

**TIMING RECOVERY FOR LOW SIGNAL TO NOISE  
RATIO RECORDING CHANNELS**

**KWEK KIAN HONG**

**NATIONAL UNIVERSITY OF SINGAPORE**

**2004**

**TIMING RECOVERY FOR LOW SIGNAL TO NOISE  
RATIO RECORDING CHANNELS**

**KWEK KIAN HONG**

(B.Eng. (Hons.), NUS)

A THESIS SUBMITTED  
FOR THE DEGREE OF MASTER OF ENGINEERING  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
NATIONAL UNIVERSITY OF SINGAPORE

2004

## **Acknowledgements**

I will like to thank the Lord my God for His love and grace. He has given me strength to overcome many difficulties. This work is dedicated to Him.

I also wish to express my gratitude to Dr. George Mathew, program manager of the Coding and Signal Processing group of the Data Storage Institute, for his guidance and teaching. He has given me many ideas and painstakingly reviewed all my drafts. His wealth of knowledge on magnetic recording has helped me in many ways.

I am also grateful to Assoc. Prof. Aleksandar Kavčić, who is the John L. Loeb Associate Professor of the Natural Sciences from Harvard University. I have benefited greatly from his contagious dynamism and the many discussions with him. He has been a great source of motivation and inspiration.

Last but not least, I would like to thank my family and friends for all their support and encouragement.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Magnetic Recording System	1
1.2	MMSE PR Equalization	3
1.3	PLL timing recovery	5
1.4	Motivation	8
1.5	Organization of the thesis	9
<b>2</b>	<b>Fast Acquisition 1a: Matching Scheme for 6T Preamble</b>	<b>10</b>
2.1	Inadequacy of variable threshold detection	10
2.2	Matching for 6T preamble	12
2.3	Probability of breakdown of matching scheme for 6T preamble	19
2.4	Improvements to the matching scheme for 6T preamble	23
2.5	Implementation issues of correction rule	27
2.6	Summary	28
<b>3</b>	<b>Fast Acquisition 1b: Matching Scheme for 8T and 10T Preambles</b>	<b>30</b>
3.1	Direct matching scheme for 8T preamble	30
3.2	Shortening the mean data collection time	33
3.3	Probability of breakdown of indirect matching scheme for 8T preamble	41
3.4	Matching schemes for 10T preamble	44
3.5	Summary	46
<b>4</b>	<b>Fast Acquisition 2: TED Output Toggling</b>	<b>48</b>
4.1	Hang-up region indicators	48
4.2	Anti-hang-up mechanism based on toggling TED outputs	58
4.3	Summary	60

<b>5</b>	<b>Joint Timing Recovery and Symbol Detection 1:</b>	<b>63</b>
	<b>Joint Timing-ISI Trellis 1</b>	
5.1	Introduction	63
5.2	Quantized timing error model	65
5.3	Joint timing-ISI trellis	67
5.4	Viterbi algorithm on the joint timing-ISI trellis	73
5.5	SOVA on the joint timing-ISI trellis and iterative decoding	76
5.6	Delivering soft decisions before the end of the trellis	80
5.7	Summary	83
<b>6</b>	<b>Joint Timing Recovery and Symbol Detection 2:</b>	<b>84</b>
	<b>Joint timing-ISI Trellis 2</b>	
6.1	Introduction	84
6.2	Timing trellis	84
6.3	Joint timing-ISI trellis	89
6.4	Forward-backward BCJR algorithm on the joint timing-ISI trellis	91
6.5	Simulation results	93
6.6	Forward-only MAP algorithm on the joint timing-ISI trellis	94
6.7	Summary	98
<b>7</b>	<b>Conclusion</b>	<b>99</b>

## Summary

Timing recovery is extremely important in digital communication systems because it deals with the determination of the optimum sampling instants in order to ensure reliable data recovery. In the first part of the thesis, we investigate timing acquisition using periodic input patterns (preamble) for partial response (PR) equalized perpendicular magnetic recording systems and in the second part of the thesis, we investigate low signal to noise ratio (SNR) timing tracking for a typical simple, yet representative, channel model.

The perpendicular magnetic recording channel is modeled using an arc tangent function and is equalized to the [1 2 2 1] PR target using a minimum mean-squared error (MMSE) equalizer. Timing recovery is done using a phase locked loop with the timing error detector (TED) being the Mueller and Müller (MM) TED. It was found through simulations that the presence of the equalizer within the timing recovery loop introduces delay resulting in aggravated hang-ups during timing acquisition. Therefore we propose two novel fast acquisition techniques to eradicate this problem.

The MM TED is driven by decisions made using threshold detection as well as the raw signal output from the equalizer. If the decisions made are wrong, then timing acquisition will not be effective i.e. it will experience hang-up or take unnecessarily long to acquire the correct sampling phase. Variable threshold detection (VTD) is a method found in the literature that modifies the detection thresholds according to knowledge about the pattern of the preamble to make the decisions, more correct. However, it was found that VTD is ineffective when the equalizer is within the timing recovery loop. Hang-ups will occur even with VTD, for various SNRs from low to high.

For the MM TED, the best possible timing acquisition, for a given set of PLL loop parameters, can be achieved when all the decisions made at the equalizer output are correct. Since periodic preambles are used and the PR target is known, we know the form the decisions at the equalizer output should assume – the number of decision levels and what level should be adjacent to what level. Upon analysis of how the noiseless sampling points are distributed with respect to the detection thresholds for various timing offsets, we discover that it is possible to decide on the ideal pattern to be sent into the MM TED

very reliably by simply observing the first  $N$  decisions made at the equalizer output, where  $N$  is the period of the preamble. The pattern identified may then be used repeatedly throughout the acquisition mode with no further threshold detections made until acquisition ends. Simulations show that such a scheme is able to completely eliminate hang-ups and steer the sampling phase to either one of the two ideal sampling phases that are closest to the initial phase offset. The reason that such a scheme will work is due to the presence of anchor points that are far away from detection thresholds and so have little likelihood of being detected wrongly. This is the first fast acquisition technique. It has also been modified to cover preambles with longer periods. Further modifications are also made to reduce the mean data collection time needed to start the scheme. This is done by noticing that ‘seeds’ may be identified and used to generate ideal patterns to be sent into the MM TED.

The second fast acquisition technique is based on the flipping of the TED outputs to eliminate undesirable phase updates. Through simulations and theoretical calculations, it is discovered that when hang-ups occur the differences between consecutive TED outputs exhibit very large values, much larger than the TED output differences during tracking and during non-hang-up acquisitions. These large values are a result of sudden changes in the direction of phase convergence and are typical during hang-ups when there is severe indecisiveness about the direction of convergence. This fact is exploited in the second fast acquisition technique that we shall call TED output toggling.

The idea is to detect the large differences using a simple threshold and then flip the sign of the TED output to agree with the general trend of convergence direction before the value is sent into the loop filter and voltage controlled oscillator. Theoretical calculations are done to determine the optimum trigger thresholds that will result in low probabilities of false trigger and false alarm.

Simulations show that both fast acquisition techniques are capable of completely eliminating hang-ups even at very low SNRs.

The second part of the thesis deals with low SNR timing tracking. This is a very important research area because powerful capacity approaching codes like Turbo and low density parity check (LDPC) codes operate at very low SNRs. Such low SNRs are not supported by conventional decision-directed timing recovery schemes because they

completely ignore the error control coding. As a result, cycle slips during tracking are rampant and they greatly erode the exceptional coding gains provided by the codes. We solve this problem by designing a novel joint timing intersymbol interference (ISI) trellis and running a soft output Viterbi algorithm (SOVA) on the trellis to combine timing tracking and symbol estimation. Soft decisions may be delivered to iterative decoders and *a priori* information may be fed back and used by the SOVA detector to refine detection. Simulations show that such a setup is able to eradicate cycle slips during tracking.

We also explored a second kind of joint timing-ISI trellis that is found in the literature and which uses the BCJR (Bahl, Cocke, Jelinek, Raviv) algorithm for optimum joint timing recovery and symbol detection. Simulations show that cycle slips may be corrected after a few iterations where extrinsic information is exchanged between the detector and a soft-in soft-out (SISO) decoder. We also present a way of doing the BCJR algorithm in a forward-only manner so that there is no need for a backward recursion.

The thesis is organized as follows. Chapter 1 gives a brief introduction to the perpendicular magnetic recording system and phase locked loop based decision-directed timing recovery, and motivates and summarizes the work in the thesis. Chapter 2 describes the first fast acquisition technique that is based on matching of preamble patterns. Chapter 3 describes an extension of the first technique to cover preambles with longer periods. A second fast acquisition technique based on toggling of the TED outputs is described in Chapter 4. In Chapter 5 we describe a novel joint timing recovery and equalization scheme that uses the soft output Viterbi algorithm (SOVA) to achieve joint timing recovery and symbol detection. In Chapter 6, we present a second joint timing recovery and equalization scheme and formulate a forward-only maximum *a posteriori* probability (MAP) algorithm for joint timing recovery and symbol estimation. Finally, the thesis is concluded in Chapter 7.



## List of Symbols and Abbreviations

$\alpha, \rho$	phase-locked loop gain parameters
$\Delta\tau$	TED output
$\tau$	sampling phase error (unnormalized)
$\rho(\tau)$	timing function
$a_n, a_k$	input bits
$D_u$	user (recording) density
$h_s(t)$	step response of recording channel
$h(t)$	pulse/dibit response of recording channel
$L$	over-sampling factor
$LLR(\cdot)$	log-likelihood ratio
$p_k$	coefficients of partial response target
$p(\cdot)$	probability density function
$P(X)$	probability of $X$
$T_{50}$	transition width of step response corresponding to 50% of amplitude
$T$	user bit duration
$w_k$	coefficients of MMSE equalizer
$Q$	number of timing quantization levels
AWGN	additive white Gaussian noise
BCJR	Bahl, Cocke, Jelinek, Raviv
BER	bit error rate
ECC	error-control coding
FIR	finite impulse response
ISI	intersymbol interference
LDPC	low density parity check code
LF	loop filter
LPF	low pass filter

MAP	maximum <i>a posteriori</i>
ML	maximum likelihood
MM	Mueller and Müller
MMSE	minimum mean squared error
PLL	phase locked loop
PR	partial response
RLL	run length limited
SISO	soft-in soft-out
SNR	signal to noise ratio
SOVA	soft output Viterbi algorithm
TED	timing error detector
VCO	voltage controlled oscillator
VTD	variable threshold detection

## List of Figures

1-1	Block schematic of a magnetic recording system.	2
1-2	Step response of perpendicular magnetic recording channel ( $A = 1$ ).	2
1-3	Block diagram of MMSE PR equalization.	3
1-4	Discrete-time model of recording channel with equalizer. The tap spacing of $h_k$ and $f_k$ is $T/L$ , i.e., $h_k = h(t)$ at $t = kT/L$ whereas the tap spacing of $w_k$ is $T$ . The bandwidth of the discrete-time white noise $\pi_m$ is $L/T$ Hz. The cut-off frequency of the LPF is $0.475/T$ Hz.	4
1-5	Block schematic of the timing recovery loop (or, phase locked loop (PLL)).	5
1-6	Implementation of timing recovery.	7
2-1	Threshold modifications for $6T$ preamble for $\theta = 0.7$ .	11
2-2	Phase convergence for $6T$ preamble without VTD (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.	14
2-3	Phase convergence for $6T$ preamble with VTD (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.	15
2-4	Noiseless output with ideal and $0.5T$ lagging phases for $6T$ preamble and target [1 2 2 1].	16
2-5	Contention between Sequence 1 and Sequence 3.	17
2-6	Phase convergence for $6T$ preamble with matching scheme (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.	18
2-7	Noise path through the low-pass filter (LPF), the down-sampler and the equalizer.	19

2-8	Probability of the $6T$ preamble matching scheme failing for various SNR.	23
2-9	Phase convergence for $6T$ preamble with matching scheme (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; SNR = 35 dB; frequency offset = 0.1%.	24
2-10	Noiseless output with ideal and $0.6T$ lagging phases for $6T$ preamble and target [1 2 2 1].	25
2-11	Noiseless output with ideal and $0.4T$ lagging phases for $6T$ preamble and target [1 2 2 1].	25
2-12	Phase convergence for $6T$ preamble with matching scheme and correction (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; SNR = 35 dB; frequency offset = 0.1%.	27
2-13	Flow chart summarizing the $6T$ matching scheme correction rule.	29
3-1	Noiseless output of the equalizer with ideal sampling phase for $8T$ preamble and target [1 2 2 1].	30
3-2	Noiseless output of the equalizer with sampling phases of $\pm 0.5T$ for $8T$ preamble and target [1 2 2 1].	31
3-3	Sequence 8 of Table 3-3 arranged in a circle.	33
3-4	Phase convergence for $8T$ preamble (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.	34
3-5	Flow chart summarizing the process of generating the TED reference sequence for $8T$ preamble.	39
3-6	Distribution of the number of samples that need to be collected before commencing phase updates. ( $8T$ preamble).	40
3-7	Phase convergence for $8T$ preamble with indirect matching scheme (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.	41
3-8	Phase convergence for $8T$ preamble with indirect matching scheme (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.	43

3-9	Noiseless output with ideal sampling phase for $10T$ preamble and target [1 2 2 1].	44
3-10	Noiseless output with sampling phases of $\pm 0.5T$ for $10T$ preamble and target [1 2 2 1].	45
3-11	Phase convergence for $10T$ preamble (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1.	47
3-12	Distribution of the number of samples that need to be collected before commencing phase updates ( $10T$ preamble).	45
4-1	A typical timing function whereby the hang-up regions occur around midway between ideal sampling instants.	48
4-2	Probability of failed detection of hang-up indicator and probability of false alarm; [1 2 2 1] target; $6T$ preamble; SNR = 30 dB.	56
4-3	Variation of optimum trigger threshold with SNR; [1 2 2 1] target; $6T$ preamble.	57
4-4	Typical phase convergence and TED output difference plots in the presence of hang-up and with no anti-hang-up mechanism; $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%; initial phase offset = $0.5T$ ; SNR = 30 dB.	58
4-5	TED outputs $(\Delta\tau_n, \Delta\tau_{n-1})$ that will result in triggering of anti-hang-up mechanism.	59
4-6	Undesirable phase updates associated with the flipping rule of equation (4-7).	60
4-7	Typical TED outputs for $6T$ preamble; $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%, phase offset = $0.5T$ .	61
4-8	Phase convergence for $6T$ preamble (80 runs); $\alpha = 2.0 \times 10^{-3}$ ; $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.	62
5-1	(a) Conventional approach to timing recovery, symbol detection and error-control coding. (b) Iterative timing recovery with joint timing recovery and symbol detection.	64

5-2	Trellis for the timing error $\tau_k$ modeled as a Markov chain of memory one.	66
5-3	A pulse with an ISI length of two that spans three symbol intervals.	67
5-4	Standard ISI trellis for the pulse of Fig. 5-3 with imperfect timing	67
5-5	An example of a realization of the process $\tau_k$ for some given input bits and $Q=5$ . Sampling instants are marked on the abscissa by bullets. Resultant noiseless waveform is drawn using thick line.	68
5-6	First two trellis sections of joint timing-ISI trellis with $Q=5$ .	70
5-7	Compact form of joint timing-ISI trellis with $Q=5$ . (Five trellis sections shown with the sampled realization of Fig. 5-5 marked by the thick path).	72
5-8	Typical phase error tracking; $\sigma_w = 0.02T$ ; $Q=5$ ; $\delta = 0.005$ ; SNR = 8 dB.	75
5-9	Bit error probability for implementing the Viterbi algorithm on the joint timing-ISI trellis of Fig. 5-6.	76
5-10	Three paths converging into a single node. (Survivor path marked by thick line).	77
5-11	SOVA based joint timing-ISI iterative receiver with 5 decoders.	80
5-12	Cycle-slip correction by the receiver of Fig. 5-11. $\sigma_w = 0.02T$ ; $Q=5$ ; $\delta = 0.005$ ; SNR = 2 dB; rate $\frac{1}{2}$ Turbo code.	81
5-13	Tracking plots of SOVA joint timing-ISI detector with early decisions. $\sigma_w = 0.02T$ ; $Q=5$ ; $\delta = 0.005$ ; SNR = 2 dB; rate $\frac{1}{2}$ Turbo code.	82
6-1	Partitioning of time axis ( $Q=5$ ).	85
6-2	One section of the timing trellis.	86

6-3	Possible sampling sequences to be considered when computing $P(1_1   1_1)$ . $Q = 5$ is used and sampling points are marked by bullets.	87
6-4	An example of a path through the timing trellis for a given sampling realization.	89
6-5	One section of the joint timing-ISI trellis.	90
6-6	BCJR joint timing-ISI iterative receiver.	93
6-7	Tracking plots of BCJR joint timing-ISI detector. $\sigma_w = 0.02T$ ; $Q = 5$ ; $\delta = 0.005$ ; SNR = 2 dB; rate $\frac{1}{2}$ Turbo code.	94

## List of Tables

2-1	Detected periodic sequence sent into TED and final phase destination.	13
2-2	List of all possible and likely detected sequences when the anchor labeled '3' in Fig. 2-4 is 'blown' by noise.	21
3-1	Cyclic shifts of detected output samples when sampling phase is ideal, noise is absent and detection thresholds are set at ' $\pm 2$ '.	32
3-2	Probability of occurrence of sequences that will lead to departure from an initially ideal sampling phase.	42
4-1	Equalizer outputs, threshold decisions and MM TED outputs when sampling phase is ideal for $6T$ preamble and [1 2 2 1] target.	49
4-2	Equalizer outputs, threshold decisions and MM TED outputs when phase offset is $+0.5T$ for $6T$ preamble and [1 2 2 1] target.	50
4-3	Differences of consecutive TED outputs for ideal phase and $+0.5T$ phase offset.	50
4-4	TED output differences for various combinations of threshold decisions when phase offset is $+0.5T$ .	51
4-5	TED output differences with large means and their means.	52
4-6	Approximations of the term $P( \Delta\tau_n - \Delta\tau_{n-1} _k < \zeta   HU_k)$ .	54
4-7	Relative probability of occurrence of TED output differences that have large means (SNR = 30 dB).	54
5-1	ISI state transitions corresponding to given timing state transitions for the $k^{\text{th}}$ to $(k+1)^{\text{th}}$ sampling instants.	69
5-2	Rules for assigning bits to a branch.	71
5-3	Number of errors obtained when implementing the SOVA joint timing-ISI detector for 200 blocks of 1000 data bits. ( $\sigma_w = 0.02T$ ; $Q = 5$ ; $\delta = 0.005$ ; rate 1/2 Turbo code).	83



6-1	Transition probabilities for the timing trellis of Fig. 6-2.	86
6-2	Derivations of $P(1_1   1_1)$ , $P(1_2   1_1)$ and $P(2   1_1)$ .	88
6-3	Derivations of $P(1_1   1_2)$ , $P(1_2   1_2)$ , $P(1_3   1_2)$ and $P(2   1_2)$ .	88
6-4	Derivations of $P(1_1   0)$ and $P(2   0)$ .	89

# Chapter 1

## Introduction

In this chapter, we first provide a brief introduction to the perpendicular magnetic recording system. The channel step response is modeled using an arc tangent function as described in [1]. We also present minimum mean squared error (MMSE) partial response (PR) equalization, which is used to shape the channel response to a specified PR target. This is followed by description of the second-order phase locked loop (PLL) used for phase and frequency acquisitions and tracking. Finally the motivation and organization of the thesis are given.

### 1.1 Magnetic Recording System

The block schematic of a general magnetic recording system is shown in Fig. 1-1. User data bits refer to the raw information bits in binary format  $\{0, 1\}$  that are to be stored on the magnetic storage medium. This binary data sequence is first sent through an error correction (ECC) modulation encoder to add error detection and correction capability to the data which will be used by the receiver to detect and correct errors. The encoded data is then passed through a run length limited (RLL) encoder to match the data to the channel characteristics and to help in timing recovery [2]. Finally the data is fed into the write circuits which will generate the electric currents used to drive the write head. The write head produces magnetic flux that will magnetize the storage medium in directions corresponding to the encoded data bits. During readback, the read head performs the opposite, flux-to-voltage conversion.

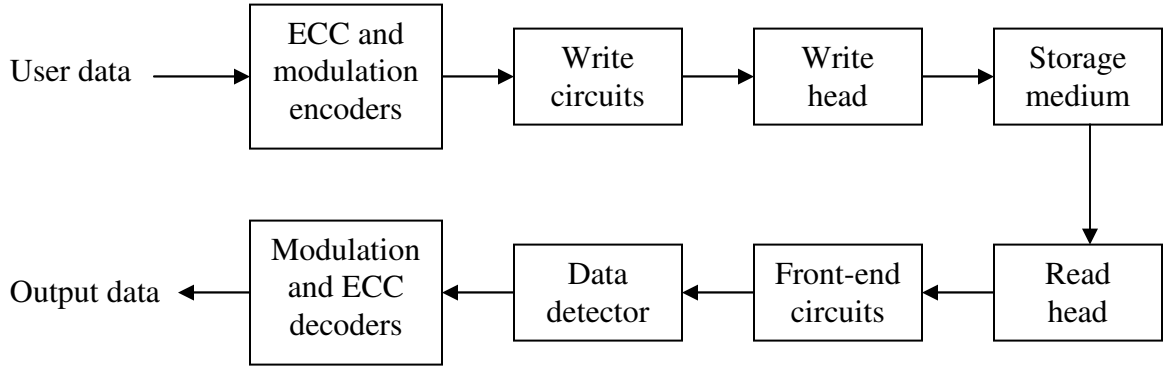


Fig. 1-1. Block schematic of a magnetic recording system.

The magnetic recording channel consisting of the cascade of the write circuits, write head, storage medium and read head can be modeled as a linear system with additive white Gaussian noise (AWGN) at the output [3]. In this project, we model the step response of this channel as [1]

$$h_s(t) = \frac{A}{\pi} \arctan\left(\frac{2t}{T_{50}}\right) \quad (1-1)$$

which corresponds to a double-layered perpendicular recording medium with a magnetoresistive read head. In (1-1),  $A$  is the signal amplitude of  $h_s(t)$  and  $T_{50}$  is the width of  $h_s(t)$  at the transition corresponding to  $0.5A$ . Fig. 1-2 illustrates the relationship between  $A$  and  $T_{50}$ .

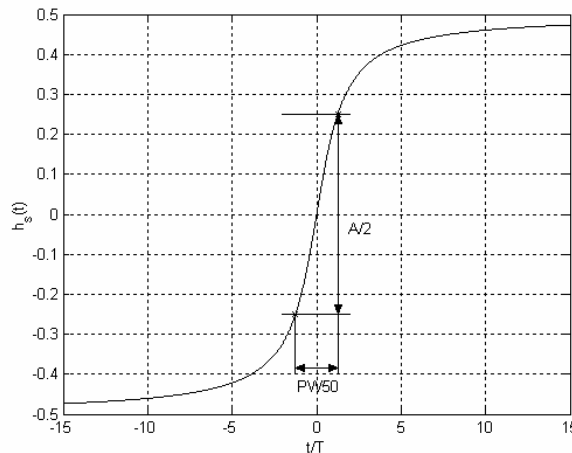


Fig. 1-2. Step response of perpendicular magnetic recording channel ( $A = 1$ ).

The value of  $T_{50}$  is dependent on the characteristics of the magnetic medium, the head, and the distance of the head to the medium. The normalized linear density (also known as user density) is defined as  $D_u = \frac{T_{50}}{T}$ , where  $T$  refers to the user bit duration. The bit response is defined as  $h(t) = h_s(t) - h_s(t-T)$  and it corresponds to the waveform obtained when the input is an isolated bit '+1'.

## 1.2 MMSE PR Equalization

As the user density increases, the duration of the channel response  $h(t)$  increases, thereby increasing the inter-symbol interference (ISI). One way to overcome the severe ISI present in high density recording systems is partial response (PR) equalization. This is the technique adopted in the first part of this project (Chapters 2 to 4).

Fig. 1-3 shows the schematic for designing an equalizer for shaping the channel response  $h(t)$  to the discrete time ( $T$ -spaced) partial response target  $p_k$ . The noise at the channel output is assumed to be AWGN. The MMSE approach is used to design the equalizer [4]. The optimum equalizer  $w_k$  minimizes the mean squared value of the error shown in Fig. 1-3. The error consists of two components: i) the mismatch between the sampled equalized channel and the PR target  $p_k$ , and ii) the noise filtered by the equalizer. By minimizing this error according to the MMSE criterion, we are simultaneously suppressing the two components of the error.

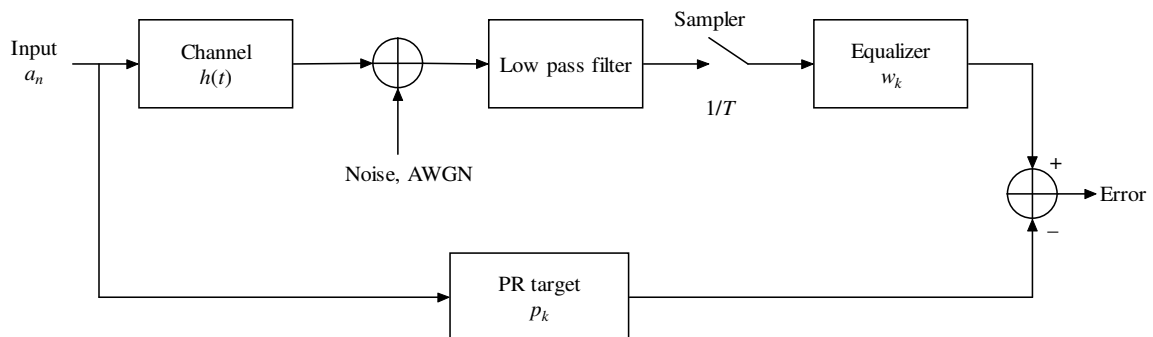


Fig. 1-3. Block diagram of MMSE PR equalization.

All the simulation studies in Chapters 2 to 4 assume the discrete-time channel model and equalizer shown in Fig. 1-4, where  $L$  refers to the over-sampling factor and LPF is a fourth-order Butterworth low pass filter used to prevent signal aliasing during down-sampling.

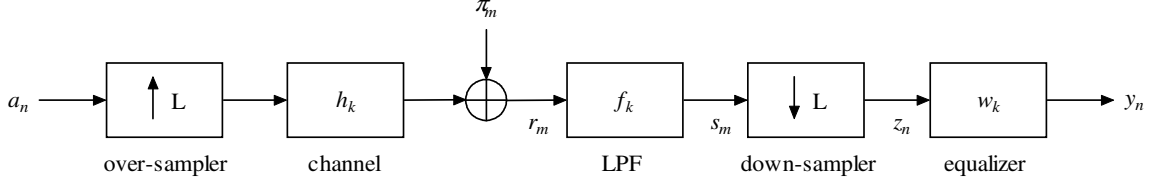


Fig. 1-4. Discrete-time model of recording channel with equalizer. The tap spacing of  $h_k$  and  $f_k$  is  $T/L$ , i.e.,  $h_k = h(t)$  at  $t = kT/L$  whereas the tap spacing of  $w_k$  is  $T$ . The bandwidth of the discrete-time white noise  $\pi_n$  is  $L/T$  Hz. The cut-off frequency of the LPF is  $0.475/T$  Hz.

Throughout Chapters 2 to 4, the user density  $D_u = 2.5$  is used for constructing the channel response  $h_k$ . The variance  $\sigma_\pi^2$  of the noise  $\pi_n$  is determined according to the signal-to-noise ratio (SNR) defined as

$$\text{SNR} = 10 \log_{10} \left( \frac{V_{op}^2}{\sigma_u^2} \right), \quad \sigma_\pi^2 = L \sigma_u^2 \quad (1-2)$$

where  $V_{op} = 2A$  is the amplitude of the isolated transition response of the channel,  $\sigma_u^2$  is the noise variance in the user bandwidth  $\frac{1}{T}$  and  $L = 2$  is the over-sampling factor.

Because the channel response  $h_k$  contains a d.c. component, PR targets of the form  $(1 + D)^n$  are appropriate, where  $D$  is the 1-bit delay operator and  $n$  is a positive integer. Other more sophisticated targets with non-integer coefficients, known as generalized partial response (GPR) targets, may be designed to yield better bit error rate (BER) performance [5][6]. However, in this thesis, the [1 2 2 1] PR target is used because it yields good BER performance while at the same time allows easy design of acquisition schemes by virtue of its integer coefficients.

### 1.3 PLL timing recovery

The block schematic of a typical second order PLL is shown in Fig. 1-5. It consists of three blocks: timing error detector (TED), loop filter (LF) and voltage controlled oscillator (VCO). The TED generates an indication of the error between the ideal sampling phase and the locally available phase from the VCO output. This is then passed into the LF which suppresses the noise components. Finally the VCO integrates the LF output to produce the reference phase of the sampling clock at the receiver.

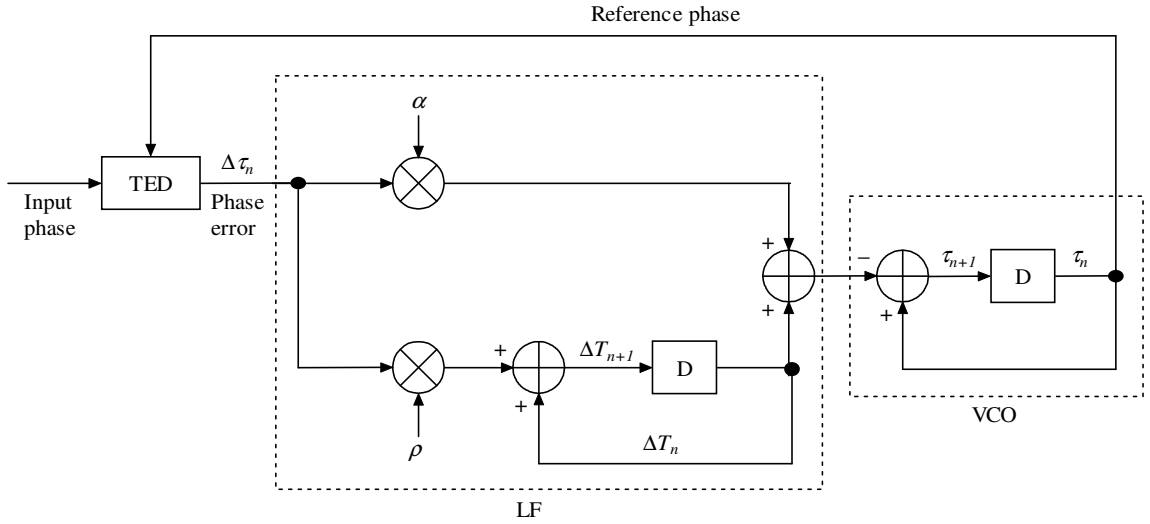


Fig. 1-5. Block schematic of the timing recovery loop (or, phase locked loop (PLL)).

For a second order PLL, the phase update equations are given by [30]

$$\tau_{n+1} = \tau_n - \alpha \Delta \tau_n - \Delta T_n \quad (1-3)$$

$$\Delta T_{n+1} = \Delta T_n + \rho \Delta \tau_n \quad (1-4)$$

where  $\Delta T_n$  represents compensation for offset between the rate of the signal received and the frequency of sampling, and  $\alpha$  and  $\rho$  are loop gain parameters. Timing phase updates are carried out with the use of timing error detectors (TEDs), which provide indication of the sampling phase error  $\tau$  (normalized with respect to  $T$ ). The TED output  $\Delta \tau_n$  can be written as

$$\Delta \tau_n = \rho(\tau) + u_n \quad (1-5)$$

where  $\rho(\tau)$  is called the timing function and  $u_n$  is a random disturbance. The timing function, which contains the desired information on the timing phase error  $\tau$ , is the average value of the TED output.

There are several timing error detection schemes present in the literature. Kobayashi [7] proposed a method of adjusting the timing phase of an equalized data transmission system to minimize the mean-squared-error, MSE, at the output of the equalizer. Qureshi [8] showed that for a partial-response system, this scheme could be reduced to a form which may be easily implemented using digital circuits. The idea of MMSE timing recovery is to obtain the derivative of the expectation of the squared error at the equalizer output with respect to the timing phase. With this gradient, one can adjust the phase offset in the direction opposite to the gradient so that ultimately, the phase will converge to the 'ideal' where the MSE is minimum.

Another type of TED is based on band edge maximization. Lyon [9] concluded that under conditions likely to be encountered in practical channels, the optimum sampling instants are accurately approximated by the timing phase which maximizes the spectrum of the sampled received signal at the band edge frequency of  $\frac{1}{2T}$ . The idea is to make the spectrum approach the ideal, which is rectangular in shape, so that there will be augmentation of aliased components. Godard [10] developed a digital timing recovery scheme to maximize the energy of the samples at the input to the detector. Farhang [11] proposed a simpler scheme based on the same concept where the TED is very simple to implement and features very fast start-up.

Mueller and Müller (MM) [12] developed a decision-directed symbol rate timing recovery scheme, which formed the basis for a class of symbol rate timing recovery techniques. The technique is based on the concept of the timing function  $\rho(\tau)$  and it is aimed at canceling ISI due to timing errors. The MM timing function is derived from a linear combination of samples of the channel's impulse response. The resulting TED is simple and easy to implement.

The system set-up used for timing recovery studies in Chapters 2 to 4 is shown in Fig. 1-6. The TED used will be the MM TED. The output from the MM TED is given by

$$\Delta \tau_n = -y_n(\tau) \hat{x}_{n-1} + y_{n-1}(\tau) \hat{x}_n \quad (1-6)$$

where  $y_n(\tau)$  is the equalizer output at sampling instant  $n$  and  $\hat{x}_n$  is the decision at sampling instant  $n$ . That is, if  $y(t)$  represents the continuous-time signal equivalent at the equalizer output, then

$$y_n(\tau) = y((n + \tau)T),$$

$$y_n(0) = y(nT) = x_n + \vartheta_n,$$

where the ideal sampling instants are given by  $nT$  for integer  $n$ ,  $x_n$  is the ideal value at the equalizer output according to the PR target chosen, and  $\vartheta_n$  is the sum of noise and residual ISI. Thus  $\hat{x}_n$  used in (1-6) is an estimate of  $x_n$ .

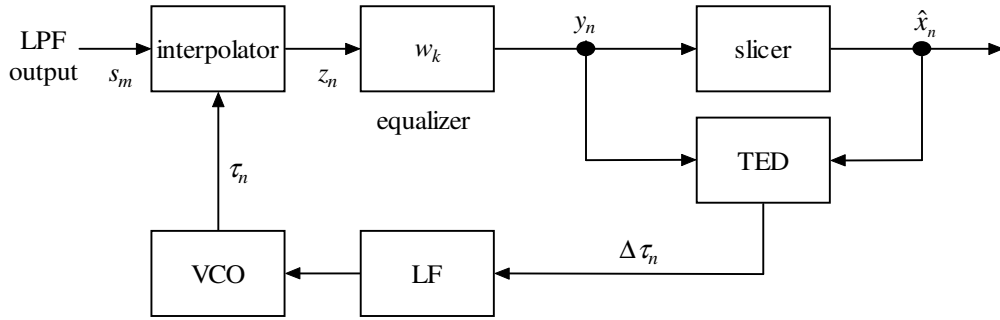


Fig. 1-6. Implementation of timing recovery.

It is a common practice to use known periodic patterns, known as preambles, during timing acquisition. This is because random data generally yields slower and less predictable acquisition behavior than periodic data [30]. The knowledge of the data is then exploited to obtain reliable decisions  $\hat{x}_n$ . Typical preamble patterns used in timing acquisition are of periods  $4T$ ,  $6T$ ,  $8T$  and  $10T$ :

$4T$  pattern:  $\{ \dots +1, +1, -1, -1, +1, +1, -1, -1, +1, +1, -1, -1, +1, +1, -1, \dots \}$ ,

$6T$  pattern:  $\{ \dots +1, +1, +1, -1, -1, -1, +1, +1, +1, -1, -1, -1, +1, +1, +1, \dots \}$ ,

$8T$  pattern:  $\{ \dots +1, +1, +1, +1, -1, -1, -1, -1, +1, +1, +1, +1, -1, -1, -1, \dots \}$ ,

$10T$  pattern:  $\{ \dots +1, +1, +1, +1, +1, -1, -1, -1, -1, -1, +1, +1, +1, +1, +1, \dots \}$ .



