# Synchronisation in Sampled Receivers for Narrowband Digital Modulation Schemes

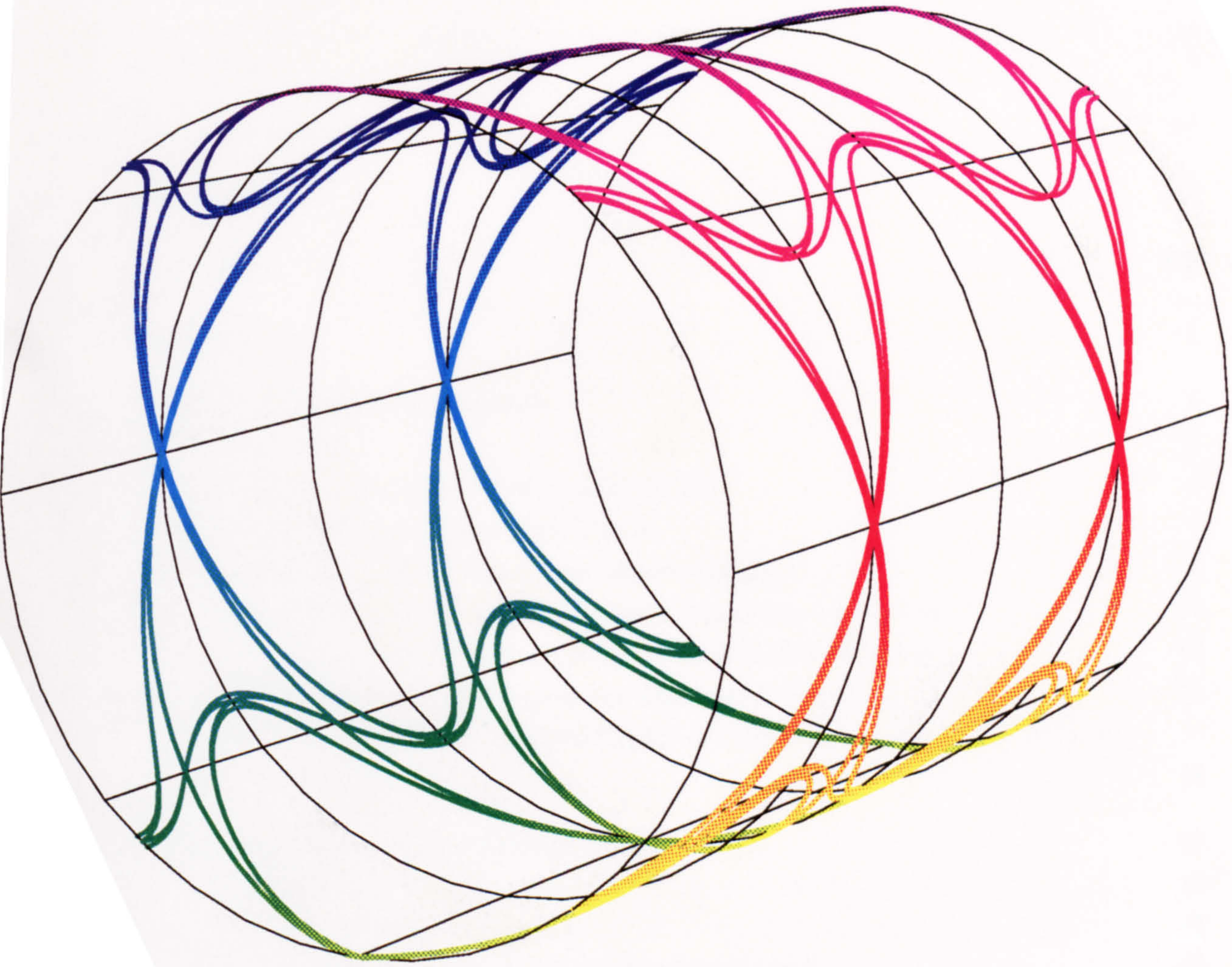
This thesis is submitted in partial fulfilment of the requirements for

Doctor of Philosophy (D.Phil.)

Dan Verdin
Communications Research Group
Department of Electronics
University of York

December 13, 1995

# Synchronisation.
# Which way is up and how far in?



# Dedication

To Bryan & Daphers for being such thoroughly nice people.

# Contents

# Acknowledgements

Thanks are due firstly to British Aerospace (Dynamics), Hatfield, and in particular the then personnel manager Jim Palmer, for allowing me to study for my D.Phil on secondment. I would also like to thank Charles Henderson of the Defence Research Agency (DRA), Malvern, who took an early interest in and gave support to the work of the Communications Research Group at York. Thanks also to James Christie also at the DRA. Dr. Rob Stevens gave me lots of technical advice on digital synchronisation which I found immensely useful. Thanks also to Dr. Steve Marsh for his continued support on the digital synchronisation work. Thanks go to the Alta Group of Cadence Design Systems for allowing us to use their CAAD software for research/teaching purposes.

I'd like to thank my wife, Wendy, who didn't type the manuscript but did bring me the odd cup of tea. Finally, I'd like to thank my supervisor, Tim Tozer, without whose gentle persuasion this thesis would not have been possible.

# Declaration

Some of the research in this thesis has resulted in publications in the open literature. These papers are included in an appendix to this thesis.

To the best knowledge of the author any work presented in this thesis as being original is so. Due references and acknowledgements of other researchers in this field have been given where appropriate.

# Abstract

As the power and availability of digital signal processing devices continues to increase, modem functions that were once performed by analogue methods are now being done digitally. Perhaps the most critical function is synchronisation. This thesis is concerned with the study of algorithms suited to digital implementation for carrier frequency, symbol timing & carrier phase synchronisation. Digital algorithms are not only interesting because of the usual advantages of digital over analogue circuits, namely that they need no alignment and can lead to reduced costs by large scale integration etc., but also because they allow one to develop algorithms from a sound theoretical basis which come close in terms of their performance to what is theoretically possible. By using digital algorithms it is possible to explore more of the "design-space" in the search for optimum receiver structures.

The performance of digital synchronisers with offset and non-offset $M$-PSK type modulations, popular for mobile radio & satellite applications, is explored via simulation and analysis and a number of new results developed.

# Chapter 1

# Introduction

### Problem Statement

Synchronisation is possibly the most important function in any receiver for without it all else will fail. Developing new esoteric modulation and coding schemes is of little use without considering how one would implement appropriate synchronisation functions. This is particularly true in modern times where emphasis is on modulation schemes that are spectrally efficient, in terms of the bits/s/Hz that they will support, and may have to operate in burst mode.

Synchronisation means the recovery of reference parameters associated with a modulated waveform and the application of these parameters to aid the data detection process. These parameters are carrier frequency, symbol timing and carrier phase (for coherent detection). Actually, one can assume an extra parameter to be estimated, namely the transmitted data sequence, as is done in a multiple parameter estimation approach.

In this thesis we are concerned with synchronisation in sampled receivers. That is to say, we are interested in those algorithms that are suited to implementation on some form of Digital Signal Processor.

The problem of synchronisation is illustrated by figure 1.1. A received passband signal, perhaps after coarse frequency correction, is down converted to in-phase (I) and quadrature (Q) components by a *fixed* local oscillator. This analogue waveform is then sampled at a *fixed* rate by the two analogue-to-digital converters. We assume that the Digital Signal Processor has no control over either the local oscillator frequency/phase or the clock frequency/phase. Any corrections for the errors between the received signal frequency and the local oscillator frequency will have to be done digitally. Similarly there will be errors between the local sampling clock and the timing *epoch* of the received signal. Timing errors may be very small, of the order of parts per million, but they are there nevertheless. Frequency errors can be much larger depending on the type of environment the receiver has to operate in. For satellite communications it is possible to have frequency errors as large as the baud rate.

As the signal processor makes no attempt to change the frequency/phase of the clock or the local oscillator these systems are referred to as being *asynchronous.*

Digital asynchronous receivers of the type depicted in figure 1.1 are attractive for a number of reasons. By reducing the amount of analogue hardware in the receiver to simply those

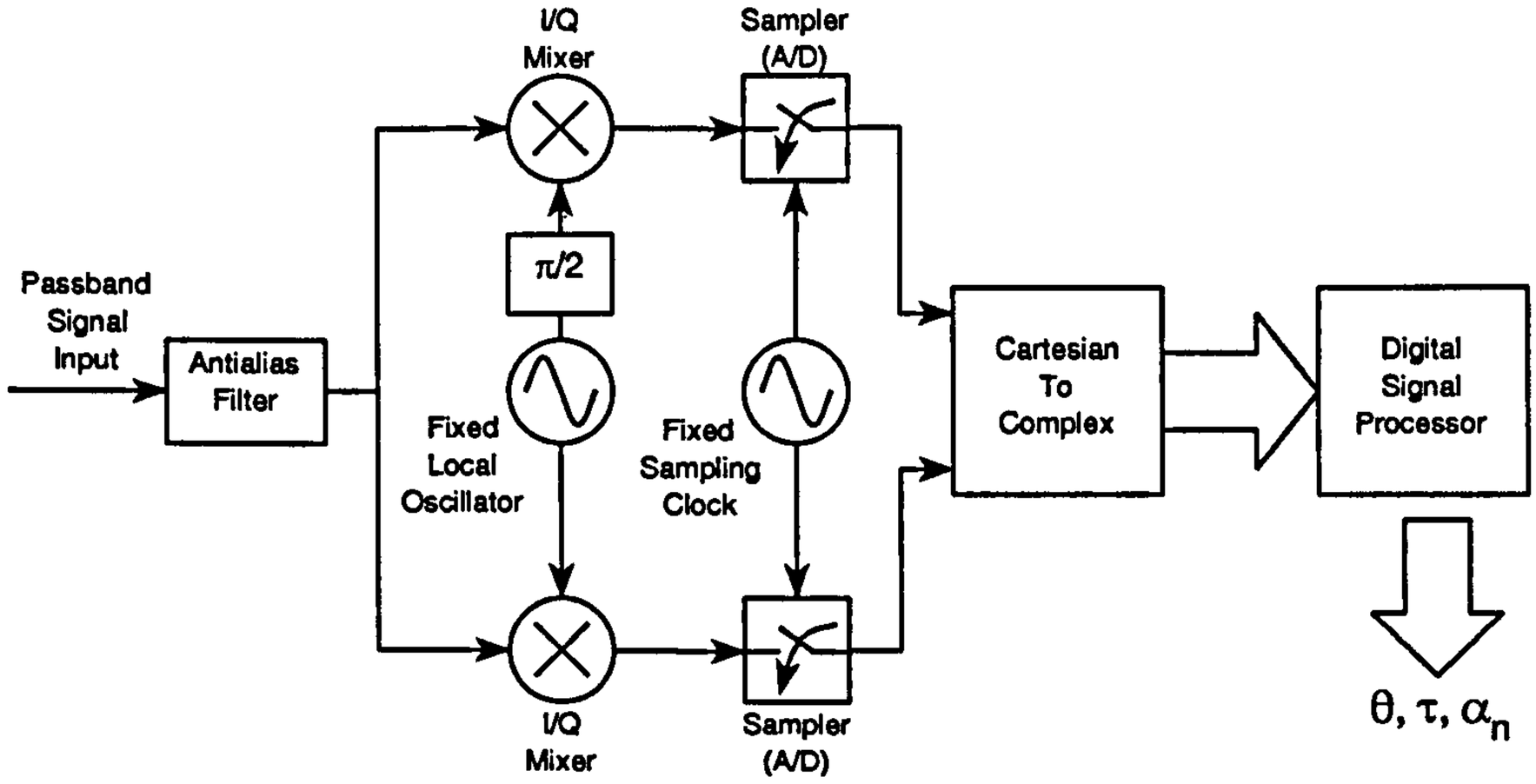



Figure 1.1: Conceptual Asynchronous Timing/Phase Recovery System

functions required at the "front-end" we get all the benefits commonly associated with digital systems, namely, low cost (in high volumes), large scale integration, and minimal variation in performance from receiver to receiver. Of more interest, perhaps, is the philosophy of developing a "generic" receiver based on the programmability of the Digital Signal Processor. Conceptually, at least, one can change the function of the receiver, to detect a different modulation scheme, for example, by changing the software rather than the hardware.

In figure 1.1 we have the conceptual block "Cartesian-To-Complex" to denote that the Digital Signal Processor is, in general, performing complex operations upon complex data. We are only concerned with these complex operations in this thesis and not with the "front-end" configuration.

Our concern here is with floating-point algorithms and hence are strictly sampled, rather than digital, in nature.

### Overview of the Development of the Subject

The study of digital algorithms for synchronisation is relatively new. The field is really a marriage between the two much more mature fields of Estimation Theory [1, 2] and Digital Signal Processing [3] and there are few useful references on the subject in the open literature. Many of the references that do exist concern analogue synchronisation methods, or are simply *digitalised* versions of of analogue methods. A useful overview of synchronisation in data communications is an IEEE special issue [4].

The earliest contributions to the development of the subject were by Mueller [5] who developed a general theory for "T"-spaced timing recovery in 1976. One of the special cases of the general theory developed in [5] is usually referred to as the Mueller and Müller algorithm. Also, Lindsey in [6] describes an early-late method of timing recovery called the Data Tran-

sition Tracking Loop (DTTL). Kobayashi [7] realised the advantages in multiple parameter estimation applied to synchronisation systems.

Gardner has been one of the main protagonists in the modern development of the subject. He published his simple non data-aided timing-error detector for asynchronous timing-recovery in 1988 [8], followed by a number of reports on digital synchronisation algorithms for the European Space Agency [9, 10, 11, 12, 13]. It is only relatively recently that researchers have realised that interpolation is an important function in asynchronous timing recovery loops. Farrow developed a continuously variable delay element in 1988 [14]. Work on interpolation was expanded upon and consideration given to appropriate control structures more recently in [15, 16].

Using methods borrowed from estimation theory Gardner developed his Maximum Likelihood frequency error detector in 1990 [12]. At the same time other researchers had developed other frequency-error detectors, some based on *ad-hoc* methods [17, 18, 19, 20, 21]. Many of these were shown to have equivalent structures in [22].

The application of estimation theory to digital synchronisation not only, in some cases, provides one with optimum structures for parameter estimation, but also provides the bounds on synchroniser performance. In other words, this typically gives one an indication of the jitter one can expect at a given value of signal to noise ratio. Bounds on synchroniser performance, generally referred to as the Cramer-Rao bound, have been developed by Moeneclaey in [23, 24] who has also developed synchroniser structures using estimation theory methods in [25, 26, 27]. More recently a *modified* Cramer-Rao bound has been developed by D'Andrea in [28] which has been applied extensively in this thesis.

Worthy attempts to give an overview of the subject, even though it still has some way to go before being fully developed, have been made by Gardner [9] and Jesupret [29].

## Overview of Thesis

It is generally accepted that the development of a thesis such as this is something of a "random-walk" process. The order of the material as it appears here is not necessarily in the order it occurred chronologically. At the outset of this project more emphasis was placed on modulation schemes for mobile radio and their performance in the flat fading channel. It was during this initial period that work was undertaken on importance sampling, modelling of the narrowband fading channel and differential & coherent detection schemes for Gaussian Minimum Shift Keying (GMSK) and the characterisation of these schemes in terms of their symbol error rates. At this stage of the project interest shifted to concentrating on how to actually synchronise these narrowband modulation schemes starting with symbol, then frequency and finally phase synchronisation.

The order then that the material appears follows a more natural flow. The next chapter gives an introduction to the tools and methods employed in the simulation of communication systems. Specifically important for synchronisers is the complex-baseband representation. Work on the use of importance sampling to improve the error rate estimate (variance minimisation) is also given here.

The order of the material then goes as: transmitter functions; channel modelling; receiver synchronisation in the order in which it would occur[1], namely, frequency, symbol timing and phase recovery; and finally some notes on the integration of various receiver functions in a sampled system.

Then we go on to characterise the modulation schemes that will be used extensively throughout the rest of the thesis. We introduce various representations of these complex modulation formats. The inter-symbol interference inherent in pre-filtered Continuous Phase Modulation (CPM) schemes is quantified to be used later in the work on decision-feedback equalisation. The root raised cosine pulse shapes and derivatives are given here which will be used throughout the rest of the thesis. Also the simulation models for various modulation schemes are given.

The next chapter is on channel modelling, specifically the narrowband fading channel appropriate to urban mobile communications. A novel method of designing the FIR correlation shaping filter based on the time domain window design method is presented. It is also noted for the first time that the characteristics of the urban mobile channel can be approximated by using a simple IIR filter with a cutoff frequency proportional, but not equal to, the required Doppler frequency. We present a general method to derive this constant of proportionality.

We then move on to the main areas of effort in this thesis, namely, frequency and symbol synchronisation. A number of algorithms for frequency synchronisation are studied. These schemes will acquire signals with a frequency offset of the order of the baud rate. The open-loop characteristics of the frequency-error detectors are determined, analytically and by simulation, and small-signal linearised models developed using standard phase-locked loop theory. Justification of the small-signal linearised models is then given by the examination of the closed-loop acquisition and tracking performance. Some novel observations are made with regard to the large-signal acquisition performance of the various loops.

The next chapter on symbol synchronisers rationalises a number of schemes available in the literature in terms of the number of samples per symbol they require and the nature of the nonlinearity in the timing-error detector. We concern ourselves with non-offset $M$-PSK type modulation schemes. We re-write the timing error detector algorithms in their complex form, from which properties such as carrier phase invariance become obvious. Researchers have in the past tended to consider the problem of symbol timing recovery as a *real* operation on a *real* signal. By re-writing the algorithms as complex we develop and characterise a new class of timing-error detectors which have performance intermediate between decision-directed and non data-aided algorithms. We characterise these new schemes analytically and by simulation. It appears that these new schemes offer considerable improvement over conventional non data-aided symbol synchronisers in that they too are carrier phase invariant, unlike decision-directed methods, but do not have a sinusoidal S-curve characteristic and hence are not prone to hangup. The gain of the new detectors are non-zero for zero modulation excess bandwidth unlike conventional non data-aided schemes which cannot be used with tightly bandlimited signals. We also look at the important subject of interpolation in sampled receivers and

[1] Neglecting for the moment joint detection schemes

argue that continuously-variable polynomial are preferable to polyphase switched-coefficient interpolators.

We then look at the problem of carrier phase synchronisation in sampled receivers. We concentrate on an interesting technique, developed by others, of improving the performance of pre-filtered CPM by decision feedback equalisation. We compare the performance of one and two bit differential phase detection of GMSK with and without decision feedback, and note that large gains in performance can be gained by application of this technique. We also look at the ideal orthogonal coherent detector for offset modulation schemes. We also show performance results of a quintessentially digital algorithm for feedforward carrier phase recovery, namely, the Viterbi & Viterbi algorithm. We make the observation that not only does one require differential encoding with such a scheme but also "phase-unwrapping."

Finally, we indicate how the various aspects of this research can be brought together by demonstrating how the various receiver functions can be integrated together. We show a coherent system for the joint recovery of clock and carrier for MSK type modulation schemes. A novel implementation of the integration of differentially detected CPM with decision feedback equalisation and symbol timing is given. Finally, a novel timing, phase and frequency recovery system for non-offset modulation schemes is demonstrated.

# Chapter 2

# Simulation Tools and Methods

## 2.1 Introduction

In this chapter an overview of the Computer Aided Analysis and Design (CAAD) tools used to model transmitter/receiver architectures and channels for digital land-mobile radio system simulation. We shall examine techniques that can be applied more generally to simulation of communications systems which will tell us the accuracy of parameter estimation and how that accuracy may be improved.

## 2.2 The Signal Processing WorkSystem (SPW)

The CAAD system simulator used for the bulk of this thesis is the Signal Processing WorkSystem. SPW is a communications/DSP systems simulator. It is made up of three basic parts; the Block Diagram Editor (BDE), the Signal Display Editor (SDE) and the Filter Design System (FDS). In the BDE one is able to *design* prototype communications/DSP systems from a suite of available building blocks. Systems are built up by schematic capture of these building blocks. The blocks available cover a wide range of signal processing functions including sophisticated system components such as Adaptive Filters. Having built up a prototype `detail` it is possible to link this with a `symbol` to represent that system function. These symbols can then be used as 'black-boxes' for use in further system designs. For example, a CAAD detail of a receiver system may be saved in the file `rx.detail` which will then be automatically represented at the higher hierarchical level by the symbol `rx.symbol`. This hierarchical nesting can continue to whatever level desired.

Custom coded blocks can also be built where one desires a particular signal processing block to undertake a function that cannot be expressed in terms of the fundamental building blocks available. One defines a symbol with as many parameters passed to and from it as required, uses SPW to compile this down to **C** code where the same parameters are then made available and one can write whatever signal processing function was desired in software. Such custom coded blocks would then take the place in a system design of the real-life functions performed by software in microprocessors.

The second part of SPW is the SDE. This is where one designs signals to be input to the

BDE and/or examines signals from the BDE. Signals to be used as inputs to the prototype systems can alternatively be defined within the BDE itself by using programmable function generators and noise sources. Signals in the SDE have further filtering and math or scalar functions available.

The FDS is where one is able to design filters from a variety of methods for inclusion in designs within the BDE. One is also able to examine the effects of quantisation on the filter's performance.

The BDE, SDE and FDS can all be controlled by their own Macro Command Language (MCL) which is similar to ADA. The MCL is useful in that it allows one to automate repetitive, tedious operations. Thus the MCL could be used to run system simulations over a range of different parameter values to determine, for example, the bit-error rate performance of a particular modulation scheme in a particular channel.

## 2.3 Complex Baseband Equivalent Signals

A general signal can be represented as follows,

$$s(t) = a(t)\cos(\omega_c t + \phi(t)) \tag{2.1}$$

where $a(t)$ is the signal amplitude, $\phi(t)$ is the phase and $\omega_c$ is its frequency. Now, noting that,

$$\cos\theta = \mathrm{Re}\{e^{j\theta}\} \tag{2.2}$$

s(t) can be written,

$$s(t) = \mathrm{Re}\{a(t)e^{j\phi(t)}e^{j\omega_c t}\} \tag{2.3}$$

or

$$s(t) = \mathrm{Re}\{u(t)e^{j\omega_c t}\} \tag{2.4}$$

The information of the real passband signal $s(t)$ is now contained in the complex baseband signal $u(t)$ [30]. The signal $u(t)$ is termed the lowpass equivalent of $s(t)$. This type of analysis applies also to other passband components such as filters.

Equivalent baseband signals are used in all the simulations performed by SPW to represent passband signals. This is possible as SPW allows the use of complex signals. The sampling frequency is then determined directly from the *information bandwidth* of the signal, rather than the absolute bandwidth.

When simulating communication systems one is generally attempting to build up a model of a process at a radio frequency (RF). It is impractical to represent these RF processes as *real* signals and one generally has to resort to a complex baseband representation of both the signal and the signal processing function. Thus, many of the systems that appear in SPW, such as Costas Loops and Phase Locked Loops, may appear to be quite different in appearance from their real counterparts as they are represented as complex equivalent models. The tutorial paper by Franks [4] gives a good overview of the complex envelope representation of signals.

In what follows, the equivalence of using the complex envelope representation of signals will be demonstrated when one configures frequency conversion elements in SPW.

In real communication receivers a mixer is used to down-convert the received frequency at RF (the Radio Frequency) to IF (the Intermediate Frequency) or baseband. This is accomplished by multiplying the received signal at a frequency of $f_{RF}$ with a Local-Oscillator (LO) at a frequency of $f_{LO}$. The result of this multiplication is to give frequency components at the sum frequency, $f_{RF} + f_{LO}$ , and the difference frequency, $f_{RF} - f_{LO}$. When considering frequency down-conversion it is the difference-frequency component that we are interested in and so the output from the multiplier has to be low-pass filtered.

One can use the complex multiplier blocks in SPW to simulate an ideal mixer. In order to illustrate this, the equivalence between the real mixer and a mixer based on the equivalent complex baseband model will now be demonstrated.

Firstly, using a real signal representation we define $Y(t)$ as the input RF signal and $R(t)$ as the locally generated LO as follows:

$$Y(t) = A(t)\cos\left(2\pi\, f_{RF}\, t + \theta(t)\right) \tag{2.5}$$

and,

$$R(t) = \cos\left(2\pi\, f_{LO}\, t + \hat{\theta}(t)\right) \tag{2.6}$$

where $\theta(t)$ is the phase of the input signal and $\hat{\theta}(t)$ is the phase of the LO (an *estimate* of $\theta(t)$). By multiplication we get:

$$\begin{aligned} Y(t) \times R(t) &= A(t)\cos\left(2\pi\, f_{RF}\, t + \theta(t)\right) \times \cos\left(2\pi\, f_{LO}\, t + \hat{\theta}(t)\right) \\ &= A(t)\cos\left(2\pi(f_{RF} - f_{LO})\, t + \theta(t) - \hat{\theta}(t)\right) \\ &+ A(t)\cos\left(2\pi(f_{RF} + f_{LO})\, t + \theta(t) + \hat{\theta}(t)\right) \end{aligned} \tag{2.7}$$

If we now apply a low-pass filter to this signal then the double frequency term can be set to zero and the result is:

$$Y(t) \times R(t) = A(t)\cos\left(2\pi(f_{RF} - f_{LO})\, t + \theta(t) - \hat{\theta}(t)\right) \tag{2.8}$$

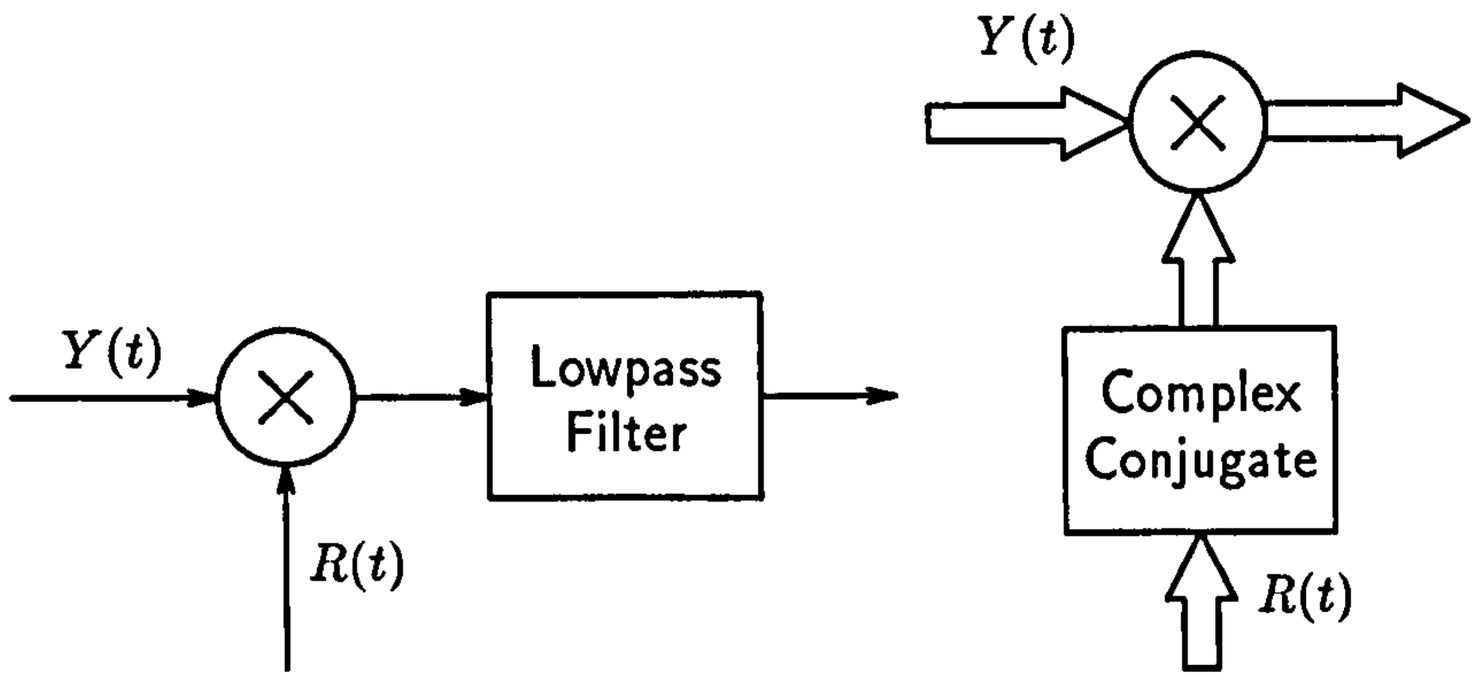
We will be concerned with frequency down-conversion to baseband or such that $f_{IF} = 0$, where $f_{IF} = f_{RF} - f_{LO}$. So when the frequency down-conversion is to zero IF the output of the ideal mixer can now be written as:

$$Y(t) \times R(t) = A(t)\cos\left(\theta(t) - \hat{\theta}(t)\right) \tag{2.9}$$

The *real* frequency down-conversion is represented schematically in figure 2.1.

In SPW we represent passband signals as equivalent complex baseband signals and so we will now show the equivalence of using complex models to simulate a real mixer. In order to represent the function of frequency down-conversion we need to replace the input RF signal and the LO signal by their complex baseband equivalents which are:

$$Y(t) = A(t)\exp\left(j\left(2\pi\, f_{RF}\, t + \theta(t)\right)\right) \tag{2.10}$$




(a) Real Mixer (b) Complex Mixer

Figure 2.1: Real and Complex Baseband Representations of Mixers

and

$$R(t) = \exp\left(j\left(2\pi f_{LO}\, t + \hat{\theta}(t)\right)\right) \tag{2.11}$$

respectively. It can be seen that the ideal mixer using complex equivalent baseband signals is arrived at by multiplying the input signal with the complex-conjugate of the local oscillator:

$$\mathrm{Re}\left\{Y(t) \times R(t)^*\right\} = \mathrm{Re}\left\{A(t)\exp\left(j\left(2\pi\left(f_{RF} - f_{LO}\right)t + \theta(t) - \hat{\theta}(t)\right)\right)\right\} \tag{2.12}$$

simplifying to :

$$\mathrm{Re}\left\{Y(t) \times R(t)^*\right\} = A(t)\cos\left(\theta(t) - \hat{\theta}(t)\right) \text{ if } f_{RF} = f_{LO} \tag{2.13}$$

Thus the schematic diagram representing the ideal mixer but using complex signals is also shown in figure 2.1. Similarly, if one were to directly multiply the signals together then the sum frequency term would result. We shall be interested in the sum-frequency terms when we look at squaring loops later on.

## 2.4 The Reliability of Bit Error Rate Measurements

The bit-error rate (BER) of a communications system is an important measurement of the system's performance. It is reasonable, therefore, that if one measures the BER either by simulation or by experiment one should be able to specify the errors on this BER measurement. The errors on the BER measurement arise in practice because one does not have an infinite sample on which to make an estimate. If one forms an estimate of the probability of symbol error by taking the ratio of the number of errors, $n_e$, to the number of symbols, $n_s$, in a given time, then this measurement could be in error by virtue of the fact that the underlying probability of symbol error, $p_e$, is small compared to the total number of symbols – it is conceivable that one might not measure any errors at all in the sample.

In order to quantify the errors associated with BER measurements, either by simulation or by experiment, one starts by deriving the probability distribution functions based on the methods by which the BER is measured. There are two obvious methods by which the BER can be measured; one can count a pre-determined number of symbols, $n_s$, and note the number of errors incurred, $\hat{n}_e$, or, one can count a pre-determined number of errors, $n_e$, and note the number of symbols, $\hat{n}_s$, required to produce these errors. These two methods shall be termed *error-counting* and *symbol-counting* respectively. Intuitively, one would expect that these two methods would be entirely equivalent in terms of the estimated probability of symbol error, and indeed this is the case, although it will be shown here that the probability distribution functions of the random variables $\hat{n}_s$ in symbol-counting and $\hat{n}_e$ in error-counting are not equivalent.

One starts by deriving the probability distribution function for the error-counting case. In this case a fixed number of symbols, $n_s$, of which $\hat{n}_e$, $0 \leq \hat{n}_e \leq n_s$, are in error. The distribution function in this case is clearly given by the discrete *binomial* distribution as –

$$P(\hat{n}_e) = \frac{n_s!}{(n_s - \hat{n}_e)!\ \hat{n}_e!}\ {p_e}^{\hat{n}_e}\ (1 - p_e)^{n_s - \hat{n}_e} \text{ for } 0 \leq \hat{n}_e \leq n_s \tag{2.14}$$

where $p_e$ is the probability of symbol error.

This distribution function in terms of its mean, $E(\hat{n}_e)$, and its variance, $E(\hat{n}_e^2) - E(\hat{n}_e)^2$, where $E(x)$ is defined as the expected value of the random variable $x$ and is given by –

$$E(x) = \sum_{\hat{n}_e=0}^{n_s} x \binom{n_s}{\hat{n}_e}\ {p_e}^{\hat{n}_e}\ (1 - p_e)^{n_s - \hat{n}_e} \tag{2.15}$$

Thus for the binomial distribution the mean and variance are given by,

$$\mu = n_s p_e \text{ and } \sigma^2 = n_s p_e (1 - p_e) \tag{2.16}$$

This gives the intuitively correct result that the expected number of errors is simply the product of the number of symbols in the sample and the probability of symbol error. It is reasonable to take the standard deviation as a measure of the error in the estimate of $n_e$; for the normal distribution one standard deviation either side of the mean covers 68% of occurrences whereas two standard deviations covers 95%. So if one defines the estimated probability of error, $\hat{p}_e$, as the ratio $\hat{n}_e/n_s$, then an error in the estimate of $n_e$ will be reflected as an error in the estimate of $p_e$ as follows –

$$\Delta p_e = \frac{\sigma}{n_s} = \frac{\Delta n_e}{n_s} = \sqrt{\frac{p_e\ (1 - p_e)}{n_s}} \tag{2.17}$$

To derive the probability distribution function for the symbol counting scenario one notes now that the number of symbols is the random variable, $\hat{n}_s$, $n_e \leq \hat{n}_s \leq \infty$, and the number of errors, $n_e$, is fixed. The error counting procedure finishes as soon as $n_e$ symbol errors have occurred. This implies that the last symbol in the $\hat{n}_s$ symbols is in error, leaving $n_e - 1$ errors in $\hat{n}_s - 1$ symbols. The probability distribution function for the random variable $\hat{n}_s$ follows directly as –

$$P(\hat{n}_s) = \frac{(\hat{n}_s - 1)!}{(\hat{n}_s - n_e)!\ (n_e - 1)!} {p_e}^{n_e}\ (1 - p_e)^{\hat{n}_s - n_e} \text{ for } 0 \leq \hat{n}_s \leq \infty \tag{2.18}$$

where again $p_e$ is the probability of symbol error. This distribution is called the *negative binomial* or the *pascal* distribution. The mean and variance are given as the expected values $E(\hat{n}_s)$, and $E(\hat{n}_s^2) - E(\hat{n}_s)^2$ where $E(x)$ is now given as,

$$E(x) = \sum_{\hat{n}_s=n_e}^{\infty} x \binom{\hat{n}_s - 1}{n_e - 1} {p_e}^{n_e} \, (1 - p_e)^{\hat{n}_s - n_e} \tag{2.19}$$

Evaluating these expected values gives the mean and variance as follows –

$$\mu = \frac{n_e}{p_e} \text{ and } \sigma^2 = \frac{n_e}{p_e} \left( \frac{1}{p_e} - 1 \right) \tag{2.20}$$

Thus the expected values for $\hat{n}_s$ is simply the number of errors divided by the probability of error. If $\sigma$ is again used as a measure of the error in the estimate of $n_s$ then the error in the estimate of $p_e$ can be written as,

$$\Delta \hat{p}_e = \frac{n_e}{n_s^2} \, \sigma = \frac{n_e}{n_s^2} \, \Delta n_s = \sqrt{\frac{p_e \, (1 - p_e)}{n_s}} \tag{2.21}$$

This is the same result as previously for the error in the estimate of $\hat{p}_e$ which shows that the two methods of determining the BER are equivalent although they are described by different probability distributions.

In all of the BER simulations undertaken using SPW a fixed number of errors have been counted and the appropriate number of symbols noted to form an estimate of the probability of symbol error. The region of interest in terms of the theoretical probability of errors that measurements will be taken between are $p_e = 10^{-1}$ to $p_e = 10^{-6}$. If all the data points will be within $2\sigma$ then for $p_e = 10^{-1}$ and $n_e = 100$ we can write the error in the estimate of $p_e$ as,

$$\Delta p_e = 2 \sqrt{\frac{p_e^2 \, (1 - p_e)}{n_e}} \approx 0.02 \tag{2.22}$$

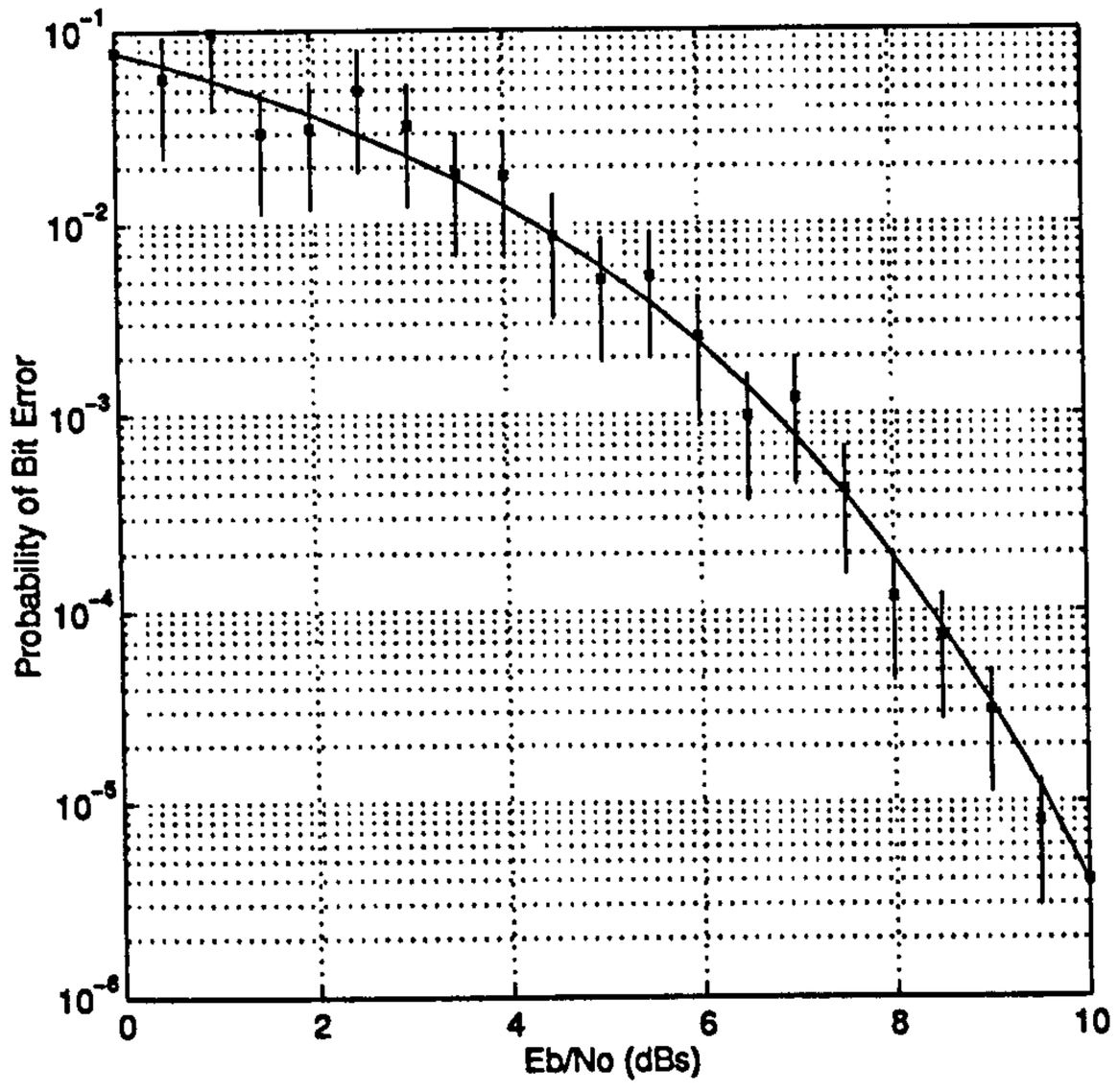
In terms of percentage accuracy this will render an estimate of $p_e$ which is 95% accurate.

Similarly, if the theoretical probability of error is $p_e = 10^{-6}$ and the number of errors to be counted is $n_e = 100$, then the error in the estimate of $p_e$ will be $\Delta p_e \approx 2 \times 10^{-7}$ and in this case the estimate is 95% accurate.

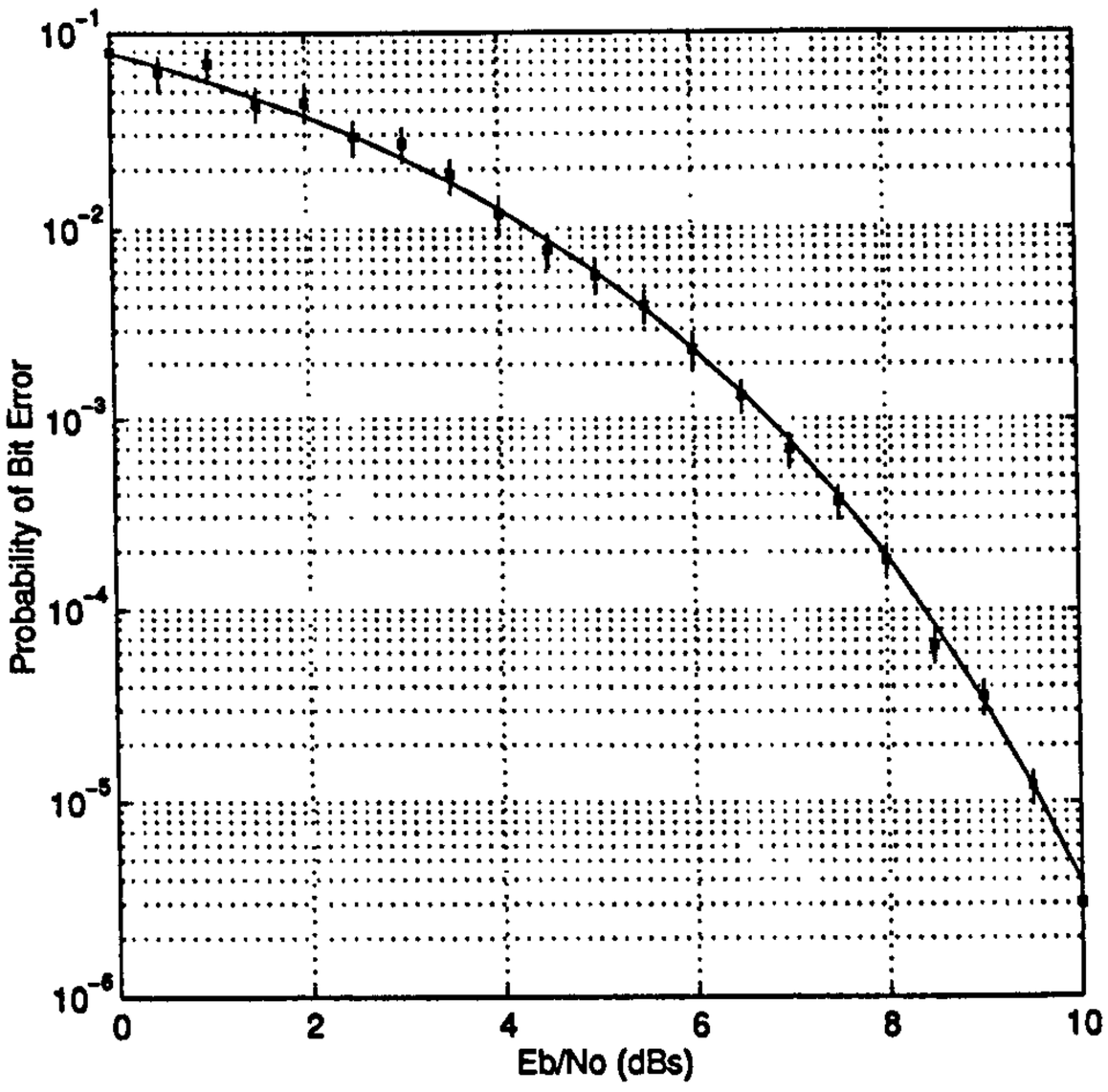
The preceding analysis is substantiated if one examines the results of an error-counting monte-carlo simulation to determine the BER for binary antipodal signalling in Additive White Gaussian Noise (AWGN) as shown in figure 2.2. The two plots in the figure indicate BER results when counting 10 and 100 errors. Together with these results are shown the 95% confidence interval of 2 stadard-deviations from the theoretical ideal. One can see that all of the results lie within 2 standard deviations of the mean. Also, counting more errors gives a smaller spread in the BER results.

## 2.5 The Estimation of Bit Error Rate using Importance Sampling

There are a number of ways to estimate the bit-error rate in the simulation of communication systems [31], although they are only generally valid when when the noise present in the system




(a) Monte-Carlo Simulation, 10 Errors Counted




(b) Monte-Carlo Simulation, 100 Errors Counted

Figure 2.2: Bit Error Rate Results for Binary-Antipodal Signalling in AWGN by Monte-Carlo Simulation.

is additive white and Gaussian. When the system contains nonlinearities then it becomes more difficult to reduce the amount of simulation time by any special methods. Importance sampling is a technique used in the simulation of communication systems which allows one to either reduce the number of iterations used in the simulation whilst maintaining the same accuracy in the bit-error rate estimate, or it will give higher accuracy when the same number of iterations are used.

### 2.5.1 Importance Sampling using a Gaussian Bias Function.

The essence of the Importance Sampling technique is to consider the *distribution* of the estimated error rate that one might get at a particular signal to noise ratio. For example, consider the simple case of binary antipodal signalling in AWGN at a signal to noise ratio of 9.0dB.

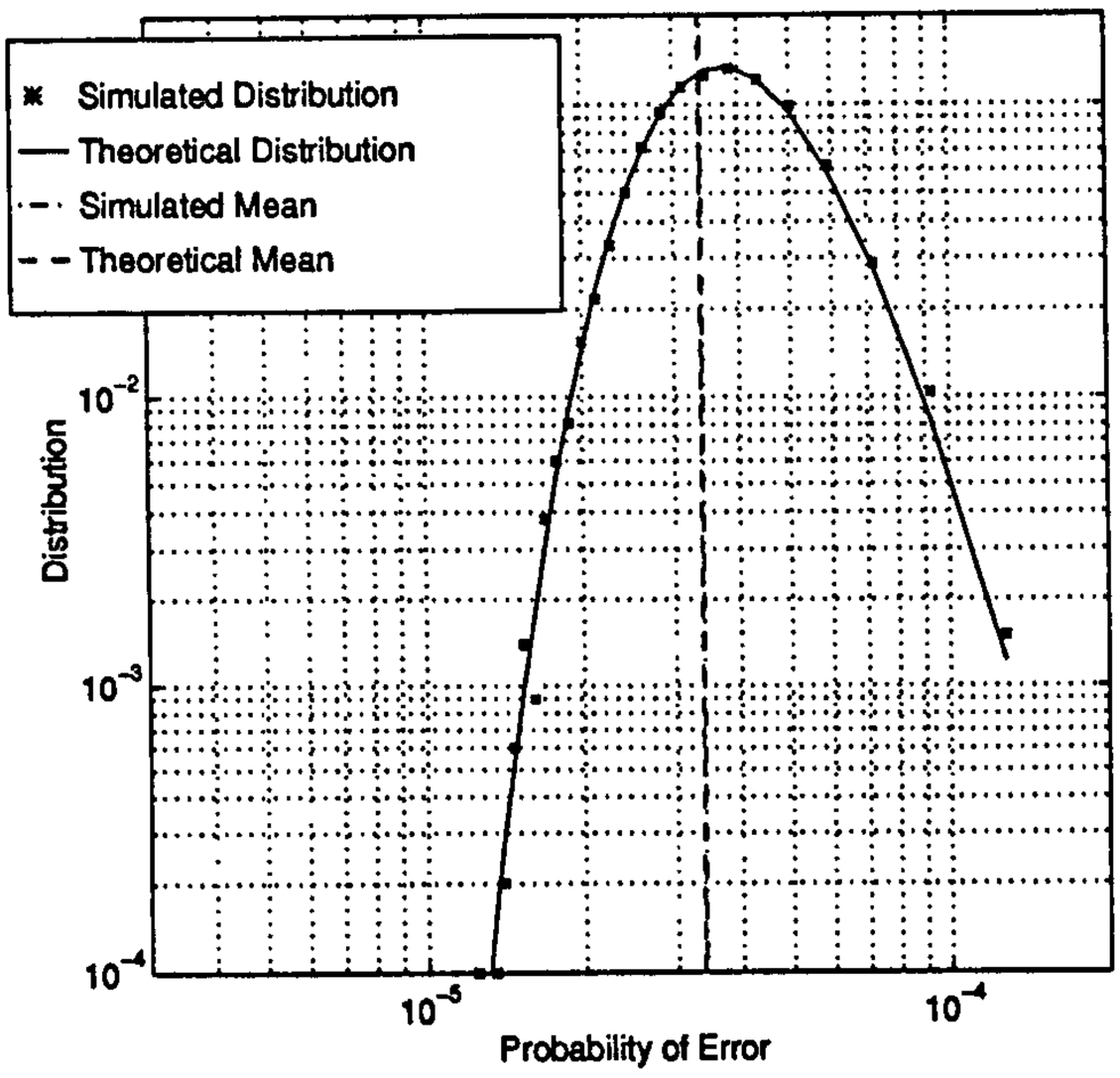
If one expected 10 errors during the Monte-Carlo simulation then one would be required to count 297378 symbols ($\approx \frac{10}{P_e}$) on average. The distribution of the estimated error rate can be found, by simulation, by selecting a random noise seed from a uniform distribution for each simulation run. This is facilitated by the use of the Code Generation System to generate a `C` source code which can be run over and over again with different noise seeds. The resulting distribution obtained is as shown at the top of figure 2.3. It can be seen that this distribution matches that of the binomial distribution put forward in the last section. The simulated and theoretical mean of the distribution is also plotted in the graph.

Figure 2.3 also shows the distribution of the estimated probability of error for binary-antipodal signalling in AWGN at an $E_b/N_o$ of 9.0dB when we run the simulation for a fixed number of symbols. Note that the mean and the variance of the distribution is the same in either case.

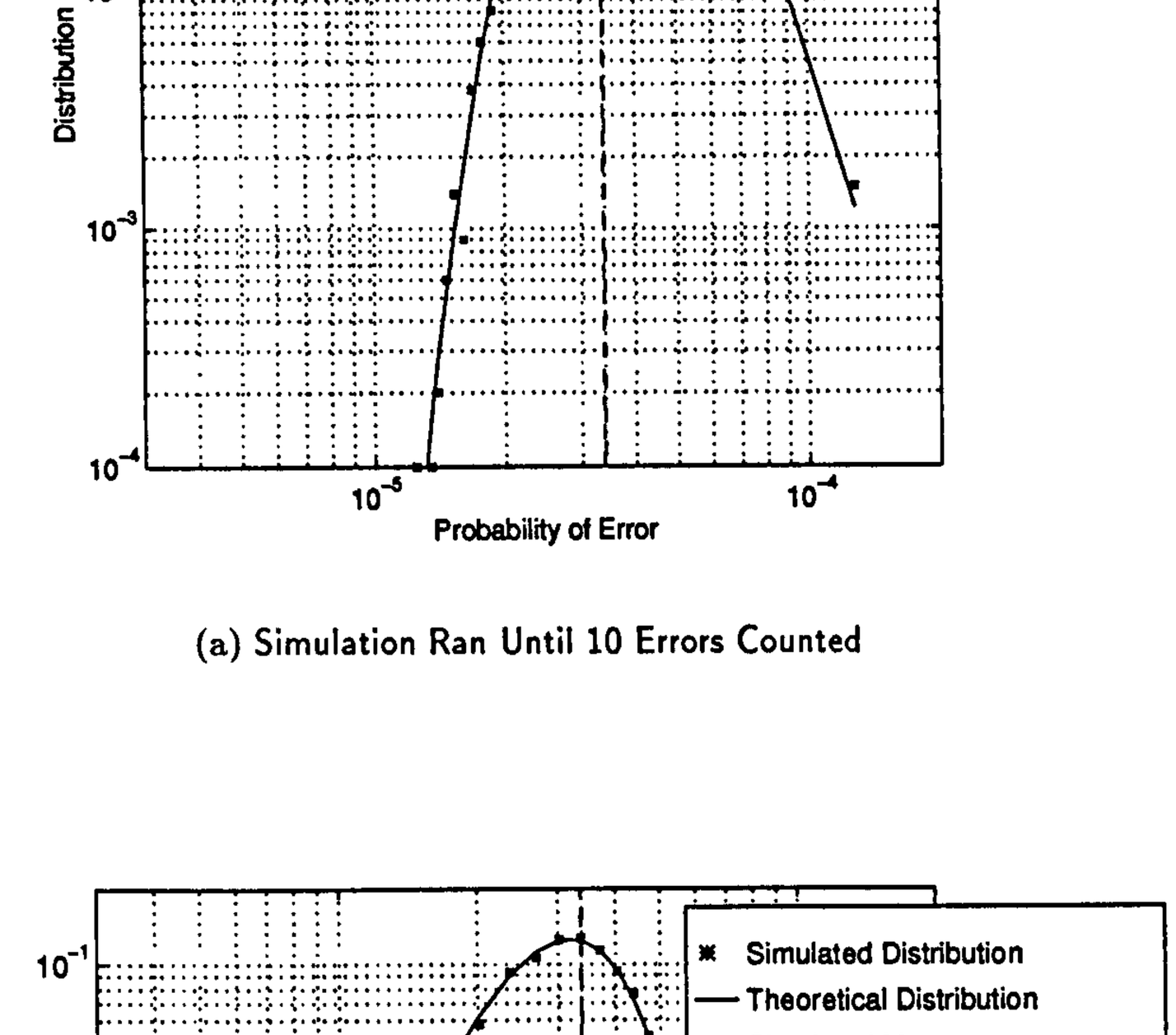
The challenge for the Importance Sampling technique is to reduce the variance, or the width of the distribution shown in figure 2.3 for a given number of sample points. This is tantamount to achieving the same distribution with a reduced number of sample points. This is done in practice by *biasing* the noise distribution in a predetermined manner in order to increase the number of errors that occur, and then un-biasing the resultant estimate of the bit-error rate.

The first bias functions that were suggested for the Importance Sampling technique simply resulted in another Gaussian function but with increased variance. The use of such a bias function can reduce the variance of the resulting distribution as is shown in figure 2.4. These results where produced by biasing the noise by a factor of 2 and only counting one error! One can see that we get a lower variance on the estimate for fewer errors counted.

The techniques of importance sampling are particularly important at high signal to noise ratios, where the corresponding BER is low, or for complex systems which may take an inordinate amount of time to run even at low signal to noise ratios.




(a) Simulation Ran Until 10 Errors Counted




(b) Simulation Ran Until 297378 Symbols Counted

Figure 2.3: Distribution of the estimate of the probability of error. The solid curve shows the theoretical distributions. Plots were produced from 10000 Monte-Carlo simulation runs with random uniformly distributed noise seeds.

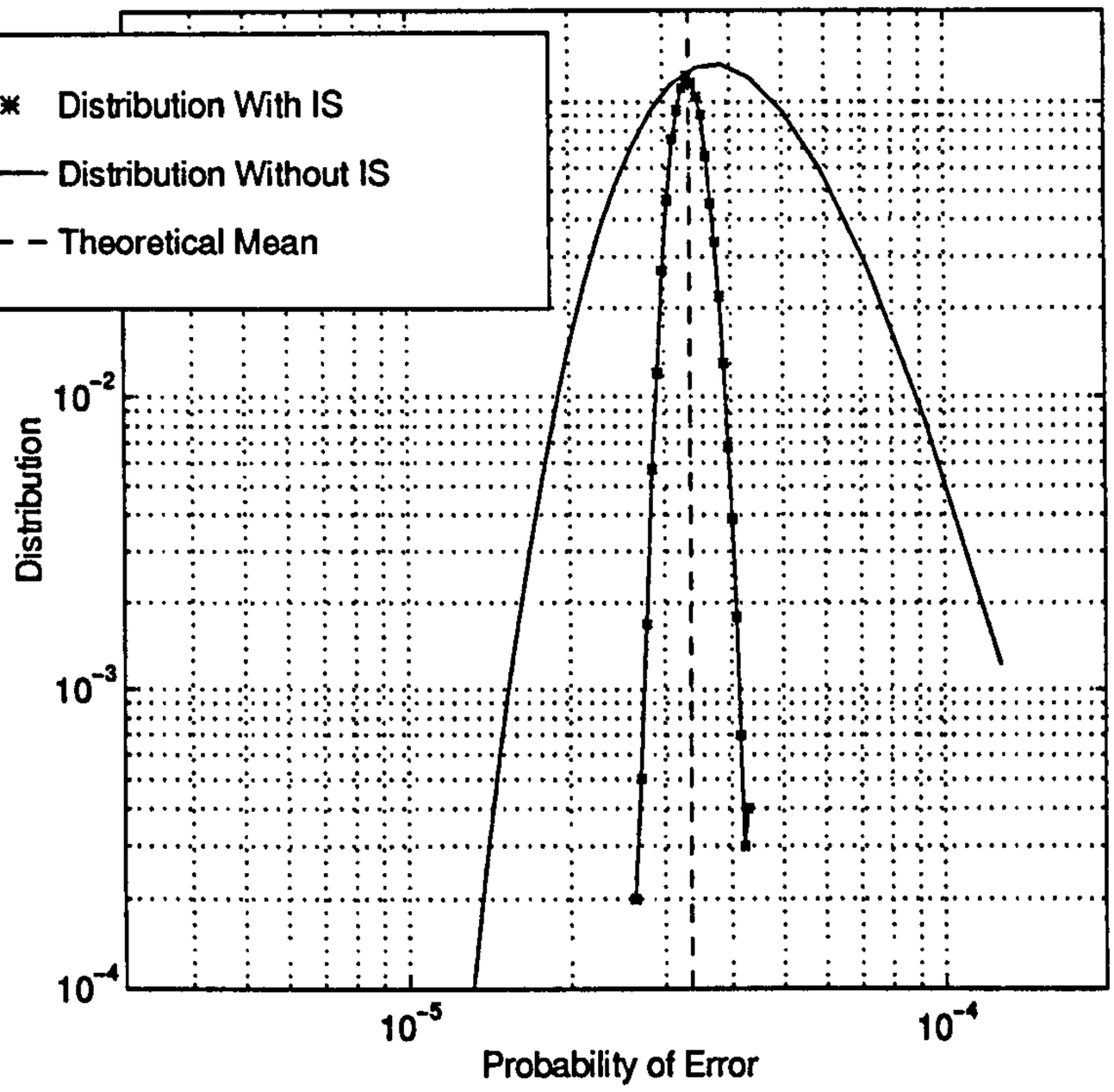



Figure 2.4: This illustrates the use of the Importance Sampling technique to reduce the variance of the resulting BER distribution.

### 2.5.2 Other Distributions used for Importance Sampling

In the development of the subject [32], the first bias function that was used resulted in another Gaussian noise distribution but with a greater variance. The results presented in the previous section were those obtained when a Gaussian function was used as the bias function but with a greater variance than the "underlying" noise variance. Recent work has used different distributions which have better performance in terms of the sample size savings they offer, or, equivalently, the variance reduction that they allow. Another distribution consists of a translated Gaussian distribution. Such a distribution has a greater sample size saving factor than the simple Gaussian distribution by virtue of the fact that, if the translation is chosen correctly, it will result in a maximum amount of errors before the BER is biased to give the final estimate. Other more recent distributions have an even better performance than the Gaussian or translated Gaussian and these are the Gaussian Tail distribution (GT) and the Rayleigh Tail distribution (RT). The GT distribution is theoretically optimum in that, if the threshold in the system is known, then the BER can be estimated *with zero variance!* However, there is a rather circular argument here in that knowing the system threshold is tantamount to knowing the BER anyway. For most systems one can only make a guess at what the BER will be. In such situations the optimum distribution to use is the RT distribution.

Thus, in terms of practical application, Importance Sampling does not provide all the

benefits that at first sight one might suspect. This is because of the ambiguity in the theory which implies that one gets the best sample size savings the more one knows about the system threshold, which, in turn is tantamount to knowing the BER, which is what the simulation should be attempting to estimate in the first place! It may be possible to use techniques which rely on the system *adaptively* adjusting the bias function to produce an accurate estimate of the underlying BER.

### 2.5.3 Relevance of the Work to CAAD of Communication Systems in Fading Channels

Much of the theory of Importance Sampling in the literature is concerned with improving the accuracy of the BER estimate, or alternatively, reducing the number of symbols that one has to count in order to obtain a BER estimate with a given accuracy, where these estimates are obtained in additive white Gaussian noise. In the simulation of mobile communication systems not only is there an additive noise component but there is also a multiplicative noise component (fading) and so the situation is somewhat different to that for which the Importance Sampling theory was developed. It is possible in the simulation of a mobile system to bias the additive component of the noise, as one would do conventionally, but it is difficult to see how Importance Sampling could be used to bias the multiplicative noise component and hence reduce simulation sample sizes. The problem of having to use large sample sizes in order to obtain accurate BER estimates is particularly acute in the simulation of slowly fading mobile systems. One could envisage a situation where, because of the slow fading, a large number of bits would be in error during a deep fade biasing the *actual* BER upwards. In such situations one would have to be sure that the simulation ran over a large number of such fades in order to get an accurate BER estimate.

# Chapter 3

# Modulation Schemes for Digital Radio

## 3.1 Introduction

In this chapter we introduce those modulation schemes which are popularly used for mobile radio communications. There are two generic types of schemes which are diametrically opposed in character, namely *wideband* and *narrowband* modulation schemes. Simply put, the demarcation between these schemes is given by $B \times T \gg 1$ for wideband schemes and $B \times T \ll 1$ for narrowband schemes, where $B$ is the bandwidth that the modulated signal requires and $T$ is the symbol period. At the present time there is some argument as to which schemes will ultimately prevail for the second-generation civilian digital cellular radio systems. The contenders are the CDMA spread spectrum system [33] and systems based on the narrowband modulation schemes $\pi/4$-QPSK and GMSK as used by the American Digital Cellular (ADC) and Special Mobile Group (GSM) [34]. It is only by extensive simulation and field-work that the relative merits of these schemes can be assessed.

The emphasis of the work undertaken in this thesis is on narrowband systems.

## 3.2 Wideband Modulation Schemes

Wideband modulation can be achieved in the analogue domain by frequency modulation with a large modulation index or, in the digital domain, by using a frequency-hopped carrier or modulation of the signal with a pseudo-random code sequence with a high chip rate.

The use of spread spectrum signals in Land Mobile Radio follows from a knowledge of the multipath structure of the channel. The channel can be modelled as a time-variant transversal filter with taps spaced by a minimum value of delay $\tau$. The total delay spread [30], $T_m$, is then accounted for by having $T_m/\tau$ taps. If an impulse were to be sent through such a channel then $T_m/\tau$ impulses, each multiplied by a complex coefficient, would be output with a delay between each of $\tau$ seconds. Alternatively, if a finite bandwidth signal were to be sent through the time-variant transversal filter channel it would be required that the bandwidth

of the signal used be given as $W \gg 1/\tau$ for each of the multipath components to be resolved. Thus by sending a wideband signal through the multipath channel one is able to resolve each component of the multipath, each having faded *independently* of any other component by virtue of the fact that they have travelled from transmitter to receiver by distinct paths. These components can then be weighted optimally and combined at the receiver to ameliorate the effects of fading. It can be seen that the use of wideband signals is a valid way of achieving diversity.

### 3.2.1 RAKE Receivers for Wideband Mobile Radio

The technique above for achieving improved performance over multipath channels was first suggested by Price in 1958 [35] for HF Communications via the ionosphere and later work was done in applying the concept to Urban Digital Radio [36]. Price coded a binary signal in to a wideband signal by using *pseudo noise* or PN - sequence, now referred to as Direct-Sequence Spread Spectrum (DS – SS). Price coined the phrase RAKE to describe this type of receiver. The receiver generates its own copy of the wideband signal to correlate with the received signal. The wideband signals used for spread spectrum must therefore exhibit good auto-correlation and cross-correlation properties. PN-sequences have the useful properties for these purposes. For $n = 1023$ the sidelobes will be 30dB. Thus, very long PN-sequences exhibit ideal correlation properties in the sense that they exhibit low sidelobe levels.

The correlator output on each tap in the RAKE is thresholded to decide whether a multipath component exists on that tap or not to avoid noise-only contributions. The correlator outputs are then multiplied by weighting factors proportional to the path strengths. In the model used here the channel coefficients were fed forward (assumed to be estimated without error) and each correlator output multiplied by the complex conjugate of the channel coefficient. This weighting ensures that stronger signals are biased over weaker ones. This weighting function can be highly nonlinear depending on the type of channel model used (more of this later) but for the simple model adopted here and in [30] where the path strengths and delays are known *a priori* and are equal the decision variable is given as a sum over all of the squared outputs. It is found that ideal RAKE processing over $L$ independent multipath components is equivalent to $L^{th}$ order diversity.

The block diagram of the RAKE equaliser for binary antipodal signals is shown in figure 3.1. In this system the channel coefficients are estimated from the received signals by using a decision feedback method [30]. Systems that derive the channel coefficients from *a posteriori* information are known as *sounding* receivers [36], the sounding signals can either be the data itself or a specific sequence for sounding only. In RAKE the data is used to sound the channel. In the figure, the filters used in estimating the channel coefficients have to be wide enough to let the channel variations through due to the Doppler spreading of the signal but narrow enough to reject noise. In the RAKE system illustrated the Butterworth filters were used with the same 3dB bandwidth as the filters that were used to build up the channel model. Note that as the channel coefficients are now being estimated there will be a residual error rate at high signal to noise ratios giving a 'bottoming out' of the BER plots.

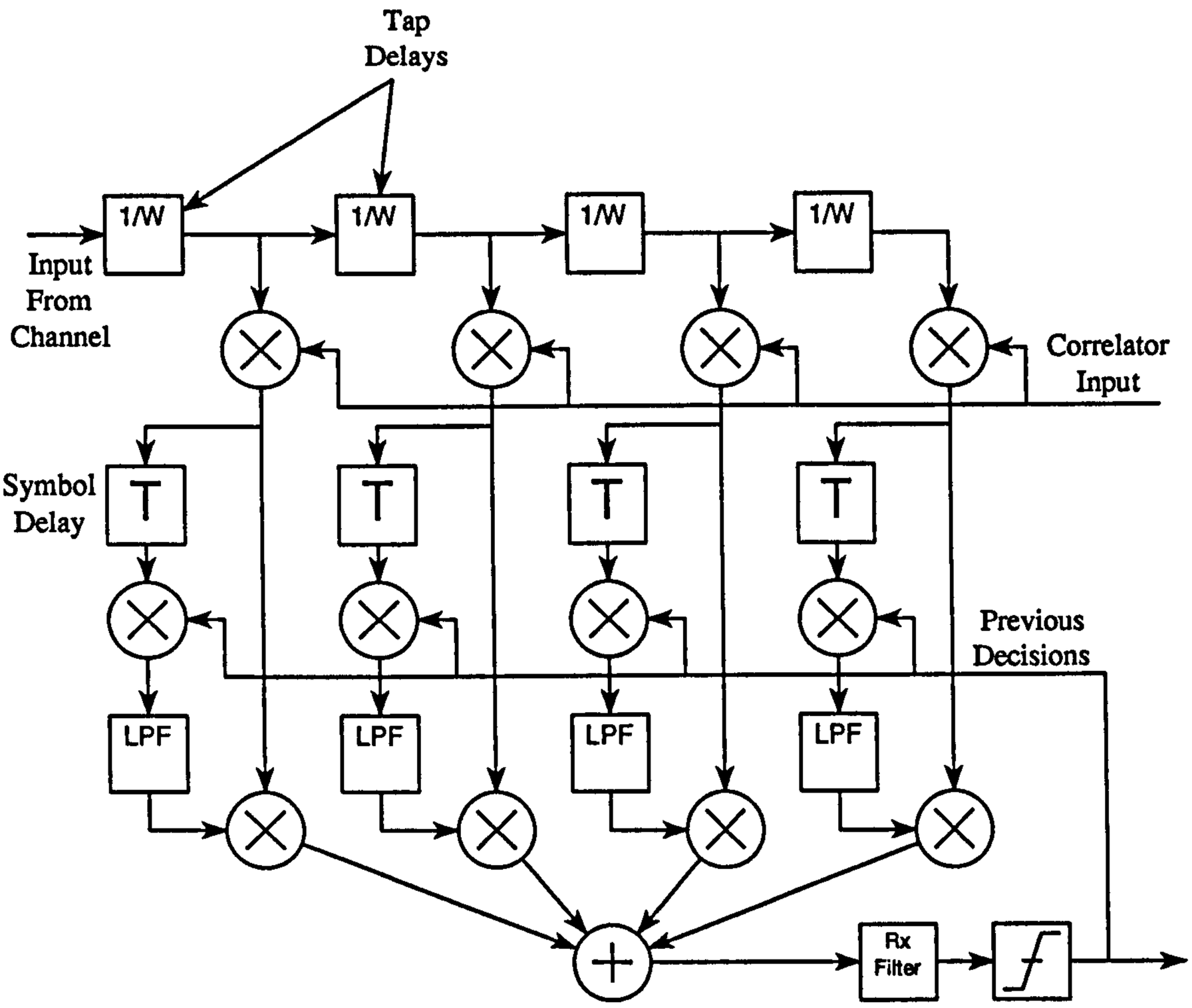



Figure 3.1: Block diagram of a RAKE receiver for binary-antipodal data using coherent detection.

This type of processing is unrealistic for the Land Mobile Channel as one generally is unable to estimate the phase of the signal. That is, coherent techniques are generally avoided in favour of differentially coherent or limiter-discriminator type methods. A RAKE processor can also be configured to operate with differentially encoded antipodal signals.

### 3.2.2 Inadequacy of Wideband Models

Ultimately, an accurate model of the wideband multipath channel must be based on empirical data rather than mathematical axioms [36].

The wideband models used above for the RAKE processing are somewhat simplistic in that there are a fixed number of paths, the paths all have identical statistical variations and the delays between the paths are constant and equal. The channel models in [36] are developed empirically from measured data over different urban multipath environments and as such the models are more reliable.

## 3.3 Narrowband Modulation Schemes

GMSK is a popular modulation scheme for digital cellular (GSM) and digital cordless (DECT & CT-2) communications. Together with $\pi/4$-QPSK, as used for the American and Japanese Digital Cellular (ADC & JDC) systems, these represent the most popular narrowband schemes used for mobile personal communications [34] at the present time. In this section we shall describe the format of these modulation schemes and their characteristics. Discussions relating to receiver architectures and performance shall be deferred until later.

It is noted that these schemes are essentially of constant amplitude. That is to say, all the modulating information is in the phase, and not the amplitude, of the signal. More spectrally efficient techniques will be exploited in the future where information is also encoded into the amplitude of the signal. Such Quadrature Amplitude Modulation (QAM) schemes have been discouraged up till now because of the amplitude variations that are imposed on the signal by the fading channel. This makes it difficult for a receiver to differentiate between amplitude fluctuations in the received signal due to the modulation and amplitude fluctuations due to the channel variations. There are techniques for alleviating these effects and may become more predominant in the future [37], [38], [39].

## 3.4 Raised-Cosine Filtering

For the QPSK and $\pi/4$-QPSK modulation schemes studied as part of this work we have generally assumed that the transmitter and receiver filters are both root-Nyquist with cosine-rolloff characteristics $H(f)$. These filters are needed in order to bandlimit the signal prior to transmission. Note that the filtering is done after the modulation. We will consider a scheme next, GMSK, in which the bandlimiting of the signal is accomplished by pre-filtering the baseband data before modulation.

The overall transfer function of the link, after matched filtering, is $G(f) = H(f)\,H^*(f)$, which is a full-Nyquist cosine rolloff shape. We thus define the following transfer characteristic for $H(f)$.

$$H(f) = \begin{cases} H_0, & |f| < \dfrac{(1-\alpha)}{2T} \\ H_0 \cos\left\{\dfrac{\pi}{2\,\alpha}\left(fT - \dfrac{1}{2} + \dfrac{\alpha}{2}\right)\right\}, & \dfrac{(1-\alpha)}{2T} \leq f < \dfrac{(1+\alpha)}{2T} \\ H_0 \cos\left\{\dfrac{\pi}{2\,\alpha}\left(fT + \dfrac{1}{2} - \dfrac{\alpha}{2}\right)\right\}, & -\dfrac{(1+\alpha)}{2T} \leq f < -\dfrac{(1-\alpha)}{2T} \\ 0, & |f| \geq \dfrac{(1+\alpha)}{2T} \end{cases} \tag{3.1}$$

In the above equations $0 \leq \alpha \leq 1$ is the excess bandwidth of the pulse. The larger the value of $\alpha$ the less tightly confined the pulse is in the frequency domain and the less the inter-symbol interference for samples not taken at the optimum sampling point. Nyquist pulses have zero inter-symbol interference at the optimum sampling points [30] but this optimum sampling is something that we have to adaptively estimate in our receiver. As we shall see later, to implement some receiver functions we will need the derivatives of the equations in (3.1). These are as follows:

$$H'(f) = \begin{cases} -\dfrac{\pi\,T\,H_0}{2\,\alpha} \sin\left\{\dfrac{\pi}{2\,\alpha}\left(fT - \dfrac{1}{2} + \dfrac{\alpha}{2}\right)\right\}, & \dfrac{(1-\alpha)}{2T} \leq f < \dfrac{(1+\alpha)}{2T} \\ -\dfrac{\pi\,T\,H_0}{2\,\alpha} \sin\left\{\dfrac{\pi}{2\,\alpha}\left(fT + \dfrac{1}{2} - \dfrac{\alpha}{2}\right)\right\}, & -\dfrac{(1+\alpha)}{2T} \leq f < -\dfrac{(1-\alpha)}{2T} \\ 0, & \text{elsewhere} \end{cases} \tag{3.2}$$

Plots of $H(f)$ and $H'(f)$ are shown in figure 3.2 for the Root-Raised Cosine filters represented by (3.1) & (3.2) for $\alpha = 0.5$.

Raised Cosine filter impulse responses for 20% and 80% excess bandwidths are shown in figure 3.3. Also shown is the filter impulse response with $x/\sin(x)$ equalisation which is necessary when the input data to the filter is square-wave data. This is because the Raised-Cosine impulse response is inter-symbol interference free only when it is convolved with a train of impulses. The MATLAB design routine for these filters is included in Appendix A.

Figure 3.4 shows the eye-diagram that results when one filters binary-antipodal data with 20% and 80% Raised Cosine filters. It is evident that at the sample points (0,1 & 2 in the figures) there is no inter-symbol interference irrespective of the filter excess bandwidth. This is the attractive property of Raised-Cosine over other, non-nyquist, pulse shapes. However, one can see that there is a large amount of inter-symbol interference, or data noise, at points between the optimum sampling points and this is worse as the filter excess bandwidth is reduced.

Figure 3.5 shows the Power Spectral Density (PSD) of 20% and 80% Raised-Cosine filtered data. The frequency axis is normalised to the baud rate.

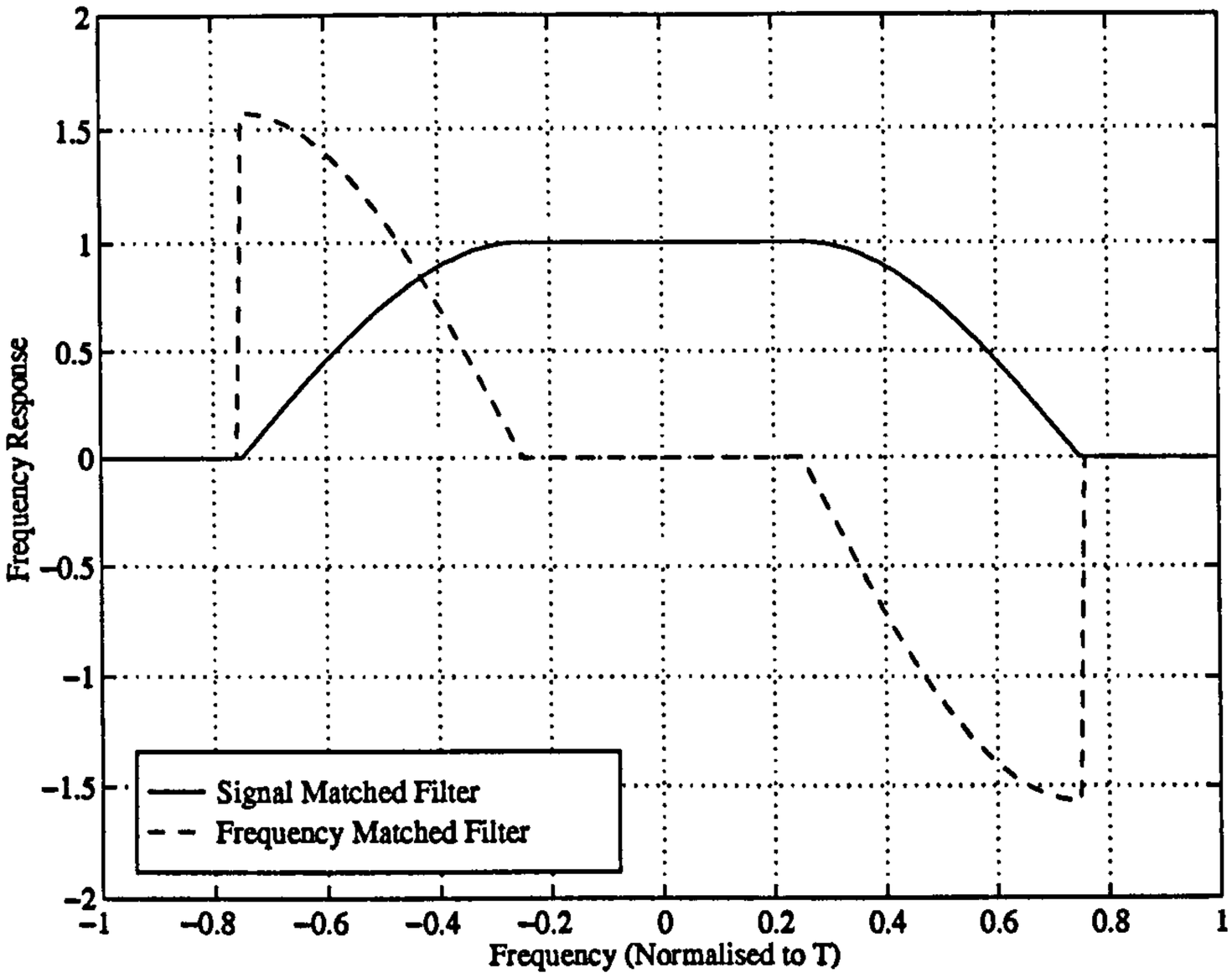



Figure 3.2: $H(f)$ (Signal Matched Filter) and $H'(f)$ (Frequency Matched Filter) for Root-Raised Cosine filter with $\alpha = 0.5$.

## 3.5 Gaussian Minimum Shift Keying

MSK can be considered to be both a form phase and frequency modulation. As such it can be considered to be a *linear* and an *nonlinear* modulation scheme. GMSK is formed by pre-filtering the *baseband* modulating data with a Gaussian filter before applying it to a frequency modulator. In the ideal case, therefore, limitation of spectral spreading is achieved by filtering at baseband rather than at RF as in many modulation schemes. This modulation scheme was first presented in [40].

### 3.5.1 Mathematical Development of the GMSK Signalling Format

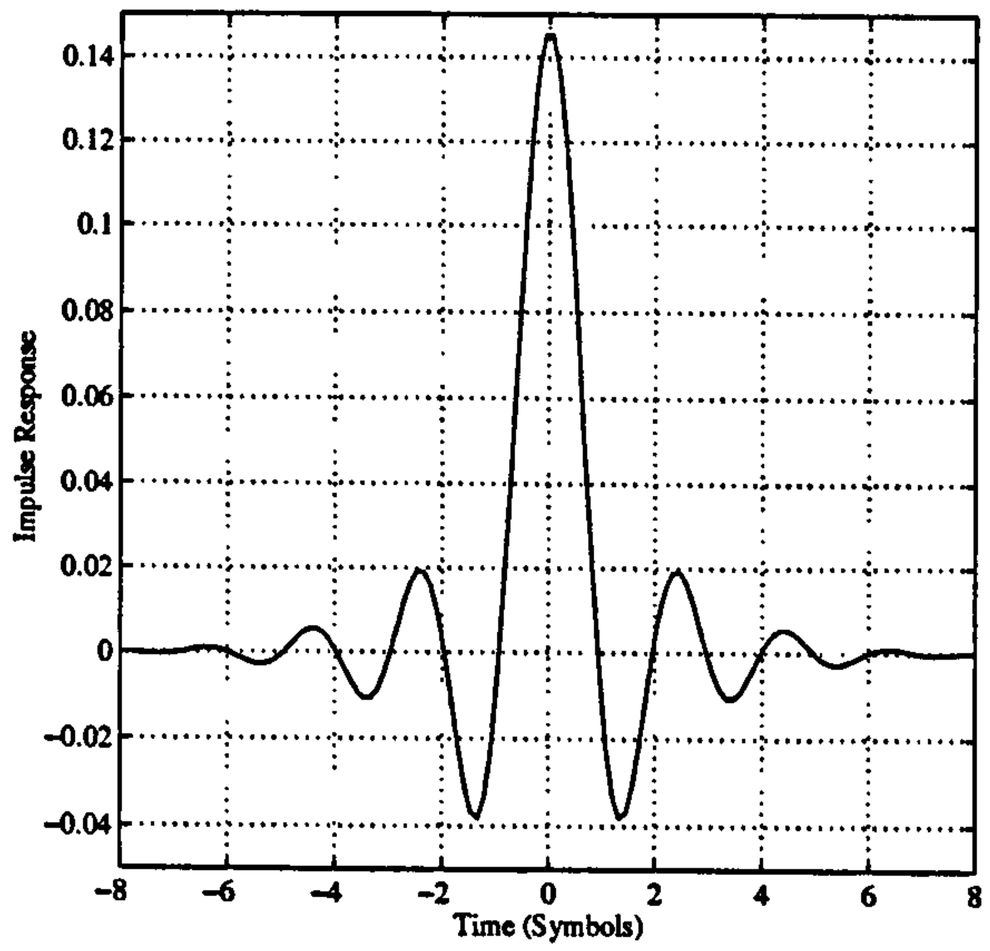
The Gaussian Minimum Shift Keyed modulation consists of applying a non-return-to-zero (NRZ) data stream to a Gaussian filter and thence on to an FM modulator with a modulation index of $\mu = \frac{1}{2}$. This scheme is illustrated in figure 3.6.

The analysis thus starts with the Gaussian lowpass filter that filters the data prior to modulation. This filter has the impulse response given by

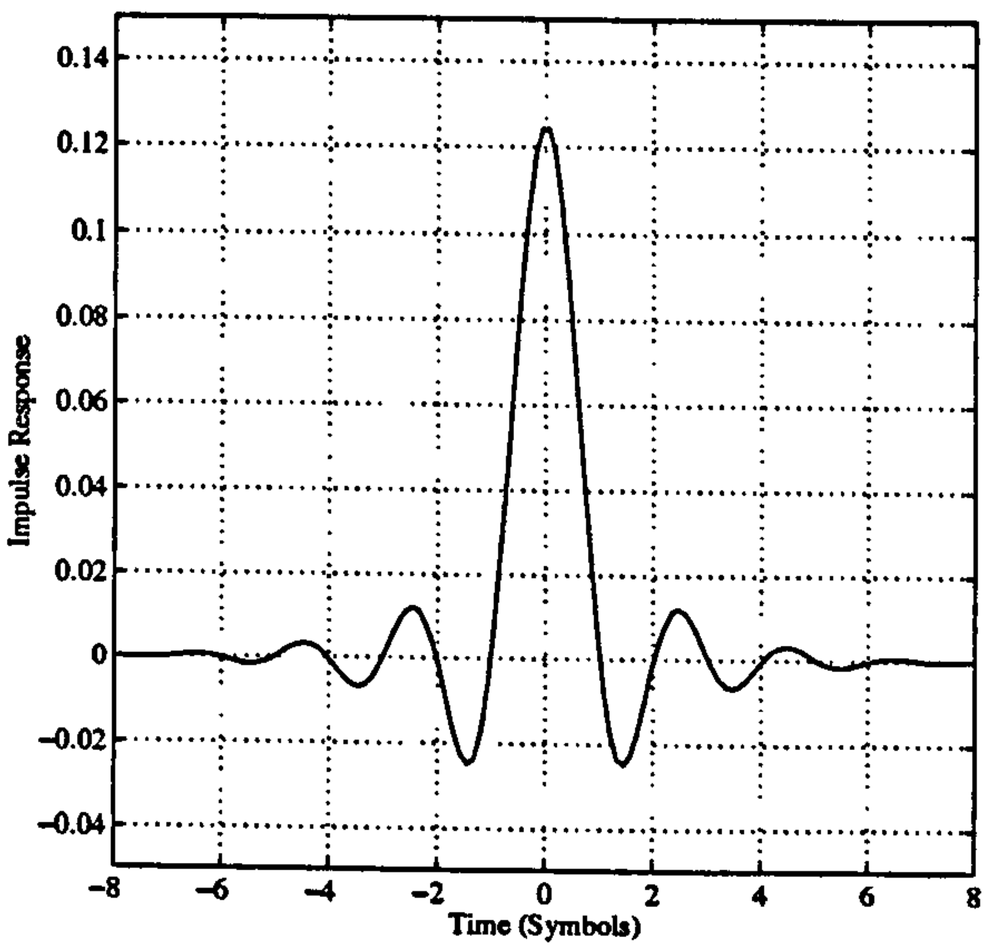
$$g(t) = \frac{\sigma}{\sqrt{2\pi}} \exp\left(-\frac{\sigma^2 t^2}{2}\right) \tag{3.3}$$

where $\sigma$ is

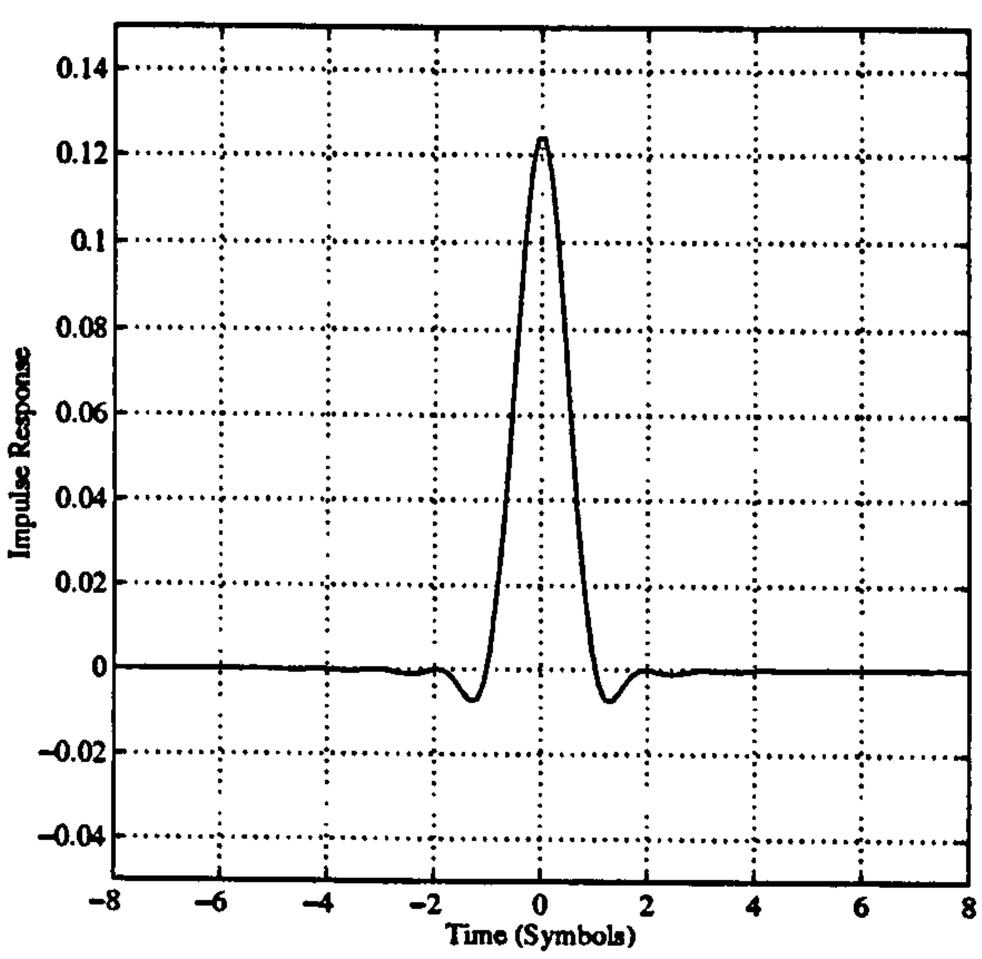
$$\sigma = \frac{2\pi B_t}{\sqrt{\log(2)}} \tag{3.4}$$




(a) $\alpha = 0.2, x/\sin(x)$ equalisation

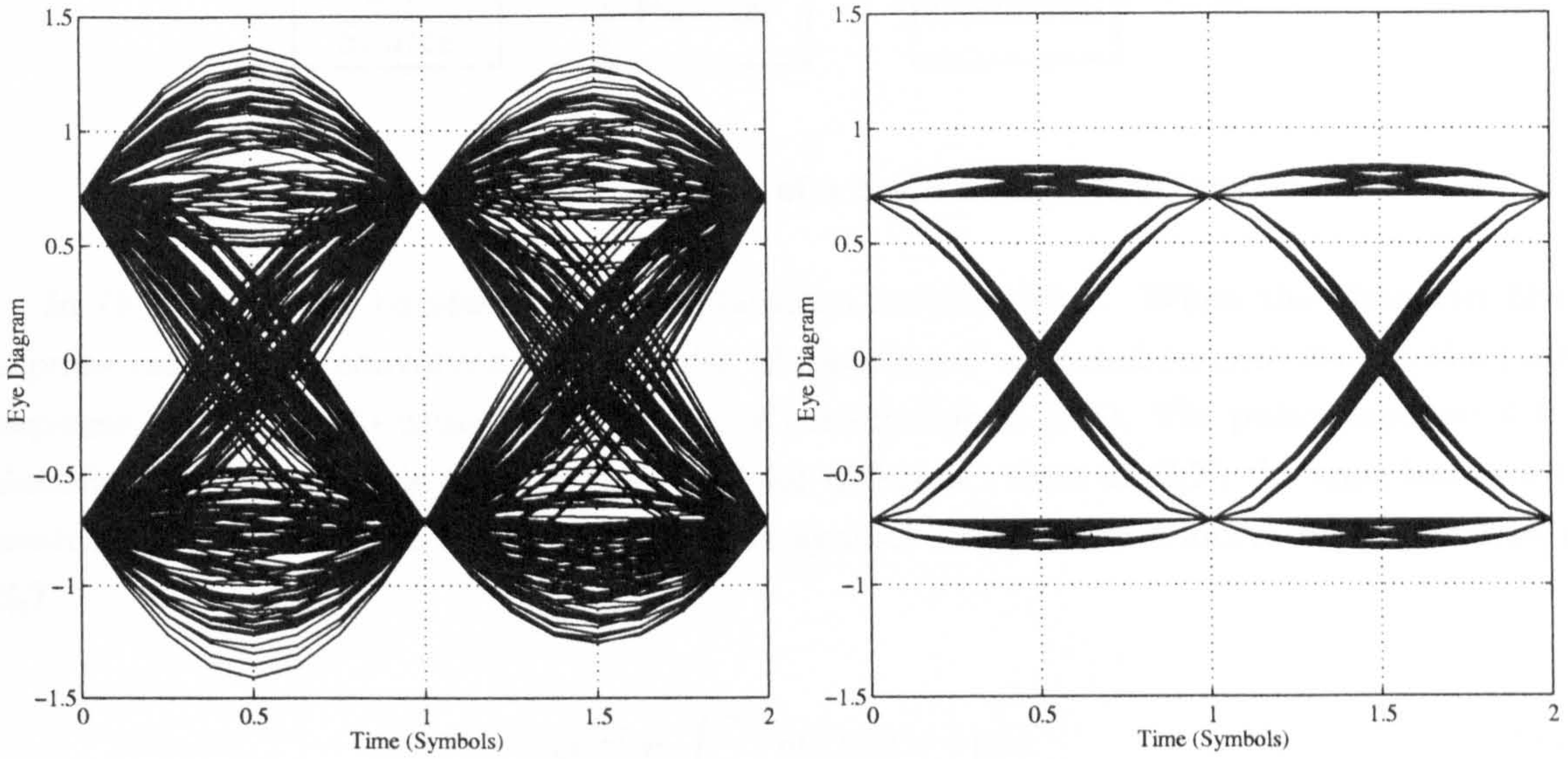



(b) $\alpha = 0.2$



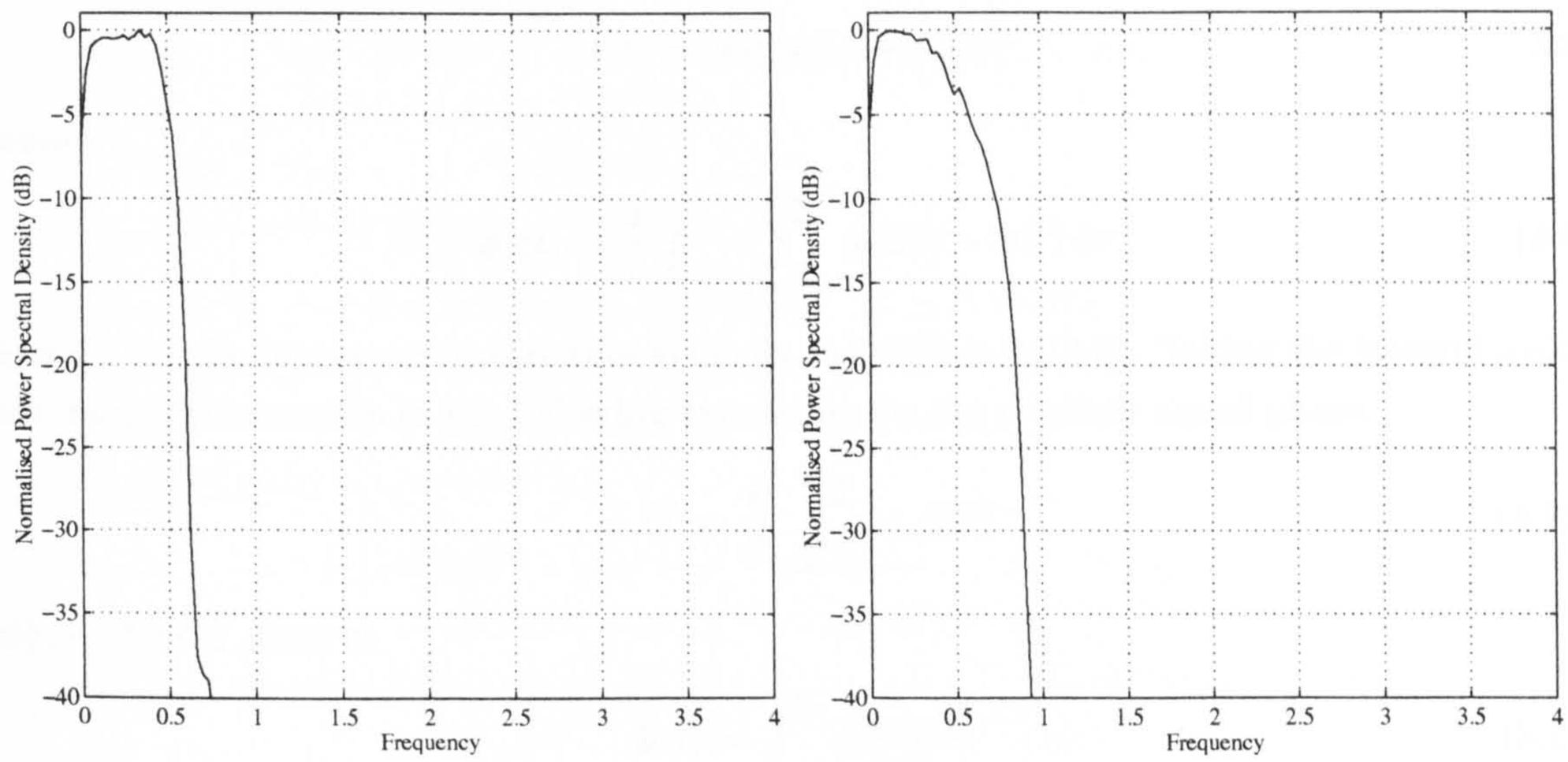

(c) $\alpha = 0.8$

Figure 3.3: Raised Cosine filter impulse responses




(a) $\alpha = 0.2$ (b) $\alpha = 0.8$

Figure 3.4: Eye Diagrams of Raised Cosine Filtered Data




(a) $\alpha = 0.2$ (b) $\alpha = 0.8$

Figure 3.5: Power Spectral Density of Root Raised Cosine Filtered QPSK


Figure 3.6: Block Diagram of a Serial GMSK Modulator

In (3.4) $B_t$ is the bandwidth of the Gaussian lowpass filter. When the Gaussian filter impulse response is convolved with a pulse of duration $T$ centered on zero we get the pulse response, $G(t)$, in (3.6) using the definition of convolution in (3.5). The pulse response of the Gaussian lowpass filter is show in figure 3.7 for different values of $B_tT$, the time-bandwidth product. Notice from the figure that the inter-symbol interference is increased for decreasing $B_tT$.

$$G(t) = \int_{-\infty}^{\infty} p(\tau)g(t-\tau)\,d\tau \tag{3.5}$$

$$G(t) = \frac{1}{2}\left\{\operatorname{erf}\left(-\frac{\sigma t_1}{\sqrt{2}}\right) + \operatorname{erf}\left(\frac{\sigma t_2}{\sqrt{2}}\right)\right\} \tag{3.6}$$

where $t_1$ and $t_2$ are defined as

$$t_1 = t - \frac{T}{2} \text{ and } t_2 = t + \frac{T}{2} \tag{3.7}$$

The output from the GMSK modulator is given as

$$s(t) = \cos[2\pi f_c t + \phi_s(t)] \tag{3.8}$$

where

$$\phi_s(t) = \frac{\pi}{2}\int_{-\infty}^{t} \sum_{n=-\infty}^{+\infty} a_n G(\tau - nT)\,d\tau \tag{3.9}$$

and $a_n$ are the binary-antipodal data symbols and $G(t)$ is in (3.6). Taking the integral inside the summation results in the following expression for the absolute signal phase.

$$\phi_s(t) = \frac{\pi}{2}\sum_{n=-\infty}^{+\infty} a_n \Phi(t) \tag{3.10}$$

where $\Phi(t)$ is given as

$$\Phi(t) = \int_{-\infty}^{t} G(\tau)\,d\tau \tag{3.11}$$

Putting (3.6) into (3.11) results in a closed-form solution to the signal phase given by (3.12).

$$\Phi(t) = \frac{1}{2} + \frac{t_2}{2}\operatorname{erf}\left(\frac{\sigma t_2}{\sqrt{2}}\right) - \frac{t_1}{2}\operatorname{erf}\left(\frac{\sigma t_1}{\sqrt{2}}\right) + \frac{1}{\sqrt{2\pi}\sigma}\left\{\exp\left(-\frac{\sigma^2 t_2^2}{2}\right) - \exp\left(-\frac{\sigma^2 t_1^2}{2}\right)\right\} \tag{3.12}$$

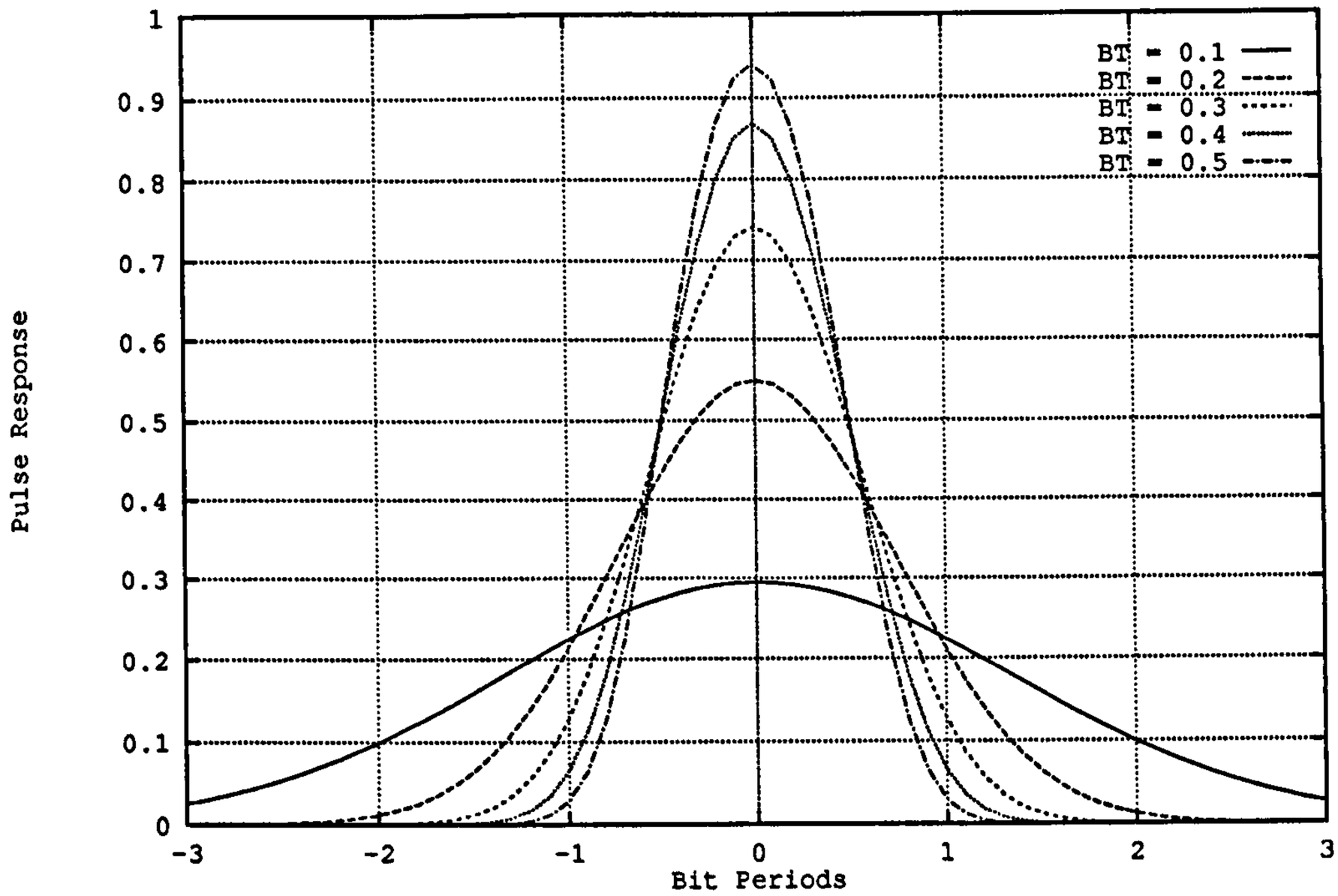



Figure 3.7: Pulse response of the Gaussian lowpass filter for different values of $B_tT$

Plots of the signal phase function for GMSK for different values of $B_tT$ is shown in figure 3.8. Note from the figure that if no pre-filtering where present, or equivalently $B_tT = \infty$, then the signal phase would be linear from $-T/2$ to $T/2$. This would then correspond to the case of Minimum Shift Keying.

### 3.5.2 Characteristics of GMSK

Here we look at how the modulation scheme can be characterised in terms of different signal representations and its minimum distance properties.

#### Phase Trellis Representation

The *phase trellis* for the GMSK signal is created by convolving the signal phase function, $\Phi(t)$, with a stream of binary antipodal symbols overlaying all the possible phase trajectories. For example, the phase trellis in figure 3.9 shows the different phase trajectories that the signal can take for each of the $2^6$ patterns of 6 symbols. The effects of inter-symbol interference on the phase trajectories is evident from the figure. Note that if there were no pre-filtering of the data then the trajectories in the phase trellis would consist of straight lines as is the case in Minimum Shift Keying. One can also appreciate by examination of the phase trellis that the optimum demodulator is one that takes this structure into account. This is why the optimum demodulators for such a scheme are the Maximum-Likelihood Sequence Estimators

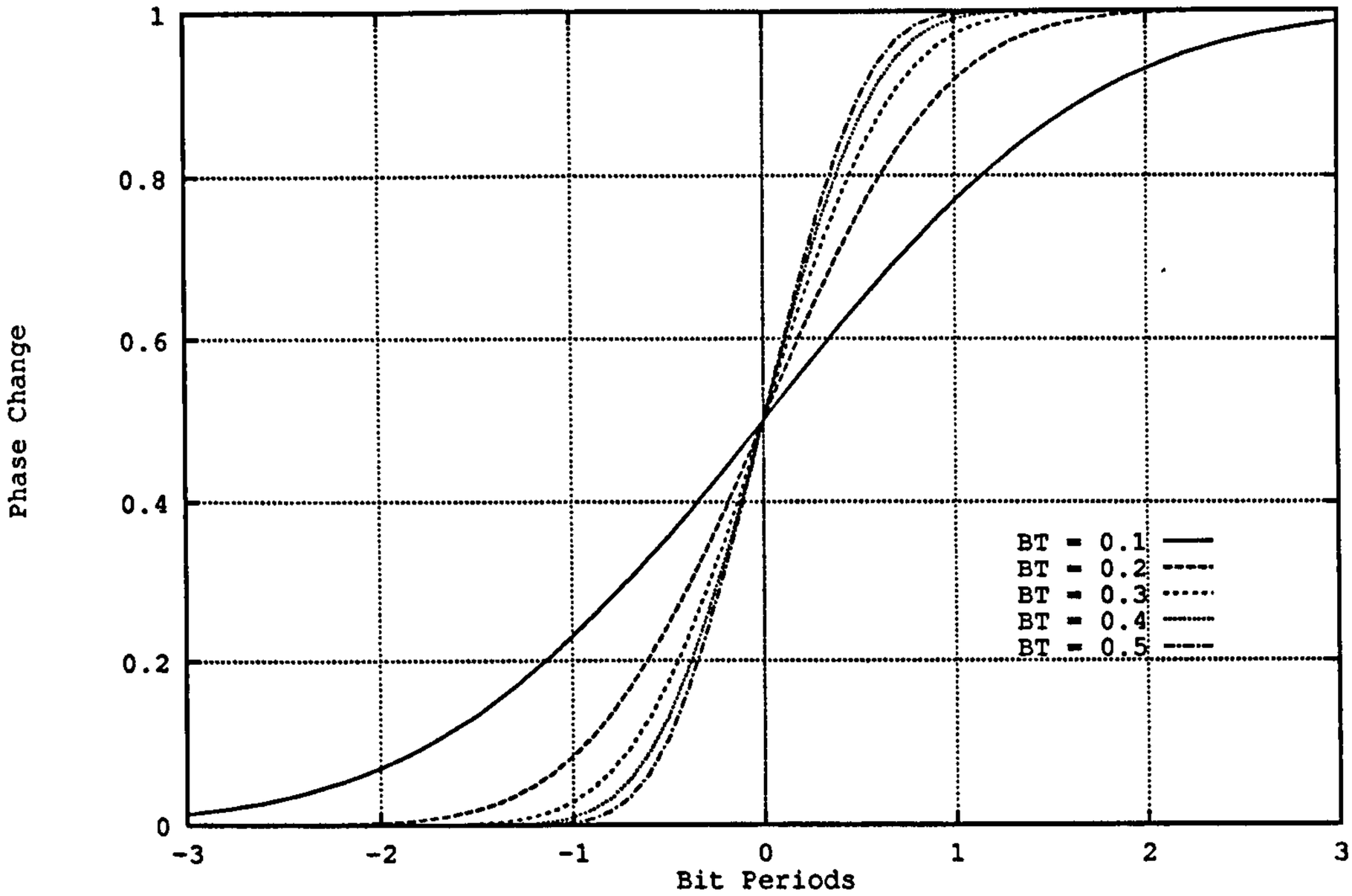



Figure 3.8: Signal phase function $\Phi(t)$ for different values of $B_tT$

(MLSEs), as will be discussed in section 7.6.1. Note that the phase trellis shown in figure 3.9 is an absolute phase trellis, the phase trellis can be reduced modulo-$2\pi$ to produce a *reduced* phase trellis. Such a reduced phase trellis is shown in figure 3.10.

### Scatter Diagram Representation

The phase trellis shows how the phase of the signal can change with time. An alternative signal representation, the *scatter diagram*, shows the phase (modulo-$2\pi$) and the amplitude of the signal. This is formed by plotting the in-phase and quadrature components of the GMSK signal at the sampling instants. Note that because of the inter-symbol interference there are extra phase states in the scatter diagram than there would be if there were no pre-filtering, that is, GMSK has extra phase states over the simpler MSK due to the data pre-filtering. This is shown in the scatter diagrams in figure 3.11 for GMSK with time-bandwidth products of 0.25 and 0.5. The plot also shows the constant-amplitude nature of the signalling scheme. The phase states shown in the scatter diagram can be derived from the analysis undertaken earlier. The displacement of the first state due to ISI is at 36° and 19° for time-bandwidths of 0.25 and 0.5 respectively. The calculated effects due to ISI are also drawn as a line on the scatter diagram.

The scatter diagram representation of the modulation will be used extensively later to gain some insight into the use of equalisers to reduce the inherent inter-symbol interference

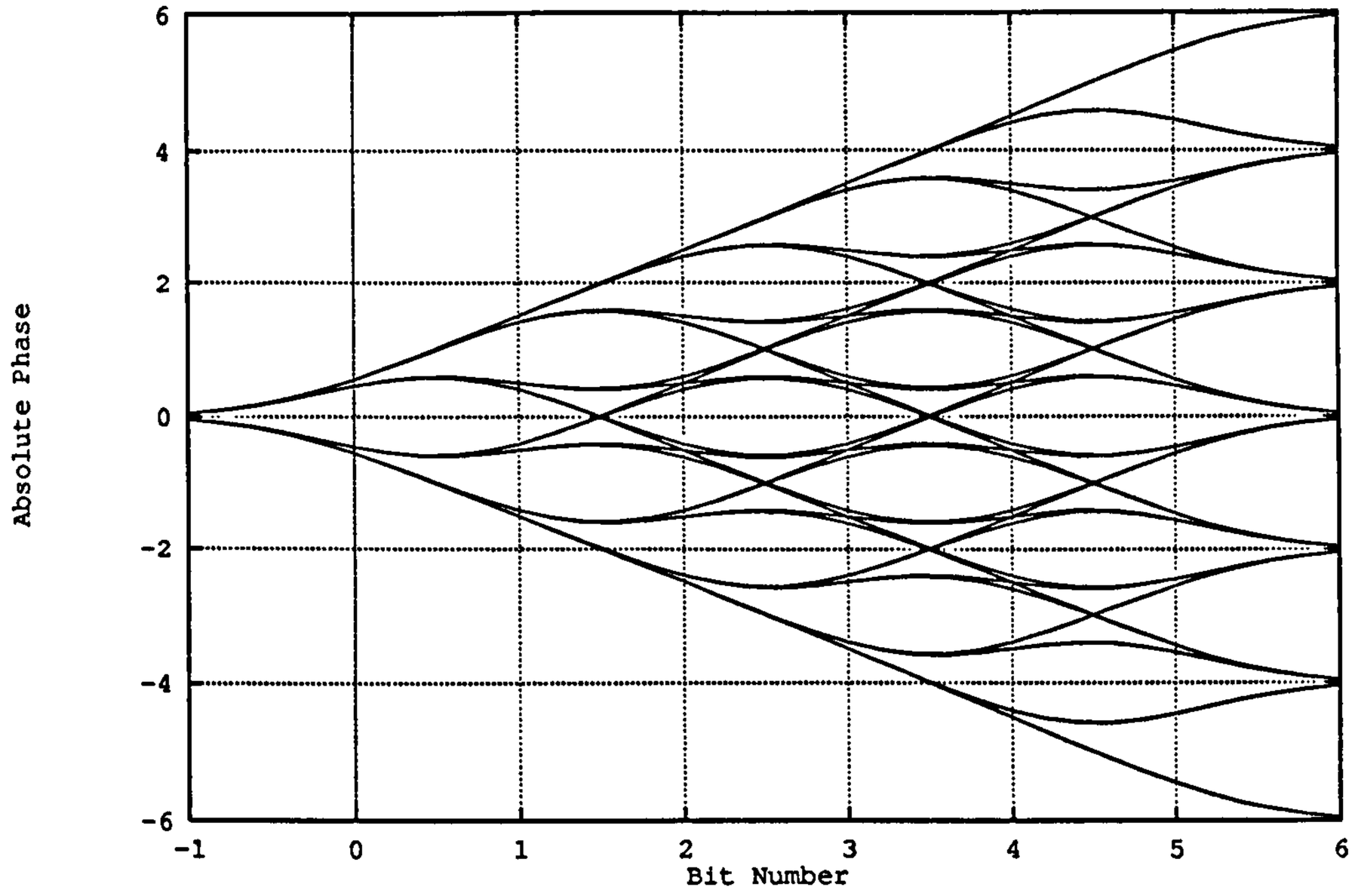



Figure 3.9: Absolute Phase Trellis for GMSK with $B_tT = 0.25$ over 6 Symbol Periods

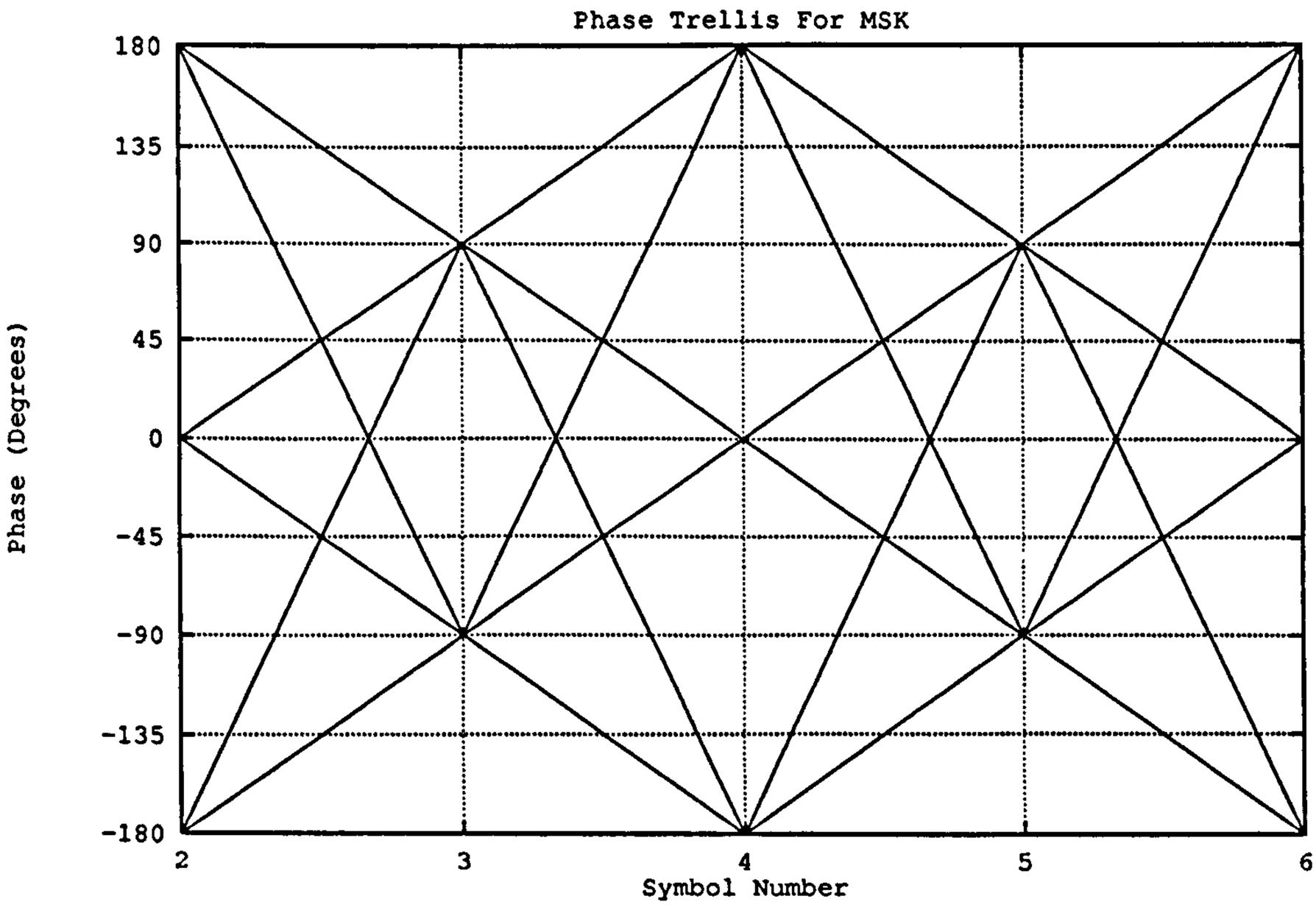
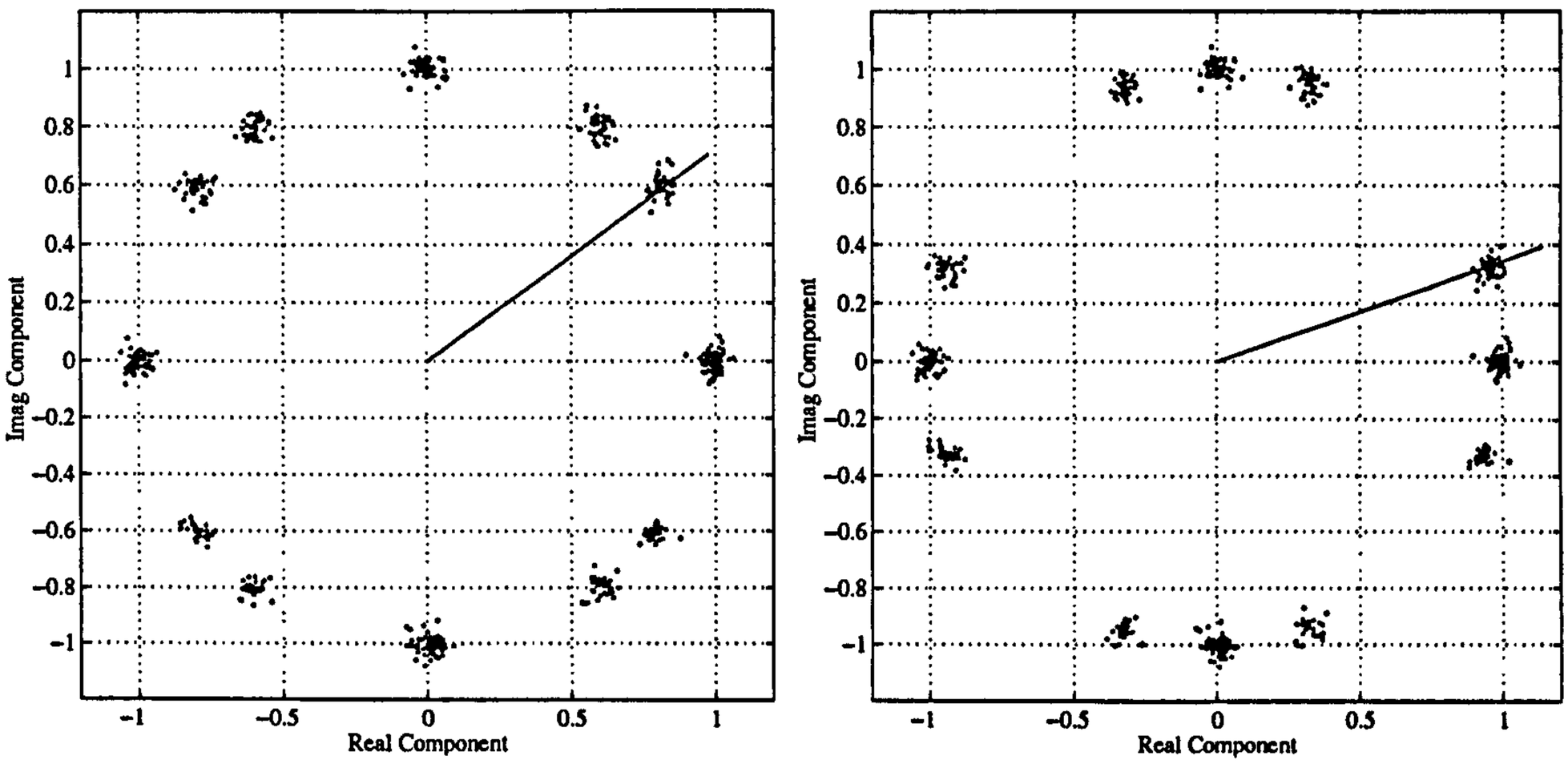



Figure 3.10: Reduced Phase Trellis for MSK




(a) $B_tT = 0.25$ (b) $B_tT = 0.5$

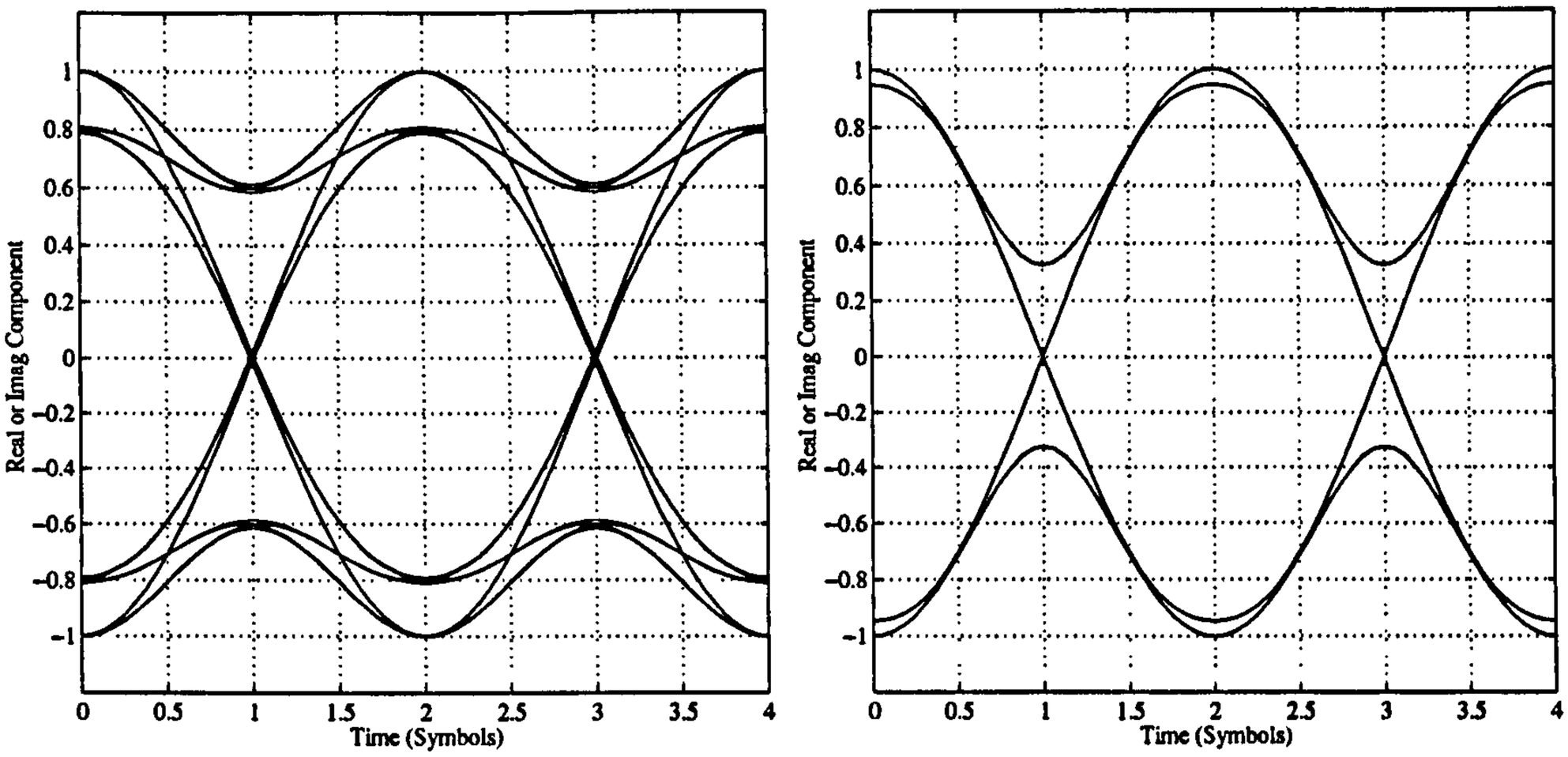
Figure 3.11: Scatter diagram of the GMSK waveform with different time-bandwidth products and $E_s/N_o = 40.0$dB

(see 7.6.4) which in effect manipulate points in the scatter diagram to increase the minimum Euclidean distance and thus reduce the bit-error rate.

Eye-diagrams for GMSK with time-bandwidth products of 0.25 and 0.5 are illustrated in figure 3.12. Note that this is the real component of the modulation scheme, the imaginary component would be offset by a bit period, hence the scheme can be viewed as a form of offset modulation. Note also that the symbol period corresponds to two bits. It is evident that the scheme is not ISI free at the optimum sampling points as is the case with Raised Cosine filtering.

### Phase Cylinder Representation

The phase cylinder representation of the GMSK signal is illustrated in 3.13 for a GMSK waveform with a time-bandwidth product of 0.25. This representation of the signal encapsulates all the information in the eye-diagrams and the scatter-diagrams. The eye-diagrams can be rendered by cutting the phase cylinder along its length and the scatter diagrams can be rendered by cutting along the cylinder cross-section at the sampling instants. Although it is more difficult to make any quantitative measurements from the phase cylinder representation, as compared with the scatter or eye-diagram representation, it does illustrate the constant-amplitude and continuous-phase nature of the GMSK signal.




(a) $B_tT = 0.25$ (b) $B_tT = 0.5$

Figure 3.12: Eye diagrams for GMSK

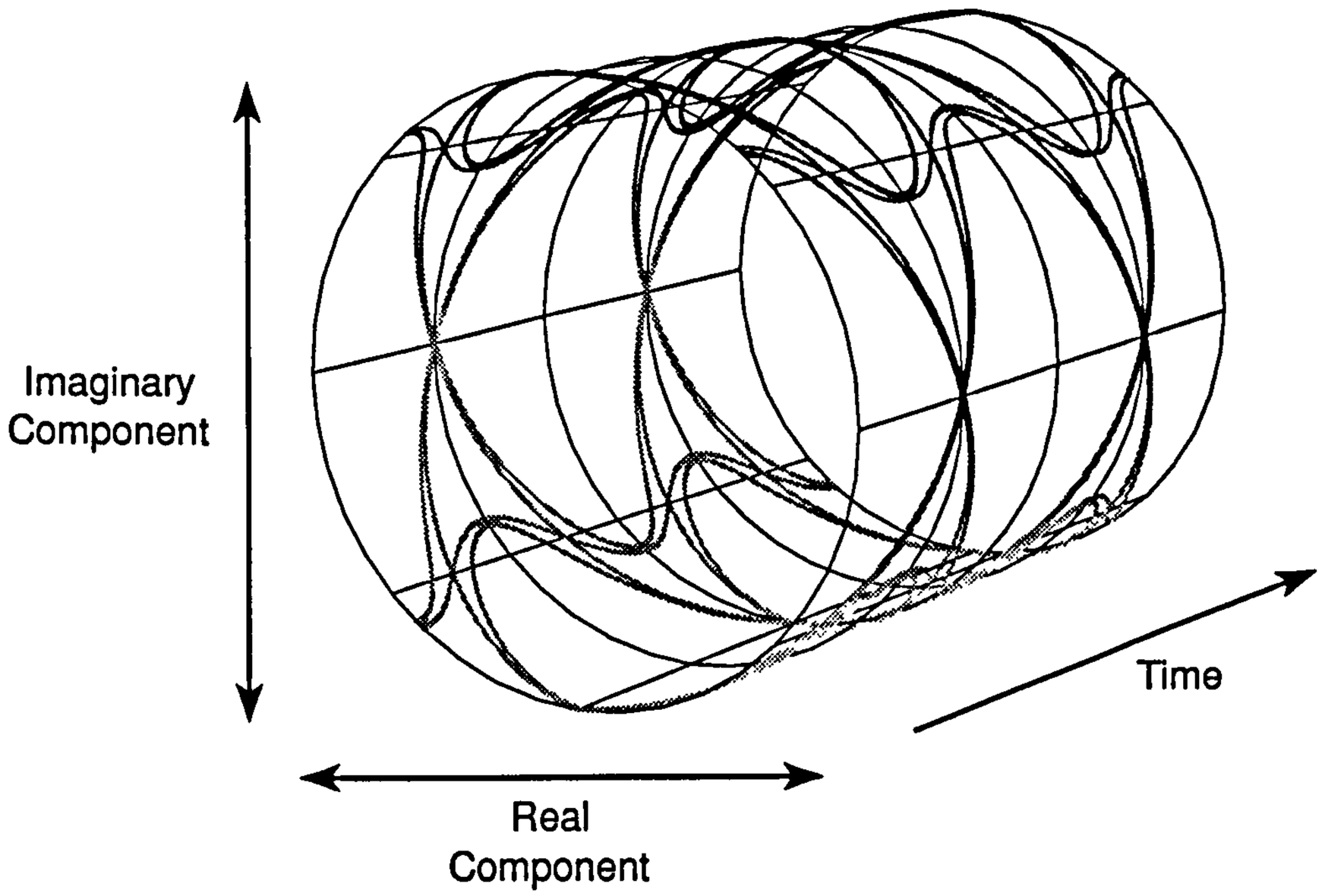
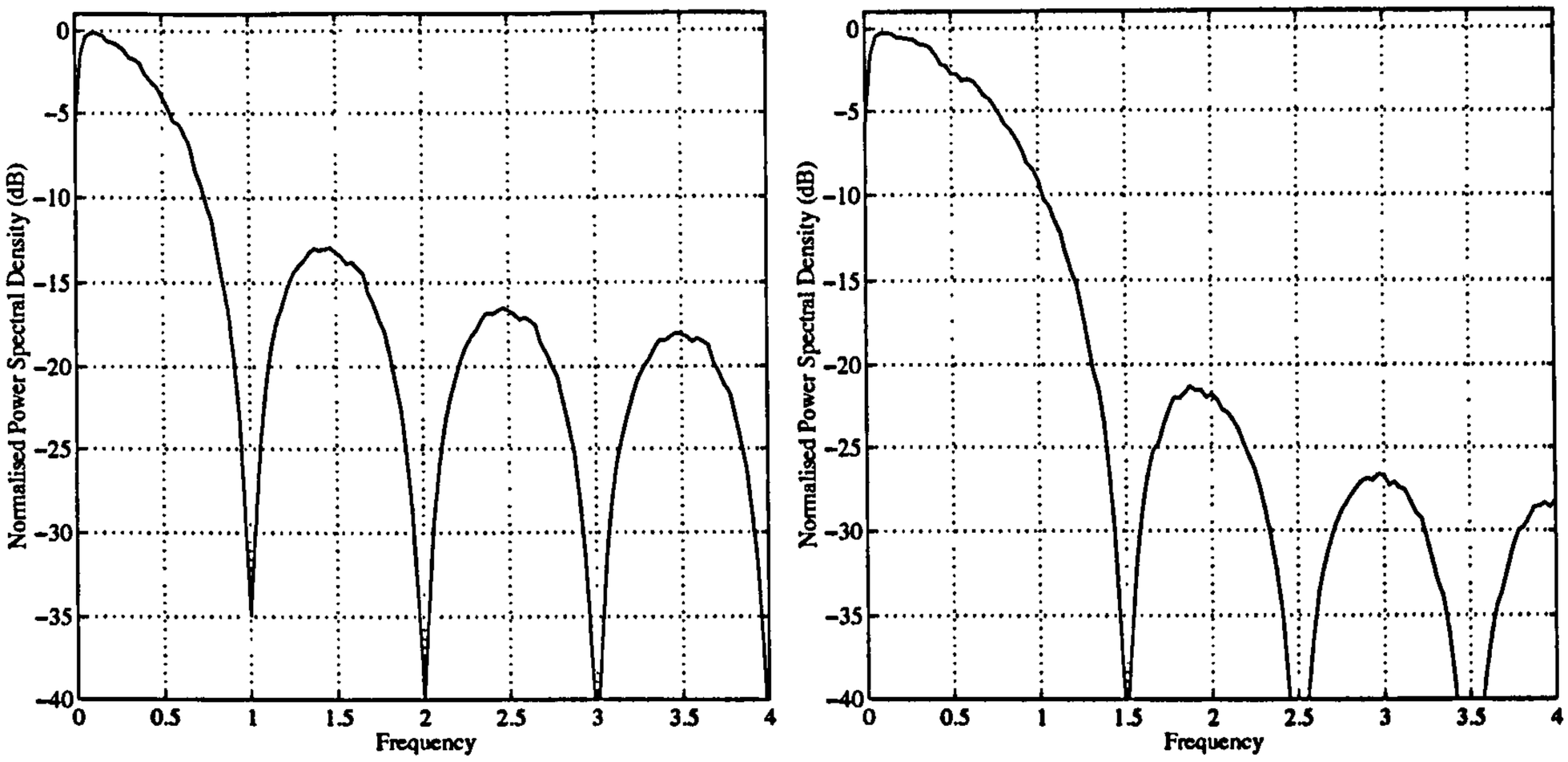



Figure 3.13: Phase cylinder representation of GMSK for $B_tT = 0.25$




(a) QPSK (b) MSK

Figure 3.14: The Power Spectral Density of QPSK vs MSK for the same baud rate

## Power Spectral Density

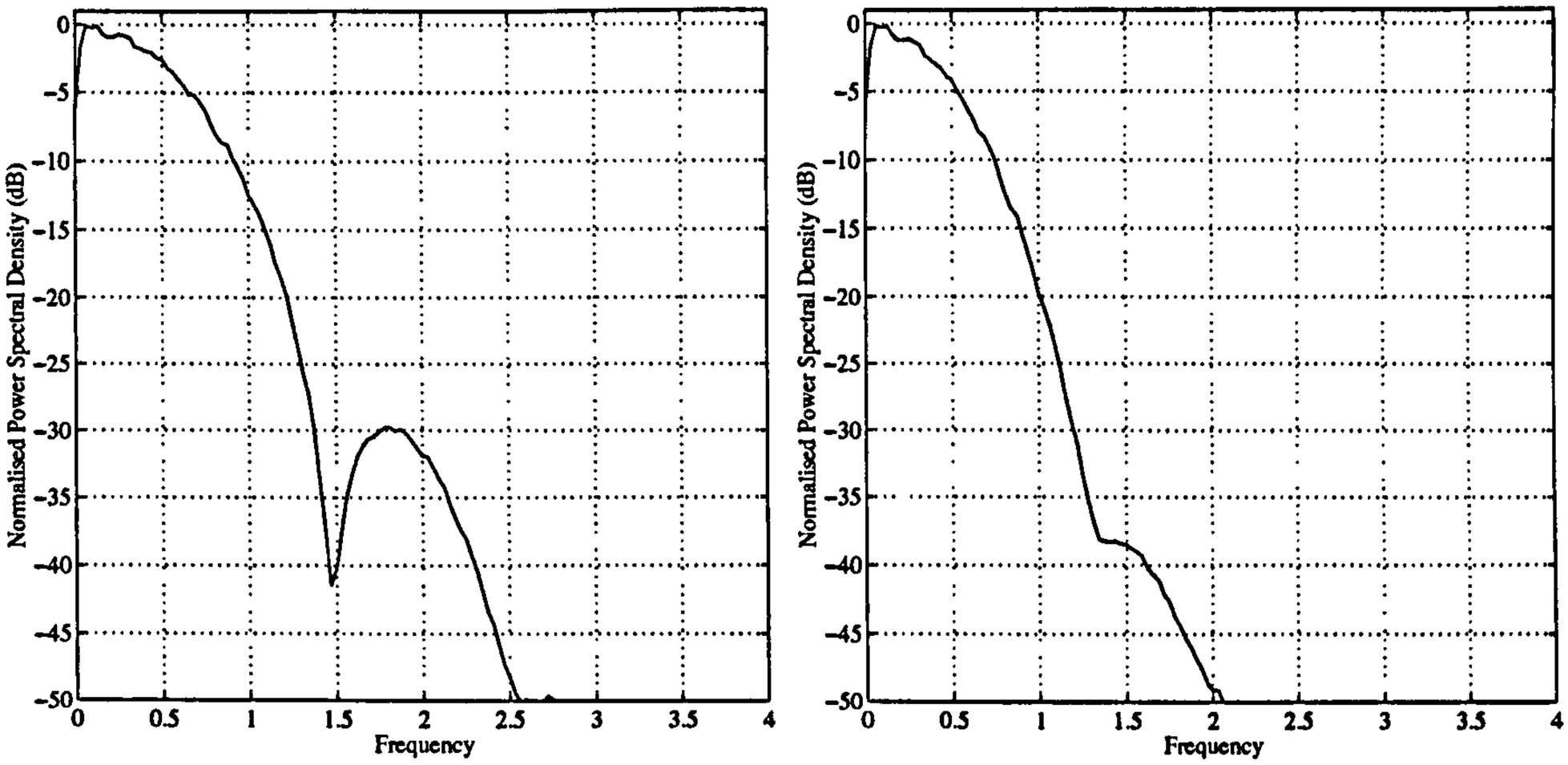
The spectral occupation of the scheme is important because of the need to avoid interference with other users. Figure 3.14 compares the spectra of the MSK signal with simple binary-antipodal (BPSK) signals. One can see from the figure that the MSK has better spectral confinement properties as a result of it having cosinusoidal pulse shaping. The spectral confinement is reduced even further if a Gaussian filter is applied at baseband.

Figure 3.15 shows the Power Spectral Density of the GMSK signalling format for time-bandwidth products $B_tT = 0.25, 0.5$. These figures were generated using the Welch-Periodogram method. This consists of splitting the data sequence into a number of sections of size $M$, where $M$ is a power of two, and thence performing an $M$-point FFT on each of these sections and averaging over the whole of the data sequence. The figure shows that the spectral occupancy of the scheme is markedly reduced when going from MSK to GMSK with $B_tT = 0.5$.

## Minimum Distance of Continuous Phase Modulation Schemes

The minimum distance, $D_{min}$, of a modulation scheme is an important measure as it will determine the bit-error rate performance. In Continuous Phase Modulation, such as GMSK, the minimum distance is dependent on the number of symbol intervals, $N$, that the receiver integrates over. The ideal receiving architecture would be based on one that takes into account the trellis structure of the modulation scheme [41],[42]. The probability of symbol error is dependent on the minimum distance via the Union Bound [41] as,

$$P_e \approx Q\left(D_{min}\sqrt{\frac{E_s}{N_o}}\right) \tag{3.13}$$




(a) $B_tT = 0.25$ (b) $B_tT = 0.5$

Figure 3.15: The Power Spectral Density of GMSK

where $Q(\cdot)$ is the Marcum Q-function.

The minimum distance of CPFSK is shown in figure 3.16 and shows the minimum distance, $D_{min}$, versus the modulation index, $h$, with the number of symbol intervals, $N$, that the receiver integrates over a parameter. Note from the figure that the minimum distance is a highly erratic function with modulation index. This is because the trellis structure of the scheme is mapped on to a phase cylinder and increasing the modulation index need not neccesarily increase the Euclidean distance between paths in the trellis. Note for the case of MSK, $h = 0.5$, an improvement is made in the performance of the scheme if one integrates the received signal over two, rather than one, symbol intervals.

### 3.5.3 CAAD Model of GMSK Modulator

The method used for implementing the GMSK modulator out of SPW kernel blocks is as shown in figure 3.17. The standard SPW block for implementing the GMSK modulator is somewhat more complicated than the implementation shown in figure 3.17 and can thus result in slow simulation runs.

## 3.6 $\pi/4$-QPSK

### 3.6.1 Mathematical Development of the $\pi/4$-QPSK Signalling Format

Let $u(t)$ and $v(t)$ denote the un-filtered baseband binary-antipodal data in the in-phase (I) and quadrature (Q) arms of the modulator. The signal levels of $u_k$ and $v_k$ which are the pulse amplitudes for the I- and Q-channel for $kT \leq t < (k+1)T$ are determined by the signal

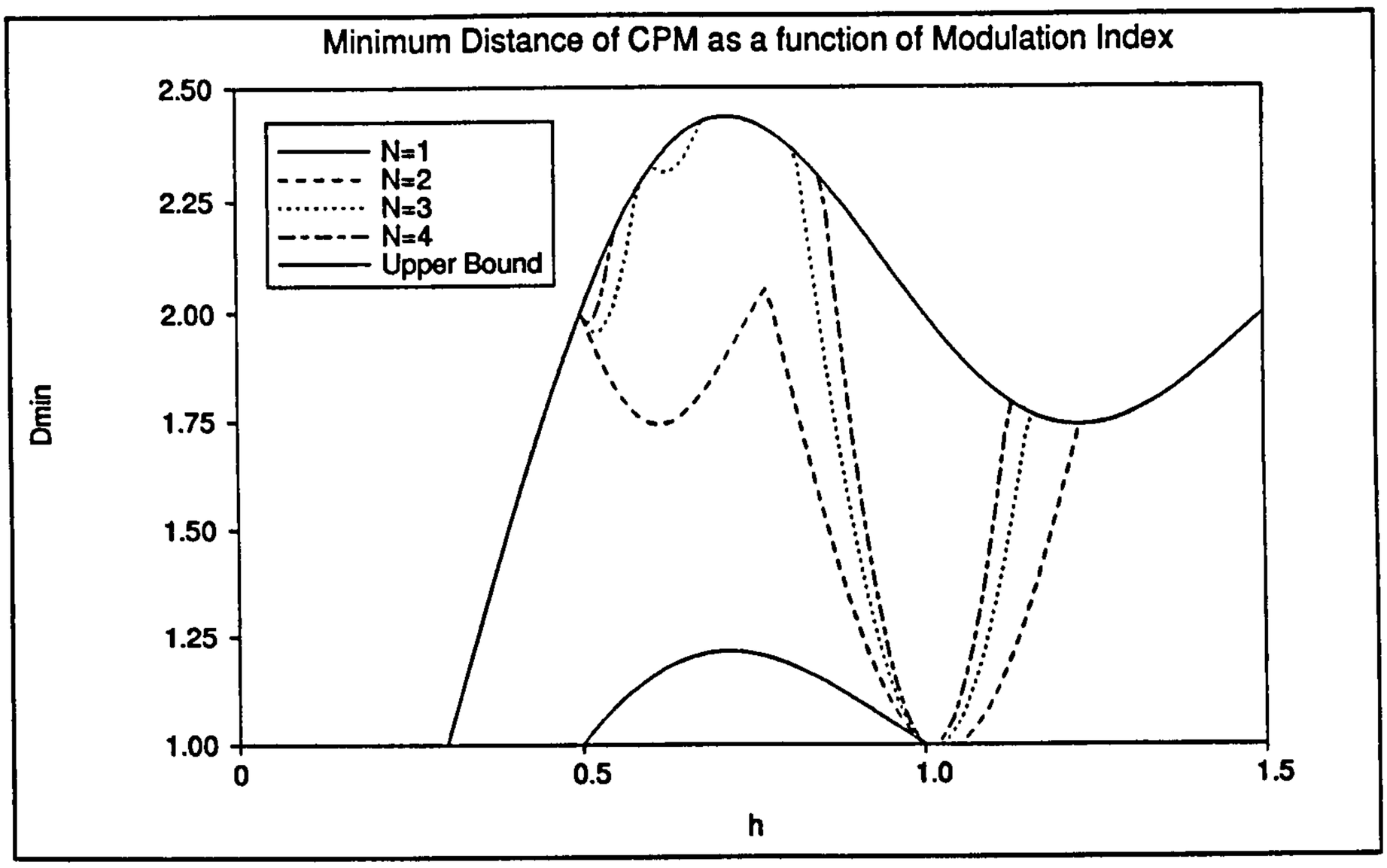



Figure 3.16: Minimum Distance of Full Response CPFSK as a function of modulation index

levels of the previous pulses and the current information symbol denoted by $\theta_k$ as follows:

$$u_k = u_{k-1}\cos\theta_k - v_{k-1}\sin\theta_k \tag{3.14}$$

$$v_k = u_{k-1}\sin\theta_k - v_{k-1}\cos\theta_k \tag{3.15}$$

In the above $\theta_k$ is determined by the current symbol out of the information source. The relationship between $\theta_k$ and the input symbol is given in the following table. Note that $u(t)$ and $v(t)$ can take the amplitudes of $\pm 1, 0$ and $\pm 1/\sqrt{2}$.

| Information | $\theta$ | $\cos\theta$ | $\sin\theta$ |
|---|---|---|---|
| 11 | $\pi/4$ | + | + |
| 01 | $3\pi/4$ | − | + |
| 00 | $-3\pi/4$ | − | − |
| 10 | $-\pi/4$ | + | − |

One can imagine that the input data is mapped onto a QPSK constellation which is rotated by $\pi/4$ every symbol interval.

### 3.6.2 Characteristics of $\pi/4$-QPSK

Figure 3.18 shows the scatter and eye diagrams for the $\pi/4$-QPSK modulation format. If one were to consider only the real and imaginary components of the signal at the sampling instants then one can see that there are a total of eight phase states. However, if one examines the eye-

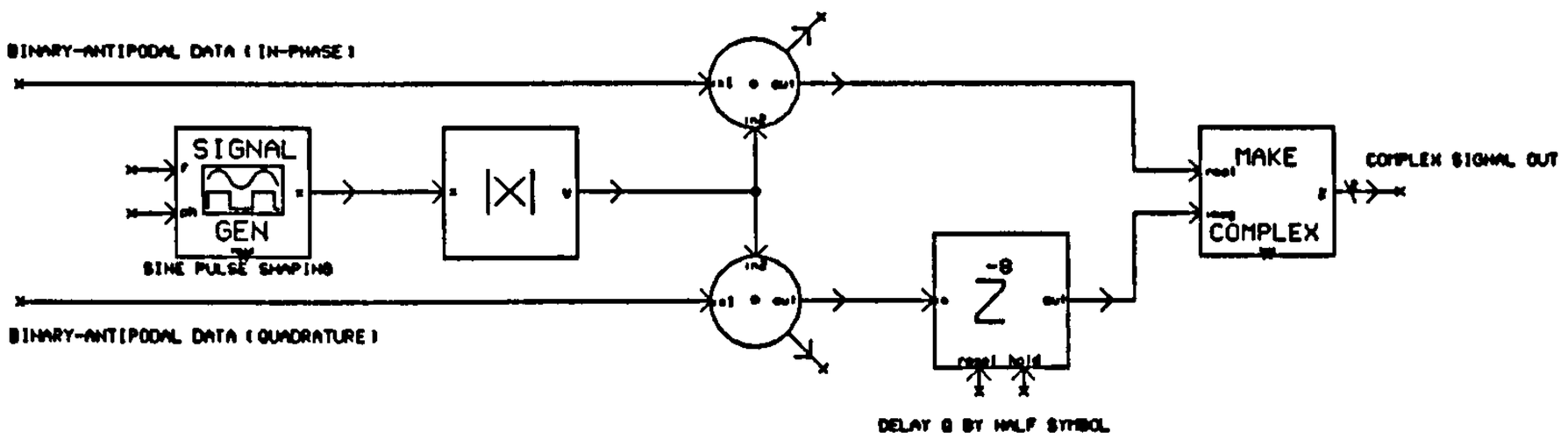
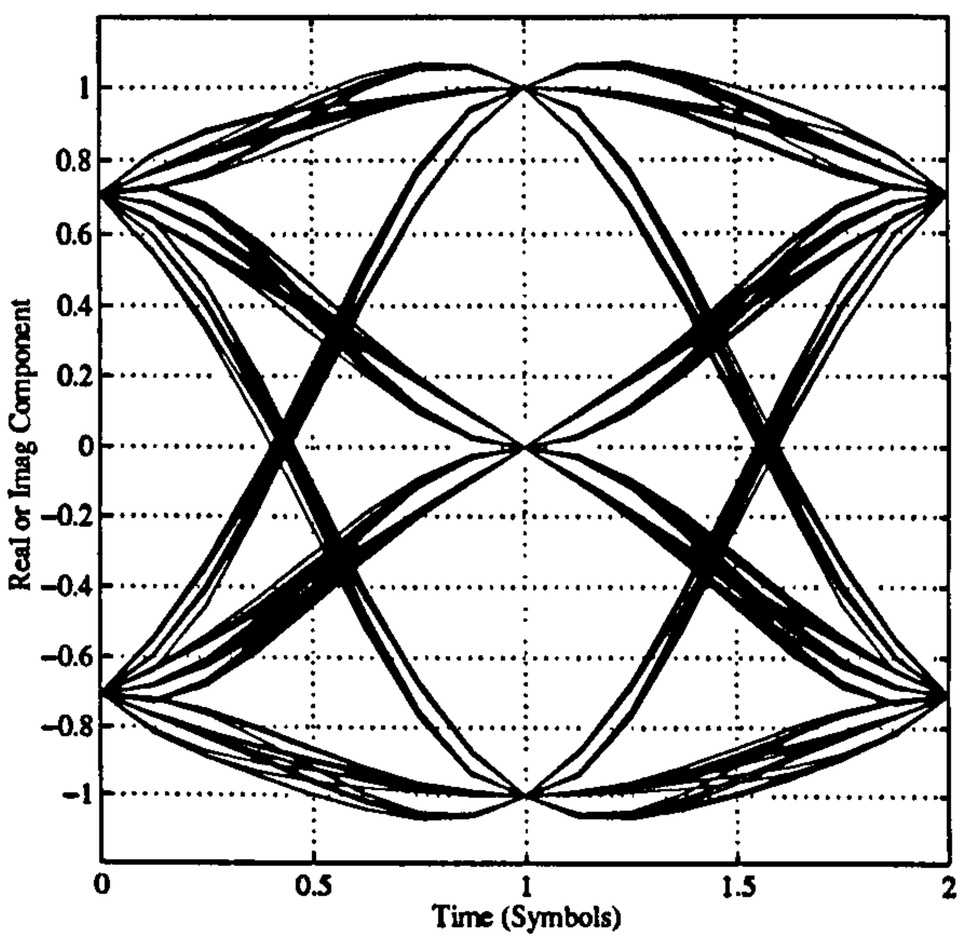
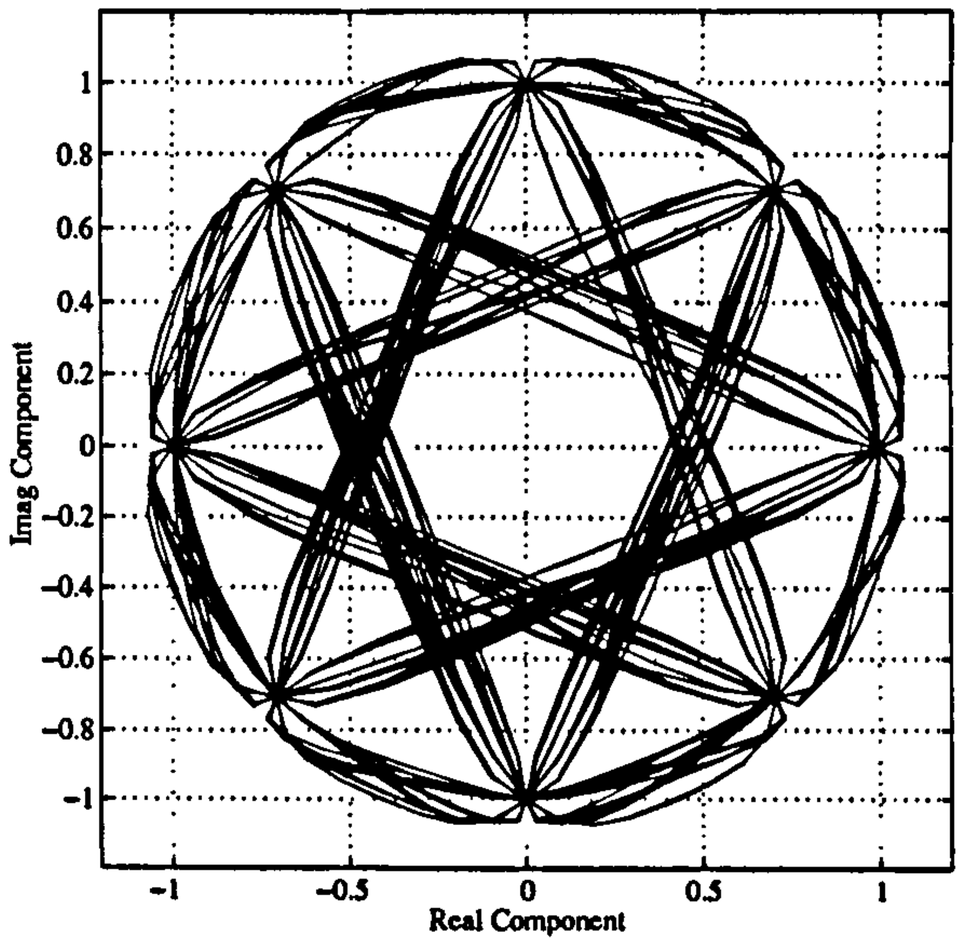



Figure 3.17: Serial & Parallel Implementations of MSK/GMSK Modulators




(a) $\pi/4$-QPSK, Eye Diagram




(b) $\pi/4$-QPSK, Phase Trajectory

Figure 3.18: Eye and scatter diagrams for raised cosine filtered ($\alpha = 0.8$) $\pi/4$-QPSK.

diagram then it becomes apparent that the scheme consists of two four point constellations that are rotated at each symbol interval.

