NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

A Study of a Symbol-Rate Timing Recovery Algorithm for a Digital ISDN U-Transceiver

by

Mirna Hage, B.A.Sc.

A thesis submitted to the
faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of
Master of Engineering

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
University of Ottawa
Ottawa, Ontario, Canada

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

ISBN 0-315-60614-2

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

A symbol rate timing recovery scheme suitable for ISDN transmission on digital subscriber loops is investigated. Four timing estimates based on this scheme are studied for steady-state and convergence behavior. Theoretical analysis and simulation results are included. The effect of the various system parameters such as channel distortion, noise and dead-zone threshold is examined for the four timing estimates. Their performance on realistic loops in a practical U-transceiver using 2B1Q line code is characterized and compared. The performance will be evaluated principally by the receiver convergence, steady-state timing jitter and noise margin allowed before convergence is lost.

Acknowledgements

I would like to thank all the people who have contributed in any small way to my studies. I wish to express my utmost thanks and appreciation to my supervisor Dr. Tyseer Aboulnasr for her guidance and encouragement, and Dr. Sami Aly for his interest, support and technical help. I wish to thank all the members of 1T23 department at Bell Northern Research, especially Babak Sayar for his invaluable expertise, Dr. Majid Taheri for his friendly advice and encouragement, Dr. Andrew Deczky, Paul Hung, Luis Viriato, Scott McClennon and Osama Baghat.

My deepest gratitude goes to Ken Yamazaki for having patiently listened to all my rambling pertaining to an infinity of issues ranging from the unpredictable CAPSIM/BLOSIM environment, the fascinating statistics and probabilities world, to my vacation plans and strong political views.

A special thanks goes to the CCSP group at North Carolina State University, namely Dr. Sasan Ardalan, Dr. Jim Faber and Ramin Nobakht for their cooperation and help with the CAPSIM/ BLOSIM issues.

Table of Contents

CHAPTER 1 ..... **1**
1.1.Introduction 1
1.2.Thesis Outline 5
1.3.Timing Recovery 6
1.4.Analog vs. Digital Timing Recovery 10
1.5.Criteria for Timing Recovery Performance 12
CHAPTER 2 ..... **14**
2.1.Survey of Timing Recovery Techniques 14
2.1.1.Continuous-time Techniques 14
2.1.2.Discrete-time Techniques 18
2.2.Synchronization Problem for Digital Subscriber Loops 22
2.3.Echo Cancellation Requirements on Timing Recovery 23
2.4.Channel Requirements on Timing Recovery 26
2.5.Equalization Requirements on Timing Recovery 27
2.6.Noise Requirements on Timing Recovery 27
2.7.Summary 28
CHAPTER 3 ..... **29**
3.1.Introduction 29
3.2.Framer and Analog Front End 31
3.3.Conversion and Input Sampling 32
3.4.Decimate/Interpolate Block 34
3.4.1.Integrator Block 34

3.4.2.COMB Filter 35
3.4.3.Interpolator Block 35
3.5.Receive Forward Filter 36
3.6.DFE/ARC Blocks 37
3.7.Automatic Threshold Quantizer 38
3.8.Echo Cancellation 38
3.8.1.Transversal EC 38
3.8.2.IIR EC 40
3.8.3.Memory EC 40
3.8.4.EC FIFO 40
3.9.Timing Recovery 41
3.10.Timing Generation 41
3.11.Summary 42

CHAPTER 4 44
4.1.Introduction 44
4.2.Review Of The Mueller and Müller Method For Timing Recovery 44

CHAPTER 5 54
5.1.Introduction 54
5.2.Timing Estimates Based On The M&M Algorithm For Timing Recovery 54
5.3.Timing Recovery Loop 58
5.4.Effect of The Finite Average Length 62
5.5.Effect of Channel Distortion And Additive Noise 71
5.6.Linearized Model For PLL 73
5.7.Timing Jitter Spectrum 79
5.8.Effect of The Dead-Zone Threshold 81
5.9.Summary 83

CHAPTER 6 84
6.1.Introduction 84
6.2.Description of Simulation Setup 84
6.3.Definition of Performance Criteria 91
6.4.Simulation for Channel Distortion and Additive Noise 92
6.4.1.Performance of Estimate #4 94

6.4.2.Results for Estimates #1,#2 and #3.....95
6.5.Effect of the DZT Factor.....99
6.6.Effect of the Initial Starting Phase.....104
6.7.Comparison of Convergence Time.....107
6.8.Summary.....111

CHAPTER 7.....112
7.1.Concluding Remarks.....112
7.2.Recommendation for Future Works.....114

References.....116

Appendices.....121

Appendix A.....122
A.1 Assumptions.....122
A.2 Estimate #1.....123
A.3 Estimate #2.....124
A.4 Estimate #3.....125
A.5 Estimate #4.....126

Appendix B.....128
B.1 Derivation of the Variance Expression:.....128
B.2 Variance of Estimate #1.....131
B.3 Variance of Estimate #2.....134
B.4 Variance of Estimate #3.....136
B.5 Variance of Estimate #4.....140

Appendix C.....143
C.1 Introduction.....143
C.2 Topology File.....145
C.3 Creating Custom Made Blocks.....148

List of Figures

Figure 1.1: ISDN Connections Structures 2
Figure 1.2: 4-Wire/2-Wire Transmission Techniques for DSL Applications 4
Figure 1.3: Deductive Timing Recovery 8
Figure 1.4: Inductive Timing Recovery System 9
Figure 1.5: Analog vs. Digital Timing Recovery 11
Figure 1.6: Eye Diagram for Binary Antipodal Signalling 13
Figure 2.1: Spectral-Line Timing Recovery Method 15
Figure 2.2: Sampled-Derivative Timing Recovery Method 16
Figure 2.3: Early-Late Gate Timing Recovery Method 17
Figure 2.4: Threshold Crossing Timing Recovery Method 17
Figure 2.5: The Wave-Difference Timing Recovery Method 19
Figure 2.6: State-Machine PLL Timing Recovery 21
Figure 2.7: Timing loop in DSL 23
Figure 2.8: Decision-Aided vs. Non-Decision-Aided Timing Recovery 26
Figure 3.1: ISDN U Transceiver Architecture 30
Figure 3.2: Framing and Analog Front End Processing 32
Figure 3.3: Sigma-Delta Modulator Structure 33
Figure 3.4: Decimation/Interpolation Blocks 34
Figure 3.5: Integrator Structure 35
Figure 3.6: Echo Canceller Structure 39
Figure 3.7: Timing Generation Block 42
Figure 4.1: Digital Receiver Architecture 45
Figure 5.1: Timing Function $f(\tau) = h_{-1}$ For B3, B4 and SL Loops 55

Figure 5.2: The Configurations of the B3, B4 and SL Loops 56
Figure 5.3: Pulse Response of the B3, B4 and SL loops..... 57
Figure 5.4: Timing Recovery Loop 59
Figure 5.5: Functionality of a Three Level Phase Quantizer..... 60
Figure 5.6: Functionality of the Timing Generation block..... 61
Figure 5.7: Simplified Receiver Configuration 63
Figure 5.8: Comparison of the Effect of TR_AVG on the Four Estimates, B4 loop..... 64
Figure 5.9: Comparison of the Mean Square Error for the Four Estimates..... 67
Figure 5.10: Closed Loop Form of Simplified Receiver..... 68
Figure 5.11: Performance of Estimates for Simplified Receiver for the B3 Loop..... 70
Figure 5.12: Plots for the Variance Expressions of the Timing Estimates 73
Figure 5.13: Linearized Model for Timing Recovery PLL..... 75
Figure 5.14: Approximation of the Loop filter for the Linearized PLL Model..... 78
Figure 5.15: Input Phase Jitter Characteristics 81
Figure 6.1: Receiver Setup for Simulation 85
Figure 6.2: Phase Behavior for Estimates #1,#2 and #3, for the B3 Loop..... 98
Figure 6.3: Phase Behavior for Estimates #1,#2 and #3 for the B4 Loop..... 103
Figure 6.4: Convergence Histograms for Estimates #1, #2 and #3 for the B3 Loop..... 109
Figure 6.5: Convergence Histograms for Estimates #1, #2 and #3 for the B4 Loop..... 110

List of Tables

Table 5.1: Four Timing Estimates Based on the Timing Function $f(\tau) = h_{-1}$ 58
Table 5.2: Expressions for the Variance of the Timing Estimates 58
Table 5.3: Comparison of the Effect of Additive Noise and Channel Distortion 72
Table 5.4: Autocorrelation Function for Input Timing Jitter 79
Table 5.5: Allowable Peak-To-Peak Phase Jitter at Steady-State 82
Table 6.1: Effect of Additive Noise and Channel Distortion 96
Table 6.2: Effect of Dead-Zone Treshold 100
Table 6.3: Effect of Initial Starting Phase 105

List of Abbreviations

Abbreviation	Meaning
2B1Q	Two Binary, One Quaternary
A/D	Analog to Digital
AAF	Anti-Aliasing Filter
ARC	Automatic Reference Control
ATQ	Adaptive Threshold Quantizer
baud	symbol
baud-rate	symbol rate
BW	Bandwidth
CCITT	International Telegraph and Telephone Consultative Committee
CONVL_SAMP	Convolution and Sampling block
CRC	Cyclic-Redundancy Check
D/A	Digital to Analog
DA	Decision Aided
DAC	Digital to Analog Converter
dB	Decibel

Abbreviation	Meaning
DF	Decimation Factor
DFE	Decision Feedback Equalization
DPLL	Digital Phase Locked Loop
DRV	(loop) Driver
DSL	Digital Subscriber Loop
DSP	Digital Signal Processing
DZT	Dead-Zone Threshold
EC	Echo Cancellation
ERR	Error (adaptive block)
FEXT	Far End Crosstalk
FIFO	First In First Out
FIR	Finite Impulse Response
Hz	Hertz
IIR	Infinite Impulse Response
ISDN	Integrated Services Digital Network
ISI	Inter-Symbol Interference
KFT	Kilo Feet
KHz	Kilo Hertz
LMS	Least-Mean-Square algorithm
LPF	Loop Filter
LSI	Large Scale Integration
LT	Line Termination
M&M	Mueller and Müller
MEC	Memory Echo Canceller

Abbreviation	Meaning
MHz	Mega Hertz
MLE	Maximum Likelihood Estimation
MMSE	Minimum Mean Square Error
MSD	Mean Square Distortion
MSE	Mean Square Error
NDA	Non Decision Aided
NEXT	Near End Crosstalk
NT	Network Termination
PAM	Pulse Amplitude Modulation
PD	Peak Distortion
PHASE_QUANTZ	Phase Quantizer
PLL	Phase Locked Loop
PQ_THR	Phase Quantizer Threshold
RMS	Root Mean Square
SDM	Sigma Delta Modulator
SFLT	Smoothing Filter
SNR	Signal to Noise Ratio
T_DFE	Transversal Decision Feedback Equalizer
TCM	Time-Compression Multiplexing
TEC	Transversal Echo Canceller
TIME_EST	Timing Estimate generator block
TR	Timing Recovery
TR_AVG	Timing Recovery Average Length
TXF	Transmit Filter

Abbreviation	Meaning
VCO	Voltage Controlled Oscillator
VLSI	Very Large Scale Integration
WDM	Wave Difference Method
WSUM	Weighted Sum

CHAPTER 1

Introduction

1.1.Introduction

Massive scale efforts on the part of the telecommunications industry have led to the development of a worldwide Integrated Services Digital Network (ISDN). In the last decade, standards relating to various aspects of ISDN such as signalling, network interfaces, and protocols have been addressed by a number of organizations, with the CCITT acting as the primary controlling body. The digital subscriber loop (DSL) in ISDN provides a wide variety of services to the user over common twisted-pair telephone links by integrating existing and newly-developed voice and data applications.

The transmission rate for DSL is 160 Kbits/s including two full-duplex 64 kbps B channels, one 16 Kbps D channel, plus an additional 16 Kbits reserved for framing, maintenance and other functions. Figure 1.1 shows the connection structure of ISDN. At the U-interface, full-duplex transmission between the network termination (NT) at the subscriber's end and the line termination (LT) at the central office end, is achieved on a single pair of wires. This necessitates the use of techniques that will separate the signals in

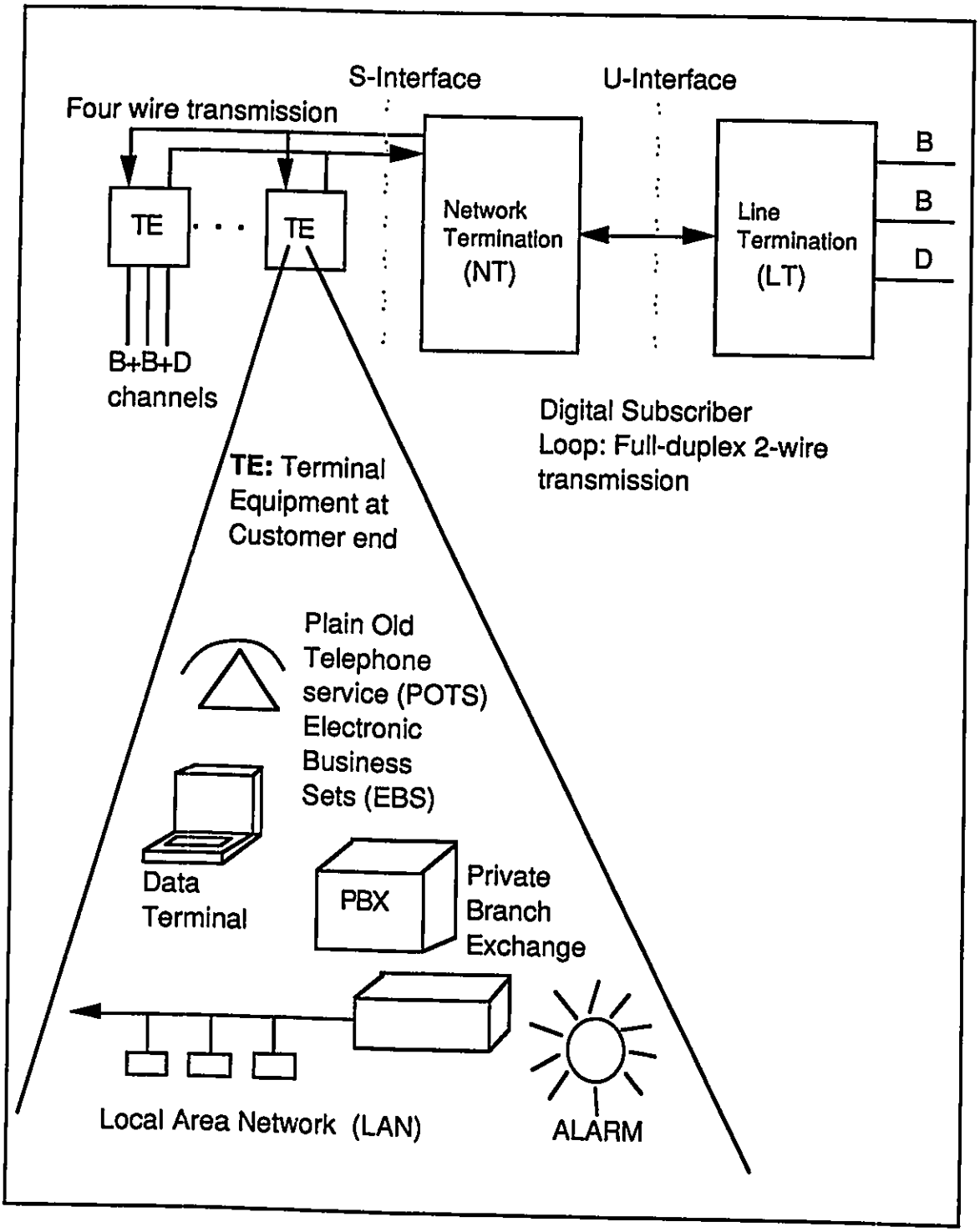

Figure 1.1: ISDN Connections Structures [Cook89][Stal85]

the two transmission directions. Two common methods are time-compression multiplexing (TCM) where the common link is shared in time with each end transmitting alternately, and echo cancellation technique (EC) where hybrids are used to isolate the transmitted signal from the received signal at each end. The two techniques are illustrated in Figures 1.2 (a) and (b) respectively. A comparison between the two methods for DSL applications is given in [Tzen85]. EC was eventually accepted by CCITT as the transmission method. Over the last few years, the amount of literature dealing with various aspects of echo cancellation in DSL systems has greatly increased [Cook89] [Falc82] [Falc85] [Mess84] [Mess86] [MoNP87] [Yama89]. Full implementations of EC mode ISDN U-transceivers using large scale integration (LSI,VLSI) have been widely reported [ACCC87] [AdCG87] [EhrT82] [FOYT89] [FTGH88] [GBKM89] [SuHM89] [SzZS86].

In general, the terminating end (NT or LT) at the U interface for DSL requires the following processing blocks [ALY88].

Analog section which includes D/A and A/D conversion at the transmitter and receiver ends respectively

Digital signal processing section which performs echo cancellation, data recovery (equalization and quantization of received signal), timing recovery (symbol synchronization), timing generation, error analysis and startup state machine.

Bit processing section for framing, maintenance and external control.

This thesis deals with the timing recovery for the EC based DSL systems described above. The overall content is as follows: First, the timing recovery problem, in general and as it relates to the echo cancellation technique for DSL, is introduced. Then, a timing recovery technique suitable for the above application is chosen and described. Four timing estimates based on this technique are selected and characterized. Finally the performance of

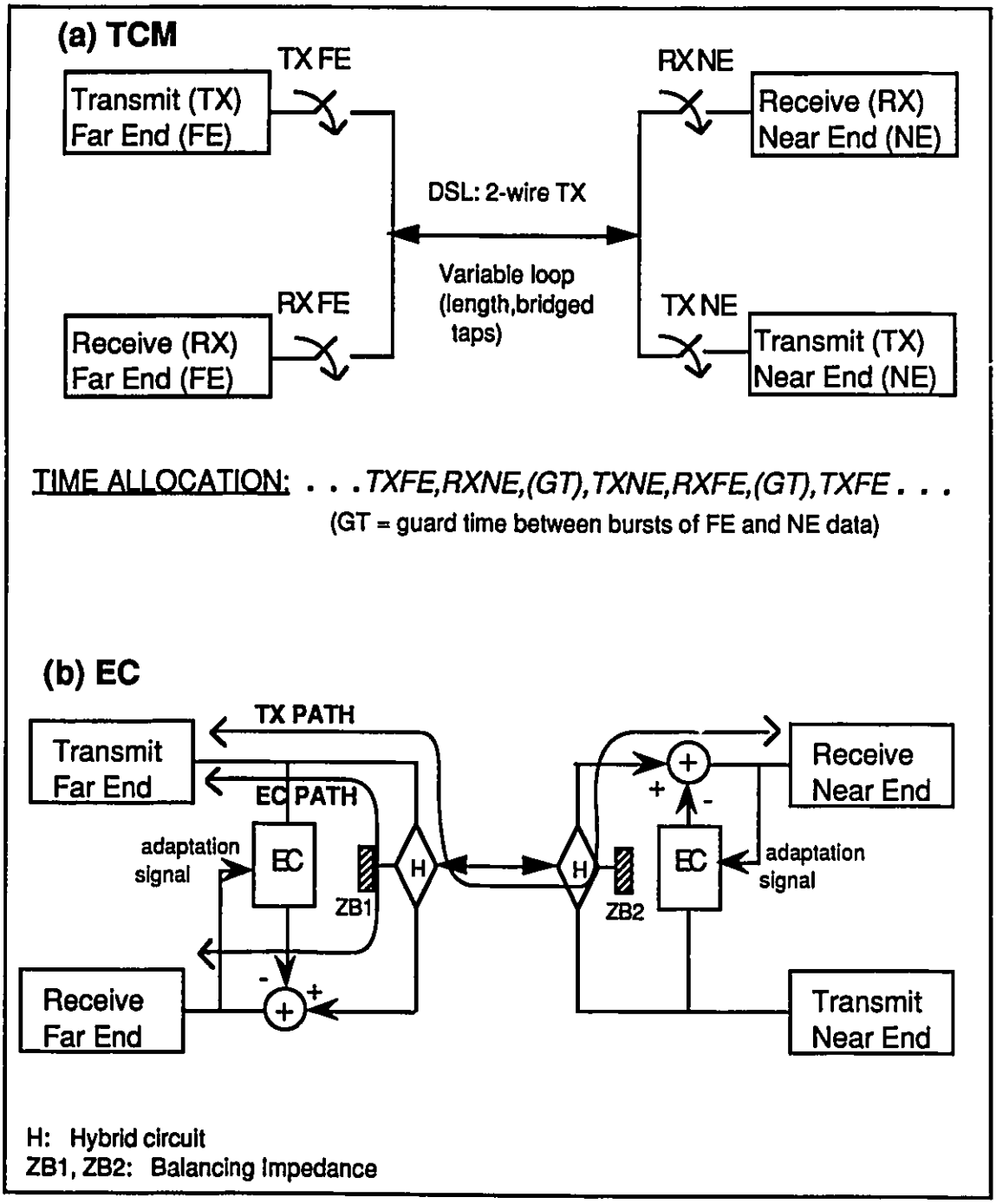

Figure 1.2: 4-Wire/2-Wire Transmission Techniques for DSL Applications (a) Time Compression Multiplexing Method (b) Echo Cancellation Method

the timing estimates in a practical U-transceiver design is investigated. In the next section, a more detailed outline of the thesis is given.

1.2.Thesis Outline

In this chapter, the timing recovery problem in general is introduced. The purpose of timing recovery in communication systems is discussed. A brief comparison between analog and digital timing recovery is given. Finally, the criteria by which timing recovery is evaluated is defined.

In chapter 2, a survey of analog and digital timing recovery techniques as classified in chapter 1, is given. From this, the most suitable methods for EC based DSL systems are chosen and compared. The timing recovery problem as it relates to DSL applications is then investigated and the requirements of the various processing blocks in a U interface transceiver as described above, on the timing recovery problem, are discussed.

In chapter 3, an EC type symbol-rate digital transceiver for ISDN U interface, which implements the above functions and meets all of the objectives discussed in chapters 1 and 2 is described. The overall architecture is outlined with a detailed description of every function. This transceiver will be used in the characterization of the timing recovery estimates in later chapters.

In chapter 4, the timing recovery algorithm used to derive the timing estimates is discussed in detail. This chapter consists mainly of a review of the theory for the algorithm and includes a description of the method, in general, adopted in the derivation of the estimates.

In chapter 5, four timing estimates based on the algorithm in chapter 4 and suitable for the implementation of the system described in chapter 3, are given. A theoretical analysis of these estimates and a comparison of their performance in general is presented. A practical timing recovery loop in which these estimates will be used is then described in

detail and the performance of the timing estimates as it relates to the timing recovery loop is investigated and compared, and some preliminary theoretical results are given.

In chapter 6, a simplified version of the ISDN U transceiver described in chapter 2 is implemented in a simulation setup. The performance of the timing estimates is then examined in the practical system on typical loops. Various system parameters and their effect on the timing recovery block and the performance of every estimate are investigated. These parameters include the channel response and noise. Simulation results are compared against the theoretical discussion given in chapter 5.

In chapter 7, some concluding remarks are given and recommendations for future works are suggested.

1.3.Timing Recovery

In a digital communication system, synchronization is defined at various levels [Fran80]. Carrier synchronization consists of recovering the reference carrier frequency as well as the signal's phase. This is needed in passband systems where the receiver uses the recovered carrier for demodulation. Symbol synchronization or timing recovery performs the equivalent role as that of carrier recovery for baseband data, including the demodulated passband signals. Timing recovery is also commonly referred to as bit synchronization especially for binary data, but also for multi-level alphabets. Word, frame and packet synchronization are used for formatting and synchronization of communication protocols. This belongs to a higher system-dependent level of synchronization and will not be discussed here. In this thesis, only timing recovery will be considered.

Timing recovery is a critical function in any synchronous data communication system. In general, digital systems do not require timing recovery since a clock can be transmitted separately from the data signal. However, for communication systems, limitations such as bandwidth and power, render transmission of a separate clock

inefficient and costly [LeeM88]. Timing recovery which derives the clock from the received signal itself is simpler and more economical to implement. Such a "self-timing" technique requires that the timing information be implicit in the received data. This imposes additional requirements on the signalling methods [LeeM88]. Furthermore, since the timing information is extracted from the received signal, then the statistics of the signal, the line code and pulse shape become a fundamental consideration in the choice of the timing recovery scheme.

The purpose of timing recovery is the synchronization in frequency, between the oscillators used in the transmitter and receiver clock circuits. The timing recovery circuit should recover the clock at the receiving end such that the continuous-time signal is sampled at the optimum phase and converted into a discrete time sequence of data symbols. In practice, the clock at the far-end transmitter cannot be perfectly regenerated. However, the timing recovery circuit must ensure that the average frequency of the derived timing is exactly equal to the average frequency of the transmitted signal. Basically, an equal number of bits must be transmitted and received. In [LeeM88], timing recovery techniques are globally classified as either inductive or deductive. In a deductive system, a timing tone with an average frequency that is exactly equal to the symbol rate is derived from the received signal, as shown in Figure 1.3 (a). Typically, sampling of the data signal is performed at the zero-crossings of the extracted timing tone. A phase-locked loop (PLL) can be used in conjunction with deductive timing recovery as shown in Figure 1.3 (b) , in order to reduce the timing jitter present in the timing tone. In inductive timing recovery shown in Figure 1.4, a PLL is used as an integral part of the system to achieve synchronization. This allows the timing recovery to be mostly performed digitally or in discrete-time , however, the sampling rate might have to be higher than the symbol rate in

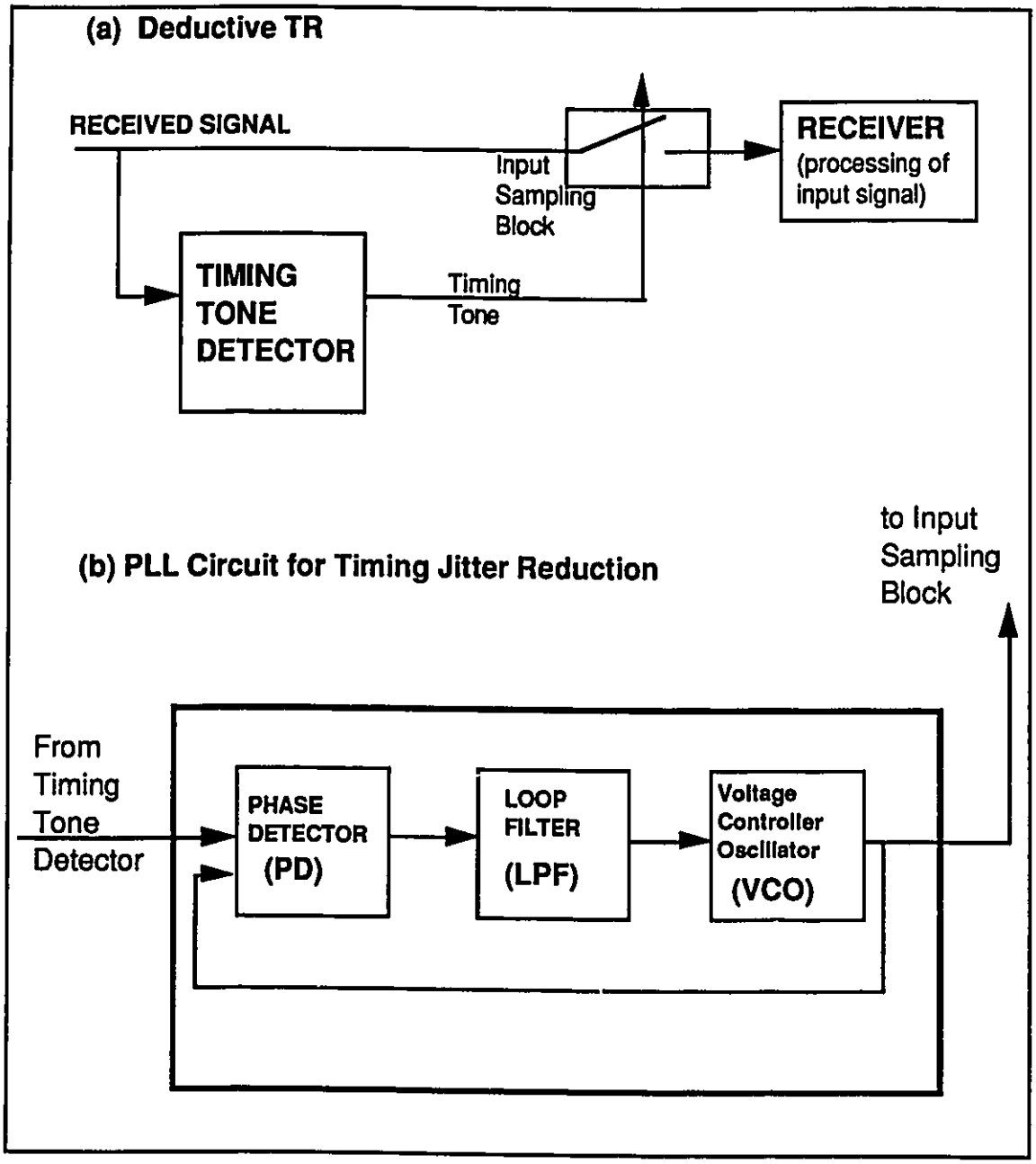

Figure 1.3: Deductive Timing Recovery [LeeM88] (a) TR system (b) PLL Circuit for Reduction of Timing Jitter in Deductive TR

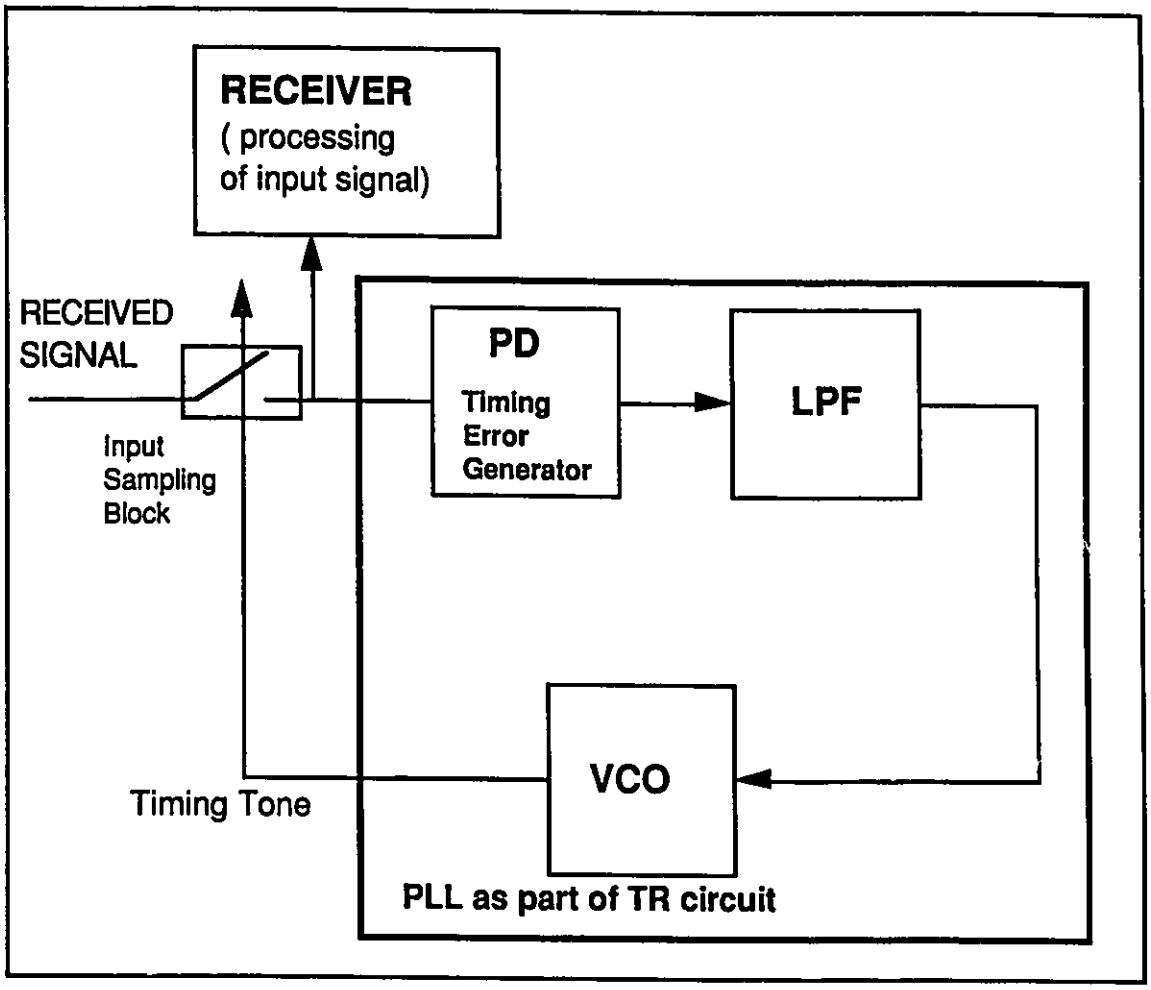

Figure 1.4: Inductive Timing Recovery system [LeeM88]

order to be able to estimate the timing error. The need and importance of digital timing recovery (inductive) as opposed to analog timing recovery (deductive) is discussed in the next section.

The timing information is derived based on some criterion which determines the best steady-state timing phase. The optimal location of the sampling instants depends on the overall channel impulse response. The unknown channel delay as well as the dispersive nature of the loop which could cause severe intersymbol interference, are important limiting factors in the performance of timing recovery.

1.4.Analog vs. Digital Timing Recovery

Traditionally, analog data receivers were used where the received signal was continuous in both time and amplitude. Thus, traditional timing recovery methods required the continuous signal for extraction of the needed timing information. In digital systems, the received signal is sampled and is available only at discrete time intervals, making traditional timing recovery techniques awkward to implement. These methods can still apply if the received signal is oversampled at a very high rate such that the timing information extracted from the analog signal can be fully reconstructed from its discrete samples. Obviously, this option is not practical since we would want the lowest sampling rate possible to minimize the complexity of the receiver. Another advantage of digital timing recovery methods over the traditional analog techniques is the elimination of analog components resulting in several improvements, including time and temperature stability, ease of manufacturing reproducibility, and integrated circuits (VLSI) implementation. Finally, the need of all-digital timing recovery can be clearly seen in Figure 1.5. For an analog system, sampling is performed on the "clean" signal after the filtering, the equalization and the other signal processing operations, such that a feedback path is not needed for timing information extraction (Figure 1.5 (a)). However, in digital systems, sampling is done on the distorted signal before any other signal processing function is performed, and so, the timing recovery scheme has to include a feedback loop as shown in Figure 1.5(b).

A survey of commonly used continuous-time and discrete-time timing recovery techniques will be given in the next chapter. However, first, the criteria by which timing recovery is evaluated will be defined in the next section.

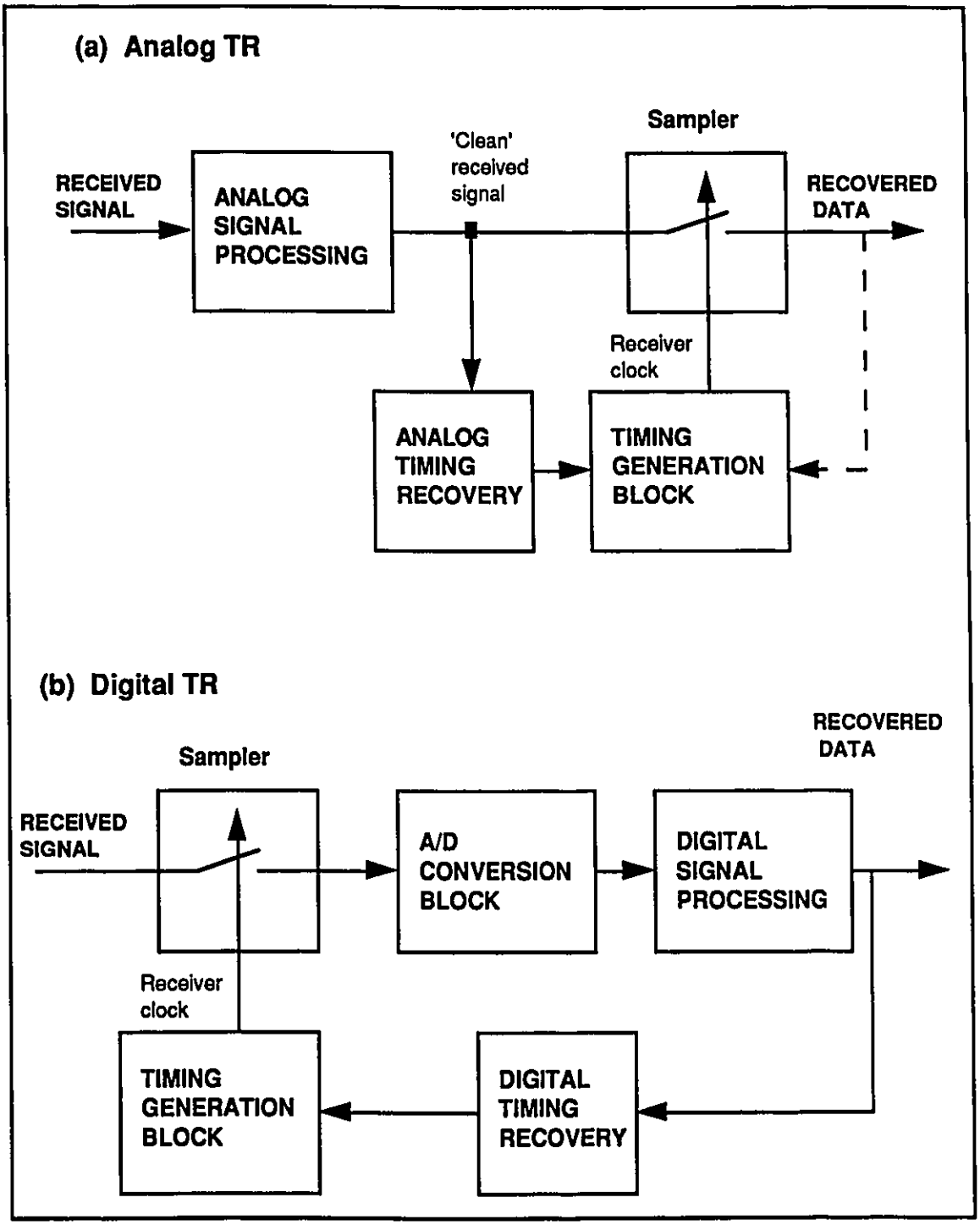

Figure 1.5: Analog vs. Digital Timing Recovery (a) System using analog TR (b) System using digital TR

1.5.Criteria for Timing Recovery Performance

A specific analytical comparison between alternative timing recovery methods is not feasible. Performance is typically evaluated based on the criteria for timing phase and timing jitter. Theoretically, the timing phase should be chosen to minimize error probability but, for practical implementation, we choose the sampling phase for maximum eye opening (minimum intersymbol interference (ISI)), or for minimum mean square error (MMSE criterion). The signal's sensitivity to the timing phase is given by the horizontal eye opening as shown in Figure 1.6 for the (a) 25% and (b) 100% excess bandwidth raised cosine binary antipodal pulse amplitude modulation (PAM) signalling [LeeM88]. The slope of the inside lid gives the sensitivity to timing jitter in the timing phase. Obviously, the 25% excess bandwidth signal has a much higher degree of sensitivity to the timing phase than the 100% excess bandwidth signal, since for the former, the horizontal eye opening closes much more rapidly as the timing phase deviates from the optimal point. For both cases, the best sampling instants are located at the peak of the pulse, at the maximum eye opening.

Timing jitter has significant effects on the system performance. First, it causes sampling to occur at a sub-optimal phase, thus increasing the ISI in the received signal samples, which in turn reduces the signal-to-noise ratio (SNR). Secondly, in long distance transmission systems where the signal passes through a chain of repeaters before arriving at the receiver's end, the timing jitter introduced at every repeater is accumulated. This causes significant degradation of the received signal. Thus the design of a practical timing recovery circuit, must take into account all of the above mentioned points.