## 1.4 Motivation

As can be seen from Fig. 1-6, the equalizer is embedded within the timing recovery loop. Even though the equalizer length is chosen to be short (10 taps), the delay through it could still aggravate hang-ups during timing acquisition. (Hang-up is the phenomenon whereby the timing recovery loop dwells for a long time around a non-ideal timing phase). Therefore in fast acquisition, there is a need to address such hang-up problems. Variable threshold detection was proposed in [13] as a method for eliminating hang-ups for the PR4 channel and  $4T$  preamble. However it is not very effective when the equalizer is within the timing loop. The fast Fourier transform (FFT) may be used to obtain an estimate of the initial phase offset [19] so that the phase could be steered away from the hang-up regions. However, it is not attractive due to its complexity. Digital zero phase start (DZPS) [20] and digital zero frequency/phase start (DZF/PS) [21] have also been proposed as simpler alternatives to the FFT approach. In the subsequent three chapters, we describe two novel and simple fast acquisition techniques that are able to completely eliminate hang-ups reliably, even at very low SNRs. This constitutes the first part of the thesis.

With the advent of powerful capacity approaching error correcting codes like Turbo [15] and low density parity check (LDPC) codes [27][28] that operate at very low SNRs, the conventional decision-directed timing recovery scheme outlined in Section 1.3 will fail completely. This is because the decisions driving the timing recovery scheme are made independent of the error-control coding. Thus, at low SNRs, these decisions will be fraught with errors and a consequence of using them for a decision-directed timing recovery scheme is that cycle slips, or loss of lock, will occur very frequently [25]. Whenever cycle slips occur, the corresponding sector must be read again. One approach to tackle the problem is to use the frequency offset feedforward timing recovery method [22] that will result in significantly smaller residual timing jitter and thus reduce the loss of lock rate. Building on the piecewise linear phase drift model of [22], a double PLL approach [23] suitable for high data rate systems has also been proposed. However, these schemes will not completely eliminate cycle slips and thus there will always be a need to reread some sectors. Thus they can only be considered as partial solutions to the problem.

A second approach to solving low SNR timing tracking is to make use of extrinsic information provided by the decoders to improve detection, offline. Such an approach, which we shall call iterative timing recovery, will thus not require the rereading of slipped sectors. In [24], a soft timing error detector is used to drive a PLL which feeds better timing estimates to a re-interpolator used to refine the samples received. Simulations show that cycle slips may be corrected after several iterations. A cycle slip detector was used in [26] to reduce the number of iterations required. In [17], a special trellis was constructed to allow joint timing and symbol estimation using a BCJR-like [18] algorithm. Again cycle slips may be corrected after iterating between the detector and the decoder. For the second part of the thesis, we propose an iterative timing recovery scheme based on the soft output Viterbi algorithm (SOVA) [14] and modify the algorithm of [17] to eliminate the need for backward recursions.

## **1.5 Organization of the thesis**

Chapter 2 describes the first fast acquisition technique that is based on matching of preamble patterns. Chapter 3 describes an extension of the first technique to cover preambles with longer periods. A second fast acquisition technique based on toggling of the TED outputs is described in Chapter 4. In Chapter 5 we describe a novel joint timing recovery and equalization scheme that uses the soft output viterbi algorithm (SOVA) [14] to achieve joint timing recovery and symbol detection. In Chapter 6, we present a second joint timing recovery and equalization scheme and formulate a forward-only maximum *a posteriori* probability (MAP) algorithm for joint timing recovery and symbol estimation. Finally, the thesis is concluded in Chapter 7.

## Chapter 2

### Fast Acquisition 1a: Matching Scheme for 6T Preamble

In this chapter we present a novel fast timing acquisition scheme based on matching the equalizer output sequence to the output corresponding to an ideal-phase preamble, which will result in the elimination of hang-ups. This scheme is based on observing the first set of outputs corresponding to one period of the input preamble and then deciding on the best decisions to be sent into the MM TED. The scheme is designed for the 6T preamble with the channel equalized to the [1 2 2 1] PR target. Theoretical analyses are also provided on the probability of breakdown of such a scheme. Fig. 1-6 is the system setup used in this chapter.

#### 2.1 Inadequacy of variable threshold detection

The variable threshold detection (VTD) scheme is described in [13] to eliminate hang-ups during timing acquisition. The general idea is to modify the slicer thresholds based on knowledge about the periodicity of the preamble and the knowledge of the PR target in order to reduce the likelihood of wrong decisions at the slicer output. For example, if the 6T preamble is sent into the [1 2 2 1] equalized channel, the ideal channel output is the sequence

$$\{x_n\} = \{ \dots +4, 0, -4, -4, 0, +4, +4, 0, -4, -4 \dots \}. \quad (2-1)$$

Previous work had been done on a VTD scheme for this channel whereby threshold modification is accomplished through the use of two independent threshold modifiers given by the equations [29]

$$\ell_1 = \theta \operatorname{sgn}(\hat{x}_{n-1} \hat{x}_{n-2}), \quad \theta > 0, \quad (2-2)$$

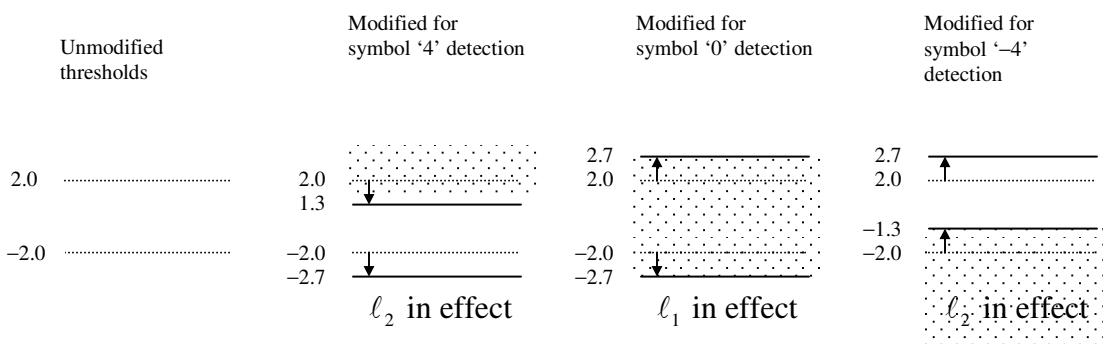
$$\ell_2 = \theta \operatorname{sgn}(\hat{x}_{n-3}), \quad \theta > 0. \quad (2-3)$$

However, in that work, the equalizer is outside the timing recovery loop.

The rules for making symbol decisions are given by

$$\hat{x}_n = \begin{cases} 4 & \text{if } y_n(\tau) \geq 2 + \ell_1 + \ell_2 \\ 0 & \text{if } -2 - \ell_1 + \ell_2 \leq y_n(\tau) < 2 + \ell_1 + \ell_2 \\ -4 & \text{if } y_n(\tau) < -2 - \ell_1 + \ell_2. \end{cases} \quad (2-4)$$

Fig. 2-1 shows the pictorial illustration of these modifications for  $\theta = 0.7$ .



Legend:

- Modified decision region of interest
- Modified detection threshold
- Original detection threshold

Fig. 2-1. Threshold modifications for  $6T$  preamble for  $\theta = 0.7$ .

In order to examine the effects of VTD, simulations are done assuming a linear phase drift model corresponding to a constant frequency offset of 0.1% of the bit rate. The loop parameters used are  $\alpha = 2.0 \times 10^{-3}$  and  $\rho = 0.5 \times 10^{-5}$ . They are chosen arbitrarily and are checked and refined through simulations to result in satisfactory timing acquisition performance. We will be using these loop parameters in all simulations on timing acquisition, regardless of whether the equalizer is within the timing loop or outside the timing loop. We admit that the acquisition performance for each system might be improved if the parameters are optimized specifically for that system. However, these loop parameters are sufficient to illustrate our point about the hang-up problem.

Fig. 2-2 shows phase plots of timing acquisition using  $6T$  preamble without any anti-hang-up schemes. The decisions are made using simple threshold detection (i.e.  $\ell_1 = 0 = \ell_2$  in (2-4)) and fed directly into the MM TED. It is amply clear from the figure

that hang-up is a serious problem and it is made worse when the equalizer is placed within the timing recovery loop. This is because the equalizer introduces additional delay into the timing loop, causing phase estimations to lag the actual timing fluctuations even more, thus aggravating the hang-up problem.

Fig. 2-3 shows the phase plots when the VTD scheme described by (2-2) to (2-4) are implemented. The left and right columns show the phase plots when the equalizer is outside and inside of the timing loop respectively. By comparing the left and right columns, we can see that VTD is ineffective in eliminating hang-up when the equalizer is embedded within the timing loop (notice that in the left column, graphs do not stay for a long time in between integer phase offsets as opposed to the graphs in the right column). Also, much improvement needs to be made at low SNR, which is around the region of 30 dB where the uncoded BER is  $10^{-2}$ , even for cases where the equalizer is outside the timing loop.

In all the phase convergence plots, we see that the converged phases are all offset from 0 or  $T$  by a small amount ( $\approx 0.1T$ ). This is due to the choice of the PLL loop gains and, to a lesser extent, due to imperfect equalization resulting in residual ISI. A different set of loop gains will cause these offsets to be reduced significantly.

## 2.2 Matching for $6T$ preamble

When the MM TED is used, hang-ups during timing acquisition can be completely eliminated if all the decisions made by the slicer are correct. When a  $6T$  preamble is used, the output should be periodic with period  $6T$ . This also means that there are six possible output sequences that could be obtained depending on the timing offset, assuming noise is absent. One (two, if the phase offset is  $\pm 0.5T$ ) of these six sequences will be ideal in the sense that it will result in convergence to the nearest integer multiple of  $T$  when fed into the MM TED. If this sequence is used **throughout** the entire acquisition stage, then it will be impossible for there to be any change in direction of phase convergence (hang-up). Therefore the objective is to identify the ideal sequence from among the six possibilities, for any initial phase offset. We show that this can be done satisfactorily by simply observing the first six decisions made by the slicer.

Assume that the  $6T$  preamble sequence (i.e. periodic repetition of  $[1 \ 1 \ 1 \ -1 \ -1 \ -1]$ ) is sent into the equalized channel  $[1 \ 2 \ 2 \ 1]$ . Table 2-1 shows where the converged phase will be for the six possible detected sequences (assuming the initial phase offset is  $+0.5$ ).

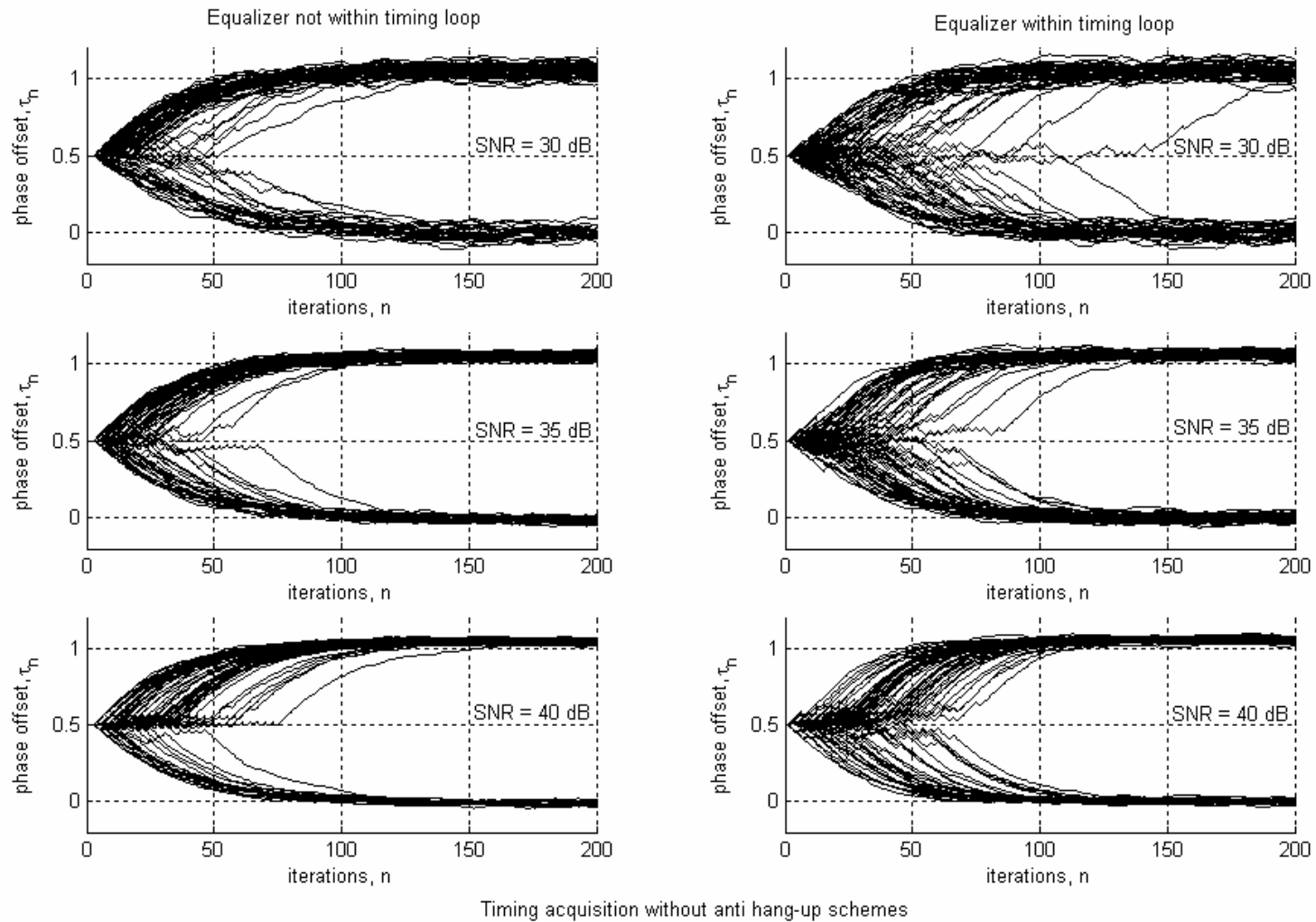
No.	Detected periodic sequence used in TED (1 period shown)						Phase destination
1	4	0	-4	-4	0	4	0
2	0	-4	-4	0	4	4	$T$
3	-4	-4	0	4	4	0	$2T$
4	-4	0	4	4	0	-4	$3T$
5	0	4	4	0	-4	-4	$-2T$
6	4	4	0	-4	-4	0	$-T$

Table 2-1. Detected periodic sequence sent into TED and final phase destination.

In simple threshold detection of  $6T$  preamble for the  $[1 \ 2 \ 2 \ 1]$  target, the thresholds are placed at  $\pm 2$ . Therefore for a phase offset of  $0.5T$  lagging, the detected output, if noise is not present, is given by

$$[4 \ -4 \ (-4) \ -4 \ 4 \ (4)],$$

where the values in brackets have no or very little possibility of being detected otherwise, because they are far from the thresholds of  $\pm 2$ . These points shall be referred to as anchor points. We can see this clearly from Fig. 2-4 where the six points are labeled '1' to '6'.



Timing acquisition without anti hang-up schemes

Fig. 2-2. Phase convergence for  $6T$  preamble without VTD (80 runs);  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.

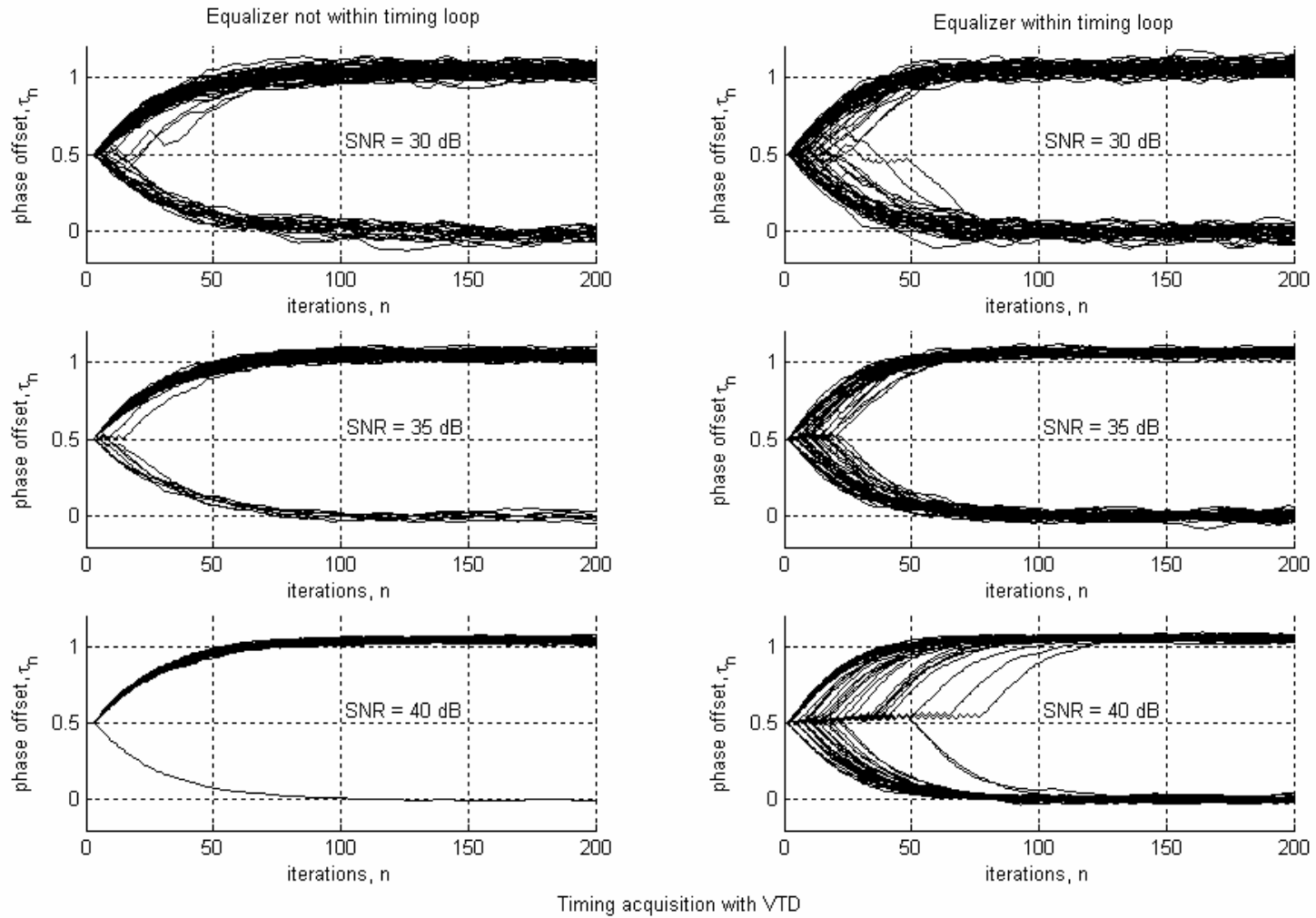


Fig. 2-3. Phase convergence for 6T preamble with VTD (80 runs);  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.



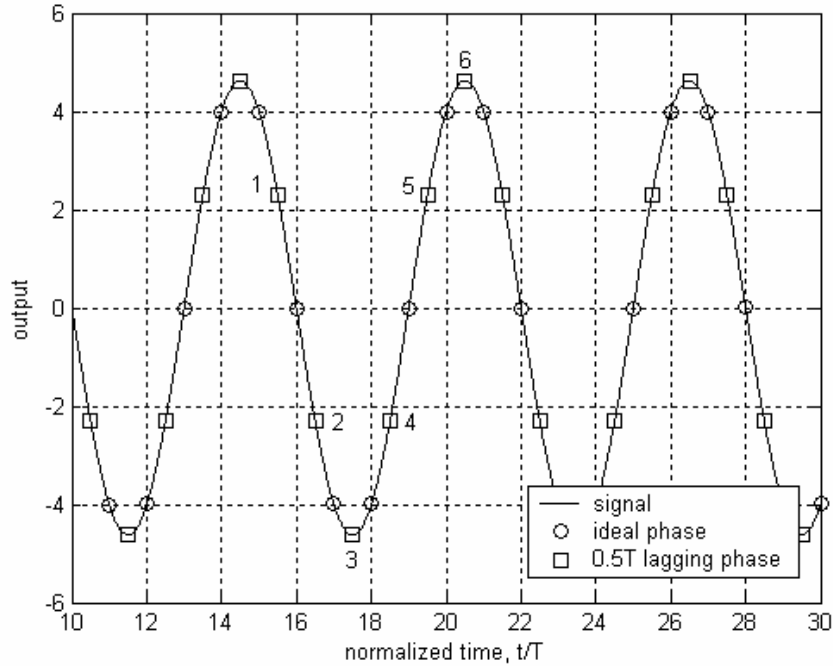


Fig. 2-4. Noiseless output with ideal and  $0.5T$  lagging phases for  $6T$  preamble and target  $[1\ 2\ 2\ 1]$ .

For the lagging phase offset of  $0.5T$ , the first and second sequences shown in Table 2-1 are ideal because they result in convergence to phase offsets of  $0$  and  $T$ , respectively, which are closest to  $0.5T$ . By the fact that the anchor points are unlikely to be detected otherwise, we observe that the detected sequence (e.g.  $[4\ -4\ (-4)\ -4\ 4\ (4)]$ ) will always have two matches each with Sequences 1 and 2 of Table 2-1 at the anchor positions of 3 and 6, irregardless of what the decisions are in the other four positions. Therefore Sequences 1 and 2 both have an advantage of two matches over the other sequences in Table 2-1 and they happen to be the ideal sequences for bringing the normalized phase offset to the nearest integer value. Therefore there arises the motivation to identify such ideal sequences by selecting the one that has the maximum number of matches with the slicer output decisions that are made over one period of the preamble.

Let us now examine if the other four non-anchor points are capable of preventing the correct selection of the ideal detected sequence when such a matching scheme is implemented. It is assumed that the anchor points are not ‘blown’ by noise, i.e. they always cross the slicer thresholds.

### Examination of Sequence 3

Let us examine whether Sequence 3 can beat Sequence 1 during similarity comparisons so that it is chosen in preference to Sequence 1. It is also of interest if Sequence 3 ‘draws’ Sequence 1 because this will result in ambiguity.

By comparison, it can be seen that if noise corrupts the noiseless output (with phase offset of  $0.5T$  lagging and referred to as ‘Source’ in Fig. 2-5) so that the detected sequence  $[0 \ -4 \ (-4) \ 0 \ 4 \ (4)]$  is obtained, there will be a ‘draw’ because both Sequence 1 and Sequence 3 will each have 2 matches and 4 mismatches. (Matches are marked by circles and mismatches by squares in Fig. 2-5).

Source:	4	-4	(-4)	-4	4	(4)
Sequence 1:	4	0	-4	-4	0	4
Detected:	0	-4	(-4)	0	4	(4)
Sequence 3:	-4	-4	0	4	4	0

Fig. 2-5. Contention between Sequence 1 and Sequence 3.

However, this detected output sequence is exactly Sequence 2, the second ideal sequence! Therefore it can be concluded that it is impossible for there to be convergence to a phase offset of  $2T$ .

### Examination of Sequence 4

Upon examining Sequence 4, as in the case of Sequence 3, it can be shown that it is impossible for it to beat Sequence 1. Therefore, there is no way for the phase offset to converge to  $3T$ .

### Examination of Sequence 5

Examining Sequence 5, it can be seen that a ‘draw’ with Sequence 1 will result if the noiseless output sequence is corrupted by noise to yield the following sequence:  $[0 \ -4 \ (-4) \ 0 \ 4 \ (4)]$ . Again, this is exactly Sequence 2, which is an ideal sequence. Therefore it is impossible for the phase offset to converge to  $-2T$ .

### Examination of Sequence 6

Examining Sequence 6, it can be seen that it is again impossible for it to beat Sequence 1, therefore there is no way for the phase offset to converge to  $-T$ .

The discussion now has been focused around a phase offset of  $+0.5T$ . We can see from Fig. 2-4 that if the initial phase offset moves away from  $+0.5T$  either towards 0 or  $T$ , then it becomes increasingly likely that ideal sequences will be selected by the matching scheme because the average distance of all the points from the detection thresholds grows increasingly larger. Therefore the matching scheme will work for all initial phase offsets between 0 and  $T$ .

The above arguments can be applied to all cyclic shifts of the input preamble sequence. Therefore, in conclusion, the entire ideal detected output can be deduced from the first six symbols that are output from the noise corrupted equalized channel. It is impossible for the scheme to fail if the two anchor points are detected correctly. Simulations have been done using the new scheme and the results are shown in Fig. 2-6.

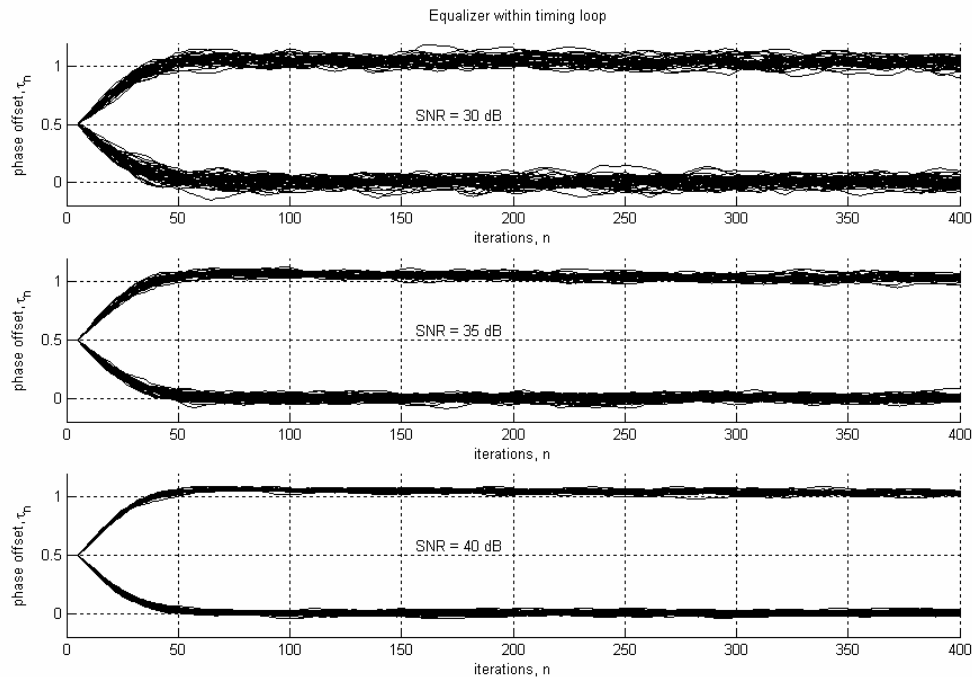


Fig. 2-6. Phase convergence for  $6T$  preamble with matching scheme (80 runs);  
 $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.

From Fig. 2-6, we can see that hang-ups have been completely eliminated. There is no longer indecisiveness around phase offset of  $0.5T$ . Neither is there any change in direction of convergence. Therefore, in conclusion, fast acquisition can be achieved by collecting the first six samples from the equalizer output and using them to identify the optimum sequence to be used throughout the acquisition phase.

### 2.3 Probability of breakdown of matching scheme for $6T$ preamble

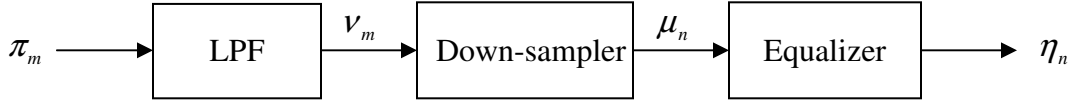


Fig. 2-7. Noise path through the low-pass filter (LPF), the down-sampler and the equalizer.

Fig. 2-7 shows how the noise at the output from the recording channel,  $\pi_m$ , is modified by the low-pass filter (LPF), down-sampler and equalizer. The noise at the output of the LPF is given by

$$v_m = \sum_{i=0}^{N_f-1} f_i \pi_{m-i} \quad (2-5)$$

where  $f_i$  refers to the  $i^{\text{th}}$  tap of the LPF and  $N_f$  refers to the length of the FIR (finite impulse response) LPF. The noise after passing through the down-sampler is

$$\mu_n = \sum_{i=0}^{N_f-1} f_i \pi_{nL-i} \quad (2-6)$$

where  $L$  is the over-sampling factor. When  $\mu_n$  is passed through the equalizer, it becomes

$$\eta_n = \sum_{i=0}^{N_w-1} w_i \mu_{n-i} = \sum_{i=0}^{N_w-1} \sum_{j=0}^{N_f-1} w_i f_j \pi_{(n-i)L-j} \quad (2-7)$$

where  $w_i$  is the  $i^{\text{th}}$  tap of the equalizer and  $N_w$  is the length of the FIR equalizer. Therefore the autocorrelation function of the noise at the output of the equalizer is

$$\begin{aligned}
\phi_{\eta\eta}(k) &= E\{\eta_n \eta_{n+k}\} \\
&= \sum_{i=0}^{N_w-1} \sum_{j=0}^{N_f-1} \sum_{a=0}^{N_w-1} \sum_{b=0}^{N_f-1} w_i f_j w_a f_b E\{\pi_{(n-i)L-j} \pi_{(n+k-a)L-b}\} \\
&= \sigma_\pi^2 \sum_{i=0}^{N_w-1} \sum_{j=0}^{N_f-1} \sum_{a=0}^{N_w-1} w_i w_a f_j f_{(k-a+i)L+j}
\end{aligned} \tag{2-8}$$

where  $\sigma_\pi^2$  is the AWGN variance at the output of the recording channel i.e. the variance of  $\pi_m$ . Since  $0 \leq (k-a+i)L+j \leq N_f-1$ , the range of values of  $k$  is given by

$$-\left\lfloor \frac{N_f-1}{L} + N_w - 1 \right\rfloor \leq k \leq \left\lfloor \frac{N_f-1}{L} + N_w - 1 \right\rfloor. \tag{2-9}$$

The power spectral density PSD of the noise at the output of the equalizer is thus

$$\Phi(\Omega) = \sum_{k=-\left\lfloor \frac{N_f-1}{L} + N_w - 1 \right\rfloor}^{\left\lfloor \frac{N_f-1}{L} + N_w - 1 \right\rfloor} \phi_{\eta\eta}(k) e^{-j2\pi k \Omega} \tag{2-10}$$

where  $\Omega$  is the frequency normalized with respect to the data rate. The variance of the noise that corrupts the waveform is given by  $\sigma_\eta^2 = \phi_{\eta\eta}(0) = \int_{-0.5}^{0.5} \Phi(\Omega) d\Omega$ .

In order to compute the worst case probability, a sampling phase offset of  $0.5T$  is chosen because this will result in the most number of noiseless output signal points being placed closest to the two thresholds at  $+2$  and  $-2$ . From Fig. 2-4, it can be seen that the maximum number of 4 signal points per period are placed closest possible to the two thresholds. The lagging phase points (square) are calculated to be at  $\pm 2.30$  and  $\pm 4.62$  using the expression

$$y(t) = \dots + 4\text{sinc}\left(\frac{t}{T}\right) - 4\text{sinc}\left(\frac{t-2T}{T}\right) - 4\text{sinc}\left(\frac{t-3T}{T}\right) + 4\text{sinc}\left(\frac{t-5T}{T}\right) + \dots \tag{2-11}$$

where  $y(t)$  is the output time waveform and  $t = 0.5T, 1.5T, 2.5T, \dots$

Two assumptions are made in order to compute the probability that the scheme will fail:

1. In the absence of noise, the recording channel is perfectly equalized by the equalizer to the target  $[1 \ 2 \ 2 \ 1]$ .

2. Phase drift due to sampling frequency mismatch is negligible because the observation window is only one preamble period, which is very short (6 samples).

Observe in Fig. 2-4 that there are anchor points at  $\pm 4.62$  (labeled '3' and '6') that are far from the detection thresholds and thus are unlikely to be detected as other than ' $\pm 4$ '. In Section 2.2, it was shown that if the two anchor points over any period are indeed detected as ' $\pm 4$ ', then it is impossible for the matching scheme to fail. Now we consider cases where one of the two anchors is 'blown' by noise and so it is detected as '0' instead of ' $\pm 4$ '. It is highly unlikely that both anchors will be 'blown' by noise during any observation window and thus the contribution of such occurrences to the probability of failure may be neglected.

Further, while it is easy for the noise to cause the Points '1' and '5' to be detected as '0', it is extremely unlikely for them to enter the detection region for symbol '-4'. Similarly, for Points '2' and '4', it is easy for them to end up being detected as '0' or '-4' but extremely unlikely for them to be detected as '+4'. Therefore such cases are eliminated from our study.

Bearing these in mind and assuming that the anchor labeled '3' is 'blown' so that that point is detected as '0', the set of all 16 possible and likely detected sequences is listed in Table 2-2.

No.	Likely detected sequence	No.	Likely detected sequence
1	4 -4 0 -4 4 4	9	4 -4 0 -4 0 4
2	0 -4 0 -4 4 4	10	0 -4 0 -4 0 4
3	4 0 0 -4 4 4	11	4 0 0 -4 0 4
4	0 0 0 -4 4 4	12	0 0 0 -4 0 4
5	4 -4 0 0 4 4	13	4 -4 0 0 0 4
6	0 -4 0 0 4 4	14	0 -4 0 0 0 4
7	4 0 0 0 4 4	15	4 0 0 0 0 4
8	0 0 0 0 4 4	16	0 0 0 0 0 4

Table 2-2. List of all possible and likely detected sequences when the anchor labeled '3' in Fig. 2-4 is 'blown' by noise.

Table 2-1 shows all possible cyclic shifts of the output obtained if there are no phase offsets. The first and second sequences in Table 2-1 are ideal for phase offset  $0.5T$ .

Comparing the sequences in Table 2-2 with that in Table 2-1, it is found that only Sequence 1 in Table 2-2 is capable of producing ambiguity during matching because it has three mismatches and three matches each with Sequences 1, 2, 3 and 6 of Table 2-1. None of the other sequences in Table 2-2 will result in the ideal sequences (1 and 2) of Table 2-1 being beaten.

