A NOVEL INFINITE-IMPULSE RESPONSE EQUALIZER DESIGN AND ITERATIVE TIMING RECOVERY FOR MAGNETIC RECORDING CHANNELS

CHANON WARISARN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2011

KMITL-2011-EN-D-018-033

Thesis Certification

Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

Thesis Title	A Novel Infinite-Impulse Response Equalizer Design and Iterative Timing Recovery for Magnetic Recording Channels
Student	Mr. Chanon Warisarn
Student Id.	49060053
Degree	Doctor of Engineering
Program	Electrical Engineering
Thesis Advisor	Assoc. Prof. Dr. Pornchai Supnithi
Thesis Co-Advisor	Assoc. Prof. Dr. Piya Kovintavewat
Thesis Reference Number	KMITL-2011-EN-D-018-033

EXAMINERS		SIGNATURES
Assoc. Prof. Dr. Kraisin	Songwatana	
Asst. Prof. Dr. Pisit	Boonsrimuang	
Asst. Prof. Dr. Phaophak	Sirisuk	
Asst. Prof. Dr. Sorawat	Chivaprecha	
Assoc. Prof. Dr. Pornchai	Supnithi	

Date 7th April 2011 **Time** 03.00-05.00 pm.

Place Building A , 5th Floors Conference room no.4

(Assoc. Prof. Dr. Suchatchavee Suwansawas)

Dean, Faculty Of Engineeing

7th April 2011

หัวข้อวิทยานิพนธ์	การออกแบบอีควอไลเซอร์ผลตอบสนองอิมพัลส์ไม่จำกัดด้วยวิธีใหม่ และระบบกู้สัญญาณทางเวลาด้วยการวนซ้ำสำหรับช่องสัญญาณการบันทึกเชิงแม่เหล็ก
นักศึกษา	นายชานนท์ วริสาร
รหัสประจำตัว	49060053
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2554
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ. ดร. พรชัย ทรัพย์นิธิ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	รศ. ดร. ปิยะ โควินท์ทวีวัฒน์

บทคัดย่อ

ในวิทยานิพนธ์เล่มนี้ ได้นำเสนอวิธีการออกแบบอีควอไลเซอร์แบบผลตอบสนองอิมพัลส์ไม่จำกัด สำหรับช่องสัญญาณการบันทึกเชิงแม่เหล็กแนวตั้ง บนพื้นฐานของวิธีการหาค่าผิดพลาดกำลังสองเฉลี่ยน้อยที่สุด และได้อธิบายการได้มาซึ่งค่าสัมประสิทธิ์ของอีควอไลเซอร์แบบผลตอบสนองอิมพัลส์ไม่จำกัดไว้อย่างชัดเจน จากนั้นได้ทำการเปรียบเทียบประสิทธิภาพของอีควอไลเซอร์ ในแบบที่ได้นำเสนอเปรียบเทียบกับอีควอไลเซอร์แบบผลตอบสนองอิมพัลส์จำกัด ทั้งในระบบที่มีและไม่มีสัญญาณรบกวนจิตเตอร์ในสื่อบันทึก (media jitter noise) ซึ่งผลที่ได้จากการจำลองระบบได้แสดงให้เห็นว่า วิธีที่ได้นำเสนอมีประสิทธิภาพที่ดีกว่าอีควอไลเซอร์แบบผลตอบสนองอิมพัลส์จำกัด เมื่อทำการจำลองระบบที่ระดับสัญญาณรบกวนจิตเตอร์ในสื่อบันทึกปานกลางถึงระดับมาก และอีควอไลเซอร์ที่ใช้มีจำนวนแท็ปน้อย ซึ่งพบว่าเมื่อเปรียบเทียบสมรรถนะในรูปแบบของอัตราบิตผิดพลาดของอีควอไลเซอร์ แบบผลตอบสนองอิมพัลส์จำกัดที่มีจำนวน 11 แท็ป กับอีควอไลเซอร์ในแบบที่ได้นำเสนอ ค่าอัตราบิตผิดพลาดที่ได้ใกล้เคียงกัน อย่างไรก็ตามอีควอไลเซอร์ในแบบที่ได้นำเสนอมีจำนวนแท็ปที่น้อยกว่า

นอกจากนั้น ได้นำเสนอวิธีการกู้สัญญาณทางเวลาด้วยการวนซ้ำแบบมอดดิฟายด์เพอเซอร์วายเวอร์ (modified per-survivor iterative timing recovery: MPS-ITR) ซึ่งได้นำเอาประโยชน์ที่ได้จากวิธีการแยกพรีแอมเบิล (preamble) แบบใหม่มาใช้ เพื่อให้ทำงานร่วมกับเพอเซอร์วายเวอร์โปรเซสซิ่งซอฟต์เอาท์พุตวิเทอร์บิอัลกอริทึม (per-survivor processing soft-output Viterbi algorithm: PSP-SOVA) โดยทั่วไปพรีแอมเบิลจะถูกวางเอาไว้หน้าส่วนที่เป็นข้อมูลและทำงานร่วมกับวงจรเฟสล็อกลูป (phase-locked loop: PLL) ในช่วงของการได้มา (acquisition) เพื่อหาค่าออฟเซตทางเวลาและค่าออฟเซตทางความถี่ที่แฝงมากับข้อมูลในช่วงเริ่มต้น อย่างไรก็ตามวิธีการกู้สัญญาณทางเวลา

แบบ MPS-ITR ได้แยกพรีแอมเบิลออกเป็นสองส่วน ส่วนแรกได้ถูกวางเอาไว้หน้าส่วนที่เป็นข้อมูลเช่นปกติ ในส่วนที่สองได้ถูกแยกออกเป็นกลุ่มย่อยหลายๆกลุ่ม จากนั้นในแต่ละกลุ่มก็จะถูกแทรกลงไปในส่วนที่เป็นข้อมูล วิธีการแยกพรีแอมเบิลเช่นนี้จะถูกนำไปใช้ประโยชน์ในการปรับค่าการคำนวณเมทริกสาขาใน PSP-SOVA เพื่อทำให้มั่นใจว่าเส้นทางที่เหลืออยู่ได้เกิดขึ้นในทิศทางที่ถูกต้อง ผลที่ได้จากการจำลองระบบชี้ให้เห็นว่า วิธีการกู้สัญญาณทางเวลาแบบ MPS-ITR ให้สมรรถนะที่ดีกว่าวิธีการกู้สัญญาณทางเวลาแบบที่ใช้งานทั่วไป อีกทั้งยังมีสมรรถนะดีกว่าวิธีการกู้สัญญาณทางเวลาด้วยการวนซ้ำแบบเพอเซอร์วายเวอร์เต็มความซับซ้อน (full per-survivor iterative timing recovery: Full PS-ITR) และแบบลดความความซับซ้อน (reduced-complexity per-survivor iterative timing recovery: PS-ITR) โดยเฉพาะอย่างยิ่งเมื่อระบบมีจิตเตอร์ทางเวลา (timing jitter) เกิดขึ้นมาก จากนั้นได้แสดงให้เห็นว่าวิธีการกู้สัญญาณทางเวลาแบบ MPS-ITR สามารถแก้ไขไซเคิลสลิป (cycle slips) ได้อย่างมีประสิทธิภาพมากกว่าวิธีอื่นๆ นอกจากนั้นแล้วยังนำเอาระบบกู้สัญญาณทางเวลาด้วยการวนซ้ำทุกวิธีไปวิเคราะห์ในระบบการบันทึกข้อมูลเชิงแม่เหล็กเสมือนจริงสำหรับช่องสัญญาณการบันทึกแนวตั้ง พร้อมทั้งทำการเปรียบเทียบสมรรถนะกับระบบการกู้สัญญาณทางเวลาแบบที่ใช้งานทั่วไปด้วย

ในลำดับสุดท้าย ได้นำเสนอวิธีการลดความซับซ้อนระบบกู้สัญญาณทางเวลาแบบ MPS-ITR (reduced-complexity modified per-survivor iterative timing recovery) ด้วยการประยุกต์ Modified PSP-SOVA ให้ทำงานอยู่บนพื้นฐานของอัลกอริทึมเอ็ม (M-algorithm) เพื่อให้ระบบกู้สัญญาณทางเวลาดังกล่าวสามารถนำไปประยุกต์ใช้ในชิปช่องสัญญาณอ่านได้จริง ผลที่ได้จากการจำลองระบบชี้ให้เห็นว่า เมื่อทุกวิธีมีความซับซ้อนที่เท่ากันแล้ว วิธีการกู้สัญญาณทางเวลาแบบลดความซับซ้อนมีสมรรถนะที่ดีกว่าระบบกู้สัญญาณทางเวลาวิธีอื่นๆ

Thesis	A Novel Infinite-Impulse Response Equalizer Design and Iterative Timing Recovery for Magnetic Recording Channels
Student	Mr. Chanon Warisarn
Student ID.	49060053
Degree	Doctor of Engineering
Program	Electrical Engineering
Year	2011
Thesis Adviser	Assoc. Prof. Dr. Pornchai Supnithi
Thesis Co-adviser	Assoc. Prof. Dr. Piya Kovintavewat

ABSTRACT

In this thesis, we propose an infinite-impulse response (IIR) equalizer for shaping the partial response (PR) targets and a modified per-survivor iterative timing recovery (MPS-ITR) scheme, in the perpendicular magnetic recording channels. The equalizer is employed to shape the readback signal to a predetermined target before performing a maximum-likelihood (ML) equalization by the Viterbi detector.

First, a method of designing the infinite-impulse response (IIR) equalizer in the perpendicular magnetic recording channels will be designed and discussed. Based on a minimum-mean squared error (MMSE) approach, the coefficients of the IIR equalizer are explicitly derived. We compare the performance of the designed IIR filter with that of a finite-impulse response (FIR) equalizer in the presence and absence of the media jitter noise. Results indicate that for a small number of equalizer taps, the proposed IIR equalizer outperforms the FIR equalizer at moderate to high jitter noise levels. When compared with an 11-tap FIR equalizer, the proposed IIR counterpart can achieve a similar bit-error rate (BER) performance, but it requires fewer equalizer taps.

Then, we will describe the MPS-ITR scheme, which exploits a new split-preamble strategy in conjunction with a per-survivor processing soft-output Viterbi algorithm (PSP-SOVA). The conventional split-preamble strategy places a preamble at the beginning of a data sector and uses it to run a phase-locked loop (PLL) during acquisition to determine an initial phase/frequency offset. However, the MPS-ITR scheme splits the preamble into two parts. The first part is placed at the beginning of the data sector, whereas the second part is divided into small clusters, each of which is then embedded uniformly within the data stream. This split preamble is utilized to

adjust the branch metric calculation in PSP-SOVA to ensure that the survivor path occurs in a correct direction. Results indicate that the MPS-ITR scheme yields a better performance than the conventional receiver, the full complexity per-survivor iterative timing recovery (Full PS-ITR), and the reduced complexity per-survivor iterative timing recovery (PS-ITR) scheme, especially when the timing jitter is large. Then, we also show that the MPS-ITR scheme can automatically correct the cycle slips much more efficiently than the others. In addition, all the iterative timing recovery schemes will be investigated in the realistic magnetic recording systems for perpendicular recording channels and then compared with the conventional schemes.

Finally, we propose a reduced-complexity modified per-survivor iterative timing recovery (MPS-ITR-M) scheme. This scheme is obtained by applying the M-algorithm to the modified PSP-SOVA to make it more implementable in read channel chip. The numerical results indicate that when all schemes have same complexity, the MPS-ITR-M scheme yields better performance than other schemes.

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to Assoc. Prof. Dr. Pornchai Supnithi and Assoc. Prof. Dr. Piya Kovintavewat for being excellent advisors and great people. They have guided me through the last four years with skill and patience, and I am the better for it. Working with them has taught me the importance of attention to detail, of clarity of thought and expression. I would also like to express my sincerely appreciation to Assoc. Prof. Narong Hemmakorn and Assoc. Prof. Nipha Leelaruji for all the support and the generousness.

I am extremely thankful to everyone in my family, especially my parents. They have always taught me to be patient, diligent, concentrated, and cheerful. Utmost regards are given to my beloved brothers and sister for their understanding and support. I am truly blessed with the great encouragement and support from my wife, Bhiriyakorn, and my daughter, Bhacharanon, whom I cherished dearly.

I would also like to thank National Electronics and Computer Technology Centre (NECTEC), National Science and Technology Development Agency (NSTDA) and I/U CRC in Data Storage Technology and Applications (D*STAR), now known as College of Data Storage Innovation, King Mongkut's Institute of Technology Ladkrabang under grant HDDA50-001D for its financial support during my Ph.D. study. Acknowledge SEAGATE Technology Thailand (Khun Kanawat Senanan, Dr. Chitiporn Pupaichitkul etc.).

Special thanks are also given to Yupin, Wannaree, Lin, Wicharn, and others in the Communications and Storage Research Group (CSRG) Laboratory at KMITL. We had many fruitful discussions which consequently shed light on many questions I faced with.

Chanon Warisarn

TABLE OF CONTENTS

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (cont.)

LIST OF FIGURES (cont.)

CHAPTER 1

INTRODUCTION

1.1 Literature Review

In the previous generations, the hard disk drive technology is based on the longitudinal magnetic recording (LMR) in which the medium magnetization is parallel to the disk plane. The quest to increase the areal density as well as the need to combat the super-paramagnetism causes the shift of the longitudinal recording to the perpendicular recording. As the areal density increases, the signal processing becomes more complex in order to maintain the acceptable performance and reliability. Therefore, highly efficient signal processing systems for read-channel chip in hard disk drive is necessary. Equalizer and timing recovery are very important parts in the read-channel chip. A target-shaping equalizer is placed after the A/D converter and is used to shape the overall channel response to a desire response. Timing recovery, on the other hand, deals with the raw analog readback signal and it is employed in a synchronization process. Some previous works have investigated equalizer design and invent the new timing recovery schemes, which can be summarized as follow.

Partial-response maximum-likelihood (PRML) technique [1] is widely used for data detection process in the perpendicular magnetic recording channels. In practice, this technique employs a finite-impulse response (FIR) equalizer to shape the readback signal to a predetermined target before performing a maximum-likelihood (ML) equalization by the Viterbi detector [2].

An infinite impulse response (IIR) equalizer has previously been studied in the literature [4], and references therein. For instance, the IIR modeling was considered in the decision feedback equalizer design [5] to reduce the number of filter taps. Also, in [6], the performance of employing the continuous-time adaptive IIR equalizers for EPR4 channels was investigated. An algorithm for the approximation of FIR filter by IIR filter and the direct method for converting FIR filter with low non-zero taps into IIR filter using the predetermined table are proposed in [7 – 8], respectively.

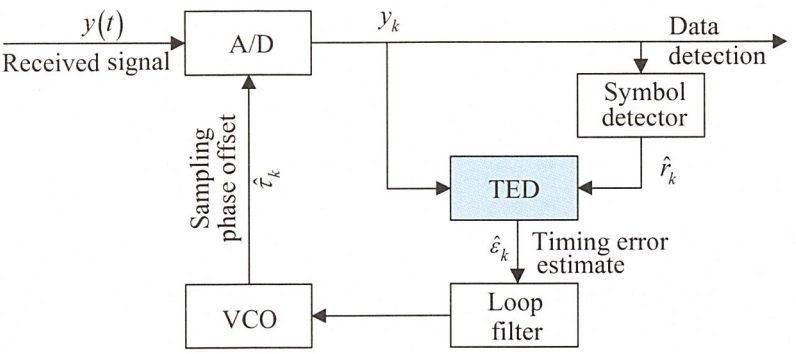


Figure 1.1 A conventional timing recovery system.

In practice, a conventional timing recovery in magnetic recording systems employ with a 2nd-order phase-locked loop (PLL), which consists of a timing error detector (TED), a loop filter, and a voltage controlled oscillator (VCO), as illustrated in Figure 1.1. A number of timing recovery systems have been proposed in the literature [3], [9], [10]. Most of them can be categorized into two types, namely deductive timing recovery and inductive timing recovery, depending on where the embedded timing information in the received analog signal is extracted [3].

Specifically, the deductive (or feed-forward) timing recovery extracts the timing information before the sampler, whereas the inductive (or feedback) one extracts the timing information after the sampler. However, both timing recovery architectures utilize a PLL to search for the location to sample the received signal. Since the inductive timing recovery is widely used in many applications [3], it will then be referred to as conventional timing recovery, of which the architecture is shown in Figure1.1.

A widely used TED is the Mueller and Müller (M&M) TED [11]. In practice, the M&M TED performs well at a high signal-to-noise ratio (SNR). However, at low SNR, the system is plagued by cycle slips and a severe timing jitter. To solve this problem, the maximum a *posteriori* (MAP)-based timing recovery has been proposed in [12], which uses a new TED algorithm. This TED employs maximum a posteriori and minimum mean-squared error techniques to estimate the timing information. Results have indicated that it can reduce a cycle-slip rate when compared with conventional timing recovery at the expense of increased computation complexity. It has been shown in [12] that the complexity of this new TED is about 4 times higher than that of the M&M TED.

Timing recovery normally performs in two modes, namely, acquisition and tracking modes [3]. The acquisition mode is performed to acquire the initial timing offset with an aid of a

preamble (or a training sequence of known symbols), whereas the tracking mode is performed to refine the timing estimates based on a user data sequence (unknown symbols). To improve the performance of acquisition mode, Hwang *et al.* [13] proposed an extended Kalman filter (EKF) to estimate and correct possibly large initial timing errors. This proposed method converges faster than a conventional timing recovery scheme, thus allowing us to reduce the length of the preamble used during acquisition mode.

It should be noted that all timing recovery architectures mentioned above performs timing recovery and maximum-likelihood (ML) equalization separately, thus performing unreliably when operating at a low SNR and a severe timing jitter. To improve their performances, Kovintavewat *et al.* [14] proposed a per-survivor timing recovery architecture, which performs timing recovery and ML equalization jointly. Furthermore, Zeng *et al.* [15] also proposed a trellis-based optimal baud-rate timing recovery loop based on a Markov model, whose structure is similar to [14]. This scheme has been shown to perform well at the expense of high complexity. Nonetheless, both schemes have been proposed for partial-response (PR) channels without error correction codes (ECCs).

The large coding gains of iterative ECCs enable reliable communication at very low SNRs. This means that timing recovery must be performed at an SNR lower than ever before. A conventional receiver performs timing recovery and turbo equalization [16] separately. Specifically, conventional timing recovery ignores the presence of ECCs and thus fails to work properly when the SNR is low enough. To improve the performance of the conventional receiver, Kovintavewat *et al.* [17] proposed a full-complexity per-survivor iterative timing recovery (Full PS-ITR) scheme, which jointly performs timing recovery, equalization, and error-correction decoding. It is realized by first applying the per-survivor processing (PSP) technique [18], to the Bahl, Cocke, Jelinek, and Raviv (BCJR) algorithm [19], resulting in a per-survivor BCJR equalizer, denoted as "PSP-BCJR" [17]. Then, PSP-BCJR iteratively exchanges soft information with a soft-in soft-out (SISO) decoder. Because of the PSP-BCJR has very high complexity, a reduced-complexity per-survivor iterative timing recovery (PS-ITR) scheme has been proposed [20], where a soft-output Viterbi algorithm (SOVA) [21] is used to instead of BCJR, resulting in a per-survivor SOVA equalizer, denoted as "PSP-SOVA". As investigated in [20], at low to moderate complexity, the PS-ITR performs better than both the Full PS-ITR and the conventional receiver. This is because the PS-ITR can automatically correct a cycle slip [3] with only a small number of turbo iterations.

Generally, to have good timing estimates before the user data sequence starts, the conventional split-preamble strategy places all known symbols at the beginning of a data sector [22]. Nonetheless, in [23] - [24], the known symbols are arranged in small contiguous clusters and are placed periodically in the data stream, subject to the power constraint on training. Additionally, Nayak *et al.* [25] proposed the optimal training symbol placement strategy to minimize a Cramér-Rao bound by splitting the known symbols into two halves and placing them at the beginning and at the end of a data sector. This split-preamble arrangement leads to a reduced frequency estimation error variance and greatly reduces the occurrence of cycle slips.

1.2 Significant of The Research

At high recording densities, the FIR equalizer with a large number of taps is required to function properly. Nevertheless, the total number of equalizer taps is practically limited by the maximum allowable loop delay in the timing recovery loop because a small loop delay provides more robust phase locking [3], which in turn improves the overall system performance. Furthermore, the benefits of the equalizer with fewer taps are three folds: (1) a smaller area on the silicon chip, (2) a shorter optimization time of read-channel chip during the production, and (3) there is a small delay in the timing loop. Consequently, in this paper, we directly design the digital IIR equalizer based on the minimum mean-square error (MMSE) approach to reduce the number of equalizer taps, and then compare its performance with the conventional FIR equalizer in a full turbo equalization setting.

As for timing recovery, it is the process of synchronizing the sampler with the received analog signal. Sampling at the wrong times can have a devastating impact on overall system performance. Therefore, the quality of synchronization is very important for all applications. To improve the performance of conventional timing recovery, a simple timing recovery architecture which consists of two timing recovery blocks running in parallel is proposed. We refer to this timing recovery architecture as "bi-directional timing recovery". It can be seen in simulations that the bi-directional timing recovery can help improve the system performance when compared with conventional timing recovery.

In general, the PS-ITR uses the conventional split- preamble strategy, which places a preamble at the beginning of a data sector and uses it to run a PLL [3] during acquisition to find the initial timing offset. To further improve the performance of the PS-ITR, The modified per-survivor iterative timing recovery (MPS-ITR) scheme has been proposed. In essence, the MPS-

ITR performs same as the PS-ITR, except that a modified PSP-SOVA is utilized instead of PSP-SOVA. This modified PSP-SOVA employs a new split-preamble strategy, which splits a preamble into two parts. The first part is placed at the beginning of a data sector, while the second part is divided into many small clusters, each of which is therefore embedded uniformly within the user data sequence. This split preamble is utilized to adjust the branch metric calculation in PSP-SOVA to guarantee that the survivor path occurs in a correct direction. It can be shown that the MPS-ITR scheme not only yields a better performance than a conventional receiver with separate timing recovery and turbo equalization, and the iterative timing recovery scheme proposed in [17, 20], especially when the timing jitter is large but also automatically correct the cycle slips much more efficiently than the others.

However, it is apparent that both the PS-ITR scheme and the MPS-ITR scheme have very high complexity. Therefore, to make it more implementable in real-life applications, we apply the M-algorithm [26] to the MPS-ITR so as to reduce its complexity. This reduced-complexity scheme will be referred to as reduced complexity modified per-survivor iterative timing recovery (MPS-ITR-M) algorithm. It can be shown that there is a tradeoff between the parameter M and the system performance. In addition, in the PR2 channel, for same complexity, the (reduced-complexity) MPS-ITR-M performs better than the (full-complexity) MPS-ITR.

1.3 Objective and Confinement of Thesis

The purpose of thesis is to propose a new digital IIR equalizer and new timing recovery schemes for the perpendicular magnetic recording channels. The digital IIR equalizer is directly designed based on the MMSE approach to reduce the number of equalizer taps. The new timing recoveries are proposed to develop the process of synchronizing the sampler with the received analog signal. The bi-directional timing recovery is proposed for uncoded partial response channels, while the MPS-ITR scheme and the MPS-ITR-M scheme are proposed for coded partial response channels.

The substances in detail of this thesis are provided as follows:

In Chapter 2, we provide an overview of the backgrounds on digital magnetic recording systems, basic communication channel models for magnetic storage, write and read processes. At the end of this chapter, we describe the noises and magnetic recording channel model.

In Chapter 3, we discuss a method of designing an IIR equalizer in perpendicular recording channels. Based on a MMSE approach, the coefficients of the IIR equalizer are explicitly

derived. Its performance will be shown by comparing with a FIR equalizer in the presence and absence of media jitter noise.

In Chapter 4, we discuss the concept of timing recovery to explain the principles of conventional timing recovery. To increase the efficiency of conventional timing recovery system, bi-directional timing recovery is proposed and described. Then, we will explain how to design the parameters of the phase locked loop using a simulation of a phase locked loop of a linearized PLL model. The performance of both methods will be shown and discussed at the end of this Chapter.

In Chapter 5, we will describe the MPS-ITR scheme and the modified PSP-SOVA. The performance comparison among the different iterative timing recovery schemes is provided. Then, we describe all iterative timing recovery schemes which will be investigated in realistic magnetic recording systems for perpendicular recording channels.

In Chapter 6, we propose the MPS-ITR-M scheme which is more implementable in real-life applications and their numerical results are also given and investigated at the end of this Chapter.

Finally, conclusions are provided in Chapter 7.

CHAPTER 2

BACKGROUNDS ON DIGITAL MAGNETIC RECORDING SYSTEMS

Digital magnetic recording has been employed in various storage applications, including hard disk drives (HDD), floppy disk drives, and tape drives. Magnetic recording typically involves a magnetic head and a recording medium for example in HDD, as shown in Figure 2.1. An inductive head consists of a horseshoe-shaped soft magnetic material with low coercivity and high permeability [27] around which coils are wound, and a recording medium that is normally comprised of a hard magnetic material with high coercivity.

There are two existing modes of magnetic recording, namely, longitudinal recording system [3] and perpendicular recording system [28]. In the previous generation, the HDD technology is based on longitudinal recording in which the medium magnetization is parallel to the disk plane as shown in Figure 2.1. Recently, the perpendicular recording is employed instead of longitudinal recording. To increase the areal densities in the longitudinal recording and boost the overall storage capacity, the data bits must be horizontally shrunk and packed more closely together. However, if the bit becomes too small, the magnetic energy holding the bit in place may also become so small that thermal energy can cause it to demagnetize, a phenomenon known as superparamagnetism. To avoid superparamagnetism, disk media manufacturers have been increasing the coercivity (the field required to write a bit) of the media. However, the fields that can be applied are limited by the magnetic materials making up the write head.

In the perpendicular recording, the magnetization of the disk, instead of lying in the disk's plane as it does in the longitudinal recording, stands on end, perpendicular to the plane of the disk. The bits are then represented as regions of upward or downward directed magnetization (in longitudinal recording, the bit magnetization lies in the plane of the disk and flips between pointing in the same and opposite directions of the head movement.). The media is deposited on a soft magnetic under-layer that functions as part of the write field return path and effectively produces an image of the recording head that doubles the recording field, enabling higher recording density than with longitudinal recording. This thesis only focuses on perpendicular magnetic recording because it is adopted in all commercial magnetic recording systems now.

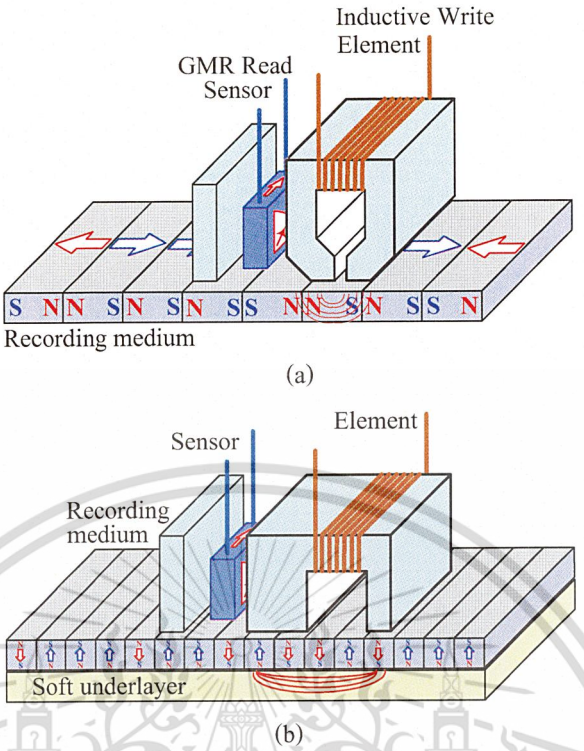


Figure 2.1 An illustration of the two modes of magnetic recording (a) longitudinal recording and (b) perpendicular recording.

2.1 Basic Communication Channel Model for Magnetic Storage

Design and analysis of coding and signal processing techniques require a suitable communications channel model for magnetic storage systems. Such a model should correctly reflect the essential physics of the read and write processes of magnetic recording, but must also provide a system-level description that allows convenient design, analysis and simulation of the communications and signal processing techniques under study.

Consider a general block diagram of a magnetic recording system for HDD as depicted in Figure 2.2. The massage bits are encoded by the ECC encoder to protect information bits from random noise. Currently, the Reed-Solomon (RS) [29, 30] code is a standard ECC in HDD. Then, the RS-encoded sequence is encoded again with the modulation encoder to control minimum and maximum distances between consecutive magnetic transitions [31]. The minimum distance constraint mitigates local medium noise and nonlinearity associated with crowded magnetic transitions, while the maximum distance constraint ensures that the signal arises often enough in the readback signal (the data sequence were not part of the d.c. component).

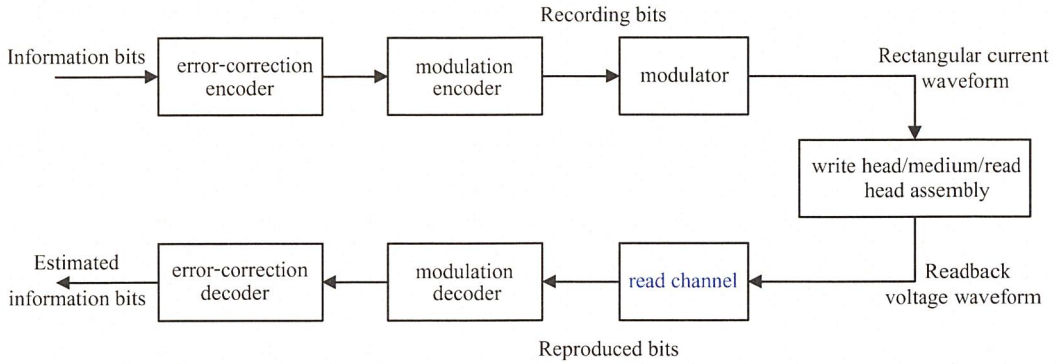


Figure 2.2 Simplified block diagram of a magnetic recording system/storage system.

The run-length limited (RLL) code [32] is often used for modulation encoding. The data sequence from the modulation encoder will be written in medium, called "recording bits". The modulated bit stream is converted into a rectangular current waveform (write current waveform) by the modulator and then stored in the medium in the form of a magnetization waveform.

During the read process, the data in the medium will be sensed by read head and then converted into the electrical signal. After preamplification, we obtain the readback signal. The readback voltage waveform will be created when the read head move through in response to a written transition. The readback is then passed through what is generally called the read channel in the data storage community. The read channel consists of a low pass filter (LPF), a sampler driven by a phase locked loop or analog-to-digital converter, an equalizer and a symbol detector (in a broader definition the read channel also includes the modulation encoder/decoder and, possibly, some auxiliary inner error correction or error detection encoder/decoder). The detected bit sequence is then sent to the modulation decoder and finally to the error correction decoder.

2.2 Write and Read Processes

2.2.1 Write Process

During the write process, the data bits are converted into a rectangular current waveform called a write current (as shown in Figure 2.3). This write current is applied to the windings of the write head to produce a magnetic write field in the medium near the head gap. The write field must be larger than the medium coercivity to magnetize the medium along the field direction. By switching the direction of the write field (or the write current), magnetization transitions can be written in the medium.

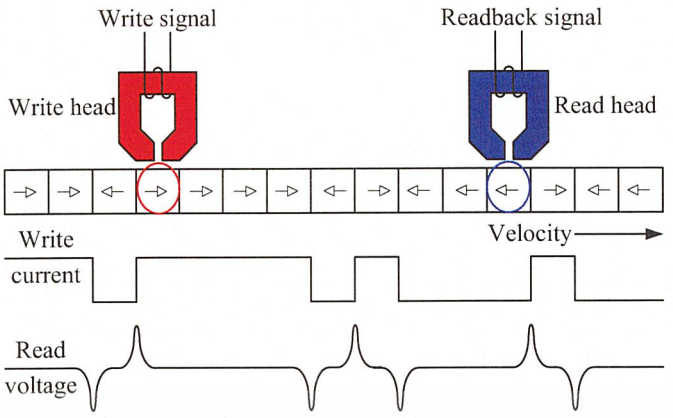


Figure 2.3 Illustration of the principle of magnetic recording.

Commercial digital recording systems normally employ binary saturation recording, i.e., the magnetization saturated on the medium in only one direction or the opposite. This is because if more than two data levels were recorded, nonlinearities would cause a major problem and signal-to-disturbance ratios would diminish considerably [3].

