High Performance Non-Symmetric Multi-h CPFSK Modulator and Demodulator Design

by

James Cuthbert

Master of Science (Theoretical Physics) *Cape Town*

Submitted to the Department of Electrical Engineering
in fulfillment of the requirements for the degree of

Doctor of Philosphy

at the

UNIVERSITY OF CAPE TOWN

December 1997

Signature of Author .....
Department of Electrical Engineering
November 17, 1997

Certified by .....
Dr. Robin M. Braun
Director of the Communications Research Group
Thesis Supervisor

Accepted by .....
Prof. Barry J. Downing
Head of Department

Contents

List of Figures

List of Tables

Notation

Signal Description

E_b	Bit Period Energy
T_b	Bit Period
$q(t)$	Frequency Pulse Shape
$g(t)$	Phase Pulse Shape
$s(t, \alpha)$	General Signal Representation

Modulation Indices

h	Modulation Index for Fixed-h CPFSK
h_i	Modulation Indices for Conventional Multi-h CPFSK
$h_i^{x_i}$	Modulation Indices for Non-symmetric Multi-h CPFSK
$p_i^{x_i}$	Numerator of h Associated with Symbol x_i
q	Denominator of h-indices
n	Size of h-set

Frequencies

ω_i	Modulating Frequency
$\omega_{i,s}$	Modulating Frequency Dependent on the Symbol x_s.
Δf_n	Deviating Frequency
Δf_c	Carrier Frequency
Δf_r	Reference Frequency
ω_c	Carrier Frequency
ω_b	Baud Rate

Phase

θ	Phase
$\theta(t, \alpha)$	Information Carrying Phase
ϕ_i	Residual Phase
$\psi(t, \alpha)$	Tilted Phase

Synchronization parameters

τ	Frequency Shift
θ	Phase Shift

Error Correction Decoder Parameters

$G(x)$	Convolutional Code Generator Polynomial
a_i, b_i	Describes Viterbi Error Correction Decoder's State $[m_i]$
x_i	Transmitted Symbol Stream
r_i	Received Symbol Stream
y_i	Output Symbol Stream
n_i	Noise
ν	Inverse Code Rate
l	Path Depth
α_i, β_i	y_i Components
$\hat{x}_i$	Estimate of Data Symbol
$\hat{\phi}_i$	Estimate of Phase
$k(l)$	Threshold Level
p	Transition Probability
Δ	Threshold Spacing
k_T	Truncation Depth

Metrics

ED	Euclidean Distance
$d(m_{i-1}, m_i)$	Branch Metric (Hamming Distance)
$d_{ED}(m_{i-1}, m_i)$	Branch Metric (Euclidean Distance)
$M_j(m_j)$	Partial Path Metric (Hamming Distance)
$M_{ED,j}(m_j)$	Partial Path Metric (Euclidean Distance)
$P_j(m_j)$	Smallest Partial Path Metric
D^2	Intermediate Squared Euclidean Distance
d_i^2	Normalized Intermediate Squared Euclidean Distance
d_{min}^2	Minimum Squared Euclidean Distance
$d(l)$	Accumulated Hamming Distance
$t(l)$	Tilted Accumulated Hamming Distance

Abstract

The continuity properties of the CPFSK signal at it's symbol-period boundaries reveals an inherent memory contained in the transmitted signal. This is utilized as an error correction property. Furthermore, it was shown that the Multi-h CPFSK construction can be accomplished through the combination of a block constructing the memoryless component of the signal and either a block of digital logic circuitry or a continuous phase encoder constructing the memory component. The implementation of the first method was seen to function through simulations performed by using the TESLA simulation package.

An extensive search for good Multi-h CPFSK h-sets was performed. The criteria for determining the performance of these h-sets was the Probability of Error gains over Minimum Shift Keying. The method of search was novel to this work. Specifically, a genetic search algorithm known as the Population Based Incremental Learning algorithm was utilized. The algorithm was implemented through the C++ programming language.

Faster error correction convolutional decoding algorithms were reviewed. Certain decoders exhibit lighter hardware demands, and in specific applications, are less susceptible to erasure problems. The Fano algorithm was selected as the best alternative to the Viterbi algorithm and was modified for the CPFSK implementation. The functionality of the implementation was tested using a C++ simulation.

Various structures used to implement the synchronization and demodulation of Multi-h CPFSK were investigated. The most comprehensive structure that could be found was

a scheme developed by Premji and Taylor using maximum likelihood techniques. This scheme was selected as it can be easily modified for the use with the large state, high speed implementation of non-symmetric Multi-h CPFSK investigated in this thesis.

The PBIL algorithm was found to be an efficient method for finding good h-sets with large numbers of phase states. Theoretical gains over MSK using this method were found to be significant. It was concluded that the Fano decoder is highly applicable to the demodulator structure proposed in this thesis and is a preferred alternative to the Viterbi decoder under specific circumstances.

Chapter 1

Introduction

The number of applications requiring digital communications are ever increasing. From computer communications with BBS's to the exponentially expanding internet. From tone/pulse dialling to cordless cell based digital communications, analogue television to Digital High Definition Television ... the list continues. This places a demand on technology to provide efficient and reliable schemes of digital communication. The work undertaken here attempts to contribute to this goal by providing a greater understanding of an already efficient modulation scheme and a further enhancement of it's performance.

Constant-envelope continuous-phase signal modulation (CPM) has received much attention in applications of digital communications over satellite and terrestrial radio channels [1, 2]. The reason is that CPM is a power and bandwidth efficient scheme and shows resistance to fading and nonlinear distortions [3].

Chapter 2 is a general introduction to Multi-h CPFSK. Chapter 3 describes two modulator design approaches. Both schemes follow an approach that decomposes Multi-h CPFSK into components that are used in conjunction with each other to construct the final signal transmitted. The data-dependent terms and data-independent terms become clearly identifiable. Chapter 4 details three different error correction algorithms. Two of these algorithms are sub-optimal, while the other is a well known optimal error correction scheme. Since these schemes were originally designed for use with convo-

lutional codes, they are illustrated with the use of *Hamming Distances* here. The differences between the *Euclidean Distance* (ED) and the *Hamming Distance* approach is explained. Chapter 5 presents the fundamental principles of certain demodulator design methods. A comprehensive scheme developed by Premji and Taylor [4, 5] is also presented. It forms the basis of the demodulation scheme proposed in this work. Chapter 6 presents a novel method of selecting design parameters. It is presented together with established methods of measuring performance and with results found using these methods. h-sets are selected so as to perform well with a sub-optimal error correction method. Chapter 7 presents conclusions and possible future developments.

Chapter 2

Multi-h CPFSK

2.1 Elements of a Digital Communications System

An overview of a general communications system is presented in figure 2-1.

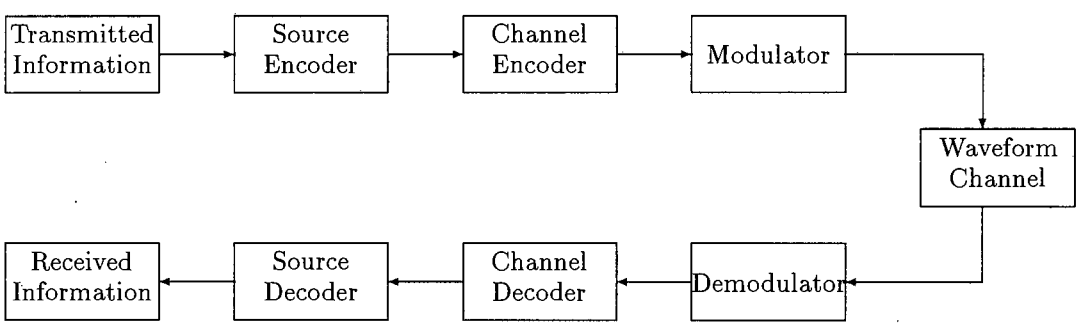

Figure 2-1: Block Diagram of a General Communications System

The *Source Encoder* is responsible for mapping the original symbol stream emitted from the *Transmitted Information* block into a form in which redundancy has been removed. Redundancy is introduced via the *Channel Encoder* to enhance the performance of the transmission over the *Waveform Channel*. The encoded information stream modulates the carrier waveform by means of the *Modulator*. The modulated waveform is then broadcast over the *Waveform Channel*. The channel intrinsically adds a noise component to the transmitted signal. The *Demodulator* estimates the most likely encoded

information that would produce the waveform actually received. The demodulated data stream is then decoded by the *Channel Decoder* and the *Source Decoder*. The *Channel Decoder* may incorporate error correction methods that make use of the redundancy introduced by the channel encoder.

The primary concern in this thesis is the transmission of a sequence of *binary* symbols. However this work is readily generalized to an M-level system. The digital source is considered to be a memoryless source of symbols that are fed to the encoder at a constant rate known as the *Baud Rate*. The channel encoder outputs a constant voltage of only two possible amplitudes every symbol period. This voltage sets the signal frequency. This modulation scheme is generally known as **C**ontinuous **P**hase **F**requency **S**hift-**K**eying (*CPFSK*).

2.2 Continuous Phase Frequency Shift Keying (CPFSK)

CPM can be represented by

$$s(t,\alpha) = \sqrt{\frac{2E_b}{T_b}}\cos(\omega_c t + \theta(t,\alpha) + \theta_0) \tag{2.1}$$

where the ω_c is the carrier frequency, θ_0 is an arbitrary initial phase, E_b is the symbol energy and T_b is the symbol period. θ_0 can be set to zero without lose of generality.

The information carrying phase is

$$\theta(t,\alpha) = 2\pi h \int_{-\infty}^{t} \sum_{i=-\infty}^{\infty} x_i g(\tau - iT)d\tau \qquad \text{where} -\infty < t < \infty. \tag{2.2}$$

α is the stream of M-ary data symbols x_i which can take on the values

$$\begin{aligned} x_i &= \pm 1, \pm 3, \pm 5, ..., \pm(M-1) \quad \text{if M is even} \\ x_i &= 0, \pm 2, \pm 4, ..., \pm(M-1) \qquad \text{if M is odd.} \end{aligned} \tag{2.3}$$

h is the modulation index and is defined as the ratio of the frequency difference between the two signaling frequencies corresponding to $|x_i| = 1$ and the symbol rate. The

amplitude of the baseband frequency pulse $g(t)$ is chosen to give the maximum phase change $|x_i|h\pi$, over each symbol period.

Following Aulin and Sundberg [6], the baseband phase pulse is defined as

$$q(t) = \int_{-\infty}^{t} g(\tau)d\tau \quad \text{where} - \infty < t < \infty, \tag{2.4}$$

the phase can be written in terms of $q(t)$ as

$$\theta(t,\alpha) = 2\pi h \sum_{i=-\infty}^{\infty} x_i q(t - iT_b) \quad \text{where} - \infty < t < \infty. \tag{2.5}$$

The frequency pulse is defined as

$$\begin{aligned} g(t) &= \text{constant} \neq 0 \quad \text{where} \ \ 0 \leq t \leq T \\ g(t) &= 0 \quad \text{elsewhere} \end{aligned} \tag{2.6}$$

and normalizing condition for the frequency pulse $g(\tau)$ is expressed as $q(T_b) = 1/2$. The CPFSK modulation schemes are a subclass of CPM where the instantaneous frequency is constant over each bit period. The CPFSK phase pulse is piecewise linear,

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

This information carrying phase $\theta(t,\alpha)$ can follow any of the phase trajectories illustrated in figure 2-2.

The conventional means of generating this type of CPFSK is accomplished through the modulator structure illustrated in figure 2-3. An impulse generator is fed by the incoming symbol stream producing impulses that are evenly spaced at intervals of the symbol period T_b. These impulses are shaped with the phase pulse shape response specified by $q(t)$. The result is scaled according to the modulation index and represents the phase information which is modulated to produce the transmitted signal.

The physical device that is capable of accomplishing much of the above process is

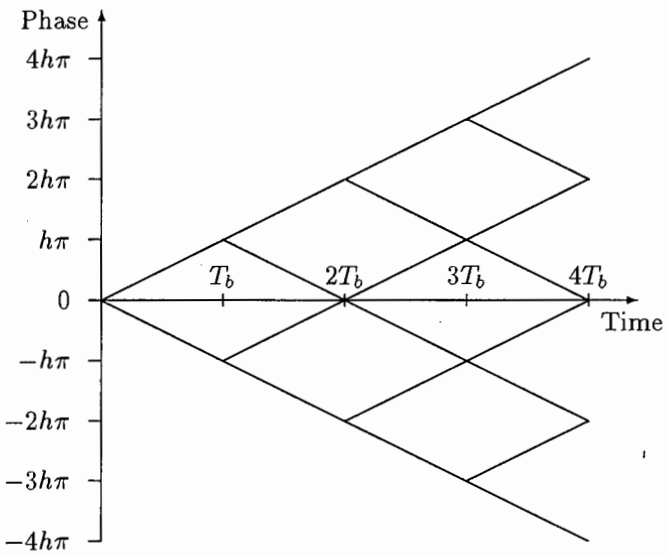

Figure 2-2: CPFSK Phase Trajectories

known as the *Voltage Controlled Oscillator (VCO)*. Essentially the gain factor, the input levels specifying the modulation index h and the curvature of the voltage input levels must be handled externally.

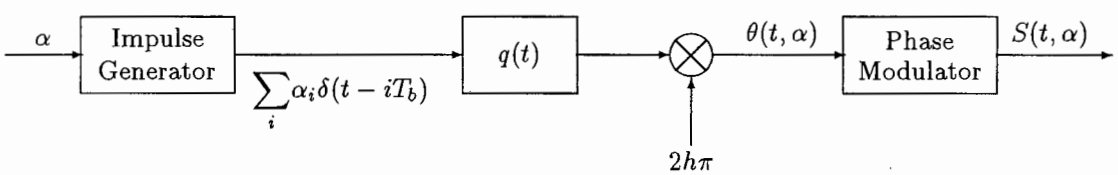

Figure 2-3: CPFSK Modulator Structure

2.3 Multi-h CPFSK

Multi-h CPFSK is a generalization of CPFSK as first described by Anderson and deBuda [36]. The use of the modulation index is broadened by changing the index once every bit period. Essentially h is replaced by h_i allowing the information carrying phase $\theta(t, \alpha)$ to be replaced by

$$\theta(t, \alpha) = 2\pi \int_{-\infty}^{t} \sum_{i=-\infty}^{\infty} x_i h_i g(\tau - iT) d\tau \quad \text{where} - \infty < t < \infty. \tag{2.8}$$

It is practical to restrict h_i to be of the form $h_i = p_i/q$ where p_i and q are integers. The restriction $p_i \leq q$ conserves power spectrum bandwidth. To keep the system manageable, h_i's are selected cyclically from a set of $h's$ ie.

$$h_i \in \{h_1, h_2, h_3, ..., h_n\} \equiv H_n. \tag{2.9}$$

The phase trellis for the h-set, $H_2 = \{2/4, 3/4\}$, is illustrated in figure 2-4.

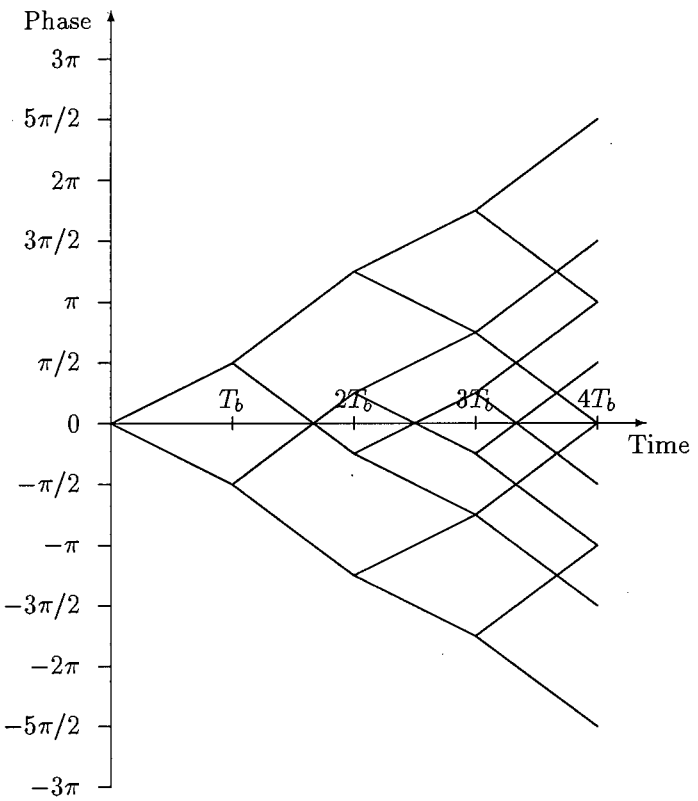

Figure 2-4: The $\{2/4, 3/4\}$ Multi-h Phase Trajectories

The information carrying phase $\theta(t, \alpha)$, can be decomposed into an accumulated residual phase ϕ_i and a time varying part, $x_i\omega_i(t - iT_b)$.

The time varying phase is associated with the *current* symbol being transmitted (in the symbol-period commencing at time $t = iT_b$) and can be expressed as [4]

$$x_i\omega_i(t - iT_b) = x_i \int_{iT_b}^{t} \frac{\pi h_i}{T_b} g(\tau - iT_b)d\tau. \tag{2.10}$$

The residual phase ϕ_i is the accumulated phase up to the commencement of the *current* bit-period and is expressed as

$$\phi_i = \sum_{k=0}^{i-1} \pi x_k h_k. \tag{2.11}$$

The complete bit-period signal representation is

$$s_i(t, x_i) = \sqrt{\frac{2E_b}{T_b}} \cos(\omega_c t + x_i \omega_i (t - iT_b) + \phi_i) \tag{2.12}$$

where $iT_b < t < (i+1)T_b$.

Using eq. 2.10 and eq. 2.7, the time-varying phase component is

$$x_i \omega_i (t - iT_b) = x_i \frac{\pi h_i}{T_b} \cdot (t - iT_b) \text{ where } iT_b < t < (i+1)T_b. \tag{2.13}$$

2.3.1 Non-Symmetric Multi-h CPFSK

Conventional Multi-h CPFSK can be generalized further. Hwang, Lee and Chen [26] introduced the idea of asymetric h-index sets. Taking this approach further, the h-index sets can be chosen to be non-symmetric.

In coventional Multi-h CPFSK (symmetric Multi-h CPFSK), the choice of h_i is dependent solely on the bit-period iT_b and has no dependence on the symbol transmitted. Non-symmetric Multi-h CPFSK utilizes the generalized modulation index, ${h_i}^{x_i}$; where the possibility that ${h_i}^{-1} \neq {h_i}^{+1}$ is feasible. ${h_i}^{x_i}$ belongs to the n cyclic set

$$h_i^{x_i} \in \{h_1^{x_1}, h_2^{x_2}, h_3^{x_3}, ..., h_n^{x_n}\}, \tag{2.14}$$

$$h_{kn+j}^{x_{kn+j}} = p_j^{x_j}/q \qquad \text{where} \quad 0 \leq j \leq n-1, \quad k = 0, 1, 2, ... \tag{2.15}$$

This alters the phase components as follows. The time varying phase in the symbol-period commencing at time $t = iT_b$ becomes

$$x_i \omega_i (t - iT_b) = x_i \int_{iT_b}^{t} \frac{\pi h_i^{x_i}}{T_b} g(\tau - iT_b) d\tau, \tag{2.16}$$

and the residual phase ϕ_i becomes

$$\phi_i = \sum_{k=0}^{i-1} \pi x_k h_k^{x_k}. \tag{2.17}$$

Using eq. 2.16 and eq. 2.7, the time-varying phase component is

$$x_i \omega_i (t - iT_b) = x_i \frac{\pi h_i^{x_i}}{T_b} \cdot (t - iT_b) \text{ where } iT_b < t < (i+1)T_b. \tag{2.18}$$

Chapter 3

Multi-h CPFSK Modulation

Multi-h CPFSK can be constructed used various methods, the simplest construction being that of using a *Voltage Controlled Oscillator* (VCO). The VCO inherently maintains phase continuity, a property that exhibits the memory content of the modulation scheme. This approach, however, does not extend understanding of the memory components or offer a way forward to enhancing coding gains. In this chapter, two decomposed methods of construction are presented.

Firstly a 'blind' approach is taken. Digital logic is designed to select a particular oscillator from a bank of oscillators for each successive symbol period, in a manner that will produce a Multi-h CPFSK signal when combined.

The second approach is a more intelligent approach. This method seeks to extract a memoryless component from a given fixed-h CPFSK signal. Coupling this approach to a mapper will enable this scheme to produce Multi-h CPFSK.

3.1 An Oscillator Approach to Modulation

Crawford [7] approached the construction of Multi-h CPFSK through the use of the continuity conditions at symbol-period boundaries. This scheme was inspired by the Massey [31] transducer for MSK. The proposed implementation of this modulation

scheme is illustrated in figure 3-1 using a 2-h scheme, namely the set $H_2 = \{h_1, h_2\}$ and four possible phases for each signalling frequency. The set $\{1/4, 3/4\}$ belongs to this group of H_2 with the phases for each oscillator coming from the set $\{0, \pi/2, \pi, 3\pi/2\}$.

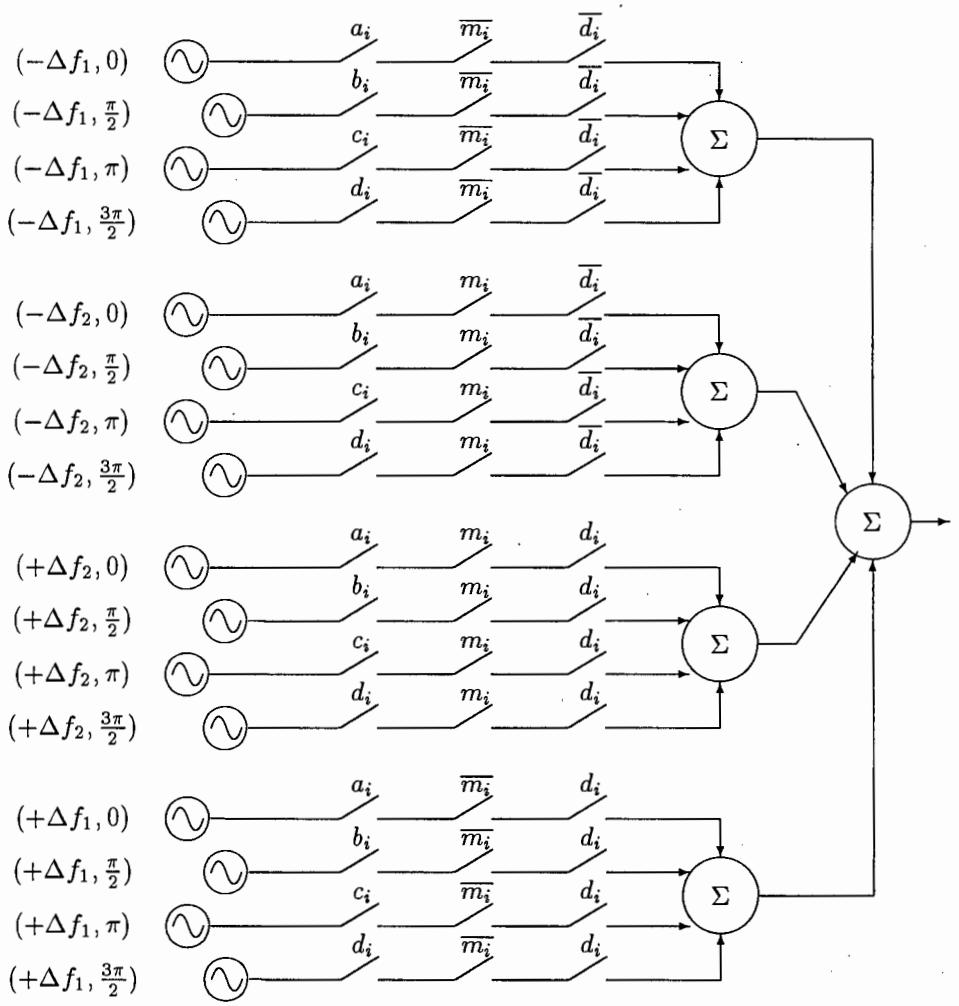

Figure 3-1: Switching Scheme for a 2-h Multi-h Modulator

In the figure, the notation $(\pm\Delta f_n, \theta)$ denotes that the oscillator is producing a sinusoidal output of frequency $f_c \pm \Delta f_n$ commencing with the phase θ at time $t = 0$. The oscillators are chosen by using a combination of three switches. a_i, b_i, c_i, d_i are the outputs of the decision logic which are directly related to the starting phase of **all** the oscillators. m_i is high if the transmission frequency is $f_c \pm \Delta f_2$ in the i^{th} bit-period and low if the transmission frequency is $f_c \pm \Delta f_1$. m_i is related directly to the superbaud clock. d_i and $\overline{d_i}$ are high if the binary bit transmitted is high and low respectively.

The possible starting phases that are used at each signal frequency can be found by looking for phase shifts as perceived by a particular signal frequency oscillator. The contribution to phase is gained and lost by transmissions in bit-periods in which another signalling frequency was used. The rate at which the phase is contributing to residual phase is proportional to the frequency difference between the signals.

Illustrating this for a 2-h system, the focus is placed on a reference frequency f_r which is chosen from the set of signalling frequencies $\{f_c - \frac{h_2}{2T_b}, f_c - \frac{h_1}{2T_b}, f_c + \frac{h_1}{2T_b}, f_c + \frac{h_2}{2T_b}\}$. If the frequency used in the ith bit-period is denoted by f_i, the accumulated phase contribution to a particular reference frequency over $t = lT_b$ will be

$$\mathrm{mod}_{2\pi}\left[\sum_{i=1}^{l} 2\pi(f_i - f_r)T_b\right] = \mathrm{mod}_{2\pi}\left[\sum_{i=1}^{l} \pi(h_i - h_r)\right]. \tag{3.1}$$

where $\mathrm{mod}_{2\pi}$ is the modulo 2π operator. A one to one relationship exists between h_i, h_r and f_i, f_r respectively.

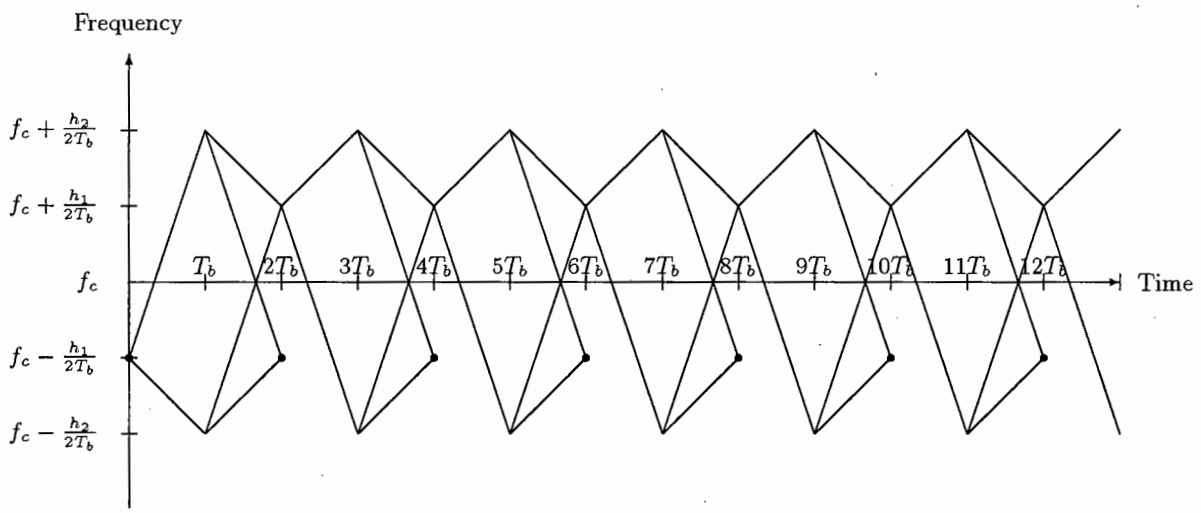

Figure 3-2: Possible Deviations from the Frequency $(f_c - h_1/2T_b)$

As an illustration, $f_r = f_c - h_1/2T_b$ is chosen from the above set and the phase paths responsible for contributing to the residual phase are illustrated in figure 3-2. Contribution to the residual phase will be gained from paths emanating from the first dark node via a path to a later dark node.

The possible phases are listed in the table 3.1 for a selected sets belonging to H_2.

Frequency	Possible Phases for Different Sets of $\{h_1, h_2\}$			
	{1/3,2/3}	{1/4,2/4}	{1/4,3/4}	{1/5,2/5}
$f_c + \frac{h_2}{2T_b}$	$0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}$	$0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}$	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	$0, \frac{\pi}{5}, \frac{2\pi}{5}, \frac{3\pi}{5}, \frac{4\pi}{5}, \pi, \frac{6\pi}{5}, \frac{7\pi}{5}, \frac{8\pi}{5}, \frac{9\pi}{5}$
$f_c + \frac{h_1}{2T_b}$	$0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}$	$0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}$	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	$0, \frac{\pi}{5}, \frac{2\pi}{5}, \frac{3\pi}{5}, \frac{4\pi}{5}, \pi, \frac{6\pi}{5}, \frac{7\pi}{5}, \frac{8\pi}{5}, \frac{9\pi}{5}$
$f_c - \frac{h_1}{2T_b}$	$0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}$	$0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}$	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	$0, \frac{\pi}{5}, \frac{2\pi}{5}, \frac{3\pi}{5}, \frac{4\pi}{5}, \pi, \frac{6\pi}{5}, \frac{7\pi}{5}, \frac{8\pi}{5}, \frac{9\pi}{5}$
$f_c - \frac{h_2}{2T_b}$	$0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}$	$0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}$	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	$0, \frac{\pi}{5}, \frac{2\pi}{5}, \frac{3\pi}{5}, \frac{4\pi}{5}, \pi, \frac{6\pi}{5}, \frac{7\pi}{5}, \frac{8\pi}{5}, \frac{9\pi}{5}$

Table 3.1 : Phase States of Selected 2-h Schemes

3.1.1 Phase Continuity Conditions

a_i, b_i, c_i, d_i are determined from *Signal Referenced* phase trellises. The *Signal Referenced* phase trellises are derived through continuous phase conditions, by observing the change in the phase, when switching between two signals. 2-h Multi-h signal referenced phase trellises are constructed in the following manner.

