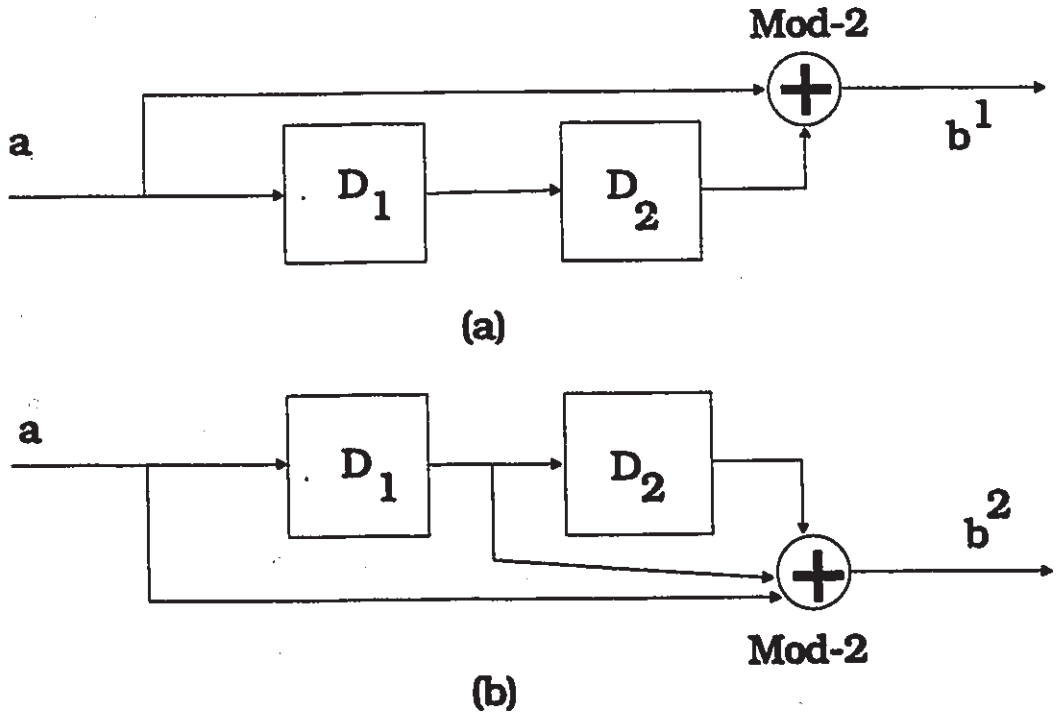


Figure B.4: Subencoder of the convolutional encoder.

where $a_{i-j} \equiv 0$ for all $i < j$. After encoding, the two output sequences are multiplexed into a single sequence, called the *code word*, for transmission over the channel. The code word is given by

$$\mathbf{b} = (b_0^1 b_0^2, b_1^1 b_1^2, \ldots) \tag{B.5}$$

If the generator sequences $\mathbf{g}_1$ and $\mathbf{g}_2$ are interlaced and then arranged in the matrix

$$\mathbf{G}_s = \left[\begin{array}{c|c|c|c|c|c|c} g_{10}g_{20} & g_{11}g_{21} & g_{12}g_{22} & g_{13}g_{23} & & & \\ & g_{10}g_{20} & g_{11}g_{21} & g_{12}g_{22} & g_{13}g_{23} & & \\ & & g_{10}g_{20} & g_{11}g_{21} & g_{12}g_{22} & g_{13}g_{23} & \\ & & & \ddots & & & \ddots \end{array}\right], \tag{B.6}$$

where the blank areas are all zeros, the encoding equations can be rewritten in matrix form as

$$\mathbf{b} = \mathbf{a} \cdot \mathbf{G}_s, \tag{B.7}$$

where all operations are modulo-2. $\mathbf{G}_s$ is called the *scalar generator matrix* of the code.

In any linear system, time-domain operations involving convolution can be replaced by more convenient transform-domain operations involving polynomial multiplication. Since a convolutional encoder is a linear system, each sequence in the encoding equations can be replaced by a corresponding polynomial, and the convolution operation replaced by polynomial multiplication. In the polynomial representation of a binary sequence, the sequence itself is represented by the coefficients of the polynomial. For example, for the above encoder, the encoding equations become

$$\begin{aligned} \mathbf{b}^1(D) &= \mathbf{a}(D)\mathbf{g}_1(D) \\ \mathbf{b}^2(D) &= \mathbf{a}(D)\mathbf{g}_2(D) \end{aligned} \tag{B.8}$$

where

$$\mathbf{a}(D) = a_0 + a_1 D + a_2 D^2 + a_3 D^3 + \ldots \tag{B.9}$$

is the information sequence, and

$$\mathbf{b}^1(D) = b_0^1 + b_1^1 D + b_1^1 D^2 + \dots$$
$$\mathbf{b}^2(D) = b_0^2 + b_1^2 D + b_2^2 D^2 + \dots \tag{B.10}$$

are the encoded sequences.