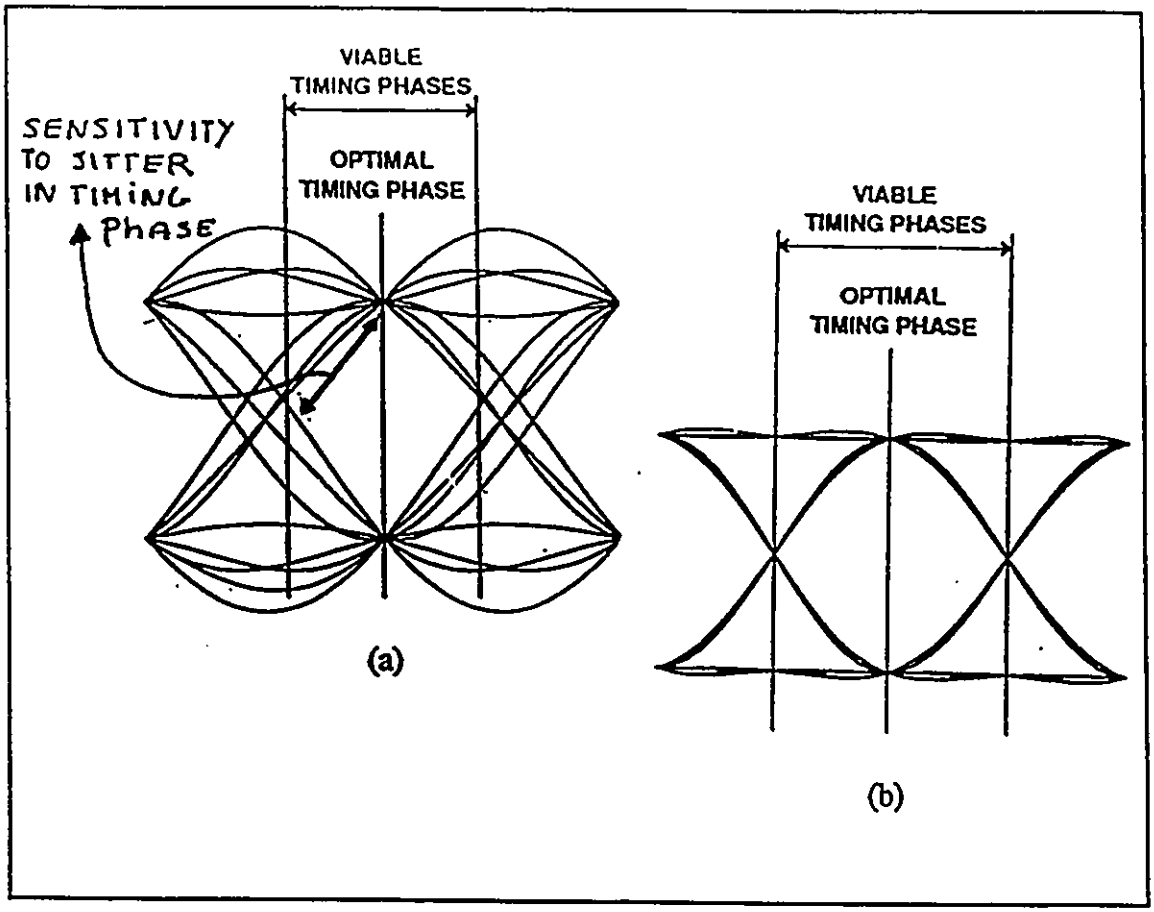

Figure 1.6: Eye Diagram for Binary Antipodal Signalling [LeeM88] (a) 25% Excess Bandwidth Raised Cosine Pulse (b) 100% Excess Bandwidth Raised Cosine Pulse

CHAPTER 2

Timing Recovery Techniques for DSL Applications

2.1.Survey of Timing Recovery Techniques

The literature surveyed comprises of a wide variety of timing recovery techniques. Both continuous-time and discrete-time methods have been considered.

2.1.1.Continuous-time Techniques

The spectral-line technique is one of the most popular among the continuous-time methods [AnMM86] [Benn58] [Frab74] [Fran80] [Gard86] [GitS71] [Lyon75]. It consists of passing the received data signal through a memoryless nonlinearity such as squaring or full-wave rectification. This results in a waveform with a deterministic mean-value that is periodic at the symbol rate [Benn58]. A timing tone can then be derived by extracting the discrete spectral-line component using a narrow bandpass filter centered at the symbol rate (Figure 2.1 (a)), or an equivalent phase-locked loop circuit (Figure 2.1

(b)). [Frab74] shows that data-dependent jitter free timing recovery is achievable if the non-linear device is a square-law rectifier. Such a frequency-domain approach is only applicable to the quadratic characteristic of the rectifier and cannot be extended to higher order nonlinearities. [AndM86] considers the general case of an arbitrary nonlinearity, and shows that a "band-limited locally symmetric" (BL-LS) class of pulses can be found such that close to jitter-free timing is achieved. In [Gard86], the theory of spectral correlation for "time series that exhibit cyclo-stationarity" is used to provide a unifying conceptual framework for the design of quadratic synchronization using PLL structure.

Other ad-hoc continuous-time timing recovery techniques have appeared over time, with several of these having been shown to be approximations to a maximum likelihood timing recovery, with a strong resemblance to MMSE methods [LeeM88]. In the sampled-derivative method [GitS71] [Salt67] shown in Figure 2.2, the sampling phase is adjusted until the derivative is zero. This occurs at the peak of the signal.

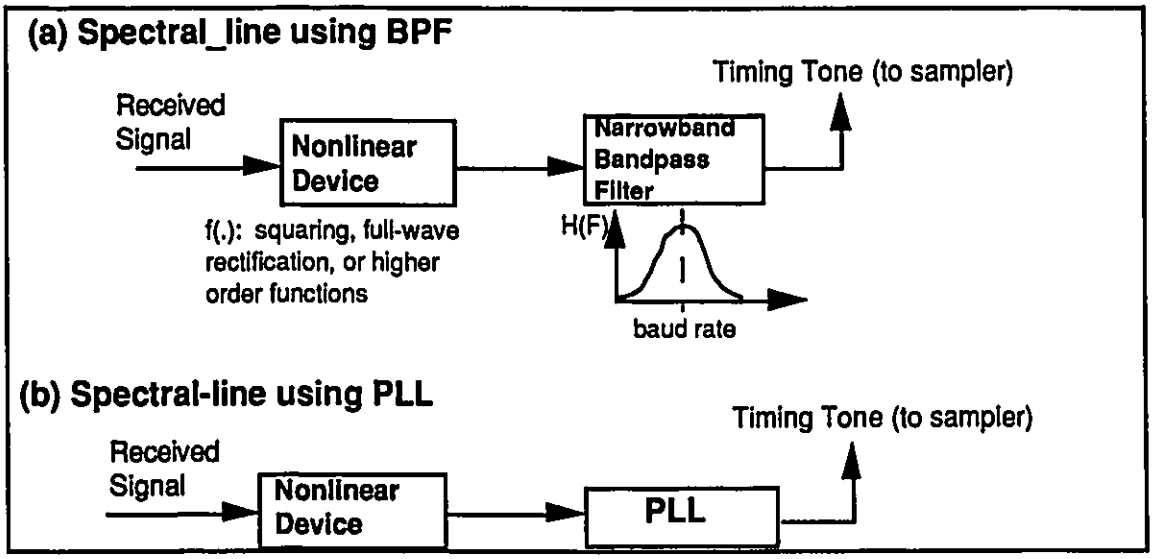

Figure 2.1: Spectral-Line Timing Recovery Method (a) System using BPF (b) System using PLL

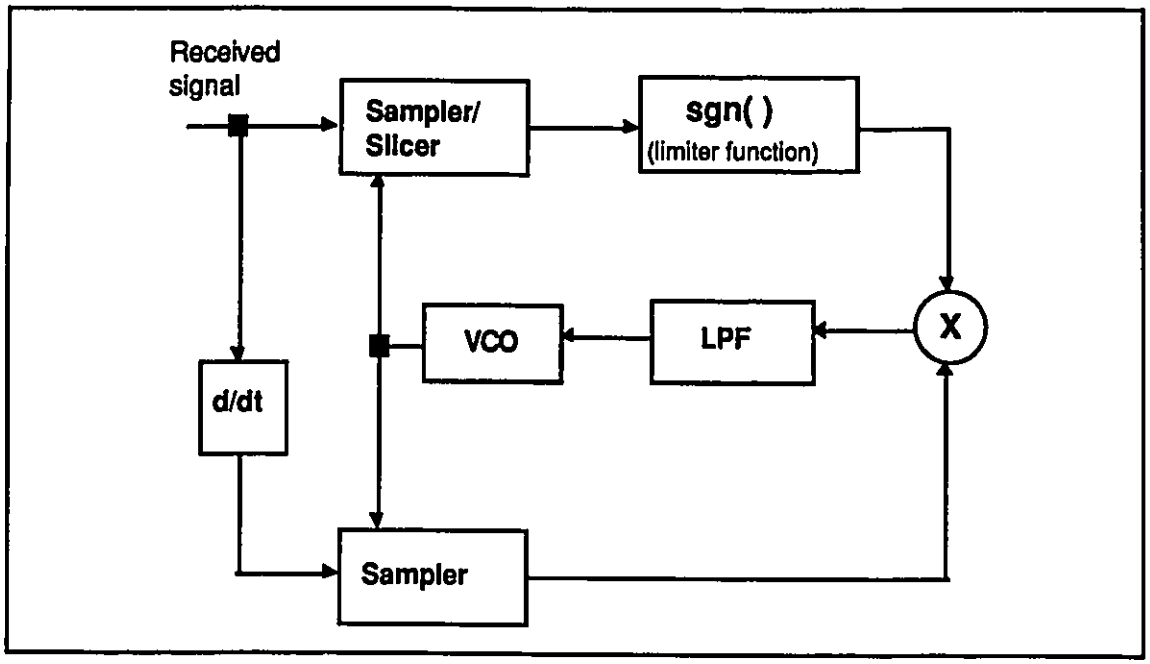

Figure 2.2: Sampled-Derivative Timing Recovery Method [GitS71]

The early-late gate method [GitS71] samples the received signal at two equally-spaced instants before (early) and after (late) the sampling phase as shown in Figure 2.3. Phase adjustments are performed until the two samples are equal. In the threshold-crossing method [Salt67] [FogG89] illustrated in Figure 2.4, an error signal that is proportional to the difference between the time of occurrence of the threshold crossing and the time of the nearest pulse of the displaced sampling wave is generated every time the received signal crosses zero. In this case, the recovered phase is the time instant such that the samples of the signal displaced by half the period, at either end of the pulse, are equal. It was shown that, on average, both the sampled-derivative and the threshold-crossing techniques lead to timing phases the are different from the time of maximum eye opening [Salt67]. However, if the channel's impulse response is symmetrical around its peak, then both methods result

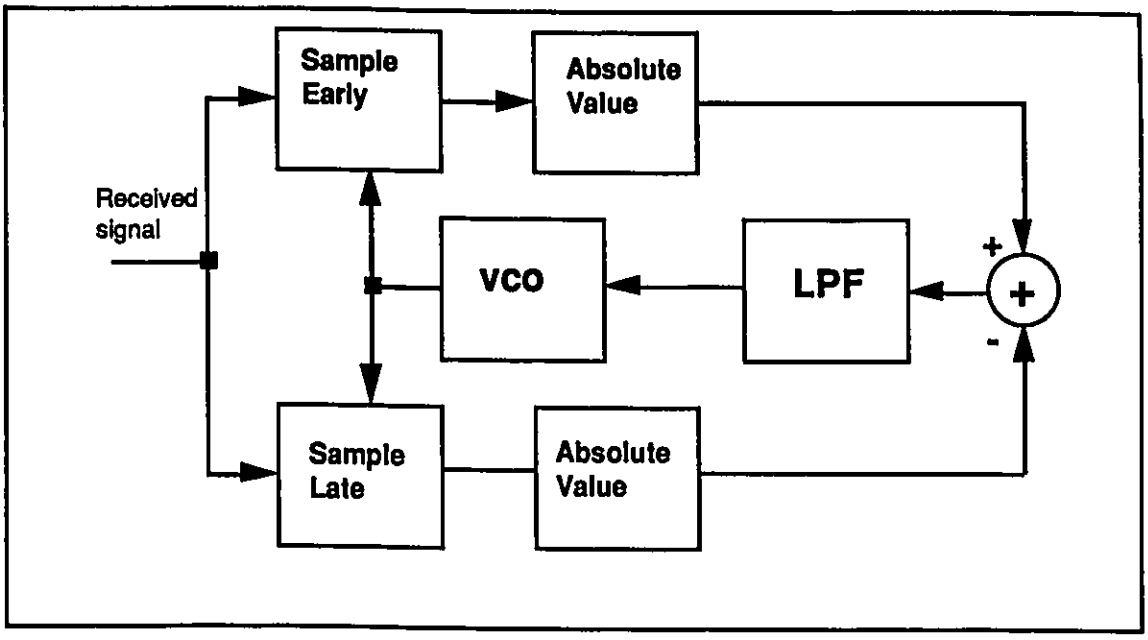

Figure 2.3: Early-Late Gate Timing Recovery Method [LeeM88]

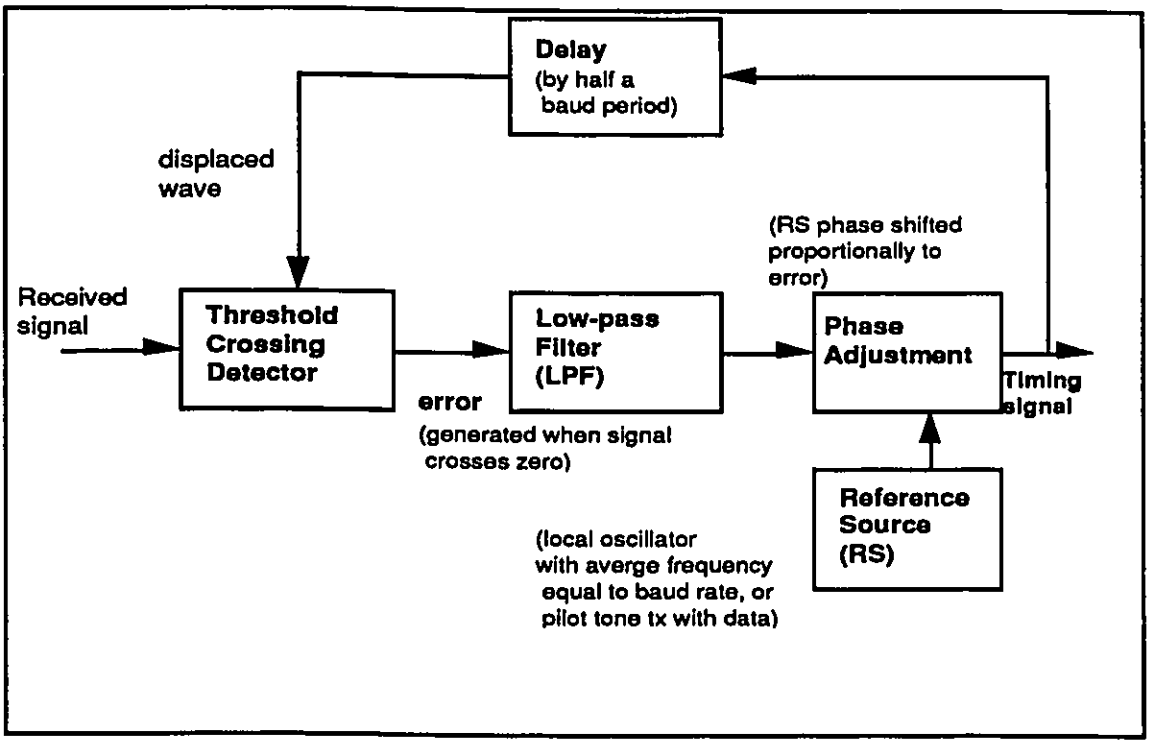

Figure 2.4: Threshold Crossing Timing Recovery Method [Salt67]

in sampling times that coincide with the maximum eye opening [Tzen85]. [Salt67] presents an improved sampled-derivative system with added correction terms which yield an average sampling time at the maximum eye opening.

2.1.2.Discrete-time Techniques

Among the sampled-data techniques, the wave-difference method (WDM) and symbol-rate sampling techniques are the most commonly analyzed [ATMH85] [JenC85] [MueM76] [OerM88] [Pate88] [Qure76] [SaDM86] [TzHM86]. Other digital implementations include state-machine PLL, and maximum-likelihood estimation (MLE) approximations [Pate88].

The WDM depicted in Figure 2.5 has been shown to be applicable to TCM and EC methods requiring a sampling rate as low as 2 to 4 times the symbol rate. The timing function is defined as the expected value of the received signal after it has been processed by some nonlinear operation. A sampled version (time average) of the time function can be computed (approximated) using a transversal filter [ATMH85]. In fact, for a square-law nonlinear operation, the WDM becomes the discrete-time counterpart of the spectral-line method, with better performance with respect to jitter. The WDM has been shown to meet the required performance for DSL systems with satisfactory phase recovery even for worst case loops [ATHM85]. A similar digital realization of the WDM is presented in [OerM88]. This latter technique consists of a squarer followed by a digital filter, which allows free running sampling oscillators. This means that the input signal is sampled at a fixed rate by a free running oscillator and all further processing needed for timing extraction is done digitally by some sort of interpolation controlled by an estimate of the current timing offset. This method extracts the timing information in a new way and proposes a new method of

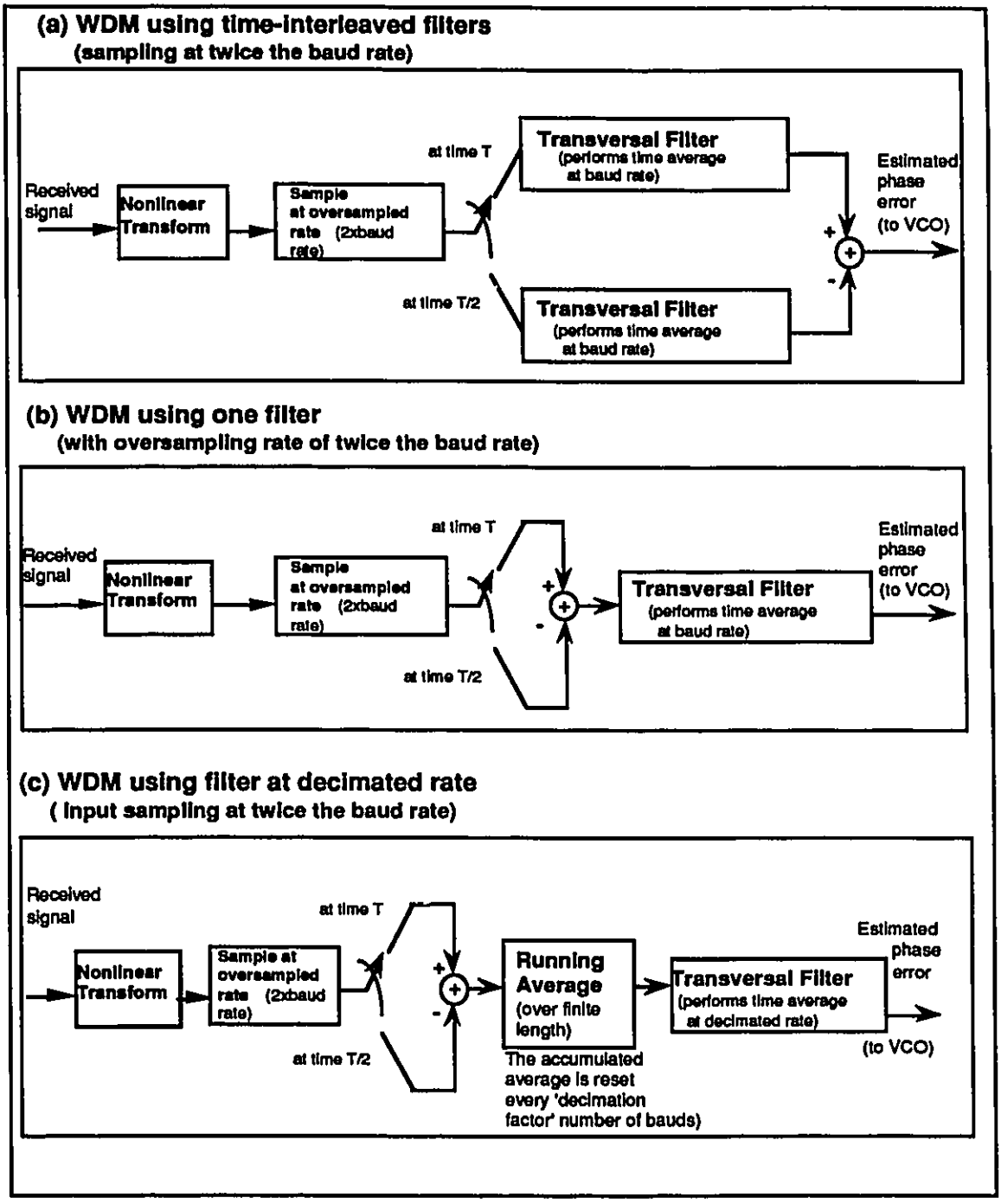

Figure 2.5: The Wave-Difference Timing Recovery Method [Tzen85] - Three configurations (a) Time-Interleaved Transversal Filters (b) Single Transversal Filter (c) Single Transversal Filter at Decimated Sampling Rate

hang-up free filtering of the timing signal . Typically, the spectral component is extracted by a PLL or a narrowband filter, but here, it is determined from finite length sections of the filtered and squared signal samples by computing its complex Fourier coefficient at the symbol rate. The estimate of the timing offset corresponds to the argument of this Fourier coefficient. This estimate can then be filtered to improve this estimation. However, this kind of postfiltering can cause the synchronization loop to be stuck in an unstable equilibrium point due to the periodic behavior of the filtered estimate. These hangups in the filter loop can be avoided if the filtering is performed on the Fourier coefficient (complex phasor) directly instead of on its corresponding periodic angle. The angle of the filtered phasor can then be used to control the sampling.

The state-machine PLL method does not require A/D conversion and can be implemented using basic digital components as shown is Figure 2.6. A timing error that is proportional to the phase difference of the two input signals is generated at the output of the phase detector. A programmable up-down counter is then used to derive an appropriate control signal to increase or decrease the phase of the output clock. This is fed to a control counter that adjusts the system clock resulting in an advance or retard of the input sampling phase.

The digital form of the MLE approximation is similar to the early-late gate method. The timing error estimates the slope of the received signal at the current input sampling phase by a difference between an early and late sample. For this, two samples per symbol is required for the timing information. At the input, an analog or digital implementation of the matched filter can be used, the latter can usually be approximated by a low-pass filter. This yields an error signal of average value zero if the estimated and actual input timing phases are equal. The error signal is then filtered using a simple first order loop filter (running average sum) such that it can be used to control the period of the sampling clock.

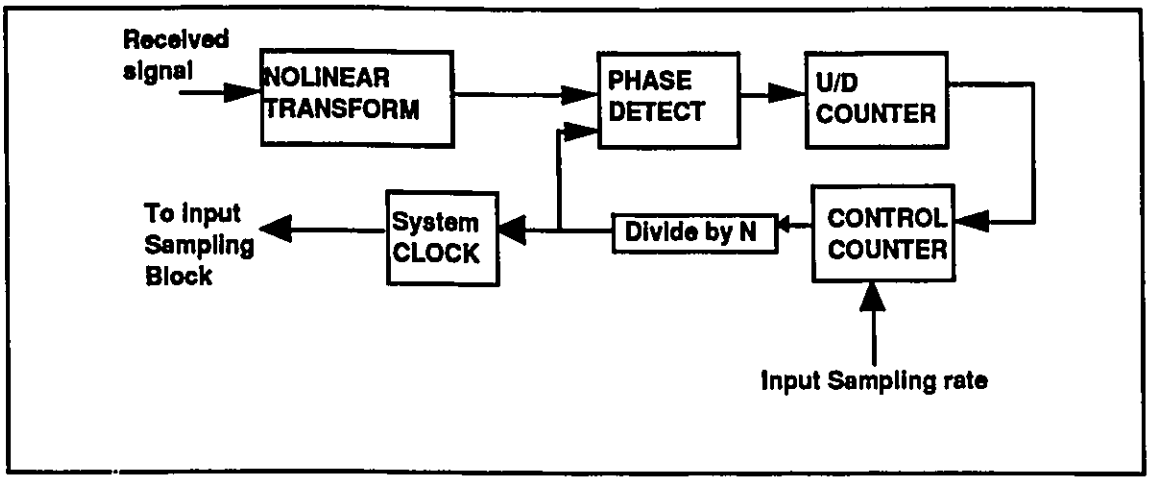

Figure 2.6: State-Machine PLL Timing Recovery [Pate88]

The symbol rate sampling technique is the best candidate for EC based DSL systems because it comprises the minimum possible rate, the symbol rate. The Mueller and Müller technique [MueM76] is by far the most commonly used for symbol rate timing recovery. It is simple and economical to implement and applies to binary, multi-level PAM and partial response signals. The M&M algorithm proves to be the most suitable for DSL applications and will be described in full detail in later chapters. [JenC85] proposes a modified version of the Mueller and Müller symbol rate technique in which timing errors are calculated only after certain data sequences. Basically, the weighting functions for the timing estimates are limited to very simple functions. Then, the finite number of sequences of data symbols which satisfy these estimates are determined, and only when these sequences are encountered, is the timing estimate computed and the timing phase adjusted.This scheme has less frequent phase adjustments making it inappropriate for systems where the optimum phases fluctuates rapidly with time, and resulting in longer convergence time. On the other hand, estimates with lower variance than those obtained by the Mueller and Müller algorithm are typically used, thus yielding lower timing jitter.

Basically, since all possible data sequences of data symbols are considered in the derivation of the Mueller and Müller timing estimates, these will result in higher variance estimates. In [SaDM86], two MMSE type algorithms are presented. The first algorithm is the conventional polarity-type MMSE, and the second is based on the minimization of the MMSE criterion which decouple ISI measurement from the carrier phase.

A multitude of other published literature on timing recovery exists, with emphasis on digital methods. These are by far more attractive allowing implementations using DSP chips. The DSL application poses an altogether different set of synchronization problems, making conventional analog timing recovery techniques unsuitable. These requirements will be discussed next, with special consideration to their effects on the choice of timing recovery.

2.2.Synchronization Problem for Digital Subscriber Loops

The digital subscriber loop application requires synchronization in both data transmission directions, thus favoring discrete-time timing recovery schemes. This differs from voiceband application where the two directions are asynchronous. In the latter case, timing recovery can be performed on the continuous-time received signal and conventional analog methods are preferable [ATMH85]. A block diagram showing the timing loop at the U-interface is given in Figure 2.7. The master clock is derived from the switch at the central office, and is used to transmit data in the LT to NT direction. At the NT side, the timing is derived from the data stream sent by the switch, and is used to sample that same incoming data stream at the NT receive end. Then, "loop-timing" is used at the NT transmitter to synchronize the data received by the LT to the switch clock. This means that the same timing derived and used at the NT receive end is also used at the NT transmit end. At the LT end, timing has to be derived again from the received signal sent by the NT, and

used to sample that same signal. Due to the "loop-timed" nature of timing, the receiver must use the recovered timing clock to perform the required signal processing functions such as echo cancellation, equalization and data recovery. Since these functions must be performed digitally to meet the specifications for VLSI implementation, the choice of the timing recovery technique is directly affected by the design requirements of the various receiver functions. The conditions that are most critical will be stated in the following sections.

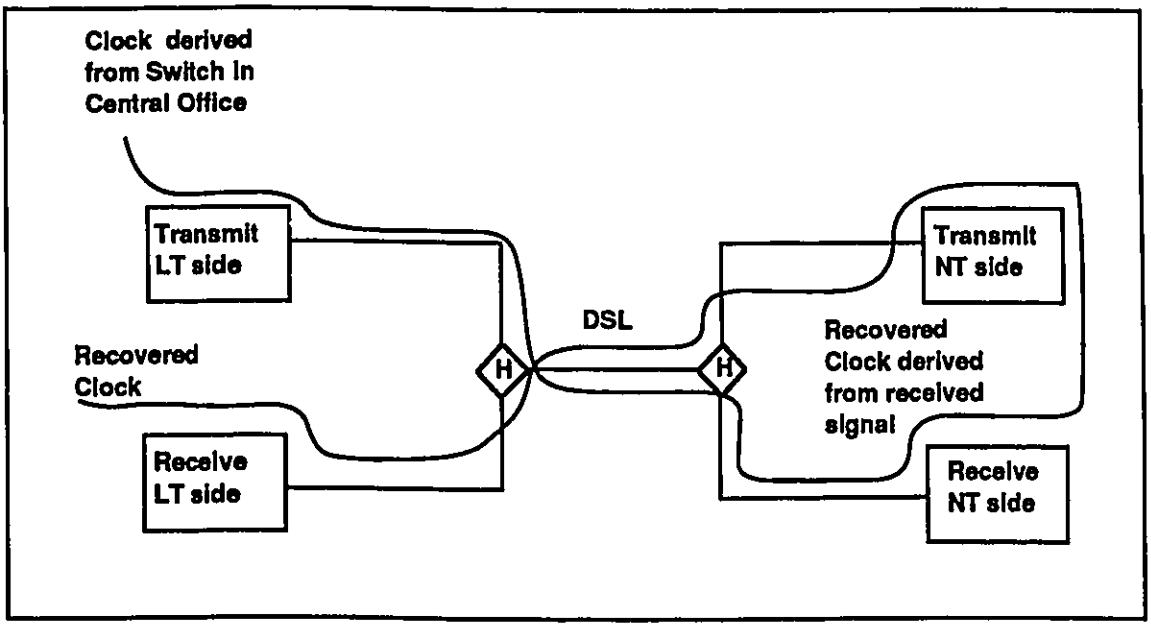

Figure 2.7: Timing loop in DSL [Tzen85]

2.3.Echo Cancellation Requirements on Timing Recovery

In an EC-based DSL system, echo cancellation is the most complex function to implement, requiring typically 60 dB of echo attenuation. This strict constraint on the echo cancellation accuracy will also affect the requirements on the timing recovery block. Timing

jitter causes a reduction in the EC accuracy at both ends. To achieve 60 dB of echo cancellation, input timing jitter must be kept as low as 0.1% of a signalling period [Tzen85]. At the NT side, timing jitter will equally affect the receive and transmitted pulses because the same recovered clock is used for sampling in both transmission directions, and thus it will not need to be tracked by the EC. However, high frequency jitter which causes a deviation of the phase on the order of the maximum echo delay will affect the EC accuracy. At the LT side, any change in the phase of the received signal will cause a corresponding shift in time of the echo path impulse response. For large phase variations, the adaptation of the EC might not be able to follow the subsequent shifts in the echo response. Moreover, since timing recovery can only be performed after the echo cancellation, the sampling rate for the timing recovery will also set the EC rate and thus the latter's complexity. Consequently, a symbol rate sampled timing recovery technique becomes the most suitable. In addition to minimizing the EC complexity, it allows the use of decision-feedback equalization (DFE) , resulting in high EC accuracy and fast convergence. In the EC coefficient adaptation, the error includes the far end signal. Therefore, to meet the above EC requirement, the adaptation step-size must be kept very small, increasing the convergence time. By using a DFE that cancels all postcursor ISI, in conjunction with a TR that minimizes the precursor part of the signal [TzHM86], the far end signal is completely eliminated from the feedback EC adaptation error. This then allows the use of a larger step-size resulting in faster convergence. By cancelling the ISI among the received pulses, we also reduce the jitter present in the input timing signal since the latter is recovered from samples of the "clean" received signal. This will increase the EC accuracy.

The timing jitter effect on echo cancellation has been greatly researched in the context of DSL systems [ATMH85] [Falc85] [Mess86] [Moen87]. A symbol rate sampled

technique was found to be the best for timing recovery allowing high EC accuracy, minimum complexity and fast convergence. In [Falc85], the timing jitter as it relates to the residual mean-squared error (MSE) of the received signal is analyzed and related to the EC adaptation, timing recovery and channel parameters. Timing recovery techniques based on a single-tuned (high-Q) filter as well as a PLL were applied to different echo canceller configurations. It was found that the high-frequency component in the timing jitter spectrum has considerable impact on the EC performance and must be minimized in order to meet EC requirements. In all cases, timing recovery using a narrow-band PLL was thus recommended. In [Mess86], an echo canceller configuration that is insensitive to timing jitter is described. The proposed technique performs adaptive interpolation of the echo response using two sets of EC coefficients. This allows the use of a simple digital PLL for timing recovery and still completely eliminates the effect of timing jitter on EC performance. The two major disadvantages of this scheme are the increase in the complexity of the EC control mechanism and the increase in the timing jitter in the received signal. However, for the latter, the slight increase in timing jitter in the signal is not as critical since an SNR of 20 dB is acceptable as opposed to required accuracy of 60 to 70 dB for the EC.

[Moen87] compares the performance of decision-aided (DA) versus non-decision-aided (NDA) timing recovery in the presence of residual echo signal. Figures 2.8 (a) and (b) illustrate the DA and NDA systems respectively. In both the DA and NDA performance, the residual echo resulted in a mean-squared (MS) timing error that is proportional to the timing recovery loop bandwidth (BW). However, for the DA synchronizer, an additional error term that is independent of the timing recovery loop BW was introduced due to the residual echo signal. This bias depends on the signal-to-echo ratio. Thus, the DA was found to outperform the NDA for small BW and high SNR.

2.4.Channel Requirements on Timing Recovery

In DSL transmission, the signal is greatly distorted by the presence of bridged taps. These introduce a $\sqrt{f}$ attenuation which causes a great reduction in the magnitude of the signal as well as dispersion of the signal pulse resulting in severe intersymbol interference. Distortion of the pulse shape also occurs which creates nulls in the spectrum of the signal. Since the line configurations are variable with respect to length and bridged taps, then the timing recovery must be chosen to be insensitive to these impairments.

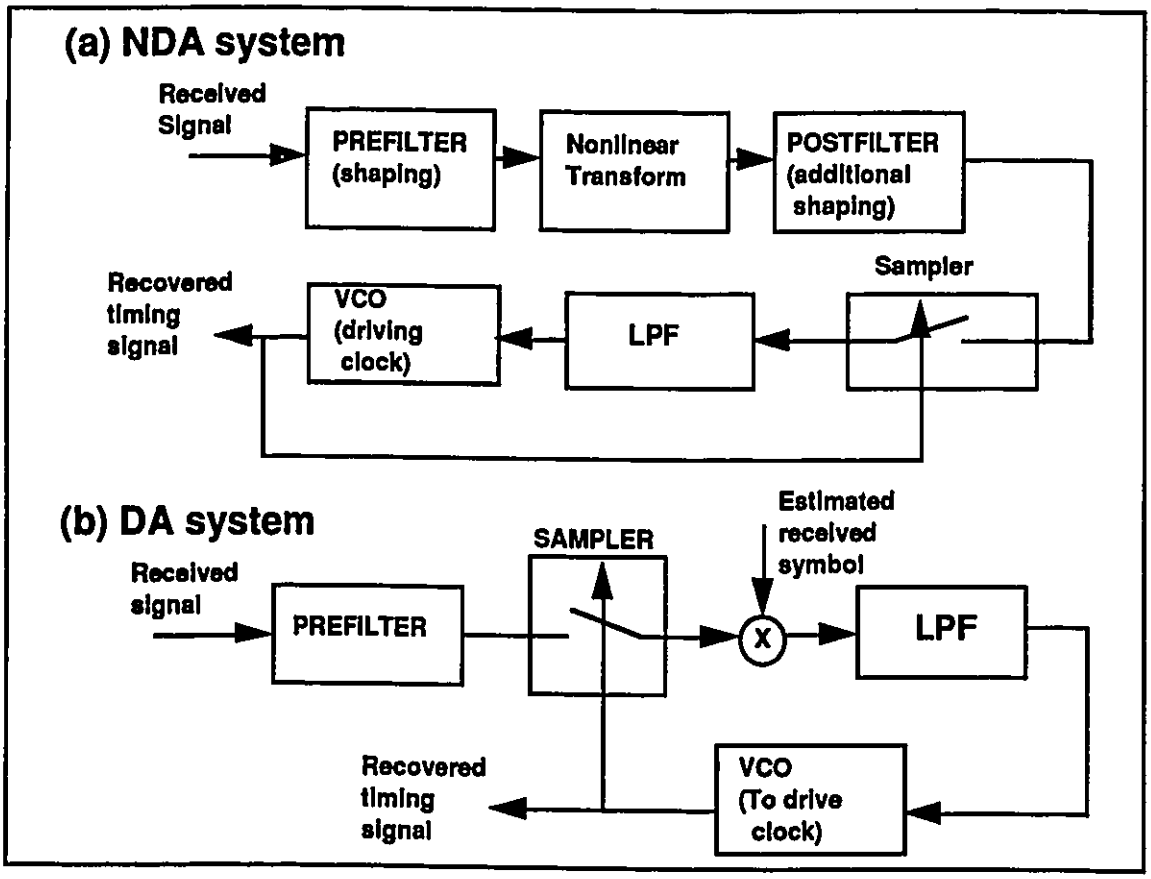

Figure 2.8: Decision-Aided vs. Non-Decision-Aided Timing Recovery [Moen87]

2.5.Equalization Requirements on Timing Recovery

The dispersive characteristics of the channel which causes ISI renders equalization on the received signal essential. Linear equalization is not efficient for VLSI implementation since it requires a large number of multiplications of digital coefficients and analog samples of the signal. The cumbersome hardware implementation problem is solved if a DFE structure is used. This operates at the symbol-rate and uses past decisions of the received signal to cancel postcursor ISI. However, the use of a linear equalizer becomes necessary if any precursor exits. To avoid this, the timing recovery must be chosen to minimize precursor ISI. If timing recovery can be designed to eliminate any significant precursor , then a simple DFE structure without any linear taps can be used to perform the equalization on the received signal, thus simplifying digital implementation.

2.6.Noise Requirements on Timing Recovery

The noise sources present in DSL applications are thermal, crosstalk, impulse and echo noise. Thermal noise is insignificant when compared to other disturbances and is usually neglected. Crosstalk noise is generally approximated as gaussian for interferences caused by a large number of wires (central limit theorem). FEXT and NEXT are far-end and near-end crosstalk respectively. FEXT effect is usually negligible, with NEXT causing most crosstalk interference. Impulse noise can be caused by external factors such as lightning which can be difficult to predict, but is mostly due to electromechanical devices. As such, implementations using all digital switches instead of electromechanical ones have considerably less impulse noise, and since digital implementations are becoming more popular, impulse noise is becoming less of a dominating noise factor. The residual echo error is random and bounded noise and its effect was discussed previously. The timing recovery method must be robust so as to be able to perform even in the presence of

high noise distortion. In our characterization, we only consider additive gaussian noise effect on the performance of timing recovery.

2.7.Summary

Timing recovery in general and as it relates to DSL applications was introduced. Various timing recovery techniques for both continuous-time and discrete-time implementations were reviewed. Specific design requirements for U-interface transceivers were discussed with emphasis on their effect on the timing recovery problem. In conclusion, the symbol-rate sampled data technique using a timing reference that minimizes the precursor part of the signal was found to be the most suitable for the EC-based DSL systems.

CHAPTER 3

System Description

3.1.Introduction

In this chapter, an EC type ISDN U transceiver that conforms to all of the objectives stated in chapter 2 is described [Aly88] [GBKM89]. The timing recovery functions characterized in this thesis were developed for this application [Saya89], thus, a full understanding of the system is required. This system is also used as the basis for the evaluation of the timing recovery functions in later chapters.

This chapter describes all the processing blocks giving detailed analysis of the DSP algorithms for echo cancellation, equalization and timing recovery. The architecture is mostly in digital domain with very little analog processing. A block diagram of the full transceiver is given in Figure 3.1. The following paragraphs describe each section in detail.

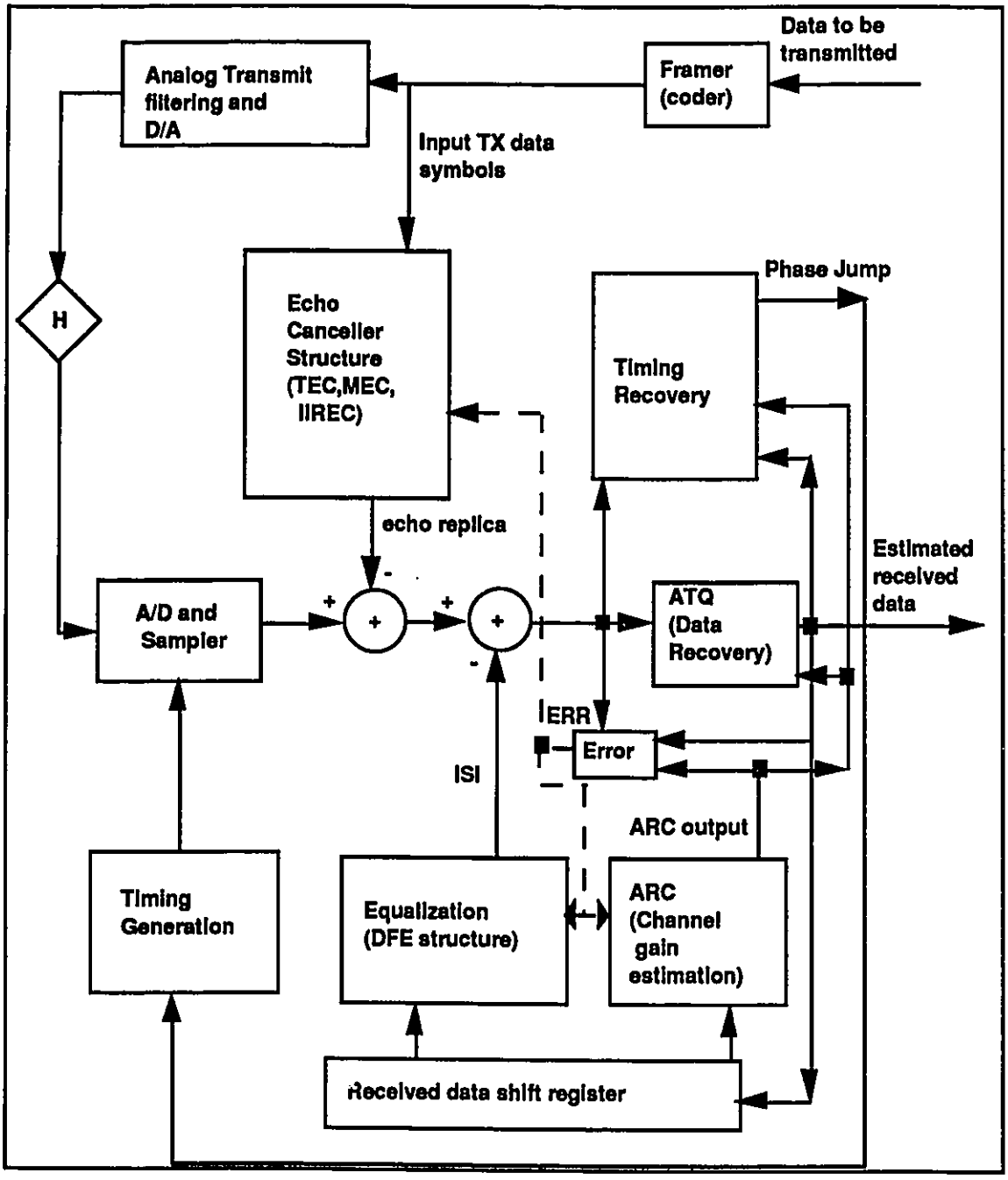

Figure 3.1: ISDN U Transceiver Architecture

3.2.Framer and Analog Front End

This section is illustrated in Figure 3.2 and includes, on the transmit end, a framing block, a digital to analog conversion block (DAC) , a transmit filter (TXF), a driver (DRV), and a smoothing filter (SFLT) , and, on the receive end, an anti-aliasing filter (AAF).