In 2-h Multi-h there are four possible symbol streams that contain the mappings of all possible transitions on a trellis. All possible phase differences are found by considering the transmissions 111111... , 0101010... , 101010... and 000000... .

When a 1 is sent in the time interval $nT_b < t < (n+1)T_b$, the signal transmitted is

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} \cos(\omega_c t + \frac{\pi h_1 t}{T_b} + \phi_1(nT_b)) \tag{3.2}$$

if n is even or

$$s_2(t) = \sqrt{\frac{2E_b}{T_b}} \cos(\omega_c t + \frac{\pi h_2 t}{T_b} + \phi_2(nT_b)) \tag{3.3}$$

if n is odd. $\phi_i(nT_b)$ is the starting phase required to maintain phase continuity when the signal $s_i(t)$ is transmitted in the stated time interval. Similarly when a 0 is sent in the time interval $nT_b < t < (n+1)T_b$, the signal transmitted is

$$s_3(t) = \sqrt{\frac{2E_b}{T_b}} \cos(\omega_c t - \frac{\pi h_1 t}{T_b} + \phi_3(nT_b)) \tag{3.4}$$

if n is even or

$$s_4(t) = \sqrt{\frac{2E_b}{T_b}} \cos(\omega_c t - \frac{\pi h_2 t}{T_b} + \phi_4(nT_b)) \tag{3.5}$$

if n is odd.

Case 1: Consider the transmission of a signal sequence of 1's only. This is accomplished by simply transmitting signals $s_1(t)$ every even bit period and $s_2(t)$ every other period. The phase is constrained by the condition that the switching between the two signals at bit period boundaries remain continuous and smooth. In other words

$$s_1(nT_b) = s_2(nT_b) \tag{3.6}$$

$$\frac{ds_1(t)}{dt}\bigg|_{t=nT_b} = \frac{ds_2(t)}{dt}\bigg|_{t=nT_b} \tag{3.7}$$

where n is an integer. This results in

$$\phi_2(nT_b) - \phi_1(nT_b) = \pi n(h_1 - h_2). \tag{3.8}$$

Case 2: Consider the transmission of a signal sequence of 0's only. This is accomplished by simply transmitting signals $s_3(t)$ every even bit period and $s_4(t)$ every other period. The phase continuity condition at the boundary $t = nT_b$ is now

$$s_3(nT_b) = s_4(nT_b) \tag{3.9}$$

$$\frac{ds_3(t)}{dt}\bigg|_{t=nT_b} = \frac{ds_4(t)}{dt}\bigg|_{t=nT_b} \tag{3.10}$$

and results in

$$\phi_3(nT_b) - \phi_4(nT_b) = \pi n(h_1 - h_2). \tag{3.11}$$

Case 3: Consider the transmission of a signal sequence of alternating 0's and 1's and commencing with 0. This is accomplished by simply transmitting signals $s_3(t)$ every even bit period and $s_2(t)$ every other period. The phase continuity condition at the boundary $t = nT_b$ is now

$$s_3(nT_b) = s_2(nT_b) \tag{3.12}$$

$$\frac{ds_3(t)}{dt}\bigg|_{t=nT_b} = \frac{ds_2(t)}{dt}\bigg|_{t=nT_b} \tag{3.13}$$

and results in

$$\phi_2(nT_b) - \phi_3(nT_b) = -\pi n(h_1 + h_2). \tag{3.14}$$

Case 4: Consider the transmission of a signal sequence of alternating 0's and 1's and commencing with 1. This is accomplished by simply transmitting signals $s_1(t)$ every even bit period and $s_4(t)$ every other period. The phase continuity condition at the boundary $t = nT_b$ is now

$$s_1(nT_b) = s_4(nT_b) \tag{3.15}$$

$$\frac{ds_1(t)}{dt}\bigg|_{t=nT_b} = \frac{ds_4(t)}{dt}\bigg|_{t=nT_b} \tag{3.16}$$

and results in

$$\phi_4(nT_b) - \phi_1(nT_b) = \pi n(h_1 + h_2). \tag{3.17}$$

Observing eq.'s 3.8, 3.11, 3.14 and 3.17, it is clear that the phase difference is dependent on n. The phase differences are presented in table 3.2 together with two specific examples, namely $\{3/4, 1/4\}$ and $\{1/3, 2/3\}$.

Bit-Period Starting Time			Use eq.	Phase	Example $\{h_1, h_2\}$	
$(2n-1)T_b$	$2nT_b$	$(2n+1)T_b$		Difference	$\{3/4, 1/4\}$	$\{1/3, 2/3\}$
1	1		3.8	$-2\pi n(h_1 - h_2)$	$-\pi n$	$\frac{2\pi}{3}n$
1	0		3.17	$-2\pi n(h_1 + h_2)$	$-2\pi n$	$-2\pi n$
0	1		3.14	$2\pi n(h_1 + h_2)$	$2\pi n$	$2\pi n$
0	0		3.11	$2\pi n(h_1 - h_2)$	πn	$-\frac{2\pi}{3}n$
	1	1	3.8	$\pi(2n+1)(h_1 - h_2)$	$(2n+1)\frac{\pi}{2}$	$-\frac{\pi}{3}(2n+1)$
	1	0	3.14	$-\pi(2n+1)(h_1 + h_2)$	$-(2n+1)\pi$	$-\pi(2n+1)$
	0	1	3.17	$\pi(2n+1)(h_1 + h_2)$	$(2n+1)\pi$	$\pi(2n+1)$
	0	0	3.11	$-\pi(2n+1)(h_1 - h_2)$	$-(2n+1)\frac{\pi}{2}$	$-\frac{\pi}{3}(2n+1)$

Table 3.2 : Phase Difference Results for Phase Continuity at $t = nT_b$

The phase trellises are constructed for $\{3/4, 1/4\}$ Multi-h in figure 3-3 and $\{1/3, 2/3\}$ Multi-h in figure 3-4. The notation $\frac{1}{0}, \frac{1}{0}$ indicates the bit before the comma was transmitted in the period $(n-1)T_b < t < nT_b$ and the bit after the comma was transmitted in the period $nT_b < t < (n+1)T_b$.

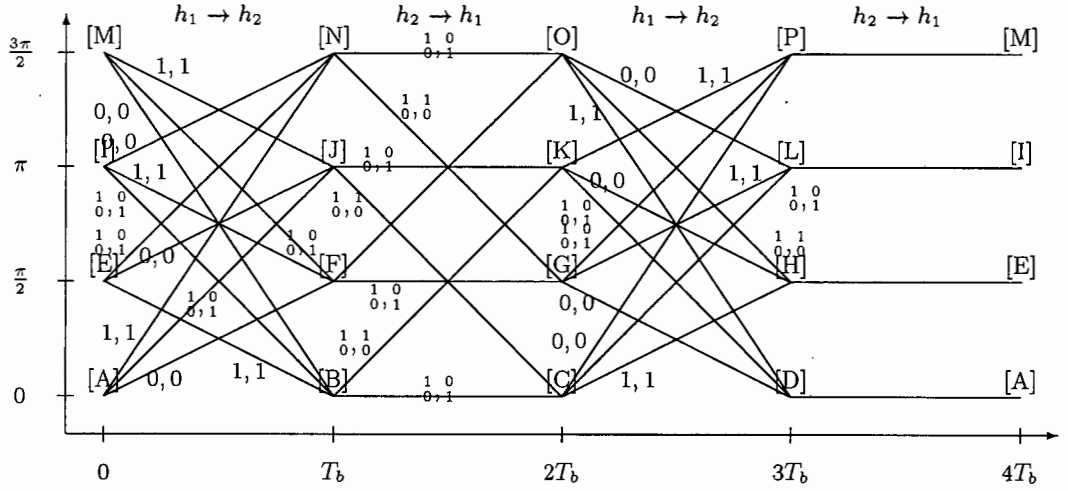

Figure 3-3: Signal Referenced Phase Trellis of $\{3/4, 1/4\}$ Multi-h Scheme

3.1.2 Transducer Based Modulator Construction

The $\{3/4, 1/4\}$ trellis (figure 3-3) repeats itself every fourth bit period, whereas the $\{1/3, 2/3\}$ trellis repeats itself every sixth bit period. Generally a 2-h Multi-h trellis repeats itself after $2qT_b$ if the sum of all p_i, i ranging over the full h-set size, is odd otherwise after qT_b.

This means that in the $\{3/4, 1/4\}$ scheme, sixteen unique nodes can be identified. These are labelled from [A] to [P] in figure 3-3. Translating the trellis into a set of Boolean algebra equations representing the possibility of transitions between these nodes, the digital logic required to develop a phase encoder can be determined. The set of Boolean equations are

$$
\begin{aligned}
A &= m_i \cdot D \cdot z^{-1} \\
B &= \overline{m_i} \cdot (E \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d) + I \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + M \cdot z^{-1} \cdot (\overline{d} \cdot z^{-1} \cdot \overline{d})) \\
C &= m_i \cdot (B \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + J \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d + \overline{d} \cdot z^{-1} \cdot \overline{d})) \\
D &= \overline{m_i} \cdot (G \cdot z^{-1} \cdot (\overline{d} \cdot z^{-1} \cdot \overline{d}) + K \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + O \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d)) \\
E &= m_i \cdot H \cdot z^{-1} \\
F &= \overline{m_i} \cdot (I \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d) + M \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + A \cdot z^{-1} \cdot (\overline{d} \cdot z^{-1} \cdot \overline{d})) \\
G &= m_i \cdot (F \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + N \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d + \overline{d} \cdot z^{-1} \cdot \overline{d})) \\
H &= \overline{m_i} \cdot (K \cdot z^{-1} \cdot (\overline{d} \cdot z^{-1} \cdot \overline{d}) + O \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + C \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d)) \\
I &= m_i \cdot L \cdot z^{-1}
\end{aligned}
$$

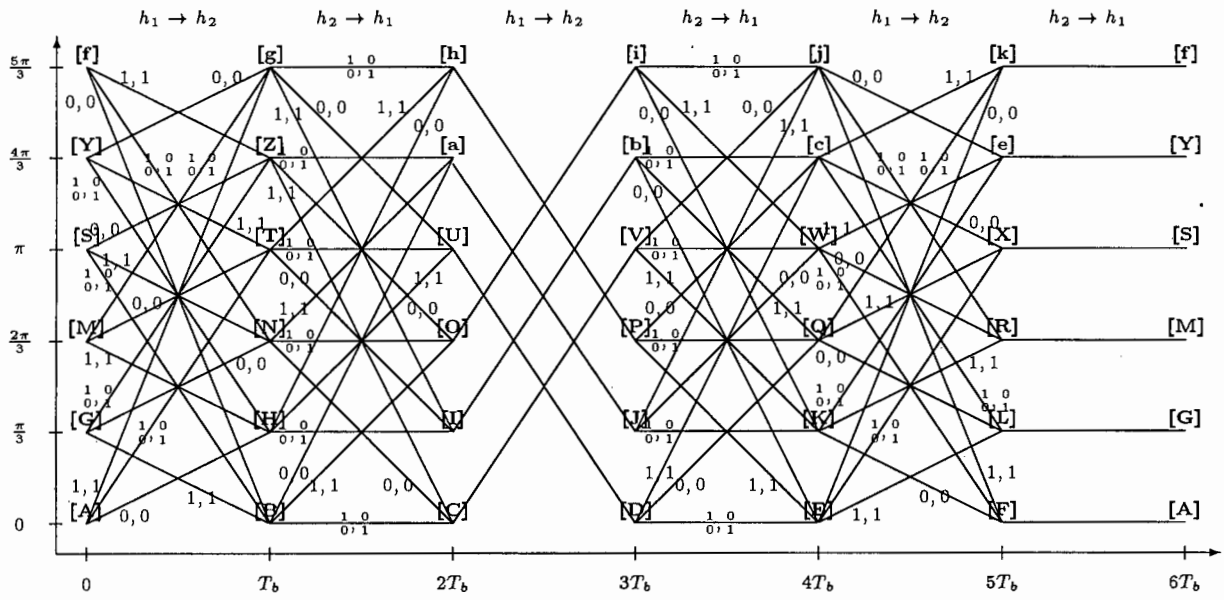

Figure 3-4: Signal Referenced Phase Trellis of $\{1/3, 2/3\}$ Multi-h Scheme

$$
\begin{aligned}
J &= \overline{m_i} \cdot (M \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d) + A \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + E \cdot z^{-1} \cdot (\overline{d} \cdot z^{-1} \cdot \overline{d})) \\
K &= m_i \cdot (J \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + B \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d + \overline{d} \cdot z^{-1} \cdot \overline{d})) \\
L &= \overline{m_i} \cdot (O \cdot z^{-1} \cdot (\overline{d} \cdot z^{-1} \cdot \overline{d}) + C \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + G \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d)) \\
M &= m_i \cdot P \cdot z^{-1} \\
N &= \overline{m_i} \cdot (A \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d) + E \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + I \cdot z^{-1} \cdot (\overline{d} \cdot z^{-1} \cdot \overline{d})) \\
O &= m_i \cdot (N \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + F \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d + \overline{d} \cdot z^{-1} \cdot \overline{d})) \\
P &= \overline{m_i} \cdot (C \cdot z^{-1} \cdot (\overline{d} \cdot z^{-1} \cdot \overline{d}) + G \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot \overline{d} + \overline{d} \cdot z^{-1} \cdot d) + K \cdot z^{-1} \cdot (d \cdot z^{-1} \cdot d)). \quad (3.18)
\end{aligned}
$$

The notation used in the Boolean algebra is as follows. m_i is high if the transition from the states on the right hand side of the equation to the left hand side used a modulation transition of $h_2 \to h_1$. The states on the right hand side were transmitted with the modulation index h_2 in other words. $\overline{m_i}$ represents the reverse h transition. d and $\overline{d}$ are high when the digit symbol is high and low respectively. z^{-1} is the delay operator.

The construction of these equations is best explained through an example. It is seen from the trellis that the state [B] is reached from any of the states [E],[I] or [M]. By convention or's are represented by '+' and this can be seen by the three distinct terms

separated by '+' 's on the right hand side of the equation $B = ...$ above.

The condition to reach [B] from [E] is that $h_1 \rightarrow h_2$ ($\overline{m_i}$), [B] is one bit-period later than [E] (z^{-1}) and the previous and present bit being transmitted were both highs ($d.z^{-1}.d$). These three conditions are all ands (.), $\overline{m_i}$ being a common condition to all three states on the right hand side of the equation.

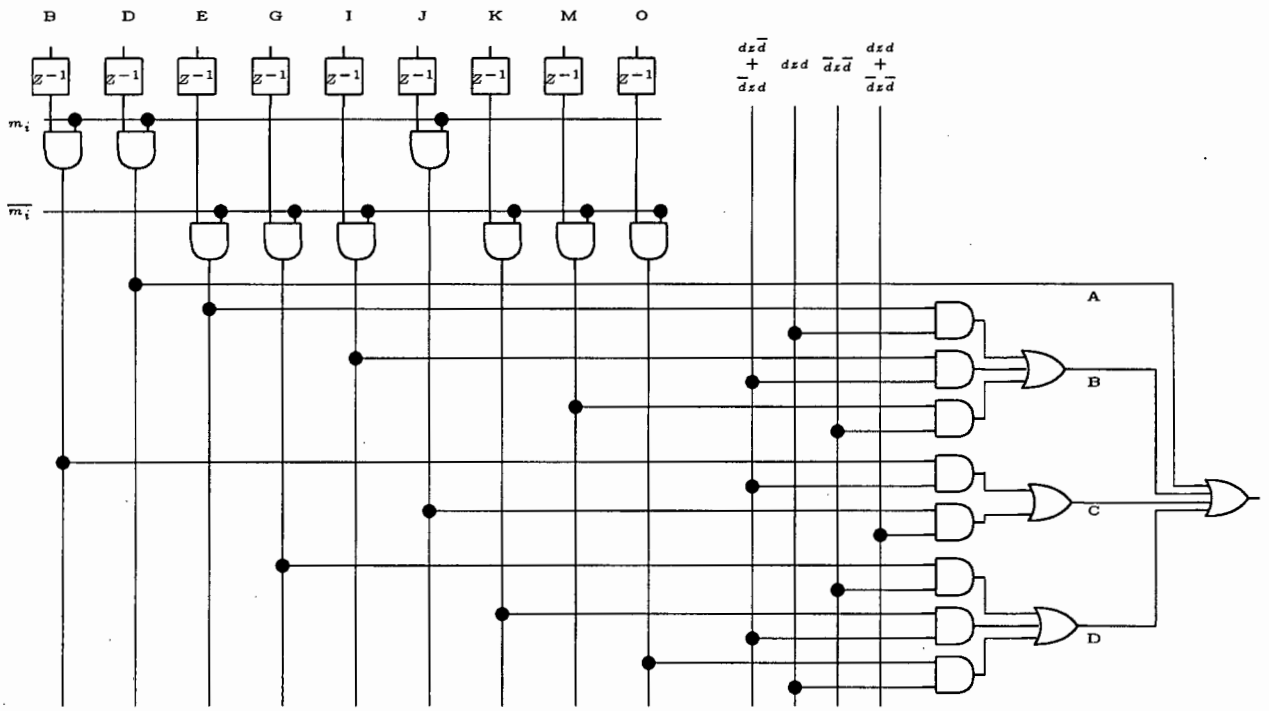

Figure 3-5: A Section of the $\{3/4, 1/4\}$ Multi-h Modulator Phase Encoder

The implementation is illustrated through the partial modulator phase encoder illustrated in figure 3-5. This illustrates how the first four equations, $A = ..., B = ..., C = ...$ and $D = ...$, are implemented.

Such an approach to Multi-h is not practical due to a number of factors. Amongst these factors are the complex phase initializations of the independent oscillators, the excessive hardware demands of the numerous accurate delays used for establishing the correct oscillator phase, the large number of oscillators actually required and the large digital logic circuitry. This approach, nevertheless, serves to illustrate the basic construction of Multi-h.

3.2 Rimoldi's Decomposition Approach to CPM

Rimoldi [8, 3] decomposed the CPM modulator into a Continuous Phase Encoder (*CPE*) and a Memoryless Modulator (*MM*) as illustrated in figure 3-6.

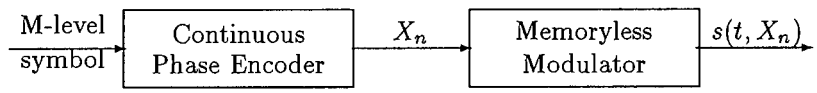

Figure 3-6: Rimoldi's Decomposition of CPM

Throughout this work, it is convenient to use an operator which returns the *physical phase,* the phase that is actually measured, from the actual phase. Essentially this is just a *modulo* 2π operator. Rimoldi introduced the notation

$$R_{2\pi}(\theta) = \theta - \left[\frac{\theta}{2\pi}\right] 2\pi \tag{3.19}$$

where [ ] denotes the largest integer not exceeding the enclosed number.

Instead of referencing the phase with the carrier, Rimoldi referenced the phase with the lowest signal frequency. This is advantageous over the previous approach, that references phase with each individual signal frequency, as phase referencing is dependent on only *one* frequency. This diminishes phase extraction complexity.

The time-invariant phase trellis is obtained through incorporating a phase shift into the previous phase. The *tilted trellis* phase is given by,

$$\psi(t, \alpha) = \theta(t, \alpha) + \pi h(M - 1)t/T_b. \tag{3.20}$$

A general CPM phase trellis is tilted upwards as illustrated in figure 3-7.

As h has been chosen to be rational (p/q), applying the operator $R_{2\pi}[\,]$ to the phase trellis in figure 3-7 will yield a *physical* phase trellis that repeats itself after every symbol period. The *physical* trellis is said to be time invariant and thus can be implemented with a *memoryless modulator(MM).* The *Continuous Phase Encoder (CPE)* will choose

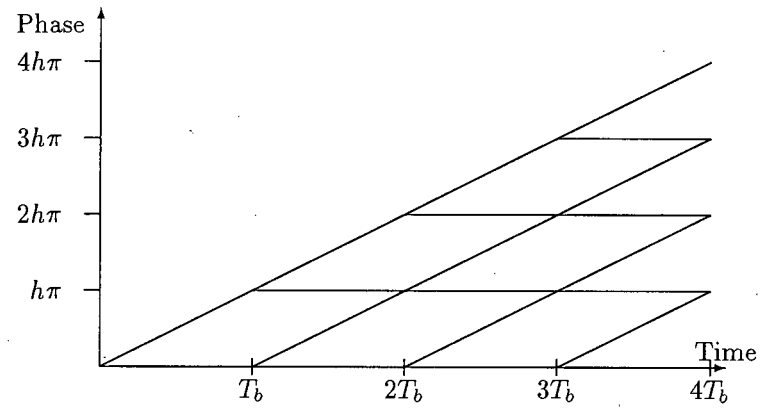

Figure 3-7: Tilted Phase Trellis

the exact trellis path that is followed for a given symbol stream.

The signal can be written as

$$s(t + nT_b, X_n) = \sqrt{\frac{2E_b}{T_b}} \cos(\omega_1(t + nT_b) + \overline{\psi(t, X_n)} + \phi_0) \quad 0 \leq t \leq T_b \tag{3.21}$$

where $\omega_1 = \omega_c - (M-1)\omega_b/q$ is the reference frequency. $\overline{\psi(t, X_n)}$ is the physical realization of $\psi(t, X_n)$. X_n is directly related to the symbol stream transmitted up to the n'th symbol period, and ϕ_0 is the initial phase. The signal can be decomposed into the in-phase and quadrature components using trigonometric identities allowing for the memoryless part to be extracted. The signal representation can be rewritten as

$$s(t, X_n) = I(t, X_n)\phi_I(t) + Q(t, X_n)\phi_Q(t) \tag{3.22}$$

where

$$I(t, X_n) = \sqrt{\frac{E_b}{T_b}} \cos(\overline{\psi}(t, X_n)) \tag{3.23}$$

$$Q(t, X_n) = \sqrt{\frac{E_b}{T_b}} \sin(\overline{\psi}(t, X_n)) \tag{3.24}$$

and

$$\phi_I(t) = \sqrt{2} \cos(\omega_1(t + nT_b) + \phi_0) \tag{3.25}$$

$$\phi_Q(t) \;=\; \sqrt{2}\sin(\omega_1(t+nT_b)+\phi_0). \tag{3.26}$$

The task of determining X_n is delegated to the Continuous Phase encoder which in turn informs the $I(t, X_n)$ and $Q(t, X_n)$ components. The components $\phi_I(t)$ and $\phi_Q(t)$ compose the memoryless modulator stage, as both of these are independent from symbol stream term X_n and are solely dependent on the frequency ω_1.

Recall that $\theta(t, \alpha)$ can be decomposed (eq.'s 2.16, 2.17). Incorporating the phase shift allows the tilted phase to be written as

$$\psi(t,\alpha) = x_i\omega_i(t-iT_b)+\theta_i+\pi(M-1)h_it/T_b. \tag{3.27}$$

The signal is specified by the same data dependent term $x_i\omega_i(t-iT_b)$ and a shifted data independent term $\theta_i+(M-1)\omega_it/2$. Eq. 2.17 allows θ_i to be calculated through a recursive relationship $\theta_{i+1} = \theta_i + \pi x_i h_i$. All that is needed to specify the current signal is thus x_{i-1} and θ_i. If X_i is assumed to contain all information about the current state, X_i must be specified by $X_i = \{x_{i-1}, \theta_i\}$. As only rational h indices are considered, there are a finite number of elements X_i.

$I(t, X_i)$ and $Q(t, X_i)$ are determined for each possible X_i. They are used as mappings that map the data X_i into the Memoryless Modulator. The continuous phase encoder makes use of the recursive relationship defined here to express the data into the form $\{x_{i-1}, \theta_i\}$.

3.2.1 Multi-h and Rimoldi's Decomposition Approach

Multi-h CPFSK has no inherent single frequency that can be accessed during every bit period, due to the cyclically changing h-index. There is no single signaling frequency to reference phase from. The slope of the phase transitions is not identical in each bit period, as the slope is proportional to the changing h. These factors mean that tilting the phase trellis will not result in a 'memoryless' trellis. Multi-h has an additional component of memory due to the cyclic nature of the h-set.

Examining Multi-h further, reveals that binary M-level Multi-h scheme with $h = 1/q$ is a subset of a combination of q-level CPFSK and (q-1)-level CPFSK, also with $h = 1/q$. Using the Multi-h example $\{2/4, 3/4\}$, the related fixed-h CPFSK is 4-level $h = 1/4$, the two levels being 2 and 3. Figure 3-8 shows the relationship between the two schemes where the darkened transitions are the Multi-h scheme.

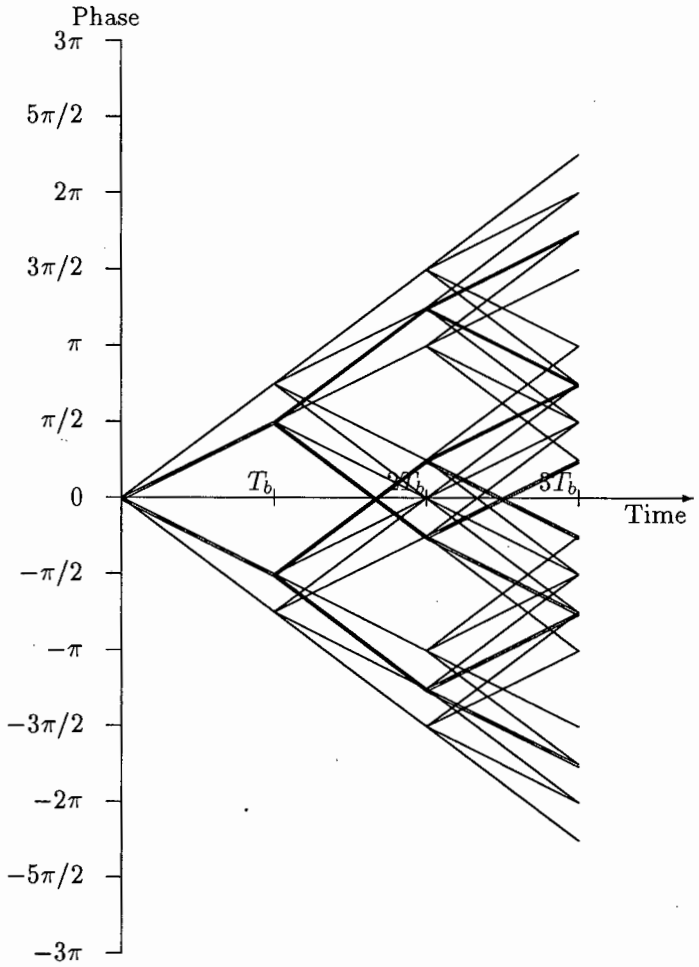

Figure 3-8: The $\{2/4, 3/4\}$ Multi-h Phase Trajectories on Selected 3-level and 4-level $h = 1/4$ CPFSK Trajectories

This observation allows Rimoldi's decomposition to be utilized by converting original information sequence to the appropriate levels before the CPE stage. The tilted phase of the $h = 1/q$ system is

$$\psi(t, \alpha) = x_i \omega_i (t - iT_b) + \theta_i + \pi(M - 1)t/qT_b. \tag{3.28}$$

The data-dependent terms are still specified by X_i. X_i' specifies the Multi-h data and must be mapped to X_i, the fixed-h CPFSK counterparts. θ_i and θ_i' are identical and $x_i = x_i'/p_i$. The implementation is shown in figure 3-9.

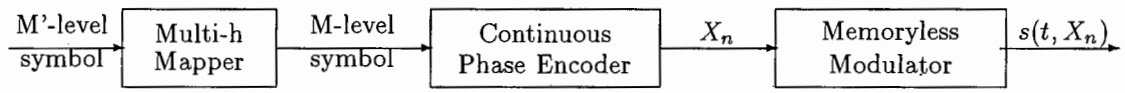

Figure 3-9: Decomposition of Multi-h CPM

Chapter 4

Error Correction Decoding Algorithms

Convolutional Encoding is a widely accepted method of incorporating the memory of the past M-ary symbols. Over the past couple of decades, a number of high performance error correction decoding algorithms have been developed, the best known being the *Viterbi algorithm.* Memory is a form of redundancy which enables the appropriate error correction decoder to find a sequence of highly probable events that would result in the received transmissions. Good codes have highly unique sequences lasting for a time no shorter than the period defined as the *constraint length*, where the *constraint length* is as long as possible.

