# Coding CPFSK for Differential Demodulation

A thesis submitted in fulfilment

of the requirements for the Degree of

**Doctor of Philosophy**

in Electrical and Electronic Engineering

from the University of Canterbury

Christchurch, New Zealand

**Anthony Griffin**

**B.E. (Hons 1)**

February 2000

# Abstract

A differential encoder is developed that preserves the phase trellis of continuous phase frequency shift keying (CPFSK) through differential demodulation. This differential encoder interfaces well with the decomposed model of CPFSK, creating a decomposed model of differentially-encoded and differentially-demodulated CPFSK (DCPFSK). The normalised minimum squared Euclidean distance $d^2_{\text{min}}$ of uncoded DCPFSK is calculated. A code search model is developed, allowing codes over rings to be specifically designed for DCPFSK. The results of code searches show that there is very little loss in $d^2_{\text{min}}$ when comparing coded DCPFSK systems with coherently-demodulated coded CPFSK systems. The performance of uncoded and coded DCPFSK systems in both additive white Gaussian noise (AWGN) and Rayleigh flat fading is analysed and simulated. DCPFSK is shown to be relatively robust to medium to slowly-varying fading, without the use of any additional techniques.

Rate-1/2 encoded quaternary DCPFSK with modulation index $h = 1/4$ is compared with coherently-demodulated uncoded MSK and differentially-encoded and differentially-demodulated minimum shift keying (DMSK) without error-control coding, in AWGN and Rayleigh flat fading. The coded system shows that significant performance improvement can be obtained through simple coding, particularly in Rayleigh flat fading.

# Acknowledgments

First, I must thank my supervisor—Professor Desmond P. Taylor—for his excellent supervision of my research. I hope he will find this thesis "Good enough".

I would like to thank Steve Bly, Umberto Mengali, Aaron Gulliver and Peter Smith for their assistance with various parts of this work.

Many thanks must also go to the Comms Lab Crew—Matt, Brian, Rob, Ben B, Adam, John, Rich, Leon, Nick, Steve, Katharine, Peter, Thawatt, Perapol, Ben S, Phil and Andrew—for their friendship and assistance.

There have been too many friends to list over the course of the thesis, but some deserve a special mention. They are—in semi-chronological order—Jason, Hamish, Tim, Anna, Nick, Charlotte, Paul Moody, Callum, Rob B, Greg, Rich B, Jim, James, Mike, Rich L, John, Bree, Chris, Pat, Josh, Craig, the Pfahlert family, Aman, the other members of Barbarella—Simon, Phil and Jason—Mal, Harley, Andy, Brighid, Matt S, Jacquie, Ruth, Steve S, Matt M, Tom, Naomi, Sian, Stasa, Brylee, Sylvia, Jane B, Jane H, Ruby, Paul Mooney, the other members of Estrada—Chris, Dean, and Tim—Dan, Nigel, Steve B, Ngaere, Rob F, Mary, Daemon and Vana.

I could not have achieved this without the support of my family, and they deserve my very, very special thanks. So Marilyn, Tim, Jim, Olwyn and Michele—I am indebted to you all.

This thesis is dedicated to all my parents and grandparents,
without whom I would not be here.

ANTHONY GRIFFIN

*The University of Canterbury*
*February 2000*

# Contents

**Abstract** iii

**Acknowledgments** v

**List of Figures** xiii

**List of Tables** xvii

**Chapter 1 Introduction** 1

1.1 Introduction . . . . . 1

1.2 Background . . . . . 1

1.3 Scope . . . . . 2

1.4 Overview . . . . . 3

1.5 Contributions . . . . . 4

1.6 Publications . . . . . 5

**Chapter 2 Continuous Phase Frequency Shift Keying** 7

2.1 Introduction . . . . . 7

2.2 Continuous Phase Frequency Shift Keying . . . . . 8

2.3 Phase Tree of CPFSK . . . . . 9

2.4 Phase Trellis of CPFSK . . . . . 11

2.5 The Decomposition of CPFSK . . . . . 13

2.5.1 Memoryless Modulator . . . . . 13

2.5.2 Continuous Phase Encoder . . . . . 15

2.6 Optimum detection of CPFSK . . . . . 16

2.7 Coherent Receiver for CPFSK . . . . . 17

2.7.1 Coherent Demodulator . . . . . 17

2.7.2 Viterbi Processor . . . . . 21

2.8 Euclidean Distance Properties of CPFSK . . . . . 21
2.8.1 Normalised Incremental Squared Euclidean Distance of CPFSK . . . 23
2.8.2 Normalised Minimum Squared Euclidean Distance for CPFSK . . . 23
2.9 Error Performance of CPFSK . . . . . 23
2.10 Simulation of a CPFSK system . . . . . 24
2.11 Summary . . . . . 24

**Chapter 3 Differential Continuous Phase Frequency Shift Keying 27**
3.1 Introduction . . . . . 27
3.2 Differential Demodulation . . . . . 28
3.3 Differentially Encoding CPFSK . . . . . 32
3.4 The Decomposition of DCPFSK . . . . . 36
3.5 Phase Tree . . . . . 37
3.6 Phase Trellis . . . . . 40
3.7 Detection of DCPFSK . . . . . 42
3.8 Receiver Structure for DCPFSK . . . . . 43
3.8.1 Viterbi Processor . . . . . 44
3.9 Euclidean Distance Properties of DCPFSK . . . . . 44
3.9.1 Incremental Squared Euclidean Distance . . . . . 46
3.9.2 Minimum Squared Euclidean Distance . . . . . 49
3.10 Error Performance of DCPFSK . . . . . 57
3.11 Simulation of a DCPFSK system . . . . . 58
3.12 Summary . . . . . 59

**Chapter 4 Coding DCPFSK 61**
4.1 Introduction . . . . . 61
4.2 Coding CPFSK . . . . . 62
4.3 Feedback-free Continuous Phase Encoder . . . . . 62
4.4 Channel Encoder . . . . . 63
4.5 Code Search Model for CPFSK . . . . . 65
4.6 Overall Encoder in a Coded CPFSK System . . . . . 67
4.6.1 Rate-1/2 Ring-Coded 4-CPFSK Overall Encoder Example . . . . . 67
4.7 Implementation of a Coded CPFSK System . . . . . 70
4.7.1 Rate-1/2 Ring-Coded 4-CPFSK Implementation Example . . . . . 70
4.8 Euclidean Distance Model of DCPFSK . . . . . 70

4.9 Code Search Model for DCPFSK . . . . . 73
4.10 Overall Encoder in a Coded DCPFSK System . . . . . 75
4.10.1 Rate-1/2 Ring-Coded 4-DCPFSK Overall Encoder Example . . . . . 75
4.11 Implementation of a Coded DCPFSK System . . . . . 77
4.11.1 Rate-1/2 Ring-Coded 4-DCPFSK Implementation Example . . . . . 77
4.12 Code Search Procedure . . . . . 78
4.13 Code Search Results . . . . . 78
4.14 Summary . . . . . 80

**Chapter 5 Performance of Coded (D)CPFSK Systems in AWGN 81**

5.1 Introduction . . . . . 81
5.2 Probability of Error Analysis . . . . . 82
5.3 Calculating the Error Coefficient . . . . . 87
5.4 Performance of Coded (D)CPFSK Systems . . . . . 87
5.4.1 Performance of 4-state, rate-1/2 encoded 4-CPFSK . . . . . 88
5.4.2 Performance of 4-state, rate-1/2 encoded 4-DCPFSK . . . . . 90
5.4.3 Performance of 8-state, rate-1/2 encoded 4-CPFSK . . . . . 91
5.4.4 Performance of 8-state, rate-1/2 encoded 4-DCPFSK . . . . . 91
5.4.5 Performance of 8-state, rate-2/3 encoded 8-CPFSK . . . . . 93
5.4.6 Performance of 8-state, rate-2/3 encoded 8-DCPFSK . . . . . 95
5.5 Performance of Uncoded (D)CPFSK Systems . . . . . 95
5.6 Comparison of Uncoded DMSK with Rate-1/2 Encoded 4-DCPFSK . . . . 99
5.7 Summary . . . . . 100

**Chapter 6 Performance of DCPFSK Systems in Rayleigh Flat Fading 101**

6.1 Introduction . . . . . 101
6.2 The Differentially-Demodulated Received Signal . . . . . 102
6.3 Pairwise Probability of Error of $M$-DCPFSK in Rayleigh Flat Fading . . . 103
6.3.1 Form of the Characteristic Function . . . . . 106
6.3.2 Calculating the Pairwise Probability of Error . . . . . 109
6.3.3 Calculating the Poles and Residues . . . . . 110
6.4 Performance of Uncoded $M$-DCPFSK in Rayleigh Flat Fading . . . . . 111
6.4.1 Approximate Performance Calculations . . . . . 112
6.4.2 Computed and Simulated Performance of Uncoded $M$-DCPFSK . . 113

6.5 Performance of Rate-1/2 Encoded 4-DCPFSK in Rayleigh Flat Fading . . . 115
6.6 Comparison of Uncoded DMSK with Rate-1/2 Encoded 4-DCPFSK . . . . 120
6.7 Summary . . . 123

**Chapter 7 Conclusions and Future Work** **125**
7.1 Conclusions . . . 125
7.2 DCPFSK Evaluation . . . 127
7.3 Future Work . . . 128

**Appendix A Properties of the Modulo Operator** **129**

**Appendix B Complex Noise** **131**

**Appendix C The Viterbi Algorithm** **133**
C.1 General statement of the problem . . . 133
C.2 The algorithm . . . 134
C.2.1 Storage . . . 135
C.2.2 Operation . . . 135
C.2.3 Complexity . . . 135

**Appendix D CPFSK Simulations** **137**
D.1 A Sampling Receiver Structure for CPFSK . . . 137
D.1.1 Sampler . . . 137
D.1.2 Viterbi Processor . . . 139
D.2 CPFSK Simulation Model . . . 141

**Appendix E CPFSK Sampling Receiver Performance** **143**

**Appendix F DCPFSK Simulations** **149**
F.1 A Sampling Receiver Structure for DCPFSK . . . 149
F.1.1 Sampler . . . 149
F.1.2 Viterbi Processor . . . 150
F.2 DCPFSK Simulation Models . . . 151

**Appendix G Full Code Search Results** **153**

**Appendix H Glossary of Abbreviations** **157**

Appendix I Glossary of Symbols 159

Bibliography 165

# List of Figures


2.1 Phase response for CPFSK . . . . . 9
2.2 Phase tree examples for CPFSK . . . . . 10
2.3 Phase trellis examples for CPFSK . . . . . 12
2.4 Decomposition of CPFSK . . . . . 13
2.5 Memoryless modulator . . . . . 14
2.6 Continuous phase encoder . . . . . 15
2.7 A MLSE receiver structure for coherent CPFSK . . . . . 17
2.8 Coherent demodulator . . . . . 18
2.9 Detection Processor . . . . . 22
2.10 Computed and simulated performance of $M$-CPFSK . . . . . 25

3.1 Differential demodulator . . . . . 29
3.2 Differentially-encoded CPFSK system . . . . . 32
3.3 Differential encoder . . . . . 35
3.4 Decomposition of DCPFSK . . . . . 36
3.5 Differential continuous phase encoder . . . . . 37
3.6 Phase trees for DMSK . . . . . 38
3.7 Phase trees for 4-DCPFSK . . . . . 39
3.8 Phase trellis examples for modulated DCPFSK . . . . . 40
3.9 Phase trellis examples for demodulated DCPFSK . . . . . 41
3.10 A receiver structure for DCPFSK . . . . . 43
3.11 Detection Processor . . . . . 45
3.12 Computed and simulated performance of $M$-DCPFSK . . . . . 59

4.1 Coded CPFSK system . . . . . 62
4.2 Removing the feedback in the CPE . . . . . 63
4.3 Rate-$(l-1)/l$ systematic ring convolutional encoder . . . . . 64

4.4 Coded CPFSK system with scrambler . . . . . 66
4.5 Coded CPFSK system model for code searches . . . . . 66
4.6 Implementation of $\mathbf{G}(D) = \left[1 \ \frac{1}{1+2D}\right]$ . . . . . 68
4.7 Trellis diagram of $\mathbf{G}(D) = \left[1 \ \frac{1}{1+2D}\right]$ . . . . . 68
4.8 Implementation of $\mathbf{G}(D)\,\mathbf{C}_2(D) = \left[\frac{1+D}{1+2D} \ \frac{D}{1+2D} \ \frac{3+2D}{1+2D} \ 1\right]$ . . . . . 69
4.9 Trellis diagram of $\mathbf{G}(D)\,\mathbf{C}_2(D) = \left[\frac{1+D}{1+2D} \ \frac{D}{1+2D} \ \frac{3+2D}{1+2D} \ 1\right]$ . . . . . 69
4.10 Actual implementation of a coded CPFSK system . . . . . 70
4.11 Implementation of $\dot{\mathbf{G}}(D) = \left[\frac{1+D}{1+2D} \ \frac{3+2D}{1+2D}\right]$ . . . . . 71
4.12 Trellis diagram of $\dot{\mathbf{G}}(D) = \left[\frac{1+D}{1+2D} \ \frac{3+2D}{1+2D}\right]$ . . . . . 71
4.13 Differential decoder . . . . . 72
4.14 Differential continuous phase encoder/decoder . . . . . 72
4.15 Euclidean distance model of DCPFSK . . . . . 73
4.16 Feedback-free DCPED . . . . . 73
4.17 Coded DCPFSK system with scrambler . . . . . 74
4.18 Coded DCPFSK system model for code searches . . . . . 74
4.19 Implementation of $\mathbf{G}(D)\,\dot{\mathbf{F}}_2(D) = \left[\frac{1+2D \boxminus D}{1+2D} \ \frac{D}{1+2D} \ \frac{1 \boxminus (1+2D)}{1+2D} \ 1\right]$ . . . . . 76
4.20 Trellis diagram of $\mathbf{G}(D)\,\dot{\mathbf{F}}_2(D) = \left[\frac{1+2D \boxminus D}{1+2D} \ \frac{D}{1+2D} \ \frac{1 \boxminus (1+2D)}{1+2D} \ 1\right]$ . . . . . 76
4.21 Implementation of a coded DCPFSK system . . . . . 77

5.1 A transmitted path in the trellis of the overall encoder . . . . . 82
5.2 An example of an error event in a four-state trellis. . . . . 83
5.3 A specific error event $\mathcal{E}_{\mathcal{S},m}$ starting from state $\mathcal{S}$ at time $k_V$ . . . . . 84
5.4 Performance of 4-state, rate-1/2 encoded 4-CPFSK . . . . . 88
5.5 Performance of 4-state, rate-1/2 encoded 4-DCPFSK . . . . . 90
5.6 Performance of 8-state, rate-1/2 encoded 4-CPFSK . . . . . 91
5.7 Performance of 8-state, rate-1/2 encoded 4-DCPFSK . . . . . 93
5.8 Performance of 8-state, rate-2/3 encoded 8-CPFSK . . . . . 94
5.9 Performance of 8-state, rate-2/3 encoded 8-DCPFSK . . . . . 95
5.10 Computed and simulated performance of $M$-CPFSK . . . . . 96
5.11 Computed and simulated performance of $M$-DCPFSK . . . . . 98
5.12 Coded performance comparison in AWGN . . . . . 99

6.1 Power spectral densities of the Rayleigh fading components . . . . . 102
6.2 The closed curve $\Gamma$ . . . . . 109
6.3 Performance of uncoded DMSK in Rayleigh flat fading . . . . . 112

6.4 Pairwise probability of error of uncoded DMSK in Rayleigh flat fading . . . 113
6.5 Performance of uncoded 4-DCPFSK in Rayleigh flat fading . . . . . 114
6.6 Performance of uncoded 8-DCPFSK in Rayleigh flat fading . . . . . 115
6.7 Performance of coded 4-DCPFSK in Rayleigh flat fading . . . . . 117
6.8 Pairwise probability of error of coded 4-DCPFSK in Rayleigh flat fading . . 118
6.9 Performance of coded 4-DCPFSK in Rayleigh flat fading . . . . . 119
6.10 Coded performance comparison in Rayleigh flat fading with $f_D T = 0.1$ . . . 120
6.11 Coded performance comparison in Rayleigh flat fading with $f_D T = 0.01$ . . 121
6.12 Coded performance comparison in Rayleigh flat fading with $f_D T = 0.001$ . 122

D.1 A sampling receiver for coherent CPFSK . . . . . 137
D.2 Autocorrelation function of the low-pass filtered white Gaussian noise . . . 138
D.3 Model used to produce $y_c(iT_s, \boldsymbol{U})$ for simulations of CPFSK in AWGN . . . 141

F.1 A sampling receiver for DCPFSK . . . . . 149
F.2 Model used to produce $y_d(iT_s, \boldsymbol{\beta})$ for simulations of DCPFSK in AWGN . . 151
F.3 Model used for simulations of DCPFSK in fading . . . . . 152

# List of Tables


2.1 $d^2_{\min}$ for various CPFSK schemes . . . . . 24

3.1 $d^2_{\min}$ for various DCPFSK schemes . . . . . 58
3.2 Comparison of empirical SNR values required to achieve a BER of $10^{-5}$ . . 59

4.1 Search results for rate-1/2 encoded 4-CPFSK and 4-DCPFSK . . . . . 79
4.2 Search results for rate-2/3 encoded 8-CPFSK and 8-DCPFSK . . . . . 79

5.1 Error coefficients for 4-state, rate-1/2 encoded 4-(D)CPFSK (depth 2) . . . 89
5.2 Error coefficients for 4-state, rate-1/2 encoded 4-(D)CPFSK (depth 6) . . . 89
5.3 Error coefficients for 8-state, rate-1/2 encoded 4-(D)CPFSK (depth 6) . . . 92
5.4 Error coefficients for 8-state, rate-2/3 encoded 8-(D)CPFSK (depth 2) . . . 94
5.5 Error coefficients for uncoded (D)MSK (depth 4) . . . . . 97
5.6 Error coefficients for uncoded 4-(D)CPFSK (depth 4) . . . . . 97
5.7 Error coefficients for uncoded 8-(D)CPFSK (depth 4) . . . . . 97

6.1 Data sequences to produce error events for coded 4-DCPFSK . . . . . 116
6.2 Data sequences to produce error events for coded 4-DCPFSK . . . . . 116
6.3 Error floor comparison . . . . . 122

F.1 Normalised cut-off frequency of noise-limiting filter . . . . . 152

G.1 Search results for rate-1/2 encoded 4-CPFSK and 4-DCPFSK . . . . . 154
G.2 Further search results for rate-1/2 encoded 4-DCPFSK . . . . . 155
G.3 Further search results for rate-1/2 encoded 4-DCPFSK . . . . . 155
G.4 Search results for rate-2/3 encoded 8-CPFSK and 8-DCPFSK . . . . . 156

# Chapter 1

# Introduction

## 1.1 Introduction

Telecommunications—and particularly digital communications—are a very important part of today's society. People like being able to communicate across long distances. This communication can be the most common of human communications—that of voice—or it can be the more recent and rapidly-growing field of computer data. No matter how advanced communication systems become, high-end users want to be able to communicate faster and more cheaply. At the same time, there are many parts of the world that do not have even the most basic communication systems. This means that there is an enormous range of communication needs and desires in the world, and that there is scope for the development of many different communication systems and products.

It is perhaps at the lower end of the spectrum that the research in this thesis is placed. We seek to further the development and analysis of a simple wireless communication scheme that will provide a low-cost, low-to-medium capacity digital service.

## 1.2 Background

Our communication scheme is called differentially-encoded and differentially-demodulated continuous phase frequency shift keying (DCPFSK). DCPFSK is based on continuous phase frequency shift keying (CPFSK) which is a simple continuous phase modulation (CPM) scheme. The main advantage of CPM schemes is the fact that they are constant-envelope modulations. This permits the amplifiers in CPM schemes to run in their non-linear but power-efficient regions. There are many different and complex CPM schemes, of which CPFSK is one of the simplest. In turn, the simplest CPFSK scheme is known

as minimum shift keying (MSK) and is very widely used, a particular example being the Global System for Mobile Communications (known as GSM) that uses Gaussian minimum shift keying (GMSK).

The state of the art of CPFSK systems up to 1986 is very well presented in [AAS86]. From then on the most important development was that of Rimoldi in [Rim88], who showed that the important class of CPFSK schemes with modulation index $h = K/P$ where $K$ and $P$ are relatively prime positive integers, could be represented by a decomposed model [Rim88] consisting of a continuous phase encoder (CPE) and a memoryless modulator (MM). This model provides insight into CPFSK, and allows its inherent coding and modulation to be studied individually. Rimoldi showed how this model could be used to design codes specifically for CPFSK in [Rim89]. Yang and Taylor went further and developed codes over rings for CPFSK using the decomposed model in [YT94]. They found codes that performed better than previous techniques developed without the model. Rimoldi and Li found similar results in [RL95].

The coded CPFSK schemes use coherent demodulation which requires an accurate carrier reference. Acquiring the carrier in more extreme channels can be difficult and/or very computationally intensive. Differential demodulation uses the previous symbol to demodulate the current signal, thereby avoiding the need for a carrier reference at the receiver at the expense of incurring a performance penalty. Differential decoding also modifies the transmitted data. To avoid this, a differential encoder can be used. A standard differential encoder—such as that used in differential phase shift keying (DPSK)—does not work with CPFSK as it removes CPFSK's inherent coding and suffers a further performance loss. Differentially-demodulated CPFSK has been discussed in various papers [AS81], [Mas90], [YL90] and [SZ98] for example, but only binary CPFSK systems are considered, and no differential encoder is proposed. A differential encoder that preserves the inherent coding has been developed in [YT92], but it too only works with binary CPFSK schemes, and does not interface well with the decomposed model.

We seek to design and analyse coding schemes for CPFSK that work well with differential demodulation in additive white Gaussian noise (AWGN) and Rayleigh flat fading.

## 1.3 Scope

We consider only the AWGN and Rayleigh flat fading channels. No intersymbol interference (ISI) other than that introduced by fading or filters at the front end of the receiver is

present in the channel. Additionally, we do not restrict the bandwidth of the transmitted signal, as we wish to determine the optimum performance of our system. We do not consider synchronisation issues, although the only synchronisation that the DCPFSK receiver would require is that of symbol timing. Finally, we only consider the code structures used in [YT94] as they were shown to perform the best.

## 1.4 Overview

In Chapter 2 we present and discuss coherently-demodulated CPFSK and its decomposed model and other relevant background material. We then discuss differential demodulation and develop a differential encoder that interfaces very well with the decomposed model and enables CPFSK to be differentially demodulated without losing its structure in Chapter 3. Our DCPFSK receiver structure is based on minimising the squared Euclidean distance (SED) between the received signal and all the possibly-transmitted signals. To aid us in determining the receiver's error performance, we calculate the normalised incremental SED (NISED) and normalised minimum SED (NMSED) of DCPFSK. We approximate the theoretical error performance of DCPFSK in additive white Gaussian noise (AWGN) and compare this to simulation results.

Chapter 4 discusses the coding method and models used in [YT94] with the aid of some examples. We develop the DCPFSK models required to perform similar code searches for DCPFSK. We show how these codes interface with DCPFSK and how they can be implemented through examples. We present the results of code searches that show that very little NMSED is lost between a coded CPFSK and a coded DCPFSK scheme.

We look at the performance of coded CPFSK and DCPFSK systems in AWGN in Chapter 5. We use some of the work in [YT94] to derive an expression for a union bound on the bit error probability of coded CPFSK and DCPFSK systems in AWGN. We then apply a limited-length, exhaustive search to find the significant terms in the union bound and compare the computed results to those generated in simulations. In almost all the cases the two sets of curves agree very well.

In Chapter 6 we investigate the performance of DCPFSK in various Rayleigh flat fading channels. A CPFSK receiver using coherent demodulation will not work at all in a fading channel without additional algorithms to recover the faded carrier. The DCPFSK receiver performs reasonably well in fading as the inherent operation of the differential demodulator partially compensates for the fading. We first determine the form of the faded and differentially-demodulated received signal, and then develop the pairwise probability

of error of DCPFSK in fading using the Gaussian quadratic form and the residue theorem. Using the union bound techniques of Chapter 5, we calculate the error performance of coded and uncoded DCPFSK in fading and compare the computed curves with simulated results. Irreducible bit error rates are observed, as the effect of the deep nulls and rapid phase changes associated with Rayleigh flat fading cannot be totally removed. In comparisons between comparable coded and uncoded systems, the coding is seen to significantly lower the irreducible error rates.

The work in the thesis is summarised in Chapter 7. We also draw some conclusions and discuss future research options that have been opened up.

## 1.5 Contributions

The original contributions in this thesis are Chapters 3 and 6 and parts of Chapters 4 and 5. Specifically they are:

- Development of a differential encoder that preserves the phase trellis of CPFSK through differential demodulation and interfaces well with the decomposed model of CPFSK.
- Analysis and discussion of differentially-encoded and differentially-demodulated continuous phase frequency shift keying (DCPFSK) including the calculation of the normalised incremental squared Euclidean distance and the normalised minimum squared Euclidean distance and the development of an approximate performance measure.
- Development of a code search model for $M$-DCPFSK systems ($M$-ary DCPFSK systems with $h = 1/M$).
- New codes for $M$-DCPFSK systems.
- Calculation of the performance of selected coded $M$-DCPFSK systems in AWGN, including the finding of significant distance terms and error coefficients for selected coded $M$-DCPFSK systems.
- Analysis and calculation of the performance of selected $M$-DCPFSK systems (both coded and uncoded) in Rayleigh flat fading.

## 1.6 Publications

The following papers have been written on the work in this thesis, and have been published, or are awaiting publication:

- A. Griffin and D. P. Taylor, "On differentially demodulated CPFSK," in *Proc. IEEE International Conference on Communications*, Dallas, TX, June 1996, vol. 1, pp. 354–358.
- A. Griffin and D. P. Taylor, "Coding CPFSK for differential demodulation," Accepted for publication in *IEEE Transactions on Communications.*

# Chapter 2

# Continuous Phase Frequency Shift Keying

## 2.1 Introduction

Continuous phase frequency shift keying (CPFSK) is a true constant-envelope modulation, which makes it an attractive communication scheme for wireless communications. It is basically frequency shift keying (FSK) with the added restriction that the phase is continuous between symbol intervals. This restriction decreases the bandwidth requirement. The phase-continuity introduces a definite trellis structure into CPFSK, which complicates analysis and understanding. A decomposed model of CPFSK was presented in [Rim88], that allows CPFSK's inherent coding and modulation to be explored independently, and provides insight into this important communications scheme.

We present a mathematical description of CPFSK in Section 2.2. This provides the necessary definitions to proceed. We then discuss the phase tree generated by CPFSK in Section 2.3 and develop the phase trellis of CPFSK in Section 2.4. This leads to the decomposed model of CPFSK which is described in Section 2.5. The optimum detection of CPFSK is discussed in Section 2.6, and a receiver that performs this optimal detection is presented in Section 2.7. The distance properties of CPFSK are presented in Section 2.8. From there the error performance of CPFSK is discussed in Section 2.9. In Section 2.10 we present our simulation results and compare them to computed curves. Finally, in Section 2.11 we present a brief summary of the chapter.

## 2.2 Continuous Phase Frequency Shift Keying

A CPFSK signal can be described by [Rim88]

$$s(t, \boldsymbol{U}) = \sqrt{\frac{2E}{T}} \cos(2\pi f_1 t + \psi(t, \boldsymbol{U}) + \varphi_0), \quad t \geq 0, \tag{2.1}$$

where $E$ is the symbol energy, $T$ is the symbol period, $\varphi_0$ is the initial phase offset, and $f_1$ the asymmetric carrier frequency, which is related to the symmetric carrier frequency $f_c$ by

$$f_1 = f_c - (M-1)\frac{h}{2T}. \tag{2.2}$$

$\psi(t, \boldsymbol{U})$ is called the *tilted (information-carrying) phase*, and is given by

$$\psi(t, \boldsymbol{U}) = 4\pi h \sum_{n=0}^{\infty} U_n q(t - nT), \quad t \geq 0. \tag{2.3}$$

This is assumed to be 0 at $t = 0$. The $M$-ary *data sequence* $\boldsymbol{U}$ is given by

$$\boldsymbol{U} = (U_0, U_1, \dots), \qquad U_n \in \{0, 1, \dots, (M-1)\}. \tag{2.4}$$

We assume that the symbols $U_n$ are independent and equiprobable. The symbol energy $E$ is related to the bit energy $E_b$ by

$$E = E_b \cdot r \cdot \log_2 M, \tag{2.5}$$

where $r$ is the rate of any external error-control coding (ECC), that is, the average number of information bits per symbol period $T$. In the case of no ECC, $r = 1$.

The parameter $h$ in (2.2) and (2.3) is called the *modulation index*. We consider here only rational modulation indices of the form

$$h = \frac{K}{P}, \tag{2.6}$$

where $K$ and $P$ are positive integers with no common factors. This ensures that the phase of the CPFSK signal has a structured form as we will see in Sections 2.3 and 2.4. The *phase response* $q(t)$ for CPFSK is

$$q(t) = \begin{cases} 0, & t \leq 0 \\ t/2T, & 0 < t \leq T \\ \frac{1}{2}, & t > T. \end{cases} \tag{2.7}$$

Figure 2.1 illustrates $q(t)$. It is the fact that $q(t)$ is continuous and semi-infinite that

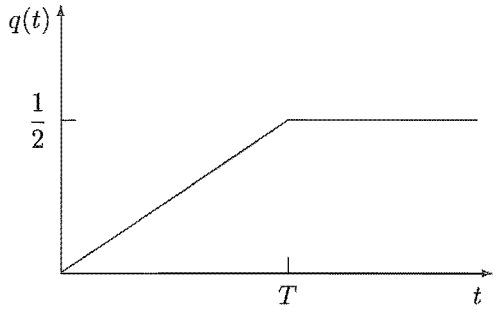



**Figure 2.1:** Phase response for CPFSK

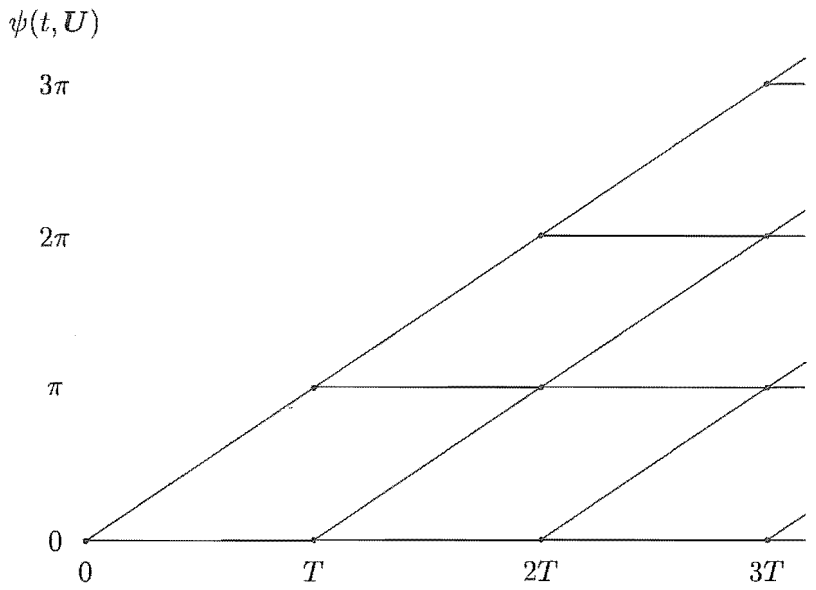
ensures that the phase is continuous in CPFSK.

The two most important parameters in a CPFSK scheme are $h$ and $M$, as they determine the scheme's information bit-rate, band-width and performance. As we will see later, an important class of CPFSK schemes results when $h = 1/M$. An example is minimum shift keying (MSK) which is CPFSK with $M = 2$ and $h = 1/2$. We will refer to an $M$-ary CPFSK scheme with $h = 1/M$ as $M$-CPFSK. Thus 4-CPFSK denotes quaternary CPFSK with $h = 1/4$, and MSK could be referred to as 2-CPFSK.

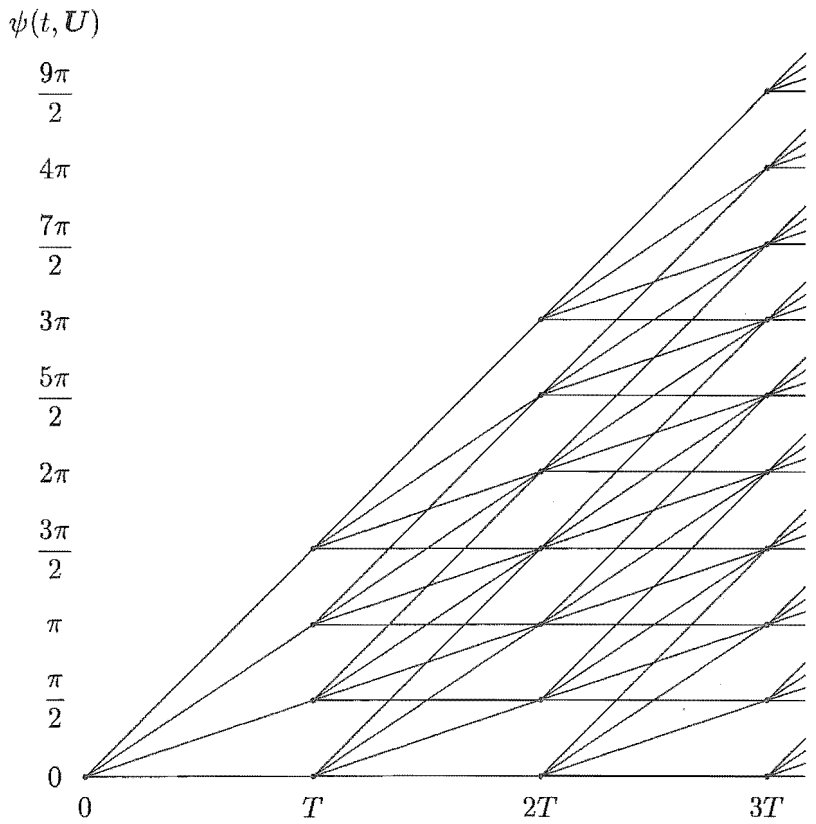
## 2.3 Phase Tree of CPFSK

If we take the phase of a CPFSK signal (described by (2.1)) and take out the contribution of the carrier frequency term $2\pi f_1 t$ and the initial phase offset $\varphi_0$, we are left with the tilted phase term $\psi(t, \boldsymbol{U})$. Assuming that $\psi(t, \boldsymbol{U})$ is zero at time $t = 0$, we can show its development over time by hypothesising all the possible values the elements of $\boldsymbol{U}$ can take. This is called a phase tree. An example for MSK is shown in Figure 2.2(a). A more complicated phase tree is shown in Figure 2.2(b), that of 4-CPFSK.

Looking closely at (2.3), we can see that $\psi(t, \boldsymbol{U})$ is made up of various scaled and time-shifted versions of $q(t)$. The superposition of these semi-infinite phase responses can be clearly seen in the phase trees. At each branch point, $M$ different phase trajectories emerge, corresponding to $M$ different frequencies, and the $M$ possible transmitted symbols. The phase tree also shows us that in each symbol period, the CPFSK signal also contains the "memory" of all the previous symbols, which is the branch point in the tree, or the starting phase of the symbol. Unfortunately, the phase tree expands with each symbol period and quickly becomes too complex to be useful.




(a) MSK ($M = 2$, $h = 1/2$)




(b) 4-CPFSK ($M = 4$, $h = 1/4$)

**Figure 2.2:** Phase tree examples for CPFSK

## 2.4 Phase Trellis of CPFSK

A more useful form of signal phase information is the phase trellis. Let us define the modulo-$x$ operator as $R_x[\theta]$, which denotes the remainder after $\theta$ is divided by $x$ (the modulo operator is defined and discussed in Appendix A). Using the fact that the cosine function operates modulo-$2\pi$ on its phase, and noting the property (A.2), we can write

$$\begin{aligned} R_{2\pi}\left[2\pi f_1 t + \psi(t,\boldsymbol{U}) + \varphi_0\right] &= R_{2\pi}\left[2\pi f_1 t + R_{2\pi}\left[\psi(t,\boldsymbol{U})\right] + \varphi_0\right] \\ &= R_{2\pi}\left[2\pi f_1 t + \bar{\psi}(t,\boldsymbol{U}) + \varphi_0\right]. \end{aligned} \tag{2.8}$$

We call $\bar{\psi}(t,\boldsymbol{U})$ the *physical* tilted phase, defined as

$$\bar{\psi}(t,\boldsymbol{U}) \triangleq R_{2\pi}\left[\psi(t,\boldsymbol{U})\right]. \tag{2.9}$$

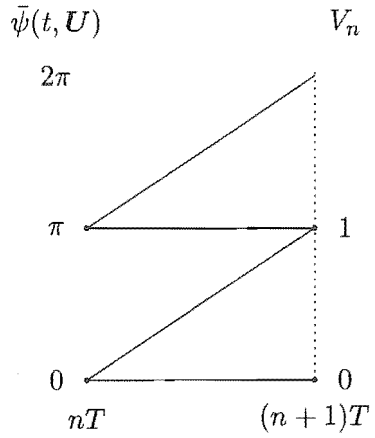
Letting $t = \tau + nT$, and noting the properties (A.2) and (A.3), we can further develop $\bar{\psi}(t,\boldsymbol{U})$ as

$$\begin{aligned} \bar{\psi}(t,\boldsymbol{U}) &= \bar{\psi}(\tau + nT, \boldsymbol{U}), && 0 \le \tau < T \\ &= R_{2\pi}\left[4\pi h \sum_{i=0}^{n} U_i\, q(\tau + nT - iT)\right], && 0 \le \tau < T \\ &= R_{2\pi}\left[2\pi h \sum_{i=0}^{n-1} U_i + 2\pi h U_n \frac{\tau}{T}\right], && 0 \le \tau < T \\ &= R_{2\pi}\left[R_{2\pi}\left[2\pi\left(\frac{K}{P}\right)\sum_{i=0}^{n-1} U_i\right] + 2\pi h U_n \frac{\tau}{T}\right], && 0 \le \tau < T \\ &= R_{2\pi}\left[2\pi\left(\frac{K}{P}\right) R_P\left[\sum_{i=0}^{n-1} U_i\right] + 2\pi h U_n \frac{\tau}{T}\right], && 0 \le \tau < T \\ &= R_{2\pi}\left[2\pi h\left(V_n + U_n \frac{\tau}{T}\right)\right], && 0 \le \tau < T. \end{aligned} \tag{2.10}$$

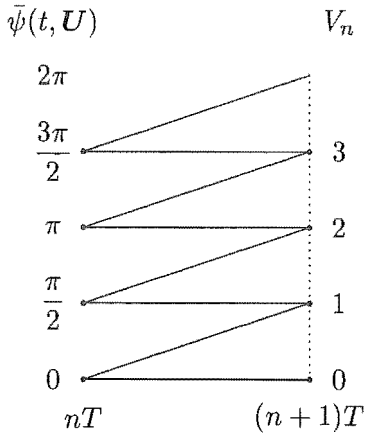
The quantity $V_n$ is called the accumulated symbol phase, and represents the contribution of all the previously transmitted symbols. It is defined as

$$V_n \triangleq R_P\left[\sum_{i=0}^{n-1} U_i\right]. \tag{2.11}$$

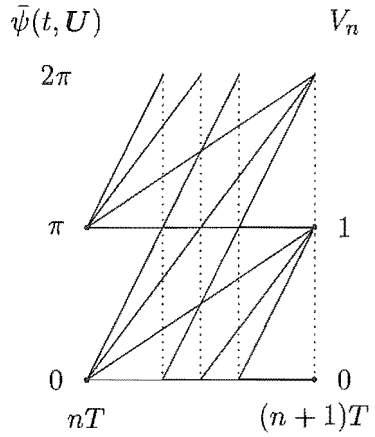
We can see from (2.10) that the phase of a CPFSK signal has a very structured form and depends on two terms. The $U_n$ term is a frequency component that depends only on the current symbol, and the $V_n$ term is a starting phase that depends on all the previous transmitted symbols and creates the trellis. It is this $V_n$ term that ensures the phase is continuous in CPFSK. The number of possible starting phases at time $nT$ is $P$, and the physical tilted phase is equal to $2\pi h V_n$ at time $nT$.



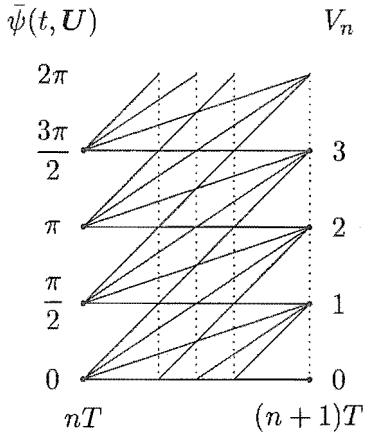

(a) MSK ($M = 2$, $h = 1/2$)




(b) CPFSK with $M = 2$ and $h = 1/4$




(c) CPFSK with $M = 4$ and $h = 1/2$




(d) 4-CPFSK ($M = 4$, $h = 1/4$)

**Figure 2.3:** Phase trellis examples for CPFSK

A plot of $\bar{\psi}(t, \boldsymbol{U})$ produces a phase trellis, which shows the possible phase trajectories at any point in time, but unlike the phase tree, does not grow with each symbol period. It is the fact that the phase trellis is time-invariant that makes it more useful than the phase tree. We can think of the modulo-$2\pi$ operator as "wrapping" the phase tree around a cylinder so that all multiples of $2\pi$ sit on top of each other. The remnant around the cylinder in one symbol period is the phase trellis. The phase trellises for various CPFSK schemes are shown in Figure 2.3. The phase trellises of Figures 2.3(a) and 2.3(d) correspond to the phase trees of Figures 2.2(a) and 2.2(b) respectively. The dotted lines show where the phase is continuous modulo-$2\pi$. From each of the $P$ phase states, $M$ different phase trajectories emanate, each one corresponding to one of the $M$ different

symbols. Thus there are $M \cdot P$ different signals possible in any one symbol period, and they are clearly shown on the phase trellis. The phase trellis is a very important aid to the understanding of CPFSK, and is analogous to the signal-constellation diagrams used in PSK.

## 2.5 The Decomposition of CPFSK

Using the definitions in Sections 2.2 and 2.4, CPFSK can be decomposed into a two-part model, consisting of a continuous phase encoder (CPE) and a memoryless modulator (MM)[Rim88], as shown in Figure 2.4. This provides for greater understanding of CPFSK, as its inherent coding and modulation aspects are isolated and can be studied independently. We now discuss each part.

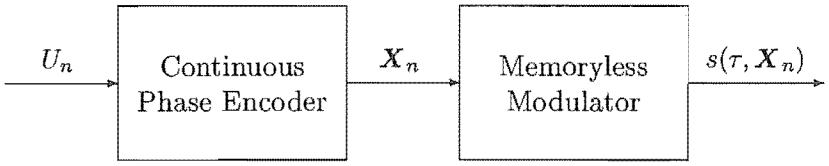



**Figure 2.4:** Decomposition of CPFSK

### 2.5.1 Memoryless Modulator

The advantage of the memoryless modulator is that it needs no knowledge of the previous transmitted signal. We have seen in Section 2.4 that the physical tilted phase has a time invariant-trellis. Thus the modulator only needs to know which of the $M \cdot P$ different signals are to be transmitted. We define the input to the memoryless modulator as

$$\boldsymbol{X}_n \triangleq \begin{bmatrix} X_n^{(1)} \\ X_n^{(2)} \end{bmatrix}. \tag{2.12}$$

The $M$-ary value $X_n^{(1)}$ selects the frequency to be transmitted, and the $P$-ary value $X_n^{(2)}$ selects its starting phase. In the coherent case with no external ECC,

$$X_n^{(1)} = U_n \qquad \text{and} \qquad X_n^{(2)} = V_n. \tag{2.13}$$

Following [Rim88], we write

$$\bar{\psi}(\tau, \boldsymbol{X}_n) \text{ instead of } \bar{\psi}(\tau + nT, \boldsymbol{U}), \qquad 0 \leq \tau < T,$$

and

$$s(\tau, \boldsymbol{X}_n) \text{ instead of } s(\tau + nT, \boldsymbol{U}), \qquad 0 \le \tau < T,$$

allowing us to write the output of the memoryless modulator as

$$s(\tau, \boldsymbol{X}_n) = \sqrt{\frac{2E}{T}} \cos[2\pi f_1(\tau + nT) + \bar{\psi}(\tau, \boldsymbol{X}_n)], \qquad 0 \le \tau < T, \tag{2.14}$$

where we have chosen the initial phase $\varphi_0$ to be zero with no loss of generality, and

$$\bar{\psi}(\tau, \boldsymbol{X}_n) = R_{2\pi}\Big[2\pi h\Big(X_n^{(2)} + X_n^{(1)} \frac{\tau}{T}\Big)\Big], \qquad 0 \le \tau < T. \tag{2.15}$$

Decomposing (2.14) into in-phase and quadrature components, we have

$$\begin{aligned} s(\tau, \boldsymbol{X}_n) &= s_{\mathrm{I}}'(\tau, \boldsymbol{X}_n) \cos[2\pi f_1(\tau + nT)] \\ &\quad - s_{\mathrm{Q}}'(\tau, \boldsymbol{X}_n) \sin[2\pi f_1(\tau + nT)], \qquad 0 \le \tau < T. \end{aligned} \tag{2.16}$$

where $s_{\mathrm{I}}'(\tau, \boldsymbol{X}_n)$ and $s_{\mathrm{Q}}'(\tau, \boldsymbol{X}_n)$ are the in-phase and quadrature components of $s(\tau, \boldsymbol{X}_n)$ respectively, referenced to $f_1$. They are defined as

$$s_{\mathrm{I}}'(\tau, \boldsymbol{X}_n) \triangleq \sqrt{\frac{2E}{T}} \cos[\bar{\psi}(\tau, \boldsymbol{X}_n)], \qquad 0 \le \tau < T \tag{2.17}$$

$$s_{\mathrm{Q}}'(\tau, \boldsymbol{X}_n) \triangleq \sqrt{\frac{2E}{T}} \sin[\bar{\psi}(\tau, \boldsymbol{X}_n)], \qquad 0 \le \tau < T. \tag{2.18}$$

Figure 2.5 shows a possible realisation of the memoryless modulator as defined by (2.10)-(2.11) and (2.16). As $\boldsymbol{X}_n$ takes on a relatively small number of values, the calculation of the signal terms in Figure 2.5 could easily be implemented as lookup tables in a DSP

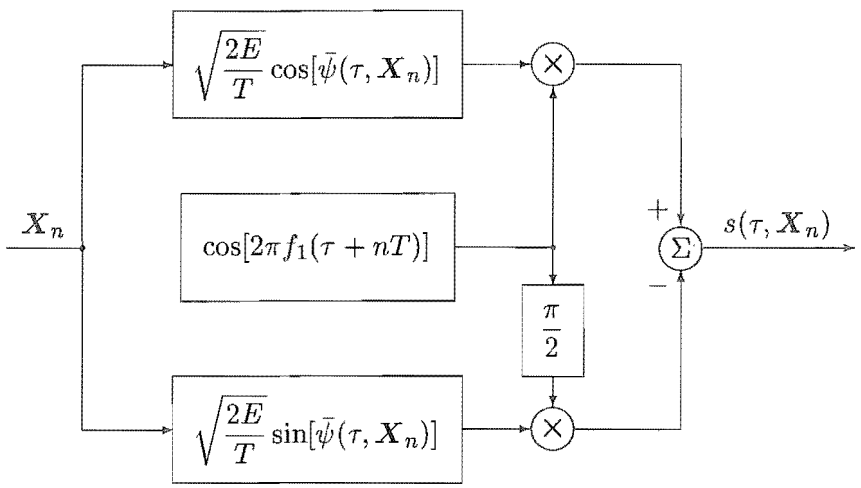



**Figure 2.5:** Memoryless modulator

implementation. Note that although we may generate the CPFSK signal in the manner above, we will also refer to it using

$$s(t, \boldsymbol{U}) \;=\; s_{\mathrm{I}}(t, \boldsymbol{U}) \cos(2\pi f_c t) - s_{\mathrm{Q}}(t, \boldsymbol{U}) \sin(2\pi f_c t), \tag{2.19}$$

where $s_{\mathrm{I}}(t, \boldsymbol{U})$ and $s_{\mathrm{Q}}(t, \boldsymbol{U})$ are the in-phase and quadrature components, referenced to $f_c$, of $s(t, \boldsymbol{U})$ respectively. They are defined by

$$s_{\mathrm{I}}(t, \boldsymbol{U}) \;\triangleq\; \sqrt{\frac{2E}{T}} \cos[\psi(t, \boldsymbol{U}) - 2\pi f_0 t] \tag{2.20}$$

$$s_{\mathrm{Q}}(t, \boldsymbol{U}) \;\triangleq\; \sqrt{\frac{2E}{T}} \sin[\psi(t, \boldsymbol{U}) - 2\pi f_0 t], \tag{2.21}$$

and $f_0$ is the difference between $f_c$ and $f_1$, defined as

$$f_0 \;\triangleq\; f_c - f_1 \;=\; (M-1)\frac{h}{2T}. \tag{2.22}$$

Including the $f_0$ term ensures that the spectra of the base-band quantities $s_{\mathrm{I}}(t, \boldsymbol{U})$ and $s_{\mathrm{Q}}(t, \boldsymbol{U})$ are symmetric around zero Hertz.

### 2.5.2 Continuous Phase Encoder

The purpose of the continuous phase encoder (CPE) is to generate the inputs to the memoryless modulator from the input data stream. These inputs are specified in (2.12) and (2.11), however a recursive form of (2.11) is required. Replacing $n$ by $n+1$ in (2.11) and again using (A.2), we obtain

$$V_{n+1} \;=\; R_P\left[\sum_{i=0}^{n} U_i\right] \;=\; R_P\left[\sum_{i=0}^{n-1} U_i + U_n\right] \;=\; R_P\left[R_P\left[\sum_{i=0}^{n-1} U_i\right] + U_n\right] \;=\; R_P[V_n + U_n]\,. \tag{2.23}$$

A diagram of the CPE to implement (2.12), (2.13) and (2.23) is shown in Figure 2.6. The addition in the CPE is modulo-$P$, and thus the CPE is a linear convolutional encoder over the ring of integers modulo-$P$ ($\mathbb{Z}_P$), which outputs $X_n$ and has the state $V_n$.

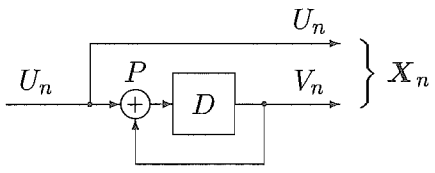



**Figure 2.6:** Continuous phase encoder

## 2.6 Optimum detection of CPFSK

We now discuss the optimum detection of CPFSK following [AAS86]. In the additive white Gaussian (AWGN) channel, the received signal $r(t, \boldsymbol{U})$ is given by

$$r(t, \boldsymbol{U}) \;=\; s(t, \boldsymbol{U}) + w(t), \tag{2.24}$$

where $s(t, \boldsymbol{U})$ was given in (2.19) and $w(t)$ is zero-mean Gaussian noise with a two-sided power spectral density (psd) $S_w(f)$ equal to $N_0/2$.

As the transmitted signal in each symbol interval depends on all the previously transmitted symbols, we must determine the state sequence (and the transitions) of the transmitter to detect the transmitted data $\boldsymbol{U}$. This is called maximum-likelihood sequence estimation (MLSE) [For72]. Using the time-invariant phase trellis and the memoryless modulator, the transmitter can be thought of as a Markov source, that is the transmitted signal $s(t, \boldsymbol{U})$ in the $n$-th symbol period depends only on the state of the CPE $V_n$ and the current information symbol $U_n$.

Let us use $\hat{\boldsymbol{U}}$ to denote a hypothesised transmitted data stream. As discussed in [AAS86], the MLSE receiver maximises the log likelihood function

$$\ln\left[\Pr\left\{r(t, \boldsymbol{U})|\hat{\boldsymbol{U}}\right\}\right] \;\approx\; -\int_{-\infty}^{\infty}\left[r(t, \boldsymbol{U}) - s(t, \hat{\boldsymbol{U}})\right]^2 dt \tag{2.25}$$

with respect to $\hat{\boldsymbol{U}}$. $\Pr\left\{r(t, \boldsymbol{U})|\hat{\boldsymbol{U}}\right\}$ is the probability density function (pdf) of the received signal $r(t, \boldsymbol{U})$ conditioned on $\hat{\boldsymbol{U}}$. The maximising sequence $\hat{\boldsymbol{U}}$ is the maximum-likelihood sequence estimate that minimises the squared Euclidean distance between the received signal $r(t, \boldsymbol{U})$ and a hypothesised transmitted signal. As CPFSK is a constant-envelope modulation, we can equivalently maximise the correlation

$$J_c(\hat{\boldsymbol{U}}) \;=\; \int_{-\infty}^{\infty} r(t, \boldsymbol{U})\, s(t, \hat{\boldsymbol{U}})\, dt. \tag{2.26}$$

To simplify processing, we define

$$J_{c,n}(\hat{\boldsymbol{U}}) \;=\; \int_{-\infty}^{(n+1)T} r(t, \boldsymbol{U})\, s(t, \hat{\boldsymbol{U}})\, dt, \tag{2.27}$$

which allows us to write the recursion equation

$$J_{c,n}(\hat{\boldsymbol{U}}) \;=\; J_{c,n-1}(\hat{\boldsymbol{U}}) + \lambda_{c,n}(\hat{\boldsymbol{U}}), \tag{2.28}$$

where

$$\begin{aligned}\lambda_{c,n}(\hat{\boldsymbol{U}}) &= \int_{nT}^{(n+1)T} r(t,\boldsymbol{U})\, s(t,\hat{\boldsymbol{U}})\, dt \\ &= \int_{nT}^{(n+1)T} r(t,\boldsymbol{U})\, \cos\left[2\pi f_1 t + \bar{\psi}(t,\hat{\boldsymbol{U}})\right]\, dt \\ \Leftrightarrow \qquad \lambda_{c,n}(\hat{U}_n,\hat{V}_n) &= \int_{nT}^{(n+1)T} r(t,\boldsymbol{U})\, \cos\left[2\pi f_1 t + 2\pi h\hat{U}_n\frac{t}{T} + 2\pi h\hat{V}_n\right]\, dt \end{aligned} \tag{2.29}$$

Thus $\lambda_{c,n}(\hat{U}_n,\hat{V}_n)$ can be generated using a bank of $M \cdot P$ filters and sampling the output of the filters at $t = (n+1)T$. The receiver correlates the received signal over one symbol period with all possible transmitted signals over that symbol period. Note that we have left out the amplitude scaling of $\sqrt{2E/T}$ in (2.29) as it does not affect performance.

## 2.7 Coherent Receiver for CPFSK

We now present the receiver structure of [AAS86] that will perform maximum likelihood sequence estimation of CPFSK. The receiver consists of two main parts: a coherent demodulator and a Viterbi processor.

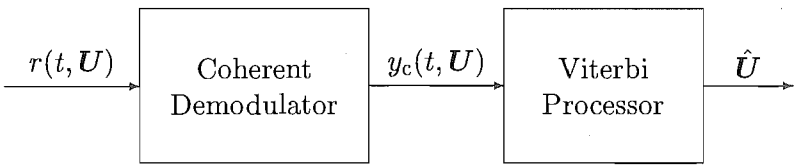



**Figure 2.7:** A MLSE receiver structure for coherent CPFSK

### 2.7.1 Coherent Demodulator

The first part of the receiver is the coherent demodulator shown in Figure 2.8, which moves the received signal in quadrature to baseband. The low-pass filters (LPFs) are ideal, passing only frequencies less than $f_L/2$. As $f_L << f_c$, the unwanted terms at twice the carrier frequency $f_c$ are removed. The canonical form of the noise in (2.24) is

$$w(t) = w_I(t)\cos(2\pi f_c t) - w_Q(t)\sin(2\pi f_c t), \tag{2.30}$$

where $w_I(t)$ and $w_Q(t)$ are both uncorrelated, zero-mean, Gaussian processes with psd's given by

$$S_{w_I}(f) = S_{w_Q}(f) = \begin{cases} N_0, & -\frac{f_w}{2} \leqslant f \leqslant \frac{f_w}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{2.31}$$

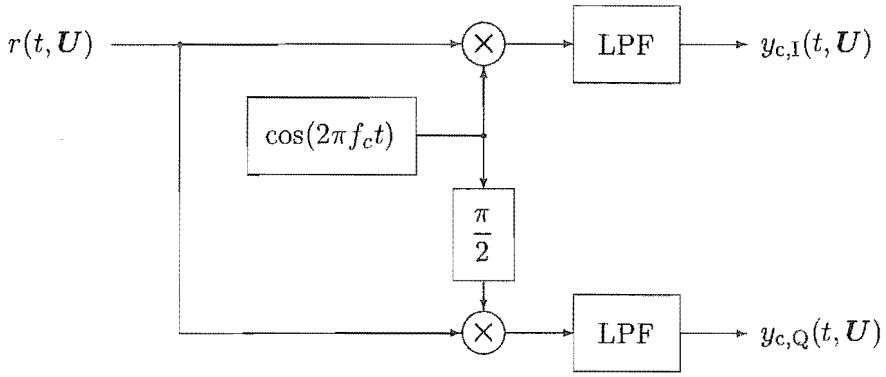



**Figure 2.8:** Coherent demodulator

Thus we assume that the noise is band-limited and that the bandwidth of the noise is some value $f_w$ ($f_w << f_c$). Making use of (2.19)-(2.21) and (2.30), we can write the canonical form of $r(t,\boldsymbol{U})$ as

$$r(t,\boldsymbol{U}) = r_{\rm I}(t,\boldsymbol{U})\cos(2\pi f_c t) - r_{\rm Q}(t,\boldsymbol{U})\sin(2\pi f_c t), \tag{2.32}$$

where

$$r_{\rm I}(t,\boldsymbol{U}) = s_{\rm I}(t,\boldsymbol{U}) + w_{\rm I}(t) \tag{2.33}$$

$$r_{\rm Q}(t,\boldsymbol{U}) = s_{\rm Q}(t,\boldsymbol{U}) + w_{\rm Q}(t). \tag{2.34}$$

We define the *complex envelopes* of $s(t,\boldsymbol{U})$, $w(t)$ and $r(t,\boldsymbol{U})$ to be

$$\tilde{s}(t,\boldsymbol{U}) \triangleq s_{\rm I}(t,\boldsymbol{U}) + j\, s_{\rm Q}(t,\boldsymbol{U}) \tag{2.35}$$

$$\tilde{w}(t) \triangleq w_{\rm I}(t,\boldsymbol{U}) + j\, w_{\rm Q}(t,\boldsymbol{U}) \tag{2.36}$$

$$\begin{aligned}\tilde{r}(t,\boldsymbol{U}) &\triangleq r_{\rm I}(t,\boldsymbol{U}) + j\, r_{\rm Q}(t,\boldsymbol{U}) \\ &= \tilde{s}(t,\boldsymbol{U}) + \tilde{w}(t).\end{aligned} \tag{2.37}$$

We show in Appendix B that the psd of $\tilde{w}(t)$ is given by

$$S_{\tilde{w}}(f) = \begin{cases} 2N_0, & -\frac{f_w}{2} \leqslant f \leqslant \frac{f_w}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{2.38}$$

Using (2.37), we can write the received signal of (2.32) as

$$\begin{aligned} r(t,\boldsymbol{U}) &= \mathrm{Re}\left[\tilde{r}(t,\boldsymbol{U})\exp(j2\pi f_c t)\right] \\ &= \frac{1}{2}\Big[\tilde{r}(t,\boldsymbol{U})\exp(j2\pi f_c t) + \tilde{r}^*(t,\boldsymbol{U})\exp(-j2\pi f_c t)\Big], \end{aligned} \tag{2.39}$$

where $\mathrm{Re}[u]$ denotes the real part of $u$, and $u^*$ denotes the complex conjugate of $u$. Similarly, the local oscillator output can be written as

$$\begin{aligned}\cos(2\pi f_c t) &= \mathrm{Re}\left[\exp(j2\pi f_c t)\right] \\ &= \frac{1}{2}\Big[\exp(j2\pi f_c t) + \exp(-j2\pi f_c t)\Big]. \end{aligned} \tag{2.40}$$

We look first at the in-phase component of the demodulated signal. Using (2.39) and (2.40), we can write $y_{\mathrm{c,I}}(t, \boldsymbol{U})$ in Figure 2.8 at the input to the low-pass filter (LPF) as

$$\begin{aligned} y_{\mathrm{c,I}}(t, \boldsymbol{U}) &= r(t, \boldsymbol{U}) \cos(2\pi f_c t) \\ &= \frac{1}{2}\Big[\tilde{r}(t, \boldsymbol{U}) \exp(j2\pi f_c t) + \tilde{r}^*(t, \boldsymbol{U}) \exp(-j2\pi f_c t)\Big] \\ &\quad \times \frac{1}{2}\Big[\exp(j2\pi f_c t) + \exp(-j2\pi f_c t)\Big] \\ &= \frac{1}{4}\Big[\tilde{r}(t, \boldsymbol{U}) \exp(j4\pi f_c t) + \tilde{r}^*(t, \boldsymbol{U}) + \tilde{r}(t, \boldsymbol{U}) + \tilde{r}^*(t, \boldsymbol{U}) \exp(-j4\pi f_c t)\Big]. \end{aligned} \tag{2.41}$$

The LPF removes the double frequency terms and further band-limits the noise. For convenience, the noise component at the output of the LPF is still denoted by $\tilde{w}(t)$, with psd given by (2.38), but now

$$f_w = f_{\mathrm{L}}. \tag{2.42}$$

Thus we can write the output of the LPF as

$$\begin{aligned} y_{\mathrm{c,I}}(t, \boldsymbol{U}) &= \frac{1}{4}\Big[\tilde{r}^*(t, \boldsymbol{U}) + \tilde{r}(t, \boldsymbol{U})\Big] \\ &= \frac{1}{2}\,\mathrm{Re}\left[\tilde{r}(t, \boldsymbol{U})\right], \end{aligned} \tag{2.43}$$

which is the in-phase component of the demodulated received signal.