If the anchor labeled ‘6’ in Fig. 2-4 is ‘blown’ rather than the anchor labeled ‘3’, then by a simple check it can be easily seen that the detected sequence that will cause ambiguity is given by  $[4 \ -4 \ -4 \ -4 \ 4 \ 0]$ . The probability of occurrence of this sequence is equal to the probability of occurrence of Sequence 1 of Table 2-2. Thus the probability of the matching scheme failing is given by twice the probability of occurrence of Sequence 1 in Table 2-2. The probability of occurrence of this sequence is given by

$$P_{seq1} = \int_{-0.3}^{\infty} \int_{-\infty}^{0.3} \int_{2.62}^{6.62} \int_{-\infty}^{0.3} \int_{-0.3}^{\infty} \int_{-2.62}^{\infty} p(\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6) d\eta_6 d\eta_5 d\eta_4 d\eta_3 d\eta_2 d\eta_1, \quad (2-12)$$

where  $p(\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6) = \frac{1}{(2\pi)^{6/2} \sqrt{\det(C)}} \exp\left(-\frac{1}{2} \underline{\eta}^T C^{-1} \underline{\eta}\right)$  is the joint Gaussian

pdf of the zero mean Gaussian random variables  $\eta_1, \dots, \eta_6$ ,  $\underline{\eta}^T$  is the row vector

$$[\eta_1 \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5 \ \eta_6] \text{ and } C = \begin{bmatrix} \phi_{\eta\eta}(0) & \cdots & \phi_{\eta\eta}(5) \\ \vdots & \ddots & \vdots \\ \phi_{\eta\eta}(5) & \cdots & \phi_{\eta\eta}(0) \end{bmatrix} \text{ is the covariance matrix.}$$

Equation (2-12) may be evaluated using numerical integration. However it is too computationally intensive. Therefore we shall approximate it by

$$\begin{aligned} P_{seq1} &= P(\eta_1 \geq -0.3, \eta_2 \leq 0.3, 2.62 \leq \eta_3 \leq 6.62, \eta_4 \leq 0.3, \eta_5 \geq -0.3, \eta_6 \geq -2.62) \\ &= P(A_1 A_2 A_3 A_4 A_5 A_6) \\ &= P(A_1 | A_2 A_3 A_4 A_5 A_6) P(A_2 | A_3 A_4 A_5 A_6) P(A_3 | A_4 A_5 A_6) P(A_4 | A_5 A_6) P(A_5 A_6) \\ &\approx P(A_1 | A_2) P(A_2 | A_3) P(A_3 | A_4) P(A_4 | A_5) P(A_5 A_6) \\ &= \frac{P(A_1 A_2)}{P(A_2)} \cdot \frac{P(A_2 A_3)}{P(A_3)} \cdot \frac{P(A_3 A_4)}{P(A_4)} \cdot \frac{P(A_4 A_5)}{P(A_5)} \cdot P(A_5 A_6), \end{aligned} \quad (2-13)$$

where  $A_1, A_2, A_3, A_4, A_5$  and  $A_6$  have been used to represent  $\eta_1 \geq -0.3$ ,  $\eta_2 \leq 0.3$ ,  $2.62 \leq \eta_3 \leq 6.62$ ,  $\eta_4 \leq 0.3$ ,  $\eta_5 \geq -0.3$  and  $\eta_6 \geq -2.62$  respectively. The approximation

of (2-13) is based on the assumption that  $\eta_k$ , where  $k \in \{1, \dots, 5\}$ , is weakly correlated with all  $\eta_l$ , where  $l > k + 1$  and  $l \in \{3, \dots, 6\}$ . Thus, the worst-case probability of failure is then given by (for phase offset =  $0.5T$ )

$$P(\text{failure} \mid \tau = 0.5) = 2P_{seq1}. \quad (2-14a)$$

Then, the probability of failure can be upper-bounded as follows:

$$\begin{aligned} P(\text{failure}) &= \int_0^{1.0} P(\text{failure} \mid \tau) p(\tau) d\tau \\ &\leq P(\text{failure} \mid \tau = 0.5) \int_0^{1.0} p(\tau) d\tau \\ &= P(\text{failure} \mid \tau = 0.5), \end{aligned} \quad (2-14b)$$

where  $p(\tau)$  denotes the pdf (uniform) of the initial phase offset  $\tau$ . Based on (2-14b), a graph of the upper bound of the probability of the matching scheme failing is shown in Fig. 2-8. From the graph, we can see that the matching scheme is very reliable.

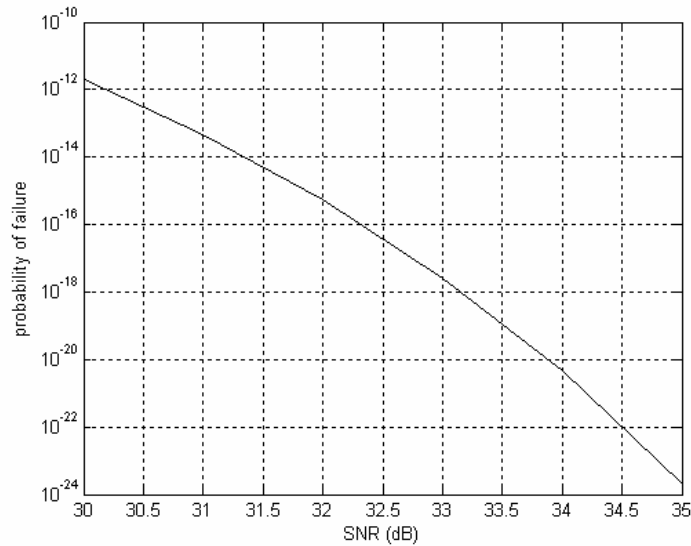


Fig. 2-8. Probability of the 6T preamble matching scheme failing for various SNR.

## 2.4 Improvements to the matching scheme for 6T preamble

Even though the matching scheme is able to eliminate hang-up completely, there is still one shortcoming. This is the focus of this section.



Let us restrict discussion about the initial phase offset to between 0 and  $T$ , which we shall call the principal range. (Due to the cyclic nature of the timing function and the periodic nature of the preamble, all properties developed within the principal range may be extended to other ranges between  $kT$  and  $(k+1)T$  where  $k$  is any integer).

If the initial phase offset is in the vicinity of  $0.5T$ , e.g.  $0.45T$  to  $0.55T$ , it is reasonable for the scheme to steer the final phase to either 0 or  $T$ . However, if the initial phase gets nearer to  $T$  (or 0), then the scheme should steer the final phase to  $T$  (or 0) i.e. the final phase should be the integer multiple of  $T$  that is closest to the initial phase offset. This will ensure the fastest possible acquisition for a given set of timing loop parameters.

For the matching scheme, it is true that if the initial phase gets closer and closer to  $T$ , then it gets increasingly likely that the final phase will be steered towards  $T$ . However, there is also a high possibility for the phase to be steered towards 0 when the phase is greater than  $0.55T$  but not sufficiently close to  $T$ , e.g.  $0.6T$ , and it represents a loss in acquisition speed. This is illustrated in Fig. 2-9 for an initial phase offset of  $0.6T$ .

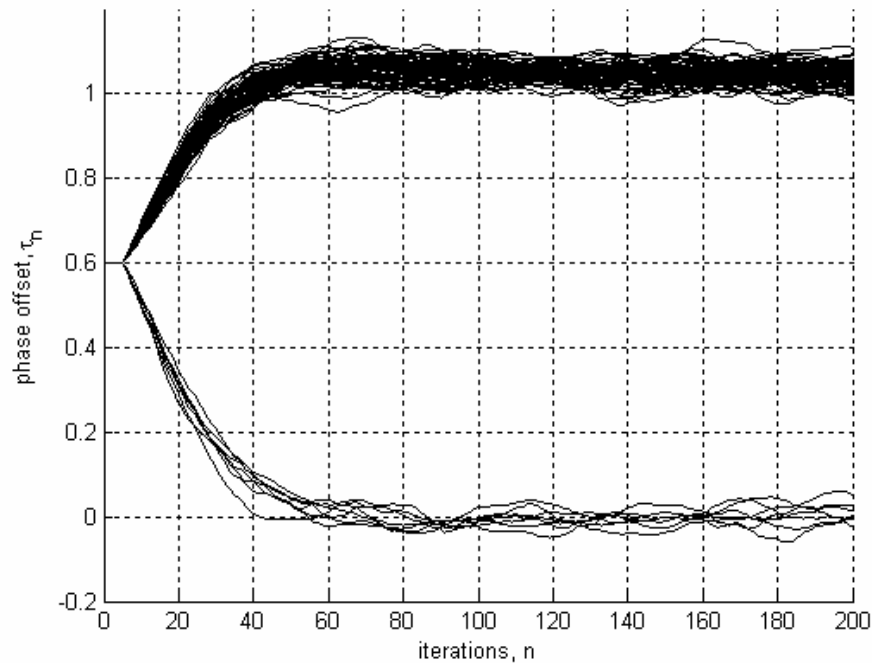


Fig. 2-9. Phase convergence for  $6T$  preamble with matching scheme (80 runs);  
 $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; SNR = 35 dB; frequency offset = 0.1%.

Fig. 2-10 and Fig. 2-11 show the noiseless output time waveform from the perfectly equalized channel when the phase offset is  $0.6T$  and  $0.4T$ , respectively.

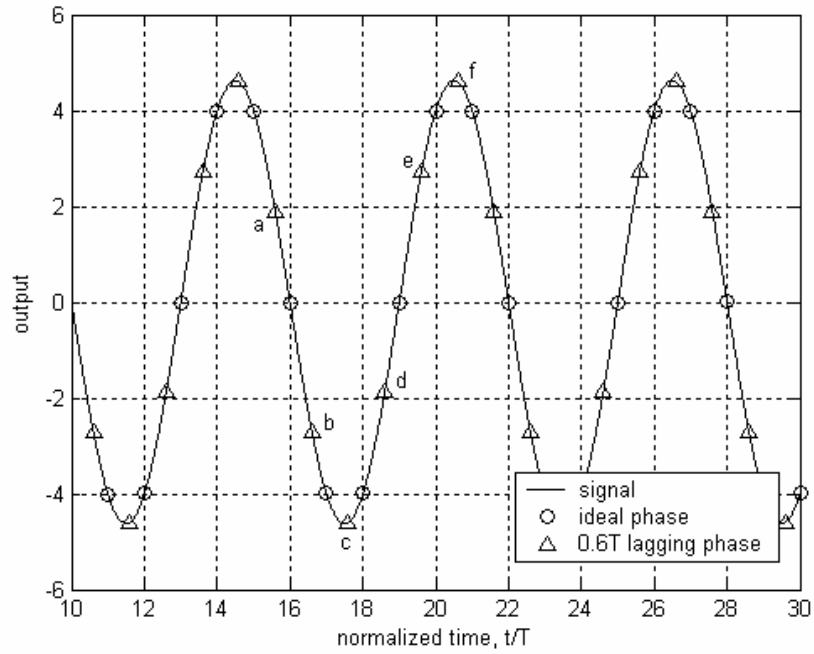


Fig. 2.10. Noiseless output with ideal and  $0.6T$  lagging phases for  $6T$  preamble and target  $[1\ 2\ 2\ 1]$ .

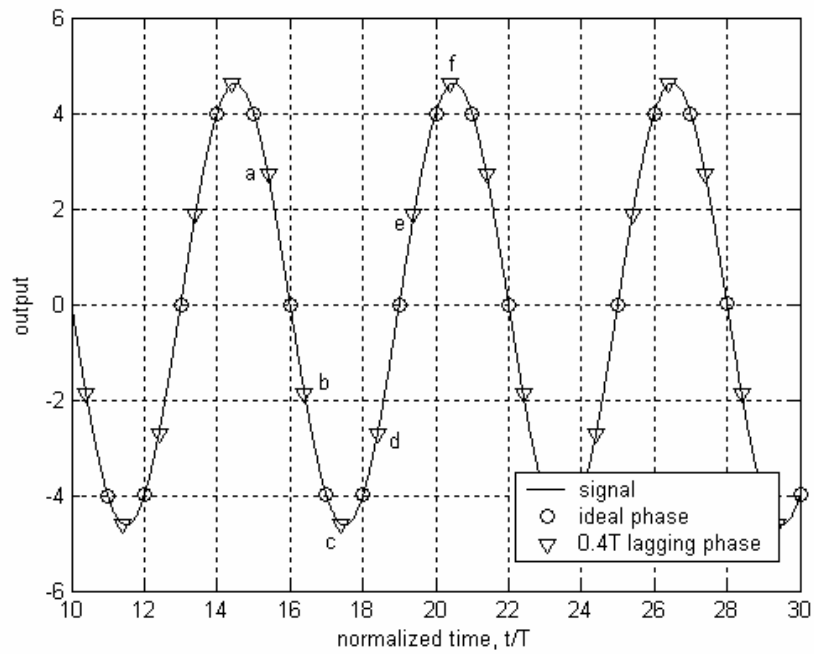


Fig. 2.11. Noiseless output with ideal and  $0.4T$  lagging phases for  $6T$  preamble and target  $[1\ 2\ 2\ 1]$ .

From Fig. 2-10, we can see that when the phase offset is  $0.6T$ , the distance (i.e. amplitude difference) between Points ‘a’ and ‘d’ is shorter than the distance between Points ‘b’ and ‘e’. From Fig. 2-11, if the phase offset is  $0.4T$ , the opposite is true, i.e. the distance between Points ‘b’ and ‘e’ is shorter than the distance between Points ‘a’ and ‘d’. Strictly speaking this will not always be true in the presence of noise but it does serve as an approximate way of estimating which half in the principal range the initial phase offset belongs to.

Since we are restricting ourselves to phases between 0 and  $T$ , the two relevant ideal detected sequences are given by

$$\begin{array}{rcccccc}
 1. & 4 & 0 & (-4) & -4 & 0 & (4) \\
 2. & 0 & -4 & (-4) & 0 & 4 & (4)
 \end{array}$$

which correspond to Sequences 1 and 2 of Table 2-1. From Fig. 2-10 and Fig. 2-11, ignoring the two anchor points (‘c’ and ‘f’), we can also see that there are always two non-anchor points that are closer to the thresholds than the other two non-anchor points. These points closest to the threshold are most likely to be detected wrongly. If indeed they are detected wrongly while the rest are detected correctly, then Sequences 1 and 2 will ‘draw’ each other and there is a need thus to decide which sequence to use (previously, the sequence to be chosen was decided by flipping a coin). The distances between Points ‘a’ and ‘d’ and between Points ‘b’ and ‘e’ may be used to select the better sequence according to the following rule

$$|y_a - y_d| \underset{Seq1}{\overset{Seq2}{>}} |y_b - y_e| \tag{2-15}$$

where  $y_a$ ,  $y_b$ ,  $y_d$  and  $y_e$  are the noise corrupted versions of points ‘a’, ‘b’, ‘d’ and ‘e’ in Fig. 2-10 and Fig. 2-11. This rule will be implemented only when there is a ‘draw’ between the two sequences that have the highest number of matches. Therefore there will still be instants when corrections are not made, e.g. if the initial phase offset is  $0.6T$  and,

due to noise, Sequence 1 has a higher number of matches than Sequence 2, resulting in Sequence 1 being chosen and the final phase destination being 0.

Despite the shortcomings of the correction rule described by (2-15), its performance is reasonably good when the SNR is not too low, e.g. 35 dB. The effect of implementing this rule on timing acquisition is illustrated in Fig. 2-12. We can see that all paths converge to  $T$ , whereas there are many paths that converge to 0 when there is no correction rule (see Fig. 2-9).

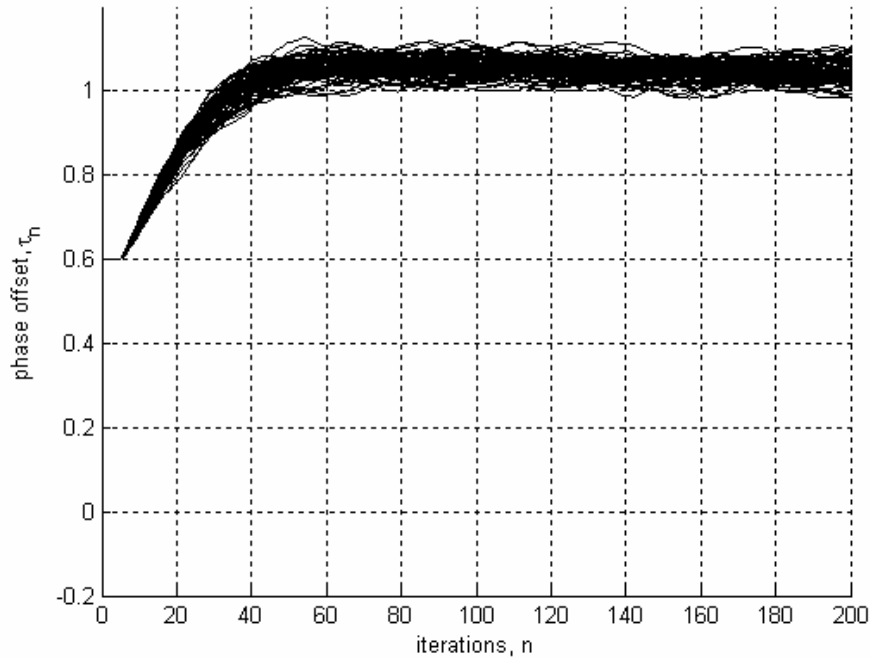


Fig. 2-12. Phase convergence for  $6T$  preamble with matching scheme and correction (80 runs);  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; SNR = 35 dB; frequency offset = 0.1%.

## 2.5 Implementation issues of correction rule

The implementation of the correction rule given by (2-15) requires locating the points ‘a’, ‘b’, ‘d’ and ‘e’. The position of these points will change with the range (range is any interval from  $kT$  to  $(k+1)T$  where  $k$  is any integer) within which is found the initial phase offset. Thus there is a need to search for these points before the rule can be used.

From Fig. 2-10 and Fig. 2-11, we see that points ‘a’ and ‘d’ both occur one bit interval after an anchor point whereas points ‘b’ and ‘e’ both occur one bit interval before an anchor point. Therefore if the anchor points (‘c’ and ‘f’) can be located, then the points ‘a’, ‘b’, ‘d’ and ‘e’ can also be located. (Note that due to the periodic nature of the preamble, the symbol one bit interval after the sixth symbol is equivalent to the first symbol and the symbol one bit interval before the first symbol is equivalent to the sixth symbol).

The correction rule is activated whenever there are two sequences with the maximum (and equal) number of matches. These two sequences will always have only two positions with symbol values that are the same. For example, Sequences 1 and 2 in Table 2-1 have the same symbol values only in positions 3 and 6. These positions where the symbol values agree are the anchor point positions. Therefore the location of anchor points can be found by searching for the places where the two sequences have the same symbol value. The flow chart in Fig. 2-13 summarizes the entire correction scheme.

## **2.6 Summary**

In this chapter, we investigated timing acquisition for the  $6T$  preamble and [1 2 2 1] PR target. Fast acquisition may be achieved by identifying and using the preamble sequences with the ideal cyclic shift. Ideal sequences may be identified by simply matching six consecutive samples with the cyclic shifts of the  $6T$  preamble and getting the preamble sequence with the most number of matches. The probability of failure for such a scheme is found to be low. A correction rule has also been developed to further speed up timing acquisition.

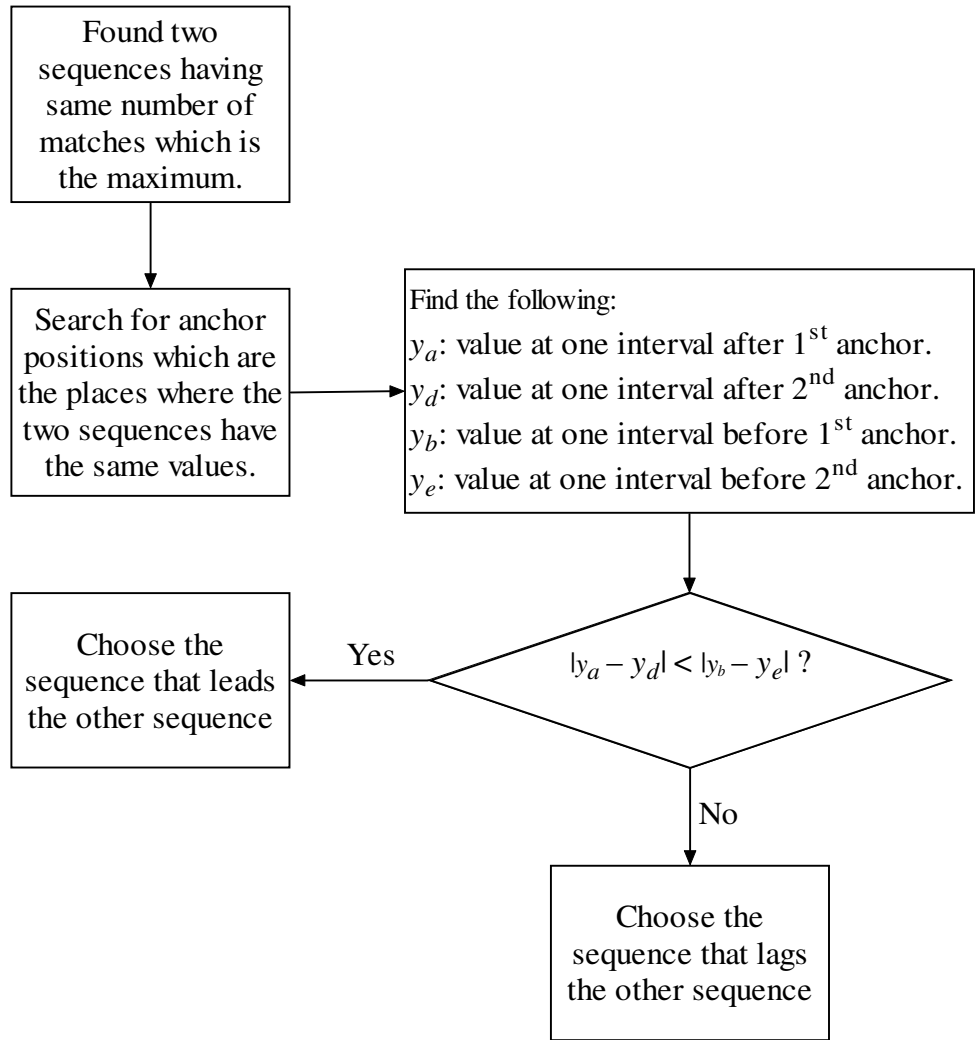


Fig. 2-13. Flow chart summarizing the 6T matching scheme correction rule.

## Chapter 3

### Fast Acquisition 1b:

### Matching Scheme for $8T$ and $10T$ Preambles

In this chapter we present extension of the matching scheme of fast timing acquisition described in Chapter 2 to cover  $8T$  and  $10T$  preambles. Even though the basic principle is the same as for the  $6T$  preamble, certain modifications need to be made when  $8T$  or  $10T$  preambles are used. Moreover, for longer period preambles, the time spent to collect a number of samples equal to the period of the preamble may no longer be justifiable because no timing updates are done during this data collection phase. In this chapter, we introduce a way to reduce the average time needed for data collection.

#### 3.1 Direct matching scheme for $8T$ preamble

Fig. 3-1 shows the noiseless output of the equalizer with ideal sampling instants when an  $8T$  preamble is used. Fig. 3-2 shows the case when the ideal sampling instants are offset by  $\pm 0.5T$ .

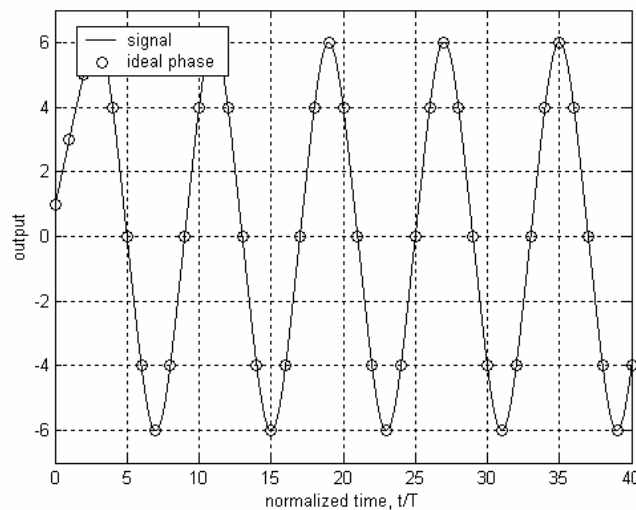


Fig. 3-1. Noiseless output of the equalizer with ideal sampling phase for  $8T$  preamble and target  $[1\ 2\ 2\ 1]$ .

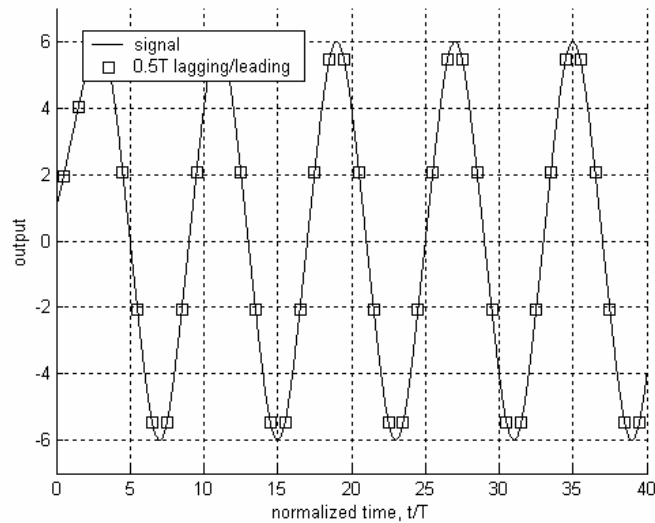


Fig. 3-2. Noiseless output of the equalizer with sampling phases of  $\pm 0.5T$  for  $8T$  preamble and target [1 2 2 1].

From Fig. 3-1, it can be seen that in ordinary threshold detection, the thresholds would be placed at  $\pm 2$  and  $\pm 5$ . However, the noise margin with respect to the  $\pm 5$  thresholds is not large. Fig. 3-2 shows that all the sample points are very close to the thresholds  $\pm 2$  and  $\pm 5$ . This is extremely undesirable because the decisions made are bound to be very erratic in the presence of noise.

Going back again to Fig. 3-1, we note that  $+6$  is always between two  $+4$ 's and  $-6$  is always between two  $-4$ 's. Therefore their locations can be deduced from the distribution of  $+4$  and  $-4$  samples. Hence, we remove the  $\pm 5$  thresholds. When this is done, from Fig. 3-2, we can see that the number of sample points per period that are near to detection thresholds is reduced by four. Thus anchor points (points that are far from detection thresholds  $\pm 2$ ; in this case, they are points that are beyond  $\pm 4$  and close to  $\pm 6$ ), which are central to the derivation of a matching scheme, are created.

With only the  $\pm 2$  thresholds, the eight cyclic shifts of each period of the detected output samples when sampling phase is ideal and noise is absent are shown in Table 3-1.

The matching scheme, in this case, is as follows. The samples detected using only  $\pm 2$  thresholds are compared with the sequences listed in Table 3-1. The sequence that has the largest number of matches with the detected sequence will then be modified by inserting a  $+6$  between two  $+4$ 's and a  $-6$  between two  $-4$ 's and sent to drive the



MM TED, and this modified sequence will be used repeatedly until the end of the acquisition phase. Such identification of the ideal sequence shall be referred to as ‘direct matching’.

Sequence	Detected							
1	0	4	4	<b>4</b>	<b>0</b>	<b>-4</b>	-4	-4
2	4	4	<b>4</b>	<b>0</b>	<b>-4</b>	-4	-4	0
3	4	<b>4</b>	<b>0</b>	<b>-4</b>	-4	-4	0	4
4	<b>4</b>	<b>0</b>	<b>-4</b>	-4	-4	0	4	4
5	0	-4	-4	<b>-4</b>	<b>0</b>	<b>4</b>	4	4
6	-4	-4	<b>-4</b>	<b>0</b>	<b>4</b>	4	4	0
7	-4	<b>-4</b>	<b>0</b>	<b>4</b>	4	4	0	-4
8	<b>-4</b>	<b>0</b>	<b>4</b>	4	4	0	-4	-4

Table 3-1. Cyclic shifts of detected output samples when sampling phase is ideal, noise is absent and detection thresholds are set at ‘ $\pm 2$ ’.

We can see that direct matching will work for the  $8T$  preamble by comparing Fig. 3-2 and Fig. 2-4. These figures show the noiseless output signals for the worst case phase offsets, when the maximum number of points end up close to the detection thresholds, for the  $8T$  and  $6T$  preambles. The ratio of anchor points to non-anchor points is 1:2 in Fig. 2-4 it is 1:1 in Fig. 3-2 (with thresholds at ‘ $\pm 2$ ’). Therefore, if direct matching works for the  $6T$  case, then it should also work for the  $8T$  case.

In summary, when the direct matching scheme is to be applied to the  $8T$  preamble case, the changes that need to be done as follows.

1. The detection thresholds at ‘ $\pm 5$ ’ must be axed leaving only the thresholds at ‘ $\pm 2$ ’ in order to create anchors.
2. Matching is done with the eight possible ideal sequences listed in Table 3-1.
3. The matched sequence is modified by inserting a ‘+6’ between two ‘+4’s and a ‘-6’ between two ‘-4’s before being sent into MM TED to be used repeatedly throughout the entire acquisition process. (Consider Fig. 3-3 that shows Sequence 8 of Table 3-3 arranged in a circle. Due to the periodic nature of the preamble, Position 1 is considered to be between Positions 8 and 2, Position 8 is considered to be between Positions 7 and 1.

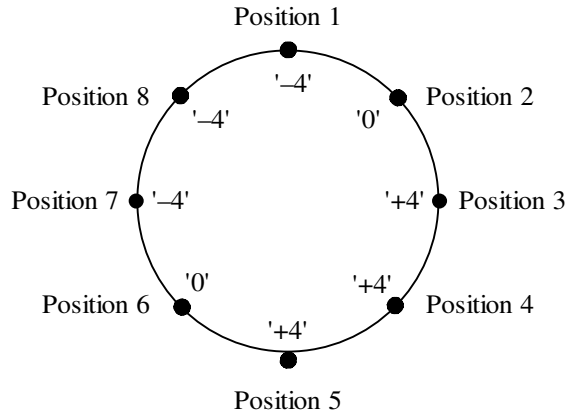


Fig. 3-3. Sequence 8 of Table 3-3 arranged in a circle.

Simulations were done and the phase convergence plots are shown in Fig. 3-4. We can see that, as predicted, the direct matching scheme works and is effective in completely eliminating hang-ups.

### 3.2 Shortening the mean data collection time

For the direct matching schemes described in Section 3.1 and Chapter 2, there is a need to detect one period of the equalizer output, e.g. six when the  $6T$  preamble is used and eight when the  $8T$  preamble is used, before any timing updates can be done. The time spent to collect this data might be insignificant for preambles with short periods but for preambles with long periods, it can no longer be considered insignificant. In this section, we present a new way of doing matching that is more ‘indirect’ than the direct matching schemes. This new method is able to reduce the average number of samples that needs to be collected before an ideal sequence is selected. Throughout this section, the preamble used is  $8T$  and the detection thresholds are at  $\pm 2$ .

From Table 3-1, we see that if the location of either ‘4 0 -4’ or ‘-4 0 4’ is known (these have been made bold), an entire sequence can be generated uniquely. We shall refer to the sub-sequences ‘4 0 -4’ and ‘-4 0 4’ as seeds. It will be shown that in the presence of sampling offsets, these seeds may be used to create the ideal sequences for use in the MM TED.

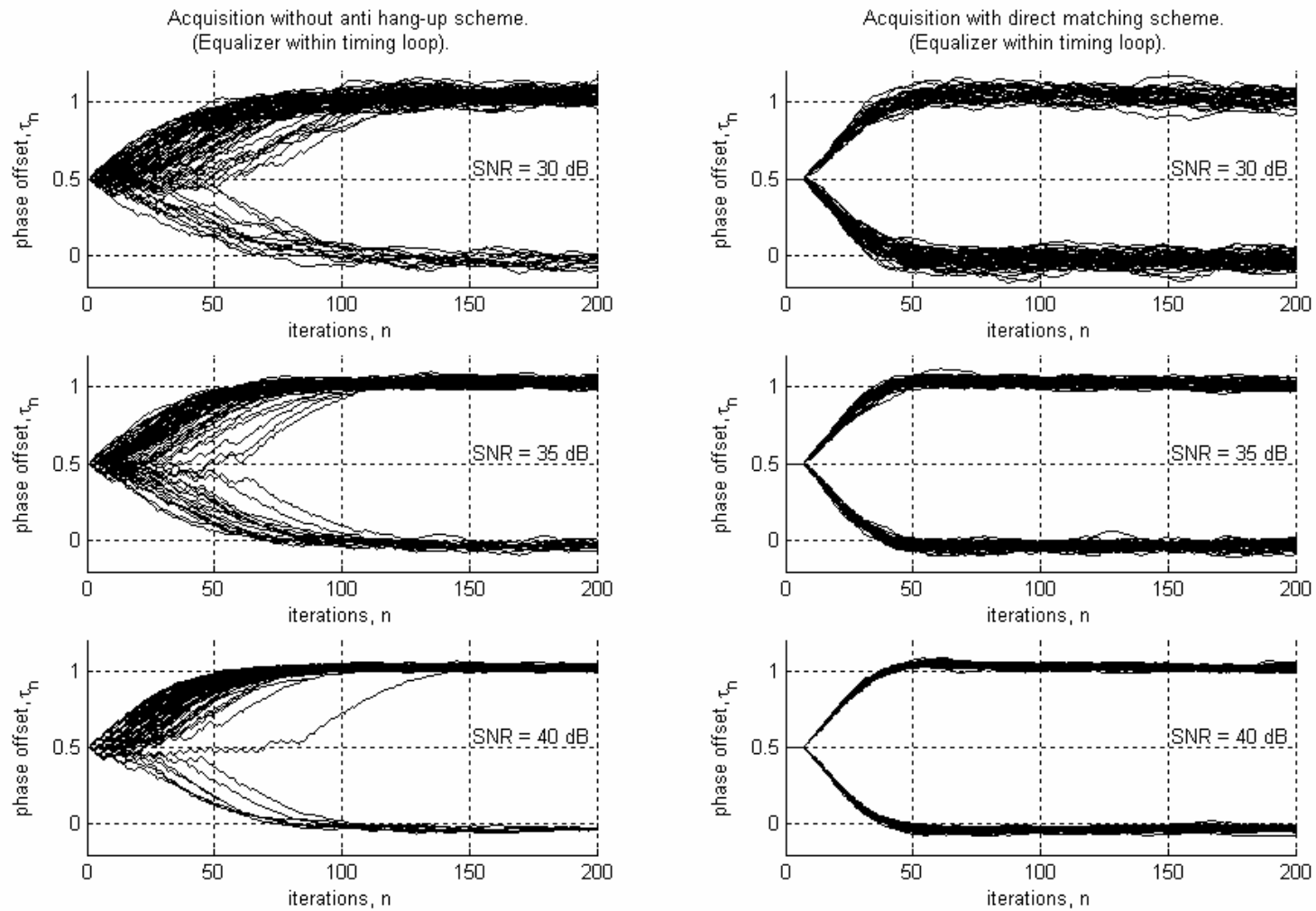
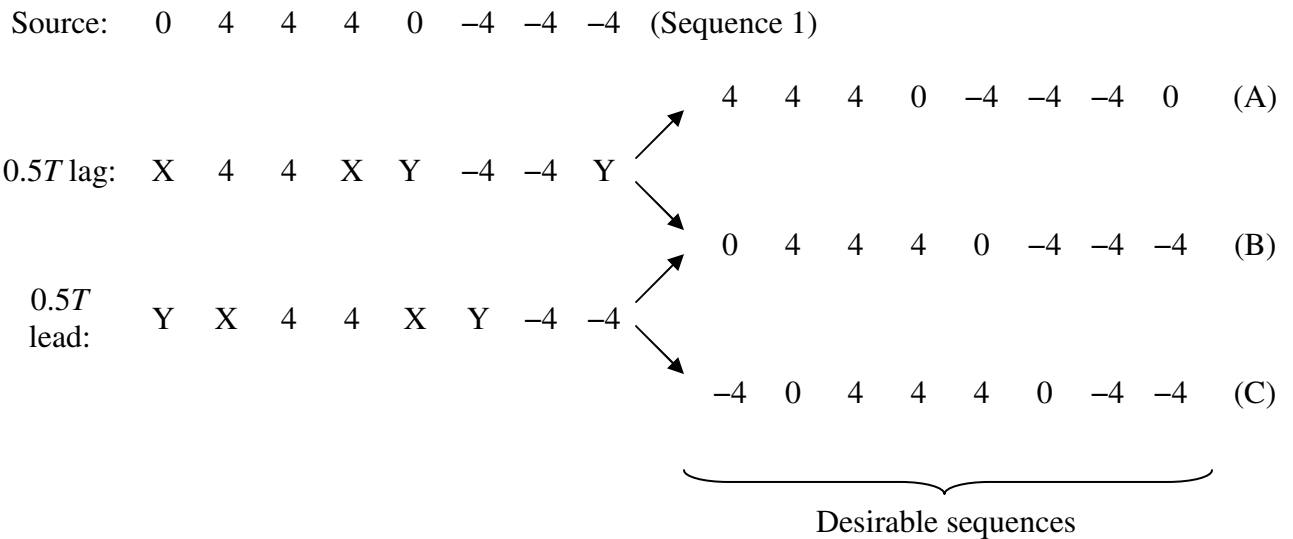


Fig. 3-4. Phase convergence for 8T preamble (80 runs);  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.

