

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 1343848

Timing recovery for two-dimensional modulation codes

**Ma, King-Yan Alan, M.S.
The University of Arizona, 1991**

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

**TIMING RECOVERY
FOR
TWO-DIMENSIONAL MODULATION CODES**

by

King-Yan Alan Ma

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING
In the Graduate College
THE UNIVERSITY OF ARIZONA

1 9 9 1

STATEMENT BY AUTHOR


This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: 

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:


Michael W. Marcellin
Assistant Professor of
Electrical and Computer Engineering

April 26, 1991
Date

To my parents

ACKNOWLEDGMENTS

I wish to gratefully acknowledge my indebtedness to all individuals and organizations who have contributed to both the content and quality of this thesis. It is hard for me to imagine how this work could have been completed without feedback from the faculty members of the Department of Electrical and Computer Engineering. I have been especially fortunate in this regard by having the opportunity to learn from my advisor, Dr. Michael W. Marcellin, and benefit from his constant encouragement and guidance. Also, I would like to thank my thesis committee members, Dr. Randall K. Bahr and Dr. Pamela A. Nielsen, for reviewing this thesis.

At The University of Arizona I have been enjoying an excellent collaboration with my colleagues, and I wish to express my deep appreciation to all of them for sharing my enthusiasm in this challenging project.

Finally, it is my pleasant duty to express my sincere gratitude to International Business Machines Corporation, The University of Arizona Foundation, and the office of the Vice President for Research, for the financial support which made this work possible.

TABLE OF CONTENTS

LIST OF FIGURES	6
LIST OF TABLES	7
ABSTRACT	8
1. Introduction	9
2. Background	13
2.1. The Digital Magnetic Recording Channel	13
2.1.1. System Development	13
2.1.2. Run-Length-Limited (RLL) Codes	14
2.1.3. The Writing and Reading Process	18
2.2. Timing Recovery	21
2.2.1. Problem Statement	21
2.2.2. Synchronization by Peak Detection	22
2.2.3. Synchronization by Nonlinearity	23
2.3. Two-Dimensional Modulation Codes	26
2.4. Phase-Locked Loops	29
3. Track Alignment	33
3.1. System Operation	33
3.2. Example	38
4. Performance of Timing Recovery Scheme	43
4.1. Synchronizer Circuit	43
4.2. Simulation Results	45
5. Summary	53
REFERENCES	56

LIST OF FIGURES

2.1. A general digital magnetic recording system.	13
2.2. A (1,3) finite-state transition diagram.	17
2.3. The waveforms at various stages for NRZI encoding method.	19
2.4. A Lorentzian pulse.	20
2.5. A peak detection synchronizer.	22
2.6. A symbol timing recovery circuit for baseband PAM signal.	24
2.7. A finite-state transition diagram for the two-dimensional (1,3) code.	28
2.8. A general structure of a phase-locked loop.	30
3.1. The circuit for track alignment.	35
3.2. The characteristic of the variable delay element.	36
3.3. The input signals.	39
3.4. The delayed input signals.	40
3.5. The inputs and output of the multiplier.	40
3.6. The output of the low-pass filter.	41
3.7. The output of the integrator.	42
3.8. A family of $v'(t)$ for different initial offsets.	42
4.1. The square-and-filter synchronizer.	43
4.2. The input signals and timing wave.	44
4.3. Timing Jitter Performance at 0.5 code bits/ T_{50}	48
4.4. Timing Jitter Performance at 1.0 code bit/ T_{50}	49
4.5. Timing Jitter Performance at 1.5 code bits/ T_{50}	50
4.6. Timing Jitter Performance at 2.0 code bits/ T_{50}	51
4.7. An eye diagram at 1.0 code bit/ T_{50}	52

LIST OF TABLES

4.1. Code rules for the Modified Frequency Modulation codes.	45
----------------------------------------------------------------------	----

ABSTRACT

An interest in the application of two-dimensional modulation codes to the coding of information in magnetic recording has led to an investigation of symbol synchronization for these channels. Binary data storage in magnetic recording systems is accomplished by mapping sequences of 0's and 1's into patterns of magnetic flux changes along a track. With two-dimensional modulation codes, data is encoded in a manner such that clocking information can be derived from any number of those tracks while ensuring against intersymbol interference in each track individually. In this thesis, timing extraction for multiple parallel tracks in the presence of additive white Gaussian noise through a square-law device is studied and the design of the synchronizer circuitry is also presented. Finally, extensive simulations are carried out to determine the performance of the system.

CHAPTER 1

Introduction

Modern communication theory has played a major role in increasing the efficiency and reliability of digital recording systems. A great deal of recent research has been devoted to the application of coding techniques to the improvement of system performance. Wolf and Ungerboeck [1] have considered trellis coding techniques for improving the reliability of digital transmission. Kobayashi and Tang [2] have modeled the magnetic recording process as a partial-response communication channel [3] [4]. Calderbank, Heegard and Lee [5] have suggested a method of designing codes for channels with intersymbol interference, while Fredrickson and Wolf [6] have introduced various coding techniques using multiple block (d, k) codes. However, many of the techniques which have been successfully implemented in communication channels have not yet found their way into magnetic recording systems.

Much of the recent work in magnetic recording research has been focused on increasing the storage capacity, that is, the amount of data that can be stored in a unit area of a magnetic media. This can be achieved by increasing the linear density (i.e., the number of bits per inch along a track) and/or by increasing the density of

the tracks (i.e., the number of tracks per inch). All of this must be done without sacrificing the reliability of the retrieved data.

Although the storage capacity of digital recording on magnetic media has been steadily increasing over the last decade, most of these gains have been achieved by means other than coding and modulation. Improved magnetic heads, media technology, actuator, and tracking systems have been some of the methods used to achieve these increase [7]–[9]. Despite the fact that researchers believe that advanced coding techniques have tremendous potential for providing further improvements in data transfer rate, storage capacity, and reliability of data storage systems [10]–[13], the possibility of increasing the storage capacity even further by using *multi-track* modulation techniques seems to have been over-looked until the birth of two-dimensional modulation codes [14].

With two-dimensional modulation codes, several tracks are encoded, written, and read in parallel. Since encoding each track independently is a special case of encoding multiple tracks in parallel, it is clear that the maximum achievable storage capacity will not be decreased by this method. In fact, storage capacities that are significantly greater than those obtainable by systems that operate on each track individually can be achieved.

Two-dimensional modulation codes are motivated by, and should prove to be very useful for, magnetic tape recording where multiple tracks are written and read in parallel and self-clocking is desired at the same time. We shall concentrate on

magnetic recording channels in which signals are modulated two-dimensionally. As a result, the detector must be able to perform the decision process on multiple tracks in parallel. This implies that timing information has to be recovered from these tracks as a whole.

Generally, symbol timing or synchronization is concerned with the problem of determining a clock for periodically sampling the received signal for the purpose of recovering the transmitted information. There are two important factors that must be considered in implementing a coherent communication system requiring symbol synchronization. First, it is necessary to maintain a high efficiency in the data detection process; inaccurate symbol synchronization directly reduces the probability of making correct decisions. Second, for a given amount of power available for transmission of data, the synchronization scheme should require as little of this power as possible.

There are two schools of thought with regard to providing symbol synchronization. One approach is to transmit a clock signal along with the information-bearing signal, in multiplexed form, to the receiver. There are at least two major drawbacks with this approach. One is that a portion of the transmitter power is allocated to the transmission of the clock signal. A second disadvantage is that the two signals must be adequately separated at the receiver so that the clock signal does not interfere with the demodulation of the information-bearing signal and vice versa. An alternative

approach is to derive the clock signal from the information-bearing signal. We confine our attention to the latter.

Before going into the details of our research, a review of the principles of run-length-limited codes, magnetic recording, timing recovery, two-dimensional modulation codes, and phase-locked loops is given in Chapter 2. Chapter 3 presents a method to make sure that signals from neighboring tracks are time-synchronous, while Chapter 4 is devoted to the discussion of the timing recovery scheme actually employed in our study. At the end of that chapter, simulation results are presented to illustrate the sensitivity of the system to noise. Also, the performance of a double-track system is compared with that of a traditional single-track system and the results are graphed. Finally, Chapter 5 serves as a summary.

CHAPTER 2

Background

2.1 The Digital Magnetic Recording Channel

2.1.1 System Development

The notion of treating the magnetic recording process as a communication channel is not a new one. This process essentially consists of two distinct operations; data writing and data reproduction (read-back detection) [15]. Figure 2.1 shows a general block diagram of such a system.

The source generates binary sequences which are to be recorded. Since uncoded binary symbols generally are not suitable for direct storage, source sequences require

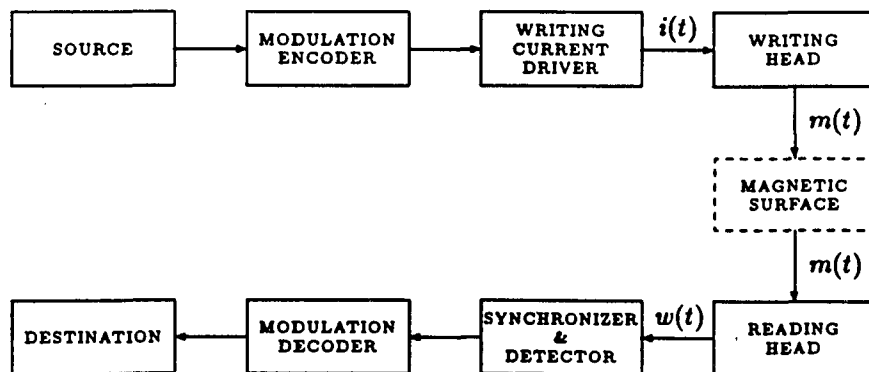


Figure 2.1: A general digital magnetic recording system.

transformation into new sequences which are suitable for storage in the media. The modulation encoder fulfills this requirement by mapping the source sequences into new binary sequences chosen to inhibit intersymbol interference while ensuring that synchronization can be maintained.

One of the major characterizing features of storage systems, which differentiates them from conventional communication systems, is their operation mode. The writing current driver generates current pulses which induce a magnetic field in the writing head to record the information on a magnetic surface in the form of polarity changes. The reading process is the opposite of the writing process. When it comes to data reproduction, the entire mechanism discussed above is reversed by the reading head, synchronizer, detector, and modulation decoder. In the following subsections, we review these ideas in a more in-depth fashion.

2.1.2 Run-Length-Limited (RLL) Codes

Most magnetic storage devices in use today employ saturation recording to store data, this means that binary data is written on a track by magnetizing the media in one of two (opposite) directions. The transition between two adjacent areas, magnetized in opposite directions, is called a *flux change*. For this thesis, we assume that non-return-to-zero-inverse (NRZI) encoding method is used, so that a 1 is represented by a flux change and a 0 is represented by no flux change. Generally, considerations

such as self-clocking and reduction of intersymbol interference require that the sequences to be recorded have special binary properties called channel constraints.

A constraint on the minimum distance between flux changes is necessary to control intersymbol interference. On the other hand, a constraint on the maximum distance between flux changes is needed to ensure that the read-out waveform contains enough transitions to recover timing information [16]. In order to fulfill these channel constraint requirements, the modulation encoder maps the source sequences into code sequences by employing run-length-limited (RLL) codes. These codes were pioneered by Franaszek [10] in the late 1960's. One particular variant of RLL codes is characterized by two parameters (d,k) , which represent, respectively, the minimum and maximum number of '0' symbols between consecutive '1' symbols.