There are two broad classes of error correction decoders. The *optimal decoder* is very exhaustive in it's approach and checks through all possible matching sequences. The *sub-optimal decoder* makes intelligent guesses to which trellis path to follow. Having not considered all possible paths, the sub-optimal decoder has made a trade-off against the exhaustive approach in return for speedier decisions and allowing for higher throughput rates.

In the *sub-optimal decoder* category, there are two further divisions [9]. These are the *sorting* types and the *non-sorting* types. The *non-sorting* types view only one path at a

time. Amongst the *sorting* types are decoders that use the Stack algorithm, the Bucket algorithm and the Merge algorithm. Amongst the *non-sorting* types are decoders that use the Single Stack algorithm and the Fano algorithm. All the sub-optimal decoders make use of a tilted accumulative metric known as the *Fano Metric.*

Anderson and Mohan [9] have established that the cost of the sorting decoders are generally higher than that of the Viterbi decoder and the non-sorting decoders. Single stack decoders have the least cost. Anderson and Mohan's cost analysis is based upon the *space complexity*, the resources required by the decoder, and the *time complexity*, the number of accesses made to these resources by the decoder.

As the Viterbi algorithm is simpler to understand, this will be discussed first. After this, the Fano algorithm will be described. The multiple stack algorithms combine methods from both of these algorithms and are discussed at the end of this chapter.

4.1 The Viterbi Algorithm

The Viterbi decoder compares the sequences received against a specific set of sequence patterns and finds the sequence pattern closest to the received sequence[10]. This sequence is selected from the set as it has the *maximum-likelihood* of being the actual transmitted sequence. In this section, the Viterbi algorithm is illustrated for a simple convolutional code before the fundamental differences between CPFSK implementation and a convolutional code implementation are detailed.

4.1.1 An Illustration of the Viterbi Algorithm

The nature of the Viterbi algorithm is illustrated for a convolutional code of constraint length 2, rate 1/2 and constructed with the generator polynomial $G(x) = (x^2 + x + 1, x^2 + 1)$. The encoder for this convolutional code is shown in figure 4-1 where D indicates a bit-period delay.

The information sequence is represented by $\vec{x} = (x_0, x_1, x_2, ...)$ where x_i is a binary

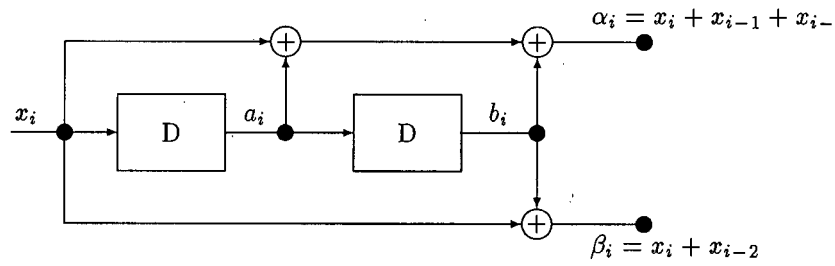

Figure 4-1: A $\{2, 1/2\}$ Convolutional Encoder

symbol. The output codewords are represented by $\vec{y} = (y_0, y_1, y_2, ...)$ where $y_i = \alpha_i \cdot 2^1 + \beta_i \cdot 2^0$. As an example, if the information sequence is $\vec{x} = (0, 0, 1, 0)$ then the transmitted codewords would be $\vec{y} = (00, 00, 11, 10)$ assuming that the initial conditions of the encoder were $a_0 = 0, b_0 = 0$.

It is convenient to represent each node in convolutional coding by the state held in the shift registers. At time iT_b, T_b being the bit period, the state will be $[m_i] = [a_i \cdot 2^1 + b_i \cdot 2^0]$. As a_i and b_i belong to $\{0, 1\}$, the possible values of $[m_i]$ are confined to the set $\{0, 1, 2, 3\}$. All possible trellis paths between the states $[m_i]$ are illustrated in figure 4-2. The upper branch leaving each state represents an input of $x_i = 0$ and the lower branch, an input of $x_i = 1$.

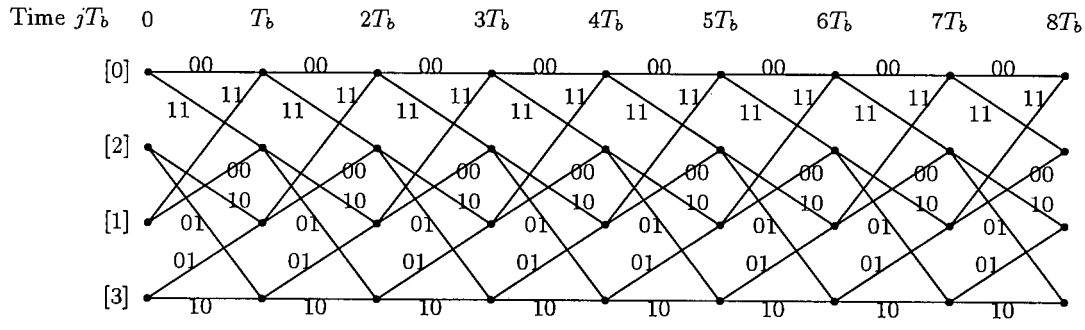

Figure 4-2: State Trellis for the $\{2, 1/2\}$ Convolutional Encoder

Let the transmitted codeword be $\vec{y} = (y_0, y_1, y_2, ...)$ and the received codeword after demodulation be $\vec{r} = (r_0, r_1, r_2, ...)$. $\vec{r}$ is the best estimation of $\vec{x}$ provided by a *maximum likelihood detector* of $\vec{y}$.

The branch metric from state $[m_{i-1}]$ to $[m_i]$ is defined by

$$d(m_{i-1}, m_i) = \|r_{i-1} - y_{i-1}\|. \tag{4.1}$$

This is the Hamming distance between the estimated detected symbol r_{i-1} and the transmitted symbol y_{i-1}. The transition corresponding to the received symbol r_{i-1} is the transition from $[m_{i-1}]$ to $[m_i]$. The lower the magnitude of the branch metric, the higher the probability that $y_{j-1} = r_{j-1}$ is true.

The partial path metric is defined as

$$M_j(m_j) = \sum_{i=1}^{j} d(m_{i-1}, m_i). \tag{4.2}$$

This is the sum of all Hamming distances between the estimated symbols and their corresponding transmitted symbols up to the time $t = jT_b$. This partial path metric can also be expressed as

$$M_j(m_j) = M_{j-1}(m_{j-1}) + d(m_{j-1}, m_j). \tag{4.3}$$

$Q_j(m_j)$ is defined to be the set of all paths that start at state [0] at time $t = 0$ and end at state $[m_j]$ at time $t = jT_b$. The smallest metric, $M_j(m_j)$ of all received paths contained in set $Q_j(m_j)$ is defined as $P_j(m_j)$. $P_j(m_j)$ comes from the best estimate of the transmitted sequence and is the weight of what is called the survivor path to state $[m_j]$ at time $t = jT_b$. The Viterbi decoder is based on finding the survivor path.

The following steps constitute the Viterbi algorithm:

Step 1: Initialize survivor path metrics for each state at time $t = 0$. If it is possible, weight the first state. This is possible in the convolutional encoding example as the assumption is made that the state registers contain zero's at time $t = 0$ or in other words the beginning node is the state [0]. Incorporating the weighting, the initialization is

$$M_j(m_0) = \begin{cases} 0 & \text{for } m_0 = 0 \\ \infty & \text{for } m_0 \neq 0 \end{cases}.$$

This ensures that all paths originate from the node 0 as in the encoder.

Step 2: Starting at time $j = 1$, the partial metrics $d(m_0, m_1)$ are computed for all possible states m_1. There are two possible $d(m_0, m_1)$ for each m_1 as two paths enter every m_1. The smaller $d(m_0, m_1)$ (survivor path) is chosen, the survivor path's path route and metric being stored.

Step 3: Increase j by 1. Compute the partial metrics $M_j(m_j)$ by adding $d(m_{j-1}, m_j)$ entering that state to the metric of the connecting survivor $P(m_{j-1})$ at the previous time $j - 1$. For each state, store the smallest path metric $P(m_j)$ and surviving path discarding all the other paths.

Step 4: Repeat step 3 until the received codewords r_j have run out. The final survivor with the smallest metric is the best approximation of the transmitted codeword $\vec{y}$.

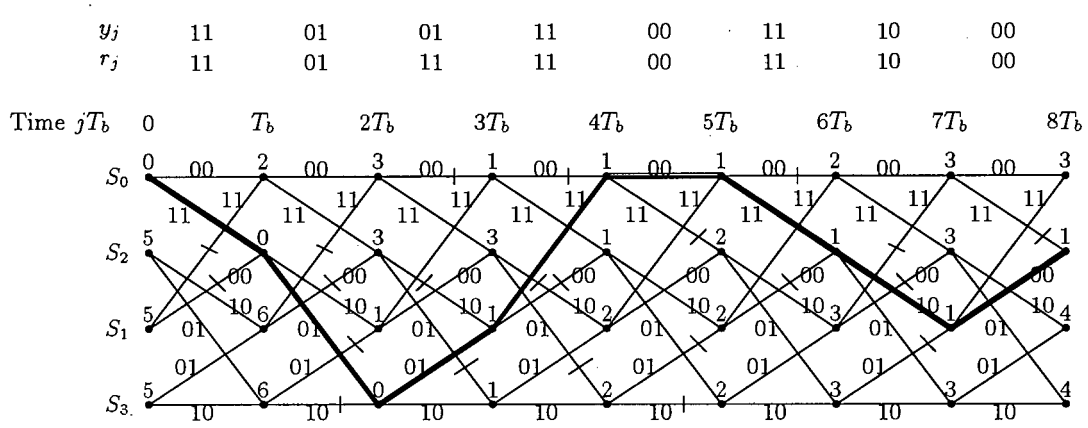

Figure 4-3: Path Tracking on the Convolutional Code State Trellis

The best way to illustrate the algorithm is by example. Let the input sequence be an 8-bit sequence $\vec{x} = \{1, 1, 0, 0, 0, 1, 0, 1\}$. The encoder output will be $\vec{y} = \{11, 01, 01, 11, 00, 11, 10, 00\}$ assuming the initial state of the encoder is [0]. Suppose that the demodulator detects the sequence $\vec{r} = \{11, 01, 11, 11, 00, 11, 10, 00\}$, the difference being an error in the detection of the third symbol. The survivor metrics are initialized to five at time $j = 0$ except for state [0] which is initialized to 0. Five is a sufficiently high metric to eliminate paths coming from a state other than [0] at time j=0. In figure 4-3 the survivor path metrics are indicated above every node. Paths that do not contribute to

the survivor path metric are eliminated and are indicated in the figure by being crossed out.

Consider step 3 for $j = 2$, state [0]. There are two paths entering this state, the upper path with the codeword 00, [0] $\rightarrow$ [0] and the lower path 11, [1] $\rightarrow$ [0]. The Hamming distance between the received codeword 01 and both of these codewords are equal to one as only one bit of the codeword differs from 01 in each case. The partial metric up to [0] at time $j = 1$ is 2 and up to [1] at time $j = 1$ is 6. By adding the Hamming distance calculated here to these metrics, the partial metric going via state [1] is much higher than by going via [0]. This means that the path [1] $\rightarrow$ [0] between time $j = 1$ and $j = 2$ is eliminated. The decoder stores the lowest partial path metric, 3, and the path [0]($j = 1$) $\rightarrow$ [0]($j = 2$) at node [0], $j = 2$.

At time $j = 8$, the lowest partial metric is 1 and is related to state [2]. Using this state and tracing back the surviving route, the original path is estimated. The best estimate is the path shown in bold in figure 4-3.

4.1.2 Viterbi Algorithm CPFSK Implementation

The Viterbi decoder receives the optimal estimate of the transmitted frequency ($\hat{x_i}$), and phase, $\hat{\phi_i}$ from the demodulator. Frequency is directly related the x_i and h_i information. Incorporating phase information utilizes the memory in the modulation scheme and thus makes use of the inherent error correction properties to improve the demodulators robustness against noise.

The branch Euclidean distance from the state m_{j-1} to the state m_j is defined as

$$d_{ED}(m_{j-1}, m_j) = \frac{1}{2E_b} \int_{jT_b}^{(j+1)T_b} [s(t, r_{j-1}) - s(t, y_{j-1})]^2 dt \tag{4.4}$$

where r_j is the symbol stream received and y_j is the symbol stream under consideration.

The partial path metric is defined as

$$M_{ED,j}(m_j) = \sum_{i=1}^{j} d_{ED}(m_{i-1}, m_i) \tag{4.5}$$

as before.

The amended algorithm is outlined below

Step 1: Initialize all the path branch metrics at $j = 0$ to the same value:

$$M_{ED,0}(m_0) = 0 \tag{4.6}$$

for all m_0.

Step 2: Starting at time $j = 1$, all partial branch Euclidean distances $M_{ED,j}(m_j)$ are calculated for all possible m_j. As before, a survivor path of those entering each node is chosen. Better correlations are detected by smaller Euclidean distances rather than smaller Hamming distances. Note that in practice, CPFSK Demodulators provide correlations rather than direct measurements of Euclidean distances to the error correction decoders.

Step 3: The process is repeated for $j = j + 1$. All the partial branch Euclidean distances, $M_{ED,j}(m_j)$ are calculated using the same recursive relationship as before. At each node, the survivor path and the associated metric are stored while the other paths are discarded.

Step 4: Step 3 is repeated until all the received codewords have run out. The final survivor is the path with the smallest Euclidean distance indicates the most probable path and is the error corrected estimate of x_i.

4.2 Fano Algorithm

A sub-optimal decoding scheme is a more dynamic system of measuring the merits of a particular path. In the optimal error correction decoder scheme, the merits of a path is measured using accumulative Euclidean or Hamming distances which have not been scaled according to the time the path has travelled. These accumulated distances are

compared to each other if and only if they relate to the same bit period. In other words, the length of the path is of no relevance in such a comparison. A sub-optimal error correction decoder needs the knowledge of distance travelled to be incorporated in the accumulated metric as it needs to make such comparisons.

It is instructive to consider the features of accumulated metrics over a time period. Wozencraft [11] provides an excellent description of these processes. Examining the error properties of Convolutional encoding, it becomes clear that if an incorrect path is followed, the accumulative Hamming distance increases at a rate of 1/2. This is a considerably higher rate than that followed by a path following the correct sequence. A line of slope gradient in between these two slope gradients can serve to distinguish whether a correct or incorrect path is being pursued. These features are best illustrated through example.

4.2.1 Convolutional Example

Consider a convolutional code of rate $r = 1/5$ as illustrated in figure 4-4. Such a code can be determined by set partitioning [10, 12, 13]. The rate indicates the ratio of incoming data word lengths to transmitted code word lengths. The example is truncated at node $l = 4$, where l indicates node depth.

If $\vec{x}$ represents the symbol stream transmitted and $\vec{y}$, the convolutional encoded version of the symbol stream, the received codewords $\vec{r}$ are given by $\vec{r} = \vec{y} + \vec{n}$ assuming $\vec{n}$ is a noise term resulting from a Binary Symmetric Channel (BSC).

The Hamming distance between any two codewords is defined by the number of disagreements made in a bitwise comparison. For example, 00010 and 00100 have only three bits which agree, thus the Hamming distance is two. If the Hamming distance between two codewords of length five is three or greater, it is difficult to match such codewords. The accumulated Hamming distance between path $\vec{r}$ and the corresponding path on the trellis, $\vec{y}$, up to the depth l, is represented by $d(l)$.

Consider if there is no noise and a divergence away from the matching codeword occurs,

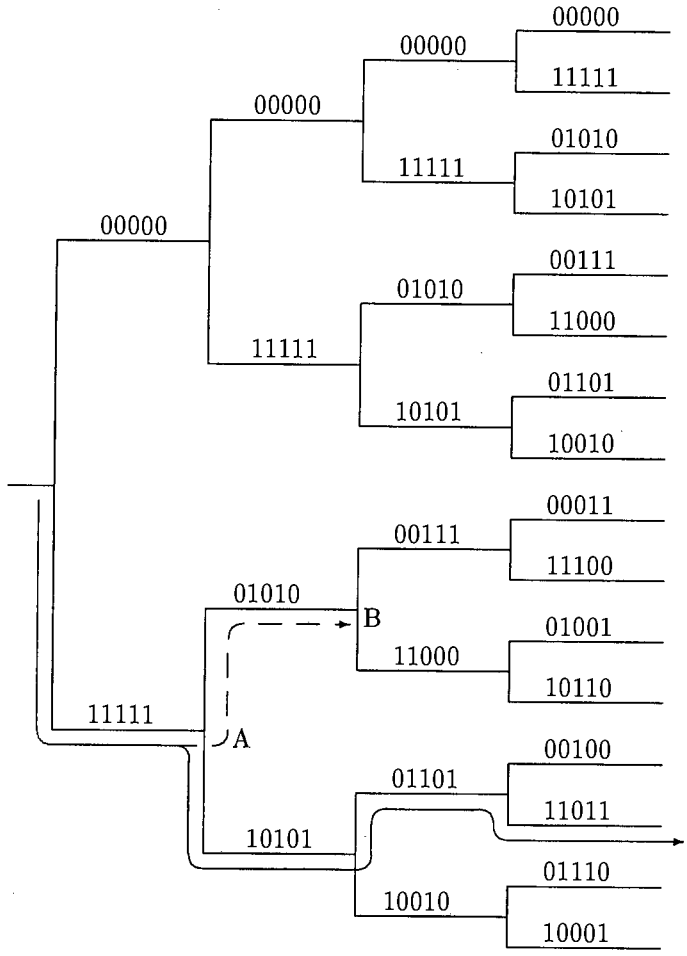

Figure 4-4: Example: Sequential Code with $\nu = 5$

all subsequent paths connected to the incorrect codeword accumulate, on average, Hamming distances at a rate in excess of $\nu/2$. If the correct path is pursued, the average rate of Hamming distance accumulation is dependent on the channel transition probability $p << \nu/2$.

Assume that the data stream and channel noise are

$$\vec{x} = (1,1,0,1) \tag{4.7}$$

$$\vec{n} = (10010, 00111, 00000, 00100) \tag{4.8}$$

respectively. The transmitted codeword vector will be

$$\vec{y} = (11111, 10101, 01101, 11011) \tag{4.9}$$

and the received codeword vector will be

$$\vec{r} = (01101, 10011, 01101, 11111). \tag{4.10}$$

There is little agreement with codewords subsequent to the point marked B on the figure even though the closest agreement between the second codeword received was the codeword between points marked A and B. This is due to the large amount of noise occurring at this period of time as given in eq. 4.8. The agreement of the third and fourth codewords for the path indicated is strong, indicating the likelihood of the second codeword being 10101.

The non-sorting decoder pursues a particular path until a strong rise in the slope of the accumulative Hamming distance is detected. Once this is detected, the decoder will search for another path, diverging slightly earlier, that holds more promise.

Consider figure 4-5. In the figure, the horizontal axis represents time and the vertical axis, the accumulated Hamming distance from time $t = 0$. If the decoder follows the correct path, the accumulated distance increases at a slope of $p\nu$. As soon as the path pursued is no longer the correct path, the accumulated Hamming distance commences to increase at a rate of $\nu/2$. The decoder detects this through observing whether the rate at which the accumulated distance is increasing by exceeds some criteria. This criteria is a specified rate, which is smaller than $\nu/2$ and higher than $p\nu$. The criteria is illustrated by the line $k(l)$ in figure 4-5 where $k(l) = p'\nu l$ and $p < p' < 1/2$.

Depending on the noise in the channel, if the criterion function $k(l)$ has too small an intercept $k(0)$, the criteria may be too stiff for even the correct path to achieve. This is rectified by relaxing the criteria through the addition of a constant increment Δ. Too large a Δ will make the correct and incorrect paths indistinguishable against the criteria and adversely affect the error probability.

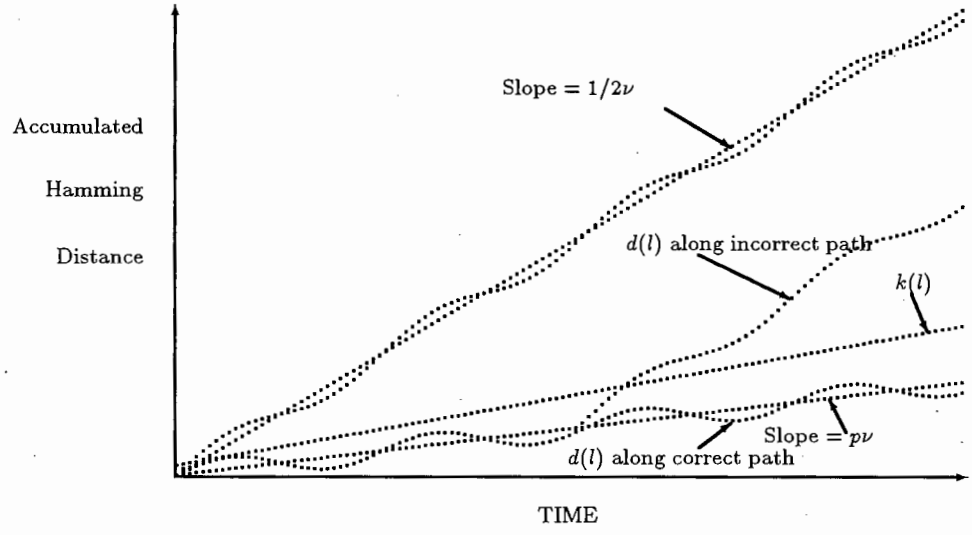

Figure 4-5: An Illustration of Accumulative Hamming Distance Trends

The way to proceed is to use a relatively small threshold spacing Δ, and if the decoder is unable to proceed forward, another increment of Δ is added to $k(l)$ repeatedly until the decoder is able to find a path (by backing up if necessary) so as to proceed forward.

On the other hand, if at a latter stage, it is determined that a lower k(l) will suffice, the threshold levels will be dropped again. For better error probabilities, tight thresholds which have sufficient space for forward movement must be maintained.

If Δ is too small, a large number of computations results. Lin and Costella [14], page 363, suggest using a Δ between 2 and 8 if the metrics are unscaled. If the metrics are scaled, then this numbers should be scaled by the same quantity.

4.2.2 Fano Algorithm Implementation

It is convenient to tilt the plot shown in figure 4-5 so as to be able to disregard the slope of $k(l)$. If the decoder's tilted accumulated metric is decreasing, this is an indication that the correct path is being pursued. The tilted metric is related to the accumulated metric by

$$t(l) = d(l) - p'\nu l \tag{4.11}$$

As the sequential decoder penetrates branch by branch deeper into the code tree, it maintains a running count of $t(l)$. After each successive penetration the decoder compares the $t(l)$ against the a discard criterion function $k(l)$. If $t(l)$ ever exceeds $k(l)$, the tentative path is discarded as too improbable. The decoder then backs up to the nearest unexplored branch for which $t(l) \leq k(l)$ and again starts moving forward as far as the discard criterion function $k(l)$ permits.

At every node under investigation, the tilted metric is computed and is said to *satisfy* the threshold level if it lies below the threshold level ($k(l)$) and *violates* the threshold level if it lies above the threshold level. The *tightest* threshold satisfied by a node is the lowest threshold level that permits the node to satisfy the threshold.

Of the branches diverging from a given node, the branch terminating on the node with the lowest $t(l)$ is said to be the *best* route and the branch with the highest is called the *worst* route.

Sequential decoders consider one node at a time. A moveable *search node pointer* is used to index the position of the node under investigation. The Fano decoder maintains a running threshold, denoted by T, and is equal to $k\Delta$, k being an integer. The running threshold is said to be tightened when k is assigned so that T is the tightest threshold satisfied by the search node.

The Fano decoder searches for the correct path by moving it's search node pointer through the received distance tree. The pointer can move backward or forward, but only to an adjacent node connected by a branch. The pointer's movement is controlled by the flow chart in figure 4-6. An essential feature is that the pointer is never moved if the threshold level will be violated. If movement is not possible, the threshold level is relaxed until such a move is possible.

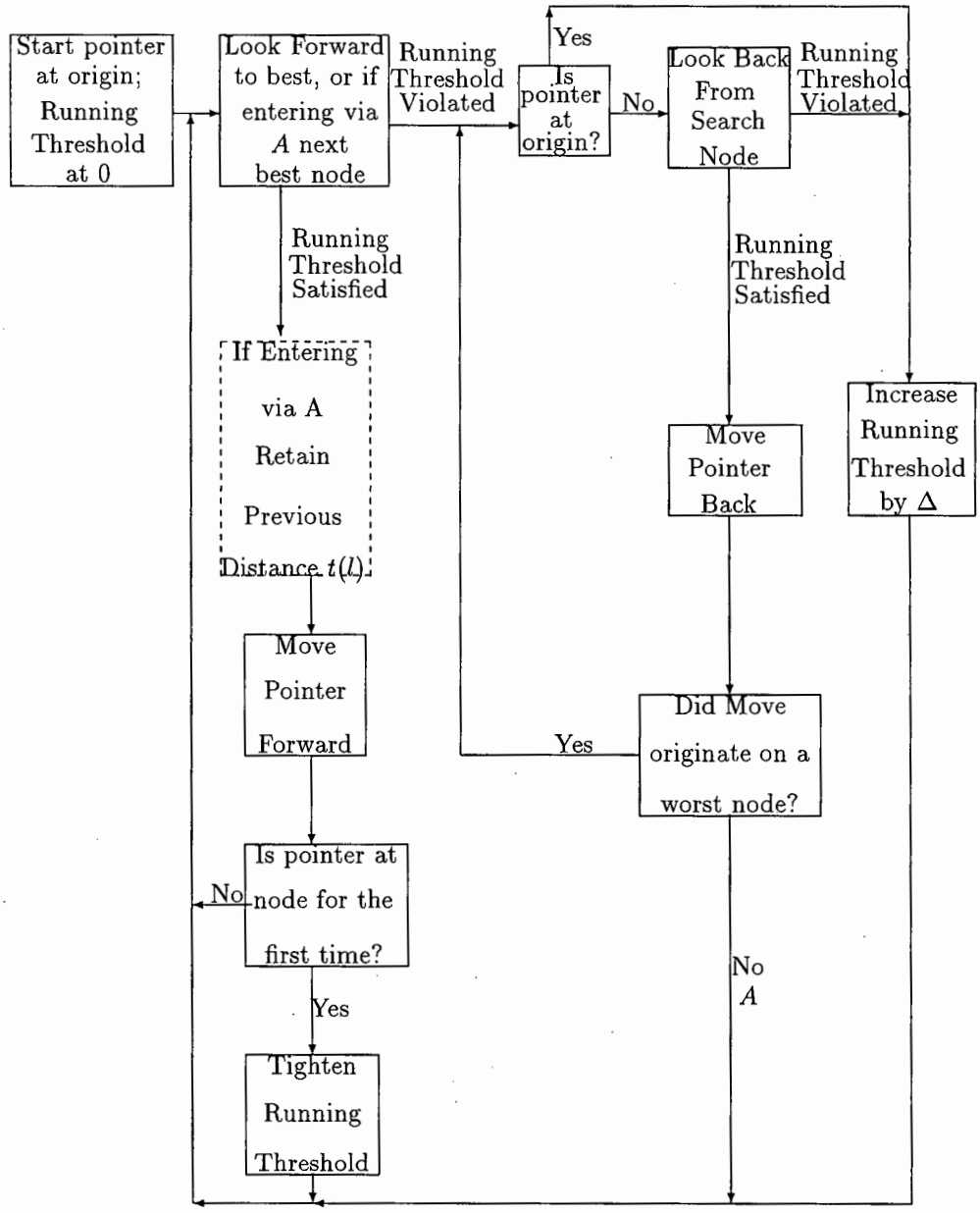

Figure 4-6: Fano Algorithm Flowchart

The best way to comprehend the algorithm is through example. Consider Figure 4-7. The decoder starts its tree search at the initial node labeled 0. The initial running threshold is zero. The decoder looks forward to node 1. Since this node will satisfy the running threshold, the pointer is moved forward.

The tightest threshold of node 1 is zero. The decoder then looks forward to the node labeled 2. As the threshold level will not be violated by a move forward, the pointer

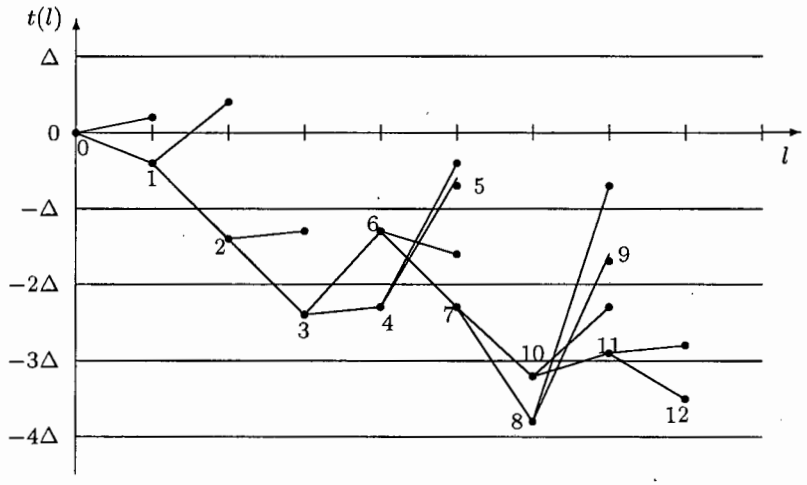

Figure 4-7: An Example of the Fano Decoder in Progress

is moved forward. The running threshold can be tightened to $-\Delta$. The procedure is continued until the pointer tries to move forward from node 4 to node 5. The threshold will be violated by such a move forward but not backwards. The pointer is moved backwards and alternative paths are investigated (node 6 and node 2). No further movement can be made without relaxing the threshold (ie. to $-\Delta$). The decoder attempts to proceed through node 5, then through node 6 as node 5 violates the current threshold. It is able to proceed through node 6. The rest of the figure is self-explanatory. The complete process is given in table 4.1.