Considering Sequence 1 in Table 3-1, let us now examine the possible sequences that can be derived from it when the sampling instant is offset by  $\pm 0.5T$ . When the offset is  $0.5T$  lag, the ideal desirable sequences are  $[4 \ 4 \ 4 \ 0 \ -4 \ -4 \ -4 \ 0]$  and  $[0 \ 4 \ 4 \ 4 \ 0 \ -4 \ -4 \ -4]$ , which will lead to converged phases  $T$  and  $0$ , respectively. When the offset is  $0.5T$  lead, the ideal desirable sequences are  $[0 \ 4 \ 4 \ 4 \ 0 \ -4 \ -4 \ -4]$  and  $[-4 \ 0 \ 4 \ 4 \ 4 \ 0 \ -4 \ -4]$ , which will lead to converged phases of  $0$  and  $-T$  respectively. This is summarized as follows.




Here, X is a sample point near the '+2' threshold and its two possible values are '+4' and '0'. Similarly, Y is a sample point near the '-2' threshold and its two possible values are '0' and '-4'. The points with numerical values of '+4' or '-4' are the anchor points. In order to seek the location of the seeds '4 0 -4' or '-4 0 4', we can search for the first occurrence of either of the sequences '4 Z -4' or '-4 Z 4', where Z can be '+4', '0' or '-4'. However, it is possible that the sequences '4 Z -4' or '-4 Z 4' cannot be found even within one whole period of the preamble. An example is given below where the  $0.5T$  lag sequence is examined.

**Possible 0.5T lag sequences**

Sequences (A) or (B) are desirable sequences for 0.5T lag.


Possibility 1:    X    4    4    **0**    **-4**    -4    -4    Y




(4    0    -4)    →    (A)

(4    0    -4)    →    (A)


Possibility 2:    X    4    4    **0**    **0**    -4    -4    Y






(4    0    -4)    →    (B)

Possibility 3:    X    4    4    **4**    **-4**    -4    -4    Y



(4    0    -4)    →    (A)

Possibility 4:    X    4    4    **4**    **0**    -4    -4    Y



(4    0    -4)    →    (B)

‘Possibility 2’ is an example of seeds being absent. It does not, however, cause any serious problem and there are two different ways of dealing with it as shown above. By running through all the four possibilities, we can see that a desirable sequence can always be obtained from a 0.5T lag sequence by searching for seeds or creating one from the first occurrence of the ‘0 0’ sequence, and then using the seed to generate the desirable

sequence. The number of samples that need to be collected before starting phase updates is either five or six. Let's now examine the  $0.5T$  lead sequences.

**Possible  $0.5T$  Lead Sequences**

Sequences (B) or (C) are desirable TED reference sequences when the phase offset is  $-0.5T$ .

Possibility 1:  $\begin{matrix} -4 & 0 & 4 & 4 & X & Y & -4 & -4 \\ \underbrace{\hspace{10em}} \\ (-4 & 0 & 4) & \rightarrow & (C) \end{matrix}$

Possibility 2:  $\begin{matrix} -4 & 4 & 4 & 4 & X & Y & -4 & -4 \\ \underbrace{\hspace{10em}} \\ (-4 & 0 & 4) & \rightarrow & (C) \end{matrix}$

Possibility 3:  $\begin{matrix} 0 & 0 & 4 & 4 & X & Y & -4 & -4 \\ \underbrace{\hspace{10em}} \\ (-4 & 0 & 4) & \rightarrow & (C) \end{matrix}$

Possibility 4a:  $\begin{matrix} 0 & 4 & 4 & 4 & 0 & -4 & -4 & -4 \\ \underbrace{\hspace{10em}} \\ (4 & 0 & -4) & \rightarrow & (B) \end{matrix}$

$$\begin{array}{cccccccc}
 & & & & (4 & 0 & -4) & \rightarrow & (B) \\
 & & & & \underbrace{\hspace{1.5cm}} & & & & \\
 \text{Possibility 4b:} & \mathbf{0} & \mathbf{4} & 4 & 4 & \mathbf{0} & \mathbf{0} & -4 & -4 \\
 & & & & & \underbrace{\hspace{1.5cm}} & & & \\
 & & & & & (4 & 0 & -4) & \rightarrow & (C)
 \end{array}$$

$$\begin{array}{cccccccc}
 \text{Possibility 4c:} & \mathbf{0} & \mathbf{4} & 4 & 4 & \mathbf{4} & \mathbf{0} & -4 & -4 & -4 \\
 & & & & & \underbrace{\hspace{1.5cm}} & & & & \\
 & & & & & (4 & 0 & -4) & \rightarrow & (B)
 \end{array}$$

$$\begin{array}{cccccccc}
 \text{Possibility 4d:} & \mathbf{0} & \mathbf{4} & 4 & 4 & \mathbf{4} & \mathbf{0} & -4 & -4 \\
 & & & & & \underbrace{\hspace{1.5cm}} & & & \\
 & & & & & (4 & 0 & -4) & \rightarrow & (C)
 \end{array}$$

Thus it can be seen that for  $0.5T$  lead, the same technique can be used to generate desirable sequences for the TED. The number of samples that need to be collected before starting phase updates is three, six or seven. (We show later that we rarely have to collect up to seven samples before phase updates may be started). Once the single period of the desirable sequence is created (i.e. a match to the input preamble has been found), it can be continuously repeated and used as the reference sequence for the TED during the entire acquisition phase. Fig. 3-5 summarizes the technique.

Using the new technique, there is now no need to wait for all eight samples from the equalizer. After the first seed is found or created, phase updates can start right away. This works due to the presence of anchors.

We carried out simulations of the new acquisition technique using an initial sampling offset of  $0.5T$ . We may remark that in a real situation this offset is equally likely to be with respect to any of the ideal sequences of Table 3-1. The distribution of the samples collected before phase updates started, is plotted in Fig. 3-6.

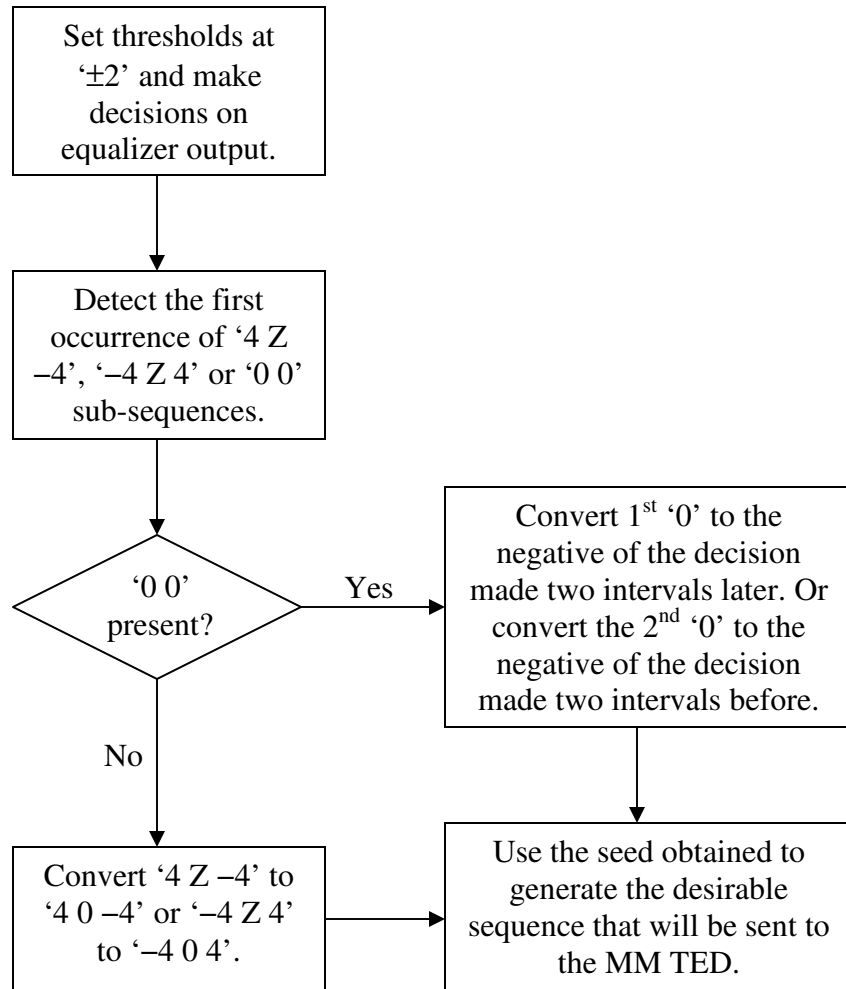


Fig. 3-5. Flow chart summarizing the process of generating the TED reference sequence for 8T preamble.

From Fig. 3-6 it can easily be calculated that the average number of samples that need to be collected before making decision on the ideal matched sequence is 3.427. This is much lower than the original direct matching scheme which requires eight samples to be collected.

The objective of the simulation is to show that it is indeed possible to reduce the mean data collection time. The mean value actually changes with changes in phase offsets. For example, if the sampling phase is of integer value, then we can deduce from Table 3-1 that the mean number of samples that need to be collected is

$$\frac{3+4+5+6}{4} = 4.5, \text{ assuming that the sampling phase takes on any of the eight integer}$$



values with equal probability so that the eight sequences in Table 3-1 have equal probability of being picked.

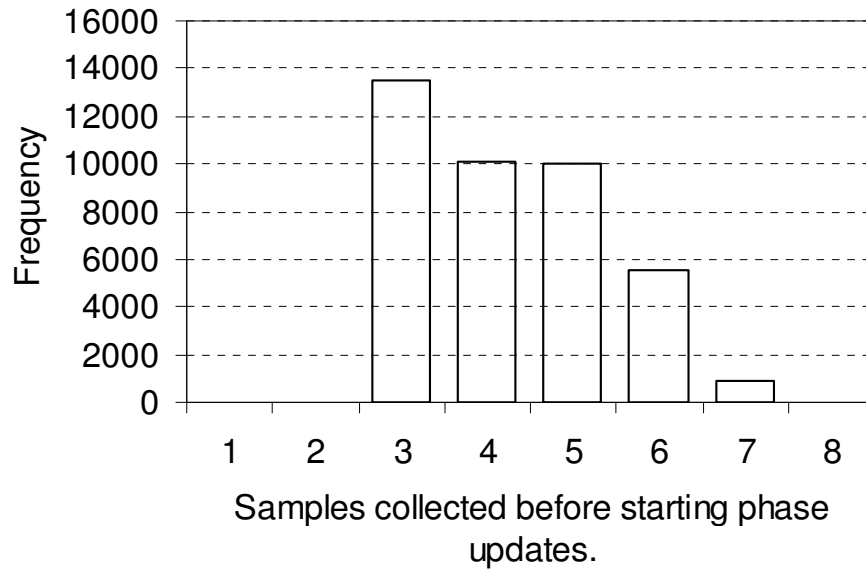


Fig. 3-6. Distribution of the number of samples that need to be collected before commencing phase updates. ( $8T$  preamble).

Fig. 3-7 shows the phase convergence plots for various SNRs. We can clearly see that the scheme works – phase updates start before eight samples have been collected and hang-ups have been completely eliminated.

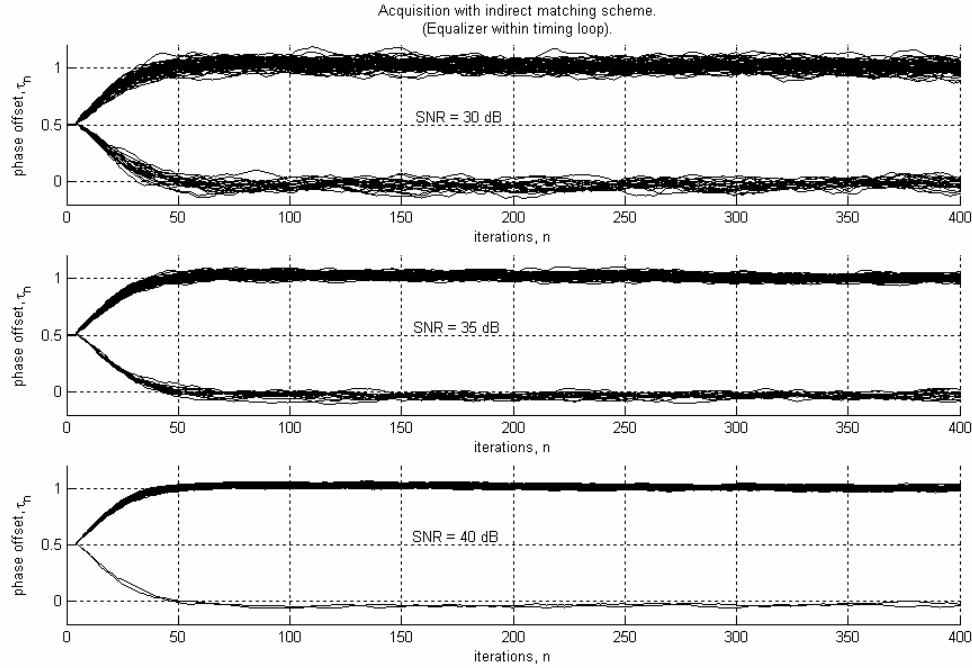


Fig. 3-7. Phase convergence for  $8T$  preamble with indirect matching scheme (80 runs);  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.

### 3.3 Probability of breakdown of indirect matching scheme for $8T$ preamble

From Fig. 3-2, we can see that the anchor points (those points that are beyond ‘ $\pm 4$ ’ and near ‘ $\pm 6$ ’) when the phase offset is  $\pm 0.5$  are very strong in the sense that they are extremely far from the detection thresholds. This means that it is almost impossible for them to be ‘blown’ by noise. But if we look at Fig. 3-1, we can see that the points at the ‘+4’s, ‘-4’s and ‘0’s are much weaker. Only the points at ‘ $\pm 6$ ’ may qualify as good anchors. Therefore we shall define the probability of breakdown as the worst case probability of the phase departing from an initial phase that is already ideal. This will require us to examine the sequences listed in Table 3-1.

From Table 3-1, we can see that due to symmetry, we need only examine the first four sequences. In order to obtain the worst case probability, we assume that the anchors at ‘+6’ and ‘-6’ are too far from the detection thresholds to be ‘blown’ by noise and we also assume that at any one time, only one of the points at ‘+4’, ‘-4’ or ‘0’ may be ‘blown’ by noise because it is extremely unlikely for multiples of them to fail. Bearing

these in mind, of the first four sequences in Table 3-1, the cases that will cause problems (the new values of the points that have failed are in the squares) are given below.

<u>Source</u>	<u>Decisions made</u>					<u>Undesirable sequences generated</u>
Sequence 1:	0	<span style="border: 1px solid black; padding: 2px;">0</span>				→ <b>-4</b> <b>0</b> <b>4</b>
	0	4	4	<span style="border: 1px solid black; padding: 2px;">0</span>	0	→ 0   4 <b>4</b> <b>0</b> -4
	0	4	4	4	<span style="border: 1px solid black; padding: 2px;">-4</span>	→ 0   4 <b>4</b> <b>0</b> -4
Sequence 2:	4	4	<span style="border: 1px solid black; padding: 2px;">0</span>	0		→ 4 <b>4</b> <b>0</b> -4
	4	4	4	<span style="border: 1px solid black; padding: 2px;">-4</span>		→ 4 <b>4</b> <b>0</b> -4
Sequence 3:	4	<span style="border: 1px solid black; padding: 2px;">0</span>	0			→ <b>4</b> <b>0</b> -4
	4	4	<span style="border: 1px solid black; padding: 2px;">-4</span>			→ <b>4</b> <b>0</b> -4

It can be seen that the seeds (bold) will generate sequences that are different from the ideal sequences. Table 3-2 shows the probability of occurrence of these problematic sequences.

<u>Source</u>	<u>Decisions made</u>					<u>Probability of occurrence</u>
Sequence 1:	0	0				→ $\int_{-2}^2 \int_{-6}^{-2} p(\eta_1, \eta_2) d\eta_2 d\eta_1$
	0	4	4	0	0	→ $\int_{-2}^2 \int_{-2}^{\infty} \int_{-4}^{\infty} \int_{-6}^{-2} \int_{-2}^2 p(\underline{\eta}) d\underline{\eta}$ where $\underline{\eta}^T = [\eta_1 \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5]$
	0	4	4	4	-4	→ $\int_{-2}^2 \int_{-2}^{\infty} \int_{-4}^{\infty} \int_{-2}^{\infty} p(\underline{\eta}) d\underline{\eta}$ where $\underline{\eta}^T = [\eta_1 \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5]$
Sequence 2:	4	4	0	0		→ $\int_{-2}^{\infty} \int_{-4}^{\infty} \int_{-6}^{-2} \int_{-2}^2 p(\eta_1, \eta_2, \eta_3, \eta_4) d\eta_4 d\eta_3 d\eta_2 d\eta_1$
	4	4	4	-4		→ $\int_{-2}^{\infty} \int_{-4}^{\infty} \int_{-2}^{\infty} \int_{-\infty}^{-2} p(\eta_1, \eta_2, \eta_3, \eta_4) d\eta_4 d\eta_3 d\eta_2 d\eta_1$
Sequence 3:	4	0	0			→ $\int_{-4}^{\infty} \int_{-6}^{-2} \int_{-2}^2 p(\eta_1, \eta_2, \eta_3) d\eta_3 d\eta_2 d\eta_1$
	4	4	-4			→ $\int_{-4}^{\infty} \int_{-2}^{\infty} \int_{-\infty}^{-2} p(\eta_1, \eta_2, \eta_3) d\eta_3 d\eta_2 d\eta_1$

Table 3-2. Probability of occurrence of sequences that will lead to departure from an initially ideal sampling phase.

The probabilities of occurrences listed in Table 3-2 are approximated using the following general equation.

$$\begin{aligned}
& P(A_1 A_2 \dots A_{N-1} A_N) \\
&= P(A_1 | A_2 \dots A_N) P(A_2 | A_3 \dots A_N) \dots P(A_{N-1} | A_N) P(A_N) \\
&\approx P(A_1 | A_2) P(A_2 | A_3) \dots P(A_{N-1} | A_N) P(A_N) \\
&= \frac{P(A_1 A_2)}{P(A_2)} \cdot \frac{P(A_2 A_3)}{P(A_3)} \cdot \dots \cdot \frac{P(A_{N-2} A_{N-1})}{P(A_{N-1})} \cdot P(A_{N-1} A_N).
\end{aligned} \tag{3-1}$$

For example,

$$\begin{aligned}
& \int_{-2}^{\infty} \int_{-4}^{\infty} \int_{-6}^{-2} \int_{-2}^2 p(\eta_1, \eta_2, \eta_3, \eta_4) d\eta_4 d\eta_3 d\eta_2 d\eta_1 \\
&= P(\eta_1 \geq -2, \eta_2 \geq -4, -2 \geq \eta_3 \geq -6, 2 \geq \eta_4 \geq -2) \\
&\approx \frac{\int_{-2}^{\infty} \int_{-4}^{\infty} p(\eta_1, \eta_2) d\eta_2 d\eta_1}{\int_{-4}^{\infty} p(\eta_2) d\eta_2} \cdot \frac{\int_{-4}^{\infty} \int_{-6}^{-2} p(\eta_2, \eta_3) d\eta_3 d\eta_2}{\int_{-6}^{-2} p(\eta_3) d\eta_3} \cdot \int_{-6}^{-2} \int_{-2}^2 p(\eta_3, \eta_4) d\eta_4 d\eta_3.
\end{aligned} \tag{3-2}$$

Finally, we average all the probabilities of occurrences to obtain the worst-case failure probability. Fig. 3-8 shows how the failure probability varies with the SNR. We see from the figure that the indirect matching scheme is very reliable.

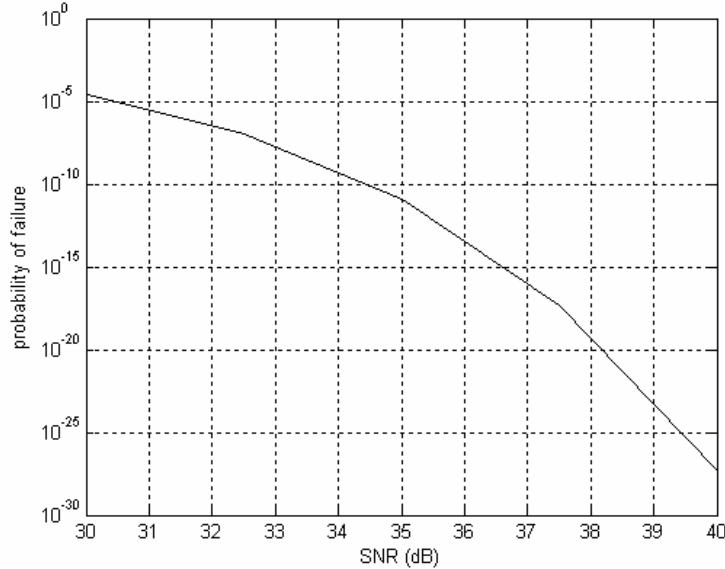


Fig. 3-8. Probability of failure of the 8T preamble indirect matching scheme.

### 3.4 Matching schemes for 10T preamble

The noiseless output time waveforms (at equalizer output) for ideal sampling phase and  $0.5T$  lead/lag sampling phase are shown in Fig. 3-9 and Fig. 3-10, respectively.

From the distribution of points shown in Fig. 3-9 and Fig. 3-10, it is easy to see that the direct/indirect matching schemes for the  $8T$  preamble can be applied to the  $10T$  preamble with only minor modifications. Now, two '+6's are placed between two '+4's and two '-6's are placed between two '-4's during the ideal sequence reconstruction. The effect of the indirect matching scheme is shown in Fig. 3-11. Fig. 3-12 shows the distribution of the number of samples that must be collected before phase updating commences. We can calculate that the average number of samples that need to be collected is 4.7313.

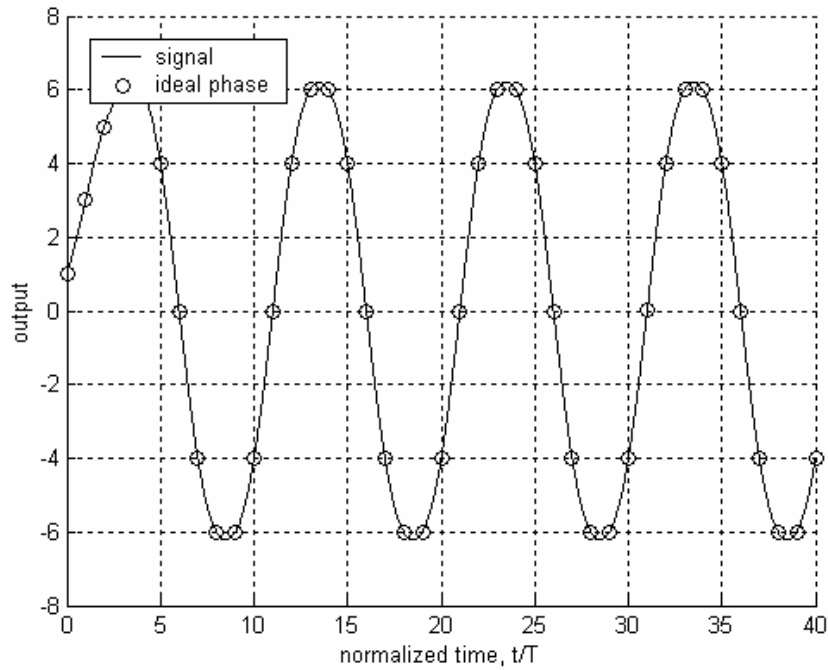


Fig. 3-9. Noiseless output with ideal sampling phase for  $10T$  preamble and target [1 2 2 1].

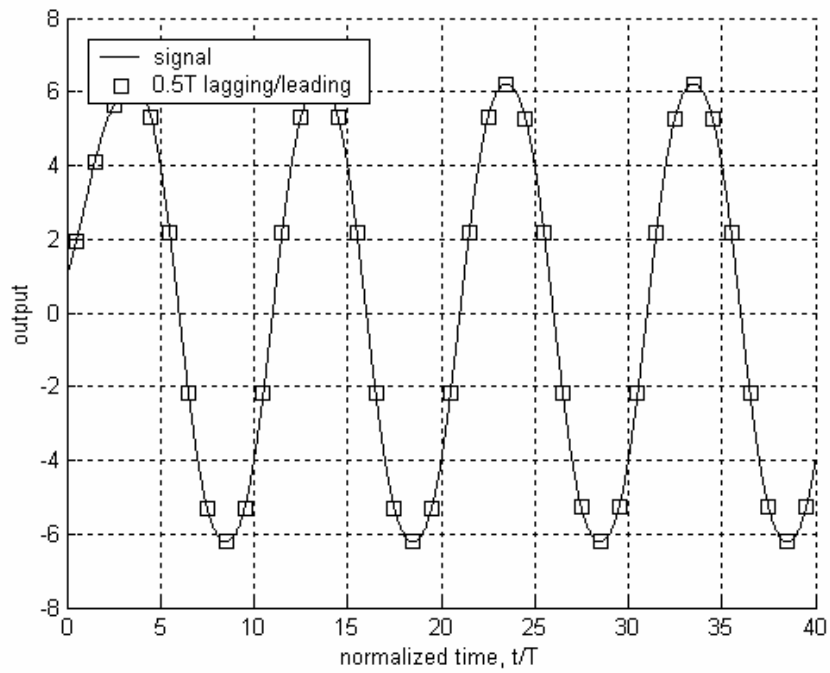


Fig. 3-10. Noiseless output with sampling phases of  $\pm 0.5T$  for  $10T$  preamble and target [1 2 2 1].

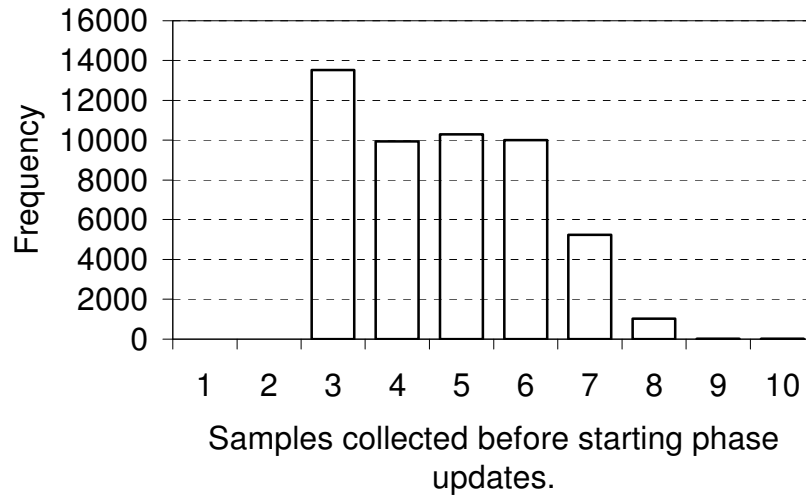


Fig. 3-12. Distribution of the number of samples that need to be collected before commencing phase updates ( $10T$  preamble).

### 3.5 Summary

In this chapter, we presented an extension of the fast acquisition technique introduced in Chapter 2. We also introduced a way to reduce the mean number of samples that needs to be collected before phase updates start. Simulations were used to show that the techniques work.

In the matching schemes presented in this chapter and in Chapter 2, we identify sequences that, when used by the MM TED, will not result in hang-ups. We should mention that the use of these techniques requires knowledge about the start of the acquisition stage so that we may know when to start collecting samples. This can be done easily by detecting for the presence of a strong readback signal.

Typical preambles used in acquisition are of lengths 100 to 150 bits. Therefore the phase should have converged after around 100 bits. From the phase convergence plots that we have plotted so far, we can see that the fast acquisition techniques allow us to achieve convergence well within 100 bits. The end of acquisition will be signaled by a sync readback signal. The sync sequence appears after the preamble and before the start of user data.

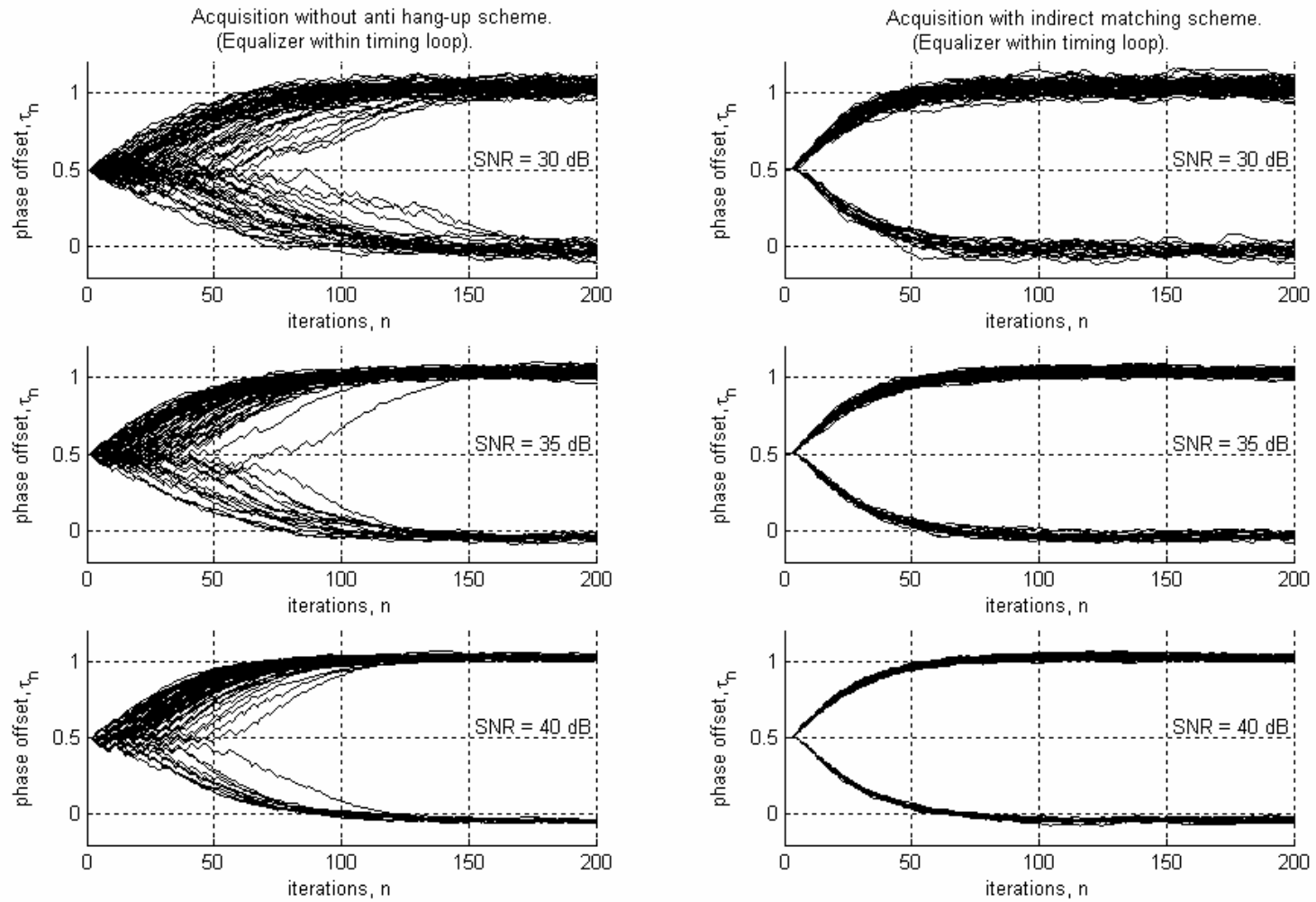


Fig. 3-11. Phase convergence for 10T preamble (80 runs);  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1.



## Chapter 4

### Fast Acquisition 2: TED Output Toggling

In this chapter we present a second fast timing acquisition scheme that is based on generating hang-up region indicators from TED outputs and then using the indicators to trigger a mechanism to push the phase out of the hang-up region. Only the  $6T$  preamble is considered.

#### 4.1 Hang-up region indicators

We know from Chapters 2 and 3 that the hang-up region occurs around midway between ideal sampling instants. Hang-up regions can also be identified by studying the timing function. Fig. 4-1 is an example of a typical timing function. The spurious zero crossings at which the slope is negative (locations 1 and 2), are unstable equilibria whereby the PLL will dwell (hang-up) for some time.

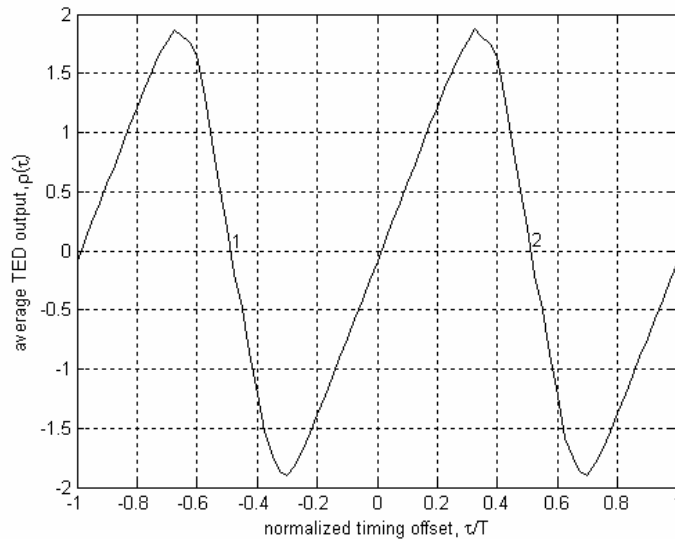


Fig. 4-1. A typical timing function whereby the hang-up regions occur around midway between ideal sampling instants.

An indicator for hang-up region should produce something very distinct for this region compared to the regions around ideal sampling phases. Within the hang-up region, the threshold decisions made are extremely erratic. On the other hand, the threshold decisions are more reliable and follow a fixed pattern when the sampling phase is near the ideal values. This contrast can be exploited to identify the occurrence of hang-up.

Since the MM TED uses threshold decisions, it should be possible to extract suitable hang-up region indicators from observations of the TED outputs. Let us examine the TED outputs that are produced (for ideal phase and phase offset of  $+0.5T$ ) when the  $6T$  preamble is used with the  $[1\ 2\ 2\ 1]$  target and thresholds ' $\pm 2$ ' (see Table 4-1 and Table 4-2). In Table 4-1, the threshold decisions are assumed to be perfectly correct, which is a reasonable assumption because the sample points are far from the detection thresholds (see Fig. 2-4).

Ideal Sampling Phase		
Equalizer Output	Threshold Decision	MM TED Output
$0 + \eta_1$	0	$4\eta_1$
$4 + \eta_2$	4	$4\eta_1$
$4 + \eta_3$	4	$4\eta_2 - 4\eta_3$
$0 + \eta_4$	0	$-4\eta_4$
$-4 + \eta_5$	-4	$-4\eta_4$
$-4 + \eta_6$	-4	$-4\eta_5 + 4\eta_6$
$0 + \eta_7$	0	$4\eta_7$
$4 + \eta_8$	4	$4\eta_7$

Table 4-1. Equalizer outputs, threshold decisions and MM TED outputs when sampling phase is ideal for  $6T$  preamble and  $[1\ 2\ 2\ 1]$  target.

In Table 4-2, the decisions made from points that are close to the  $+2$  and  $-2$  thresholds are labeled  $X$  and  $Y$ , respectively. If the previous TED output is subtracted from the current TED output, we can see from Table 4-1 that the resulting value will have a mean of zero when the sampling phase is ideal. However, if the sampling phase is offset by  $+0.5T$  and we do similar subtractions, the resulting values could become very big depending on the combination of threshold decisions. These very large values could serve as indicators of the hang-up region. Table 4-3 shows the differences of consecutive TED outputs for the sequences of Table 4-1 and Table 4-2.