The problem faced by the coding theorist is the construction of a simple, efficient correspondence, or code mapping, between the arbitrary binary strings that a user might want to store on a magnetic disk and the (d,k) constrained code strings which the modulation decoder can more easily recover correctly. We now give the term "efficient" quantitative meaning by introducing a third important code parameter, the *code rate*.

The conversion of arbitrary strings to constrained (d,k) strings can be accomplished as follows [17]. Pick a code word length n and list all concatenable strings of length n which satisfy the (d,k) constraint. If there are at least 2^m such strings (or code words), assign a unique code word to each of the 2^m possible binary input

words of length m . This kind of code mapping is commonly referred to as a *block code*. The ratio, m/n , of input word length m to code word length n is called the code rate. Clearly, there are only 2^n unconstrained binary strings of length n , and there can be no more than this number of constrained code words. Therefore, the rate must satisfy $m/n \leq 1$. Also, there is a maximum achievable rate, called the *Shannon capacity* $C(d, k)$, which we now discuss.

Shannon [18] proved that, as the code word length grows, the number of constrained code words approaches 2^{Cn} from below, for some constant $C = C(d, k)$ which depends on the code constraints. This result implies that the rate m/n of any code mapping for that constraint must satisfy $m/n \leq C(d, k)$. Roughly speaking, a code is called efficient if the rate m/n is close to $C(d, k)$. Shannon's proof also showed that a block code is possible at any rate $m/n < C(d, k)$, provided that long enough code words are used.

When using a (d, k) code, the minimum distance between flux changes is $d+1$ code bit intervals. Hence, we can write these digits on a track at a density of $(d+1)F$ code bits per inch without violating the condition that there will be no more than F flux reversals per inch. This suggests that we can increase the recording capacity by a factor of $d+1$ with (d, k) codes. Unfortunately, this is not the case since the average number of source bits per code bit is upper bounded by $C(d, k) < 1$. The maximum possible recording capacity is thus $(d+1)FC(d, k)$ in units of source bits/inch where d , k , and F are fixed constants defined by the channel. Hence, for a given channel,

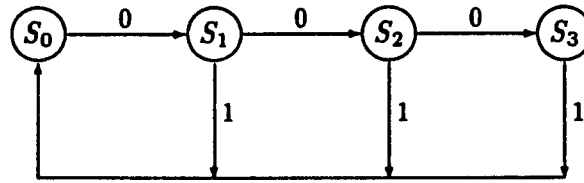


Figure 2.2: A (1,3) finite-state transition diagram.

the maximum possible storage capacity depends only on the capacity of the code. It is the task of the design engineer to find a code at a rate close to the capacity $C(d, k)$.

One can employ a finite-state transition diagram (FSTD) in order to conveniently represent the binary strings satisfying the (d, k) constraint. In general, it is a diagram in which each of the meaningful past histories of the channel inputs are distinguished by a node called a state. We imagine a number of possible states labeled S_0, S_1, \dots, S_k . Each state corresponds to one or more of the possible patterns that the recent input sequence has passed through. Loosely, we say that the channel is in one of these states, but to be precise, we must say that the channel input history is in that state. Paths in the graph show how the state may change with time.

For each state, only certain symbols from the set $\{0, 1\}$ may be transmitted next. At each point in time, one of the allowed input symbols is transmitted and the channel moves to a new state, depending both on the old state and on the particular symbol transmitted. Figure 2.2 shows an FSTD for an RLL code with $(d, k) = (1, 3)$. In this figure, state S_j denotes that j 0's have occurred since the most recent 1. Hence, when the system is in state S_0 , a 0 must be the next symbol transmitted to satisfy the $d = 1$ constraint. In this case, the next state is S_1 where either a 1 or a 0

may be transmitted to cause a transition to either state S_0 or S_2 , respectively. The transitions from the other states are easily deduced in the same way.

The capacity $C(d, k)$ of the RLL (d, k) constrained channel is directly related to the structure of the FSTD. The state transition matrix T associated with the FSTD with states S_0, \dots, S_k has elements given by t_{ij} , which is the number of arrows in the FSTD starting in state S_i and ending in state S_j . For the $(d, k) = (1, 3)$ case,

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

The capacity $C(d, k)$, in units of source bits per code bit, was shown by Shannon to be

$$C(d, k) = \log_2 \lambda \quad (2.1)$$

where λ is the largest eigenvalue of T . In the $(1, 3)$ case, $\lambda \approx 1.465$, so that $C(d, k) \approx 0.5515$. Hence, any code with rate close to 0.5515 can be considered efficient.

2.1.3 The Writing and Reading Process

Once the source sequences are encoded, they are ready to be processed by the writing current driver which creates a magnetic field to record the information on the magnetic media. In Figure 2.1, $i(t)$ is the writing current generated by the writing current driver, $m(t)$ is the magnetization pattern created by the writing head, and $w(t)$ is the read-back voltage which is proportional to the time derivative of the magnetic flux through the read transducer's core.

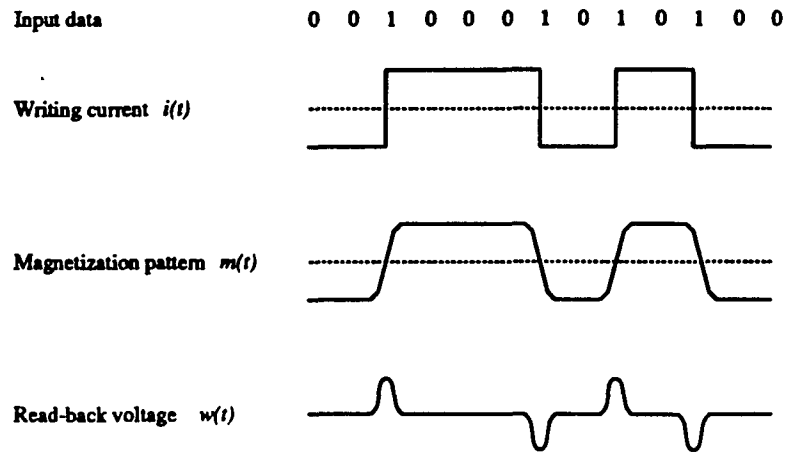


Figure 2.3: The waveforms at various stages for NRZI encoding method.

The writing current waveform is constant in magnitude and alternates in polarity. The reading head output voltage for a step change in writing current is a pulse. Since a writing current waveform is composed of a succession of alternating step-like changes in current, the associated output voltage waveform will inherently consist of an alternating pulse sequence. Relationships among a sample of coded binary data, $i(t)$, $m(t)$, and $w(t)$, for NRZI encoding method, are illustrated in Figure 2.3.

A digital storage system is fundamentally characterized by its step response $g(t)$. It plays the same role in magnetic recording systems as the impulse response does for conventional linear systems. The output voltage response of the read circuit, due to a step input current to the write circuit, is a voltage pulse which can be modeled by a Lorentzian pulse. This pulse is given by

$$g(t) = \frac{1}{1 + \left(\frac{2t}{T_{50}}\right)^2} \quad (2.2)$$

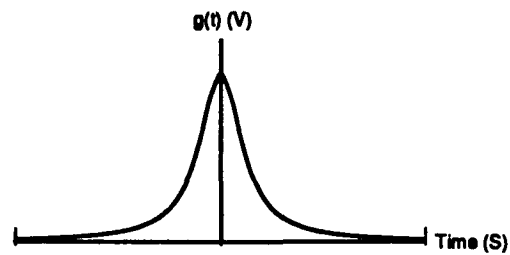


Figure 2.4: A Lorentzian pulse.

where T_{50} is referred to as the pulse width at half amplitude. Figure 2.4 shows an isolated Lorentzian pulse. Justification for the use of the Lorentzian pulse comes from the fact that it is the derivative of the arc-tangent function, which has been widely accepted as a good representation of the magnetization distribution in an isolated transition region [19].

Clearly, the shape and amplitude of the read-back pulse is very important to the recovery of the recorded data. At low bit densities the output pulses are resolved with no difficulty. As the density of the recorded bits is increased, several problems are encountered. The first is the reduction of amplitude of read-back pulses as a result of intersymbol interference. The second problem, which imposes the most severe limitation on the performance of digital magnetic recording systems, is the peak-shift problem. Peak-shift is defined as the outward displacement of the peaks of the read-back signal when there are two successive flux reversals preceded and followed by nonreversal situations [20]. At this point the necessity of the use of (d,k) codes to ensure that flux reversals are sufficiently separated should be obvious.

2.2 Timing Recovery

2.2.1 Problem Statement

In coherent digital communications, accurate synchronization of the transmitter and receiver timing is a necessity. Timing information is typically supplied to a receiver in one of two ways: 1) by transmission of a reference signal with the information waveform, or 2) by derivation of synchronization information to generate a local clock from the incoming waveform itself. Generally, the first represents a means of simplifying receiver hardware at the cost of increased transmitter power. The second is used where transmitter efficiency is of prime importance and receiver complexity of secondary concern.

In magnetic recording, signals are read from a magnetic media using the usual type of read heads. A flux reversal on a track causes a pulse at the output of the reading head. A detector circuit must decide on the presence or absence of a pulse for each time interval that may contain such a pulse. Therefore, an accurate timing circuit is required as part of the detector. In order to guarantee that there is a steady stream of pulses from which timing can be derived, we must ensure that there is not a long period of time during which there are no output pulses. Again, this justifies the use of (d,k) codes.

In general, the signal read from the magnetic tape can be expressed as

$$w(t) = \sum_k a_k g(t - kT - \tau) + n(t) \quad (2.3)$$

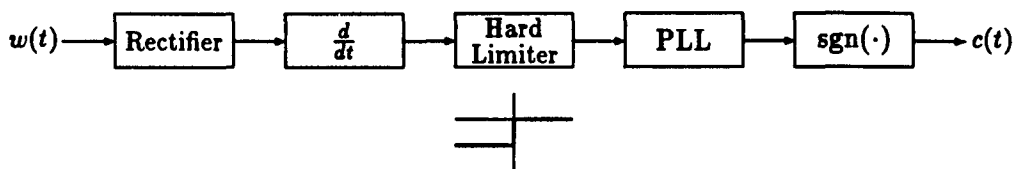


Figure 2.5: A peak detection synchronizer.

where $g(t)$ is the Lorentzian pulse, T is the code bit period, $n(t)$ is a white Gaussian noise process contributed by the read/write electronics, and a_k takes on values -1 , 0 , or 1 , corresponding to a negative-going transition, no transition, or positive-going transition of $i(t)$, respectively. Notice that $w(t)$ contains a constant τ . This constant is the (unknown) time-shift of the writing head time axis relative to the reading head time axis. In order to retrieve the information a_k , we typically sample the recorded signal every T seconds at the instants when the amplitude of the pulse is maximum. The knowledge of timing information required to accomplish this can be achieved by using a synchronizer.

2.2.2 Synchronization by Peak Detection

In a magnetic recording system, it is essential to have timing information indicating the proper time instants to operate the sampling device which samples the recorded signal, and the timing error must be held to a small fraction of the pulse repetition interval for satisfactory performance. A timing recovery scheme frequently used for magnetic recording is based on peak detection [21]. Figure 2.5 shows a mechanism which converts input signals into a clock signal with zero-crossings occurring at the proper sampling instants.