$$\mathbf{g}_1(D) = g_{10} + g_{11} D + g_{12} D^2$$
$$\mathbf{g}_2(D) = g_{20} + g_{21} D + g_{22} D^2 \tag{B.11}$$

are the generator polynomial of the code, and all operations are modulo-2. After multiplexing, the code word becomes

$$\mathbf{b}(D) = \mathbf{b}^1(D^2) + D \cdot \mathbf{b}^2(D^2) \tag{B.12}$$

The indeterminate D can be interpreted as a delay operator, the power of D denoting the number of time units a symbol is delayed with respect to the initial symbol in the sequence. The generator polynomials of an encoder can be determined directly from its circuit diagram. Since each shift register state represents a one-time-unit delay, the sequence of connections (a 1 representing a connection and a 0 no connection) from a shift register to an output is the sequence of coefficients in the corresponding generator polynomial (i.e., it is the generator sequence). For example, in Figure B.4, the sequence of connections from the shift register to the first output is $\mathbf{g}_1 = (101)$, and the corresponding generator polynomial is $\mathbf{g}_1(D) = 1 + D^2$. Equation (B.8) can be represented as a 1×2 matrix, called *polynomial matrix*,

$$\mathbf{G}(D) = [\mathbf{g}_1(D), \mathbf{g}_2(D)] \tag{B.13}$$

Using the polynomial matrix, the encoding equation for this encoder can be expressed as

$$\mathbf{b}(D) = \mathbf{a}(D) \cdot \mathbf{G}(D) \tag{B.14}$$

By elementary operations, the polynomial matrix $\mathbf{G}(D)$ can be written as

$$\mathbf{G}'(D) = [1, \frac{1 + D + D^2}{1 + D^2}] \tag{B.15}$$

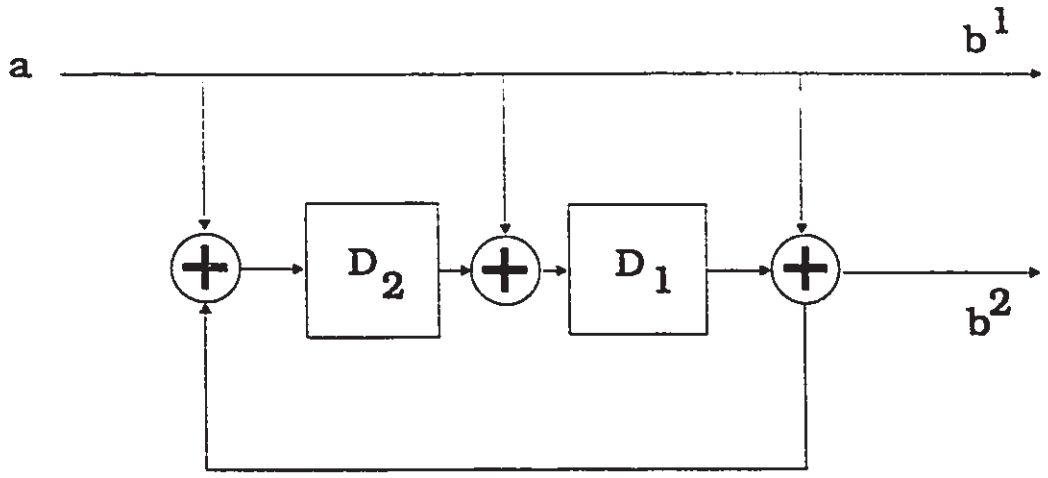


Figure B.5: Feedback realization of convolutional encoder.

To implement this encoder one can use a feedback shift register [65] as shown in Figure B.5 [50]. Trellis diagram for this encoder is shown in Figure B.6. Form this figure, we can see that feedback encoder generate the *same* code word as the one without feedback (however, the input data are different, but the output are the same.) We say this two encoder are *equivalent* since they generate the same code words. This is not a special example. In fact, every feed-forward polynomial has a equivalent feedback encoder [50].

B.1.1 Definitions and Propositions of Convolutional Codes

A convolutional code produces l encoded symbols for each k information symbols, i.e. $R_c = k/l$, is called a rate-k/l convolutional code. The $k \times l$ polynomial function matrix can be used to generate this code is [64]

$$\mathrm{G}(D) = \left[\begin{array}{c|c|c|c} g_{11}(D) & g_{12}(D) & \cdots & g_{1l}(D) \\ g_{21}(D) & g_{22}(D) & \cdots & g_{2l}(D) \\ \vdots & \vdots & & \vdots \\ g_{k1}(D) & g_{k2}(D) & \cdots & g_{kl}(D) \end{array}\right]. \tag{B.16}$$

The above polynomial matrix can be realized with a finite number of memory cells capable of storing field elements, a finite number of scalars and adders that perform