The framer converts binary data into a four level 2B1Q code as specified by the T1D1 specifications [ANSI89]. It then combines the symbols into frames of 120 symbols each, and superframes of 8 frames each. It also appends a synchword to each frame, with the beginning of the superframe identified by an inverse synchword. Some maintenance control bits are also added to comply with the defined U-interface standards, and CRC is performed for error monitoring. This block is clocked by the transmit clock at 80 KHz and controlled with a frame marker generated by the deframer so that it can synchronize with the far end modem.

The digital output of the framer is then converted to analog data quantized by a simple DAC. The DAC's output is then passed to a TXF which is a low-pass butterworth filter for shaping. Finally the signal passes through a driver which provides the differential output, followed by a SFILT, to the loop coupling circuit. The analog transmitted pulses as well as the echo signal passes through an AAF for bandlimiting of the received waveform. Following the above analog processing, quantization and input sampling of the received signal is performed.

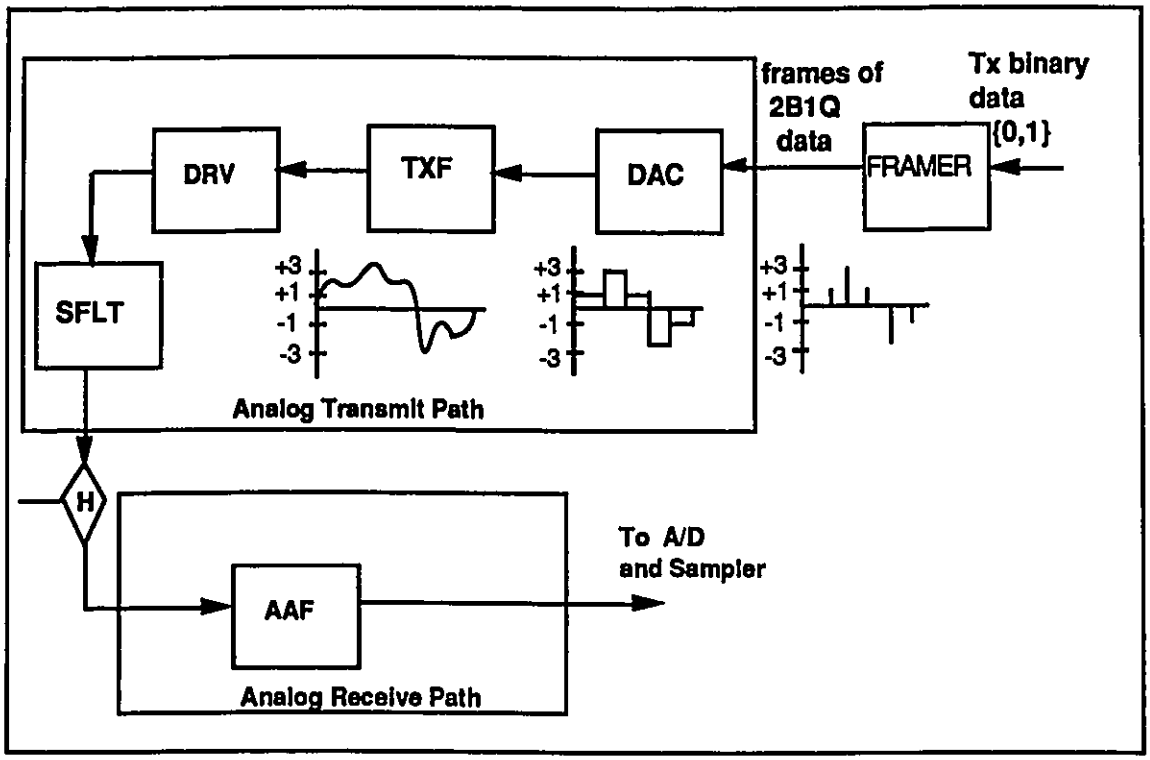

Figure 3.2: Framing and Analog Front End Processing

3.3.Conversion and Input Sampling

The A/D conversion of the analog signal at the input of the receiving end of the transceiver is performed via a sigma-delta modulator (SDM) as shown in Figure 3.3. The SDM is a high speed, low resolution A/D converter clocked at 10.24 MHz rate, with a one bit output. A second order SDM using two cascaded first order structures is used such that the input to the first modulator is the analog signal itself, while the input to the second modulator is the quantization noise of the first modulator. The output is such that the quantization noise of the first modulator is cancelled and only the noise of the second modulator remains. The output of the SDM is fed to a decimator block with a filter that

eliminates the quantization noise from the A/D conversion, retaining the required received input signal.

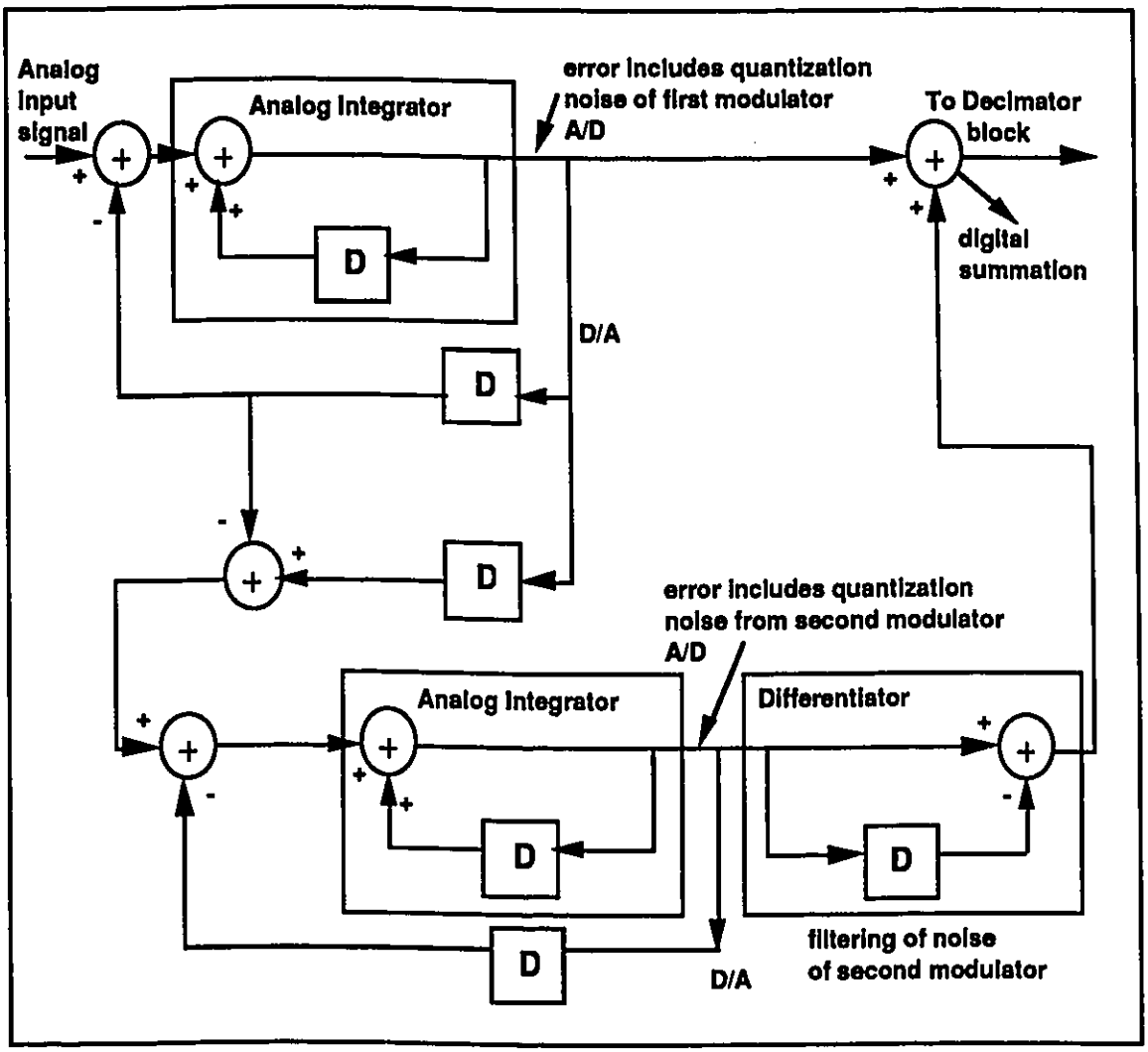

Figure 3.3: Sigma-Delta Modulator Structure

3.4.Decimate/Interpolate Block

This block converts the one bit SDM output generated at the high rate of 10.24 MHz to a multibit word output at the decimated rate of 80 KHz. This block consists of three parts: an integrator, a COMB filter and an interpolator, as shown in Figure 3.4.

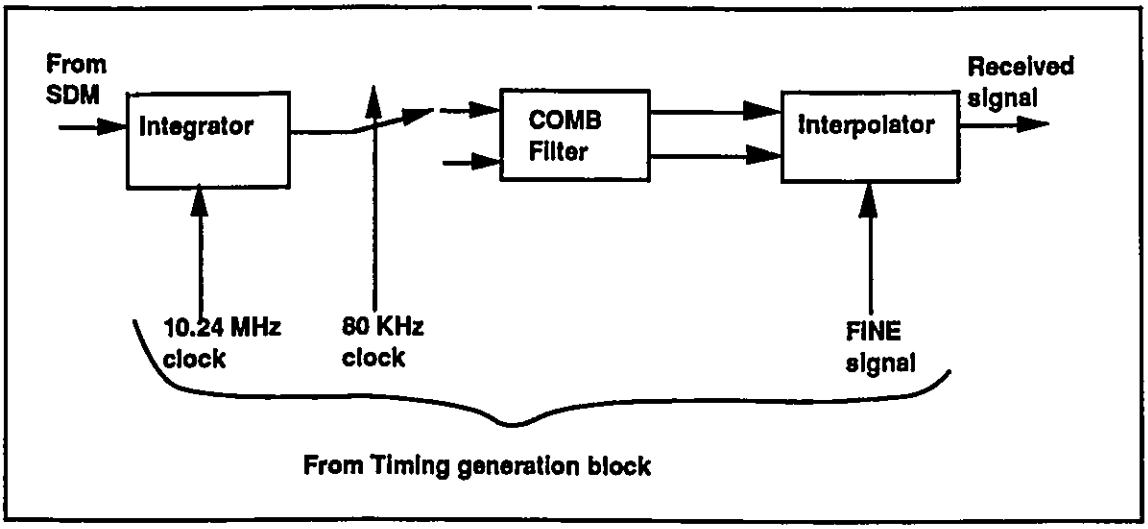

Figure 3.4: Decimation/Interpolation Blocks

3.4.1.Integrator Block

This block consists of three cascaded integrators as shown in Figure 3.5. It runs at the upsampled rate of 10.24 MHz with overall transfer function $1/(1-D)^3$, where D is the unit delay at the 10.24 MHz rate. This part implements the digital lowpass filter function on the SDM output.

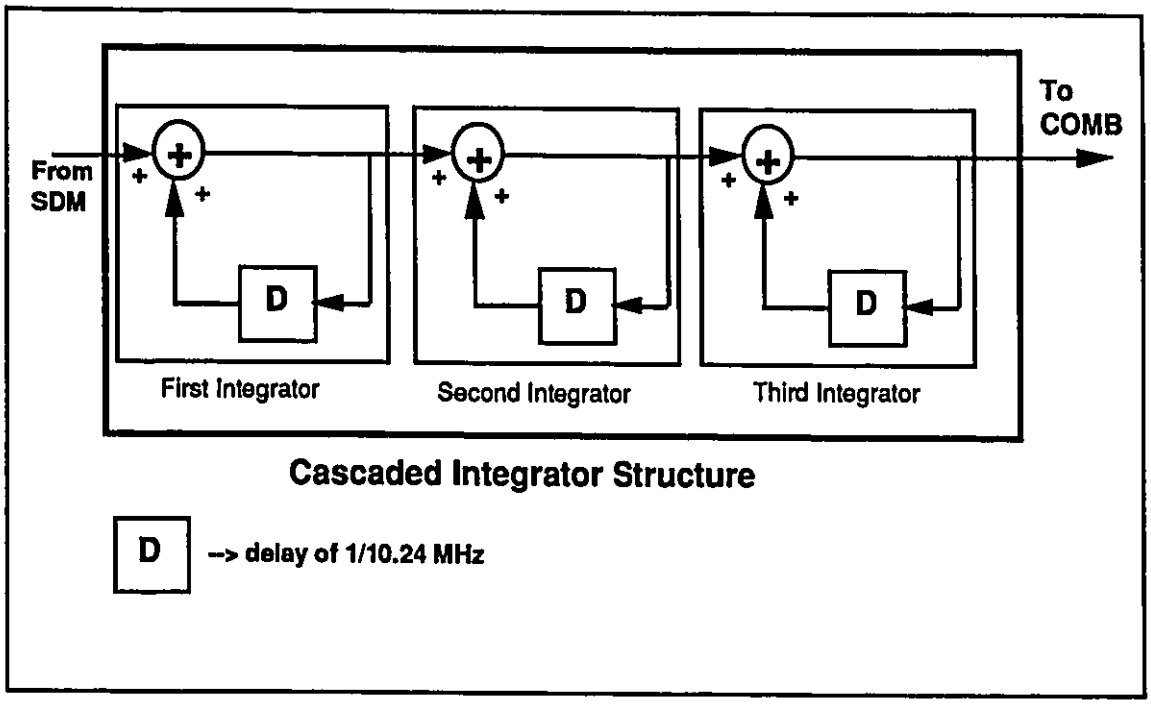

Figure 3.5: Integrator Structure

3.4.2.COMB Filter

This second stage is a sampler that takes the integrator output at the upsampled rate as input and outputs the signal at the decimated rate of 80 KHz (symbol rate). It reads in the input samples and passes them through the COMB filters which consist of two low speed differentiators. This structure of a high speed integrator, followed by the low-speed differentiators is termed the CIC - Cascaded Integrator COMB decimation filter.

3.4.3.Interpolator Block

This block is to allow for smaller phase jumps. In the NT mode, coarse phase jumps of 98 nanoseconds apart can be mostly used because the transmit and receive clocks are locked together. Also, the main high rate clock which is used to generate the transmit and receive clocks in the NT mode, is derived from an oscillator which can be very jittery.

This is mainly due to economical limitations. To reduce this effect, a large phase jump is used. However, we must ensure that the jumps allowed are not too large so as not to be able to track the incoming signal. For the LT mode, the phase jumps required are much finer since in this case, the transmit clock is derived from the central office switch and is very stable such that in steady-state fine jumps are sufficient. Moreover, the transmit clock is not affected by the phase jump, whereas the receive clock is because it is synchronized to the incoming signal. Hence, when a phase jump occurs, a large error is produced at the echo cancellation block since the coefficients correspond to the previous phase, and the new sampled echo is at the new phase. For these reasons, coarse jumps are not suitable at the LT side, and interpolation of the received signal becomes necessary to allow smaller phase jumps. Interpolation is performed on the pair of output by the COMB filter section.

3.5.Receive Forward Filter

This block has the following objectives:

- low-frequency noise reduction
- Pulse tail reduction of echo and transmission paths impulse response
- Shaping of precursor part of pulse to achieve maximum eye opening for optimum phase recovery.

It consists of a compromise forward equalizer that uses a simple fixed FIR structure that works at the symbol rate. The design of this filter is critical because it affects the behavior of the other DSP blocks , especially timing recovery. This block is not investigated in this thesis; a pre-developed filter that has been tested using this transceiver and found satisfactory but not optimum, was used in the characterization of the timing recovery functions.

3.6.DFE/ARC Blocks

Equalization of the postcursor part of the far end pulse is performed with an adaptive DFE structure. It consists of an N-tap transversal filter. Its operation is based on the LMS algorithm as follows:

$$ISI(k) = \sum_{i=1}^{N} h_i(k)*DRX(k-i) \tag{3.1}$$

$$h_i(k+1) = h_i(k) + \beta * ERR(k)*DRX(k-i) \quad i=1,2,...,N \tag{3.2}$$

$$ERR(k) = X(k) - ARC(k-1)*DRX(k) \tag{3.3}$$

$$ARC(k) = ARC(k-1) + \beta_{TDFE} * ERR(k)*DRX(k) \tag{3.4}$$

ISI(k) is the estimated postcursor interference at time k, to be subtracted from the received signal. $h_i(k)$ are the DFE coefficients updated at every symbol. β_{TDFE} is the adaptation step-size. DRX(k) is the current estimated received data symbol. ERR(k) is the LMS error used for all adaptive algorithms. X(k) is the received signal after echo and ISI cancellation. The adaptive reference control output (ARC) is the estimated channel gain calculated at every symbol. This can be seen as the zero-th tap of the DFE, except that it doesn't directly affect the signal. This block is used by the slicer and timing recovery blocks.

3.7.Automatic Threshold Quantizer

The automatic threshold quantizer slices the signal to output a 2B1Q symbol. The decisions of the slicer are used by the timing recovery and equalization blocks. The ATQ uses the estimate of the channel gain from the ARC block for the thresholds as follows:

$$DRX(k) = \begin{cases} +3 & \text{if } X(k) > 2*ARC(k-1) \\ +1 & \text{if } 0 \le X(k) \le 2*ARC(k-1) \\ -1 & \text{if } -2*ARC(k-1) \le X(K) < 0 \\ -3 & \text{if } X(k) < -2*ARC(k-1) \end{cases} \quad (3.5)$$

3.8.Echo Cancellation

The echo canceller estimates the echo path impulse response, calculating a replica of the echo signal predicted to be present in the received signal. The echo signal is then directly subtracted from the received signal. The echo canceller structure comprises of a transversal, IIR and memory EC blocks. These are illustrated in Figure 3.6 and each section is briefly explained below.

3.8.1.Transversal EC

The transversal echo canceller uses the LMS algorithm to cancel the main part of the echo. Its operation is formulated as follows:

$$TEC(k) = \sum_{i=1}^{NTEC} C_i(k)*DTX(k-i+1) \quad (3.6)$$

$$C_i(k+1) = C_i(k) + \beta_{TEC} * ERR(k) * DTX(k-i+1) \quad i=1,2,...,NTEC \quad (3.7)$$

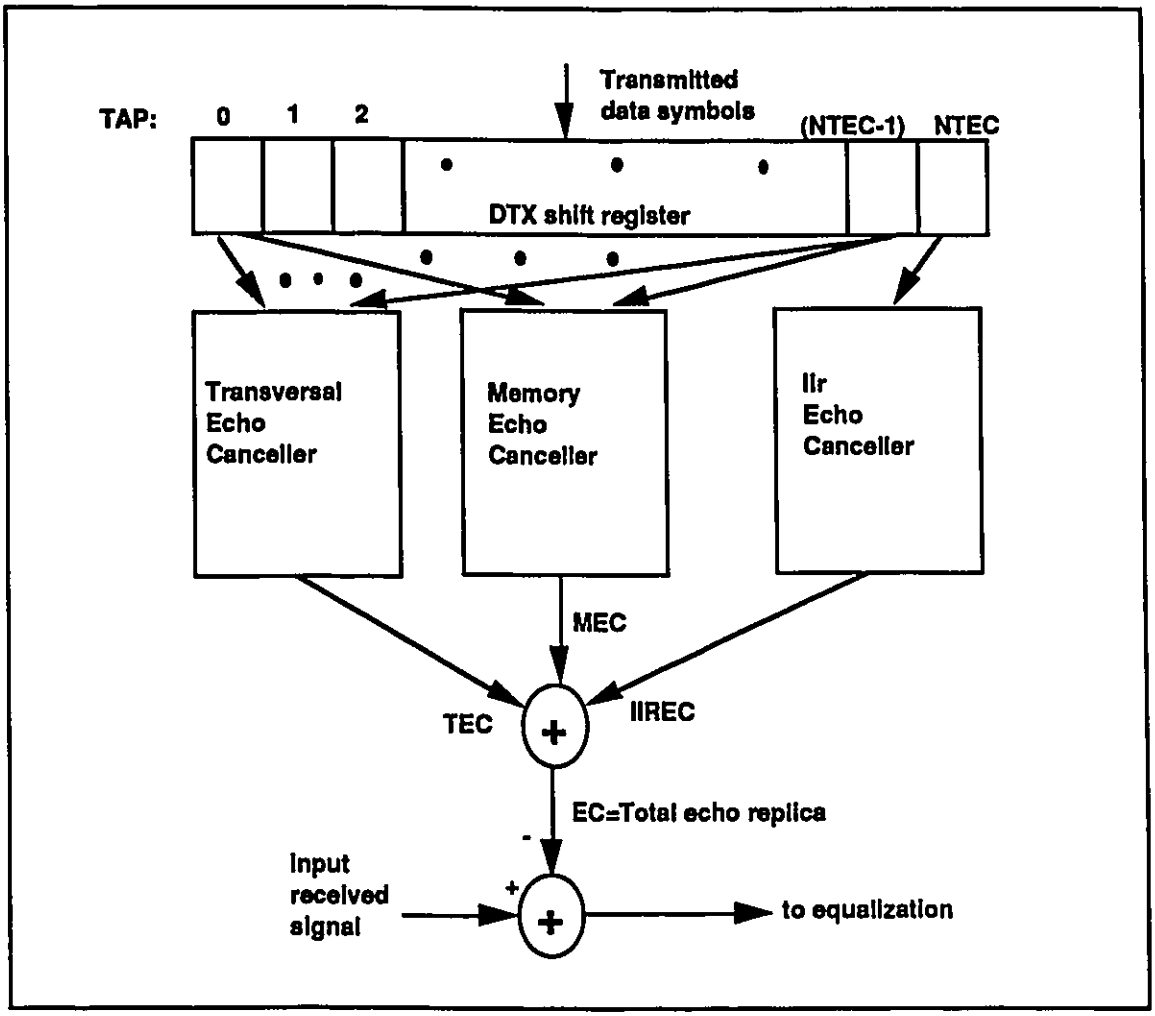

Figure 3.6: Echo Canceller Structure

TEC(k) is the transversal filter estimate of the echo signal, based on the update of NTEC coefficients $C_i(k)_{\ i=1 \text{ to NTEC}}$. DTX(k) is the transmitted data symbols. β_{TEC} is the adaptation step size and ERR(k) is the LMS error calculated as in (3.3).

3.8.2.IIR EC

The IIR (infinite-impulse response) echo canceller is used to eliminate the tail part of the echo caused mainly by the far end and near end transformers. It allows the use of a shorter transversal filter, and works on the echo tail that is not covered by the transversal EC. It approximates the decay of the pulse tail using an exponential function.

$$IIREC(k) = EC\_F(k) * EC\_H(k) \tag{3.8}$$

$$EC\_F(k) = EC\_F(k-1) * P_{IIREC} + DTX(k - NTEC) \tag{3.9}$$

IIREC(k) is the echo replica, EC_F(k) is the exponential function and EC_H(k) is the coefficient. P_{IIREC} is a fixed pole between 0 and 1, chosen based on the transformer inductance of the loop coupling circuit. The input to the filter at time k is the transmitted data symbol DTX(k - NTEC) delayed by the number of taps in the transversal EC, thus covering the tail part which the transversal EC filter does not.

3.8.3.Memory EC

The memory echo canceller is used to remove the nonlinear component of the echo signal which cannot be cancelled by the transversal EC. From Figure 3.6, MEC(k) is the echo estimate at symbol k, estimated using ERR(K), the residual signal error as given in (3.3).

3.8.4.EC FIFO

As explained above, in the LT mode the transmit clock is derived from the switch and is fixed whereas the receive clock is derived from the signal and is adjustable. This can

cause an offset between the two clocking sides such that a new EC coefficient is needed for both the transversal and memory echo cancellers. This is resolved by using a FIFO buffer that can accommodate the storage of the extra needed samples.

3.9.Timing Recovery

This block is responsible for the synchronization of the two digital subscriber loop ends as previously indicated. A symbol-rate technique based on the Mueller and Müller method in conjunction with a digital phase locked-loop (DPLL) is employed in this transceiver. This algorithm and loop will be described in full detail in the next chapters and four timing estimates based on this technique and compatible with this system will be studied and compared.

The function of this timing recovery section is to search for the optimum phase for maximum eye opening and generate a subsequent control signal to the timing generation block to adjust the clock as required. This is performed by fixed phase jumps of ±98 ns.

As mentioned previously, the fixed phase jumps cause significant EC degradation. One possible alternative is to perform the phase adjustments during the synchword.

3.10.Timing Generation

This block provides the clock signals required for both modes, LT and NT, as shown in Figure 3.7. It takes the phase jump signal output from the timing recovery as input and provides the desired clocking signals to the input sampler blocks (SDM). For LT mode, the transmit clock is derived from the central office, the receive clock is derived from the transmit clock but adjusted according to the phase jump. For this mode, we need to do smaller phase jumps. For the NT mode, the transmit clock is directly taken from the NT receive clock. The NT receive clock is generated with a digital count-down counter

that is loaded depending on the phase jump. The high rate clock is at 10.24 MHz with a nominal ratio of 128 (counter load of 127) such that receive clock rate is 10.24 MHz / 128 = 80 KHz. This is the initial load for zero phase movements. For positive and negative coarse phase adjustments, the initial load is decreased and increased by one respectively.

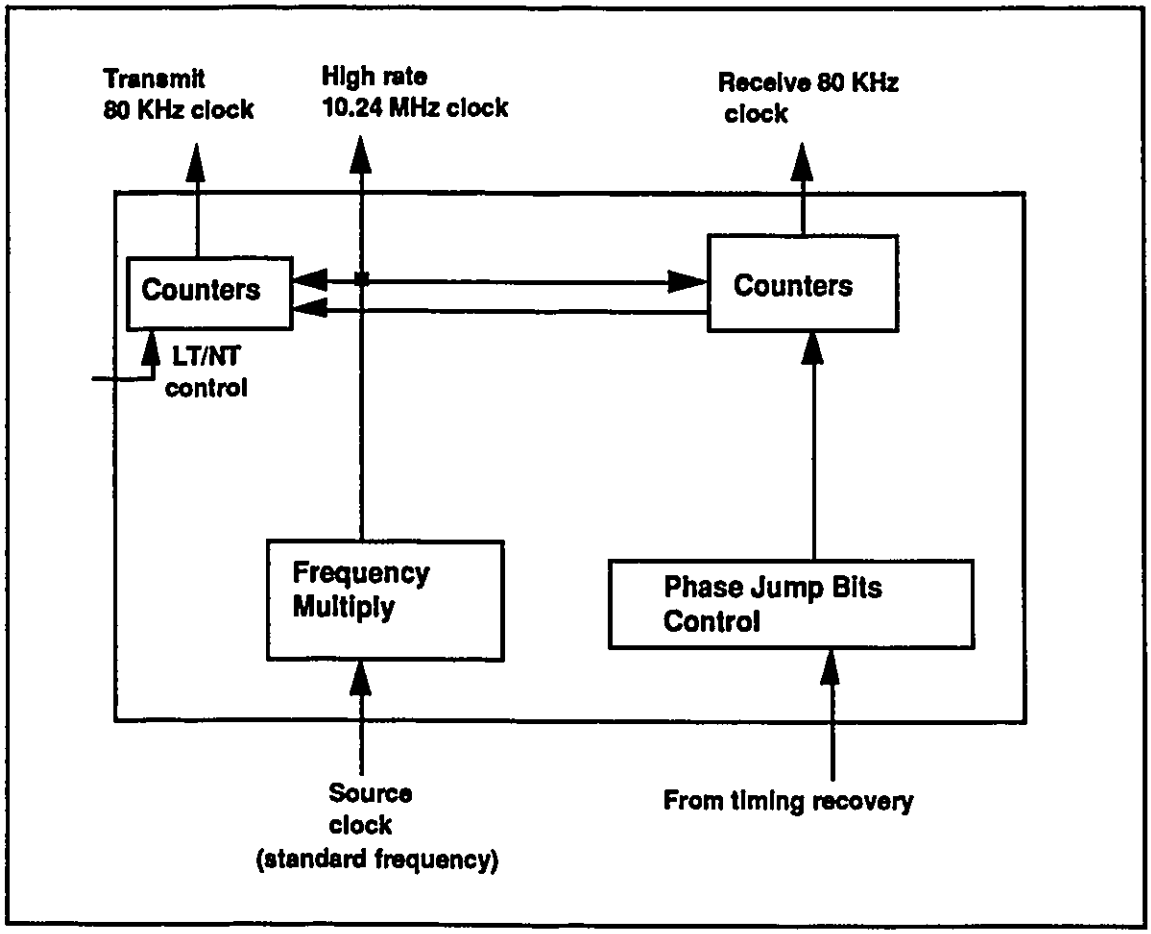

Figure 3.7: Timing Generation Block

3.11.Summary

In this chapter, we described a full-duplex 2B1Q transceiver applicable to digital subscriber loops. The full architecture has been simulated using the CAPSIM/BLOSIM

software package, with half-duplex transmission and EC modes having been completely tested and fully functional. The code is structured to allow full flexibility and supports both half and full-duplex transmission. The integration of all the blocks for full-duplex operation has not been done. The simulation setup will be described in greater detail in the coming chapters when it is used in the half-duplex mode to characterize the performance of the timing recovery. In the next chapter, the timing recovery algorithm will be introduced. Then, four timing functions based on the M&M method and suitable for 2B1Q line coding scheme will be compared theoretically. Finally, the timing recovery loop introduced above will be described and the performance of each timing function in the practical TR loop, investigated.

CHAPTER 4

Timing Recovery Theory

4.1.Introduction

In this chapter, a symbol rate sampled-data technique for timing recovery will be described in which a timing reference is defined from samples of the channel impulse response and estimated from samples of the actual received signal. Next, a practical scheme in which these timing estimates are used in an adaptive control loop to achieve timing recovery will be outlined.

4.2.Review Of The Mueller and Müller Method For Timing Recovery

In the classical 1976 paper [MueM76], Mueller and Müller described an efficient method for timing recovery applicable to synchronous digital baseband receivers, using binary or multilevel PAM, as well as partial response signals. The algorithm yields a simple and direct approach in which timing information is obtained from symbol rate samples of the input signal and estimated data symbols after quantization. In this section we will describe this algorithm in detail, outlining all the steps required to derive a near-optimal timing estimate. We will also define the control loop in which these timing

estimates can be used to iteratively adjust the input sampling phase such that the timing offset, with respect to a given steady-state sampling criterion, is minimum. The following will consist mainly of a review of the Mueller and Müller timing recovery technique and we will heavily rely on [MueM76] for the content. First, a timing function to be used as the reference for the steady-state sampling phase is derived based on some aspect of the channel impulse response. Then, a sub-optimum estimate of this function, expressed in terms of available samples of the input signal and its corresponding estimated data values is computed.

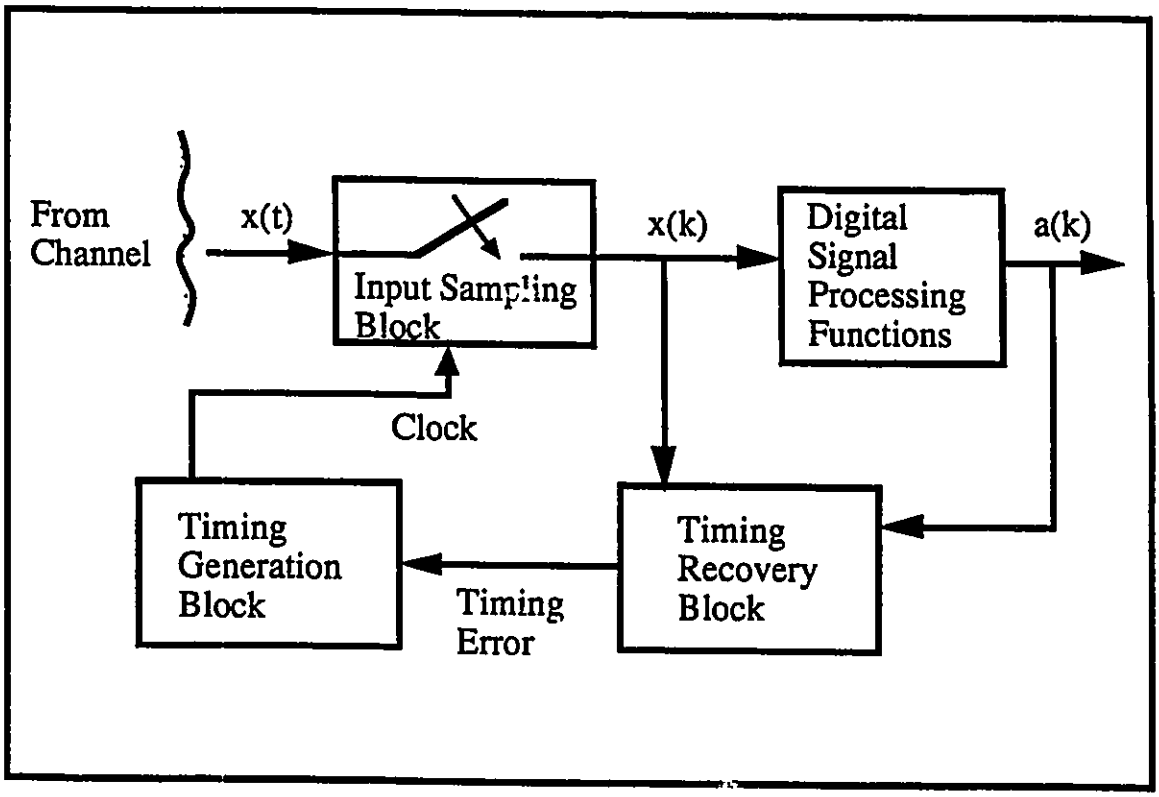

Figure 4.1: Digital Receiver Architecture

Given the receiver architecture in Figure 4.1, we define the following terms:

$$x(t) = \sum_{j=-\infty}^{\infty} a_j h(t - jT) \tag{4.1}$$

$x(t)$ is the input to the receiver of a synchronous baseband data transmission system with a noiseless channel of overall impulse response $h(t)$. The channel is assumed to be band-limited such that its impulse response covers several bauds in length. The a_j's are equally spaced data symbols assumed to be uncorrelated and equiprobable. After sampling, the received signal becomes

$$x_k = \sum_{j=-\infty}^{\infty} a_{k-i}\, h_i \tag{4.2}$$

where $x_k = x(\tau + kT) \Rightarrow$ samples of the received signal (with $i = k - j$)

$\tau \Rightarrow$ input sampling phase

$T \Rightarrow$ symbol interval

$h_i = h(\tau + iT) \Rightarrow$ samples of the channel impulse response

Ideally, the sampling phase τ should be chosen so as to minimize error probability. However, in practical implementation, a more convenient less constricting performance measure can be used. A reasonable alternative criterion would be to minimize the intersymbol interference (ISI) or the mean square error (MSE). These objectives can be achieved if the peak or mean square distortion given (4.3) and (4.4) respectively, are minimized.

$$\text{MSD} = D(\tau) = \sum_{i}' |h_i| \tag{4.3}$$

$$\text{PD} = \varepsilon(\tau) = \sum_{i}' h_i^2 \tag{4.4}$$

A prime (') on a summation implies absence of zero term such that i≠0. For a sampling phase τ such that $D(\tau) = \varepsilon(\tau) = 0$, the channel is said to be ideal, and distortion free transmission is achieved. Thus, the steady-state phase should be chosen so as to minimize one of the measures in (4.3) and (4.4). In the Mueller and Müller algorithm, a timing function $f(\tau)$ is defined such that its root lies close to the minimum of one of the performance measures.

$f(\tau)$ is given as a linear combination of the samples of the impulse response $h(t)$, in the following form:

$$f(\tau) = \sum_{i}' u_i h_i = \mathbf{u}^T\mathbf{h} \tag{4.5}$$

where (^{T}) denotes transpose, and the coefficients u_i's are dimensionless.

The timing function employed must be based on some reliable aspect of the channel impulse response. Basically, we must choose a reference for our timing recovery based on some characteristic of $h(t)$ that is relatively insensitive to channel imperfections such as line length and bridged taps. A commonly used reference in digital line transmission is the zero-crossing of the first precursor. For DSL, it was shown [TzHM86] , that only the postcursor part of the pulse is affected by line impairments and that the precursor zero-crossing is preserved by the line. For this case, $f(\tau)$ is given by :

$$f(\tau) = h_{-1} \tag{4.6}$$

which yields from (4.5)

$$\begin{cases} u_i = 1 & i = -1 \\ = 0 & \text{otherwise} \end{cases}$$

This zero crossing can be introduced at the transmitter by digital coding or at the receiving end by means of analog filtering [TzHM86]. Note that the derived phase for (4.6) is such

that the first precursor is zero, thus reducing ISI distortion. The optimal phase τ_0 is reached when $f(\tau = \tau_0) = 0$.

Since in practice, the samples of the impulse response of the channel $h(t)$ are not directly available to determine $f(\tau)$, we must obtain an estimate of $f(\tau)$ based on the received signal samples $\mathbf{x}_k$ and the decoded data symbols $\mathbf{a}_k$. The proposed timing estimate z_k in [MueM76] is based on a linear combination of $\mathbf{x}_k$ and some weighting coefficients $\mathbf{g}_k$ chosen as a function of the data symbols $\mathbf{a}_k$, such that the expected value of z_k equals $f(\tau)$. We then have:

$$E\{ z_k \} = f(\tau) \tag{4.7}$$

with $$z_k = \mathbf{g}_k^T \mathbf{x}_k \tag{4.8}$$

The vectors $\mathbf{g}_k$ and $\mathbf{x}_k$ contain each m values, defined as follows:

$$\mathbf{x}_k = \begin{bmatrix} x_{k-m+1} \\ x_{k-m+2} \\ \cdot \\ \cdot \\ x_k \end{bmatrix} \quad \text{and} \quad \mathbf{g}_k = \begin{bmatrix} g_{k-m+1} \\ g_{k-m+2} \\ \cdot \\ \cdot \\ g_k \end{bmatrix}$$

with $$\mathbf{a}_k = \begin{bmatrix} a_{k-m+1} \\ a_{k-m+2} \\ \cdot \\ \cdot \\ a_k \end{bmatrix}$$

Another possible form for z_k would be a function of the error signal e_k such that

$$z_k = \mathbf{g}_k^T \mathbf{e}_k \tag{4.9}$$

with $$\mathbf{e}_k = \mathbf{x}_k - h_0 \mathbf{a}_k \tag{4.10}$$

where h_0 is the gain factor associated with the channel. Note that h_0 strongly depends on the overall system attenuation and the sampling phase τ. The channel gain can be easily estimated using an Automatic Reference Control (ARC) block.

