Title
Advanced Control Algorithms for Discrete Linear Repetitive Processes in Self-servowriting of Hard Disk Drives

Author
Dong, Feng Dan

Publication Date
2011

Peer reviewed|Thesis/dissertation

Advanced Control Algorithms for Discrete Linear Repetitive Processes in Self-servowriting of Hard Disk Drives

by

Feng Dan Dong

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Masayoshi Tomizuka, Chair
Professor Roberto Horowitz
Professor J. Karl Hedrick
Professor S. Shankar Sastry

Spring 2011

Advanced Control Algorithms for Discrete Linear Repetitive Processes in Self-servowriting of Hard Disk Drives

by Feng Dan Dong

Abstract

Advanced Control Algorithms for Discrete Linear Repetitive Processes in Self-servowriting of Hard Disk Drives

by

Feng Dan Dong

Doctor of Philosophy in Engineering – Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

This dissertation presents advanced control algorithms for discrete linear repetitive processes in hard disk drives. Repetitive processes are characterized by a series of iterations or tracks, through a set of dynamics which has finite duration or length. On each iteration, a profile is produced, which in turn acts as a forcing function and contributes to the dynamics of the next iteration. As a result, it is inevitable that errors are propagated from iteration to iteration. In addition, the error propagation within each individual iteration along the time direction and the presence of repeatable error are two other common but significant problems in discrete linear repetitive processes, which aggravate the iteration-to-iteration error propagation. Such outcomes are clearly not desirable and hence appropriate control actions are needed to correct them.

This dissertation first introduces the unique features and control problems in discrete linear repetitive processes. In illustration, two physical processes are studied: concentric self-servowriting process and spiral based self-servowriting process. These two repetitive processes are representative servowriting techniques used in the current hard disk drive manufacturing industry. They have the same servo system consisting of two control loops: position control loop and timing control loop, but each has its own distinct control problems and objectives. Different control algorithms have been designed in this dissertation to deal with their respective control problems and improve the control system performance.

To contain the iteration-to-iteration error propagation in discrete linear repetitive processes, an iterative learning control scheme is proposed and applied in concentric self-servowriting position control loop. This is an optimization-based method which mitigates the system error by minimizing the maximum magnitude of the position deviation profile. An alternative approach, two-dimensional control scheme, is also developed for the ease

of analysis. By applying the two-dimensional systems theory, the convergence problems in the iterative learning control system are translated to the stability problems in a two-dimensional system.

To take a step further, this dissertation also proposes a novel adaptive feedforward control scheme to deal with the error propagation along both the time direction and the iteration direction, which is often the case for the timing control loop in concentric self-servowriting process. The control objective in this repetitive process is to attenuate the closure error within each individual track and contain the timing error propagation from track to track. To achieve this objective, the adaptive filter is designed by applying the filtered-x least mean square algorithm and the filter coefficients are adaptively updated for every servo sector by minimizing both the radial timing error energy and circumferential timing error energy.

The timing control loop in spiral based self-servowriting process is a typical example of discrete linear repetitive process whose dynamics is dominated by the repeatable timing error coming from the prewritten spiral tracks. Analyzing this problem from two different perspectives, two control algorithms are derived for this repetitive process. Using the classical control theory, a recursive least square based parameter adaptation algorithm is proposed to estimate and cancel the repeatable timing error. Examined from the computer system perspective, however, this repetitive process is a real-time or event-driven control system since the write head clock is updated only at the time instants when the read head detects the sync marks from the spiral tracks, i.e. the system sampling is triggered by the read head signal. From this perspective, the repeatable timing error is interpreted as the sampling jitter in the system, which causes the sampling period to be non-uniform. To tackle with the non-uniform sampling issue in this real-time control system, a novel control scheme based on Kalman filter theory is presented.

The effectiveness of the control algorithms proposed in this dissertation is verified through simulation studies using the real disk drive models and disturbance/noise data from the industry.

Professor Masayoshi Tomizuka
Dissertation Committee Chair

To dearest Kris

Contents

List of Figures

List of Tables

Acknowledgement

I was an electrical engineering student when I was pursuing my Bachelor & Master of engineering degrees. Majoring in control, I learnt that there is a mastermind in this field who is a professor in the mechanical engineering department of UC Berkeley. So when I applied for Ph.D. in UC Berkeley, I decided to switch from electrical engineering to mechanical engineering, for the simple reason that I want to learn from the best of the best - my now research advisor, Professor Masayoshi Tomizuka. I would like to express my deepest gratitude and respect to him for his never-ending guidance and support which is the key to my successful completion of the Ph.D. program. Professor Tomizuka is a man of enthusiasm, vision, and wisdom. He passed on to me not only the knowledge and problem-solving skills, but also the courage to face the difficulties, to have faith in myself, and always strive to be a good person.

I thank Professor Roberto Horowitz, Professor J. Karl Hedrick, and Professor S. Shankar Sastry for serving on my dissertation committee, providing constructive criticism about my work, and patiently helping me finish my dissertation. Computer Mechanics Laboratory (CML) had been the sponsor for the work in this dissertation. I would like to thank Mr. Edgar Sheh in Western Digital Corporation sincerely for his continuing technical help that made my dissertation possible.

I would like to express my sincere gratitude to the previous and current members in the Mechanical Systems Control (MSC) laboratory. I want to give sincere thanks to the disk drive control group members, Max Chen and Steve Zheng, for the group meetings and discussions we had together and for their professional advice and creative ideas. I also want to thank Dr. Sandipan Mishra, Dr. Benjamin Fine, Dr. Takashi Nagata, Dr. Kiyonori Inaba, Dr. Ahmed El-Shaer, Dr. Nora Han, Sanggyum Kim, Mike Chan, Even Chang-Siu, Kan Kanjanapas, Emma Yu, Lucy Fan, Sumio Sugita, Yasuyuki Matsuda, Wenjie Chen, Chi-Shen Tsai for their great friendship. Without the members in the lab, the graduate study would not have gone that fast. They made my stay in Berkeley enjoyable and memorable. Specially, I am grateful to Hoday Stearns. Thank you so much for being my best cheerleader and always encouraging me and giving me the warm cares in study and life.

I want to thank all my friends in Berkeley, especially Jared Wood, Alex Byrne, Paul Maas, Adrienne Higa, Christian Wang, Lik Chuan Lee, Maggie Chung, and Heui Yin Ng. They have been supporting me through the ups and downs in life. Without them, the life in Berkeley would not have been so colorful and fun.

I am indebted to my dearest mom, dad, and brother, for always supporting my dream and for their unlimited love. Last, and most importantly, I want to give special thanks to Kris Schouterden, for being my motivation, for always inspiring me all these years. I hope the rest of my life will be brilliant for you.

Chapter 1

Introduction

1.1 Background

Hard disk drives (HDDs) use the embedded servo systems in which the position information (i.e. the servo bursts) and timing information (i.e. the synchronization marks) are written onto the disk surface at discrete locations (i.e. the servo sectors) along the circular servo data tracks. The position and timing information are demodulated to determine the position error and timing error signals, which are used by the servo system and control the heads to accurately track-follow on or seek to a target track.

In conventional servowriting process, the position information and timing information are written by writing the servo tracks individually using the external servo writers in the clean room environment. Therefore, such process is time-consuming and expensive. Self-servowriting (SSW) technique has been attractive because of the potential savings in the cost of servowriting drive as well as reduced capital expenditure on external servo writers and in the amount of clean room floor space required.

In the current HDD manufacturing industry, two leading SSW techniques are widely employed: concentric SSW and spiral based SSW. From the viewpoint of a control system, both of these two processes are discrete linear repetitive process, which are characterized by hundreds of thousand data tracks through the same dynamics (i.e. the feedback control loop) defined over a fixed length (i.e. the number of servo sectors in one data track).

In concentric SSW process, the current track profile is generated by referring to the previously written track profile. A fundamental obstacle in this repetitive process is the propagation of the written-in errors as successive tracks are written, which is known as the radial error propagation (REP). In the worst scenario, within a few iterations, the written-in errors rapidly build up, causing the deviation from circularity and variable track-to-track spacing. As a result, the quality of the written servo patterns degrades. Since the objective of SSW process is to write the servo sectors uniformly in all tracks along perfectly concentric circles with the required constant track-to-track spacing, good position control loop and good timing control loop are required in this repetitive process. Respectively, good position control loop should be able to contain the REP from track to track and improve the quality of the written tracks, which ensure the servo sectors are

written in perfectly circular tracks. While good timing control loop aims to contain the timing error propagation from track to track (along the track direction) and attenuate the closure error within each individual track (along the time direction), which ensure the servo sectors are uniformly and accurately aligned along the circular track.

In spiral based SSW process, one set of spiral tracks are prewritten onto the disk and used as the reference for writing the product servo patterns in the circular tracks. The main problem in this repetitive process is the amplification of the repeatable timing error which is caused by the disk eccentricity and written-in timing errors in the prewritten spiral tracks. This results in unevenly spaced servo sectors along the circular tracks. Hence, in spiral based SSW servo system, the timing control loop is more significant and an effective control algorithm is highly desired to reduce the dominance of the repeatable timing error and write the product servo patterns uniformly along the circular tracks.

In this dissertation, a few effective control algorithms are proposed for these discrete linear repetitive processes to deal with their respective control problems, in an effort to enhance the control system performance and improve the quality of self-servowriting process.

The remainder of this chapter is organized as follows. Section 1.2 summaries some existing control approaches for discrete linear repetitive processes. Section 1.3 states the motivations and contributions of the dissertation, and Section 1.4 gives the outline of the dissertation.

1.2 Control of Repetitive Process

Repetitive processes are characterized by a series of iterations (or tracks in this dissertation) of the same dynamics with finite duration or length. Many manufacturing processes fall into this general class, such as metal rolling, rapid thermal processing, industrial robotics assembly, wafer scanning, and, for this dissertation, self-servowriting process in hard disk drives. The repetitiveness of the process can be exploited in several ways to enhance the system performance. Some of these methods are described below:

1. **Repetitive Control (RC).** Repetitive control uses a periodic signal generator in the feedback control loop in order to remove periodic (or repeating) disturbances. It is usually used in the processes that do not have specified start and finish times; instead they run continuously. Typical applications of repetitive control can be found in the control of hard disk drives, computer-numerically controlled (CNC) machines, and industrial robotics. There are several repetitive control design techniques based on the internal model principle [25], Fourier series expansion, external model control, etc. A comprehensive discussion and comparison of these are presented in [48].
2. **Iterative Learning Control (ILC).** ILC is a feedforward signal design technique that iteratively fine-tunes and adjusts the feedforward signal by considering the error from the previous iterations (or tracks) of the repetitive process. In other words, we iteratively reshape the input signals to the closed-loop system from one iteration to the next in order to reduce the tracking error. This method requires

that the repetitive process has specified the start and finish states (and times). Typical applications of ILC include industrial robotics, rapid thermal processing, metal rolling, wafer stages, and self-servowriting process in hard disk drives.

3. **Iterative Feedback Tuning (IFT).** IFT is similar to ILC, but iteratively fine-tunes and adjusts the feedback as well as the feedforward controller parameters by using the error from the previous iterations (or tracks) of the repetitive process. The controller parameters are tuned iteratively so as to minimize a certain cost function [37, 38, 39]. Normally, such optimization would require a complete model of the plant and may result in a high order controller. In IFT, the optimization is performed iteratively by using an estimate of the gradient of the cost function with respect to the tunable controller parameters that are calculated based on the data collected from the experiment iterations, thereby avoiding the need for complete plant model knowledge.
4. **Two-Dimensional (2D) Control.** 2D control is a controller design technique using the 2D systems theory to fulfill the performance requirements in both the time and iteration directions. A repetitive process is a 2D system in nature, where the dynamical behavior over the time is mainly determined by the feedback control loop, while the iterative process introduces the dynamics along the iteration direction. In this design technique, the repetitive process is equivalently represented by a 2D model. As a result, the convergence problems in the repetitive process can be translated to the stability problems in the 2D system. 2D systems theory provides a powerful tool to model, analyze, and design controllers for repetitive processes [52, 30, 66].

The focus of this dissertation is the design of various effective control algorithms for discrete linear repetitive processes.

1.3 Motivations and Contributions

There are five main contributions in this dissertation, outlined as below:

1.3.1 Iterative Learning Control Design

In this dissertation, an iterative learning control scheme is proposed to deal with the error propagation from iteration to iteration in discrete linear repetitive processes, particularly in the position control loop of HDD concentric self-servowriting (SSW) process.

Concentric SSW process in HDD manufacturing is a typical discrete linear repetitive process, which consists of two independent control loops: position control loop and timing control loop. In this repetitive process, the current track profile is generated by amplifying the previous-track profile through the closed-loop dynamics. As a result, the position error is propagated from track to track (known as the radial error propagation) and the minimum spacing between two adjacent tracks (known as the AC track squeeze) is getting smaller, which deteriorates the quality of the written servo patterns, or even causes the self-servowriting process to fail. Hence, the control objective in its position

control loop is to contain the radial error propagation from track to track and improve the quality of the written tracks.

In the proposed control design, the previous-track position error information is utilized for deriving the learning filter which is constructed as a finite impulse response (FIR) filter consisting of both causal and non-causal terms. An L_1 optimal control problem is formulated in the sense of minimizing the maximum magnitude of the position deviation in one track, which is stronger than the method used in [82]. In [82], the learning filter was designed by using H_∞ method in the sense of minimizing the energy of the position deviation in one track. However, the monotonic convergence of this repetitive process requires that the peak value of the position deviation decreases successively as the track number increases. Hence, L_1 optimal control method is more effective for this repetitive process, as agreed by the simulation study.

1.3.2 Two-Dimensional Control Design

A 2D control scheme is presented as an alternative approach for the iteration-to-iteration error propagation problem in discrete linear repetitive processes.

Since concentric self-servowriting process has dynamical behavior along two directions: the track (i.e. radial) direction and the time (i.e. circumferential) direction, it can be considered as a 2D discrete linear system, which is mathematically described by a 2D Roesser state-space model. The convergence problems in this repetitive process are then translated to the stability problems in a 2D system. By applying the 2D systems theory, a 2D state feedback controller is designed to achieve asymptotic stability along both the time and the track directions. To take a step further, an error learning filter is added by utilizing the previous-track position error information. As a result, the monotonic convergence along both the time and the track directions is guaranteed and the radial error propagation from track to track is contained. Since the 2D control scheme is able to guarantee the feedback control loop performance and the feedforward control loop performance simultaneously, it results in even better performance than the iterative learning control scheme. The performance of the proposed 2D control scheme is verified through the computer simulation.

1.3.3 Adaptive Feedforward Control Design

In this dissertation, a novel adaptive feedforward control (AFC) scheme is proposed to deal with the error propagation along both the time direction and the iteration direction, which is often the case for the timing control loop in concentric SSW process.

In concentric self-servowriting process, although the block diagrams of its timing control loop and position control loop are the same, the control problems and objectives are different. As stated in Section 1.3.1, the problems in the position control loop are the radial error propagation (along the track direction) and AC track squeeze (also along the track direction). However, in the timing control loop, the main issues are the closure error within each individual track (along the time direction) and timing error propagation

(along the track direction). Therefore the control objective in concentric SSW timing control loop is not only to contain the timing error propagation from track to track, but also to attenuate the closure error within each individual track. In other words, the iterative learning control and 2D control schemes proposed for the position control loop are not suitable for the timing control loop.