# Chapter 4

# Channel Modelling

## 4.1 Introduction

We want to determine the performance of the schemes developed thus far in the narrowband, frequency non-selective fading channel. We therefore have to develop an appropriate channel model for inclusion in our system simulator. As a result of novel research in this area [43], we shall show that an appropriate design method for the correlation shaping filter used in the simulation of the narrowband fading channel pertinent to urban land-mobile radio communications is to use an FIR filter with a windowed-Bessel impulse response.

Often the desired Doppler bandwidth of the correlation shaping filter is very small relative to the system sampling rate. Important to the implementation of a fading system is the Doppler frequency normalised to the baud rate, or $f_dT$. This product can, under certain circumstances, be very small. As an example, consider a vehicle moving at a speed of $v =$ 50km/Hr, a carrier frequency of $\nu = 1$GHz, and a channel baud rate of $R_b = 200$ksymbols/s. These parameters are typical of the GSM system. Then the maximum Doppler is,

$$f_d = v\frac{\nu}{c} \approx 46 \text{ Hz} \tag{4.1}$$

and, $f_dT = 2.3 \times 10^{-4}$. This is not an uncommon scenario.

The important point is that when we simulate such a system we may normalise the baud rate to one and use four times oversampling and then we need to have a Doppler shaping filter with a cutoff very much smaller than the system sample rate. In order to simulate such a highly correlated process we would require an inordinate number of taps if the correlation were produced with an FIR filter.

An alternative is to approximate the required transfer function with an IIR filter. Design methods do exist that can be used to design arbitrarily shaped IIR filters, such as the Yule-Walker design method, but these tend to be unstable for filter cutoff frequencies very much less than the sampling rate of the system.

A better method is to use a multirate filter based on a polyphase filter. This filter is similar to the multirate methods that were employed to derive a fractional delay element for asynchronous timing-recovery loops. We shall discuss this method next.

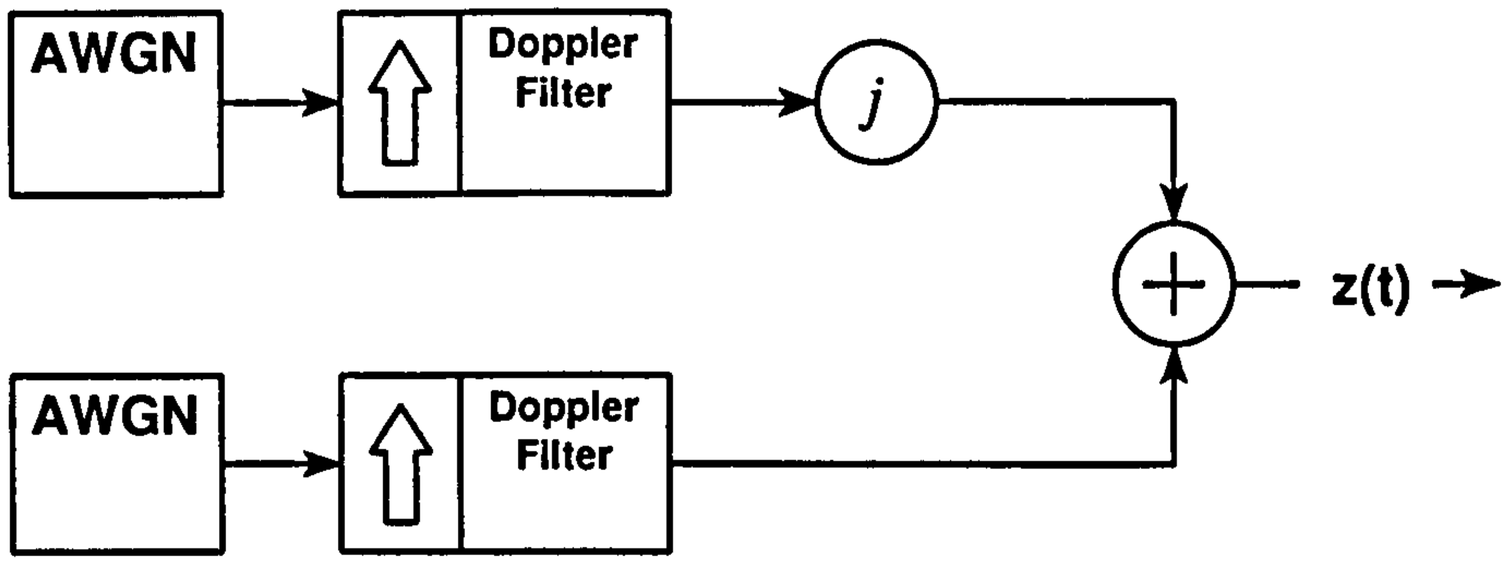



Figure 4.1: Complex baseband model to generate fading process.

## 4.2 Implementation in a Multirate Digital Filter

The narrowband channel is modelled as a time-varying multiplicative complex noise source. The wideband channel can be considered to be a transversal filter each tap of which is produced by a time-varying noise source. So the narrowband frequency non-selective fading channel can be considered a special case of the wideband channel. That is to say, as the delay spread of the wideband channel drops to zero then we get the narrowband channel.

The narrowband Land-Mobile Radio channel has a power spectral density given by the following equation where the mobile receiver has no direct line-of-sight to the transmitter [44],

$$W(f) = \begin{cases} \dfrac{\sigma_s^2}{\pi\sqrt{f_d^2 - f^2}} & \text{if } |f| \leq f_d \\ 0 & \text{if } |f| > f_d \end{cases} \tag{4.2}$$

In the simulation of the narrowband LMR channel one would use an equivalent complex baseband representation as shown in figure 4.1. The real and imaginary components of the fading process will be independent but identically distributed random processes generated by filtering white Gaussian noise with identical filters. The response of the individual filters is given as the square-root of equation (4.2).

It is worth deriving the closed-form expression for the ideal impulse-response of these filters as this will then lead to an appropriate design procedure. The filter impulse response is found to be :

$$x(t) = \int_{-f_d}^{f_d} \cos(2\pi f t)\sqrt{W(f)}\, dt = K\, \frac{J_{1/4}(2\pi f_d t)}{\sqrt[4]{t}} \tag{4.3}$$

where $K$ is a time-independent constant.

It is also worth noting for the filter design that

$$\lim_{t\to 0} \frac{J_{1/4}(2\pi f_d t)}{\sqrt[4]{t}} = \frac{\sqrt[4]{f_d \pi}}{\Gamma(5/4)} \approx 1.468813\, \sqrt[4]{f_d} \tag{4.4}$$

Just as an approximation to the ideal lowpass filter is given as a windowed-sinc an approximation to the ideal filter for the simulation of the LMR channel is given as a windowed-Bessel

function. Good routines that allow the computation of fractional-order Bessel functions are available [45]. A MATLAB routine for generating the FIR filter coefficients using fractional-order bessel functions is included in the appendix B.

Results showing the simulated performance of the correlation shaping filter are shown in figure 4.2 for a Hamming window and in 4.3 for a Boxcar (rectangular) window. In the results $f_dT = 1$ and $f_s = 4$. It can be seen from the figures that a large number of taps are needed in the design to achieve close agreement between the theoretical and simulated power spectral densities. Incidentally, these power spectral density plots were generated using the Welch periodogram with a window size of $N = 256$.

To achieve lower Doppler frequencies we can cascade a noise shaping filter with a polyphase interpolation filter as shown in figure 4.4. We could alternatively combine the functions of interpolation and noise shaping in a manner described next.

### 4.2.1 Design Method

A highly-correlated fading process is generated by configuring the cutoff frequency on the Doppler filters to be small. This can present problems for IIR filters as they can become unstable and direct FIR filtering would result in long, computationally-expensive convolutions that have to be performed on every iteration. Instead we use multirate filtering techniques [46]. An $M$ tap filter is arranged by up-sampling the noise sources by a factor $U$ and performing a $K$-tap convolution on every iteration where $M = U \times K$. The functions of correlation shaping and image rejection are accomplished jointly provided $f_d < f_s/(2\,U)$, where $f_s$ is the sampling frequency.

The design procedure for implementing the filters in software is as follows –

1. *Filter Initialisation :*

   Define indices $u$ and $k$ such that

   $$0 \leq u < U \text{ and } 0 \leq k < K \tag{4.5}$$

   and the time index

   $$t = \frac{1}{f_s}\left(k\,U + u - \frac{1}{2}(M-1)\right) \tag{4.6}$$

   then the $M$ tap filter $H_m$ has coefficients given by :

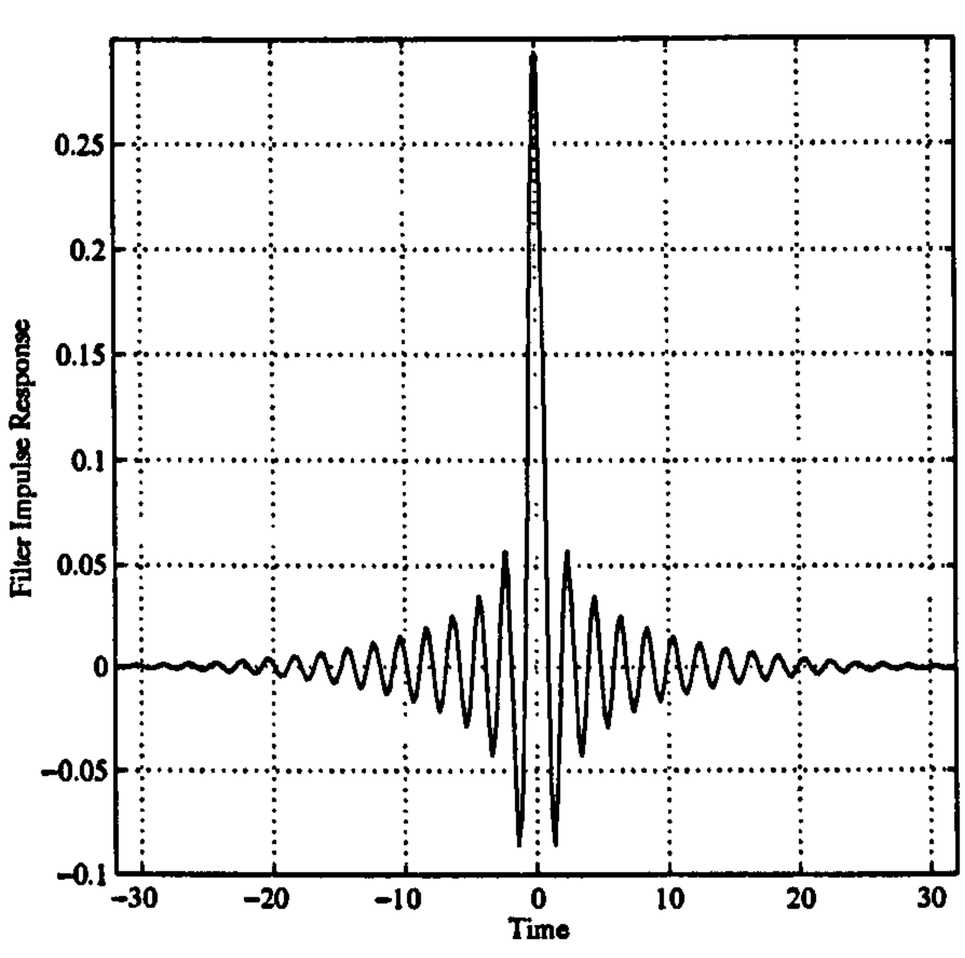
   $$H_{k+u\,U} = \begin{cases} W_{k\,U+u}\, J_{1/4}(2\pi f_d|t|)/\sqrt[4]{|t|} & \text{if } |t| > 0 \\ \\ W_{k\,U+u}\, 1.468813\, \sqrt[4]{f_d} & \text{if } t = 0 \end{cases} \tag{4.7}$$

   where $W_m$ are the window weighting coefficients.

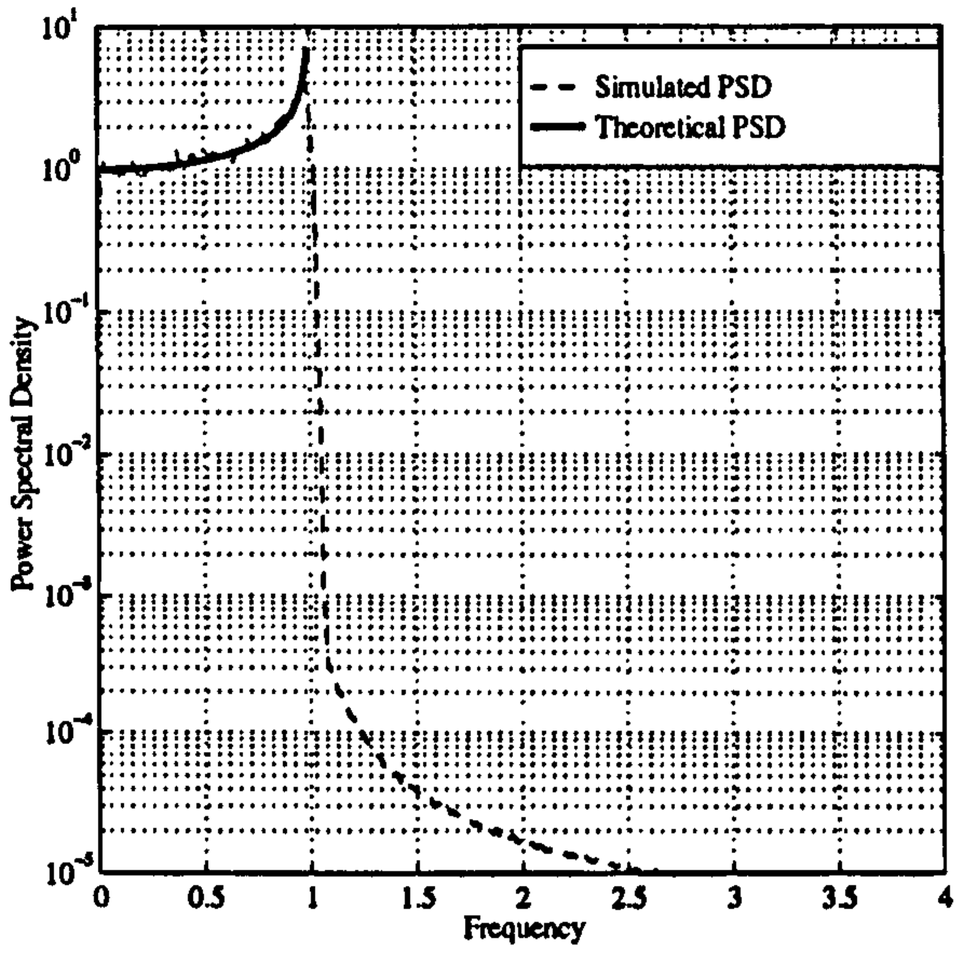
   The coefficients are normalised to render the correlation functions independent of up-sample by multiplying the filter taps by

   $$a = \sqrt{\frac{U}{2\sum_{m=0}^{M-1} H_m^2}} \tag{4.8}$$

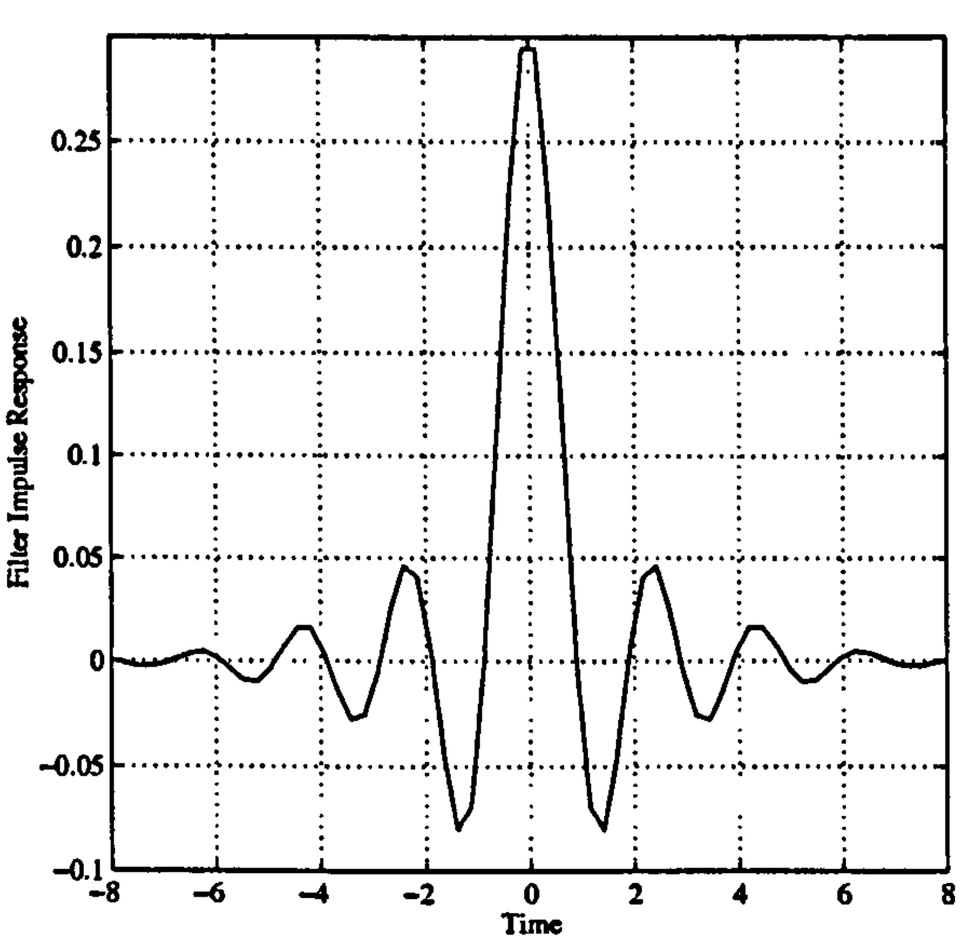
   Set filter index $i_u = U$.



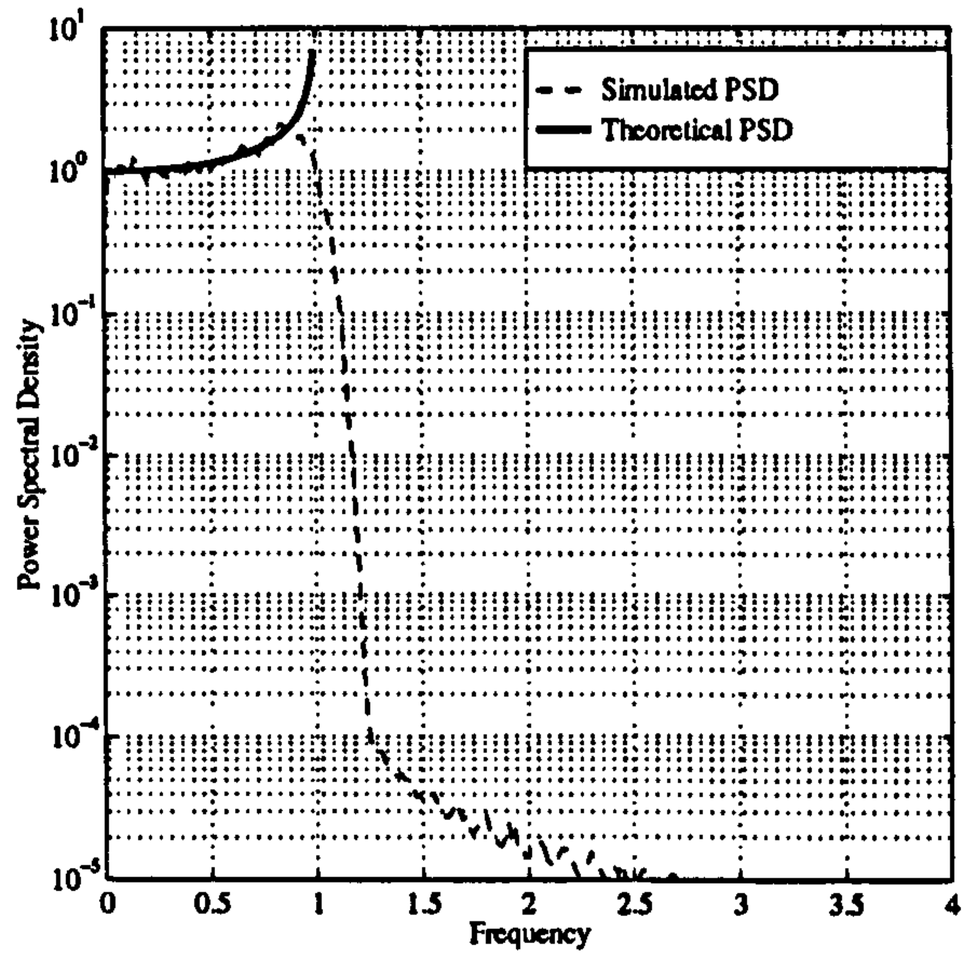

(a) Filter Impulse Response $N = 256$




(b) Measured Noise PSD

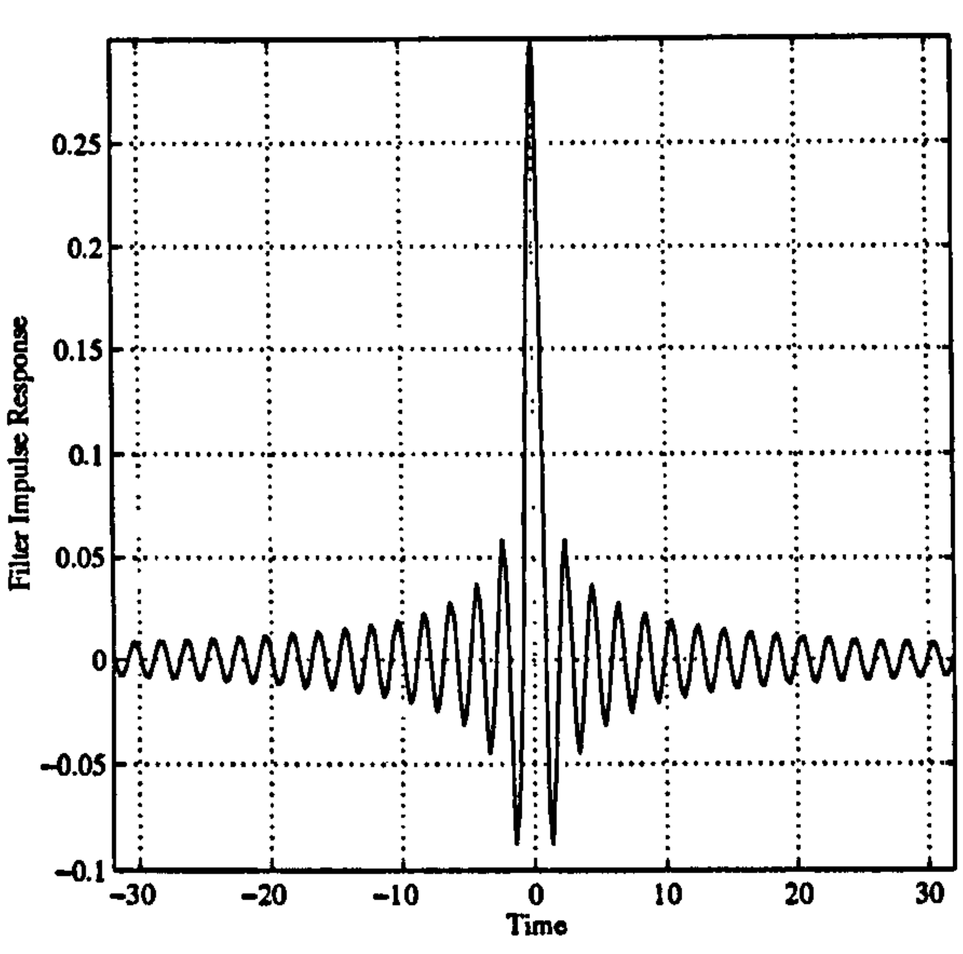



(c) Filter Impulse Response $N = 64$

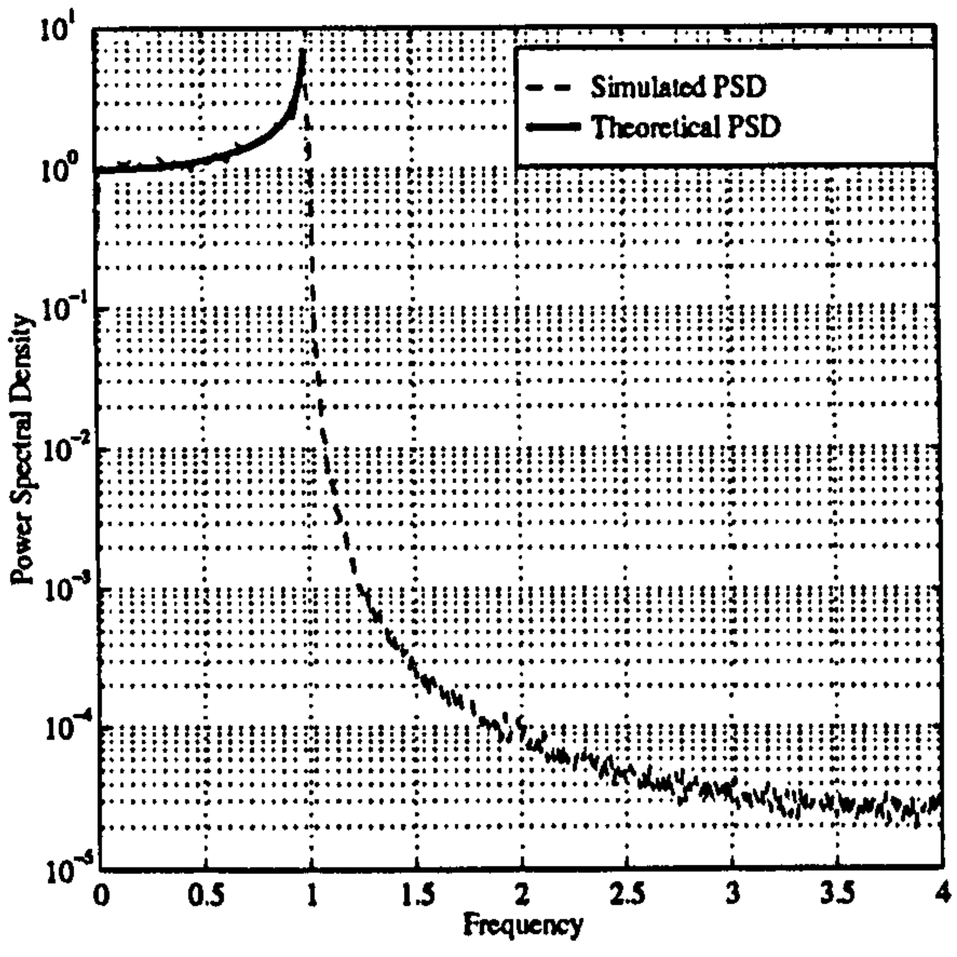



(d) Measured Noise PSD

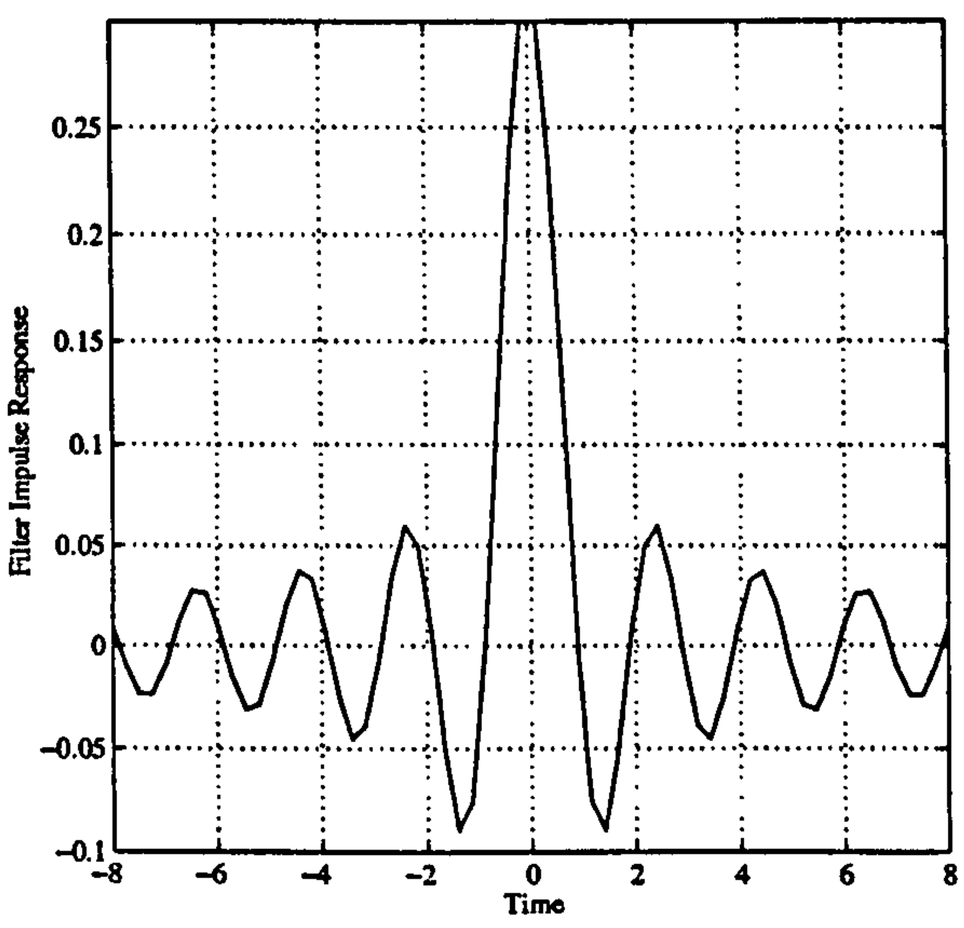
Figure 4.2: Filter design and results for the narrowband fading channel model. A Hamming window was used for the design method.



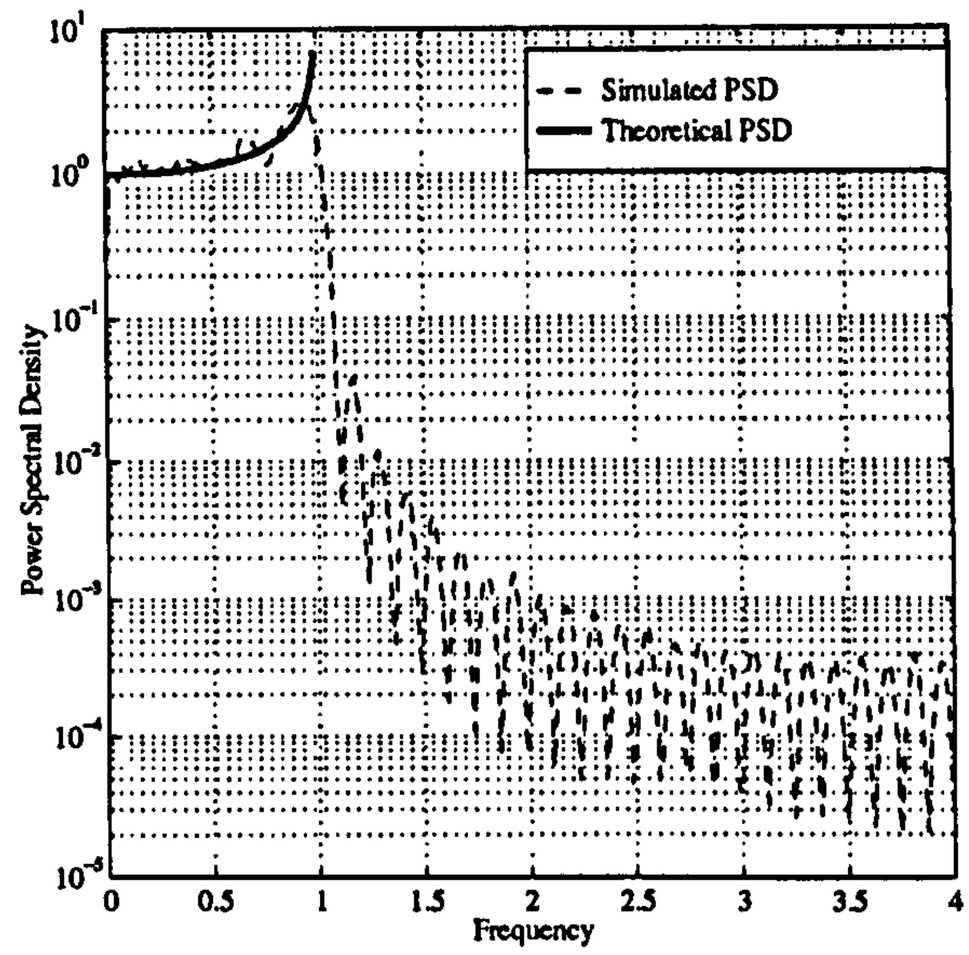

(a) Filter Impulse Response $N = 256$




(b) Measured Noise PSD




(c) Filter Impulse Response $N = 64$




(d) Measured Noise PSD

Figure 4.3: Filter design and results for the narrowband fading channel model. A Boxcar window was used for the design method.

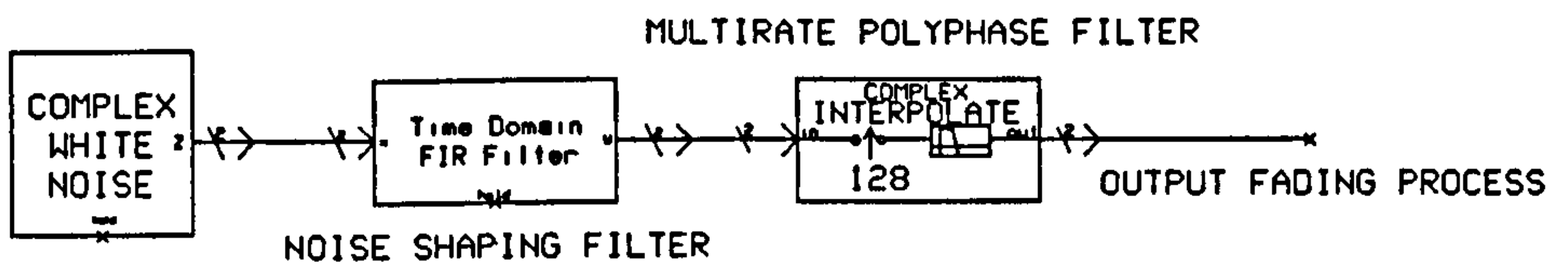



Figure 4.4: Method of generating a narrowband fading process with precise autocorrelation function

2. *Run-Time Procedure :*

   If $i_u = U$ then read in noise sample to $K$-tap state register $x_k$ and set $i_u = 0$.

   Filter output $y$ is formed by performing $K$-point convolution

$$y = \sum_{k=0}^{K-1} x_k H_{k+i_u K} \tag{4.9}$$

   Increment index $i_u$ by 1 and return to start of run-time procedure.

An example of the effect of multiplying a narrowband fading signal with an MSK modulated signal is shown in figures 4.5 & 4.6. Both differential and frequency-discriminator detection suffer from an irreducible error-rate at high signal to noise. This effect is illustrated by considering the diagrams shown in figure 4.5 and figure 4.6 which show the effect that the Rayleigh fading has on differentially detected data for fading rates of $f_d T = 0.001$ and $f_d T = 0.005$.

Some salient features about the nature of errors introduced by the flat fading channel can be gleaned from figure 4.5 and figure 4.6. The *average* signal to noise ratio is high and yet errors in the data transmission still occur. This is due to the effect that the fading channel has on the *instantaneous* value of the signal to noise (the top trace in the figures). It is evident that errors occur when the instantaneous value of the signal to noise is low because of destructive multipath interference. Note that the errors then appear to come in bursts. This is a marked difference to data transmission over, for example, the satellite channel where any errors will be uniformly distributed in time. This bursty nature of the mobile channel has implications with regard to appropriate choice of coding/interleaving. The two figures compare the error performance for two different fading rates. Note that in the faster fading scenario the error bursts occur more frequently but are very short. The slower fading case will have a lower average error rate but when error bursts do occur they will be longer in duration. This has an implication with regard to appropriate coding/interleaving design and might even allow for the use of adaptive techniques.

It is evident from the figure how the use of multirate signal processing allows one to generate processes that are changing at very different rates.

An illustration of the output spectrum from the correlated noise source is shown in figure 4.7 for an upsample factor of 10. The plots illustrate the image rejection of the multirate polyphase filter for different values of filter multiplier $K$. This is the size of the convolution

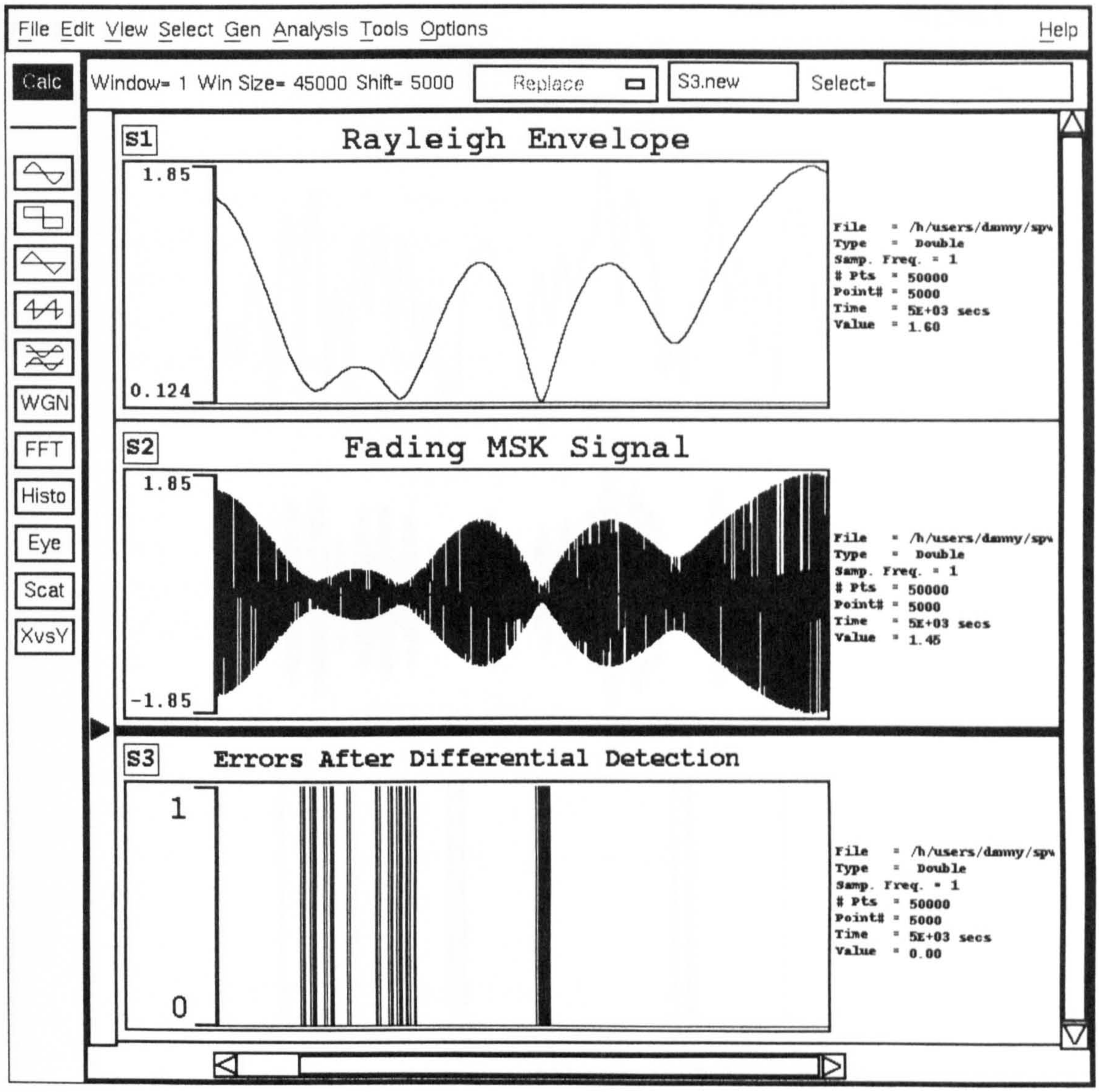



Figure 4.5: Plot showing qualitatively the effect that flat Rayleigh fading has on the error rate of Differentially Detected MSK. The fading rate (normalised to the bit rate) is $f_d T = 0.001$.

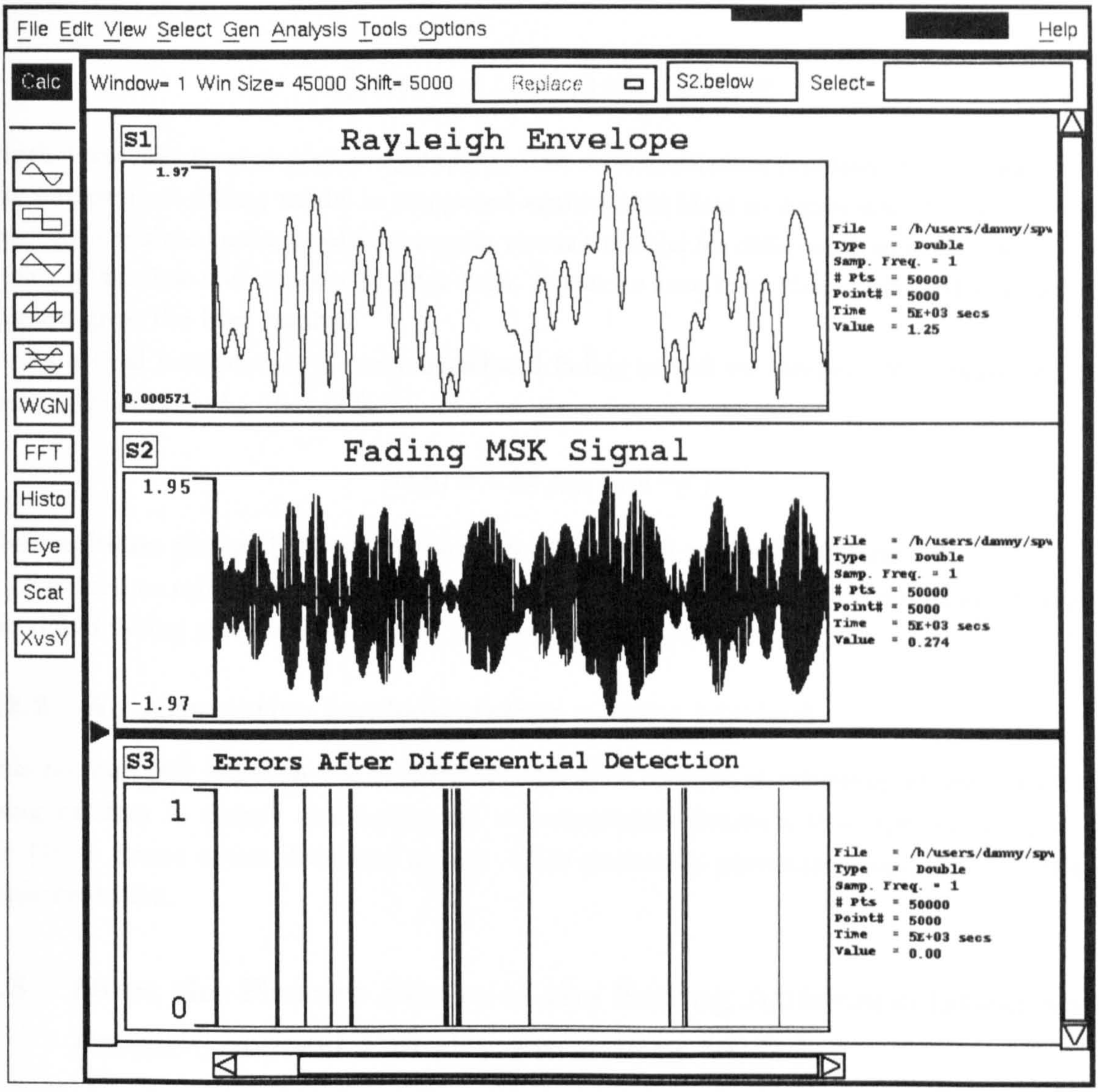



Figure 4.6: Plot showing qualitatively the effect that flat Rayleigh fading has on the error rate of Differentially Detected MSK. The fading rate (normalised to the bit rate) is $f_d T = 0.005$.

performed on every iteration and so it should be kept reasonably small. Having this value too small, however, results in inadequate interpolation and interference results between the wanted component at $f = 0$Hz and its images. It has been found that moderate values of $K$, in the region of 15–25, will usually suffice.

It is the autocorrelation function of the fading process that impacts strongly on the system symbol error rate as we shall see later. The theoretical autocorrelation function of the fading process is given as,

$$\sigma_s^2 \rho_s(\tau) = \int_{-\infty}^{+\infty} W(f) \exp(j2\pi f\tau)\, df \tag{4.10}$$

which turns out to give $\rho(\tau) = \mathrm{J}_0(2\pi f_d \tau)$. The autocorrelation function of the noise out of our narrowband fading model is compared against this ideal in figure 4.8. The plots again show the simulated autocorrelation results versus the ideal for different design choices, namely choice of window and number of filter taps. It can be seen from these results that a Boxcar window gives the best results.

As a final justification of our narrowband fading model we compare the simulated level crossing rate with the ideal [47]

$$N(R) = \sqrt{2\pi}\, f_d\, \rho \exp(-\rho^2) \tag{4.11}$$

The results are plotted in figure 4.9 on both a linear and a logarithmic scale.

For further references on the implementation of multirate filter for producing a highly correlated fading process see [46], [48] and our paper in [43].

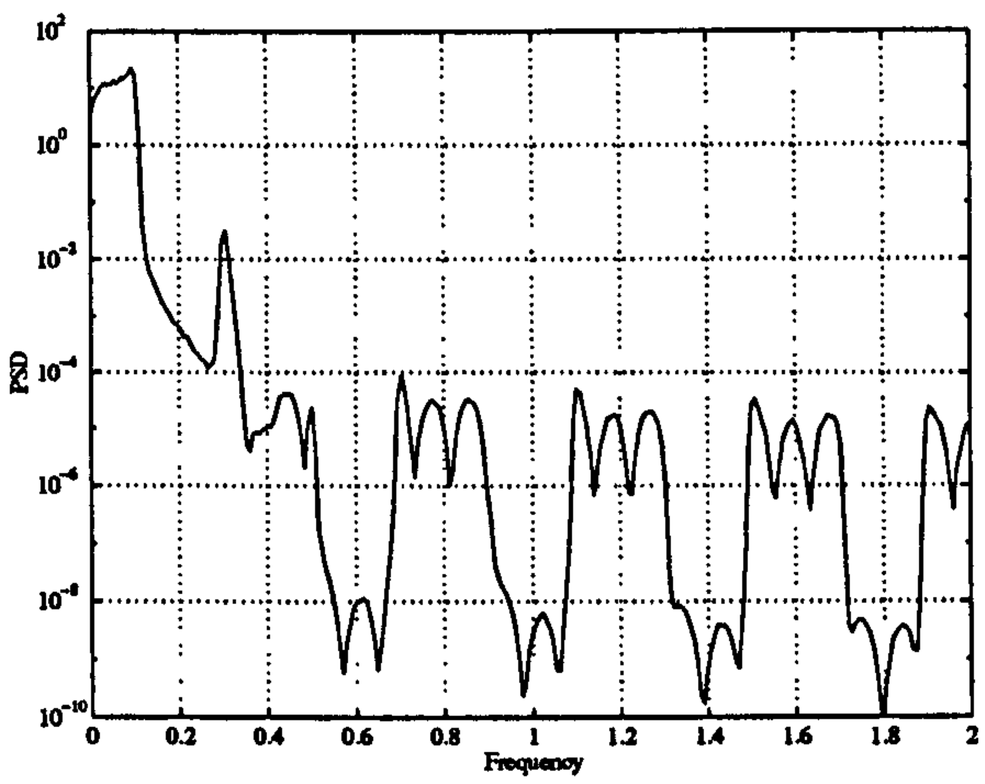
### 4.2.2 An Alternative Implementation – Jakes Method

This method [49] relies on the asymptotic sum of a number of randomly-phased sinusoids being random in nature and having an autocorrelation function that approaches that of the Gan's Power spectral density model. This method is particularly suited to hardware implementation.

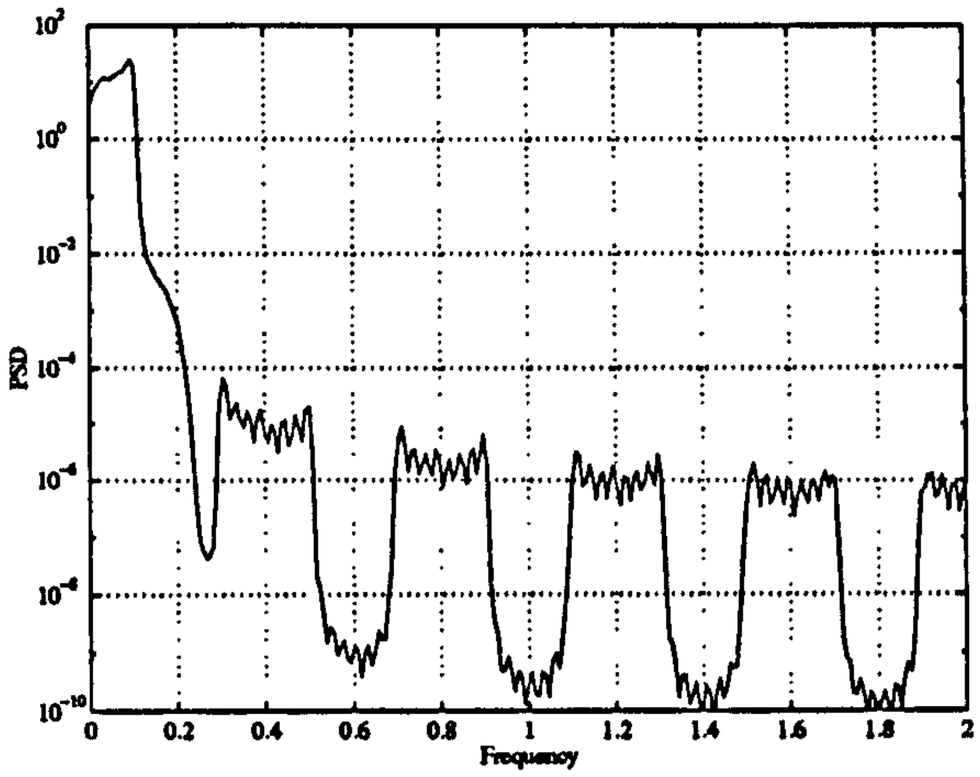
## 4.3 Does the Precise Shape of the Fading Autocorrelation and Power Spectral Density Functions Matter?

Much effort can be expended in the simulation of the flat fading channel in developing a channel model with the *precise* characteristics determined by the Gan's Power Spectral Density model given above. An obvious question is how accurate does it have to be, especially when one is up-sampling the fading signal, sometimes by factors as large as 100 or more? We are essentially dealing with a very narrowband random process, spectrally confined around DC complex frequency, and so one might ask is it necessary to attempt to regenerate the fine detail as required by the theory.

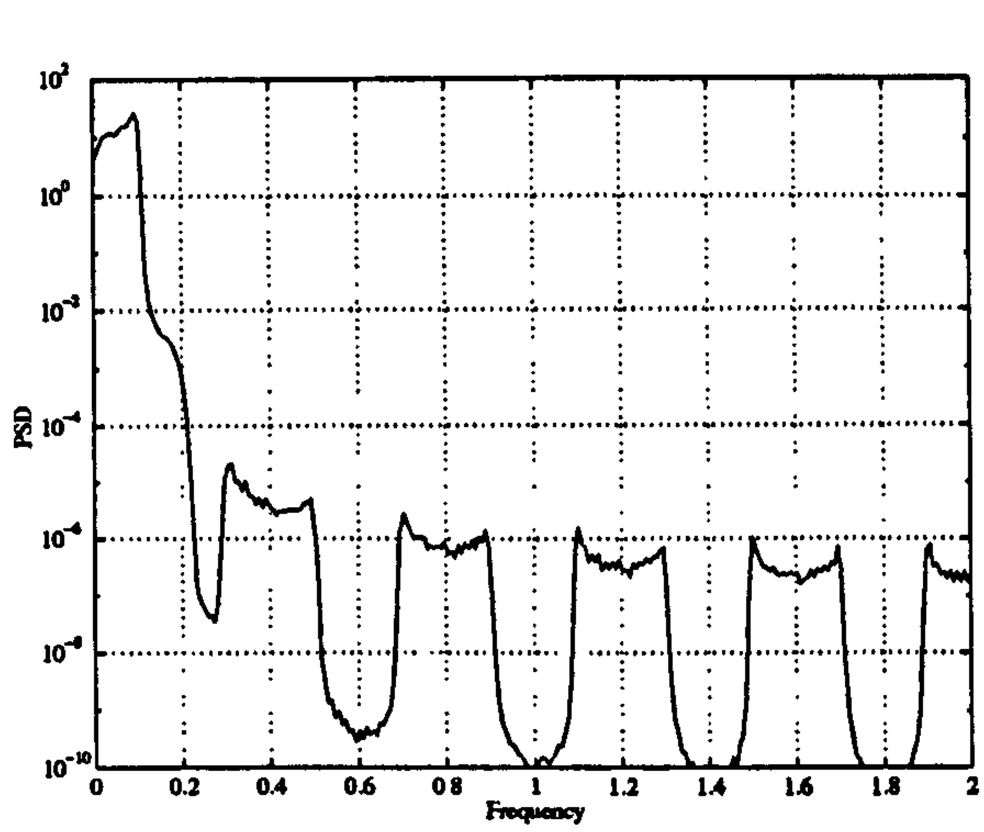
It has been pointed out [50] that what impacts the most on system error performance in the flat fading channel is the autocorrelation function. This has to be accurate relative to the




(a) $K = 5$

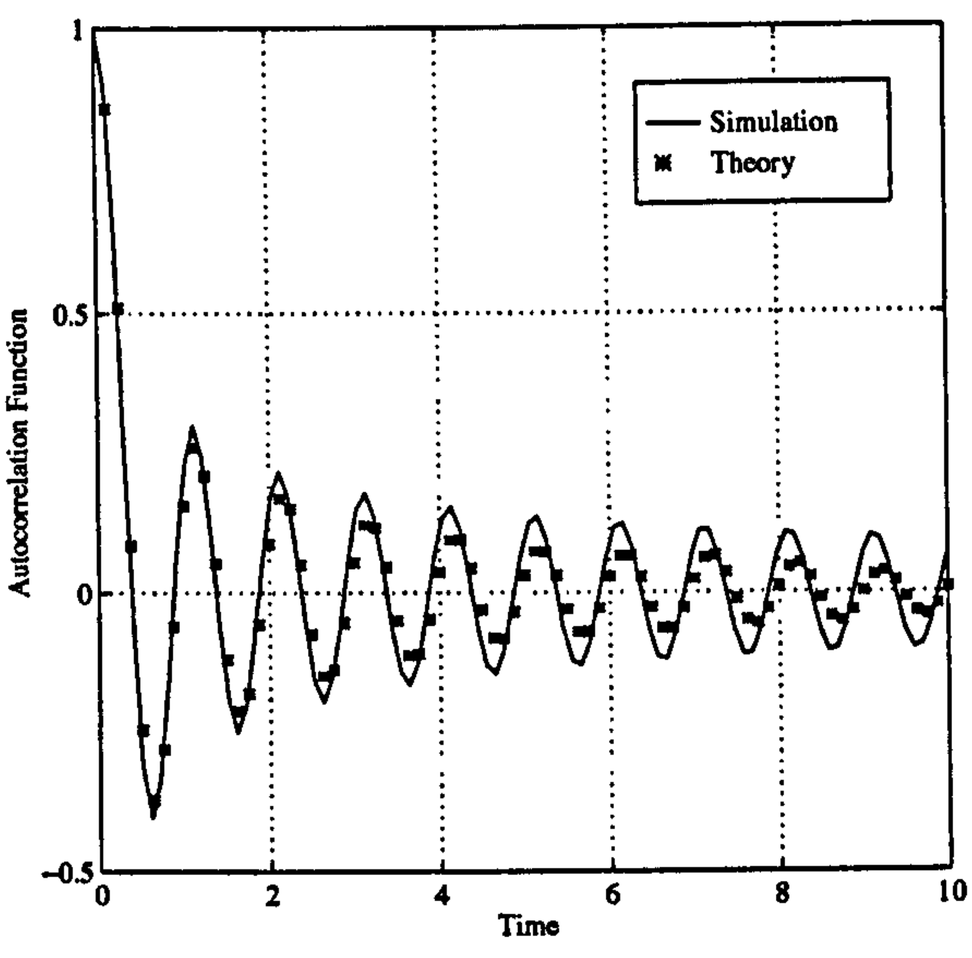



(b) $K = 15$

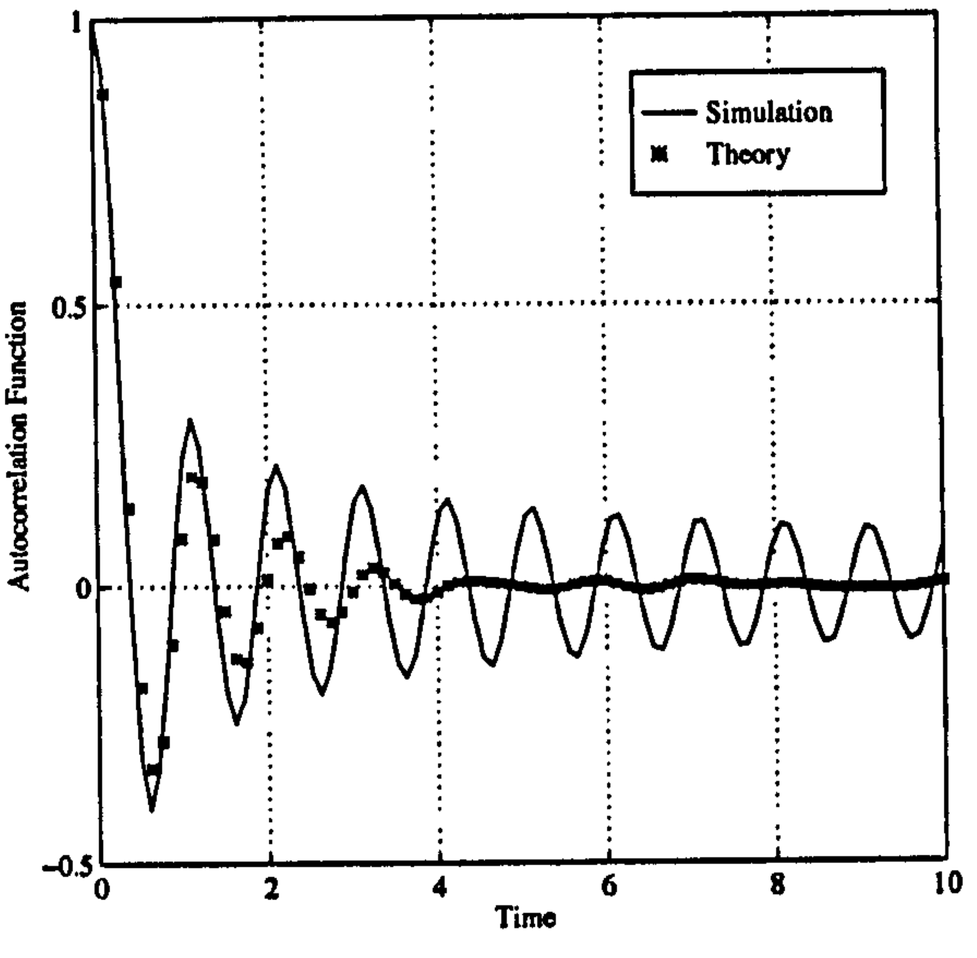



(c) $K = 25$

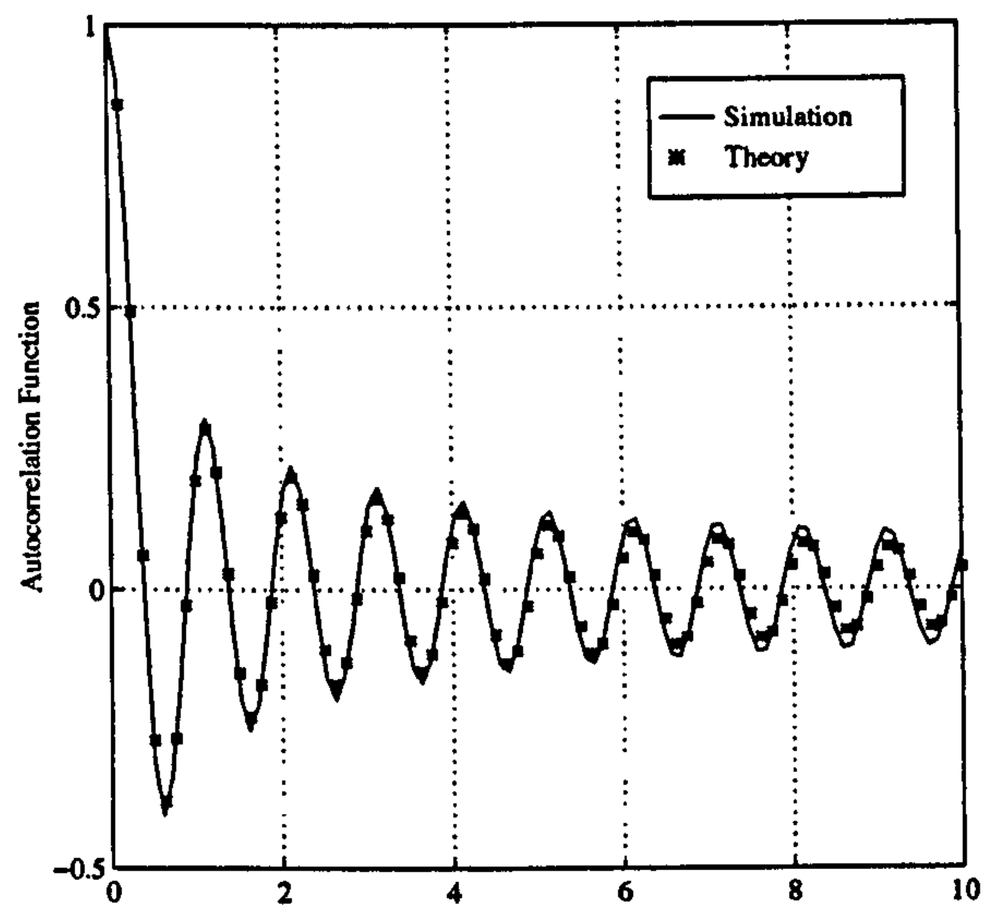
Figure 4.7: Resulting Power Spectral Density of Narrowband Fading Process After Interpolation. Figure illustrates the effect of the multiplier length, $K$, in the design of the interpolation filter



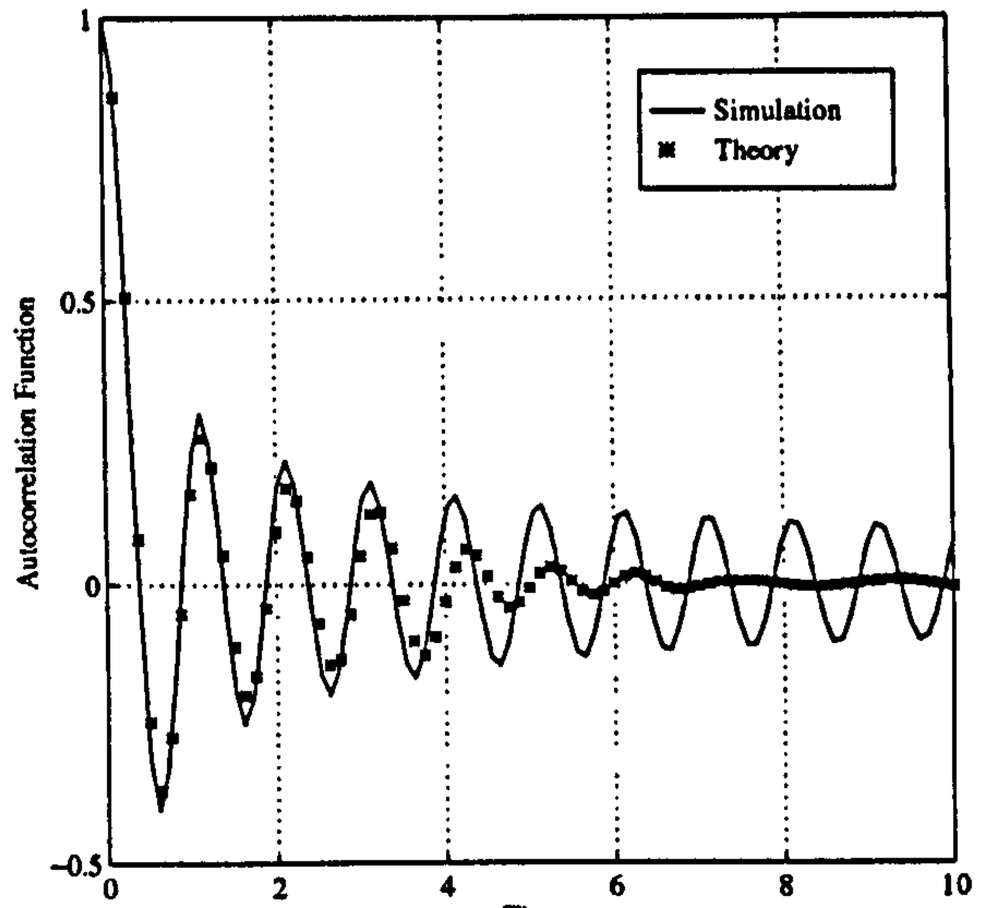

(a) Hamming Window & $N = 256$




(b) Hamming Window & $N = 64$

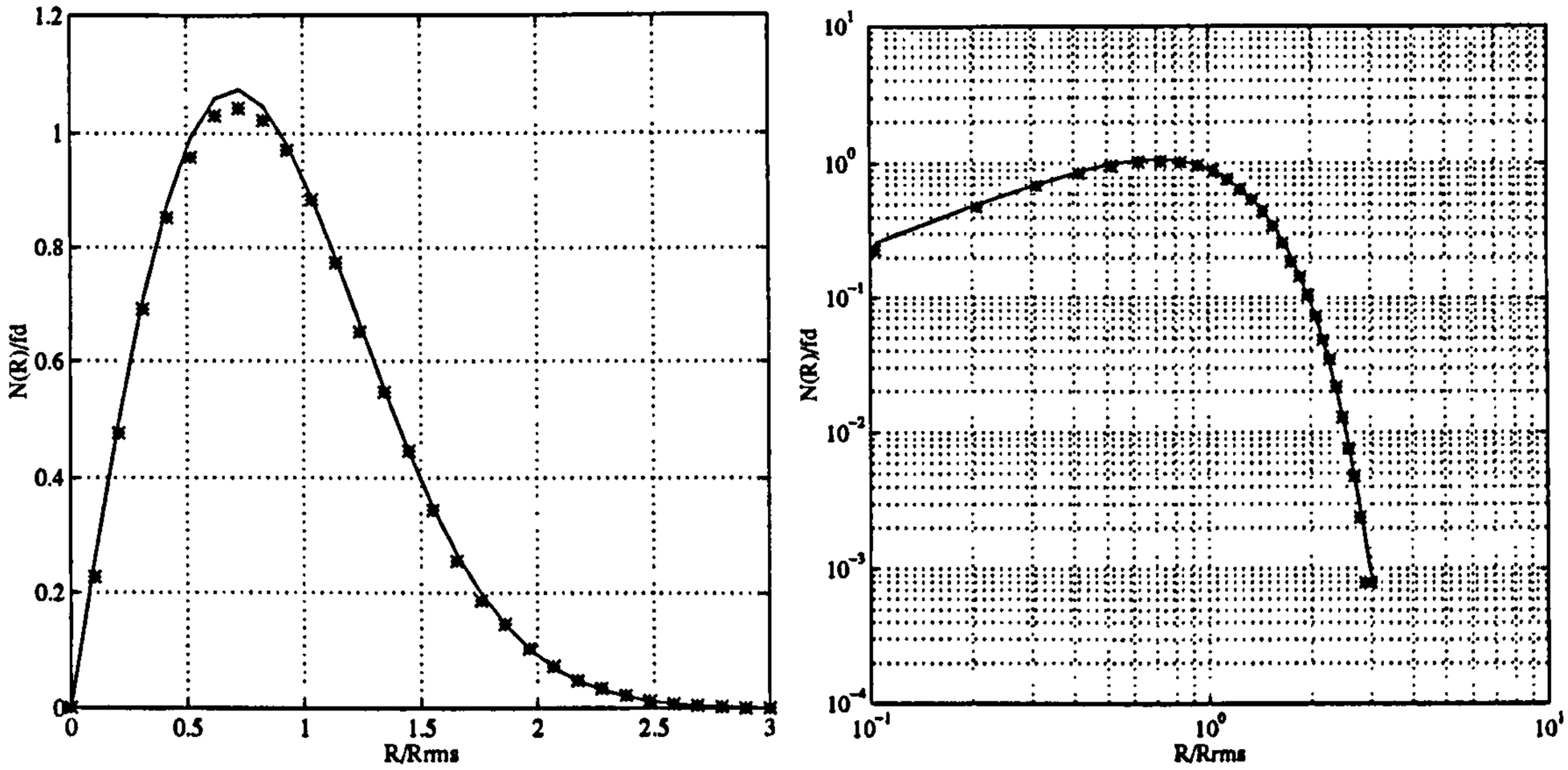



(c) Boxcar Window & $N = 256$




(d) Boxcar Window & $N = 64$

Figure 4.8: Ideal and simulated autocorrelation functions for the narrowband fading channel.




(a) Linear Scale

(b) Logarithmic Scale

Figure 4.9: Simulated Level Crossing Rates vs Theory for $f_d = 0.1$ & $N = 256$ taps in the filter design.

it seems that the accuracy with which one has to model the channel is dependent to some extent on the receiver structure. If one is making symbol by symbol decisions then the autocorrelation function need only be accurate for periods of the order of the symbol. For different receiver structures, incorporating more memory, such as Viterbi decoders or other trellis-searching algorithms, then the channel characteristics have to be more accurate.

Given two correlated random processes, with different noise shaping filters, we can approximately equate their autocorrelation functions by equating their RMS values. We shall demonstrate this now for the ideal Gan's Power Spectral Density and that produced by an ideal brick-wall filter (which we will later approximate with a high-order Butterworth). Thus for,

$$W(f) = \frac{1}{\pi\sqrt{f_d^2 - f^2}} \tag{4.12}$$

the RMS value of the frequency is,

$$f_{rms}^2 = \int_{-f_d}^{+f_d} f^2 \frac{1}{\pi\sqrt{f_d^2 - f^2}}\, df = \frac{f_d^2}{2} \tag{4.13}$$

But for an ideal brick-wall filter,

$$f_{rms}^2 = \int_{-f_d}^{+f_d} f^2 \frac{1}{2\,f_d} = f_d^2/3 \tag{4.14}$$

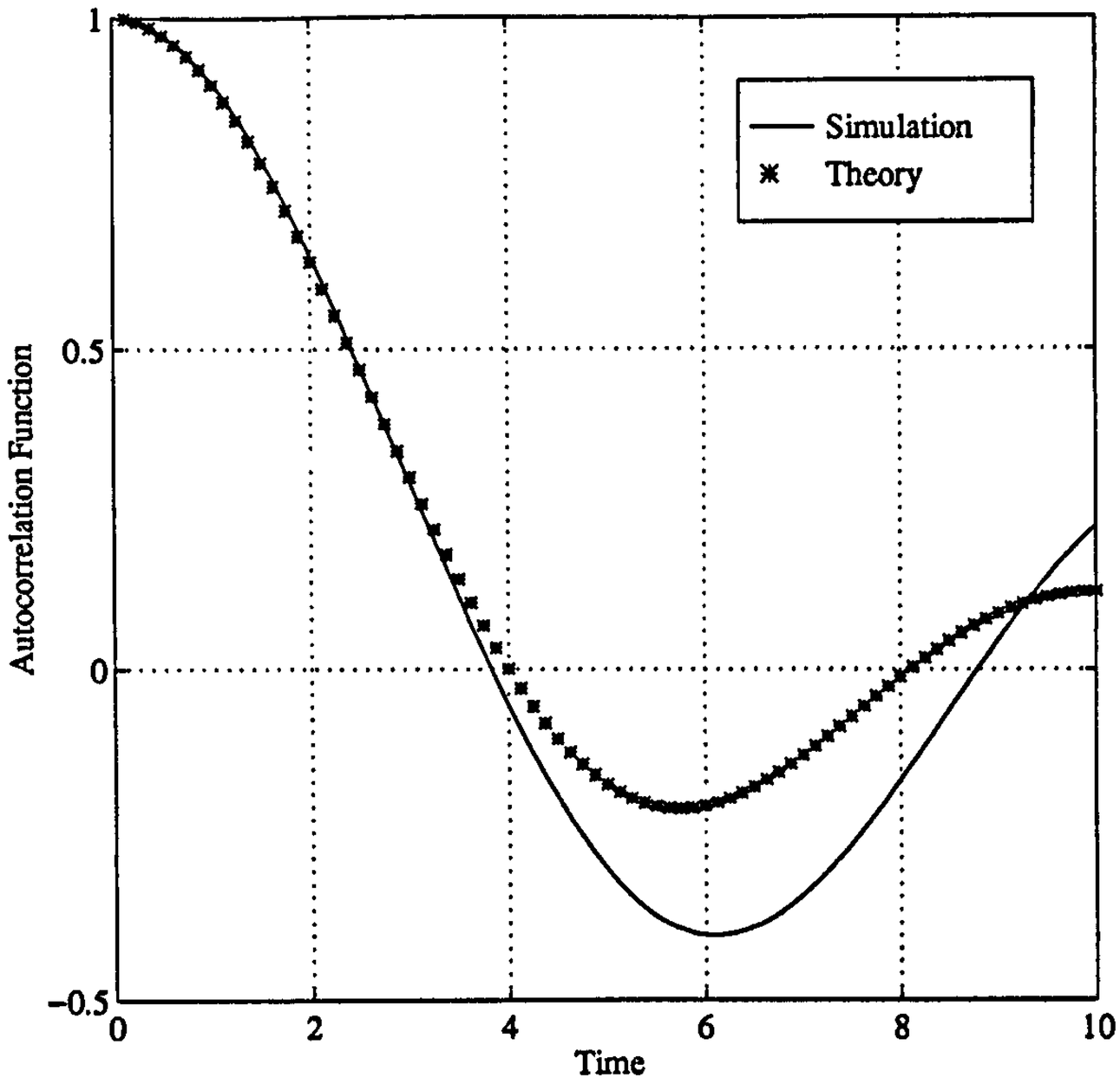



Figure 4.10: Comparison of the autocorrelation function of white Gaussian noise passed through an 8th-order Butterworth filter with 3dB cutoff set according to the theory. Also plotted is the ideal autocorrelation function for the urban mobile channel.

This then implies that we can approximate a narrowband fading process with a Power Spectral Density given by the Gan's model by using a brick-wall filter with a cutoff frequency of,

$$f_c = \sqrt{\frac{3}{2}}\, f_d \tag{4.15}$$

We can illustrate this theory by comparing the autocorrelation function of the narrowband fading process delivered by an $8^{\text{th}}$-order Butterworth filter, but with 3dB cutoff frequency set to $\sqrt{3/2}\, f_d$ and $f_d = 0.1$. The results are as shown in figure 4.10.

It is evident that the correlation provided by the Butterworth filter and the ideal filter are close for time intervals of the order of four symbols. If the value of $f_d$ were much smaller, as it often is, then there would be a close correspondence for even longer intervals.

The conclusion of this work is that rather than using a complicated design procedure for the correlation shaping filter one may be able to use a simpler filter but set the filter parameters in order to give the correct autocorrelation function.

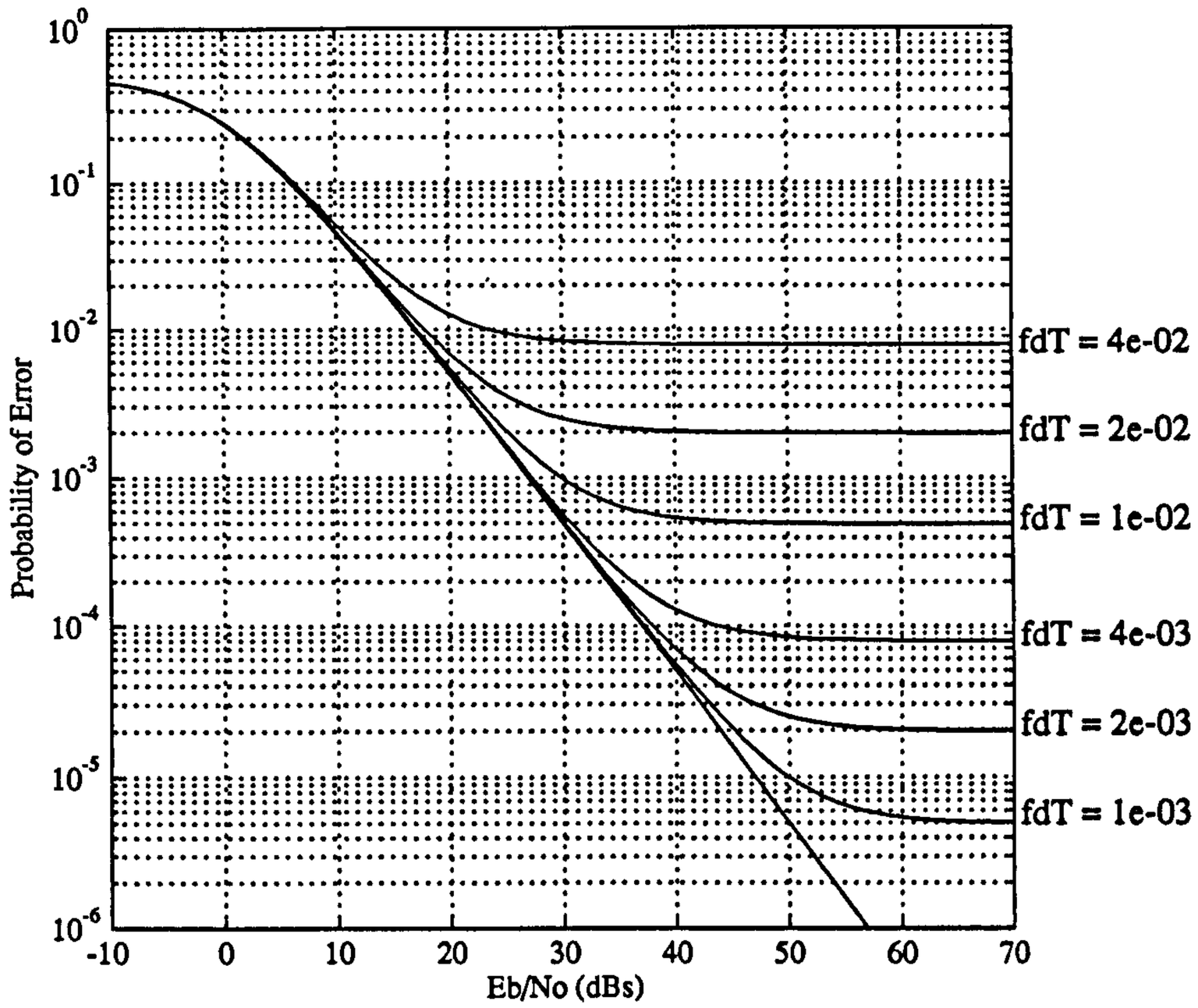



Figure 4.11: Probability of Error for Differentially Detected MSK with the fading rate $f_dT$ a parameter

## 4.4 Ideal Performance of Detection Schemes in the Narrowband Fading Channel

The error rate performance of various ideal detection schemes appropriate for the narrowband fading channel are shown in figures 4.11,4.12, & 4.13 for differential phase detection of MSK, limiter-discriminator detection of MSK and a comparison of the schemes with ideal coherent detection. These graphs are plotted from analytical results given in [51, 52, 53] for differential phase detection, and [54, 55, 56, 57].

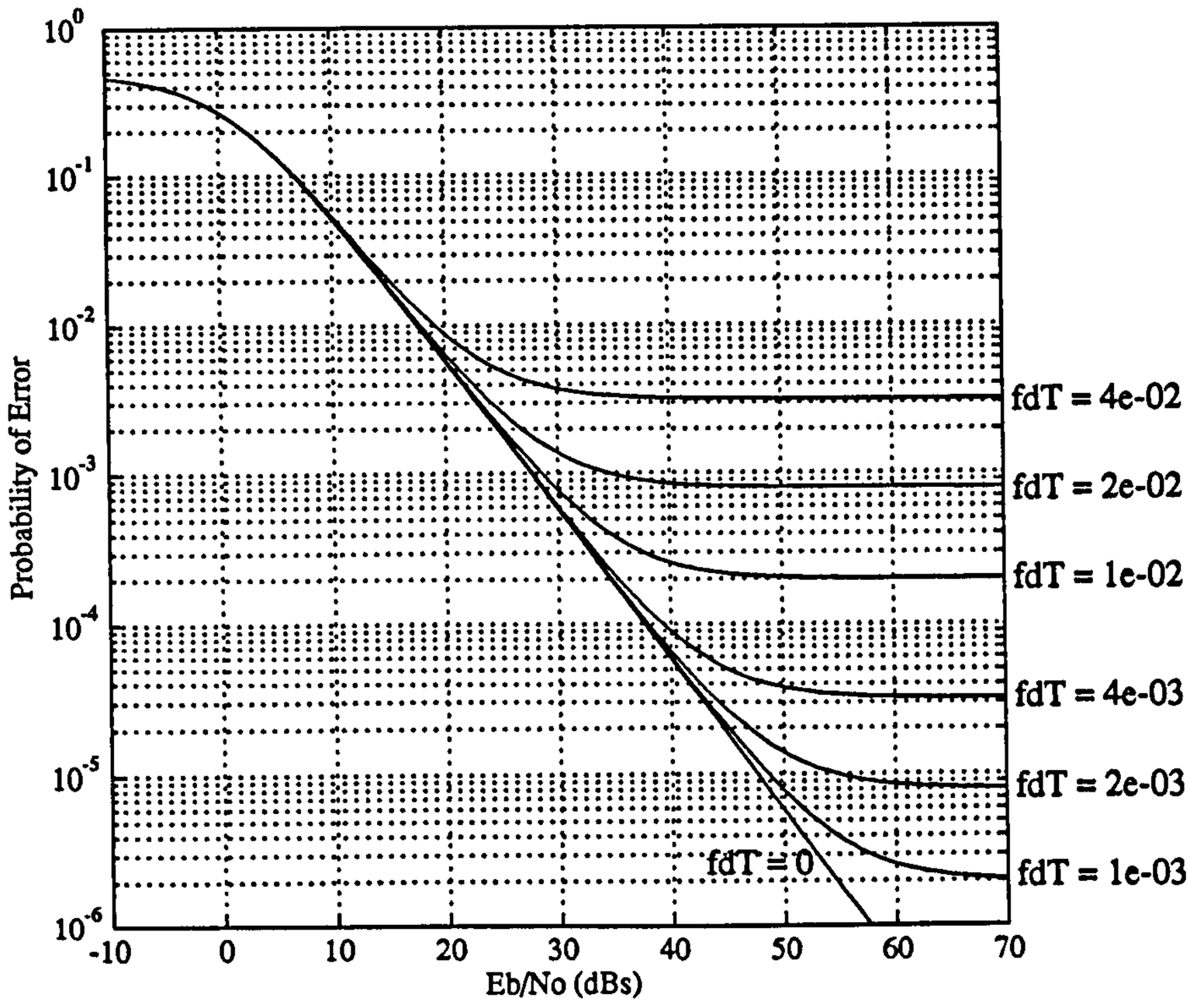



Figure 4.12: Probability of Error for Discriminator Detected MSK with the fading rate $f_d T$ a parameter.

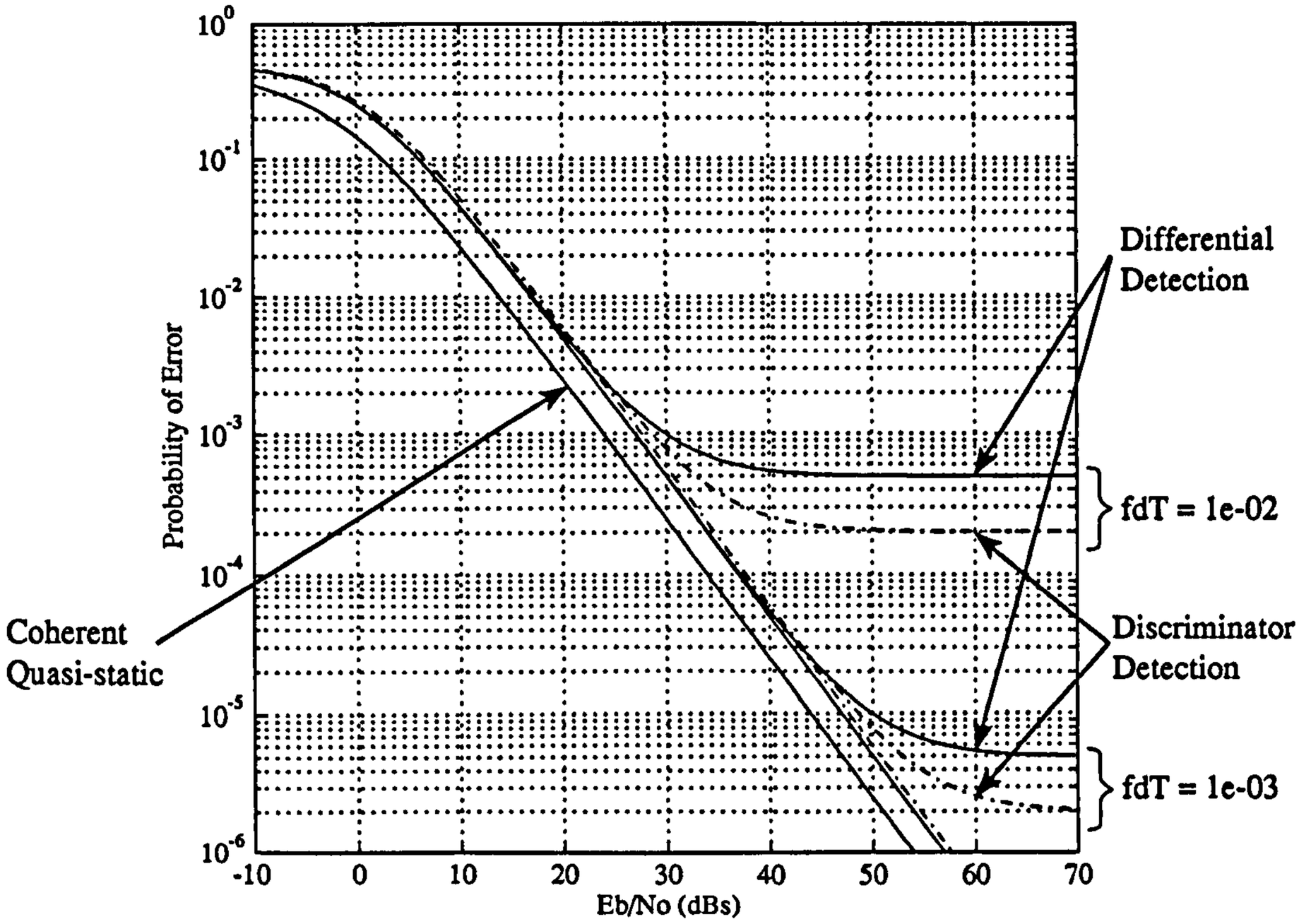



Figure 4.13: Comparison of Coherent, Differential & Discriminator Detected MSK with the fading rate $f_d T$ a parameter.

# Chapter 5

# Frequency Synchronisation

## 5.1 Introduction

The purpose of Frequency Error Detectors (FEDs) is to estimate the carrier frequency of a modulated signal generally embedded in white Gaussian noise. There have been a large number of algorithms proposed in the literature and what is of interest in this work is to compare and contrast these different schemes and to perhaps give some indication of where their use is most appropriate.

The purpose of an FED is to give an indication of the error in the carrier frequency where the signal is generally outside of the receiver filter bandwidth. This error can be used in a feed-forward or feedback manner in a Frequency-Locked Loop to correct for the frequency error bringing the received signal into the centre of the receiver filter bandwidth. Only once the signal is within the receiver filter bandwidth can the functions of symbol timing and carrier phase recovery be applied as these rely on having matched-filter outputs. Frequency acquisition is therefore one of the first functions that will be performed in a receiver where appreciable frequency offsets are expected, with the possible exception of Automatic Gain Control (AGC). Typically large Doppler offsets may be experienced in the Satellite channel and so it is here that such frequency acquisition circuits as discussed in this thesis are used. Frequency errors may be as large as the reciprocal of the baud rate, that is $1/T$, and so the signal could be *completely* outside the receiver bandwidth in the absence of any frequency correction.

It is important to make the distinction between aided-acquisition in Phase-Locked Loops and frequency acquisition. In the aided acquisition a frequency detector is used in parallel with a phase detector to alleviate the problems of cycle-slipping and the associated long acquisition times whereas in frequency acquisition the initial frequency offsets may be much larger. Acquisition structures capable of dealing with offsets of the order of $1/T$ will *generally* concern us in this thesis – with one exception. It has been noted by some authors [58] that when using feed-forward carrier phase synchronisation (for example, in short burst TDMA applications), a small frequency offset can substantially increase the phase error variance in the subsequent carrier phase recovery circuit. Thus in such an application frequency correction is needed where the frequency errors may be much smaller than $1/T$. Such FEDs are referred

to as rotational detectors in the literature and are treated separately later.

As will be seen in this thesis there are a number of apparently different structures for FEDs. This difference may in part be due to the different approaches by different researchers in this field. A recent paper by Moeneclaey [22] has shown the equivalence between some of the different FED structures. One of the early papers dealt with aided-acquisition and hence discussed the rotational detector in [17]. Gardner undertook work on quadricorrelators, dual-filter detectors and rotational detectors in [9] and later derived FEDs via Maximum-Likelihood derivation in [12]; this work was undertaken for the European Space Agency, reflecting the interest in being able to cope with large frequency offsets in receiver design. Other work on quadricorrelators and dual-filter detectors (it is also referred to as differential-power measurement DPM) has been carried out in [18]. Much work is still going on in this area with recent publications on the design and performance of quadricorrelators in [19] and [20]. Work comparing the performance of the digital quadricorrelator and the maximum-likelihood FED has been undertaken by D'Andrea in [21]. Also of use is [28] which gives the theoretically optimum bounds for frequency estimators – referred to as the "Modified Cramer Rao Bound". This bound is applied later to the performance of both symbol timing and frequency recovery loops.

In the next section we outline the basic algorithms for carrier frequency estimation and then go on to compare and contrast the performance of the different schemes. Work is outlined on integration of different system functions together and an example of joint timing, carrier phase and frequency recovery system is given.

A unified treatment of symbol-timing error detectors for non-offset modulation schemes is also given.

## 5.2 Algorithms for Large Frequency Offsets

In this section we will discuss the Dual-Filter Detector, the Balanced-Quadricorrelator and the Maximum-Likelihood frequency error detectors. These detectors are all capable of dealing with frequency offsets of the order of the reciprocal of the baud rate.

### 5.2.1 The Balanced Quadricorrelator (BQ-FED)

The quadricorrelator is derived from the *ideal* structure for a frequency error detector. We start therefore by showing how this structure is derived by considering an un-modulated carrier signal.

#### Un-modulated Carrier

Consider first the simple scenario where we have an un-modulated carrier. We do this here to illustrate the different approaches of the Balanced Quadricorrelator (BQ) and the Dual Filter Detector (DFD). The equivalent complex baseband representation for an un-modulated

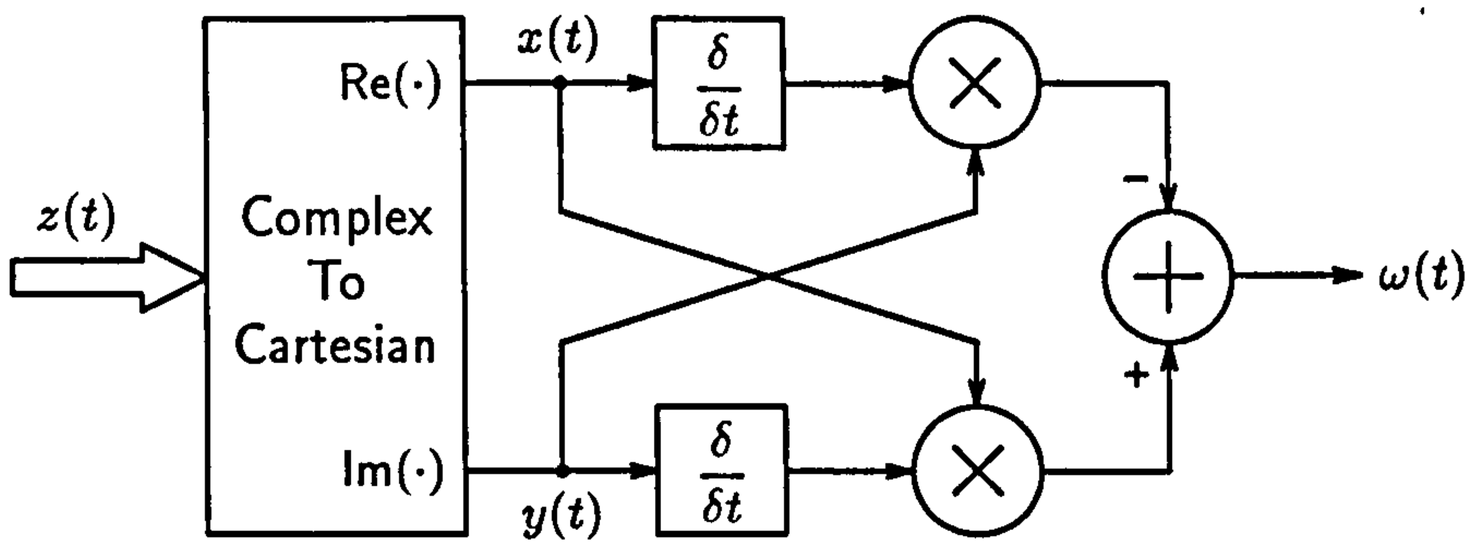



Figure 5.1: Ideal frequency error detector based on balanced quadricorrelator

carrier is given by a simple complex phasor $z(t)$ as

$$z(t) = \exp(j\,\omega_c\,t) = x(t) + j\,y(t) \tag{5.1}$$

where $\omega_c$ is the angular frequency of the carrier and $x(t)$ and $y(t)$ represent the in-phase and quadrature components of the signal. These would be produced in an orthogonal-coherent detector. The function of the FED is to produce an estimate of the carrier frequency from these in-phase and quadrature components. It can be seen that the angular frequency in (5.1) is given as the derivative of the imaginary component of the argument as

$$\begin{aligned} \omega_c &= \frac{d}{dt}\,\mathrm{Im}\,\{\ln z(t)\} \\ &= \mathrm{Im}\left\{\frac{\dot{z}(t)}{z(t)}\right\} \\ &= \frac{x(t)\,\dot{y}(t) - y(t)\,\dot{x}(t)}{x^2(t) + y^2(t)} \end{aligned} \tag{5.2}$$

In the above $\mathrm{Im}\{\cdot\}$ represents the imaginary component. As the amplitude of the phasor in (5.1) is unity we can omit the denominator in (5.2) to give the balanced quadricorrelator structure shown in figure 5.1.

The FED in figure 5.1 is a theoretical ideal and a digital implementation will be limited by the approximation that has to be made in the differentiation. We make the following approximation for the derivative.

$$\dot{x}(t) \approx \frac{x[n] - x[n-1]}{T_s} \tag{5.3}$$

where $T_s$ is the sampling period. We make a similar approximation for $\dot{y}(t)$ and substitute in (5.2) (again ignoring the denominator) to give

$$\omega_n = \frac{x[n-1]\,y[n] - x[n]\,y[n-1]}{T_s} \tag{5.4}$$

We can check that (5.4) gives the required result by defining a discrete-time phasor $z[n]$ as

$$z[n] = z(t)|_{t=n\,T_s} = \cos(\omega_c\,n\,T_s) + j\,\sin(\omega_c\,n\,T_s) \tag{5.5}$$

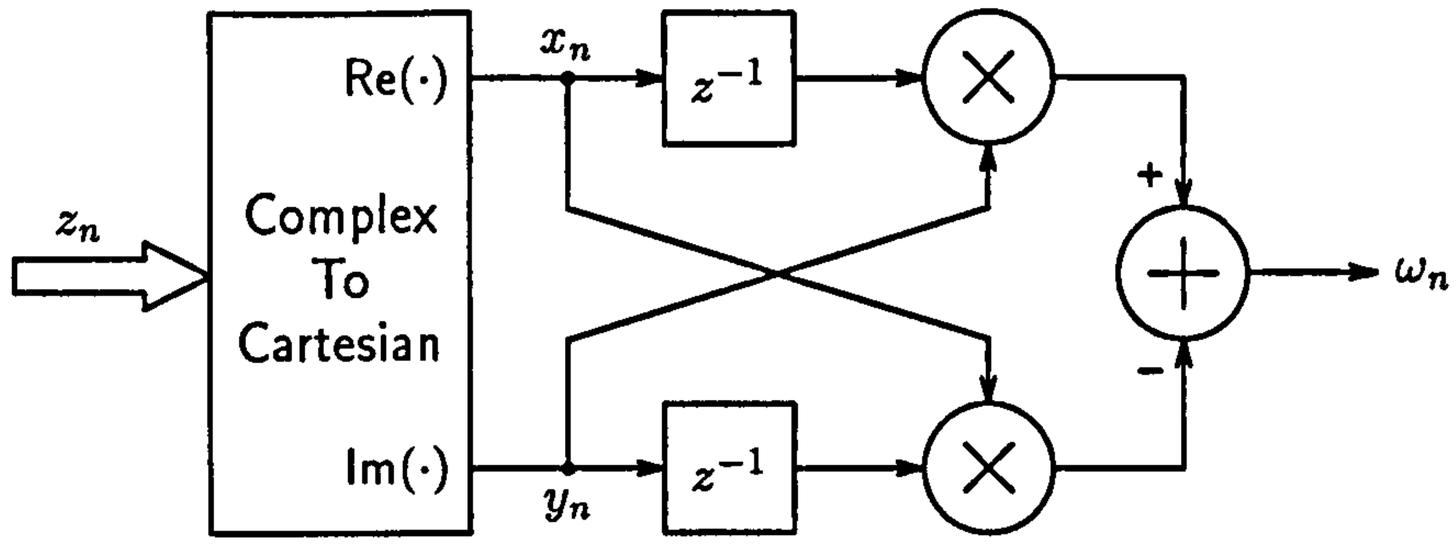



Figure 5.2: Digital implementation of frequency error detector based on balanced quadricorrelator

and substituting this into (5.4) and simplifying we get

$$\omega_c[n] = \frac{\sin(\omega_c n T_s)}{T_s} \tag{5.6}$$

The detector "S-curve" of the frequency-error detector (voltage-out as a function of frequency-error in) is represented by (5.6) and can only be considered linear for small values of frequency error. Such an FED will allow tracking of frequency errors of $\Delta\omega_c = \pm\pi/2$ or $\Delta f_c = \pm 1/4\,T_s$. The structure of the digital frequency error detector is as shown in figure 5.2 and comes directly from (5.4).

### Modulated Carrier

The problem with the analysis above is that we have ignored the data that will almost certainly be modulated on to the carrier waveform. The structure depicted by (5.4) gives the *instantaneous* frequency of the signal. This is not the same as frequency error. The instantaneous frequency of the carrier may be a rapidly varying quantity determined by the modulation. If the data is modulated via some form of Frequency Shift Keying then the output from the FED will not only produce an output proportional to the average frequency error but will also produce noise output due to the modulation. Thus using the direct structure as in figure 5.2 will lead to *self-noise* or *pattern-noise*. This could degrade the tracking performance of the FED and needs to be minimised when the frequency error is zero for correct operation.

Recent work by D'Andrea & Mengali [19, 20, 21] has looked at Balanced Quadricorrelator structures which give zero self-noise at zero frequency error. The structure of these is similar to that shown in figure 5.2 but with appropriate filtering to reduce the effects of self-noise. It should be noted that the rather idealised block in figure 5.2 labelled "Complex To Cartesian" is intended to represent the functionality of an ideal orthogonal *quasi-coherent* detector.

The generic structure of the Balanced Quadricorrelator for Automatic Frequency Control is as shown in figure 5.3. It is found that self-noise can be minimised for both offset and non-offset modulations by appropriate choice of the filters $H(f)$ and $D(f)$. Note that the structure depicted in figure 5.3 can be re-written wholly in the complex domain as shown in figure 5.4. In figure 5.4 $H(f)$ can be seen to be a real pre-filter and $D(f)$ is a real discriminator

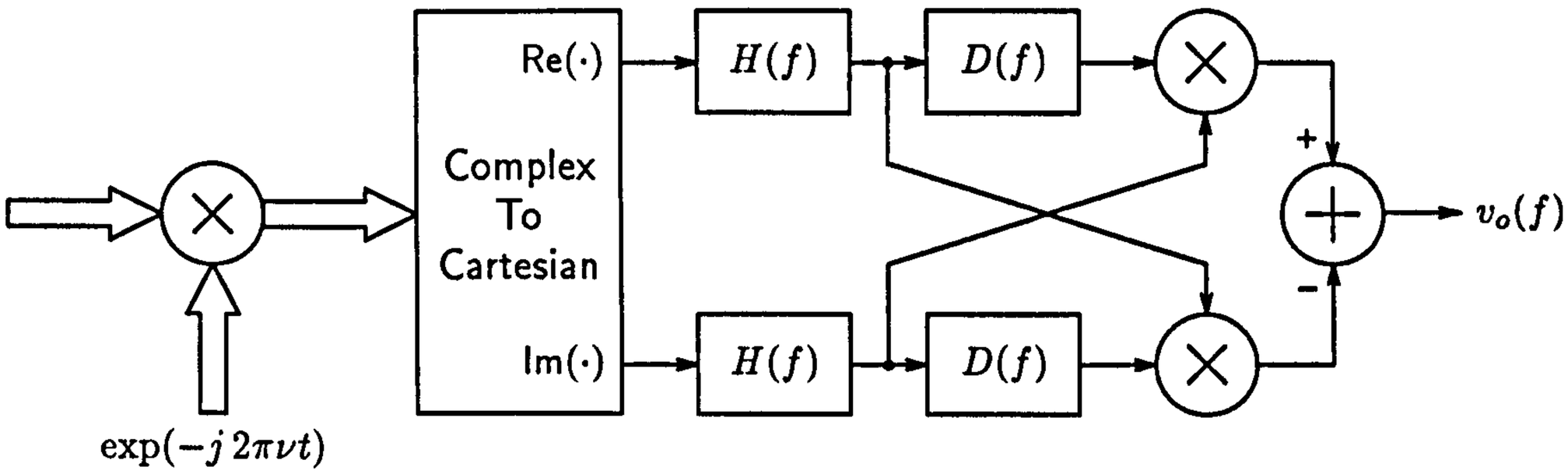



Figure 5.3: Generic Structure of Balanced Quadricorrelator

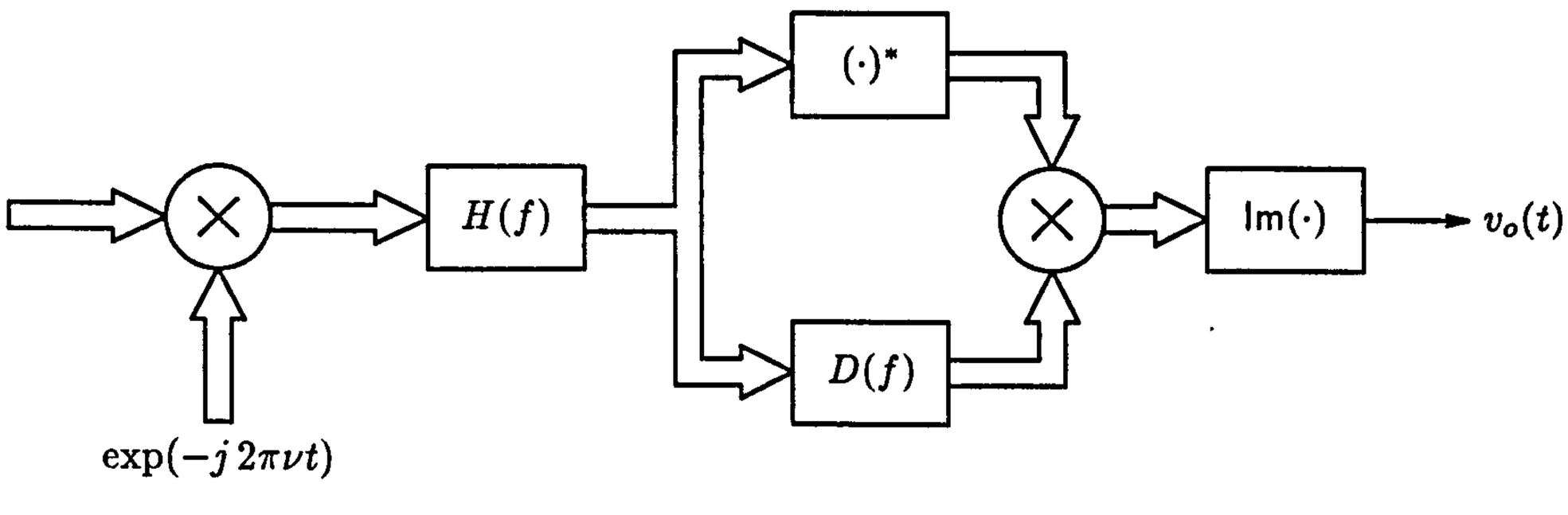



Figure 5.4: Complex Realisation of the Balanced Quadricorrelator

filter. The BQ-FED output can be written in the form $\mathrm{Im}(z_1(t)^* z_2(t))$ which is a measure of the instantaneous phase shift between $z_1(t)$ and $z_2(t)$ as,

$$\mathrm{Im}(z_1(t)^* z_2(t)) = |z_1(t)|\,|z_2(t)|\,\sin(\phi_{z_1}(t) - \phi_{z_2}(t)) \tag{5.7}$$

where,

$$\phi_{z_1}(t) = \arg(z_1(t)) \quad \text{and} \quad \phi_{z_2}(t) = \arg(z_2(t)) \tag{5.8}$$

The mean detector output as a function of frequency error input, the S-curve, is given as,

$$SC(\nu) = \int |G(\nu)|^2 P(f+\nu)\, df \tag{5.9}$$

where $G(f)$ is the baseband pulse shaping and $P(f)$ is written in terms of the $H(f)$ and $D(f)$ above as,

$$P(f) = |H(f)|^2 \,\mathrm{Im}(D(f)) \tag{5.10}$$

As $H(f)$ and $D(f)$ are the transfer functions of real filters, $|H(f)|^2$ is even in $f$ and $\mathrm{Im}(D(f))$ is odd in $f$. Consequently, when $G(f)$ is even in $f$ (as is the case for most modulation formats), the average BQ-FED output is zero for zero frequency error.

A systematic method of designing quadricorrelators for AFC systems is given in [20]. It is shown that the quadricorrelator can be designed so as to make self-noise negligible in the steady state, both with staggered and non-staggered modulations, this is, QPSK and OQPSK type modulation schemes. These structures are referred to as *optimised* quadricorrelators and are different from the original research undertaken by Gardner in [13]. The important point here is that the BQ-FED can track offset modulations, such as MSK, with zero self-noise if the filters are designed correctly. This is not the case at the moment for the other FEDs. No FED similar in structure to the DFD-FED & the ML-FED can track offset modulations with zero self-noise.

A suggestion was made in [20] for a practical implementation of the BQ-FED. It consists of having two 2-pole Butterworth bandpass filters as the $H(f)$, with bandwidth $0.2/T$ and centre frequency $0.6/T$ (and this is the same for QPSK and OQPSK), and the $D(f)$ filter is simply a $T/2$ delay for non-offset modulation and a $T$ delay for offset modulation.

### 5.2.2 The Dual-Filter Detector (DFD-FED)

Gardner undertook a study of the DFD-FED in [9]. The work was motivated by the fact that the BQ-FED is limited by self-noise for modulation schemes with a large alphabet size and the rotational detectors are limited in their range of acquisition. For example, the acquisition range of the rotational FED for QPSK is 1/16 of the baud rate which for many applications is inadequate. It should be noted, however, that the BQ-FED that Gardner dismisses as being plagued by self-noise is that derived by Messerschmitt in [17] and is not the *optimised* BQ-FED derived by D'Andrea in [19, 20]. The block diagram for the DFD-FED is as shown below.

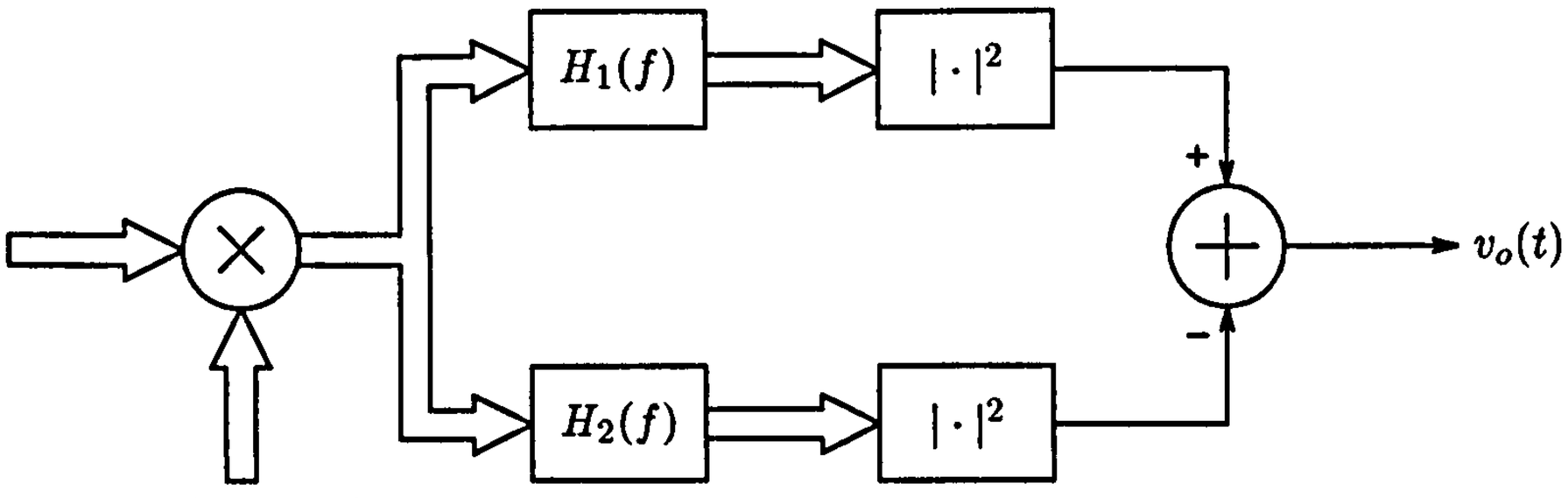



Figure 5.5: Structure of DFD-FED

The DFD-FED is perhaps the easiest to understand in terms of its operation by referring to figure 5.6. If the signal spectrum is centred on the frequency axis then the power outputs from the two filters will be equal. Thus when the frequency error is zero, the mean output from the DFD-FED will also be zero. If the spectrum is then shifted in a positive or negative direction along the frequency axis the output from the DFD-FED will increase or decrease accordingly. This error-signal is generally used in a feedback manner to correct for the frequency error.

The average output of the DFD-FED is given by,

$$SC(\nu) = \int \{|H_1(f)|^2 - |H_2(f)|^2\}\, G(f+\nu)\, df \tag{5.11}$$

Again, for most modulations $G(f)$ is even in $f$, then $|H_1(f)| = |H_2(-f)|$ guarantees that the average DFD-FED output will be zero when $\nu = 0$. This means that the responses of $H_1(f)$ and $H_2(f)$ are symmetrically disposed about zero Hz (complex frequency) as indicated in figure 5.6.

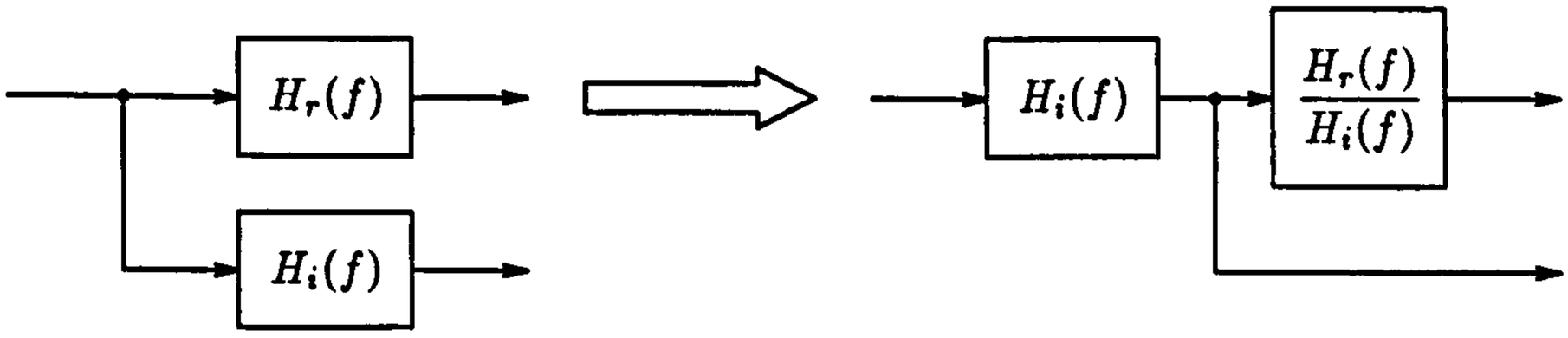
The computation of the DFD output requires two complex-valued filtering operations on complex signals, which in general corresponds to eight real-valued filtering operations on real signals. However, by putting,

$$H_1(f) = H_2^*(-f) \quad \text{or equivalently} \quad h_1(t) = h_2^*(t) \tag{5.12}$$

and hence we can split the impulse responses into real and imaginary components as,

$$h_1(t) = h_r(t) + j\, h_i(t) \quad \text{and} \quad h_2(t) = h_r(t) - j\, h_i(t) \tag{5.13}$$

the structure of the DFD-FED can be modified as shown in figure 5.7. This structure now looks remarkably similar to the BQ-FED structure shown in figure 5.3. In fact, it has been shown in [22] that the DFD-FED and the BQ-FED are mathematically equivalent. It is easy to see the equivalence between these two structures by noting the following transformation.



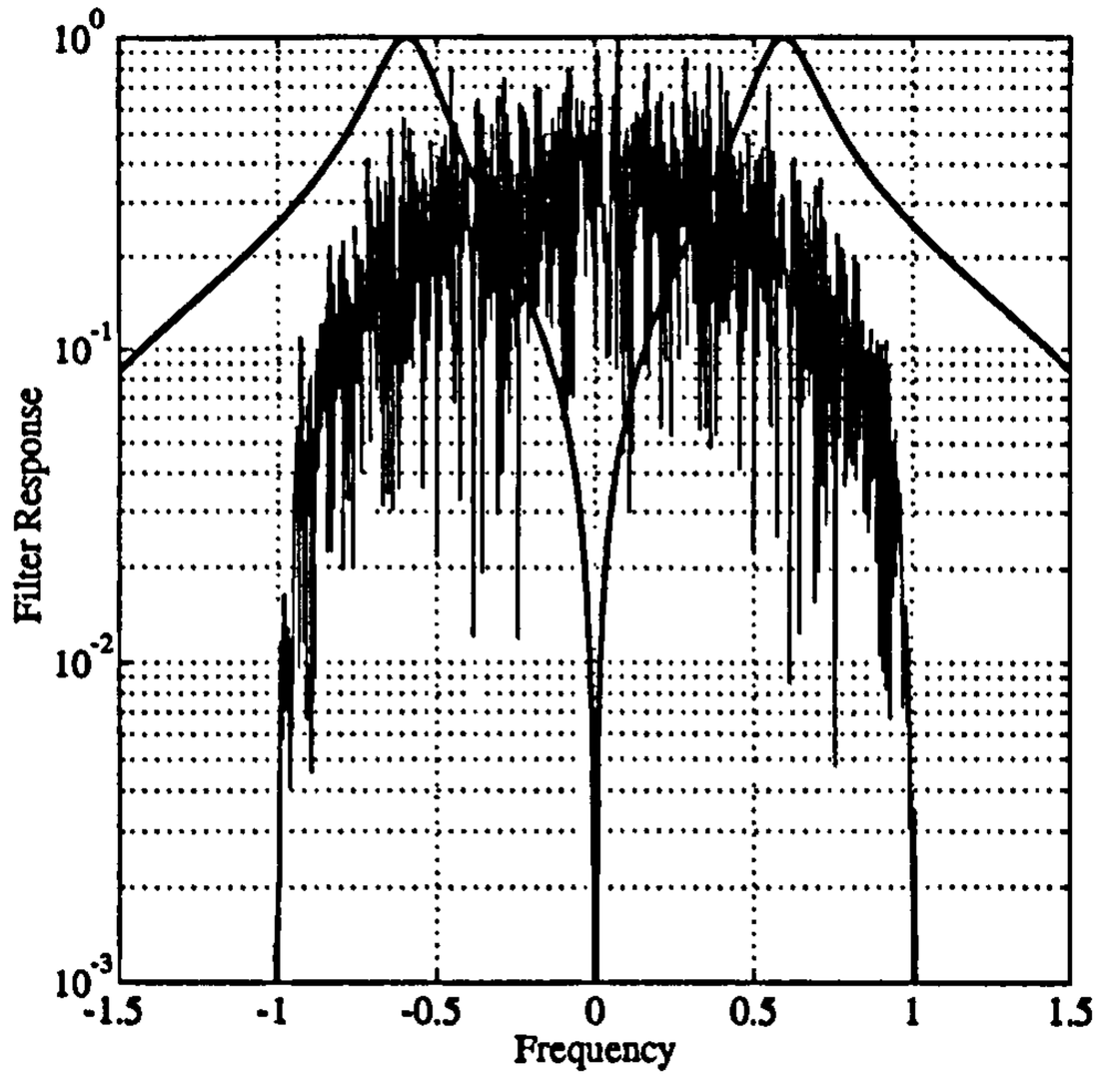



Figure 5.6: Spectrum of Raised Cosine signal and DFD filter response

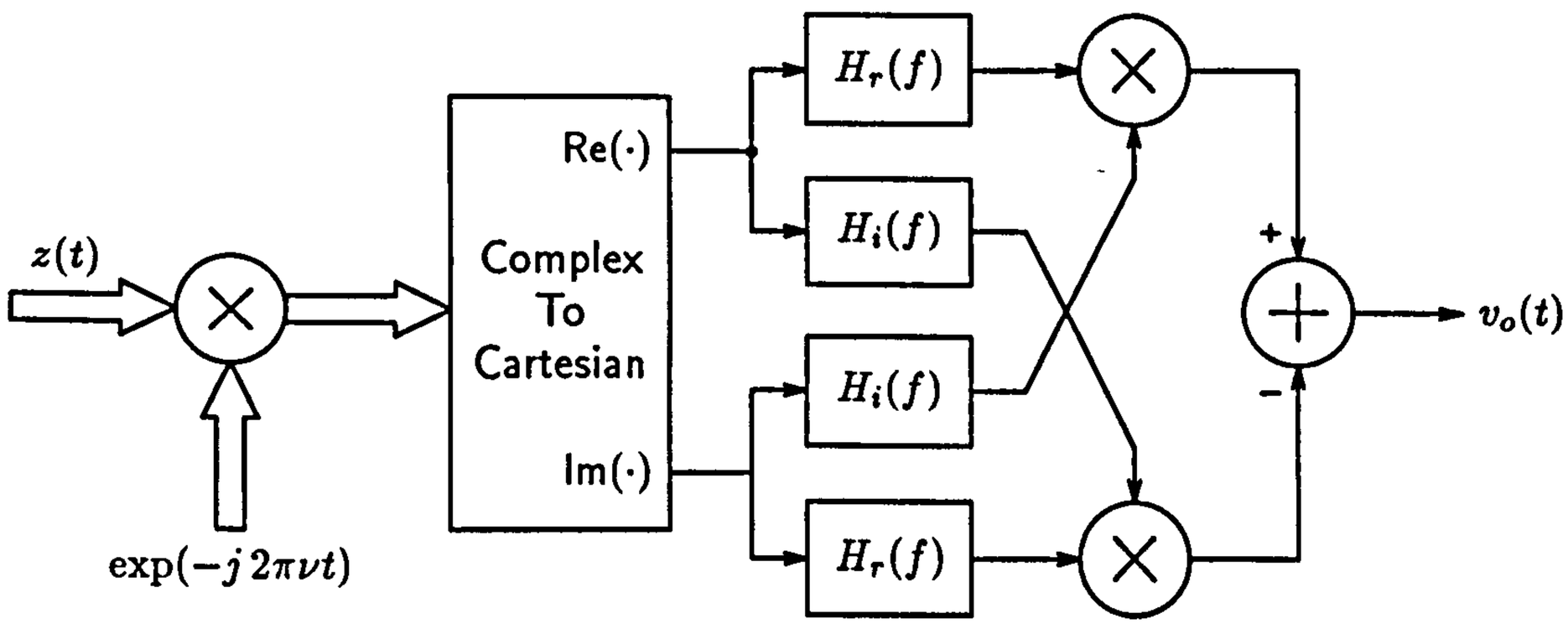



Figure 5.7: Simplified Structure of DFD-FED

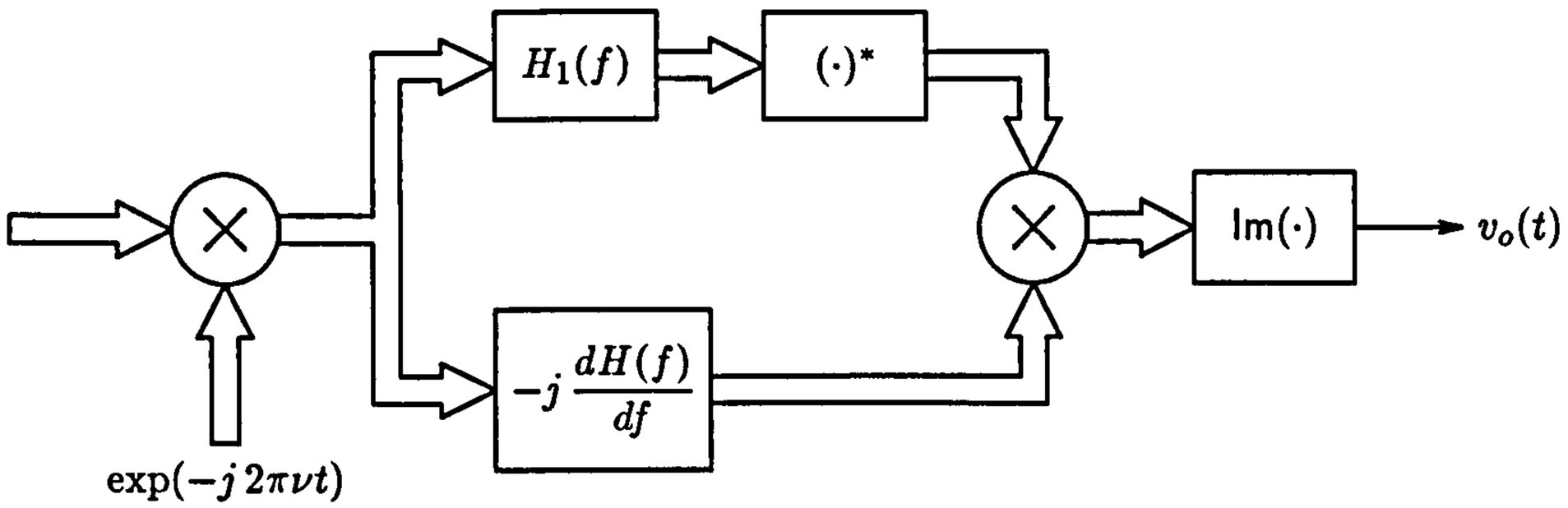



Figure 5.8: Maximum-Likelihood Frequency Error Detector

### 5.2.3 The Maximum-Likelihood Frequency Error Detector (ML-FED)

In the literature there are two frequency error detectors derived via the Maximum Likelihood principle. The first is a non-data aided (NDA) maximum-likelihood (ML) algorithm for frequency-offset estimation which is independent of timing. This algorithm has been derived by Moeneclaey in [22]. It has a structure as follows,

Note that the ML-FED requires two filters: $H(f)$, referred to as the signal matched filter; and $-j\,dH(f)/df$, referred to as the frequency matched filter. Note that if $h(t)$ is the impulse response of the filter $H(f)$, then $2\pi t h(t)$ is the impulse response of the filter $-j\,dH(f)/df$.

In Gardner's derivation of the ML-FED [12] a structure is derived similar to that in figure 5.8 except that the output is sampled. The output is sampled at the baud rate for non-offset modulations and twice the baud rate for offset modulations. Although the output is sampled this does not necessarily imply that one must have symbol timing in order to use this scheme, the sampled ML-FED will work whatever the sampling phase. It will, however, be optimal, in the sense that the frequency error signal will have minimum variance, if the sampling phase is correct. If the sampling error is zero then the sampled ML-FED will outperform the non-sampled version in terms of estimation error variance. If the timing error is a quarter of a symbol then the two schemes perform equally well. If, however, the timing error approaches the worst case scenario of $\pm T/2$ then the estimator error variance becomes very large. It is possible to acquire timing and frequency using the sampled ML-FED in combination with a NDA timing recovery loop. In particular, a novel combination of sampled ML-FED and amplitude directed (AD) timing recovery is explored, together with an examination of where in such a configuration one should place the interpolation filters.

## 5.3 Algorithms for Small Frequency Offsets

### 5.3.1 The Rotational Frequency Error Detector (ROT-FED)

As was mentioned earlier rotational detectors are appropriate where the frequency error is small, that is $F_e \ll 1/T$. A rotational detector operates on the complex matched-filter output samples and the optimum timing instant. The scheme is only suitable for $M$-PSK type

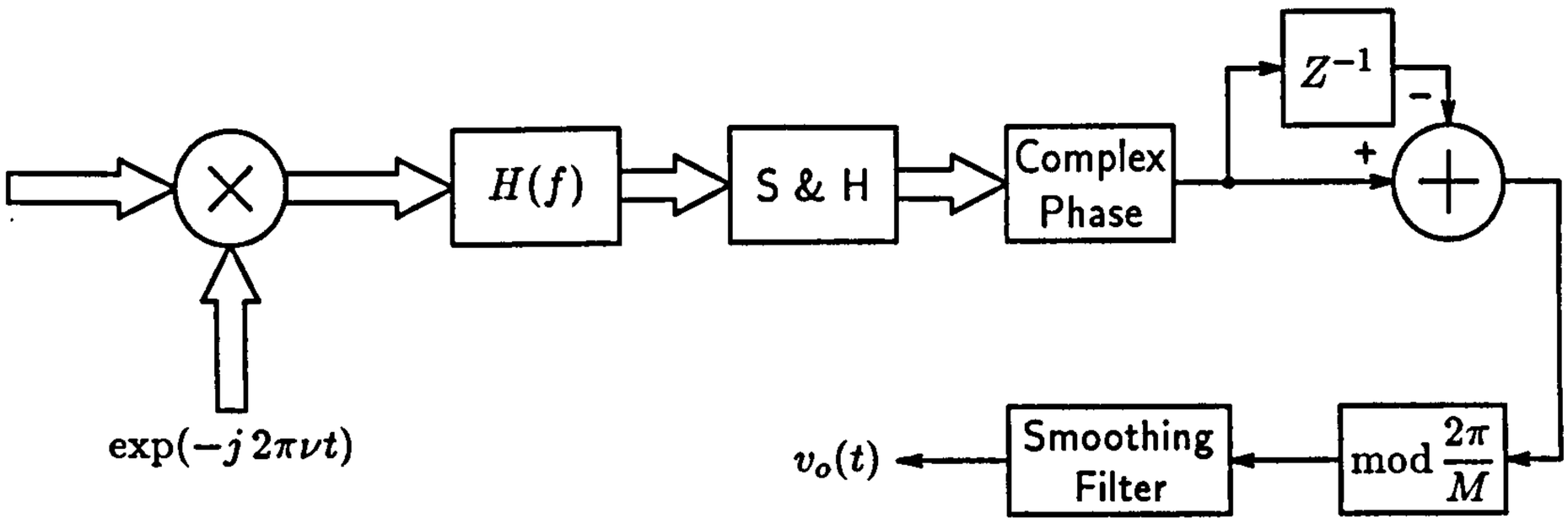



Figure 5.9: Structure of Feedforward Rotational Frequency Error Detector

modulation schemes. Thus there has to be little distortion due to frequency error through the receiver filter and the timing has to have been recovered. It is appropriate where the signal constellation points are rotating slowly. By considering the rotation of the constellation points in the scatter diagram from one symbol period to the next it is possible to determine exactly what the frequency offset is. It is possible then to correct for the frequency offset in a feedforward manner. We consider the scheme due to Bellini [58] which gives an estimate of $F_e$ as

$$2\pi F_e T = \sum_{k=-K+1}^{K} \beta_k \left(\phi_k - \phi_{k-1}\right) \mathrm{mod}(2\pi/M) \tag{5.14}$$

In the above equation the $\beta_k$ are the filter coefficients. Correct operation of the above algorithm requires that $F_e < 1/2MT$. If one sets the above filter coefficients to $\beta_k = 1/K - 1$ then the summation is simply a sliding average filter operation. The structure of the rotational FED is then as shown in figure 5.9. Note now that this frequency error detector is fundamentally different in structure from those studied previously. It is used in a feedforward manner to correct for small frequency errors.

## 5.4 Theoretical Bounds on Performance of Frequency Error Detectors

New results on the performance of both timing and frequency error detectors have been presented by D'Andrea in [28]. In this paper entitled, "The Modified Cramer-Rao Bound and Its Application to Synchronisation Problems," the authors develop a new lower bound on the error variance in parameter estimation and apply this theory to the special case of frequency estimation. The analysis is involved and relies on work done previously by Mengali in [59]. Only the pertinent results will be quoted here.

The theoretical closed-loop performance for the ML-FED shown in figure 5.8 is, after some calculation, given by

$$\sigma^2_{\nu \mathrm{ML}} = \frac{4\alpha B_L T}{\pi^2 T^2} \frac{1}{E_s/N_o} \left(1 + \frac{1}{E_s/N_o}\right) \tag{5.15}$$

where $B_L$ is the loop noise bandwidth. This is considerably worse than what is theoretically possible for parameter estimation of carrier frequency. This is because the estimate of the carrier frequency is conditioned upon the timing phase. The lower bound, the performance of the best possible estimator, is given by the MCRB and has two forms. The first form is applicable to a feedforward estimator which uses sliding-window over $LT$ seconds. For this case we have,

$$MCRB_\nu = \frac{3T}{2\pi^2(LT)^3}\frac{1}{E_s/N_o} \tag{5.16}$$

For the feedback frequency locked loop we need to replace the sliding-window with an equivalent loop noise bandwidth of $B_L = 1/(2LT)$. We then get,

$$MCRB_\nu = \frac{12(B_LT)^3}{\pi^2T^2}\frac{1}{E_s/N_o} \tag{5.17}$$

In the next section we compare the simulated results for the various frequency error estimators with these analytical results. It transpires upon comparing equations (5.15) and (5.17) that the ML-FED performs considerably worse than what is theoretically possible, whereas, as we shall see when we come on to discuss the simulation results, the rotational frequency error detectors have performance which meets the MCRB. The rotational frequency error detectors are feedforward structures and hence described by equation (5.16).

The other point worth noting is that these expressions are only applicable for high-values of signal to noise. The expressions of variance of estimators at low values of signal to noise may involve more than one power of the loop noise bandwidth. This is because of the synchroniser self-noise, which generally implies that the noise in the loop cannot be considered as white.

## 5.5 Performance of Frequency-Error Detectors

The two frequency error detectors discussed in the previous section were the Balanced Quadricorrelator (BQ-FED) and the Maximum-Likelihood frequency error detector (ML-FED). The derivation of the structure for the two FEDs was discussed previously; here we shall compare their performance. The comparative evaluation between the two FEDs will be in terms of their open and closed-loop performance. Here useful analytical results are obtained and compared with the results obtained by simulation. Determination of the behaviour of these frequency locked loops (FLLs) would be tedious, if not impossible, using analytical methods alone but it is useful to pursue the analysis as far as possible as it gives us confidence in the predictions made by simulation. The open-loop performance is characterised in terms of the FED S-curve, which gives the mean error signal out as a function of the input frequency error. Also of interest is the open-loop variance of the error signal as a function of input frequency error. Ideally the variance of the error signal at zero frequency error should be as small as possible. This is indicative of the amount of self-noise in the FED.

The closed-loop performance of the FEDs is examined in terms of the acquisition and tracking properties of the frequency locked loop (FLL). The two loops are normalised so that they have the same equivalent loop bandwidth. Analytical expressions are given to determine

the loop filter structure (be the loop first or second order) in terms of the required loop bandwidth and the open-loop gain (or the slope of the S-curve characteristic). Minimum requirements are determined for the ML-FLL in terms of the number of filter taps required for the signal and frequency matched filters and attention is given to the appropriate design method. It is noted that for the ML-FLL timing information is needed to achieve optimum performance. Novel methods are examined which combine timing recovery and the ML-FLL together.

The simulation and analytical work undertaken as part of this study involve not only the CAAD tool SPW but also MATLAB and MAPLE V and programs and scripts written using these tools have been included where it is deemed useful to do so.

## 5.6 Pulse Shapes Used in S-Curve Derivations

Generally the frequency error detector (FED) will operate on a signal prior to matched-filtering. Frequency errors may be as large as $\pm 1/T$ which could imply that very little power is actually in the receiver filter, so frequency error correction has to be done before matched filtering. In the simulations undertaken in this work we have assumed that the transmitter and receiver filters are both root-Nyquist with cosine-rolloff characteristics $H(f)$. Therefore, the overall transfer function of the link, after matched filtering, is $G(f) = H(f)\,H^*(f)$, which is a full-Nyquist cosine rolloff shape. These pulses have been defined analytically in equation (3.1). To implement the ML-FED we will need the derivatives of the equations in (3.1), the so-called frequency-matched filter, and this has been done already in equation (3.2).

Plots of $H(f)$ and $H'(f)$ are shown in figure 3.2 for the Root-Raised Cosine filters represented by (3.1) & (3.2) for $\alpha = 0.5$. In the analytical and simulation results that follow we assume that the complex baseband data has been filtered by a Root Raised Cosine filter.

## 5.7 Open-Loop Performance

We shall now look at the open-loop performance of the two frequency error detectors. We are interested in the mean and variance of the error signal out of the two FEDs as functions of the input frequency error with perhaps the excess bandwidth of the pulse shaping as a parameter. It is important to determine the slope of the FED at zero frequency error as this determines the loop bandwidth when operating closed-loop.

### 5.7.1 ML-FED Characterisation

According to Gardner [12] the S-Curve for the frequency error detector derived by maximum-likelihood methods is given by the following:

$$\begin{aligned} U_\nu(\nu,\tau) = -\frac{A_0^2}{T}\Bigg\{ & \int_{-\infty}^{+\infty} H(-f)\,H(f-\nu)\,H'(f)\,H(\nu-f)\,df \\ & + \cos\left(\frac{2\pi\tau}{T}\right)\int_{-\infty}^{+\infty} H\left(-\frac{1}{T}-f\right)\,H\left(\frac{1}{T}+f-\nu\right)\,H'(f)\,H(\nu-f)\,df \\ & + \cos\left(\frac{2\pi\tau}{T}\right)\int_{-\infty}^{+\infty} H\left(\frac{1}{T}-f\right)\,H\left(-\frac{1}{T}+f-\nu\right)\,H'(f)\,H(\nu-f)\,df\Bigg\} \end{aligned} \tag{5.18}$$

In the above $T$ is the pulse width, $H(f)$ and $H'(f)$ are the signal and frequency matched filters respectively, $\tau$ is the timing error, $\nu$ is the frequency error and $A_0$ is the signal amplitude. The above equation is only applicable to non-offset modulation. Note the dependence on the timing error in (5.18). Equation (5.18) has to be determined for the piecewise continuous functions given by (3.1) & (3.2). By considering the regions of overlap in the integrals we can determine analytically a closed-form solution to the S-curve for the ML-FED as follows.

Rewrite the above equations for the signal and frequency matched filters using the following:

$$\begin{aligned} G_{11}(f) &= \cos\left\{\frac{\pi}{2\,\alpha}\left(f+\frac{1}{2}-\frac{\alpha}{2}\right)\right\} \\ G_{12}(f) &= \cos\left\{\frac{\pi}{2\,\alpha}\left(f-\frac{1}{2}+\frac{\alpha}{2}\right)\right\} \\ G_{21}(f) &= -\frac{\pi}{2\alpha}\sin\left\{\frac{\pi}{2\alpha}\left(f+\frac{1}{2}-\frac{\alpha}{2}\right)\right\} \\ G_{22}(f) &= -\frac{\pi}{2\alpha}\sin\left\{\frac{\pi}{2\alpha}\left(f-\frac{1}{2}+\frac{\alpha}{2}\right)\right\} \end{aligned} \tag{5.19}$$

As the frequency pulses in (3.1) & (3.2) are defined as being piecewise continuous then so the ML-FED S-curve characteristic is itself piecewise continuous. After some consideration it transpires that these regions are functions of the excess bandwidth, $\alpha$. Consider first the portion of the integral in (5.18) independent of timing phase $\tau$. There are five integrals that are needed to determine the S-curve. These are as follows:

$$\begin{aligned} SC_1(\nu) = & \int_{-f2+\nu}^{-f1} F_1(f)\,F_3(f-\nu)df + \int_{f1}^{f1+\nu} F_2(f)df + \int_{f1+\nu}^{f2} F_2(f)\,F_4(f-\nu)df \\ SC_2(\nu) = & \int_{-f2+\nu}^{-f1} F_1(f)\,F_3(f-\nu)df + \int_{f1}^{-f1+\nu} F_2(f)\,F_3(f-\nu)df + \cdots \\ & \cdots + \int_{-f1+\nu}^{f1+\nu} F_2(f)df + \int_{f1+\nu}^{f2} F_2(f)\,F_4(f-\nu)df \\ SC_3(\nu) = & \int_{f1}^{f2} F_2(f)df \\ SC_4(\nu) = & \int_{f1}^{-f1+\nu} F_2(f)\,F_3(f-\nu)df + \int_{-f1+\nu}^{f2} F_2(f)df \\ SC_5(\nu) = & \int_{-f2+\nu}^{f2} F_2(f)\,F_3(f-\nu)df \end{aligned} \tag{5.20}$$

where

$$f1 = \frac{1-\alpha}{2}$$
$$f2 = \frac{1+\alpha}{2} \tag{5.21}$$

and

$$\begin{aligned} F_1(f) &= G_{11}(f)\, G_{21}(f) \\ F_2(f) &= G_{12}(f)\, G_{22}(f) \\ F_3(f) &= G_{11}(f)^2 \\ F_4(f) &= G_{12}(f)^2 \end{aligned} \tag{5.22}$$

The portions of the integral in (5.18) having coefficients dependent upon the timing phase are given as follows:

$$\begin{aligned} SC_\tau(\nu) = & \int_{f1+\nu}^{f2} G_{22}(f)\, G_{12}(f-\nu)\, G_{11}(f-1)\, G_{11}(f-1-\nu)df + \cdots \\ & \cdots + \int_{-f2+\nu}^{-f1} G_{21}(f)\, G_{11}(f-\nu)\, G_{12}(f+1)\, G_{12}(f+1-\nu)df \end{aligned} \tag{5.23}$$

We now have all the equations for our piecewise definition for the S-curve characteristic for the ML-FED. There are two definitions depending on whether $\alpha <> 1/2$ as follows:

$$SC(\nu) = \begin{cases} \left.\begin{array}{ll} \cos(2\pi\tau)\, SC_\tau(\nu) + SC_1(\nu), & \nu < \alpha \\ SC_3(\nu), & \alpha \le \nu < 1-\alpha \\ SC_4(\nu), & 1-\alpha \le \nu < 1 \\ SC_5(\nu), & 1 \le \nu < 1+\alpha \\ 0, & \nu > 1+\alpha \end{array}\right\} \alpha \le 1/2 \\ \left.\begin{array}{ll} \cos(2\pi\tau)\, SC_\tau(\nu) + SC_1(\nu), & \nu < 1-\alpha \\ \cos(2\pi\tau)\, SC_\tau(\nu) + SC_2(\nu), & 1-\alpha \le \nu < \alpha \\ SC_4(\nu), & \alpha \le \nu < 1 \\ SC_5(\nu), & 1 \le \nu < 1+\alpha \\ 0, & \nu > 1+\alpha \end{array}\right\} \alpha > 1/2 \end{cases} \tag{5.24}$$

The analytical results were derived using MAPLE V. Simulations where also carried out to check the analytical results using MATLAB.

Plots showing the dependence of the frequency-error detector characteristic upon timing and frequency error are shown in figures 5.10 to 5.11 for values of excess bandwidth of 0.3, 0.5 and 0.8. In the figures the solid curves represent the analytical results whereas the points represent that values produced by simulation. We see that the S-curve is dependent upon timing error for values of frequency offset $\nu \le \alpha$. It is evident from the plots that the correspondence between the analysis and simulation is close.

In the previous chapter we derived the structure of the ML-FED and it was noted that the output has to be sampled at the optimum sampling points. An optimum sampling clock will not generally be available prior to frequency acquisition as timing error detectors, such as

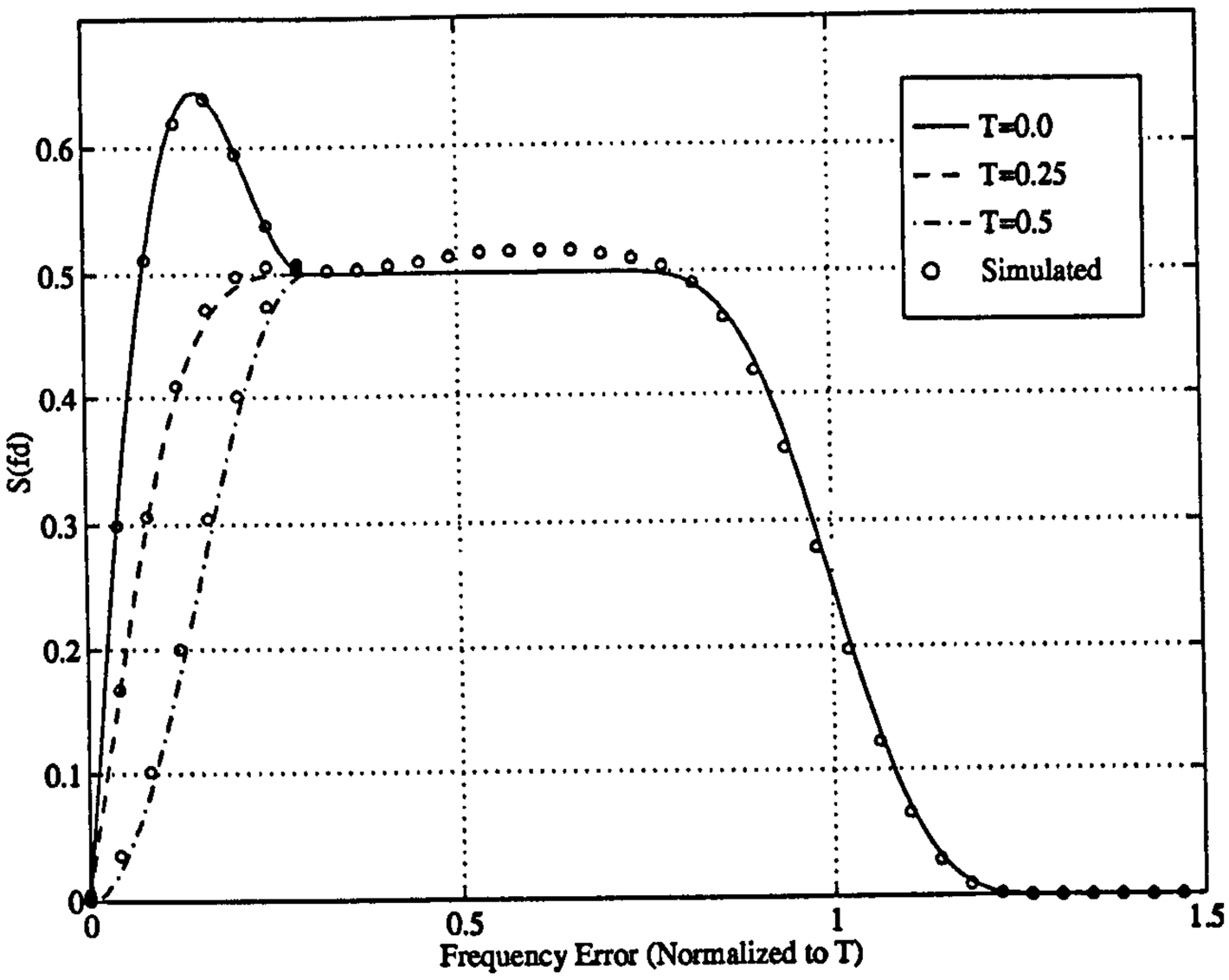



Figure 5.10: ML-FED detector characteristic for non-offset, root raised cosine filtered data with $\alpha = 0.3$. Theory vs simulated results. Only positive values of frequency error are shown.

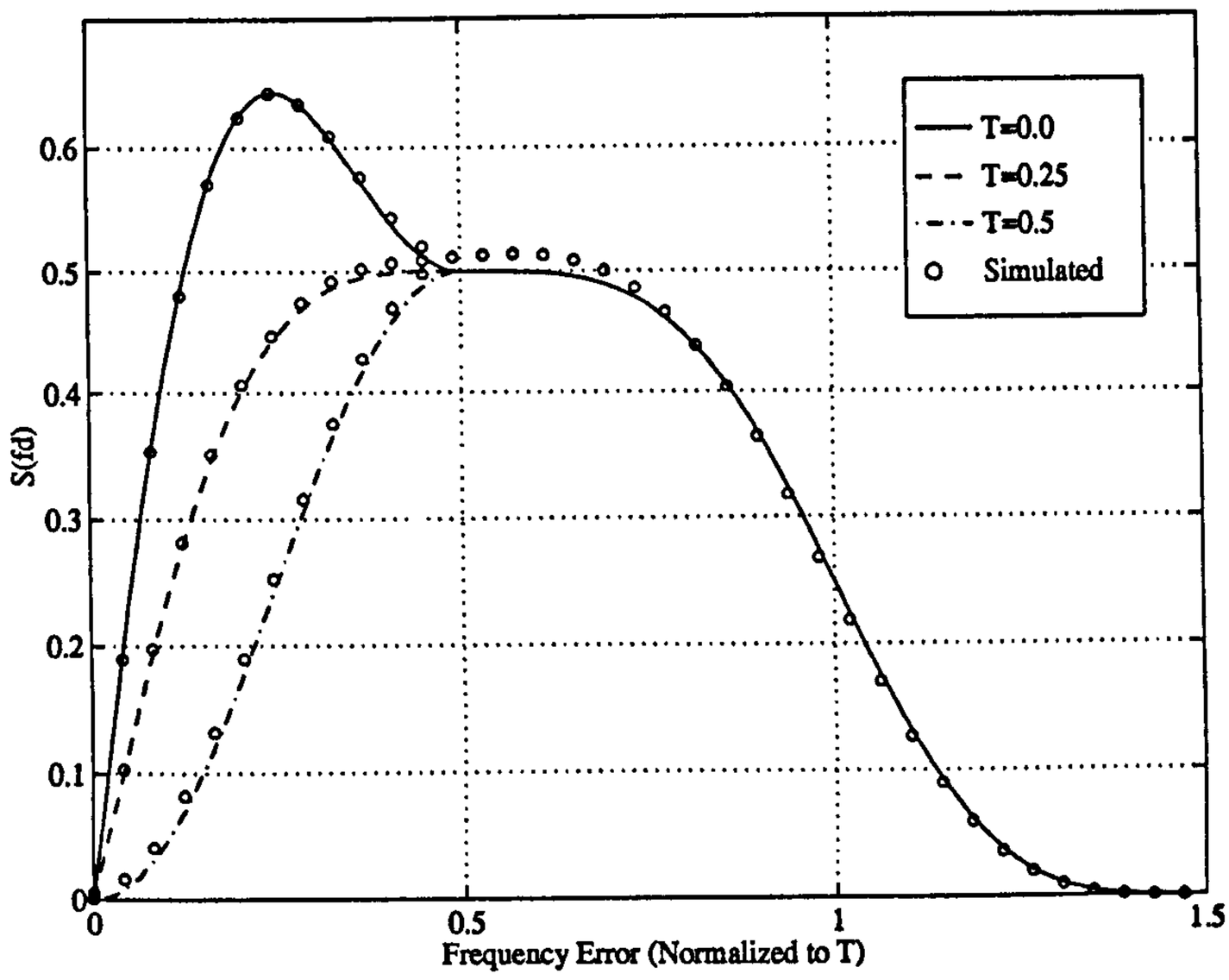



Figure 5.11: ML-FED detector characteristic for non-offset, root raised cosine filtered data with $\alpha = 0.5$. Theory vs simulated results. Only positive values of frequency error are shown.

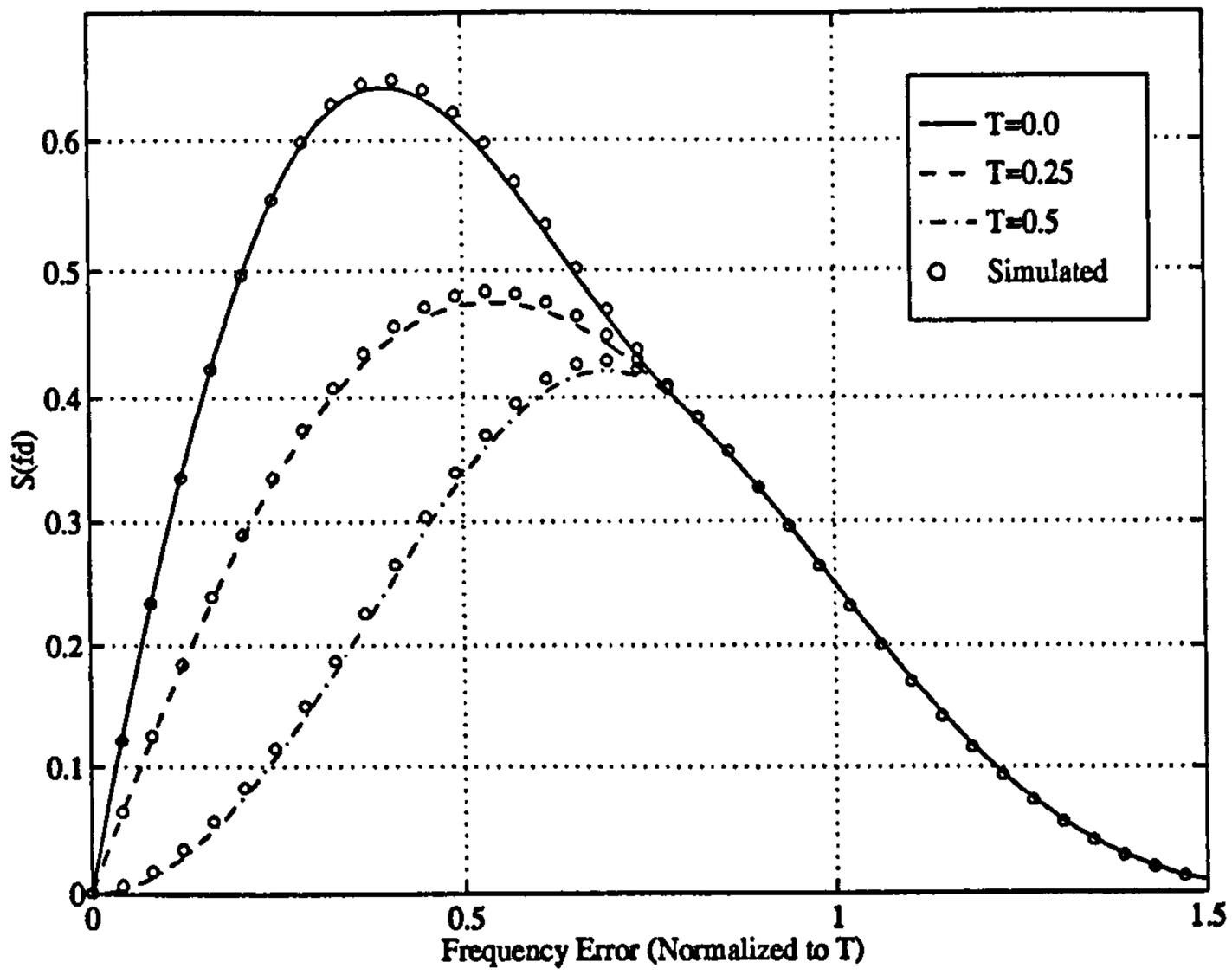



Figure 5.12: ML-FED detector characteristic for non-offset, root raised cosine filtered data with $\alpha = 0.8$. Theory vs simulated results. Only positive values of frequency error are shown.

the two-point non-data aided Gardner algorithm, may perform badly with a frequency offset and will generally rely on the received signal being in the centre of the receiver matched-filter. Frequency acquisition is the first function undertaken by the receiver. Thus, the important feature to note about the ML-FED is that the gain of the detector at $\nu \approx 0$ is a function of the timing-phase. This is not the case in the BQ-FED where all the data samples are used to generate the error signal.