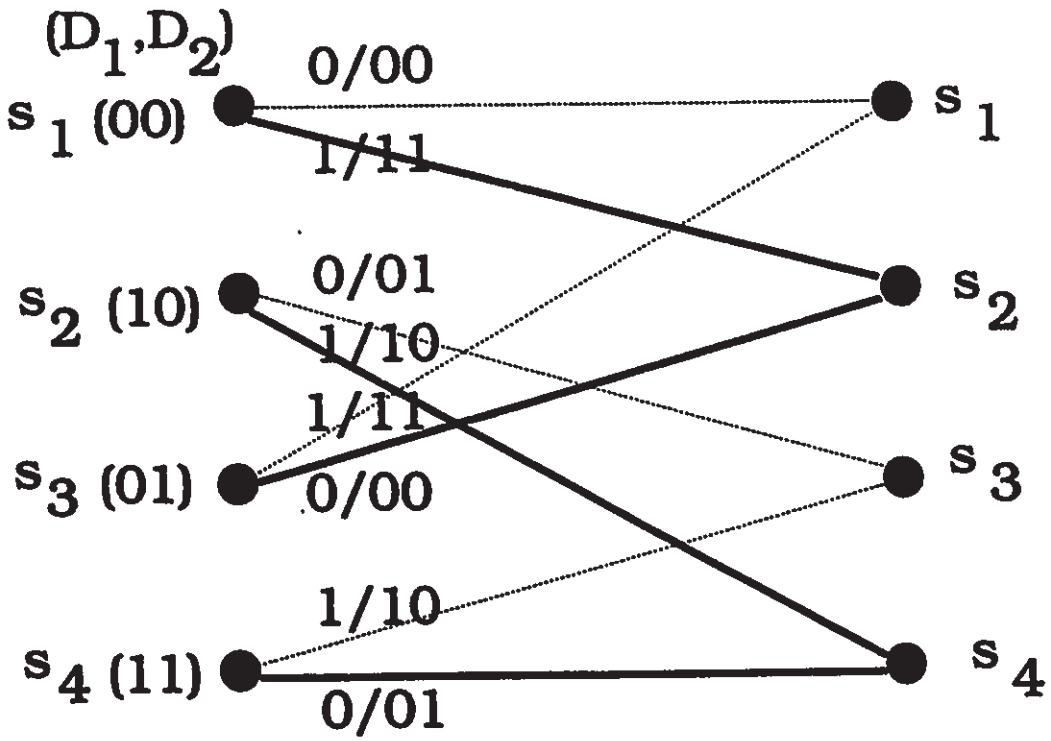


Figure B.6: Trellis diagram of feedback encoder.

multiplication by constants and additions, respectively, within the field F. In [50] this encoder can be implemented by an *obvious realization* [50]. One example of obvious realization is shown in the last section for a rate-1/2 encoder with two delay cells. The number of delay cells needed to implement the above encoder depends on a coefficient *constraint length*, ν. It is defined as [50]

$$\nu = \sum_{i=1}^{k} \nu_i \tag{B.17}$$

where ν_i is the constraint length for input i, defined as

$$\nu_i = \max_{1 \le j \le l} [\deg(g_{ij}(D))] \tag{B.18}$$

Definition B.1 *Minimal Encoder: A given realizable generator matrix $\mathbf{G}(D)$ is minimal, if there exists a realization of $\mathbf{G}(D)$ that uses the least number of encoder states required to generate the corresponding code [54].*

Definition B.2 *Catastrophic Encoder: A infinite nonzero input symbols that can cause only finite nonzero symbols output.*

The catastrophic encoder is called catastrophic can be explained as following: if the input sequence has finite nonzero symbols and produce a finite nonzero symbols codewords. The encoded sequence is transmitted through the channel and is disturbed by noise to produce finite number of errors. If it happen that the decoded sequence correspond to a infinite nonzero input symbols, then compare to the sending finite nonzero inputs symbols, we have made infinite errors at the receiver. That is the reason to call such a encoder *catastrophic*. In digital communications, we must avoid the overall systems to be catastrophic.

Proposition B.1 *Every polynomial encoder is equivalent to a systematic encoder*

Proposition B.2 *Every systematic encoder is minimal and noncatastrophic*

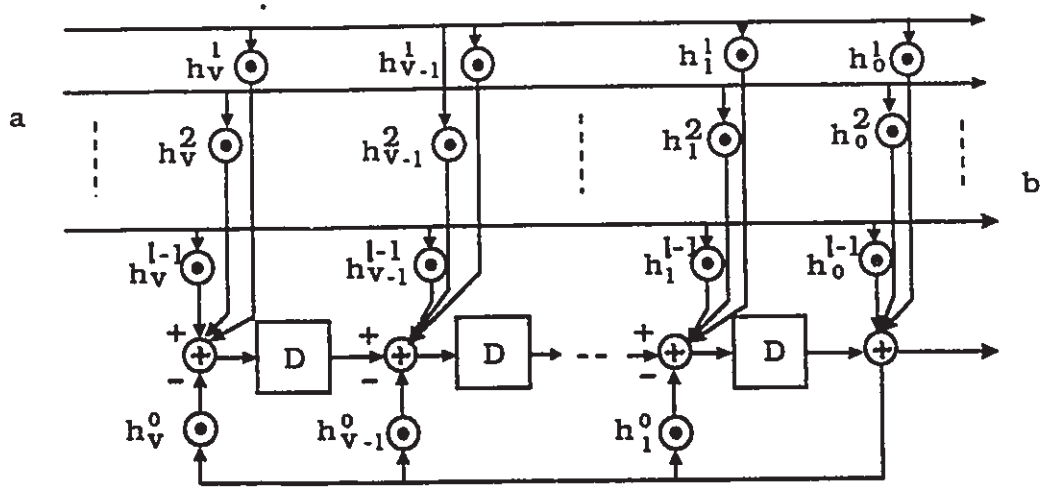


Figure B.7: Rate-$l/l+1$ systematic feedback encoder define over GF(P).