In the proposed AFC scheme, the adaptive filter is designed by applying the filtered-x least mean square (FXLMS) algorithm to attenuate the closure error within each individual track and contain the timing error propagation from track to track. Firstly, the filter coefficients of the current track are iteratively updated from sector to sector (before moving to the next track) by minimizing the circumferential timing error energy. And then they are used to derive the filter coefficients of the next whole track by minimizing the radial timing error energy. These two LMS based update algorithms repeat alternatively until all the tracks are written. The performance of the proposed AFC scheme is better than that of the conventional AFC schemes since the conventional AFC schemes update the adaptive filter coefficients track by track only, i.e. they are merely able to contain the error propagation from track to track. Moreover, the simulation results show that the timing control loop has much better performance by applying the proposed AFC scheme than by applying the ILC scheme which is proposed for concentric self-servowriting position control loop.

1.3.4 Recursive Least Square based Parameter Adaptation Algorithm Design

To tackle with another significant problem in discrete linear repetitive processes, the domination of repeatable error, a recursive least square (RLS) based parameter adaptation algorithm (PAA) is proposed in this dissertation. The other equally important servo-writing technique in the current HDD manufacturing industry - spiral based SSW process - is an example of discrete linear repetitive process whose dynamics is dominated by the repeatable error present in the process. In this repetitive process, the written timing profile (i.e. the write head clock) is generated through the closed-loop dynamics, by amplifying the repeatable timing error which is caused by the disk eccentricity and written-in timing errors in the prewritten spiral tracks. Similar to concentric SSW process, in spiral based SSW process, the system error is the only measurable signal and the control objective is to achieve uniform timing patterns in the write head, i.e. the write head phase signal should ideally converge to zero. In this feedback control system, however, the reference signal is the repeatable timing error which is read from the spiral tracks and not directly measurable. Hence, it is intuitive and necessary to design a correction signal for estimating and cancelling the repeatable timing error.

In the proposed control algorithm, the PAA is designed to estimate the repeatable timing error and cancel it by adding the negative value of the estimated signal. The estimated parameters are derived by using the recursive least square (RLS) algorithm. The simulation study shows the repeatable timing error is cancelled up to 90% by applying the proposed control algorithm.

1.3.5 Real-time Control Design

A real-time control scheme is analyzed and proposed as an alternative approach to deal with the repeatable timing error in spiral based SSW timing control loop.

When the product servo patterns are written based on the prewritten spiral tracks, the shifts of spiral locations and sync marks' position error appear as the repeatable timing error. Analyzed from the computer system perspective, this repetitive process is a real-time or event-driven control system because the write head signal is updated only at the time instants when the read head detects the reference information from the spiral tracks. However, the time instants of the read head operation are shifted due to the repeatable timing error. From this perspective, the repeatable timing error is interpreted as the sampling jitter in the system, which causes the sampling period of the servo system to be non-uniform. The feedback controller, however, is classically designed. In other words, it is designed by assuming the sampling period is strictly constant. When such feedback controller is implemented in a non-uniform sampling system, the control input to the plant is inaccurate, and therefore causes the controlled system response to degrade or even become unstable.

The real-time control scheme is designed based on Kalman filter theory to deal with the non-uniform sampling issue in this repetitive process. The proposed scheme resolves this issue and is implementable as simply as the current algorithm in the industry.

1.4 Overview of Dissertation

The dissertation outline is best described by Figure 1.1. This figure also represents the control algorithms which are proposed for their corresponding problems in discrete linear repetitive processes.

The remainder of this dissertation is organized as follows:

Chapter 2: A general abstract model for discrete linear repetitive processes is given. The unique characteristics of discrete linear repetitive processes are presented by studying two physical examples – concentric self-servowriting process and spiral based self-servowriting process, which are representative servowriting techniques in the current HDD manufacturing industry.

Chapter 3: An iterative learning control scheme is proposed to prevent the iteration-to-iteration error propagation. It is applied in concentric self-servowriting position control loop. The conditions of asymptotic convergence and monotonic convergence are derived and formulated into an L_1 optimal control problem for designing the error learning filter.

Chapter 4: Presents an alternative design for iterative learning control scheme based on two-dimensional (2D) systems theory. The discrete linear repetitive process is mathematically described by a 2D Roesser state-space model. A 2D state feedback controller and an error learning filter are designed together by employing the 2D systems theory, so as to guarantee asymptotic stability and monotonic convergence of the repetitive process.

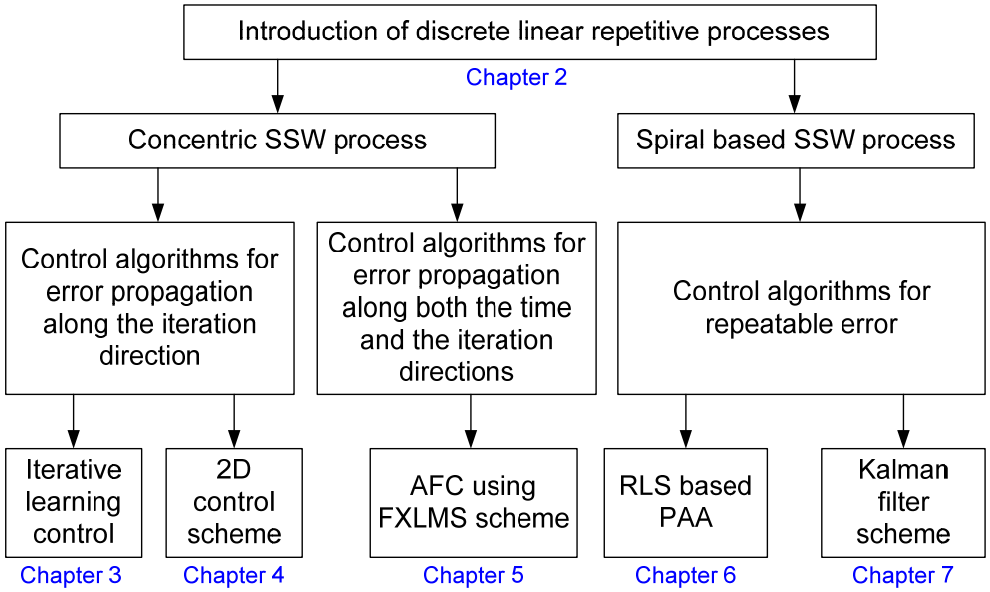

Figure 1.1: Schematic overview of the dissertation

Chapter 5: Proposes a novel adaptive feedforward control scheme to deal with the error propagation along both the time and the iteration directions in discrete linear repetitive process. In the design example, the adaptive filter is designed by applying the filtered-x least mean square algorithms to minimize both the circumferential timing error energy and the radial timing error energy.

Chapter 6: Discusses the compensation algorithms for reducing the dominance of the repeatable error present in the repetitive process. Taking the timing control loop in spiral based self-servowriting process as the design example, a recursive least square based parameter adaptation algorithm is proposed to estimate and cancel the repeatable timing error. To improve its estimation accuracy and reduce the effect of non-repeatable timing error, a comb filter is designed as the error shaping filter.

Chapter 7: Analyzes the timing control loop in spiral based self-servowriting process from the computer system perspective. The repeatable timing error is interpreted as the sampling jitter which causes non-uniform sampling period in the system. A real-time control scheme based on Kalman filter theory is proposed as an alternative approach to deal with the non-uniform sampling issue in this discrete linear repetitive process.

Chapter 8: Concludes the dissertation and summarizes the various control schemes in terms of performance and applicability.

Chapter 2

Discrete Linear Repetitive Processes

2.1 Introduction

Repetitive processes, also termed as multipass processes in the early literature, are characterized by a series of iterations, or passes, or tracks in this dissertation, through a set of dynamics with finite duration or length. On each iteration, an output – termed as the iteration profile – is produced, which in turn acts as a forcing input and contributes to the dynamics of the next iteration.

The concept of the repetitive process was first introduced in the early 1970's by the University of Sheffield, UK, on the modeling and control of long-wall coat cutting and metal rolling operations [23], [63], [67], [68]. In this chapter, the unique features and control problems of discrete linear repetitive processes are examined through two physical examples – concentric self-servowriting (SSW) process and spiral based SSW process, which are widely used in the current hard disk drive (HDD) manufacturing industry. There are two distinguishing features in these applications: (1) The productive work is undertaken by hundreds of thousand servo tracks through a set of dynamics (i.e. the feedback control loop) which is defined over a fixed finite duration known as the track length (i.e. the number of servo sectors in one data track). (2) The process evolves from the given initial conditions, and the performance deteriorates after the first track due to the effect of the previous track profile on the dynamics of the current track. Thus, it is inevitable to generate the errors which propagate from track to track and increase in amplitude. Such behavior is clearly not desirable and requires appropriate control actions to correct it.

The remainder of this chapter is organized as follows. Section 2.2 presents the definition of discrete repetitive processes. Two examples of such processes, concentric SSW process and spiral based SSW process, are introduced in Section 2.3. Finally the chapter summary is given in Section 2.4.

2.2 Discrete Repetitive Processes

The discrete repetitive process is an iterative process, and there involves identical signals in each iteration, i.e.

$$y_{i+1}(k) = y_i(k) \quad (0 \le k \le N\text{-}1 \text{ and } i = 0, 1, 2, \cdots) \tag{2.1}$$

where the index i denotes the ith iteration, k is the time index in each iteration, and N is the time length for the signal.

In a control system, when the controlled plant subjects to a repetitive process, the control problem is often called a repetitive control problem. The controlled plant may or may not have the same initial conditions when a new repetitive iteration begins. Robots performing repeating the same task repeatedly may begin each task from the same or approximately same initial state. If the end of one iteration of the repetitive process becomes the beginning of the next iteration of the repetitive process, the plant state at the beginning of each iteration may be different from one iteration to another, in particular, during transient before the system has reached the steady state where all the signals are periodic. The track following problem in hard disk drives (HDDs) servo system is such a problem.

2.3 Discrete Linear Repetitive Processes Examples and System Descriptions

Many industrial processes are discrete linear repetitive process. In the HDD industry, two popular ones are concentric SSW process and spiral based SSW process.

2.3.1 Servowriting Process in Hard Disk Drives

Hard disk drives use the embedded servo systems, where the data tracks and servo burst information (i.e. the servo sectors) are defined onto the disk surface as shown in Figure 2.1 by a process known as servo track-writing (or servowriting for short). The embedded servo sectors comprise the coarse positioning information (e.g. the data track number) and the fine positioning information (e.g. the servo bursts) for tracking the centerline of the data track while writing data to the disk and reading data from the disk. A typical disk drive consists of one or more circular platters on which the data is stored magnetically in tracks or cylinders.

There are several conventional methods used for writing the servo patterns. The common one is to write the position information (i.e. the servo burst) and the timing information (i.e. the synchronization marks) onto the disk surface by using an external servo writer machine which processes the disk drives in assembly line fashion during manufacturing. The external servo writer employs precise head positioning mechanics, such as a laser interferometer, for positioning the head at precise radial locations with respect to the previously servo-written tracks so as to achieve high track densities. In addition, the head disk assembly (HDA) within the disk drive is typically exposed to the environment through apertures which allow access to the disk drive's actuator arm and the insertion of a clock head. This requires the servowriting process to take place in a clean room environment free of contaminant particles. Further, the manufacturing

throughput is limited by the number of servo writers available as well as the cost of each servo writer and clean room space.

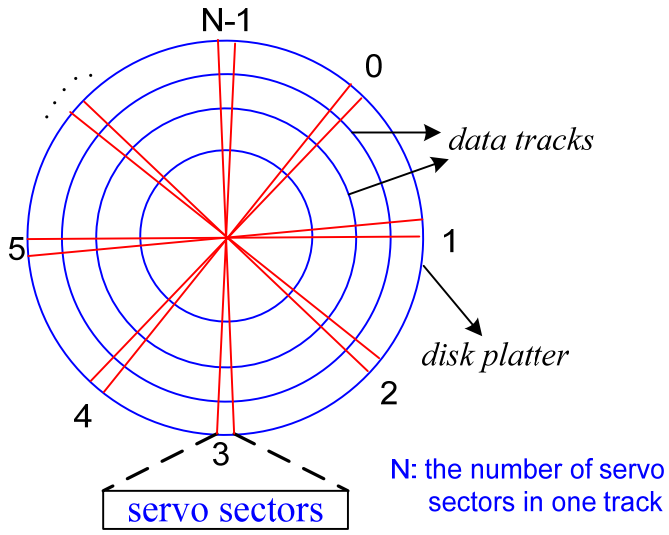

Figure 2.1: Data tracks and servo sectors on a disk platter

Self-servowriting a disk is a technique that uses the internal components of the drive and obviates the need for external servo writers, thereby decreasing the manufacturing cost and increasing the manufacturing throughput. In the current HDD industry, there are two widely used SSW techniques: concentric SSW and spiral based SSW, which are described in Section 2.3.2 and Section 2.3.3, respectively. Both of them are discrete linear repetitive process and they have the same servo system consisting of two basic control loops: position control loop and timing control loop as shown in Figure 2.2. In the position control loop, a voice coil motor (VCM) is controlled to maintain the read/write heads over the centerline of the target track during reading and writing operations. And in the timing control loop, a phase lock loop (PLL) circuit is generally used to generate the servowriting clock signal for the write head to write the product servo patterns along the circular data tracks.

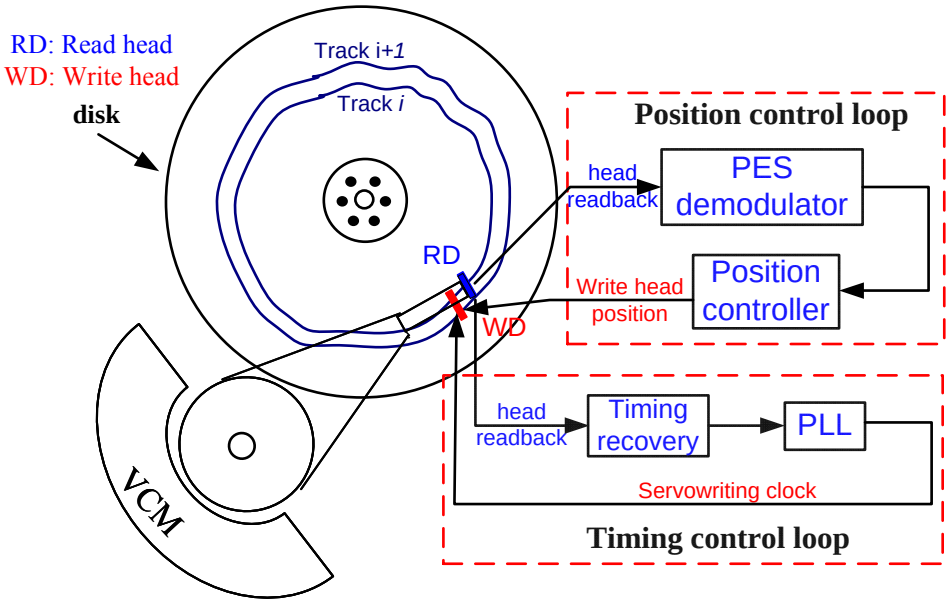

Figure 2.2: Block diagram of SSW servo control system

2.3.2 Concentric Self-Servowriting Process

Concentric SSW is also known as the seeded SSW, where the position information (in the radial direction) and the timing information (in the circumferential direction) are generated by referring to the information in the seeded track(s) or the previously written track. The process of concentric SSW is illustrated in Figure 2.3, where $i = 0, 1, 2, \cdots$ denotes the track number index and $k = 0, 1, 2, \cdots$ denotes the servo sector number index in the track. It generally involves the following steps:

(1) A seed track profile $y_0(k)$ written by the external servo writer is available on the disk. The position error signal (PES) and timing error signal (TES) are obtained when the read head is track-following on the seed track.