We now turn to the quadrature component of the demodulated signal. The phase-shifted version of the local oscillator is given by

$$\begin{aligned} \cos(2\pi f_c t - \pi/2) &= \sin(2\pi f_c t) \\ &= \mathrm{Im}\left[\exp(j2\pi f_c t)\right] \\ &= \frac{1}{2j}\Big[\exp(j2\pi f_c t) - \exp(-j2\pi f_c t)\Big], \end{aligned} \tag{2.44}$$

where $\mathrm{Im}[u]$ denotes the imaginary part of $u$. We can thus write $y_{\mathrm{c,Q}}(t, \boldsymbol{U})$ in Figure 2.8

at the input to the LPF as

$$\begin{aligned} y_{c,Q}(t,\boldsymbol{U}) &= r(t,\boldsymbol{U})\sin(2\pi f_c t) \\ &= \frac{1}{2}\Big[\tilde{r}(t,\boldsymbol{U})\exp(j2\pi f_c t) + \tilde{r}^*(t,\boldsymbol{U})\exp(-j2\pi f_c t)\Big] \\ &\quad \times \frac{1}{2j}\Big[\exp(j2\pi f_c t) - \exp(-j2\pi f_c t)\Big] \\ &= \frac{1}{4j}\Big[\tilde{r}(t,\boldsymbol{U})\exp(j4\pi f_c t) + \tilde{r}^*(t,\boldsymbol{U}) - \tilde{r}(t,\boldsymbol{U}) - \tilde{r}^*(t,\boldsymbol{U})\exp(-j4\pi f_c t)\Big]. \end{aligned} \tag{2.45}$$

Again the LPF removes the double frequency terms, so that we can write the quadrature component of the demodulated complex-baseband received signal as

$$\begin{aligned} y_{c,Q}(t,\boldsymbol{U}) &= \frac{1}{4j}\Big[\tilde{r}^*(t,\boldsymbol{U}) - \tilde{r}(t,\boldsymbol{U})\Big] \\ &= -\frac{1}{2}\,\mathrm{Im}\left[\tilde{r}(t,\boldsymbol{U})\right]. \end{aligned} \tag{2.46}$$

We define the demodulated received signal $y_c(t,\boldsymbol{U})$ as

$$y_c(t,\boldsymbol{U}) \triangleq \sqrt{2}\left[y_{c,I}(t,\boldsymbol{U}) - j\, y_{c,Q}(t,\boldsymbol{U})\right], \tag{2.47}$$

where the factor $\sqrt{2}$ has been introduced to ensure that a noiseless $y_c(t,\boldsymbol{U})$ will integrate to $E$, similar to the transmitted signal $s(t,\boldsymbol{U})$. Substituting (2.43) and (2.46) into (2.47), we obtain

$$\begin{aligned} y_c(t,\boldsymbol{U}) &= \frac{1}{\sqrt{2}}\Big[\mathrm{Re}\left[\tilde{r}(t,\boldsymbol{U})\right] + j\,\mathrm{Im}\left[\tilde{r}(t,\boldsymbol{U})\right]\Big] \\ &= \frac{1}{\sqrt{2}}\,\tilde{r}(t,\boldsymbol{U}) \\ &= \frac{1}{\sqrt{2}}[\tilde{s}(t,\boldsymbol{U}) + \tilde{w}(t)]. \end{aligned} \tag{2.48}$$

Note that from (2.20) and (2.21), $\tilde{s}(t,\boldsymbol{U})$ is given by

$$\begin{aligned} \tilde{s}(t,\boldsymbol{U}) &= \sqrt{\frac{2E}{T}}\cos[\psi(t,\boldsymbol{U}) - 2\pi f_0 t] + j\sqrt{\frac{2E}{T}}\sin[\psi(t,\boldsymbol{U}) - 2\pi f_0 t] \\ &= \sqrt{\frac{2E}{T}}\exp(j[\psi(t,\boldsymbol{U}) - 2\pi f_0 t]), \end{aligned} \tag{2.49}$$

so that the demodulated signal in a coherent CPFSK system is given by

$$y_c(t,\boldsymbol{U}) = \sqrt{\frac{E}{T}}\exp(j[\psi(t,\boldsymbol{U}) - 2\pi f_0 t]) + \frac{1}{\sqrt{2}}\,\tilde{w}(t). \tag{2.50}$$

### 2.7.2 Viterbi Processor

The Viterbi processor takes $y_c(t, \boldsymbol{U})$ and forms the metric

$$\begin{aligned}\lambda_{c,n}(\hat{U}_n, \hat{V}_n) &= \cos(2\pi h\hat{V}_n) \int_{nT}^{(n+1)T} y_{c,I}(t, \boldsymbol{U}) \cos\left[2\pi h\hat{U}_n \frac{t}{T}\right] dt \\ &- \cos(2\pi h\hat{V}_n) \int_{nT}^{(n+1)T} y_{c,Q}(t, \boldsymbol{U}) \sin\left[2\pi h\hat{U}_n \frac{t}{T}\right] dt \\ &- \sin(2\pi h\hat{V}_n) \int_{nT}^{(n+1)T} y_{c,Q}(t, \boldsymbol{U}) \cos\left[2\pi h\hat{U}_n \frac{t}{T}\right] dt \\ &- \sin(2\pi h\hat{V}_n) \int_{nT}^{(n+1)T} y_{c,I}(t, \boldsymbol{U}) \sin\left[2\pi h\hat{U}_n \frac{t}{T}\right] dt, \end{aligned} \tag{2.51}$$

which is equivalent to the metric given in (2.29) [AAS86]. This can be implemented by $2M$ baseband filters with the impulse responses

$$h_{c,I,\hat{U}_n}(t) = \begin{cases} 2\cos\left(2\pi h\hat{U}_n \dfrac{T-t}{T}\right), & 0 \leqslant t < T \\ 0, & \text{otherwise,} \end{cases} \tag{2.52}$$

$$h_{c,Q,\hat{U}_n}(t) = \begin{cases} 2\sin\left(2\pi h\hat{U}_n \dfrac{T-t}{T}\right), & 0 \leqslant t < T \\ 0, & \text{otherwise.} \end{cases} \tag{2.53}$$

Figure 2.9 shows how this might be implemented. The Viterbi processor uses the Viterbi algorithm (discussed in Appendix C) to perform MLSE and produce an estimate $\hat{\boldsymbol{U}}$ of the transmitted data. Note that the multiplication and addition structure is repeated $P$ times, to hypothesise each possible value of $V_n$.

## 2.8 Euclidean Distance Properties of CPFSK

As discussed in Section 2.6, the Euclidean distance properties of a modulation are very important, as they determine its performance. Let $\boldsymbol{U}$ and $\hat{\boldsymbol{U}}$ be two information sequences, the normalised squared Euclidean distance (NSED) between two signals $s(t, \boldsymbol{U})$ and $s(t, \hat{\boldsymbol{U}})$ is defined as

$$d^2(\boldsymbol{U}, \hat{\boldsymbol{U}}) \triangleq \frac{\log_2 M}{2E} \int_{-\infty}^{\infty} \left| s(t, \boldsymbol{U}) - s(t, \hat{\boldsymbol{U}}) \right|^2 dt. \tag{2.54}$$

This can be re-written as

$$\begin{aligned} d^2(\boldsymbol{U}, \hat{\boldsymbol{U}}) &\triangleq \frac{\log_2 M}{2E} \sum_n \int_{nT}^{(n+1)T} \left| s(t, \boldsymbol{U}) - s(t, \hat{\boldsymbol{U}}) \right|^2 dt \\ &= \sum_n d_n^2(\boldsymbol{U}, \hat{\boldsymbol{U}}), \end{aligned} \tag{2.55}$$

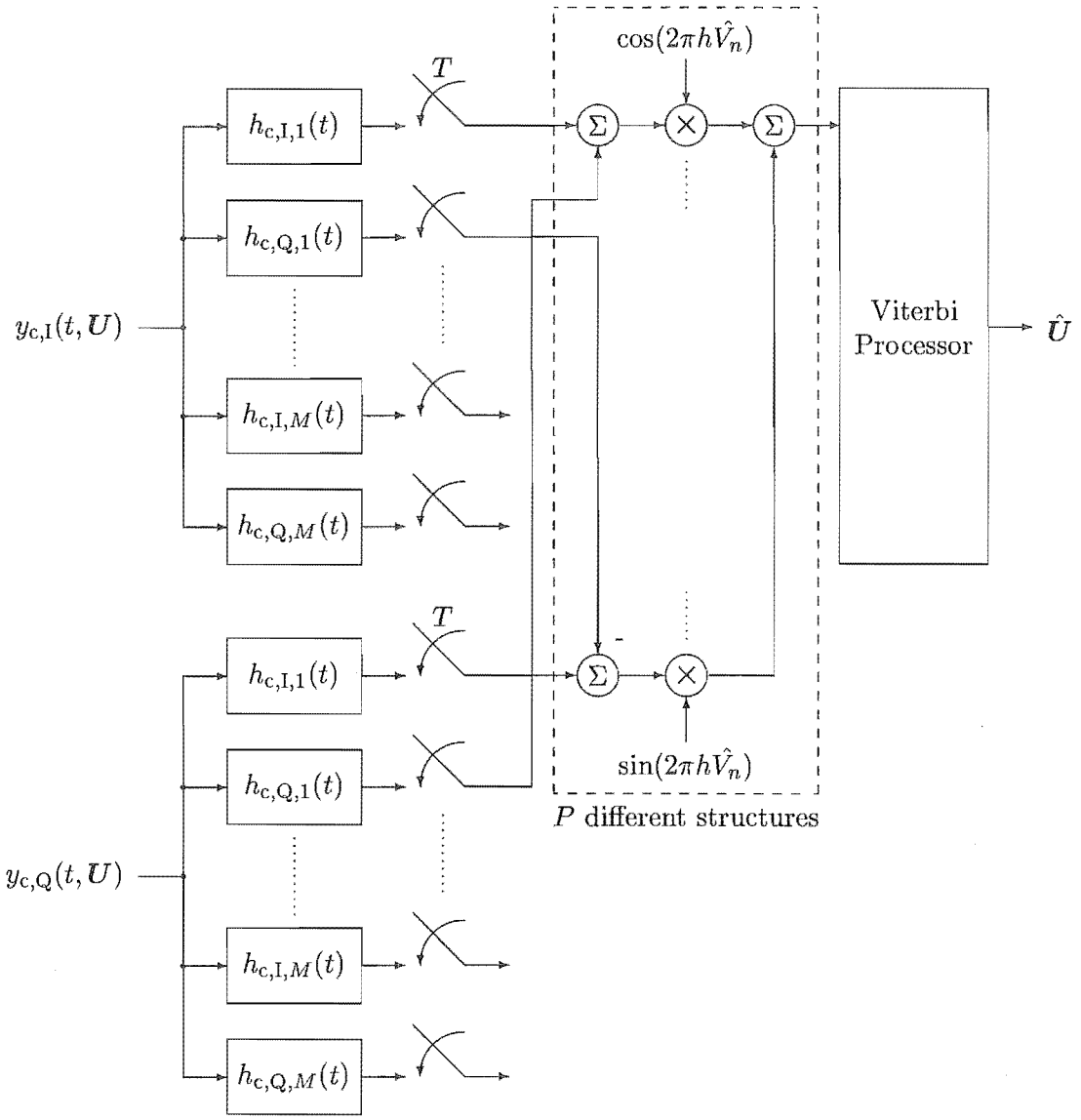



**Figure 2.9:** Detection Processor

where $d_n^2(\boldsymbol{U},\hat{\boldsymbol{U}})$ is called the normalised incremental squared Euclidean distance (NISED), and is defined as

$$d_n^2(\boldsymbol{U},\hat{\boldsymbol{U}}) \triangleq \frac{\log_2 M}{2E}\int_{nT}^{(n+1)T}\left|s(t,\boldsymbol{U})-s(t,\hat{\boldsymbol{U}})\right|^2 dt. \tag{2.56}$$

The most important Euclidean distance property is the normalised minimum squared Euclidean distance (NMSED), defined as

$$d_{\min}^2 \triangleq \min_{\substack{\boldsymbol{U},\hat{\boldsymbol{U}}\\ \boldsymbol{U}\neq\hat{\boldsymbol{U}}}} d^2(\boldsymbol{U},\hat{\boldsymbol{U}}). \tag{2.57}$$

We now present the NISED and NMSED of coherently-demodulated CPFSK.

### 2.8.1 Normalised Incremental Squared Euclidean Distance of CPFSK

The NISED for uncoded coherently-demodulated $M$-ary CPFSK modulation with $h = K/P$, where $K$ and $P$ are relatively prime positive integers is given by [Rim91]

$$d_n^2(\boldsymbol{U}, \hat{\boldsymbol{U}}) = \begin{cases} \log_2 M \left[1 - \frac{\sin\left(2\pi h\left[\sum_{i=0}^{n} U_i - \sum_{i=0}^{n} \hat{U}_i\right]\right) - \sin\left(2\pi h\left[\sum_{i=0}^{n-1} U_i - \sum_{i=0}^{n-1} \hat{U}_i\right]\right)}{2\pi h(U_n - \hat{U}_n)}\right], & U_n \neq \hat{U}_n \\ \log_2 M \left[1 - \cos\left(2\pi h \left[\sum_{i=0}^{n} U_i - \sum_{i=0}^{n} \hat{U}_i\right]\right)\right], & U_n = \hat{U}_n. \end{cases} \tag{2.58}$$

This can be written in terms of $U_n$ and $V_n$ as

$$d_n^2\left(\begin{bmatrix} U_n \\ V_n \end{bmatrix}, \begin{bmatrix} \hat{U}_n \\ \hat{V}_n \end{bmatrix}\right) = \begin{cases} \log_2 M \left[1 - \frac{\sin(2\pi h[U_n + V_n - \hat{U}_n - \hat{V}_n]) - \sin(2\pi h[V_n - \hat{V}_n])}{2\pi h(U_n - \hat{U}_n)}\right], & U_n \neq \hat{U}_n \\ \log_2 M \left[1 - \cos(2\pi h[V_n - \hat{V}_n])\right], & U_n = \hat{U}_n, \end{cases} \tag{2.59}$$

and in terms of the inputs to the memoryless modulator as

$$d_n^2(\boldsymbol{X}_n, \hat{\boldsymbol{X}}_n) = \begin{cases} \log_2 M \left[1 - \frac{\sin\left(2\pi h\left[X_n^{(1)} + X_n^{(2)} - \hat{X}_n^{(1)} - \hat{X}_n^{(2)}\right]\right) - \sin\left(2\pi h\left[X_n^{(2)} - \hat{X}_n^{(2)}\right]\right)}{2\pi h\left(X_n^{(1)} - \hat{X}_n^{(1)}\right)}\right], & X_n^{(1)} \neq \hat{X}_n^{(1)} \\ \log_2 M \left[1 - \cos\left(2\pi h \left[X_n^{(2)} - \hat{X}_n^{(2)}\right]\right)\right], & X_n^{(1)} = \hat{X}_n^{(1)}. \end{cases} \tag{2.60}$$

### 2.8.2 Normalised Minimum Squared Euclidean Distance for CPFSK

The NMSED for coherently-demodulated $M$-ary CPFSK modulation with $h = K/P \leq 1/2$ where $K$ and $P$ are relatively prime positive integers is [Rim91]

$$d_{\min}^2 = \begin{cases} \min\left[\log_2 M,\ 2\log_2 M \left(1 - \frac{\sin 2\pi h}{2\pi h}\right)\right] & M > P \\ 2\log_2 M \left(1 - \frac{\sin 2\pi h}{2\pi h}\right) & M \leq P \end{cases} \tag{2.61}$$

## 2.9 Error Performance of CPFSK

For coherently-demodulated CPFSK in the additive white Gaussian noise (AWGN) channel, the probability of bit-error $P_e$ at high SNR is closely approximated by [AAS86]

$$P_e = Q\left(\sqrt{\frac{d_{\min}^2 E_b}{N_0}}\right), \tag{2.62}$$

where $E_b$ is the bit energy and $N_0$ is the one-sided power spectral density of the noise, $d^2_{\min}$ is the NMSED of the scheme in question. The $Q$-function is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{v^2}{2}\right) dv. \tag{2.63}$$

In Table 2.1 we present $d^2_{\min}$ for various CPFSK schemes, calculated using (2.61). There is an obvious loss in $d^2_{\min}$ as $M$ increases, and from (2.62) this results in poorer performance. However this performance degradation is tempered by an increase in throughput measured in bits/symbol. The degradation is very similar to that experienced by phase shift keying (PSK).

**Table 2.1**

$d^2_{\min}$ for various CPFSK schemes

| Scheme | $d^2_{\min}$ | |
|---|---|---|
| MSK | 2 | 2 |
| 4-CPFSK | $4 - \frac{8}{\pi}$ | 1.45 |
| 8-CPFSK | $6 - \frac{12\sqrt{2}}{\pi}$ | 0.60 |

## 2.10 Simulation of a CPFSK system

To reduce the long run-times of the simulation of a band-pass system, simulations were run in complex baseband. Appendix D discusses the receiver and simulation model used to obtain the results. The data was assumed to be uniformally distributed, 16 samples per symbol were generated, and the Viterbi algorithm had a decision depth of at least 10 symbols. Figure 2.10 shows the results of the simulations compared with the theoretical performance of various coherently-demodulated CPFSK schemes. The theoretical curves were calculated using (2.62) and the results in Table 2.1. The simulation results agree well with the computed curves.

## 2.11 Summary

In this chapter we have introduced CPFSK and its properties. We looked at the structure of the phase of a CPFSK signal and presented the idea of a phase tree and then a phase trellis. This led to the two-part decomposed model of CPFSK, which provides greater insight into how its inherent coding and modulation aspects interact. A coherent CPFSK receiver

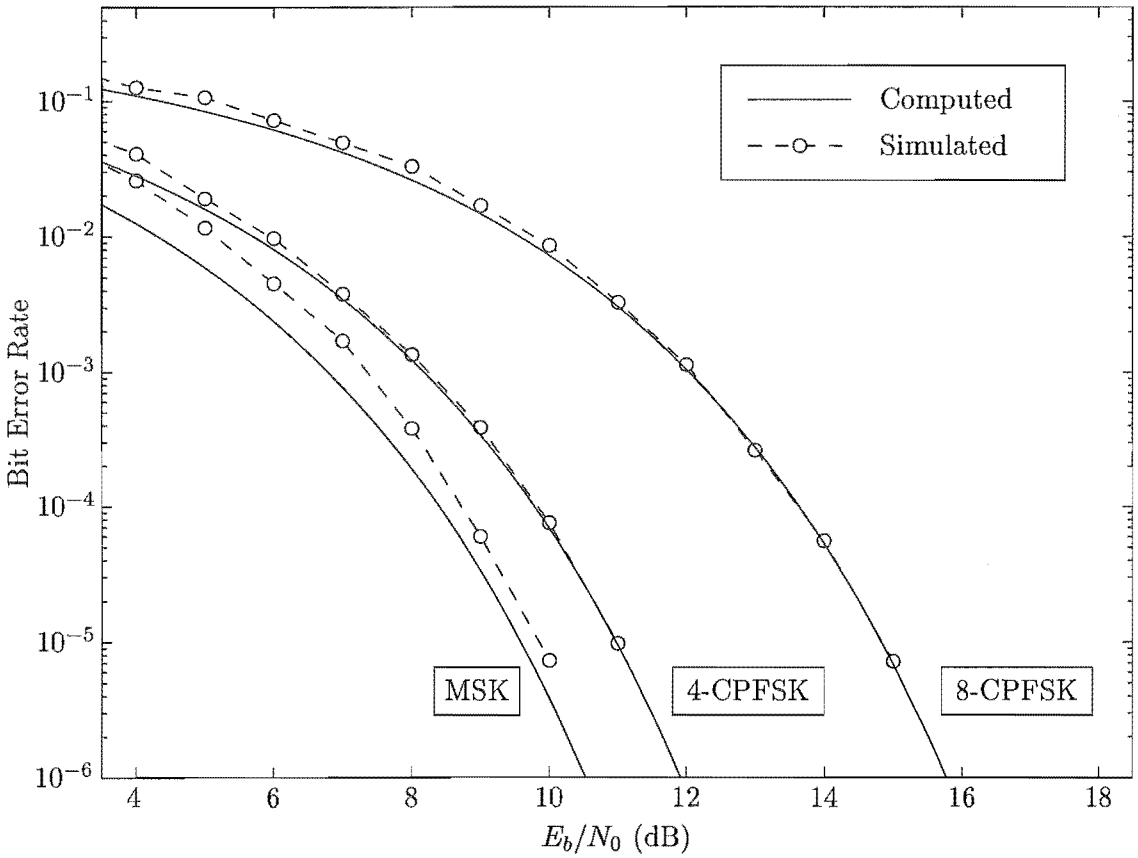



**Figure 2.10:** Computed and simulated performance of various $M$-CPFSK schemes

structure was presented and discussed. The distance properties of CPFSK were presented, followed by the theoretical error performance of coherently-demodulated CPFSK. We then discussed how simulations had been performed and showed how their results compared very well with the theoretical curves previously presented.

# Chapter 3

# Differential Continuous Phase Frequency Shift Keying

## 3.1 Introduction

Although coherent demodulation performs very well, phase and frequency differences between the oscillators in the transmitter and receiver must be compensated for, and the required algorithms can be very computationally intensive. A possible solution is differential demodulation. The idea is to use one symbol as a phase reference to demodulate and detect the next, thereby avoiding the need for a local oscillator in the receiver. This simplifies receiver structure at the expense of performance, as the carrier reference is corrupted by the channel. However we find that without specific differential encoding, we can detect the original data values, but lose the knowledge of the phase trellis, and thus suffer a further performance loss.

The work of [YT92], developed a differential encoder for CPFSK that preserves the phase trellis through differential demodulation. However that encoder is designed for binary multi-h CPFSK, and does not extend to $M$-ary signals. It accepts standard binary bi-polar CPM symbols ($\alpha_n = \pm 1$) and outputs symbols in the set $\{-P, \ldots, -1, 0, 1, \ldots, P-1\}$, and will not interface well with the decomposition of CPFSK, or any external error-control encoders.

As a consequence, we now develop an $M$-ary differential encoder for CPFSK that is also a linear encoder over the ring of integers modulo-$P$ ($\mathbb{Z}_P$). This should interface well with the decomposition of CPFSK and provide greater understanding of differentially-encoded and differentially-demodulated CPFSK (DCPFSK). A linear differential encoder will allow us to cascade the channel encoder, the differential encoder and the CPE to form

an overall linear encoder over $\mathbb{Z}_P$ and thus simplify code searching.

In Section 3.2 we discuss differential demodulation and the form of a differentially-demodulated CPFSK signal. This leads on to a method to differentially encode CPFSK in Section 3.3, and then to the decomposition of DCPFSK in Section 3.4. We then discuss the phase trees and trellises of DCPFSK in Sections 3.5 and 3.6 respectively. The detection of DCPFSK is discussed in Section 3.7, and a receiver structure for DCPFSK based on Euclidean distance is presented in Section 3.8. In Section 3.9 we analyse DCPFSK to determine its Euclidean distance properties. The error performance of DCPFSK is then discussed in Section 3.10. We compare our simulation results to computed curves in Section 3.11. Finally, we present a summary of the chapter in Section 3.12

## 3.2 Differential Demodulation

A general structure for differential demodulation is shown in Figure 3.1. The received signal is multiplied in quadrature by a copy of the received signal that has been phase-shifted and delayed by a symbol period. The resulting signals are low-pass filtered to obtain the desired in-phase and quadrature base-band signals. As a delayed version of the received signal is used as the carrier reference, extra noise is introduced into the system and performance suffers. Note that as there is no local oscillator, the receiver is much simpler to build as the only synchronisation required is that of symbol timing. Thus differential demodulation trades off receiver complexity against performance.

As in the coherent case, the received signal $r(t, \boldsymbol{U})$ is given by

$$r(t, \boldsymbol{U}) = s(t, \boldsymbol{U}) + w(t). \tag{3.1}$$

Using (2.32)-(2.38), we can write the received signal as

$$\begin{aligned} r(t, \boldsymbol{U}) &= \mathrm{Re}\left[\tilde{r}(t, \boldsymbol{U}) \exp(j2\pi f_c t)\right] \\ &= \frac{1}{2}[\tilde{r}(t, \boldsymbol{U}) \exp(j2\pi f_c t) + \tilde{r}^*(t, \boldsymbol{U}) \exp(-j2\pi f_c t)]. \end{aligned} \tag{3.2}$$

This allows us to write $y_{\mathrm{d,I}}(t, \boldsymbol{U})$ in Figure 3.1 at the input to the low-pass filter (LPF) as

$$\begin{aligned} y_{\mathrm{d,I}}(t, \boldsymbol{U}) &= r(t, \boldsymbol{U}) r(t - T, \boldsymbol{U}) \\ &= \frac{1}{2}\Big[\tilde{r}(t, \boldsymbol{U}) \exp(j2\pi f_c t) + \tilde{r}^*(t, \boldsymbol{U}) \exp(-j2\pi f_c t)\Big] \\ &\quad \times \frac{1}{2}\Big[\tilde{r}(t - T, \boldsymbol{U}) \exp(j2\pi f_c [t - T]) + \tilde{r}^*(t - T, \boldsymbol{U}) \exp(-j2\pi f_c [t - T])\Big] \end{aligned}$$

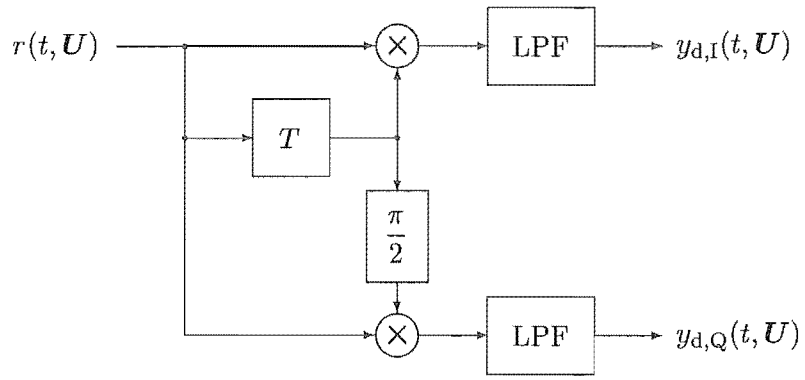



**Figure 3.1:** Differential demodulator

$$
\begin{aligned}
= \frac{1}{4}\Big[&\tilde{r}(t,\boldsymbol{U})\tilde{r}(t-T,\boldsymbol{U})\exp(j2\pi f_c[2t-T]) \\
&+\tilde{r}^*(t,\boldsymbol{U})\tilde{r}(t-T,\boldsymbol{U})\exp(-j2\pi f_c T) \\
&+\tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\exp(j2\pi f_c T) \\
&+\tilde{r}^*(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\exp(-j2\pi f_c[2t-T])\Big]. \qquad (3.3)
\end{aligned}
$$

The low-pass filter removes the double frequency terms, and assuming $f_cT$ is equal to some integer, we may write the output of the LPF on the in-phase chain as

$$
\begin{aligned}
y_{\mathrm{d,I}}(t,\boldsymbol{U}) &= \frac{1}{4}\Big[\tilde{r}^*(t,\boldsymbol{U})\tilde{r}(t-T,\boldsymbol{U})+\tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\Big] \\
&= \frac{1}{2}\,\mathrm{Re}\left[\tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\right]. \qquad (3.4)
\end{aligned}
$$

Note that if $f_cT$ is not equal to some integer, then $\exp(j2\pi f_cT)$ and its conjugate will cause some phase rotation. However, as both $f_c$ and $T$ are known, the rotation can easily be accounted for.

Before proceeding with $y_{\mathrm{d,Q}}(t,\boldsymbol{U})$, we need to consider the effect of the $\pi/2$-phase shifter. Let us denote the $T$-delayed and $\pi/2$-phase shifted version of $r(t,\boldsymbol{U})$ as $\dot{r}(t-T,\boldsymbol{U})$, then from (3.2)

$$
\begin{aligned}
\dot{r}(t-T,\boldsymbol{U}) &= \mathrm{Re}\left[\tilde{r}(t-T,\boldsymbol{U})\exp(j2\pi f_c[t-T]-\pi/2)\right] \\
&= \mathrm{Re}\left[\tilde{r}(t-T,\boldsymbol{U})\exp(j2\pi f_c[t-T])\exp(-j\pi/2)\right] \\
&= \frac{1}{2}\Big[\tilde{r}(t-T,\boldsymbol{U})\exp(j2\pi f_c[t-T])\,\exp(-j\pi/2) \\
&\qquad +\tilde{r}^*(t-T,\boldsymbol{U})\exp(-j2\pi f_c[t-T])\,\exp(j\pi/2)\Big]
\end{aligned}
$$

$$= \frac{1}{2j}\Big[\tilde{r}(t-T,\boldsymbol{U})\exp(j2\pi f_c[t-T]) - \tilde{r}^*(t-T,\boldsymbol{U})\exp(-j2\pi f_c[t-T])\Big]. \quad (3.5)$$

We can now write $y_{\mathrm{d,Q}}(t,\boldsymbol{U})$ in Figure 3.1 at the input to the low-pass filter (LPF) as

$$\begin{aligned} y_{\mathrm{d,Q}}(t,\boldsymbol{U}) &= r(t,\boldsymbol{U})\dot{r}(t-T,\boldsymbol{U}) \\ &= \frac{1}{2}\Big[\tilde{r}(t,\boldsymbol{U})\exp(j2\pi f_c t) + \tilde{r}^*(t,\boldsymbol{U})\exp(-j2\pi f_c t)\Big] \\ &\quad \times \frac{1}{2j}\Big[\tilde{r}(t-T,\boldsymbol{U})\exp(j2\pi f_c[t-T]) \\ &\qquad - \tilde{r}^*(t-T,\boldsymbol{U})\exp(-j2\pi f_c[t-T])\Big] \\ &= \frac{1}{4j}\Big[\tilde{r}(t,\boldsymbol{U})\tilde{r}(t-T,\boldsymbol{U})\exp(j2\pi f_c[2t-T]) \\ &\qquad + \tilde{r}^*(t,\boldsymbol{U})\tilde{r}(t-T,\boldsymbol{U})\exp(-j2\pi f_c T) \\ &\qquad - \tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\exp(j2\pi f_c T) \\ &\qquad - \tilde{r}^*(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\exp(-j2\pi f_c[2t-T])\Big]. \end{aligned} \quad (3.6)$$

Following the LPF, and again using the assumption that $f_c T$ is equal to some integer, we have

$$\begin{aligned} y_{\mathrm{d,Q}}(t,\boldsymbol{U}) &= \frac{1}{4j}\Big[\tilde{r}^*(t,\boldsymbol{U})\tilde{r}(t-T,\boldsymbol{U}) - \tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\Big] \\ &= -\frac{1}{2}\operatorname{Im}\left[\tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\right]. \end{aligned} \quad (3.7)$$

We define the demodulated signal $y_{\mathrm{d}}(t,\boldsymbol{U})$ as

$$y_{\mathrm{d}}(t,\boldsymbol{U}) \triangleq \sqrt{\frac{T}{E}}\left[y_{\mathrm{d,I}}(t,\boldsymbol{U}) - j\, y_{\mathrm{d,Q}}(t,\boldsymbol{U})\right], \quad (3.8)$$

where the factor $\sqrt{T/E}$ has been introduced to facilitate energy calculations. Substituting (3.4) and (3.7) into (3.8), we obtain

$$\begin{aligned} y_{\mathrm{d}}(t,\boldsymbol{U}) &= \sqrt{\frac{T}{E}}\left[y_{\mathrm{d,I}}(t,\boldsymbol{U}) - j\, y_{\mathrm{d,Q}}(t,\boldsymbol{U})\right] \\ &= \frac{1}{2}\sqrt{\frac{T}{E}}\left\{\operatorname{Re}\left[\tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\right] + j\operatorname{Im}\left[\tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U})\right]\right\} \\ &= \frac{1}{2}\sqrt{\frac{T}{E}}\,\tilde{r}(t,\boldsymbol{U})\tilde{r}^*(t-T,\boldsymbol{U}). \end{aligned} \quad (3.9)$$

Substituting (2.37) into (3.9), we find that the output of the differential demodulator is

$$\begin{aligned} y_{\mathrm{d}}(t,\boldsymbol{U}) &= \frac{1}{2}\sqrt{\frac{T}{E}}\Big[\{\tilde{s}(t,\boldsymbol{U}) + \tilde{w}(t)\}\{\tilde{s}^*(t-T,\boldsymbol{U}) + \tilde{w}^*(t-T)\}\Big] \\ &= \frac{1}{2}\sqrt{\frac{T}{E}}\Big[\tilde{s}(t,\boldsymbol{U})\,\tilde{s}^*(t-T,\boldsymbol{U}) + \tilde{w}(t)\,\tilde{s}^*(t-T,\boldsymbol{U}) \\ &\qquad + \tilde{s}(t,\boldsymbol{U})\,\tilde{w}^*(t-T) + \tilde{w}(t)\,\tilde{w}^*(t-T)\Big] \end{aligned} \quad (3.10)$$

It is important to note that the power spectral densities of $\tilde{w}(t)$ and its components, given by (2.38) and (2.31), are limited to $f_w$ as in the coherent case.

The information we are interested in is contained in the signal × signal term. Using (2.49), this can be developed as

$$\begin{aligned} y_{\mathrm{d}}(t,\boldsymbol{U}) &= \frac{1}{2}\sqrt{\frac{T}{E}}\left[\tilde{s}(t,\boldsymbol{U})\,\tilde{s}^*(t-T,\boldsymbol{U})\right] \\ &= \frac{1}{2}\sqrt{\frac{T}{E}}\left[\sqrt{\frac{2E}{T}}\exp(j[\psi(t,\boldsymbol{U})-2\pi f_0 t]) \right. \\ &\qquad \left. \times\sqrt{\frac{2E}{T}}\exp(-j[\psi(t-T,\boldsymbol{U})-2\pi f_0(t-T)])\right] \\ &= \sqrt{\frac{E}{T}}\exp(j[\psi(t,\boldsymbol{U})-\psi(t-T,\boldsymbol{U})-2\pi f_0 T)]) \\ &= \sqrt{\frac{E}{T}}\exp(j[\psi_d(t,\boldsymbol{U})-2\pi f_0 T])\,, \end{aligned} \tag{3.11}$$

where

$$\begin{aligned} \psi_d(t,\boldsymbol{U}) &\triangleq \psi(t,\boldsymbol{U})-\psi(t-T,\boldsymbol{U}) \\ &= 4\pi h\left[\sum_{i=0}^{n} U_i q(t-iT)-\sum_{i=0}^{n-1} U_i q[t-(i+1)T]\right] \\ &= 2\pi h\left[\sum_{i=0}^{n-1} U_i + U_n\frac{t-nT}{T}-\sum_{i=0}^{n-2} U_i - U_{n-1}\frac{t-nT}{T}\right], \quad nT\le t<(n+1)T \\ &= 2\pi h\left[U_{n-1}+(U_n-U_{n-1})\frac{t-nT}{T}\right], \qquad nT\le t<(n+1)T. \end{aligned} \tag{3.12}$$

Note that from now on we will ignore the $2\pi f_0 T$ term in (3.11) as it is just a known phase rotation and can easily be accounted for. Thus the noise-free version of $y_{\mathrm{d}}(t,\boldsymbol{U})$ is given by

$$y_{\mathrm{d}}(t,\boldsymbol{U}) = \sqrt{\frac{E}{T}}\exp\left(j\,2\pi h\left[U_{n-1}+(U_n-U_{n-1})\frac{t-nT}{T}\right]\right), \qquad nT\le t<(n+1)T. \tag{3.13}$$

Looking at the phase of $y_{\mathrm{d}}(t,\boldsymbol{U})$, we see that we can detect $U_n$, but we have lost the phase trellis, as there is no term depending on all the previously transmitted symbols. Without the phase trellis, $y_{\mathrm{d}}(t,\boldsymbol{U})$ is memoryless and the performance suffers. This is because the use of the Viterbi algorithm provides no performance improvement in the case of a memoryless signal.

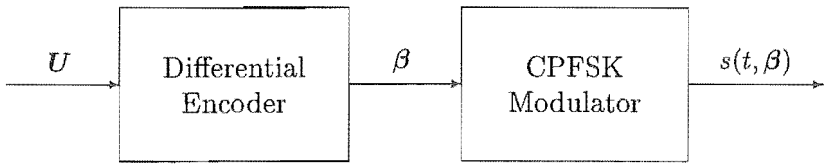



**Figure 3.2:** Differentially-encoded CPFSK system

## 3.3 Differentially Encoding CPFSK

Let us assume that there exists a differential encoder preceding the CPE as shown in Figure 3.2, that accepts the uncoded $M$-ary symbols $\boldsymbol{U}$ and outputs the differentially-encoded symbols $\boldsymbol{\beta}$.

The transmitted signal in such a system is given by

$$s(t,\boldsymbol{\beta}) = \sqrt{\frac{2E}{T}}\cos\left(2\pi f_1 t + 2\pi h\sum_{i=0}^{n-1}\beta_i + 2\pi h\beta_n\frac{t-nT}{T}\right), \qquad nT \le t < (n+1)T. \tag{3.14}$$

As shown in Section 3.2, this would result in a noise-free differentially-demodulated signal of

$$y_{\mathrm{d}}(t,\boldsymbol{\beta}) = \sqrt{\frac{E}{T}}\exp\left(j\,2\pi h\left[\beta_{n-1} + (\beta_n - \beta_{n-1})\frac{t-nT}{T}\right]\right), \qquad nT \le t < (n+1)T. \tag{3.15}$$

The noise-free output of a coherent system can be written as

$$y_{\mathrm{c}}(t,\boldsymbol{U}) = \sqrt{\frac{E}{T}}\exp\left(j\,2\pi h\left[\sum_{i=0}^{n-1}U_i + U_n\frac{t-nT}{T}\right]\right), \qquad nT \le t < (n+1)T. \tag{3.16}$$

We now require the phases of (3.15) and (3.16) to be equal. Taking the first term in the phases, we have

$$\beta_{n-1} = \sum_{i=0}^{n-1}U_i. \tag{3.17}$$

The second term gives us

$$\beta_n - \beta_{n-1} = U_n. \tag{3.18}$$

An obvious solution is to define $\beta_n$ as

$$\beta_n \triangleq \sum_{i=0}^{n}U_i, \tag{3.19}$$

as this would satisfy both (3.17) and (3.18). Although this is a mathematically elegant solution, it is unfortunately not a practical one. Looking at the transmitted signal, given by (3.14), we see that there are three terms in the phase of $s(t,\boldsymbol{\beta})$. The first term $2\pi f_1 t$ is a term dependent solely on the carrier frequency and thus unaffected by differential encoding. The second term $2\pi h \sum_{i=0}^{n-1} \beta_i$ is a data-dependent phase term that is constant over each symbol period, and thus is also not significantly affected by differential encoding. The third term $2\pi h \beta_n (t - nT)/T$ is a data-dependent frequency term. As $U_n$ is non-negative, the definition in (3.19) has $\beta_n$ increasing without bound, thus the third term in (3.14) would result in an ever-increasing frequency. This of course would mean that $s(t,\boldsymbol{\beta})$ had an ever-increasing band-width, a highly undesirable characteristic. Thus (3.19) is not a good choice as the definition of the differential encoder.

As seen in Section 2.4, we can reduce any term in the phase of a signal modulo-$2\pi$. Noting our restriction on $h$, given by (2.6), we can take the constant phase terms of (3.15) and (3.16) and write (3.17) as

$$\begin{aligned}
& R_{2\pi}[2\pi h \beta_n] = R_{2\pi}\left[2\pi h \sum_{i=0}^{n-1} U_i\right] \\
\Leftrightarrow \quad & R_{2\pi}\left[2\pi \left(\frac{K}{P}\right) \beta_n\right] = R_{2\pi}\left[2\pi \left(\frac{K}{P}\right) \sum_{i=0}^{n-1} U_i\right] \\
\Leftrightarrow \quad & R_{2\pi}\left[2\pi \left(\frac{K}{P}\right) R_P[\beta_n]\right] = R_{2\pi}\left[2\pi \left(\frac{K}{P}\right) R_P\left[\sum_{i=0}^{n-1} U_i\right]\right] \\
\Leftrightarrow \quad & R_P[\beta_n] = R_P\left[\sum_{i=0}^{n-1} U_i\right], \qquad (3.20)
\end{aligned}$$

where we have made use of the properties (A.2) and (A.3). Thus we could define $\beta_n$ as

$$\beta_n \triangleq R_P\left[\sum_{i=0}^{n-1} U_i\right]. \qquad (3.21)$$

In this case $\beta_n$ can take on one of $P$ values, namely $\{0, 1, \ldots, P-1\}$. Thus the data-dependent frequency term in $s(t,\boldsymbol{\beta})$ is limited to $P$ different frequencies.

Using the definition (3.21), $\beta_n$ is a $P$-ary symbol, so that

$$\beta_n - \beta_{n-1} \in \{-(P-1), -(P-2), \ldots, P-1\}. \qquad (3.22)$$

Thus (3.18) is not satisfied. However, once we have detected the value of $\beta_n - \beta_{n-1}$, we can reduce this modulo-$P$ to find $R_P[U_n]$ as (from (A.2))

$$R_P[\beta_n - \beta_{n-1}] = R_P\left[R_P\left[\sum_{i=0}^{n} U_i\right] - R_P\left[\sum_{i=0}^{n-1} U_i\right]\right]$$

$$= R_P\left[\sum_{i=0}^{n} U_i - \sum_{i=0}^{n-1} U_i\right]$$
$$= R_P[U_n]\,. \tag{3.23}$$

This leaves us with the question: is $R_P[U_n]$ equal to $U_n$? Obviously it is if $M = P$. If $P > M$ then we can further reduce (3.23) modulo-$M$ to obtain $U_n$, as (from (A.5))

$$\begin{aligned} R_M[R_P[\beta_n - \beta_{n-1}]] &= R_M[R_P[U_n]] \\ &= R_M[U_n] \\ &= U_n, \end{aligned} \tag{3.24}$$

if and only if $P$ is a multiple of $M$. So the differential encoder of (3.21) generates $P$ states for the demodulated trellis, and allows us to detect the transmitted symbol $U_n$, if $P$ is a multiple of $M$.

For the case where $M > P$, the differential encoder of (3.21) does not preserve enough information to transmit $M$ symbols. A solution is to define the differential encoder as

$$\beta_n \triangleq R_M\left[\sum_{i=0}^{n-1} U_i\right]. \tag{3.25}$$

Once the quantity $\beta_n - \beta_{n-1}$ has been detected, $U_n$ can be detected by modulo-$M$ reduction as

$$\begin{aligned} R_M[\beta_n - \beta_{n-1}] &= R_M\left[R_M\left[\sum_{i=0}^{n} U_i\right] - R_M\left[\sum_{i=0}^{n-1} U_i\right]\right] \\ &= R_M\left[\sum_{i=0}^{n} U_i - \sum_{i=0}^{n-1} U_i\right] \\ &= R_M[U_n] \\ &= U_n. \end{aligned} \tag{3.26}$$

We need to show that the definition of (3.25) also produces $P$ states for the phase trellis to fulfil the requirement of (3.17). As discussed above, we can reduce the constant phase term of (3.15) modulo-$2\pi$, and using the definition of (3.25) and noting (2.6), we have

$$\begin{aligned} R_{2\pi}[2\pi h\beta_n] &= R_{2\pi}\left[2\pi\left(\frac{K}{P}\right) R_M\left[\sum_{i=0}^{n-1} U_i\right]\right] \\ &= R_{2\pi}\left[2\pi\left(\frac{K}{P}\right) R_P\left[R_M\left[\sum_{i=0}^{n-1} U_i\right]\right]\right] \\ &= R_{2\pi}\left[2\pi\left(\frac{K}{P}\right) R_P\left[\sum_{i=0}^{n-1} U_i\right]\right] \end{aligned}$$

$$= R_{2\pi}\left[2\pi h \sum_{i=0}^{n-1} U_i\right] \tag{3.27}$$

where we have made use of the properties (A.2), (A.3) and (A.5), and incurred the restriction that $P$ must be a factor of $M$. Thus the differential encoder of (3.25) allows us to detect the transmitted symbol $U_n$ and generates $P$ states for the demodulated trellis, if $P$ is a factor of $M$.

We can now finally define the differential encoder in general terms as

$$\beta_n = R_B\left[\sum_{i=0}^{n} U_i\right], \tag{3.28}$$

where

$$B = \max\{P, M\}, \tag{3.29}$$

with the restriction that

$$\max\{P, M\} = k \min\{P, M\}, \tag{3.30}$$

where $k$ is an arbitrary positive integer. We can write (3.28) in a recursive form as

$$\begin{aligned}
\beta_n &= R_B\left[\sum_{i=0}^{n} U_i\right] \\
&= R_B\left[\sum_{i=0}^{n-1} U_i + U_n\right] \\
&= R_B\left[R_B\left[\sum_{i=0}^{n-1} U_i\right] + U_n\right] \\
&= R_B[\beta_{n-1} + U_n]\,.
\end{aligned} \tag{3.31}$$

This differential encoder is shown in Figure 3.3, where the addition is modulo-$B$. The differential encoder is a linear encoder over $\mathbb{Z}_B$. If $B = M$, then the differential encoder is a scrambler [For70], as it replaces one stream of $M$-ary symbols with another stream of $M$-ary symbols in a one-to-one permutation.

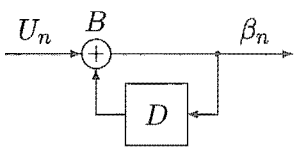



**Figure 3.3:** Differential encoder

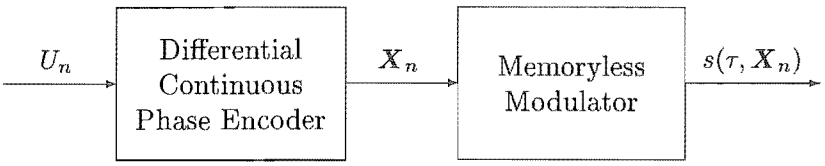



**Figure 3.4:** Decomposition of DCPFSK

## 3.4 The Decomposition of DCPFSK

A DCPFSK system can be decomposed into two parts, a differential continuous-phase encoder (DCPE) and a memoryless modulator (MM), as shown in Figure 3.4. The purpose of the DCPE is to provide an input to the memoryless modulator so that the memoryless modulator produces a CPFSK signal suitable for differential demodulation. The DCPE consists of a differential encoder cascaded with a CPE. It has the information symbol $U_n$ as input, and outputs two quantities: $\beta_n$ and $v_n$. $\beta_n$ is the differentially-encoded information symbol, defined in (3.28)-(3.31), and $v_n$ is the accumulated symbol phase of the transmitted DCPFSK signal, defined as

$$v_n \triangleq R_P\left[\sum_{i=0}^{n-1}\beta_i\right]. \tag{3.32}$$

Note that we can re-write (3.32) in a recursive form as

$$\begin{aligned} v_n &= R_P\left[\sum_{i=0}^{n-1}\beta_i\right] \\ &= R_P\left[\sum_{i=0}^{n-2}\beta_i + \beta_{n-1}\right] \\ &= R_P\left[R_P\left[\sum_{i=0}^{n-2}\beta_i\right] + \beta_{n-1}\right] \\ &= R_P[v_{n-1} + \beta_{n-1}]. \end{aligned} \tag{3.33}$$

The DCPE is shown in Figure 3.5, where all the addition is modulo, with the modulo base denoted by the quantity above each adder. It is important to note that if $M$ and $P$ are such that $B = P$, the DCPE will be a linear encoder over the ring of integers modulo-$P$.

The memoryless modulator in Figure 3.4 is exactly the same as that of Section 2.5.1. Its input is defined in (2.12) with elements

$$X_n^{(1)} = \beta_n \quad \text{and} \quad X_n^{(2)} = v_n, \tag{3.34}$$

for a DCPFSK system with no external error-control coding.

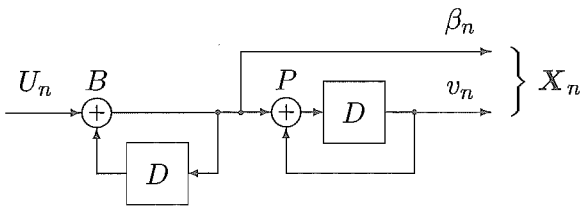


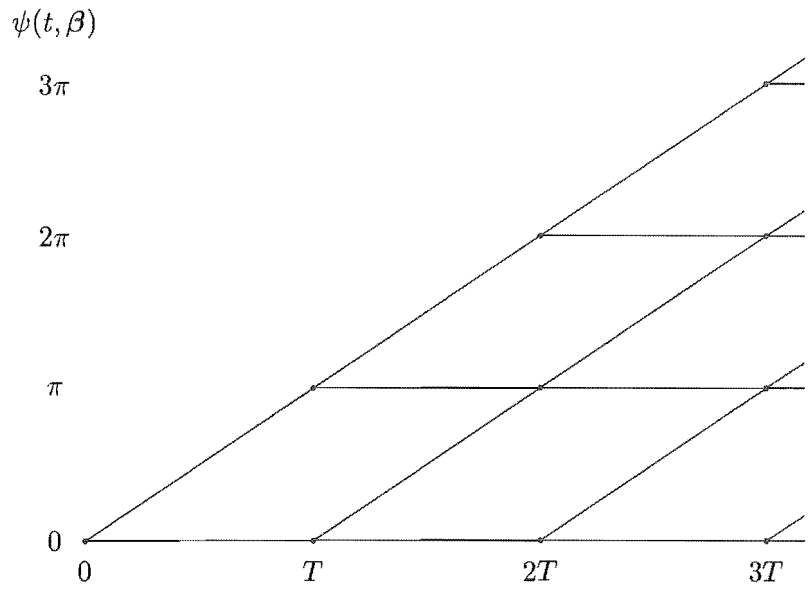

**Figure 3.5:** Differential continuous phase encoder

As with CPFSK, the two most important parameters in a DCPFSK scheme are $h$ and $M$ as they determine the scheme's information bit-rate, band-width and performance. Again, an important class of DCPFSK schemes result when $h = 1/M$, and it is convenient to refer to an $M$-ary DCPFSK scheme with $h = 1/M$ as $M$-DCPFSK. Thus 4-DCPFSK denotes quaternary DCPFSK with $h = 1/4$, and DMSK could be referred to as 2-DCPFSK.

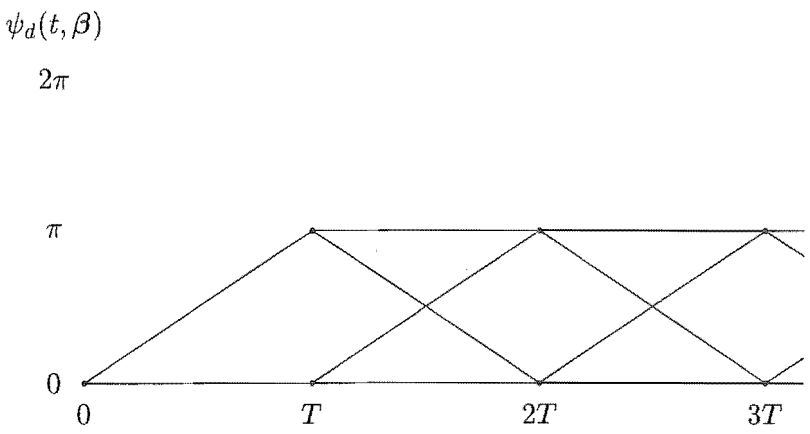
## 3.5 Phase Tree

As discussed in Sections 2.3 and 2.4, the phase tree (and therefore the trellis) of a CPFSK system is based on the information-dependent part(s) of the transmitted signal. With coherent demodulation, the phase tree is unchanged by the demodulation process. This is because the only part of the modulated and demodulated signals $s(t, \boldsymbol{U})$ and $y_c(t, \boldsymbol{U})$ that depends on the information symbols is $\psi(t, \boldsymbol{U})$. This is not the case with differential demodulation. In a differential system, the transmitted signal is $s(t, \boldsymbol{\beta})$, whose information-dependent part is $\psi(t, \boldsymbol{\beta})$. The demodulated signal $y_d(t, \boldsymbol{\beta})$'s information content is contained in $\psi_d(t, \boldsymbol{\beta})$, which defined in (3.12). Thus the phase trees of the modulated and demodulated signals are different in a DCPFSK system.

The phase tree is generated in the same manner as the coherent case, namely by hypothesising all the possible signal paths and plotting them on the same axes. In Figure 3.6(a) we show the phase tree of a modulated DMSK signal. The phase tree is identical to that of coherent MSK. This is not surprising as in a DMSK system $M$ and $P$ are both equal to 2, and therefore $B = 2$. Thus the differential encoder will just substitute one binary sequence for another, as discussed in Section 3.3, and the modulated signal will just be another MSK signal. The phase tree of the demodulated DMSK signal is shown in Figure 3.6(b). As a result of the differential encoding, demodulation and detection, $\psi_d(t, \boldsymbol{\beta})$ for DMSK never goes above $\pi$. The phase trees of modulated and demodulated 4-DCPFSK are shown in Figures 3.7(a) and (b) respectively. They exhibit similar behaviour to those of DMSK. The phase tree of the modulated signal is identical to that of 4-CPFSK.



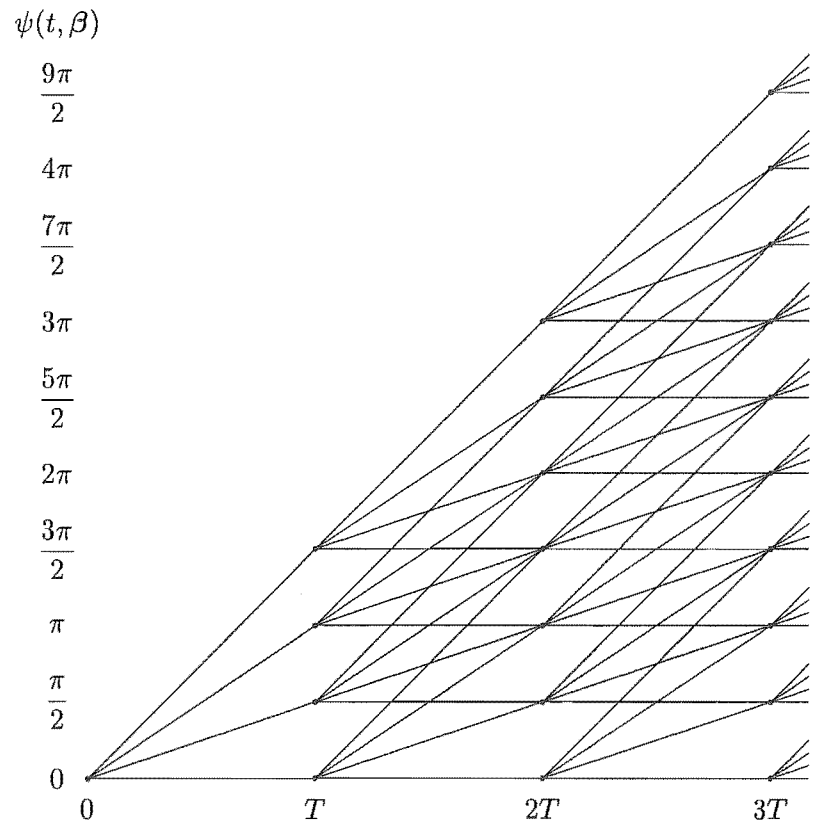

(a) Modulated



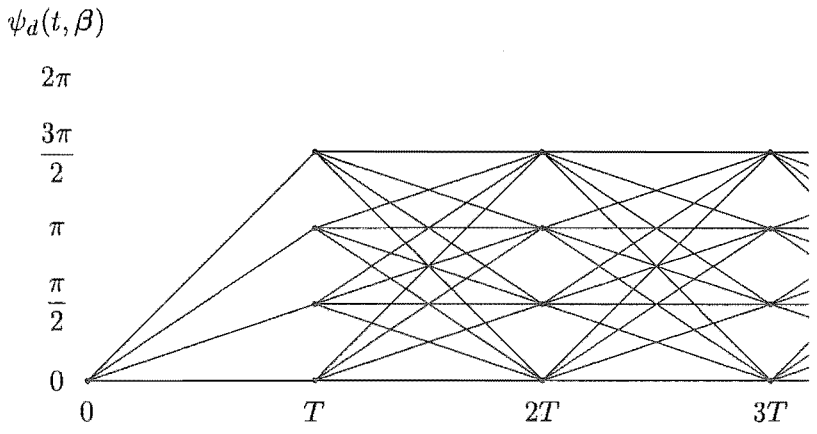

(b) Demodulated

**Figure 3.6:** Phase trees for DMSK

This will always be the case when $M$ and $P$ are such that $B = M$. Also, the phase tree of the demodulated signal does not go above $2\pi$. In fact, the phase tree of a demodulated DCPFSK signal will never go above $2\pi h(B-1)$, so that only DCPFSK schemes with $M > P$ will have demodulated signals whose phase trees go above $2\pi$. It is important to note that the demodulated phase trees in Figures 3.6(b) and 3.7(b) contain new slopes when compared to their demodulated coherent counterparts. As the slope of the phase of signal determines its frequency content, demodulated DCPFSK signals have different spectra to that of demodulated CPFSK signals.



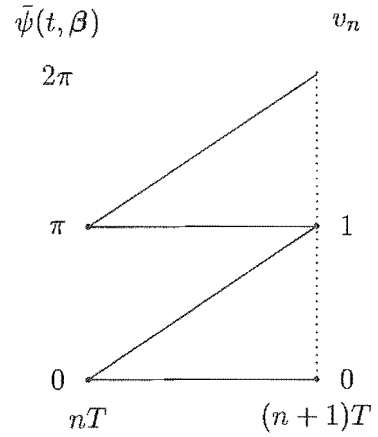

(a) Modulated



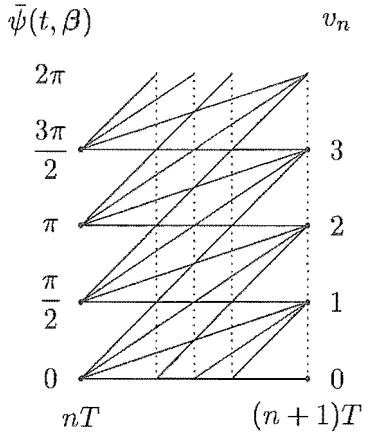

(b) Demodulated

**Figure 3.7:** Phase trees for 4-DCPFSK

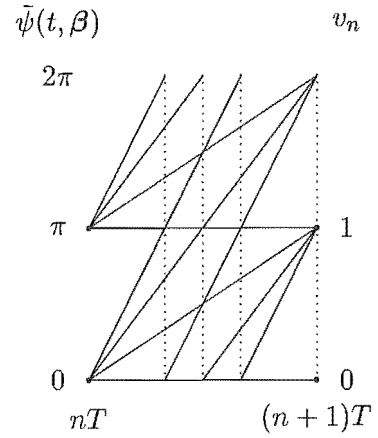
Although we have again looked at the phase trees of DMSK and 4-DCPFSK, they are used only as a development towards the phase trellises. As in the coherent case, these are more important, and are considered in the next section.



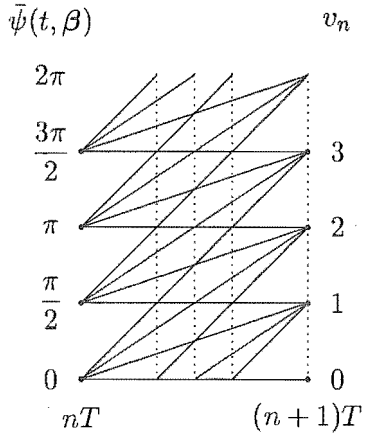

(a) DMSK ($M = 2$, $h = 1/2$)




(b) DCPFSK with $M = 2$ and $h = 1/4$




(c) DCPFSK with $M = 4$ and $h = 1/2$

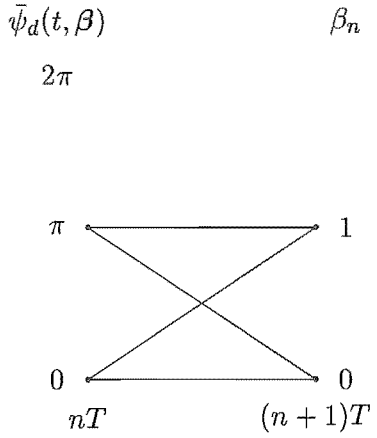



(d) 4-DCPFSK ($M = 4$, $h = 1/4$)

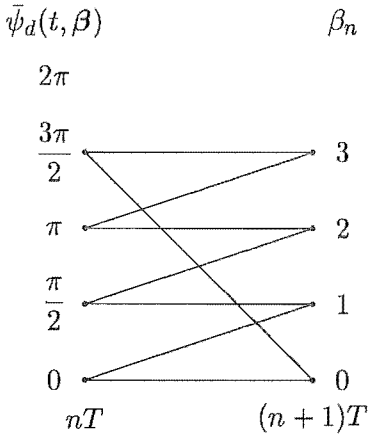
**Figure 3.8:** Phase trellis examples for modulated DCPFSK

## 3.6 Phase Trellis

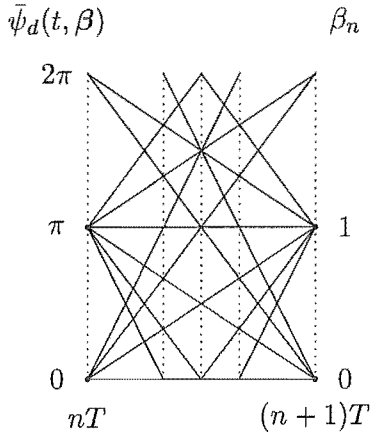
The phase trellis is generated in the same manner to Section 2.4, by taking the phase tree modulo-$2\pi$. Consequently, modulated and demodulated signals in a DCPFSK system have different phase trellises. We look first at some examples of the phase trellises of modulated DCPFSK signals, shown in Figure 3.8. Except for DCPFSK with $M = 2$ and $h = 1/4$, these are identical to those in Figure 2.3. As discussed in Section 3.5, this is because the case in Figure 3.8(b) is the only one where $B \neq M$. Note too that DCPFSK with $M = 2$ and $h = 1/4$ and 4-DCPFSK have the same modulated phase trellis. However, this is a little misleading. In the 4-DCPFSK case, the signal transmitted depends only on the



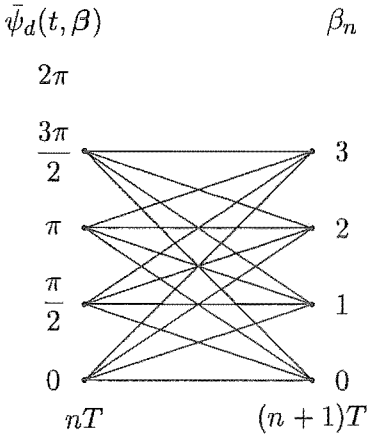

(a) DMSK ($M = 2$, $h = 1/2$)




(b) DCPFSK with $M = 2$ and $h = 1/4$




(c) DCPFSK with $M = 4$ and $h = 1/2$




(d) 4-DCPFSK ($M = 4$, $h = 1/4$)

**Figure 3.9:** Phase trellis examples for demodulated DCPFSK

starting phase $v_n$. Whereas in the $M = 2$ and $h = 1/4$ case, the signal path taken depends on the two previous values of $v_n$. Nonetheless, Figure 3.8(b) shows all the signal paths that can be transmitted in a DCPFSK system with $M = 2$ and $h = 1/4$.