2.2.2 Read Process

During the read process, the read head senses the change in the flux via the transitions of the magnetization pattern, resulting in an induced voltage pulse in the coil because of the Faraday's law. For an isolated transition, the read head produces a read voltage pulse, $g(t)$, or its inverse, $-g(t)$, depending on the direction of the transition (see Figure 2.3). The pulse $g(t)$ is commonly known as the transition response [3], which has finite amplitude and a finite half-amplitude pulse width. The transition response for longitudinal recording (also known as the Lorentzian pulse) is given by [3]

$$g(t)=\frac{1}{1+\left(\frac{2t}{\mathrm{PW}_{50}}\right)^2}, \tag{2.1}$$

where PW_{50} determines the width of $g(t)$ at half of its peak value. For perpendicular recording, we are interested in a transition response of the form [33]

$$g(t)=\operatorname{erf}\left(\frac{2t\sqrt{\ln 2}}{\mathrm{PW}_{50}}\right), \tag{2.2}$$

where $\operatorname{erf}(\bullet)$ is an error function defined by $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ [34], and PW_{50} determines the width of the derivative of $g(t)$ at half its maximum. In the context of magnetic recording, the ratio $ND = PW_{50}/T$ (where T is the bit duration) represents a normalized recording density [3], which defines how many data bits can be packed within the resolution unit PW_{50}.

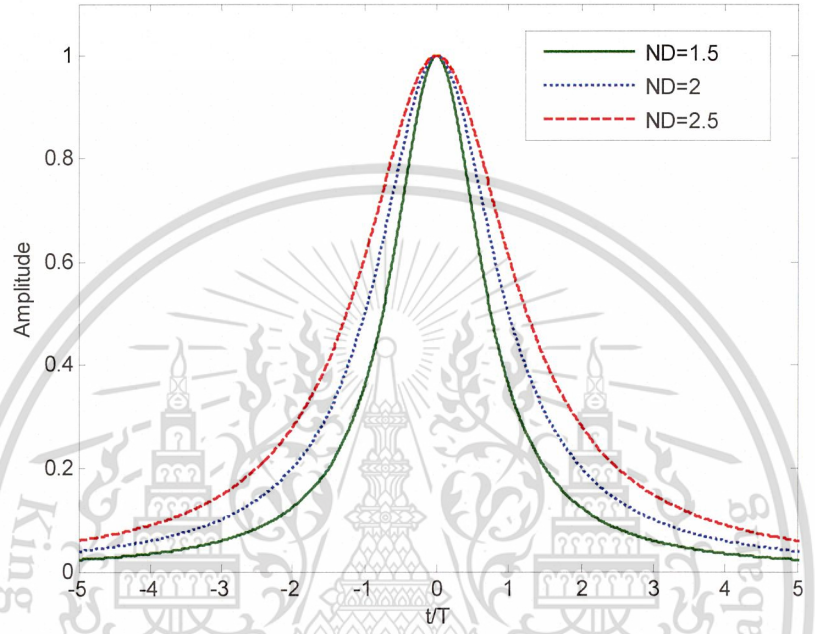


Figure 2.4 Transition responses for longitudinal recording

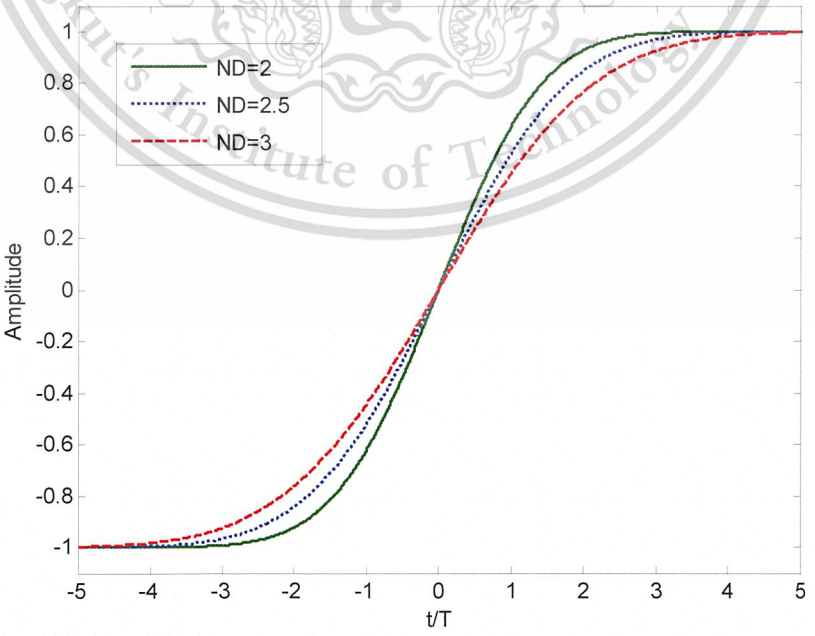


Figure 2.5 Transition responses for perpendicular recording.

The transition responses of longitudinal and perpendicular recording channels are plotted in Figure 2.4 and Figure 2.5, respectively, for different NDs. Clearly, the transition response spans many symbol intervals as ND increases. This implies that the effect of intersymbol interference (ISI) becomes more severe as ND increases.

Additionally, the response of the head to an isolated bit is commonly known as the dibit response [3], which is expressed as

$$m(t) = g(t) - g(t-T), \tag{2.3}$$

as shown in Figure 2.6 for longitudinal recording and perpendicular recording, respectively.

It is easy to show that the frequency response of $m(t)$ for longitudinal recording is given by

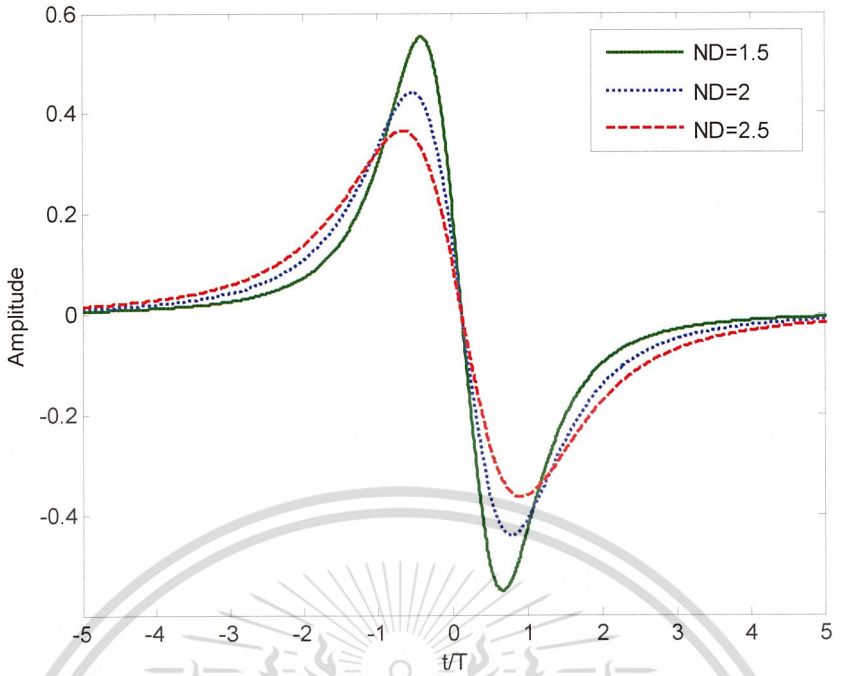
$$M(\Omega) = \exp(-\pi|\Omega|\mathrm{ND}) \bullet \{1 - \exp(-j2\pi\Omega)\}, \tag{2.4}$$

whereas for perpendicular recording it is

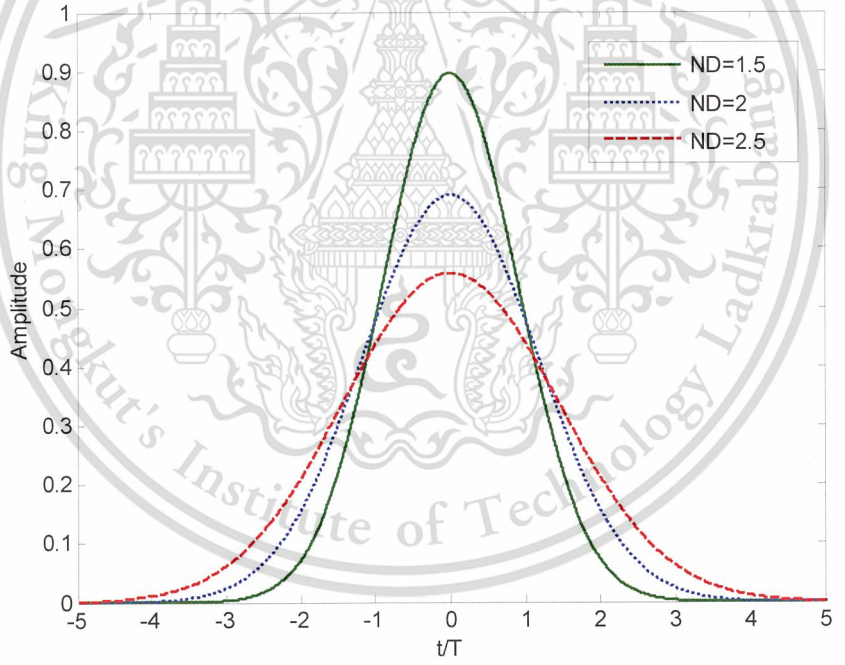
$$M(\Omega) = \frac{T}{j\pi\Omega} \bullet \exp\left(-\frac{\pi^2\Omega^2 ND^2}{ln16}\right) \bullet \{1 - \exp(-j2\pi\Omega)\}, \tag{2.5}$$

where $\Omega = fT$ is a normalized frequency variable, f is a frequency variable in Hertz, $|x|$ takes on the absolute value of x, and $j = \sqrt{-1}$ is an imaginary number.

Figure 2.7 shows the normalized frequency responses of the dibit responses for different ND's. Apparently, the signal energy becomes more concentrated at low frequencies as ND increases for both channels. Furthermore, a longitudinal recording channel exhibits a spectral null at d.c., while a perpendicular recording channel contains a d.c. component.



(a)



(b)

Figure 2.6 Dibit responses for (a) longitudinal and (b) perpendicular recording.

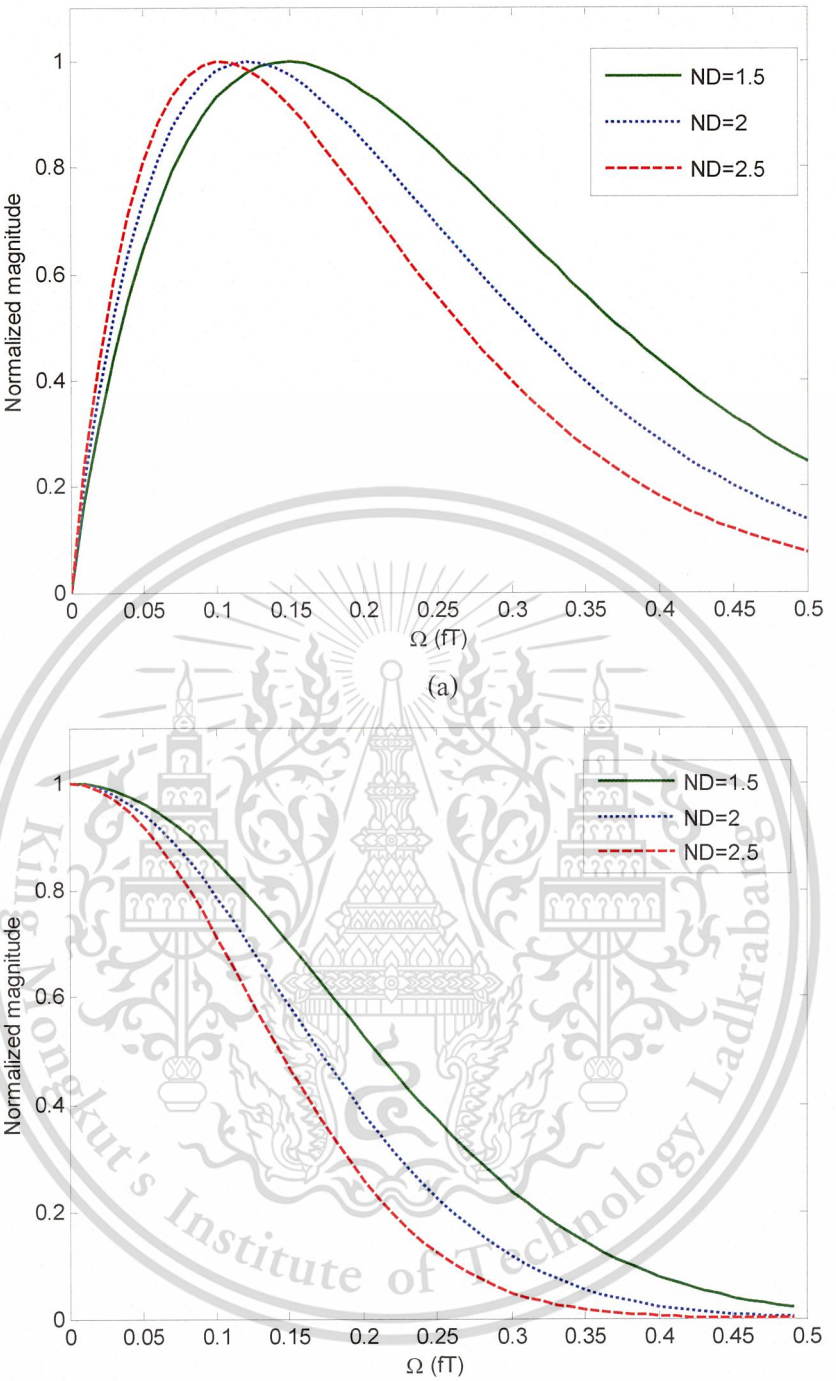


Figure 2.7 Frequency responses of the dibit responses for (a) longitudinal and (b) perpendicular recording.

2.3 Noises in Magnetic Recording System

In practice, the readback signal which is obtained from read head will be distorted due to the many factors such as noise, distortion, and interference. These events occur during the recording process, which can be summarized as follow.

2.3.1 Medium Noise

The magnetic medium is a main source of noise in magnetic recording system. The readback signal in magnetic recording is directly related to the medium magnetization distribution. Therefore, medium noises are due to the fluctuation or uncertainties in the medium magnetization. Generally, the medium noise can be classified in to the three main types:

2.3.1.1 Transition noise

Transition noise occurs from the magnetization fluctuation concentrated near the recorded transition centers. It is the dominant noise source in the thin film disks. The transition noise depends on the recording pattern. Normally, the severity of transition noise will be increased when the capacity of hard disk drive is higher. Transition noise can be analyzed by numerical micro-magnetic modeling [35, 36]. Due to the "Zigzag" nature of the magnetic transitions in the thin film disks, the transition center, defined as the average transition center across data track, becomes uncertain. This uncertainty is normally called transition position jitter. The transition position jitter will lead to transition noise. The zigzag nature also causes fluctuation in read pulse shape, resulting in additional transition noises.

2.3.1.2 Particulate or granularity noise

Particulate or granularity noise occurs due to the random dispersion of magnetic particles or grains in magnetic medium. This is the dominant noise source in magnetic tapes, floppy disks and particulate thin film hard disks. The particulate noise in magnetic media is typically stationary, meaning that the noise is independent of the locations along the data track.

2.3.1.3 Modulation noise

Modulation noise occurs due to the magnetization fluctuation proportional to recorded magnetization between magnetic transitions, which can be observed in both particulate and continuous thin film disks. The modulation noise in magnetic media is also non-stationary.

2.3.2 Magnetic Head Noise

When a magnetic read head is electrically activated, far from magnetic disks, it can still generate considerable noise, there are noises from head itself and from head preamplifier (electronic noise) which may magnify read voltage. Head noise occurs from the fluctuations of magnetic domain walls of the head core material (*Barkhausen noise*), or may also occurs from the resistive dissipation in the head (*Johnson noise*) [27].

The electronic noise occurs due to the random fluctuations in time of electric carriers. Normally, the preamplifier generates the dominant electronic noise in data detection circuitry. In magnetic recording system, thermal noise and shot noise often dominates electronic noise. Both thermal noise and shot noise might be more or less than white noise, which depend on frequency. In other words, the white noise spectrum density is a constant at all frequencies.

2.3.3 Nonlinearity Distortion

Nonlinearity is a phenomenon that causes nonlinear superposition. In the magnetic write process, nonlinearities include partial erasure, nonlinear transition shift, and hard transition shift. Nonlinearities are described as following.

2.3.3.1 Partial Erasure

The percolation occurs at high density recording as shown in Figure 2.8. The regions across the track become partially erased and the amplitude loss results from an effective trackwidth narrow. Numerical micro-magnetic modeling and experimental observations confirmed that large exchange coupling produces large, but few, percolation regions while low exchange coupling produces many percolation regions [27].

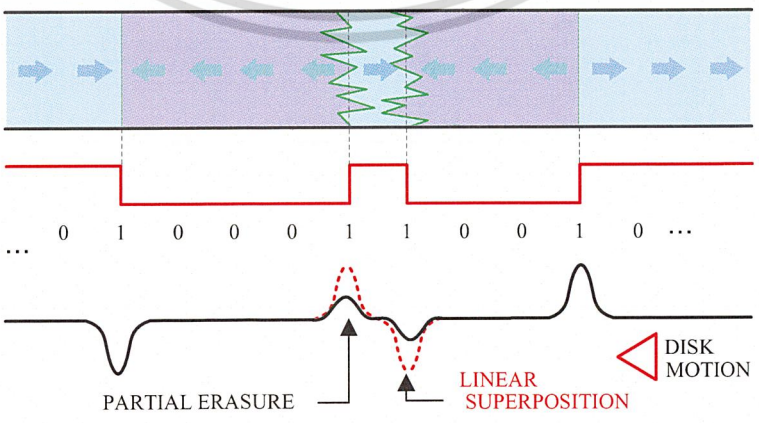


Figure 2.8 Partial erasures of two closely spaced transitions (dibit).

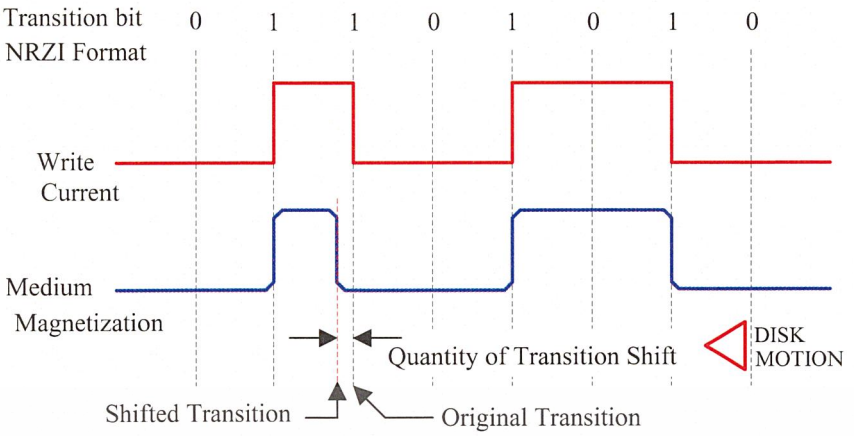


Figure 2.9 Illustration of the NLTS effect.

2.3.3.2 Nonlinear Transition Shift (NLTS)

Nonlinear transition shift or bit shift refers to the transition shift due to the demagnetizing field from previously written transition. Since the magnetizing field drops quite fast with the distance, the location of the transition is shifted mostly by the perturbing demagnetizing from the nearest-neighbor previous transition.

The neighboring magnetic transitions must have opposite magnetic charge sign, and their mutual magnetostatic interaction is manifested by demagnetizing field. Since opposite charges attract each other, the two magnetic transitions (dibit) must be shift closer. In other word, NLTS always causes the transition to be written earlier than desired as illustrated in Figure 2.9. One need not worry about the effect if the dibit is far apart, but it is very serious at high recording densities. However, we can employ the write precompensation to manage the problem of NLTS, i.e., to intentionally delay switching the write current so that the resulting transition centre is in the desired location.

2.3.3.3 Hard Transition Shift

The erasure of old information in digital magnetic recording system is performed by directly writing new data pattern over the old data pattern. In this process, a hard transition will be written when the head field is opposite to the direction of the incoming magnetization, while an easy transition will be written when the head field is in the same direction of the incoming magnetization as illustrated in Figure 2.10.

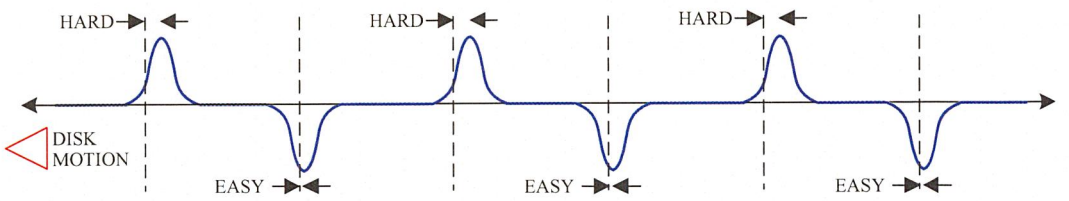


Figure 2.10 Illustration of the example of the hard transition shift (HTS) and the easy transition shift (ETS).

In practice, the hard transition is difficult to write because it requires more head field to saturate the magnetization formed at the leading edge of the write bubble, while the easy transition will not occur during the writing. The hard transition shift is similar to nonlinear bit shift due to previous transitions, the demagnetizing field from the leading edge of the write bubble leads to hard transition shift. The hard transition always gets shifted later than desired in the absence of other nonlinear effects. The magnitude of the hard transition shift greatly decreases with the increasing size of the writing bubble as the magnetic charges become farther apart. Hard transition shift cannot be removed by the write precompensation since the incoming magnetization direction is unknown during the write process.

2.3.3.4 Overwrite

In digital magnetic recording to erase the old information, the old data pattern will be overwritten with a new data pattern. The basic requirement in overwriting is that the write field must be sufficient to reduce any residual old information to the levels low enough to ensure that it will not a cause any error while reading the new data.

The most commonly cited overwrite ratio is defined as follows [27]. First, write a square wave pattern at frequency f_1, then overwrite with a square wave pattern at frequency f_2 (most often, $f_2 = 2f_1$). The overwrite ratio ("f_1/ f_2" ratio) is the residual f_1 signal $V_{f2}(f_1)$ divided by the original f_1 signal level $V_{f1}(f_1)$.

Normally, the overwrite ratio is given in decibels (dB): $OW = 20 \log V_{f2} / V_{f1}$. The absolute value of the overwrite ratio is $|OW| = 20 \log V_{f1} / V_{f2}$. Note that both signals are measured at old frequency f_1. For a typical magnetic recording system, the required OW is <-30 dB, or $V_{f2} < 0.032\ V_{f1}$. The common sources of overwrite residual signal are listed in the following.

Residual recorded due to in complete erasure of the f_1 signal when the write current is no sufficient, hard transition shift of f_2 signal at f_1 due to the demagnetizing fields from the incoming f_1 pattern, and residual track edge effects due to incomplete erasure of track-edge magnetization. The first two sources can be minimized by using a "good" write head with a sufficient write field, and the second source can be reduced by a "write wide, read narrow" scheme.

2.3.4 Types of errors

There are mainly two types of errors that occur in magnetic recording system, that are single-bit errors and bursts of errors. Normally, single bit errors occur caused a short-duration noise event, which results in an extra pulse or a missing pulse as shown is in Figure 2.11 (a). Burst of errors occur when a bit's group is detected erroneously an example of a burst of three errors in Figure 2.11 (b). The defect of magnetic medium such as a scratch of a defective spot spanning over many bit periods might be the cause of burst of errors. However, burst of errors may also occur at the output of the ML detector in a PRML channel. The ML detector decision contains a group of channel bit. Any event in the ML detector will result in detecting a wrong sequence of bits.

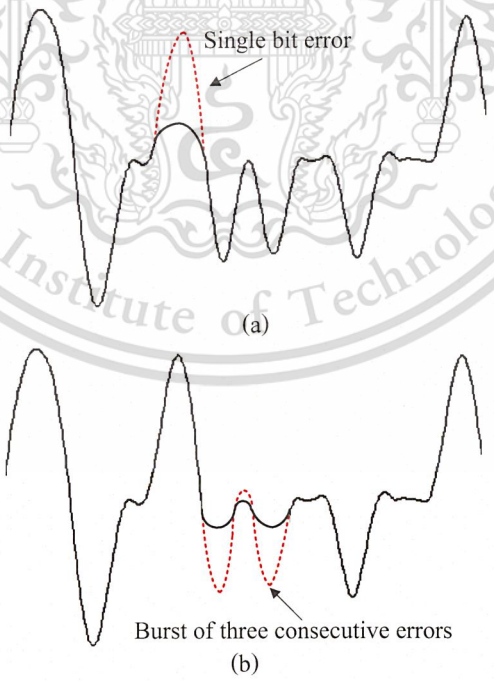


Figure 2.11 Illustration of a single bit error and a burst of errors.

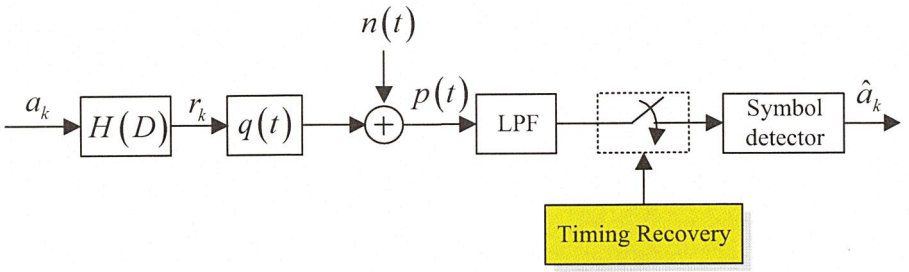


Figure 2.12 An ideal magnetic recording channel model.

2.4 Magnetic Recording Channel Model

In general, the channel model of the magnetic recording system can be modeled as main two types, there are an ideal channel model and a realistic channel model. The ideal channel model is often used to study and analyze the basics of signal processing systems of the HDD, since the model is redundant. While the realistic channel model has employed to investigate in many researches on magnetic recording systems [37-38], which can be described as follows.

2.4.1 Ideal Channel Model

If we assume that the magnetic recording system is perfectly equalized (perfect equalization). The ideal channel model can be illustrated in Figure 2.12. The binary input sequence $a_k \in \{\pm 1\}$ with bit period T is modulated with ideal zero-excess-bandwidth Nyquist pulse, $q(t) = \sin(\pi t / T) / (\pi t / T)$ [39]. Then, the signal is disturbed with noise. At the receiver, the readback signal, $p(t)$ is filtered by a low-pass filter (LPF) to eliminate the out-of-band noise and is sampled at the instants controlled by the timing recovery block. Finally, the sampler output is passed through the symbol detector to determine the most likely input sequence.

The generally accepted PR target [40] for longitudinal recording is of the form $H(D) = (1-D)(1+D)^n$ [3], whereas the PR target for perpendicular recording is $H(D) = (1+D)^n$ [41], where n is an integer. Apparently, the $(1-D)$ is not needed for perpendicular recording because the perpendicular recording channel contains a d.c. component. Table 2.1 illustrates the typical PR targets for longitudinal and perpendicular recording. Figure 2.13 and Figure 2.14 compare the frequency responses of the different targets. It is clear that as ND increases, a larger value of n is required because the effect of ISI becomes more severe at high ND. Hence, at high ND, a longer target allowing more controlled ISI will provide a better match to the channel response than a shorter target.

Table 2.1 The typical PR targets in HDD.

PR Target	$n = 1$	$n = 2$	$n = 3$
Longitudinal recording	$1 - D^2$ PR4 [1 0 -1]	$1 + D - D^2 - D^3$ EPR4 [1 1 -1 -1]	$1 + 2D - 2D^3 - D^4$ EEPR4 [1 2 0 -2 -1]
Perpendicular recording	$1 + D$ PR1 [1 1]	$1 + 2D + D^2$ PR2 [1 2 1]	$1 + 3D + 3D^2 + D^3$ EPR2 [1 3 3 1]

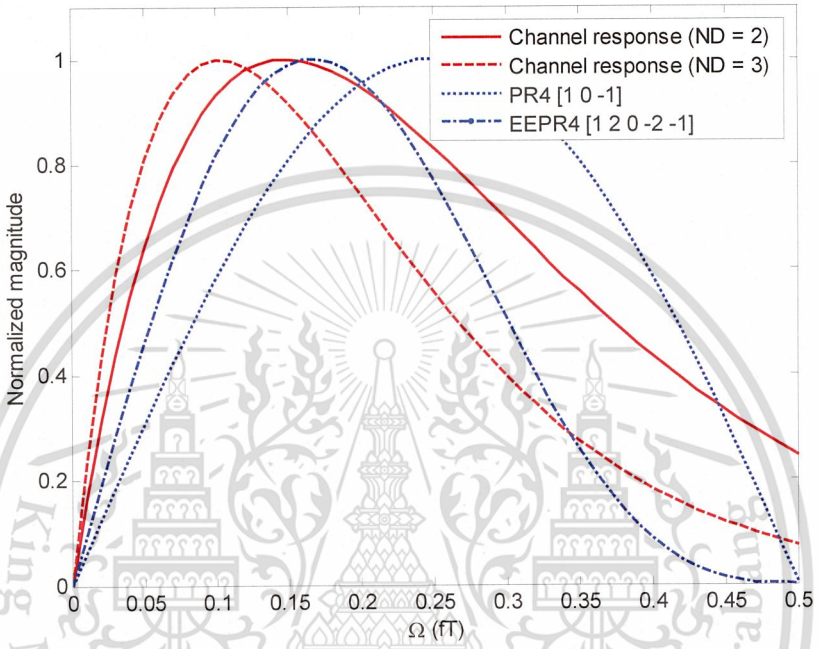


Figure 2.13 Frequency responses of different targets for longitudinal recording channels.

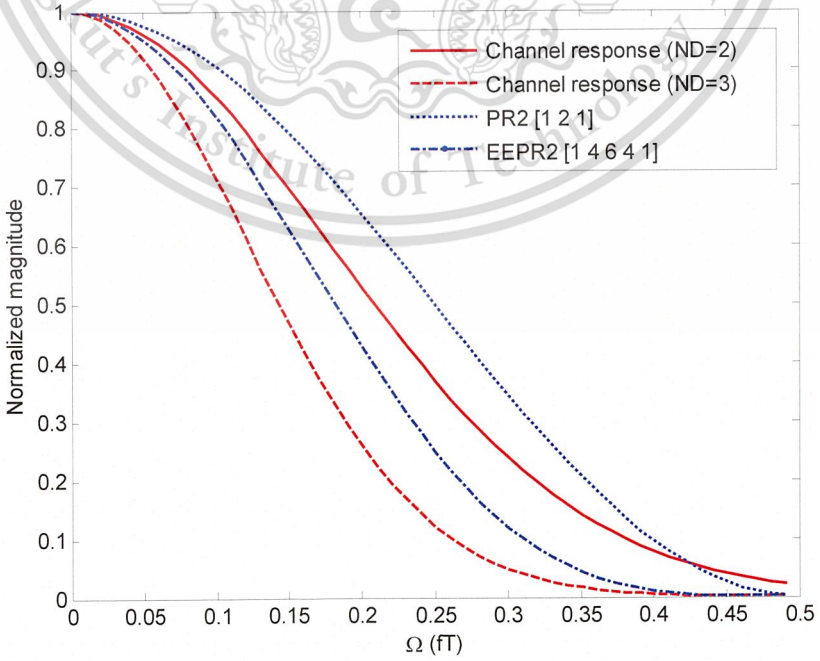


Figure 2.14 Frequency responses of different targets for perpendicular recording channels.

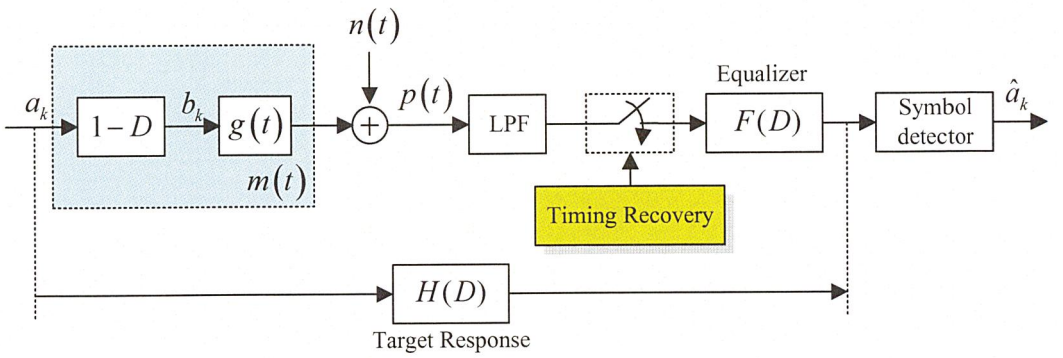


Figure 2.15 A realistic magnetic recording channel model.

2.4.2 Realistic Channel Model

A magnetic recording system can also be expressed, as depicted in Figure 2.15. This system model will be referred to as a realistic channel model because it represents all the components that are employed in magnetic recording read-channel chip architectures. A binary input data sequence $a_k \in \{\pm 1\}$ with bit period T is filtered by an ideal differentiator $1-D$, where D is the delay operator, to form a transition sequence, $b_k \in \{-2, 0, 2\}$, where $b_k \in \{\pm 2\}$ corresponds to a positive or a negative transition, and $b_k = 0$ corresponds to the absence of a transition. The transition sequence b_k passes through the channel represented by the transition response $g(t)$ and is corrupted by the noise $n(t)$. The readback signal, $p(t)$ is filtered by a low-pass filter (LPF) to eliminate the out-of-band noise and is sampled at the instants controlled by the timing recovery block. Then, the sampler output is equalized by an equalizer so that the equalizer output resembles the desired sample. Eventually, the symbol detector performs ML equalization to determine the most likely input sequence.