Phase Offset of $+0.5T$		
Equalizer Output	Threshold Decision	MM TED Output
$2.30 + \eta_1$	$X_1$	-
$4.62 + \eta_2$	4	$4\eta_1 - X_1\eta_2 - 4.62X_1 + 9.2$
$2.30 + \eta_3$	$X_2$	$X_2\eta_2 - 4\eta_3 + 4.62X_2 - 9.2$
$-2.30 + \eta_4$	$Y_1$	$Y_1\eta_3 - X_2\eta_4 + 2.3X_2 + 2.3Y_1$
$-4.62 + \eta_5$	-4	$-4\eta_4 - Y_1\eta_5 + 4.62Y_1 + 9.2$
$-2.30 + \eta_6$	$Y_2$	$Y_2\eta_5 + 4\eta_6 - 4.62Y_2 - 9.2$
$2.30 + \eta_7$	$X_3$	$X_3\eta_6 - Y_2\eta_7 - 2.3X_3 - 2.3Y_2$
$4.62 + \eta_8$	4	$4\eta_7 - X_3\eta_8 - 4.62X_3 + 9.2$

Table 4-2. Equalizer outputs, threshold decisions and MM TED outputs when phase offset is  $+0.5T$  for  $6T$  preamble and  $[1\ 2\ 2\ 1]$  target.

Ideal Phase	$+0.5T$ Phase Offset
TED Output Differences, $\Delta\tau_n - \Delta\tau_{n-1}$	TED Output Differences, $\Delta\tau_n - \Delta\tau_{n-1}$
$-4\eta_1 + 4\eta_2 - 4\eta_3$	$X_2\eta_2 - 4\eta_3 + 4.62X_2 - 4\eta_1 + X_1\eta_2 + 4.62X_1 - 18.4$
$-4\eta_2 + 4\eta_3 - 4\eta_4$	$Y_1\eta_3 - X_2\eta_4 + 2.3X_2 + 2.3Y_1 - X_2\eta_2 + 4\eta_3 - 4.62X_2 + 9.2$
0	$-4\eta_4 - Y_1\eta_5 + 4.62Y_1 - Y_1\eta_3 + X_2\eta_4 - 2.3X_2 - 2.3Y_1 + 9.2$
$4\eta_4 - 4\eta_5 + 4\eta_6$	$Y_2\eta_5 + 4\eta_6 - 4.62Y_2 + 4\eta_4 + Y_1\eta_5 - 4.62Y_1 - 18.4$
$4\eta_5 - 4\eta_6 + 4\eta_7$	$X_3\eta_6 - Y_2\eta_7 - 2.3X_3 - 2.3Y_2 - Y_2\eta_5 - 4\eta_6 + 4.62Y_2 + 9.2$
0	$4\eta_7 - X_3\eta_8 - 4.62X_3 - X_3\eta_6 + Y_2\eta_7 + 2.3X_3 + 2.3Y_2 + 9.2$

Table 4-3. Differences of consecutive TED outputs for ideal phase and  $+0.5T$  phase offset.

In order to detect the occurrence of hang-up, we could use a simple threshold. If the magnitude of the TED output difference exceeds the threshold, then a hang-up canceling mechanism could be triggered. The problem now is to determine the optimum threshold such that the anti-hang-up mechanism has high probability of being triggered during hang-ups and low probability of being triggered after the phase has converged. Table 4-4 shows all possible combinations of the first three TED output differences in Table 4-3 (only three needs to be considered due to symmetry and cyclostationarity) for  $+0.5T$  phase offset.

Source:	$X_2\eta_2 - 4\eta_3 + 4.62X_2 - 4\eta_1 + X_1\eta_2 + 4.62X_1 - 18.4$				
Equalizer Output:	$2.30 + \eta_1$ ( $X_1$ )	$4.62 + \eta_2$	$2.30 + \eta_3$ ( $X_2$ )	TED Output Difference	Condition for Occurrence
Threshold Decisions:	0	4	0	$-4\eta_1 - 4\eta_3 - 18.4$	$-4.3 < \eta_1 < -0.3, -4.3 < \eta_3 < -0.3$
	0	4	4	$-4\eta_1 + 4\eta_2 - 4\eta_3 + 0.08$	$-4.3 < \eta_1 < -0.3, \eta_3 > -0.3$
	4	4	0	$-4\eta_1 + 4\eta_2 - 4\eta_3 + 0.08$	$\eta_1 > -0.3, -4.3 < \eta_3 < -0.3$
	4	4	4	$-4\eta_1 + 8\eta_2 - 4\eta_3 + 18.56$	$\eta_1 > -0.3, \eta_3 > -0.3$
Source:	$Y_1\eta_3 - X_2\eta_4 + 2.3X_2 + 2.3Y_1 - X_2\eta_2 + 4\eta_3 - 4.62X_2 + 9.2$				
Equalizer Output:	$4.62 + \eta_2$	$2.30 + \eta_3$ ( $X_2$ )	$-2.30 + \eta_4$ ( $Y_2$ )	TED Output Difference	Condition for Occurrence
Threshold Decisions:	4	0	0	$4\eta_3 + 9.2$	$-4.3 < \eta_3 < -0.3, 4.3 > \eta_4 > 0.3$
	4	0	-4	0	$-4.3 < \eta_3 < -0.3, \eta_4 < 0.3$
	4	4	0	$-4\eta_2 + 4\eta_3 - 4\eta_4 - 0.08$	$\eta_3 > -0.3, 4.3 > \eta_4 > 0.3$
	4	4	-4	$-4\eta_2 - 4\eta_4 - 9.28$	$\eta_3 > -0.3, \eta_4 < 0.3$
Source:	$-4\eta_4 - Y_1\eta_5 + 4.62Y_1 - Y_1\eta_3 + X_2\eta_4 - 2.3X_2 - 2.3Y_1 + 9.2$				
Equalizer Output:	$2.30 + \eta_3$ ( $X_2$ )	$-2.30 + \eta_4$ ( $Y_1$ )	$-4.62 + \eta_5$	TED Output Difference	Condition for Occurrence
Threshold Decisions:	0	0	-4	$-4\eta_4 + 9.2$	$-4.3 < \eta_3 < -0.3, 4.3 > \eta_4 > 0.3$
	0	-4	-4	$4\eta_3 - 4\eta_4 + 4\eta_5 - 0.08$	$-4.3 < \eta_3 < -0.3, \eta_4 < 0.3$
	4	0	-4	0	$\eta_3 > -0.3, \eta_4 > 0.3$
	4	-4	-4	$4\eta_3 + 4\eta_5 - 9.28$	$\eta_3 > -0.3, \eta_4 < 0.3$

Table 4-4. TED output differences for various combinations of threshold decisions when phase offset is  $+0.5T$ .

When the  $6T$  preamble is used with the  $[1\ 2\ 2\ 1]$  target, any three consecutive decisions made by the threshold detector should assume one of the six forms:  $[0\ 4\ 4]$ ,  $[4\ 4\ 0]$ ,  $[4\ 0\ -4]$ ,  $[0\ -4\ -4]$ ,  $[-4\ -4\ 0]$  or  $[-4\ 0\ 4]$ . All other sequences apart from these six imply the presence of decision errors. If we look at Table 4-4, we can see that all sequences that **do not** assume any of the six forms will result in the TED output differences to have very large mean values. Therefore the difference between two consecutive TED outputs can indeed serve as a good means to trigger a correction mechanism. Table 4-5 shows all the large TED output differences collected from Table 4-4, (rearranged in the order of decreasing magnitude of mean).

	TED Output Differences with large means	Mean
1.	$-4\eta_1 + 8\eta_2 - 4\eta_3 + 18.56$	18.56
2.	$-4\eta_1 - 4\eta_3 - 18.4$	-18.4
3.	$-4\eta_2 - 4\eta_4 - 9.28$	-9.28
4.	$4\eta_3 + 4\eta_5 - 9.28$	-9.28
5.	$4\eta_3 + 9.2$	9.2
6.	$-4\eta_4 + 9.2$	9.2

Table 4-5. TED output differences with large means and their means.

In order to detect these large differences, we can use a single positive threshold at  $+\zeta$  and compare the absolute value of the TED differences with this threshold. If the threshold is crossed, a correction mechanism can be triggered. This is described by the following equation:

$$|\Delta\tau_n - \Delta\tau_{n-1}| \begin{matrix} \text{trigger} \\ > \\ < \\ \text{no trigger} \end{matrix} \zeta. \quad (4-1)$$

We shall define the probability of failed detection as the probability of the trigger threshold not being crossed when hang-up occurs i.e. following the appearance of **any** of the six TED output differences listed in Table 4-5.

$$\begin{aligned}
P_{failed} &= P(|\Delta\tau_n - \Delta\tau_{n-1}| < \zeta \mid \text{hang-up occurs}) \\
&= \frac{P(|\Delta\tau_n - \Delta\tau_{n-1}| < \zeta, \text{ hang-up occurs})}{P(\text{hang-up occurs})} \\
&= \frac{\sum_{k=1}^6 P(|\Delta\tau_n - \Delta\tau_{n-1}|_k < \zeta, HU_k)}{\sum_{l=1}^6 P(HU_l)} \\
&= \sum_{k=1}^6 \left[ P(|\Delta\tau_n - \Delta\tau_{n-1}|_k < \zeta \mid HU_k) \frac{P(HU_k)}{\sum_{l=1}^6 P(HU_l)} \right],
\end{aligned} \tag{4-2}$$

where  $|\Delta\tau_n - \Delta\tau_{n-1}|_k$  is the  $k^{\text{th}}$  difference listed in Table 4-5 and  $HU_k$  is the condition for its occurrence. The term  $P(|\Delta\tau_n - \Delta\tau_{n-1}|_k < \zeta \mid HU_k)$  in Equation (4-2) will be approximated as shown in Table 4-6, whereby we have assumed that the conditional terms dropped are weakly correlated with the random variable  $|\Delta\tau_n - \Delta\tau_{n-1}|_k$ . The term

$\frac{P(HU_k)}{\sum_{l=1}^6 P(HU_l)} = P_{norm,k}$  is the relative probability of occurrence of the  $k^{\text{th}}$  TED difference.

The relative probability of occurrence for each TED output difference is obtained by calculating the probability of occurrence according to the conditions listed in Table 4-4 and then normalizing with the sum of the probabilities of occurrences of all the cases listed in Table 4-5. For example, the probability of occurrence of the first difference in Table 4-5 is given by

$$P(HU_1) = \frac{1}{2\pi\sqrt{\det(C)}} \int_{-0.3}^{+\infty} \int_{-0.3}^{+\infty} \exp\left[-\frac{1}{2} \cdot (\eta_1 \ \eta_3) \cdot C^{-1} \cdot \begin{pmatrix} \eta_1 \\ \eta_3 \end{pmatrix}\right] d\eta_1 d\eta_3 \tag{4-3}$$

where  $C = \begin{bmatrix} \phi_{\eta\eta}(0) & \phi_{\eta\eta}(2) \\ \phi_{\eta\eta}(2) & \phi_{\eta\eta}(0) \end{bmatrix}$  is the covariance matrix and  $\det(C)$  is the determinant of  $C$ .

The relative probability of the first entry is thus given by

$$P_{norm,1} = \frac{P(HU_1)}{\sum_{l=1}^6 P(HU_l)}. \tag{4-4}$$

Table 4-7 lists the relative probability of occurrence of the six differences shown in Table 4-5 when the SNR is 30 dB.

$k$	$ \Delta\tau_n - \Delta\tau_{n-1} _k$	$HU_k$	Approximation of $P( \Delta\tau_n - \Delta\tau_{n-1} _k < \zeta   HU_k)$
1	$-4\eta_1 + 8\eta_2 - 4\eta_3 + 18.56$	$\eta_1 > -0.3,$ $\eta_3 > -0.3$	$P(-4\eta_1 + 8\eta_2 - 4\eta_3 + 18.56 < \zeta   \eta_1 > -0.3)$
2	$-4\eta_1 - 4\eta_3 - 18.4$	$-4.3 < \eta_1 < -0.3,$ $-4.3 < \eta_3 < -0.3$	$P(-4\eta_1 - 4\eta_3 - 18.4 < \zeta   -4.3 < \eta_1 < -0.3)$
3	$-4\eta_2 - 4\eta_4 - 9.28$	$\eta_3 > -0.3,$ $\eta_4 < 0.3$	$P(-4\eta_2 - 4\eta_4 - 9.28 < \zeta   \eta_3 > -0.3)$
4	$4\eta_3 + 4\eta_5 - 9.28$	$\eta_3 > -0.3,$ $\eta_4 < 0.3$	$P(4\eta_3 + 4\eta_5 - 9.28 < \zeta   \eta_4 < 0.3)$
5	$4\eta_3 + 9.2$	$-4.3 < \eta_3 < -0.3,$ $4.3 > \eta_4 > 0.3$	$P(4\eta_3 + 9.2 < \zeta   -4.3 < \eta_3 < -0.3)$
6	$-4\eta_4 + 9.2$	$-4.3 < \eta_3 < -0.3,$ $4.3 > \eta_4 > 0.3$	$P(-4\eta_4 + 9.2 < \zeta   4.3 > \eta_4 > 0.3)$

Table 4-6. Approximations of the term  $P(|\Delta\tau_n - \Delta\tau_{n-1}|_k < \zeta | HU_k)$ .

	TED Output Differences with large means	Relative Probability of Occurrence
1.	$-4\eta_1 + 8\eta_2 - 4\eta_3 + 18.56$	0.3241
2.	$-4\eta_1 - 4\eta_3 - 18.4$	0.0323
3.	$-4\eta_2 - 4\eta_4 - 9.28$	0.3068
4.	$4\eta_3 + 4\eta_5 - 9.28$	0.3068
5.	$4\eta_3 + 9.2$	0.0150
6.	$-4\eta_4 + 9.2$	0.0150

Table 4-7. Relative probability of occurrence of TED output differences that have large means (SNR = 30 dB).

The probability of failed detection may then be easily calculated using (4-2), the approximations of Table 4-6 and the relative probabilities of Table 4-7.

We also define the probability of false alarm as the probability that the correction mechanism is triggered after the phase has converged or when there are no decision errors.

$$\begin{aligned}
P_{fa} &= P(|\Delta\tau_n - \Delta\tau_{n-1}| > \zeta \mid \text{no hang-up occurs}) \\
&= \frac{P(|\Delta\tau_n - \Delta\tau_{n-1}| > \zeta, \text{no hang-up occurs})}{P(\text{no hang-up occurs})} \\
&= \frac{\sum_k P(|\Delta\tau_n - \Delta\tau_{n-1}|_k > \zeta, \overline{HU_k})}{\sum_l P(\overline{HU_l})},
\end{aligned} \tag{4-5}$$

where  $\overline{HU_l}$  is the condition for the occurrence of a certain  $l^{\text{th}}$  TED output difference that has small mean e.g. for a phase offset of  $0.5T$ , from Table 4-4, the TED output difference  $-4\eta_1 + 4\eta_2 - 4\eta_3 + 0.08$  has small mean with condition of occurrence being  $(-4.3 < \eta_1 < -0.3, \eta_3 > -0.3)$ .  $\overline{HU} = \bigcup_l \overline{HU_l}$  represents the entire probability space for no hang-ups and the condition  $(-4.3 < \eta_1 < -0.3, \eta_3 > -0.3)$  is therefore a subset of this probability space.

From the TED output differences for ideal phase listed in Table 4-3 and from the TED output differences for no decision errors listed in Table 4-4 (ignoring those TED output differences that are exactly zero because they have zero probabilities of crossing the trigger threshold), we can see that the mean is zero or very close to zero when there are no decision errors. Therefore we shall assume that the random variable  $|\Delta\tau_n - \Delta\tau_{n-1}|_k$  has zero mean and compute (4-5) using suitable approximations similar to the method used to evaluate the probability of failed detection. Fig. 4-2 shows the probability of failed detection and false alarm when the SNR is 30 dB.



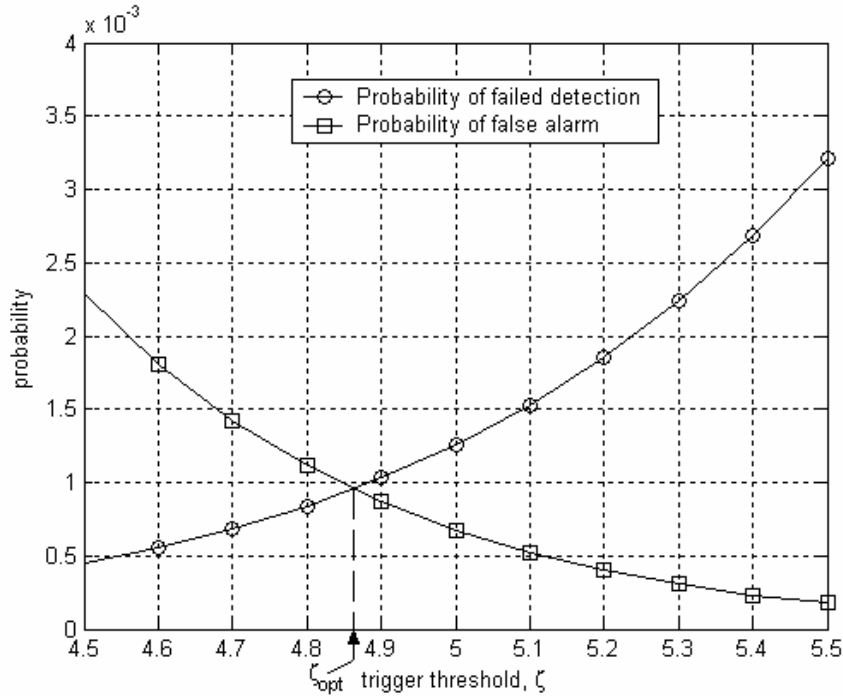


Fig. 4-2. Probability of failed detection of hang-up indicator and probability of false alarm; [1 2 2 1] target; 6T preamble; SNR = 30 dB.

We shall set the optimum trigger threshold at the intersection shown in Fig. 4-2 where both probabilities are equally small. This optimum threshold is  $\zeta_{opt} = +4.86$  Fig. 4-3 shows how this optimum threshold varies with SNR. Generally, the optimum threshold decreases slightly with increasing SNR. However, the probabilities of failed detection and false alarm will be very low at high SNR even if the threshold is placed slightly offset from the optimum value. Therefore, for simplicity, we can just choose +5 as the single threshold and it will work well for various SNR.

Typical phase convergence and TED output difference plots for initial phase offsets of  $+0.5T$  when there is no anti-hang-up mechanism are shown in Fig. 4-4. Horizontal lines at  $\pm 4.86$  have been drawn on the difference plot to indicate the optimum trigger threshold. From the difference plot, we can see that there are indeed large spikes in the hang-up region that may be distinguished easily from the small spikes in the non-hang-up region. Large spikes are also produced whenever there is a change in direction of phase convergence.

Observe also that the smaller of the large spikes are mostly negative. This can be explained by the relative probabilities shown in Table 4-7. From Table 4-7, we can see that negative large spikes have much higher probability to be generated by TED differences that have means of  $-9.28$  than by TED differences that have means of  $-18.4$ . The opposite is true for positive large spikes: they have much higher probability to be generated by TED differences that have means of  $+18.56$  than by TED differences that have means of  $+9.2$ .

Observe also that the trigger thresholds are effective in separating the large spikes that result from reversals in phase convergence direction, from the small spikes that are harmless.

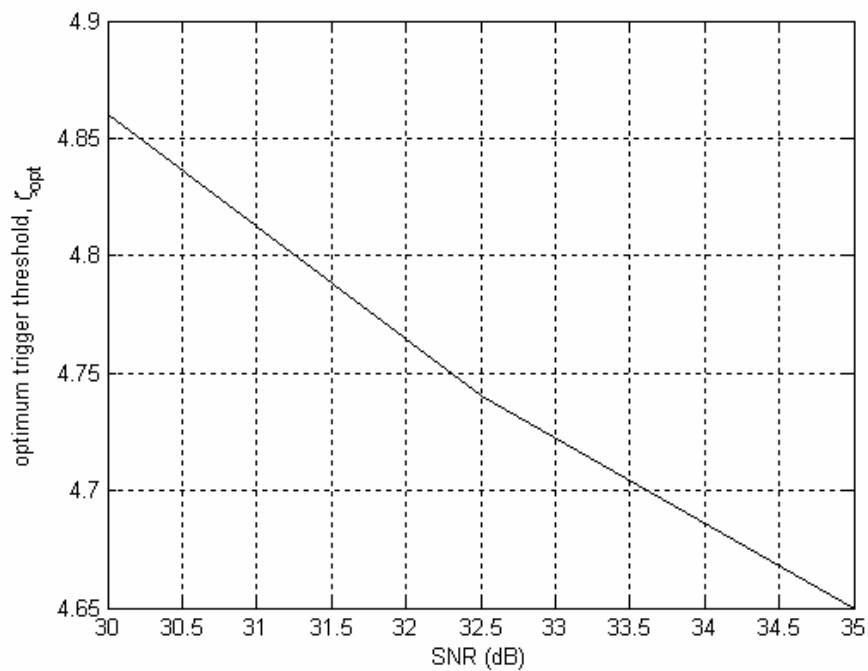


Fig. 4-3. Variation of optimum trigger threshold with SNR; [1 2 2 1] target;  $6T$  preamble.

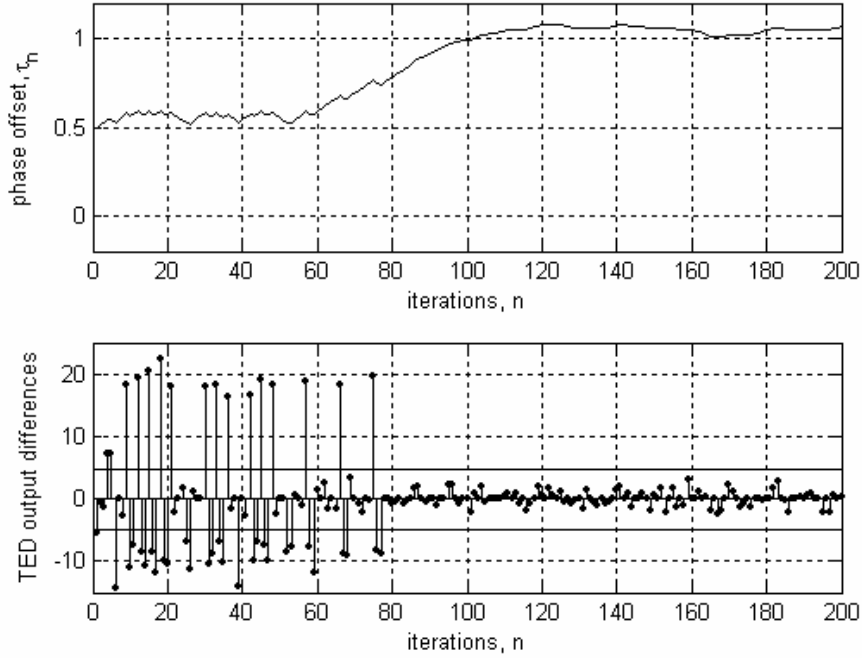


Fig. 4-4. Typical phase convergence and TED output difference plots in the presence of hang-up and with no anti-hang-up mechanism;  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%; initial phase offset =  $0.5T$ ; SNR = 30 dB.

#### 4.2 Anti-hang-up mechanism based on toggling TED outputs

An anti-hang-up mechanism may be activated whenever the magnitudes of TED output differences exceed the trigger threshold at  $+\zeta_{opt}$ . This is described by the equation,

$$\begin{array}{c}
 \text{trigger} \\
 > \\
 |\Delta\tau_n - \Delta\tau_{n-1}| > \zeta_{opt} \\
 < \\
 \text{no trigger}
 \end{array} \quad (4-6)$$

where  $\Delta\tau_0$  is the first TED output and  $\Delta\tau_{-1} = 0$ .

There are several configurations of TED outputs  $(\Delta\tau_n, \Delta\tau_{n-1})$  that could result in triggering and they are listed in Fig. 4-5. From Fig. 4-5, we see that it will be good if there is a flip of the current TED output sign so that it is always the same as the previous TED output sign. (This will only apply when the anti-hang-up mechanism is triggered)

and so will not affect tracking, where it is perfectly alright if consecutive TED outputs have opposite signs). Thus, whenever the trigger threshold  $\zeta_{opt}$  is crossed, we propose to modify the TED output according to

$$\Delta\tau_n \leftarrow \text{sgn}(\Delta\tau_{n-1}) \cdot |\Delta\tau_n|. \quad (4-7)$$

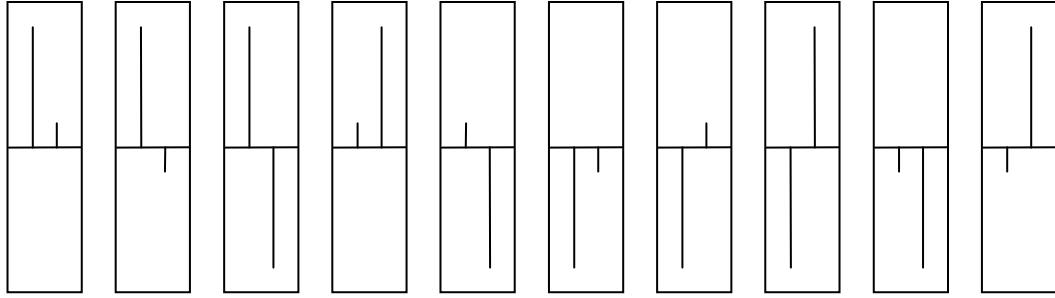


Fig. 4-5. TED outputs  $(\Delta\tau_n, \Delta\tau_{n-1})$  that will result in triggering of anti-hang-up mechanism.

However the flipping rule described by (4-7) could still produce undesirable phase updates and these are illustrated in Fig. 4-6. In order to eradicate these problems, there is a need to preserve an initial sign of flipping and use it for all future flips. This initial sign of flipping shall be referred to as  $\lambda$  and it is determined based on the first trigger. Suppose the first trigger occurs at instant  $m$ ,

$$\lambda = \begin{cases} \text{sgn}(\Delta\tau_m), & \text{if } m = 0 \\ \text{sgn}(\Delta\tau_{m-1}), & \text{if } m \neq 0 \end{cases} \quad (4-8)$$

The new flipping equation is then given by

$$\Delta\tau_n := \lambda \cdot |\Delta\tau_n|, \quad n \geq 0. \quad (4-9)$$

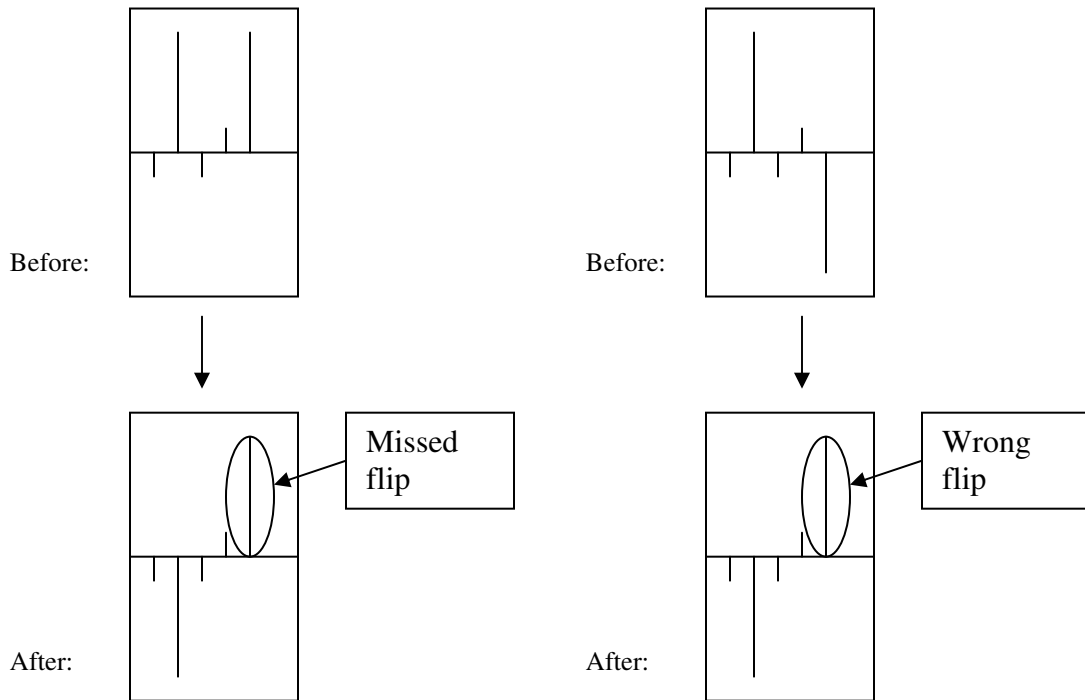


Fig. 4-6. Undesirable phase updates associated with the flipping rule of equation (4-7).

Figs. 4-7 and 4-8 show the results of implementing the flipping rule of equation (4-9). It is obvious that hang-ups may be completely eliminated.

### 4.3 Summary

In this chapter we presented a second fast acquisition technique that is based on the toggling of TED outputs to eliminate hang-ups. Both theory and simulations show that hang-up region indicators can be obtained reliably and used for triggering an anti-hang-up mechanism. We implemented such an anti-hang-up scheme and showed via simulations that it is effective even at low SNRs.

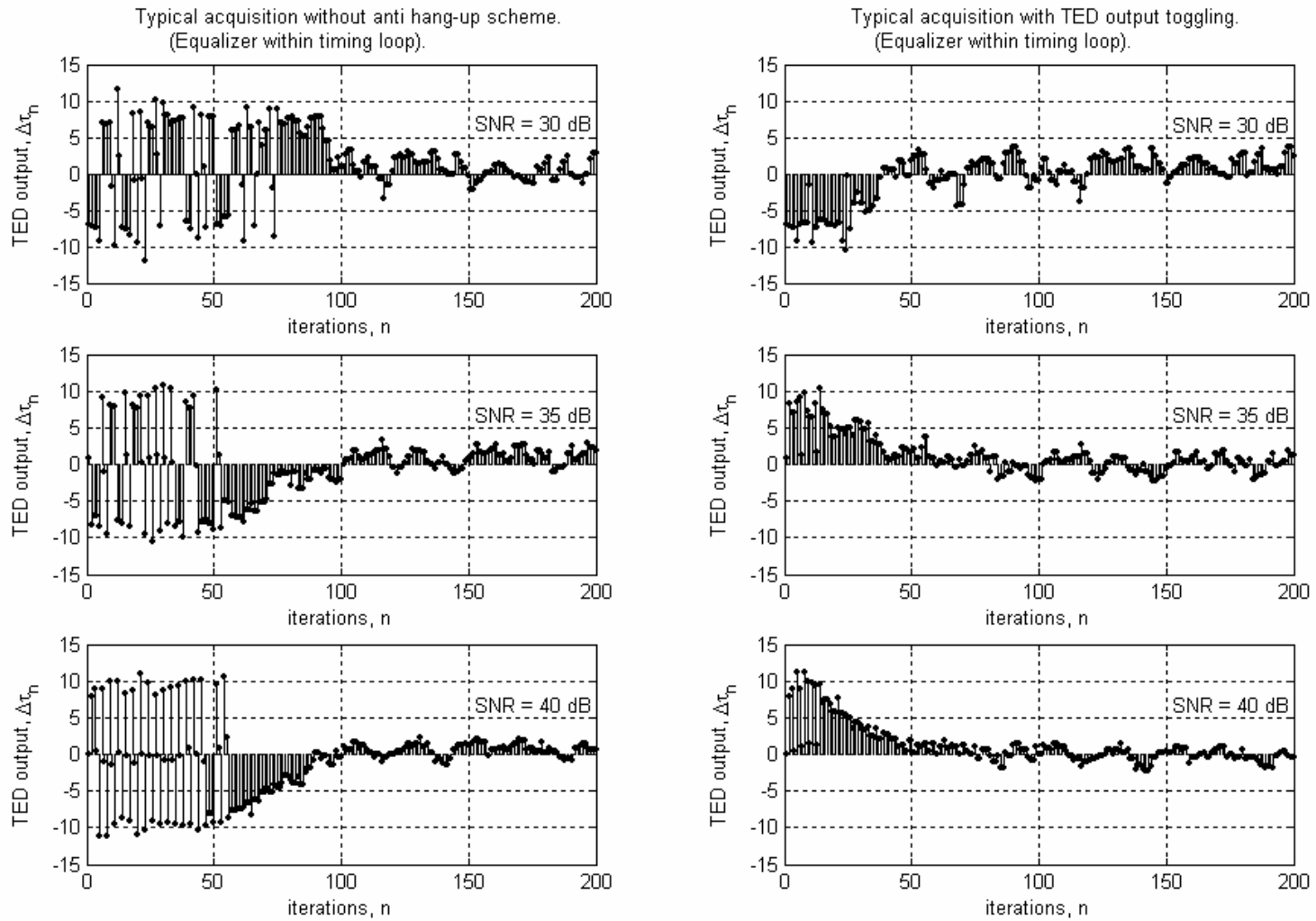


Fig. 4-7. Typical TED outputs for  $6T$  preamble;  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%, phase offset =  $0.5T$ .

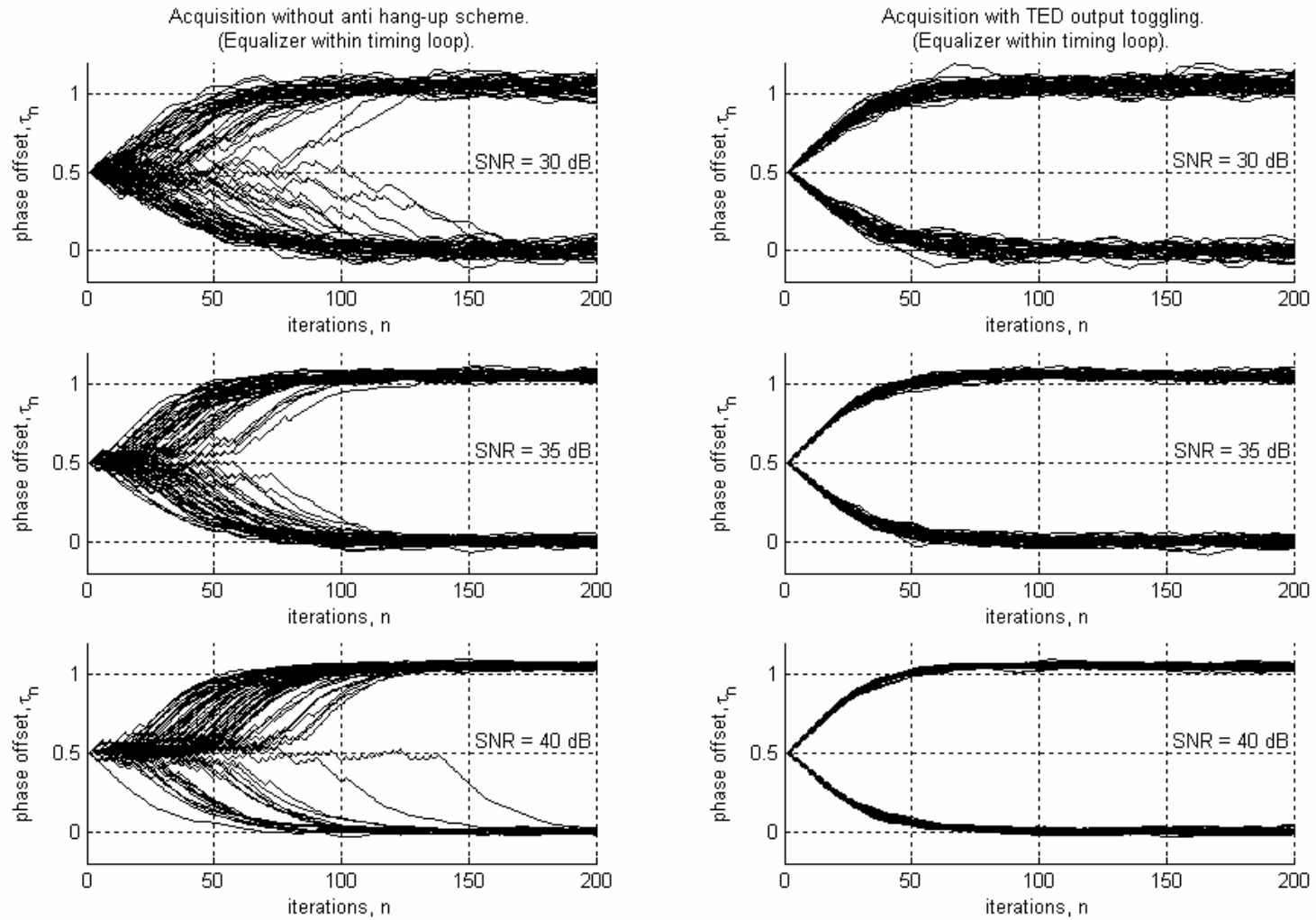


Fig. 4-8. Phase convergence for 6T preamble (80 runs);  $\alpha = 2.0 \times 10^{-3}$ ;  $\rho = 0.5 \times 10^{-5}$ ; frequency offset = 0.1%.