Figure 3.9 shows some example phase trellises of demodulated DCPFSK signals. All the phase trellises are very different to those in both Figures 2.3 and 3.8. The most important point is that the slope of the phase path taken depends on the starting phase value. In the coherent case the phase slope was independent of the starting phase value. Again, as in the phase trees of the demodulated DCPFSK signals, the phase trellises have new slopes, and therefore different spectra to the equivalent coherent cases.

Note that only DCPFSK with $M = 4$ and $h = 1/2$ "wraps" modulo-$2\pi$, as it is the only case where $M > P$. The DCPFSK with $M = 4$ and $h = 1/2$ case is also unique as it is the only one where more than $M$ different slopes are possible from each phase state. As discussed in Section 3.3, we can detect the transmitted symbol by reducing the slope modulo-$M$.

Previously, we pointed out that the signal path taken in Figure 3.8(b) depended on the two previous values of $v_n$. This is not the case with the demodulated phase trellis in Figure 3.9(b), as its signals depend only on the starting phase. The phase trellis of a demodulated DCPFSK signal is always time-invariant and thus we can again use the Viterbi algorithm to detect the transmitted symbols.

## 3.7 Detection of DCPFSK

The presence of the noise × noise and signal × noise terms in $y_{\mathrm{d}}(t, \boldsymbol{\beta})$ makes it non-Gaussian, and thus the optimal detection of DCPFSK is a very complex problem. Moreover, a simpler differential modulation scheme—differential phase shift keying (DPSK)—shows very little improvement in performance when an optimal differential metric is used in place of a sub-optimal one based on the optimal coherent metric [van96]. We choose to use a receiver for DCPFSK similar to the coherent case, that seeks to minimise the squared Euclidean distance between the *differentially-demodulated* received signal and the possible *differentially-demodulated* transmitted signals. This may perform worse than an optimal receiver, but will be considerably less complicated.

Thus our receiver seeks to maximise the function

$$-\int_{-\infty}^{\infty} \left| y_{\mathrm{d}}(t, \boldsymbol{\beta}) - \hat{y}_{\mathrm{d}}(t, \hat{\boldsymbol{\beta}}) \right|^2 dt \tag{3.35}$$

with respect to $\hat{\boldsymbol{U}}$, where $\hat{y}_{\mathrm{d}}(t, \hat{\boldsymbol{\beta}})$ is the signal × signal term of $y_{\mathrm{d}}(t, \hat{\boldsymbol{\beta}})$. $\Pr\left\{ y_{\mathrm{d}}(t, \boldsymbol{\beta}) | \hat{\boldsymbol{U}} \right\}$ is the probability density function (pdf) of the received signal $y_{\mathrm{d}}(t, \boldsymbol{\beta})$ conditioned on $\hat{\boldsymbol{U}}$. The maximising sequence $\hat{\boldsymbol{U}}$ is the sequence estimate that minimises the squared Euclidean distance between the differentially-demodulated received signal $y_{\mathrm{d}}(t, \boldsymbol{\beta})$ and a hypothesised differentially-demodulated transmitted signal. As DCPFSK signals are constant envelope, we can equivalently maximise the complex correlation

$$J_d(\hat{\boldsymbol{U}}) \;=\; \int_{-\infty}^{\infty} y_{\mathrm{d}}(t, \boldsymbol{\beta})\, \hat{y}_{\mathrm{d}}^*(t, \hat{\boldsymbol{\beta}})\, dt. \tag{3.36}$$

To simplify processing, we define

$$J_{d,n}(\hat{\boldsymbol{U}}) \;=\; \int_{-\infty}^{(n+1)T} y_{\mathrm{d}}(t, \boldsymbol{\beta})\, \hat{y}_{\mathrm{d}}^*(t, \hat{\boldsymbol{\beta}})\, dt, \tag{3.37}$$

which allows us to write the recursion equation

$$J_{d,n}(\hat{\boldsymbol{U}}) \;=\; J_{d,n-1}(\hat{\boldsymbol{U}}) + \lambda_{d,n}(\hat{\boldsymbol{U}}), \tag{3.38}$$

where

$$\begin{aligned}
\lambda_{d,n}(\hat{\boldsymbol{U}}) \;&=\; \int_{nT}^{(n+1)T} y_{\mathrm{d}}(t,\boldsymbol{\beta})\, \hat{y}_{\mathrm{d}}^{*}(t,\hat{\boldsymbol{\beta}})\, dt \\
&=\; \int_{nT}^{(n+1)T} y_{\mathrm{d}}(t,\boldsymbol{\beta}) \exp\Big[2\pi f_0 t - \bar{\psi}_d(t,\hat{\boldsymbol{\beta}})\Big]\, dt \\
\Leftrightarrow \quad \lambda_{d,n}(\hat{\beta}_n,\hat{\beta}_{n-1}) \;&=\; \int_{nT}^{(n+1)T} y_{\mathrm{d}}(t,\boldsymbol{\beta}) \exp\left[2\pi f_0 t - 2\pi h(\hat{\beta}_n - \hat{\beta}_{n-1})\frac{t}{T} - 2\pi h\hat{\beta}_{n-1}\right] dt.
\end{aligned} \tag{3.39}$$

As in the coherent case, $\lambda_{d,n}(\hat{\beta}_n,\hat{\beta}_{n-1})$ can be realised using a bank of $M\cdot P$ filters that are sampled at $t=(n+1)T$. The receiver correlates the differentially-demodulated received signal over one symbol period with all possible differentially-demodulated transmitted signals over that symbol period. Note that we have left out the amplitude scaling of $\sqrt{E/T}$ in (3.39) as it does not affect performance.

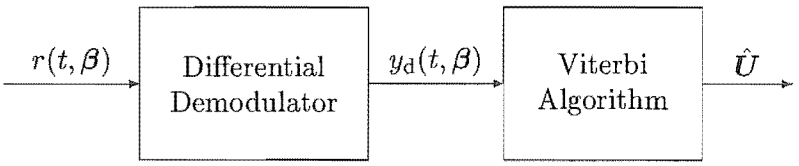



**Figure 3.10:** A receiver structure for DCPFSK

## 3.8 Receiver Structure for DCPFSK

Our receiver consists of two main blocks: a differential demodulator and a Viterbi processor as shown in Figure 3.10. The differential demodulator was discussed in Section 3.2, it accepts the received signal $r(t,\boldsymbol{\beta})$ given in (3.1) with $\boldsymbol{U}$ replaced by $\boldsymbol{\beta}$, and outputs the demodulated DCPFSK signal $y_{\mathrm{d}}(t,\boldsymbol{\beta})$ given by (3.10), again with $\boldsymbol{U}$ replaced by $\boldsymbol{\beta}$. We now discuss the Viterbi processor.

### 3.8.1 Viterbi Processor

The metric given in (3.39) can be rewritten as

$$\begin{aligned}\lambda_{c,n}(\hat{U}_n, \hat{V}_n) &= \cos(2\pi h\hat{\beta}_{n-1}) \int_{nT}^{(n+1)T} y_{\mathrm{d,I}}(t,\beta) \cos\left[2\pi h(\hat{\beta}_n - \hat{\beta}_{n-1})\frac{t}{T}\right] dt \\ &- \cos(2\pi h\hat{\beta}_{n-1}) \int_{nT}^{(n+1)T} y_{\mathrm{d,Q}}(t,\beta) \sin\left[2\pi h(\hat{\beta}_n - \hat{\beta}_{n-1})\frac{t}{T}\right] dt \\ &- \sin(2\pi h\hat{\beta}_{n-1}) \int_{nT}^{(n+1)T} y_{\mathrm{d,Q}}(t,\beta) \cos\left[2\pi h(\hat{\beta}_n - \hat{\beta}_{n-1})\frac{t}{T}\right] dt \\ &- \sin(2\pi h\hat{\beta}_{n-1}) \int_{nT}^{(n+1)T} y_{\mathrm{d,I}}(t,\beta) \sin\left[2\pi h(\hat{\beta}_n - \hat{\beta}_{n-1})\frac{t}{T}\right] dt.\end{aligned} \tag{3.40}$$

Due to the $M-1$ possible values of $\hat{\beta}_n - \hat{\beta}_{n-1}$, this can be computed by $4M-2$ baseband filters with the impulse responses

$$h_{\mathrm{d,I},\hat{\beta}_n-\hat{\beta}_{n-1}}(t) = \begin{cases} 2\cos\left[2\pi h(\hat{\beta}_n - \hat{\beta}_{n-1})\dfrac{T-t}{T}\right], & 0 \leqslant t < T \\ 0, & \text{otherwise,} \end{cases} \tag{3.41}$$

$$h_{\mathrm{d,Q},\hat{\beta}_n-\hat{\beta}_{n-1}}(t) = \begin{cases} 2\sin\left[2\pi h(\hat{\beta}_n - \hat{\beta}_{n-1})\dfrac{T-t}{T}\right], & 0 \leqslant t < T \\ 0, & \text{otherwise.} \end{cases} \tag{3.42}$$

Figure 3.11 shows how this might be implemented. The Viterbi processor uses the Viterbi algorithm (discussed in Appendix C) to perform MLSE and produce an estimate $\hat{U}$ of the transmitted data. Note that the multiplication and addition structure is repeated up to $P$ times (depending on the filter), to hypothesise each possible value of $\beta_{n-1}$.

The Viterbi algorithm then proceeds as described in Appendix C to produce an estimate of the transmitted data sequence $\hat{U}$.

## 3.9 Euclidean Distance Properties of DCPFSK

As discussed in Section 3.8, our receiver structure is based on squared Euclidean distance (SED). Thus the Euclidean distance properties of a DCPFSK scheme will contribute significantly to the scheme's performance. With coherent demodulation, the signal paths before and after demodulation are the same, and the SED can be calculated using the inputs to the memoryless modulator, shown in Section 2.8. This is not the case with differential demodulation, as the differential demodulator in Figure 3.1 also performs differential decoding. However, the differential decoding is not the exact inverse of the differential

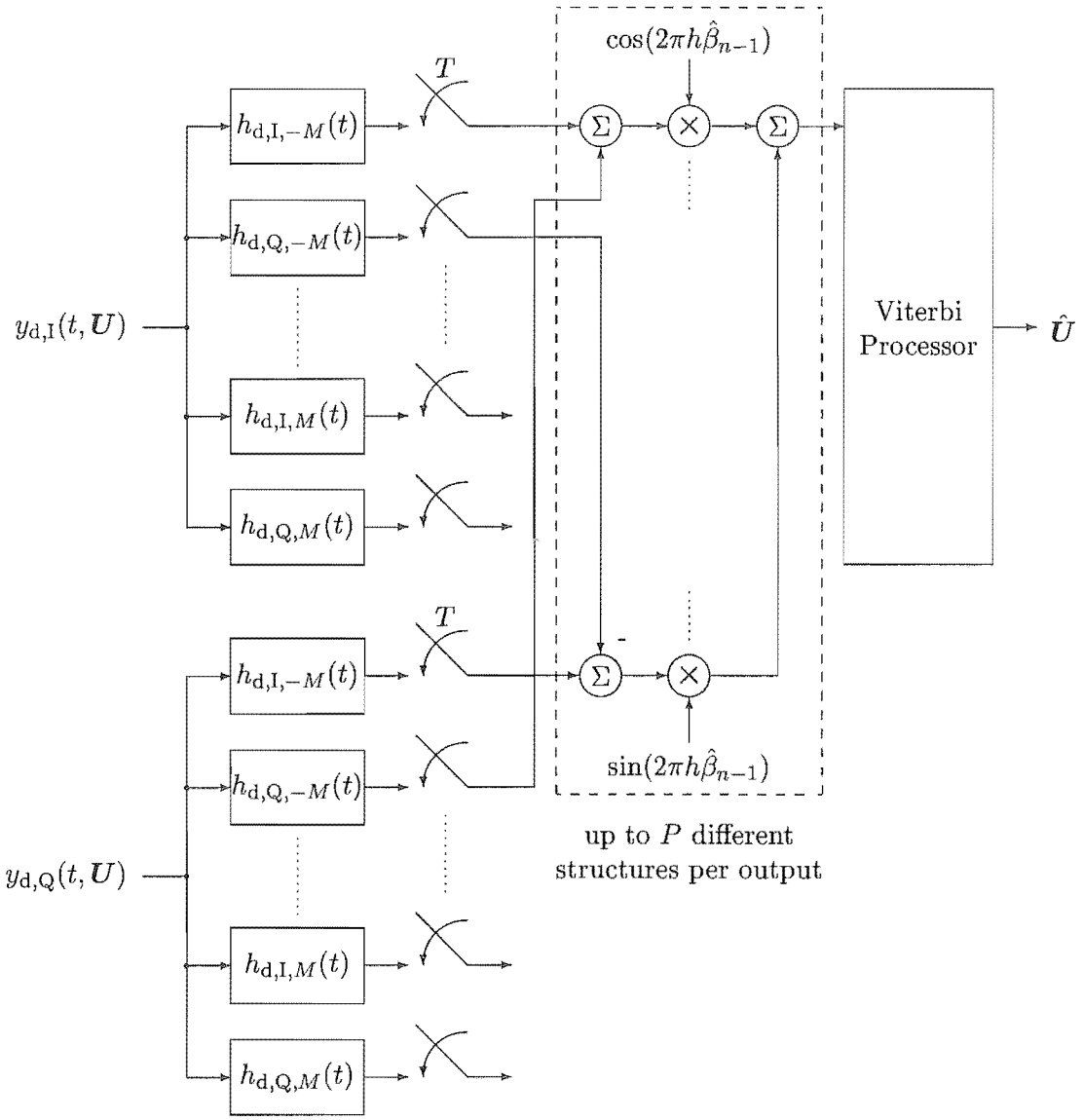



**Figure 3.11:** Detection Processor

encoding, so that the signal paths are changed. Thus the SED of DCPFSK is different to that of CPFSK, we must look at the SED between *demodulated* DCPFSK signals rather than the transmitted signals as in the coherent case.

Thus we define the normalised squared Euclidean distance (NSED) between two DCPFSK signals $y_{\mathrm{d}}(t,\beta)$ and $y_{\mathrm{d}}(t,\hat{\beta})$ as

$$d^2(\beta,\hat{\beta}) \triangleq \frac{\log_2 M}{2E}\int_{-\infty}^{\infty}\left|y_{\mathrm{d}}(t,\beta)-y_{\mathrm{d}}(t,\hat{\beta})\right|^2 dt, \tag{3.43}$$

which can be re-written as

$$d^2(\beta,\hat{\beta}) = \frac{\log_2 M}{2E}\sum_n \int_{nT}^{(n+1)T} \left|y_{\rm d}(t,\beta) - y_{\rm d}(t,\hat{\beta})\right|^2 dt$$
$$= \sum_n d_n^2(\beta,\hat{\beta}), \tag{3.44}$$

where $d_n^2(\beta,\hat{\beta})$ is called the normalised incremental squared Euclidean distance (NISED), and is defined as

$$d_n^2(\beta,\hat{\beta}) = \frac{\log_2 M}{2E}\int_{nT}^{(n+1)T} \left|y_{\rm d}(t,\beta) - y_{\rm d}(t,\hat{\beta})\right|^2 dt. \tag{3.45}$$

Finally, the most important Euclidean distance property is the normalised minimum squared Euclidean distance (NMSED), which is defined as

$$d_{\rm min}^2 \triangleq \min_{\substack{\beta,\hat{\beta}\\ \beta\neq\hat{\beta}}} d^2(\beta,\hat{\beta}). \tag{3.46}$$

We now calculate the NISED and NMSED of DCPFSK.

### 3.9.1 Incremental Squared Euclidean Distance

Let us write the noise-free version of $y_{\rm d}(t,\beta)$ as

$$y_{\rm d}(t,\beta) = \sqrt{\frac{E}{T}}\exp(j\left[\Omega_{n-1} + \phi_n(t-nT)\right]), \qquad nT \le t < (n+1)T, \tag{3.47}$$

where, for convenience, we have defined

$$\Omega_{n-1} \triangleq 2\pi h\beta_{n-1} \tag{3.48}$$
$$\phi_n \triangleq \frac{2\pi h}{T}(\beta_n - \beta_{n-1}). \tag{3.49}$$

Note that $\phi_n T = \Omega_n - \Omega_{n-1}$, as we will use this in the following. We can re-write the NISED given in (3.45) as

$$d_n^2(\beta,\hat{\beta}) = \frac{\log_2 M}{2E}\int_{nT}^{(n+1)T} \left|y_{\rm d}(t,\beta)\right|^2 dt + \frac{\log_2 M}{2E}\int_{nT}^{(n+1)T} \left|y_{\rm d}(t,\hat{\beta})\right|^2 dt$$
$$- \frac{\log_2 M}{E}\,{\rm Re}\left[\int_{nT}^{(n+1)T} y_{\rm d}(t,\beta)\, y_{\rm d}^*(t,\hat{\beta})\, dt\right]. \tag{3.50}$$

Now as $y_{\mathrm{d}}(t,\beta)$ and $y_{\mathrm{d}}(t,\hat{\beta})$ are both constant envelope, the first two terms are easily calculated. Looking at the first term, we have

$$\begin{aligned}\frac{\log_2 M}{2E}\int_{nT}^{(n+1)T}\left|y_{\mathrm{d}}(t,\beta)\right|^2 dt &= \frac{\log_2 M}{2E}\int_{nT}^{(n+1)T}\left|\sqrt{\frac{E}{T}}\exp(j\left[\Omega_{n-1}+\phi_n(t-nT)\right])\right|^2 dt\\ &= \frac{\log_2 M}{2E}\int_{nT}^{(n+1)T}\frac{E}{T}\,dt\\ &= \frac{\log_2 M}{2E}\cdot E\\ &= \frac{\log_2 M}{2}. \end{aligned} \tag{3.51}$$

Similarly

$$\frac{\log_2 M}{2E}\int_{nT}^{(n+1)T}\left|y_{\mathrm{d}}(t,\hat{\beta})\right|^2 dt = \frac{\log_2 M}{2}. \tag{3.52}$$

The integral in the third term is

$$\begin{aligned}&\int_{nT}^{(n+1)T} y_{\mathrm{d}}(t,\beta)\,y_{\mathrm{d}}^{*}(t,\hat{\beta})\,dt\\ &\quad= \frac{E}{T}\int_{nT}^{(n+1)T}\exp(j\left[\Omega_{n-1}+\phi_n(t-nT)+\right])\exp\Big(-j\left[\hat{\Omega}_{n-1}+\hat{\phi}_n(t-nT)\right]\Big)\,dt\\ &\quad= \frac{E}{T}\int_{nT}^{(n+1)T}\exp\Big(j\left[\Omega_{n-1}-\hat{\Omega}_{n-1}+(\phi_n-\hat{\phi}_n)(t-nT)\right]\Big)\,dt\\ &\quad= \frac{E}{jT(\phi_n-\hat{\phi}_n)}\Big[\exp\Big(j\left[\Omega_{n-1}-\hat{\Omega}_{n-1}+(\phi_n-\hat{\phi}_n)(t-nT)\right]\Big)\Big]_{t=nT}^{(n+1)T}\\ &\quad= \frac{E}{jT(\phi_n-\hat{\phi}_n)}\Big[\exp\Big(j\left[\Omega_{n-1}-\hat{\Omega}_{n-1}+(\phi_n-\hat{\phi}_n)T\right]\Big)-\exp\Big(j\left[\Omega_{n-1}-\hat{\Omega}_{n-1}\right]\Big)\Big]\\ &\quad= \frac{E}{jT(\phi_n-\hat{\phi}_n)}\Big[\exp\Big(j\left[\Omega_n-\hat{\Omega}_n\right]\Big)-\exp\Big(j\left[\Omega_{n-1}-\hat{\Omega}_{n-1}\right]\Big)\Big] \end{aligned} \tag{3.53}$$

Substituting (3.51)-(3.53) into (3.50), we have

$$\begin{aligned}d_n^2(\beta,\hat{\beta}) &= \log_2 M-\frac{\log_2 M}{T(\phi_n-\hat{\phi}_n)}\,\mathrm{Re}\left[\frac{1}{j}\exp\Big(j\left[\Omega_n-\hat{\Omega}_n\right]\Big)-\exp\Big(j\left[\Omega_{n-1}-\hat{\Omega}_{n-1}\right]\Big)\right]\\ &= \log_2 M-\frac{\log_2 M}{T(\phi_n-\hat{\phi}_n)}\,\mathrm{Im}\left[\exp\Big(j\left[\Omega_n-\hat{\Omega}_n\right]\Big)-\exp\Big(j\left[\Omega_{n-1}-\hat{\Omega}_{n-1}\right]\Big)\right]\\ &= \log_2 M-\frac{\log_2 M\left[\sin\Big(\Omega_n-\hat{\Omega}_n\Big)-\sin\Big(\Omega_{n-1}-\hat{\Omega}_{n-1}\Big)\right]}{T(\phi_n-\hat{\phi}_n)}\\ &= \log_2 M\left[1-\frac{\sin\Big(\Omega_n-\hat{\Omega}_n\Big)-\sin\Big(\Omega_{n-1}-\hat{\Omega}_{n-1}\Big)}{\Omega_n-\Omega_{n-1}-\hat{\Omega}_n+\hat{\Omega}_{n-1}}\right]. \end{aligned} \tag{3.54}$$

Using (3.48), we can write (3.54) in terms of $\beta_n$

$$d_n^2\left(\begin{bmatrix}\beta_n\\ \beta_{n-1}\end{bmatrix},\begin{bmatrix}\hat{\beta}_n\\ \hat{\beta}_{n-1}\end{bmatrix}\right) = \log_2 M\left[1-\frac{\sin(2\pi h[\beta_n-\hat{\beta}_n])-\sin(2\pi h[\beta_{n-1}-\hat{\beta}_{n-1}])}{2\pi h(\beta_n-\beta_{n-1}-\hat{\beta}_n+\hat{\beta}_{n-1})}\right]. \tag{3.55}$$

In the case where

$$\beta_n - \beta_{n-1} = \hat{\beta}_n - \hat{\beta}_{n-1}, \tag{3.56}$$

the denominator of the second term in (3.55) equal to zero and $d_n^2(\beta, \hat{\beta})$ is undefined. However (3.57) also implies

$$\begin{aligned} \beta_n - \hat{\beta}_n &= \beta_{n-1} - \hat{\beta}_{n-1} \\ \Leftrightarrow \quad \sin 2\pi h(\beta_n - \hat{\beta}_n) &= \sin 2\pi h(\beta_{n-1} - \hat{\beta}_{n-1}), \end{aligned} \tag{3.57}$$

so that the numerator of the second term in (3.55) is also equal to zero. For convenience, we define $\Phi_n$ as

$$\Phi_n \triangleq \beta_n - \hat{\beta}_n. \tag{3.58}$$

We will use L'Hôpital's rule to find the limit of $d_n^2(\beta, \hat{\beta})$ as $\Phi_n \to \Phi_{n-1}$ by differentiating the numerator and denominator of the second term in (3.55) with respect to $\Phi_n$. Thus,

$$\begin{aligned} \lim_{\Phi_n \to \Phi_{n-1}} d_n^2(\beta, \hat{\beta}) &= \lim_{\Phi_n \to \Phi_{n-1}} \log_2 M \left[1 - \frac{\sin 2\pi h\Phi_n - \sin 2\pi h\Phi_{n-1}}{2\pi h(\Phi_n - \Phi_{n-1})}\right] \\ &= \log_2 M \left[1 - \lim_{\Phi_n \to \Phi_{n-1}} \frac{\sin 2\pi h\Phi_n - \sin 2\pi h\Phi_{n-1}}{2\pi h(\Phi_n - \Phi_{n-1})}\right] \\ &= \log_2 M \left[1 - \lim_{\Phi_n \to \Phi_{n-1}} \cos 2\pi h\Phi_n\right] \\ &= \log_2 M \left[1 - \cos 2\pi h\Phi_{n-1}\right] \\ &= \log_2 M \left[1 - \cos 2\pi h(\beta_{n-1} - \hat{\beta}_{n-1})\right] \end{aligned} \tag{3.59}$$

We can now define the normalised incremental squared Euclidean distance of DCPFSK as

$$d_n^2\left(\begin{bmatrix}\beta_n \\ \beta_{n-1}\end{bmatrix}, \begin{bmatrix}\hat{\beta}_n \\ \hat{\beta}_{n-1}\end{bmatrix}\right) \triangleq$$

$$\begin{cases} \log_2 M \left[1 - \dfrac{\sin(2\pi h[\beta_n - \hat{\beta}_n]) - \sin(2\pi h[\beta_{n-1} - \hat{\beta}_{n-1}])}{2\pi h(\beta_n - \beta_{n-1} - \hat{\beta}_n + \hat{\beta}_{n-1})}\right], & \beta_n - \beta_{n-1} \neq \hat{\beta}_n - \hat{\beta}_{n-1} \\ \log_2 M \left[1 - \cos 2\pi h(\beta_{n-1} - \hat{\beta}_{n-1})\right], & \beta_n - \beta_{n-1} = \hat{\beta}_n - \hat{\beta}_{n-1} \end{cases} \tag{3.60}$$

### 3.9.2 Minimum Squared Euclidean Distance

The NMSED of DCPFSK can be found in a manner similar to [Rim91]. We first note that from the definition of $\beta_n$,

$$\begin{aligned}
\sin(2\pi h\beta_n) &= \sin\left(R_{2\pi}[2\pi h\beta_n]\right) \\
&= \sin\left(R_{2\pi}\left[2\pi\left(\frac{K}{P}\right)R_M\left[\sum_{i=0}^{n-1}U_i\right]\right]\right) \\
&= \sin\left(R_{2\pi}\left[2\pi\left(\frac{K}{P}\right)R_P\left[R_M\left[\sum_{i=0}^{n-1}U_i\right]\right]\right]\right) \\
&= \sin\left(R_{2\pi}\left[2\pi h R_P\left[\sum_{i=0}^{n-1}U_i\right]\right]\right) \\
&= \sin\left(2\pi h R_P\left[\sum_{i=0}^{n-1}U_i\right]\right) \qquad (3.61)
\end{aligned}$$

where the second-to-last step holds because, by definition, $B = P$, or $P$ is a factor of $B$. Let us look further at the NISED of DCPFSK given by (3.60). For simplicity, we only consider the case where $\beta_n - \beta_{n-1} \neq \hat{\beta}_n - \hat{\beta}_{n-1}$. Substituting (3.28) into (3.60), and noting (3.61), we obtain

$$\begin{aligned}
d_n^2(\beta,\hat{\beta}) &= \log_2 M \times \\
&\left[1 - \frac{\sin\left[2\pi h\left(R_P[\sum_{i=0}^{n}U_i] - R_P\left[\sum_{i=0}^{n}\hat{U}_i\right]\right)\right] - \sin\left[2\pi h\left(R_P\left[\sum_{i=0}^{n-1}U_i\right] - R_P\left[\sum_{i=0}^{n-1}\hat{U}_i\right]\right)\right]}{2\pi h\left(R_B[\sum_{i=0}^{n}U_i] - R_B\left[\sum_{i=0}^{n-1}U_i\right] - R_B\left[\sum_{i=0}^{n}\hat{U}_i\right] + R_B\left[\sum_{i=0}^{n-1}\hat{U}_i\right]\right)}\right] \qquad (3.62) \\
&= \log_2 M\left[1 - \frac{\sin(2\pi h[\Theta_n - \hat{\Theta}_n]) - \sin(2\pi h[\Theta_{n-1} - \hat{\Theta}_{n-1}])}{2\pi h(\beta_n - \beta_{n-1} - \hat{\beta}_n + \hat{\beta}_{n-1})}\right], \qquad (3.63)
\end{aligned}$$

where we have defined

$$\Theta_n \triangleq R_P\left[\sum_{i=0}^{n}U_i\right], \qquad (3.64)$$

and

$$\hat{\Theta}_n \triangleq R_P\left[\sum_{i=0}^{n}\hat{U}_i\right]. \qquad (3.65)$$

We can think of $\Theta_n$ and $\hat{\Theta}_n$ as the state of the transmitter and receiver respectively.

$\Theta_n$ can be determined from $\Theta_{n-1}$ as

$$\begin{aligned}\Theta_n &= R_P\left[\sum_{i=0}^{n-1} U_i + U_n\right] \\ &= R_P\left[R_P\left[\sum_{i=0}^{n-1} U_i\right] + U_n\right] \\ &= R_P[\Theta_{n-1} + U_n] \end{aligned} \tag{3.66}$$

and $\Theta_n$ can be determined from $\Theta_{n+1}$ using

$$\begin{aligned}\Theta_n &= R_P\left[\sum_{i=0}^{n+1} U_i - U_n\right] \\ &= R_P\left[R_P\left[\sum_{i=0}^{n+1} U_i\right] - U_n\right] \\ &= R_P[\Theta_{n+1} - U_n]\,. \end{aligned} \tag{3.67}$$

Similarly

$$\hat{\Theta}_n = R_P\left[\hat{\Theta}_{n-1} + \hat{U}_n\right] \tag{3.68}$$

and

$$\hat{\Theta}_n = R_P\left[\hat{\Theta}_{n+1} - \hat{U}_{n+1}\right]. \tag{3.69}$$

We can construct a super trellis whose states are defined by the super states $(\Theta_n, \hat{\Theta}_n)$, and whose transitions depend on the inputs $U_n$ and $\hat{U}_n$, as in (3.66) and (3.68). As $\Theta_n$ and $\hat{\Theta}_n$ each have $P$ states, there are $P^2$ super states in the super trellis. When $\Theta_n \neq \hat{\Theta}_n$, there are errors in the system. Consequently we call such a state an *error state.* There are $P$ *error-free states*, occurring when $\Theta_n = \hat{\Theta}_n$. We call the $P$ paths that trace through these error-free states in the super trellis the *error-free paths.* It is clear that the error-free paths will have the input $U_n = \hat{U}_n$, $\forall\ n$.

We wish to find one path that achieves the smallest non-zero squared Euclidean distance and to compute the SED of that path. Such a path will lie on an error-free path for most of the way. As the trellis is time-invariant, it is not important when such a path leaves an error-free path. Let us assume that this happens in the $i$-th interval. So we are looking for a detour that leaves an error-free path in the $i$-th interval, stays unmerged over an unknown number of $L$ intervals and merges with any error-free path in the $(i+L)$-th interval, such that the total SED is minimal. We refer to the first interval of the detour as the *diverge* and the last interval as the *merge.*

We first consider detours of length 1. In this case the diverge and the merge are in the same interval. We will denote the minimum SED of such a detour as $d^2_{\text{min},1}$. Since the detour begins in the $i$-th interval, we have $\Theta_{i-1} = \hat{\Theta}_{i-1}$ and $U_i \neq \hat{U}_i$. As the merge is in the same interval we must also have $\Theta_i = \hat{\Theta}_i$. Using (3.66) and (3.68) gives us

$$\begin{aligned} & \Theta_i = \hat{\Theta}_i \\ \Leftrightarrow \quad & R_P[\Theta_{i-1} + U_i] = R_P\left[\hat{\Theta}_{i-1} + \hat{U}_i\right] \\ \Leftrightarrow \quad & R_P[U_i] = R_P\left[\hat{U}_i\right] \\ \Leftrightarrow \quad & U_i = \hat{U}_i + kP, \end{aligned} \tag{3.70}$$

where we have made use of (A.6). We have already stated that $U_i \neq \hat{U}_i$, so $k$ must be a non-zero integer in (3.70). As $U_i$ and $\hat{U}_i$ are $M$-ary digits, condition (3.70) can only be satisfied if and only if $M > P$. Substituting $\Theta_{i-1} = \hat{\Theta}_{i-1}$ and $\Theta_i = \hat{\Theta}_i$ into (3.63), we find

$$d^2_{\text{min},1} = \log_2 M \left[1 - \frac{0}{2\pi h(\beta_i - \beta_{i-1} - \hat{\beta}_i + \hat{\beta}_{i-1})}\right] = \log_2 M. \tag{3.71}$$

However, we need to show that $\beta_i - \beta_{i-1} - \hat{\beta}_i + \hat{\beta}_{i-1} \neq 0$. Let us look at the quantity $\beta_i - \beta_{i-1}$ first. Let us suppose that $\beta_{i-1}$ equals some $M$-ary value, $\Omega$. Then, from (3.31) with $B = M$, we have

$$\begin{aligned} \beta_i &= R_M[\beta_{i-1} + U_i] \\ &= R_M[\Omega + U_i] \\ &= \Omega + U_i - kM \end{aligned} \tag{3.72}$$

where

$$k = \begin{cases} 0 & \Omega + U_i < M \\ 1 & \Omega + U_i \geq M. \end{cases} \tag{3.73}$$

Thus we can write

$$\begin{aligned} \beta_i - \beta_{i-1} &= \Omega + U_i - kM - \Omega \\ &= U_i - kM \end{aligned} \tag{3.74}$$

and through similar reasoning,

$$\hat{\beta}_i - \hat{\beta}_{i-1} = \hat{U}_i - \hat{k}M \tag{3.75}$$

where

$$\hat{k} = \begin{cases} 0 & \hat{\Omega} + \hat{U}_i < M \\ 1 & \hat{\Omega} + \hat{U}_i \geq M, \end{cases} \tag{3.76}$$

and $\hat{\Omega}$ is some $M$-ary value. Using (3.74) and (3.75),

$$\beta_i - \beta_{i-1} - \hat{\beta}_i + \hat{\beta}_{i-1} = U_i - \hat{U}_i - (k - \hat{k})M. \tag{3.77}$$

As $U_i$ and $\hat{U}_i$ are $M$-ary symbols, $U - \hat{U} < |M - 1|$. Also $k - \hat{k} \in \{-1, 0, 1\}$, so that the only way that the RHS of (3.77) can be equal to zero is if $k - \hat{k} = 0$ and $U_i = \hat{U}_i$. But we have already stated that $U_i \neq \hat{U}_i$, so $\beta_i - \beta_{i-1} - \hat{\beta}_i + \hat{\beta}_{i-1}$ must be non-zero, and (3.71) holds.

Thus the NMSED of detours of length 1 when $M > P$ is given by

$$d^2_{\text{min},1} = \log_2 M. \tag{3.78}$$

For the case where $M \leq P$, there are no detours of length 1.

We now consider detours of length $L > 1$. We look first at a diverge, as we start in an error-free state we must have

$$\Theta_{i-1} = \hat{\Theta}_{i-1} \tag{3.79}$$

and

$$\Theta_i \neq \hat{\Theta}_i \tag{3.80}$$

which imply $U_i \neq \hat{U}_i$. It is convenient to consider the cases $M \leq P$ and $M > P$ separately. We first consider the case where $M \leq P$, so that $B = P$, $\beta_n = \Theta_n$ and $\hat{\beta}_n = \hat{\Theta}_n$. This allows us to re-write (3.63) as

$$d^2_n(\beta, \hat{\beta}) = \log_2 M \left[ 1 - \frac{\sin(2\pi h[\Theta_n - \hat{\Theta}_n]) - \sin(2\pi h[\Theta_{n-1} - \hat{\Theta}_{n-1}])}{2\pi h(\Theta_n - \Theta_{n-1} - \hat{\Theta}_n + \hat{\Theta})} \right]. \tag{3.81}$$

Substituting (3.79) and (3.80) into (3.81), we find the NSED of a diverge is

$$\begin{aligned} d^2_{i,\text{diverge}}(\beta, \hat{\beta}) &= \log_2 M \left[ 1 - \frac{\sin(2\pi h[\Theta_i - \hat{\Theta}_i])}{2\pi h(\Theta_i - \hat{\Theta}_i)} \right] \\ &= \log_2 M \left[ 1 - \frac{\sin 2\pi h \Psi_i}{2\pi h \Psi_i} \right], \end{aligned} \tag{3.82}$$

where we have defined

$$\Psi_i \triangleq \Theta_i - \hat{\Theta}_i. \tag{3.83}$$

Note that $\Psi_i \in \{\pm 1, \pm 2, \ldots, \pm(P-1)\}$ as $\Theta_i \neq \hat{\Theta}_i$. Thus the normalised minimum SED of a diverge in the $i$-th interval is

$$\begin{aligned} d^2_{\text{min,diverge}} &= \log_2 M \min_{\Psi_i} \left[1 - \frac{\sin 2\pi h \Psi_i}{2\pi h \Psi_i}\right] \\ &= \log_2 M \left(1 - \max_{\Psi_i} \left[\frac{\sin 2\pi h \Psi_i}{2\pi h \Psi_i}\right]\right), \qquad \Psi_i \in \{\pm 1, \ldots, \pm(P-1)\}. \end{aligned} \tag{3.84}$$

Theorem 1 of [Rim91] states that for any non-zero integer $k$, and for $0 < h < 1/2$,

$$\frac{\sin 2\pi hk}{2\pi hk} \leq \frac{\sin 2\pi h}{2\pi h} \tag{3.85}$$

with equality if and only if $k = \pm 1$. Applying (3.85) to (3.84) we obtain

$$d^2_{\text{min,diverge}} = \log_2 M \left[1 - \frac{\sin 2\pi h}{2\pi h}\right]. \tag{3.86}$$

Thus a diverge occurring in the $i$-th interval achieves minimum SED if $\Psi_i = \pm 1$.

For the case where $M = k_M P$, with $k_M$ being an integer greater than one, $B = M$ and (3.81) does not hold. However from (A.1), we can write $\sum_{i=0}^{n} U_i$ as

$$\begin{aligned} \sum_{i=0}^{n} U_i &= R_P \left[\sum_{i=0}^{n} U_i\right] + kP \\ &= \Theta_n + kP \end{aligned} \tag{3.87}$$

where $k$ is a non-negative integer. Using (3.87) in the definition of $\beta_n$ with $B = M$, we have

$$\begin{aligned} \beta_n &= R_M \left[\sum_{i=0}^{n} U_i\right] \\ &= R_M[\Theta_n + kP] \\ &= R_M[\Theta_n + R_M[kP]] \\ &= R_M[\Theta_n + k_{1,n}P] \\ &= \Theta_n + k_{1,n}P \end{aligned} \tag{3.88}$$

where $k_{1,n}$ is a non-negative integer less than $k_M$, and the last step makes use of (A.8) as $\Theta_n + k_{1,n}P$ is non-negative and less than $M$. Similarly, we can show that

$$\hat{\beta}_n = \hat{\Theta}_n + k_{2,n}P \tag{3.89}$$

where $k_{2,n}$ is a non-negative integer less than $k_M$. Using (3.88) and (3.89) in (3.90), we obtain

$$d^2_n(\beta, \hat{\beta}) = \log_2 M \left[1 - \frac{\sin(2\pi h[\Theta_n - \hat{\Theta}_n]) - \sin(2\pi h[\Theta_{n-1} - \hat{\Theta}_{n-1}])}{2\pi h(\Theta_n - \Theta_{n-1} - \hat{\Theta}_n + \hat{\Theta}_{n-1} + k_n P)}\right], \tag{3.90}$$

where

$$k_n \;=\; k_{1,n} - k_{1,n-1} - k_{2,n} + k_{2,n-1}, \tag{3.91}$$

and $k_{1,n}, k_{1,n-1}, k_{2,n}, k_{2,n-1} \in \{0, 1, \ldots, k_M - 1\}$. Note that $k_n \in \{0, \pm 1, \ldots, \pm 2(k_M - 1)\}$. Substituting (3.79) and (3.80) into (3.90), we find the NSED of a diverge is

$$d^2_{i,\text{diverge}}(\beta, \hat{\beta}) \;=\; \log_2 M \left[1 - \frac{\sin(2\pi h[\Theta_i - \hat{\Theta}_i])}{2\pi h(\Theta_i - \hat{\Theta}_i + k_i P)}\right]. \tag{3.92}$$

We note now that we can replace $\Theta_i - \hat{\Theta}_i$ with $\Theta_i - \hat{\Theta}_i + k_i P$ in the argument of the sine function as

$$\begin{aligned}
\sin(2\pi h[\Theta_i - \hat{\Theta}_i + k_i P]) \;&=\; \sin(R_{2\pi}\Big[2\pi h[\Theta_i - \hat{\Theta}_i + k_i P]\Big]) \\
&=\; \sin\left(R_{2\pi}\left[2\pi\left(\frac{K}{P}\right) R_P\Big[\Theta_i - \hat{\Theta}_i + k_i P\Big]\right]\right) \\
&=\; \sin\left(R_{2\pi}\left[2\pi\left(\frac{K}{P}\right) R_P\Big[\Theta_i - \hat{\Theta}_i + R_P[k_i P]\Big]\right]\right) \\
&=\; \sin\left(R_{2\pi}\left[2\pi\left(\frac{K}{P}\right) R_P\Big[\Theta_i - \hat{\Theta}_i\Big]\right]\right) \\
&=\; \sin(2\pi h[\Theta_i - \hat{\Theta}_i]).
\end{aligned} \tag{3.93}$$

Using (3.93) in (3.92) we have

$$\begin{aligned}
d^2_{i,\text{diverge}}(\beta, \hat{\beta}) \;&=\; \log_2 M \left[1 - \frac{\sin(2\pi h[\Theta_i - \hat{\Theta}_i + k_i P])}{2\pi h(\Theta_i - \hat{\Theta}_i + k_i P)}\right] \\
&=\; \log_2 M \left[1 - \frac{\sin(2\pi h\tilde{\Psi}_i)}{2\pi h\tilde{\Psi}_i}\right]
\end{aligned} \tag{3.94}$$

where we have defined

$$\tilde{\Psi}_i \;\triangleq\; \Theta_i - \hat{\Theta}_i + k_i P. \tag{3.95}$$

Note that $\tilde{\Psi}_i \in \{\pm 1, \pm 2, \ldots, \pm([2k_M - 1]P - 1)\}$ as $\Theta_i \neq \hat{\Theta}_i$. Equation (3.94) is in the same form as (3.82), so we can apply similar reasoning to find the NMSED of a diverge in the $i$-th interval is

$$\begin{aligned}
d^2_{\text{min,diverge}} \;&=\; \log_2 M \min_{\tilde{\Psi}_i}\left[1 - \frac{\sin(2\pi h\tilde{\Psi}_i)}{2\pi h\tilde{\Psi}_i}\right], \quad \tilde{\Psi}_i \in \{\pm 1, \pm 2, \ldots, \pm([2k_M - 1]P - 1)\} \\
&=\; \log_2 M \left[1 - \frac{\sin 2\pi h}{2\pi h}\right],
\end{aligned} \tag{3.96}$$

and attained when $\tilde{\Psi}_i = \pm 1$.

We now turn to a merge, occurring in any interval, say the $l$-th interval. Our detour will be returning from an error state to an error-free state. Thus we will have

$$\Theta_{l-1} \;\neq\; \hat{\Theta}_{l-1} \tag{3.97}$$

and

$$\Theta_l = \hat{\Theta}_l \tag{3.98}$$

which imply $U_l \neq \hat{U}_l$. Again we consider the cases $M \leq P$ and $M > P$ separately. We first look at the case where $M \leq P$, so that $B = P$ and (3.81) holds. Substituting (3.97) and (3.98) into (3.81), we find the NSED of a merge is

$$\begin{aligned} d^2_{l,\text{merge}}(U_l, \hat{U}_l) &= \log_2 M \left[1 - \frac{\sin(2\pi h[\Theta_{l-1} - \hat{\Theta}_{l-1}])}{2\pi h(\Theta_{l-1} - \hat{\Theta}_{l-1})}\right] \\ &= \log_2 M \left[1 - \frac{\sin 2\pi h \Psi_{l-1}}{2\pi h \Psi_{l-1}}\right], \end{aligned} \tag{3.99}$$

Note that $\Psi_{l-1} \in \{\pm 1, \pm 2, \ldots, \pm(P-1)\}$ as $\Theta_{l-1} \neq \hat{\Theta}_{l-1}$. As (3.99) is in a similar form to (3.82), we can apply similar reasoning to obtain the normalised minimum SED of a merge in the $l$-th interval as

$$\begin{aligned} d^2_{\text{min,merge}} &= \log_2 M \min_{\Psi_{l-1}} \left[1 - \frac{\sin 2\pi h \Psi_{l-1}}{2\pi h \Psi_{l-1}}\right], \qquad \Psi_{l-1} \in \{\pm 1, \pm 2, \ldots, \pm(P-1)\} \\ &= \log_2 M \left[1 - \frac{\sin 2\pi h}{2\pi h}\right]. \end{aligned} \tag{3.100}$$

Thus a merge occurring in the $l$-th interval achieves minimum SED if $\Psi_{l-1} = \pm 1$.

For the $M > P$ case, $B = M$ so we must use (3.90). Substituting (3.97) and (3.98) into (3.81), we find the NSED of a merge occurring in the $l$-th interval is

$$d^2_{n,\text{merge}}(\beta, \hat{\beta}) = \log_2 M \left[1 - \frac{\sin(2\pi h[\Theta_{l-1} - \hat{\Theta}_{l-1}])}{2\pi h(\Theta_{l-1} - \hat{\Theta}_{l-1} + k_{l-1}P)}\right]. \tag{3.101}$$

Again we replace $\Theta_{l-1} - \hat{\Theta}_{l-1}$ with $\Theta_{l-1} - \hat{\Theta}_{l-1} + k_{l-1}P$ in the argument of the sine function to obtain

$$\begin{aligned} d^2_{n,\text{merge}}(\beta, \hat{\beta}) &= \log_2 M \left[1 - \frac{\sin(2\pi h[\Theta_{l-1} - \hat{\Theta}_{l-1} + k_{l-1}P])}{2\pi h(\Theta_{l-1} - \hat{\Theta}_{l-1} + k_{l-1}P)}\right] \\ &= \log_2 M \left[1 - \frac{\sin(2\pi h \tilde{\Psi}_{l-1})}{2\pi h \tilde{\Psi}_{l-1}}\right], \end{aligned} \tag{3.102}$$

where $\tilde{\Psi}_{l-1} \in \{\pm 1, \pm 2, \ldots, \pm([2k_M - 1]P - 1)\}$ as $\Theta_{l-1} \neq \hat{\Theta}_{l-1}$. As (3.102) is of the same form as (3.82), we can again apply similar reasoning to find the NMSED of a merge in the $l$-th interval is

$$\begin{aligned} d^2_{\text{min,merge}} &= \log_2 M \min_{\tilde{\Psi}_{l-1}} \left[1 - \frac{\sin(2\pi h \tilde{\Psi}_{l-1})}{2\pi h \tilde{\Psi}_{l-1}}\right], \ \tilde{\Psi}_{l-1} \in \{\pm 1, \pm 2, \ldots, \pm([2k_M - 1]P - 1)\} \\ &= \log_2 M \left[1 - \frac{\sin 2\pi h}{2\pi h}\right], \end{aligned} \tag{3.103}$$

and attained when $\tilde{\Psi}_{l-1} = \pm 1$.

So we have found the NMSED of a diverge and a merge for the cases where $P \leq M$ and $P = M$. Let us now consider detours of length 2, which consist only a diverge and merge. If there exists a path (or path) of length 2 whose diverge and merge achieve $d^2_{\text{min,diverge}}$ and $d^2_{\text{min,merge}}$ respectively, then this path would clearly achieve $d^2_{\text{min},2}$, the minimum distance for any path of 2 intervals. As $d^2_n(\beta, \hat{\beta})$ is non-negative and zero on the error-free paths only, the SED of any detour of greater than two unmerged intervals will be larger than $d^2_{\text{min},2}$.

Thus our task is to show that $d^2_{\text{min},2} = d^2_{\text{min,diverge}} + d^2_{\text{min,merge}}$ can be achieved by one or more paths. Such a path will start on an error-free path, diverge in the $i$-th interval to a state such that $\Psi_i = \pm 1$ (or $\tilde{\Psi}_i = \pm 1$) and then merge to an error-free path. The following equations describe this sequence,

$$\Theta_{i-1} = \hat{\Theta}_{i-1} \tag{3.104}$$

$$\Theta_i = \hat{\Theta}_i \pm 1 \tag{3.105}$$

$$\Theta_{i+1} = \hat{\Theta}_{i+1}. \tag{3.106}$$

At the start of the detour, $\Theta_{i-1}$ and $\hat{\Theta}_{i-1}$ are both equal to a certain value. We will call this value $\Omega$, where $\Omega \in \{0, 1, \ldots, P-2\}$. With the inputs

$$U_i = 1 \qquad \hat{U}_i = 0 \tag{3.107}$$

$$U_{i+1} = 0 \qquad \hat{U}_{i+1} = 1, \tag{3.108}$$

we obtain the sequences

$$\begin{array}{llll} \Theta_{i-1} = \Omega & \hat{\Theta}_{i-1} = \Omega & \Psi_{i-1} = 0 & \tilde{\Psi}_{i-1} = 0 \\ \Theta_i = \Omega + 1 & \hat{\Theta}_i = \Omega & \Psi_i = 1 & \tilde{\Psi}_i = 1 \\ \Theta_{i+1} = \Omega + 1 & \hat{\Theta}_{i+1} = \Omega + 1 & \Psi_{i+1} = 0 & \tilde{\Psi}_{i+1} = 0. \end{array}$$

So we have shown that $\Psi_n$ and $\tilde{\Psi}_n$ can indeed follow the sequence $(\ldots, 0, 1, 0, \ldots)$, and thus

$$\begin{aligned} d^2_{\text{min},2} &= d^2_{\text{min,diverge}} + d^2_{\text{min,merge}} \\ &= 2 \log_2 M \left[ 1 - \frac{\sin 2\pi h}{2\pi h} \right]. \end{aligned} \tag{3.109}$$

Note that although we have only shown above that there are $P - 1$ paths that achieve $d^2_{\text{min},2}$, they are by no means the only paths to do this. Indeed, if we reverse the inputs

in (3.107) and (3.108) we find another $P-1$ paths, and more can be found using other inputs. We are now in a position to present the generalised result.

The minimum normalised squared Euclidean distance (NMSED) of a DCPFSK system with $h = K/P \leq 1/2$, where $K$ and $P$ are relatively-prime positive integers, and whose choice of $M$ and $P$ follow $\max\{M,P\} = k\min\{M,P\}$, where $k$ is a positive integer, is given by

$$d^2_{\min} = \begin{cases} \min\left[\log_2 M,\ 2\log_2 M\left(1-\dfrac{\sin 2\pi h}{2\pi h}\right)\right] & M > P \\ 2\log_2 M\left(1-\dfrac{\sin 2\pi h}{2\pi h}\right) & M \leq P \end{cases} \tag{3.110}$$

It is interesting to note that this is the same as the result for CPFSK.

## 3.10 Error Performance of DCPFSK

The differential demodulator uses the previous symbol which is corrupted by AWGN to demodulate the current symbol. The presence of this "extra" AWGN means that $y_{\rm d}(t,\beta)$ has a non-zero mean, quadratic Gaussian form which makes exact analysis very difficult. Let us look at $y_{\rm d}(t,\beta)$ again,

$$\begin{aligned} y_{\rm d}(t,\beta) = \frac{1}{2}\sqrt{\frac{T}{E}}\Big[&\tilde{s}(t,\beta)\,\tilde{s}^*(t-T,\beta) + \tilde{w}(t)\,\tilde{s}^*(t-T,\beta) \\ &+ \tilde{s}(t,\beta)\,\tilde{w}^*(t-T) + \tilde{w}(t)\,\tilde{w}^*(t-T)\Big]. \end{aligned} \tag{3.111}$$

We assume that at high SNR the noise $\times$ noise term is dominated by the other three terms, and we neglect it. We further assume that the two noise $\times$ signal terms are independent, and that they act so that the performance of DCPFSK is equivalent to that of CPFSK with twice the noise. Thus we assume that the probability of bit error for DCPFSK at high SNR is

$$P_e \approx Q\left(\sqrt{\frac{d^2_{\min}E_b}{2N_0}}\right). \tag{3.112}$$

Similar assumptions and conclusions can be found in [AS81] for DCPFSK and [Pro95] for differential PSK (DPSK). For the simplest DCPFSK case, that of DMSK, there is a solution that avoids the above assumptions. The probability of bit error for DMSK is given by (see for example [Mas90])

$$P_{e,\rm DMSK} = \frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right), \tag{3.113}$$

which is exactly the same as the result for differential binary PSK [Hay88][Pro95].

We present the $d^2_{\min}$ for various DCPFSK schemes in Table 3.1, calculated using (3.110). As (3.110) is identical to (2.61), the values in Table 3.1 are the same as those in Table 2.1.

**Table 3.1**

$d^2_{\min}$ for various DCPFSK schemes

| Scheme | $d^2_{\min}$ | |
|---|---|---|
| DMSK | 2 | 2 |
| 4-DCPFSK | $4 - \frac{8}{\pi}$ | 1.45 |
| 8-DCPFSK | $6 - \frac{12\sqrt{2}}{\pi}$ | 0.60 |

## 3.11 Simulation of a DCPFSK system

The simulation of a DCPFSK system is a much more complicated problem than that of its coherent counterpart. We discuss the simulations and the model used in Appendix F. It is important to note that the simulations do not use the assumptions discussed in Section 3.10, and thus they take into account the effect of noise $\times$ noise and signal $\times$ noise terms.

Figure 3.12 shows the results of the simulations compared with the computed performance of various coherently-demodulated CPFSK schemes. The computed curves were calculated using (3.113), (3.112) and the results in Table 2.1. The simulation results and the computed curves agree reasonably well at high SNR, given that the computed curves are only approximations based on significant assumptions. The data was assumed to be uniformally distributed, at least 16 samples per symbol were generated, and the Viterbi algorithm had a decision depth of 20 symbols.

It is interesting to compare the performance of CPFSK with DCPFSK, and in Table 3.2 we tabulate the approximate SNR values required in the simulations to achieve a BER of $10^{-5}$ for different $M$-CPFSK and $M$-DCPFSK schemes. For the cases with $M$ equal to 4 and 8, an increase of approximately 3dB is required when going from CPFSK to DCPFSK. An increase of less than a dB is required when going from MSK to DMSK. Similar figures have been observed in [Hay88] in comparisons of $M$-ary PSK and $M$-ary DPSK.

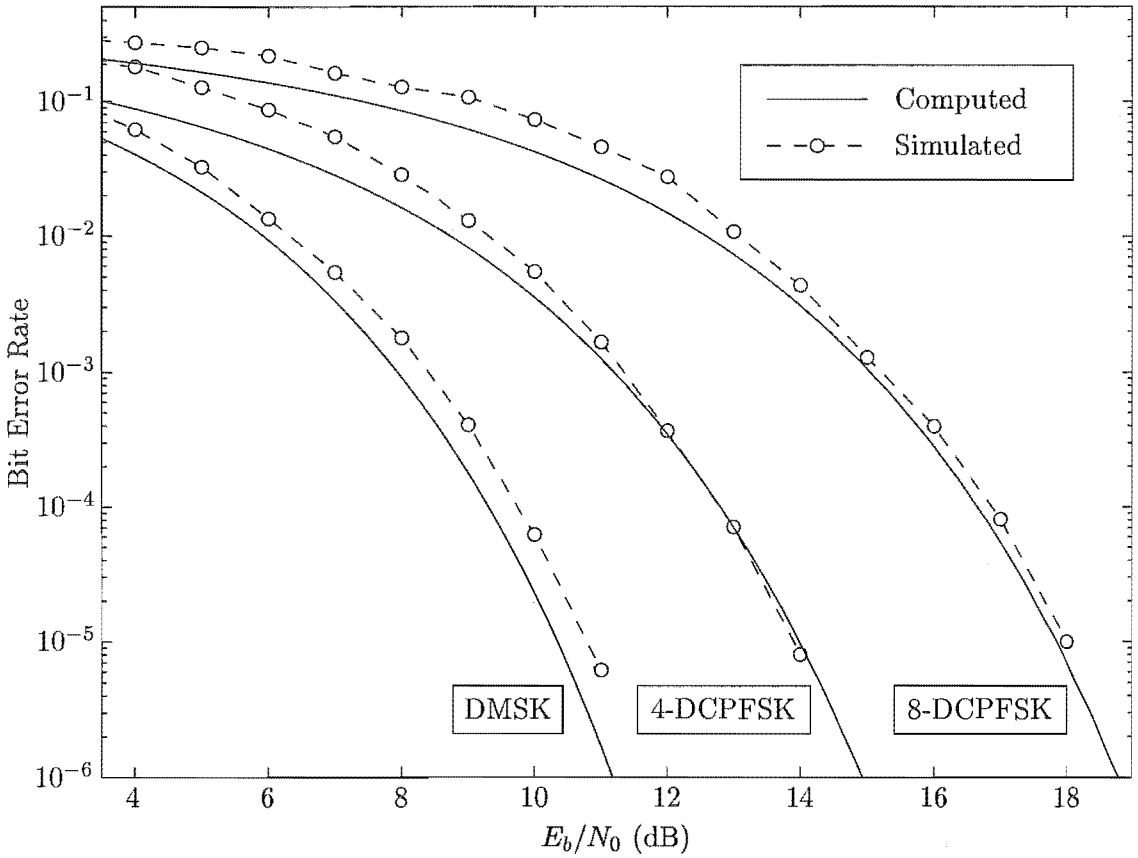



**Figure 3.12:** Computed and simulated performance of various $M$-DCPFSK schemes

**Table 3.2**

Comparison of empirical SNR values required to achieve a BER of $10^{-5}$ for different $M$-CPFSK and $M$-DCPFSK schemes

| | Approximate SNR values | |
|---|---|---|
| $M$ | $M$-CPFSK | $M$-DCPFSK |
| 2 | 9.9 | 10.6 |
| 4 | 11.0 | 14.0 |
| 8 | 14.8 | 17.9 |

## 3.12 Summary

In this chapter we have presented and discussed DCPFSK. The differential encoder developed preserves the phase trellis through differential demodulation and combines well with the decomposed model of CPFSK. The differential demodulator operates in such a manner that the phase tree and trellis of a demodulated DCPFSK signal differs from those of a modulated DCPFSK signal. Both phase trees and trellises are important as

they provide important insight into the behaviour of DCPFSK.

We presented a receiver structure for DCPFSK that was based on Euclidean distance and discussed each of its components. We also calculated the normalised incremental squared Euclidean distance (NISED) and normalised minimum squared Euclidean distance (NMSED) for DCPFSK. The NISED of DCPFSK differs from that of CPFSK, but both have the same NMSED in the case with no external error-control coding. Our receiver is not maximum-likelihood due to the extra noise components introduced by differential demodulation. These noise components make analysis very difficult, but through some simple assumptions the error performance of DCPFSK is found to be approximately 3dB worse than that of a CPFSK system with the same NMSED. The results of our simulations were seen to agree with this approximation.

# Chapter 4

# Coding DCPFSK

## 4.1 Introduction

Trellis coding has long been used to improve a system's performance without increasing its bandwidth. CPFSK has previously been encoded using a binary convolutional channel encoder, but work in [MM89] suggests that the most "natural" codes for $M$-ary phase modulation are linear codes over the ring of integers modulo-$M$ ($\mathbb{Z}_M$). This idea has been combined with the decomposed model of CPFSK [Rim88], using convolutional encoders over the ring of integers modulo-$P$ $\mathbb{Z}_P$ [YT94] [RL95]. This allowed the structure within CPFSK to be exploited to improve the code performance while minimally increasing the complexity. The codes found performed better than any previous coding schemes used.

As discussed in Section 3.9, the normalised incremental squared Euclidean distance (NISED) of DCPFSK is different to that of CPFSK. Although CPFSK and DCPFSK have the same normalised minimum squared Euclidean distance (NMSED) in the uncoded case, it does not necessarily follow that the coded systems will have the same NMSED. We wish to use the decomposed model of DCPFSK developed in Chapter 3 to develop convolutional codes over the ring of integers modulo-$M$ $\mathbb{Z}_M$ for $M$-DCPFSK in a similar manner to [YT94]. We restrict ourselves to $M$-DCPFSK coded systems. The differential encoders and continuous phase encoders (CPE) in these systems will both be linear encoders over $\mathbb{Z}_M$, allowing for easier interfacing with a channel encoder.

We begin this chapter in Section 4.2 by reviewing the method used in [YT94] to code CPFSK. We then discuss in Section 4.3 why a feedback-free CPE is necessary when performing code searches for CPFSK systems. In Section 4.4 we present the appropriate terminology and the structure of the channel encoder in [YT94] that we will be using. This allows us to present the code search model for an encoded CPFSK system in Section 4.5.

The overall encoder in an encoded CPFSK system is discussed in Section 4.6, and the implementation of such a system is discussed in Section 4.7. We then move on to coded DCPFSK systems. We first develop a Euclidean distance model of DCPFSK in Section 4.8, which allows us to construct a code search model for an encoded DCPFSK system in Section 4.9. We discuss the overall encoder in an encoded DCPFSK system in Section 4.10, and in Section 4.11 we discuss the implementation of such a system. Having presented and developed the necessary material, we discuss the code search procedure used in this thesis in Section 4.12, and present its results in Section 4.13. The important points from the chapter are summarised in Section 4.14.

## 4.2 Coding CPFSK

Figure 4.1 shows the general idea of a coded $M$-CPFSK system [YT94]. The channel encoder (CE) accepts $l-1$ $M$-ary data symbols and outputs $l$ $M$-ary encoded symbols. The commutator then allows the continuous phase encoder (CPE) and memoryless modulator (MM) to process these encoded symbols individually to generate the transmitted signal. The channel encoder and CPE form an overall encoder. If, as we specify, the channel encoder and CPE are both linear encoders over $\mathbb{Z}_M$, the overall encoder is also a linear encoder over $\mathbb{Z}_M$.

