ITERATIVE DECODING OF CODED GMSK WITH DISCRIMINATOR DETECTION

by

Eddy Wong

A thesis submitted in conformity with the requirements

for the degree of Masters of Applied Science

Graduate Department of Electrical and Computer Engineering

University of Toronto

0-612-63037-4

ITERATIVE DECODING OF CODED GMSK WITH DISCRIMINATOR DETECTION

by

Eddy Wong

Masters of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2001

Abstract

In this thesis, a communications system using the modulation scheme Gaussian minimum shift keying (GMSK) together with a serially concatenated coding system is studied under additive white Gaussian noise conditions and fading. The receiver used is the non-coherent frequency discriminator combined with an iterative decoding scheme. Using both *maximum a posteriori* (MAP) based algorithms and the soft-output Viterbi algorithm (SOVA) for decoding, the performance trade-off associated with applying a sub-optimal algorithm to the system is investigated. The impact of using a less accurate likelihood statistic to simplify calculations at the output of the discriminator is also examined. In a fading channel, the likelihood statistic is improved by adding side-information obtained from the signal envelope. Other parameters considered include the outer code and joint demodulation / decoding. Through simulation, it is shown that good performance can be obtained by this system even with relatively simple codes.

Acknowledgements

I would like to thank my thesis supervisor, Prof. S. Pasupathy, for his guidance during my work at the U of T. His patience and support throughout these years is greatly appreciated.

Also, I would like to express my gratitude to the University of Toronto for the financial support and providing a great environment in which to work and learn. It has been a wonderful experience.

Lastly, but most importantly, I wish to thank my parents and family. None of my accomplishments would have been possible without their love and support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

A coding scheme, known as "Turbo Codes" [1] was presented by Berrou, Glavieux and Thitimajshima in 1993. This coding scheme can achieve a coding performance that is close to the theoretical Shannon capacity limit for data transmission over a noisy channel. Using parallel concatenated codes and a decoding method involving feedback, this performance was obtained without a significant increase in decoding and computational complexity. In the years following, research in this area has extended to using serial concatenated codes [2] and applying different algorithms in the decoding process [3].

Another area of interest is in bandwidth efficient digital modulation techniques. One such technique is continuous phase modulation (CPM), which has good spectral properties. Due to its bandwidth efficient design, CPM is used in a large number of applications. In particular, a type of CPM signalling known as Gaussian minimum shift keying (GMSK) [4] is used in several communication standards such as the Global System for Mobile telecommunications (GSM) [5], Digital European Cordless Telecommunication (DECT) [6], and the wireless local area network standard HIPERLAN [7].

To demodulate a GMSK signal, both coherent and non-coherent methods can be used. Most standards do not specify a method that must be followed; therefore, it is desirable to choose one that has a good balance between complexity and performance. A receiver structure that is often used to demodulate CPM signals is the frequency discriminator. A frequency discriminator has a simple design, which is a desirable trait for mobile devices.

This demodulator, however, is non-coherent, so the performance is worse than that of the more complex coherent receivers.

Several approaches can be taken to improve the performance of a system employing the use of a frequency discriminator. The application of error control codes could be used to correct for some of the errors introduced by non-coherent demodulation. Maximum likelihood sequence estimation (MLSE) provides a method to remove the intersymbol interference and improve decoding performance. In a fading environment, some of the information is carried in the amplitude of the signal as well as in the phase and this could also be exploited to improve the demodulation performance [8].

The use of a spectrally efficient modulation, a simple receiver, and a powerful coding/decoding scheme can form a system that performs well while keeping the system complexity to a minimum. Szeto [9] studied the iterative decoding of coded GMSK under coherent detection and found that it can provide good results. Now we will investigate a similar coded system with discriminator detection to find out how iterative decoding can be designed to help overcome the performance degradation of non-coherent detection.

Among the issues discussed are how the memory of the GMSK signal is affected by the non-linear processing of the discriminator and the impact this has on iterative decoding. How the system performs when using a joint demodulation / decoding approach will also be examined. Another issue involves the likelihood statistics from the channel. Due to the nature of the discriminator, accurate modelling of the phase noise may not be feasible. In this case, we investigated how assuming a less accurate statistic affects the system and if using other side-information will help improve the performance.

In terms of complexity and performance trade-offs, there are some areas where the design can be changed to satisfy the application. Using various algorithms with different levels of complexity and optimality in the decoding process, the effect the algorithms have on the system performance is studied. Finally, we examine the significance of the outer code in the concatenated scheme and the relationship between the memory length of the outer code and its corresponding effect on the coding performance.

1.1 Organization of Thesis

Chapter 2 gives the background information to the various components of the discriminator-based, iterative decoding system. This includes the modulation technique, the demodulation components, the coding scheme and the iterative decoding method. The different algorithms used are also described in that chapter.

In chapter 3, the simulation results of the system operating under the additive white Gaussian noise channel are presented. The noise statistics at the output of the discriminator is described and used to calculate the channel reliabilities. Other parameters investigated include outer codes, inner codes and decoding algorithms.

The performance of the system under multi-path fading is studied in chapter 4. The use of amplitude information for the decoding metric is discussed. Some of the parameters studied in chapter 3 will also be investigated to determine their effects under fading conditions.

Chapter 5 concludes and summarizes the results found in this research. Suggestions for further work in this area are also given.

Chapter 2

System Components

The communication system presented in this thesis will be divided into three major components. The three components are the transmitter, the channel and the receiver. Information will pass through the system as shown in Figure 2.1. In this chapter, the details of these components will be discussed.

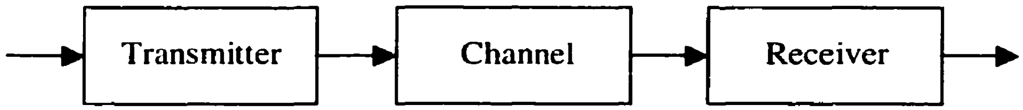

Figure 2.1: Communication system model.

2.1 Transmitter Components

The transmitter component consists of the coding and modulation aspects of the communication system. For coding, we will be using serially concatenated convolutional codes. This will allow us to decode the information using an iterative decoding approach at the receiver. The modulation scheme used in our system is GMSK.

2.1.1 Concatenated Codes

Since the introduction of turbo codes, a great deal of research has been made towards the study of concatenated codes with iterative decoding. There are two possible methods used to concatenate convolutional codes. Turbo codes, as presented in 1993 by Berrou et al., involve the parallel concatenation of two recursive systematic convolutional (RSC) codes decoded using a feedback-decoding rule. In [2], Benedetto et al. uses a serial concatenation scheme to achieve coding performances that are similar to those of turbo codes.

In both parallel and serial concatenation schemes, two constituent codes are used and are linked by an interleaver. The performance of the coding subsystem is interdependently affected by the design of the constituent codes and the interleaver [10]. Figure 2.2 and Figure 2.3 shows the basic structures of parallel and serial concatenation used in iterative decoding respectively.

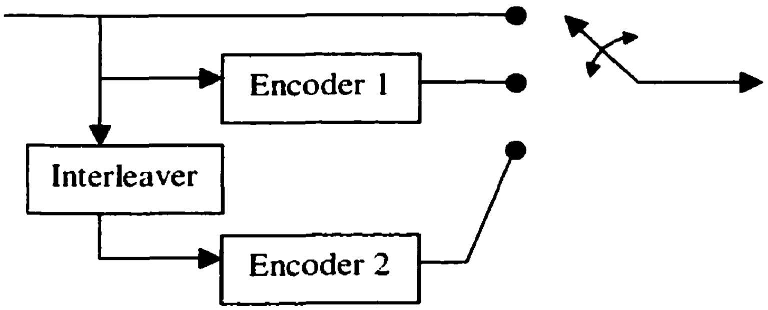

Figure 2.2: Parallel concatenation.

In the case of parallel concatenation, the data bits are fed into one of the encoders and an interleaved version of the data is fed into the other encoder. Note that the two encoders do not have to be identical. The input bits are transmitted systematically along with the redundant error-control bits. The overall coding rate, R, of this system is

$$\frac{1}{R}=\frac{1}{R_1}+\frac{1}{R_2}+1,$$

where R_1 and R_2 are the coding rates of the first and second encoders respectively. By puncturing the redundant error-control bits, a higher coding rate may be obtained.

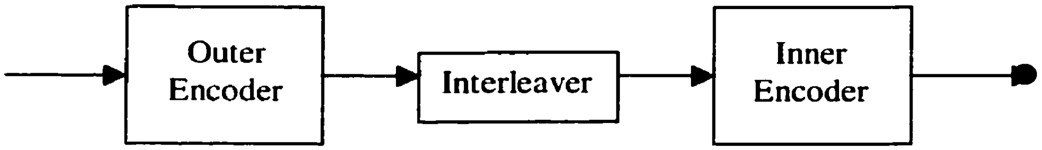

Figure 2.3: Serial concatenation.

For serial concatenation, the output of one encoder is used as the input to the other encoder in a sequential manner. The data bits are fed into the first encoder, called the outer encoder. Through the interleaver, an interleaved version of the output from the outer encoder is fed into the second encoder, called the inner encoder. If the coding rates of the outer code and the inner code are $R_o = \frac{k}{p}$ and $R_i = \frac{p}{n}$ respectively, then the overall coding rate of the serial concatenated system is $R = \frac{k}{n}$.

2.1.2 Gaussian Minimum Shift Keying

A popular modulation scheme used in radio applications is CPM. The constant envelope property and its spectrally efficient design make it a desirable choice for digital data transmission. To achieve its spectral efficiency, the CPM signal is constrained to have a continuous phase with no abrupt changes to eliminate the high frequency components. There are several parameters in a CPM system that can be manipulated to obtain the desired properties in the spectra.

In a constant envelope modulation scheme such as CPM, the information is carried in the frequency or phase of the signal. A transmitted signal of this type is given [11] by

$$s(t,\alpha)=\sqrt{\frac{2E}{T}}\cos(2\pi f_o t+\phi(t,\alpha)+\phi_o)\,, \tag{2.1}$$

where E is the symbol energy, T is symbol duration, f_o is the carrier frequency and ϕ_o is the phase offset. The information is carried in the pulse amplitude modulated (PAM) signal

$$\phi(t,\alpha)=2\pi h\sum_{i=0}^{\infty}\alpha_i f(t-iT)\,, \tag{2.2}$$

which is also known as the information-carrying phase. The M-ary information stream is given by $\alpha=(\,\alpha_0, \alpha_1, \ldots\,)$, with $\alpha_i \in \{\,\pm 1, \pm 3, \ldots, \pm \text{M-1}\,\}$ if M is even, and $\alpha_i \in \{\,0, \pm 2, \pm 4, \ldots, \pm \text{M-1}\,\}$ if M is odd. We will restrict our investigations in this thesis to the binary case, with M=2 and $\alpha_i \in \{\,\pm 1\,\}$. h is the modulation index and $f(t)$ is the phase response. The phase response pulse is the integral of the frequency pulse $g(t)$ such that it satisfies

$$f(t)=\begin{cases} 0 & \text{if } t\,0\!\leq\quad, \\ \int_0^t g(\tau)\,d\tau & \text{if } 0<t\ \leq LT, \\ \frac{1}{2} & \text{if } t\!>\ LT \end{cases}$$

The pulse $g(t)$ is a smooth function over the interval $0 \leq t \leq LT$ and is zero elsewhere. $g(t)$ is normalized so that the pulse integrates to $\frac{1}{2}$. L is a positive integer known as the memory of the modulation scheme, representing the number of bit periods that the frequency pulse extends over. For L equal to 1, the scheme is known as full-response CPM; for L greater than 1, the scheme is known as partial-response CPM.

A special case of binary CPM with a modulation index $h = 0.5$ is known as minimum shift keying (MSK). MSK may be viewed as a form of offset quadrature phase shift keying (OQPSK) or as continuous phase frequency shift keying (CPFSK) [12]. There are many implementations of MSK and [13] shows how one could convert from one scheme to another. The frequency and phase response pulses of MSK are shown in Figure 2.4. From the figure, it can be seen that for MSK, $L = 1$ and therefore it is a full-response signal.

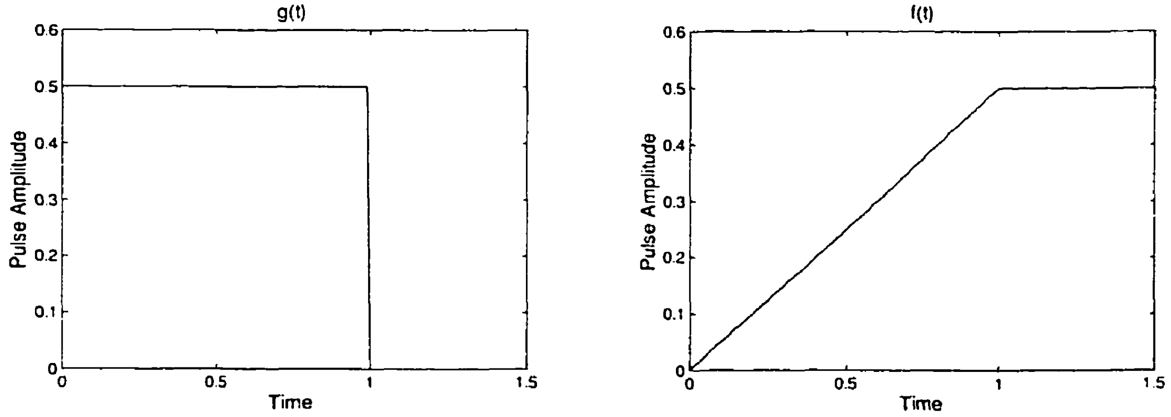

Figure 2.4: MSK frequency ($g(t)$) and phase ($f(t)$) pulses.

To make the spectrum of the signal more compact, the frequency pulse $g(t)$ in Figure 2.4 can be manipulated to give the desired characteristics. One of the more popular shapes is the Gaussian curve. This form of CPM is called Guassian minimum shift keying and it is the one used in the GSM standard. In GMSK, the rectangular non-return to zero (NRZ)[1]

[1] In a NRZ waveform, a positive level (+V) represents a binary 1and a negative level (-V) represents a binary 0. The waveform changes level when the data changes from a 1 to a 0 or from 0 to a 1.

input pulses are passed through a Gaussian low-pass filter (LPF) called the premodulation filter before being modulated by the FM modulator. This is shown in Figure 2.5. The response of the Gaussian LPF to a single NRZ pulse is equivalent to the frequency pulse $g(t)$ for GMSK, given by

$$g(t)=\frac{1}{2T}\left[\mathrm{Q}\left(\frac{2\pi B_o}{\sqrt{\ln 2}}\left(t-\frac{T}{2}\right)\right)-\mathrm{Q}\left(\frac{2\pi B_o}{\sqrt{\ln 2}}\left(t+\frac{T}{2}\right)\right)\right], \tag{2.3}$$

where Q(x) is the complementary error function defined as:

$$\mathrm{Q}(x)=\frac{1}{\sqrt{2\pi}}\int_x^\infty \exp(-u^2/2)\,du\,.$$

The normalized bandwidth of the filter, B_oT, is a design parameter that affects the performance of the system. A lower value of B_oT will give a narrower power spectrum and as $B_oT \rightarrow \infty$, the GMSK waveform becomes the MSK waveform. Common values used for B_oT are 0.25, 0.3, and 0.5.

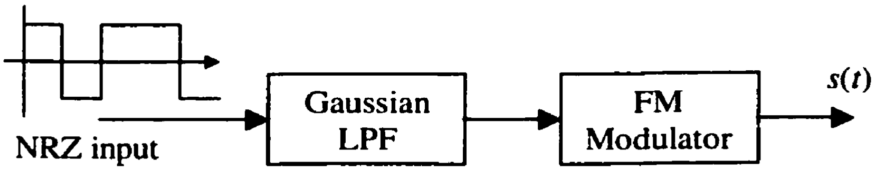

Figure 2.5: GMSK modulator.

In this study, we'll be using $h=0.5$ and $B_oT=0.3$ as specified in the GSM standard. Figure 2.6 shows the frequency and phase pulses created using this parameter. GMSK is a partial-response CPM signal, with L = 3 in our case. The actual Gaussian pulse extends to infinity on both sides, but it's truncated to after one symbol to either side since no significant inter-symbol interference (ISI) occurs beyond that.

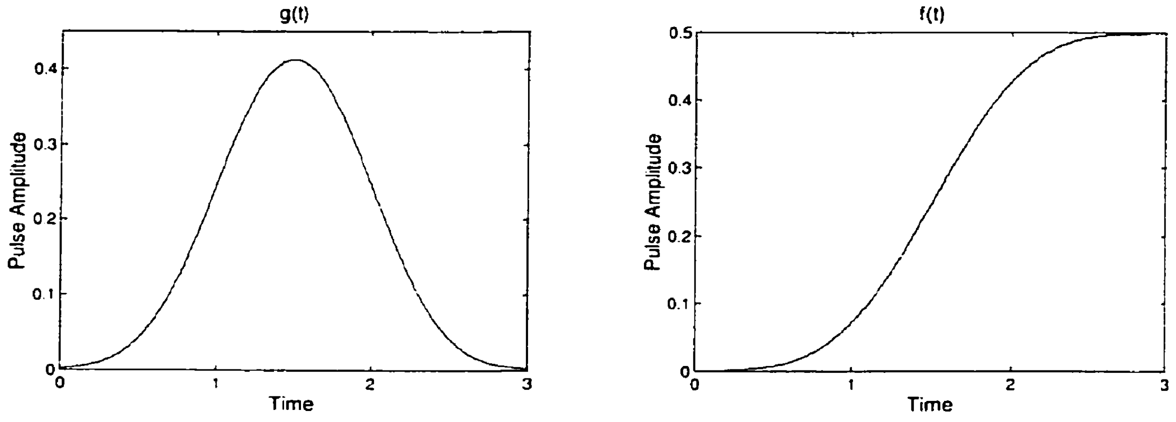

Figure 2.6: GMSK frequency ($g(t)$) and phase ($f(t)$) pulses, $B_bT = 0.3$.

2.1.3 Coded Modulation

To further improve bandwidth efficiencies and conserve energy, we could combine coding and modulation into a single scheme. This is known as coded modulation. Anderson and Sundberg [14] shows examples of coded modulation as applied to CPM.

An interesting concept used in studying the memory aspect of CPM is by separating the memory imposed by the continuous phase constraint from the signal modulation. Rimoldi [15] proposed a method in which a CPM system is decomposed into a linear continuous phase encoder and a memoryless modulator. It was developed to provide different realizations of CPM and decoding algorithms. Using this approach, the continuous phase encoder can be viewed as a convolutional encoder. This allows for simpler implementations when combining CPM with other convolutional encoders and using CPM in conjunction with concatenated codes.