## Chapter 5

### Joint Timing Recovery and Symbol Detection 1:

#### Joint Timing-ISI Trellis 1

In this chapter we present a way of doing joint timing recovery and symbol detection on a joint timing-intersymbol-interference (ISI) trellis. A soft output Viterbi algorithm (SOVA [14]) will be developed for the trellis to generate soft decisions.

#### 5.1 Introduction

Fig. 5-1(a) shows the conventional approach to bit detection in a magnetic recording system that uses iteratively decodable codes like Turbo [15] and LDPC [27][28] codes. Iterations, when used, are limited to symbol detection and decoding processes. The tasks of timing recovery and iterative detection/decoding are done separately. In some systems, iterations are limited to only the decoding process. This approach, as depicted in Fig. 5-1(a), works well at high SNRs but is doomed to failure at low SNRs where the powerful capacity-approaching codes operate in. This is because the probability of cycle-slips during tracking increases steeply with decreasing SNRs [23] and if the timing recovery scheme does not make use of the error-control coding to improve its performance, it will suffer from rampant cycle-slips. When a cycle-slip happens, the symbol sequence received by the decoder will be shifted by one or more symbols either forward or backward. This is because the PLL will continue tracking after a cycle-slip has occurred, but it will be tracking a sampling phase that is offset from its true value by an integer multiple of the bit interval. Obviously this must be corrected, otherwise the symbols sent to the decoders will be fraught with errors and thus greatly erode the power of the error-control coding scheme.

In this chapter, we present a way of correcting cycle-slips by merging the timing recovery and symbol detection blocks into a single block and subsequently making it exchange extrinsic information with the decoding block (see Fig. 5-1(b)). This setup will



ensure that the power of the error-control coding is harnessed to aid in both timing recovery and symbol detection.

We shall be using a simple channel model and will be creating a simplified timing error model which will be used to approximate an actual random walk timing error process. We do these in order to reduce the computational intensity of simulations. No generality is lost though as the method can easily be applied to more complicated systems.

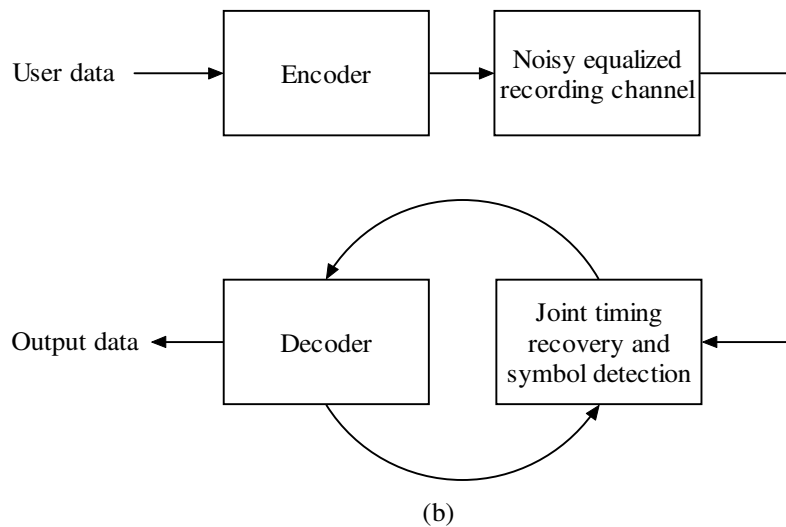
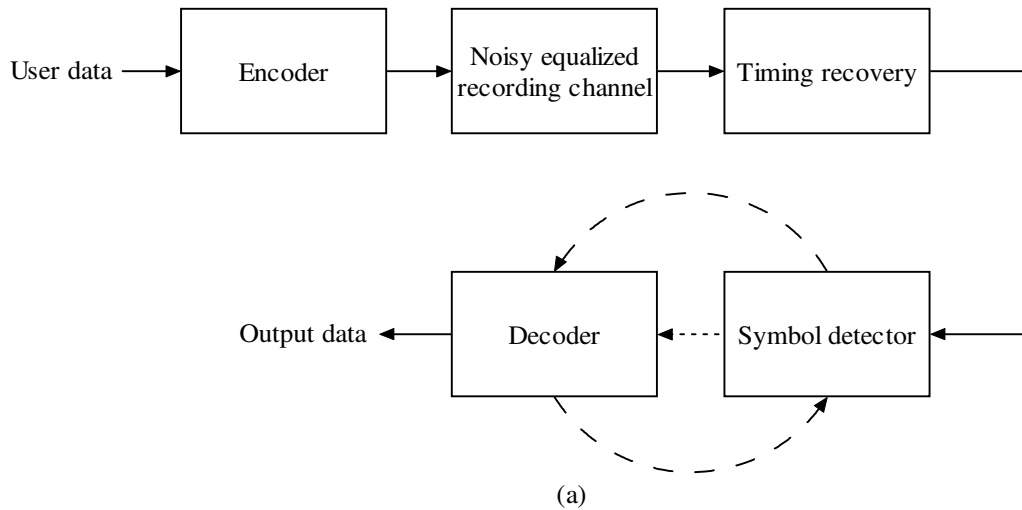


Fig. 5-1. (a) Conventional approach to timing recovery, symbol detection and error-control coding. (b) Iterative timing recovery with joint timing recovery and symbol detection.

## 5.2 Quantized timing error model

Consider a simple model for the readback signal with timing errors from a digital storage system:

$$r(t) = \sum_n a_n h(t - nT - \varepsilon_n) + \eta(t) \quad (5-1)$$

where  $a_n \in \{\pm 1\}$  is the  $n$ -th binary input code bit,  $h(t)$  is the channel bit response,  $T$  is the bit interval,  $\varepsilon_n$  is the timing uncertainty associated with the  $n$ -th bit and  $\eta(t)$  is additive white Gaussian noise. A low pass filter is used to remove out-of-band noise from the readback waveform  $r(t)$ , producing a bandlimited waveform which is then sampled at the bit rate  $1/T$  to obtain samples  $y_k$ . The optimum sampling instants are  $\{kT + \varepsilon_k\}$ . A timing recovery scheme is able to yield estimates,  $\hat{\varepsilon}_k$ , of the timing offset  $\varepsilon_k$  before sampling. The received sampled signal will therefore be given by

$$y_k = \sum_n a_n h(kT - nT + \hat{\varepsilon}_k - \varepsilon_n) + \eta_k. \quad (5-2)$$

Assuming that both  $\varepsilon_n$  and  $\hat{\varepsilon}_k$  are slowly time-varying, (5-2) may be written as

$$\begin{aligned} y_k &= \sum_n a_n h(kT - nT + \tau_k) + \eta_k \\ &= \sum_l a_{k-l} h(lT + \tau_k) + \eta_k \end{aligned} \quad (5-3)$$

where  $\tau_k = \hat{\varepsilon}_k - \varepsilon_n$  is the timing error. Ideally, the timing error  $\tau_k$  should be treated as a real-valued random process for any timing error model to be accurate. However, in order to facilitate the setting up of a joint timing-ISI trellis, the values of  $\tau_k$  are assumed to be integer multiples of  $\frac{T}{Q}$ , where  $Q$  is the number of timing quantization levels in the bit interval  $T$ :

$$\tau_k \in \left\{ \dots, \frac{-3T}{Q}, \frac{-2T}{Q}, \frac{-T}{Q}, 0, \frac{T}{Q}, \frac{2T}{Q}, \frac{3T}{Q}, \dots \right\}. \quad (5-4)$$

If  $Q$  is large, not much accuracy will be lost. We also assume that  $\tau_k$  is a Markov chain. This is a valid assumption because, given that  $Q$  is large, it is intuitively clear that any discrete time random process may be approximated by a Markov chain of sufficiently

large memory. However, for the sake of simplicity, we shall assume that the memory length of the Markov chain is one. Extension to Markov chains of larger memory is trivial. We also assume that the Markov chain is governed by the following transition probabilities

$$P(\tau_k | \tau_{k-1}) = \begin{cases} \delta & \text{if } \tau_k = \tau_{k-1} + \frac{T}{Q} \\ \delta & \text{if } \tau_k = \tau_{k-1} - \frac{T}{Q} \\ 1 - 2\delta & \text{if } \tau_k = \tau_{k-1} \\ 0 & \text{otherwise.} \end{cases} \quad (5-5)$$

Fig. 5-2 shows the resulting timing trellis, assuming that the initial phase offset is zero. From Fig. 5-2, we see that the timing trellis will continue to grow unbounded as time goes on.

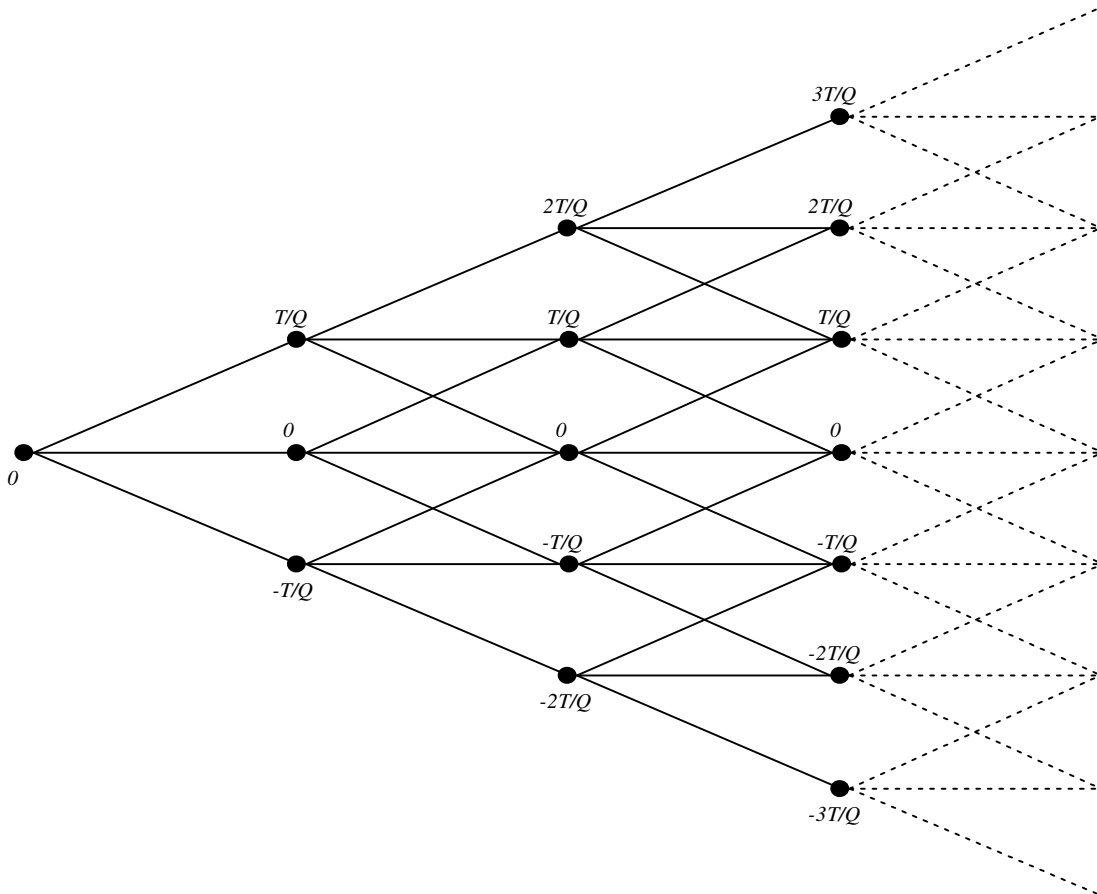


Fig. 5-2. Trellis for the timing error  $\tau_k$  modeled as a Markov chain of memory one.

### 5.3 Joint timing-ISI trellis

For the sake of simplicity, we shall model  $h(t)$  as

$$h(t) = \begin{cases} \frac{4}{3} \sin^2\left(\frac{2\pi(t+T)}{6T}\right) & \text{for } -T \leq t \leq 2T \\ 0 & \text{otherwise.} \end{cases} \quad (5-6)$$

This pulse, shown in Fig. 5-3, has an ISI length of two when there is a timing offset, and it spans over three symbol intervals. The ISI coefficients, when timing is perfect, are given by  $h(0)=1$  and  $h(T)=1$ . (Generalization to other pulses with different waveforms and ISI lengths is again trivial). With imperfect timing and consequently a channel memory length of two, the resulting ISI trellis will have four states. This ISI trellis is shown in Fig. 5-4.

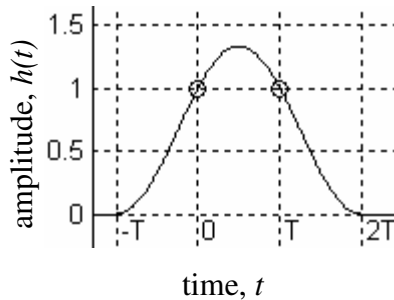


Fig. 5-3. A pulse with an ISI length of two that spans three symbol intervals.

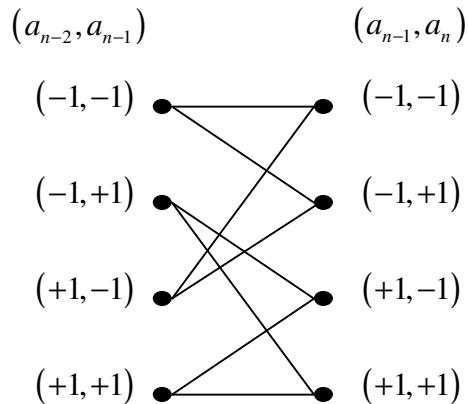


Fig. 5-4. Standard ISI trellis for the pulse of Fig. 5-3 with imperfect timing.

We shall assume that a prior timing acquisition stage is able to give us an initial timing offset of zero, or close to zero, and that before the start of the message block, at least two  $-1$  bits are transmitted so that the initial ISI state is  $(-1, -1)$ . The timing trellis of Fig. 5-2 may now be combined with the ISI trellis of Fig. 5-4 to create a joint timing-ISI trellis.

In order to understand how this joint timing-ISI trellis may be constructed, let us consider an example of a realization of the process  $\tau_k$ , for  $Q = 5$ . This is shown in Fig. 5-5 (sampling instants are marked on the abscissa by bullets), where the input bits  $\{a_{-1}, a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$  are set arbitrarily to be  $\{-1, -1, +1, +1, -1, +1, -1, -1\}$ . The superposition of the received pulses is also plotted using thick line.

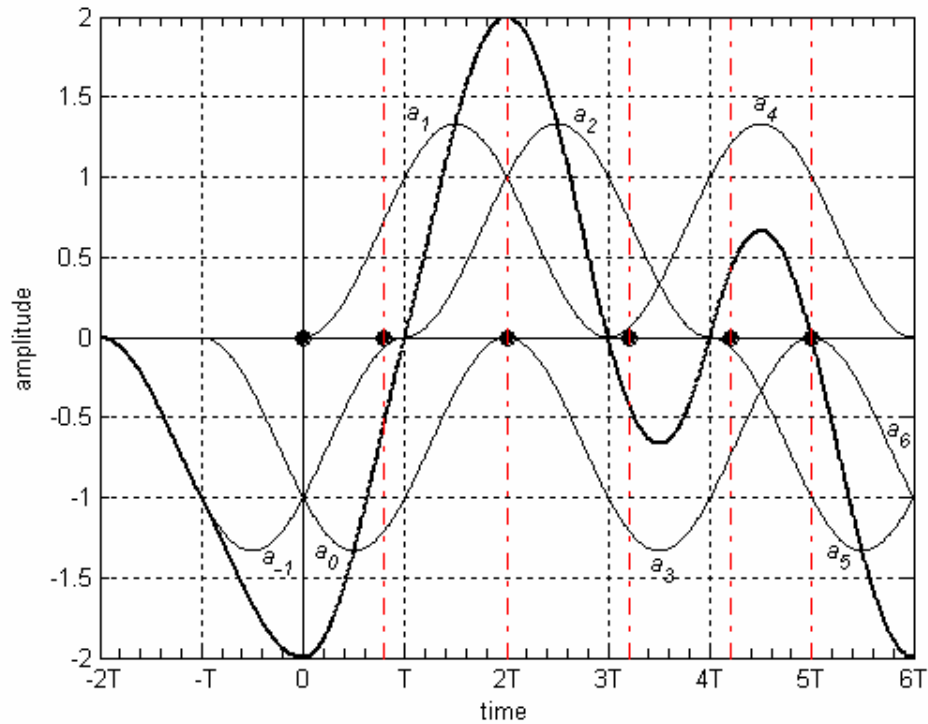


Fig. 5-5. An example of a realization of the process  $\tau_k$  for some given input bits and  $Q = 5$ . Sampling instants are marked on the abscissa by bullets. Resultant noiseless waveform is drawn using thick line.

The first bullet at time instant 0 represents the initial starting sampling phase. The ISI state is  $(a_{-1}, a_0)$ . Moving to the next bullet at instant  $0.8T$ , we see that in order to compute the noiseless value, we need to know the values of  $a_{-1}$ ,  $a_0$ , and  $a_1$ . Therefore

the destination ISI state will be given by  $(a_0, a_1)$ . The same reasoning may be applied to the ISI state for the next (third) bullet, which will be  $(a_1, a_2)$ . The fourth bullet is interesting because in order to compute the noiseless value for this sampling instant, we need to know  $a_2$ ,  $a_3$ , and  $a_4$ . Therefore the destination ISI state should be  $(a_3, a_4)$  rather than  $(a_2, a_3)$ . Similarly, the ISI state corresponding to the fifth bullet is  $(a_4, a_5)$ . The sixth bullet is also interesting because only  $a_4$  and  $a_5$  are required for computing the noiseless value. Thus the destination ISI state is  $(a_4, a_5)$ . For the sampling realization of Fig. 5-5, the ISI state transitions will therefore be  $(a_{-1}, a_0) \rightarrow (a_0, a_1) \rightarrow (a_1, a_2) \rightarrow (a_3, a_4) \rightarrow (a_4, a_5) \rightarrow (a_4, a_5)$ .

In summary, the rules used in assigning the ISI states to nodes in the joint timing-ISI trellis are listed in Table 5-1, where  $g$  is an integer and  $gT$  (e.g.  $gT = 0$ ) is a possible value for the timing offset at the  $k^{\text{th}}$  sampling instant. Further, it is assumed that at the  $k^{\text{th}}$  sampling instant, the ISI state is derived from the bits  $a_{n-2}$  and  $a_{n-1}$ . (Note that  $k$  and  $n$  may not be equal and there is no need to know the exact relationship between  $k$  and  $n$  when implementing the trellis). With these assignment rules, the noiseless value for each branch can always be calculated from the ISI and timing states of the two nodes that the branch links. These noiseless values may be stored in memory beforehand and retrieved upon request. With these in mind, we may now construct the joint timing-ISI trellis. The first two trellis sections of this combined trellis are shown in Fig. 5-6.

$k^{\text{th}}$ sampling instant $\rightarrow$ $(k+1)^{\text{th}}$ sampling instant	
Timing state transition	ISI state transition
$gT \rightarrow \left(gT + \frac{T}{Q}\right)$	$(a_{n-2}, a_{n-1}) \rightarrow (a_n, a_{n+1})$
$\left(gT + \frac{T}{Q}\right) \rightarrow gT$	$(a_{n-2}, a_{n-1}) \rightarrow (a_{n-2}, a_{n-1})$
All other timing state transitions	$(a_{n-2}, a_{n-1}) \rightarrow (a_{n-1}, a_n)$

Table 5-1. ISI state transitions corresponding to given timing state transitions for the  $k^{\text{th}}$  to  $(k+1)^{\text{th}}$  sampling instants.

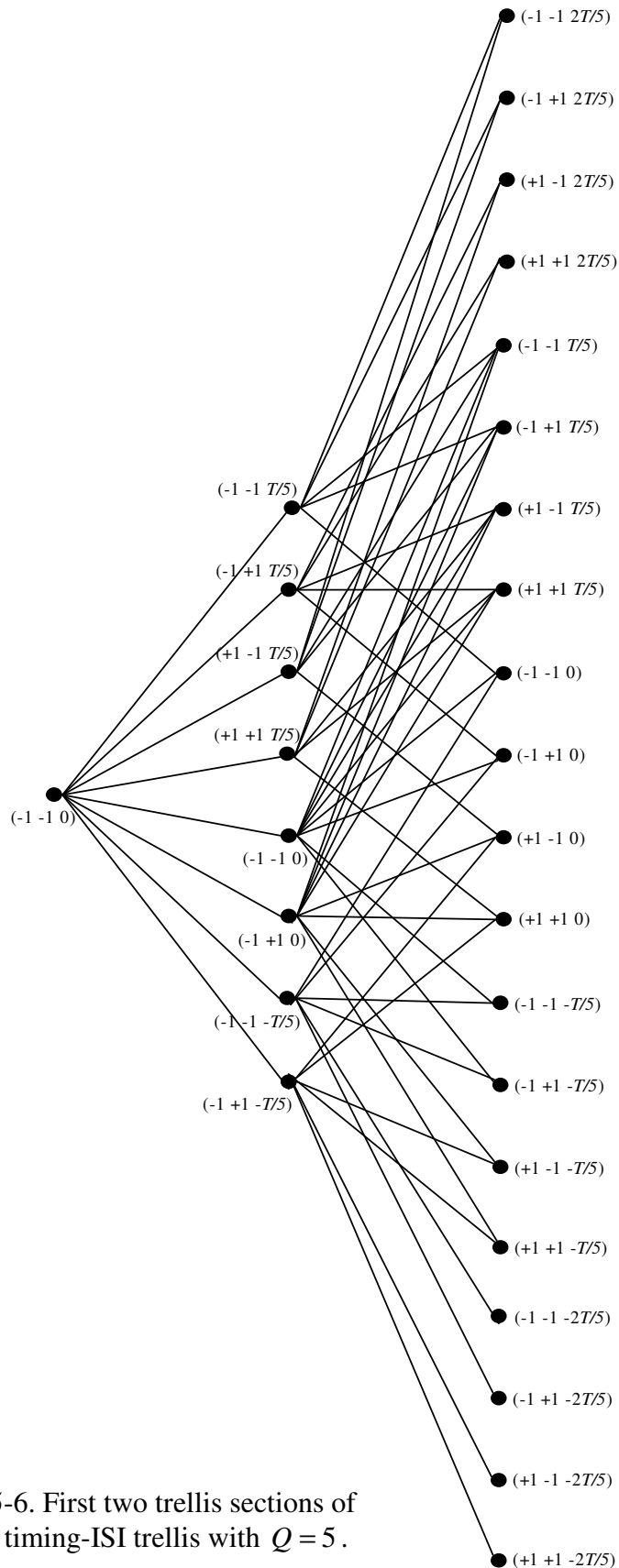


Fig. 5-6. First two trellis sections of joint timing-ISI trellis with  $Q = 5$ .

From Fig. 5-6, we see that each state of the combined trellis is described by two portions – an ISI state portion and a timing state portion. Transitions from one state to another in the combined trellis are thus determined by two factors – the input bits and the timing transitions. Transitions that are allowed between two nodes are marked by branches. We shall refer to the input bits that will cause a given transition as the “bits carried by the branch” that link the two nodes corresponding to that transition. All paths through the trellis (made up by joining branches together) that converge into any node should carry the same number of bits. In order for this to be true, it is not possible for all branches to carry the same number of bits – all branches that proceed from a timing offset of  $gT$ , where  $g$  is an integer, to a timing offset of  $gT + \frac{T}{Q}$  carry two bits; all branches that proceed from a timing offset of  $gT + \frac{T}{Q}$  to a timing offset of  $gT$  carry no bits; all other branches carry one bit each. On top of these, the bits carried by a branch are also related to the ISI state transitions. Table 5-2 summarizes the rules for assigning the bits carried by branches, and we have again assumed that at the  $k^{\text{th}}$  sampling instant, the ISI state is derived from the bits  $a_{n-2}$  and  $a_{n-1}$ .

$k^{\text{th}}$ sampling instant $\rightarrow$ $(k+1)^{\text{th}}$ sampling instant		
Timing state transition	ISI state transition	Bits carried by branch
$gT \rightarrow \left(gT + \frac{T}{Q}\right)$	$(a_{n-2}, a_{n-1}) \rightarrow (a_n, a_{n+1})$	$a_n a_{n+1}$
$\left(gT + \frac{T}{Q}\right) \rightarrow gT$	$(a_{n-2}, a_{n-1}) \rightarrow (a_{n-2}, a_{n-1})$	$\phi$ (no bits)
All other timing state transitions	$(a_{n-2}, a_{n-1}) \rightarrow (a_{n-1}, a_n)$	$a_n$

Table 5-2. Rules for assigning bits to a branch.

The trellis of Fig. 5-6 may be written in a compact form as in Fig. 5-7 that shows five sections of the joint timing-ISI for  $Q = 5$ . The bits carried by each branch have been marked using  $a_n$ . The branches that carry no bits are indicated using the symbol ‘ $\phi$ ’. The sampled realization of Fig. 5-5 is also marked on the figure by the thick path.



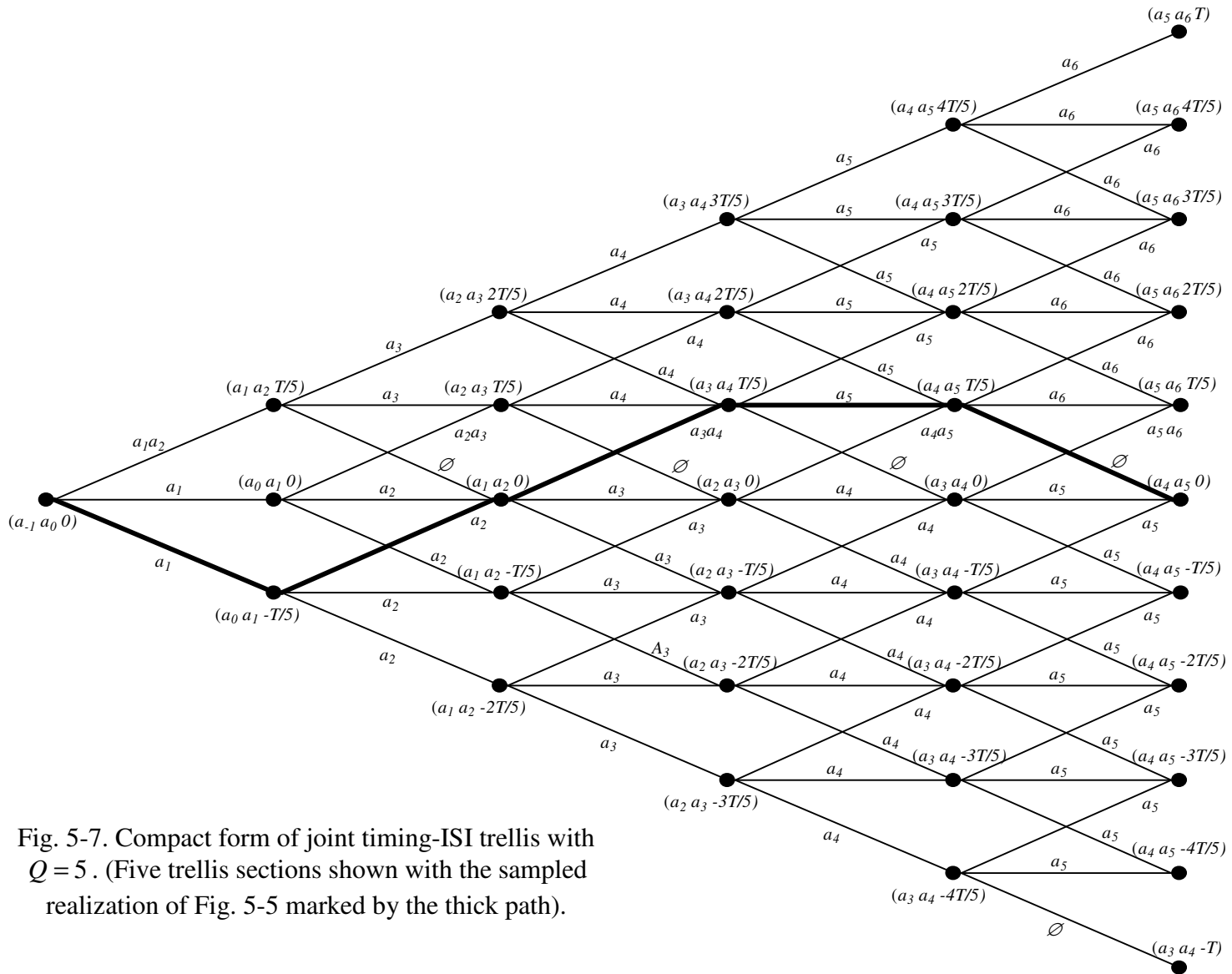


Fig. 5-7. Compact form of joint timing-ISI trellis with  $Q = 5$ . (Five trellis sections shown with the sampled realization of Fig. 5-5 marked by the thick path).

## 5.4 Viterbi algorithm on the joint timing-ISI trellis

With the trellis set up as in Fig. 5-7, the Viterbi algorithm may now be applied. The likelihood of transition from one node to another is given by the expression

$$\gamma = P(a) \cdot \frac{1}{\sqrt{2\pi\sigma_\eta^2}} e^{-\frac{(y-s)^2}{2\sigma_\eta^2}} \cdot P(\tau) \quad (5-7)$$

where  $P(a)$  is the probability of the input bits that will cause the transition,  $y$  is the noise corrupted received signal sample,  $s$  is the noiseless value of the received signal sample,  $\sigma_\eta^2$  is the variance of the additive white Gaussian noise and  $P(\tau)$  is the probability of timing state transition and is given by (5-5).

The likelihood of any path through the trellis is obtained by the product of all the likelihoods of the transitions that constitute the path. The likelihood of a certain  $l^{th}$  path that is obtained after  $I$  samples and which carries  $N$  bits, is therefore given by

$$\begin{aligned} p_l &= \left[ \prod_{n=1}^N P(\hat{a}_n^{(l)}) \right] \left[ \prod_{i=1}^I \frac{1}{\sqrt{2\pi\sigma_\eta^2}} e^{-\frac{(r_i - s_i^{(l)})^2}{2\sigma_\eta^2}} P_i^{(l)}(\delta) \right] \\ &= \left[ \prod_{n=1}^N \frac{1}{1 + e^{-\hat{a}_n^{(l)} LLR(a_n)}} \right] \left[ \prod_{i=1}^I \frac{1}{\sqrt{2\pi\sigma_\eta^2}} e^{-\frac{(r_i - s_i^{(l)})^2}{2\sigma_\eta^2}} P_i^{(l)}(\delta) \right] \end{aligned} \quad (5-8)$$

where  $\hat{a}_n^{(l)}$  is the decision made on the  $n^{th}$  bit if the  $l^{th}$  path is the final survivor path and

$LLR(a_n) = \log\left(\frac{P(a_n = +1)}{P(a_n = -1)}\right)$  is the log-likelihood ratio of the  $n^{th}$  input bit, which could

be provided by a soft-output decoder.

In the standard Viterbi algorithm,  $LLR(a_n)$  is equals to zero because it is equally likely for  $a_n$  to be +1 or -1. Whenever, two or more paths merge into a node, the path with the higher likelihood will be selected as the survivor path and all the other paths will be discarded. There is a problem though when the Viterbi algorithm is applied to the trellis of Fig. 5-6 – the trellis grows unbounded with time and so the number of states that needs to be tracked will also grow unbounded with time. A solution to tackle this

problem will be to track only a certain fixed number of states e.g. 256. This means that whenever the total number of states exceeds 256, the likelihoods of the various paths will be compared and the 256 paths with the highest likelihoods will be retained, while the rest will be discarded.

When the algorithm is implemented, the number of samples collected should be greater than the number of input bits that has been sent through the channel. This will ensure that the samples will always be able to cover all the input bits with high probability irregardless of the timing error sequence. This also means that a sequence of  $-1$  (or  $+1$ ) bits should be padded to the end of the input bits and transmitted together with it.

At the end of the trellis, the path with the maximum likelihood will be selected and the sequence of bits carried by it will be output as the hard decisions. Fig. 5-8 shows a typical phase error tracking where the timing model of Section 5.2, with  $Q$  set to 5, is used to approximate the phase error fluctuations derived from a random walk with standard deviation  $\sigma_w = 0.02T$ . SNR<sup>1</sup> is 8 dB. The value of  $\delta$  may be obtained by equating the variance of the timing random walk with the variance of our timing model. The variance of our timing model is given by

$$\begin{aligned}\sigma_m^2 &= \delta \cdot \left(\frac{T}{Q}\right)^2 + \delta \cdot \left(-\frac{T}{Q}\right)^2 + (1-2\delta) \cdot 0^2 \\ &= \frac{2\delta T^2}{Q^2}.\end{aligned}\tag{5-9a}$$

Equating this to  $\sigma_w^2$ , we obtain

$$\delta = \frac{1}{2} \left(\frac{\sigma_w}{T}\right)^2 Q^2.\tag{5-9b}$$

For  $\sigma_w = 0.02T$  and  $Q = 5$ , the value of  $\delta$  is 0.005.

---

<sup>1</sup> In this chapter, SNR is defined as  $10 \log_{10} \frac{E_b}{2\sigma_\eta^2}$ , where  $\sigma_\eta^2$  is the noise variance and  $E_b$  is the user bit energy which is assumed to be 1.

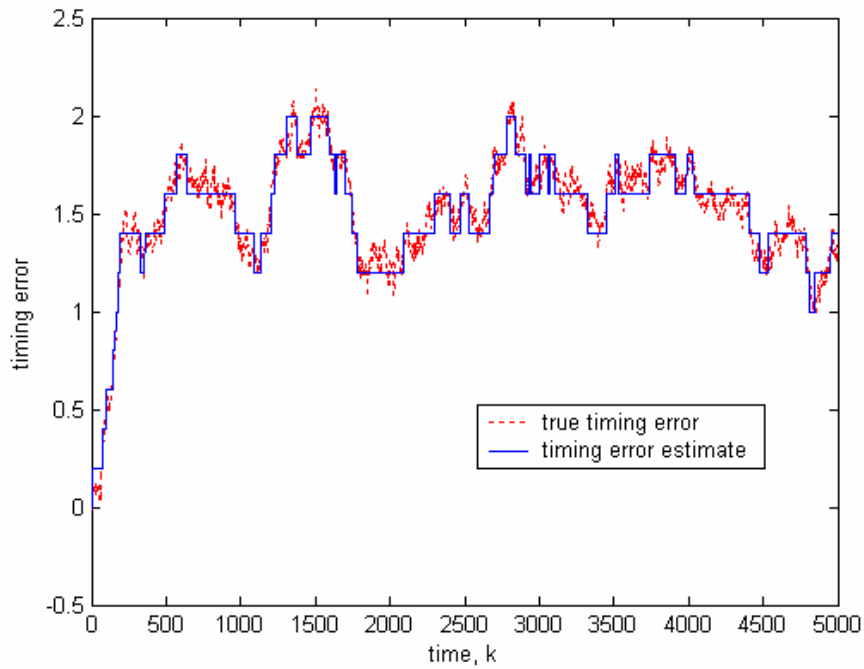


Fig. 5-8. Typical phase error tracking;  
 $\sigma_w = 0.02T$  ;  $Q = 5$  ;  $\delta = 0.005$  ; SNR = 8 dB.