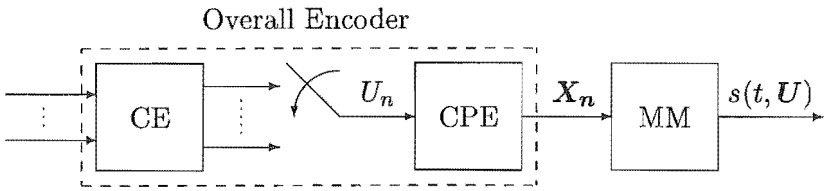



**Figure 4.1:** Coded CPFSK system

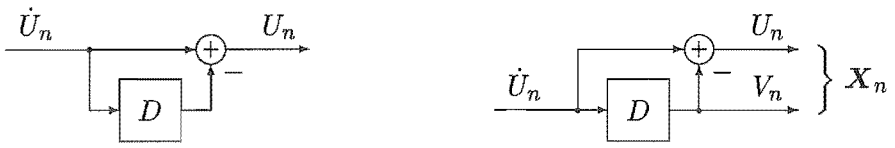
## 4.3 Feedback-free Continuous Phase Encoder

The CPE of Section 2.5.2 is a feedback encoder, and catastrophic convolutional channel encoders may combine with the feedback-CPE to produce an overall encoder that is non-catastrophic [MMP88]. This means that catastrophic channel encoders should also be considered when finding the best codes for CPFSK. The problem of determining which catastrophic channel encoders combine to produce overall non-catastrophic encoders would also need to be solved. By using a feedback-free CPE these problems can be avoided, as

non-catastrophic channel encoders combined with a feedback-free CPE always produce overall non-catastrophic encoders, and catastrophic channel encoders combined with a feedback-free CPE always produce overall catastrophic encoders [YT94].

The feedback in the CPE can be removed by using the precoder shown in Figure 4.2(a), where $\dot{U}$ is some data sequence that will produce $U$ when passed through the precoder. Like the differential encoder, this precoder is also a scrambler [For70], as it replaces one stream of $M$-ary symbols with another stream of $M$-ary symbols in a one-to-one permutation. We will henceforth refer to this precoder as the scrambler.

The feedback-free CPE resulting from the cascade of the scrambler and the feedback CPE is illustrated in Figure 4.2(b), and will combine with a non-catastrophic channel encoder to form an overall non-catastrophic encoder.




(a) Scrambler (b) Feedback-free CPE

**Figure 4.2:** Removing the feedback in the CPE

## 4.4 Channel Encoder

The channel encoder employed in [YT94] was a convolutional encoder defined over $\mathbb{Z}_P$, but we will deal with one defined over $\mathbb{Z}_M$. Let us introduce the terminology for a *delay polynomial* $x(D)$ where

$$x(D) \triangleq x_0 + x_1 D + \cdots + x_{k_V} D^{k_V} + \ldots, \tag{4.1}$$

and $D$ is a unit delay. Let $R = \mathbb{Z}_M$ denote a commutative ring [MM90], and let $R(D)$ denote the ring of rational functions over $R$, whose numerators are the polynomials with coefficients in $\mathbb{Z}_M$, as are the denominators, with the restriction that their zeroth delay terms are equal to one. Every rate-$(l-1)/l$ convolutional code over $R$ can be generated by some encoding matrix $\mathbf{G}(D) = g_{ik}(D)$, $1 \leqslant i < l-1$, $1 \leqslant k < l$, $g_{ik}(D) \in R(D)$ with kernel $\mathrm{Ker}[\mathbf{G}(D)] = 0$. Every generator matrix whose components are causal, can be realised with a finite number of memory cells capable of storing ring elements and a finite number of modulo-$M$ scalars and adders. The uncoded input sequence is denoted $\boldsymbol{a}(D)$,

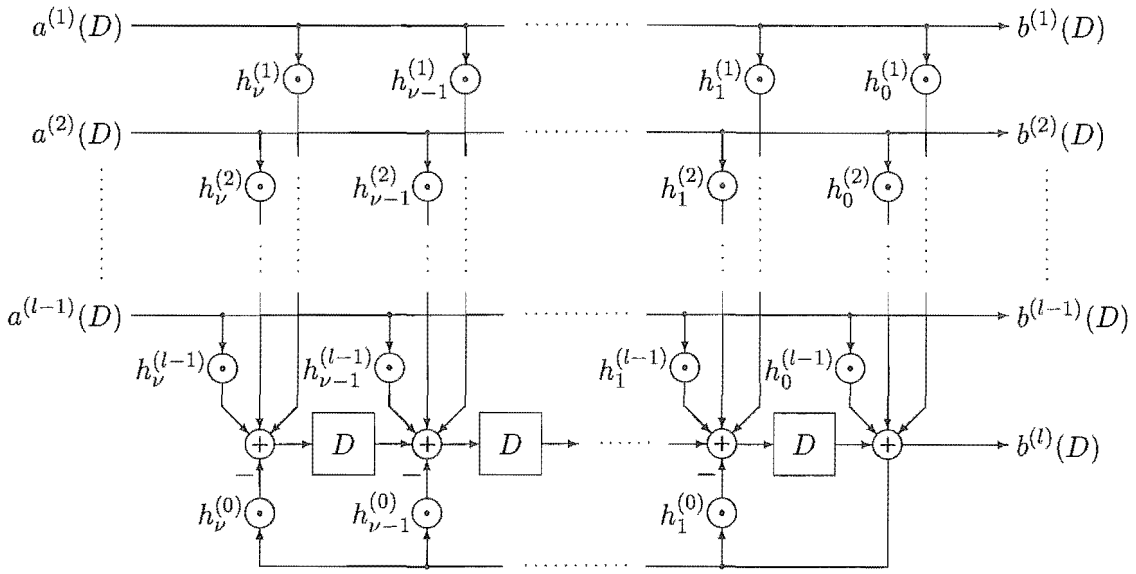



**Figure 4.3:** Rate-$(l-1)/l$ systematic ring convolutional encoder

where each $\boldsymbol{a}_{k_V}$ is a vector of $l-1$ $M$-ary integers, that is,

$$\boldsymbol{a}_{k_V} = \left[a_{k_V}^{(1)} \quad a_{k_V}^{(2)} \quad \dots \quad a_{k_V}^{(l-1)}\right]^\dagger, \qquad a_{k_V}^{(i)} \in \{0, 1, \dots, (M-1)\}, \tag{4.2}$$

where $\boldsymbol{x}^\dagger$ represents the non-conjugate transpose of $\boldsymbol{x}$. The coded output sequence $\boldsymbol{b}(D)$, where each $\boldsymbol{b}_{k_V}$ is a vector of $l$ $M$-ary symbols similar to $\boldsymbol{a}_{k_V}$, is related to $\boldsymbol{a}(D)$ by

$$\boldsymbol{b}(D)^\dagger = \boldsymbol{a}(D)^\dagger \mathbf{G}(D). \tag{4.3}$$

As in the field case, a systematic ring convolutional encoder is always minimal and non-catastrophic [MM89]. That is, there is no infinite weight input sequence that produces a finite weight output sequence.

The feedback realisation of a rate-$(l-1)/l$ systematic convolutional encoder is shown in Figure 4.3. The $\odot$ and $\oplus$ symbols represent modulo-$M$ multiplication and addition respectively. The transfer function for such an encoder is

$$\mathbf{G}(D) = \begin{bmatrix} 1 & \cdots & 0 & \dfrac{H^{(1)}(D)}{H^{(0)}(D)} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & \dfrac{H^{(l-1)}(D)}{H^{(0)}(D)} \end{bmatrix}, \tag{4.4}$$

where $H^{(i)}(D) = h_0^{(i)} + h_1^{(i)} D + \cdots + h_\nu^{(i)} D^\nu$ and $\nu$ is the number of delay cells in the encoder. Note that the element $h_0^{(0)}$ is always equal to one.

Permutations between the redundancy-carrying output line and the information-carrying output lines are necessary, as the different permutations may produce differ-

ent overall encoders when combined with CPFSK [YT94]. A permutation between the redundancy-carrying output line and the $i$-th information-carrying output line corresponds to a permutation of the $l$-th column and the $i$-th column of $\mathbf{G}(D)$. Permutations of the information lines are not necessary, as we assume the input data is random. For a rate-$(l-1)/l$ encoder there are $l$ permutations and thus $l$ different encoders.

## 4.5 Code Search Model for CPFSK

As discussed in Section 2.9, the NMSED of a coherent CPFSK system significantly affects its performance. A channel encoder seeks to increase the NMSED of the system. Thus for a given throughput and complexity, coded CPFSK systems are compared the basis of their NMSED. A code search model should allow the NMSED of the system to be easily determined. In Section 2.8.1 we presented a formula for the NISED of a CPFSK system which was based only on $\boldsymbol{X}_n$, the input to the memoryless modulator at time $n$. By determining the transfer function of the overall encoder formed by the cascade of the channel encoder and the feedback-free CPE, we can calculate the NISED of the coded system using $\boldsymbol{X}_n$.

The transfer function of the feedback CPE is

$$\mathbf{C}(D) = \left[1 \quad \frac{D}{1-D}\right], \tag{4.5}$$

and that of the scrambler

$$\mathbf{T}(D) = \left[1-D\right]. \tag{4.6}$$

The transfer function of the feedback-free CPE is obtained by cascading $\mathbf{T}(D)$ and $\mathbf{C}(D)$,

$$\dot{\mathbf{C}}(D) = \mathbf{T}(D)\cdot\mathbf{C}(D) = \left[1-D \quad D\right]. \tag{4.7}$$

Figure 4.4 illustrates the feedback-free CPFSK system encoded with a convolutional channel encoder over $\mathbb{Z}_M$. The input data sequence $\boldsymbol{a}(D)$ is encoded by $\mathbf{G}(D)$ and the $l$ elements of the output encoded sequence $\boldsymbol{b}(D)$ are then individually selected by the commutator and processed by the feedback-free CPE $\dot{\mathbf{C}}(D)$ to produce the input to the memoryless modulator $\boldsymbol{X}_n$, from which the NMSED of the channel encoder can be calculated.

It is convenient to move the commutator through the system to just in front of the memoryless modulator. Thus the scrambler and the CPE must process $l$ symbols at a time. Let $\mathbf{T}_l(D)$ and $\mathbf{C}_l(D)$ denote the equivalent $l\times l$ and $l\times 2l$ versions of $\mathbf{T}(D)$ and $\mathbf{C}(D)$ respectively. The equivalent $l\times 2l$-version of $\dot{\mathbf{C}}(D)$ is denoted $\dot{\mathbf{C}}_l(D)$ and its output

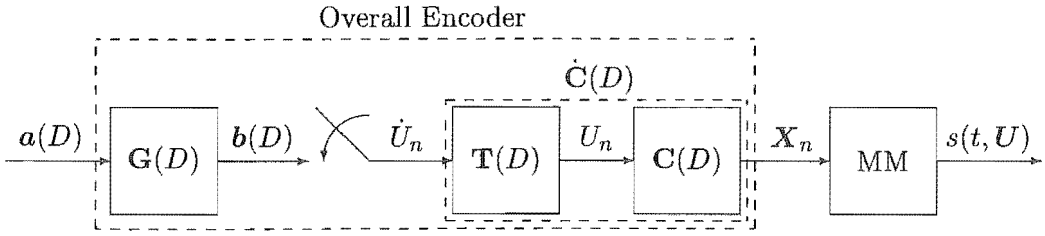



**Figure 4.4:** Coded CPFSK system with scrambler

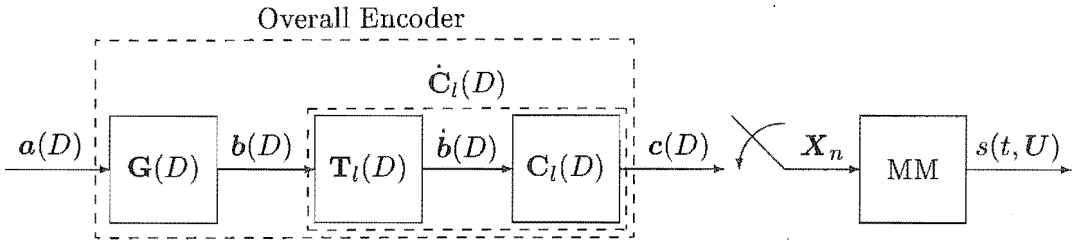



**Figure 4.5:** Coded CPFSK system model for code searches

is denoted $\boldsymbol{c}(D)$ where each $\boldsymbol{c}_{k_V}$ is a vector of $2l$ $M$-ary symbols, similar to $\boldsymbol{a}_{k_V}$. Figure 4.5 shows the system using these equivalent encoders. Note that $\boldsymbol{X}_n$ is a 2-element vector, and the commutator selects all the elements of each $\boldsymbol{c}_{k_V}$ in pairs, that is,

$$\boldsymbol{X}_n = \begin{bmatrix} c_{k_V}^{(2i-1)} \\ c_{k_V}^{(2i)} \end{bmatrix}, \qquad i = 1, 2, \ldots, l, \qquad k_V l \leqslant n < (k_V + 1)l. \tag{4.8}$$

Using the model in Figure 4.5 the NISED between two encoded sequences $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$ over the $k_V$-th trellis interval, denoted $\delta_{k_V}^2(\boldsymbol{c}, \hat{\boldsymbol{c}})$, is defined as

$$\delta_{k_V}^2(\boldsymbol{c}, \hat{\boldsymbol{c}}) \triangleq \sum_{i=1}^{l} d_i^2 \left( \begin{bmatrix} c_{k_V}^{(2i-1)} \\ c_{k_V}^{(2i)} \end{bmatrix}, \begin{bmatrix} \hat{c}_{k_V}^{(2i-1)} \\ \hat{c}_{k_V}^{(2i)} \end{bmatrix} \right), \tag{4.9}$$

where

$$d_i^2 \left( \begin{bmatrix} c_{k_V}^{(2i-1)} \\ c_{k_V}^{(2i)} \end{bmatrix}, \begin{bmatrix} \hat{c}_{k_V}^{(2i-1)} \\ \hat{c}_{k_V}^{(2i)} \end{bmatrix} \right) \triangleq$$

$$\begin{cases} r \log_2 M \left[ 1 - \dfrac{\sin\left(2\pi h \left[c_{k_V}^{(2i-1)} + c_{k_V}^{(2i)} - \hat{c}_{k_V}^{(2i-1)} - \hat{c}_{k_V}^{(2i)}\right]\right) - \sin\left(2\pi h \left[c_{k_V}^{(2i)} - \hat{c}_{k_V}^{(2i)}\right]\right)}{2\pi h \left(c_{k_V}^{(2i-1)} - \hat{c}_{k_V}^{(2i-1)}\right)} \right], & c_{k_V}^{(2i-1)} \neq \hat{c}_{k_V}^{(2i-1)} \\ r \log_2 M \left[ 1 - \cos\left(2\pi h \left[c_{k_V}^{(2i)} - \hat{c}_{k_V}^{(2i)}\right]\right)\right], & c_{k_V}^{(2i-1)} = \hat{c}_{k_V}^{(2i-1)}, \end{cases} \tag{4.10}$$

and $r$ is the rate of the encoder. Using (4.9), we write the normalised squared Euclidean

distance between two coded sequences $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$ as $d^2(\boldsymbol{c}, \hat{\boldsymbol{c}})$, given by

$$d^2(\boldsymbol{c}, \hat{\boldsymbol{c}}) = \sum_{k_V} \delta^2_{k_V}(\boldsymbol{c}, \hat{\boldsymbol{c}}). \tag{4.11}$$

Similarly, the NMSED of a coded CPFSK system is given as

$$d^2_{\min} = \min_{\substack{\boldsymbol{c}, \hat{\boldsymbol{c}} \\ \boldsymbol{c} \neq \hat{\boldsymbol{c}}}} d^2(\boldsymbol{c}, \hat{\boldsymbol{c}}). \tag{4.12}$$

## 4.6 Overall Encoder in a Coded CPFSK System

Like its components—the CPE and the channel encoder—the overall encoder is also a linear encoder over $\mathbb{Z}_M$. Both the $1 \times 2$- and $l \times 2l$-forms of the CPE contain only one delay element, so the overall encoder will contain $\nu + 1$ delay elements. Let $S_G$ be the number of states in the channel encoder. There will be $M$ states in the CPE, but as the overall encoder is not systematic, it will not necessarily have $S_G \cdot M$ states. In [YT94], Yang and Taylor prove that the number of states in the overall encoder, denoted $S_V$, is given by

$$S_V = S_G \cdot n_s, \tag{4.13}$$

where $n_s$ is the number of different values that the $2l$-th element in $\boldsymbol{b}_{k_V}$ takes before the channel encoder merges to the zero state. The quantity $S_V$ will have a large influence on the complexity of an encoded CPFSK system, as the size of the system trellis will depend on the number of states in the overall encoder $S_V$, and the number of branches per state, which is equal to $M^{l-1}$.

### 4.6.1 Rate-1/2 Ring-Coded 4-CPFSK Overall Encoder Example

Let us consider rate-1/2 encoded 4-CPFSK. The channel encoder used is then a convolutional encoder over $\mathbb{Z}_M$, with transfer function

$$\mathbf{G}(D) = \begin{bmatrix} 1 & \dfrac{1}{1+2D} \end{bmatrix}. \tag{4.14}$$

Using the general form of $\mathbf{G}(D)$ given in (4.4) and Figure 4.3, an implementation of (4.14) can be found. This is shown in Figure 4.6, where the summer to left of the delay element has been removed as it only had one input. The minus sign into that summer has also been removed, as $2 = -2$ modulo-4. The trellis diagram of this channel encoder is shown in Figure 4.7, where content of the delay element is shown, along with the input and output

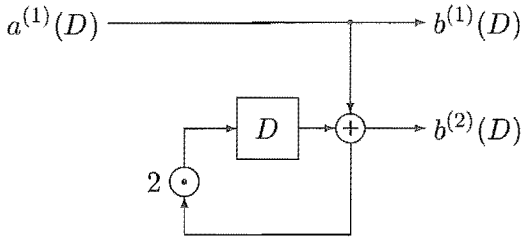



**Figure 4.6:** Implementation of $\mathbf{G}(D) = \left[1 \;\; \frac{1}{1+2D}\right]$

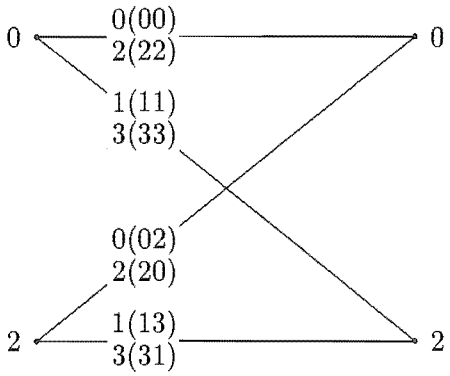



**Figure 4.7:** Trellis diagram of $\mathbf{G}(D) = \left[1 \;\; \frac{1}{1+2D}\right]$

for each transition as $a^{(1)}(b^{(1)}b^{(2)})$. Note that there are parallel transitions between the states.

By cascading the channel encoder with the $2 \times 4$-version of the feedback-free CPE, we obtain an overall encoder that is a rate-1/4 encoder with transfer function

$$\begin{aligned} \mathbf{G}(D)\,\mathbf{C}_2(D) &= \left[1 \;\; \frac{1}{1+2D}\right] \begin{bmatrix} 1 & 0 & 3 & 1 \\ 3D & D & 1 & 0 \end{bmatrix} \\ &= \left[\frac{1+D}{1+2D} \;\; \frac{D}{1+2D} \;\; \frac{3+2D}{1+2D} \;\; 1\right]. \end{aligned} \tag{4.15}$$

This overall encoder is shown in Figure 4.8, and its trellis is illustrated in Figure 4.9. The contents of the two delay cells are used to identify the states, and the input and output for each transition are labelled as $a^{(1)}(c^{(1)}c^{(2)}c^{(3)}c^{(4)})$. If we look at the value of $b^{(2)}$ in each codeword that merges to the zero state in Figure 4.7, we find that there are two distinct values, zero and two, thus $n_s = 2$, and from inspection of Figure 4.7 we find that there two states in the channel encoder so that $S_G = 2$. Substituting these values into (4.13) we note that there should be four states in the overall encoder, and from inspection of Figure 4.9, we find this to be the case. It is important to note that each transition in the trellis of Figure 4.9 represents *two* symbol intervals, as the memoryless modulator processes two elements of $\boldsymbol{c}_{k_V}$ each symbol period.

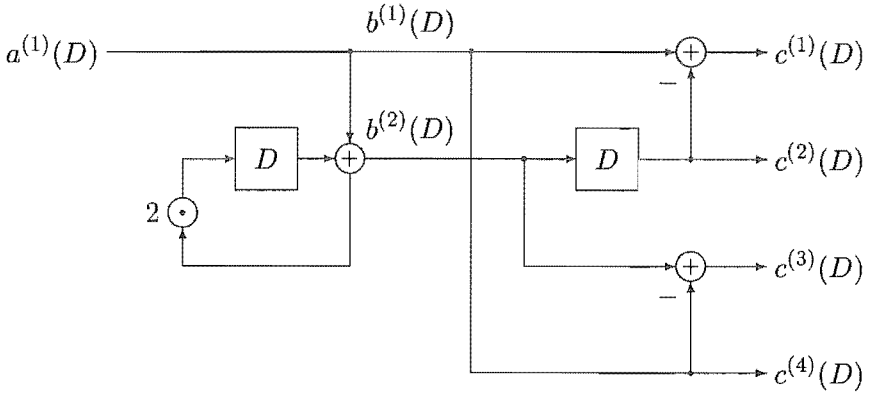



**Figure 4.8:** Implementation of $\mathbf{G}(D)\,\mathbf{C}_2(D) = \left[\frac{1+D}{1+2D} \;\; \frac{D}{1+2D} \;\; \frac{3+2D}{1+2D} \;\; 1\right]$

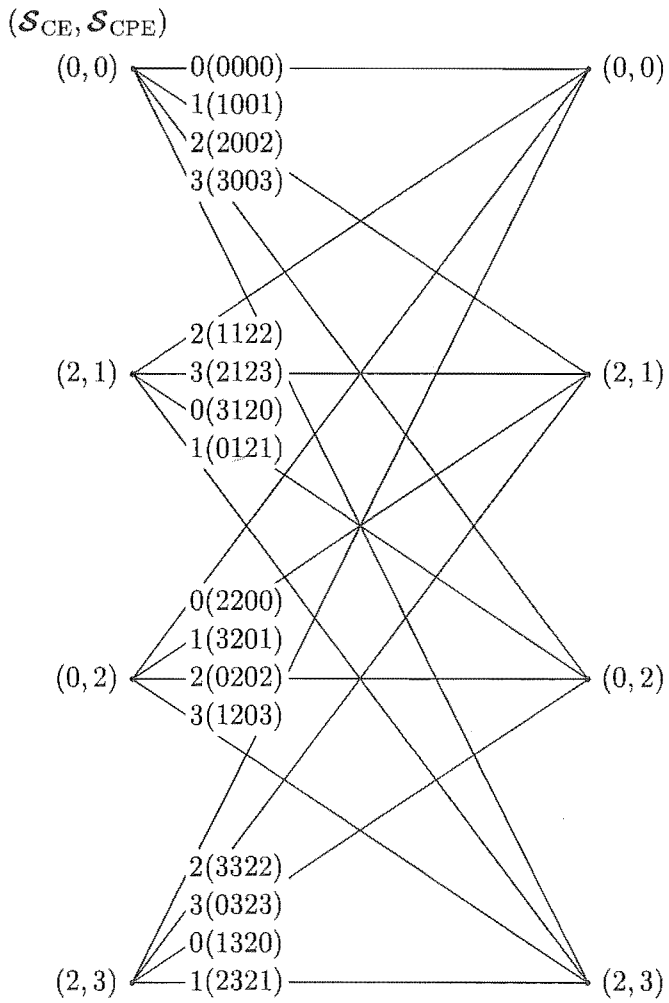



**Figure 4.9:** Trellis diagram of $\mathbf{G}(D)\,\mathbf{C}_2(D) = \left[\frac{1+D}{1+2D} \;\; \frac{D}{1+2D} \;\; \frac{3+2D}{1+2D} \;\; 1\right]$

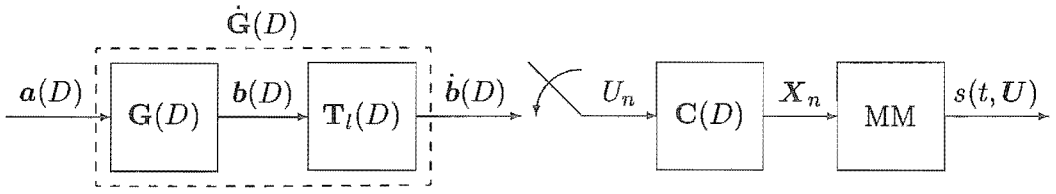



**Figure 4.10:** Actual implementation of a coded CPFSK system

## 4.7 Implementation of a Coded CPFSK System

Although we may use the model in Figure 4.5 for code searches, the actual implementation in a communication system is more like that shown in Figure 4.10. The code chosen to be used in the system is scrambled by $\mathbf{T}_l(D)$ to form $\dot{\mathrm{G}}(D)$, which is then used to encode a standard CPFSK system. Note that the scrambled code is no longer systematic.

### 4.7.1 Rate-1/2 Ring-Coded 4-CPFSK Implementation Example

We use the same channel encoder as that in the example in Section 4.6.1. This is then scrambled by the $2 \times 2$-version of $\mathbf{T}(D)$ to form the transfer function

$$\begin{aligned} \dot{\mathrm{G}}(D) &= \mathrm{G}(D) \cdot \mathbf{T}_2(D) \\ &= \begin{bmatrix} 1 & \dfrac{1}{1+2D} \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 3D & 1 \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{1+D}{1+2D} & \dfrac{3+2D}{1+2D} \end{bmatrix}. \end{aligned} \tag{4.16}$$

This scrambled channel encoder is shown in Figure 4.11, and its trellis is illustrated in Figure 4.12. The contents of the two delay cells are used to identify the states, and the input and output for each transition are labelled as $a^{(1)}(\dot{b}^{(1)}\dot{b}^{(2)})$. The output of the scrambled channel encoder $\dot{b}(D)$ can then be sequentially fed into a standard CPFSK system, to produce a coded CPFSK system with a system trellis which is that shown in Figure 4.9.

## 4.8 Euclidean Distance Model of DCPFSK

As discussed in Section 3.8, we have chosen to use a DCPFSK receiver that seeks to minimise the squared Euclidean distance (SED) between the received signal and the possibly-transmitted signals. We assume that the performance of such a receiver is dominated by its NMSED (see Section 3.10), thus we wish to find codes for DCPFSK with maximum

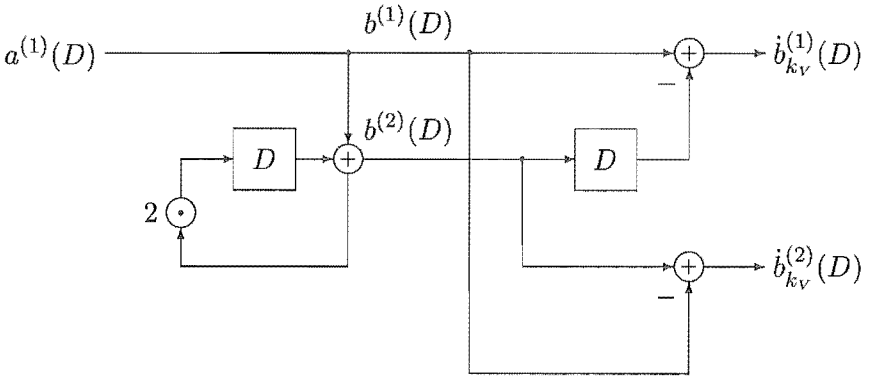



**Figure 4.11:** Implementation of $\dot{\mathbf{G}}(D) = \left[\frac{1+D}{1+2D} \quad \frac{3+2D}{1+2D}\right]$

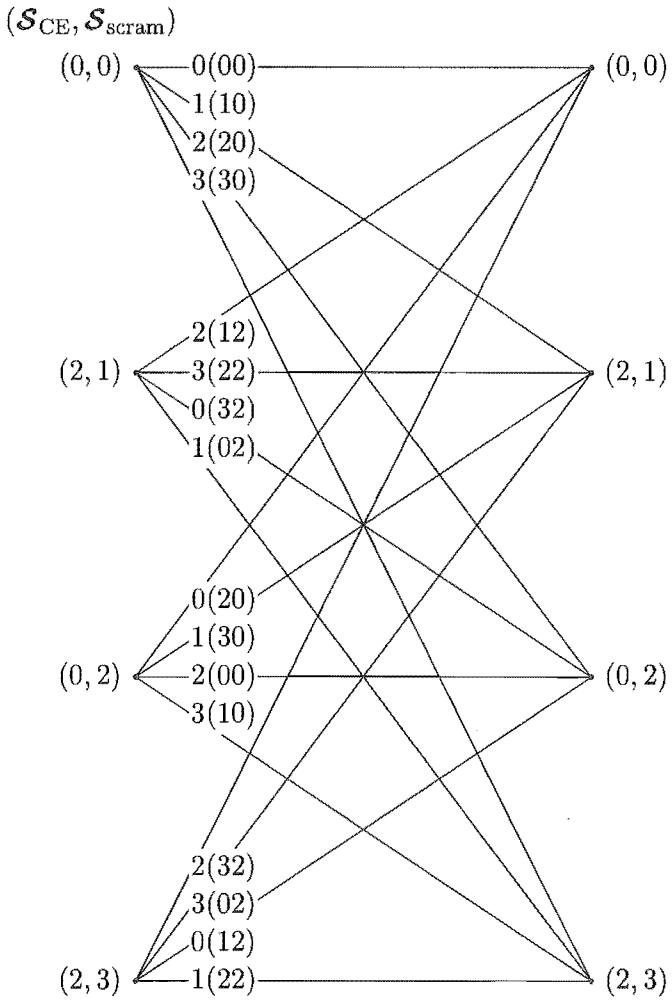



**Figure 4.12:** Trellis diagram of $\dot{\mathbf{G}}(D) = \left[\frac{1+D}{1+2D} \quad \frac{3+2D}{1+2D}\right]$

NMSED. In order to preform code searches, we need to develop a model that allows the SED characteristics of the demodulated DCPFSK signal to be calculated from the input to the memoryless modulator. By comparing the NISED of CPFSK and DCPFSK given by (2.60) and (3.60) respectively, we see that if we introduce a 2×1 vector $\breve{\boldsymbol{X}}_n$, with elements

$$\breve{X}_n^{(1)} \triangleq \beta_n - \beta_{n-1} \qquad \text{and} \qquad \breve{X}_n^{(2)} \triangleq \beta_{n-1}, \tag{4.17}$$

then we can write the NISED of DCPFSK, as given in (3.60), as

$$d_n^2(\breve{\boldsymbol{X}}_n, \hat{\breve{\boldsymbol{X}}}_n) = \begin{cases} \log_2 M \left[1 - \dfrac{\sin\left(2\pi h\left[\breve{X}_n^{(1)} + \breve{X}_n^{(2)} - \hat{\breve{X}}_n^{(1)} - \hat{\breve{X}}_n^{(2)}\right]\right) - \sin\left(2\pi h\left[\breve{X}_n^{(2)} - \hat{\breve{X}}_n^{(2)}\right]\right)}{2\pi h\left(\breve{X}_n^{(1)} - \hat{\breve{X}}_n^{(1)}\right)}\right], & \breve{X}_n^{(1)} \neq \hat{\breve{X}}_n^{(1)} \\ \log_2 M \left[1 - \cos\left(2\pi h\left[\breve{X}_n^{(2)} - \hat{\breve{X}}_n^{(2)}\right]\right)\right], & \breve{X}_n^{(1)} = \hat{\breve{X}}_n^{(1)}. \end{cases} \tag{4.18}$$

A coherently-demodulated CPFSK system with $\breve{\boldsymbol{X}}_n$ as the input to its memoryless modulator will produce signals with a NISED given by (4.18). An encoder to produce $\breve{\boldsymbol{X}}_n$ from $\beta_n$ is shown in Figure 4.13. We call this encoder the differential decoder as it removes the effect of the differential encoder. Note the use of the $\boxplus$ to indicate standard addition and not modulo addition. The transfer function of the differential decoder is

$$\mathbf{R}(D) = \begin{bmatrix} 1 \boxminus D & D \end{bmatrix}. \tag{4.19}$$

The $\boxminus$ is used to denote standard subtraction and not modulo subtraction.

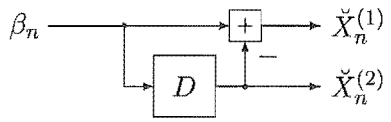



**Figure 4.13:** Differential decoder

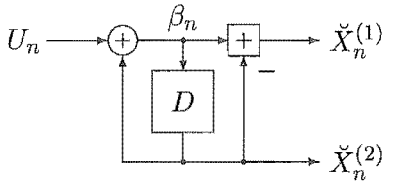



**Figure 4.14:** Differential continuous phase encoder/decoder

The encoder to produce $\breve{\boldsymbol{X}}_n$ from $U_n$ is just the differential encoder of Figure 3.3 cascaded with the differential decoder, and is shown in Figure 4.14. We call this the

differential continuous phase encoder/decoder (DCPED). The transfer function of the differential encoder is

$$\mathbf{E}(D) = \left[\frac{1}{1 \boxminus D}\right], \tag{4.20}$$

so the that the transfer function of the DCPED is

$$\mathbf{F}(D) = \mathbf{E}(D) \cdot \mathbf{R}(D) = \left[\frac{1 \boxminus D}{1 - D} \quad \frac{D}{1 - D}\right]. \tag{4.21}$$

Thus for Euclidean distance purposes, we can model a DCPFSK system as shown in Figure 4.15. We have effectively split the differential demodulator in a DCPFSK system into a differential decoder before the memoryless modulator and a coherent demodulator in the receiver.

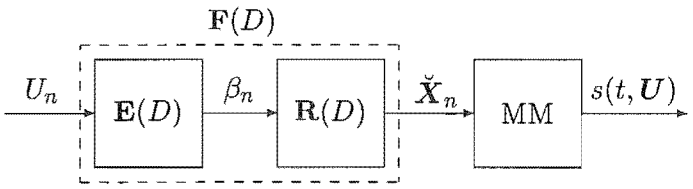



**Figure 4.15:** Euclidean distance model of DCPFSK

## 4.9 Code Search Model for DCPFSK

We use the channel encoder from Section 4.4 to encode DCPFSK. Consequently, we must remove the feedback in our DCPFSK model, and this is done by using the scrambler discussed in Section 4.3. By cascading the scrambler with the DCPED we have the transfer function

$$\dot{\mathbf{F}}(D) = \mathbf{T}(D) \cdot \mathbf{F}(D) = \left[1 \boxminus D \quad D\right]. \tag{4.22}$$

Note the similarity with the transfer function of the feedback-free CPE given in (4.7), the only difference is in the arithmetic. The feedback-free DCPED is shown in Figure 4.16, and the similarity with Figure 4.2(b) is evident.

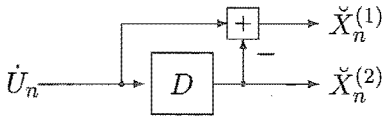



**Figure 4.16:** Feedback-free DCPED

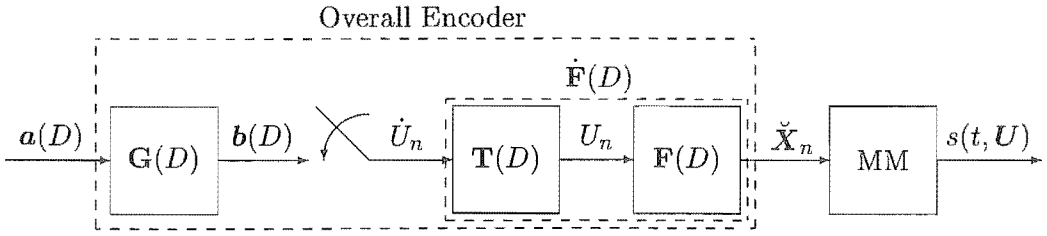



**Figure 4.17:** Coded DCPFSK system with scrambler

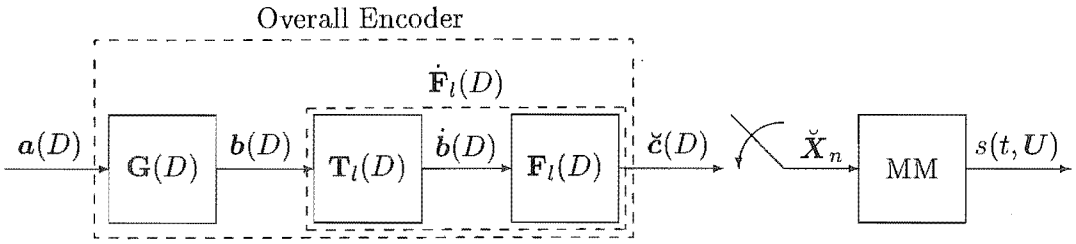



**Figure 4.18:** Coded DCPFSK system model for code searches

In Figure 4.17 we illustrate the coded DCPFSK system using the scrambler, which is very similar to that of the CPFSK case shown in Figure 4.4. Let $\mathbf{F}_l(D)$ and $\dot{\mathbf{F}}_l(D)$ denote the equivalent $l \times 2l$-versions of $\mathbf{F}(D)$ and $\dot{\mathbf{F}}(D)$ respectively, and let the output of $\dot{\mathbf{F}}_l(D)$ be denoted $\breve{\boldsymbol{c}}(D)$ where each $\breve{\boldsymbol{c}}_{k_V}$ is a $2l \times 1$ vector whose elements $\breve{c}_{k_V}^{(i)}$ are such that

$$\breve{c}_{k_V}^{(i)} \in \begin{cases} \{-(M-1), \ldots, -1, 0, 1, \ldots, M-1\}, & i \text{ odd} \\ \{0, 1, \ldots, M-1\}, & i \text{ even} \end{cases} \tag{4.23}$$

Although the overall range for the odd-numbered elements of $\breve{\boldsymbol{c}}_{k_V}$ has $2M-1$ values, it is important to note that at any one time the odd-numbered elements of $\breve{\boldsymbol{c}}_{k_V}$ can only take on one of $M$ values due to the structure of the DCPED.

The final system model for DCPFSK code searches is shown in Figure 4.18, which again compares to the CPFSK model in Figure 4.5. Note that $\breve{\boldsymbol{X}}_n$ is a $2 \times 1$ vector, and the commutator selects all the elements of each $\breve{\boldsymbol{c}}_{k_V}$ in pairs, so that

$$\breve{\boldsymbol{X}}_n = \begin{bmatrix} \breve{c}_{k_V}^{(2i-1)} \\ \breve{c}_{k_V}^{(2i)} \end{bmatrix}, \qquad i = 1, 2, \ldots, l, \qquad k_V l \leqslant n < (k_V + 1)l, \tag{4.24}$$

as in the CPFSK case.

Using the model in Figure 4.18 the NISED between two encoded sequences $\breve{\boldsymbol{c}}$ and

$\hat{\breve{\boldsymbol{c}}}$ over the $k_V$-th trellis interval, denoted $\delta^2_{k_V}(\breve{\boldsymbol{c}}, \hat{\breve{\boldsymbol{c}}})$, is defined as

$$\delta^2_{k_V}\left(\breve{\boldsymbol{c}}, \hat{\breve{\boldsymbol{c}}}\right) \triangleq \sum_{i=1}^{l} d_i^2\left(\begin{bmatrix}\breve{c}_{k_V}^{(2i-1)}\\ \breve{c}_{k_V}^{(2i)}\end{bmatrix}, \begin{bmatrix}\hat{\breve{c}}_{k_V}^{(2i-1)}\\ \hat{\breve{c}}_{k_V}^{(2i)}\end{bmatrix},\right), \tag{4.25}$$

where

$$d_i^2\left(\begin{bmatrix}\breve{c}_{k_V}^{(2i-1)}\\ \breve{c}_{k_V}^{(2i)}\end{bmatrix}, \begin{bmatrix}\hat{\breve{c}}_{k_V}^{(2i-1)}\\ \hat{\breve{c}}_{k_V}^{(2i)}\end{bmatrix},\right) \triangleq$$

$$\begin{cases} r\,\log_2 M\left[1 - \dfrac{\sin\left(2\pi h\left[\breve{c}_{k_V}^{(2i-1)} + \breve{c}_{k_V}^{(2i)} - \hat{\breve{c}}_{k_V}^{(2i-1)} - \hat{\breve{c}}_{k_V}^{(2i)}\right]\right) - \sin\left(2\pi h\left[\breve{c}_{k_V}^{(2i)} - \hat{\breve{c}}_{k_V}^{(2i)}\right]\right)}{2\pi h\left(\breve{c}_{k_V}^{(2i-1)} - \hat{\breve{c}}_{k_V}^{(2i-1)}\right)}\right], & \breve{c}_{k_V}^{(2i-1)} \neq \hat{\breve{c}}_{k_V}^{(2i-1)} \\ r\,\log_2 M\left[1 - \cos\left(2\pi h\left[\breve{c}_{k_V}^{(2i)} - \hat{\breve{c}}_{k_V}^{(2i)}\right]\right)\right], & \breve{c}_{k_V}^{(2i-1)} = \hat{\breve{c}}_{k_V}^{(2i-1)} \end{cases}. \tag{4.26}$$

Using (4.25), we write the normalised squared Euclidean distance between two coded sequences $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$ as $d^2(\boldsymbol{c}, \hat{\boldsymbol{c}})$, given by

$$d^2(\breve{c}, \hat{\breve{c}}) = \sum_{k_V} \delta^2_{k_V}(\breve{c}, \hat{\breve{c}}). \tag{4.27}$$

Similarly, the NMSED of a coded DCPFSK system is given as

$$d^2_{\min} = \min_{\substack{\breve{\boldsymbol{c}}, \hat{\breve{\boldsymbol{c}}} \\ \breve{\boldsymbol{c}} \neq \hat{\breve{\boldsymbol{c}}}}} d^2(\breve{c}, \hat{\breve{c}}). \tag{4.28}$$

## 4.10 Overall Encoder in a Coded DCPFSK System

Although the output of the overall encoder in Figure 4.18 follows the slightly peculiar format of (4.23), the overall encoder is still a linear encoder over $\mathbb{Z}_M$. Also, all the delay elements of the overall encoder will contain only elements of $\mathbb{Z}_M$. Due to the similarity between the feedback-free CPE and the feedback-free DCPED, similar reasoning to Section 4.6 can be applied, and the number of states in the overall encoder in a coded DCPFSK system can be determined using (4.13).

### 4.10.1 Rate-1/2 Ring-Coded 4-DCPFSK Overall Encoder Example

We again use the same channel encoder as that in the example in Section 4.6.1. When this is cascaded with the $2 \times 4$-version of the DCPED we obtain an overall encoder that is a rate-1/4 encoder with transfer function

$$\begin{aligned} \mathbf{G}(D)\,\dot{\mathbf{F}}_2(D) &= \begin{bmatrix} 1 & \dfrac{1}{1+2D} \end{bmatrix} \begin{bmatrix} 1 & 0 & \boxminus D & 1 \\ \boxminus D & D & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{1+2D \boxminus D}{1+2D} & \dfrac{D}{1+2D} & \dfrac{1 \boxminus (1+2D)}{1+2D} & 1 \end{bmatrix}. \end{aligned} \tag{4.29}$$

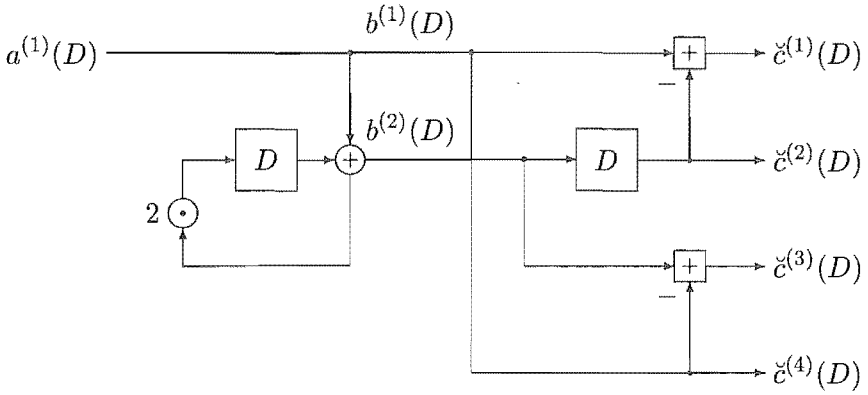



**Figure 4.19:** Implementation of $\mathbf{G}(D)\,\dot{\mathbf{F}}_2(D) = \left[\frac{1+2D \boxminus D}{1+2D} \quad \frac{D}{1+2D} \quad \frac{1 \boxminus (1+2D)}{1+2D} \quad 1\right]$

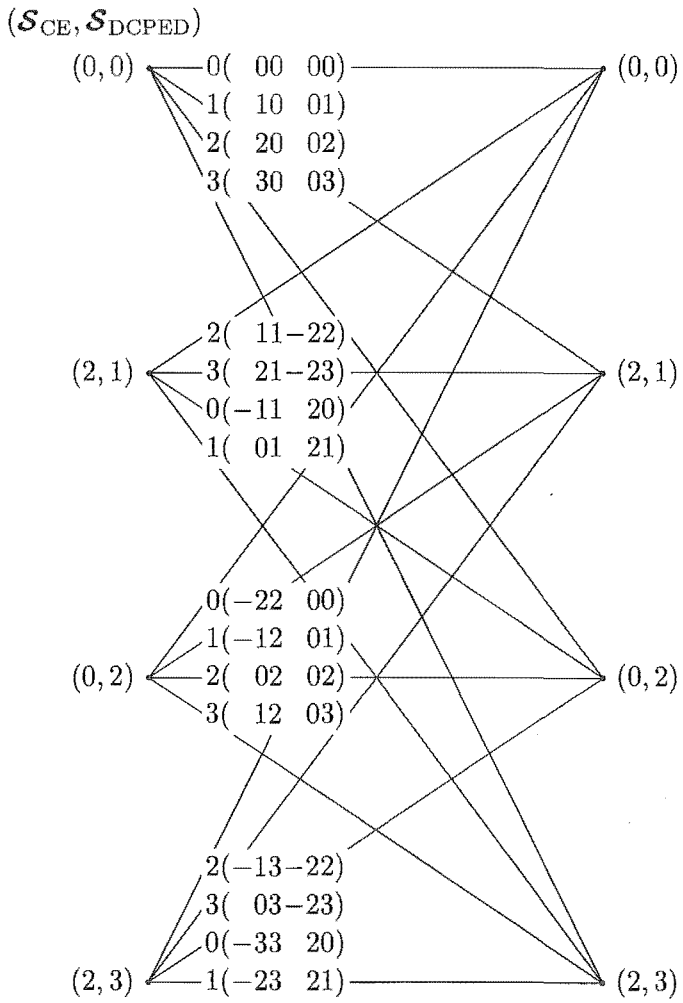



**Figure 4.20:** Trellis diagram of $\mathbf{G}(D)\,\dot{\mathbf{F}}_2(D) = \left[\frac{1+2D \boxminus D}{1+2D} \quad \frac{D}{1+2D} \quad \frac{1 \boxminus (1+2D)}{1+2D} \quad 1\right]$

This overall encoder is shown in Figure 4.19, and its trellis is illustrated in Figure 4.20. The contents of the two delay cells are used to identify the states, and the input and output for each transition are labelled as $a^{(1)}(\breve{c}^{(1)}\breve{c}^{(2)}\breve{c}^{(3)}\breve{c}^{(4)})$. Note that some of the elements in the first and third positions are negative, and this is the difference between the CPFSK and DCPFSK cases.

As the channel encoder is unchanged from Section 4.6.1, $n_s = 2$ and $S_G = 2$ so that from (4.13) we would expect the overall encoder to have four states. From inspection of Figure 4.20 we find this to be the case. Again, each transition in the trellis of Figure 4.20 represents *two* symbol intervals, as the memoryless modulator processes two elements of $\breve{c}_{k_V}$ each symbol period.

## 4.11 Implementation of a Coded DCPFSK System

In an actual coded DCPFSK system, the output of the channel encoder would be processed by a standard DCPFSK transmitter, consisting of a differential continuous phase encoder (DCPE) and a memoryless modulator. The transfer function of the DCPE $\breve{\mathbf{C}}(D)$ is the cascade of those of the differential encoder and the CPE,

$$\breve{\mathbf{C}}(D) = \mathbf{E}(D) \cdot \mathbf{C}(D) = \begin{bmatrix} \dfrac{1}{1-D} & \dfrac{D}{(1-D)^2} \end{bmatrix}. \tag{4.30}$$

Figure 4.21 illustrates the implementation of a coded DCPFSK system, the chosen code is scrambled and fed into a standard DCPFSK system.

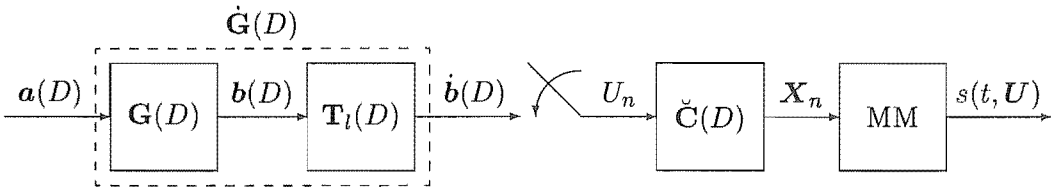



**Figure 4.21:** Implementation of a coded DCPFSK system

### 4.11.1 Rate-1/2 Ring-Coded 4-DCPFSK Implementation Example

We continue to consider the channel encoder of Section 4.6.1, and as the scrambled code is independent of the demodulation used, the transfer function of the scrambled code is given by (4.15) and its implementation and trellis are illustrated in Figures 4.11 and 4.12 respectively. Once fed into a standard DCPFSK system, the demodulated received signal would have the system trellis given in Figure 4.20.

## 4.12 Code Search Procedure

For each value of $\nu$ every possible generator matrix $\mathbf{G}(D)$ is generated and combined with the $l \times 2l$-version of the either the feedback-free CPE or the feedback-free DCPE to form the required overall encoder. Note that there will be $M^{l(\nu+1)-1}$ different generator matrices. The NMSED of the coded system can be calculated using either (4.12) or (4.28) and the Viterbi algorithm [BDMS91]. As well as the NMSED of the coded system, the value of $S_V$ must be recorded to ensure that coded systems of similar complexity can be compared.

## 4.13 Code Search Results

Two coded CPFSK and DCPFSK systems were considered, rate-1/2 encoded 4-CPFSK and rate-1/2 encoded 4-DCPFSK, which compare to uncoded MSK and uncoded DMSK respectively, and rate-2/3 encoded 8-CPFSK and rate-2/3 encoded 8-DCPFSK which compare to uncoded 4-CPFSK and uncoded 4-DCPFSK respectively, in terms of their information bit-rates. Tables 4.1 and 4.2 show the results of the searches, an * by the NMSED indicates that a search was not complete.

The results are ordered according to $S_V$ as this determines the system complexity. The $\mathbf{G}(D)$ given are just sample codes, as for each set of parameters there are a few codes that have the $d^2_{\text{min}}$ given, and the full list of codes can be found in Appendix G. All the coherent results found agree exactly with [YT94]—in terms of $d^2_{\text{min}}$—except in the octal, $S_V = 16$, $\nu = 1$ case. The best $d^2_{\text{min}}$ given for this case in [YT94] was 2.38, compared to the 2.81 found here. Only one code was given for each coherent case in [YT94], but a later paper [LJKK97] performed similar code searches for rate-1/2 encoded 4-CPFSK with $S_V = 4, 8, 16$. We found the same codes as [LJKK97] along with one additional code for the $S_V = 4$ case.

For rate-1/2 encoders working with 4-CPFSK and 4-DCPFSK systems (Table 4.1), there is always a loss in $d^2_{\text{min}}$ going from coherent to differential demodulation, although this may be as low as 0.03. Note that for the $S_V = 4$ and $S_V = 8$ cases, the same codes have maximum $d^2_{\text{min}}$ for both coherent and differential demodulation. In all the other $M = 4$ cases, different codes are required to achieve the maximum $d^2_{\text{min}}$ for coherent and differential demodulation.

For rate-2/3 encoders working with 8-CPFSK and 8-DCPFSK systems (Table 4.2),

**Table 4.1**

Search results for rate-1/2 encoded 4-CPFSK and 4-DCPFSK

| | | Coherent | | Differential | |
|---|---|---|---|---|---|
| $S_V$ | $\nu$ | $d^2_{\min}$ | $\mathbf{G}(D)$ | $d^2_{\min}$ | $\mathbf{G}(D)$ |
| 4 | 1 | 3.15 | $\left[1 \quad \frac{1}{2D+1}\right]$ | 3.00 | $\left[1 \quad \frac{1}{2D+1}\right]$ |
| 8 | 1 | 4.09 | $\left[1 \quad \frac{2D+1}{D+1}\right]$ | 4.00 | $\left[1 \quad \frac{2D+1}{D+1}\right]$ |
| 16 | 2 | 5.15 | $\left[\frac{2D^2+3D+2}{D+1} \quad 1\right]$ | 4.94 | $\left[1 \quad \frac{D+2}{2D^2+3D+1}\right]$ |
| 32 | 3 | 6.00 | $\left[\frac{D+2}{2D^3+D+1} \quad 1\right]$ | 5.45 | $\left[1 \quad \frac{D+1}{2D^3+D^2+3D+1}\right]$ |
| 64 | 3 | 6.42 | $\left[1 \quad \frac{2D^3+D^2+3}{D^3+2D+1}\right]$ | 6.39 | $\left[1 \quad \frac{D^2+D+2}{2D^3+D^2+3D+1}\right]$ |
| 128 | 4 | 7.60 | $\left[1 \quad \frac{2D^4+D^3+2D+1}{D^4+2D^3+D^2+1}\right]$ | 7.00 | $\left[1 \quad \frac{D^2+D+2}{2D^4+2D^3+D^2+3D+1}\right]$ |

**Table 4.2**

Search results for rate-2/3 encoded 8-CPFSK and 8-DCPFSK

(the * indicates that the search was not complete)

| | | Coherent | | Differential | |
|---|---|---|---|---|---|
| $S_V$ | $\nu$ | $d^2_{\min}$ | $\mathbf{G}(D)$ | $d^2_{\min}$ | $\mathbf{G}(D)$ |
| 8 | 1 | 2.18 | $\begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}$ | 2.18 | $\begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}$ |
| 16 | 1 | 2.81 | $\begin{bmatrix} 1 & 0 & 6D+4 \\ 0 & 1 & 4D+2 \end{bmatrix}$ | 2.81 | $\begin{bmatrix} 1 & 0 & 6D+4 \\ 0 & 1 & 4D+2 \end{bmatrix}$ |
| 32 | 2 | 2.93* | $\begin{bmatrix} 1 & 0 & \frac{2D+4}{3D^2+4D+1} \\ 0 & 1 & \frac{D+1}{3D^2+4D+1} \end{bmatrix}$ | 2.93* | $\begin{bmatrix} 1 & 0 & \frac{2D+4}{3D^2+4D+1} \\ 0 & 1 & \frac{D+1}{3D^2+4D+1} \end{bmatrix}$ |

the $d^2_{\min}$ of a code was found to be the exactly the same for both coherent and differential reception, at least for the $S_V = 8$ and $S_V = 16$ cases. A very limited search of $S_V = 32$ codes showed the same characteristic. Note that these trellises are very big, as a rate-2/3 encoder with octal symbols implies 64 branches from each state in the trellis.

## 4.14 Summary

In this chapter we presented the method used to encode CPFSK in [YT94]. A systematic convolutional channel encoder over $\mathbb{Z}_M$ is combined with a feedback-free $M$-CPFSK system. The combination of the channel encoder and the feedback-free CPE forms an overall encoder that is also a linear encoder over $\mathbb{Z}_M$. A model for code searching can then be constructed using this overall encoder as in [YT94]. We then applied a similar method to develop a Euclidean distance model of DCPFSK which led on to a code search model of DCPFSK.

The results of the code searches showed that very little performance is lost (in terms of NMSED) when going from CPFSK to DCPFSK. In the case of rate-2/3 encoded 8-CPFSK and 8-DCPFSK, it was found that there was no difference in NMSED between the encoded coherent and differential systems.

# Chapter 5

# Performance of Coded (D)CPFSK Systems in AWGN

## 5.1 Introduction

Although the normalised minimum squared Euclidean distance (NMSED or $d^2_{\text{min}}$) of a coded (D)CPFSK system is the most important performance figure, it does not tell the whole story. In the uncoded cases, the simple approximations of (2.62) and (3.112) approximate a system's performance adequately, as an uncoded system has a very small system trellis. In the coded cases, the system trellises are much more complicated, so that further analysis is required to accurately predict their performance. The problem is that there may be many different paths through the trellis that achieve $d^2_{\text{min}}$, so that each path's contribution to the probability of bit error $P_e$ must be considered. Additionally, there may be other paths through the trellis that have a normalised squared Euclidean distance (NSED) very close to the $d^2_{\text{min}}$ whose contribution to $P_e$ may also be significant.

In [Yan94], Yang applies average transfer function techniques to this problem. However, the computation is very complex and significant approximations need to be made. In this chapter we apply a simpler brute-force method to determine the contribution of the paths through the system trellis. In Section 5.2 we apply the work in [Yan94] to our coded models of CPFSK and DCPFSK to obtain a union bound on $P_e$, which consists of a sum of distance terms and error coefficients. We discuss how to calculate these error coefficients in Section 5.3. We present the computed and simulated results for various coded and uncoded (D)CPFSK schemes in Sections 5.4 and 5.5 respectively. We compare comparable systems' performance in Section 5.6 to determine the improvements available through coding. We summarise the chapter and draw some conclusions in Section 5.7.

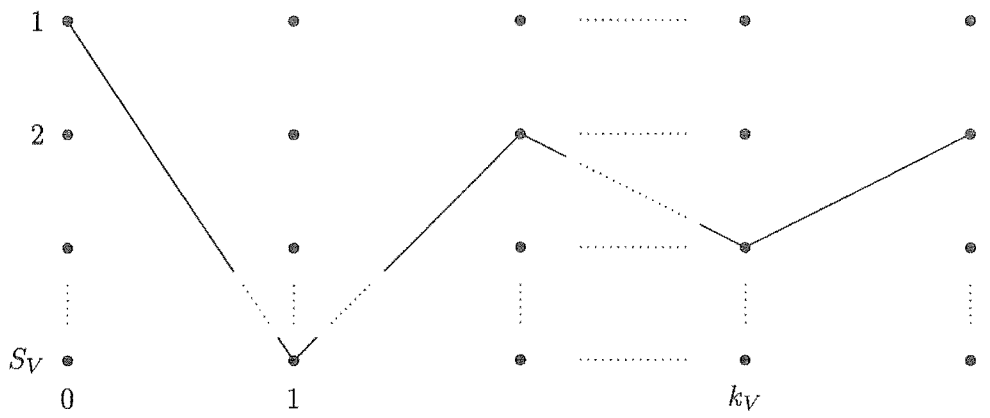



**Figure 5.1:** A transmitted path in the trellis of the overall encoder

## 5.2 Probability of Error Analysis

The analysis focuses on the code search models for coded CPFSK and DCPFSK shown in Figures (4.5) and (4.18) respectively, as they enable us to treat all the encoders in a system as a single overall encoder. Figure 5.1 shows the an example transmitted path in an overall encoder. In each system trellis interval $l-1$ $M$-ary information symbols are encoded and transmitted. The information symbols transmitted in the $k_V$-th trellis interval are denoted $\boldsymbol{a}_{k_V}$, as defined in (4.2), and the entire sequence is denoted $\boldsymbol{a}(D)$. The elements $a_{k_V}^{(i)}$ are assumed to be independent and identically distributed. The received and decoded sequence is denoted $\hat{\boldsymbol{a}}(D)$. There are $S_V$ states in the overall encoder and the state sequence that the transmitter goes through is denoted $\mathcal{S}(D)$, where

$$\mathcal{S}(D) \triangleq \mathcal{S}_0 + \mathcal{S}_1 D + \cdots + \mathcal{S}_{k_V} D^{k_V} + \ldots, \qquad \mathcal{S}_{k_V} \in \{1, 2, \ldots, S_V\}. \tag{5.1}$$

The state sequence that the receiver goes through is similarly denoted by $\hat{\mathcal{S}}(D)$.

Let us assume that the transmitted sequence $\boldsymbol{a}(D)$ is $L$ trellis intervals long. The total number of information symbols in the transmitted sequence is then $(l-1)\,L$, and the total number of transmitted bits is equal to $(l-1)\log_2 M\,L$. Let $m_b[\boldsymbol{a}(D), \hat{\boldsymbol{a}}(D)]$ be the number of bits that differ between $\boldsymbol{a}(D)$ and $\hat{\boldsymbol{a}}(D)$, using an appropriate mapping method (e.g., natural or Gray). The probability of a bit error is defined as the expected number of information bit errors per decoded information bit,

$$P_e \triangleq \frac{\mathrm{E}\left[m_b[\boldsymbol{a}(D), \hat{\boldsymbol{a}}(D)]\right]}{(l-1)\log_2 M\,L} \tag{5.2}$$

where the expectation is over the sequence pairs $(\boldsymbol{a}(D), \hat{\boldsymbol{a}}(D))$. The upper bound on $P_e$ is then obtained by letting $L \to \infty$. Strictly speaking, unless the transmission is assumed to

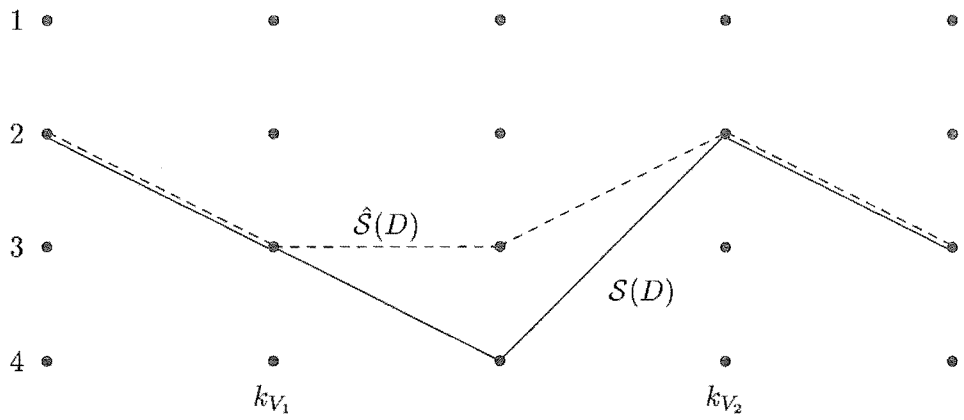



**Figure 5.2:** An example of an error event in a four-state trellis. $S(D)$ is the transmitted path, and $\hat{S}(D)$ the receiver's decided path

start at time $-\infty$ and end at time $+\infty$, this probability is a function of the discrete time $k_V$ at which the error event starts. We assume that the transmission is long enough that we may disregard this difficulty.