The gain of the ML-FED transfer characteristic at $\nu \approx 0$ is also important as it determines the effective loop bandwidth during tracking. Examination of the above figures indicate that this gain is dependent upon both the excess bandwidth and the timing phase. We can determine what this is by differentiating the ML-FED characteristic given by (5.24) to give

$$A_{ML} = \frac{\delta}{\delta\nu}\, SC(\nu)|_{\nu=0} = \frac{\delta}{\delta\nu}\, \{\cos(2\pi\tau)\, SC_\tau(\nu) + SC_1(\nu)\}|_{\nu=0} = \frac{\pi^2}{8\alpha}(\cos(2\pi\tau) + 1) \tag{5.25}$$

Note from this equation that as the excess bandwidth of the modulation scheme becomes smaller the ML-FED gain will become larger. Maximum gain will be achieved for a given value of excess bandwidth when the timing phase is zero, as expected.

The ML-FED detector characteristic as shown in figures 5.10 to 5.12 are representative of the *mean* value of the error signal. We are also interested in the *variance* of the error signal as a function of frequency offset and timing phase error. Such plots have been obtained by simulation and are shown in figures 5.13 to 5.15. Note that the ML-FED has the desirable property of having zero variance at $\nu = 0$ but this depends on optimum timing phase also. This implies that during tracking the ML-FLL will have a small contribution due to self-noise,

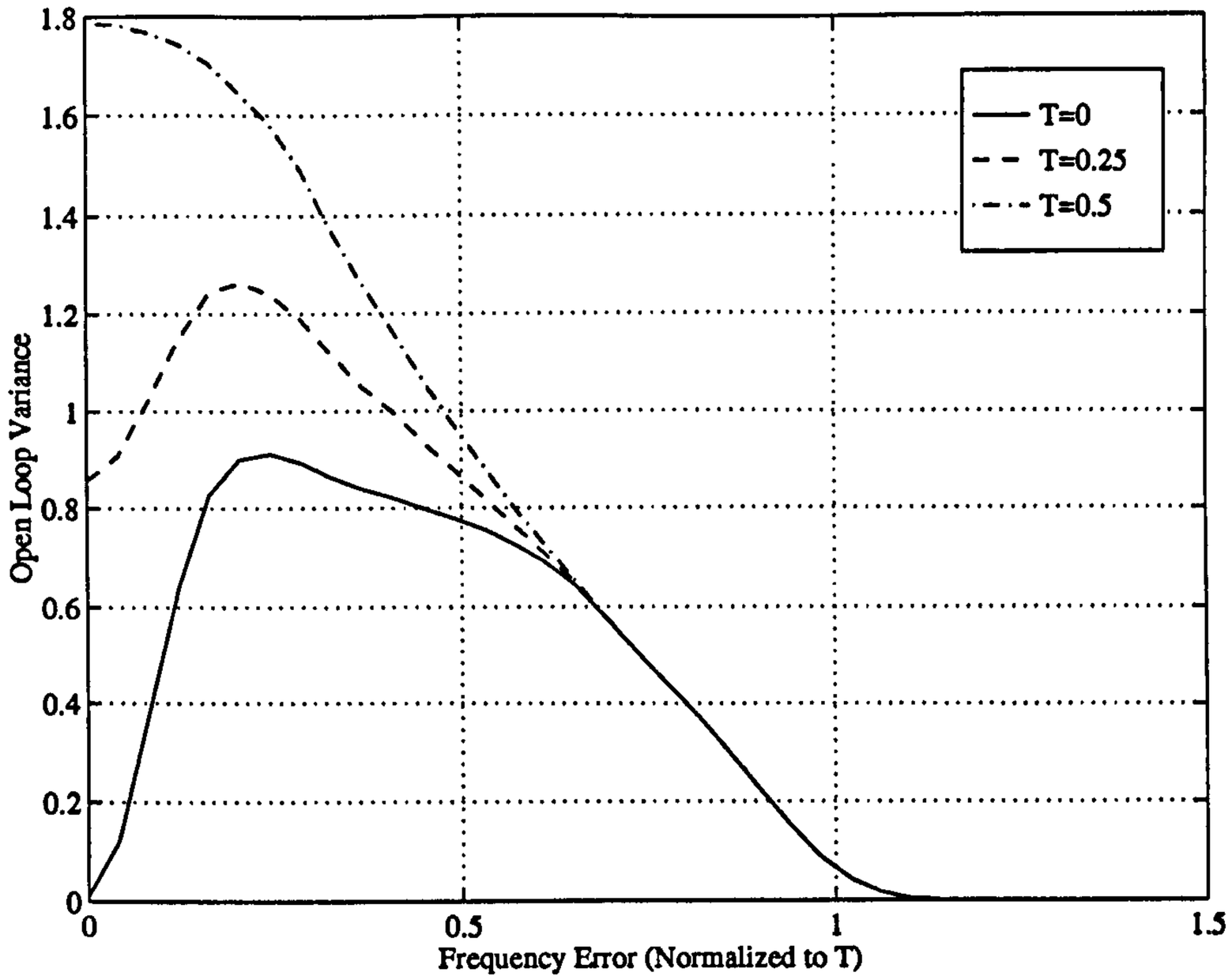



Figure 5.13: ML-FED detector variance for non-offset, root raised cosine filtered data with $\alpha = 0.3$. Only positive values of frequency error are shown.

with the important caveat that optimum timing-phase is available. Also, if the timing-phase is not optimum, then the amount of self-noise is dependent on the excess bandwidth.

The open-loop results for the mean and variance of the ML-FED were obtained by using 31 taps for the FIR filters and a rectangular window in the filter design. We shall come on to discuss more the design method and the optimum number of taps for the ML-FED later.

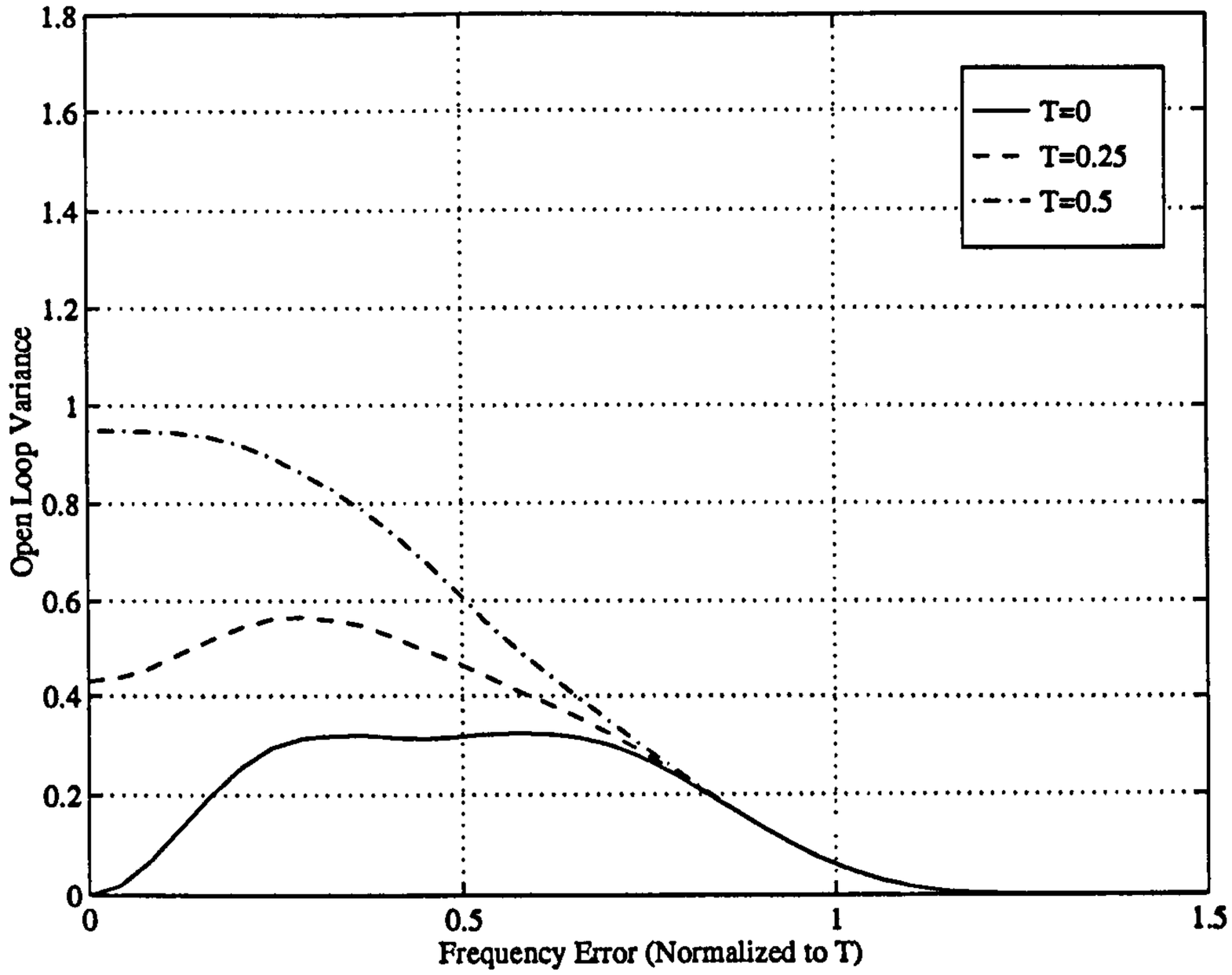



Figure 5.14: ML-FED detector variance for non-offset, root raised cosine filtered data with $\alpha = 0.5$. Only positive values of frequency error are shown.

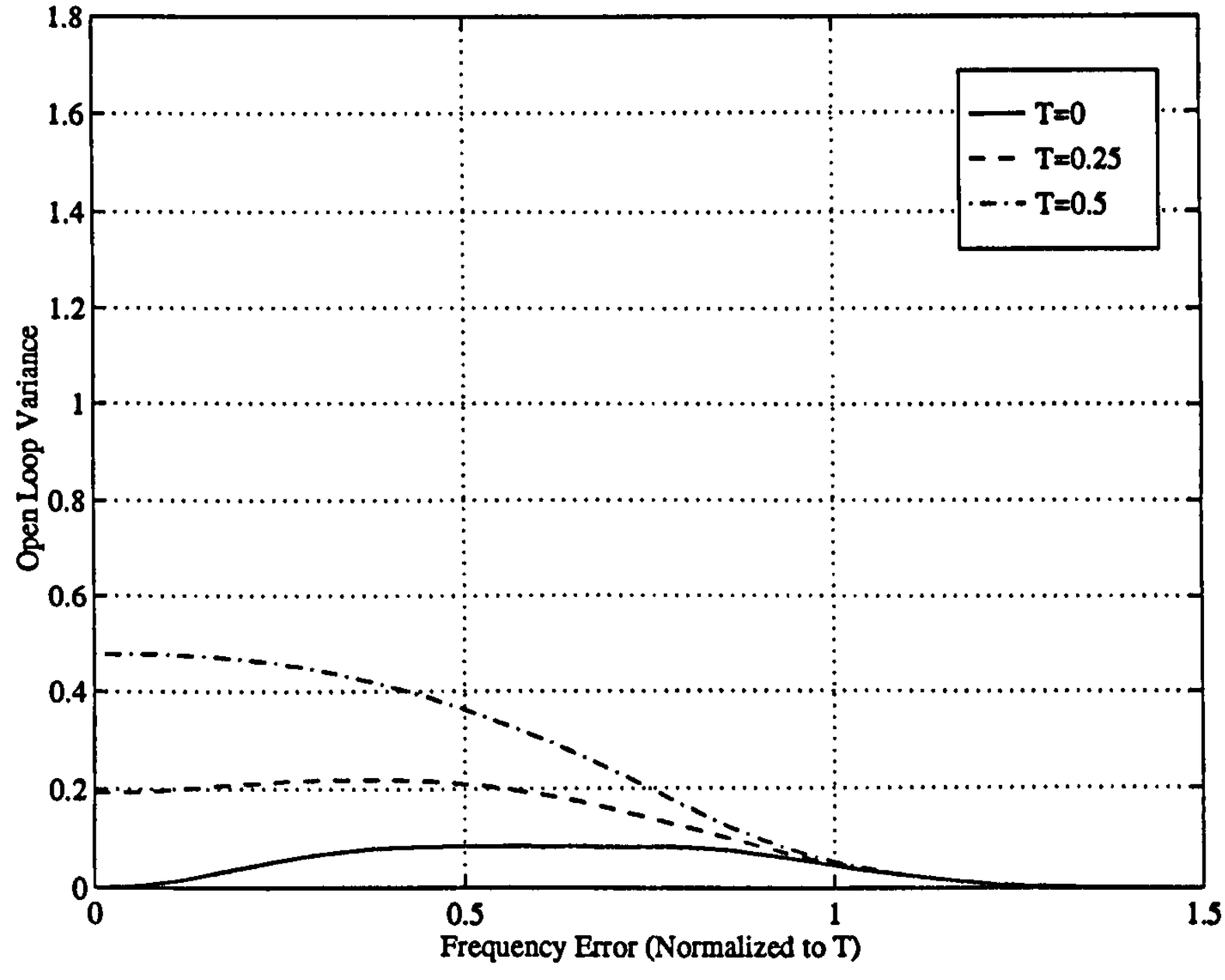



Figure 5.15: ML-FED detector variance for non-offset, root raised cosine filtered data with $\alpha = 0.8$. Only positive values of frequency error are shown.

### 5.7.2 BQ-FED Characterisation

The structure of the BQ-FED was discussed in the previous section. We consider a general structure for the BQ-FED as proposed by D'Andrea in [20] where the $H(f)$ and the $D(f)$ are a Butterworth bandpass filter and a $T/2$ delay respectively. The Butterworth filter is second order with a 3dB bandwidth of 0.2/T and centre frequency of 0.6/T. Whereas we have seen that the ML-FED has filters $H(f)$ and $H'(f)$ which are matched to the transmitted pulse this is not the case with the BQ-FED. Here, it is intended to arrive at a structure which can be used for a number of different pulse shapes and will still give reasonable performance.

The detector characteristic for the BQ-FED can be written as [20],

$$SC(\nu) = \int_{-\infty}^{+\infty} |G(f)|^2 P(f+\nu)\, df \tag{5.26}$$

where $G(f)$ is the baseband pulse shaping (in the following root raised cosine) and $P(f)$ is

$$P(f) = |H(f)|^2 Im(D(f)) \tag{5.27}$$

The above integral can be understood to be the baseband spectrum weighted by an odd-symmetric function $P(f+\nu)$. In (5.27) $Im(D(f))$ is the imaginary component of a $T/2$ delay which in complex notation is simply $\sin(\pi f)$. The function $P(f)$ for the specified values of $H(f)$ and $D(f)$ is as plotted in figure 5.16. One can see how the S-curve detector characteristic arises by examination of such a function.

As with the ML-FED, the gain of the detector is important in determining the closed-loop noise equivalent bandwidth. We thus need the derivative of (5.26) at $\nu = 0$. This is,

$$A_{BQ} = \int_{-\infty}^{+\infty} |G(f)|^2 \dot{P}(f)\, df \tag{5.28}$$

As with the ML-FED detector characteristic we use MAPLE V as an aid to determine the detector characteristic. As with the MLE-FED an easy way to determine the open-loop S-curve characteristic is to use MATLAB. This then allows us to determine if the analytical results are correct before implementation in SPW. The MATLAB script for determining the open-loop detector characteristics is included in Appendix B. Unlike the ML-FED, no closed-form analytical results are available for the detector characteristic and its derivative. The integrals in (5.26) and (5.28) have been evaluated numerically using Newton-Cotes integration [60].

Comparisons between the numerical integrations on MAPLE and the simulations in MATLAB for the mean value of the error signal for excess bandwidths of 0.3, 0.5 and 0.8 are included in figures 5.17 to 5.19. The correspondence between the analysis and the simulation is close.

The gain of the BQ-FED has been determined as follows as a function of the pulse excess bandwidth as,

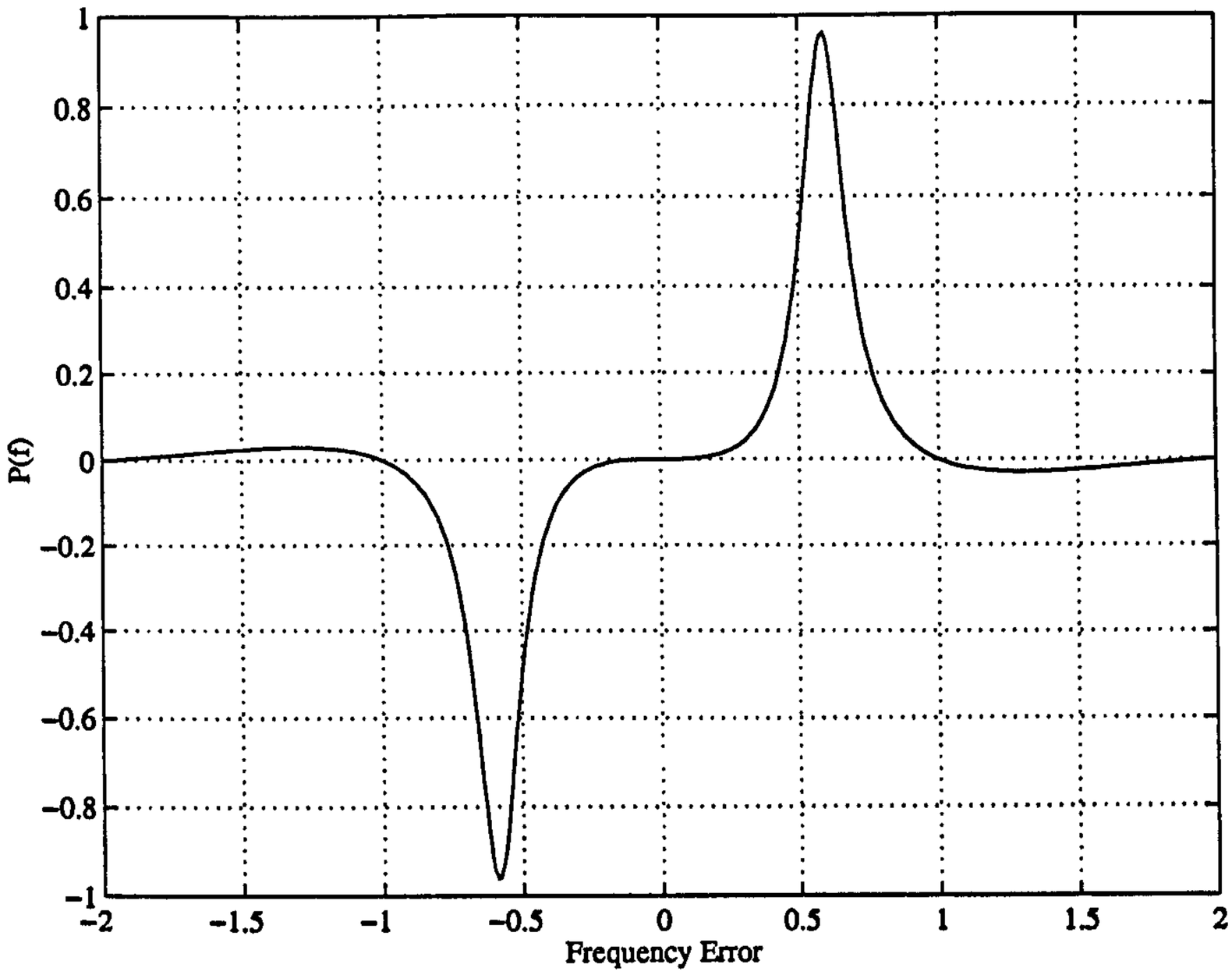



Figure 5.16: Function $P(f)$ for $H(f)$ a 2-pole Butterworth filter with centre $0.6/T$ and bandwidth $0.2/T$ and $D(f)$ are $T/2$ delays.

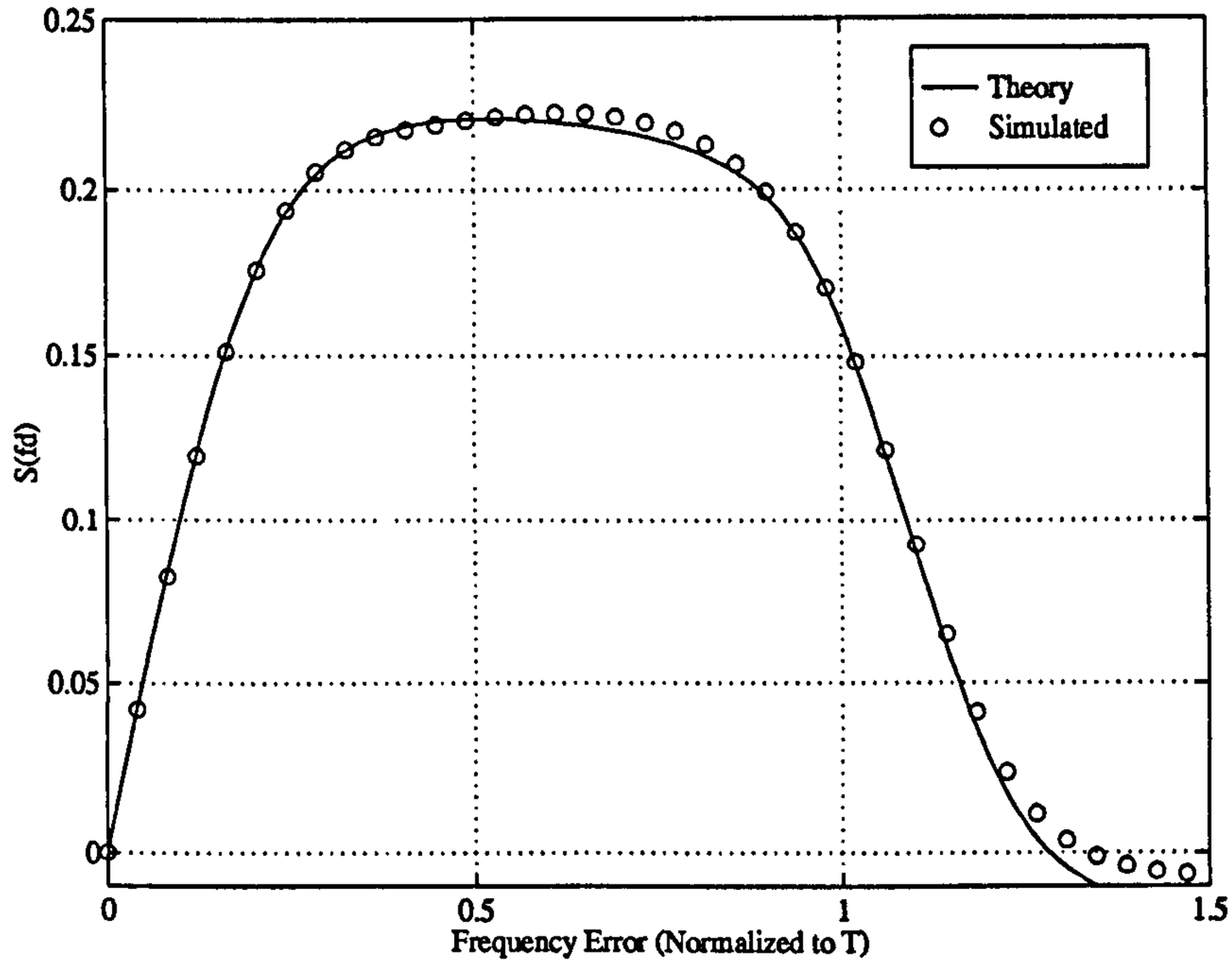



Figure 5.17: BQ-FED detector characteristic for non-offset, root raised cosine filtered data with $\alpha = 0.3$. Theory vs simulated results. Only positive values of frequency error are shown.

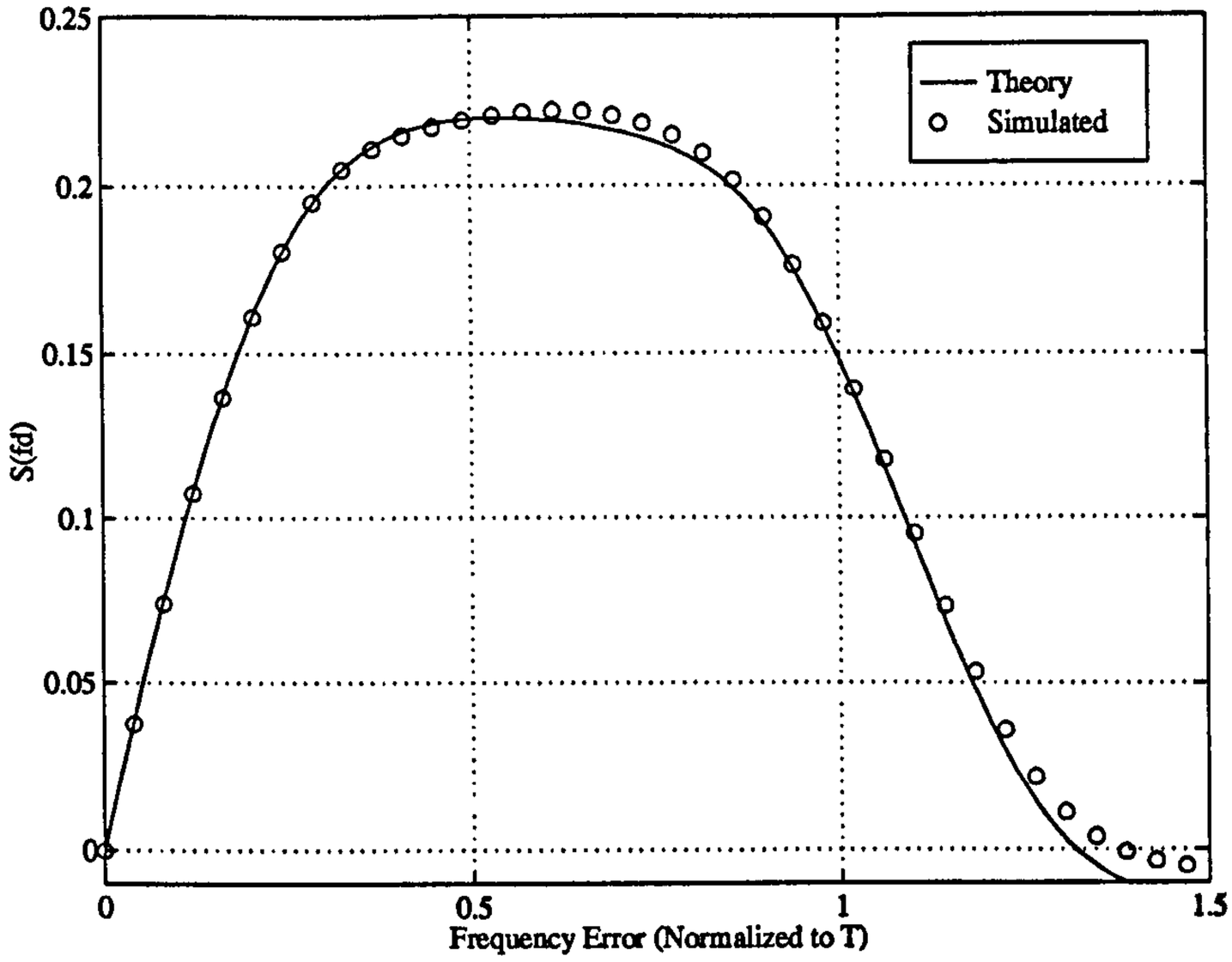



Figure 5.18: BQ-FED detector characteristic for non-offset, root raised cosine filtered data with $\alpha = 0.5$. Theory vs simulated results. Only positive values of frequency error are shown.

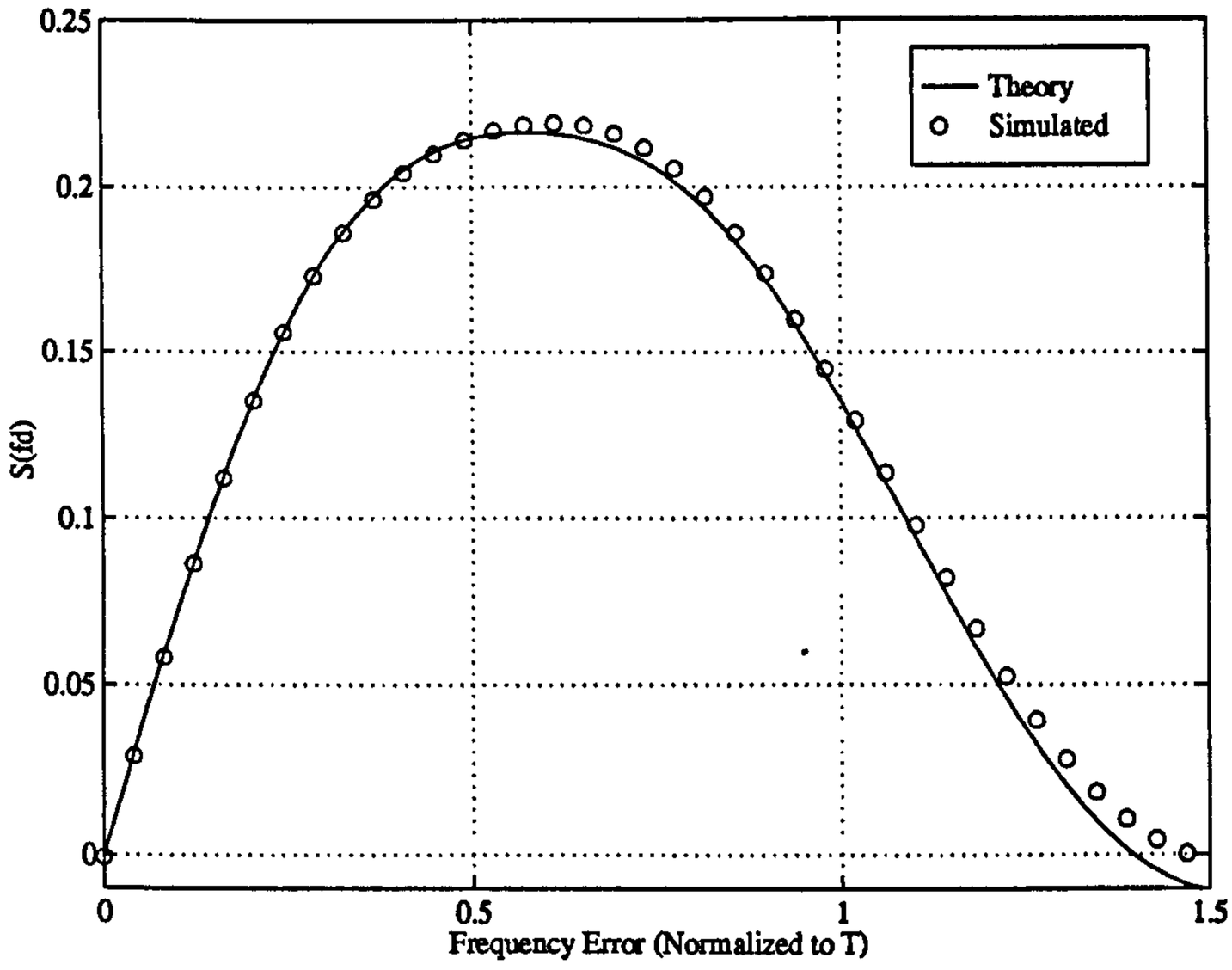



Figure 5.19: BQ-FED detector characteristic for non-offset, root raised cosine filtered data with $\alpha = 0.8$. Theory vs simulated results. Only positive values of frequency error are shown.

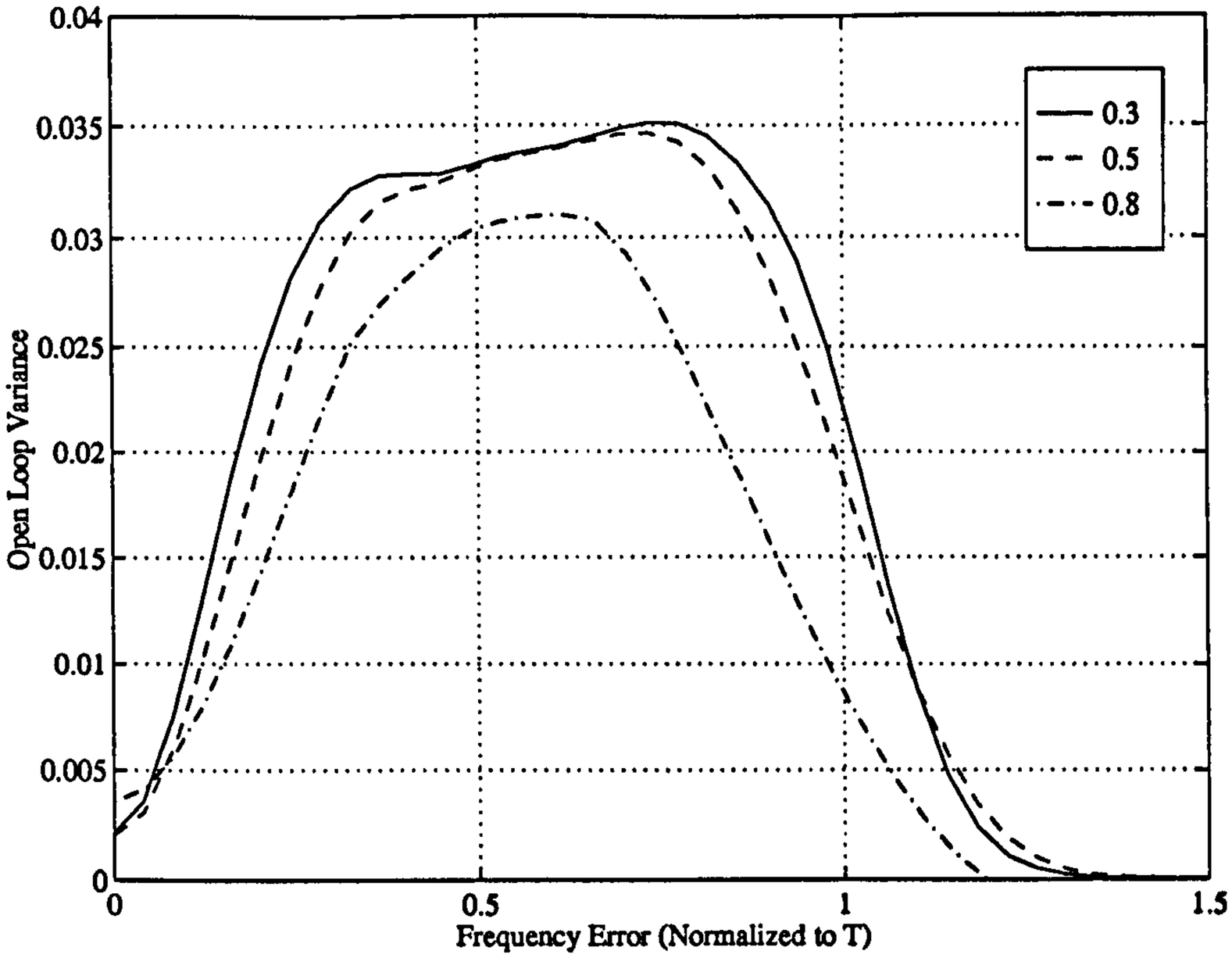


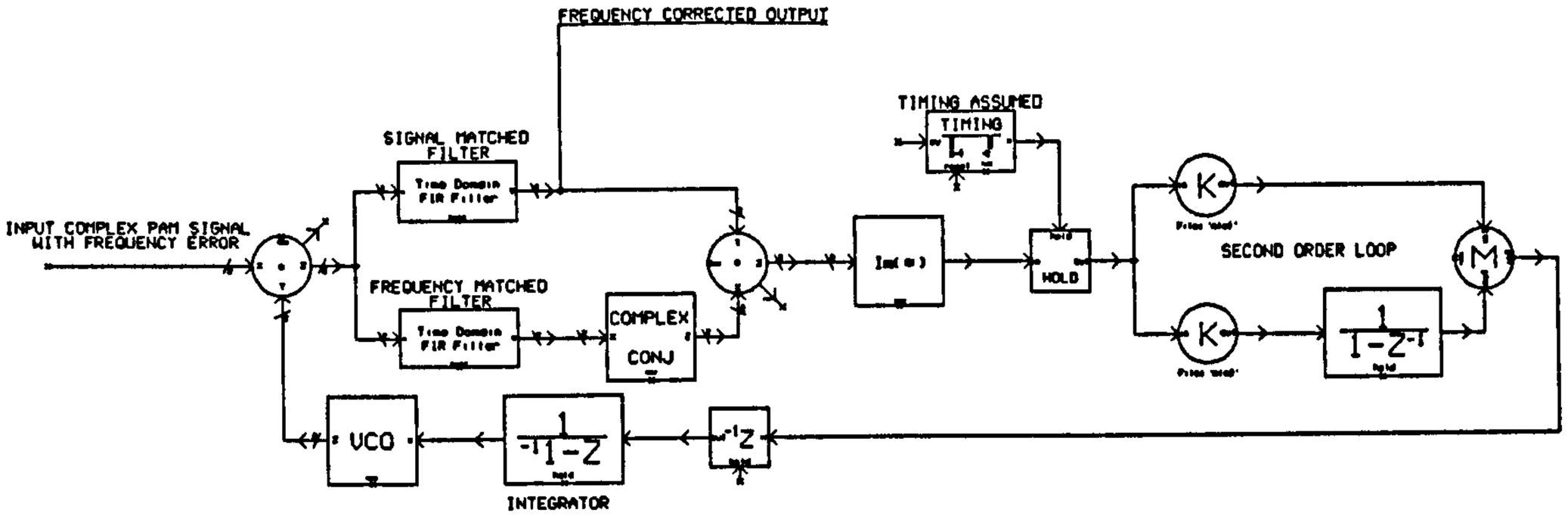

Figure 5.20: BQ-FED detector variance for non-offset, root raised cosine filtered data with $\alpha = 0.3, 0.5, 0.8$. Only positive values of frequency error are shown.

| $\alpha$ | $A_{BQ}$ |
|---|---|
| 0.1 | 1.022597 |
| 0.2 | 1.061600 |
| 0.3 | 1.062845 |
| 0.4 | 1.017011 |
| 0.5 | 0.947377 |
| 0.6 | 0.872343 |
| 0.7 | 0.800519 |
| 0.8 | 0.735010 |
| 0.9 | 0.676466 |
| 1.0 | 0.624554 |

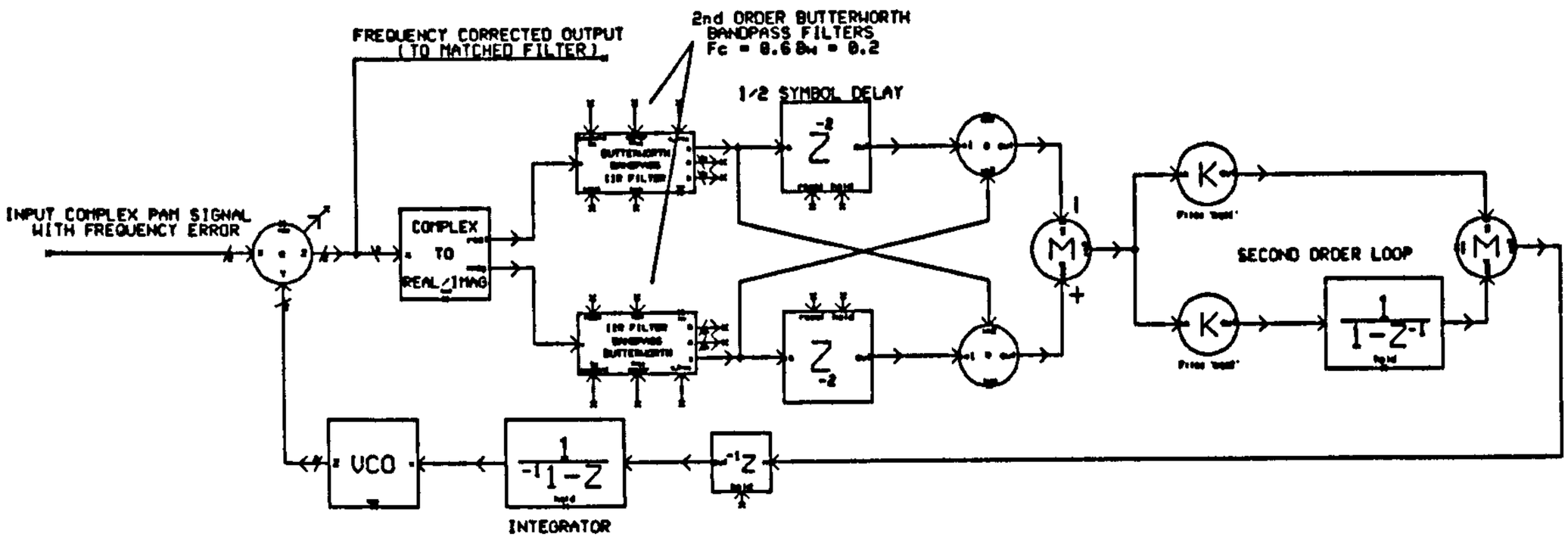
Again, also of interest is the variance of the error signal as a function of frequency error. This is shown in figure 5.20. To compare the performance of the ML-FED and BQ-FED we need to look at their closed-loop tracking performance. We know the gains of the two detectors, $A_{ML}$ and $A_{BQ}$, and so we can normalise the two loops to have the same noise equivalent bandwidth, $B_{EQ}$. We shall discuss this more in the next section.

## 5.8 Closed-Loop Performance

The SPW system for comparing both the acquisition and tracking performance of the ML-FLL and BQ-FLL is shown in figure 5.21. Notice that in the figure there are a number of




(a) Maximum-Likelihood FED




(b) Balanced-Quadricorrelator FED

Figure 5.21: SPW implementations of the Balanced-Quadricorrelator & Maximum-Likelihood FEDs

blocks that allow parameters to be read in at the start of the simulation and other blocks which allow values to be written out at the end of the simulation. It was found that this is the most convenient method of iterating over parameters and the details of this new methodology is discussed in appendix A. SPW is used essentially to develop the simulation code and then the code can be exercised with different sets of parameters by just changing the values in a file. Typically then we have a main program generated by the SPW Code Generation System and a control program written by the user to change the parameters in a file and re-run the simulation.

### 5.8.1 Noise Equivalent Bandwidths for Frequency Locked Loops

Here we give an analysis of the equivalent noise bandwidth for generalised first and second order feedback tracking structures in terms of the loop filter coefficients. The objective is to

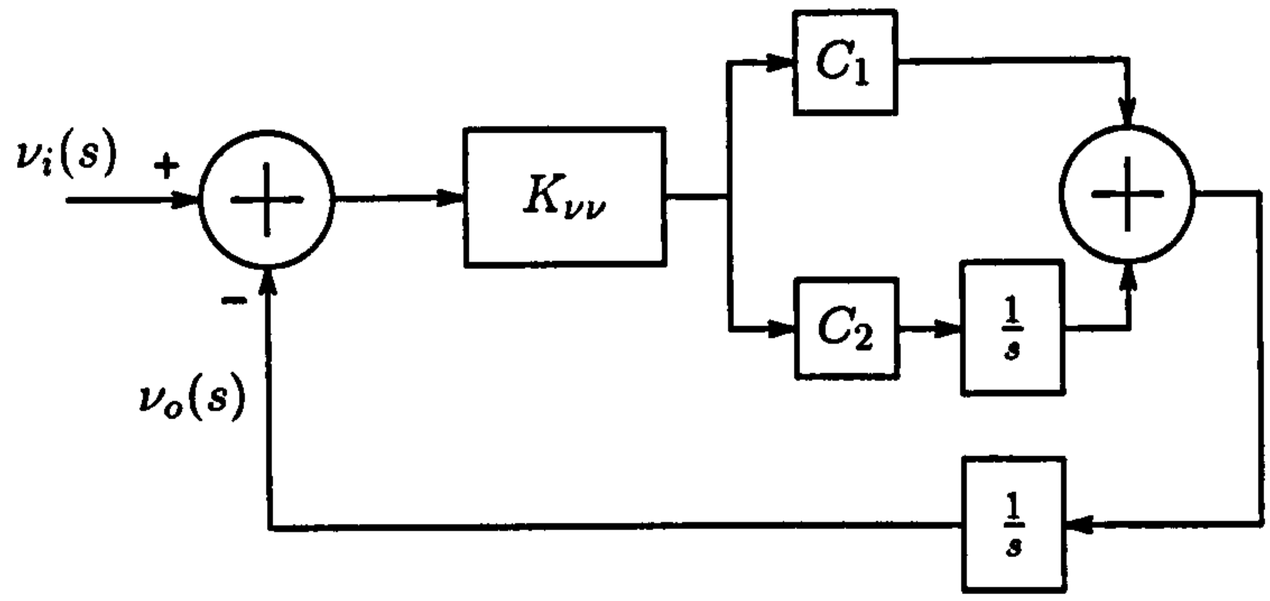



Figure 5.22: General Error-Tracking Structure.

set the loop bandwidth, as the independent variable, and this will then automatically set the loop filter coefficients. This effort is worthwhile in that any general tracking structure, be it a frequency, phase or symbol-timing recovery loop will behave, as a first approximation, according to standard Phase Locked Loop theory.

Consider the general tracking structure shown in figure 5.22. The notation in the figure is specific to frequency locked loops but the general theory is applicable to any generic tracking structure of this form.

This general model incorporates a first order loop (if $C_2 = 0$) and a second-order loop with the standard "proportional-plus-integral" loop filter. The parameter $K_{\nu\nu}$ is the gain of the detector, here denoted as frequency but could equally be timing or carrier phase. This is then denoted as,

$$K_{\nu\nu} = \frac{\delta}{\delta\nu} S(\nu)\Big|_{\nu=0} \tag{5.29}$$

or, in other words, the slope of the detector S-curve at the origin. This model is referred to as the linearised, or small-signal, PLL model and is used often in the literature for analysis purposes as the nonlinear model is invariably intractable. Often the model does not work well for large signals, due perhaps to large amounts of additive noise, or during signal acquisition. It should also be borne in mind that the slope $K_{\nu\nu}$ will in general be a function of a number of system parameters. We saw for the case of frequency detection that $K_{\nu\nu}$ will be a function of the modulation excess bandwidth as well as the timing phase. For decision-directed parameter estimation $K$ will also depend on the signal to noise ratio. This presents an interesting dilemma in terms of what the appropriate definition should be for the loop noise bandwidth in decision-directed parameter estimation. Does one use the value of $K$ assuming no noise in the system or does one assume *a priori* knowledge of the signal to noise and then calculate $K$ accordingly? We think that this latter approach is not feasible and so the detector slope is that at high signal to noise unless stated otherwise.

An interesting example is in symbol timing recovery using a PLL in a flat fading channel. As the signal goes into a fade, or alternatively, the signal to noise ratio decreases, the loop gain gets smaller and so the loop bandwidth will decrease accordingly. Then the symbol timing

recovery loop will tend to "free-wheel" and will not lose synchronisation. As the signal comes out of the fade the loop bandwidth increases and the symbol-timing recovery loop can then make any small adjustments that it has to. The conclusion of this is that the variability in the instantaneous noise bandwidth for symbol timing recovery can actually be beneficial and that it is not advisable to attempt to Automatic Gain Control prior to the symbol timing recovery loop in a flat fading channel.

To derive the appropriate settings for the loop filter coefficients $C_1$ and $C_2$ in terms of the single sided equivalent noise bandwidth $B_L$ refer to figure 5.22. We can write the transfer characteristic of such a loop using standard PLL terminology as

$$H(s) = \frac{K_{vv}\, F(s)/s}{1 + K_{vv}\, F(s)/s} \tag{5.30}$$

where $F(s)$ is the transfer function of the loop filter and is given as

$$F(s) = C_1 \tag{5.31}$$

for a first order loop, and,

$$F(s) = C_1 + \frac{C_2}{s} \tag{5.32}$$

for a second order loop. The single-sided noise bandwidth is defined as

$$B_L = \frac{1}{2\pi} \int_0^\infty |H(j\,\omega)|^2\, d\omega \tag{5.33}$$

For the first order loop this equation can be evaluated to give

$$B_L = \frac{K_{vv}\, C_1}{4} \tag{5.34}$$

So, for a first order loop we set the coefficients as

$$\boxed{C_1 = \frac{4\, B_L}{K_{vv}} \quad \text{and} \quad C_2 = 0} \tag{5.35}$$

The derivation for a second order loop is somewhat more involved. The transfer function can now be written in terms of the loop filter coefficients as

$$H(s) = \frac{K_{vv}\, C_1 s + K_{vv}\, C_2}{s^2 + K_{vv}\, C_1 s + K_{vv}\, C_2} \tag{5.36}$$

However, from standard PLL theory it is more usual to express the transfer characteristic in terms of the damping factor, $\gamma$, and the natural frequency $\omega_n$ as

$$H(s) = \frac{2\gamma\omega_n\, s + \omega_n^2}{s^2 + 2\gamma\omega_n\, s + \omega_n^2} \tag{5.37}$$

with,

$$C_1 = \frac{2\gamma\omega_n}{K_{vv}} \quad \text{and} \quad C_2 = \frac{\omega_n^2}{K_{vv}} \tag{5.38}$$

The integral for the loop noise bandwidth for the second order loop can be evaluated to give

$$B_L = \frac{\omega_n}{2}\left(\gamma + \frac{1}{4\gamma}\right) \tag{5.39}$$

We can therefore rearrange these equations to give the loop coefficients in terms of the damping factor and the loop noise bandwidth as

$$\boxed{C_1 = \frac{16\, B_L \gamma^2}{(4\gamma^2+1)\, K_{\nu\nu}} \quad \text{and} \quad C_2 = \frac{64\, B_L^2 \gamma^2}{(4\gamma^2+1)^2\, K_{\nu\nu}}} \tag{5.40}$$

When $\gamma = 1$, the loop is said to be *critically damped*; if $\gamma > 1$, it is *over-damped*; and if $\gamma < 1$, it is *underdamped*. A good compromise between stability and speed of response for a second-order system is to put $\gamma = 1/2$. Alternatively, putting $\gamma = 1/\sqrt{2}$ ensures a loop transfer function the same as a first order Butterworth filter. Thus with damping factors of $\gamma = 1/\sqrt{2}, 1/2, 1$ the loop filter coefficients take the following values,

- $\underline{\gamma = 1/\sqrt{2}}$

$$C_1 = \frac{8}{3}\frac{B_L}{K_{\nu\nu}} \quad \text{and} \quad C_2 = \frac{32}{9}\frac{B_L^2}{K_{\nu\nu}} \tag{5.41}$$

- $\underline{\gamma = 1/2}$

$$C_1 = 2\frac{B_L}{K_{\nu\nu}} \quad \text{and} \quad C_2 = 4\frac{B_L^2}{K_{\nu\nu}} \tag{5.42}$$

- $\underline{\gamma = 1}$

$$C_1 = \frac{16}{5}\frac{B_L}{K_{\nu\nu}} \quad \text{and} \quad C_2 = \frac{64}{25}\frac{B_L^2}{K_{\nu\nu}} \tag{5.43}$$

### 5.8.2 Acquisition Performance of Error-Tracking Loops in Terms of the Loop Filter Coefficients

In the previous section we determined how the loop coefficients are related to the single-sided loop noise bandwidth for first and second order tracking loops. For the second order loops we considered appropriate loop filter coefficients for damping factors of $\gamma = 1/\sqrt{2}, 1/2, 1$. In this section we shall determine the small-signal acquisition performance of the generalised tracking structure as shown in figure 5.22 for first and second-order tracking loops. These analytical results are then applicable to acquisition of phase, frequency and symbol timing as we shall later demonstrate.

The tracking error, $\psi(t)$, of the loops for various input phase functions, $\phi(t)$, can be obtained by taking the inverse Laplace transform of $\psi(s)$ given in terms of the loop transfer function as,

$$\psi(s) = (1 - H(s))\,\phi(s) \tag{5.44}$$

In the above $\phi(s)$ is the laplace transform of the input function $\phi(t)$ which could be a static phase offset, a phase ramp corresponding to a constant frequency offset, or a quadratic rate

of change of phase corresponding to a frequency chirp. We are only really interested in the acquisition of the error-tracking loops when presented with a static phase offset or a constant frequency offset. In the above equations $1 - H(s)$ is referred to as the phase-error transfer function.

The input functions of a static phase offset and a constant frequency offset are then described mathematically as,

- A Phase Step

$$\phi(t) = \Delta\phi\, u(t) \quad \xrightarrow{\mathcal{L}} \quad \phi(s) = \frac{\Delta\phi}{s} \tag{5.45}$$

- A Constant Frequency Offset

$$\phi(t) = \Delta\omega t\, u(t) \quad \xrightarrow{\mathcal{L}} \quad \phi(s) = \frac{\Delta\phi}{s^2} \tag{5.46}$$

Determination of the closed-form analytical results for the acquisition performance of the tracking loops is accomplished by solving for the inverse-Laplace transform of $\psi(s)$ as,

$$\psi(t) = \mathcal{L}^{-1}\{\psi(s)\} = \mathcal{L}^{-1}\{(1 - H(s))\,\phi(s)\} \tag{5.47}$$

This has been done using MAPLE V together with the `laplace` library to give the following results,

1. First Order Loop

   - A Phase Step

$$\phi(t) = e^{-4\,B_L t} \tag{5.48}$$

   - A Constant Frequency Offset

$$\phi(t) = \frac{1}{4\,t} - \frac{e^{-4\,B_L t}}{4\,t} \tag{5.49}$$

2. Second Order Loop

   - A Phase Step
     - $\gamma = 1/\sqrt{2}$

$$\phi(t) = \left(\cos\left(\frac{4\,B_L t}{3}\right) - \sin\left(\frac{4\,B_L t}{3}\right)\right) e^{-4\,B_L t/3} \tag{5.50}$$

     - $\gamma = 1/2$

$$\phi(t) = \left(\cos(\sqrt{3}B_L t) - \frac{\sqrt{3}\sin(\sqrt{3}B_L t)}{3}\right) e^{-B_L t} \tag{5.51}$$

     - $\gamma = 1$

$$\phi(t) = \left(1 - \frac{8\,B_L t}{5}\right) e^{-8\,B_L t/5} \tag{5.52}$$

- A Constant Frequency Offset
  - $\gamma = 1/\sqrt{2}$

$$\phi(t) = \frac{3\,\sin(\frac{4\,B_L t}{3})}{4\,B_L} e^{-4\,B_L t/3} \tag{5.53}$$

  - $\gamma = 1/2$

$$\phi(t) = \frac{\sqrt{3}\sin(\sqrt{3}B_L t)}{3\,B_L} e^{-B_L t} \tag{5.54}$$

  - $\gamma = 1$

$$\phi(t) = t\,e^{-8\,B_L t/5} \tag{5.55}$$

Note in the above equations that the responses for critical damping, $\gamma = 1$, contain only exponential terms.

### 5.8.3 Design of Frequency and Signal Matched Filters for the ML-FED

We have seen that for the implementation of the ML-FED we need to implement a signal matched filter, with impulse response $h(-t)$, where $h(t)$ is the shape of the transmitted pulse, and a frequency matched filter which is the frequency derivative of this. Specifically, if $h(-t)$ is the impulse response of the signal matched filter then it has a corresponding transfer function given as,

$$H^*(f) = \int_{-\infty}^{+\infty} h(-t)\exp(-j\,2\pi f t)\,dt \tag{5.56}$$

To obtain the frequency matched filter we differentiate this to give,

$$\frac{\delta H^*(f)}{\delta f} = \int_{-\infty}^{+\infty} h(-t)\,(-j\,2\pi f t)\exp(-j\,2\pi f t)\,dt \tag{5.57}$$

$$= -\,j\,2\pi \int_{-\infty}^{+\infty} t\,h(-t)\exp(-j\,2\pi f t)\,dt \tag{5.58}$$

Thus, the impulse response corresponding to the frequency matched filter transfer function $\delta H^*(f)/\delta f$ is $-j\,2\pi t h(-t)$. In the frequency domain, the frequency matched filter has a transfer function that is the derivative of the transfer function of the signal matched filter. In the time domain, the frequency matched filter has an impulse response that is the same as that of the signal matched filter, except weighted by a factor $-j\,2\pi t$.

We have used the MATLAB script in appendix B to design the signal and frequency matched filters using the least-squares FIR filter design method. The coefficients are saved in a form suitable for transfer to SPW. The signal matched filter for the following simulation results is specified as a root raised cosine filter, the excess bandwidth and the number of taps required are input by the user, and then the impulse response for the frequency matched filter is generated directly from the signal matched filter. There is a design choice here, whether to simply generate the frequency matched filter directly from the signal matched filter (as described above) or to design the two filters independently.

D'Andrea in [21] uses 17 taps for the design of the filters when the modulation excess bandwidth is $\alpha = 0.5$ and the loop bandwidth is $B_L = 0.005$. In general the number of taps to choose for the filter design will depend on these two parameters. If the excess banwidth, $\alpha$, is small then a large number of taps will have to be chosen so that the filter responses do not deviate too far from the theoretical ideal. On the other hand, choosing a large number of taps for the filter design will introduce a large delay in the loop which could, depending upon the setting of the loop filter bandwidth, make the loop unstable. What still needs to be done here is a loop stability analysis using the root-locus techniques from control theory to determine stable configurations for the loop coefficients $C_1$, $C_2$ the detector gain $K_{\nu\nu}$ and the number of filter taps, $N$.

Figure 5.23 shows the frequency and time responses of the signal and frequency matched filters for root raised cosine filter with excess bandwidth $\alpha = 0.2$ and $N = 41$ taps. Figure 5.24 show similar results but for $\alpha = 0.8$ and $N = 17$ taps.

Before going on to look at the acquisition and tracking performance of the ML & BQ-FEDs we need to determine the minimum number of taps for the signal and frequency matched filters. As mentioned above, D'Andrea in [21] suggests using $N = 17$ taps at an excess bandwidth of $\alpha = 0.5$ but his results assume perfect timing, that is a "cheat-wire", in the simulation, and also he has not given any other figures for large/small values of excess bandwidth. In our simulations to determine the required number of taps for the ML-FED filters the timing was estimated simultaneously with the frequency using a method to be described shortly. Obviously, the bandwidth of the timing recovery loop was set to be very much smaller than the frequency recovery loop.

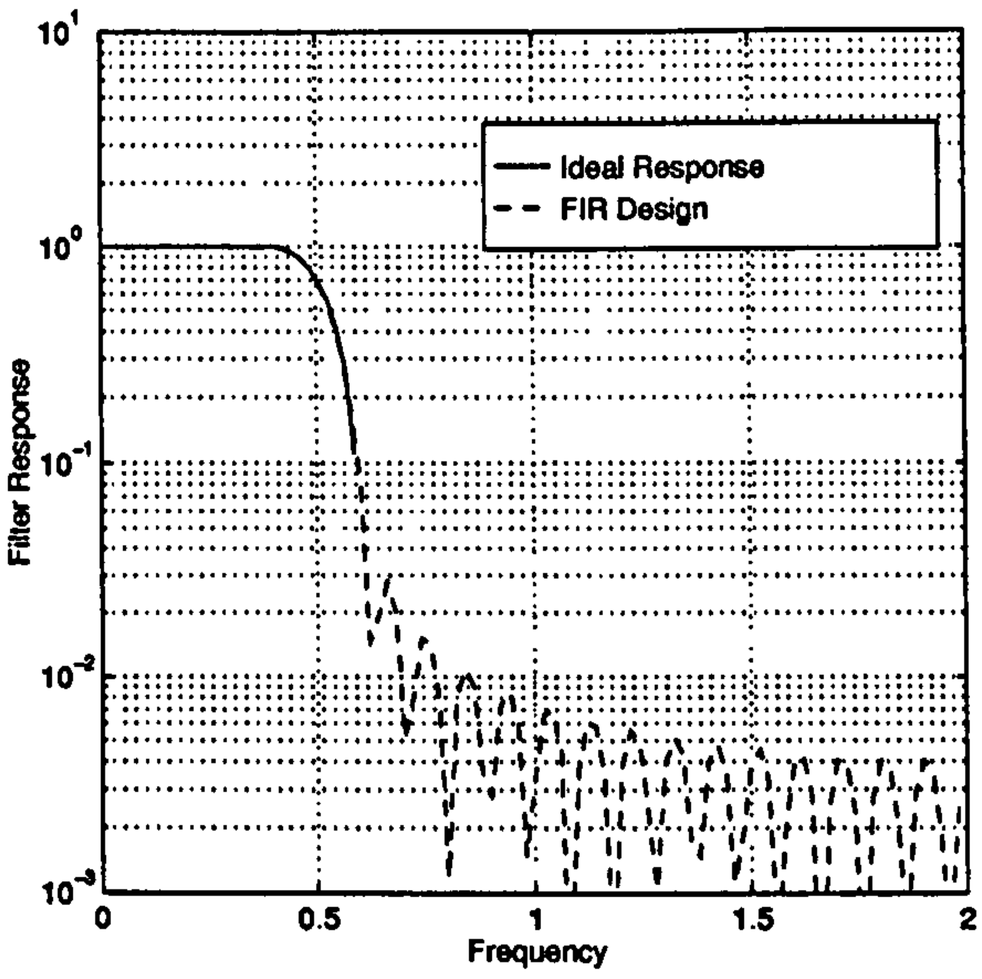
We ran simulations to measure the frequency error variance at a fixed value of signal to noise as a function of the number of taps in the signal and frequency matched filters. The results are as shown in figures 5.25 & 5.26 for excess bandwidths of $\alpha = 0.1, 0.5, 0.9$. All the simulations were run with a loop noise bandwidth of $B_L = 0.001$ and signal to noise of $E_s/N_o = 15$dB. Also plotted as a dotted line on the graphs is the theoretical value of the frequency jitter at these values of $\alpha, B_L$ and $E_s/N_o$ from the equation [28],

$$\sigma_\nu^2 = \frac{4\alpha B_L T}{\pi^2 T^2} \frac{1}{E_s/N_o} \left(1 + \frac{1}{E_s/N_o}\right) \tag{5.59}$$

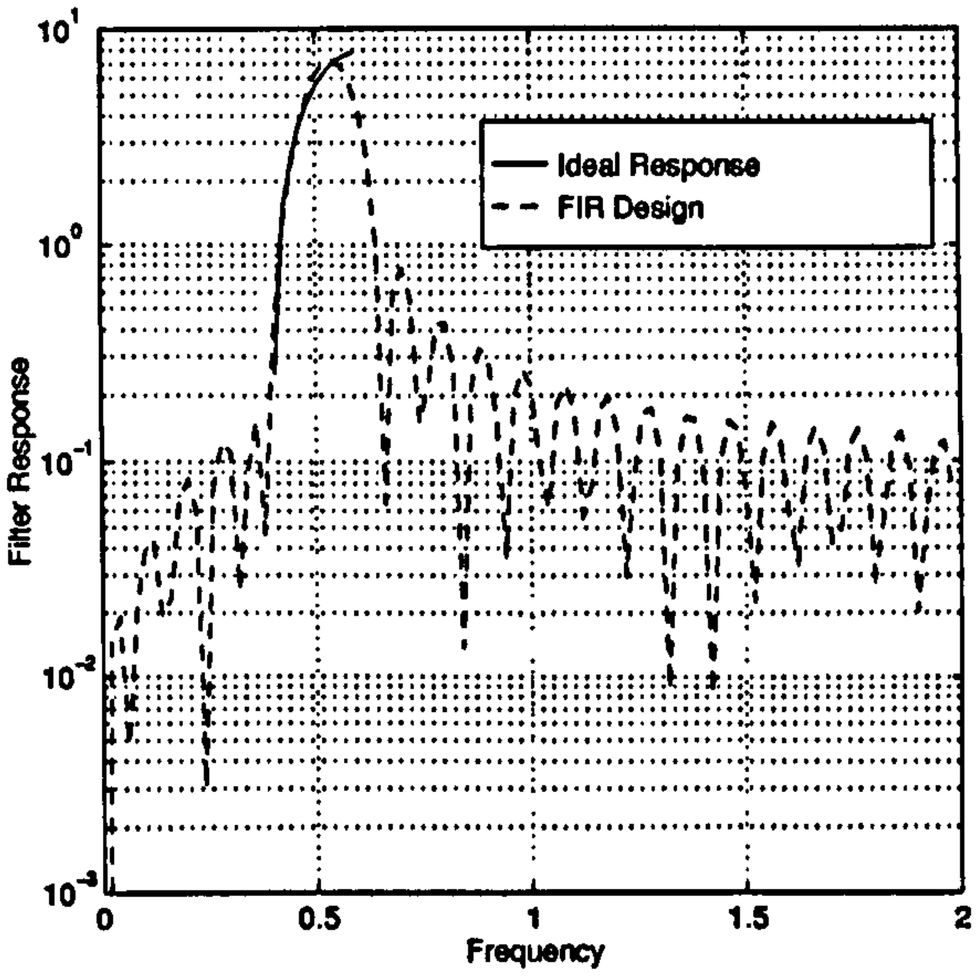
Note that because $\alpha$ is small the number of taps needed to minimise the frequency error variance is correspondingly large. The results appear erratic, showing that the frequency error variance does not monotonically decrease with increasing the number of filter taps illustrating the peculiar loop dynamics. There is scope for further research in this area.

### 5.8.4 Tracking Performance of BQ & ML Frequency-Locked Loops

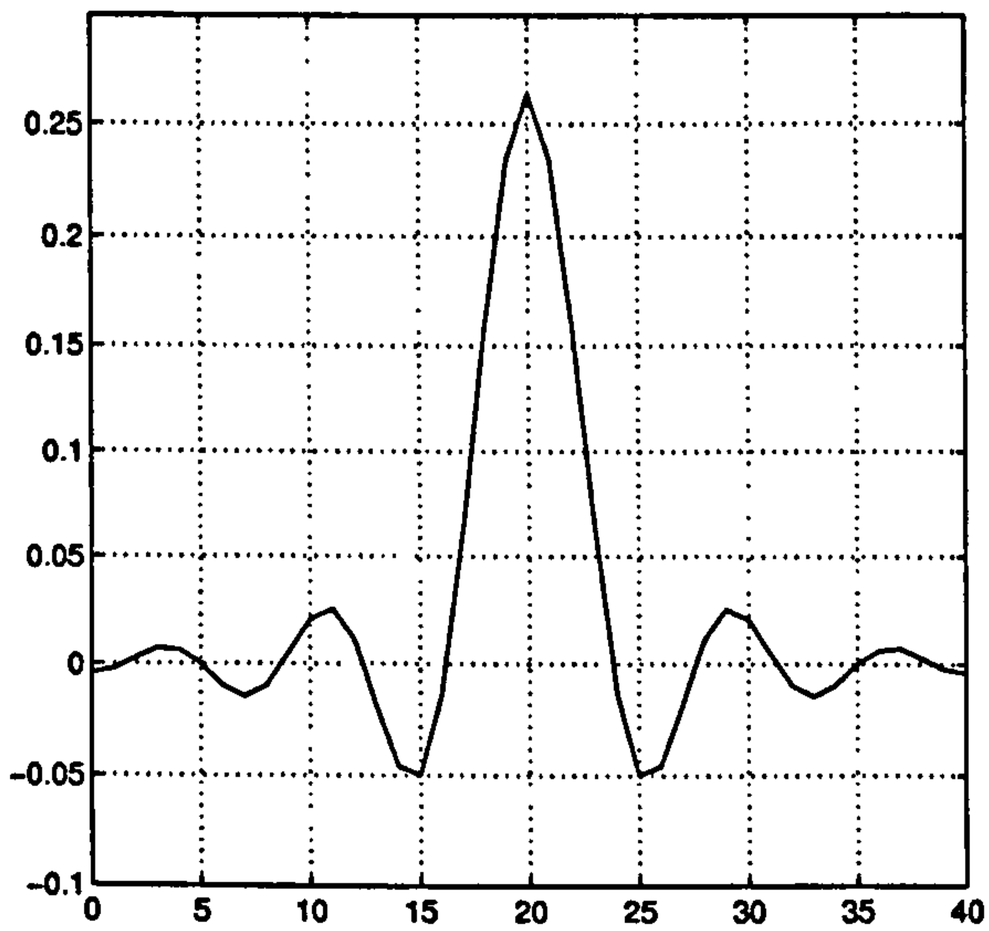
The tracking performance of the Balanced-Quadricorrelator and the Maximum-Likelihood frequency-locked loops is shown in figures 5.27 & 5.28 for a first order tracking loop. It is considered by the author that a first-order FLL will generally be adequate for most AFC applications. The purpose of the FLL is to reduce the frequency error so that some other tracking device, such as a PLL, can take over. Then any higher-order rates of change of the



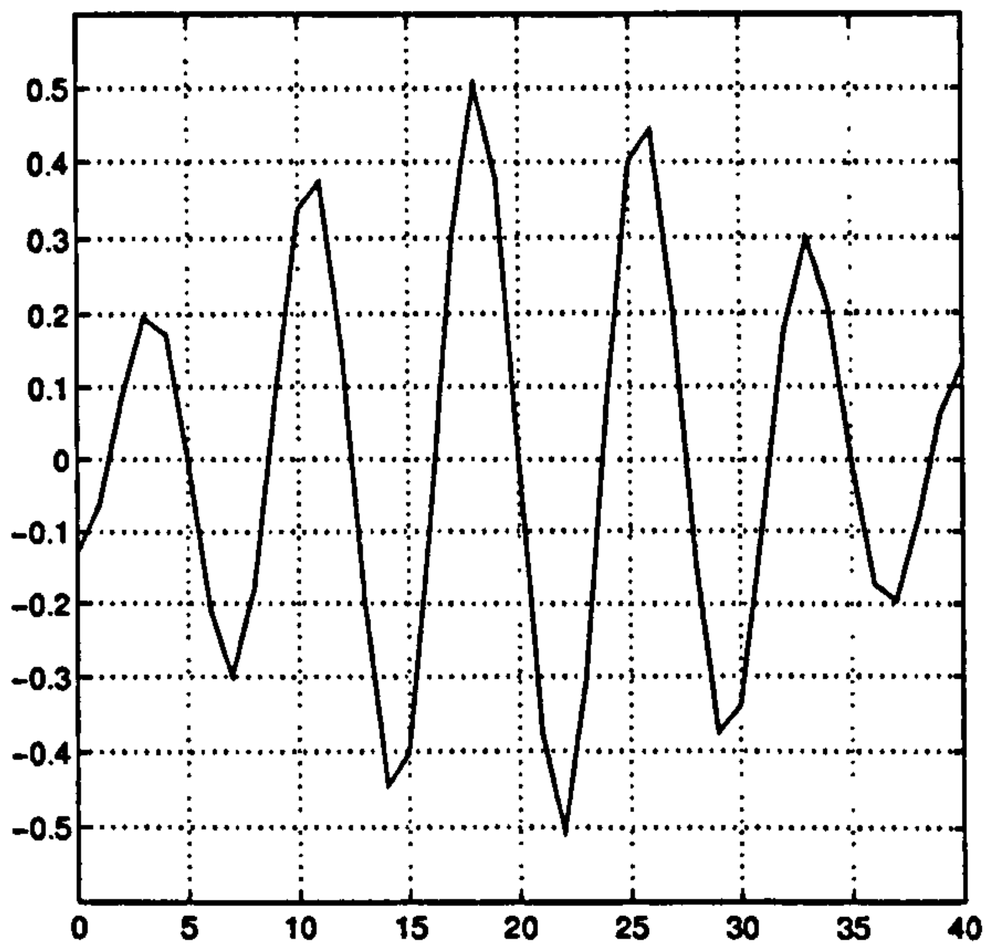

(a) Signal Matched Filter




(b) Frequency Matched Filter

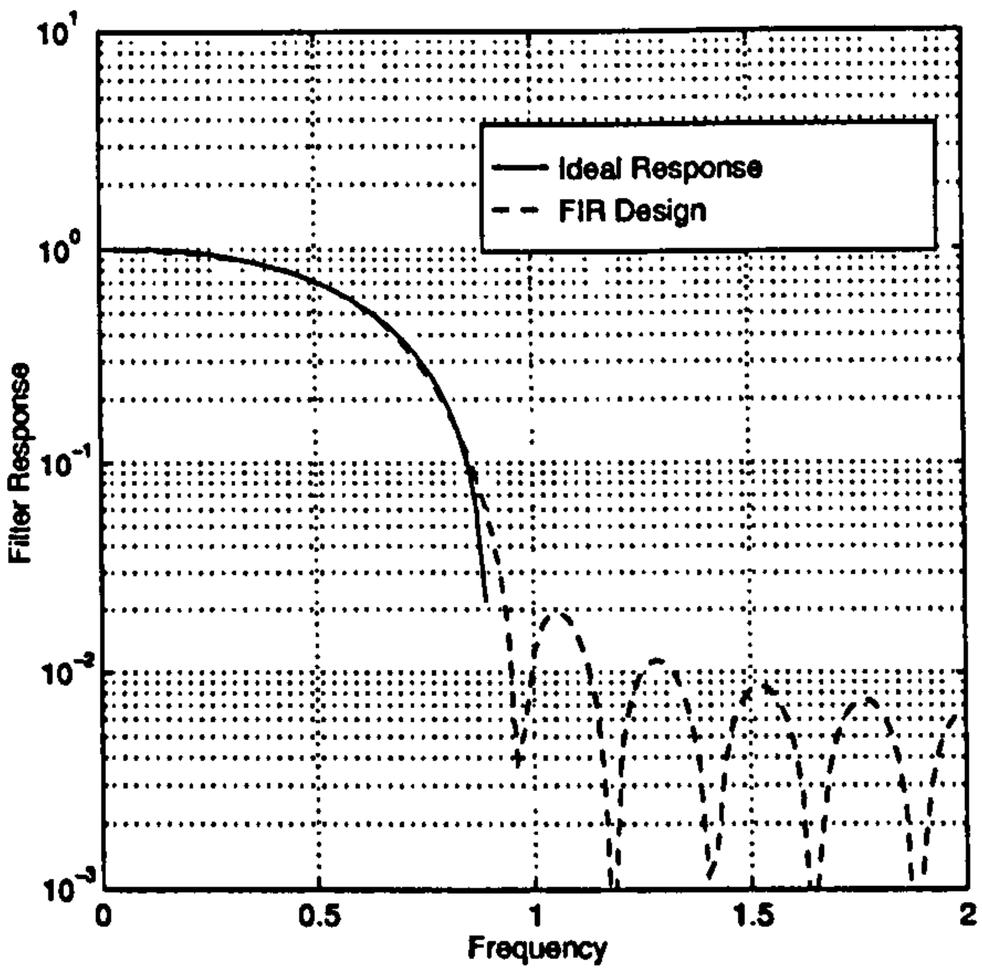


(c) Signal Matched Filter

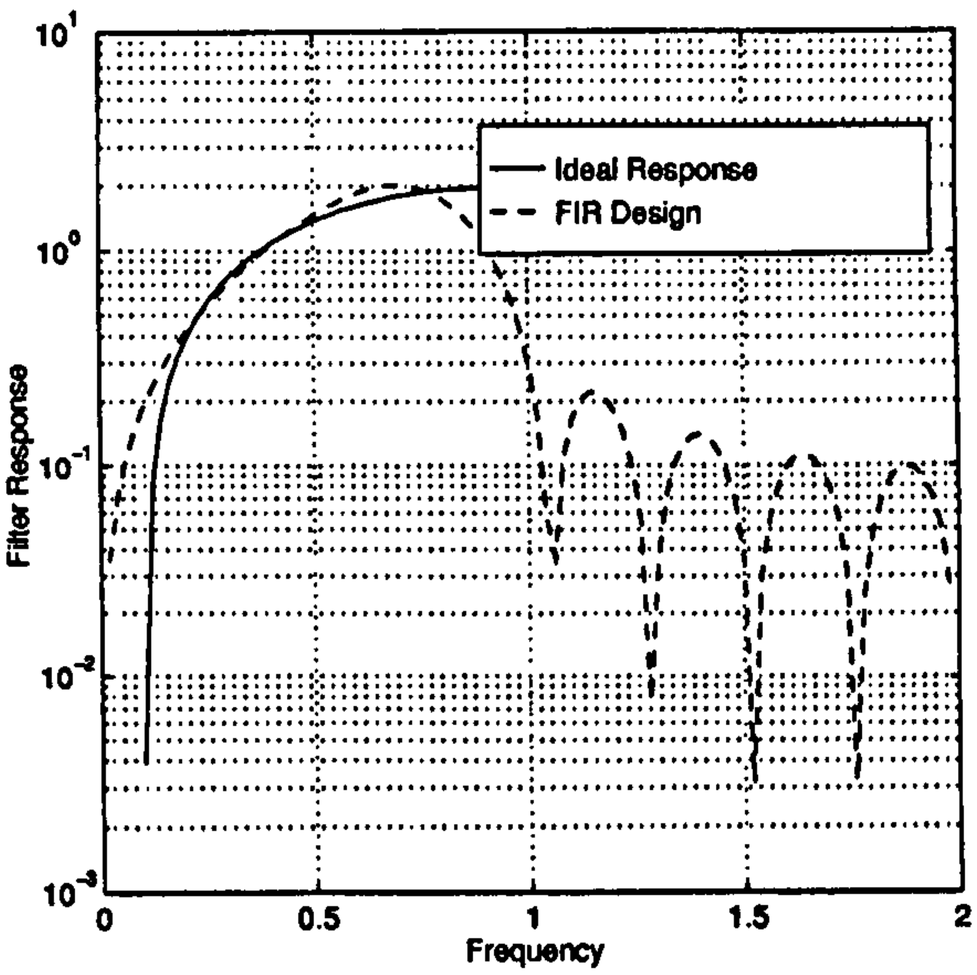


(d) Frequency Matched Filter

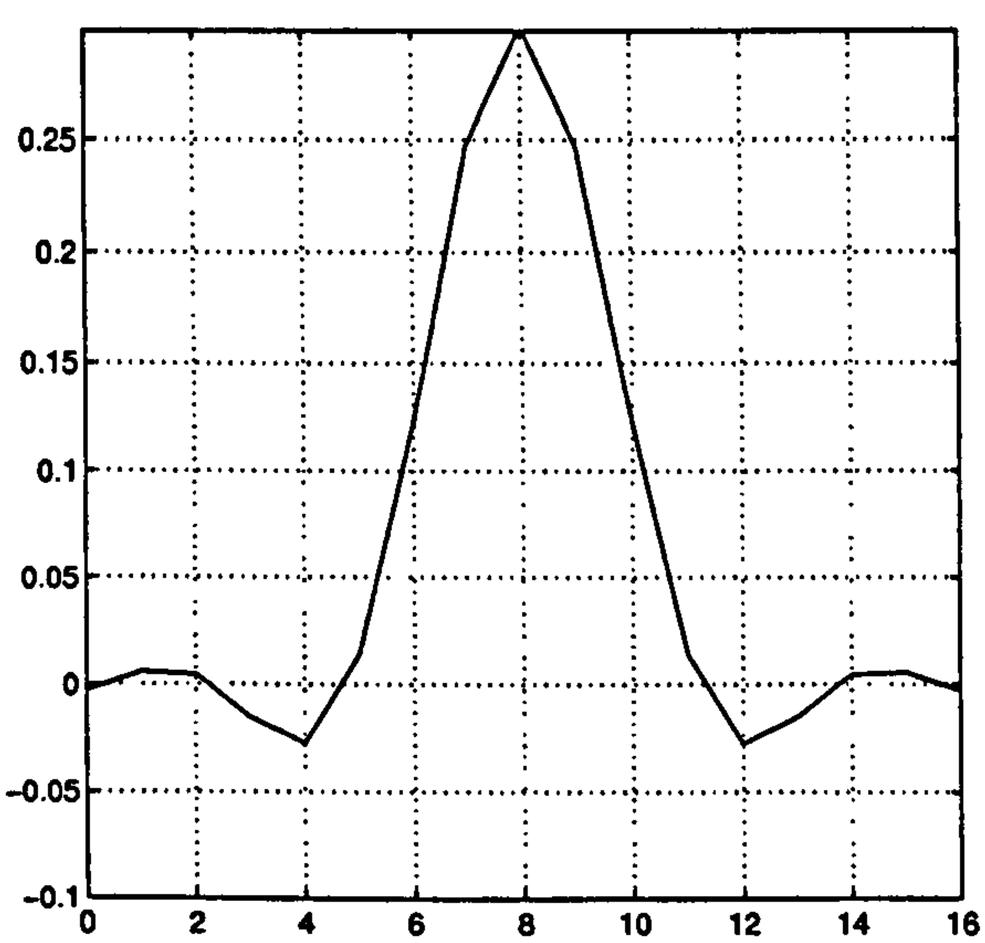
Figure 5.23: Frequency (top) and time responses of the Signal & Frequency Matched Filters. Design for excess bandwidth $\alpha = 0.2$ and $N = 41$ taps. Shown also is theoretical ideal for the frequency response.



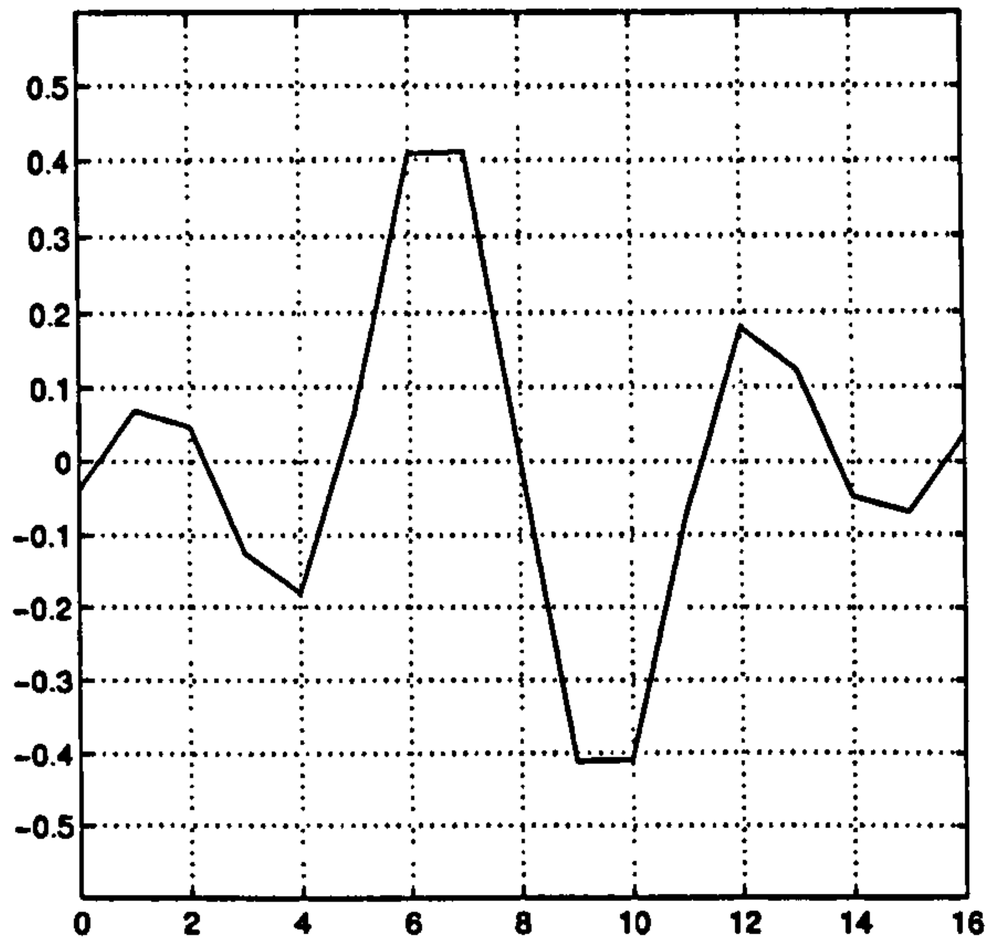

(a) Signal Matched Filter




(b) Frequency Matched Filter

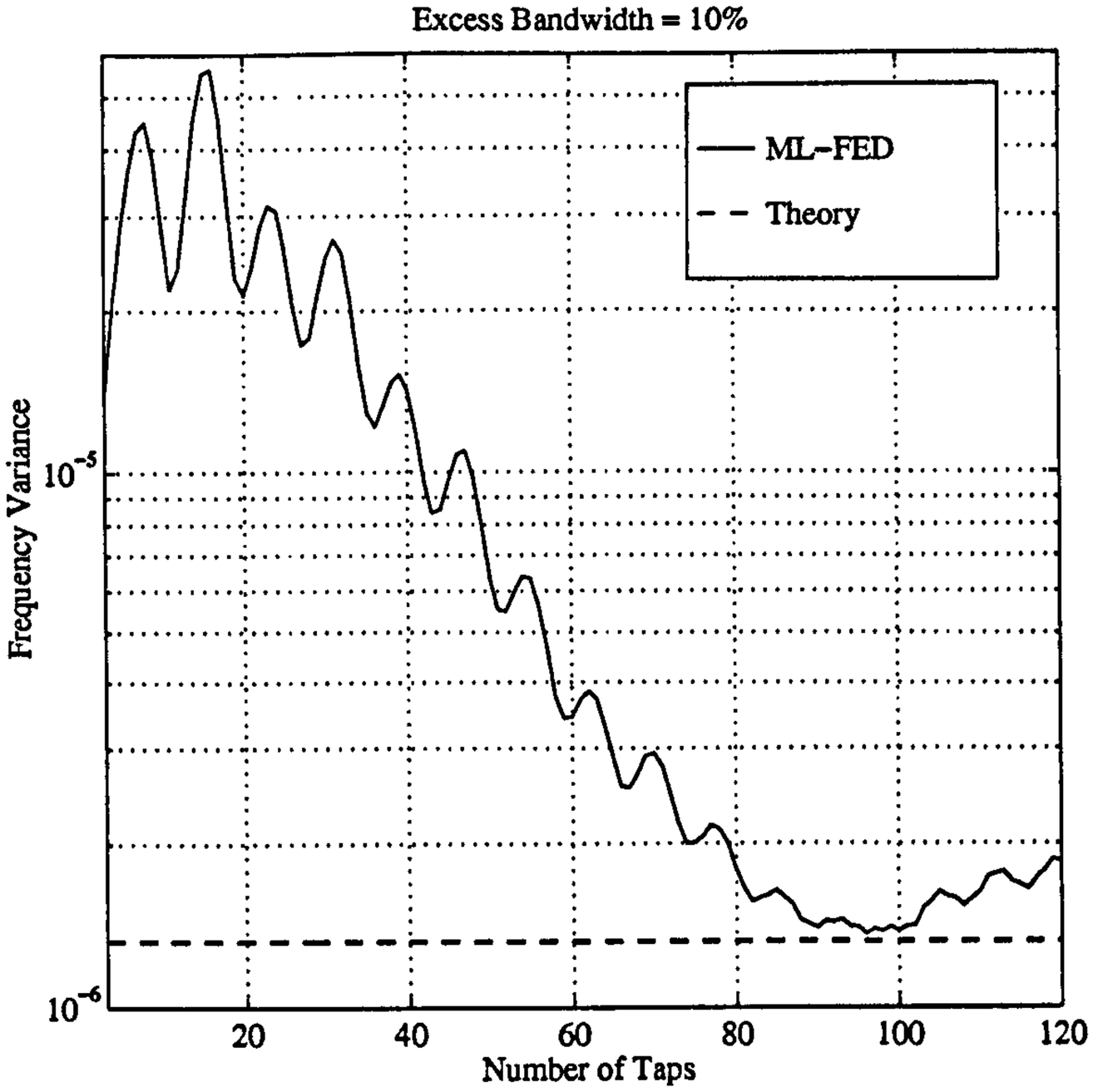



(c) Signal Matched Filter




(d) Frequency Matched Filter

Figure 5.24: Frequency (top) and time responses of the Signal & Frequency Matched Filters. Design for excess bandwidth $\alpha = 0.8$ and $N = 17$ taps. Shown also is theoretical ideal for the frequency response.



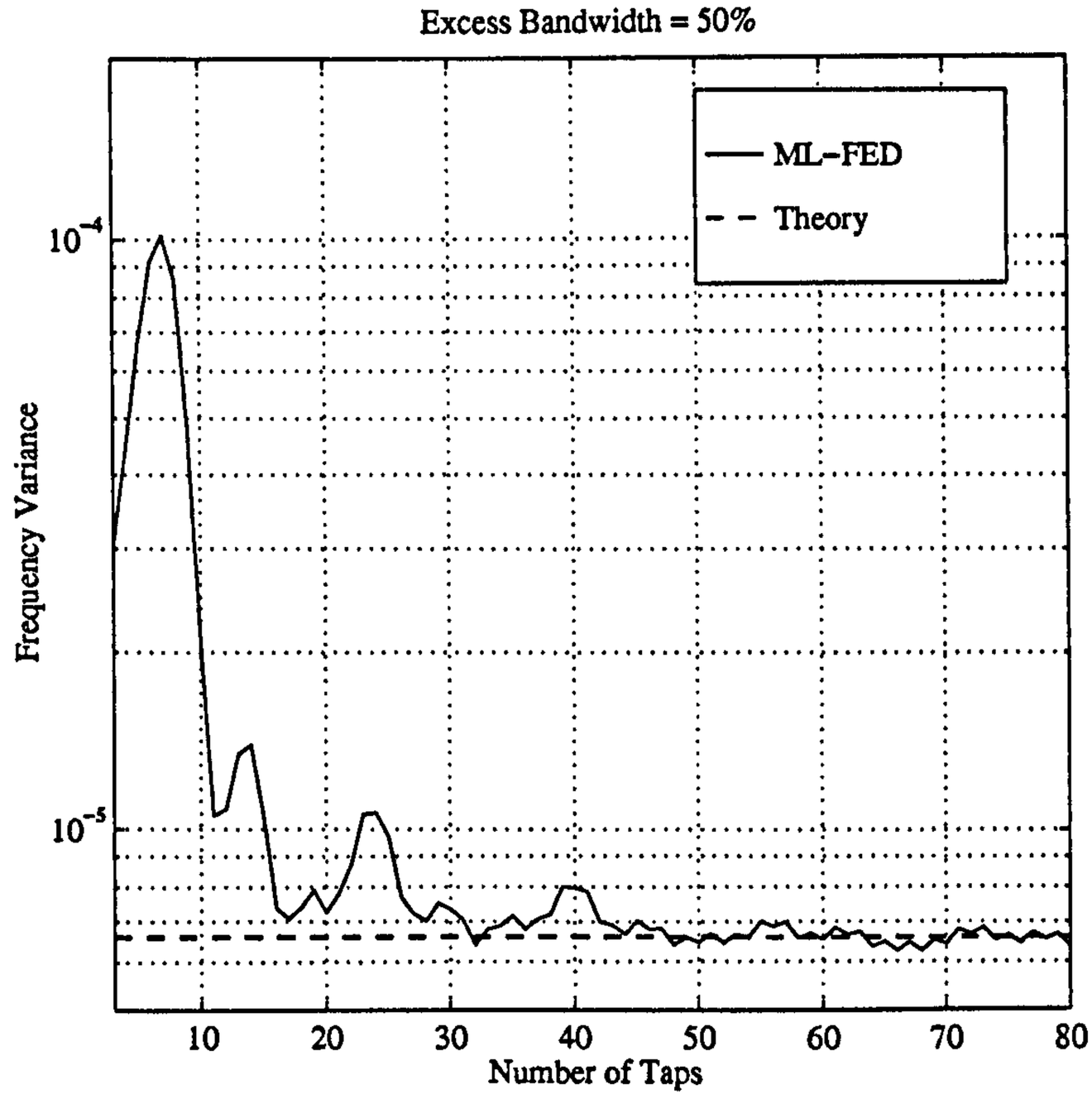




Figure 5.25: Frequency error variance as a function of the number of taps in the filter implementations. $E_s/N_o = 15$dB & $B_N = 0.001$.

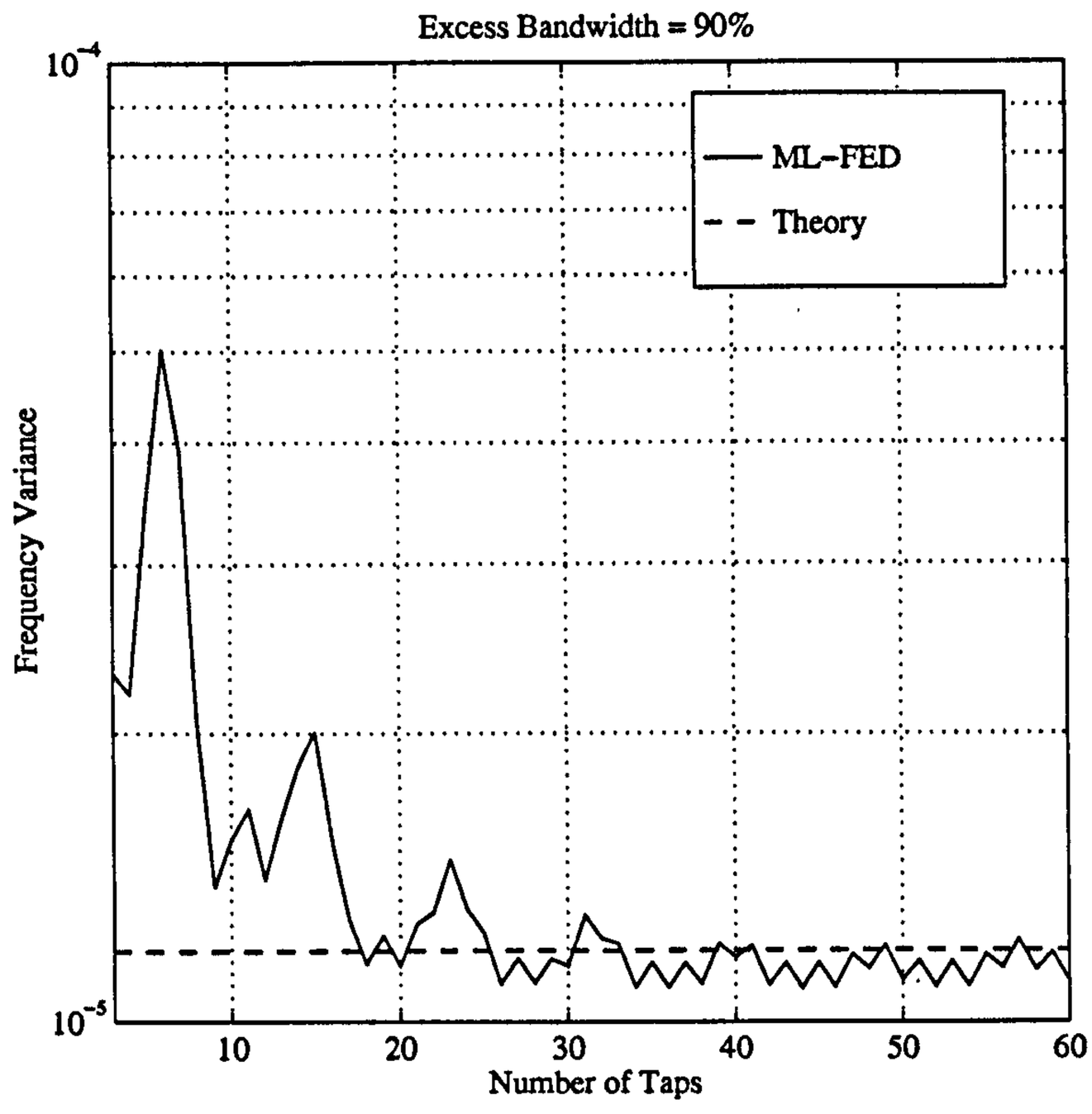



Figure 5.26: Frequency error variance as a function of the number of taps in the filter implementations. $E_s/N_o = 15$dB & $B_N = 0.001$.

input signal phase is tracked by this latter device and not the FLL. For the results shown, the modulation scheme is non-offset raised-cosine filtered QPSK and each plot corresponds to a different value of excess bandwidth. The results for the ML-FED loop are plotted for three values of timing phase, for $\tau = 0, 1/4, 1/2$ symbols, illustrating the need to have accurate timing information in order that the ML-FED works correctly. Of course, the BQ-FED does not require timing information, that is, it uses all of the samples available. From the previous simulations we determined the number of taps required for the design of the signal and frequency matched filters in the ML-FED for excess bandwidths of $\alpha = 0.1, 0.5, 0.9$ as $N = 99, 51, 31$. These selections are arbitrary based on the simulation results in figures 5.25 & 5.26. With these values the performance results match the theoretical ideal but the loops would probably work adequately with very much fewer taps.

The theoretical values in the figures showing the frequency tracking variance as a function of signal to noise is the ideal performance of the ML-FED, assuming perfect timing and perfect filter implementation. There is a close correspondence between the theory and the simulation results. Note that the theoretical values plotted are *not* the Cramer-Rao Lower Bound (CRLB). Both the BQ & ML-FEDs have performance which is someway off the CRLB, that is, the performance of the best linear unbiased estimator. A frequency error detector which *does* meet the CRLB is the rotational frequency error detector to be covered shortly.

Note that the performance of the ML-FED depends strongly upon the correct timing phase. We later illustrate a structure which will acquire timing and frequency simultaneously. Note that the performance of the BQ-FED is approximately the same as that of the ML-FED with a timing error of $\tau = 1/4$ symbols. This is intuitively correct as the BQ-FED in effect averages over all values of timing phase. The advantage of the BQ-FED is that, in this case at least, it is a fixed structure for each value $\alpha$ and being constructed of simple IIR filters it is computationally less expensive than a direct FIR implementation of the ML-FED. This is particularly true at low values of excess bandwidth.

### 5.8.5 Acquisition Performance of BQ & ML Frequency-Locked Loops

#### Acquisition for Small Frequency Offsets

The acquisition results for a small frequency offset of $f = 0.1/T$ for the BQ & ML-FED are as shown in figure 5.29. The results are compared with the theoretical analysis undertaken earlier for the small-signal model. One has to ensure that the error-tracking model one adopts is appropriate for the synchronisation loop in question. In this particular example we are interested in the response of a frequency locked loop to a static frequency offset; by examination this is the same as the response of a phase locked loop to a static phase offset. This was derived previously for the first order loop as,

$$\phi(t) = e^{-4 B_L t} \tag{5.60}$$

This equation has been plotted together with the simulation results in figure 5.29. As the frequency offset is small, and we are then allowed to use the linearised loop model, we see that the correspondence between theory and the simulation results is good.

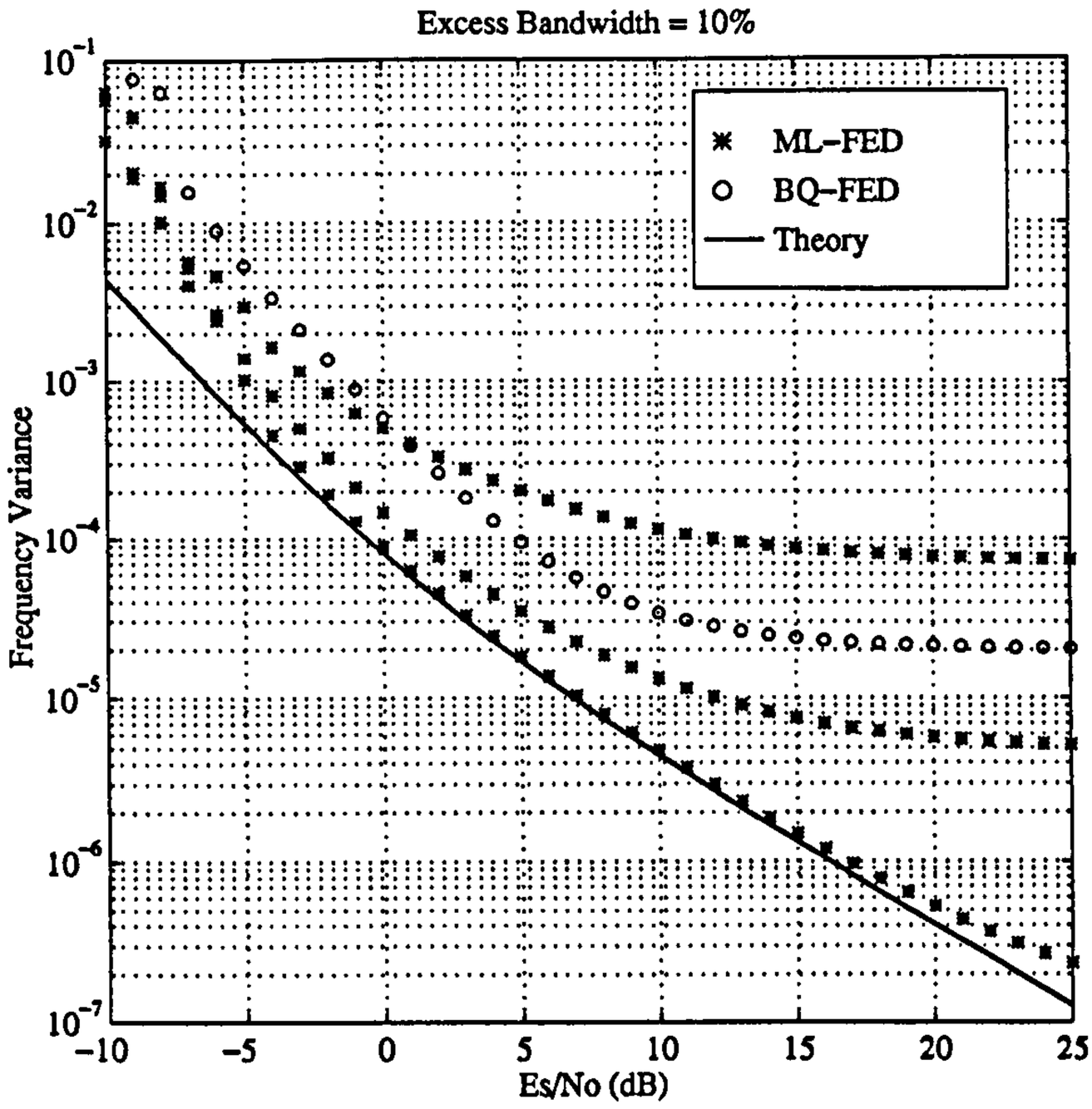



Figure 5.27: Tracking Performance of ML-FED & BQ-FED. Equivalent Noise Bandwidth is 0.001. Different plots for the ML-FED represent timing errors of $\tau = 0, 0.25, 0.5$. First order loop.

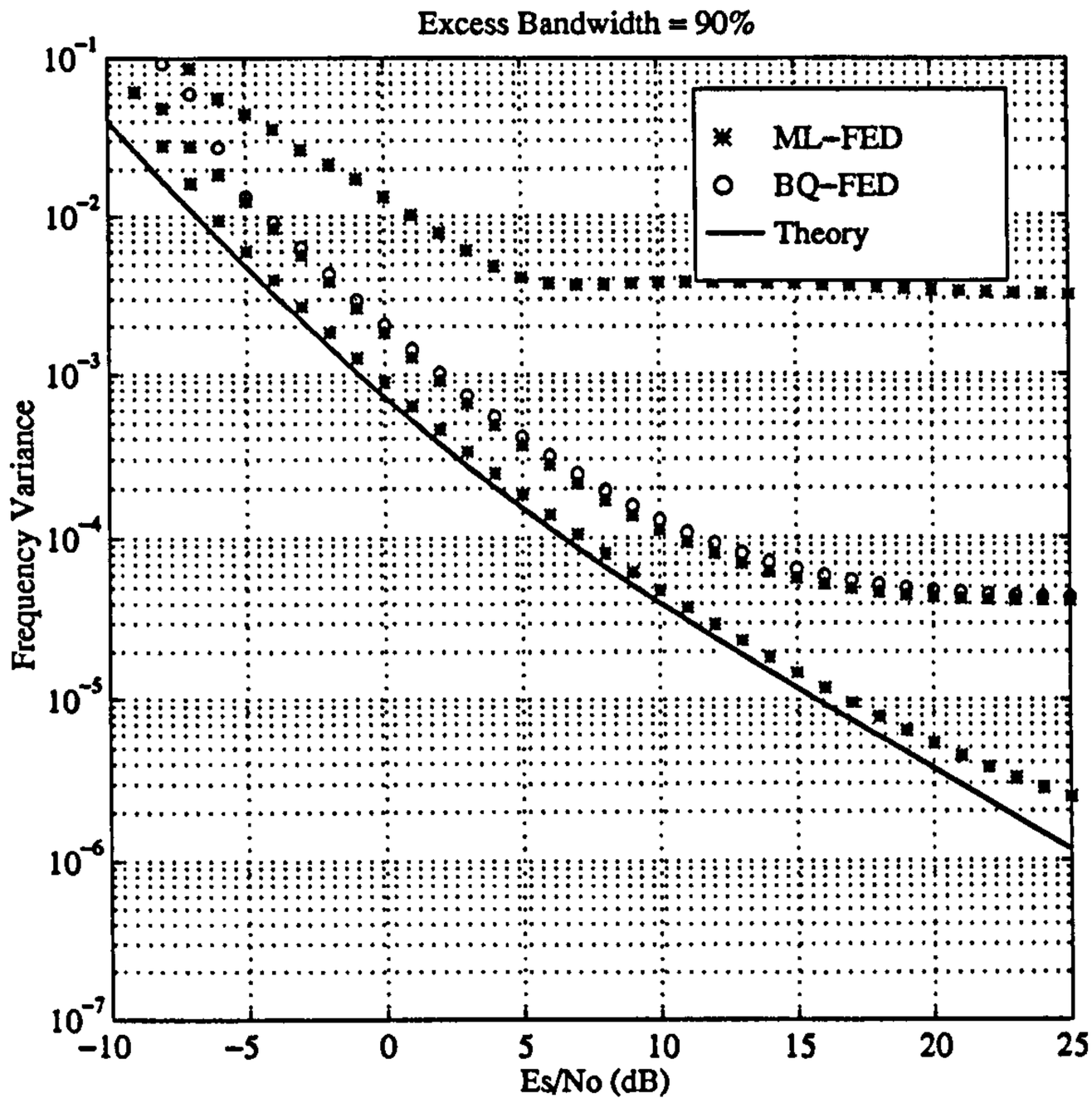
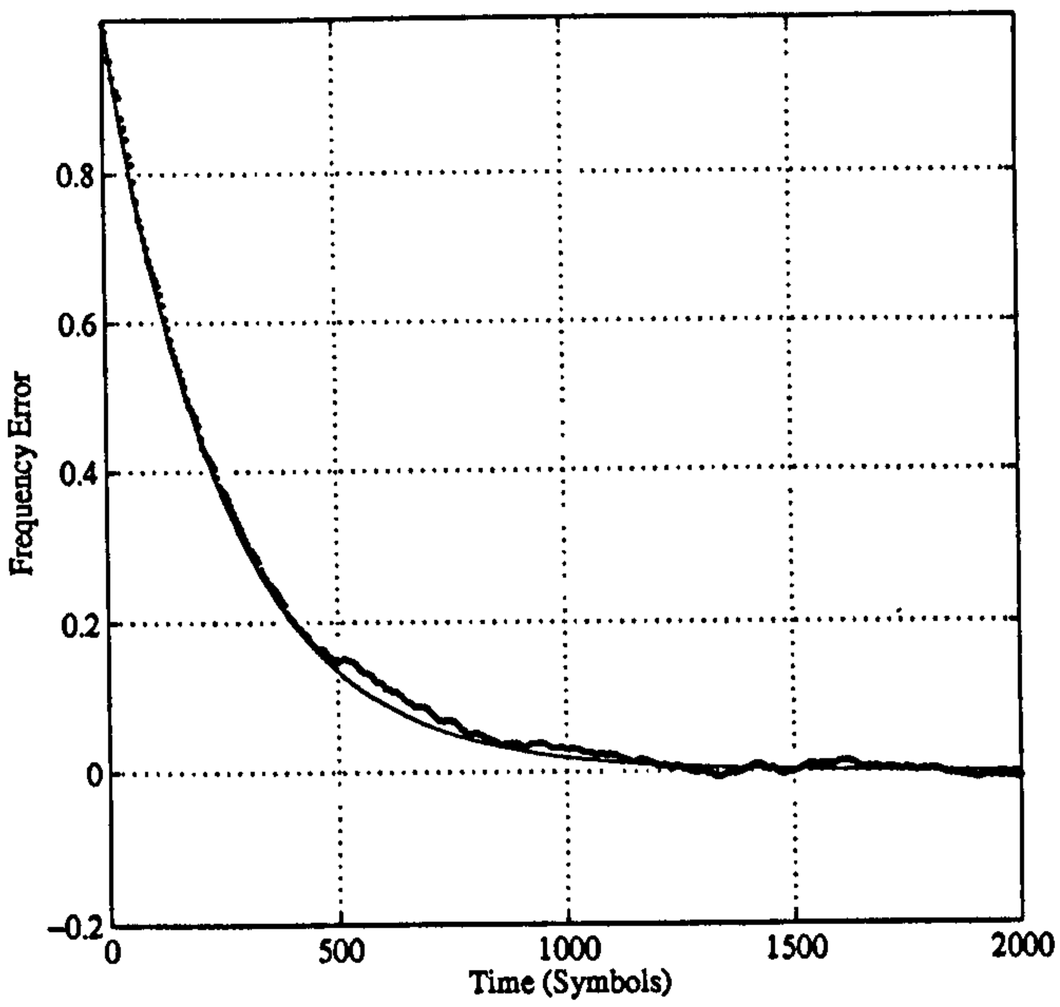
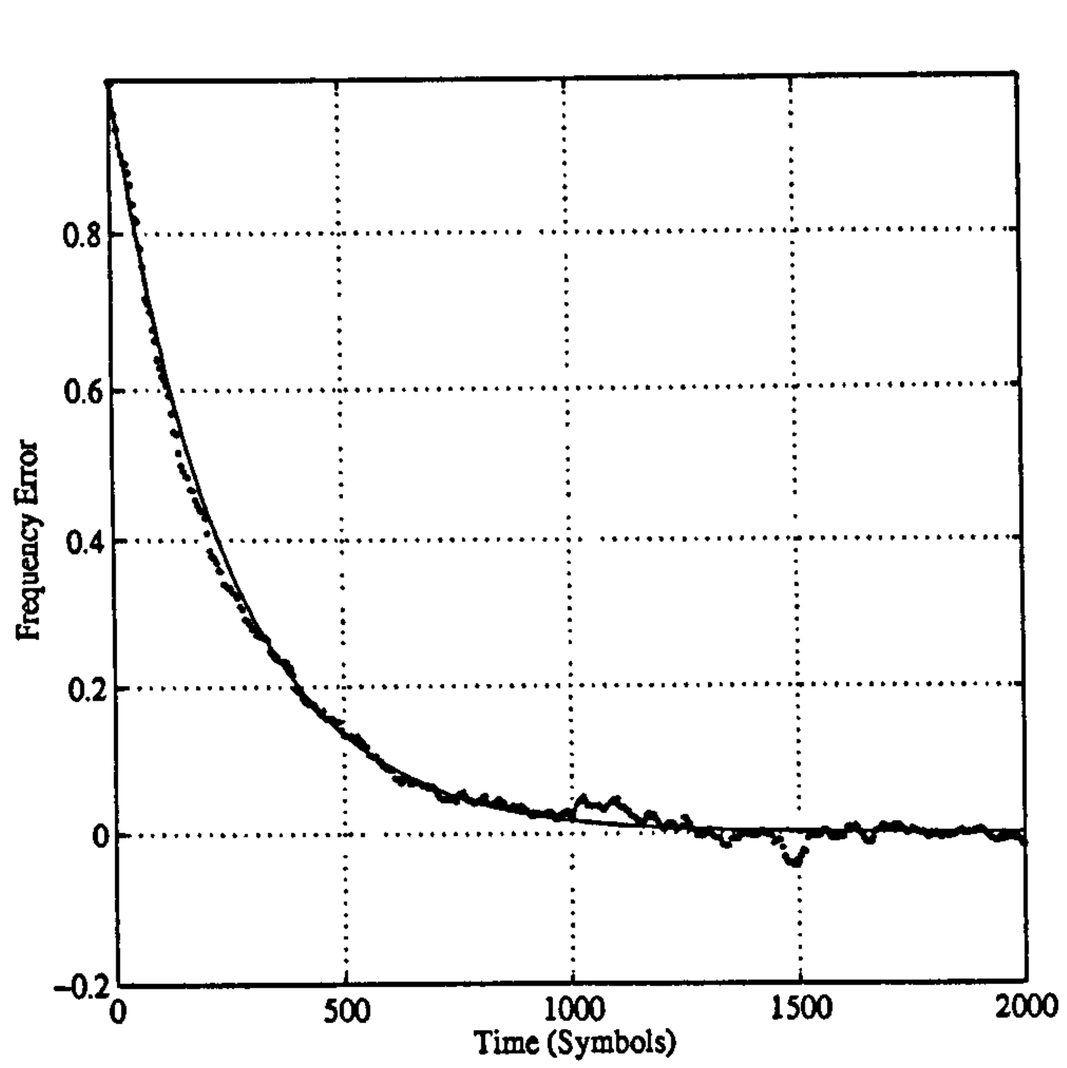



Figure 5.28: Tracking Performance of ML-FED & BQ-FED. Equivalent Noise Bandwidth is 0.001. Different plots for the ML-FED represent timing errors of $\tau = 0, 0.25, 0.5$. First order loop.




(a) ML-FED




(b) BQ-FED

Figure 5.29: Acquisition performance of the BQ & ML frequency locked loops with a small initial frequency error of $f = 0.1/T$. The modulation excess bandwidth is $\alpha = 0.5$ and the number of taps for the ML-FED filters is $N = 31$. The loop is first order with loop bandwidth $B_L = 0.001$.

### Acquisition for Large Frequency Offsets

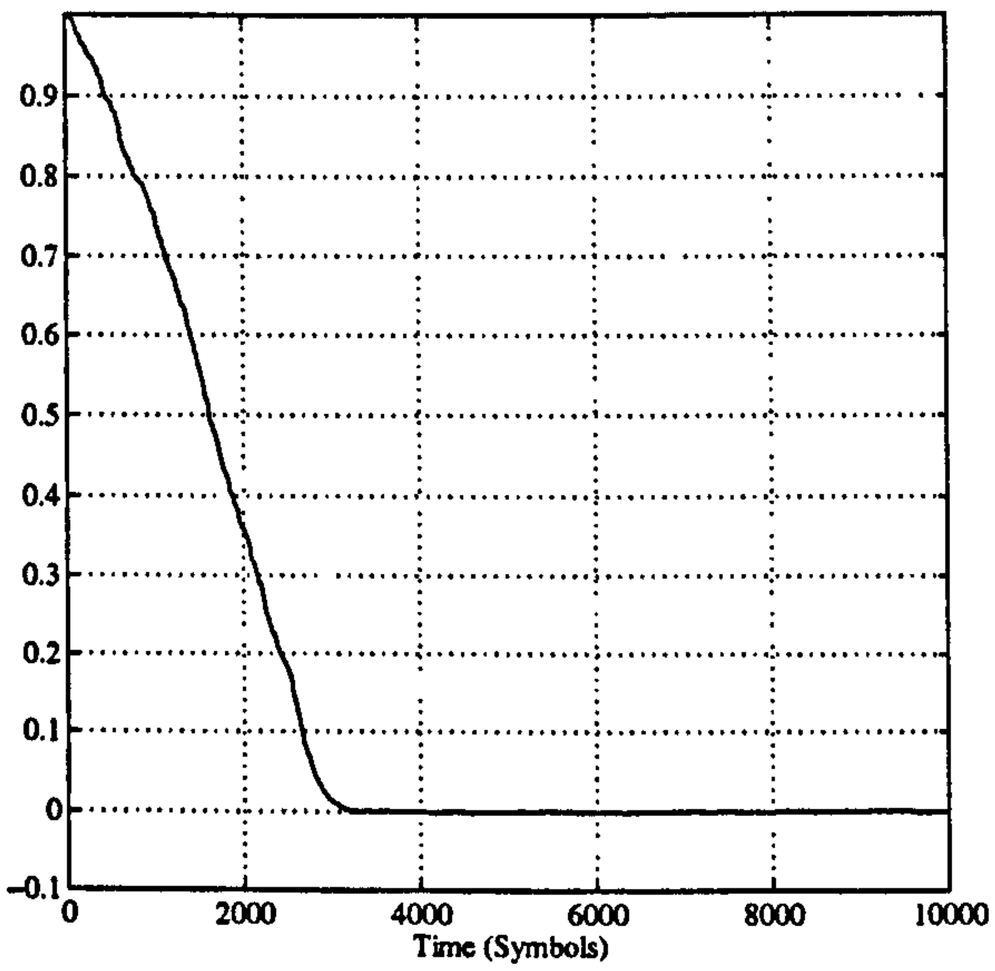
The purpose of the frequency-locked loop, or automatic frequency control (AFC) circuit, is to bring the carrier frequency close enough to the quiescent frequency of the PLL to facilitate carrier phase lock. Thus the immediate requirement for any AFC system is to demonstrate frequency acquisition for large carrier frequency offsets. Frequency errors could be as large as $f = \pm 1/T$, that is, of the order of the baud rate, in some satellite communication systems. Figure 5.30 illustrates frequency acquisition for a large frequency offset of $f = 1/T$ for frequency locked loops using both the ML & BQ-FEDs. The modulation excess bandwidth is $\alpha = 0.2$ and consequently a large number ($N = 41$) of taps is needed in the filter design. The loop is first-order with loop noise bandwidth $B_L = 0.001$. Figure 5.31 illustrates similar results but for an excess bandwidth of $\alpha = 0.8$. Plotted in the figures is carrier frequency and phase error for the two types of detectors as a function of time in symbols. The analytical determination of these curves would be difficult as one cannot use the linearised, small-signal model for the tracking loop. However, quasi-analytical methods involving phase-plane techniques may be applicable and may yield useful results.

The salient features of the acquisition plots are that the ML-FLL is slow to acquire but has better tracking performance when compared to the BQ-FLL. The acquisition time is also increased with decreasing excess bandwidth. These results are for no noise in the system. Intuitive reasons for the slow acquisition of the ML-FLL relative to the BQ-FLL is that the latter can make adjustments to the control loop on every sample, whereas the ML-FLL makes adjustments at the baud rate. This suggests that the ML-FLL might benefit in terms of its acquisition performance by using every sample during acquisition and only samples taken at the baud rate during tracking. Note also from the figures that although the *mean* frequency error is small there is enough frequency jitter to cause large random variations in the carrier phase. This "random-walk" or phase noise process has to be dealt with by the PLL. This leads to the dilemma in AFC system design we refer to as the "Phase Noise Trap." The PLL has to have a bandwidth wide enough to track the phase-noise produced by the FLL, but narrow enough to reject additive noise.

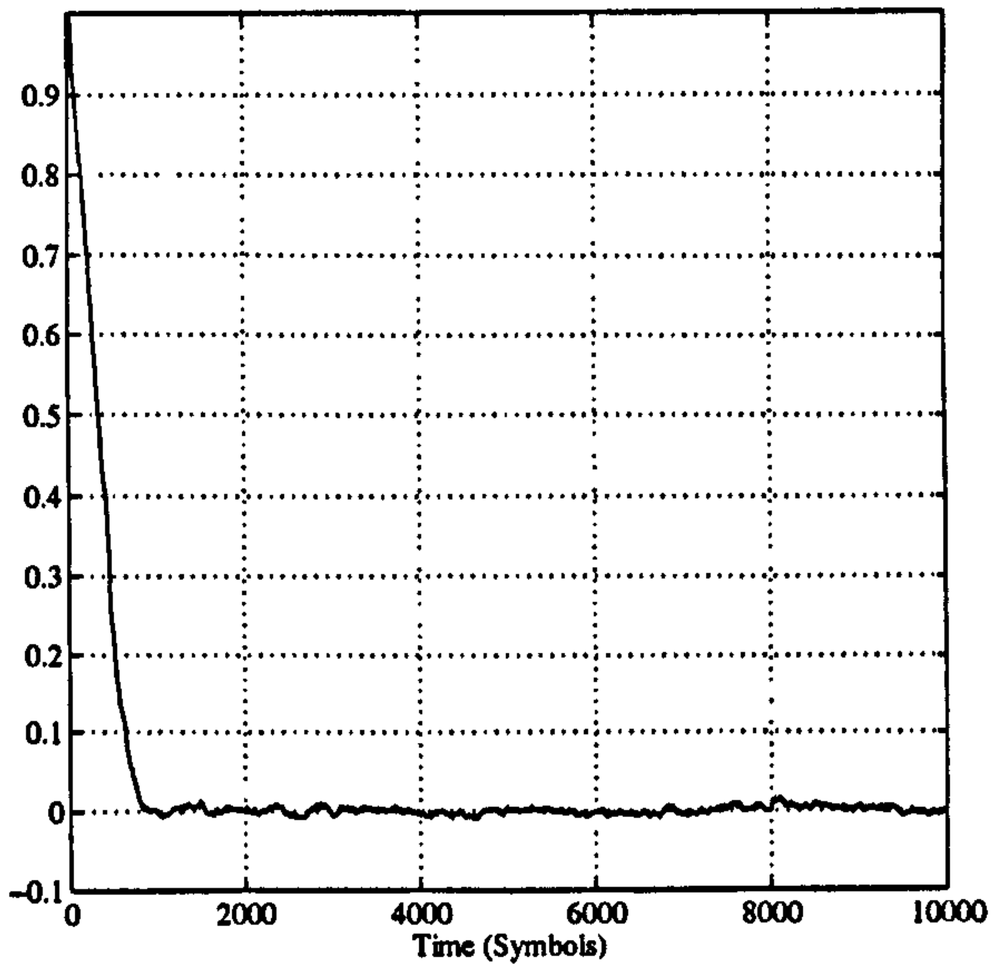
## 5.8.6 Tracking Performance of Rotational Frequency Error Detectors

The rotational frequency error detector is implemented as a feedforward loop. A typical system implementation in SPW is as shown in figure 5.32. The theory of operation of this system was discussed previously. The rotational FED is only suitable for small frequency offsets and might typically be used in conjunction with non-coherent, for example differential phase, detection or where feedforward carrier phase recovery, such as the V & V algorithm [61], is employed.

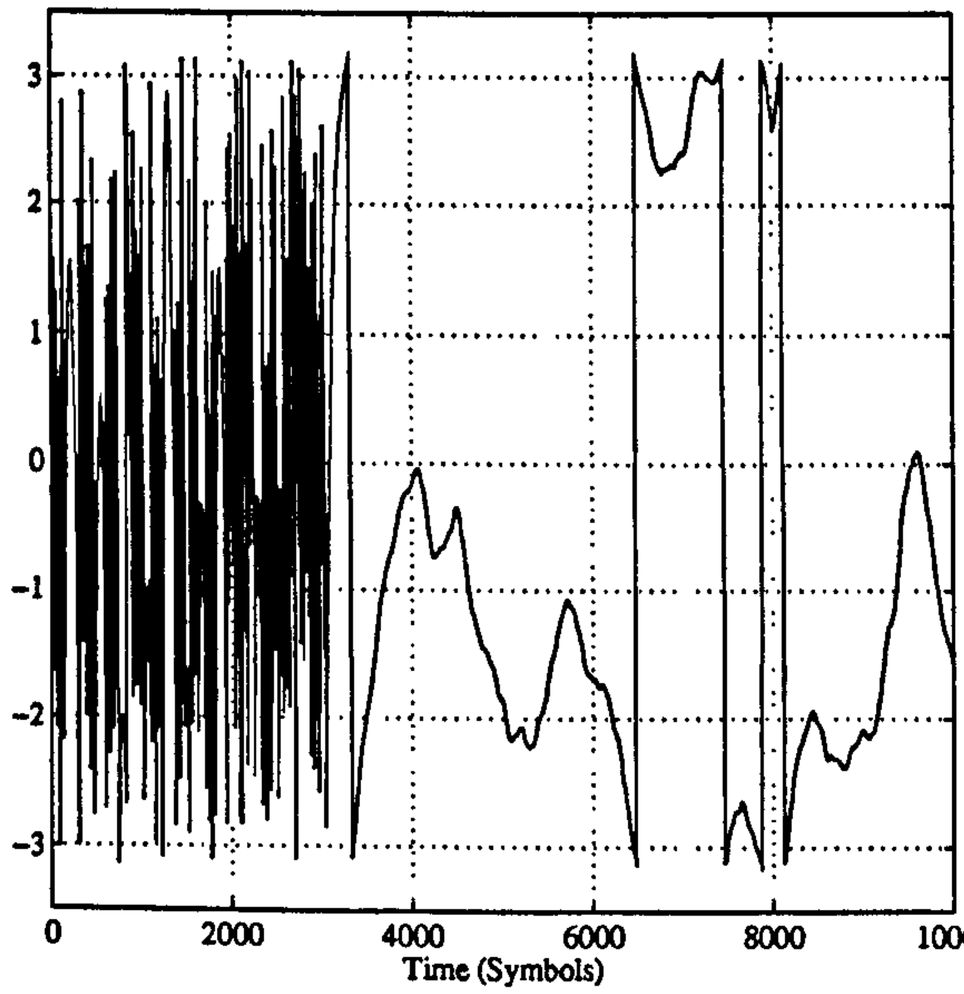
The tracking performance of this FED, unlike the other schemes studied earlier, converges on to the CRLB provided the smoothing filter is designed appropriately. It has been shown [22] that the optimum filter is not a sliding-mean filter but a filter which has coefficients $\beta_k$ given by equation (5.61). This is because the noise samples after receiver filtering are no longer



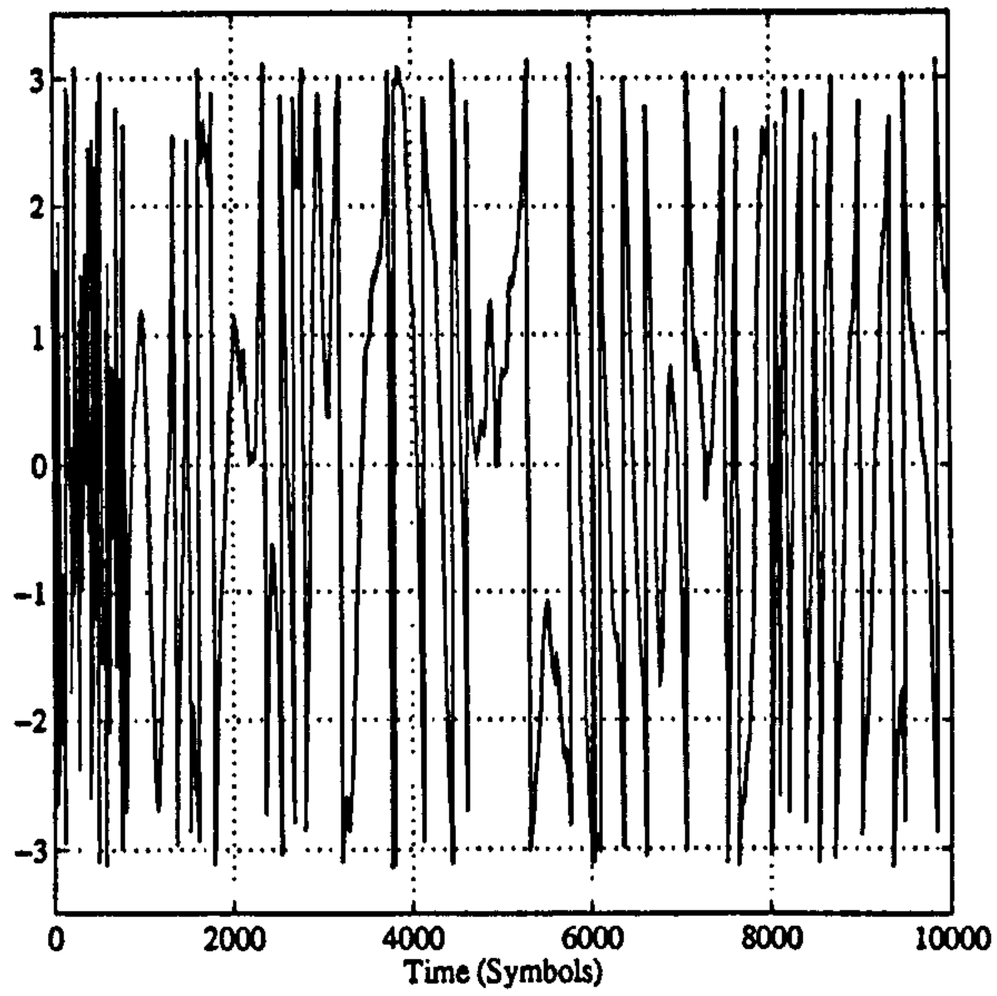

(a) ML-FED Frequency Error




(b) BQ-FED Frequency Error

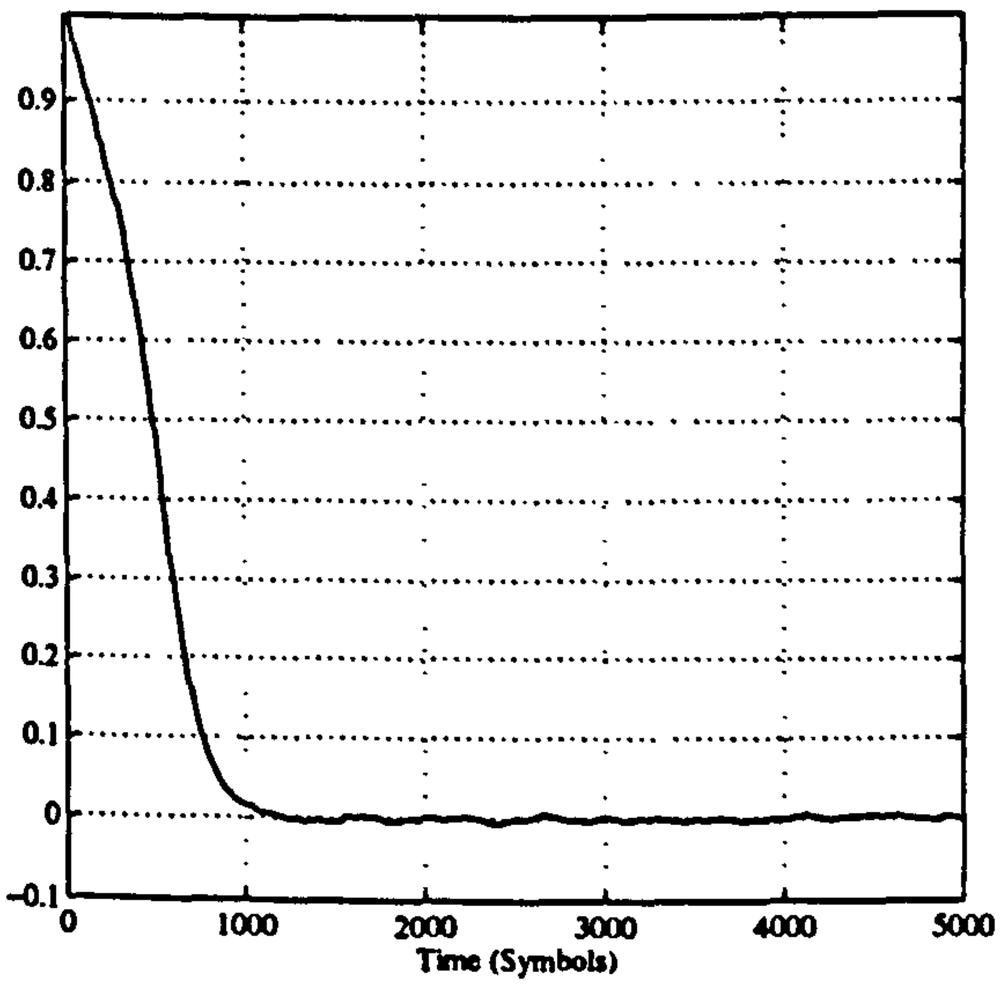



(c) ML-FED Phase Error

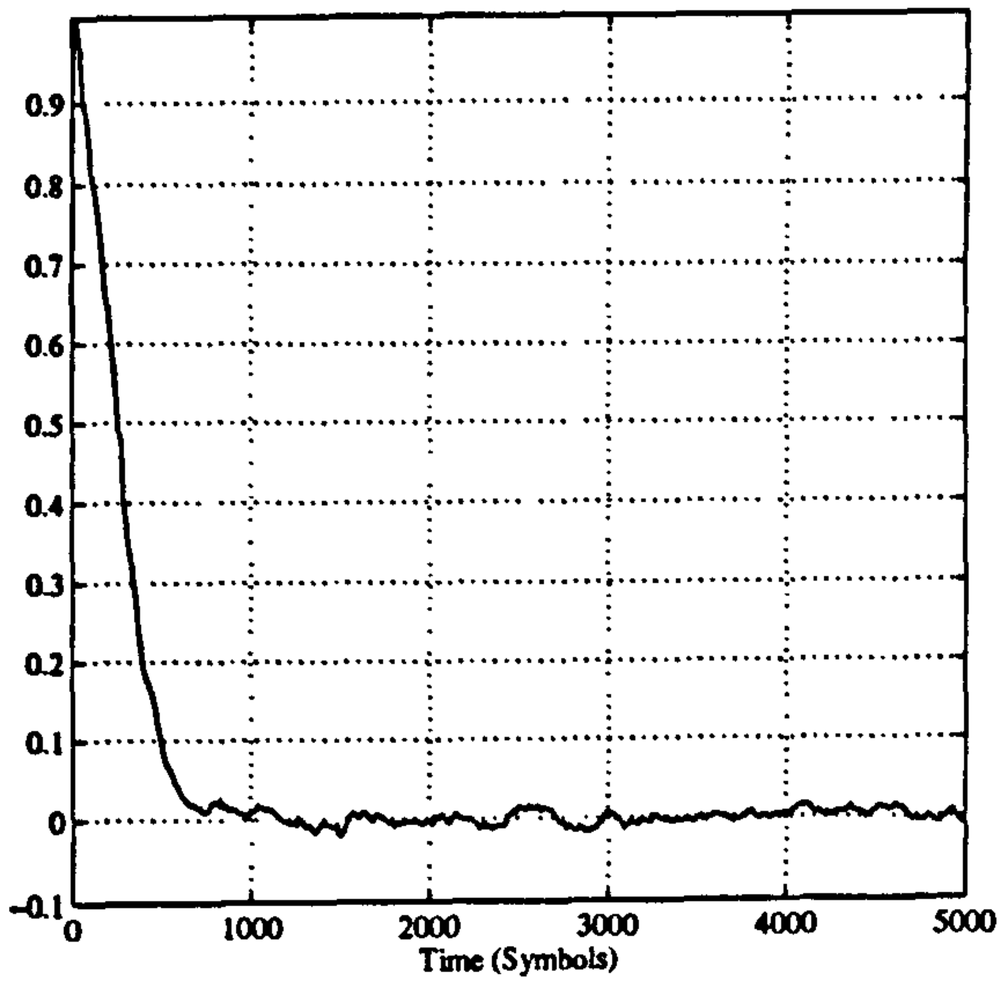



(d) BQ-FED Phase Error

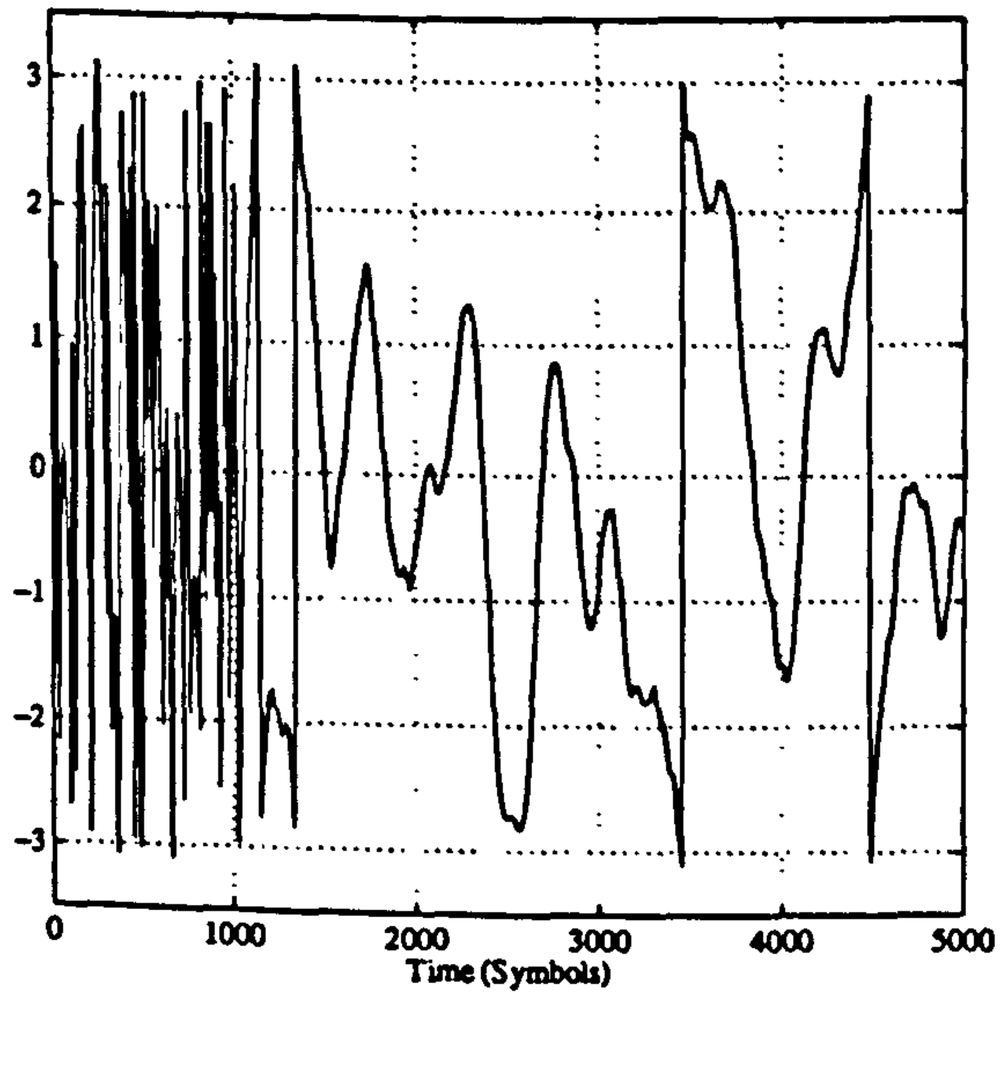
Figure 5.30: Acquisition performance of the BQ & ML frequency locked loops with a large initial frequency error of $f = 1/T$. The modulation excess bandwidth is $\alpha = 0.2$ and the number of taps for the ML-FED filters is $N = 41$. The loop is first order with loop bandwidth $B_L = 0.001$.



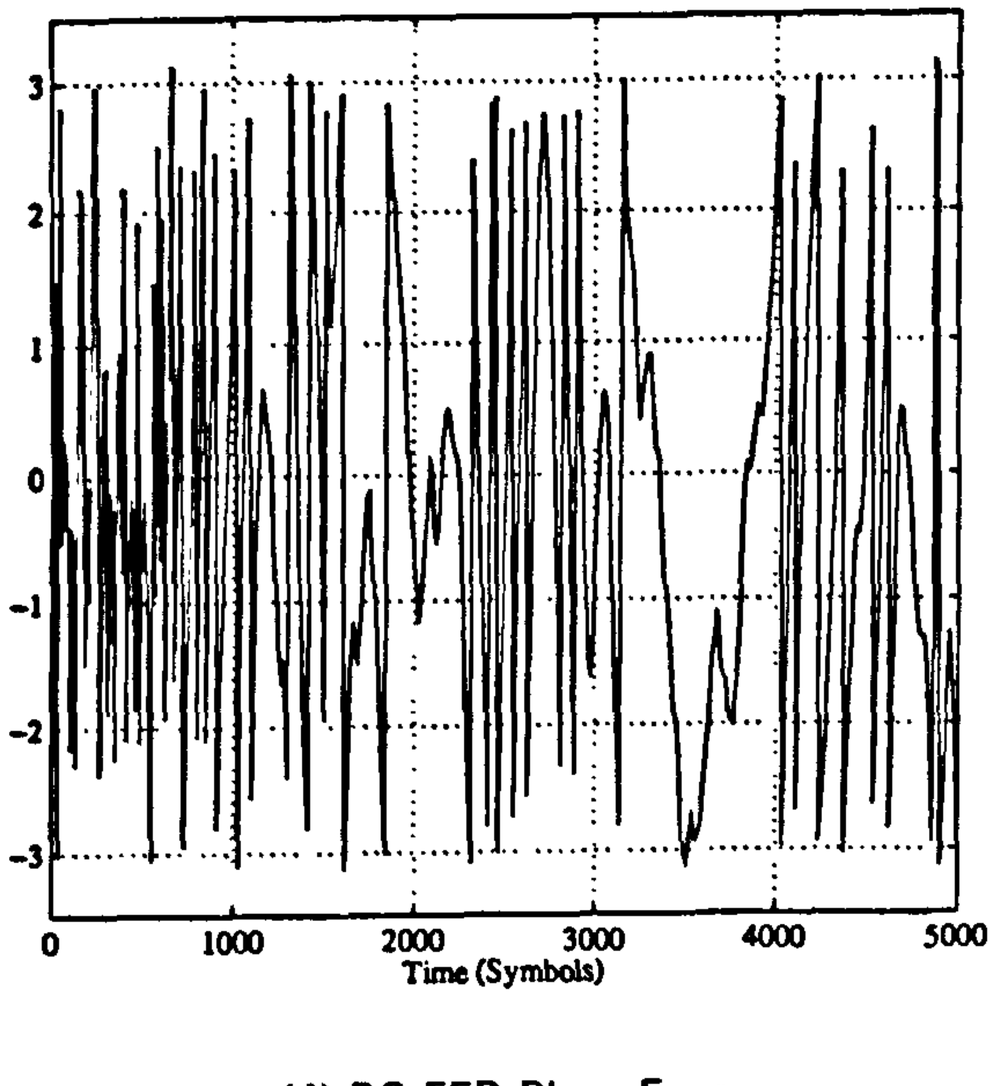

(a) ML-FED Frequency Error




(b) BQ-FED Frequency Error




(c) ML-FED Phase Error




(d) BQ-FED Phase Error

Figure 5.31: Acquisition performance of the BQ & ML frequency locked loops with a large initial frequency error of $f = 1/T$. The modulation excess bandwidth is $\alpha = 0.8$ and the number of taps for the ML-FED filters is $N = 17$. The loop is first order with loop bandwidth $B_L = 0.001$.

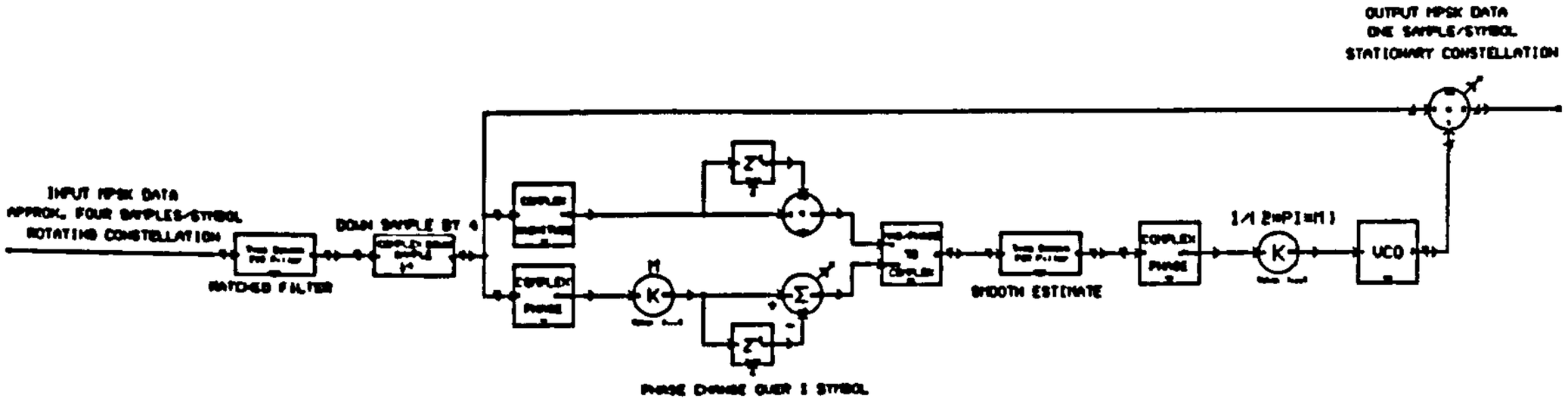



Figure 5.32: SPW Implementation of Rotational FED.

uncorrelated.

$$\beta_k = \frac{1}{2}\left[\frac{K^2-1}{4} - k(k+1)\right] \tag{5.61}$$

Figure 5.33 shows the tracking results for the rotational FED with a sliding mean and an optimum filter. Both filters are of length $L = 17$ taps. The MATLAB script for generating the appropriate filter coefficients is as shown in appendix B.

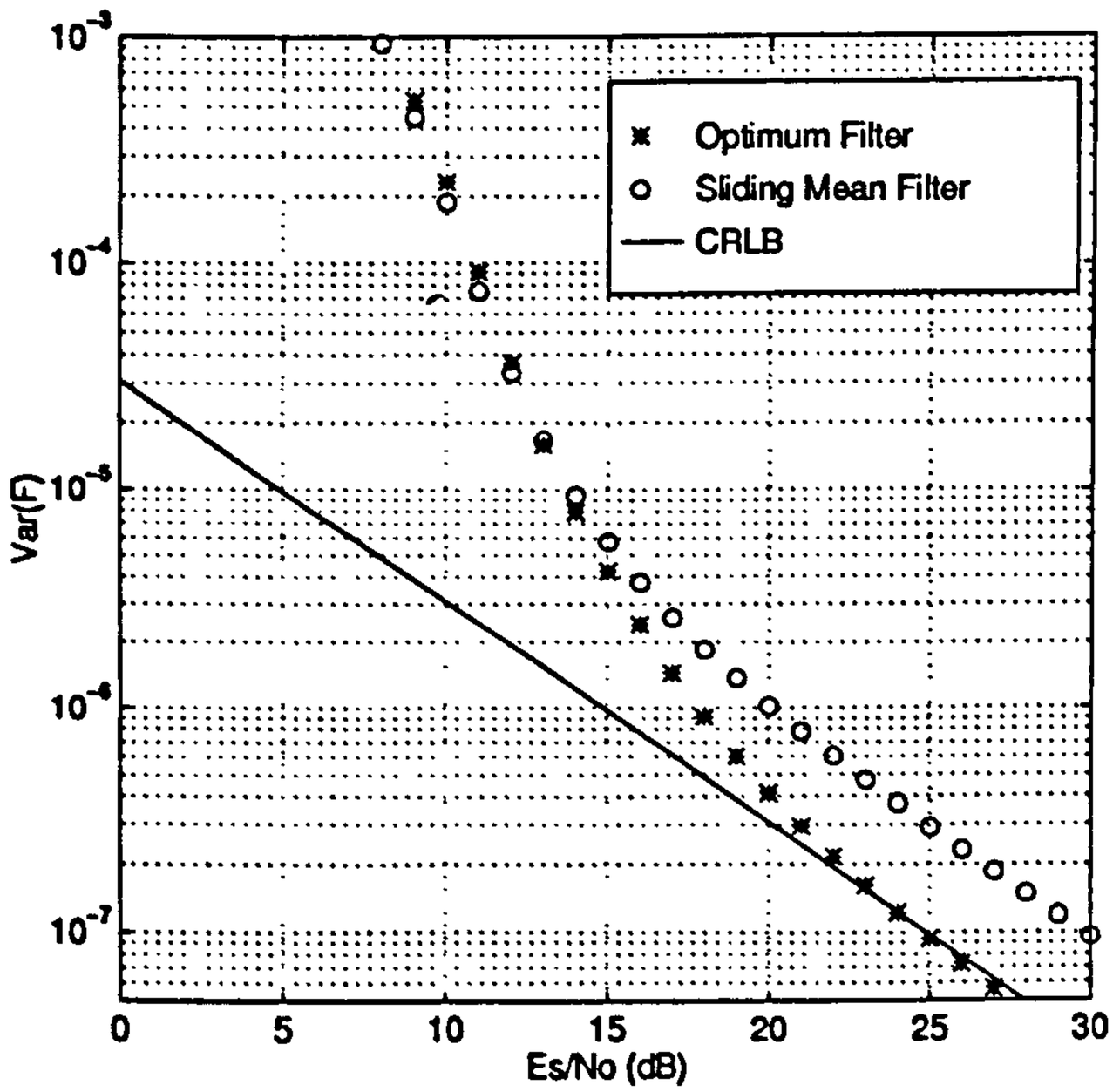



Figure 5.33: Tracking Performance of ROT-FED. Number of taps in feedforward filter is $L = 17$. Results compare the optimum filter with a sliding-mean filter.

# Chapter 6

# Symbol Synchronisation

## 6.1 Introduction

The timing in a data receiver must be synchronised to the symbols of the incoming data signal. The function of the timing synchroniser is to ensure that the clock ticks, or strobes, coincide with the widest opening in the signal eye-diagram. This then ensures that symbol decisions will have maximum immunity to the effects of noise.

In the implementation of modems by digital techniques the signal is sampled before being processed by some form of digital signal processor (DSP). There are a number of ways that this can be accomplished [15] as shown in figure 6.1. There are two means by which the sampling of the signal is accomplished; in the top two diagrams in figure 6.1 the phase of the sampling clock is adjusted so that some of the data samples coincide with the optimum sample points of the input PAM data stream. These schemes are referred to as being *synchronous.* In the analogue scheme (top) the phase of the sampling clock is adjusted by an analogue pre-processor whilst in the hybrid scheme (middle) the phase of the sampling clock is adjusted by the DSP. There are instances, however, where it is desirable to leave the sampling clock, or analogue to digital converter (A/D), to "free-run" any correction being made to the sampled PAM data stream by the DSP. This is shown as the digital scheme (bottom) in figure 6.1. Such schemes are referred to as being *asynchronous.* Intuitively the asynchronous schemes are more attractive than their synchronous counterparts. Provided that the PAM data stream is sampled at a high enough rate, above the Nyquist rate, then all the information in the input analogue waveform is available in the sampled version. This implies that the optimum sampling points can be *reconstructed* from the points available in our sampled output.

The problem of asynchronous timing recovery is illustrated in figure 6.2 which shows the input analogue wave (solid line) and the samples available to our DSP (shown as o and *). In the figure the PAM waveform has been sampled at two samples per symbol. The number of samples per symbol required by a synchronizer is an important figure as it sets the required processing bandwidth. In figure 6.2 the o mark the strobe samples and the * mark the extra samples used by the synchroniser for steering purposes. We shall see that most schemes rely on 1, 2 or 4 samples per symbol.