2.2 Channel

We will be evaluating the performance of our system under two different channel models. The first will the standard additive white Gaussian noise (AWGN) channel. Under this model, each noise sample is uncorrelated and independent of others. An important characteristic of this noise process is that the power spectral density is a constant, equal to $N_o/2$, for all frequencies. The discrete-time bandlimited noise samples from this channel will be Gaussian distributed with a zero-mean and a variance equal to $N_o/2$.

The other channel used is the Rayleigh fading channel model. In a mobile environment, when there may be no direct line-of-sight from the transmitter to the receiver, transmission is done via the reflection and scattering of the signal from the surfaces of the obstacles. As such, more than one "copy" (usually with time variations) of the transmitted signal travelling from different paths will be detected at the receiver. Depending on the time delay between the signals, there could be constructive or destructive interference at the receiver. When destructive interference occurs, the received signal will be weak and this is known as "fading". With the Rayleigh fading model, the amount of fading is described by Rayleigh density function. For a detailed discussion on fading models and multi-path phenomena, refer to [16].

2.3 Receiver Components

The receiver section contains the demodulation and decoding components of the system. To demodulate the CPM signal, we use a frequency discriminator circuit. An iterative decoding approach will be used for the decoding of the concatenated code. This will help recover the performance losses associated with non-coherent detection without significantly increasing the receiver complexity. Several different decoding algorithms can be used for the iterative decoding process, and these will also be discussed in this section.

2.3.1 Frequency Discriminator

A frequency discriminator is a common and well-known circuit used to detect both analog and digital FM signals. This is a non-coherent detection method, and it's expected to perform worse than coherent detection. The advantage of this detection method is the simple design and ease of implementation in mobile devices. Different methods of decoding the CPM signals under discriminator detection have been studied in [17] and [18] to improve performance.

The output of the discriminator is proportional to the frequency deviation of the input signal from the carrier frequency, f_c. In an ideal discriminator, the response of the circuit should be linear, with the zero crossing at f_c as shown in Figure 2.7. For the signal given by Equation (2.1), the output of the ideal discriminator (in the absence of noise) is given by

$$y(t) = K_D \frac{d\phi}{dt}, \tag{2.4}$$

where K_D is the amplitude of the signal. It should be noted that a frequency discriminator does respond to variations in the amplitude, via the variable K_D. For CPM, this is constant. However, in the presence of noise, this will not be the case for the received signal. To remove any amplitude variations introduced to the signal envelope due to fading and noise, a limiter is placed before the discriminator. This combination is commonly referred to as a limiter-discriminator.

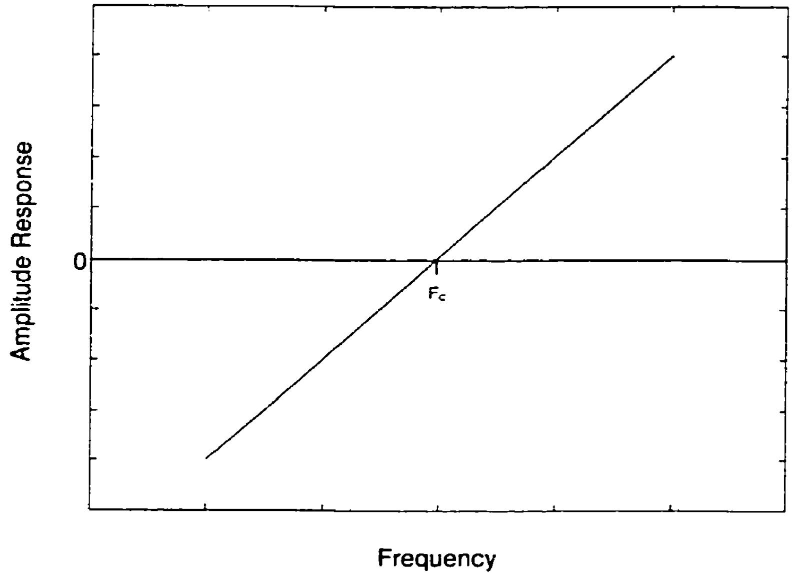

Figure 2.7: Ideal response of a frequency discriminator.

In practice, the discriminator could be implemented using bandpass filters. Two bandpass filters with center frequencies staggered around the carrier frequency are used. The received signal is passed through both filters, and then the difference of their outputs is taken. This yields a response that approximates the ideal response. Figure 2.8 gives an illustrated example of how this is implemented. From the figure, we can see that there is a linear region around f_c that approximates the ideal response. The transmitter and receiver must be designed so that that the signal does not fall beyond this linear operating range. Operating beyond the linear region will result in erroneous detection and performance degradation.

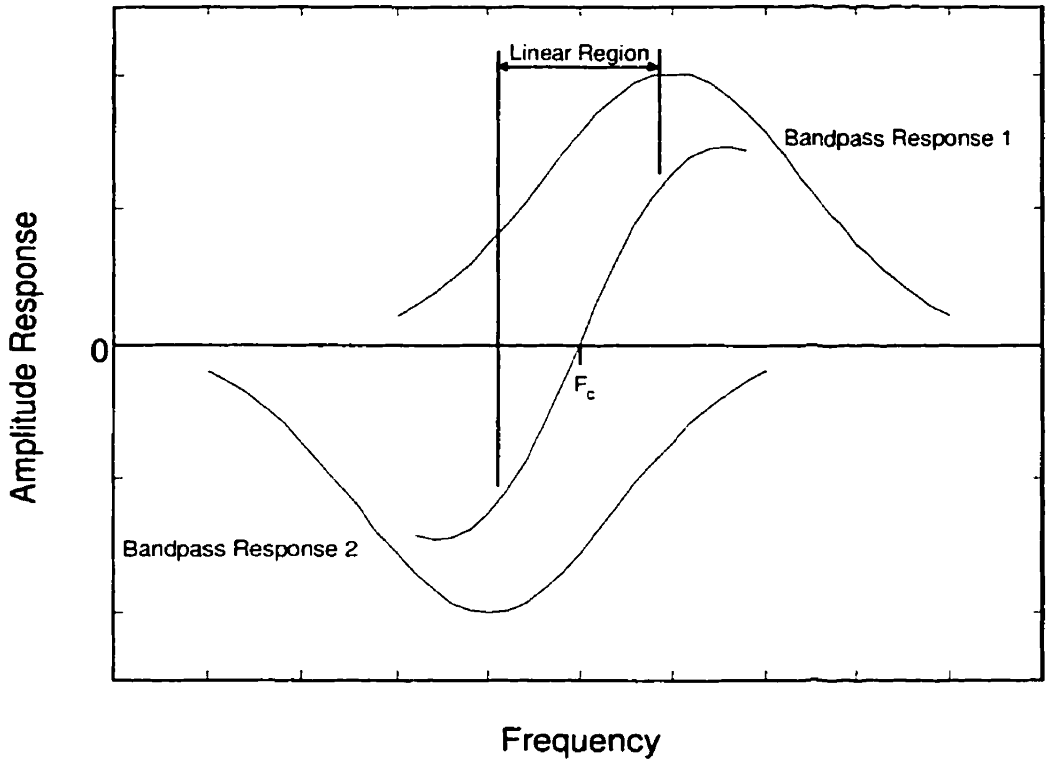

Figure 2.8: Frequency discriminator implemented using bandpass filters.

There are two common methods used to read the output from the discriminator. One is to pass it through an integrate-sample-and-dump (ISD) filter. This filter integrates the output of the frequency discriminator over one bit period and produces a phase difference. The other is a simple sample-and-hold (SH) method that gives the instantaneous frequency of the received signal at the sample time. The ISD filter performs better than SH since it has an averaging effect on the noise over the whole bit period.

2.3.2 Iterative Decoding of Concatenated Codes

One of the ideas that gives turbo codes its good performance is its feedback decoding technique, also known as iterative decoding. The premise of iterative decoding of concatenated codes is that by passing decoding information obtained from the second decoder back to the first decoder and performing the decoding operation again, the decoder could improve the estimation of the transmitted bits. To accomplish this transfer of decoding information, soft decoding rather than hard decoding must be used.

In soft decoding, the algorithm will make a weighted (soft) decision instead of a hard decision. That is, the output will contain the probabilities of the data bit being a 0 or a 1. This is usually in the form of a log-likelihood ratio of the *a posteriori* probability of the data bit, u_t,

$$\lambda(u_t) = \log \frac{\Pr(u_t = 1 \mid Y_1^\tau)}{\Pr(u_t = 0 \mid Y_1^\tau)},$$

where Y_1^τ is the observed output sequence from the channel.

Regardless of whether the code is concatenated in parallel or in serial, it is decoded in a serial manner. The decoding structure for both are similar, but we will focus our attention on serial concatenated codes, which is used in this thesis. A main component of the decoding structure is the soft-input soft-output (SISO) module as described in [19]. These modules decode the constituent codes by updating the *a priori* probabilities using known information from the trellis structure of the codes. The module takes in two inputs and provides two outputs, shown in Figure 2.9. The inputs are **P(c**; *I*) and **P(u**; *I*), which are the probabilities of the codewords, c, from observation and the *a priori* probabilities of the input bits, u, respectively. After processing the information, two *a* *posteriori* probabilities, $\mathbf{P}(\mathbf{c}; O)$, the updated codeword probability and $\mathbf{P}(\mathbf{u}; O)$, the updated probability of the input bits will be output. The letters I and O indicates whether the probabilities refer to the inputs or the outputs of the module. Different algorithms can be used for the SISO module, and the algorithm will affect the complexity and performance of the module depending on which algorithm is chosen. We will investigate the performance of three different algorithms and they will be discussed in their own sections.

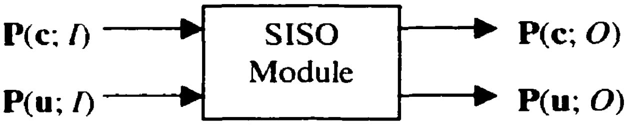

Figure 2.9: SISO module.

Figure 2.10 shows how the SISO modules are arranged to form the iterative decoding structure for a serially concatenated code. The superscripts i and o denotes the variables of the inner decoder and the outer decoder respectively. The output from the demodulator, $\mathbf{P}(\mathbf{c}^i; I)$, is entered into the inner decoder along with the *a priori* information of the input bits of the inner code, $\mathbf{P}(\mathbf{u}^i; I)$ (which is the interleaved version of the output bits from the outer code). For the first iteration[2], since no additional information is known about the output from the outer code, $\mathbf{P}(\mathbf{u}^i; I)$ is assumed to have a uniform distribution. The output, $\mathbf{P}(\mathbf{u}^i; O)$, is the updated *a posteriori* probabilities of the input bits to the inner code, which is then de-interleaved and used as the input ($\mathbf{P}(\mathbf{c}^o; I)$) to the outer decoder. Unless the distribution of the pre-encoded information data stream is known, a uniform distribution is assumed for the *a priori* probability of the input to the

[2] We define the "first iteration" as the first pass through the decoding process, before any feedback.

outer encoder, $\mathbf{P}(\mathbf{u}^o; I)$. From the outer decoder, $\mathbf{P}(\mathbf{u}^o; O)$ contains the estimates of data bits, which is used to make a hard decision.

The power of iterative decoding of serial concatenated codes comes from the updated *a posteriori* probability of the outer code codewords, $\mathbf{P}(\mathbf{c}^o; O)$. It contains "extra" information that is introduced by the decoding operation from the coding trellis. This is known as the extrinsic information. After passing through the interleaver, this is fed back to the inner decoder to be used as the *a priori* probability of the input bits, $\mathbf{P}(\mathbf{u}^i; I)$ and the code is decoded again. With updated *a priori* information, the estimate of the data will improve with the second iteration. Because of the presence of the interleaver between the inner and outer codes, this extrinsic information is weakly correlated with the output from the demodulator [1] and can be used as additional decoding information. If the extrinsic information is strongly correlated, then no additional information is contained and performance will not improve with each feedback step. The decoding operation can be repeated to obtain a lower bit error rate (BER), but we will see that with each iteration, there will be diminishing returns.

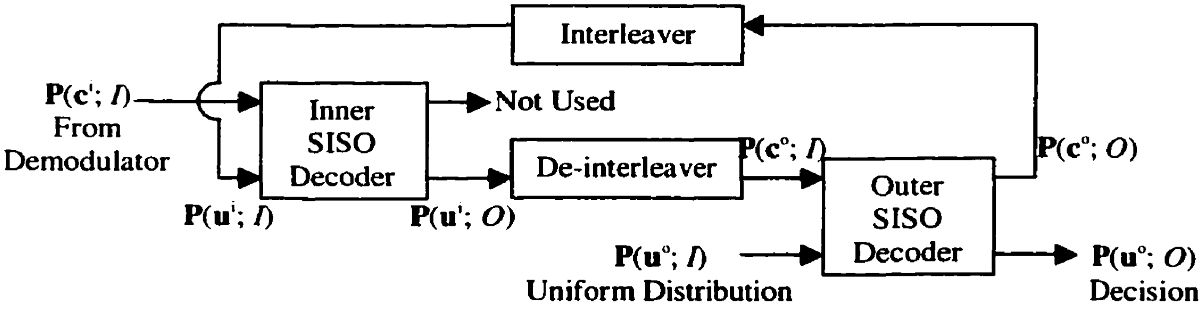

Figure 2.10: Iterative decoding scheme for serial concatenated codes.

2.3.3 Maximum *A Posteriori* Algorithm

The maximum *a posteriori* (MAP) algorithm attempts to maximize the a posteriori probabilities of the state transitions and the output of the codes. We will be using a version of the symbol-by-symbol MAP algorithm derived by Bahl, Cocke, Jelinek and Raviv [20], also known as the BCJR algorithm. The MAP algorithm we implement remains identical to the BCJR algorithm, but the form of the output is extended to include the necessary information needed for iterative decoding.

The algorithm operates in block form. That is, it receives and saves the entire transmitted block before the algorithm operates on the data. This is needed because it performs both forwards and backwards recursion on the data sequence. The memory and computational complexity of this algorithm grows linearly with the block length. Due to the memory requirements and complexity of this algorithm, it may not be practical for use when the block length is large.

For use in a SISO module, the algorithm must calculate **P(c**; *O*) and **P(u**; *O*) from the probability distributions $\Pr(Y_t \mid c_t)$ (the input **P(c**; *I*)) and $\Pr(u_t)$ (the input **P(u**; *I*)). These values can be calculated using the equations:

$$\mathbf{P}(\mathbf{c};O):\quad \Pr(c_t \mid Y_1^\tau) = K_c \sum_{\substack{(m',m)\\ c(m',m)=c_t}} \alpha_{t-1}(m')\beta_t(m)\Pr(u_t = u(m',m)) \quad , \qquad (2.5)$$

$$\mathbf{P}(\mathbf{u};O):\quad \Pr(u_t \mid Y_1^\tau) = K_u \sum_{\substack{(m',m)\\ u(m',m)=u_t}} \alpha_{t-1}(m')\beta_t(m)\Pr(Y_t \mid c_t = c(m',m)) \ . \qquad (2.6)$$

The variables and notation used are:

m',m = the previous state (at *t*-1), and the current state (at *t*) respectively.

$c(m',m)$ = the codeword associated with the state transition from m' to m.

$u(m',m)$ = the input associated with the state transition m' to m.

K_c, K_u = normalization constants.

The values $\alpha(m)$ and $\beta(m)$ are calculated using forward and backward recursions respectively:

$$\alpha_t(m) = \sum_{m'} \alpha_{t-1}(m') \Pr(u_t = u(m',m)) \Pr(Y_t \mid c_t = c(m',m)),$$

$$\beta_{t-1}(m') = \sum_{m} \beta_t(m) \Pr(u_t = u(m',m)) \Pr(Y_t \mid c_t = c(m',m)).$$

Then, to decode, we first initialize $\alpha_0(m)$ and $\beta_\tau(\mathrm{m})$ using the boundary conditions. For a block properly terminated at the zero-state at both ends, it is:

$$\alpha_0(m) = \begin{cases} 1 & m = 0 \\ 0 & m = 1, \ldots, M-1 \end{cases} \quad \text{and} \quad \beta_\tau(m) = \begin{cases} 1 & m = 0 \\ 0 & m = 1, \ldots, M-1 \end{cases},$$

where M is the number of states in the code trellis. The forward recursion $\alpha(m)$ can be calculated and stored as the symbols Y are received at the input. When the entire sequence Y_1^τ is received, calculate and store the backward recursion $\beta(m)$. After obtaining the values $\alpha(m)$ and $\beta(m)$, the outputs can be calculated using Equations (2.5) and (2.6). The full derivation of these equations can be found in Appendix A.

Compared with the other algorithms used in this thesis, the MAP algorithm is optimal and has the best performance. However, the complexity is also the highest among the three algorithms. In the next section, we will describe an approximation that will reduce the complexity of the MAP algorithm while sacrificing a small BER performance.

2.3.4 Max-Log-MAP Algorithm

In the previous section, the MAP algorithm was described in probability form. The algorithm could also work in the logarithm domain without a performance penalty. Inputs and outputs could be accepted in log form, or converted internally. The recursion formulas become:

$$L_{\alpha,t}(m) = \ln\left[\sum_{m'} \exp\left(L_{\alpha,t-1}(m') + \lambda_t(u;I) + \lambda_t(c;I) \right) \right],$$

$$L_{\beta,t-1}(m') = \ln\left[\sum_{m} \exp\left(L_{\beta,t}(m) + \lambda_t(u;I) + \lambda_t(c;I) \right) \right],$$

where $\lambda(u;I)$ and $\lambda(c;I)$ are the log forms of the inputs. Then the output equation can be calculated using:

$$\lambda(c;O) = \ln\left[K_c \sum_{\substack{(m',m) \\ c(m',m)=c_t}} \exp\left(L_{\alpha,t-1}(m') + L_{\beta,t}(m) + \lambda(u;I) \right) \right],$$

$$\lambda(u;O) = \ln\left[K_u \sum_{\substack{(m',m) \\ u(m',m)=u_t}} \exp\left(L_{\alpha,t-1}(m') + L_{\beta,t}(m) + \lambda(c;I) \right) \right].$$

These equations will yield performance results equivalent to the non-log forms since no actual changes have been made except the transfer to the log domain. To reduce the complexity of the MAP algorithm, we can use the approximation $\ln(e^{L_1} + \ldots + e^{L_n}) = \max(L_1, \ldots, L_n)$. Thus, we can replace the summations with the max operator and simplify the calculations. The equations now become:

$$L_{\alpha,t}(m) = \max_{m'} \left(\; L_{\alpha,t-1}(m') + \lambda_t(u;I) + \lambda_t(c;I) \; \right),$$

$$L_{\beta,t-1}(m') = \max_{m} \left(\; L_{\beta,t}(m) + \lambda_t(u;I) + \lambda_t(c;I) \; \right),$$

$$\lambda(c;O) = \max_{\substack{(m',m) \\ c(m',m)=c_t}} \left(\; L_{\alpha,t-1}(m') + L_{\beta,t}(m) + \lambda(u;I) \; \right) + \ln(K_c),$$

$$\lambda(u;O) = \max_{\substack{(m',m) \\ u(m',m)=u_t}} \left(\; L_{\alpha,t-1}(m') + L_{\beta,t}(m) + \lambda(c;I) \; \right) + \ln(K_u).$$

This algorithm using the simplified equations is known as the max-log-MAP algorithm. The memory requirements are the same as the MAP algorithm, while the computing complexity is slightly simpler.