Basically, we must choose a $\mathbf{g}_k$ such that a low-variance estimate z_k is obtained. The variance of z_k is a measure of the mean square error in estimating $f(\tau)$. The expected value of z_k in (4.8) and (4.9) as well as an expression for the variance of z_k are derived in [MueM76]. For convenience, these expressions are summarized below in equations (4.11) through (4.15), and some important terms are defined.

$$\text{If } z_k = \mathbf{g}_k^T \mathbf{x}_k \qquad E\{ z_k \} = \mathbf{h}^T E\{ \mathbf{A}_k \mathbf{g}_k \} \tag{4.11}$$

$$\text{If } z_k = \mathbf{g}_k^T \mathbf{e}_k \qquad E\{ z_k \} = \mathbf{h}^T E\{ \mathbf{A}_k \mathbf{g}_k \} - h_0 E\{ \mathbf{g}_k^T \mathbf{a}_k \} \tag{4.12}$$

$$\begin{aligned} \mathrm{Var}\{ z_k \} &= E\{ z_k^2 \} - E^2\{ z_k \} \\ &= E\{(\mathbf{g}_k^T \upsilon_k)^2\} - E^2\{ \mathbf{g}_k^T \upsilon_k \} + E\{ \mathbf{g}_k^T \mathbf{Q} \mathbf{g}_k \} \end{aligned} \tag{4.13}$$

Where $\upsilon_k = E\{\mathbf{x}_k / \mathbf{a}_k \} = \mathbf{A}_k^T \mathbf{h}$ is the conditional expectation of $\mathbf{x}_k$ given $\mathbf{a}_k$. (4.14)

with $\mathbf{h} = \begin{bmatrix} h_{1-m} \\ \vdots \\ h_{-1} \\ h_0 \\ h_1 \\ \vdots \\ h_{m-1} \end{bmatrix}$ a vector of the truncated system impulse response of $(2m\text{-}1)$ samples,

and $\mathbf{A}_k = \begin{bmatrix} a_k & 0 & \cdots & 0 \\ a_{k-1} & a_k & \cdots & 0 \\ \cdots & & & \cdots \\ a_{k-m+1} & a_{k-m+2} & \cdots & a_k \\ 0 & a_{k-m+1} & \cdots & a_{k-1} \\ \cdots & & & \cdots \\ 0 & 0 & \cdots & a_{k-m+1} \end{bmatrix}$ $(2m\text{-}1)\times m$ matrix

$$\mathbf{Q} = E\{a_k^2\} \sum_p h_{i-p} h_{j-p} = q_{ij} \quad ,p \in \{1,m\}, \quad (\text{an } m\times m \text{ matrix}) \tag{4.15}$$

Combining equations (4.5) and (4.7), we obtain:

$$E\{ z_k \} = f(\tau) = \mathbf{h}^T \mathbf{u} \tag{4.16}$$

Comparing the above to (4.11) and equating the right side of both equations, we get:

$$\mathbf{u} = E\{\mathbf{A}_k \ \mathbf{g}_k\} \tag{4.17}$$

Given a $\mathbf{u}$, the above equation does not result in a single solution for $\mathbf{g}_k$, but allows many possibilities. The optimal choice of $\mathbf{g}_k$ is obviously the one which will yield the lowest variance. The best estimate is obtained by solving (4.17) such that the expression of the variance of z_k given in (4.13) is minimized. However, in [MueM76], it was shown that such a solution is not practical, in that the evaluation of the optimal $\mathbf{g}_k$ would be too involved and will only be optimum for one particular channel. For a simple, channel independent solution, a sub-optimum estimate must be used. Equation (4.17) can be expressed as

$$\mathbf{A}_k \ \mathbf{g}_k = \mathbf{u} + \mathbf{d}_k \tag{4.18}$$

where $\mathbf{d}_k$ is a vector with zero mean random variables. Substituting (4.14) and (4.18) in the variance expression of (4.13), we have:

$$\begin{aligned} Var\{ z_k \} &= E\{ (\mathbf{h}^T \mathbf{A}_k \ \mathbf{g}_k)^2 \} - E^2\{ \mathbf{h}^T \mathbf{A}_k \ \mathbf{g}_k \} + E\{ \mathbf{g}_k^T \mathbf{Q} \ \mathbf{g}_k \} \\ &= E\{ (\mathbf{h}^T (\mathbf{u} + \mathbf{d}_k))^2 \} - E^2\{ \mathbf{h}^T(\mathbf{u} + \mathbf{d}_k) \} + E\{ \mathbf{g}_k^T \mathbf{Q} \ \mathbf{g}_k \} \end{aligned}$$

Expanding and simplifying by setting $E\{\mathbf{d}_k\} = 0$, we finally get:

$$Var\{ z_k \} = E\{ (\mathbf{h}^T \mathbf{d}_k)^2 \} + E\{ \mathbf{g}_k^T \mathbf{Q} \ \mathbf{g}_k \} \tag{4.19}$$

To keep z_k a low variance estimate, $\mathbf{d}_k$ must be kept small. To solve for $\mathbf{g}_k$, we must satisfy (4.18) and the constraint of zero mean for $\mathbf{d}_k$. The system of (4.18) yields $(2m\text{-}1)$ equations for the solution of the m elements in $\mathbf{g}_k$. The method for solving for $\mathbf{g}_k$ involves a heuristic approach where m equations are chosen and then substituted back into the $(2m\text{-}1)$ set to verify that the other equations are satisfied. In [MueM76] paper, some helpful hints are given for a tentative solution where the m equations are chosen symmetrically around the center of the set in (4.18) such that it includes a smaller $m \times m$ square matrix from the center of the original rectangular $(2m\text{-}1)\times m$ $\mathbf{A}_k$ matrix. Also, for a first try, the components of $\mathbf{d}_k$ can be set to zero, thus eliminating its significant effect on the variance. Varying the $\mathbf{d}_k$ elements and choosing a different set of equations are two possibilities for a second iteration or for the derivation of a different z_k. The center component of $\mathbf{d}_k$ must be zero in order for the variance to be independent of the h_0 term.

Even though the above method seems somewhat loosely defined, allowing many possible choices for the timing estimates for the same $\mathbf{u}$ coefficients, it proves nonetheless to be quite efficient in determining good timing estimates with near-minimum variance. Obviously not all estimates derived for the same $f(\tau)$ will be equally good, even though all do satisfy the condition of (4.18). In such a case, the variances could be compared as a measure of the error in each z_k in estimating $f(\tau)$. However, a lower variance does not always ensure the better performance of a timing estimate. In a practical system, there are many parameters involved and the interaction between the various DSP functions is strong, such that many other factors contribute to the performance of the timing recovery loop. There could exist some cases where an estimate of higher variance might perform better than one with a lower variance. An instance of this will be shown in the following sections.

Presently, we have limited our discussion to the noiseless channel case. Since noise is present in any practical communication system, it is important to study its effect on the timing estimates and to get a feel for how it deteriorates the system's performance as

opposed to the other system parameters. In [MueM76], the case of additive white gaussian noise where the received signal is:

$$x_k = \sum_{i=-\infty}^{\infty} a_{k-i} h_i + N_k \tag{4.20}$$

is considered. The noise is assumed to be a stationary, zero mean random process, with power σ^2 . The expected value of z_k is not affected by the additive noise, and the variance expressions, given in (4.13) and (4.19), become:

$$\mathrm{Var}\{z_k\} = E\{ (\mathbf{g}_k^T \upsilon_k)^2 \} - E^2\{\mathbf{g}_k^T \upsilon_k\} + E\{\mathbf{g}_k^T \mathbf{Q} \mathbf{g}_k\} + \sigma^2 E\{\mathbf{g}_k^T \mathbf{I} \mathbf{g}_k\} \tag{4.21}$$

$$\mathrm{Var}\{z_k\} = E\{ (\mathbf{h}^T \mathbf{d}_k)^2 \} + E\{\mathbf{g}_k^T \mathbf{Q} \mathbf{g}_k\} + \sigma^2 E\{\mathbf{g}_k^T \mathbf{I} \mathbf{g}_k\} \tag{4.22}$$

where **I** is the identity matrix.

(4.21) and (4.22) are identical to equations (4.13) and (4.19) with the exception of the last term obtained by replacing the matrix **Q** by $(\mathbf{Q} + \sigma^2 \mathbf{I})$. Looking at the added term, the effect of channel noise could be a critical consideration in the choice of z_k . For a noisy channel, z_k must be selected so that the expected value of the squared weighting vector $\mathbf{g}_k$ is minimum. This factor is especially important during steady-state. After convergence of the equalizer, most of the ISI will be cancelled and the noise term becomes the dominant factor in the variance.

Finally, we would like to note that the timing estimate could be improved if a longer memory is used. However, the derivation of estimates of $m > 2$ could get quite involved and in general, the added complexity does not justify the small improvement in performance. Also, by increasing m, we increase error propagation. This is obvious since the correlation factor between succeeding estimates will be much stronger for longer memory length. Since estimates of $m \leq 2$ have been proven to be sufficiently accurate in

practice with near-minimum variance [MueM76], timing schemes using larger memory become even less attractive when the added complexity and consequently cost of implementation is considered.

Now that the estimates have been derived, we will see how to use these estimates to iteratively adjust the phase (stochastic adjustment algorithm). The control loop needed to perform the phase adjustments can be represented by the following:

$$\tau_{k+1} = \tau_k - \gamma_k z_k(\tau_k) \tag{4.23}$$

where γ_k is the adaptation step-size.

Given that the optimal timing phase is defined at the zero-crossing of $f(\tau)$, if γ_k is chosen based on the procedure such that it eventually decreases to zero at steady-state, then (4.23) will completely stop updating once τ_0. is reached. However for the realistic case where γ_k = constant, then the instantaneous phase corrections in general will not be zero, and some steady-state timing jitter will result. This is due to the dependence of the variable z_k on the timing phase, the channel distortion and additive noise, and the statistics of the data symbols. In this case, the steady-state timing error will fluctuate randomly around zero such that the average of the phase adjustments is zero. A typical timing recovery loop based on this stochastic algorithm will be defined later on and the resulting steady-state jitter as it relates to a particular timing estimate, will be investigated.

In this chapter, we have described the Mueller and Müller method for symbol rate timing recovery. The steps for the derivation of a sub-optimal timing estimate for a timing reference based on the channel impulse response have been given in detail. In the next chapter, we will study four timing estimates based on the timing function in (4.6).

CHAPTER 5

Theoretical Analysis of the Timing Estimates

5.1.Introduction

In this chapter, we use the Mueller and Müller algorithm described in chapter 4 to derive four timing estimates suitable for implementation in the ISDN U-transceiver for 2B1Q code. The timing recovery loop in which these estimates are used is defined and the performance of each timing estimate within this loop is investigated for practical channel responses. The effect of the channel characteristics, additive noise, finite average length and dead-zone threshold on the performance of the timing estimates is analyzed and some theoretical results are obtained. Finally, a linearized model of the timing recovery loop is developed and the corresponding timing jitter spectrum derived.

5.2.Timing Estimates Based On The M&M Algorithm For Timing Recovery

The timing function to be used in this thesis is $f(\tau) = h_{-1}$. As mentioned previously, the above function is a good choice for our timing reference because it is

unaffected by the channel imperfections. This timing function is plotted in Figure 5.1 for various loops which will be used to obtain our results in both the theoretical and simulation sections. The three loops are commonly used for standard testing in DSL, and provide a good representation of the various channels encountered in this type of application.

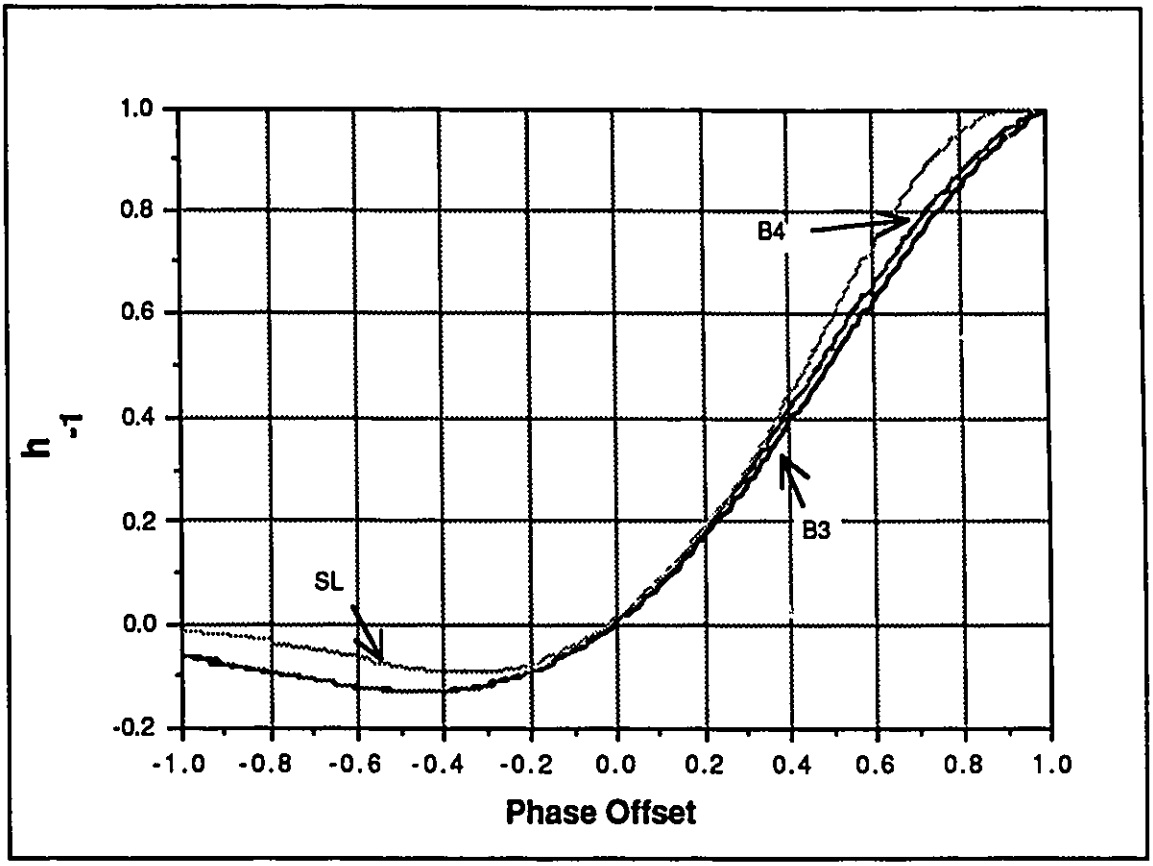

Figure 5.1: Timing Function $f(\tau)$ = h_{-1} For B3, B4 And SL Loops

B3 and B4 are standard Bellcore loops whose specifications can be found in [ANSI89], and SL is a short loop of length 0.5 KFT. The loops' exact configuration and pulse response are plotted in Figures 5.2 and 5.3 respectively. The frequency and pulse response of these channels were generated using two programs, HOFF and ISDN7 [Hung88]. The frequency response is first generated by the HOFF program which derives

the transfer function of a transmission line based on the analysis of a hybrid circuit model using ABCD parameters. The resulting file is then used as input to the ISDN7 program which performs transmit and receive filtering functions in the frequency domain and generates the equivalent time domain pulse response. The following filtering is included in the B3, B4 and SL loops pulse response:

- Pulse shaping function: 100% duty cycle rectangular pulse of 80 Kbauds
- Transmit filter: Second order Butterworth low-pass filter with 3 dB point at 80 KHz.
- Decimation filter: Decimation from 10.24 MHz to 80 KHz. The equivalent transfer function is $H(z) = (1 - z)^3 / (1 - z^{1/DF})^3$ where DF is the decimation factor of 128, and z is a delay by the 80 KHz clock period (symbol period).
- Receive forward filter: 3-tap symbol rate digital FIR filter.

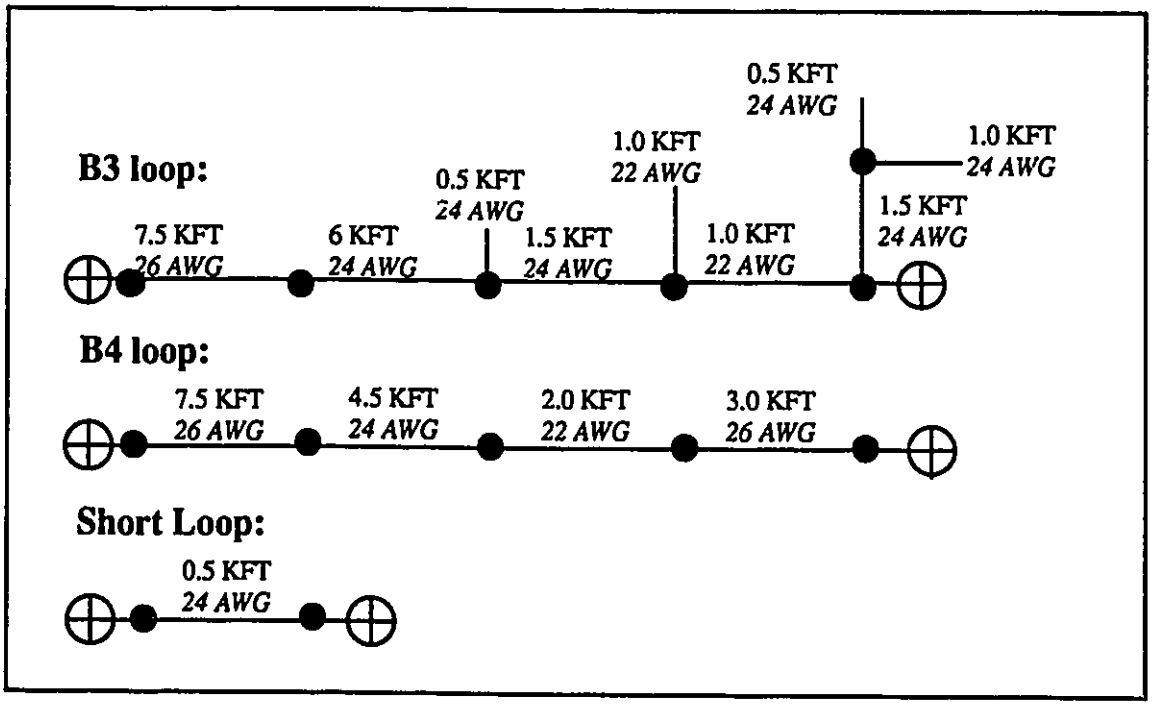

Figure 5.2: The Configurations Of The B3, B4 And SL Loops

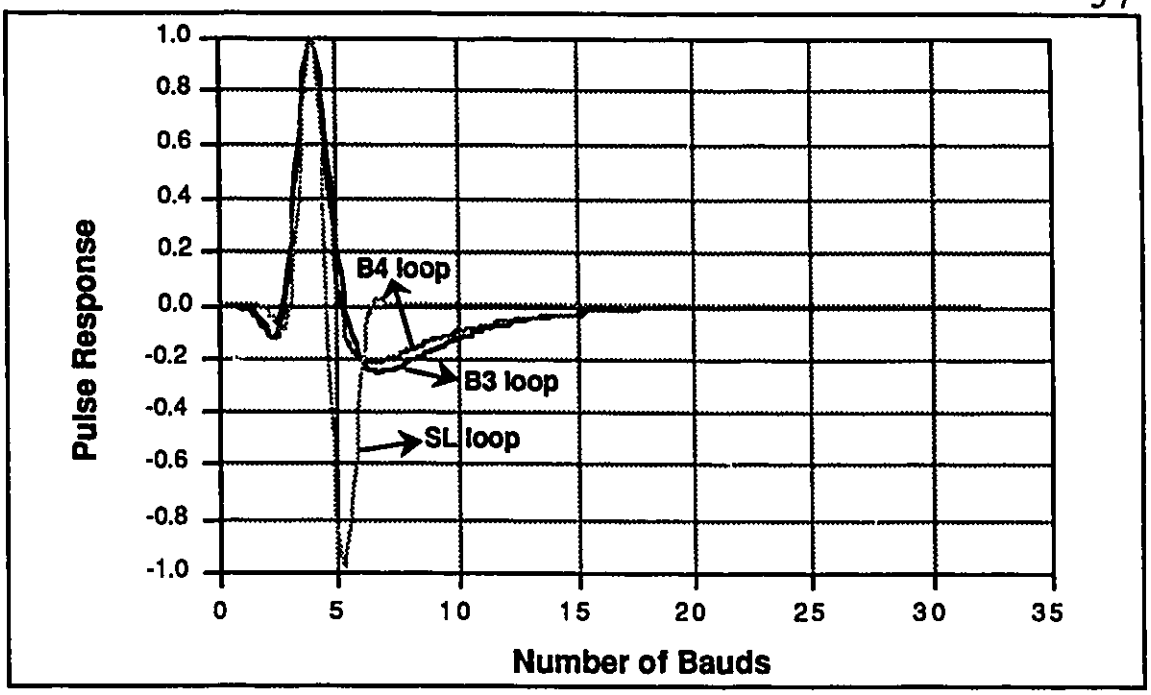

Figure 5.3: Pulse Response Of The B3, B4 And SL loops

Four timing estimates of memory length $m = 2$ and their corresponding variance expressions are given in Tables 5.1 and 5.2 respectively. These estimates were derived by B. Sayar [Saya89] using the method described in the previous section. z_{1k}, z_{2k} and z_{4k} are based on the received signal where as z_{3k} is a function of the error signal.

The $E\{z_{ik}\}_{i=1,2,3,4}$ is derived in Appendix A and shown to be equal to $f(\tau)$. In Appendix B, the variances for these estimates, given in Table 5.2 are derived. It is shown that although all four estimates satisfy (4.7) or (4.9) and (4.18), their variances differ. To be able to compare the four timing estimates and evaluate their performance, we need to specify the timing recovery loop in which they will be used in our simulation setup, and define all parameters essential to the derivation of any pertinent results.

i	z_{ik}
1	$\frac{(a_k^2 - 5)}{16}(a_k x_{k-1} - a_{k-1} x_k)$
2	$\frac{1}{(\sqrt{5}a_k + 5)}(-a_k x_{k-1} + a_{k-1} x_k)$
3	$a_k (x_{k-1} - h_0 a_{k-1}) \frac{1}{E\{a_k^2\}}$
4	$\frac{1}{E\{a_k^2\}}(a_k x_{k-1} - a_{k-1} x_k) + h_1$

Table 5.1: Four Timing Estimates Based On The Timing Function $f(\tau) = h_{-1}$

i	$\mathrm{Var}\{z_{ik}\}$
1	$3.125 \sum{}' h_i^2 + (h_1^2 - 3.125 h_1 h_{-1}) + 0.625 \sigma^2$
2	$7.25 \sum{}' h_i^2 - 8.25 h_1 h_{-1} + 2(h_1^2 + h_{-1}^2) + 1.4 \sigma^2$
3	$\sum{}' h_i^2 - 0.36 h_{-1}^2 + 0.2 \sigma^2$
4	$2 \sum{}' h_i^2 - 0.36 (h_1^2 + h_{-1}^2) + 0.4 \sigma^2$

Table 5.2: Expressions For The Variance Of The Timing Estimates

5.3.Timing Recovery Loop

The block diagram for the timing recovery loop used here is shown in Figure 5.4. The timing estimate generator implements any of the four timing estimates. It receives the input x_k and output a_k of the slicer as inputs, as well as the output of an ARC block which approximates the channel gain h_0, and the output of the first DFE tap which approximates the first postcursor sample h_1 when z_{3k} and z_{4k} are being evaluated respectively. The output of the timing estimate generator consists of the instantaneous value of the timing

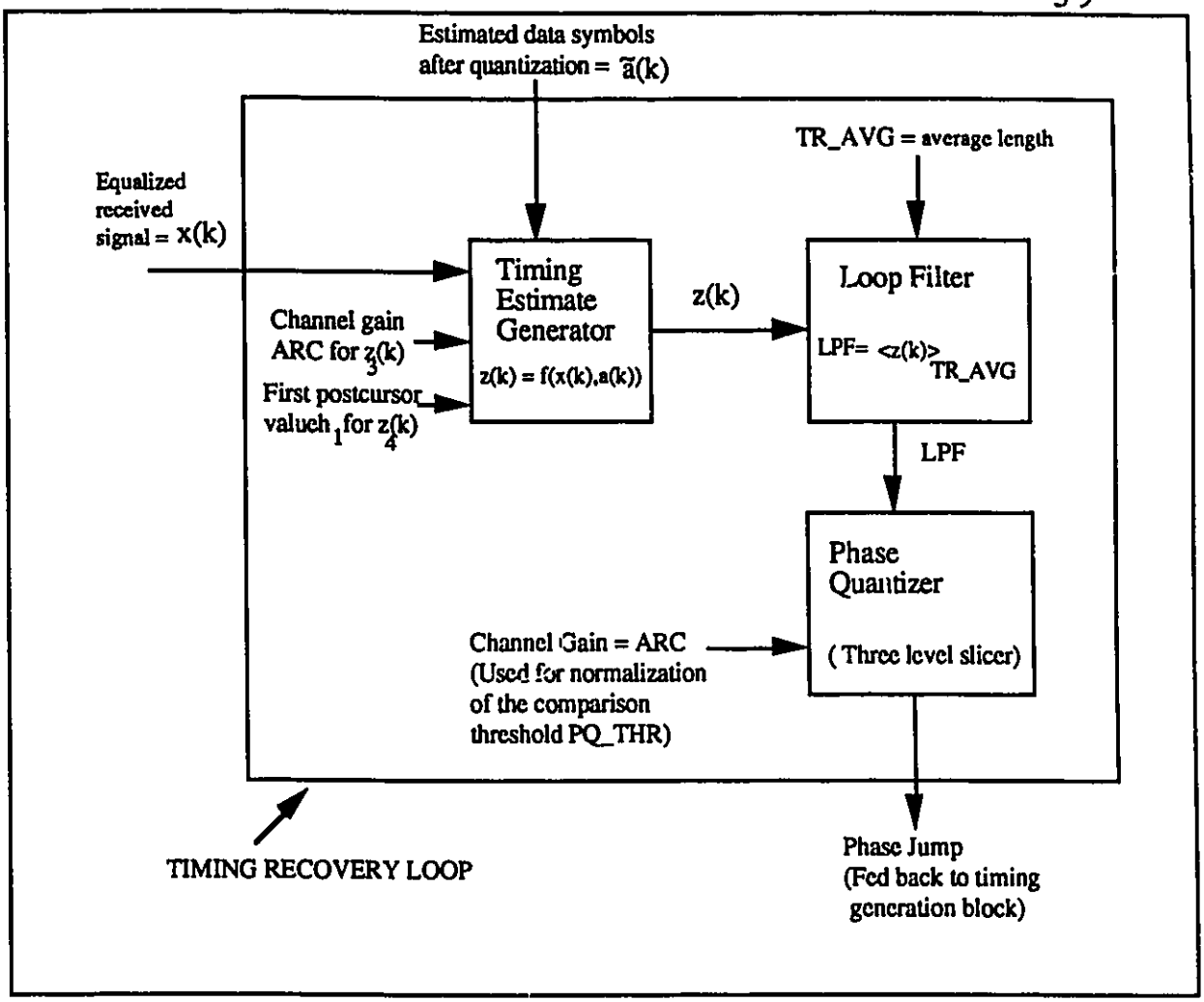

Figure 5.4: Timing Recovery Loop

estimate $\{z_{ik}\}_{i=1,2,3,4}$ exactly as given in Table 5.1, calculated at every symbol. The loop filter performs a time average over a fixed length of bauds, TR_AVG. We will discuss how the choice of the average length affects the performance of the timing estimates in the following section. The average length also limits the frequency of the adjustments of the timing phase. The timing information is updated once at the beginning of every TR_AVG. This is a first order loop where all system variables and states are reset every TR_AVG. The phase quantizer block compares the output of the loop filter to a threshold

(dead-zone) and depending on the outcome, a NO CHANGE, ADVANCE or RETARD clock flag is issued to the timing generation block. The functionality of the phase quantizer is shown in Figure 5.5. The threshold in the phase quantizer is normalized by the channel gain (output of ARC) as follows:

$$PQ\_THR = DZT \times ARC(\tau_K) \tag{5.1}$$

The DZT (Dead-Zone Threshold) is a fixed factor, pre-selected by the designer, and $ARC(\tau_K)$ is an estimate of the main pulse h_0 at the input sampling phase τ.

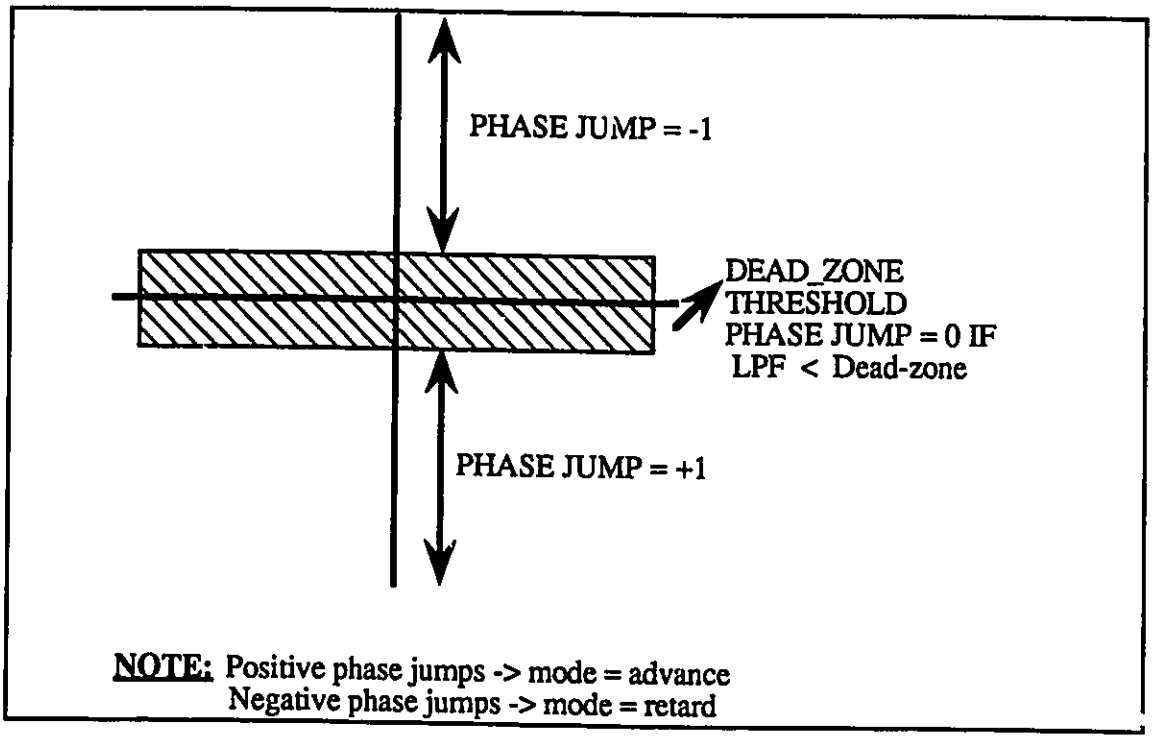

Figure 5.5: Functionality Of A Three Level Phase Quantizer

Normalization by the channel gain is necessary since the DZT value is not variable and must be applicable to a wide range of loops. The choice of the DZT can be crucial to the performance of the timing recovery loop. This will be discussed in more detail in the following sections, but we must note that the optimum value for the DZT should result in

minimum steady-state timing phase jitter while allowing complete convergence of the timing recovery loop, with insignificant steady-state timing phase offset. The output of the phase quantizer defines the direction of the phase jump only; the size of the phase adjustment is fixed. This is a limiting factor for various reasons. The convergence behavior of the loop is greatly affected because the timing phase correction is fixed for all phase offsets. Irrespective of how far we are from our timing reference, we can only advance one step at a time. This limitation is eliminated if we use a multi-level phase quantizer where the size of the phase correction is proportional to the offset between the current and optimum timing phase. The output of the phase quantizer is fed back to the timing generation where the input sampling phase is adjusted accordingly, as shown in Figure 5.6.

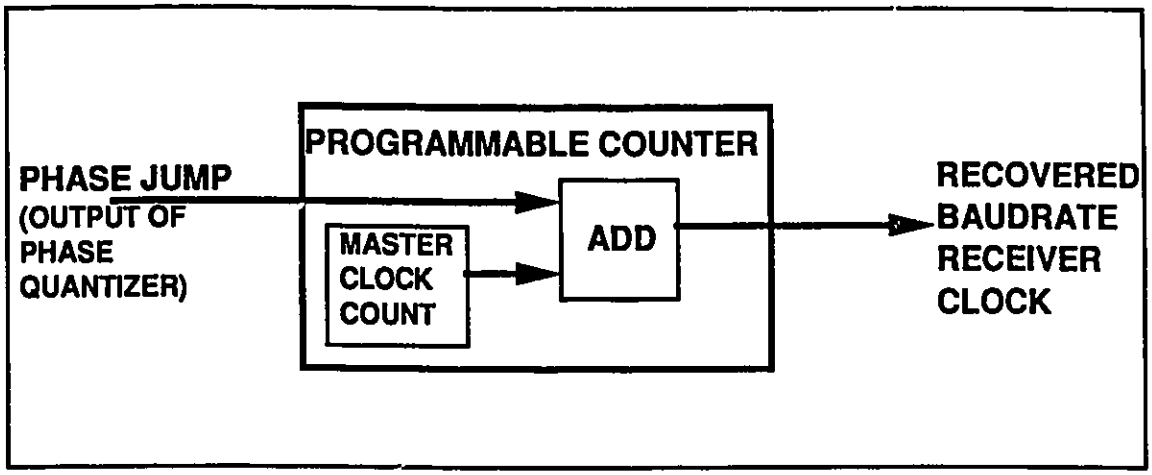

Figure 5.6: Functionality of The Timing Generation block

The loop as described herein cannot result in steady-state free of timing jitter because of the fixed size of the phase adjustments. This means that the optimum steady-state timing phase cannot be exactly reached but that the input sampling phase will hover around it, such that the average of the steady-state phase equals the optimum phase τ_0, i.e. the average of phase corrections is zero.

In this section, we described the timing recovery loop, highlighting the important system parameters which might affect the performance of the timing blocks. These and other factors found to degrade the system's performance will be investigated in the following sections.

5.4.Effect of The Finite Average Length

In the previous section, we stated that the choice of the average length TR_AVG can have a significant effect on the performance of the timing estimates. In our practical system, we are using the time average of z_k to estimate its expected value (statistical average). For TR_AVG$\rightarrow\infty$, these two averages are equal since z_k is wide sense stationary random sequence. This follows from the assumption of uncorrelated data symbols [TzHM86]. The longer we allow TR_AVG to be, the closer the output of the phase quantizer approaches the real expected value of z_k. Since $E\{ z_k \} = f(\tau)$ is the timing information we seek, we need to minimize e_{TR_AVG}, the error in the approximation of the ensemble average of z_k by time averaging as given in (5.2).

$$MSE_{TR\_AVG} = < e_{TR\_AVG}^2 > = (< z_k >_{TR\_AVG} - f(\tau))^2 \qquad (5.2)$$

In contrast, a long average is unpractical since the phase corrections are performed once every TR_AVG so that the convergence time of the timing recovery loop is directly proportional to the TR_AVG length. In all our simulations, TR_AVG = 1 frame = 120 bauds is set to correspond to the existing system. However, it is important to understand the dependence of the timing estimates on the averaging length, TR_AVG. To isolate the effect of the average length only, the channel pulse response was truncated to eliminate all postcursor ISI, and the true transmitted symbols were used in generating z_k. The system used to obtain these results is shown in Figure 5.7 and was implemented using the CAPSIM/BLOSIM DSP Simulation Package which is described in Appendix C. The mean

of z_k is calculated for various phase offsets and directly compared to the $E\{ z_k \} = f(\tau)$ $= h_{-1}$. Rectangular pulses (100% duty cycle) were used with an upsample rate of 128 such that the size of the phase jump is fixed at $\frac{1}{128} \times$ symbol period. In Figure 5.8 - (a) to (d), the output of the phase quantizer for estimates #1 to 4 respectively, is plotted for the TR_AVG values of 0.5, 1, and 10 frames, for the B4 loop. The B3 and SL loops yield similar results.

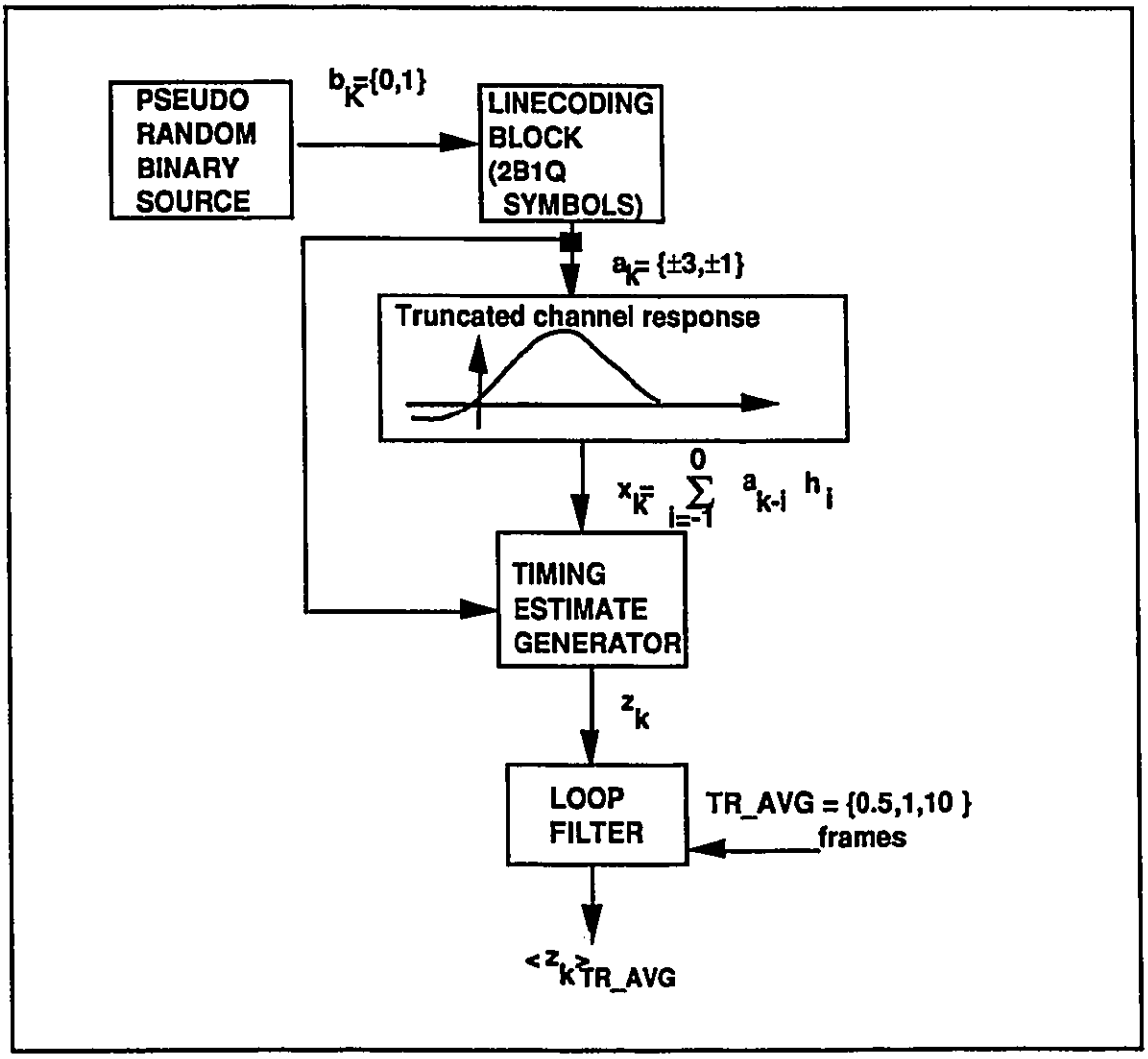

Figure 5.7: Simplified Receiver Configuration for The Study of the Effect of the Finite Average Length

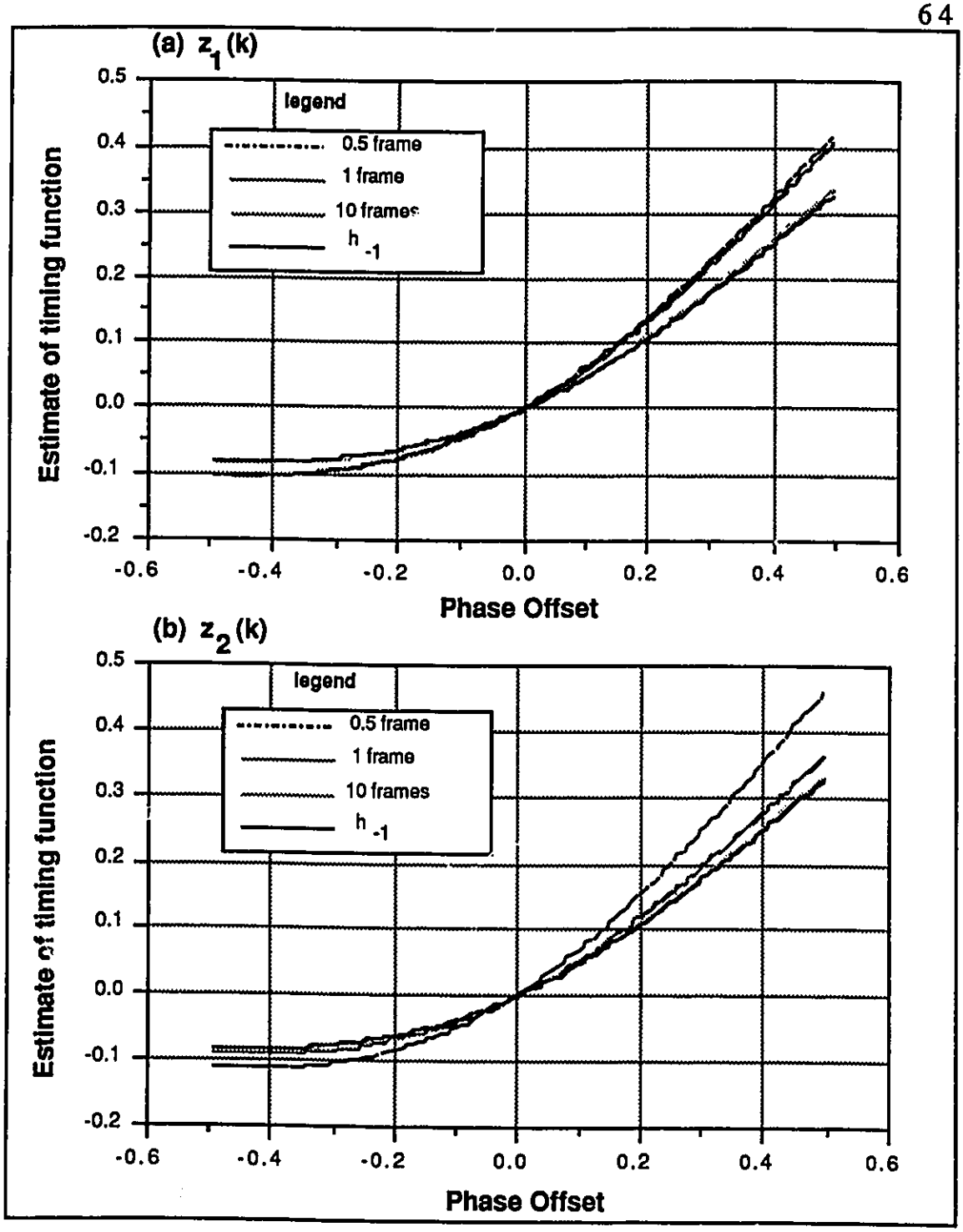

Figure 5.8: Comparison of the Effect of TR_AVG on the Four Estimates, B4 loop. (a) Estimate #1 (b) Estimate #2

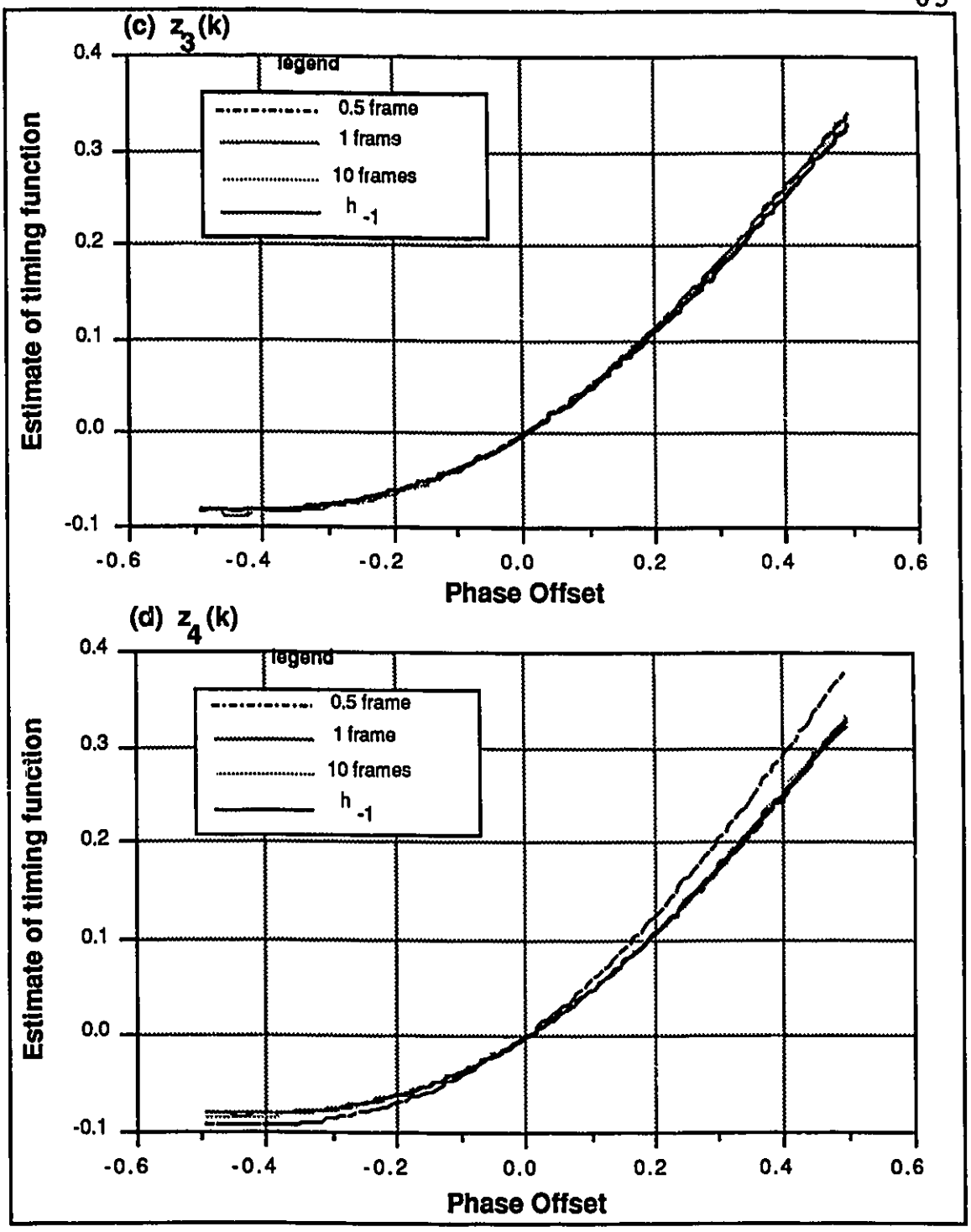

Figure 5.8: Comparison of the Effect of TR_AVG on the Four Estimates, B4 loop. (c) Estimate #3 (d) Estimate #4

From the plots, it is evident that the mean of z_k approaches $E\{z_k\} = f(\tau)$ as TR_AVG gets longer, for all four estimates. It is interesting to note that as we increase TR_AVG, the degree of improvement in each estimate varies. The MSE_{TR_AVG} given in (5.2) is plotted for $\{z_{ik}\}_{i=1,2,3,4}$ in Figure 5.9(a-c) respectively. We show the B3, B4 and SL loops for different phase offsets. In all three cases, z_{4k} results in the smallest error, followed by z_{3k} and z_{2k}, with the worst being z_{1k}. For estimate #4, h_1 was set to zero. e_{TR_AVG} decreases as τ approaches τ_0. This is obvious since at τ_0, $f(\tau) = 0$. Also, at the region close to τ_0, the error increases symmetrically from either side, and this is explained by the linear behavior of the timing function in the vicinity of τ_0. As we get further from τ_0, the symmetry is lost because the slope at either side will be different, with a steeper slope on the main pulse side, giving a higher error at these timing phases. This behavior is not crucial to the performance of the timing loop since we are mostly interested in the linear region close to τ_0.