A widely used ML detector in magnetic recording systems is a Viterbi detector [2]. Since the complexity of the Viterbi detector grows exponentially with channel memory, the equalizer is usually employed to shape the overall channel impulse response into a shorter response called the target response [3, 42], $H(D)$, thus reducing the complexity of the Viterbi detector. The technique of using the equalizer in conjunction with the Viterbi detector is commonly known as a *partial response maximum-likelihood* (PRML) technique [3, 43], which is practically utilized in magnetic recording systems. This is done in two steps. First, the received signal is equalized to a PR target whose response is as close to the actual channel response as possible. Then, the Viterbi detector performs the ML equalization on the modified readback sequence based on PR trellis.

2.5 Summary

In this chapter, we describe the backgrounds of digital magnetic recording systems and basic communication channel model for magnetic storage, as well as principle process of writing and reading process in the hard disk drives. The transition responses, dibit responses, and frequency responses for both of longitudinal and perpendicular magnetic recording are shown. In addition, we also explain the dominant noises which occur in a magnetic recording system. Finally, the realistic channel model and the ideal channel model are described. The two models will be used for the system simulation in the next chapter.

CHAPTER 3

TARGET-SHAPING EQUALIZER DESIGN

In this chapter, a method of designing an infinite impulse response (IIR) equalizer for turbo equalization in the perpendicular magnetic recording channels will be designed and described. Based on a minimum mean-squared error (MMSE) approach, the coefficients of the IIR equalizer are explicitly derived. Then, we compare its performance with a finite-impulse response (FIR) equalizer in the presence and absence of the media jitter noise. Results indicate that for a small number of equalizer taps, the proposed IIR equalizer outperforms the FIR equalizer at moderate to high jitter noise levels. Also if compared with an 11-tap FIR equalizer, the proposed IIR counterpart can achieve a similar BER performance, but it requires a fewer number of equalizer taps.

3.1 Introduction

The PRML technique [43] is widely used for data detection process in the perpendicular magnetic recording channels. In practice, in this technique, an FIR equalizer is employed to shape the readback signal to a predetermined target before performing an ML equalization by the Viterbi detector [2].

The FIR equalizer with a large number of taps is required to function properly at high recording densities. Nevertheless, the total number of equalizer taps is practically limited by the maximum allowable loop delay in the timing recovery loop because a small loop delay provides a more robust phase locking [3], which in turn improves the overall system performance. Furthermore, the benefits of the equalizer with fewer taps are three folds (i) A smaller area on the silicon chip. (ii) A shorter optimization time of read-channel chip during the production. (iii) A small delay in the timing loop.

An IIR equalizer has previously been studied in the literature [4], and references therein. For instance, the IIR modeling was considered in the decision feedback equalizer design [5] to reduce the number of filter taps. Also, in [6], the performance of employing the continuous-time adaptive IIR equalizers for EPR4 channels was investigated.

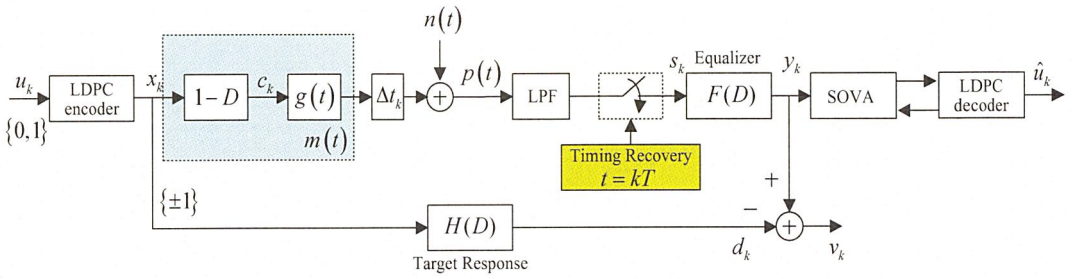


Figure 3.1 System model for turbo equalization and target design.

An algorithm for the approximation of FIR filter by the IIR filter and the direct method for converting the FIR filter with low non-zero taps into the IIR filter using the predetermined table are proposed in [7 – 8], respectively. However, in this thesis, we directly design the digital IIR equalizer based on the MMSE approach, and then compare its performance with the conventional FIR equalizer in a full turbo equalization setting.

The rest of this chapter is organized as follows. After explaining our system model in Section 3.2, we briefly explain the design of the conventional FIR equalizer, and also describe the design of the IIR equalizers for partial response channels in Section 3.3. Simulation results are provided in Section 3.4. Finally, summary is given in Section 3.5.

3.2 System Model

Figure 3.1 illustrates the channel model of a perpendicular magnetic recording system with jitter noise. A message input sequence $u_k \in \{0,1\}$ is encoded by a low-density parity-check (LDPC) code [44] and is mapped to a binary input sequence $x_k \in \{\pm 1\}$ with a bit period T, Then, the sequence x_k is filtered by an ideal differentiator $1\text{-}D$ to form a transition sequence $c_k = (x_k - x_{k-1})$ where $c_k = \pm 2$ corresponds to a positive or negative transition and $c_k = 0$ corresponds to the absence of a transition. The sequence c_k is then passed through the perpendicular recording channel, represented by the transition response $g(t)$ in equation (2.2).

The readback signal $p(t)$ in Figure 3.1 can be expressed as

$$p(t) = \sum_k c_k g(t - kT + \Delta t_k) + n(t), \tag{3.1}$$

where $n(t)$ is an additive white Gaussian noise (AWGN) with the two-sided power spectral density of $N_0/2$ (W/Hz). The jitter noise Δt_k is modeled as a truncated Gaussian probability distribution

function with $\mathcal{N}\left(0, |c_k|\sigma_j^2\right)$, where σ_j specified as a percentage of T determines the severity of the jitter noise.

The sampled output s_k is equalized by a digital equalizer of the form

$$F_{FIR}(D) = \sum_{k=-K}^{K} f_k D^k , \tag{3.2}$$

where f_k is the k-th equalizer coefficients, and $2K+1$ is the total number of equalizer taps, such that the output sequence y_k resembles the desired target output sequence d_k. Finally, the sequence y_k is fed to a turbo equalizer, which iteratively exchanges the soft information between the soft output SOVA equalizer [21] and the LDPC decoder, respectively.

3.3 Target and Equalizer Design

3.3.1 MMSE FIR Equalizer

A typical MMSE FIR filter design is based on Figure 3.1. Consider a partial-response target of the form

$$H(D) = \sum_{k=0}^{L-1} h_k D^k , \tag{3.3}$$

where L is the target length, and h_k is the k-th target coefficients. From Figure 3.1, the equalizer output and the target output are expressed as

$$y_k = s_k * f_k , \tag{3.4}$$

and

$$d_k = x_k * h_k , \tag{3.5}$$

respectively, where * denotes the convolution operator. Equation (3.4) and (3.5) can also be written in the matrix form as

$$y_k = \mathbf{f}^{\mathrm{T}}\mathbf{s}, \tag{3.6}$$

and

$$d_k = \mathbf{h}^{\mathrm{T}}\mathbf{x}, \tag{3.7}$$

where $\mathbf{f} = [f_{-K}, f_{-K+1}, ..., f_{K-1}, f_K]^{\mathrm{T}}$, $\mathbf{s} = [s_{k+K}, s_{k+K-1}, ..., s_{k-K}]^{\mathrm{T}}$, $\mathbf{h} = [h_0, h_1, ..., h_{L-1}]^{\mathrm{T}}$, $\mathbf{x} = [x_k, x_{k-1}, ..., x_{k-L+1}]^{\mathrm{T}}$, and $[\bullet]^{\mathrm{T}}$ is the transpose operation. Note that in this thesis the bold characters denoted as the matrix, and italic characters denoted as the parameter. The target and its corresponding FIR equalizer can then be simultaneously obtained based on a minimum mean-square error (MMSE) approach. The mean-square error (MSE) between the equalizer output y_k and the target output d_k is given by [42]

$$\begin{aligned} E\left[v_k^2\right] &= E\left[\left(y_k - d_k\right)^2\right] \\ &= \mathbf{f}^{\mathrm{T}}\mathbf{R}\mathbf{f} + \mathbf{h}^{\mathrm{T}}\mathbf{A}\mathbf{h} - 2\mathbf{f}^{\mathrm{T}}\mathbf{P}\mathbf{h}, \end{aligned} \tag{3.8}$$

where $E[\bullet]$ is the expectation operator, $\mathbf{R} = E\left[\mathbf{s}\mathbf{s}^{\mathrm{T}}\right]$ is an N-by-N auto-correlation matrix of the sampled channel output s_k, $\mathbf{A} = E\left[\mathbf{x}\mathbf{x}^{\mathrm{T}}\right]$ is an L-by-L auto-correlation matrix of the input sequence x_k, and $\mathbf{P} = E\left[\mathbf{s}\mathbf{x}^{\mathrm{T}}\right]$ is an N-by-L cross-correlation matrix between s_k and x_k.

Given a specified constraint, the MSE in (3.8) can be minimized by differentiating (3.8) with respect to $\mathbf{h}$ and $\mathbf{f}$, and setting the results to zero. For a fixed partial-response target $\mathbf{h}$, the equalizer coefficients can be computed from

$$\mathbf{f} = \mathbf{R}^{-1}\mathbf{P}\mathbf{h}, \tag{3.9}$$

Alternative method to obtain the FIR filter coefficients is based on the neural network [45-46]. The extended Kalman filter (EKF) algorithms based on the real-time recurrent learning (RTRL) for the decision feedback recurrent neural equalizer (DFRNE) was proposed in [45] to solve drawback of slow convergence rate. Then, a new design method of the simplified neural network equalizer (NNE) with the noise whitening function was proposed in [46] for a generalized partial response (GPR) channel is proposed.

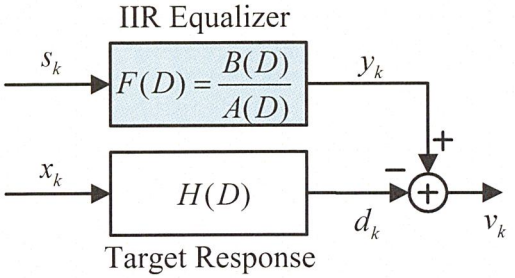


Figure 3.2 Block diagram for designing an IIR equalizer.

3.3.2 MMSE IIR Equalizer

The method for designing the proposed IIR equalizer can be described by a block diagram as shown in Figure 3.2. An IIR equalizer $F(D)$ is expressed as [4]

$$F(D)=\frac{B(D)}{A(D)}=\frac{\sum_{k=-N}^{N} b_k D^k}{\sum_{k=0}^{M} a_k D^k}, \tag{3.10}$$

where a_k and b_k are the k-th coefficients of the denominator and the numerator of $F(D)$ respectively, and both N and M are integers. In our study, we consider the case where $M+1<2N+1$. From Figure 3.2, we see that

$$y_k = s_k * f_k, \tag{3.11}$$

which can be rewritten as

$$s_k * b_k = y_k * a_k. \tag{3.12}$$

By substituting $y_k = d_k + v_k$ into (3.12), we obtain

$$\begin{aligned} s_k * b_k &= (d_k + v_k) * a_k, \\ v_k * a_k &= s_k * b_k - d_k * a_k. \end{aligned} \tag{3.13}$$

Assuming that $a_0 = 1$, (3.13) can be written as

$$v_k + \sum_{i=1}^{M} v_{k-i} a_i = s_k * b_k - d_k * a_k, \tag{3.14}$$

and

$$\begin{aligned} v_k &= s_k * b_k - \left(d_k + \sum_{i=1}^{M} d_{k-i} a_i \right) - \sum_{i=1}^{M} v_{k-i} a_i \\ &= s_k * b_k - \tilde{d}_k * \tilde{a}_k - \tilde{v}_k * \tilde{a}_k - d_k. \end{aligned} \tag{3.15}$$

For convenience, (3.15) can be written in the matrix form as

$$v_k = \mathbf{s}^{\mathrm{T}}\mathbf{b} - \tilde{\mathbf{d}}^{\mathrm{T}}\tilde{\mathbf{a}} - \tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{a}} - d_k, \tag{3.16}$$

where $\mathbf{s} = \left[s_{k+N}, ..., s_k, ..., s_{k-N} \right]^{\mathrm{T}}$, $\mathbf{b} = \left[b_{-N}, ..., b_0, ..., b_N \right]^{\mathrm{T}}$, $\tilde{\mathbf{d}} = \left[d_{k-1}, d_{k-2}, ..., d_{k-M} \right]^{\mathrm{T}}$, $\tilde{\mathbf{a}} = \left[a_1, ..., a_M \right]^{\mathrm{T}}$, and $\tilde{\mathbf{v}} = \left[v_{k-1}, v_{k-2}, ..., v_{k-M} \right]^{\mathrm{T}}$ are $(2N+1)$, $(2N+1)$, M, M, and $M-$ element column vector, respectively. Then, the MSE of (3.16) is given by

$$\begin{aligned} E\left[v_k^2 \right] &= E\left[\left(\mathbf{b}^{\mathrm{T}}\mathbf{s} - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{d}} - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{v}} - d_k \right)\left(\mathbf{s}^{\mathrm{T}}\mathbf{b} - \tilde{\mathbf{d}}^{\mathrm{T}}\tilde{\mathbf{a}} - \tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{a}} - d_k \right) \right] \\ &= E\Big[\mathbf{b}^{\mathrm{T}}\mathbf{s}\mathbf{s}^{\mathrm{T}}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\mathbf{s}\tilde{\mathbf{d}}^{\mathrm{T}}\tilde{\mathbf{a}} - \mathbf{b}^{\mathrm{T}}\mathbf{s}\tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{a}} - \mathbf{b}^{\mathrm{T}}\mathbf{s}d_k \\ &\quad - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{d}}\mathbf{s}^{\mathrm{T}}\mathbf{b} - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{d}}\tilde{\mathbf{d}}^{\mathrm{T}}\tilde{\mathbf{a}} - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{d}}\tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{a}} - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{d}}d_k \\ &\quad - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{v}}\mathbf{s}^{\mathrm{T}}\mathbf{b} - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{v}}\tilde{\mathbf{d}}^{\mathrm{T}}\tilde{\mathbf{a}} - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{v}}\tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{a}} - \tilde{\mathbf{a}}^{\mathrm{T}}\tilde{\mathbf{v}}d_k \\ &\quad - d_k\mathbf{s}^{\mathrm{T}}\mathbf{b} - d_k\tilde{\mathbf{d}}^{\mathrm{T}}\tilde{\mathbf{a}} - d_k\tilde{\mathbf{v}}^{\mathrm{T}}\tilde{\mathbf{a}} - d_k d_k \Big], \end{aligned} \tag{3.17}$$

given $\mathbf{c} = E\left[\mathbf{s}d_k \right]$, $\mathbf{d} = E\left[\tilde{\mathbf{d}}d_k \right]$, $\mathbf{e} = E\left[\tilde{\mathbf{v}}d_k \right]$ are $(2N+1)$, M, and $M-$element column vectors, respectively. We can rearrange (3.17) as

$$\begin{aligned} E\left[v_k^2 \right] &= \mathbf{b}^{\mathrm{T}}\mathbf{S}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\mathbf{X}\tilde{\mathbf{a}} - \mathbf{b}^{\mathrm{T}}\mathbf{P}\tilde{\mathbf{a}} - \mathbf{b}^{\mathrm{T}}\mathbf{c} - \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{b} + \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{D}\tilde{\mathbf{a}} \\ &\quad + \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{W}\tilde{\mathbf{a}} + \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{d} - \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{b} + \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\tilde{\mathbf{a}} + \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}\tilde{\mathbf{a}} \\ &\quad + \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{e} - \mathbf{c}^{\mathrm{T}}\mathbf{b} + \mathbf{d}^{\mathrm{T}}\tilde{\mathbf{a}} + \mathbf{e}^{\mathrm{T}}\tilde{\mathbf{a}} + d_k^2, \end{aligned} \tag{3.18}$$

where $\mathbf{S} = E\left[\mathbf{s}\mathbf{s}^{\mathrm{T}}\right]$, $\mathbf{X} = E\left[\mathbf{s}\tilde{\mathbf{d}}^{\mathrm{T}}\right]$, $\mathbf{P} = E\left[\mathbf{s}\tilde{\mathbf{v}}^{\mathrm{T}}\right]$, $\mathbf{D} = E\left[\tilde{\mathbf{d}}\tilde{\mathbf{d}}^{\mathrm{T}}\right]$, $\mathbf{W} = E\left[\tilde{\mathbf{d}}\tilde{\mathbf{v}}^{\mathrm{T}}\right]$, and $\mathbf{V} = E\left[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^{\mathrm{T}}\right]$ are $(2N+1)$-by-$(2N+1)$, $(2N+1)$-by-M, $(2N+1)$-by-M, M-by-M, M-by-M, and M-by-M matrices, respectively.

To find the coefficients of an IIR equalizer, we differentiate (3.18) with respect to $\tilde{\mathbf{a}}$ and $\mathbf{b}$, respectively, and set the results to zero, to obtain

$$\frac{\partial E\left[v_k^2\right]}{\partial \tilde{\mathbf{a}}} = 2\mathbf{S}\mathbf{b} - 2\mathbf{X}\tilde{\mathbf{a}} - 2\mathbf{P}\tilde{\mathbf{a}} - 2\mathbf{c} = 0, \tag{3.19}$$

and

$$\frac{\partial E\left[v_k^2\right]}{\partial \mathbf{b}} = 2\mathbf{X}^{\mathrm{T}}\mathbf{b} - 2\mathbf{P}^{\mathrm{T}}\mathbf{b} + 2\mathbf{D}\tilde{\mathbf{a}} + 2\mathbf{W}\tilde{\mathbf{a}} + 2\mathbf{d} + 2\mathbf{W}^{\mathrm{T}}\tilde{\mathbf{a}} + 2\mathbf{V}\tilde{\mathbf{a}} + 2\mathbf{e} = 0. \tag{3.20}$$

An equivalent matrix suited for calculation is obtained by rearranging (3.19) and (3.20) as

$$\underbrace{\begin{bmatrix} \mathbf{S} & -[\mathbf{X}+\mathbf{P}] \\ -\mathbf{X}^{\mathrm{T}} & \mathbf{D}+\mathbf{W}+\mathbf{W}^{\mathrm{T}}+\mathbf{V} \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} \mathbf{b} \\ \tilde{\mathbf{a}} \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} \mathbf{c} \\ -[\mathbf{d}+\mathbf{e}] \end{bmatrix}}_{\mathbf{u}}. \tag{3.21}$$

Because the matrix $\mathbf{Q}$ in (3.21) is a square matrix, the coefficients of $F(D)$ in a vector $\mathbf{z}$ can be easily computed from

$$\mathbf{z} = \mathbf{Q}^{-1}\mathbf{u}. \tag{3.22}$$

3.4 Simulation Results and Discussions

We consider the EEPR2 target [47] $H(D) = 1 + 4D + 6D^2 + 4D^3 + D^4$ for perpendicular magnetic recording systems. The $(2K+1)$-tap FIR equalizer is designed based on the MMSE approach, which also yields an error sequence v_k that will be used to design the IIR equalizer. The signal-to-noise ratio (SNR) is defined as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{E_i}{N_0}\right) \ (\mathrm{dB}), \tag{3.23}$$

where E_i is the energy of the channel impulse response. All equalizers are designed at the SNR required to achieve bit-error rate (BER) = 10^{-5}. Each BER point is computed using as many 4096-bits data sectors as needed to collect 500 error bits, whereas the equalizer taps are designed using only one data sector.

3.4.1 Uncoded System

First, we consider an uncoded system, where the LDPC encoder and decoder are not used. Thus, we investigate the performance of different equalizers, namely, the FIR equalizers with 3, 5, and 11 taps and the IIR equalizers with 2 and 4 zeros (one pole each) at the output of the SOVA detector.

Figure 3.3 plots the SNR required to achieve BER = 10^{-4} as a function of NDs in the absence of the jitter noise ($\sigma_j^2 = 0\%$), where the term "$vZmP$ IIR" refers to the IIR equalizer with $v = 2N$ zeros (equivalent to $v+1$ taps) and $m = M$ poles. It should be noted that a $(2K+1)$ tap FIR filter causes the same amount of delays as a $2KZmP$ IIR filter with $2K$ zeros. As depicted in Figure 3.3, when the number of equalizer taps is small (e.g., 3 taps) and ND is high, the IIR equalizers with the same delays perform better than the FIR equalizer.

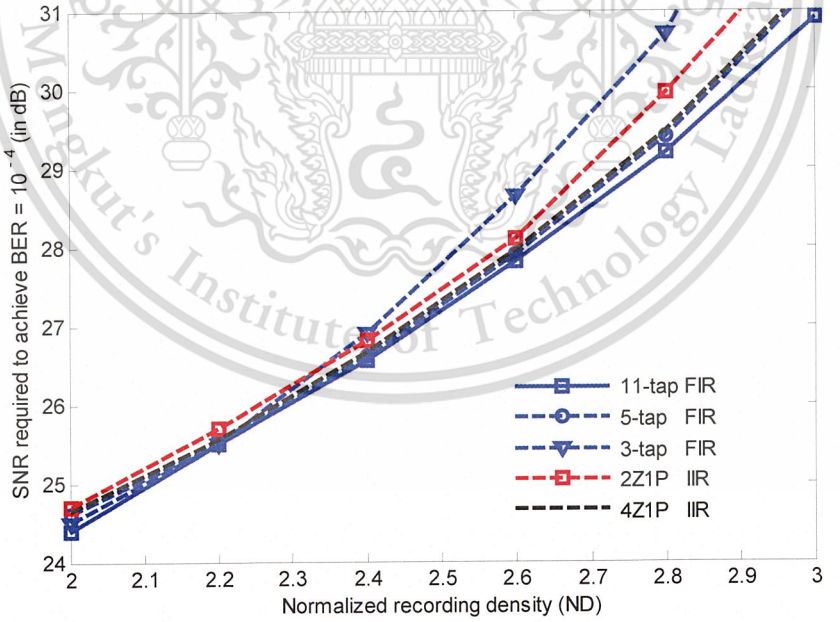


Figure 3.3 Performance comparisons between the FIR and the IIR equalizers at different NDs.

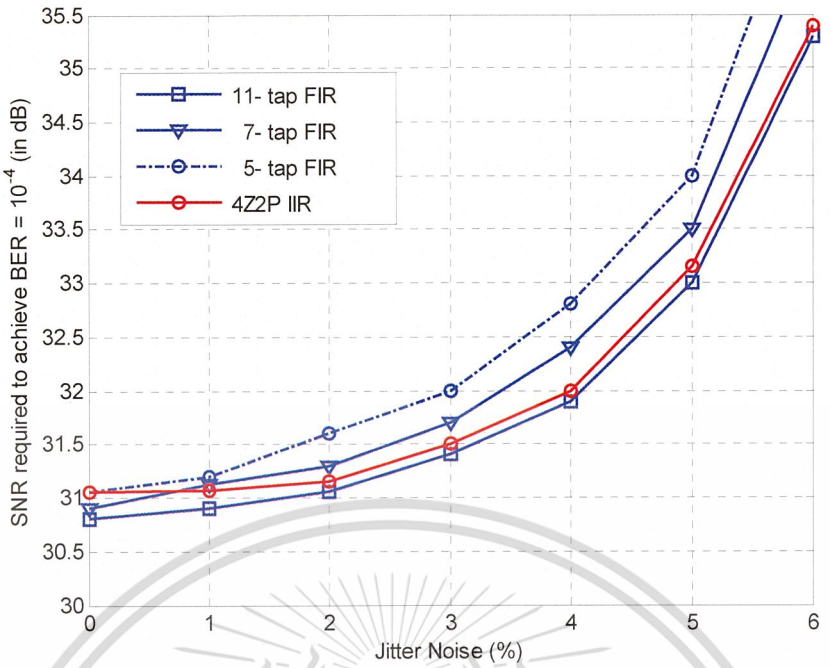


Figure 3.4 Performance comparisons at different jitter noise levels.

In Figure 3.4, we pick ND = 3 and this time compare the performance of different equalizers as a function of the jitter noise amounts from 0% to 6%. It is evident that the $4Z2P$ IIR equalizer requires lower SNR to achieve BER = 10^{-4} than both the 5-tap and the 7-tap FIR equalizers. In addition, the $4Z2P$ IIR filter performs close to the 11-tap FIR equalizer at all jitter noise levels. From the viewpoint of delays, it can be concluded that the $4Z2P$ IIR equalizer is more advantageous than the 11-tap FIR equalizer because it introduces only two delays in the system rather than five delays introduced by the 11-tap FIR equalizer.

3.4.2 Coded System

We also investigate the performance of the IIR equalizers in turbo equalization setting for 4 iterations with and without jitter noise as shown in Figure 3.5. The LDPC code is from a modified array code (MAC) with parameter $(p, j, k) = (107, 4, 38)$ [48]. We consider a rate $(1 - j/k) = 0.8947$ coded system in which a block of 3,638 message bits $\{u_k\}$ is encoded by an LDPC encoder, resulting in a coded block length of 4,066 bits, and is mapped to an input data sequence $\{x_k\}$ as shown in Figure 3.1. Similarly, the detected bits are also decoded by the LDPC decoder to obtain an estimated message sequence $\{\hat{u}_k\}$.

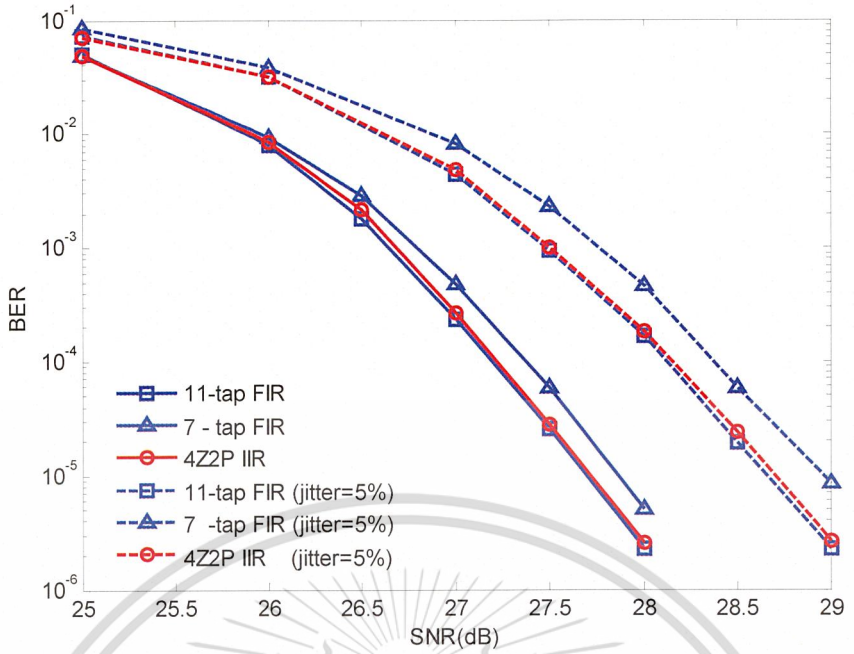


Figure 3.5 Performance comparisons in the turbo equalization setting with 4 iterations.

To account for the code rate, the user density (D_u) used instead of ND in this simulation, is defined as

$$D_u = \mathrm{ND} \times R\,, \tag{3.24}$$

where $R = 0.8947$ is a code rate for this simulation setup. We also pick ND = 3.0 such that $D_u = 2.6841$. In Figure 3.5, it is apparent that the $4Z2P$ IIR filter achieves a similar BER level to the 11-tap FIR filter. In addition, at BER = 10^{-5}, the proposed IIR filter yields the performance gain of 0.2 dB (without jitter noise) and 0.4 dB (with jitter noise 6%) over the 7-tap FIR filter.

The reason that the IIR equalizer provides a better performance than the FIR equalizer because it can shape the readback signal to the PR target better than the FIR equalizer does, especially when the number of equalizer taps is small. This can be explained by plotting the frequency responses of different equalizers in Figure 3.6 for the perpendicular magnetic recording channel at ND = 3.0. If we assume that the 11-tap FIR equalizer is the best, the $4Z2P$ IIR equalizer whose frequency response closely matches the frequency response of the 11-tap FIR equalizer than that of the 7-tap FIR equalizer.

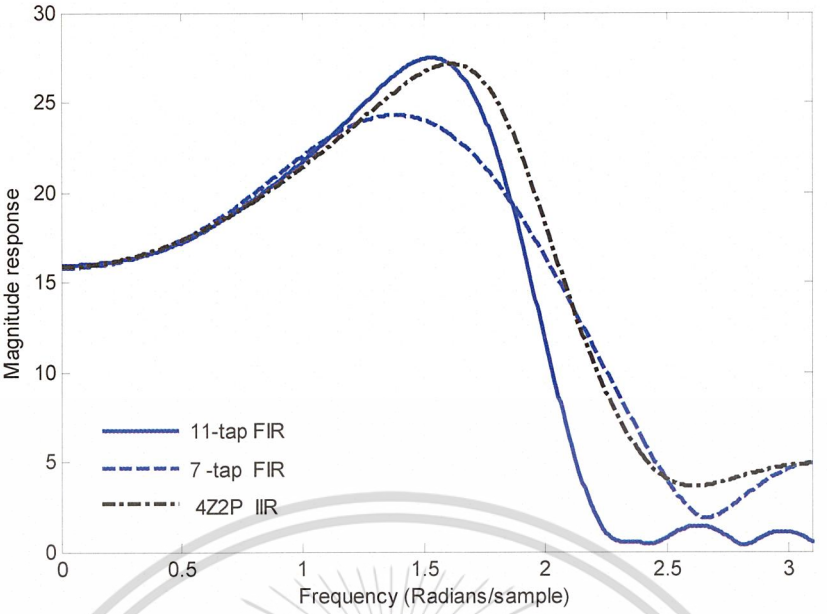


Figure 3.6 Frequency responses of the readback signals at ND = 3.

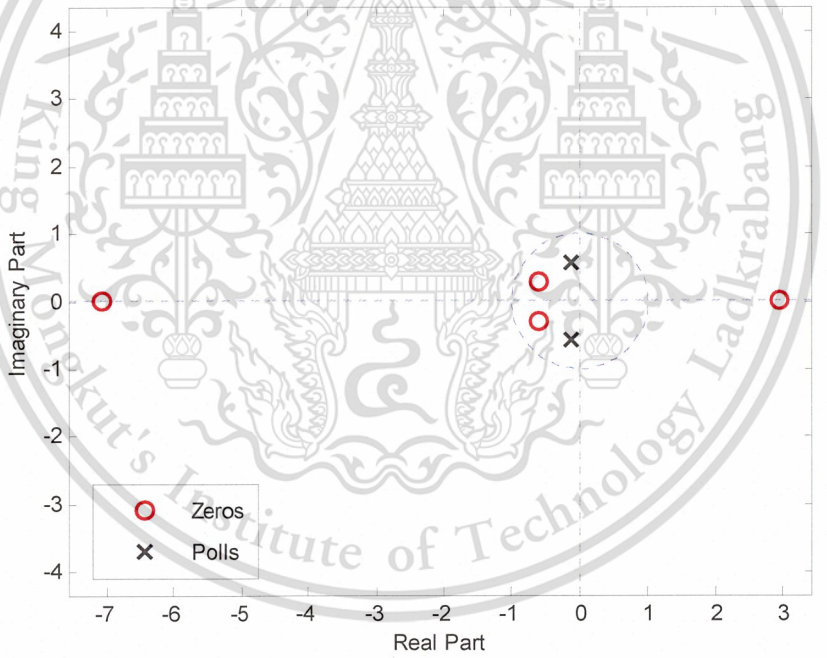
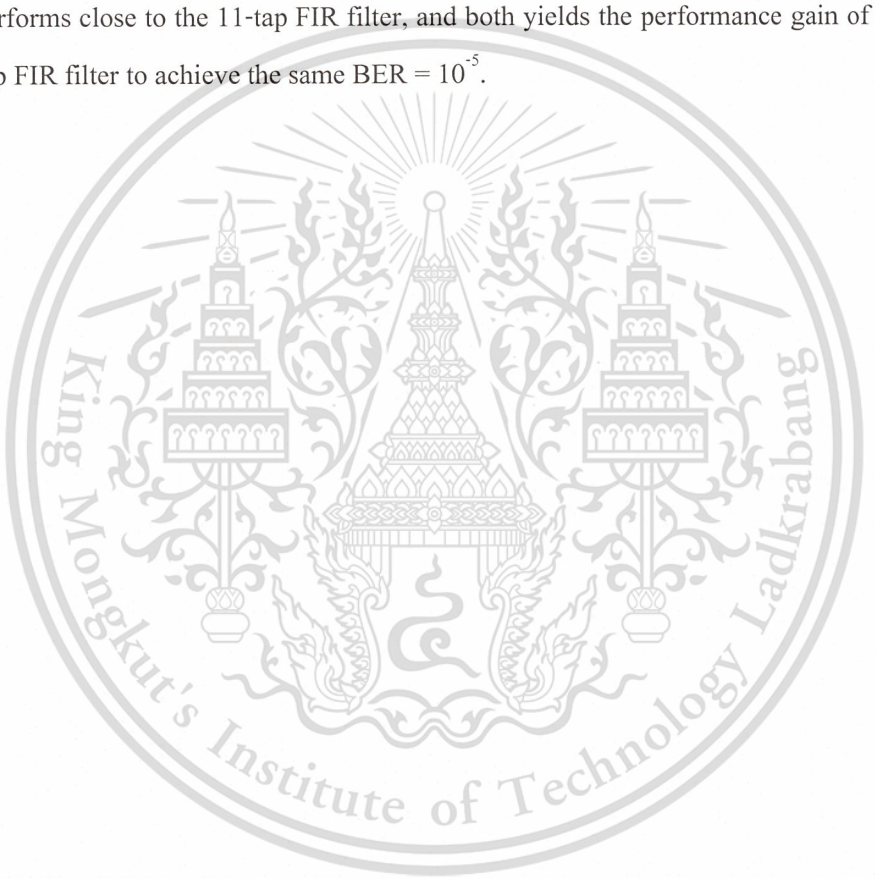


Figure 3.7 Poles-zeros diagram of the proposed $4Z2P$ IIR equalizer.