2.3.5 Soft Output Viterbi Algorithm

Both the MAP and max-log-MAP algorithms attempt to minimize the bit error probabilities, with max-log-MAP using approximations to reduce complexity. The Viterbi algorithm (VA) attempts to minimize the probability of a sequence error. Unlike the MAP-based algorithms, the VA maximizes the *a posteriori* probability of the entire sequence instead of each state separately.

The advantage of the VA over the MAP algorithms is that the computational complexity and memory requirements are lower. Also, while the VA could delay decoding until the entire sequence is received (like the MAP algorithms), we'll see that it is not necessary and the algorithm could operate in "continuous-mode". Classical VA made hard decisions, which is unsuitable for use in iterative decoding, where soft decisions are needed. In [21], Hagenauer and Hoeher made modifications to the VA to provide soft outputs, called the Soft-Output Viterbi Algorithm (SOVA), which was later adapted to be used in a form of feedback decoding [22].

First, the basis of the VA will be described, and then the modifications for soft output will be presented. The VA works by maximizing the *a posteriori* probability of the entire received sequence. This is done by calculating the metrics of the paths going through the trellis and choosing the one with the smallest metric. An exhaustive search of every path through the trellis is impractical and the VA avoids this by calculating and storing only the survivor paths at each node of the trellis.

Consider a trellis where for each node, there are two branches entering and two branches exiting. At time *t*, there are two paths, path 1 and path 2, with different path metrics entering a node *n* (Figure 2.11). Assume that path 1 has the smaller path metric. Then, it is easy to see that regardless of the metrics of branches U and V leaving node *n*, the paths exiting node *n* that continue from path 1 will always have a lower path metric than the paths that continue from path 2. Path 1 is called the survivor path. Since for all paths exiting node *n*, the ones continuing from the survivor path will have a lower path metric, all non-survivor paths can be discarded and removed from consideration without loss of information. Thus, at each stage of the trellis, only one survivor path needs to be saved per node.

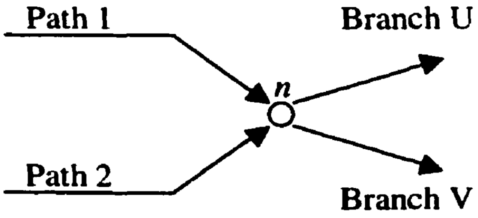

Figure 2.11: Viterbi algorithm illustration.

Until the trellis terminates at a known state, it is not known which survivor path from the different nodes is the most likely candidate. It would seem that we could not decode until the entire sequence is received and a single survivor path is chosen. However, beyond a certain depth in the past stages, it is likely that all the survivor paths have merged. That is, past a certain depth, all the previous states and state transitions for the all the survivor paths are the same. Once a merge occurs, we can decode all the bits up to the point of the merge.

A merge is not guaranteed, and it may be impractical to store all survivor paths for the entire sequence (with memory requirements increasing linearly with sequence length). The common approach to this problem is to truncate at a certain depth, δ, called the truncation depth. At time t, a decoding decision is made at stage t-δ, regardless of a merge. The decoding is done using the survivor path that currently has the lowest path metric. Once a decision is made, all the information prior to the truncation depth may be discarded. To minimize decoding errors, the truncation depth should be chosen such that the survivor paths will be merged with high probability.

The steps to decode using the VA (at stage t) are as follows:

1. Calculate the path metrics $M_t\,(m', m)$ of the paths entering each state m.
2. Determine and store the survivor path and its corresponding metric $M_t(m) = \min_{m'}\{\ M_t(m',m)\ \}$ for every m.
3. Make decision at truncation depth δ, choosing the survivor that currently has the lowest path metric $\min_{m}\{\ M_t(m)\ \}$.

The path metric is defined recursively as:

$$M_t(m',m) = M_t(m) + B_t(m',m), \tag{2.7}$$

where $B_t(m', m)$ is the branch metric given by (in log form):

$$B_t(m',m) = -\ln\left[\ \Pr(u_t = u(m',m))\ \right] - \ln\left[\ \Pr(Y_t \mid c_t = c(m',m))\ \right], \tag{2.8}$$

which can also be described in log-likelihood form if preferred. For classical VA with no *a priori* information (which assumes equal probability) under AWGN, the branch metric simplifies to the familiar metric $B_t(m', m) = (y_t - x_t)^2$, which is the square of the Euclidean distance.

Now we will describe the procedure to obtain soft information from the VA that was first presented in [21] and later improved upon in [23]. The algorithm operates as before, calculating the path metrics and decoding hard decisions, but with an additional step to calculate the reliabilities.

Again, consider a trellis node at time t with two paths entering it. Label the two paths, path 1 and path 2, with metrics M_1 and M_2 respectively. Then, the probability of the correct path is proportional to the exponential of the negative of the metrics,

$$\Pr(\text{path } p) \propto e^{-M_p}\ , \quad p = 1,2\,.$$

Let path 1 be the path with the smaller metric (the survivor path), so that $M_1 \leq M_2$. The probability of selecting the wrong path (i.e., selecting path 1, when path 2 is the correct path) is given by,

$$p_e = \frac{e^{-M_2}}{e^{-M_1} + e^{-M_2}} = \frac{1}{1 + e^{M_2 - M_1}} = \frac{1}{1 + e^{\Delta}}, \tag{2.9}$$

where Δ is difference of the path metrics M_2 - M_1, which is always ≥ 0 from the definition above. With probability p_e, the algorithm has made errors in all stages where the information bits of the two paths differ.

Assume that the probability of error at stage j in path 1, $\hat{p}_j^1$, has been calculated and stored from a previous step. When the information bits of the two paths differ, which is the case when $\hat{u}_j^1 \neq \hat{u}_j^2$, the algorithm can then update the probabilities according to the equation:

$$\hat{p}_j^1 \leftarrow p_e(1 - \hat{p}_j^1) + (1 - p_e)\hat{p}_j^1. \tag{2.10}$$

That is to say that the probability of error of bit j is equal to the probability of choosing the wrong path when bit j was previously correct plus the probability of choosing the correct path when bit j was previously incorrect. To ease calculations, this update can also be done using the log-likelihood ratio. The log-likelihood is given by,

$$\hat{L}_j^1 = \ln \frac{1 - \hat{p}_j^1}{\hat{p}_j^1}. \tag{2.11}$$

Substituting (2.9) and (2.11) into the update equation (2.10) and simplifying gives us the equation,

$$\hat{L}_j^1 \leftarrow \ln \frac{1+\exp(\Delta+\hat{L}_j^1)}{\exp(\hat{L}_j^1)+\exp(\Delta)},$$

which can be approximated by,

$$\hat{L}_j^1 \leftarrow \min(\Delta, \hat{L}_j^1)\,. \tag{2.12}$$

This is the update criterion used in [21], which ignores the case when $\hat{u}_j^1 = \hat{u}_j^2$, leading to simpler implementations.

In [23] and [24], the case with $\hat{u}_j^1 = \hat{u}_j^2$ is also considered. When $\hat{u}_j^1 = \hat{u}_j^2$, it does not matter which of the two paths is chosen as the survivor, since the same decision will be made. Therefore, Δ does not update $\hat{L}_j^1$ since there is no new reliability information. Now, consider a third path, path 3, that merged with path 2 in the past. This is illustrated in Figure 2.12. Assume that the bit j between path 2 and path 3 is different. It can be seen that $\hat{L}_j^2$ contains the path reliability between these two paths. Since $\hat{u}_j^1 = \hat{u}_j^2$ and $\hat{u}_j^2 \neq \hat{u}_j^3$, then $\hat{u}_j^1 \neq \hat{u}_j^3$. In this case, $\hat{L}_j^1$ should be updated using the path reliability between path 1 and path 3. This is given by $\Delta + \hat{L}_j^2$. For the positions when $\hat{u}_j^1 = \hat{u}_j^2$ the algorithm updates $\hat{L}_j^1$ using the equation:

$$\hat{L}_j^1 \leftarrow \min(\Delta + \hat{L}_j^2, \hat{L}_j^1)\,. \tag{2.13}$$

From the reliability information and decisions made by the algorithm, the *a posteriori* probabilities of the input bits, $\Pr(u_t \mid Y_1^\tau)$, can be calculated. For iterative decoding of serial concatenated codes, we will also need the *a posteriori* probabilities of the codewords, $\Pr(c_t \mid Y_1^\tau)$. These can be obtained using the same methods as the input bits. At each stage of decoding, the algorithm will also store and compare the output of the state transition in addition to the input that causes the transition. The reliability is updated using the same criterion as the input bits, with the equations:

$$c^1_{j,k} = c^2_{j,k}: \qquad \hat{L}^1_{c_{j,k}} \leftarrow \min(\Delta + \hat{L}^2_{c_{j,k}}, \hat{L}^1_{c_{j,k}}),$$
$$c^1_{j,k} \neq c^2_{j,k}: \qquad \hat{L}^1_{c_{j,k}} \leftarrow \min(\Delta, \hat{L}^1_{c_{j,k}}).$$

where $c_{j,k}$ represents the k^{th} bit of the output codeword at stage j. With the output codewords of the chosen path together with the reliability information, $\Pr(c_t \mid Y_1^\tau)$ for the SISO module can be obtained.

Without having to calculate backwards recursion, the SOVA is computationally less complex than both the MAP and max-log-MAP algorithms. The performance of the three algorithms used in an iterative decoding system will be analyzed in the following chapters.

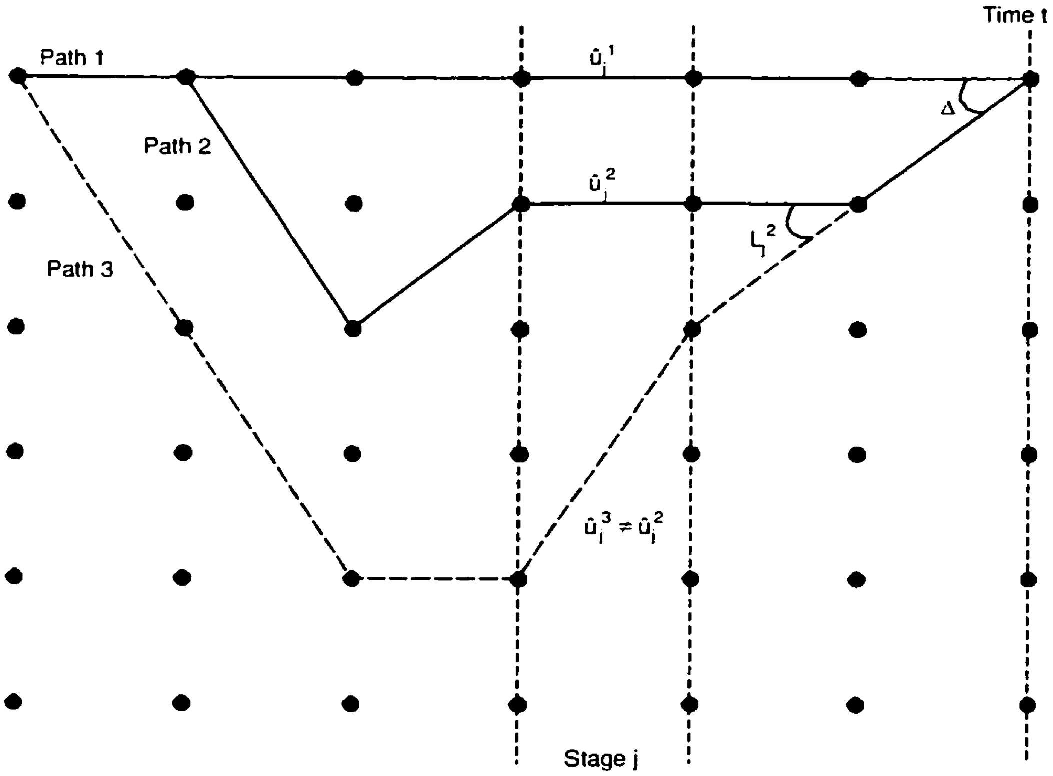

Figure 2.12: Trellis paths.

Chapter 3

Non-Coherent Decoding underAWGN

In this chapter, the performance of a CPM system demodulated using a frequency discriminator together with iterative decoding under AWGN conditions will be investigated. Various parameters are used and their effects on the overall system performance are compared. Among these parameters are the effects of the outer code, the inner code, combined demodulation/decoding, the use of different algorithms, and the metrics used in demodulation.

3.1 System Model

In this section, the model used for the simulations will be described. In chapter 2, the background information of the different components was presented. It will now be shown how these will form the system used in this thesis.

3.1.1 Transmitter

As stated in chapter 2, the transmitter consists of the coding and modulation aspects of the system. A serial concatenated coding system together with GMSK modulation is used. Using the CPM decomposition by Rimoldi [15], the modulation is separated into a continuous phase encoder and a memoryless modulator. The inner code is then combined with the CPE to form a new encoder. This transmission system is shown in Figure 3.1.

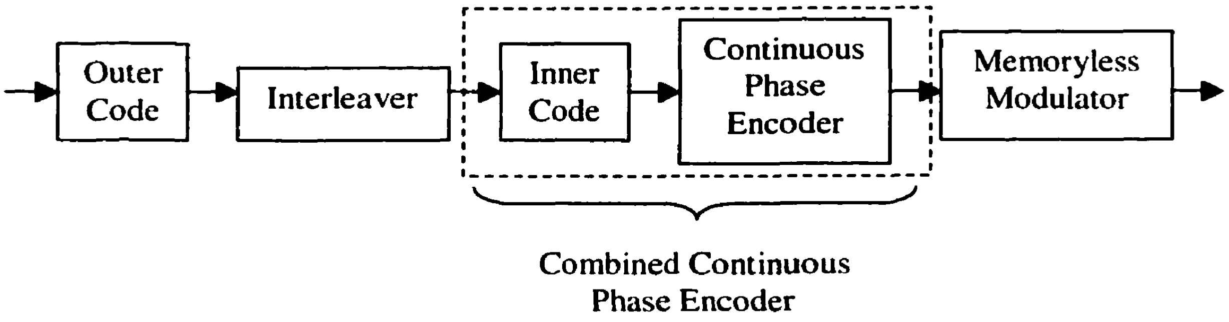

Figure 3.1: Coded CPM Transmission System using Serial Concatenated Codes.

The interleaver used throughout the simulations is a pseudo-random interleaver with a length equal to the block length. An interleaver specifically matched to the constituent codes may yield better performance, but the design of such an interleaver is beyond the scope of this thesis.

For our GMSK system, the CPE representation of the modulation is illustrated in Figure 3.2. It can be seen that it consists of delayed taps plus a differential encoder. This modulation memory can be used as the inner code of the serial concatenated coding system, but under non-coherent demodulation, we will show that this may not be enough to obtain the "turbo"-property of iterative decoding. With the memoryless modulator, the outputs of the CPE are mapped to the in-phase and quadrature components of a baseband signal, which is then modulated up to the desired carrier frequency. The modulator is shown in Figure 3.3.

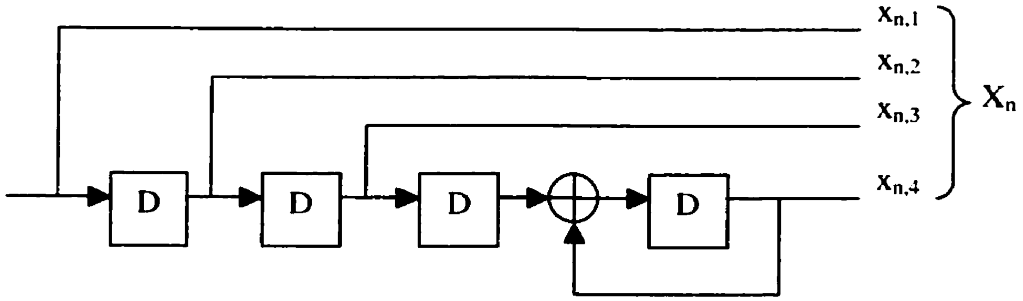

Figure 3.2: Continuous Phase Encoder for GMSK $L = 0.3$.

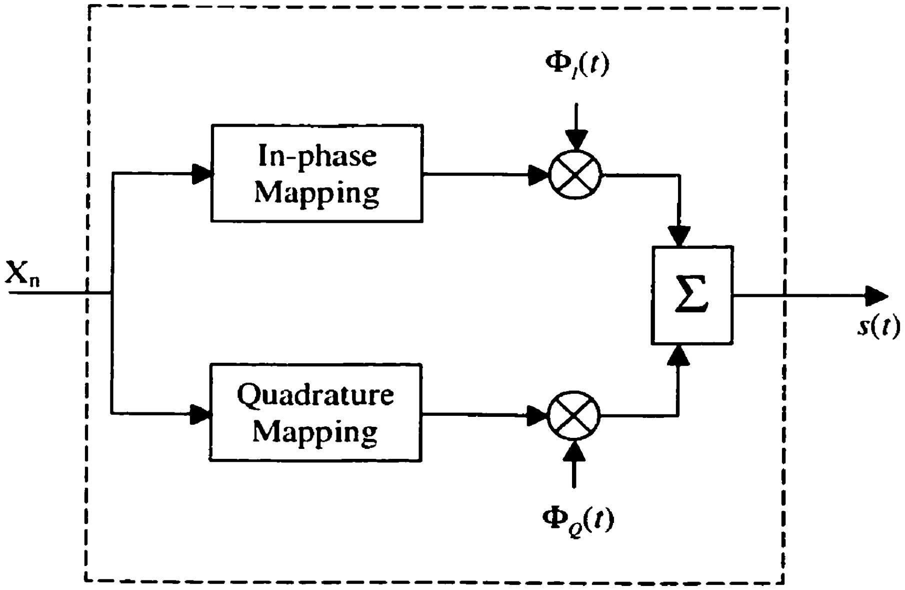

Figure 3.3: Memoryless modulator.