These results do not predict which estimate will perform better since there exist many other system parameters which will have a stronger effect. However, for the ideal case described in Figure 5.7, our result will directly apply. Depending on the level of the interferences of the various system error sources, the performance of the each timing estimate will be as indicated in Figure 5.8 only if e_{TR_AVG} is the major influencing factor. In a practical system, the error due to TR_AVG will not be a significant factor, and even though all other error sources are kept minimal, there will be not be a difference in the performance of the four estimates. This is verified in the next example. Figure 5.10 shows a simplified version of the overall receiver, which is basically the closed loop form of the system in Figure 5.7. We added an Adaptive Threshold Quantizer (ATQ) and an ARC block. The ATQ takes symbol rate samples of the received signal x_k and estimates the corresponding 2B1Q symbol. The ARC calculates the channel gain which is used for normalization by the ATQ and phase quantizer block. The system parameters are as follows:

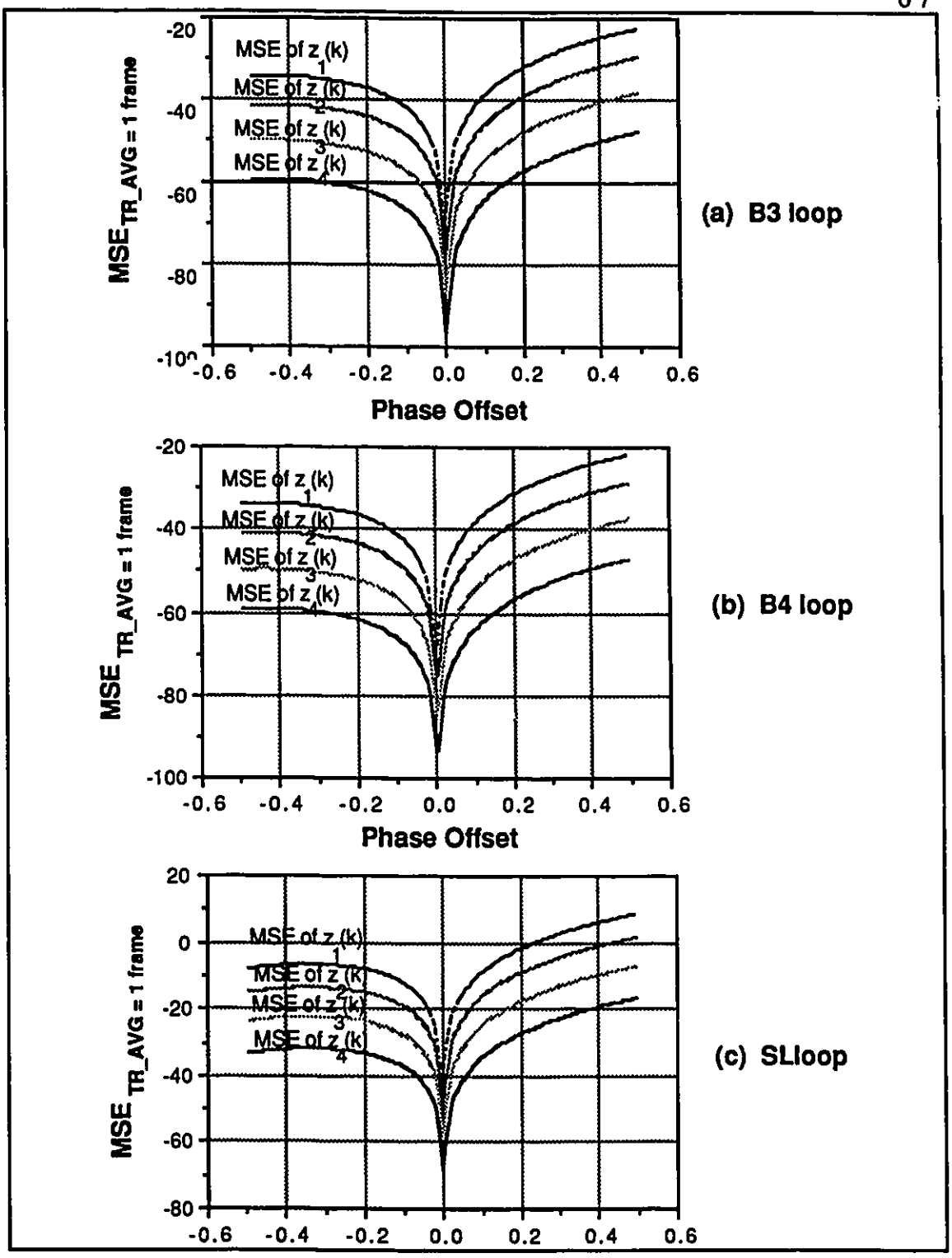

Figure 5.9: Comparison of the Mean Square Error for the Four Estimates. (a) B3 loop (b) B4 loop (c) SL loop

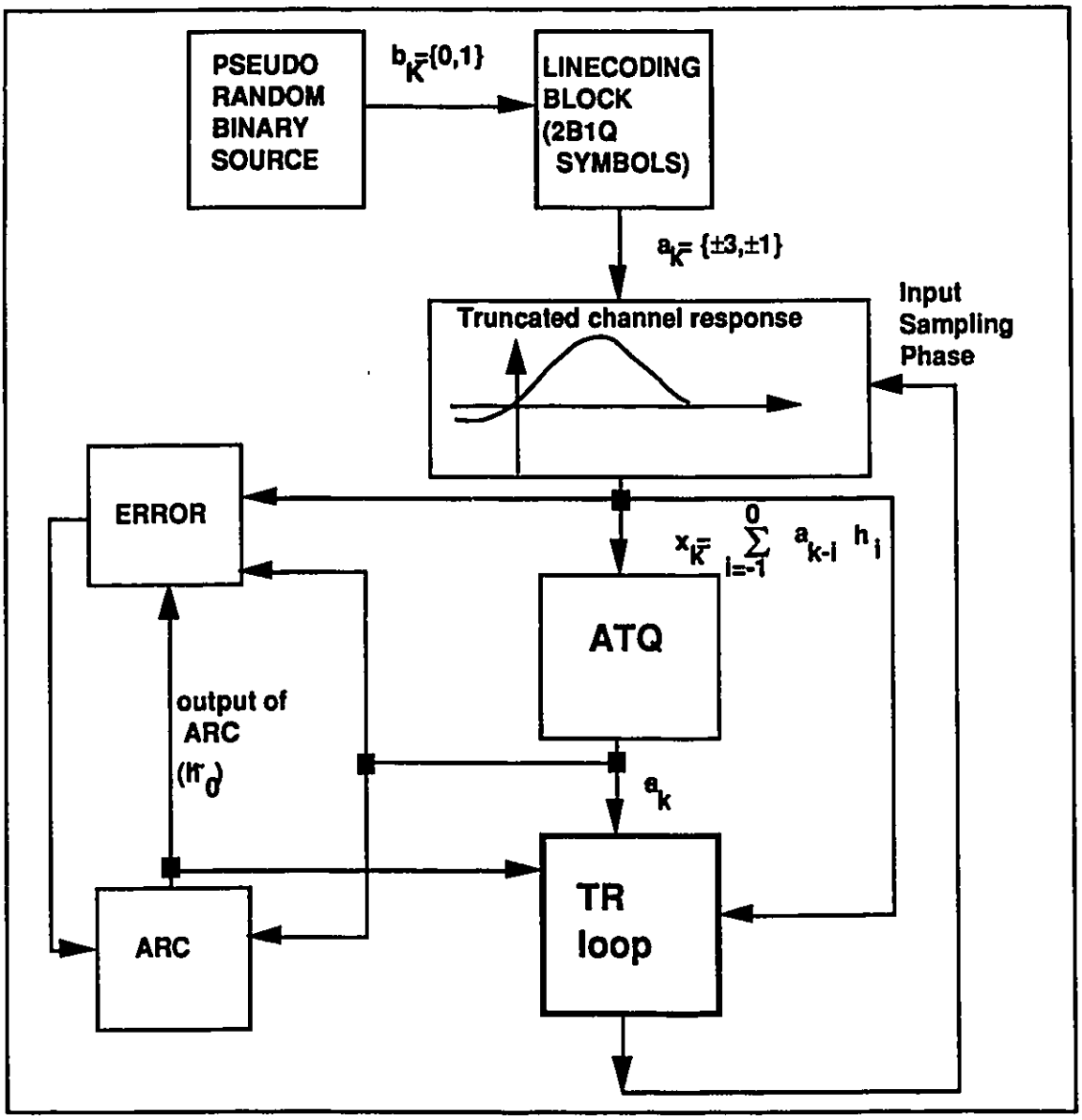

Figure 5.10: Closed Loop Form of Simplified Receiver

° TR_AVG = 1 frame = 120 bauds

° phase jump = $\frac{1}{128}\times$ symbol period ≅ 98 nanoseconds

(symbol rate = 80 Kbauds/second)

° DZT = 0 (simple comparator)

° Loop: B3 (truncated)

° initial phase offset = -0.25

° simulation length = 100 frames

The loop filter output , $< z_k >_{TR\_AVG}$ is shown in Figure 5.11 (a) against the timing reference $f(\tau) = h_{-1}$ for all four estimates, with the corresponding MSE_{TR_AVG} similarly plotted in Figure 5.11(b). The error in $\{z_{ik}\}_{i=1,2,3,4}$ in estimating $f(\tau)$ is equal. In Figure 5.11 (c), the timing phase as it converges to τ_0 is plotted and its behavior is identical for all z_k 's. At steady-state, the sampling phase fluctuates continuously and uniformly around the optimum sampling point due to the finite nature of the phase adjustments. This represents the ideal case resulting in minimum steady-state timing phase jitter, which could be eliminated by increasing the DZT. In practice, variable channel noise, uncanceled ISI and other system disturbances will cause a higher level of steady-state timing phase jitter which will be more difficult to control by the DZT. This parameter is fixed and cannot be chosen to satisfy all possible situations.

Although our results indicate that the error due to finite averaging will not be significant in our practical loop and will not affect the performance of the timing estimates to any appreciable extent, it should be nonetheless considered. If the timing updates are performed at a much higher rate, every one or few bauds, with a first order timing loop, the error due to finite averaging might be significant. In such a case, the results in this section should be considered in choosing among the four timing estimates.

In this section, we have investigated the effect of finite averaging on the performance of the timing estimates and concluded that in a practical loop, the error is not significant. We have compared e_{TR_AVG} for $\{z_{ik}\}_{i=1,2,3,4}$ for the ideal case of a noiseless, ISI free channel. In the next section, we will follow our characterization of the timing estimates by discussing the effect of channel distortion and noise on each of them.

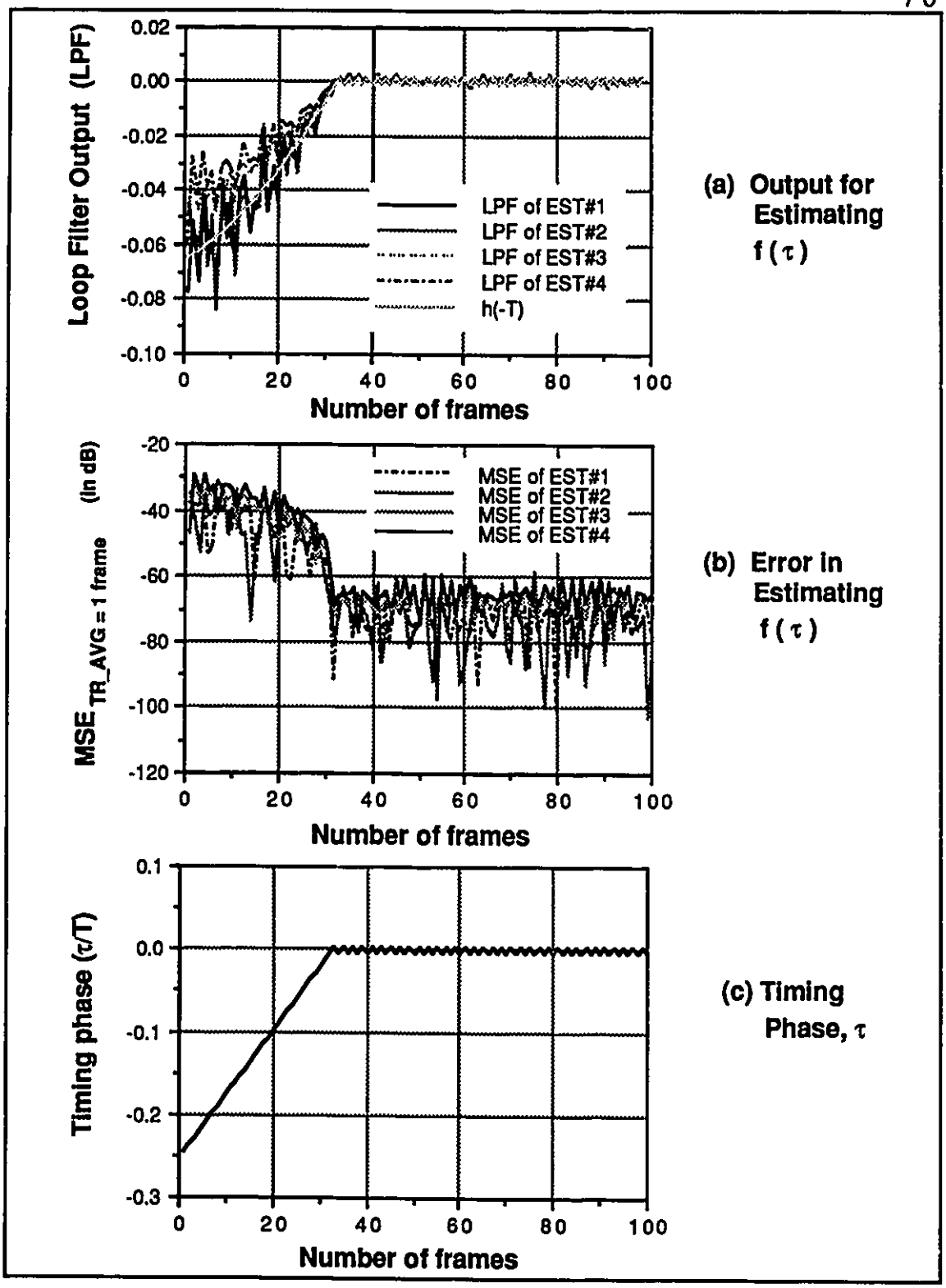

Figure 5.11: Performance of Estimates for Closed Loop Form of Simplified Receiver for the B3 Loop

5.5.Effect of Channel Distortion And Additive Noise

The effect of additive white gaussian noise and ISI on the performance of the timing estimate is quantified in its variance. The expressions for the variance of $\{z_{ik}\}_{i=1,2,3,4}$ specified in Table 5.2 are functions of the mean square distortion of the channel and of the noise power σ^2. Since $E\{ z_k \} = f(\tau)$ is the information we seek, then the variance of z_k, $Var\{ z_k \} = E\{ z_k^2 \} - E^2\{ z_k \}$, by definition, will give us a measure of the error in z_k in estimating $f(\tau)$. Consequently, the timing estimate with the lowest variance is the better one.

If we assume a noiseless, distortion free channel, a perfect zero-variance estimate is obtained in all four cases. For the more realistic case where both noise and ISI are present, the best timing estimate will depend on the level of degradation due to each one of the two error sources. From Table 5.2, we could see that the relative effect of ISI and additive noise is similar for all z_k's. Table 5.3 compares the variance expressions of the four estimates. Estimate #3 is the best, resulting in an MSE that is 9 dB lower than that of estimate #2 which is the worst case, and 5 dB and 3 dB lower than estimates #1 and 4 respectively. Estimate #4 is second best with an error that is 2 dB lower than estimate #1. These results represent the margin for noise and channel distortion of each z_k. In our comparison, we ignore the h_{-1} and uncancelled h_1 terms since both values are zero at steady-state. In $VAR\{z_{ik}\}_{i=1,2,3,4}$ the noise power term is ~ 7 dB lower than the MSD value. This becomes significant at steady-state since after convergence of the equalization blocks, most of the channel distortion effect will be cancelled such that the additive channel noise term will be the dominant influence. This additional factor in the MSD value in the variance represents the power in the 2B1Q data symbols given by the expression $E\{a_k^2\}$.

$10\log_{10}\left\{\frac{Var(z_i)}{Var(z_j)}\right\}$	j			
i	1	2	3	4
1	0 dB	-3.7 dB	5 dB	2 dB
2	3.7 dB	0 dB	9 dB	5.6 dB
3	-5 dB	-9 dB	0 dB	-3 dB
4	-2 dB	-5.6 dB	3 dB	0 dB

Table 5.3: Comparison Of The Effect Of Additive Noise And Channel Distortion On The Four Timing Estimates

The variance expressions are plotted in Figure 5.12 for the B3 loop against various phase offsets. The four estimates are compared for a noiseless channel with some residual ISI (Figure 5.12a), and for an SNR=14 dB (Figure 5.12b). The smooth curve represents the theoretical expressions in Table 5.2, whereas the noisy curve is a direct calculation of the variance of z_k for the example shown in Figure 5.7. For the latter, we simply calculate the variance $(E\{z_k^2\}-E\{z_k\}^2)$ of the output of the timing estimate generation block for TR_AVG = 1 frame. This verifies our theoretical expressions and reinforces the above comparison of the four timing estimates.

Up to now, we have examined the effect of channel distortion and additive noise by comparing the error resulting from each effect for the four timing estimates. Our next step is to relate these quantities to the performance of each z_k in the timing recovery loop. To this end, we will give in the next section a simplified model for the timing recovery loop in terms of a linearized Phase Locked Loop, and derive an expression for the steady-state timing jitter associated with our system for each timing estimate.

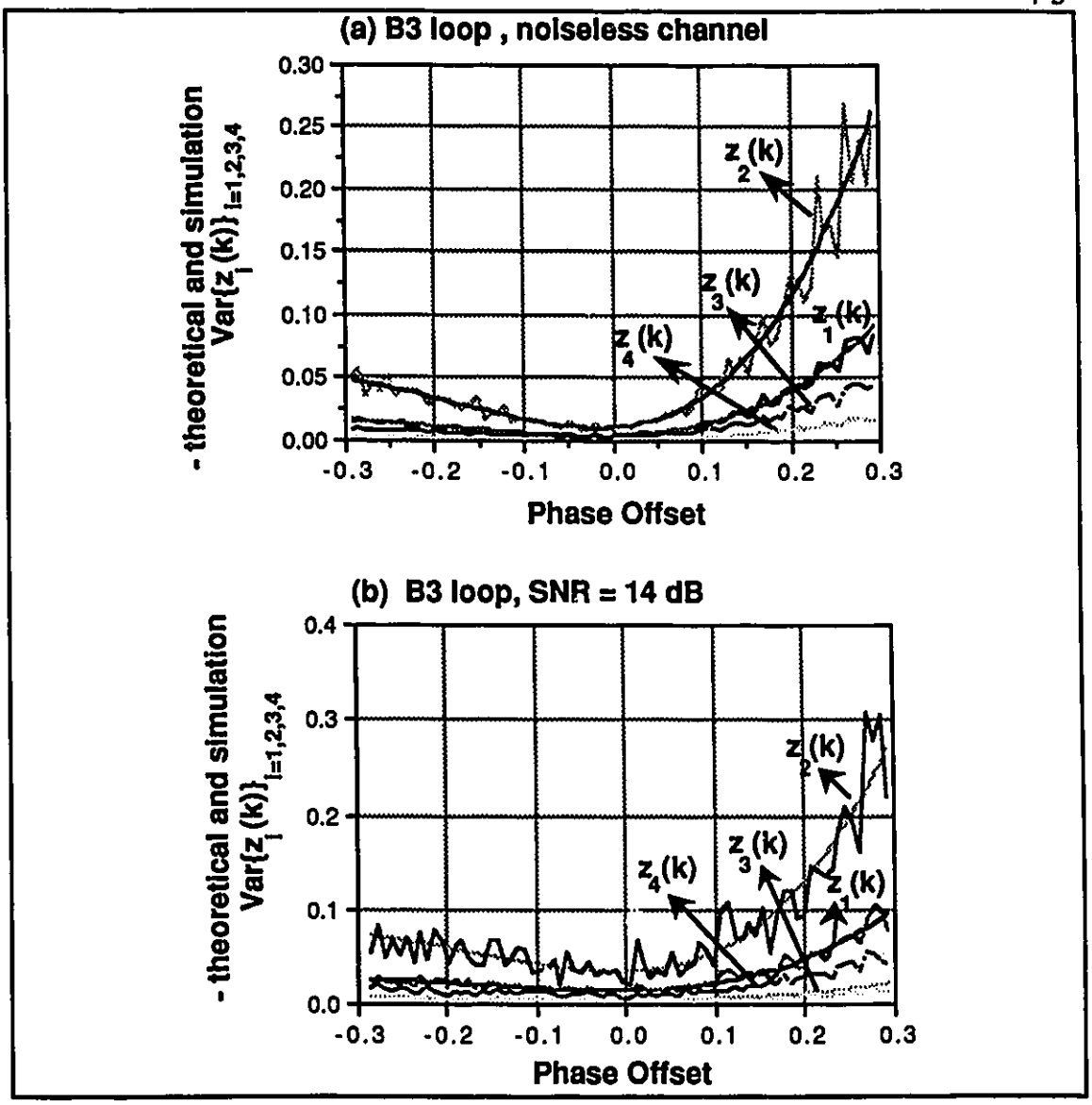

Figure 5.12: Plots for the Variance Expressions of the Timing Estimates for the Effect of Channel Distortion and Additive Noise

5.6.Linearized Model For PLL

Our timing recovery loop can be simplified to approximate the linearized dynamics of a PLL as shown in Figure 5.13. During steady-state, the phase error in the timing recovery loop is small enough to assume that the phase detector is linear [TzHM86]. To map the timing estimate generation block to a linearized phase detector, we must determine

the relationship between the input phase $\theta_{ik} = \frac{\tau}{T}$ and the incoming signal. During steady-state, z_k represents the jitter in the signal and to relate it to the jitter present in the phase of the signal, we simply divide by the slope of our timing function [TzHM86], such that:

$$\theta_{ik} = \frac{z_k}{sT} \tag{5.3}$$

where s is the slope of $f(\tau) = h_{-1}$. This transformation allows us to treat the timing recovery loop as a linear system (Figure 5.13(b)) which can be easily analyzed.

The linearized phase detector compares the phases of the incoming signal to that of the clock at each sampling instant, such that its output consists of the error in the input phase as follows:

$$\varepsilon_k = \theta_{ik} - \theta_{ok} \tag{5.4}$$

The loop filter smooths the error signal from the phase detector to generate a correction signal c_k which will be used by the timing generation block for the adjustment of the output phase as described by the difference equation:

$$\theta_{ok+1} - \theta_{ok} = c_k \tag{5.5}$$

The digital clock is similar in its function to that of a Voltage Controlled Oscillator (VCO) in the analog PLL. In a digital VCO, the ratio of the frequency divider is controlled as opposed to controlling the frequency in the analog VCO. The correction signal is used to update the clock's time period T_c so as to decrease the phase error.

$$T_{ck} = T_o - c_{k-1} \tag{5.6}$$

where T_o corresponds to the basic clock period when we are dividing by N.

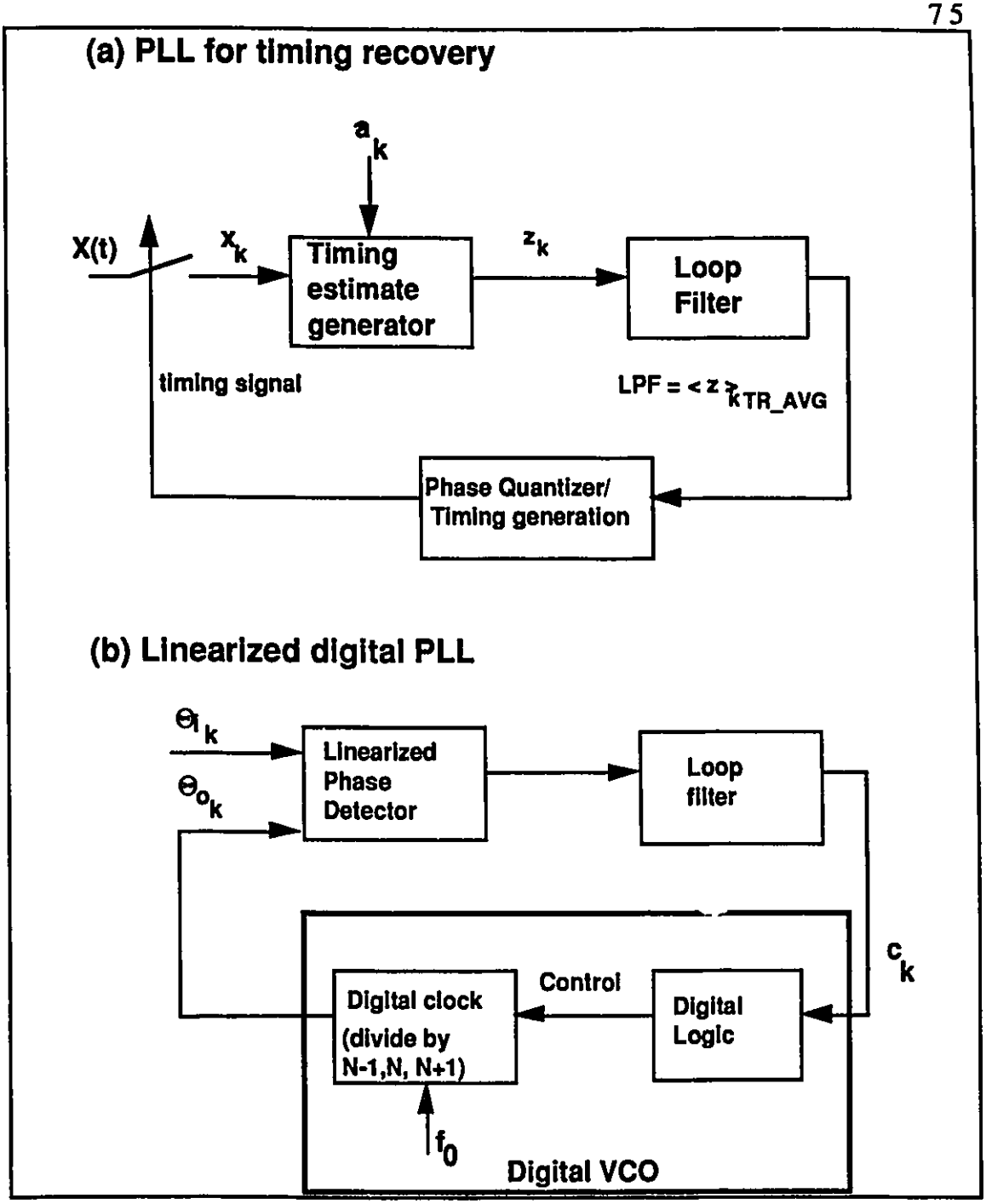

Figure 5.13: Linearized Model for Timing Recovery PLL

In our case, we must consider an additional digital logic block to include the phase quantizer effect. Basically, the error signal c_k is compared against a threshold PQ_THR as given by (5.1), and a control flag ± 1 is output based on the following test:

c_k < - PQ_THR — {Sample later; T_c increased; Divide by N+1

c_k > PQ_THR — {Sample earlier; T_c decreased; Divide by N-1

During steady-state, after convergence to the optimal phase, the error signal c_k fluctuates around ± PQ_THR such that the divide ratio of the VCO is alternating between N and N+1 or N and N-1. It is easy to see that for our case, the lock range of the PLL is determined by the nominal divide ratio, N. The maximum and minimum input frequency which can be held by the PLL is given when the VCO is dividing by N-1 and N+1 respectively. In our system, f_0 = 10.24 MHz and N = 128. Having described the function of each block in the linearized PLL, we will now derive an expression for the transfer function of this system. For convenience we will work in the Z-domain [LeeM88].

Given the input-output relationship of a linear system, the loop filter output can be expressed as:

$$C(Z) = E(Z) \times L(Z) \tag{5.7}$$

where $L(Z)$ is the discrete transfer function of the loop filter and $E(Z)$ is the Z-transform of the error signal ε_k as given in equation (5.4), such that :

$$E(Z) = \theta_i(Z) - \theta_o(Z) \tag{5.8}$$

Taking the Z-transform of the output phase in equation (5.5), we obtain:

$$Z\theta_o(Z) - \theta_o(Z) = C(Z)$$

$$\theta_o(Z) = \frac{C(Z)}{Z-1} \tag{5.9}$$

Substituting equations (5.7) and (5.8) in (5.9), we get an expression for the overall transfer function :

$$H(Z) = \frac{\theta_o(Z)}{\theta_i(Z)} = \frac{L(Z)}{L(Z) + Z - 1} \tag{5.10}$$

For a first order Phase Locked Loop, the loop filter has a constant transfer function of the form $L(Z) = G_{LPF}$, where G_{LPF} is some gain factor. Then, $H(Z)$ can be simplified as given:

$$H(Z) = \frac{G_{LPF}}{Z - (1 - G_{LPF})} \tag{5.11}$$

The above expression shows a real pole at $Z = 1 - G_{LPF}$. In order for the system to be stable, the pole must be inside the unit circle, therefore, if $-1 < Z < +1$, the loop filter's gain is limited to $0 < G_{LPF} < 2$. For our system, the loop filter can be approximated according to the above as shown in Figure 5.14. Assuming a small and slow varying steady-state error, the output of the loop filter becomes a scaled ("smoothed") version of the filter's input. The transfer function $L(Z)$ can thus be approximated to a constant gain G_{LPF} as defined by TR_AVG. For TR_AVG = 1 symbol, G_{LPF} is the unity gain. For TR_AVG>1, $G_{LPF}<1$ and the phase transfer function becomes a simple low-pass filter. This is true in our case since averaging is performed over several bauds as in our previous example where TR_AVG = 1 frame = 120 bauds.

In brief, with the assumption that the PLL has converged such that the average of the sampling phase output by the VCO corresponds to the desired timing phase, and the steady-state timing jitter is very small, it is possible to represent the timing loop as a linear system with low-pass characteristics. In the next section, we will derive the power spectrum of the input and output jitter and compare these results for the four timing estimates.

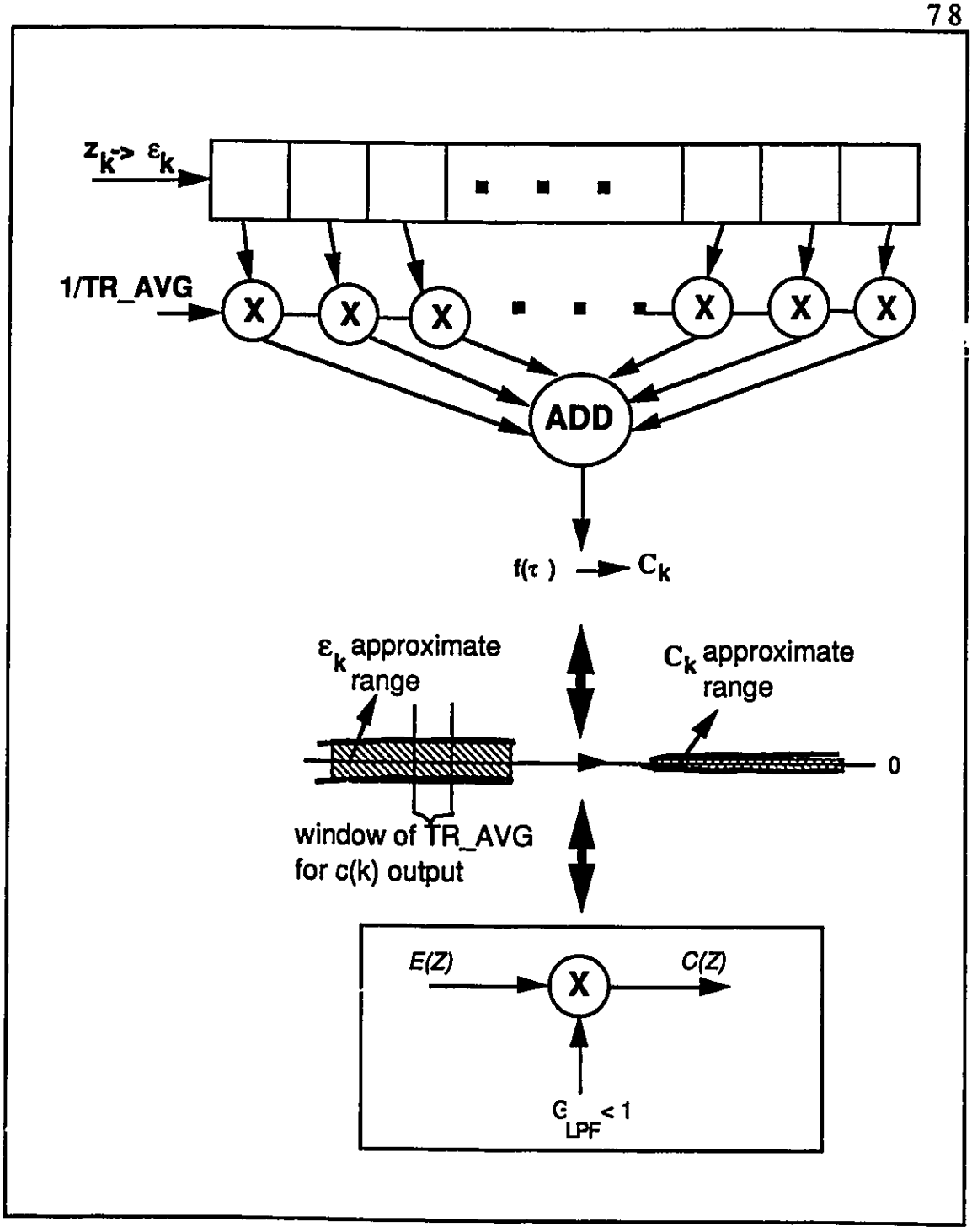

Figure 5.14: Approximation of the Loop filter for the Linearized PLL Model

5.7.Timing Jitter Spectrum

The input timing jitter as given in (5.3) is a wide-sense stationary random sequence [TzHM86]. Its autocorrelation function function can be expressed as:

$$R_{\theta_i}(m) = E[\theta_{ik+m}\ \theta_{ik}] \tag{5.12}$$

The input timing jitter and the corresponding autocorrelation function were derived for estimates #1 to #4 and are given in Table 5.3. R_{θ_i} (m) was determined directly from its definition in (5.12). First, we replace the received signal samples x_k and x_{k-1} by its equivalent expressions $x_k = \sum_{i=-\infty}^{\infty} a_{k-i}\, h_i$ and $x_{k-1} = \sum_{i=-\infty}^{\infty} a_{k-1-i}\, h_i$, we simplify the h_0 term which cancels out and eliminate the h_{-1} term which is zero after convergence of the timing recovery loop. Assuming uncorrelated data symbols such that $E\{a_k a_{k-i}\}=E\{a_k\}E\{a_{k-i}\}$, $(i\neq 0)$, (5.12) can be easily determined for each of the four timing estimates.

θ_i	$R_{\theta_i}(m)$	
i	$m=0$	$m\neq 0$
1	$\frac{1}{(sT)^2}\{3.125 \sum_{(i\neq 1)}' h_i^2 + 4.125 h_1^2\}$	0
2	$\frac{1}{(sT)^2}\{7.25 \sum_{(i\neq 1)}' h_i^2 + 9.25\, h_1^2\}$	0
3	$\frac{1}{(sT)^2} \sum_{}' h_i^2$	0
4	$\frac{1}{(sT)^2}\{2 \sum_{(i\neq 1)}' h_i^2 + 1.64\, h_1^2\}$	0

Table 5.4: Autocorrelation Function for Input Timing Jitter

The power spectrum can be directly obtained from $R_{\theta_i}(m)$,

$$S_{\theta_i}(e^{j\omega T}) = \sum_{m=-\infty}^{\infty} R_{\theta_i}(m)\, e^{-j\omega m T} \tag{5.13}$$

For our case, $R_{\theta_i}(m)$ is defined at m=0 only, and is zero elsewhere, resulting in a constant power spectrum as shown in Figure 5.15. $R_{\theta_i}(0)$ can be interpreted as the power in the input timing jitter which is a white random process. For the linearized loop, the output timing jitter can be directly related to $R_{\theta_i}(m)$ as:

$$S_{\theta_o}(e^{j\omega T}) = S_{\theta_i}(e^{j\omega T}) \left|H(e^{j\omega T})\right|^2 \tag{5.14}$$

where $H(e^{j\omega T})$ is the effective transfer function $H(Z)$ of the PLL described in the previous section, at $z = e^{j\omega T}$. Given that $H(e^{j\omega T})$ is a simple low-pass function and that the same $H(e^{j\omega T})$ multiplies all of the timing estimates, the lowest output timing jitter will result from the estimate with lowest input timing jitter. Then, from Table 5.4, we can directly compare the steady-state timing jitter resulting from each timing estimate as applied to the timing recovery loop . To add white gaussian noise, we simply replace the received signal in z_k by $x_k = \sum_{i=-\infty}^{\infty} a_{k-i}\, h_i + N_k$. The noise factors will be similar to those found in the variance expression.