Pointer at Node	Running Threshold	Action (x Indicates Threshold Violation)				
0	0	look at 1		point at 1		
1	0	look at 2		point at 2		set $T = -\Delta$
2	$-\Delta$	look at 3		point at 3		set $T = -2\Delta$
3	-2Δ	look at 4		point at 4		
4	-2Δ	look at 5	x	point at 3		point to 3
3	-2Δ	look at 6	x	point at 2	x	set $T = -\Delta$
3	$-\Delta$	look at 4		point at 4		
4	$-\Delta$	look at 5	x	point at 3		point to 3
3	$-\Delta$	look at 6		point at 6		
6	$-\Delta$	look at 7		point at 7		set $T = -2\Delta$
7	-2Δ	look at 8		point at 8		set $T = -3\Delta$
8	-3Δ	look at 9	x	point at 7	x	set $T = -2\Delta$
8	-2Δ	look at 9	x	point at 7		point to 7
7	-2Δ	look at 10		point at 10		set $T = -3\Delta$
10	-3Δ	look at 11	x	point at 7	x	set $T = -2\Delta$
10	-2Δ	look at 11		point at 11		
11	-2Δ	look at 12		point at 12		set $T = -3\Delta$

Table 4.1 : Example of Fano Algorithm in Progress

4.2.3 Fano Metric

For a code of constraint length K with code rate $r = k/n$ over a discrete memoryless channel, the bit metric can be represented by

$$BM_i(r_{ij}, y_{ij}) = \log_2 \left(\frac{p(r_{ij}|y_{ij})}{p(r_{ij})} \right) - r. \tag{4.12}$$

y_{ij} denotes the i^{th} bit of the binary label on the j^{th} branch and r_{ij} denotes the corresponding bit of the received sequence.

The j^{th} Fano branch metric is given by

$$BM(r_j, y_j) = \sum_{i=1}^{n} \left[\log_2 \left(\frac{P(r|y)}{P(r)} \right) - r \right] \tag{4.13}$$

where r is the transmitted symbol and y the received symbol. It is common practice to scale $BM(r, y)$ by a common factor so that the result is near integer by determining $\min |P(r|y)|$.

4.2.4 Enhancements to the Fano Algorithm

Two notable enhancements have been made to the Fano algorithm. Fano [11] noted that keeping track of whether the pointer had been at a node beforehand to be memory intensive. He proposed the introduction of a toggle θ which eliminated this requirement.

The amended Fano algorithm starts with $\theta = 0$ and changes to $\theta = 1$ if the threshold is violated. When deciding if the pointer was at the node for the first time, if $\theta = 0$, the threshold hasn't been violated so the pointer hasn't been to that node before. If $\theta = 1$ and the move did not originate or terminate on a node violating the threshold $T - \Delta$, then the pointer has visited the node previously. If $\theta = 1$ and the previous condition is not the case, then θ must be reset to zero.

Katakol and Maskara [15, 16] noted that stringent threshold conditions used when backtracking causes some instability in that there is much movement forwards and backwards. This movement can be decreased by retaining the current $t(l)$ on the backwards movement if entering via A as indicated by the dashed block in figure 4-6.

4.2.5 Fano Algorithm CPFSK Implementation

Consider the example of CPFSK with $h = 1/4$. The squared intermediate Euclidean distances, d^2 are calculated for each signal comparison and are shown in Table 4.2.

Phase	d^2, $x_i = x_j$	d^2, $x_i \neq x_j$
0	0.000	0.363
$\pi/4$	0.292	1.000
$\pi/2$	1.000	1.637
$3\pi/4$	1.707	1.900
π	2.000	1.637
$5\pi/4$	1.707	1.000
$3\pi/2$	1.000	0.363
$7\pi/4$	0.292	0.100

Table 4.2 : Intermediate Euclidean Distances of $h = 1/4$ CPFSK

If a single error in detection occurs, the path pursued will increase at a rate of between 0.292 and 2.000. If there is great disagreement between the path being pursued and the correct path, the average increase rate, as determined from the average of the last column, is 1.000. Based on such observations, the important figure serving as a criteria must be 0.292. p' must satisfy $0.000 < p' < 0.292$.

This particular CPFSK implementation is symmetric and due to coding, initial phase differences of odd numbers of $\pi/4$ are less probable. The criteria on p' can be loosened to be $0.000 < p' < 1.000$. Choosing p' to be 0.500, the tilted Euclidean distances, t, are recorded in Table 4.3.

Phase	t^2, $x_i = x_j$	t^2, $x_i \neq x_j$	$[t^2]$, $x_i = x_j$	$[t^2]$, $x_i \neq x_j$
0	-0.500	-0.137	-7	-2
$\pi/4$	-0.208	0.500	-3	7
$\pi/2$	0.500	1.137	7	16
$3\pi/4$	1.207	1.400	17	20
π	1.500	1.137	21	16
$5\pi/4$	1.207	0.500	17	7
$3\pi/2$	0.500	-0.137	-7	-2
$7\pi/4$	-0.208	-0.400	-3	-6

Table 4.3 : Tilted Intermediate Euclidean Distances of $h = 1/4$ CPFSK

Furthermore, by dividing all t^2 by 0.07, the nearest integers found can be used in the implementation. Lin and Costello[14], suggest using Δ between 2 and 8 multiplied by the scaling factor which has the effect of allowing for reasonable mobility forward. In this example, the upper integral boundary of p' equality is 7. It is suggested that $10 < \Delta < 25$.

The actual demodulator structure uses a set of signal correlations rather than the free distance between signal paths. There is a direct relationship between the correlations determined and the free distances. The difference in correlations between two phase trellis paths is twice the free distance between the paths.

4.3 Multiple Stack Algorithms

Multiple stack algorithms can be thought of as a hybrids between the optimal error correction decoding algorithms such as the Viterbi algorithm and single stack algorithms such as the Fano algorithm. They are considered to be advantageous over the Fano algorithm in very high-speed applications.

The Viterbi decoder evaluates a number of feasible paths (in fact it is exhaustive), a multiple stack decoder also evaluates a number of feasible paths (not exhaustively but intelligently). Like the Fano decoder, multiple stack decoders intelligently choose feasible paths to investigate, by using the Fano Metric. Multiple stack decoders do not necessarily compare paths of identical lengths as is the case of the Viterbi decoder. There is a price however. Intelligent choice requires computations, the larger the stack, the larger the number of computations to be made. The type of computations made by the Viterbi decoder are simple and predictable but have large hardware storage requirements.

The specific multiple stack algorithm known as the 'Stack algorithm' works as follows: Note that the accumulated tilted metrics are stored together with complete path information originating from a common origin node.

• Start at the begin node. Initialize the Accumulated Metric to zero
and store both the metric and associated path route at the top of the stack

- Evaluate the Accumulated Tilted Metric resulting from moves
  commencing from last node of the path at the top of the stack.
- Remove the top entry of the stack and insert the newly evaluated entries.

• Loop until Top metric in stack has a path associated with it that
has a terminating node.

- Order Metrics in stack placing best metric and associated path
  route at the top. Remove the worst Metric if there is insufficient
  space in the stack.
- Determine Accumulated Metrics resulting from moves
  commencing from the terminating trellis position of the path placed on
  the top of the stack.
- Remove the top entry of the stack and insert the newly evaluated entries.

• The Top Metric estimates the best path.

This algorithm is illustrated in figure 4-8 and is best illustrated by example. Using the code illustrated in figure 4-4, suppose that the transmitted sequence was

$$\bar{r} = \{0001, 0110, 1110, 0000, 0011\}. \tag{4.14}$$

The procedure proceeds as follows

Step 1	Step 2	Step 3	Step 4	Step 5	
0(-1)	01(-2)	010(-3)	1(-4)	0101(-5)	(Terminate)
1(-4)	1(-4)	1(-4)	0101(-5)	00(-5)	
	00(-5)	00(-5)	00(-5)	10(-6)	
		011(-6)	0100(-6)	0100(-6)	
			0111(-6)	0111(-6)	
				11(-7)	

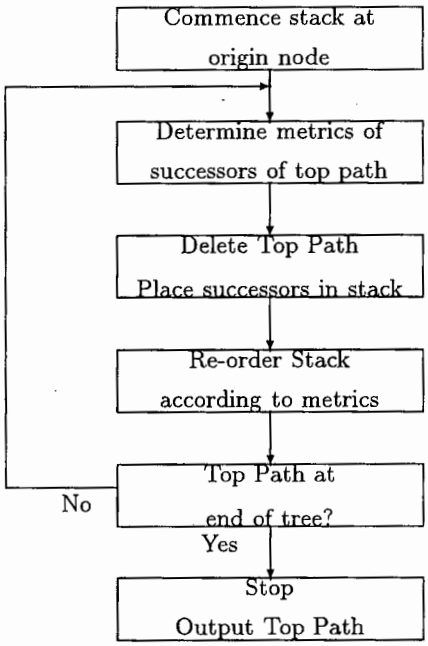

Figure 4-8: Flowchart of the Stack Algorithm

The numbers in the brackets are the negative image of the accumulated Hamming distances and are associated with the path indicated.

Of course, re-sorting the stack every step is computation intensive and in any case only the top of the stack is of interest. Once the metric is determined, it could be placed with other results of similar order and only the best candidates need be sorted. This is the basis of the Bucket algorithm in which the metrics between certain boundaries are stored, not in any order.

It may be determined that two paths merge, if this happens it is possible to discard the worse of the two. If the path is discarded, it will save the decoder from considering a path which will not, in any case, result in a best estimate.

4.4 Comparative Analysis

Each method of error correction has unique cost and performance merits and must be considered before deciding on a particular method. A comparative cost and performance based analysis using the cost function introduced by Anderson and Mohan [9] is performed.

Amongst the sub-optimal decoders, sorting decoders are generally much more expensive than non-sorting decoders. Sub-optimal decoders favor schemes with large number of states in which only a few code tree paths need to be pursued. Strong applications of sub-optimal methods are amongst others the identification of image contours, text and speech recognition and digitizing of images [9].

Traditionally, the means of determining decoder efficiency has solely been the *node computation* (the number of branches visited during the progress of the scheme divided by the branches released as output). Anderson and Mohan [9] have broadened the measure of decoder cost to include the number of storage elements in a scheme and the number of accesses to them. The *space complexity* of a decoder is the size of the resources required by the decoder and the *time complexity* is the number of accesses made by the decoder to this resource. The product of these two, the *space-time complexity* and the sum of these two are both useful measures of overall cost. The *space-time* product assumes that storage blocks "'wear out" after a certain number of accesses and is proportional to their speed.

Three variables dominate the cost of the decoders. These are the length of the path the decoder can retain, L, the number of paths it can retain, S and the expected node computation, $< C >$. L and S can be assumed to be finite and fixed constants.

There are two schemes in which the decoder releases branches of the estimated path. It can either release the paths in blocks of L or incrementally. Both approaches commence from a root node at some point i and proceed to determine the most lightly path segment up to the point $i + L$. The block approach then dumps this path segment and re-initializes the root path to be the node at $i + L$ of the path dumped. It then

proceeds to search for the best path from the point $i + L$ to $i + 2L$ and so on. The incremental approach dumps only the branch from i to $i + 1$ and re-initializes the root node to the node at point $i + 1$ on the path. It then proceeds to find the best path from $i + 1$ to $i + L + 1$. It can be seen that in the block method the block is examined only once. In the incremental method, the block is examined effectively $(1+(L-1)/L+(L-2)/L+...(1/L) = 1+(L-1)/2$. The incremental method will have examined partial paths terminating $L-1$ ahead of all released branches whereas in the block method some branches will have had no future examination of extending paths and will be subsequently be used in producing future root nodes. In concluding, the block method is much faster but is far less reliable and only the incremental method will be considered.

Truong [17] has described an economic method of implementing the incremental method of the Viterbi algorithm. The data is released after twice the truncation depth.

It may be possible to calculate specific metrics when they are sort after rather than storing metrics exhaustively. As an example, the demodulator in the following chapter determines in-phase and quadrature-phase correlations. These are processed by a branch metric calculator. This can reduce the calculation $< C >$ overhead.

The Fano decoder compares paths emanating from the current node before moving it's focus. At best, it needs to consider ML nodes (single path) to trace the correct node. At worst, it will consider L^M nodes, the same as the Viterbi decoder.

The results of the cost analysis are summarized in table 4.4.

Decoder	*Space-Time* Product	Comments
Fano	$L < C_{Fano} >$	$< C_{ML} > < < C_{Fano} > \leq < C_{Viterbi} >$
Viterbi	$L < C_{Viterbi} >= S_{Viterbi}$	
Stack	$LS^2{}_{Stack} < C_{Stack} >$	

Table 4.4 : Cost Analysis of Error Correction Decoders

Chapter 5

Multi-h CPFSK Demodulation

The demodulation process is responsible for recovery of the symbol stream from the received signal. In CPFSK, this is accomplished through the prior knowledge of the symbol-frequency mappings and using correlations to determine the inverse mappings. Clearly, timing is critical in establishing reliable correlations. Multi-h CPFSK demodulation requires the accurate knowledge of three clock rates, namely the baud rate (symbol timing), the carrier frequency and the superbaud rate (the h-set size modulo of the baud rate).

The continuous nature of CPFSK restricts the possible phase changes that can occur at bit-period interfaces to be zero. This results in symbol stream information being incorporated in the carrier phase. In fact, accurate knowledge of the phase at the bit-period boundaries is adequate information to reconstruct the symbol stream. Generally, the phase information is used in conjunction with frequency information, via means of the branch metric and error correction methods to determine the transmitted symbol stream.

The first demodulation method that was investigated, correlates the incoming signal with reference signals at the actual signaling frequencies. In chapter 3, it was shown that Multi-h CPFSK can be constructed without a carrier, by using a bank of the oscillators selected by digital logic circuitry (a state machine). By using the non-linear element described in section 5.1 and correlating the incoming signal with the required

multiplicative factor of the various signaling frequencies, the symbol stream can be estimated using the inverse symbol-frequency mapping. In this manner, there is no need to track the carrier frequency or the superbaud rate. The non-linear element destroys the phase information held in the signal. The phase information must be retrieved through separate phase recovery circuitry. This method is rather complex and suffers numerous synchronization problems as there is no single reference frequency and relationships need to be physically established between the different correlators. If there are many phase states, this method becomes excessively hardware intensive. It is for this reason that this method was abandoned.

The second method is to decouple the incoming signal from the carrier using in-phase and quadrature-phase arms and using matched filters to determine the signalling frequencies. The two arms can be used to reveal the phase information.

Mazur and Taylor [18] use tentative decisions (decisions based upon a single observation interval). By modelling the carrier phase as a first-order Markov process, Wilson and Hsu [19] used an extended state form of the Viterbi Decoder to jointly estimate data and carrier phase. Liebetreu [20] used random phase acquisition models for simultaneously tracking phase and decoding data symbols. Premji and Taylor's[4, 5] receiver scheme jointly establishes all modes of synchronization with the demodulation process.

In the first section of this chapter, the timing retrieval used in some of the above approaches is discussed. The remainder of this chapter is devoted to the Premji-Taylor receiver structure. Firstly, the development of the receiver structure through Maximum Likelihood Techniques is detailed. Secondly, error correction methods are applied to the output of the demodulator and incorporated into the synchronization circuitry of the receiver. Lastly the synchronization analysis is detailed for non-symmetric Multi-h indices.

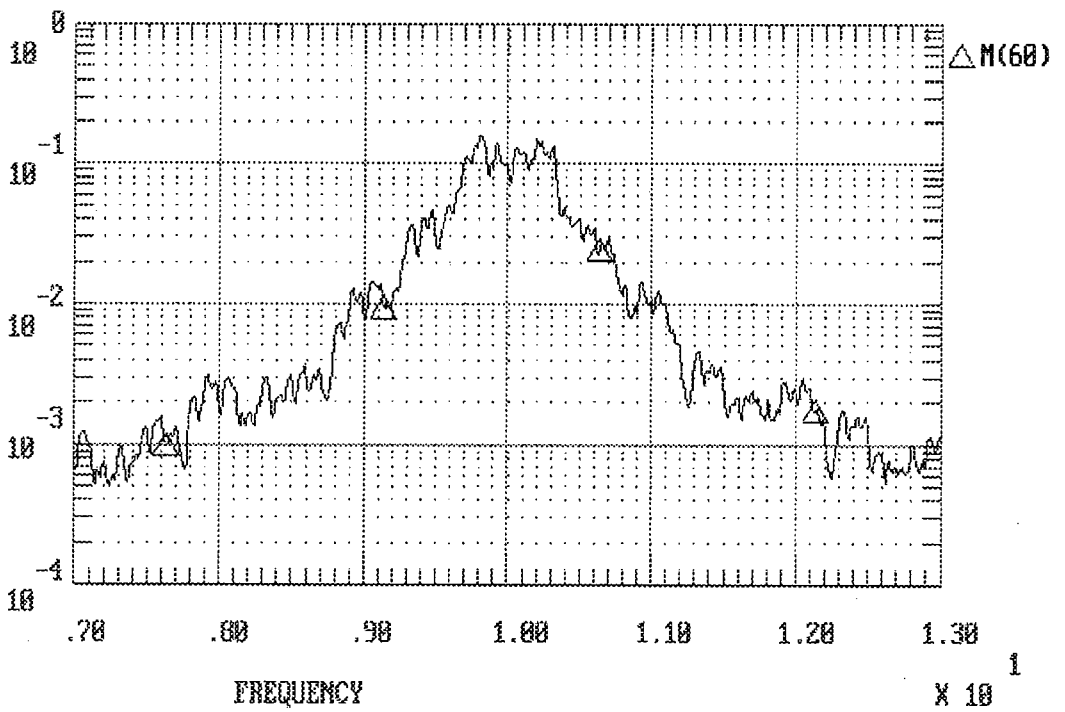

Figure 5-1: {1/4,3/4} Multi-h Spectrum, $f_c = 10Hz, f_b = 1Hz$

5.1 Dual Synchronization Demodulation Structure

Lereim [21, 18] observed that if the condition

$$\prod_{\text{all } i} |h_i| < 1 \tag{5.1}$$

is satisfied, the Multi-h spectrum frequency does not exhibit a line structure. Mazur and Taylor [18] suggested using a non-linear element to produce a spectrum with a line structure so that the symbol timing as well as the superbaud timing could be extracted. Consider a Multi-h signal is described by the h set,

$$h_i \in \{h_1, h_2, h_3, ..., h_n\}, \quad \text{such that } h_i = \frac{p_i}{q}\ , p_i, q \in \mathcal{Z} \tag{5.2}$$

and modulated about f_c with the baud rate f_b. By raising this signal to the power of q, a signal that contains the identical frequency spectral components of the Multi-h

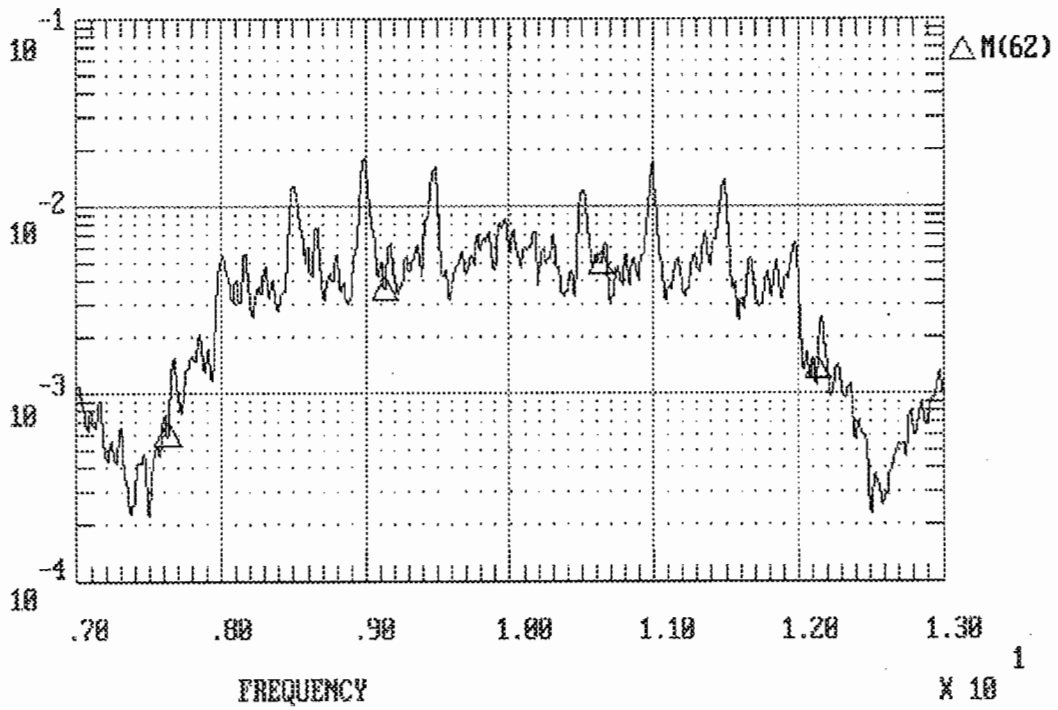

Figure 5-2: $\{1/4,3/4\}^4$ Multi-h Spectrum, $f_c = 10Hz, f_b = 1Hz$

signal described by the h set,

$$h_i \in \{qh_1, qh_2, qh_3, ..., qh_n\} \tag{5.3}$$

modulated about qf_c with the baud rate qf_b is obtained.

The non-linear element eliminates all the phase information in the resulting signal and produces a signal having a line spectrum. Particular lines in this spectrum are directly related to the signaling frequencies and some other lines are spaced at the baud rate frequency. This property is clearly evident by the frequency spectrum of a $\{1/4, 3/4\}$ Multi-h spectrum shown in figure 5-1, constructed with the simulation package TESLA.

If the same $\{1/4,3/4\}$ Multi-h signal is raised to the power of four, the frequency spectrum of the resulting signal will be that shown in figure 5-2. The line components are clearly evident. Note that the difference between the line components is related to the baud rate.

Mazur and Taylor's Multi-h CPFSK receiver structure [18] uses this non-linear device

to produce the required timing information in their demodulator.

5.2 Premji-Taylor Multi-h Receiver Structures

Premji and Taylor [4, 5] developed receiver structures which incorporate the timing retrieval in the demodulator. These structures were developed from direct application of maximum likelihood estimation techniques (ML).

In this section, the same analysis is followed and extended to accommodate the use of non-symmetric Multi-h CPFSK indices. To accomplish this analysis, the notation ω_{i,x_i} or in shorthand $\omega_{i,i}$ is introduced. ω_{i,x_i} is dependent on both the bit-period interval commencing at $t = iT_b$ and the symbol x_i transmitted in this interval. ω_{i,x_i} is the cyclically selected frequency $\omega_{i \bmod n, x_i}$. To illustrate this notation, suppose that the h-set size $n = 10$ and the transmission symbol in the twelfth bit-period is given by $x_{12} = -1$, the corresponding frequency is then given by $\omega_{12,-1} = h_2{}^{-1}\pi/T_b$.

5.2.1 Maximum Likelihood Demodulation

The received signal $y(t)$, consists of the original signal component $x(t)$, summed with a noise component $n(t)$, assuming that the channel is an Additive White Gaussian Noise (AWGN) channel. If the baseband components are indicated with a tilde, the received signal in the i^{th} bit-period can be decomposed as

$$y(t) = \mathrm{Re}[\tilde{y}(t)\exp(j\omega_c t)] = \mathrm{Re}\left[(\tilde{x}(t,\tau_0,\theta_0,x_i) + \tilde{n}(t))\exp(j\omega_c t)\right]. \tag{5.4}$$

Here

$$x(t,\tau_0,\theta_0,x_i) = \mathrm{Re}\left[\tilde{x}(t,\tau_0,\theta_0,x_i)\exp(j\omega_c t)\right], \tag{5.5}$$

$$\tilde{x}(t,\tau_0,\theta_0,x_i) = \sqrt{\frac{2E_b}{T_b}}\exp\left[j\left(x_i\omega_i(t - iT_b - \tau_0) + \phi_i + \theta_0\right)\right] \tag{5.6}$$

and $\tilde{n}(t)$ is bandlimited AWGN with power spectral density $N_0/2$. ϕ_i is the residual phase which is the accumulated phase changes, with reference to the carrier, up to the $(i-1)^{th}$ bit-period. The synchronization parameters, θ_0 and τ_0 indicate phase

and frequency shifts between the transmitted and received signals respectively. The receiver must be able to estimate θ_0 and τ_0 to maintain lock between the receiver's reference clocks and the incoming signal. These parameters can be found by observing that $\theta = \theta_0$ and $\tau = \tau_0$ maximize the maximum likelihood function,

$$L(\tau, \theta, x_i) = C \exp\left[-\frac{1}{N_0}\int_{T_0}(y(t) - x(t, \tau, \theta, x_i))^2 dt\right] \quad , \tag{5.7}$$

over the observation period $T_0 = NT_b$ where N is an integer. C is a constant that can be discarded as it plays no role in the maximization. In logarithmic form, the maximization is

$$\begin{aligned} \Lambda(\tau, \theta, x_i) &= \ln[L(\tau, \theta, x_i)] \\ &= -\frac{1}{N_0}\int_{T_0}\left\{y^2(t) - 2y(t)x(t, \tau, \theta, x_i) + x^2(t, \tau, \theta, x_i)\right\} dt. \end{aligned} \tag{5.8}$$

As $y^2(t)$ has no dependence on the maximization parameters, τ or θ, it can be discarded from the maximization process. The terms dependent on the maximization parameters resulting from the term $x^2(t, \tau, \theta, x)$ are of order $1/2\omega_c T_b$. As $\omega_c T_b >> 1$ in general, $x^2(t, \tau, \theta, x)$ can be ignored in the maximization process. The maximization function of interest written in complex notation is given by

$$\Lambda'(\tau, \theta, x_i) = \mathrm{Re}\left[\frac{1}{N_0}\int_{T_0}\tilde{y}(t)\exp(j\omega_c t)\left(\tilde{x}(t, \tau, \theta, x_i)\cdot\exp(j\omega_c t)\right)^* dt\right]. \tag{5.9}$$

Maximizing $\Lambda'(\tau, \theta, x_i)$ is done by determining the parameters τ_0 and θ_0. In practice only the estimates $\hat{\tau}_0$ and $\hat{\theta}_0$ are available. Assuming that these estimates are sufficiently accurate then maximizing $\Lambda'(\hat{\tau}_0, \hat{\theta}_0, x_i)$ is equivalent to choosing the sequence of x_i that maximizes

$$\begin{aligned} &\frac{1}{N_0}\sqrt{\frac{2E_b}{T_b}}\sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{x_k, x_i}\mathrm{Re}\left[\int_{iT_b+\tau_0}^{(i+1)T_b+\tau_0}\tilde{y}(t)\exp(j\omega_c t)\right. \\ &\left.\cdot\ \exp[-j(\omega_c t + x_k w_{i,k}(t - iT_b - \hat{\tau}_0) + \hat{\theta}_0 + \phi_i)]dt\right] \end{aligned} \tag{5.10}$$

where $x_k = +1, -1, +3, -3, ..., +(M-1), -(M-1)$ for $k = 1, 2, 3, ..., M$ and $\omega_{i,k} = \omega_{i,x_k}$. The real part of the integral can be physically realized by multiplying the received signal

$y(t)$, by the reference signals $\cos((\omega_c t + x_k w_{i,k}(t - iT_b - \hat{\tau}_0) + \hat{\theta}_0 + \phi_i)$, low pass filtering to remove the double frequency terms and integrating over the corresponding bit-period interval. By restricting the h-set to contain only rational values of denominator q, the phase ϕ_i will always be $l\pi/q$, where l is an integer. This restricts the number of correlations that need to be considered. By using the cosine summation identity $\cos(A + B) = \cos A \cos B - \sin A \sin B$, the partial branch metrics can be expressed as

$$\begin{aligned} BM(x_k\omega_{i,k}, l) &= \cos(l\pi/q) \int_{iT_b+\hat{\tau}_0}^{(i+1)T_b+\hat{\tau}_0} y(t) \cos(\omega_c t + x_k\omega_{i,k}(t - iT_b - \hat{\tau}_0) + \hat{\theta}_0)dt \\ &- \sin(l\pi/q) \int_{iT_b+\hat{\tau}_0}^{(i+1)T_b+\hat{\tau}_0} y(t) \sin(\omega_c t + x_k\omega_{i,k}(t - iT_b - \hat{\tau}_0) + \hat{\theta}_0)dt. \end{aligned} \tag{5.11}$$

The determination of these metrics can be done through the structure illustrated in figure 5-3.