(2) Assume the offset between the read head and the write head is one track width. Make the read head to track-follow on the seed track in the usual track-following mode while the write head writes the product servo patterns in the next track and generates the track profile $y_1(k)$.

(3) The read head uses the newly written track profile $y_i(k)$ as the track-following reference while the write head writes on the next track and generates the track profile $y_{i+1}(k)$.

(4) Repeats step (3) until all the servo tracks are written.

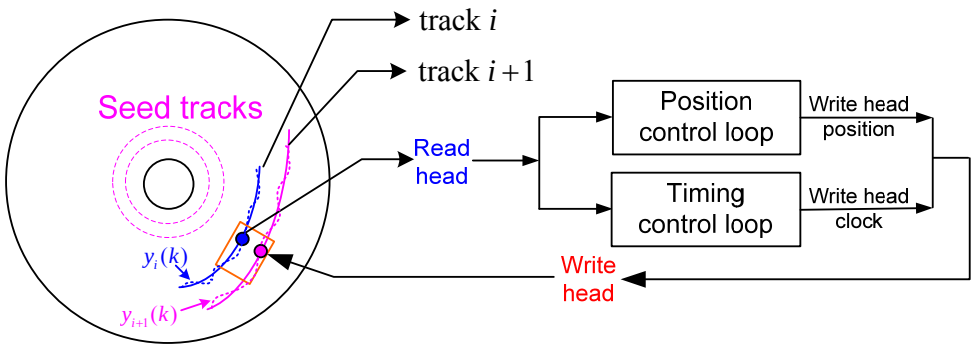

Figure 2.3: Illustration of concentric SSW process

To describe the control system in concentric SSW process, we take its position control loop as an example. Its block diagram is shown in Figure 2.4, which consists of a typical track-following control loop with the plant, i.e. the voice coil motor (VCM) $G(z)$ and a feedback controller $C(z)$. The read head track-follows on the track i with profile $y_i(k)$, while the write head writes on the track $i+1$ to generate the profile $y_{i+1}(k)$. This is possible because of the offset in the radial direction between the read head and the write head.

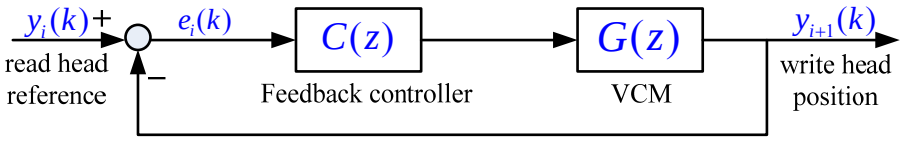

Figure 2.4: Block diagram of concentric SSW position control loop

From Figure 2.4, we have the following equation to describe the controlled process dynamics over $0 \le k \le N-1$ and $i = 0, 1, 2, \cdots$

$$y_{i+1}(k) = T(z)y_i(k) \tag{2.2}$$

where $T(z) = G(z)C(z)\big(1+G(z)C(z)\big)^{-1}$ is the complimentary sensitivity function and N denotes the number of servo sectors in one track.

Notice that Equation (2.2) is the evolution of the track profile, i.e. $T(z)$ is the response from the current track profile $y_i(k)$ to the next track profile $y_{i+1}(k)$. Therefore the track profile may grow and increase in amplitude, severely from track to track (i.e. in the i direction). The performance must deteriorate after the seed track due to the effect of the previous track profile. In other words, the output dynamics in any track acts as a forcing function (or disturbance) and therefore contributes to the dynamics of the next track. This interaction between the successive tracks is a unique characteristic of all discrete linear repetitive processes. Consequently, should there be any error, it would

propagate and grow from track to track along the radial direction. Such outcome is clearly not desirable and requires appropriate control actions to correct it.

Following the above analysis, it can be concluded that the repetitive process in Figure 2.4 is a typical case of discrete linear repetitive process, with the following characteristics:

- The system error $e_i(k)$ is the only measurable signal during the process. In other words, the read head signal $y_i(k)$ or the write head signal $y_{i+1}(k)$ is not directly measurable.
- The control objective is to minimize the write head profile. Ideally, its magnitude $|y_i(\bullet)|$ should converge to 0, i.e. $\lim_{i\to\infty}|y_i(\bullet)|=0$.

Chapter 3 and Chapter 4 present detailed descriptions on the control motivations and control algorithms design for this discrete linear repetitive process.

2.3.3 Spiral based Self-Servowriting Process

Spiral based SSW is an alternate self-servowriting technique in HDD industry. Unlike concentric SSW, it is widely used to write the product servo sectors $0, 1, \cdots, N-1$ along the circular tracks (as illustrated by the dashed lines in Figure 2.5) by referring to the spiral tracks (as illustrated by the solid lines in Figure 2.5) which are prewritten on the disk by using an external spiral writer.

The steps in spiral based SSW process are described as follows:

(1) A set of spiral tracks $0, 1, \cdots, N-1$ are prewritten on the disk by using the external spiral writer, where N represents the number of spiral tracks, which is the same as the number of servo sectors in one circular track.
(2) The read head in HDD unit reads these spiral tracks and generates a spiral wedge crossing signal which is processed by a demodulation circuit for recovering the position information and timing information from the spiral tracks.
(3) The demodulated position and timing information are respectively used by the position control loop and timing control loop to generate the position error signal (PES) and servo write clock [74] for the write head.
(4) The write head writes the concentric product servo patterns along the circular data tracks.
(5) Repeat (2) – (4) until all the circular tracks are written.

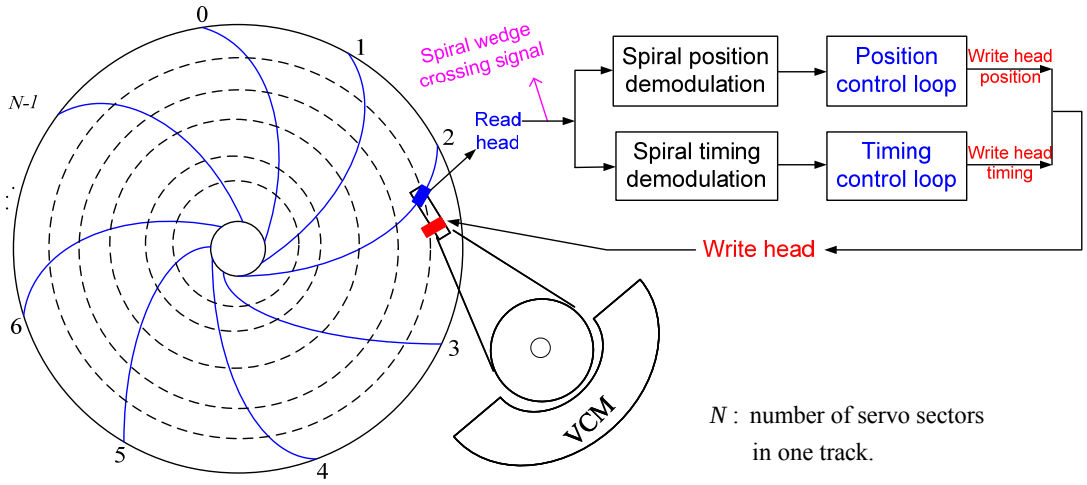

Figure 2.5: Illustration of spiral based SSW process

For spiral based SSW process, we take its timing control loop as the example to describe the control system in this repetitive process, wherein the PLL circuit is nonlinear and its basic operations are presented in Section 5.2. The general linearized block diagram of spiral based SSW timing control loop is shown in Figure 2.6.

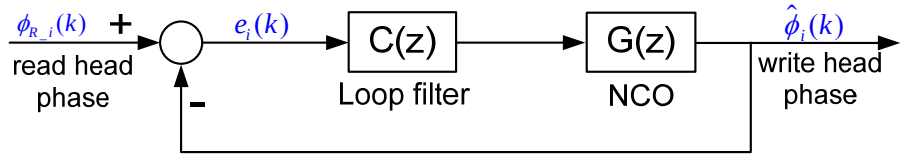

Figure 2.6: Linearized block diagram of timing control loop in spiral based SSW

The signals and blocks in Figure 2.6 are defined as follows:

- $\phi_{R\_i}(k)$: the reference phase signal at the sector k in the track i, which is read from the spiral tracks and appears as the repeatable timing error during SSW process (see Section 6.2 or [20] for details), i.e.

$$\phi_{R\_i+1}(k) = \phi_{R\_i}(k) \tag{2.3}$$

- $\hat{\phi}_i(k)$: the write head phase signal at the sector k in the track i, which is the timing deviation written in the concentric product servo patterns.
- $e_i(k)$: the system phase error at the sector k in the track i.
- $C(z)$: the loop filter in the PLL circuit.
- $G(z)$: the controlled plant, i.e. the numerically controlled oscillator (NCO) in the PLL circuit.

From the block diagram in Figure 2.6 and Equation (2.3), we have the process dynamics over $0 \le k \le N-1,\ i = 1,2,\cdots$

$$\hat{\phi}_{i+1}(k) \ = \ \hat{\phi}_i(k) \tag{2.4}$$

Equation (2.4) shows that all the track profiles are the same in this repetitive process, because they simply copy from the previous-track profile. Moreover, they can also be expressed as

$$\begin{aligned} \hat{\phi}_i(k) \ &= \ \frac{G(z)C(z)}{1+G(z)C(z)} \ \phi_{R\_i}(k) \\ &= \ T(z)\phi_{R\_i}(k) \end{aligned} \tag{2.5}$$

From Equation (2.5), we know that the write head phase profile $\hat{\phi}_i(k)$ is amplified through the closed-loop dynamics $T(z)$ from the reference profile $\phi_{R\_i}(k)$.

Same as concentric SSW process, spiral based SSW process is also a typical case of discrete linear repetitive process, with the following characteristics:

- The system phase error $e_i(k)$ is the only measurable signal. In other words, neither the reference signal from the spiral tracks $\phi_{R\_i}(k)$ nor the write head signal $\hat{\phi}_i(k)$ is directly measurable.
- The control objective is to minimize the write head phase profile. Ideally, its magnitude $\left|\hat{\phi}_i(\bullet)\right|$ should converge to 0, i.e. $\lim_{i\to\infty}\left|\hat{\phi}_i(\bullet)\right| = 0$.

Chapter 6 and Chapter 7 present detailed descriptions on the control motivations and control algorithms design for this discrete linear repetitive process.

Remark 2.2.1

Throughout the dissertation, we assume the track length is fixed for all the servo tracks in self-servowriting process.

2.4 Chapter Summary

In this chapter, a general abstract model for discrete linear repetitive processes was introduced.

The unique characteristics of discrete linear repetitive processes were presented by studying two physical examples – concentric SSW process and spiral based SSW process, which are representative servowriting techniques in the current HDD manufacturing industry. The control loops in these two repetitive processes have some similarity in that only the system error signal is measurable and the control objective is to minimize the system output (i.e. the written position/timing deviation) profile. But the control problems encountered in the two processes are different, ranging from the error propagation in the iteration direction, the error propagation over the time, to the repeatable error dominating

the system's dynamics. Therefore, various control methods are required to handle these problems.

From next chapter onwards, we examine these control problems in discrete linear repetitive processes one by one and focus on the development of analysis framework and the design of new control algorithms to resolve them accordingly.

Chapter 3

Iterative Learning Control Design

3.1 Introduction

Iterative learning control (ILC), as the name suggests, is based on the paradigm of human learning. In a repetitive process, the information from earlier iteration(s) of the process can be used to improve the performance in the current iteration. The key motivation behind the design of novel ILC schemes is the efficient use of the information from the previous iterations to maximize the performance, minimize the tracking error, improve the robustness, and accelerate the convergence rate. The early rigorous formulations of ILC were developed by Arimoto [8] and Uchiyama [77]. Since then, ILC has been implemented in many industrial processes because of its simplicity in design, analysis, and implementation. In particular, it has been successfully implemented in industrial robotics [60, 76], computer-numerical control tools [50], injection molding systems [40], rapid thermal process [15], micro-scale robotic deposition [9], and self-servowriting process in hard disk drives [22, 61, 62].

Along with the applications, many ILC algorithms that guarantee better robustness, tracking performance, and faster rates of convergence have been developed. The proportional (P) -type and proportional – derivative (PD) -type ILC schemes developed by Arimoto were designed primarily based on the stability constraints and were tuned for linear systems. Similar schemes for nonlinear systems were also designed based on the stability of ILC loop [2]. For guaranteeing the robustness and performance, the design based on stability was found to be insufficient. Alleyne et al. [10] proposed a frequency domain based ILC design using the time-varying filters for improving the robustness and performance. ILC design based on optimization of a cost function was introduced by Yamakita et al. [27]. Phan et al. [26] used the term linear quadratic optimal learning control (LQL) to describe the ILC counterpart of optimal linear quadratic regulator (LQR) problems in the standard control system design. Amann et al. [6] proposed an ILC design strategy based on optimization of a quadratic performance index with a forgetting factor. More recently, wavelet transform based ILC control strategies have been developed [81]. Significant research has also been done on the design and analysis of monotonically convergent ILC algorithms [10, 54].

For analysis and design, three alternative formulations of ILC systems have been proposed in literature. The first formulation is based on the frequency response of the plant and the error learning filter. This formulation, although restricted only to linear time invariant (LTI) plant and error learning filter design, is attractive because of its simplicity. The second approach uses a lifted domain representation [16] of the repetitive process to formulate the ILC design problem. Using the lifted formulation, it is possible to analyze and design ILC algorithms for linear time-varying (LTV) plants. However, as the iteration length becomes larger, the design and analysis complexity increases dramatically. An alternative approach towards ILC stability and design, based on two-dimensional (2D) systems theory, has also been developed in [7]. This approach is especially well suited for higher order ILC design for linear time invariant ILC analysis. Analysis and control design based on 2D systems theory for discrete linear repetitive processes are presented in Chapter 4.

Considering the repetitive nature of concentric SSW process in hard disk drives, an optimization-based ILC scheme is proposed in this chapter to mitigate the track-to-track radial error propagation (REP) and improve the quality of the written servo tracks in its position control loop. The conditions for asymptotic convergence and monotonic convergence are derived, which are stronger than what have been discussed by Wu et al. [83]. Based on these conditions, the ILC design is formulated into an L_1 optimal control problem, which can be easily solved by the optimization programming.

This chapter is organized as follows. Section 3.2 formulates the control problems in concentric SSW position control loop. The design of an iterative learning controller based on L_1 optimal control theory is presented in Section 3.3. Section 3.4 discusses the consideration of disturbance and noise in practical SSW process. The effectiveness of the iterative learning controller is demonstrated by simulation results in Section 3.5. Finally, Section 3.6 summaries the chapter.

3.2 Control Problem Formulation

3.2.1 Radial Error Propagation

From Section 2.3.2, we know that during concentric SSW process, each servowriting step writes a ‘memory’ of all preceding track errors. These written-in errors propagate along the radial direction (i.e. from track to track), which is the so-called radial error propagation (REP). REP is the main issue in concentric SSW position control loop and the servowriting process may be caused to fail without significantly controlling such error propagation.

The evolution of the track profile is written in Equation (2.2). In a general system, the gain of the complimentary sensitivity function $T(z)$, as shown in Figure 3.1, is larger than 1 in some range of frequencies, which amplifies the frequency components of the track profile $y_i(k)$ and causes the written-in errors to build up in the radial direction across the disk, as shown in Figure 3.2. So in order to contain the REP, we need to control the

magnitude of $y_i(k)$. Ideally, it is desired to design a correction signal to ensure $\lim_{i\to\infty}|y_i(\bullet)|=0$, as illustrated in Figure 3.3.