Because of the above propositions, it is convenient to design codes by using a rate-$l/l+1$ canonical systematic encoders, shown in Figure B.7 to enumerate codes. This encoder is both minimal and noncatastrophic. All the multiplication and addition in Figure B.7 are defined on GF(P).

B.1.2 Maximum-Likelihood Decoding for Convolutional Codes: The Viterbi algorithm

Refer to trellis representation of a rate-1/2 convolutional encoder, see Figure B.3. We can draw it from level 0 to level n as shown in Figure B.8. The dotted line represents an input "0", corresponds to the selection of the upper branch, and the solid line represents an input "1", corresponds to the selection of the lower branch. Each possible input sequence corresponds to a particular path through the trellis. For instance, an input sequence of (11110000) can be seen to provide an output sequence of (1110010110110000).

The problem of decoding a convolutional code can be thought of as attempting to find a path through the trellis diagram by making use of some decoding rule. An effective approach, which was proposed by Viterbi [66], is now known as the

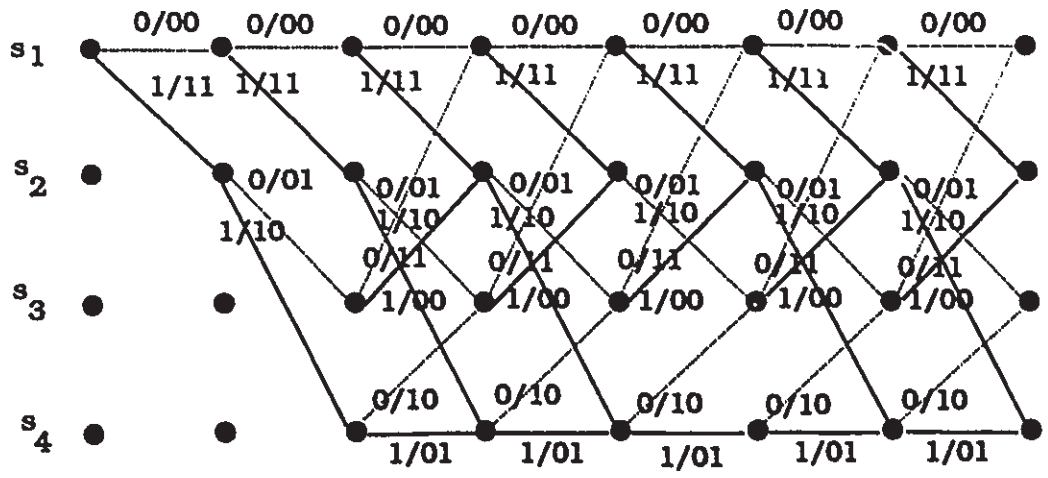


Figure B.8: Trellis extends to level 8.

Viterbi algorithm. The Viterbi algorithm is best explained by using the trellis diagram for the $R = 1/2$, $\nu = 2$ code as a reference. We will now explain how the trellis diagram can be used to decode a convolutional code. Suppose the encoder defined in the last section is being used on a binary symmetrical channel (BSC) with raw bit error probability $p < 1/2$. The information input is (11110000) and that $\mathbf{R} = [11, 11, 01, 11, 10, 11, 00, 00, 00, \ldots]$ is received. A maximum likelihood decoder must find the path which is closest in Hamming distance to **R**. Let us draw a new version of Figure B.8 in which each trellis edge is labeled with the Hamming distance between the Figure B.8 label and the corresponding two bits of **R**. For example, let s_{ij} represent the node with state s_i at level j, then s_{23} to s_{34} edge gets the label $d_H(00, 11) = 2$. The result is the trellis of Figure B.9.

If we think of the edge labels in Figure B.9 as lengths, the total Hamming distance between **R** and a given codeword is just the total length of the trellis path corresponding to the codeword. For example, the Hamming distance between **R** and the codeword $(00, 11, 01, 00, 10, 10, 11, 00, 00, \ldots)$ corresponding to the path $s_{10}, s_{11}, s_{22}, s_{33}, s_{24}, s_{45}, s_{36}, s_{07}, s_{08}, \ldots$, is $2 + 0 + 0 + 2 + 0 + 1 + 2 = 7$. Thus the problem of finding the codeword closest to **R** becomes the problem of finding the shortest path from s_{10} to s_{18} in the trellis of Figure B.9, and we now focus on this

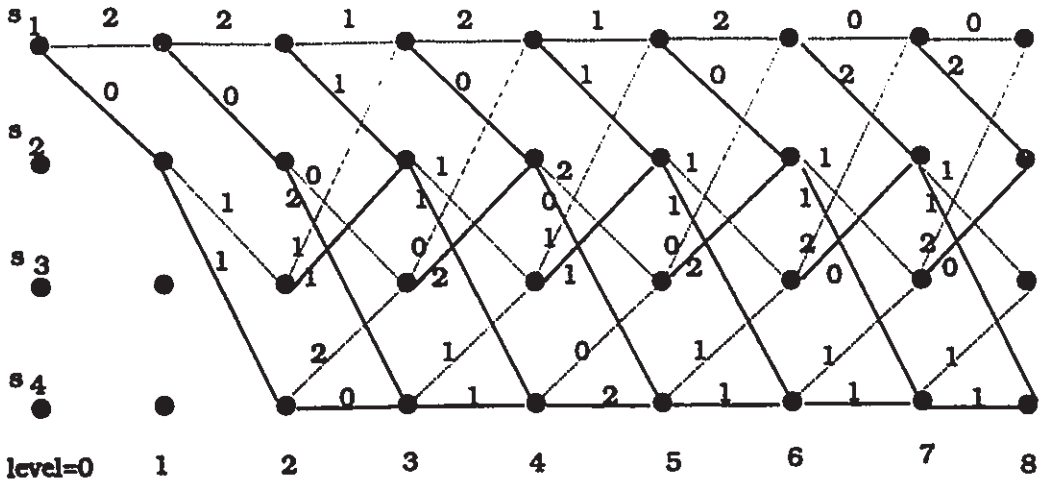


Figure B.9: Trellis labeled with Hamming distance corresponding to **R**.

shortest-path problem.