The equations in the memoryless modulator used to modulate the baseband signal to passband are,

$$\Phi_I(t) = \sqrt{1/2}\cos[\ 2\pi f_1(t+nT)+\phi_o\ ],$$

$$\Phi_Q(t) = \sqrt{1/2}\sin[\ 2\pi f_1(t+nT)+\phi_o\ ].$$

Note that f_1 is a shifted version of the actual carrier frequency, a result of the method used to separate the coding from the modulation. f_1 is related to f_c (for binary signalling) by

$$f_1 = f_c - {}^{h}\!/_{2T} ,$$

where h is the modulation index of the CPM signal, which is 0.5 in our case.

3.1.2 Receiver

We use a frequency discriminator system to demodulate the received signal and determine the frequency (or phase) of the signal at the sample times. Both ISD and SH sample methods could be used. For post-detection processing, an iterative decoding strategy is used, based on the combined sequence estimation for GMSK and the decoding of the inner and outer codes. This is illustrated in Figure 3.4.

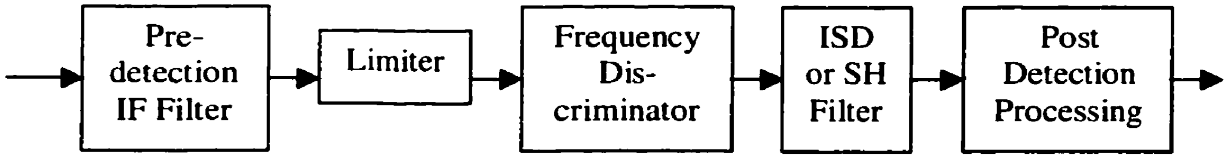

Figure 3.4: Frequency Discriminator Demodulation/Decoding Structure.

Other than the different sampling methods, the frequency discriminator demodulation remains the same for all simulations. The parameters varied in the decoding portion are the decoding algorithms in the SISO modules and the metrics used to calculate channel reliability. Comparing the three different algorithms, we can determine how the system performs using both optimal and sub-optimal decoding algorithms. The metrics used are the optimal metric for a discriminator and the simpler but less accurate Gaussian metric.

By studying the two metrics, we can find out if using the Gaussian metric is adequate in an iterative decoding system.

The effects of separating the demodulation from the decoding will also be investigated. Separating the demodulation from decoding could reduce the computation needed during the decoding, but it will also lose the benefit of the modulator memory in subsequent iterations.

The pre-detection IF filter used in the demodulator system is a Gaussian filter, with a frequency response given by

$$H(f) = e^{-\frac{\ln 2}{2}\left(\frac{f}{B}\right)^2},$$

where B is the 3 dB bandwidth. This filter is commonly used with discriminator detection and provides good performance over other filters [27], [25]. The filter bandwidth of our filter was chosen to be $BT = 1$, which was found to remove a high percentage of the out-of-band noise without overly distorting the signal.

3.1.3 Noise Statistics of a Frequency Discriminator

The channel used in obtaining the results for this chapter is the AWGN channel. For a signal, $s(t)$, sent from the transmitter, the received signal is

$$r(t) = s(t) + n(t),$$

where $n(t)$ is white Gaussian noise with a dual-sided power spectral density of $N_0/2$. Frequency discriminator detection demodulates using the frequency or phase of the signal. During processing, the signal undergoes some non-linear transformations. As a result, the phase noise at the output of the frequency discriminator is not Gaussian. Some simplified expressions of this noise are given by Pawula in [25] and [26]. The expression for the probability density of the phase noise at the output of a frequency discriminator with an ISD filter is, for $|\psi| \le \pi$,

$$p(\psi) = \frac{e^{-U}}{2\pi}\left[\ \cosh V + \frac{1}{2}\int_0^{\pi} (U \sin\alpha + W\cos\psi)\cdot\cosh(V\cos\alpha)\cdot e^{W\sin\alpha\cos\psi}\, d\alpha\ \right], \quad (3.1)$$

where

$$U = \frac{1}{2}[\ \rho(t) + \rho(t-T)\ ]$$

$$V = \frac{1}{2}[\ \rho(t) - \rho(t-T)\ ]$$

$$W = \sqrt{U^2 - V^2}\ .$$

$\rho(t)$ is the time-varying SNR given by

$$\rho(t) = \frac{E}{N_o}\frac{a^2(t)}{T\int_{-\infty}^{\infty}|H(f)|^2\, df},$$

with $H(f)$ being the frequency response of the IF filter and $a^2(t)$ being the square of the normalized filtered signal amplitude given by

$$a^2(t) = \left(\ \int_{-\infty}^{t} h(t-\tau)\cdot\cos\theta(\tau)\, d\tau\ \right)^2 + \left(\ \int_{-\infty}^{t} h(t-\tau)\cdot\sin\theta(\tau)\, d\tau\ \right)^2 .$$

To simplify our analysis and calculations, let us assume the case that $a^2(t)$ is invariant between bit periods, so that $\rho(t) = \rho(t\text{-}T)$. This gives $U = \rho(t)$, $V = 0$, $W = U$ and Equation (3.1) simplifies to

$$p(\psi) = \frac{e^{-U}}{2\pi}\left[\ 1 + U\int_0^{\pi/2} (\sin\alpha + \cos\psi)\cdot e^{U\sin\alpha\cos\psi}\ d\alpha\ \right]. \tag{3.2}$$

Evaluating this expression to obtain the channel reliability is still fairly computationally intensive. There is also the need to obtain the SNR for the variable U, which may be difficult and add unwanted complexity to the demodulation. For practical systems, assuming the noise is Gaussian is a desirable alternative and it is used in [17] and [18] for Viterbi decoding with good results. Both metrics are used in this study to determine the performance penalty involved in assuming Gaussian noise over using the more accurate noise statistic.

3.2 Simulation Results

In this section, the simulation results of the frequency discriminator based, iterative decoding system using various parameters is presented. Table 3.1 shows the standard parameter values used throughout this thesis, which do not change unless stated otherwise.

A sampling rate of 10 samples / symbol was used in our simulations. It gives a good representation of the analog GMSK signal and using more samples will give a negligible gain in the simulation's accuracy. For the coding blocksize, 10 000 bits per block was chosen. This ensures that the transient effects associated with the IF filters affecting the results are kept to a minimum. 10 iterations were used in the iterative decoding process.

This number provides good BER performance in most cases, with diminishing returns for each iteration thereafter.

The standard outer code used throughout our simulations is given by the octal generators (23, 35), which has 16 trellis states. According to [2], conventional non-recursive convolutional codes with a large and possibly odd free distance should be used as the outer code. The code (23, 35) has a free distance of 7, and it was used in [28] with good results. A rate-1 differential encoder is used as an inner code in addition to the modulation memory. The reason for this extra inner coding will be discussed in section 3.2.2.

Table 3.1: Simulation Parameters.

Sampling Rate	10 samples / symbol
Blocksize	10 000 bits
Outer Code	(23, 35)
Inner Code	Rate 1 Differential Encoder, $G = [\frac{1}{1+D}]$
Modulation Method	GMSK, B_oT=0.3, L=3
Demodulation Method	Frequency Discriminator, Combined Demod/Decoding
Noise Statistic	Equation (3.2)
Number of Iterations	10

3.2.1 Coherent vs. Non-coherent Demodulation

This section will briefly describe the SNR penalty associated with using a non-coherent discriminator demodulation over coherent matched filter demodulation. For coherent demodulation, we used a bank of matched filters to obtain the channel reliabilities as in [9]. The first graph, Figure 3.5, shows the uncoded performance of both coherent and non-coherent demodulation with maximum likelihood sequence estimation (MLSE) using the MAP algorithm. At a BER of 10^{-4}, the coherent detection is about 3.5 dB better than the non-coherent discriminator detection. Note that due to the nature of discriminator detection removing some memory from the signal, its trellis has fewer states than for coherent detection.

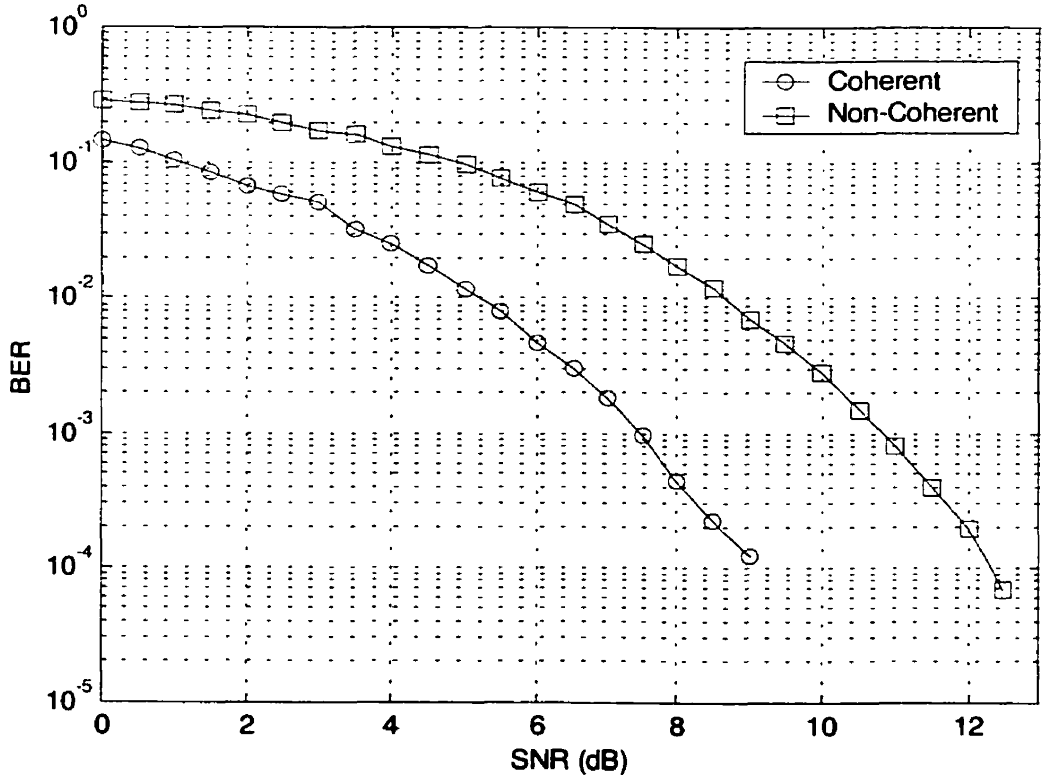

Figure 3.5: Comparison of coherent and non-coherent detection under AWGN.

The coded performances are shown in Figure 3.6. The difference in SNR performance of the convolutionally coded coherent and non-coherent systems has increased to 5.0 dB. Using iterative decoding, both detection methods obtained significant improvement in performance, but the difference between them remained at approximately 5.0 dB. This shows that for coded GMSK, there will be a significant SNR penalty by using non-coherent over coherent detection when using the same codes. To lessen this penalty, more powerful codes must be used when demodulating with a discriminator detector.

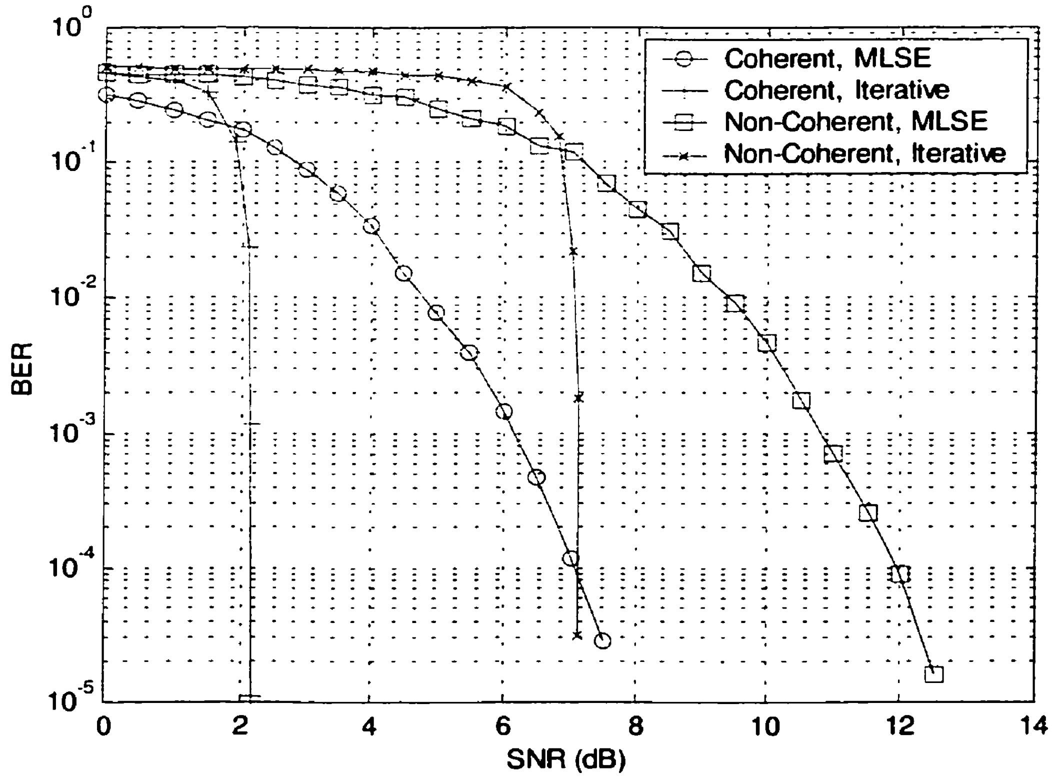

Figure 3.6: Performance of coding between coherent and non-coherent detection.

3.2.2 Presence of Coding

It is well known that applying error control coding will improve the performance of a transmission system beyond a certain crossover SNR. We will study how various combinations of coding will offset the performance loss associated with the use of a frequency discriminator receiver. Four different coding cases, decoded using the MAP algorithm, are shown in Figure 3.7. The four cases are:

- Uncoded, MLSE only.
- Convolutional coding (23, 35) + MLSE.
- Serial concatenated coding, GMSK as inner code, iterative decoding 10 iterations.
- Serial concatenated coding, GMSK+ differential inner code, 10 iterations.

From Figure 3.7, we see that applying a relatively weak convolutional code only improved the performance by 0.5 dB at a BER of 10^{-4}. However, applying the same code with an interleaver and using iterative decoding, we obtain a gain of approximately 3.5 dB at the same BER over just convolutional coding. This indicates that using an iterative decoding approach may be more preferable than using more powerful and complex codes.

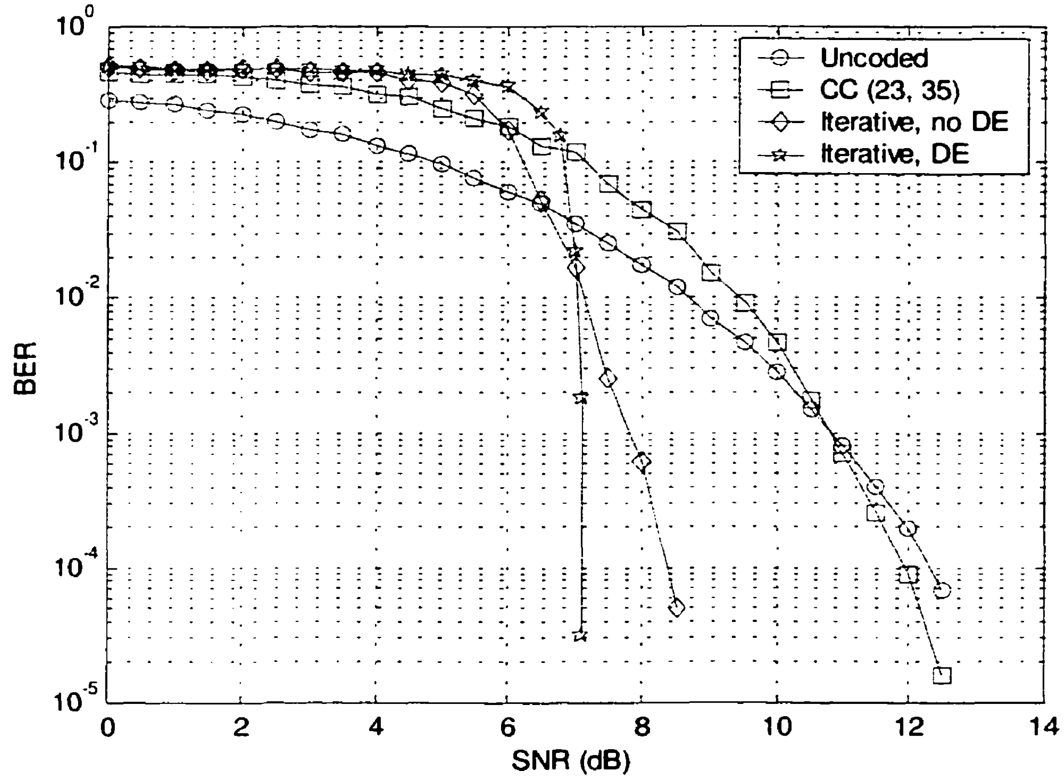

Figure 3.7: Performance of coding under AWGN.

Comparing between the serial concatenated coding schemes, it is obvious that with an added inner differential encoder, the BER drop off is much steeper than without. While the system with no additional inner code start to exhibit a "turbo"-effect at a lower SNR, the effect is not as pronounced. This is an unfortunate side effect of using a frequency discriminator for demodulation. The demodulation method removes modulation memory from the system, and takes away the recursive nature of the GMSK signal. As stated in [2], a recursive inner code is necessary for designing good serial concatenated codes used for iterative decoding. Without the recursive memory, only the memory caused by the imposed ISI remains. Strictly speaking, this is not a convolutional code, but it can be seen that it could still be used for iterative decoding in a limited capacity. Fortunately, adding a simple rate-1 recursive differential code to the system would solve this problem.

Using the combined CPE approach, the complexity added to the transmitter is little, if any. On the receiver side, the decoding complexity remains the same, since the differential encoding can be added to the trellis without requiring extra states.

It should also be noted that without an additional recursive inner code, the feedback information would reach convergence with very few iterations. Figure 3.8 shows that after 5 iterations, the gain of having further iteration steps is negligible.