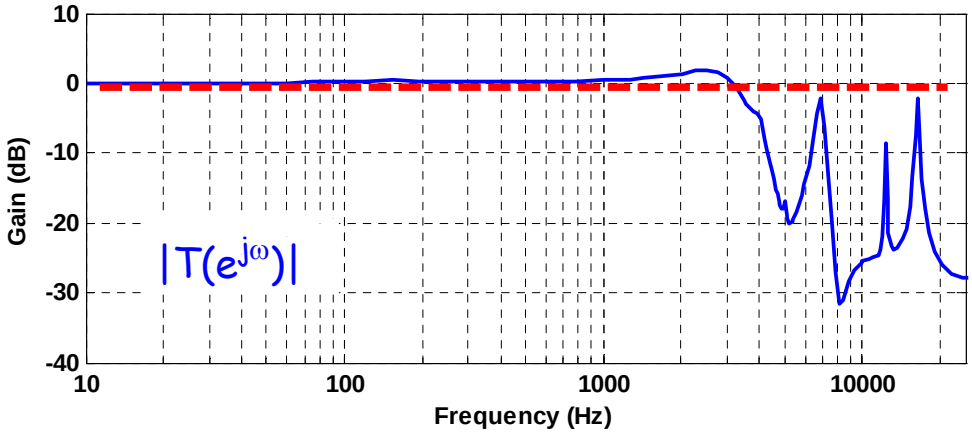

Figure 3.1: Frequency response of general complementary sensitivity function

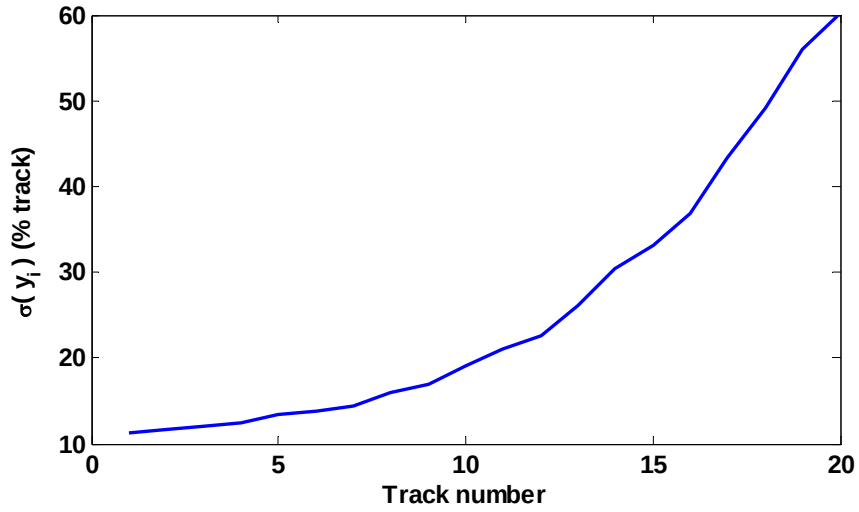

Figure 3.2: Actual written track profile due to REP

3.2.2 AC Track Squeeze

In addition to containing the REP, another significant control objective in concentric SSW position control loop is to assure good servowriting quality. An important quantity in this regard is the AC track squeeze, which is defined as the minimum spacing between

two adjacent tracks as shown in Figure 3.4. The AC track squeeze for the track i, denoted by $T_{sq}(i)$, can be written as

$$T_{sq}(i) \triangleq \min_{k\in[0,\,N-1]} \{1+y_i(k)-y_{i-1}(k)\}\times 100\% \tag{3.1}$$

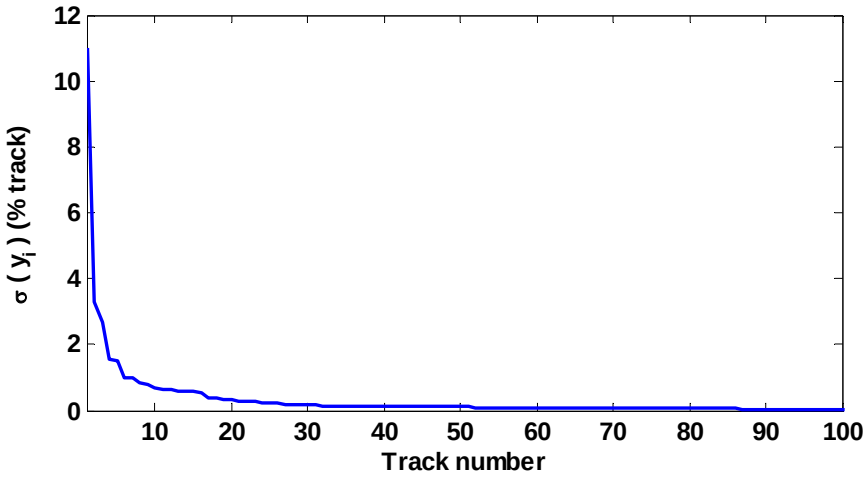

Figure 3.3: Ideal written track profile

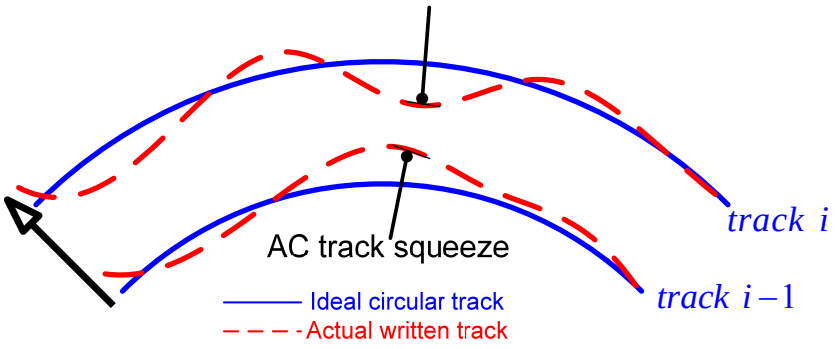

Figure 3.4: Illustration of AC track squeeze

Remark 3.2.1

- $y_i(k)$ represents the written track profile at the sector k in the track i. Its value denotes the deviation of the written track center from the ideal circular track center.
- $y_i(k)$ is positive (or negative) when the written track profile deviates outwards (or inwards) as illustrated by the arrow in Figure 3.4.
- The magnitude of $y_i(k)$ is normalized to the percentage of one track width.

From Equation (3.1), we can see that the ideal $T_{sq}(i)$ should satisfy: $\lim_{i\to\infty} T_{sq}(i) = 100\%$, as shown in Figure 3.5. If the AC track squeeze is too small, two adjacent tracks with narrow spacing may interfere with each other and cause data corruption.

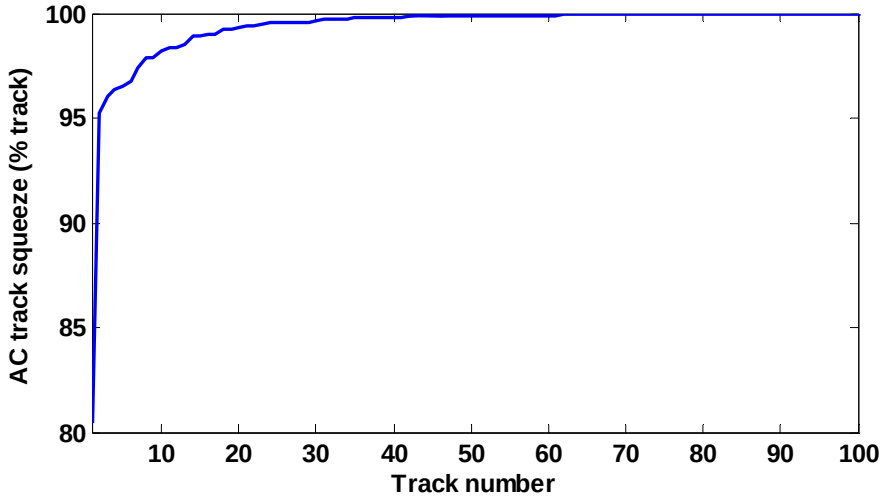

Figure 3.5: Ideal AC track squeeze profile

Notice that Equation (3.1) can be further written as

$$\begin{aligned} T_{sq}(i) &= \left\{1+\min_{k\in[0,N-1]}\left[y_i(k)-y_{i-1}(k)\right]\right\}\times 100\% \\ &\geq \left\{1-\max_{k\in[0,N-1]}\left|y_i(k)-y_{i-1}(k)\right|\right\}\times 100\% \\ &\geq \left\{1-\left\|y_i(\bullet)\right\|_\infty-\left\|y_{i-1}(\bullet)\right\|_\infty\right\}\times 100\% \end{aligned} \tag{3.2}$$

where $\|y_i(\bullet)\|_\infty$ represents the maximum magnitude of the head position deviation profile along the track i, i.e. $\|y_i(\bullet)\|_\infty = \max_{0\le k\le N-1}(|y_i(k)|)$ $(i=0,1,\ldots)$.

Equation (3.2) shows that the AC track squeeze $T_{sq}(i)$ is determined by $\|y_i(\bullet)\|_\infty$ and $\|y_{i-1}(\bullet)\|_\infty$. Therefore, it is intuitive to control $\|y_i(\bullet)\|_\infty$ in order to improve the AC track squeeze.

3.3 Iterative Learning Control Design using L_1 Optimal Control

3.3.1 Analysis in Lifted Domain

In this section, a feedforward correction signal $u_i(k)$ is designed and plugged into concentric SSW position control loop as shown in Figure 3.6.

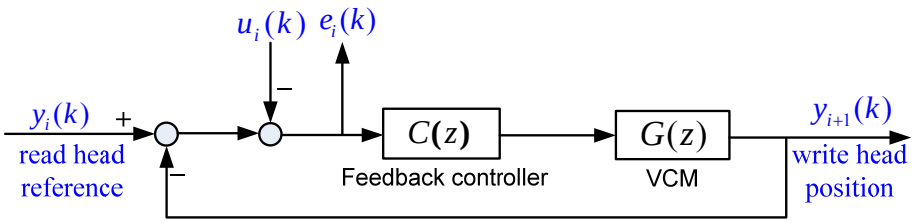

Figure 3.6: Block diagram of concentric SSW position control loop with ILC

Notice that in concentric SSW process, the read head or write head position signal $y_i(k)$ is not measurable. However, the position error signal (PES) $e_i(k)$ and its previous-track values $e_{i-1}(k)$ are available. Hence, it is feasible to use $e_{i-1}(k)$ to generate the correction signal $u_i(k)$, i.e.

$$u_i(k) = F(z)\cdot e_{i-1}(k) \tag{3.3}$$

where $F(z)$ is the learning filter to be designed.

In general, $F(z)$ has both causal and non-causal terms, as written in the following equation

$$F(z) = \cdots + f_2^N z^2 + f_1^N z + f_0^C + f_1^C z^{-1} + f_2^C z^{-2} + \cdots$$

where $f_n^N \ (n \geq 1)$ represents the coefficient of non-causal term and $f_n^C \ (n \geq 0)$ represents the coefficient of causal term.

Remark 3.3.1

According to Equation (3.3), the error signal from the previous track has been recorded when writing the current track. That means $\{e_{i-1}(k) : 0 \leq k \leq N-1\}$ is available for control computation from the beginning of the current track i. Therefore, a non-causal $F(z)$ is realizable with any non-zero coefficient $f_n^N \ (n \geq 1)$.

From the block diagram in Figure 3.6, we obtain the evolution of the track profile with the correction signal $u_i(k)$, i.e. the relationship between the read head profile $y_i(k)$ and the write head profile $y_{i+1}(k)$,

$$\begin{aligned} y_{i+1}(k) &= T(z)\left[y_i(k) - u_i(k)\right] \\ &= T(z)y_i(k) - T(z)F(z)e_{i-1}(k) \\ &= T(z)y_i(k) - T(z)F(z)\left(G(z)C(z)\right)^{-1} y_i(k) \\ &= \left[T(z) - S(z)F(z)\right] y_i(k) \end{aligned} \tag{3.4}$$

where $S(z) = 1/(1 + G(z)C(z))$ is the sensitivity function and $T(z) = 1 - S(z)$ is the complementary sensitivity function.

Notice that the position error signal (PES) profile $e_i(k)$ has the same evolution, i.e.

$$\begin{aligned} e_i(k) &= y_i(k) - u_i(k) - y_{i+1}(k) \\ &= y_i(k) - F(z)e_{i-1}(k) - G(z)C(z)e_i(k) \\ &= S(z)\left[y_i(k) - F(z)e_{i-1}(k)\right] \\ &= S(z)\left[G(z)C(z)e_{i-1}(k) - F(z)e_{i-1}(k)\right] \\ &= \left[T(z) - S(z)F(z)\right]e_{i-1}(k) \end{aligned} \tag{3.5}$$

Now, we define the following super-vectors:

$$y_i = \begin{bmatrix} y_i(0) \\ y_i(1) \\ y_i(2) \\ \vdots \\ y_i(N-1) \end{bmatrix}, \quad e_i = \begin{bmatrix} e_i(0) \\ e_i(1) \\ e_i(2) \\ \vdots \\ e_i(N-1) \end{bmatrix}, \quad u_i = \begin{bmatrix} u_i(0) \\ u_i(1) \\ u_i(2) \\ \vdots \\ u_i(N-1) \end{bmatrix}$$

then the track profile evolution in Equation (3.4) can be written in the lifted domain

$$\begin{aligned} y_{i+1} &= Ty_i - Tu_i \\ &= Ty_i - TFe_{i-1} \\ &= Ty_i - TF(GC)^{-1}y_i \\ &= \left[T - SF\right]y_i \end{aligned} \tag{3.6}$$

where S, T, G, C, and F are Markov matrices of $S(z)$, $T(z)$, $G(z)$, $C(z)$, and $F(z)$.

Analogously, the lifted-domain representation of the PES profile evolution in Equation (3.5) is

$$\begin{aligned} e_i &= y_i - y_{i+1} - u_i \\ &= S\left[y_i - Fe_{i-1}\right] \\ &= \left[T - SF\right]e_{i-1} \end{aligned} \tag{3.7}$$

Notice that the impulse response of $S(z)$ is represented as

$$S(z) = s_0 + s_1 z^{-1} + s_2 z^{-2} + \cdots + s_{N-1} z^{-N+1} + \cdots$$

where $s_0, s_1, \cdots, s_{N-1}$ are Markov parameters of $S(z)$.

By using the Markov parameters, we construct the Markov matrix S as an $N \times N$ matrix

$$S = \begin{bmatrix} s_0 & 0 & 0 & \dots & 0 \\ s_1 & s_0 & 0 & \dots & 0 \\ s_2 & s_1 & s_0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ s_{N-1} & s_{N-2} & s_{N-3} & \dots & s_0 \end{bmatrix}$$

Matrices T and F are constructed analogously and written as

$$T = \begin{bmatrix} t_0 & 0 & 0 & \dots & 0 \\ t_1 & t_0 & 0 & \dots & 0 \\ t_2 & t_1 & t_0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ t_{N-1} & t_{N-2} & t_{N-3} & \dots & t_0 \end{bmatrix}, \quad F = \begin{bmatrix} f_0^C & f_1^N & f_2^N & \cdots & & \\ f_1^C & f_0^C & f_1^N & f_2^N \cdots & & \\ f_2^C & f_1^C & f_0^C & f_1^N \cdots & & \\ \vdots & f_2^C & f_1^C & f_0^C & f_1^N & \ddots \\ & \ddots & \ddots & \ddots & \ddots & f_2^N \\ & & \cdots & f_2^C & f_1^C & f_0^C & f_1^N \\ & & & \cdots & f_2^C & f_1^C & f_0^C \end{bmatrix}$$

where $t_0, t_1, \cdots, t_{N-1}$ are Markov parameters of $T(z)$.