Assuming that NRZI encoding method is used, the input signal $w(t)$ can be defined by (2.3). This signal is then rectified and differentiated. The output of the differentiator is a signal whose negative-going zero-crossings are indicators of magnetic transitions. The positive portions of this signal are suppressed to zero by the hard-limiter with the characteristic as shown. The resulting signal has negative-going square pulses corresponding to 1's in the original code sequence. At this point a phase-locked loop (PLL)¹ is used to generate a sinusoidal timing wave, which is in turn hard-limited to form a square wave clock $c(t)$ to provide the detector timing information to recover the code sequence.

2.2.3 Synchronization by Nonlinearity

The schemes utilizing peak detection are commonly used for derivative equalization channels. For waveform restoration channels, a practical and popular method is the square-and-filter approach which produces a "timing wave" which ideally has some periodic attribute, such as uniformly spaced zero-crossings [21].

It is possible that the timing circuit might be simply a narrow-band filter tuned to a harmonic of the pulse repetition frequency. This scheme works in the situation where the signal has discrete spectral components. This situation occurs only when the data sequence has a nonzero mean value and the Fourier transform of the data pulse shape does not vanish at some multiple of the pulse repetition frequency [22]. In general, it is often desirable to design a system where neither of these conditions

¹The principles of the PLL will be reviewed at the end of this chapter.

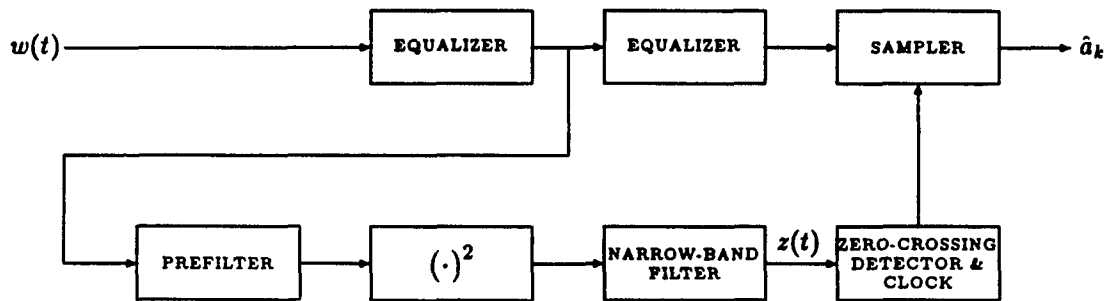


Figure 2.6: A symbol timing recovery circuit for baseband PAM signal.

hold in order to meet power and bandwidth requirements. It has been recognized that the above scheme will still work if a square-law device is inserted before the narrow-band filter [23].

From (2.3), we see that the output of the reading head can be treated as a PAM signal. This suggests that a reasonable approach to timing recovery is the square-and-filter method which has been widely used in PAM systems. In the discussion to follow, we shall first study the noise-free case for ease of terminology and notation. The effects of additive noise will be studied in Chapter 4.

Figure 2.6 shows a general square-law symbol timing recovery circuit. For simplicity, we assume that no equalization or prefiltering is done so that with an uncoded data sequence, the input signal to the square-law device is given by

$$w(t) = \sum_k a_k g(t - kT - \tau) \quad (2.4)$$

where a_k is a zero-mean stationary sequence with independent elements in $\{-1, 0, +1\}$ and $g(t)$ is the signaling pulse. We assume that $g(t)$ is so defined that the best

sampling instants are at $t = kT + \tau$. The objective here is to recover a good estimate of τ .

It is easily shown that $w(t)$ is a zero-mean cyclostationary process with no periodic components present. However, the output of the square-law device can be interpreted as a new signal with modified pulse shape and a new pulse amplitude sequence which has a nonzero mean value. Since $E[a_k a_m] = 0$ for $k \neq m$,

$$E[w^2(t)] = \sum_k \sum_m E[a_k a_m] g(t - kT - \tau) g(t - mT - \tau) \quad (2.5)$$

$$= \sigma_a^2 \sum_k g^2(t - kT - \tau) \quad (2.6)$$

where $\sigma_a^2 = E[a_k^2]$. Using the Poisson Sum Formula [24]

$$\sum_k b(t - kT) = \frac{1}{T} \sum_l B\left(\frac{l}{T}\right) e^{\frac{j2\pi l t}{T}} \quad (2.7)$$

where $b(t)$ and $B(f)$ are a Fourier transform pair, we can express (2.6) in the more convenient form of a Fourier series whose coefficients are given by the Fourier transform of $g^2(t)$,

$$E[w^2(t)] = \frac{\sigma_a^2}{T} \sum_l A_l e^{\frac{j2\pi l}{T}(t-\tau)} \quad (2.8)$$

where

$$A_l = \int_{-\infty}^{\infty} G\left(\frac{l}{T} - \lambda\right) G(\lambda) d\lambda \quad (2.9)$$

and $G(f)$ is the Fourier transform of $g(t)$.

Now if the data pulse has a bandwidth in excess of the minimum required bandwidth of $1/(2T)$, then $E[w^2(t)]$ has periodically varying components due to the $l \neq 0$ terms in (2.8). If the bandwidth in excess of the minimum required bandwidth is less

than 100%, then $G(f) = 0$ for $|f| > 1/T$ [25]. This means that $E[w^2(t)]$ contains a sinusoidal component at a frequency of $1/T$. This suggests that an effective timing recovery circuit can be implemented by the previously discussed square-and-filter arrangement, where the center frequency of the narrow-band filter is tuned to the pulse repetition frequency, $1/T$.

Due to the nonzero phase of any implementable narrow-band filter, there is an offset in the zero-crossings of the timing wave from the desired sampling instants, which must be compensated for by the clocking circuitry. After proper adjustment, the positive-going zero-crossings of $z(t)$, the filtered timing wave, can be used as indicators of the proper sampling instants for the detector to recover the code sequences.

2.3 Two-Dimensional Modulation Codes

In existing magnetic tape recording systems, each track is encoded independently with the same run-length-limited modulation code being used on each track. The idea of *two-dimensional* modulation codes was recently introduced by Marcellin and Weber [14]. With two-dimensional modulation codes, multiple tracks are encoded simultaneously.

With this technique, it is possible to achieve substantial increase in storage capacity. The reason for this is that the clocking constraint k can be somewhat relaxed. The d constraint must be satisfied in each track independently to avoid intersymbol interference, however, the k constraint can be satisfied in a “joint” manner. Since

multiple tracks are read out in parallel, it is not necessary for each track to provide complete clocking information as long as timing information can be recovered from some subset of the tracks being read.

As an example, consider grouping tracks into pairs and designing a single two-dimensional modulation code that will be used for each pair of tracks. The code must be designed to satisfy the d constraint in each track, but the k constraint can be satisfied in a joint manner. That is, there should not be more than k code bit intervals during which both tracks contain a 0 bit simultaneously. This requirement will ensure that no more than k code bit intervals pass without a transition in at least one track of the pair.

As a specific example, consider a two-dimensional $(1, 3)$ code for a pair of tracks. In each track there should be at least one 0 between consecutive 1's. Also, if the two tracks are logically "OR"ed together bit-by-bit, the resulting sequence should have no more than three 0's between consecutive 1's. An example of a pair of sequences satisfying the two-dimensional $(1, 3)$ constraints is shown below.

Track 1 001010000000100100101000

Track 2 010000010001010000000001

Each sequence satisfies the $d = 1$ constraint independently, but neither sequence satisfies the $k = 3$ constraint. However, the sequences satisfy the constraint (as a pair) that no more than three bit intervals should occur without a 1 in at least one of the sequences.

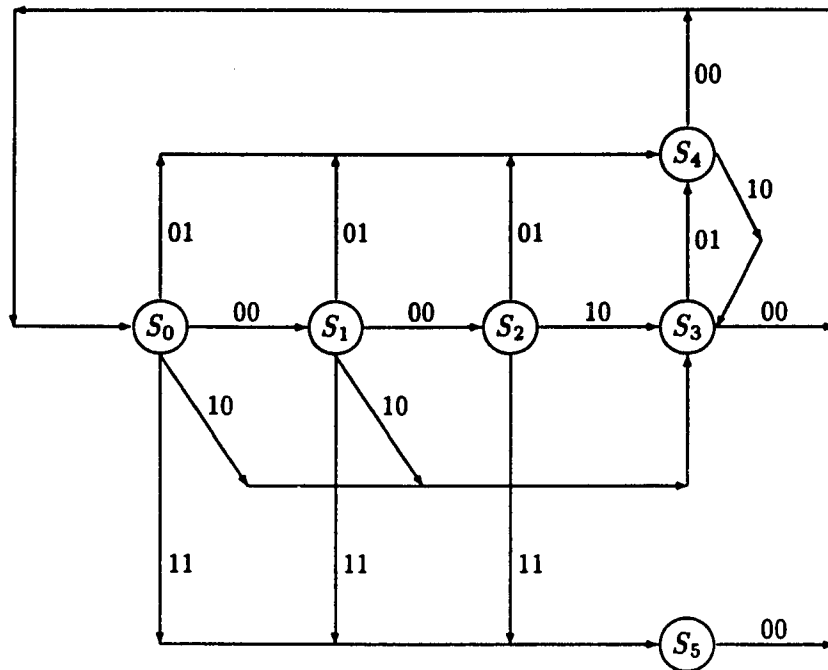


Figure 2.7: A finite-state transition diagram for the two-dimensional (1,3) code.

We now derive the capacity of a system satisfying the two-dimensional (1,3) constraints described above. Following the formulation of Blahut [17], we derive an FSTD representing all possible code sequences. Such a diagram is shown in Figure 2.7. In this figure, the state transitions are labeled with output bit pairs of the code. The first bit of each pair specifies the code bit for track 1 and the second bit specifies the code bit for track 2. As an example, consider the state labeled S_4 . Close examination of this state will show that the previous code bit pair was 01. Clearly, the next code bit pair must be either 00 or 10. If the next code bit pair were 01 or 11, there would be two consecutive 1's in the second track which would violate the d constraint. Hence, only two arrows leave state S_4 ; one labeled 10 and one labeled 00.

The state transition matrix for the FSTD of Figure 2.7 is given by

$$T = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and the capacity is $C_2(1,3) \approx 1.360$ where the subscript 2 indicates that the capacity is per pair of tracks. The improvement in storage capacity of this scheme over one in which a (1,3) code is used independently for each track is given by

$$\frac{C_2(1,3) - 2C(1,3)}{2C(1,3)} \times 100\% \approx 23.3\%.$$

One can see that this scheme yields a better storage capacity, and it should be obvious that the efficiency can be further improved by encoding more than two tracks in parallel.

The capacity for a very specific example of a two-dimensional modulation code was given to demonstrate the potential to increase the storage capacity. This example was concerned with encoding data in a manner such that timing information could be derived from any pair of tracks. In [14], capacities are derived for arbitrary (d, k) constraints for recording n tracks in parallel.