Suppose the shortest path P from s_{10} to s_{18} passes through some intermediate node, labeled x in Figure B.10. Let us denote the segment of P joining s_{10} to s_x as P_1 and the segment of P joining s_x to s_{18} as P_2. If another path joining s_{10} to s_x, say Q, were shorter, the path QP_2 would be shorter than $P = P_1P_2$, contradicting the minimality of P. This observation is the key to the *Viterbi decoding algorithm*, which works by constructing, for each j, a list of the shortest paths from s_{10} to the vertices at depth j. The $(j+1)$st list is easily constructed from the jth list since if $s_{10} \ldots s_{xj}s_{y(j+1)}$ is a shortest path to $s_{y(j+1)}$ then $s_{10} \ldots s_{xj}$ must be a shortest path to s_{xj}, and so the shortest depth $j+1$ paths can be obtained by extending the shortest depth j paths by one edge.

Before we state Viterbi's algorithm, first we need some notation (This algorithm is from [63]). Denote by S the set of states $\{s_1, s_2, s_3, s_4\}$. Next if $s_x, s_y \in S$ and there is an edge going from s_x to s_y in the state diagram, we define $B(s_x, s_y)$ to be 0 or 1 according to whether the transition from s_x to s_y corresponds to a 0 or a 1 input. If there is no such edge, $B(s_x, s_y)$ is not defined (see Figure B.11). Also, for $s_x, s_y \in S$, we define $l_{j-1,j}(s_x, s_y)$ to be the label on the trellis edge joining $s_{x(j-1)}$ to s_{yj}. If there is no such edge, we set $l_{j-1,j}(s_x, s_y) = +\infty$. For example, $l_{0,1}(s_1, s_2) = 1$,

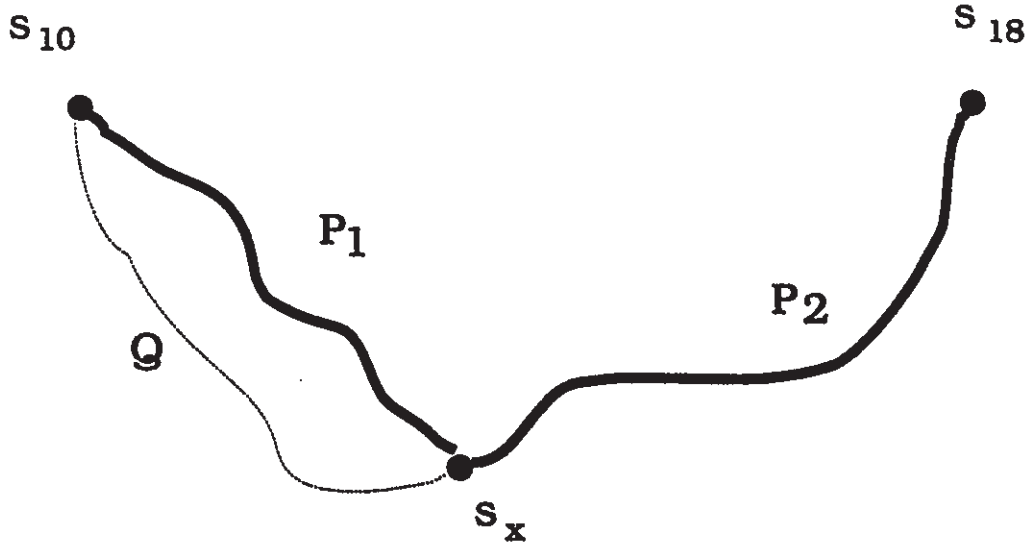


Figure B.10: Two paths lead to the same node.

$B(s_x, s_y)$	s_1	s_2	s_3	s_4
s_1	0	1	–	–
s_2	–	–	0	1
s_3	0	1	–	–
s_4	–	–	0	1

Figure B.11: $B(s_x, s_y)$.

$l_{2,3}(s_4, s_3) = 1$, $l_{1,2}(s_1, s_3) = +\infty$.

We are prepared to state Viterbi's algorithm. It computes two things, *metrics* and *survivors*. The *metric* $m_j(s_x), s_x \in S$, represents the length of the shortest path from s_{10} to s_{xj}; the *survivor* $B_j(s_x)$ is a binary string of length j which represents a shortest path from s_{10} to s_{xj}. Thus $B_4(s_2) = 0101$ means that the shortest path from s_{10} to s_{24} is $s_{10}s_{21}s_{42}s_{43}s_{24}$. The Viterbi algorithm is given as [63]: [1]

1. Initially set $m_0 = 0$, and $m_0(s_x) = +\infty$ for all $s_x \neq s_{10}$. Also, set $B_0(s_1) = \phi$, and $j = 1$.