Remark 3.3.2

Equation (3.6) (or Equation (3.7)) implies that the matrix $[T - SF]$ is essential in relating the current track profile vector y_i and the next track profile vector y_{i+1} (or the current-track PES profile vector e_i and the next-track PES profile vector e_{i+1}).

3.3.2 Asymptotic Convergence Analysis

Definition 3.3.1 Concentric SSW position control loop is said to be **asymptotically convergent** if the head position deviation profile $y_i(\bullet)$ converges to zero, i.e. $\lim_{i\to\infty} y_i(\cdot) = 0$.

It is easy to see that the condition for asymptotic convergence of the system in Equation (3.6) is

$$\left|\lambda\left(T - SF\right)\right| < 1 \tag{3.8}$$

where $\lambda(T - SF)$ denotes the eigenvalue of the matrix $(T - SF)$.

3.3.3 Monotonic Convergence Analysis

To improve the AC track squeeze, asymptotic convergence is insufficient and a stronger condition is required.

Definition 3.3.2 Concentric SSW position control loop is said to be **monotonically convergent** if the maximum magnitude of the head position deviation profile, i.e. $\|y_i(\bullet)\|_\infty$, decreases successively, as the track number i increases.

Notice that the above definition of monotonic convergence can be equivalently expressed as

$$\|y_{i+1}(\bullet)\|_\infty \; < \; \|y_i(\bullet)\|_\infty \quad (\textit{for} \;\; i = 0,1,\cdots) \tag{3.9}$$

For the frequency-domain representation in Equation (3.4), the condition for fulfilling Equation (3.9) is

$$\|T(z) - S(z)F(z)\|_1 \; < \; 1 \tag{3.10}$$

where $\| \bullet \|_1$ denotes the L_1 norm.

For the lifted-domain representation in Equation (3.6), the condition for fulfilling Equation (3.9) is

$$\|T - SF\|_\infty \; < \; 1 \tag{3.11}$$

where $\| \bullet \|_\infty$ denotes the infinity norm.

Analysis

- Equation (3.10) or Equation (3.11) is an L_1 optimal control problem.
- Since the PES profile has the evolution as written in Equation (3.5) or Equation (3.7), so Equation (3.10) or Equation (3.11) also implies that

  $$\|e_{i+1}(\bullet)\|_\infty \; < \; \|e_i(\bullet)\|_\infty \quad (\textit{for} \;\; i = 0,1,\cdots)$$

  That means under the condition of monotonic convergence in Equation (3.9), the peak value of the PES decreases monotonically as well as the peak value of write head position deviation. As a result, the quality of self-servowriting (i.e. the AC track squeeze) is improved.
- $\|y_{i+1}(\bullet)\|_\infty < \|y_i(\bullet)\|_\infty$ is stronger than what has been discussed by Wu et al. [85]: $\|y_{i+1}(\bullet)\|_2 < \|y_i(\bullet)\|_2$, which indicates monotonic decrease of the energy. It requires $\|T(z) - S(z)F(z)\|_\infty < 1$, which is an H_∞ control problem. In the lifted-domain representation, it requires $\|T - SF\|_2 < 1$.
- Let $\bar{t}(k)$ be the impulse response of $T(z) - S(z)F(z)$. Then $\|T(z) - S(z)F(z)\|_\infty$ can be computed as

  $$\max_k \left|\bar{t}(k)\right|$$

  and $\|T(z) - S(z)F(z)\|_1$ can be computed as

$$\sum_k \left|\tilde{t}(k)\right|$$

Hence, $\|T(z)-S(z)F(z)\|_1 < 1$ is more conservative than $\|T(z)-S(z)F(z)\|_\infty < 1$. Equivalently to say, $\|y_{i+1}(\bullet)\|_\infty < \|y_i(\bullet)\|_\infty$ is the sufficient condition of $\|y_{i+1}(\bullet)\|_2 < \|y_i(\bullet)\|_2$.

L_1 optimal control

In [80], Vidyasagar first formulated the L_1 optimal control problem. Similar to the H_∞ optimal control problem, it seeks to minimize the influence of additive disturbances on the system outputs. In the H_∞ control problem, this influence is measured in terms of the amplification of the disturbance input energy. In the L_1 control problem, this influence is measured in terms of the maximal possible amplification of the disturbance input magnitude. A complete solution to the L_1 optimal control problem was given in [65]. Through the use of a parameterization of all stabilizing controllers, an optimal, possibly dynamic, linear controller is obtained by solving appropriate linear programming.

3.3.4 L_1 Optimal Control Formulation

As a design example, the learning filter $F(z)$ is considered as a finite impulse response (FIR) filter with three causal terms and two non-causal terms, i.e.

$$F(z) = f_2^N z^2 + f_1^N z + f_0^C + f_1^C z^{-1} + f_2^C z^{-2} \tag{3.12}$$

Now we define the following new transfer functions and their corresponding Markov matrices:

(a) $S_1(z)$ is defined as one-step advance of $S(z)$, i.e. $S_1(z) = z \cdot S(z)$ and S_1 is its Markov matrix. S_2 is defined as Markov matrix of $S_2(z)$, which is two-step advance of $S(z)$, i.e. $S_2(z) = z^2 \cdot S(z)$. Matrices S_1 and S_2 are written as

$$S_1 = \begin{bmatrix} s_1 & s_0 & 0 & \dots & 0 \\ s_2 & s_1 & s_0 & \dots & 0 \\ s_3 & s_2 & s_1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & s_{N-1} & s_{N-2} & \dots & s_1 \end{bmatrix}, \quad S_2 = \begin{bmatrix} s_2 & s_1 & s_0 & \dots & 0 \\ s_3 & s_2 & s_1 & \dots & 0 \\ s_4 & s_3 & s_2 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & s_{N-1} & \dots & s_2 \end{bmatrix} \tag{3.13}$$

(b) Similarly, $S_3(z)$ and $S_4(z)$ are defined as one-step delay and two-step delay of $S(z)$, i.e. $S_3(z) = z^{-1} \cdot S(z)$ and $S_4(z) = z^{-2} \cdot S(z)$; their corresponding Markov matrices are S_3 and S_4, which are written as

$$S_3 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ s_0 & 0 & 0 & \cdots & 0 \\ s_1 & s_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ s_{N-2} & s_{N-3} & s_{N-4} & \cdots & 0 \end{bmatrix}, \quad S_4 = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ s_0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ s_{N-3} & s_{N-4} & s_{N-5} & \cdots & 0 \end{bmatrix} \tag{3.14}$$

Consequently, the matrix SF is expressed as

$$\begin{aligned} SF &= f_0^C \cdot S_0 + f_1^N \cdot S_1 + f_2^N \cdot S_2 + f_1^C \cdot S_3 + f_2^C \cdot S_4 \\ &= \sum_{\kappa=0}^{4} S_k f_k \end{aligned} \tag{3.15}$$

where

$$S_0 = S, \ f_0 = f_0^C, \ f_1 = f_1^N, \ f_2 = f_2^N, \ f_3 = f_1^C, \ f_4 = f_2^C$$

Using the definition of the matrix infinity norm, the following holds

$$\begin{aligned} \left\| T - SF \right\|_\infty &= \left\| T - \sum_{\kappa=0}^{4} S_k f_k \right\|_\infty \\ &= \max_i \left(\sum_{j=1}^{N} \left| T_{ij} - \sum_{\kappa=0}^{4} S_{k\_ij} f_k \right| \right) \end{aligned} \tag{3.16}$$

where S_{k_ij} is the element (i, j) of matrix S_k $(k = 0,1,\cdots,4)$; and T_{ij} is the element (i, j) of matrix T.

By introducing Equation (3.16), Equation (3.11) can be expressed as

$$\max_i \left(\sum_{j=1}^{N} \left| T_{ij} - \sum_{\kappa=0}^{4} S_{k\_ij} f_k \right| \right) < 1 \tag{3.17}$$

Notice that Equation (3.17) is an optimization control problem.

3.4 Consideration of Disturbance and Noise

During practical concentric SSW process, the spindle/disks disturbance and the sensor noise always exist. They inevitably deviate the read/write heads from the desired track center and deteriorate the quality of the written servo tracks. Hence, the spindle/disks disturbance $d_i(k)$ and the sensor noise $n_i(k)$ are taken into consideration in the simulation study, as shown in Figure 3.7.

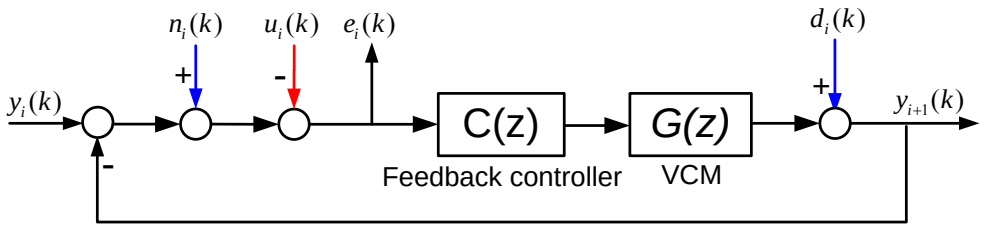

Figure 3.7: Concentric SSW position control loop with disturbance and noise

By considering $d_i(k)$ and $n_i(k)$, the written tracks and PES have new profile evolution. For example, the track profile evolution in Equation (3.6) is rewritten as

$$\begin{aligned} y_{i+1} &= Ty_i - Tu_i + Tn_i + Sd_i \\ &= Ty_i - TFe_{i-1} + Tn_i + Sd_i \\ &= Ty_i - TF(GC)^{-1} y_i + TF(GC)^{-1} d_{i-1} + Tn_i + Sd_i \\ &= \left[T - SF\right] y_i + Tn_i + Sd_i + SFd_{i-1} \end{aligned} \tag{3.18}$$

From Equation (3.18), it can be seen that the next-track profile y_{i+1} is affected not only by the current-track disturbance n_i and d_i, but also by the previous-track disturbance d_{i-1}. Since the response from $n_i(k)$ to $y_{i+1}(k)$ is $T(z)$, and the response from $d_i(k)$ to $y_{i+1}(k)$ is $S(z)$, so the rejection performance on n_i and d_i is determined by $T(z)$ and $S(z)$, which are fixed by the feedback controller $C(z)$. On the other hand, from Equation (3.18), it can be seen that the rejection performance on d_{i-1} is determined by $S(z)F(z)$. Thus, in order to effectively attenuate d_{i-1}, the learning filter $F(z)$ should be designed such that $\|SF\|_\infty \le \|S\|_\infty$, or equivalently

$$\|F\|_\infty \le 1 \tag{3.19}$$

Notice that matrix F has the following structure

$$F = \begin{bmatrix} f_0^C & f_1^N & f_2^N & & & & \\ f_1^C & f_0^C & f_1^N & f_2^N & & & \\ f_2^C & f_1^C & f_0^C & f_1^N & & & \\ & f_2^C & f_1^C & f_0^C & f_1^N & & \\ & & \ddots & \ddots & \ddots & \ddots & f_2^N \\ & & & f_2^C & f_1^C & f_0^C & f_1^N \\ & & & & f_2^C & f_1^C & f_0^C \end{bmatrix} \tag{3.20}$$

From Equation (3.20), we further conclude that $\|F\|_\infty \le 1$ can be rewritten as

$$\left(\left|f_2^C\right|+\left|f_1^C\right|+\left|f_0^C\right|+\left|f_1^N\right|+\left|f_2^N\right|\right) \le 1 \tag{3.21}$$

Now the design problems in equations (3.17) and (3.21) are formulated into one optimization problem to obtain the feasible filter coefficients. This optimization problem can be solved by using the semidefinite programming (SDP) in Yalmip package [87].

3.5 Simulation Study

In this section, the proposed ILC scheme is applied to a benchmark hard disk drive (HDD) developed by IEEJapan technical committee on nano-scale servo (NSS) system [88]. The simulated HDD has 220 servo sectors per track, i.e. $N = 220$. The spindle rotation speed is 7200 rpm. Thus, the sampling frequency is $220 \times (7200/60) = 26,400$ Hz. The bode plot of VCM in the position control loop is shown in Figure 3.8.

The performance of the learning filter $F(z)$ is firstly examined. Using the SDP in Yalmip package obtains $F(z)$ as

$$\begin{aligned} F(z) &= f_2^N z^2 + f_1^N z + f_0^C + f_1^C z^{-1} + f_2^C z^{-2} \\ &= -0.2872z^2 + 0.0095z - 0.3094 - 0.1482z^{-1} - 0.0876z^{-2} \end{aligned} \tag{3.22}$$

From Equation (3.22), we can verify that

$$\left(\left|f_2^C\right|+\left|f_1^C\right|+\left|f_0^C\right|+\left|f_1^N\right|+\left|f_2^N\right|\right) = 0.842 < 1$$

i.e. Equation (3.21) is fulfilled.

Figure 3.9 shows the frequency response of $F(z)$. It is clearly seen that the magnitude of $F(z)$ is below 0 *db* at all frequencies, which is consistent with Equation (3.19). The frequency responses of $S(z)$ and $S(z)F(z)$ are shown in Figure 3.10. The dashed line represents the response of $S(z)$ and the solid line represents the response of $S(z)F(z)$. It is observed that $S(z)F(z)$ not only preserves the disturbance attenuation property of $S(z)$ at low frequencies, but also has much lower values than $S(z)$ at high frequencies. In other words, $S(z)F(z)$ has better rejection performance on d_{i-1} than $S(z)$.