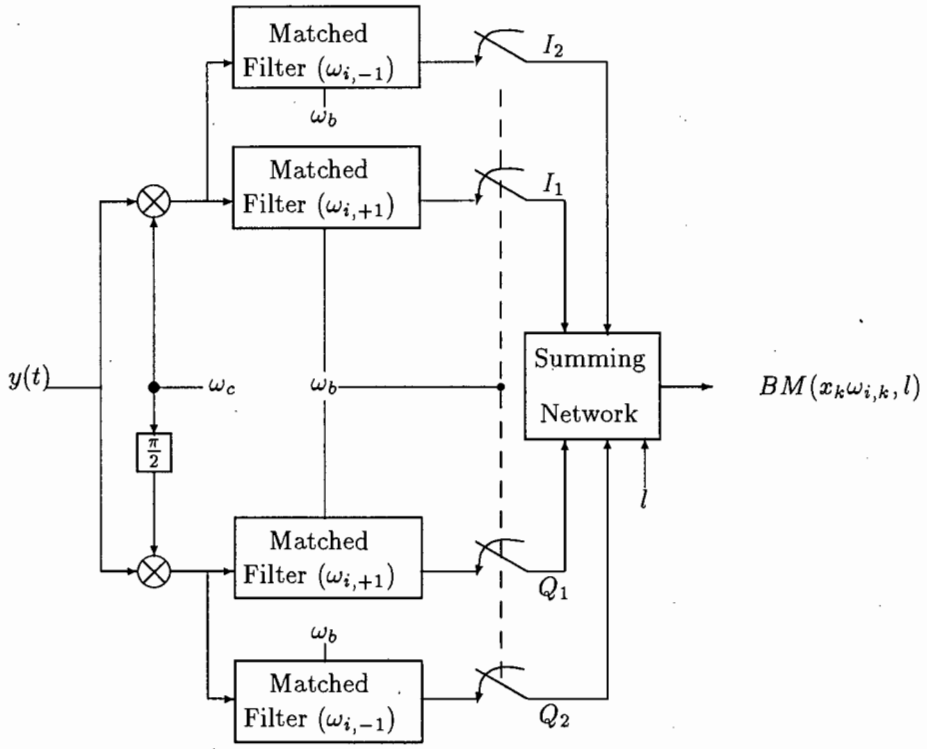

Figure 5-3: Binary Non-symmetric Multi-h CPFSK Demodulator Structure

The timing frequencies, ω_c, ω_b and the superbaud $n\omega_b$, are provided as byproducts of

the (error correction) decoder. There are two arms, an in-phase and a quadrature-phase arm, for each separate modulating frequency. These arms are characterized uniquely by the matched filters described by the set of impulse responses

$$\begin{aligned} \{H(h_i^{+1},t)\} &= \{\cos(\omega_{i,+1}(t-iT_b)),\sin(\omega_{i,+1}(t-iT_b))\} \\ \{H(h_i^{-1},t)\} &= \{\cos(\omega_{i,-1}(t-iT_b)),\sin(\omega_{i,-1}(t-iT_b))\}. \end{aligned} \tag{5.12}$$

The set $\{H(h_i^{+1},t)\}$ is used in the arms I_1 and Q_1 and the set $\{H(h_i^{-1},t)\}$ is used in the arms I_2 and Q_2 as shown in figure 5-3.

In the binary non-symmetric implementation under consideration, two sets of these arms are needed in each bit period, the same number as a 4-level symmetric implementation.

These in-phase and quadrature arms are responsible for evaluating the integrals in eq. 5.11. The summing network is responsible for combining these two integrals to form the branch metrics $BM(x_k\omega_{i,k},l)$ for any given l.

The summing network may be incorporated into the decoder to save hardware storage space as the decoder may need to recall only selected $BM(x_k\omega_{i,k},l)$. If this is the case, only the correlations I_K and Q_K need to be stored.

Demodulator Correlations

The demodulator correlations are described by eq. 5.11. This equation can provide further insight in choosing good Multi-h index sets. In this section, the in-phase and quadrature-phase terms are examined in greater detail. The following analysis assumes that the incoming signal in the i'th bit-period can be written as

$$y(t) = \cos\left(\omega_c t + x_i\omega_{i,i}(t-iT_b) + \phi_i(iT_b)\right) \tag{5.13}$$

assuming conditions of no noise and $\tau_0 = 0$ (perfect baud synchronization). Eq. 5.11 can be concisely expressed as

$$BM(x_k\omega_{i,k}, l) = \cos(l\pi/q)Y_I(x_k\omega_{i,k}, i) - \sin(l\pi/q)Y_Q(x_k\omega_{i,k}, i). \tag{5.14}$$

For brevity, the notation $\omega_S = \omega_{i,k}x_k + \omega_{i,i}x_i$ and $\omega_D = \omega_{i,k}x_k - \omega_{i,i}x_i$ is introduced. If the symbol expected is not the transmitted symbol, $(x_k \neq x_i)$, the unfiltered correlation components will be

$$\begin{aligned} Y_I(x_k\omega_{i,k}, i) &= \frac{A^2}{2} \cdot \frac{1}{2\omega_c + \omega_S} \sin\left[2\omega_c t + \omega_S \cdot (t - iT_b) + \phi_i(iT_b) + \theta_0\right]\Big|_{iT_b}^{(i+1)T_b} \\ &\quad + \frac{A^2}{2} \cdot \frac{1}{\omega_D} \sin\left[\omega_D \cdot (t - iT_b) + \phi_i(iT_b) - \theta_0\right]\Big|_{iT_b}^{(i+1)T_b} \end{aligned} \tag{5.15}$$

and

$$\begin{aligned} Y_Q(x_k\omega_{i,k}, i) &= -\frac{A^2}{2} \cdot \frac{1}{2\omega_c + \omega_S} \cos\left[2\omega_c t + \omega_S \cdot (t - iT_b) + \phi_i(iT_b) + \theta_0\right]\Big|_{iT_b}^{(i+1)T_b} \\ &\quad + \frac{A^2}{2} \cdot \frac{1}{\omega_D} \cos\left[\omega_D \cdot (t - iT_b) + \phi_i(iT_b) - \theta_0\right]\Big|_{iT_b}^{(i+1)T_b}. \end{aligned} \tag{5.16}$$

If the symbols are identical, $(x_k = x_i)$, then the correlation components will be

$$\begin{aligned} Y_I(x_k\omega_{i,k}, i) &= \frac{A^2}{2} \cdot \frac{1}{2\omega_c + \omega_S} \sin\left[2\omega_c t + \omega_S \cdot (t - iT_b) + \phi_i(iT_b) + \theta_0\right]\Big|_{iT_b}^{(i+1)T_b} \\ &\quad + \frac{A^2}{2} \cdot T_b \cos[\phi_i(iT_b) - \theta_0] \end{aligned} \tag{5.17}$$

and

$$\begin{aligned} Y_Q(x_k\omega_{i,k}, i) &= -\frac{A^2}{2} \cdot \frac{1}{2\omega_c + \omega_S} \cos\left[2\omega_c t + \omega_S \cdot (t - iT_b) + \phi_i(iT_b) + \theta_0\right]\Big|_{iT_b}^{(i+1)T_b} \\ &\quad - \frac{A^2}{2} \cdot T_b \sin[\phi_i(iT_b) - \theta_0]. \end{aligned} \tag{5.18}$$

The partial branch metrics are as follows: For $x_k \neq x_i$

$$BM(x_k\omega_{i,k}, l) = \frac{A^2}{2} \cdot \frac{1}{2\omega_c + \omega_S} \sin\left[2\omega_c t + \omega_S \cdot (t - iT_b) + \phi_i(iT_b) + \theta_0 + \frac{l\pi}{q}\right]\Big|_{iT_b}^{(i+1)T_b}$$

$$+\frac{A^2}{2}\cdot\frac{1}{\omega_D}\sin\left[\omega_D\cdot(t-iT_b)+\phi_i(iT_b)-\theta_0-\frac{l\pi}{q}\right]\Bigg|_{iT_b}^{(i+1)T_b} \tag{5.19}$$

and for $x_k = x_i$

$$\begin{aligned} BM(x_k\omega_{i,k},l) &= \frac{A^2}{2}\cdot\frac{1}{2\omega_c+\omega_S}\sin\left[2\omega_c t+\omega_S\cdot(t-iT_b)+\phi_i(iT_b)+\theta_0+\frac{l\pi}{q}\right]\Bigg|_{iT_b}^{(i+1)T_b} \\ &\quad +\frac{A^2}{2}\cdot T_b\cos\left[\phi_i(iT_b)-\theta_0-\frac{l\pi}{q}\right]. \end{aligned} \tag{5.20}$$

Consider if $\theta_0 = 0$ and the double frequency term to be insignificant, then the partial branch metric is

$$BM(x_k\omega_{i,k},l) = \frac{A^2}{2}\cdot\frac{1}{\omega_D}\left\{\sin\left[\omega_D+\phi_i(iT_b)-\frac{l\pi}{q}\right]-\sin\left[\phi_i(iT_b)-\frac{l\pi}{q}\right]\right\} \tag{5.21}$$

if $x_k \neq x_i$ and is

$$BM(x_k\omega_{i,k},l) = \frac{A^2}{2}\cdot T_b\cos\left[\phi_i(iT_b)-\frac{l\pi}{q}\right] \tag{5.22}$$

if $x_k = x_i$.

The demodulator's filter is partially responsible for the attenuation of the correlation described by eq. 5.21. By choosing large ω_D (signal frequency spacing), this correlation can be reduced. If ω_D is always an integral factor of 2π, then the correlation in eq. 5.21 is zero. This choice can be expressed as

$$p_k^{x_k}+p_i^{x_i} = \frac{2nq}{f_b} \quad \text{where } n\in\mathcal{Z}. \tag{5.23}$$

If the correct path is followed, the contribution to the accumulated Euclidean metric will be $A^2T_b/2$ per bit period as this is the coefficient in eq. 5.22. Due to the continuous nature of CPFSK and the choice of the h-sets, if a bit was incorrectly decoded, the following phase difference will be non-zero and thus result in a contribution of fraction of $A^2T_b/2$ per bit period until the correct phase is re-established. The maximum commencing slope of this wrong path will be twice the coefficient in eq. 5.21,

$$A^2\frac{T_bq}{\pi(p_i^{x_i}+p_k^{x_k})}. \tag{5.24}$$

5.2.2 Error Control Features

The quantity,

$$g(\tau,\theta,\hat{x}_k,\hat{\phi}_i) = \frac{2}{N_0}\int_{iT_b+\tau_0}^{(i+1)T_b+\tau_0} y(t)x(t,\tau,\theta,\hat{x}_k,\hat{\phi}_i)dt, \tag{5.25}$$

is determined by the demodulator. This function has maxima when $\tau = \tau_0$, $\theta = \theta_0$ and $\hat{x}_k = \hat{x}_i$. The maxima occurs when the two conditions,

$$\begin{aligned} \left.\frac{\partial\, g(\tau,\theta,\hat{x}_i,\hat{\phi}_i)}{\partial\theta}\right|_{\substack{\tau=\tau_0\\ \theta=\theta_0}} &= 0 \\ \left.\frac{\partial\, g(\tau,\theta,\hat{x}_i,\hat{\phi}_i)}{\partial\tau}\right|_{\substack{\tau=\tau_0\\ \theta=\theta_0}} &= 0 \end{aligned} \tag{5.26}$$

are simultaneously satisfied. Observe that there is a sign difference between τ and θ in expression of $x(t,\tau,\theta,\hat{x}_i,\hat{\phi}_i)$ (eq. 5.6). This means that the partial derivatives above have opposite signs on either side of their maxima and can therefore be used as error signals in a tracking loop. If the tracking loop bandwidth is smaller than the baud rate, these derivatives can be simplified as

$$\begin{aligned} \frac{\partial\, g(\tau,\theta,\hat{x}_i,\hat{\phi}_i)}{\partial\theta} = \epsilon_{i\theta} &= -\frac{2}{N_0}\sqrt{\frac{2E_b}{T_b}}\left[\hat{Y}_Q(\hat{x}_i\omega_{i,i},i)\cos(\hat{\phi}_i) + \hat{Y}_I(\hat{x}_i\omega_{i,i},i)\sin(\hat{\phi}_i)\right] \\ \frac{\partial\, g(\tau,\theta,\hat{x}_i,\hat{\phi}_i)}{\partial\tau} = \epsilon_{i\tau} &= \hat{x}_i\omega_{i,i}\frac{2}{N_0}\sqrt{\frac{2E_b}{T_b}}\left[\hat{Y}_Q(\hat{x}_i\omega_{i,i},i)\cos(\hat{\phi}_i) + \hat{Y}_I(\hat{x}_i\omega_{i,i},i)\sin(\hat{\phi}_i)\right] \\ &\quad +\text{Self Noise Term} \\ &\approx -\hat{x}_i\omega_{i,i}\epsilon_{i\theta}. \end{aligned} \tag{5.27}$$

These error signals can be generated in the decoder structure illustrated in figure 5-4 and can be used to construct the timing signals ω_c and ω_b as shown.

The error signals are constructed with current estimates $\hat{x}_i$ and $\hat{\phi}_i$ provided as a byproduct of the decoder. Consider the path being tracked in figure 5-5 as the most likely path based upon the decoder's observation from the root node at $(i-D)T_b$ until iT_b. It is very likely that the next correct branch will commence from the node of this path terminating at iT_b and thus a 'current' estimate $\hat{x}_i$, based on the observation from

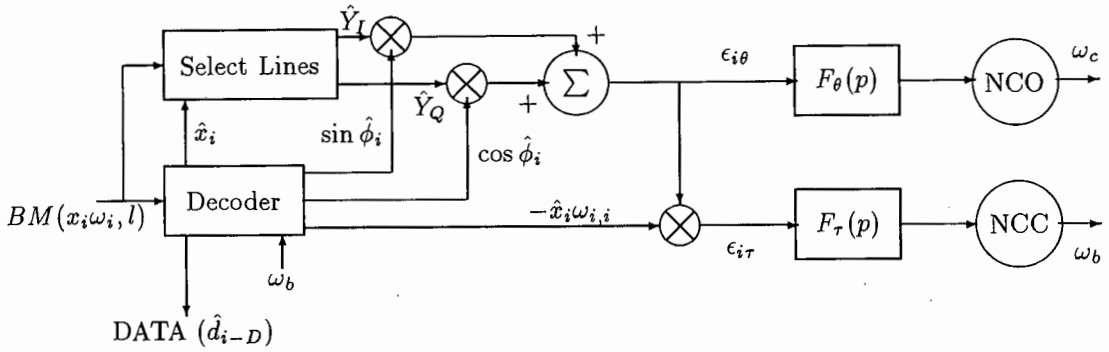

Figure 5-4: Premji-Taylor Timing Retrieval Structure

$iT_b \rightarrow (i+1)T_b$ can be made as illustrated.

$\hat{Y}_I(\hat{x}_i\omega_{i,i}, i)$ and $\hat{Y}_Q(\hat{x}_i\omega_{i,i}, i)$ are selected according to the estimate $\hat{x}_i$. The two error signals $\epsilon_{i\theta}$ and $\epsilon_{i\tau}$ are constructed by combining these two estimates with the estimate of $\hat{\phi}_i$. These error signals are fed into two filters and are subsequently utilized as the input of the NCO/VCO (Numerical Controlled Oscillator/Voltage Controlled Oscillator) and NCC/VCC (Numerical Controlled Clock/Voltage Controlled Clock). The demodulator shown here is for a numerical implementation using an NCO and NCC. The analogue equivalent will use a VCO and VCC instead. The carrier signal is generated by a NCO centered on the carrier frequency with the error voltage input $\epsilon_{i\theta}$ and the clock by an NCC centered at the symbol frequency with the error voltage input $\epsilon_{i\tau}$. The decoder outputs a delayed symbol stream $\hat{x}_{i-D}$ as the final output data.

Superbaud is the rate at which the matched filters in the demodulator are cycled. When this synchronization is not yet established, the h-values assumed in the receiver filters and the decoder are a cyclic permutation of the values used by the transmitter. This leads to incorrect projections of phase transitions ϕ_i. Phase state errors which occur as multiples of π/q radians represent large phase errors which cause the symbol synchronization error process to wander and cross one of the boundaries at $\pm T_b/2$ with high probability. The advancement or retardation of the error process will eventually result in an alignment of the cycle of h-values.

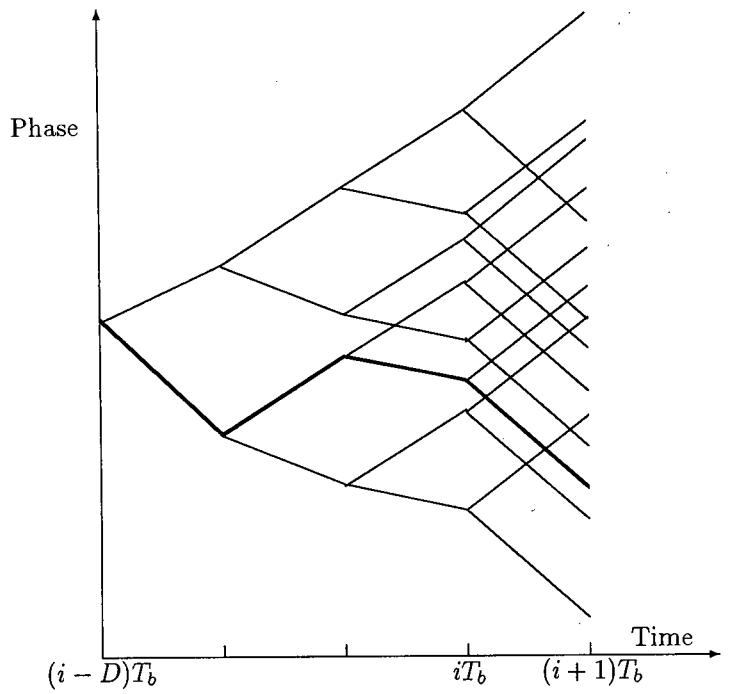

Figure 5-5: Snapshot of Path Recovery

Synchronization Structure Analysis

Premji and Taylor [5, 4] have performed an in depth analysis of the Synchronization Structure of their receivers by using the approach developed by Simon and Smith [22] for QASK receiver structures. Both of the loops which regenerate the timing information are examined.

The NCO loop is described by the equation

$$\dot{\theta} = K_{\nu\theta} F_\theta(p) e_\theta(t) \tag{5.28}$$

where $K_{\nu\theta}$ is the NCO gain in rad/V/s and $F_\theta(p)$ represents the loop filter response. The local estimate of the phase $\hat{\theta}_0(t)$ is given by $\theta(t) = \hat{\theta}_0(t) - \theta_e(t)$. This analysis assumes that the estimate $\hat{\theta}_0(t)$ is constant. This means that the NCO loop can be written as

$$\dot{\theta}_e = -K_{\nu\theta} F_\theta(p) e_\theta(t). \tag{5.29}$$

$\epsilon_{i\theta}$ in eq. 5.27 is used to substitute for $e_\theta(t)$. As the phase-error process varies con-

siderably slower than either the signal or noise processes, $e_\theta(t)$ can be replaced by it's statistical average.

Assuming accurate estimates of phase ϕ_i, the loop equation can be written as

$$\theta_e = -K_{\nu\theta} F_\theta(p)[G_\theta(\theta_e;\tau_e) + N_\theta(t)] \tag{5.30}$$

where $N_\theta(t)$ is approximately white Gaussian noise.

The carrier loop phase detector characteristic $G_\theta(\theta_e;\tau_e)$ in a binary non-symmetric Multi-h scheme is

$$\begin{aligned} G_\theta(\theta_e;\tau_e) &= \sqrt{\frac{E_b T_b}{2}} \sin(\theta_e) \left\{ \frac{(T_b - |\tau_e|)}{2nT_b} \sum_{x_k=\frac{-1}{+1},} \sum_{i=1}^{n} \cos(\omega_{i,k}\tau_e) \right. \\ &+ \frac{1}{4nT_b} \sum_{x_k=\frac{-1}{+1},} \sum_{x_j=\frac{-1}{+1},} \sum_{i=1}^{n} \left\{ (\delta_{\omega_{i,k},\omega_{i-1,j}} + \delta_{\omega_{i,k},\omega_{i+1,j}})|\tau_e| \cos(\omega_{i,k}\tau_e) \right. \\ &\left. \left. + \left[\frac{\omega_{i,k}(1 - \delta_{\omega_{i,k},\omega_{i-1,j}})}{\omega_{i,k}^2 - \omega_{i-1,j}^2} + \frac{\omega_{i,k}(1 - \delta_{\omega_{i,k},\omega_{i+1,j}})}{\omega_{i,k}^2 - \omega_{i+1,j}^2} \right] \sin(\omega_{i,k}|\tau_e|) \right\} \right. \end{aligned} \tag{5.31}$$

where $\delta_{i,j}$ is the Kronecker delta defined as

$$\delta_{i,j} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} . \tag{5.32}$$

The terms associated with Kronecker deltas arise due to certain adjacent $\omega_{i,j}$ being identical. In symmetric Multi-h, identical adjacent ω_i leads to premature merges and is undesirable. This is not the case in non-symmetric Multi-h where such a code may lead to large free Euclidean distances.

$G_\theta(\theta_e;\tau_e)$ is shown in figure 5-6 for the 4-h non-symmetric code found by the PBIL in chapter 6. The characteristic shown here does not exhibit as much curvature as the 4-level cases considered by Premji and Taylor. This is due to smaller signaling frequency spacing. The curvature is even less for larger h-sets, due to more closely packed signaling frequencies. Nevertheless the overall characteristic features of $G_\theta(\theta_e;\tau_e)$ are the same.

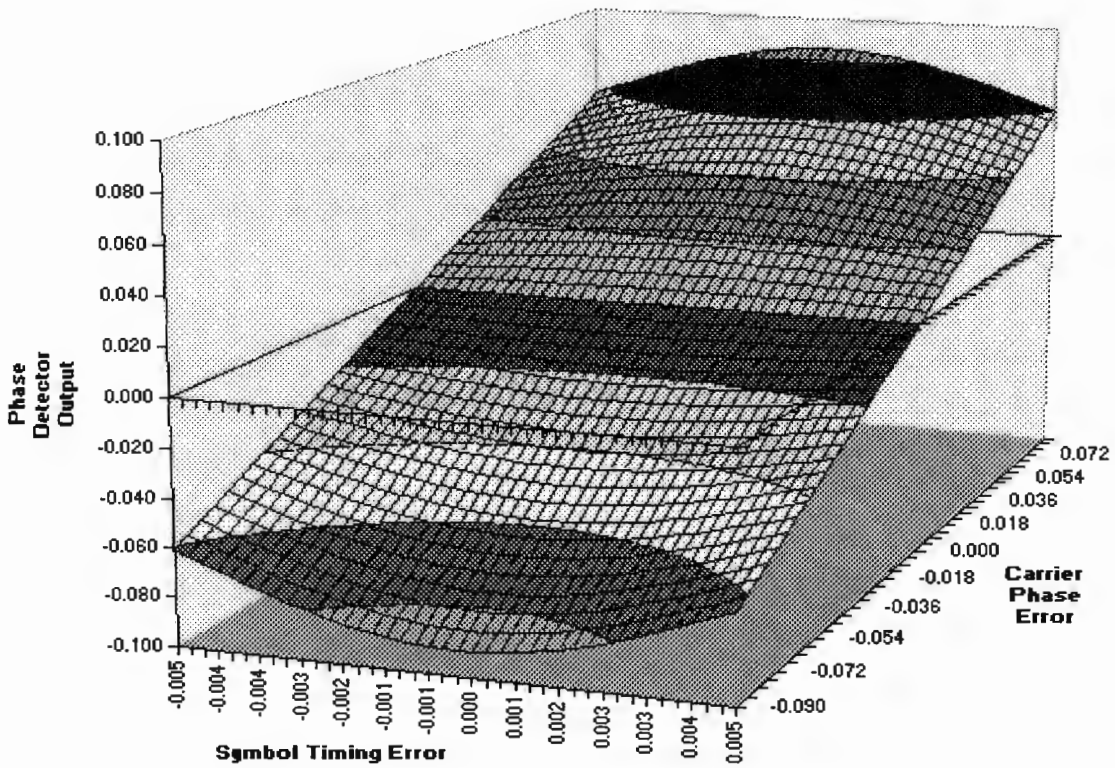

Figure 5-6: Carrier Recovery Loop Phase Detector Characteristic

The carrier phase characteristic exhibits the same sign change for small τ_e as the 4-level Multi-h case considered by Premji and Taylor [4]. The error signal has the same sign as the carrier phase. As long as the curved characterisitic of $G_\theta(\theta_e;\tau_e)$ dominates $N_\theta(t)$ in equation 5.30, the NCO loop has the ability to lock. Lessening the curvature of $G_\theta(\theta_e;\tau_e)$, as large non-symmetric h-sets do, dramatically reduces the locking ability of the loop. This is particular significant in conditions of low to moderate signal to noise ratios.

The demodulator does not distinguish between carrier phase errors that are multiples of π/q. The decoder chooses an estimate of phase $\hat{\phi}_i$ such that the magnitude of the residual phase error will be less than $\pi/2q$. Assuming a first order loop $F_\theta(p) = 1$, then by applying Fokker-Planck techniques to the loop equation, the probability density function of the modulo π/q reduced carrier phase error, is found to be

$$p(\theta_e) = C_\theta \exp[\mathrm{SNR}_{L\theta}(\cos(\theta_e) - 1)] \quad \text{where } |\theta_e| < \pi/2 \cdot q, \tag{5.33}$$

the normalization constant is given by

$$C_\theta = \left\{ \int_{-\pi/2q}^{\pi/2q} \exp[\mathrm{SNR}_{L\theta}(\cos(x) - 1)] dx \right\}^{-1} \tag{5.34}$$

and the loop SNR is given by

$$\mathrm{SNR}_{L\theta} = \frac{4E_b A_\theta(\tau_e)}{K_{0\theta} N_0}. \tag{5.35}$$

$K_{0\theta}$ is related to the Numerically Controlled Oscillator (NCO) gain $K_{\nu\theta}$ by $K_{0\theta} = \sqrt{E_b T_b/2} K_{\nu\theta}$. $A_\theta(\tau_e)$ is given by

$$A_\theta(\tau_e) = \sqrt{2/E_b T_b} G_\theta(\theta_e; \tau_e)/\sin(\theta_e). \tag{5.36}$$

In a similar fashion to the carrier frequency loop, the clock recovery loop can be characterized using the carrier synchronization error. The loop equation is given by

$$\dot{\tau}(t) = K_{\nu\tau} F_\tau(p) e_\tau. \tag{5.37}$$

Normalizing the timing error ($\eta_e = \tau_e/T_b$), the loop equation becomes

$$\dot{\eta}_e = -K_{\nu\eta} F_\eta(p)[G_\eta(\eta_e; \theta_e) + H_\eta^{1/2}(\eta_e) N_\eta(t)]. \tag{5.38}$$

$K_{\nu\eta}$ is the NCC gain, $N_\eta(t)$ is approximately white Gaussian noise and $H_\eta(\eta_e)$ is given by

$$H_\eta(\eta_e) = 1/2n \sum_{x_k=-1,+1} \sum_{i=1}^{n} (\omega_{i,k} T_b)^2. \tag{5.39}$$

The symbol synchronization loop phase detector characteristic is given by

$$\begin{aligned} G_\eta(\eta_e; \theta_e) &= \sqrt{\frac{E_b T_b}{2}} \cos(\theta_e) \left\{ \frac{(1 - |\eta_e|)}{2n} \sum_{x_k=\frac{-1}{+1},} \sum_{i=1}^{n} \omega_{i,k} T_b \cdot \sin(\omega_{i,k} T_b \eta_e) \right. \\ &+ \frac{|\eta_e|}{4n} \sum_{x_k=\frac{-1}{+1},} \sum_{x_j=\frac{-1}{+1},} \sum_{i=1}^{n} \omega_{i,k} T_b \cdot (\delta_{\omega_{i,k},\omega_{i-1,j}} + \delta_{\omega_{i,k},\omega_{i+1,j}}) \cdot \sin(\omega_{i,k} T_b \eta_e) \\ &- \left. \frac{\mathrm{sgn}[\eta_e]}{8n} \sum_{x_k=\frac{-1}{+1},} \sum_{x_j=\frac{-1}{+1},} \sum_{i=1}^{n} \omega_{i,k} (1 - \delta_{\omega_{i,k},\omega_{i-\mathrm{sgn}[\eta_e],k}}) \right. \end{aligned}$$

$$\cdot \left[(\omega_{i,k} - \omega_{i-\mathrm{sgn}[\eta_e],j})^{-1} + (\omega_{i,k} + \omega_{i-\mathrm{sgn}[\eta_e],j})^{-1} \right]$$
$$\cdot \left[\cos(\omega_{i,k} T_b \eta_e) - \cos(\omega_{i-\mathrm{sgn}[\eta_e],j} T_b \eta_e) \right] \Bigg\}. \tag{5.40}$$

$G_\eta(\eta_e;\theta_e)$ is shown in figure 5-7 for the same 4-h non-symmetric code as before. This characteristic, like the 4-level case, has no dependence on carrier phase error. As with $G_\theta(\theta_e;\tau_e)$, the characterisitic of $G_\eta(\eta_e;\theta_e)$ shown in figure 5-7 has less curvature than its 4-level counterpart. Larger non-symmetric h-sets reduce the curvature of $G_\eta(\eta_e;\theta_e)$ which in turn reduces the locking ability of the NCC loop. This is particularly notable in conditions of low to moderate signal to noise ratios as evident from equation 5.38.