Generally, the IIR filter comes with possible concern about stability; however, based on our extensive simulations, we have been able to conclude that the proposed IIR equalizer is highly stable for PR channels. This can be explained by plotting the poles-zeros diagram of the proposed $4Z2P$ IIR equalizer in Figure 3.7 for the perpendicular magnetic recording channel at ND = 3.0. We can see that theses poles lies inside the unit circle IIR equalizer.

3.5 Summary

In this chapter, we describe how to design an IIR equalizer for perpendicular magnetic recording channels based on the MMSE approach. Based on our extensive simulations, we found that the proposed IIR equalizer is highly stable for PR channels. Simulation results show that for small number of equalizer taps, the proposed IIR equalizers with suitable poles and zeroes can outperform the FIR equalizers and also perform close to the 11-tap FIR filter for all jitter noise levels. This is because the IIR equalizer can shape the readback signal to the PR target better than the FIR equalizer. Furthermore, in the turbo equalization setting, we found that the $4Z2P$ IIR filter performs close to the 11-tap FIR filter, and both yields the performance gain of 0.4 dB over the 7-tap FIR filter to achieve the same BER = 10^{-5}.



CHAPTER 4

BI-DIRECTIONAL TIMING RECOVERY

This chapter discusses the concept of the conventional timing recovery and proposes bi-directional timing recovery to increase the efficiency of conventional timing recovery system. Both systems work on the basis of the phase locked loop (PLL). We will explain how to design the parameters of the phase locked loop using a simulation of a phase locked loop of a linearized PLL model. Then, we compare the performance of both methods.

4.1 Introduction

Timing recovery is the process of synchronizing the sampler with the received analog signal. Sampling at the wrong times can have a devastating impact on overall performance. Therefore, the quality of synchronization is very important.

Most practical timing recovery schemes are based on a PLL [3], which consists of a timing error detector (TED), a loop filter, and a voltage-controlled oscillator (VCO). Typically, there are two configurations of timing recovery, namely, deductive and inductive timing recovery schemes [3], depending on whether the timing information is extracted before or after the sampler.

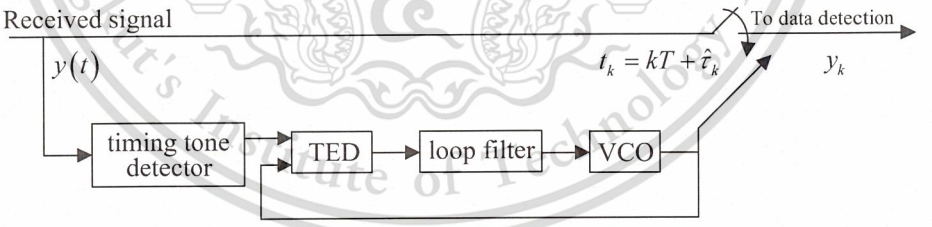


Figure 4.1 Deductive (or feed-forward) timing recovery.

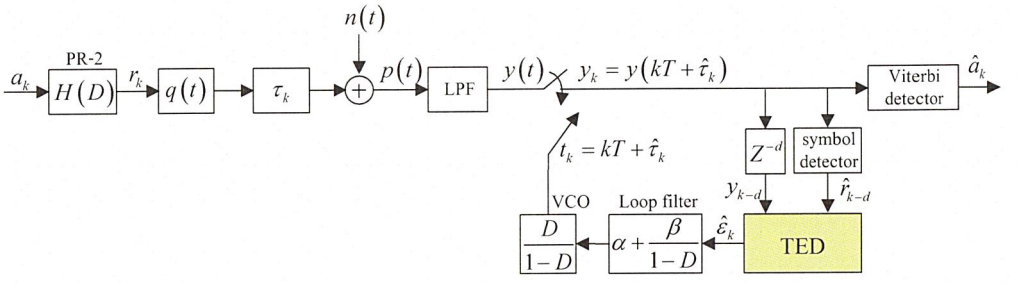


Figure 4.2 The perfectly equalized channel model with inductive (feedback) timing recovery.

Deductive timing recovery directly extracts the timing tone [39] from the incoming signal before the sampler, as shown in Figure 4.1, where a PLL is used to reduce the effect of timing jitter [39]. Inductive timing recovery, on the other hands, employs a feedback loop using a PLL to extract the timing information, as depicted in Figure 4.2. The key advantage of inductive timing recovery is that it can be implemented digitally. In this work, only inductive timing recovery is considered, which will be referred to as conventional timing recovery. A reader interested in deductive (or feed-forward) timing recovery can refer to [9] for a brief discussion.

4.2 Conventional Timing Recovery

Consider the perfectly equalized channel model (also referred to as an ideal channel model) shown in Figure 4.2. An input data sequence $a_k \in \{\pm 1\}$ with bit period T is filtered by the channel represented by $H(D) = \sum_{k=0}^{\nu} h_k D^k$, where h_k is the k-th channel coefficient, D is the delay operator, and ν is channel memory. The readback signal, $p(t)$, can then be written as

$$p(t) = \sum_k r_k q(t - kT - \tau_k) + n(t), \tag{4.1}$$

where $r_k = \sum_i a_{k-i} h_i$ is the noiseless channel output, $q(t) = \sin(\pi t / T) / (\pi t / T)$ is an ideal zero-excess-bandwidth Nyquist pulse [8], τ_k is the k-th unknown timing offset, and $n(t)$ is additive white Gaussian noise (AWGN) with two-sided power spectral density $N_0 / 2$ W/Hz. Unless otherwise specified, the timing offset τ_k used throughout this chapter is modeled as a random walk [49] according to

$$\tau_{k+1} = \tau_k + w_k, \tag{4.2}$$

where w_k is an independent and identically distributed (*i.i.d.*) zero-mean Gaussian random variable with variance σ_w^2, i.e., $w_k \sim \mathcal{N}(0, \sigma_w^2)$, and σ_w determines the severity of the timing jitter. The random walk model is chosen because of its simplicity and its ability to represent a variety of channels by changing only one parameter.

At the front-end receiver, the readback signal is filtered by a low-pass filter (LPF), whose impulse response is $q(t)/T$ (i.e., a cutoff frequency is at $1/2T$), to eliminate the out-of-band noise and is then sampled at time $kT + \hat{\tau}_k$ creating

$$y_k = y(kT + \hat{\tau}_k) = \sum_i r_i q(kT + \hat{\tau}_k - iT - \tau_i) + n_k \,, \tag{4.3}$$

where $\hat{\tau}_k$ is an estimate of τ_k (or the k-th sampling phase offset), and n_k is an *i.i.d.* zero-mean Gaussian random variable with variance $\sigma_n^2 = N_0 / (2T)$, i.e., $n_k \sim \mathcal{N}(0, \sigma_n^2)$.

A decision-directed timing error detector (TED) [3] is used to compute the receiver's estimate of the timing error $\varepsilon_k = \tau_k - \hat{\tau}_k$, which is the misalignment between the phase of the received signal and that of the sampling clock. Several TED algorithms have been proposed in the literature [3, 50], depending on how they incorporate the information available at the TED input. Typically, the overall performance of timing recovery is dominated by the effectiveness of the TED. In this thesis, we consider the well-known Mueller and Müller (M&M) TED algorithm [11], where the estimated timing error is given by

$$\hat{\varepsilon}_k = K_T \{ y_k \hat{r}_{k-1} - y_{k-1} \hat{r}_k \} \,, \tag{4.4}$$

where $\hat{r}_k$ is an estimate of r_k. The constant K_T is used to ensure that there is no bias at high SNR so that $E[\hat{\varepsilon}_k | \varepsilon] = \varepsilon$ (or, in other words, the slope of the S-curve [3] is unity at the origin). As shown in (4.4), the TED performance depends on the decisions $\{\hat{r}_k\}$. Therefore, timing recovery performance is a strong function of the reliability of decisions and hence of the operating SNR. This explains why the symbol detector used in the timing loop is a Viterbi detector with a short decision delay, dT, instead of a memoryless multi-level slicer in most real-life applications [1]. Next, the estimated timing error $\hat{\varepsilon}_k$ is filtered by a loop filter to eliminate the noise in the timing error signal. Then, the next sampling phase offset is updated by a second-order PLL according to [3]

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \beta \hat{\varepsilon}_k \,, \tag{4.5}$$

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha \hat{\varepsilon}_k + \hat{\theta}_{k+1} \,, \tag{4.6}$$

where $\hat{\theta}_k$ represents an estimate of frequency error [1], and α and β are the PLL gain parameters [3], which determine the loop bandwidth and the rate of convergence. The larger the value of PLL gains parameters, the larger the loop bandwidth, the faster the convergence rate, and thus the more the noise allowed perturbing the system. Note that a first-order PLL can only

handle phase error, not frequency error. A first-order PLL update equation is easily obtained by setting $\beta = 0$ in (4.5). It can be rewritten as

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha\hat{\varepsilon}_k , \tag{4.7}$$

In practice, timing recovery is performed in two modes, namely, acquisition and tracking modes. An acquisition mode is performed at the beginning of the data sector with the aid of a known data pattern called a preamble [1] (or a training sequence) to acquire the initial phase and frequency estimates. Hence, a tracking mode is performed using the samples corresponding to transmitted unknown data so as to refine these initial estimates. Since the preamble is known at the receiver, large values of α and β can be used to expedite the convergence rate. However, the values of α and β should be lowered during tracking mode so as to reduce the effect of the noise [51]. Therefore, designers must tradeoff between the loop bandwidth and the convergence rate when designing α and β.

4.3 Design of PLL Gain Parameters

From the simulation point of view, the best way to choose the PLL gain parameters (both α and β) is to optimize them based on minimizing the BER at the detector output. Nevertheless, this method is impractical and time-consuming. Instead, one usually designs α and β based on a linearized model of PLL [3]. One possible criterion is to choose α and β so that the system response can catch a phase and/or frequency change in the system within "*C*" samples (or bit periods). It should be noted that this criterion can also be viewed as the rate of convergence, i.e., the smaller the *C*, the faster the convergence rate.

4.3.1 Linear Analysis of First-Order PLL

A first-order PLL is of restricted practical interest because it can only handle the phase error, not the frequency error. Nonetheless, its analysis is a good start for understanding a higher-order PLL. In this analysis, the phase error is modeled as a step function according to $\tau_k = T$ for $k \geq 0$, and $\tau_k = 0$ for $k < 0$.

Consider a first-order PLL update equation, which is given by

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha\hat{\varepsilon}_{k-d}, \tag{4.8}$$

where d is a normalized delay (with respect to T) in the timing loop, $\hat{\varepsilon}_k = \varepsilon_k + v_k$ is the estimated timing error, $\varepsilon_k = \tau_k - \hat{\tau}_k$ is the residual timing error, and v_k is the noise in the TED. By assuming that v_k is negligible, the system transfer function of (4.8) can be obtained by taking the Z-transform [39], i.e.,

$$G(z) = \frac{\hat{\Gamma}(z)}{\Gamma(z)} = \frac{\alpha z^{-(d+1)}}{1 - z^{-1} + \alpha z^{-(d+1)}}, \tag{4.9}$$

where $\hat{\Gamma}(z)$ and $\Gamma(z)$ are the Z-transform of $\hat{\tau}_k$ and τ_k, respectively. Accordingly, the error transfer function (Z-transform of ε_k) can be written as

$$E(z) = \Gamma(z) - \hat{\Gamma}(z) = \frac{1 - z^{-1}}{1 - z^{-1} + \alpha z^{-(d+1)}}\Gamma(z). \tag{4.10}$$

One possible criterion that may be used to choose α is to pick the one that satisfies both the system stability and the convergence rate, for a given d. This can be achieved by using either (4.9) or (4.10). It is done by first finding α's that satisfy the system stability. As seen in (4.9) and (4.10), the PLL is stable whenever all poles (i.e., roots of the denominator) of (4.9) and (4.10) lie inside the unit circle [39]. It can be shown that the value of α resulting in the system stability can be expressed as [3]

$$0 < \alpha < 2\sin\left(\frac{\pi}{4d+2}\right). \tag{4.11}$$

Figure 4.3 shows the maximum value of α that retains the system stability for different delays. Apparently, the stability range of α's decreases dramatically as d increases. Among a set of α's that satisfies the system stability, we select one α so that the system response can catch the step response within C samples with ±5% tolerance.

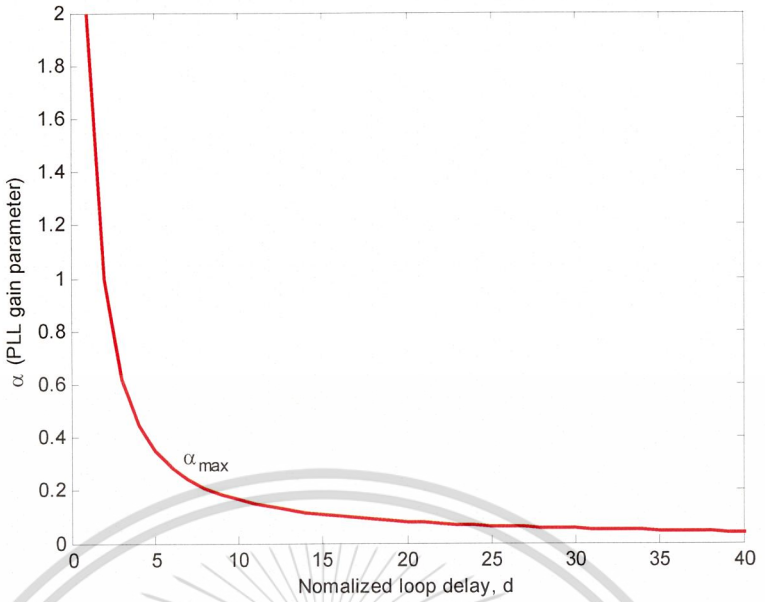


Figure 4.3 Maximum value of α satisfying the system stability for different loop delay.

The 5% tolerance is introduced to relax our criterion so as to reduce the effect of the noise in the timing loop. Figure 4.4 shows α_C's that satisfy both the system stability and the convergence rate of C samples. Clearly, the faster the convergence rate, the larger the value of α. As depicted in Figure 4.4(a), for a given C, there are only some α's that satisfy the system stability up to a certain loop delay. For example, there exists α_{100} up to a loop delay of $30T$.

Figure 4.4(b) shows the system step responses using (4.9) and α_{100} for $d = 0$ to 30, which coincides with our criterion. Since we design α_{100} so that the system can catch the step response within 100 samples with ±5% tolerance, this implies that the absolute value of the magnitude of the error response $E(z)$ given in (4.10) should also be less than or equal to 0.05 after 100 samples, as shown in Figure 4.5 for $d = 14$. Observe that there are two α_{100}'s (i.e., $\alpha = 0.0218$ and $\alpha = 0.055$) that satisfy the convergence rate of 100 samples with ±5% tolerance. Nonetheless, it is desirable in practice to employ a small α in the timing loop to minimize the loop bandwidth, thus reducing the effect of the noise in the timing loop.

4.3.2 Linear Analysis of Second-Order PLL

When there is a frequency offset component in a system, a second-order PLL must be employed. To design the second-order PLL gain parameters (both α and β), we first design α, for given d and C, by assuming that there is only phase error in the system. This is achieved by the method described in Section 4.3.1.

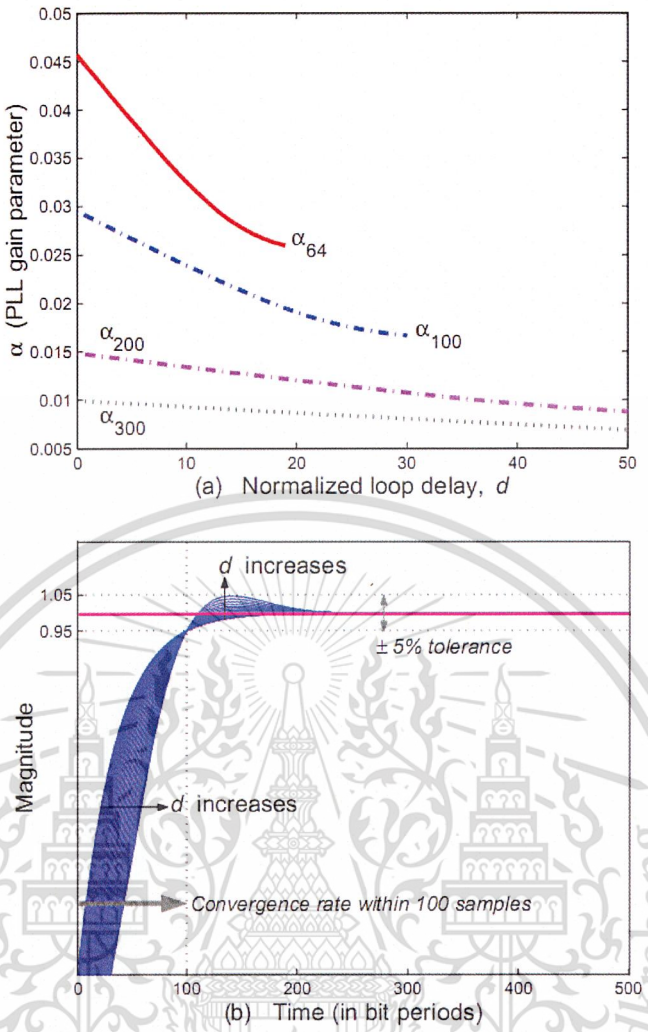


Figure 4.4 (a) α_C's satisfying the system stability and the convergence rate of C samples for different delays, and (b) the system step responses using α_{100} for the delays ranging from 0 to 30T.

After obtaining α, we can then design β based on a linear analysis of second-order PLL for a given amount of frequency offset. In this analysis, the frequency error is modeled as $\tau_k = kf_d$ where f_d is the amount of frequency offset as a fraction of T.

Consider the second-order PLL update equations, which are given by

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \beta\hat{\varepsilon}_{k-d}, \tag{4.12}$$

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha\hat{\varepsilon}_{k-d} + \hat{\theta}_{k+1}. \tag{4.13}$$

Assuming that there is no noise in TED (i.e., using $\hat{\varepsilon}_k = \varepsilon_k = \tau_k - \hat{\tau}_k$), the system transfer function of a second-order PLL can be expressed as

$$G(z) = \frac{\hat{\Gamma}(z)}{\Gamma(z)} = \frac{(\alpha+\beta)z^{-(d+1)} - \alpha z^{-(d+2)}}{1 - z^{-1} + z^{-2} + (\alpha+\beta)z^{-(d+1)} - \alpha z^{-(d+2)}}, \tag{4.14}$$

and its corresponding error transfer function is

$$E(z) = \Gamma(z) - \hat{\Gamma}(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 - z^{-1} + z^{-2} + (\alpha+\beta)z^{-(d+1)} - \alpha z^{-(d+2)}}\Gamma(z). \tag{4.15}$$

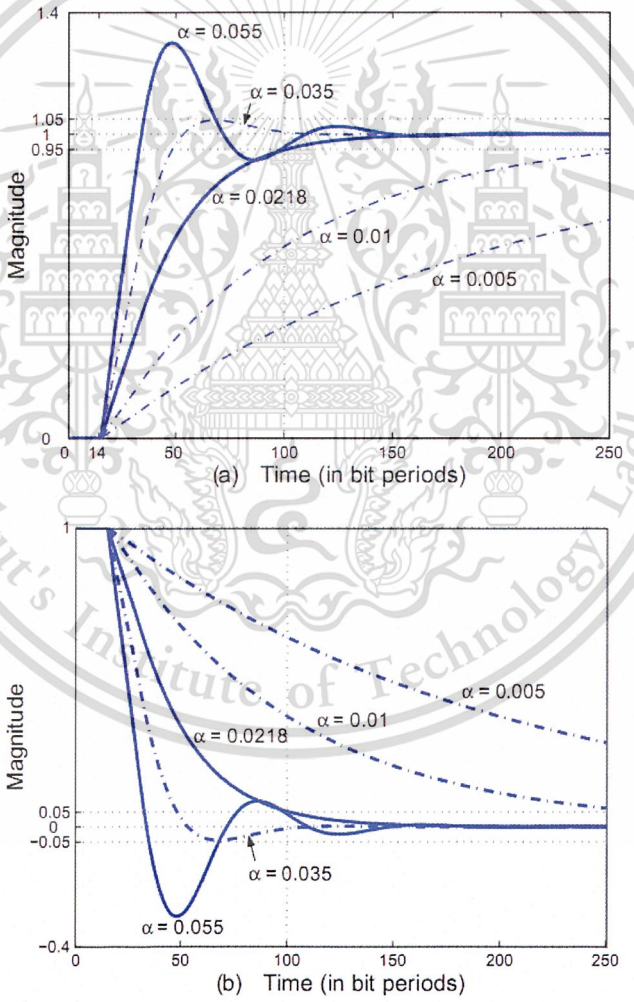


Figure 4.5 (a) The system step responses and (b) the error responses with different α's for d = 14.

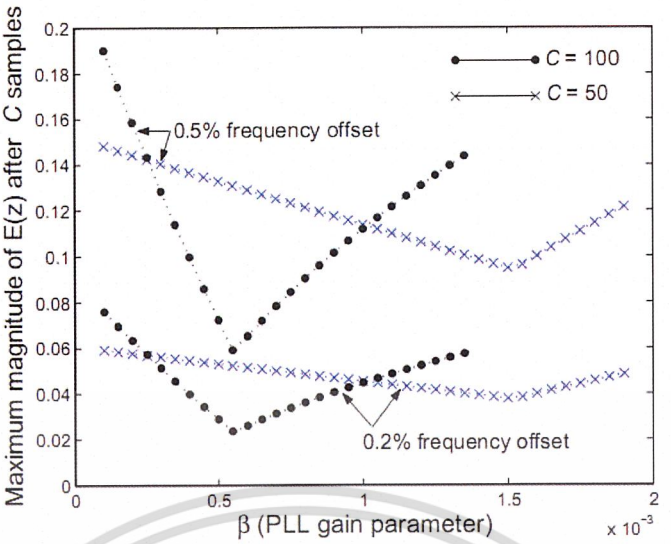


Figure 4.6 Maximum magnitude of $E(z)$ after C samples using $d = 14$ and α_C.

Again, one possible criterion to choose β, for given d, C, and α_C, is to pick the one that satisfies both the system stability and the convergence rate for a given amount of frequency offset. This can be done as follows. Given d, C, and α_C, we first find β's that satisfy the system stability. As seen in (4.14) and (4.15), the PLL is stable whenever all poles of (4.14) or (4.15) are inside the unit circle. Among a set of β's that satisfies the system stability, we pick one β so that it yields the lowest magnitude of $E(z)$ after C samples to reduce the effect of the noise in the timing loop. Figure 4.6 plots the maximum magnitude of (4.15) after C samples as a function of β's for $d = 14$ and α_C. Interestingly, it turns out that this analysis yields the same β, regardless of the amount of frequency offset, but the corresponding magnitude of $E(z)$ is, however, proportional to the amount of frequency offset.

It should be noted that the method of designing the PLL gain parameters explained so far is based on the assumptions that the S-curve slope of (4.4) is unity at the origin and there is no noise in the system. Therefore, before using α and β obtained from this analysis in the timing loop, we must normalize the S-curve slope of TED to be one at the origin.

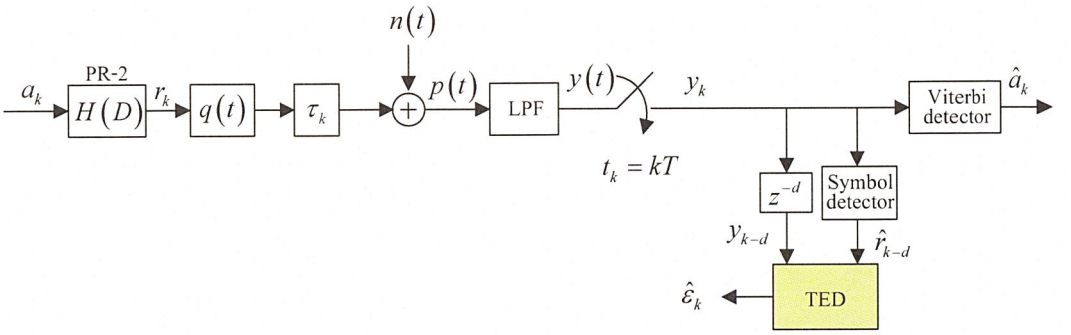


Figure 4.7 The perfectly equalized channel model to find the S-curve.

4.3.3 Determination of the S-curve

The S-curve or timing function [50] is defined as the mean of $\{\hat{\varepsilon}_k\}$, assuming that all decisions are correct (i.e., $r_k = \hat{r}_k$ for all k) and the input data symbols are uncorrelated with unit energy, i.e.,

$$S_{\text{TED}}(\varepsilon) = E\left[\hat{\varepsilon}_k \middle| \varepsilon, r_k = \hat{r}_k \text{ for } \forall k\right], \tag{4.16}$$

where $\varepsilon = \tau - \hat{\tau}$ is the timing error. Because this function looks like an "S" (rotated by 90 degrees), it is named the "S-curve". The S-curve can be used to determine the performance of TED.

For a given channel impulse response, the S-curve can be derived analytically as described in [50]. For instance, let us consider the perfectly equalized PR2 channel model shown in Figure 4.7, i.e., $H(D) = 1+2D+D^2$. The timing function of the M&M TED for this channel can be computed from

$$\begin{aligned} S_{\text{TED}}(\varepsilon) &= E\left[\hat{\varepsilon}_k \middle| \varepsilon, \hat{r}_{k-1} = r_{k-1}, \hat{r}_k = r_k\right] \\ &= E\left[y_k r_{k-1} - y_{k-1} r_k\right] \\ &= E\left[y_k r_{k-1}\right] - E\left[y_{k-1} r_k\right], \end{aligned} \tag{4.17}$$

where $E[\bullet]$ is the expectation operator. For a PR2 channel, the noiseless channel output is given by

$$r_k = a_k + 2a_{k-1} + a_{k-2}, \tag{4.18}$$

and the sampler output can be expressed as

$$y_k = \sum_i a_i[\text{sinc}(kT - iT + \varepsilon) + 2\text{sinc}(kT - T - iT + \varepsilon) + \text{sinc}(kT - 2T - iT + \varepsilon)], \tag{4.19}$$

where $\text{sinc}(t)$ is a sinc function. Given (4.18) and (4.19), the first term in (4.17) can be expressed as

$$E[y_k r_{k-1}] = \text{sinc}(-T + \varepsilon) + 4\text{sinc}(\varepsilon) + 6\text{sinc}(T + \varepsilon) + 4\text{sinc}(2T + \varepsilon) + \text{sinc}(3T + \varepsilon), \tag{4.20}$$

and the second term in (4.17) can be written as

$$E[y_{k-1} r_k] = \text{sinc}(T + \varepsilon) + 4\text{sinc}(\varepsilon) + 6\text{sinc}(-T + \varepsilon) + 4\text{sinc}(-2T + \varepsilon) + \text{sinc}(-3T + \varepsilon). \tag{4.21}$$

Substituting (4.20) and (4.21) into (4.17) yields

$$S_{\text{TED}}(\varepsilon) = \frac{\sin(\varepsilon\pi/T)}{3\pi + \varepsilon\pi/T} - 4\frac{\sin(\varepsilon\pi/T)}{2\pi + \varepsilon\pi/T} + 5\frac{\sin(\varepsilon\pi/T)}{\pi + \varepsilon\pi/T} - 5\frac{\sin(\varepsilon\pi/T)}{-\pi + \varepsilon\pi/T} + 4\frac{\sin(\varepsilon\pi/T)}{-2\pi + \varepsilon\pi/T} - \frac{\sin(\varepsilon\pi/T)}{-3\pi + \varepsilon\pi/T}. \tag{4.22}$$

Assuming that ε is very small, (4.22) can be rewritten as

$$\begin{aligned} S_{\text{TED}}(\varepsilon) &\approx \frac{\sin(\varepsilon\pi/T)}{3\pi} - 4\frac{\sin(\varepsilon\pi/T)}{2\pi} + 5\frac{\sin(\varepsilon\pi/T)}{\pi} \\ &\quad + 5\frac{\sin(\varepsilon\pi/T)}{\pi} - 4\frac{\sin(\varepsilon\pi/T)}{2\pi} + \frac{\sin(\varepsilon\pi/T)}{3\pi} \\ &= \frac{40}{6\pi}\sin(\varepsilon\pi/T). \end{aligned} \tag{4.23}$$

The S-curve slope can be obtained by differentiating (4.23) with respect to ε, i.e.,

$$\frac{\partial S_{\text{TED}}(\varepsilon)}{\partial \varepsilon} = \frac{\pi}{T}\frac{40}{6\pi}\cos\left(\frac{\varepsilon\pi}{T}\right). \tag{4.24}$$

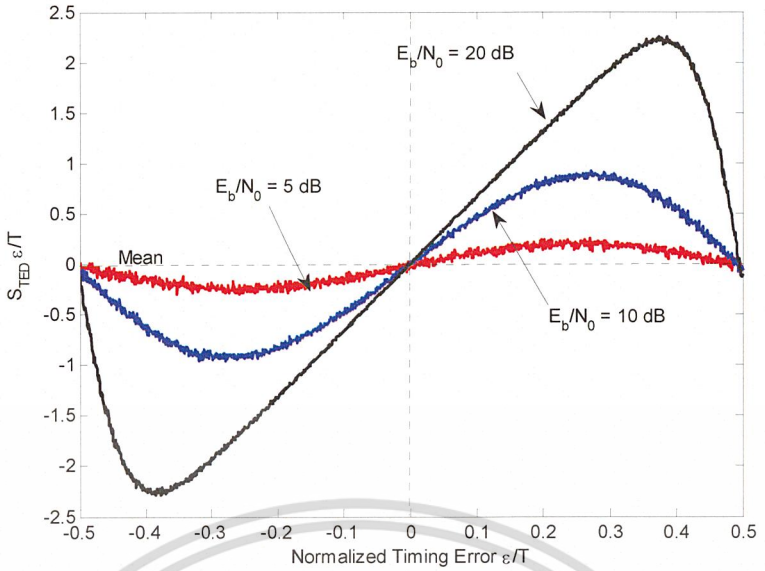


Figure 4.8 S-curves of the M&M TED for a PR2 channel based on conventional timing recovery with instantaneous decision.

Then, the S-curve slope of a PR-2 channel can be obtained by setting $\varepsilon = 0$ in (4.24), i.e.,

$$\frac{\partial S_{\text{TED}}(\varepsilon)}{\partial \varepsilon} = \frac{40}{6T}. \tag{4.25}$$

As a consequence, the estimated timing error $\hat{\varepsilon}_k$ must be scaled by $40/6T$ so as to make the S-curve slope to be one at the origin. That is the $\hat{\varepsilon}_k$ must be computed from

$$\hat{\varepsilon}_k = \frac{6T}{40}\left[y_k \hat{r}_{k-1} - y_{k-1}\hat{r}_k\right]. \tag{4.26}$$

On the other hand, when the channel impulse response is not known, one can still obtain the S-curve by simulation. This is done by opening the timing loop in Figure 4.7 (i.e., discarding a loop filter and a VCO), sampling the received signal $y(t)$ at time kT (i.e., $\hat{\tau} = 0$), and replacing τ with ε. Hence, we measure the time average of $\{\hat{\varepsilon}_k\}$ for a given ε to obtain a single value of $S_{\text{TED}}(\varepsilon)$. We compute $S_{\text{TED}}(\varepsilon)$ for ε ranging from $-0.5T$ to $0.5T$. Eventually, the S-curve is obtained by plotting a graph between ε/T and $S_{\text{TED}}(\varepsilon)/T$.