The output timing jitter derived here represents the case where the dead-zone threshold PQ_THR=0, i.e. worst case. By choosing PQ_THR appropriately, the steady-state timing jitter can be significantly reduced. However, the relative performance between $\{z_{ik}\}_{i=1,2,3,4}$ will remain the same. In the next section, we will briefly elaborate on the effect of the dead-zone threshold (DZT) parameter .

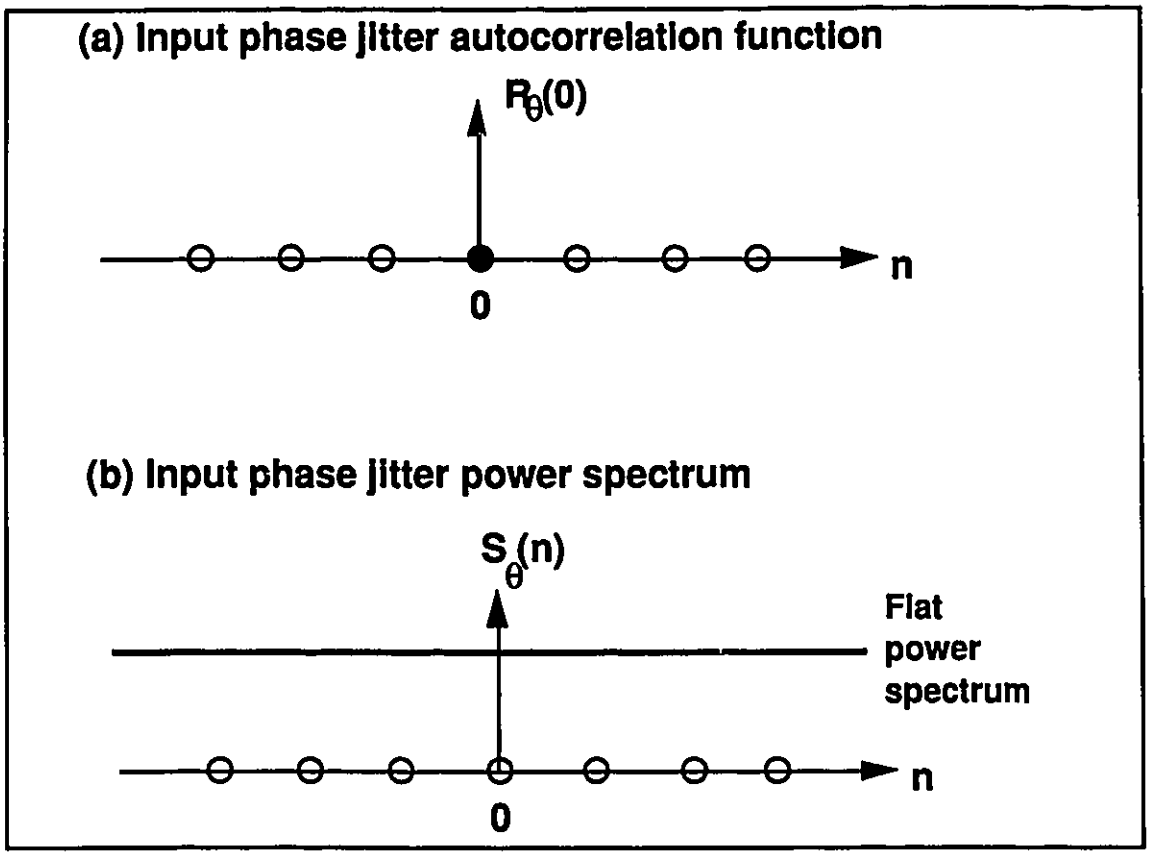

Figure 5.15: Input Phase Jitter Characteristics (a) Autocorrelation Function (b) Power Spectrum

5.8.Effect of The Dead-Zone Threshold

The dead-zone threshold in the phase quantizer allows a reduction of the timing jitter by allowing a certain amount of phase jitter to be undetected. PQ_THR should be chosen large enough to eliminate jitter due to the randomness of the signal without causing a significant steady-state phase offset. This is achieved if the following equation is satisfied:

$$<\tau_{ss}> = \tau_0 = 0 \qquad (5.15)$$

Where τ_{ss} is the timing phase reached after convergence of the timing recovery loop. In choosing a DZT, it is more practical to follow a somewhat empirical approach where a

compromise value is obtained through observation of a sufficiently large and varied sample of possible cases. The exact effect of the DZT is difficult to quantify because of the varying system parameters, but especially because of the feedback nature of the loop coupled with the strong interaction between the DSP blocks.

The allowable peak-to-peak phase jitter for a certain DZT can be directly determined from the loop's functionality:

$$\theta_{opp} \approx \frac{2 \times \text{DZT } h_0}{s\,T \times \text{TR\_AVG}} \tag{5.15}$$

Table 5.5 gives approximate values for the allowable timing jitter for loops B3 , B4 and SL for different DZT values that will be used in the simulation in the next chapter.

DZT	B3 LOOP	B4 LOOP	SHORT LOOP
0	0	0	0
1	0.0129	0.0124	0.0133
2	0.0258	0.0248	0.0266
3	0.0386	0.0372	0.0398
4	0.0520	0.0459	0.0531
5	0.0644	0.0619	0.0664
6	0.0773	0.0743	0.0797
7	0.0900	0.0867	0.0930

Table 5.5: Allowable Peak-To-Peak Phase Jitter ($\frac{\tau}{T}$) At Steady-State

5.9.Summary

In this chapter, we analyzed the timing estimates analytically looking at the variance expressions and the effect of the finite average length, channel distortion and additive noise. We also developed a linear model of the PLL timing recovery block to derive and compare the input and output timing jitter spectrum for each of the timing estimates. It was found that timing estimate #3 behaved the best overall resulting in the lowest output timing jitter power spectrum as well as the least sensitivity to the various noise sources enumerated above. Estimate #4 was second best followed closely by estimate #1 with finally estimate #2 yielding the highest output timing jitter. In the next chapter, we will verify these theoretical results through simulation.

CHAPTER 6

Simulation Results

6.1.Introduction

In this chapter, we verify the theoretical analysis and discussion presented in the previous chapter through simulation. A version of the U-transceiver described in chapter 3 is simulated for half-duplex mode. The setup is used to perform the characterization and comparison of the performance of the timing estimates for the various system parameters discussed in chapter 5. The simulation setup is first presented, then a detailed investigation of the behavior of the timing estimates is given for the specified parameters. The theoretical and simulation results are shown to be very close.

6.2.Description of Simulation Setup

The architecture for the half-duplex receiver used in the simulation is shown in Figure 6.1. All simulations were generated and performed in the CAPSIM/BLOSIM environment. A detailed description of the DSP software simulation package is given in Appendix C. This package is generic and provides the hooks needed to generate building

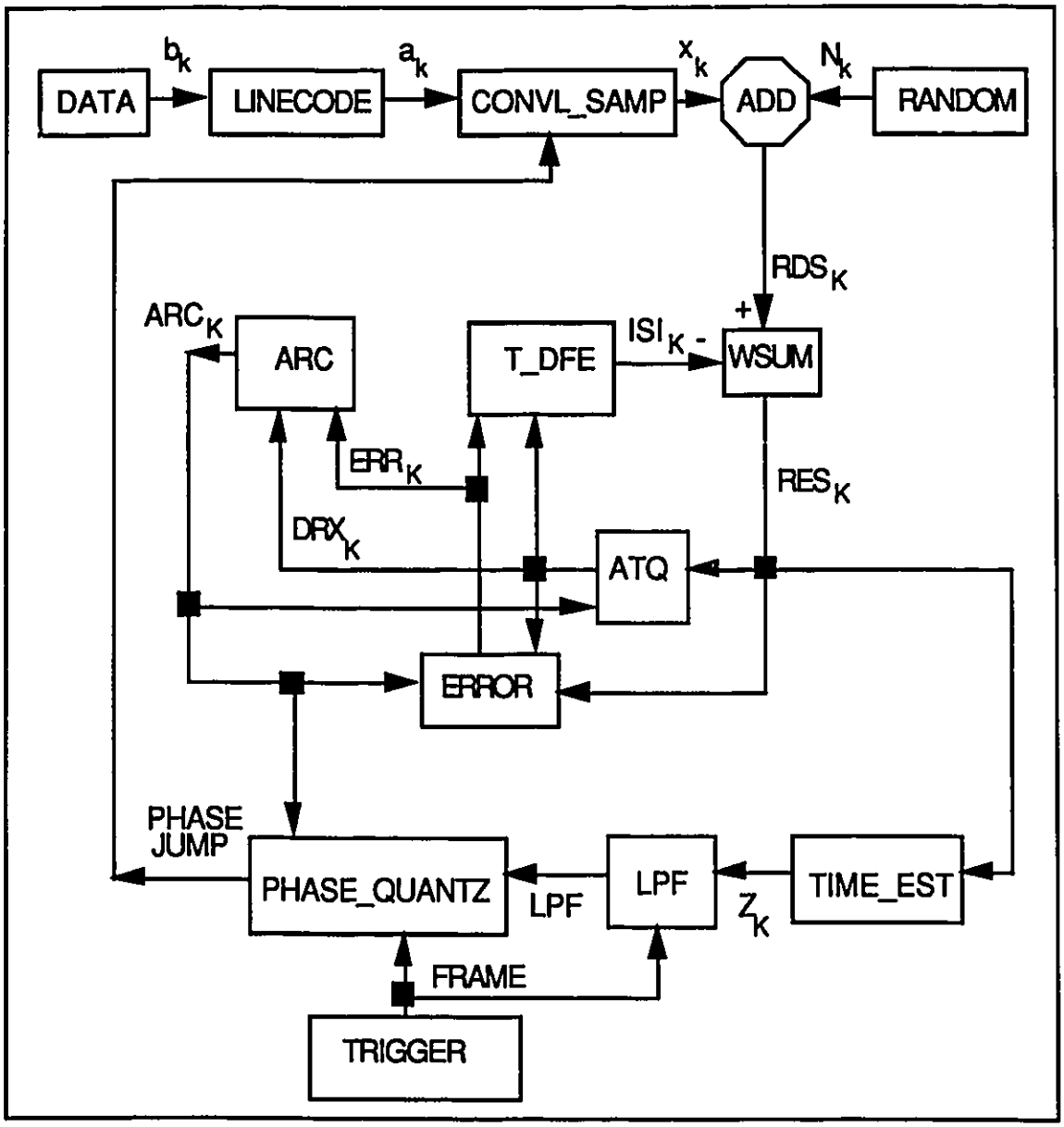

Figure 6.1: Receiver Setup for Simulation

blocks in a hierarchical fashion for the implementation of a complex DSP system. The source code for all the blocks in the simulation of the half-duplex receiver was written as part of the thesis work using the programming language pseudo-C. A description of this language is included in Appendix C and the source code for all the blocks shown in Figure

6.1 are given in [HaYa89]. In CAPSIM, the term topology is used to denote any generic grouping of interconnected blocks. Figure 6.1 shows the topology used for all the simulations for the four timing estimates. We should note that this topology was setup to run at symbol rate, i.e. the DSP functions are executed simultaneously with every block visited once per symbol. This was necessary due to the feedback nature of the loop. In this section, the topology of Figure 6.1 including the functionality of the blocks, their inputs, outputs and parameters will be described. An input to a block is defined as any signal coming from other blocks, needed for processing by that block. An output from a block is defined as any signal resulting from processing performed on inputs within that block and passed out to be used by other blocks. A block's parameter is defined as a constant (floating-point, integer or character) which needs to be varied for different simulation runs. Any parameter for any block within a topology could be adjusted at the beginning of the simulation run. A parameter cannot be changed once the simulation starts executing. A block can have none or many inputs and/or outputs and/or parameters.

Within our topology, we can identify three types of blocks: source blocks, front end blocks and adaptive blocks. A source block is a block that does not require any inputs and is used to provide the data needed to execute a simulation. The length of the simulation is usually set by these blocks. Front end blocks are the blocks that could be executed at a higher rate than the symbol rate, i.e. any block outside the feedback path. Adaptive blocks include all processing blocks within the feedback loop, requiring past values for the computation of current samples. A brief description of the blocks in the receiver topology will now be given based on the above classification.

Source Blocks:

DATA: Generates a pseudo-random sequence using the polynomial $(x^{10} + x^3 + 1)$

Outputs: b_k = binary bits {0,1}

Parameters: Initial shift register value = 12 (*NOTE*: same random sequence is used for all simulations described in this chapter.)

RANDOM: Generates random variables that are normally distributed with mean 0 and variance σ^2. The polar method is used to generate normally distributed samples from a sequence that is uniform on (-1,1). The resulting distribution is described exactly as $N(0,\sigma^2)$. This method is based on the inverse distribution function.

Outputs: N_k = additive white Gaussian noise samples

Parameters: σ, RMS power of the output signal

TRIGGER: General purpose block which could be used to provide a triggering signal to other blocks. It is used here to set the loop filter average length for the timing recovery adaptation. It is connected to the loop filter and phase quantizer blocks. It outputs a trigger signal of value 1 every average length TR_AVG, and a 0 at all other times. On the reception of a set control signal, the loop filter sends its average to the phase quantizer and resets all its variables. The phase quantizer then generates the appropriate phase jump control signal and sends it to the input timing generation block. Both the loop filter and the phase quantizer output zeros at all other times, thus disabling any phase adjustments.

Outputs: TR_CONTROL = timing recovery control

Parameters: Background control value = 0, Trigger control value = 1, Trigger period = 120 bauds. (*NOTE*: TR_AVG is thus set to 1 frame for all simulations)

Front-end blocks:

LINECODE: Implements 2B1Q linecoding scheme

Inputs: b_k = output of DATA

Outputs: a_k = 2B1Q symbols $\{\pm 3, \pm 1\}$

CONVL_SAMP: Linear convolution and adaptive phase sampling block. In general, it performs a linear convolution of a finite length impulse response with an input signal. Here it is used for the front end sampling and timing generation functions. The channel pulse response and 2B1Q data symbols are used in the convolution. The phase at which the convolution is performed is read in at every symbol via the phase jump control signal output by the phase quantizer block. The channel pulse response is oversampled by a factor equivalent to the required phase jump increment . When the phase jump control signal is 0, the same phase as the previous symbol is used (hold mode), when it is 1, the input phase is advanced by one phase increment, if it is -1, it is retarded by one phase increment. Thus, convolution is performed at symbol rate with the timing phase defined at every symbol. This block connects the linear and feedback paths of the loop.

Inputs: a_k = outputs of LINECODE, JUMP = phase jump control signal from phase quantizer.

Outputs: x_k = convolved received signal

Parameters: filename = name of file containing pulse response (one of B3, B4 and SL loops), initial phase offset value (τ/T) = phase value at which convolution is performed at startup with the phase adjusted from this point.

Adaptive blocks:

ARC: Updates the ARC tap which corresponds to the h_0 term as described in chapter 3. This is used by the phase quantizer for normalization of the dead zone threshold, by the ATQ for the decision thresholds, by the error block for the computation of the error at every symbol and by the timing estimate generator block for the calculation of EST#3, as described in the previous chapter.

Inputs: DRX = output of ATQ, ERR = output of ERROR

Outputs: ARC_tap = estimated channel gain at every symbol

Parameters: Initial ARC_tap value = 0 (throughout), β = adaptation step-size.

T_DFE: Transversal decision feedback equalizer uses past decisions to cancel the postcursor ISI as described in chapter 3. It consists of an N-tap transversal filter.

Inputs: DRX = output of ATQ, ERR = output of ERROR block

Outputs: ISI = total ISI calculated at every symbol

Parameters: N = number of taps, β = adaptation step-size

ERROR: Calculate the error in the received signal at every symbol as described in chapter 3. Its output is used by the T_DFE and ARC blocks for the LMS adaptation.

Inputs: RES = received signal after ISI cancellation, DRX = output of ATQ, ARC_TAP = output of ARC

Outputs: ERR = error signal estimated at every symbol

ATQ: Automatic Threshold Quantizer slices the input signal to decide on a data symbol estimate as described in chapter 3. It is a 4-level uniform 2B1Q quantizer with thresholds (0, $\pm$2*ARC_TAP)

Inputs: RES = the clean received signal, ARC_TAP = output of ARC

Outputs: DRX = decided 2B1Q data symbol.

TIME_EST: Computes a timing estimate of maximum length m=3, based on the Mueller and Müller method of general form: $z_k = g_0x_k + g_1x_{k-1} + g_2x_{k-2}$. It is used to implement the four timing estimates derived in chapter 5.

Inputs: RES = received clean signal, DRX = output of ATQ

Outputs: z_k = timing estimate calculated at every symbol

Parameters: METHOD = used to choose which estimate to calculate (see source code for exact description).

LPF: Generates a filtered version of the input signal. All the variables and states are reset upon a high control input.

Inputs: z_k = timing estimate , CONTROL = output of TRIGGER

Outputs: TF_AVG = filtered timing estimate output once per CONTROL trigger.

PHASE_QUANTZ: Quantizes a signal to obtain a fixed phase jump control signal. The following modes are allowed:

If CONTROL asserted:

0=hold	if |signal| ≤threshold
1=advance	if signal < - threshold
-1=retard	if signal > threshold

Otherwise: hold

Inputs: TF_AVG = output of LPF, ARC_TAP = output of ARC, CONTROL = output of TRIGGER

Outputs: phase jump control , sampling phase (τ/T) used for plotting

The same adaptation step-size β is used for both the T_DFE and ARC blocks. Its value is chosen as follows:

$$\beta = \frac{1}{2N\langle a^2 \rangle} \tag{6.1}$$

where N is the number of T_DFE taps, and $\langle a^2 \rangle$ is the MS power in the 2B1Q symbols calculated as $\langle a^2 \rangle = \sum_i a_i p_i$ where p_i is the probability of each symbol with assumption of equiprobability, and i is the length of the alphabet.

The topology described in this section will now be used to compare the performance of the four timing estimates. The performance criteria used in the comparison will be described in the following section.

6.3.Definition of Performance Criteria

The simulation results are all based on steady-state behavior. For this, we define steady-state as the point at which the RMS phase jitter is less than or equal to 3% of the signalling period (~ 30 dB down). This is somewhat of a less stringent constraint compared to the requirement of 0.1% of a symbol period stated in chapter 2. However, it was selected to accomodate the high noise levels that will be used and proved to be sufficient for our purely comparative purpose. The estimates are compared regarding:

1 - Peak to peak phase jitter $= \frac{\text{(largest phase reached - smallest phase reached)}}{T}$

2 - RMS phase jitter $= \frac{\sqrt{\langle \text{phase}^2 \rangle - \langle \text{phase} \rangle^2}}{T}$

3- Fraction of phase jitter $= \frac{\text{number of phase jumps}}{\text{total number of phase corrections}}$

4- Phase offset $= \frac{\text{average steady-state phase} - \tau_o}{T}$

All of the above factors are normalized with respect to the signalling period and will be given in a percentage fraction form for convenience.

6.4.Simulation for Channel Distortion and Additive Noise

Given the receiver topology of Figure 6.1, we ran simulations for the four estimates for the B3, B4 and SL loops for various initial phases and DZT factors, with and without additive noise. Multiple initial starting phases were tried to ensure that the estimates are unbiased to a particular sampling point. This characteristic is critical in the convergence of the timing recovery loop and heavily depends on the channel response. This behavior usually occurs for one particular phase with all other starting points converging to the correct sampling phase. As such it is difficult to verify for all possible initial phases for all the loops, and only a small sample of initial phase offsets can be tried in a testing environment. However, in a practical system, the starting point is unknown and convergence to an incorrect sampling phase could cause unrecoverable errors. In our simulation, it was found that the SL loop exhibited such behavior for one phase offset. This is due to the receive filter that shapes the channel response for timing recovery. As mentioned in chapter 5, the receive filter design used is not optimum further improvement is required. In the SL loop, the filter produced two precursor zero crossings approximately one symbol away from each other (h_{-1} and h_{-2}). The timing recovery loop converges to the one phase as opposed to the other depending on the shortest distance between the initial phase and the two zero crossings. If the initial phase is closer to the optimum phase, the timing recovery loop converged to this correct phase consistently for all estimates. On the other hand, when the initial phase is approximately in the middle of the distance between the zero crossings, convergence is unstable and the phase to which the timing recovery loop locks on to cannot be predicted. This depends on the noise within the system. We should consider that the first precursor undershoot is much bigger in magnitude than the second one and convergence to the wrong phase is more likely to occur for very low noise

levels where the value of the second precursor can be relatively significant. Typically, if the noise level present is very high, the second precursor will be covered by noise such that it won't be detected by the timing recovery blocks. Also, if the second zero-crossing is further than a symbol away, it will be undetected due to the periodicity of the timing phase within a symbol period. Finally, we would like to point out that the SL loop is the most unstable for convergence because of the large postcursor magnitude (relatively to the main cursor), which can cause frequent decision errors (initially) and error propagation. This is especially true for timing estimates #3 and #4 which rely on the ARC and DFE tap estimates respectively. The initial adaptation parameters (DFE taps, ARC and initial phase) should be optimized for the startup procedure such that receiver lock-up to the desired sampling phase is possible even if a reduction in the convergence speed is entailed. It is important to note that multilevel startup (decision-directed convergence) as used in our simulations is not common for existing data communication systems employing multilevel transmission. Binary training where the threshold levels for the slicer are initially known and the noise margin is higher resulting in more reliable decisions for the adaptation of the receiver, is more popular than multilevel training (e.g. four-level) which relies on the received signal for its decisions and thus results in convergence problems even under ideal channel conditions. However, the timing estimates were derived for four-level data and as such the conditions which guaranteed an unbiased, low-variance estimate are no longer satisfied for binary data. This means that we need two different timing estimates (binary and 2B1Q) if binary training is to be used in our multilevel transmission system. The behavior of the timing estimates during convergence will not be investigated in this thesis and multilevel convergence is used throughout the simulations. Problems as discussed above were only encountered for the SL loop which will be excluded from this point on

from our discussion, and correct convergence was obtained for all other cases (B3 and B4 loops).

In the next section, we will discuss the results of the simulations performed in the half-duplex receiver for the four timing estimates for the various additive noise levels . The system parameters used are summarized as follows:

- Simulation length: 1 second
- Loops: B3 and B4
- SNR levels: ~ 24 dB, 16 dB, 13 dB and 10 dB
- DZT = 5 (compromise value reached through trial and error, its effect will be discussed in later sections).
- Initial phases are chosen arbitrarily (neither worst not best case) $\tau/t = 0.2$ to 0.3.
- DFE made long enough to cancel most significant postcursor ISI. (N on the order of 17 - 28 taps). There is always some residual ISI, especially due to precursor effect which is not eliminated.

6.4.1.Performance of Estimate #4

The above receiver configuration resulted in the timing recovery loop failing to converge for estimate #4. The B3 and B4 loops were tried for various initial phases and DZT factors, with and without additive noise. The timing recovery loop failed in all cases. This seems to be caused by the dependency of estimate #4 on the h_1 term, the first postcursor value. Since the adaptations of the DFE, ARC and TR loops are performed simultaneously, there is a very strong interaction between the various DSP functions. This becomes more obvious when we take into account the successful performance of estimate #4 for the truncated channel (very small h_1 term) as described in chapter 5. In this

instance, convergence to the correct steady state timing phase occurred for all cases. This problem should be investigated in more detail by studying the interaction between the DSP blocks. Estimate #4 could still be used, especially if equalization and timing acquisition are not required to converge simultaneously. For example, one of the DSP functions could be initially trained alone and after a certain elapsed time or after satisfying some error criterion, the other blocks are started up. An instance of this in our application would be the blind search timing acquisition described in chapter 3.

6.4.2.Results for Estimates #1,#2 and #3

The performance measures defined in section 6.3 are calculated for the three estimates for the receiver parameters given above, and the results are shown in Table 6.1. Each entry consists of average values for both B3 and B4 loops.

SNR	ESTIMATE #1	ESTIMATE #2	ESTIMATE #3
Average Peak to Peak Phase Jitter			
24 dB	0.008	0.016	0.004
16 dB	0.016	0.04	0.012
13 dB	0.028	0.06	0.024
10 dB	0.072	0.096	0.024
Average RMS Phase Jitter			
24 dB	0.0034	0.0032	0.0048
16 dB	0.0045	0.0069	0.0049
13 dB	0.0068	0.012	0.0082
10 dB	0.014	0.018	0.009
Average Fraction of Phase Jumps			
24 dB	0.0015	0.013	0.00045
16 dB	0.012	0.12	0.0019
13 dB	0.046	0.21	0.005
10 dB	0.20	0.38	0.025
Average Phase Offset			
24 dB	0.02	0.004	0.024
16 dB	0	0	0.012
13 dB	0	0	0
10 dB	0	0	0

Table 6.1: Effect of additive noise and channel distortion with average values for B3 and B4 loops.

Looking at the overall results, it is obvious that estimate #3 performs better on average than both estimates #1 and #2. For instance, estimate #3 yields peak to peak phase jitter values of up to 4 times lower than those of estimate #2 and up to 3 times better than those of estimate #1. Note that for high SNR levels, the results for estimates #1 and #3 are close (same order of magnitude). However, looking at the RMS phase jitter values, estimate #1 shows slightly better results. These values could be misleading if not interpreted correctly. The RMS phase jitter, as derived in section 6.3 measures the deviation of the steady-state phase with respect to the average steady-state phase. If the peak to peak phase deviation is approximately equal for two estimates, the phase which oscillates more frequently and symmetrically around the average steady-state phase will show a lower RMS phase jitter. This behavior can be seen in the phase plots for the three estimate in Figure 6.2. The peak to peak phase jitter is much worse for estimate #2. Even though it behaves more like that of estimate #1 as opposed to estimate #3, the RMS phase jitter values are worse than those of estimate #3. For estimates #1 and #3, when the peak to peak phase jitter values are similar (SNR's = 24,16 13 dB), estimate #1 yields lower RMS phase jitter. But, for SNR=10dB, the peak to peak phase jitter is much higher than estimate #3 and thus its RMS phase jitter is also higher.

For the average fraction of phase jump, the resulting values differ greatly between the three estimates. Again, estimate #3 gives very low average values, followed by estimate #1 with results of 3 to 9 times higher than the former. Estimate #2 yields worst results of up to 16 times higher than estimate #3 (SNR = 16 dB). It is interesting to note that the ratio of the fraction of phase jumps of estimate #2 over estimate #3, which gives the relative difference in the results between the two estimates, is high for high SNR and decreases with the SNR level, whereas the ratio of estimate #1 over estimate #3 increases as the SNR decreases.

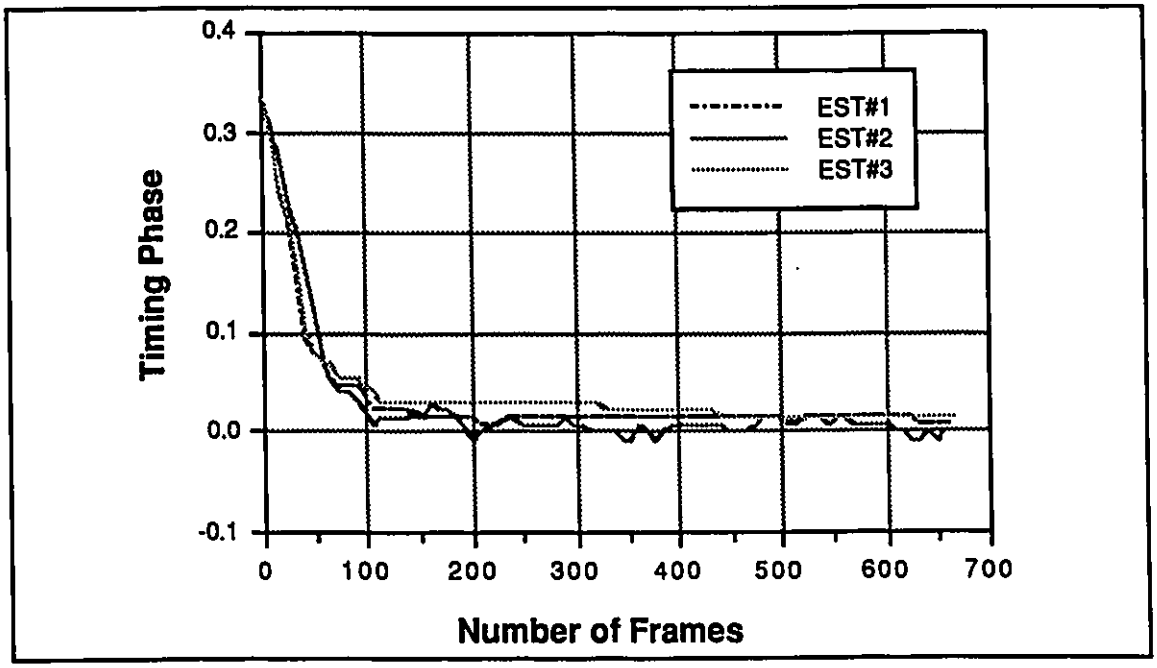

Figure 6.2: Phase Behavior for Estimates #1,#2 and #3, for the B3 Loop, SNR = 16 dB

Finally, the average phase offset given in the last section of Table 6.1 shows contrasting results compared to the other criteria discussed above. The steady-state phase reached by estimate #2, on average, is much closer to τ_0 than estimate #1, but especially estimate #3. However, the results improve as the SNR level decreases. This can be somewhat explained by the DZT. The DZT factor was chosen for SNR=∞ such that the timing phase jitter is zero for the worst case (estimate #2). But, for a very high SNR, the jitter within the system is very small. For a DZT factor of 5, the threshold in the phase quantizer might be too high such that τ_0 can never be exactly reached, because timing adjustments stop as soon as the timing average is within the threshold limits. As the noise increases, the jitter present in the timing information is increased such that a high DZT will not stop the timing phase from reaching τ_0. On the other hand, because the timing

information is more noisy, it becomes more difficult to stay locked to τ_0 exactly. For our comparison, the choice of the DZT is not critical. It is sufficient to verify that for practical SNR levels, the three estimates perform as expected, without any significant differences between them.

In conclusion, the simulation results in this section agree with the theoretical findings of chapter 5. The comparative effect of additive noise and channel distortion on the three estimates is as predicted by the variance expressions and timing jitter power in the previous chapter. Estimate #3 gives the best overall performance, followed by estimate #1, with finally, estimate #2 resulting in much higher timing phase jitter levels. In the next section, we will examine how the DZT factor affects the phase behavior of each timing estimate. We will compare the three estimates and verify that the results correspond to those found in this section.

6.5.Effect of the DZT Factor

In this section, we fixed all system parameters and ran the topology of Figure 6.1 for each of the three timing estimates for various DZT factors. The configurations for the parameters are summarized below:

- Simulation length: 1 second
- Loops: B3 and B4
- SNR: 14 dB
- DZT: 0,1,2,3,4,5,6,7
- Initial phase: (chosen arbitrarily) $\tau/t = 0.2$ to 0.3
- DFE made long enough to cancel all postcursor ISI with $N = 28$.
- $\beta = 0.0036$

All the performance measures defined in section 6.3 were calculated for every simulation run, and average values are given in Table 6.2 for each estimate.

DZT	ESTIMATE #1	ESTIMATE #2	ESTIMATE #3
Average Peak to Peak Phase Jitter			
0	0.07	0.066	0.054
1	0.054	0.066	0.038
2	0.052	0.064	0.034
3	0.044	0.056	0.02
4	0.012	0.06	0.02
5	0.03	0.06	0.022
6	0.02	0.048	0.024
7	0.036	0.048	0.028
Average RMS Phase Jitter			
0	0.013	0.013	0.0094
1	0.010	0.013	0.0075
2	0.009	0.011	0.0056
3	0.008	0.011	0.0054
4	0.0068	0.011	0.0058
5	0.0073	0.012	0.009
6	0.0062	0.011	0.011
7	0.0098	0.011	0.010

Table 6.2: Effect of dead-zone threshold with average values for B3 and B4 loops, using SNR=14dB.

DZT	ESTIMATE #1	ESTIMATE #2	ESTIMATE #3
Fraction of Phase Jumps			
0	1.0	1.0	1.0
1	0.71	0.8	0.545
2	0.44	0.6	0.22
3	0.223	0.445	0.05
4	0.105	0.31	0.0051
5	0.046	0.225	0.005
6	0.0165	0.14	0.0055
7	0.0125	0.072	0.0055
Average Phase Offsets			
0	-0.0043	-0.00624	0.00312
1	-0.0035	-0.00624	0.00196
2	-0.003	-0.00408	0.0008
3	-0.00196	-0.00328	0.00156
4	-0.00164	-0.0022	0.00116
5	0.00196	0.00196	0.0097
6	0.00488	0.00196	0.012
7	0.007	0.00604	0.0318

Table 6.2: Effect of dead-zone threshold with average values for B3 and B4 loops, using SNR=14dB (continued).

The results in Table 6.2 correspond to those found in the previous section. Between the three estimates, estimate #3 has the overall lowest peak to peak phase jitter values, as well as fraction of phase jumps and phase offset. However, the phase behavior for the three estimates differ as the DZT factor is varied. Estimate #3 deteriorates in performance at a much higher rate than either estimate #1 or #2. Estimates #1 and #2 are not as affected by the high dead-zone values. Estimate #2 is especially more 'resistant' to the dead-zone effect. This can be seen in its resulting peak to peak and RMS phase jitter that remain almost constant over a wide range or DZT values. Estimate #1 yields bad results for very low and very high DZT values, and is at its best in the middle of the range. However, contrary to estimate #3, its performance does not vary as greatly with the DZT. The different phase behavior of the three timing estimates was discussed in the previous section, and is shown again in Figure 6.3 (a) to (c). In these plots, we show the convergence of the phase for the three estimates for different DZT values. It is important to note that the phase offset results are as predicted in section 6.4.2. Due to the higher 'sensitivity' of estimate #3 to the effect of the DZT, the phase offset tends to be high, especially for high DZT factors. We can see that estimate #3 has lower phase offsets than the other two estimates for the small DZT values. As the DZT becomes large, the phase offset resulting from estimate #3 becomes worse than both other estimates. However, this is dependent on the SNR level as well. The effect of the DZT on the performance of the timing estimates as the SNR level decreases. In a practical system, channel distortion and noise are among the largest contributing error sources. Since the DZT could not be varied, it cannot be chosen large enough for worst case SNR levels. It should accommodate typical system conditions that occur most frequently. Therefore, a compromise DZT value is chosen, typically corresponding to the middle of our range.

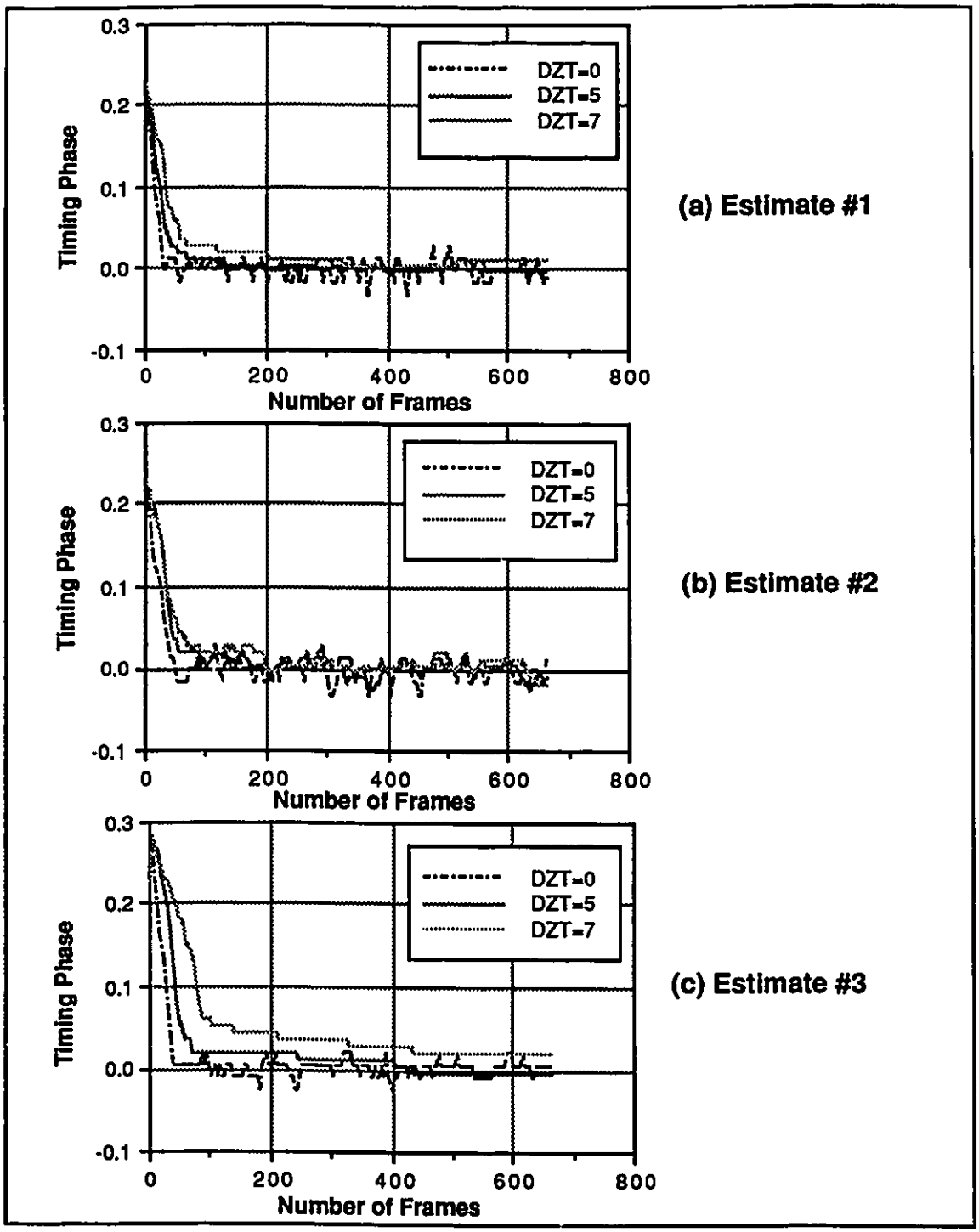

Figure 6.3: Phase Behavior for Estimates #1,#2 and #3 for the B4 loop, SNR = 14 dB, DZT = 0,5,7

For this reason, the effect of the DZT on estimate #3 is not significant in a practical system. Its improvement over estimates #1 and #2 with respect to noise and channel distortion will be the bigger influence on the timing recovery performance. As we have seen in section 6.4.2 as well as in chapter 5, this improvement is quite significant and should be considered when choosing the timing estimate. Again, the choice of the timing estimate depends on the overall system, the channel and particular applications. The optimal choice of the DZT depends on the timing estimate and its performance with respect to the system, channel, as well as on the DZT itself. However, in general, estimate #3 gives the best overall performance for most situations. Moreover, if the timing phase jitter criterion for a particular system is very strict, estimate #3 might be the only possible choice.

In the next section, we will briefly investigate the performance of the three timing estimates for varying initial phase offsets.

6.6.Effect of the Initial Starting Phase

The performance of the three timing estimates is analyzed as in the previous sections using the initial starting phase as the reference. Simulations for the three estimates were performed using six uniformly spaced starting phases and the worst case possibility. The system parameters are as follows:

- Simulation length: 1 second
- Loops: B3 and B4
- SNR: ~14 dB
- DZT = 5
- Initial phase offsets: $\tau/T = \pm 0.07, \pm 0.2, \pm 0.3, \pm 0.5$
- DFE made long enough to cancel all postcursor ISI: N = 28
- $\beta = 0.0036$

The results are summarized below in Table 6.3. Each section of the table includes average values for both loops.

iph	ESTIMATE #1	ESTIMATE #2	ESTIMATE #3
Average Peak to Peak Phase Jitter			
±0.05	0.032	0.064	0.024
±0.2	0.034	0.062	0.026
±0.3	0.030	0.06	0.026
±0.5	0.032	0.064	0.03
Average RMS Phase Jitter			
±0.05	0.0072	0.0120	0.0066
±0.2	0.0075	0.0120	0.0078
±0.3	0.0070	0.0121	0.009
±0.5	0.0072	0.0124	0.0096
Average Fraction of Phase Jumps			
±0.05	0.052	0.25	0.0061
±0.2	0.050	0.23	0.0066
±0.3	0.050	0.22	0.007
±0.5	0.05	0.23	0.0074

Table 6.3: Effect of initial starting phase with average values for B3 and B4 loops, using SNR=14dB.