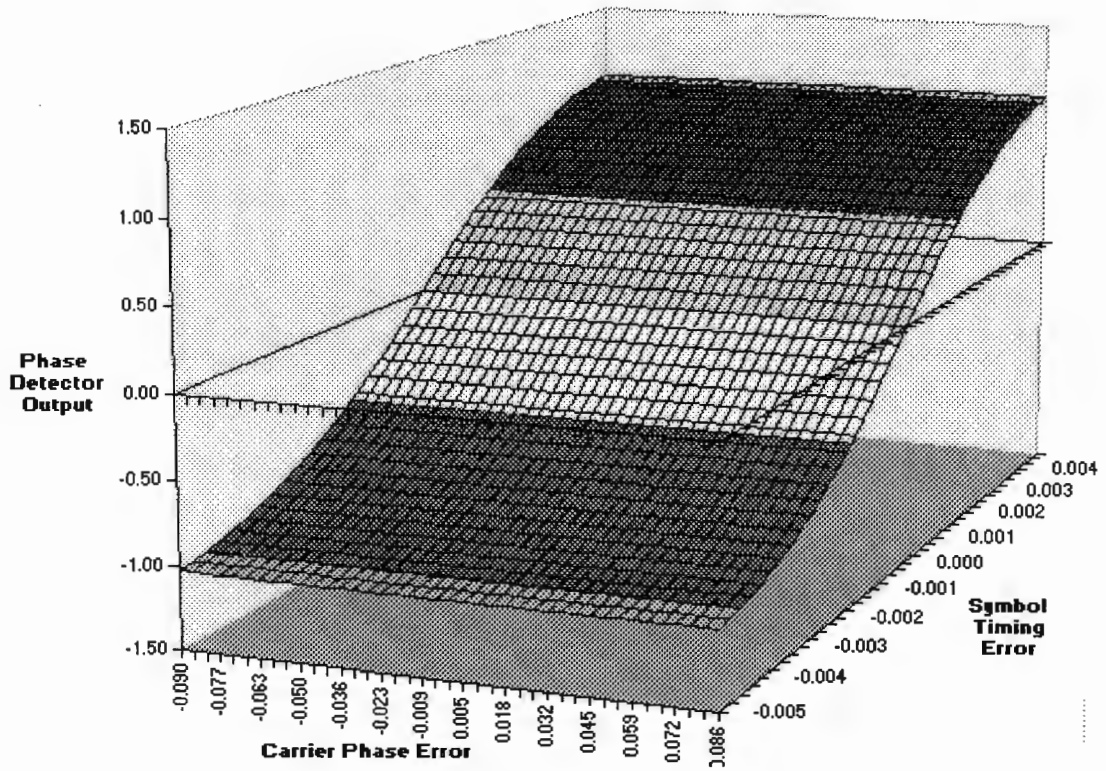

Figure 5-7: Clock Synchronization Loop Phase Detector Characteristic

Assuming a first order loop ($F_{\eta_e}(p) = 1$), the steady state probability density function of the reduced normalized clock error is

$$p(\eta_e) = C_\eta \exp[-\mathrm{SNR}_{L\eta} \cdot I_\eta(\eta_e)] \tag{5.41}$$

where

$$\mathrm{SNR}_{L\eta} = 4A_\eta(\theta_e)E_b/N_0K_{0\eta}, \tag{5.42}$$

$$I_\eta(\eta_e) = \int_0^{\eta_e} \frac{G_\eta(x;\theta_e)}{[\partial G_\eta(x;\theta_e)/\partial x|_{x=0}]}dx \tag{5.43}$$

and the normalization constant is given by

$$C_\eta = \left[\int_{-0.5}^{0.5} \exp(\mathrm{SNR}_{L\eta} \cdot I_\eta(x))dx\right]^{-1}. \tag{5.44}$$

In a similar fashion as before $K_{0\eta}$ is related to the NCC gain by

$$K_{0\eta} = \left[\frac{E_bT_b}{4n} \sum_{x_k=-1,+1} \sum_{i=1}^{n} (\omega_{i,k}T_b)^2\right]^{1/2} K_{\nu\eta}. \tag{5.45}$$

Poor synchronization can negate strong gains made through choice of codes with good squared free Euclidean distances. It is necessary to determine the characteristics given by $G_\theta(\theta_e;\tau_e)$ and $p(\theta_e)$ for the carrier loop phase and $G_\eta(\eta;\theta_e)$ and $p(\eta_e)$ for the symbol synchronization loop phase. It is necessary to ensure that $G_\theta(\theta_e;\tau_e)$ dominates $N_\theta(t)$ and $G_\eta(\eta;\theta_e)$ dominates $H_\eta^{1/2}N_\eta(t)$ through significantly large signal to noise ratios. This becomes a more difficult condition to satisfy when larger non-symmetric h-sets are used due to the reduced curvature of both $G_\theta(\theta_e;\tau_e)$ and $G_\eta(\eta;\theta_e)$.

Both $p(\theta_e)$ and $p(\eta_e)$ are dependent on the loop signal-to-noise ratios. The probability density functions of the 4-h non-symmetric Multi-h code and the 4-level 2-h code used by Premji and Taylor have identical characteristics in high signal to noise ratio conditions. Under these circumstances $p(\eta_e)$ is independent of carrier phase and $p(\theta_e)$ is strongly dependent on τ_e. Premji and Taylor observed that an increase in τ_e leads to greater steady-state jitter in the carrier recovery loop.

Chapter 6

Performance and Design Parameters

Multi-h CPFSK is a scheme which is conservative in frequency bandwidth and simple to implement. As with every communication scheme, it has benefits and drawbacks. It is necessary to know where the actual tradeoffs lie before making commitments in the choice of such a system. The purpose of this chapter is to examine these tradeoffs in detail, and to show methods of determining parameters that result in excellent performance.

This work has been based on the assumption that Multi-h has good spectral properties and inherent error coding. The performance of the error coding, while not as good as some Trellis Coded Modulation (TCM) schemes, shows strong gains over Minimum Shift Keying (MSK) and fixed-h CPFSK. The frequency bandwidth properties of Multi-h CPFSK are very similar to that of fixed-h CPFSK and generally superior to that of the *equivalent* TCM. Construction of Multi-h CPFSK is less hardware intensive than TCM and only slightly more complex than that of fixed-h CPFSK, due to the introduction of cyclic indices.

In high signal-to-noise ratio conditions, an optimal (error correction) decoder places greater limits on the throughput rate that the decoder can operate at than a sub-

optimal (error correction) decoder does.

In this chapter, investigation into the choice of h-sets that enhance the performance of sub-optimal decoders is undertaken. The measurement of the performance is based on the probability of error. In the first section of this chapter the determination of the probability of error is presented. The probability of error is shown to be dependent on the choice of the h-sets. In the second section, the PBIL algorithm is introduced as a method in selecting h-sets that produce good probability of error results. The influence of the sub-optimal decoder is elaborated on. Finally, the frequency power spectra of strongly performing h-set codes are shown.

6.1 The Probability of Error P_e

The classical distance between two signals is expressed as the Euclidean distance. In mathematical terms, the Euclidean distance between two differing signals $s(t,\alpha)$ and $s(t,\beta)$, both commencing at time $t = nT_b$ and lasting for NT_b is given by

$$D^2 = \sum_{i=n}^{n+N-1} \int_{iT_b}^{(i+1)T_b} [s(t,\alpha) - s(t,\beta)]^2 dt. \tag{6.1}$$

The different data sequences are $\alpha = \{\alpha_0, \alpha_1, \alpha_2, ...\}$ and $\beta = \{\beta_0, \beta_1, \beta_2, ...\}$ where α_i and β_i belong to $\{-M+1, -M+3, ..., M-3, M-1\}$. The sequences that are of interest are those that have phase paths on the phase trellis that diverge at some time, say $t = nT_b$ and only converge no sooner than $t = (n+N-1)T_b$. The distance between two such paths is known as the free distance. From this point on, D^2 will refer to the free Euclidean distances only. By considering all sets of paths that diverge and later converge, the minimum D^2 can be determined. For large signal-to-noise ratios (SNR), the minimum distance D^2_{min} dominates the error probability performance. The quantity D^2_{min} describes the closest proximity that two paths can have and the smallest free distance that the demodulator must be able to distinguish clearly. In a Gaussian channel having a one-sided noise power spectral density N_0, the probability of error is

given by

$$P_e \approx Q\left(\sqrt{\frac{D^2_{min}}{2N_0}}\right). \tag{6.2}$$

The function $Q(x)$ is closely related to the complimentary error function erfc(x) and is given by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-z^2/2} dz. \tag{6.3}$$

In general, one only considers the normalized squared Euclidean distance

$$d^2 = \frac{D^2}{2E_b} \tag{6.4}$$

where E_b is the energy per bit. Normalization varies amongst different researchers but as a standard, the minimum normalized squared Euclidean distance is referenced to that of MSK which has a normalized squared Euclidean distance $d^2_{min} = 2.0$. The probability of error is

$$P_e \approx Q\left(\sqrt{\frac{E_b \cdot d^2_{min}}{N_0}}\right). \tag{6.5}$$

6.1.1 Minimum Squared Euclidean Distance $d^2_{\min}$

The exact formula for the minimum squared Euclidean distance of fixed-h CPFSK has been determined by Rimoldi [23] and Ekanayake and Liyanapathirana [24]. The i^{th} bit-period component to the minimum Euclidean distance is

$$d_i^2 = \frac{1}{2E_b} \int_{iT_b}^{(i+1)T_b} [s(t, \alpha_i) - s(t, \beta_i)]^2 dt. \tag{6.6}$$

The corresponding Multi-h CPFSK signals, that need not have symmetric indices, are represented in general by

$$\begin{aligned} s(t, \alpha_i) &= \sqrt{\frac{2E_b}{T_b}} \cos\left\{\omega_c t + \frac{\pi x_i^\alpha h_i^\alpha}{T_b}(t - iT_b) + \phi_\alpha(iT_b)\right\} \\ s(t, \beta_i) &= \sqrt{\frac{2E_b}{T_b}} \cos\left\{\omega_c t + \frac{\pi x_i^\beta h_i^\beta}{T_b}(t - iT_b) + \phi_\beta(iT_b)\right\} \end{aligned} \tag{6.7}$$

during the period from $iT_b \rightarrow (i+1)T_b$. x_i^α is the i^{th} component of the α data stream, h_i^α is the h-index used in the i^{th} period and is dependent on the symbol x_i^α if non-symmetric indices are used. $\phi_\alpha(iT_b)$ is the signal phase at the commencement of the i^{th} bit-period. For conciseness in the following derivations, the above two equations are abbreviated as

$$\begin{aligned} s(t,\alpha_i) &= A\cos\{a(t)\} \\ s(t,\beta_i) &= A\cos\{b(t)\}. \end{aligned} \tag{6.8}$$

The i^{th} bit-period component is

$$\begin{aligned} d_i^2 &= A^2\int_{iT_b}^{(i+1)T_b}[\cos\{a(t)\}-\cos\{b(t)\}]^2dt \\ &= A^2\int_{iT_b}^{(i+1)T_b}\Big[1+\frac{1}{2}\cos\{2a(t)\}+\frac{1}{2}\cos\{2b(t)\} \\ &\quad -\cos\{a(t)+b(t)\}-\cos\{a(t)-b(t)\}\Big]\,dt. \end{aligned} \tag{6.9}$$

In general $\omega_c >> 1$. The middle three terms in the integral will have coefficients of $\approx 1/(2\omega_c) << 1$ after being integrated and are heavily dominated by the other two terms. They can therefore be ignored. The calculation of d_i^2, after re-substitution, reduces to [25]

$$d_i^2 = 2E_b\left[1-\frac{1}{\pi(x_i^\alpha h_i^\alpha - x_i^\beta h_i^\beta)}\left(\sin\left\{\pi(x_i^\alpha h_i^\alpha - x_i^\beta h_i^\beta)+\phi_\alpha-\phi_\beta\right\}-\sin\{\phi_\alpha-\phi_\beta\}\right)\right] \tag{6.10}$$

where $x_i^\alpha \neq x_i^\beta$ and

$$d_i^2 = 2E_b\left[1-\cos\{\phi_\alpha-\phi_\beta\}\right] \text{ where } x_i^\alpha = x_i^\beta. \tag{6.11}$$

These results are closely related to the correlation results given in eq. 5.21 and eq. 5.22.

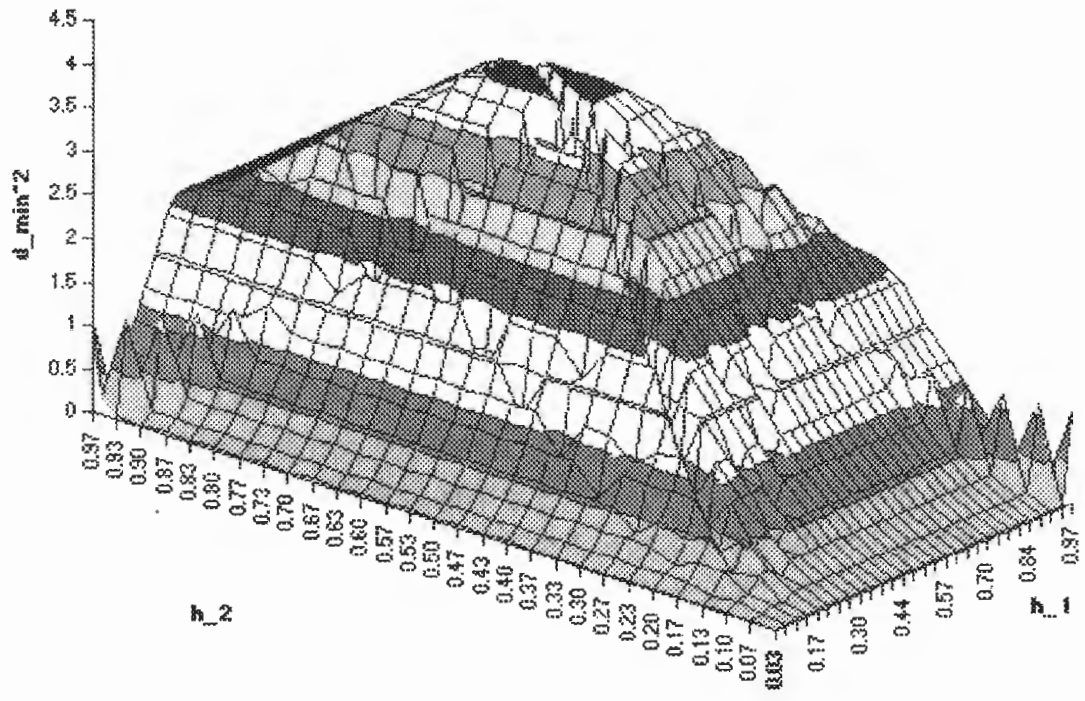

Figure 6-1: Minimum Squared Euclidean Distances for Symmetric 2-h CPFSK

6.1.2 Upper Bound Estimations of the d^2_{min}

The minimum Euclidean distance is the smallest d^2 found from all possible free distances where N ranges from the minimum *constraint length* to *infinity*. Hwang et. al. [26] only considers merges in which N is the constraint length of the code. Through the calculations undertaken here, this is seen not to be an accurate assumption. Aulin and Sundberg [25] consider a limited number of merging possibilities that have $N \geq$ constraint length to obtain closer d^2_{min} estimations than that of Hwang et. al.

By observing numerous free distance paths, it is seen that the upper bound becomes tightly bound in the vicinity of $N \approx$ constraint length. The upper bound of the minimum Euclidean distances are shown in figure 6-1 for a symmetric 2-h system.

There are inevitable merges for all binary Multi-h systems as the modulation set is finite[25]. Considering a 3-h set, an inevitable merge is illustrated in figure 6-2.

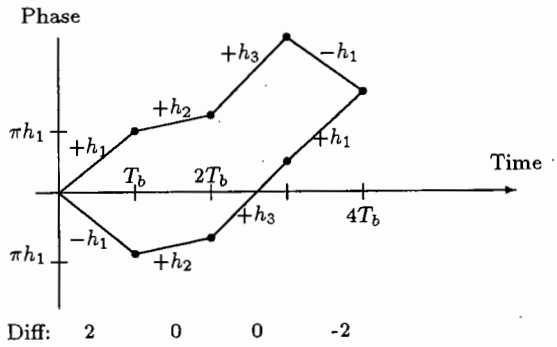

Figure 6-2: First Inevitable Merge for 3-h CPFSK

The distance for this figure is

$$d^2 = 2\left(1 - \frac{\sin 2\pi h_1}{2\pi h_1}\right) + 2(1 - \cos 2\pi h_1). \tag{6.12}$$

If a non-symmetric set is considered, the merge still occurs at $4T_b$ as illustrated in the diagram and the Euclidean distance is now

$$d^2 = 2\left(1 - \frac{\sin \pi(h_1^{+1} + h_1^{-1})}{\pi(h_1^{+1} + h_1^{-1})}\right) + 2(1 - \cos \pi(h_1^{+1} + h_1^{-1})). \tag{6.13}$$

Non-symmetric indices allow for greater minimum Euclidean distances than symmetric indices of the same denominator due to phase difference possibilities. The phase difference between two paths in conventional binary Multi-h is always an integral multiple of $2\pi/q$ while in non-symmetric Multi-h, the phase difference is an integral multiple of π/q.

Using either case, the *inevitable* first merge for any h-set of length l will be $(l+1)T_b$. If the elements of the h-set are chosen badly, it is possible to obtain a premature merge.

6.1.3 Error coefficients N_{d_l}

The approximation of P_e, given by equation 6.5, depends on the number of paths at a distance d_{min} from the correct path, $N_{d_{min}}$, being small.

The very nature of symmetric Multi-h CPFSK is to spread the free Euclidean distances

d_l apart unlike non-symmetric Multi-h CPFSK which tends to squash the d_l's towards each other. This is best illustrated by example. Consider the free Euclidean distances of the first 20 meeting places for each starting position in the two 7-h sets found by the PBIL algorithm (Table 6.2). These distances are listed in Table 6.1 below.

Range : ${d_l}^2 - {d_{min}}^2$	Symmetric N_{d_l}	Non-Symmetric N_{d_l}
0.00 - 0.05	16	18
0.30 - 0.35	15	20
0.35 - 0.40	1	-
1.15 - 1.20	-	16
1.20 - 1.25	4	-
1.65 - 1.70	-	20
1.80 - 1.85	-	39
1.85 - 1.90	-	17
2.55 - 2.60	16	-
3.35 - 3.40	40	-
3.45 - 3.50	20	-
3.50 - 3.55	15	-
5.00 ++	3	-
TOTAL	140	140

Table 6.1 : Lowest 140 free Euclidean distances for select 7-h Multi-h CPFSK

It is evident that the lowest free distances of the non-symmetric Multi-h CPFSK are far closer to the minimum squared Euclidean distance than in the symmetric case. To reduce the probability of error, it is desirable to maximize the quantity

$$\sum_{i=1}^{M} N_{d_i} \cdot ({d^2}_i - {d^2}_{min})$$

where M is the total number of free distance meeting places considered.

6.1.4 Exponential Error Bound R_0

Anderson and Taylor [37] calculated the error bound for Multi-h based on the cutoff rate R_0. The determination of R_0 is detailed in Anderson, Aulin and Sundberg's book [38]. Anderson and Taylor's calculation of R_0 was performed for Multi-h with low h-denominators (q). Their derivations are general and follow through to non-symmetric Multi-h with large q.

The parameter R_0 is defined as

$$R_0 \equiv -(1/N)\log_2 P_e \tag{6.14}$$

where N is the codeword length. The minimum codeword length that is used in the free Euclidean distance calculations is the constraint length of the code.

Anderson and Taylor [37] used the following theorem to calculate the parameter R_0 :
Theorem: For two codewords in a code chosen at random from the CPFSK ensemble of Multi-h codes having fixed q and codeword length N, N being a multiple of M,

$$P_e \leq \exp\left\{-N\left[-(1/M)\ln E[e^{-L_l E/4N_0}]\right]\right\}. \tag{6.15}$$

L_l is given by

$$L_l(s_1, s_2) \equiv \sum_{n=(l-1)M+1}^{lM-1} {d^2}_n(1,2), \tag{6.16}$$

s_1 and s_2 being codewords in the ensemble. The expectation value tends to the limit as follows:

$$\{E[e^{-L_l E/4N_0}]\}^{1/M} \rightarrow \frac{\eta_{max}}{q} \tag{6.17}$$

as $M \rightarrow \infty$. η_{max} is the largest eigenvalue of the matrix Λ, whose (i,j) element is given by

$$\Lambda_{ij} = \begin{cases} (1-c_{ij})\exp\left\{\lambda\left[2-2\left(\frac{\sin\Delta\phi_{n+1}-sin\Delta\phi_n}{\Delta\phi_{n+1}-\Delta\phi_n}\right)\right]\right\} \\ +c_{ij}\exp\left\{\lambda\left[2-2\left(\frac{\sin\Delta\phi_{n+1}-sin\Delta\phi_n}{\Delta\phi_{n+1}-\Delta\phi_n+2\pi}\right)\right]\right\}, i \neq j \\ \exp\left\{\lambda\left[2-2\cos\Delta\phi_n\right]\right\}, i = j. \end{cases} \tag{6.18}$$

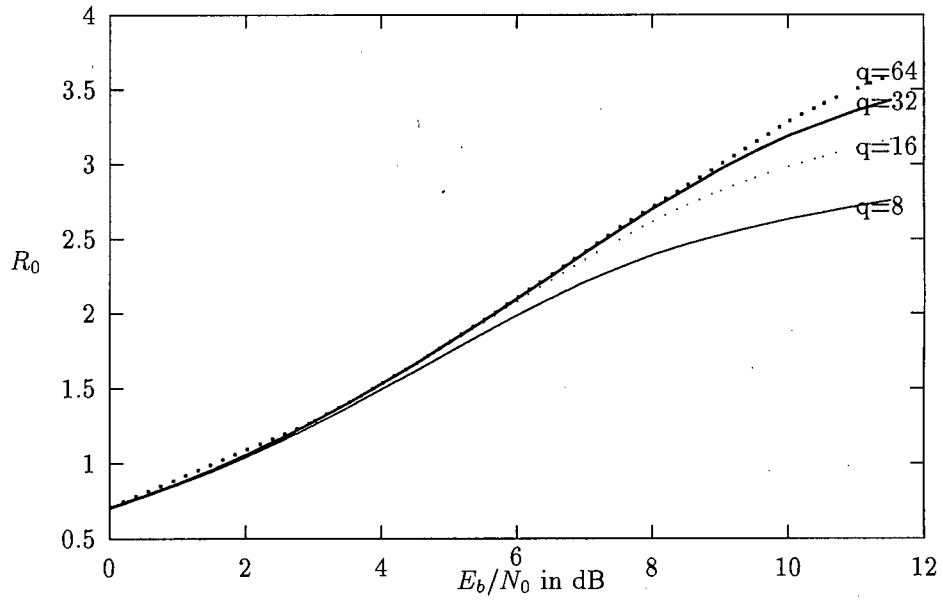

Figure 6-3: Error Parameter R_0 versus E_b/N_0 for CPFSK with various q.

Here

$$\begin{aligned} c_{ij} &= \frac{\Delta\phi_{n+1} - \Delta\phi_n}{2\pi} \\ \Delta\phi_{n+1} &= (j-1)2\pi/q \\ \Delta\phi_n &= (i-1)2\pi/q \\ \lambda &= -E/4N_0 \end{aligned} \tag{6.19}$$

where i and j represent the multiples of $2\pi/q$ at the beginning of the interval.

R_0 can be calculated directly from

$$R_0 = -\log_2(\eta_{max}/q). \tag{6.20}$$

The results are shown in figure 6-3 for select q. Note that R_0 converges at low E_b/N_0. These curves are practical upper limits on the information throughput of this signalling scheme and can be used to determine the upper bounds on P_e.

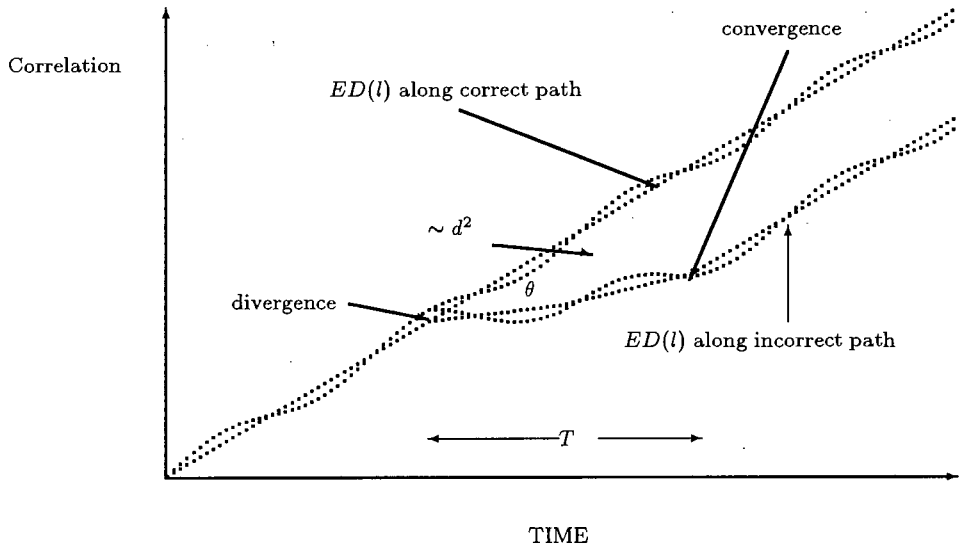

Figure 6-4: Multi-h Demodulator Correlation Trends

6.2 Search for Good h Index Sets

There are a number of matters that need consideration in the search for good h-sets. Obviously the h-set must illustrate strong minimum squared Euclidean distances. The decoder type should also be considered to further refine the choice, in particular how the decoder handles the demodulator's correlation outputs.

Demodulator Correlations

The likelihood of optimum detection in CPFSK is dependent on the measurement of correlations rather than Hamming distances. The accumulation of Hamming distances was illustrated in figure 4-5 for selected path pursuits. This equivalent correlation relationship is illustrated in figure 6-4.

The higher path represents the accumulated correlations found if the correct symbol stream is pursued by the demodulator. The lower path represents a path which diverges from the higher path at some point and later converges with the correct path.

The ability to quickly detect which path is being pursued is dependent upon the angle

θ illustrated and the time period indicated as T. The measure of this ability is closely related to the area between the two paths over the time period marked T. This area is quantified specifically by the *Euclidean Distance.*

The commencing slope can be maximized by maximizing the quantity

$$A^2 \frac{T_b q}{\pi(p_i^{x_k} + p_i^{x_j})} \tag{6.21}$$

for all h-indices where $x_k \neq x_j$.

Desired Features of the h-sets

Here, all the desired features of the h-index set are summarized

- The indices h_i must be confined to $0 < h_i \leq 1$ to create power efficient Multi-h CPFSK.
- The indices should be rational, ($h_i = p_i/q$,where $p_i, q \in \mathcal{Z}$) so as to make implementation manageable.
- The constraint length needs to be as long as possible. This maximizes the minimum T indicated in figure 6-4. The constraint length is bounded by the h-set size and q.

  The relationship between the constraint length and the h-set size, given that there is no premature merge, is : constraint length = h-set size + 1.

  The dependence of the constraint length on q can be seen by considering a binary scheme in which all paths diverging from an identical phase node are independent from one another for a period of kT_b. These paths will terminate on independent phase nodes only if there are 2^{kT_b} separate phase nodes available. If $q < 2^{kT_b}$, then there must be convergence between at least two paths as q places a bound on the number of phase nodes.
- Non-symmetric h_i allows for more parameter permutations enhancing the probability of finding better free minimum Euclidean distances.

- Certain maximizations on the numerators reveal stronger codes. Recall eq. 5.23 $(p_i^{x_k} + p_i^{x_j}) = 2nq/f_b$ and that the maximization of the sum $(p_i^{x_k} + p_i^{x_j})$ leads to enhanced error detection, where $x_k \neq x_j$.

J. Greene [27] has described the *Population Based Incremental Learning* (PBIL) method as fast, efficient and wide searching algorithm. This scheme is highly applicable to determining the optimal parameter sets of Multi-h CPFSK. Much investigation has been undertaken into determining specific variances of this method.

6.3 PBIL Algorithm

The *Population Based Incremental Learning (PBIL) algorithm* originates from the Genetic algorithm [28]. It works on the premise that through marrying two good parameter sets, one is likely to obtain a better parameter set. It is an efficient numerical method for approximating global optima of multidimensional functions.

In each generation, a number of sample parameter sets are generated. The parameter sets are written in such a way that no meaning is placed on the individual parameters used in the optimization process. The parameters are written as a string of binary digits (bits), each bit having a probability associated with it. The PBIL algorithm treats all bits in the parameter list equally and generates each sample based upon the information given in the associated probability vector. A fitness function is used to determine the performance of the multidimensional function given the parameter set. After each generation, the probability vector is incrementally adjusted increasing the probability of the previous generation's best parameter set being regenerated. Mutations are introduced so as to prevent the PBIL algorithm focusing on local maxima and thus widening the search.

The algorithm is as follows:

$P \leftarrow$ initialize probability vector. (Each position = 0.5)

loop # GENERATIONS

Generate Samples

$i \leftarrow$ loop # SAMPLES

$sample_i \leftarrow$ generate sample vector according to the probabilities in P

$evaluation_i \leftarrow$ evaluate $(sample_i)$

Find Best Sample

$max \leftarrow$ find vector corresponding to maximum evaluation

Update Probability Vector

$l \leftarrow$ #LENGTH

$P_i \leftarrow P_i \cdot (1.0 - LR) + max_i \cdot LR$

Mutate Probability Vector

$l \leftarrow$ loop #LENGTH

if $(random(0,1] < MUT\_PROBABILITY)$

$P_i \leftarrow P_i \cdot (1.0 - MUT\_SHIFT) + random(0.0 or 1.0) \cdot (MUT\_SHIFT)$

This algorithm is presented as a flowchart in figure 6-5. The constants are as following with the recommended values [27].