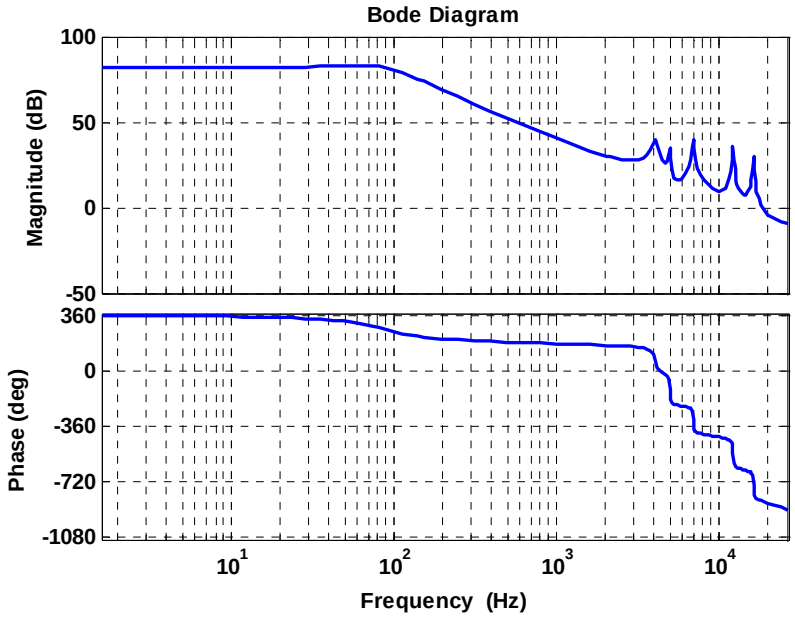

Figure 3.8: Bode plot of VCM in SSW position control loop

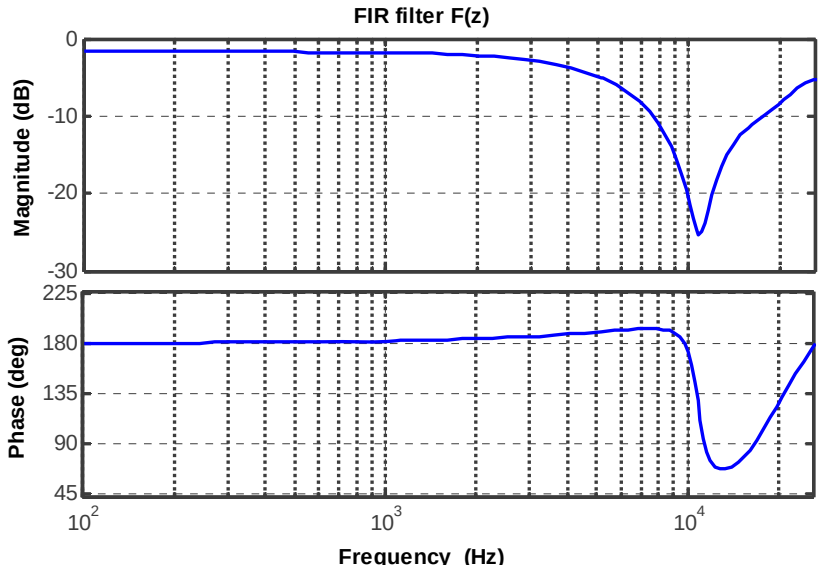

Figure 3.9: Frequency response of $F(z)$

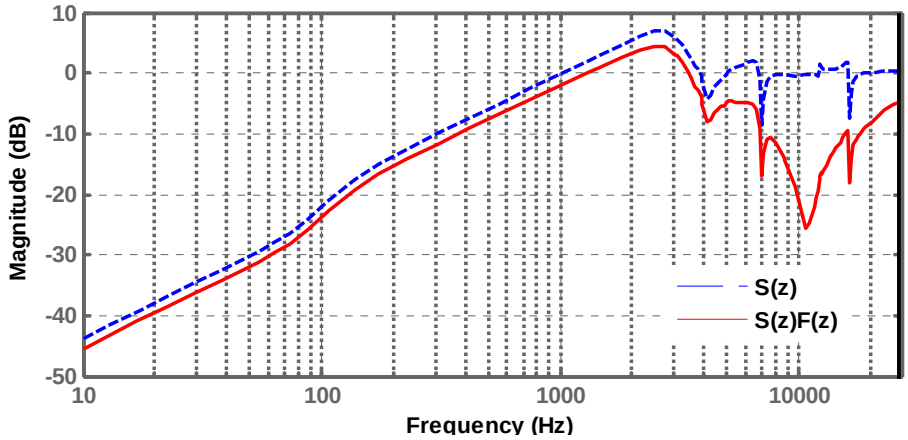

Figure 3.10: Frequency responses of $S(z)$ and $S(z)F(z)$

Figure 3.11 and Figure 3.12 show the closed-loop responses before and after applying the proposed ILC algorithm, respectively. From Equation (3.22), we obtain $\|T - SF\|_\infty = 0.9963$, which satisfies the condition of monotonic convergence in Equation (3.11). As a result, it can be seen that after applying the proposed ILC scheme as shown in Figure 3.12, the closed-loop response, i.e. $T(z) - S(z)F(z)$, has the maximum gain of less than 1, which ensures that the effect of non-circular tracks is diminishing as the track number i increases.

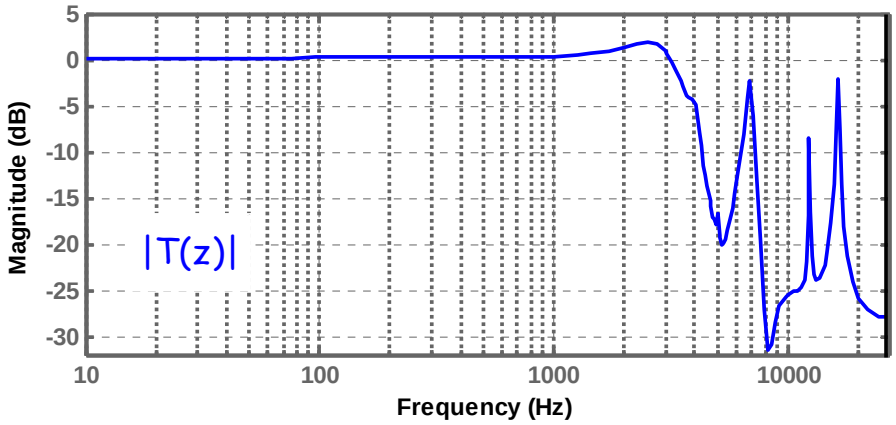

Figure 3.11: Closed-loop response before applying ILC

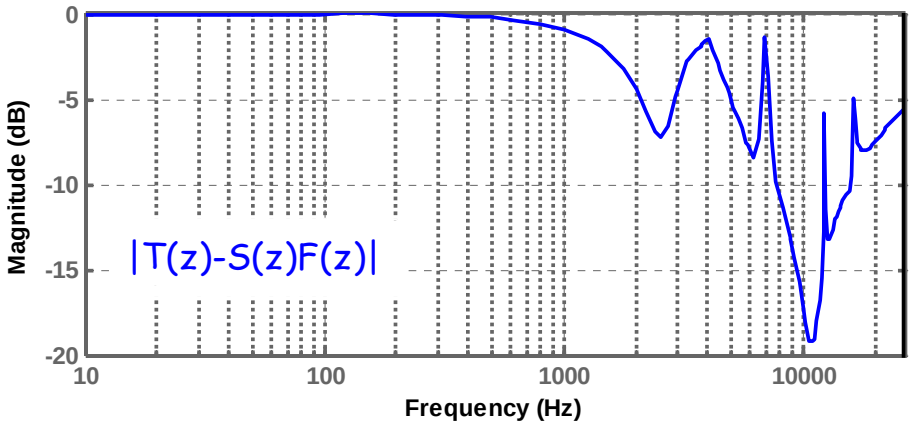

Figure 3.12: Closed-loop response after applying ILC

In the simulation study, the modeled sensor noise $\{n_i(k)\}$ is white noise with average $\sigma[n_i]=1.5\%$ track width as shown in Figure 3.13; and the disk/spindle disturbance $\{d_i(k)\}$ is modeled as white noise with average $\sigma[n_i]=1.7\%$ track width as shown in Figure 3.14. The seed track profile $y_0(k)$ (as shown in Figure 3.15) is assumed to be given with the average $\sigma[y_0]=11.0\%$ track width. In the simulation, 1000 servo tracks were written by applying the proposed ILC algorithm.

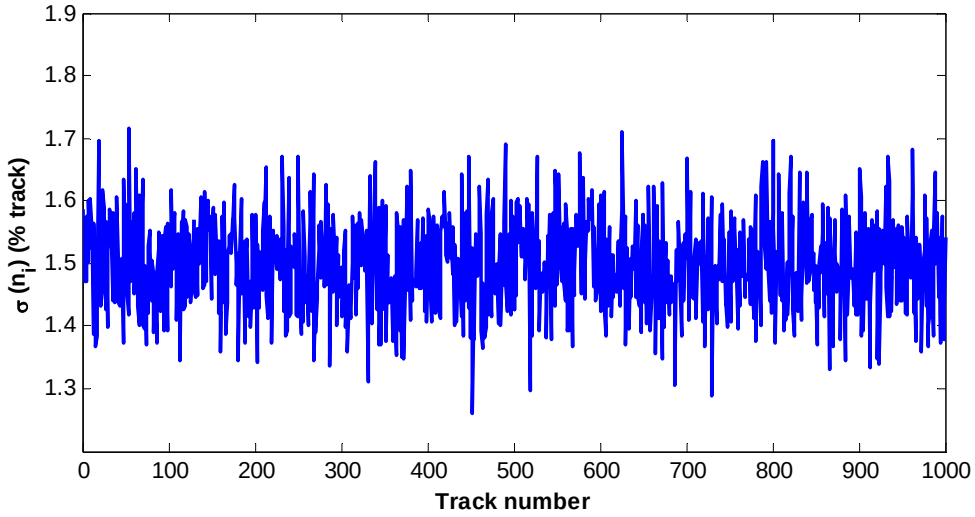

Figure 3.13: Sensor noise in SSW position control loop

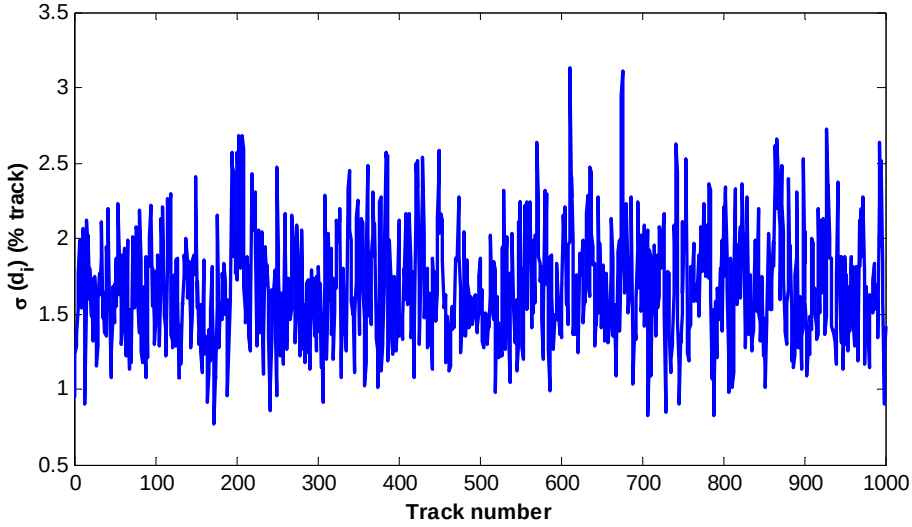

Figure 3.14: Disk/spindle disturbance in SSW position control loop

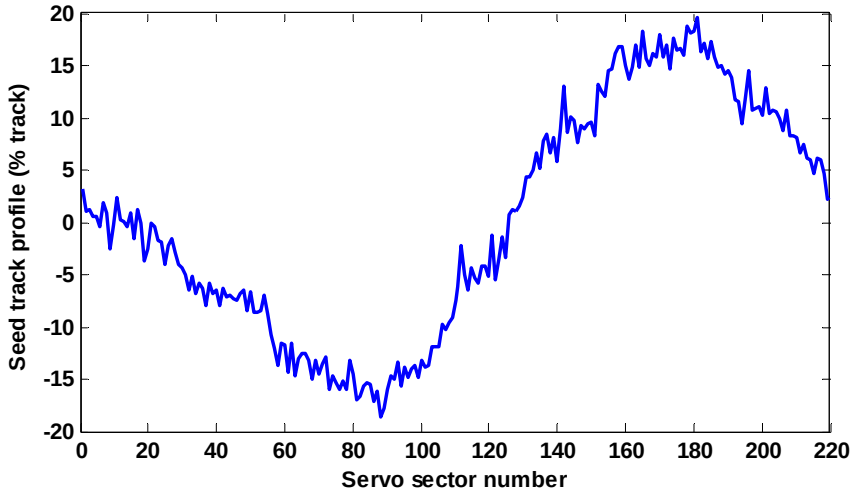

Figure 3.15: Seed track profile

Figure 3.16 shows the track profile with the effect of disturbance and sensor noise. The resulting average $\sigma[y_i]$ is 2.41% track width, which is much smaller than that of the seed track y_0. It is also observed that the written track profile y_i is convergent. In order to verify the effectiveness of the proposed ILC algorithm, we compare the result in Figure 3.16 with the result in the original system, i.e. before applying the proposed ILC scheme. The comparison is shown in Figure 3.17. The solid line shows the written track profile with the ILC algorithm, and the dotted line shows the written track profile without the

ILC algorithm. It is learnt that in the original system, the resulting written track profile is much worse, with a high average $\sigma[y_i] = 30.1\%$ track width. Hence, we can conclude that the REP is contained well in the system with the proposed ILC scheme.

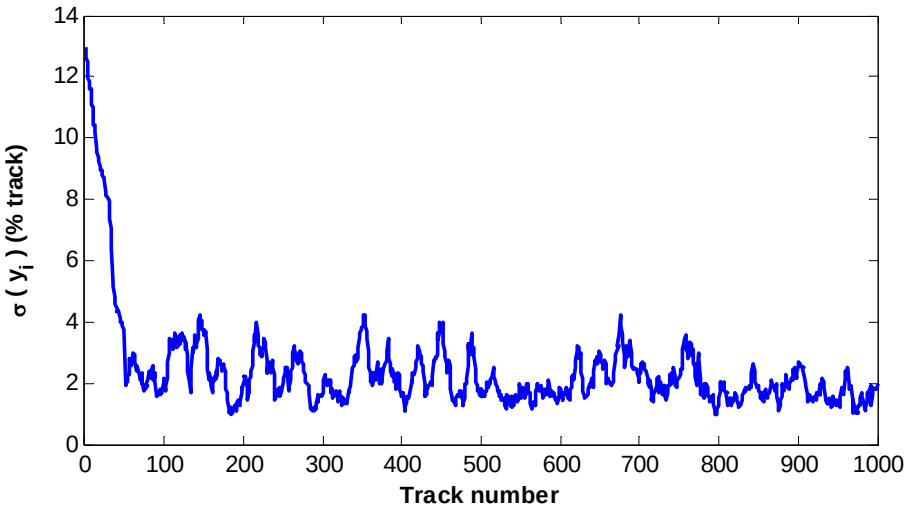

Figure 3.16: Written track profile

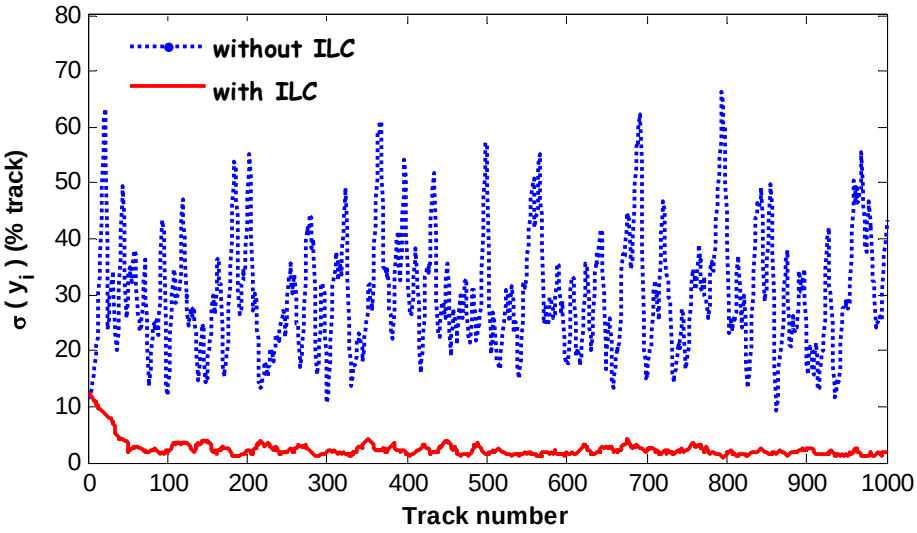

Figure 3.17: Comparison of written track profile with and without ILC scheme

Another important performance metrics is the AC track squeeze. From Equation (3.1), we know that the ideal AC track squeeze is one track width. If the AC track squeeze is too small, two adjacent tracks with narrow spacing may cause some data corruption.