2.4 Phase-Locked Loops

As mentioned in previous chapters, with two-dimensional modulation codes, multiple tracks are encoded simultaneously. Therefore, when it comes to detection of these tracks, the detector must be able to handle multiple tracks in parallel. In other

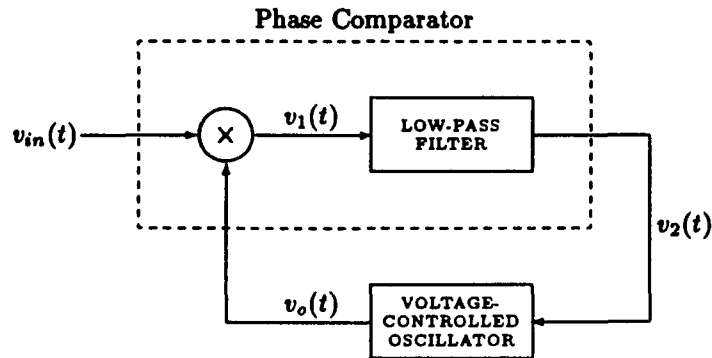


Figure 2.8: A general structure of a phase-locked loop.

words, timing information has to be recovered by some kind of operation which is able to take care of more than one track at a time. A practical way to attack this problem is by using a weighted sum of these tracks to extract timing information. This is possible because a weighted sum of these PAM look-alike signals is just another PAM signal, provided that signal transitions occur at the same time. Clearly, we need a way to make sure that these tracks are in phase before any timing recovery scheme takes place.

The idea of how to align the transitions of these tracks is inspired by the principle of the phase-locked loop (PLL). Figure 2.8 shows a basic PLL. It is a feedback system which consists of a voltage-controlled oscillator (VCO) (which is a sine-wave generator whose frequency is determined by a voltage applied to it from an external source), and a phase comparator (which consists of a multiplier and a low-pass filter). The basic aim of a PLL is to *lock* or *synchronize* the instantaneous phase of the VCO output to the instantaneous phase of the input signal. For this purpose, the PLL must perform *phase comparison*.

We assume that initially we have adjusted the VCO so that when the control voltage $v_2(t)$ is zero, the frequency of the VCO is precisely set at f_c , the nominal frequency of the input signal. Suppose that the input signal applied to the PLL is defined by

$$v_{in}(t) = A_{in} \cos(2\pi f_c t - \theta_{in}) \quad (2.10)$$

where A_{in} and θ_{in} are the amplitude and the constant phase of the signal, respectively. Also, let the VCO output be defined by

$$v_o(t) = A_o \cos(2\pi f_c t - \theta_o(t)). \quad (2.11)$$

With a control voltage $v_2(t)$ applied to the VCO input, we have

$$\theta_o(t) = 2\pi k_o \int_0^t v_2(t) dt \quad (2.12)$$

where k_o is the frequency sensitivity of the VCO. The incoming $v_{in}(t)$ and the VCO output $v_o(t)$ are then applied to the multiplier, producing a high frequency component given by $(1/2)k_m A_{in} A_o \cos(4\pi f_c t - \theta_{in} - \theta_o(t))$, and a low frequency component given by $(1/2)k_m A_{in} A_o \cos(\theta_{in} - \theta_o(t))$, where k_m is the multiplier gain. The high frequency component is eliminated by the low-pass filter. Therefore, discarding the high frequency component, the input to the low-pass filter is given by

$$v_1(t) = \frac{1}{2} k_m A_{in} A_o \cos(\epsilon(t) - 90^\circ) \quad (2.13)$$

$$= \frac{1}{2} k_m A_{in} A_o \sin(\epsilon(t)) \quad (2.14)$$

where $\epsilon(t) = \theta_{in} - \theta_o(t) + 90^\circ$. We interpret $\epsilon(t)$ as the *phase error*, which is a critical parameter in a PLL. The low-pass filter then operates on $v_1(t)$ to produce the output

$$v_2(t) = \int_{-\infty}^{\infty} v_1(\lambda)h(t - \lambda)d\lambda \quad (2.15)$$

where $h(t)$ is the impulse response of the filter and $v_2(t)$ is applied to adjust the frequency of the VCO output for continuous phase comparison.

For example, suppose we have a situation in which $v_{in}(t)$ is leading $v_o(t)$ by less than 90 degrees; in other words, $0 \leq \theta_o(t) - \theta_{in} < 90^\circ$. This implies that $\epsilon(t)$ is positive and will generate a positive $v_1(t)$ which will in turn generate a positive $v_2(t)$. Notice that from (2.12) a positive $v_2(t)$ will result in an increasing $\theta_o(t)$ and thus a decreasing $\epsilon(t)$. A similar analysis shows that, if $v_o(t)$ is leading $v_{in}(t)$ then $\epsilon(t) < 0$, which yields negative values for $v_1(t)$ and $v_2(t)$, which in turn results in a decreasing $\theta_o(t)$ and an increasing $\epsilon(t)$. Thus, the PLL will drive $\theta_o(t)$ toward the "locked" condition of $\epsilon(t) = 0$, provided that the initial value of $|\theta_o(t) - \theta_{in}|$ is less than 90° .

After studying the principles of the PLL, we see that its main characteristic is the ability to drive the phase difference between two signals to a predetermined value and lock the phase once this is achieved. The problem of track alignment we discussed previously is basically a matter of phase synchronization. Hence, the idea utilized by the PLL will be carried over.

CHAPTER 3

Track Alignment

3.1 System Operation

For ease of terminology and notation, we shall consider only two tracks $w_1(t)$ and $w_2(t)$ with phases θ_1 and θ_2 , respectively. However, it should be clear that the discussion to follow can be extended to multiple tracks in parallel.

We are going to achieve track alignment by operating on deterministic preambles of these two signals. The argument here is that once the preambles are in phase, the rest of the tracks with random data will also be in phase. It is obvious at this point that we are assuming that the phase offset between these two signals is constant everywhere.

The actual code bit period T is unknown to the synchronizer, but the nominal code bit period T' and its tolerance a are known. The circuit which is used to align the transitions of these tracks is designed in a way so that a fixed but unknown code bit period is allowed as long as the phase offset between $w_1(t)$ and $w_2(t)$ is less than or equal to $T_{po} = (d + 1)(T' - a)/2$, where we assume that $w_1(t)$ and $w_2(t)$ satisfy (d, k) constraints.

The circuit which is used to align the transitions of the tracks is shown in Figure 3.1. $w_1(t)$ and $w_2(t)$ are deterministic preambles which can be written as

$$w_1(t) = \sum_k a_k g(t - kT - \theta_1) \quad (3.1)$$

$$w_2(t) = \sum_k a_k g(t - kT - \theta_2) \quad (3.2)$$

where a_k takes on alternate 1 and -1 separated by d 0's and $g(t)$ is the Lorentzian pulse. Phase synchronization is achieved by passing $w_2(t)$ through a variable delay element. If $w_1(t)$ is initially lagging $w_2(t)$, we can always delay $w_2(t)$ until they are in phase. What if $w_1(t)$ is initially leading $w_2(t)$? Since there is no such thing as a negative delay, it is impossible to shift $w_2(t)$ in the negative direction to align the transitions. This problem can be solved by initially delaying $w_1(t)$ by a fixed amount $T_{pf} = (d + 1)(T' + a)/2$, which is slightly bigger than the assumed maximum offset so that $x_1(t)$, the delayed version of $w_1(t)$, will always be lagging $w_2(t)$.

Our next task is to delay $w_2(t)$ by an appropriate amount. As mentioned above, this is achieved by using a variable delay element whose delay is a function of the input voltage $v(t)$ which is assumed to be zero initially. The characteristic of such a device is shown in Figure 3.2. The linearity shown is actually not a necessity. The characteristic curve can assume any shape as long as it is monotonic increasing from 0 to $2T_{pf}$ with y-intercept at T_{pf} . The reasons for these specifications will become clear shortly.

The delayed versions of $w_1(t)$ and $w_2(t)$ can be expressed as

$$x_1(t) = w_1(t - T_{pf})$$

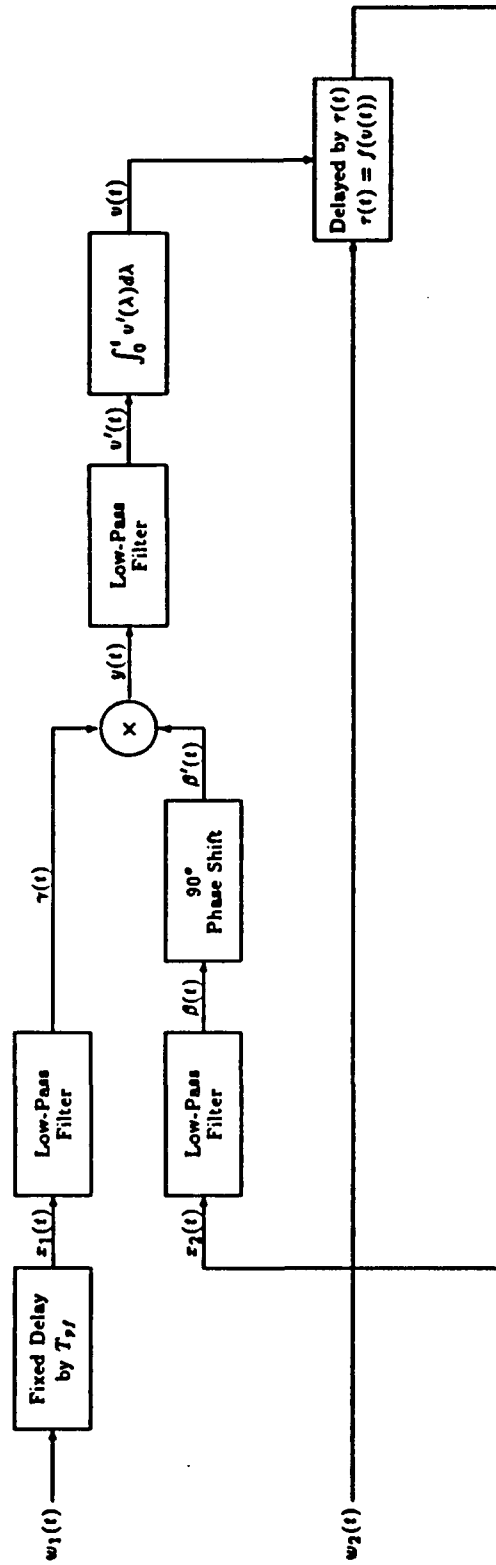


Figure 3.1: The circuit for track alignment.

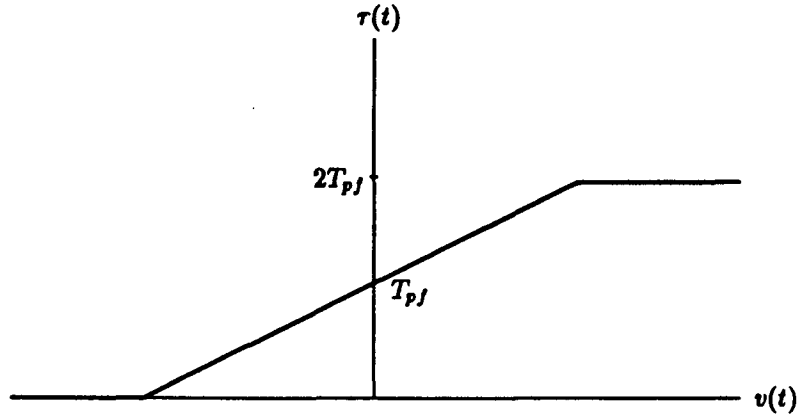


Figure 3.2: The characteristic of the variable delay element.

$$= \sum_k a_k g(t - kT - \phi_1) \quad (3.3)$$

and

$$\begin{aligned} x_2(t) &= w_2(t - \tau(t)) \\ &= \sum_k a_k g(t - kT - \phi_2(t)) \end{aligned} \quad (3.4)$$

where

$$\phi_1 = T_{pf} + \theta_1 \quad (3.5)$$

and

$$\phi_2(t) = \tau(t) + \theta_2. \quad (3.6)$$

Our goal is then to alter $\tau(t)$ until synchronization between $x_1(t)$ and $x_2(t)$ is achieved.