The receiver signal processor could simply choose those samples closest to the widest eye-

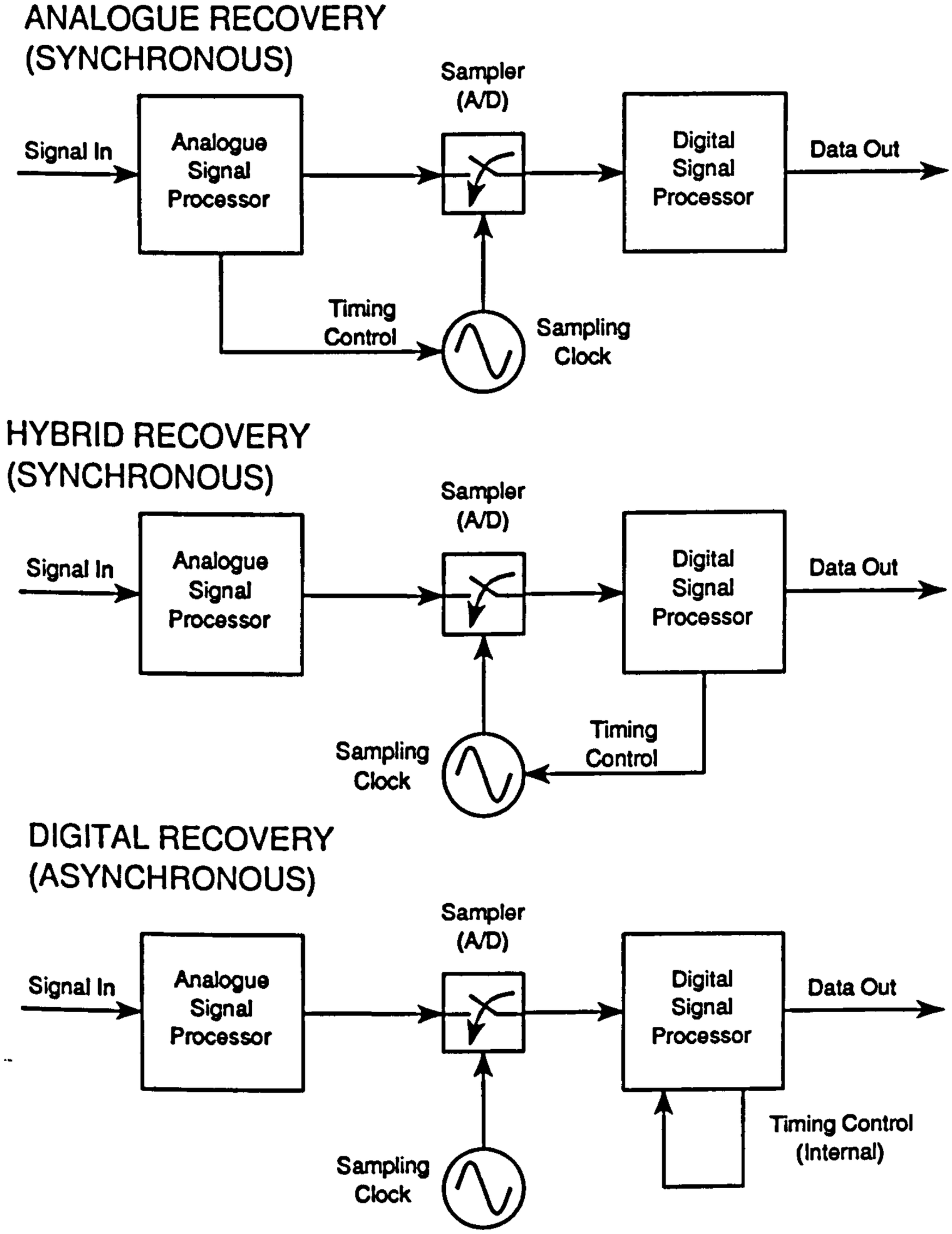



**Figure 6.1: Timing recovery methods.**

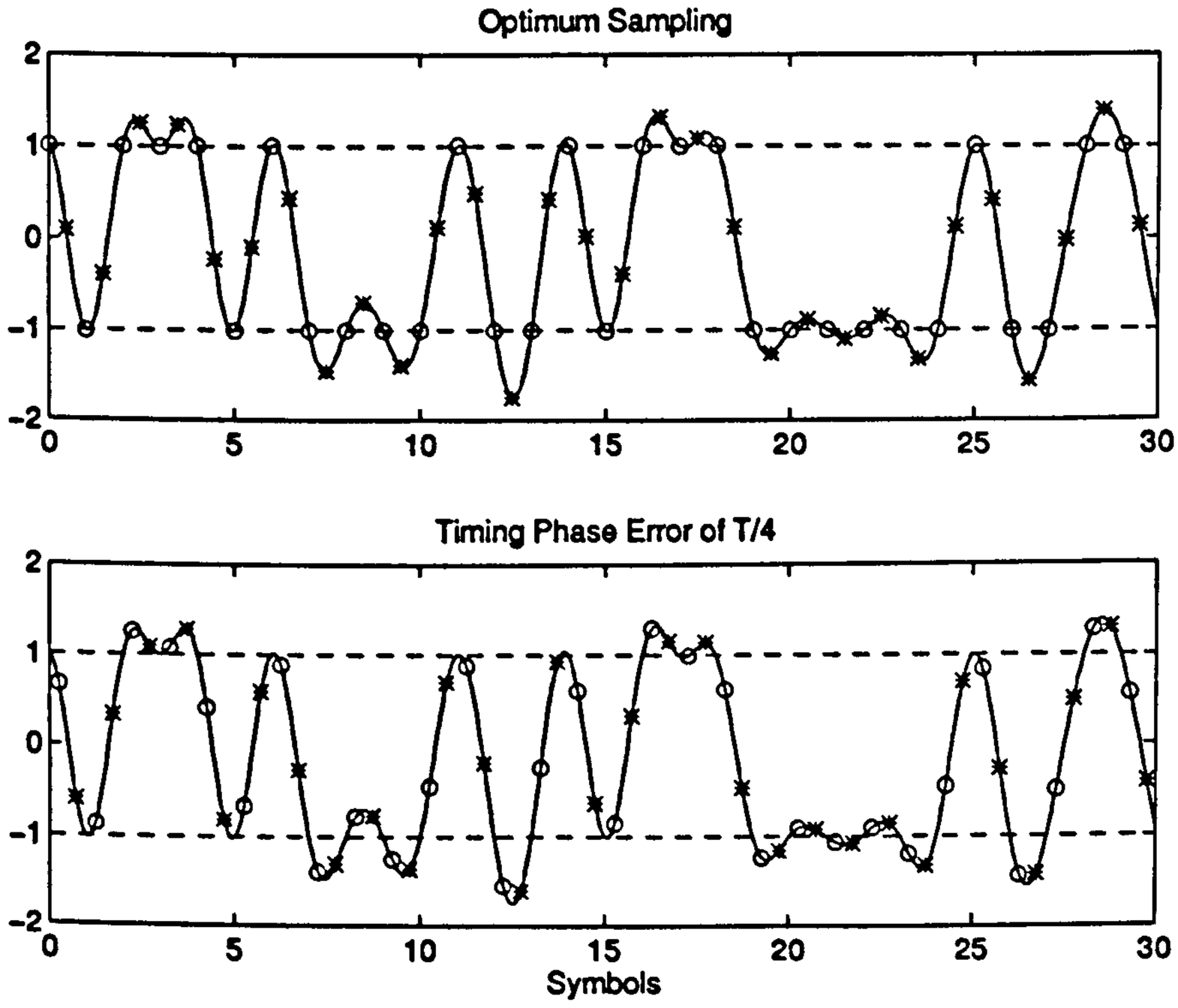



Figure 6.2: Sampling of PAM waveform. The waveform here is Raised-Cosine filtered ($\alpha = 0.3$) binary-antipodal data.

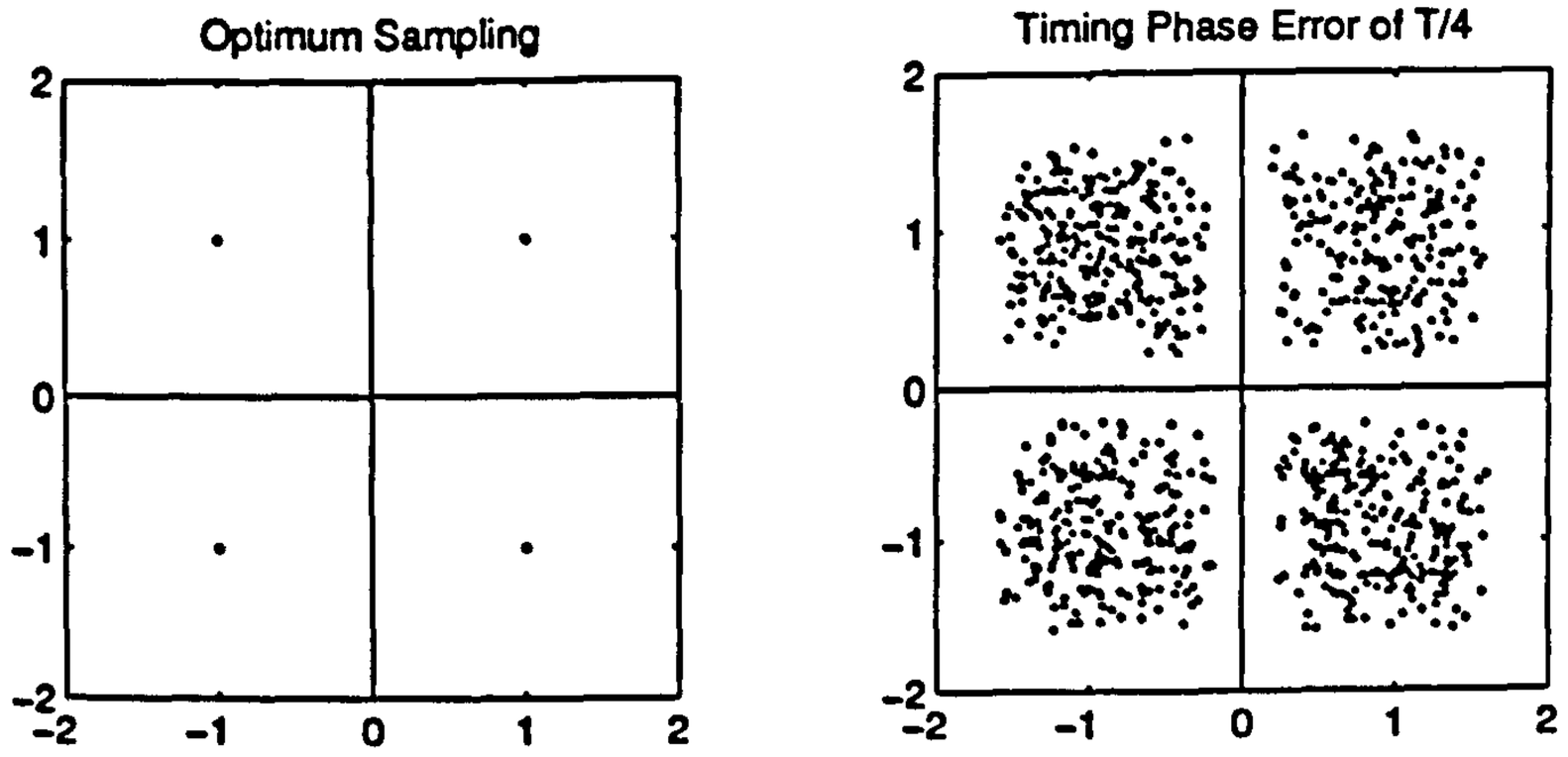



Figure 6.3: Ideal scatter diagram (top) and with a timing error of $T_s/4$. The complex PAM waveform here is 30% Raised-Cosine filtered QPSK.

opening. This is sub-optimal however as is illustrated by examining the scatter diagram of the signal. Figure 6.3 shows the scatter plot of the signal corresponding to that in figure 6.2. In this example the the signal is a 30% Raised-Cosine filtered QPSK data stream and so should give zero inter-symbol interference (being a Nyquist pulse) at the optimum sampling instants. The ideal constellation for QPSK is now perturbed by data-noise resulting in some points being moved closer to the decision boundaries. There is no additive noise due to receiver impairments here, the spreading of the constellation points is solely due to data-noise. One can immediately appreciate that the effect of such sub-optimal sampling is to degrade the error-rate [6] in much the same way that the error rate is worsened by having a carrier phase error in an orthogonal coherent detector, that is, both perturbations act to move the ideal constellation points closer to the decision boundaries. The problem is exacerbated further by having modulation schemes with large alphabets.

Therefore, in asynchronous timing recovery loops to achieve optimum performance we must *reconstruct* the optimum sampling points from our sampled PAM data stream. We shall see below that this is accomplished by interpolation.

### 6.1.1 Digital Modelling of Analogue Methods

As mentioned earlier and in [62], synchronisers that are derived as *digitalised* versions of analogue methods are generally suboptimum. It is, however, entirely possible to model analogue synchronisation methods on a digital computer provided that the sampling frequency in the simulation is chosen to be high enough. An example, say for BPSK, might be to square the input waveform and then track the resulting harmonics with a PLL, passing the output to a nonlinear device to reproduce a synchronous timing wave. We will see that in the true digital implementations of data synchronisers a radically different approach is taken.

Analogue synchronisation methods will not be considered further in this thesis.

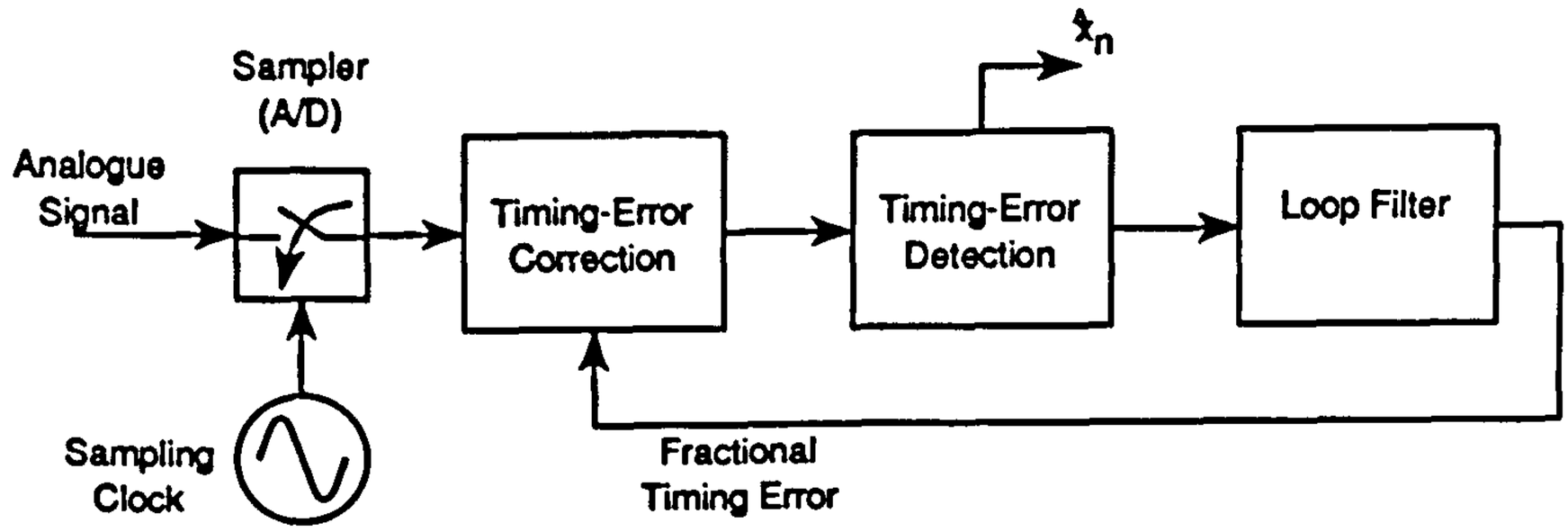



Figure 6.4: Generic model of asynchronous timing-recovery loop.

### 6.1.2 True Digital Implementations – Generic Loop Structure

#### Feedforward (search) vs Feedback (tracker) Configurations

A digital data synchroniser makes no attempt to reproduce a synchronous timing wave at the receiver. The input data stream and the clock in the receiver are considered to be asynchronous. The error in timing between the input data and the data strobe is measured, filtered to produce an error signal and then used to correct *the input data signal*. This correction takes the form of a digital delay element. Fractional delays, for accurate synchronisation, are obtained by using an interpolation filter.

The generic structure of an asynchronous timing recovery loop is as shown in figure 6.4. Such a control-loop is radically different in structure from most of the timing-recovery loops treated in the literature. Generally, the output from the error-detection block is used to change the phase of the sampling clock – not in this case. Note also, that if the timing-error detection block is Decision Directed (DD) then it will also provide the output symbol decisions, $\hat{x}_n$.

In a synchronous timing recovery loop an error-detector would detect the difference in phase between the incoming data and the strobe and the strobe would then be *adjusted* accordingly.

## 6.2 Timing Error Detection

What follows are the basic methods for measuring the timing error in a digital data synchroniser. Non Data-Aided (NDA) methods do not use symbol decisions to derive a timing error signal whereas Decision Directed (DD) methods do. The theory behind Maximum Likelihood Estimation (MLE) states that optimum estimation of the parameters is achieved when all of the parameters are estimated jointly, or knowledge of other parameters are available when estimating another parameter. Thus knowledge of the transmitted data symbols when estimating the timing or carrier phase will improve the estimate — such a scheme is strictly Data-Aided (DA). However, at high signal to noise ratios, the number of symbols received in

error is small and thus DD scheme becomes approximately DA [62]. DD schemes will generally outperform NDA schemes when the signal to noise ratio is high enough. At low signal to noise values, when error-propagation may occur, or whilst the signal is being acquired, NDA schemes may be better.

The schemes are thus characterised in terms of the number of samples they require per symbol and whether they are decision-directed or not. The theoretically optimum is the Mueller & Müller Synchroniser at 1 sample/symbol. We shall now give an overview of a number of timing error detectors common in the literature [9] and then go on to give a unified overview of these schemes when applied to non-offset $M$-PSK modulation.

### 6.2.1 Non Data-Aided Methods

Non data-aided schemes do not rely on a knowledge of the input data stream to form the error signal for use in the timing recovery loop. Such schemes may be of use when the error-rate is high. So it may be that NDA will out-perform DD schemes, in terms of the variance of the timing-estimate, at low $E_b/N_o$ or when a carrier phase error is present. It is reported in [8] that it is possible to recover symbol timing information *before* the carrier has acquired. We shall illustrate this in chapter 8.

#### The Gardner Synchroniser

The Gardner Algorithm [8] is the only NDA algorithm to be studied here. It is based on the simple algorithm

$$e_t(n) = \{x(n) - x(n-2)\}\, x(n-1) \tag{6.1}$$

In the above the $x(\cdot)$ represent the sampled PAM data stream. The block-diagram model for the Gardner Tracker is as shown in figure 6.5. Note that this block works on two samples per symbol. It produces one sample out per symbol which is filtered and then used as an error signal to control the delay element according to figure 6.4.

The S-curve [4] is the error detector characteristic. It shows the expected output, $\mathcal{E}(e_\tau(n))$, from the timing-phase error detector block as a function of the input delay $\tau$. The Gardner error-detector has a simple closed-form solution for the S-curve given by the following [8] :

$$\mathcal{E}(e_\tau(n)) = -\frac{4}{T}\sin\left(\frac{2\pi\tau}{T}\right) \times \int_0^{1/T} G(f)G\left(\frac{1}{T} - f\right)\sin(\pi f T)\, df \tag{6.2}$$

where $T$ is the sample period and $G(f)$ is the frequency pulse corresponding the PAM data stream. Note that the value of the integrand in (6.2) is in fact the gain of the S-Curve and is proportional to the excess-bandwidth of $G(f)$.

A salient feature of the result above is that the *gain* of the S-curve decreases as the Raised-Cosine filter roll-off, $\alpha$. An important consequence of this is that there is a direct conflict between spectral confinement of a particular modulation scheme and its *synchronisability*

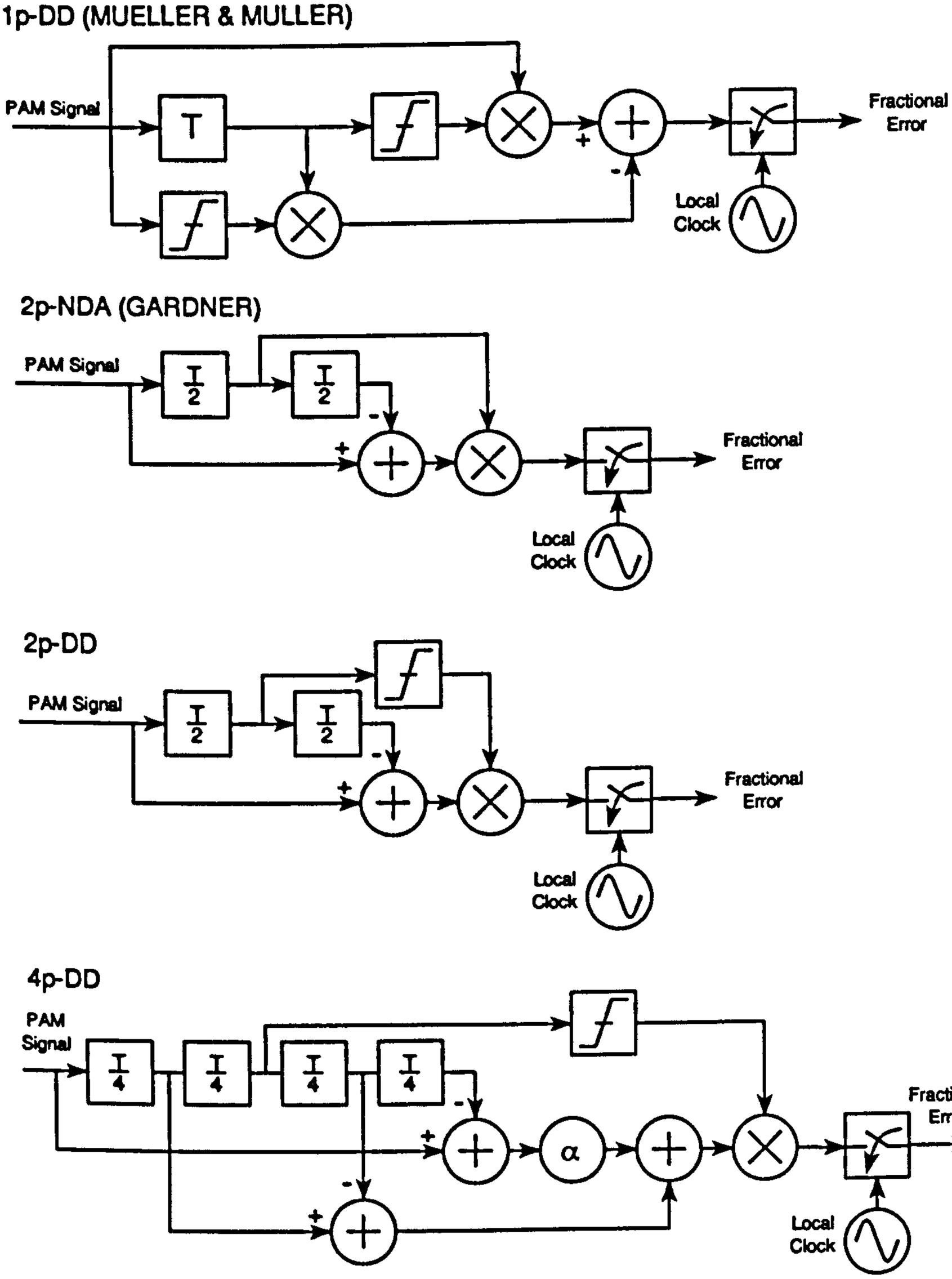



Figure 6.5: Block diagram models of timing-error detectors.

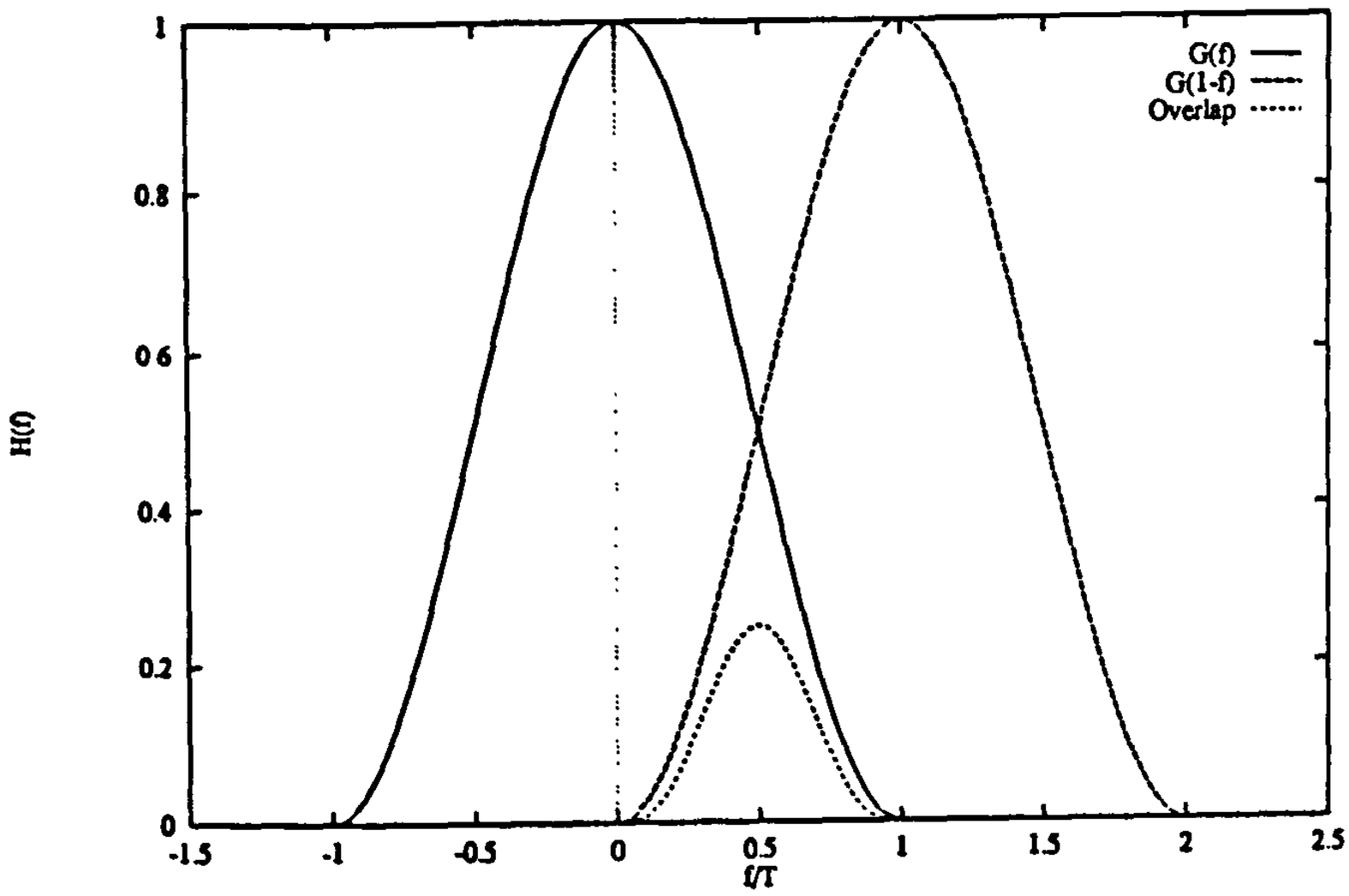



Figure 6.6: Calculation of Gain Factor in Gardner Algorithm for Raised-Cosine Filtered Data Stream.

(when using NDA timing recovery). This fact is often ignored in system design where much of the effort is concentrated on finding spectrally efficient modulation schemes.

The gain of the S-curve in the Gardner algorithm is proportional to the amount of overlap between two frequency pulses $G(f)$ and $G(1-f)$ as shown in figure 6.6 and is consistent with equation (6.2). If we have a knowledge of the filtering of the PAM data stream then we can calculate what the gain factor will be. We shall revisit the detailed derivation of the Gardner algorithm later when we shall present a unified overview of the different timing-error detectors for sampled receivers.

These results justify to some extent what has already been mentioned earlier; that schemes that occupy a narrow bandwidth are more difficult to synchronise than those that do not. This problem with being able to synchronise narrowband schemes could be a debilitating effect in non-stationary channels.

### 6.2.2 Decision Directed Methods

Decision Directed methods actually use data decisions in the error-tracking algorithms. There are four DD schemes that will be covered here.

#### Mueller & Müller Synchroniser

The algorithm for this scheme for the simple case of a binary-antipodal data stream (BPSK) is given by [5]

$$e_t(n) = \hat{x}(n)\,x(n-1) + x(n)\,\hat{x}(n-1) \tag{6.3}$$

In the above the $x(\cdot)$ represent the sampled PAM data stream and the $\hat{x}(\cdot)$ represent receiver decisions. The Mueller & Müller synchroniser is theoretically optimum in the sense that it only requires one sample per symbol. The block-diagram implementation of the Mueller & Müller synchroniser is as shown in figure 6.5. We will show that the gain of the scheme is still a function of the modulation excess bandwidth.

### Two Point Tracker

The two point tracker is similar to the Gardner algorithm except data-decisions are used [62]. It also uses two-points per symbol and is given as follows:

$$e_t(n) = \{x(n) - x(n-2)\}\ \hat{x}(n-1) \tag{6.4}$$

The model for the DD two-point tracker is as shown in figure 6.5.

### Crossing Detector

The crossing detector uses two-points per symbol and is based on the following equation [62].

$$e_t(n) = \{\hat{x}(n) - \hat{x}(n-2)\}\ x(n-1) \tag{6.5}$$

The model for the crossing detector is as shown in figure 6.5. Note that the output from the crossing detector involves two data decisions. The two-point tracker tracks the peak of the input pulse stream whereas the crossing tracker follows the crossing points. This scheme has also been referred to as the Decision-Directed Early-Late (DDEL) synchroniser in [29]. It is intuitively obvious that crossing detectors will always have lower jitter than non-crossing detectors. The Gardner algorithm can be considered to be a NDA crossing detector. A two-point crossing detector, such as the Gardner detector, will output an error signal once per strobe, which is formed by the difference between two sample points which are ideally in the optimum sampling position multiplied by one sample point which is a crossing point. The two samples in the centre of the eye-opening will be ISI free (for a Nyquist pulse) and only the crossing point will contribute to self-noise. Actually, for 100% Raised Cosine filtered data the crossing point is identically zero and so the Gardner algorithm has no self noise here. A non-crossing detector, as would be formed by multiplying the output of the Gardner detector by $-1$, will have more self noise because the error signal is now formed as the difference between two points which are crossing points multiplied by one point which is at the centre of the eye-opening. Hence, a non-crossing detector will have more self-noise than a crossing detector.

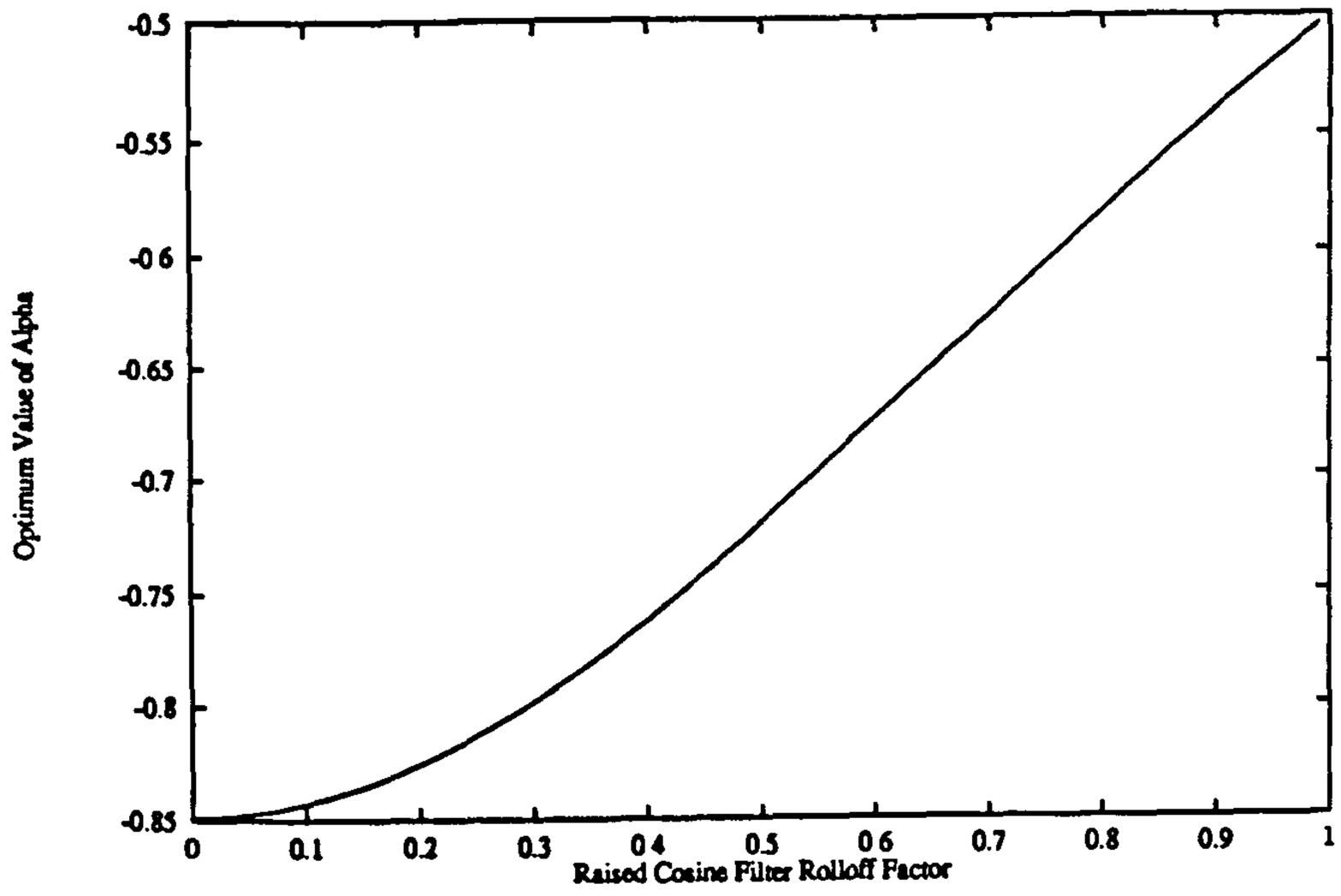



Figure 6.7: Optimum values of $\beta$ as a function of the Raised-Cosine filter roll-off factor, $\alpha$, for the Four Point algorithm.

**Four Point Tracker**

This DD scheme is based on four sample points per symbol [62]. It it based on the following equation:

$$e_t(n) = \{x(n-1) - x(n-3)\}\ \hat{x}(n-2) + \beta\ \{x(n) - x(n-4)\}\ \hat{x}(n-2) \tag{6.6}$$

where $0 \leq \beta \leq 1$.

The model of the four point tracker is as shown in figure 6.5. The value of $\beta$ can be chosen to eliminate the effects of self-noise in the synchroniser [62]. Note that is $\beta = 0$ we simply get the two-point tracker in (6.4). The optimum value of $\beta$ is given by the following equation.

$$\beta = \frac{g\left(\frac{3T}{2}\right) - g\left(\frac{T}{2}\right)}{g(0) - g(2T)} \tag{6.7}$$

where $g(\cdot)$ represents the time-pulse corresponding to the PAM data stream. This choice of $\beta$ eliminates adjacent-pulse self noise for any symmetric $g(t)$ whatever [62]. Equation (6.7) is plotted in figure 6.7 for raised-cosine filtered data and an illustration of this scheme to minimise the variance of the timing estimate by appropriate selection of $\beta$ is shown in figure 6.8.

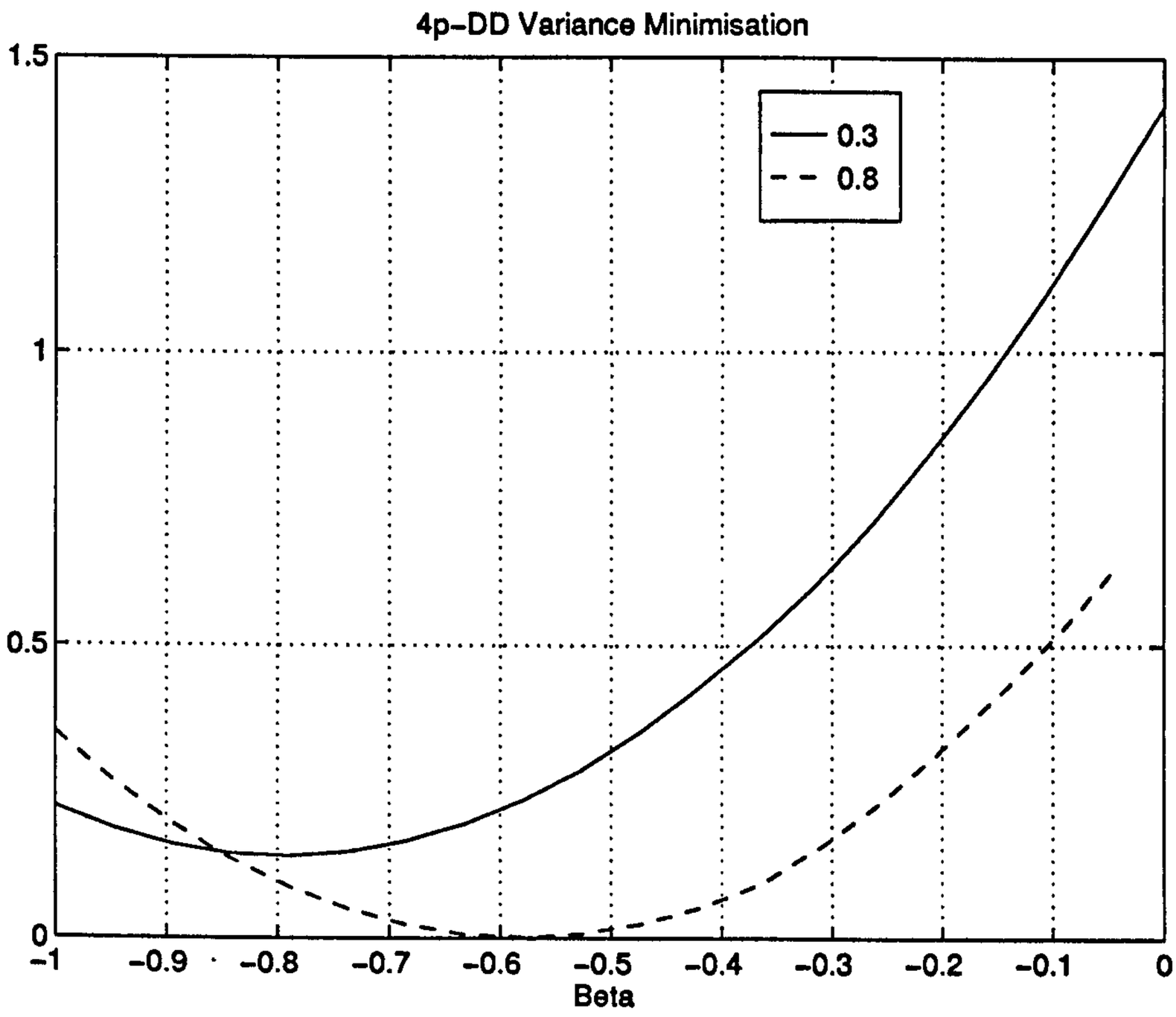



Figure 6.8: Four Point Detector Variance. The sensitivity to $\beta$ is illustrated.

## 6.3 Rationalisation of Timing Error Detection Schemes for Sampled Receivers

In this section we present a novel unified approach to the problem of symbol timing recovery in sampled receivers. In the previous section we considered the problem of symbol timing-error detection as being fundamentally a *real* operation upon a *real* baseband signal. Many authors including Gardner [8] and Mueller & Müller [5] also take this approach. However, when one considers synchronisation of a complex-baseband signal, as delivered to a digital signal processor by an orthogonal (quasi-coherent) detector, then one can consider the function of symbol-timing error detection to be a *complex* operation upon a *complex* signal. It is obvious that the functions of carrier phase and frequency synchronisation are fundamentally complex, but less obvious that complex equivalent timing-error detectors can be derived. By writing the function of timing-error detection of complex PAM modulation schemes as a complex operation we can easily illustrate that the Non data aided (NDA) schemes are carrier phase invariant, whereas the decision directed (DD) schemes are degraded by any carrier phase error. This has the important consequence that DD timing-recovery cannot be used before carrier phase lock. Also carrier recovery schemes optimally require one complex sample per symbol at the correct timing epoch. Thus if timing and phase are recovered in a sequential manner, that is timing first and then phase, then we are unable to use DD symbol timing recovery.

The NDA schemes may be carrier phase invariant but they are not invariant to a carrier frequency-error, as shall be illustrated later.

By considering the function of timing-error detection to be a complex operation on a complex PAM data stream we develop a new class of schemes operating at one and two samples per symbol. We derive the open-loop detector mean (S-curve) for these two new schemes. We refer to these schemes as being "amplitude-directed" (AD) for reasons which will become clear later.

In this section we attempt to rationalise timing-recovery for sampled orthogonal receivers for non-offset complex PAM data streams. We restrict the study to M-PSK type modulation and we consider timing error detectors that utilise a maximum number of two complex samples per symbol. The schemes that we consider are then

- **1p-DD :** One sample per symbol, decision directed, the "Mueller & Müller" algorithm [5].
- **1p-AD :** One sample per symbol, amplitude directed (new).
- **2p-DD :** Two samples per symbol, decision directed, developed by Gardner in [9].
- **2p-AD :** Two samples per symbol, amplitude directed (new).
- **2p-NDA :** Two samples per symbol, Non data aided, the "Gardner" algorithm.

For each of the above timing-error detectors we shall determine analytically the detector S-curve, that is the *mean* voltage out as a function of timing-error in. This is important as

it allows us to determine the closed-loop noise bandwidth which impacts upon the tracking and acquisition performance of the loop. Setting the detector gains equal for each of the timing-error detectors in effect sets the noise bandwidth of the loop to be the same for each loop. Any difference in loop performance will then be attributable to detector *self-noise*. We shall characterise the detectors in terms of their self-noise performance. Also, we shall look at the effect that high-order modulations have on the detector characteristic.

In terms of closed loop performance we are interested in the acquisition and tracking performance of the loop. For small signal offsets we can assume that each timing-error detector is approximately linear and the acquisition and tracking performance using such small signal models is reasonably accurate. However, for large signal offsets and perturbations due to noise we can only determine the performance of the loop by simulation, as we can no longer assume that the loop is linear. Because the detectors are non-linear the performance is difficult to determine analytically; suffice it to say that some of the schemes exhibit a phenomenon known as *hangup* which is the propensity for some schemes to dwell at the incorrect timing position for inordinately long periods of time. Such behaviour may have an impact upon timing error detectors for fading systems, where the data may come out of a fade and the synchroniser would be required to rapidly re-acquire the data. This is particularly true as in such an instance one could not use training pre-ambles for the synchroniser. During tracking one would desire the timing-error detector to have low jitter (self-noise) as this can impact upon symbol error rate. This is particularly true for tightly-bandlimited data where, as we shall see, the symbol timing detectors perform poorly.

## 6.4 Open-Loop Performance : Detector Mean

### 6.4.1 Data Model

We assume that we are attempting to recover timing information from a complex-baseband PAM signal defined by,

$$\begin{aligned} \mathbf{z}(t) &= z_I(t) + j\, z_Q(t) \\ &= \sum_n (a_n + j\, b_n)\, g(t - nT) \\ &= \sum_n \alpha_n\, g(t - nT) \end{aligned} \tag{6.8}$$

In the above we assume that $z_I(t)$, $z_Q(t)$ are the in-phase and quadrature components of $\mathbf{z}(t)$, which in turn is represented as a convolution-sum of the random complex symbols, $\alpha_n$, with the real filter impulse response $g(t)$. In the sequel $\mathbf{z}(t)$ refers to a complex signal and $z(t)$ to a real signal. The equations above thus describe linear, non-offset modulation schemes.

### 6.4.2 Time-Domain Representation of Detector S-Curve Characteristic

#### 1p-DD

Mueller & Müller in their classic paper "Timing Recovery in Digital Synchronous Data Receivers" [5] undertook the first research to address the problem of timing synchronisation in *sampled* receivers. Other work prior to this had dealt with the problem of timing recovery (and other synchronisation functions) but only in an abstract, theoretical way [7]. One sample per symbol timing-recovery schemes are also referred to as "baud-rate" or "T-spaced" timing recovery schemes. Actually the theory expounded by Mueller & Müller in [5] is entirely general – a mechanism is given for deriving baud-rate timing-recovery algorithms using samples from any number of the previous symbols. A simple algorithm appropriate for use with pulse waveforms $g(t)$ that are even-symmetric (as will be delivered by a matched-filter) and Nyquist-1 is a special example of the theory and is as follows,

$$u(\tau,p) = \hat{z}(\tau+(p-1)T)\,z(\tau+pT) - \hat{z}(\tau+pT)\,z(\tau+(p-1)T) \tag{6.9}$$

Actually, so popular is the above algorithm because of its ease of implementation that it is generally referred to as the "Mueller & Müller" algorithm [63, 29]. In (6.9) $p$ refers to the ensemble number. The mean output of the timing-error detector as a function of timing-error, $\tau$, is called the detector S-curve and is found by determining the expected value of $u(\tau,p)$ over all $p$. The S-curve characteristic is needed to set the timing recovery loop parameters (loop noise bandwidth & acquisition). The S-curve characteristic is then defined as follows,

$$U_m(\tau) = \mathbf{E}[u(\tau,p)] = \lim_{P\to\infty}\frac{1}{P}\sum_{p=-P/2}^{+P/2} u(\tau,p) \tag{6.10}$$

We shall start with a time-domain description of the S-curve characteristic and then develop a frequency domain description. The reason for the two approaches is that the expectation is initially done in the time-domain to remove the dependence of the S-curve on the data symbols, assuming that the data is random. The S-curve might then be represented as an arbitrary summation of terms involving the data pulse response $g(t)$. We shall see that the summation extends over an arbitrary number of symbols because the pulse is, theoretically, infinite in the time-domain. It is more usual for us to deal with *band-limited*, rather than *time-limited*, pulses and so a frequency domain description involves the summation over a *definite* number of terms. We thus start with a time-domain derivation of the S-curve characteristic, and, if necessary, develop an equivalent frequency domain description later.

For simplicity we determine the S-curve characteristic of the M&M for real baseband data. Substituting,

$$z(t) = R\sum_n a_n\,g(t-nT)$$

into (6.9) we get,

$$u(\tau,p) = R\sum_n a_n\,g(\widehat{\tau+(p-n-1)T})\sum_n a_n\,g(\tau+(p-n)T) - R\sum_n a_n\,g(\widehat{\tau+(p-n)T})\sum_n a_n\,g(\tau+(p-n-1)T) \tag{6.11}$$

In the above $R$ is the magnitude of the received signal. If we assume data decisions are correct then we can write,

$$u(\tau,p) = R\Big\{a_{p-1}\sum_n a_n\, g(\tau+(p-n)T) - a_p\sum_n a_n\, g(\tau+(p-n-1)T)\Big\} \tag{6.12}$$

We can expand this equation as,

$$\begin{aligned} u(\tau,p) = R\Big\{ & a_{p-1}^2\, g(\tau+T) - a_p^2\, g(\tau-T) \\ & + a_{p-1}a_p\, g(\tau) - a_p a_{p-1}\, g(\tau) \\ & + a_{p-1}\sum_{n\neq p,p-1} a_n\, g(\tau+(p-n)T) - a_p\sum_{n\neq p,p-1} a_n\, g(\tau+(p-n-1)T)\Big\} \end{aligned} \tag{6.13}$$

Only the first line of the above equation contributes to the mean output of the timing-error detector. Thus the detector S-curve for the M&M algorithm is,

$$\boxed{U_m(\tau) = R\Big\{g(\tau+T) - g(\tau-T)\Big\}} \tag{6.14}$$

The second line in (6.13) is obviously zero and so contributes nothing to the expected value or the self-noise. The last line in (6.13) does contribute to the self-noise of the detector for some arbitrary pulse $g(t)$. However, for Nyquist pulses and for $\tau = 0$, that is during tracking, the pulse samples $g(pT) = 0$, $\forall p \neq 0$. So, for this special case the M&M algorithm has the attractive feature that it is entirely self-noise free.

The M&M algorithm can be used for multilevel signals at baseband or can be used with complex baseband signals from an orthogonal coherent detector provided that we have carrier phase (more on this later). The general structure of the M&M algorithm as applied to complex baseband signals is as shown at the top of figure 6.9.

The complex baseband versions of the timing-error detection schemes shown in figure 6.9 are arrived at for non-offset linear modulation schemes by blindly applying the same real timing-error detection functions to the in-phase and quadrature components of the signal and summing the result. This is illustrated for the M&M algorithm as follows,

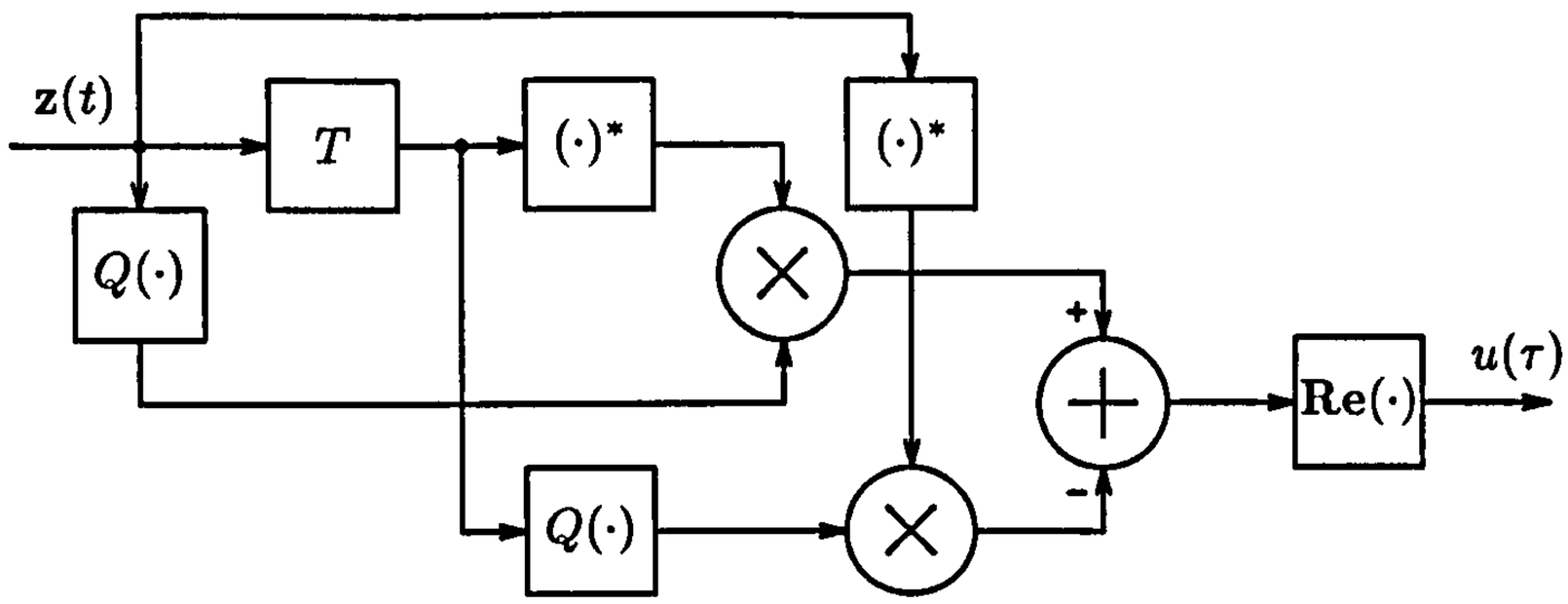
$$\begin{aligned} u(\tau,p) = {} & \hat{z}_I(\tau+(p-1)T)\, z_I(\tau+pT) - \hat{z}_I(\tau+pT)\, z_I(\tau+(p-1)T) \\ & \hat{z}_Q(\tau+(p-1)T)\, z_Q(\tau+pT) - \hat{z}_Q(\tau+pT)\, z_Q(\tau+(p-1)T) \end{aligned} \tag{6.15}$$

which, using the notation in (6.8), can be re-written,

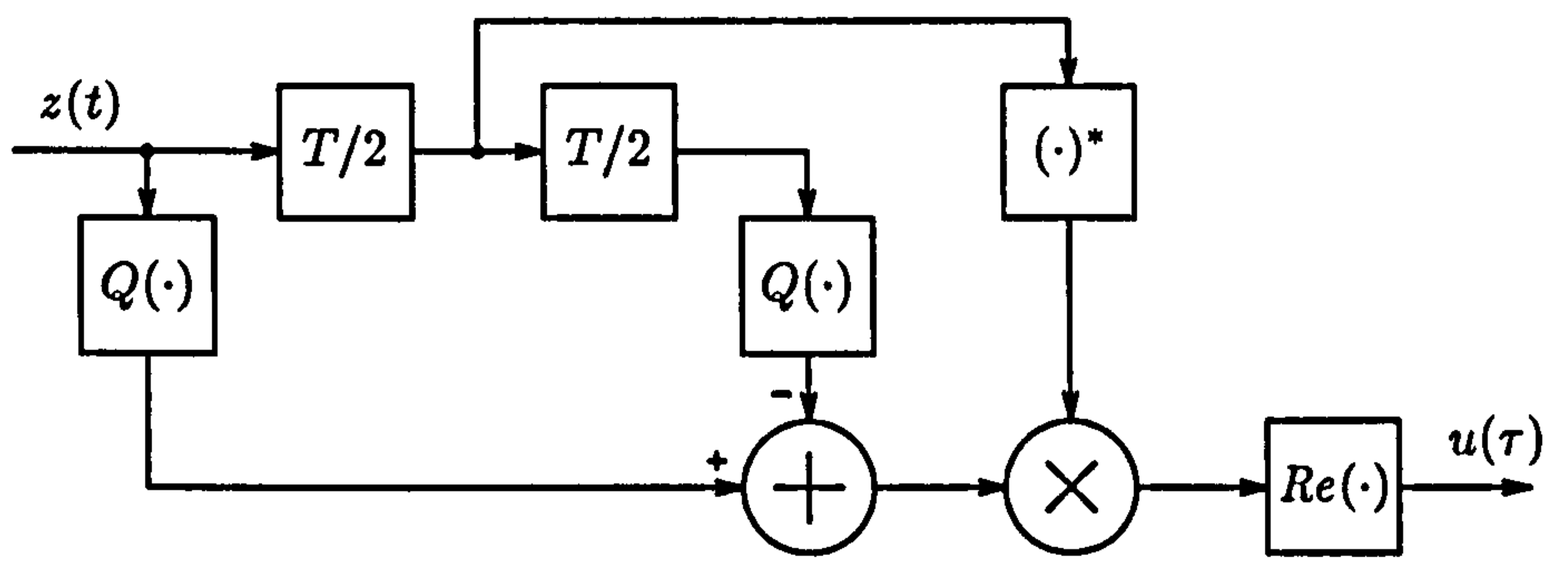
$$u(\tau,p) = Re\Big[\hat{z}^*(\tau+(p-1)T)\, z(\tau+pT) - \hat{z}^*(\tau+pT)\, z(\tau+(p-1)T)\Big] \tag{6.16}$$

where $z^*$ denotes complex conjugate and $Re(\cdot)$ the real component. The $\hat{z}$ is now used to denote complex quantization; a symbol will be returned which is the closest in terms of Euclidean distance from the input in a fully decision-directed implementation.

Using the representation for complex-baseband timing-error detectors as depicted by figure 6.9 allows us to use alternative quantising functions for $Q(\cdot)$ as we shall see. If we define $Q(\cdot)$ to be an MPSK quantiser then we get the same S-curve as in (6.14).




(a) $T$-spaced




(b) $T/2$-spaced

Figure 6.9: Complex baseband representation of timing error detectors suitable for timing extraction from linear non-offset modulation schemes. The quantizing function is denoted $Q(\cdot)$.

### 1p-AD

We now introduce a new method of timing-error detection for non-offset linear modulation schemes by defining the quantising function to be that of the complex signum function as,

$$\begin{aligned} Q(z(\tau)) &= signum(z(\tau)) \\ &= \frac{z(\tau)}{|z(\tau)|} \end{aligned} \tag{6.17}$$

These schemes were developed in a *ad-hoc* manner by reasoning as follows. Decision directed schemes quantise a received signal in both magnitude and phase returning a complex point from the allowable signal constellation. These DD schemes may have attractive features over NDA schemes provided that decisions are correct. If we wish to recover timing before carrier phase lock then we would expect the received signal constellation to be rotated relative to the decision regions. This will result in a timing-error detector with very small gain (as we shall demonstrate later) and consequently poor performance. We can, however, modify the DD timing-error detector in order that it is invariant to carrier phase error by quantising only in magnitude. Hence, these new schemes are referred to as "amplitude directed." The

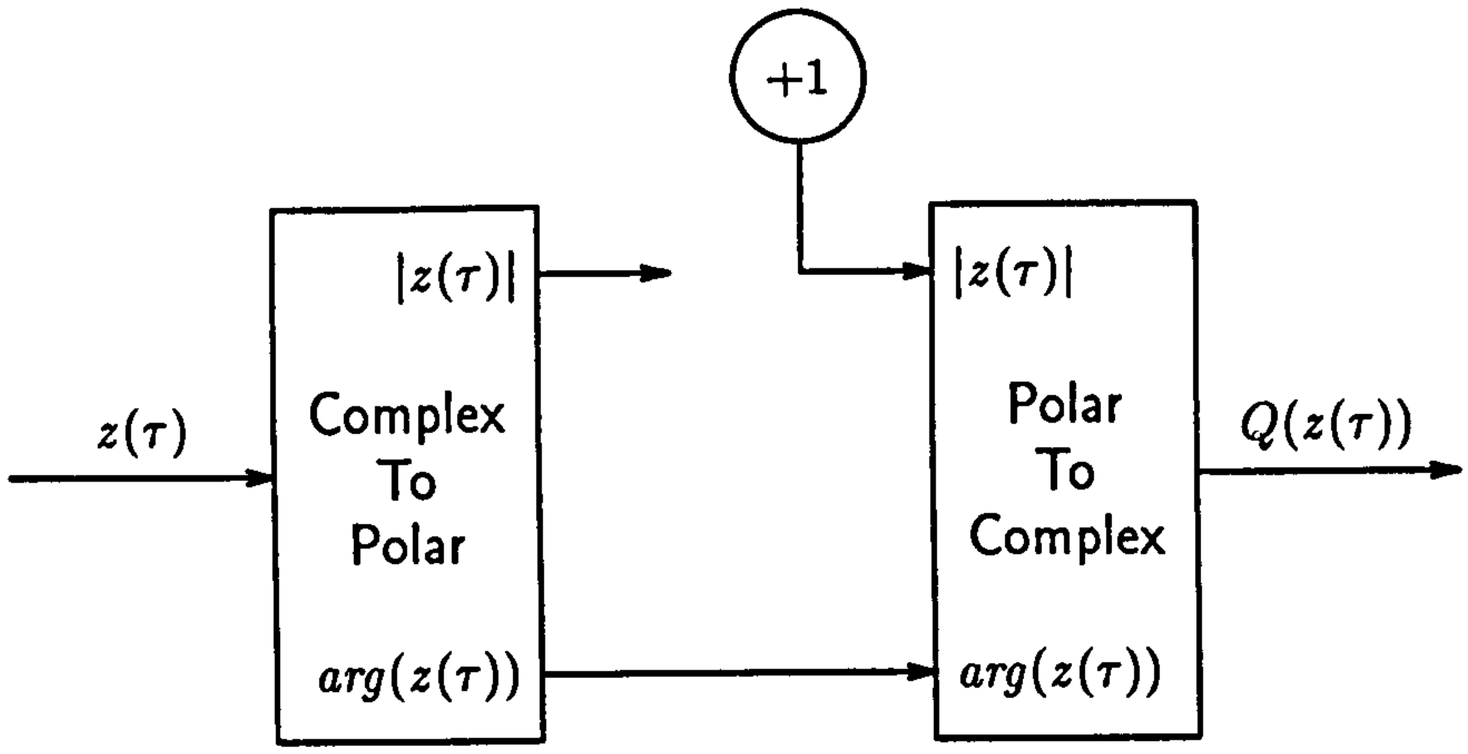



Figure 6.10: Block diagram implementation of signum function

quantising function can alternatively be represented as shown in figure 6.10[1]. The 1p-AD timing-error detector is then given by,

$$u(\tau,p) = Re\left[\frac{z^*(\tau+(p-1)T)\,z(\tau+pT)}{|z(\tau+(p-1)T)|} - \frac{z^*(\tau+pT)\,z(\tau+(p-1)T)}{|z(\tau+pT)|}\right] \tag{6.18}$$

Substituting,

$$z(t) = R\sum_n \alpha_n\, g(t-nT)$$

into (6.18) we get,

$$u(\tau,p) = R \times Re\left[\frac{\sum_n \alpha_n^*\, g(\tau+(p-n-1)T)\sum_n \alpha_n\, g(\tau+(p-n)T)}{\left|\sum_n \alpha_n^*\, g(\tau+(p-n-1)T)\right|} - \frac{\sum_n \alpha_n^*\, g(\tau+(p-n)T)\sum_n \alpha_n\, g(\tau+(p-n-1)T)}{\left|\sum_n \alpha_n^*\, g(\tau+(p-n)T)\right|}\right] \tag{6.19}$$

In order to determine the timing-error detector S-curve one needs to evaluate $E[u(\tau,p)]$ for all possible data ensembles. This is tedious to do directly and so a different approach was taken. By using the algebraic manipulation language MAPLE [60] a conjecture was developed as to what the S-curve would be, and then this conjecture was substantiated later via simulation. The S-curve was determined by using algebraic manipulation as follows,

1. Set the number of symbols per ensemble, $N$, and the number of elements in the signal alphabet, $M$. The total number of ensembles is then $P = M^N$.

[1]This scheme is analogous to the Viterbi & Viterbi feedforward carrier phase synchroniser as we shall see later. The complex data is converted to polar co-ordinates and a nonlinear operation is performed on the signal amplitude. If we take the analogy further $\mathcal{F}(\rho) = \infty$ corresponds to AD whereas $\mathcal{F}(\rho) = \rho$ corresponds to NDA timing recovery. It might be interesting to characterise other schemes depending on $\mathcal{F}(\rho) = \rho^{\|}$ as has been done for the Viterbi & Viterbi algorithm.

2. Form $P \times N$ matrix $a_{p,n}$ containing all possible permutations of $N$ symbols of alphabet size $M$.

3. Specify $p^{th}$ convolution sum as,

$$z_p(t) = \sum_{n=1}^{N} a_{p,n}\, g(t - nT)$$

4. Can use $z_p(t)$ in whatever timing-error detector algorithm wanted, for example,

$$u_p(\tau) = Re\left[\frac{z_p^*(\tau - T)\, z_p(\tau)}{|z_p(\tau - T)|} - \frac{z_p^*(\tau)\, z_p(\tau - T)}{|z_p(\tau)|}\right]$$

5. Then form expectation over P ensembles as,

$$U_m(\tau) = \frac{1}{P} \sum_{p=1}^{P} u_p(\tau)$$

6. $U_m(\tau)$ can be further simplified by applying appropriate assumptions.

Such an approach was used here to arrive at a conjecture that the S-curve for the 1p-AD timing-error detector is the same as that of the 1p-DD scheme for a modulation order $M = 2$ (BPSK) and is given by,

$$\boxed{U_m(\tau) = \frac{R}{2}\Big\{g(\tau + T) - g(\tau - T)\Big\}} \tag{6.20}$$

for $M \geq 4$ (QPSK and higher). The same algebraic-manipulation approach was used to determine the time-domain equations for the 2p-AD scheme as we shall see.

### 2p-DD

The decision-directed algorithm operating at two samples per symbol was originally derived by Gardner in [9]. There he refers to it as a "zero-crossing detector." Actually, all the two sample per symbol schemes considered here are crossing detectors and by referring to figure 6.9 one can see why. During tracking, when $E[u(\tau, p)] \approx 0$, the crossing point will be midway between the two $T/2$ delays as represented at the bottom of figure 6.9 and hence this configuration is a crossing-tracker. The crossing-tracker will work best if the data pulse is even-symmetric. Again by referring to the figure, if the crossing point is midway between the $T/2$ delays and the data pulse $g(t)$ is even-symmetric then the peaks of $g(t)$ (the optimum sampling points) are delivered either side of the $T/2$ delays. A problem does arise, however, when the pulse is not even-symmetric as there will be a bias in the optimum sampling point. If the receiver consists of an ideal matched filter then the pulse $g(t)$ will *always* be even symmetric and so the problem of bias does not arise. For an even number of samples per symbol crossing detectors are always preferred over non-crossing detectors as they have lower self-noise.

The algorithm for the 2p-DD algorithm is then written as,

$$u(\tau,p) = Re\Big[z^*(\tau+(p-1/2)T)\big\{\hat{z}(\tau+(p-1)T)-\hat{z}(\tau+pT)\big\}\Big] \tag{6.21}$$

Using an analogous approach to that for the 1p-DD scheme we find that the S-curve is then

$$\boxed{U_m(\tau) = R\Big\{g(\tau+T/2)-g(\tau-T/2)\Big\}} \tag{6.22}$$

**2p-AD**

The 2p-AD scheme is a new development resulting from this work and uses two samples per symbol and the complex signum as the quantising function. As for the 1p-AD scheme, this scheme is only applicable to timing-error detection of linear non-offset modulation schemes. The 2p-AD timing-error detector is given by,

$$u(\tau,p) = Re\left[z^*(\tau+(p-1/2)T)\left\{\frac{z(\tau+(p-1)T)}{|z(\tau+(p-1)T)|}-\frac{z(\tau+pT)}{|z(\tau+pT)|}\right\}\right] \tag{6.23}$$

By using the same algebraic-manipulation procedure as for the 1p-AD scheme we arrive at the rather surprising result for the S-curve as,

$$\boxed{\begin{aligned} U_m(\tau) = \frac{R}{2}\Bigg\{ & \\ & \frac{\sum_n g(\tau-(n+1/2)T)\,g(\tau-(n+1)T)-\sum_n g(\tau-(n+1/2)T)\,g(\tau-nT)}{\sqrt{\sum_n g(\tau-nT)^2}} \\ & +g(\tau+T/2)-g(\tau-T/2)\Bigg\} \end{aligned}} \tag{6.24}$$

provided $M \geq 4$. If $M = 2$ (BPSK) then the 2p-AD has the same S-curve as the 2p-DD timing-error detector. Note that for this time-domain description the summation is over an arbitrary number of data pulses. This is because we are dealing with pulses which are theoretically infinite in the time-domain. A more succinct result can be arrived at by assuming the pulses are band-limited and converting to the frequency-domain. This shall be illustrated later.

The interesting point about (6.24) is that it can be recognised as a combination of 2p-DD and (as shall be demonstrated next) 2p-NDA S-curves, with the modification factor of,

$$\frac{1}{R\sqrt{\sum_n g(\tau-nT)^2}}$$

We shall see in the results that follow that this new "amplitude-directed" algorithm offers considerable performance improvements over the conventional 2p-NDA algorithm for the same advantages, namely that of carrier phase invariance. This is especially true for small values of excess-bandwidth.

**2p-NDA**

Gardner [8, 9] invented a simple but effective Non data aided timing-error detector based on two samples per symbol. It has been shown in [64] that this timing-error detector could be derived from the maximum-likelihood principle. This does not imply, however, that it is the minimum variance unbiased estimator of timing-error. Re-writing Gardner's algorithm in complex notation to make it consistent with the treatment here we get,

$$u(\tau,p) = Re\Big[z^*(\tau+(p-1/2)T)\Big\{z(\tau+(p-1)T) - z(\tau+pT)\Big\}\Big] \tag{6.25}$$

Taking the expectation as previously we arrive at a time-domain representation of the S-curve characteristic for the Gardner algorithm as,

$$\boxed{U_m(\tau) = R^2\Big[\sum_n g(\tau-(n+1/2)T)\,g(\tau-(n+1)T) - \sum_n g(\tau-(n+1/2)T)\,g(\tau-nT)\Big]} \tag{6.26}$$

The big advantage of NDA and AD timing-error detectors is that they are invariant to carrier phase error as shall be demonstrated next.

### 6.4.3 Effect of Carrier Phase and Frequency Errors Upon Timing-Error Detector Characteristic

We are interested in the possibility of recovering symbol timing before we have carrier-phase lock. The effect that the inevitable carrier phase and frequency errors (relative to a local VCO) will have on the timing-error detectors will now be quantified.

Refer back to figure 6.9. Starting firstly with the "T-spaced" scheme, the output from the timing-error detector can be written as,

$$u(\tau,p) = Re\Big[Q^*(z(\tau+(p-1)T))\,z(\tau+pT) - Q^*(z(\tau+pT))\,z(\tau+(p-1)T)\Big] \tag{6.27}$$

where $Q(\cdot)$ is the general quantising function as considered previously. For the NDA and AD timing-error detectors the quantising function is given as,

$$Q(z(t)) = z(t) \quad \text{(NDA Algorithm)}$$
$$Q(z(t)) = signum(z(t)) \quad \text{(AD Algorithm)}$$

For either of the above schemes it is obvious that,

$$Q(z(t)\exp(j\,\phi)) = Q(z(t))\exp(j\,\phi)$$

Thus, we can show carrier phase invariance for the "T-spaced" algorithms by putting,

$$z(t) \rightarrow z(t)\exp(j\,\phi)$$

in (6.27) above. If the carrier phase-error is static then it is obvious that the NDA and AD schemes will be independent of $\phi$. However, the schemes will not be invariant to a carrier *frequency* error as we now illustrate. Assume that there is a constant frequency error of $f$. Then make the substitution

$$z(t) \rightarrow z(t)\,\exp(j\,\{2\pi f t + \phi_0\})$$

where $\phi_0$ is some arbitrary phase offset. Then (6.27) becomes

$$\begin{aligned} u(\tau,p) = Re\Big[&Q^*(z(\tau+(p-1)T))\,z(\tau+pT)\,e^{(j\,2\pi fT)} - \\ &Q^*(z(\tau+pT))\,z(\tau+(p-1)T)\,e^{(-j\,2\pi fT)}\Big] \\ = Re\Big[&Q^*(z(\tau+(p-1)T))\,z(\tau+pT) - Q^*(z(\tau+pT))\,z(\tau+(p-1)T)\Big]\,\cos(2\pi fT) \end{aligned} \quad (6.28)$$

Thus the gain of the timing-error detector will decrease as a function of the frequency error. When the frequency error is large, say, $f \approx \pm 1/2T$, that is half the baud rate, then the gain of the S-curve for the "T-spaced" schemes will drop to zero.

Now consider the effect of carrier frequency error upon the "T/2-spaced" NDA and AD timing-error detectors (bottom of figure 6.9). Here the algorithms are given by,

$$u(\tau,p) = Re\Big[z^*(\tau+(p-1/2)T)\Big\{Q(z(\tau+(p-1)T)) - Q(z(\tau+pT))\Big\}\Big] \quad (6.29)$$

Again consider the effect of a constant frequency error by putting,

$$z(t) \rightarrow z(t)\,\exp(j\,\{2\pi f t + \phi_0\})$$

in (6.29) above to give,

$$\begin{aligned} u(\tau,p) = Re\Big[&z^*(\tau+(p-1/2)T)\,e^{-j\,\{2\pi f(\tau+(p-1/2)T)+\phi_0\}} \\ &\times\Big\{Q(z(\tau+(p-1)T))\,e^{j\,\{2\pi f(\tau+(p-1)T)+\phi_0\}} - Q(z(\tau+pT))\,e^{j\,\{2\pi f(\tau+pT)+\phi_0\}}\Big\}\Big] \end{aligned} \quad (6.30)$$

which after simplification gives,

$$u(\tau,p) = Re\Big[z^*(\tau+(p-1/2)T)\Big\{Q(z(\tau+(p-1)T)) - Q(z(\tau+pT))\Big\}\Big]\,\cos(\pi fT) \quad (6.31)$$

Note that for the "T/2-spaced" timing-error detectors the characteristic gain will fall to zero as the frequency error approaches the baud rate, that is, $f \approx \pm 1/T$. Therefore, *"T/2-spaced" timing recovery is more tolerant of frequency error than "T-spaced" schemes*; the former allowing timing recovery for frequency errors as large as the baud rate.

It is not entirely clear at first glance what the effect of carrier frequency/phase errors will be on DD timing-error detection. This is because for DD timing-error detection,

$$Q(z(t)\,\exp(j\,\phi)) \neq Q(z(t))\,\exp(j\,\phi)$$

Problems could arise as a result of this in joint timing and phase recovery schemes for the following reasons. When one considers recovery of both timing and phase/frequency jointly the system is no longer described by an S-curve but rather by an "S-surface" parameterised by $(\tau, \phi)$. In order to be sure that there is no problem of false-locking one must ensure that there is no point at which $U_m(\tau, \phi) = 0$ with negative slope. This is difficult to do for the DD case as one must determine what the surface looks like by simulation for all values of $\phi, \tau$. It appears for the NDA and AD cases though that the problem is *separable*, that is to say we can represent the two dimensional characteristic as,

$$U_m(\tau, \phi) = U_m(\tau)\, U_m(\phi)$$

It was demonstrated that $U_m(\phi)$ is sinusoidal with no false lock points (the effect of a frequency error is to simply decrease the gain of the detector). The conclusion is that having a frequency error does not in itself mean that the problem of false locking will necessarily occur for NDA and AD timing-error detectors.

Interestingly this behaviour is mirrored by the maximum-likelihood frequency-error detector where the gain of the detector is, in turn, dependent upon the timing-phase and it too drops to zero as the timing phase approaches $\tau \approx \pm T/2$.

### 6.4.4 Frequency-Domain Representation of Detector S-Curve Characteristic

Refer back to equations (6.24) and (6.26) describing the S-curve characteristic for the 2p-AD and 2p-NDA timing-error detectors. For both detectors the S-curve, $U_m(\tau)$, is described by an arbitrary summation if no assumption is made about the data pulse $g(t)$. It is usual in digital communications, however, to deal with bandlimited pulses. It might typically be assumed that the spectrum is bandlimited to twice the baud-rate. One can then develop a frequency-domain equivalent of the timing-error detector S-curves in (6.24) and (6.26) which are made up of an *exact* number of pulses but this time in the frequency-domain.

This transformation between the time and frequency domains is accomplished by using the Poisson Sum Formula [4],

$$\boxed{\sum_n \alpha(\tau - nT) = \frac{1}{T} \sum_r A\left(\frac{r}{T}\right) e^{+j\,2\pi r\tau/T}} \tag{6.32}$$

where $\alpha(\tau)$ and $A(f)$ are Fourier Transform pairs, such that,

$$\begin{aligned} A(f) &= \mathcal{F}\Big\{\alpha(\tau)\Big\} \\ &= \int \alpha(\tau)\, e^{-j2\pi f\tau}\, d\tau \end{aligned} \tag{6.33}$$

In the following the Inverse Fourier Transform will also be used which is,

$$\begin{aligned} \alpha(\tau) &= \mathcal{F}^{-1}\Big\{A(f)\Big\} \\ &= \int A(f)\, e^{+j\,2\pi f\tau}\, df \end{aligned} \tag{6.34}$$

The Poisson Sum Formula is often used in the analysis of timing recovery schemes [65]. It shall be used here to derive frequency domain equivalents of the 2p-NDA and 2p-AD algorithms. The derivation of the 2p-NDA algorithm was originally performed in [8] by a slightly different and, the author believes, more complicated route than that followed here. This extension to the 2p-AD scheme is wholly original.

Transformation of the S-curves into the frequency domain offers useful insight into the performance of these schemes as a function of the excess bandwidth of the modulation scheme as we shall see. In the following analysis all integrals and summations are assumed to be doubly infinite unless otherwise stated.

**2p-NDA**

We re-state the time-domain S-curve characteristic for the 2p-NDA algorithm as,

$$U_m(\tau) = R^2\Big[\sum_n g(\tau-(n+1/2)T)\,g(\tau-(n+1)T) - \sum_n g(\tau-(n+1/2)T)\,g(\tau-nT)\Big] \tag{6.35}$$

Define the partial terms $\alpha(\tau)$, $\beta(\tau)$ as,

$$\alpha(\tau) = g(\tau-T/2)\,g(\tau-T) \text{ and } \beta(\tau) = g(\tau-T/2)\,g(\tau) \tag{6.36}$$

Then (6.35) can be written,

$$U_m(\tau) = R^2\Big[\sum_n \alpha(\tau-nT) - \sum_n \beta(\tau-nT)\Big] \tag{6.37}$$

This is now in a form required by the Poisson Sum Formula. From (6.32) the Fourier Transforms of $\alpha(\tau)$ and $\beta(\tau)$ also need to be determined. Start by determining $\mathcal{F}(\alpha(\tau))$ as,

$$\begin{aligned} A(f) &= \int \alpha(\tau)\,e^{-j\,2\pi f\tau}\,d\tau \\ &= \int g(\tau-T/2)g(\tau-T)\,e^{-j\,2\pi f\tau}\,d\tau \end{aligned} \tag{6.38}$$

Substituting,

$$g(\tau-T) = \int G(\nu)\,e^{+j\,2\pi\nu(\tau-T)}\,d\nu \tag{6.39}$$

into (6.38) gives,

$$\begin{aligned} A(f) &= \int \left[\int G(\nu)\,e^{+j\,2\pi\nu(\tau-T)}\,d\nu\right] g(\tau-T/2)\,e^{-j\,2\pi f\tau}\,d\tau \\ &= \int G(\nu)\,e^{-j\,2\pi\nu T}\int g(\tau-T/2)\,e^{-j\,2\pi(f-\nu)\tau}\,d\tau\,d\nu \end{aligned} \tag{6.40}$$

Putting,

$$\tau' = \tau - \frac{T}{2} \tag{6.41}$$

into (6.40) gives,

$$\begin{aligned} A(f) &= \int G(\nu)\, e^{-j\,2\pi\nu T} \int g(\tau')\, e^{-j\,2\pi(f-\nu)(\tau'+T/2)}\, d\tau'\, d\nu \\ &= \int G(\nu)\, e^{-j\,\pi(f+\nu)T} \int g(\tau')\, e^{-j\,2\pi(f-\nu)\tau'}\, d\tau'\, d\nu \\ &= \int G(\nu)\, G(f-\nu)\, e^{-j\,\pi(f+\nu)T}\, d\nu \end{aligned} \tag{6.42}$$

Then $A(f)$ can finally be written as,

$$\boxed{A(f) = e^{-j\,\pi fT} \int G(\nu)\, G(f-\nu)\, e^{-j\,\pi\nu T}\, d\nu} \tag{6.43}$$

Now follow a similar procedure for determining $\mathcal{F}(\beta(\tau))$ as,

$$\begin{aligned} B(f) &= \int \beta(\tau)\, e^{-j\,2\pi f\tau}\, d\tau \\ &= \int g(\tau - T/2) g(\tau)\, e^{-j\,2\pi f\tau}\, d\tau \end{aligned} \tag{6.44}$$

Substituting,

$$g(\tau) = \int G(\nu)\, e^{+j\,2\pi\nu\tau}\, d\nu \tag{6.45}$$

into (6.44) gives,

$$\begin{aligned} B(f) &= \int \left[\int G(\nu)\, e^{+j\,2\pi\nu\tau}\, d\nu\right] g(\tau - T/2)\, e^{-j\,2\pi f\tau}\, d\tau \\ &= \int G(\nu) \int g(\tau - T/2)\, e^{-j\,2\pi(f-\nu)\tau}\, d\tau\, d\nu \end{aligned} \tag{6.46}$$

Putting,

$$\tau' = \tau - \frac{T}{2} \tag{6.47}$$

into (6.46) gives,

$$\begin{aligned} B(f) &= \int G(\nu) \int g(\tau')\, e^{-j\,2\pi(f-\nu)(\tau'+T/2)}\, d\tau'\, d\nu \\ &= \int G(\nu)\, e^{-j\,\pi(f-\nu)T} \int g(\tau')\, e^{-j\,2\pi(f-\nu)\tau'}\, d\tau'\, d\nu \\ &= \int G(\nu)\, G(f-\nu)\, e^{-j\,\pi(f-\nu)T}\, d\nu \end{aligned} \tag{6.48}$$

Then $B(f)$ can finally be written as,

$$\boxed{B(f) = e^{-j\,\pi fT} \int G(\nu)\, G(f-\nu)\, e^{+j\,\pi\nu T}\, d\nu} \tag{6.49}$$

Using these derivations for $A(f)$, $B(f)$ together with the Poisson Sum Formula (6.32)

allows the 2p-NDA S-curve to be re-written as,

$$\begin{aligned} U_m(\tau) =& R^2\Big[\sum_n \alpha(\tau - nT) - \sum_n \beta(\tau - nT)\Big] \\ =& \frac{R^2}{T}\left[\sum_k A\left(\frac{k}{T}\right) e^{+j\,2\pi k\tau/T} - \sum_k B\left(\frac{k}{T}\right) e^{+j\,2\pi k\tau/T}\right] \\ =& \frac{R^2}{T}\left[\sum_k \int G(\nu)G\left(\frac{k}{T} - \nu\right) e^{-j\,\pi\nu T}\, d\nu\, e^{-j\,\pi k(T-2\tau)/T} - \right. \\ & \left. \sum_k \int G(\nu)G\left(\frac{k}{T} - \nu\right) e^{+j\,\pi\nu T}\, d\nu\, e^{-j\,\pi k(T-2\tau)/T}\right] \end{aligned} \tag{6.50}$$

Thus far in translating from the time-domain equations (6.35) to the frequency domain (6.50) no assumptions have been made about the frequency pulse $G(\nu)$. Firstly, it is common to say that the pulse is bandlimited to twice the baud-rate and secondly the pulse is even-symmetric. This second assumption arises also because of the theory of matched-filtering; if we assume that $g(t)$ is delivered by a matched filter then it is by definition even-symmetric. These assumptions are written mathematically as,

$$G(\nu) = 0 \quad \text{if} \quad \nu > 1/T \quad \text{or} \quad \nu < -1/T \qquad \text{(Band-Limited)} \tag{6.51}$$

and,

$$G(\nu) = G(-\nu) \qquad \text{(Matched-Filtering)} \tag{6.52}$$

Using these assumptions the summations in (6.50) can be expanded for selected values of $k$.

<u>$k = 0$</u>

$$\begin{aligned} U_m(\tau)|_{k=0} =& \frac{R^2}{T}\left[\int G(\nu)\,G(-\nu)\,e^{-j\,\pi\nu T}\,d\nu - \int G(\nu)\,G(-\nu)\,e^{+j\,\pi\nu T}\,d\nu\right] \\ =& \frac{R^2}{T}\left[\int G(\nu)\,G(-\nu)(e^{-j\,\pi\nu T} - e^{+j\,\pi\nu T})\,d\nu\right] \\ =& -2j\frac{R^2}{T}\left[\int G(\nu)\,G(-\nu)\,\sin(\pi\nu T)\,d\nu\right] \\ =& 0 \end{aligned} \tag{6.53}$$

This integral evaluates to zero by invoking arguments of symmetry; $G(\nu)\,G(-\nu)$ is obviously even and $\sin(\pi\nu T)$ odd, therefore the integral of their product is zero. Now consider the other terms in the summation of (6.50).

$\underline{k=-1}$

$$\begin{aligned} U_m(\tau)|_{k=-1} =& \frac{R^2}{T}\left[\int G(\nu)\,G\left(-\frac{1}{T}-\nu\right)\,e^{-j\,\pi\nu T}\,d\nu\,e^{+j\,\pi(T-2\tau)/T} - \right. \\ & \left. \int G(\nu)\,G\left(-\frac{1}{T}-\nu\right)\,e^{+j\,\pi\nu T}\,d\nu\,e^{+j\,\pi(T-2\tau)/T}\right] \\ =& \frac{R^2}{T}\left[\int G(\nu)\,G\left(-\frac{1}{T}-\nu\right)\,\left(e^{-j\,\pi\nu T}-e^{+j\,\pi\nu T}\right)\,d\nu\,e^{+j\,\pi(T-2\tau)/T}\right] \\ =& -2j\frac{R^2}{T}\left[\int G(\nu)\,G\left(-\frac{1}{T}-\nu\right)\,\sin(\pi\nu T)\,d\nu\,e^{+j\,\pi(T-2\tau)/T}\right] \end{aligned} \tag{6.54}$$

$\underline{k=+1}$

$$\begin{aligned} U_m(\tau)|_{k=+1} =& \frac{R^2}{T}\left[\int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,e^{-j\,\pi\nu T}\,d\nu\,e^{-j\,\pi(T-2\tau)/T} - \right. \\ & \left. \int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,e^{+j\,\pi\nu T}\,d\nu\,e^{-j\,\pi(T-2\tau)/T}\right] \\ =& \frac{R^2}{T}\left[\int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,\left(e^{-j\,\pi\nu T}-e^{+j\,\pi\nu T}\right)\,d\nu\,e^{-j\,\pi(T-2\tau)/T}\right] \\ =& -2j\frac{R^2}{T}\left[\int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,\sin(\pi\nu T)\,d\nu\,e^{-j\,\pi(T-2\tau)/T}\right] \end{aligned} \tag{6.55}$$

Again by considering arguments of symmetry one is able to write,

$$\int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,\sin(\pi\nu T)\,d\nu = -\int G(\nu)\,G\left(-\frac{1}{T}-\nu\right)\,\sin(\pi\nu T)\,d\nu \tag{6.56}$$

The above equations can be combined to give,

$$\begin{aligned} U_m(\tau) =& U_m(\tau)|_{k=-1} + U_m(\tau)|_{k=+1} \\ =& -2j\frac{R^2}{T}\left[\int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,\sin(\pi\nu T)\,d\nu\,e^{-j\,\pi(T-2\tau)/T} - \right. \\ & \left. \int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,\sin(\pi\nu T)\,d\nu\,e^{+j\,\pi(T-2\tau)/T}\right] \\ =& -4\frac{R^2}{T}\sin\left(\frac{2\pi\tau}{T}\right)\int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,\sin(\pi\nu T)\,d\nu \end{aligned} \tag{6.57}$$

So finally, the time-domain equation for the S-curve of the 2p-NDA timing-error detector in (6.26) can be re-written in the frequency domain as,

$$\boxed{U_m(\tau) = -4\frac{R^2}{T}\sin\left(\frac{2\pi\tau}{T}\right)\int_0^{+1/T} G(\nu)\,G\left(\nu-\frac{1}{T}\right)\,\sin(\pi\nu T)\,d\nu} \tag{6.58}$$

Note that the limits of the integral have now been restricted as the integrand only has non-zero value between $\nu = 0$ and $\nu = 1/T$ due to the band-limited definition of $G(\nu)$.

A similar approach can be applied to determine the frequency-domain description of the S-curve for the 2p-AD timing-error detector as follows.

### 2p-AD

Restating the time-domain equation for the S-curve of the 2p-AD timing-error detector (6.24) as,

$$U_m(\tau) = \frac{R}{2}\frac{\sum_n g(\tau-(n+1/2)T)\,g(\tau-(n+1)T) - \sum_n g(\tau-(n+1/2)T)\,g(\tau-nT)}{\sqrt{\sum_n g(\tau-nT)^2}} + \frac{R}{2}\left(g(\tau+T/2)-g(\tau-T/2)\right) \tag{6.59}$$

It can be recognised from the above equation that the numerator represents the 2p-NDA timing-error detector S-curve derived previously. To develop a frequency-domain timing-error detector characteristic one needs to calculate the frequency domain equivalent of the denominator, namely,

$$\sqrt{\sum_n g(\tau-nT)^2} \tag{6.60}$$

This is accomplished by using the Poisson Sum Formula (6.32) as previous. Then,

$$\sum_n g(\tau-nT)^2 = \sum_n \alpha(\tau-nT) \tag{6.61}$$

where,

$$\alpha(\tau) = g(\tau)^2 \tag{6.62}$$

Then,

$$\begin{aligned} A(f) &= \int \alpha(\tau)\,e^{-j2\pi f\tau}\,d\tau \\ &= \int g(\tau)^2\,e^{-j2\pi f\tau}\,d\tau \\ &= \int G(\nu)\int g(\tau)\,e^{-j2\pi(f-\nu)\tau}\,d\tau\,d\nu \\ &= \int G(\nu)\,G(f-\nu)\,d\nu \end{aligned} \tag{6.63}$$

Putting this into the Poisson Sum Formula gives,

$$\sum_n \alpha(\tau-nT) = \frac{1}{T}\sum_k A\left(\frac{k}{T}\right)e^{+j2\pi k\tau/T} \tag{6.64}$$

Again, taking individual terms.

<u>$k=0$</u>

$$\frac{A(0)}{T} = \frac{1}{T}\int G(\nu)^2\,d\nu \tag{6.65}$$

$\underline{k=-1}$

$$\frac{1}{T}A\left(-\frac{1}{T}\right)\,e^{-j\,2\pi\tau/T}=\frac{1}{T}\int G(\nu)\,G\left(-\frac{1}{T}-\nu\right)\,d\nu\,e^{-j\,2\pi\tau/T} \tag{6.66}$$

$\underline{k=+1}$

$$\frac{1}{T}A\left(\frac{1}{T}\right)\,e^{+j\,2\pi\tau/T}=\frac{1}{T}\int G(\nu)\,G\left(\frac{1}{T}-\nu\right)\,d\nu\,e^{+j\,2\pi\tau/T} \tag{6.67}$$

All of the above terms can then be combined to give,

$$\sqrt{\sum_n g(\tau-nT)^2}=\sqrt{\frac{1}{T}\int G(\nu)^2\,d\nu+\frac{2}{T}\cos\left(\frac{2\pi\tau}{T}\right)\int G(\nu)\,G\left(\nu-\frac{1}{T}\right)\,d\nu} \tag{6.68}$$

and finally,

$$\boxed{\begin{aligned}U_m(\tau)=&\frac{-\frac{2R}{T}\sin\left(\frac{2\pi\tau}{T}\right)\int_0^{1/T}G(\nu)\,G\left(\nu-\frac{1}{T}\right)\,\sin(\pi\nu T)\,d\nu}{\sqrt{\frac{1}{T}\int_{-1/T}^{+1/T}G(\nu)^2\,d\nu+\frac{2}{T}\cos\left(\frac{2\pi\tau}{T}\right)\int_0^{+1/T}G(\nu)\,G\left(\nu-\frac{1}{T}\right)\,d\nu}}\\&+\frac{R}{2}\left(g(\tau+T/2)-g(\tau-T/2)\right)\end{aligned}} \tag{6.69}$$

In the final equation above the limits have been applied to the integrals taking into account the band-limited nature of $G(\nu)$. Also it is instructive to note that the integrands consist of the frequency pulse, $G(\nu)$, multiplied by a translated version of itself, $G(\nu-1/T)$. It is therefore only those regions of overlap that contribute to the value of the integral, components outside of the overlap contribute only to self-noise and hence can be rejected. The conclusion of this observation is that what may be considered to be an optimum pulse-shape from the point of view of data-detection will not necessarily be the best choice for symbol-timing recovery. This shall be discussed further when dealing with the subject of self-noise.

### 6.4.5 Specific Timing-Error Detector S-Curves for Raised-Cosine Pulse-Shaping

Here we consider $G(\nu)$ to be that of a Raised-Cosine function. This is defined as,

$$G(\nu)=\begin{cases}G_1(\nu) & \text{if } -f_2\le f<-f_1,\\ 1 & \text{if } -f_1\le f<f_1,\\ G_2(\nu) & \text{if } f_1\le f<f_2,\\ 0 & \text{elsewhere.}\end{cases} \tag{6.70}$$

where,

$$G_1(\nu)=\frac{1}{2}+\frac{1}{2}\sin\left(\frac{1}{2}\frac{\pi(2\nu+1)}{\alpha}\right)\ \text{and}\ G_2(\nu)=\frac{1}{2}-\frac{1}{2}\sin\left(\frac{1}{2}\frac{\pi(2\nu-1)}{\alpha}\right) \tag{6.71}$$

and,

$$f_1 = \frac{1}{2} - \frac{1}{2}\alpha \text{ and } f_2 = \frac{1}{2} + \frac{1}{2} \tag{6.72}$$

The impulse response of $G(\nu)$ is,

$$g(t) = \int G(\nu)\, e^{j\,2\pi\nu t}\, d\nu = \frac{\sin(\pi t)}{\pi t} \frac{\cos(\alpha\pi t)}{1 - 4t^2\alpha^2} \tag{6.73}$$

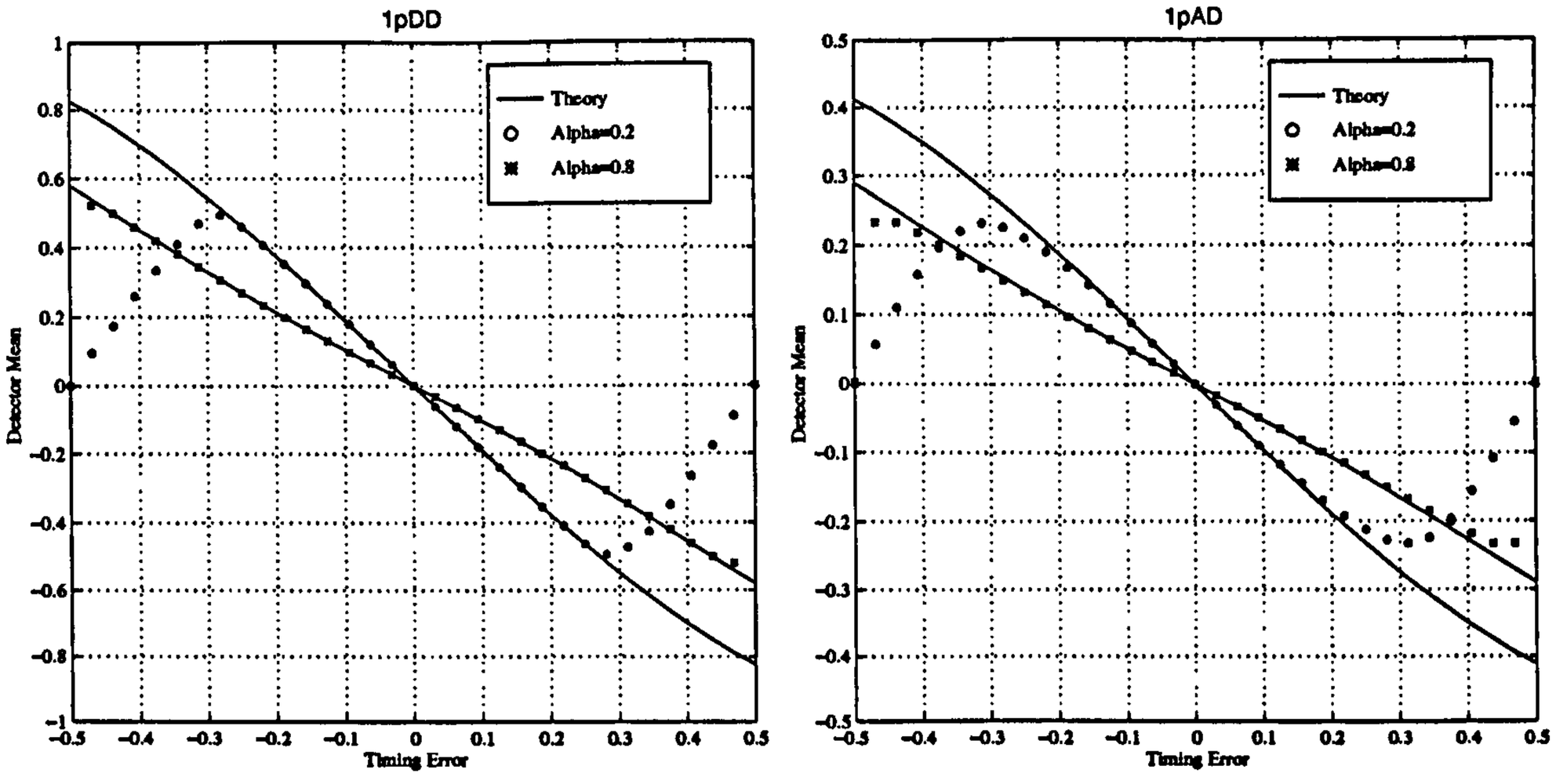
In the above we have normalised the baud rate to unity ($T = 1$) and $\alpha$ is the excess bandwidth. We are interested in the specific analytical solution for $U_m(\tau)$ for $G(\nu)$ as defined above. In order to design the timing-recovery control-loop we need to then determine the small-signal timing-error detector gain[2]. This is defined as,

$$K_{\tau\tau}(\alpha) = \lim_{\tau \to 0} \frac{\partial}{\partial \tau} U_m(\tau) \tag{6.74}$$

The linearised loop gain is shown to be a function of the excess bandwidth $\alpha$. We shall compare the gains of these different detectors as a function of the excess bandwidth shortly. Closed-form analytical solutions to the timing-error detector S-curves for the 1p-DD, 1p-AD, 2p-DD, 2p-AD & 2p-NDA schemes are readily available by inserting the above definitions for $G(\nu)$ and $g(t)$ into equations (6.14),(6.20),(6.22),(6.58), & (6.69) respectively. The results are too long to show here but in the next section we compare these closed-form analytical solutions to the timing-error detector S-curves obtained by an open-loop simulation using MATLAB.

### 6.4.6 Open-Loop Simulation vs Theory

To substantiate the previous analysis an open-loop simulation program was written in MATLAB to determine the timing-error detector S-curve characteristics. The program is as shown in appendix B. The results are shown in figure 6.11 for the two "T-spaced" schemes and in figure 6.12 for the three "T/2-spaced" schemes. The modulation scheme is Raised-Cosine filtered QPSK with excess bandwidth $\alpha$ and there is no additive noise in the system. The timing-error detector characteristic is generated for two values of excess bandwidth $\alpha = 0.2$ and $\alpha = 0.8$ by the simulation program, the results being shown as points in the plot. The solid curves represent the analytical results derived previously. Note that there is close agreement between the theoretical predictions and simulation for the NDA scheme, whereas for the other schemes the agreement is only close for small values of timing-error. This is because in the DD and AD schemes assumptions have been made with regard to the data decision in order to derive analytically the S-curve characteristics, and these assumptions become inaccurate as the timing-error becomes larger. This is even more true for tightly bandlimited signals, that is, there is less correspondence between theory and simulation for $\alpha = 0.2$ than $\alpha = 0.8$. This is not important, however, as the analysis is needed to predict only the small-signal gain of each of the detectors.




(a) Decision Directed

(b) Amplitude Directed

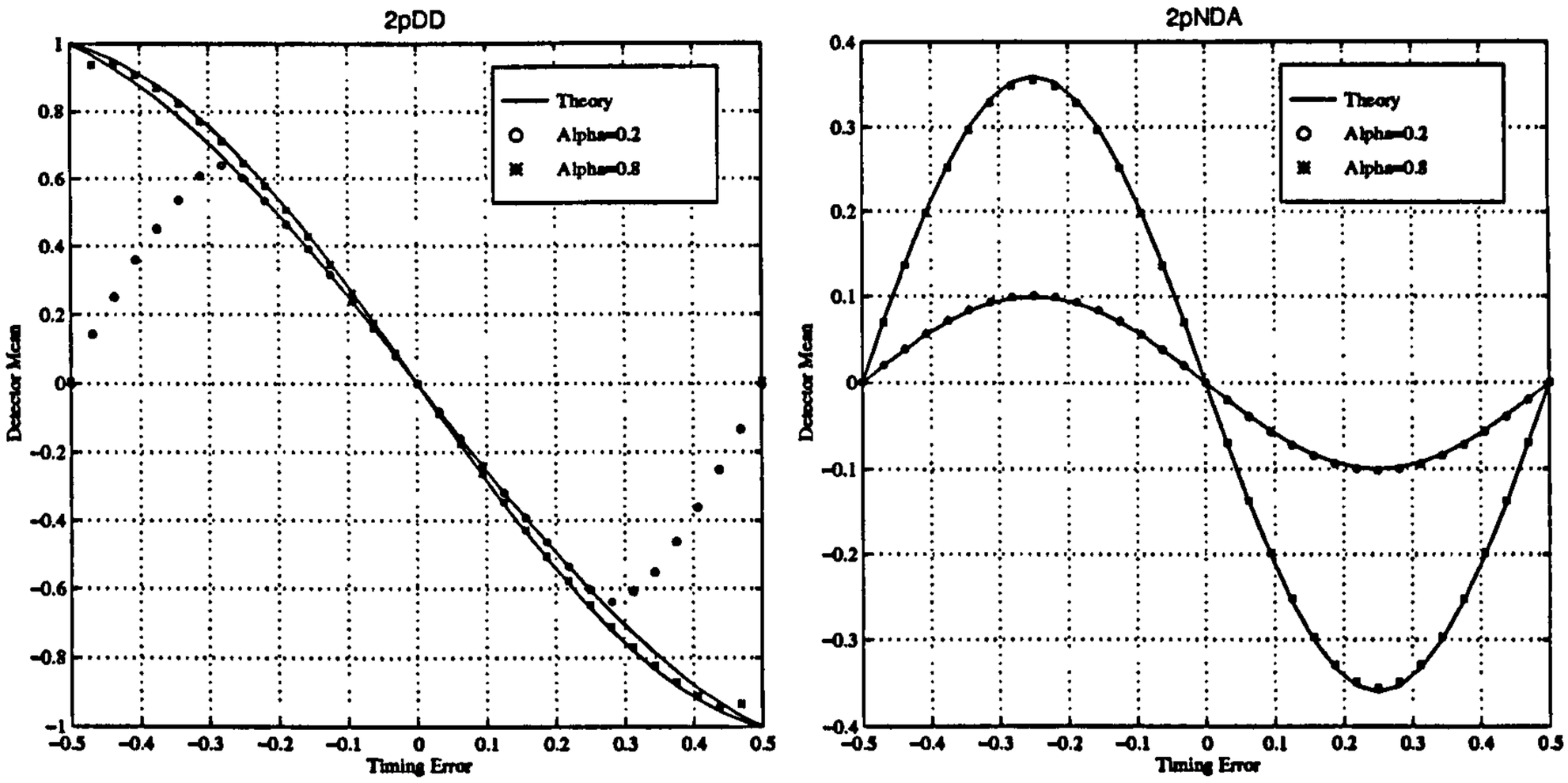
Figure 6.11: Timing-Error Detector S-Curves for $T$-Spaced Schemes

It is worth noting some important features of the detector S-curves. The 2p-NDA Gardner scheme has a sinusoidal detector characteristic. This will mean that the detector output will be small when the timing-error is $\tau \approx \pm T/2$. We shall see later that this will cause the loop to *hangup* when the initial timing-error is large. The other schemes do not suffer from this as their S-curve characteristics become discontinuous at $\tau \approx \pm T/2$. This lack of hangup is particularly useful in the AD schemes as these, like the NDA schemes, can be use prior to carrier phase lock. A scenario where this resistance to hangup may become important is when the timing-error detector has to operate on fading signal. Under such circumstances the symbol-timing recovery loop will have to quickly re-acquire the signal as it comes out of a fade.

It is not only the detector S-curve, which is the mean of the detector output, that determines whether a paticular scheme is prone to hangup, but one must also consider the detector variance, or jitter, as a function of timing error. If the mean and the variance of the detector output is small then the scheme is more likely to hangup than if there is some jitter to knock the loop into lock.

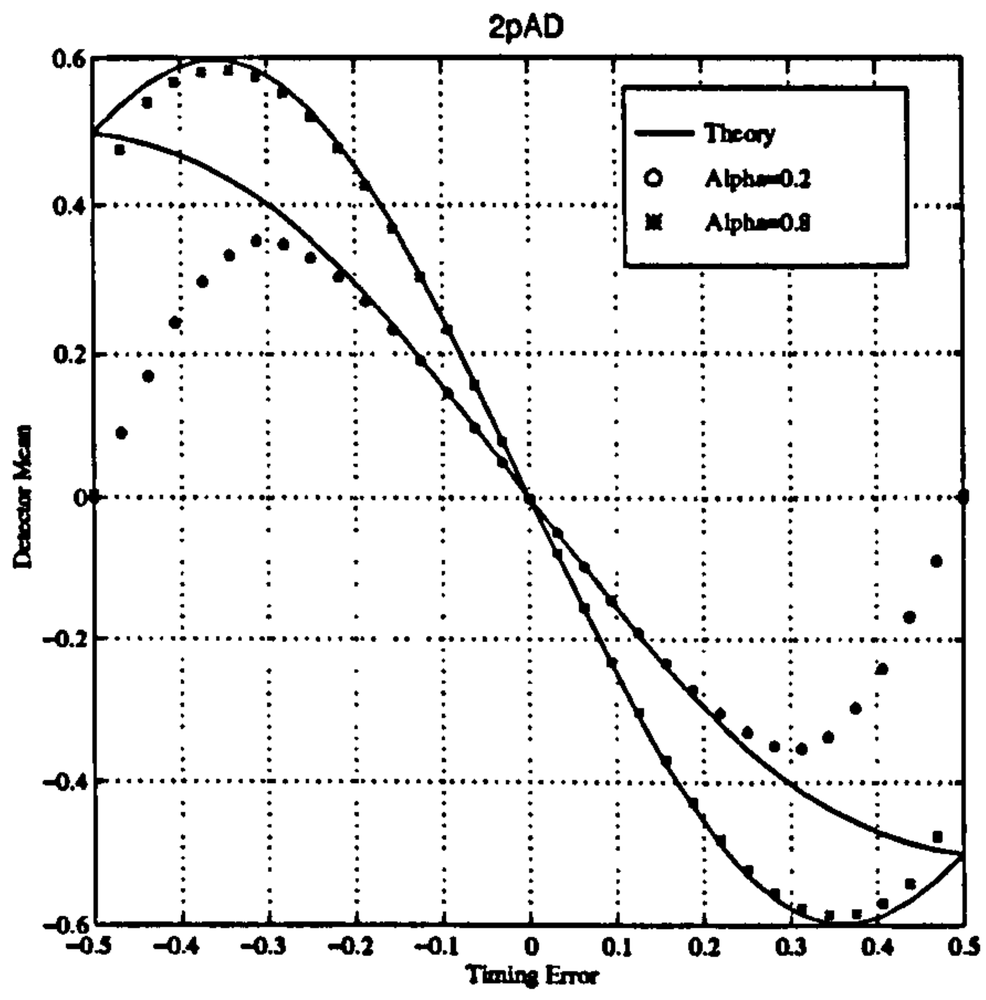
Another feature of the detector S-curve characteristics shown in figure 6.11 and 6.12 is that one can see the effect that the excess bandwidth has on the detector S-curve. The $T/2$-spaced schemes all operate better at larger values of excess bandwidth whereas the opposite is true for the $T$-spaced schemes. We can make this dependency clear by plotting the gains of each of the detectors as a function of excess bandwidth as shown in figure 6.13. These

[2] As the the frequency-locked loops previously we needed to determine $K_{vv}$.




(a) Decision Directed

(b) Non data Aided




(c) Amplitude Directed

Figure 6.12: Timing-Error Detector S-Curves for $T/2$-Spaced Schemes

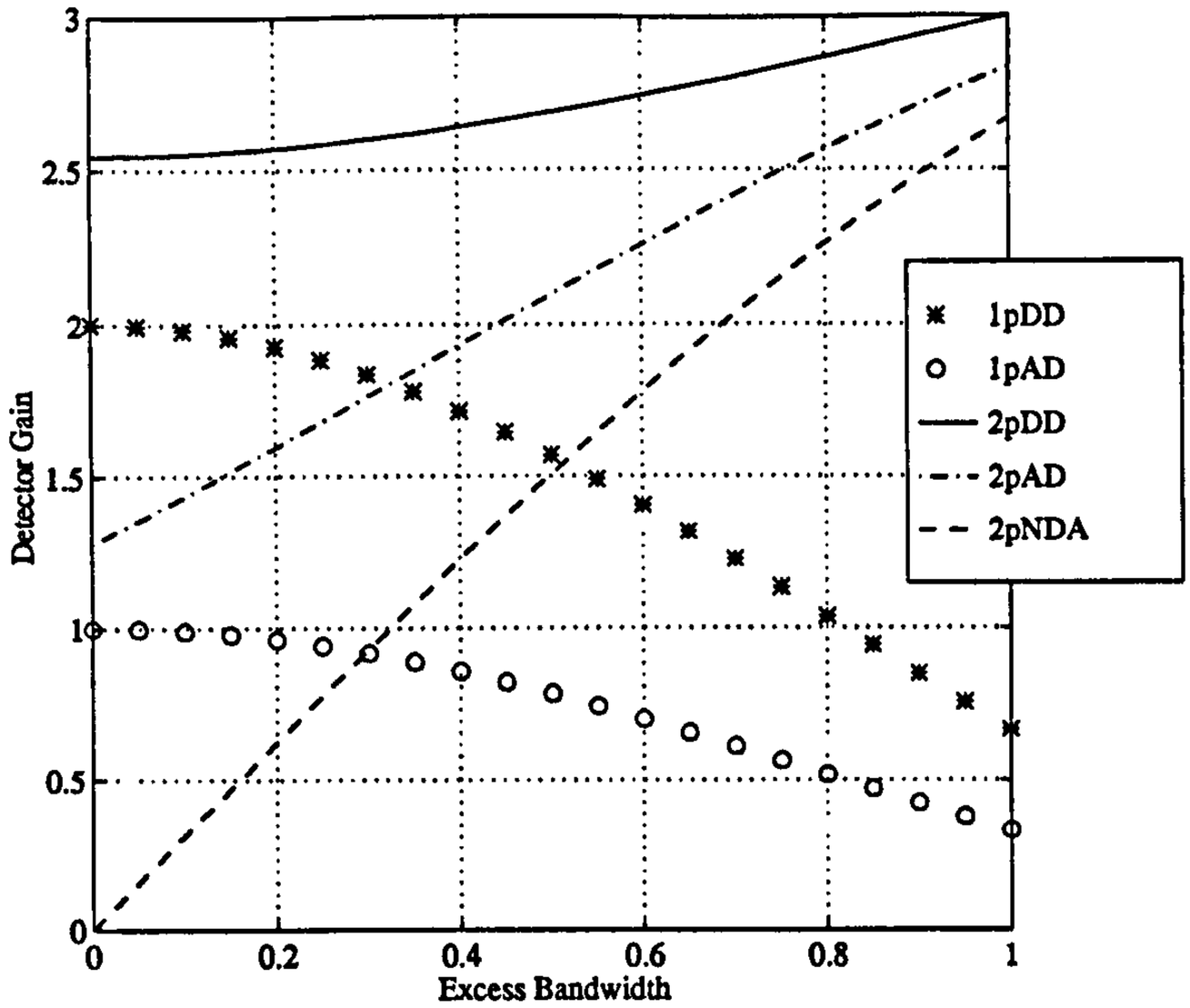



Figure 6.13: Gain of the timing-error detectors as a function of excess bandwidth

results were derived analytically by differentiating the detector characteristic derived earlier. Note that the gain of the 2p-NDA scheme drops to zero as the excess bandwidth drops. This suggests that the 2p-NDA scheme will perform poorly if the excess bandwidth is small. Closed form solutions are available for all of the timing-error detectors operating with Raised-Cosine filtered data. The gains for the 1p-DD and 2p-NDA schemes are quite simple and are therefore worth writing down as

$$K_{1pDD}(\alpha) = 2\,\frac{\cos(\pi\alpha)}{4\alpha^2 - 1} \qquad \text{and} \qquad K_{2pNDA} = 8\,\frac{\sin\left(\frac{\pi\alpha}{2}\right)}{4 - \alpha^2} \tag{6.75}$$

The timing-error detector gains for each of the five schemes for excess bandwidths of $\alpha = 0, 0.5, 1$ are

| | $\alpha = 0$ | $\alpha = 1/2$ | $\alpha = 1$ |
|---|---|---|---|
| 1p-DD | 2 | $\pi/2$ | 2/3 |
| 1p-AD | 1 | $\pi/4$ | 1/3 |
| 2p-DD | $8/\pi$ | $\frac{4\sqrt{2}(3\pi+4)}{9\pi}$ | 3 |
| 2p-AD | $4/\pi$ | $\frac{2\sqrt{2}(20+27\pi)}{45\pi}$ | 17/6 |
| 2p-NDA | 0 | $\frac{16\sqrt{2}}{15}$ | 8/3 |

### 6.4.7 Effect of Perturbations Upon Timing-Error Detector S-Curve Characteristic

It was demonstrated previously that the NDA timing-error detector scheme was not affected by carrier phase error. This is not, however, the case for the decision-directed schemes. The effect of a carrier-phase error is to move the constellation points closer to the decision-regions which will increase the error-rate, assuming that signal is corrupted by some noise, which will in turn reduce the amplitude of the timing-error detector characteristic. The relationship between detector characteristic and signal to noise is somewhat involved to derive but has been undertaken in [66].

The simulated effect that a static carrier-phase error has on the detector S-curves is as shown in figure 6.14 for QPSK modulation with excess bandwidth $\alpha = 0.2$. All the schemes except the 2pNDA scheme, which is invariant to a carrier phase error, are shown. One can see that for small values of timing error the S-curves are only perturbed slightly. Note that the new AD schemes are not perturbed at all by a carrier phase error, justifying their use in place of the conventional NDA scheme.

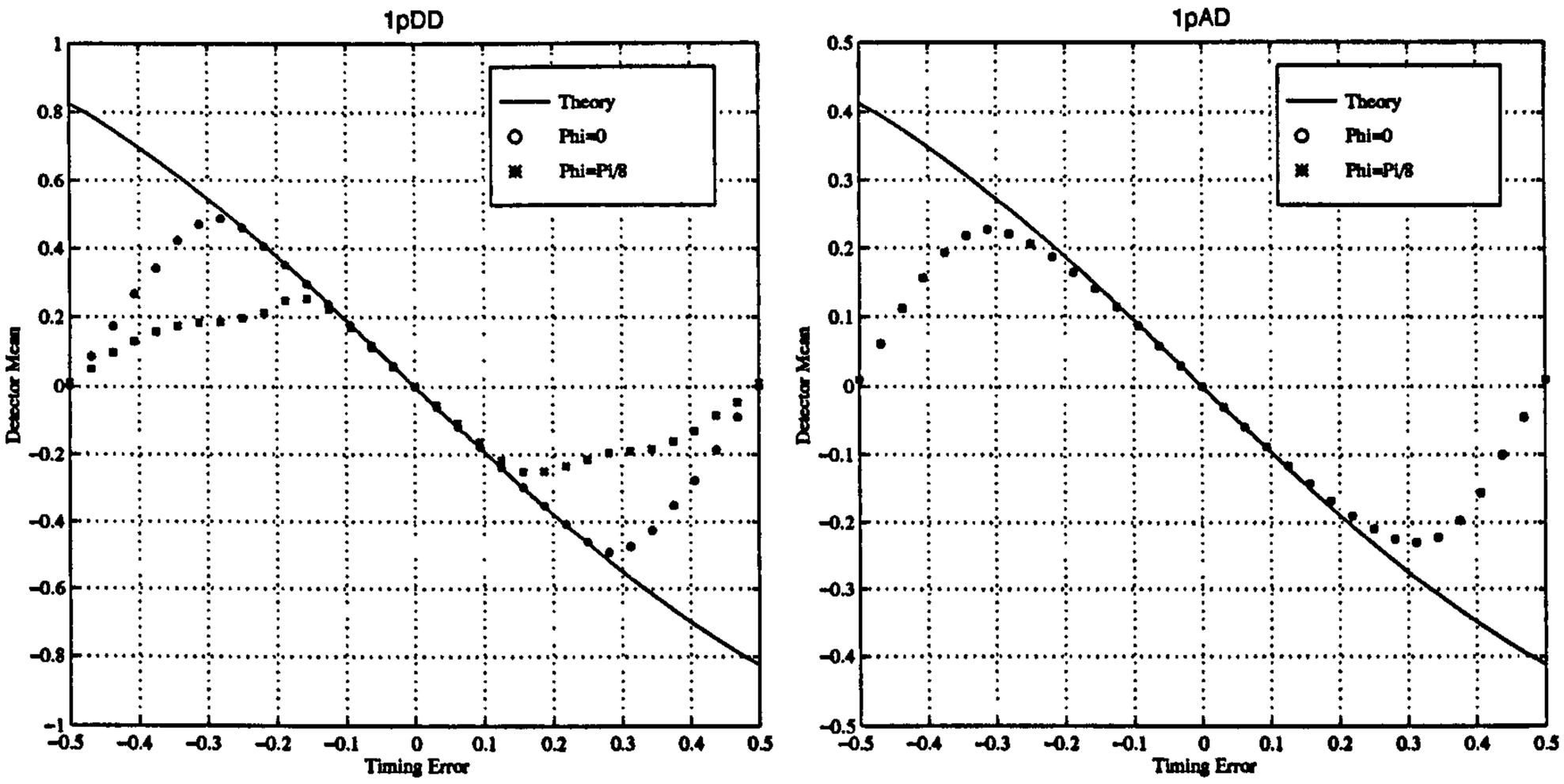
One can alternatively consider the timing-error detector characteristic to be an "S-Surface" as is shown in figure 6.15 for the 2pDD scheme.

The timing-error detector characteristic will also be influenced by the size of the alphabet in the $M$-PSK modulation. If $M$ is large and the signal excess bandwidth is small then one only needs a small timing error before their is considerable inter-symbol interference resulting in a reduction of the timing-error detector gain. This is illustrated in figure 6.16 for tightly bandlimited ($\alpha = 0.2$) QPSK and 16-PSK. Again, the AD schemes are not perturbed by the increase in the signal alphabet unlike the DD schemes.

## 6.5 Open-Loop Performance : Detector Self-Noise

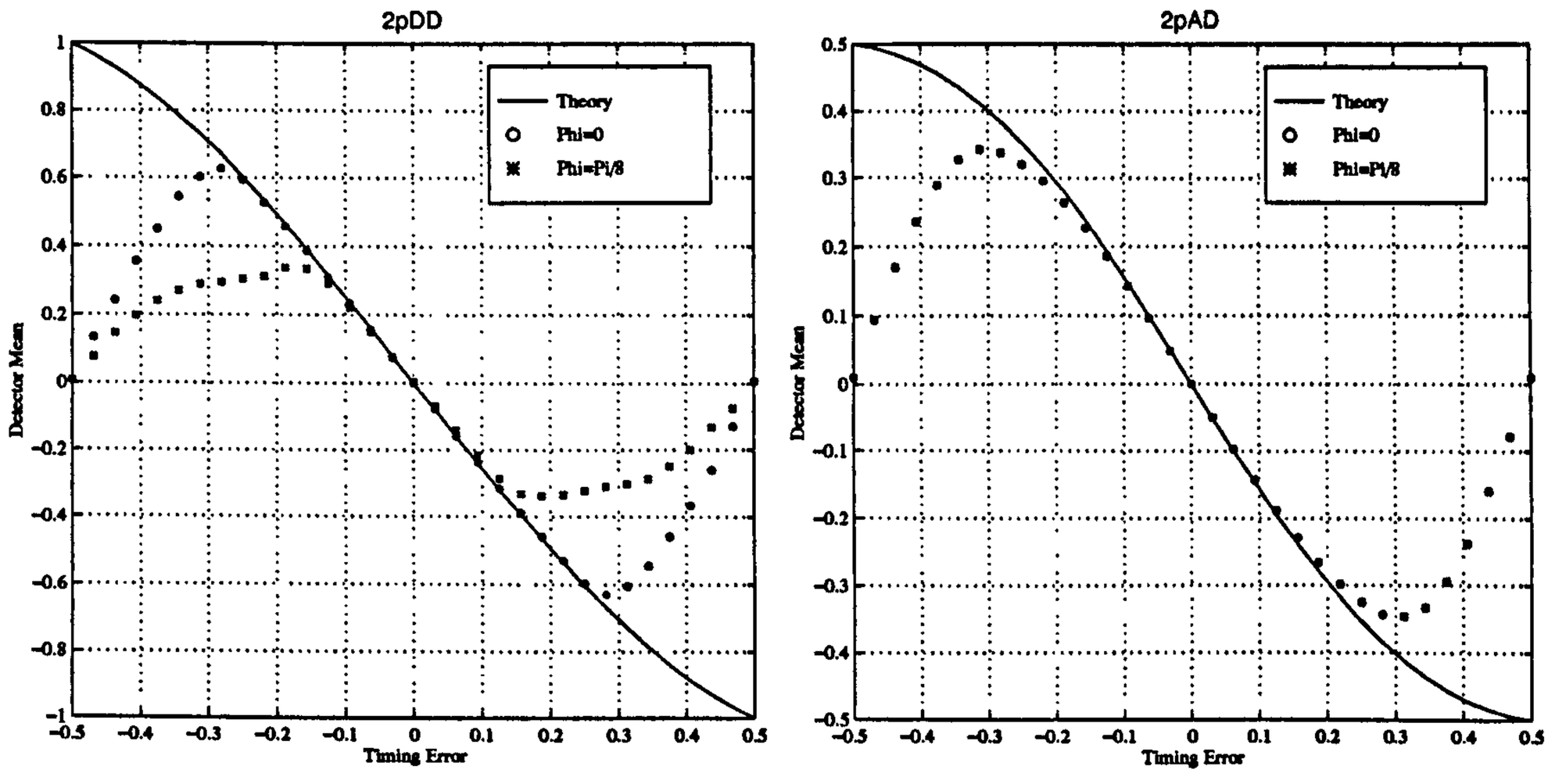
Not only is the mean value of the timing-error detector important in determining the system performance but also the open-loop variance. In fact, a figure of merit for all synchronisers, not only symbol synchronisers, is the ratio $\sigma^2/K_{\nu\nu}^2$, in other words the measured open-loop jitter normalised to the square of the open-loop gain [29]. Determining the self-noise in synchronisers using analytical methods alone is very difficult if not impossible [67]. We can determine the open-loop jitter via simulation quite easily using the program in appendix B. The timing jitter for the decision-directed and non data-aided schemes for both crossing and non-crossing detectors is as shown in figure 6.17. This figure shows the timing jitter of the detectors after normalisation by the detector gain at $\tau = 0$. It can be seen that the crossing detectors perform consistently better than do the non-crossing detectors.

A methodology for determining closed-loop tracking performance from open-loop measurements has been discussed in the ESTEC report in [29]. An attempt has been made as part of this work to substantiate the results derived in [29] but thus far without much success. Part of the problem is that the method of modelling the small-signal tracking performance of a synchroniser with self-noise discussed in [29] appears nowhere else in the open litera-




(a) Decision Directed

(b) Amplitude Directed




(c) Decision Directed

(d) Amplitude Directed

Figure 6.14: Effect of Carrier Phase Error on Timing-Error Detector Characteristic

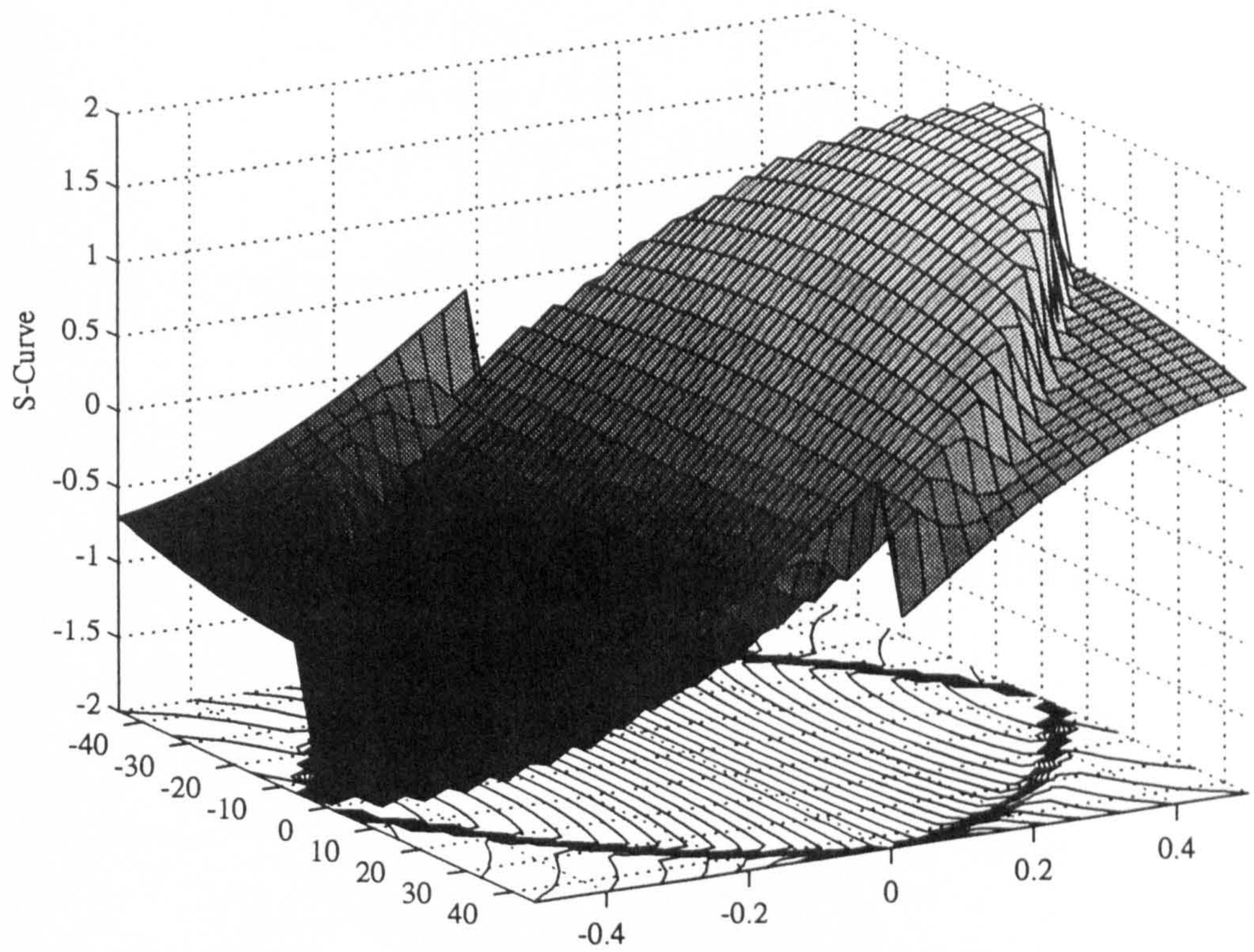
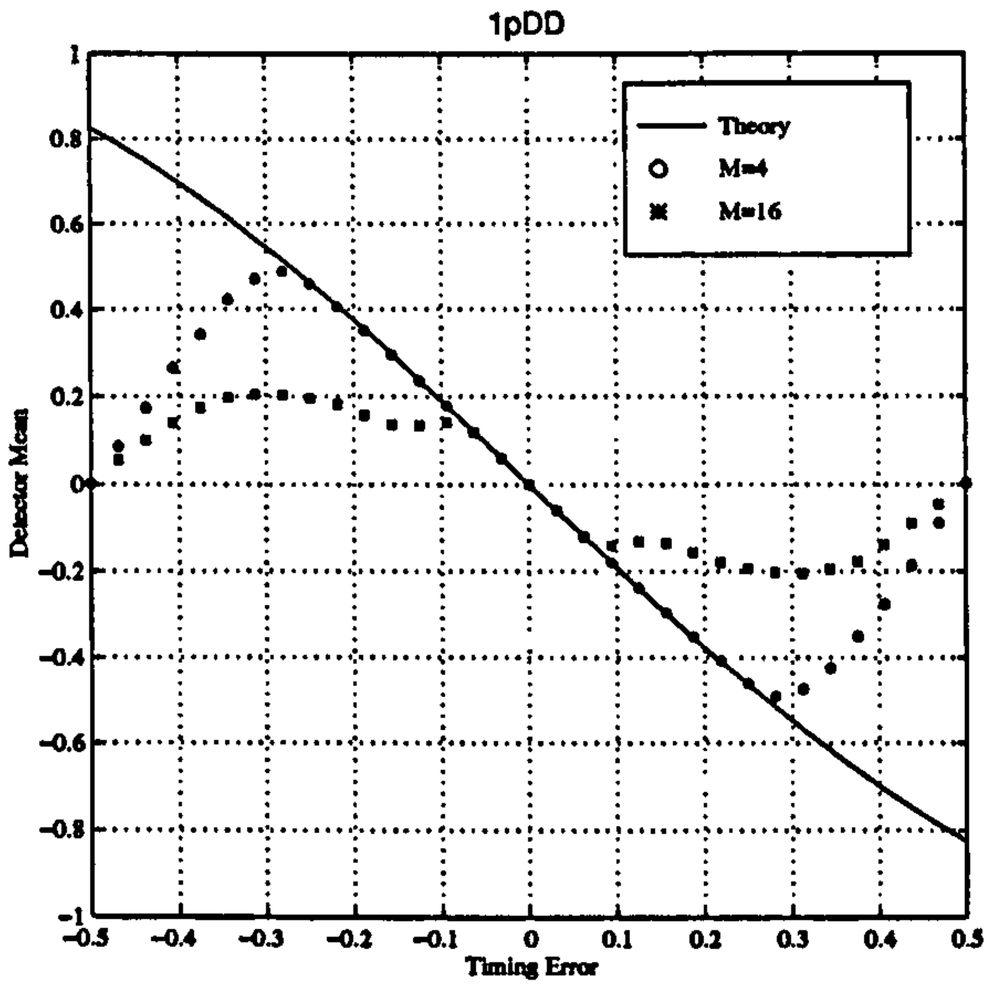
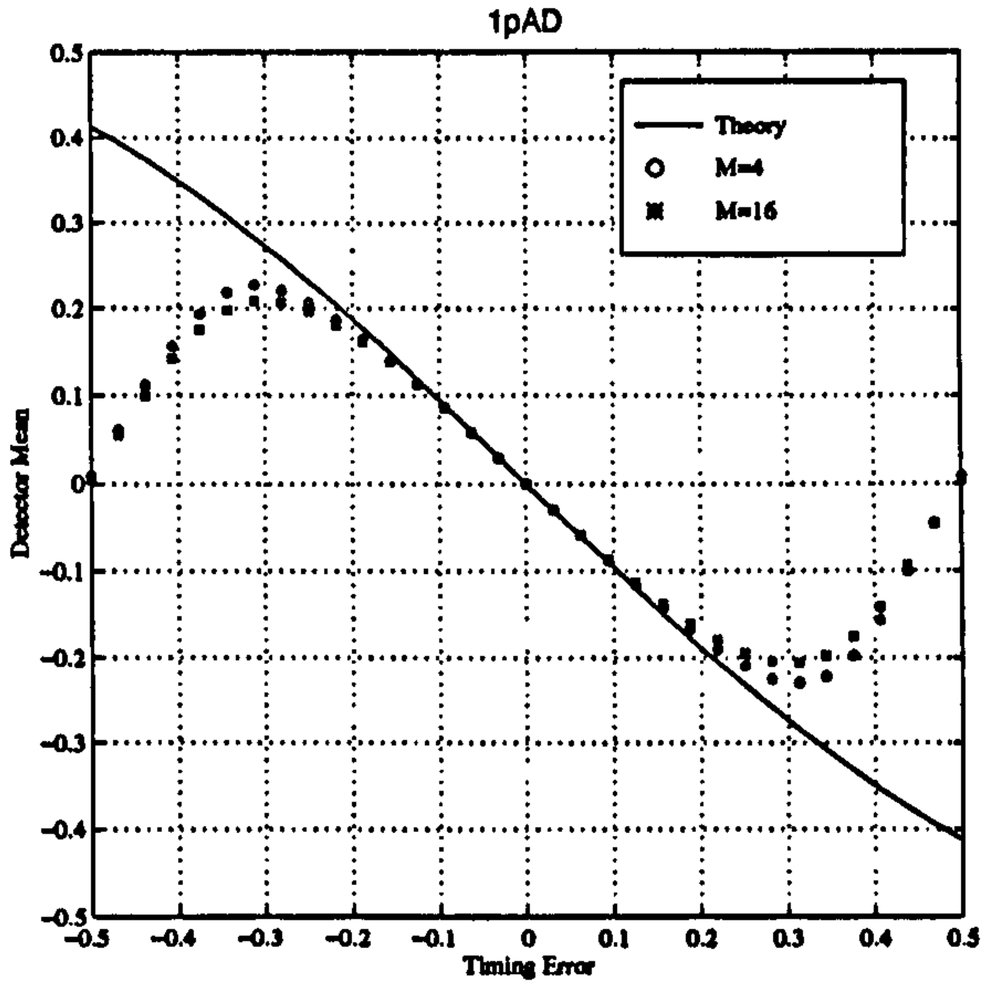


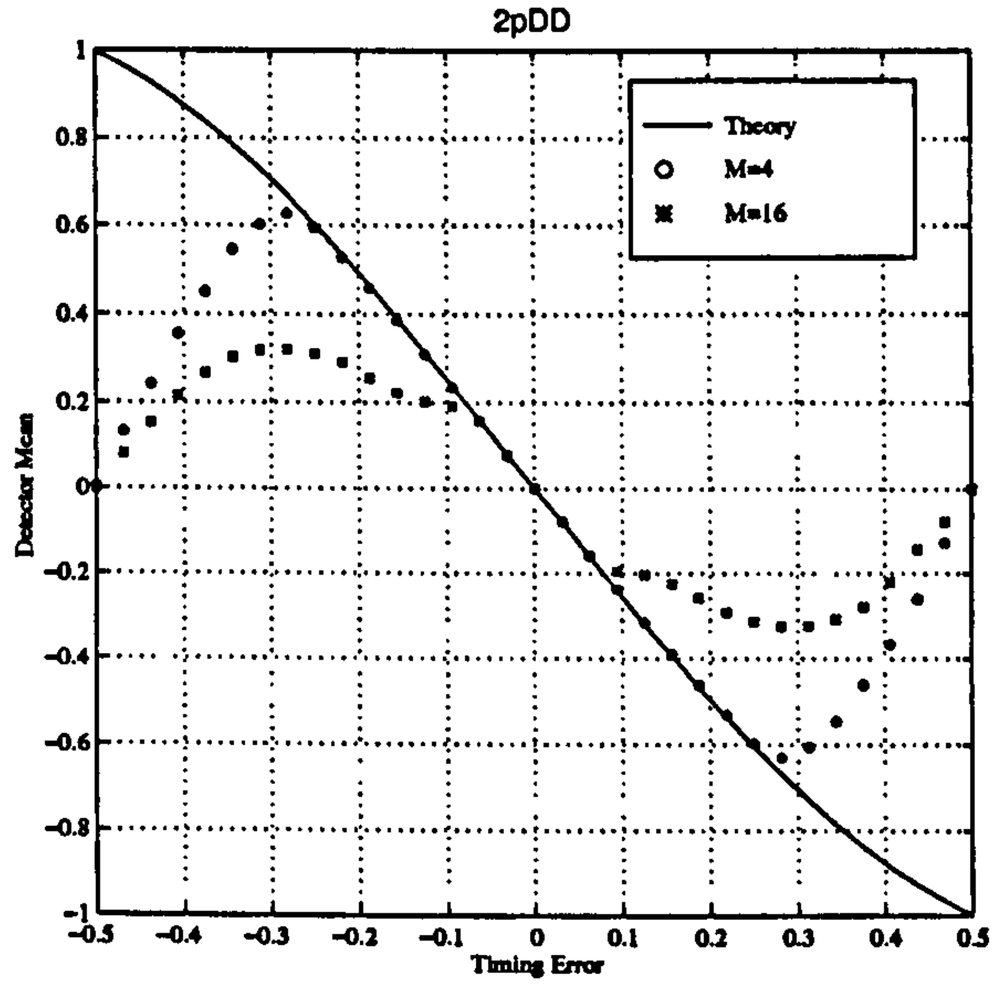

Figure 6.15: S-Surfaces showing the detector sensitivity as a function of delay and phase-error.



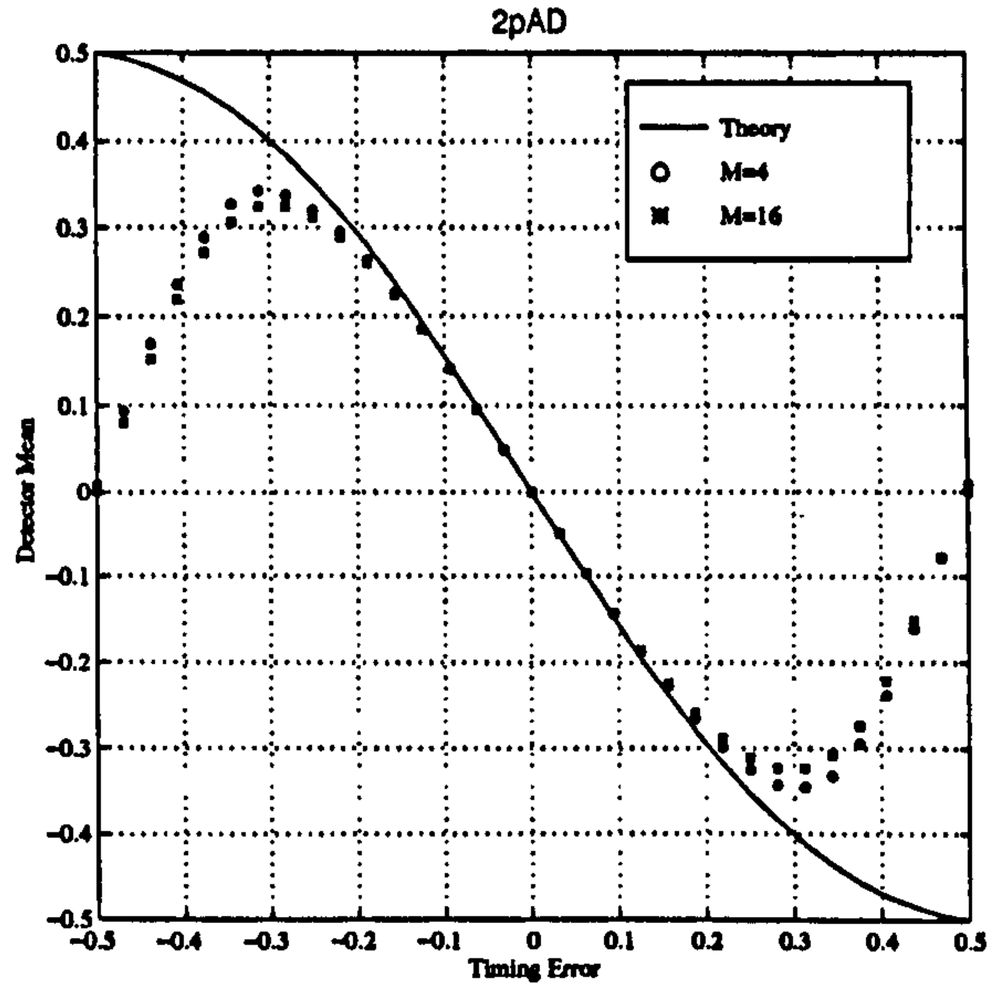

(a) Decision Directed




(b) Amplitude Directed




(c) Decision Directed




(d) Amplitude Directed

Figure 6.16: Effect of Modulation Order on Timing-Error Detector Characteristic

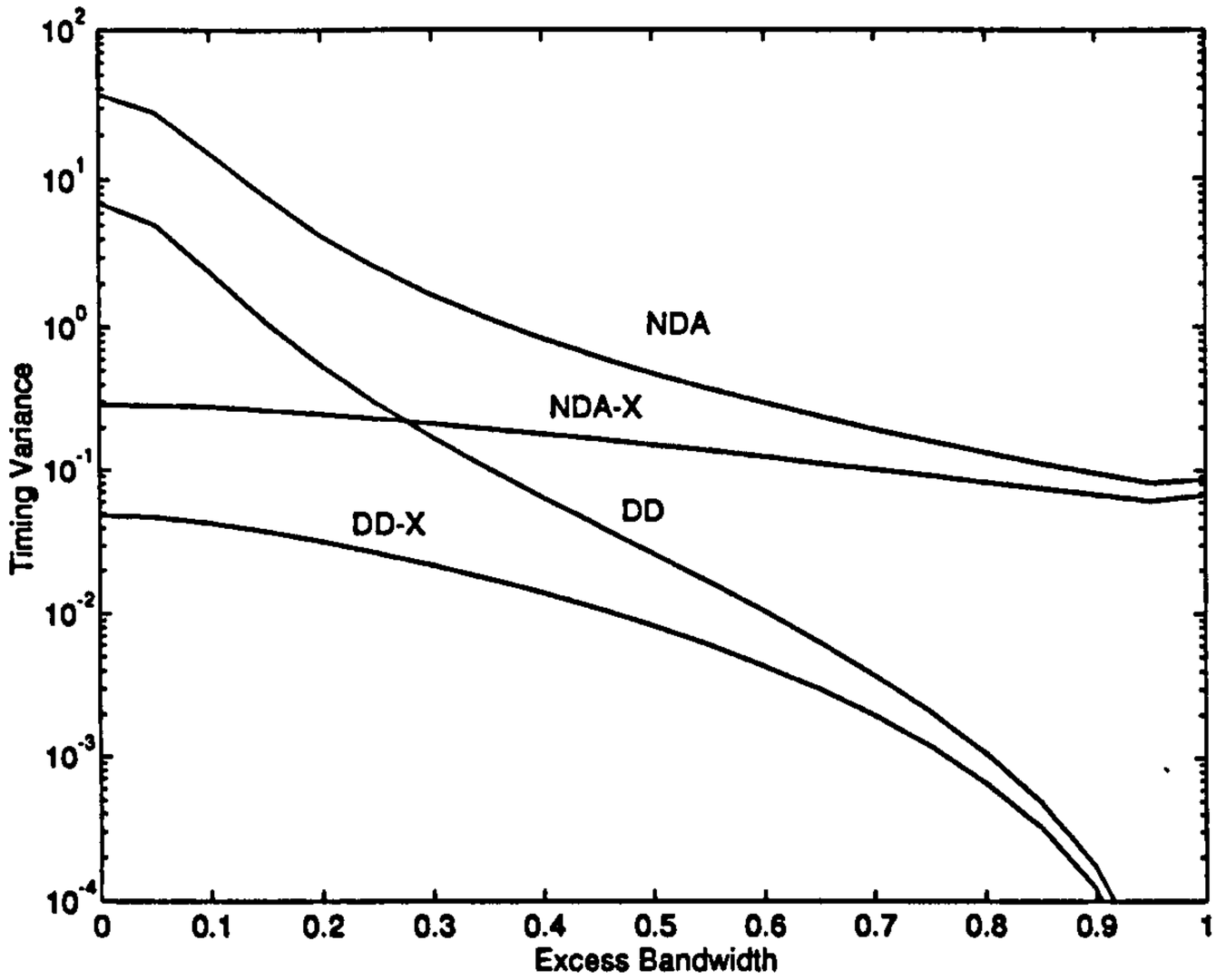



Figure 6.17: Open-Loop Variance of Decision-Directed and Non data Aided Timing Recovery Schemes.

ture. The problem with using an open-loop analysis to determine closed loop performance is that it inherently assumes a small-signal linearised loop model and so this approach may be somewhat limited.

In the case of most synchronisers the loop noise is made up of terms due to additive noise and terms due to self-noise. The spectrum of the noise is generally not flat. This is illustrated in figure 6.18 which shows the spectrum of the detector error signal at high signal to noise for the 2pNDA timing error detector operating on a raised-cosine data stream with excess bandwidth $\alpha = 0.2$. The coloured nature of the loop-noise is common in synchronisers as illustrated by figure 6.18 and this implies that the closed-loop noise variance is not directly proportional to the loop noise bandwidth, $B_L$, but rather a polynomial in $B_L$. This is completely different from a PLL tracking a sinusoid in AWGN where the variance is represented as

$$\sigma^2 = \frac{2\,B_L T}{K^2}\,S(0) \tag{6.76}$$

When the noise is not white we have to evaluate,

$$\sigma^2 = \frac{1}{K^2}\int_{-\infty}^{+\infty} S(f)\,|H(j\,2\pi f)|^2\,df \tag{6.77}$$

where $S(f)$ is the power spectral density of the generally coloured noise process. As we are dealing with discrete processes the loop noise spectrum is given as the discrete Fourier Transform of the autocorrelation sequence of the loop noise samples $R(m)$ as

$$S(f) = \sum R(m)\,\exp(-j\,2\pi m f T) \tag{6.78}$$

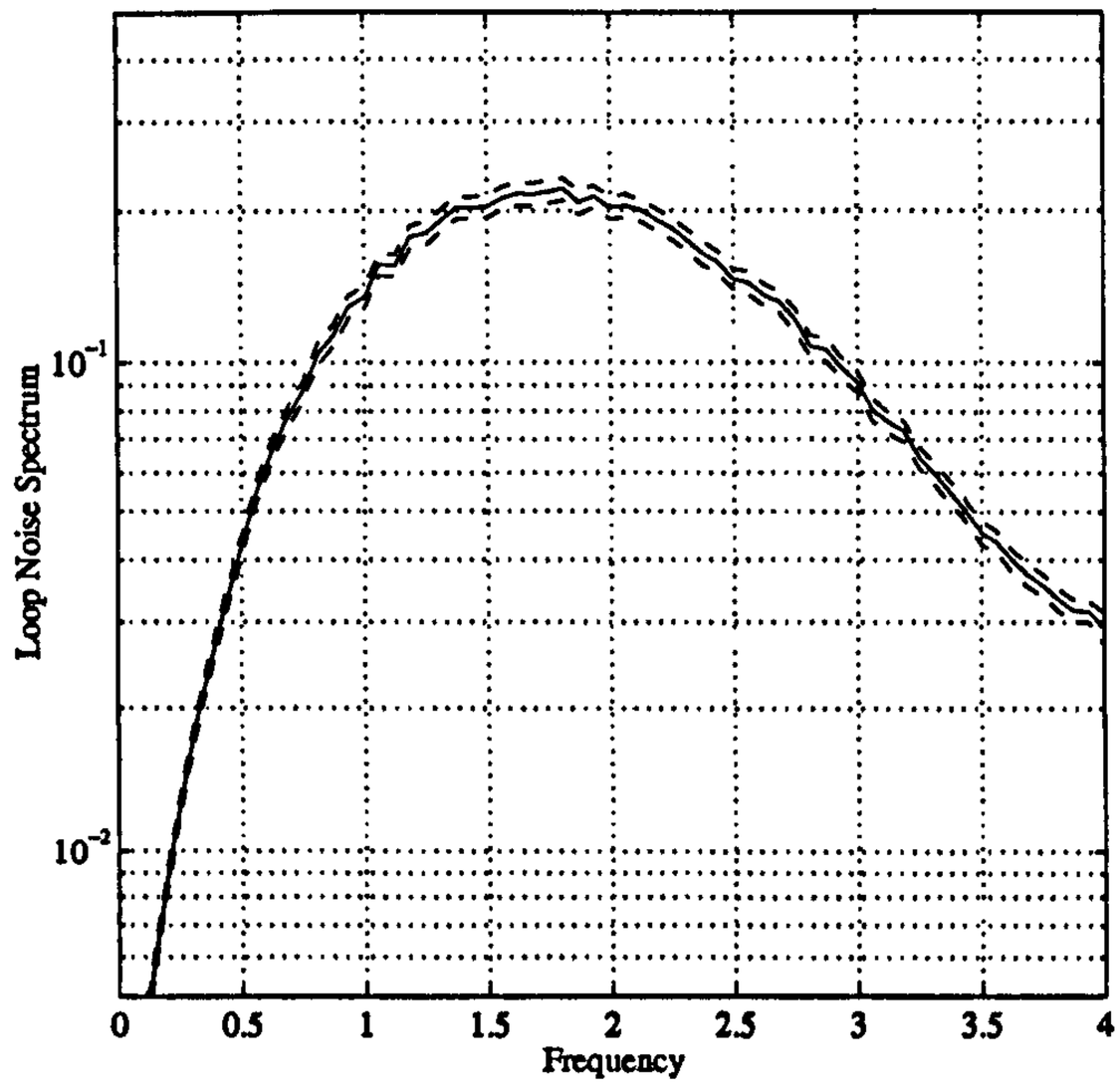



Figure 6.18: Spectrum of error signal for 2pNDA timing error detector. Excess bandwidth is $\alpha = 0.2$. No additive noise.

The method in [29] approximates the above integral by a Taylor series expansion in $S(f)$ to give,

$$\sigma^2 = \frac{2\,B_L T}{K^2}\,S(0) + \kappa(2\,B_L T)^2 \sum \frac{|m| - R(m)}{K^2} \tag{6.79}$$

where the factor $\kappa$ is determined by the choice of estimator structure, be it a feedforward or feedback loop.

The point is that the variance has the general form

$$\sigma^2 = (2\,B_L T)\frac{A}{2\,E_s/N_o} + \kappa(2\,B_L T)^2 B \tag{6.80}$$

where $A, B$ are coefficients determining the performance at moderate signal to noise and high signal to noise respectively. It is rather optimistically claimed in [29] that this model is suitable to describe all synchronisers be they feedback or feedforward. In [29] tables of $A$ and $B$ coefficients are provided for both symbol timing and carrier phase recovery. The coefficient $\kappa$ is determined by the loop structure. For a feedback loop with first order filter we get,

$$B_L = \frac{K}{4} \text{ and } \kappa = 2 \tag{6.81}$$

For a second order loop we get,

$$B_L T = \frac{\gamma\omega_n T}{2}\left(1 + \frac{1}{4\gamma^2}\right) \text{ and } \kappa = \left(\frac{4\gamma^2}{1+4\gamma^2}\right)^2 \tag{6.82}$$

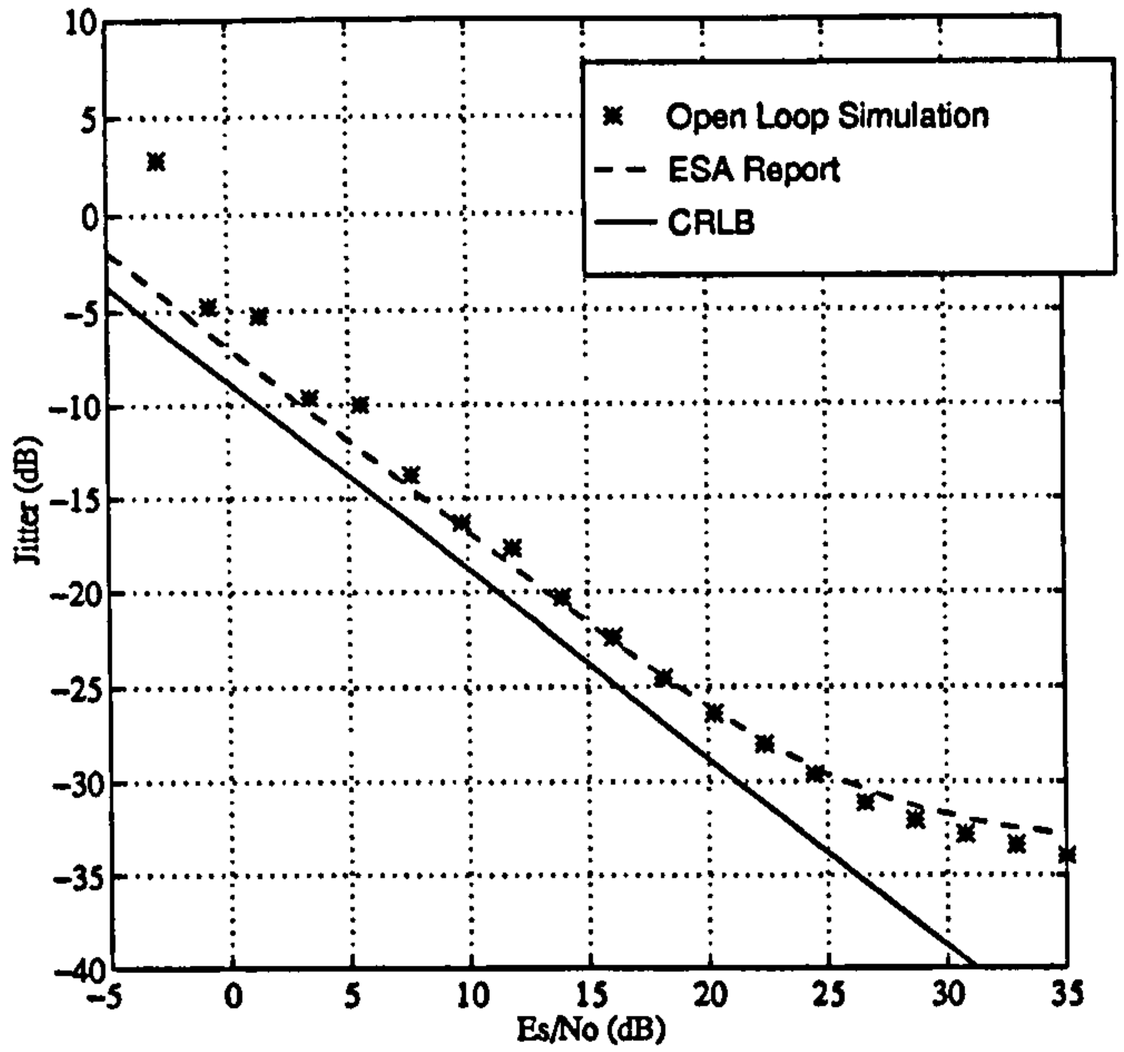



Figure 6.19: Open-loop detector self-noise for the 2pNDA TED.