Figure 3.18 and 3.19 show the AC track squeeze profile and its histogram, respectively. The resulting average AC track squeeze is 92.1% track width, which is within the acceptable limit and better than the result in [83]. Figure 3.20 shows the comparison of this result with the result in the original system. The solid line shows the AC track squeeze with the proposed ILC scheme, i.e. the result in Figure 3.18. The dotted line shows the AC track squeeze in the original system, i.e. without the ILC algorithm. From Figure 3.20, we can clearly see that with the proposed ILC scheme, the AC track squeeze is greatly improved and its average value is much higher than the result (-61.9% track width) in the original system.

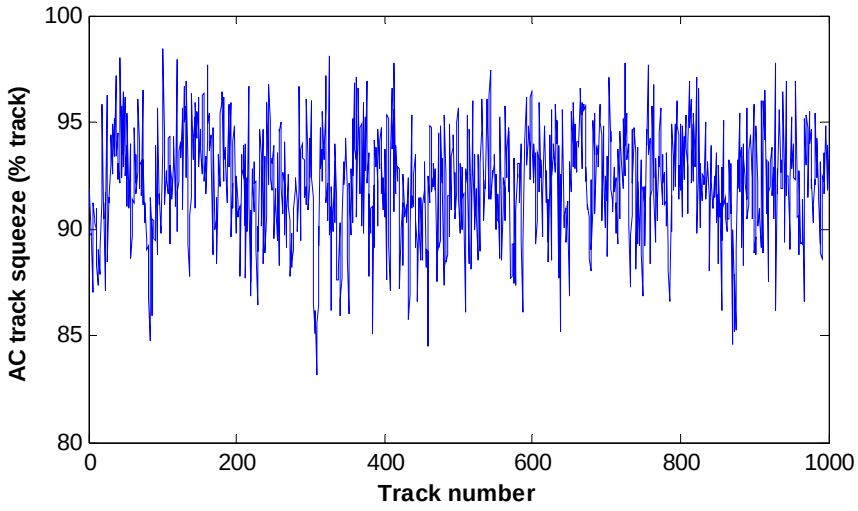

Figure 3.18: AC track squeeze profile

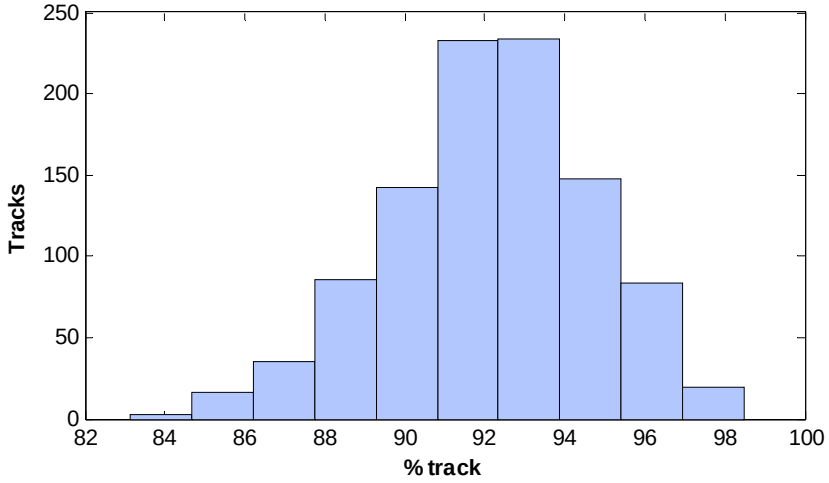

Figure 3.19: Histogram of AC track squeeze

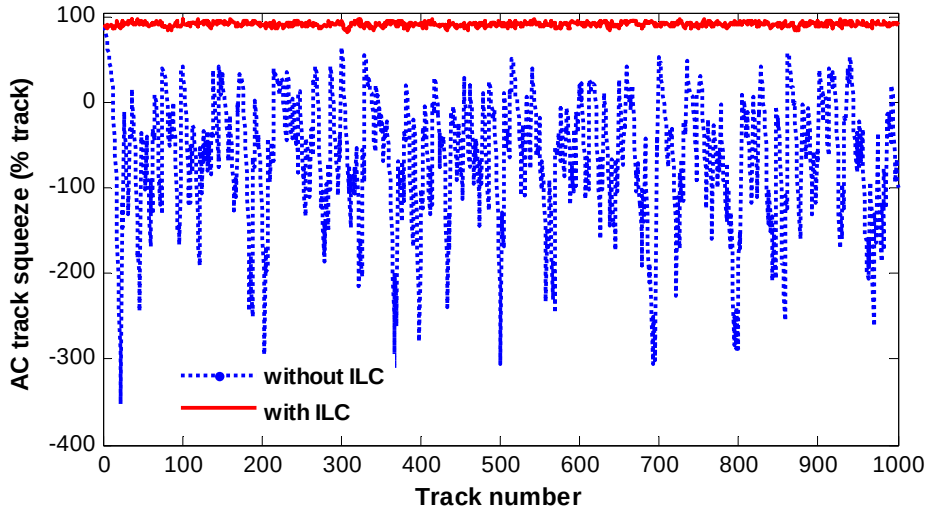

Figure 3.20: Comparison of AC track squeeze with and without ILC scheme

Figure 3.21 shows the comparison of the position error $e_i(k)$ in the systems with and without the proposed ILC scheme. By applying the ILC scheme as shown in the solid line, the resulting average $\sigma[e_i] = 3.27\%$ track width, which is much smaller than the result (52.4% track width) in the system without the ILC scheme as shown in the dotted line.

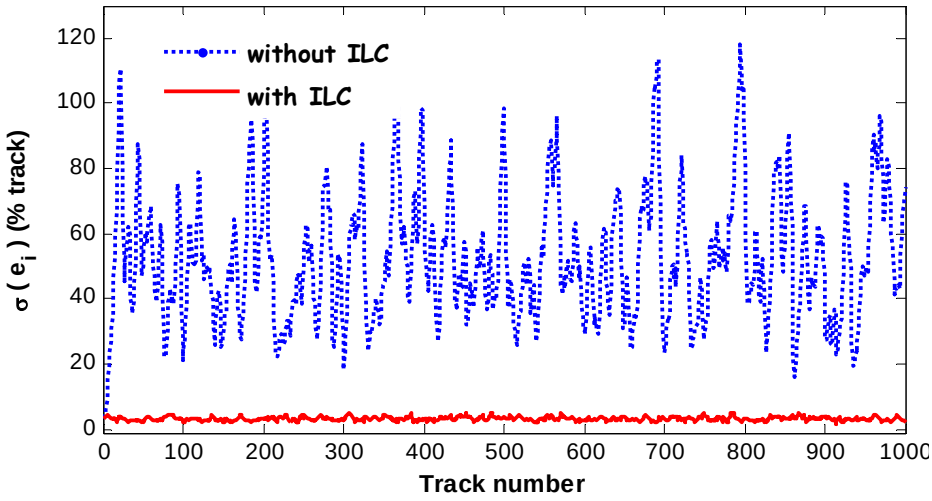

Figure 3.21: Comparison of position error with and without ILC scheme

3.6 Chapter Summary

In this chapter, an iterative learning control scheme was proposed to deal with the iteration-to-iteration error propagation issue in discrete linear repetitive processes.

The position control loop in HDD concentric SSW process was taken as an example. In this repetitive process, the current-track profile is used as the reference and amplified through the closed-loop dynamics to generate the next-track profile. As a result, the current-track error is also propagated to the next track and amplified. An optimization-based iterative learning control scheme was proposed to contain such error propagation. The previous-track error information was recorded and utilized for constructing the learning control law. The conditions of asymptotic convergence and monotonic convergence were derived and used for designing the learning filter. An L_1 optimal control problem was formulated by sense of minimizing the maximum magnitude of the written position deviation profile.

The effectiveness of the proposed control scheme was verified through the simulation study. The proposed learning controller shows good convergence property and the radial error propagation was contained well, which resulted in good-quality written tracks with small track distortion.

Chapter 4

Two-Dimensional Representations and Control Design

4.1 Introduction

Repetitive processes have dynamic behavior along two directions, i.e. the time and the iteration directions. This type of processes can be considered as two-dimensional (2D) systems in which the inputs, outputs, and internal states are all defined as functions of two independent variables. In other words, the dynamics of a 2D system is propagated along two independent directions.

In comparison to one-dimensional (1D) model, 2D model can describe a complex process such as repetitive process better. Another driving force for 2D systems theory development comes from the iterative learning control (ILC) design, which is undergoing rapid development with many successful applications for solving practical control engineering problems. An ILC system can be considered as a special 2D system with dynamic evolution along both the time and the iteration directions. The key difference from a general 2D system is that an ILC system has dynamic evolution repeated along the time over a finite duration. In [52], Roesser has developed a well-known state-space model for 2D discrete systems. Geng et al. [36] first pointed out the relationship between an ILC system and a linear 2D system in Roesser model and conducted the design and analysis of an ILC system from a 2D system viewpoint. A direct ILC synthesis based on 2D systems theory was proposed by Kurek and Zaremba [51]. On the basis of robust control theory developed for 2D system in Roesser model, the authors in [72] extended this idea by proposing an integrated design method for an ILC system. This design allows investigation of robust convergence along both the time and the iteration directions in an ILC design sense.

There are three benefits of treating a repetitive process from a 2D system viewpoint:

- In a 2D sense, the properties of a repetitive process along the time and the iteration directions can be investigated in both controller design and stability analysis.
- The convergence problems in a repetitive process can be translated to the stability problems in a 2D system.

- Designing the control scheme for a repetitive process in a 2D system frame can result in an iteration-wise error learning algorithm combined with a time-wise feedback control algorithm to ensure the control performance not only along the iterations but also over the time.

Over the past decades, 2D systems have received considerable attention in both theoretical and practical research.

- On the theoretical side, the early works [43, 55, 47, 44] mainly focused on the stability issues of 2D linear systems. Recently, a number of results on the robust control of 2D systems have been reported. On the basis of linear matrix inequality (LMI), Du and Xie [19] conducted a robust stability analysis and synthesized a robust controller for the 2D system in Fornasini-Marchesini (FM) model with norm-bounded parameter uncertainties. Guan et al. [35] and Xie et al. [86] extended the robust guaranteed cost control and robust H_∞ control of 1D systems to FM-type 2D discrete systems. For a 2D system in Roesser model, Du et al. [18] proposed a robust H_∞ control scheme and Lam et al. [53] synthesized a robust output feedback control design. Galkowski et al. [31, 32, 33, 34] have conducted a comprehensive analysis and control synthesis in both discrete and continuous context, for a class of repetitive processes described by the 2D Roesser model. All of the above works, however, were conducted based on the system stabilization, leading to conservative linear time invariant (LTI) 2D state feedback control. Different from the above stability based methods, Sebek and Kraus [75] investigated a 2D stochastic linear quadratic control (LQC) scheme via 2D polynomial technique.
- On the practical side, 2D modeling and system theories have been successfully applied in 2D digital signal processing [56], image processing [45], and mining processing [69].

There are two independent dynamic processes in a 2D system, in the context of HDD concentric SSW position control system. We use one of them to reflect the dynamics in the time domain and the other to reflect the dynamics in the iteration/ track domain. There are several motivations and significances of employing 2D systems theory to analyze and design controller for concentric SSW position control loop:

(a) Concentric SSW process is a 2D system in nature. The 2D Roesser model is able to reveal the 2D characteristics of this repetitive process, i.e. its dynamic propagation along the time (i.e. circumferential) and the iteration/track (i.e. radial) directions.

(b) By using the 2D Roesser model, the convergence problems in concentric SSW system can be solved by using the 2D systems theory, which provides a powerful approach to guarantee the stability and convergence of concentric SSW process in both the time and the iteration domains simultaneously.

This chapter presents an alternative control algorithm for concentric SSW position control loop. The remainder of this chapter is organized as follows. Section 4.2 reviews some useful definitions and theorems in 2D discrete linear systems. Section 4.3 describes

the concentric SSW position control loop by a 2D Roesser state-space model. A 2D control scheme is proposed in Section 4.4. Using the 2D systems theory, the 2D state feedback controller is designed together with an error learning filter to guarantee asymptotic stability and monotonic convergence of concentric SSW position control loop along both the time and the track directions. A design example is provided in Section 4.5. The effectiveness and feasibility of the proposed 2D control scheme are verified through the simulation study in Section 4.6. Finally, Section 4.7 gives the chapter summary.

4.2 2D Systems Theory Review

In this section, some useful definitions and theorems in 2D discrete linear systems are briefly reviewed.

4.2.1 2D Discrete Systems in Roesser Model

In a 2D Roesser state-space model [52], the local state of the system is decomposed into two components, namely the horizontal state x^h and the vertical state x^v, i.e. $x(k,i)=\left[x^h(k,i)\ \ x^v(k,i)\right]^T$, where the indexes k and i are non-negative integer-valued horizontal and vertical coordinates. $x^h \in \mathbb{R}^{n_1}$ and $x^v \in \mathbb{R}^{n_2}$ are the state components which are propagated horizontally and vertically. The compact matrix form of 2D Roesser state-space model is written as

$$\begin{aligned}\begin{bmatrix} x^h(k+1,i) \\ x^v(k,i+1) \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} x^h(k,i) \\ x^v(k,i) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}u(k,i) \\ y(k,i) &= \begin{bmatrix} C_1 & C_2 \end{bmatrix}\begin{bmatrix} x^h(k,i) \\ x^v(k,i) \end{bmatrix}\end{aligned} \tag{4.1}$$

where $u \in \mathbb{R}^r$ is the input vector and $y \in \mathbb{R}^m$ is the output vector of the 2D system. The boundary conditions of the system in Equation (4.1) are $x^h(0,i)$ and $x^v(k,0)$, for $i=0,1,2\cdots$ and $k=0,1,2\cdots$.

4.2.2 Bounded-Input Bounded-Output Stability of 2D Discrete Systems in Roesser Model

Stability is the most significant issue in 2D systems.

Definition 4.2.1 [52] The 2D discrete linear system in Equation (4.1) is said to be **bounded-input bounded-output (BIBO) stable** if and only if for every $M>0$, there exists an $N>0$ such that, if $\|u(k,i)\| \le M$ for all (k,i), then $\|y(h,j)\| \le N$ for all (h,j), where $\| v \|$ denotes a norm of v.

Theorem 4.2.1 [52] The following three statements are necessary conditions for the 2D Roesser model system in Equation (4.1) to be BIBO stable:

(1) A_{11} is stable;

(2) A_{22} is stable;

(3) $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ is stable.

Remark 4.2.1

- A square matrix is said to be stable if all of its eigenvalues lie in the interior of the unit circle in the complex plane.
- Theorem 4.2.1 is useful for checking the instability of the 2D system in Equation (4.1). However, it is not sufficient in general case.

The followings show how far Theorem 4.2.1 is from implying BIBO stability of the 2D system in Roesser model.