- $GENERATIONS$: number of iterations to allow learning.
- $SAMPLES$: the population size, number of samples to produce per generation
- $LENGTH$: length of encoded solution
- $MUT\_PROBABILITY$: probability of mutation occurring in each position [0.02]
- MUT_SHIFT: amount for mutation to affect the probability vector [0.05]
- LR: learning rate [0.05]

The variables are as follows:

- P: probability vector
- $sample_{i..SAMPLES}$: solution vectors
- $evaluation_{i..SAMPLES}$: evaluations of the solution vectors
- max: solution vector corresponding to maximum evaluation

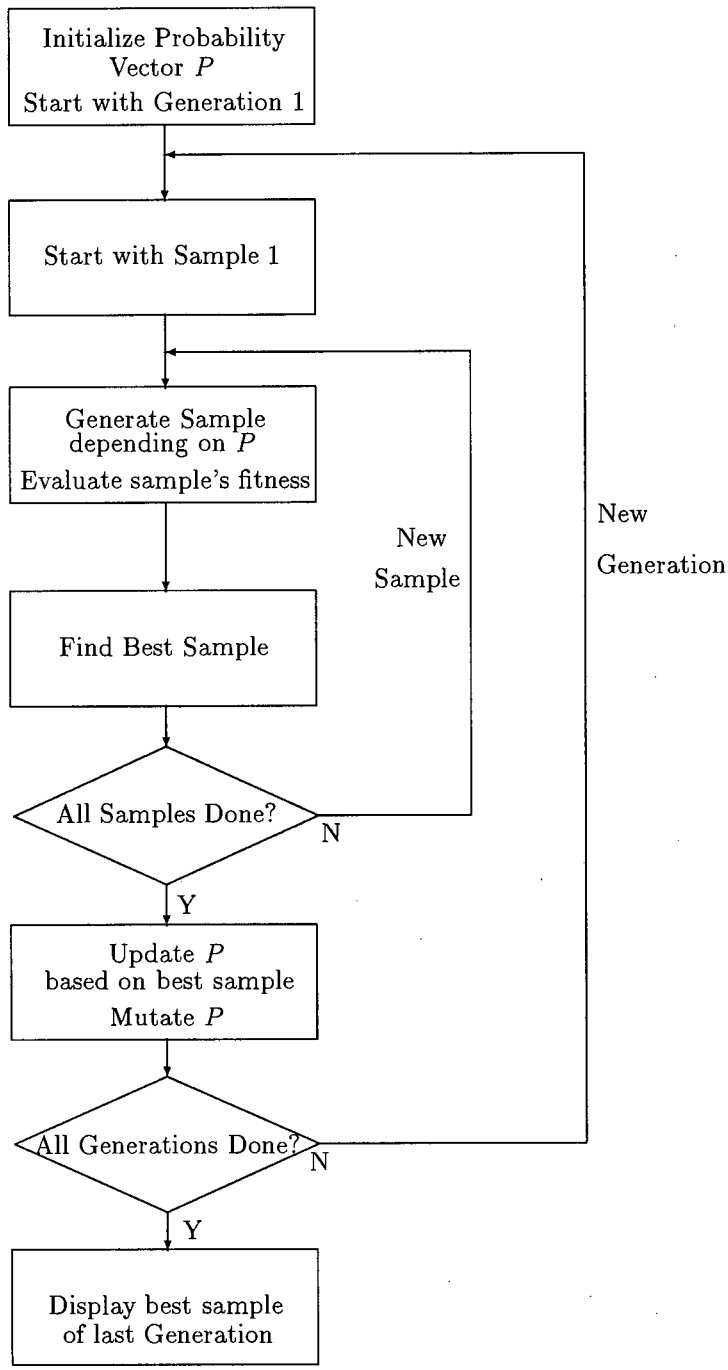

Figure 6-5: Flowchart of the PBIL Algorithm

6.3.1 Multi-h and the PBIL Algorithm

In the Multi-h CPFSK optimization process, the fitness function of interest is the Probability of Error. As $P_e \approx Q(d^2_{min} E_b/N_0)$, optimizing d^2_{min} is equivalent to optimizing P_e and can be used alternatively. d^2_{min} is dependent solely on the h parameter sets. The h parameter sets are placed in a parameter vector by decomposing the numerator into it's binary components and placing these parameters one after another. The number of digits representing each numerator is chosen by knowledge of the maximum numerator that can be used. For example, if the denominator is 16, the desirable range of the numerator will be from 1 to 15, thus four binary bits are required.

As an example of how this algorithm is utilized, consider a symmetric case searching for a two h-set with denominator 16. If the first random set generated is $\{13/16, 7/16\}$, it will be represented by the initial vectors

vector	h_1				h_2			
sample	1	1	0	1	0	1	1	1
P	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

In each generation, there will a number of samples generated like the one indicated above. The generation of the sample vector is based on the probability vector. Random numbers are generated between 0 and 1 for each element in the of the sample vector. If the random number is greater than the value held in the corresponding probability vector element, then the sample vector element generated is 1, else the element is 0.

d^2_{min} is calculated for each of these samples and is stored in $evaluation_i$. The best $evaluation_i$ amongst this set of samples is found. The vector P_i has it's individual members adjusted by an amount set by the learning rate (LR) and in the direction so as to increase the probability of best $evaluation_i$ re-occurring. Assume that the previous given sample to be the best sample and that the learning rate is 0.05. The probability vector is updated as

vector	h_1				h_2			
sample	1	1	0	1	0	1	1	1
P	0.45	0.45	0.55	0.45	0.55	0.45	0.45	0.45

in the next generation.

Note the direction in which the probability vector was adjusted. 0.45 allows for a greater chance of 1 being generated than 0. In this manner, when a local maxima is found, the PBIL algorithm homes in on the sample $evaluation_i$ vectors close to the maxima.

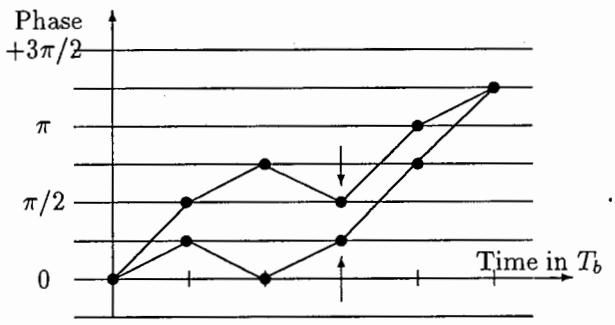

Figure 6-6: Event of a Premature Merge

Consider figure 6-6. If the two paths in the figure represent the paths of the soonest meeting place, adjusting only one of the parameters in the h-set can induce an earlier meeting place (*premature merge*). In the figure, this event is illustrated by convergence of the two nodes highlighted by arrows. This event results in a considerably lower d^2_{min}. If the same parameter list is adjusted further, these nodes will no longer merge and d^2_{min} will resume a value of order similar to that of paths without the premature meeting place.

The surface of the fitness function is relatively smooth with the exception of the case a premature meet as defined here. In a binary system, a meeting place will always come from the binary data streams $\{+1, +1, +1, .. + 1, -1\}$ and $\{-1, +1, +1, .. + 1, +1\}$, each stream being of identical length $n + 1$, and n is the size of the h-set. It makes sense to discard all of the parameter sets that lead to premature meets. In this manner, all the

desired features of admissible h-sets are incorporated into parameter selection.

Some PBIL implementations use a negative learning rate. The negative learning rate is used in the same manner that the learning rate is used but is based on the worst *evaluation* of the current generation. It is used to decrease the probability of reproducing the worst parameter vector again. Under the circumstances used here, it is not applicable, notably as the parameter set used is cyclic.

To maintain a wide area search, the search must be 'defocussed' from the current probability vector direction. The main method is the introduction of bitwise mutations performed on the probability vector. By generating a random number between 0 and 1, if the result is below a mutation probability threshold, then the probability vector is adjusted by a mutation rate in a similar fashion to the learning rate.

Another method of keeping the search 'defocussed' is to decrease the learning rate, the further the distance that the probability vector parameter is from 0.50. In other words, an 'asymptotic' learning rate is used instead of a 'linear' learning rate. Implementation can be accomplished by scaling the constant learning rate by the factor of $(0.50 - |Prob_i - 0.50|)$.

6.3.2 Some Good Sets Found Using the PBIL Algorithm

The correlator must provide an output that will allow both phase and frequency to be clearly identified. Obviously the smaller the denominator of the h's, the greater the resolution needed. Higher h-denominators do however allow for better Euclidean distances.

If the definition ability is 0.5°, then q = 360 and the largest possible constraint length will be 8 as $2^8 < 360 < 2^9$. The results obtained are given in Table 6.2.

Set size	q	p_i^{+1} p_i^{-1}	d^2_{min}	Gain (dB) over MSK
		p_i (Symmetric)		
3	360	331 184 60 57 208 331	6.083	4.831
3	360	195 194 197	6.083	4.831
4	360	79 84 213 100 302 304 169 286	8.054	6.050
4	360	182 202 207 200	7.924	5.979
5	360	150 196 49 18 227 257 208 359 342 160	9.883	6.938
5	360	207 176 198 209 194	9.764	6.886
6	360	316 253 211 90 328 206 56 123 199 308 83 211	11.656	7.655
6	360	180 178 207 160 164 224	10.194	7.073
7	360	18 114 148 114 142 102 337 340 287 278 246 311 180 43	12.151	
7	360	249 191 246 217 195 175 180	10.580	7.234
8	360	18 114 148 114 250 129 102 337 340 287 278 246 20 215 180 43	12.00	7.782

Table 6.2 : Optimal h-index Sets Found Using the PBIL Algorithm

In searching for good h-sets, it has been found that the PBIL converges quickly with a learning rate of 0.05. It must be borne in mind that a cyclic permutation of an h-set can produce an identical signal. The PBIL algorithm will home in on a particular random permutation of a good set in any case. The surface of the fitness function is not altogether smooth. In a catastrophic case, by changing an intermediate parameter in the current set, an earlier merge and thus considerably lower distance is induced. This is handled through the rejection of low constraint length measurements and large populations per generation.

The probability of error for all non-symmetric h-sets in Table 6.2 and for MSK are plotted in figure 6-7.

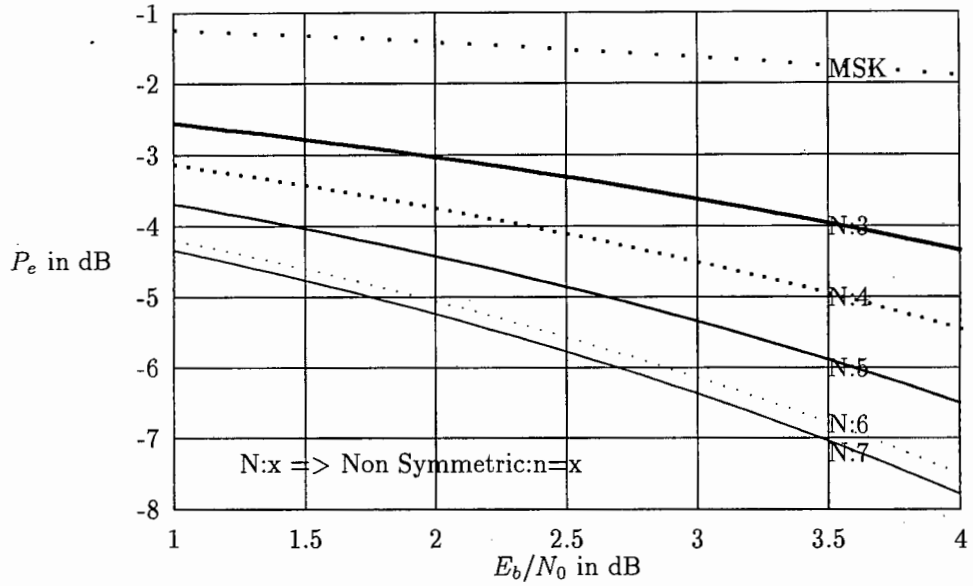

Figure 6-7: Probability of Error Curves for Non-Symmetric h-sets and MSK

6.4 The Fano Decoder's Influence on the Probability of Error

Under high signal-to-noise conditions, the performance of the Fano decoder is equal to that of the Viterbi decoder. However, under low signal-to-noise conditions the Fano decoder suffers from catastrophic failure. The resulting effect on the probability of error P_e is shown as a function of the signal-to-noise ratio (SNR) in figure 6-8.

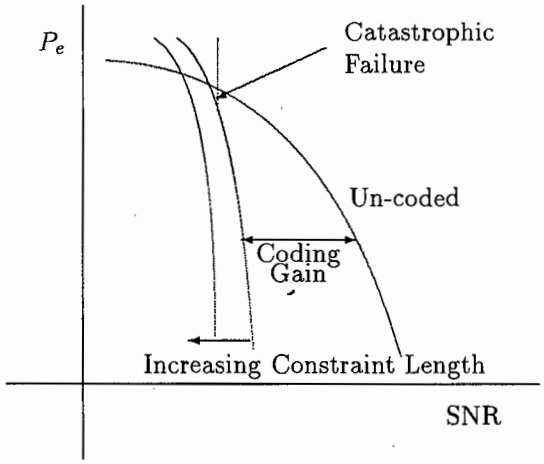

Figure 6-8: Coding Gain Characteristics

The parameters of the Fano decoder that strongly influence the probability of error are the discrete threshold spacing Δ and the truncation depth k_T. Both these quantities play an important role in the phenomena of catastrophic failure.

The decoder makes decisions based upon comparisons between the accumulated correlations of various signal paths found on the phase trellis with the incoming signal. The phase trellis paths are chosen from a set of paths, each path having some likelihood of having been the actual path followed. The finite truncation depth decoder will consider likely paths for the section of the phase trellis commencing k_T before the period iT_b currently being demodulated and terminating at the period iT_b. The decoder will output the path segment estimate for the period $iT_b - k_T \rightarrow iT_b - k_T + T_b$ when the decoder's focus is moved on by a bit-period. The output of the path segment has the effect of assuming that all likely paths have a common original node at the time $iT_b - k_T$. The larger k_T is, the greater the accuracy of this assumption becomes.

The larger k_T is, the greater the dependence the probability of error has on free distances between phase paths diverging and converging from one another within a period of time no greater than k_T. In particular, the probability of error will be dominated by the minimum squared Euclidean distance. In determining the conditions that promote catastrophic failure, the influence of Δ on the paths associated with the minimum squared Euclidean distance is considered to be a major factor. Firstly consider the following two Δ cases. The threshold spacing Δ is normalized according to the E_b/N_0 ratio that the decoder has been designed to perform for.

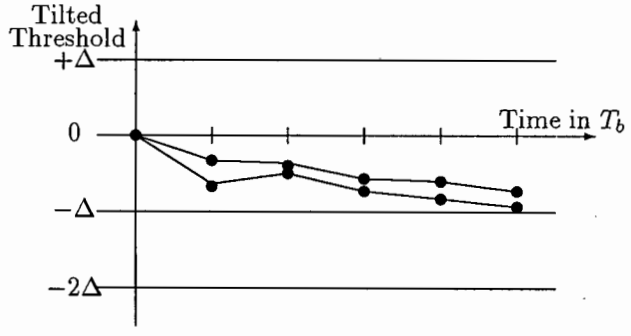

Figure 6-9: Large Δ Parameter

- **A Large Δ Parameter** : If Δ is too big, the decoder will not be able to distinguish between two nearby paths. The truncation depth is generally chosen to be larger than the depth needed for two paths associated with the minimum free distance to diverge and converge again. The paths become undistinguishable if the inequality

$$\frac{E_b}{N_0} d^2_{\min} < \Delta \tag{6.22}$$

is satisfied. This is obviously dependent on noise and will lead to catastrophic failure of the Fano decoder when this equality is valid.

- **A Small Δ Parameter** : If Δ is too small, the decoder will need to make many adjustments to the threshold to move to the correct path if the incorrect path has been chosen. Following Lin and Costello's [14] approach, the decoder is able to make μ decisions in $k_T T_b$ before the resources of the decoder are fully saturated.

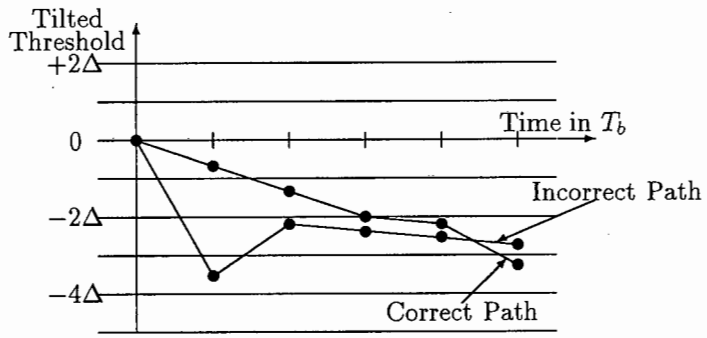

Figure 6-10: Small Δ Parameter

Consider figure 6-10 in which the incorrect path commences by yielding more attractive accumulated correlations than the correct path. Once the decoder discovers that the path followed is incorrect path, it must backtrack to follow the correct path. If the decoder chooses the erroneous path, at best it will need to investigate two paths at a time on it's backwards movement towards the correct path. Assuming the threshold to be very tight, the decoder will need at most $2 \cdot (\frac{E_b}{N_0} d^2/\Delta)$ investigations before the correct path can be reached. d^2 is the free euclidean distance between the two paths. If d^2_{min} dominates the probability of

error, the number of investigations that satisfy the inequality

$$2 \cdot (\frac{E_b}{N_0} d^2_{\min}/\Delta) > \mu, \tag{6.23}$$

will possibly lead to erasure.

Δ will be chosen in a situation under which good signal-to-noise ratios, being limited by μ. This enforces a lower signal-to-noise ratio that the decoder will be able to handle. For high signal-to-noise ratios, this type of error is improbable.

An example is used to illustrate the effects of different decoder speeds and threshold spacings.

Example:

- *Suppose that the channel conditions allow for a signal-to-noise ratio of* $E_b/N_0 = 10^6$.
- *The h-parameter set chosen has the coding properties* $d^2_{\min} = 10.0$ *with the length* $k = 5$ *being associated with the* $d^2_{\min}$.
- *The decoder can perform* 10 *calculations per bit period and has a truncation depth* k_T *of 10 (*μ *is 100). It is decided that given the present channel conditions,* Δ *is chosen so that there are 1000 divisions between the paths identifying the minimum squared Euclidean distance. The number of divisions are given by* $E_b/N_0 \cdot d^2_{\min}/\Delta = 1000$ *which yields* $\Delta = 10^4$.

Solution:

- *Consider the small* Δ *condition. If an error similar to that occurring in figure 6-10, and the number of divisions are* 1000, *the number of computations required, as determined by the left side of eq. 6.23 is* 2000. *As* $2000 > 100$, *eq. 6.23 is satisfied, there is a probability of catastrophic failure. The probability of errors occurring that lead to the condition illustrated in figure 6-10 is diminished by increasing the ratio* E_b/N_0, *however, it becomes more difficult to escape catastrophic failure if such an error pattern occurs.*

- *Suppose a faster decoder having $\mu = 2000$ can be obtained. Consider the large Δ condition. The inequality in eq. 6.22 is satisfied when $E_b/N_0 < 10^3$. The decoder will not be able to function when $E_b/N_0 < 10^3$.*
- *If a faster decoder cannot be obtained, the number of divisions must be reduced. Δ is found by finding the boundary condition (equality) in eq. 6.23. This occurs when $\Delta = 2 \cdot 10^5$. The inequality in eq. 6.23 is now satisfied when $E_b/N_0 < 2 \cdot 10^5$.*

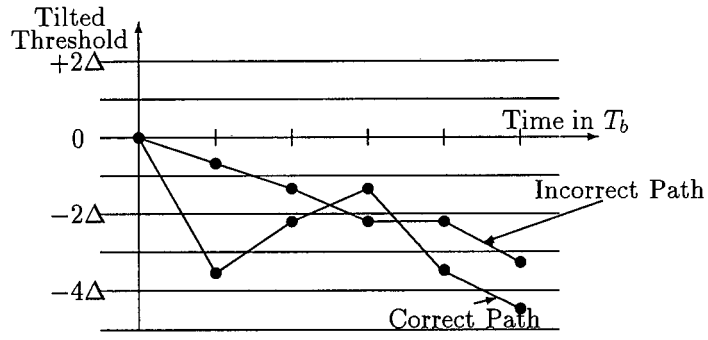

Figure 6-11: Premature Truncation Depth

- **Truncation Depth :** Wozencraft [11] considered a particular situation which results in incorrect decisions made by a sequential decoder. Due to the truncation of the feasible search back, a sequential decoder has a probability of selecting the incorrect path. This occurs if (a) the tilted metric of the correct path is greater than the mistaken path at some time n_0T_b where $l < n_0 < l + k_T$, l being the period where the two paths diverge and the k_T is the decision depth, (b) the incorrect path does not violate the threshold in the time period between n_0T_b and $(l+k_T)T_b$, the threshold having been weakened due the correct path's tilted metric at time nT_b, and (c) k_T is smaller than the depth needed for the two diverging under consideration to merge. If the decoder manages to detect the error at the end of the observation period k_T, the decoder will have to make the number of threshold moves given by the average of the drops in threshold levels of both the incorrect path and correct path in order to move to the correct path.

  This case is remotely similar to that of the small Δ case.

The approximation that is used here in determining catastrophic failure is that described by the large Δ case. With long k_T and high signal-to-noise ratios, the latter two cases dominate the probability of catastrophic failure and are small. To obtain more accurate estimates of catastrophic failure, these models need to be considered.

6.5 Power Spectra

There are a number of different methods used to calculate the power spectra of Multi-h CPFSK. According to Sasase and Mori [1], most of these methods fall into three broad categories. These are the 'direct' approach [2], the Markhov chain method [21] and the autocorrelation function approach [29]. These methods are discussed in greater depth by Crawford [7]. Under this work, the different methods will be briefly outlined and only the simulation method will be used to calculate certain spectra of interest. It must be borne in mind that these methods have been designed with symmetric h-index sets being of interest and amendments should be made to account for the non-symmetric nature of the h-sets used. Crawford has discussed the merits of using a simulation to produce Multi-h spectra. It has been shown that the time-varying nature of Multi-h has the effect of smearing the spectrum and filling up spectral nulls.

6.5.1 The Autocorrelation Function Method

The power spectrum is computed by the numerical Fourier transformation. The autocorrelation of Multi-h CPFSK is obtained as a simple closed form expression by using the characteristic function of the probability density of the phase change variable [30]. Lereim investigated the spectral properties of Multi-h signals extensively using the autocorrelation function method and the Markov chain method.

6.5.2 The Markhov Chain Approach

Multi-h intrinsically has a phase trellis with q phases that vary periodically at the superbaud rate n. This means that the signal can be described as a Markhov chain amongst nqM states. In the Markov chain approach, numerical computations are not generally required however care is required to the state modelling of the state modelling of the Multi-h code. The computational complexity grows exponentially with M and L.

All possible signal states and transitions are equiprobable over the super-interval $T' = qT_S$. If the signal probabilities are represented by p_i and p_j and the transition probability by p_{ij} where i and j denote two separate symbol periods, $j > i$. The autocorrelation function of these signals is given by

$$r_{ij}(t) = \begin{cases} \sum_{\tau=1}^{nqM} p_i \cdot p_{ij}(j-1) \cdot s_i^*(\tau - iT_s) \cdot s_j(t - jT_s) \text{ if } i \neq j \\ \sum_{\tau=1}^{nqM} p_i \cdot s_i^*(\tau - iT_s) \cdot s_j(t - jT_s) \text{ if } i = j \end{cases} \quad (6.24)$$

The resultant expression for the power spectrum is derived from the Fourier transform of the auto-correlation functions just described [21].

$$G(f) = \frac{1}{T_S} \sum_{j=1}^{nqM} \sum_{k=1}^{nqM} \sum_{l=-\infty}^{\infty} \frac{1}{nqM} \exp(i2\pi f l T_s) P_{jk}^l S_j^*(f) S_k(f) \quad (6.25)$$

where P_{jk}^l denotes the jk element of the state transition matrix P raised to the l^{th} power. $S_j(f)$ denotes the Fourier transform of the low pass complex envelope of the j^{th} waveform among the set of N state waveforms.

6.5.3 The 'Direct' Approach

The power spectrum is obtained by Fourier transforming a finite time segment and taking the expectation value with respect to the random phase and data sequence. This method has the capability to compute the exact spectral density functions numerically.

Wilson and Gaus treated the Multi-h signal as an M^K-ary signal over the time KT_s and determined spectrums for various M-ary full response M-ary Multi-h schemes.

The one-sided low-pass spectrum given by Wilson [2] is

$$G(f) = \frac{2}{nT_b}\left[P(f) + 2\mathrm{Re}\left\{F(f)F_b^*(f)\exp(-j2\pi fnT_b) + F(f)F_b^*(f)\Lambda(f)\right\}\right] \quad (6.26)$$

where

$$\Lambda(f) = \exp(-j4\pi fnT_b)\frac{C(nT_b)}{1 - C(nT_b)\exp(-j2\pi fnT_b)}. \quad (6.27)$$

The short hand notation is

$$\begin{aligned} b_k(t) &= \pi\int_0^t \left[d_{1k}h_1g(\tau) + d_{2k}h_2g(\tau - T_b) + ... + d_{nk}h_ng(\tau - nT_b)\right]d\tau, k = 1, 2, ..., 2^n \\ &\equiv \pi\int_0^t f_k(\tau)d\tau, k = 1, 2, ..., 2^n \quad (6.28) \\ F_k(f) &= \int_0^{nT_b} \exp(-j2\pi ft + jb_k(t))dt \quad (6.29) \\ C(nT_b) &= E\{\exp[jb(nT_b)]\} \quad (6.30) \\ F(f) &= E\{F_k(f)\} = 2^{-n}\sum_k F_k(f) \quad (6.31) \\ F_b^*(f) &= E\{F_k^*(f)\exp[jb(nT_b)]\} = 2^{-n}\sum_k F_k^*(f)\exp[jb(nT_b)] \quad (6.32) \\ P(f) &= E\{|F_k(f)|^2\} = 2^{-n}\sum_k |F_k(f)|^2 \quad (6.33) \end{aligned}$$

where E denotes the expectation operator. This method has been considered to be the most accurate one.

6.5.4 Simulation

A pseudo-random data sequence is generated and is used to generate the modulation signal. A numerical encoded Voltage Controlled oscillator is used to generate the signal through choice of the gain factor and by scaling the data sequence according to a preset pattern determined by the h-set. A Fast Fourier Transform (FFT) is performed on a

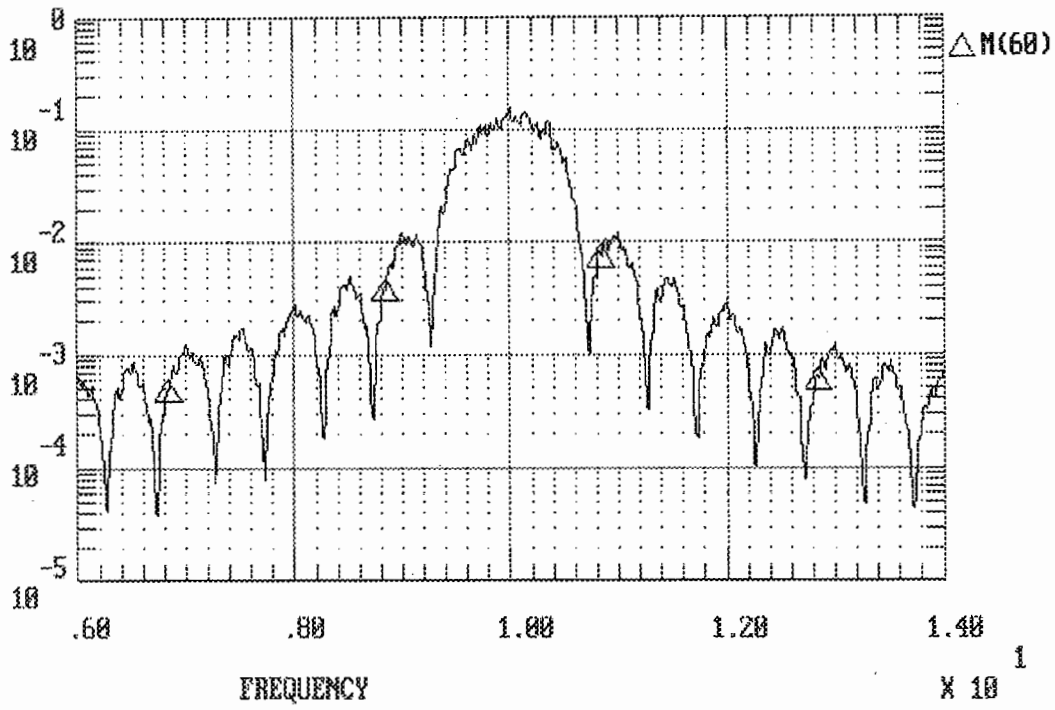

Figure 6-12: 3-h Symmetric Multi-h, $f_c = 10$Hz, $f_b = 1$Hz,

time simulation of the signal to compute the spectral components directly.