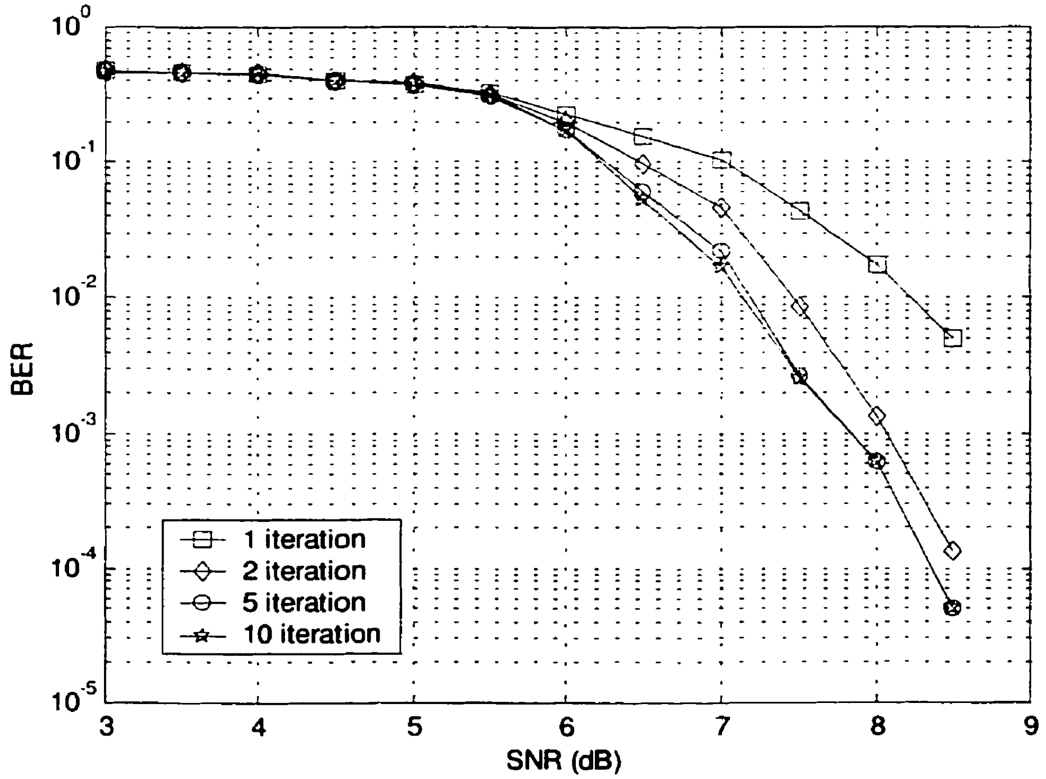

Figure 3.8: Effect of no additional inner code on iterative decoding.

3.2.3 Combined Demodulation / Decoding

It has been established previously that without an additional recursive inner code, a frequency discriminator iterative decoding system performs poorly compared with a system with a recursive inner code. To avoid the early convergence in the iterative process, a simple recursive rate-1 differential code should always be added for a frequency discriminator with iterative decoding. This leaves us with two possible demodulation/decoding methods. We can demodulate the signal fully, giving us a bit probability from the channel, and then use the iterative decoding technique on the differential inner code and the outer code. Or, we can combine the trellises of the CPM signal and the differential code and operate jointly on the demodulation and decoding.

While combining the CPM and differential code trellis does not increase the number of the states over the CPM trellis, by separating the demodulation and decoding, we can reduce the number of states necessary for the decoding of the inner code. This will reduce the computing operations required when decoding the inner code, which can be significant as the number of iterations increase. Of course, with a reduction of decoding states and not using the extra memory from the modulation, the performance can be expected to degrade. Looking at Figure 3.9, we see that combining the trellises improves the performance by 0.5 dB at a BER of 10^{-4} over the case when the trellises are decoded separately.

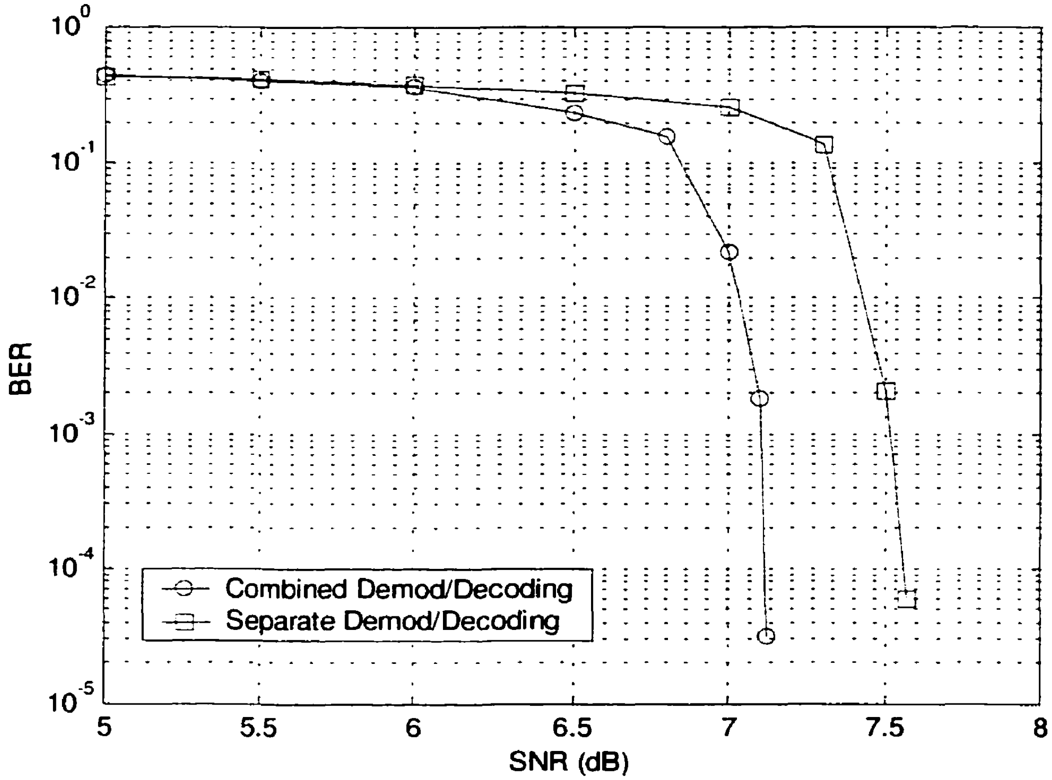

Figure 3.9: Performance of combined demodulation/decoding.

3.2.4 Performance of Different Algorithms

As was presented in chapter 2, different algorithms could be used in the SISO modules for iterative decoding. The three popular algorithms that are used are the MAP, max-log-MAP and the SOVA. Out of the three, the MAP algorithm is the most complex but it is also optimal in calculating the *a posteriori* probabilities. The max-log-MAP simplifies the MAP algorithm by making an approximation using the max operator instead summations. With the SOVA, the least complex of the three, a survivor path approach is used to minimize the sequence errors, but leads to sub-optimal bit error probabilities. A comparison of these algorithms can be found in [29] and their computing complexities are compared in [30].

Here, we will concentrate on how these different algorithms affect the performance of the discriminator detection system. The plots are shown in Figure 3.10. The algorithm labelled as "SOVA (non-optimal)" is the SOVA algorithm that updates the reliabilities only when the bits between the merging paths are not equal, as described in [21]. This makes the implementation simpler with a slight decrease in performance.

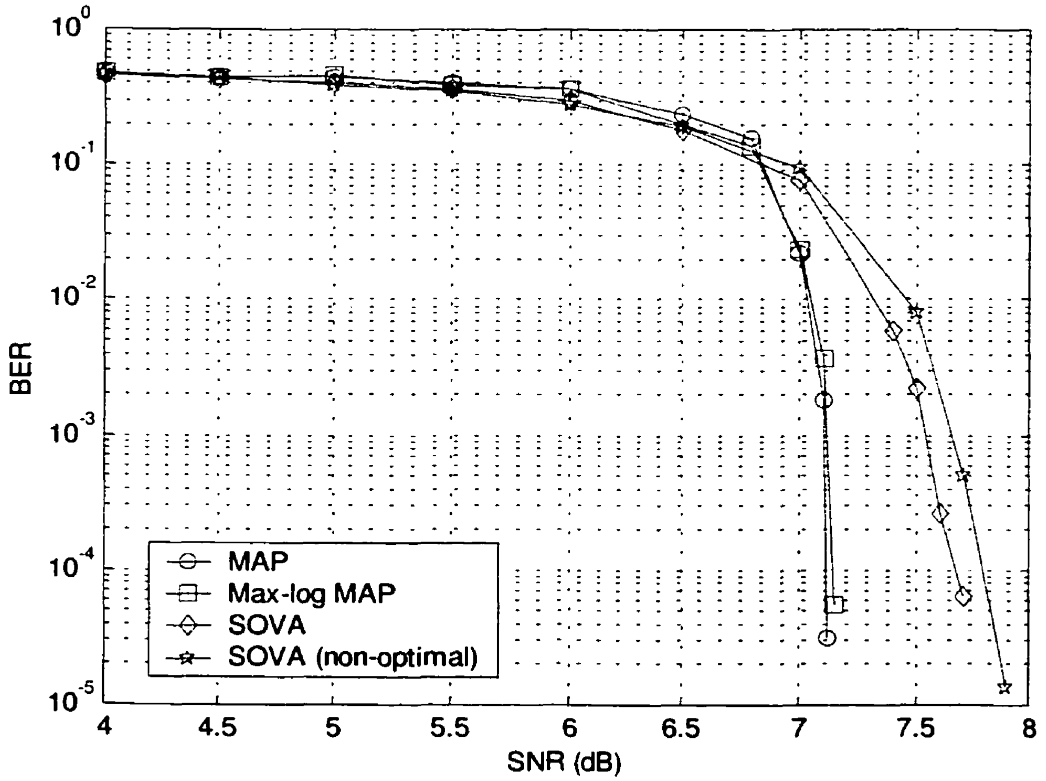

Figure 3.10: Performance of different algorithms in discriminator detection.

Between the MAP and max-log-MAP algorithms, there is little difference in perfomance when used in our iterative decoding system. Using the SOVA over the MAP algorithms incurs a penalty of about 0.6 dB in SNR at a BER beyond 10^{-4}. The memory and computational requirements for the SOVA is significantly less than the MAP algorithms, so this may be an acceptable trade-off when considering the algorithm to use in the SISO modules. Another consideration is the "non-optimal" SOVA. At a BER of 10^{-4}, the difference between the two SOVA implementations is approximately 0.1 dB. This indicates that the probability of a previously merged path having a better reliability is low and ignoring the update when the bits between the merging paths are equal does not have a large effect on the algorithm's performance.

For comparison purposes, the performance of the four algorithms under coherent detection is shown in Figure 3.11. From this graph, we see that the difference in SNR between the four algorithms under coherent detection is greater than under discriminator detection. There is a 0.3 dB difference between the MAP and the max-log-MAP algorithms, and a 0.9 dB difference between MAP and SOVA. The difference between the two SOVA has also increased to 0.25 dB. A possible explanation as to why using a more optimal algorithm yields less gain in a frequency discriminator detector system is that the channel reliability is less accurate. That is, the noise statistic used does not model the noise of the channel completely. For a coherent detection system under AWGN, we know that the noise at the matched filter output is Gaussian. For a discriminator detector, we use Equation (3.2) by making assumptions and approximations about the channel, so that the accuracy is not exact. The effect of the accurate modelling of the noise statistic on the performance of an iterative discriminator system is the topic of the next section.

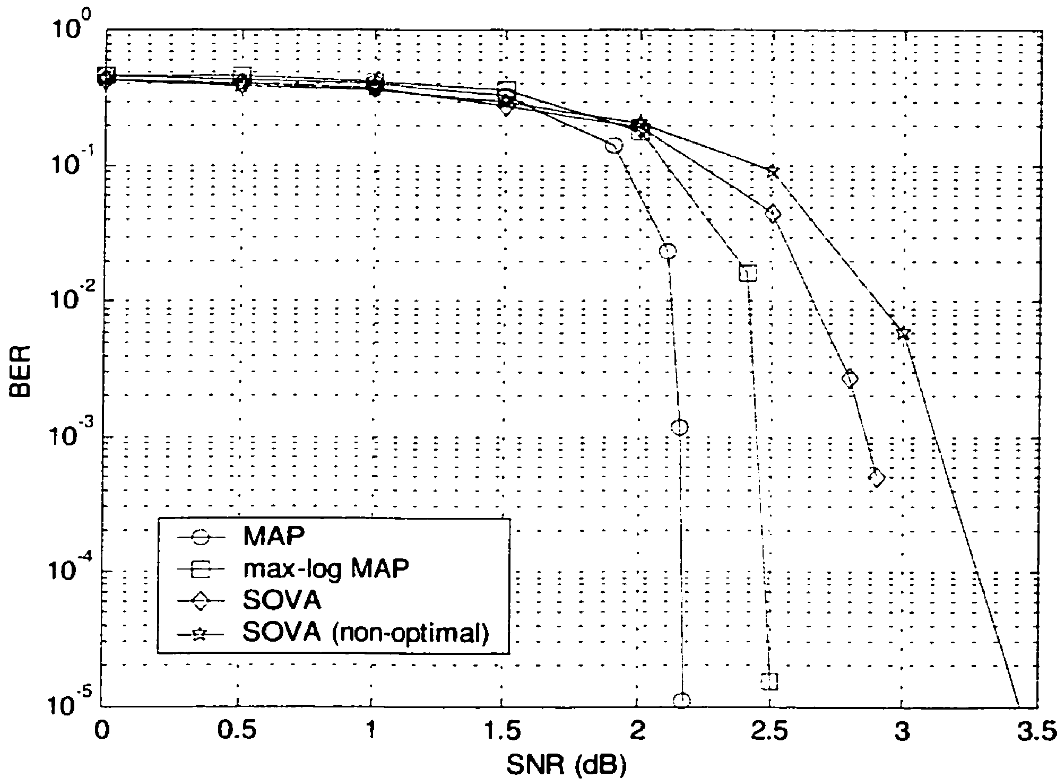

Figure 3.11: Performance of different algorithms under coherent detection.

3.2.5 Dependence on the Accuracy of the Noise Statistic

The noise statistic assumed for our non-coherent discriminator output is Equation (3.2). This equation, while simplified from (3.1), is still computationally complex to use in a practical receiver system. For non-encoded MLSE detection of GMSK, assuming Gaussian noise at the output of the discriminator, Iwanami [17] and Fonseka [18], was still able to obtain good results without using the more accurate noise statistic. How the noise statistic accuracy will affect the discriminator iterative decoding system will now be investigated. Figure 3.12 shows the uncoded performance of using Equation (3.2) and Gaussian noise statistics for both MAP and SOVA algorithms. We see that for uncoded MLSE, there is no difference between Equation (3.2) and Gaussian noise when using

SOVA. When using the MAP algorithm, there is only a 0.3 dB difference. This suggests that with the optimality of the MAP algorithm, it was able use the more accurate noise information to improve its decoding performance, while the sub-optimal SOVA was not able to take advantage of the more accurate information.

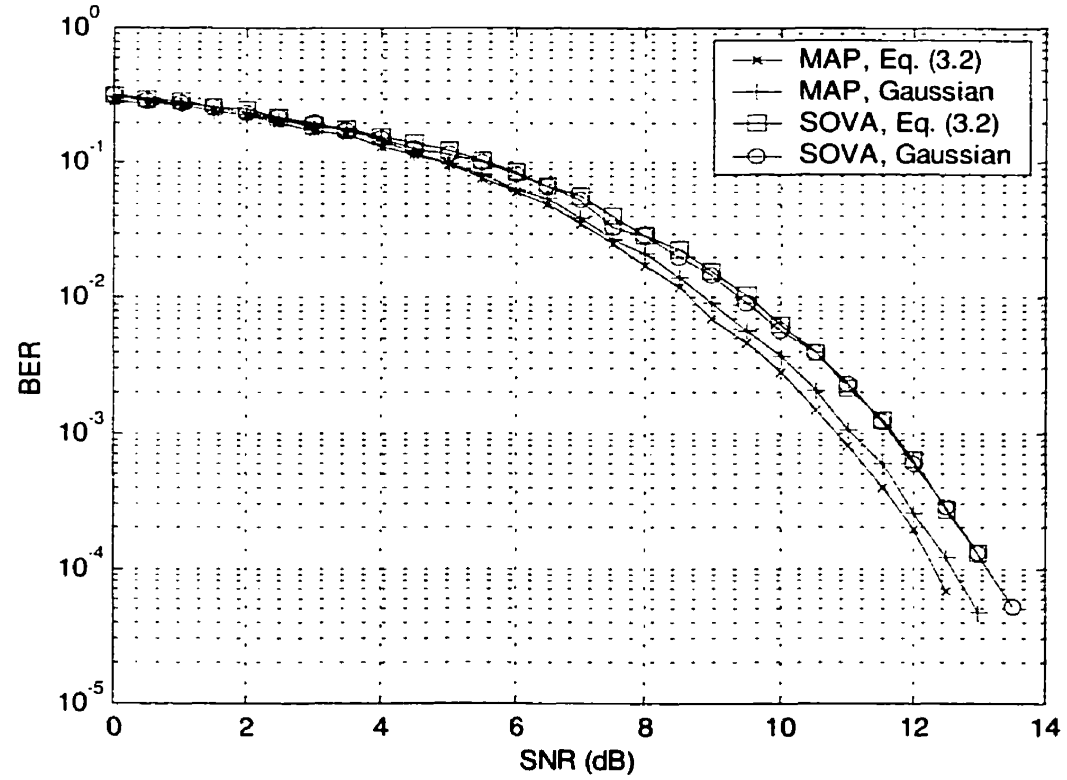

Figure 3.12: Uncoded performance of different noise statistics.

With iterative decoding, there is a drastic change between assuming Gaussian noise and Equation (3.2). The plots are shown in Figure 3.13. For both MAP and SOVA algorithms, when assuming Gaussian noise, there is a performance loss of about 2 dB at a BER of 10^{-4}. This indicates that while for MLSE, the improved noise statistic was unable to improve the performance, the increased reliability of the soft output is important for iterative decoding. When using the Gaussian statistic, there are times when it will be overly optimistic or too conservative with the reliability. Then, during feedback, the updated *a priori* probabilities may not be able to correct for the demodulation error. The loss in SNR when using the Gaussian statistic with SOVA is a little more than with MAP, increasing the difference between the two algorithms to 0.9 dB from 0.6 dB.