Figure 4.8 shows the S-curve of the M&M TED for a PR2 channel based on conventional timing recovery with instantaneous hard decision, which is extracted by a simple ternary symbol-by-symbol decision with 5 threshold level, i.e.,

$$\hat{r}_k = \begin{cases} 4 & \text{if } y_k > 3 \\ 2 & \text{if } 1 \geq y_k \geq 3 \\ 0 & \text{otherwise} \\ -2 & \text{if } -1 \leq y_k \leq -3 \\ -4 & \text{if } y_k < -3 \end{cases}, \tag{4.27}$$

Clearly, the timing function is odd symmetric with respect to $\varepsilon = 0$. This implies that the sampling phase offset updated by a PLL using the M&M TED will settle down in the steady state at $\varepsilon = 0$. It appears that simulations agree well with the timing function only for small ε/T's. This is because the assumption that $r_k = \hat{r}_k$ for $\forall k$ is no longer valid as ε/T increases. That is the reason why the range in which $S_{\text{TED}}(\varepsilon)$ matches the normalized timing function at high per-bit SNR, E_b / N_0, is larger than that at low E_b / N_0.

It should be pointed out that zero crossings of the S-curve represent the equilibrium points of operation, where the PLL can track the timing offsets well. In practice, beyond ε/T, the PLL also exhibits other equilibrium points at $\varepsilon = 0, \pm T, \pm 2T, ..., \pm nT$, where n is an integer. The noise and other disturbances in a system can produce a large phase deviation, which may cause a transition from one equilibrium point to another. When this happens, we say that a cycle slip is occurred, which can cause a burst of errors in data detection process. Figure 4.9 demonstrates an example of a cycle slip.

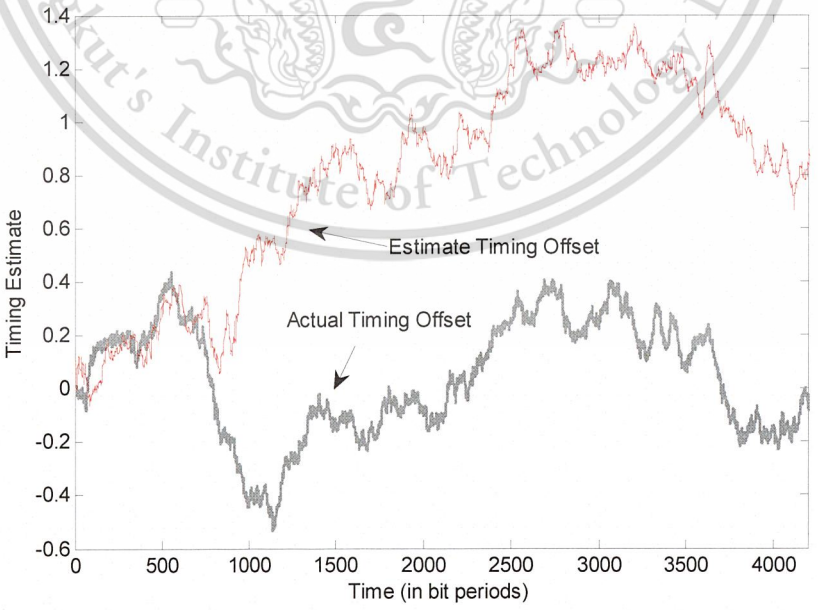


Figure 4.9 An example of a cycle slip.

This figure implies that the PLL can track the actual timing offset, τ, well at the beginning of the data packet. Then, when a cycle slip occurs, the timing recovery unit gradually loses track of the actual timing offset until it settles down at the offset corresponding to multiples of symbol durations. In other words, a cycle slip causes the PLL to operate at another equilibrium point. That is why in this example, the estimated timing offset, $\hat{\tau}$, differs from τ by approximately T at the end of the data packet. Several approaches have been proposed to deal with a cycle slip [52, 53, 54, 55]. The next Section, we will describe new simple timing recovery architectures to develop the overall performance of conventional timing recovery.

4.4 Bi-Directional Timing Recovery

This Section, we proposes a simple timing recovery architecture, which consists of two timing recovery blocks running in parallel as depicted in Figure 4.10. The first block (i.e., branch **A**) employs a conventional timing recovery to sample the readback signal, while the second block (i.e., branch **B**) reverses the whole readback signal before passing the reversed readback signal to conventional timing recovery. The outputs of the two timing recovery blocks are averaged and sent the resulting sequence to the Viterbi detector to determine the most likely input sequence. We refer to the proposed timing recovery architecture as "bi-directional timing recovery". It can be seen in simulations that the bi-directional timing recovery can help improve the system performance if compared to conventional timing recovery.

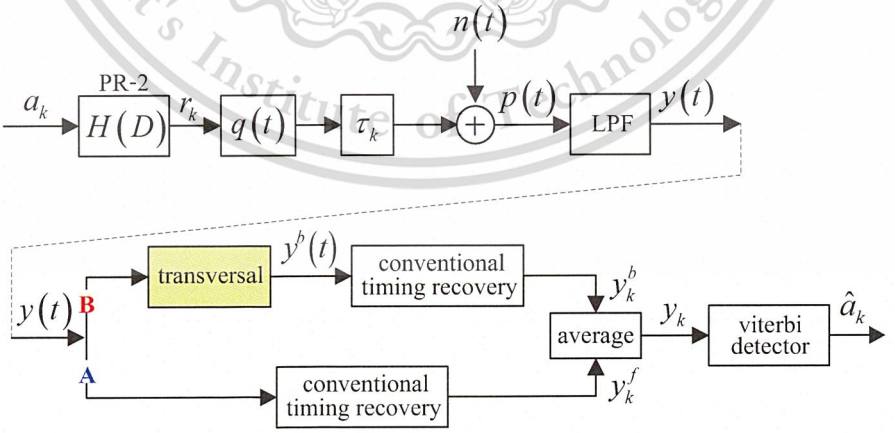


Figure 4.10 A perfectly equalized PR2 channel model with bi-directional timing recovery.

The key idea of bi-directional timing recovery is to sample the readback signal both in forward direction and in backward direction. Specifically, for forward direction (i.e., branch **A**), the readback signal is sampled by the same conventional timing recovery as explained in Section 4.2 to obtain a sequence y_k^f. Nonetheless, for backward direction (i.e., branch **B**), the whole readback signal is reversed to obtain the reversed signal $y^b(t)$ before passing it to conventional timing recovery to obtain a sequence y_k^b.

The signal $y^b(t)$ is sampled at time $kT + \hat{\tau}_k^b$ to obtain

$$y_k^b = y^b\left(kT + \hat{\tau}_k^b\right), \tag{4.28}$$

where $\hat{\tau}_k^b$ is the k-th sampling phase offset in backward direction. Note that to sample the reversed signal $y^b(t)$, we set $\hat{\tau}_0^b = -\hat{\tau}_L^f$, where $\hat{\tau}_0^b$ the first sampling phase is offset in backward direction and $\hat{\tau}_L^f$ is the last sampling phase offset in forward direction. We still use the M&M TED algorithm to compute the estimate of the backward timing error, ε_k^b, which can be obtained by

$$\hat{\varepsilon}_k^b = \frac{6T}{40}\left\{y_k^b \hat{r}_{k-1} - y_{k-1}^b \hat{r}_k\right\}. \tag{4.29}$$

Then, the next sampling phase offset in backward direction is updated by a 1st-order PLL according to

$$\hat{\tau}_{k+1}^b = \hat{\tau}_k^b + \alpha\hat{\varepsilon}_k^b. \tag{4.30}$$

where the same PLL gain parameter, α, is employed.

Because conventional timing recovery in forward direction produces a set of $\left\{y_k^f, \hat{\tau}_k^f\right\}$ and that in backward direction also produces a set of $\left\{y_k^b, \hat{\tau}_k^b\right\}$, there are two options to exploit this information to improve the performance of synchronization. The first option is to find the averaged sampling phase offset according to

$$\hat{\tau}_k = \frac{1}{2}\left(\hat{\tau}_k^f + \hat{\tau}_k^b\right). \tag{4.31}$$

Then, we resample the readback signal $y(t)$ using a set of $\{\hat{\tau}_k\}$ to obtain $y_k = y(kT + \hat{\tau}_k)$. However, to reduce the complexity, we can directly average the sampler outputs $\{y_k^f, y_k^b\}$ according to

$$y_k = \frac{1}{2}\left(y_k^f + y_k^b\right). \tag{4.32}$$

Consequently, a sequence $\{y_k\}$ is sent to the Viterbi detector to perform sequence detection. Figure 4.11 shows the algorithm of bi-directional timing recovery, which will be used to compared the performance with the conventional timing recovery.

(A-1) Initialize $\hat{\tau}_0^f = 0$

(A-2) For $k = 0,1,\ldots, L-1$

(A-3) $y_k^f = y^f\left(kT + \hat{\tau}_k^f\right)$

(A-4) $\hat{\varepsilon}_k^f = \frac{6T}{40}\left\{y_k^f \hat{r}_{k-1} - y_{k-1}^f \hat{r}_k\right\}$

(A-5) $\hat{\tau}_{k+1}^f = \hat{\tau}_k^f + \alpha\hat{\varepsilon}_k^f$

(A-6) End

(A-7) Reverse the readback signal $y(t)$ to obtain $y^b(t)$

(A-8) Initialize $\hat{\tau}_0^b = -\hat{\tau}_L^f$

(A-9) For $k = 0,1,\ldots, L-1$

(A-10) $y_k^b = y^b\left(kT + \hat{\tau}_k^b\right)$

(A-11) $\hat{\varepsilon}_k^b = \frac{6T}{40}\left\{y_k^b \hat{r}_{k-1} - y_{k-1}^b \hat{r}_k\right\}$

(A-12) $\hat{\tau}_{k+1}^b = \hat{\tau}_k^b + \alpha\hat{\varepsilon}_k^b$

(A-13) End

(A-14) Average the sampler outputs by $y_k = \frac{1}{2}\left(y_k^f + y_k^b\right)$

(A-15) Send y_k to the Viterbi detector to obtain $\hat{a}_k$

Figure 4.11 An algorithm of bi-directional timing recovery.

4.5 Performance Results

4.5.1 Performance of Conventional Timing Recovery

Consider the perfectly equalized PR2 channel model shown in Figure 4.2. Apparently, the conventional receiver performs timing recovery and ML equalization separately. Therefore, the overall performance of an uncoded system is mainly determined by how well conventional timing recovery is. In this Section, we investigate the performance of conventional timing recovery when operating in a system without frequency offset. We consider the case where the symbol detector used in the timing loop is a hard slicer (i.e., a memoryless multi-level slicer), where the instantaneous hard decision is given by (4.27). For data detection process, the sampler outputs y_k are applied to the Viterbi detector with a decision delay of $60T$ to determine the most likely input sequence. Each BER point was computed using as many data packets as needed to collect 1000 error bits.

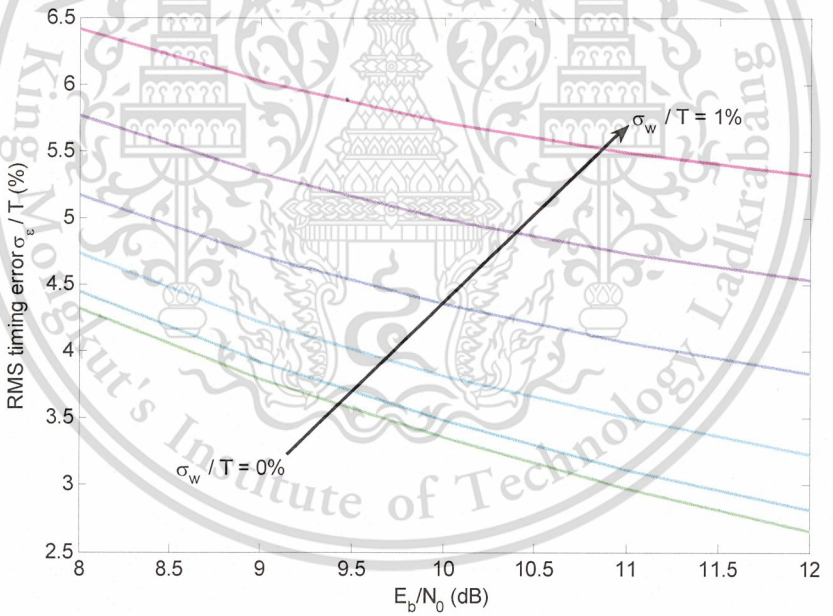


Figure 4.12 RMS timing jitter σ_ε / T performances as a function of E_b / N_0's for the perfectly equalized PR2 channel with different σ_w / T's (without frequency offset).

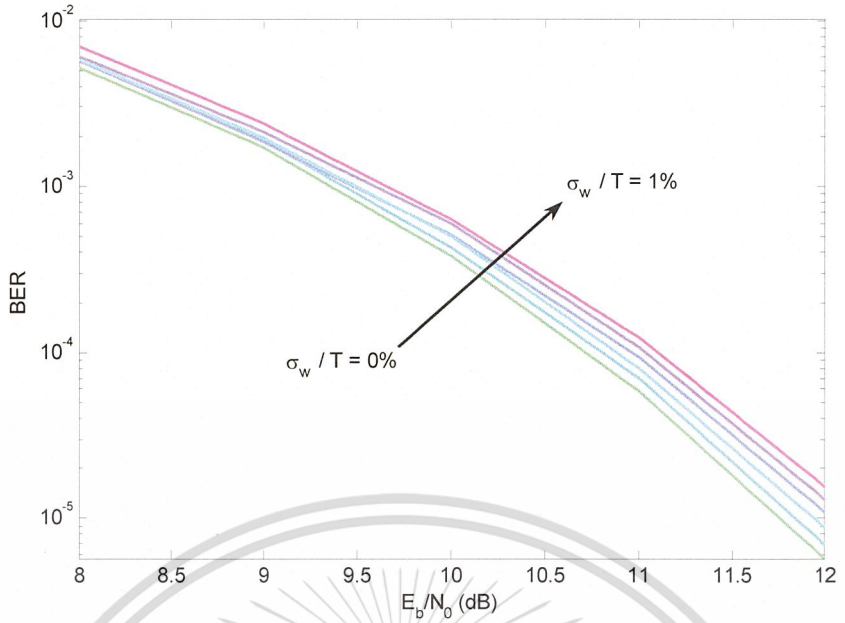


Figure 4.13 BER performances as a function of E_b / N_0's for the perfectly equalized PR2 channel with different σ_w / T's (without frequency offset).

For the system without frequency offset, a first-order PLL is sufficient to be used for the timing update operation. In this case, we assume perfect acquisition by setting $\tau_0 = 0$ so that a preamble is not needed, and one data packet consists of 4096 data bits. Figure 4.12 and Figure 4.13 compare the performance of the RMS timing error, $\sigma_\varepsilon = \sqrt{E\left[\left(\tau_k - \hat{\tau}_k\right)^2\right]}$, and the BER as a function of E_b / N_0's, respectively. Note that the PLL gain parameter, α, was designed to recover the phase change within 100 symbols based on a linearized model of first-order PLL (i.e., $\alpha_{100} = 0.0270$), as described in Section 4.3.1. It is evident that the larger the random walk parameter σ_ε / T, the worse the performance (both in terms of σ_ε / T and BER). Observe that the lower the σ_ε / T, the lower the BER. Therefore, one can use either σ_ε / T or BER as a measure to compare the performance of different timing recovery schemes.

As illustrated in Figure 4.13, it can be implied that conventional timing recovery does not perform well in uncoded systems when the timing error is large or when operating in a system that requires fast convergence. The easiest way to improve the performance of conventional timing recovery is to replace the hard slicer with either the soft slicer [55] or the Viterbi detector with a short decision delay [1]. However, we observed that only a small performance improvement is obtained.

4.5.2 Performance of Bi-directional Timing Recovery

We consider the system in moderate and severe timing offsets (i.e., $\sigma_w / T = 0.7\%$ and $\sigma_w / T = 1.2\%$). We employ the PLL gain parameter, α, designed to recover phase change within C = 100 symbols based on a linearized model of PLL [3], assuming that the S-curve slope [3] is one at the origin, and there is no noise in the system. The α designed for the delay of $4T$ is 0.0270. We also assume that one data packet consists of 4096 data bits. The performance of different timing recovery schemes will be compared in terms of both RMS and BER.

We first compare the performance of different schemes at moderate timing offset, i.e., $\sigma_w / T = 0.7\%$, by plotting σ_ε / T performance as a function of per-bit SNRs (E_b / N_0's) in decibel (dB), as depicted in Figure 4.14 (a), where the curve labeled "Trained PLL" is conventional timing recovery whose PLL has access to all correct decisions, thus serving as a lower bound for all symbol-rate timing recovery schemes that are based on PLL.

Clearly, the bi-directional timing recovery yields lower RMS timing error than other (symbol-rate) timing recovery schemes. This might be because the bi-directional timing recovery can be viewed as oversampled timing recovery [56] that oversamples the readback signal by twice the symbol rate to get more timing information to perform synchronization.

Figure 4.14 (b) compares the RMS performance of different timing recovery schemes at severe timing offset, i.e., $\sigma_w / T = 1.2\%$. Again, the bi-directional timing recovery still performs better than other timing recovery schemes in terms of σ_ε / T. In addition, we can see that the performance gain obtained from the bi-directional timing recovery increases as the severity of the timing offset, σ_ε / T, increases (as shown in Figure 4.14(a) and Figure 4.14(b)).

We also compare the BER performance of different timing recovery schemes at $\sigma_w / T = 0.7\%$ and $\sigma_w / T = 1.2\%$ as depicted in Figure 4.15, where the curve labeled "Perfect timing" represents the conventional timing recovery system that uses $\tau_k = \hat{\tau}_k$ to sample $y(t)$. It is evident that the bi-directional timing recovery has lower BER than conventional timing recovery at both $\sigma_w / T = 0.7\%$ and $\sigma_w / T = 1.2\%$. Specifically, at BER = 10^{-4}, for severe timing offsets ($\sigma_w / T = 1.2\%$), the bi-directional timing recovery provides a performance gain of 0.2 dB and 0.3 dB over the Trained PLL and the conventional timing recovery , respectively.

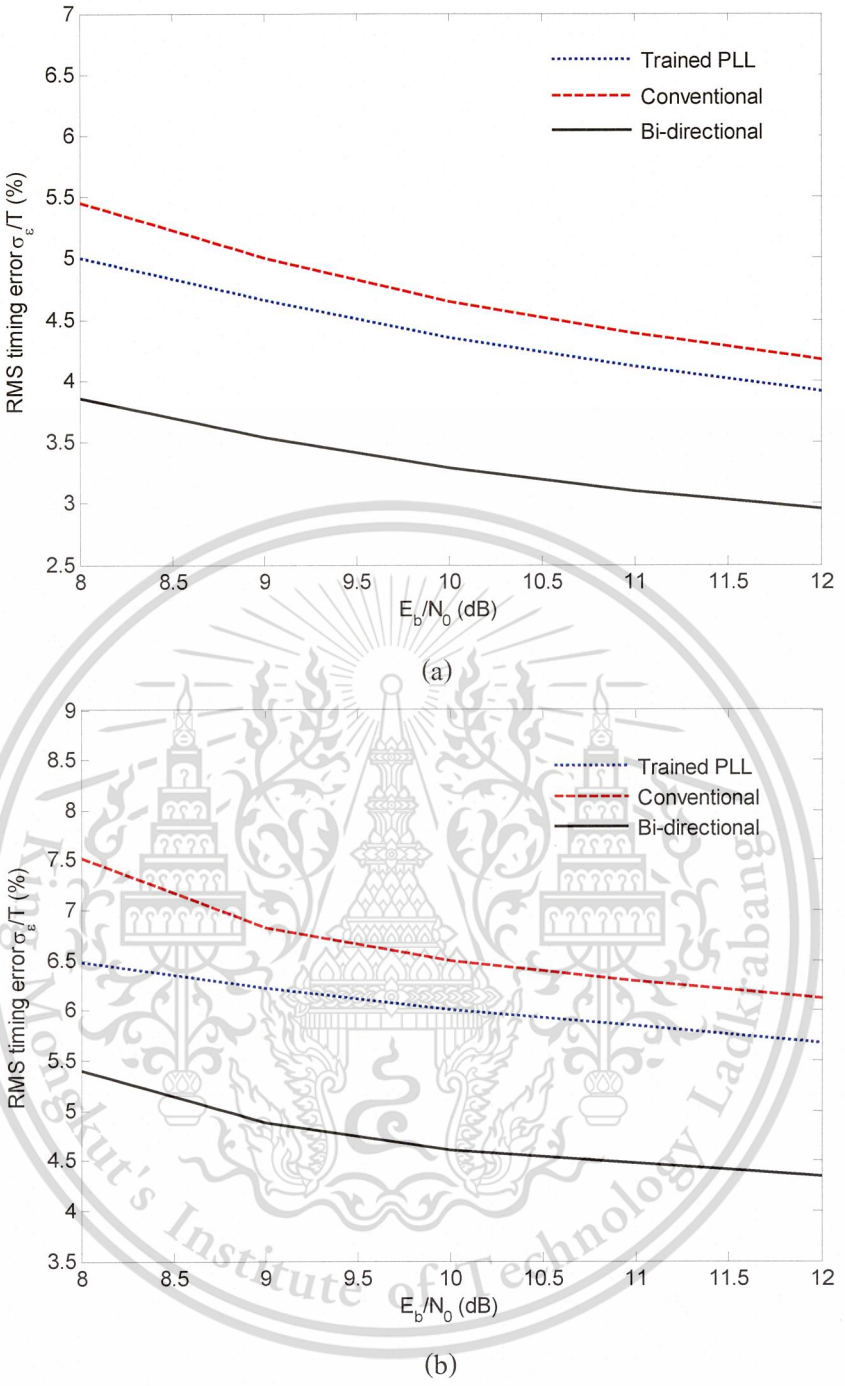
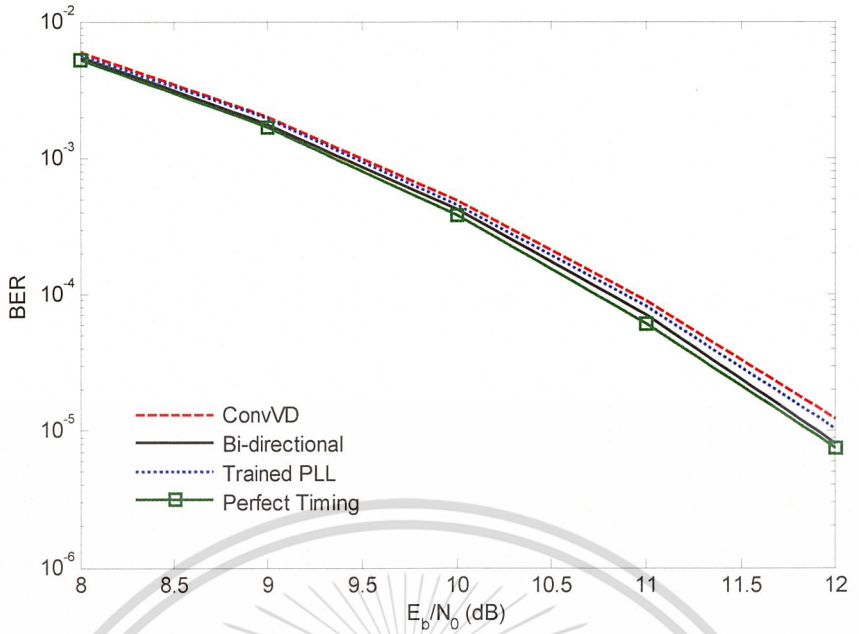
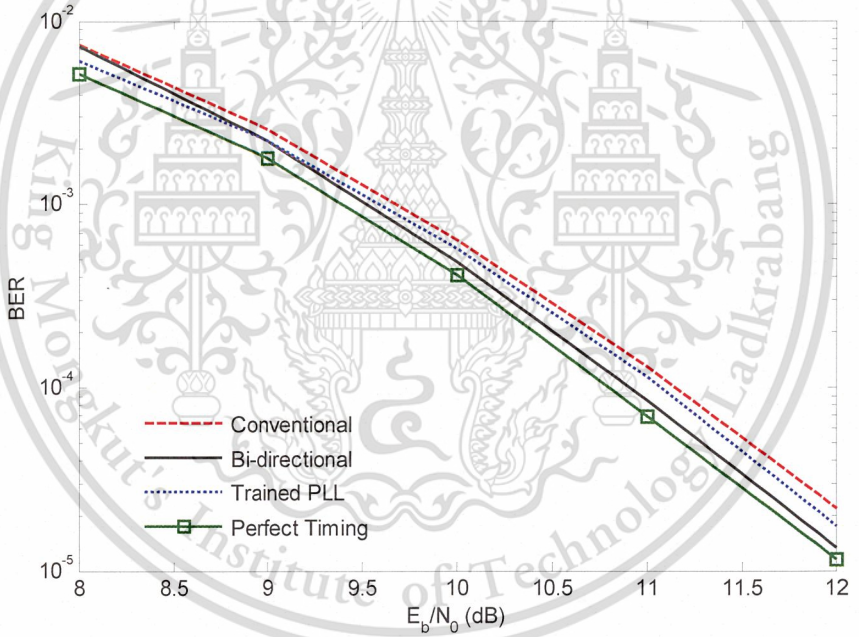


Figure 4.14 RMS performances of different timing recovery schemes at (a) $\sigma_w / T = 0.7\%$ and (b) $\sigma_w / T = 1.2\%$.



(a)



(b)

Figure 4.15 BER performances of different timing recovery schemes at (a) $\sigma_w / T = 0.7\%$ and (b) $\sigma_w / T = 1.2\%$.

Based on extensive simulations, we found that bi-directional timing recovery based on both options yields a similar result. Therefore, in this thesis, we will consider only bi-directional timing recovery based on the second option because it has less complexity if compared to the first option.

However, we see that the performance of bi-directional timing recovery can slightly only improve the performance when is compared to the conventional timing recovery. Thus, a new more effective timing recovery system is required to instead both of these. Therefore, to improve the performance of both the conventional receiver and bi-directional timing recovery, we also offer a new timing recovery called "modified per-survivor iterative timing recovery" [57]. The results showed that the new timing recovery better than the conventional receiver and bi-directional timing recovery as presented in the next chapter.

4.6 Summary

We briefly explained how conventional timing recovery that is based on a PLL. A method of designing the PLL gain parameters based on a linearized model of PLL was given. According to our design criterion, it turns out that this method results in the same values of the PLL gain parameters, regardless of the amount of frequency offset. Then, we have demonstrated that for low-to-moderate SNRs, the conventional timing recovery does not perform well, especially when the timing error is large or when the system requires a fast convergence.

Next, we propose the bi-directional timing recovery for perpendicular magnetic recording systems, which utilizes the conventional timing recovery to sample the readback signal both in the forward and in the backward direction. Simulation results show that the bi-directional timing recovery performs better than the trained PLL and the conventional timing recovery, especially when the timing error is large. This is because the bi-directional timing recovery acts as the oversampled timing recovery, which oversamples the readback signal by twice the symbol rate to get more timing information to perform synchronization.

CHAPTER 5

A MODIFIED PER-SURVIVOR ITERATIVE TIMING RECOVERY (MPS-ITR) FOR CODED PARTIAL RESPONSE CHANNELS

In this chapter, we will describe a modified per-survivor iterative timing recovery scheme, which exploits a new split-preamble strategy in conjunction with a per-survivor processing soft-output Viterbi algorithm (PSP-SOVA). The conventional split-preamble strategy places a preamble at the beginning of a data sector and uses it to run a phase-locked loop during acquisition to find an initial phase/frequency offset. However, the modified per-survivor iterative timing recovery scheme splits the preamble into two parts. The first part is placed at the beginning of the data sector, whereas the second part is divided into small clusters, each of which is then embedded uniformly within the data stream. This split preamble is utilized to adjust the branch metric calculation in PSP-SOVA to ensure that the survivor path occurs in a correct direction. Results indicate that the modified per-survivor iterative timing recovery scheme yields a better performance than the conventional receiver with the timing recovery separate from turbo equalization, and the iterative timing recovery scheme proposed in [17, 20], especially when the timing jitter is large. Then, we also show that the modified per-survivor iterative timing recovery scheme can automatically correct the cycle slips much more efficiently than the others. In addition, all the iterative timing recovery schemes will be investigated in the realistic magnetic recording systems for perpendicular recording channels (as described in Section 2.4.2) and then compared with the conventional schemes. This study will help us decide whether or not the proposed schemes can be feasibly employed in real-life applications as compared to conventional ones.

5.1 Introduction

Timing recovery is a crucial component in magnetic recording systems. It is the process of synchronizing the sampler with the received analog signal. The quality of synchronization has a direct impact on overall performance. The large coding gains of iterative ECCs enable reliable communication at very low SNRs. This means that timing recovery must be performed at an SNR

lower than ever before. A conventional receiver performs the timing recovery and turbo equalization [16] *separately*. Specifically, the conventional timing recovery ignores the presence of ECCs and thus fails to work properly when the SNR is low.

To improve the performance of the conventional receiver, Kovintavewat *et al.* [17] proposed a PS-ITR scheme, which *jointly* performs timing recovery, equalization, and error-correction decoding. It is realized by first applying the PSP technique [18], a technique of jointly estimating a data sequence and unknown parameters, to the SOVA [21], resulting in a per-survivor SOVA equalizer, denoted as "PSP-SOVA" [20]. Then, PSP-SOVA iteratively exchanges soft information with a SISO decoder. As investigated in [20], the PS-ITR outperforms the conventional receiver because it can automatically correct a cycle slip [53] after a few number of turbo iterations.

In practice, timing recovery performs in two modes, namely acquisition and tracking modes [3]. The acquisition mode is performed to acquire the initial timing offset with an aid of a preamble (or a training sequence of known symbols), whereas the tracking mode is performed to refine the timing estimates based on a user data sequence (unknown symbols). In general, to have good timing estimates before the user data sequence starts, the conventional split-preamble strategy places all known symbols at the beginning of a data sector [58]. Nonetheless, in [23] – [24], the known symbols are arranged in small contiguous clusters and are placed periodically in the data stream, subject to the power constraint on training. Additionally, Nayak *et al.* [25] proposed the optimal training symbol placement strategy to minimize a Cramér-Rao bound by splitting the known symbols into two halves and places them at the beginning and at the end of a data sector. This split-preamble arrangement leads to a reduced frequency estimation error variance and also greatly reduces the occurrence of cycle slips.

In general, the PS-ITR uses the conventional split-preamble strategy, which places a preamble at the beginning of a data sector and uses it to run a PLL [3] during acquisition to find the initial timing offset. As a consequence, to further improve the performance of the PS-ITR, we propose a MPS-ITR scheme. In essence, the MPS-ITR is same as the PS-ITR, except that a *modified* PSP-SOVA is utilized instead of PSP-SOVA. This modified PSP-SOVA employs the new split-preamble strategy, which splits a preamble into two parts. The first part is placed at the beginning of a data sector, while the second part is divided into many small clusters, each of which is therefore embedded uniformly within the user data sequence.

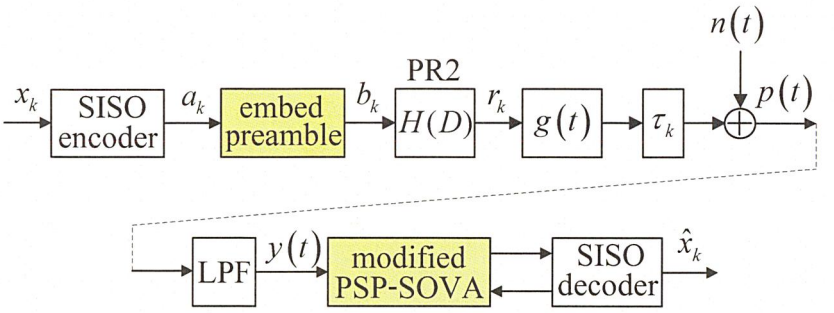


Figure 5.1 A channel model with the modified per-survivor iterative timing recovery (MPS-ITR) scheme.

This split preamble is utilized to adjust the branch metric calculation in PSP-SOVA to guarantee that the survivor path occurs in a correct direction.

This chapter is organized as follows. After explaining the system model in Section 5.2, In Section 5.3 we explain the concept of the PSP-SOVA, and also describe how the modified PSP-SOVA works in Section 5.4. The complexity comparison of the different iterative timing recovery schemes is given in Section 5.5. In Section 5.6, we investigate the MPS-ITR in a realistic channel system. Simulation results and discussion are given in Section 5.7. Finally, Section 5.8 summarizes this chapter.