In other words, our task is completed when $\phi_2(t) = \phi_1$.

In order to generate a sinusoidal signal for the purpose of phase comparison, all harmonic components (except the fundamental) are removed from $x_1(t)$ and $x_2(t)$ by

(identical) low-pass filters to yield

$$\gamma(t) = A \cos(2\pi f_c t - \phi_1 - \delta) \quad (3.7)$$

and

$$\beta(t) = A \cos(2\pi f_c t - \phi_2(t) - \delta) \quad (3.8)$$

where δ is the phase and A is the amplitude of the low-pass filters at frequency $f_c = 1/(2(d+1)T)$. Since a traditional PLL locks the phase when the reference signal is 90 degrees ahead of the other, and we are interested in a zero phase difference between $x_1(t)$ and $x_2(t)$, a 90-degree phase-shift is introduced into $\beta(t)$ to obtain

$$\beta'(t) = A \sin(2\pi f_c t - \phi_2(t) - \delta). \quad (3.9)$$

The output of the multiplier can be expressed as

$$y(t) = \frac{1}{2} A^2 [\sin(\phi_1 - \phi_2(t)) + \sin(4\pi f_c t - \phi_1 - \phi_2(t) - 2\delta)]. \quad (3.10)$$

Again, since it is only the phase difference that we are interested in, another low-pass filter is used to get rid of the high frequency term to yield the control signal

$$v'(t) = \frac{1}{2} A^2 \sin(\phi_1 - \phi_2(t)). \quad (3.11)$$

Instead of using a VCO as in a traditional PLL, a combination of an integrator and a variable delay element is used at this point. The integrator performs the function of generating a signal $v(t)$ whose nature (increasing, decreasing, or constant) depends on the sign (positive, negative, or zero) of $v'(t)$. $v(t)$ is then applied to the variable delay element to adjust the delay for $x_2(t)$.

As an example, suppose $x_1(t)$ is initially lagging $x_2(t)$. Then ϕ_1 is larger than $\phi_2(t)$ which yields, by (3.11), a positive value of $v'(t)$ and hence, an increasing $v(t)$. This causes $\tau(t)$ to increase which in turn increases $\phi_2(t)$ (see (3.6)). Similarly, if $\phi_2(t) > \phi_1$, then $v'(t) < 0$ which causes a decrease in $v(t)$, $\tau(t)$, and $\phi_2(t)$. Finally, if $\phi_2(t) = \phi_1$, $v'(t) = 0$ which results in $v(t)$, $\tau(t)$, and $\phi_2(t)$ remaining constant, as desired. The behavior of the circuit is summarized as follows:

$$\begin{aligned}
 \phi_2(t) < \phi_1 &\Rightarrow v'(t) > 0 \\
 &\Rightarrow v(t) \text{ gets bigger} \\
 &\Rightarrow \phi_2(t) \text{ gets bigger} \\
 \\
 \phi_2(t) = \phi_1 &\Rightarrow v'(t) = 0 \\
 &\Rightarrow v(t) \text{ stays the same} \\
 &\Rightarrow \phi_2(t) \text{ stays the same} \\
 \\
 \phi_2(t) > \phi_1 &\Rightarrow v'(t) < 0 \\
 &\Rightarrow v(t) \text{ gets smaller} \\
 &\Rightarrow \phi_2(t) \text{ gets smaller.}
 \end{aligned}$$

3.2 Example

An easy way to further understand how the system operates is to examine a specific example. For the following demonstration, we assume that deterministic preambles of $w_1(t)$ and $w_2(t)$ which satisfy a (1,3) constraint are used. Thus a_k will take on alternate 1 and -1 separated by one 0. A Lorentzian pulse with $T_{50} = 1.0$ second is chosen to be the step response, along with a code bit period of $T = 1.0$ second and a tolerance of 10%. This yields a pulse density of 1.0 code bit/ T_{50} .

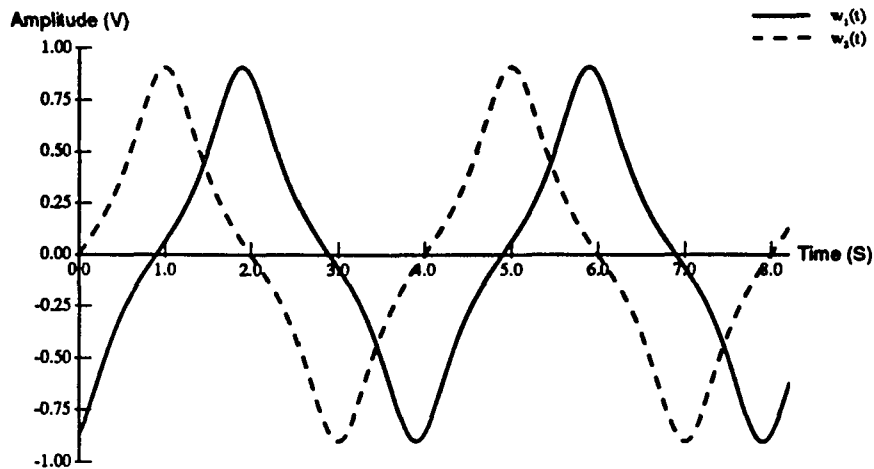


Figure 3.3: The input signals.

Let us assume that $w_2(t)$ is initially leading $w_1(t)$ by an amount equal to the maximum allowable offset, which is $T_{po} = 0.9$ seconds. Figure 3.3 is a graph of these two signals for the first 8 seconds of operation. At time zero, $w_1(t)$ and $w_2(t)$ are delayed by $T_{pf} = 1.1$ seconds to obtain $x_1(t)$ and $x_2(t)$ as shown in Figure 3.4. Each of these signals is fed into a low-pass filter to generate sinusoidal waves of the same frequencies for the purpose of phase comparison. $\beta(t)$ is further delayed by 90 degrees to make sure that the phase of $x_2(t)$ will be locked when the phase difference between $x_1(t)$ and $x_2(t)$ is zero.

Figure 3.5 shows the inputs and output of the multiplier. The signal $v'(t)$ in Figure 3.6 is obtained by low-pass filtering $y(t)$. The curve for $v'(t)$ crosses the time axis at about the 47th second. At this point, the delay generated has become too large. As mentioned before, a negative $v'(t)$ causes the integrator and the variable delay element to reduce the delay and eventually, the phase difference is driven to

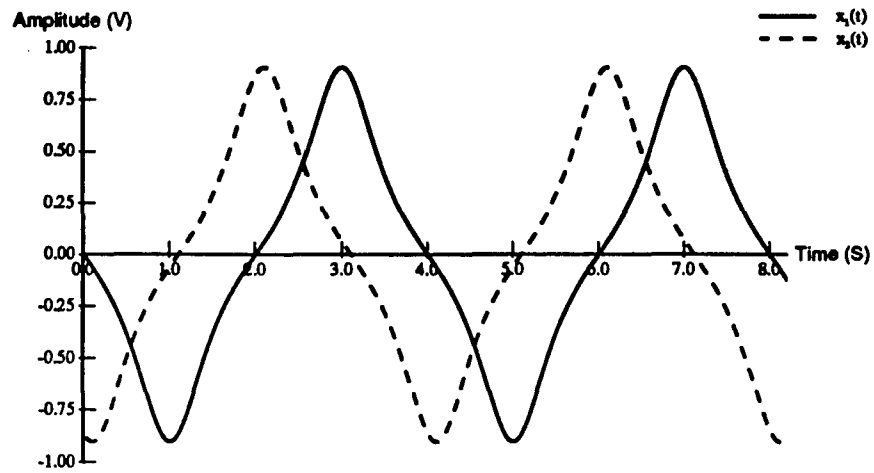


Figure 3.4: The delayed input signals.

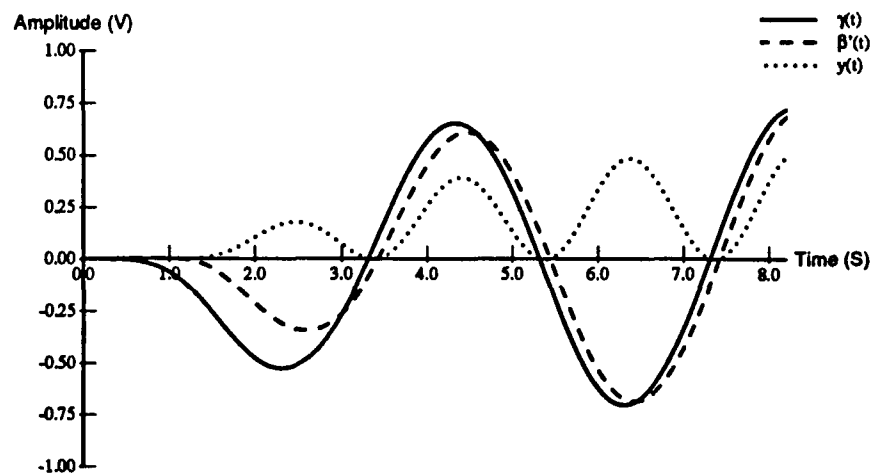


Figure 3.5: The inputs and output of the multiplier.

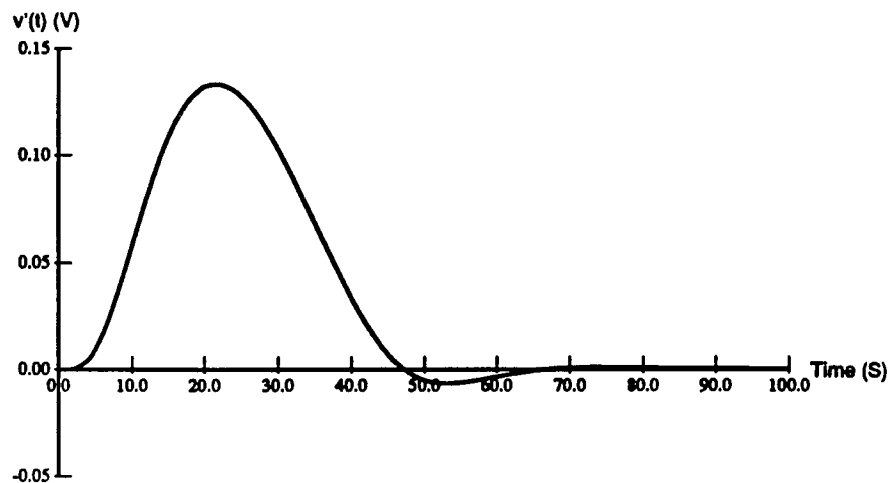


Figure 3.6: The output of the low-pass filter.

zero. A graph of $v(t)$ is shown in Figure 3.7. The curve is increasing until the time when $v'(t)$ goes negative, then it starts to decrease to reduce the corresponding delay. Notice that the curve stays constant as soon as the tracks are in phase.