We now need to define what an *error event* is. An error event $\mathcal{E}$ is said to extend from time $k_{V_1}$ to $k_{V_2}$ if $\hat{S}(D)$ is equal to $S(D)$ at times $k_{V_1}$ and $k_{V_2}$, but nowhere in between, so that

$$\begin{aligned} \hat{S}_{k_{V_1}} &= S_{k_{V_1}} \\ \hat{S}_{k_V} &\neq S_{k_V}, \qquad k_{V_1} < k_V < k_{V_2} \\ \hat{S}_{k_{V_2}} &= S_{k_{V_2}}. \end{aligned} \tag{5.3}$$

The length of the error event is defined as $\zeta = k_{V_2} - k_{V_1}$. An example error event in a four-state trellis is illustrated in Figure 5.2. Note that for $\hat{S}(D)$ and $S(D)$ to diverge at time $k_{V_1}$, we must have $\hat{\boldsymbol{a}}_{k_{V_1}} \neq \hat{\boldsymbol{a}}_{k_{V_2}}$.

Whenever a decoding error occurs, an error event must be in progress or starting. Assume the sequences $\boldsymbol{a}(D)$ and $\hat{\boldsymbol{a}}(D)$ generate a sequence of error events. The random variables $W_{k_V}$, $k_V = 0, 1, \ldots, L$, are defined in the following manner: if $\boldsymbol{a}(D)$ and $\hat{\boldsymbol{a}}(D)$ are such that an error event starts at time $k_V$, then let $W_{k_V}$ denote the number of information bit errors given by the entire error event in question. If $\boldsymbol{a}(D)$ and $\hat{\boldsymbol{a}}(D)$ are such that an error event does not start at time $k_V$ (or an error event is already in progress), then let $W_{k_V} = 0$. The total number of information bit errors caused by $\boldsymbol{a}(D)$ and $\hat{\boldsymbol{a}}(D)$ is

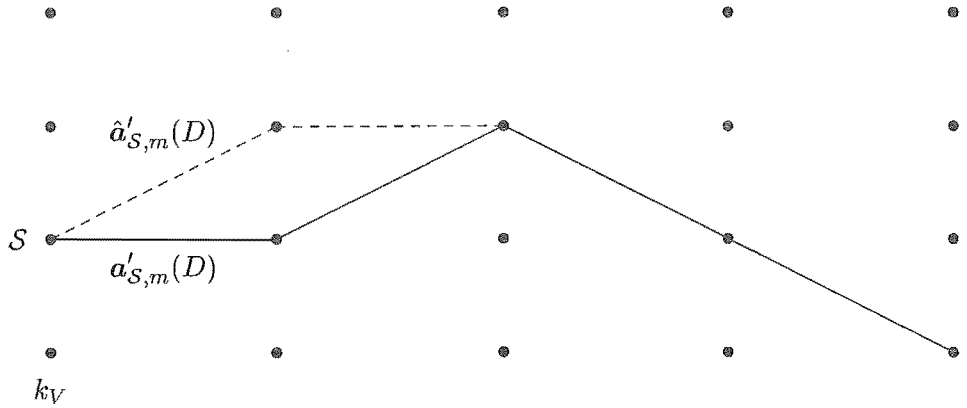



**Figure 5.3:** A specific error event $\mathcal{E}_{\mathcal{S},m}$ starting from state $\mathcal{S}$ at time $k_V$

$\sum_{k_V=0}^{L-1} W_{k_V}$, and the bit error probability can be written as

$$\begin{aligned} P_e &= \frac{1}{(l-1)\log_2 M\, L}\, \mathrm{E}\left[\sum_{k_V=0}^{L-1} W_{k_V}\right] \\ &= \frac{1}{(l-1)\log_2 M\, L} \sum_{k_V=0}^{L-1} \mathrm{E}\left[W_{k_V}\right]. \end{aligned} \tag{5.4}$$

To find $\mathrm{E}[W_{k_V}]$, let us study the set of all error events starting at time $k_V$. Take an arbitrary state $\mathcal{S}$, $\mathcal{S} \in \{1, 2, \ldots, S_V\}$. Find all the error events which start from this state and number them $\mathcal{E}_{\mathcal{S},1}, \mathcal{E}_{\mathcal{S},2}, \ldots, \mathcal{E}_{\mathcal{S},m}, \ldots$ Each error event $\mathcal{E}_{\mathcal{S},m}$ is completely described by its start state and the pair of sequences $(\boldsymbol{a}'_{\mathcal{S},m}(D), \hat{\boldsymbol{a}}'_{\mathcal{S},m}(D))$ which generate $\mathcal{E}_{\mathcal{S},m}$, as shown in Figure 5.3. As (D)CPFSK codes are not linear, this process must then be repeated for the other $S_V - 1$ states. The length (in trellis intervals) of each error event is called $\zeta_{\mathcal{S},m}$, and the number of information bit errors contributed by each error event $\iota_{\mathcal{S},m}$. The normalised squared Euclidean distance (NSED) associated with the specific error event $\mathcal{E}_{\mathcal{S},m}$ is $d^2_{\mathcal{S},m}$. So far, we have studied the set of all error events starting at time $k_V$. Each specific member in this set is (partially) characterised by the three parameters $\iota_{\mathcal{S},m}$, $\zeta_{\mathcal{S},m}$ and $d^2_{\mathcal{S},m}$.

Our goal is to upper bound $\mathrm{E}[W_{k_V}]$ in (5.4). To be able to do this we define the events $\varepsilon_{k_V,\mathcal{S},m}$ and $\hat{\varepsilon}_{k_V,\mathcal{S},m}$, $k_V = 0, 1, \ldots, L-1$, $\mathcal{S} = 1, 2, \ldots, S_V$, $m = 1, 2, \ldots$, ($m$ is finite), as

$\varepsilon_{k_V,\mathcal{S},m}$: the event that $\hat{\boldsymbol{a}}(D)$ is such that $\mathcal{S}_{k_V} = \mathcal{S}$ and
$$\boldsymbol{a}_{k_V} + \boldsymbol{a}_{k_V+1} D + \cdots + \boldsymbol{a}_{k_V+\zeta_{\mathcal{S},m}-1} D^{\zeta_{\mathcal{S},m}-1} = \boldsymbol{a}'_{\mathcal{S},m}(D)$$

$\hat{\varepsilon}_{k_V,\mathcal{S},m}$: the event that $\hat{\boldsymbol{a}}(D)$ is such that $\hat{\mathcal{S}}_{k_V} = \mathcal{S}$ and
$$\hat{\boldsymbol{a}}_{k_V} + \hat{\boldsymbol{a}}_{k_V+1} D + \cdots + \hat{\boldsymbol{a}}_{k_V+\zeta_{\mathcal{S},m}-1} D^{\zeta_{\mathcal{S},m}-1} = \hat{\boldsymbol{a}}'_{\mathcal{S},m}(D).$$

Thus the probability that the event $\varepsilon_{k_V,\mathcal{S},m}$ will occur is given by

$$\Pr\{\varepsilon_{k_V,\mathcal{S},m}\} = \Pr\{\mathcal{S}_{k_V} = \mathcal{S}\} \Pr\Big\{ \boldsymbol{a}_{k_V} + \boldsymbol{a}_{k_V+1} D + \cdots + \boldsymbol{a}_{k_V+\zeta_{\mathcal{S},m}-1} D^{\zeta_{\mathcal{S},m}-1} = \boldsymbol{a}'_{\mathcal{S},m}(D) \Big\} . \tag{5.5}$$

As we are assuming that the information bits are independent and equally likely, we have

$$\Pr\{\mathcal{S}_{k_V} = \mathcal{S}\} \;=\; \frac{1}{S_V}, \qquad \mathcal{S} = 1, 2, \ldots, S_V, \tag{5.6}$$

and

$$\Pr\Big\{ \boldsymbol{a}_{k_V} + \boldsymbol{a}_{k_V+1} D + \cdots + \boldsymbol{a}_{k_V+\zeta_{\mathcal{S},m}-1} D^{\zeta_{\mathcal{S},m}-1} = \boldsymbol{a}'_{\mathcal{S},m}(D) \Big\} \;=\; \left(\frac{1}{M}\right)^{(l-1)\,\zeta_{\mathcal{S},\,m}} . \tag{5.7}$$

Substituting (5.6) and (5.7) in (5.5), we obtain

$$\Pr\{\varepsilon_{k_V,\mathcal{S},m}\} \;=\; \frac{1}{S_V} \left(\frac{1}{M}\right)^{(l-1)\,\zeta_{\mathcal{S},\,m}} . \tag{5.8}$$

The situation when the specific error event $\mathcal{E}_{\mathcal{S},m}$ actually occurs is described by the joint event $(\varepsilon_{k_V,\mathcal{S},m},\ \hat{\varepsilon}_{k_V,\mathcal{S},m})$ starting at level $k_V$. By applying the union bound to $\mathrm{E}[W_{k_V}]$, we have

$$\begin{aligned} \mathrm{E}[W_{k_V}] \;&=\; \sum_{\iota} \iota \Pr\{W_{k_V} = \iota\} \\ &\leqslant \sum_{\mathcal{S}} \sum_{m} \iota_{\mathcal{S},m} \Pr\{\varepsilon_{k_V,\mathcal{S},m}, \hat{\varepsilon}_{k_V,\mathcal{S},m}\} \\ &= \sum_{\mathcal{S}} \sum_{m} \iota_{\mathcal{S},m} \Pr\{\hat{\varepsilon}_{k_V,\mathcal{S},m} | \varepsilon_{k_V,\mathcal{S},m}\} \Pr\{\varepsilon_{k_V,\mathcal{S},m}\} . \end{aligned} \tag{5.9}$$

For coherent reception, the quantity $\Pr\{\hat{\varepsilon}_{k_V,\mathcal{S},m} | \varepsilon_{k_V,\mathcal{S},m}\}$ is given by [For72]

$$\Pr\{\hat{\varepsilon}_{k_V,\mathcal{S},m} | \varepsilon_{k_V,\mathcal{S},m}\} \;=\; Q\left(\sqrt{\frac{d^2_{\mathcal{S},m} E_b}{N_0}}\right) . \tag{5.10}$$

Using (5.8) and (5.10) in (5.9), we obtain

$$\mathrm{E}[W_{k_V}] \;\leqslant\; \frac{1}{S_V} \sum_{\mathcal{S}} \sum_{m} \iota_{\mathcal{S},m} \left(\frac{1}{M}\right)^{(l-1)\,\zeta_{\mathcal{S},\,m}} Q\left(\sqrt{\frac{d^2_{\mathcal{S},m} E_b}{N_0}}\right) . \tag{5.11}$$

We can rewrite (5.8) in terms of the variables $\iota$, $\zeta$, $d$ as

$$\mathrm{E}[W_{k_V}] \;\leqslant\; \frac{1}{S_V} \sum_{\mathcal{S}} \sum_{\iota} \sum_{\zeta} \sum_{d \in \mathcal{D}} \eta(\mathcal{S}, \iota, \zeta, d)\, \iota \left(\frac{1}{M}\right)^{(l-1)\,\zeta} Q\left(\sqrt{\frac{d^2 E_b}{N_0}}\right) , \tag{5.12}$$

where $\eta(\mathcal{S}, \iota, \zeta, d)$ is the number of error events starting from state $\mathcal{S}$, assuming the trellis is infinite, having $\iota$ information bit errors, a length of $\zeta$ trellis intervals and a NSED equal to $d^2$, and $\mathcal{D}$ is the set of all NSEDs for the code in question.

As we have assumed that the transmitted sequence is sufficiently long ($L \to \infty$), we can say that $\mathrm{E}[W_{k_V}]$ is independent of $k_V$ [Vit71], so that

$$\mathrm{E}[W_{k_V}] = \mathrm{E}[W], \quad \forall k_V, \tag{5.13}$$

and consequently (5.4) becomes

$$P_e = \frac{1}{(l-1)\log_2 M}\,\mathrm{E}[W_{k_V}]. \tag{5.14}$$

Using (5.12) in (5.14), we obtain

$$P_e \leqslant \frac{1}{(l-1)\log_2 M S_V} \sum_{\mathcal{S}} \sum_{\iota} \sum_{\zeta} \sum_{d\in\mathcal{D}} \eta(\mathcal{S},\iota,\zeta,d)\,\iota \left(\frac{1}{M}\right)^{(l-1)\zeta} Q\left(\sqrt{\frac{d^2E_b}{N_0}}\right). \tag{5.15}$$

By defining the *error coefficient* $\Xi(d)$ as

$$\Xi(d) \triangleq \frac{1}{(l-1)\log_2 M S_V} \sum_{\mathcal{S}} \sum_{\iota} \sum_{\zeta} \eta(\mathcal{S},\iota,\zeta,d)\,\iota \left(\frac{1}{M}\right)^{(l-1)\zeta}, \tag{5.16}$$

and using (5.15) we can write the union bound on the probability of bit error for a rate-$(l-1)/l$ coded $M$-CPFSK system as

$$P_{e,\mathrm{coh}} \leqslant \sum_{d\in\mathcal{D}} \Xi(d)\, Q\left(\sqrt{\frac{d^2E_b}{N_0}}\right), \tag{5.17}$$

where $\Xi(d)$ and $d^2$ in (5.17) can be calculated using (5.16) and (4.11).

By using the assumption that differential demodulation incurs a 3dB noise penalty over coherent demodulation—as discussed in Section 3.10—we can write the union bound on the probability of bit error for a rate-$(l-1)/l$ coded $M$-DCPFSK system as

$$P_{e,\mathrm{diff}} \leqslant \sum_{d\in\mathcal{D}} \Xi(d)\, Q\left(\sqrt{\frac{d^2E_b}{2N_0}}\right). \tag{5.18}$$

Note that $\Xi(d)$ and $d^2$ in (5.18) should be calculated using (5.16) and (4.27).

At high SNR, the contribution of the Q-function to both (5.17) and (5.18) is generally more significant than that of $\Xi(d)$. Thus the term with the NMSED $d^2_{\min}$ will dominate performance, and approximations to (5.17) and (5.18) are given by

$$P_{e,\mathrm{coh}} \approx \Xi(d_{\min})\, Q\left(\sqrt{\frac{d^2_{\min}E_b}{N_0}}\right) \tag{5.19}$$

$$P_{e,\mathrm{diff}} \approx \Xi(d_{\min})\, Q\left(\sqrt{\frac{d^2_{\min}E_b}{2N_0}}\right). \tag{5.20}$$

A further approximation may be obtained by setting $\Xi(d_{\min})$ equal to one, indeed, (2.62) and (3.112) can be obtained in this manner.

## 5.3 Calculating the Error Coefficient

The expression for the error coefficient in (5.16) assumes an infinite trellis, which implies that the sums over $\zeta$ and $\iota$ may be infinite, and makes exact calculation of $\Xi(d)$ by a search method impossible. In [Yan94] Yang avoids this by reformulating (5.15) into an average transfer function problem, but then has difficulty with computation requirements.

We choose instead to use a limited-length exhaustive search. By limiting the length of the error events to a set number of trellis intervals, we ensure that all the sums in (5.16) will be finite. To facilitate a limited-length search, we rearrange (5.16) into the form

$$\Xi(d) \triangleq \frac{1}{(l-1)\log_2 MS_V} \sum_{\zeta} \left(\frac{1}{M}\right)^{(l-1)\zeta} \sum_{\mathcal{S}} \sum_{\iota} \eta(\mathcal{S}, \iota, \zeta, d)\, \iota \,. \tag{5.21}$$

The quantity $\sum_{\mathcal{S}} \sum_{\iota} \eta(\mathcal{S}, \iota, \zeta, d)\, \iota$ in (5.21), can be thought of as the total number of bit errors for all error events with length $\zeta$ and NSED $d^2$. This enables us to organise the error event information by NSED and length. During the exhaustive search, each error event that merges within the set number of trellis intervals must be noted, along with its length (in trellis intervals), number of bit errors and NSED. Once this is done, an approximate $P_e$ can be easily calculated. It is also very important to record the smallest NSED of the unmerged error events. The reason for this will become apparent in the next section.

## 5.4 Performance of Coded (D)CPFSK Systems

In this section we present the results of the limited-length error coefficient searches, and compare them with those of simulations. The simulation models used are the same as those discussed in Sections 2.10 and 3.11, except for the extra encoding and decoding. The graphs presented all follow a similar format, comparing various computed approximations with simulation results. There are three different computed performance approximations used: a partial union bound, a 2-dB approximation, and a simple approximation. Each is based on the results of the error coefficient searches and either (5.17) or (5.18). The partial union bound is denoted by a solid line and uses all the contributions of *all* the error coefficients and distances found in the searches. It is important to note that this is a truncated bound as the searches only find the error events up to a certain length. The 2-dB approximation is denoted by a dashed line, and only uses the contributions of the error coefficients whose distances are less than 2 dB away from the $d^2_{\min}$ of the system. The simple approximation is denoted by a dotted line, and uses only the $d^2_{\min}$ term with $\Xi(d_{\min}) = 1$, similar to (2.62) and (3.112) used in Sections 2.9 and 3.10 respectively.

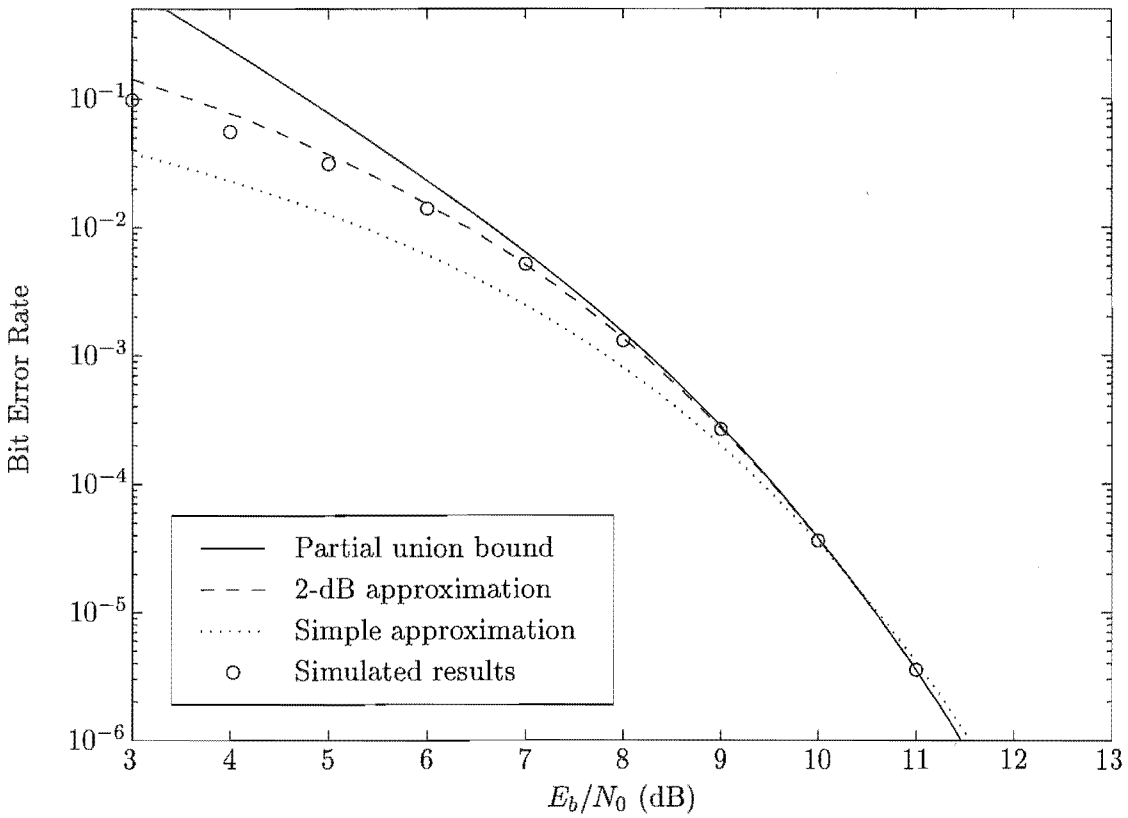



**Figure 5.4:** Computed and simulated performance of 4-state, rate-1/2 encoded 4-CPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$. The computed approximations and bounds were calculated using (5.17) and Table 5.2(a).

### 5.4.1 Performance of 4-state, rate-1/2 encoded 4-CPFSK

Table 5.1(a) shows the results of the exhaustive error coefficient search for 4-state, rate-1/2 encoded 4-CPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$, the error events are limited to two trellis intervals. The smallest $d^2$ of the unmerged error events is 2.09, which is less than the $d^2_{\text{min}}$ of the system. The squared Euclidean distance (SED) incurred over each unmerged trellis interval is always positive, but may be small enough to add to the error coefficient for the 3.15-distance term, or even to introduce an entry between the 3.15 and 4.00 entries. In Table 5.2(a), which has been truncated due to the large number of entries, we present the results of the exhaustive error coefficient search for the same system but limited to six trellis intervals, and it is evident that the longer error events introduce two new terms between the 3.15 and 4.00 entries. The smallest $d^2$ of the unmerged error events in Table 5.2(a) is 5.00, which is 2 dB greater than the $d^2_{\text{min}}$ of the system. Using (5.17) and results in Table 5.2(a), we can produce computed approximations for the performance of this system, as shown in Figure 5.4. All the computed curves agree with the simulation results at high

**Table 5.1**

Error coefficients for 4-state, rate-1/2 encoded 4-(D)CPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{1}{2D+1}\right]$

(2 intervals deep, smallest $d^2$ of unmerged error events: 2.09)

(a) 4-CPFSK

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota} \eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 3.15 | 2 | 72 | 0.56 |
| 4.00 | 2 | 368 | 2.88 |
| 4.85 | 2 | 72 | 0.56 |

(b) 4-DCPFSK

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota} \eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 3.00 | 2 | 64 | 0.50 |
| 3.15 | 2 | 120 | 0.94 |
| 3.49 | 2 | 24 | 0.19 |
| 4.00 | 2 | 232 | 1.81 |
| 4.34 | 2 | 24 | 0.19 |
| 4.85 | 2 | 48 | 0.38 |

**Table 5.2**

Error coefficients for 4-state, rate-1/2 encoded 4-(D)CPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{1}{2D+1}\right]$

(6 intervals deep, smallest $d^2$ of unmerged error events: 5.00)

(a) 4-CPFSK (421 entries)

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota} \eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 3.15 | 2 | 72 | 0.56 |
| 3.73 | 3 | 288 | 0.56 |
| 3.88 | 3 | 96 | 0.19 |
| 4.00 | 2 | 368 | 2.88 |
| 4.45 | 4 | 576 | 0.28 |
| 4.51 | 3 | 192 | 0.38 |
| 4.58 | 3 | 192 | 0.38 |
| 4.60 | 4 | 120 | 0.06 |
| 4.73 | 3 | 320 | 0.62 |
| 4.85 | 2 | 72 | 0.56 |
| 5.09 | 4 | 1152 | 0.56 |
| 5.15 | 3 | 192 | 0.38 |
| 5.18 | 5 | 864 | 0.11 |
| 5.24 | 4 | 480 | 0.23 |
| 5.30 | 4 | 384 | 0.19 |
| ⋮ | ⋮ | ⋮ | ⋮ |

(b) 4-DCPFSK (1000 entries)

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota} \eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 3.00 | 2 | 64 | 0.50 |
| 3.15 | 2 | 120 | 0.94 |
| 3.49 | 2 | 24 | 0.19 |
| 3.73 | 3 | 320 | 0.62 |
| 3.88 | 3 | 320 | 0.62 |
| 4.00 | 2 | 232 | 1.81 |
| 4.34 | 2 | 24 | 0.19 |
| 4.45 | 4 | 1280 | 0.62 |
| 4.51 | 3 | 384 | 0.75 |
| 4.58 | 3 | 128 | 0.25 |
| 4.60 | 4 | 800 | 0.39 |
| 4.73 | 3 | 384 | 0.75 |
| 4.85 | 2 | 48 | 0.38 |
| 5.00 | 3 | 128 | 0.25 |
| 5.09 | 4 | 1536 | 0.75 |
| ⋮ | ⋮ | ⋮ | ⋮ |

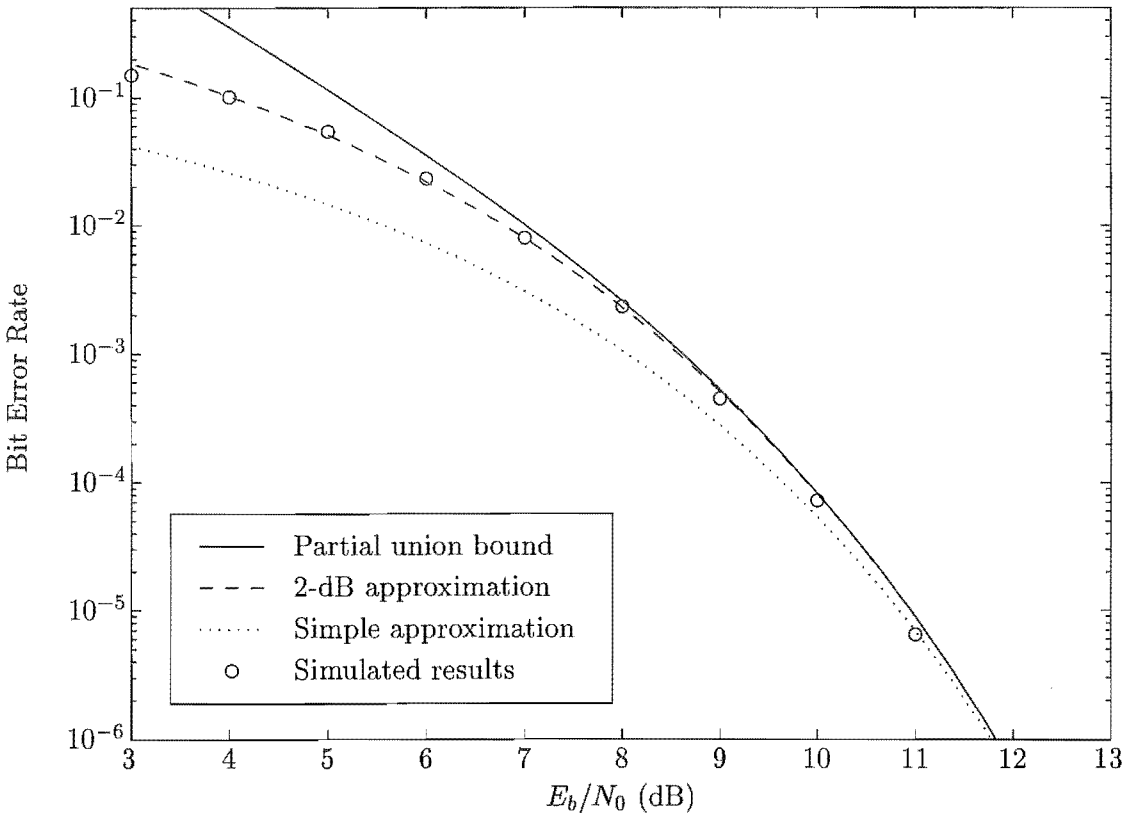



**Figure 5.5:** Computed and simulated performance of 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$. The computed approximations and bounds were calculated using (5.18) and Table 5.2(b).

SNR, but only the 2-dB approximation predicts the performance at lower SNR values. The 2-dB approximation uses the 10 terms whose $d^2 \leqslant 5.00$.

### 5.4.2 Performance of 4-state, rate-1/2 encoded 4-DCPFSK

The results of the exhaustive error coefficient search for 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ are presented in Table 5.1(b), where the error events are limited to two trellis intervals. As in the coherent case, the smallest $d^2$ of the unmerged error events is 2.09, which is less than the smallest $d^2$ of the merged error events. The truncated results for the longer length search are shown in Table 5.2(b), and the smallest $d^2$ of the unmerged error events is 5.00, which is 2.2 dB greater than the $d^2_{\text{min}}$ of the system. It is also interesting to note that the differential system has more unique distances than the equivalent coherent system. The simulated and computed performance of this system is shown in Figure 5.5. The computed curves were calculated using (5.18) and the results in Table 5.2(b). Again, the 2-dB approximation predicts the system's performance very

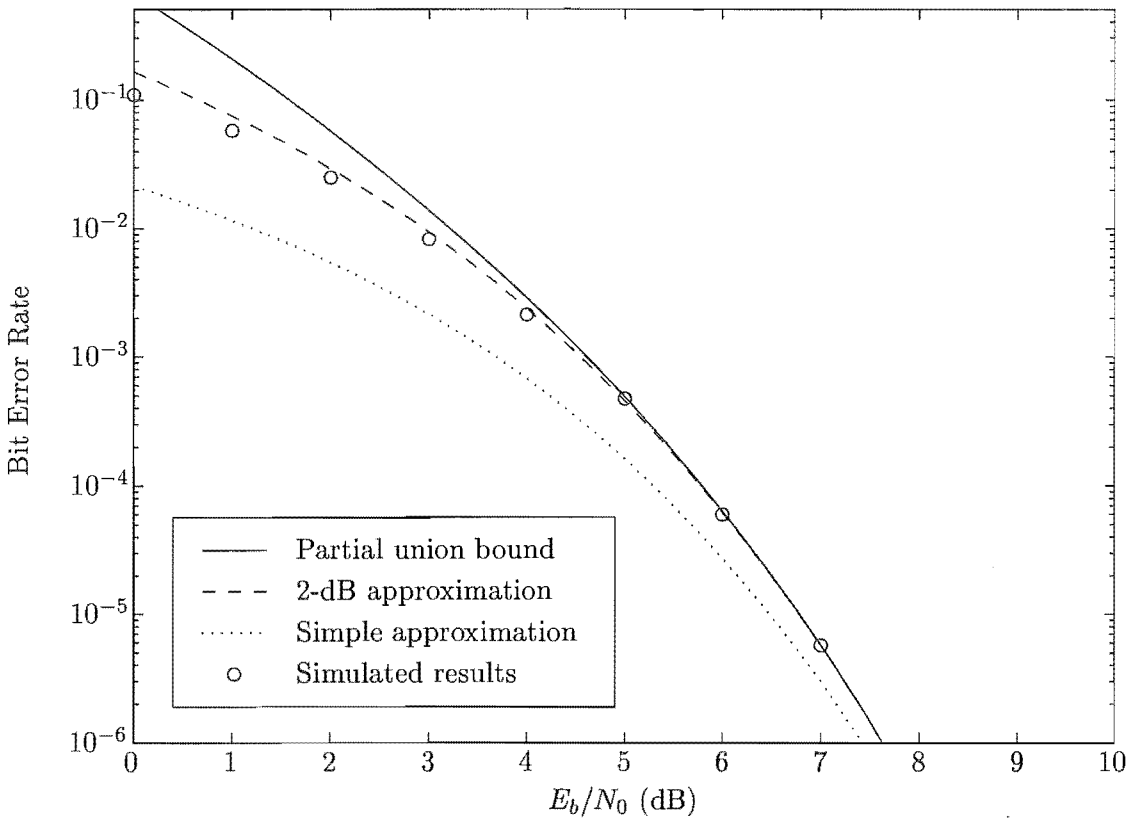



**Figure 5.6:** Computed and simulated performance of 8-state, rate-1/2 encoded 4-CPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{2D+1}{D+1}\right]$. The computed approximations and bounds were calculated using (5.17) and Table 5.3(a).

well. The 2-dB approximation uses the 12 terms whose $d^2 \leqslant 4.75$.

### 5.4.3 Performance of 8-state, rate-1/2 encoded 4-CPFSK

Table 5.3(a) shows the results of the exhaustive error coefficient search for 8-state, rate-1/2 encoded 4-CPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{2D+1}{D+1}\right]$. Due to the computation requirements, the error events were limited to six trellis intervals. The smallest $d^2$ of the unmerged error events is 5.00, which is only 0.9 dB greater than the $d^2_{\text{min}}$ of the system, but the 2-dB approximation still predicts the performance of the system very well, as shown in Figure 5.6. The 2-dB approximation uses the 26 terms whose $d^2 \leqslant 6.48$. Note that the simple approximation is slightly optimistic for all SNR values.

### 5.4.4 Performance of 8-state, rate-1/2 encoded 4-DCPFSK

Table 5.3(b) shows the results of the exhaustive error coefficient search for 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{2D+1}{D+1}\right]$. As in the coherent case, the computation

**Table 5.3**

Error coefficients for 8-state, rate-1/2 encoded 4-(D)CPFSK using $\mathbf{G}(D) = \left[1 \quad \frac{2D+1}{D+1}\right]$

(6 intervals deep, smallest $d^2$ of unmerged error events: 5.00)

(a) 4-CPFSK (346 entries)

| $d^2$ | $\zeta$ | $\sum_S \sum_\iota \eta(S,\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 4.09 | 3 | 1280 | 1.25 |
| 4.15 | 3 | 256 | 0.25 |
| 4.24 | 4 | 320 | 0.08 |
| 4.94 | 3 | 512 | 0.50 |
| 4.97 | 5 | 1600 | 0.10 |
| 5.00 | 3 | 1536 | 1.50 |
| 5.09 | 4 | 2176 | 0.53 |
| 5.36 | 3 | 1280 | 1.25 |
| 5.45 | 4 | 6400 | 1.56 |
| 5.51 | 4 | 320 | 0.08 |
| 5.54 | 5 | 7680 | 0.47 |
| 5.60 | 5 | 5120 | 0.31 |
| 5.69 | 6 | 5760 | 0.09 |
| 5.79 | 3 | 256 | 0.25 |
| 5.82 | 5 | 4608 | 0.28 |
| 5.85 | 3 | 256 | 0.25 |
| 5.94 | 4 | 2176 | 0.53 |
| 6.00 | 2 | 512 | 2.00 |
| 6.09 | 4 | 5120 | 1.25 |
| 6.18 | 5 | 12800 | 0.78 |
| 6.21 | 3 | 512 | 0.50 |
| 6.24 | 5 | 1600 | 0.10 |
| 6.30 | 4 | 5504 | 1.34 |
| 6.36 | 4 | 2176 | 0.53 |
| 6.39 | 5 | 10752 | 0.66 |
| 6.45 | 5 | 12288 | 0.75 |
| 6.54 | 6 | 33920 | 0.52 |
| 6.67 | 5 | 8128 | 0.50 |
| 6.79 | 4 | 2816 | 0.69 |
| 6.82 | 5 | 23040 | 1.41 |
| 6.88 | 4 | 768 | 0.81 |
| | 5 | 10240 | |
| 6.91 | 6 | 102400 | 1.56 |
| ⋮ | ⋮ | ⋮ | ⋮ |

(b) 4-DCPFSK (1080 entries)

| $d^2$ | $\zeta$ | $\sum_S \sum_\iota \eta(S,\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 4.00 | 2 | 128 | 0.50 |
| 4.09 | 3 | 1728 | 1.69 |
| 4.15 | 3 | 576 | 0.56 |
| 4.24 | 4 | 1248 | 0.30 |
| 4.49 | 3 | 192 | 0.19 |
| 4.94 | 3 | 576 | 0.56 |
| 4.97 | 5 | 4032 | 0.25 |
| 5.00 | 2 | 256 | 1.94 |
| | 3 | 960 | |
| 5.09 | 4 | 5696 | 1.39 |
| 5.34 | 3 | 64 | 0.06 |
| 5.36 | 3 | 576 | 0.56 |
| 5.43 | 4 | 384 | 0.09 |
| 5.45 | 4 | 5184 | 1.27 |
| 5.54 | 5 | 35424 | 2.16 |
| 5.60 | 5 | 12096 | 0.74 |
| 5.69 | 6 | 46080 | 0.70 |
| 5.79 | 3 | 192 | 0.19 |
| 5.82 | 5 | 10080 | 0.62 |
| 5.85 | 3 | 256 | 0.34 |
| | 4 | 384 | |
| 5.88 | 4 | 2592 | 0.63 |
| 5.94 | 4 | 5728 | 1.66 |
| | 5 | 4224 | |
| 6.00 | 2 | 128 | 1.00 |
| | 3 | 512 | |
| 6.03 | 5 | 2496 | 0.15 |
| 6.09 | 4 | 3456 | 0.84 |
| 6.16 | 5 | 1472 | 0.09 |
| 6.18 | 5 | 15552 | 0.95 |
| 6.21 | 3 | 768 | 0.75 |
| 6.28 | 4 | 1216 | 0.30 |
| 6.30 | 4 | 6912 | 1.69 |
| ⋮ | ⋮ | ⋮ | ⋮ |

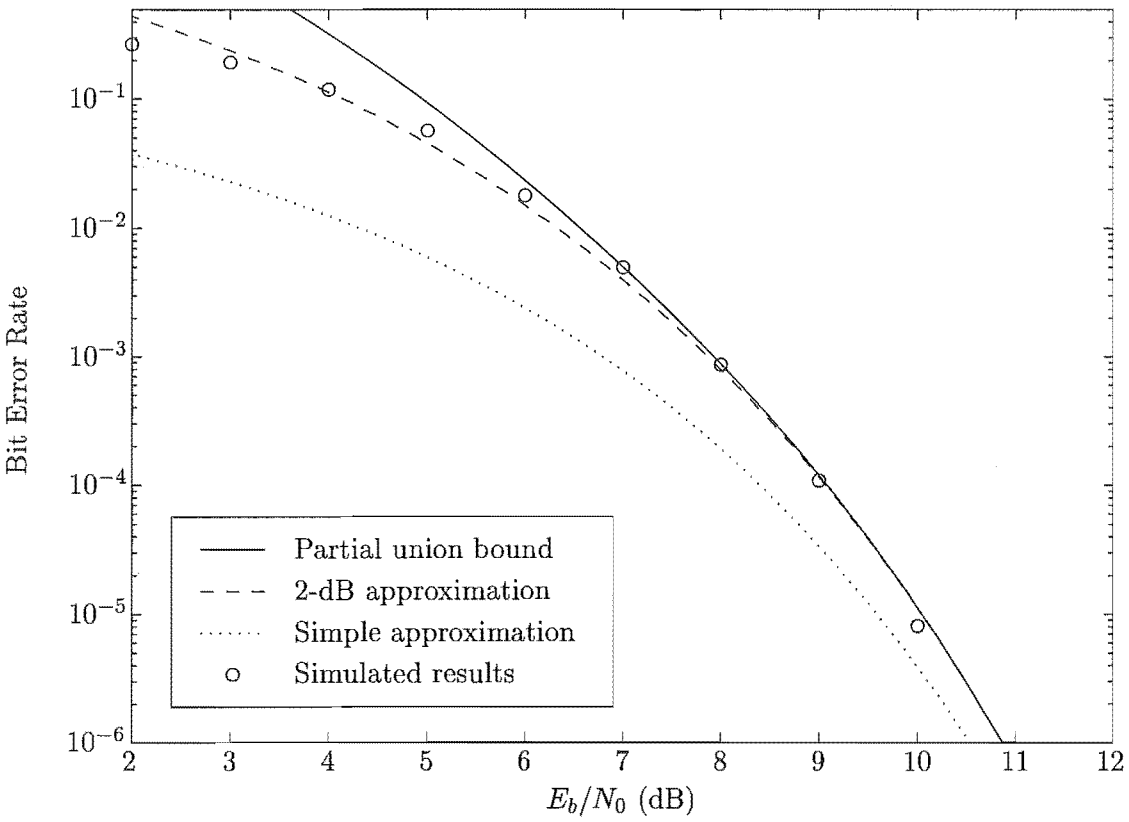



**Figure 5.7:** Computed and simulated performance of 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{2D+1}{D+1}\right]$. The computed approximations and bounds were calculated using (5.18) and Table 5.3(b).

requirements limited the length of the error events to six trellis intervals. The smallest $d^2$ of the unmerged error events is 5.00, is only 1 dB greater than the $d^2_{\text{min}}$ of the system, but again the 2-dB approximation predicts the performance of the system well, as shown in Figure 5.7. The 2-dB approximation uses the 29 terms whose $d^2 \leqslant 6.34$. Again the simple approximation is optimistic.

### 5.4.5 Performance of 8-state, rate-2/3 encoded 8-CPFSK

Table 5.4(a) shows the results of the exhaustive error coefficient search for 8-state, rate-2/3 encoded 8-CPFSK using $\mathbf{G}(D) = \begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}$. Due to the computation requirements, the error events were limited to two trellis intervals. The smallest $d^2$ of the unmerged error events is 1.45, less than the $d^2_{\text{min}}$ of the system, so the computed curves would be expected to be less accurate. However Figure 5.8 shows that the 2-dB approximation still predicts the performance of the system adequately. The 2-dB approximation uses the 9 terms whose $d^2 \leqslant 3.46$.

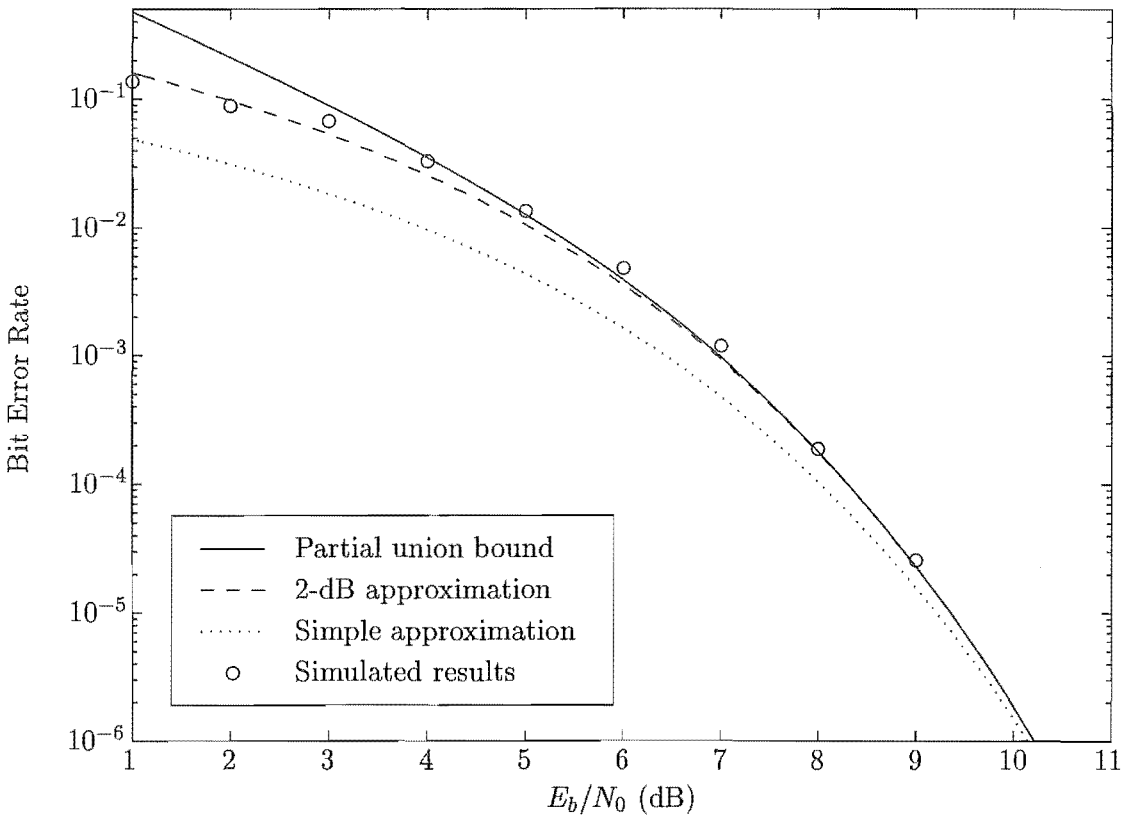



**Figure 5.8:** Computed and simulated performance of 8-state, rate-2/3 encoded 8-CPFSK using $\mathbf{G}(D) = \begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}$. The computed approximations and bounds were calculated using (5.17) and Table 5.4(a).

**Table 5.4**

Error coefficients for 8-state, rate-2/3 encoded 8-CPFSK using $\mathbf{G}(D) = \begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}$

(2 intervals deep, smallest $d^2$ of unmerged error events: 1.58)

(a) 8-CPFSK (1342 entries)

| $d^2$ | $\zeta$ | $\sum_S \sum_\iota \eta(S, \iota, \zeta, d)$ | $\Xi(d)$ |
|---|---|---|---|
| 2.18 | 1 | 1152 | 0.57 |
| | 2 | 37632 | |
| 2.44 | 2 | 451584 | 2.30 |
| 2.56 | 2 | 43200 | 0.22 |
| 2.58 | 2 | 42336 | 0.22 |
| 2.93 | 2 | 33600 | 0.17 |
| 3.03 | 2 | 5376 | 0.03 |
| 3.26 | 2 | 87360 | 0.44 |
| 3.31 | 2 | 95508 | 0.49 |
| 3.45 | 1 | 2304 | 0.75 |
| ⋮ | ⋮ | ⋮ | ⋮ |

(b) 8-DCPFSK (1986 entries)

| $d^2$ | $\zeta$ | $\sum_S \sum_\iota \eta(S, \iota, \zeta, d)$ | $\Xi(d)$ |
|---|---|---|---|
| 2.18 | 1 | 2304 | 0.97 |
| | 2 | 43008 | |
| 2.44 | 2 | 602112 | 3.06 |
| 2.56 | 2 | 96768 | 0.49 |
| 2.58 | 2 | 96768 | 0.49 |
| 2.93 | 2 | 120960 | 0.62 |
| 3.26 | 2 | 152064 | 0.77 |
| 3.31 | 2 | 193536 | 0.98 |
| 3.45 | 1 | 2304 | 0.75 |
| 3.63 | 2 | 120960 | 0.62 |
| ⋮ | ⋮ | ⋮ | ⋮ |

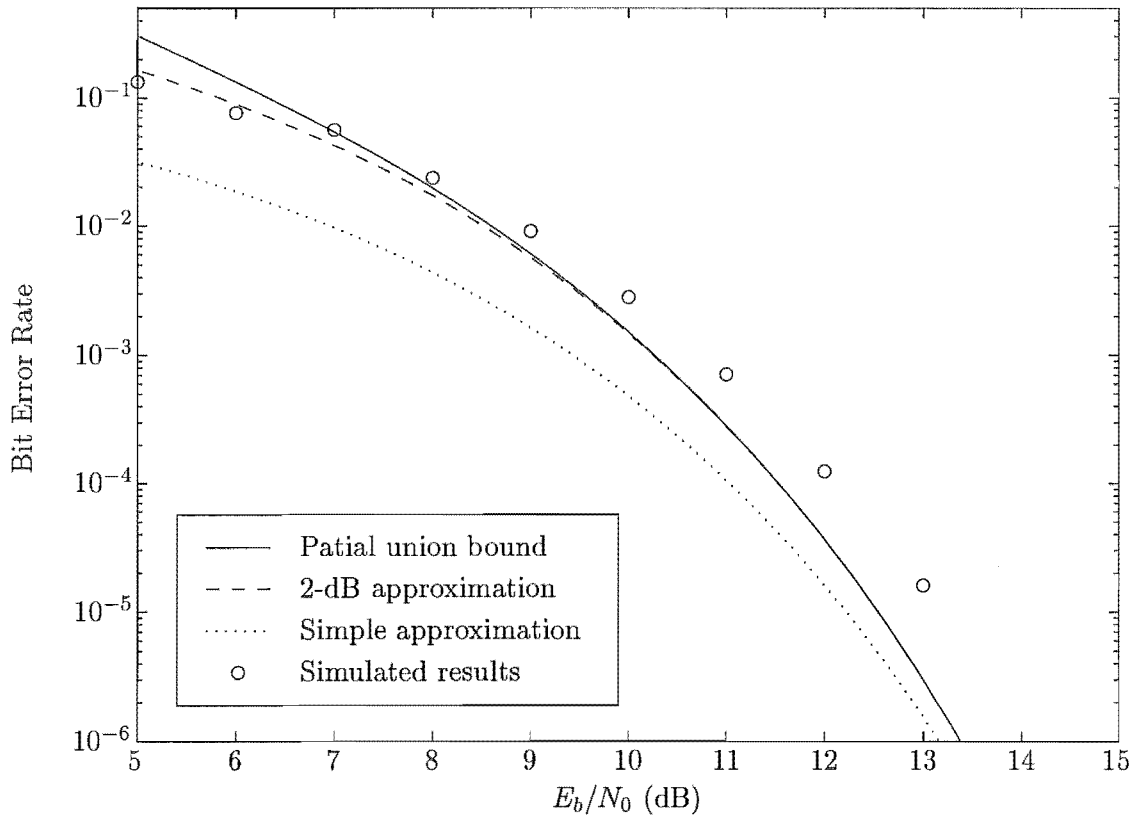



**Figure 5.9:** Computed and simulated performance of 8-state, rate-2/3 encoded 8-DCPFSK using $\mathbf{G}(D) = \begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}$. The computed approximations and bounds were calculated using (5.18) and Table 5.4(b).

### 5.4.6 Performance of 8-state, rate-2/3 encoded 8-DCPFSK

Table 5.4(b) shows the results of the exhaustive error coefficient search for 8-state, rate-2/3 encoded 8-DCPFSK using $\mathbf{G}(D) = \begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}$. As in the coherent case, the computation requirements limited the length of the error events to two trellis intervals, and the smallest $d^2$ of the unmerged error events is 1.45, less than the $d^2_{\text{min}}$ of the system, so the computed curves would be expected to be less accurate. Figure 5.8 shows that this is indeed the case, as none of the approximations predict the system's performance satisfactorily. The 2-dB approximation uses the 8 terms whose $d^2 \leqslant 3.46$.

## 5.5 Performance of Uncoded (D)CPFSK Systems

With a slight modification to (5.16), (5.17) or (5.18) can be used to find the probability of bit error for an uncoded $M$-CPFSK or $M$-DCPFSK scheme. The uncoded error coefficient

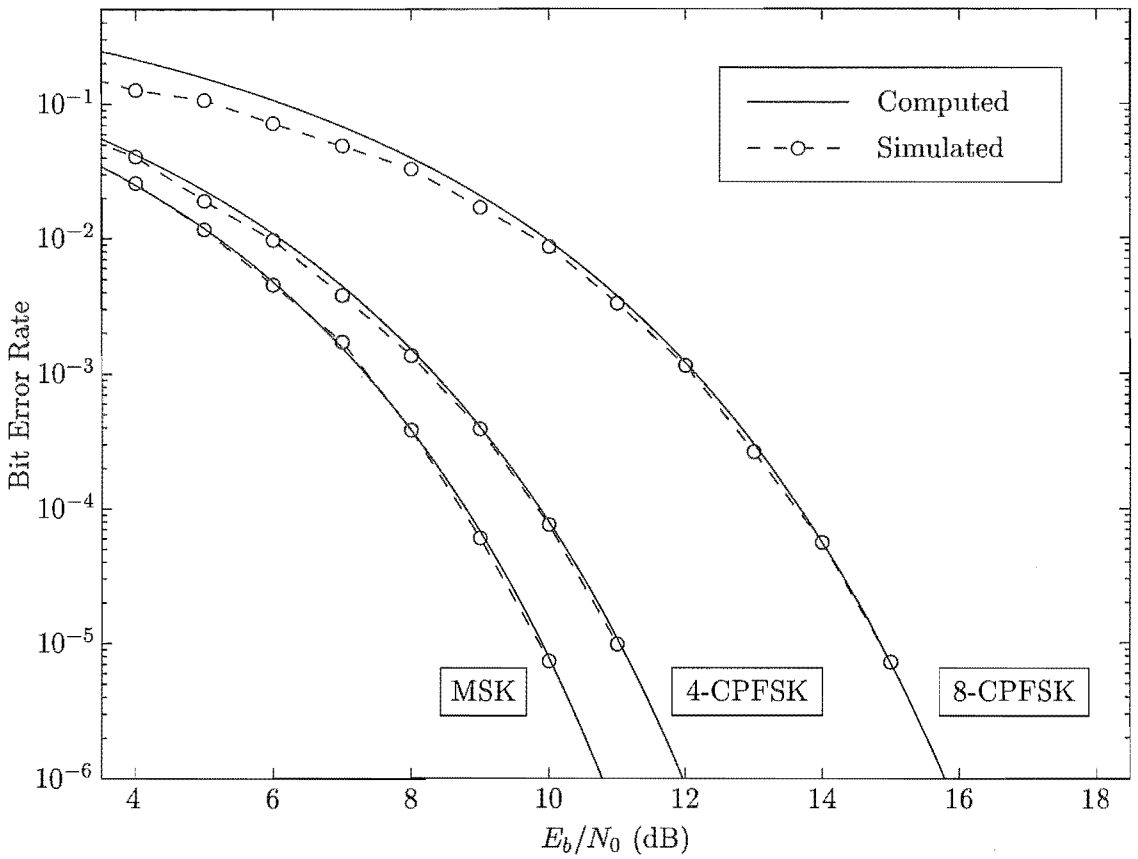



**Figure 5.10:** Computed and simulated performance of various $M$-CPFSK schemes. The 2-dB approximations were calculated using (5.17) and (5.22) and Tables 5.5(a)–5.7(a).

is given by

$$\Xi(d) \triangleq \frac{1}{\log_2 MS_V} \sum_{\mathcal{S}} \sum_{\iota} \sum_{\zeta} \eta(\mathcal{S}, \iota, \zeta, d)\, \iota \left(\frac{1}{M}\right)^{\zeta}, \tag{5.22}$$

which we reformulate into

$$\Xi(d) \triangleq \frac{1}{\log_2 MS_V} \sum_{\zeta} \left(\frac{1}{M}\right)^{\zeta} \sum_{\mathcal{S}} \sum_{\iota} \eta(\mathcal{S}, \iota, \zeta, d)\, \iota, \tag{5.23}$$

for the error coefficient searches.

In Tables 5.5(a)–5.7(a) we present the results of the error coefficient search for uncoded $M$-CPFSK and $M$-CPFSK schemes to a depth of four intervals. All the smallest $d^2$ of the unmerged error events are at least 3 dB greater than the respective $d^2_{\text{min}}$. In Figure 5.10 we present a revamped version of Figure 2.10 where the computed curves are the 2-dB approximations for each scheme. Each curve uses (5.17) and the results in Tables 5.5(a)–5.7(a). For the MSK case, the 2-dB approximation uses the only term whose $d^2 \leqslant 3.16$, for the 4-CPFSK case the 2-dB approximation uses the two terms whose

**Table 5.5**

Error coefficients for uncoded (D)MSK
(4 intervals deep, smallest $d^2$ of unmerged error events - MSK: 6.00, DMSK: 4.00)

(a) MSK

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota}\eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 2.00 | 2 | 16 | 2.00 |
| 4.00 | 3 | 32 | 2.00 |
| 6.00 | 4 | 64 | 2.00 |

(b) DMSK

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota}\eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 2.00 | 2 | 16 | 2.00 |
| 3.00 | 3 | 16 | 1.00 |
| 4.00 | 3 | 16 | 1.50 |
| | 4 | 16 | |
| 5.00 | 4 | 32 | 1.00 |
| 6.00 | 4 | 16 | 0.50 |

**Table 5.6**

Error coefficients for uncoded 4-(D)CPFSK
(4 intervals deep, smallest $d^2$ of unmerged error events: 2.91)

(a) 4-CPFSK (40 entries)

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota}\eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 1.45 | 2 | 144 | 1.12 |
| 2.18 | 3 | 576 | 1.12 |
| 2.91 | 4 | 1728 | 0.84 |
| 3.15 | 2 | 96 | 0.75 |
| 3.45 | 3 | 576 | 1.12 |
| ⋮ | ⋮ | ⋮ | ⋮ |

(b) 4-DCPFSK (46 entries)

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota}\eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 1.45 | 2 | 192 | 1.50 |
| 2.18 | 3 | 1152 | 2.25 |
| 2.91 | 4 | 5184 | 2.53 |
| 3.45 | 3 | 576 | 1.12 |
| 4.00 | 2 | 256 | 2.00 |
| ⋮ | ⋮ | ⋮ | ⋮ |

**Table 5.7**

Error coefficients for uncoded 8-(D)CPFSK
(4 intervals deep, smallest $d^2$ of unmerged error events: 1.20)

(a) 8-CPFSK (803 entries)

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota}\eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 0.60 | 2 | 1568 | 1.02 |
| 0.90 | 3 | 18816 | 1.53 |
| 1.20 | 4 | 169344 | 1.72 |
| 1.48 | 3 | 12544 | 1.02 |
| 1.78 | 4 | 301056 | 3.06 |
| ⋮ | ⋮ | ⋮ | ⋮ |

(b) 8-DCPFSK (624 entries)

| $d^2$ | $\zeta$ | $\sum_{\mathcal{S}}\sum_{\iota}\eta(\mathcal{S},\iota,\zeta,d)$ | $\Xi(d)$ |
|---|---|---|---|
| 0.60 | 2 | 1792 | 1.17 |
| 0.90 | 3 | 25088 | 2.04 |
| 1.20 | 4 | 263424 | 2.68 |
| 1.48 | 3 | 12544 | 1.02 |
| 1.78 | 4 | 351232 | 3.57 |
| ⋮ | ⋮ | ⋮ | ⋮ |

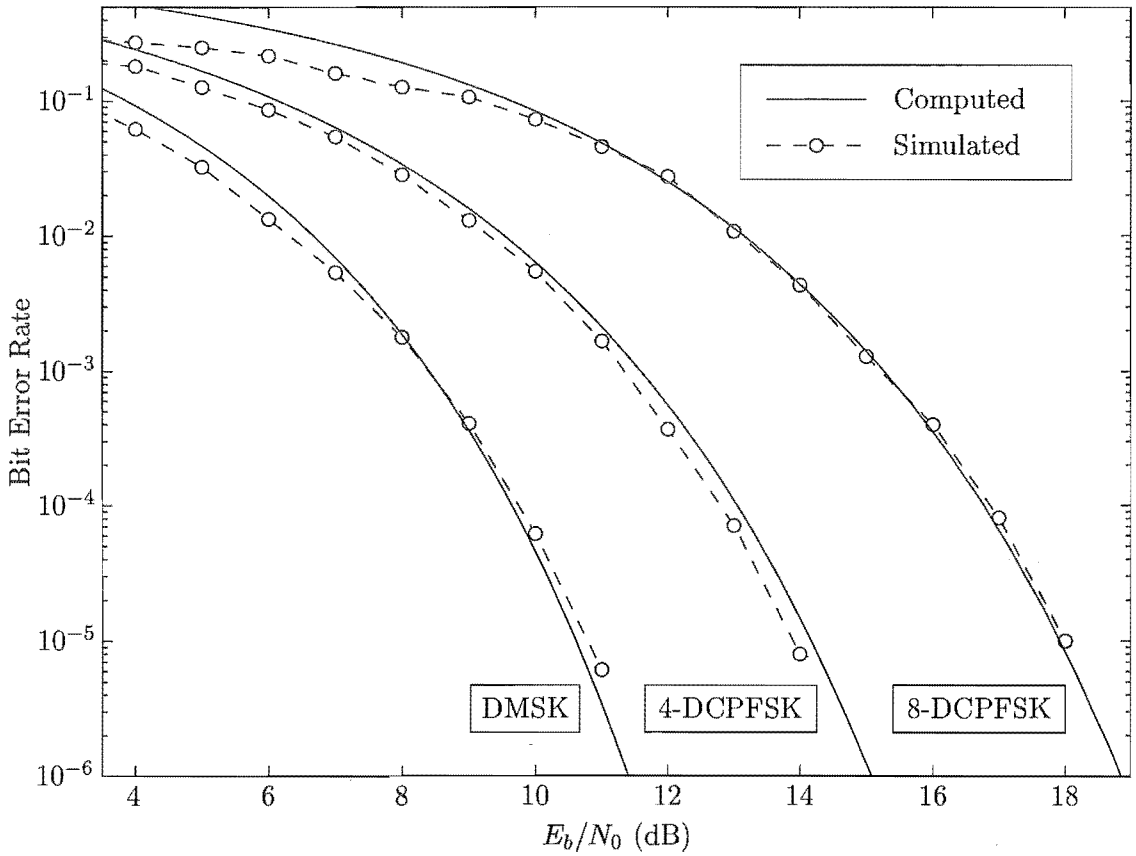



**Figure 5.11:** Computed and simulated performance of various $M$-DCPFSK schemes. The 2-dB approximations were calculated using (5.24), (5.18) and (5.22) and Tables 5.5(b)–5.7(b).

$d^2 \leqslant 2.30$ and the 2-dB approximation for the 8-CPFSK case uses the two terms whose $d^2 \leqslant 0.95$. The 2-dB approximations agree very well with the simulation results.

Tables 5.5(b)–5.7(b) list the results of the error coefficient search for uncoded $M$-DCPFSK and $M$-DCPFSK schemes to a depth of four intervals. The smallest $d^2$ of the unmerged error events are 3 dB greater than the respective $d^2_{\text{min}}$. In Figure 5.11 we present a revamped version of Figure 3.12 where the computed curves are the 2-dB approximations for each scheme. The DMSK case, the 2-dB approximation uses the two terms of 5.5(b) whose $d^2 \leqslant 3.16$, and a bound based on (3.113), given by

$$P_{e,\text{DMSK}} \leqslant \frac{1}{2} \sum_{d \in \mathcal{D}} \Xi(d) \exp\left(-\frac{E_b}{N_0}\right). \tag{5.24}$$

The other two cases use (5.18) and the results in Tables 5.6(b) and 5.7(b). For the 4-DCPFSK case the 2-dB approximation uses the two terms whose $d^2 \leqslant 2.30$ and the 2-dB approximation for the 8-DCPFSK case uses the two terms whose $d^2 \leqslant 0.95$. The 2-dB approximations agree reasonably well with the simulation results.