The theory can also be applied to feedforward estimators operating over $L$ samples to give,

$$B_L = \frac{1}{2L} \text{ and } \kappa = 1 \tag{6.83}$$

In the above expansion the coefficient in $B_L T$ represents the synchroniser performance at moderate values of signal to noise, whereas the term in $(B_L T)^2$ represents the self-noise component. If this analysis proves to be useful then this theory could also be applied to frequency error detectors.

The advantage of this method is that by characterising the open-loop performance of the detector, be it frequency, phase or timing, one can then apply the results to feedforward or feedback loops of a given bandwidth.

Results comparing the open-loop 2pNDA timing error detector performance with the theory and the CRLB is shown in figure 6.19 for an excess bandwidth of $\alpha = 0.5$. The MATLAB script used to determine the open-loop performance of the timing-error detectors is in appendix B. More work needs to be done to understand this approach to determining closed-loop synchroniser performance.

## 6.6 Timing Error Correction

We have so far looked at the different timing-error detector schemes that can be employed, corresponding to our model of a generic asynchronous timing-recovery loop in figure 6.4. In the generation of the detector S-curves we have used a delay block at a higher sampling frequency. This will not be possible in reality and we will have to use an interpolation filter in order to achieve fractional-sample delays. Note that if the error-detection scheme is based on

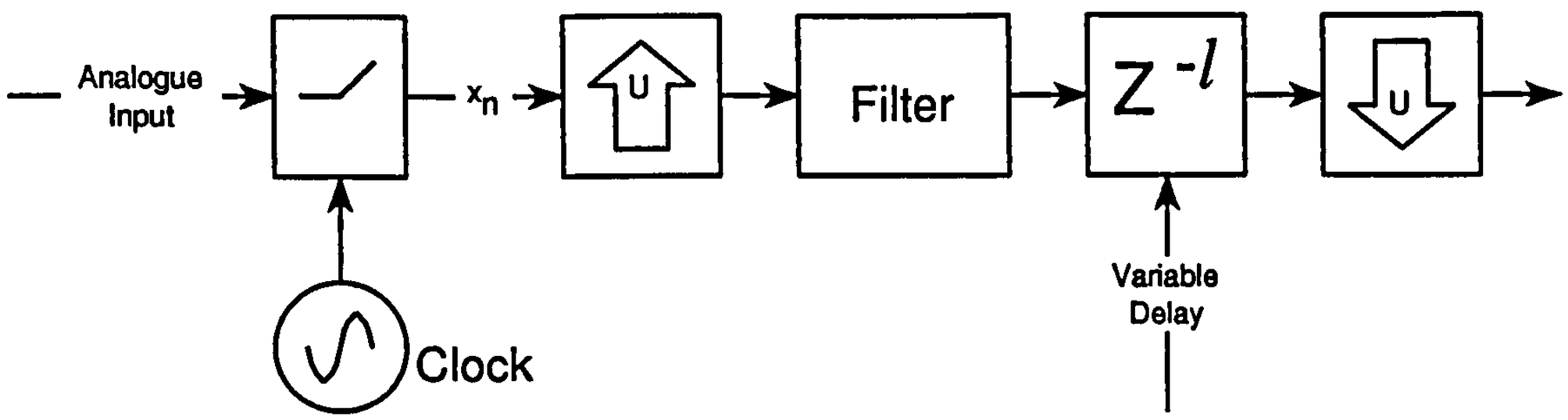



Figure 6.20: Conceptual model of how to acheive a fraction of a sample delay using multirate techniques.

$B$ samples per symbol then we will have to institute a bulk delay of up to $B$ samples as well as a fraction sample delay. It would be advantageous if the functions of bulk and fractional delay could be combined into the one functional block.

In this research we studied two methods of providing fractional delays in asynchronous timing recovery loops. The first method is to use a multirate polyphase, or switched-coefficient filter, and the second is to use a continuously-variable polynomial interpolator. For reasons which will become clear later there are good reasons to use the polynomial rather than the polyphase interpolator. We will now consider each method in turn.

### 6.6.1 The Polyphase Filter as a Fractional Delay Element

It is well known that one can achieve fractional delays by using multirate techniques [68]. Referring to figure 6.20, the $x_n$ samples out of the ADC are up-sampled by a factor $U$ and then filtered by an image-rejection or interpolation filter to produce $B \times U$ sample points per symbol. This sequence can then be delayed to choose those points closer to the optimum sampling points and then down-sampled by a factor $U$. If $M$ is the number of taps in the interpolation filter and we arrange it such that $M = U \times K$ then the process of up-sampling and interpolation by a factor $U$ can be represented as follows.

$$\begin{aligned}
\begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-U} \end{bmatrix} &= \begin{bmatrix} h_0 & h_U & \cdots & h_{(K-1)U} \\ h_1 & h_{U+1} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ h_{U-1} & h_{2U-1} & \cdots & h_{KU-1} \end{bmatrix} \times \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-U} \end{bmatrix} \\
&= \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{U-1} \end{bmatrix} \times \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-U} \end{bmatrix}
\end{aligned} \tag{6.84}$$

In (6.84) the $y_n$ represent the up-sampled and interpolated sequence, the $h_n$ are the $M$-tap

coefficients of the interpolation filter and the $P_n$ are the $K$-tap polyphase filter sections. The function of the polyphase filter as a fractional delay element is to select one of the $U$ values of the $y_n$ for every $x_n$ input, that is, there is no sample rate change overall. This is equivalent to filtering the input $x_n$ with a changing polyphase filter section $P_l$ dependent on the fractional delay, $l/U$, that is required. Such a switched-coefficient filter would then be time-variant.

Figure 6.21 shows the polyphase filter sections for an up-sample factor of $U = 5$ and $K = 19$. Note that for these particular values the filter sections are symmetrical – thus it would only be necessary to store $(U-1)/2$ sections and use reverse indexing if coefficient memory were scarce. Note also, that the middle section corresponding to $\delta = 0$ simply reproduces the signal that is present at the input (plus some bulk delay of $(U-1)/2$). The delays available with this filter cover the range $\delta = -2/5$ to $\delta = +2/5$ relative to the middle section. The effect of convolving an input signal which has four samples per symbol with each of these polyphase filter sections is as shown in figure 6.23. Similar results but for an input data stream with only two samples per symbol is shown in figure 6.22. This shows the eye diagram of the input data after it has been convolved with each polyphase filter section. It is evident that the effect of instituting a fractional delay is to change the phase of the samples relative to the *underlying* analogue wave — it is important to note that the sampling instants remain the same. One can see by inspecting where the solid curve intersects the left-hand axis of each graph that the phase of the underlying analogue wave is moving from right to left as $\delta$ goes from $\delta = -2/5$ to $\delta = +2/5$. It is also evident from the figures that interpolation is more important when operating at fewer samples per symbol.

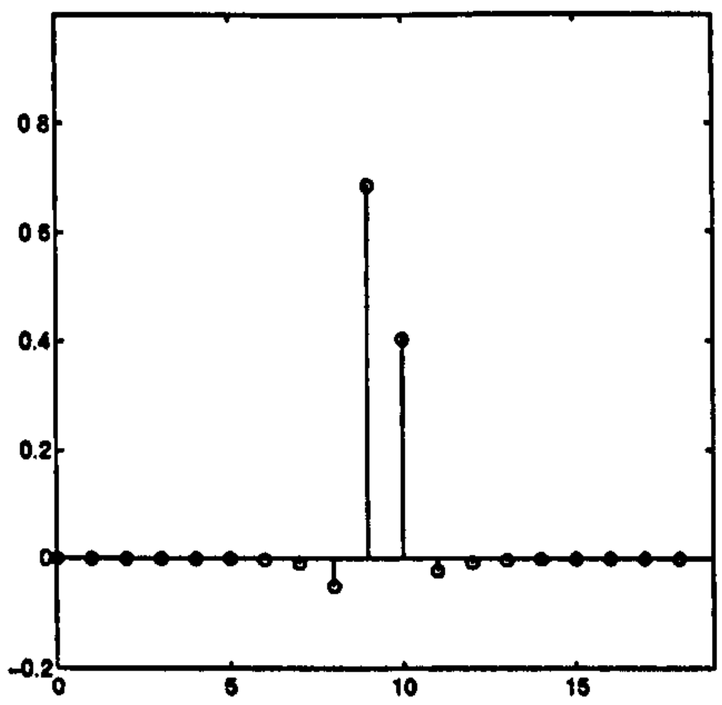
Note that because of the phase of the samples in the input data stream the polyphase section that provides the optimum sampling points can easily be seen to be that corresponding to $\delta = 0$. One can appreciate that the higher the value of upsample the smaller the *granularity* of the delays that one can choose. Obviously, the penalty that has to be paid for this higher accuracy is that of higher memory usage. There are more involved algorithms which reduce the memory requirements but keep the same granularity by calculating polyphase sections as and when they are needed [62] but it felt that these place a heavier computing burden on the synchroniser than the method discussed here and might preclude real-time operation.

### Note on Raised-Cosine filtered PAM Data Streams

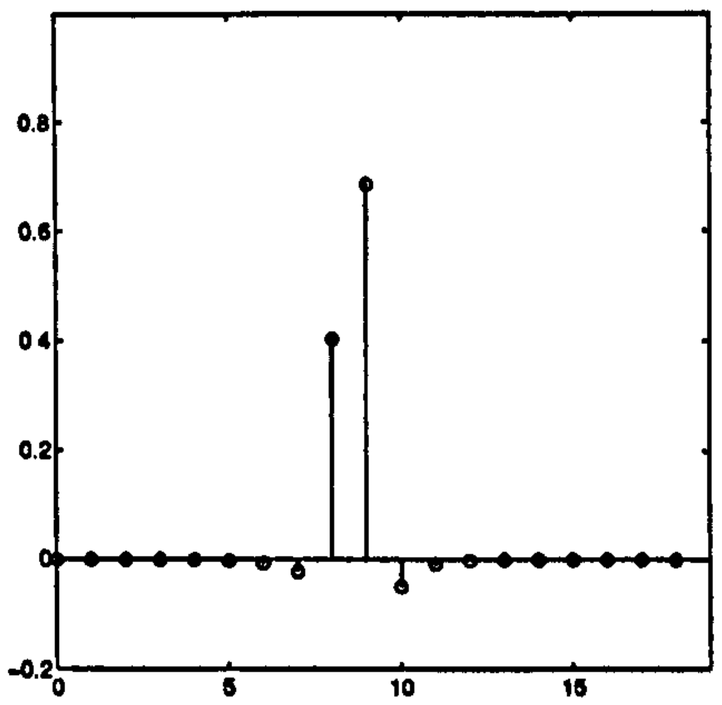
The PAM data streams that are used in this study are Raised-Cosine filtered. The Raised-Cosine filters that are available as standard in SPW are based on the Fast Fourier Transform (FFT) and hence will have an even number of taps. Thus, if one uses such filter to filter data with an even sampling frequency then the optimum sampling points will not be present in the signal. For this reason, a filter design method for the Filter Design System was implemented which would allow one to design Raised-Cosine filters with an arbitrary number of taps.

### Choice of Interpolation Filter

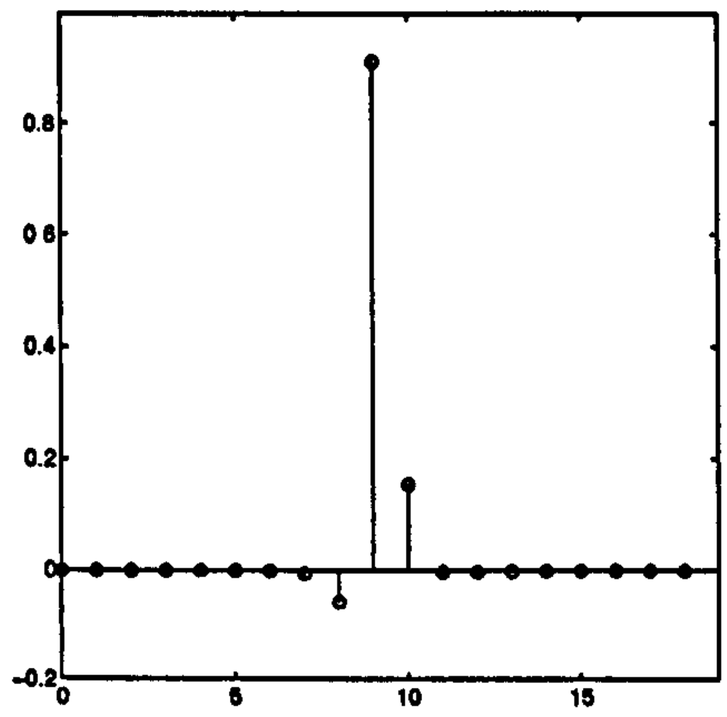
The primary function of the interpolation filter in a multirate system is, obviously, to interpolate between up-sampled points. An alternative and equivalent view from the frequency



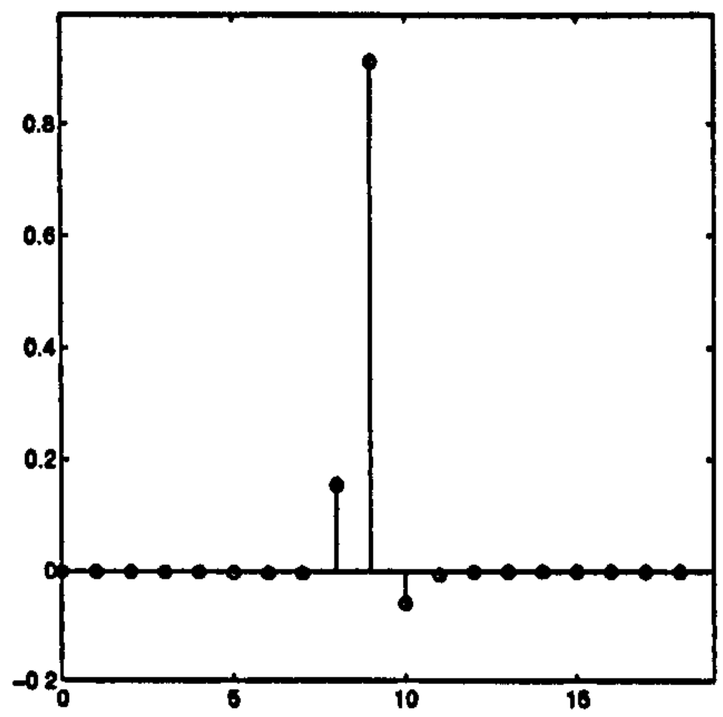

(a) $\delta = -2/5$



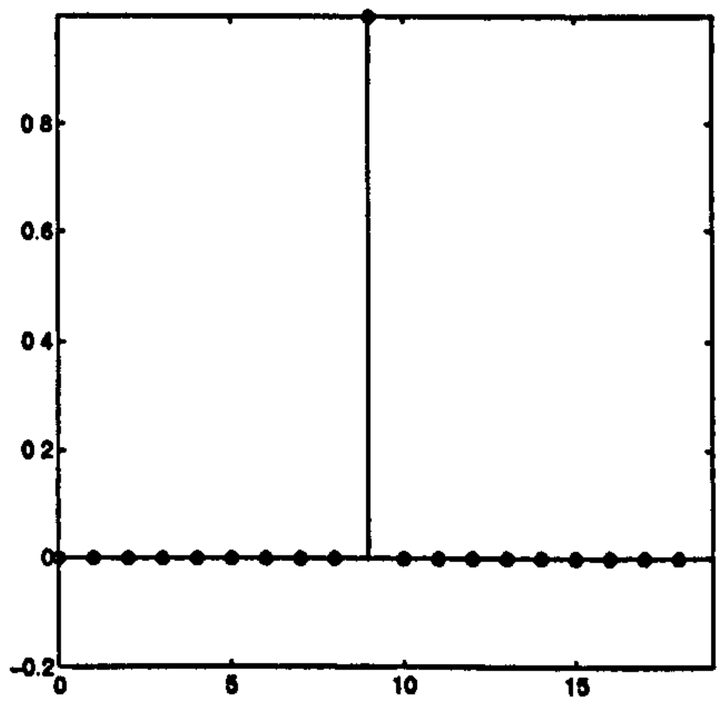

(b) $\delta = +2/5$




(c) $\delta = -1/5$

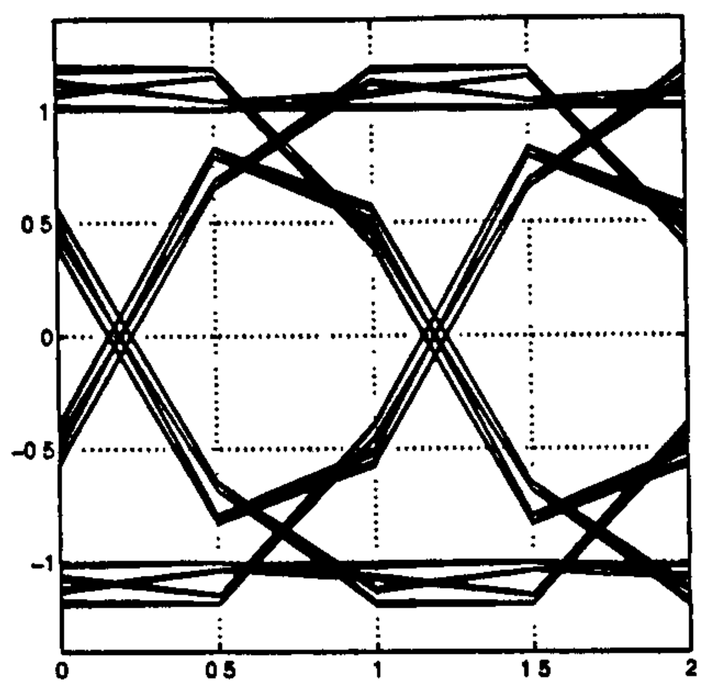



(d) $\delta = +1/5$

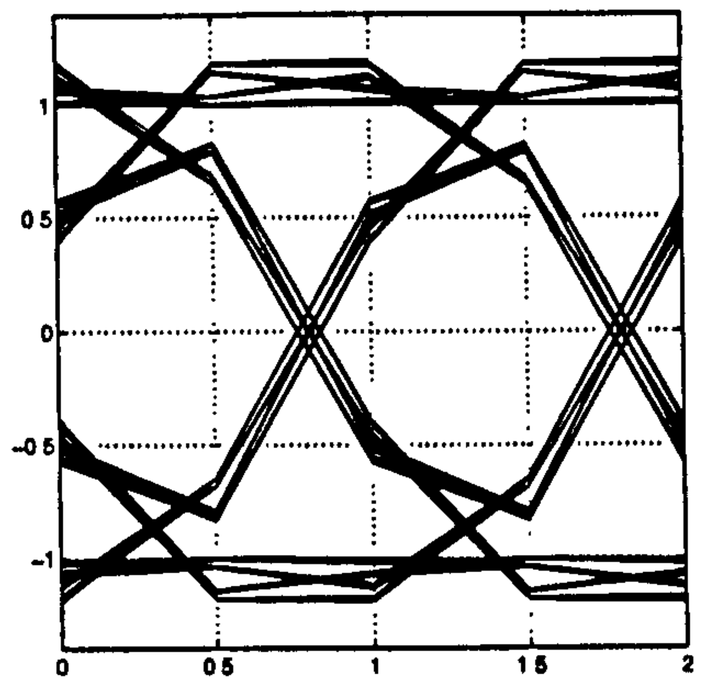



(e) $\delta = 0$

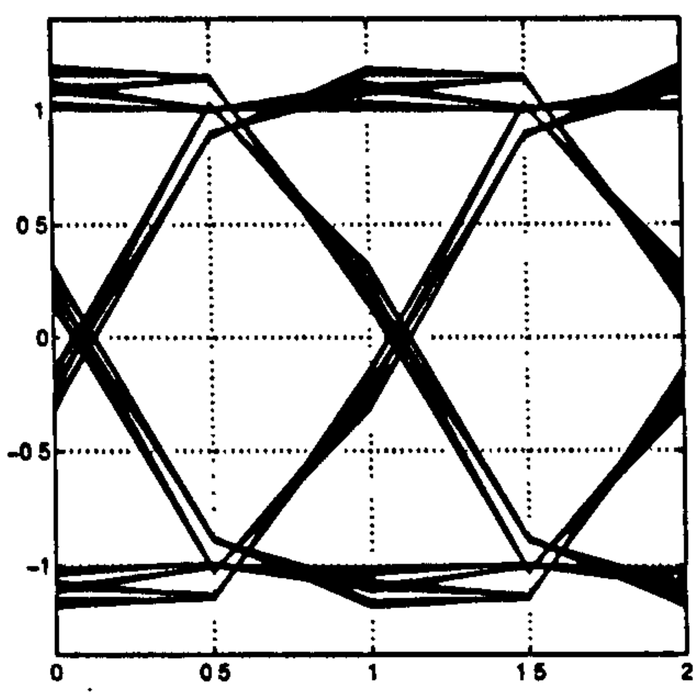
Figure 6.21: Individual filter sections for Polyphase filter with $U = 5$ & $K = 19$



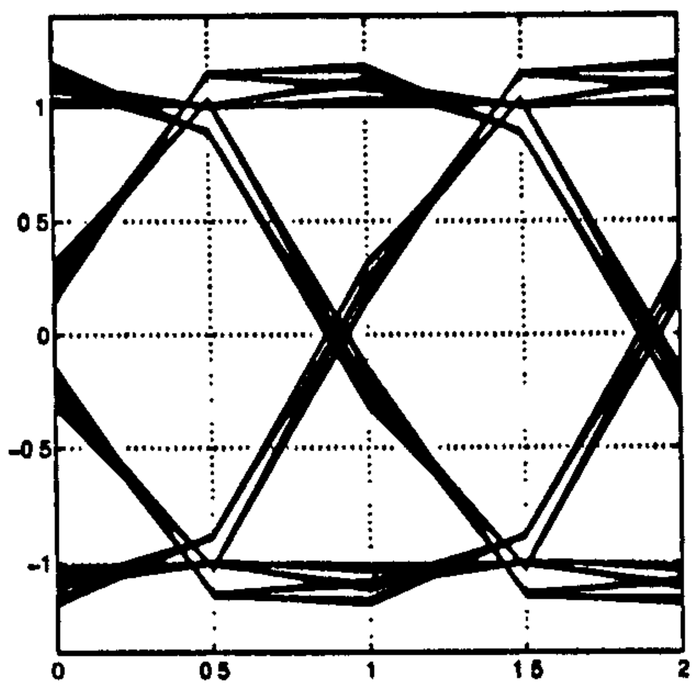

(a) $\delta = -2/5$



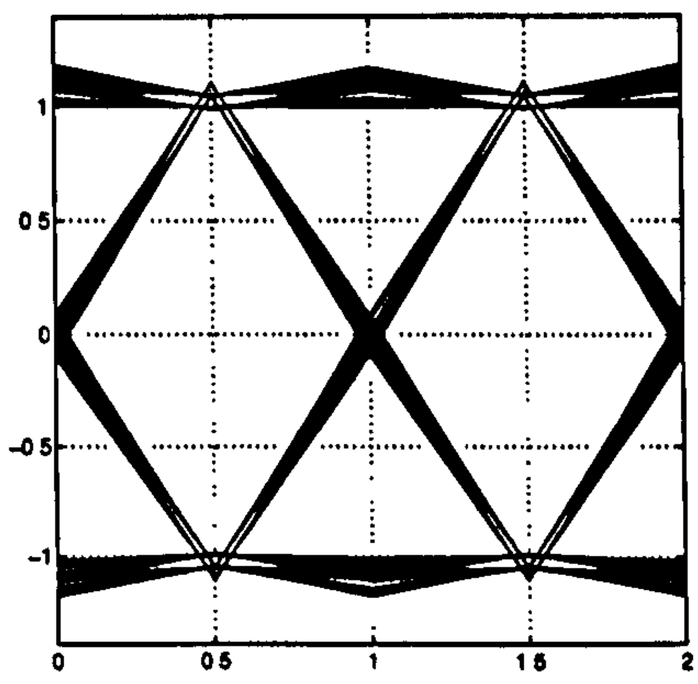

(b) $\delta = +2/5$




(c) $\delta = -1/5$

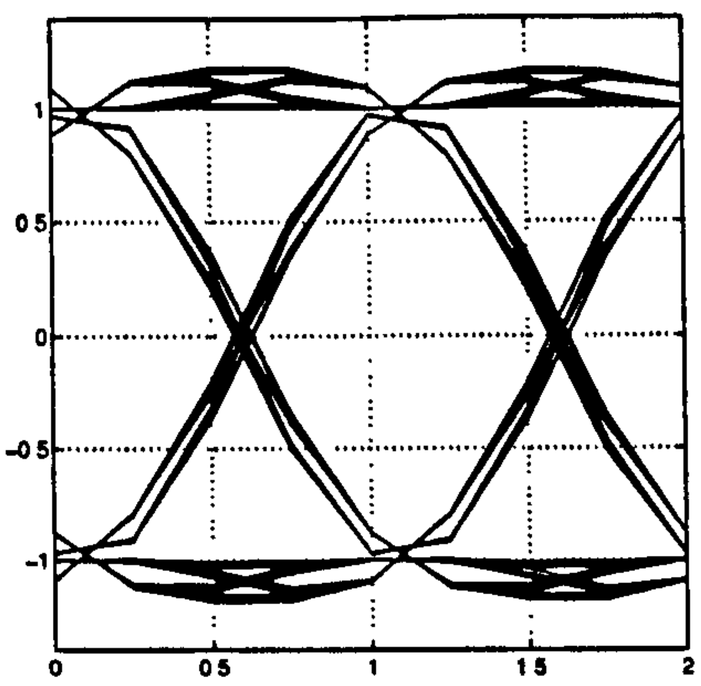



(d) $\delta = +1/5$

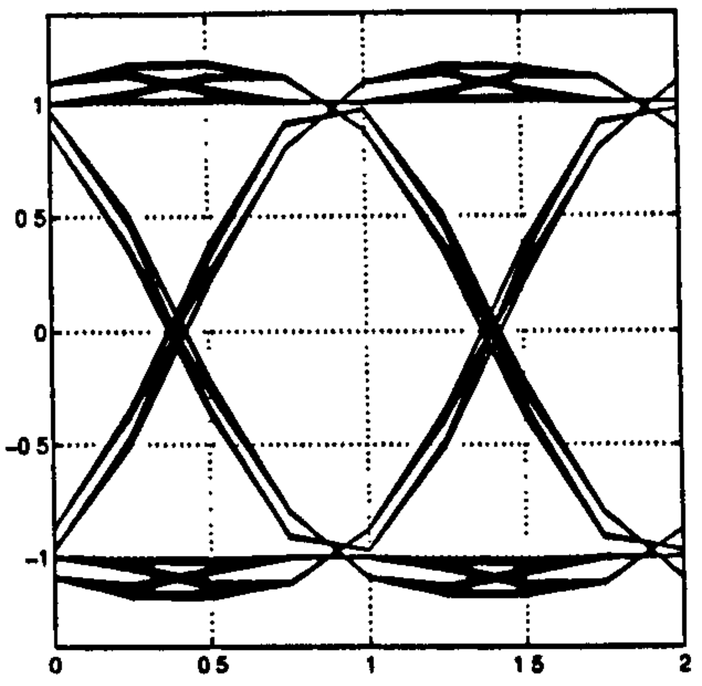



(e) $\delta = 0$

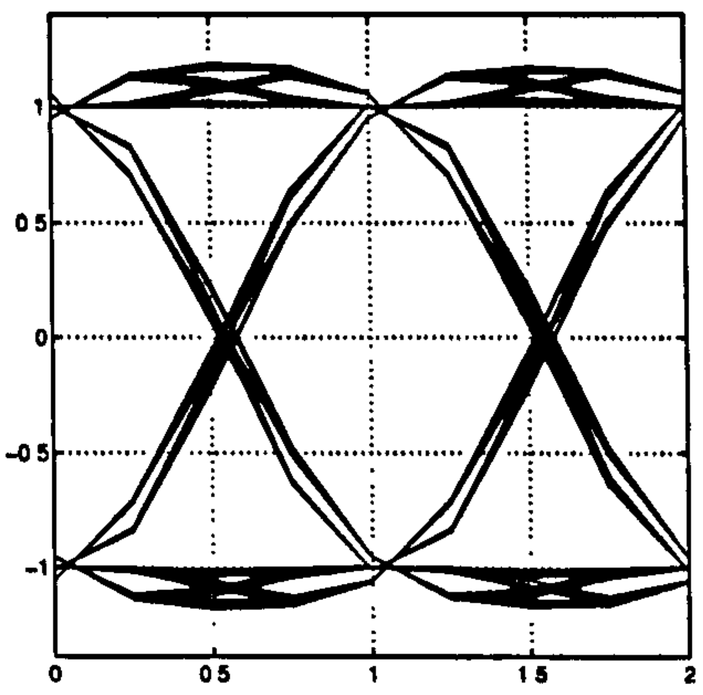
Figure 6.22: Illustration of fractional delay for an input raised-cosine filtered data stream with $\alpha = 0.8$ and two samples/symbol.



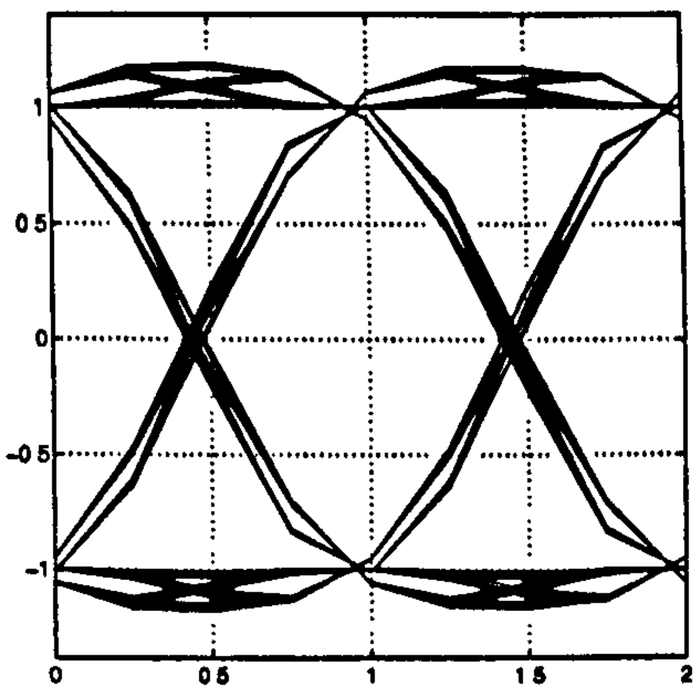

(a) $\delta = -2/5$



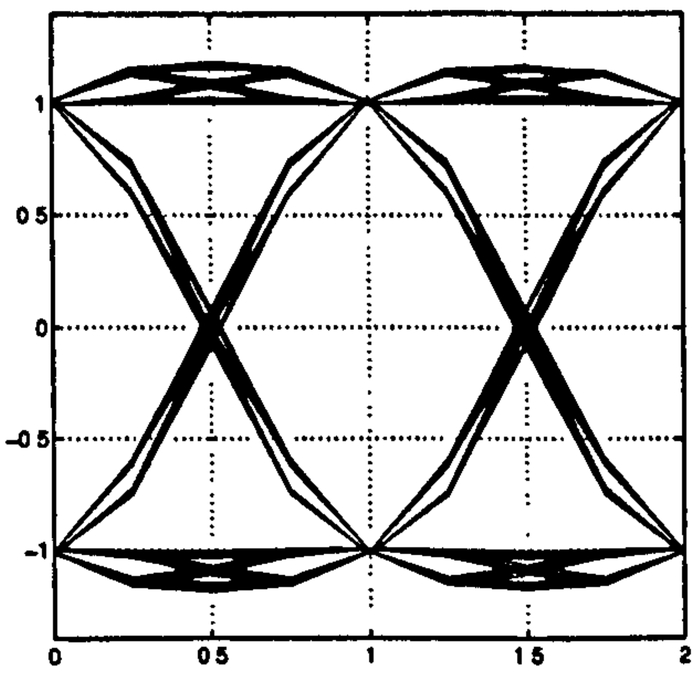

(b) $\delta = +2/5$




(c) $\delta = -1/5$




(d) $\delta = +1/5$




(e) $\delta = 0$

Figure 6.23: Illustration of fractional delay for an input raised-cosine filtered data stream with $\alpha = 0.8$ and four samples/symbol.

domain is that of image-rejection. There are other filtering functions that will need to be undertaken in the receiver, such as data filtering, and it is possible to merge these transfer functions into that of the interpolation filter. However, it is noted in [62] that this is sub-optimal, that is, the best filter for data filtering will not be generally the best filter for interpolation. Also, it is felt by the author that this is more of an implementation issue that can be addressed when one has a particular receiver architecture that one wishes to consider.

For these reasons the choice of the interpolation filter was based solely on an optimum filter for interpolation. It is found [46] that a good approximation to the ideal interpolation filter is that of the Raised Cosine filter. Referring back to (6.84) we can now express the filter coefficients mathematically.

$$h_n = h_{rc}\left(n - \frac{KU-1}{2}\right) \tag{6.85}$$

where

$$h_{rc}(t) = \frac{\sin\left(\frac{\pi t}{U}\right)}{\left(\frac{\pi t}{U}\right)} \times \frac{\cos\left(\frac{\pi\beta t}{U}\right)}{\left(1 - \frac{4\beta^2 t^2}{U^2}\right)} \tag{6.86}$$

and $\beta$ is the filter roll-off factor.

### Complete Algorithm Including Bulk Delay

It has been illustrated how a fractional delay of $l/U$ samples can be achieved by using a polyphase filter with up-sample factor $U$. A simple procedure for implementing bulk delays taking into account the $B$-modulo nature of the delay is now described.

Bulk delay can be included by simply extending the shift register containing the input $x_n$ by $B-1$ and offsetting the filtering operation by the bulk delay required, $d$, with $0 \leq d < B$.

The complete algorithm is as follows

- Required delay, $\tau_e$, is input.
- Shift data, $x_n$, into $K+B-1$ length register.
- Reduce input delay modulo $B$ by putting

$$m = \begin{cases} \lfloor \tau_e U \rfloor / U \bmod B + B & \text{if } \tau_e < 0 \\ \lfloor \tau_e U \rfloor / U \bmod B & \text{if } \tau_e \geq 0 \end{cases} \tag{6.87}$$

- Fix bulk delay $d$ as

$$d = B - 1 - \lfloor m \rfloor \tag{6.88}$$

- Fix fractional delay $u$ as

$$u = (m - \lfloor m \rfloor) \times U \tag{6.89}$$

- Filter data and output interpolated sample

$$y_{n-u} = \sum_{i=0}^{K} P_{u_i}\, x_{i+d} \tag{6.90}$$

It is noted that the nature of the algorithm may make it suitable for DSP implementation. Obviously, the filtering operation in 6.90 is suited to DSP implementation, but the extra overheads for calculating and implementing the offsets $u$ and $d$ in coefficient and data memory would not be significant.

## 6.7 Loop Control

When there is a timing-drift between transmitter and receiver then the receiver must accommodate for this. This will nearly always be the case where the transmitter and receiver operate asynchronously. There will not be just a difference in the timing-phase but this difference will change with time. That is to say, there will be a frequency difference between the received data rate and the local clock. Generally, this frequency difference is very small (of the order of parts per million of a symbol period) and so the use of AFC for symbol synchronisation is not generally required. If one corrects for differences in timing-phase by using a delay element then this delay element will be continuously increasing or decreasing depending on whether the frequency difference is positive or negative. There would not be a problem if the delay element had infinite memory – but this will never be the case. Sooner or later symbols will either have to be dropped or stuffed. As an example suppose the received baud rate was 0.1% greater than the locally generate clock. Then for every 1000 clock tics at the receiver one would have to clock out 1001 bits — that is an extra bit has to accommodated. There are two ways of achieving this; the first is to use the polyphase filter to detect overflow/underflow conditions and tell the receiver system that it then has to drop/stuff a symbol and the second is to use a continuously variable number-controlled oscillator (NCO). The modification of the polyphase filter to implement bit stuffing/dropping is shown in figure 6.24.

Loop control via NCO works as follows. The interpolator is used with the NCO to institute only a fractional delay between 0 and 1 of a sample. If a delay is greater than a sample is required then the phase of the NCO is adjusted accordingly. The bit stuffing/dropping strategy would insert an extra clock pulse midway between the usual clock pulses on detection of the underflow/overflow condition on the polyphase filter. The NCO, however, can use any one of the 4 samples to output a clock tic and one would therefore imagine that this is much more flexible than the previously proposed scheme. This is discussed further in the next section.

## 6.8 Polynomial (Farrow) Interpolation

Following on from the recent research in [15], [16] which was a cumulation of work done in [9], [10], [11], [12], [13] and [14] a symbol timing recovery loop has been implemented which

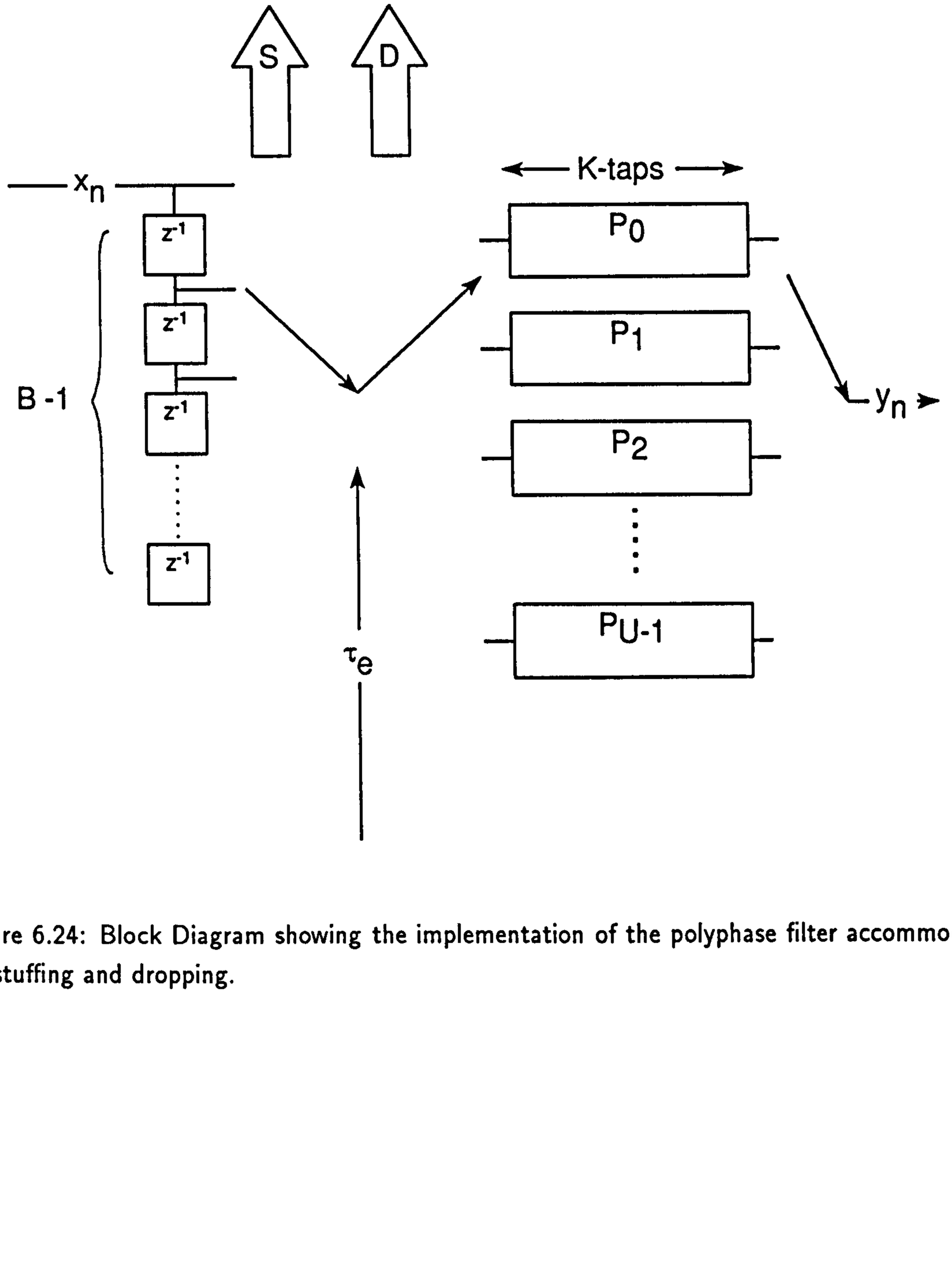



Figure 6.24: Block Diagram showing the implementation of the polyphase filter accommodating bit stuffing and dropping.

has a fractional delay provide via polynomial interpolation (rather than a polyphase filter) and the control via a NCO. The results of this work is covered briefly here.

Through any two points there is a unique line. Through any three points there is a unique quadratic — and so on. The interpolating polynomial of degree $N-1$ through the points $x_0 = f(t_0), x_1 = f(t_1), \ldots, x_{N-1} = f(t_{N-1})$ is given explicitly by Lagrange's classical formula,

$$\begin{aligned} P(t) &= \frac{(t-t_2)(t-t_3)\cdots(t-t_N)}{(t_1-t_2)(t_1-t_3)\cdots(t_1-t_N)}x_0 + \frac{(t-t_1)(t-t_3)\cdots(t-t_N)}{(t_2-t_1)(t_2-t_3)\cdots(t_2-t_N)}x_1 + \cdots \\ &\cdots + \frac{(t-t_1)(t-t_2)\cdots(t-t_{N-1})}{(t_N-t_1)(t_N-t_2)\cdots(t_N-t_{N-1})}x_{N-1} \end{aligned} \tag{6.91}$$

or, put more succinctly,

$$P(t) = \sum_{n=0}^{N-1} C_n(t)\, x_n \tag{6.92}$$

where

$$C_n(t) = \prod_{i=0, i\neq n}^{N-1} \frac{t-t_i}{t_n-t_i} \tag{6.93}$$

From these equations we can derive a structure that would act as an interpolator using $N$ data samples. In the following we illustrate the method using four data samples, that is a cubic polynomial interpolator.

To derive a practical signal processing architecture for a four-point interpolator we need to examine figure 6.25. In the figure the samples available to our simulation are the $x_n$ taken at uniform time samples $t_n$. The solid curve represents the interpolation polynomial $P(t)$ which is a weighted sum of the $x_n$ samples as follows.

$$P(t) = C_0(t)\, x_0 + C_1(t)\, x_1 + C_2(t)\, x_2 + C_3(t)\, x_3 \tag{6.94}$$

where the $C_n(t)$ are given as

$$C_0(t) = \frac{(t-t_1)(t-t_2)(t-t_3)}{(t_0-t_1)(t_0-t_2)(t_0-t_3)}; \quad C_1(t) = \frac{(t-t_0)(t-t_2)(t-t_3)}{(t_1-t_0)(t_1-t_2)(t_1-t_3)};$$
$$C_2(t) = \frac{(t-t_0)(t-t_1)(t-t_3)}{(t_2-t_0)(t_2-t_1)(t_2-t_3)}; \quad C_3(t) = \frac{(t-t_0)(t-t_1)(t-t_2)}{(t_3-t_0)(t_3-t_1)(t_3-t_2)}.$$

Simplifications to the above equations can be made by noting that $(t_1 - t_0) = T_s$, etc. Also we define the time variable $t = t_1 + \mu$ to give a unit sample delay over the centre portion of the curve as indicated in figure 6.25. After simplification (and normalisation by $T_s$) this gives the $C_n(\mu)$ as

$$\begin{aligned} C_0(\mu) &= -\frac{\mu^3}{6} + \frac{\mu^2}{3} - \frac{\mu}{3} \\ C_1(\mu) &= \frac{\mu^3}{2} - \mu^2 - \frac{\mu}{2} + 1 \\ C_2(\mu) &= -\frac{\mu^3}{2} + \frac{\mu^2}{2} + \mu \\ C_3(\mu) &= \frac{\mu^3}{6} - \frac{\mu}{6} \end{aligned} \tag{6.95}$$

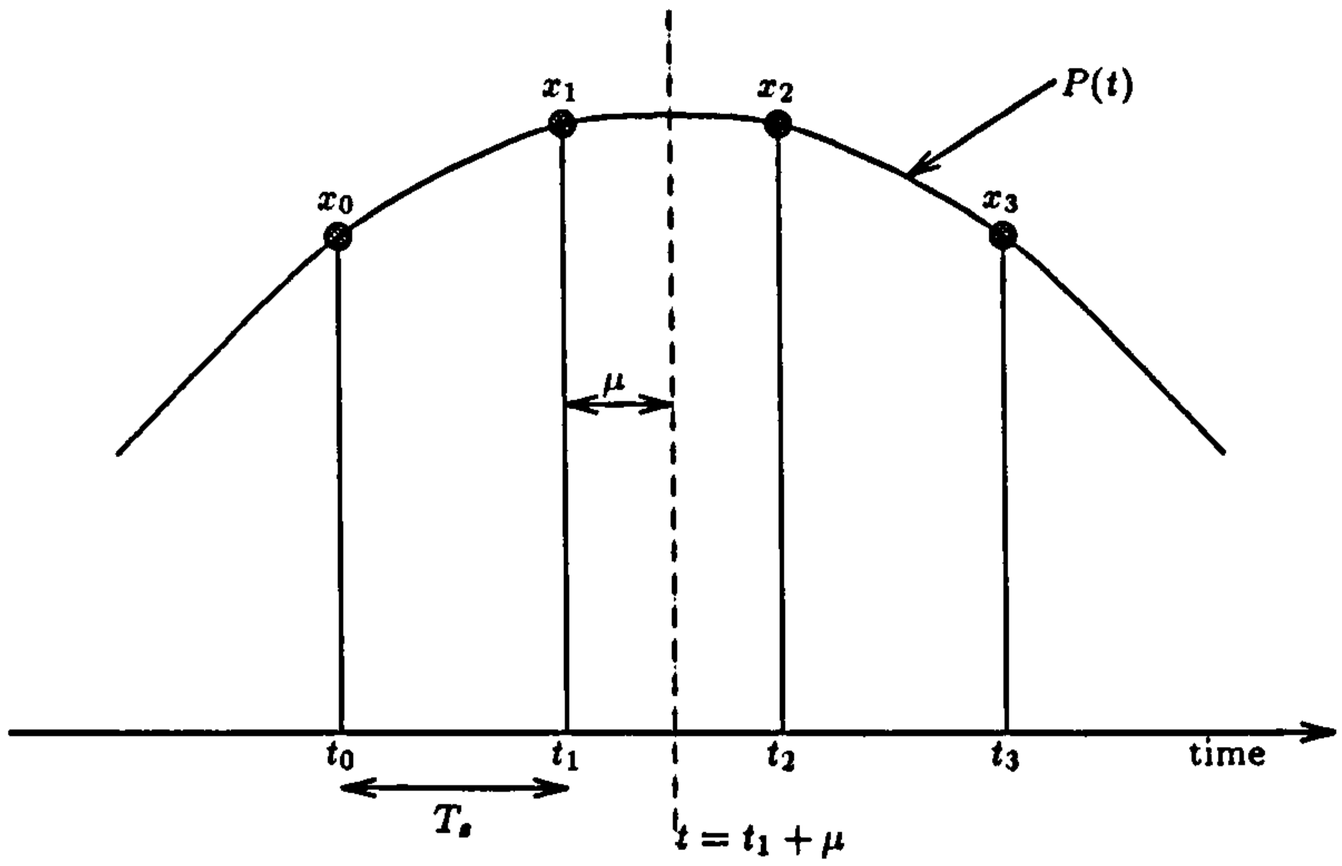



Figure 6.25: Quadratic Interpolation on Four Base Points

The SPW implementation of the above Farrow interpolator [10] which fits a four base-points to a cubic polynomial is as shown in figure 6.27. The filter taps in the figure are taken directly from the $C_n(\mu)$ above. The linear interpolator in figure 6.26 is also a special case of this theory. A simplification of the cubic interpolator is arrived at by fitting a quadratic polynomial to four base-points [16]. The simplified interpolator is as shown in figure 6.28. The performance of these interpolators is shown in figure 6.29 for 40% RC-BPSK modulation with four samples per symbol. The advantage of these interpolators over the previous polyphase interpolator is that they provide a continuously variable delay.

It has been noted as a result of this research that the effect of using a polyphase switched-coefficient filter as the interpolation filter is to result in a staircase approximation to the ideal detector s-curve, unlike the polynomial interpolator that produces a smooth detector characteristic as illustrated by figure 6.29. The author has noted in [69] that this *granularity* in the detector S-curve can actually exacerbate the effects of hangup in NDA timing recovery loops. This is illustrated in figure 6.30 for the 2pNDA & 2pDD timing-error detectors.

A comparison of the closed-loop tracking performance of the linear, cubic and reduced cubic interpolators are shown in figure 6.31 together with the modified Cramer-Rao bound. Closed-loop tracking performance will be discussed in greater detail later, but here we wish to just compare the performance of the different interpolators. Actually it appears that there is not a great deal of difference in terms of tracking performance between the linear, cubic and reduced-cubic interpolators, but the results shown are for a raised-cosine filtered data stream with $\alpha = 0.8$ and four samples per symbol. If we had fewer samples per symbol and a smaller value of excess bandwidth then the difference between the interpolators may be more marked.

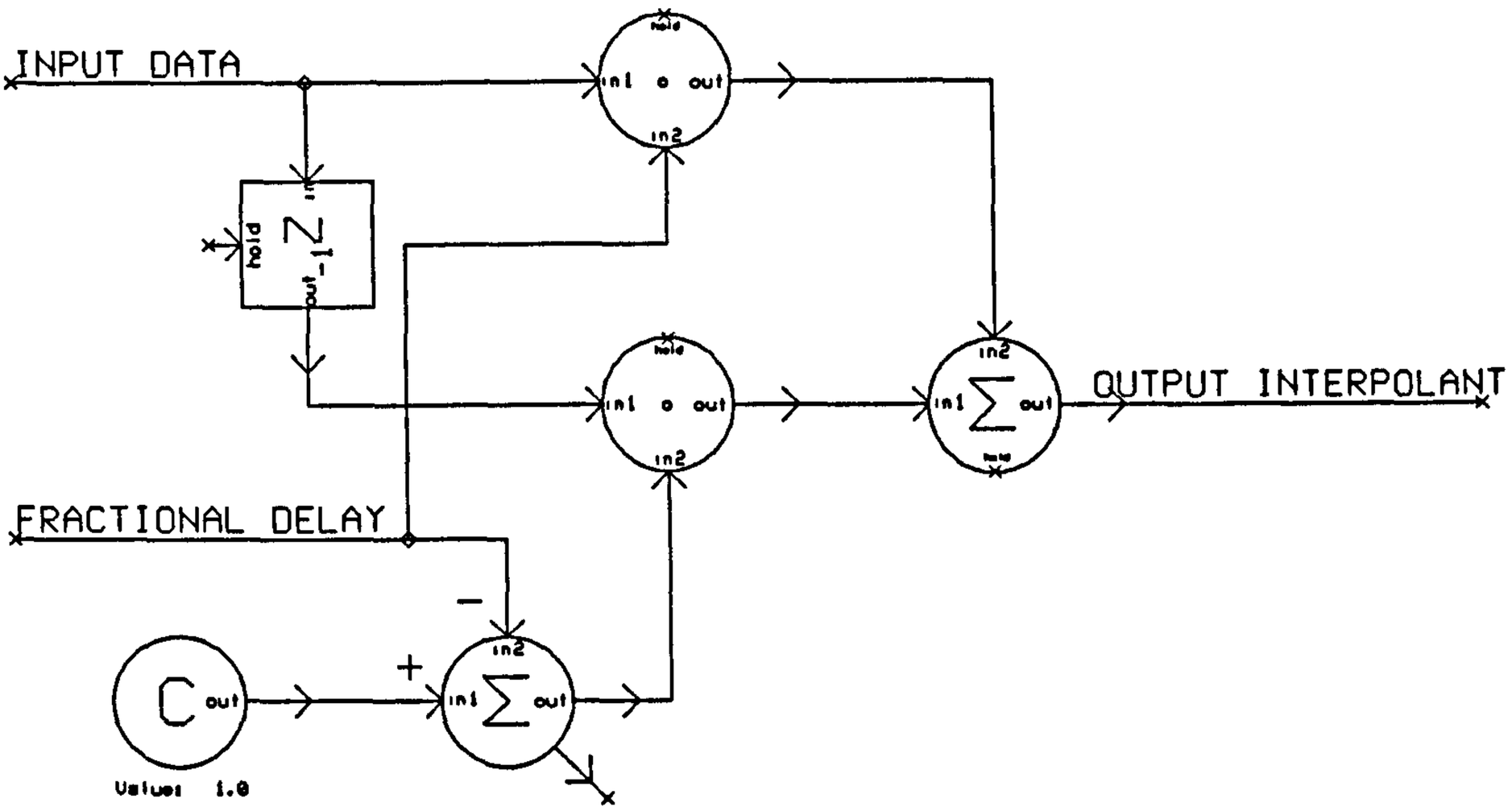



Figure 6.26: Block Diagram of the linear interpolator.

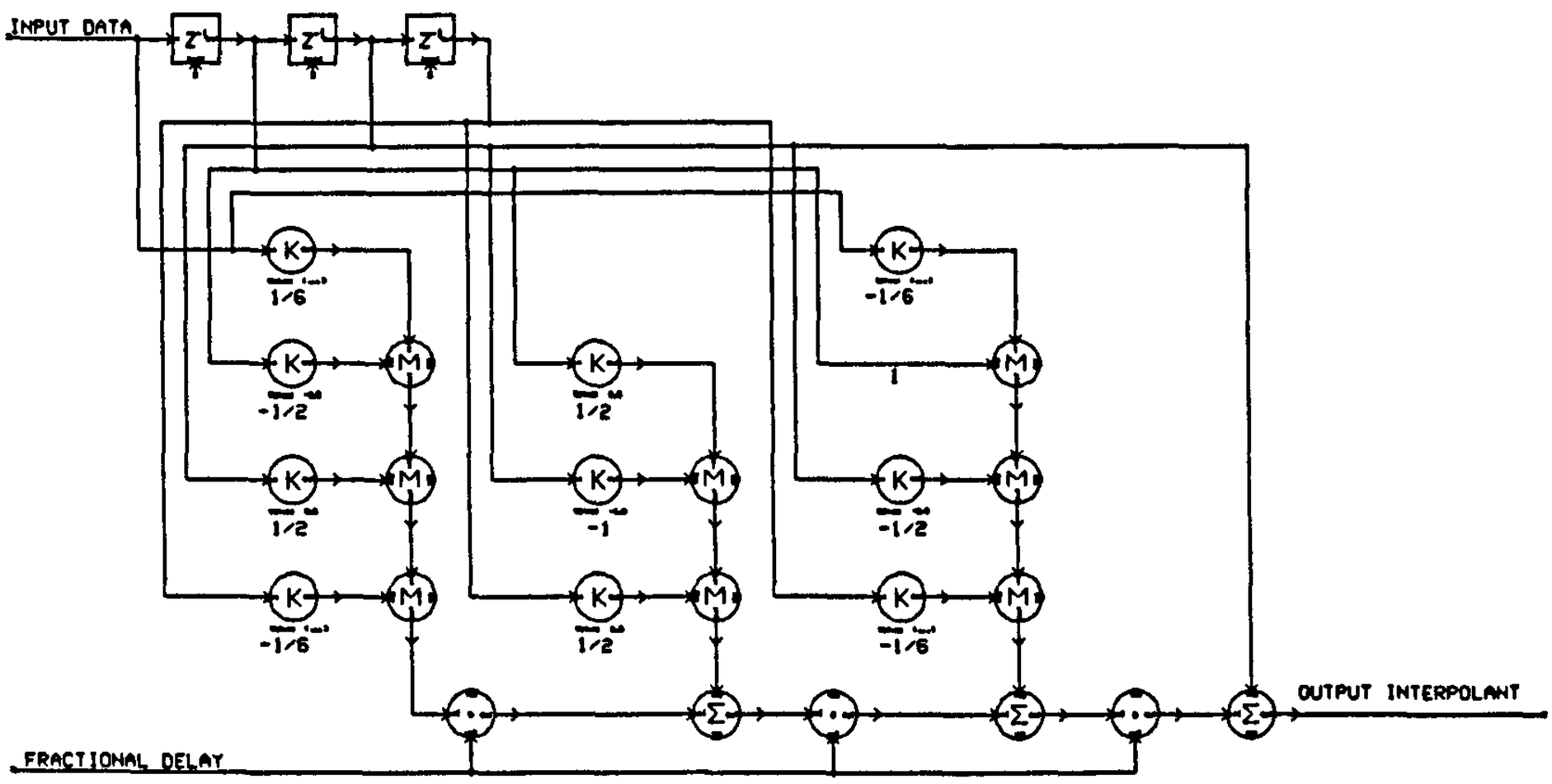



Figure 6.27: Block Diagram of the four-point cubic Farrow interpolator.

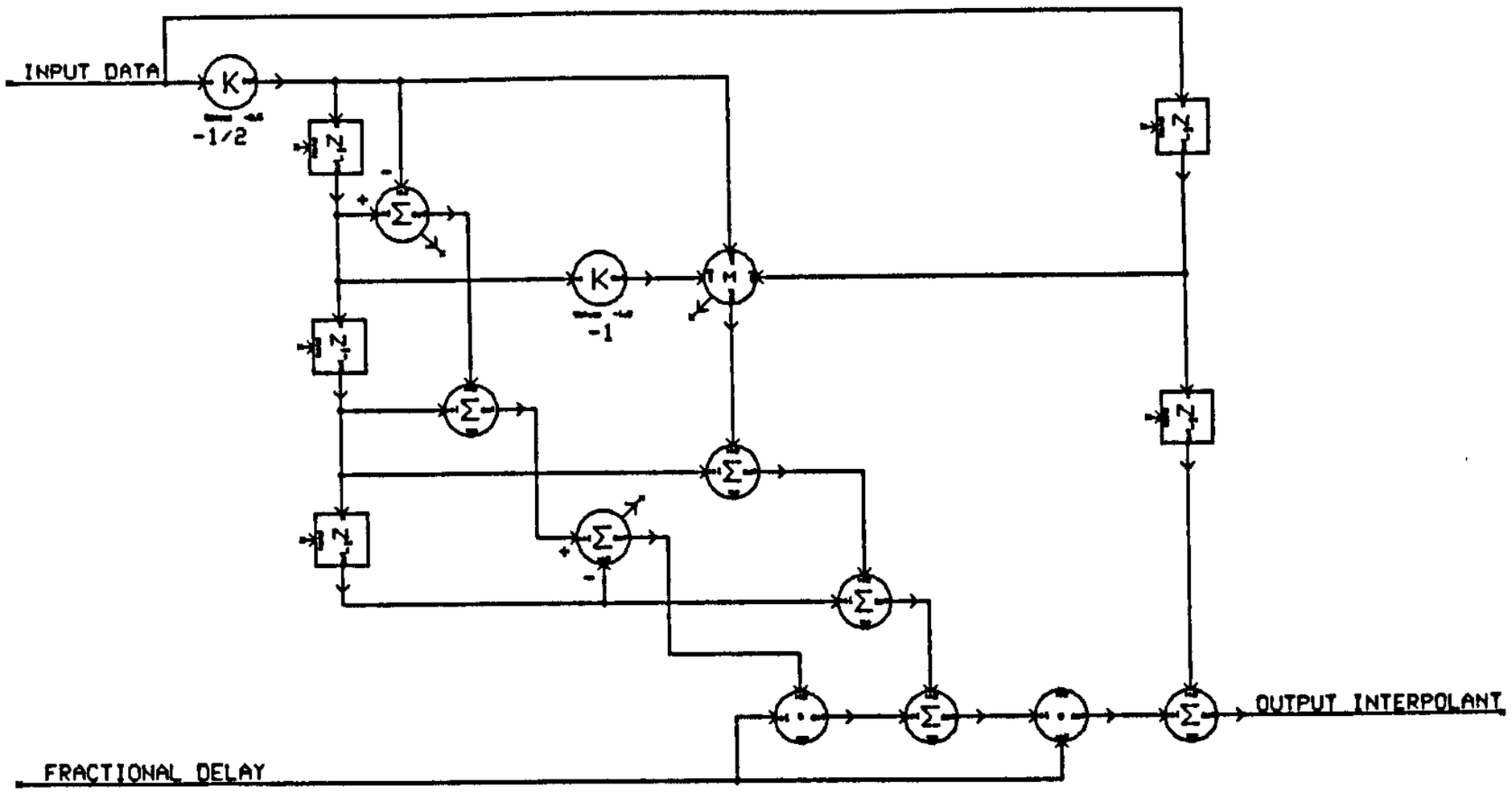



Figure 6.28: Block Diagram of the four-point quadratic Farrow interpolator.

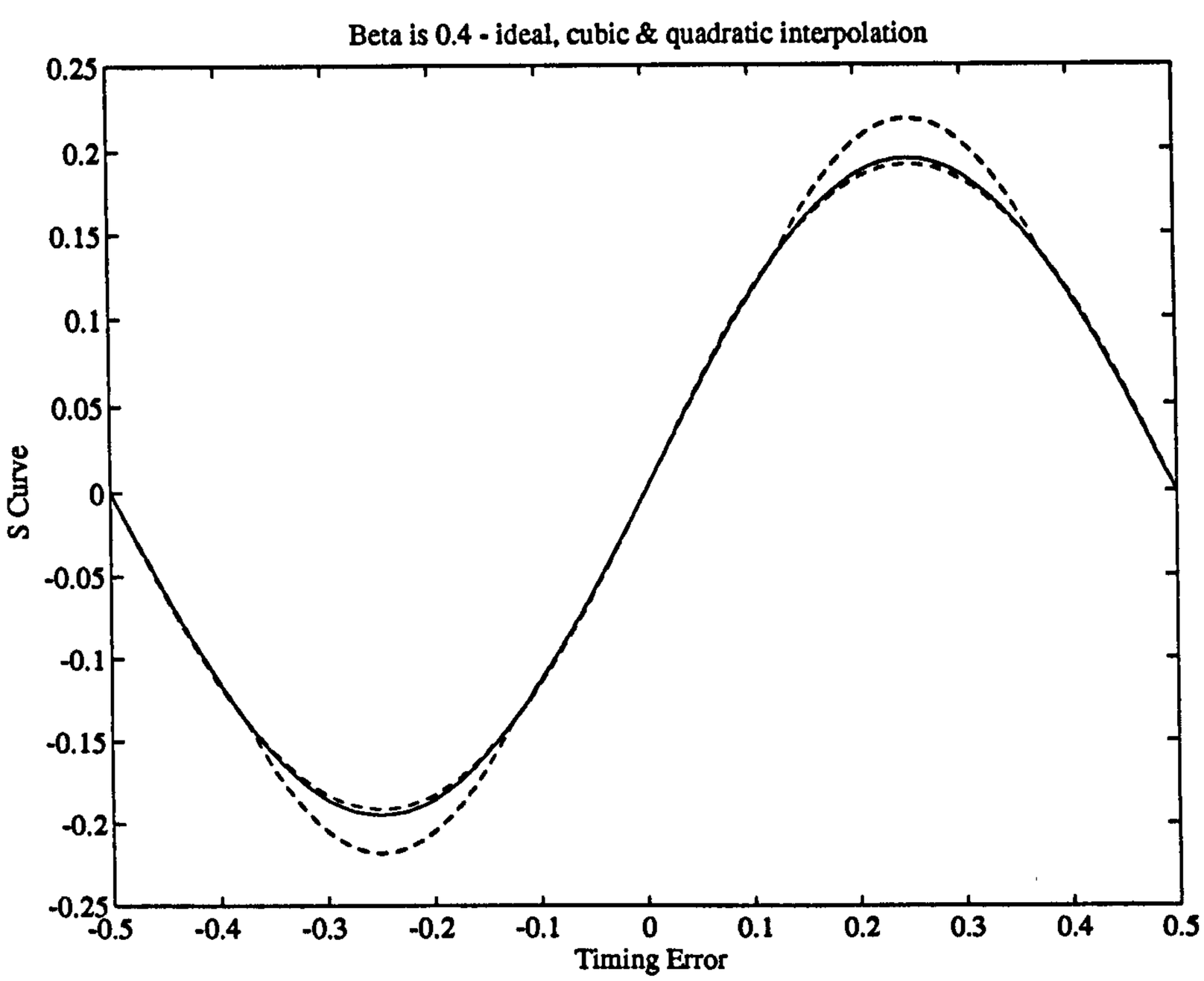
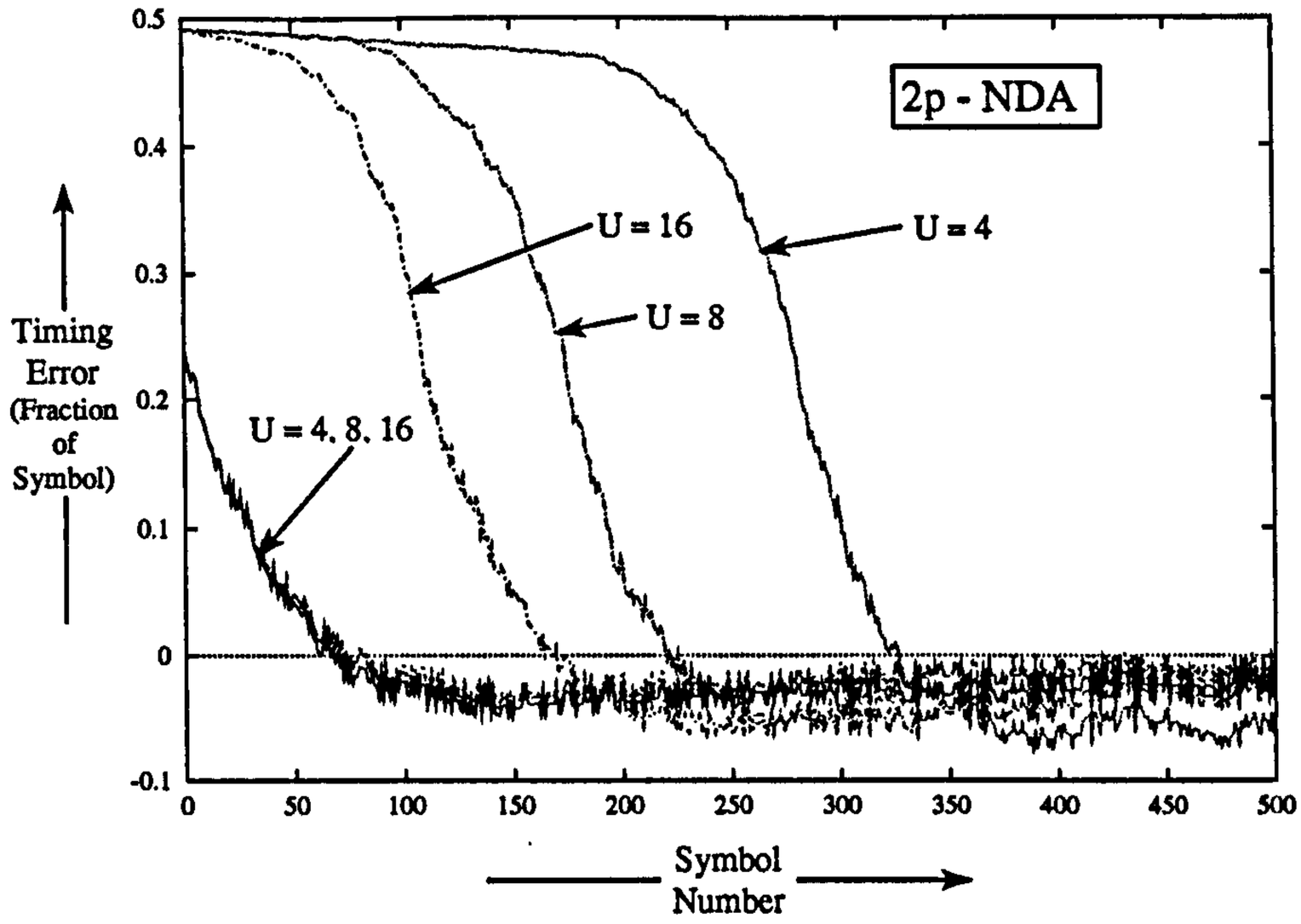
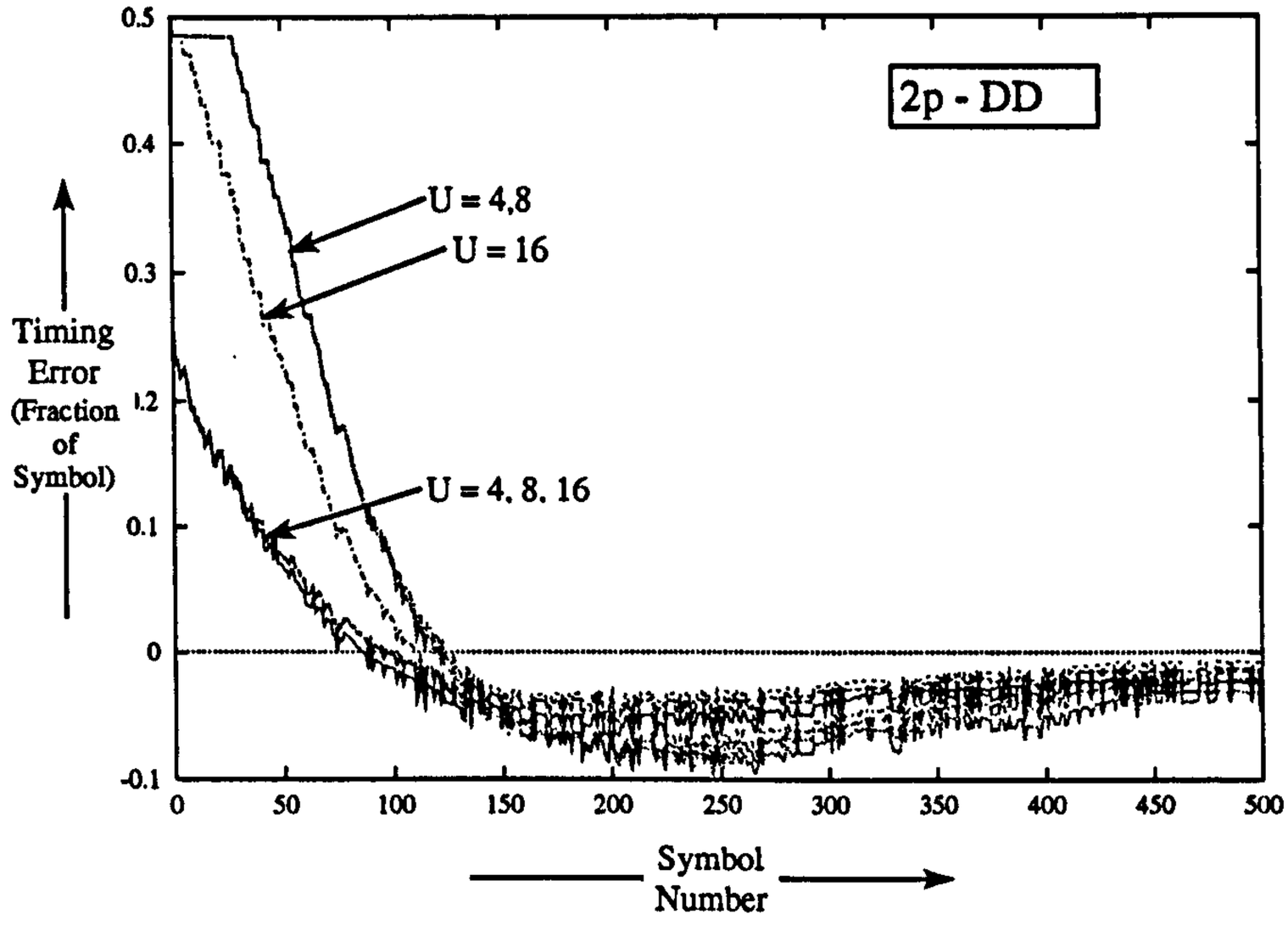



Figure 6.29: Simulated S-Curves for ideal (solid curve) and cubic and quadratic Farrow interpolation preceding Gardner algorithm for RC-QPSK with $\beta = 0.4$. Note that the line closest to the solid curve is that for cubic interpolation.




(a) Non Data-Aided




(b) Decision-Directed

Figure 6.30: Hangup in asynchronous timing recovery loops using a polyphase filter as the interpolator. The effect of hangup is exacerbated by having the interpolation factor $U$ too small

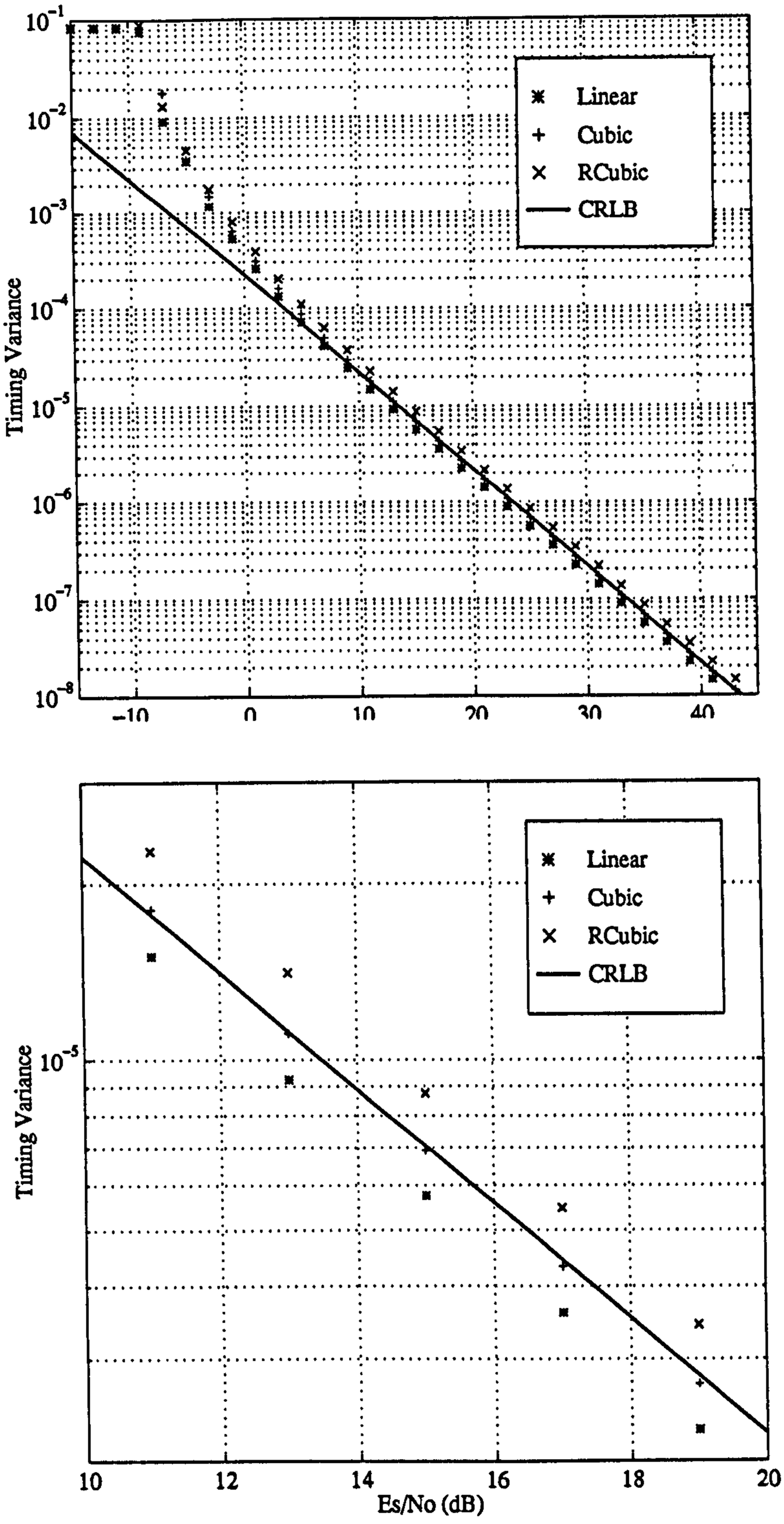



Figure 6.31: Comparison of the effect of interpolator order on tracking performance

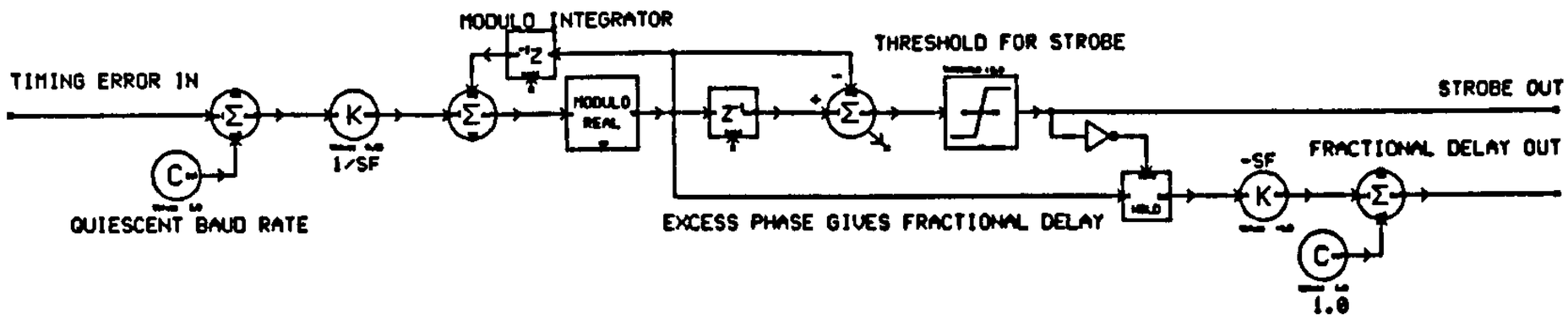



Figure 6.32: SPW Implementation of the Gardner NCO.

### 6.8.1 Control via NCO

An NCO based on the work in [15] has been built in SPW and is shown in figure 6.32. The NCO has two outputs; one is a clock output and the other represents the excess-phase in the NCO which is used to control the fractional delay element. This is illustrated by examining the waveforms out of the NCO as shown in figure 6.33. One can understand the operation of the Gardner NCO by examination of the traces in figure 6.33. The triangular outputs in traces S1 and S5 represent the clock tics and the dotted lines represent the instantaneous phase of the NCO. Note that if the instantaneous phase of the NCO is $t_0 = 0$ or $t_0 = 0.75$ of a sample then the clock tics are still output at the same instant. One finds out what the fractional interpolation should be by examining the excess phase of the NCO as in traces S2 ($t_0 = 0$) and S3 ($t_0 = 0.75$). Note however that if the initial phase of the NCO is set to be $t_0 = -0.75$ then this is accommodated for by a bulk delay of 1 (the NCO slips one sample) plus an excess phase of 0.25. This can be seen by examining traces S4 and S5 in figure 6.33.

## 6.9 Closed-Loop Performance : Acquisition for Small Offsets

Having determined the gain of our timing error detectors we are now in a position to configure our sampled symbol timing recovery loop in such a way that it will behave according to standard PLL theory, for small signals and small values of loop noise bandwidth at least. The SPW implementation of the sampled symbol timing recovery loop is as shown in figure 6.34 where all the system components have been discussed elsewhere as part of this work. The system shown is using the 2pNDA Gardner timing error detector but it is straightforward to replace this with any of the other timing error detectors. Knowledge of the timing error detector gain, $K_{\tau\tau}$, the required loop bandwidth, $B_L$, and whether the loop is to be first or second order, and if second order the required value of damping factor $\gamma$, will allow one to set the filter coefficients $C_1$ and $C_2$ in a manner discussed previously.

The symbol timing recovery loops then behave according to standard PLL theory as is illustrated by the following small-signal acquisition results. Figures 6.35 & 6.36 show acquisition results for the five timing error detector schemes discussed in this chapter operating on a raised cosine filtered QPSK data stream with excess bandwidth of $\alpha = 0.2$ using a first order loop. Figures 6.37 & 6.38 show similar results but for an excess bandwidth of $\alpha = 0.8$. Figures 6.39 & 6.42 show the results obtained using a second order loop (with $\gamma = 1/2$). All

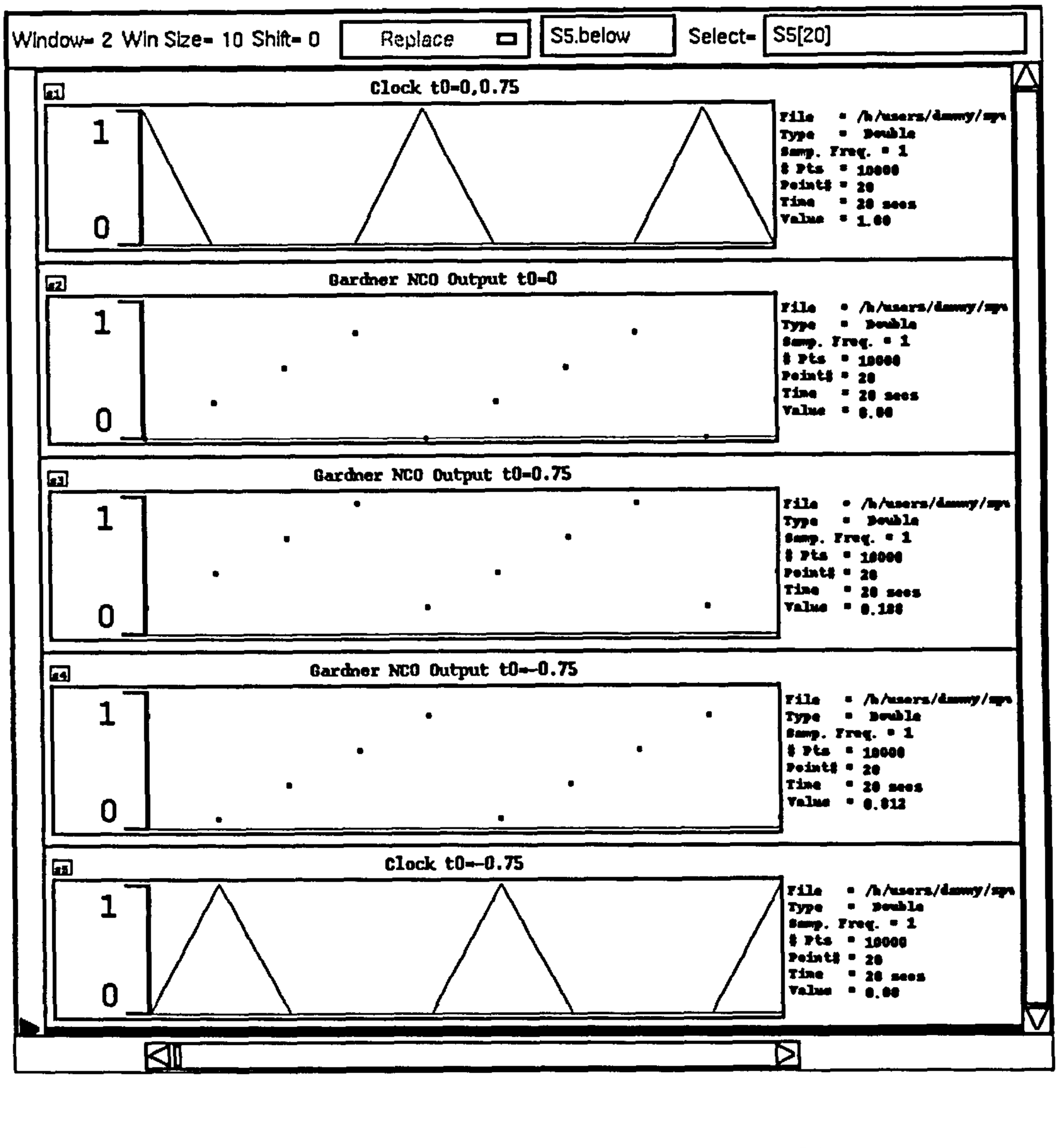



Figure 6.33: Waveforms out of the Gardner NCO.

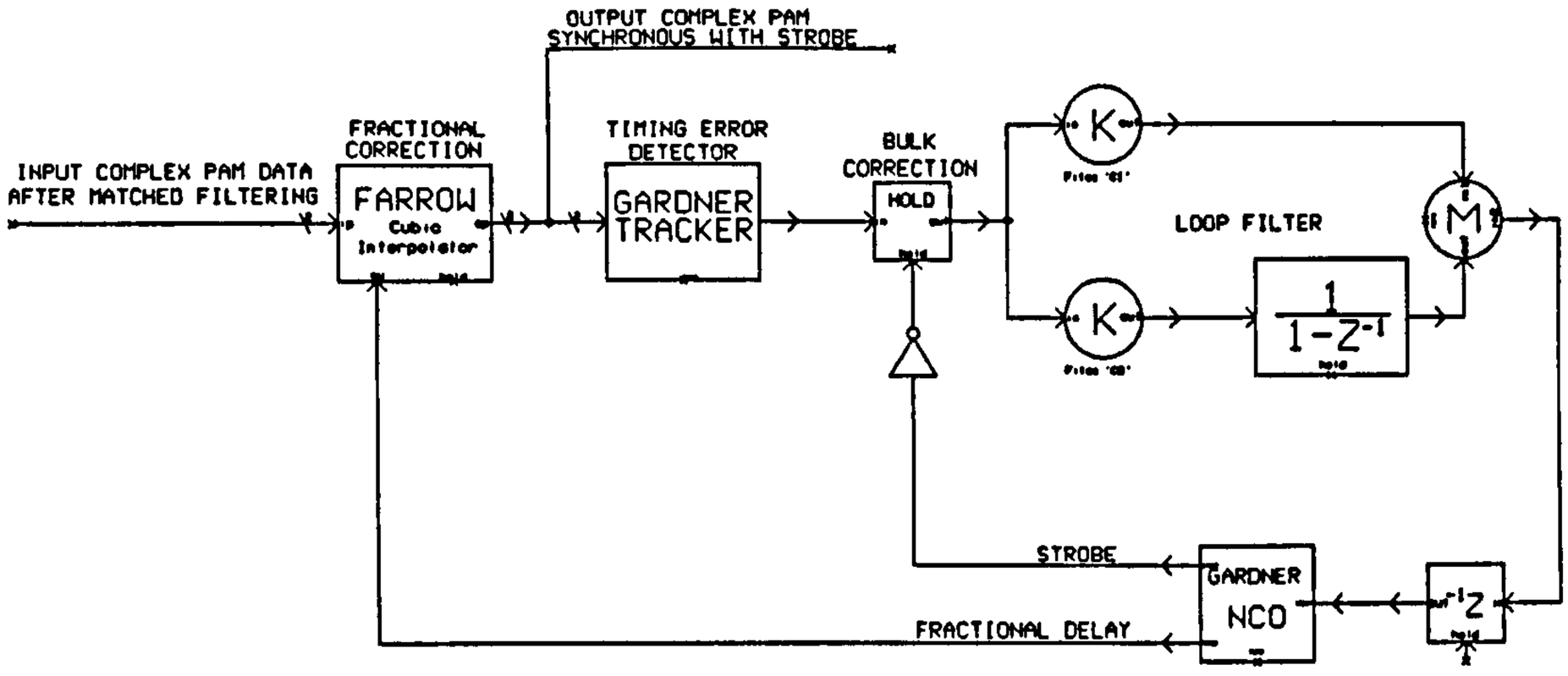



Figure 6.34: SPW model of Symbol-Timing Recovery Loop.

the acquisition results are compared with the theory which as we saw previously for the first order loop is

$$\phi(t) = e^{-4B_L t} \tag{6.96}$$

and for a second order system with $\gamma = 1/2$ is

$$\phi(t) = \left(\cos(\sqrt{3}B_L t) - \frac{\sqrt{3}\sin(\sqrt{3}B_L t)}{3}\right) e^{-B_L t} \tag{6.97}$$

It can be noted from the curves that some schemes are more perturbed by self-noise than others. There is no additive noise in the results shown; any jitter is wholly due to self-noise.

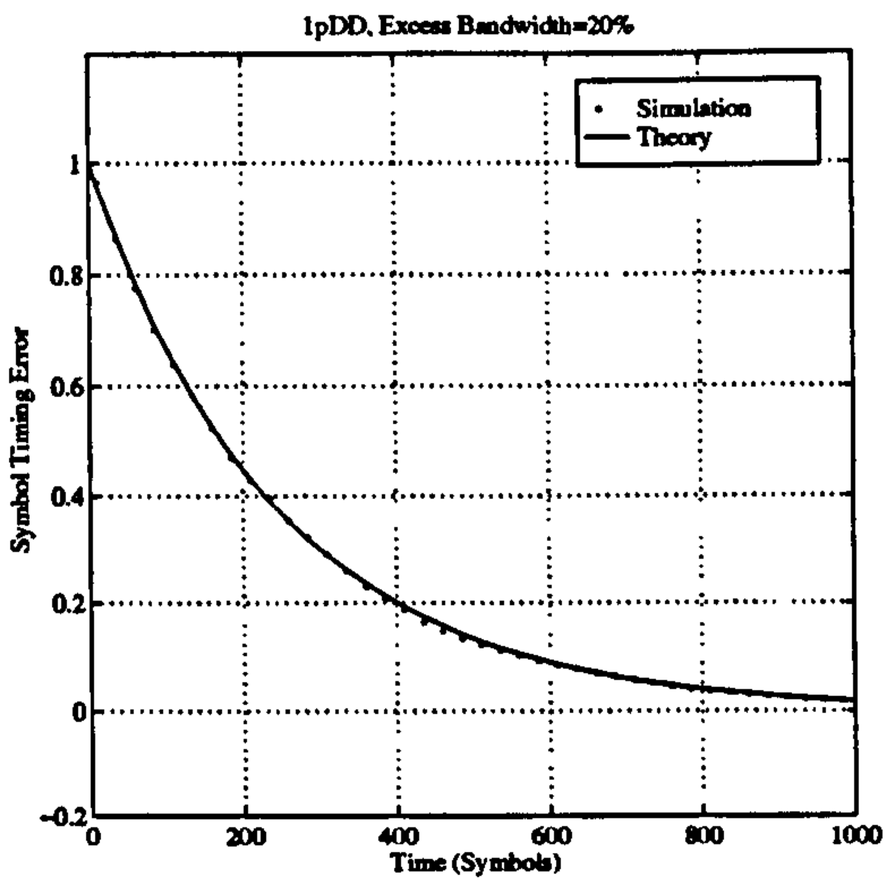
These results validate the modelling approach taken as part of this work.

### 6.9.1 Effect of Interpolator Upon Tracking Performance

It was clear from the discussion relating to the operation of the interpolator and NCO previously, that the NCO must provide both a bulk (strobe) and a fractional delay. If the actual required delay is close to a whole number of samples then it is possible for the fractional delay to switch between 0–1 samples whilst the NCO will drop/stuff strobes. This phenomena will cause extra jitter in the loop as is illustrated in figure 6.43. This suggests that the tracking performance measures will actually depend upon the *absolute* timing phase offset. In reality there will be a small frequency offset between transmitter and receiver (even if it is of the order of parts per million) and so this effect should be averaged out. It does suggest though that the averaged tracking variance results may be some way off the theory due to this effect.

### 6.9.2 Acquisition for Timing-Frequency Offsets

Usually timing frequency offsets are small, of the order of parts per million, and therefore less of a problem than carrier frequency offsets. However, the symbol timing recovery loops




(a) $T$-Spaced,Decision-Directed

(b) $T$-Spaced, Amplitude-Directed

Figure 6.35: Acquisition Performance of 1p-DD & 1p-AD timing recovery schemes vs theory. Noise equivalent bandwidth is 0.001. Excess bandwidth = 20%. First order loop. Modulation is QPSK.

will still pull-in a frequency offset. They still exhibit all the behaviour of standard PLLs as indicated by figure 6.44. The figure illustrates the frequency acquisition performance of, in this case, the 2pNDA timing error detector with frequency errors of $1.0e-04/T$ & $1.0e-03/T$. Note that the first of these is small enough to acquire without cycle-slipping, being within the hold-in range of the loop, whereas the second frequency offset results in cycle-slipping. Being a high-gain second order loop it does eventually pull in the timing-phase (modulo $T/2$).

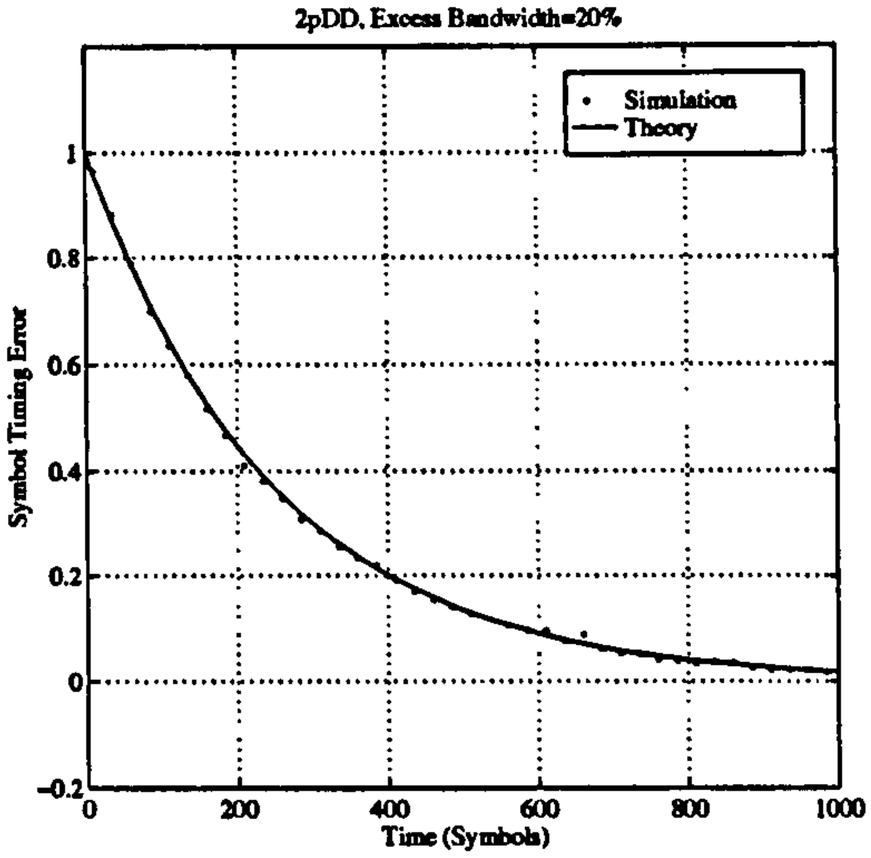
## 6.10 Closed-Loop Performance : Tracking

The tracking performance of the sampled symbol timing recovery loop for each of the five timing error detectors is as shown in figures 6.45 to 6.46 for the first order loop and figures 6.47 to 6.48 for the second order loop with $\gamma = 1/2$. The modulation scheme is raised cosine filtered QPSK with excess bandwidths of $\alpha = 0.1, 0.5, 0.9$. Also plotted with the results is the theoretical Cramer-Rao bound for timing error detection. Actually, this is the *modified* Cramer-Rao bound as derived recently in [28].

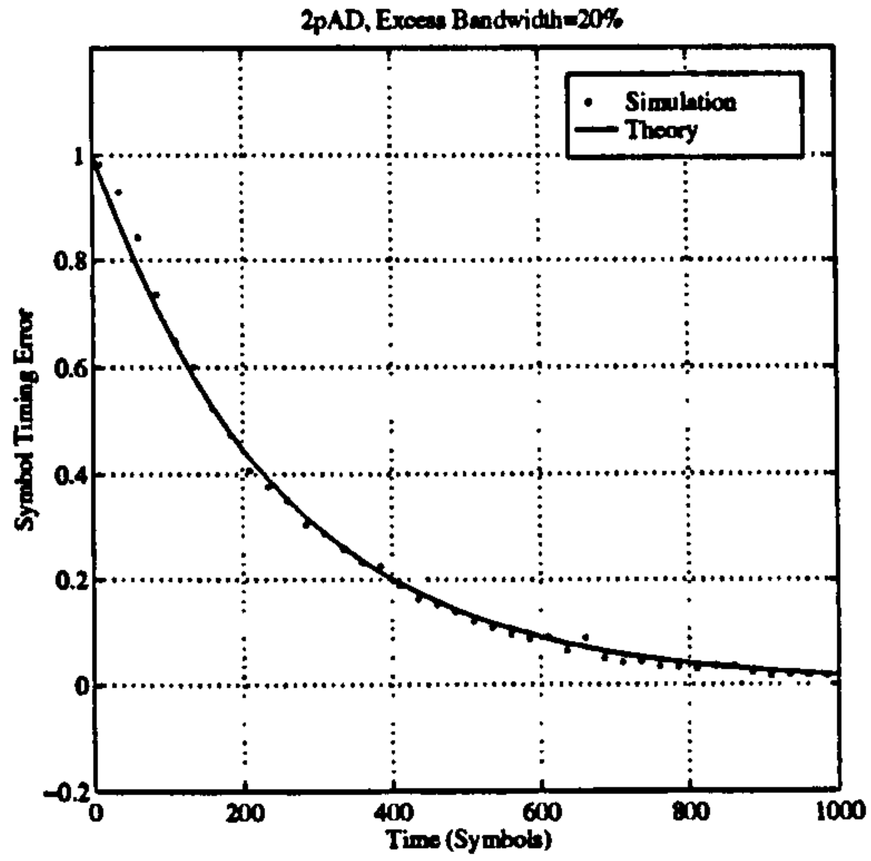
The modified Cramer-Rao bound proves to be useful when the observed data depends upon a number of parameters other than the parameter one is attempting to estimate. The modified Cramer-Rao bound for timing recovery is found to be

$$\sigma^2_{\text{CRB}} = \frac{B_L T}{4\pi\eta^2} \frac{T^2}{E_s/N_o} \tag{6.98}$$

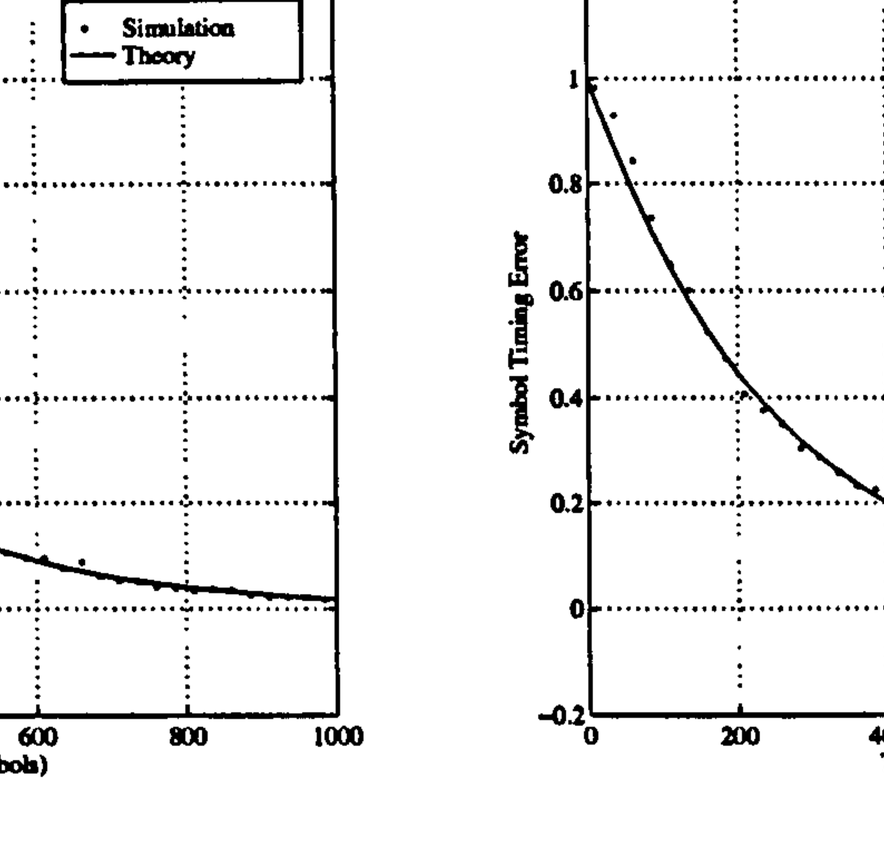
where $\eta$ is a dimensionless coefficient depending only on the shape of the modulating frequency




(a) $T/2$-Spaced,Decision-Directed

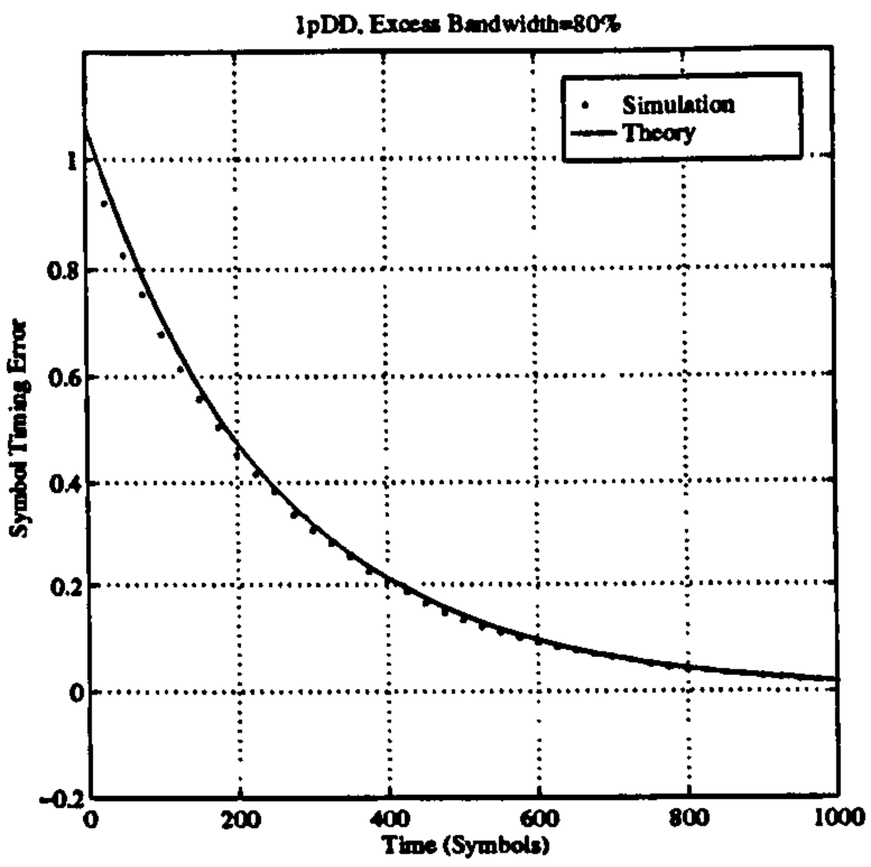



(b) $T/2$-Spaced,Amplitude-Directed




(c) $T/2$-Spaced,Non data Aided

Figure 6.36: Acquisition Performance of 2p-DD, 2pAD & 2p-NDA timing recovery schemes. Noise equivalent bandwidth is 0.001. Excess bandwidth = 20%. First order loop. Modulation is QPSK.




(a) $T$-Spaced,Decision-Directed

(b) $T$-Spaced,Amplitude-Directed

Figure 6.37: Acquisition Performance of 1p-DD & 1p-AD timing recovery schemes vs theory. Noise equivalent bandwidth is 0.001. Excess bandwidth = 80%. First order loop. Modulation is QPSK.

pulse $G(f)$ as

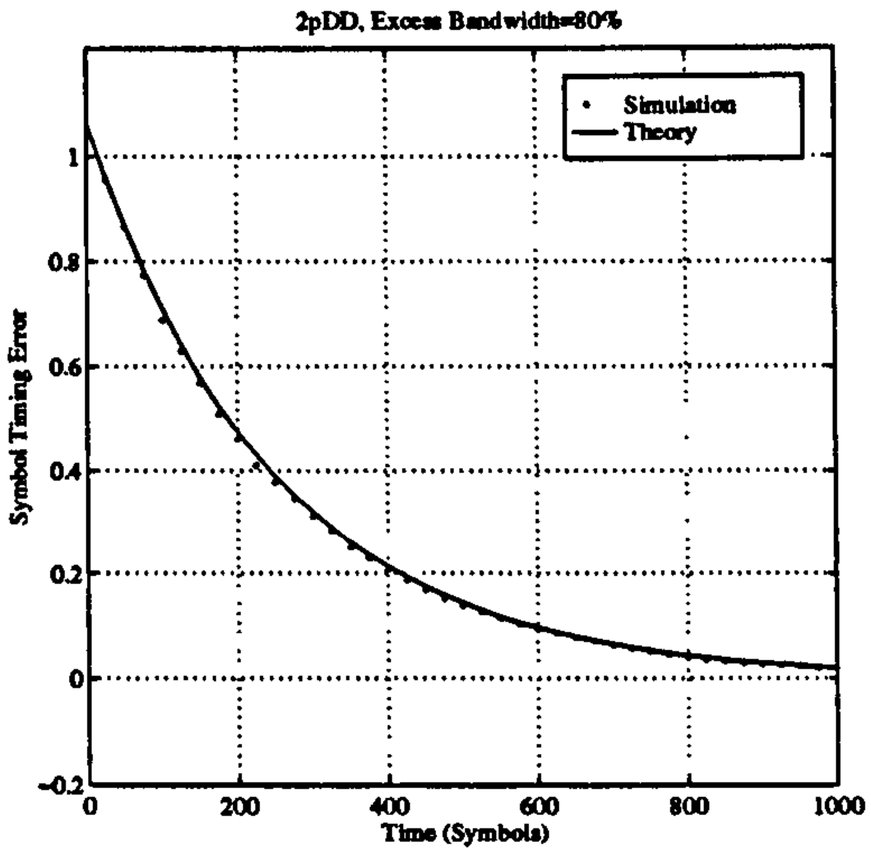
$$\eta = \frac{\int_{-\infty}^{+\infty} T^2 f^2 |G(f)|^2 \, df}{\int_{-\infty}^{+\infty} |G(f)|^2 \, df} \tag{6.99}$$

We have evaluated this for $G(f)$ corresponding to a raised cosine filter to give

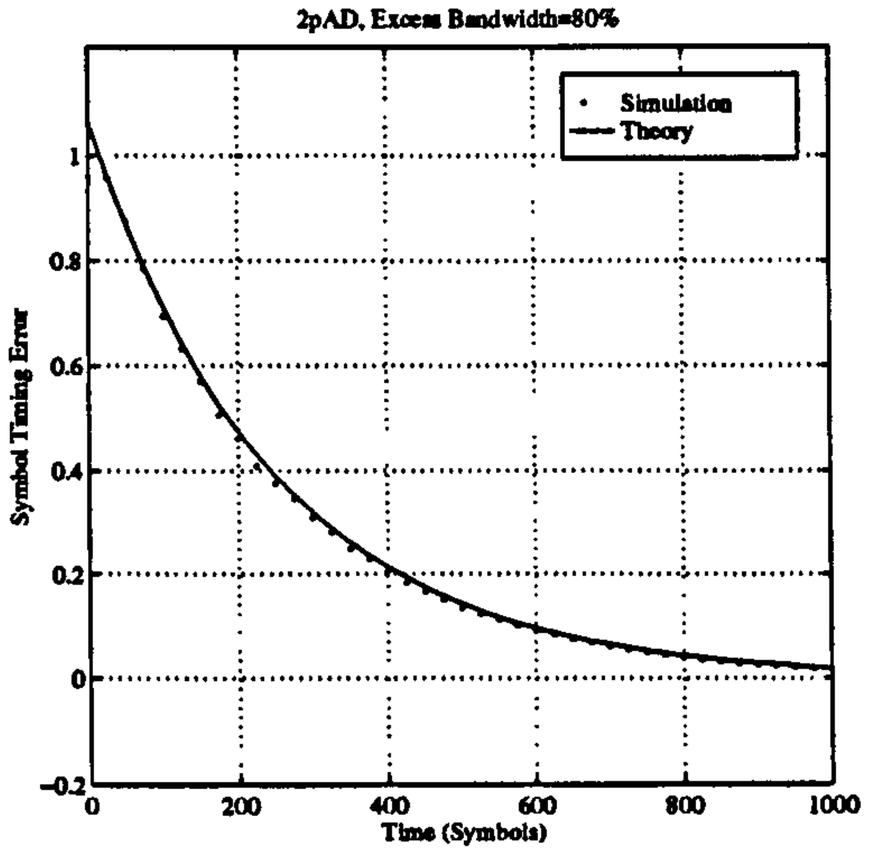
$$\eta = \frac{3\alpha^2\pi^2 - 24\alpha^2 + \pi^2}{12\pi^2} \tag{6.100}$$

This result is then used to plot the modified Cramer-Rao bound together with the simulation results in the following figures.

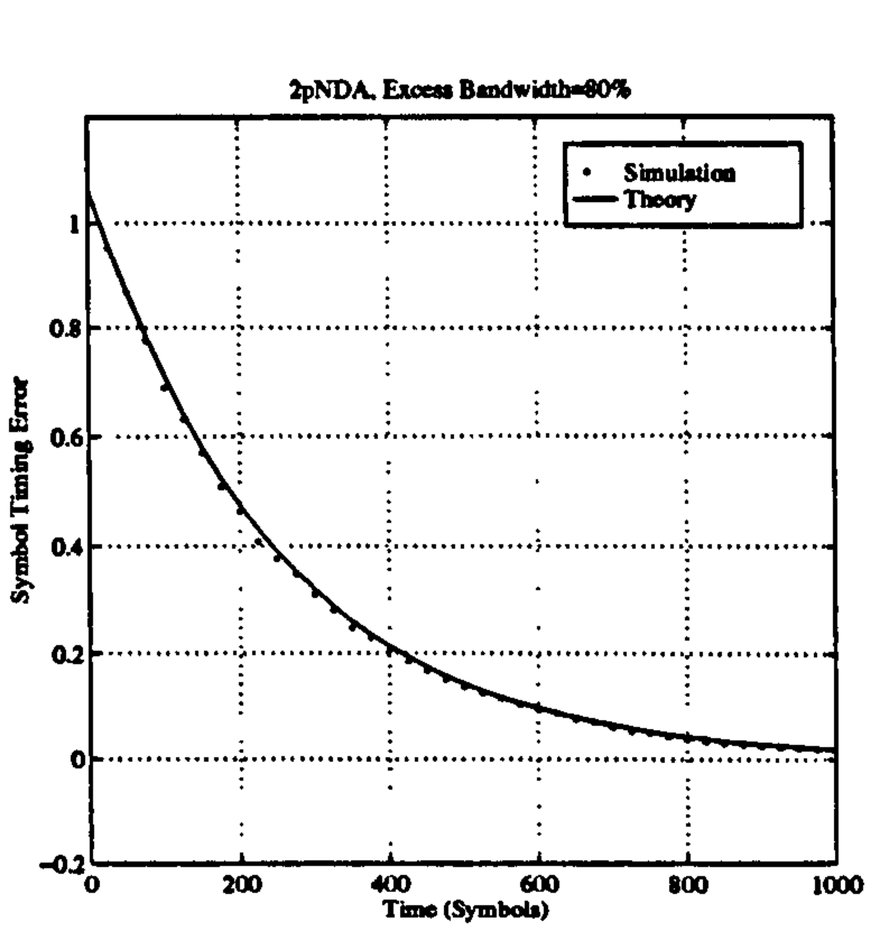
It is worth pointing out some salient features with regard to the tracking performance of the symbol timing recovery schemes. Not unsurprisingly, the tracking performance is linked with the gains of the various timing error detectors as plotted earlier in figure 6.13 as a function of the modulation excess bandwidth. In particular, the 2pNDA scheme has poor performance for low values of excess bandwidth. Also, all of the $T/2$-spaced schemes have performance which gets better for larger values of excess bandwidth, in contrast with the $T$-spaced schemes. The 2pDD scheme shows good tracking performance for all values of excess bandwidth as does the newly discovered 2pAD scheme for $M$-PSK modulations. The 2pAD schemes have performance second only to the 2pDD schemes, in terms of dBs degradation from the Cramer-Rao bound, for excess bandwidths larger than about $\alpha = 0.4$.




(a) $T/2$-Spaced,Decision-Directed

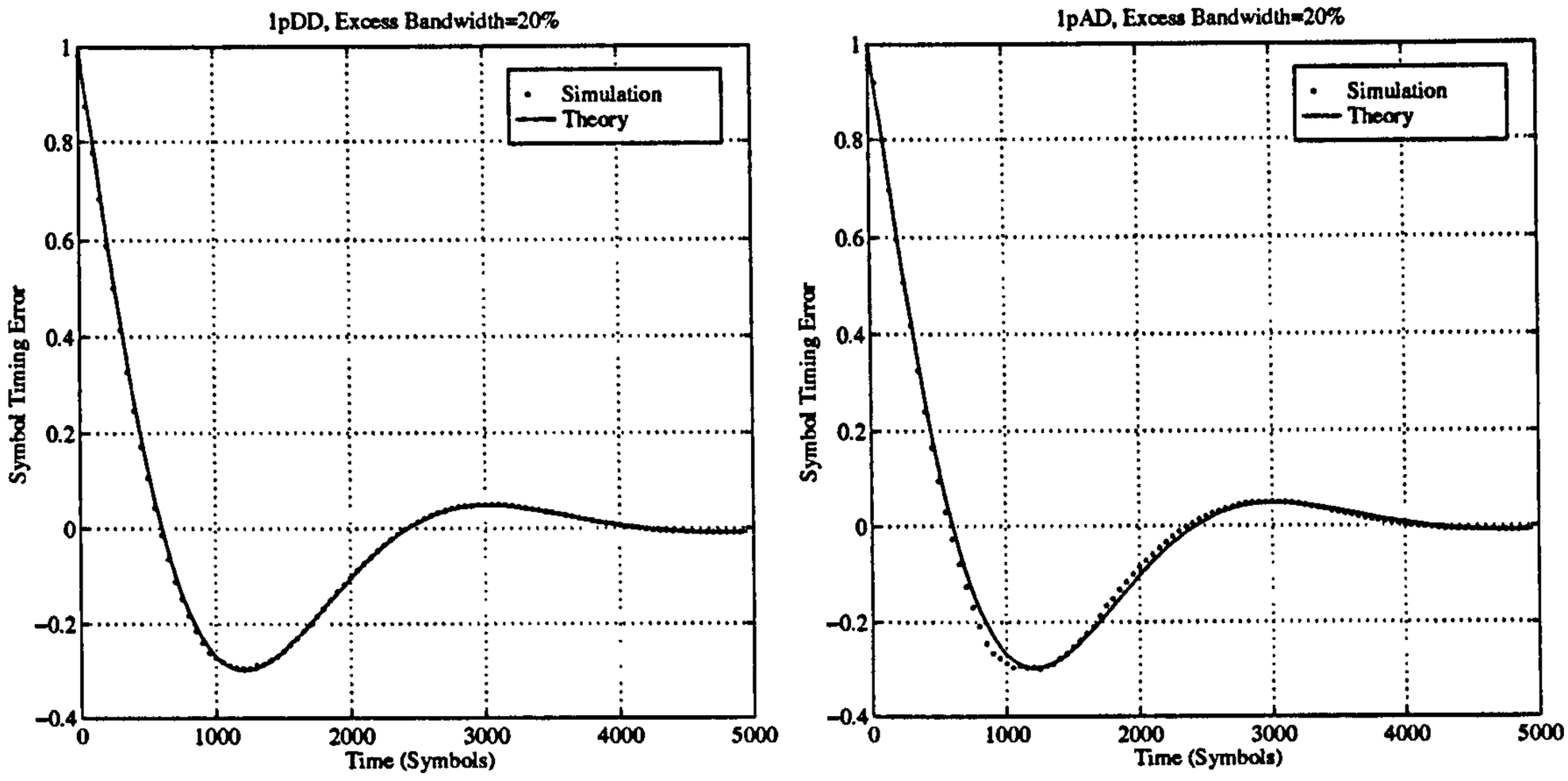



(b) $T/2$-Spaced,Amplitude-Directed




(c) $T/2$-Spaced,Non data Aided

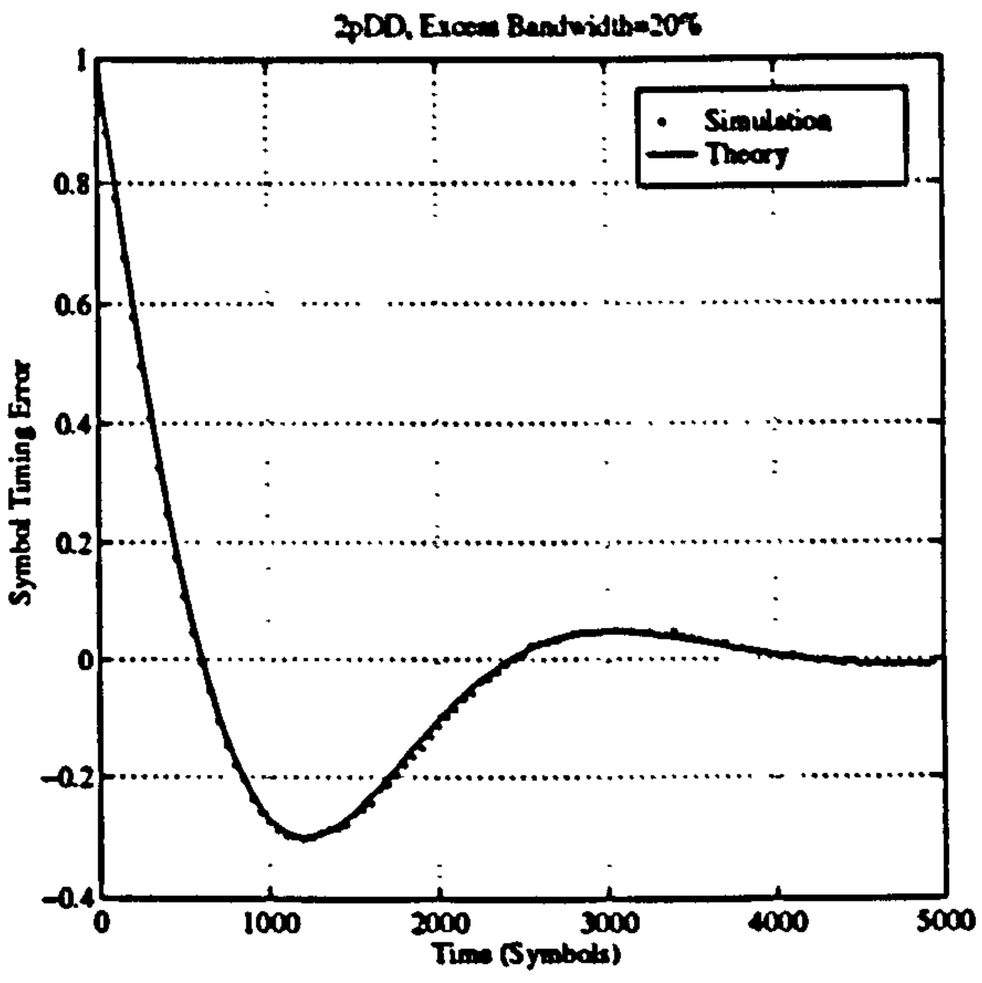
Figure 6.38: Acquisition Performance of 2p-DD, 2pAD & 2p-NDA timing recovery schemes. Noise equivalent bandwidth is 0.001. Excess bandwidth = 80%. First order loop. Modulation is QPSK.



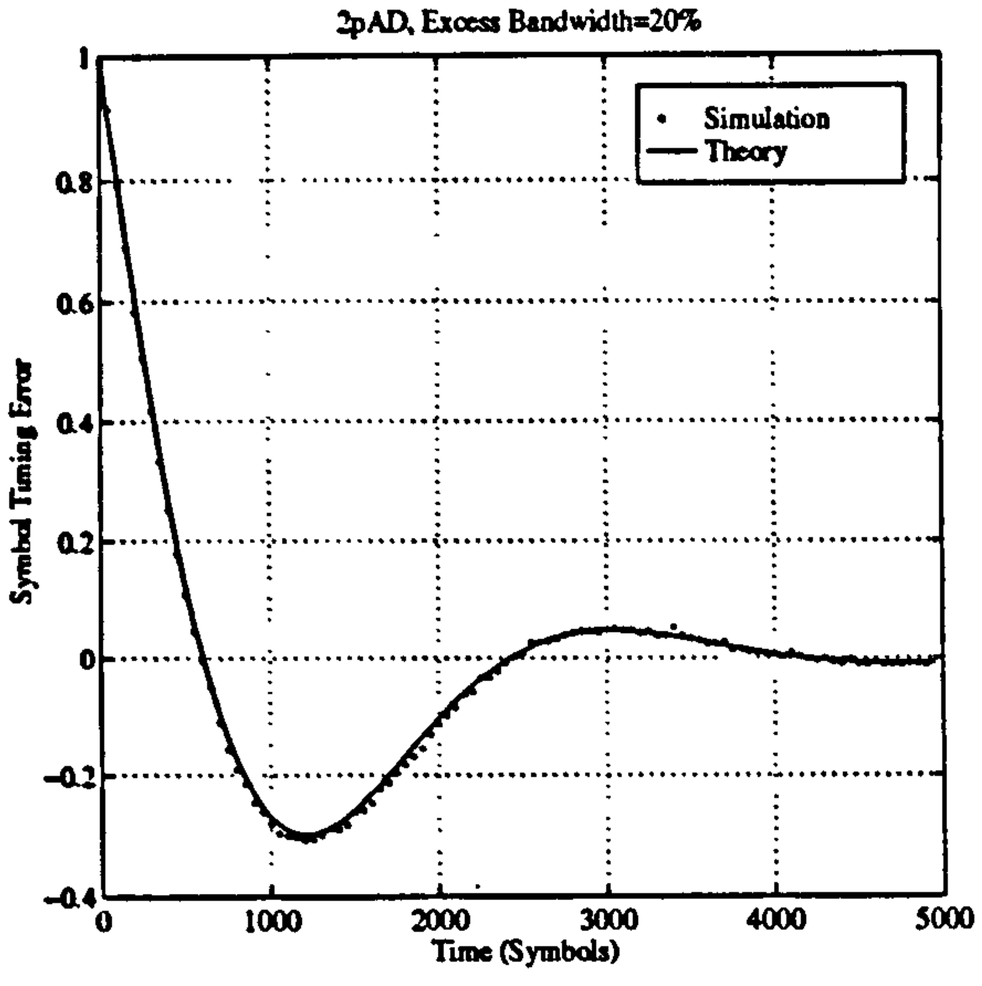

(a) $T$-Spaced,Decision-Directed

(b) $T$-Spaced, Amplitude-Directed

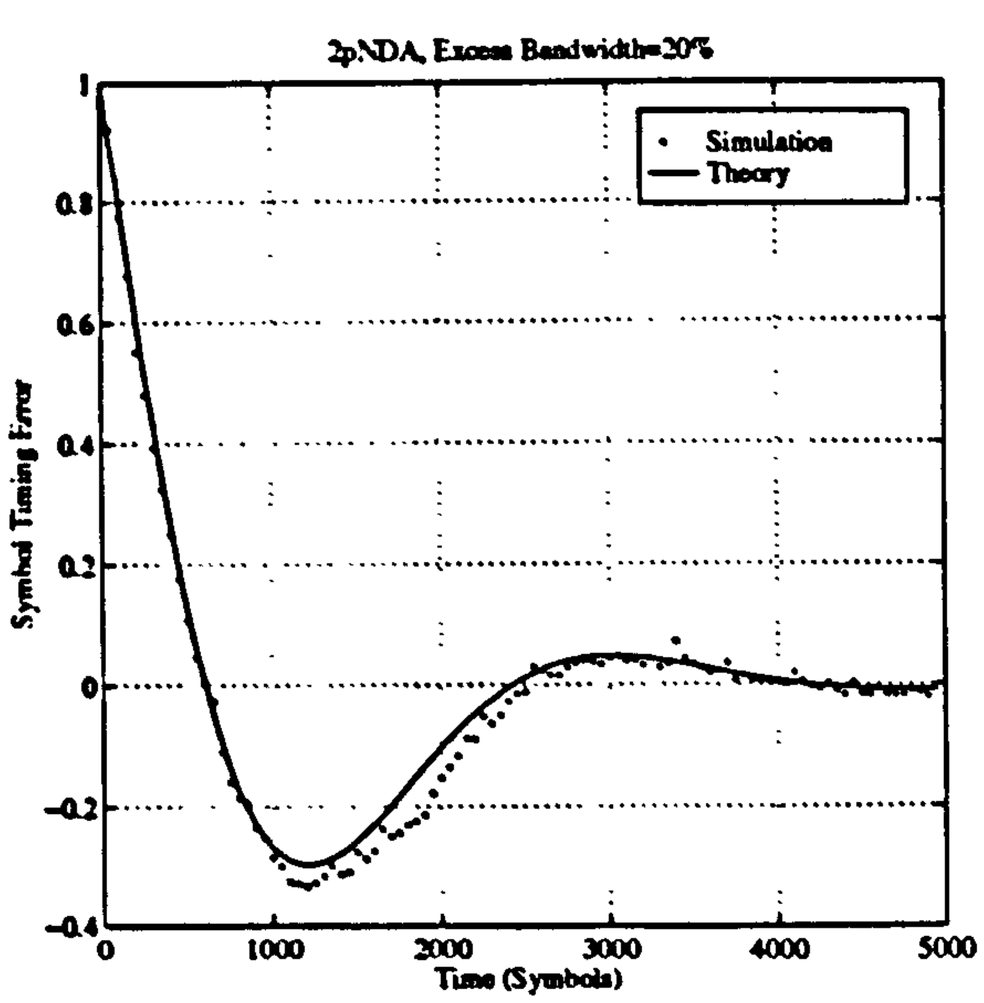
Figure 6.39: Acquisition Performance of 1p-DD & 1p-AD timing recovery schemes vs theory. Noise equivalent bandwidth is 0.001. Excess bandwidth = 20%. Second order loop. Modulation is QPSK.




(a) $T/2$-Spaced,Decision-Directed

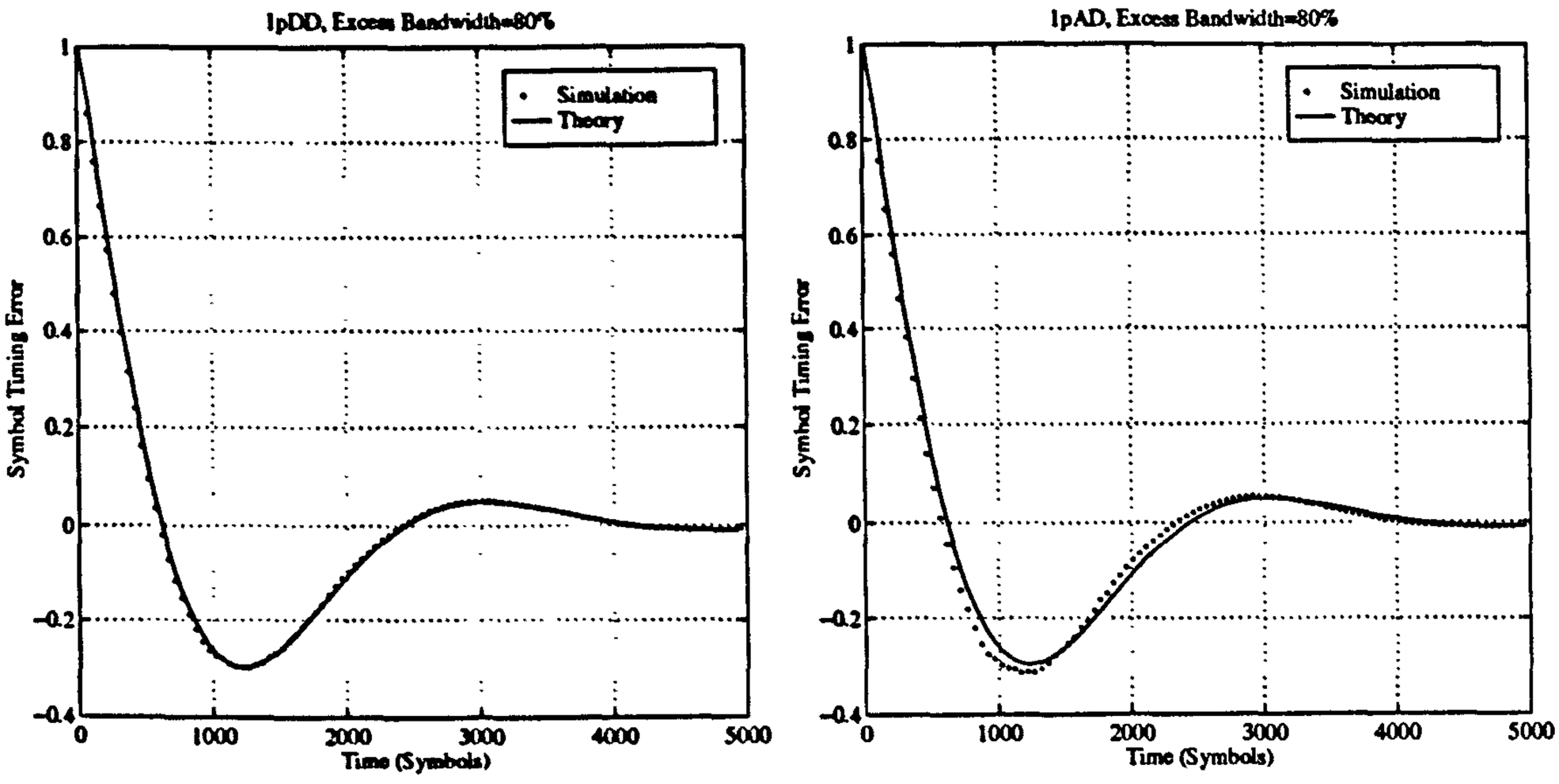



(b) $T/2$-Spaced,Amplitude-Directed




(c) $T/2$-Spaced,Non data Aided

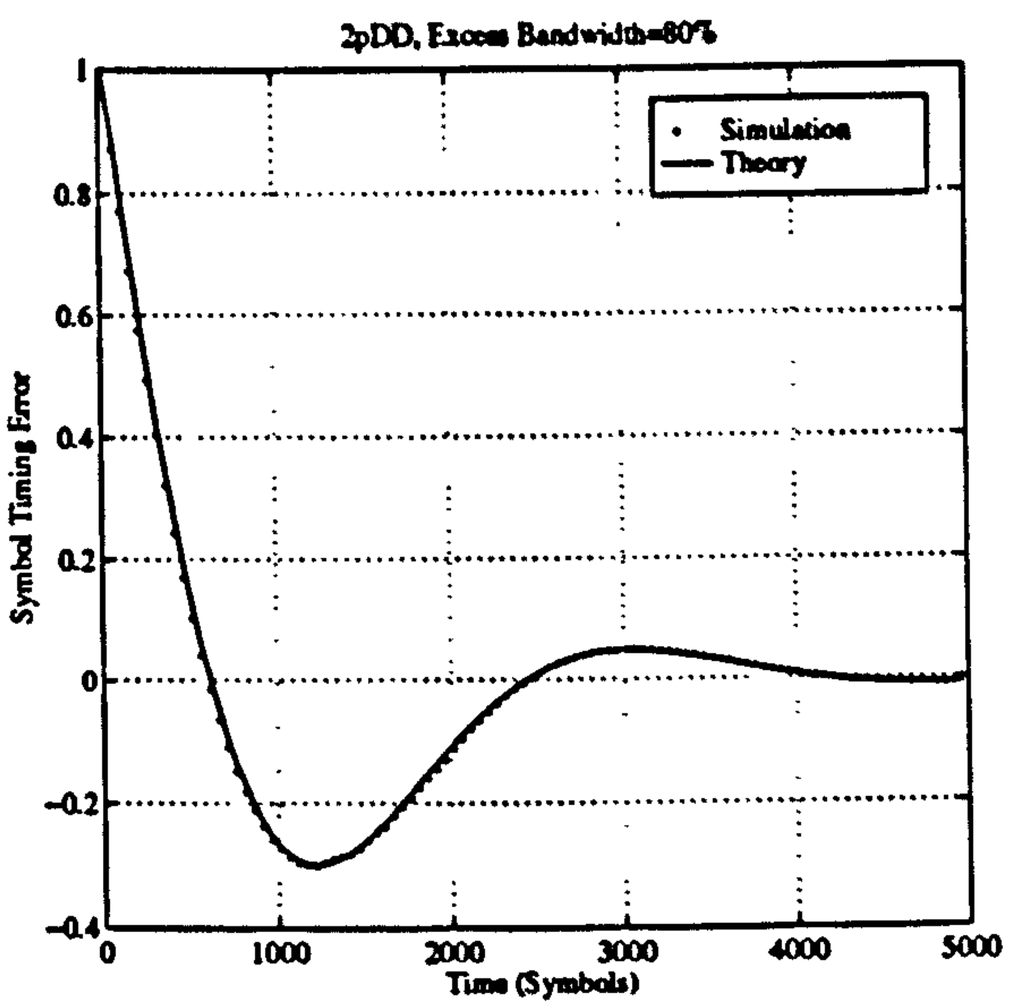
Figure 6.40: Acquisition Performance of 2p-DD, 2pAD & 2p-NDA timing recovery schemes. Noise equivalent bandwidth is 0.001. Excess bandwidth = 20%. Second order loop. Modulation is QPSK.



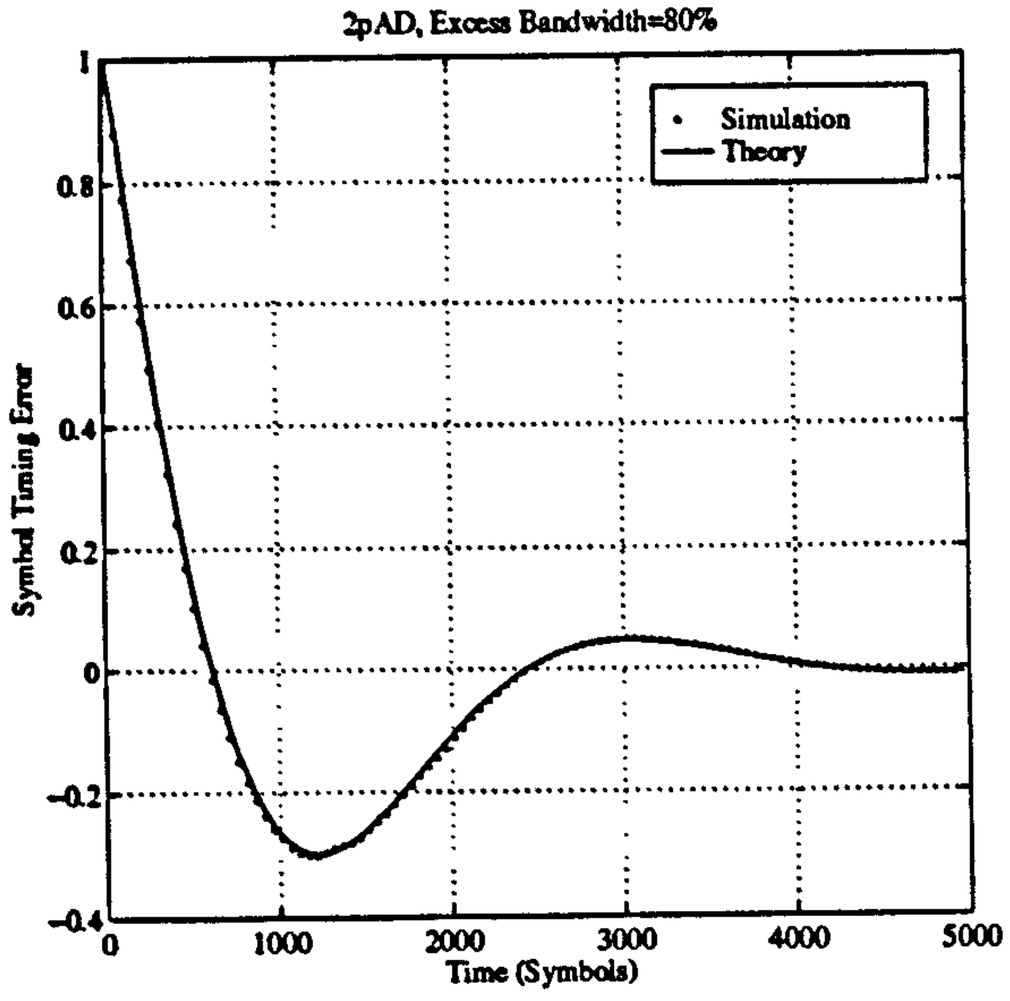

(a) $T$-Spaced,Decision-Directed

(b) $T$-Spaced,Amplitude-Directed

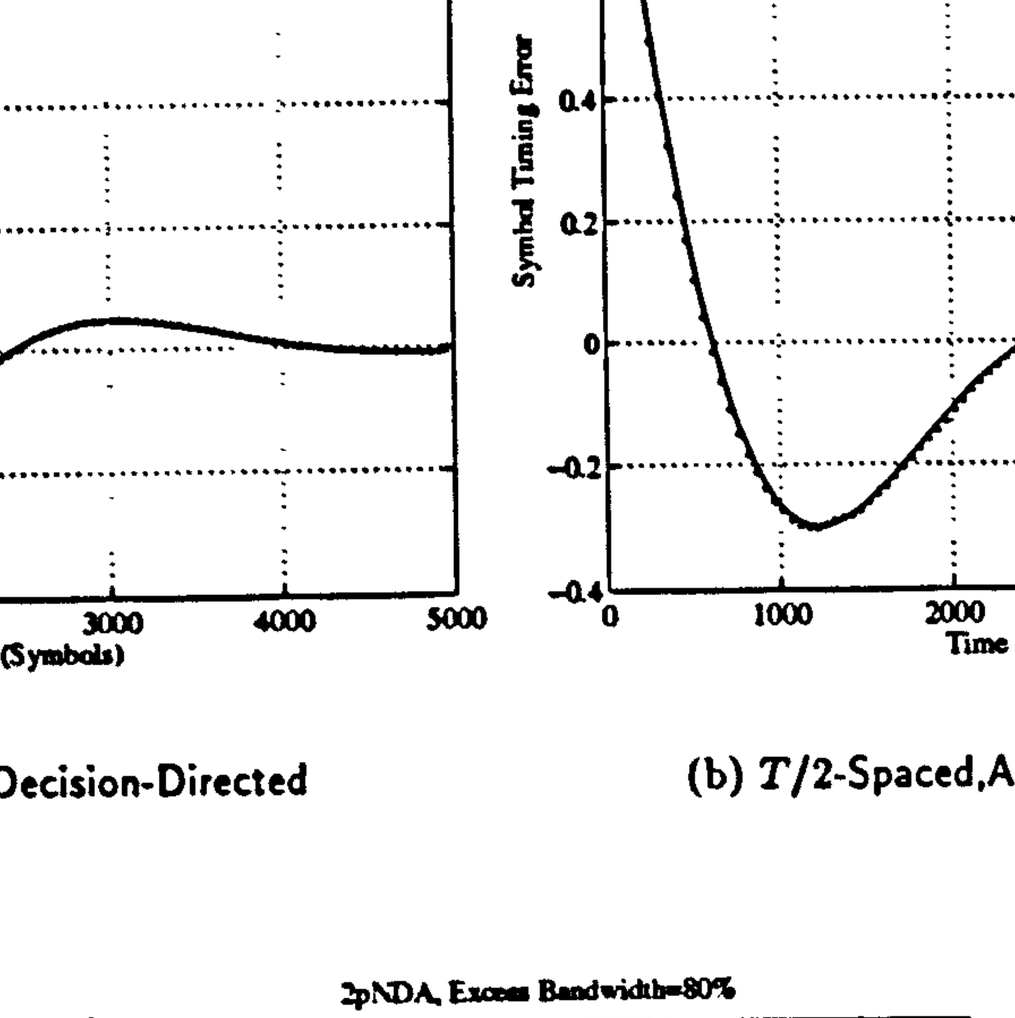
Figure 6.41: Acquisition Performance of 1p-DD & 1p-AD timing recovery schemes vs theory. Noise equivalent bandwidth is 0.001. Excess bandwidth = 80%. Second order loop. Modulation is QPSK.




(a) $T/2$-Spaced,Decision-Directed

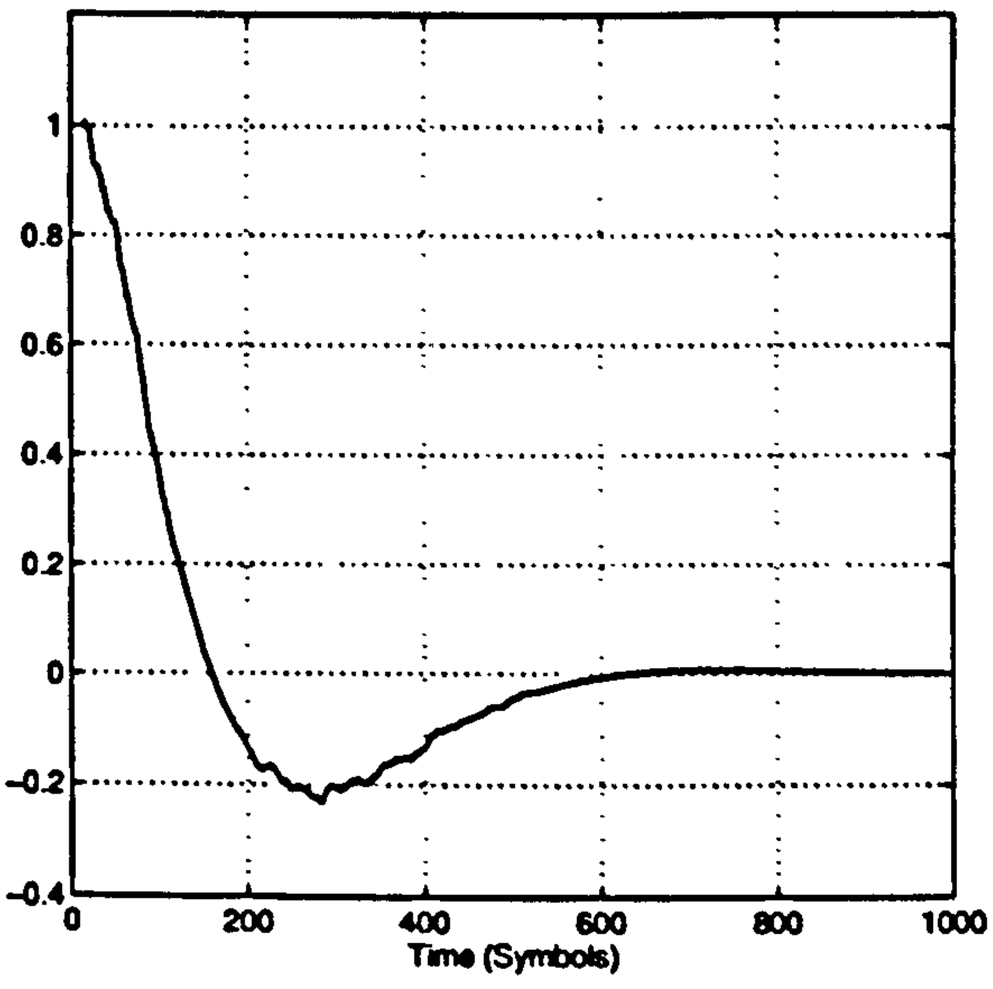



(b) $T/2$-Spaced,Amplitude-Directed

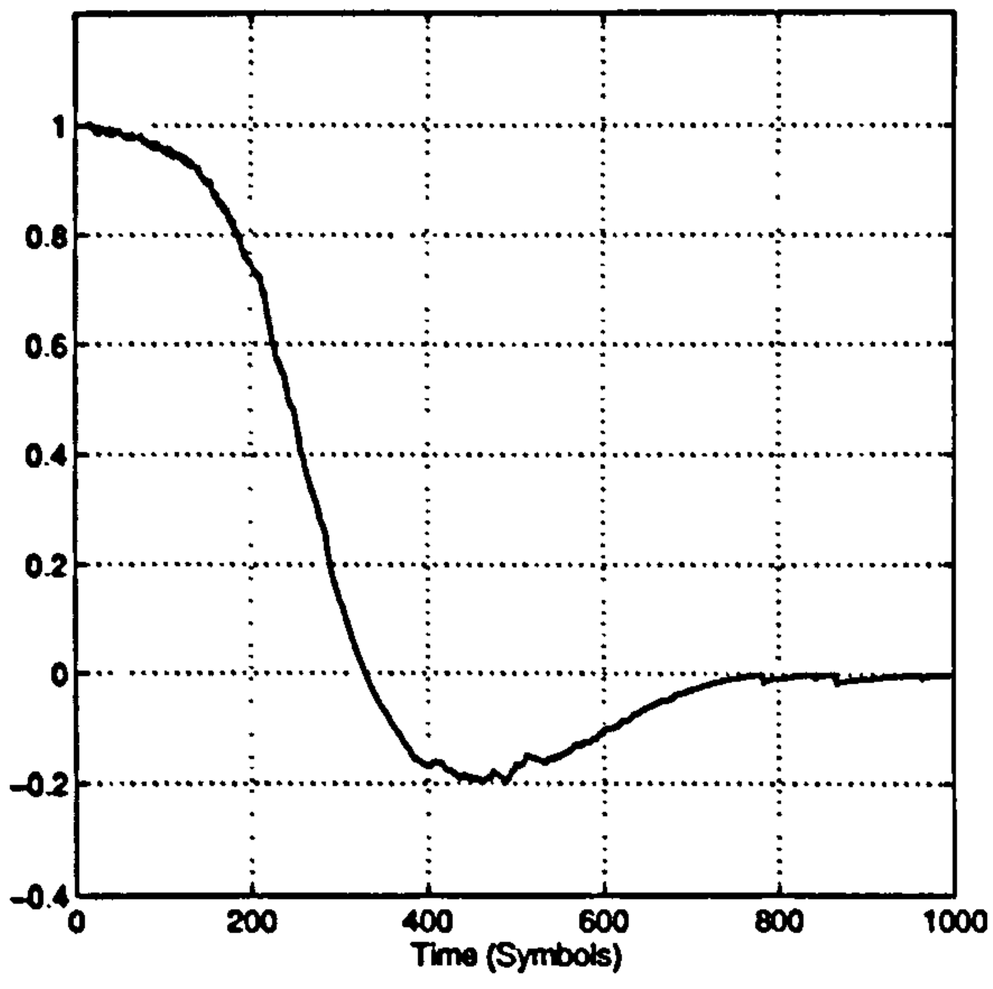



(c) $T/2$-Spaced,Non data Aided

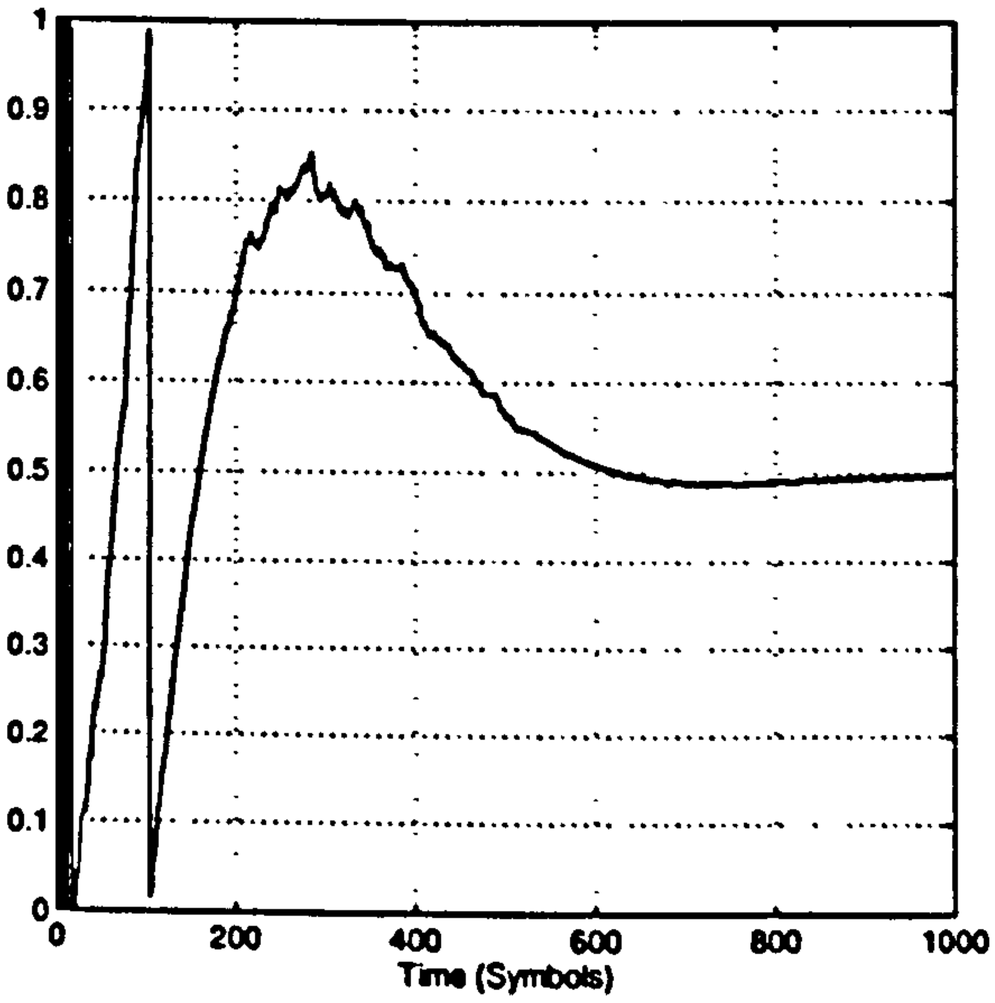
Figure 6.42: Acquisition Performance of 2p-DD, 2pAD & 2p-NDA timing recovery schemes. Noise equivalent bandwidth is 0.001. Excess bandwidth = 80%. Second order loop. Modulation is QPSK.