5.2 System Descriptions

Consider the coded partial-response (PR) channel in Figure 5.1, where $H(D)=\sum_{k=0}^{\nu-1} h_k D^k$ $=1+2D+D^2$ is a PR2 channel, D is the delay operator, and ν is channel memory. The message sequence $x_k \in \{0,1\}$ is encoded by an error-correction encoder and is mapped to a binary sequence $a_k \in \{\pm 1\}$ of length L. Next, a preamble is inserted in a sequence a_k to obtain a sequence $b_k \in \{\pm 1\}$. The readback signal can then be written as

$$p(t)=\sum_k r_k q(t-kT-\tau_k)+n(t), \tag{5.1}$$

where $r_k = b_k * h_k \in \{0,\pm 2,\pm 4\}$ is the noiseless channel output, $*$ is the convolution operator, $q(t) = \sin(\pi t/T)/(\pi t/T)$ is an ideal zero-excess-bandwidth Nyquist pulse, T is a bit period, and $n(t)$ is an additive white Gaussian noise with a two-sided power spectral density $N_0/2$. The uncertainty in the timing is captured by the timing offset τ_k, which is modeled as a

random walk [49] according to $\tau_{k+1} = \tau_k + \mathcal{N}\left(0, \sigma_w^2\right)$, where σ_w controls the severity of the timing jitter. The random walk model is chosen because of its simplicity and its ability to represent a variety of channels by changing only one parameter.

At the receiver, the readback signal $p(t)$ is filtered by an ideal low-pass filter (LPF), whose impulse response is $q(t)/T$, to eliminate the out-of-band noise, and is then sampled at time $kT + \hat{\tau}_k$, creating

$$y_k = y\left(kT + \hat{\tau}_k\right) = \sum_i r_i q\left(kT + \hat{\tau}_k - iT - \tau_i\right) + n_k, \tag{5.2}$$

where $\hat{\tau}_k$ is the receiver's estimate of τ_k, and n_k is *i.i.d.* zero-mean Gaussian random variable with variance $\sigma_n^2 = N_0/(2T)$, i.e., $n_k \sim \mathcal{N}\left(0, \sigma_n^2\right)$.

Conventional timing recovery is based on a 2nd-order PLL [3], which consists of a TED, a loop filter, and a VCO. A decision-directed TED computes the receiver's estimate of the timing error $\varepsilon_k = \tau_k - \hat{\tau}_k$ using the well-known M&M TED algorithm [11] according to

$$\hat{\varepsilon}_k = \frac{6T}{40}\left\{y_k \hat{r}_{k-1} - y_{k-1}\hat{r}_k\right\}, \tag{5.3}$$

where $\hat{r}_k$ is an estimate of r_k, and the constant $6T/40$ [61] is used to normalize the timing function of the M&M TED in (5.3) to have unit slope at origin [3]. For simplicity, this thesis considers the M&M TED algorithm because its complexity is much lower than the TED proposed in [12]. We also assume no frequency offset in the system. Thus, the next sampling phase offset can be updated by a first-order PLL according to

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha\hat{\varepsilon}_k, \tag{5.4}$$

where α is a PLL gain parameter [3].

In a conventional setting, conventional timing recovery is followed by a turbo equalizer [16], which iteratively exchanges soft information between the SISO equalizer for the PR2 channel and the SISO decoder for the outer code.

5.3 Per-Survivor Iterative Timing Recovery (PS-ITR)

5.3.1 Per-Survivor Processing Soft-Output Viterbi Algorithm (PSP-SOVA)

As shown in (5.3), the performance of conventional timing recovery relies on the decision $\hat{r}_k$ provided by its own symbol detector, which might yield an unreliable decision. To overcome this drawback, a reliable decision can be extracted by utilizing the already-given information inside the trellis structure [2]. Specifically, each state transition in the trellis uniquely specifies a corresponding symbol. Thus, at least one state transition in each trellis stage will correspond to a correct decision. Using that decision for the timing update operation will improve the performance of timing recovery. The idea of using the information available in the trellis to estimate other unknown parameters is known as PSP [18].

The PSP concept is applied to the PSP-SOVA by embedding the timing recovery step inside the SOVA equalizer so as to perform timing recovery and equalization *jointly*. Figure 5.2 shows the PSP-SOVA algorithm, where the lines starting with * are the additional steps beyond the conventional SOVA. The detail on how PSP-SOVA performs can be explained as follows.

Consider the PR2 trellis structure shown in Figure 5.3. Let $\mathbf{\Psi}_k = \{a_{k-1} \;\; a_{k-2}\}$ denote the state at time k. There are $Q = 2^{\upsilon} = 4$ states in this trellis, labeled as state 0 to state 3, where $\upsilon = 2$ is the PR2 channel memory. Let (p, q) be the state transition from state p to state q, and let $\pi_k(p)$ denote a *predecessor* for state p at time k, defined as the starting state associated with the best state transition leading to state p at time k. We define $\hat{\tau}_k(p)$ as the k-th sampling phase offset for state p at time k, which is used to sample $y(t)$ at time k for the state transition emanating from state p at time k, e.g., $y_k(p) = y_k\left(kT + \hat{\tau}_k(p)\right)$, where $y_k(p)$ is the k-th sampler output for state p at time k.

Consider the k-th stage of the trellis structure. There are two state transitions arriving at state 1 at time k+1, i.e., (0, 1) and (2, 1). First, we sample $y(t)$ using $\hat{\tau}_k(0)$ and $\hat{\tau}_k(2)$ to obtain $y_k(0)$ and $y_k(2)$, respectively. Next, we compute the branch metrics $\rho_k(0,1)$ and $\rho_k(2,1)$ based on (A-6), where $\hat{r}(p,q)$ is the noiseless channel output associated with (p, q). Then, the starting state associated with the best state transition leading to state 1 at time k+1 is chosen according to (A-7).

(A-1) Initialize $\mathbf{\Phi}_0(p) = 0$ for $\forall p$

*(A-2) Initialize $\hat{\tau}_0(p) = 0$ for $\forall p$

(A-3) For $k = 0,1,...,L+\upsilon-1$

(A-4) For $q = 0,1,...,Q$

*(A-5) $y_k(p) = y_k\left(kT + \hat{\tau}_k(p)\right)$ for $\forall p$

(A-6) $\rho_k(p,q) = \left|y_k(p) - \hat{r}(p,q)\right|^2$ for $\forall p$

(A-7) $\pi_{k+1}(q) = \arg\min_p \left\{\mathbf{\Phi}_k(p) + \rho_k(p,q)\right\}$

(A-8) $\mathbf{\Phi}_{k+1}(q) = \mathbf{\Phi}_k\left(\pi_{k+1}(q)\right) + \rho_k\left(\pi_{k+1}(q), q\right)$

(A-9) $\mathbf{S}_{k+1}(q) = \left[\mathbf{S}_k\left(\pi_{k+1}(q)\right) \middle| \pi_{k+1}(q)\right]$

*(A-10) $\hat{\varepsilon} = \frac{6T}{40}\left\{y_k\left(\pi_{k+1}(q)\right)\hat{r}\left(\pi_k\left(\pi_{k+1}(q)\right), \pi_{k+1}(q)\right)\right.$
$\left. - y_{k-1}\left(\pi_k\left(\pi_{k+1}(q)\right)\right)\hat{r}\left(\pi_{k+1}(q), q\right)\right\}$

*(A-11) $\hat{\tau}_{k+1}(q) = \hat{\tau}_k\left(\pi_{k+1}(q)\right) + \alpha\hat{\varepsilon}$

(A-12) $\Delta_{k+1}(q) = \max_p \left\{\mathbf{\Phi}_k(p) + \rho_k(p,q)\right\} - \mathbf{\Phi}_{k+1}(p)$

(A-13) Initialize $\hat{L}_k(q) = +\infty$ [Soft decision update]

(A-14) For $j = k-\nu,...,k-\delta$

Compare the two paths merging in state q (i.e., $\mathbf{\Psi}_{k+1} = q$)

If $\hat{a}_j^{(1)}\left(\mathbf{\Psi}_{j+1}\right) \neq \hat{a}_j^{(2)}\left(\mathbf{\Psi}_{j+1}\right)$ then

Update $\hat{L}_j\left(\mathbf{\Psi}_{j+1}\right) = \min\left(\hat{L}_j\left(\mathbf{\Psi}_{j+1}\right), \Delta_{k+1}(q)\right)$

End

(A-15) End

(A-16) if $k \geq \delta$ then

Output the soft decision according to $\lambda_k^p = \hat{a}_{k-\delta}\,\hat{L}_{k-\delta}$, which can be extracted from the survivor path that minimizes $\mathbf{\Phi}_{k+1}$

(A-17) End

(A-18) End

Figure 5.2 The PSP-SOVA algorithm, where the lines starting with * are the additional steps beyond the conventional SOVA.

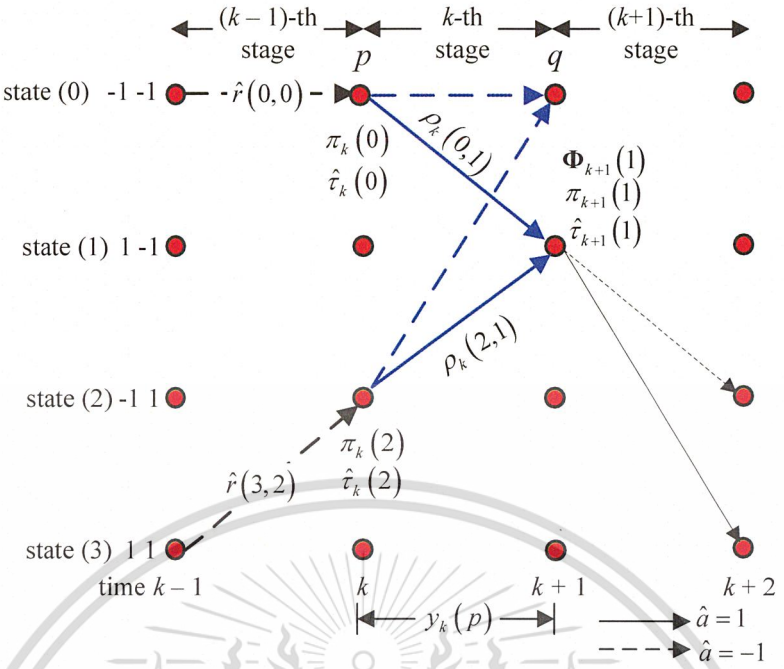


Figure 5.3 The PR2 trellis structure demonstrating how PSP-SOVA performs.

Suppose the state transitions (0, 1) is the best state transition leading to state 1 at time k+1 so that $\pi_{k+1}(1) = 0$. The path metric for state 1 at time k+1, $\Phi_{k+1}(1)$, is updated based on (A-8), and the survivor path for state 1 at time k+1, $\mathbf{S}_{k+1}(1)$, is extended according to (A-9). Hence, we update the next sampling phase offset, $\hat{\tau}_{k+1}(1)$, based on (A-10) and (A-11) using the information from $\mathbf{S}_{k+1}(1)$. This $\hat{\tau}_{k+1}(1)$ will be employed to sample $y(t)$ at time k+1 for the state transitions emanating from state 1 at time k+1. Then, follow these steps according to the SOVA for an entire received signal.

The PSP-SOVA has extra steps for approximating the *a posteriori* log-likelihood ratio (LLR) of each data bit (or a soft decision), which can be briefly explained as follows. At each k-th stage, the metric difference for state q at time k+1, $\Delta_{k+1}(q)$, is computed by (A-12). Then, the tentative LLR is updated based on (A-13), where $\hat{L}_k(q)$ is the k-th LLR associated with state q, $\hat{a}_j^{(m)}(\mathbf{\Psi}_{j+1})$ is the j-th estimated data bit associated with the m-th path that passes $\mathbf{\Psi}_{j+1}$, and $m \in \{1,2\}$ is used to indicate the two paths that merge in state q (where m = 1 represents a correct path and m = 2 represents a wrong path). After a decoding depth, δ, the *a posteriori* LLR of a_k, λ_k^p, is produced by (A-16).

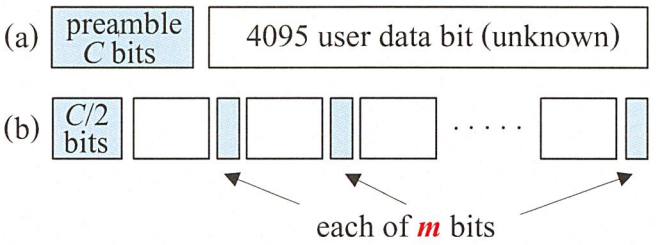


Figure 5.4 (a) the conventional preamble arrangement, and (b) the proposed split-preamble arrangement.

5.4 Modified Per-Survivor Iterative Timing Recovery (MPS-ITR)

5.4.1 Modified Per-Survivor Processing Soft-Output Viterbi Algorithm (Modified PSP-SOVA)

The modified PSP-SOVA is developed based on the PSP-SOVA [57] with an aid of the new split-preamble strategy. Specifically, the PSP-SOVA uses the conventional preamble arrangement, which places all C known bits at the beginning of the data sector as shown in Figure 5.4 (a). We propose a new split-preamble strategy, which splits a C-bit preamble into two parts. The first part of $C/2$ bits is placed at the beginning of the data sector, and the second part of $C/2$ bits is divided into $C/(2m)$ clusters (e.g., m = 1, 2, or 4 bits), each of which is then embedded uniformly within the user data stream as illustrated in Figure 5.4 (b). This split preamble is utilized to adjust the branch metric calculation in modified PSP-SOVA to ensure that the survivor path occurs in a correct direction. Based on extensive simulation search, we found that modified PSP-SOVA with the one-bit split-preamble arrangement (i.e., m = 1) provides the best performance.

Figure 5.5 shows the modified PSP-SOVA algorithm, where (A-7) starting with * is an additional step beyond the PSP-SOVA algorithm, and a constant $6T/40$ in (A-11) is only for the PR2 channel, which can be included in the PLL gain parameter. It should be noted that the modified PSP-SOVA works in a *same* manner as PSP-SOVA does (as described in Section 5.3), except that the modified PSP-SOVA has an *extra* step to account for the split preamble, which can be briefly described as follows.

(A-1) Initialize $\mathbf{\Phi}_0(p) = 0$ for $\forall p$

(A-2) Initialize $\hat{\tau}_0(p) = 0$ for $\forall p$

(A-3) For $k = 0,1,...,L+\upsilon-1$

(A-4) For $q = 0,1,...,Q$

(A-5) $y_k(p) = y_k\left(kT + \hat{\tau}_k(p)\right)$ for $\forall p$

(A-6) $\rho_k(p,q) = \left|y_k(p) - \hat{r}(p,q)\right|^2$ for $\forall p$

*(A-7) If $k = i$ then [i is the preamble position]

For $p = 0,1,...,Q-1$

If $\hat{b}(p,q) \neq f_i$ then

$\rho_k(p,q) = \Delta$ [Δ is a large number]

End

(A-8) $\pi_{k+1}(q) = \arg\min_p \left\{\mathbf{\Phi}_k(p) + \rho_k(p,q)\right\}$

(A-9) $\mathbf{\Phi}_{k+1}(q) = \mathbf{\Phi}_k\left(\pi_{k+1}(q)\right) + \rho_k\left(\pi_{k+1}(q), q\right)$

(A-10) $\mathbf{S}_{k+1}(q) = \left[\mathbf{S}_k\left(\pi_{k+1}(q)\right) \middle| \pi_{k+1}(q)\right]$

(A-11) $\hat{\varepsilon} = \frac{6T}{40}\left\{y_k\left(\pi_{k+1}(q)\right)\hat{r}\left(\pi_k\left(\pi_{k+1}(q)\right), \pi_{k+1}(q)\right)\right.$

$\left. - y_{k-1}\left(\pi_k\left(\pi_{k+1}(q)\right)\right)\hat{r}\left(\pi_{k+1}(q), q\right)\right\}$

(A-12) $\hat{\tau}_{k+1}(q) = \hat{\tau}_k\left(\pi_{k+1}(q)\right) + \alpha\hat{\varepsilon}$

(A-13) $\Delta_{k+1}(q) = \max_p \left\{\mathbf{\Phi}_k(p) + \rho_k(p,q)\right\} - \mathbf{\Phi}_{k+1}(p)$

(A-14) Initialize $\hat{L}_k(q) = +\infty$ [Soft decision update]

(A-15) For $j = k-\nu,...,k-\delta$

Compare the two paths merging

in state q (i.e., $\mathbf{\Psi}_{k+1} = q$)

If $\hat{a}_j^{(1)}\left(\mathbf{\Psi}_{j+1}\right) \neq \hat{a}_j^{(2)}\left(\mathbf{\Psi}_{j+1}\right)$ then

Update $\hat{L}_j\left(\mathbf{\Psi}_{j+1}\right) = \min\left(\hat{L}_j\left(\mathbf{\Psi}_{j+1}\right), \Delta_{k+1}(q)\right)$

End

(A-16) End

(A-17) if $k \geq \delta$ then

Output the soft decision according to $\lambda_k^p = \hat{a}_{k-\delta}\,\hat{L}_{k-\delta}$, which can be extracted from the survivor path that minimizes $\mathbf{\Phi}_{k+1}$

(A-18) End

(A-19) End

Figure 5.5 The modified PSP-SOVA algorithm, where (A-7) is an additional step beyond the PSP-SOVA.

Denote $f_k \in \{\pm 1\}$ as a $C/2$ -bit preamble that is embedded uniformly within the user data stream at the *i*-th position. In other words, each *m*-bit preamble is inserted at every *i*-th data bit, where $i = \lfloor 8190m/C \rfloor$ is the lowest integer close to $8190m/C$. Following the modified PSP-SOVA algorithm, during the branch metric calculation at each *k*-th stage, we check the condition in (A-7) to adjust the branch metrics $\rho_k(p,q)$.

Specifically, at the *k*-th stage, if $\hat{b}_k(p,q) \neq f_i$, where $\hat{b}_k(p,q) \in \{\pm 1\}$ is the data bit that corresponds to the state transition from state p to state q, and f_i is the preamble bit at the *i*-th position, we then set $\rho_k(p,q) = \Delta$, where Δ is a large number to guarantee that the modified SOVA will not choose this branch as part of a survivor path.

5.5 Complexity Comparison

To measure the complexity of iterative timing recovery schemes, we consider the total number of additions and multiplications as a criterion used in each scheme. For other mathematical functions, such as $\log(x)$, $\exp(x)$ and etc., we assume they can be implemented by the lookup tables, and that we ignore their complexity. Note that we attempt to fairly count the number of operations (both addition and multiplication) for each scheme so that the memory requirement is minimized.

The complexity of each component is given in Table 5.1 for modified PSP-SOVA and Table 5.2 for PSP-BCJR, where N_{sinc} is the total number of ideal sinc interpolation filter taps used to sample the analog signal and to refine the samples at each iteration [62] based on a set of the previous samples and their corresponding sampling phase offsets, the value $Q = 2^{\nu}$ is the number of trellis states [1], δ is the decoding depth used to output the soft decision in SOVA [21], k is a parameter of a LDPC code [63], N_{in} is the internal iterations used in the LDPC decoder, and R is a code rate.

Based on Table 5.1 and 5.2, we can summarize the complexity of each iterative timing recovery schemes as given in Table 5.3, where we employ $N_{sinc} = 21$, $\nu = 2$, $\delta = 5(\nu+1)$ [62], $N_{in} = 5$, and N is the number of turbo iterations. It should be noted that a multiplication has much more complexity than an addition in terms of circuit implementation. Thus, we consider only the number of multiplications when comparing the performance of different iterative timing recovery schemes. Because the modified PSP-SOVA has few extra computation steps beyond the PSP-SOVA, we can then assume that both schemes have the same complexity.

Table 5.1 The total number of operations (per bit) of each function used in the MPS-ITR scheme.

Module	Number of iterations (**per bit**)	
	Addition	Multiplication
Ideal sinc interpolation filter	$(4N_{sinc}-1)Q$	$(N_{sinc}-1)Q$
1st-order PLL	Q	Q
SOVA	$7Q+\frac{\delta^2+9\delta+9}{2}+1$	$6Q+1$
Modified PSP-SOVA	$(7+4N_{sinc})Q+\frac{\delta^2+9\delta+9}{2}+1$	$(7+N_{sinc})Q+1$
LDPC decoder	$\left(1+(k-1)(1-R)\right)N_{in}+1$	$(1-R)N_{in}$

Table 5.2 The total number of operations (per bit) of each function used in the Full PS-ITR scheme.

Module	Number of iterations (**per bit**)	
	Addition	Multiplication
Ideal sinc interpolation filter	$2(4N_{sinc}-1)Q$	$2(N_{sinc}-1)Q$
1st-order PLL	$2Q$	$2Q$
BCJR	$12Q-2$	$20Q+1$
PSP-BCJR	$(10+8N_{sinc})Q-2$	$(22+2N_{sinc})Q+1$
LDPC decoder	$\left(1+(k-1)(1-R)\right)N_{in}+1$	$(1-R)N_{in}$

Table 5.3 Complexity (per bit) of different iterative timing recovery schemes, where N is the number of turbo iterations.

System	Number of iterations (**per bit**)	
	Addition	Multiplication
Conventional receiver	$27+223.94N$	$9+25.56N$
Full PS-ITR sacheme	$730.44N$	$257.56N$
PS-ITR scheme	$569.94N$	$113.56N$
MPS-ITR scheme	$569.94N$	$113.56N$

5.6 The MPS-ITR in a Realistic Channel System

In this Section, the MPS-ITR is investigated in a perpendicular magnetic recording channel (realistic channel model) as shown in Figure 5.6 instead of the perfectly equalized channel model (ideal channel model) as shown in Figure 5.1. The readback signal can then be written as

$$p(t) = \sum_k c_k g(t - kT - \tau_k) + n(t), \tag{5.5}$$

where $g(t)$ is the transition response of perpendicular magnetic recording is given in (2.2) and $n(t)$ is an additive white Gaussian noise with a two-sided power spectral density $N_0/2$. The readback signal is filtered by a seventh-order Butterworth low pass filter. As for all of the others parameter were defined same as described in Section 5.2.

We consider a perpendicular recording channel at ND = 2 with $\sigma_w / T = 0.6\%$. The SNR is defined as given in (3.23). During acquisition mode, the PLL gain parameter, α, for the conventional receiver, PS-ITR, and MPS-ITR are designed to recover the phase change within 256, 256, and 128 symbols, respectively. However, we consider the case where the α designed to recover the phase change within 256 symbols is used for all schemes during tracking mode. To account for a coded system, we define a user density, D_u, as D_u = ND × code rate. We employ a 11-tap equalizer designed (described in Section 3.31) at the SNR required to achieve BER=10^{-5} to equalize the readback signal to the PR2 target. The performance of MPS-ITR will be compared with another iterative timing recovery scheme and conventional timing recovery in the next Section.

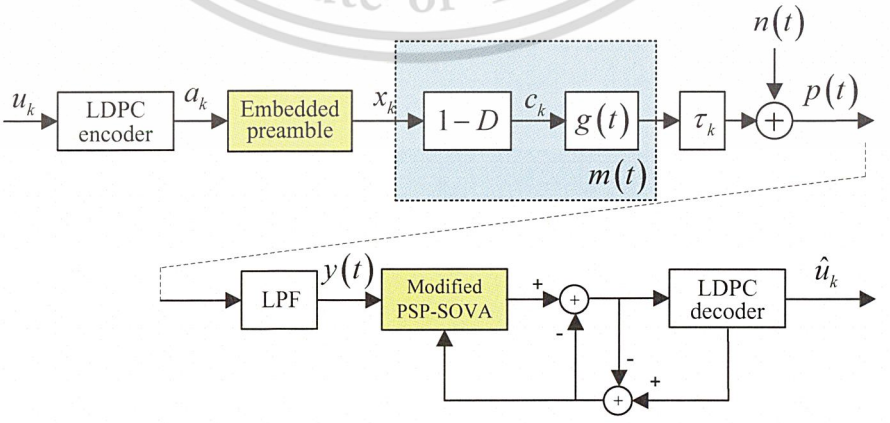


Figure 5.6 A realistic channel model with the modified per-survivor iterative timing recovery (MPS-ITR) scheme.

5.7 Simulation Results and Discussion

5.7.1 MPS-ITR in an Ideal Channel Model

The MPS-ITR scheme exchanges information between the modified PSP-SOVA and the SISO decoder as illustrated in Figure 5.1. Consider a rate-8/9 system in which a block of 3640 message bits is encoded by a regular (3, 27) LDPC code [63], resulting in a coded block length of 4095 bits. The parity-check matrix has 3 ones in each column and 27 ones in each row.

The SISO equalizer is implemented based on SOVA, whereas the SISO decoder is implemented based on the message-passing algorithm with 5 internal iterations. During an acquisition mode, the PLL gain parameter, α, for the conventional receiver, PS-ITR, Full PS-ITR, MPS-ITR and the trained PLL receiver are designed to recover the phase change within 256, 256, 256, 128, and 256 symbols, respectively, based on a linearized model of PLL [3], assuming that the S-curve slope is one at the origin, and there is no noise in the system. However, we consider the case where the α designed to recover the phase change within 256 symbols is used for all schemes during tracking mode.

Figure 5.7 (a) compares the BER performance of different iterative timing recovery schemes at the 5-th iteration for the system with a moderate random walk parameter $\sigma_w / T = 0.6\%$, which implies a low probability of occurrence of cycle slips, as a function of per-bit SNRs, E_b / N_0's. Note that the number inside the parenthesis in Figure 5.7 indicates the total number of iterations used to generate each curve. The curve labeled as "Perfect timing" represents the conventional receiver that uses $\hat{\tau}_k = \tau_k$ to sample $y(t)$. Also, the curve labeled as "Trained PLL" is the conventional timing recovery, whose PLL has access to all correct decisions, thus serving as a lower bound for all timing recovery schemes based on a PLL.

Apparently, the MPS-ITR performs better than the conventional receiver, and yields about 0.15 dB gain at BER = 10^{-4} over the PS-ITR. However, the MPS-ITR is slightly better than the Full PS-ITR. In addition, the MPS-ITR performs close to the trained PLL receiver and is only a 0.1 dB away from the system with perfect timing at BER = 10^{-4}. We also compare the performance of different schemes in Figure 5.7 (b) at a severe random walk parameter $\sigma_w / T = 1.2\%$, which implies a high probability of occurrence of a cycle slip. It is obvious that the MPS-ITR provides a large performance gain over both the PS-ITR, the Full PS-ITR, and outperforms the conventional receiver. Again, the MPS-ITR is only 0.25 dB and 0.45 dB away from the trained PLL receiver and the system with perfect timing at BER = 10^{-4}, respectively.

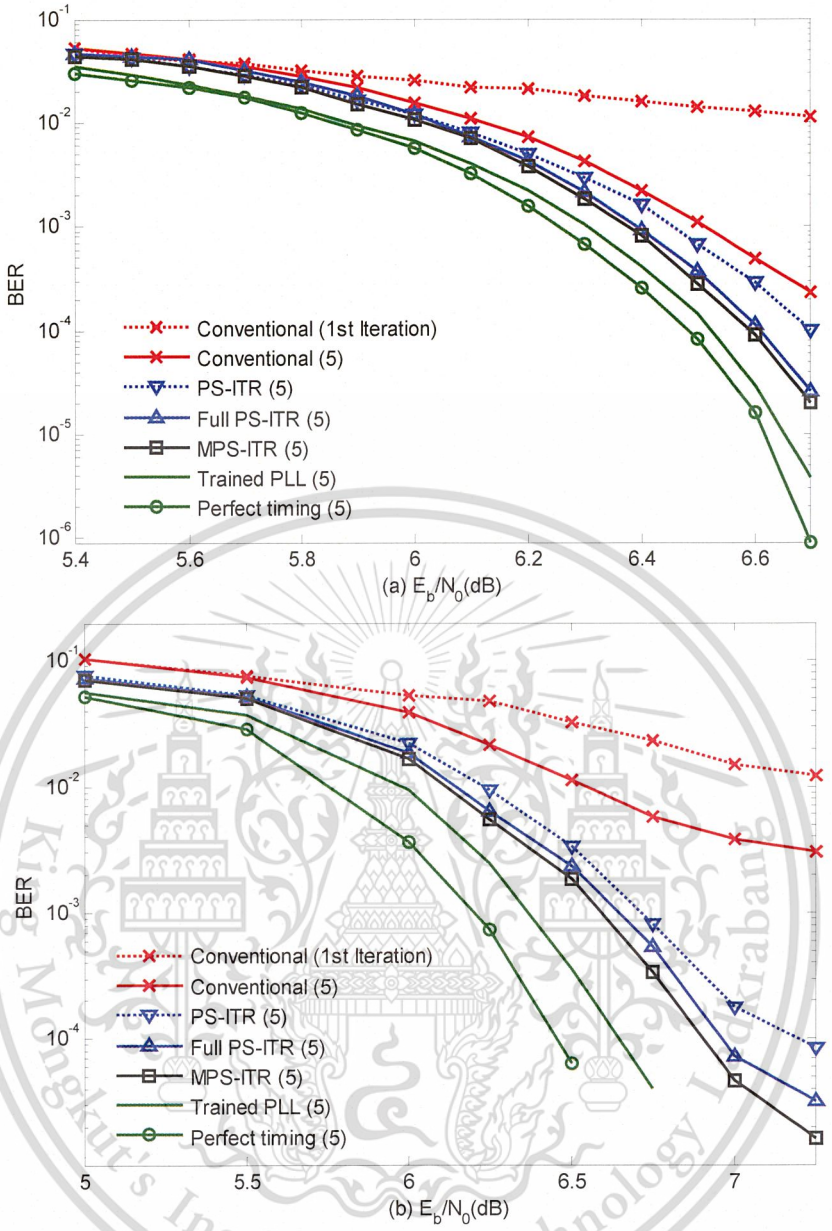


Figure 5.7 Performance comparison of the different iterative timing recovery schemes for (a) $\sigma_w / T = 0.6\%$ and (b) $\sigma_w / T = 1.2\%$.

The reason that we choose the number of iterations to be 5 iterations in simulations is because the performance of the MPS-ITR begins to saturate after 5 iterations, which can be confirmed by plotting the E_b/N_0 required to achieve BER = 10^{-4} versus the number of iterations in Figure 5.8.

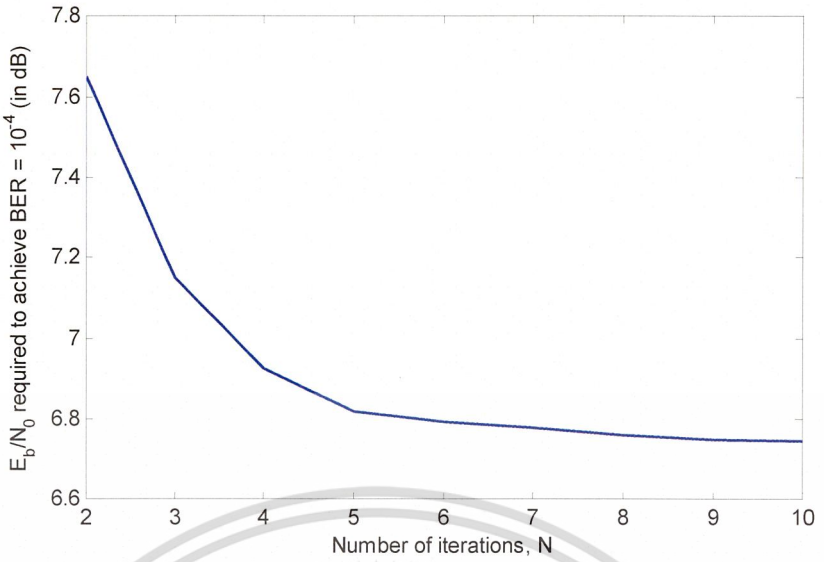


Figure 5.8 Convergence of the MPS-ITR at $\sigma_w / T = 1.2\%$

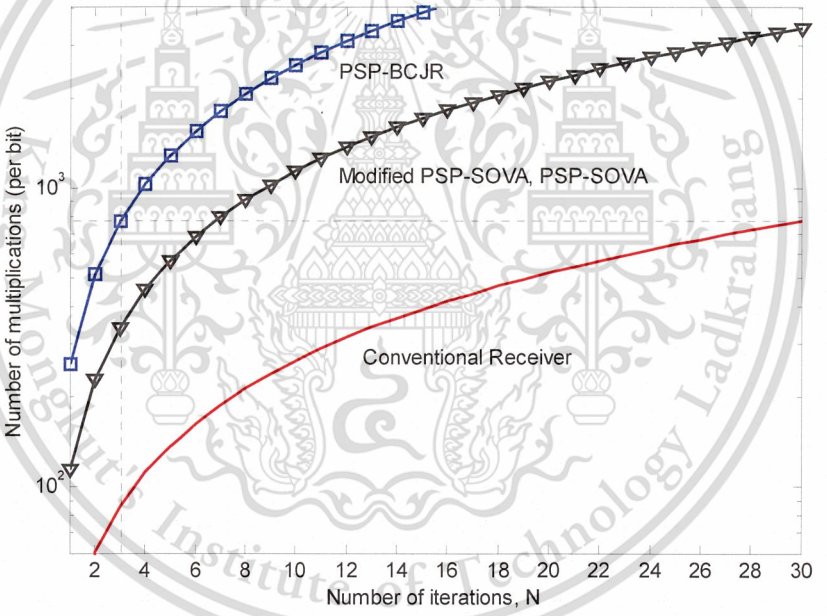


Figure 5.9 Complexity comparison (based on a PR2 channel).