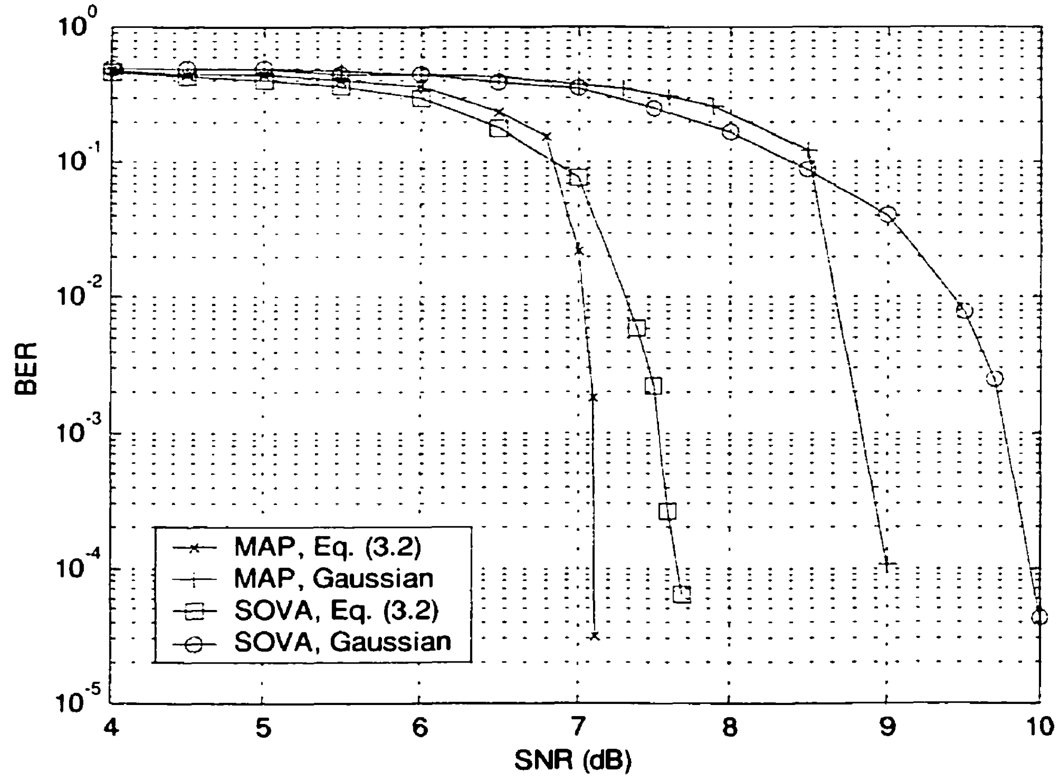

Figure 3.13: Iterative decoding performance of different noise statistics.

3.2.6 Effect of Outer Code

The performance issues of having an inner code in a discriminator detection system were discussed in a previous section. Now we turn our attention to the outer code to determine how the outer code will affect the performance of the system. The standard outer code used throughout was the (23, 35) code with a free distance of 7. We will compare how the system performs when codes with distances of 5, 6 and 8 are used. The convolutional codes are all rate-½ codes. The statistics of these codes are shown in Table 3.2. These codes are taken from [31], and they provide the highest free distance for their memory length.

Table 3.2: Convolutional codes used in this thesis.

Generator (Octal Form)	Memory	d_{min}	Trellis States
(5, 7)	2	5	4
(15, 17)	3	6	8
(23, 35)	4	7	16
(53, 75)	5	8	32

According to [2], a recursive outer code is not necessary to obtain good performance for iterative decoding and non-recursive codes are sufficient. As seen from our various results, unlike the inner code, a non-recursive outer code does not negatively affect the convergence of the iterative decoding process. Regardless of its recursive nature, it is expected that with a more powerful outer code, it will improve the overall performance of the coding system. The results obtained with the codes from Table 3.2 using SOVA is shown in Figure 3.14. From the plot, it can be seen that using a more powerful code does indeed improve the performance of the system. However, the relative gain diminishes as the complexity of the codes is increased. While there is almost a 0.7dB gain between the best and worst code used, the increase between using the codes (23, 35) and (53, 75) is less than 0.1dB. As an outer code, the codes (23, 35) and (15, 17) are adequate for the

AWGN channel from a performance standpoint without the added complexities of the more powerful rate-½ codes.

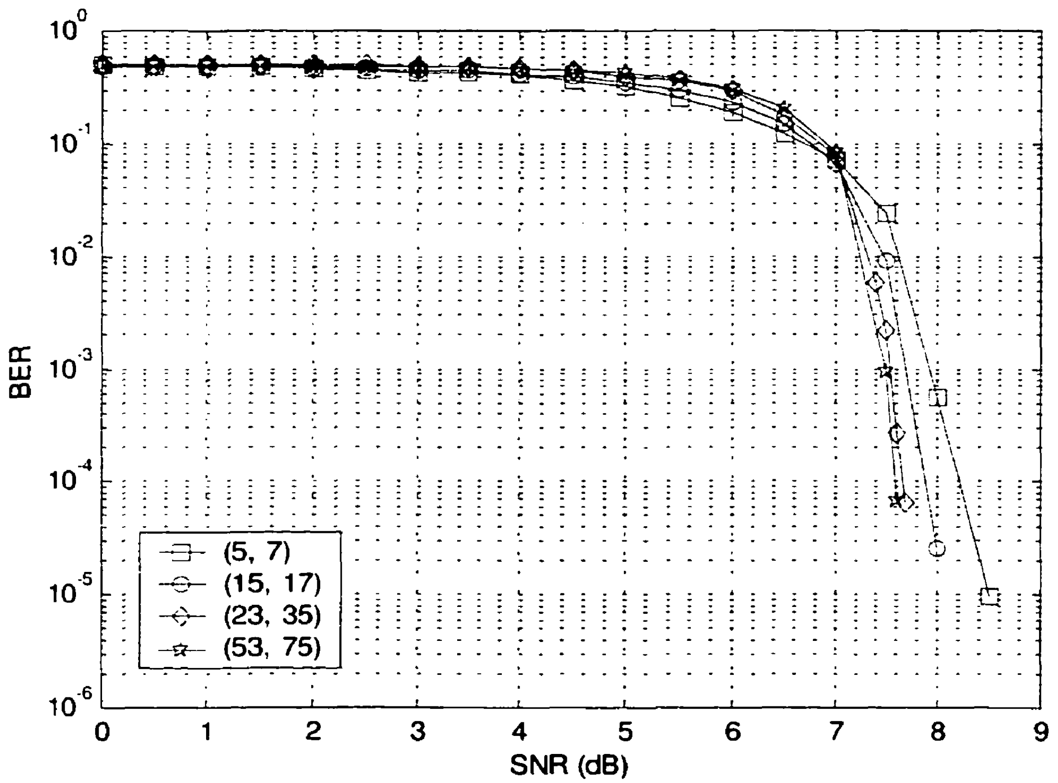

Figure 3.14: Effect of outer code on system performance.

Chapter 4

Non-Coherent Decoding under Fading

In the previous chapter, the performance of a discriminator system with iterative decoding under an AWGN channel was presented. This chapter will discuss the performance of the system when used in fading conditions. The channel is modelled as a multiplicative Rayleigh distributed fading amplitude followed by AWGN as shown in Figure 4.1. Due to the nature of fading causing SNR to drop and Doppler shifts causing slight changes in the received frequency, the performance will be worse than the standard AWGN channel.

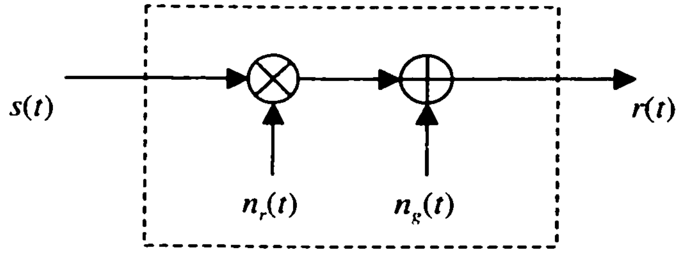

Figure 4.1: Fading channel model.

4.1 Fading Channel Model

The fading channel used in our studies is the Rayleigh fading channel. That is, the fading amplitude, $n_r(t)$, follows a Rayleigh PDF. This channel is derived from a propagation model where there is no direct line-of-sight between the transmitter and the receiver. All transmission is done via reflections and refractions of the original signal from various obstacles and the different signal components are assumed to have approximately equal strength. If there is a direct line-of-sight between the transmitter and the receiver, then one signal component will have a greater received power and the PDF of the fading amplitude will become Rician.

A fading channel can be described as being frequency selective or non-frequency selective. In the case when the fading is frequency selective, the amount of fading is different for different frequencies within the signal bandwidth. In non-frequency selective fading, the fading amplitude is the same for all frequencies of the signal. For our simulations, we have chosen to assume a non-frequency selective channel, which is sufficient for the narrowband GMSK signal.

An important parameter in a fading channel model is the Doppler spread of the channel. The relative motion between the transmitter and the receiver causes the received frequencies to shift, with each signal component shifting by different amounts due to the angle of arrival, resulting in the Doppler spread. Using a scattering model developed by Clarke [16], the power spectrum of the Doppler spread is given by,

$$S(f)=\begin{cases}\dfrac{E}{4\pi f_m}\dfrac{1}{\sqrt{1-\left(\dfrac{f-f_c}{f_m}\right)^2}} & |f-f_c|\le f_m\\ 0 & \text{elsewhere,}\end{cases} \tag{4.1}$$

where f_m is the maximum Doppler shift. The underlying assumptions of the Clarke model are that the received signal is composed of horizontally travelling plane waves, each with a statistically independent random phase and an arrival angle that is distributed uniformly between 0 and 2π. Figure 4.2 shows the power spectrum described by Equation (4.1). By adjusting the maximum Doppler shift, f_m, the rate of fading can be controlled.

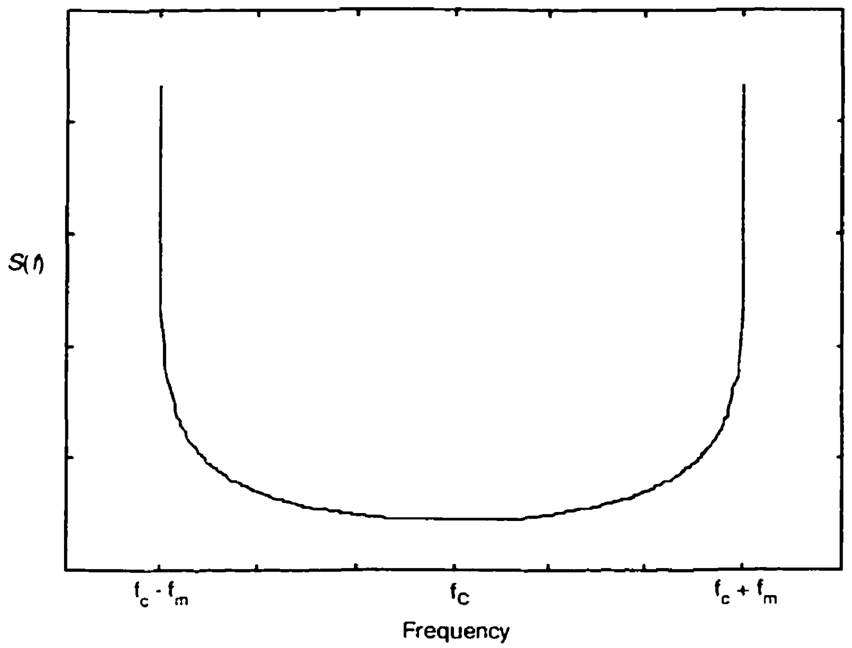

Figure 4.2: Doppler power spectrum.

4.1.1 Simulation Model

To produce the multiplicative Rayleigh fading amplitude noise, we use two independent filtered Gaussian noise sources. The block diagram of the structure is shown in Figure 4.3. The two low-pass filters are identical and it is used to shape the fading spectrum to be equal to the desired Doppler spread. For our channel, the frequency response H(f) of the two filters is given by Equation (4.1) with $f_c = 0$ for a baseband low-pass response. It is possible to simulate a different spectrum, such as one based on a more generalized scattering model developed by Aulin [16] by using this simulation method.

The two noise sources represent the two quadrature components of the fading noise. The final noise output, $n_r(t)$, is the sum of the two noise components and will have an envelope that is distributed according to a Rayleigh PDF.

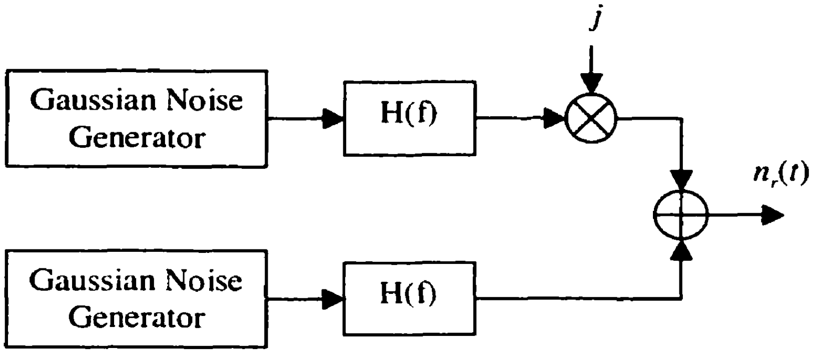

Figure 4.3: Fading amplitude simulator.

4.2 Amplitude Information in Decoding

In a fading channel, the performance of the system can be expected to degrade. In chapter 3, we found that better likelihood information could improve the performance greatly. It might be possible to supplement our likelihood with additional side-information to increase its accuracy. One such additional side-information could be obtained from the amplitude of the received signal.

For a constant envelope CPM signal, no data information is transmitted in the amplitude of the signal. However, due to the multi-path fading environment and the pre-detection narrowband IF filter, the envelope of the signal is distorted and spread over several bit intervals. The demodulation performance could be improved by using amplitude information to remove the non-linear ISI associated with discriminator detection under a fading environment. Studies in this area have been done by Asano and Pasupathy [32] and by Chang and Rohani [8]. More recently, research by Sánchez -Pérez [33], [34], [35] developed metrics for use with the VA combining frequency information from the discriminator and amplitude information from an envelope detector to improve demodulation performance.

Using the amplitude weighted frequency error metric from [35], we hope to further improve the likelihood information for our iterative decoding system under fading conditions. The metric is given by,

$$d(R\theta) = (R\theta)^2, \tag{4.2}$$

where R is the amplitude of the signal and θ is the instantaneous frequency. This metric is Gaussian, so it does not depend on the SNR and it is simple to calculate. The receiver modification to this is fairly simple, only adding an additional envelope detector and multiplier to the system. The new receiver is shown in Figure 4.4.

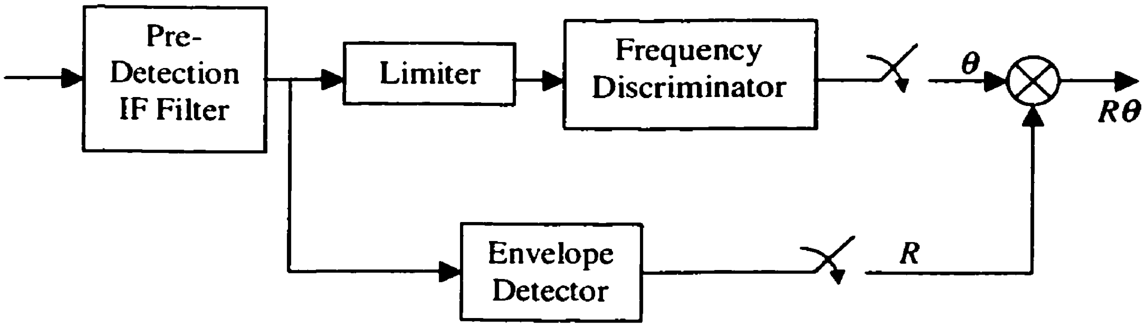

Figure 4.4: Frequency discriminator receiver with amplitude information.

Under AWGN conditions, there is no advantage in adopting this new metric. Without multi-path fading, the envelope remains constant and no additional information is obtained by using the amplitude weighted metric. Figure 4.5 illustrates this for uncoded GMSK using the SOVA.

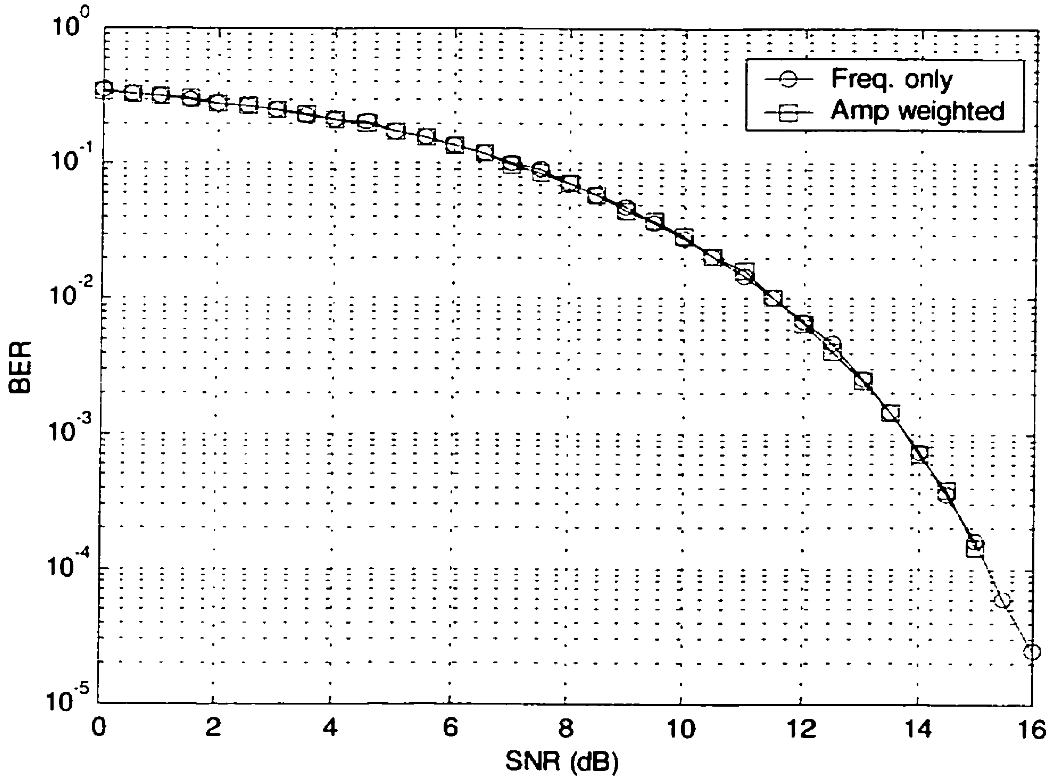

Figure 4.5: Amplitude weighted metric under AWGN.