We conclude this chapter with a graph of the control signal $v'(t)$ for different initial offsets. In Figure 3.8, one can see that as long as the phase offset is within the assumed range, the system drives the phase difference to zero.

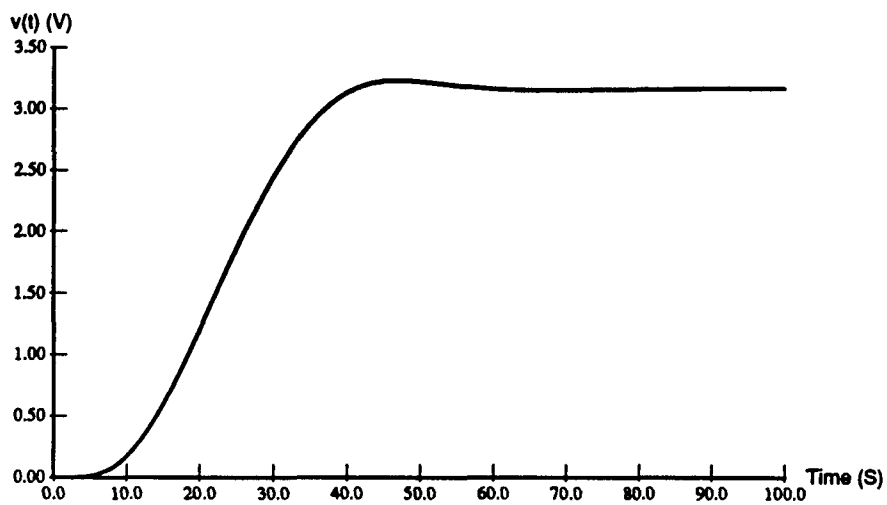


Figure 3.7: The output of the integrator.

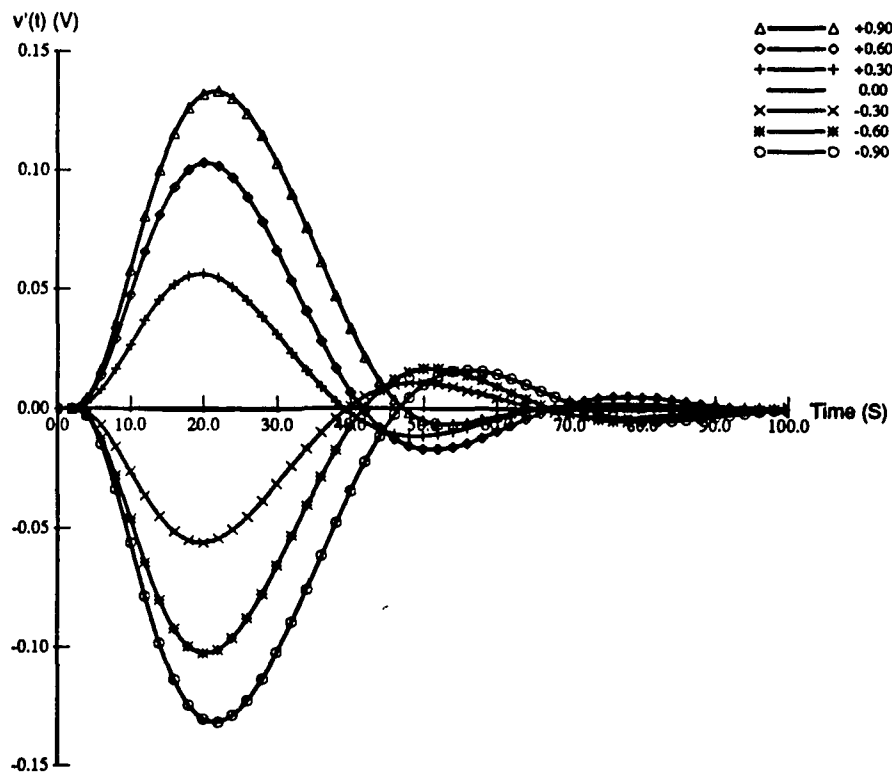


Figure 3.8: A family of $v'(t)$ for different initial offsets.

CHAPTER 4

Performance of Timing Recovery Scheme

Chapter 3 presented a method to align the transitions of two tracks during periods when known data sequences are read, such as during part of a preamble of a data packet. In this chapter, a timing recovery scheme for the resulting time-synchronous PAM look-alike signals will be presented. We shall employ the symbol timing recovery circuit discussed in Chapter 2.

4.1 Synchronizer Circuit

Figure 4.1 shows a diagram of the synchronizer used in our study. $x_1(t)$ and $x_2(t)$ are signals whose transitions are aligned, and $n_1(t)$ and $n_2(t)$ are independent white Gaussian noise processes. The corrupted signals are multiplied by two different constants a_1 and a_2 . We have chosen $a_1 = 1$ and $a_2 = 2$ to make sure that the sum

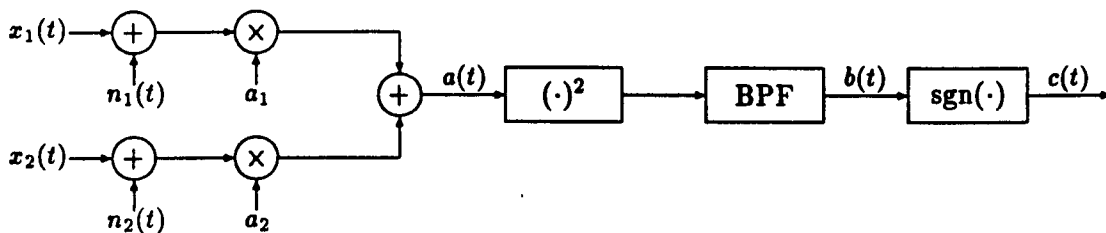


Figure 4.1: The square-and-filter synchronizer.

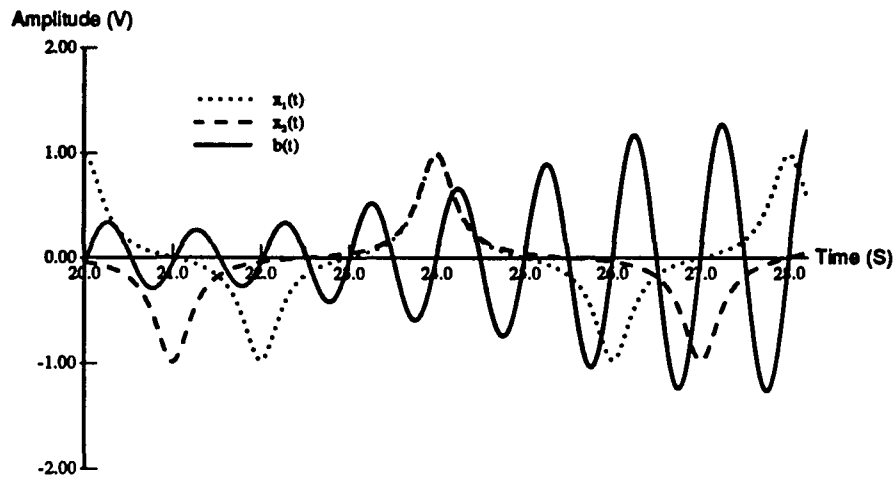


Figure 4.2: The input signals and timing wave.

will not be zero even when one signal is the exact opposite of the other. This ensures that the summing process does not destroy the self-clocking capability provided by the (d, k) code.

The signal $a(t)$ can be considered as a new PAM signal on which a square-and-filter timing recovery scheme is employed. Finally, a hard-limiter is used on the timing wave $b(t)$ to create a square-wave clock. The BPF in our simulations was phase compensated so that the positive-going transitions of $b(t)$ correspond to the ideal sampling instants. Figure 4.2 shows two input signals (each encoded using the Modified Frequency Modulation (MFM) code, as described in the following subsection), and the resulting timing wave.

Table 4.1: Code rules for the Modified Frequency Modulation codes.

Source bits	Code bits
1	01
(1)0	00
(0)0	10

4.2 Simulation Results

Often communication systems are too complicated to analyze completely by using only analytical techniques. In these cases, digital computer simulations are used to provide a practical approach to system design and analysis.

Since uncoded signals are very vulnerable to intersymbol interference and timing jitter, they are seldom directly recorded on magnetic media. In our simulations, the Modified Frequency Modulation (MFM) code [26] is employed to encode the original source sequences individually, so that we can get an idea of deterioration due to having the synchronizer recover timing information from multiple tracks. The MFM code is a (1,3) code. Each source bit is separately encoded as two code bits. The specific code rules are listed in Table 4.1. A 1 is encoded as 01 while a 0 is encoded as 00 or 10, depending on whether it is preceded by a 1 or a 0. For example, the source sequence 1010011 will be encoded to become 01000100100101.

$x_1(t)$ and $x_2(t)$ are then generated directly as the superposition of time-shifted versions of the step response given by

$$x_i(t) = \sum_k a_{k,i} g(t - kT - \tau) \quad (4.1)$$

where the random sequence a_{k_i} takes on values of -1 , 0 , and $+1$ (alternating $+1$, -1 for code bits taking a value of '1' and 0 for code bits taking a value of '0'). The code bit period T is chosen to be 1.0 second and we perform our simulations using sampled versions of all signals with a sampling frequency of 100 Hz. This value was chosen so that timing errors could be measured very precisely. Each simulation was performed using 25,000 source bits (50,000 code bits).

In the results to be described, relative root-mean-square (RMS) timing jitter is compared against signal-to-noise ratio (SNR) for different recording densities measured in code bits per T_{50} . The relative RMS timing jitter is the ratio of the square root of the average value of $(\tau - \hat{\tau})^2$ to the code bit period T , and the SNR is defined by

$$\text{SNR} = 10 \log_{10} \frac{E_a}{N_0} \quad (4.2)$$

where E_a is the average energy per code bit and N_0 is twice the (two-sided) power spectral density of the noise.

A performance comparison is made between the traditional timing recovery scheme (single-track) and our proposed scheme (double-track) in Figure 4.3. This figure plots relative RMS timing jitter versus SNR for a system operating at 0.5 code bits/ T_{50} and reveals the fact that we are indeed paying a price by having the synchronizer recover timing information from multiple tracks in parallel. This is because the signal $a(t)$ fed into the square-law device looks like a 7-ary PAM signal (taking values in $(-3, -2, -1, 0, +1, +2, +3)$) rather than a 3-ary PAM signal (taking values in

(-1, 0, +1)) [27]. Figures 4.4 through 4.6 show timing recovery comparisons at 1.0, 1.5, and 2.0 code bits/ T_{50} , respectively. As might be expected, when the pulse density is increased, the intersymbol interference gets worse and the timing jitter increases.

We have treated several aspects of the timing recovery problem. One aspect that has been omitted is the effect of timing jitter on system error probability. A discussion on this subject requires the design and evaluation of an optimal detector, which we have not attempted at the current stage of our research. However, if we assume a simple "sample-and-threshold" detector, we can get a feel for the performance penalty incurred by the use of our multi-track scheme.