2. For each $s_y \in S$, find a $s_x \in S$ for which $m_{j-1}(s_x) + l_{j-1,j}(s_x, s_y)$ is a minimum. The set

$$\begin{aligned} m_j(s_y) \quad &= \quad m_{j-1}(s_x) + l_{j-1,j}(s_x, s_y), \\ B_j(s_y) \quad &\text{equals to} \quad B_{j-1}(s_x) \text{ concatenated with } B(s_x, s_y). \end{aligned} \tag{B.19}$$

3. If $j = L + \nu$, output first L bits of $B_j(s_1)$ and stop; otherwise set $j = j + 1$ and go to step 2.

The $m_j(s_y)$ computed by Viterbi's algorithm is in fact the length of a shortest path from s_{10} to s_{yj} and that $B_j(s_y)$ described such a path.

We can describe the performance of Viterbi's algorithm on the trellis of Figure B.9 graphically, as in Figure B.12, where the metric $m_j(s_y)$ appears above the node s_{yj} and the survivor $B_j(s_y)$ is represented by the unique path from s_{10} to s_{yj}. For example, $m_4(s_4) = 2$ and $B_4(s_4) = 1111$. The shortest path from s_{10} to s_{18} is seen to be $s_{10}s_{21}s_{42}s_{43}s_{44}s_{35}s_{16}s_{17}s_{18}$. The decoder's output is therefore 11110000, and this is the maximum likelihood estimate of the information sequence $\mathbf{a} = (11110000)$ that gave rise to $\mathbf{R}$.

The Viterbi algorithm remains valid for an arbitrary discrete memoryless channels, except that the edge labels $l_{j-1,j}(s_{x,y})$ must be redefined.

[1] We assume the Viterbi algorithm will output the first L bits when the binary string B_j exceed $L + \nu$, ν is the constraint length. L is also called truncation length.

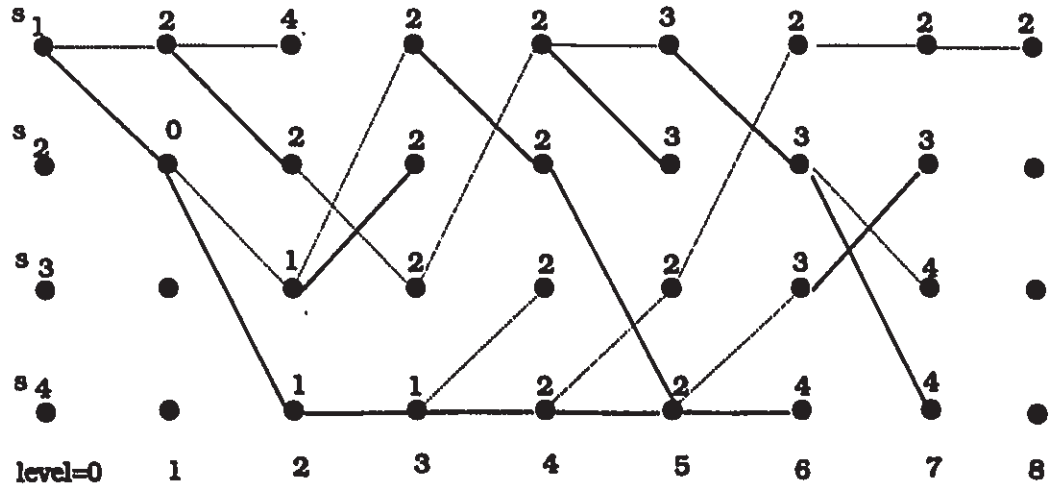


Figure B.12: The Viterbi algorithm applies to decode the R

Appendix C

List of Abbreviations

ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BSC	Binary Symmetric Channel
CE	Channel Encoder
CPE	Continuous Phase Encoder
CPFSK	Continuous Phase Frequency Shift Keying
CPM	Continuous Phase Modulation
DMC	Discrete Memoryless Channel
DMSK	Differential Minimum Shift Keying
FFSK	Fast Frequency Shift Keying
ISED	Incremental Squared Euclidean Distance
MLSE	Maximum Likelihood Sequence Estimation
MM	Memoryless Modulator
MPSK	M-ary Phase Shift Keying
MSK	Minimum Shift Keying
NMSED	Normalized Minimum Squared Euclidean Distance
NSED	Normalized Squared Euclidean Distance

PSK	Phase Shift Keying
SNR	Signal to Noise Ratio
TFM	Tamed-FM
QAM	Quadrature Amplitude Modulation
SED	Squared Euclidean Distance
TCM	Trellis Coded Modulation

Bibliography

[1] G. Lindell, T. Aulin, and C. E. Sundberg, "Minimum Euclidean distance for combination of short rate 1/2 convolutional codes and CPFSK modulation," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 509–519, May 1984.

[2] S. V. Pizzi and S. G. Wilson, "Convolutional coding combined with continuous phase modulation," *IEEE Trans. Commun.*, vol. COM-33, pp. 20–29, Jan. 1984.

[3] F. Morales-Moreno and S. Pasupathy, "Structure, optimization, and realization of FFSK trellis codes," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 730–751, July 1988.