4.3 Simulation Results for Fading

For the fading channel, results were obtained using the SOVA for the SISO algorithm. While the SOVA suffers a performance penalty over the MAP algorithms, it is a simpler algorithm to implement and it is our intention to determine the performance of a less complex system. Since the amplitude weighted metric was developed for the instantaneous frequency, the ISD filter after the discriminator is not used and instead the receiver uses the SH method. When using SH over ISD, Equation (3.2) is no longer a more accurate metric; so instead, the Gaussian metric will be used.

Other parameters remain the same as those listed in Table 3.1. A fading rate of $f_m = 0.2 / T$ which is representative for fast fading environments is used for the simulations.

4.3.1 Performance of Iterative Decoding under Fading

Figure 4.6 shows the performance of the discriminator detected iterative decoding system under Rayleigh fading conditions. As expected for non-coherent detection, there is a large SNR penalty when it is used in a fading channel. Using only MLSE with no additional coding, the system has not yet reached a BER of less than 10^{-3} at a SNR of 22 dB. With the addition of an iterative decoding system, a BER of 10^{-4} can be reach at 15 dB using an outer code of (23, 35) after 10 iterations. This is a vast improvement over the uncoded case, and it makes the discriminator detection system a much more viable option in a fading channel.

Note that under fading, the use of a more powerful outer code provides a larger improvement in SNR performance than under AWGN. Referring back to Figure 3.14 in chapter 3, there was only about a 0.5 dB improvement using the code (23, 35) over (5, 7) at a BER of 10^{-4}. However, in a multi-path fading environment, the same codes yield an improvement of 2 dB at the same BER. This suggests that the increased distance properties of the more powerful outer codes help correct for the errors introduced due to fading.

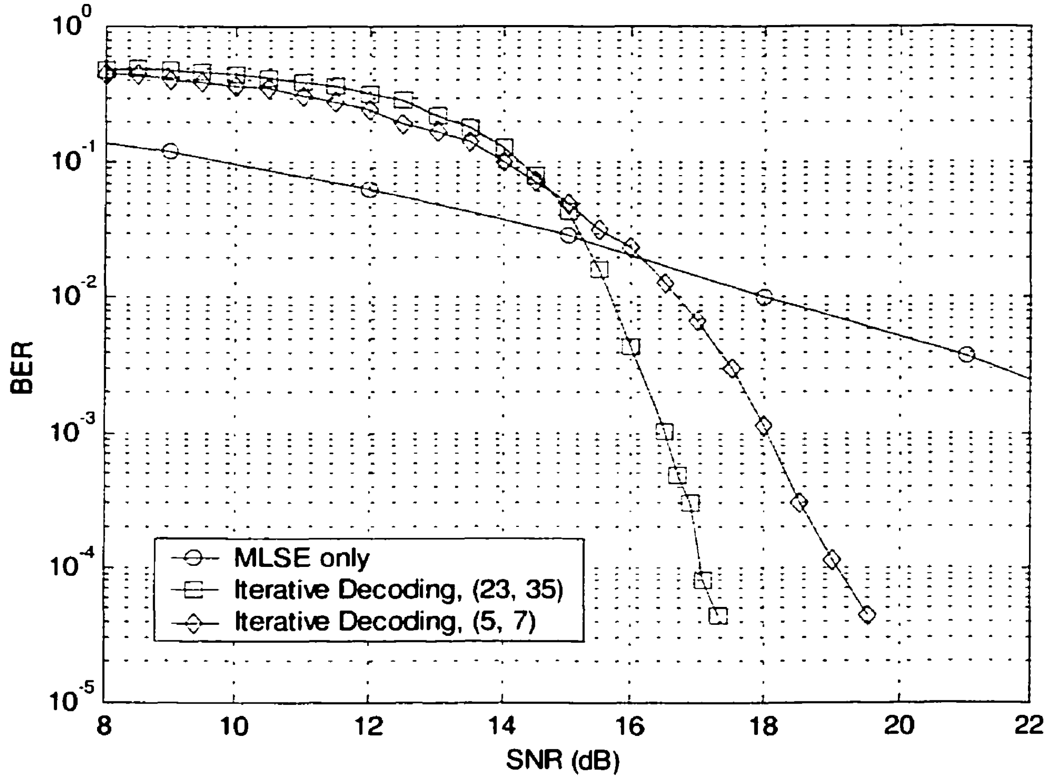

Figure 4.6: Performance of iterative decoding in fading channel.

4.3.2 Effect of Amplitude Information on Performance

Now we present the results of using the amplitude weighted metric over the frequency only metric. Figure 4.7 and Figure 4.8 shows the result of using the amplitude weighted metric with the outer codes (23, 35) and (5, 7) respectively. Unlike in the uncoded AWGN case (Figure 4.5), there is much improvement when the amplitude weighted metric is used. With the additional information from the envelope detector, the decoding performance at around a BER of 10^{-4} improved by 1.2 dB for the (5, 7) outer code. For the (23, 35) code, the improvement was even greater at 2 dB.

This indicates that with improved channel reliabilities and therefore improved decoding of the inner code, the outer codes will have a bigger impact on the decoding performance. The more powerful outer code is able to utilize the additional amplitude information in the decoding process better, resulting in more SNR gain. Together with the results found in the previous section, where the effect of the outer code is more prominent compared with the AWGN channel, it may be worthwhile to use more powerful outer codes to increase the performance of the iterative decoding system in fading conditions.

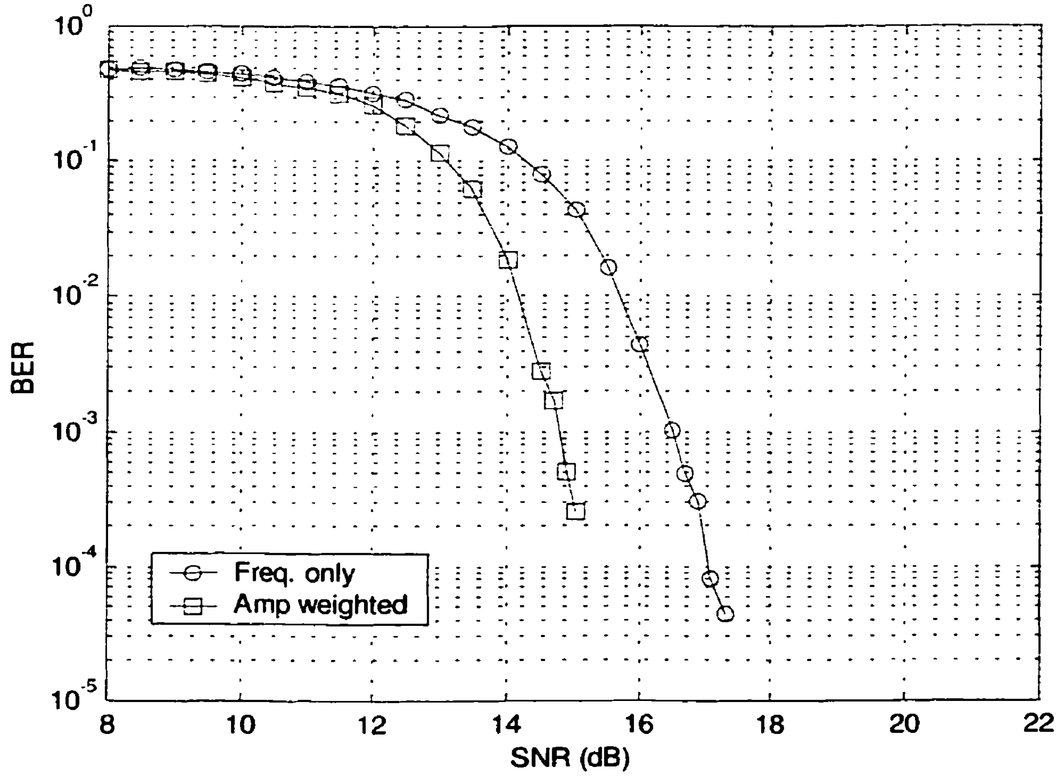

Figure 4.7: Amplitude metric performance with (23, 35) outer code.

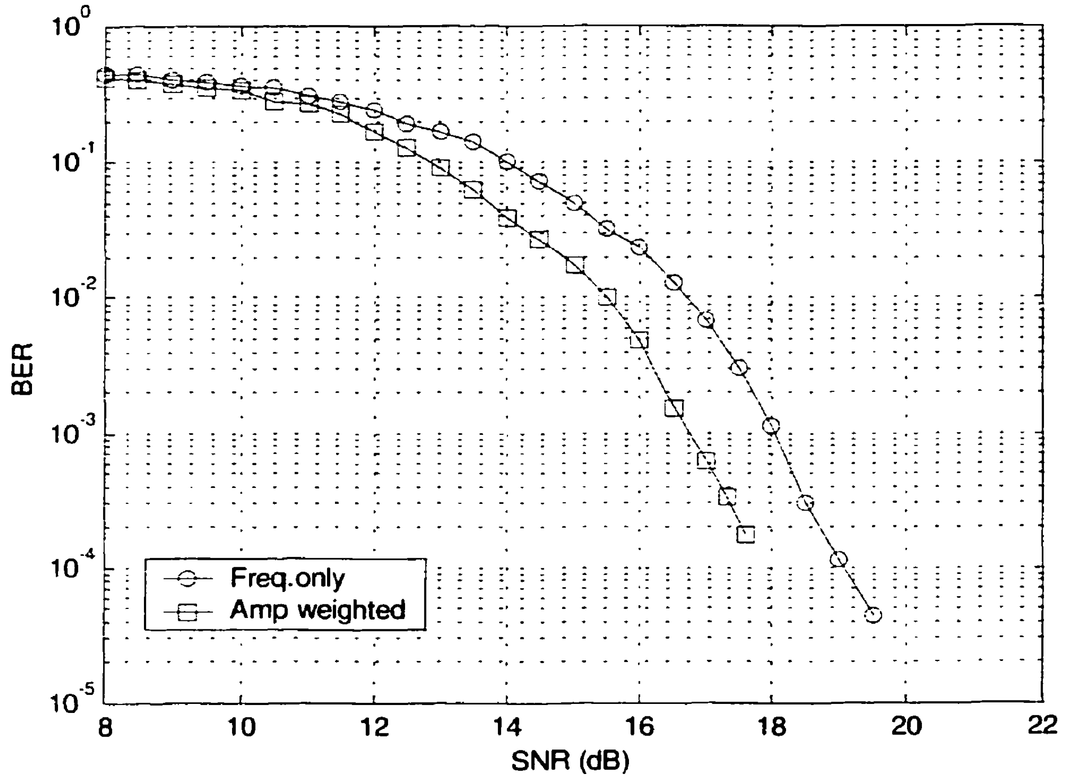

Figure 4.8: Amplitude metric performance with (5, 7) outer code.

4.3.3 Other Parameters' Effect on Performance

The effect of the fading rate of the channel on the performance of the iterative decoding system is shown in Figure 4.9. The difference in SNR to obtain a low BER is quite significant between the fastest ($f_m = 0.8 / T$) and slowest rate ($f_m = 0.02 / T$) presented. Here, we see the system performing better under a hıgher fading rate channel than a slower one. This is to be expected. Since under slower fading, the bursts of error are longer, which is harder for the coding system to correct.

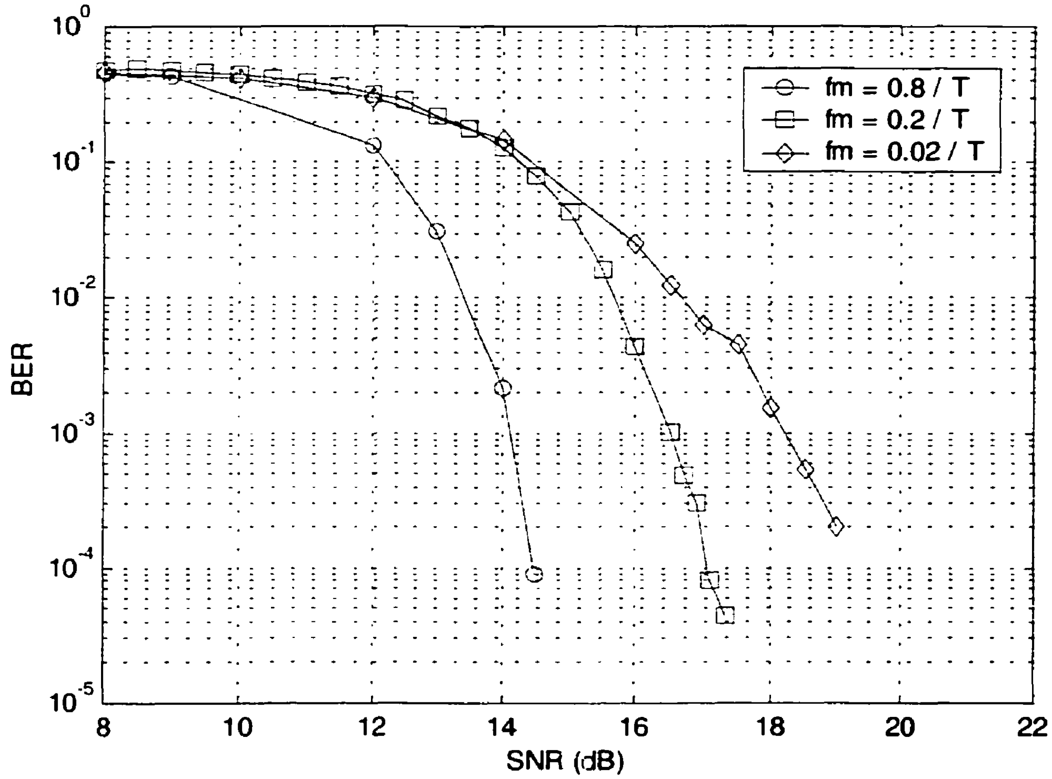

Figure 4.9: Effect of fading rate on performance.

As a reference, the performance difference between the SOVA and the MAP algorithm is shown in Figure 4.10. Compared with the difference in AWGN, it can be seen that the improvement provided by the MAP algorithm is greater in the fading channel. However, the improvement is small compared with the gains of using more powerful outer codes. The MAP algorithm could be used in conjunction with more powerful codes, but the complexity of such a system will be much higher over a similar SOVA system.

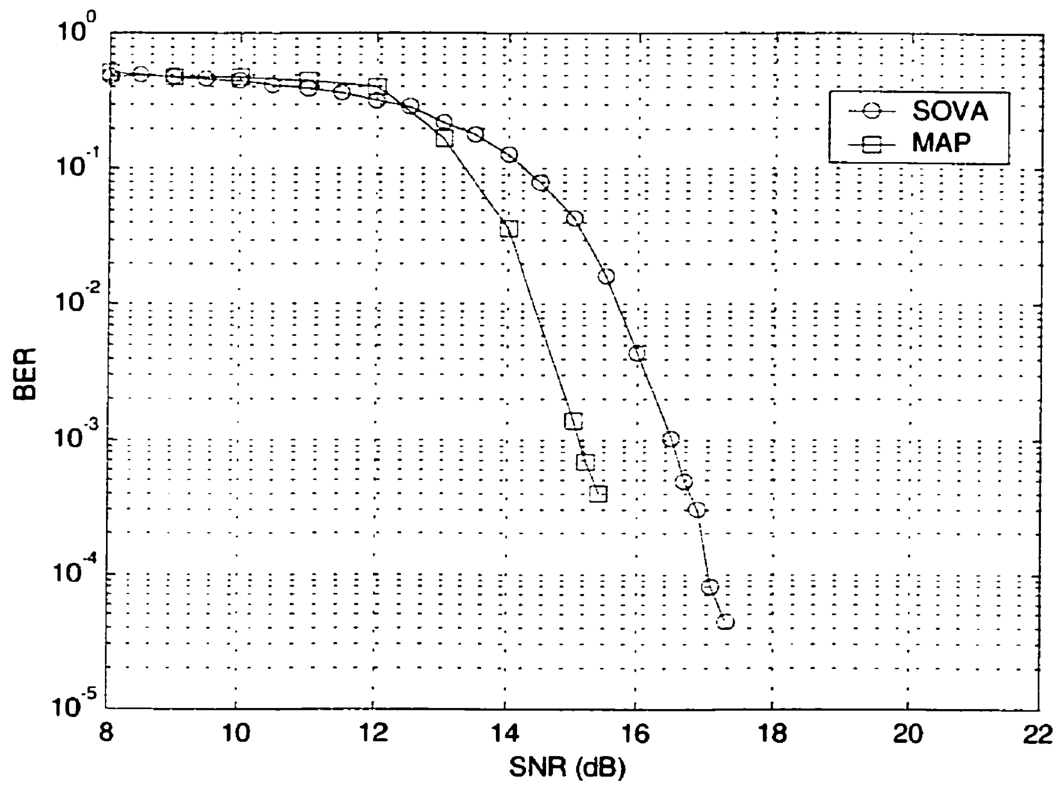

Figure 4.10: Comparison of MAP and SOVA under fading.

Chapter 5

Conclusion

5.1 Summary of Results

We have investigated the performance of a frequency discriminator based, iterative decoding system with GMSK under various conditions and parameters. Two channels were used, the AWGN channel and the Rayleigh fading channel. Different decoding algorithms were compared and tested. The performance of using different metrics was also examined.

Under AWGN, it was found that using a non-coherent discriminator detector suffers a near constant degradation compared with a similarly coded coherent system. Using iterative decoding, the system gained significant improvement, but the discriminator removed the recursive nature of the GMSK signal. This led to a serial concatenated coding system that converges early, and therefore does not perform as well as expected. By reintroducing a rate-1 recursive differential code, the system can be made to perform better without an increase in receiver complexity. The choice of outer code used in the concatenated coding system shows little effect when used in this channel.

The different algorithms were tested and as expected, the more complex and optimal MAP algorithms performed better than the SOVA. The MAP algorithm showed very little gain over the max-log-MAP algorithm. When investigating the effect of the noise statistics' accuracy, it was found that in an uncoded system, there was little gain in using the more accurate statistic for decoding. However, in an iterative decoding system, the difference can be significant when using an accurate statistic over the Gaussian statistic.

The discriminator system was also studied under multi-path fading. Non-coherent discriminator detection performs very poorly without any coding even when using MLSE to remove ISI. With the addition of the iterative decoding system, the performance showed vast improvement. Whereas for the MLSE case, the BER was 2.5×10^{-3} at a SNR of 22 dB, the iterative decoding system reached a BER of 10^{-4} at 17 dB. This was further improved by 2 dB with the use of amplitude information from an envelope detector. More powerful outer codes were also found to give more improvement under fading than in AWGN.