The steady state performance of a timing estimate with respect to timing phase jitter should not be affected by the initial phase. This results directly from the Mueller and

Müller algorithm as described in the previous chapter. The starting sampling instant should only affect the convergence time. This will be examined in the next section. Here, we verify that the timing estimates are not biased towards any sampling point and that convergence to the optimal phase occurs independently of the initial sampling phase. It is important to verify that there exists only one phase which satisfies $f(\tau_0) = 0$, such that the timing recovery loop doesn't converge to a bad sampling point. Also, the peak to peak and RMS phase jitter as well as the fraction of phase jumps, measured at some point after steady state as we have defined it, should not be significantly different for the various initial phases. Our results concur with the latter in every case. The levels for each criteria are, on average, very close for all the initial phases given the same timing estimate.

Estimates #1 and #2 are very consistent in that manner. Estimate #3 seems to behave the worst for larger initial phase offsets. Again, this could be explained by its higher sensitivity to the DZT. As explained previously, a high DZT factor could significantly affect the performance of estimate #3, especially when compared to the other estimates. The small discrepancies in the results of estimate #3 are caused by this factor since a value of 5 for the DZT is not negligible even for the low SNR of 14 dB. However, the differences in the results are very minimal with the worst case yielding maximum values of only 1.4 times higher. Here, the crucial issue is to examine the effect on the steady state phase offset. On average, we obtain similar steady state phase offsets values for all the initial starting points for each of the three estimates, including estimate #3. The results are as stated in Table 6.2 for the DZT factor of 5.

In order to conclude that the performance of the timing recovery loop is not affected by the starting sampling phase, it is sufficient to verify that the phase to which the timing recovery loop converges, and its performance within that steady state region, is essentially the same independent of the initial starting point given similar overall system conditions.

This is verified for all three estimates as can be seen in the above discussion. It should also be noted that the relative performance of the three estimates is consistent with the previous results of sections 6.4 and 6.5.

6.7.Comparison of Convergence Time

This section examines the convergence behavior of the three estimates by measuring the convergence time for each of them for a variety of system parameters. B3 and B4 loops are considered, each taken at four different initial phase offsets for five SNR levels. The results are compiled in histograms, one per loop for every estimate. convergence time is defined as the time it takes the timing recovery loop to reach the steady state sampling phase as defined in section 6.3. The simulation parameters are as follows:

- Simulation length: 1 second
- Loops: B3 and B4
- SNR : 10.5, 11, 11.5, 12 and 14 dB
- DZT = 0
- Initial phase offsets: 0.05, 0.2, 0.3 and 0.4
- N = 28
- $\beta = 0.0036$

Even though the SNR levels are low such that the DZT should not affect the performance of the timing estimates, especially the convergence time, the DZT is set to zero to completely eliminate any such possible effect. In this case, phase adjustments will occur continuously until the steady state sampling phase is exactly reached. The effect of the DZT on the convergence time will not be investigated, however it is obvious that a larger DZT will increase (lengthen) the convergence time. In practice, this should not be significant because during convergence, the channel distortion is much higher than the DZT threshold. The effect of the DZT on the convergence time will also depend on the noise

level as well as the channel pulse response. Our objective is to briefly compare the convergence time for the three estimates using a sample group from the infinite number of situations possible in such a system. The sample group is chosen large enough such that it conveys a good representation of the overall convergence behavior of the timing estimates in the given system. Very low SNR levels are included to estimate worst case convergence times. The results are shown in Figures 6.4 (a) to (c) for the B3 loop, and Figures 6.5 (a) to (c) for the B4 loop. For the B3 loop, convergence occurs in less than 300 ms in 65% to 75% of the cases. B4 loop shows better results, with similar convergence time in 80% to 100% of the cases. The B3 loop results in very long convergence times for all three estimates for the low SNR levels of 10.5 and 11 dB. The timing recovery loop failed to converge for SNR levels less than 10 dB for all three estimates and for both loops with a negligible relative difference of less than 1 dB between the three estimates. On average, estimate #3 yields the lowest convergence time for both loops. The results for estimate #2 and #1 are very close. Estimate #1 is slightly better, giving a higher percentage for convergence times of less than 150 ms. However, overall, the differences in the results between estimates #1 and #2 are not significant. The convergence behavior was not analyzed theoretically and our discussion is based on simulation results only. This exercise helps us to globally compare the convergence behavior of the three estimates such that any significant discrepancies are revealed. This is important because steady state behavior in itself is not sufficient to truly compare the performance of the three timing estimates. This brief comparison of convergence time allows us to complete the characterization of the estimates. Our results in general, agree for all stages in the operation of the timing recovery loop, such that further analysis should not reveal any differing outcome which would weaken the validity of the overall comparison.

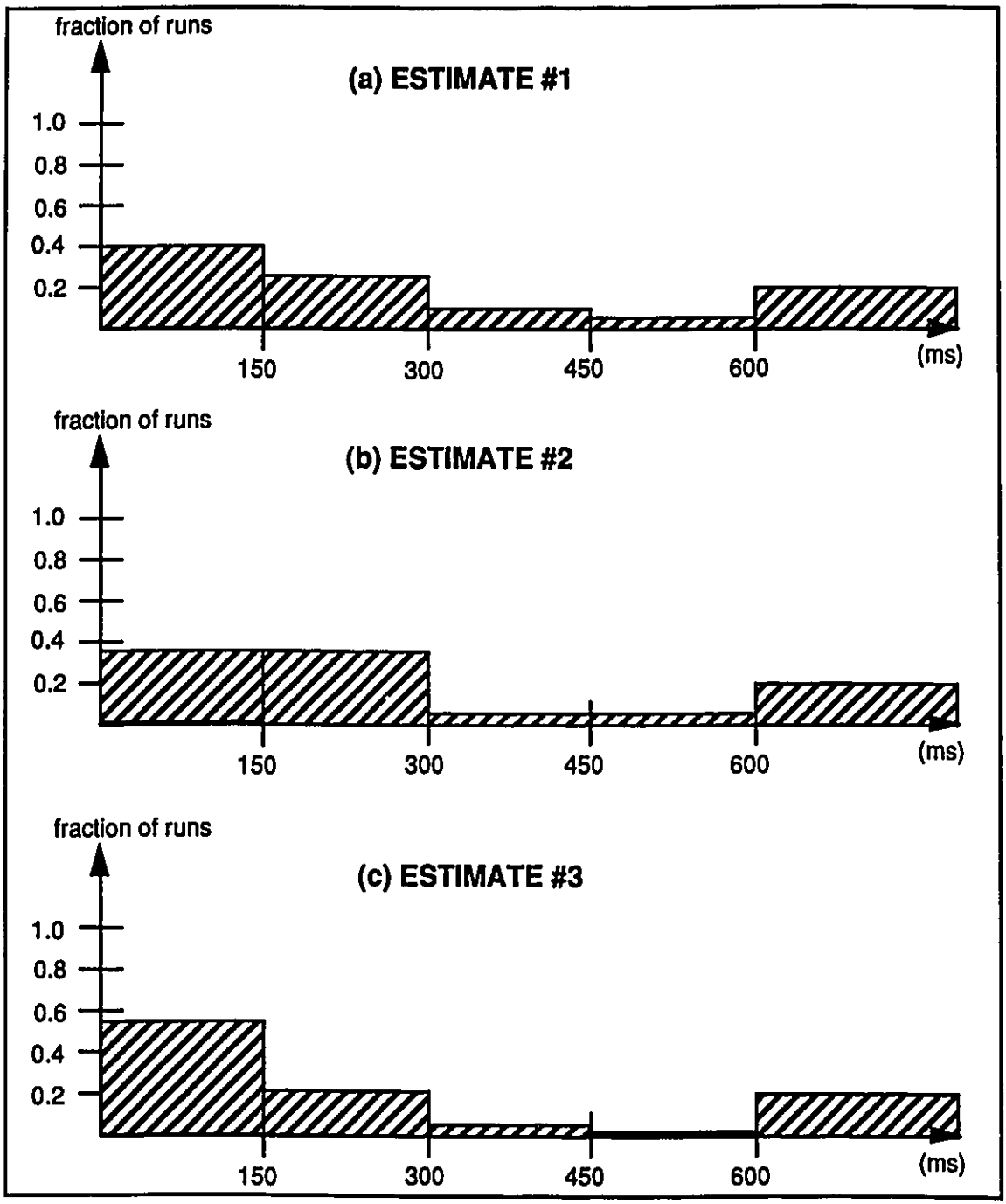

Figure 6.4: Convergence Histograms for Estimates #1, #2 and #3 for the B3 loop

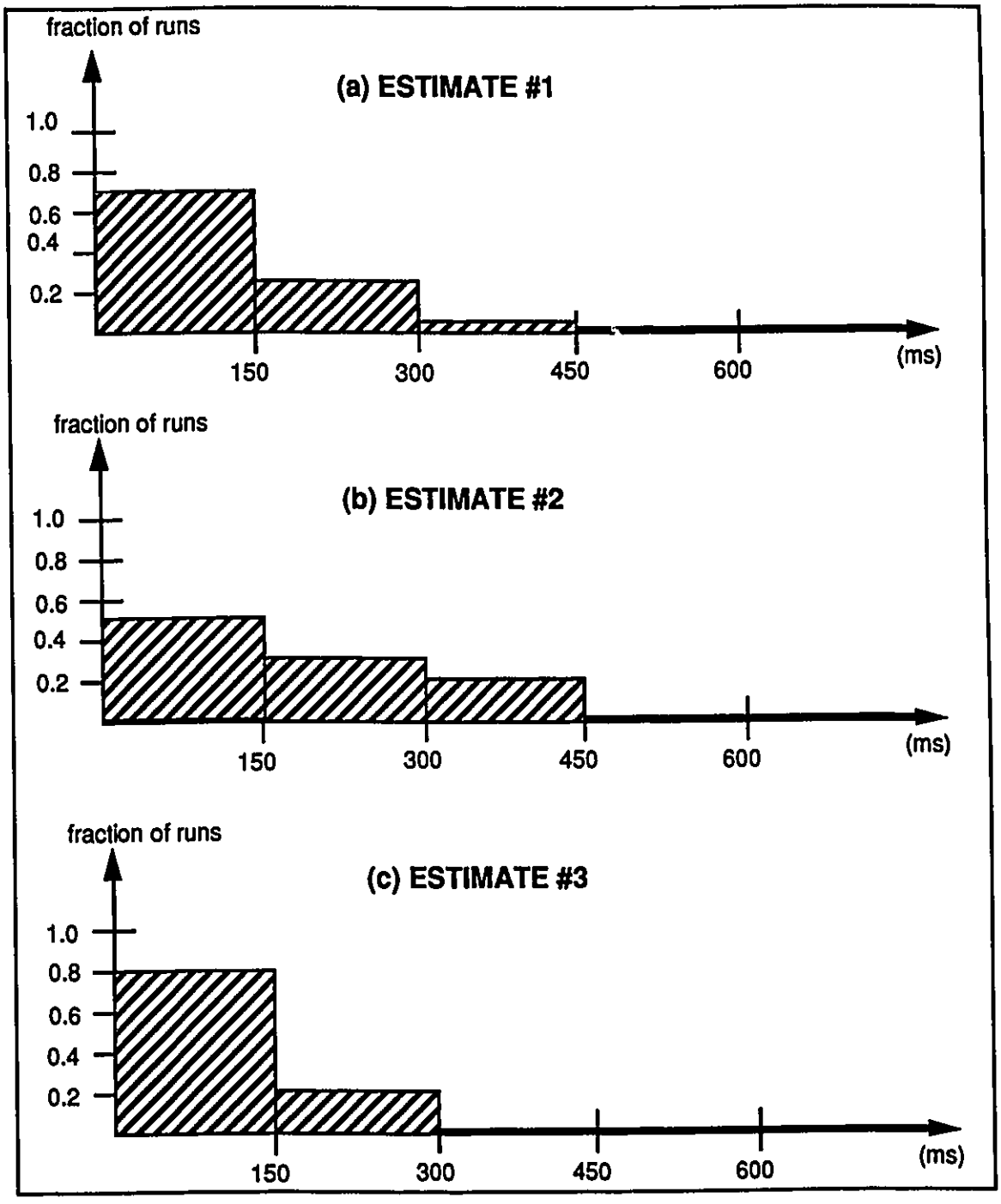

Figure 6.5: Convergence Histograms for Estimates #1, #2 and #3 for the B4 Loop

6.8.Summary

In this chapter, we used software simulation as a tool to compare the performance of the timing estimates in the practical timing recovery loop. Code was written for the CAPSIM/BLOSIM environment for all the blocks needed to implement the system described in chapter 3. The effects of additive noise, channel distortion, dead-zone threshold and initial phase offsets on the steady state performance were investigated. In all cases, our results agree with those expected from our theoretical analysis in chapter 5. Direct relation between timing phase jitter quantities calculated from simulation results and those predicted through theoretical analysis is not possible due to the complexity of the system and strong interactions between blocks. However, the relative performance of the timing estimates can be directly compared. Accordingly, we found that estimate #3 performs best overall with higher margin for noise and channel distortion and lower steady state phase jitter. Estimate #1 is chosen as second best with results close to those of estimate #3 more frequently than those of estimate #2. This latter produces the worst steady state performance with significant differences as predicted in chapter 5. The convergence time was also examined and similar results were found. The results pertaining to the effect of the DZT showed estimate#3 with a higher 'sensitivity' to this threshold than both other estimates. However we found that this effect can be quite easily reduced. In general, the performance of estimate #3 was better for low and high dead zone thresholds. Finally, estimate #4 was found unstable in its performance in the half-duplex receiver of Figure 6.1 and was thus suppressed from the comparison. A brief discussion was given to that effect.

CHAPTER 7

Conclusions

7.1.Concluding Remarks

In this thesis, we have examined the behavior of four timing estimates based on a symbol rate timing recovery algorithm described in [MueM76], for the timing function $f(\tau)=h_{-1}$, as it applies to DSL for ISDN applications in a 2B1Q U transceiver. Theoretical and simulation results were compared for the four timing estimates given in Table 5.1. We show that even though all satisfy the conditions stated in [MueM76], and all are of equal length m=2, their performance in a practical system differ greatly.

It was found from the variance expressions given in Table 5.2, that estimate #3 results in the smallest error due to channel distortion and additive noise. Estimate #4 was second best with an error that is twice as high as that of estimate #3. Then follows estimate #1, and finally estimate #2 with an error of ~ 7 times that of estimate #3. Estimate #4 showed better results for the effect of the average length TR_AVG, yielding the lowest MSE_{TR_AVG} given in (5.2). It was followed by estimate #3, then estimate #2 with estimate #1 being the worst. This effect was found to be negligible for the practical

TR_AVG values of 1 frame and higher, but should be taken into consideration if only one or a few symbol rate samples of z_k are used to estimate $f(\tau)$. A linearized model of the PLL for the timing recovery loop was developed and the output jitter power spectrum was compared for the four estimates with results similar to those found from the variance expressions.

A half-duplex receiver that corresponds to the U-transceiver described in chapter 2 was simulated and the behavior of the four estimates with respect to the various system parameters for three test loops was investigated. Estimate #4 failed to converge for all cases when the various DSP functions were added to the system. This seems to be caused by its dependence on the first postcursor term, resulting in a strong interaction with the other DSP blocks (equalization). Estimate #3 yielded the best results, giving the lowest steady state phase jitter for high and low SNR and MSD values. It was followed by estimate #2 with very close results for high SNR levels. Estimate #2 gave the highest timing jitter for steady state behavior. The simulation results, in general, agreed with the those found theoretically, with the exception of estimate #4. The choice of the dead-zone threshold was discussed, and estimate #3 was found to be the most sensitive to this effect. The steady state phase offset was high for large DZT values, especially when compared with the other two estimates. However, in situations where noise is predominant, this would be an advantage at steady state. Convergence times were also compared. These resulted in shortest convergence times for estimate #3, followed closely by estimate #1 and finally estimate #2 taking the longest times, on average, to reach the defined steady state region.

The above results were obtained for two test loops with and without bridge taps. The short loop (SL loop) was found to have problems converging to the right phase

because of a second precursor zero-crossing. This would imply that the design of the forward equalizer is very important to ensure guaranteed convergence to the right phase.

In conclusion, the Mueller and Müller algorithm allows you an infinite choice of timing estimates for the same timing function and line code. The optimal choice depends on the particular system, channel response and present noise. Ease of implementation and economical limitations are also important factors that should be considered. Estimate #3 and estimate #4 both have a direct dependence on terms of the channel response. This might prove more complex to implement in a practical system. Estimates #1 and 2 are similar in their complexity, however, their performance greatly varies. For digital subscriber loop applications, an optimal solution might be hard to identify analytically due to the wide range of loops and variety of states of the system. A brief preliminary theoretical analysis of the problem with more elaborate simulation performed on a large sample of applicable situations proves to be the best practical solution.

7.2.Recommendation for Future Works

A detailed study is essential to determine why estimate #4 fails to converge in our practical receiver. The interaction between the various DSP blocks should be more closely investigated, and the startup procedure for the various blocks to optimize receiver convergence and decouple the interaction of the blocks should be developed. This could solve the convergence failure of estimate #4. The design of the receive filter and its effect on the timing recovery loop requires further study. The performance of the timing estimates should also be verified for the short loop.

Full-duplex integration of the U transceiver blocks should be completed and simulated, to compare the performance of the four estimates in full-duplex mode, taking into consideration echo cancellation requirements and the various noise sources described in chapter 2 (crosstalk, impulse, echo noise).

The use of a higher order PLL (second order loop) in the timing recovery block could be investigated to improve the performance of the timing recovery algorithm. This will allow the use of previously detected timing errors for the adjustments of the input phase. A closer look at the choice of the averaging length during convergence and steady state could improve overall timing acquisition Finally, a more detailed analysis of the PLL to characterize the convergence behavior would greatly improve the understanding of the problem and facilitate design changes for overall system improvement.

References

[ACCC87] P.F.Adams, S.A.Cox, R.B.P.Carpenter and N.G.Cole,"A Long Reach Digital Subscriber Loop Transceiver," British Telecom Technical Journal, Vol.5, No.1, January 1987.

[AdCG84] P.F. Adams, S.A.Cox, P.J.Glen,"Long Reach Duplex Transmission Systems for ISDN Access", British Telecom Technical Journal, Vol.2, No.2, April 1984.

[Aly88] Sami Aly, "ISDN U Inteface Transceiver Using 2B1Q Line Code", BNR internal document, March 1988.

[AnMM86] A.N.D'Andrea, U.Mengali and M. Maro, "Nearly Optimum Prefiltering in Clock Recovery", IEEE Transactions on Communications, Vol.COM-34, No.11, November 1986.

[ANSI89] American National Standard for Telecommunications, "Integrated Services Digital Network (ISDN) - Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT Layer 1 Specification", ANSI T1.601 - 1988.

[ATMH85] Oscar Agazzi, C.P. Jeremy Tzeng, David G. Messerschmitt and David A. Hodges, "Timing Recovery in Digital Subscriber Loop", IEEE Transactions on Communications, Vol.COM-33, No.6, June 1985.

[Benn58] W.R. Bennett, "Statistics of Regenerative Digital Transmission", Bell Systems Technical Journal, November 1958.

[Cook89] C.G. Cook, "Two Stage Data Echo Cancellation Utilizing Orthonormal Functions for Echo Tail Cancellation", Master thesis, Carleton University, February 1989.

[EhrT82] C.A. Ehrenbard and M.F. Tompsett, "A Baud-Rate Line-Interface for Two-Wire High-Speed Digital Subscriber Loop", Globecom'82, 1982.

[Falc85] David D. Falconer, "Timing Jitter Effects on Digital Subscriber Loop Echo Cancellers - PART I: Analysis of the Effect - PART II: Considerations for Squaring Loop Timing Recovery", IEEE Transactions on Communications, Vol.COM-33, No.8, August 1985.

[Falc82] David Falconer, " Adaptive Reference Echo Cancellation", IEEE Transactions on Communications, Vol.COM-30, No. 9, September 1982.

[FogG89] Eli Fogel and Motti Gavish, "Performance Evaluation of Zero-Crossing-Based Bit Synchronizers", IEEE Transactions on Communications, Vol.37, No. 6, June 1989.

[FOYT89] M. Fukuda, S. Ohta, K. Yamaguchi, T. Tsuda, T. Gotohda, H. Gambe, S. Miyoshi and Y. Awata, "An Approach to LSI Implementation of a 2B1Q coded Echo Canceller for ISDN Subscriber Loop Transmission", IEEE-ICC'89 Conference Record, 1989.

[FraB74] L.E.Franks and J.P.Bubrouski,"Statistical Properties on Timing Jitter in a PAM Timing Recovery Scheme", IEEE Transactions on Communications, Vol.COM-22, No.7, July 1974.

[Fran80] L.E.Franks, "Carrier and Bit Synchronization in Data Communication - A Tutorial Review", IEEE Transactions on Communications, Vol.COM-28, No. 8, August 1980.

[FTGH88] Misao Fukuda, Toshitaka Tsuda Hirohisa Gambe and Youzi Hino, " A Line Terminating LSI Using Echo Cancelling Method for ISDN Subscriber Loop Transmission", IEEE Journal on selected Areas in Communications, Vol.6, No.3, April 1988.

[Gard86] W.A.Gardner, "The Role of Spectral Correlation in Design and Performance Analysis of Synchronizers", IEEE Transactions on Communications, Vol.COM-34, No.11, November 1986.

[GBKM89] J. Girardeau, N. van Bavel, W. Kuenast, S.McCaslon, M. Pendleton, T. Williams, S.Aly, B. Sayar, P. Hung, G. Koleyni, A. Deczky, P. Ferland and S. Gaulet, " ISDN U Transceiver Algorithms Development System, and Performance", Globecom'89, 1989.

[GitS71] R.D.Gitlin and J. Salz, "Timing Recovery in PAM Systems", Bell Systems Technical Journal, Vol. 50, No. 5, May-June 1971.

[HaYa88] Mirna Hage and Ken Yamazaki, "CAPSIM/BLOSIM Demonstration", BNR internal document, December 1988.

[HaYa89] Mirna Hage and Ken Yamazaki, "NOVA Manual", BNR internal document, January 1989.

[Hung88] Paul Hung, "Info for ISDN/HOFF Simulation Package", BNR internal document, October 20, 1988.

[JenC85] A. Jennings and B.R. Clarke, "Data-Sequence Selective Timing Recovery for PAM Systems", IEEE Transactions on Communications, Vol.COM-33, No.7, July 1985.

[LeeM88] Edward A. Lee and David G. Messerschmitt, "Digital Communication", Kluwer Academic Publishers, U.S.A., 1988.

[Lyon75] D.L.Lyon, "Timing Recovery in Synchronous Equalized Data Communication", IEEE Transactions on Communications, February 1975.

[Mess86] David G. Messerschmitt, "Asynchronous and Timing Jitter Insensitive Data Echo Cancellation", IEEE Transactions on Communications, Vol.COM-34, No.12, December 1986.

[Mess84] David G.Messerschmitt, "Echo Cancellation in Speech and Data Transmission", IEEE Journal on Selected Areas in communications, Vol. SAC-2, No. 2, March 1984.

[Moen87] Marc Moeneclaey, "Timing Recovery in the Presence of a Residual Echo Signal", IEEE Transactions on Communications, Vol.COM-35, No.8, August 1987.

[MoNP87] C.Mogavero, G.Nervo and Paschetta, "Mixed Recursive Echo Canceller (MREC)", CSELT Technical Reports, Vol.XV, No.2, March 1987.

[MueM76] Kurt H. Mueller and Markus Müller, "Timing Recovery in Digital Synchronous Data Receivers", IEEE Transactions on Communications, Vol.COM-24, No.5, May 1976.

[NoPA89] Ramin A. Nobakht, Prayson Pate and Sasan H. Ardalan, "CAPSIM: A Graphical Simulation Tool For Communication Systems", ICC'89 Conference Records, 1989.

[OerM88] Martin Oerder, Heinrich Meyr, "Digital Filter and Square Timing Recovery", IEEE Transactions on Communications, Vol.36, No. 5, May 1988.

[Pate88] Prayson Pate, " Digital Methods of Timing Recovery", Centre for Communications and Signal Processing, North Carolina State University, CCSP-TR88/12, April 1988.

[Qure76] S. Qureshi, "Timing Recovery for Equalized Partial-Response Systems", IEEE Transactions on Communications, December 1976.

[SaDM86] Hikmet Sari, Lydie Desperben and Said Moidi, "Minimum Mean-Square Error Timing Recovery Schemes for Digital Equalizers", IEEE Transactions on Communications, Vol.COM-34, No.7, July 1986.

[Salt67] B.R.Saltzberg,"Timing Recovery for Synchronous Binary Data Transmission", Bell Systems Technical Journal, March 1967.

[Saya89] Babak Sayar, "Timing Functions for h_{-1}", BNR internal memo, May 30, 1988.

[Stal85] William Stallings, "Data and Computer Communications", Macmillan Pub. Co., U.S.A., 1985.

[StFA88] John T. Stonick, L. James Faber and Sasan H. Ardalan, " CAPSIM User's Guide", Centre for Communication and Signal Processing, North Carolina State University, Raleigh, North Carolina, November 1988.

[SuHM89] Satoru Sugimoto, Kazuhiro Hayashi and Fumio Mano, "Design of 2B1Q Transceiver for ISDN Subscriber Loops", IEEE-ICC'89 Conference Record, 1989.

[SzZS86] Kálmán Széchényi, Franz Zapf and Danny Sallaerts, "Integrated Full-Digital U-Interface Circuit for ISDN Subscriber Loops", IEEE Journal on Selected Areas in Communications, Vol.SAC-4, No. 8, November 1986

[Tzen85] C.P.Jeremy Tzeng, "Timing Recovery in Digital Subscriber Loop", Ph.D. Dissertation, University of California, Berkeley, April 1985.

[TzHM86] C.P.Jeremy Tzeng, David A. Hodges and David G. Messerschmitt, "Timing Recovery in Digital Subscriber Loops Using Baud-Rate Sampling", IEEE Journal on Selected Areas in Communications. Vol. SAC-4, No.8, November 1986.

[Yama89] Ken Yamazaki, "Convergence Behavior of A Jointly-Adaptive Transversal And Memory-Based Echo Canceller", Master thesis, Carleton University, September 1989.

Appendices

Appendix A

Verification of $E\{z_k\}=f(\tau)$

A.1 Assumptions

1. Equiprobable and uncorrelated 2B1Q data symbols with alphabet $\{-1,+1,-3,+3\}$, such that: $E\{a_{k-i}\ a_{k-j}\} = E\{a_{k-i}\}\,Ea_{k-j}\}$ for $i \neq j$.
2. $E\{a_k\} = \sum_L a_{kL} P_L$

   where L = number of symbols in alphabet = 4

   P_L = probability of corresponding symbol $= \frac{1}{4}$ for equiprobable a_k 's.

* Using the above expressions, we get the following:

- $E\{a_k\} = (1)\frac{1}{4} + (-1)\frac{1}{4} + (+3)\frac{1}{4} + (-3)\frac{1}{4} = 0$
- $E\{a_k^n\} = 0$, for n odd.
- $E\{a_k^2\} = (1)^2\frac{1}{4} + (-1)^2\frac{1}{4} + (+3)^2\frac{1}{4} + (-3)^2\frac{1}{4} = 5$
- $E\{a_k^4\} = (1)^4\frac{1}{4} + (-1)^4\frac{1}{4} + (+3)^4\frac{1}{4} + (-3)^4\frac{1}{4} = 41$
- $E\{a_k^6\} = (1)^6\frac{1}{4} + (-1)^6\frac{1}{4} + (+3)^6\frac{1}{4} + (-3)^6\frac{1}{4} = 365$
- $E\{a_k^8\} = (1)^8\frac{1}{4} + (-1)^8\frac{1}{4} + (+3)^8\frac{1}{4} + (-3)^8\frac{1}{4} = 3281$

A.2 Estimate #1

$$z_{1k} = \frac{(a_k^2 - 5)}{16}(a_k x_{k-1} - a_{k-1} x_k)$$

$$E\{z_{1k}\} = E\{\frac{(a_k^2 - 5)}{16}(a_k x_{k-1} - a_{k-1} x_k)\}$$

$$E\{z_{1k}\} = E\{\frac{(a_k^2 - 5)}{16}(a_k x_{k-1})\} - E\{\frac{(a_k^2 - 5)}{16}(a_{k-1} x_k)\}$$

* Replacing $x_k = \sum_i h_i a_{k-i}$

$$E\{z_1(k)\} = E\{\frac{(a_k^2 - 5)}{16} a_k \sum_i h_i a_{k-1-i}\} - E\{\frac{(a_k^2 - 5)}{16} a_{k-1} \sum_i h_i a_{k-i}\}$$

* Taking the first term, we get;

$$E\{\frac{(a_k^2 - 5)}{16} a_k \sum_i h_i a_{k-1-i}\} = E\{\frac{(a_k^2 - 5)}{16} \sum_i h_i a_{k-1-i}\} E\{a_k\} \quad \text{for } i \neq -1$$

* with $E\{a_k\} = 0$, then, the second term = 0 for $i \neq -1$.

$$E\{\frac{(a_k^2\ 5)}{16} a_k \sum_i h_i a_{k-1-i}\} = h_{-1}\ E\{\frac{(a_k^2 - 5)}{16} a_k^2\} \quad \text{for } i = -1$$

$$= h_{-1}\ E\{\frac{(a_k^4)}{16} - 5\frac{(a_k^2)}{16}\} \qquad \text{for } i = -1$$

* with $E\{a_k^2\} = 5$ and $E\{a_k^4\} = 41$, the first term becomes;

$$= h_{-1}\ E\{\frac{41}{16} - 5\frac{5}{16}\} = h_{-1} \qquad \text{for } i = -1$$

* Then the first term $= h_{-1}$ for all i's

* Taking the second term of the above expression;

$$E\{\frac{(a_k^2 - 5)}{16} a_{k-1} \sum_i h_i a_{k-i}\} = E\{\frac{(a_k^2 - 5)}{16} \sum_i h_i a_{k-i}\} E\{a_{k-1}\} \quad \text{for } i \neq 1$$

* We have $E\{a_{k-1}\} = E\{a_k\} = 0$, then the above expression becomes;

$$E\{\frac{(a_k^2 - 5)}{16} a_{k-1} \sum_i h_i a_{k-i}\} = 0 \quad \text{for } i \neq 1$$

$$E\{\frac{(a_k^2 - 5)}{16} a_{k-1} \sum_i h_i a_{k-i} \} = h_1 \; E\{\frac{(a_k^2 - 5)}{16}\} \; E\{ a_{k-1}^2 \} \text{ for } i=1$$

* We have $E\{ a_{k-1}^2 \} = E\{a_k^2\} = 5$, then the above expression becomes;

$$E\{\frac{(a_k^2 - 5)}{16} a_{k-1} \sum_i h_i a_{k-i} \} = h_1 \; \{\frac{E\{a_k^2\}}{16} - \frac{5}{16}\} E\{a_k^2\} = h_1 \{\frac{5}{16} - \frac{5}{16}\}5 = 0$$

* the second term = 0 for all i's.

* Hence, $E\{z_{1k}\}$ = E {first term} - E {second term}, resulting from above,

$$\boxed{E\{z_{1k}\} = h_{-1} = f(\tau)}$$

A.3 Estimate #2

$$z_{2k} = \frac{1}{\sqrt{5 a_k + 5}} (-a_k x_{k-1} + a_{k-1} x_k)$$

$$E\{z_{2k}\} = E\{ \frac{1}{\sqrt{5 a_k + 5}} (-a_k x_{k-1} + a_{k-1} x_k) \}$$

$$E\{z_{2k}\} = E\{\frac{1}{\sqrt{5 a_k + 5}} (-a_k x_{k-1})\} + E\{\frac{1}{\sqrt{5 a_k + 5}} (a_{k-1} x_k) \}$$

* Replacing $x_k = \sum_i h_i a_{k-i}$

$$E\{z_{2k}\} = E\{\frac{1}{\sqrt{5 a_k + 5}} (-a_k \sum_i h_i a_{k-1-i})\} + E\{\frac{1}{\sqrt{5 a_k + 5}} (a_{k-1} \sum_i h_i a_{k-i})\}$$

* Taking the first term;

$$E\{\frac{1}{\sqrt{5 a_k + 5}} (-a_k \sum_i h_i a_{k-1-i})\} = - \sum_i h_i \; E\{\frac{a_k}{\sqrt{5 a_k + 5}}\} E\{a_{k-1-i}\} \text{ for } i \neq -1$$

* We have $E\{a_{k-1-i}\} = E\{a_k\} = 0$, then

* The first term = 0, for $i \neq -1$.

$$E\{\frac{1}{\sqrt{5 a_k + 5}} (-a_k \sum_i h_i a_{k-1-i})\} = -h_{-1} \; E\{\frac{a_k^2}{\sqrt{5 a_k + 5}}\} \text{ for } i = -1$$

* Using assumption (2.) ,we calculate $E\{\frac{a_k^2}{\sqrt{5 a_k + 5}}\} = -1$. Then, the above expression becomes:

$$E\{\frac{1}{\sqrt{5}a_k + 5}(-a_k \sum_i h_i a_{k-1-i})\} = -h_{-1}(-1) = h_{-1} \quad \text{for } i = -1$$

* The first term = h_{-1} for all i's.

* Taking the second term;

$$E\{\frac{1}{\sqrt{5}a_k + 5}(a_{k-1} \sum_i h_i a_{k-i})\} = E\{\frac{1}{\sqrt{5}a_k + 5} \sum_i h_i a_{k-i}\} E\{a_{k-1}\} \text{ for } i \neq 1$$

* We have $E\{a_{k-1}\} = E\{a_k\} = 0$, then the second term =0, for $i \neq 1$

$$E\{\frac{1}{\sqrt{5}a_k + 5}(a_{k-1} \sum_i h_i a_{k-i})\} = h_1 E\{\frac{1}{\sqrt{5}a_k + 5}\} E\{a_{k-1}^2\} \text{ for } i = 1$$

* From (2.), we have $E\{\frac{1}{\sqrt{5}a_k + 5}\} = 0$, then the second term = 0, for i=1

* The second term = 0 for all i's.

* Then, $E\{z_{2k}\}$ = E {first term} + E {second term}, resulting from above,

$$\boxed{E\{z_{2k}\} = h_{-1} = f(\tau)}$$

A.4 Estimate #3

$$z_{3k} = a_k (x_{k-1} - h_0 a_{k-1}) \frac{1}{E\{a_k^2\}}$$

$$E\{z_{3k}\} = E\{a_k (x_{k-1} - h_0 a_{k-1}) \frac{1}{E\{a_k^2\}}\}$$

* We have $E\{\frac{1}{E\{a_k^2\}}\} = \frac{1}{E\{a_k^2\}} = \frac{1}{5}$ (E {constant} = constant):

$$E\{z_{3k}\} = \frac{1}{5} E\{a_k x_{k-1}\} - \frac{1}{5} h_0 E\{a_k a_{k-1}\}$$

$$E\{z_{3k}\} = \frac{1}{5} E\{a_k x_{k-1}\} - \frac{1}{5} h_0 E\{a_k\} E\{a_{k-1}\}$$

* We have $E\{a_k\} = E\{a_{k-1}\} = 0$, then the above expression simplifies to;

$$E\{z_{3k}\} = \frac{1}{5} E\{a_k x_{k-1}\}$$

* Replacing $x_k = \sum_i h_i a_{k-i}$,

$$E\{ z_{3k} \} = \frac{1}{5} E\{ a_k \sum_i h_i a_{k-1-i} \}$$

$$E\{ z_{3k} \} = \frac{1}{5} E\{ a_k \} E\{ \sum_i h_i a_{k-1-i} \} \quad \text{for } i \neq -1$$

* Again, given $E\{ a_k \} = 0$, the above expression = 0 for $i \neq -1$.

$$E\{ z_{3k} \} = \frac{1}{5} h_{-1} \; E\{ a_k^2 \} \quad \text{for } i = -1$$

* Replacing $E\{ a_k^2 \} = 5$ in the above expression, we get;

$$E\{ z_{3k} \} = \frac{1}{5} h_{-1} \; (5) = h_{-1}$$

$$\boxed{E\{ z_{3k} \} = h_{-1} \quad = f(\tau)}$$

A.5 Estimate #4

$$z_{4k} = \frac{1}{E\{a_k^2\}} (a_k x_{k-1} - a_{k-1} x_k) + h_1$$

$$E\{z_{4k}\} = E\{ \frac{1}{E\{a_k^2\}} (a_k x_{k-1} - a_{k-1} x_k) + h_1 \}$$

* $E\{h_1\} = h_1$ (E {constant} = constant), then the above expression becomes;

$$E\{z_{4k}\} = \frac{1}{E\{a_k^2\}} E\{a_k x_{k-1}\} - \frac{1}{E\{a_k^2\}} E\{a_{k-1} x_k\} + h_1$$

* Replacing $x_k = \sum_i h_i a_{k-i}$ in the above expression, we get;

$$E\{z_{4k}\} = \frac{1}{E\{a_k^2\}} E\{a_k \sum_i h_i a_{k-1-i}\} - \frac{1}{E\{a_k^2\}} E\{a_{k-1} \sum_i h_i a_{k-i}\} + h_1$$

* Taking the <u>first term</u>;

$$\frac{1}{E\{a_k^2\}} E\{a_k \sum_i h_i a_{k-1-i}\} = \frac{1}{E\{a_k^2\}} E\{a_k\} E\{\sum_i h_i a_{k-1-i}\} \quad \text{for } i \neq -1$$

* Since $E\{a_k\} = 0$, replacing in above, the second term = 0, for $i \neq -1$

$$\frac{1}{E\{a_k^2\}} E\{a_k \sum_i h_i a_{k-1-i}\} = \frac{1}{E\{a_k^2\}} E\{a_k^2\} h_{-1} \quad \text{for } i = -1$$

* Replacing both expressions of $E\{a_k^2\} = 5$, we get;

$$\frac{1}{E\{a_k^2\}} E\{a_k \sum_i h_i a_{k-1-i}\} = \frac{1}{5}(5) \; h_{-1} = h_{-1}$$

* The first term = h_{-1} for all i's.

* Taking the second term;

$$\frac{1}{E\{a_k^2\}} E\{a_{k-1} \sum_i h_i a_{k-i}\} = \frac{1}{E\{a_k^2\}} E\{a_{k-1}\} E\{\sum_i h_i a_{k-i}\} \quad \text{for } i \neq 1$$

* With $E\{a_{k-1}\} = 0$, the second term = 0 for $i \neq 1$

$$\frac{1}{E\{a_k^2\}} E\{a_{k-1} \sum_i h_i a_{k-i}\} = \frac{1}{E\{a_k^2\}} E\{a_{k-1}^2\} h_1$$

* Replacing $E\{a_k^2\} = E\{a_{k-1}^2\}$, then the above term simplifies to h_1

* The second term = h_1 for all i's.

* Then, $E\{z_{4k}\}$ = E {first term} - E {second term} + h_1

* Replacing we get, $E\{z_{4k}\} = h_{-1} - h_1 + h_1 = h_{-1}$

$$\boxed{E\{z_{4k}\} = h_{-1} = f(\tau)}$$

Appendix B

Derivation of the Variances

B.1 Derivation of the Variance Expression:

In this section, we derive the variance expressions for the four timing estimates using equation (4.22), repeated below for convenience:

$$\mathrm{Var}\{\, z_k \,\} = \mathrm{E}\,\{(\,\mathbf{h}^{\mathrm{T}}\,\mathbf{d}_k\,)^2\,\} + \mathrm{E}\,\{\,\mathbf{g}_k^{\mathrm{T}}\,\mathbf{Q}\,\mathbf{g}_k\,\} + \sigma^2\,\mathrm{E}\,\{\,\mathbf{g}_k^{\mathrm{T}}\,\mathbf{I}\,\mathbf{g}_k\,\} \qquad \text{(B.1)}$$

Given (4.18) $\mathbf{A}_k\,\mathbf{g}_k = \mathbf{u} + \mathbf{d}_k$, then for $f(\tau) = h_{-1}$ for m=2, (4.18) becomes:

$$\begin{bmatrix} a_k & 0 \\ a_{k-1} & a_k \\ 0 & a_{k-1} \end{bmatrix} \begin{bmatrix} g_k \\ g_k \end{bmatrix} = \begin{bmatrix} 1 & (=u_{-1}) \\ 0 & (=u_0) \\ 0 & (=u_1) \end{bmatrix} + \begin{bmatrix} d_{k-2} \\ d_{k-1} \\ d_k \end{bmatrix} \qquad \text{(B.2)}$$

Multiplying each side, we obtain:

$$\begin{bmatrix} a_k\, g_{k-1} \\ a_{k-1}\, g_{k-1} + a_k\, g_k \\ a_{k-1}\, g_k \end{bmatrix} = \begin{bmatrix} -1 + d_{k-2} \\ d_{k-1} \\ d_k \end{bmatrix} \tag{B.3}$$

The above expression yields the following equations:

$$\begin{cases} d_{k-2} = 1 + a_k\, g_{k-1} \\ d_{k-1} = a_{k-1}\, g_{k-1} + a_k\, g_k \\ d_k = a_{k-1}\, g_k \end{cases} \tag{B.4}$$

(B.4) will be used to calculate the d_k terms for each estimate from the corresponding g_k 's.