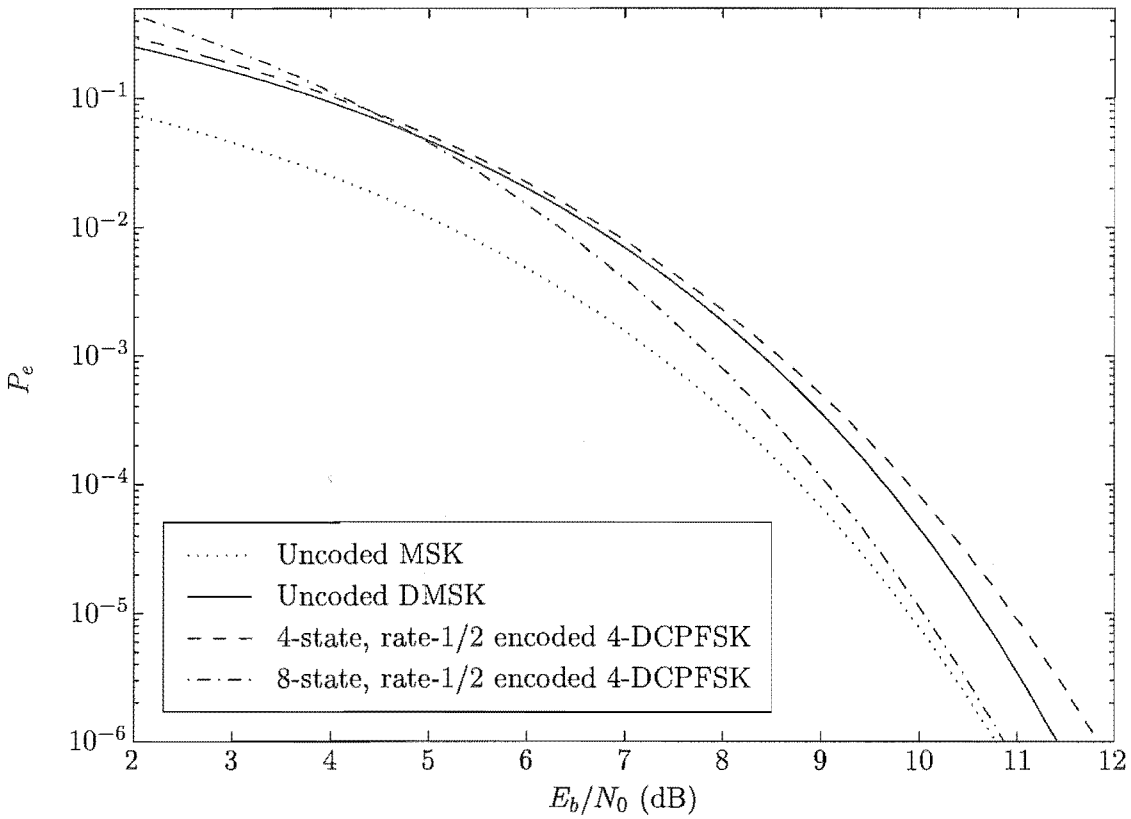



**Figure 5.12:** Computed 2-dB approximations on the probability of bit error for of uncoded MSK, uncoded DMSK, and rate-1/2 coded 4-DCPFSK in AWGN. The 4-state code has $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$, and the 8-state code has $\mathbf{G}(D) = \left[1 \ \frac{2D+1}{D+1}\right]$.

## 5.6 Comparison of Uncoded DMSK with Rate-1/2 Encoded 4-DCPFSK

Having established the agreement between the simulation results and the 2-dB approximations, we can now compare systems using only their 2-dB approximations. Uncoded DMSK and rate-1/2 encoded 4-DCPFSK are comparable system in that they have the same information throughput. The computed 2-dB approximations for uncoded DMSK and two rate-1/2 encoded 4-DCPFSK systems are shown in Figure 5.12, along with that of uncoded coherently-demodulated MSK. The 4-state code does not perform quite as well as uncoded DMSK, but the 8-state code shows a 0.5 dB gain, and performs within 0.1 dB of uncoded MSK. This is significant, as 8-state, rate-1/2 encoded 4-DCPFSK is not much more complicated than uncoded DMSK, and could be considered less complicated than uncoded MSK as the 4-DCPFSK does not require any synchronisation other than symbol timing. Obviously if one of the more powerful codes in Table 4.1 were used, the

performance of rate-1/2 encoded 4-DCPFSK would be even better. For example, a rate-1/2 encoded 4-DCPFSK system using $\mathbf{G}(D) = \left[1 \quad \frac{D^2+D+2}{2D^3+D^2+3D+1}\right]$ should perform around 3 dB better than uncoded DMSK and over 2 dB better than uncoded MSK at a BER of $10^{-6}$.

## 5.7 Summary

In this chapter we have explored the computed performance of rate-$(\log_2 M - 1)/\log_2$ encoded $M$-CPFSK and $M$-DCPFSK systems. By following a similar method to [Yan94], we developed an upper bound on the probability of bit error, consisting of a sum of distance terms and error coefficients. Calculating the error coefficients is a computationally-intensive task, and we chose to limit the length of the error events we would consider and then perform an exhaustive search to determine the distance terms and their associated error coefficients. It was also found to be important to record the smallest squared Euclidean distance of the unmerged error events, as this can indicate whether or not the significant terms have been found.

We compared simulation results with a bound calculated using the error coefficients whose distance terms were within 2-dB of the $d^2_{\text{min}}$ of the system, which we called the "2-dB approximation". In all but one case, the 2-dB approximation agreed very well with simulation results. The rate-2/3 encoded 8-DCPFSK case—encoded using $\mathbf{G}(D) = \begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}$—did not agree very well, but this can be explained by the fact that the computational requirements meant that not all the significant distance terms and associated error coefficients were calculated. Another important result to come out of this chapter was that the simple performance approximations used in Chapters 2 and 3 provide a reasonable indication of the coded system's performance.

We also showed the improvement available through using rate-1/2 encoded 4-DCPFSK over both uncoded DMSK and uncoded coherently-demodulated MSK. The improvement against uncoded coherently-demodulated MSK is perhaps more interesting as the coded 4-DCPFSK system uses more DSP processing power to achieve its performance but does not need the complicated phase and frequency synchronisation circuitry and algorithms required for the coherent demodulation used in uncoded MSK.

# Chapter 6

# Performance of DCPFSK Systems in Rayleigh Flat Fading

## 6.1 Introduction

Due to the fact that a transmitted signal may travel many different paths on its way to a receiver, the received signal in many communication systems is corrupted by a phenomenon known as fading [Jak74]. Typically this results in the received signal signal undergoing quasi-periodic deep nulls and rapid fade changes. A very important fading channel is the Rayleigh flat-fading channel. The term Rayleigh is used to describe the phenomenon that occurs when there is no direct path between transmitter and receiver, and the term flat-fading further specifies that the fading is not frequency-selective, that is, every frequency across the band of interest experiences the same fading. Another important parameter is the maximum Doppler shift $f_D$ [Jak74], which determines the spectral properties of the fading process, and thus its autocorrelation. Often the maximum Doppler shift is normalised to the symbol period $T$ and the expression $f_DT$ is used. A fading process with $f_DT = 0.001$ is said to be "slow" and one with $f_DT = 0.1$ is said to be "fast". These channels can make coherent demodulation very difficult, as carrier recovery becomes a complex problem. Differential demodulation avoids this problem, as its very nature eliminates the need for carrier recovery.

In this chapter we investigate the performance of differentially encoded and demodulated CPFSK (DCPFSK) systems, with and without external error-correcting coding. In Section 6.2 we develop the differentially-demodulated received signal that has been perturbed by Rayleigh flat fading. From there we analyse the pairwise probability of error of $M$-DCPFSK in Rayleigh flat fading in Section 6.3. This allows us to calculate the

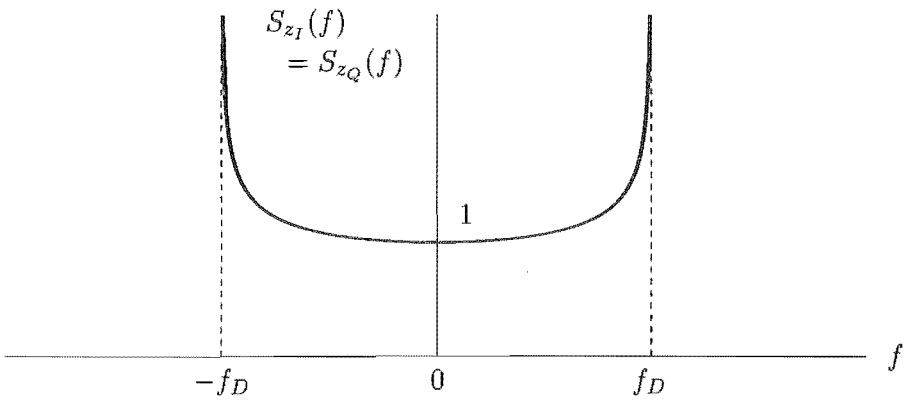



**Figure 6.1:** Power spectral densities of the Rayleigh fading components

performance of uncoded $M$-DCPFSK and coded $M$-DCPFSK systems in Section 6.4 and Section 6.5 respectively. We then compare two systems to determine the improvements available through coding in Section 6.6. Finally, we present a summary of the developments of the chapter in Section 6.7.

## 6.2 The Differentially-Demodulated Received Signal

Let the narrow-band Rayleigh fading process be given by [Jak74]

$$z(t) \triangleq z_I(t)\cos(2\pi f_c t) - z_Q(t)\sin(2\pi f_c t), \tag{6.1}$$

where $z_I(t)$ and $z_Q(t)$ are zero-mean Gaussian processes with power spectral densities (psd's) shown in Figure 6.1 and given by

$$S_{z_I}(f) = S_{z_Q}(f) = \left[1 - \left(\frac{f}{f_D}\right)^2\right]^{-1/4}. \tag{6.2}$$

We can write the received signal in Rayleigh flat fading as

$$r(t,\beta) \triangleq z(t)\, s(t,\beta) + w(t), \tag{6.3}$$

where $s(t,\beta)$ was defined in (3.14), and $w(t)$ is described by (2.30) and (2.31). Let us define the complex envelope of $z(t)$ as

$$\tilde{z}(t) \triangleq z_I(t) + j\, z_Q(t), \tag{6.4}$$

Using (6.4) and the definitions of $\tilde{s}(t,\beta)$ and $\tilde{w}(t)$ given in (2.35) and (2.36) respectively, we can define the complex envelope of $r(t,\beta)$ as

$$\tilde{r}(t,\beta) \triangleq \tilde{z}(t)\, \tilde{s}(t,\beta) + \tilde{w}(t), \tag{6.5}$$

and write (6.3) as

$$r(t,\beta) \;=\; \mathrm{Re}\left[\tilde{r}(t,\beta)\exp(j2\pi f_c t)\right]. \tag{6.6}$$

We can then follow a similar development to that in Section 3.2 to obtain the differentially-demodulated received signal $y_{\mathrm{d}}(t,\beta)$ given by

$$y_{\mathrm{d}}(t,\beta) \;=\; \frac{1}{2}\sqrt{\frac{T}{E}}\,\tilde{r}(t,\beta)\,\tilde{r}^*(t-T,\beta). \tag{6.7}$$

As the following analysis of the pairwise probability of error requires sampled signals, we use the sampling receiver described in Appendix F. It should be noted that this receiver is just a sampling version of the receiver for DCPFSK in additive white Gaussian noise (AWGN) presented in Section 3.8. Thus we assume that the differential demodulator accounts for the fading, and no additional circuitry or processing is used to alleviate its effects.

In the sampling receiver, the output of the differential demodulator is sampled at $r_s$ Hz. The sample period $T_s$ is equal to $1/r_s$. To ensure that we have an integral number of samples per symbol, $T_s$ is chosen such that

$$\frac{T}{T_s} \;=\; D_s, \tag{6.8}$$

where $D_s$ is a positive integer. The sampled version of $y_{\mathrm{d}}(t,\beta)$ is $y_{\mathrm{d}}(iT_s,\beta)$, given by

$$\begin{aligned} y_{\mathrm{d}}(iT_s,\beta) \;&\triangleq\; \frac{1}{2}\sqrt{\frac{T}{E}}\,\tilde{r}(iT_s,\beta)\,\tilde{r}^*(iT_s-T,\beta) \\ &=\; \frac{1}{2}\sqrt{\frac{T}{E}}\,\left[\tilde{z}(iT_s)\,\tilde{s}(iT_s,\beta)+\tilde{w}(iT_s)\right] \\ &\quad\times\left[\tilde{z}^*(iT_s-T)\,\tilde{s}^*(iT_s-T,\beta)+\tilde{w}^*(iT_s-T)\right]. \end{aligned} \tag{6.9}$$

This signal is then processed by the Viterbi processor described in Appendix F.

## 6.3 Pairwise Probability of Error of $M$-DCPFSK in Rayleigh Flat Fading

The pairwise probability of error is the probability that the receiver will decide that a particular *incorrect* data sequence has been sent instead of the actually transmitted data sequence. Let the transmitted data be $\beta$ and the receiver's estimate of $\beta$ be $\hat{\beta}$. As we require sampled signals for our analysis, we now introduce two instances of a discrete metric that operates over the entire received signal. The first is based on the mean squared

error (MSE) between the samples of the demodulated received signal $y_{\mathrm{d}}(iT_s, \beta)$ and the samples of the complex envelope of a noise-free version of the demodulated transmitted signal $\dot{y}_{\mathrm{d}}(iT_s, \beta)$,

$$\Upsilon(y_{\mathrm{d}}, \beta) \;=\; \sum_{i=-\infty}^{\infty} \left| y_{\mathrm{d}}(iT_s, \beta) - \dot{y}_{\mathrm{d}}(iT_s, \beta) \right|^2, \tag{6.10}$$

where $\dot{y}_{\mathrm{d}}(iT_s, \beta)$ is defined as

$$\dot{y}_{\mathrm{d}}(iT_s, \beta) \;\triangleq\; \frac{1}{2}\sqrt{\frac{T}{E}}\, \tilde{s}(iT_s, \beta)\, \tilde{s}^*(iT_s - T, \beta). \tag{6.11}$$

Note that the MSE is the sampled analogue of squared Euclidean distance. Due to the effects of the differential noise and fading which has only been partially accounted for, this metric is still not optimal, but we are seeking to determine the performance of our DCPFSK AWGN receiver in Rayleigh flat fading.

The second instance of the metric is based on the MSE between $y_{\mathrm{d}}(iT_s, \beta)$ and some other demodulated possibly-transmitted signal $\dot{y}_{\mathrm{d}}(iT_s, \hat{\beta})$, and is defined as

$$\Upsilon(y_{\mathrm{d}}, \hat{\beta}) \;=\; \sum_{i=-\infty}^{\infty} \left| y_{\mathrm{d}}(iT_s, \beta) - \dot{y}_{\mathrm{d}}(iT_s, \hat{\beta}) \right|^2. \tag{6.12}$$

As the Viterbi processor in the receiver chooses the metric with the smallest value, an error will occur if

$$\Upsilon(y_{\mathrm{d}}, \beta) \;>\; \Upsilon(y_{\mathrm{d}}, \hat{\beta}). \tag{6.13}$$

That is, the distance between the received signal $y_{\mathrm{d}}(iT_s, \beta)$ and a reference signal based on incorrect data $\hat{\beta}$ is less than that between $y_{\mathrm{d}}(iT_s, \beta)$ and the reference signal based on the actual transmitted data $\beta$. We are interested in the probability of this occurrence, which is given by

$$\begin{aligned} \Pr\left\{\hat{\beta} \neq \beta\right\} \;&=\; \Pr\left\{\Upsilon(y_{\mathrm{d}}, \beta) \;>\; \Upsilon(y_{\mathrm{d}}, \hat{\beta})\right\} \\ &=\; \Pr\left\{\Upsilon(y_{\mathrm{d}}, \beta) - \Upsilon(y_{\mathrm{d}}, \hat{\beta}) > 0\right\} \\ &=\; \Pr\left\{\Upsilon_e > 0\right\}, \end{aligned} \tag{6.14}$$

where we have defined

$$\Upsilon_e \;\triangleq\; \Upsilon(y_{\mathrm{d}}, \beta) - \Upsilon(y_{\mathrm{d}}, \hat{\beta}). \tag{6.15}$$

If we can determine $p(\Upsilon_e)$, the pdf of $\Upsilon_e$, then the pairwise probability of error can be calculated by integrating $p(\Upsilon_e)$ from 0 to $\infty$,

$$\Pr\left\{\Upsilon_e > 0\right\} \;=\; \int_0^{\infty} p(\Upsilon_e)\, d\Upsilon_e. \tag{6.16}$$

In this case, it is simpler to find $G_{\Upsilon_e}(\xi)$, the characteristic function of $\Upsilon_e$, which is related to $p(\Upsilon_e)$ as

$$G_{\Upsilon_e}(\xi) \;=\; \int_{-\infty}^{\infty} p(\Upsilon_e)\, e^{j\xi\Upsilon_e}\, d\Upsilon_e. \tag{6.17}$$

Thus $p(\Upsilon_e)$ can be written as

$$p(\Upsilon_e) \;=\; \frac{1}{2\pi}\int_{-\infty}^{\infty} G_{\Upsilon_e}(\xi)\, e^{-j\xi\Upsilon_e}\, d\xi, \tag{6.18}$$

and substituting (6.18) in (6.16) we obtain

$$\begin{aligned}\Pr\{\Upsilon_e > 0\} \;&=\; \int_0^{\infty} \frac{1}{2\pi}\int_{-\infty}^{\infty} G_{\Upsilon_e}(\xi)\, e^{-j\xi\Upsilon_e} d\xi\; d\Upsilon_e \\ &=\; \frac{1}{2\pi}\int_{-\infty}^{\infty} G_{\Upsilon_e}(\xi) \int_0^{\infty} e^{-j\xi\Upsilon_e} d\Upsilon_e\; d\xi. \end{aligned} \tag{6.19}$$

Using the unit-step function $u(x)$, defined as

$$u(x) \;\triangleq\; \begin{cases} 0, & x < 0 \\ 1, & x \geqslant 0, \end{cases} \tag{6.20}$$

we can write the inner integral in (6.19) as

$$\int_0^{\infty} e^{-j\xi\Upsilon_e} d\Upsilon_e \;=\; \int_{-\infty}^{\infty} u(\Upsilon_e)\, e^{-j\xi\Upsilon_e}\, d\Upsilon_e. \tag{6.21}$$

The RHS of (6.21) is the Fourier transform of $u(\Upsilon_e)$ which is equal to [Hay83]

$$\int_{-\infty}^{\infty} u(\Upsilon_e)\, e^{-j\xi\Upsilon_e}\, d\Upsilon_e \;=\; \frac{\delta(\frac{\xi}{2\pi})}{2} + \frac{1}{j\xi}, \tag{6.22}$$

where $\delta(\cdot)$ is the delta function, with the properties

$$\int_{-\infty}^{\infty} \delta(x)\, dx \;=\; 1 \tag{6.23}$$

$$\int_{-\infty}^{\infty} f(x)\, \delta\Big(\frac{x}{k}\Big)\, dx \;=\; k\, f(0) \tag{6.24}$$

where $f(x)$ is some function of $x$. Using (6.21)–(6.24) in (6.19), we obtain

$$\begin{aligned}\Pr\{\Upsilon_e > 0\} \;&=\; \frac{1}{2\pi}\int_{-\infty}^{\infty} G_{\Upsilon_e}(\xi) \left(\frac{\delta(\frac{\xi}{2\pi})}{2} + \frac{1}{j\xi}\right) d\xi \\ &=\; \frac{G_{\Upsilon_e}(0)}{2} + \frac{1}{2\pi j}\int_{-\infty}^{\infty} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi. \end{aligned} \tag{6.25}$$

We now need to determine the characteristic function $G_{\Upsilon_e}(\xi)$.

### 6.3.1 Form of the Characteristic Function

We note that we can develop (6.10) as

$$\begin{aligned}\Upsilon(y_{\mathrm{d}},\boldsymbol{\beta}) &= \sum_{i=-\infty}^{\infty} |y_{\mathrm{d}}(iT_s,\boldsymbol{\beta}) - \dot{y}_{\mathrm{d}}(iT_s,\boldsymbol{\beta})|^2 \\ &= \sum_{i=-\infty}^{\infty} \left\{ |y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})|^2 + |\dot{y}_{\mathrm{d}}(iT_s,\boldsymbol{\beta})|^2 - 2\,\mathrm{Re}\left[y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\,\dot{y}_{\mathrm{d}}^*(iT_s,\boldsymbol{\beta})\right] \right\} \\ &= \sum_{i=-\infty}^{\infty} |y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})|^2 + \sum_{i=-\infty}^{\infty} |\dot{y}_{\mathrm{d}}(iT_s,\boldsymbol{\beta})|^2 - 2\sum_{i=-\infty}^{\infty} \mathrm{Re}\left[y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\,\dot{y}_{\mathrm{d}}^*(iT_s,\boldsymbol{\beta})\right],\end{aligned} \tag{6.26}$$

and similarly, (6.12) can be re-written as

$$\Upsilon(y_{\mathrm{d}},\hat{\boldsymbol{\beta}}) = \sum_{i=-\infty}^{\infty} |y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})|^2 + \sum_{i=-\infty}^{\infty} \left|\dot{y}_{\mathrm{d}}(iT_s,\hat{\boldsymbol{\beta}})\right|^2 - 2\sum_{i=-\infty}^{\infty} \mathrm{Re}\left[y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\,\dot{y}_{\mathrm{d}}^*(iT_s,\hat{\boldsymbol{\beta}})\right]. \tag{6.27}$$

Substituting (6.26) and (6.27) into (6.15), and noting that CPFSK signals are constant-envelope so that

$$|\dot{y}_{\mathrm{d}}(iT_s,\boldsymbol{\beta})|^2 = \left|\dot{y}_{\mathrm{d}}(iT_s,\hat{\boldsymbol{\beta}})\right|^2, \tag{6.28}$$

we have

$$\begin{aligned}\Upsilon_e &= 2\sum_{i=-\infty}^{\infty} \mathrm{Re}\left[y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\,\dot{y}_{\mathrm{d}}^*(iT_s,\hat{\boldsymbol{\beta}})\right] - 2\sum_{i=-\infty}^{\infty} \mathrm{Re}\left[y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\,\dot{y}_{\mathrm{d}}^*(iT_s,\boldsymbol{\beta})\right] \\ &= 2\sum_{i=-\infty}^{\infty} \mathrm{Re}\left[y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\{\dot{y}_{\mathrm{d}}^*(iT_s,\hat{\boldsymbol{\beta}}) - \dot{y}_{\mathrm{d}}^*(iT_s,\boldsymbol{\beta})\}\right] \\ &= \sum_{i=-\infty}^{\infty} \left[y_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\{\dot{y}_{\mathrm{d}}^*(iT_s,\hat{\boldsymbol{\beta}}) - \dot{y}_{\mathrm{d}}^*(iT_s,\boldsymbol{\beta})\} + y_{\mathrm{d}}^*(iT_s,\boldsymbol{\beta})\{\dot{y}_{\mathrm{d}}(iT_s,\hat{\boldsymbol{\beta}}) - \dot{y}_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\}\right] \\ &= \frac{1}{2}\sqrt{\frac{T}{E}} \sum_{i=-\infty}^{\infty} \Big[\tilde{r}(iT_s,\boldsymbol{\beta})\,\tilde{r}^*(iT_s - T,\boldsymbol{\beta})\{\dot{y}_{\mathrm{d}}^*(iT_s,\hat{\boldsymbol{\beta}}) - \dot{y}_{\mathrm{d}}^*(iT_s,\boldsymbol{\beta})\} \\ &\qquad + \tilde{r}^*(iT_s - T,\boldsymbol{\beta})\,\tilde{r}(iT_s,\boldsymbol{\beta})\{\dot{y}_{\mathrm{d}}(iT_s,\hat{\boldsymbol{\beta}}) - \dot{y}_{\mathrm{d}}(iT_s,\boldsymbol{\beta})\}\Big]\end{aligned} \tag{6.29}$$

Let us now suppose that the signals are finite in length (say $N_s$ samples long, where $N_s$ is a multiple of $D_s$), so that we can define the vector $\tilde{\boldsymbol{r}}$ as

$$\tilde{\boldsymbol{r}} \triangleq \begin{bmatrix} \tilde{r}(T_s,\boldsymbol{\beta}) \\ \vdots \\ \tilde{r}(iT_s,\boldsymbol{\beta}) \\ \vdots \\ \tilde{r}(N_sT_s,\boldsymbol{\beta}) \end{bmatrix} = \tilde{\boldsymbol{z}} \otimes \tilde{\boldsymbol{s}} + \tilde{\boldsymbol{w}}, \tag{6.30}$$

where

$$\tilde{\boldsymbol{z}} \triangleq \begin{bmatrix} \tilde{z}(T_s) \\ \vdots \\ \tilde{z}(iT_s) \\ \vdots \\ \tilde{z}(N_sT_s+T) \end{bmatrix} \quad \tilde{\boldsymbol{s}} \triangleq \begin{bmatrix} \tilde{s}(T_s,\beta) \\ \vdots \\ \tilde{s}(iT_s,\beta) \\ \vdots \\ \tilde{s}(N_sT_s,\beta) \end{bmatrix} \quad \tilde{\boldsymbol{w}} \triangleq \begin{bmatrix} \tilde{w}(T_s,\beta) \\ \vdots \\ \tilde{w}(iT_s,\beta) \\ \vdots \\ \tilde{w}(N_sT_s,\beta) \end{bmatrix}, \tag{6.31}$$

and the $\otimes$ represents element-by-element multiplication, known as the Kronecker product. Equations (6.30) and (6.31) allow us to write (6.29) in matrix form as

$$\Upsilon_e \;=\; \tilde{\boldsymbol{r}}^H \,\mathbf{A}\, \tilde{\boldsymbol{r}}, \tag{6.32}$$

where $\boldsymbol{x}^H$ represents the conjugate-transpose of $\boldsymbol{x}$, and $\mathbf{A}$ is a $N_s$-by-$N_s$ matrix that scales the required products, defined as

$$\mathbf{A} \triangleq \begin{bmatrix} 0 & \cdots & 0 & A_{D_s+1} & 0 & & \cdots & & 0 \\ \vdots & & & 0 & A_{D_s+2} & \ddots & & & \\ 0 & & & & 0 & \ddots & 0 & & \vdots \\ A^*_{D_s+1} & 0 & & & & \ddots & A_i & \ddots & \\ 0 & A^*_{D_s+2} & 0 & & & & 0 & \ddots & 0 \\ & \ddots & \ddots & \ddots & & & & \ddots & A_{N_s} \\ \vdots & & 0 & A^*_i & 0 & & & & 0 \\ & & & \ddots & \ddots & \ddots & & & \vdots \\ 0 & & \cdots & & 0 & A^*_{N_s} & 0 & \cdots & 0 \end{bmatrix} \tag{6.33}$$

where

$$A_i \;\triangleq\; \frac{1}{2}\sqrt{\frac{T}{E}}\, \dot{y}^*_{\mathrm{d}}(iT_s,\hat{\beta}) - \dot{y}^*_{\mathrm{d}}(iT_s,\beta). \tag{6.34}$$

Obviously, $\mathbf{A}$ is Hermitian. Equation (6.32) is a Gaussian quadratic form, whose characteristic function is given by [SBS66]

$$G_{\Upsilon_e}(\xi) \;=\; \frac{\exp\left\{ j\xi\, \mathrm{E}\left[\tilde{\boldsymbol{r}}\right]^H \left(\mathbf{A}^{-1} - 2j\xi \mathbf{R}^*_{\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}}\right)^{-1} \mathrm{E}\left[\tilde{\boldsymbol{r}}\right] \right\}}{\det\left(\mathbf{I}_{N_s} - 2j\xi \mathbf{R}^*_{\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}}\mathbf{A}\right)}, \tag{6.35}$$

where $\mathbf{I}_{N_s}$ is the $N_s \times N_s$ identity matrix, and $\mathbf{R}_{\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}}$ is the auto-covariance matrix of the vector $\tilde{\boldsymbol{r}}$, defined as

$$\mathbf{R}_{\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}} \;=\; \frac{1}{2}\,\mathrm{E}\left[ \left(\tilde{\boldsymbol{r}} - \mathrm{E}\left[\tilde{\boldsymbol{r}}\right]\right)^* \left(\tilde{\boldsymbol{r}} - \mathrm{E}\left[\tilde{\boldsymbol{r}}\right]\right)^\dagger \right]. \tag{6.36}$$

The expected value of $\tilde{\boldsymbol{r}}$ is

$$\mathrm{E}[\tilde{\boldsymbol{r}}] \;=\; \mathrm{E}[\tilde{\boldsymbol{z}} \otimes \tilde{\boldsymbol{s}} + \tilde{\boldsymbol{w}}] \;=\; \tilde{\boldsymbol{s}} \otimes \mathrm{E}[\tilde{\boldsymbol{z}}] + \mathrm{E}[\tilde{\boldsymbol{w}}] \;=\; \mathbf{0}, \tag{6.37}$$

as both $\tilde{\boldsymbol{z}}$ and $\tilde{\boldsymbol{w}}$ are zero-mean vectors. Applying (6.37) to (6.36), we obtain

$$\begin{aligned}
\mathbf{R}_{\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}} &= \frac{1}{2}\,\mathrm{E}\left[\tilde{\boldsymbol{r}}^{*}\tilde{\boldsymbol{r}}^{\dagger}\right] \\
&= \frac{1}{2}\,\mathrm{E}\left[(\tilde{\boldsymbol{z}} \otimes \tilde{\boldsymbol{s}} + \tilde{\boldsymbol{w}})^{*}(\tilde{\boldsymbol{z}} \otimes \tilde{\boldsymbol{s}} + \tilde{\boldsymbol{w}})^{\dagger}\right] && (6.38)\\
&= \frac{1}{2}\,\mathrm{E}\left[(\tilde{\boldsymbol{z}} \otimes \tilde{\boldsymbol{s}})^{*}(\tilde{\boldsymbol{z}} \otimes \tilde{\boldsymbol{s}})^{\dagger}\right] + \frac{1}{2}\,\mathrm{E}\left[\tilde{\boldsymbol{w}}^{*}\tilde{\boldsymbol{w}}^{\dagger}\right] && (6.39)\\
&= \mathbf{R}_{\tilde{\boldsymbol{z}}\tilde{\boldsymbol{s}}\,\tilde{\boldsymbol{z}}\tilde{\boldsymbol{s}}} + \mathbf{R}_{\tilde{\boldsymbol{w}}\tilde{\boldsymbol{w}}}, && (6.40)
\end{aligned}$$

as we assume that $\tilde{z}(iT_s)$ and $\tilde{w}(iT_s)$ are uncorrelated. The element in the $i$-th row and the $k$-th column of $\mathbf{R}_{\tilde{\boldsymbol{z}}\tilde{\boldsymbol{s}}\,\tilde{\boldsymbol{z}}\tilde{\boldsymbol{s}}}$ is

$$\frac{1}{2}\,\mathrm{E}\left[\tilde{z}^{*}(iT_s)\,\tilde{s}^{*}(iT_s,\beta)\,\tilde{z}(kT_s)\,\tilde{s}(kT_s,\beta)\right] \;=\; \frac{1}{2}\,\tilde{s}(iT_s,\beta)\,\tilde{s}(kT_s,\beta)\,\mathrm{E}\left[\tilde{z}(iT_s)\,\tilde{z}(kT_s)\right]. \tag{6.41}$$

Assuming isotropic scattering, the autocorrelation of fading samples is given by [Jak74]

$$\mathrm{E}\left[\tilde{z}^{*}(iT_s)\,\tilde{z}^{*}(kT_s)\right] \;=\; J_0(2\pi f_D T\,|i-k|\,T_s)\,, \tag{6.42}$$

where $J_0(\cdot)$ is the zeroth order Bessel function. Thus the element in the $i$-th row and the $k$-th column of $\mathbf{R}_{\tilde{\boldsymbol{z}}\tilde{\boldsymbol{s}}\,\tilde{\boldsymbol{z}}\tilde{\boldsymbol{s}}}$ is

$$\frac{1}{2}\,\tilde{s}(iT_s,\beta)\,\tilde{s}(kT_s,\beta)\,J_0(2\pi f_D T\,|i-k|\,T_s)\,. \tag{6.43}$$

As $\tilde{\boldsymbol{r}}$ is zero-mean, we can reduce (6.35) to the much simpler form

$$G_{\Upsilon_e}(\xi) \;=\; \frac{1}{\det\left(\mathbf{I}_{N_s} - 2j\xi\mathbf{R}^{*}_{\tilde{\boldsymbol{r}}\tilde{\boldsymbol{r}}}\mathbf{A}\right)}. \tag{6.44}$$

As all the matrices in (6.44) are $N_s$-by-$N_s$, the determinant will be a $N_s$-th order polynomial in $\xi$. If we use $\xi_i$ to denote the $i$-th pole of $G_{\Upsilon_e}(\xi)$, we can write (6.44) as

$$G_{\Upsilon_e}(\xi) \;=\; \prod_{i=1}^{N_s} \frac{1}{1-\xi/\xi_i}. \tag{6.45}$$

As we are investigating the pairwise probability of error, we will specify a particular $\beta$ and $\hat{\beta}$, so that the characteristic function $G_{\Upsilon_e}(\xi)$ given in (6.44) is completely specified, and we may proceed to its calculation.

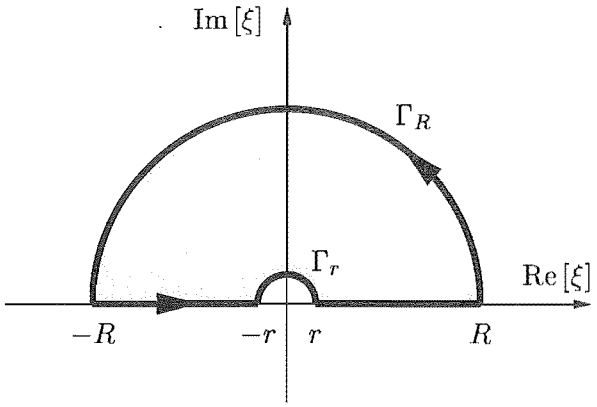



**Figure 6.2:** The closed curve $\Gamma$

### 6.3.2 Calculating the Pairwise Probability of Error

The residue theorem states that if a function $f(z)$ is single-valued and analytic inside and on a simple closed curve $\Gamma$ except at the singularities $z_1, z_2, z_3, \ldots$ which have residues given by $g_{-1,1}, g_{-1,2}, g_{-1,3}, \ldots$ then

$$\oint_{\Gamma} f(z)\, dz \;=\; 2\pi j(g_{-1,1} + g_{-1,2} + g_{-1,3} + \cdots). \tag{6.46}$$

That is, the integral of $f(z)$ around $\Gamma$ (traversed counter-clockwise) is $2\pi j$ times the sum of the residues of $f(z)$ at the singularities enclosed by $\Gamma$. By appropriate choice of the simple closed curve, we can use the residue theorem to perform the integration in (6.25), which is a line integral along the real axis.

A suitable curve $\Gamma$ is shown in Figure 6.2, which is made up of the real axis and two semi-circles, one of radius $R$, and one of radius $r$ denoted $\Gamma_R$ and $\Gamma_r$. The smaller semi-circle $\Gamma_r$ is used to avoid the pole at $\xi = 0$. As $R \to \infty$ and $r \to 0$, this curve will encompass the entire real axis.

We need to calculate the contribution of $\Gamma_R$ and $\Gamma_r$ to the sum of residues. The following theorem addresses the large semi-circle $\Gamma_R$ [Spi81].

**Theorem 1**

*If* $|f(z)| \leqslant m/R^k$ *for* $z = Re^{j\theta}$, *where* $k > 1$ *and* $m$ *are constants, then if* $\Gamma_A$ *is a semi-circle such as* $\Gamma_R$,

$$\lim_{R\to\infty} \int_{\Gamma_A} f(z)dz \;=\; 0. \tag{6.47}$$

Thus we require

$$\left|\frac{G_{\Upsilon_e}(\xi)}{\xi}\right| \leqslant \frac{m}{R^k}, \qquad \text{for } \xi = Re^{j\theta}. \tag{6.48}$$

But this will be the case as from (6.45), $G_{\Upsilon_e}(\xi)$ is an inverse polynomial in $\xi$ that is at least quadratic, and will therefore be bounded above by $m/R^k$ for $\xi = Re^{j\theta}$. Thus we have shown that

$$\lim_{R\to\infty} \int_{\Gamma_R} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi \;=\; 0. \tag{6.49}$$

The contribution of the smaller semi-circle $\Gamma_r$ can be directly calculated. Along $\Gamma_r$, $\xi = re^{j\theta}$, $0 \leqslant \theta \leqslant \pi$, and

$$\begin{aligned}
\lim_{r\to 0} \int_{\Gamma_r} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi \;&=\; \lim_{r\to 0} \int_{\pi}^{0} \frac{G_{\Upsilon_e}(re^{j\theta})}{re^{j\theta}}\, j\, re^{j\theta}\, d\theta \\
&=\; j\, G_{\Upsilon_e}(0) \int_{\pi}^{0} d\theta \\
&=\; -j\,\pi\, G_{\Upsilon_e}(0).
\end{aligned} \tag{6.50}$$

Thus using the results of (6.49) and (6.50) we can write

$$\begin{aligned}
\lim_{\substack{r\to 0 \\ R\to\infty}} \oint_{\Gamma} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi \;&=\; \int_{-\infty}^{\infty} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi + \lim_{R\to\infty} \int_{\Gamma_R} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi + \lim_{r\to 0} \int_{\Gamma_r} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi \\
&=\; \int_{-\infty}^{\infty} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi - j\,\pi\, G_{\Upsilon_e}(0).
\end{aligned} \tag{6.51}$$

But from the residue theorem in (6.46), the LHS of (6.51) is equal to $2\pi j$ times the sum of the residues of $G_{\Upsilon_e}(\xi)/\xi$ whose poles are contained by the closed curve $\Gamma$. Let us define $g_{-1,i}$ as the $i$-th residue of $G_{\Upsilon_e}(\xi)/\xi$ at $\xi = \xi_i$, this allows us to re-write (6.51) as

$$\int_{-\infty}^{\infty} \frac{G_{\Upsilon_e}(\xi)}{\xi}\, d\xi \;=\; 2\pi j \sum_{\substack{i \\ \mathrm{Im}[\xi_i]>0}} g_{-1,i} \;+\; j\,\pi\, G_{\Upsilon_e}(0). \tag{6.52}$$

Finally, using (6.52) in (6.25), and noting that $G_{\Upsilon_e}(0) = 0$ we obtain

$$\begin{aligned}
\Pr\{\Upsilon_e > 0\} \;&=\; \frac{G_{\Upsilon_e}(0)}{2} + \frac{1}{2\pi j} \left( 2\pi j \sum_{\substack{i \\ \mathrm{Im}[\xi_i]>0}} g_{-1,i} \;+\; j\,\pi\, G_{\Upsilon_e}(0) \right) \\
&=\; 1 \;+\; \sum_{\substack{i \\ \mathrm{Im}[\xi_i]>0}} g_{-1,i}
\end{aligned} \tag{6.53}$$

### 6.3.3 Calculating the Poles and Residues

The poles of $G_{\Upsilon_e}(\xi)$ are the values of $\xi$ that satisfy

$$\det\left(\mathbf{I}_{N_s} - 2j\xi \mathbf{R}^{*}_{\tilde{r}\tilde{r}} \mathbf{A}\right) \;=\; 0. \tag{6.54}$$

Unfortunately, these poles cannot be determined analytically, and thus must be computed numerically. Many numerical packages have library routines for finding eigenvalues, and

the solutions to (6.54) are closely related to the eigenvalues of $\mathbf{R}^*_{\tilde{r}\tilde{r}}\mathbf{A}$. The eigenvalues are the values of $\lambda$ of that satisfy

$$\det\left(\lambda\mathbf{I}_{N_s} - \mathbf{R}^*_{\tilde{r}\tilde{r}}\mathbf{A}\right) = 0. \tag{6.55}$$

Manipulating the LHS of (6.55), we obtain

$$(\lambda)^{N_s}\det\left(\mathbf{I}_{N_s} - \frac{1}{\lambda}\mathbf{R}^*_{\tilde{r}\tilde{r}}\mathbf{A}\right) = 0. \tag{6.56}$$

Equating (6.54) and (6.56), we find that the poles of $G_{\Upsilon_e}(\xi)$ are given by

$$\xi_i = \frac{-j}{2\lambda_i}. \tag{6.57}$$

However, the eigenvalue calculation can be numerically inaccurate, particularly if $N_s$ is large [Har96]. The solution is to use only the eigenvalues greater than $N_s \times \lambda_{\max} \times \epsilon$, where $\epsilon$ is the floating point precision of the machine used for calculations.

Having found the poles, we now proceed to the computation of their residues. The $i$-th residue of $f(z)$ at $z = z_i$, where $z_i$ is a pole of order $k_i$ is given by [Spi81]

$$g_{-1,i} = \lim_{z\to z_i}\frac{1}{(k_i-1)!}\frac{d^{k_i-1}}{dz^{k_i-1}}\left[(z-z_i)^{k_i}f(z)\right]. \tag{6.58}$$

For the *simple* poles (order 1), we have

$$\begin{aligned}
g_{-1,i} &= \lim_{\xi\to\xi_i}(\xi-\xi_i)\frac{G_{\Upsilon_e}(\xi)}{\xi} \\
&= \lim_{\xi\to\xi_i}\frac{(\xi-\xi_i)}{\xi}\prod_{k=1}^{N_s}\frac{1}{(1-\xi/\xi_k)}, && \xi_i \neq 0 \\
&= \frac{1}{\xi_i}\prod_{\substack{k=1\\k\neq i}}^{N_s}\frac{1}{(1-\xi_i/\xi_k)}\lim_{\xi\to\xi_i}\frac{(\xi-\xi_i)}{(1-\xi/\xi_i)}, && \xi_i \neq 0 \\
&= -\prod_{\substack{k=1\\k\neq i}}^{N_s}\frac{1}{(1-\xi_i/\xi_k)}, && \xi_i \neq 0.
\end{aligned} \tag{6.59}$$

The restriction $\xi_i \neq 0$ is not important as $\Gamma$ does not encompass this pole.

## 6.4 Performance of Uncoded $M$-DCPFSK in Rayleigh Flat Fading

In this section we present and discuss the results of the approximate performance calculations and simulations of uncoded $M$-DCPFSK in Rayleigh flat fading. The simulation model used is described in Appendix F and we now discuss the performance calculations.

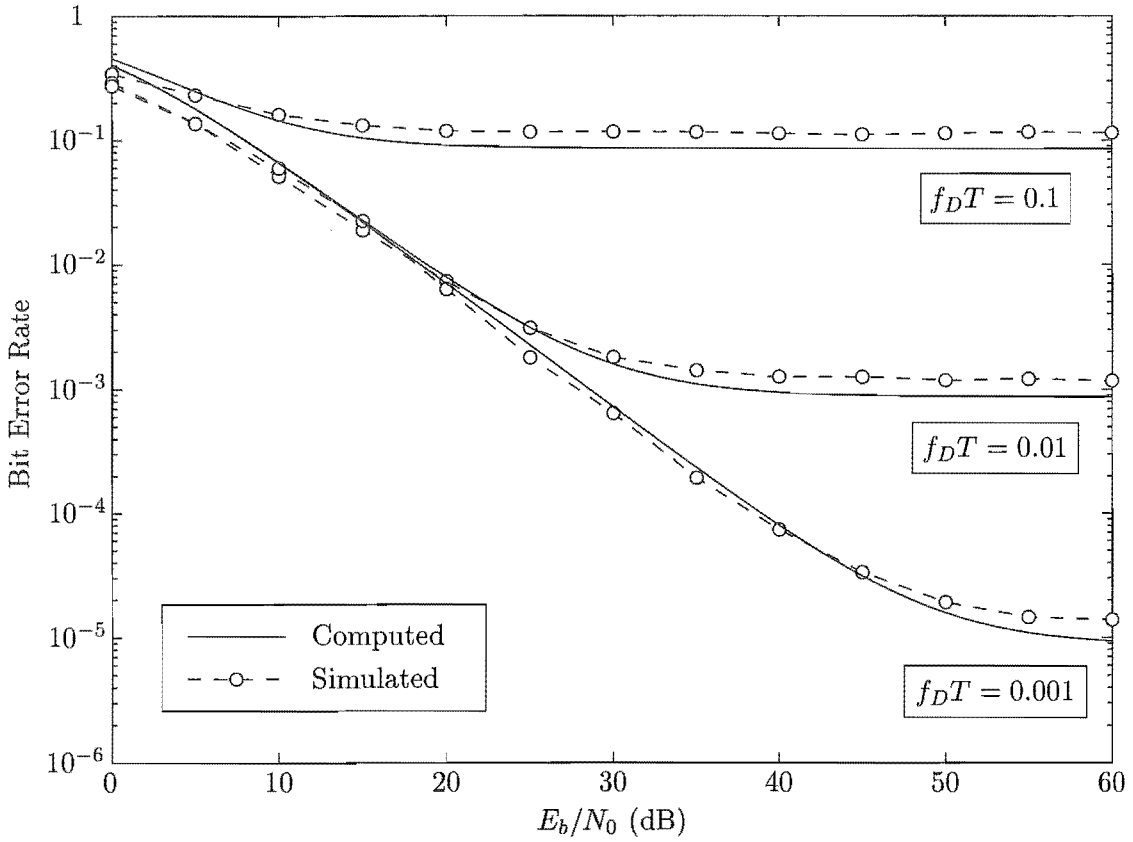



**Figure 6.3:** Computed and simulated performance of uncoded DMSK in Rayleigh flat fading with various values of $f_DT$

### 6.4.1 Approximate Performance Calculations

The approximate performance curves were calculated using just the pairwise probability of error for minimum distance term and its associated error coefficient as discussed in Chapter 5. The transmitted signal $\tilde{s}(iT_s, \beta)$ and an appropriate incorrectly-detected signal $\tilde{s}(iT_s, \hat{\beta})$ are required by the matrix **A**. If all the delay elements in the differential continuous phase encoder (DCPE) are set to zero, the sequences $\boldsymbol{U} = [0\ 1\ 0]$ and $\hat{\boldsymbol{U}} = [0\ 0\ 1]$ will generate the necessary signals to create an error event with the minimum distance for uncoded $M$-DCPFSK, as discussed in Section 3.9.2.

The autocorrelation matrix of the filtered noise $\mathbf{R}_{\tilde{w}\tilde{w}}$ was approximated by generating noise samples, filtering them by the filter used in the simulations and determining a mean for each time lag. It is important also to note that our approximate calculations assume that the signal $\tilde{s}$ and fading $\tilde{z}$ are unchanged by the filtering in the demodulator, which is not the case as the band-pass filter has quite a tight 3dB point. However, this

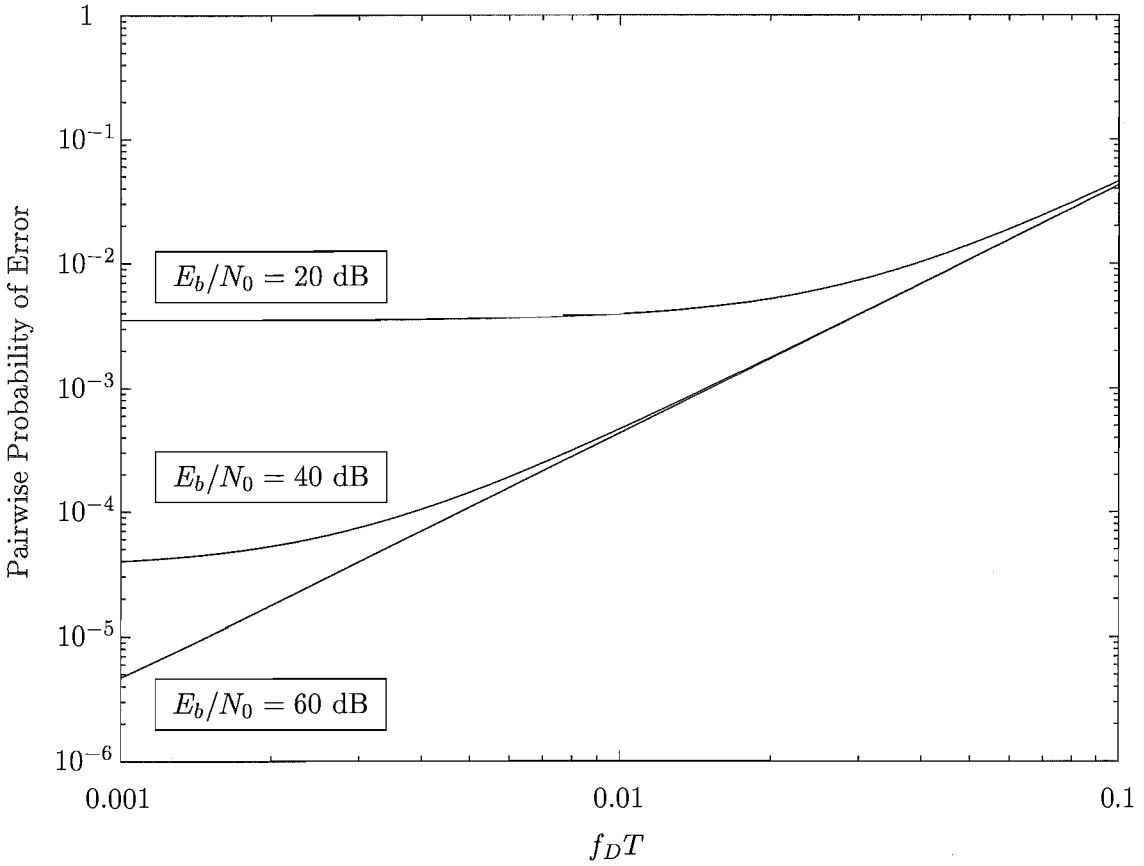



**Figure 6.4:** Pairwise probability of error vs $f_DT$ of uncoded DMSK in Rayleigh flat fading for various values of $E_b/N_0$

assumption greatly simplifies the calculations, and as will be seen, the computed performance agrees very well with the simulated performance.

### 6.4.2 Computed and Simulated Performance of Uncoded $M$-DCPFSK

Figure 6.3 shows the computed and simulated performance of uncoded DMSK in Rayleigh flat fading with various values of $f_DT$. It is evident that the computed curves and the simulated results agree very well. The performance for each value of $f_DT$ changes almost linearly with the SNR until it reaches an irreducible error rate, also known as an "error floor". The error floor is due to deep fades and their associated rapid phase changes. The performance is best in "slow" fading ($f_DT = 0.001$) and gets progressively worse as $f_DT$ is increased. This can be attributed to the fact that differential demodulation uses a sample of the faded carrier from the previous symbol period to demodulate the current symbol. As the value of $f_DT$ increases, the faded carrier changes more and more quickly

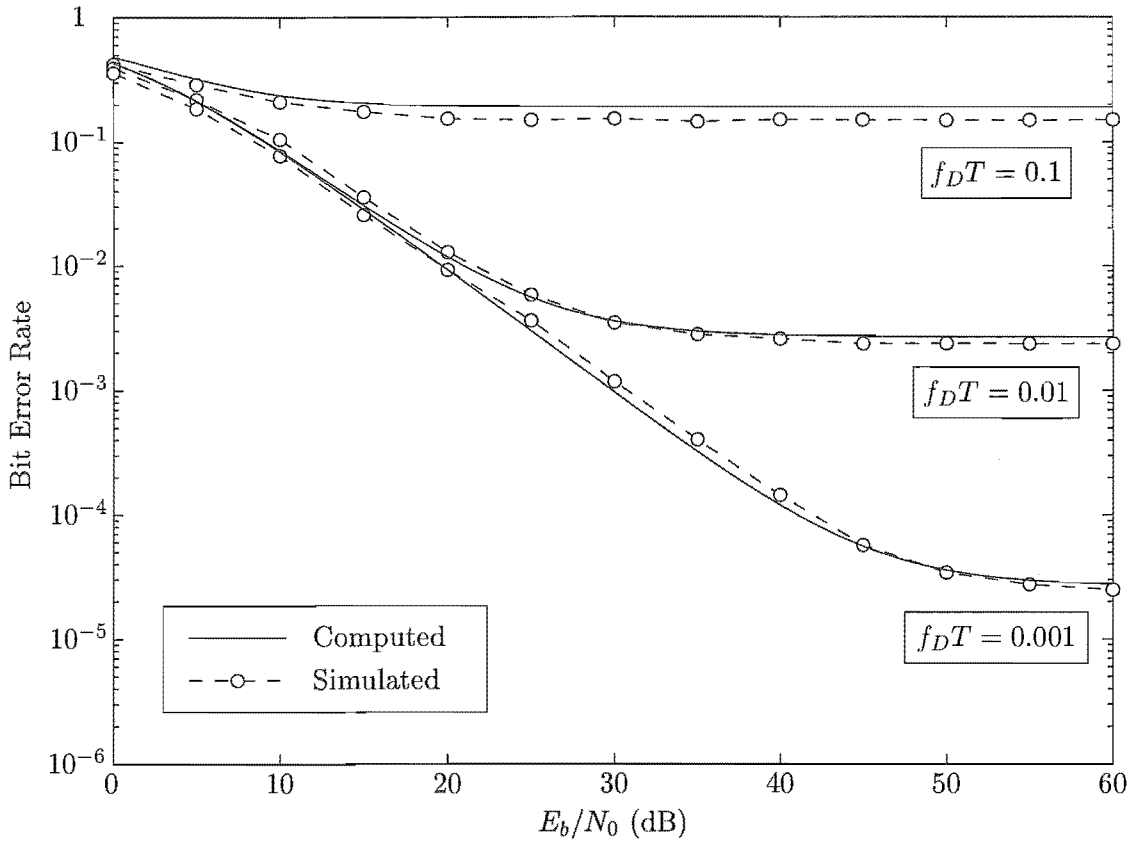



**Figure 6.5:** Computed and simulated performance of uncoded 4-DCPFSK in Rayleigh flat fading with various values of $f_DT$

and so becomes a progressively worse reference for demodulation. This is well illustrated in Figure 6.4, where the computed pairwise probability of error for uncoded DMSK for various values of $E_b/N_0$ is graphed against $f_DT$. The curve with $E_b/N_0 = 60$ dB best illustrates the dependence on $f_DT$, showing the degradation in performance as $f_DT$ increases. It is interesting to note that the 60-dB curve can be thought of as the asymptotic performance of uncoded DMSK as from Figure 6.3 we see that the performance has reached an error floor when $E_b/N_0 = 60$ dB and no further increase in $E_b/N_0$ will improve performance.

The computed and simulated performance of uncoded 4-DCPFSK and uncoded 8-DCPFSK are shown in Figure 6.5 and Figure 6.6 respectively. Both uncoded 4-DCPFSK and uncoded 8-DCPFSK exhibit similar error floors to those of uncoded DMSK, and the computed curves and the simulated results agree very well. The performance of both uncoded 4-DCPFSK and uncoded 8-DCPFSK exhibit similar dependence on $f_DT$ to that of DMSK.

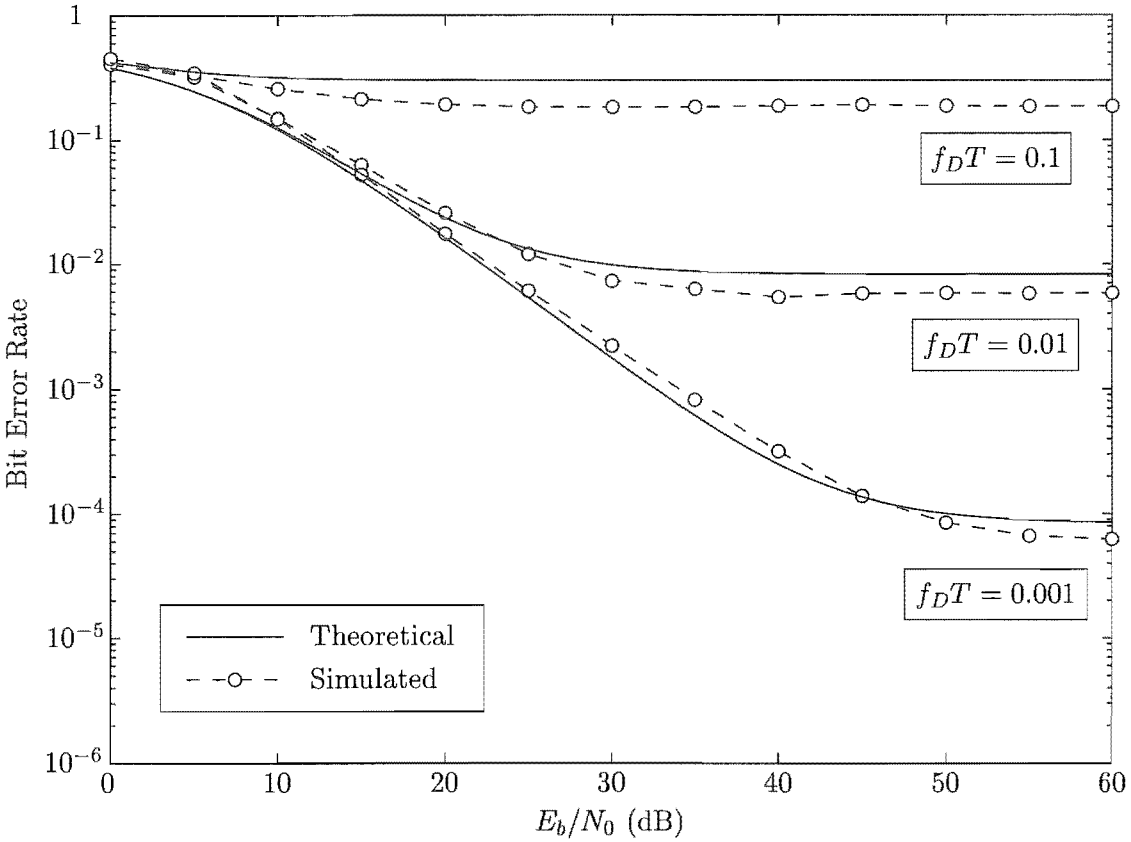



**Figure 6.6:** Computed and simulated performance of uncoded 8-DCPFSK in Rayleigh flat fading with various values of $f_DT$

## 6.5 Performance of Rate-1/2 Encoded 4-DCPFSK in Rayleigh Flat Fading

The theoretical performance of coded 4-DCPFSK is more complicated to calculate as more terms than just the minimum distance in the union bound are required to adequately predict performance. The calculation of the pairwise probability of error is just the same as that described in Section 6.3, but the data sequences $\boldsymbol{a}$ and $\hat{\boldsymbol{a}}$ to generate the error events with the required distances must be determined to perform the calculations. In Tables 6.1 and 6.2 we present the data sequences that will generate an error event with the given distance in 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{1}{2D+1}\right]$ and 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{2D+1}{D+1}\right]$ respectively, if all the delay elements in the transmitter and receiver are set to zero. It is important to note that these are not the only data sequences that will generate the required error events. For each distance there are many different error events and thus many different generating

**Table 6.1**

Data sequences to produce error events from the all zeros state for 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{1}{2D+1}\right]$

| $d^2$ | $\zeta$ | $\Xi(d)$ | $\boldsymbol{a}$ | $\hat{\boldsymbol{a}}$ |
|---|---|---|---|---|
| 3.00 | 2 | 0.50 | [0 1 0 0] | [0 1 2 0] |
| 3.15 | 2 | 0.94 | [0 1 0 0] | [0 0 2 0] |
| 3.49 | 2 | 0.19 | [1 1 0 0] | [1 2 2 0] |
| 3.73 | 3 | 0.62 | [0 0 0 0 0] | [0 1 0 0 0] |
| 3.88 | 3 | 0.62 | [0 1 2 2 0] | [0 2 1 0 0] |
| 4.00 | 2 | 1.81 | [0 0 0 0] | [0 0 2 0] |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

**Table 6.2**

Data sequences to produce error events from the all zeros state for 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \;\; \frac{2D+1}{D+1}\right]$

| $d^2$ | $\zeta$ | $\Xi(d)$ | $\boldsymbol{a}$ | $\hat{\boldsymbol{a}}$ |
|---|---|---|---|---|
| 4.00 | 2 | 0.50 | [0 2 0 0 0] | [0 2 2 2 0] |
| 4.09 | 3 | 1.69 | [0 1 0 0 0] | [0 0 1 2 0] |
| 4.15 | 3 | 0.56 | [0 1 0 0 0 0] | [0 1 1 1 0 0] |
| 4.24 | 4 | 0.30 | [0 1 0 0 0 0] | [0 0 0 1 0 0] |
| 4.49 | 2 | 0.19 | [0 2 0 0 0] | [0 1 3 0 0] |
| 4.94 | 3 | 0.56 | [0 1 0 0 2 0] | [0 1 3 1 0 0] |
| 4.97 | 3 | 0.25 | [0 1 0 2 1 0 0] | [0 0 0 2 0 0 0] |
| 5.00 | 2 | 0.94 | [0 2 0 0 0] | [0 0 2 0 0] |
| 5.00 | 3 | 1.00 | [0 1 0 0 0] | [0 0 3 0 0] |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

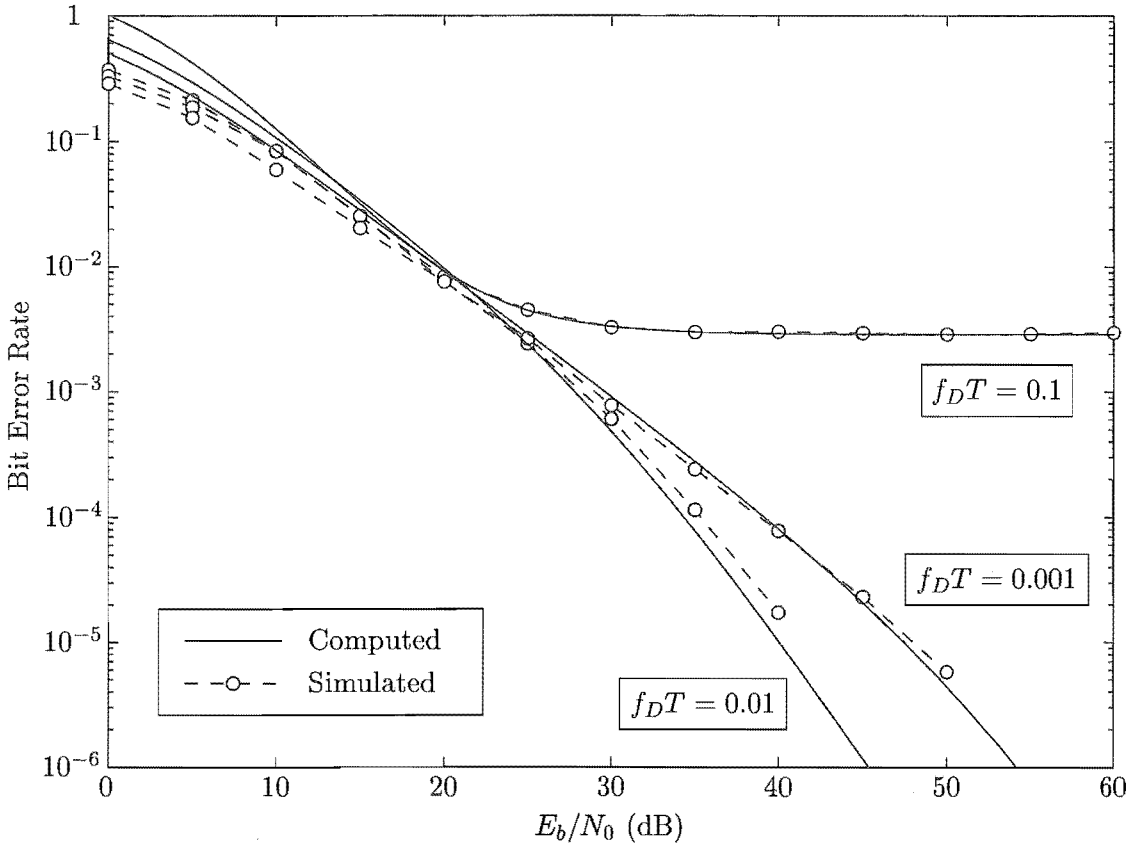



**Figure 6.7:** Computed and simulated performance of 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ in Rayleigh flat fading with various values of $f_DT$.

sequences. Once the pairwise probabilities of error have been found, they are multiplied by their associated error coefficient and then summed.