Figure 4.7 shows an eye diagram¹ of the read-out waveform for a system using the MFM code at a pulse density of 1.0 code bit/ T_{50} . For an idea of the scale in this figure, the dashed lines delineate a window that is 0.1 code bits wide. This window contains all sampling instants corresponding to a timing error of 0.05 or less. Since the envelope of the eye diagram is relatively flat near the optimal sampling instant, small timing errors do not significantly reduce the noise margin. Therefore, we expect that even the maximum increase in the relative timing jitter caused by the multi-track scheme (about 0.02) will not significantly affect system error probability. In cases where the increased jitter does create a problem, equalization can be used to widen the eye diagram [28].

¹The presence of a small amount of intersymbol interference is apparent in the lack of a well defined '0' symbol. In a practical system, an equalizer would be used to further "open" the lower and upper "eyes."

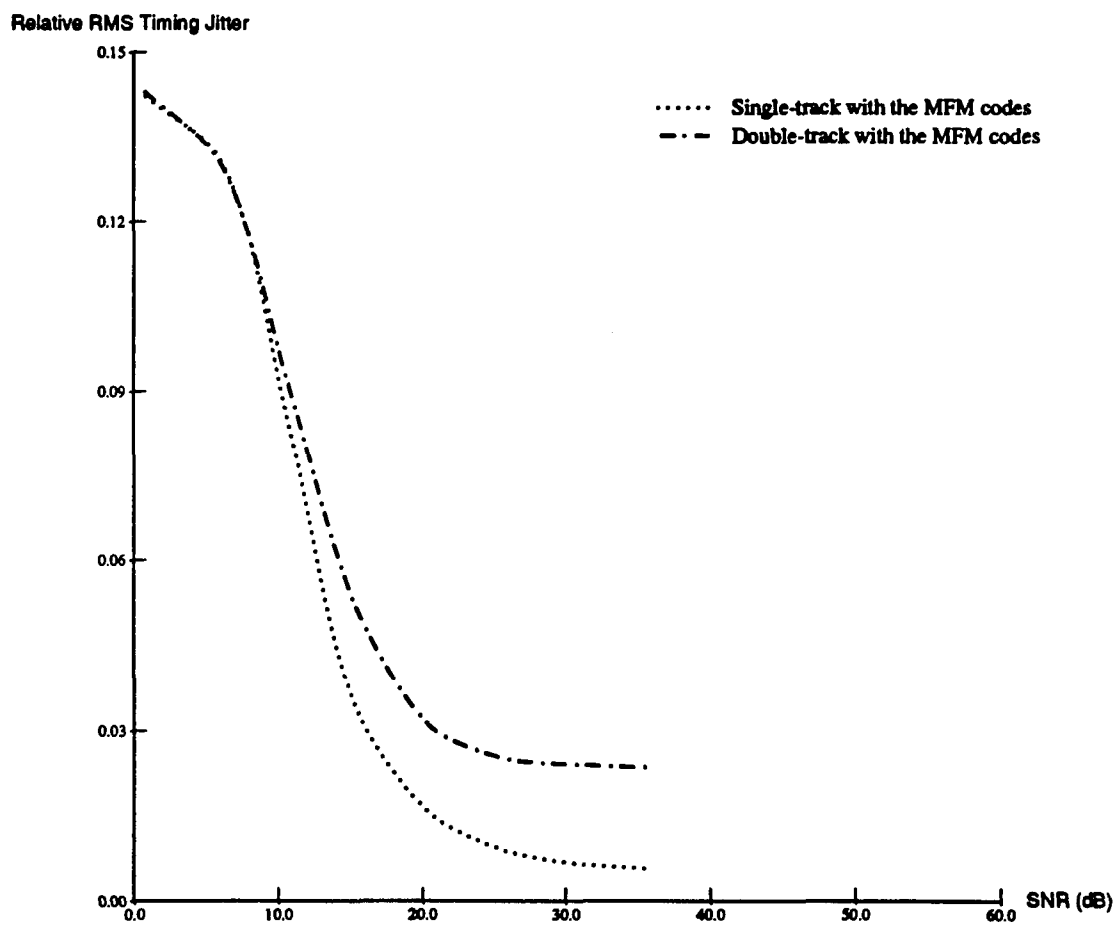


Figure 4.3: Timing Jitter Performance at 0.5 code bits/ T_{50} .

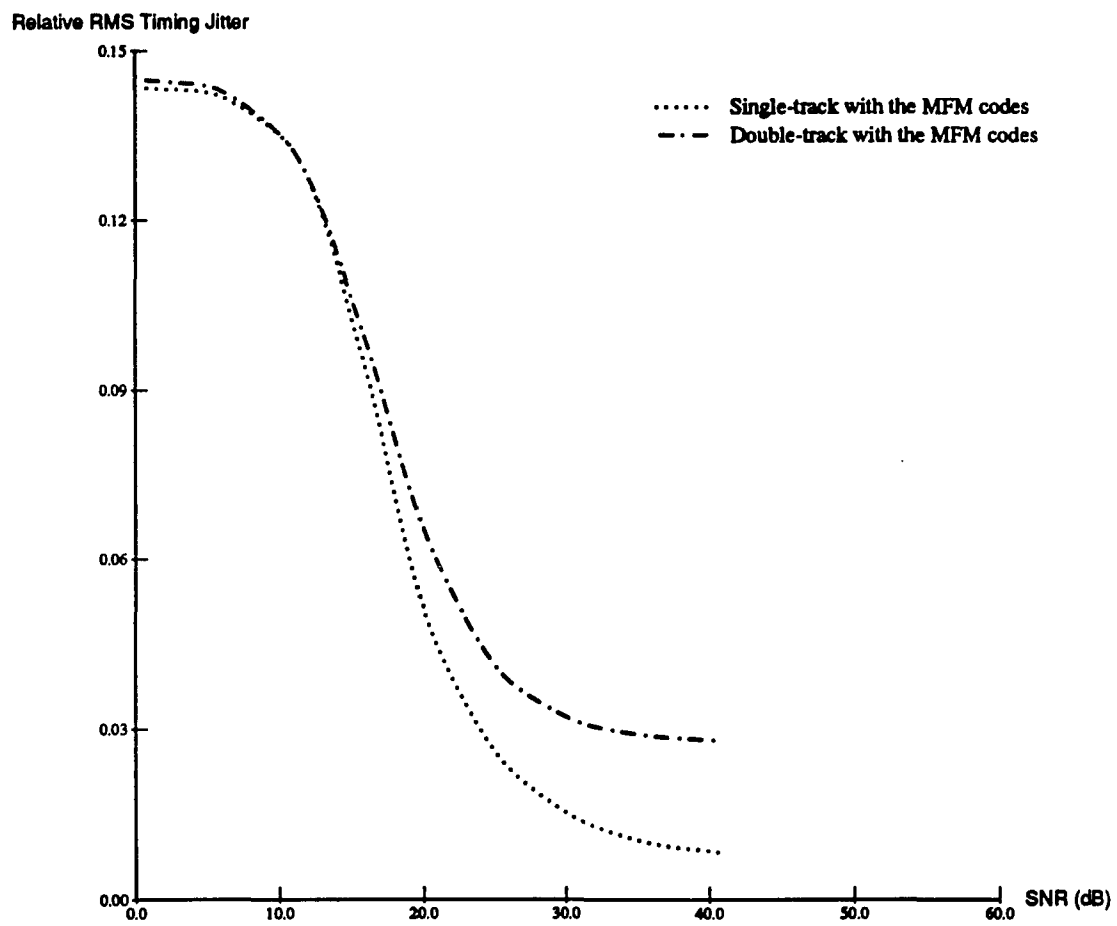


Figure 4.4: Timing Jitter Performance at 1.0 code bit/ T_{50} .

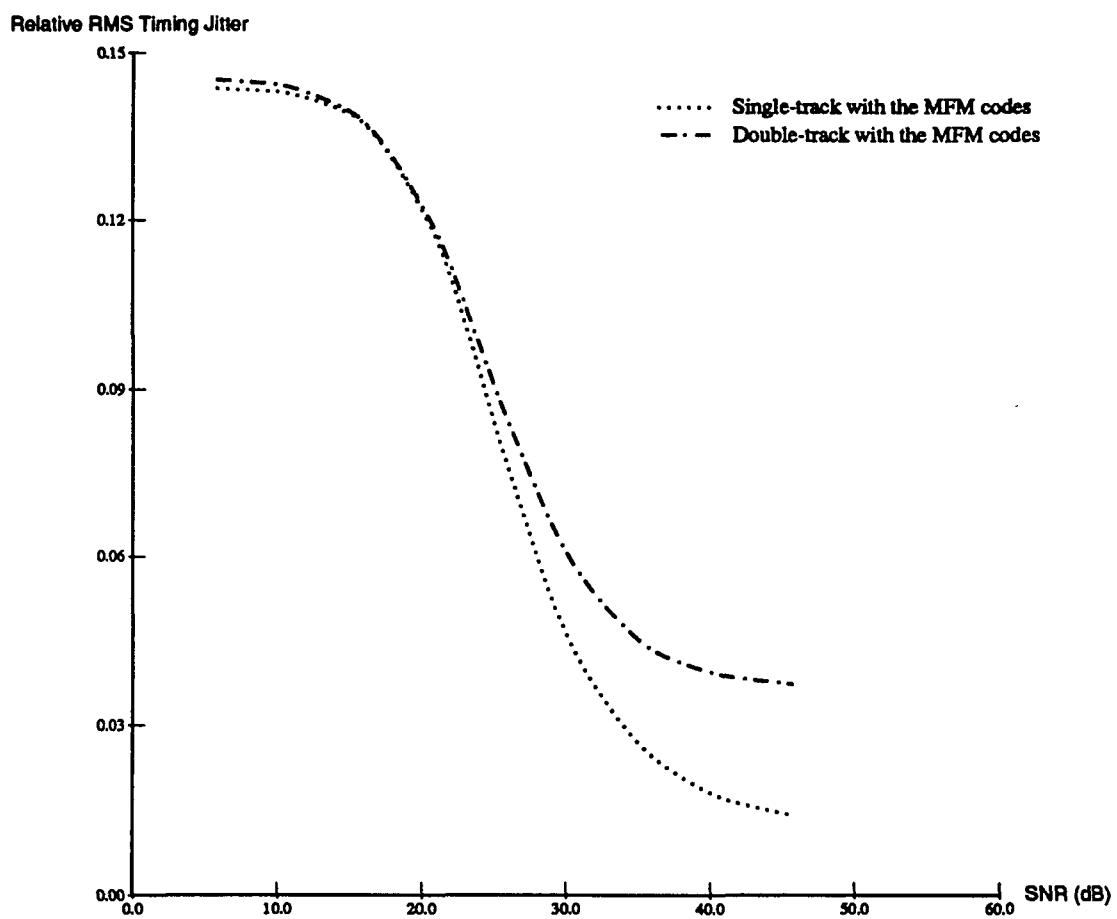


Figure 4.5: Timing Jitter Performance at 1.5 code bits/ T_{50} .