The bit error rate (BER) plot is shown in Fig. 5-9. From Fig. 5-9, it is obvious that the scheme will only work well when the SNR is high ( $> 5$  dB). This is because at low SNR, cycle-slips will happen. We show later, by simulations, that cycle-slips may be corrected when *a priori* probabilities about the input bits are fed back from powerful iterative decoders. However, for the iterations to work, the joint timing-ISI detector must be able to supply soft decisions to the decoders. This will be described in the next section

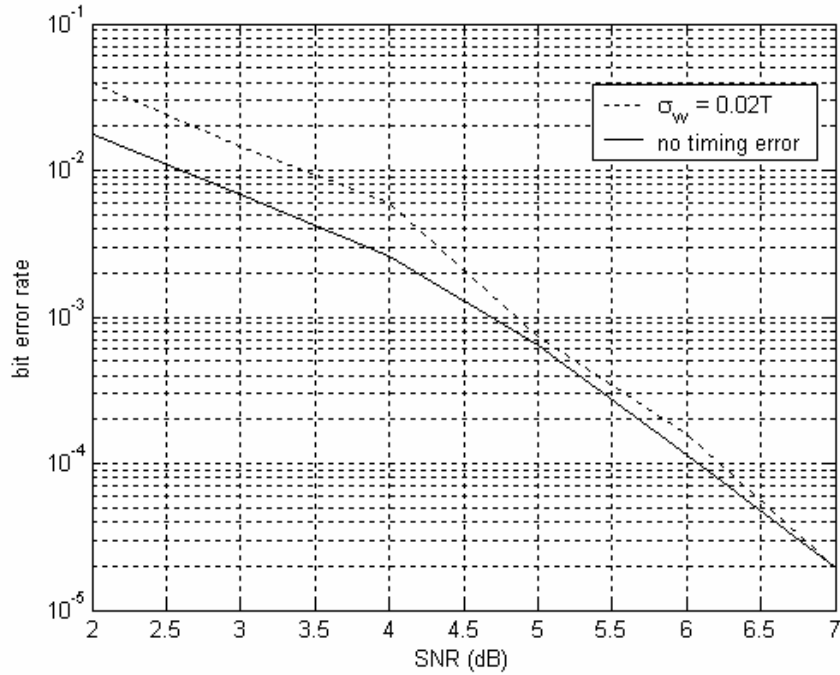


Fig. 5-9. Bit error probability for implementing the Viterbi algorithm on the joint timing-ISI trellis of Fig. 5-6.

### 5.5 SOVA on the joint timing-ISI trellis and iterative decoding

From the joint timing-ISI trellis of Fig. 5-7, it can be seen that every path that converges into a node will carry the same number bits. This fact is made use of in developing SOVA for detection using the joint trellis.

Let us consider a simple case of three paths converging into one node as shown in Fig. 5-10. Let this occur after  $I$  samples are collected and let each path carry  $N$  number of bits (these bits are candidate hard decisions and they shall be referred to simply as “hard decisions”). Also, let Path 1 be the path with the highest likelihood among the three (this means that it will be the survivor path). The notation is explained as follows:

$\hat{a}_n^{(l)}$ , where  $n \leq N$  : Hard decision made on the  $n^{\text{th}}$  bit,  $a_n$ , by the  $l^{\text{th}}$  path converging into the node.

$\hat{q}_n^{(l)}$ , where  $n \leq N$  : Probability that the decision made on  $a_n$  by the  $l^{\text{th}}$  path is a correct decision.

$p_l$  (see eqn 5-8) : Likelihood of Path  $l$ .

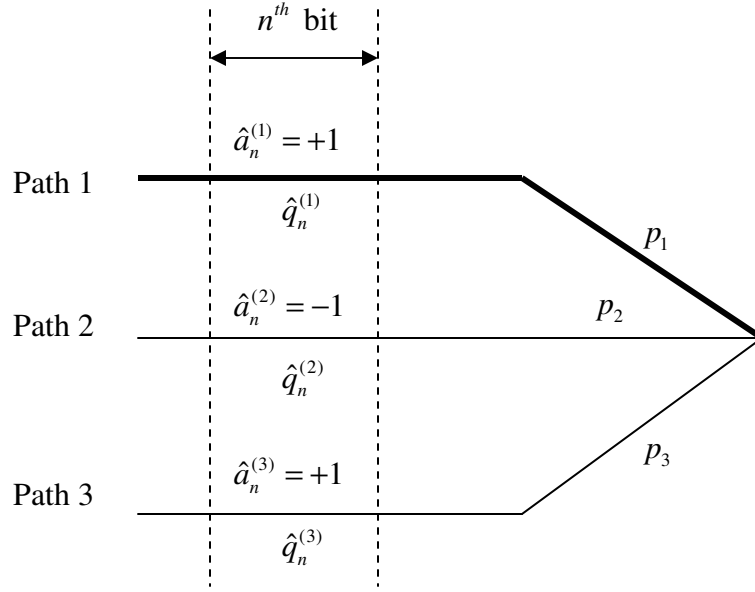


Fig. 5-10. Three paths converging into a single node.  
(Survivor path marked by thick line).

The probabilities of Paths 1, 2 and 3 being the correct path are given by

$$P(\text{Path 1 is correct}) = \frac{p_1}{p_1 + p_2 + p_3}, \quad (5-10a)$$

$$P(\text{Path 2 is correct}) = \frac{p_2}{p_1 + p_2 + p_3}, \quad (5-10b)$$

$$P(\text{Path 3 is correct}) = \frac{p_3}{p_1 + p_2 + p_3}. \quad (5-10c)$$

With Path 1 being selected,  $\hat{q}_n^{(1)}$  may be updated according to

$$\begin{aligned} \hat{q}_n^{(1)} \leftarrow & P(\text{Path 1 is correct}) \times \hat{q}_n^{(1)} + \\ & P(\text{Path 2 is correct}) \times (1 - \hat{q}_n^{(2)}) + \\ & P(\text{Path 3 is correct}) \times \hat{q}_n^{(3)}, \quad n = 1, \dots, N \end{aligned} \quad (5-11a)$$

$$\Rightarrow \hat{q}_n^{(1)} \leftarrow \frac{p_1}{p_1 + p_2 + p_3} \hat{q}_n^{(1)} + \frac{p_2}{p_1 + p_2 + p_3} (1 - \hat{q}_n^{(2)}) + \frac{p_3}{p_1 + p_2 + p_3} \hat{q}_n^{(3)}, \quad n = 1, \dots, N. \quad (5-11b)$$

This recursion may be performed on the log-likelihood ratio

$$\hat{L}_n^{(l)} = \log \left( \frac{\hat{q}_n^{(l)}}{1 - \hat{q}_n^{(l)}} \right), \quad (5-12)$$

so that it becomes

$$\hat{L}_n^{(1)} \leftarrow \log \left( \frac{\frac{P_1}{1 + e^{-\hat{L}_n^{(1)}}} + \frac{P_3}{1 + e^{-\hat{L}_n^{(3)}}} + \frac{P_2}{1 + e^{\hat{L}_n^{(2)}}}}{\frac{P_1}{1 + e^{\hat{L}_n^{(1)}}} + \frac{P_3}{1 + e^{\hat{L}_n^{(3)}}} + \frac{P_2}{1 + e^{-\hat{L}_n^{(2)}}}} \right), \quad n = 1, \dots, N. \quad (5-13)$$

Generalization to multiple paths that converge into any node may be made according to the following update equation,

$$\hat{L}_n^{(c)} \leftarrow \log \left( \frac{\sum_i \frac{P_{si}}{1 + e^{-\hat{L}_{sn}^{(i)}}} + \sum_j \frac{P_{dj}}{1 + e^{\hat{L}_{dn}^{(j)}}}}{\sum_i \frac{P_{si}}{1 + e^{\hat{L}_{sn}^{(i)}}} + \sum_j \frac{P_{dj}}{1 + e^{-\hat{L}_{dn}^{(j)}}}} \right), \quad n = 1, \dots, N, \quad (5-14)$$

where path  $c$  is the survivor path and the subscript  $s$  is for paths for which  $\hat{a}_n^{(l)} = \hat{a}_n^{(c)}$  and the subscript  $d$  is for paths for which  $\hat{a}_n^{(l)} \neq \hat{a}_n^{(c)}$ . The soft decisions along a survivor path  $c$  are then given by

$$\hat{\Lambda}_n^{(c)} = \hat{a}_n^{(c)} \hat{L}_n^{(c)}, \quad n = 1, \dots, N. \quad (5-15)$$

The procedure is summarized as follows:

- 
- Step 1:** Start from the  $(-1, -1, 0)$  timing-ISI state.
  - Step 2:** Whenever a sample is received, extend the trellis (see Fig. 5-6 & Fig. 5-7). Set the reliabilities of all the decisions (if there are any) carried by the extension branches to infinities.
  - Step 3:** Select the survivor paths and update the reliabilities along the survivor paths according to (5-14).
  - Step 4:** Repeat Steps 1 to 3 until the end of the trellis.
  - Step 5:** Compute the soft decisions for each survivor paths using (5-15).
- 

At the end of the trellis, suppose a total of  $J$  samples are collected, the soft decisions along path  $l$  may be written as

$$\hat{\Lambda}_n^{(l)} = \log \frac{P(a_n^{(l)} = +1 | r_1^J, \tau_J = \varphi^{(l)})}{P(a_n^{(l)} = -1 | r_1^J, \tau_J = \varphi^{(l)})} \quad (5-16)$$

where  $\varphi^{(l)}$  is the final timing error for path  $l$ . The extrinsic information to be sent for decoding by iterative decoders, e.g. turbo decoders, may then be evaluated as

$$L_{ext,n}^{(l)} = \log \frac{P(a_n^{(l)} = +1 | r_1^J, \tau_J = \varphi^{(l)})}{P(a_n^{(l)} = -1 | r_1^J, \tau_J = \varphi^{(l)})} - \log \left( \frac{P(a_n = +1)}{P(a_n = -1)} \right) \quad (5-17)$$

There is no way to know what the final timing offset is and consequently there is no way to know which path's extrinsic information should be sent to the decoders. At low SNRs, the survivor path with the highest likelihood will not necessarily have a final timing offset that matches the actual final timing offset. The maximum likelihood (ML) path may actually be a cycle-slipped path. Therefore we cannot simply choose the ML path, discard the rest of the survivor paths, and send its extrinsic information into the decoders.

One possible solution to this problem is to choose several survivor paths and send their extrinsic information to the decoders for decoding. This can be done in parallel by using a bank of multiple decoders, each decoder working on the extrinsic information supplied by each survivor path. There is no need to choose every survivor path. In fact a total of five should be sufficient – the ML path plus four other survivor paths with final timing offsets that differ from the ML path's final timing offset by an integer multiple of  $T$  on either side. This choice is important for correcting cycle-slips.

Each of the iterative decoders will then produce extrinsic information which may be sent back to the SOVA joint timing-ISI detector. Again, we do not know which decoder's extrinsic information should be used. Therefore all the log-likelihood ratios supplied by the various decoders will be averaged using equal weights and sent back to the SOVA detector, which will use it as *a priori* information.

The number of iterations in the decoders should be large (preferably around ten). After a few outer iterations where extrinsic information from the decoders has been fed back to the SOVA detector a few times, only the extrinsic information supplied by the ML path need be sent for decoding to yield the final log-likelihood ratios (LLR) of the transmitted bits. Fig. 5-11 illustrates the resultant iterative receiver.



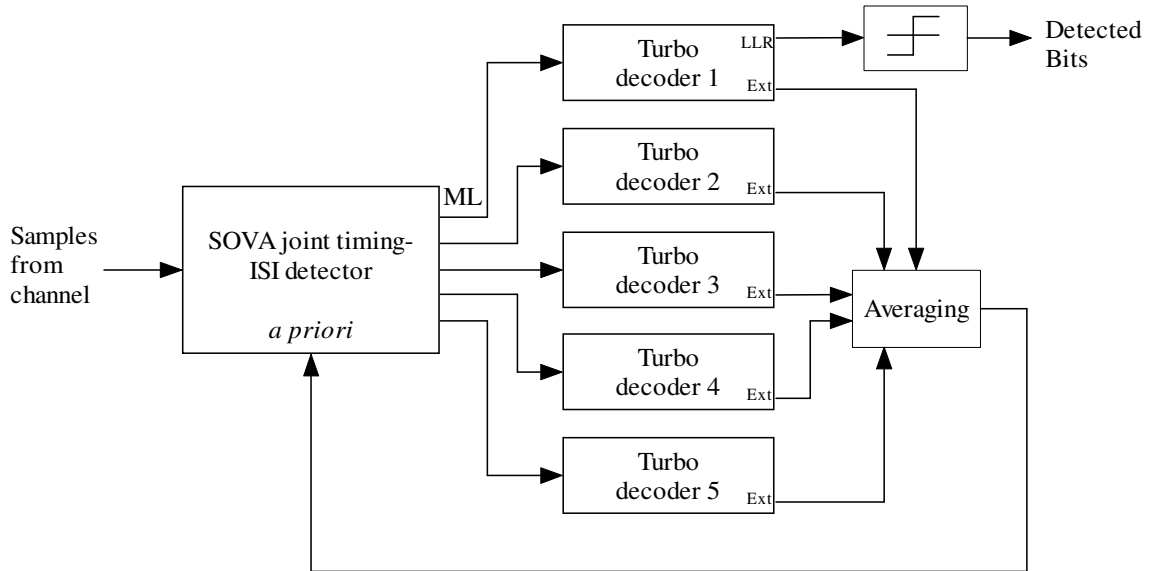


Fig. 5-11. SOVA based joint timing-ISI iterative receiver with 5 decoders.

Simulations show that the receiver of Fig. 5-11 is capable of correcting cycle-slips. Fig. 5-12 shows the tracking trajectories of the ML path for the same set of parameters as Fig. 5-8 except that now the SNR is 2 dB and a rate  $\frac{1}{2}$  Turbo code (encoded using parallel concatenated (37, 21) recursive systematic convolutional encoders [15]) is used. From Fig. 5-12 we can see that the receiver of Fig. 5-11 is indeed capable of correcting cycle-slips. By the fourth iteration, the cycle-slip has been completely corrected.

### 5.6 Delivering soft decisions before the end of the trellis

In the algorithm described in the previous section, the five (more could be chosen) paths that are sent into the five decoders are decided at the end of the trellis i.e. after all the samples have been collected. Therefore the delay will be undesirably large. It is actually possible to output the soft decisions earlier, before the end of the trellis. This, coupled with the fact that the decoding algorithm used in the Turbo decoders may be written in a forward-only fashion [16], could greatly reduce the overall delay of the receiver.

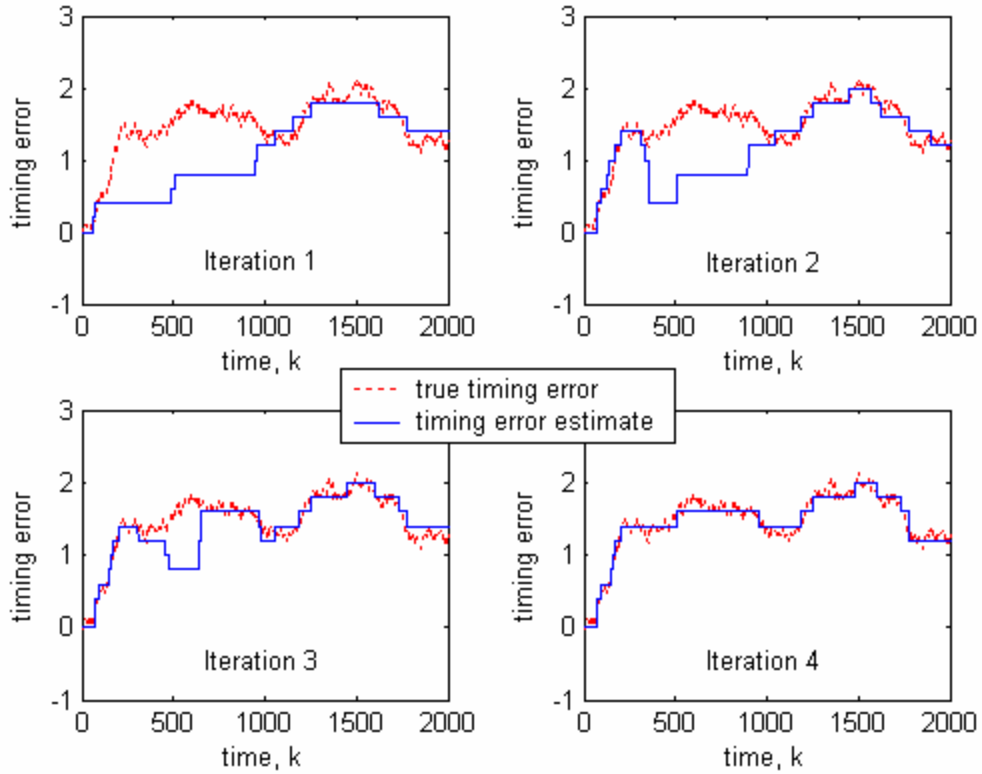


Fig. 5-12. Cycle-slip correction by the receiver of Fig. 5-11.

$$\sigma_w = 0.02T ; Q = 5 ; \delta = 0.005 ; \text{SNR} = 2 \text{ dB} ; \text{rate } \frac{1}{2} \text{ Turbo code.}$$

In order to output soft decisions early, actually all that is needed is to shift forward what was done previously at the end of the trellis. After a short delay (approximately 50 samples should be enough), the ML path can start sending soft decisions to its decoder. Similarly, those paths having timing offsets that are integer multiples of  $T$  from the ML path's timing offset can also start sending soft decisions to their respective decoders. For example, if five decoders are used, the five paths that will be sending decisions at the  $i^{\text{th}}$  ( $i$  is greater than the delay) sampling instant, will have the timing states of  $\varphi_i^{ML}$ ,  $\varphi_i^{ML} + T$ ,  $\varphi_i^{ML} - T$ ,  $\varphi_i^{ML} + 2T$ ,  $\varphi_i^{ML} - 2T$ , where  $\varphi_i^{ML}$  is the timing state of the ML path at the  $i^{\text{th}}$  sampling instant. It does not matter what the ISI states of each of the five paths are because the delay should ensure that all the paths with the same timing state at that instant have come from the same point in the past. Note that it will not be possible to always find all the other four paths with the timing states as

above. If a path with the required timing state cannot be found, the soft decision to be sent can simply be obtained from the ML path as a replacement.

If this is implemented, it will mean that the soft decisions sent to a decoder will actually come from various tracking paths. This is in contrast to the approach used in Section 5.5 where all soft decisions delivered to a particular decoder come from a single tracking path. Fig. 5-13 shows the manifestation of this – observe the existence of ‘spikes’ and ‘jumps’ in the tracking plots that are larger than the quantization setp-size of  $\frac{T}{5}$ . The receiver will still work despite this, although with degradation in performance, and we can see from Fig. 5-13 that after three iterations, the ‘spikes’ and ‘jumps’ have been greatly eliminated and the cycle-slip has also been corrected. Table 5-3 shows the number of errors obtained in 200000 data bits (200 blocks of 1000 bits).

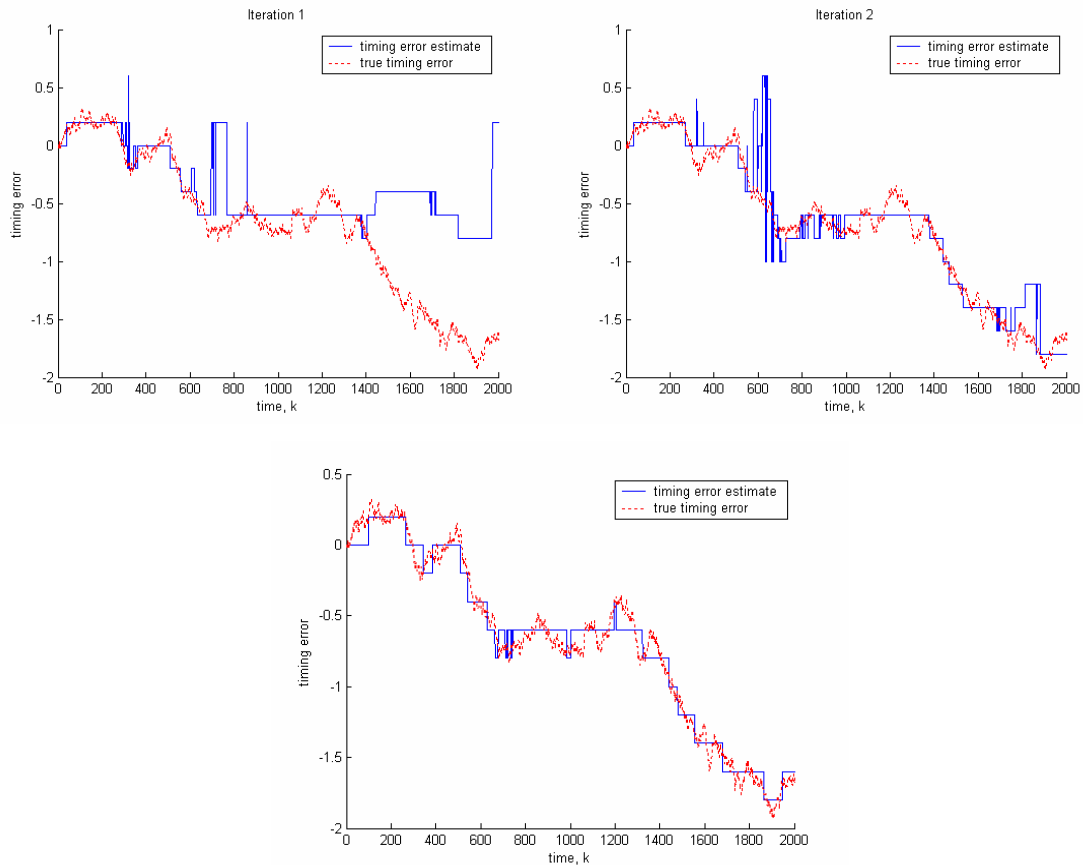


Fig. 5-13. Tracking plots of SOVA joint timing-ISI detector with early decisions.

$$\sigma_w = 0.02T ; Q = 5 ; \delta = 0.005 ; \text{SNR} = 2 \text{ dB} ; \text{rate } \frac{1}{2} \text{ Turbo code.}$$

SNR	Number of errors after 1 <sup>st</sup> iteration	Number of errors after 2 <sup>nd</sup> iteration	Number of errors after 3 <sup>rd</sup> iteration
2.00	27297	1181	0
2.75	14905	576	0
3.50	9355	320	0

Table 5-3. Number of errors obtained when implementing the SOVA joint timing-ISI detector for 200 blocks of 1000 data bits. ( $\sigma_w = 0.02T$  ;  $Q = 5$  ;  $\delta = 0.005$ ; rate 1/2 Turbo code).

## 5.7 Summary

In this chapter, we considered a recording channel with error-control coding. A novel joint timing-ISI trellis is proposed. This joint trellis is constructed by merging the ISI trellis of a simple channel model with a timing trellis derived from a simple timing error model. A Viterbi algorithm that uses the joint trellis is implemented and found through simulations to yield satisfactory detection and phase tracking performance when the SNR is high. The Viterbi algorithm is modified so that soft decisions may be delivered to iterative decoders. Cycle-slips, which occur frequently at low SNRs, may be corrected after several iterations between the SOVA detector and the iterative decoders.

## Chapter 6

### Joint Timing Recovery and Symbol Detection 2:

#### Joint timing-ISI Trellis 2

A second type of joint timing-ISI trellis [17] is described in this chapter. We show that the forward-backward BCJR algorithm [18] may be implemented on the trellis and soft decisions may be generated and delivered to soft-in soft-out (SISO) decoders. Simulations are used to show that cycle-slips may be corrected upon iterations with the SISO decoders. In order to remove the need for backward recursion, we also modify the forward-only MAP algorithm of [16] and apply it to the joint timing-ISI trellis.

#### 6.1 Introduction

The block schematic for the joint timing recovery and symbol detection scheme described in this chapter is the same as that depicted in Fig. 5-1b. Without loss of generality, we shall use the same channel pulse  $h(t)$  as Chapter 5 (see Fig. 5-3). Consequently, the ISI trellis will also be the same (see Fig. 5-4). The quantization of the timing error will be done identical to that described in Section 5-2. However, the timing trellis will be constructed in a different manner with a different set of timing state transition probabilities so that the timing trellis will not grow unbounded with time. (A trellis that will not grow unbounded is critical for the derivation of a MAP algorithm.) Thus the resultant joint timing-ISI trellis will be different from the joint timing-ISI trellis of Fig. 5-6 and Fig. 5-7.

#### 6.2 Timing trellis

In order to construct the timing trellis, the time axis shall be partitioned into non-overlapping semi-open intervals  $\left[ (k-1)T, kT \right]$ , where the length of each interval is the

symbol duration  $T$ . Each interval shall be further divided into  $Q$  sections. Fig. 6-1 shows an example of a partitioned time axis with  $Q = 5$ .

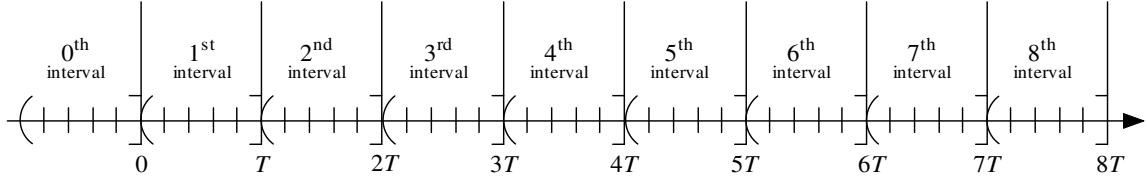


Fig. 6-1. Partitioning of time axis ( $Q = 5$ ).

Note that the different sections of each interval are separated by tick marks on the time axis. Since the timing error is quantized to multiples of  $\frac{T}{Q}$ , the sampling points will fall on the tick marks. We shall now define a new Markov timing state process and denote the state of the process by  $\zeta_k$ . This state is associated with the time interval  $((k-1)T, kT]$  and belongs to the set

$$\zeta_k \in \mathbb{T} = \{0, 1_1, 1_2, \dots, 1_Q, 2\}. \quad (6-1)$$

The notation is explained as follows:

- State  $\zeta_k = 0 \in \mathbb{T}$  denotes that the  $k^{\text{th}}$  symbol interval is not sampled at all.
- State  $\zeta_k = 1_q \in \mathbb{T}$ , where  $1 \leq q \leq Q$ , denotes that the  $k^{\text{th}}$  symbol interval is sampled once at the  $q^{\text{th}}$  tick.
- State  $\zeta_k = 2 \in \mathbb{T}$  denotes that the interval  $((k-1)T, kT]$  is sampled twice at the  $1^{\text{st}}$  and  $Q^{\text{th}}$  ticks.

Given the assumption of (5-5), which is reproduced here,

$$P(\tau_k | \tau_{k-1}) = \begin{cases} \delta & \text{if } \tau_k = \tau_{k-1} + \frac{T}{Q} \\ \delta & \text{if } \tau_k = \tau_{k-1} - \frac{T}{Q} \\ 1 - 2\delta & \text{if } \tau_k = \tau_{k-1} \\ 0 & \text{otherwise,} \end{cases} \quad (6-2)$$

a timing trellis may now be constructed. This trellis is shown in Fig. 6-2 for  $Q = 5$  and it captures all the possible valid timing state transitions.

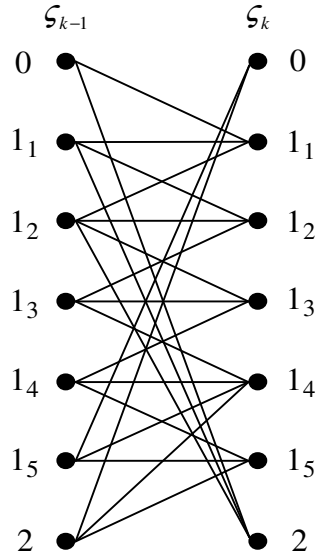


Fig. 6-2. One section of the timing trellis.

The corresponding timing state transition probabilities may be derived from (6-2). They are listed in Table 6-1.

$P(0 1_5) = \delta$	$P(0 2) = \delta$		
$P(1_1 0) = 1 - \delta$	$P(1_1 1_1) = 1 - 2\delta$	$P(1_1 1_2) = \delta(1 - \delta)$	
$P(1_2 1_1) = \frac{\delta}{1 - \delta}$	$P(1_2 1_2) = 1 - 2\delta$	$P(1_2 1_3) = \delta$	
$P(1_3 1_2) = \delta$	$P(1_3 1_3) = 1 - 2\delta$	$P(1_3 1_4) = \delta$	
$P(1_4 1_3) = \delta$	$P(1_4 1_4) = 1 - 2\delta$	$P(1_4 1_5) = \delta$	$P(1_4 2) = \delta$
$P(1_5 1_4) = \delta$	$P(1_5 1_5) = 1 - 2\delta$	$P(1_5 2) = 1 - 2\delta$	
$P(2 0) = \delta$	$P(2 1_1) = \frac{(1 - 2\delta)\delta}{1 - \delta}$	$P(2 1_2) = \delta^2$	

Table 6-1. Transition probabilities for the timing trellis of Fig. 6-2.

The derivations for some of the probabilities listed in Table 6-1 may not be obvious. For example, when computing the probability of transition from  $1_1 \rightarrow 1_1$ , we need to consider three intervals instead of two (see Fig. 6-3 where  $Q = 5$  is used).

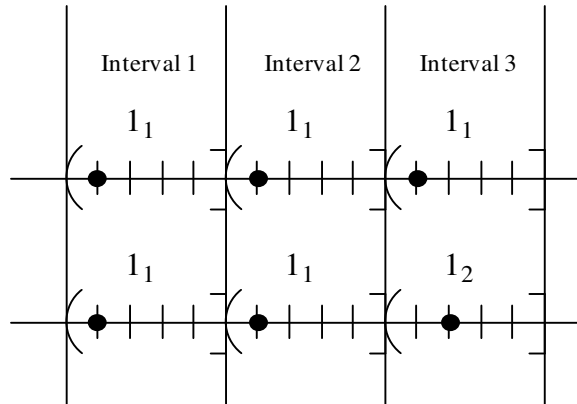


Fig. 6-3. Possible sampling sequences to be considered when computing  $P(1_1 | 1_1)$ .  $Q = 5$  is used and sampling points are marked by bullets.

This is to ensure that when we move from the sampling point in Interval 2 to the sampling point in Interval 3, the change in timing offset is not  $-\frac{T}{5}$ , otherwise the timing state in Interval 2 will be 2 instead of  $1_1$ . Tables 6-2 to 6-4 provide the derivations for the timing state transition probabilities that are less obvious.



$1_1 \rightarrow 1_1$		$1_1 \rightarrow 1_2$		$1_1 \rightarrow 2$	
Sampling sequence to be considered	Probability of sampling transitions	Sampling sequence to be considered	Probability of sampling transitions	Sampling sequence to be considered	Probability of sampling transitions
$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$(1-2\delta)(1-2\delta)$	$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$\delta$	$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$(1-2\delta)\delta$
$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$(1-2\delta)\delta$				
<b>Sum</b>	$(1-2\delta)(1-\delta)$	<b>Sum</b>	$\delta$	<b>Sum</b>	$(1-2\delta)\delta$
$P(1_1   1_1)$ $= \frac{(1-2\delta)(1-\delta)}{(1-2\delta)(1-\delta) + \delta + (1-2\delta)\delta}$ $= (1-2\delta)$		$P(1_2   1_1)$ $= \frac{\delta}{(1-2\delta)(1-\delta) + \delta + (1-2\delta)\delta}$ $= \frac{\delta}{1-\delta}$		$P(2   1_1)$ $= \frac{(1-2\delta)\delta}{(1-2\delta)(1-\delta) + \delta + (1-2\delta)\delta}$ $= \frac{(1-2\delta)\delta}{1-\delta}$	

Table 6-2. Derivations of  $P(1_1 | 1_1)$ ,  $P(1_2 | 1_1)$  and  $P(2 | 1_1)$ .

$1_2 \rightarrow 1_1$		$1_2 \rightarrow 1_2$		$1_2 \rightarrow 1_3$		$1_2 \rightarrow 2$	
Sampling sequence to be considered	Probability of sampling transitions	Sampling sequence to be considered	Probability of sampling transitions	Sampling sequence to be considered	Probability of sampling transitions	Sampling sequence to be considered	Probability of sampling transitions
$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$\delta(1-2\delta)$	$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$(1-2\delta)$	$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$\delta$	$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$\delta^2$
$-(\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow)$	$\delta^2$						
<b>Sum</b>	$\delta(1-\delta)$	<b>Sum</b>	$(1-2\delta)$	<b>Sum</b>	$\delta$	<b>Sum</b>	$\delta^2$
$P(1_1   1_2)$ $= \frac{\delta(1-\delta)}{\delta(1-\delta) + (1-2\delta) + \delta + \delta^2}$ $= \delta(1-\delta)$		$P(1_2   1_2)$ $= \frac{(1-2\delta)}{\delta(1-\delta) + (1-2\delta) + \delta + \delta^2}$ $= (1-2\delta)$		$P(1_3   1_2)$ $= \frac{\delta}{\delta(1-\delta) + (1-2\delta) + \delta + \delta^2}$ $= \delta$		$P(2   1_2)$ $= \frac{\delta^2}{\delta(1-\delta) + (1-2\delta) + \delta + \delta^2}$ $= \delta^2$	

Table 6-3. Derivations of  $P(1_1 | 1_2)$ ,  $P(1_2 | 1_2)$ ,  $P(1_3 | 1_2)$  and  $P(2 | 1_2)$ .

$0 \rightarrow 1_1$		$0 \rightarrow 2$	
Sampling sequence to be considered	Probability of sampling transitions	Sampling sequence to be considered	Probability of sampling transitions
$\leftarrow \text{++++} \text{ } \leftarrow \text{++++} \text{ } \leftarrow \text{++++} \text{ } \leftarrow$	$(1-2\delta)$	$\leftarrow \text{++++} \text{ } \leftarrow \text{++++} \text{ } \leftarrow$	$\delta$
$\leftarrow \text{++++} \text{ } \leftarrow \text{++++} \text{ } \leftarrow \text{++++} \text{ } \leftarrow$	$\delta$		
<b>Sum</b>	$(1-\delta)$	<b>Sum</b>	$\delta$
$P(1_1 0) = \frac{(1-\delta)}{(1-\delta)+\delta} = (1-\delta)$		$P(2 0) = \frac{\delta}{(1-\delta)+\delta} = \delta$	

Table 6-4. Derivations of  $P(1_1|0)$  and  $P(2|0)$ .

Fig. 6-4 shows an example of a path (thick line) through the timing trellis for a given sampling realization. The sample points are marked on the time axis by bullets.