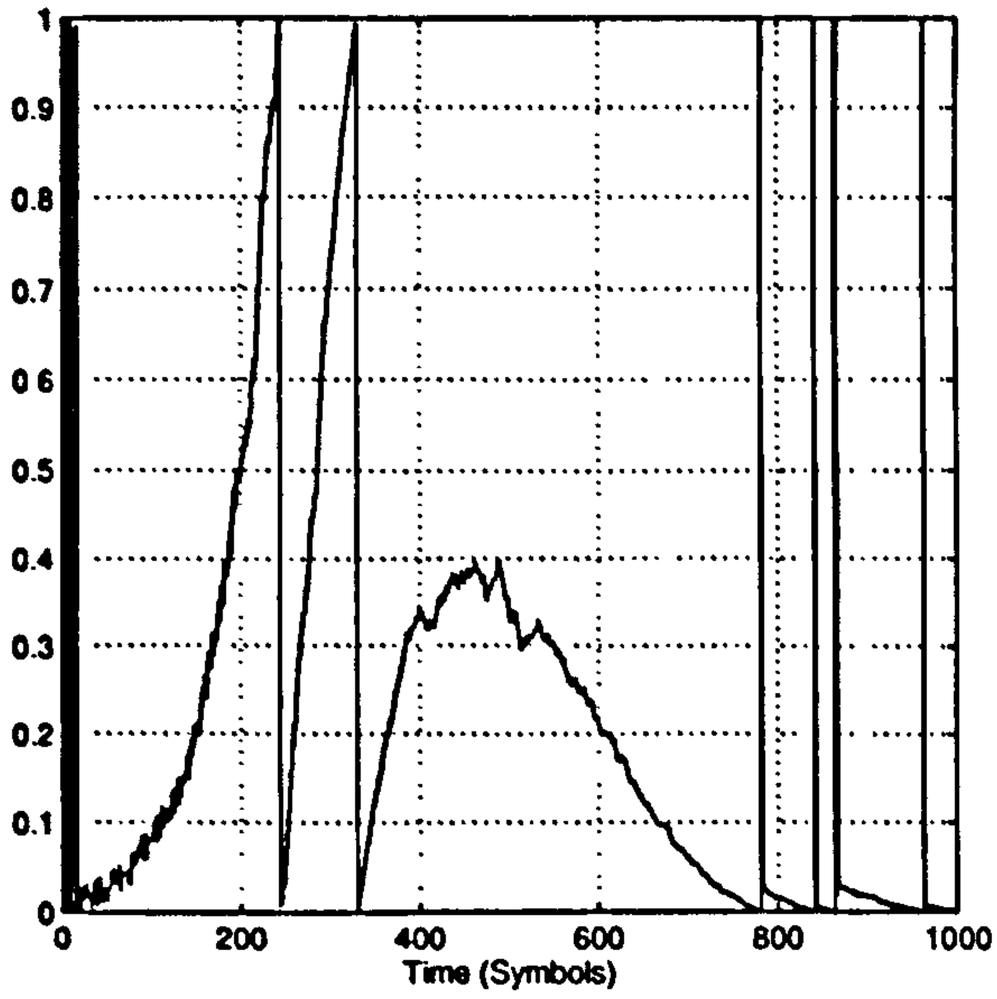

(a) Un-wrapped Timing Phase




(b) Un-wrapped Timing Phase

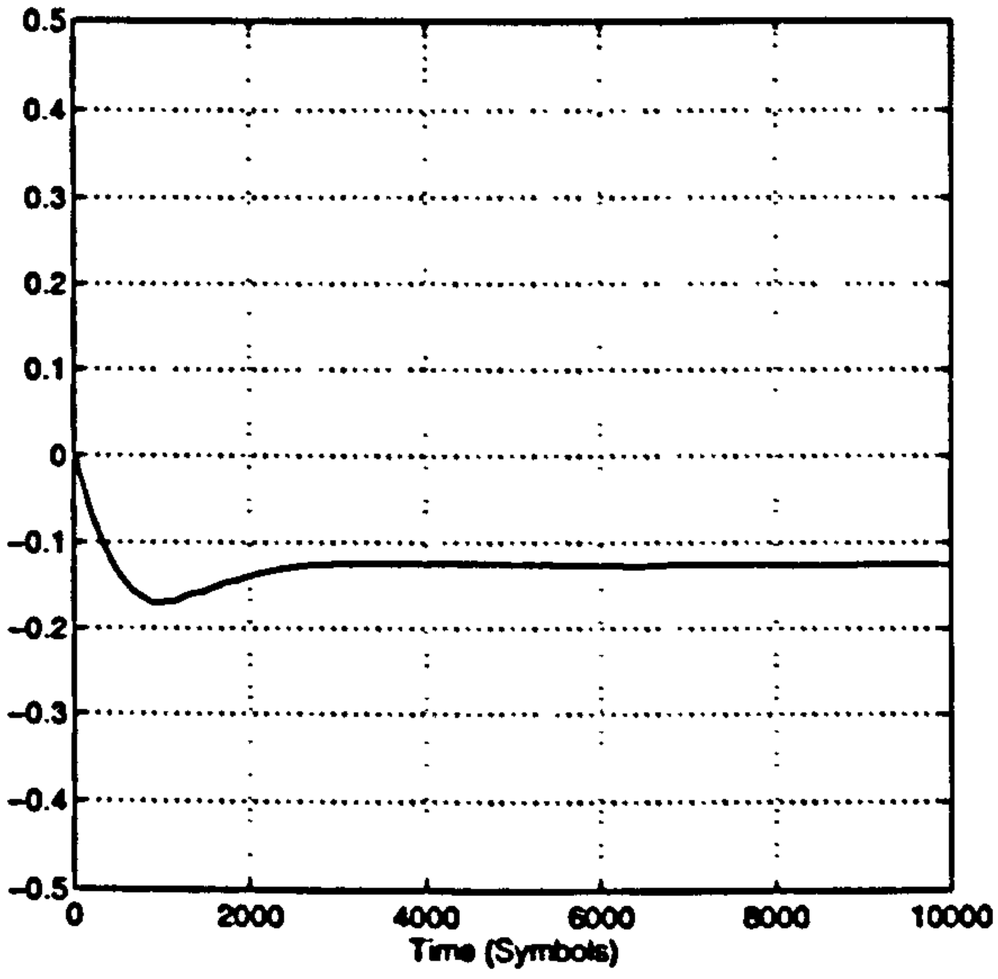



(c) Fractional Phase

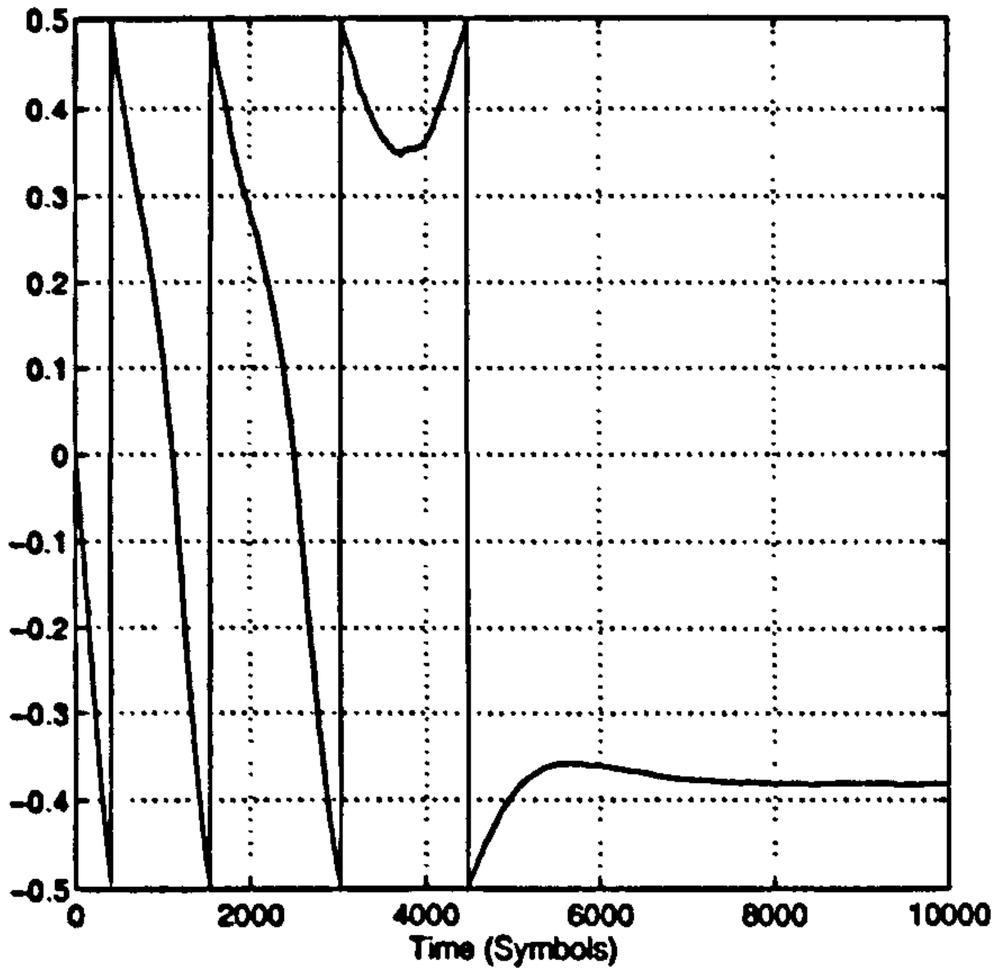



(d) Fractional Phase

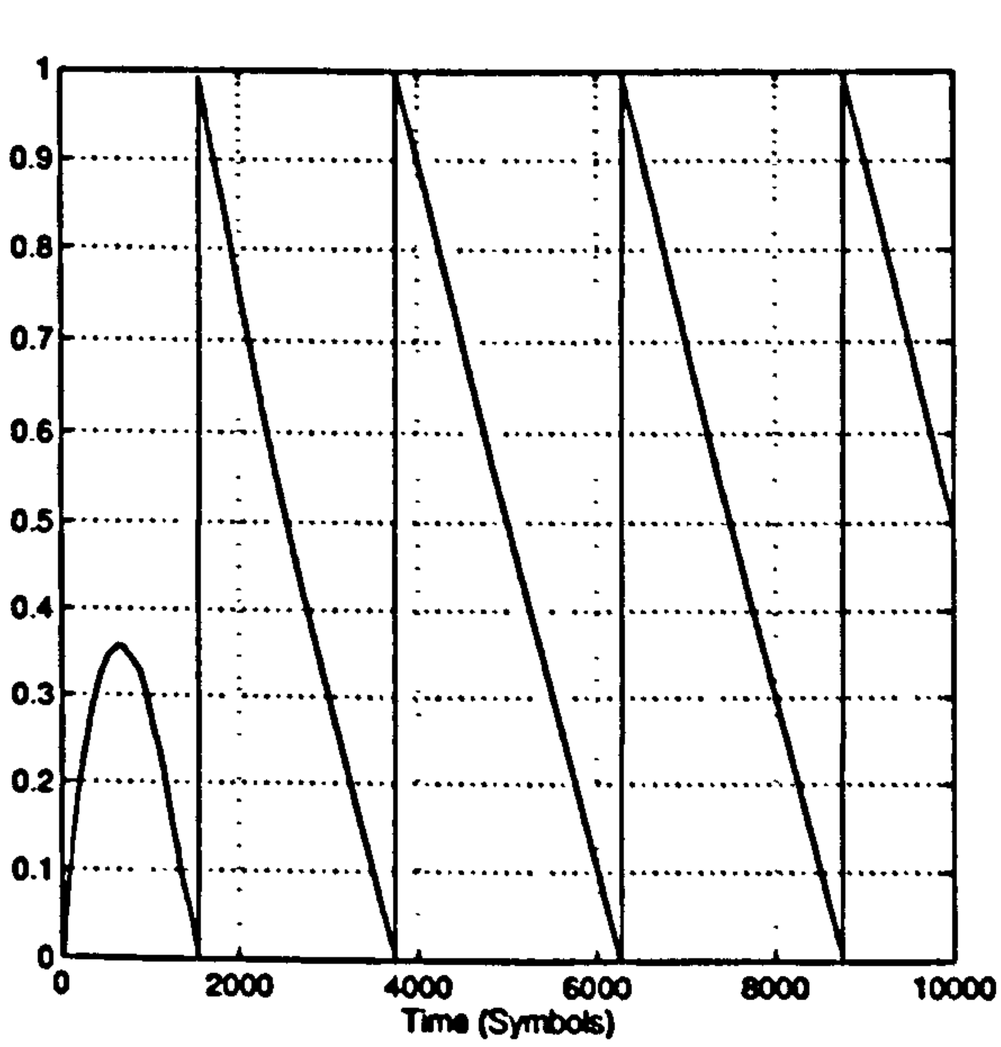
Figure 6.43: Jitter in asynchronous timing recovery loop due to delay required being a whole number of samples (right) and a fraction of a sample (left)



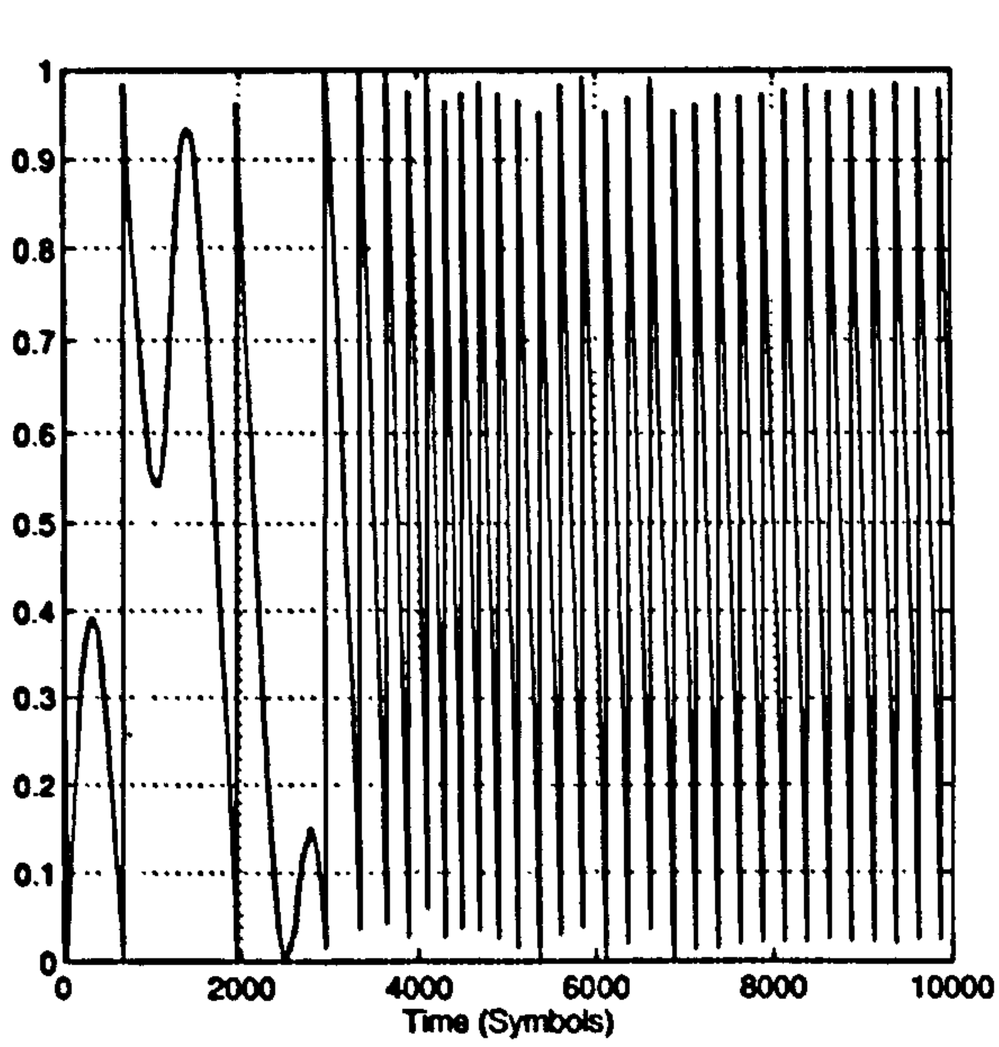

(a) Un-wrapped Timing Phase




(b) Un-wrapped Timing Phase

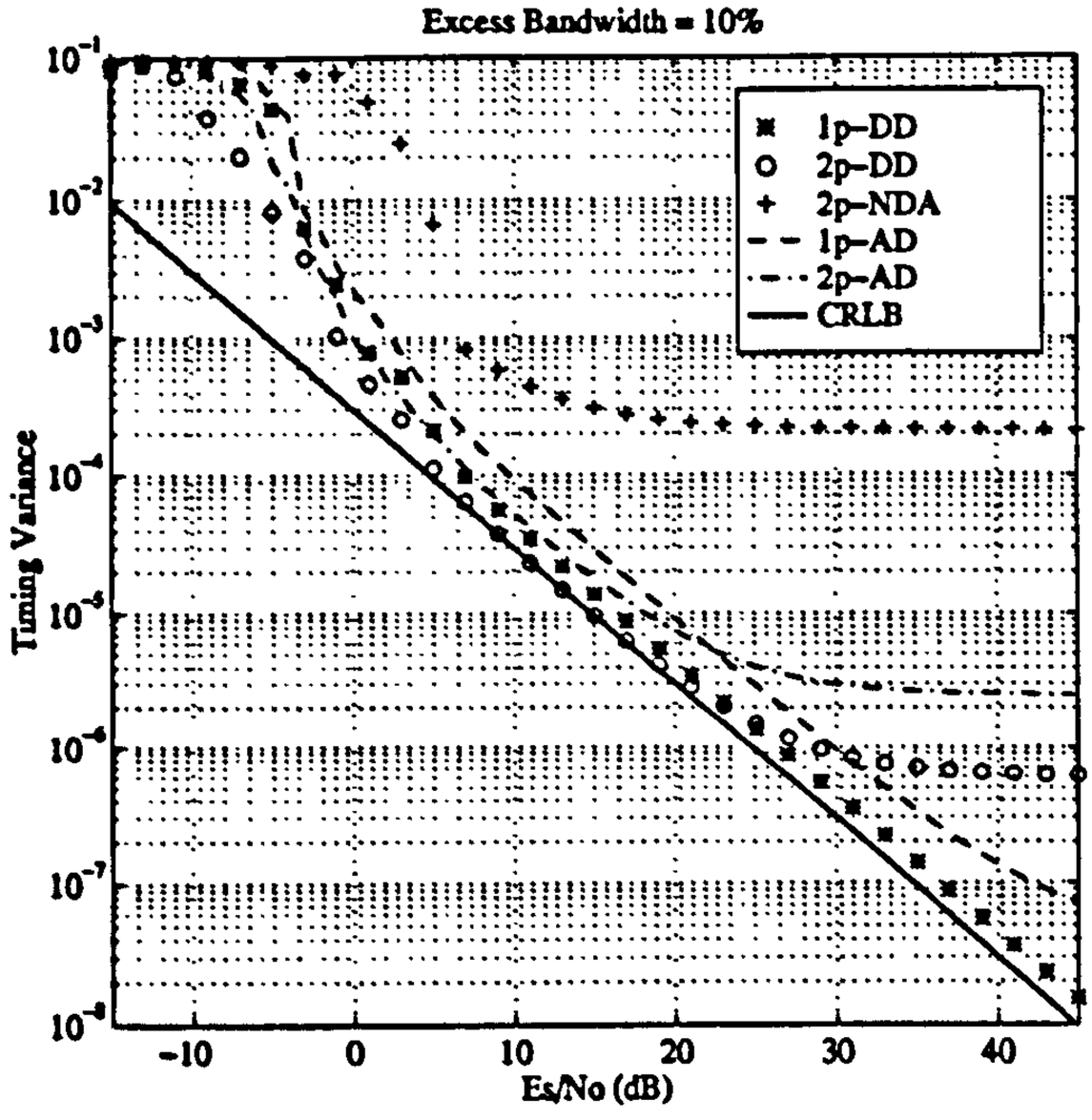



(c) Fractional Phase

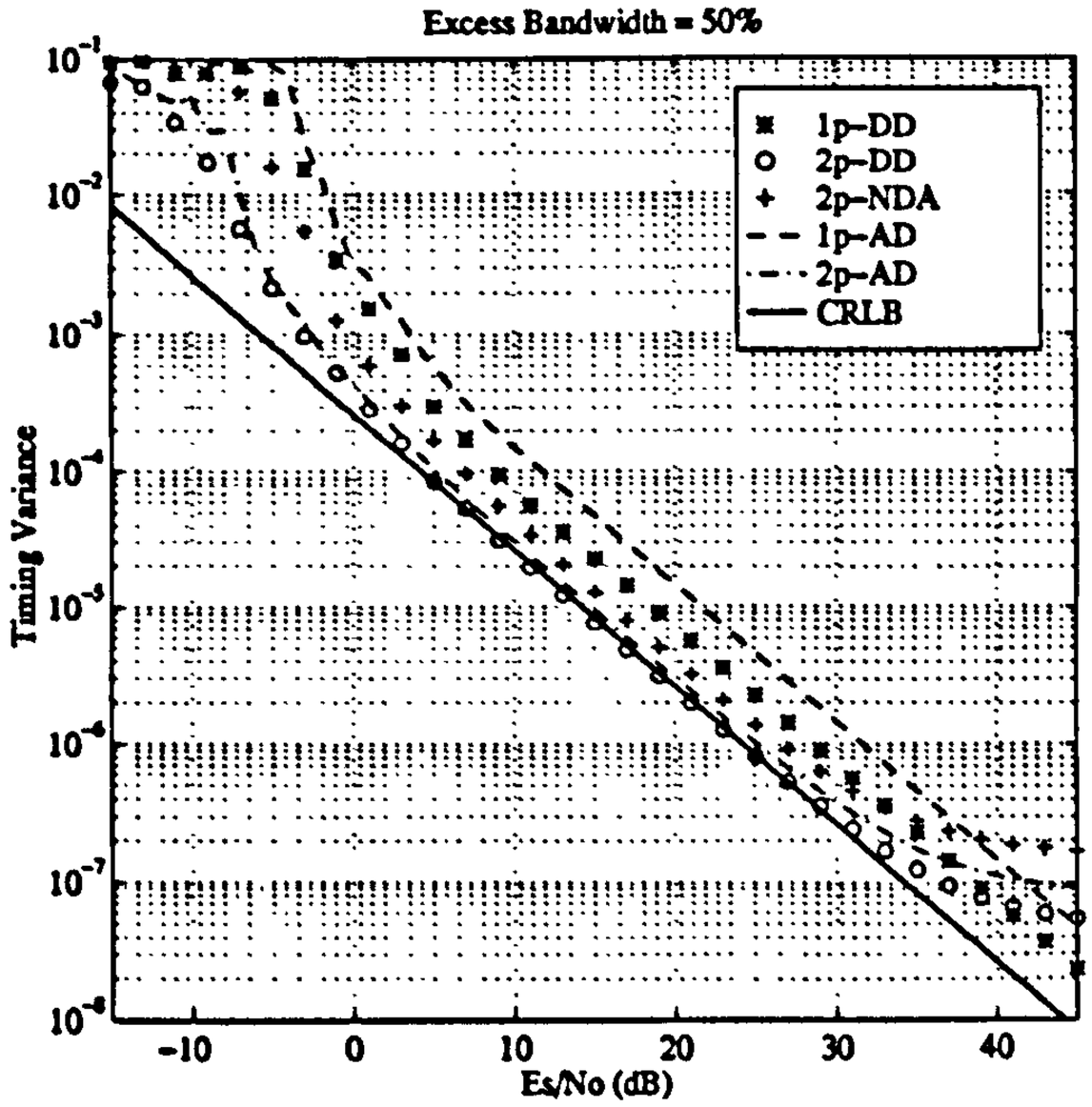



(d) Fractional Phase

Figure 6.44: Timing frequency acquisition in the 2pNDA asynchronous timing recovery loop. The excess bandwidth is $\alpha = 0.8$, the loop bandwidth $B_L = 0.001$, the damping factor is $\gamma = 1/2$. The loop responses are for input data rates of $R_b = 0.001/T$ (left) and $R_b = 0.0001/T$ (right)

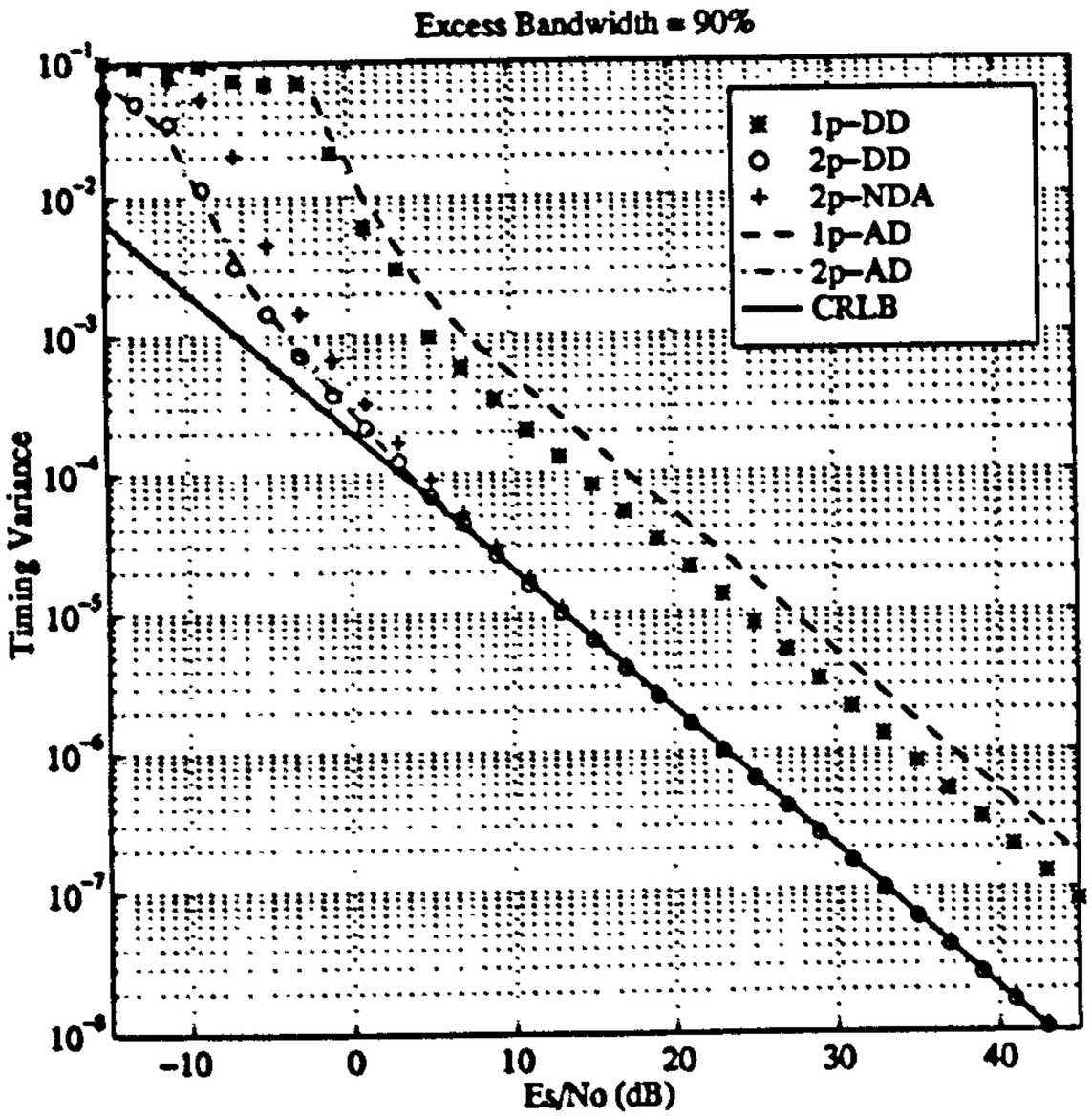



(a) $\alpha = 0.1$



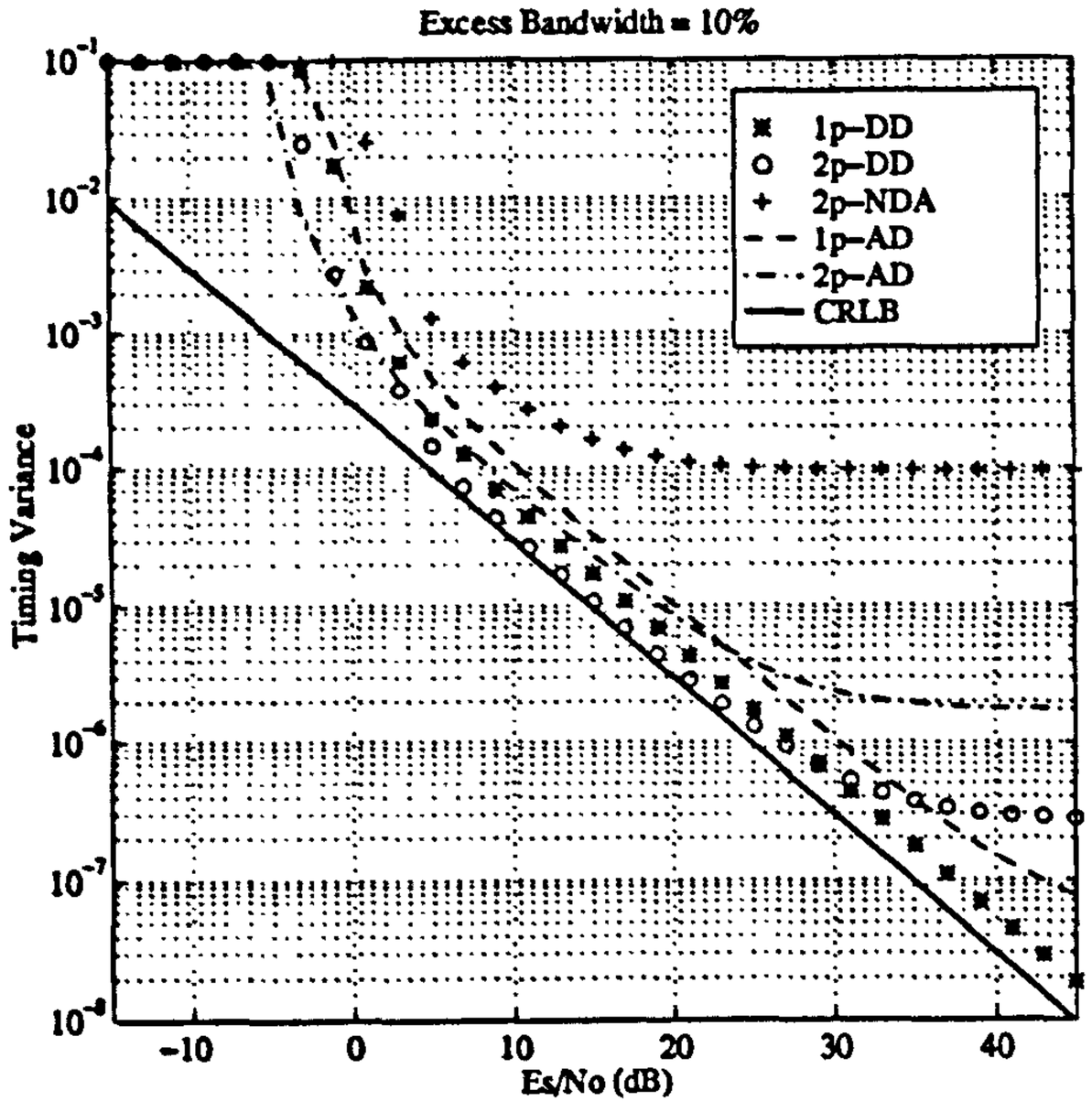

(b) $\alpha = 0.5$

Figure 6.45: Tracking Performance of Symbol Timing Recovery Schemes. Noise equivalent bandwidth is 0.001. First order loop. Modulation is QPSK.

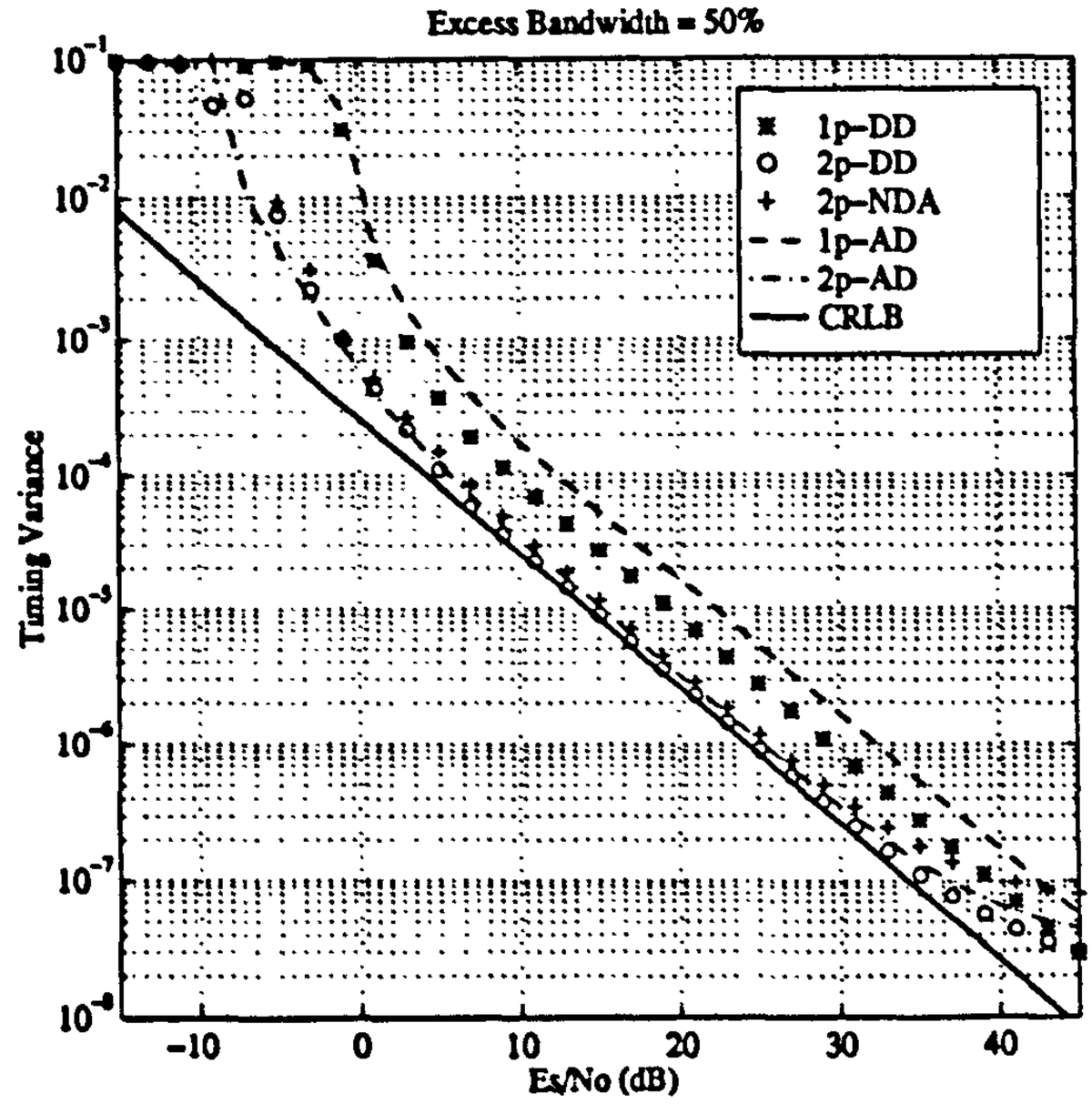



(a) $\alpha = 0.9$

Figure 6.46: Tracking Performance of Symbol Timing Recovery Schemes. Noise equivalent bandwidth is 0.001. First order loop. Modulation is QPSK.

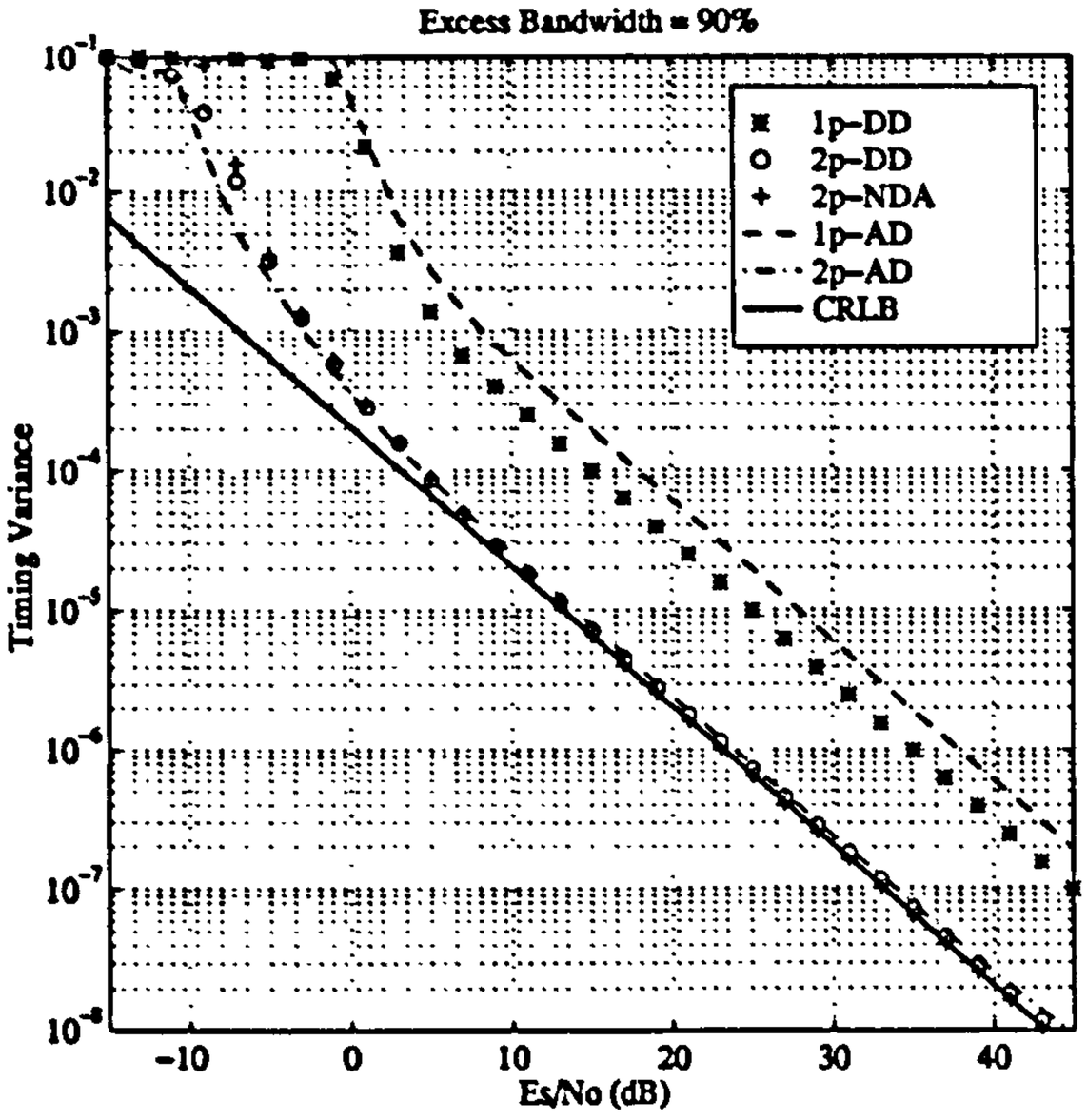



(a) $\alpha = 0.1$




(b) $\alpha = 0.5$

Figure 6.47: Tracking Performance of Symbol Timing Recovery Schemes. Noise equivalent bandwidth is 0.001. Second order loop. Modulation is QPSK.




(a) $\alpha = 0.9$

Figure 6.48: Tracking Performance of Symbol Timing Recovery Schemes. Noise equivalent bandwidth is 0.001. Second order loop. Modulation is QPSK.

Comparing the tracking performance as a function of the loop order shows that the second order loop is better at suppressing self-noise than the first order loop. This is consistent with the model suggested in [29] as

$$\sigma^2 = (2\,B_L T)\frac{A}{2\,E_s/N_o} + \kappa(2\,B_L T)^2 B \tag{6.101}$$

in which $\kappa$ is dependent on the loop order. For a first order loop $\kappa = 2$ and for a second order loop $\kappa = 1/4$ if $\gamma = 1/2$. The simulated tracking results were fitted to the model above using a simplex minimisation algorithm. This then returns values of $A, B$ for each of the timing error detectors and for each value of excess bandwidth. These results are contained in the following tables. Also shown is the deviation in dBs of the given scheme from the modified Cramer-Rao bound. Having determined the model coefficients it is tempting to say that one could predict the timing variance at other values of loop bandwidth and signal to noise and other authors have suggested this [29], but because of the highly nonlinear nature of the timing error detectors one suspects that the small signal model only holds true for small values of loop noise bandwidth. This still needs to be quantified but the basic problem is illustrated in figure 6.49 which shows the timing error detector variance as a function of loop noise bandwidth at a fixed value of $E_s/N_o = 15$dB. Note that it is only at low values of loop bandwidth $B_L$ that the simulation results meet the CRLB. As the loop bandwidth is increased the simulated results and the theoretical ideal diverge. What is also of concern is that some of the symbol timing recovery loops appear to become unstable at larger values of $B_L$. One can see that this is particularly true of the 2pNDA scheme which for $\alpha = 0.5$ and $E_s/N_o = 15$dB appears to become unstable at $B_L = 0.05$. A comprehensive study of the stability of synchronisation loops still needs to be undertaken so that we can determine the upper bounds on $B_L$ for a given set of system parameters.

First Order Loop, $\alpha = 0.1$

| • | A | B | Deviation (dB) |
|---|---|---|---|
| 1pDD | 4.379792e-01 | 1.742096e-04 | 1.610930 |
| 2pDD | 2.874241e-01 | 7.683007e-02 | -0.218373 |
| 2pNDA | 3.057760e+00 | 2.685251e+01 | 10.050430 |
| 1pAD | 9.191184e-01 | 5.278636e-03 | 4.830110 |
| 2pAD | 4.952968e-01 | 2.699101e-01 | 2.145050 |

First Order Loop, $\alpha = 0.5$

| • | A | B | Deviation (dB) |
|---|---|---|---|
| 1pDD | 7.232687e-01 | 8.560388e-05 | 4.341720 |
| 2pDD | 2.473378e-01 | 5.915212e-03 | -0.318373 |
| 2pNDA | 3.878496e-01 | 1.966114e-02 | 1.635356 |
| 1pAD | 1.466985e+00 | 2.258734e-04 | 7.412980 |
| 2pAD | 3.014387e-01 | 9.725167e-03 | 0.540714 |

First Order Loop, $\alpha = 0.9$

| • | A | B | Deviation (dB) |
|---|---|---|---|
| 1pDD | 2.629720e+00 | 8.175369e-05 | 11.015350 |
| 2pDD | 2.022927e-01 | 6.673070e-05 | -0.123943 |
| 2pNDA | 2.068088e-01 | 1.139864e-04 | -0.028057 |
| 1pAD | 5.238679e+00 | 1.233352e-03 | 14.008472 |
| 2pAD | 2.071913e-01 | 8.315970e-05 | -0.020032 |

Second Order Loop, $\alpha = 0.1$

| • | A | B | Deviation (dB) |
|---|---|---|---|
| 1pDD | 5.473117e-01 | 1.010689e-03 | 2.578743 |
| 2pDD | 3.341354e-01 | 2.677848e-01 | 0.435621 |
| 2pNDA | 2.043379e+00 | 9.411516e+01 | 8.299886 |
| 1pAD | 1.040224e+00 | 4.206473e-02 | 5.367662 |
| 2pAD | 5.528005e-01 | 9.386596e-01 | 2.622080 |

Second Order Loop, $\alpha = 0.5$

| • | A | B | Deviation (dB) |
|---|---|---|---|
| 1pDD | 8.784951e-01 | 8.459773e-04 | 5.186117 |
| 2pDD | 2.880665e-01 | 2.087812e-02 | 0.343650 |
| 2pNDA | 3.519271e-01 | 6.872532e-02 | 1.213251 |
| 1pAD | 1.706958e+00 | 2.385412e-03 | 8.070951 |
| 2pAD | 3.219109e-01 | 3.414165e-02 | 0.826081 |

Second Order Loop, $\alpha = 0.9$

| • | A | B | Deviation (dB) |
|---|---|---|---|
| 1pDD | 3.131263e+00 | 1.781306e-04 | 11.773449 |
| 2pDD | 2.300565e-01 | 2.330659e-04 | 0.434599 |
| 2pNDA | 2.034151e-01 | 3.967856e-04 | -0.099914 |
| 1pAD | 6.079612e+00 | 5.797920e-03 | 14.655013 |
| 2pAD | 2.162328e-01 | 3.126147e-04 | 0.165469 |

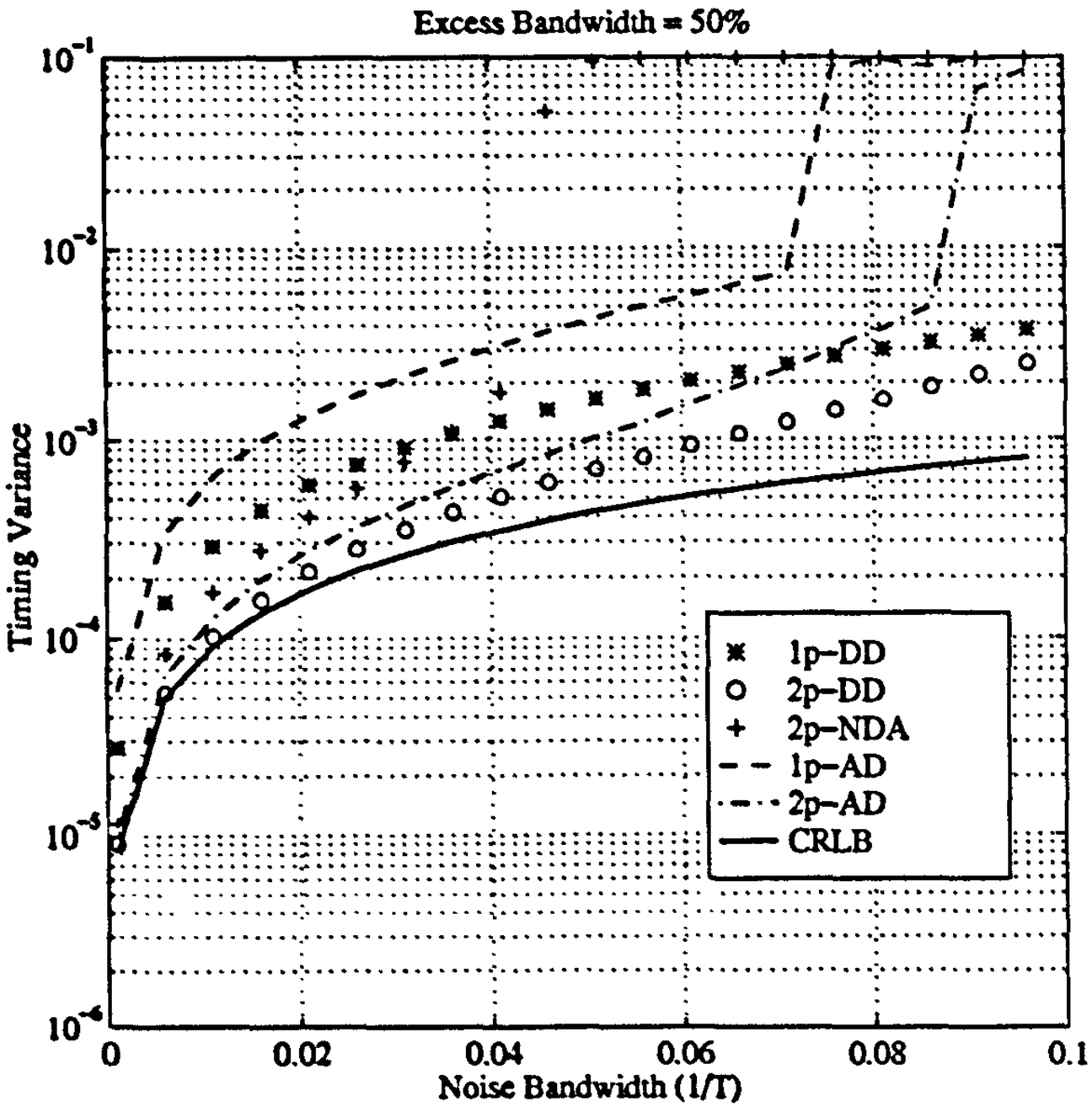



Figure 6.49: Tracking Performance of Symbol Timing Recovery Schemes as a Function of Loop Noise Bandwidth. $E_s/N_o = 15.0$dB. Second order loop. Modulation is QPSK with $\alpha = 0.5$.

# Chapter 7

# Carrier Synchronisation

## 7.1 Introduction

A brief overview of carrier phase synchronisation in sampled receivers will now be given in this chapter. We will concentrate firstly on the complex-baseband equivalent model of a Phase-Locked Loop and then illustrate how this can used for carrier phase synchronisation for double-sideband suppressed-carrier modulation schemes. We shall look also at some schemes which are peculiar to digital implementation such as the feedforward Viterbi & Viterbi algorithm.

We also look at schemes for detection of offset modulation, namely (G)MSK. These consist of the standard orthogonal-coherent and differential-phase detectors. The problem with using differential-phase detection with tightly pre-filtered-CPM is that the eye-opening is closed due to the inherent inter-symbol interference in the scheme. This can be countered to some extent by using a fixed Decision Feedback Equaliser (DFE). The implementation of one and two bit differential phase detection of GMSK with and without Decision Feedback Equalisation is demonstrated.

## 7.2 Phase Locked Loops

Many synchronisation systems utilise the Phase-Locked Loop (PLL). A general PLL can be represented by the diagram in figure 7.1.

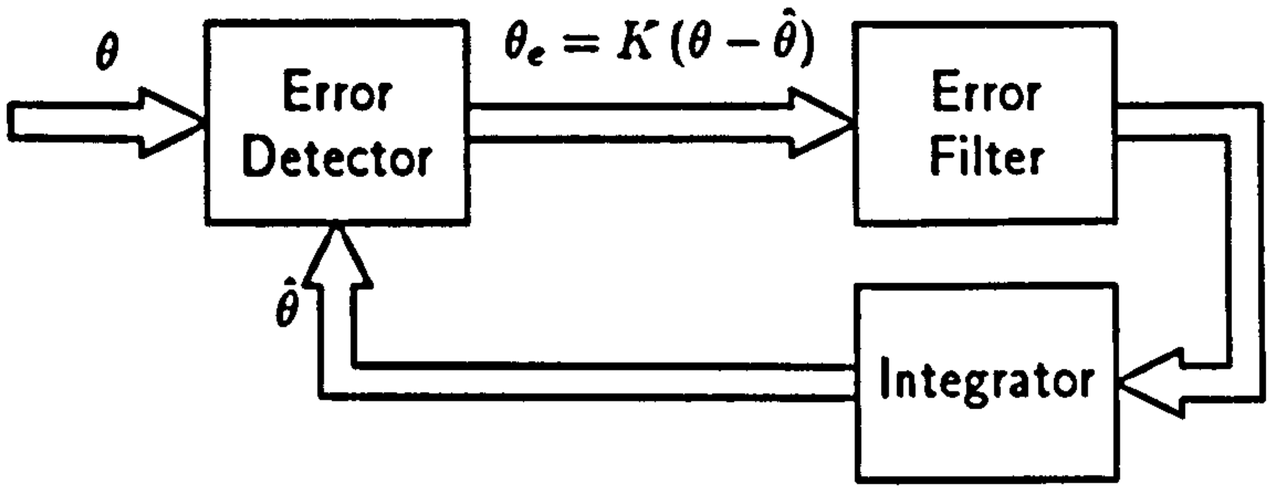



Figure 7.1: Generic Structure of Phase Locked Loop

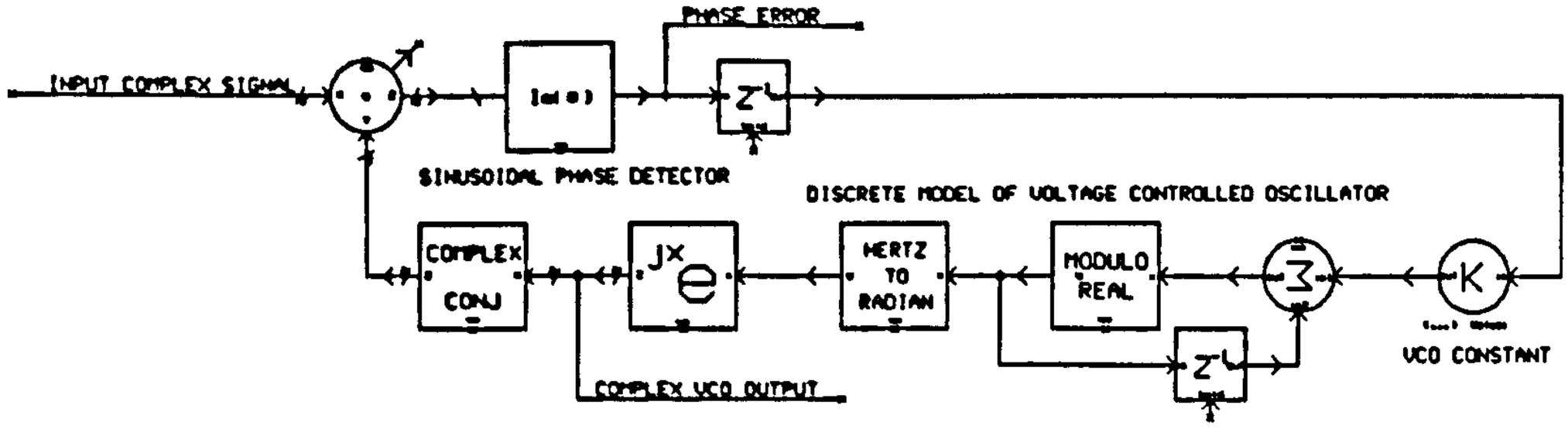



Figure 7.2: SPW Implementation of a First Order Phase-Locked Loop.

The representation of the PLL as shown in figure 7.1 looks deceptively simple. The PLL is in fact quite difficult to analyse because of its non-linear nature. The non-linearity arises because of the nature of the phase detector in that phase differences between the input signal and the VCO output can only be measured modulo-$2\pi$. Note that in the complex-equivalent baseband representation the phase detector is implemented by taking the imaginary part of the product of two complex phasors. This is represented by the following equation:

$$\sin\left(\theta(t) - \hat{\theta}(t)\right) = \mathrm{Im}\left\{\exp\left(j\,\theta(t)\right) \times \exp\left(-j\,\hat{\theta}(t)\right)\right\} \tag{7.1}$$

In the following sections we shall show how the various PLLs can be implemented in the complex-baseband and shall attempt to reconcile their analytical performance with that obtained from simulation.

### 7.2.1 First Order Phase Locked Loops

The order of a PLL is determined by the characteristics of the loop filter, $h(s)$. A first order loop has no loop filter. A second order loop has a loop filter with one pole in its transfer function. In general an Nth order PLL is characterised by having a loop filter with N-1 poles. Here we shall only be concerned with first and second order PLLs.

The general transfer function for a PLL in terms of its loop filter is given as:

$$H(s) = \frac{\hat{\theta}(s)}{\theta(s)} = \frac{K\,h(s)}{s + K\,h(s)} \tag{7.2}$$

where $K$ is the total loop gain. The detail of a first order PLL implemented using SPW is as shown in figure 7.2.

The transfer function for the first order loop is found by putting $h(s) = C$ in equation 7.2 to give:

$$H(s) = \frac{KC}{s + KC} \tag{7.3}$$

This represents a simple lowpass filter with a pole at $s = -KC$. By using such a PLL to *tune* to a carrier embedded in noise, the signal-to-noise ratio can be improved dramatically. However, the first order PLL is not used very much in practice because the hold-in range

of a first order PLL and its loop bandwidth are both dependent on the total loop gain. That is, if we wish to use the PLL to hold-in frequencies over a large bandwidth then the loop-bandwidth has to be large also and the ability of the PLL to reject noise-components is reduced. Expressed mathematically, the hold-in range of a general PLL can be described in terms of its loop filter and loop gain as:

$$|\Delta f| \leq K \, |h(0)| \tag{7.4}$$

This hold-in range is a direct consequence of the limits of the phase detector to track phase changes greater than $\pm\pi$. The second order PLL does not have the same inter-dependence of loop-bandwidth and hold-in range as shall be illustrated next.

### 7.2.2 Second Order Phase Locked Loops

In the following we present a linearised analysis of the PLL, whereas in the simulations we are obviously dealing with sampled PLLs. There will be little difference in performance between continuous and discrete PLLs provided the loop bandwidth is small.

#### Type I Second Order Phase Lock Loop

The type I second order PLL as implemented in the complex baseband has a lead-lag filter as the PLL loop filter. The loop filter is referred to as an imperfect integrator and has the frequency response given by:

$$h(s) = \frac{\tau_2 s + 1}{\tau_1 s + 1} \tag{7.5}$$

In order to simulate the above filter as the loop filter for the PLL we need to transform the filter response from the S-domain to the Z-domain. To do this we use the *Bilinear Transformation* as given by the following expression:

$$s = 2f_s \left( \frac{z-1}{z+1} \right) \tag{7.6}$$

There are other transformations which are essentially discrete-time approximations to the continuous-time integral $1/s$ but the most popular for the simulation of communication systems is the backward-difference and the bilinear-transform [32]. These are illustrated in figure 7.3.

By using the bilinear transformation we arrive at a discrete-time implementation of the lead-lag filter as follows:

$$h(z) = \frac{a_0 + a_1 z^{-1}}{1 + b_0 z^{-1}} \tag{7.7}$$

with the filter coefficients given by the following:

$$a_0 = \left( \frac{1 + 2\tau_2 f_s}{1 + 2\tau_1 f_s} \right) \, , a_1 = \left( \frac{1 - 2\tau_2 f_s}{1 + 2\tau_1 f_s} \right) \text{ and } b_0 = \left( \frac{1 - 2\tau_1 f_s}{1 + 2\tau_1 f_s} \right) \tag{7.8}$$

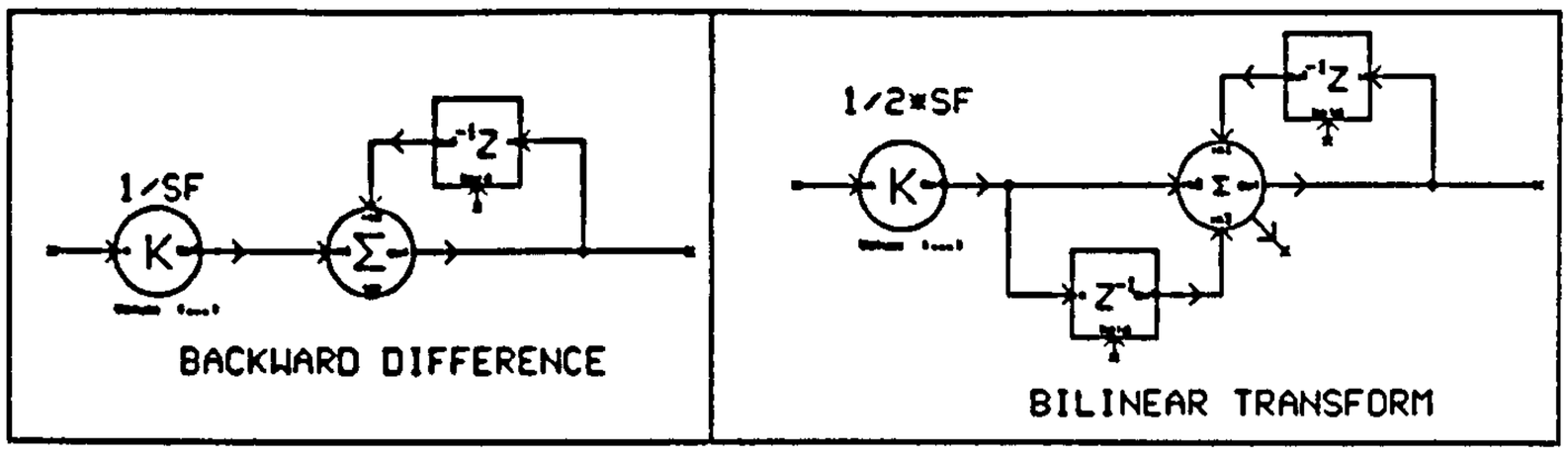



Figure 7.3: Different discrete-time approximations to the ideal integrator

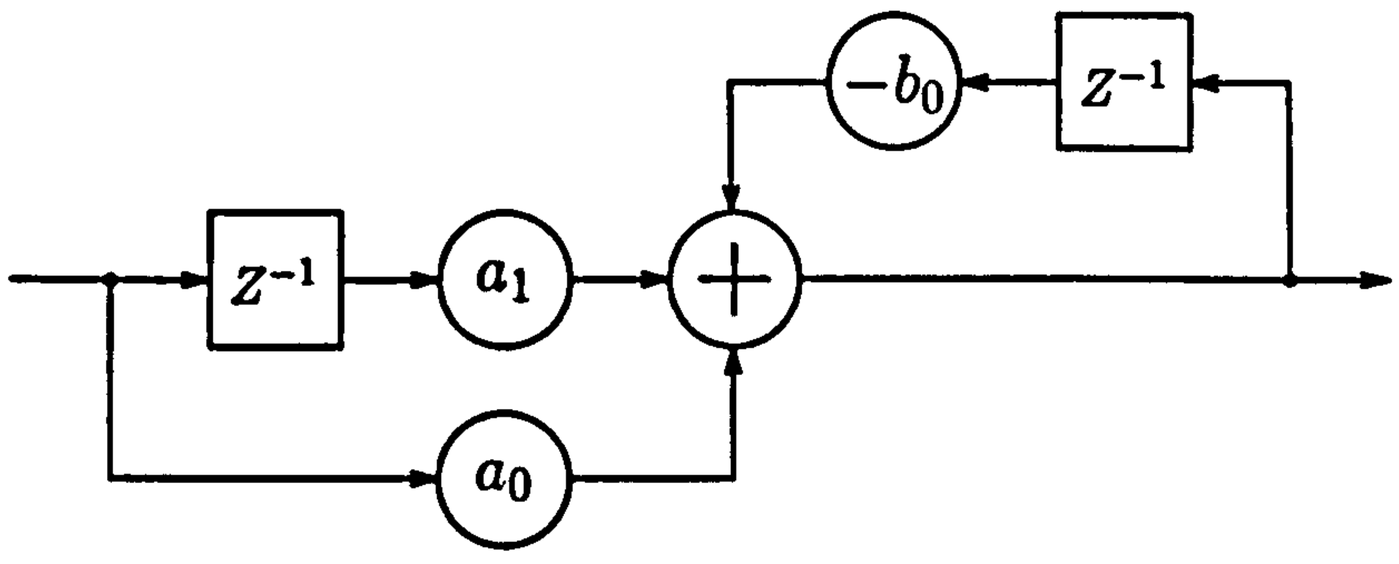



Figure 7.4: Discrete Implementation of Lead-Lag Filter (Imperfect Integrator).

In PLL theory the filter in figure 7.4 is referred to as an *imperfect integrator.* It is realised in practice by a passive low-pass filter structure. Such a filter leads to a non-zero phase error for a steady state frequency offset on the input (as does indeed the first order PLL). It does have a better bandwidth performance over the first order loop however. The bandwidth for the type I second order PLL is given by putting the expression for $h(s)$ into the closed-loop transfer function to give:

$$\begin{aligned} H(s) &= \frac{K\,h(s)}{s + K\,h(s)} \\ &= \frac{K\,(\tau_2 s + 1)}{s^2\tau_1 + s\,(1 + K\,\tau_2) + K} \end{aligned} \tag{7.9}$$

Introducing two new variables the natural frequency, $\omega_n$, and the damping factor, $\gamma_n$, the above equation is then written as:

$$H(s) = \frac{\left[2\,\gamma_n\omega_n - \left(\frac{\omega_n^2}{K}\right)\right] s + \omega_n^2}{s^2 + 2\,\gamma_n\omega_n s + \omega_n^2} \tag{7.10}$$

where:

$$\tau_1 = \frac{K}{\omega_n^2} \text{ and } \tau_2 = \frac{2\gamma_n}{\omega_n}\left(1 - \frac{\omega_n}{K}\frac{1}{2\gamma_n}\right) \tag{7.11}$$

Also the hold-in range for this loop is the same as the first order loop that is $h(0) = 1$.

### Type II Second Order Phase Lock Loop

A better loop performance is achieved for the second-order type II PLL by using an active filter for the loop filter. This then has the following frequency response:

$$h(s) = \frac{\tau_2 s + 1}{\tau_1 s} \tag{7.12}$$

This filter is referred to as a *perfect-integrator.* Note because of the pole at $s = 0$ this filter now has infinite gain at zero frequency. However, the loop filter when used in a PLL does not have to be stable in order for the complete loop transfer function to be stable. This filter can also be configured according to the basic structure shown in figure 7.4 by again using the bilinear transformation from the analogue domain to the discrete domain. The filter coefficients in this case are given as follows:

$$a_0 = \left(\frac{1}{2 f_s \tau_1} + \frac{\tau_2}{\tau_1}\right) \text{ , } a_1 = \left(\frac{1}{2 f_s \tau_1} - \frac{\tau_2}{\tau_1}\right) \text{ and } b_0 = -1 \tag{7.13}$$

This loop filter leads to a structure which is called a type II second-order loop PLL. As with the type I loop, the overall loop bandwidth can be found as follows:

$$H(s) = \frac{K(\tau_2 s + 1)}{s^2 \tau_1 + s K \tau_2 + K} \tag{7.14}$$

Again, we can put this in terms of the natural frequency, $\omega_n$, and the damping factor, $\gamma_n$ to give:

$$H(s) = \frac{2\gamma_n \omega_n s + \omega_n^2}{s^2 + 2\gamma_n \omega_n s + \omega_n^2} \tag{7.15}$$

where:

$$\tau_1 = \frac{K}{\omega_n^2} \text{ and } \tau_2 = \frac{2\gamma_n}{\omega_n} \tag{7.16}$$

Note that because of the infinite gain in the loop filter at DC the hold-in range for the type II second order PLL is theoretically infinite (this will not be true in reality of course because of the saturation of components).

Note that the frequency responses of the type I and type II second order loops becomes equal if the gain is large enough. Generally this is the case. Figure 7.5 shows the frequency response of a high-gain second-order PLL. The frequency axis is normalised to the natural frequency of the loop and the damping factor is shown as a parameter.

## 7.3 Phase Locked Loop Tracking Performance

To substantiate the complex-baseband model for the type II second order PLL we examine its tracking performance. This is understood to mean the ability of the loop to generate a VCO output which changes as the input signal. What is an important criterion is the phase error between the input signal and the VCO output. We examine three different input signals to the second order PLL and look at the transient and steady state outputs from the loop.

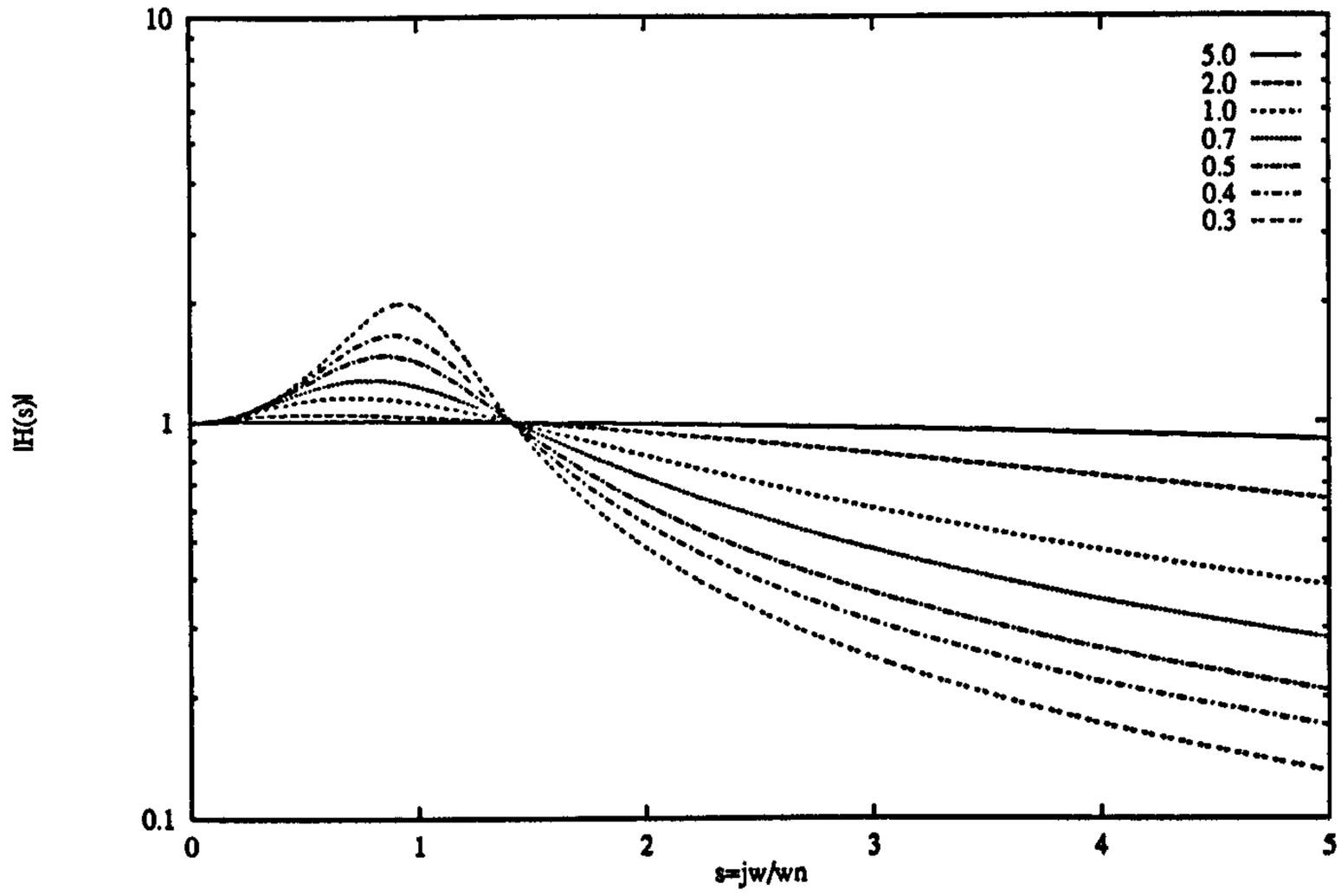



Figure 7.5: Frequency Response of a High-Gain Second-Order PLL with $\gamma_n$ a parameter.

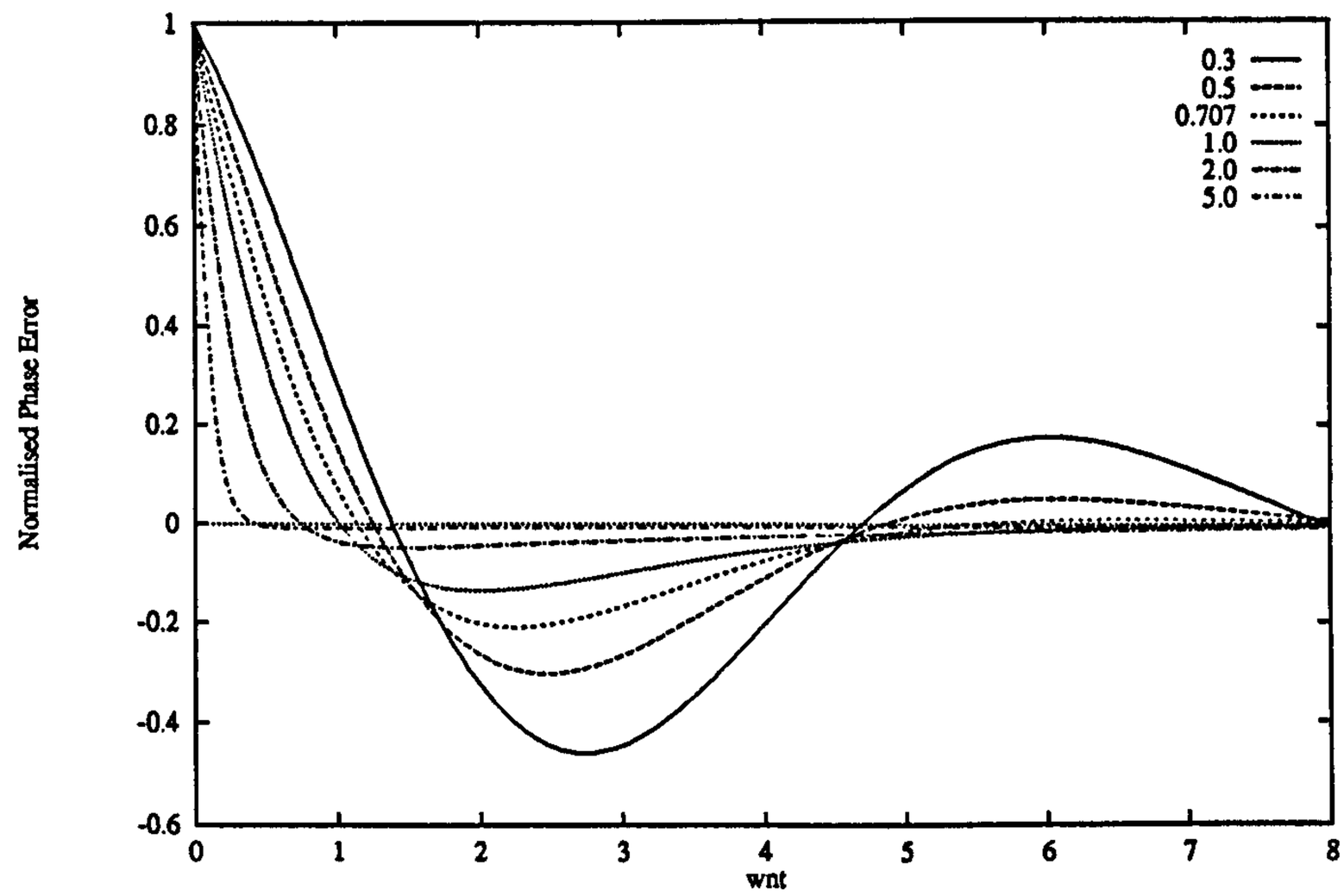



Figure 7.6: Phase error due to a step in phase $\Delta\theta$ with $\gamma_n$ a parameter.

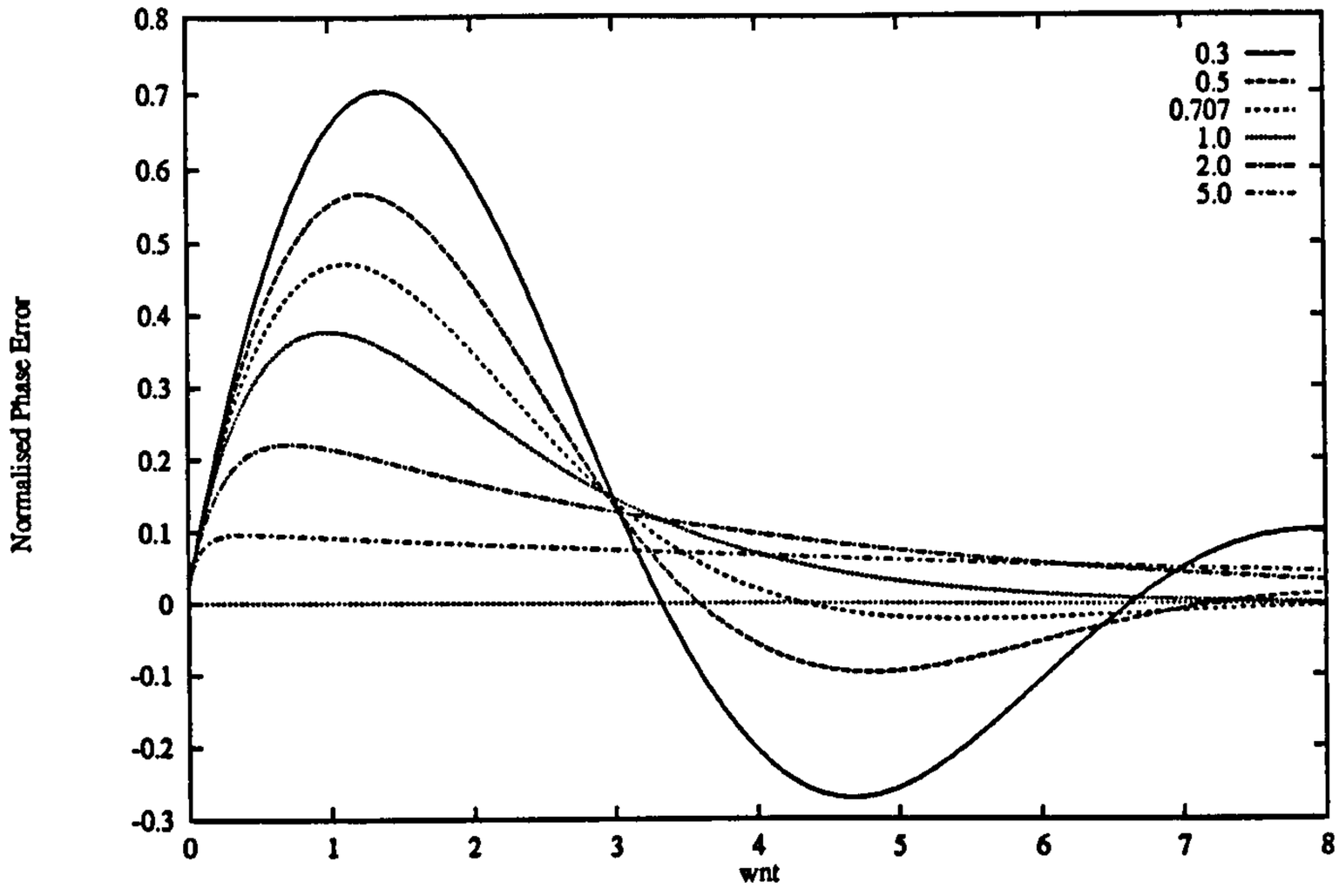



Figure 7.7: Phase error due to a step in frequency $\Delta\omega$ with $\gamma_n$ a parameter.

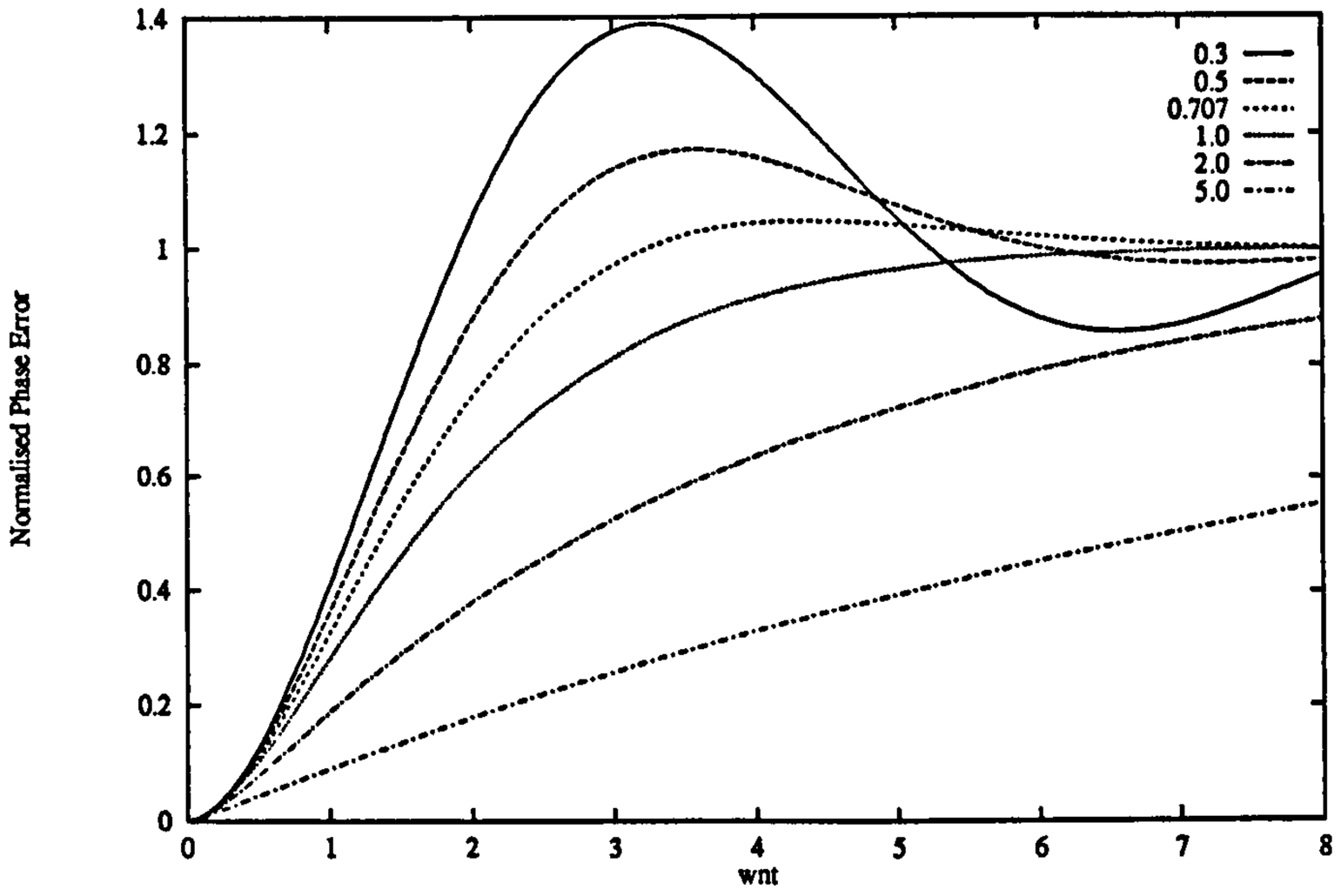



Figure 7.8: Phase error due to a ramp in frequency $\Delta\dot{\omega}$ with $\gamma_n$ a parameter.

The three signals that we shall look at are: a step in phase $\Delta\theta$; a step in frequency $\Delta\omega$; and a frequency ramp $\Delta\dot{\omega}$. The simulated phase error to each of these input signals is as shown in figures 7.6 to 7.8.

The theory behind the curves shown in the figure is covered in [70] for the ideal analogue PLL. The complex-baseband model of the type II second order PLL agrees well with the theoretical predictions provided the oversampling in the simulation is sufficiently large. Discrete-time equations describing the tracking performance of Digital PLLs can be found in [71] and these may need to be used if the oversampling is not large enough. The *integration method* [32] that one selects when simulating the VCO will also effect the results for small values of sampling frequency.

Some salient features of the curves in figures 7.6 to 7.8 are worth studying. Firstly, a second order PLL will always track a phase step on the input with zero steady state phase error (as will a first order loop). The steady state phase error that results when a frequency step is input to the loop is given as follows:

$$\theta_e = \frac{\Delta f}{K\, h(0)} \tag{7.17}$$

So, for a type II loop with $h(0) = +\infty$ the phase error will be zero. For a type I loop the phase error will decrease as the loop gain is increased. Finally, the steady state response of the second order PLL to a frequency ramp is to give a constant phase error as is shown in figure 7.8. Should we wish to track a frequency ramp with zero phase error then we would need to use a third order PLL.

### 7.3.1 Phase Plane Representation

The performance of the PLL can also be represented in the *phase plane*. This representation is formed by plotting the frequency error against the phase error of the PLL for different input disturbances. Figure 7.9 shows the performance of a high gain second order PLL when a frequency step is present on the input. In the figure the oversampling rate was set at 128 and the input frequency was at 8 Hz, so this is really a discrete model of an analogue PLL. The natural frequency of the loop was set at $\omega_n = 1$ and the damping factor $\gamma_n = 0.5$. One can see from such a representation that the loop will eventually lock to the input signal with zero phase and frequency error. Using the Phase Plane method is useful in characterising nonlinear control systems in general.

## 7.4 Carrier Recovery

In this section we shall be concerned with carrier recovery for Double-Sideband Suppressed Carrier modulations (DSBSC). This covers all the schemes BPSK, QPSK, QAM etc. In these schemes a frequency component is not present at the carrier frequency (hence, suppressed carrier). We shall examine two methods to regenerate the carrier using PLLs in the following.

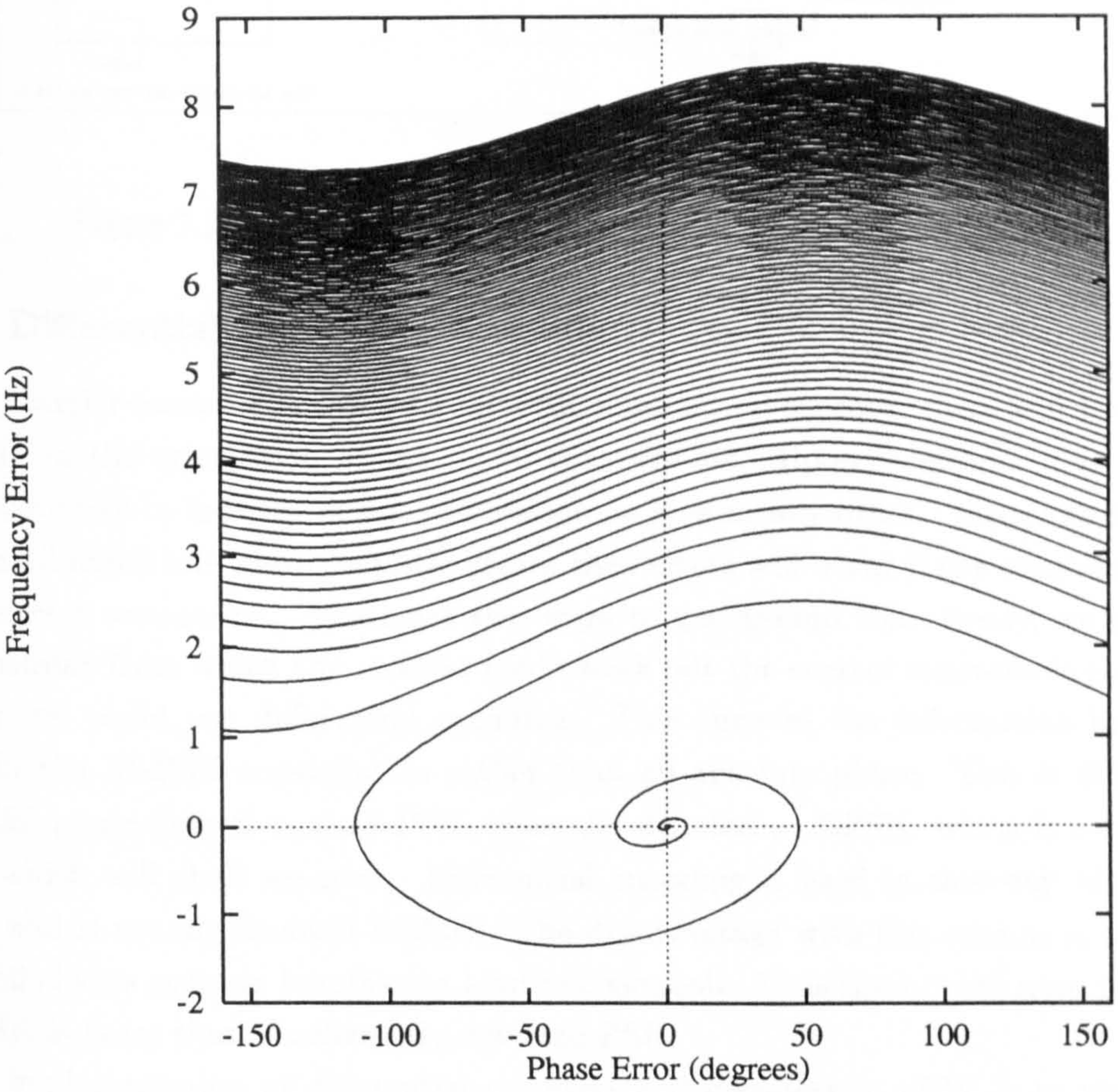



Figure 7.9: Phase plane plot with high gain type II second order PLL with a frequency step input disturbance.

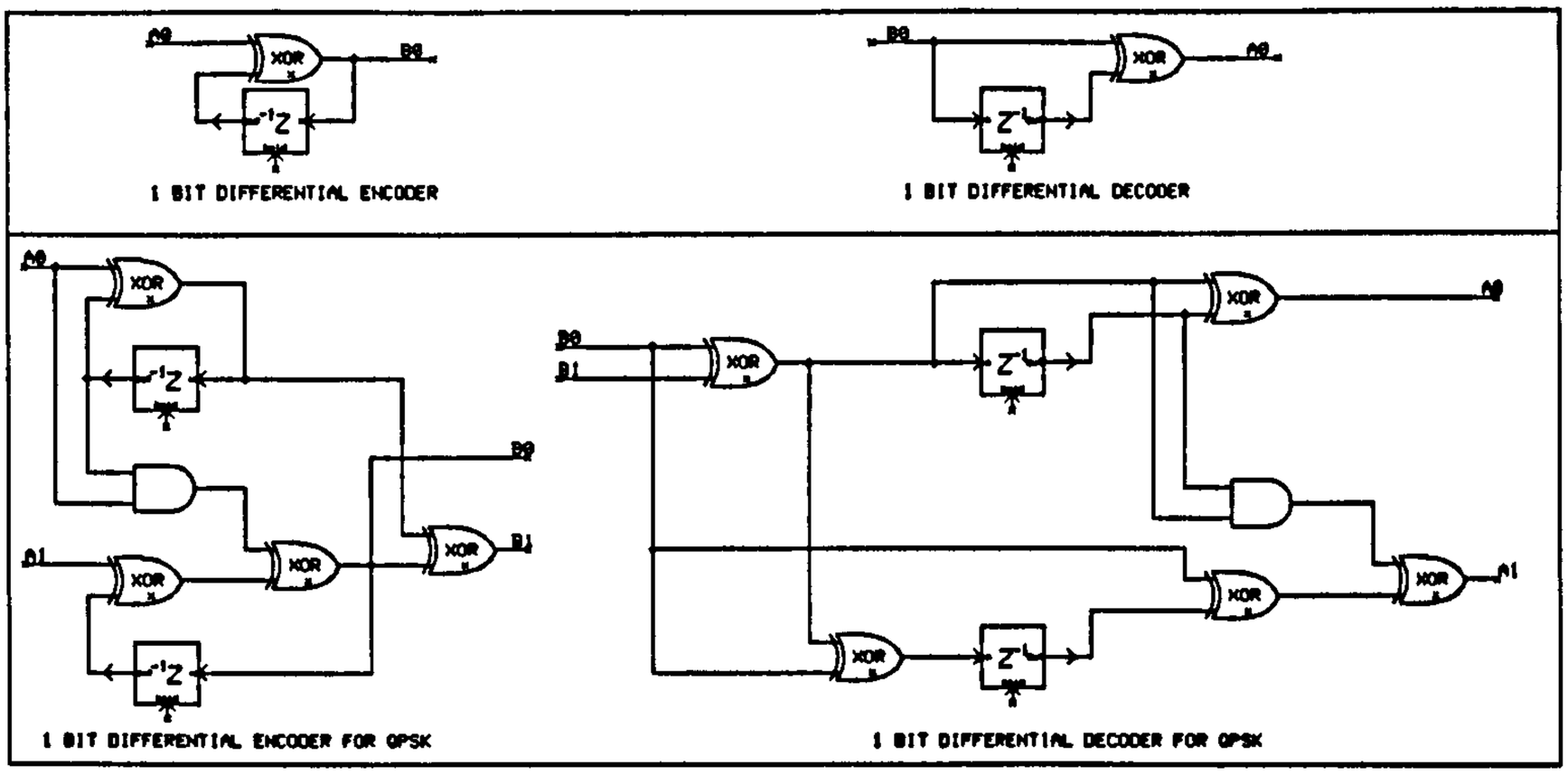



Figure 7.10: SPW implementation of differential encoders & decoders

### 7.4.1 Differential Encoding/Decoding

In many carrier-recovery systems for $M$-PSK modulation schemes there will be an $M$-fold ambiguity in the orientation of the signal constellation. Upon reception, it is as easy for the synchroniser to lock up in one orientation as it is in any other. Therefore if the input data is modulated into absolute phase states then errors will occur if the constellation is not in the correct orientation. There are two ways to get around this: firstly, we could use a data preamble from which the receiver could work out the correct constellation orientation; secondly, we could use differential encoding. This encodes the information into a phase change in the $M$-PSK constellation rather than an absolute phase. This is different from differential phase detection of $M$-PSK, generally denoted as DPSK, which is a noncoherent scheme which will shall see next. Differential encoding is used in this way with coherent systems and is usually denoted DEPSK. The disadvantage with this scheme is that symbol errors will always occur in bursts of at least two symbols. Thus the symbol error performance of DEPSK is twice that of coherently detected PSK.

The implementation of differential encoders and decoders in SPW is as shown in figure 7.10. The top pair are simply 1 bit differential encoders whereas the lower pair are for Gray-encoded QPSK each accepting an in-phase and quadrature modulating data stream.

### 7.4.2 M-Law Device

Consider first a BPSK signal. One method to regenerate the carrier is to square the signal and hence generate frequency components at $2\,f_c$. This form of carrier recovery is referred to as *waveform-regeneration* [62]. The effect of squaring the BPSK waveform is to remove the modulation. This can be generalised to M-ary PSK as follows. The M-ary PSK signal can be represented as:

$$s(t) = A\,e^{j\left(2\pi f_c t + \phi + \frac{2\pi}{M}(m-1)\right)} \text{ for } m = 1, 2 \ldots M \tag{7.18}$$

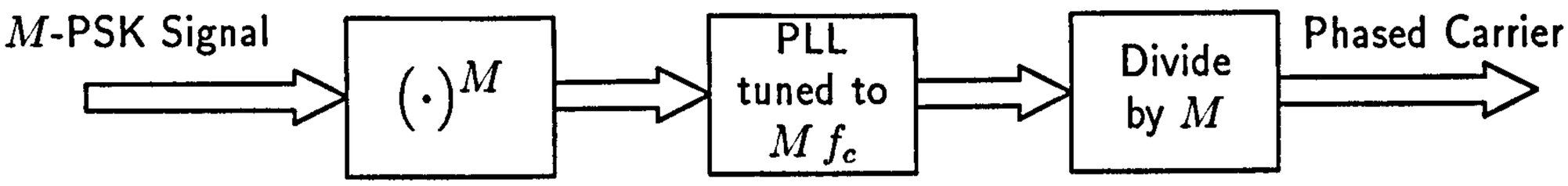



Figure 7.11: Carrier recovery using M-law device.

We see that raising this signal to the power M destroys the modulation as:

$$\begin{aligned} \{s(t)\}^M &= A^M e^{j(2\pi M f_c t + M\phi + 2\pi(m-1))} \text{ for } m = 1, 2 \ldots M \\ &= A^M e^{j(2\pi M f_c t + M\phi)} \end{aligned} \tag{7.19}$$

So the PLL tracks the signal out of the M-law device at $M\,f_c$. We need to follow the PLL circuit with a frequency divide-by-M to regenerate a phased-carrier. The general structure of such a carrier recovery loop is shown in figure 7.11.

An illustration of how the modulation is removed from an M-ary PSK signal by raising it to the power M is illustrated for a QPSK waveform in figure 7.12. In the figure the spectrum of the input, the input squared and the input to the power four is shown. Note that the complex frequency on the input QPSK signal was set to 1 so that after the M-law device a tone was generated at the complex frequency 4.

The frequency divide by M is also configured using complex-baseband PLLs.

### 7.4.3 Costas Loop

The complex-baseband model of the Costas loop is similar to the basic PLL model but both the real and imaginary components from the complex multipliers are used representing the in-phase and quadrature components of a DSBSC signal. It should be noted that if the loop filters used are identical with the loop filters used for the M-law loop then the performance of these two methods should be the same.

In the Costas loop model the input signal is first down-converted in-phase and quadrature components and then the modulation is destroyed, in the case of BPSK, by multiplication of the in-phase and quadrature components. Such a configuration is referred to as a *tracker* [62].

### 7.4.4 Note on implementation of synchroniser

Gardner in his report [62] refers to the M-law device as discussed above as a waveform regenerator and illustrates why it is not suitable (in comparison to waveform trackers) for digital implementation. A waveform regenerator is a nonlinear device that operates on an incoming signal to generate signals at a suitable reference frequency, in the case of the M-law device at a frequency of $M \times f_c$. In order to retain the integrity of the information in the regenerated carrier it is necessary to sample at a high rate. This would place too large a burden in terms of computing power on the receiver system. Adequate synchronisation

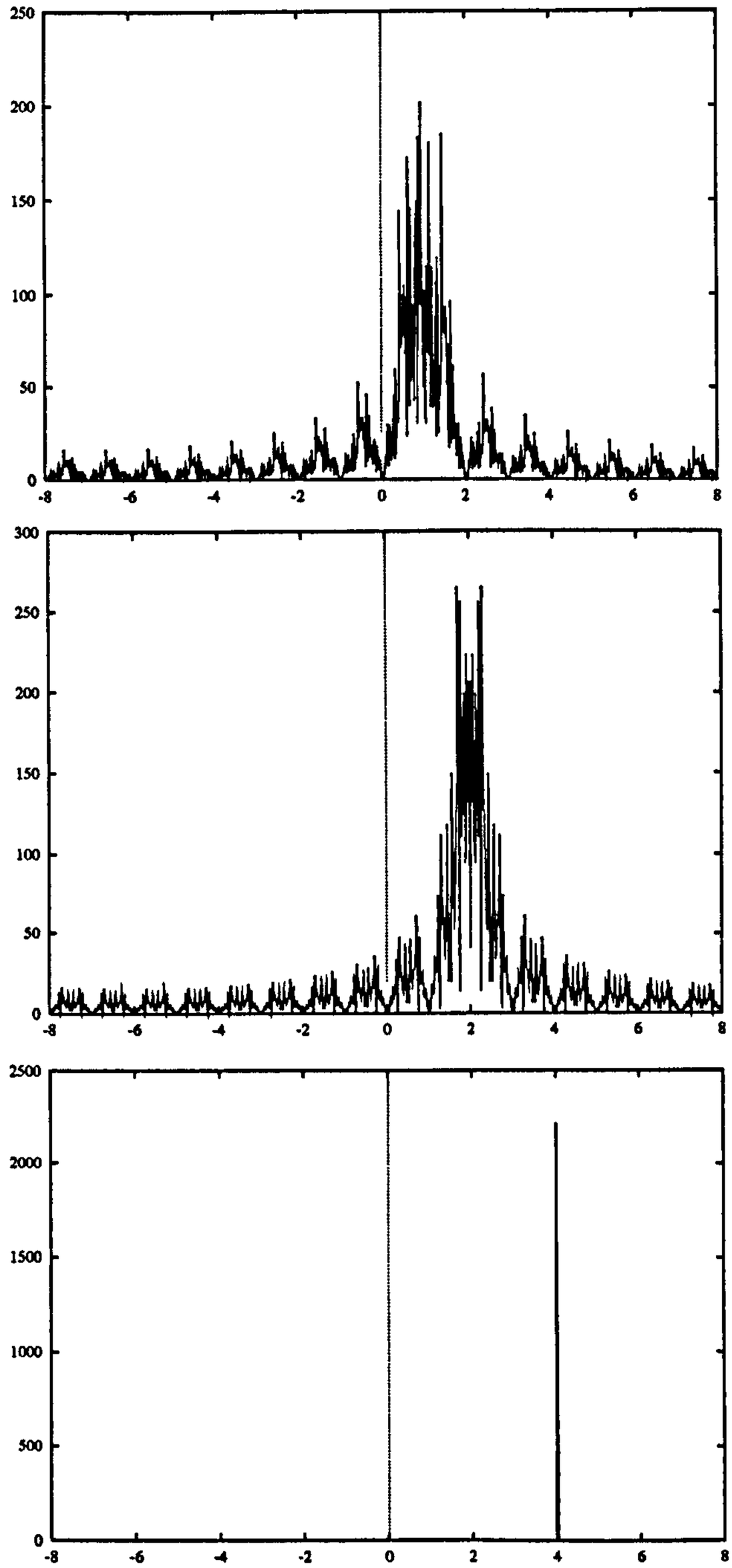



Figure 7.12: QPSK spectrum (top) after squaring (middle) and after raising to the power four (bottom).

algorithms are available that require only one or two samples per symbol and so it is wasteful to use waveform regeneration in an all-digital receiver. So it is of more use from a practical implementation point of view to consider tracking architectures, such as the Costas Loop, over waveform regeneration.

## 7.5 Feedforward Carrier Phase Recovery – The V & V Algorithm

The Viterbi family presented a method for feedforward carrier phase recovery in [61]. As is typical in the study of synchronisation systems, the method is easy to visualise but extremely difficult to analyse. In fact the analysis in [61], although long and arduous, is only partially complete as we shall see. The algorithm works as follows; the input complex baseband $M$-PSK signal (with timing assumed so we are processing at the baud rate) is converted into polar co-ordinates as,

$$z_n = x_n + i\,y_n = \rho_n \exp(i\,\phi_n) \tag{7.20}$$

Then, operations are performed on the magnitude and phase components independently. If we consider the received phase component as,

$$\phi_n = \alpha_n \left(\frac{2\pi}{M}\right) + \psi \tag{7.21}$$

where the $\alpha_n = 0 \ldots M-1$ correspond to the $M$-PSK modulation and the $\psi$ is some arbitrary static phase offset to be estimated. Multiplying $\phi_n$ by $M$ in the complex exponential then gives,

$$\exp(i\,M\phi_n) = \exp(i\,(\alpha_n\,2\pi + M\psi_n)) = \exp(i\,M\psi) \tag{7.22}$$

Therefore the phase could be estimated as,

$$\psi = \frac{1}{M} \arg(\exp(i\,M\phi_n)) \tag{7.23}$$

This is, in fact, exactly the same operation undertaken in any $M$-law device used for carrier recovery for $M$-PSK double sideband suppressed carrier signals. The V & V algorithm is slightly different in that one has an extra degree of freedom as to how the signal amplitude is used. It is not clear how one could separate out amplitude and phase in this way in an analogue system. Having multiplied the signal phase $\phi_n$ by $M$ one can apply a nonlinear function to $\rho_n$ to give,

$$z'_n = \mathcal{F}(\rho_n) \exp(i\,M\phi_n) \tag{7.24}$$

The paper [61] has considered the analysis of a feedforward carrier phase recovery system where,

$$\mathcal{F}(\rho_n) = \rho_n^K \tag{7.25}$$

where $K = 0, 2 \ldots M$. The analysis for this is very complex and considers only even powers of $\rho_n$. This is not because one cannot use odd powers of $\rho_n$ but it is only for even powers that the analysis is tractable.

The $z'_n$ are then averaged using a sliding-mean filter of length $N = 2L + 1$ samples. A sliding-mean filter is optimal provided the symbol to symbol noise samples are uncorrelated. So, finally the output phase estimate from the V & V algorithm is,

$$\psi = \frac{1}{M} \arg \left\{ \frac{1}{2L+1} \sum_{n=-L}^{+L} \rho_n^K \exp(i M \phi_n) \right\} \tag{7.26}$$

It appears that the real design choice for the V & V algorithm is the power to which the amplitude is raised, $K$. The "stripped-down" implementation of the V & V algorithm in SPW is as shown in figure 7.13. Note from the figure that the input signal is delayed by $L$ samples before being phase-rotated. This is because if the input signal has a static frequency offset then the phase estimate in the centre of the sliding-mean filter is unbiased.

One should note that the V & V algorithm in its basic form presents an estimate of the carrier phase which is unbiased modulo $2\pi/M$. To prevent frequency jitter one needs to *unwrap* the carrier phase. This is shown in figure 7.13. This implies that the basic system described above and shown in figure 7.13 will cycle slip when tracking a static frequency offset – unlike a feedback error-tracking structure. Viterbi suggests getting around this by differentially encoding the data on transmission and differentially decoding the data upon reception, but error bursts of two symbols will still occur on every cycle slip. This will occur less often the smaller the frequency offset and may be tolerable. If it is not tolerable, because the frequency error is too large, then one should combine a frequency error detector with the V & V algorithm such as the rotational FED discussed previously.

The V & V algorithm is simply a modification to the ML estimate of the phase of a complex tone in AWGN. When using a sliding-mean filter of length $L$ the Cramer-Rao lower bound is simply [29],

$$\text{var}(\phi) \leq \frac{1}{L\gamma} \tag{7.27}$$

where $\gamma$ is the signal to noise ratio. Justification for this is shown in figure 7.14 for sliding-mean filters of length $L = 10$ and $L = 100$.

Note at low values of signal to noise the curves are upper bounded by the variance of a uniform phase distribution over $-\pi \leq \phi \leq \pi$ for a tone and $-\pi/M \leq \phi \leq \pi/M$ for $M$-PSK modulation. For $M$-PSK modulation this upper bound is,

$$\text{var}(\phi) \leq \frac{(\pi/M)^2}{3} \tag{7.28}$$

This value then corresponds to approximately 3.28 in figure 7.14 and 0.21 in figure 7.15 which shows the tracking performance of the V & V algorithm as a function of signal to noise (actually $E_s/N_o$) with $K$ a parameter. The ideal Cramer-Rao bound is also shown and it can be seen that the results asymptotically approach this limit for all values of $K$. Inspection of the graphs reveal that for moderate to high values of signal to noise ratio $K = 1$ is the best

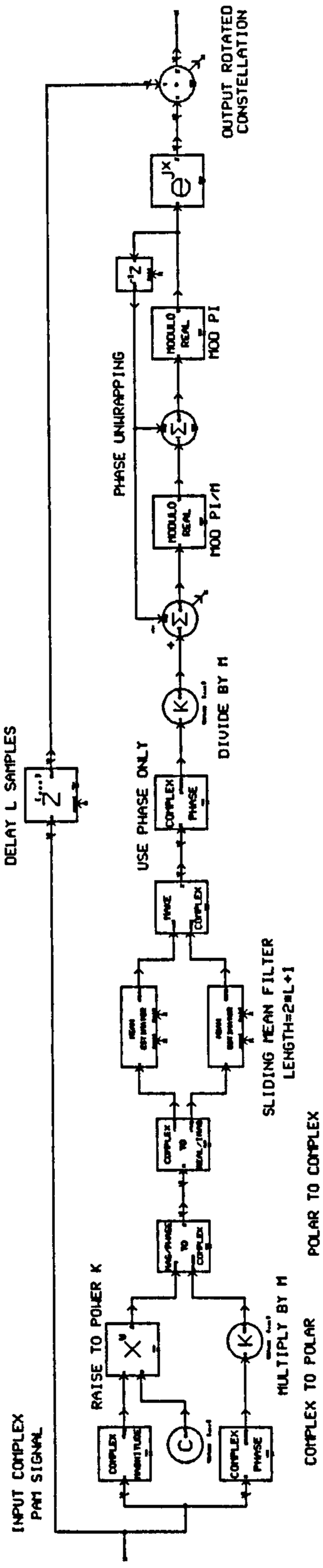
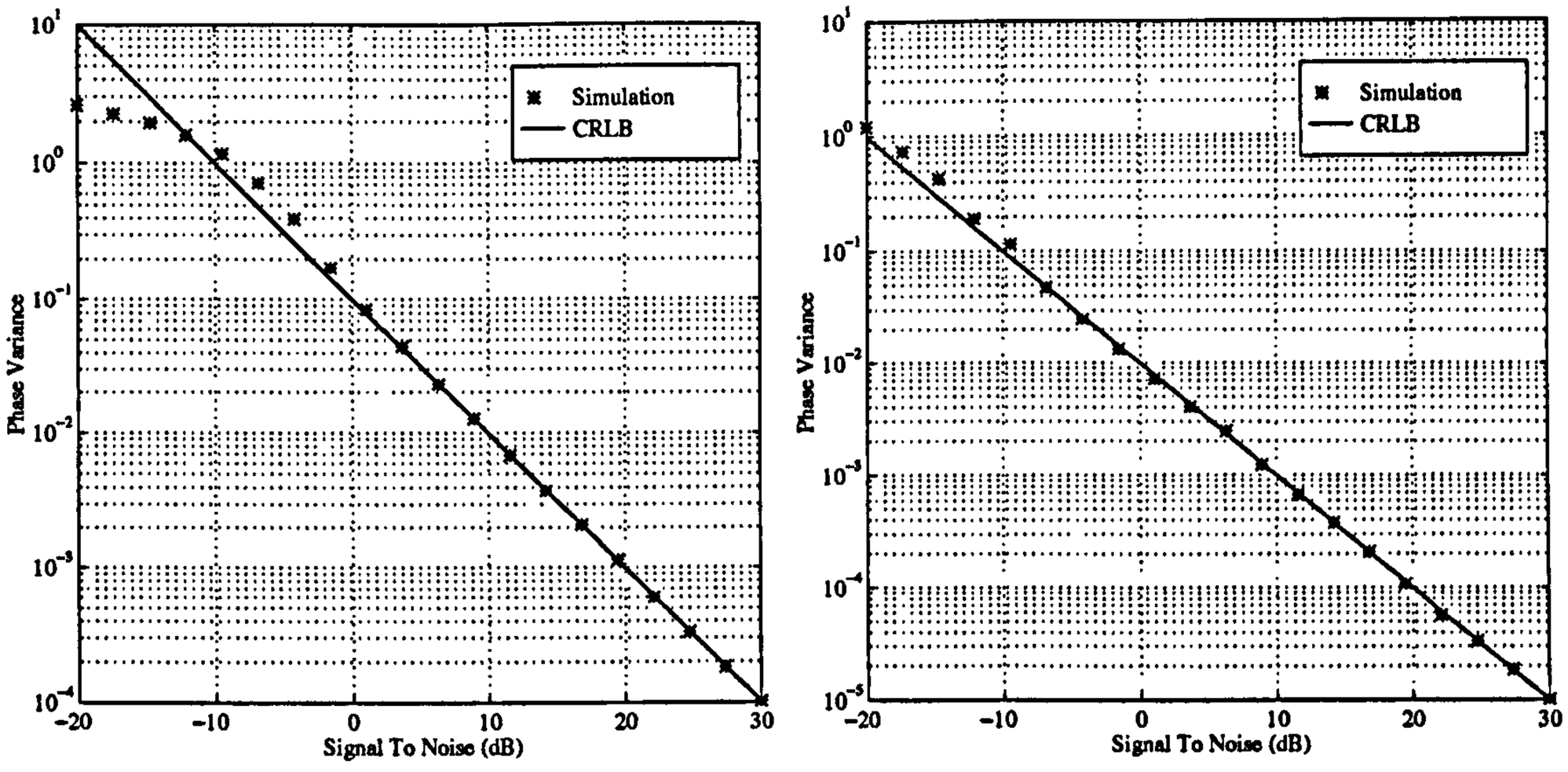



Figure 7.13: SPW implementation of feedforward carrier phase synchronisation using the V & V algorithm.




(a) $L = 10$ (b) $L = 100$

Figure 7.14: Estimation of the Phase of a Complex Tone in AWGN using a Sliding-Mean Filter of Length $L$. Performance Results Together with Cramer-Rao Bound.

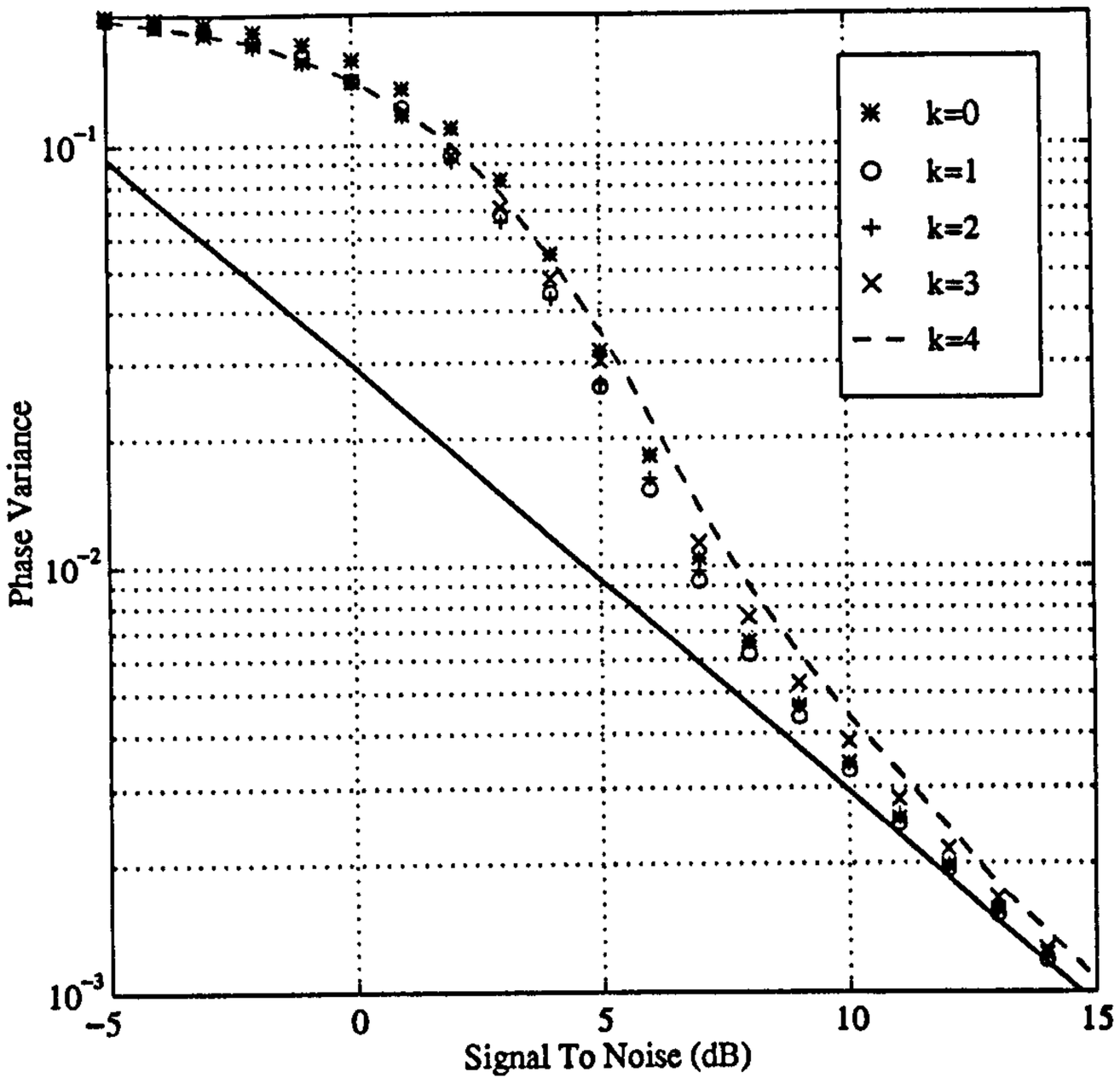
choice. This result was not shown analytically in [61] but was substantiated by simulation. It does appear to have been disregarded in the conclusions in the paper, however, perhaps because of a lack of analysis for this particular case. For low values of signal to noise ratio $K = 0$ is the best choice. This has an obvious advantage from an implementation point of vie in that automatic gain control (AGC) becomes unnecessary, for the nonlinearity can be implemented as a bandpass limiter. Setting $K = 4$ appears to be a poor choice for all values of signal to noise ratio.

The results in figure 7.15 were obtained using the MATLAB simulation program in appendix B. The number of iterations per point were 100000 and the number of taps in the sliding-mean filter was $L = 17$.

## 7.6 Detection Schemes for GMSK

### 7.6.1 Ideal Orthogonal Coherent Detection GMSK

A simple orthogonal-coherent detector for the detection of MSK-type waveforms assuming perfect synchronisation is shown in figure 7.16 in its ideal form and in figure 7.17 configured from basic SPW building blocks [72]. The JK-Flip Flops shown in the detail of the orthogonal coherent detector are configured from the SPW kernel blocks as shown in figure 7.18. The bit error rate performance of the receiver system illustrated in figure 7.17 matches that of the theory in the literature [40].



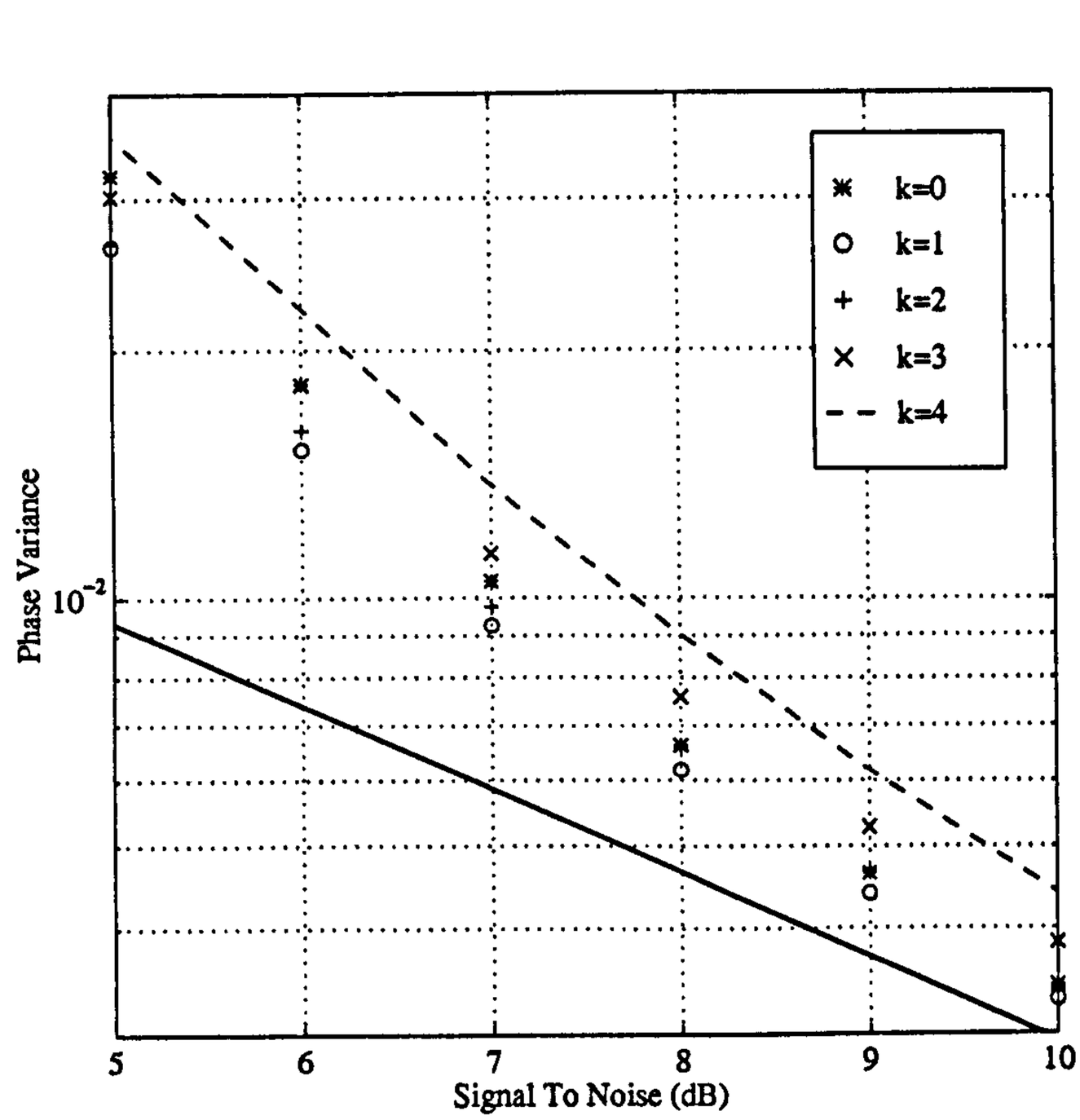




Figure 7.15: Tracking Performance of V & V algorithm. Number of taps in sliding-mean filter is $L = 17$. Finer detail shown in lower graph at "moderate" values $E_s/N_o$.

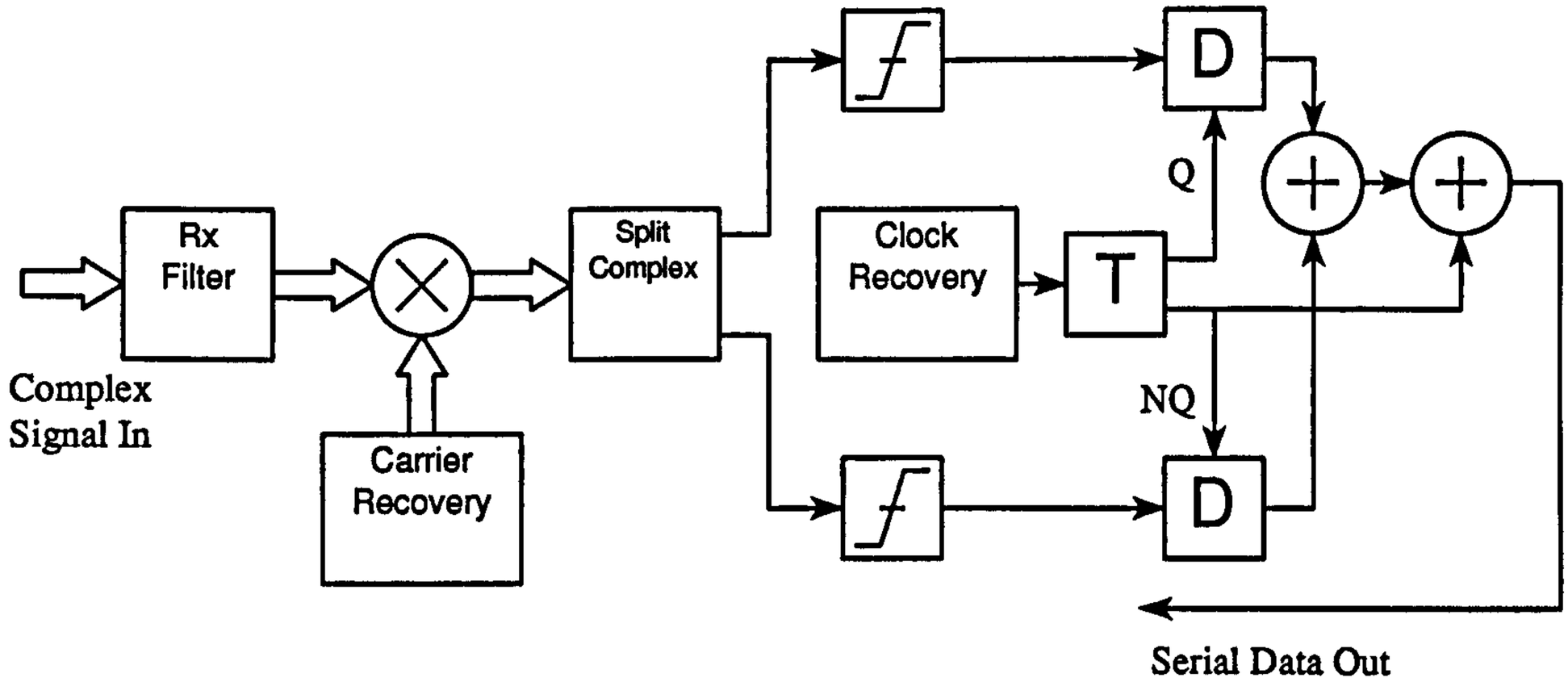



Figure 7.16: Ideal Orthogonal coherent detector for MSK type waveforms

The input signal is filtered by a Gaussian receiver filter and then split directly into real and imaginary components (in-phase and quadrature). These are then thresholded and clocked by an odd and an even clock at half the bit rate. Thus, the JK flip-flop in the centre of the figure acts as a T-type latch whereas the two JK flip-flops in the arms of the receiver act as D-type latches. The two exclusive-or gates on the output of the receiver are necessary to decode the MSK data.

The ideal orthogonal-coherent system, with perfect synchronisation assumed and in a white Gaussian noise channel can be characterised quite easily using analytical methods. Analytical methods become less useful where one has to derive clock and carrier synchronisation and the channel noise statistics may be non-stationary. Under such circumstances one generally has to resort to simulation or a combination of simulation and analytical techniques. In the next chapter we will illustrate a method of jointly recovering the clock and the carrier for MSK-type modulations.

### 7.6.2 Noncoherent Differential Phase Detection of GMSK

In this section we shall examine the one and two-bit differential detection of GMSK and then go on to show how decision-feedback equalisation can be used to improve the bit-error rate performance.

#### One Bit Differential Detection GMSK

The one-bit detection process consists of multiplying the received signal by a delayed and 90° phase shifted version of itself and then lowpass filtering this product. This is illustrated mathematically as follows:

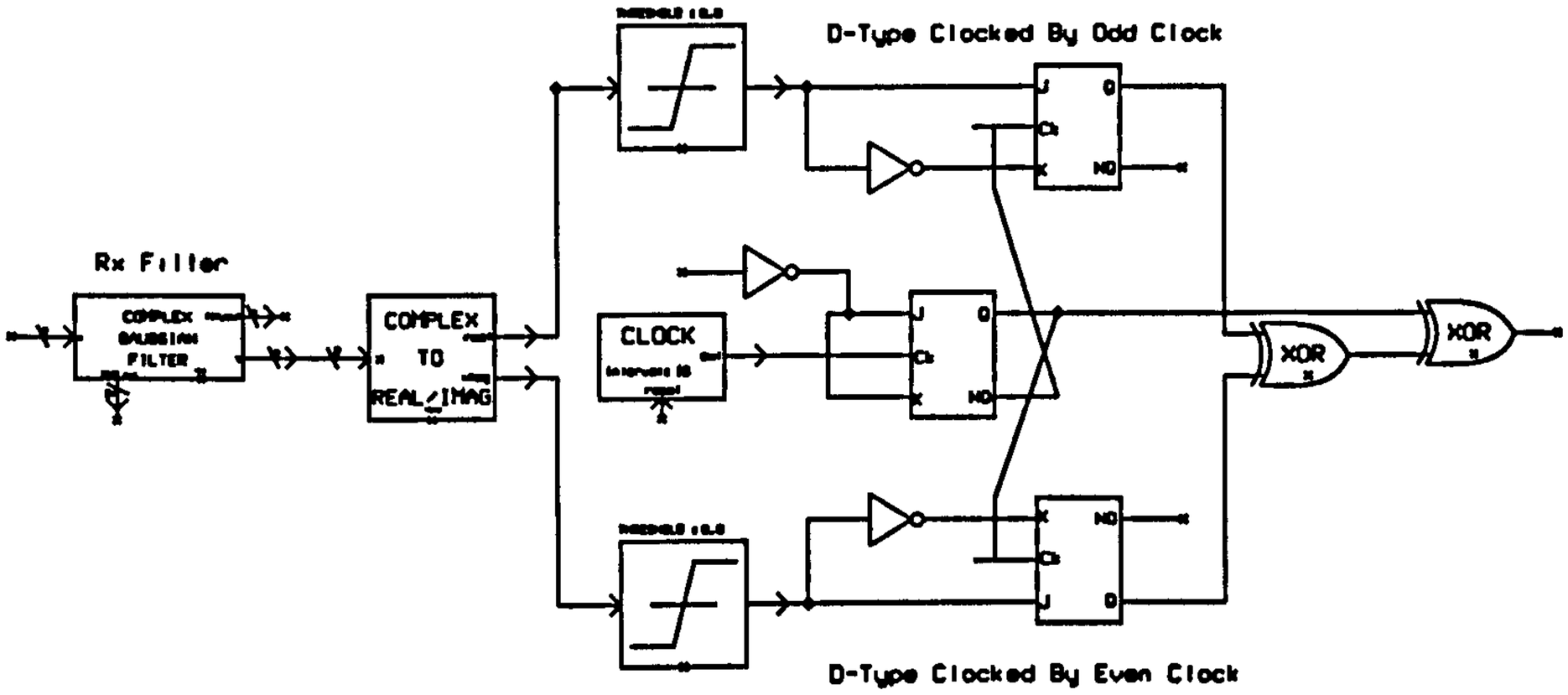



Figure 7.17: Orthogonal coherent detector assuming perfect synchronisation

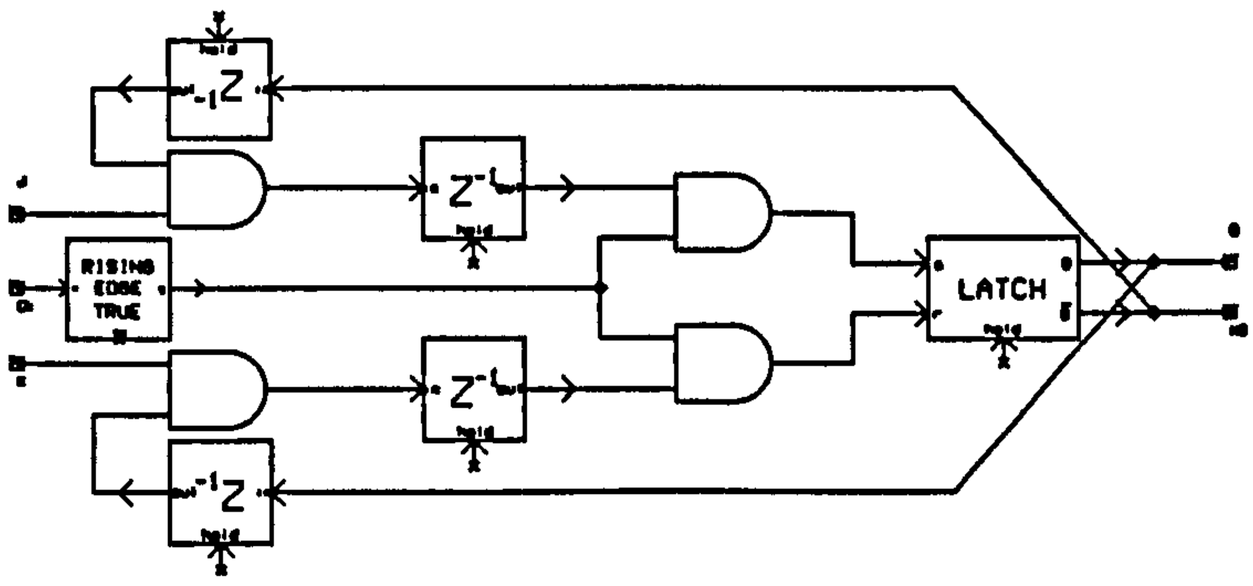



Figure 7.18: Detail of the JK Flip-Flop used in model of orthogonal-coherent detectors

$$Y_1(t) = r(t-T)\cos\{\omega_c(t-T) + \phi(t-T) - \frac{\pi}{2}\} \times r(t)\cos\{\omega_c t + \phi(t)\} \tag{7.29}$$

When this is lowpass filtered to reject second-harmonic terms we get

$$Y_1(t) = \frac{1}{2}r(t-T)r(t)\sin(\Delta\Phi(T)) \tag{7.30}$$

where we have used the fact that for MSK $\omega_c T = 2\pi k$ where $k$ is an integer and $\Delta\Phi(T) = \phi(t) - \phi(t-T)$.

The decision rule for one-bit differential detection is simple, if $Y_1(T) > 0$ then a $+1$ was sent else it was a $-1$.

### Two Bit Differential Detection GMSK

For the two bit differential detector the input data stream is multiplied by a two-bit delayed version of itself and lowpass filtered. The analysis of this is as follows:

$$Y_2(t) = r(t-2T)\cos\{\omega_c(t-2T) + \phi(t-2T)\} \times r(t)\cos\{\omega_c t + \phi(t)\} \tag{7.31}$$

Again, when this is lowpass filtered we get,

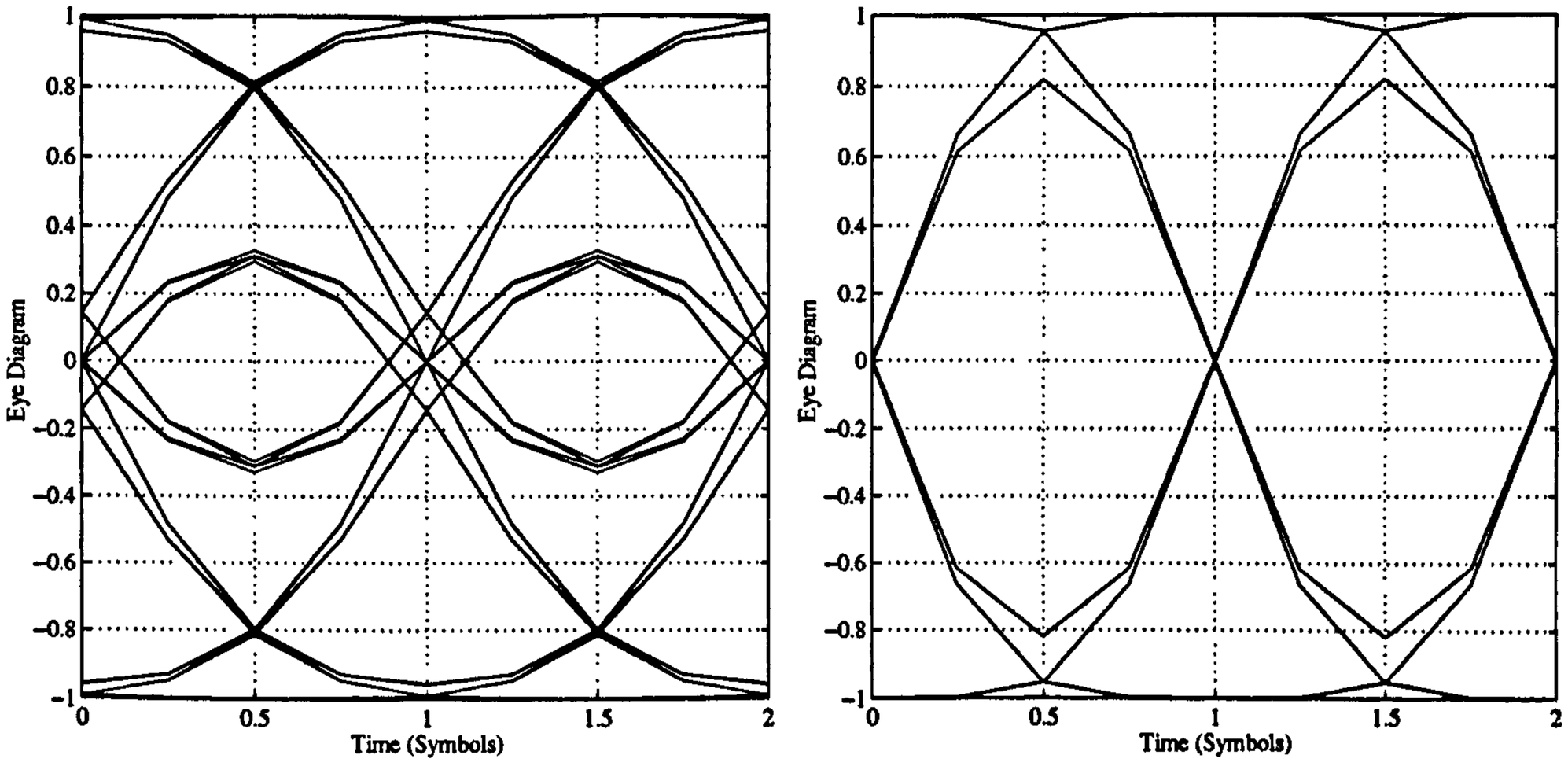
$$Y_2(t) = \frac{1}{2}r(t-2T)r(t)\cos(\Delta\Phi(2T)) \tag{7.32}$$

Note that the output from the 2bDD is asymmetrical when detecting schemes with ISI. This is illustrated by comparing the outputs from the 1bDD and 2bDD when detecting GMSK as shown in figure 7.19 and figure 7.21. The 1bDD scheme depends only on the phase change over the one bit period whereas the 2bDD scheme will also depend upon the signal amplitudes. A hard-limiter is thus used in conjunction with the 2bDD to improve the performance [53].

## 7.6.3 Eye & Scatter Diagrams of Differentially Detected GMSK

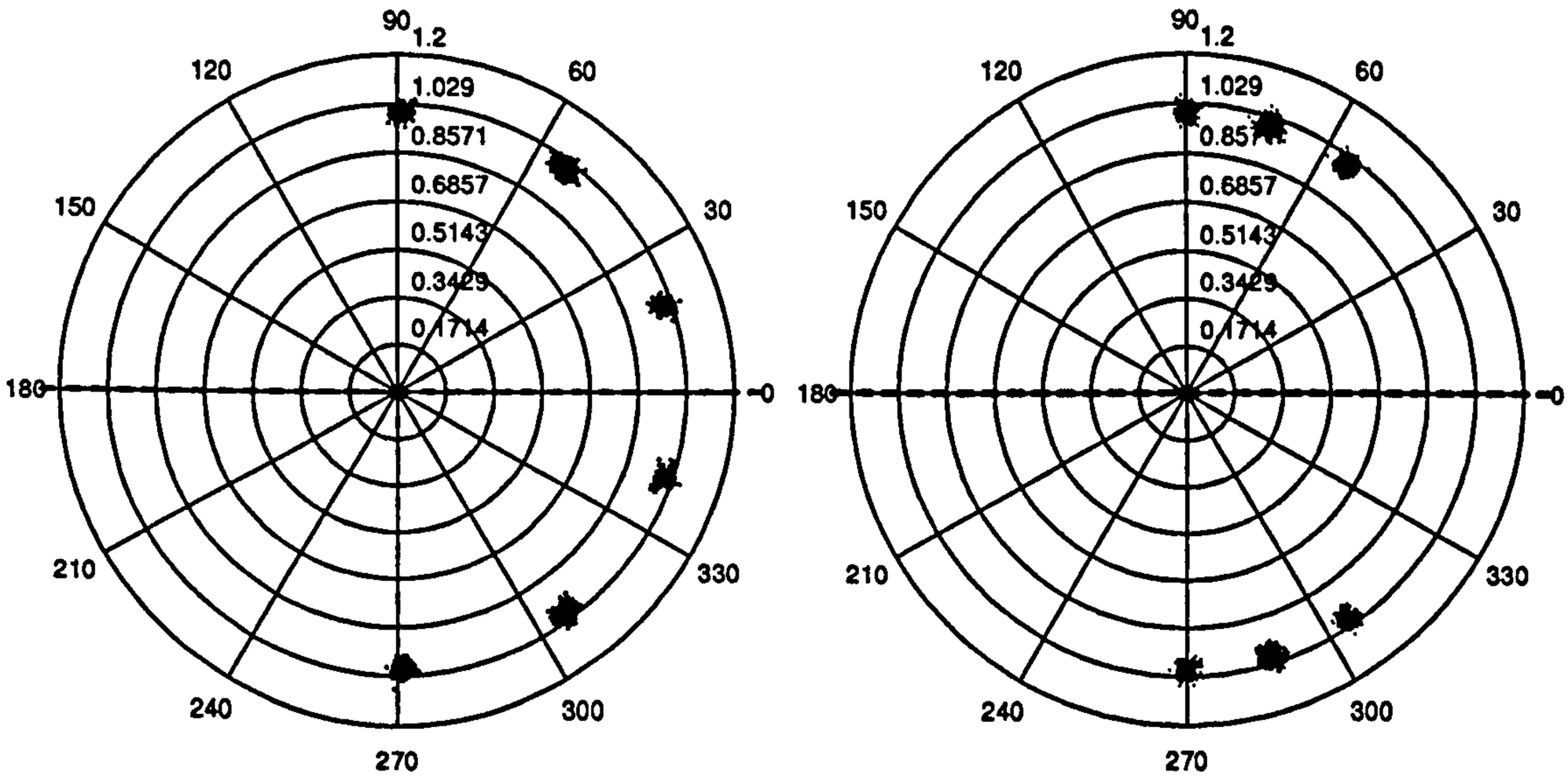
For the case of 1bDD the eye-diagrams at the output of the receiver are shown in figure 7.19 for pre-modulation time-bandwidth, $B_t T$, products of 0.25 and 0.5 respectively. Notice how the Inter-Symbol Interference (ISI) is less pronounced in those schemes that have a higher value of $B_t T$. Minimum Shift Keying has the lowest value with a non-existent pre-modulation filter (or a filter with a $B_t T = \infty$). Note also, that because the output of the detector is dependent on a sine function then the eye-diagram is symmetric.

Also plotted in figure 7.20 are the scatter diagrams corresponding to this scheme. For ideal differential detection of MSK there would be two states in the scatter diagram at $\pm 90°$. The extra phase states are due to the inherent inter-symbol interference in the scheme. We can quantify what this inter-symbol interference is, however, and hence determine *a priori*




(a) $B_tT = 0.25$ (b) $B_tT = 0.5$

Figure 7.19: Eye diagrams at input to decision circuit for 1-Bit differential detection GMSK.




(a) $B_tT = 0.25$ (b) $B_tT = 0.5$

Figure 7.20: Scatter diagrams at input to decision circuit for 1-Bit differential detection GMSK.

what these phase states are. We use the equation derived in (3.12) for the phase trajectory of Gaussian pre-filtered MSK which is restated here as:

$$\Phi(t) = \frac{1}{2} + \frac{t_2}{2}\,\mathrm{erf}\left(\frac{\sigma t_2}{\sqrt{2}}\right) - \frac{t_1}{2}\,\mathrm{erf}\left(\frac{\sigma t_1}{\sqrt{2}}\right) + \frac{1}{\sqrt{2\pi}\sigma}\left\{\exp\left(-\frac{\sigma^2 t_2^2}{2}\right) - \exp\left(-\frac{\sigma^2 t_1^2}{2}\right)\right\} \tag{7.33}$$

As we saw earlier, MSK would have a phase trajectory over 90° in a single symbol interval as it does not have any inter-symbol interference. It is the excess phase of $\Phi(t)$ in an adjacent symbol period which determines quantitatively what the ISI is as,

$$\theta_{ISI} = \left\{\Phi\left(\frac{3T}{2}\right) - \Phi\left(\frac{T}{2}\right)\right\} \times 90° \tag{7.34}$$

This then determines the ISI for $B_tT = 0.25$ and $B_tT = 0.5$ as[1]:

$$\theta_{ISI} = 17.9396° \quad \text{for} \quad B_tT = 0.25 \quad \text{and} \quad \theta_{ISI} = 9.5142° \quad \text{for} \quad B_tT = 0.5 \tag{7.35}$$

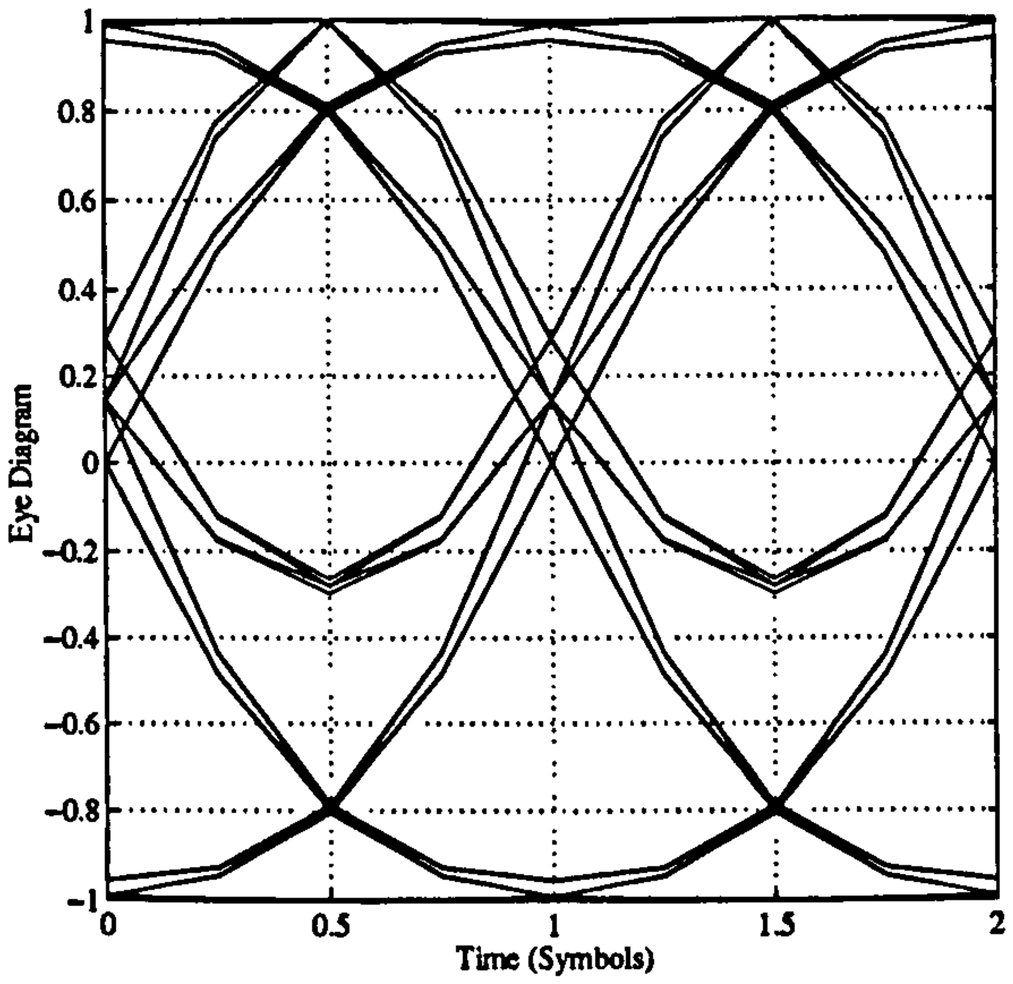
It is found that the use of 2-Bit Differential Detection improves the performance over the one bit case due to the fact that the eye-openings are wider, but it is noticed that they are now asymmetric with respect to the Y-axis. This is due to the cosine function in the detector characteristics and illustrated by figure 7.21. This asymmetry becomes more marked for smaller values of pre-modulation filter time-bandwidth product. It is noted [53] that because of the asymmetry a non-zero detection threshold is optimum for such a scheme.

### 7.6.4 Decision Feedback Equalisation Schemes to Improve the Bit Error Rate of DD-GMSK

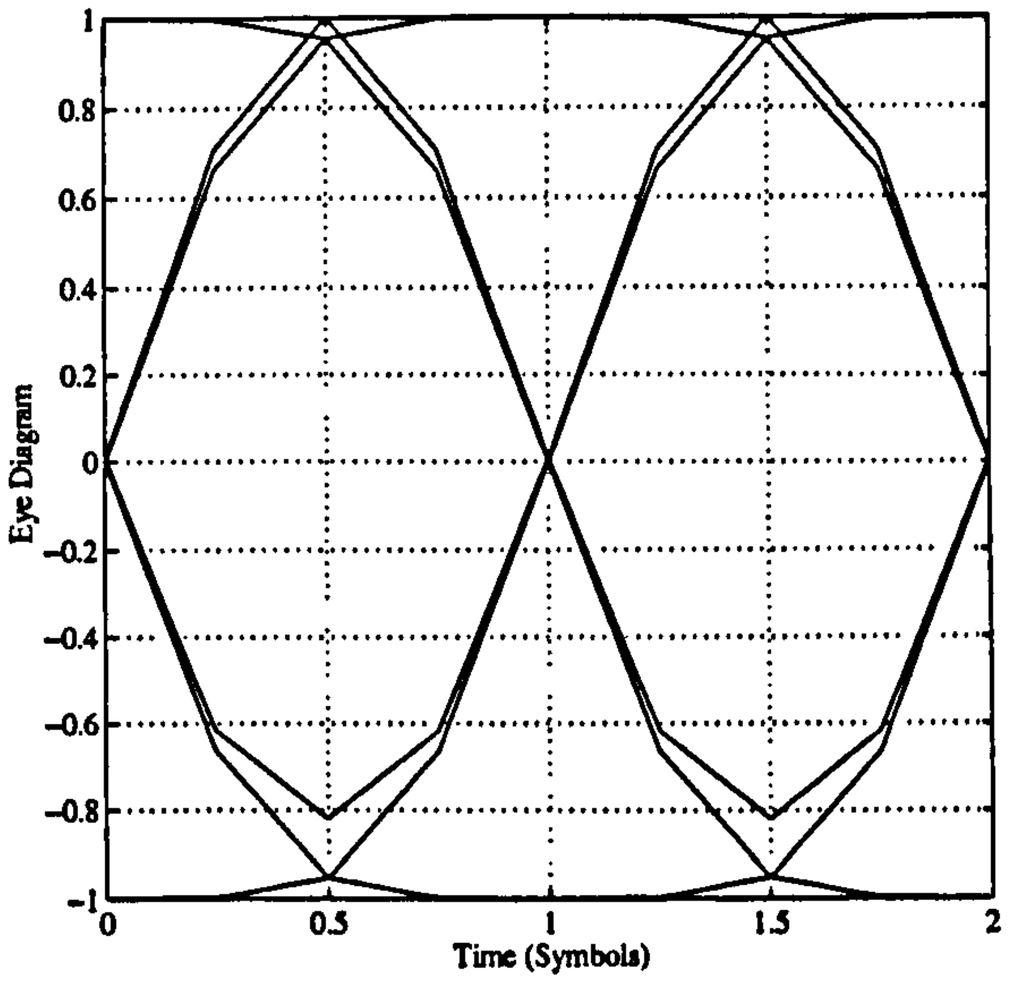
This section is concerned with applying techniques for reducing Inter-Symbol Interference (ISI) in Gaussian Minimum Shift Keying (GMSK). The technique studied is called Decision-Feedback Equalisation (DFE) and is a means of reducing ISI at the receiver from a knowledge of how much interference is present from adjacent data symbols. The technique is applicable mainly to incoherent detection of GMSK as this is where ISI is more of a problem. By reducing the ISI in any given modulation scheme, one automatically improves the error performance of that scheme. The ISI is there in the first place though because of the need to have a narrowband modulation scheme.

The technique of reducing the ISI in the receiver is known as *equalisation.* Linear equalisation is where one cascades with the channel response a filter such that the overall response has no, or a minimum amount, of ISI. When the characteristics of the channel are changing then the response of the receiver filter has to change in sympathy in order to maintain a fixed overall response. Such equalisation is called Adaptive Equalisation and is used extensively in mobile communications where the channel response is modelled as a time-variant linear

[1]Note these results are slightly at odds with the following discussion where I have said that $\theta_{ISI} = 18.5°$ for $B_tT = 0.25$. This is because later results where determined by numerical integration of the gaussian pulse response whereas these results are analytic. The difference is small though.

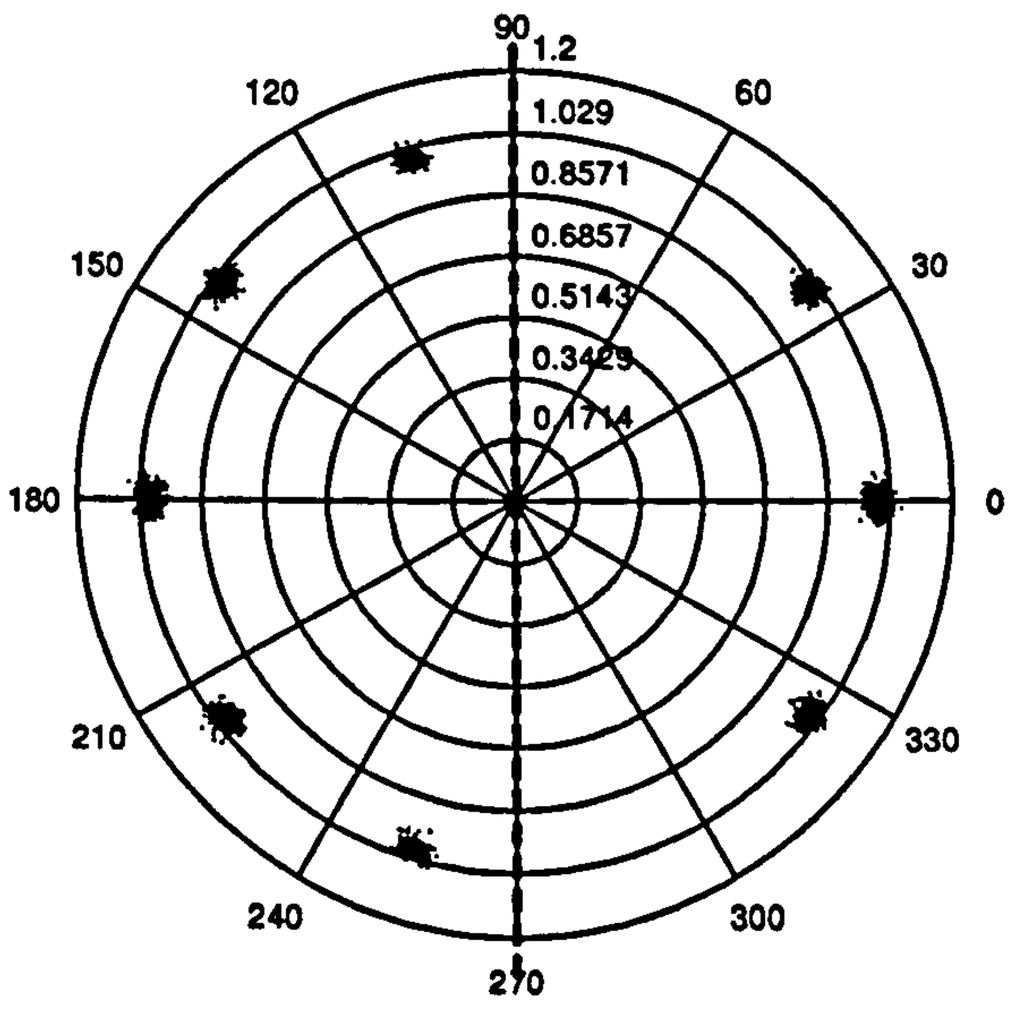



(a) $B_tT = 0.25$

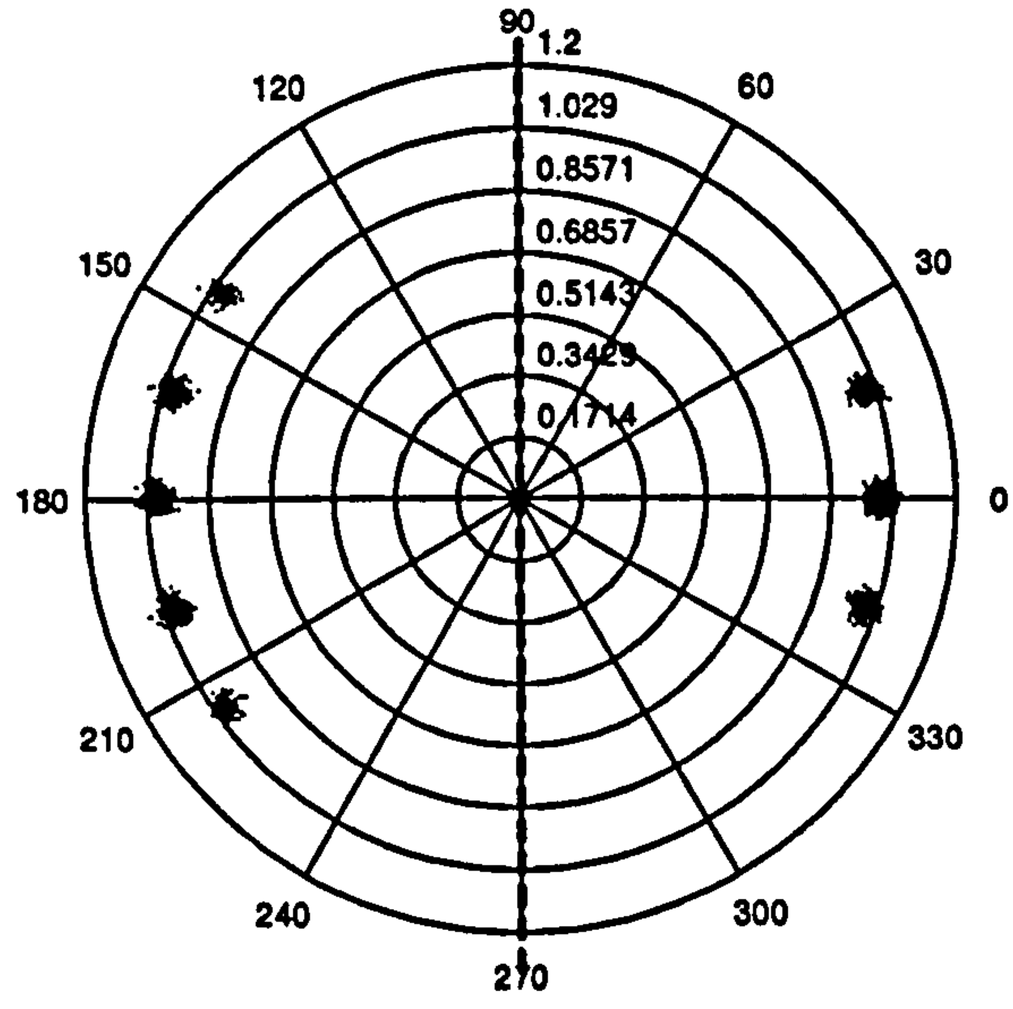



(b) $B_tT = 0.5$

Figure 7.21: Eye Diagram at Input to Decision Circuit For 2-Bit Differential Detection GMSK.