Figure 6.7 shows the computed and simulated performance of 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ in Rayleigh flat fading with various values of $f_DT$. For the $f_DT = 0.1$ case, the first 4 terms were used in the union bound. For the $f_DT = 0.01$ and $f_DT = 0.001$ cases, the first 5 and 6 terms respectively were used. The computed curves agree well with the simulation results. Only the $f_DT = 0.1$ case exhibits an error floor at SNR values less than 60 dB, but the computed curves do indeed show error floors. The $f_DT = 0.01$ case exhibits an error floor of $10^{-9}$ from about $E_b/N_0 = 65$ dB onwards, and the $f_DT = 0.001$ case exhibits an error floor of $4 \times 10^{-15}$ from about $E_b/N_0 = 100$ dB onwards. Thus the coding has dramatically reduced the error floors of uncoded 4-DCPFSK, except in very fast fading.

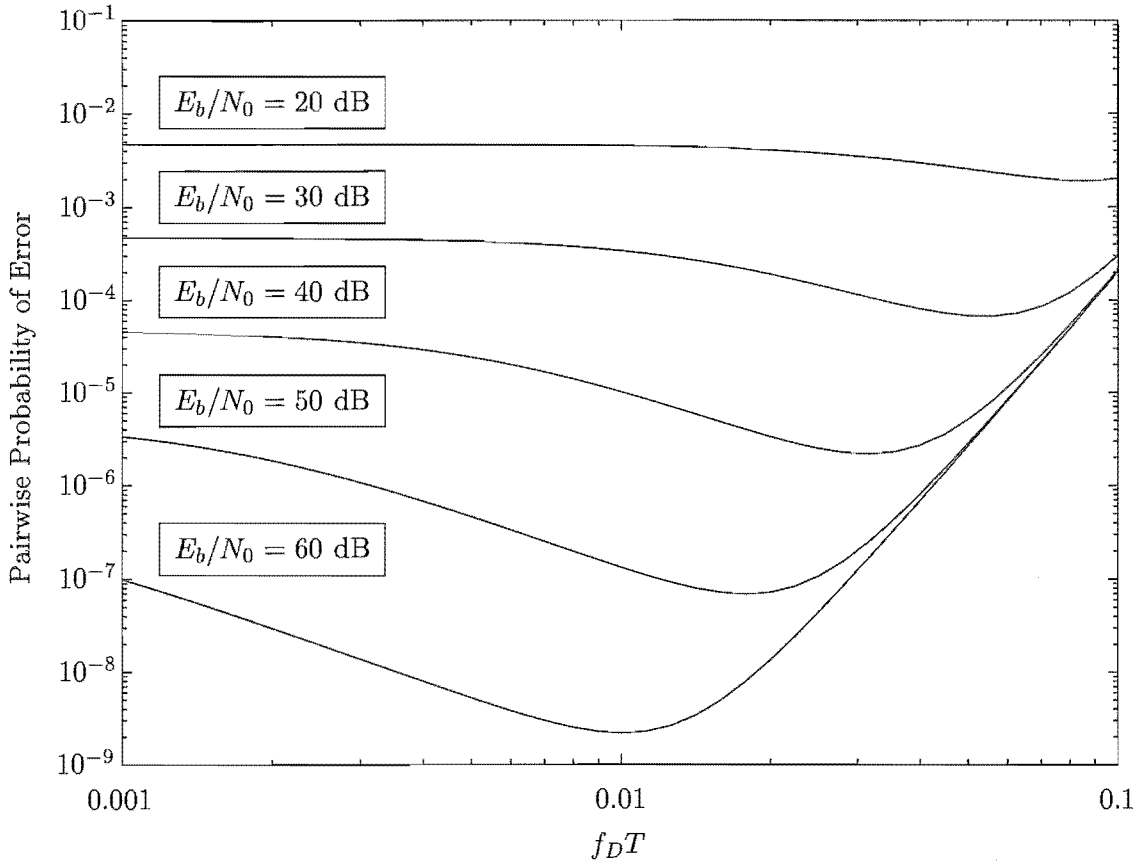



**Figure 6.8:** Computed pairwise probability of error vs $f_DT$ of 4-state, rate-1/2 encoded 4-DCPFSK using $\mathrm{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ in Rayleigh flat fading for various values of $E_b/N_0$.

Perhaps the most interesting point in Figure 6.7 is the fact that the $f_DT = 0.01$ case performs better than the $f_DT = 0.001$ case up until the $E_b/N_0 = 70$ dB point. In Figure 6.8 we further highlight the variation in the performance versus $f_DT$. It is important to note that the performance in Figure 6.8 is based simply on the minimum distance term and its error coefficient. Whereas the performance of uncoded DMSK steadily worsens as $f_DT$ increases, the performance in Figure 6.8 at $E_b/N_0$ values greater than 30 dB can improve somewhat as $f_DT$ increases. This can be explained by the fact that the error events in the coded case are much longer than those of the uncoded case, and as $f_DT$ increases the fading becomes less and less correlated, and the coding can exploit time diversity in the fading. However, once the $f_DT$ gets up to a certain value, the fading becomes so uncorrelated that the differential demodulation corrupts the signal too much. A similar improvement in *coherent* coded performance with increasing $f_DT$ has also been noted in [van96].

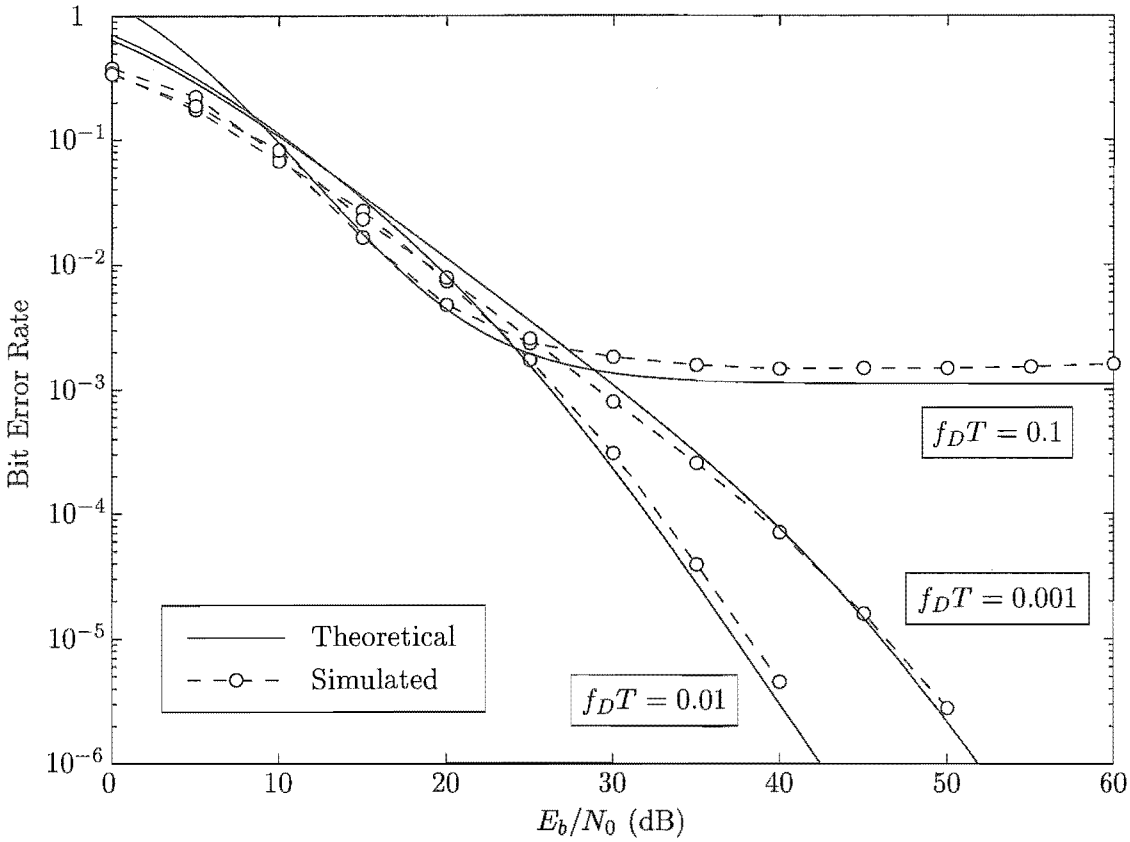



**Figure 6.9:** Computed and simulated performance of 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ in Rayleigh flat fading with various values of $f_DT$.

We now turn our attention to the computed and simulated performance of 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$, shown in Figure 6.9. For the $f_DT = 0.1$ case, the first 3 terms were used in the union bound. For the $f_DT = 0.01$ and $f_DT = 0.001$ cases, the first 4 and 9 terms respectively were used. Again, the computed curves agree well with the simulation results, and only the $f_DT = 0.1$ case exhibits an error floor at SNR values less than 60 dB. The computed curves show error floors. The computed curve for the $f_DT = 0.01$ case exhibits an error floor of $4 \times 10^{-10}$ from about $E_b/N_0 = 65$ dB onwards, and the $f_DT = 0.001$ case exhibits an error floor of $2 \times 10^{-15}$ from about $E_b/N_0 = 100$ dB onwards. Figure 6.9 also exhibits a similar variation in performance with $f_DT$ to that of Figure 6.7.

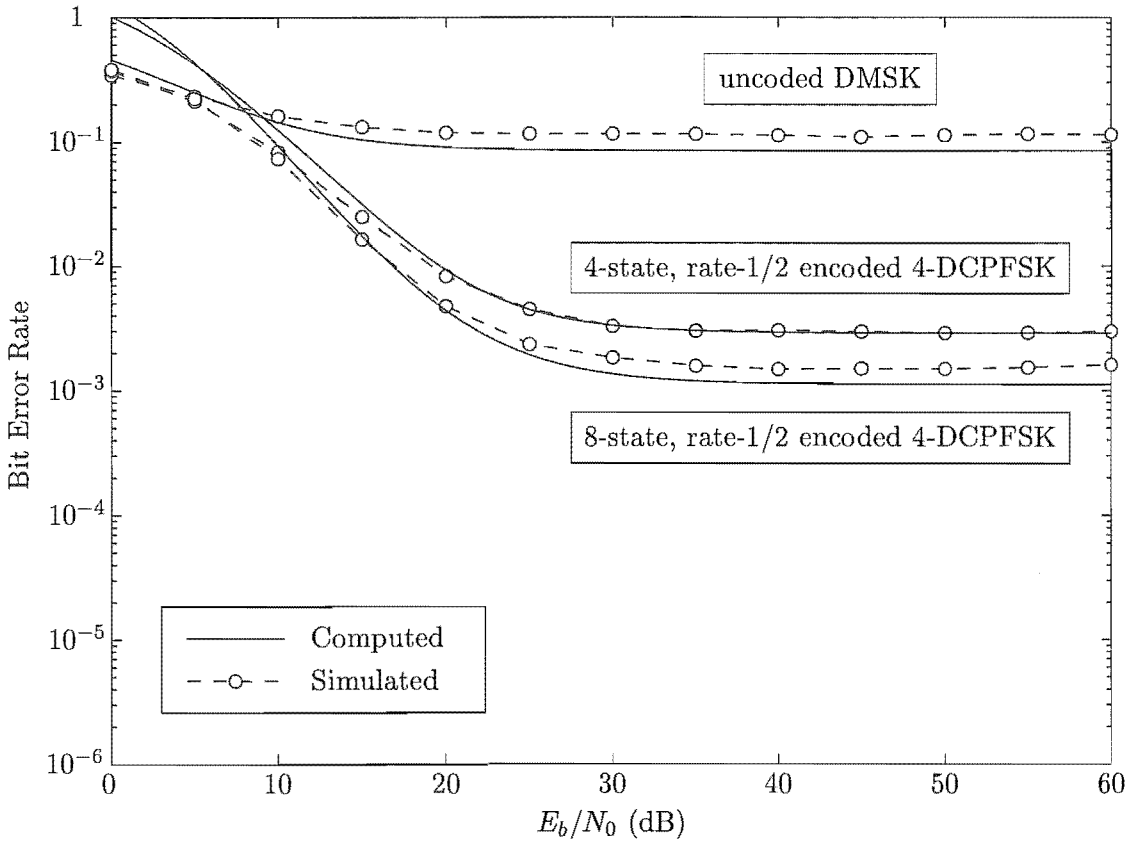



**Figure 6.10:** Computed and simulated performance of uncoded DMSK, 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ and 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{2D+1}{D+1}\right]$ in Rayleigh flat fading with $f_D T = 0.1$.

## 6.6 Comparison of Uncoded DMSK with Rate-1/2 Encoded 4-DCPFSK in Rayleigh Flat Fading

Uncoded DMSK and rate-1/2 encoded 4-DCPFSK are comparable systems in terms of information throughput. In this section we present a performance comparison of uncoded DMSK, 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ and 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{2D+1}{D+1}\right]$. These are the same systems that were compared in Section 5.6, but note that we do not consider the coherently-demodulated uncoded MSK case as it has no method to combat the fading.

Figure 6.10 compares the three systems in Rayleigh flat fading with $f_D T = 0.1$. The coding lowers the error floor from $10^{-1}$ to about $10^{-3}$. There is not much difference

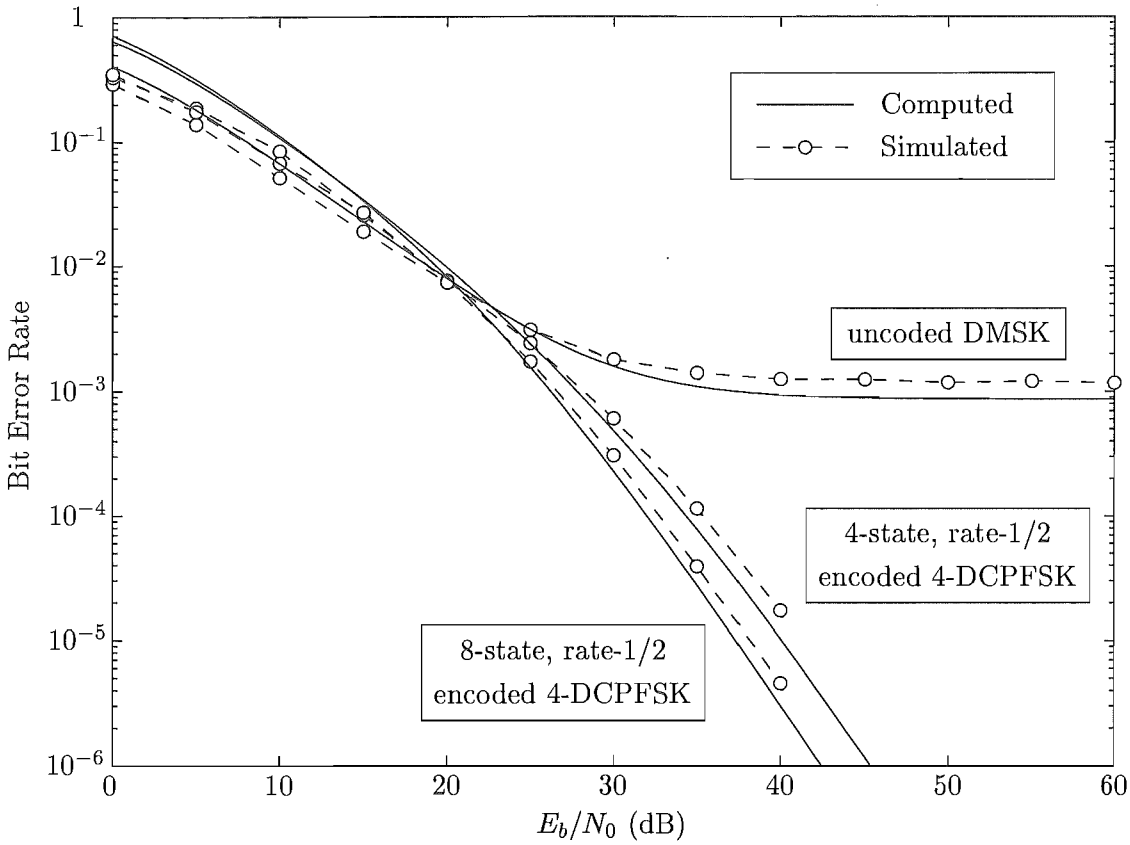



**Figure 6.11:** Computed and simulated performance of uncoded DMSK, 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ and 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{2D+1}{D+1}\right]$ in Rayleigh flat fading with $f_D T = 0.01$.

in the performance of the two coded systems, but this is to be expected as the 8-state code has a $d^2_{\text{min}}$ of 4.00, which is only 1.25 dB better than the $d^2_{\text{min}}$ of the 4-state code, and only a slight performance improvement was observed in the additive white Gaussian noise (AWGN) case in Figure 5.12.

A much more dramatic performance improvement obtained from coding is evident in the $f_D T = 0.01$ case, shown in Figure 6.11. Up to the 25-dB point, the three systems perform similarly, but any further increases in $E_b/N_0$ do not improve the BER of uncoded DMSK whilst those of the coded systems continue to decrease steeply. Again there is not a great performance difference between the codes. Finally, we present the performance of the two systems in Rayleigh flat fading with $f_D T = 0.001$ in Figure 6.12. The three systems perform similarly up to the 45-dB point, but from then on the DMSK encounters

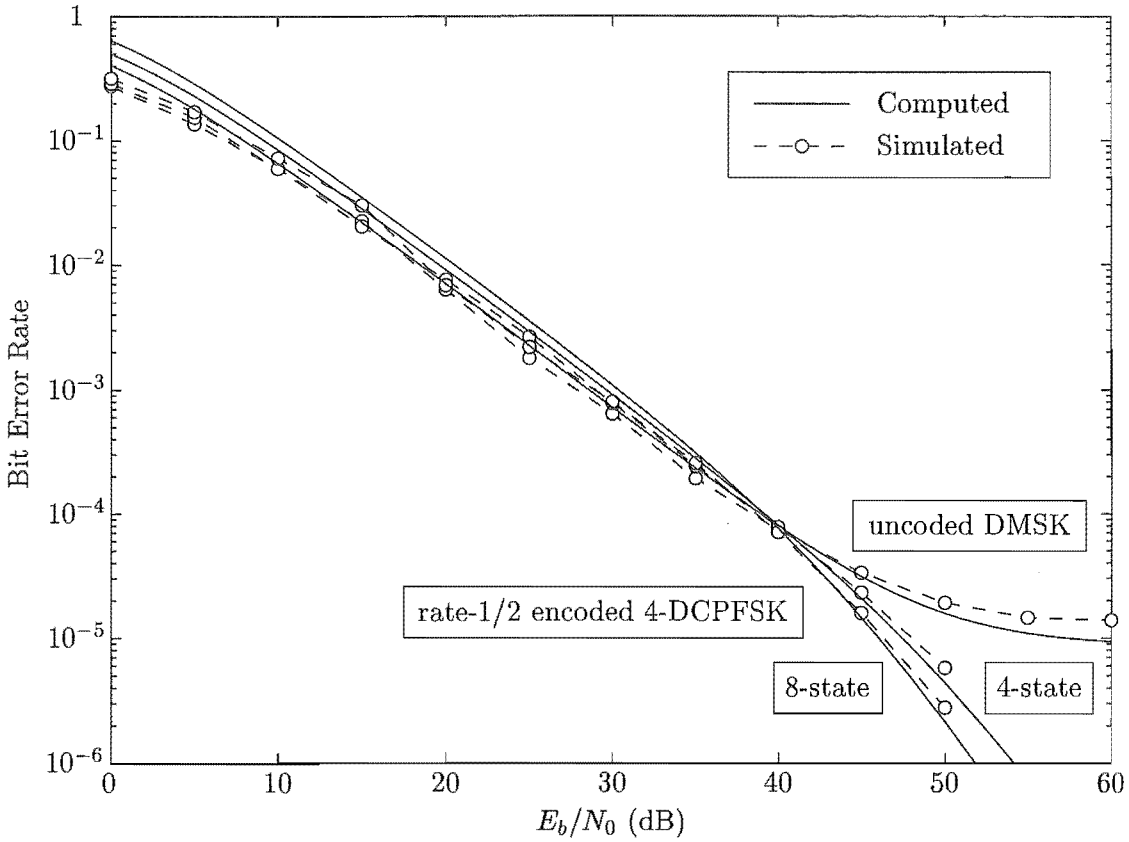



**Figure 6.12:** Computed and simulated performance of uncoded DMSK, 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ and 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{2D+1}{D+1}\right]$ in Rayleigh flat fading with $f_D T = 0.001$.

**Table 6.3**

Approximate BER values of the error floors of uncoded DMSK, 4-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{1}{2D+1}\right]$ and 8-state, rate-1/2 encoded 4-DCPFSK using $\mathbf{G}(D) = \left[1 \ \frac{2D+1}{D+1}\right]$

| | $f_D T$ | | |
|---|---|---|---|
| | 0.1 | 0.01 | 0.001 |
| Uncoded DMSK | $1 \times 10^{-1}$ | $1 \times 10^{-3}$ | $1 \times 10^{-5}$ |
| 4-state, rate-1/2 encoded 4-CPFSK | $3 \times 10^{-3}$ | $1 \times 10^{-9}$ | $4 \times 10^{-15}$ |
| 8-state, rate-1/2 encoded 4-CPFSK | $1 \times 10^{-3}$ | $4 \times 10^{-10}$ | $2 \times 10^{-15}$ |

an error floor while the coded systems continue to improve. We summarise the difference in error floors for the three systems in Table 6.3. It is evident that the coding lowers the error floor in all the cases, and that the error floor lowers as $f_D T$ decreases. There is only a slight difference in the error floors of the two coded systems.

## 6.7 Summary

In this chapter we have analysed the performance of uncoded and coded $M$-DCPFSK systems. Once we determined the form of the differentially-demodulated received signal that had been corrupted by Rayleigh flat fading, we were able to calculate the pairwise probability of error for $M$-DCPFSK. This involved developing an error metric which turned out to be a Gaussian quadratic form. We transformed the characteristic function of the error metric using the residue theorem, and determined the pdf of the error metric which we then integrated over the error region to find the pairwise probability of error. By applying the bounds and error coefficients of Chapter 5, we calculated the bit-error probability of both uncoded and coded $M$-DCPFSK systems. These computed curves were seen to agree very well with simulation results, and could be used to predict performance outside the scope of the simulations.

An interesting variation in error performance as the value of $f_DT$ changed was observed. In the uncoded systems the error performance gets progressively worse as $f_DT$ increases as the corrupted carrier used in the differential demodulator becomes a less and less reliable reference. The coded systems are able to exploit the time diversity introduced by the increasing $f_DT$ up to a point, where they too encounter the corrupted carrier reference problem. It is also evident that there is a much bigger performance improvement between the coded and uncoded cases in fading than in the AWGN case, as the simplest of the coded 4-DCPFSK systems is able to significantly lower the error floor of DMSK. However, the difference in performance between the two coded cases considered is similar to that of the AWGN case.

# Chapter 7

# Conclusions and Future Work

## 7.1 Conclusions

In this thesis we have developed a differential encoder that enables the preservation of the phase trellis of CPFSK through differential demodulation. The differential encoder is a linear encoder over the ring of integers modulo-$B$ ($\mathbb{Z}_B$). Differentially-encoded and differentially-demodulated CPFSK (DCPFSK) performs worse than coherently-demodulated CPFSK in additive white Gaussian noise (AWGN), and this is due to the presence of extra noise terms in the demodulated signal. These extra noise terms make exact performance analysis very difficult, but through some assumptions the performance of DCPFSK is found to be approximately 3 dB worse than that of CPFSK. Simulation results agree reasonably well with this theoretical approximation. The DCPFSK receivers are considerably simpler than their coherent counterparts, requiring no synchronisation other than that of symbol timing. Thus receiver complexity can be reduced at the expense of performance by using DCPFSK instead of CPFSK.

$M$-ary DCPFSK schemes with modulation index $h = 1/M$—which we call $M$-DCPFSK schemes—have a differential encoder defined over $\mathbb{Z}_M$. These schemes are particularly important as they have similar spectral characteristics to differentially-encoded and differentially-demodulated minimum shift keying (DMSK) and interface well with the decomposed model of CPFSK to form a decomposed model of $M$-DCPFSK. Additionally, these schemes combine well with external error-control encoders defined over $\mathbb{Z}_M$ to enable codes to be designed specifically for $M$-DCPFSK.

We developed a code search model of $M$-DCPFSK and used it find codes over $\mathbb{Z}_4$ for rate-1/2 encoded 4-DCPFSK and codes over $\mathbb{Z}_8$ for rate-2/3 encoded 8-DCPFSK. Results

of the code searches demonstrate that very little performance—in terms of the normalised minimum squared Euclidean distance $d^2_{\text{min}}$—is lost when going from coded $M$-CPFSK to coded $M$-DCPFSK.

The performance of both coded CPFSK and coded DCPFSK schemes can be approximated by a truncated union bound, which consists of a sum of pairwise probabilities of error that are based on error event distances and their associated error coefficients. By performing an exhaustive search on the error events up to certain length, we were able to obtain a theoretical approximation which agreed well with simulation results of coded systems in AWGN. We call this the "2-dB approximation" as it uses the contribution of all the error events within 2 dB of the $d^2_{\text{min}}$ of the coded system. It was found that the approximation was only reliable if *all* the error events within 2 dB had been found. In coded systems with large trellises, the error event search may become very computationally intensive.

By applying these performance measures, we were able to determine that the best 8-state, rate-1/2 encoded 4-DCPFSK system performs 0.5 dB better than uncoded DMSK in AWGN, and within 0.1 dB of coherently-demodulated MSK. Thus for a slight increase in the system's trellis, expensive and computationally intensive carrier recovery algorithms can be eliminated from the system.

Carrier recovery can be particularly difficult in a fading channel. By its very nature, differential demodulation can alleviate some of the problems encountered in fading channels. We analysed the performance of uncoded and coded DCPFSK systems in Rayleigh flat fading. A truncated union bound similar to the one encountered in AWGN was used, with the only difference being the pairwise probabilities of error. This agreed well with simulation results. It was found that the bit error rate (BER) of uncoded DCPFSK decreased almost linearly with increasing SNR until it encountered an irreducible BER, known as an "error floor". The level of this error floor is dependent on the maximum Doppler shift $f_D$ multiplied by the symbol period $T$. As $f_DT$ increases, so does the level of the error floor.

The performance of coded DCPFSK systems in Rayleigh flat fading exhibited quite different behaviour to that of the uncoded systems. The BER still encountered an error floor whose level increased as $f_DT$ increased, but the BER decreased with an increasing rate of descent. Interestingly, the BER of the coded systems can improve somewhat as $f_DT$ increases. This is due to the fact that the coding can take advantage of time diversity as the fading becomes less and less correlated. However, at a certain point—dependent

on SNR—the fading becomes too uncorrelated for the differential demodulation process to work, and the BER deteriorates as $f_D T$ is increased.

Comparisons between DMSK and rate-1/2 encoded 4-DCPFSK showed that the simplest 4-state code was able to lower the error floor significantly—by 2 to 10 orders of magnitude—in Rayleigh flat fading with values of $f_D T$ between 0.1 and 0.001. However, the difference in performance between codes was similar to their performance difference in AWGN, suggesting that the major gain was that of time diversity due to the fact that the error events in a coded system span more symbol intervals.

## 7.2 DCPFSK Evaluation

Five of the most important areas of a communication scheme's performance are power efficiency, delay, capacity, bit error rate and cost. Let us investigate how our DCPFSK system performs in each of these areas:

- *Power Efficiency*: DCPFSK transmitters are very power-efficient as DCPFSK is a constant-envelope modulation and the power amplifiers may be run in their non-linear, but power-efficient, regions.

- *Delay*: The DCPFSK systems considered here have very low delay, on the order of tens of symbols.

- *Capacity*: The DCPFSK systems considered here are not particularly spectrally efficient, so that they could not be used in a high-capacity system.

- *Bit Error Rate*: DCPFSK systems perform considerably worse than coherently-demodulated schemes in AWGN. With relatively simple coding however, this performance can be improved. In medium to slowly-varying flat fading channels, DCPFSK systems require no additional circuitry to achieve a reasonable level of performance. Again, simple codes can improve this performance markedly.

- *Cost*: DCPFSK receivers are very simple as symbol timing is the only synchronisation required. Combined with their power efficiency, DCPFSK systems are low cost.

Thus DCPFSK systems are power efficient, low-delay, low-capacity, medium-level BER performance and low-cost, most suitable for mobile voice or satellite applications.

## 7.3 Future Work

There are many avenues of future work opened up by this thesis, some of them are:

- The coded systems considered here had $h = 1/M$. Codes for DCPFSK systems with $h \neq 1/M$ could be designed. This would allow for a wider range of coded DCPFSK schemes to be considered.

- The coded performance improvement in fading seems to be due to time diversity. This could be further explored by using lower-rate coded DCPFSK systems. For instance, codes over $\mathbb{Z}_8$ for rate-1/3 encoded 8-DCPFSK, which is comparable to uncoded DMSK.

- The codes found here were designed for AWGN, where $d^2_{\text{min}}$ dominates performance. The searches could be performed with other search criteria—such as error event length—to find codes specifically designed for fading.

- The use of multi-$h$ DCPFSK would improve performance in fading, and could also be combined with coding. However, the complexity would increase significantly, and the differential encoder would need to be modified, as would the code search model.

- Performance in fading may also be improved by using partial response signalling. This would serve to lengthen error events, possibly increasing time diversity. Combining this with coding would also be an interesting problem. Again, complexity would increase significantly, and the differential encoder and the code search model would need to be modified. The resulting scheme would not actually be DCPFSK, but some other form of differentially-encoded and differentially-demodulated continuous phase modulation (DCPM).

- Using partial response signalling would also result in a scheme with better spectral efficiency than that of DCPFSK. Use of a smoother pulse shape would also achieve this. Yet again, the differential encoder and the code search model would need to be modified, and the result would be some other DCPM scheme.

- Finally, improvements in computing power—or better algorithms—would result in the ability to perform code searches for coded systems with larger trellises, and enable performance analysis of larger coded systems than those analysed in this thesis.

# Appendix A

# Properties of the Modulo Operator

Let $A$ and $B$ be positive integers, let $x$ and $y$ be real numbers, and let $\Gamma$ be a positive real number. We define the modulo-$\Gamma$ operator $R_\Gamma[x]$ as

$$R_\Gamma[x] \triangleq x - \left\lfloor \frac{x}{\Gamma} \right\rfloor \Gamma \tag{A.1}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, i.e. the largest integer not exceeding the enclosed number. Note that $R_\Gamma[x]$ will always be non-negative, regardless of the sign of $x$.

We now present some properties of the modulo operator. This list is by no means exhaustive, these are just some properties that are of use in this thesis

$$R_\Gamma[x \pm y] = R_\Gamma[x \pm R_\Gamma[y]] = R_\Gamma[R_\Gamma[x] \pm R_\Gamma[y]] \tag{A.2}$$

$$R_\Gamma\left[x\left(\frac{A}{B}\right)\right] = R_\Gamma\left[R_B[x]\left(\frac{A}{B}\right)\right] \tag{A.3}$$

$$R_\Gamma[xy] = R_\Gamma[R_\Gamma[x]\, y] \qquad \text{if and only if } y \text{ is an integer.} \tag{A.4}$$

$$R_A[R_B[x]] = R_A[x] \qquad \text{if and only if } A \text{ is a factor of } B. \tag{A.5}$$

$$\begin{aligned} & R_A[x] = R_A[y] \\ \Leftrightarrow \qquad & x = y + kA, \qquad k \text{ is an integer.} \end{aligned} \tag{A.6}$$

$$R_A[x] = y + kA, \qquad 0 \leqslant y < A, \text{ and } k \text{ is an integer.} \tag{A.7}$$

$$R_A[x] = x, \qquad \text{if } 0 \leqslant x < A. \tag{A.8}$$

# Appendix B

# Complex Noise

Let $w_{\mathrm{I}}(t)$ and $w_{\mathrm{Q}}(t)$ be independent, zero-mean, Gaussian processes, with psd's given by

$$S_{w_{\mathrm{I}}}(f) \;=\; S_{w_{\mathrm{Q}}}(f) \;=\; \begin{cases} N_0, & -\frac{f_w}{2} \leqslant f \leqslant \frac{f_w}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{B.1}$$

The autocorrelations of $w_{\mathrm{I}}(t)$ and $w_{\mathrm{Q}}(t)$ are the inverse Fourier transforms of $S_{w_{\mathrm{I}}}(f)$ and $S_{w_{\mathrm{Q}}}(f)$, given by

$$R_{w_{\mathrm{I}}}(\tau) \;=\; \mathrm{E}\left[w_{\mathrm{I}}(t) w_{\mathrm{I}}(t+\tau)\right] \;=\; N_0 f_w \,\mathrm{sinc}\left(f_w \tau\right) \tag{B.2}$$

$$R_{w_{\mathrm{Q}}}(\tau) \;=\; \mathrm{E}\left[w_{\mathrm{Q}}(t) w_{\mathrm{Q}}(t+\tau)\right] \;=\; N_0 f_w \,\mathrm{sinc}\left(f_w \tau\right). \tag{B.3}$$

As stated, $w_{\mathrm{I}}(t)$ and $w_{\mathrm{Q}}(t)$ are independent, so that

$$\mathrm{E}\left[w_{\mathrm{I}}(t) w_{\mathrm{Q}}(t+\tau)\right] \;=\; 0. \tag{B.4}$$

Let $\tilde{w}(t)$ be a complex noise process, defined as

$$\tilde{w}(t) \;\triangleq\; w_{\mathrm{I}}(t) + j\, w_{\mathrm{Q}}(t). \tag{B.5}$$

Like its component processes, $\tilde{w}(t)$ is also zero-mean, as shown by

$$\mathrm{E}\left[\tilde{w}(t)\right] \;=\; \mathrm{E}\left[\tilde{w}(t) + j\,\tilde{w}(t)\right] \;=\; \mathrm{E}\left[\tilde{w}(t)\right] + j\,\mathrm{E}\left[\tilde{w}(t)\right] \;=\; 0. \tag{B.6}$$

The autocorrelation of $\tilde{w}(t)$ is given by

$$\begin{aligned} R_{\tilde{w}}(\tau) \;&=\; \mathrm{E}\left[\tilde{w}(t)\tilde{w}^*(t+\tau)\right] \\ &=\; \mathrm{E}\left[\{w_{\mathrm{I}}(t) + j\, w_{\mathrm{Q}}(t)\}\{w_{\mathrm{I}}(t+\tau) - j\, w_{\mathrm{Q}}(t+\tau)\}\right] \\ &=\; \mathrm{E}\left[w_{\mathrm{I}}(t) w_{\mathrm{I}}(t+\tau) - j\, w_{\mathrm{I}}(t) w_{\mathrm{Q}}(t+\tau) + j\, w_{\mathrm{Q}}(t) w_{\mathrm{I}}(t+\tau) + w_{\mathrm{Q}}(t) w_{\mathrm{Q}}(t+\tau)\right] \end{aligned}$$

$$
\begin{aligned}
&= \mathrm{E}\left[w_{\mathrm{I}}(t)w_{\mathrm{I}}(t+\tau)\right] - j\,\mathrm{E}\left[w_{\mathrm{I}}(t)w_{\mathrm{Q}}(t+\tau)\right] + j\,\mathrm{E}\left[w_{\mathrm{Q}}(t)w_{\mathrm{I}}(t+\tau)\right] \\
&\quad + \mathrm{E}\left[w_{\mathrm{Q}}(t)w_{\mathrm{Q}}(t+\tau)\right] \\
&= N_0 f_w \,\mathrm{sinc}\left(f_w\tau\right) - j\,0 + j\,0 + N_0 f_w \,\mathrm{sinc}\left(f_w\tau\right) \\
&= 2N_0 f_w \,\mathrm{sinc}\left(f_w\tau\right). \qquad \text{(B.7)}
\end{aligned}
$$

The psd of $\tilde{w}(t)$ is the Fourier transform of $R_{\tilde{w}}(\tau)$, given by

$$
S_{\tilde{w}}(f) = \begin{cases} 2N_0, & -\frac{f_w}{2} \leqslant f \leqslant \frac{f_w}{2} \\ 0, & \text{otherwise.} \end{cases} \qquad \text{(B.8)}
$$

The final remark is that the "unconjugated" autocorrelation of $\tilde{w}(t)$ is equal to zero, as

$$
\begin{aligned}
\mathrm{E}\left[\tilde{w}(t)\tilde{w}(t+\tau)\right] &= \mathrm{E}\left[\{w_{\mathrm{I}}(t) + j\,w_{\mathrm{Q}}(t)\}\{w_{\mathrm{I}}(t+\tau) + j\,w_{\mathrm{Q}}(t+\tau)\}\right] \\
&= \mathrm{E}\left[w_{\mathrm{I}}(t)w_{\mathrm{I}}(t+\tau) + j\,w_{\mathrm{I}}(t)w_{\mathrm{Q}}(t+\tau) + j\,w_{\mathrm{Q}}(t)w_{\mathrm{I}}(t+\tau)\right. \\
&\quad \left. - w_{\mathrm{Q}}(t)w_{\mathrm{Q}}(t+\tau)\right] \\
&= \mathrm{E}\left[w_{\mathrm{I}}(t)w_{\mathrm{I}}(t+\tau)\right] + j\,\mathrm{E}\left[w_{\mathrm{I}}(t)w_{\mathrm{Q}}(t+\tau)\right] + j\,\mathrm{E}\left[w_{\mathrm{Q}}(t)w_{\mathrm{I}}(t+\tau)\right] \\
&\quad - \mathrm{E}\left[w_{\mathrm{Q}}(t)w_{\mathrm{Q}}(t+\tau)\right] \\
&= N_0 f_w \,\mathrm{sinc}\left(f_w\tau\right) + j\,0 + j\,0 - N_0 f_w \,\mathrm{sinc}\left(f_w\tau\right) \\
&= 0. \qquad \text{(B.9)}
\end{aligned}
$$

# Appendix C

# The Viterbi Algorithm

In this Appendix we present a brief summary of the workings of the Viterbi algorithm (VA) as it relates to this thesis. We draw largely from Forney's paper [For73].

## C.1 General statement of the problem

The VA can be thought of as a solution to the problem of maximum *a posteriori* probability (MAP) estimation of the state sequence of a finite-state, discrete-time Markov process observed in memoryless noise.

The underlying Markov process is characterised as follows. Time is discrete. The state $\sigma_{k_V}$ at time $k_V$ is one of a finite number $S_V$. The state sequence is represented by the vector $\boldsymbol{\sigma} = [\ldots, \sigma_{k_V-1}, \sigma_{k_V}, \sigma_{k_V+1}, \ldots]$.

As the process is Markov, the probability of being in state $\sigma_{k_V+1}$ at time $k_V + 1$, given all states up to time $k_V$, depends only on the state $\sigma_{k_V}$ at time $k_V$:

$$\Pr\{\sigma_{k_V+1} \,|\, \sigma_{k_V}, \sigma_{k_V-1}, \ldots\} \;=\; \Pr\{\sigma_{k_V+1} \,|\, \sigma_{k_V}\}. \tag{C.1}$$

The transition probabilities $\Pr\{\sigma_{k_V+1} \,|\, \sigma_{k_V}\}$ may be time-varying, but we do not explicitly indicate this in the notation.

It is convenient to define the *transition* $\xi_{k_V}$ at time $k_V$ from $\sigma_{k_V}$ to $\sigma_{k_V+1}$ as the pair of states $\sigma_{k_V+1}$ and $\sigma_{k_V}$:

$$\xi_{k_V} \;\triangleq\; \{\sigma_{k_V+1}, \sigma_{k_V}\}. \tag{C.2}$$

We let $\Xi$ be the (possibly time-varying) set of transitions $\xi_{k_V} = \{\sigma_{k_V+1}, \sigma_{k_V}\}$ for which $\Pr\{\sigma_{k_V+1} \,|\, \sigma_{k_V}\} \neq 0$, and $|\Xi|$ their number. Clearly $|\Xi| \leq S_V^2$. There is evidently a one-to-one correspondence between state sequences $\boldsymbol{\sigma}$ and transition sequences $\boldsymbol{\xi} = [\ldots, \xi_{k_V-1}, \xi_{k_V}, \xi_{k_V+1}, \ldots]$.

The process is assumed to be observed in memorlyess noise; that is, there is a sequence $z$ of observations $z_{k_V}$ in which $z_{k_V}$ depends probabilistically only on the transition $\xi_{k_V}$ at time $k_V$:

$$\Pr\{z \mid \sigma\} = \Pr\{z \mid \xi\} = \prod_{k_V} \Pr\{z_{k_V} \mid \xi_{k_V}\}. \tag{C.3}$$

We can describe $z$ as the output of some memoryless channel whose input sequence is $\xi$. Again, though we shall not indicate it explicitly, the channel may be time varying in the sense that $\Pr\{\sigma_{k_V} \mid \xi_{k_V}\}$ may be a function of $k_V$.

The Markov process can be described as a trellis, where each node corresponds to a distinct state $\sigma_{k_V}$ at a given time $k_V$, and each branch represents a transition $\xi_{k_V}$ to some new state $\sigma_{k_V+1}$ at the next instant of time $k_V + 1$. The most important property of the trellis is that every possible state sequence $\sigma$ corresponds to a unique path through the trellis and vice versa.

Our problem is thus to find the state sequence $\sigma$ for which $\Pr\{z \mid \sigma\}$ is maximum. It is convenient however, to restate this problem as finding the state sequence $\sigma$ for which the "length" $-\ln\Pr\{z \mid \sigma\}$ is minimum. Using (C.3) we find

$$-\ln\Pr\{z \mid \sigma\} = -\ln\left[\prod_{k_V} \Pr\{z_{k_V} \mid \xi_{k_V}\}\right] = \sum_{k_V} (-\ln\Pr\{z_{k_V} \mid \xi_{k_V}\}). \tag{C.4}$$

We now define the "length" of each transition as

$$\lambda_{\mathrm{VA}}(\xi_{k_V}) \triangleq -\ln\Pr\{z_{k_V} \mid \xi_{k_V}\}, \tag{C.5}$$

which we call the *branch metric*. This allow us to finally state our problem as that of finding the state sequence $\sigma$ for which the "length" or *path metric*

$$\sum_{k_V} \lambda_{\mathrm{VA}}(\xi_{k_V}) \tag{C.6}$$

is minimum.

## C.2 The algorithm

At any particular time, there will be several paths terminating at a particular node (or state) in the trellis. Amongst the paths terminating in state $\sigma_{k_V}$ at time $k_V$, the one with the smallest path metric is called the *survivor* corresponding to the state $\sigma_{k_V}$ and denoted by $\hat{\sigma}(\sigma_{k_V})$. The path metric of the survivor $\hat{\sigma}(\sigma_{k_V})$ is denoted by $\Lambda_{\mathrm{VA}}(\sigma_{k_V})$, defined as

$$\Lambda_{\mathrm{VA}}(\sigma_{k_V}) \triangleq \sum_{i=0}^{k_V-1} \lambda_{\mathrm{VA}}(\xi_i), \tag{C.7}$$

where the transitions are all from the survivor $\hat{\sigma}(\sigma_{k_V})$. Each time step, an estimate of the state sequence $K_d$-time intervals previously is output. The quantity $K_d$ is called the *decision depth*. The choice of $K_d$ is important as the delay through the VA and the storage requirements are proportional to it. However if $K_d$ is too small performance may suffer.

### C.2.1 Storage

Only $K_d$ states for each of the $S_V$ survivor paths $\hat{\sigma}(\sigma_{k_V})$, need to be stored, along with the $S_V$ path metrics $\Lambda_{\mathrm{VA}}(\sigma_{k_V})$. Unless the system starts in a known state, these should all be initialised to the same arbitrary value.

### C.2.2 Operation

For each time $k_V$, the VA runs as follows:

- Calculate the provisional path metrics $\tilde{\Lambda}_{\mathrm{VA}}(\sigma_{k_V+1}, \sigma_{k_V})$:

$$\tilde{\Lambda}_{\mathrm{VA}}(\sigma_{k_V+1}, \sigma_{k_V}) \triangleq \Lambda_{\mathrm{VA}}(\sigma_{k_V}) + \lambda_{\mathrm{VA}}(\xi_{k_V} = \{\sigma_{k_V+1}, \sigma_{k_V}\}) \tag{C.8}$$

  for each possible transition.

- For each state set

$$\Lambda_{\mathrm{VA}}(\sigma_{k_V+1}) = \min_{\sigma_{k_V}} \tilde{\Lambda}_{\mathrm{VA}}(\sigma_{k_V+1}, \sigma_{k_V}). \tag{C.9}$$

- Store $\Lambda_{\mathrm{VA}}(\sigma_{k_V+1})$ and its associated path $\hat{\sigma}(\sigma_{k_V+1})$ for each trellis state.

- Find the $\Lambda_{\mathrm{VA}}(\sigma_{k_V+1})$ with the minimum value, and from its associated $\hat{\sigma}(\sigma_{k_V+1})$, output the state transition at time $k_V + 1 - K_d$.

- Set $k_V$ to $k_V + 1$ and repeat.

### C.2.3 Complexity

The complexity of the VA depends on the size of the trellis of the system, particularly the total number of branches, as each one requires a metric calculation.

# Appendix D

# CPFSK Simulations

Simulations on a computer must be performed in discrete-time. In this appendix we describe our CPFSK simulation model and a sampling receiver for CFPSK, which relates much more accuarately to the simulations that were performed. We show in Appendix E that this receiver acheives the performance of (2.62).

## D.1 A Sampling Receiver Structure for CPFSK

A sampling receiver for coherent CPFSK consists of three main blocks, a coherent demodulator, a sampler, and a Viterbi processor as shown in Figure D.1. The coherent demodulator is exactly the same as that described in Section 2.7.1. We now discuss the other two elements.

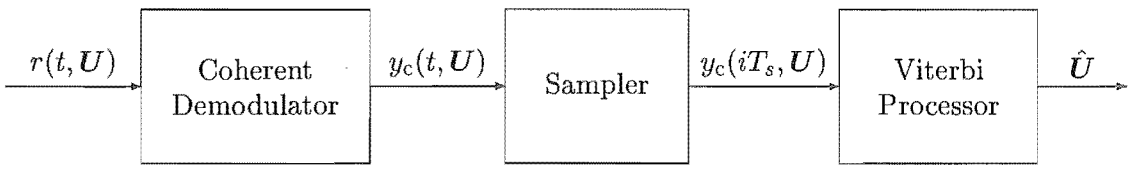



**Figure D.1:** A sampling receiver structure for coherent CPFSK

### D.1.1 Sampler

The output of the coherent demodulator $y_c(t, \boldsymbol{U})$ is sampled at $r_s$ Hz. The sample period $T_s$ is given by

$$T_s = \frac{1}{r_s}. \tag{D.1}$$

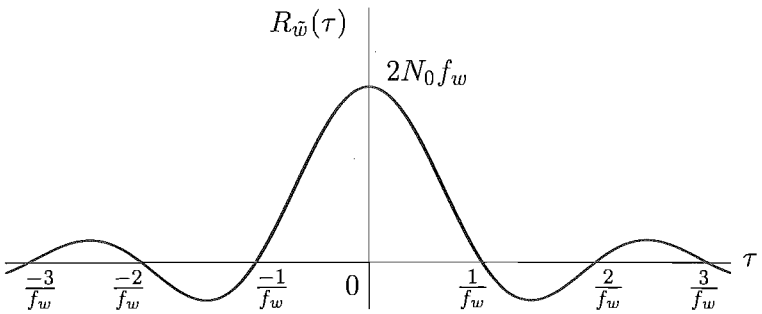



**Figure D.2:** Autocorrelation function of the low-pass filtered white Gaussian noise

To simplify processing, we choose $T_s$ such that

$$\frac{T}{T_s} = D_s, \tag{D.2}$$

where $D_s$ is a positive integer. This ensures that we have an integral number of samples per symbol. The sampled version of $y_c(t, \boldsymbol{U})$ is $y_c(iT_s, \boldsymbol{U})$, given by

$$y_c(iT_s, \boldsymbol{U}) = \frac{1}{\sqrt{2}}[\tilde{s}(iT_s, \boldsymbol{U}) + \tilde{w}(iT_s)]. \tag{D.3}$$

We want the samples of $\tilde{w}(t)$ to be uncorrelated. The autocorrelation of $\tilde{w}(t)$ is the inverse Fourier transform of $S_{\tilde{w}}(f)$ in (2.38), given by

$$R_{\tilde{w}}(\tau) = 2N_0 f_w \operatorname{sinc}(f_w \tau). \tag{D.4}$$

$R_{\tilde{w}}(\tau)$ is shown in Figure D.2. It is clear that if we choose $r_s$ such that

$$r_s = f_w, \tag{D.5}$$

then

$$T_s = \frac{1}{f_w}, \tag{D.6}$$

and the noise samples $\tilde{w}(iT_s)$ will be uncorrelated and thus white. Formally, their autocorrelation is

$$R_{\tilde{w}}(iT_s, kT_s) = \mathrm{E}[\tilde{w}(iT_s)\tilde{w}^*(kT_s)] = 2N_0 f_w \, \delta_{i,k}, \tag{D.7}$$

where $\delta_{i,k}$ is the *Kronecker delta function*, defined as

$$\delta_{i,k} \triangleq \begin{cases} 1, & i = k \\ 0, & \text{otherwise,} \end{cases} \tag{D.8}$$

where $i$ and $k$ are integers. Thus as long as (D.5) is adhered to, the demodulated signal will consist of the transmitted signal perturbed by AWGN. As the band-width of the low-pass filters in the demodulator determine the band-width of the base-band noise, we must increase the band-width of the low-pass filters in the demodulator if we wish to increase the sampling rate, so that

$$f_{\mathrm{L}} = r_s. \tag{D.9}$$

### D.1.2 Viterbi Processor

As the transmitted signal in each symbol interval depends on all the previously transmitted symbols, we must determine the state sequence of the transmitter to detect the transmitted data $\boldsymbol{U}$. This is called maximum-likelihood sequence estimation (MLSE) [For72]. Using the time-invariant phase trellis and the memoryless modulator, the transmitter can be thought of as a Markov source, that is the transmitted signal $\tilde{s}(t, \boldsymbol{U})$ in the $n$-th symbol period depends only on the state of the CPE $V_n$ and the current information symbol $U_n$. Thus the output of the sampler, $y_{\mathrm{c}}(iT_s, \boldsymbol{U})$, consists of the output of a Markov source plus AWGN.

Let us use $\tilde{\boldsymbol{s}}(\boldsymbol{U})$ and $\boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})$ to denote the sequences $\tilde{s}(iT_s, \boldsymbol{U})$ and $y_{\mathrm{c}}(iT_s, \boldsymbol{U})$ respectively. Our problem then is to find the sequence $\tilde{\boldsymbol{s}}(\boldsymbol{U})$ that maximises $\Pr\{\tilde{\boldsymbol{s}}(\boldsymbol{U}) \mid \boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})\}$, that is the probability that $\tilde{\boldsymbol{s}}(\boldsymbol{U})$ was the transmitted sequence given that $\boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})$ was received. This is equivalent to finding the sequence $\tilde{\boldsymbol{s}}(\boldsymbol{U})$ that maximises the joint probability $\Pr\{\tilde{\boldsymbol{s}}(\boldsymbol{U}), \boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})\}$ [For73], as

$$\Pr\{\tilde{\boldsymbol{s}}(\boldsymbol{U}), \boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})\} = \Pr\{\tilde{\boldsymbol{s}}(\boldsymbol{U}) \mid \boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})\} \Pr\{\boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})\}. \tag{D.10}$$

However, assuming $\Pr\{\tilde{\boldsymbol{s}}(\boldsymbol{U})\}$ to be a uniform distribution, $\Pr\{\tilde{\boldsymbol{s}}(\boldsymbol{U}), \boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})\}$ also factors as

$$\Pr\{\tilde{\boldsymbol{s}}(\boldsymbol{U}), \boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U})\} = \Pr\{\boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U}) \mid \tilde{\boldsymbol{s}}(\boldsymbol{U})\} \Pr\{\tilde{\boldsymbol{s}}(\boldsymbol{U})\}, \tag{D.11}$$

and our problem can be restated as finding the sequence $\tilde{\boldsymbol{s}}(\boldsymbol{U})$ that maximises the conditional probability $\Pr\{\boldsymbol{y}_{\mathrm{c}}(\boldsymbol{U}) \mid \tilde{\boldsymbol{s}}(\boldsymbol{U})\}$. As discussed in Appendix C, this is a problem suitable for the Viterbi algorithm.

In a CPFSK system with no external ECC, the system trellis interval is just the symbol period, so that the transition in the $k_V$-th trellis interval $\xi_{k_V}$, is just the transmitted signal in the $n$-th symbol interval, which we call $\tilde{s}(n, \boldsymbol{U})$ and define in sampled form as

the vector

$$\tilde{s}(n,\boldsymbol{U}) \triangleq \left[\tilde{s}(nD_sT_s,\boldsymbol{U}) \quad \tilde{s}([nD_s+1]T_s,\boldsymbol{U}) \quad \cdots \quad \tilde{s}([(n+1)D_s-1]T_s,\boldsymbol{U})\right]^\dagger, \quad \text{(D.12)}$$

where $\boldsymbol{x}^\dagger$ represents the non-conjugate transpose of $\boldsymbol{x}$. Let the coherently-demodulated received signal in the $n$-th interval be similarly named and defined:

$$\boldsymbol{y}_c(n,\boldsymbol{U}) \triangleq \left[y_c(nD_sT_s,\boldsymbol{U}) \quad y_c([nD_s+1]T_s,\boldsymbol{U}) \quad \cdots \quad y_c([(n+1)D_s-1]T_s,\boldsymbol{U})\right]^\dagger. \quad \text{(D.13)}$$

Following (C.3), we can write $\Pr\{\boldsymbol{y}_c(\boldsymbol{U}) \mid \tilde{s}(\boldsymbol{U})\}$ as

$$\Pr\{\boldsymbol{y}_c(\boldsymbol{U}) \mid \tilde{s}(\boldsymbol{U})\} = \prod_n \Pr\{\boldsymbol{y}_c(n,\boldsymbol{U}) \mid \tilde{s}(n,\boldsymbol{U})\}. \quad \text{(D.14)}$$

The probability density function (pdf) of $\Pr\{\boldsymbol{y}_c(n,\boldsymbol{U}) \mid \tilde{s}(n,\boldsymbol{U})\}$ is that of a $D_s$-dimension Gaussian random vector where each element $y_c(iT_s,\boldsymbol{U})$ has mean $\tilde{s}(iT_s,\boldsymbol{U})/\sqrt{2}$ and variance $2N_0f_w$, as discussed in Section D.1.1. Thus

$$\Pr\{\boldsymbol{y}_c(n,\boldsymbol{U}) \mid \tilde{s}(n,\boldsymbol{U})\} = \frac{1}{(4\pi N_0 f_w)^{D_s/2}} \exp\left(-\frac{1}{4\pi N_0 f_w}\sum_i \left|y_c(iT_s,\boldsymbol{U}) - \frac{1}{\sqrt{2}}\tilde{s}(iT_s,\boldsymbol{U})\right|^2\right). \quad \text{(D.15)}$$

Before proceeding, we need to identify the different possible transitions. In each symbol period there are $P \cdot M$ unique possible transitions, which we will call *reference signals*. Let $k'$ be an integer ranging from 1 to $P \cdot M$, used to enumerate all the possible memoryless modulator inputs $\boldsymbol{X}_{k'}$. The samples of the complex envelope of the $k'$-th reference signal in the $n$-th symbol period are defined as

$$\tilde{s}'(iT_s,\boldsymbol{X}_{k'}) \triangleq \sqrt{\frac{E}{T}} \exp\left(j\,2\pi\left[h\,X_{1,k'} + h\,X_{2,k'}\frac{iT_s - nT}{T} + f_0 iT_s\right]\right), \quad nT \le iT_s < (n+1)T, \quad \text{(D.16)}$$

The $D_s$ samples in the $n$-th symbol period are denoted

$$\tilde{s}'(n,\boldsymbol{X}_{k'}) = \left[\tilde{s}'(nD_sT_s,\boldsymbol{X}_{k'}) \quad \tilde{s}'([nD_s+1]T_s,\boldsymbol{X}_{k'}) \quad \cdots \quad \tilde{s}'([(n+1)D_s-1]T_s,\boldsymbol{X}_{k'})\right]^\dagger. \quad \text{(D.17)}$$

Following (C.5) and replacing the actual transition $\tilde{s}(n,\boldsymbol{U})$ with the hypothesised ones $\tilde{s}'(n,\boldsymbol{X}_{k'})$ in (D.15), we can write the $k'$-th branch metric in the $n$-th symbol period as

$$\begin{aligned}\lambda_{\mathrm{VA}}\left(\xi_{k_V},k'\right) &= \lambda_{\mathrm{VA}}\left(\tilde{s}'[n,\boldsymbol{X}_{k'}]\right)\\ &= -\ln\Pr\left\{\boldsymbol{y}_c(n,\boldsymbol{U}) \mid \tilde{s}'(n,\boldsymbol{X}_{k'})\right\}\\ &= \frac{D_s}{2}\ln(4\pi N_0 f_w) + \frac{1}{4\pi N_0 f_w}\sum_i \left|y_c(iT_s,\boldsymbol{U}) - \tilde{s}'(iT_s,\boldsymbol{X}_{k'})\right|^2. \quad \text{(D.18)}\end{aligned}$$

To minimise computation, we ignore the offset and scaling that are independent of $k'$ so that our branch metrics are

$$\lambda_{\mathrm{VA}}\left(\tilde{s}'[n, X_{k'}]\right) = \sum_{i=nD_s}^{(n+1)D_s-1} \left|y_{\mathrm{c}}(iT_s, \boldsymbol{U}) - \tilde{s}'(iT_s, X_{k'})\right|^2. \tag{D.19}$$

The Viterbi algorithm then proceeds as described in Appendix C to produce an estimate of the transmitted data sequence $\hat{\boldsymbol{U}}$.

## D.2 CPFSK Simulation Model

Figure D.3 shows the model used to generate $y_{\mathrm{c}}(iT_s, \boldsymbol{U})$ for simulations. A modified CPFSK transmitter generates samples of the complex envelope of the transmitted signal $\tilde{s}(iT_s, \boldsymbol{U})$, to which the independently identically distributed, zero-mean, white, complex Gaussian random variables $\tilde{w}(iT_s)$ are added. Note that the real and imaginary components of $\tilde{w}(iT_s)$ are independent and that each has a variance of $N_0 f_w$ as we assume that (D.5) holds. The output of the summer is then scaled by $1/\sqrt{2}$ to generate $y_{\mathrm{c}}(iT_s, \boldsymbol{U})$ as specified in (D.3). The scaling is not necessary, but ensures that the signal component of $y_{\mathrm{c}}(iT_s, \boldsymbol{U})$ has energy $E$. The Viterbi processor then uses $y_{\mathrm{c}}(iT_s, \boldsymbol{U})$ to produce an estimate of the transmitted data $\hat{\boldsymbol{U}}$, as described in Section D.1.2.