[4] B. Rimoldi, "Design of coded CPFSK modulation systems for bandwidth and energy efficiency," *IEEE Trans. Comm.*, vol. COM-37, pp. 897–905, Sep. 1989.

[5] J. Huber and W. Liu, "An alternative approach to reduced-complexity of CPM-receivers," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 1437–1449, Dec. 1989.

[6] H. Yang, F. Morales-Moreno, and D. P. Taylor, "Efficient design of coded CPFSK," in *IEEE Globecom*, pp. 907.3.1–907.3.5, 1990.

[7] R. H. Yang and D. P. Taylor, "Trellis coded continuous phase frequency shift keying with ring convolutional codes," *to appear in IEEE Trans. Inform. Theory*, Oct. 1993.

[8] J. B. Anderson, T. Aulin, and C. E. Sundberg, *Digital Phase Modulation.* New York: Plenum, 1986.

[9] S. Benedetto, E. Biglieri, and V. Castellani, *Digital transmission theory.* New Jersey: Prentice-Hall, 1987.

[10] J. L. Massey, "Coding and modulation in digital communications," in *Proc. Int. Zurich Sem. on Digital Commu.*, pp. E2(1)–E2(4), Mar. 1974.

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

[12] G. D. Forney, "Coset codes. Part I: Introduction and geometrical classification," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1123–1187, Sept. 1988.

[13] E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to trellis-coded modulation with applications.* New York: Macmillan Publishing Company, 1991.

[14] D. Divsalar and M. K. Simon, "Multiple trellis coded modulation (MTCM)," *IEEE Trans. Communications*, vol. COM-36, April. 1988.

[15] J. K. Wolf and G. Ungerboeck, "Trelllis coding for partial response channels," *IEEE Trans. Commun.*, vol. COM-34, Aug 1986.

[16] J. L. Massey and T. Mittelholzer, "Convolutional codes over rings," in *Proceedings of the Fourth Joint Swedish-USSR Int. Workshop on Information Theory*, pp. 14–18, Aug. 1989.

[17] G. D. Forney and M. D. Trott, "The dynamics of linear codes over groups: state spaces, trellis diagrams and canonical encoders," *submitted to IEEE Trans. Inform. Theory*, Feb. 1992.

[18] H.-A. Loeliger, "Signal sets matched to groups," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1675–1682, Nov. 1991.

[19] J. B. Anderson and C. E. Sundberg, "Advances in constant envelope coded modulation," *IEEE Communication Magazine*, pp. 36–45, December 1991.

[20] J. B. Anderson and D. P. Taylor, "A bandwidth-efficient class of signal-space codes," *IEEE Transactions on Information Theory*, vol. IT-24, pp. 703–712, Nov. 1978.

[21] C.-E. Sundberg, "Combined channel coding and constant amplitude continuous phase modulation," *Lecture Notes in Topics in Coding Theory in Honor of Lars H. Zetterborg, Control and Information Sciences.*, vol. 128, pp. 137–176, Springer Verlag. 1989.

[22] T. Aulin and C.-E. Sundberg, "Continuous phase modulation - Part I: full response signaling," *IEEE Trans. Commun.*, vol. COM-29, pp. 196–209, Mar. 1981.

[23] T. Aulin, N. Rydbeck, and C.-E. Sundberg, "Continuous phase modulation - Part II: partial response signaling," *IEEE Trans. Commun.*, vol. COM-29, pp. 210–225, Mar. 1981.

[24] P. Ho and P. J. McLane, "Spectrum, distance and receiver complexity of encoded continuous phase modulation," *IEEE Trans. Inform. Theory*, vol. IT-88, pp. 1021–1032, Sept. 1988.

[25] J. L. Massey, "The how and why of channel coding," in *Proc. Int. Zurich Sem. on Digital Commu.*, pp. F11(67)–F17(73), Mar. 6-8 1984.

[26] B. Rimoldi, "A decomposition approach to CPM," *IEEE Transactions on Information Theory*, vol. IT-34, pp. 260–270, March 1988.

[27] B. Rimoldi, *Continuous Phase Modulation and Coding for Bandwidth and Energy Efficiency.* Switzerland: Ph.D. dissertation, Swiss Fed. Inst. Tech., 1988.

[28] F. Morales-Moreno and S. Pasupathy, "Optimal concatenation of trellis codes and generalized MSK signals," in *Proc. IEEE Global Telecommun. Confer.*, Nov. 1984.

[29] F. Morales-Moreno, W. Holubowicz, and S. Pasupathy, "Optimization of trellis coded $(1 + D)^2$-FM via matched codes," To be published in IEEE Trans. Commun.

[30] W. Liu and J. Huber, "Double trellis coded CPM," in *IEEE ICC*, pp. 51.4.1–51.4.6, 1988.

[31] R. H. Yang and D. P. Taylor, "On trellis coded CPFSK," in *16th Queen's Biennial Conference on Communication, Queen's University*, pp. 87–90, 1992.

[32] J. Hui, *Joint coding and modulation designs for banalimited satellite channels.* Cambridge, Massachusetts: Master thesis, MIT, 1981.

[33] J. Massey, T. Mittelholzer, T. Riedel, and M. Vollenweider, "Ring convolutional codes for phase modulation," in *IEEE Int. Symp. on Information Theory*, Jan. 1990.