To summarize, when designing a discriminator system using GMSK and iterative decoding, these are the major areas to consider:

- The discriminator removes the recursive nature of GMSK, so an additional recursive inner code is needed for good performance.
- The MAP-based algorithms perform better than the SOVA, with a corresponding increase in computational complexity and memory requirements.
- Using a more accurate noise statistic (likelihood) can have significant impact on system performance, but its calculation could be infeasible in practical situations.
- In a fading channel, additional side-information from the signal amplitude can be used to improve the performance when using a Gaussian noise statistic.
- Increasing the memory length of the outer codes has a diminishing effect on the system performance. It may be better to use a simple outer code and improve performance using another method.

5.2 Suggestions for Further Research

In this thesis, we have shown that iterative decoding can help overcome the performance degradation due to discriminator detection, especially under multi-path fading. There are, however, some areas that still need to be explored. It was found that an additional inner code was needed for the system for it to perform well. We used a simple rate-1 differential code to minimize the complexity added to the system. More powerful inner codes could also be used and it would be worthwhile to find codes that perform well in a discriminator system. This also suggests the design of a concatenated coding scheme matched to the discriminator receiver. With the inner codes, the outer codes and the interleaver designed to complement each other, a better system performance should be possible.

In multi-path fading, the use of additional amplitude information increased the performance of our system considerably. The fading channel was a non-frequency selective Rayleigh channel. In [33], there are other envelope-aided metrics that are found to perform better than the plain amplitude weighted metric in a frequency selective setting. It would be helpful to investigate how these metrics would perform in a channel that can take advantage of the extra amplitude information.

Appendix A

Derivation of the MAP algorithm

The MAP algorithm used in this thesis is based on the BCJR algorithm derived in [20] and the extension for use in the SISO module based on [19].

A.1 Estimating State and Transition Probabilities

The BCJR algorithm was developed to estimate the *a posteriori* probability of the states, S, and its transitions by examining the entire received sequence, Y_1^τ. These two probabilities are given by

$$\Pr(S_t = m \mid Y_1^\tau) = \Pr(S_t = m, Y_1^\tau) / \Pr(Y_1^\tau) \tag{A.1}$$

and

$$\Pr(S_{t-1} = m', S_t = m \mid Y_1^\tau) = \Pr(S_{t-1} = m', S_t = m, Y_1^\tau) / \Pr(Y_1^\tau) \tag{A.2}$$

respectively. The joint probabilities are defined as

$$\lambda_t(m) = \Pr(S_t = m, Y_1^\tau),$$

and

$$\sigma_t(m', m) = \Pr(S_{t-1} = m', S_t = m, Y_1^\tau).$$

Since $\Pr(Y_1^\tau)$ is constant for a given received sequence, Equations (A.1) and (A.2) can be obtained by normalizing $\lambda_t(m)$ and $\sigma_t(m',m)$. For our SISO module, only the state transition probability is needed, so the quantity $\lambda_t(m)$ will be ignored. Now we will define $\sigma_t(m',m)$ with three new quantities:

$$\alpha_t(m) = \Pr(S_t = m, Y_1^\tau)$$

$$\beta_t(m) = \Pr(Y_{t+1}^\tau \mid S_t = m)$$

$$\gamma_t(m',m) = \Pr(S_t = m, Y_t \mid S_{t-1} = m').$$

Expressing $\sigma_t(m',m)$ in terms of the above functions gives:

$$\begin{aligned}
\sigma_t(m',m) &= \Pr(S_{t-1} = m', S_t = m, Y_1^\tau) \\
&= \Pr(S_{t-1} = m', Y_1^{t-1}) \cdot \Pr(S_t = m, Y_t^\tau \mid S_{t-1} = m', Y_1^{t-1}) \\
&= \Pr(S_{t-1} = m', Y_1^{t-1}) \cdot \Pr(S_t = m, Y_t^\tau \mid S_{t-1} = m') \\
&= \Pr(S_{t-1} = m', Y_1^{t-1}) \cdot \Pr(S_t = m, Y_t \mid S_{t-1} = m') \cdot \Pr(Y_{t+1}^\tau \mid S_{t-1} = m', S_t = m, Y_t) \\
&= \Pr(S_{t-1} = m', Y_1^{t-1}) \cdot \Pr(S_t = m, Y_t \mid S_{t-1} = m') \cdot \Pr(Y_{t+1}^\tau \mid S_t = m) \\
&= \alpha_{t-1}(m') \cdot \gamma_t(m',m) \cdot \beta_t(m).
\end{aligned}$$

Steps 3 and 5 are the result of the Markov property. When the current state is given, the probability will no longer depend on the previous states and observation.

A.2 Deriving γ

$\gamma_t(m',m)$ is the probability of arriving at state m and observing the received symbol Y given that the previous state was m'. This is determined the coding and channel characteristics of the system.

$$
\begin{aligned}
\gamma_t(m',m) &= \Pr(S_t = m, Y_t \mid S_{t-1} = m') \\
&= \sum_c \Pr(S_t = m, Y_t, c_t = c \mid S_{t-1} = m') \\
&= \sum_c \Pr(S_t = m \mid S_{t-1} = m') \cdot \Pr(Y_t, c_t = c \mid S_{t-1} = m', S_t = m) \\
&= \sum_c \Pr(S_t = m \mid S_{t-1} = m') \cdot \Pr(c_t = c \mid S_{t-1} = m', S_t = m) \\
&\qquad \cdot \Pr(Y_t \mid S_{t-1} = m', S_t = m, c_t = c).
\end{aligned}
$$

c are the codeword symbols outputted by the state transitions. Given the state transition (m',m), there can be only one possible codeword. That is, $\Pr(c_t = c \mid S_{t-1} = m', S_t = m) = 0$ for all c, except for the one case with $c_t = c(m',m)$ when $\Pr(c_t = c \mid S_{t-1} = m', S_t = m) = 1$. Also, when given state m', the probability of arriving at state m is determined by the *a priori* probability of the input that causes the transition from m' to m. Then $\gamma_t(m',m)$ can be simplified to:

$$
\begin{aligned}
\gamma_t(m',m) &= \Pr(S_t = m \mid S_{t-1} = m') \cdot \Pr(Y_t \mid S_t = m, S_{t-1} = m', c_t = c(m',m)) \\
&= \Pr(u_t = u(m',m)) \cdot \Pr(Y_t \mid c_t = c(m',m)). \qquad \text{(A.3)}
\end{aligned}
$$

A.3 The Recursion Formulas

The quantities α and β can be defined and calculated using recursion. The forward recursion, $\alpha_t(m)$ is given by:

$$
\begin{aligned}
\alpha_t(m) &= \Pr(S_t = m, Y_1^t) \\
&= \sum_{m'} \Pr(S_{t-1} = m', S_t = m, Y_1^t) \\
&= \sum_{m'} \Pr(S_{t-1} = m', Y_1^{t-1}) \cdot \Pr(S_t = m, Y_t \mid S_{t-1} = m', Y_1^{t-1}) \\
&= \sum_{m'} \Pr(S_{t-1} = m', Y_1^{t-1}) \cdot \Pr(S_t = m, Y_t \mid S_{t-1} = m') \\
&= \sum_{m'} \alpha_{t-1}(m') \cdot \gamma_t(m', m) \\
&= \sum_{m'} \alpha_{t-1}(m') \Pr(u_t = u(m', m)) \Pr(Y_t \mid c_t = c(m', m)) \qquad \text{(A.4)}
\end{aligned}
$$

The backward recursion, $\beta_{t-1}(m')$, is given by:

$$
\begin{aligned}
\beta_{t-1}(m') &= \Pr(Y_t^\tau \mid S_{t-1} = m') \\
&= \sum_{m} \Pr(Y_t^\tau, S_t = m \mid S_{t-1} = m') \\
&= \sum_{m} \Pr(Y_t, S_t = m \mid S_{t-1} = m') \cdot \Pr(Y_{t+1}^\tau \mid S_t = m, S_{t-1} = m', Y_t) \\
&= \sum_{m} \Pr(Y_t, S_t = m \mid S_{t-1} = m') \cdot \Pr(Y_{t+1}^\tau \mid S_t = m) \\
&= \sum_{m} \gamma_t(m', m) \cdot \beta_t(m) \\
&= \sum_{m} \beta_t(m) \Pr(u_t = u(m', m)) \Pr(Y_t \mid c_t = c(m', m)). \qquad \text{(A.5)}
\end{aligned}
$$

A.4 The Updated Output Equations

First we derive the *a posteriori* probabilities of the input bits, $\Pr(u_t \mid Y_1^\tau)$. Recall that $\Pr(Y_1^\tau)$ is constant for a given sequence. Therefore, we can calculate the joint probability $\Pr(u_t, Y_1^\tau)$ and then normalize for all u_t to obtain $\Pr(u_t \mid Y_1^\tau)$.

$$
\begin{aligned}
\Pr(u_t \mid Y_1^\tau) &= \Pr(u_t, Y_1^\tau) / \Pr(Y_1^\tau) \\
&= K \cdot \Pr(u_t, Y_1^\tau) \\
&= K \sum_{\substack{(m',m) \\ u(m',m)=u_t}} \underbrace{\Pr(S_{t-1} = m', S_t = m, Y_1^\tau)}_{\sigma_t(m',m)} \\
&= K \sum_{\substack{(m',m) \\ u(m',m)=u_t}} \alpha_{t-1}(m') \cdot \beta_t(m) \cdot \gamma_t(m',m) \\
&= K \sum_{\substack{(m',m) \\ u(m',m)=u_t}} \alpha_{t-1}(m') \cdot \beta_t(m) \cdot \Pr(u_t = u(m',m)) \cdot \Pr(Y_t \mid c_t = c(m',m))
\end{aligned}
$$

From the definition of the summation we are only summing the transitions that is caused by $u(m',m) = u_t$; therefore, $\Pr(u_t = u(m',m))$ remains constant for all (m',m) in the summation and can be taken out. This leads to:

$$
\begin{aligned}
\Pr(u_t \mid Y_1^\tau) &= K \cdot \Pr(u_t = u(m',m)) \sum_{\substack{(m',m) \\ u(m',m)=u_t}} \alpha_{t-1}(m') \cdot \beta_t(m) \cdot \Pr(Y_t \mid c_t = c(m',m)) \\
&= K_u \sum_{\substack{(m',m) \\ u(m',m)=u_t}} \alpha_{t-1}(m') \cdot \beta_t(m) \cdot \Pr(Y_t \mid c_t = c(m',m)). \qquad \text{(A.6)}
\end{aligned}
$$

K_u is the normalization constant such that $\sum_{u_t} \Pr(u_t \mid Y_1^\tau) = 1$.

The derivation of the *a posteriori* probabilities of the output symbols, $\Pr(c_t \mid Y_1^\tau)$, is similar to the above.

$$
\begin{aligned}
\Pr(c_t \mid Y_1^\tau) &= \Pr(c_t, Y_1^\tau) / \Pr(Y_1^\tau) \\
&= K \cdot \Pr(c_t, Y_1^\tau) \\
&= K \sum_{\substack{(m',m) \\ c(m',m)=c_t}} \underbrace{\Pr(S_{t-1} = m', S_t = m, Y_1^\tau)}_{\sigma_t(m',m)} \\
&= K \sum_{\substack{(m',m) \\ c(m',m)=c_t}} \alpha_{t-1}(m') \cdot \beta_t(m) \cdot \gamma_t(m',m) \\
&= K \sum_{\substack{(m',m) \\ c(m',m)=c_t}} \alpha_{t-1}(m') \cdot \beta_t(m) \cdot \Pr(u_t = u(m',m)) \cdot \Pr(Y_t \mid c_t = c(m',m)) \\
&= K \cdot \Pr(Y_t \mid c_t = c(m',m)) \sum_{\substack{(m',m) \\ c(m',m)=c_t}} \alpha_{t-1}(m') \cdot \beta_t(m) \cdot \Pr(u_t = u(m',m)) \\
&= K_c \sum_{\substack{(m',m) \\ c(m',m)=c_t}} \alpha_{t-1}(m') \cdot \beta_t(m) \cdot \Pr(u_t = u(m',m)) \qquad \text{(A.7)}
\end{aligned}
$$

Again, due to the definition of the summation, $\Pr(Y_t \mid c_t = c(m',m))$ is constant throughout, and can be taken out of the summation. The normalization constant K_c is chosen such that $\sum_{c_t} \Pr(c_t \mid Y_1^\tau) = 1$.

The MAP algorithm in the SISO module will calculate the updated *a posteriori* probability of the input bits and the codewords using Equations (A.6) and (A.7) respectively. The only inputs required are the *a priori* probabilities, $\Pr(u_t = u(m',m))$, and the channel reliabilities, $\Pr(Y_t \mid c_t = c(m',m))$.

References

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes", *Proc. ICC '93*, Geneva, pp. 1064-1070, May 1993.

[2] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding", *IEEE Transaction on Information Theory*, vol. 44, no. 3, pp. 909-926, May 1998.

[3] J. Hagenauer, E. Offer and L. Papke, "Iterative Decoding for Binary Block and Convolutional Codes", *IEEE Transaction on Information Theory*, vol. 42, no. 2, pp. 429-445, Mar. 1996.

[4] K. Murota and K. Hirade, "GMSK Modulation for Digital Mobile Radio Telephony", *IEEE Transactions on Communications*, vol. COM-29, no. 7, pp. 1044-1050, July 1981.

[5] European Telecommunications Standards Institute (ETSI), "Digital cellular telecommunications system; Modulation (GSM 05.04 version 5.0.1)", ETS 300 959, 1997.

[6] European Telecommunications Standards Institute (ETSI), "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 2: Physical Layer (PHL)", ETS 300 175-2, 1996.

[7] European Telecommunications Standards Institute (ETSI), "Radio Equipment and Systems (RES); HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional specification", ETS 300 652, 1996.

[8] K. Chung and B. Rohani, "Noncoherent detection scheme for GSM system", *IEE Electronics Lettters*, vol. 34, no. 8, pp. 728-730, April 1998.

[9] V. Szeto, "Iterative Decoding of Coded Continuous Phase Modulation", MASc Thesis, Toronto, ON: University of Toronto, 1998.

[10] D. Divsalar and R. McEliece, "On the Design of Concatenated Coding Systems with Interleavers", *The Telecommunications and Mission Operations Progress Report 42-134, April-June 1998*, Jet Propulsion Laboratory, Pasadena, California, August 13, 1998.

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

[12] S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation", *IEEE Communications Mag.*, pp. 14-22, July 1979.

[13] B. Böhm, J. Schoonees and R. Braun, "Data to Frequency Mappings in Various MSK Schemes", *Electronics & Communication Engineering Journal*, pp. 13-20, Feb. 1994.

[14] J. Anderson and C. Sundberg, "Advances in Constant Envelope Coded Modulation", *IEEE Communications Mag.*, pp. 36-45, Dec. 1991.

[15] B. Rimoldi, "A Decomposition Approach to CPM", *IEEE Transaction on Information Theory*, vol. 34, no. 2, pp. 260-270, Mar. 1988.

[16] J. Parsons, *The Mobile Propagation Channel*, London: Pentech Press, 1992.

[17] Y. Iwanami, "Performance of Sequence Estimation Scheme of Narrowband Digital FM Signals with Limiter-Discriminator Detection", *IEEE Journal on Selected Areas in Communications*, vol 13, no. 2, pp. 310-315, Feb. 1995.

[18] J. Fonseka, "Noncoherent Detection with Viterbi Decoding for GMSK Signals", *IEE Proc. Communications*, vol. 143, no. 6, Dec. 1996.

[19] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "A Soft -Input Soft-Output A Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes", *The Telecommunications and Data Acquisition Progress Report 42-127, July-September 1996*, Jet Propulsion Laboratory, Pasadena, California, Nov. 15, 1996.

[20] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Transaction on Information Theory*, pp. 284-287, Mar. 1974.

[21] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications", *Proc. IEEE Globecom '89*, pp. 1680-1686, Nov. 1989.

[22] J. Hagenauer, "Source-Controlled Channel Decoding", *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2449-2457, Sep. 1995.

[23] L. Lin and R. Cheng, "Improvements In SOVA-Based Decoding For Turbo Codes", *Proc. ICC'97*, Montreal, Canada, pp. 1473-1478, June 1997.

[24] M. Fossorier, F. Burkert, S. Lin and J. Hagenauer, "On Equivalence Between SOVA and Max-Log-MAP Decodings", *IEEE Communications Letters*, vol. 2, no. 5, pp. 137-139, May 1998.

[25] R. Pawula, "On the Theory of Error Rates for Narrow-Band Digital FM", *IEEE Transactions on Commnications*, vol. COM-29, no. 11, pp. 1634-1643, Nov. 1981.

[26] R. Pawula, "Refinements to the Theory of Error Rates for Narrow-Band Digital FM", *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 509-513, Apr. 1988.

[27] T. Tjhung, K. Yeo, and P. Wittke, "Effects of Pulse Shaping and Soft Decisions on the Performance of Digital FM with Discriminator Detection", *IEEE Transactions on Communications*, vol. COM-34, no. 11, pp. 1116-1122, Nov. 1986.

[28] M. Peleg, I. Sason, S. Shamai, and A. Elia, "On Interleaved, Differentially Encoded Convolutional Codes", *IEEE Transaction on Information Theory*, vol. 45, no. 7, pp. 2572-2582, Nov. 1999.

[29] J. Woodard and L. Hanzo, "Comparative Study of Turbo Decoding Techniques: An Overview", *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2208-2233, Nov. 2000.

[30] P. Robertson, E. Villebrun, and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain", *Proc. ICC'95*, Seattle, pp. 1009-1013, Jun. 1995.

[31] J. Proakis, *Digital Communications*, 3rd ed., New York: McGraw-Hill, 1995.

[32] D. Asano and S. Pasupathy, "Improved Post-Detection Processing for Limiter-Discriminator Detection of CPM in a Rayleigh, Fast Fading Channel", *IEEE Transactions on Vehicular Technology*, vol. 44, no. 4, pp. 729-734, Nov. 1995.

[33] R. Sánchez-Pérez, F. Casajús-Quirós and S. Pasupathy, "Frequency/Amplitude Detector for Digital FM Data Communication in Multipath Environment", *Proc. IEEE PIMRC'99*, pp. B8-2, Sep. 1999.

[34] R. Sánchez-Pérez, F. Casajús-Quirós and S. Pasupathy, "Robust FM Data Transmission in Multipath Propagation Environment", *Proc. IEEE VTS'99*, pp. 2959-2963, Sep. 1999.

[35] R. Sánchez-Pérez, F. Casajús-Quirós and S. Pasupathy, "Envelope-Aided Viterbi Receivers for GMSK Signals with Limiter-Discriminator Detection", *Manuscript submitted to IEEE Transaction of Communications*, Dec. 1999.