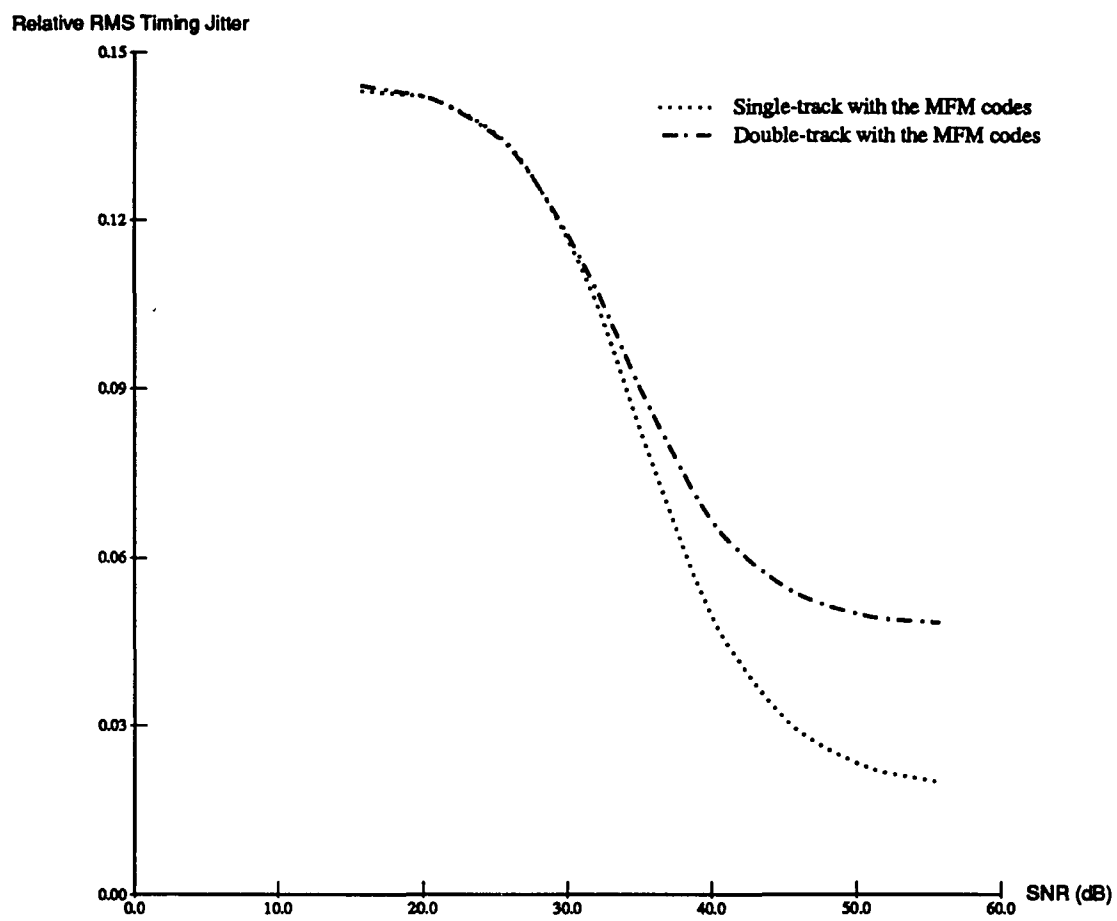


Figure 4.6: Timing Jitter Performance at 2.0 code bits/ T_{50} .

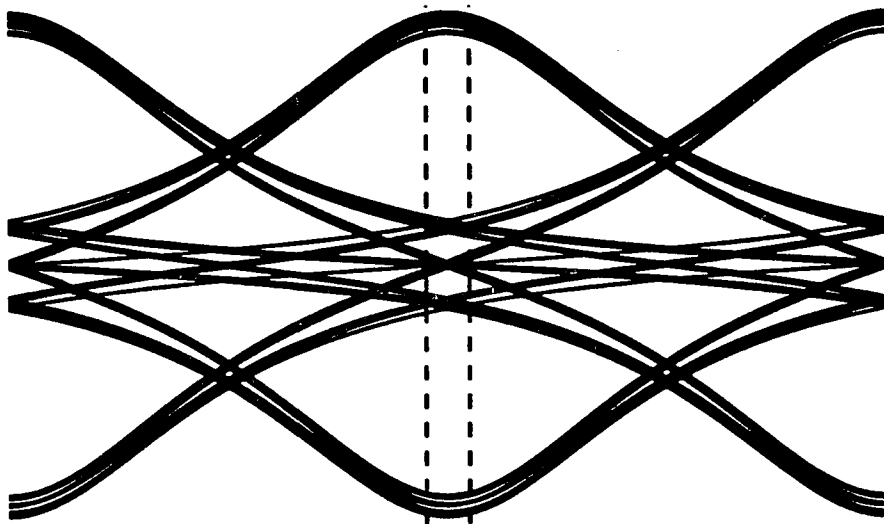


Figure 4.7: An eye diagram at 1.0 code bit/ T_{50} .

CHAPTER 5

Summary

A digital magnetic recording system stores data in the form of binary symbols and then permits retrieval of information at a later time. There are only two unique states of magnetization which are required for the realization of different signals. This is an advantageous feature of digital recording which provides highly reliable performance for information storage.

The magnetic recording process essentially consists of two distinct operations, data writing and data reproduction. The writing circuit records information on the media in the form of magnetic transitions and the reading circuit reads a degraded signal back from the media. In order to obtain reliable results, it is necessary to modify the data signal before it is written and again after it is read from the reading head.

Modulation coding is the modification made to the data source to facilitate its passage through the media. Many frequently employed modulation codes can be classified as run-length-limited codes. These codes are binary codes in which the separation of adjacent magnetic transitions has an upper bound to provide sufficient signal energy to recover timing information as well as a lower bound to control intersymbol interference.

In an effort to provide increased storage capacities, a new class of run-length-limited codes was recently introduced. These codes are called two-dimensional modulation codes. Two-dimensional modulation codes provide substantial increase in data storage capacity for multi-track recording systems by operating on multiple tracks in parallel. The work reported in this thesis was an investigation into timing recovery schemes for two-dimensionally modulated signals.

Because the signals are modulated two-dimensionally, the synchronizer must be able to recover timing information from multiple tracks. We used the fact that the output of the writing head could be treated as a PAM signal. Knowing that the sum of PAM signals is just another PAM signal (provided that these signals have transitions at the same time), a traditional square-and-filter timing recovery scheme can then be applied to generate a timing wave whose positive-going zero-crossings indicate the proper sampling instants for the detector to recover the coded information.

In Chapter 3, we proposed a method which utilized deterministic preambles to align the transitions of these PAM look-alike signals based on the principle of the phase-locked loop, while simulations were employed in Chapter 4 to determine the performance of the square-and-filter timing recovery scheme for multi-track recording. Relative RMS timing jitter was calculated and plotted against various signal-to-noise ratios for the traditional single-track scheme and the proposed double-track scheme. In each case, the Modified Frequency Modulation codes were used to modulate each signal individually to investigate the degradation of timing jitter due to the summing

process alone. It was shown that the single-track scheme has better timing recovery performance than the double-track scheme. However, the increased storage capacities obtained by using two-dimensional modulation codes may be sufficient to offset this small penalty in timing jitter.

REFERENCES

- [1] G. Ungerboeck and J. K. Wolf, "Trellis coding for partial-response channels," *IEEE Trans. Commun.*, vol. COM-34, pp. 765-773, Aug. 1986.
- [2] H. Kobayashi and D. T. Tang, "Application of partial-response channel coding to magnetic recording systems," *IBM J. Res. Dev.*, pp. 368-375, July 1970.
- [3] P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Trans. Commun.*, vol. COM-23, pp. 921-934, Sept. 1975.
- [4] A. Lender, "Correlative digital communication techniques," *IEEE Trans. Commun. Tech.*, pp. 128-135, Dec. 1964.
- [5] A. R. Calderbank, C. Heegard, and T. A. Lee, "Binary convolutional codes with application to magnetic recording," *IEEE Trans. Inform. Th.*, vol. IT-32, pp. 797-815, Nov. 1986.
- [6] L. J. Fredrickson and J. K. Wolf, "Coding using multiple block (d,k) codes," in *IEEE International Conference on Communications*, vol. 3, pp. 1623-1627, 1989.
- [7] J. P. Harris, W. B. Philips, J. Wells, and W. D. Winger, "Innovations in the design of magnetic tape subsystems," *IBM J. Res. Dev.*, vol. 25, pp. 691-699, Sept. 1981.
- [8] L. D. Stevens, "The evolution of magnetic storage," *IBM J. Res. Dev.*, vol. 25, pp. 663-675, Sept. 1981.
- [9] D. W. Brede, J. M. Harker, R. E. Pattison, G. R. Santana, and L. G. Taft, "A quarter century of disk file innovation," *IBM J. Res. Dev.*, vol. 25, pp. 677-689, Sept. 1981.
- [10] P. A. Franaszek, "Sequence-state methods for run-length-limited coding," *IBM J. Res. Dev.*, vol. 14, pp. 376-383, July 1970.

- [11] H. Kobayashi, "A survey of coding schemes for transmission or recording of digital data," *IEEE Trans. Commun. Tech.*, vol. COM-19, pp. 1087-1100, Dec. 1971.
- [12] A. M. Patel, "Adaptive cross-parity check (AXP) code for a high density magnetic tape subsystem," *IBM J. Res. Dev.*, vol. 29, pp. 546-562, Nov. 1985.
- [13] L. R. Bahl and D. T. Tang, "Blocks codes for a class of constrained noiseless channels," *Inform. Contr.*, vol. 17, pp. 436-461, 1970.
- [14] M. W. Marcellin and H. J. Weber, "Two-dimensional modulation codes," *IEEE Journal on Selected Areas in Communications*, to appear.
- [15] G. R. Davis and B. Kamali, "Characterization of digital magnetic recording systems," *Proceedings of IEEE*, vol. PCCC 85, pp. 134-141, Mar. 1985.
- [16] J. K. Wolf and E. Zehavi, "On runlength codes," *IEEE Trans. Inform. Th.*, vol. IT-34, pp. 45-54, Jan. 1988.
- [17] R. E. Blahut, *Principles and Practice of Information Theory*. Reading, MA: Addison-Wesley, 1987.
- [18] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol. 27, pp. 379-423; 623-656, Oct. 1948.
- [19] N. D. MacKintosh, "A superposition-based analysis of pulse-slimming techniques for digital recording," *The Radio and Electronic Engineer*, vol. 50, pp. 307-314, June 1980.
- [20] A. S. Hoagland, *Digital Magnetic Recording*. New York, NY: Wiley, 1963.
- [21] E. D. Daniel and C. D. Mee, *Magnetic Recording Handbook*. McGraw-Hill, 1990.
- [22] W. A. Gardner, "Characterization of cyclostationary random signal process," *IEEE Trans. Inform. Th.*, vol. IT-21, pp. 4-14, Jan. 1975.
- [23] Y. Takasaki, "Timing extraction in baseband pulse transmission," *IEEE Trans. Commun.*, vol. COM-20, pp. 877-884, Oct. 1972.
- [24] L. E. Franks, *Signal Theory*. Englewood Cliffs, NJ: Prentice Hall, 1969.

- [25] L. E. Franks, "Carrier and bit synchronization in data communication-a tutorial review," *IEEE Trans. Commun.*, vol. COM-28, pp. 1107-1121, Aug. 1980.
- [26] P. H. Siegel, "Recording codes for digital magnetic storage," *IEEE Trans. Mag.*, vol. MAG-21, pp. 1344-1349, Sept. 1985.
- [27] J. P. Bubrouski and L. E. Franks, "Statistical properties of timing jitter in a PAM timing recovery scheme," *IEEE Trans. Commun.*, vol. COM-22, pp. 913-920, July 1974.
- [28] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1989.