(a) $B_tT = 0.25$




(b) $B_tT = 0.5$

Figure 7.22: Scatter Diagram at Input to Decision Circuit For 2-Bit Differential Detection GMSK.

filter. Decision-Feedback Equalisers (DFE) and Maximum-Likelihood Sequence Estimators (MLSE) are non-linear methods of equalisation. This is because they need to rely on previous estimates of the data symbols. The process of making a decision about a particular data symbol involves a thresholding which is a non-linear operation.

### DFE Applied to 1bDD GMSK

One can quantify the Inter-Symbol Interference effects that the Gaussian lowpass filter has on the basic MSK modulation format by looking at the inphase and quadrature components at the sampling instants. This is illustrated by a scatter diagram as shown in figure 3.11. Simple MSK can be represented as four phase PSK with phase states at 0°, ±90° and ±180°. The effect of the Gaussian lowpass filter, with $B_tT = 0.25$, is to produce extra states at approximately ±37°, ±53°, ±127° and ±143° where the deviation from the four-phase MSK representation is equal twice the ISI given by approximately 18.5°. According to Proakis [30] if the modulation index in CPM can be represented as the ratio of two relatively prime integers such that $h = m/p$, then the number of phase states over $L$ symbol intervals is given by,

$$S_t = pM^{L-1} \text{ for m even and } S_t = 2pM^{L-1} \text{ for m odd}$$

where $M$ is the alphabet size. Thus, for GMSK with $h = \frac{1}{2}$ and $BT = 0.25$, if we put $L = 2$ symbols then $S_t = 8$ and if we put $L = 3$ symbols then $S_t = 16$. However, the gaussian impulse response of the remodulating filter extends over all time and so there are an infinite number of phase states which get successively closer together. In figure 3.11 the number of main phase states is 12 which implies that the impulse response of the Gaussian lowpass filter is 'significant' over approximately 2.6 symbol periods which is a further justification of only taking in to account ISI due to the first adjacent pulse. The maximum ISI occurs for sequences of the form $+1 -1 +1$ and $-1 +1 -1$.

If we look at the one-bit differential detector scatter plot then the points now represent a given phase change for an input data sequence. Thus for the sequences $+1 -1 +1$ and $-1 +1 -1$ represent a phase change of $\pm 16°$ respectively, which is $\pm(90° - 4 \times \theta_{ISI})$, whereas the sequences $+1 +1 +1$ and $-1 -1 -1$ represent the maximum possible phase change of ±90° respectively. Note that as we are now dealing only with how the phase *changes* from one symbol interval to the next the points in the scatter diagram only occupy the positive real axis. The relationship between the input data sequence and the occupied phase states is

shown in the following table:

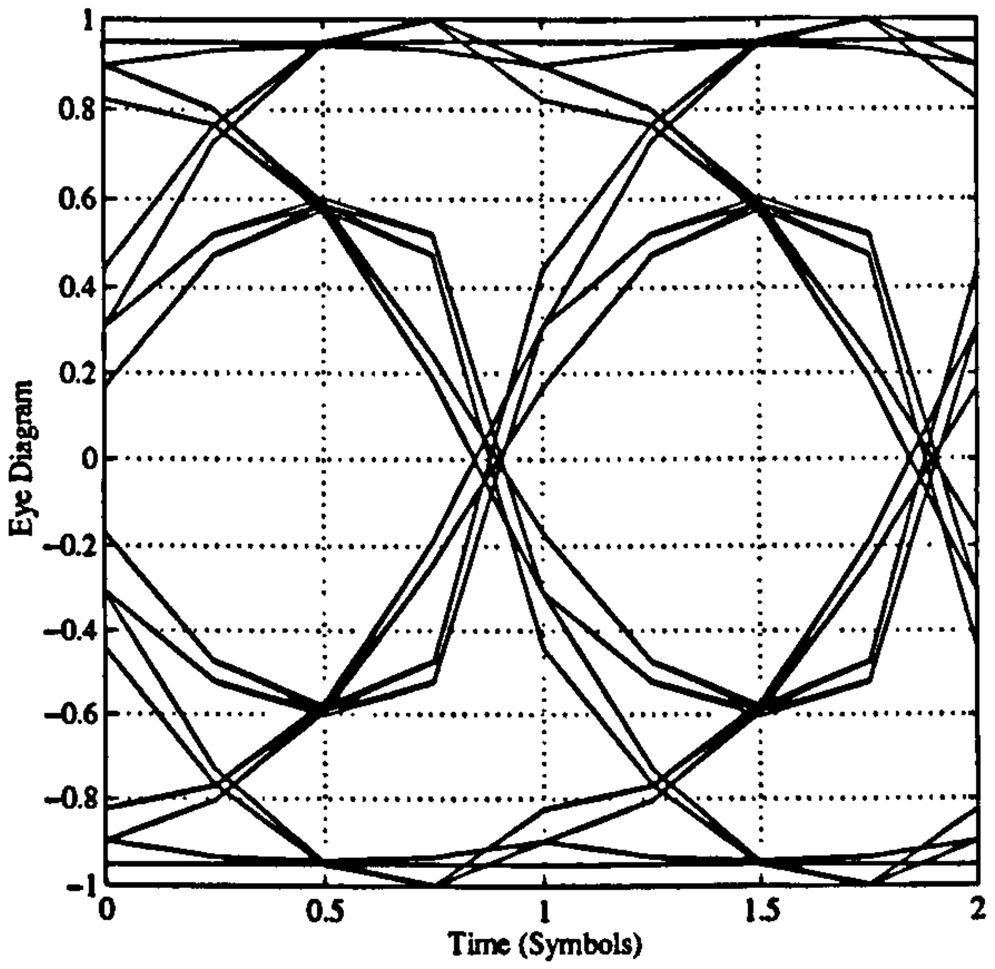
| $b_{k-1}$ | $b_k$ | $b_{k+1}$ | $\theta$ |
|---|---|---|---|
| 0 | 0 | 0 | −90° |
| 0 | 0 | 1 | −53° |
| 0 | 1 | 0 | +16° |
| 0 | 1 | 1 | +53° |
| 1 | 0 | 0 | −53° |
| 1 | 0 | 1 | −16° |
| 1 | 1 | 0 | +53° |
| 1 | 1 | 1 | +90° |

The basis by which DFE works is to manipulate the points in the scatter diagram to increase the minimum angular separation of the points in the positive and negative half of the scatter plot. Thus, without feedback the minimum separation is 32°. We can increase this by applying the following rule, if the previous bit was a −1 then rotate the signal by vector by +18.5° and if the previous bit was +1 then rotate the signal vector by −18.5°. The result of this is to change the signal constellation from six points with a minimum separation of 32°, to four points with a minimum separation of 71° (compare figure 7.20 with figure 7.24). This corresponds to an increase in the eye opening as shown in figure 7.23 and a corresponding improvement in the BER performance as we shall see.

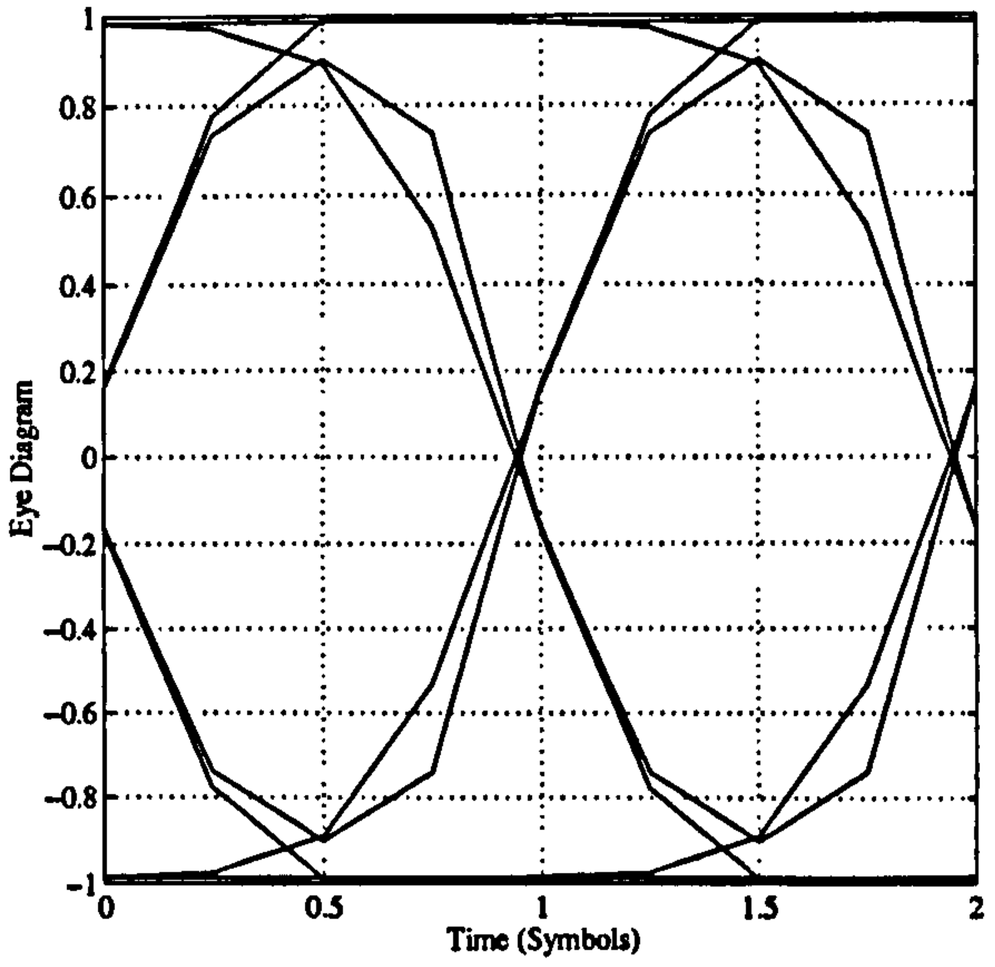
Note that we can not completely get rid of all the ISI because we do not have prior knowledge of the data bit $b_{k+1}$.

### DFE Applied to 2bDD GMSK

DFE applied to two bit differential detection is directly analogous to the one bit case, except here we are looking at the data stream over two bit intervals. Intuitively, one would imagine that by looking at the data stream over a longer period of time would lead to less corruption from ISI – this is indeed the case. In two bit differential detection we are looking at phase *changes* over that two bit interval. Thus the scatter diagram for two bit differential detection has phase states throughout the whole of the ±180° range. The input data sequence for two

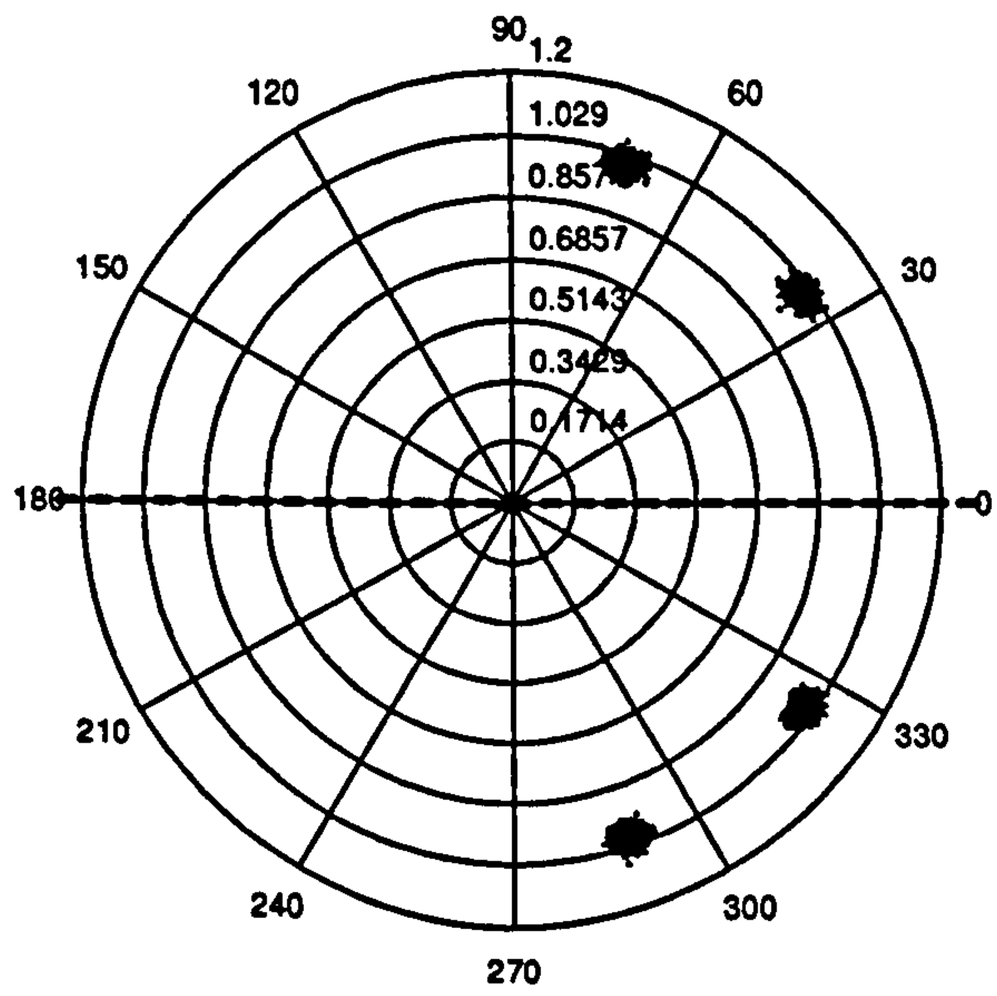



(a) $B_tT = 0.25$

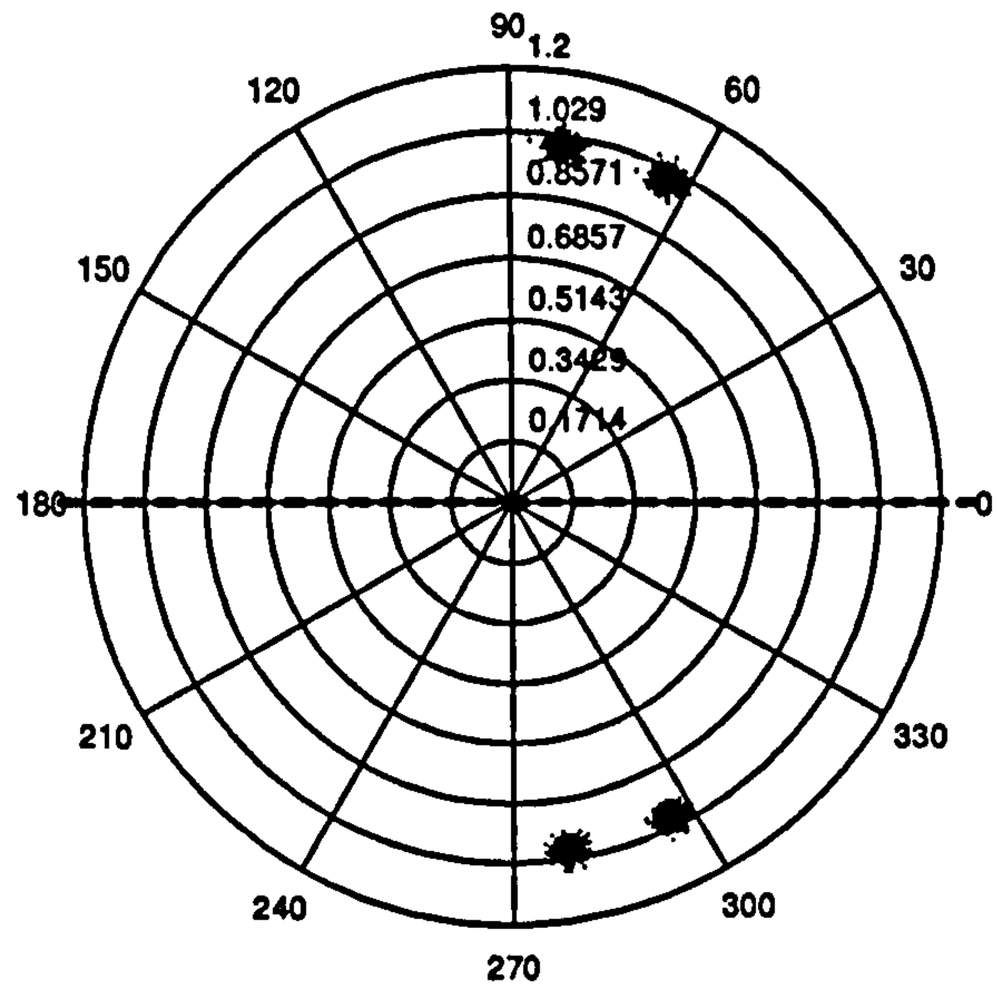



(b) $B_tT = 0.5$

Figure 7.23: Eye diagrams at input to decision circuit for 1-Bit differential detection GMSK with decision feedback equalisation.




(a) $B_tT = 0.25$




(b) $B_tT = 0.5$

Figure 7.24: Scatter diagrams at input to decision circuit for 1-Bit differential detection GMSK with decision feedback equalisation.

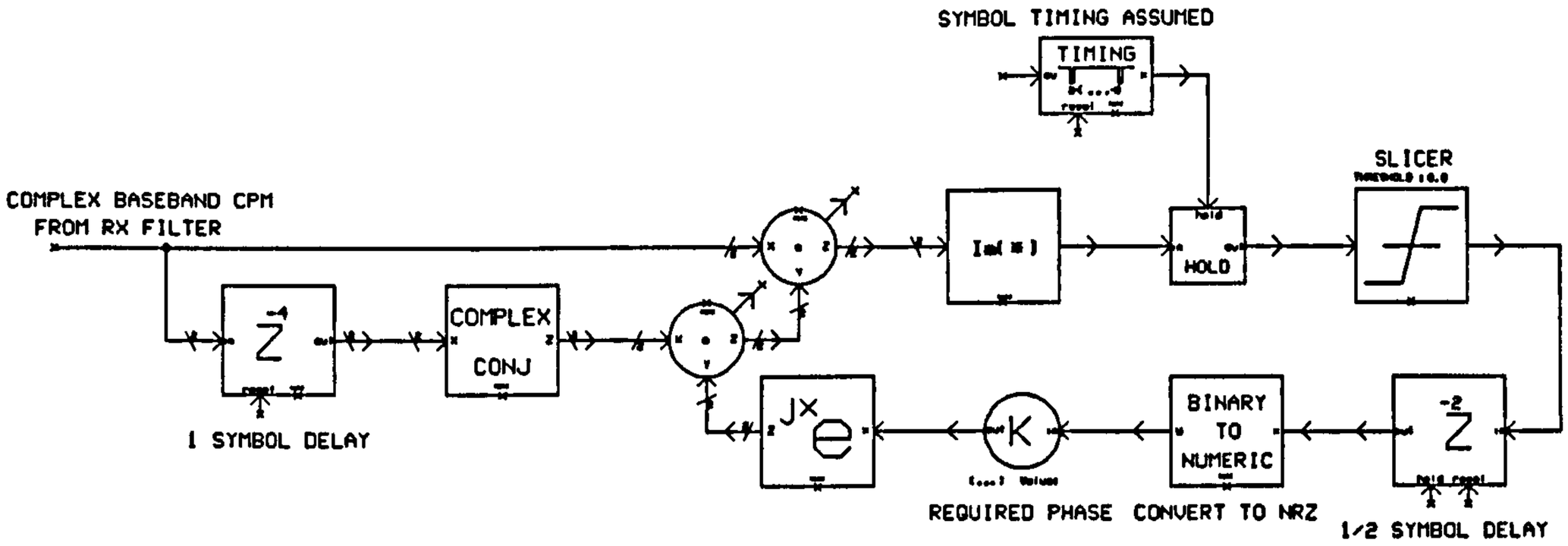


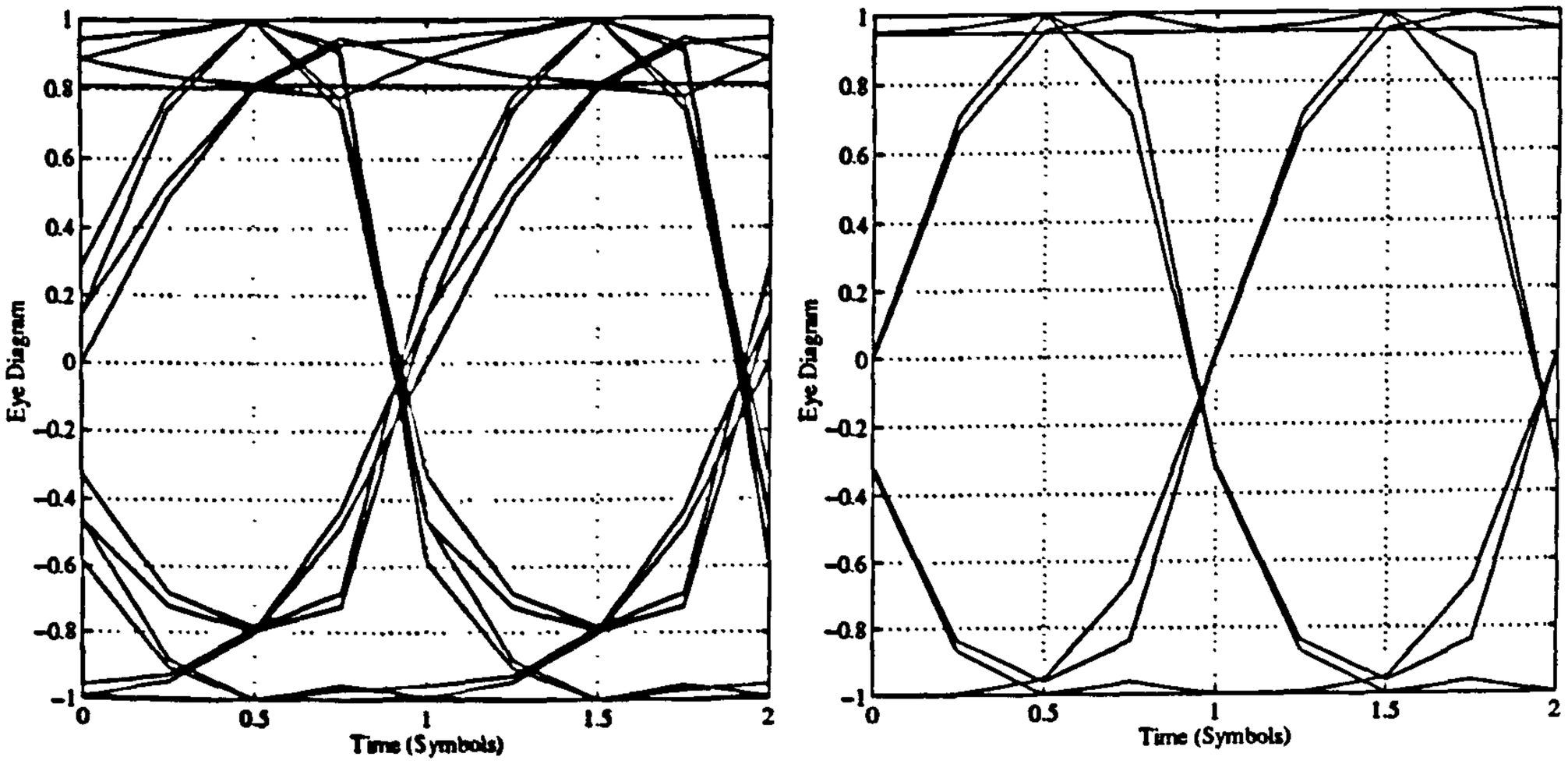

Figure 7.25: SPW implementation of 1bit differential detection of prefiltered MSK with decision feedback equalisation

bit differential detection and the corresponding phase states are shown in the table below:

| $b_{k-2}$ | $b_{k-1}$ | $b_k$ | $b_{k+1}$ | $\theta$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $-180°$ |
| 0 | 0 | 0 | 1 | $-143°$ |
| 0 | 0 | 1 | 0 | $-37°$ |
| 0 | 0 | 1 | 1 | $0°$ |
| 0 | 1 | 0 | 0 | $-37°$ |
| 0 | 1 | 0 | 1 | $0°$ |
| 0 | 1 | 1 | 0 | $+106°$ |
| 0 | 1 | 1 | 1 | $+143°$ |
| 1 | 0 | 0 | 0 | $-143°$ |
| 1 | 0 | 0 | 1 | $-106°$ |
| 1 | 0 | 1 | 0 | $0°$ |
| 1 | 0 | 1 | 1 | $+37°$ |
| 1 | 1 | 0 | 0 | $0°$ |
| 1 | 1 | 0 | 1 | $+37°$ |
| 1 | 1 | 1 | 0 | $+143°$ |
| 1 | 1 | 1 | 1 | $+180°$ |

Note that the scatter diagram is asymmetric as is the eye-diagram . Also, for two-bit differential detection the decision rule is based on $\cos(\Delta\Phi(2T))$ and so the function of the DFE is to manipulate the points so that they are a maximum distance away from the imaginary axis in the scatter diagram. Before decision feedback the minimum angular separation is due to those points at $\pm 106°$. The DFE moves these points around to $\pm 143°$ respectively by feeding back the following phase shift:

$$\lambda = b_{k-2} \oplus b_{k-1} \times (2b_{k-2} - 1) \times 37° \tag{7.36}$$




(a) $B_tT = 0.25$ (b) $B_tT = 0.5$

Figure 7.26: Eye diagrams at input to decision circuit for 2-Bit differential detection GMSK with decision feedback equalisation.

The block diagram for simulating this DFE system is shown in figure 7.28 and the expanded eye-diagram due to the DFE is shown in figure 7.26.

This method of using decision feedback equalisation can be extended beyond 2 bit differential detection where the reduction in ISI is even more pronounced.

## 7.7 Performance of Schemes

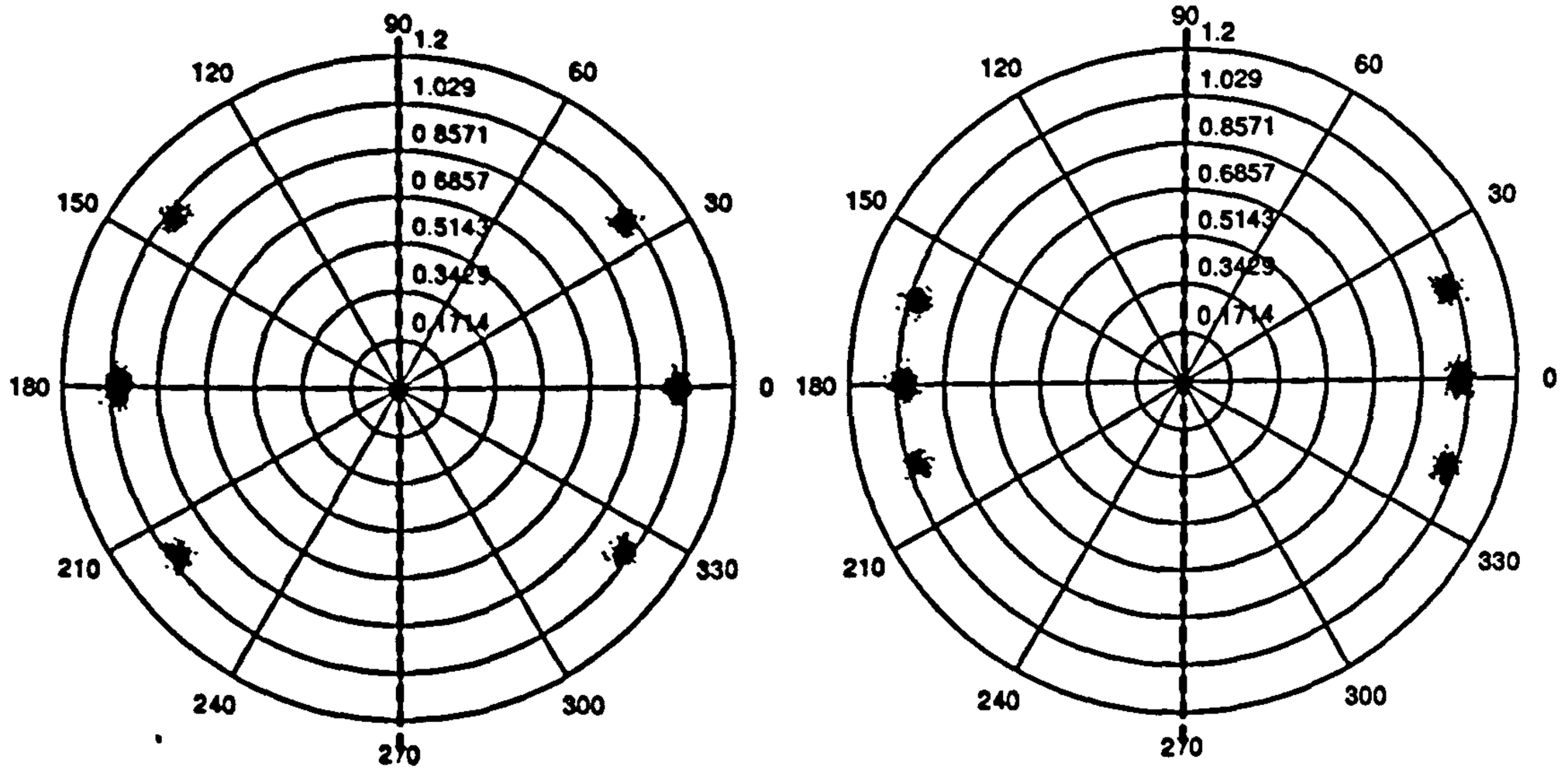
### 7.7.1 Coherent Schemes

The optimum Gaussian receiver filter bandwidth for the ideal orthogonal coherent detector is found by simulation to be have a cutoff of $B_tT = 0.63$ [40]. The BER performance of the coherent synchronisers in AWGN is as shown in figure 7.29 for MSK and for GMSK with $B_tT = 0.25$.

### 7.7.2 Noncoherent Schemes

The performance improvement that can be achieved for 1bDD GMSK by using Decision Feedback Equalisation is illustrated in figure 7.30.

Similarly the performance improvement in using 2bDD over 1bDD is illustrated in the BER results for 2bDD GMSK shown in figure 7.31. It can be seen from the figure that the performance of the scheme when there is no decision feedback equalisation is sensitive to




(a) $B_tT = 0.25$ (b) $B_tT = 0.5$

Figure 7.27: Scatter diagrams at input to decision circuit for 2-Bit differential detection GMSK with decision feedback equalisation.

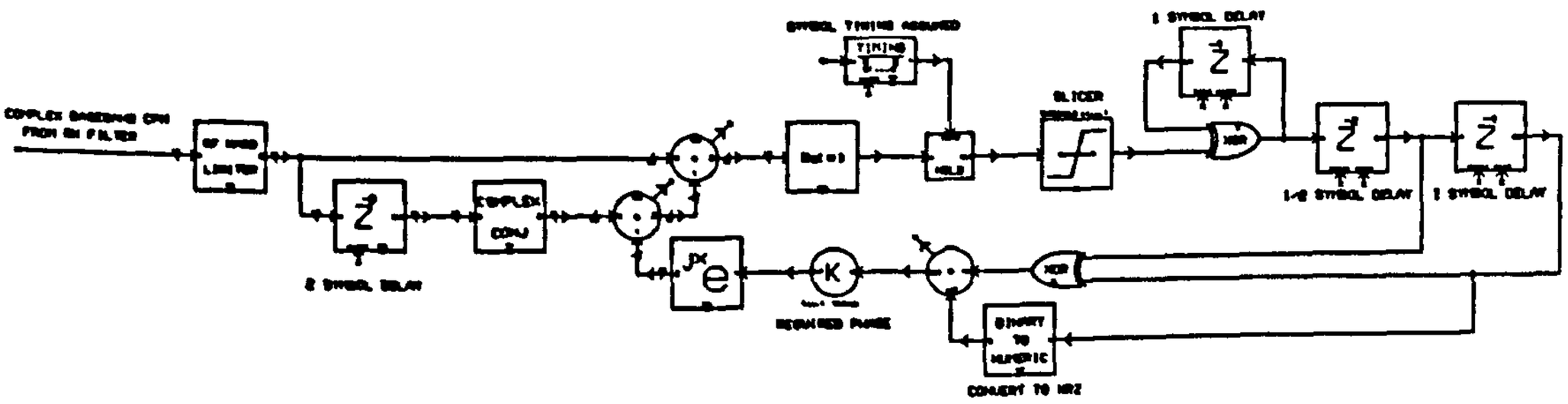



Figure 7.28: SPW implementation of 2 bit differential detection of prefiltered MSK with decision feedback equalisation

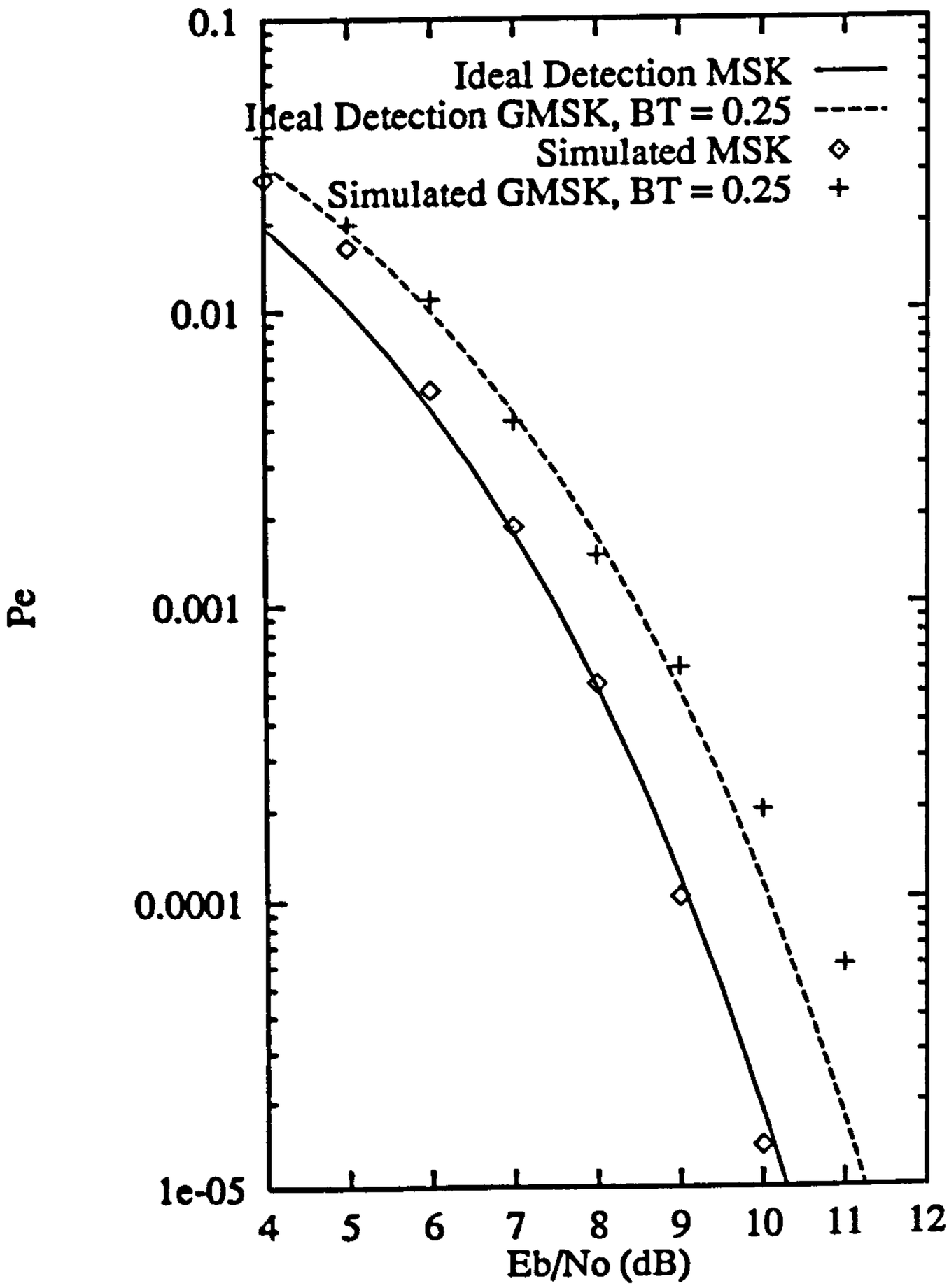



Figure 7.29: BER performance of the coherent synchroniser for MSK and GMSK $B_tT = 0.25$ versus that predicted by theory.

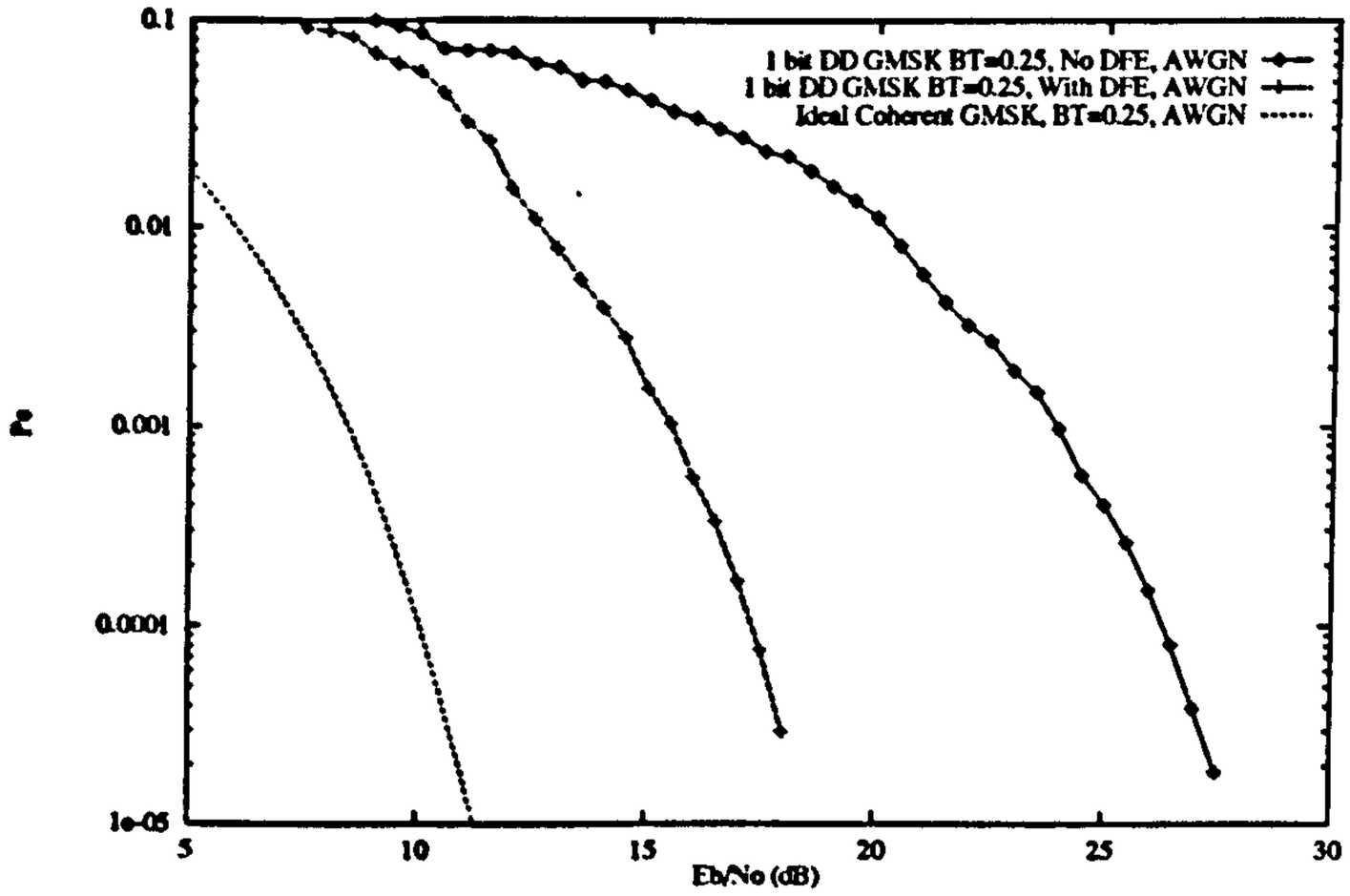



Figure 7.30: BER performance of 1 bit Differential Detection of GMSK with $B_tT = 0.25$ with and without DFE. Also plotted are the results for the coherent synchroniser.

the decision threshold. This is because of the asymmetry in the eye-diagram as discussed previously.

Closed form solutions to the bit error rate performance of some modulation schemes in the land mobile radio channel are available and allow us to compare the SPW models with the theory. Analytical expressions for the bit-error rate for MSK in the narrowband fading channel with cochannel interference and 1 bit differential and limiter-discriminator detection have been derived in [55]. Comparison between the theory and the SPW simulations are shown in figure 7.32 for 1 bit differential detection and figure 7.33 for limiter-discriminator detection.

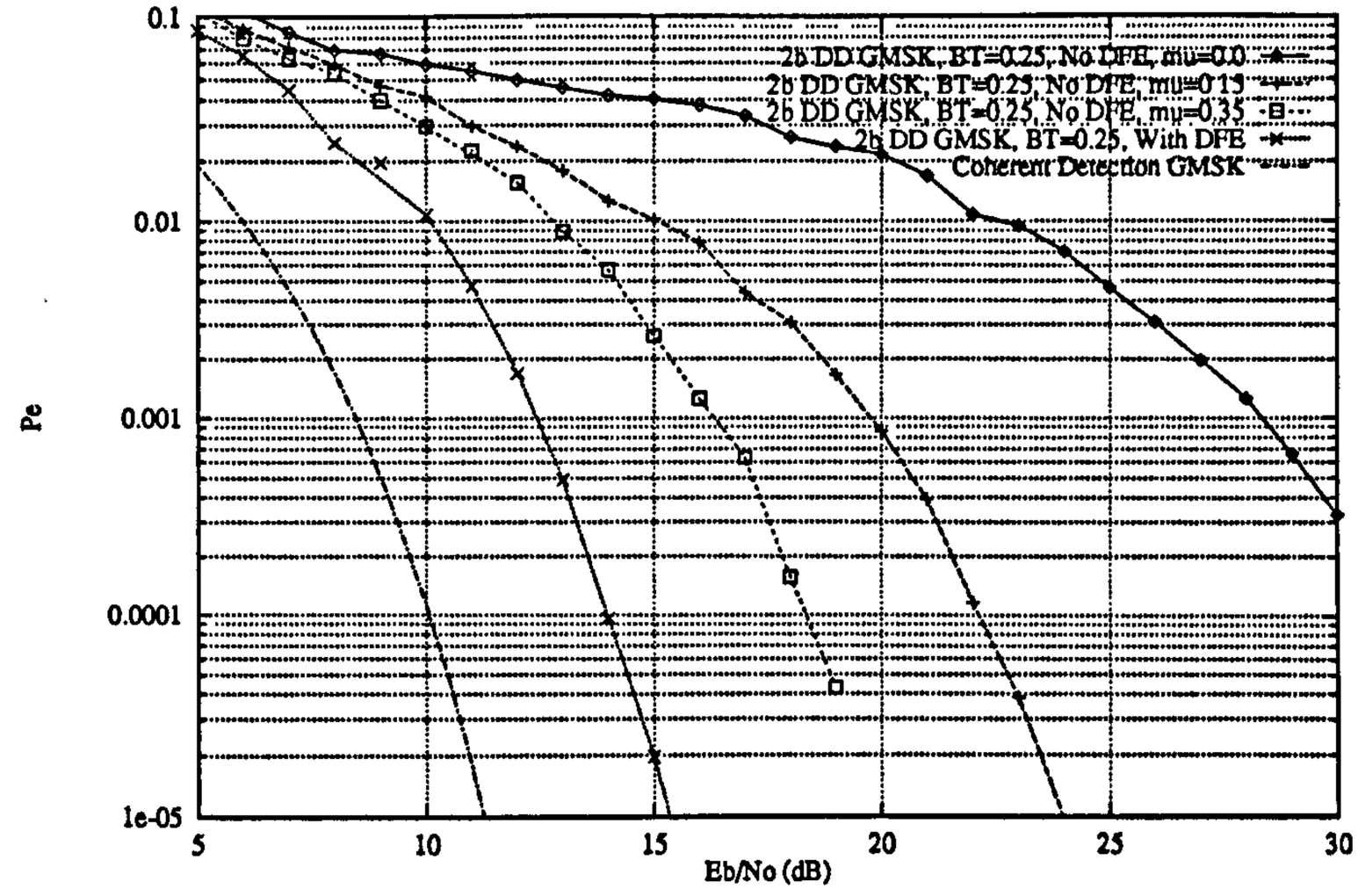



Figure 7.31: BER performance of 2 bit Differential Detection of GMSK with $B_tT = 0.25$ with and without DFE. Also plotted are the results for the coherent synchroniser. The results for where there is no DFE show the decision threshold a parameter.

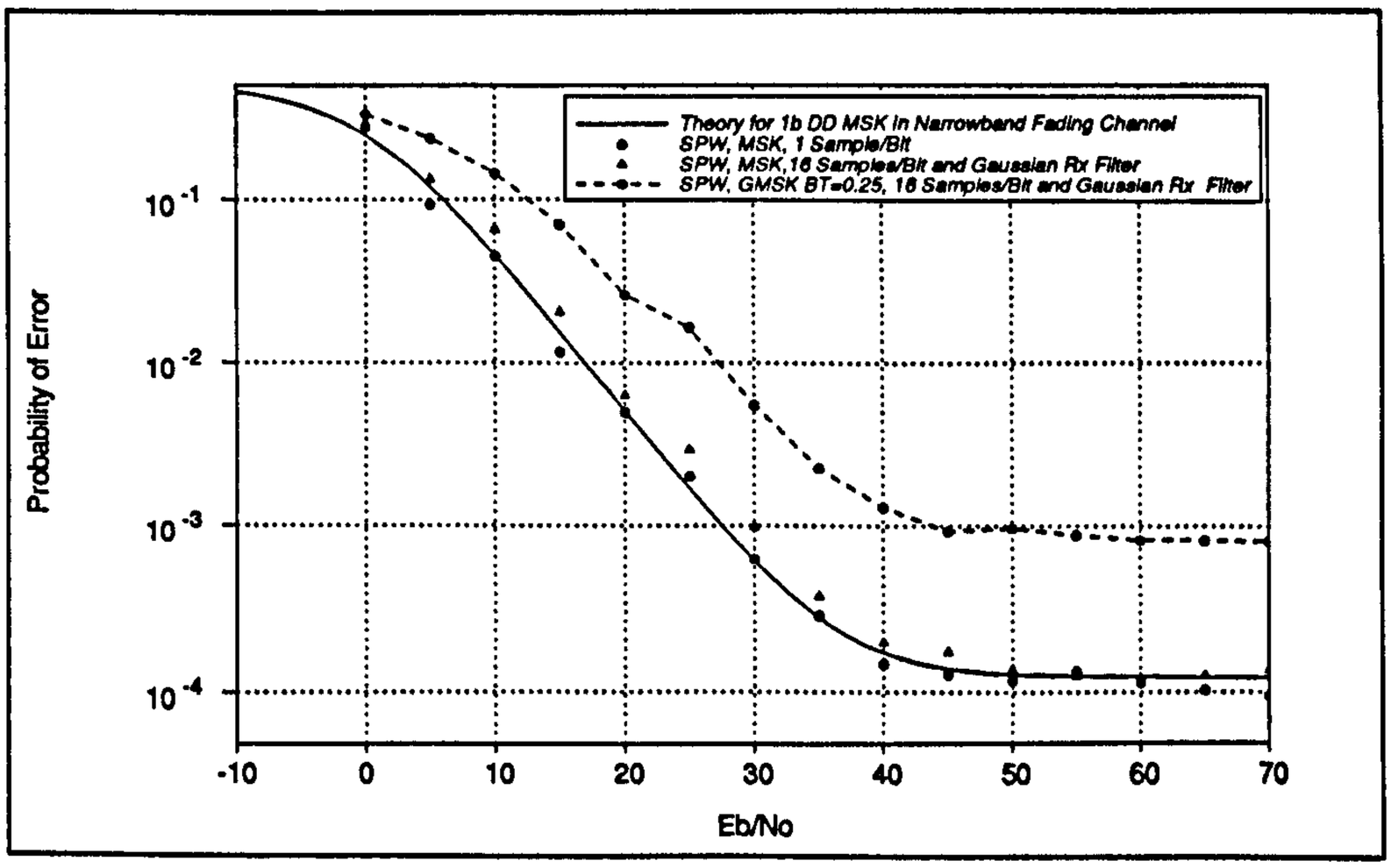



Figure 7.32: SPW Simulation Results for 1b DD MSK and GMSK in narrowband fading, $f_DT_b = 0.005$.

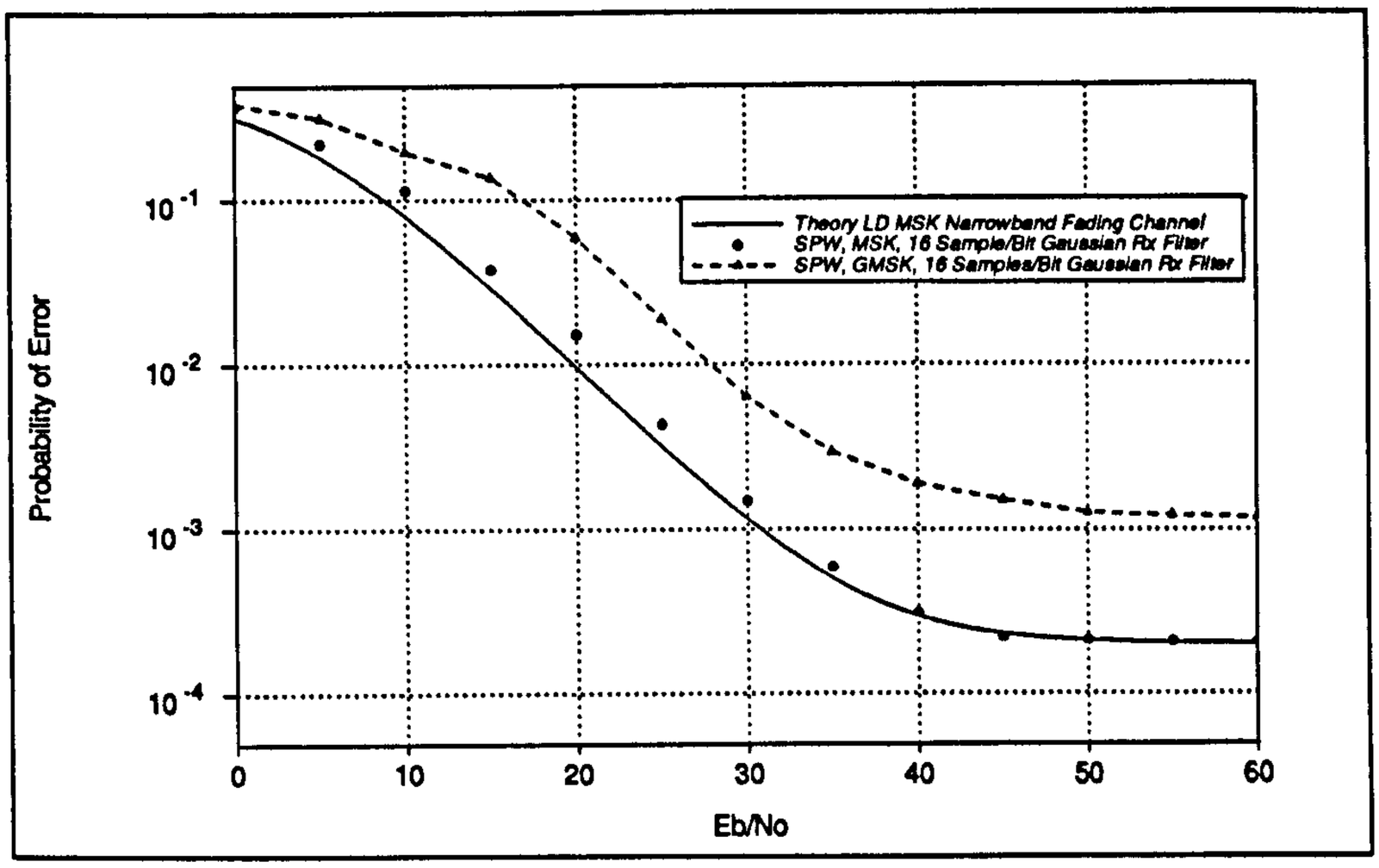



Figure 7.33: SPW Simulation Results for LD MSK and GMSK in narrowband fading, $f_D T_b = 0.01$.

# Chapter 8

# Receiver Integration

## 8.1 Introduction

In this chapter we pull together some of the ideas on synchronisation in sampled receivers developed in previous chapters. Specifically, we demonstrate a joint carrier and clock recovery system for MSK-type waveforms; how symbol timing recovery can be incorporated in a complex baseband differential phase detector for GMSK which also utilises decision feedback equalisation; and finally we demonstrated a system for acquiring frequency, timing and phase for $M$-PSK type modulation schemes and give results of its acquisition performance for large values of frequency offset and timing drift.

## 8.2 Joint Carrier and Clock Recovery for MSK-Type Waveforms

A system which is an extension of the system illustrated in figure 7.17 but now derives the clock and the carrier synchronisation according to the method proposed in [73] is shown in figure 8.1.

This synchroniser arises because of the synchronisation information that is available in the square of an MSK-type waveform. This is illustrated in figures 8.2 for MSK.

It seen from these figures that the result of squaring the input MSK-type waveform is to produce tones at $2\,f_c + R_b/2$ and $2\,f_c - R_b/2$, where $f_c$ is the carrier frequency and $R_b$ is the baud rate. These tones are then tracked by two PLLs with quiescent frequencies set to these values. If the output from the PLLs are mixed together then the double frequency component will be a clean tone centred at $4\,f_c$ and the difference frequency will be at $R_b$. So the recovered carrier can be derived by applying the double frequency to a divide-by-four PLL and the clock is available directly. The clock has to be divided down by a factor of two in order to clock the orthogonal arms of the detector. The configuration shown is that of a feed-forward loop.

There is an opportunity of the synchroniser to *false-lock* to the wrong harmonic in the squared signal. It has been found experimentally that it will false-lock for relatively small

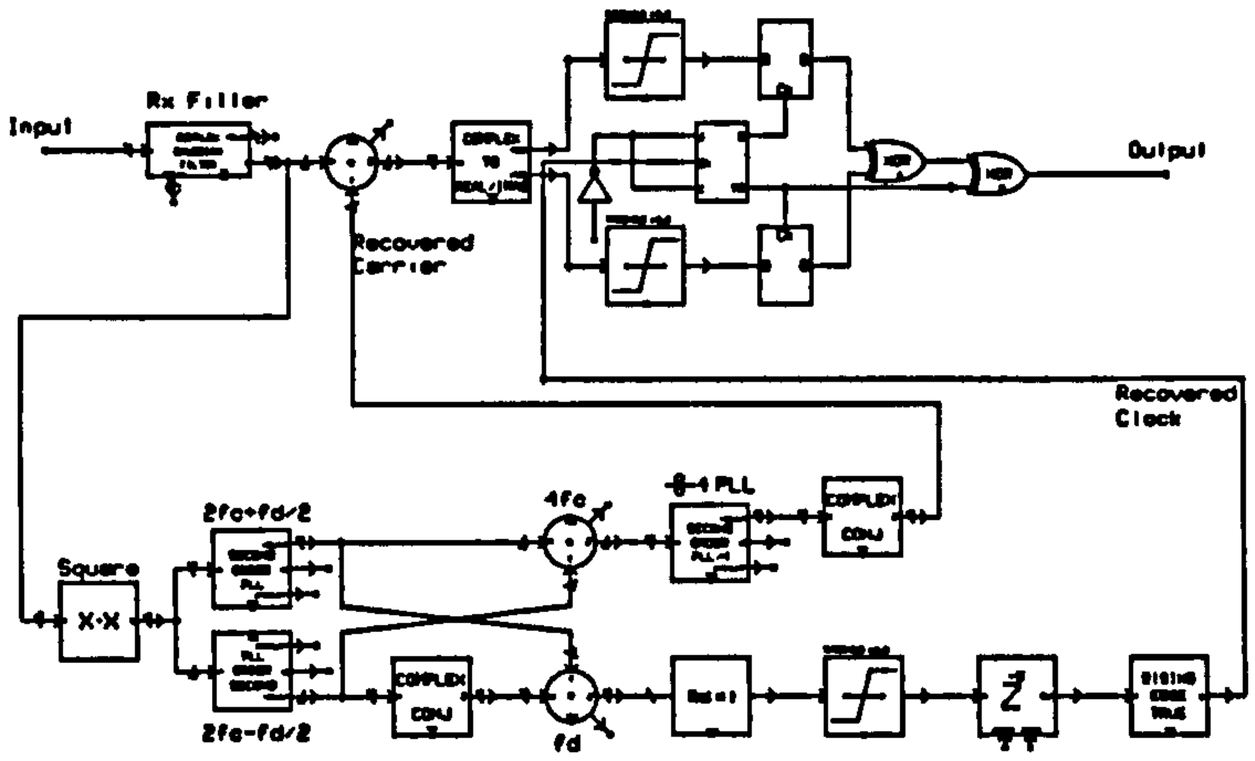


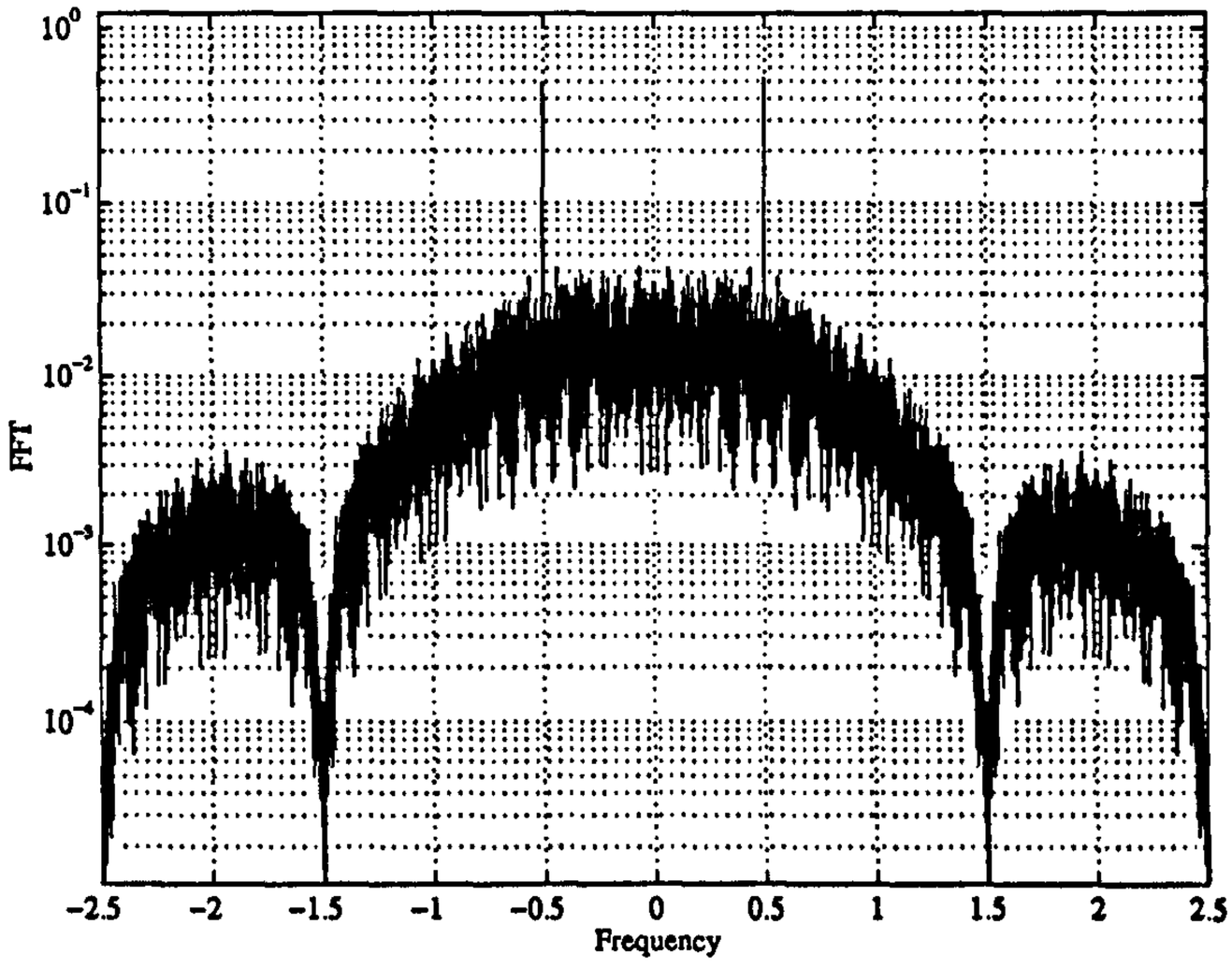

Figure 8.1: Synchroniser for MSK-type signals

values of input frequency offset. It seems that this synchroniser may be limited to those situations where the offset frequency, $f_d$, and the symbol period, $T = 1/R_b$ are such that $f_d T << 0.005$. This would preclude the use of this synchroniser for all but the slowest fading rates.

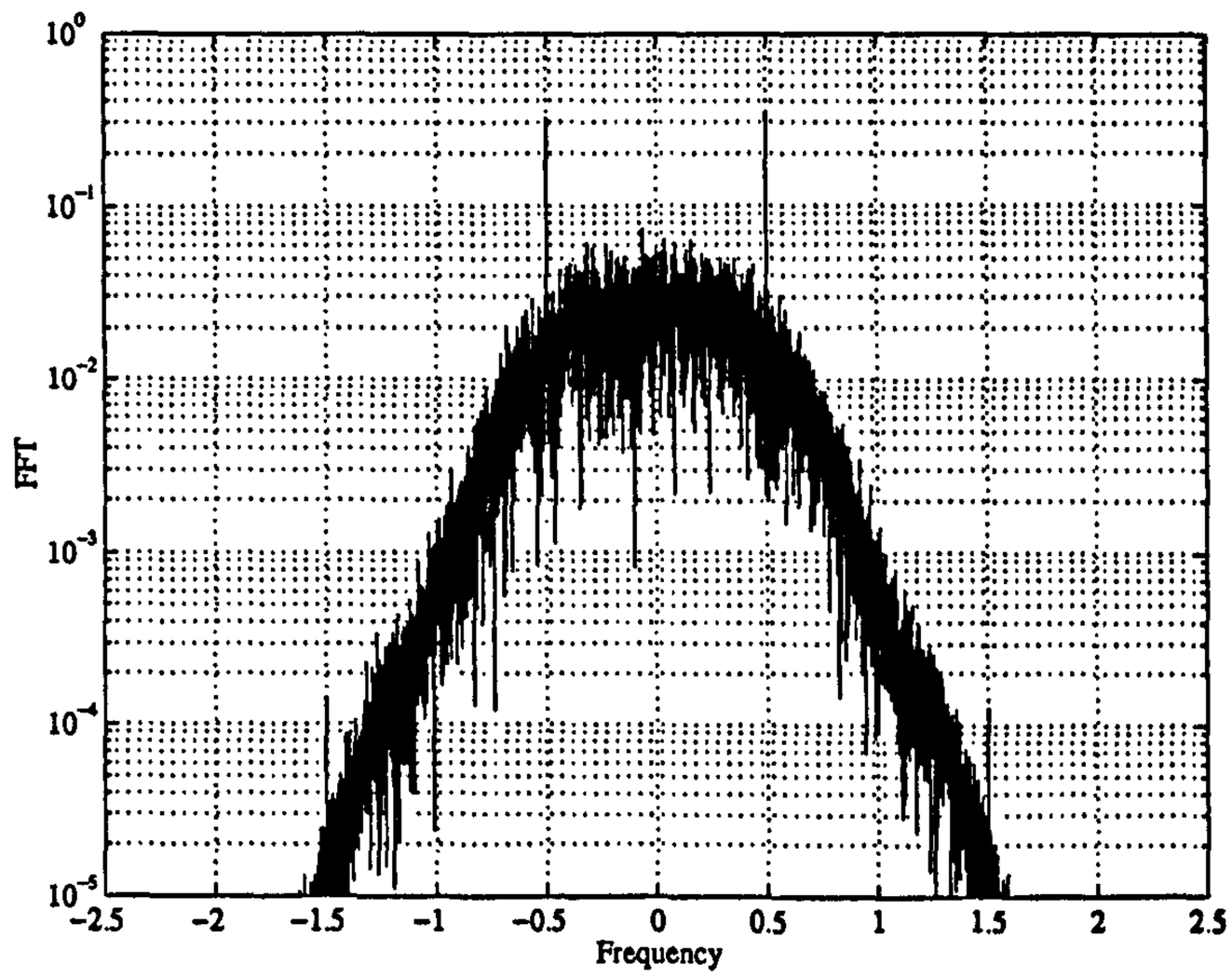
There are coherent architectures that may be of more use that the incoherent schemes in the fast fading channel but these involve complex signal processing [74]. Other configurations for synchronisers for MSK-type waveforms can be found in [75].

## 8.3 Integration of Symbol Timing Recovery, Differential Phase Detection & Decision Feedback Equalisation of GMSK

Figure 8.3 shows how one can incorporate symbol timing with a complex baseband implementation of differentially detected GMSK with decision feedback equalisation. As we are operating at complex baseband we can separate out the functions of two bit differential detection and the decision feedback equalisation as is shown in the figure. It was found that because of the erratic nature of the equalised waveform it was necessary to have the equaliser and the timing recovery operating in parallel on the same input waveform. The system operates at 4 samples/symbol. Note that the timing recovery implementation here is rather unusual. The timing error detection is accomplished by operating on a real signal, which as we saw previously had a asymmetrical eye opening for two bit differential detection of GMSK, so the data slicers shown have a threshold which is optimally non-zero. Although the timing error detector operates on a real signal the timing error correction is complex, that is the Farrow interpolator operates on both the real and the imaginary components of the signal.




(a) MSK




(b) GMSK, $B_tT = 0.25$

Figure 8.2: Spectrum of the square of an MSK and a GMSK signal.

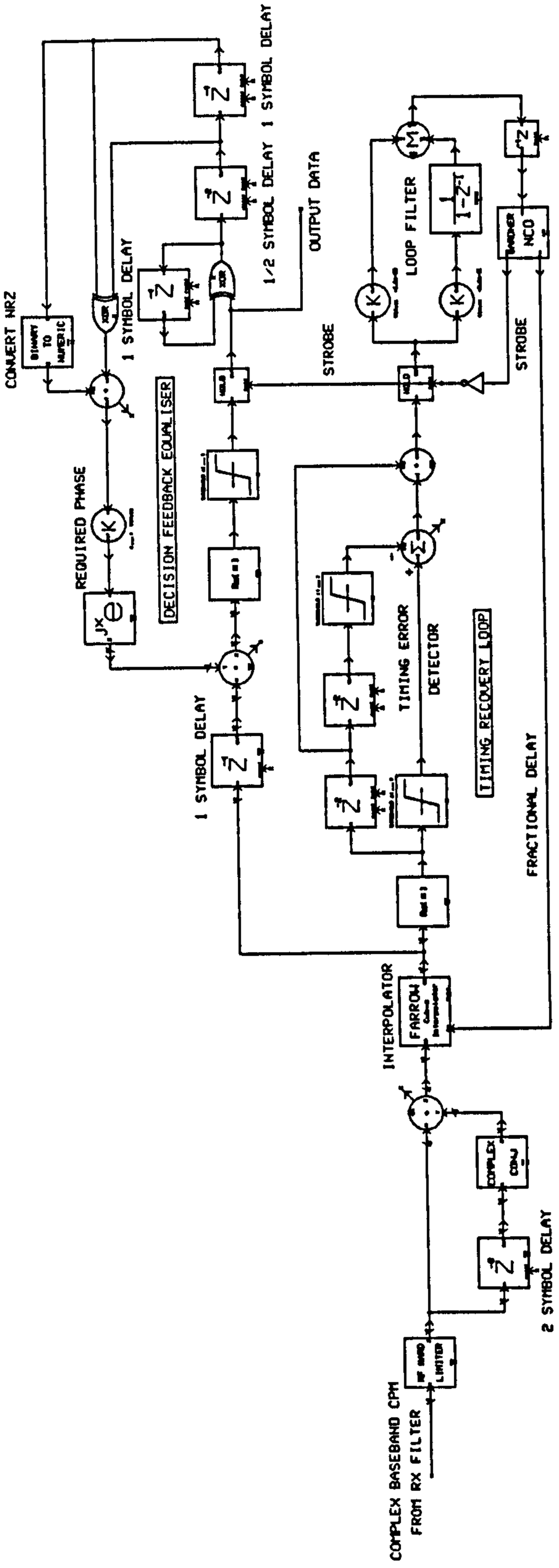



Figure 8.3: SPW implementation of 2 bit differential detection of prefiltered MSK with decision feedback equalisation and symbol synchronisation

## 8.4 Timing, Frequency & Phase Recovery for Non-offset M-PSK Data

We are now in a position to integrate a number of system components together. Figure 8.4 shows a "stripped-down" schematic-capture of the symbol-timing, carrier-frequency and carrier-phase recovery system. The frequency error detector is the ML-FED, the symbol-timing recovery is the NDA Garder algorithm operating at two samples per symbol and the carrier phase recovery is decision-directed.

There are a large number of different system configurations that could be considered apart from that in the figure. The system shows the frequency and phase locked loops combined, that is, the two error signals are added together before being applied to the VCO. One could, however, have the two loops completely separate, that is, in a serial configuration. One could also have feedback carrier frequency recovery in conjunction with feedforward carrier phase. The frequency locked loop is shown as a first order loop[1] whereas the phase locked loop is second order. Extensive research still needs to be done on the relative settings of loop bandwidths for frequency and phase. If the loop bandwidth of the frequency locked loop is too large then the resulting phase noise swamps the PLL. This results in cycle-slipping, tantamount to rotation of the signal constellation and hence an increase in the symbol error rate of the system. A study of cycle-slipping in carrier-recovery loops conditioned on the AFC circuit bandwidth does not appear to have been done and may prove fruitful.

The ML-FED is implemented in its complex-baseband form. The output from the signal matched filter is driven directly to the symbol timing recovery loop via the cubic Farrow interpolator. Note that there are two such interpolators in the system on the outputs of the signal and frequency matched filters. It has been found that this is necessary as if there is only one interpolator prior to filtering then the delay through the filter results in the data strobe and the fractional-delay being asynchronous. One expects the local NCO to be asynchronous with the received complex PAM data stream but the delivered strobe and fractional delay must be synchronous. If this is not the case then it has been found by simulation that excessive timing jitter results. Note that the generated timing strobe is applied simultaneously to the phase, frequency and, of course, timing recovery loops.

The symbol timing recovery loop is shown as the 2pNDA Gardner algorithm. This allows timing recovery on a rotating signal constellation. In the following results we see that timing is actually acquired before frequency – this would not be possible if we had used decision-directed timing recovery. Phase will always be acquired last; by definition one must already know the symbol timing and carrier frequency, and so decision-directed phase recovery gives the best performance.

Figure 8.5 shows the acquisition performance of this system for a large initial frequency error of $f = 1/T$. There is also a symbol timing error on the symbol timing recovery loop – hence the ramping of the NCO phase. Carrier phase acquires shortly after carrier frequency. Also shown in the figure is the real part of the synchronised complex PAM data stream.

[1] First order in frequency, that is. There are still two *phase* integrators in the loop.

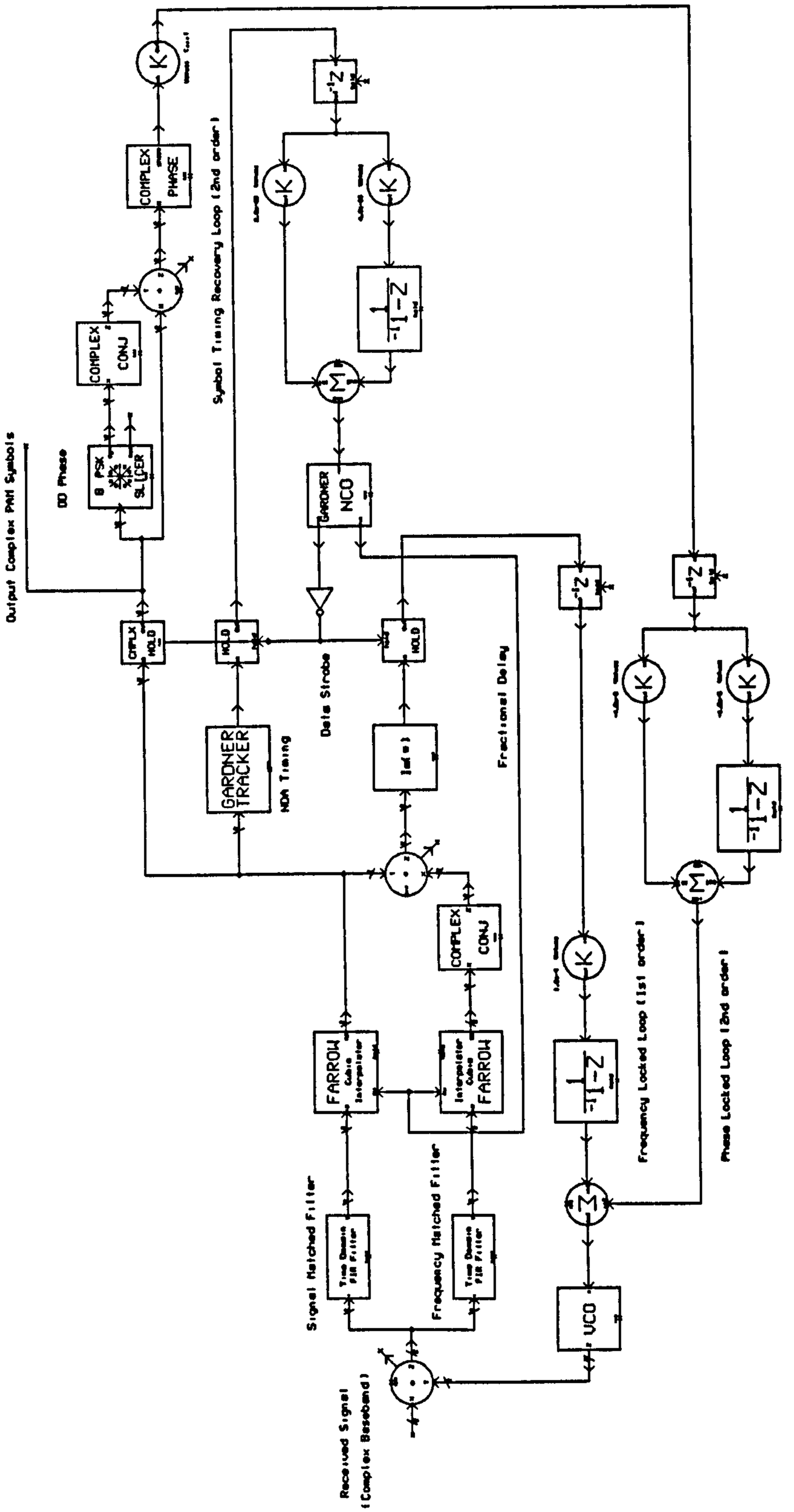



Figure 8.4: SPW implementation of ML frequency, DD phase & NDA timing recovery system.

Note that when the frequency offset is large the signal level is reduced because the signal is actually outside of the receiver signal matched filter. This effect is perhaps better illustrated by figure 8.6 which shows the scatter diagram of the signal during different time frames of 1000 symbols. Note that the loop bandwidth has been set to be purposely small in order to illustrate the loop dynamics.

For this system the filter coefficients specifying the loop bandwidth were chosen arbitrarily as, for the frequency locked loop,

$$C_1 = 1.0e-04$$

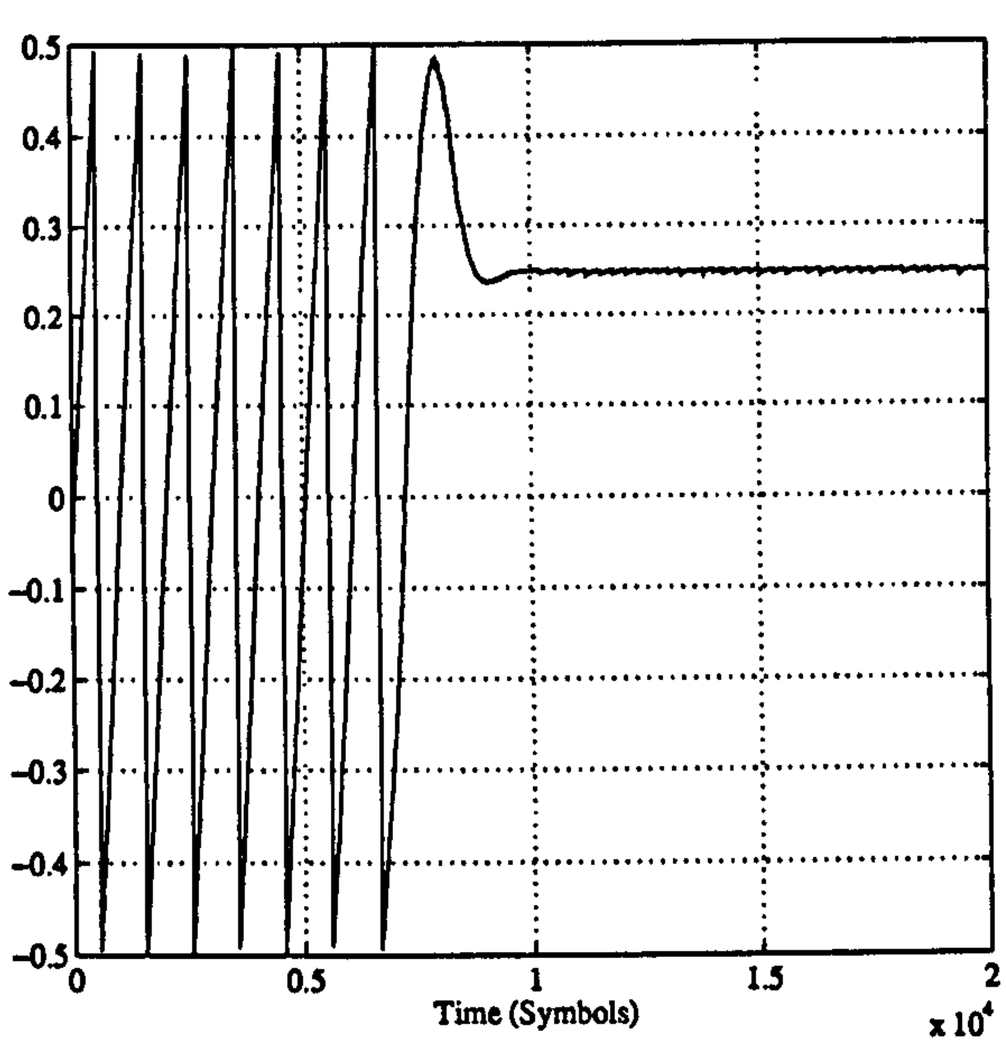
for the phase locked loop,

$$C_1 = 1.0e-03 \text{ and } C_2 = 1.0e-05$$

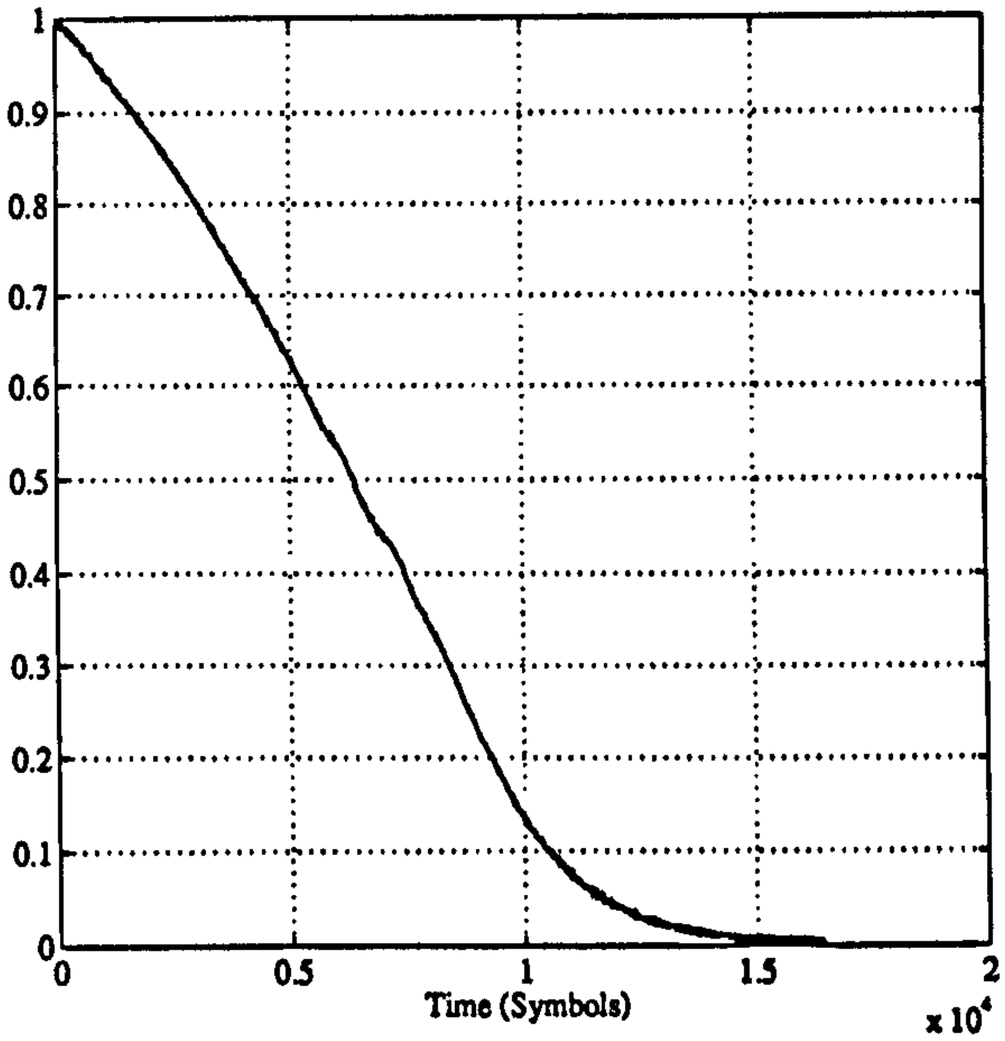
and for the timing recovery loop,

$$C_1 = 2.0e-03 \text{ and } C_2 = 4.0e-05$$

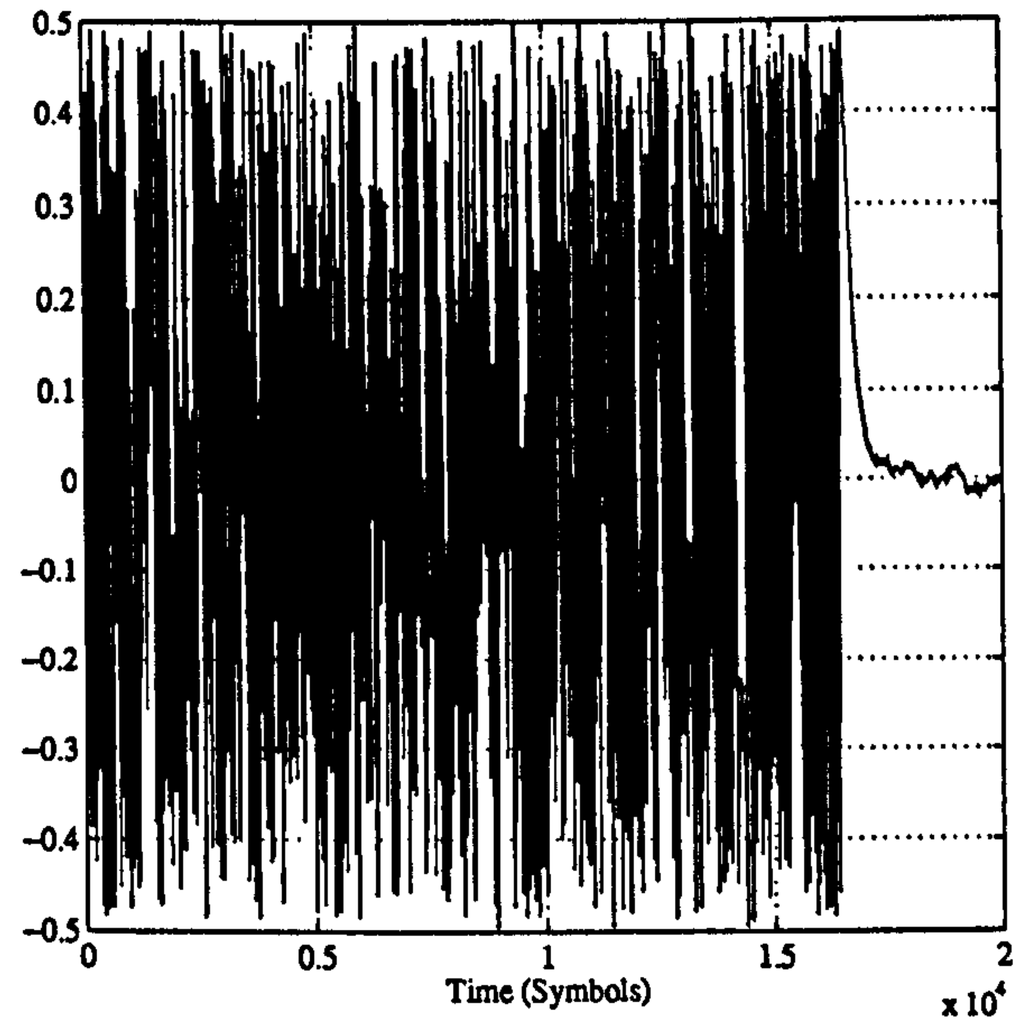
The actual loop noise bandwidths could also be evaluated by using the knowledge that $\alpha = 0.8$. The number of taps used for the filter implementation is $N = 31$.



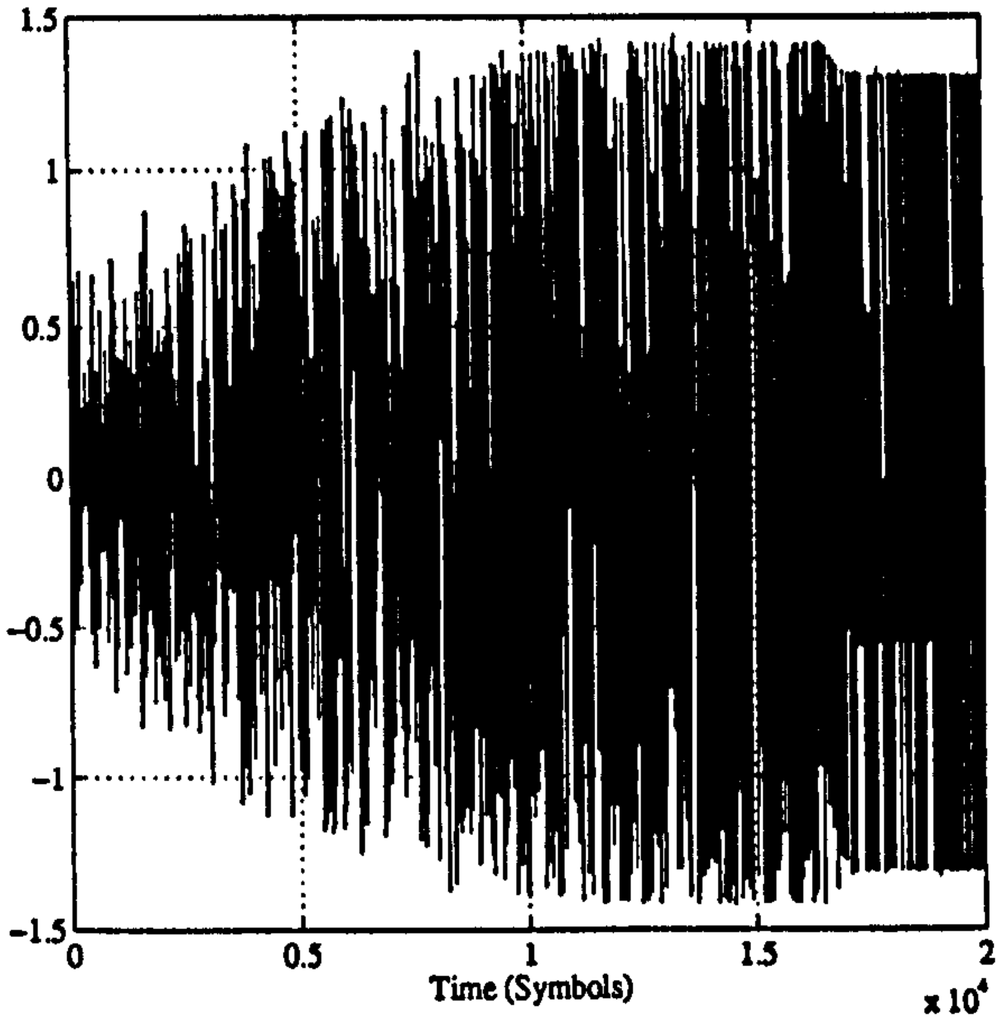

(a) Timing Phase




(b) Carrier Frequency Error




(c) Carrier Phase Error




(d) Output M-PSK Symbols (Real Part)

Figure 8.5: Acquisition performance of the timing, frequency & phase recovery system. Data is raised-cosine filtered 8-PSK with $\alpha = 0.8$.

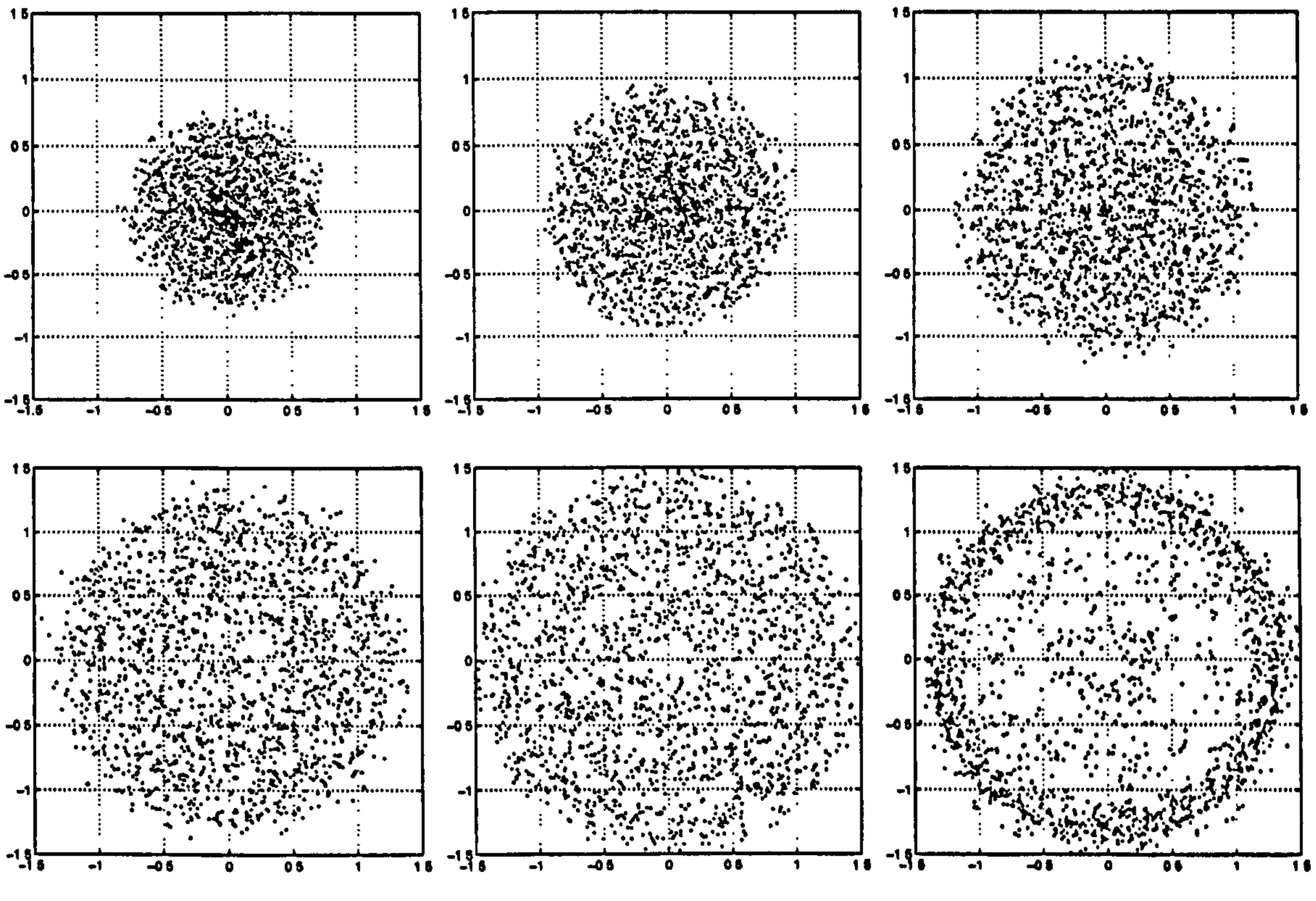


Figure 8.6: Acquisition performance of the timing, frequency & phase recovery system. Data is raised-cosine filtered 8-PSK with $\alpha = 0.8$. There are 1000 symbols per frame but note the loop bandwidth is set to be small in order to illustrate the loop dynamics

# Chapter 9

# Conclusions

A number of techniques for the synchronisation of carrier frequency, symbol timing and carrier phase in sampled receivers have been developed and analysed as the main part of this thesis. Auxiliary to this the narrowband modulation schemes to which these synchronisation methods are appropriate have been quantified; the narrowband fading channel has been modelled and description of techniques specific to the simulation of communication systems given. A summary of the main facets of this research and indications as to the original contribution of this thesis are given as follows.

The characteristics of a number of modulation schemes that are commonly used for narrowband land-mobile and satellite communication systems were described. In particular, the phase trajectory of GMSK and the pulse shapes for raised cosine filtered $M$-PSK, and the derivatives thereof, were given as these schemes are to be used with the synchronisers later.

A novel technique is then described for modelling the narrowband fading channel appropriate to radio propagation in an urban environment. This shows that a good approximation to the ideal noise correlation shaping filter is a windowed-Bessel function, in much the same way that a windowed-sinc function is a good approximation to the ideal lowpass filter. The taps thus derived can then be incorporated into a multirate polyphase filter to achieve a highly correlated process with a specific autocorrelation and power spectral density profile. This work has been published in [43]. However, it is pointed out for the first time that it is not always necessary to model the *precise* detail of the power spectral density function when modelling the narrowband fading channel, especially when the signal thus produced is being up-sampled by a large factor. One need only ensure that the rms value of the frequency is the same as would be produced by a filter with the theoretically ideal power spectral density. This is demonstrated with a high order butter-worth filter.

Next we examine frequency synchronisation in sampled receivers. We look at a number of techniques and show that some of these techniques are indeed equivalent. We compare two schemes for frequency error detection, namely, the Balanced Quadricorrelator (BQ) and a Maximum-Likelihood (ML) method. The latter also requires timing, for minimum jitter operation, which is treated separately later on. The design procedure for these FEDs is described and examples given for different values of excess bandwidth. We derive the open-loop S-curve characteristics for these FEDs and illustrate how these can be used to derive equiva-

lent small-signal phase-locked loop type structures amenable to standard analysis techniques. This approach may be implicit in the research undertaken by others but it is clarified here. We note for the first time that although the two loops have the same acquisition time for small values of frequency offset, when setting the loop bandwidths appropriately, for frequency offsets greater than the excess bandwidth the BQ system will acquire faster than the ML system.

A study of symbol synchronisation for non-offset $M$-PSK type modulation schemes was then undertaken. There are a plethora of schemes in the literature and what we have attempted to do for the first time was to rationalise these schemes according to the number of samples per symbol they require for operation and the form of nonlinearity they use (if any). We re-wrote these timing-error detector algorithms in the complex domain, which although obvious was not done in the original papers, and when in this form properties such as carrier phase invariance becomes obvious. Also by looking at timing-recovery as a complex operation upon a complex signal we conjectured that there is an alternative to decision directed and non-aided recovery and we refer to this scheme as being "amplitude directed." This scheme has intermediate performance between non-data aided and fully decision directed but, unlike the decision-directed case, is carrier-phase invariant, and, unlike the non-data aided system, is not prone to hangup as it does not have a sinusoidal S-curve characteristic. The open-loop characteristics for all the schemes are determined analytically and substantiated by simulation. A carrier-phase invariant timing-error detection scheme has also been published in [76].

We next looked at the control and interpolator structures appropriate for asynchronous timing-recovery loops. We considered the polyphase (switched-coefficient) and the polynomial (Farrow) interpolators. The structure of the polyphase interpolator appropriate for timing control in asynchronous loops was published in [77, 78]. We noted for the first time that the use of a polyphase interpolator together with a non-data aided timing error detector can exacerbate the effects of hangup due to the *discretisation* of the detector S-curve characteristic. This effect gets worse for small values of interpolator upsample. This effect was published in [69]. The polynomial interpolators on the other hand allow a continuously variable delay and are not difficult to implement and so appear to be more attractive than polyphase interpolation. Having characterised the structures open-loop we are then able to configure them in a way so that we can determine their closed-loop tracking and acquisition performance, in much the same way as the frequency synchronisers previously. Peculiarities of the loop dynamics are noted here that appear nowhere else.

Next we look at the problem of carrier synchronisation in sampled receivers. We give a review of the differences between first and second order (type I & II) PLL systems and how to implement these in the complex baseband. We discuss the difference between waveform regenerators and tracking structures and emphasise the importance of the latter. We look at a quintessentially digital algorithm for feedforward carrier phase recovery using the Viterbi & Viterbi algorithm and characterise it in terms of its tracking performance. It is interesting to note that this system has an optimum configuration, determined by simulation, but not so far tractable by analysis. The need for *both* differential encoding, which is also

detailed, and "phase-unwrapping" is stressed.

We next look at one and two bit differential detection for (G)MSK. We note that the scheme is severely limited in terms of its performance by the inter-symbol inherent interference. Two bit differential detection offers improved performance over the one bit system but has a rather unusual asymmetrical eye-diagram demanding a non-zero detection threshold. Optimum receivers for these modulation schemes are Maximum-Likelihood Sequence Estimators but these are complex to implement. We illustrated a method, proposed by others, of reducing the inherent inter-symbol interference by decision feedback equalisation. Results are given for these noncoherent schemes in both the AWGN and the flat fading channels. This work is also published in [79].

Finally, we illustrate how various synchroniser sub-systems may be integrated together. We demonstrate a joint carrier and clock recovery system for MSK type waveforms, proposed elsewhere. We then describe two new sampled receiver structures; a system that incorporates symbol timing recovery with two bit differential detection of GMSK with decision feedback equalisation and a system for bandlimited $M$-PSK that will acquire frequency, with an offset of the baud rate, symbol timing and carrier phase.

# Bibliography


[1] Louis L. Scharf. *Statistical Signal Processing.* Addison-Wesley, 1991.

[2] H.L. Van Trees. *Detection, Estimation and Modulation Theory.* Wiley, 1968.

[3] John G. Proakis. *Introduction to Digital Signal Processing.* Macmillan, 1988.

[4] L. E. Franks. Carrier and Bit Synchronization in Data Communication — A Tutorial Review. *IEEE Trans. Comm.*, 28(8):1107–1120, August 1980.

[5] Kurt H. Mueller and Markus Müller. Timing Recovery in Digital Synchronous Data Receivers. *IEEE Trans. Comm.*, 14:516–531, May 1976.

[6] William C. Lindsey. *Telecommunication Systems Engineering.* Prentice-Hall, 1973.

[7] Hisashi Kobayashi. Simultaneous Adaptive Estimation and Decision Algorithm for Carrier Modulated Data Transmission Systems. *IEEE Trans. Comm.*, 19(3):268–280, June 1971.

[8] Floyd M. Gardner. A BPSK/QPSK Timing-Error Detector for Sampled Receivers. *IEEE Trans. Comm.*, 34(5):423–429, May 1986.

[9] Floyd M. Gardner. Demodulator Reference Recovery Techniques Suited for Digital Implementation. *European Space Agency Final Report ESTEC Contract 6847/86/NL/DG*, 1988.

[10] Floyd M. Gardner. Clock and carrier synchronisation : prefilter and antihang-up investigations. *European Space Agency Report ESA CR-984*, 1977.

[11] Floyd M. Gardner. Timing adjustment via interpolation in digital demodulators. *European Space Agency Final Report : Part I ESTEC Contract 8022/88/NL/DG*, 1990.

[12] Floyd M. Gardner. Frequency Detectors for Digital Demodulators via Maximum Likelihood Derivation. *European Space Agency Final Report : Part II ESTEC Contract 8022/88/NL/DG*, 1990.

[13] Floyd M. Gardner. Multipath Link and Modem Design for Satellite-Mobile Communications. *European Space Agency Final Technical Report : ESTEC Contract 5146/82/NL/GM(SC)*, 1983.

[14] C.W. Farrow. A continuously variable digital delay element. In *IEEE International Symposium on Circuits and Systems*, pages 2641–2645, June 6–9 1988. Espoo, Finland.

[15] Floyd M. Gardner. Interpolation in digital modems—part I : fundamentals. *IEEE Trans. Comm.*, 41(3):501–507, March 1993.

[16] F.M.Gardner L. Erup and R.A.Harris. Interpolation in digital modems—part II : implementation and performance. *IEEE Trans. Comm.*, 41(6):998–1008, June 1993.

[17] David G. Messerschmitt. Frequency Detectors for PLL Acquisition in Timing and Carrier Recovery. *IEEE Trans. Comm.*, 27(9):1288–1295, September 1979.

[18] Thomas Alberty and Volker Hespelt. A New Pattern Jitter Free Frequency Error Detector. *IEEE Trans. Comm.*, 37(2):159–163, February 1989.

[19] Aldo N. D'Andrea and Umberto Mengali. Performance of a Quadricorrelator Driven by Modulated Signals. *IEEE Trans. Comm.*, 38(11):1952–1957, November 1990.

[20] Aldo N. D'Andrea and Umberto Mengali. Design of Quadricorrelators for Automatic Frequency Control Systems. *IEEE Trans. Comm.*, 41(6):988–997, June 1993.

[21] Aldo N. D'Andrea and Umberto Mengali. Noise Performance of Two Frequency-Error Detectors Derived from Maximum Likelihood Estimation Methods. *IEEE Trans. Comm.*, 42(2/3/4):793–802, February/March/April 1994.

[22] Marc Moeneclaey. Overview of Digital Algorithms for Carrier Frequency Synchronization. In *Fourth International Workshop on Digital Signal Processing Techniques Applied to Space Communications*. European Space Agency (ESA), 1994.

[23] Marc Moeneclaey. A Simple Lower Bound on the Linearized Performance of Practical Symbol Synchronizers. *IEEE Trans. Comm.*, 31(9):1029–1032, September 1983.

[24] Marc Moeneclaey. A Fundamental Lower Bound on the Performance of Practical Joint Carrier and Bit Synchronizers. *IEEE Trans. Comm.*, 32(9):1007–1012, September 1984.

[25] Marc Moeneclaey. Twon Maximum Likelihood Symbol Synchronizers with Superior Tracking Performance. *IEEE Trans. Comm.*, 32(11):1178–1185, November 1984.

[26] Marc Moeneclaey. A Comparison of Twon Types of Symbol Synchronizers for Which Self-Noise is Absent. *IEEE Trans. Comm.*, 31(3):329–334, March 1983.

[27] Marc Moeneclaey. Synchronisation Problems in PAM Systems. *IEEE Trans. Comm.*, 28(8):1130–1136, August 1980.

[28] Aldo N. D'Andrea and Umberto Mengali. The Modified Cramer-Rao Bound and its Application to Synchronization Problems. *IEEE Trans. Comm.*, 42(2/3/4):1391–1399, February/March/April 1994.

[29] M. Moeneclaey T. Jesupret and G. Ascheid. Digital Demodulator Synchronization – Performance Analysis. Technical Report ESTEC 8437/89/NL/RE, European Space Agency, June, 1991.

[30] John G. Proakis. *Digital Communications.* McGraw-Hill, 1989.

[31] Michel C. Jeruchim. Techniques for estimating the bit error rate in the simulation of digital communication systems. *IEEE Jnl. Selected Areas in Comms.*, pages 153–170, January 1984.

[32] Michel C. Jeruchim. *Simulation of Communication Systems.* Plenum, 1992.

[33] Andrew J. Viterbi. Implications of mobile cellular CDMA. *IEEE Comms. Mag.*, pages 38—41, December 1992.

[34] Kamilo Feher. MODEMS for emerging digital cellular-mobile radio system. *IEEE Trans. Vehicular Tech.*, 40(2):355–365, May 1991.

[35] Price. A communications technique for fading multipath channels. *PROCEEDINGS OF THE INSTITUTE OF RADIO ENGINEERS*, pages 555—570, March 1958.

[36] Turin. Introduction to spread spectrum antimultipath techniques and their application to urban digital radio. *PROCEEDINGS OF THE IEEE*, 68(3):328—353, March 1980.

[37] J. P. McGeehan. Phase-locked transparent tone in band (TTIB). *IEEE Trans. Comm.*, 32(1):81—87, January 1984.

[38] J. P. McGeehan. Theoretical and experimental investigation of feedforward signal regeneration as a means of combatting multipath propogation effects in pilot-based SSB mobile-radio systems. *IEEE Trans. Vehicular Tech.*, 32(1):106—120, February 1983.

[39] J. P. McGeehan. Coherent demodulation of pilot-tone single sideband with phase-locked TTIB/FFSR processing. *IEE Proceedings*, 132(5), August 1985.

[40] Kazuaki Murota. GMSK modulation for digital mobile radio telephony. *IEEE Trans. Comm.*, 29(7):1044–1050, July 1981.

[41] Tor Aulin. Continuous phase modulation — part I : Full response signalling. *IEEE Trans. Comm.*, 29(3):196–209, March 1981.

[42] Tor Aulin. Continuous phase modulation — part II : Partial response signalling. *IEEE Trans. Comm.*, 29(3):210–225, March 1981.

[43] D. Verdin and T.C. Tozer. Generating a Fading Process for the Simulation of Land-Mobile Radio Communications. *Elec. Lett.*, 29(23):2011–2012, November 1993.

[44] Michael J. Gans. A power-spectral theory of propagation in the mobile-radio environment. *IEEE Trans. Vehicular Tech.*, 21(1):27–38, January 1972.

[45] William H. Press et al. *Numerical Recipes in C.* Cambridge University Press, second edition, 1992.

[46] Krzysztof Wesolowski. Computer generation of a slowly varying pseudorandom process. *IEE Proceedings*, 130(4):314–316, June 1983.

[47] J. D. Parsons. *Mobile Communication Systems.* Blackie, 1989.

[48] H. Brehm. Experimental results on non-fading time intervals in mobile radio by means of software simulation. In *PROCEEDINGS 1989 URSI INTERNATIONAL SYMPOSIUM ON SIGNALS, SYSTEMS, AND ELECTRONICS*, pages 385–388, 1989.

[49] W. C. Jakes. *Microwave mobile communications.* Wiley, 1974. Book now out of print in UK.

[50] Fumiyuki Adachi. BER Performance of QDPSK with Postdetection Diversity Reception in Mobile Radio Channels. *IEEE Trans. Vehicular Tech.*, 40(1):237–249, February 1991.

[51] Said M. Elnoubi. Analysis of GMSK with Differential Detection in Land Mobile Radio Channels. In *36th IEEE VEHICULAR TECHNOLOGY CONFERENCE*, pages 90–95. IEEE, 1986.

[52] Said M. Elnoubi. Analysis of GMSK with Two-Bit Differential Detection in Land Mobile Radio Channels. *IEEE Trans. Vehicular Tech.*, 35(2):237–240, February 1987.

[53] Marvin K. Simon. Differential Detection of Gaussian MSK in a Mobile Radio Environment. *IEEE Trans. Vehicular Tech.*, 33(4):307–320, November 1984.

[54] Kenkichi Hirade. Mobile radio communications. In Kamilo Feher, editor, *Advanced Digital communications*, chapter 10, pages 487–572. Prentice-Hall, 1987.

[55] Kenkichi Hirade. Error-Rate Performance of Digital FM with Differential Detection in Land Mobile Radio Channels. *IEEE Trans. Vehicular Tech.*, 28(3):204–212, August 1979.

[56] Said M. Elnoubi. Analysis of GMSK with Discriminator Detection in Land Mobile Radio Channels. *IEEE Trans. Vehicular Tech.*, 35(2):71–76, May 1986.

[57] Hirono. Multilevel Decision Method for Band-Limited Digital FM with Limiter-Discriminator Detection. *IEEE Jnl. Selected Areas in Comms.*, 2(4):498–506, July 1984.

[58] C.Molinari S.Bellini and G. Tartara. Digital Frequency Estimation in Burst Mode QPSK Transmission. *IEEE Trans. Comm.*, 38(7):959–961, July 1990.

[59] Umberto Mengali. Joint Phase and Timing Acquisition in Data-Transmission. *IEEE Trans. Comm.*, 25(10):1174—1185, October 1977.

[60] Bruce W. Char. *Maple V Library Reference Manual.* Springer-Verlag, 1991.

[61] Andrew J. Viterbi and Audrey M. Viterbi. Nonlinear Estimation of PSK-Modulated Carrier Phase with Application to Burst Digital Transmission. *IEEE Trans. Inf. Theory*, 29(4):543–551, July 1983.

[62] Floyd M. Gardner. Demodulator reference recovery techniques suited for digital implementation. *European Space Agency Final Report ESTEC Contract 6847/86/NL/DG*, 1988.

[63] T.Aboulnasr. Characterization of a Symbol Rate Timing Recovery Technique for a 2B1Q Digital Receiver. *IEEE Trans. Comm.*, 42(2/3/4):1409–1414, February/March/April 1994.

[64] Martin Oerder. Derivation of Gardner's Timing-Error Detector from the Maximum-Likelihood Principle. *IEEE Trans. Comm.*, 35(6):684–685, June 1987.

[65] Edward A. Lee. *Digital Communications.* Kluwer Academic Publishers, 1988.

[66] W.G. Cowley L.P. Sabel. Performance Analysis of Decision Directed Symbol Timing Estimators with Decision Errors. *Elec. Lett.*, 30(10):754–756, May 1994.

[67] Floyd M. Gardner. Self-Noise in Synchronizers. *IEEE Trans. Comm.*, 28(8):1159–1163, August 1980.

[68] Jae S. Lim and Alan Victor Oppenheim. *Advanced Topics in Signal Processing.* Prentice Hall, 1988.

[69] D. Verdin and T.C. Tozer. Hangup in Asynchronous Timing Recovery Loops. *Elec. Lett.*, 29(22):1914–1915, October 1993.

[70] Floyd M. Gardner. *Phaselock Techniques.* Wiley, 1979.

[71] William C. Lindsey. A survey of digital phase-locked loops. *Proc. IEEE*, 69(4):410–431, April 1981.

[72] Rudi De Buda. Coherent demodulation of frequency-shift keying with low deviation ratio. *IEEE Trans. Comm.*, 20:420–435, June 1972.

[73] Aldo N. D'Andrea. Carrier phase and clock recovery for continuous phase modulated signals. *IEEE Trans. Comm.*, 35(10):1095–1101, October 1987.

[74] Reinhold Haeb. A systematic approach to carrier recovery and detection of digitally phase modulated signals on fading channels. *IEEE Trans. Comm.*, 37(7):748–754, July 1989.

[75] Steven H. Goode. An open-loop technique for the detection of minimum shift keyed signals. $36^{th}$ *IEEE Vehicular Technology Conference, Dallas, Texas*, pages 90–101, 1986.

[76] D. Verdin and T.C. Tozer. Symbol timing recovery scheme tolerant to carrier phase error. *Elec. Lett.*, 30(2):116–117, January 1994.

[77] D. Verdin and T.C. Tozer. Interpolator Filter Structure For Asynchronous Timing Recovery Loops. *Elec. Lett.*, 29(5):490–492, March 1993.

[78] IEE Colloquium on Communications Simulation and Modelling Techniques, York. *Simulating Asynchronous Timing Recovery Loops Using Multirate Techniques*, 1993.

[79] D. Verdin and T.C. Tozer. Simulation and Modelling of Mobile Communcation Systems using the CAAD Tool SPW. In *Digital Signal Processing*, pages 737–742, 1991.

# Appendix A

# Methodology for Iterating Over Parameters Using SPW

It is difficult to use SPW to directly iterate over parameters. I have developed a number of blocks which will read in parameters at the start of a simulation and write out parameters during termination. Using these blocks in cojunction with CGS and a controlling program it is easy to use SPW as an aid to code development. Simulations that characterise a system can then be bulit up and are controlled by an external controlling program. The following code is for the "write-on-stop" block which writes out the value of the parameter on its input node during the termination section of the code. Note that read and writes are done relative to the CGS_RUN_DIR environment variable if it is set. If this environment variable is not set then file input and output is done relative to the HOME environment variable. This is needed as when one is running the code directly from the BDE then it will read its input parameters and write the output simulation results relative to the users home directory and not deep in the SPW tooldeck directory. The CGS code for the write-on-stop block is as follows

```
/*
 *  Expression file for chip coding
 *
 *  Library: danlib, Function: wrtonstop, Model: symbol(0)A
 *  Tue Feb 22 12:07:30 1994
 *
 *  You may make modifications to this file where there are braces:
 *  @INCLUDE_FILES: {<add include files>}
 *  @INCLUDE_DIRS: {<add include directories>}
 *  @LINK_TO: {<add files to be linked with this one>}
 *  @PARAMETER_DECLARATIONS: {<delete unneeded declarations>}
 *  @STATE_DECLARATIONS: {<add declarations for state variables>}
 *  @BLOCK_DECLARATIONS: {<add declarations for block variables>}
 *  @INITIALIZATION_CODE: @RUN_OUT_CODE: @RUN_IN_CODE:
 *  @TERMINATION_CODE:
 *    {<add appropriate sections of code>}
 */

#include "spwtypes.h"

@FEED_THROUGH_TYPE: ALL_FEED_THROUGH

@INCLUDE_FILES: { "cgsfile.h" }

@INCLUDE_DIRS: { }

@LINK_TO: { }

@INPUT_DECLARATIONS:
STATIC Double I_x;
```

```
@OUTPUT_DECLARATIONS:


@PARAMETER_DECLARATIONS:
{
STATIC Char *P_Filename;
}
@STATE_DECLARATIONS:
{
}
@BLOCK_DECLARATIONS:
{
}
@INITIALIZATION_CODE:
{
}
@RUN_OUT_CODE:
{
}
@TERMINATION_CODE:
{
  {
    FILE * fp;
    char * home;
    char filename[50];

    /* This program interrogates first the CGS_RUN_DIR environment */
    /* variable and then the HOME directory. The same directories are */
    /* used for the file constant block */

    if (getenv("CGS_RUN_DIR")){
      home=(char *)getenv("CGS_RUN_DIR");
    }else if (getenv("HOME")){
      home=(char *)getenv("HOME");
    }else{
      printf(">>Environment variables HOME or CGS_RUN_DIR have to be set\n");
      exit(0);
    }
    sprintf(filename,"%s/%s",home,P_Filename);
    fp=fopen(filename,"a");
    if (!fp) {
      printf(">> Cannot open filename %s for writing\n",filename);
      printf(">> Exiting.....\n");
      exit(0);
    }
    fprintf(fp,"%le\n",I_x);
    fclose(fp);
  }
}
```

Having developed the code under CGS and ported it to a remote workstation we can compile it using a command of the form

```
cc -linclude -o main.exe -O2 main.c lib/cgs.a -lm
```

Then set the environment variable which tells the code where to get its file input and output

```
setenv CGS_RUN_DIR 'pwd'
```

Then a controlling program can be written to iterate over system parameters. The following controlling program is used to iterate over input signal to noise ratio and the output timing error variance is recorded.

# Appendix B

# MATLAB Scripts

## MATLAB Program to Determine Tracking Performance of the Feedforward V & V Carrier Phase Estimator.

```
% Simulation in MATLAB of Feedforward Carrier Phase Estimation of a
% M-PSK Signal in AWGN compared with the Cramer Rao Lower Bound.
%
close all;
clear all;
N=500000;
L=17;
sm=ones(1,L)./L;
theta=pi/4;
M=4;
K=2;
n=0;
for K=0:4,
K
for snrdb=-5:1:15,
    disp(['Es/No=',num2str(snrdb),' dB ']);
    x=randmpsk(N,1,M).*exp(i*theta);
    x=x+awgn(x,snrdb,1,0.5,1);
    x=abs(x).^K.*exp(i*M*atan2(imag(x),real(x)));
    x=fftfilt(sm,x);
    phi=atan2(imag(x(100:N)),real(x(100:N)))./M;
    var(snrdb+6,K+1)=cov(phi);
end;
end;
save data var;
snrdb=-5:1:25;
return;
figure;
set(semilogy(snrdb,var,'*'),'MarkerSize',3);
hold on;
snr=2*(10.^(snrdb./10));
set(semilogy(snrdb,1./(snr.*L)),'LineWidth',1);
grid;
axis('square');
%axis([-10 15 1e-05 10]);
set(gca,'FontName','Times');
xlabel('Signal To Noise (dB)');
ylabel('Phase Variance');
lgd=legend('*','Simulation','-','CRLB');
set(lgd,'Position',[0.6014    0.7649    0.2038    0.1099]);
axes(lgd);
refresh;
```

# MATLAB Program to Determine TED Open-Loop Characteristics

```
% This MATLAB routine is used to determine the open-loop timing-error
% detector S-curves for the 1p-DD, 1p-AD, 2p-DD, 2p-NDA & 2p-DD schemes.
% It generates RC-filtered M-PSK data and then averages over a number of
% ensembles of a given length to determine the detector characteristic.
%
% The user can specify
%   Number of Symbols per Ensemble and Number of Ensembles
%   Modulation Order
%   Sampling Frequency
%   Number of Filter Taps
%   Excess Bandwidth
%   Signal Amplitude
%
% To produce further results this script is easily modified to include the
% effects of carrier phase error and additive white noise. A similar approach
% can also be used to determine the timing-error detector characteristic for
% offset modulation schemes.
%
%
%
% Dan Verdin, Communcations Research Group, University of York.
%
clear all;
close all;
%
kbd=input('Would you like to generate data or process only (g/p) : ','s');
if strcmp(kbd,'g')
    NSymbols=input('Number of Symbols Per Ensemble : ');
    NEns=input('Number of Ensembles : ');
    Beta=input('Filter Roll-Off Factor : ');
M=input('Modulation Order : ');
R=input('Signal Amplitude : ');

Sf=input('Sampling Frequency : ');
NTaps=input('Number of Filter Taps : ');
%
t=linspace(-NTaps/(2*Sf),NTaps/(2*Sf),NTaps);
pr=rcos(t,Beta);
plot(linspace(-NTaps/(2*Sf),NTaps/(2*Sf),NTaps),pr);
if strcmp(input('This ok (y/n) : ','s'),'n') return; end;
%
for n=1:NEns,
    n
    disp('Filtering Data ...');
    zi=upsample(R*randmpsk(NSymbols,1,M),Sf);
    zo=filter(pr,1,zi);

    NPoints=NSymbols*Sf;
    start=(NTaps+1)/2+Sf/2;

    for t=0:Sf,
t
s0=(start+t:Sf:NPoints-Sf-start);
s1=(start+Sf/2+t:Sf:NPoints-Sf/2-start);
s2=(start+Sf+t:Sf:NPoints-start);
% 1pDD
u=real(conj(mpskqntz(zo(s0),M)).*zo(s2)-conj(mpskqntz(zo(s2),M)).*zo(s0));
sc1pDDm(n,t+1)=mean(u);
% 1pAD
u=real(conj(zo(s2)).*zo(s0)./abs(zo(s0))-conj(zo(s2)).*zo(s0)./abs(zo(s2)));
sc1pADm(n,t+1)=mean(u);
% 2pNDA
u=real(conj(zo(s1)).*(zo(s0)-zo(s2)));
sc2pNDAm(n,t+1)=mean(u);
% 2pAD
u=real(conj(zo(s1)).*(zo(s0)./abs(zo(s0))-zo(s2)./abs(zo(s2)) ));
sc2pADm(n,t+1)=mean(u);
```

```
        % 2pDD
        u=real(conj(zo(s1)).*(mpskqntz(zo(s0),M)-mpskqntz(zo(s2),M)));
        sc2pDDm(n,t+1)=mean(u);
    end;
end;
if NEns>1
    sc1pDDm=mean(sc1pDDm);
    sc1pADm=mean(sc1pADm);
    sc2pNDAm=mean(sc2pNDAm);
    sc2pADm=mean(sc2pADm);
    sc2pDDm=mean(sc2pDDm);
end;
save scdata sc1pDDm sc1pADm sc2pNDAm sc2pADm sc2pDDm Sf R M Beta;
end;
%
% Now Process This Data Against Whatever Theoretical Values We Have
%
clear all;
close all;
load scdata;
t=linspace(-0.5,0.5,Sf+1);
kbd=input('Input Linestyle : ','s');
%
% 1pDD
%
figure(1);
set(plot(t,sc1pDDm,kbd),'MarkerSize',3);
hold on;
plot(t,R*(rcos(t+1,Beta)-rcos(t-1,Beta)));
title('1pDD');
%
% 1pAD
%
figure(2)
set(plot(t,sc1pADm,kbd),'MarkerSize',3);
hold on;
if M==2
    plot(t,R*(rcos(t+1,Beta)-rcos(t-1,Beta)));
else
    plot(t,(R*0.5)*(rcos(t+1,Beta)-rcos(t-1,Beta)));
end;
title('1pAD');
%
% 2pDD
%
figure(3);
set(plot(t,sc2pDDm,kbd),'MarkerSize',3);
hold on;
ddresp=rcos(t+1/2,Beta)-rcos(t-1/2,Beta);
plot(t,R*ddresp);
title('2pDD');
%
% 2pNDA
%
figure(4)
set(plot(t,sc2pNDAm,kbd),'MarkerSize',3);
hold on;
ndaresp=zeros(size(t));
for n=-50:50,
    ndaresp=ndaresp+rcos(t-(n+0.5),Beta).*rcos(t-(n+1),Beta)-rcos(t-(n+0.5),Beta).*rcos(t-n,Beta);
end;
plot(t,R*R.*ndaresp);
title('2pNDA');
%
% 2pAD
%
figure(5)
set(plot(t,sc2pADm,kbd),'MarkerSize',3);
hold on;
denom=zeros(size(t));
for n=-50:50,
```

```
    denom=denom+rcos(t-n,Beta).*rcos(t-n,Beta);
end;
denom=sqrt(denom);
if M==2
    plot(t,R*ddresp);
else
    plot(t,0.5*R*(ndaresp./denom+ddresp) );
end;
title('2pAD');
```

# MATLAB Program to Determine ML-FED Open-Loop Characteristics

```
%
% MATLAB Script for determining the S-curve characteristic for the ML-FED
%
clear all;
close all;
NS=input('How many symbols ');
SF=input('Sampling frequency ');
Beta=input('Root-raised cosine roll-off ');
NTaps=input('Number of Taps in Filters ');
NPoints=input('Number of Points in S-Curve ');
%
% Create Matrix of NSxSF
%
rdata=zeros(NS,SF);
idata=zeros(NS,SF);
%
% Set First Column to be +-1/sqrt(2)
%
rdata(:,1)=(SF/sqrt(2))*(2.0.*(rand(NS,1)>0.5)-1.0);
rdata=reshape(rdata',1,NS*SF);
idata(:,1)=(SF/sqrt(2))*(2.0.*(rand(NS,1)>0.5)-1.0);
idata=reshape(idata',1,NS*SF);
RandData=rdata+i*idata;
clear rdata;
clear idata;
%
% Signal Matched Filter is Root Raised Cosine
% Design this using Window method
%
f=linspace(0,SF/2);
f1=(1-Beta)*0.5;
f2=(1+Beta)*0.5;
m=zeros(size(f));
i=find(f < f1);
m(i)=ones(size(i));
i=find((f >= f1) & (f < f2));
m(i)=cos(pi*0.5*(f(i)-0.5+0.5*Beta)/Beta);
f=linspace(0,1,max(size(m)));
smf=fir2(NTaps-1,f,m,boxcar(NTaps));
%
% Frequency Matched Filter is derived from Signal Matched Filter
% Impulse Response as follows
%
t=linspace(-NTaps/(2*SF),NTaps/(2*SF),NTaps);
fmf=-2*pi*t.*smf;
disp('Hit any key to start simulation');
pause;
RandData=filter(smf,1,RandData);
i=sqrt(-1);
f=linspace(0,1.5,NPoints);
for n=1:NPoints,
    n
    phasor=exp(i*2*pi*f(n)*(0:SF*NS-1)/SF);
    input=RandData.*phasor;
    fed=filter(smf,1,real(input)).*filter(fmf,1,imag(input))...
```

```
    -filter(smf,1,imag(input)).*filter(fmf,1,real(input));
  for k=1:3,
      scm(n,k)=mean(fed(k+NTaps+1:SF:SF*NS));
  scv(n,k)=cov(fed(k+NTaps+1:SF:SF*NS));
  end;
end;
figure(1);
plot(f,scm(:,1));
hold on;
plot(f,scm(:,2),'--');
plot(f,scm(:,3),'-.');
figure(2);
plot(f,scv(:,1));
hold on;
plot(f,scv(:,2),'--');
plot(f,scv(:,3),'-.');
```

# MATLAB Program to Determine BQ-FED Open-Loop Characteristics

```
%
% MATLAB Script for determining the S-curve characteristic for the BQ-FED
%
clear all;
close all;
NS=input('How many symbols ');
SF=input('Sampling frequency ');
Beta=input('Root-raised cosine roll-off ');
NTaps=input('Number of Taps in Filters ');
NPoints=input('Number of Points in S-Curve ');
%
% Create Matrix of NSxSF
%
rdata=zeros(NS,SF);
idata=zeros(NS,SF);
%
% Set First Column to be +-1/sqrt(2)
%
rdata(:,1)=(SF/sqrt(2))*(2.0.*(rand(NS,1)>0.5)-1.0);
rdata=reshape(rdata',1,NS*SF);
idata(:,1)=(SF/sqrt(2))*(2.0.*(rand(NS,1)>0.5)-1.0);
idata=reshape(idata',1,NS*SF);
RandData=rdata+i*idata;
clear rdata;
clear idata;
%
% Signal Matched Filter is root raised cosine
% Design this using the window design method
%
f=linspace(0,SF/2);
f1=(1-Beta)*0.5;
f2=(1+Beta)*0.5;
m=zeros(size(f));
i=find(f < f1);
m(i)=ones(size(i));
i=find((f >= f1) & (f < f2));
m(i)=cos(pi*0.5*(f(i)-0.5+0.5*Beta)/Beta);
f=linspace(0,1,max(size(m)));
smf=fir2(NTaps-1,f,m,boxcar(NTaps));
%
% For simulation of Balanced Quadricorrelator need to define Butterworth
% filter. Center frequency is 0.6, bandwidth is 0.2. This is relative to a
% sample frequency of 4.
%
[bwb,bwa]=butter(1,[0.25,0.35]);
disp('Hit any key to start simulation');
pause;
RandData=filter(smf,1,RandData);
```

```
i=sqrt(-1);
f=linspace(0,1.5,NPoints);
strt=200;
bqre=zeros(1,SF*NS);
bqim=zeros(1,SF*NS);
fed=zeros(1,SF*NS);
for n=1:NPoints,
    n
    phasor=exp(i*2*pi*f(n)*(0:SF*NS-1)/SF);
    input=RandData.*phasor;
    bqre=filter(bwb,bwa,real(input));
        bqim=filter(bwb,bwa,imag(input));
    fed=bqre(strt:1:SF*NS-2).*bqim(strt+2:1:SF*NS)-bqim(strt:1:SF*NS-2).*bqre(strt+2:1:SF*NS);
        scm(n)=mean(fed);
    scv(n)=cov(fed);
    end;
end;
figure(1);
plot(f,scm);
figure(2);
plot(f,scv);
```

# MATLAB Program to Design Filters for the ML-FED

```
clear all;
close all;
SF=input('Sampling frequency ');
Beta=input('Root-raised cosine roll-off ');
NTaps=input('Number of Taps in Filters ');
% Write value of roll-off factor to a file
fp=fopen('beta','W');
fprintf(fp,'%e\n',Beta);
fclose(fp);
% SMF is root raised cosine with transfer function
f=linspace(0,SF/2);
f1=(1-Beta)*0.5;
f2=(1+Beta)*0.5;
m=zeros(size(f));
i=find(f < f1);
m(i)=ones(size(i));
i=find((f >= f1) & (f < f2));
m(i)=cos(pi*0.5*(f(i)-0.5+0.5*Beta)/Beta);
% Plot desired response
figure;
semilogy(f,m);hold on;
fd=linspace(0,1,max(size(m)));
% Design filter using window
smfboxcar=firls(NTaps-1,fd,m);
% Normalise coefficients
smfboxcar=smfboxcar./sum(smfboxcar);
% Plot actual response
[h,fp]=freqz(smfboxcar,1,100,SF);
semilogy(fp,abs(h),'--');
axis([0 2 0.001 10]);
xlabel('Frequency');
ylabel('Filter Response');
grid;
axis('square');
lgd=legend('Ideal Response','FIR Design');
set(lgd,'Position',[0.4836 0.7193 0.3216 0.1195]);
axes(lgd);
% Now do FMF generate this from SMF
t=linspace(-NTaps/(2*SF),NTaps/(2*SF),NTaps);
fmfboxcar=-2*pi*t.*smfboxcar;
% Work out ideal response for FMF
m=zeros(size(f));
i=find((f >= f1) & (f < f2));
m(i)=(pi*0.5/Beta)*sin(pi*0.5*(f(i)-0.5+0.5*Beta)/Beta);
% Plot The FMFs
```

```
figure;
semilogy(f,m);hold on;
[h,fp]=freqz(fmfboxcar,1,100,SF);
semilogy(fp,abs(h),'--');
axis([0 2 0.001 10]);
xlabel('Frequency');
ylabel('Filter Response');
grid;
axis('square');
lgd=legend('Ideal Response','FIR Design');
set(lgd,'Position',[0.4836 0.7193 0.3216 0.1195]);
axes(lgd);
figure;
plot(0:NTaps-1,smfboxcar);
grid;
axis('square');
axis([0 NTaps-1 -0.1 0.3])
figure;
plot(0:NTaps-1,fmfboxcar);
grid;
axis('square');
axis([0 NTaps-1 -0.6 0.6]);
%
spwsavefilt(smfboxcar,1,'smf');
spwsavefilt(fmfboxcar,1,'fmf');
```

# MATLAB Program to Design Estimation Filters for the Rotational FED

```
% Filter design script under MATLAB for designing smoothing filter
% for Rotational Frequency Error Detectors.
% There are two methods
%   (i) Sliding Average
%   (ii)    More involved based on Moenclaey's work
%
% Last Edit 12-10-94
% Dan Verdin
%
clear all;
ntaps=input('Number of taps in filter design method : ');
method=input('Design method. Sliding Average (1) or Optimum (2) : ');
if method ==1
    b=(1/ntaps).*ones(size(1:ntaps));
    a=1;
    spwsavefilt(b,a,'rotfilt');
else
    k=0:(ntaps-1)/2;
    K=ntaps;
    b((ntaps+1)/2:ntaps) = 6/(K*(K*K-1))*( (K*K-1)/4-k.*(k-1));
    b(1:1:(K-1)/2) = b(K:-1:(K+3)/2);
    s=sum(b);
    b=b./s;
    a=1;
    spwsavefilt(b,a,'rotfilt');
end
```

# MATLAB Program to Determine Open-Loop Characteristics of Timing-Error Detectors

```
%
% This MATLAB file determines the open-loop gain of the timing error
% detectors as a function of signal to noise ratio.
%
```

```
close all;
clear all;
%
% Get user inputs
%
tic;
nsymbols=100000;
nens=2;
m=4;
r=1.0;
sf=4.0;
ntaps=33;
pltpts=20;
alpha=[0.2 0.5 1.0];
% 5 TEDs in this order 1pAD, 1pDD, 2pNDA, 2pAD, 2pDD.
for ted=1:5,
    for a=1:3,
        %
        % Work out gain of detector
        %
        % Generate Raised Cosine filter
        %
        rcir=rrcos(1.0,sf,alpha(a),ntaps,1);
        %
        % Apply a small fractional delay to this
        %
        n=-(ntaps-1)/2:(ntaps-1)/2;
        ts=0.01;
        intrp=sinc(n-ts*sf).*hanning(max(size(n)))';
        intrp=intrp./sum(intrp);
        rcir1=conv(rcir,intrp);
        %
        % Average over each ensemble to give more accurate result
        %
        npts=nsymbols*sf;
        start=(ntaps+sf)/2;
        start=start+5*sf/2;
        s0=(start:sf:npts-sf-start);
        s1=(start+sf/2:sf:npts-sf/2-start);
        s2=(start+sf:sf:npts-start);
        esno=linspace(-5.0,35.0,pltpts);
        %
        for n=1:pltpts,
            for ens=1:nens,
                if ted==1 disp('1pAD: ');
                elseif ted==2 disp('1pDD: ');
                elseif ted==3 disp('2pNDA: ');
                elseif ted==4 disp('2pAD: ');
                else disp('2pDD: ');
                end;

                disp(['alpha = ',num2str(alpha(a)),', ','EsNo(',num2str(n),') = ',num2str(esno(n)),...
                      ', ','Ensemble = ',num2str(ens),'   ']);
                % Generate random M-PSK data
                zi=upsample(sf*r*randmpsk(nsymbols,1,m),sf);
                % Pass through transmit filter
                zo=fftfilt(rcir,zi);
                %
                zo=zo+awgn(zo,esno(n),2*ntaps+sf,0.5,sf);
                % Pass through receiver filter
                z1=fftfilt(rcir1,zo);
                % Description of Timing Error Detector
                if ted==1
                u=real(conj(z1(s2)).*z1(s0)./abs(z1(s2))-conj(z1(s0)).*z1(s2)./abs(z1(s0))); % 1pAD
                elseif ted==2
                u=real(conj(mpskqntz(z1(s2),m)).*z1(s0)-conj(mpskqntz(z1(s0),m)).*z1(s2)); % 1pDD
                elseif ted==3
                u=real(conj(z1(s1)).*(z1(s2)-z1(s0))); % 2pNDA
                elseif ted==4
                u=real(conj(z1(s1)).*(z1(s2)./abs(z1(s2))-z1(s0)./abs(z1(s0)))); % 2pAD
                else
```

```
    u=real(conj(z1(s1)).*(mpskqntz(z1(s2),m)-mpskqntz(z1(s0),m))); % 2pDD
    end;

gain(ens)=mean(u)/ts;
% Pass through receiver filter
zo=fftfilt(rcir,zo);
% Description of Timing Error Detector
    if ted==1
    u=real(conj(zo(s2)).*zo(s0)./abs(zo(s2))-conj(zo(s0)).*zo(s2)./abs(zo(s0))); % 1pAD
    elseif ted==2
    u=real(conj(mpskqntz(zo(s2),m)).*zo(s0)-conj(mpskqntz(zo(s0),m)).*zo(s2)); % 1pDD
elseif ted==3
        u=real(conj(zo(s1)).*(zo(s2)-zo(s0))); % 2pNDA
        elseif ted==4
        u=real(conj(zo(s1)).*(zo(s2)./abs(zo(s2))-zo(s0)./abs(zo(s0)))); % 2pAD
    else
        u=real(conj(zo(s1)).*(mpskqntz(zo(s2),m)-mpskqntz(zo(s0),m))); % 2pDD
        end;
    h=psd(u(2:max(size(u))),64,2,boxcar(64),'none');
    var(ens)=h(1);
    end; % of Ensemble loop
    G(n)=abs(mean(gain))
    V(n)=mean(var)
end; % of Pltpts loop
if ted==1
   if a==1
       save 1pad1 esno G V;
    elseif a==2
        save 1pad2 esno G V;
    else
        save 1pad3 esno G V;
    end;
elseif ted==2
    if a==1
        save 1pdd1 esno G V;
    elseif a==2
        save 1pdd2 esno G V;
    else
        save 1pdd3 esno G V;
    end;
elseif ted==3
    if a==1
        save 2pnda1 esno G V;
    elseif a==2
        save 2pnda2 esno G V;
    else
        save 2pnda3 esno G V;
    end;
elseif ted==4
    if a==1
        save 2pad1 esno G V;
    elseif a==2
        save 2pad2 esno G V;
    else
        save 2pad3 esno G V;
        end;
    else
        if a==1
            save 2pdd1 esno G V;
        elseif a==2
            save 2pdd2 esno G V;
        else
            save 2pdd3 esno G V;
        end;
    end;
  end; % of alpha loop
end; % of ted loop
return;
% Plot the simulation results

plot(esno,10*log10(V./(G.*G)),'g*');hold on;
```

```
% Plot analytical solution to TED
% A=0.222; % 2pDD, alpha=1
A=0.406; B=0.0215; % 2pNDA, alpha=0.5
% A=0.211; % 2pNDA, alpha=1
plot(esno,10*log10(0.5*A./(10.^(esno./10))+B.^2));
% This is the CRLB. Plot this independent of TED
crlb=1.0/(2.0*(alpha*alpha*pi*pi+pi*pi/3.0-8.0*alpha*alpha));
plot(esno,-esno+10.0*log10(crlb),'r');
toc;
```

# "Communications" Toolbox for MATLAB

## Function to Generate Random M-PSK Data

```
function x = randmpsk(nbits,Fs,M)
%RANDDATA Generates random MPSK symbols.
%   RANDMPSK(NBITS,FS,M) returns randomly generated MPSK symbols
%   sample frequency FS.
error(nargchk(2,3,nargin));
if nargin<3
    M=2;
end
x=zeros(Fs,nbits);
tmp=rand(1,nbits);
for m=0:M-1,
    n=find((tmp>m/M) & (tmp<=(m+1)/M) );
    tmp(n)=(m/M).*ones(size(n))+1/(2*M);
end;
x=exp(i*2*pi*ones(Fs,1)*tmp);
x=reshape(x,nbits*Fs,1);
```

## Function to Implement MPSK Quantizer

```
function y = mpskqntz(x,M)
%MPSKQNTZ Quantizer for MPSK
%   MPSKQNTZ(X,M) Quantizes MPSK Symbols To Nearest Constellation

error(nargchk(1,2,nargin));
if nargin<2
   M=2;
end
tmp=zeros(size(x));
% Use Angle Only
arg=angle(x)./(2*pi);
for m=-M/2:(M/2-1),
    n=find( (arg>m/M) & (arg<=(m+1)/M) );
    tmp(n)=(m/M).*ones(size(n))+1/(2*M);
end;
y=exp(i*2*pi*tmp);
```

## Function to Implement Raised Cosine Filter

```
function rcir = rrcos(baud,sf,alpha,ntaps,rootflag,sincflag)
%RRCOS  Raised & Root Raised Cosine filter design.
%   Filter design done in frequency domain.
%       Y=RRCOS(baud,sf,alpha,ntaps,rootflag,sincflag) returns fir filter taps.
%       baud is the baud rate
%   sf is the sampling frequency
%       alpha is the excess bandwidth 0 < alpha < 1
%   ntaps is the number of taps
%   rootflag specfies whether to use raised cosine (0)
%       or root raised cosine (1) filter
%   sincflag specifies whether to include x/sin(x) equalisation
%       (0) no, use impulsive data
%       (1) yes, use square-wave data
```

```
%

error(nargchk(6,6,nargin))
f=linspace(0,sf*0.5);
f1=(1-alpha)*0.5;
f2=(1+alpha)*0.5;
m=zeros(size(f));
i=find(f < f1);
m(i)=ones(size(i));
i=find((f >= f1) & (f < f2));
m(i)=cos(pi*0.5*(f(i)-0.5+0.5*alpha)/alpha);
if rootflag==0 m(i)=m(i).*m(i); end
if sincflag~=0
    sineq=zeros(size(f));
    i=find((f > 0) & (f < 1));
    sineq(i)=(pi*f(i))./sin(pi*f(i));
    sineq(1)=1;
    m=m.*sineq;
end;
f=linspace(0,1,max(size(m)));
rcir=firls(ntaps-1,f,m);
% Normalise filter coefficients
rcir=rcir./sum(rcir);
```

# Function to Generate Correlation Shaping Filter & Narrowband Noise

```
function [y,ir] = nbfade(npts,fdt,ntaps,window,sf)
%NBFADE Returns a narrowband frequency non-selective fading process
%   NPTS is the number of points
%   FDT is the Doppler Frequency Normalised to the Baud Rate
%   NTAPS is the number of taps in the noise shaping filter
%   Window is the weighting window to be used
%   SF is the Sample Frequency
%   The function returns a fading process with a PSD shaped appropriately
%   for the Land-Mobile Radio Channel.

error(nargchk(5,5,nargin))
% Generate unit variance complex white noise process
y=(randn(1,npts+ntaps)+i*randn(1,npts+ntaps))/sqrt(2);
% Design filter using time domain procedure
t=linspace(-ntaps/(2*sf),ntaps/(2*sf),ntaps);
ir=1.468813.*ones(size(t));
n=find(t);
ir(n)=besselj(0.25,2*pi*fdt*abs(t(n)))./(abs(t(n)).^0.25);
ir=ir.*window';
ir=ir./sum(ir);
% Done the filter design now filter noise
y=fftfilt(ir,y);
% Don't return start-up values of filter
y=y(1+ntaps:npts+ntaps);
```

# Function to Plot Eye Diagram

```
function y = eyed(x,Ns,Fs,Offset)
%EYED Plots Eye Diagram.
%   EYED(X,NS,FS) Plots the eye diagram for the REAL signal in X.
%       NS is the number of symbols displayed.
%   FS is the sample frequency.
%
%   EYED(X,NS,FS,Offset) Adds in an optional offset

error(nargchk(3,4,nargin))

if nargin<4,
    Offset=0;
end

Npts = max(size(x));
```

```
if (Offset>Npts),
    error('Requested Offset is Greater Than The Total Number of Points');
end

Nens = fix((Npts-Offset)/(Ns*Fs));
t = linspace(0,Ns,Ns*Fs+1);
x = reshape(x(1+Offset:Nens*Ns*Fs+Offset),Ns*Fs,Nens);
x(Ns*Fs+1,1:Nens-1)=x(1,2:Nens);
x(Ns*Fs+1,Nens)=nan;
plot(t,x,'y-');
```

## Function to Generate Gaussian Filter

```
function y = gaussian(t,B)
%GAUSSIAN Returns pulse response of Gaussian filter using time-domain formula.
%    Y=GAUSSIAN(t,B) returns fir filter taps.
%    NTaps is the time vector.
%    B is the 3dB bandwidth
%    Baud rate of the data is assumed to be one

error(nargchk(2,2,nargin))

t1=t-1/2;
t2=t+1/2;
sigma=sqrt(log(2)/2)/B;
y = 0.5*(erf(-pi*t1/sigma)+erf(pi*t2/sigma));
```

## Function to Generate $\Phi(t)$ Function for GMSK

```
function phi=gmskphi(t,B)
%GMSKPHI Returns phase trajectory for GMSK modulation
%        T is time vector
%        B is premodulation filter time-bandwidth product

error(nargchk(2,2,nargin))

s=2*pi*B/sqrt(log(2));

t1=t-0.5;
t2=t+0.5;

phi=0.5+0.5*t2.*erf(s*t2/sqrt(2))-0.5*t1.*erf(s*t1/sqrt(2))...
       +(exp(-0.5*s^2*t2.^2)-exp(-0.5*s^2*t1.^2))/(sqrt(2*pi)*s);
```

## Function to Measure Level Crossing Rates

```
function y = lcr(x,levels,sf)
%LCR Returns the level crossing rate
%    X is the Input Signal
%    LEVELS are the required levels

error(nargchk(3,3,nargin))


Nlevels=max(size(levels));
Npts=max(size(x));

for l=1:Nlevels,
    y(l)=0.5*sf*sum(((x(1:Npts-1)<=levels(l)) & (x(2:Npts)>levels(l))) ..
                    | ((x(1:Npts-1)>=levels(l)) & (x(2:Npts)<levels(l))))/Npts;
end;
```

# Appendix C

# Publications

- "Simulation and Modelling of Mobile Communication Systems using the CAAD Tool SPW," Presented at the International Digital Signal Processing Conference 1991, Florence, ITALY.

- "Interpolator Filter Structure For Asynchronous Timing Recovery Loops," Electronics Letters, 4th March 1993, Vol. 29, No. 5, pp 490.

- "Simulating Asynchronous Timing-Recovery Loops Using Multirate Techniques," IEE Colloquium on Communications Simulation and Modelling Techniques, University of York, Heslington, York, September 1993.

- "Hangup in Asychronous Timing Recovery Loops," Electronics Letters, 28th October 1993, Vol. 29, No. 22, pp 1914.

- "Generating a Fading Process for the Simulation of Land-Mobile Radio Communications," Electronics Letters, 28th November 1993, Vol. 29, No. 23, pp 2011.

- "Symbol Timing Recovery Scheme Tolerant to Carrier Phase Error," Electronics Letters, 20th January 1994, Vol. 30, No. 2, pp 116

DIGITAL SIGNAL PROCESSING - 91
V. Cappellini and A.G. Constantinides (eds.)




# Simulation And Modelling Of Mobile Communication Systems Using The CAAD Tool SPW

D. Verdin & T. C. Tozer

Communications Research Group, University Of York,
YORK, YO1 5DD, ENGLAND.

This paper is concerned with the application of the Computer Aided Analysis and Design (CAAD) tool The Signal Processing Worksystem (SPW) to the modelling and simulation of mobile communication systems. Specific systems addressed are the coherent and incoherent detection schemes for the modulation format Gaussian Minimum Shift Keying (GMSK) including Decision Feedback to reduce the inherent Intersymbol Interference. An SPW model of the wideband fading channel is also presented as part of this work.

## 1 Introduction

As the complexity of DSP and Digital Communication Systems continues to increase together with the need to have these systems implemented quickly and at minimal cost, the use of Computer-Aided Analysis and Design (CAAD) tools to speed up the design phase is becoming evermore important [1].

This paper illustrates the use of the CAAD tool The Signal Processing Worksystem (SPW), to model and simulate mobile communication systems. We illustrate the use of the CAAD tool through the application to a real-world problem, that of the design of appropriate schemes for the detection of the narrowband modulation scheme, Gaussian Minimum Shift Keying (GMSK), and the modelling of the mobile channel.

## 2 The Signal Processing WorkSystem

SPW is a block-oriented CAAD software package for the modelling and simulation of DSP and Digital Communication Systems. SPW designs are built up via schematic capture of blocks available within system libraries to reflect the Signal-Flow diagram of the intended system. When simulating a system, the Simulation Program Builder (SPB) translates this Signal-Flow diagram directly into executable code. The blocks that are available within the SPW libraries are extensive, covering the majority of the requirements for DSP and Digital Communication Systems. If this is not the case however, then provision is made for the inclusion of user written routines in the form of Custom Coded Blocks.

The user interfaces to the SPW package are the Block Diagram Editor (BDE) and the Signal Display Editor (SDE). Within the BDE the complexity of any system design can be controlled by using a hierarchical design approach. Designs are stored in three ways within the BDE in terms of increasing hierarchy as `.symbol`, `.detail` and at the topmost level `.system` files. A `.detail` of a particular signal processing system can be linked to a user-designed `.symbol`, and then this `.symbol` used as a black-box in further designs. This hierarchical nesting can proceed to any degreee desired, and serves to modularise the design for complex systems.

Having built up a complex DSP or digital communication system design, consisting of a number of hierarchical layers, it is possible to define parameters occuring at the `.symbol` level in terms of functions of those occuring at the `.system` level by using the Parameter Expression Language (PEL). In such a way it is possible to manipulate parameters in the topmost level of the design which will filter down to the lowest levels in the design. Thus having made a block `.detail` and having decided that this works as it should, it is possible to ignore the details of its inner-workings from a system level viewpoint.

The overall simulation can be controlled at an operating system level by the Macro Command Language (MCL). By using the MCL one can perform a whole series of simulations with different sets of parameters. This is especially useful in, for example, the bit-error rate characterisation of a particular system.

Examination of the signals at various points within the designs is provided by adding Signal Sink blocks or Signal Probes to the BDE design. The Signal Probes allow one to analyse signals within the SDE as the simulation is running. Within the SDE one also has analysis functions such as the display of Eye-Diagrams, Scatter Diagrams and Fourier Transforms. Signals may also be designed in the SDE to be used as Signal Source blocks within the BDE.

Having built up a design within the BDE the Code Generation System (CGS) allows one to generate code

for a number of host computers. Thus it is possible to generate optimised C code for a complex system and port this code onto a remote machine to speed up the computation time required for the simulation, or even onto a DSP chip as a first step toward practical implementation.

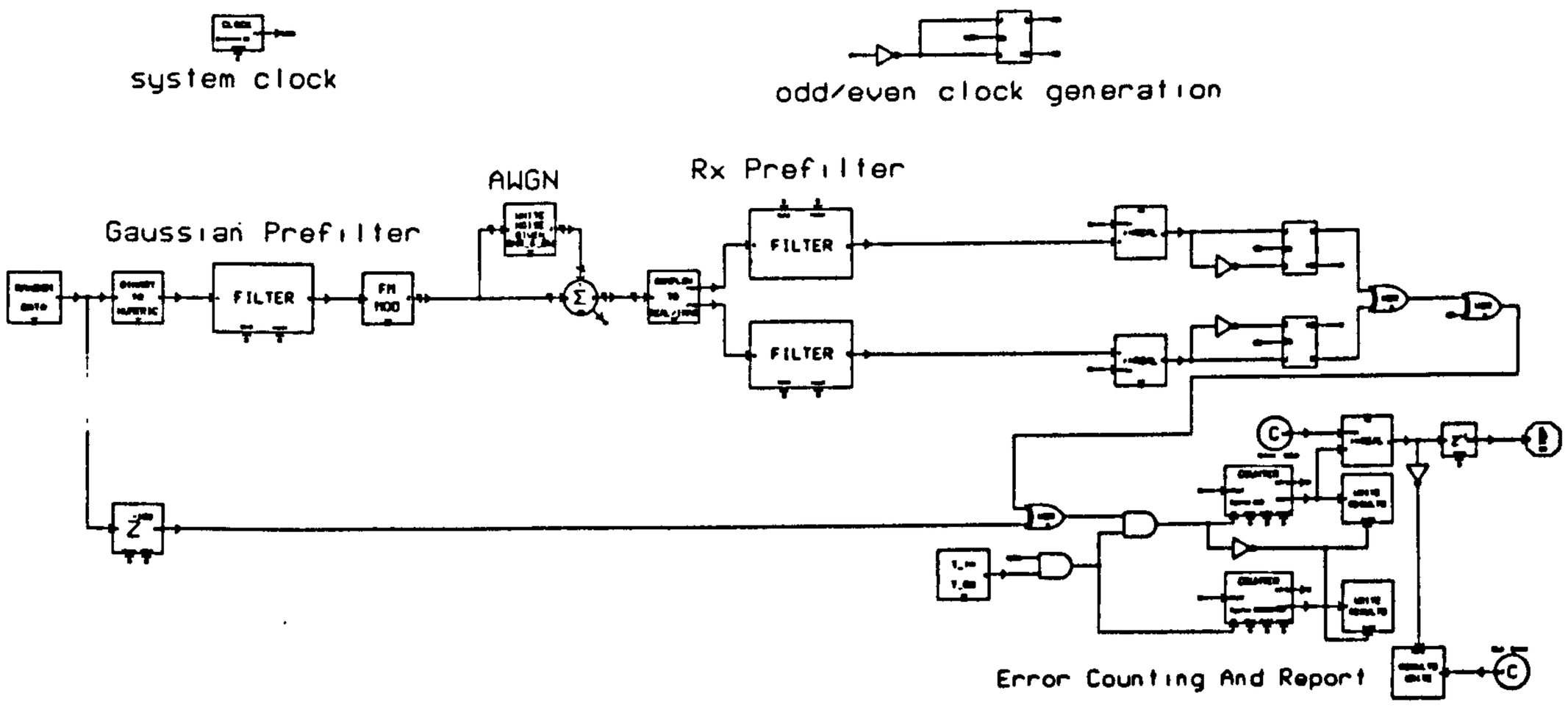



Figure 1: BDE Display Of Coherent Scheme For The Detection Of GMSK

# 3 SPW Implementation of Schemes for the Detection of GMSK

Gaussian Minimum Shift Keying (GMSK) has been adopted as the modulation standard for the new pan-European digital cellular network to be introduced in 1991. It is a narrowband scheme that presents good spectrum utilisation by keeping Adjacent Channel Interference (ACI) down to 60-80 dBs. GMSK can be detected coherently [2], by differential detection [3] or by limiter-discriminator detection [4]. The latter two schemes are incoherent and are hence more suitable to the fading channel although their theoretical performance is somewhat inferior to coherent detection.