This method approaches the exact signal spectrum only if the data sequence is long and perfectly random. The transient spectra introduced by the digital sampling of the continuous time signal will diminish asymptotically in the limit $\omega \to \infty$. As the Fourier transform of the sampled sequence is a superposition of an infinite number of shifted Fourier transforms of the unsampled sequence scaled by $1/T$, an important consideration is that of aliasing. If spectral estimates are desired several times the symbol rate, a sampling rate of sixteen times per symbol should suffice [2].

Consideration must also be paid to the nature of the pseudo-random sequence. Only in the limit $t \to \infty$ is each binary state in a data stream equiprobable. The power spectra of various Multi-h schemes selected from the h-sets in Table 6.2, have been determined using the simulation package TESLA and are reproduced in figures 6-12 to 6-18. The y-axes have logarithmic scales of the amplitude (in V) and the x-axes have linear scales in Hertz.

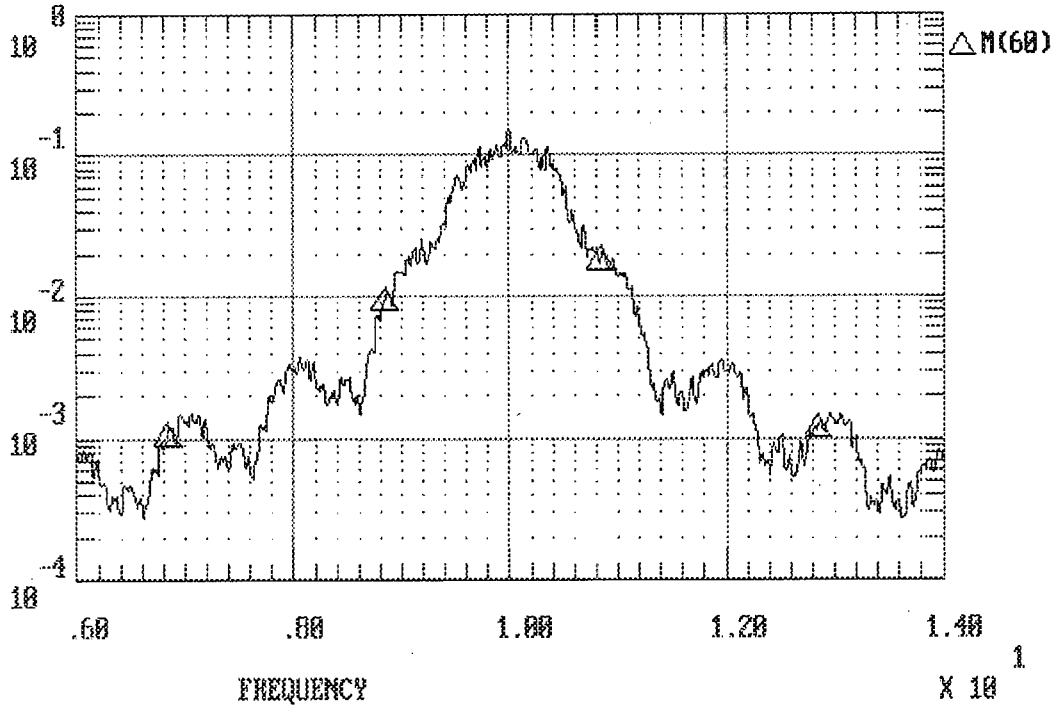

Figure 6-13: 3-h Non-Symmetric Multi-h, $f_c = 10\text{Hz}$, $f_b = 1\text{Hz}$,

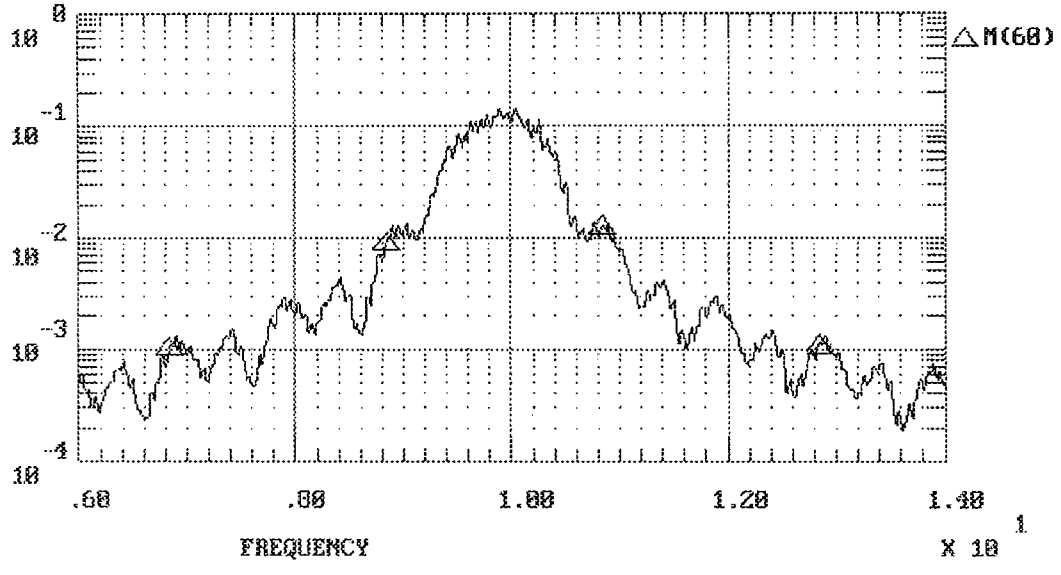

Figure 6-14: 4-h Non-Symmetric Multi-h, $f_c = 10\text{Hz}$, $f_b = 1\text{Hz}$,

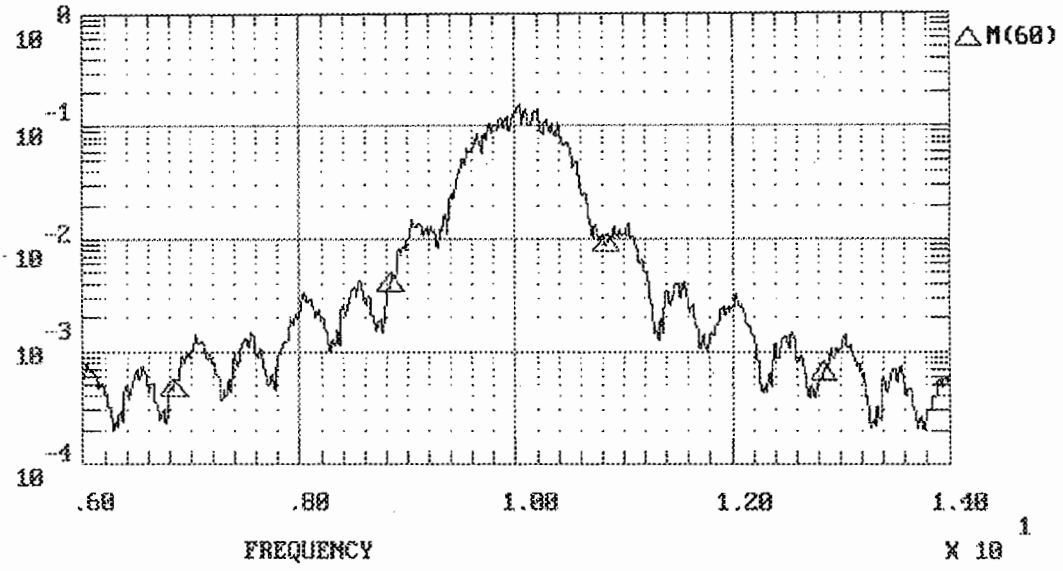

Figure 6-15: 5-h Non-Symmetric Multi-h, $f_c = 10$Hz, $f_b = 1$Hz,

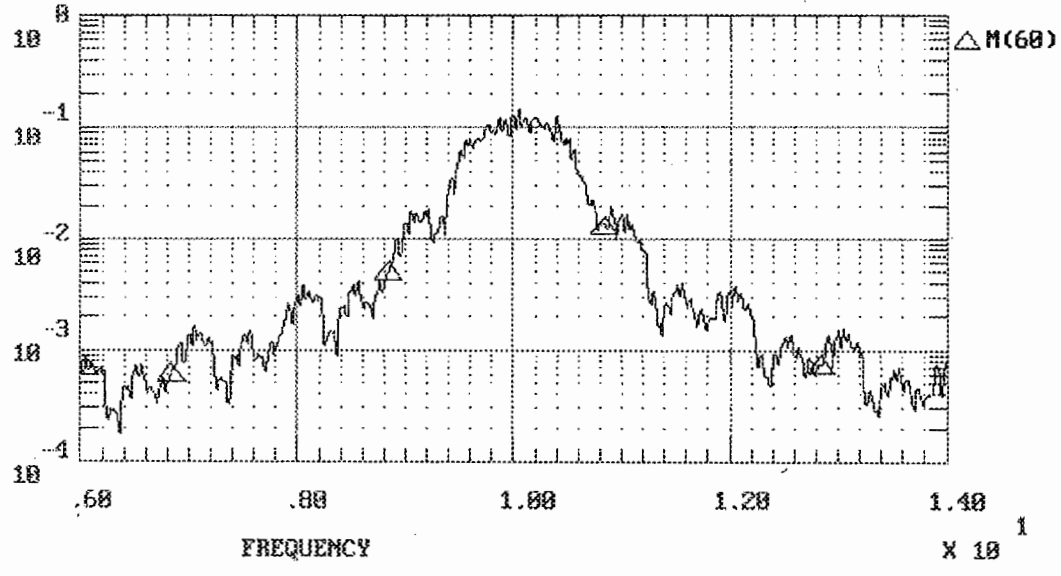

Figure 6-16: 6-h Non-Symmetric Multi-h, $f_c = 10$Hz, $f_b = 1$Hz,

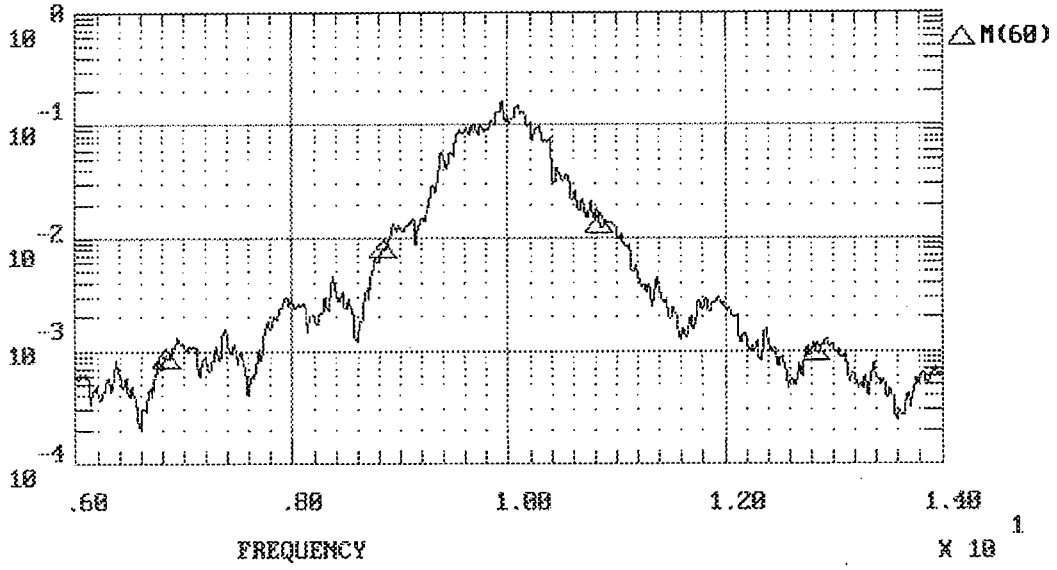

Figure 6-17: 7-h Non-Symmetric Multi-h, $f_c = 10$Hz, $f_b = 1$Hz,

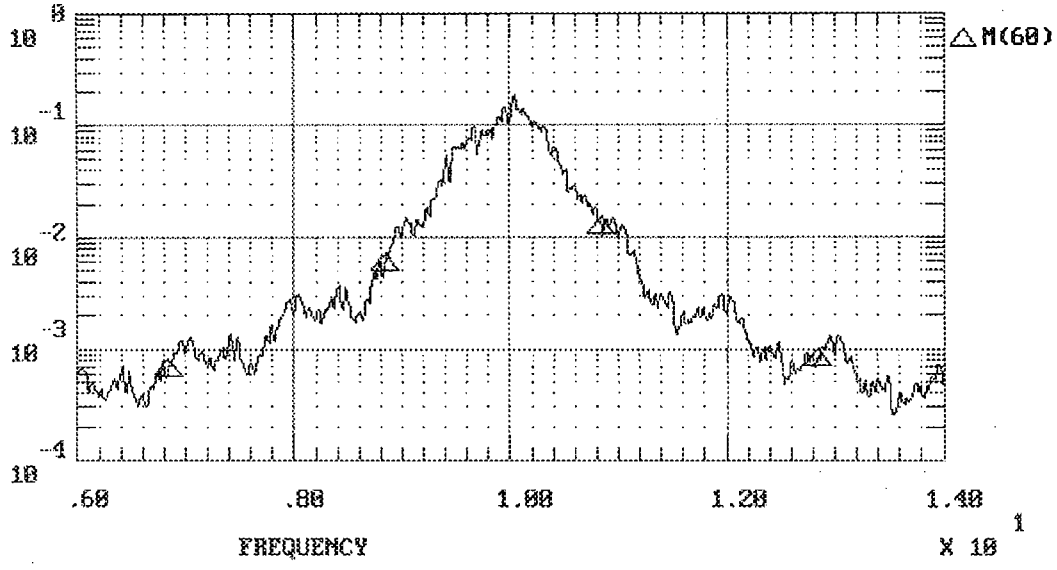

Figure 6-18: 8-h Non-Symmetric Multi-h, $f_c = 10$Hz, $f_b = 1$Hz,

Chapter 7

Conclusion

Multi-h CPFSK inherently possesses memory as is evident through the signal's continuous nature. As a result, Multi-h CPFSK can be implemented through the combination of a separate modulator components, each being responsible for the construction of the memory component and the memoryless component. Two main methods of this type of Multi-h CPFSK construction were investigated.

The first approach was an oscillator approach to Multi-h CPFSK. While this method is not a feasible option to construct Multi-h CPFSK due to the modulator complexity, it serves to illustrate the explicit memory features of Multi-h CPFSK.

A Transducer based demodulator approach such as the *Massey-Hodgart MSK Modulator/Demodulator* [31, 32] for CPFSK was sought after. Rimoldi [33] separated CPFSK into a *Continuous Phase Encoder* and *Memoryless Modulator* and showed that this MSK structure could be derived as a special case of the CPFSK structures that he had constructed. Kritzinger [34] unsuccessfully attempted to incorporate the Viterbi algorithm into a CPFSK demodulator structure in the same manner Rimoldi did for MSK. The construction of CPFSK using a *Continuous Phase Encoder* and a *Memoryless Modulator* was the second approach. Multi-h CPFSK can be constructed from a symbol-level mapper together with a CPFSK modulator structure as designed by Rimoldi.

The inherent memory can be used by an error correction decoding algorithm, allowing for better coding gains to be attained. While optimal error correction decoding algorithms are generally used in such schemes, sub-optimal error decoding methods can be advantageous in codes with numerous states and high speed applications. Certain sub-optimal error correction decoding algorithms require less hardware intensive implementations. In particular, the Fano decoder is not very hardware intensive and is easily implemented for Multi-h CPFSK use.

A number of demodulation schemes were considered including a method that follows Crawford's [7] modulator design. This led to the conclusion that Premji and Taylor's [4] is a superior method due to it's compact design and strong synchronization qualities. The synchronization acquisition makes maximal use of information contained in the memory of coding scheme (up to the truncation depth). The scope of the scheme can be broadened, so as to track large state non-symmetric codes developed in this work and can be enhanced through incorporation of faster error correction methods.

Considering the demodulator structure utilized, high performance h-index sets were found in a novel approach using the PBIL algorithm. The theoretical coding gains over **M**inimum **S**hift **K**eying found are enormous. The h-index set can be user defined through the selection of the denominator of the h-indexes and the set size which is limited by this denominator. The PBIL algorithm determines an area of strongly performing h-sets. The final sets are selected by knowledge of the demodulator characteristics. Non-symmetric h-index sets show improved minimum squared Euclidean distances over their symmetric counterparts but can encounter loses due to synchronization problems encountered in the demodulator.

7.1 Future Developments

The scheme presented here can be used to find the best performing modulation scheme using linear phase pulse (full response), has a continuous signal and a finite number of states. It is likely that another coding scheme, perhaps continuous Trellis Coded

Modulation, can be found that will be implement the scheme with these particular features. It may, however, not suffer the complex synchronization problems that this scheme does leading to smaller losses and may also be simpler to implement.

Synchronization must be considered for every h-set considered for implementation. The period of time needed for carrier and symbol synchronization to acquire lock needs to be established through experimentation or simulation. The influence of Δ on the locking characteristics of both loops also needs to be taken to consideration if the Fano decoder is utilized.

Premji-Taylor's demodulator scheme was derived using maximum likelihood methods with a linear phase ramp pulse shape. Spectral properties of Multi-h CPFSK can be enhanced by other pulse shapes and can be employed in a the same Premji-Taylor structure.

7.2 Acknowledgements

This work would not have been possible without the funding of the Foundation for Research and Development for which I am indebted. I am especially grateful for the guidance of my supervisor, Robin Braun, for direction and encouragement of this work. Through the duration of this work, the Communications Research Group has provided a good grounding for this development. I would like to thank all my colleagues in this group for assisting me in gaining my theoretical background in this field. I would also like to acknowledge the coding assistance I received from my colleagues in both the Microwave and Image Processing Groups.

Appendix A

PBIL C++ Code

The *Population Based Incremental Learning (PBIL) Algorithm* was implemented through an object orientated approach. The separate objects are as follows:

A.1 File Initialization

The initialization requires many parameters and thus is handled by editing an initialization file separately. This object reads in all the initialization data and feeds all other objects with the initialization parameters.

A.2 Meet Path

Given a h-index set, this object finds paths diverging from one another and having the earliest convergence nodes thereafter. A time threshold for the search is provided by the user. The search is undertaken by *FindMP( )* to avoid re-initialization. The *PBIL Algorithm* requires multiple searches on the same instance of the *Meet Path* object. All meets are indexed and the path can be retraced with the *RouteFinder(index)*. *Meet Path* also offers a check on early meets through *Premature_Meet(Threshold_desired)* and is used in the discard criteria of the *PBIL Algorithm*.

A.3 Euclidean Distance

The *Euclidean Distance* Object is responsible for determining the accumulated Euclidean distance between any two different paths that are given to it. It is uses the internal function *IntmedED(hcycle_position, Path 1 symbol, Path 2 symbol). CalculateED(Paths, index)* calculates the accumulated Euclidean distance of the *Paths* set number *index* utilizing *IntmedED. BestEucDis(Paths)* determines the weakest Euclidean distance calculated by *CalculateED(Paths, index).*

A special case of *BestEucDis(Paths)* is used for a 2-h plot and is made up of *CalculateGrid(Paths)* and an output function named *TwoOut(h set).*

The *PBIL Algorithm* uses *BestEDGivenHS(Paths)* instead of *BestEucDis(Paths)* as it needs to control the initialization of the h-sets directly.

Two functions are also incorporated into this object, namely *ProbError(Signal-to-Noise Ratio)* and *LogProbError(dB Signal-to-Noise Ratio)*, due to the strong relationship between error probability and Euclidean distances.

An external function *GainOverMSK(Euclidean Distance)* provides the Gain evaluations.

A.4 Probability Density Function Objects

These objects are designed to calculate the characteristics given by $G_\theta(\theta_e;\tau_e)$ and $p(\theta_e)$ for the carrier loop phase and $G_\eta(\eta;\theta_e)$ and $p(\eta_e)$ for the symbol synchronization loop phase in Chapter 5. These expressions are able to establish these functions for non-symmetric h-index sets.

A.5 Graph Plotting Objects

Graphics routines are written for usage with DOS in mind utilizing BGI graphics. These encompass multiple window and 3D plot capabilities.

A.6 File Output Objects

Two and Three dimensional file output routines are written as an alternative to the Graph Plotting Objects.

A.7 Genetic

The *Genetic* Object creates it's own instances of the *Meet Path* object and the *Euclidean Distance* object. It is entirely responsible for implementing the *PBIL Algorithm.*

A.8 Maths Routines and Other Functions

A.8.1 R(value,mod)

This is the remainder function which is necessary *inter alia* in determining the physical phase. It returns the remainder of value $\div$ mod.

A.8.2 PU(TimePeriod) and PL(TimePeriod)

These two functions return the correct numerator of $h(TimePeriod)$.

A.8.3 Trellis(TimePeriod,Symbol)

This calculates the Phase shift for a given transmission of *Symbol* in *TimePeriod.*

A.8.4 Q(x)

This is the complimentary error function required in the Probability of Error calculations.

Appendix B

Multi-h Modulator-Demodulator Simulation C++ Code

The *Multi-h Modulator-Demodulator* was simulated through an object orientated approach. The separate objects are as follows:

B.1 File Initialization

The initialization requires many parameters and thus is handled by editing an initialization file separately. This object reads in all the initialization data and feeds all other objects with the initialization parameters.

B.2 Data Generator

A random binary sequence is generated in this object. It is possible to adjust the distribution of low and bits through the threshold *DecisionLevel*. The output *PassData()* returns the bit stream for a requested bit-period while the output *Output()* returns the same stream for a particular time.

B.3 MH Modulator

A random bit-stream is modulated. *CarrierContrib()*, *ResidualPhase* and *CurrentContrib* constitute the three components of the final signal. The pulse shape is specified by *CPFSKPulse()*.

B.4 Noisy Signal

Given a specific signal to noise ratio, a noise component is added to the modulated signal. The mathematical function *gasdev()*, developed in Numerical Recipes for C [35], is used to simulate an AWGN channel.

B.5 MH Demodulator

This objects determines the correlations found in the multi-h demodulator. Synchronization is taken for granted. The individual branch metrics are computed by *BranchMetric()* or all in one go by *CalculateBMS()*. The arms are individually calculated by *IP_integ()* and *QP_integ()* using Simpsons rule for the integration. The reference signal is constructed in component form by *CarrierContrib()* and *CurrentContrib()*, specified by phase pulse shape *CPFSKPulse()*.

B.6 Buffer

This object buffers the correlations to be feed into the error correction decoders. This object can provide synthetic correlation output through *TestBufferOutput()* or the buffered output through *DemodOut()*. The decomposed synthetic correlations can be accessed using *TestI()* and *TestQ()*.

B.7 Fano

This object applies the Fano Error correction decoder to the buffered correlations. *Findings()* compares the decoder bit stream to the original stream as a measure of performance.

B.8 Viterbi

This object applies the Viterbi Error correction decoder to the buffered correlations. *Findings()* compares the decoder bit stream to the original stream as a measure of performance.

B.9 Graph Plotting Objects

Graphics routines are written for usage with DOS in mind utilizing BGI graphics. These encompass multiple window and 3D plot capabilities. This are common to the PBIL code implementation.

B.10 Maths Routines and Other Functions

B.10.1 R(value,mod)

This is the remainder function which is necessary *inter alia* in determining the physical phase. It returns the remainder of value $\div$ mod.

B.10.2 PU(TimePeriod) and PL(TimePeriod)

These two functions return the correct numerator of $h(TimePeriod)$.

B.10.3 Trellis(TimePeriod,Symbol)

This calculates the Phase shift for a given transmission of *Symbol* in *TimePeriod.*

B.10.4 gasdev(x)

This function returns a normally distributed deviate with zero mean and unit variance.

Bibliography

[1] I. Sasase and S. Mori. Multi-h phase-coded modulation. *IEEE Communications Magazine*, 29(12):46–56, December 1991.

[2] S.G. Wilson and Richard C. Gaus. Power spectra of Multi-h phase codes. *IEEE Transactions on Communications*, COM-29(3):250–266, March 81.

[3] B.E. Rimoldi. *Continuous-Phase Modulation and Coding for Bandwidth and Energy Efficiency.* PhD thesis, Swiss Federal Institute of Technology Zurich, 1988.

[4] D.P. Taylor and A. Premji. Receiver structures for Multi-h signaling formats. *IEEE Transactions on Communications*, 35(4):439–451, April 87.

[5] D.P. Taylor and A. Premji. A practical receiver structure for Multi-h CPM signals. *IEEE Transactions on Communications*, COM-35(9):901–908, September 1987.

[6] C-E.W. Sundberg and T. Aulin. Continuous Phase Modulation - Part I: Full response signaling. *IEEE Transactions on Communications*, COM-29(3):196–209, March 1981.

[7] B.P. Crawford. *Characteristics of Multi-h coded modulation.* MSc thesis, University of Cape Town, 1994.

[8] B.E. Rimoldi. A decomposition approach to CPM. *IEEE Transactions on Information Theory*, 34(2):260–270, March 1988.

[9] J.B. Anderson and Seshadri Mohan. Sequential coding algortihms: A survey and cost analysis. *IEEE Transactions on Communications*, 32(2):169–176, February 84.

[10] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Transactions on Information Theory*, IT-28(1):55–67, January 1982.

[11] J.M. Wozencraft and I.M. Jacobs. *Principles of Communication Engineering.* John Wiley and Sons, 1965.

[12] G. Ungerboeck. Trellis-coded modulation with redundant signal sets: Part I: Introduction. *IEEE Communications Magazine*, 25(2):5–11, February 1987.

[13] G. Ungerboeck. Trellis-coded modulation with redundant signal sets: Part II: State of the art. *IEEE Communications Magazine*, 25(2):12–21, February 1987.

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

[15] B.S. Katakol and S.L. Maskara. Computationally efficient modified Fano algorithm for sequential decoding. *Electronics Letters*, 21(23):1109–1111, November 85.

[16] B.S. Katakol and S.L. Maskara. Performance of modified Fano algorithm in AWGN and fading channels. *Proceedings of the IEEE*, 75(7):962-964, July 87.

[17] T.K. Truong, Irving S. Reed, Ming-Tang Shih, and E.H. Satorius. VLSI design for a trace-back Viterbi decoder. *IEEE Communications Magazine*, 40(3):616–624, March 92.

[18] D.P. Taylor and B.A. Mazur. Demodulation and carrier synchronization of Multi-h phase codes. *IEEE Transactions on Communications*, COM-29(3):257–266, March 1981.

[19] S.G. Wilson and C-D. Hsu. Joint MAP data/phase sequence estimation for trellis phase codes. June 1980.

[20] John M. Liebetreu. Joint carrier phase estimation and data detection algorithms for Multi-h CPM data transmissions. *IEEE Transactions on Communications*, 34(9):873–881, September 86.

[21] D.P. Taylor and A.T. Lereim. Spectral properties of Multi-h phase codes. Technical Report CRL-57, Communications Research Laboratory, McMaster University, July 78.

[22] M.K. Simon and J.G. Smith. Carrier synchronization and detection of QASK signal sets. *IEEE Transactions on Communications*, 22(2):98–106, February 74.

[23] B.E. Rimoldi. Exact formula for the minimum squared Euclidean distance of CPFSK. *IEEE Transactions on Communications*, 39(9):1280–1282, July 1991.

[24] R. Liyanapathirana and N. Ekanayake. On the exact formula for the minimum squared Euclidean distance of CPFSK, November 94.

[25] C-E.W. Sundberg and T. Aulin. On the minimum Euclidean distance for a class of signal space codes. *IEEE Transactions on Information Theory*, IT28(1):43–55, January 82.

[26] H-K. Hwang, L-S. Lee, and S-H. Chen. Multi-h phase-coded modulations with asymmetric modulation indexes. *IEEE Journal on Selected Areas in Communications*, 7(9):1450–1461, December 1989.

[27] J. Greene. Population-based incremental learning. *Lecture and Private Communications,1995.*

[28] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive. Technical report, Carnegie Mellon University, June 94. School of Computer Science.

[29] C-E.W. Sundberg and T. Aulin. Autocorrelation functions and power spectral densities for band-pass filtered and hard-limited continuous phase modulation. volume 3, pages 6F.6.1–6F.6.5. International Communications Conference, 1982.

[30] T. Maseng. The autocorrelation function for Multi-h coded signals. *IEEE Transactions on Communications*, COM-33(5):481–484, May 1985.

[31] J.L. Massey. A generalized formulation of Minimum Shift Keying modulation. In *Conference Record of the International Communications Conference ICC '80*, volume 2, pages 26.5.1–26.5.4, June 1980.

[32] J.L. Massey. The how and why of channel coding. In *Proceedings of the International Zurich Seminar*, pages E 1.1–E 1.7. Proceedings of the International Zurich Seminar, March 1984.

[33] B.E. Rimoldi. Design of coded CPFSK modulation systems for bandwidth and energy efficiency. *IEEE Transactions on Communications*, 37(9):897–905, September 1989.

[34] S.E. Kritzinger. *An investigation into 4-CPFSK modulation and demodulation design.* MSc thesis, University of Cape Town, 1994.

[35] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C The Art of Scientific Computing.* Cambridge Univeristy Press, Cambridge, 92.

[36] J. B. Anderson and R. de Buda. Better Phase-modulation error performance using Trellis Phase Codes. *Electronic Letters*, 12(22):587–588, October 76.

[37] J.B. Anderson and D.P. Taylor. A bandwidth-efficient class of signal space codes. *IEEE Trans. on Information Theory*, 24(6):703-712, November 1978.

[38] J.B. Anderson, T. Aulin and C.E. Sundberg. *Digital Phase Modulation.* Plenum Press, 86.