It is worth comparing their performance given the same complexity. Therefore, we assume that the current technology can support the total number of multiplications equal to 3 iterations of the Full PS-ITR scheme, which is approximately equal to 7, and 30 iterations of the MPS-ITR, and the conventional receiver, respectively, as shown in Figure 5.9.

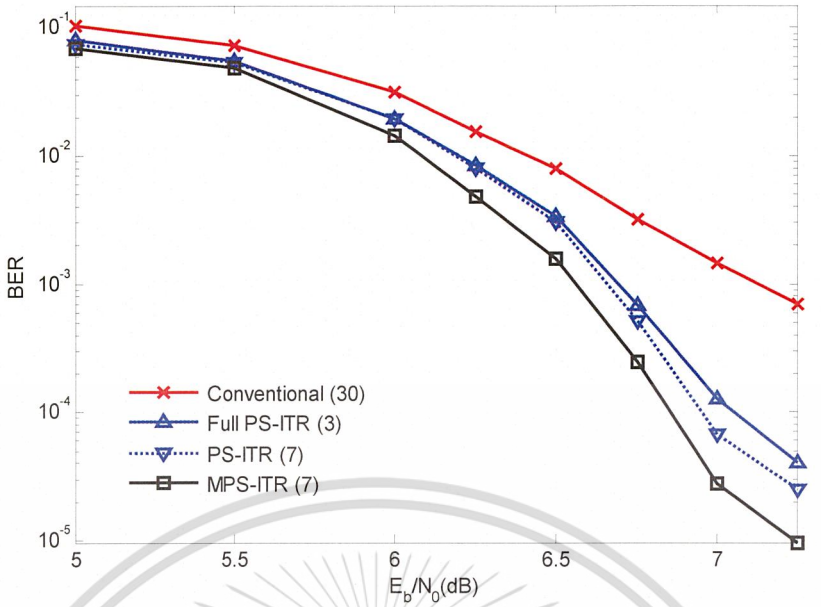


Figure 5.10 Performance comparison with same complexity at $\sigma_w / T = 1.2\%$.

Figure 5.10 compares the BER performance of different iterative timing recovery schemes when they have same complexity at $\sigma_w / T = 1.2\%$. It is apparent that the MPS-ITR performs better than other schemes and PS-ITR is better than the Full PS-ITR, especially at high SNR. Consequently, it is worth employing the MPS-ITR in the system when the complexity is limited to a low-to-moderate amount.

One reason that MPS-ITR performs better than both the PS-ITR and the Full PS-ITR is because the split preamble embedded uniformly within the data stream helps reduce the occurrence of cycle slips. This can be verified by plotting the percentage of occurrence of a cycle slip at $E_b / N_0 = 5$ dB as shown in Figure 5.11. Clearly, the MPS-ITR experiences fewer cycle slips than the PS-ITR, the Full PS-ITR, and the conventional receiver, especially when the timing jitter is severe. Furthermore, the one-bit split-preamble arrangement (i.e., $m = 1$) provides the best performance.

Figure 5.12 shows the timing waveforms for a sample packet at $E_b / N_0 = 6$ dB and $\sigma_w / T = 1.2\%$, where the gray curve represents the actual τ sequence. It is clear that at the first iteration, the PS-ITR, the Full PS-ITR, and the MPS-ITR cannot keep track of the rapid change in the actual τ sequence after about 1,000 symbols. Nonetheless, the MPS-ITR can correct a cycle slip within two iterations, while both the PS-ITR and the Full PS-ITR require four iterations to correct it. This implies that the MPS-ITR corrects a cycle slip faster than both the PS-ITR and the Full PS-ITR. In other words, the MPS-ITR requires fewer turbo iterations than other scheme to yields good performance.

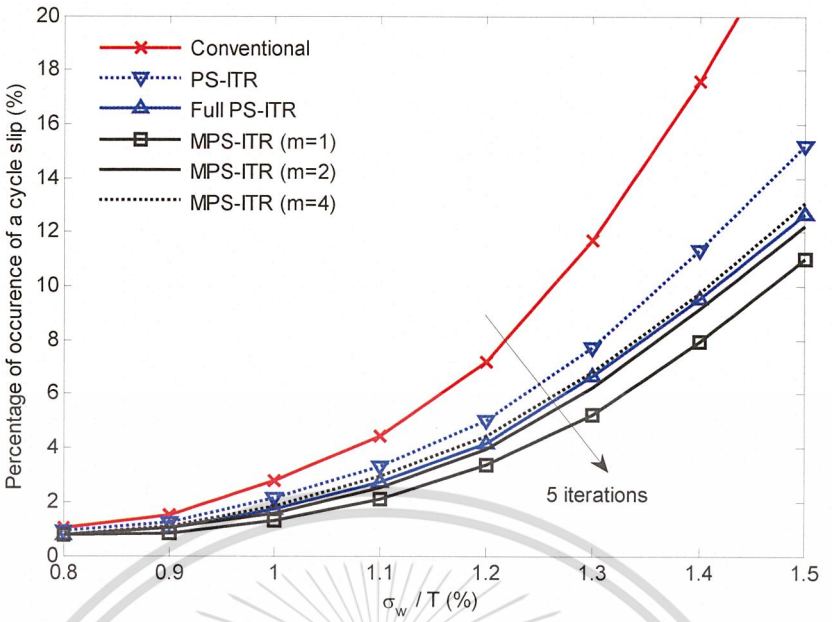


Figure 5.11 Percentage of occurrence of a cycle slip at $E_b / N_0 = 5$ dB.

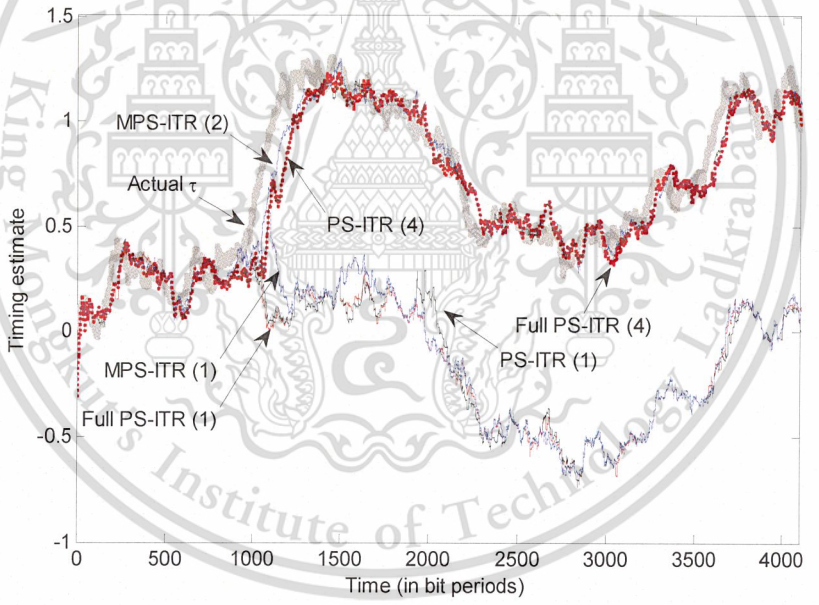


Figure 5.12 Cycle slip correction at $E_b / N_0 = 6$ dB and $\sigma_w / T = 1.2\%$.

We also have investigated and compared the performance of different timing recovery schemes under the assumption that the initial timing offset is perfectly obtained (i.e., perfect acquisition), at $\sigma_w / T = 1.2\%$, and found that the MPS-ITR still performs better than other schemes as now shown in Figure 5.13. Therefore, it can be implied in this case that the performance gain comes from the structure of timing recovery scheme itself, not from perfect acquisition.

In a practical receiver, a preamble is used not only for timing recovery, but also for automatic gain control (AGC) [2]. This AGC is utilized to adjust the amplitude of the readback signal such that its amplitude remains constant throughout the data packet.

We assume that the readback signal is modeled by the linear superposition of the PR2 pulses, and there is no variation in signal amplitude. Hence, we did not include an AGC in our channel model. To check whether the proposed shortened preamble affects the AGC, we have investigated the behavior of the AGC gain, g_i, in the PR2 channel at SNR = 10 dB with 128-bit preamble (same channel as used in our paper, but without error-correction codes) with the results as shown in Figure 5.14. We use an initial AGC gain g_0 = 3.33[2], and the AGC update step size $\beta = 1.25\text{x}10^{-2}, 8.75\text{x}10^{-3}$, [19]. The gain update follows the approach in [2].

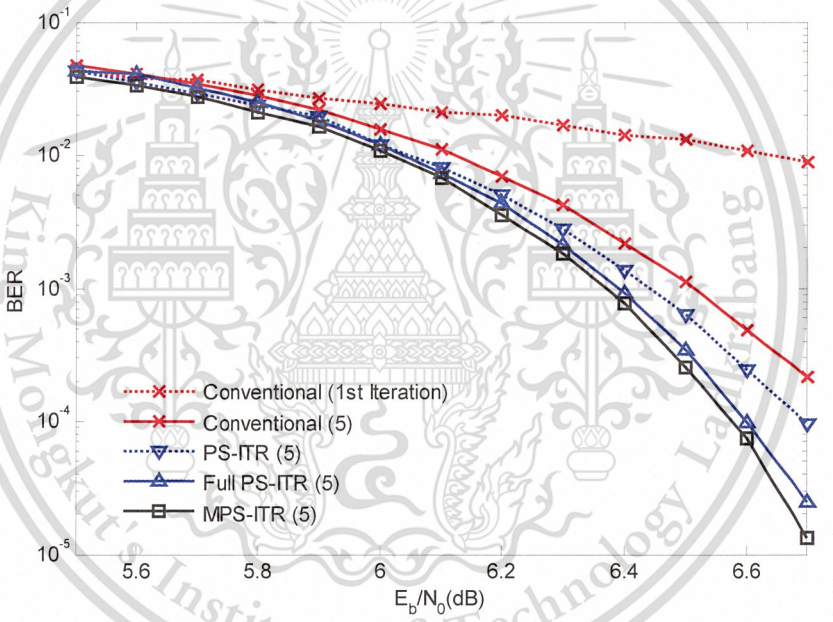


Figure 5.13 Performance comparison of different iterative timing recovery scheme with perfect acquisition at $\sigma_w / T = 0.6\%$.

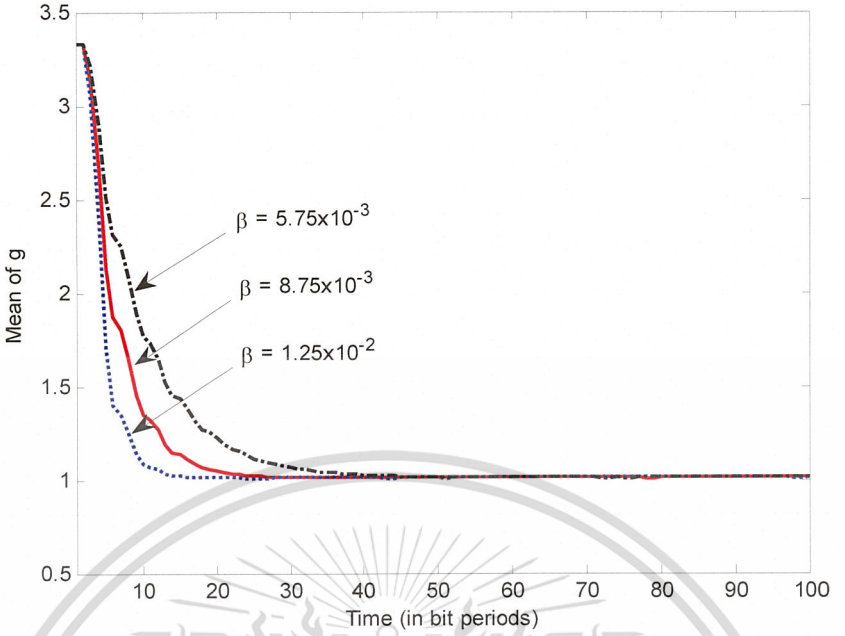


Figure 5.14 The averaged AGC gain based on 2000 data packets during acquisition.

In this channel, the readback signal is modeled by the linear superposition of the PR2 pulses, and there is no variation in signal amplitude. Therefore, we can see that the AGC gain [43] converges to the value "1" very fast, depending on the AGC update step size β. This means that for our channel, the AGC can perform well within 50 bit periods. Because our proposed method employs 128-bit preamble during acquisition, there should not be any problem with the functionality of AGC.

From the simulation result on conventional timing recovery as shown in Figure 5.14, based on the same algorithm as in [2], but for PR2 signals, we found that the number of required preamble bits for convergence of the gain coefficient is less than 50 bits, and well below 128 preamble bits proposed in this work. Therefore, a shorter preamble in this case does not affect the overall performance of timing recovery system.

5.7.2 MPS-ITR in Realistic Channel Model

First, we compare the BER performance of different iterative timing recovery schemes at the 5-th iteration for the system with a moderate random walk parameter $\sigma_w / T = 0.6\%$. Note that the number inside the parenthesis in Figure 5.15 indicates the total number of iterations used to generate each curve. The curve labeled as "Perfect timing" represents the conventional receiver that uses $\tau_k = \hat{\tau}_k$ to sample $y(t)$.

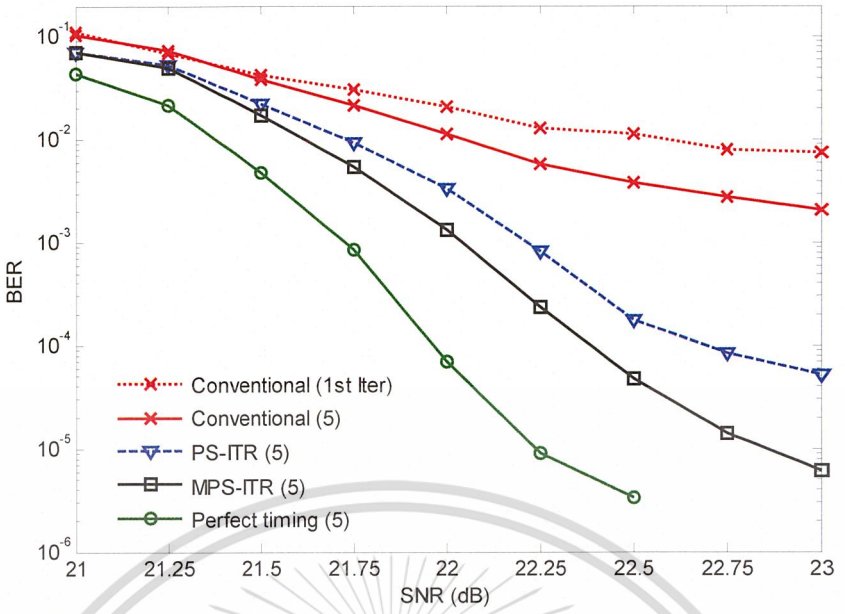


Figure 5.15 Performance comparison of different iterative timing recovery schemes in the realistic channel model.

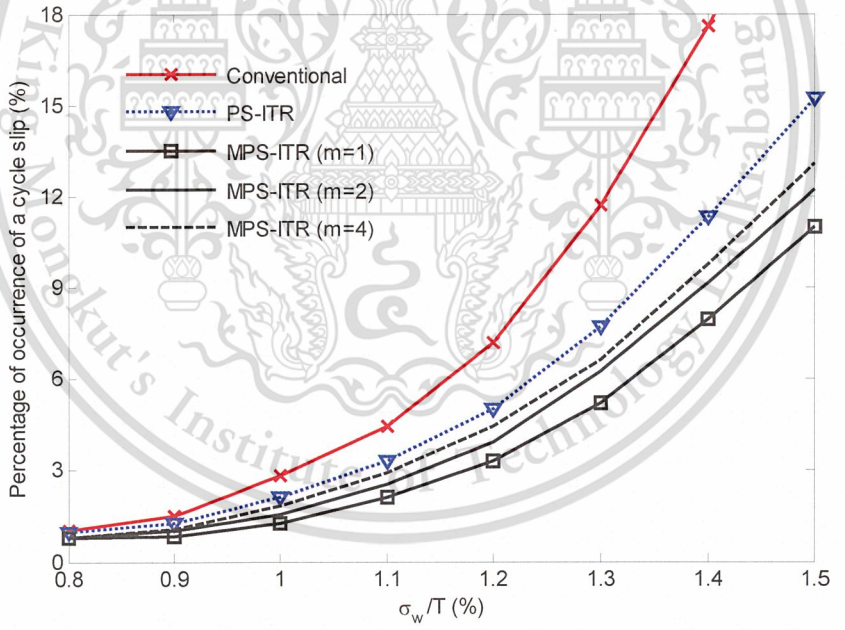


Figure 5.16 Percentage of occurrence of a cycle slip at SNR = 21.5 dB.

Apparently, the MPS-ITR performs better than the conventional receiver, and yields about 0.3 dB gain at BER = 10^{-4} over the PS-ITR. In addition, the MPS-ITR performs only a 0.4 dB away from the system with perfect timing at BER = 10^{-4}.

Because the split preamble embedded uniformly within the data stream still helps reduce the occurrence of cycle slips, even though, it is employed in realistic channel, that is the reason why the MPS-ITR performs better than the PS-ITR. Which can be verified by plotting the percentage of occurrence of a cycle slip at SNR = 21.5 dB as shown in Figure 5.16. Clearly, the MPS-ITR experiences a fewer number of cycle slips than both the PS-ITR and the conventional receiver, especially when the timing jitter is large. Furthermore, the one-bit split-preamble arrangement (i.e., $m = 1$) still provides the best performance in realistic channel model.

5.8 Summary

In this Chapter, we propose the MPS-ITR scheme, which iteratively exchanges the soft information between the modified PSP-SOVA and the decoder. This modified PSP-SOVA uses the new split-preamble strategy, which divides a preamble into two parts, each of which is used to adjust the branch metric calculation in PSP-SOVA so as to guarantee that the survivor path occurs in a correct direction. In addition, we also propose to use the MPS-ITR in a realistic channel model.

Simulation results have shown the MPS-ITR performs better than the full PS-ITR, the PS-ITR, and the conventional receiver, especially when the timing jitter is severe. Then, we also show the MPS-ITR has a lower complexity than the full PS-ITR. In addition, we observed that the MPS-ITR can reduce the occurrence of cycle slips and can also automatically correct a cycle slip much more efficiently than both the PS-ITR and the full PS-ITR. In other words, the MPS-ITR requires a fewer number of turbo iterations than other schemes to correct a cycle slip.

Finally, we show that the MPS-ITR can be employed in the realistic perpendicular magnetic recording channel and also yields the better performance than another.

CHAPTER 6

REDUCED-COMPLEXITY MODIFIED PER-SURVIVOR ITERATIVE TIMING RECOVERY FOR CODED PARTIAL RESPONSE CHANNELS

A modified per-survivor iterative timing recovery (MPS-ITR), which jointly performs timing recovery, equalization, and error-correction decoding, has been described in Chapter 5, which it is proposed to deal with the problem of timing recovery operating at low signal-to-noise ratios. In practice, this scheme exploits a split-preamble strategy in conjunction with a per-survivor soft-output Viterbi algorithm equalizer to make it more robust against severe timing jitters or cycle slips. Although the MPS-ITR outperforms other iterative timing recovery schemes [57], it still has very high complexity. In this chapter, we propose a reduced-complexity MPS-ITR scheme (denoted as MPS-ITR-M) to make it more implementable in real-life applications. This is achieved by applying the M-algorithm [64] to the MPS-ITR. Numerical results indicate that when all schemes have same complexity, the MPS-ITR-M yields better performance than other schemes.

6.1 Introduction

Timing recovery is the process of synchronizing the sampler with the received analog signal. Sampling at the right times is critical to achieving good overall performance. The last decade has seen the development of iteratively decodable error-control codes of unprecedented power, whose large coding gains enable reliable communication at very low SNR. A by-product of this trend is that timing recovery must be performed at an SNR lower than ever before. Conventional timing recovery ignores the presence of error-control coding and is thus doomed to fail when the SNR is low enough.

To improve the performance of the conventional receiver, Kovintavewat *et al.* [20] proposed a PS-ITR scheme, which *jointly* performs timing recovery, equalization, and error-correction decoding. It is realized by first applying the PSP technique [18], a technique of jointly estimating a data sequence and unknown parameters, to the SOVA [21], resulting in a per-survivor SOVA equalizer, denoted as "PSP-SOVA" [20]. Then, PSP-SOVA iteratively exchanges soft information with a SISO decoder. As investigated in [20], the PS-ITR outperforms the

conventional receiver because it can automatically correct a cycle slip [3] after a few number of turbo iterations.

To make the PS-ITR more robust against severe timing jitters, a MPS-ITR scheme has been proposed in [57], which iteratively exchanges soft information between the *modified* PSP-SOVA and the decoder. This modified PSP-SOVA uses the new split-preamble strategy, which divides a preamble into two parts, each of which is used to adjust the branch metric calculation in PSP-SOVA so as to guarantee that the survivor path occurs in a correct direction. It has been shown [57] that the MPS-ITR performs better than the PS-ITR, and both outperforms the conventional receiver, especially when the timing jitter is severe.

The M-algorithm was first introduced by Simmons and Mohan [64]. It has been employed in many applications, including source coding [64] and channel decoding [65]. Nevertheless, Iki *et al.* [66] proposed the application of M-algorithm and stack algorithm to the trellis shaping with peak-to-average power ratio reduction for single-carrier signal. Additionally, the M-algorithm has also been used in data storage systems. For instance, a simplified noise-predictive partial response maximum likelihood system in conjunction with the M-algorithm was proposed in [67] for dual-layered perpendicular magnetic recording channels, and a low-complexity non-iterative detector for magnetic and optical multitrack high-density data storage was proposed in [68], whose detector is based on the M-algorithm.

Because the MPS-ITR scheme has very high complexity, we therefore apply the M-algorithm [64] to the MPS-ITR, resulting in the *reduced-complexity* MPS-ITR scheme (denoted as MPS-ITR-M), so as to reduce its complexity and to make it more implementable in real-life applications. Additionally, we consider only a coded partial response channel in this thesis because this channel is widely used in magnetic recording systems [15, 69]. Thus, it will be shown later that at low-to-moderate complexity, the MPS-ITR-M performs better than the MPS-ITR and the PS-ITR.

Note that we also consider the system model of the coded partial-response channel in Figure 6.1, which was explained in Section 5.2. The rest of this chapter is organized as follows. Section 6.2 briefly describes how the MPS-ITR-M works. Complexity comparison is provided in Section 6.3. Simulation results and discussion are given in Section 6.4. Finally, Section 6.5 summarizes this chapter.

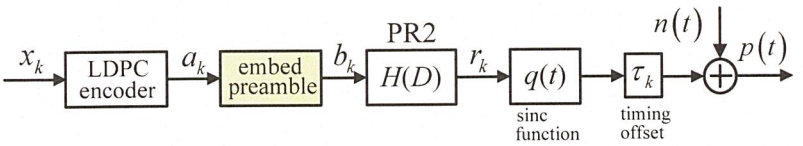


Fig. 6.1 Data encoding with a PR2 channel model.

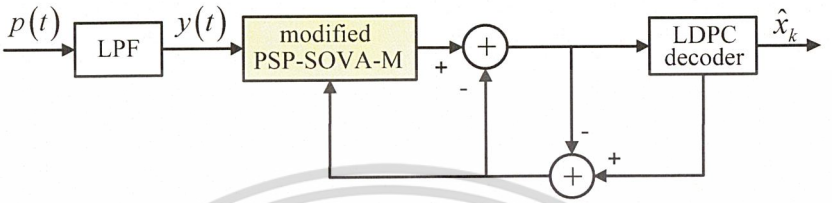


Fig. 6.2 Reduced-complexity modified per-survivor iterative timing recovery (MPS-ITR-M).

6.2 The Modified PSP-SOVA-M algorithm

The proposed scheme, i.e., MPS-ITR-M, will iteratively exchange soft information between a modified PSP-SOVA-M module and the SISO decoder as shown in Figure 6.2. This modified PSP-SOVA-M is obtained by applying the M-algorithm [64] to the modified PSP-SOVA proposed as described in Chapter 5, which can be explained how it performs as follows.

The M-algorithm [64] was originally proposed to reduce complexity of the Viterbi algorithm, which can be described how it works as follows. Figure 6.3 shows the trellis diagram of the PR2 channel, which has 4 states. At each time instant, the M-algorithm first finds the minimum path metric leading to each trellis state. Hence, it retains only the M paths (M must be less than the total number of states in one stage of the trellis) with the lowest path metrics among all survivor paths. For example, in Fig. 6.3, if we assume that $M = 3$ and the state 0 has maximum path metric at time k. As a result, *only* three states (i.e., state 1, 2, and 3) will be used in branch metric calculation at the k-th stage.

Because the modified PSP-SOVA is developed based on the Viterbi algorithm, its complexity grows exponentially with the channel memory [2]. Therefore, to reduce its complexity, we apply the M-algorithm [64] to the modified PSP-SOVA, denoted as modified PSP-SOVA-M. Figure 6.4 shows the modified PSP-SOVA-M algorithm, where a constant $6T/40$ in (A-11) is only for the PR2 channel, which can be included in the PLL gain parameter.

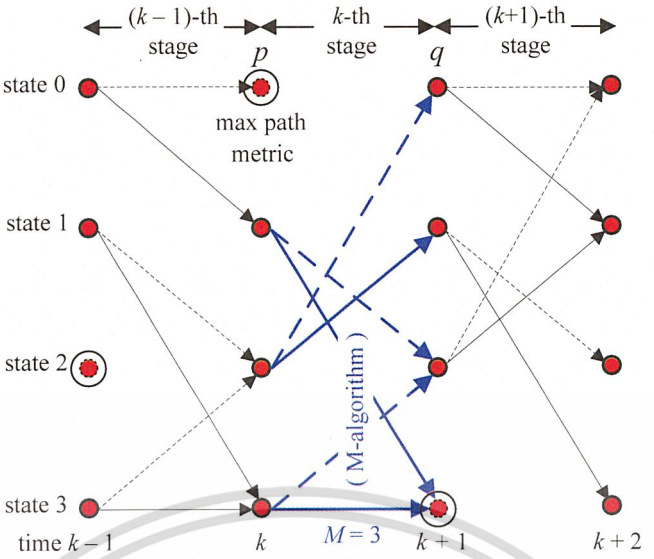


Figure 6.3 The PR2 trellis structure illustrating how M-algorithm performs.

It should be noted that the modified PSP-SOVA-M works in a *same* manner as the modified PSP-SOVA does, except that the modified PSP-SOVA-M has an *extra* step according to the M-algorithm, which can be briefly explained as follows.

Consider the trellis diagram in Figure 6.3 at the k-th stage, where we denote $\mathbf{M}$ as a set of all states (e.g., $\mathbf{M} = \{1, 2, 3\}$) that still remain at time k according to the M-algorithm. To reduce the number of states, a state p will be chosen from $\mathbf{M}$, i.e., $p \in \mathbf{M}$. In other words, only the branches emanating from $\mathbf{M}$ will be used in branch metric calculation at the k-th stage.

It should be noted that the M parameter must be carefully chosen according to the channel model used. For the PR2 channel with 4 trellis states, we found that $M = 3$ is a suitable value for employing in the MPS-ITR. Additionally, the results reported in this paper are still valid for any PR channel, given that the M parameter is chosen suitably.

(A-1) Initialize $\mathbf{\Phi}_0(p) = 0$ for $\forall p$

(A-2) Initialize $\hat{\tau}_0(p) = 0$ for $\forall p$

(A-3) For $k = 0,1,\ldots, L+\upsilon-1$

(A-4) For $q = 0,1,\ldots, Q-1$

(A-5) $y_k(p) = y_k\left(kT + \hat{\tau}_k(p)\right)$ for $p \in \mathbf{M}$

(A-6) $\rho_k(p,q) = \left|y_k(p) - \hat{r}(p,q)\right|^2$ for $p \in \mathbf{M}$

(A-7) If $k = i$ then [i is the preamble position]

For $p \in \mathbf{M}$

If $\hat{b}(p,q) \neq f_i$ then

$\rho_k(p,q) = \Delta$ [Δ is a large number]

End

(A-8) $\pi_{k+1}(q) = \arg\min_{p\in\mathbf{M}}\left\{\mathbf{\Phi}_k(p) + \rho_k(p,q)\right\}$

(A-9) $\mathbf{\Phi}_{k+1}(q) = \mathbf{\Phi}_k\left(\pi_{k+1}(q)\right) + \rho_k\left(\pi_{k+1}(q), q\right)$

(A-10) $\mathbf{S}_{k+1}(q) = \left[\mathbf{S}_k\left(\pi_{k+1}(q)\right)\middle|\pi_{k+1}(q)\right]$

(A-11) $\hat{\varepsilon} = \frac{6T}{40}\Big\{y_k\left(\pi_{k+1}(q)\right)\hat{r}\left(\pi_k\left(\pi_{k+1}(q)\right), \pi_{k+1}(q)\right)$

$- y_{k-1}\left(\pi_k\left(\pi_{k+1}(q)\right)\right)\hat{r}\left(\pi_{k+1}(q), q\right)\Big\}$

(A-12) $\hat{\tau}_{k+1}(q) = \hat{\tau}_k\left(\pi_{k+1}(q)\right) + \alpha\hat{\varepsilon}$

(A-13) $\Delta_{k+1}(q) = \max_{p\in\mathbf{M}}\left\{\mathbf{\Phi}_k(p) + \rho_k(p,q)\right\} - \mathbf{\Phi}_{k+1}(p)$

(A-14) Initialize $\hat{L}_k(q) = +\infty$ [Soft decision update]

(A-15) For $j = k-\nu,\ldots,k-\delta$

Compare the two paths merging

in state q (i.e., $\mathbf{\Psi}_{k+1} = q$)

If $\hat{a}_j^{(1)}\left(\mathbf{\Psi}_{j+1}\right) \neq \hat{a}_j^{(2)}\left(\mathbf{\Psi}_{j+1}\right)$ then

Update $\hat{L}_j\left(\mathbf{\Psi}_{j+1}\right) = \min\left(\hat{L}_j\left(\mathbf{\Psi}_{j+1}\right), \Delta_{k+1}(q)\right)$

End

(A-16) If $k \geq \delta$ then

Output the soft decision according to $\lambda_k^p = \hat{a}_{k-\delta}\,\hat{L}_{k-\delta}$,

which can be extracted from the survivor path

that minimizes $\mathbf{\Phi}_{k+1}$

End

End

Figure 6.4 The modified PSP-SOVA-M algorithm.

Table 6.1 Complexity (per bit) of different iterative timing recovery schemes, where N is the number of turbo iterations.

System	Number of iterations (**per bit**)	
	Addition	Multiplication
Conventional receiver	27+223.94N	9+25.56N
Per-survivor iterative: - with PSP-SOVA	569.94N	113.56N
Modified per-survivor iterative: - with modified PSP-SOVA	569.94N	113.56N
- with modified PSP-SOVA-M1	478.94N	85.56N
- with modified PSP-SOVA-M2	387.94N	57.56N

6.3 Complexity Comparisons

In this thesis, we consider the proposed scheme with $M = 3$ (referred to as MPS-ITR-M1) and $M = 2$ (referred to as MPS-ITR-M2). Based on Table 5.1, we can summarize the complexity of each iterative timing recovery schemes as given in Table 6.1, where we employ $N_{\text{sinc}} = 21$, $\nu = 2$, $\delta = 5(\nu + 1)$ [63], and $N_{in} = 5$, and N is the number of turbo iterations.

It should be pointed out that multiplication has much more complexity than addition in terms of circuit implementation. Thus, in this chapter, we also consider only the number of multiplications when comparing the performance of different iterative timing recovery schemes.

6.4 Simulation Results and Discussion

Consider a rate-8/9 system in which a block of 3640 message bits is encoded by a regular (3, 27) LDPC code [64], resulting in a coded block length of 4095 bits. The parity-check matrix has 3 ones in each column and 27 ones in each row. The SISO equalizer is implemented based on SOVA, whereas the SISO decoder is implemented based on the message-passing algorithm with 5 internal iterations ($N_{in} = 5$). Note that one data sector consists of 256-bit preamble and 4095 coded bits. Each bit-error rate (BER) was computed by using as many data sectors as needed to collect 1000 error bits at the 5-th turbo iteration.