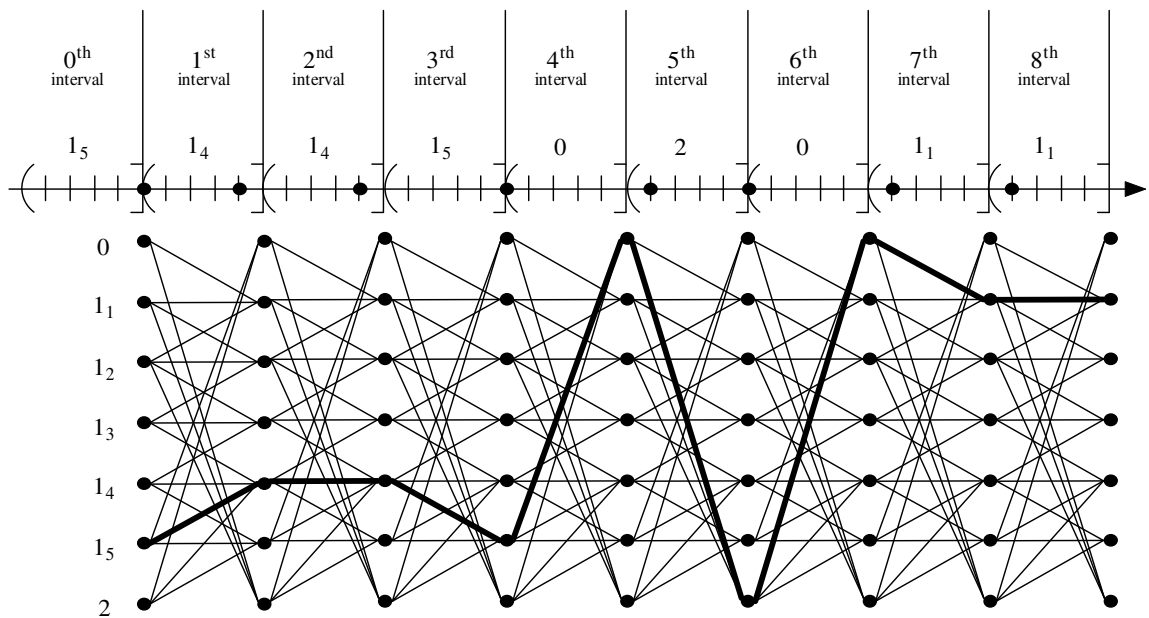


Fig. 6-4. An example of a path through the timing trellis for a given sampling realization.

### 6.3 Joint timing-ISI trellis

A joint timing-ISI trellis may be constructed by combining the timing trellis of Fig. 6-2 and the ISI trellis of Fig. 5-4. Denoting the set of ISI states by  $l$  so that  $l = \{(-1, -1), (-1, +1), (+1, -1), (+1, +1)\}$ , the joint timing-ISI trellis states will belong to

the set  $F = T \times I$ . The resultant trellis is shown in Fig. 6-5. Each state of the joint trellis is made up of an ISI state and a timing state.

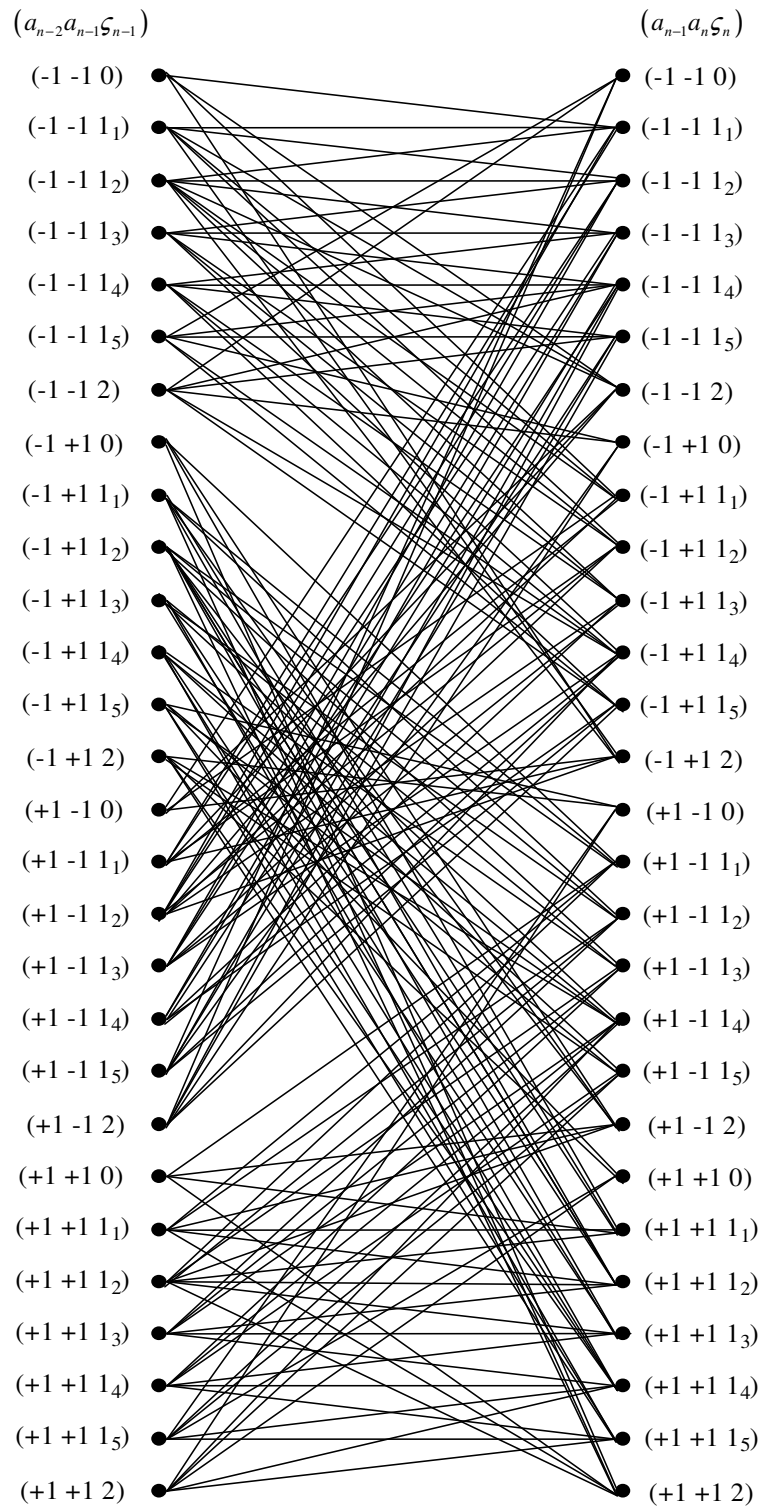


Fig. 6-5. One section of the joint timing-ISI trellis.

## 6.4 Forward-backward BCJR algorithm on the joint timing-ISI trellis

The BCJR algorithm [18] is a symbol-by-symbol maximum *a posteriori* (MAP) algorithm. It provides *a posteriori* probabilities of symbols by using systematic forward and backward recursions and is able to minimize the symbol error probability in either decoding or symbol detection. We shall now describe how to modify the forward-backward BCJR algorithm so that it may be implemented on the joint timing-ISI trellis.

Denoting the state of the joint timing-ISI trellis at time  $k$  by  $S_k \in \mathbb{F}$  and using the Markov property of the timing trellis, the conditional probability of reaching state  $S_k$  from state  $S_{k-1}$  is given by

$$P(S_k | S_{k-1}, S_{k-2}, \dots) = P(S_k | S_{k-1}) = P(a_k)P(\zeta_k | \zeta_{k-1}). \quad (6-3)$$

Let  $Z_k$  represent the vector of output samples corresponding to input bit  $a_k$ . Thus  $Z_k$  may include one sample, two samples or no samples, depending on how many samples fall in the interval  $((k-1)T, kT]$ . We may now proceed to define the following variables that are needed for executing the forward-backward BCJR algorithm on the joint timing-ISI trellis.

- The  $\alpha$ -coefficient  $\alpha(k, m, i)$  is defined as the probability that the  $k^{\text{th}}$  state  $S_k$  equals  $m \in \mathbb{F}$  and that there are  $i$  samples during the first  $k$  bit intervals:

$$\alpha(k, m, i) = P(S_k = m, Z_1^k = r_1^i). \quad (6-4)$$

- Let  $L$  be the total number of samples and  $N$  be the total number of transmitted bits. The  $\beta$ -coefficient  $\beta(k, m, i)$  is defined as the conditional probability that there are  $L-i$  samples in the last  $N-k$  bit intervals, given that the  $k^{\text{th}}$  timing-ISI state  $S_k$  equals  $m \in \mathbb{F}$ :

$$\beta(k, m, i) = P(Z_{k+1}^N = r_{i+1}^L | S_k = m). \quad (6-5)$$

- Let  $S_{k-1} = m' \in \mathbb{F}$  be the state at time  $k-1$  and let  $S_k = m \in \mathbb{F}$  be the state at time  $k$ . Then the  $\gamma$ -coefficient for the branch  $(m', m)$  is defined as the joint

probability of the state at time  $k$  and the sample vector  $Z_k$  at time  $k$ , given the knowledge of the state at time  $k-1$ :

$$\gamma(k, m', m, i) = \begin{cases} P(S_k = m, Z_k = r_i | S_{k-1} = m') & \text{if } m \in \mathbb{I} \times \{1, \dots, 1_Q\} \\ P(S_k = m, Z_k = \emptyset | S_{k-1} = m') & \text{if } m \in \mathbb{I} \times \{0\} \\ P(S_k = m, Z_k = r_{i-1}^i | S_{k-1} = m') & \text{if } m \in \mathbb{I} \times \{2\}. \end{cases} \quad (6-6)$$

With this notation, the forward-backward recursions are done as follows:

$$\alpha(k, m, i) = \begin{cases} \sum_{m' \in \mathbb{F}} \alpha(k-1, m', i-1) \gamma(k, m', m, i) & \text{if } m \in \mathbb{I} \times \{1, \dots, 1_Q\} \\ \sum_{m' \in \mathbb{F}} \alpha(k-1, m', i) \gamma(k, m', m, i) & \text{if } m \in \mathbb{I} \times \{0\} \\ \sum_{m' \in \mathbb{F}} \alpha(k-1, m', i-2) \gamma(k, m', m, i) & \text{if } m \in \mathbb{I} \times \{2\}. \end{cases} \quad (6-7)$$

$$\begin{aligned} \beta(k, m, i) = & \sum_{m' \in \mathbb{I} \times \{1, \dots, 1_Q\}} \beta(k+1, m', i+1) \gamma(k+1, m, m', i+1) \\ & + \sum_{m' \in \mathbb{I} \times \{0\}} \beta(k+1, m', i) \gamma(k+1, m, m', i) \\ & + \sum_{m' \in \mathbb{I} \times \{2\}} \beta(k+1, m', i+2) \gamma(k+1, m, m', i+2). \end{aligned} \quad (6-8)$$

The initial coefficients are chosen to reflect the prior knowledge that we have of the timing error and transmitted symbols at the beginning and the end of the transmitted block. If we assume that a prior timing acquisition stage will give us an initial timing error of zero and that we start from the  $(-1, -1)$  ISI state, the initial coefficients may be set as

$$\alpha(0, m, i) = \begin{cases} 1 & \text{if } i = 0 \text{ and } m = (-1, -1, 1_Q) \\ 0 & \text{otherwise,} \end{cases} \quad (6-9)$$

$$\beta(N, m, i) = \begin{cases} 1 & \text{if } i = L \text{ and all } m \in \mathbb{F} \\ 0 & \text{otherwise,} \end{cases} \quad (6-10)$$

The *a posteriori* probabilities of the joint timing-ISI states  $S_k = m \in \mathbb{F}$  are given by

$$\lambda_k(m) = P(S_k = m, Z_1^N = r_1^L) = \sum_{i=1}^L \alpha(k, m, i) \beta(k, m, i), \quad (6-11)$$

and the soft output is given by

$$\log \left( \frac{P(a_k = +1 | r_1^L)}{P(a_k = -1 | r_1^L)} \right) = \log \left( \frac{\sum_{m:a_k=+1} \lambda_k(m)}{\sum_{m:a_k=-1} \lambda_k(m)} \right), \quad (6-12)$$

from which the extrinsic information to be delivered to decoders may be obtained as

$$L_{ext,k} = \log \frac{P(a_k = +1 | r_1^L)}{P(a_k = -1 | r_1^L)} - \log \left( \frac{P(a_k = +1)}{P(a_k = -1)} \right). \quad (6-13)$$

Finally, the *a posteriori* probabilities of the timing error may also be computed using the  $\alpha$  - and  $\beta$  -coefficients:

$$\begin{aligned} P(\varepsilon_i = \varphi | r_1^L) = & \sum_{k,j:(n-i)T + \frac{jT}{Q} = \varphi} \left[ \sum_{m:m \in 1 \times \{1_j\}} \alpha(k,m,i) \beta(k,m,i) \right] \\ & + \sum_{k:(k-i)T + T = \varphi} \left[ \sum_{m:m \in 1 \times \{2\}} \alpha(k,m,i) \beta(k,m,i) \right] \\ & + \sum_{k:(k-i)T + \frac{T}{Q} = \varphi} \left[ \sum_{m:m \in 1 \times \{2\}} \alpha(k,m,i+1) \beta(k,m,i+1) \right], \end{aligned} \quad (6-14)$$

for all  $1 \leq i \leq L$ .

## 6.5 Simulation results

The forward-backward joint timing-ISI detector is capable of correcting cycle-slips in tracking when used with iterative decoders. The block schematic of the BCJR based joint timing-ISI iterative receiver is shown in Fig. 6-6.

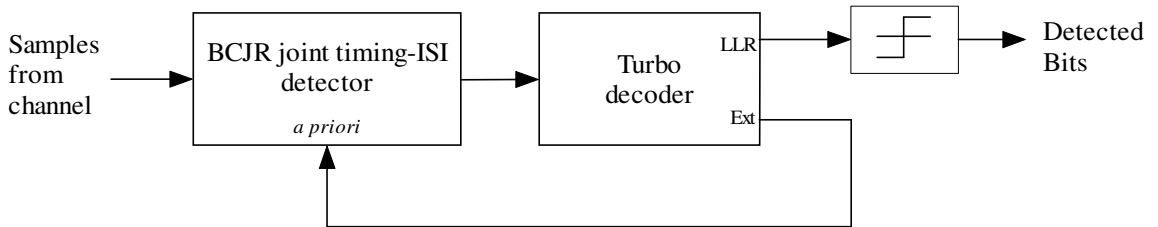


Fig. 6-6. BCJR joint timing-ISI iterative receiver.

Fig. 6-7 shows a typical cycle-slip correction when a rate  $\frac{1}{2}$  Turbo code (encoded using parallel concatenated (37, 21) recursive systematic convolutional encoders [15]) is used. The timing error estimates plotted have maximum *a posteriori* probabilities, obtained through (6-14).

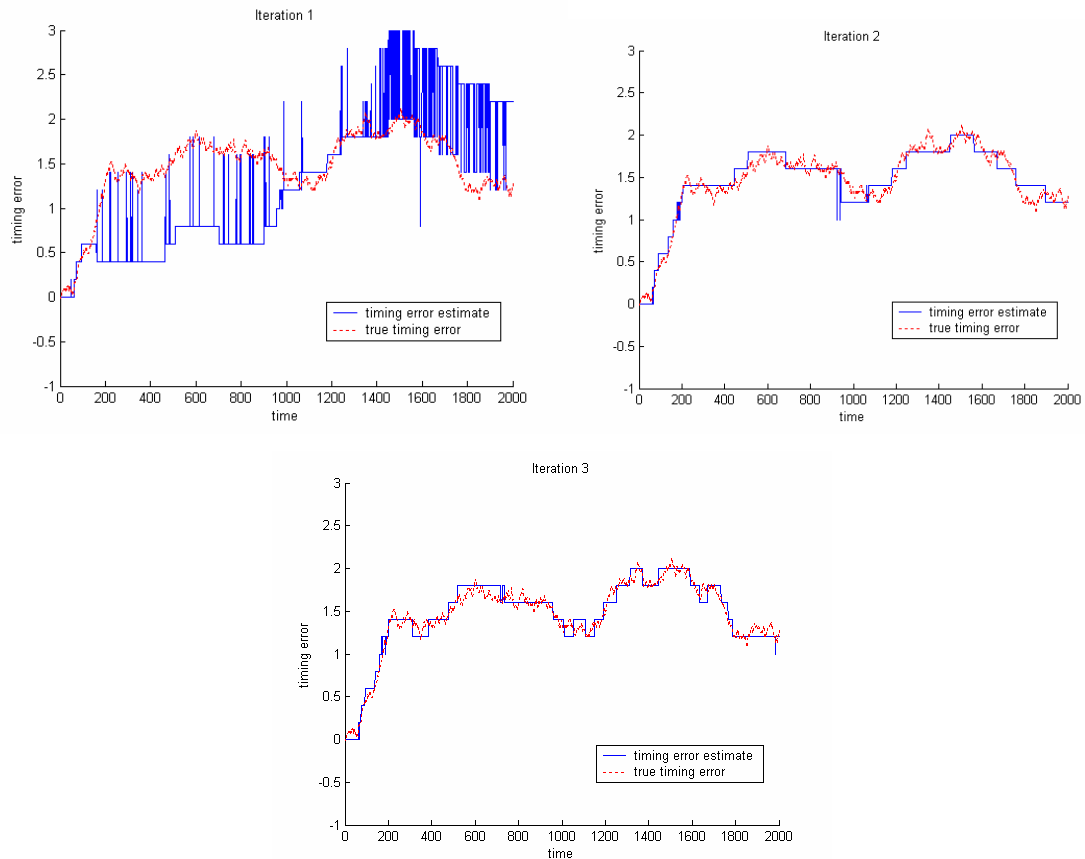


Fig. 6-7. Tracking plots of BCJR joint timing-ISI detector.

$$\sigma_w = 0.02T ; Q = 5 ; \delta = 0.005 ; \text{SNR} = 2 \text{ dB} ; \text{rate } \frac{1}{2} \text{ Turbo code.}$$

## 6.6 Forward-only MAP algorithm on the joint timing-ISI trellis

In this section, we present a forward-only MAP algorithm on the joint timing-ISI trellis that will eliminate the need for a backward recursion. This will reduce the delay caused by the detector because computations can start after a small number of samples have been collected. Such an algorithm will be useful in real-time systems.

The forward-only MAP algorithm has four steps: initialize, extend, update and collect. We will be using the  $\alpha$ -coefficient of (6-4) and the  $\gamma$ -coefficient of (6-6). In addition, we define two additional variables as follows.

- The  $\chi^{(+)}$ -coefficient is defined as the probability that the  $k^{th}$  state  $S_k$  equals  $m \in \mathbb{F}$ , the  $j^{th}$  input bit is a +1 and that there are  $i$  samples during the first  $k$  bit intervals:

$$\chi^{(+)}(k, m, j, i) = P(S_k = m, a_j = +1, Z_1^k = r_1^i). \quad (6-15)$$

- Similarly, the  $\chi^{(-)}$ -coefficient is defined as

$$\chi^{(-)}(k, m, j, i) = P(S_k = m, a_j = -1, Z_1^k = r_1^i). \quad (6-16)$$

The forward-only MAP algorithm may then be formulated as follows.

### **Initialize**

Define  $m_0 = (-1, -1, 1_Q)$ . Again we have assumed that a prior timing acquisition stage is able to deliver an initial timing error of zero and that we end with at least two  $-1$  bits at the end of the sync sequence before the start of the message sequence. The  $\chi^{(+)}$ -coefficient and  $\chi^{(-)}$ -coefficient are initialized as follows:

$$\chi^{(+)}(1, m, 1, 1) = \gamma(1, m_0, m, 1) \quad \text{if } m \in \{(-1, +1)\} \times \{1_{Q-1}, 1_Q\}. \quad (6-17a)$$

$$\chi^{(-)}(1, m, 1, 1) = \gamma(1, m_0, m, 1) \quad \text{if } m \in \{(-1, -1)\} \times \{1_{Q-1}, 1_Q\}. \quad (6-17b)$$

$$\chi^{(+)}(1, m, 1, 0) = \gamma(1, m_0, m, 1) \quad \text{if } m = (-1, +1, 0). \quad (6-17c)$$

$$\chi^{(-)}(1, m, 1, 0) = \gamma(1, m_0, m, 1) \quad \text{if } m = (-1, -1, 0). \quad (6-17d)$$

$$\chi^{(\pm)}(1, m, 1, i) = 0 \quad \text{for all other } i \text{ and } m. \quad (6-17e)$$

### **Extend**

Upon receiving a sample, the trellis will be extended according to the following equations:



$$\begin{aligned}
& \alpha(k-1, m', i) \\
&= P(S_{k-1} = m', Z_1^{k-1} = r_1^i) \\
&= P(S_{k-1} = m', a_{k-1} = +1, Z_1^{k-1} = r_1^i) + P(S_{k-1} = m', a_{k-1} = -1, Z_1^{k-1} = r_1^i) \\
&= \chi^{(+)}(k-1, m', k-1, i) + \chi^{(-)}(k-1, m', k-1, i), \tag{6-18a}
\end{aligned}$$

$$\begin{aligned}
& \chi^{(+)}(k, m, k, i) \\
&= P(S_k = m, a_k = +1, Z_1^k = r_1^i) \\
&= \begin{cases} \sum_{m'} P(S_{k-1} = m', Z_1^{k-1} = r_1^{i-1}, Z_k = r_i, S_k = m) & \text{if } m \in \{(\pm 1, +1)\} \times \{1_q\} \\ \sum_{m'} P(S_{k-1} = m', Z_1^{k-1} = r_1^i, Z_k = \emptyset, S_k = m) & \text{if } m \in \{(\pm 1, +1)\} \times \{0\} \\ \sum_{m'} P(S_{k-1} = m', Z_1^{k-1} = r_1^{i-2}, Z_k = r_{i-1}^i, S_k = m) & \text{if } m \in \{(\pm 1, +1)\} \times \{1_\emptyset\} \end{cases} \\
&= \begin{cases} \sum_{m'} \alpha(k-1, m', i-1) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, +1)\} \times \{1_q\} \\ \sum_{m'} \alpha(k-1, m', i) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, +1)\} \times \{0\} \\ \sum_{m'} \alpha(k-1, m', i-2) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, +1)\} \times \{1_\emptyset\}, \end{cases} \tag{6-18b}
\end{aligned}$$

$$\begin{aligned}
& \chi^{(-)}(k, m, k, i) \\
&= P(S_k = m, a_k = -1, Z_1^k = r_1^i) \\
&= \begin{cases} \sum_{m'} P(S_{k-1} = m', Z_1^{k-1} = r_1^{i-1}, Z_k = r_i, S_k = m) & \text{if } m \in \{(\pm 1, -1)\} \times \{1_q\} \\ \sum_{m'} P(S_{k-1} = m', Z_1^{k-1} = r_1^i, Z_k = \emptyset, S_k = m) & \text{if } m \in \{(\pm 1, -1)\} \times \{0\} \\ \sum_{m'} P(S_{k-1} = m', Z_1^{k-1} = r_1^{i-2}, Z_k = r_{i-1}^i, S_k = m) & \text{if } m \in \{(\pm 1, -1)\} \times \{1_\emptyset\} \end{cases} \\
&= \begin{cases} \sum_{m'} \alpha(k-1, m', i-1) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, -1)\} \times \{1_q\} \\ \sum_{m'} \alpha(k-1, m', i) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, -1)\} \times \{0\} \\ \sum_{m'} \alpha(k-1, m', i-2) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, -1)\} \times \{1_\emptyset\}. \end{cases} \tag{6-18c}
\end{aligned}$$

## Update

After extension of the trellis, we update the past probabilities for all  $1 \leq j \leq k-1$  according to the equations,

$$\begin{aligned}
& \chi^{(+)}(k, m, j, i) \\
&= P(S_k = m, a_j = +1, Z_1^k = r_1^i) \\
&= \begin{cases} \sum_{m'} P(S_{k-1} = m', a_j = +1, Z_1^{k-1} = r_1^{i-1}) \cdot P(S_k = m, Z_k = r_i | S_{k-1} = m') & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{1_q\} \\ \sum_{m'} P(S_{k-1} = m', a_j = +1, Z_1^{k-1} = r_1^i) \cdot P(S_k = m, Z_k = \emptyset | S_{k-1} = m') & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{0\} \\ \sum_{m'} P(S_{k-1} = m', a_j = +1, Z_1^{k-1} = r_1^{i-2}) \cdot P(S_k = m, Z_k = r_{i-1}^i | S_{k-1} = m') & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{1_\emptyset\} \end{cases} \\
&= \begin{cases} \sum_{m'} \chi^{(+)}(k-1, m', j, i-1) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{1_q\} \\ \sum_{m'} \chi^{(+)}(k-1, m', j, i) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{0\} \\ \sum_{m'} \chi^{(+)}(k-1, m', j, i-2) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{1_\emptyset\}. \end{cases} \quad (6-19a)
\end{aligned}$$

$$\begin{aligned}
& \chi^{(-)}(k, m, j, i) \\
&= P(S_k = m, a_j = -1, Z_1^k = r_1^i) \\
&= \begin{cases} \sum_{m'} P(S_{k-1} = m', a_j = -1, Z_1^{k-1} = r_1^{i-1}) \cdot P(S_k = m, Z_k = r_i | S_{k-1} = m') & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{1_q\} \\ \sum_{m'} P(S_{k-1} = m', a_j = -1, Z_1^{k-1} = r_1^i) \cdot P(S_k = m, Z_k = \emptyset | S_{k-1} = m') & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{0\} \\ \sum_{m'} P(S_{k-1} = m', a_j = -1, Z_1^{k-1} = r_1^{i-2}) \cdot P(S_k = m, Z_k = r_{i-1}^i | S_{k-1} = m') & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{1_\emptyset\} \end{cases} \\
&= \begin{cases} \sum_{m'} \chi^{(-)}(k-1, m', j, i-1) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{1_q\} \\ \sum_{m'} \chi^{(-)}(k-1, m', j, i) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{0\} \\ \sum_{m'} \chi^{(-)}(k-1, m', j, i-2) \gamma(k, m', m, i) & \text{if } m \in \{(\pm 1, \pm 1)\} \times \{1_\emptyset\}. \end{cases} \quad (6-19b)
\end{aligned}$$

## Collect

Finally the *a posteriori* probabilities of the input bits may be obtained by collecting the

$\chi^{(+)}$  - and  $\chi^{(-)}$  -coefficients:

$$P(a_k = +1, Z_1^N = r_1^i) = \sum_m \chi^{(+)}(N, m, k, i), \quad (6-20a)$$

$$P(a_k = -1, Z_1^N = r_1^i) = \sum_m \chi^{(-1)}(N, m, k, i). \quad (6-20b)$$

## 6.7 Summary

In this chapter, we presented a different timing trellis from that of Chapter 5. This trellis will not grow unbounded with time and consequently, the joint timing-ISI trellis will also not grow unbounded with time. The forward-backward BCJR algorithm is adapted to make use of the joint timing-ISI trellis in order to achieve MAP detection of symbols and timing phases. Simulations show that cycle-slips may be corrected after several iterations between the BCJR detector and iterative decoders. We also show that the BCJR algorithm may be written in a forward-only manner so that there will no longer be a need for a backward recursion. This will result in a shorter detector delay, which is important for real-time systems. Due to resource constraints, computer simulations are not done for the forward-only BCJR algorithm.

## Chapter 7

### Conclusion

In this thesis, we investigated timing acquisition for the PR equalized perpendicular magnetic recording channel. The problem of hang-up is identified when the channel equalizer is placed within the timing recovery loop. This is because the delay caused by the equalizer causes the phase updates to lag the actual timing variations significantly. Variable threshold detection (VTD) is inadequate in solving the hang-up problem. Therefore two novel acquisition techniques were developed to tackle the problem. We also investigated low SNR timing tracking using a simple channel model and a random walk phase drift model. A novel joint timing-ISI trellis was constructed and SOVA was implemented on the trellis. A forward-only MAP algorithm was also developed for a second joint timing-ISI trellis.

The contributions of the thesis may be divided into two parts. Part 1, consisting of Chapters 2, 3 and 4, is about timing acquisition for the PR equalized perpendicular magnetic recording channel. The channel is equalized to the [1 2 2 1] PR target using MMSE equalization and the MM TED is used to generate the timing error estimates. Timing acquisition is done with a second order phase locked loop and preamble sequences. In the first fast acquisition technique, we proposed to collect a number of samples equal to the period of the preamble and use them to identify an ideal sequence to be delivered to the TED. This is termed direct matching. The concept of anchor points was introduced to explain why this scheme works. Subsequently, the matching scheme was extended to cover preambles of longer periods and the indirect matching scheme was proposed as a means of reducing the mean data collection time needed before starting phase updates. This is based on the identification of sequences that may operate as ‘seeds’ to create ideal sequences for sending to the TED. Simulations showed that hang-ups during acquisition may be completely eliminated even when the SNR is very low.

A second fast acquisition technique was also proposed and this is based on flipping undesirable TED outputs in order to eliminate reversals in the direction of phase

convergence during acquisition. Through theory and simulations, it was found that hang-up region indicators may be created that could be used to trigger such a correction mechanism. We provided theoretical derivations for optimum thresholds that could be used for detecting these indicators with low probabilities of false alarm and false trigger. The acquisition technique was again found through simulations to be effective in completely eliminating hang-ups during timing acquisition even for very low SNRs.

Part 2 of the thesis, consisting of Chapters 5 and 6, is concerned with timing recovery at low SNRs, which is a crucial research topic because powerful capacity approaching codes like Turbo and LDPC codes operate at very low SNRs. Conventional decision-directed timing recovery schemes fail completely at such low SNRs because they suffer from cycle slips. Consequently, the large coding gains offered by the codes are severely eroded. We designed a novel joint timing-ISI trellis for doing joint timing recovery and symbol detection and implemented SOVA on the trellis to generate soft decisions that may be delivered to SISO decoders. This joint trellis is obtained by merging the standard ISI trellis for the given channel response and a timing trellis that is derived from a quantized Markov timing error model. Soft decisions may be output early before the end of the data block is reached and simulations showed that upon a few iterations where extrinsic information is exchanged between the SOVA detector and the SISO decoder, cycle slips during tracking may be corrected.

We also investigated a second type of joint timing-ISI trellis and implemented the BCJR algorithm on the trellis. Cycle slip corrections are again possible when extrinsic information is exchanged between the detector and the decoder. In order to eliminate the need for backward recursion, we have also formulated a forward-only BCJR algorithm on the trellis.

Lastly, there are several possible areas of interest for further work and they are listed as follows.

- Design of reduced complexity SOVA based joint timing-ISI detector. Specifically, equation (5-14) can be simplified by using suitable approximations.
- Design of a single LDPC decoder that is able to make use of soft decisions from multiple tracking paths that are provided by the SOVA joint timing-ISI detector. This will eliminate the need for multiple decoders as illustrated in Fig. 5-11.

- Design of reduced complexity BCJR based detector for the joint timing-ISI trellis of Chapter 6. Some  $\alpha$ - and  $\beta$ -coefficients are negligibly small and thus should be identified and eliminated from the forward-backward recursions to save on computational costs.
- Implementation of the forward-only MAP algorithm described in Section 6-6.
- Design and simulations of joint timing-ISI detectors for more realistic channel models (e.g. PR4) as well as evaluation of performance when higher rate codes are used.

## Bibliography

- [1] Y. Okamoto, H. Sumiyoshi, T. Kishigami, M. Akamatsu, H. Osawa, H. Saito, H. Muraoka, and Y. Nakamura, "A study of PRML systems for perpendicular recording using double layered medium," *IEEE Trans. Magn.*, vol. 36, pp. 2164-2166, Sept. 2000.
- [2] P. H. Siegel and J. K. Wolf, "Modulation and coding for information storage," *IEEE Commun. Magn.*, vol. 29, pp. 68-86, Dec. 1991.
- [3] J. J. Moon and L. R. Carley, "Performance comparison of detection methods in magnetic recording," *IEEE Trans. Magn.*, vol. 26, pp. 3155-3172, Nov. 1990.
- [4] J. G. Proakis, "Digital Communications," *McGraw-Hill International Edition*, 4<sup>th</sup> edition, 2001, chap. 11.
- [5] J. Moon and W. Zeng, "Equalization for maximum-likelihood detectors," *IEEE Trans. on Magnetics*, vol. 31, no. 2, pp. 1083-1088, March 1995.
- [6] P. Kovintavewat; I. Ozgunes; E. Kurtas, J. R. Barry and S. W. McLaughlin, "Generalized partial-response targets for perpendicular recording with jitter noise", *IEEE Trans. on Magnetics*, vol. 38, pp. 2340-2342, Sept. 2002.
- [7] H. Kobayashi, "Simultaneous adaptive estimation and decision algorithm for carrier modulated data transmission systems," *IEEE Trans. Commun. Technol.*, vol. 19, pp. 268-286, June 1971.
- [8] S. Qureshi, "Timing recovery for equalized partial-response systems," *IEEE Trans. Commun.*, vol. 24, pp. 1326-1331, Dec. 1976.
- [9] D. L. Lyon, "Timing recovery in synchronous equalized data communication," *IEEE Trans. Commun.*, vol. 23, pp. 269-274, Feb. 1975.
- [10] D. N. Godard, "Passband timing recovery in an all-digital modem receiver," *IEEE Trans. Commun.*, vol. 26, pp. 517-523, May 1978.
- [11] B. Farhang-Boroujeny, "Near-optimum timing recovery for digitally implemented data receivers," *IEEE Trans. Commun.*, vol. 38, pp. 1333-1336, Sept. 1990.
- [12] K. H. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. 24, pp. 516-530, May 1976.
- [13] F. Dolivo, W. Schott, and G. Ungerböck, "Fast timing recovery for partial-response signaling systems", in *Proc. IEEE Intl. Conf. Commun.*, 1989, vol. 1, pp. 573-577.

- [14] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft decision outputs and its applications," in *Proc. IEEE Intl. Conf. Global Telecommun. (GLOBECOM)*, Dallas, Texas, Nov. 1989, pp. 1680-1688.
- [15] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Geneva, Switzerland, May 1993, pp. 1064-1070.
- [16] X. Ma and A. Kavčić, "Path partitions and forward-only trellis algorithms", *IEEE Trans. Inform. Theory*, vol. 49, pp. 38-52, Jan. 2003.
- [17] W. Zeng and A. Kavčić, "Optimal soft-output detector for channels with intersymbol interference and timing errors," *IEEE Trans. Magn.*, vol. 39, pp. 2555-2557, Sept. 2003.
- [18] L. R. Bahl, J. C. Cocke, F. Jelinek, and J. Raviv, "Optimum decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287, Mar. 1974.
- [19] S. M. Kay, "Fundamentals of statistical signal processing," in *Volume I: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [20] B. Buch, "Method for estimating sampling phase from synchronously demodulated samples of sinusoidal waveforms," U.S. Patent 6 191 906, Feb. 2001.
- [21] J. Liu, H. Song, and B. V. K. Vijaya Kumar, "Timing acquisition for low-SNR data storage channels," *IEEE Trans. Magn.*, vol. 39, pp. 2558-2560, Sept. 2003.
- [22] J. Liu, H. Song and B. V. K. Vijaya Kumar, "Symbol timing recovery for low-SNR partial response channels," in *Proc. IEEE Intl. Conf. Global Telecommun. (GLOBECOM) 2002*, available at <http://www.ece.cmu.edu/~jingfeng>.
- [23] J. Liu, H. Song and B. V. K. Vijaya Kumar, "Dual segmented Kalman filters based symbol timing recovery for low-SNR partial response data storage channels," in *Proc. IEEE Intl. Conf. Global Telecommun. (GLOBECOM) 2003*, available at <http://www.ece.cmu.edu/~jingfeng>.
- [24] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Joint timing recovery and turbo equalization for coded partial response channels," *IEEE Trans. Magn.*, vol. 48, pp. 2295-2297, Sept. 2002.
- [25] X. Jin and A. Kavčić, "Cycle-slip detection using soft-output information," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, June 2001, pp. 2706-2710.
- [26] X. Jin and A. Kavčić, "Cycle-slip-detector-aided timing recovery," *IEEE Trans. Magn.*, vol. 38, pp. 2292-2294, Sept 2002.



- [27] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21-28, Jan. 1962.
- [28] D. J. C. MacKay and R. M. Neal, "Near Shannon-limit performance of low-density parity-check codes," *Electronic Letters*, vol. 32, pp. 1645-1646, Aug. 1996.
- [29] K. H. Kwek and George Mathew, "A novel timing acquisition technique for perpendicular magnetic recording channels", in *Proc. IEEE Intl. Conf. Global Telecommun. (GLOBECOM)*, San Francisco, California, Dec. 2003.
- [30] J. W. M. Bergmans, "Digital baseband transmission and recording," *Boston: Kluwer Academic Publishers*, 1996, chaps. 9-11.