Here we shall present the SPW implementation of both the coherent and the differential detection of GMSK and equalisation schemes using Decision Feedback to reduce the Intersymbol Interference inherent in such a narrowband scheme.

## 3.1 Coherent Detection of GMSK

A serial implementation of the GMSK modem is chosen as it is easier to introduce prefiltering to control the output power spectrum of the modulated signal. The BDE representation of the system for testing the bit-error rate (BER) performance of GMSK in AWGN is shown in fig 1. The scheme shown here is of the *orthogonal-coherent* detector type.

The important parameter for the Gaussian premodulation filter is its time-bandwidth product, $B_bT$. For GMSK $B_bT$ is generally taken to be 0.25 as this has optimum properties in terms of low spectral occupation and ease of demodulation [2]. The receiver filter is also a Gaussian filter with $B_bT = 0.63$, which represents an optimum configuration in terms of BER performance. The results of the BER simulation of the system represented in fig 1 correspond well with the theory outlined in [2] and serve to illustrate the ease with which simulations can be designed using SPW.

## 3.2 Differential Detection of GMSK

SPW was used to simulate the one- and two- bit Differential Detection of GMSK. The one-bit detection process consists of multiplying the received signal by a delayed and a 90° phase shifted version of itself and lowpass filtering the product. This is illustrated mathematically as

$$Y_1(t) = \cos\{\omega_c(t-T) + \phi(t-T) + \frac{\pi}{2}\}\cos\{\omega_c t + \phi(t)\}$$

When this is lowpass filtered to reject the second harmonic terms we get

$$Y_1(t) = \frac{1}{2}\sin(\Delta\Phi(T)) \qquad (1)$$

where we have used the fact that for MSK $\omega_c T = 2\pi k$ where $k$ is an integer and $\Delta\Phi(T) = \phi(t) - \phi(t-T)$ is the phase change over a bit period.

Because SPW allows one to use an analytical representation of signals, the above functions were obtained by the multiplication of the appropriate complex exponentials. Representation of the actual carrier frequency was thus not necessary.

For the two bit differential detector the input data stream is multiplied by a two bit delayed version of itself and lowpass filtered. The analysis of this gives

$$Y_2(t) = \frac{1}{2}\cos(\Delta\Phi(2T)) \qquad (2)$$

Equations (1) and (2) determine the shape of the eye diagrams when ISI is present as with GMSK. The eye diagrams for these two schemes are shown via the SDE in fig 2 & fig 3. Note that for the two bit case, the resultant eye-diagram is asymmetrical. This asymmetry leads to the fact that the optimum performance of two bit differential detection occurs with a non-zero detector threshold [5].

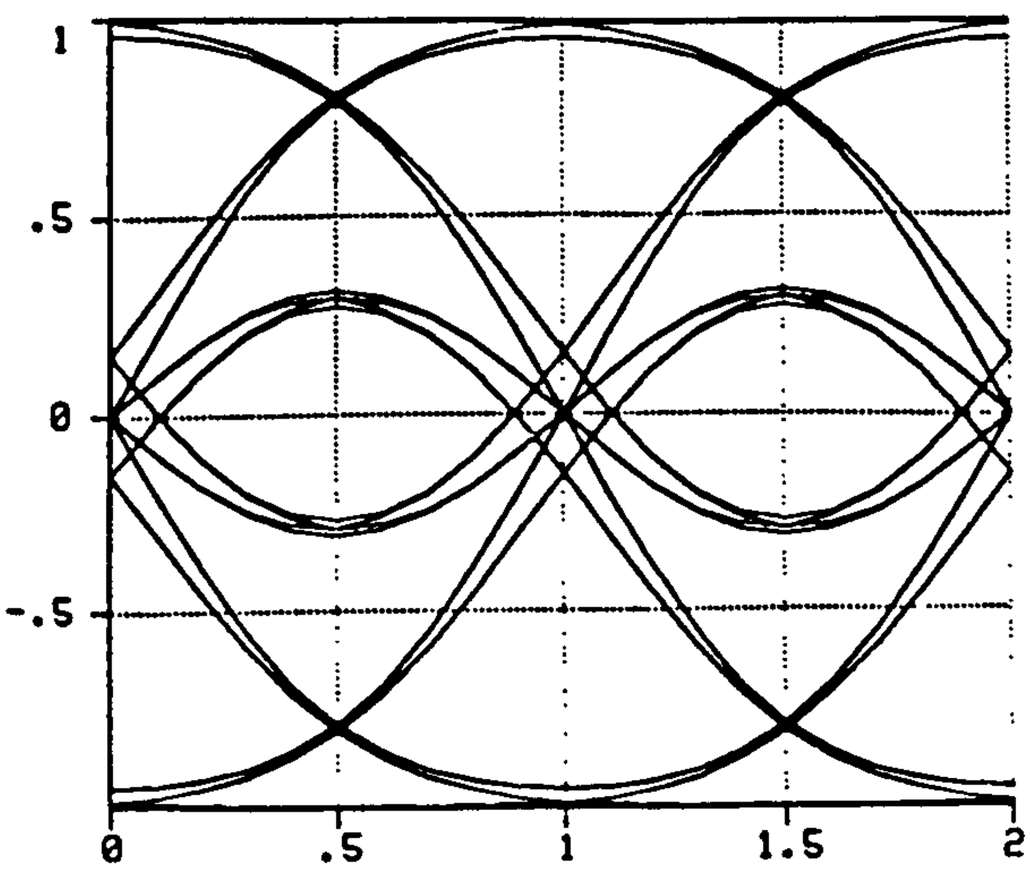



Figure 2: SDE Analysis Output of Eye Diagram for One-Bit Differential Detection of GMSK

### 3.2.1 Differential Detection Schemes with DFE

Decision Feedback Equalisation (DFE) can be used to reduce the ISI inherent in GMSK and thus improve the BER performance of the scheme [3].

Simple MSK can be represented as four phase PSK with phase states at 0°, ±90°, and ±180°. The effect of the Gaussian lowpass filter is to produce extra states at approximately ±37°, ±53°, ±127° and ±143° where

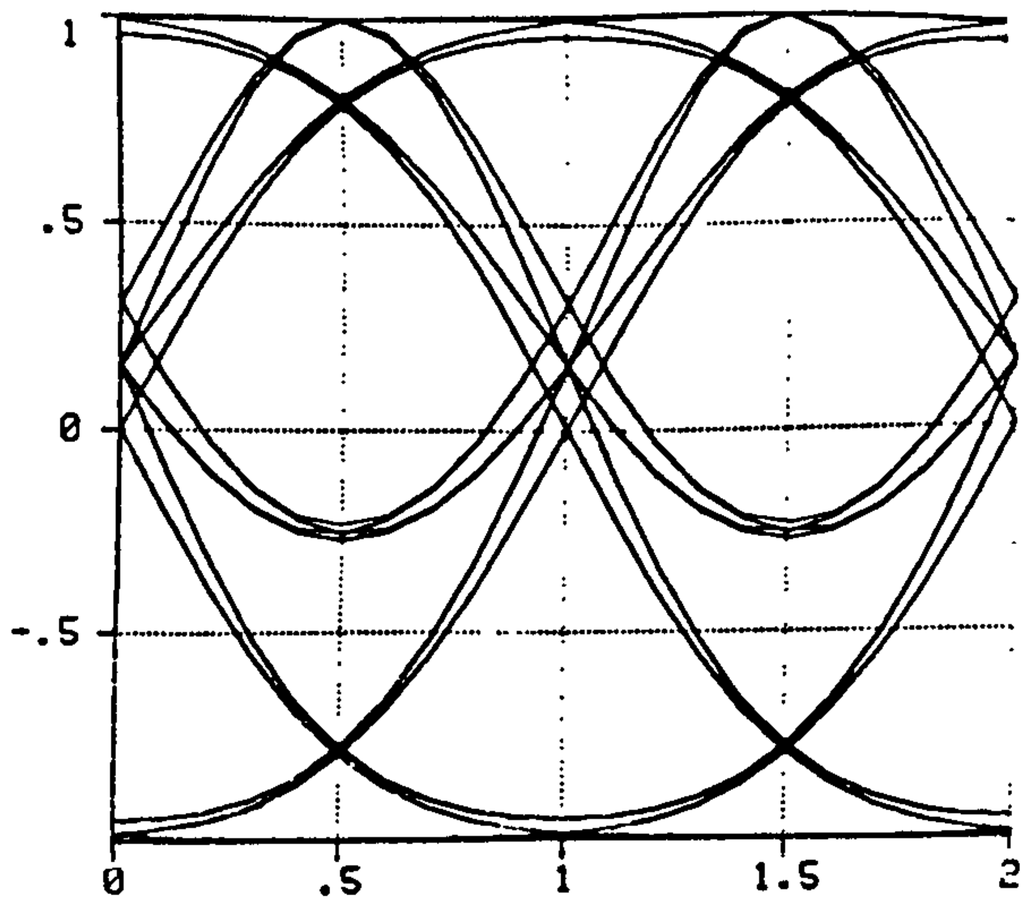



Figure 3: SDE Analysis Output of Eye Diagram for Two-Bit Differential Detection of GMSK

the deviation from the four-phase MSK representation is equal to twice the ISI, given by approximately 18.5° for $B_bT = 0.25$. These points are translated into the following table with one bit differential detection where the $b_k$ represents the $k^{th}$ bit.

| $b_{k+1}$ | $b_k$ | $b_{k-1}$ | $\theta$ |
|---|---|---|---|
| 0 | 0 | 0 | −90° |
| 0 | 0 | 1 | −53° |
| 0 | 1 | 0 | +16° |
| 0 | 1 | 1 | +53° |
| 1 | 0 | 0 | −53° |
| 1 | 0 | 1 | −16° |
| 1 | 1 | 0 | +53° |
| 1 | 1 | 1 | +90° |

The basis by which DFE works is to manipulate the points in the scatter diagram represented by the table above to increase the minimum Euclidean distance between the points in the positive and negative half of the scatter plot. Thus, without feedback the minimum separation is 32°. We can increase this by applying the following rule: if the previous bit was a 0 then rotate the signal vector by +18.5°, and if the previous bit was 1 then rotate the signal by −18.5°. The result is to change the signal constellation from six points with a minimum separation of 32°, to four points with a minimum separation of 69°. This corresponds to an increase in the eye opening as shown in fig 4 and an improvement in the BER performance of the system. Note that we can not completely get rid of the ISI as we do not have prior knowledge of the data bit $b_{k+1}$.

DFE applied to two bit differential detection is directly analogous to the one bit case, except here we

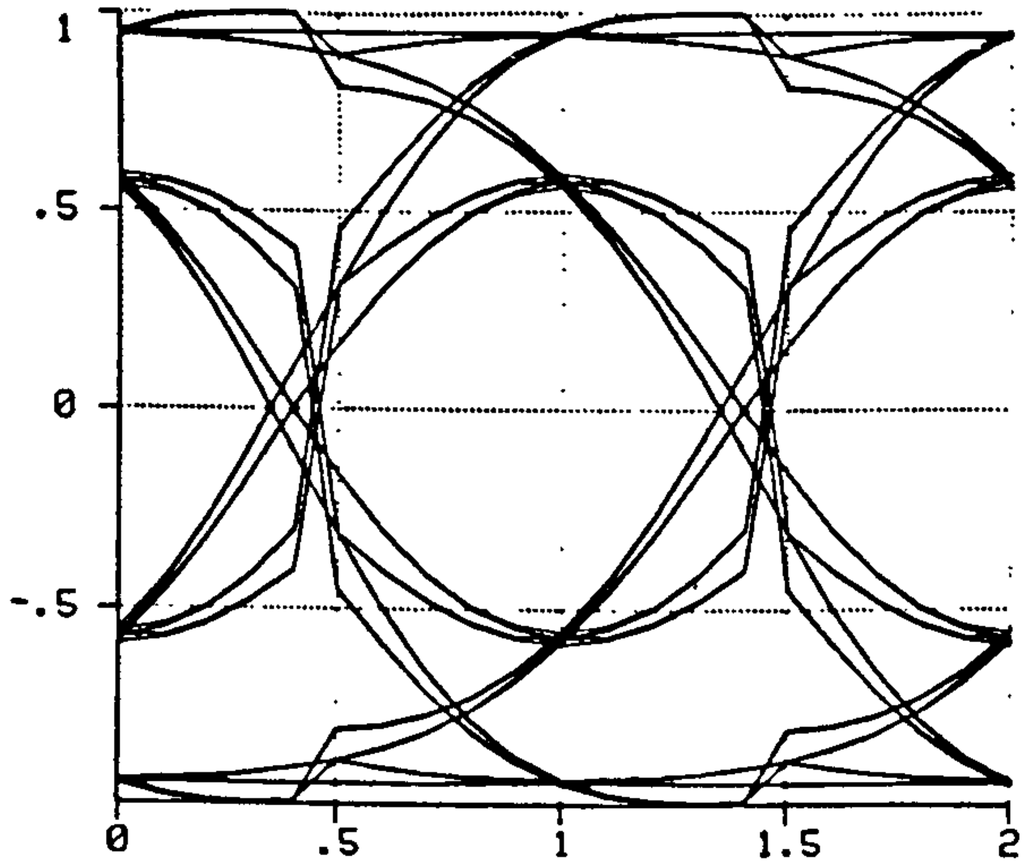



Figure 4: SDE Analysis Output of Eye Diagram for One-Bit Differential Detection of GMSK with DFE

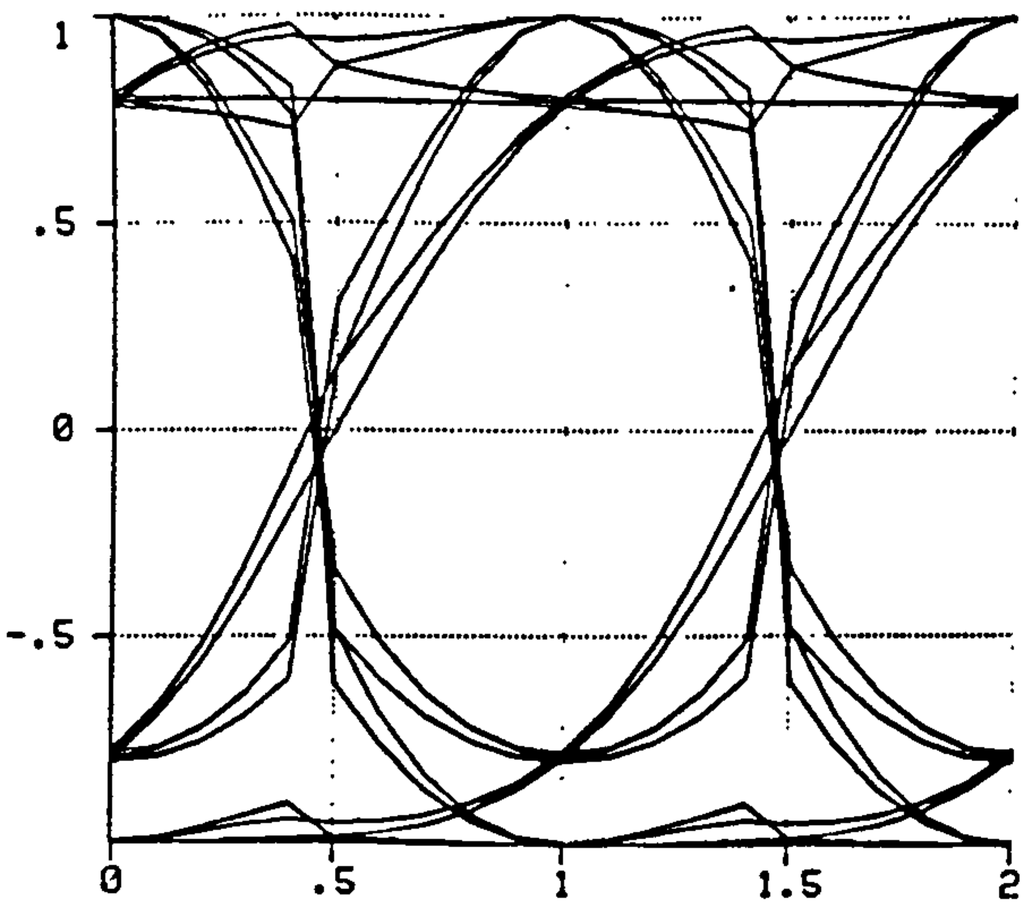



Figure 5: SDE Analysis Output of Eye Diagram for Two-Bit Differential Detection of GMSK with DFE

are looking at the data stream over two bit intervals. Intuitively, one would imagine that by looking at the data stream over a longer period of time would lead to less corruption from ISI – as is indeed the case. In two bit differential detection we are looking at phase *changes* over that two bit interval. Thus the scatter diagram for two bit differential detection has phase states throughout the whole of the $\pm 180°$ range. The two bit DFE scheme implemented using SPW is represented in fig 6. Note from the figure that the input data is differentially encoded, then passed through the GMSK `.symbol`, which is representative of a subsystem performing the GMSK modulation. It then passes through an AWGN channel with a $4^{th}$ order Butterworth receiver filter and is then differentially detected. A variable phase delay is induced in one arm of the detector, this phase being determined by the previous two bits. The improved eye diagram that results from using this DFE scheme is readily displayed in SPW, see fig 5.

### 3.3 Determination of the BER of Systems using SPW

The BDE block diagrams shown in fig 1 and fig 6 were compiled down to C code using the CGS. This code was then ported onto remote machines powerful enough to facilitate the BER simulation at low signal to noise ratios. The MCL was used at an operating system level to regenerate and run the code for different signal to noise conditions and store the output BER estimates. The results of the BER performance of the schemes discussed here is shown in fig 8 and agrees well with the theory presented in the references.

## 4 Simulation of Fading Channels Using SPW

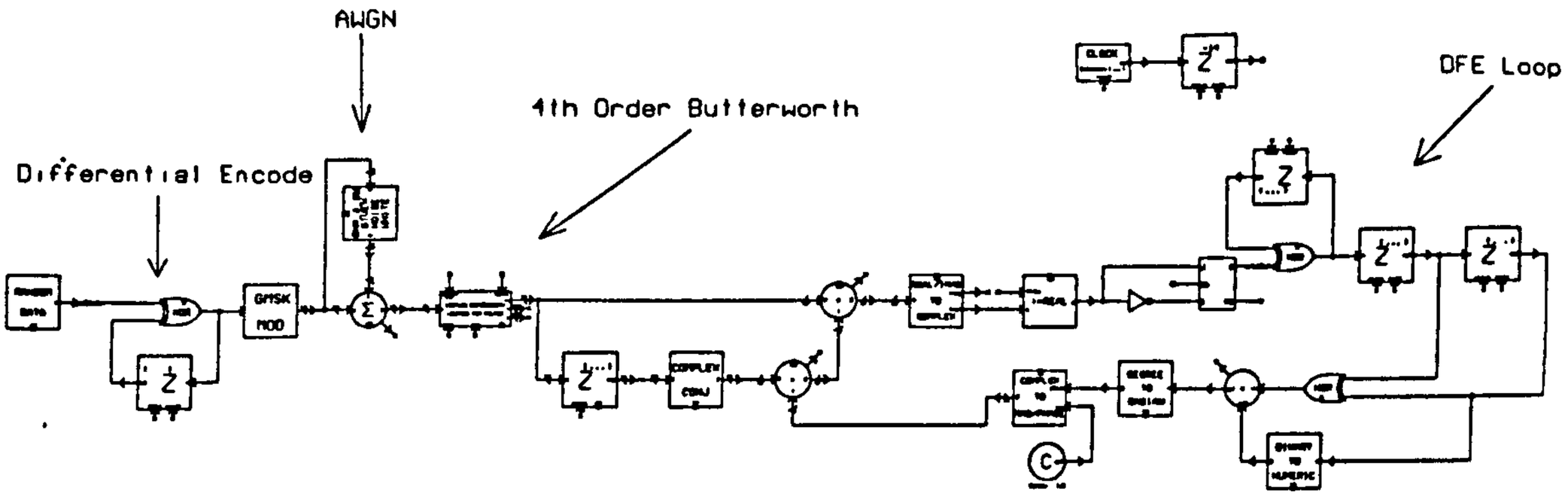
An SPW model of a wideband fading channel, based on a linear transversal filter was also developed. The number of taps in this filter was indicative of the number of paths required in the model. The tap coefficients of the filter were given by zero-mean complex Gaussian noise which was bandlimited in proportion to the Doppler frequency of the channel. The PEL was used to related the vehicle speed and carrier frequency to the bandlimiting of the filter taps. This can be seen by referring to fig 7 and noting the parameter screen.

The filter used to bandlimit the Gaussian noise was a Butterworth filter and thus the PEL was used to set each of the noise variances to

$$\sigma^2 = \frac{n f_s \sin(\frac{\pi}{2n})}{L \pi f_c}$$

were $n$ is the order of the lowpass Butterworth filter, $f_s$ is the sampling frequency, $L$ is the number of paths and $f_c$ is the cutoff frequency of the filter (which can be related to the Doppler spread of the channel).

This model was sucessfully used to simulate the effects of a wideband Rayleigh Fading channel. It would be simple to modify the block diagram in order to include any other effects of the mobile channel that one may wish to model, such as the inclusion of a Ricean fading component.



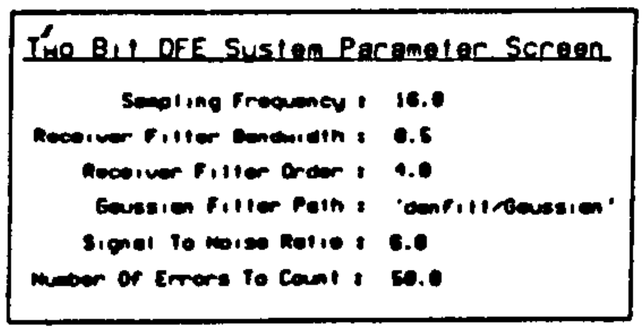




Figure 6: BDE Output for the Two-Bit Differential Detection Scheme of GMSK with DFE

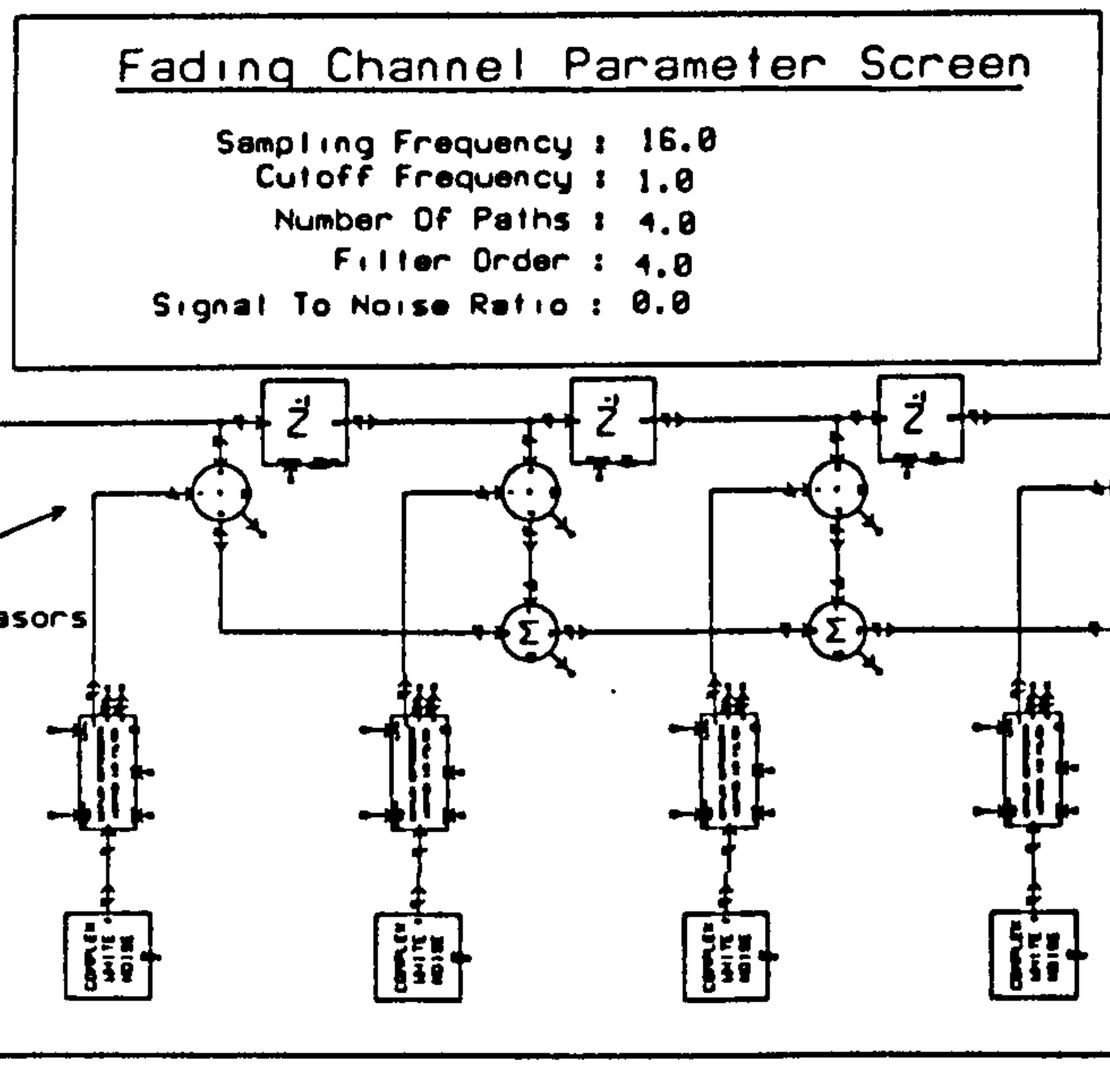



Figure 7: Wideband Fading Channel `.detail`

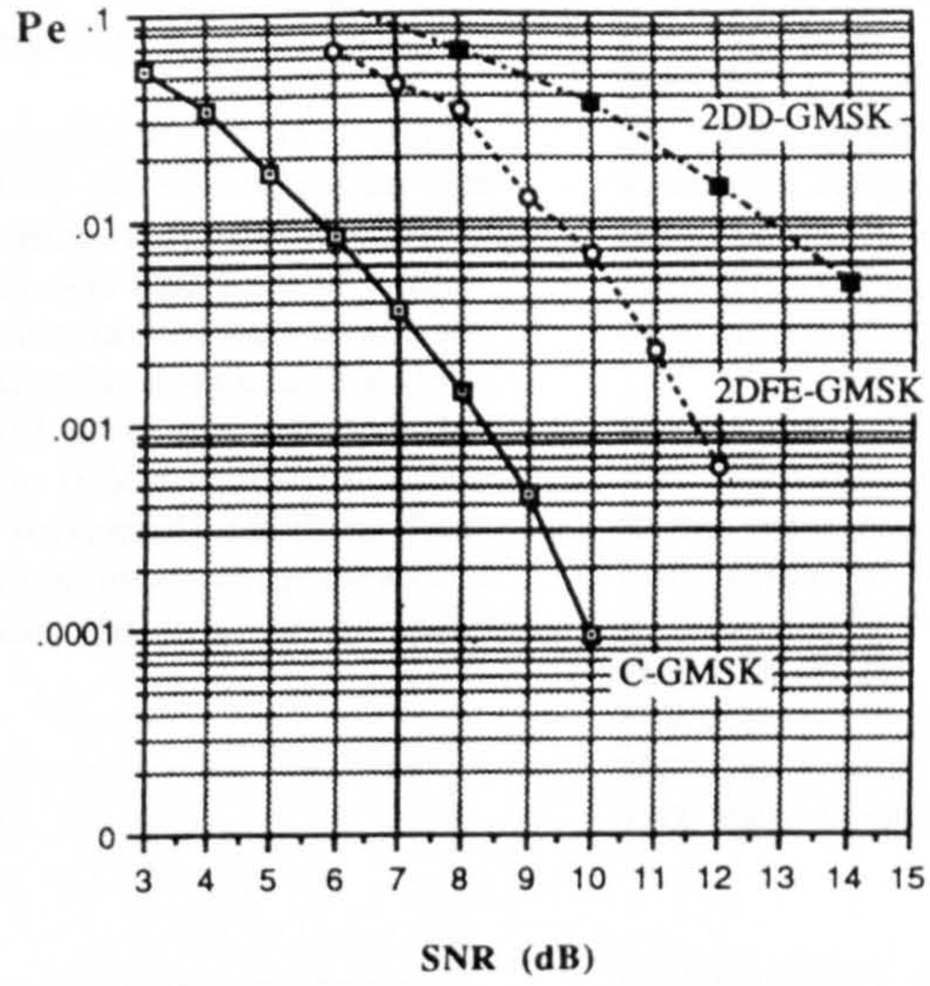



Figure 8: BER Performance Of Various GMSK Schemes In AWGN

# 5 Conclusions

This paper has served to illustrate the application of the CAAD tool SPW to the modelling and simulation of mobile communication systems. SPW has been shown to be highly flexible as a result of its extensive system libraries and its modular design approach.

# 6 Acknowledgements

The authors would like to thank RSRE & BAe for their support for this work. The Signal Processing Worksystem (SPW) is produced by Comdisco Systems Inc.

# References

[1] Shanmugan, S. 'Simulation Based CAAD Tools For Communication And Signal Procressing Systems', IEEE Int. Conf. Comms, Boston, June 1989

[2] Murota, K. 'GMSK Modulation For Mobile Radio Telephony', IEEE TRANS. Vol. COM-29, No 7, JULY 1981.

[3] Yongacoglu, A. 'Differential Detection Of GMSK Using Decision Feedback, IEEE TRANS. Vol. COM-36, No. 6, JUNE 1988

[4] Hirono M. 'Multilevel Decision Method For Band-Limited Digital FM With Limiter-Discriminator Detection', IEEE JSACS, Vol. 2, No. 4, JULY 1984

[5] Simon M. K.'Differential Detection Of Gaussian Minimum Shift Keying In A Mobile Radio Environment', IEEE TRANS VT, Vol. VT-33, No 4, November 1984.

implant doses were, $4 \times 10^{16}$ ion/cm$^2$ and $5 \times 10^{16}$ ion/cm$^2$, respectively, at an energy of 2·5 MeV. Refractive index profiles were determined from the mode spectra which were measured using a prism coupler in the dark mode configuration. Analysis of the mode data was performed using the reflectivity method [2].

Fig. 1 shows the refractive index profile of the $^4He^+$ implanated waveguide. The index increases by ~0·34% near the surface region, and decreases by a small amount (~0·12%) at the end of the ion range, producing in this way a small barrier. In general, nuclear damage domination (at the end of the ion range) leads to a decrease of the refractive index, and a combination of damage ionisation and diffusion may lead to either increase (silica glass) [3] or decrease (some multicomponent glasses) [4]. The optical well type waveguide resulting from refractive index enhancement has better mode confinement as there are nontunnelling modes. Thus, the loss due to leaky modes is completely eliminated.

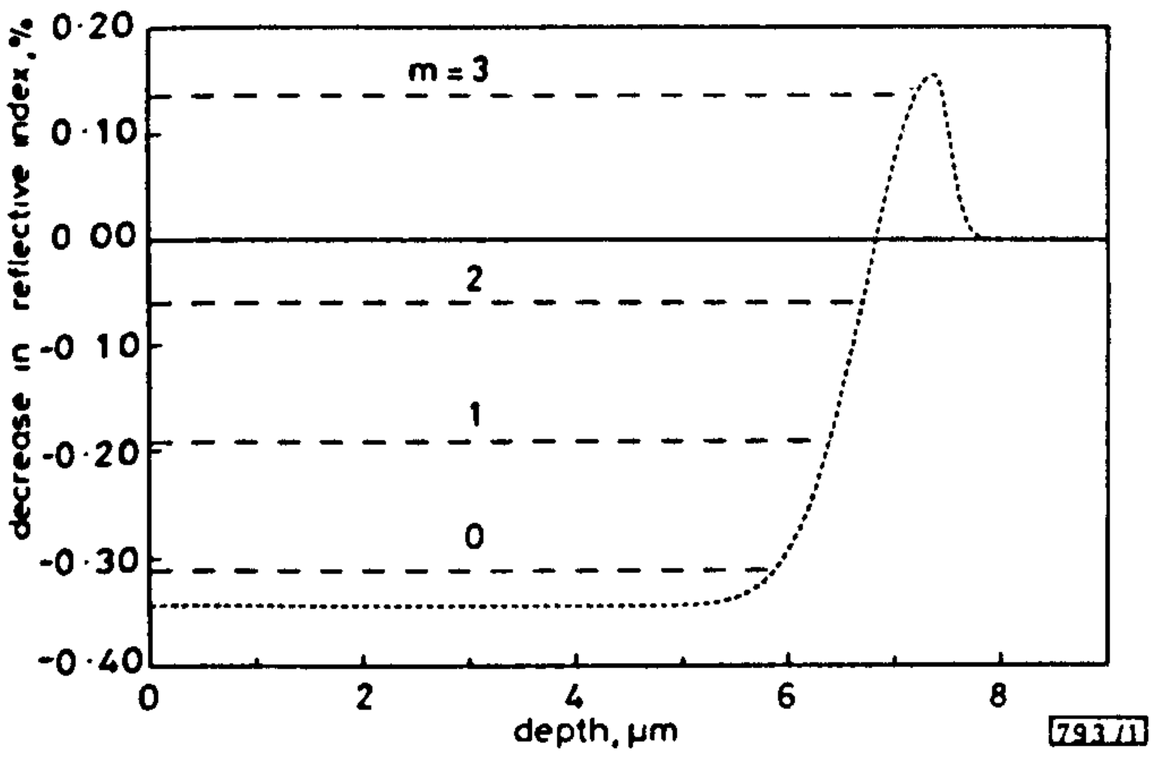



**Fig. 1** *Refractive index profile of lead germanate glass after $^4He^+$ at 2·5 MeV and dose of $4 \times 10^{16}$ ion/cm$^2$ at 77 K*

Mode values were measured at 633 nm

During ion implantation, ionisation energy may produce colour (absorption) centres which sometimes give high propagation losses within the waveguide. Fortunately, these defects are unstable and they are found to anneal out between 200 and 300°C in this glass.

Loss measurements were performed using the end-coupling configuration. A ×20 microscope objective lens was used to focus an He–Ne (633 nm) laser beam on one polished edge of the sample, and the output light from the waveguide was focused onto a silicon detector connected with a power meter.

Fig. 2 shows the loss reduction with annealing temperature for two waveguides. The first one (solid line) was implanted with $^4He^+$ (2·5 MeV, $4 \times 10^{16}$ ion/cm$^2$), and the second one with $^3He^+$ (2·5 MeV, $5 \times 10^{16}$ ion/cm$^2$). The $^3He^+$ isotope, because it is lighter, gives a longer range, and thus a wider waveguide and better mode confinement. The loss in the $^3He^+$ implanted guide is lower for most of the annealing temperatures. The spot size of the focused laser beam is ~10 μm, which is larger than the 6·4 μm width of the waveguide, resulting in a launch efficiency of less than 100%. Nevertheless, assuming a 100% launch efficiency the loss, at 200°C for the $^3He^+$ implant, is as low as 0·15 dB/cm. The lowest loss for the $^4He^+$ implanted waveguide is 0·3 dB/cm after the 300°C annealing temperature. For both of the cases we can assume a reproducibility error of ~±0·05 dB/cm. After it reaches its minimum value, for both waveguides, the loss increases due to destruction of the waveguide.

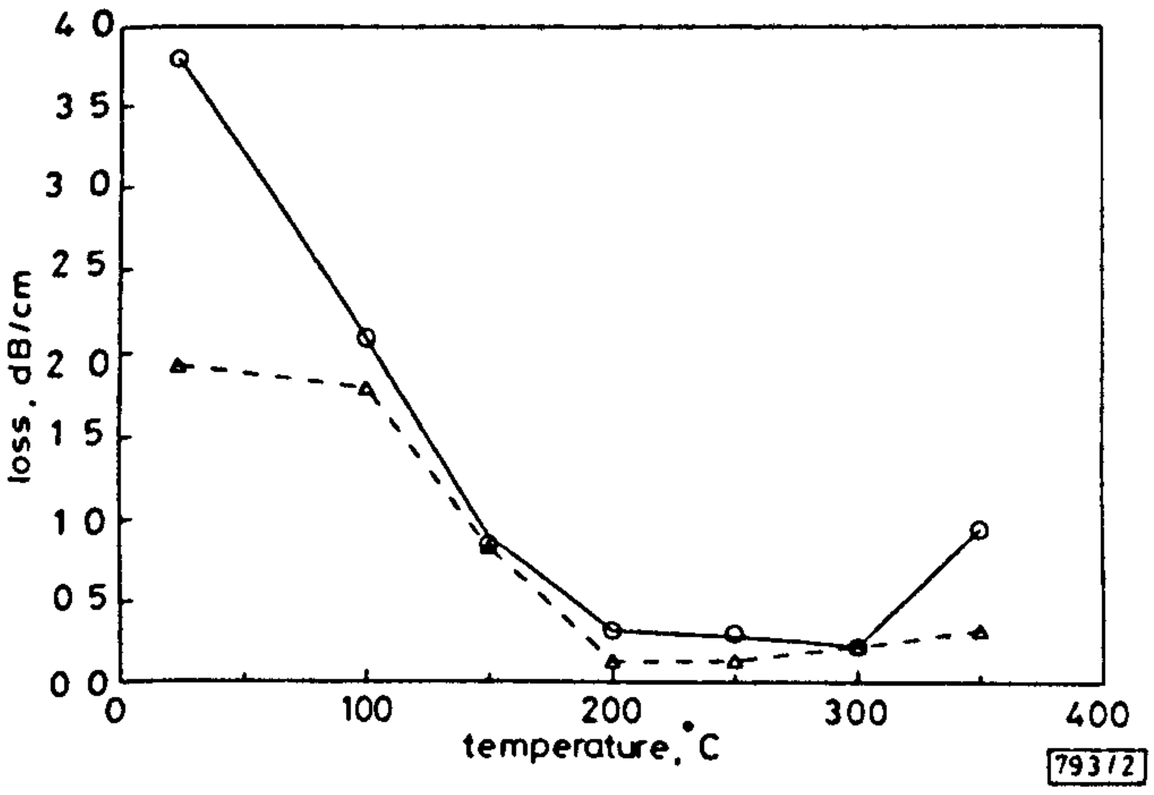



**Fig. 2** *Multimode losses for planar waveguides of $^4He^+$ and $^3He^+$ implanted lead germanate glass, after thermal annealing*

○ $^4He^+$
△ $^3He^+$

*Conclusions:* We have demonstrated the first example of an ion implanted waveguide in lead germanate glass. Furthermore, these guides have the lowest loss ever reported for He ion implanted waveguides in either glass or crystalline substrates. Further optimisation of the edge polishing and minimisation of the surface scattering should lead to even lower propagation losses. This should readily allow lower pumping thresholds and better waveguide laser performance for the rare earth doped lead germanate glass.

*26th January 1993*

G. Kakarantzas and P. D. Townsend (*School of Mathematical and Physical Sciences, University of Sussex, Brighton BN1 9QH, United Kingdom*)

J. Wang (*ORC, University of Southampton, Southampton SO9 5NH, United Kingdom*)

## References

1 CHANDLER, P.J., ZHANG, L., and TOWNSEND, P. D.: 'Optical waveguides by ion implantation', *Solid State Phenomena*, 1992, **27**, pp. 129–162

2 CHANDLER, P. J., and LAMA, F. I.: 'A new approach to the determination of planar waveguide profiles by means of a non-stationary mode index calculation', *Optica Acta*, 1986, **33**, pp. 127–143

3 TOWNSEND, P. D.: 'Optical effects of ion implantation', *Rep. Prog. Phys.*, 1987, **50**, pp. 501–558

4 KAKARANTZAS, G., ZHANG, L., and TOWNSEND, P. D.: 'Ion implanted waveguides in laser glasses'. E-MRS Symp. Proc. 29, Strasbourg, 1991, pp. 97–102

# INTERPOLATOR FILTER STRUCTURE FOR ASYNCHRONOUS TIMING RECOVERY LOOPS

D. Verdin and T. C. Tozer

*Indexing terms: Synchronisation, Digital signal processing*

A novel structure for the interpolation filter used as a timing-correction element in asynchronous timing recovery loops is introduced. Multirate techniques are employed in an algorithm to institute both bulk and fractional delays in the loop. The performance of the new algorithm is illustrated by generation of the S curves for a two-point NDA tracker and a four-point DD tracker.

*Introduction:* The generic asynchronous timing-recovery loop is shown in Fig. 1. The input analogue waveform is sampled to give $B$ samples per symbol although the phase of the sam-

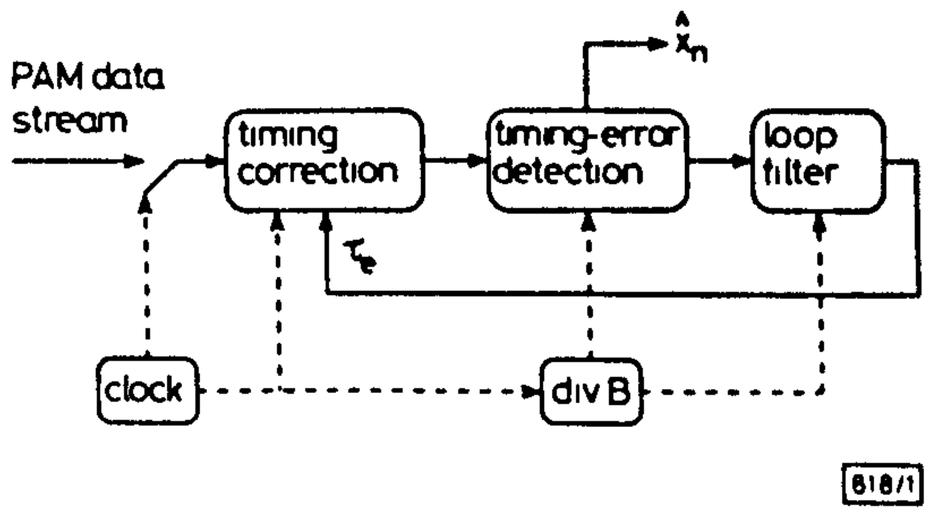



**Fig. 1** *Generic model of asynchronous timing-recovery loop*

pling clock may be such that the optimum sampling points are missed. The timing correction block interpolates the correct sampling points according to the fed-back error signal from the error-detection block. The error signal is generated and filtered at the symbol rate.

Examples of timing-error detectors for binary-antipodal filtered data are as follows noting the number of samples per symbol that the scheme requires.

(i) the Meuller and Müller algorithm [1], 1*s/s*, decision directed:

$$e_r(n) = \hat{x}(n)x(n-1) + x(n)\hat{x}(n-1) \tag{1}$$

(ii) the Gardner algorithm [2], 2*s/s*, nondata aided:

$$e_r(n) = \{x(n) - x(n-2)\}x(n-1) \tag{2}$$

(iii) four point algorithm [3], 4*s/s*, decision directed:

$$e_r(n) = \{x(n-1) - x(n-3)\}\hat{x}(n-2) + \alpha\{x(n) - x(n-4)\}\hat{x}(n-2) \tag{3}$$

where $0 \leq \alpha \leq 1$.

In the above the $x(.)$ represent the sampled PAM data stream and the $\hat{x}(.)$ represent receiver decisions.

For timing recovery, the function of the timing-correction block is to institute a delay of up to $B$ samples.

*Fractional delay using multirate polyphase filter:* It is well known that we can achieve fractional delays by using multirate techniques [4]. The $x_n$ samples out of the ADC are upsampled by a factor $U$ and then filtered by an image-rejection or interpolation filter to produce $B \times U$ sample points per symbol. This sequence can then be delayed to choose those points closer to the optimum sampling points and then downsampled by a factor $U$. If $M$ is the number of taps in the interpolation filter and we arrange it such that $M = U \times K$ then the process of upsampling and interpolation by a factor $U$ can be represented as

$$\begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-U} \end{bmatrix} = \begin{bmatrix} h_0 & h_U & \cdots & h_{(K-1)U} \\ h_1 & h_{U+1} & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ h_{U-1} & h_{2U-1} & \cdots & h_{KU-1} \end{bmatrix} \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-U} \end{bmatrix} = \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{U-1} \end{bmatrix} \times \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-U} \end{bmatrix} \tag{4}$$

In eqn. 4 the $y_n$ represent the up-sampled and interpolated sequence, the $h_n$ are the $M$-tap coefficients of the interpolation filter and the $P_n$ are the $K$-tap polyphase filter sections. The function of the polyphase filter as a fractional delay element is to select one of the $U$ values of the $y_n$ for every $x_n$ input, that is, there is no sample rate change overall. This is equivalent to filtering the input $x_n$ with a changing polyphase filter section $P_l$ dependent on the fractional delay $l/U$ that is required. Such a switched-coefficient filter would then be time-variant.

*Complete algorithm including bulk delay:* Bulk delay can be included by simply extending the shift register containing the input $x_n$ by $B - 1$ and offsetting the filtering operation by the bulk delay required, $d$, with $0 \leq d < B$.

The complete algorithm is as follows:

(i) input the required delay, $\tau_e$

(ii) shift data, $x_n$, into $K + B - 1$ length register

(iii) reduce the input delay modulo $B$ by putting

$$m = \begin{cases} \lfloor \tau_e U \rfloor / U \bmod B + B & \text{if } \tau_e < 0 \\ \lfloor \tau_e U \rfloor / U \bmod B & \text{if } \tau_e \geq 0 \end{cases} \tag{5}$$

(iv) fix bulk delay $d$ as

$$d = B - 1 - \lfloor m \rfloor \tag{6}$$

(v) fix fractional delay $u$ as

$$u = (m - \lfloor m \rfloor) \times U \tag{7}$$

(vi) filter the data and output the interpolated sample

$$y_{n-u} = \sum_{i=0}^{K} P_{ui} x_{i+d} \tag{8}$$

*Discrete-time S curves for NDA two-point tracker and DD four-point tracker:* The S curve shows the expected output from the error-detection algorithm, $\varepsilon[e_r(n)]$, as a function of the delay.

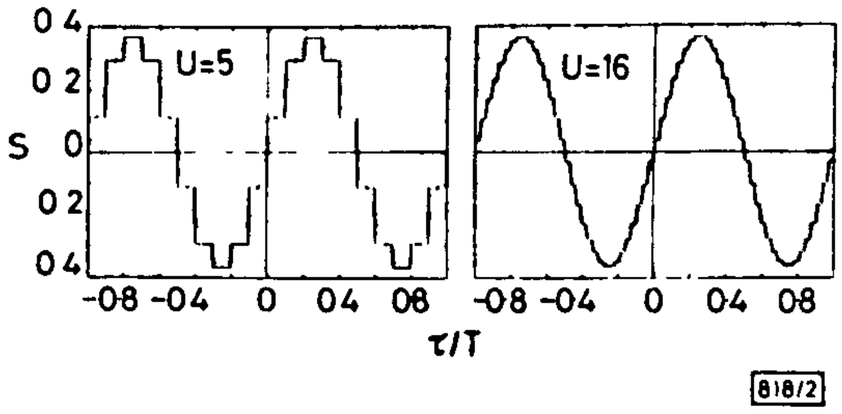



**Fig. 2** *Discrete S curves for two-point Gardner algorithm produced with polyphase delay element with different values of upsample*

The performance of the algorithm introduced here is shown in the S curves in Fig. 2 for the two point NDA Gardner algorithm and in Fig. 3 for the four point DD algorithm. The input binary-antipodal data stream is filtered with a raised-cosine filter with rolloff factor $\beta = 0{\cdot}8$ and the curves are generated by averaging over 5000 symbols. A raised-cosine filter was used as an approximation to the ideal interpolation filter [5]. The results in Fig. 2 compare the performance of the discrete-time delay element with the continuous-time theory in Reference 2 where it is shown that the S curve is given by a sinusoid as

$$\varepsilon[e_r(n)] = -\frac{4}{T}\sin\left(\frac{2\pi\tau}{T}\right) \times \int_0^{1/T} G(f)G\left(\frac{1}{T} - f\right)\sin(\pi f T)\, df \tag{9}$$

In eqn. 9 $T$ is the sample period and $G(f)$ is the frequency pulse corresponding to the PAM data stream. Note that the value of the integrand in eqn. 9 is in fact the gain of the S curve and is proportional to the excess bandwidth of $G(f)$.

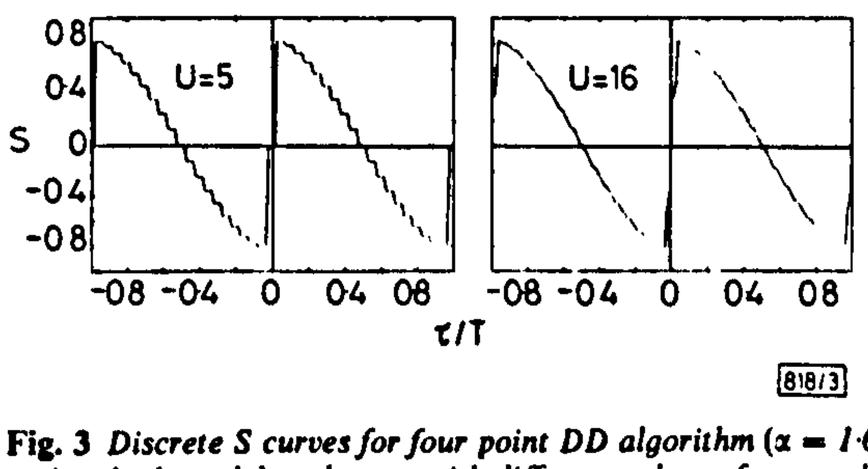



**Fig. 3** *Discrete S curves for four point DD algorithm ($\alpha = 1{\cdot}0$) produced with polyphase delay element with different values of upsample*

Note that as the upsample factor is increased from $U = 5$ to $U = 16$ the shape of the S curve approaches that of the ideal.

*Acknowledgments:* The authors would like to thank the Defence Research Agency, Malvern, for their continued support for this research.

*28th January 1993*

D. Verdin and T. C. Tozer (*Communications Research Group, Department of Electronics, University of York YO1 5DD, United Kingdom*)

## References

1 MUELLER, K. H., and MÜLLER, M.: 'Timing recovery in digital synchronous data receivers', *IEEE Trans.*, May 1976, **COM-14**, pp. 516–531

2 GARDNER, F. M.: 'A BPSK/QPSK timing-error detector for sampled receivers', *IEEE Trans.*, May 1986, **COM-24**, (5), pp. 423–429

3 GARDNER, F. M.: 'Demodulator reference recovery techniques suited for digital implementation'. European Space Agency Final Report ESTEC Contract 6847/86/NL/DG, 1988

4 LIM, J. S., and OPPENHEIM, A. V.: 'Advanced topics in signal processing' (Prentice-Hall, 1988)

5 WESOLOWSKI, K.: 'Computer generation of a slowly varying pseudorandom process', *IEE Proc. F*, June 1983, **130**, (4), pp. 314–316


# 13 Gbit/s Si BIPOLAR PREAMPLIFIER FOR OPTICAL FRONT ENDS


M. Neuhäuser, H.-M. Rein, H. Wernz and A. Felder



*Indexing terms: Integrated circuits, Amplifiers*



A preamplifier IC for a 10 Gbit/s optical fibre link was realised in a 0·8 $\mu$m silicon bipolar preproduction technology. Data rates up to 13 Gbit/s at high transimpedance (615 $\Omega$) and low noise (10·5 pA/$\sqrt{(Hz)}$ for 10 Gbit/s) were achieved. The frequency response can be individually optimised by adjusting a novel on-chip network via external potentiometers.


*Introduction:* In receivers for long-distance optical-fibre links a low-noise preamplifier is required which is directly driven by the photodiode. At high data rates this amplifier, which is usually of the transimpedance type, is one of the most critical circuits in the link because of the strong tradeoff between high bandwidth and high sensitivity (low noise). To the best of the authors' knowledge the highest data rate reported for silicon preamplifiers is 10 Gbit/s [1, 2]. The highest cutoff frequency of 15·5 GHz is reported in Reference 3 but without giving the maximum achievable data rate. In all these cases the cost of the high speed was the use of a low shunt feedback resistance (only 270 $\Omega$ in Reference 3), which degrades the noise behaviour (not given in References 1–3) and, moreover, causes a decrease in the transimpedance.

In this Letter a preamplifier for a 10 Gbit/s optical-fibre transmission system is developed. The circuit had to be designed in such a manner that, in contrast to Reference 1, no external (off-chip) equalising network is required. Moreover, the equivalent input noise current density should be low enough to allow the use of commercially available *pin* photodiodes at sufficient receiver sensitivity, and the transimpedance should be high enough to reduce the demands on the driven main amplifier, with respect to input sensitivity and voltage gain.

*Circuit concept and design:* Fig. 1 shows the circuit diagram of the preamplifier on the chip as well as the driving and loading circuitry for measuring the electrical characteristics. The photodiode (PD) is modelled by a simple network. The amplifier consists of a transimpedance stage with a comparatively high shunt feedback resistance $R_F = 615\,\Omega$, two emitter followers for impedance transformation and broadband level shifting, as well as an output transadmittance stage ($R_S = 20\,\Omega$) for driving matched 50 $\Omega$ transmission lines. The latter stage also allows the frequency response to be optimised by using peaking capacitors C1 and C2 as well as a special network (NW) for optimal adjustment of the frequency response (see below). The capacitors are realised by the emitter and collector junction capacitances of a transistor connected in parallel (emitter–collector short). C3 and C4 are external (off-chip) decoupling capacitors. An on-chip output resistor $R_Q = 100\,\Omega$ is provided to reduce double reflections and thus time jitter. Here, the low-frequency value of the total transimpedance $Z_{TI}$ equals the feedback resistance $R_F$. The strong increase of the equivalent input noise current density with increasing frequency can be reduced by properly adjusting the input bond inductance L1 (~1·5–2 nH) to the capacitance $C_{PD}$ of the photodiode (see Fig. 1).

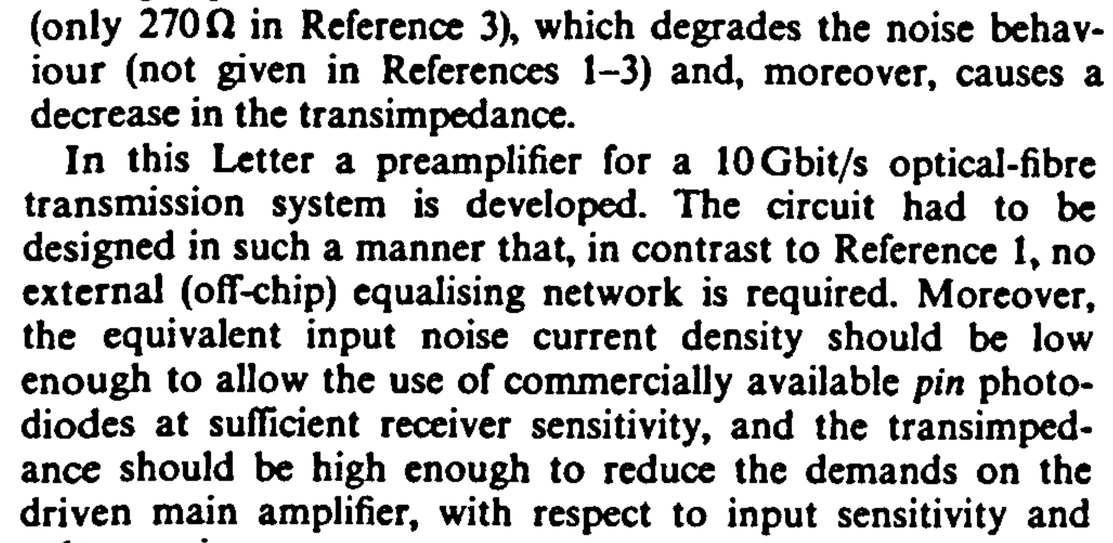



**Fig. 1** *Circuit diagram of preamplifier chip and driving circuitry*
Resistances are given in $\Omega$

The approach in Fig. 1 shows a negative shunt feedback $R_F$ only over a single stage. The advantages compared to the feedback over several stages were discussed in Reference 4. One main advantage is that a flat frequency response of both the magnitude and the phase delay, required for optimum eye diagrams, can be easily adjusted by peaking. For this, besides the constant capacitor C1 shunting the feedback resistor $R_S$, a second capacitor C2 is provided which is connected to the output of the novel network NW in Fig. 2, consisting of two cascaded emitter followers. It can be shown that a large area of complex values for the output impedance $Z_N$ can be covered by adjusting the collector currents of T3 and T4 (via the potentiometers P1 and P2). Moreover, C2 can be changed by varying the voltage across the corresponding *pn* junctions via P3. Thus the impedance of the series connection of C2 and $Z_N$ can easily be adjusted with respect to the desired frequency response of the amplifier. The cutoff frequency of the amplifier reaches the lower limit if the influence of C2 is minimised by switching off the collector current of T3. Such a possibility of simply influencing the frequency response of the front end after chip fabrication and mounting is highly recommended especially in the early stages of system and technology development in order to be able to compensate for several uncertainties in this stage.

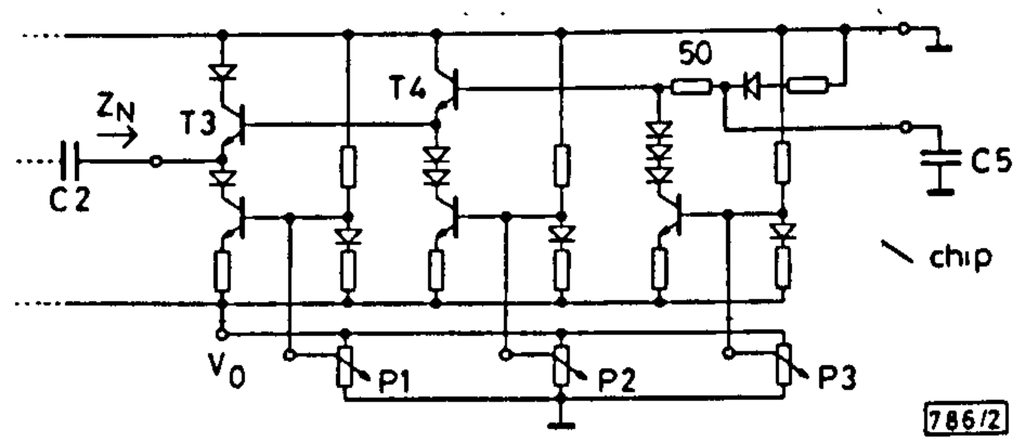



**Fig. 2** *Network NW in Fig. 1 for optimising the frequency response of the preamplifier via external potentiometers*
C5 is a decoupling capacitor

The preamplifier was designed on the basis of a selfaligning double-polysilicon preproduction technology of Siemens characterised by a 0·8 $\mu$m lithography and a nominal transit frequency of 25 GHz ($V_{CE} = 1$ V). The technology has been described in Reference 5 and typical transistor parameters are given, for example, in Reference 6. According to the intended application in a digital system, the main aim was to achieve maximum data rate at sufficient input sensitivity rather than maximum cutoff frequency. At the desired data rate the cutoff frequency should be chosen to be as low as possible, by properly adjusting the network NW, in order to increase the input sensitivity (due to the reduction of the noise bandwidth).

*Experimental results:* The presented experimental results were achieved with the first technological run and without needing any redesign. For the direct measurement of the preamplifier, the chip was mounted unpackaged on a thin film ceramic substrate (16 × 13 mm$^2$) using a conventional bonding technique. This test fixture contains all the external components

# Simulating Asynchronous Timing-Recovery Loops Using Multirate Techniques

D. Verdin and T.C. Tozer[1]

## Introduction

In this paper we study the application of multirate techniques to the simulation of asynchronous timing recovery loops. The statement of the problem is as follows; when simulating a receiver we will usually be operating with as small a number of samples per symbol as possible and generally these samples will not coincide with the optimum samples of the *underlying* analogue waveform. We have to *reconstruct* the optimum sample points from the samples available.

## Generic Loop Structure

The generic loop structure for the timing recovery loop is as shown in figure 1. We shall now discuss each of the loop components.

### Timing-Error Detection

The timing-error detectors are characterised by the number of samples per symbol and whether they are decision-directed (DD) or non-data aided (NDA). Some examples are as follows :

1. **2p-NDA Gardner [1] :**

$$e_t(n) = \{x(n) - x(n-2)\}\, x(n-1) \tag{1}$$

2. **1p-DD Mueller & Müller [2] :**

   The algorithm for this scheme for the simple case of a binary-antipodal data stream (BPSK) is given by

$$e_t(n) = \hat{x}(n)\, x(n-1) + x(n)\, \hat{x}(n-1) \tag{2}$$

3. **2p-DD Two-Point Tracker [1] :**

$$e_t(n) = \{x(n) - x(n-2)\}\, \hat{x}(n-1) \tag{3}$$

4. **2p-DD Crossing Tracker [1] :**

$$e_t(n) = \{\hat{x}(n) - \hat{x}(n-2)\}\, x(n-1) \tag{4}$$

In the design of high-speed modems one would generally require only 1 and at most 2 samples per symbol to achieve timing-recovery.

The S-curve [3] is the error detector characteristic. It shows the expected output, $\mathcal{E}(e_\tau(n))$, from the timing-phase error detector block as a function of the input delay $\tau$. The Gardner error-detector has a simple closed-form solution for the S-curve given by the following [1].

$$\mathcal{E}(e_\tau(n)) = -\frac{4}{T}\sin\left(\frac{2\pi\tau}{T}\right) \times \int_0^{1/T} G(f)G\left(\frac{1}{T} - f\right)\sin(\pi f T)\, df \tag{5}$$

Detector S-Curves for the 2p schemes are shown in figure 2. Note that the gain of the detectors is proportional to the excess-bandwidth of the modulation scheme (spectral efficiency ⇒ difficult synchronisation!).

---

[1]Authors are with the Communications Research Group, Department of Electronics, University of York, YO1 5DD

Note that if the error-detection scheme is based on $B$ samples per symbol then we will have to institute a bulk delay of up to $B$ samples as well as a fractional delay. It would be advantageous if the functions of bulk and fractional delay could be combined into the one functional block.

### The Polyphase Filter as a Fractional Delay Element

It is well known that one can achieve fractional delays by using multirate techniques [4]. Referring to figure 3, the $x_n$ samples out of the ADC are upsampled by a factor $U$ and then filtered by an image-rejection or interpolation filter to produce $B \times U$ sample points per symbol. This sequence can then be delayed to choose those points closer to the optimum sampling points and then downsampled by a factor $U$. If $M$ is the number of taps in the interpolation filter and we arrange it such that $M = U \times K$ then the process of up-sampling and interpolation by a factor $U$ can be represented as follows.

$$\begin{aligned}
\begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-U} \end{bmatrix} &= \begin{bmatrix} h_0 & h_U & \cdots & h_{(K-1)U} \\ h_1 & h_{U+1} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ h_{U-1} & h_{2U-1} & \cdots & h_{KU-1} \end{bmatrix} \times \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-K} \end{bmatrix} \\
&= \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{U-1} \end{bmatrix} \times \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-K} \end{bmatrix}
\end{aligned} \tag{6}$$

In (6) the $y_n$ represent the up-sampled and interpolated sequence, the $h_m$ are the $M$-tap coefficients of the interpolation filter and the $P_k$ are the $K$-tap polyphase filter sections. The function of the polyphase filter as a fractional delay element is to select one of the $U$ values of the $y_n$ for every $x_n$ input, that is, there is no sample rate change overall. This is equivalent to filtering the input $x_n$ with a changing polyphase filter section $P_l$ dependent on the fractional delay, $l/U$, that is required. Such a switched-coefficient filter would then be time-variant.

Figure 4 shows the polyphase filter sections for an upsample factor of $U = 7$ and $K = 9$. Note that for these particular values the filter sections are symmetrical – thus it would only be necessary to store $(U-1)/2$ sections and use reverse indexing if coefficient memory were scarce. Note also, that the middle section corresponding to $\delta = 0$ simply reproduces the signal that is present at the input (plus some bulk delay of $(U-1)/2$). The delays available with this filter cover the range $\delta = -3/7$ to $\delta = +3/7$ relative to the middle section. The effect of multiplying an input signal which has two samples per symbol with each of these polyphase filter sections is as shown in figure 5. In the figure the solid curve represents the *underlying* analogue waveform and the points represent the samples. The effect of instituting a fractional delay is to change the phase of the samples relative to the *underlying* analogue wave — it is important to note that the sampling instants remain the same. One can see by inspecting where the solid curve intersects the left-hand axis of each graph that the phase of the underlying analogue wave is moving from right to left as $\delta$ goes from $\delta = -2$ to $\delta = +2$.

Note that because of the phase of the samples in the input data stream the polyphase section that provides the optimum sampling points can easily be seen to be that corresponding to $\delta = 0$. One can appreciate that the higher the value of upsample the smaller the *granularity* of the delays that one can choose. Obviously, the penalty that has to be paid for this higher accuracy is that of higher memory usage. There are more involved algorithms which reduce the memory requirements but keep the same granularity by calculating polyphase sections as and when they are needed [5] but it felt that these place a heavier computing burden on the synchroniser than the method discussed here and might preclude real-time operation. Other schemes rely on polynomial interpolation.

### Choice of Interpolation Filter

The primary function of the interpolation filter in a multirate system is, obviously, to interpolate between upsampled points. An alternative and equivalent view from the frequency domain is that of image-rejection. There are other filtering functions that will need to be undertaken in the

receiver, such as data filtering, and it is possible to merge these transfer functions into that of the interpolation filter. However, it is noted in [5] that this is sub-optimal, that is, the best filter for data filtering will not be generally the best filter for interpolation.

It is found [6] that a good approximation to the ideal interpolation filter is that of the Raised Cosine filter. Referring back to (6) we can now express the filter coefficients mathematically.

$$h_n = h_{rc}\left(n - \frac{KU-1}{2}\right) \tag{7}$$

where

$$h_{rc}(t) = \frac{\sin\left(\frac{\pi t}{U}\right)}{\left(\frac{\pi t}{U}\right)} \times \frac{\cos\left(\frac{\pi\beta t}{U}\right)}{\left(1 - \frac{4\beta^2 t^2}{U^2}\right)} \tag{8}$$

and $\beta$ is the filter roll-off factor.

### Complete Algorithm Including Bulk Delay

It has been illustrated how a fractional delay of $l/U$ samples can be achieved by using a polyphase filter with upsample factor $U$. A simple procedure for implementing bulk and fractional delays taking into account the $B$-modulo nature of the delay has been described by the authors in [7].

## Block-Oriented Implementation

An example of a complete 2pDD timing-recovery loop for QPSK is as shown in figure 6. This has been implemented in the CAAD tool SPW and could easily be download to a DSP for system prototyping.

## Performance Results

Determination of the performance of these loops would be extremely difficult using analytical methods alone as they are highly nonlinear, recursive and time-varying and so we must rely on simulation methods.

The performance of the polyphase filter as a delay element is shown in the S-curves in figure 7 for the two point NDA Gardner algorithm. The input binary-antipodal data stream is filtered with a Raised-Cosine filter with roll-off factor $\beta = 0.8$ and the curves are generated by averaging over 5000 symbols. The results in figure 7 compare the performance of the discrete-time delay element with the continuous-time theory in (5).

Such delay quantisation in the loop has been found to have an important effect upon the acquisition and tracking performance of these loops.

# Conclusions

We have shown that when simulating symbol synchronisers one has to rely on multirate techniques. The implementation of a fractional delay-element based on a polyphase filter has been described. The effect of delay-quantisation upon the detector S-curve characteristic has also been demonstrated.

# Acknowledgements

The authors would like to thank the Defence Research Agency, Malvern, for their continued support for this research.

# References

[1] Floyd M. Gardner. A BPSK/QPSK timing-error detector for sampled receivers. *IEEE TRANSACTIONS ON COMMUNICATIONS*, 34(5):423–429, May 1986.

[2] K. H. Mueller and M. Müller. Timing recovery in digital synchronous data receivers. *IEEE TRANSACTIONS ON COMMUNICATIONS*, 14:516–531, May 1976.

[3] L. E. Franks. Carrier and bit synchronization in data communication — a tutorial review. *IEEE TRANSACTIONS ON COMMUNICATIONS*, 28(8):1107–1120, August 1980.

[4] Jae S. Lim and Alan Victor Oppenheim. *Advanced Topics in Signal Processing.* Prentice Hall, 1988.

[5] Floyd M. Gardner. Demodulator reference recovery techniques suited for digital implementation. *European Space Agency Final Report ESTEC Contract 6847/86/NL/DG*, 1988.

[6] Krzysztof Wesolowski. Computer generation of a slowly varying pseudorandom process. *IEE PROCEEDINGS*, 130(4):314–316, June 1983.

[7] D. Verdin and T.C. Tozer. Interpolator filter structure for asynchronous timing recovery loops. *ELECTRONICS LETTERS*, 29(5):490–492, March 1993.

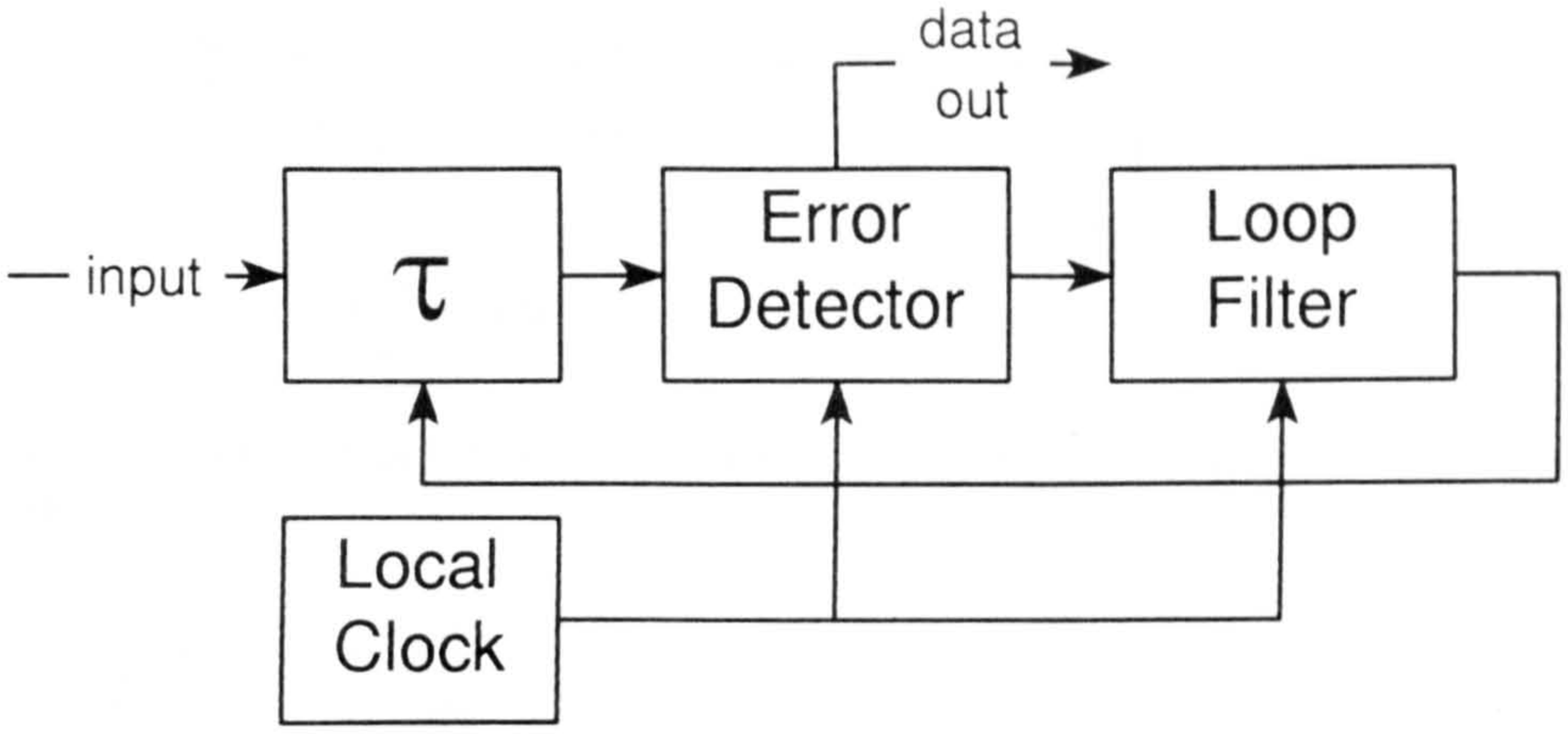
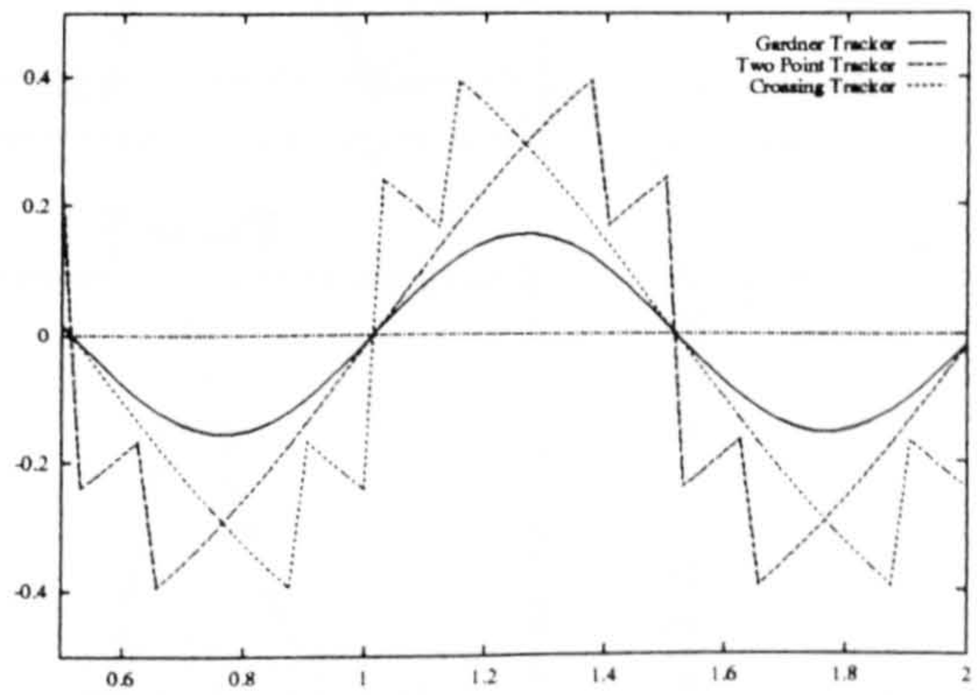
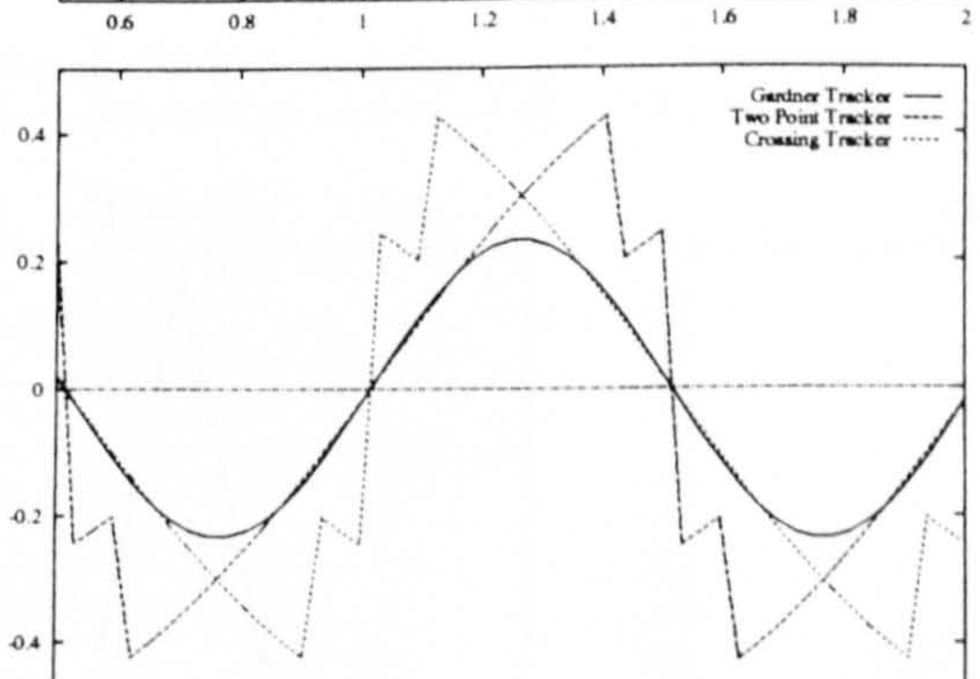


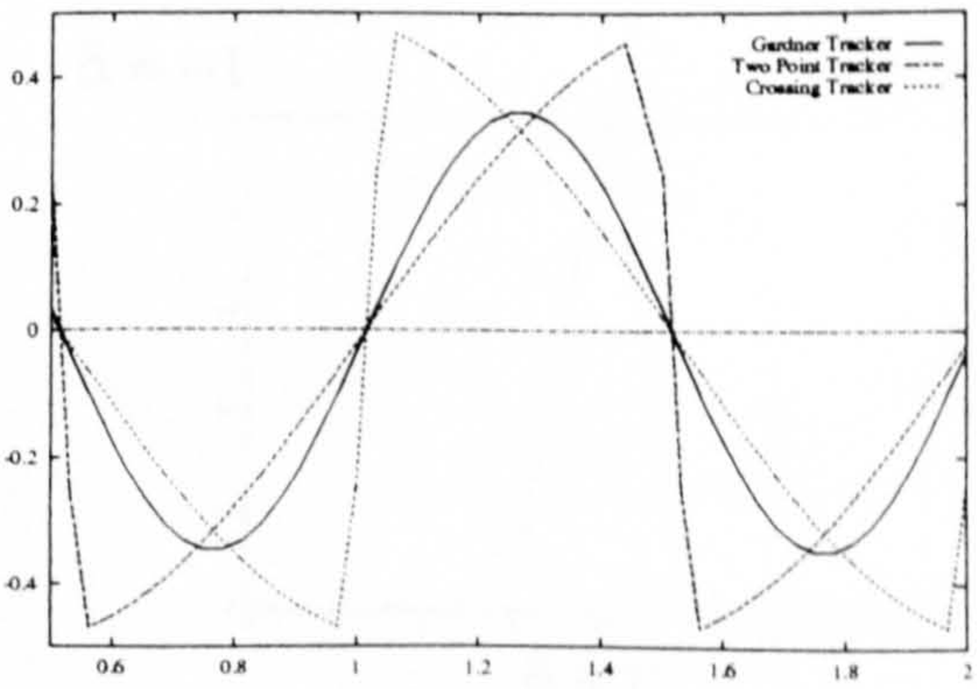

Figure 1: Generic Model of Timing Recovery Loop.




β = 0.3




β = 0.5




β = 0.8

Figure 2: Comparison of Two Point Tracker S Curves.

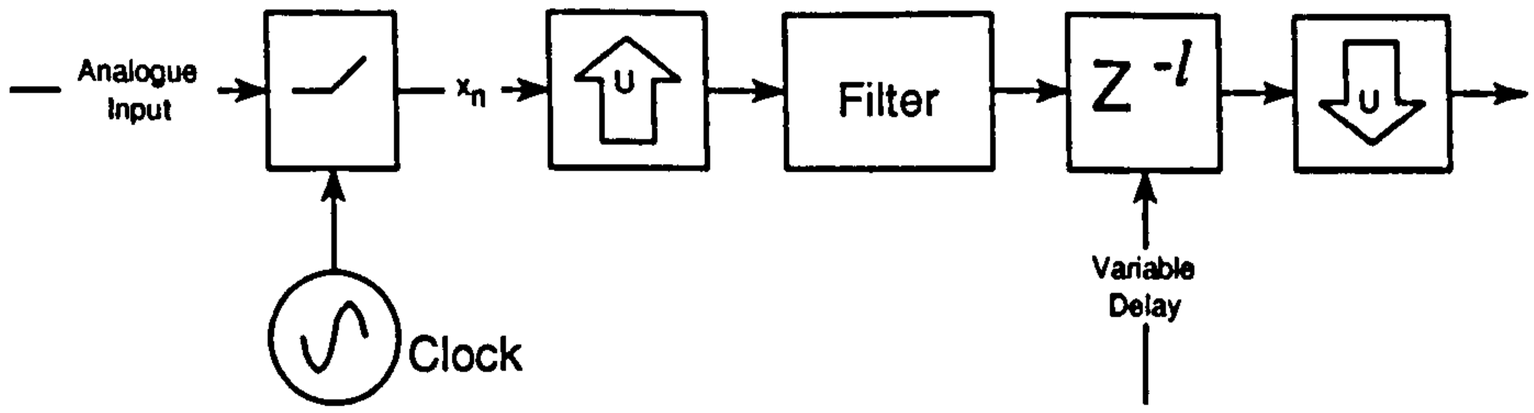



Figure 3: Schematic showing how a fraction-sample delay of $l/U$ can be achieved by using multirate techniques.

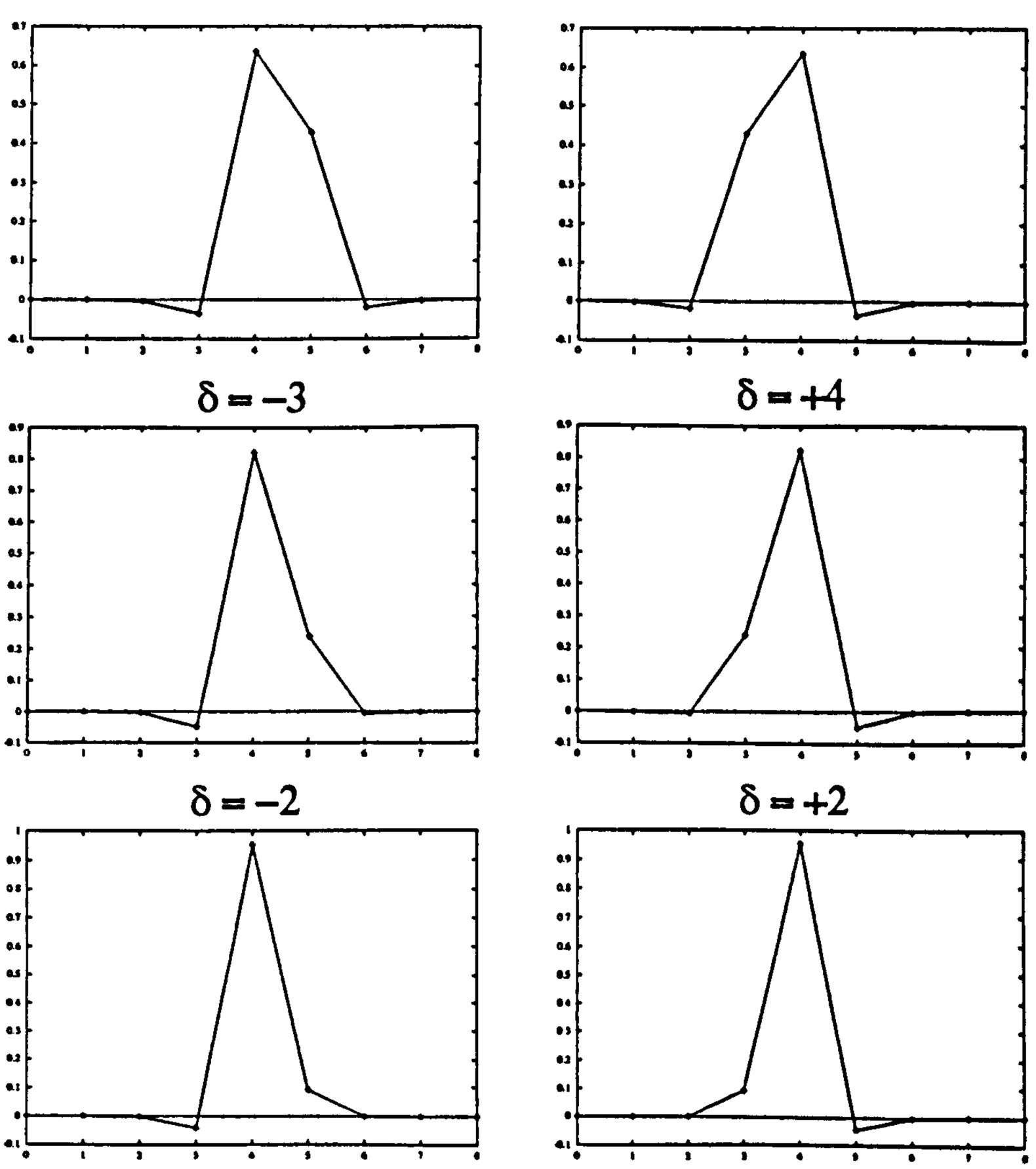



Figure 4: Polyphase Sections Impulse Responses.

U = 5, 2 samples/symbol

$\beta = 0.8$ in interpolation filter

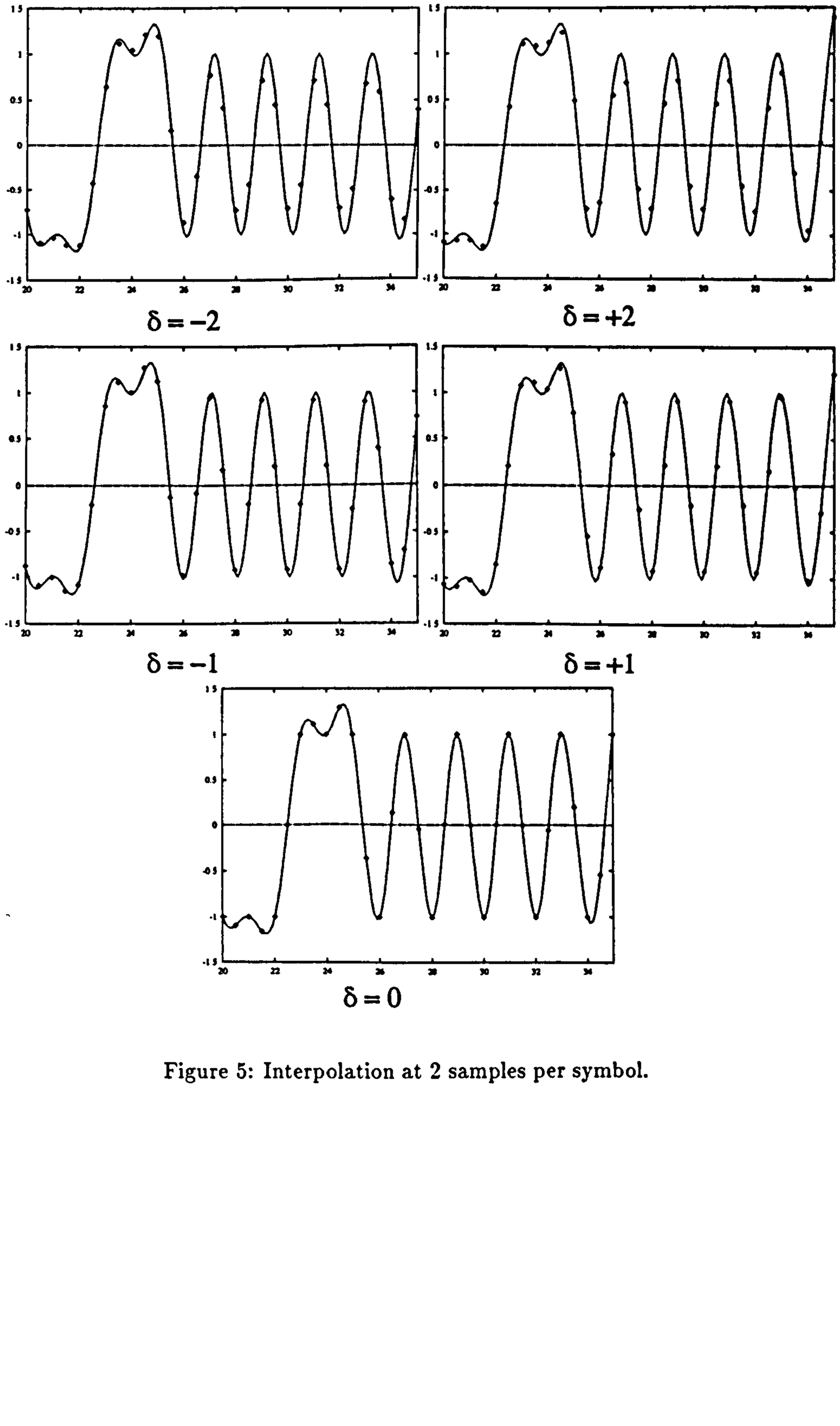



Figure 5: Interpolation at 2 samples per symbol.

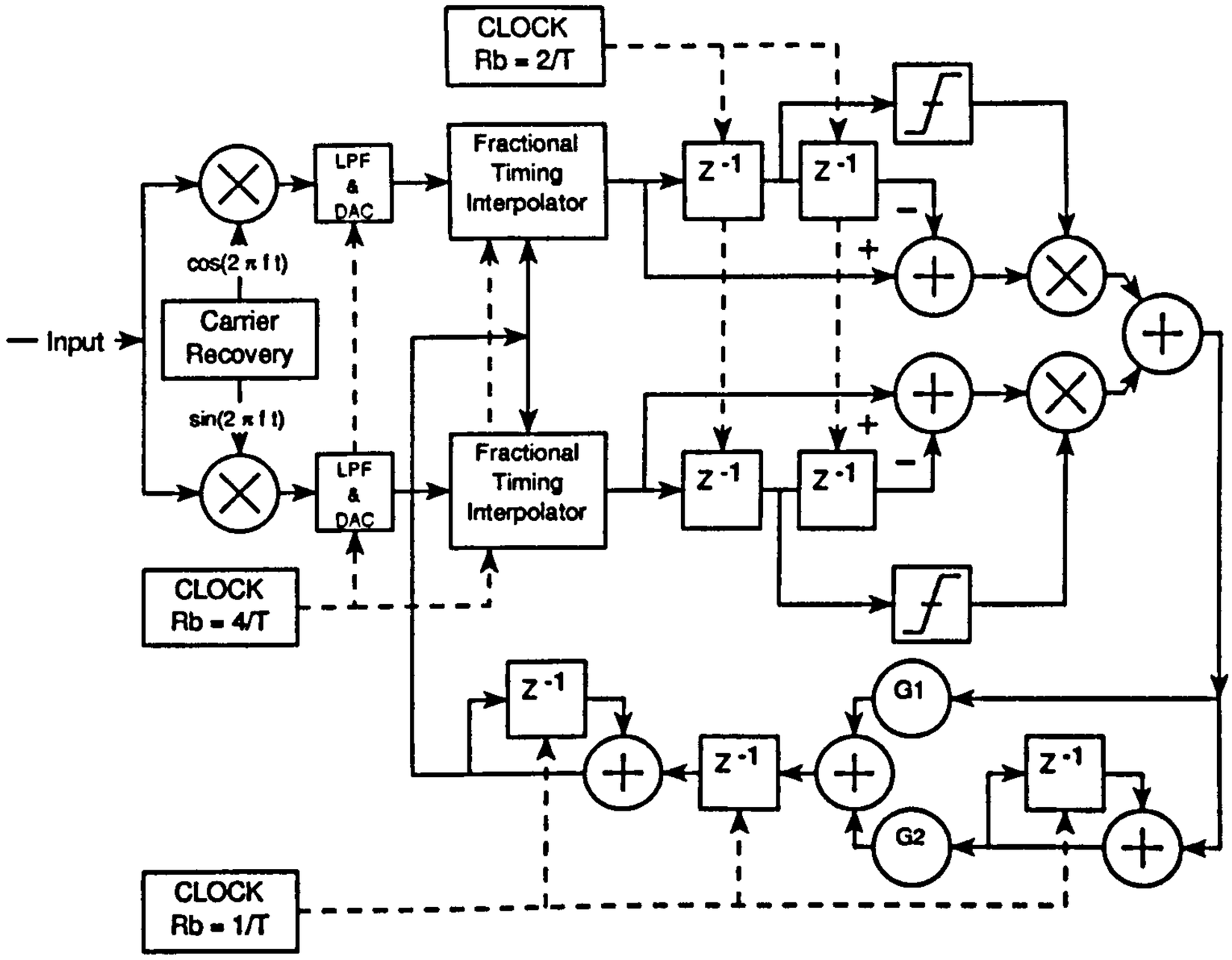



Figure 6: Schematic of 2-Point Decision-Directed Timing Recovery Loop

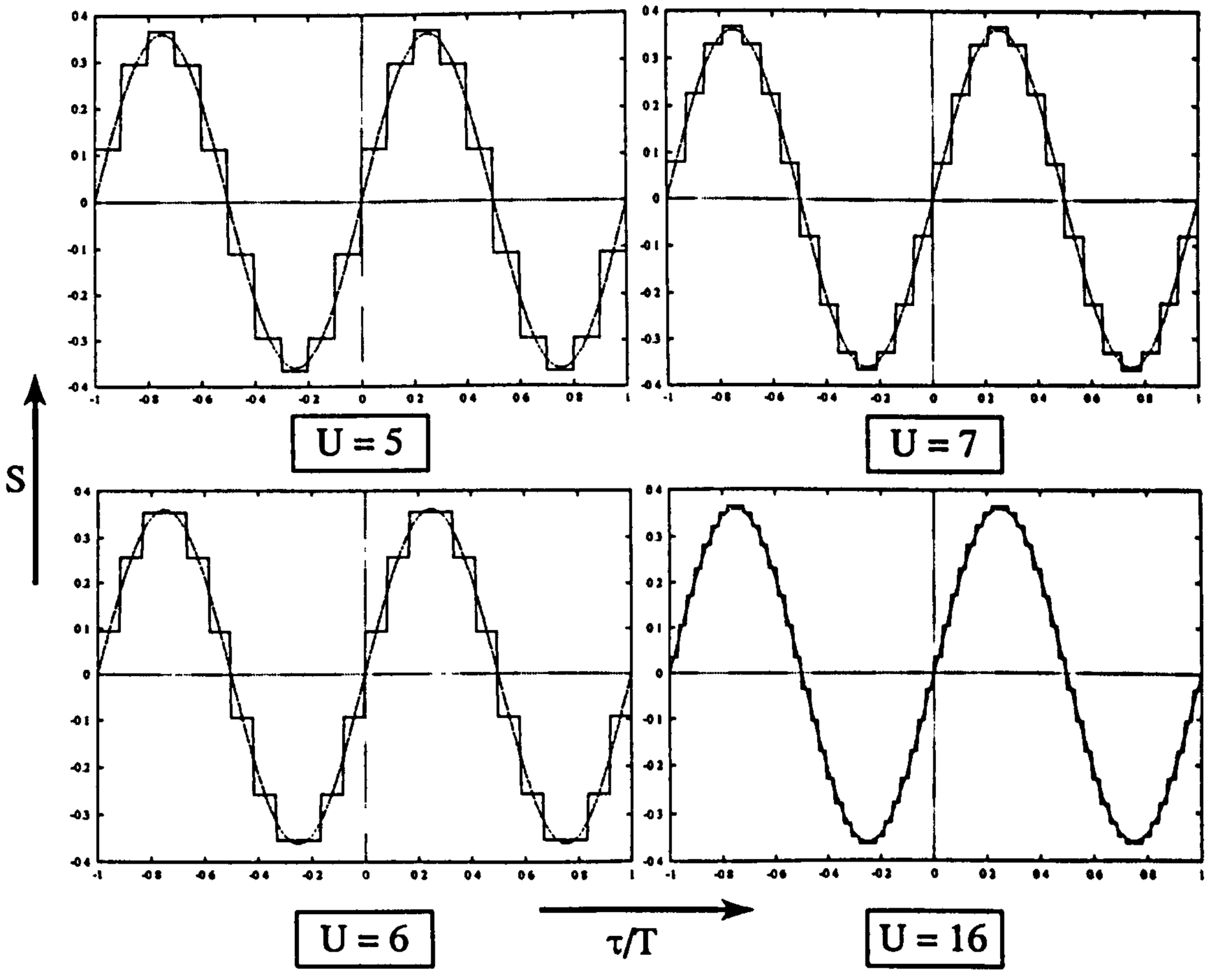



Figure 7: Effects of Discrete Interpolator on the S-Curves for the NDD Gardner Algorithm.

For given digital equipment working without digital equalisation on a given radio path during a given period of time, the correct value of outage time can be obtained using an iterative computation process adjusting the $S$ value derived from eqn. 1, characterising the digital equipment used, to the $S$ value derived from the equiprobability curve of $S$ and fade depths $A_F$, characterising the given radio path during the given period of time, as presented in Fig. 1. $D_c$, the excess of carrier to noise ratio above the flat fade margin $M_B$, is taken into account during the iterative computation process.

When linear time domain equalisation is used, the $S_E$ value of IBAD giving a BER equal to $\sim 10^{-3}$ is derived from the $S$ value without equalisation, computed by eqn. 1, using the following relation:

$$S_E = S \times k_E$$

where

$$k_E = k + \frac{2.46}{N}$$

and

$$\log_{10} k = 0.6636 \log_{10} N - 0.577$$

$N$ being the number of modulation levels. Eqn. 2 is valid for the following multilevel modulations: 8PSK, and 16QAM to 256QAM. It is valid for a seven tap equaliser.

For given digital equipment working with digital equalisation on a given radio path during a given period of time, the correct value of outage time can be obtained using an iterative computation process adjusting the $S_E$ value derived from eqns. 1 and 2 to the $S$ value derived from the equiprobability curve of $S$ and fade 2 depth $A_F$ taking account of $D_c$ (Fig. 1).

This method of calculating the outage time is valid for any transmission system subject to channel distortions on any radio path assuming that the autocorrelation function $\rho$ of fade depths at two fixed frequencies is known for this path.


*28 July 1993*
*Electronics Letters Online No: 19931256*

L. Martin (*France Telecom/CNET/PAB/SIIM - 38-40, rue du Général Leclerc - 92131 Issy les Moulineaux, France*)

B. Giraud and A. Bouidene (*France Telecom/CNET/LAB - Route de Trégastel - 22301 Lannion, France*)

## References

1 KOMAKI *et al.*, S.: 'Characteristics of a high capacity 16 Q AM digital radio system in multipath fading', *IEEE Trans. Comm.*, 1979, COM-27, (12), pp. 1857–1861
2 ITU – Radiocommunication Sector (CCIR): 'Propagation in non-ionized medium'. Recommendation 530-4 - Vol. RPN, Genève 1992, pp. 214-236
3 MARTIN, L., and PAN, X.Y.: 'Influence of radio and path characteristics and the IBPD measured on L.O.S. links'. IEEE Global Telecommunications Conf., 28-1 November-December 1988, (Hollywood, USA)
4 HIGUTI, I., and MORITA, K.: 'Diversity effects of propagation characteristics during multipath fading in microwave links', *Rev. Electrical Comm. Lab.*, 1982, 30, (3), pp. 544-551
5 MARTIN *et al.*, L.: 'La prévision de qualité de transmission sur les faisceaux hertziens numériques sans et avec diversité d'espace', L'Echo des Recherchers, September 1993

# Hangup in asynchronous timing recovery loops

D. Verdin and T.C. Tozer

*Indexing terms: Clock recovery, Phase shift keying*

The effects of hangup in digital asynchronous timing recovery loops are investigated. It is demonstrated that delay quantisation in the interpolation filter exacerbates the problem of hangup. It is also noted that the effects of hangup are worse for the non-data aided (NDA) loops than they are for the decision-directed (DD) loops. This may preclude the use of the former in TDMA applications.

*Introduction:* In asynchronous timing recovery loops a fractional sample delay is obtained by using a polyphase filter [1]. In effect, the delay that can be instituted in the timing recovery loop is quantised by the number of sections in the polyphase filter. A small number of polyphase filter sections will result in a 'staircase' approximation to the ideal S-curve characteristic. This quantisation of the S-curve characteristic for DD and NDA loops and the structure of the interpolation filter for such applications have been reported by the authors in [2]. In this Letter we investigate the effects of delay quantisation on acquisition in 2sample/symbol NDA and DD schemes.

The complete structure for the 2p-DD second-order timing recovery loop for a QPSK signal is as shown in Fig. 1. The 2p-NDA loop is as shown in Fig. 1 but with the threshold blocks removed. The interpolation filter would normally operate using four samples per symbol (to avoid signal distortion) whereas the error-detector scheme uses two samples per symbol providing one error signal per symbol which is then filtered at the baud rate. The clocks are shown separately here but in reality would be derived from the same source. Note that the loop filter consists of a proportional-plus-integral filter cascaded with a perfect integrator.

Analogous to standard phase-locked loops we write the loop filter parameters $G_1$ and $G_2$ in terms of the loop damping factor $\gamma$, the natural frequency $\omega_n$, and the timing-error detector gain $\kappa_\tau$, as

$$G_2 = \frac{\omega_n^2}{\kappa_\tau} \quad \text{and} \quad G_1 = \frac{2\gamma\omega_n}{\kappa_\tau} \qquad (1)$$

where the gain of the timing error detector for QPSK is given as

[3]

$$\kappa_\tau = \frac{8}{T} \times \int_0^{1/T} G(f)G(\frac{1}{T} - f)\sin(\pi f T)\,df \qquad (2)$$

where $G(f)$ is the frequency pulse of the modulation. Note therefore that the loop gain $\kappa_\tau$ is proportional to the excess bandwidth of the modulation scheme.

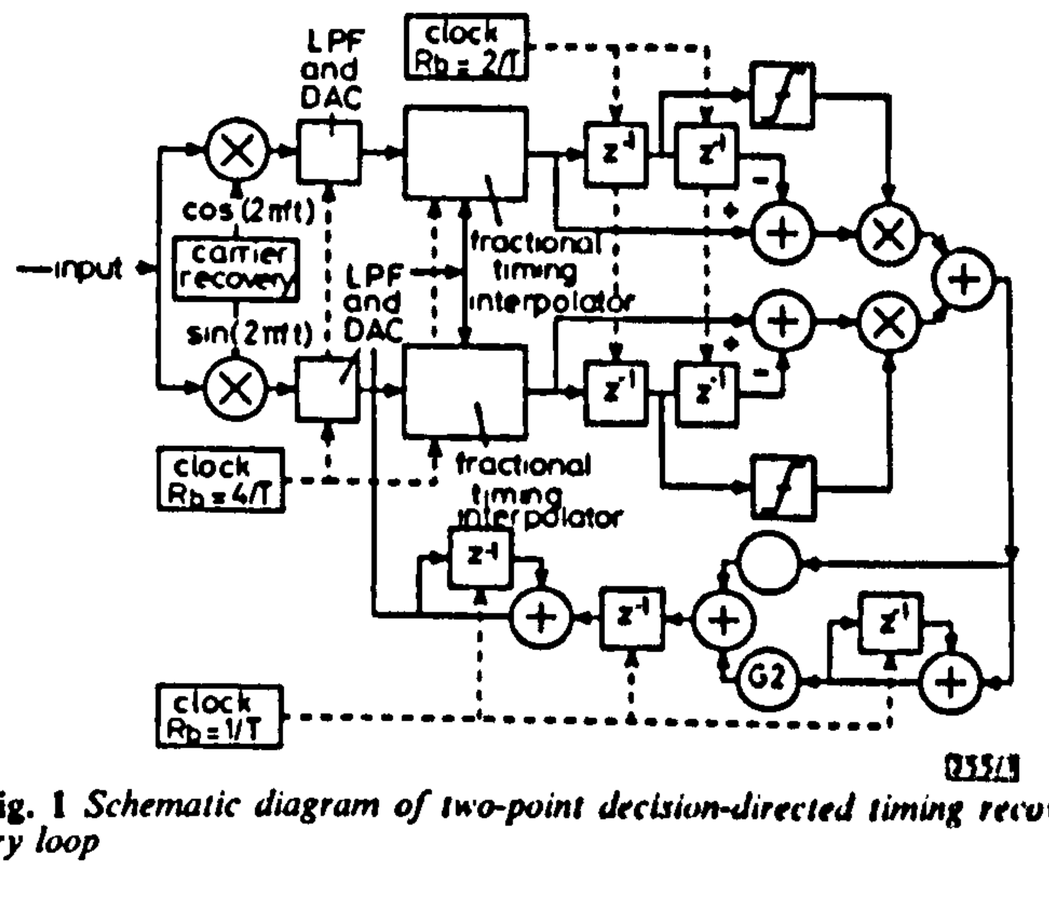



**Fig. 1** *Schematic diagram of two-point decision-directed timing recovery loop*

*Acquisition results:* Simulations were undertaken to determine the acquisition performance of the NDA and DD loops as a function of initial timing offset and interpolation filter upsample factor $U$. The results are shown in Fig. 2 for the NDA loop and Fig. 3 for the DD loop. The results show the error in the timing estimate as a function of time for initial offsets of $T/2$ and $T/4$, where $T$ is the symbol period. The polyphase filter upsample factor $U$, is also shown as a parameter. The loop parameters are $\gamma = 0.5$ and $\omega_n = 0.005$ and the input data are random QPSK filtered with a raised-cosine filter with rolloff factor $\beta = 0.8$. There is no additive noise in the system. The jitter that can be seen on the acquisition curves is due to 'self-noise' alone.

For an initial timing offset of $T/2$ we can see the tendency for the NDA loop to hang. This is because the loop is operating near the reverse-slopes unstable null of the detector S-curve characteris-

TEXT CUT OFF IN ORIGINAL

tic [4]. However, the problem is exacerbated by the delay quantisa-

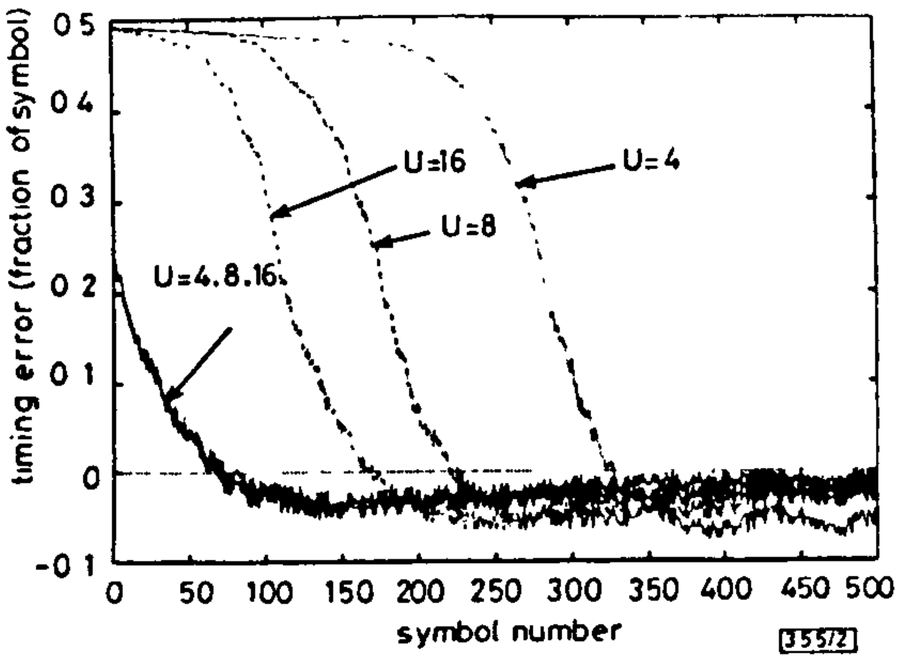



**Fig. 2** *Acquisition for 2p-NDA loop*

tion introduced by the polyphase filter. Decreasing the polyphase filter upsample factor $U$ degrades the acquisition performance of the NDA loop for large values of initial timing error. For smaller initial timing offsets the acquisition curves become exponential and are independent of delay quantisation as can be seen for an initial offset of $T/4$.

The effects of hangup are greatly alleviated by using a DD loop as is shown Fig. 3 although the effects are still evident for small values of $U$.

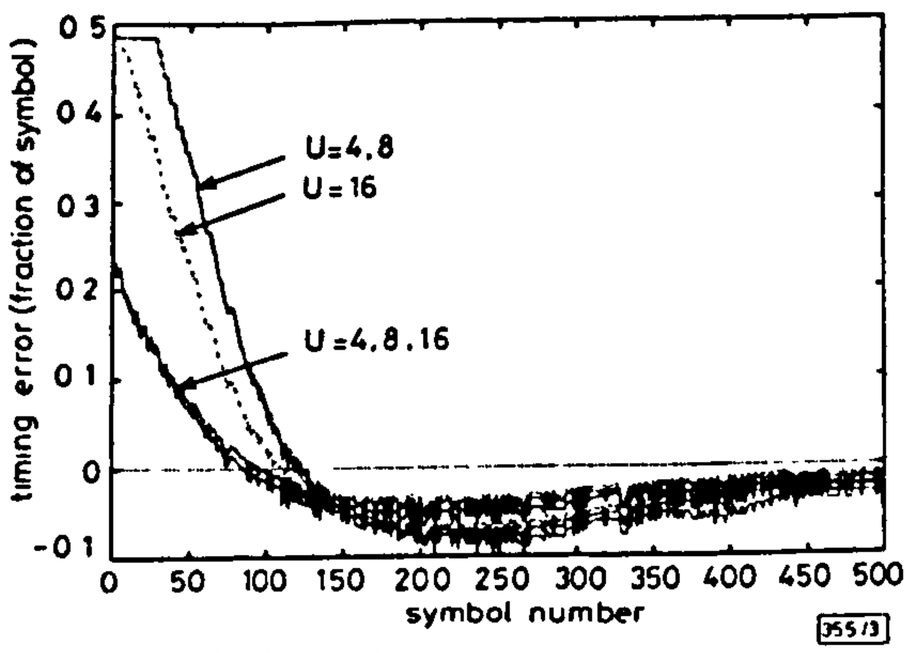



**Fig. 3** *Acquisition for 2p-DD loop*

*Conclusions:* We have demonstrated that the effects of hangup in NDA timing recovery loops is exacerbated by the delay quantisation introduced by the polyphase filter. It is apparent that DD loops offer better acquisition performance although still affected by delay quantisation. For burst-mode applications requiring fast acquisition it would be prudent to use a polyphase filter with a large number of sections or else use a delay element based on an alternative strategy such as polynomial interpolation.

*Acknowledgment:* The authors would like to thank the Defence Research Agency, Malvern, for their continued support for this research.


*1 September 1993*
*Electronics Letters Online No: 19931273*

D. Verdin and T. C. Tozer (*Communications Research Group, Department of Electronics, University of York YO1 5DD, United Kingdom*)

## References

1 GARDNER, F.M.: 'Demodulator reference recovery techniques suited for digital implementation'. European Space Agency Final Report ESTEC Contract 6847/86/NL/DG, 1988
2 VERDIN, D., and TOZER, T.C: 'Interpolator filter structure for asynchronous timing recovery loops', *Electron. Lett.*, 1993, **29**, (5), pp. 490–492
3 GARDNER, F.M: 'A BPSK/QPSK timing-error detector for sampled receivers', *IEEE Trans.*, 1986, **COM-34**, (5), pp. 423–429
4 GARDNER, F.M: 'Hangup in phase-lock loops', *IEEE Trans*, 1977, **COM-25**, pp. 1210–1214

# Hierarchical block-motion estimation using linear model-based postprocessing

K.-S. Seo and J.-K. Kim

*Indexing terms: Image processing, Motion estimation*

A two-hierarchy block-motion estimation method is proposed which uses a postprocessor based on a linear model of the block-motion vector field. Postprocessing in each hierarchy uses the linear model to correct false motion vectors estimated by blockmatching. By depending on a weighting parameter and window size for the correction, the global or local motion is effectively generated at each hierarchy.

*Introduction:* In video coding, accurate estimation of the motion vector field (MVF) is very important for both motion-compensated prediction (MCP) and interpolation (MCI) [1]. Block-motion estimation by blockmatching assumes basically independent translational motion of small blocks. It often generates inaccurate or false motion vectors, that are locally optimised regardless of object motion [1,2]. This is more severe for a real image with noise. The use of independently estimated block-motion vectors also increases the number of bits required for transmitting differential motion vectors, even though it makes some prediction gain in MCP. Furthermore, it degrades the interpolated image quality in MCI. Hierarchical block-motion estimation (HBME) [2,3] somewhat overcomes this false motion problem by estimating the global part of a whole motion at the first hierarchy and then the remaining local part at the later hierarchies. But the false motion problem still remains in the HBME method due to blockmatching based on the assumption of independent translational motion.

To solve this problem in block-motion estimation, we propose a new hierarchical method using a simple postprocessing of block-MVF based on a linear block-motion model. The motion model is a first-order polynomial that encompasses both rotation and change of scale, in addition to the conventional translation.

*New hierarchical block-motion estimation:* Fig. 1 shows the block-diagram of the proposed HBME method with two hierarchies. The first hierarchy measures global motion of longer range, and then the second hierarchy provides local refining motion of shorter range.

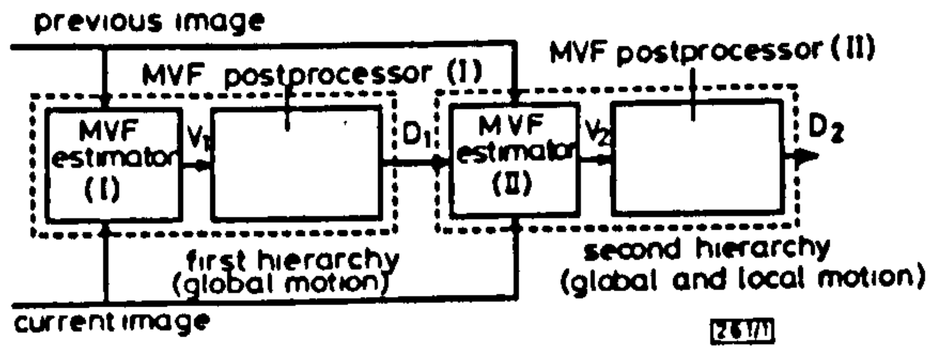



**Fig. 1** *Block diagram of proposed HBME method with two hierarchies*

Each hierarchy consists of an MVF estimator and an MVF post processor. At the first hierarchy, the MVF estimator computes the MVF of blocks of 16 × 16 pixels by using a blockmatching technique. The blockmatching results in local and inhomogeneous MVF even for one moving object due to small block size and noise. But we may assume that both current and neighbouring blocks are segments of an object and thus have a similar global motion. Under this assumption, the MVF postprocessor computes a global motion parameter set per window of $K \times L$ blocks by a linear model of the MVF. In the second hierarchy, the MVF estimator is basically the same as in the first hierarchy, but uses a shorter search range and the global motion $D_1$ of the first hierarchy as an initial estimate. Similar postprocessing can be applied to the second MVF $\{V_2\}$ in consideration of both global and local characteristics of the motion.

*MVF postprocessing by linear block-motion model:* MVF postprocessing corrects the false motion vectors within MVFs $\{V_1\}$ and $\{V_2\}$ by a linear block-motion model represented as follows:

## References

1 MOSES, R.L and LIU, D.: 'Optimal nonnegative definite approximation of estimated moving average covariance sequences', *IEEE Trans. Sig. Process.*, 1991, **39**, (9), pp. 2007–2015
2 FEYH, G., and MULLIS, C.T.: 'Moving average separation'. Proc. Int. Conf. Acoust., Speech, Signal Processing, 1988, pp. 2280–2283
3 JACKSON, L.B: 'Approximate factorization of unfactorable spectral models'. Proc. Int. Conf. Acoust., Speech, Signal Processing, 1985, pp. 324–326
4 CADZOW, J.A., and SUN, Y.: 'Sequences with positive semidefinite Fourier transforms', *IEEE Trans.*, 1986, **ASSP-34**, (6), pp. 1502–1510
5 STOICA, P., and MOSES, R.L.: 'On the unit circle problem: The Schur-Cohn procedure revisited', *Signal. Process.*, 1992, **26**, pp. 95–118


# Family of binary sequences with useful correlation properties


K. Byard



*Indexing terms: Binary sequences, Correlation*



A method for the synthesis of an infinite family of binary sequences with perfectly flat cross-correlation sidelobes is presented. The corresponding postprocessing arrays are similarly two-valued, and so the sequences may be particularly suited to applications in high noise environments.


*Introduction:* Sequences which possess periodic correlation functions having perfectly flat out-of-phase sidelobes are of interest in radio antenna design [1], communications [2] and image processing [3]. Such sequences, having length $v$ and denoted by $A(i)$, are characterised by cross-correlation functions $R(k)$ given by

$$R(k) = \sum_{i=0}^{v-1} A(i)G(i+k) = \begin{cases} C_0 & \text{if } k \equiv 0 \pmod v \\ C_1 & \text{if } k \not\equiv 0 \pmod v \end{cases} \quad (1)$$

where $G(i)$ is the postprocessing array, $i = 0, 1, \ldots, v-1$, and $i + k$ is taken modulo $v$. Of these, particular attention has been paid to binary (0, 1) sequences [3, 4], either because their use simplifies implementation or because the constraints of a given application render the use of multiple valued sequences impossible. Furthermore, noise considerations often require the postprocessing array $G$ to be similarly two-valued [5], possibly even unimodular (i.e. consisting entirely of +1 and −1, for example) [6]. We are concerned with such sequences and postprocessing arrays here, and present a simple method for the synthesis of an infinite family of these flat-sidelobe binary sequences (FSB sequences), along with the corresponding two-valued postprocessing arrays.

*Synthesis of the family:* FSB sequences $A(i)$ and postprocessing arrays $G(i)$ with the correlation property of eqn. 1 exist for all lengths $v$ of the form $v = 4n^2 + 1$, where $n$ is a positive integer, and can be synthesised as follows. For any given $v$ of the above form, let $A^*$ be the set of integers given by

$$A^* = \{a_j\} = \{(2n+1)j - n\} \quad (2)$$

and let $G^*$ be the set of integers given by

$$G^* \equiv \{2na_j\} \text{ (modulo } v) \quad (3)$$

where $j = 1, 2, \ldots, n$. Now construct the FSB sequence $A$, and postprocessing array $G$ as follows:

$$\left.\begin{aligned} A(i) &= 1 \\ A(v-i) &= 1 \end{aligned}\right\} \text{ if } i \in A^* \qquad A(i) = 0 \quad \text{otherwise} \quad (4)$$

$$\left.\begin{aligned} G(i) &= g_1 \\ G(v-i) &= g_1 \end{aligned}\right\} \text{ if } i \in G^* \qquad G(i) = g_0 \quad \text{otherwise} \quad (5)$$

where $i = 0, 1, \ldots, v-1$. Note that the sequence $A$ is binary, and the postprocessing array $G$ is two-valued, following the conditions stipulated above. The functions $A$ and $G$ possess the useful flat-sidelobe property of eqn. 1, with $C_0 = 2_{n\pi 0}$ and $C_1 = g_1 + (2n - 1)g_0$.

*Discussion:* FSB sequences can be constructed for any length of the form $v = 4n^2 + 1$, and an infinite family therefore exists. Because each FSB sequence in the family has exactly $2n$ 1s and $v - 2n$ 0s, their relative throughput (i.e. the ratio of 1s to the total number of elements in the sequence), is equal to $T = 2n/v = \sqrt{(v-1)/v}$. Thus for a given application, the required relative throughput may be selected by choosing an appropriate value of $v$. Furthermore, the binary complement of an FSB sequence is also an FSB sequence, obeying eqn. 1. This property enables the synthesis of an infinite family of complementary FSB sequences, each having relative throughput $T = 1 - \sqrt{(v-1)/v}$.

Depending on the requirements of a particular application, the postprocessing array values $(g_0, g_1)$ can be selected accordingly; for example a unimodular postprocessing array is possible by setting $g_1 = -g_0$. In particular, *perfect* sequences are available, namely those which possess zero sidelobes ($C_1 = 0$) and an in-phase cross-correlation value equal to the number of 1s in the sequence ($C_0 = 2n$) [3]. These are possible by setting the postprocessing array values to $g_0 = 1$ and $g_1 = 1 - 2n$.



*9 September 1993*
*Electronics Letters Online No: 19931330*



K. Byard (*Department of Physics, The University of Southampton, Southampton SO9 5NH, United Kingdom*)



## References

1 GOLAY, M.J.E.: 'Point arrays having compact nonredundant autocorrelations', *J. Opt. Sec. Am.*, 1971, **61**, pp. 272-273
2 GOLOMB, S.W, (Ed.): 'Digital communications with space applications' (Prentice-Hall, Engelwood Cliffs, New Jersey, 1964)
3 BYARD, K.: 'Synthesis of binary arrays with perfect correlation properties - Coded aperture imaging', to be published in *Nucl. Instrum. Meth. Phys. Res. A*, 1993, **A336**
4 LÜKE, H.D., BÖMER, L., and ANTWEILER, M.: 'Perfect binary arrays', *Signal Process.*, 1989, **17**, pp. 69–80
5 FENIMORE, E.E., and CANNON, T.M.: 'Coded aperture imaging with uniformly redundant arrays', *Appl. Opt.*, 1978, **17**, pp. 337–347
6 GOTTESMAN, S.R., and FENIMORE, E.E.: 'New family of binary arrays for coded aperture imaging', *Appl. Opt.*, 1989, **28**, pp. 4344–4352


# Generating a fading process for the simulation of land-mobile radio communications


D. Verdin and T.C. Tozer



*Indexing terms: Mobile radio systems, Stochastic processes*



An appropriate design method for the correlation shaping filter used in the simulation of the narrowband land-mobile radio channel is shown to be that of a windowed-Bessel function. The coefficients are configured in a multirate filter, combining both the functions of noise shaping and image rejection, to allow the generation of a fading process with precise correlation properties with only moderate size convolutions at each iteration.


*Introduction:* It is well known that the Doppler power spectral density for the ideal LMR channel has the following response [1]:

$$W(f) = \begin{cases} \frac{1}{\pi\sqrt{f_d^2 - f^2}} & \text{if } |f| \le f_d \\ 0 & \text{elsewhere} \end{cases} \quad (1)$$

where $f$ is frequency and $f_d$ is the Doppler frequency.

In the simulation of the narrowband LMR channel we would use an equivalent complex baseband representation as shown in Fig. 1. The real and imaginary components of the fading process

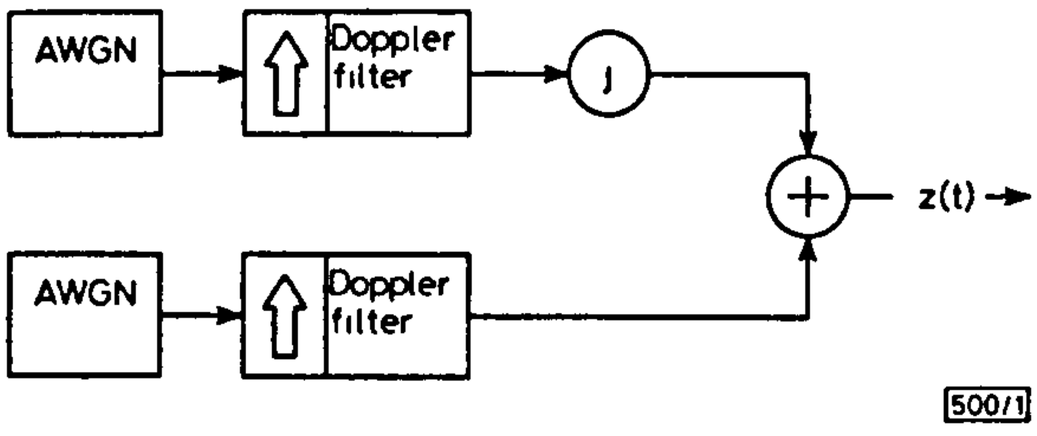



**Fig. 1** *Complex baseband model to generate fading process*

will be independent but identically distributed random processes generated by filtering white Gaussian noise with identical filters. The response of the individual filters is given as the square root of eqn. 1. It is worth deriving the closed-form expression for the ideal impulse response of these filters as this will then lead to an appropriate design procedure. The filter impulse response is found to be

$$x(t) = \int_{-f_d}^{f_d} \cos(2\pi f t)\sqrt{W(f)}\,dt = K\frac{J_{1/4}(2\pi f_d t)}{\sqrt[4]{t}} \qquad (2)$$

where $K$ is a time-independent constant.

It is also worth noting for the filter design that

$$\lim_{t\to 0}\frac{J_{1/4}(2\pi f_d t)}{\sqrt[4]{t}} = \frac{\sqrt[4]{f_d \pi}}{\Gamma(5/4)} \simeq 1.468813\sqrt[4]{f_d} \qquad (3)$$

Just as an approximation to the ideal lowpass filter is given as a windowed sinc, an approximation to the ideal filter for the simulation of the LMR channel is given as a windowed-Bessel function. Good routines that allow the computation of fractional-order Bessel functions are available [2].

*Design method:* A highly-correlated fading process is generated by configuring the cutoff frequency in the Doppler filters to be small. This can present problems for IIR filters as they can become unstable and direct FIR filtering would result in long, computationally-expensive convolutions that have to be performed on every iteration. Instead we use multirate-filtering techniques [3]. An $M$ tap filter is arranged by up-sampling the noise sources by a factor $U$ and performing a $K$-tap convolution on every iteration where $M = U \times K$. The functions of correlation shaping and image rejection are accomplished jointly provided $f_d < f_s/(2U)$, where $f_s$ is the sampling frequency.

The design procedure for implementing the filters in software is as follows:

(i) *Filter initialisation:* Define indices $u$ and $k$ such that

$$0 \le u \le U \quad \text{and} \quad 0 \le k \le K \qquad (4)$$

and the time index

$$t = \frac{1}{f_s}\left(kU + u - \frac{1}{2}(M-1)\right) \qquad (5)$$

then the $M$ tap filter $H_m$ has coefficients given by

$$H_{k+uU} = \begin{cases} W_{kU+u} J_{1/4}(2\pi f_d|t|)/\sqrt[4]{|t|} & \text{if } |t| > 0 \\ W_{kU+u}\,1.468813\sqrt[4]{f_d} & \text{if } t = 0 \end{cases} \qquad (6)$$

where $W_m$ are the window weighting coefficients.

The coefficients are normalised to render the correlation functions independent of upsampling by multiplying the filter taps by

$$a = \sqrt{\frac{U}{2\sum_{m=0}^{M-1} H_m^2}} \qquad (7)$$

Set filter index $i_u = U$.

(ii) *Run-time procedure:* If $i_u = U$ then read in the noise sample to the $K$-tap state register $x_k$ and set $i_u = 0$. Filter output $y$ is formed by performing a $K$-point convolution

$$y = \sum_{k=0}^{K-1} x_k H_{k+i_u K} \qquad (8)$$

Increment index $i_u$ by 1 and return to the start of run-time procedure.

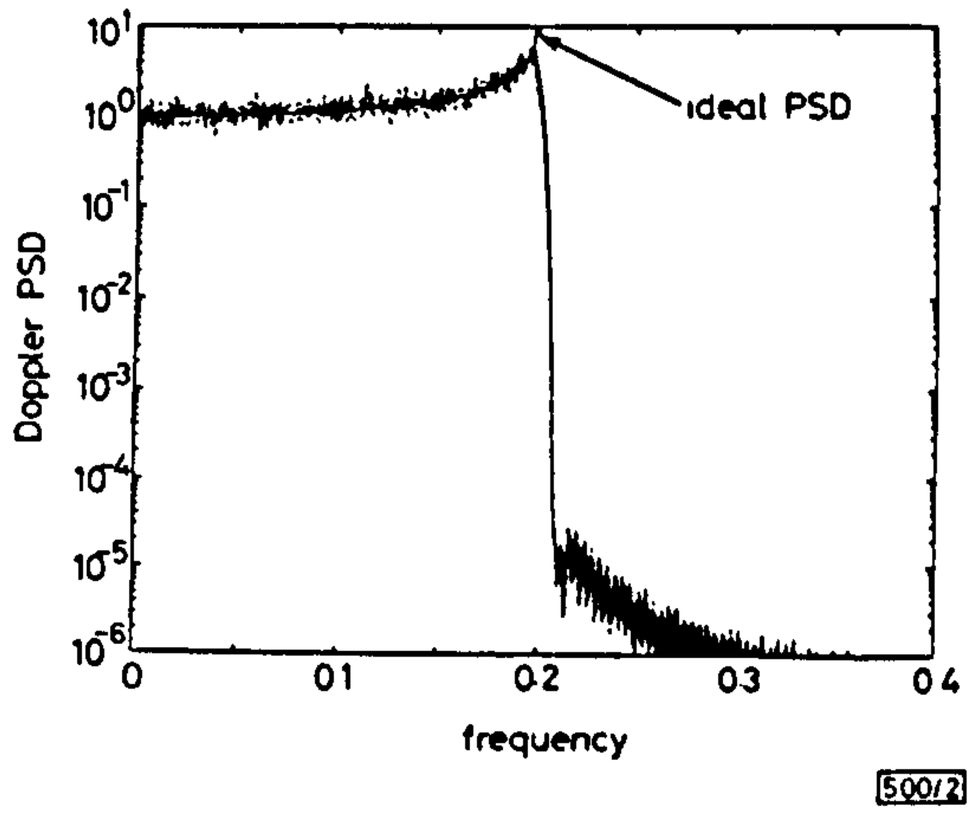



**Fig. 2** *Welch periodogram of fading signal*

Signal was generated with parameters $f_c = 0.2$, $U = 2$, $K = 128$, $f_s = 1.0$; curves also show 95% confidence limits on power spectrum

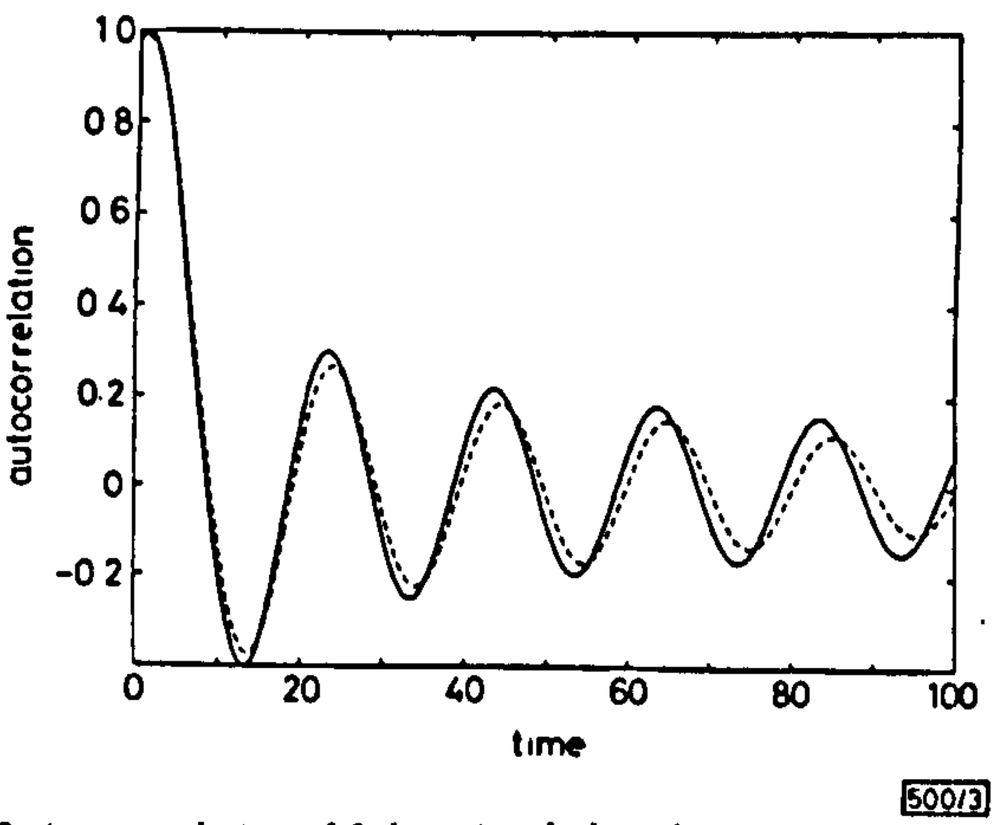



**Fig. 3** *Autocorrelation of fading signal plotted against theoretical ideal* $J_0(2\pi f_c \tau)$

Signal was generated with parameters $f_c = 0.05$, $U = 8$, $K = 128$, $f_s = 1.0$
– – – – autocorrelation
——— theoretical ideal

*Results:* Simulation results showing the correlation properties of the fading process generated by this method are shown in Fig/.2 for the periodogram and in Fig. 3 for the autocorrelation function. The simulations were run for 100000 points. The theoretical ideals are also plotted in these diagrams and the correspondence can be seen to be close.

*Acknowledgments:* The authors would like to thank the Defence Research Agency, Malvern, for their continued support for this research.


*Electronics Letters Online No: 19931338*

*16 September 1993*

D. Verdin and T. C. Tozer (*Communications Research Group, Department of Electronics, University of York, York YO1 5DD, United Kingdom*)

## References

1 CLARKE, R.H.: 'A statistical theory of mobile-radio reception', *Bell Syst. Tech. J.*, 1968, 47, pp. 957–1000
2 WILLIAM, H. Press *et al.*: 'Numerical recipes in C' (Cambridge University Press, 2nd Edn. 1992)
3 WESOLOWSKI, K.: 'Computer generation of a slowly varying pseudorandom process', *IEEE Proc.*, 1983, 130, (4), pp. 314–316

# Symbol timing recovery scheme tolerant to carrier phase error

D. Verdin and T.C. Tozer

*Indexing term: Clock recovery*

A novel symbol-timing recovery scheme is presented that is insensitive to carrier-phase error and is not susceptible to hangup at large values of timing error as is the case for the conventional two-point non-data aided (2p-NDA) scheme. The new scheme is applicable for symbol-timing recovery for *M*-ary PSK modulation formats.

*Introduction:* Decision-directed (DD) symbol-timing recovery schemes have faster acquisition time and lower timing jitter when corrupted by noise than non-data aided (NDA) schemes. The 2p-NDA scheme has a sinusoidal S-curve characteristic which can result in loop hangup at large values of timing error. However, the 2p-DD S-curve characteristic will be a function of carrier-phase error as well as timing error. At large values of carrier-phase error the 2p-DD scheme will become unusable as the constellation points are moved closer to the decision boundaries as is shown in Fig. 1. We here present a modification to the conventional 2p-NDA scheme which has an S-curve characteristic independent of carrier phase but will also avoid hangup at large values of timing error.

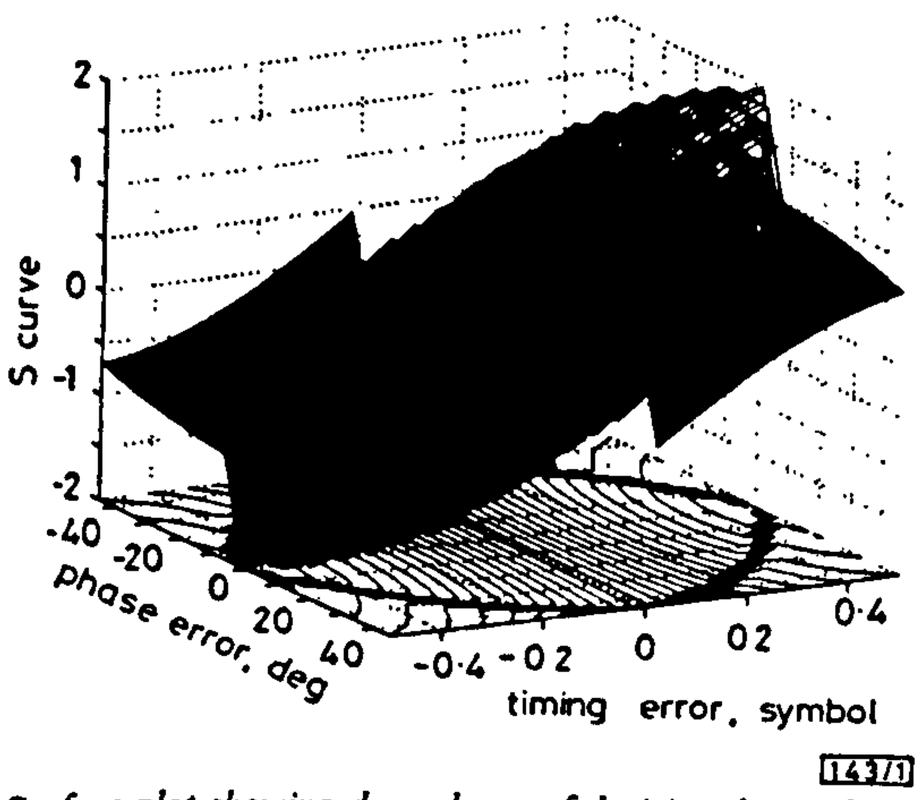



**Fig. 1** *Surface plot showing dependence of decision-directed characteristics on timing error and carrier-phase error*

Data are raised-cosine filtered ($\alpha = 1$) QPSK

*Two point symbol-timing error detectors:* Two samples per symbol from the in-phase $x_I(n)$ and quadrature $x_Q(n)$ components of the complex-baseband PAM signal are used to derive an error-signal. For the 2p-NDA scheme [1] this is given as

$$e_{nda} = \{x_I(n) - x_I(n-2)\}x_I(n-1) + \{x_Q(n) - x_Q(n-2)\}x_Q(n-1) \quad (1)$$

For the 2p-DD scheme [2] the error-signal is given as

$$e_{dd} = \{x_I(n) - x_I(n-2)\}\hat{x}_I(n-1) + \{x_Q(n) - x_Q(n-2)\}\hat{x}_Q(n-1) \quad (2)$$

where the $\hat{x}(n-1) \in \{-1, +1\}$ represent data decisions. The new 2p-NDA scheme is as follows:

$$e_{nda-\phi} = \{x_I(n) - x_I(n-2)\}\cos(\hat{\phi}) + \{x_Q(n) - x_Q(n-2)\}\sin(\hat{\phi}) \quad (3)$$

where

$$\hat{\phi} = \arctan\left(\frac{x_Q(n-1)}{x_I(n-1)}\right) \quad (4)$$

represents the instantaneous phase of the signal. Note that this does not imply that we have already achieved phase lock. We can write eqn. 3 in terms of eqn. 1 as follows:

$$e_{nda-\phi} = \frac{1}{|x(n-1)|}e_{nda} \quad (5)$$

*Statistical characterisation of symbol timing error detectors:* Fig. 2 shows the statistical performance of the three symbol timing-error detectors at high signal-to-noise and a carrier phase error of 40°. The modulation scheme used was raised-cosine filtered ($\alpha = 1$) QPSK and the curves formed by averaging over 5000 symbols.

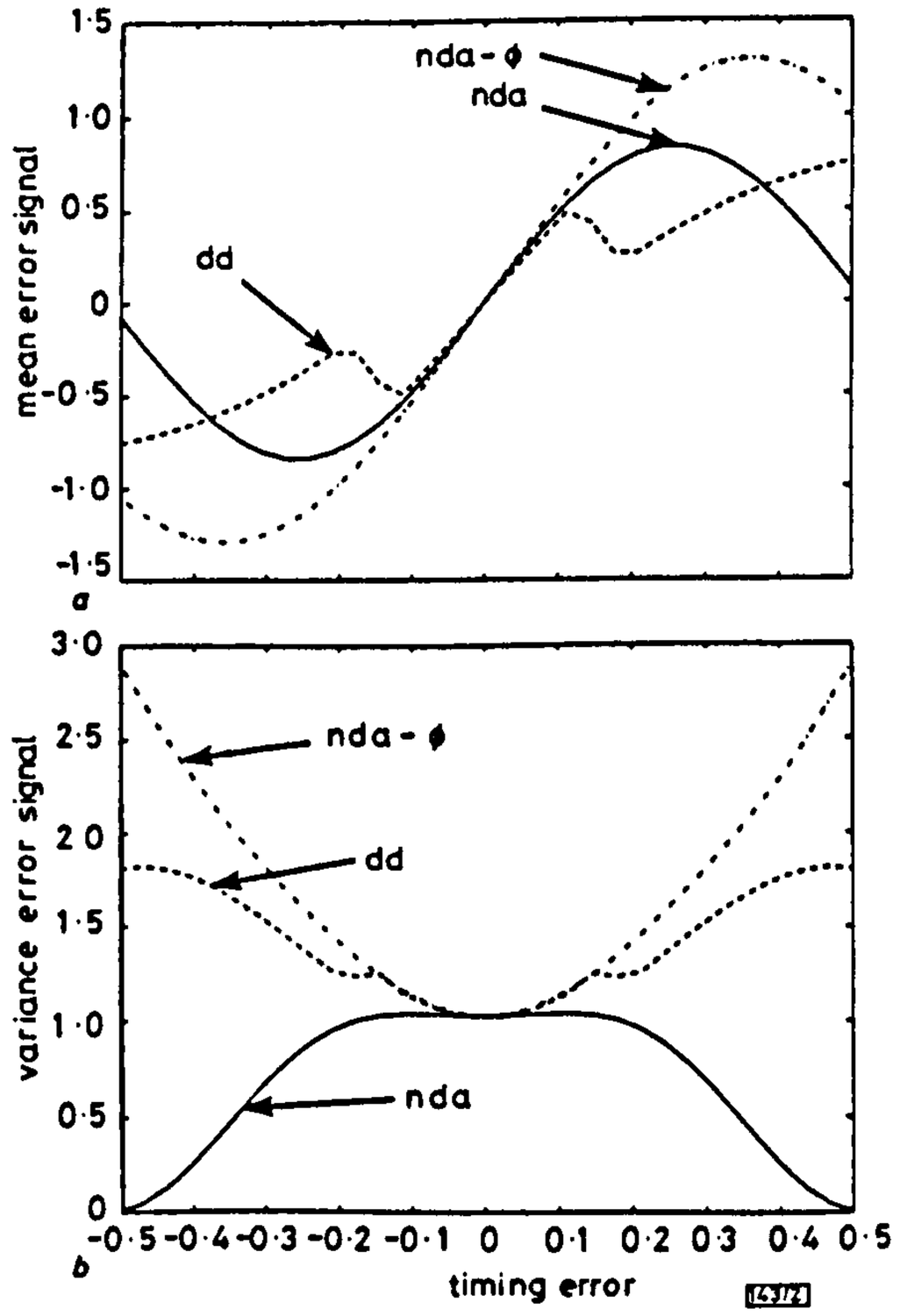



**Fig. 2** *Mean and variance of error signal for symbol timing-error detectors as function of timing error with high signal to noise*

Data are raised-cosine filtered ($\alpha = 1$) QPSK with static carrier-phase error of 40°

*a* Mean
*b* Variance

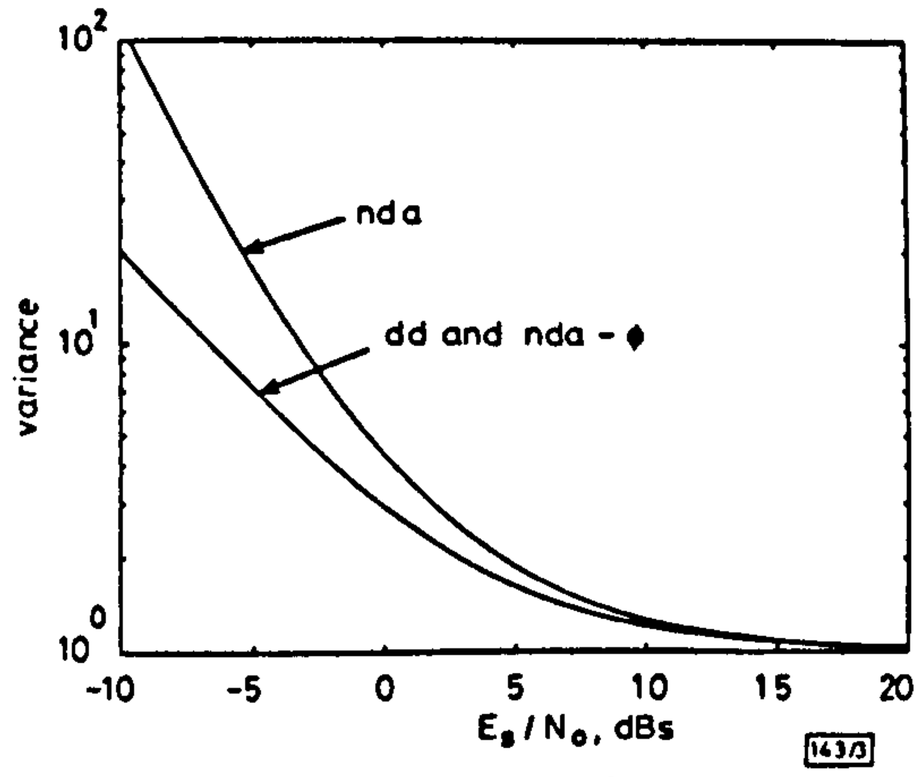



**Fig. 3** *Variance of error signal for symbol timing-error detectors as function of $E/N_o$*

Data are raised-cosine filtered ($\alpha = 1$) QPSK with no carrier-phase error

The two graphs show the mean and the variance of the error signal for each of the detectors. Note that the $e_{nda}$ scheme has small mean and variance when the timing error is $\tau = \pm T/2$ resulting in the possibility of loop hangup. The new scheme $e_{nda-\phi}$ has the desirable property of large mean and variance at $\tau = \pm T/2$ and this is independent of carrier phase. The decision-directed scheme has poor performance because the constellation points are close to the

decision boundaries. All the schemes have the same variance at $\tau = 0$.

Fig. 3 shows the open-loop variance of the three detectors as a function of signal-to-noise with timing error $\tau = 0$. It appears through simulation that the new scheme has the same performance as the fully decision-directed scheme.

*Acknowledgments:* The authors would like to thank the Defence Research Agency, Malvern, for their continued support for this research.


*16 November 1993*
*Electronics Letters Online No: 19940102*

D. Verdin and T. C. Tozer (*Communications Research Group, Department of Electronics, University of York, YO1 5DD, United Kingdom*)

## References

1 GARDNER, F.M.: 'A BPSK/QPSK timing-error detector for sampled receivers', *IEEE Trans.*, 1986, **COM-34**, (5), pp. 423–429
2 GARDNER, F.M.: 'Demodulator reference recovery techniques suited for digital implementation'. *European Space Agency Final Report ESTEC Contract 6847/86/NL/DG*, 1988

# Frequency domain (1kHz–40GHz) characterisation of thin films for multichip module packaging technology

W.-T. Liu, S. Cochrane, X.-M. Wu, P.K. Singh, X. Zhang, D.B. Knorr, J.F. McDonald, E.J. Rymaszewski, J.M. Borrego and T.-M. Lu

*Indexing terms: Thin film cappcitors, Dielectric thin films*

Parallel plate capacitors for the broadband dielectric characterisation of both high (amorphous $BaTiO_3$ and amorphous $TaO_x$) and low (parylene) dielectric constant thin films were fabricated at low temperature (< 200°C). The dielectric constant and loss tangent were dermined through the measurement of *C*, *G* and the *S* parameters of the capacitors. These thin film dielectrics exhibit no dispersion in the frequency range 1kHz – 40GHz.

*Introduction:* Thin film deposited multichip module technology (MCM-D) employs low dielectric constant thin films for signal line isolation to achieve maximum signal propagation speed and minimum crosstalk noise [1, 2]. Future designs using MCM-D might be improved by using a high dielectric constant thin film between the power and ground planes as a low inductance bypass capacitor in the power distribution system [3]. For both high and low dielectric systems, a nondispersive dielectric constant for operating frequencies into the gigahertz realm is desirable. In this Letter, we describe the method and results of a broadband characterisation for three thin film dielectrics which are potential candidates for MCM-D packages.

*Test vehicle fabrication and measurement:* Low temperature deposition techniques were used for the preparation of thin film dielectrics. Amorphous $BaTiO_3$, amorphous $TaO_x$, and parylene (poly-p-xylylene) thin films with a thickness of 0.3, 0,2, and 1.0μm respectively, were deposited using the techniques of reactive partially ionised beam [4], DC reactive sputtering [5], and vacuum vapour deposition [6], respectively. The deposition temperature is less than 60°C for $BaTiO_3$ and parylene, and less than 200°C for $TaO_x$. The dielectric constant $k'$ and loss tangent tanδ of the films were characterised in three domains of frequency using an HP4282A precision LCR meter (1kHz – 1MHz), an HP4195A network analyser (0.1 – 500MHz), and an HP8510C vector network analyser (45MHz – 40GHz).

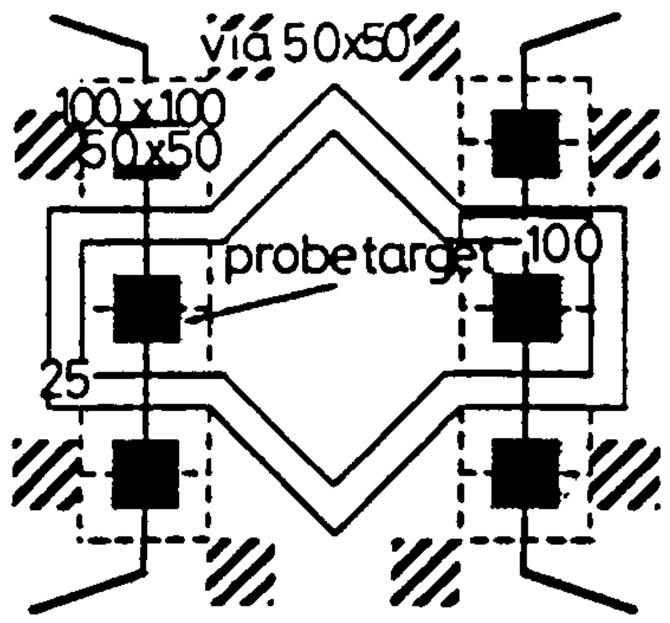



**Fig. 1** *Layout of coplanar capacitor and cross-sections of capacitors fabricated by two step mask process*

*a* Layout of coplanar capacitor; areas of capacitors are $5.1 \times 10^{-5}$, $5.1 \times 10^{-4}$, $5.1 \times 10^{-3}$, and $5.1 \times 10^{-2}$cm²; metal via size is $50 \times 50$μm²; dark solid lines denote positions of microwave probes
*b* Cross sections of capacitors fabricated by two mask process

Coplanar parallel plate test structures were used. The layout of the test structures embedded in a multipurpose test wafer is shown in Fig. 1*a*. The test vehicle consists of four capacitors with electrode areas covering four orders of magnitude. These structures allow dielectrics with $k'$ from 2.5 to 300 and thickness from 1000Å to 1μm to be characterised over the frequency range of interest. The cross-section of the test structure, fabricated by a two mask process, is shown in Fig. 1*b*. A 5μm Al bottom electrode was deposited on an oxidised silicon substrate, followed by the deposition of the dielectric. Thc dielectric was patterned by standard optical lithography and etched by aqueous HF, $CF_4/O_2$ plasma etching, and $O_2$ reactive ion etching for $BaTiO_3$, $TaO_x$, and parylene, respectively. The second layer of 5μm Al was then deposited, patterned, and etched to complete the processing. The highest temperature encountered by the wafers during the processing is during dielectric deposition. No anneal of the wafers was performed before measurement.

In the frequency range 1kHz – 1MHz, the capacitors were probed using standard microprobes. The $k'$ and tanδ of the dielectrics were determined through measurement of the capacitance *C*, shunt conductance *G*, the dielectric thickness, and the area of the capacitor plates. Parasitics presented by the test setup were modelled as a short piece of lossy transmission line. Values for the series and shunt elements of the lumped representation of the transmission line were determined by measuring a short and an open at every measurement frequency. The effects of the parasitics were removed from the raw data.

In the frequency range 1MHz – 40GHz, the structure was probed by a 50Ω Cascade Microtech microwave probe with 125μm pitch between ground and signal contact. The capacitors were modelled as a susceptance with a shunt conductance. Measurements of the scattering parameters $S_{11}$ (input reflection coefficient) and $S_{21}$ (insertion loss) were made on the capacitors and then were used to determine the dielectric constant $k'$, and the loss tangent tanδ.

*Dielectric characteristics:* Fig. 2*a* shows the plot of dielectric constant $k'$ and tanδ of the parylene thin film. No dispersion is observed in this range of frequencies. The material has an average dielectric constant $k'$ of 2.67 and tanδ of 0.003. These results are expected because the value of $k'$ is nearly equal to the square of the reactive index of the film (n = 1.63) at a wavelength of 6328Å. Hence, the dielectric constant of parylene is the result of optical