Taking the first term in (B.1) and expanding, we get:

$$\mathbf{h}^T \mathbf{d}_k = (h_{-1} \quad h_0 \quad h_1) \begin{pmatrix} d_{k-2} \\ d_{k-1} \\ d_k \end{pmatrix} = h_{-1}\, d_{k-2} + h_0\, d_{k-1} + h_1\, d_k \tag{B.5}$$

Replacing (B.4) in (B.5), we get:

$$\mathbf{h}^T \mathbf{d}_k = h_{-1}\,(1 + a_k\, g_{k-1}) + h_0\,(a_{k-1}\, g_{k-1} + a_k\, g_k) + h_1\, a_{k-1}\, g_k \tag{B.6}$$

Squaring (B.5) and taking the expected value:

$$E\{(\mathbf{h}^T \mathbf{d}_k)^2\} = E\{(h_{-1}\, d_{k-2} + h_0\, d_{k-1} + h_1\, d_k)^2\} \tag{B.7}$$

Expanding (B.7), we get:

$$\begin{aligned} E\{(\mathbf{h}^T \mathbf{d}_k)^2\} = E\{ & h_{-1}^2 d_{k-2}^2 + h_0^2 d_{k-1}^2 + h_1^2 d_k^2 + 2 h_0 h_{-1} d_{k-2} d_{k-1} + \\ & 2 h_{-1} h_1 d_{k-2} d_k + 2 h_{-1} h_0 d_k d_{k-1} \} \end{aligned} \tag{B.8}$$

Simplifying (B.8) by removing the constants from the expectation:

$$E\{(\mathbf{h}^T \mathbf{d}_k)^2\} = h_{-1}^2 E\{d_{k-2}^2\} + h_0^2 E\{d_{k-1}^2\} + h_1^2 E\{d_k^2\} + 2h_0 h_{-1} E\{d_{k-2} d_{k-1}\} + 2h_{-1} h_1 E\{d_{k-2} d_k\} + 2h_{-1} h_0 E\{d_k d_{k-1}\} \quad (B.9)$$

After determing the d_k terms for each estimate from (B.4), (B.9) will be used to calculate the first term in the variance expression in (B.1).

Taking the second term of (B.1), we expand the matrix Q from (4.15) as follows:

$$\mathbf{Q} = E\{a_k^2\} \sum_{p(\neq 1,2)} h_{i-p} h_{j-p} = m_{ij\ i=1,2\ j=1,2} \quad (B.10)$$

From (B.10), we expand the first two terms of the second term of (B.1) to obtain:

$$\mathbf{g}_k^T \mathbf{Q} = (g_{k-1} \quad g_k) E\{a_k^2\} \begin{pmatrix} \sum_{p(\neq 1,2)} h_{1-p}^2 & \sum_{p(\neq 1,2)} h_{1-p} h_{2-p} \\ \sum_{p(\neq 1,2)} h_{1-p} h_{2-p} & \sum_{p(\neq 1,2)} h_{2-p}^2 \end{pmatrix} \begin{matrix} i=1 \\ i=2 \end{matrix} \quad (B.11)$$

Multiplying the terms of (B.11), we get:

$$\mathbf{g}_k^T \mathbf{Q} = E\{a_k^2\} \left(g_{k-1} \sum_{p(\neq 1,2)} h_{1-p}^2 + g_k \sum_{p(\neq 1,2)} h_{1-p} h_{2-p} \quad g_{k-1} \sum_{p(\neq 1,2)} h_{1-p} h_{2-p} + g_k \sum_{p(\neq 1,2)} h_{2-p}^2 \right) \quad (B.12)$$

Adding the last term of the second term of (B.1) to (B.12), we get:

$$\mathbf{g}_k^T \mathbf{Q}\, \mathbf{g}_k = E\{a_k^2\} \begin{pmatrix} g_{k-1} \\ g_k \end{pmatrix} \left(g_{k-1} \sum_{p(\neq 1,2)} h_{1-p}^2 + g_k \sum_{p(\neq 1,2)} h_{1-p} h_{2-p} g_{k-1} \sum_{p(\neq 1,2)} h_{1-p} h_{2-p} + g_k \sum_{p(\neq 1,2)} h_{2-p}^2 \right) \quad (B.13)$$

Multiplying the terms in (B.13):

$$g_k^T Q\, g_k = E\{a_k^2\}\Big(g_{k-1}^2 \sum_{p(\neq 1,2)} h_{1-p}^2 + 2 g_k g_{k-1} \sum_{p(\neq 1,2)} h_{1-p} h_{2-p} + g_k^2 \sum_{p(\neq 1,2)} h_{2-p}^2 \Big) \qquad \text{(B.14)}$$

Taking the expected value of (B.14), we obtain:

$$E\{g_k^T Q g_k\} = E\{a_k^2\}\Big(E\{g_{k-1}^2\} \sum_{p(\neq 1,2)} h_{1-p}^2 + 2E\{g_k g_{k-1}\} \sum_{p(\neq 1,2)} h_{1-p} h_{2-p} + E\{g_k^2\} \sum_{p(\neq 1,2)} h_{2-p}^2\Big) \qquad \text{(B.16)}$$

(B.16) will be used to calculate the second term in the variance expression of (B.1).

Taking the third term of the variance expresion of (B.1), we expand as follows:

$$g_k^T I\, g_k = \begin{pmatrix} g_{k-1} & g_k \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} g_{k-1} \\ g_k \end{pmatrix} \qquad \text{(B.17)}$$

Multiplying the tems in (B.17);

$$g_k^T I\, g_k = g_{k-1}^2 + g_k^2 \qquad \text{(B.18)}$$

Taking the expected value of (B.18) and replacing in the last term of (B.1):

$$\sigma^2 \left(E\{g_k^T I\, g_k\} \right) = \sigma^2 \left(E\{g_{k-1}^2\} + E\{g_k^2\} \right) \qquad \text{(B.19)}$$

We will use (B.19) to calculate the last term of the variance expression of (B.1).

B.2 Variance of Estimate #1

For the estimate #1, the weighting coefficients are :

$$g_{k-1} = \frac{1}{16} a_k (a_k^2 - 5); \qquad g_k = -\frac{1}{16} a_{k-1} (a_k^2 - 5)$$

Then from (B.4), the d_k 's are :

$$\begin{cases} d_{k-2} = 1 + \frac{1}{16} a_k{}^2 \, (a_k{}^2 - 5) \\ d_{k-1} = \frac{1}{16} a_{k-1} a_k \, (a_k{}^2 - 5) - \frac{1}{16} a_{k-1} a_k \, (a_k{}^2 - 5) = 0 \\ d_k = -\frac{1}{16} a_{k-1}{}^2 \, (a_k{}^2 - 5) \end{cases}$$

From (B.9), we need to calculate $E\{ d_{k-2}{}^2 \}, E\{ d_k{}^2 \}, E\{ d_{k-2} \, d_k \}$:

$$E\{ d_{k-2}{}^2 \} = E\{(1 + \frac{1}{16} a_k{}^4 - \frac{5}{16} a_k{}^2)^2\} = \frac{25}{16}$$

$$E\{ d_k{}^2 \} = E\{\frac{1}{(16)^2} a_{k-1}{}^4 \, (a_k{}^2 - 5)^2 \} = \frac{41}{16}$$

$$E\{d_{k-2} \, d_k\} = E\{-\frac{1}{16} a_{k-1}{}^2 (a_k{}^2 - 5)\} + E\{-\frac{1}{(16)^2} a_{k-1}{}^2 a_k{}^2 \, (a_k{}^2 - 5)^2 \} = -\frac{25}{16}$$

Then, the first term in the variance is calculated:

$$E\{ (\mathbf{h}^T \, \mathbf{d}_k)^2\} = \frac{25}{16} h_{-1}{}^2 + \frac{41}{16} h_1{}^2 + \frac{25}{8} h_1 \, h_{-1}$$

For the second term of the variance, we need $E\{ g_k{}^2 \}, E\{ g_{k-1}{}^2 \}, E\{ g_{k-1} \, g_k \}$:

$$E\{ g_{k-1}{}^2 \} = \frac{1}{(16)^2} E\{a_k{}^2 (a_k{}^2 - 5)^2 \}$$

$$E\{ g_{k-1}{}^2 \} = \frac{1}{(16)^2} \{ E\{ a_k{}^6 \} - 10 E\{ a_k{}^4 \} + 25 E\{a_k{}^2 \}\} = \frac{5}{16}$$

$$E\{ g_k{}^2 \} = \frac{1}{(16)^2} E\{a_{k-1}{}^2\} \, E\{(a_k{}^2 - 5)^2 \}$$

$$E\{ g_k{}^2 \} = \frac{1}{(16)^2} E\{ a_{k-1}{}^2 \} \{ E\{ a_k{}^4 \} - 10 E\{a_k{}^2 \} + 25\} = \frac{5}{16}$$

$$E\{ g_k \, g_{k-1} \} = -\frac{1}{(16)^2} E\{a_{k-1}\} E\{a_k (a_k{}^2 - 5)^2 \} = 0$$

Replacing the above in (B.16), we get:

$$E\{ g_k^T Q g_k \} = \frac{25}{16} \{ \sum_{p(\neq 1,2)} h_{1-p}^2 + \sum_{p(\neq 1,2)} h_{2-p}^2 \}$$

Set i = 1 - p, if p ≠1,2 then i ≠ 0,-1; and j = 2 - p, if p ≠ 1,2 then j ≠ 0,+1; replacing in the above equation , we obtain:

$$E\{ g_k^T Q g_k \} = \frac{25}{16} \{ \sum_{i(\neq -1)}{}' h_i^2 + \sum_{j(\neq 1)}{}' h_j^2 \}$$

(*Note:* A (') prime on a summation denotes absence of zero term)

To calculate the last term of the variance, we use (B.19) and the terms $E\{ g_k^2 \}$, $E\{ g_{k-1}^2 \}$ as determined above:

$$\sigma^2 E\{ g_k^T I g_k \} = \sigma^2 \{E\{ g_k^2 \} + E\{ g_{k-1}^2 \}\} = \sigma^2 (\frac{5}{16} + \frac{5}{16}) = \frac{5}{8}\sigma^2 = 0.625\ \sigma^2$$

Adding the three terms, we get:

$$Var\{z_{1k}\} = \frac{25}{16} h_{-1}^2 + \frac{41}{16} h_1^2 - \frac{25}{8} h_1 h_{-1} + \frac{25}{16}\{ \sum_{i(\neq -1)}{}' h_i^2 + \sum_{j(\neq 1)}{}' h_j^2 \} + 0.625\sigma^2$$

Grouping the terms and simplifying we get:

$$Var\{z_{1k}\} = \frac{25}{16} \sum_{i(\neq -1)}{}' h_i^2 + \frac{25}{16} h_{-1}^2 + \frac{25}{16} \sum_{j(\neq 1)}{}' h_j^2 + \frac{25}{16} h_1^2 + h_1^2 - \frac{25}{8} h_1 h_{-1} + 0.625\sigma^2$$

Combining the first two terms yields $\frac{25}{16}\sum_i{}' h_i^2$ and similarly, the second two terms yield $\frac{25}{16}\sum_j{}' h_j^2$, and combining both summations, we get:

$$\boxed{\mathrm{Var}\{z_{1k}\} = \frac{25}{8}\sum_i{}' h_i^2 + h_1^2 - \frac{25}{8} h_1 h_{-1} + 0.625\sigma^2}$$

B.3 Variance of Estimate #2

For estimate #2, the weighting coefficients and d_k terms are as follows:

$$g_{k-1} = \frac{-a_k}{\sqrt{5}\,a_k + 5}; \qquad g_k = \frac{a_{k-1}}{\sqrt{5}a_{k-1} + 5}$$

$$\begin{cases} d_{k-2} = \dfrac{-a_k^2}{\sqrt{5}\,a_k + 5} + 1 \\ d_{k-1} = \dfrac{-a_k a_{k-1}}{\sqrt{5}\,a_k + 5} + \dfrac{a_k a_{k-1}}{\sqrt{5}\,a_k + 5} = 0 \\ d_k = \dfrac{a_{k-1}}{\sqrt{5}\,a_k + 5} \end{cases}$$

From (B.9), we calculate $E\{d_{k-2}^2\}, E\{d_k^2\}, E\{d_k d_{k-1}\}$

$$E\{d_{k-2}^2\} = E\left\{\frac{a_k^4}{(5a_k^2 + 10\sqrt{5}\,a_k + 25)} - 2\frac{a_k^2}{\sqrt{5}\,a_k + 5} + 1\right\} = \frac{245}{40}$$

$$E\{d_k^2\} = E\left\{\frac{a_{k-1}^4}{(a_k^2 + 2\sqrt{5}\,a_k + 25)}\right\} = \frac{205}{40}$$

$$E\{d_{k-2}\,d_k\} = E\left\{\frac{-a_k^2 a_{k-1}^2}{5a_k^2 + 10\sqrt{5}\,a_k + 25}\right\} = \frac{-165}{40}$$

Then, the first term of the variance becomes:

$$E\{(\mathbf{h}^T\ \mathbf{d}_k\ )^2\}=\frac{245}{40}h_{-1}^{\ 2}+\frac{205}{40}h_1^{\ 2}-\frac{165}{20}h_1\ h_{-1}$$

For the second term of the variance, we calculate $E\{g_k^{\ 2}\}$,$E\{g_{k-1}^{\ 2}\}$,$E\{g_{k-1}\ g_k\}$

$$E\{g_{k-1}^{\ 2}\}=E\{\frac{a_k^{\ 2}}{5a_k^{\ 2}+10\sqrt{5}\ a_k\ +25}\}=\frac{33}{40}$$

$$E\{g_k^{\ 2}\}=E\{\frac{a_{k-1}^{\ 2}}{5a_k^{\ 2}+10\sqrt{5}\ a_k\ +25}\}=\frac{25}{40}$$

$$E\{g_k\ g_{k-1}\}=-E\{a_{k-1}\}E\{\frac{a_k}{(\sqrt{5}a_k\ +5)^2}\}=0$$

Replacing in (B.16), we get;

$$E\{\mathbf{g}_k^T\mathbf{Q}\ \mathbf{g}_k\}=5\left(\frac{33}{40}\right)\sum_{p(\neq 1,2)}h_{1-p}^{\ 2}+5\left(\frac{25}{40}\right)\sum_{p(\neq 1,2)}h_{2-p}^{\ 2}$$

$$E\{\mathbf{g}_k^T\mathbf{Q}\ \mathbf{g}_k\}=\left(\frac{165}{40}\right)\sum_{i(\neq -1)}{}'\ h_i^{\ 2}+\left(\frac{125}{40}\right)\sum_{j(\neq 1)}{}'\ h_j^{\ 2}$$

For the last term in the variance, we use (B.19):

$$\sigma^2(E\{\mathbf{g}_k^T\mathbf{I}\ \mathbf{g}_k\})=\sigma^2(E\{g_{k-1}^{\ 2}\}+E\{g_k^{\ 2}\})=\sigma^2\left(\frac{33}{40}+\frac{25}{40}\right)=\frac{58}{40}\sigma^2$$

Adding the three terms, grouping and simplifying, we obtain the variance of estimate #2 as follows:

$$\mathrm{Var}\{z_{2k}\} = \frac{245}{40} h_{-1}^{\,2} + \frac{205}{40} h_1^{\,2} - \frac{165}{20} h_1 h_{-1} + \frac{165}{40} \sum_{i(\neq -1)}{}' h_i^2 + \left(\frac{125}{40}\right) \sum_{j(\neq 1)}{}' h_j^2 + \frac{58}{40}\sigma^2$$

$$\mathrm{Var}\{z_{2k}\} = \frac{165}{40} \sum_{i(\neq -1)}{}' h_i^2 + \frac{165}{40} h_{-1}^{\,2} + \frac{125}{40} \sum_{j(\neq 1)}{}' h_j^2 + \frac{125}{40} h_1^{\,2} + \frac{80}{40} h_{-1}^{\,2} + \frac{80}{40} h_1^{\,2} - \frac{165}{20} h_1 h_{-1} + \frac{58}{40}\sigma^2$$

$$\boxed{\mathrm{Var}\{z_{2k}\} = 7.25 \sum_{i}{}' h_i^2 + 2\, h_{-1}^2 + 2\, h_1^2 + 8.25 h_1 h_{-1} \quad 1.45\, \sigma^2}$$

B.4 Variance of Estimate #3

For estimate #3, the weighting coeffecients are:

$$g_{k-1} = \frac{a_k}{\mathrm{E}\{a_k^2\}}\,; \qquad g_k = 0$$

This is based on the error signal as given in equation (4.12) instead of the received signal, for this, a slightly different approach is required. We will determine the variance from its definition: $\mathrm{Var}\{z_k\} = \mathrm{E}\{z_k^{\,2}\} - \mathrm{E}^2\{z_k\}$. First we will calculate the $\mathrm{E}\{z_k^{\,2}\}$.

$$\mathrm{E}\{z_k^{\,2}\} = \mathrm{E}\{\, g_k^{\mathrm{T}} e_k\, e_k^{\mathrm{T}} g_k \,\} = \mathrm{E}\{\, g_k^{\mathrm{T}} (\mathbf{x}_k - h_0\, \mathbf{a}_k)(\mathbf{x}_k - h_0\, \mathbf{a}_k)^{\mathrm{T}} g_k \,\}$$

Expanding the above expression into matrices and vectors:

$$\mathrm{E}\{z_k^{\,2}\} = \mathrm{E}\left\{ (g_{k-1} \;\; g_k) \begin{Bmatrix} x_{k-1} - h_0 a_{k-1} \\ x_k - h_0 a_k \end{Bmatrix} (x_{k-1} - h_0 a_{k-1} \;\; x_k - h_0 a_k) \begin{pmatrix} g_{k-1} \\ g_k \end{pmatrix} \right\}$$

Multiplying and simplifying, we get:

$$E\{z_k{}^2\} = E\{\, g_k{}^T x_k x_k{}^T g_k \,\} - 2h_0\, E\{\, g_k{}^T x_k\, a_k{}^T g_k \,\} + h_0{}^2 E\{\, g_k{}^T a_k\, a_k{}^T g_k \,\}$$

The second term of the variance $E^2\{\, z_k \,\}$ will be calculated using equation (4.12).

$$E^2\{z_k\} = E^2\{g_k{}^T x_k\} - 2h_0\; E\{g_k{}^T x_k\}\, E\{g_k{}^T a_k\} + h_0{}^2\; E^2\{g_k{}^T a_k\}$$

Subtracting the second term from the first:

$$\begin{aligned} Var\{\, z_k \,\} = {} & E\{\, g_k{}^T x_k x_k{}^T g_k \,\} - E^2\{g_k{}^T x_k\} - 2h_0\, E\{\, g_k{}^T x_k\, a_k{}^T g_k \,\} + \\ & 2h_0\; E\{g_k{}^T x_k\}\, E\{g_k{}^T a_k\} + h_0{}^2\, E\{\, g_k{}^T a_k\, a_k{}^T g_k \,\} - h_0{}^2\, E^2\{g_k{}^T a_k\} \end{aligned}$$

The first and second term of the above expression give the variance expression in (B.1) and will be calculate as for the previous estimates. Then, the d_k terms are:

$$\begin{cases} d_{k-2} = \dfrac{a_k{}^2}{E\{a_k{}^2\}} - 1 \\ d_{k-1} = \dfrac{a_k\, a_{k-1}}{E\{a_k{}^2\}} \\ d_k = 0 \end{cases}$$

$$E\{\, d_{k-2}{}^2 \,\} = \frac{1}{25} E\{\, a_k{}^4 \,\} - \frac{2}{5}\, E\{a_k{}^2\} + 1 = \frac{16}{25}$$

$$E\{\, d_{k-1}{}^2 \,\} = \frac{1}{25} E\{\, a_k{}^2 \,\}\, E\{\, a_{k-1}{}^2 \,\} = 1$$

$$E\{d_{k-1}\, d_{k-2}\} = E\{a_{k-1}\}\; E\{\frac{a_k{}^3}{25} - \frac{a_k}{5}\} = 0$$

Calculating (B.9) from the above:

$$E\{(\mathbf{h}^T\ \mathbf{d}_k\ )^2\} = \frac{-20}{25}\ h_{-1}^{\,2}\ +\ h_0^{\,2}$$

Now, we calculate (B.16); for this we need $E\{\ g_{k-1}^2\ \}$. The other term $g_k = 0$..

$$E\{\ g_{k-1}^2\ \} = \frac{1}{25}\ E\{a_k^2\ \} = \frac{1}{5}$$

Then from (B.16), we get:

$$E\{\ \mathbf{g}_k^T\ \mathbf{Q}\ \mathbf{g}_k\ \} = 5\frac{1}{5}\sum_{i(\neq -1)}' h_i^2$$

Now we calculate (B.19) for the noise term:

$$\sigma^2\ E\{\ \mathbf{g}_k^T\ \mathbf{I}\ \mathbf{g}_k\ \} = \frac{1}{5}\sigma^2 = 0.2\sigma^2$$

Now we calculate the third term in the variance i.e. $-2\ h_0^{\,2}\ E\{\ \mathbf{g}_k^T\ \mathbf{x}_k\ \mathbf{a}_k^T\ \mathbf{g}_k\ \}$.

$$-2h_0^{\,2}\ E\{\mathbf{g}_k^T\mathbf{x}_k\ \mathbf{a}_k^T\mathbf{g}_k\} = -2h_0^{\,2}\ (\ g_{k-1}\quad g_k\ )\begin{pmatrix} x_{k-1} \\ x_k \end{pmatrix}(a_{k-1}\quad a_k\ )\begin{pmatrix} g_{k-1} \\ g_k \end{pmatrix}$$

We multiply the above, simplify by replacing the weighting coefficients in g_k 's, and x_k by $\sum_i h_i\, a_{k-i}$, and calculate the expectation of every simplified term:

$$-2h_0^{\,2}\ E\{\mathbf{g}_k^T\mathbf{x}_k\ \mathbf{a}_k^T\mathbf{g}_k\} = -\ 2h_0^{\,2}$$

Now we calculate the fourth term, $2h_0\ \ E\{\mathbf{g}_k^T\mathbf{x}_k\}\ E\{\mathbf{g}_k^T\ \mathbf{a}_k\}$:

$$2h_0\ E\{\mathbf{g}_k^T\mathbf{x}_k\ \ E\{\mathbf{g}_k^T\mathbf{a}_k\} = 2h_0^{\,2}E\{(g_{k-1}\quad g_k\ )\begin{pmatrix} x_{k-1} \\ x_k \end{pmatrix}\}\ E\{(g_{k-1}\quad g_k\ )\begin{pmatrix} a_{k-1} \\ a_k \end{pmatrix}\}$$

Multiplying, simplifying and replacing as in previous term, we get:

$$2h_0 \, E\{g_k^T x_k \; E\{g_k^T a_k\} = 0$$

Now we calculate the fifth term, $h_0^2 \, E\{ g_k^T a_k \; a_k^T g_k \}$:

$$h_0^2 \, E\{ g_k^T a_k \; a_k^T g_k \} = h_0^2 \, (g_{k-1} \quad g_k \;) \begin{pmatrix} a_{k-1} \\ a_k \end{pmatrix} (a_{k-1} \quad a_k \;) \begin{pmatrix} g_{k-1} \\ g_k \end{pmatrix}$$

Multiplying and simplifying:

$$h_0^2 \, E\{ g_k^T a_k \; a_k^T g_k \} = h_0^2 \; E \{ g_{k-1}^2 \, a_{k-1}^2 \} = h_0^2$$

Now we calculate the sixth term, $- h_0^2 \, E^2\{g_k^T a_k\}$:

$$- h_0^2 \, E^2 \{g_k^T a_k\} = - h_0^2 \; E^2 \{(g_{k-1} \quad g_k \;) \begin{pmatrix} a_{k-1} \\ a_k \end{pmatrix} \}$$

Multiplying and simplifying:

$$- h_0^2 \; E \{g_k^T a_k\} = - h_0^2 \; E^2 \{ g_{k-1} \; a_{k-1} \} = 0$$

Adding all the terms together, we obtain $\mathrm{Var}\{ z_{3k} \}$:

$$\mathrm{Var}\{ z_{3k} \} = \frac{16}{25} h_{-1}^2 + h_0^2 + \sum_{i(\neq -1)}' h_i^2 + 0.2 \, \sigma^2 - 2 h_0^2 + h_0^2$$

Cancelling the terms and recombining the h_{-1} term with the summation:

$$\mathrm{Var}\{ z_{3k} \} = \frac{25}{25} h_{-1}^2 + \sum_{i(\neq -1)}' h_i^2 - \frac{9}{25} h_{-1}^2 + 0.2 \, \sigma^2$$

Grouping the terms:

$$\boxed{\mathrm{Var}\{\, z_{3k} \,\} = \sum_{i(\neq -1)}{}' h_i^2 \; - 0.36 h_{-1}^2 \; + 0.2\, \sigma^2}$$

B.5 Variance of Estimate #4

This estimate uses the signal x_k and some constant offset h_1 , then (B.1) can still be applied since the expectation operator is linear; and we can find the variance using the same steps as for estimates #1 and #2, with minor manipulations. However, it proves to be easier to derive the variance of estimate #4 directly from its definition as follow:

$$g_{k-1} = - a_k \frac{1}{\mathrm{E}\{a_k^2\}}; \qquad g_k = a_{k-1} \frac{1}{\mathrm{E}\{a_k^2\}}$$

$$\mathrm{Var}\{z_{4k}\} = \mathrm{E}\{z_{4k}^2\} - \mathrm{E}^2\{z_{4k}\}$$

$$\mathrm{E}\{z_{4k}^2\} = \mathrm{E}\{(g_k^{\mathrm{T}} x_k + h_1)^2\} = \mathrm{E}\{g_k^{\mathrm{T}} x_k x_k^{\mathrm{T}} g_k\} + 2 h_1 \mathrm{E}\{g_k^{\mathrm{T}} x_k\} + h_1^2$$

$$\mathrm{E}\{z_{4k}\} = \mathrm{E}\{g_k^{\mathrm{T}} x_k\} + h_1 = h_{-1} \;\; \text{----->} \;\; \mathrm{E}\{g_k^{\mathrm{T}} x_k\} = h_{-1} - h_1$$

Replacing in $\mathrm{E}\{z_k^2\}$;

$$\mathrm{E}\{z_{4k}^2\} = \mathrm{E}\{g_k^{\mathrm{T}} x_k x_k^{\mathrm{T}} g_k\} + 2 h_1 (h_1 - h_{-1}) + h_1^2$$

$$\mathrm{E}^2\{z_{4k}\} = h_{-1}^2$$

$$\mathrm{Var}\{ z_{4k} \} = E\{ g_k^T x_k x_k^T g_k \} + 2 h_1 h_{-1} - h_1^2 - h_{-1}^2$$

Now we evaluate the term $E\{ g_k^T x_k x_k^T g_k \}$:

$$E\{ g_k^T x_k x_k^T g_k \} = E\{ g_{k-1}^2 x_{k-1}^2 + g_k^2 x_k^2 + 2 g_{k-1} g_k x_{k-1} x_k \}$$

Replacing the g_k 's with the weighting coefficients and x_k terms with $\sum_i h_i a_{k-i}$, we get:

$$E\{ g_k^T x_k x_k^T g_k \} = \frac{1}{E^2\{a_k^2\}} [E\{ a_k^2 \sum_i h_i^2 a_{k-i}^2 \} + E\{ a_{k-1}^2 \sum_j h_j^2 a_{k-j}^2 \} + 2 E\{ a_k a_{k-1} \sum_i \sum_j h_i a_{k-i} h_j a_{k-j} \}]$$

$$E\{ g_k^T x_k x_k^T g_k \} = \frac{1}{E^2\{a_k^2\}} [E\{a_k^2\} E\{ a_{k-1}^2 \} h_0^2 + E\{a_k^4\} h_{-1}^2 + E\{a_k^2\} E\{a_{k-1-i}^2\} \sum_{i(\neq -1)}' h_i^2 + E\{ a_{k-1}^2 \} E\{a_k^2\} h_0^2 + E\{ a_{k-1}^4 \} h_1^2 + E\{ a_{k-1}^2 \} E\{a_{k-j}^2\} \sum_{j(\neq 1)}' h_j^2 - 2 E\{a_k^2\} E\{a_{k-1}^2\} h_0^2 - 2 E\{a_k^2\} E\{ a_{k-1}^2 \} h_{-1} h_1]$$

$$E\{ g_k^T x_k x_k^T g_k \} = \frac{41}{25} h_{-1}^2 + \sum_{i(\neq -1)}' h_i^2 + \frac{41}{25} h_1^2 + \sum_{j(\neq 1)}' h_j^2 - 2 h_1 h_{-1}$$

Replacing in the variance expression of estimate #4:

$$\mathrm{Var}\{z_{4k}\} = \frac{41}{25} h_{-1}^2 + \sum_{i(\neq -1)}{}' h_i^2 + \frac{41}{25} h_1^2 + \sum_{j(\neq 1)}{}' h_j^2 - 2h_1 h_{-1} + 2h_1 h_{-1} - h_1^2 - h_{-1}^2$$

Rearranging and grouping the terms, we simplify as follows:

$$\mathrm{Var}\{z_{4k}\} = \frac{25}{25} h_{-1}^2 + \sum_{i(\neq -1)}{}' h_i^2 + \frac{25}{25} h_1^2 + \sum_{j(\neq 1)}{}' h_j^2 - \frac{9}{25} h_{-1}^2 - \frac{9}{25} h_1^2$$

$$\mathrm{Var}\{z_{4k}\} = 2 \sum_{i}{}' h_i^2 - \frac{9}{25} h_{-1}^2 - \frac{9}{25} h_1^2$$

As mentioned previously, the above expression can be also determined from (B.1). It was verified and similar results were found. This will be used for the noise expression:

$$\sigma^2 \; \mathrm{E}\{ g_k^{\mathrm{T}} \mathrm{I} \; g_k \} = \sigma^2 (\mathrm{E}\{ g_{k-1}^2 \} + \mathrm{E}\{ g_k^2 \})$$

$$\mathrm{E}\{ g_{k-1}^2 \} = \frac{1}{\mathrm{E}^2\{a_k^2\}} \mathrm{E}\{a_k^2\} = \frac{1}{5}$$

$$\mathrm{E}\{ g_k^2 \} = \frac{1}{\mathrm{E}^2\{a_k^2\}} \mathrm{E}\{ a_{k-1}^2 \} = \frac{1}{5}$$

$$\sigma^2 \; \mathrm{E}\{ g_k^{\mathrm{T}} \mathrm{I} \; g_k \} = \sigma^2 \left(\frac{1}{5} + \frac{1}{5}\right) = \frac{2}{5}\sigma^2 = 0.4 \; \sigma^2$$

Replacing in $\mathrm{Var}\{z_{4k}\}$:

$$\boxed{\mathrm{Var}\{z_{4k}\} = 2 \sum_{i}{}' h_i^2 - \frac{9}{25} h_{-1}^2 - \frac{9}{25} h_1^2 + 0.4 \; \sigma^2}$$

Appendix C

CAPSIM/BLOSIM Simulations

C.1 Introduction

CASPIM/BLOSIM or BLOck SIMulator/CAPture SIMulate, is a time driven signal processing simulation program suitable for digital signal processing and sampled-data communication systems. It provides great flexibility in the setup, configuration and alteration of simulations by allowing a hierachical structure of systems from basic simple building blocks to higher level groupings of these blocks. CAPSIM is the front end for BLOSIM providing the graphical interface for simulations, plotting facilities, sequential simulation analysis and easy retrieval of block-related information.

There are three basic building blocks: stars, galaxies and universes. A star corresponds to small pieces of simulation programming written in pseudo-C code which will be described later. It can have none, one or multiple inputs and/or outputs as well as user-definable parameters. Stars performing basic common signal processing operations (ex. add, multiply) are provided in the CORE (main) library. User-defined stars can be any specific operations implemented in code and stored in the user library. The creation of custom made blocks will be described in the following sections. A galaxy is a larger

building block created out of interconnected stars and other galaxies with none, one or multiple inputs/outputs/paramters. Any of the galaxy's blocks' inputs, outputs or parameters can be defined as the galaxy's inputs, outputs or parameters. The collection of stars and galaxies that form the executable simulation is called a universe. This highest level block does not have any floating inputs or outputs. All inputs should be connected to signal sources and all outputs should be connected to signal sinks. A universe is specified by its topology. The names of stars and galaxies, their respective parameters and the specification of the interconnections between them is called a topology. Topologies can be stored and reloaded. Topology files will be described in detail in the next section.

A simulation (universe) is constructed in the graphic domain by creating a block diagram of the system. This corresponds to bringing in instances of the required blocks and galaxies into the user's workingspace, interconnecting the inputs and outputs and specifying the parameters of every block. Simulation input can be derived from files or data generation blocks. After the universe has been defined graphically by the user, CAPSIM reads or "captures" the topology information in a non-graphical form which is used by CAPSIM/BLOSIM when the simulation is run. Simulation output can be plotted or stored in data files which can later be used for plotting, printing or as inputs to other simulations. The window-based user-interface is friendly with icons and pop-up menus for mouse-selectable functions and keyboard entry.

A more detailed description of the CAPSIM/BLOSIM environment is given in [[StFA88], [NoPA89] and [HaYa88]. The complete description of the simulation environment created to model the ISDN U transceiver with all the stars including their source code, galaxies and universe topologies used, is given in [HaYa89]. A detailed description of all the blocks described in chapter 6 for the simulation of the half-duplex receiver as well as all the blocks needed to incorporate the full-duplex transceiver as

decribed in chapter 3 is included in [HaYa89]. In the next sections, we will briefly describe the topology file and the pseudo-C template provided for the creation of a custom-made block (star), as well as give an example related to the ISDN transceiver simulation for each.

C.2 Topology File

The topology file is divided in two sections: the star/galaxy information and the interconnection information. For every star/galaxy in the topology, the star/galaxy information includes the type of the block (star or galaxy) star, the ID of the block, and the block's name. This is followed by a list of all of the block's parameters, numbered from zero and up, with the type and value of every parameter. A parameter can be integer, floating-point or "file" (character) type. The value of the parameter is the actual numerical or string entered by the user. The interconnection includes a list of the interconnected inputs and outputs Every interconnection is specified on one line with the keyword "connect", followed by the input's star ID, the input number, the output's star ID and the output number, as well as an optional user-specified signal name (for the signal formed by the connection).

The following is an example of the topology for the ISDN half-duplex transceiver shown in Figure 6.1, for estimate #1, for the B3 loop.

```
**************************************************************************
 est1_b3l.t
param int 2
param int 8
star 10228 linecode

param array 2  1  -1
```

```
star 15151 sum

star 15234 atq

star 15340 error

param int 0
star 17744 time_est

star 17853 loop_filt

star 19712 fork

star 19812 fork

star 20012 fork

param int 0
param int 1
param int 120
param int 150000
param int 1
star 20321 trig

param int 300000
param int 12
param int 2
star 21258 data_new

param int 0
param int 128
star 21584 adaptive_phase_ver2

star 21812 fork

param int 120
star 21981 tr_variance

param int 168
param file stb37d_8br_short.prq
```

```
param int 0
star 22363 linear_conv

param float 16
param float 0
param int 0
star 22447 resmpl

param float 5
param int 0
param int 128
star 23291 phase_quantz_ver3

param float 0
param float 6.000000e-03
star 23392 arc_ver2

param int 17
param float 6.000000e-03
param file est1_b3_snr16db.taps
star 23493 t_dfe_ver2

param int 1
param int 150000
param float 0.1
param int 1
star 23667 random_new

star 23710 add

star 23831 sink

star 23942 drain

connect 10228 0 22363 0    2B1Q_pulses
connect 10228 1 23831 0
connect 15151 0 15234 0    X_K
connect 15151 1 15340 0    X_K
connect 15151 2 17744 0    X_K
connect 15234 0 19812 0    DRX
```

```
connect 15340 0 20012 0    ERROR
connect 17744 0 21812 0    Z_K
connect 17853 0 23291 0    tf_avg
connect 19712 0 15234 1    ARC
connect 19712 1 15340 2    ARC
connect 19712 2 23291 1    ARC
connect 19812 0 15340 1    DRXC
connect 19812 1 23392 0    DRX
connect 19812 2 23493 0    DRX
connect 19812 3 17744 1    DRX
connect 20012 0 23493 1    ERROR
connect 20012 1 23392 1    ERROR
connect 20321 0 23291 2    FRAME
connect 20321 1 17853 1    FRAME
connect 21258 0 10228 0    TX_bits
connect 21584 0 23710 1    signal
connect 21812 0 17853 0    Z_K
connect 21812 1 21981 0    z_k
connect 21981 0 23942 1
connect 21981 1 23942 2
connect 22363 0 22447 0    recsig_8br
connect 22447 0 21584 0    recsig_128br
connect 23291 0 21584 1    phase_jump
connect 23291 1 23942 0
connect 23392 0 19712 0    ARC
connect 23493 0 15151 1    ISI
connect 23667 0 23710 0
connect 23710 0 15151 0    sig_noise
```

****

C.3 Creating Custom Made Blocks

The CAPSIM/BLOSIM package provides a simple structured pseudo-C template which the user fills in. Pseudo-C is a "code structure" specific to the BLOSIM/CAPSIM program, based on the C language. The source code of a star consists of nine sections: Comments, parameters, input_buffers, output_buffers, declarations, states,

initialization_code, main_code and wrapup-code. Each of these sections will be described in detail below:

1. /* COMMENT SECTION */

   This consists of any user comments identifying the star name, author, date, and functionality.

2. **Parameters**

```
type    parameter_name          /* type = int, float or file(char) */
end                             /* parameter_name is user specified */
```

3. **Input_buffers**

```
min_delay = delay value         /* minimum number of past input values*/
                                /* accessed by input buffer*/
max_delay = delay value         /* maximum number of past input values*/
                                /* accessed by input buffer */
type      buffer_name           /* type=(int or float), buffer_name is user*/
                                /* specified*/
end
```

4. **Output_buffers**

```
min_delay = delay value         /* minimum number of past output values*/
                                /* accessed by output buffer*/
max_delay = delay value         /* maximum number of past output values*/
                                /* accessed by output buffer */
type      buffer_name           /* type=(int or float), buffer_name is user*/
                                /* specified*/
end
```

5. Declarations

```
type      Variable_name        /* type=any C type variable, variable_name*/
                               /* is user-specified. These variables are */
                               /* temporary storage for a star, i.e. their */
                               /* content is lost every time the star is */
                               /* visited and exited, during the whole */
                               /* simulation run */
end
```

6. States

```
type      Variable_name        /* type=any C type variable, variable_name*/
                               /* is user-specified. These variables are */
                               /* permanent storage for a star, i.e. their */
                               /* content is saved every time the star is */
                               /* visited and exited, during the whole */
                               /* simulation run */
end
```

7. Initialization_code

User defined C code.

Ex: memory allocation, opening input/output files for read/write access, setting up arrays and variables, basic error checking. This code is executed once, the first time the star is visited during a simulation run.

end

8. main_code

User defined C code

Ex: the block's functionality is implemented in this section. This code is

executed every time the block is visited during a simulation run. This should include reading of input buffers, processing of data and output of results on output_buffers.

end

9. Wrapup_code

User defined C code

Ex: Deallocation of memory, printing of results, file closing. This code is executed once the last time the block is visited.

end

An example of a star's source code is given below for the implementation of the timing estimate generator block as defined in chapter 4. (Note: this version of time_est implements estimates #1 and #2 only)

```
/*   time_est.s   */
/*************************************************************************
Star:  TIME_EST - time estimate

Description: This block computes a timing function estimate based on
the Mueller & Muller method of general form:

      z(k) = (g0*x(k) + g1*x(k-1) + g2*x(k-2))*beta

where the g's are a function of the data symbols.

Parameters:

      method       this parameter determines the type of function
            = 0 ---> 2B1Q data, 2 taps
                  with
                  g0 = DRX(K-1)
                  g1 = -DRX(K)
```

```
                    g2 = 0
                    beta = (|DRX(k)| - 2)/4;
Inputs:

   X_K      data samples at input of data quantizer
   DRX      data symbols at output of data quantizer

Outputs:

   z_k      timing estimate


Programmer:     Mirna Hage
Date:           July, 1988
Version:        1.0
Modified:       November 17, 1988 by Mirna Hage
                - Changed DRX input buffer to type float
                February 2, 1989 by Mirna Hage
                - Added another timing estimate which implements
                the following:
                   g0 = -DRX(k-1)
                   g1 = DRX(k)
                   g2 = 0

                   beta = 1/( 5 +  (\/ 5 * DRX(k)) )
                This estimate is executed if method = 1.

*****************************************************************/
#include <math.h>

parameters

     int  method;

end

input_buffers

     delay_max = 2;
```

```
        float X_K;

        delay_max = 2;
        float DRX;

end

output_buffers

        float z_k;

end

declarations

        int i, drxi;
        float g0, g1, g2;
        float beta;

end

iniatialization_code

        if(( method != 0 ) && ( method != 1))
        {
           fprintf(stderr,"TIME_EST: Invalid method parameter \n");
           return(1);
        }

main_code

        for( i=min_avail(); i>0; i-- )
        {
                it_in(0);
                it_in(1);

                if( method ==0 )
                {
                        g0 = DRX(1);
                        g1 = -DRX(0);
```

```
                g2 = 0;

                beta = (abs((int)DRX(0)) - 2.0)/4.0;
        }
        else if( method == 1 )
        {
                g0 = -DRX(1);
                g1 = DRX(0);
                g2 = 0;
                beta = 1.0/(5.0 + ((sqrt(5.0))*DRX(0)));
        }
        it_out(0);
        z_k(0) =((g0*X_K(0)) + (g1*X_K(1)) + (g2*X_K(2)))*beta;

    }
    return(0);

end
*************************************************************************
```