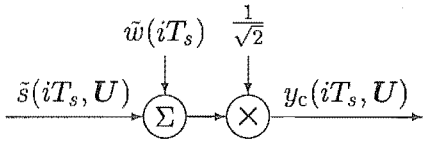



**Figure D.3:** Model used to produce $y_{\mathrm{c}}(iT_s, \boldsymbol{U})$ for simulations of CPFSK in AWGN

# Appendix E

# CPFSK Sampling Receiver Performance

The Viterbi algorithm in the receiver operates on the signal

$$y_c(iT_s, \boldsymbol{U}) \;=\; \frac{1}{\sqrt{2}}\left[\tilde{s}(iT_s, \boldsymbol{U}) + \tilde{w}(iT_s)\right]. \tag{E.1}$$

Note that the noise samples are zero-mean, white and Gaussian, with an autocorrelation given by

$$\mathrm{E}\left[\tilde{w}(iT_s)\tilde{w}^*(kT_s)\right] \;=\; 2N_0 f_w\, \delta_{i,k} \tag{E.2}$$

where $\delta_{i,k}$ was defined in (D.8). Another important property of $\tilde{w}(iT_s)$ is

$$\mathrm{E}\left[\tilde{w}(iT_s)\tilde{w}(kT_s)\right] \;=\; 0. \tag{E.3}$$

Let the transmitted data be given by $\boldsymbol{U}$ and the receiver's estimate of $\boldsymbol{U}$ be given by $\hat{\boldsymbol{U}}$. We now introduce two instances of a discrete metric that operates over the entire received signal. The first is based on the mean squared error (MSE) between the samples of the demodulated received signal $y_c(iT_s, \boldsymbol{U})$ and the samples of the complex envelope of the transmitted signal $\tilde{s}(iT_s, \hat{\boldsymbol{U}})$, which we define as

$$\Upsilon(y_c, \boldsymbol{U}) \;\triangleq\; \sum_{i=-\infty}^{\infty} \left| y_c(iT_s, \boldsymbol{U}) - \frac{1}{\sqrt{2}}\tilde{s}(iT_s, \boldsymbol{U}) \right|^2. \tag{E.4}$$

The second is based on the MSE between $y_c(iT_s, \boldsymbol{U})$ and $\tilde{s}(iT_s, \hat{\boldsymbol{U}})$ and defined as

$$\Upsilon(y_c, \hat{\boldsymbol{U}}) \;\triangleq\; \sum_{i=-\infty}^{\infty} \left| y_c(iT_s, \boldsymbol{U}) - \frac{1}{\sqrt{2}}\tilde{s}(iT_s, \hat{\boldsymbol{U}}) \right|^2. \tag{E.5}$$

As the VA chooses the metric with the smallest value, an error will occur if

$$\Upsilon(y_c, \boldsymbol{U}) \;>\; \Upsilon(y_c, \hat{\boldsymbol{U}}), \tag{E.6}$$

that is the distance between the received signal $y_c(iT_s, \boldsymbol{U})$ and a signal based on incorrect data $\hat{\boldsymbol{U}}$ is less than that between $y_c(iT_s, \boldsymbol{U})$ and the transmitted data $\boldsymbol{U}$. We are interested in the probability of this occurance, which is called the *pairwise probability of error*, and given by

$$\begin{aligned}\Pr\left\{\hat{\boldsymbol{U}} \neq \boldsymbol{U}\right\} &= \Pr\left\{\Upsilon(y_c, \boldsymbol{U}) > \Upsilon(y_c, \hat{\boldsymbol{U}})\right\} \\ &= \Pr\left\{\Upsilon(y_c, \boldsymbol{U}) - \Upsilon(y_c, \hat{\boldsymbol{U}}) > 0\right\} \\ &= \Pr\left\{\Upsilon_e > 0\right\}, \end{aligned} \tag{E.7}$$

where we have implicitly defined

$$\Upsilon_e \triangleq \Upsilon(y_c, \boldsymbol{U}) - \Upsilon(y_c, \hat{\boldsymbol{U}}). \tag{E.8}$$

We note now that we can develop (E.4) as

$$\begin{aligned}\Upsilon(y_c, \boldsymbol{U}) &= \sum_{i=-\infty}^{\infty} \left| y_c(iT_s, \boldsymbol{U}) - \frac{1}{\sqrt{2}} \tilde{s}(iT_s, \boldsymbol{U}) \right|^2 \\ &= \sum_{i=-\infty}^{\infty} \left\{ |y_c(iT_s, \boldsymbol{U})|^2 + \frac{1}{2} |\tilde{s}(iT_s, \boldsymbol{U})|^2 - \sqrt{2}\,\mathrm{Re}\left[y_c(iT_s, \boldsymbol{U})\, \tilde{s}^*(iT_s, \boldsymbol{U})\right] \right\} \\ &= \sum_{i=-\infty}^{\infty} |y_c(iT_s, \boldsymbol{U})|^2 + \frac{1}{2} \sum_{i=-\infty}^{\infty} |\tilde{s}(iT_s, \boldsymbol{U})|^2 - \sqrt{2} \sum_{i=-\infty}^{\infty} \mathrm{Re}\left[y_c(iT_s, \boldsymbol{U})\, \tilde{s}^*(iT_s, \boldsymbol{U})\right], \end{aligned} \tag{E.9}$$

and similarly, (E.5) can be re-written as

$$\Upsilon(y_c, \hat{\boldsymbol{U}}) = \sum_{i=-\infty}^{\infty} |y_c(iT_s, \boldsymbol{U})|^2 + \frac{1}{2} \sum_{i=-\infty}^{\infty} \left| \tilde{s}(iT_s, \hat{\boldsymbol{U}}) \right|^2 - \sqrt{2} \sum_{i=-\infty}^{\infty} \mathrm{Re}\left[ y_c(iT_s, \boldsymbol{U})\, \tilde{s}^*(iT_s, \hat{\boldsymbol{U}}) \right]. \tag{E.10}$$

Substituting (E.4) and (E.5) into (E.8), and noting that CPFSK signals are constant-envelope so that

$$|\tilde{s}(iT_s, \boldsymbol{U})|^2 = \left| \tilde{s}(iT_s, \hat{\boldsymbol{U}}) \right|^2, \tag{E.11}$$

we have

$$\begin{aligned}\Upsilon_e &= \sqrt{2} \sum_{i=-\infty}^{\infty} \mathrm{Re}\left[ y_c(iT_s, \boldsymbol{U})\, \tilde{s}^*(iT_s, \hat{\boldsymbol{U}}) \right] - \sqrt{2} \sum_{i=-\infty}^{\infty} \mathrm{Re}\left[ y_c(iT_s, \boldsymbol{U})\, \tilde{s}^*(iT_s, \boldsymbol{U}) \right] \\ &= \sqrt{2} \sum_{i=-\infty}^{\infty} \mathrm{Re}\left[ y_c(iT_s, \boldsymbol{U}) \{ \tilde{s}^*(iT_s, \hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s, \boldsymbol{U}) \} \right] \end{aligned} \tag{E.12}$$

Using (E.1) in (E.12), we obtain

$$\begin{aligned}\Upsilon_e &= \sum_{i=-\infty}^{\infty} \mathrm{Re}\left[\{\tilde{s}(iT_s,\boldsymbol{U})+\tilde{w}(iT_s)\}\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-\tilde{s}^*(iT_s,\boldsymbol{U})\}\right] \\ &= \sum_{i=-\infty}^{\infty} \mathrm{Re}\left[\tilde{s}(iT_s,\boldsymbol{U})\,\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-|\tilde{s}(iT_s,\boldsymbol{U})|^2+\tilde{w}(iT_s)\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-\tilde{s}^*(iT_s,\boldsymbol{U})\}\right].\end{aligned} \tag{E.13}$$

As $\tilde{w}(iT_s)$ is Gaussian, $\Upsilon_e$ is also Gaussian. Its mean is

$$\begin{aligned}\mathrm{E}[\Upsilon_e] &= \mathrm{E}\left[\sum_{i=-\infty}^{\infty} \mathrm{Re}\left[\tilde{s}(iT_s,\boldsymbol{U})\,\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-|\tilde{s}(iT_s,\boldsymbol{U})|^2+\tilde{w}(iT_s)\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-\tilde{s}^*(iT_s,\boldsymbol{U})\}\right]\right] \\ &= \sum_{i=-\infty}^{\infty} \mathrm{Re}\left[\tilde{s}(iT_s,\boldsymbol{U})\,\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-|\tilde{s}(iT_s,\boldsymbol{U})|^2+\mathrm{E}[\tilde{w}(iT_s)]\,\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-\tilde{s}^*(iT_s,\boldsymbol{U})\}\right] \\ &= \sum_{i=-\infty}^{\infty} \left\{\mathrm{Re}\left[\tilde{s}(iT_s,\boldsymbol{U})\,\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})\right]-|\tilde{s}(iT_s,\boldsymbol{U})|^2\right\}\end{aligned} \tag{E.14}$$

We now use (E.11) to note that

$$|\tilde{s}(iT_s,\boldsymbol{U})|^2 = \frac{1}{2}|\tilde{s}(iT_s,\boldsymbol{U})|^2+\frac{1}{2}\left|\tilde{s}(iT_s,\hat{\boldsymbol{U}})\right|^2, \tag{E.15}$$

which allows us to write (E.14) as

$$\begin{aligned}\mathrm{E}[\Upsilon_e] &= -\frac{1}{2}\sum_{i=-\infty}^{\infty}\left\{|\tilde{s}(iT_s,\boldsymbol{U})|^2+\left|\tilde{s}(iT_s,\hat{\boldsymbol{U}})\right|^2-2\,\mathrm{Re}\left[\tilde{s}(iT_s,\boldsymbol{U})\,\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})\right]\right\} \\ &= -\frac{1}{2}\sum_{i=-\infty}^{\infty}\left|\tilde{s}(iT_s,\boldsymbol{U})-\tilde{s}(iT_s,\hat{\boldsymbol{U}})\right|^2\end{aligned} \tag{E.16}$$

We now define the normalised MSE (NMSE) between the two signals $\tilde{s}(iT_s,\boldsymbol{U})$ and $\tilde{s}(iT_s,\hat{\boldsymbol{U}})$ as

$$\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}}) \triangleq \frac{r\,\log_2 M}{4E}\sum_{i=-\infty}^{\infty}\left|\tilde{s}(iT_s,\boldsymbol{U})-\tilde{s}(iT_s,\hat{\boldsymbol{U}})\right|^2. \tag{E.17}$$

Using (E.17), we can re-write (E.16) as

$$\begin{aligned}\mathrm{E}[\Upsilon_e] &= -\frac{2E}{r\,\log_2 M}\,\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}}) \\ &= -2E_b\,\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}}),\end{aligned} \tag{E.18}$$

as from (2.5), $E_b = E/(r\,\log_2 M)$.

Having found the mean of $\Upsilon_e$, we now turn our attention to its variance. We first note that from (E.13) and (E.14)

$$\begin{aligned}\Upsilon_e-\mathrm{E}[\Upsilon_e] &= \sum_{i=-\infty}^{\infty}\mathrm{Re}\left[\tilde{w}(iT_s)\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-\tilde{s}^*(iT_s,\boldsymbol{U})\}\right] \\ &= \frac{1}{2}\sum_{i=-\infty}^{\infty}\left[\tilde{w}(iT_s)\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}})-\tilde{s}^*(iT_s,\boldsymbol{U})\}+\tilde{w}^*(iT_s)\{\tilde{s}(iT_s,\hat{\boldsymbol{U}})-\tilde{s}(iT_s,\boldsymbol{U})\}\right].\end{aligned} \tag{E.19}$$

Using (E.19), the variance of $\Upsilon_e$ is given by

$$
\begin{aligned}
\mathrm{Var}\{\Upsilon_e\} &= \mathrm{E}\left[(\Upsilon_e - \mathrm{E}[\Upsilon_e])^2\right] \\
&= \frac{1}{4}\,\mathrm{E}\left[\left(\sum_{i=-\infty}^{\infty}\left[\tilde{w}(iT_s)\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\} + \tilde{w}^*(iT_s)\{\tilde{s}(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}(iT_s,\boldsymbol{U})\}\right]\right)^2\right] \\
&= \frac{1}{4}\,\mathrm{E}\left[\left(\sum_{i=-\infty}^{\infty}\tilde{w}(iT_s)\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\} + \sum_{i=-\infty}^{\infty}\tilde{w}^*(iT_s)\{\tilde{s}(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}(iT_s,\boldsymbol{U})\}\right)\right. \\
&\quad \left.\times\left(\sum_{k=-\infty}^{\infty}\tilde{w}(kT_s)\{\tilde{s}^*(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(kT_s,\boldsymbol{U})\} + \sum_{k=-\infty}^{\infty}\tilde{w}^*(kT_s)\{\tilde{s}(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}(kT_s,\boldsymbol{U})\}\right)\right] \\
&= \frac{1}{4}\,\mathrm{E}\left[\sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\tilde{w}(iT_s)\tilde{w}(kT_s)\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\}\{\tilde{s}^*(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(kT_s,\boldsymbol{U})\}\right. \\
&\qquad + \sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\tilde{w}(iT_s)\tilde{w}^*(kT_s)\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\}\{\tilde{s}(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}(kT_s,\boldsymbol{U})\} \\
&\qquad + \sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\tilde{w}^*(iT_s)\tilde{w}(kT_s)\{\tilde{s}(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}(iT_s,\boldsymbol{U})\}\{\tilde{s}^*(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(kT_s,\boldsymbol{U})\} \\
&\qquad \left. + \sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\tilde{w}^*(iT_s)\tilde{w}^*(kT_s)\{\tilde{s}(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}(iT_s,\boldsymbol{U})\}\{\tilde{s}(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}(kT_s,\boldsymbol{U})\}\right] \\
&= \frac{1}{4}\sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\mathrm{E}\left[\tilde{w}(iT_s)\tilde{w}(kT_s)\right]\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\}\{\tilde{s}^*(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(kT_s,\boldsymbol{U})\} \\
&\quad + \sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\mathrm{E}\left[\tilde{w}(iT_s)\tilde{w}^*(kT_s)\right]\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\}\{\tilde{s}(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}(kT_s,\boldsymbol{U})\} \\
&\quad + \sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\mathrm{E}\left[\tilde{w}^*(iT_s)\tilde{w}(kT_s)\right]\{\tilde{s}(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}(iT_s,\boldsymbol{U})\}\{\tilde{s}^*(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(kT_s,\boldsymbol{U})\} \\
&\quad + \sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\mathrm{E}\left[\tilde{w}^*(iT_s)\tilde{w}^*(kT_s)\right]\{\tilde{s}(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}(iT_s,\boldsymbol{U})\}\{\tilde{s}(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}(kT_s,\boldsymbol{U})\} \\
&= \frac{1}{4}\sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}2N_0 f_w\,\delta_{i,k}\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\}\{\tilde{s}(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}(kT_s,\boldsymbol{U})\} \\
&\quad + \sum_{i=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}2N_0 f_w\,\delta_{i,k}\{\tilde{s}(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}(iT_s,\boldsymbol{U})\}\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\} \\
&= N_0 f_w\sum_{i=-\infty}^{\infty}\{\tilde{s}^*(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}^*(iT_s,\boldsymbol{U})\}\{\tilde{s}(kT_s,\hat{\boldsymbol{U}}) - \tilde{s}(kT_s,\boldsymbol{U})\} \\
&= N_0 f_w\sum_{i=-\infty}^{\infty}\left|\tilde{s}(iT_s,\hat{\boldsymbol{U}}) - \tilde{s}(iT_s,\boldsymbol{U})\right|^2 \\
&= 4N_0 f_w E_b\,\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}}). \qquad \text{(E.20)}
\end{aligned}
$$

Let us use $\mu_{\Upsilon_e}$ to denote $\mathrm{E}[\Upsilon_e]$, and $\sigma^2_{\Upsilon_e}$ to denote $\mathrm{Var}\{\Upsilon_e\}$. The probability density

function (pdf) of $\Upsilon_e$ is a Normal distribution, given by

$$\Pr\{\Upsilon_e\} = \frac{1}{\sqrt{2\pi}\sigma_{\Upsilon_e}} \exp\left(-\frac{(\Upsilon_e - \mu_{\Upsilon_e})^2}{2\,\sigma_{\Upsilon_e}^2}\right). \tag{E.21}$$

The pairwise probability of error can then be written as

$$\Pr\{\Upsilon_e > 0\} = \frac{1}{\sqrt{2\pi}\sigma_{\Upsilon_e}} \int_0^\infty \exp\left(-\frac{(\Upsilon_e - \mu_{\Upsilon_e})^2}{2\,\sigma_{\Upsilon_e}^2}\right) d\Upsilon_e. \tag{E.22}$$

We wish to get this into a standard form. Let

$$z = \frac{\Upsilon_e - \mu_{\Upsilon_e}}{\sigma_{\Upsilon_e}}, \tag{E.23}$$

so that

$$d\Upsilon_e = \sigma_{\Upsilon_e}\, dz \tag{E.24}$$

Note that as $\Upsilon_e \to \infty$, $z \to \infty$, and

$$\text{when } \Upsilon_e = 0, \quad z = \frac{-\mu_{\Upsilon_e}}{\sigma_{\Upsilon_e}}. \tag{E.25}$$

Using (E.23)-(E.25) in (E.22), we obtain

$$\begin{aligned}\Pr\{\Upsilon_e > 0\} &= \frac{1}{\sqrt{2\pi}} \int_{-\mu_{\Upsilon_e}/\sigma_{\Upsilon_e}}^\infty \exp\left(-\frac{z^2}{2}\right) dz \\ &= Q\left(\frac{-\mu_{\Upsilon_e}}{\sigma_{\Upsilon_e}}\right),\end{aligned} \tag{E.26}$$

where $Q(x)$ is called the $Q$-function and defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{z^2}{2}\right) dz. \tag{E.27}$$

Substituting (E.18) and (E.20) into (E.26), we find the pairwise probability of error between two CPFSK signals based on the data sequences $\boldsymbol{U}$ and $\hat{\boldsymbol{U}}$, is given by

$$\begin{aligned}\Pr\{\Upsilon_e > 0\} &= Q\left(\frac{2E_b\,\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}})}{\sqrt{4N_0 f_w E_b\,\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}})}}\right) \\ &= Q\left(\sqrt{\frac{E_b}{N_0}\,\frac{\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}})}{f_w}}\right).\end{aligned} \tag{E.28}$$

Let us look at the quantity $\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}})/f_w$. As we have specified that $f_w = 1/T_s$, we have

$$\frac{\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}})}{f_w} = \varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}})\,T_s = \frac{r\,\log_2 M}{4E} \sum_{i=-\infty}^{\infty} \left|\tilde{s}(iT_s,\boldsymbol{U}) - \tilde{s}(iT_s,\hat{\boldsymbol{U}})\right|^2 T_s. \tag{E.29}$$

As we increase the sampling frequency, $T_s \to 0$, so that the summation approaches an integration, and

$$\frac{\varepsilon^2(\boldsymbol{U},\hat{\boldsymbol{U}})}{f_w} \to \frac{r \log_2 M}{4E} \int_{-\infty}^{\infty} \left|\tilde{s}(t,\boldsymbol{U}) - \tilde{s}(t,\hat{\boldsymbol{U}})\right|^2 dt = d^2(\boldsymbol{U},\hat{\boldsymbol{U}}). \tag{E.30}$$

Using (E.30) in (E.28), we obtain pairwise probability of error for a DCPFSK system is

$$\Pr\{\Upsilon_e > 0\} = Q\left(\sqrt{\frac{E_b}{N_0} d^2(\boldsymbol{U},\hat{\boldsymbol{U}})}\right). \tag{E.31}$$

The performance of the receiver will be dominated by the pairwise probability of error of the minimum distance error event $d^2_{\min}$. Thus probability of error for our receiver is

$$P_e = Q\left(\sqrt{\frac{E_b}{N_0} d^2_{\min}}\right), \tag{E.32}$$

which agrees with the result in Section 2.9.

# Appendix F

# DCPFSK Simulations

Simulations on a computer must be performed in discrete-time. In this appendix we describe our DCPFSK simulation model and a sampling receiver for DCFPSK, which relates much more accuarately to the simulations that were performed.

## F.1 A Sampling Receiver Structure for DCPFSK

Our sampling receiver for DCPFSK consists of three main blocks, a differential demodulator, a sampler, and a Viterbi processor as shown in Figure F.1. The differential demodulator is exactly the same as that described in Section F.1. We now discuss the other two elements.

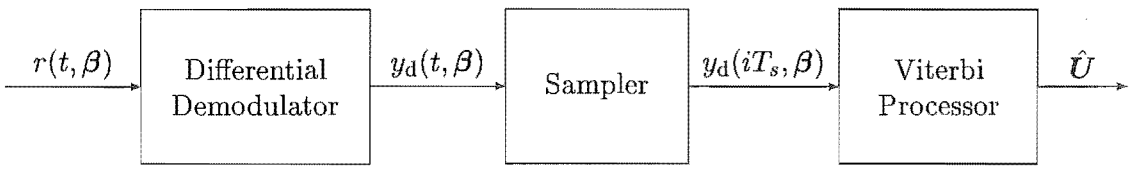



**Figure F.1:** A sampling receiver structure for DCPFSK

### F.1.1 Sampler

The output of the differential demodulator, $y_{\mathrm{d}}(t,\beta)$ is sampled at $r_s$ Hz. The sample period $T_s$ is equal to $1/r_s$. To ensure that we have an integral number of samples per symbol, $T_s$ is chosen such that

$$\frac{T}{T_s} = D_s, \tag{F.1}$$

where $D_s$ is a positive integer. The sampled version of $y_{\rm d}(t,\beta)$ is $y_{\rm d}(iT_s,\beta)$, given by

$$y_{\rm d}(iT_s,\beta) = \frac{1}{2}\sqrt{\frac{T}{E}}\Big[\tilde{s}(iT_s,\beta)\,\tilde{s}^*(iT_s-T,\beta) + \tilde{w}(iT_s)\,\tilde{s}^*(iT_s-T,\beta) + \tilde{w}^*(iT_s-T)\,\tilde{s}(iT_s,\beta) + \tilde{w}(iT_s)\,\tilde{w}^*(iT_s-T)\Big]. \tag{F.2}$$

### F.1.2 Viterbi Processor

Our Viterbi processor seeks to minimise the squared Euclidean distance between the received signal and the possible transmitted signals (see Section 3.7). We use the trellis structure of $y_{\rm d}(iT_s,\beta)$ discussed in Section 3.6 to estimate the transmitted data. Let us look at the noise-free version of $y_{\rm d}(iT_s,\beta)$. From (3.13) this is given by

$$\begin{aligned} y_{\rm d}(iT_s,\beta) &= \frac{1}{2}\sqrt{\frac{T}{E}}\,\tilde{s}(iT_s,\beta)\,\tilde{s}^*(iT_s-T,\beta) \\ &= \sqrt{\frac{E}{T}}\exp\left(j\,2\pi h\left[\beta_{n-1}+(\beta_n-\beta_{n-1})\frac{iT_s-nT}{T}\right]\right), \quad nT \le iT_s < (n+1)T. \end{aligned} \tag{F.3}$$

In the $n$-th symbol period this depends only on $\beta_n$ and $\beta_{n-1}$. As discussed in Section 3.6 there are only possible $M \times P$ combinations of $\beta_n - \beta_{n-1}$ and $\beta_{n-1}$ that produce *unique* values of the signal × signal term. Let us use $k'$ to enumerate these possible combinations, and the $2 \times 1$ element vector $\breve{\boldsymbol{X}}_{k'}$ to identify them. We denote the samples of the complex envelope of the $k'$-th reference signal as $\tilde{s}'(iT_s,\breve{\boldsymbol{X}}_{k'})$ and define them as

$$\tilde{s}'(iT_s,\breve{\boldsymbol{X}}_{k'}) \triangleq \sqrt{\frac{E}{T}}\exp\left(j\,2\pi h\left[\breve{X}_{k'}^{(2)}+\breve{X}_{k'}^{(1)}\frac{iT_s-nT}{T}\right]\right), \quad nT \le iT_s < (n+1)T. \tag{F.4}$$

The $D_s$ samples in the $n$-th symbol period are denoted as

$$\tilde{\boldsymbol{s}}'(n,\breve{\boldsymbol{X}}_{k'}) = \begin{bmatrix} \tilde{s}'(nD_sT_s,\breve{\boldsymbol{X}}_{k'}) \\ \tilde{s}'([nD_s+1]T_s,\breve{\boldsymbol{X}}_{k'}) \\ \vdots \\ \tilde{s}'([(n+1)D_s-1]T_s,\breve{\boldsymbol{X}}_{k'}) \end{bmatrix}. \tag{F.5}$$

Using (F.4) and (F.5), the $k'$-th branch metric in the $n$-th symbol period is defined as

$$\lambda_{\rm VA}\left(\tilde{\boldsymbol{s}}'[n,\breve{\boldsymbol{X}}_{k'}]\right) = \sum_{i=nD_s}^{(n+1)D_s-1}\left|y_{\rm d}(iT_s,\beta)-\tilde{s}'(iT_s,\breve{\boldsymbol{X}}_{k'})\right|^2. \tag{F.6}$$

The Viterbi algorithm then proceeds as described in Appendix C to produce an estimate of the transmitted data sequence $\hat{\boldsymbol{U}}$.

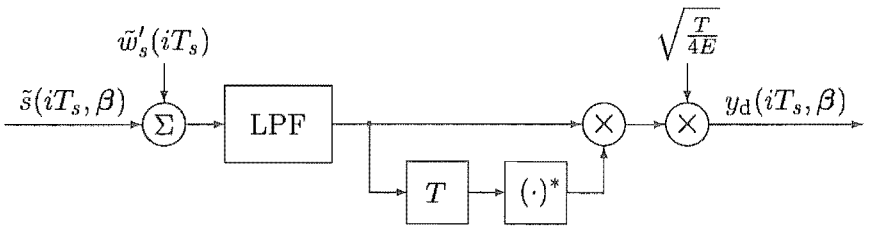



**Figure F.2:** Model used to produce $y_{\mathrm{d}}(iT_s,\beta)$ for simulations of DCPFSK in AWGN

## F.2 DCPFSK Simulation Models

The simulation of a DCPFSK system is a much more complicated problem than that of its coherent counterpart. Figure F.2 illustrates how $y_{\mathrm{d}}(iT_s,\beta)$ was generated for the AWGN simulations. A modified DCPFSK transmitter generates samples of the complex envelope of the transmitted signal $\tilde{s}(iT_s,\beta)$, to which the independently identically distributed, zero-mean, white, complex Gaussian random variables $\tilde{w}'_s(iT_s)$ are added. The real and imaginary components of $\tilde{w}'_s(iT_s)$ are independent and each has a variance of $N_0 r_s$. It was found that a low-pass filter (LPF) was needed to obtain the best possible performance in the simulations. Note that in our receiver model, the LPF would be a band-pass filter before the differential demodulator. For the LPF, we chose to use a $2D_s$-tap finite impulse response (FIR) filter whose transfer function is a Hamming window with a cut-off frequency equal to $f_{\mathrm{B}}/2$. The output of the LPF is multiplied by a copy that has been delayed by $T$ and conjugated. The result is scaled by $\sqrt{T/(4E)}$ to produce $y_{\mathrm{d}}(iT_s,\beta)$ as specified in (F.2). The scaling is not necessary, but ensures that the signal $\times$ signal component of $y_{\mathrm{d}}(iT_s,\beta)$ has energy of $E$. The Viterbi algorithm uses $y_{\mathrm{d}}(iT_s,\beta)$ to produce an estimate of the transmitted data $\hat{\boldsymbol{U}}$, as described in Section F.1.2. It is important to note that the simulations do not use the assumptions discussed in Section 3.10, and thus they take into account the effect of noise $\times$ noise and signal $\times$ noise terms.

The choice of $f_{\mathrm{B}}$ is important as it significantly affects performance. The power in the noise $\times$ noise term increases as $f_{\mathrm{B}}$ is increased, but decreasing $f_{\mathrm{B}}$ damages the desired signal $\times$ signal term, and the best-performing values of $f_{\mathrm{B}}$ were found empirically. Table F.1 shows the choice of $f_{\mathrm{B}}$ for each of the cases shown in Figure 3.12.

The simulation model used to generate $y_{\mathrm{d}}(iT_s,\beta)$ for the fading simulations is shown in Figure F.3, which is very similar to that in Figure F.2, the only difference being the mul-

**Table F.1**

Normalised cut-off frequency of noise-limiting filter for various DCPFSK schemes

| Scheme | $f_B T$ |
|---|---|
| DMSK | 1.0 |
| 4-DCPFSK | 2.5 |
| 8-DCPFSK | 2.5 |

tiplication of $\tilde{s}(iT_s, \beta)$ by $\tilde{z}(iT_s)$. The samples of the fading process $\tilde{z}(iT_s)$ were generated using the model in [Jak74], which simply sums up a number of appropriately-weighted offset oscillators. It was found that 15 oscillators produced an adequate representation of a Rayleigh flat fading process.

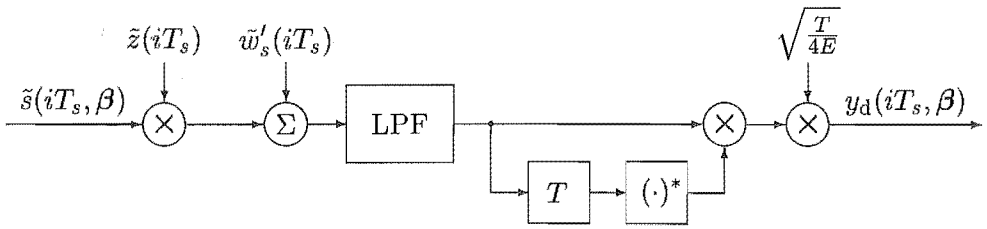



**Figure F.3:** Model used to produce $y_d(iT_s, \beta)$ for simulations of DCPFSK in Rayleigh flat fading

# Appendix G

# Full Code Search Results

In this appendix we present the full results of the code searches. When more than one code is given for a particular set of parameters, the given codes have the same $d^2_{\text{min}}$ to four decimal places. Tables G.1-G.3 list the best codes found for rate-1/2 encoded 4-CPFSK and 4-DCPFSK. For rate-2/3 codes, it was found that the codes performed exactly the same for both 8-CPFSK and 8-DCPFSK, so the results in Table G.4 are not separated for the coherent and differential cases.

It must be noted that these are the best codes for CPFSK and DCPFSK systems using feedback-free continuous phase encoders (CPE's), for standard CPFSK and DCPFSK systems (systems employing feedback CPE's), the given codes $\mathbf{G}(D)$ should be scrambled by $\mathbf{T}_l(D)$ to produce

$$\dot{\mathbf{G}}(D) = \mathbf{G}(D) \cdot \mathbf{T}_l(D), \tag{G.1}$$

which are suitable for feedback systems. In the case of the rate-1/2 coded quaternary systems, the form of the scrambler is

$$\mathbf{T}_2(D) = \begin{bmatrix} 1 & 3 \\ 3D & 1 \end{bmatrix}, \tag{G.2}$$

and the form of the scrambler for the rate-2/3 coded octal systems is

$$\mathbf{T}_3(D) = \begin{bmatrix} 1 & 7 & 0 \\ 0 & 1 & 7 \\ 7D & 0 & 1 \end{bmatrix}. \tag{G.3}$$

**Table G.1**

Search results for rate-1/2 encoded 4-CPFSK and 4-DCPFSK

| | | Coherent | | Differential | |
|---|---|---|---|---|---|
| $S_V$ | $\nu$ | $d^2_{\min}$ | $\mathbf{G}(D)$ | $d^2_{\min}$ | $\mathbf{G}(D)$ |
| 4 | 1 | 3.15 | $\left[1 \quad \frac{1}{2D+1}\right]$, $\left[\frac{1}{2D+1} \quad 1\right]$, $\left[D+2 \quad 1\right]$, $\left[1 \quad 2D+1\right]$, $\left[2D+1 \quad 1\right]$ | 3.00 | $\left[1 \quad \frac{1}{2D+1}\right]$, $\left[\frac{1}{2D+1} \quad 1\right]$, $\left[D+2 \quad 1\right]$, $\left[1 \quad 2D+1\right]$, $\left[2D+1 \quad 1\right]$ |
| 8 | 1 | 4.09 | $\left[\frac{D+1}{2D+1} \quad 1\right]$, $\left[1 \quad \frac{2D+1}{D+1}\right]$ | 4.00 | $\left[\frac{D+1}{2D+1} \quad 1\right]$, $\left[1 \quad \frac{2D+1}{D+1}\right]$ |
| 16 | 2 | 5.15 | $\left[\frac{3D+2}{2D^2+3D+1} \quad 1\right]$, $\left[\frac{2D^2+3D+2}{D+1} \quad 1\right]$ | 4.94 | $\left[1 \quad \frac{D+2}{2D^2+3D+1}\right]$, $\left[\frac{D^2+3D+2}{2D+1} \quad 1\right]$, $\left[1 \quad \frac{2D^2+D+2}{D+1}\right]$ |
| 32 | 3 | 6.00 | $\left[\frac{D+2}{2D^3+D+1} \quad 1\right]$, $\left[\frac{2D^2+D+2}{2D^3+2D^2+3D+1} \quad 1\right]$, $\left[\frac{2D^3+D+2}{2D^2+D+1} \quad 1\right]$, $\left[\frac{2D^3+2D^2+D+2}{3D+1} \quad 1\right]$ | 5.45 | 16 entries, see Table G.2 |
| 64 | 3 | 6.42 | $\left[\frac{D^3+2D^2+3}{2D^3+D^2+2D+1} \quad 1\right]$, $\left[1 \quad \frac{2D^3+D^2+3}{D^3+2D+1}\right]$, $\left[1 \quad \frac{2D^3+3D^2+2D^2+3}{3D^3+2D^2+1}\right]$, $\left[\frac{3D^3+2D+3}{2D^3+3D^2+1} \quad 1\right]$ | 6.39 | $\left[1 \quad \frac{D^2+D+2}{2D^3+D^2+3D+1}\right]$, $\left[\frac{D^3+3D^2+D^2+2}{2D^2+D+1} \quad 1\right]$, $\left[1 \quad \frac{2D^3+3D^2+D^2+2}{3D^2+D+1}\right]$ |
| 128 | 4 | 7.60 | $\left[\frac{D^4+3D^2+2D+1}{2D^4+D^3+2D^2+1} \quad 1\right]$, $\left[\frac{D^4+2D^3+D^2+1}{2D^4+D^3+2D+1} \quad 1\right]$, $\left[1 \quad \frac{2D^4+D^3+2D+1}{D^4+2D^3+D^2+1}\right]$, $\left[1 \quad \frac{2D^4+D^3+2D^2+1}{D^4+3D^2+2D+1}\right]$, $\left[1 \quad \frac{2D^4+3D^3+1}{3D^4+2D^3+D^2+2D+1}\right]$, $\left[1 \quad \frac{2D^4+3D^3+2D^2+2D+1}{3D^4+3D^2+1}\right]$, $\left[\frac{3D^4+3D^2+1}{2D^4+3D^3+2D^2+2D+1} \quad 1\right]$, $\left[\frac{3D^4+2D^3+D^2+2D+1}{2D^4+3D^3+1} \quad 1\right]$ | 7.00 | 38 entries, see Table G.3 |

**Table G.2**

Search results for rate-1/2 encoded 4-DCPFSK with $\nu = 3$, $S_V = 32$ and $d^2_{\min} = 5.45$

| G($D$) |
|---|
| $\left[1 \quad \frac{D+1}{2D^3+D^2+3D+1}\right]$, $\left[1 \quad \frac{3D+1}{2D^3+D^2+3D+1}\right]$, $\left[1 \quad \frac{2D^2+D+1}{3D^2+D+1}\right]$, $\left[1 \quad \frac{2D^2+3D+1}{3D^2+D+1}\right]$, |
| $\left[\frac{3D^2+D+1}{2D^2+D+1} \quad 1\right]$, $\left[\frac{3D^2+D+1}{2D^2+3D+1} \quad 1\right]$, $\left[\frac{D^3+2D^2+1}{2D^3+D^2+1} \quad 1\right]$, $\left[\frac{D^3+2D^2+1}{2D^3+3D^2+2D+1} \quad 1\right]$, |
| $\left[1 \quad \frac{2D^3+D^2+1}{D^3+2D^2+1}\right]$, $\left[1 \quad \frac{2D^3+D^2+1}{3D^3+2D+1}\right]$, $\left[\frac{2D^3+D^2+3D+1}{D+1} \quad 1\right]$, $\left[\frac{2D^3+D^2+3D+1}{3D+1} \quad 1\right]$, |
| $\left[1 \quad \frac{2D^3+3D^2+2D+1}{D^3+2D^2+1}\right]$, $\left[1 \quad \frac{2D^3+3D^2+2D+1}{3D^3+2D+1}\right]$, $\left[\frac{3D^3+2D+1}{2D^3+D^2+1} \quad 1\right]$, $\left[\frac{3D^3+2D+1}{2D^3+3D^2+2D+1} \quad 1\right]$ |

**Table G.3**

Search results for rate-1/2 encoded 4-DCPFSK with $\nu = 4$, $S_V = 128$ and $d^2_{\min} = 7.00$

| G($D$) |
|---|
| $\left[1 \quad \frac{D^2+D+2}{2D^4+2D^3+D^2+3D+1}\right]$, $\left[1 \quad \frac{D^2+3D+2}{2D^4+2D^3+3D^2+D+1}\right]$, $\left[1 \quad \frac{D^3+3D+2}{2D^4+D^3+2D^2+2D+1}\right]$, |
| $\left[1 \quad \frac{2D^3+D+2}{2D^4+3D^3+2D^2+D+1}\right]$, $\left[1 \quad \frac{2D^3+D^2+D+2}{3D^4+2D^3+D^2+D+1}\right]$, $\left[1 \quad \frac{2D^3+D^2+3D+2}{D^4+3D^3+D^2+1}\right]$, |
| $\left[1 \quad \frac{2D^3+2D^2+D+2}{3D^3+3D+1}\right]$, $\left[1 \quad \frac{2D^3+3D^2+D+2}{D^4+D^2+3D+1}\right]$, $\left[1 \quad \frac{2D^3+3D^2+D+2}{2D^4+3D^2+D+1}\right]$, |
| $\left[1 \quad \frac{2D^3+3D^2+3D+2}{2D^4+D^2+3D+1}\right]$, $\left[1 \quad \frac{2D^3+3D^2+3D+2}{3D^4+3D^3+3D^2+2D+1}\right]$, $\left[\frac{3D^3+D+2}{2D^3+3D^2+D+1} \quad 1\right]$, |
| $\left[\frac{3D^3+D^2+D+2}{2D^3+3D^2+2D+1} \quad 1\right]$, $\left[\frac{3D^3+2D^2+D+2}{2D^3+3D^2+3D+1} \quad 1\right]$, $\left[1 \quad \frac{3D^3+2D^2+3D+2}{2D^4+3D^3+1}\right]$, |
| $\left[\frac{3D^3+3D^2+D+2}{2D^3+3D^2+1} \quad 1\right]$, $\left[\frac{D^4+2D^2+3D+2}{2D^4+3D^3+3D^2+3D+1} \quad 1\right]$, $\left[\frac{D^4+D^3+D^2+3D+2}{2D^4+3D^3+2D+1} \quad 1\right]$, |
| $\left[\frac{D^4+2D^3+3D+2}{2D^4+3D^3+D^2+D+1} \quad 1\right]$, $\left[\frac{D^4+2D^3+2D^2+D+2}{2D^3+3D^2+1} \quad 1\right]$, $\left[\frac{D^4+3D^3+3D^2+3D+2}{2D^4+3D^3+2D^2+1} \quad 1\right]$, |
| $\left[1 \quad \frac{2D^4+D^2+3D+2}{D^4+3D^3+D^2+2D+1}\right]$, $\left[1 \quad \frac{2D^4+3D^2+D+2}{2D^3+D^2+D+1}\right]$, $\left[1 \quad \frac{2D^4+3D^2+3D+2}{2D^3+3D^2+3D+1}\right]$, |
| $\left[1 \quad \frac{2D^4+3D^2+3D+2}{3D^4+3D^3+3D^2+1}\right]$, $\left[\frac{2D^4+D^3+D^2+D+2}{3D^2+2D+1} \quad 1\right]$, $\left[1 \quad \frac{2D^4+D^3+2D^2+3D+2}{D^3+2D^2+1}\right]$, |
| $\left[\frac{2D^4+D^3+3D^2+D+2}{3D^2+1} \quad 1\right]$, $\left[1 \quad \frac{2D^4+2D^3+D^2+D+2}{3D^2+3D+1}\right]$, $\left[1 \quad \frac{2D^4+2D^3+D^2+3D+2}{D^2+D+1}\right]$, |
| $\left[\frac{2D^4+3D^3+D+2}{D^2+D+1} \quad 1\right]$, $\left[1 \quad \frac{2D^4+3D^3+3D+2}{3D^3+2D+1}\right]$, $\left[\frac{2D^4+3D^3+2D^2+D+2}{D^2+3D+1} \quad 1\right]$, |
| $\left[\frac{3D^4+D+2}{2D^3+D^2+2D+1} \quad 1\right]$, $\left[\frac{3D^4+2D^2+3D+2}{2D^4+D^3+D^2+3D+1} \quad 1\right]$, $\left[\frac{3D^4+D^3+3D^2+3D+2}{2D^4+D^3+2D^2+1} \quad 1\right]$, |
| $\left[\frac{3D^4+2D^3+3D+2}{2D^4+D^3+3D^2+D+1} \quad 1\right]$, $\left[\frac{3D^4+3D^3+D^2+3D+2}{2D^4+D^3+2D+1} \quad 1\right]$ |

**Table G.4**

Search results for rate-2/3 encoded 8-CPFSK and 8-DCPFSK
(the * indicates that the search was not complete)

| $S_V$ | $\nu$ | $d^2_{\min}$ | $\mathbf{G}(D)$ |
|---|---|---|---|
| 8 | 1 | 2.18 | $\begin{bmatrix} 1 & 0 & 4D+6 \\ 0 & 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 & \frac{4D+6}{D+1} \\ 0 & 1 & \frac{2}{D+1} \end{bmatrix}$, $\begin{bmatrix} 1 & 0 & \frac{4D+6}{3D+1} \\ 0 & 1 & \frac{2}{3D+1} \end{bmatrix}, \begin{bmatrix} 1 & 0 & \frac{4D+6}{4D+1} \\ 0 & 1 & \frac{2}{4D+1} \end{bmatrix}$, $\begin{bmatrix} 1 & 0 & \frac{4D+6}{5D+1} \\ 0 & 1 & \frac{2}{5D+1} \end{bmatrix}, \begin{bmatrix} 1 & 0 & \frac{4D+6}{7D+1} \\ 0 & 1 & \frac{2}{7D+1} \end{bmatrix}$, $\begin{bmatrix} 1 & 0 & \frac{6}{2D+1} \\ 0 & 1 & \frac{4D+2}{2D+1} \end{bmatrix}, \begin{bmatrix} 1 & 0 & \frac{6}{3D+1} \\ 0 & 1 & \frac{4D+2}{3D+1} \end{bmatrix}$, $\begin{bmatrix} 1 & 0 & \frac{6}{6D+1} \\ 0 & 1 & \frac{4D+2}{6D+1} \end{bmatrix}, \begin{bmatrix} 1 & 0 & \frac{6}{7D+1} \\ 0 & 1 & \frac{4D+2}{7D+1} \end{bmatrix}$ |
| 16 | 1 | 2.81 | $\begin{bmatrix} 1 & 0 & 6D+4 \\ 0 & 1 & 4D+2 \end{bmatrix}, \begin{bmatrix} 1 & 0 & \frac{6D+4}{4D+1} \\ 0 & 1 & \frac{4D+2}{4D+1} \end{bmatrix}$ |
| 32 | 2 | 2.93* | $\begin{bmatrix} 1 & 0 & \frac{2D+4}{3D^2+4D+1} \\ 0 & 1 & \frac{D+1}{3D^2+4D+1} \end{bmatrix}$ |

# Appendix H

# Glossary of Abbreviations

| | |
|---|---|
| AWGN | additve white Gaussian noise |
| BER | bit error rate |
| BPF | band-pass filter |
| CE | channel encoder |
| CPE | continuous phase encoder |
| CPFSK | continuous phase frequency shift keying |
| CPM | continuous phase modulation |
| DCPE | differential continuous phase encoder |
| DCPED | differential continuous phase encoder/decoder |
| DCPFSK | differentially-encoded and differentially-demodulated CPFSK |
| DCPM | differentially-encoded and differentially-demodulated CPM |
| DMSK | differential minimum shift keying |
| DPSK | differential phase shift keying |
| ECC | error-control coding |
| FSK | frequency shift keying |
| GSM | Global System for Mobile Communications |
| GMSK | Gaussian minimum shift keying |
| LHS | left-hand side |
| LPF | low-pass filter |
| $M$-CPFSK | $M$-ary CPFSK with $h = 1/M$ |
| $M$-DCPFSK | $M$-ary DCPFSK with $h = 1/M$ |
| MLSE | maximum-likelihood sequence estimation |
| MM | memoryless modulator |
| MSK | minimum shift keying |
| NISED | normalised incremental squared Euclidean distance |
| NMSED | normalised minimum squared Euclidean distance |
| NSED | normalised squared Euclidean distance |
| pdf | probability density function |
| psd | power spectral density |
| PPE | pairwise probability of error |
| PSK | phase shift keying |
| RHS | right-hand side |
| SED | squared Euclidean distance |
| SNR | signal-to-noise ratio |

# Appendix I

# Glossary of Symbols

| | |
|---|---|
| $\boldsymbol{a}_{k_V}$ | vector of $l-1$ uncoded $M$-ary symbols input to the channel encoder in the $k_V$-th trellis interval |
| $\mathbf{A}$ | matrix in Gaussian quadratic form |
| $A_i$ | elements of $\mathbf{A}$ |
| $\boldsymbol{b}_{k_V}$ | vector of $l$ coded $M$-ary symbols output by the channel encoder in the $k_V$-th trellis interval |
| $B$ | modulo base of the differential encoder |
| $\boldsymbol{c}_{k_V}$ | vector of $2l$ coded $M$-ary symbols output by $\dot{\mathbf{C}}_l(D)$ in the $k_V$-th trellis interval |
| $\check{\boldsymbol{c}}_{k_V}$ | vector of $2l$ coded $M$-ary symbols output by $\dot{\mathbf{F}}_l(D)$ in the $k_V$-th trellis interval |
| $\mathbf{C}(D)$ | transfer function of the feedback CPE |
| $\mathbf{C}_l(D)$ | equivalent $l \times 2l$ version of $\mathbf{C}(D)$ |
| $\dot{\mathbf{C}}(D)$ | transfer function of the feedback-free CPE |
| $\dot{\mathbf{C}}_l(D)$ | equivalent $l \times 2l$ version of $\dot{\mathbf{C}}(D)$ |
| $d^2(\boldsymbol{U}, \hat{\boldsymbol{U}})$ | normalised SED between two CPFSK signals, $s(t, \boldsymbol{U})$ and $s(t, \hat{\boldsymbol{U}})$ |
| $d_n^2(\boldsymbol{U}, \hat{\boldsymbol{U}})$ | normalised incremental SED between two CPFSK signals, $s(t, \boldsymbol{U})$ and $s(t, \hat{\boldsymbol{U}})$ in the $n$-th symbol period |
| $d_{\min}^2$ | normalised minimum SED of a (D)CPFSK scheme |
| $D$ | a dummy variable representing delay |
| $D_s$ | number of samples per symbol (integer value) |
| $\mathcal{D}$ | the set of all NSEDs for a particular code |
| $E$ | symbol energy |
| $E_b$ | bit energy |
| $\mathcal{E}_{\mathcal{S},i}$ | the $i$-th error event starting from the state $\mathcal{S}$ |
| $\mathbf{E}(D)$ | transfer function of the differential encoder |
| $f$ | variable representing frequency |
| $f_c$ | carrier frequency |
| $f_1$ | asymmetric carrier frequency |
| $f_0$ | frequency representing the difference between $f_c$ and $f_1$ |
| $f_{\mathrm{B}}$ | one-sided band-width of BPF at the input of differential demodulator |
| $f_{\mathrm{L}}$ | one-sided band-width of LPFs at the output of demodulators |

| | |
|---|---|
| $f_w$ | one-sided band-width of $w(t)$ |
| $f_D$ | maximum Doppler shift |
| $\mathbf{F}(D)$ | transfer function of the feedback DCPED |
| $\mathbf{F}_l(D)$ | equivalent $l \times 2l$ version of $\mathbf{F}(D)$ |
| $\dot{\mathbf{F}}(D)$ | feedback-free version of $\mathbf{F}(D)$ |
| $\dot{\mathbf{F}}_l(D)$ | equivalent $l \times 2l$ version of $\dot{\mathbf{F}}(D)$ |
| $g_{ik}(D)$ | elements of $\mathbf{G}(D)$ |
| $g_{-1,i}$ | the $i$-th residue of $G_{\Upsilon_e}(\xi)/\xi$ at $\xi = \xi_i$ |
| $G_{\Upsilon_e}(\xi)$ | characteristic function of $\Upsilon_e$ |
| $\mathbf{G}(D)$ | transfer function of the channel encoder |
| $\dot{\mathbf{G}}(D)$ | scrambled version of $\mathbf{G}(D)$ |
| $h$ | modulation index |
| $h_{\mathrm{c,I},\hat{U}_n}(t)$ | impulse response of in-phase matched filter in coherent receiver |
| $h_{\mathrm{c,Q},\hat{U}_n}(t)$ | impulse response of quadrature matched filter in coherent receiver |
| $h_{\mathrm{d,I},\hat{\beta}_n-\hat{\beta}_{n-1}}(t)$ | impulse response of in-phase matched filter in differential receiver |
| $h_{\mathrm{d,Q},\hat{\beta}_n-\hat{\beta}_{n-1}}(t)$ | impulse response of quadrature matched filter in differential receiver |
| $i$ | general purpose index |
| $\mathbf{I}_{N_s}$ | the $N_s \times N_s$ identity matrix |
| $j$ | the square root of $-1$ |
| $J_c(\hat{\boldsymbol{U}})$ | infinite correlation performed in coherent receiver to determine $\boldsymbol{U}$ |
| $J_{c,n}(\hat{\boldsymbol{U}})$ | semi-infinite correlation performed in coherent receiver to determine $\boldsymbol{U}$ |
| $J_d(\hat{\boldsymbol{U}})$ | infinite correlation performed in differential receiver to determine $\boldsymbol{U}$ |
| $J_{d,n}(\hat{\boldsymbol{U}})$ | semi-infinite correlation performed in differential receiver to determine $\boldsymbol{U}$ |
| $k$ | general purpose index or integer |
| $k_M$ | positive integer relating $M$ and $P$ |
| $k_{1,n}$ | non-negative integer less than $k_M$ |
| $k_{2,n}$ | non-negative integer less than $k_M$ |
| $k_n$ | integer used in NMSED of DCPFSK calculations |
| $k'$ | index used to enumerate reference signals in VA |
| $k_V$ | index used to identify system trellis intervals |
| $K$ | numerator of $h$ |
| $l$ | number of symbols output by the channel encoder |
| $L$ | number of transmitted symbols (in system trellis intervals) |
| $M$ | size of symbol alphabet |
| $n$ | index representing symbol period |
| $n_s$ | number of different values that the $2l$-th element in $\boldsymbol{b}_{k_V}$ takes before the channel encoder merges to the zero state |
| $N_0$ | one-sided psd of white noise |
| $N_s$ | total number of elements in vectors used in fading performance calculations |
| $p(\Upsilon_e)$ | probability density function of $\Upsilon_e$ |
| $P$ | denominator of $h$ |
| $P_e$ | probability of bit error |

| | |
|---|---|
| $q(t)$ | phase response of CPFSK |
| $Q(x)$ | the $Q$-function |
| $r$ | information bit-rate of the channel encoder |
| $r_s$ | sampling rate |
| $r(t, \boldsymbol{U})$ | received signal |
| $\dot{r}(t-T, \boldsymbol{U})$ | $T$-delayed and $\pi/2$-phase shifted version of $r(t, \boldsymbol{U})$ |
| $R_w(\tau)$ | autocorrelation of $w(t)$ |
| $\mathbf{R}_{\tilde{r}\tilde{r}}$ | the autocorrelation matrix of $\tilde{r}$ |
| $\mathbf{R}(D)$ | transfer function of the differential decoder |
| $s(t, \boldsymbol{U})$ | transmitted CPFSK signal |
| $s(t, \boldsymbol{\beta})$ | transmitted DCPFSK signal |
| $s_{\mathrm{I}}(t, \boldsymbol{U})$ | symmetric in-phase component of $s(t, \boldsymbol{U})$ |
| $s_{\mathrm{Q}}(t, \boldsymbol{U})$ | symmetric quadrature component of $s(t, \boldsymbol{U})$ |
| $s(\tau, \boldsymbol{X}_n)$ | output of the memoryless modulator in the $n$-th symbol period |
| $s'_{\mathrm{I}}(\tau, \boldsymbol{X}_n)$ | asymmetric in-phase component of $s(\tau, \boldsymbol{X}_n)$ |
| $s'_{\mathrm{Q}}(\tau, \boldsymbol{X}_n)$ | asymmetric quadrature component of $s(\tau, \boldsymbol{X}_n)$ |
| $S_w(f)$ | power spectral density of $w(t)$ |
| $S_G$ | number of states in the channel encoder |
| $S_V$ | number of states in the overall encoder |
| $\mathcal{S}_{k_V}$ | state of the transmitter in the $k_V$-th trellis interval |
| $\hat{\mathcal{S}}_{k_V}$ | state of the receiver in the $k_V$-th trellis interval |
| $t$ | variable representing time |
| $T$ | symbol period |
| $T_b$ | bit period |
| $T_s$ | sample period |
| $\mathbf{T}(D)$ | transfer function of the scrambler |
| $\mathbf{T}_l(D)$ | equivalent $l \times l$ version of $\mathbf{T}(D)$ |
| $u(t)$ | the unit step function |
| $U_n$ | uncoded $M$-ary symbol at the input of the CPE in the $n$-th symbol period |
| $\boldsymbol{U}$ | vector of successive $U_n$ |
| $v_n$ | accumulated symbol phase of the transmitted DCPFSK signal in the $n$-th symbol period |
| $V_n$ | accumulated symbol phase of the transmitted CPFSK signal in the $n$-th symbol period |
| $\boldsymbol{V}$ | vector of successive $V_n$ |
| $w(t)$ | Gaussian noise process |
| $W_{k_V}$ | number of information |
| $\boldsymbol{X}_n$ | input to the memoryless modulator in the $n$-th symbol period |
| $\breve{\boldsymbol{X}}_n$ | differential input to the memoryless modulator in the $n$-th symbol period |
| $y_{\mathrm{c}}(t, \boldsymbol{U})$ | coherently-demodulated CPFSK signal |
| $y_{\mathrm{d}}(t, \boldsymbol{U})$ | differentially-demodulated CPFSK signal |
| $y_{\mathrm{d}}(t, \boldsymbol{\beta})$ | differentially-demodulated DCPFSK signal |
| $z(t)$ | narrow-band Rayleigh flat fading process |
| $\mathbb{Z}_P$ | ring of integers modulo-$P$ |

| | |
|---|---|
| $\alpha_n$ | standard CPM symbols |
| $\beta_n$ | output of the differential encoder in the $n$-th symbol period |
| $\boldsymbol{\beta}$ | vector of successive $\beta_n$ |
| $\delta_{i,k}$ | the Kronecker delta function |
| $\delta(x)$ | the delta function |
| $\varepsilon_{k_V,\mathcal{S},m}$ | an event used in calculating the performance of coded (D)CPFSK systems |
| $\hat{\varepsilon}_{k_V,\mathcal{S},m}$ | an event used in calculating the performance of coded (D)CPFSK systems |
| $\delta^2_{k_V}(\boldsymbol{c},\hat{\boldsymbol{c}})$ | normalised SED between two encoded sequences, $\boldsymbol{c}_{k_V}$ and $\hat{\boldsymbol{c}}_{k_V}$, in the $k_V$-th trellis interval |
| $\zeta$ | length of an error event (in system trellis intervals) |
| $\eta(\mathcal{S},\iota,\zeta,d)$ | the number of error events starting from state $\mathcal{S}$, assuming the trellis is infinite, having $\iota$ information bit errors, a length of $\zeta$ trellis intervals and a NSED equal to $d^2$ |
| $\Theta_n$ | term used in NMSED of DCPFSK calculations |
| $\iota$ | number of bit errors in an error event |
| $\lambda_i$ | the $i$-th eigenvalue of $\mathbf{R}^*_{\tilde{r}\tilde{r}}\mathbf{A}$ |
| $\lambda_{c,n}(\hat{\boldsymbol{U}})$ | metric used in coherent receiver to perform sequence estimation of $\boldsymbol{U}$ |
| $\lambda_{c,n}(\hat{U}_n,\hat{V}_n)$ | metric used in coherent receiver to perform sequence estimation of $\boldsymbol{U}$ |
| $\lambda_{d,n}(\hat{\boldsymbol{U}})$ | metric used in differential receiver to perform sequence estimation of $\boldsymbol{U}$ |
| $\lambda_{d,n}(\hat{\beta}_n,\hat{\beta}_{n-1})$ | metric used in differential receiver to perform sequence estimation of $\boldsymbol{U}$ |
| $\nu$ | the number of delay cells in the encoder |
| $\xi$ | dummy variable used to transform $G_{\Upsilon_e}(\xi)$ |
| $\xi_i$ | poles of $G_{\Upsilon_e}(\xi)$ |
| $\Xi(d)$ | the error coefficient for error events with NSED equal to $d^2$ |
| $\tau$ | alternate time variable |
| $\Upsilon_e$ | error metric used in performance calculations |
| $\Upsilon(y_{\mathrm{d}},\boldsymbol{\beta})$ | metric used in performance calculations |
| $\varphi_0$ | initial phase offset |
| $\phi_n$ | frequency term in $y_{\mathrm{d}}(t,\boldsymbol{\beta})$ in the $n$-th symbol period |
| $\Phi_n$ | difference between $\beta_n$ and $\hat{\beta}_n$ |
| $\psi(t,\boldsymbol{U})$ | tilted phase |
| $\bar{\psi}(t,\boldsymbol{U})$ | physical tilted phase |
| $\bar{\psi}(\tau,\boldsymbol{X}_n)$ | physical tilted phase in the $n$-th symbol period |
| $\psi_d(t,\boldsymbol{\beta})$ | differentially-demodulated tilted phase |
| $\bar{\psi}_d(t,\boldsymbol{\beta})$ | physical differentially-demodulated tilted phase |
| $\Psi_i$ | difference between $\Theta_i$ and $\hat{\Theta}_i$ |
| $\tilde{\Psi}_i$ | difference between $\Theta_i$ and $\hat{\Theta}_i$ plus a multiple of $P$ |
| $\Omega_{n-1}$ | phase term in $y_{\mathrm{d}}(t,\boldsymbol{\beta})$ in the $n$-th symbol period |
| | |
| $x^*(t)$ | the complex conjugate of $x(t)$ |
| $\tilde{x}(t)$ | the complex envelope of $x(t)$ |
| $x_{\mathrm{I}}(t)$ | the in-phase component of $\tilde{x}(t)$ |

| | |
|---|---|
| $x_{\mathrm{Q}}(t)$ | the quadrature component of $\tilde{x}(t)$ |
| $\hat{x}(t)$ | an estimate (or hypothesised value) of $x(t)$ |
| $x(D)$ | delay polynomial of $x$ |
| $\boldsymbol{x}^{\dagger}$ | the non-conjugate transpose of $\boldsymbol{x}$ |
| $\boldsymbol{x}^{H}$ | the conjugate transpose of $\boldsymbol{x}$ |

# Bibliography


[AAS86] J.B. Anderson, T. Aulin, and C.-E. Sundberg, *Digital Phase Modulation*, Plenum Press, 1986.

[AS81] T. Aulin and C.-E. Sundberg, "On differential detection of partial response continuous phase modulated signals," in *Proc. IEEE International Conference on Communications*, Seattle, WA, June 1981, vol. 3, pp. 56.1.1–56.1.6.

[BDMS91] E. Biglieri, D. Divsalar, P. McLane, and M. K. Simon, *Introduction to trellis-coded modulation with applications*, MacMillan, 1991.

[For70] G.D. Forney Jr, "Convolutional codes I: Algebraic structure," *IEEE Transactions on Communications*, vol. IT-16, no. 6, pp. 720–738, Nov. 1970.

[For72] G.D. Forney Jr, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. IT-18, no. 3, pp. 363–378, May 1972.

[For73] G.D. Forney Jr, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[Har96] B. Hart, *MLSE Diversity Receiver Structures*, Ph.D. thesis, University of Canterbury, 1996.

[Hay83] S. Haykin, *Communication Systems*, John Wiley & Sons, 2nd edition, 1983.

[Hay88] S. Haykin, *Digital Communications*, John Wiley & Sons, 1988.

[Jak74] W.C. Jakes Jr, Ed., *Microwave Mobile Communications*, Wiley, 1974.

[LJKK97] J.Y. Lee, P.Y. Jou, K.H. Kim, and C.E. Kang, "Power-bandwidth performance of smoothed phase modulation codes," *Electronics Letters*, vol. 33, no. 24, pp. 2020–2021, Nov. 1997.

[Mas90] T. Masamura, "Intersymbol interference reduction for differential MSK by nonredundant error correction," *IEEE Transactions on Vehicular Technology*, vol. 39, no. 1, pp. 27–36, Feb. 1990.

[MM89] J.L. Massey and T. Mittleholzer, "Convolutional code over rings," in *Proceeding of Fourth Joint Swedish-USSR Int. Workshop on Information Theory*, Aug. 1989, pp. 14–18.

[MM90] J.L. Massey and T. Mittleholzer, "Systematicity and rotational invariance of convolutional codes over rings," in *Proc. 2nd Int. Workshop on Algebraic and Combinatorial Coding Theory*, Sept. 1990, pp. 154–159.

[MMP88] F. Morales-Moreno and S. Pasupathy, "Structure, optimization and realization of FFSK trellis codes," *IEEE Transactions on Information Theory*, vol. 34, pp. 730–751, July 1988.

[Pro95] J. G. Proakis, *Digital Communications*, McGraw-Hill, third edition, 1995.

[Rim88] B. Rimoldi, "A decomposition approach to CPM," *IEEE Transactions on Information Theory*, vol. 34, no. 2, pp. 260–270, Mar. 1988.

[Rim89] B. Rimoldi, "Design of coded CPFSK modulation systems for bandwidth and energy efficiency," *IEEE Transactions on Communications*, vol. 37, no. 9, pp. 897–905, Sept. 1989.

[Rim91] B. Rimoldi, "Exact formula for the minimum squared Euclidean distance of CPFSK," *IEEE Transactions on Communications*, vol. 39, no. 9, pp. 1280–1282, Sept. 1991.

[RL95] B. Rimoldi and Q. Li, "Coded continuous phase modulation using ring convolutional codes," *IEEE Transactions on Communications*, vol. 43, no. 11, pp. 2714–2720, Nov. 1995.

[SBS66] M. Schwartz, W.R. Bennett, and S. Stein, *Communication Systems and Techniques*, McGraw-Hill, 1966.

[Spi81] Murray R. Spiegel, *Schaum's Outline Series - Theory and Problems of Complex Variables SI metric edition*, McGraw-Hill, 1981.

[SZ98] V. Sunkad and R. E. Ziemer, "Performance of CPFSK in delay and Doppler spread multipath," in *IEEE Vehicular Technology Conference Record*, Ottawa, Canada, May 1998, vol. 3, pp. 2272–2276.

[van96] R. van Nobelen, *Coding for the Rayleigh Fading Channel*, Ph.D. thesis, University of Canterbury, 1996.

[Vit71] A.J. Viterbi, "Convolutional codes and their performance in communication system," *IEEE Transactions on Communication Technology*, vol. COM-19, pp. 751–772, Oct. 1971.

[Yan94] R. H. Yang, *On trellis coded continuous phase frequency shift keying*, Ph.D. thesis, McMaster University, 1994.

[YL90] R.J. Young and J.H. Lodge, "Linear-prediction-aided differential detection of CPM signals transmitted over Rayleigh flat fading channels," in *IEEE Vehicular Technology Conference Record*, Orlando, Florida, May 1990, pp. 437–442.

[YT92] L. Yuan and D. P. Taylor, "Differential detection of differentially encoded multi-h CPM using Viterbi decoding," Tech. Rep. 250, Communication Research Laboratory, McMaster University, Canada, May 1992.

[YT94] R. H. Yang and D. P. Taylor, "Trellis coded continuous phase frequency shift keying with ring convolution codes," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1057–1067, July 1994.