[34] J. Massey, "A short introduction to coding theory and practice," in *Proc. Int. Symposium on Signals, Systems and Electronics, Erlangen, Germany*, pp. 629–633, 18-20 Sept. 1989.

[35] R. B. Filho and P. G. Farrell, "Coded modulation with convolutional codes over rings," in *Eurocode'90, International Symposium on Coding Theory and Applications, Udine, Italy, Nov. 1990. Lectures notes in Computer Science, Springer-Verlag*, pp. 271–280, 1991.

[36] J. Massey and T. Mittelholzer, "Systematicity and rotational invariance of convolutional codes over rings," in *Proc. 2nd Int. Workshop on Algebraic and Combinatorial coding Theory*, pp. 154–159, Sept. 16-22 1990.

[37] T. Mittelholzer, *personal communication.* 14 May 1992.

[38] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 751–772, Oct. 1971.

[39] A. J. Viterbi and J. K. Omura, *Principles of Digital Communications and Coding.* New York: McGraw-Hill, 1979.

[40] G. Lindell and C.-E. Sundberg, "An upper bound on the bit error probability of combined convolutional coding and continuous phase modulation," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1263–1269, Sep. 1988.

[41] T. Aulin, "Symbol error probability bounds for coherent Viterbi detected continuous phase modulated signals," *IEEE Trans. Commun.*, vol. COM-29, Nov. 1981.

[42] D. Divsalar, *Performance of mismatched receivers on bandlimited channels.* University of Californiz, Los Angeles: Ph.D. dissertation, 1978.

[43] G. Lindell, *On Coded Continuous Phase Modulation.* Sweden: Dr. Techn. thesis, University of Lund, 1985.

[44] R. deBuda, "Coherent demodulation of frequency-shift keying with low deviation ration," *IEEE Trans. Communications*, vol. COM-20, pp. 429–435, June 1972.

[45] M. G. Pelchat, R. C. Davis, and M. B. Luntz, "Coherent demodulation of continuous phase binary FSK signals," in *Proceedings of the International Telemetry Conference, Washington, D.C.*, pp. 181–190, 1971.

[46] H. Miyakawa, H. H. Harashima, and Y. Tanaka, "A new digital modulation scheme, multimode CPFSK," in *Proc. Third INternational Conference Digital Satellite Communications, Kyoto, Japan*, 105-112 1975.

[47] J. K. Omura and D. E. Jackson, "Cutoff rates for channels using bandwidth efficient modulations." in *IEEE 1980 National Telecommunications Conference.* pp. 14.1.1 – 14.1.14, 1980.

[48] F. Amoroso and J. A. Kivett, "Simplified MSK signaling technique," *IEEE Trans. Commun.*, vol. COM-25, pp. 433–441, Apr. 1977.

[49] W. L. Liu, *Complexity Reduction of Coherent Receivers for Digital Continuous Phase Modulation.* Munich, Germany: Ph.D. dissertation, University of the Federal Armed Forces, 1990.

[50] G. D. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.

[51] B. Rimoldi, "Exact formula for the minimum squared Euclidean distance of CPFSK," *IEEE Trans. on Commun.*, vol. COM-39, pp. 1280–1282, Sep. 1991.

[52] F. Morales-Moreno,. *Matched Trellis Codes for Finite Memory Modulation.* Canada: Ph.D. dissertation, University of Toronto, 1987.

[53] H. Yang, *Efficient Design of Trellis Encoded CPFSK.* Newark, New Jersey: Master thesis, NJIT, 1989.

[54] T. Mittelholzer, "Minimal encoders for convolutional codes over rings," in *Proc. UK-USSR Int. Symposium on Communication Theory and Applications, Crieff, Scotland*, 9-13 Sept. 1991.

[55] T. W. Hungerford, *Abstract Algebra: An Introduction.* Orlando, Florida: Holt, Rinehart and Winston, Inc., 1990.

[56] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering.* New York: Wiley, 1965.

[57] J. K. Omura, "Performance bounds for Viterbi algorithm," in *IEEE ICC*, pp. 2.2.1 - 2.2.5., 1981.

[58] D. Divsalar, M. K. Simon, and J. H. Yuen, "Trellis coding with asymmetric modulations," *IEEE Trans. Communications*, vol. COM-35, pp. 130–141, Feb. 1987.

[59] M. G. Mulligan and S. G. Wilson, "An improved algorithm for evaluating trellis phase codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 846–851, Nov. 1984.

[60] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C.* New York: Cambridge University Press, 1988.

[61] S. Haykin, *Communication Systems.* New York: John Wiley & Sons, 1983.

[62] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems.* New York: Plenum, 1992.

[63] R. J. McEliece, *The Theory of Information and Coding, Encyclopedia of Mathematics and Its Applications, Volume 3.* Massachusetts: Addison-Wesley, 1977.

[64] S. Lin and J. D. J. Costello, *Error Control Coding: Fundamentals and Applications.* New Jersey: Prentice-Hall, 1983.

[65] W. W. Peterson and J. E. J. Weldon, *Error-Correcting Codes, 2nd ed.* Cambridge, Mass.: MIT Press, 1972.

[66] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. on Information Theory*, vol. IT-13, pp. 260–269, 1967.