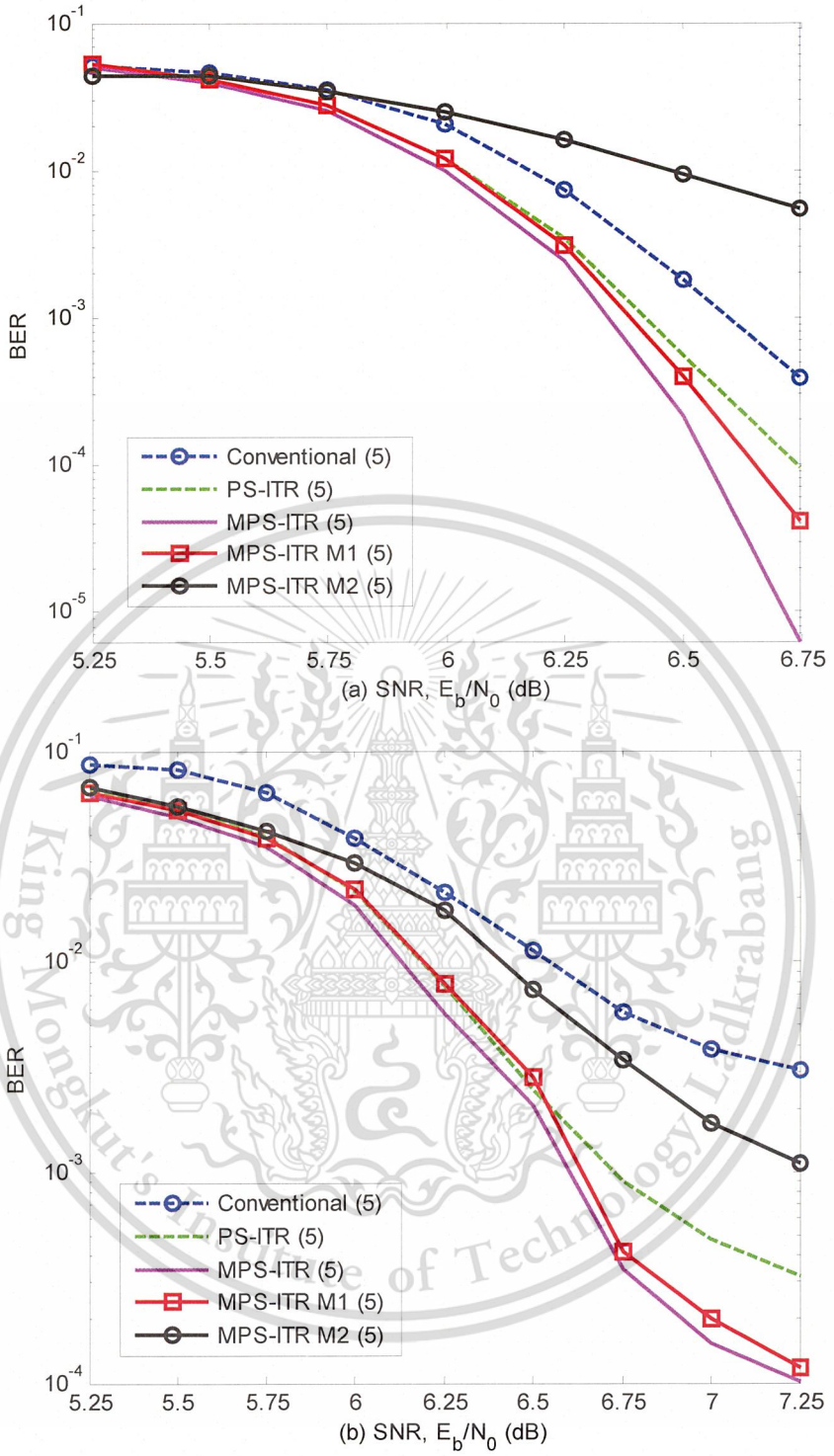


Figure 6.5 Performance comparison at the 5-th iteration when (a) $\sigma_w / T = 0.6\%$ and (b) $\sigma_w / T = 1.2\%$

During an acquisition mode, the PLL gain parameters (α's) for the conventional receiver and PS-ITR were designed to recover the phase change within 256 symbols (according to its preamble), whereas those for MPS-ITR, MPS-ITR-M1, and MPS-ITR-M2 were designed to

recover the phase change within 128 symbols because the preamble was divided into two parts (according to the modified PSP-SOVA-M algorithm as explained in Section 6.2). Note that the α 's for all schemes were designed based on a linearized model of PLL [3], assuming that the S-curve slope is one at the origin, and there is no noise in the system. Furthermore, we consider the case where the α designed to recover the phase change within 256 symbols is used for all schemes during a tracking mode.

Figure 6.5 compares the BER performance of different iterative timing recovery schemes at the 5-th iteration for the system with a moderate random walk parameter $\sigma_w / T = 0.6\%$ (which implies a low probability of occurrence of cycle slips) and $\sigma_w / T = 1.2\%$ (which implies a high probability of occurrence of cycle slips) as a function of per-bit SNRs, E_b/N_0's. Note that the number inside the parenthesis in Fig. 5 indicates the total number of iterations used to generate each curve. Apparently, the MPS-ITR-M1 performs better than the MPS-ITR-M2, the PS-ITR, and the conventional receiver, especially when σ_w / T is large. Furthermore, it is evident that for given the number of iterations, the MPS-ITR provides better performance than the other schemes because the MPS-ITR can reduce the occurrence of cycle slips and can also automatically correct a cycle slip much more efficiently than the PS-ITR [57]. Nevertheless, we will show later that the MPS-ITR-M1 scheme can perform better than the MPS-ITR scheme when operating at low-to-moderate complexity.

Figure 6.6 compares the total number of multiplications for each iterative timing recovery scheme. Clearly, the MPS-ITR scheme has very high complexity if compared with other schemes. Additionally, we assume that the current technology can support the total number of multiplications equal to 3 iterations of the MPS-ITR scheme, which is approximately equal to 4, 6, and 13 iterations of the MPS-ITR-M1, the MPS-ITR-M2, and the conventional receiver, respectively. Therefore, it is worth comparing their performance when they all have same complexity.

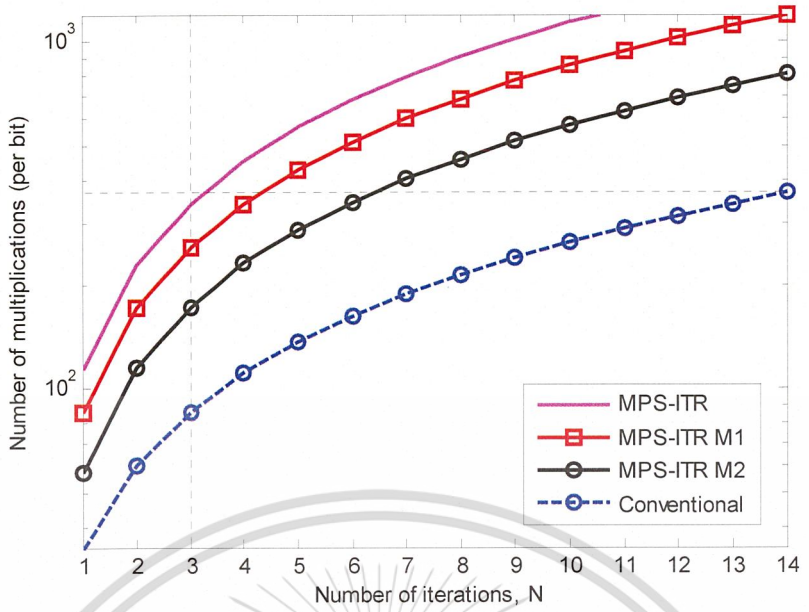


Figure 6.6 Complexity comparison (based on the PR2 channel).

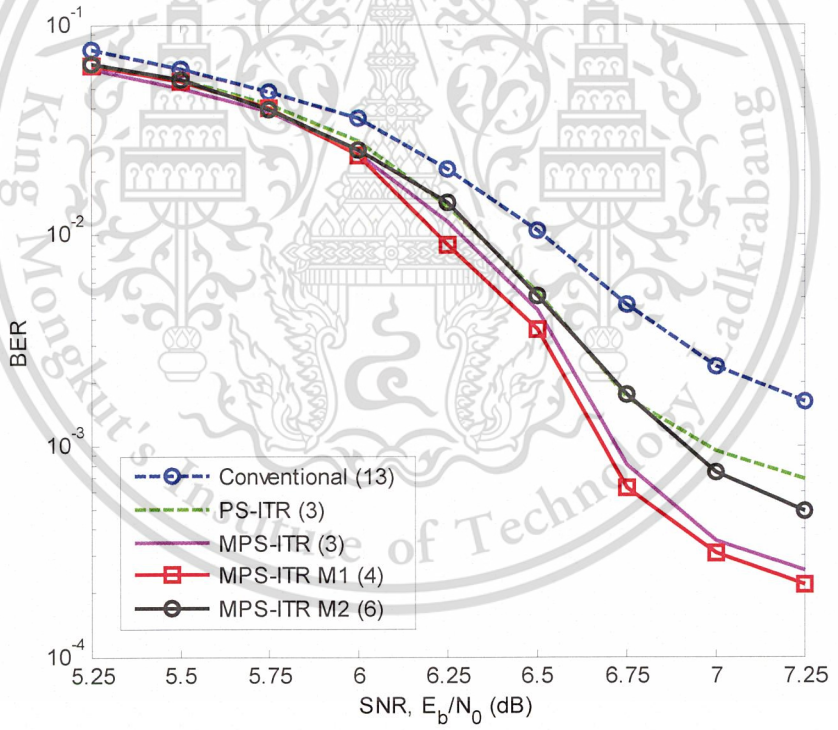


Figure 6.7 Performance comparison with same complexity at $\sigma_w / T = 1.2\%$.

Figure 6.7 compares the BER performance of different iterative timing recovery schemes when they have same complexity at $\sigma_w / T = 1.2\%$. It is apparent that the MPS-ITR-M1 performs better than other schemes. Consequently, it is worth employing the MPS-ITR-M1 in the system when the complexity is limited to a low-to-moderate amount.

6.5 Summary

In this chapter, we proposed a reduced-complexity modified per-survivor iterative timing recovery scheme to jointly perform timing recovery, equalization, and error-correction decoding. This scheme is obtained by applying the M-algorithm to the modified PSP-SOVA to make it more implementable in real-life applications. In addition, we found that the choice of M's is crucial to the overall system performance. Specifically, the M parameter mainly depends on the channel used. For the PR2 channel, $M = 3$ is a good choice for our proposed scheme. Simulation results show that at a low-to-moderate complexity, the reduced-complexity modified per-survivor iterative timing recovery scheme (with $M = 3$) performs better than other iterative timing recovery schemes and the conventional receiver.

CHAPTER 7

CONCLUSIONS

Timing recovery is the process of synchronizing the sampler with the received analog signal, sampling at the wrong times can have a devastating impact on the overall performance. The equalizer is used to shape the overall channel response to a desire response. Improving the performance of both timing recovery and the equalizer will give rise to improved reliability of an entire system. In this work, we developed and investigated new timing recovery schemes and the IIR equalizer.

In Chapter 2, we overview the backgrounds of digital magnetic recording systems and basic communication channel model for magnetic storage, as well as the writing and reading process in the hard disk drives. We also describe the dominant noises present in a magnetic recording system. Then, the realistic channel model and the ideal channel model are described.

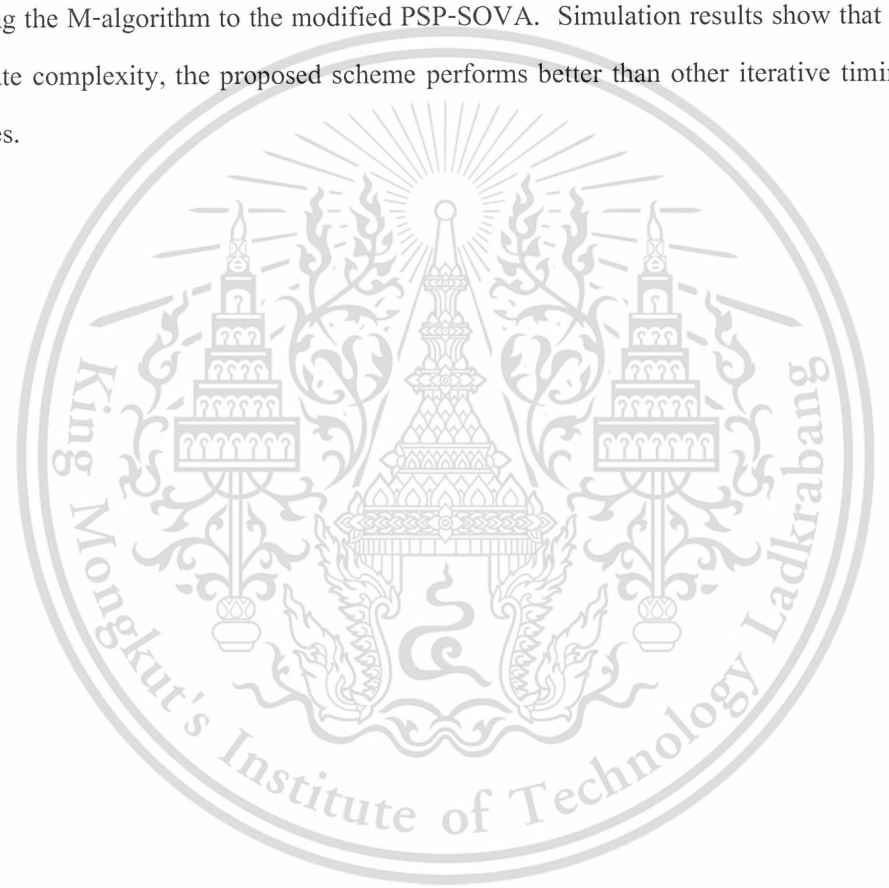
In Chapter 3, we describe how to design an IIR equalizer for the perpendicular magnetic recording channels based on the MMSE approach. We found that the proposed IIR equalizer is highly stable for PR channels. Simulation results show that for the small number of equalizer taps, the proposed IIR equalizers with the suitable poles and zeroes can outperform the FIR equalizers and they are comparable to the 11-tap FIR filter for the jitter noise levels of 0-6%. Furthermore, in the turbo equalization setting, we found that the four-zero, two-pole IIR filter performs close to the 11-tap FIR filter.

In Chapter 4, we explain the conventional timing recovery that is based on a PLL. A method of designing the PLL gain parameters based on a linearized model of PLL was given. Then, we propose the bi-directional timing recovery for perpendicular magnetic recording systems. Simulation results show that the bi-directional timing recovery is superior to the trained PLL and the conventional timing recovery, especially when the timing error is large.

In Chapter 5, we propose the MPS-ITR scheme, which iteratively exchanges soft information between the modified PSP-SOVA and the decoder. This modified PSP-SOVA uses the new split-preamble strategy, which divides the preamble into two parts, each of which is used to adjust the branch metric calculation in PSP-SOVA so as to guarantee that the survivor path occurs in a correct direction. Simulation results both in ideal and realistic channel have shown the MPS-ITR performs better than the Full PS-ITR, the PS-ITR, and the conventional receiver, especially when

the timing jitter is severe. Then, we also show the MPS-ITR has a lower complexity than the Full PS-ITR. In addition, we observed that the MPS-ITR can reduce the occurrence of cycle slips and can also automatically correct a cycle slip much more efficiently than both the PS-ITR and the Full PS-ITR. In other words, the MPS-ITR requires fewer turbo iterations than other schemes to correct a cycle slip. Finally, we show that the MPS-ITR can be employed in the realistic perpendicular magnetic recording channel and also yield better performance than others.

In Chapter 6, we propose a reduced-complexity version of MPS-ITR to jointly perform the timing recovery, equalization, and error-correction decoding. This scheme is obtained by applying the M-algorithm to the modified PSP-SOVA. Simulation results show that at a low-to-moderate complexity, the proposed scheme performs better than other iterative timing recovery schemes.



REFERENCES

[1] Cideciyan R. D., Dolivo F., Hermann R., Hirt W., and Schott W. "A PRML System for Digital Magnetic Recording." **IEEE J. Selected Areas Commun.,** vol. 10, Jan. 1992. pp. 38–56.

[2] Forney G. D. "Maximum-likelihood Sequence Estimation of Digital Sequences in The Presence Of Intersymbol Interference." **IEEE Trans. Inform. Theory,** vol. IT-18, May 1972. pp. 363–378.

[3] Bergman J. W. M. **Digital baseband transmission and recording.** Boston, Massachusetts : Kluwer academic publishers. 1996.

[4] Kovintavewat P., Warisarn C., and Supnithi P. "An MMSE Infinite Impulse Response Equalizer for Perpendicular Recording Channels with Jitter Noise." in **Proc. of ITC-CSCC 2008, Shimonoseki, Yamagushi, Japan**, Jul. 2008.

[5] Crespo P. M., Honig M. L. "Pole-zero Decision Feedback Equalization with a Rapidly Converging Adaptive IIR Algorithm." **IEEE J. Selected Areas Commun.,** vol. 9, Aug. 1991. pp. 817–829.

[6] Kim Y., Moon J. "Noise-predictive Maximum-Likelihood Method Combined with Infinite Impulse Response Equalization." **IEEE Trans. on Magn.,** vol. 35, Nov 1999. pp. 4538–4543.

[7] Beliczynski B., Kale J., and Cain G. D. "Approximation of FIR by IIR Digital Filter An Algorithm Based on Balanced Model Reduction." **IEEE Trans. on Signal Processing,** vol.40, Mar. 1992. pp. 532–542.

[8] Moon J. H., Kang B. H., and Park P. "Direct Method for Converting FIR Filter with Low Nonzero Tap into IIR Filter." **IJCSE ,** vol. 2, no.1, Winter 2008.

[9] Raghavan S., Thapar H. K. "Feed-forward Timing Recovery for Digit Magnetic Recording." in **Proc. of ICC'91**, 1991. pp. 794-498.

[10] Moon J., Lee J. "Timing Recovery in Conjunction with Maximum Likelihood Sequence Detection in The Presence of Intersymbol Interference." **IEEE Trans. Circuit and System,** vol. 55, no. 9, Oct. 2008. pp. 2884-2897.

[11] Mueller K. H., Muller M. "Timing Recovery in Digital Synchronous Data Receivers." **IEEE Trans. Commun.,** vol. 24, no. 5, May 1976. pp. 516-531.

[12] Venkataramani R., Erden M.F. "MAP-based Timing Recovery for Magnetic Recording." in **Proc. of ICC '08,** May 2008. pp. 1982-1985.

[13] Hwang E., Negi R., and Kumar B.V.K. "Extended Kalman Filter based Acquisition Timing Recovery for Magnetic Recording Read Channels." in **Proc. of ICC '08,** May 2008. pp. 1986-1990.

[14] Kovintavewat P., Barry J.R., Erden M.F., and Kurtas E.M. "Per-survivor Iterative Timing Recovery for Uncoded Partial Response Channels." in **Proc. of ICC'04,** vol.27, no.1, Jun. 2004. pp. 2715-2719.

[15] Wei Z., Erden M.F., Kavcic A., Kurtas E.M., and Venkataramani R.C. "Trellis-Based Optimal Baud-Rate Timing Recovery Loops for Magnetic Recording Systems." **IEEE Trans. on Magn.,** vol. 43, issue 7, 2007. pp. 3324-3332.

[16] Souvignier T., Friedmann A., Oberg M., Siegel P., Swanson R., and Wolf J. "Turbo Decoding for PR4: Parallel vs. Serial Concatenation." in **Proc. of ICC'99,** vol. 3, 1999. pp. 1638–1642.

[17] Kovintavewat P., Barry J.R., Erden M.F., and Kurtas E.M. "Per-survivor Iterative Timing Recovery for Coded Partial Response Channels." in **Proc. of Globecom'04, Dallas, Texas, USA,** Dec. 2004.

[18] Raheli R., Polydoros A., and Tzou C. K. "The Principle of Per-Survivor Processing: A General Approach to Approximate and Adaptive (MLSE)." in **Proc. of Globecom'91,** vol. 2, Dec. 1991. pp. 1170–1175.

[19] Bahl L. R., Cocke J., Jelinek F., and Raviv J. "Optimal Decoding Of Linear Codes for Minimizing Symbol Error Rate." **IEEE Trans. Inform Theory,** vol. IT-20, Mar. 1974. pp. 248-287.

[20] Kovintavewat P., Barry J. R., Eeden M. F., and Kurtas E. "Reduced-Complexity Per-Survivor Iterative Timing Recovery for Coded Partial Response Channels." **IEEE ICASSP'05, Philadelphia, USA,** vol. 3, Mar. 2005. pp. 841–844.

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

[22] Annampedu V., Aziz P. M. "Adaptive Algorithms for Asynchronous Detection of Coded Servo Signals Based On Interpolation." **IEEE Trans. on Magn.,** vol. 41, no. 10, Oct. 2005. pp. 2890-2892.

[23] Adireddy S., Tong L., and Viswanathan H. "Optimal Placement of Known Symbols for Frequency-Selective At-Fading Channels." **IEEE Trans. Inform. Theory,** vol. 48, no. 8, Aug. 2002. pp. 2338-2353.

[24] Negi R., Cioffi J. "Pilot Tone Selection for Channel Estimation in a Mobile OFDM System." **IEEE Trans. Consum. Electron.,** vol. 44, no. 3, Aug. 1998. pp. 1122-1128.

[25] Nayak A. R., Barry J. R., and McLaughlin S. W. "Optimal Placement of Training Symbols for Frequency Acquisition: A Cramer-Rao Bound Approach." **IEEE Trans. on Magn.,** vol. 42, no. 6, Jun. 2006. pp. 1730-1742.

[26] Simmons J. B., Mohan S. "Sequential Coding Algorithms: A Survey and Cost Analysis." **IEEE Trans. Commun.,** vol. COM-32, Feb. 1984. pp. 169–176.

[27] Wang S. X., Taratorin A. M. **Magnetic Information Storage Technology**. San Diego : Academic Press, 1999.

[28] Suzuki T. "Perpendicular Magnetic Recording: Its Basics and Potential for the Future." **IEEE Trans. on Magn.,** vol. MAG-20, Sep. 1984. pp. 675–680.

[29] Wicker S. B. **Error control systems for digital communication and storage.** Upper Saddle River, New Jersey : Printice Hall International, 1995.

[30] Vasic B., Kurstas E. M. **Coding and signal processing for magnetic recording system.** New York : CRC Press, 2005.

[31] Siegel P.H. "Recording Codes for digital Magnetic Storage." **IEEE Trans. on Magn.,** vol. MAG-21, Sep. 1985. pp. 1344-1349.

[32] Immink K. A. S. "Runlength-Limited Sequences." **Proceeding of the IEEE,** vol. 78, Nov. 1990. pp. 1745–1759,

[33] Roscamp T. A., Boerner E. D., and Parker G. J. "Three-dimensional Modeling of Perpendicular Recording with Soft Underlayer." **J. of Applied Physics,** vol. 91, May 2002.

[34] Leon-Garcia A., **Probability and random processes for electrical engineering.** 2nd ED. New York : Addison-Wesley Publisher Company, Inc. 1994.

[35] Zhu J. G., Bertam H. N. "Recording and Transition Noise Simulations in Thin Film Media." **IEEE Tran. on Magn.,** vol. 24, no. 6, Nov. 1988. pp. 2706-2708.

[36] Zhu J. G. **Micromagnetic of Thin Film Media.** in Magnetic Recording Technology edited by C. D. Mee and E. D. Daniel. New York : McGraw-Hill, 1995.

[37] Erden M. F., Kurtas E. M. "Baseline Wander Compensation for Perpendicular Recording." **IEEE Trans. on Magn.,** vol. 40, no. 4, Jul. 2004. pp. 3114–3116.

[38] Sawaguchi H., Nishida Y., Takano H., and Aoi H. "Performance Analysis of Modified PRML Channels for Perpendicular Recording Systems." **J. Magn. Mater.,** vol. 235, Jan. 2001. pp. 265–272.

[39] Barry J. R., Lee E. A., and Messerschmittm D. G. **Digital Communication.** 3rd ED. Boston, Massachusetts : Kluwer Academic Publishers, 2003.

[40] Thapar H. K., Patel A. M. "A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording." **IEEE Trans. on Magn.,** vol. 23, Sep. 1987. pp. 3666–3668.

[41] Kovintavewat P., Ozgunes I., Kurtas E., Barry J. R., and McLaughlin S. W. "Generalized Partial Response Targets for Perpendicular Recording with Jitter Noise." **IEEE Trans. on Magn.,** vol. 38, Sep. 2002. pp. 2340–2342.

[42] Moon J., Zeng W. "Equalization for Maximum Likelihood Detector." **IEEE Trans. on Magn.,** vol. 31, no. 2, Mar. 1995. pp. 1083-1088.

[43] Cideciyan R. D., Dolivo F., Hermann R., Hirt W., and Schott W. "A PRML System for Digital Magnetic Recording." **IEEE J. Selected Areas Commun.,** vol. 10, Jan. 1992. pp. 38–56.

[44] Eleftheriou E., Olcer S. "Low-Density Parity Check Codes for Digital Subscriber Lines." **in Proc. 2002 Int. Conf. on Comm.,** , Apr.–May. 2002. pp.1752-1757.

[45] Choi J., Bouchard M., and Yeap T. H. "Decision Feedback Recurrent Neural Equalization with Fast Convergence Rate." **IEEE Trans. on Neural Networks**, vol. 16, no. 3, May 2005. pp. 699-708.

[46] Osawa H., Hino M., Shinohara N., Okamoto Y., Nakamura Y., and Muraoka H. "Simplified Neural Network Equalizer with Noise Whitening Function for GPRML System." **IEEE Trans. on Magn.,** Vol. 44, no. 11, Nov. 2008. pp. 3777-3780.

[47] Tyner D. J., Proakis J. G. "Partial Response Equalizer Performance in Digital magnetic Recording Channels." **IEEE Trans. on Magn.,** vol. 29, no. 6, Nov. 1993. pp. 4194–4208.

[48] Richardson T.J., Shokrollahi M.A., and Urbranke R.L. "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes." **IEEE Trans. Information Theory,** vol. 47, Feb. 2001. pp. 619-637.

[49] Andrea A. N., Mengali U., and Vitetta G. M. "Approximate ML Decoding of Coded (PSK) With No Explicit Carrier Phase Reference." **IEEE Trans. Commun.,** vol. 42, Feb./Mar./Apr. 1994. pp. 1033-1039.

[50] Mengali U., Andrea A. N. **Synchronization techniques for digital receivers.** New York : Plenum Press, 1997.

[51] Shafiee H. "Timing Recovery for Sampling Detectors in Digital Magnetic Recording." in **Proc. of ICC'96,** vol. 1, Jan. 1996. pp. 577–581.

[52] Barry J. R., Kavcic, A. McLaughlin S. W., and Nayak A. R. "Iterative Timing Recovery." **IEEE Signal Processing Magazine,** vol. 21, Jan. 2004. pp. 89–102.

[53] Jin X., Kavcic A. "Cycle-slip Detection Using Soft-Output Information." in **Proc. of ICC'01,** vol. 9, Jun. 2001. pp. 2706–2710.

[54] Nayak A. R. "**Iterative timing recovery for magnetic recording channels with low signal-to-noise ratio.**" Ph.D. Thesis of Georgia Institute of Technology, Georgia, Jun. 2004.

[55] Nayak A. R., Barry J. R., and McLaughlin S. W. "Joint Timing Recovery and Turbo Equalization for Coded Partial Response Channels." **IEEE Trans. on Magn.,** vol. 38, Sep. 2003. pp. 2295–2297.

[56] Kovintavewat P., Erden M. F., Kurtas E. M., and Barry J. R. "Oversampled Timing Recovery for Magnetic Recording Channels." in **Proc. of the IEEE International Conference on Magnetics (Intermag) 2003**, 2003. pp. DT-06.

[57] Warisarn C., Kovintavewat P., and Supnithi P. "Iterative Timing Recovery with the Split-Preamble Strategy for Coded Partial Response Channels." **IEICE Trans. Elec.,** vol. E94-C, no. 3, Mar. 2011. pp. 368-374.

[58] Annampedu V., Aziz P. M. "Adaptive Algorithms for Asynchronous Detection of Coded Servo Signals Based on Interpolation." **IEEE Trans. on Magn.,** vol. 41, no. 10, Oct. 2005. pp. 2890–2892.

[59] Adireddy S., Tong L., and Viswanathan H. "Optimal Placement of Known Symbols for Frequency-Selective Flat-Fading Channels." **IEEE Trans. Inform. Theory,** vol. 48, no. 8, Aug. 2002. pp. 2338–2353.

[60] Negi R., Cioffi J. "Pilot Tone Selection for Channel Estimation In A Mobile OFDM System." **IEEE Trans. Consum. Electron.,** vol. 44, no. 3, Aug. 1998. pp. 1122–1128.

[61] Warisarn C., Kovintavewat P., and Supnithi P. "Bi-Directional Timing Recovery for Perpendicular Magnetic Recording Channels." in **Proc. of TENCON 2009, Singapore,** Nov, 2009.

[62] Kovintavewat P., Barry J. R. **Iterative timing recovery: a per-survivor approach.** VDM Verlag Publisher, Germany, Nov. 2009.

[63] Gallager R. "Low-Density Parity-Check Codes." **IRE Trans. Inform. Theory,** vol. IT-8, Jan. 1962. pp. 21–28.

[64] Simmons J. B., Mohan S. "Sequential Coding Algorithms: A Survey and Cost Analysis." **IEEE Trans. Commun.,** vol. COM-32, Feb. 1984. pp. 169–176.

[65] Seshadri N., Anderson J. B. "Decoding of Severely Filtered Modulation Codes Using the (M,L) Algorithm." **IEEE J. Selected Areas Commun.,** vol. 7, Aug. 1989. pp. 1006–1016.

[66] Iki M., Takanashi M., and Ochiai H. "Peak Power Reduction of Single-Carrier Signals Using Trellis Shaping with M and Stack Algorithms." **Proc. of the 70th IEEE Vehicular Techology Conference,** 2009. pp. 1-5.

[67] Joohyun L., Jaejin L. "A Simplified Noise-Predictive Partial Response Maximum Likelihood Detection Using M-Algorithm for Perpendicular Magnetic Recording Channels." **IEEE Trans. on Magn.,** vol.41, no.2, Feb. 2006. pp.1064-1066.

[68] Tosi S., Power M., and Conway T. "Multitrack Digital Data Storage: A Reduced Complexity Architecture." **IEEE Trans. on Magn.,** vol. 43, no. 3, Mar. 2007. pp. 1101-1111.

[69] Nayak A. R., Barry J. R., Feyh G., and McLaughlin S. W. "Timing Recovery with Frequency Offset and Random Walk: Cramér–Rao Bound and a Phase-Locked Loop Postprocessor." **IEEE Trans. on Magn.,** vol. 54, no. 11, Nov. 2006. pp. 2004-2013.

AUTHOR BIOGRAPHY

Author:	Mr. Chanon Warisarn
Date of Birth:	November 4, 1982
Address:	2/66 M. The Nature City, Leabklongphasrijarearn, Phetkhasem 69 Phetkhasem Rd., Nongkham, Bangkok 10160.
Bachelor Degree:	Electronics Engineering Technology (First Hon.) King Mongkut's University of Technology North Bangkok.
Year of Graduation:	2006.
Interested Research:	Target equalizer design, information theory, and signal processing in timing recovery for magnetic recording system.

Related Publication

[1] Warisarn C., Kovintavewat P., and Supnithi P. "Iterative Timing Recovery with the Split-Preamble Strategy for Coded Partial Response Channels." **IEICE Trans. Elec.**, vol. E94-C, no. 3, Mar. 2011. pp. 368-374.

[2] Warisarn C., Kovintavewat P., and Supnithi P. "Modified Per-Survivor Iterative Timing Recovery In Perpendicular Magnetic Recording." in Proc. of **PMRC 2010, Sendai, Japan,** Oct. 2010.

[3] Warisarn C., Kovintavewat P., and Supnithi P. "Reduced-Complexity Modified Per-Survivor Iterative Timing Recovery for Coded Partial Response Channels." in Proc. of **ICT2010, Bangkok, Thailand**, Aug. 2010.

[4] Warisarn C., Kovintavewat P., and Supnithi P. "Iterative Timing Recovery with the Split-Preamble Strategy for Magnetic Recording Channels." **IEEE MMM-Intermag Conference (INTERMAG2010), Washington DC, USA,** Jan. 2010. pp. GH-09.

[5] Warisarn C., Kovintavewat P., and Supnithi P. "Bi-Directional Timing Recovery for Magnetic Recording Systems." in Proc. of **TENCON2009, Singapore,** Nov. 2009.

[6] Kovintavewat P., Warisarn C., and Supnithi P. "An MMSE Infinite Impulse Response Equalizer for Perpendicular Recording Channels with Jitter Noise." in **Proc. of ITC-CSCC 2008, Shimonoseki City, Japan**, Jul. 2008. pp. 929–932.

[7] Warisarn C., Kovintavewat P., and Supnithi P. "An Infinite Impulse Response Equalizer for Magnetic Recording Channels." **KKU Research Journal**, vol.13, no. 4, May 2008. pp. 521–525.

[8] Warisarn C., Supnithi P. "The Effects of Modulation Code Rate and Channel Pulses on Noise Margin of the Magnetic Recording Channel." **KKU Engineering Journal**, vol. 34, no. 4, Aug. 2007. pp. 431–438.