Now consider a nonsingular transformation for the state variable in Equation (4.1)

$$\begin{bmatrix} \underline{x}^h(k,i) \\ \underline{x}^v(k,i) \end{bmatrix} \triangleq \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} \begin{bmatrix} x^h(k,i) \\ x^v(k,i) \end{bmatrix} \triangleq T \begin{bmatrix} x^h(k,i) \\ x^v(k,i) \end{bmatrix} \tag{4.2}$$

which brings Equation (4.1) to

$$\begin{aligned} \begin{bmatrix} \underline{x}^h(k+1,i) \\ \underline{x}^v(k,i+1) \end{bmatrix} &= \begin{bmatrix} \underline{A}_{11} & \underline{A}_{12} \\ \underline{A}_{21} & \underline{A}_{22} \end{bmatrix} \begin{bmatrix} \underline{x}^h(k,i) \\ \underline{x}^v(k,i) \end{bmatrix} + \begin{bmatrix} \underline{B}_1 \\ \underline{B}_2 \end{bmatrix} u(k,i) \\ &= \underline{A} \begin{bmatrix} \underline{x}^h(k,i) \\ \underline{x}^v(k,i) \end{bmatrix} + \underline{B}\, u(k,i) \end{aligned} \tag{4.3}$$

$$y(k,i) = \begin{bmatrix} \underline{C}_1 & \underline{C}_2 \end{bmatrix} \begin{bmatrix} \underline{x}^h(k,i) \\ \underline{x}^v(k,i) \end{bmatrix} = \underline{C} \begin{bmatrix} \underline{x}^h(k,i) \\ \underline{x}^v(k,i) \end{bmatrix}$$

where

$$\underline{A} = T \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} T^{-1}, \quad \underline{B} = T \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad \underline{C} = \begin{bmatrix} C_1 & C_2 \end{bmatrix} T^{-1}$$

Theorem 4.2.2 [57] The following three statements are sufficient conditions for the 2D Roesser model in Equation (4.1) to be BIBO stable

(1) A_{11} and A_{22} are stable;

(2) A_{11} and A_{22} are diagonalizable and the transformation in Equation (4.2) is chosen such that $\underline{A}_{11}$ and $\underline{A}_{22}$ in Equation (4.3) are diagonal;

(3) $\|\underline{A}_{12}\|\|\underline{A}_{21}\| < (1-\lambda_{11})(1-\lambda_{22})$.

where $\lambda_{11} = \max_{1 \le i \le n_1} |\lambda_i(A_{11})|$, $\lambda_{22} = \max_{1 \le i \le n_2} |\lambda_i(A_{22})|$, $\lambda_i(\cdot)$ denotes the eigenvalues of a matrix, and $\|A\|$ is the induced norm of matrix A and defined as

$$\|A\| = \max_{1 \le i \le n} \sqrt{\lambda_i(A^T A)}$$

where n is the dimension of matrix A.

4.2.3 Asymptotic Stability of 2D Discrete Systems in Roesser Model

Definition 4.2.2 [52] The 2D Roesser model in Equation (4.1) is said to be **asymptotically stable** if and only if for zero-input (i.e. $u(k,i) = 0$) and any finite boundary conditions $x^h(0,i)$ and $x^v(k,0)$, $\sup_{k,i} \|x(k,i)\|$ is finite and the following equation holds

$$\lim_{\substack{k \to \infty \text{ and / or} \\ i \to \infty}} x(k,i) = \lim_{\substack{k \to \infty \text{ and / or} \\ i \to \infty}} \begin{bmatrix} x^h(k,i) \\ x^v(k,i) \end{bmatrix} = 0 \tag{4.4}$$

4.2.4 2D Transfer Function and Related Representations

By denoting z_1 and z_2 as the shift operators along the horizontal direction (described by the index k) and the vertical direction (described by the index i), we define the 2D z-transformation as

$$\begin{aligned} Y(z_1, z_2) &= Z\left[y(k,i)\right] \\ &= \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} y(k,i)\, z_1^{-k} z_2^{-i} \end{aligned} \tag{4.5}$$

Applying the above z-transformation definition to the 2D Roesser model in Equation (4.1) yields the 2-D transfer function matrix

$$\begin{aligned} G(z_1, z_2) &= \frac{Y(z_1, z_2)}{U(z_1, z_2)} \\ &= \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} z_1 I - A_{11} & -A_{12} \\ -A_{21} & z_2 I - A_{22} \end{bmatrix}^{-1} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \end{aligned} \tag{4.6}$$

The characteristic polynomial of the 2D Roesser model in Equation (4.1) is defined as

$$d(z_1, z_2) = \det \begin{bmatrix} z_1 I - A_{11} & -A_{12} \\ -A_{21} & z_2 I - A_{22} \end{bmatrix}$$

Theorem 4.2.3 [52] The 2D Roesser model in Equation (4.1) is asymptotically stable if and only if

$$d(z_1, z_2) = \det\begin{bmatrix} z_1 \cdot I_{n_1} - \mathrm{A}_{11} & -A_{12} \\ -A_{21} & z_2 \cdot I_{n_2} - A_{22} \end{bmatrix} \neq 0 \quad (|z_1| \geq 1 \text{ or } |z_2| \geq 1) \tag{4.7}$$

Applying matrix manipulations on Equation (4.7) yields

$$\left.\begin{aligned} & d(z_1, z_2) \\ = \ & \det(z_1 I_{n_1} - A_{11}) \cdot \det\left(z_2 I_{n_2} - \left[A_{22} + A_{21}(z_1 I_{n_1} - A_{11})^{-1} A_{12}\right]\right) \\ = \ & \det(z_2 I_{n_2} - A_{22}) \cdot \det\left(z_1 I_{n_1} - \left[A_{11} + A_{12}(z_2 I_{n_2} - A_{22})^{-1} A_{21}\right]\right) \end{aligned}\right\} \neq 0 \quad (|z_1| \geq 1 \text{ or } |z_2| \geq 1) \tag{4.8}$$

Notice that the asymptotic stability condition in Equation (4.8) can be reduced to the following two equivalent conditions:

$$\begin{cases} \det(z_1 I_{n_1} - A_{11}) \neq 0, \quad |z_1| \geq 1 \\ \det\left(z_2 I_{n_2} - \left[A_{22} + A_{21}(z_1 I_{n_1} - A_{11})^{-1} A_{12}\right]\right) \neq 0 \quad (|z_1| = 1 \text{ and } |z_2| \geq 1) \end{cases} \tag{4.9a}$$

or

$$\begin{cases} \det(z_2 I_{n_2} - A_{22}) \neq 0, \quad |z_2| \geq 1 \\ \det\left(z_1 I_{n_1} - \left[A_{11} + A_{12}(z_2 I_{n_2} - A_{22})^{-1} A_{21}\right]\right) \neq 0 \quad (|z_2| = 1 \text{ and } |z_1| \geq 1) \end{cases} \tag{4.9b}$$

Theorem 4.2.2 and Theorem 4.2.3 are useful for deriving the following significant theorem.

Theorem 4.2.4 [52] The 2D Roesser model in Equation (4.1) is asymptotically/BIBO stable if

$$\begin{cases} \|A_{11}\| < 1 \\ \|A_{22}\| + \|A_{21}\|\left(1 - \|A_{11}\|\right)^{-1} \|A_{12}\| < 1 \end{cases} \tag{4.10a}$$

or

$$\begin{cases} \|A_{22}\| < 1 \\ \|A_{11}\| + \|A_{12}\|\left(1 - \|A_{22}\|\right)^{-1} \|A_{21}\| < 1 \end{cases} \tag{4.10b}$$

4.3 2D Representation of Concentric SSW Position Control System

From the block diagram of concentric SSW position control system in Figure 2.4, the state-space description of the controlled plant voice coil motor (VCM) can be written as

$$\begin{aligned} x_i(k+1) &= A_p x_i(k) + B_p u_i(k) \\ y_{i+1}(k) &= C_p x_i(k) \end{aligned} \tag{4.11}$$

where $(A_p, B_p, C_p, 0)$ is the state-space realization of the plant VCM, and $u_i(k)$ represents its control input.

It can be seen that the dynamics in Equation (4.11) contains two independent components as shown in Figure 4.1. One is propagated in the time domain (i.e. along the circumferential direction) which is described by the index k; the other is propagated in the track domain (i.e. along the radial direction) which is described by the index i. From this point of view, the 2D Roesser model introduced in Section 4.2 is an appropriate mathematical model to describe the entire dynamics in concentric SSW position control system.

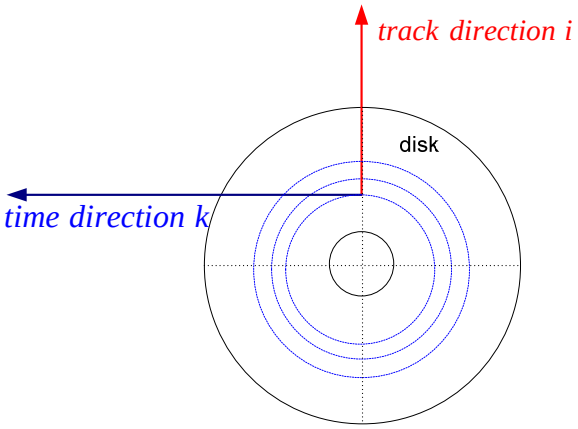

Figure 4.1: Illustration of 2D domains on the disk

Firstly, we define z_1 and z_2 as the shift operators along the time domain described by k and the track domain described by i, respectively, i.e.

$$\begin{aligned} y(k+1,i) &= z_1 \cdot y(k,i) \\ y(k,i+1) &= z_2 \cdot y(k,i) \end{aligned} \tag{4.12}$$

We also define $x(k,i)$ and $y(k,i)$ as the horizontal state and vertical state, respectively. Then the plant VCM in Equation (4.11) can be rewritten as the following 2D state-space equation

$$\begin{bmatrix} x(k+1,i) \\ y(k,i+1) \end{bmatrix} = \begin{bmatrix} A_p & 0 \\ C_p & 0 \end{bmatrix} \begin{bmatrix} x(k,i) \\ y(k,i) \end{bmatrix} + \begin{bmatrix} B_p \\ 0 \end{bmatrix} u(k,i) \tag{4.13}$$

The position error signal (PES) $e_i(k)$ in Figure 2.4 can be equivalently expressed as

$$\begin{aligned} e(k,i) &= y(k,i) - y(k,i+1) \\ &= y(k,i) - C_p x(k,i) \\ &= \begin{bmatrix} -C_p & 1 \end{bmatrix} \begin{bmatrix} x(k,i) \\ y(k,i) \end{bmatrix} \end{aligned} \tag{4.14}$$

Notice that Equation (4.13) and Equation (4.14) are state-space representation of a 2D Roesser model. In other words, these two equations form a 2D Roesser model $\overline{G}(z_1, z_2)$ with the state-space realization $\left(\begin{bmatrix} A_p & 0 \\ C_p & 0 \end{bmatrix}, \begin{bmatrix} B_p \\ 0 \end{bmatrix}, \begin{bmatrix} -C_p & 1 \end{bmatrix}, 0 \right)$, where $u(k,i)$ is the control input signal and $e(k,i)$ is the output signal.

Using the definition in Equation (4.12) and the 2D Roesser model $\overline{G}(z_1, z_2)$ in equations (4.13) – (4.14), the block diagram of concentric SSW position control loop in Figure 2.4 is reformulated into Figure 4.2.

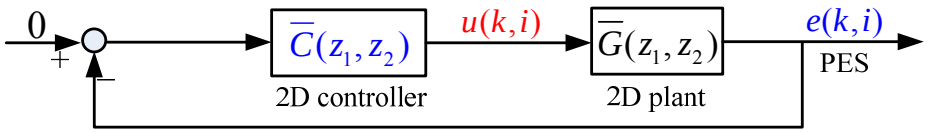

Figure 4.2: Block diagram of concentric SSW position control loop in 2D representation

In Figure 4.2 and equations (4.13)-(4.14), the signals and variables are defined as follows

- k : the servo sector index, i.e. $k = 0, 1, \cdots, N-1$;
- N : the number of servo sectors in one track;
- i : the data track index, i.e. $i = 0, 1, 2 \cdots$;
- $y(k,i)$: the read head position signal at the sector k in the track i;
- $y(k,i+1)$: the write head position signal at the sector k in the track $(i+1)$;
- $e(k,i)$: the position error signal (PES) at the sector k in the track i;
- $u(k,i)$: the control input at the sector k in the track i;
- $\overline{C}(z_1, z_2)$: the 2D controller to be designed;
- $\overline{G}(z_1, z_2)$: the 2D plant described by the 2D Roesser model in equations (4.13) - (4.14).

4.4 2D Controller Design for Concentric SSW Position Control Loop

4.4.1 Proposed Control Scheme

The control algorithm for the 2D Roesser model $\overline{G}(z_1, z_2)$ in Figure 4.2 is proposed as

$$u(k,i) = \begin{bmatrix} -K_1 & -K_2 \end{bmatrix} \begin{bmatrix} x(k,i) \\ y(k,i) \end{bmatrix} + \sum_{d=0}^{l} F_d \cdot e(k+d, i-1) \tag{4.15}$$

Remark 4.4.1

- K_1 is the constant feedback gain for the horizontal state $x(k,i)$.
- K_2 is the constant feedback gain for the vertical state $y(k,i)$.
- $F_d \left(d = 0, 1, 2, \cdots, l \right)$ is constant weight for the error signal $e(k+d, i-1)$ and $l+1$ is the error learning step.
- In Equation (4.15), the previous-track error $e(k, i-1)$ $\left(k = 0, 1, \cdots, N-1 \right)$ is utilized because all of $e(k, i-1)$ for $k \in [0, N-1]$ are available when self-servowriting is conducted in the current track i. However, the current-track error $e(k+d, i)$ is not available when $d \geq 1$.

It is known that only the error signal $e(k,i)$ is measurable in concentric SSW position control system, i.e. the 2D state vector $\begin{bmatrix} x(k,i) \\ y(k,i) \end{bmatrix}$ is not available for designing the controller. Hence, the control algorithm in Equation (4.15) is not implementable and it is necessary to design an observer and estimate the 2D state vector $\begin{bmatrix} x(k,i) \\ y(k,i) \end{bmatrix}$ first.

The 2D state observer can be designed as

$$\begin{aligned} \hat{x}(k+1,i) &= A_p \hat{x}(k,i) + B_p u(k,i) + L_1 \left(e(k,i) - \hat{e}(k,i) \right) \\ \hat{y}(k,i+1) &= C_p \hat{x}(k,i) + L_2 \left(e(k,i) - \hat{e}(k,i) \right) \end{aligned} \tag{4.16}$$

where $\hat{x}(k,i)$ and $\hat{y}(k,i)$ are the estimates of $x(k,i)$ and $y(k,i)$, respectively; L_1 and L_2 are constant observer gains.

The estimated error signal is written as

$$\hat{e}(k,i) = \begin{bmatrix} -C_p & 1 \end{bmatrix} \begin{bmatrix} \hat{x}(k,i) \\ \hat{y}(k,i) \end{bmatrix} \tag{4.17}$$

Thus, by introducing Equation (4.16) and Equation (4.17), the control algorithm in Equation (4.15) is modified as

$$
\begin{aligned}
u(k,i) &= \begin{bmatrix} -K_1 & -K_2 \end{bmatrix} \begin{bmatrix} \hat{x}(k,i) \\ \hat{y}(k,i) \end{bmatrix} + \sum_{d=0}^{l} F_d \cdot e(k+d, i-1) \\
&= -K_1 \cdot \hat{x}(k,i) - K_2 \cdot \hat{y}(k,i) + F_0 \cdot e(k,i-1) + F_1 \cdot e(k+1,i-1) + \ldots \\
&\quad + F_l \cdot e(k+l,i-1) \qquad (4.18) \\
&= -K_1 \cdot \hat{x}(k,i) - K_2 \cdot \hat{y}(k,i) + \left(F_0 + F_1 z_1 + F_2 z_1^2 + \ldots + F_l z_1^l \right) \cdot e(k,i-1) \\
&= -K_1 \cdot \hat{x}(k,i) - K_2 \cdot \hat{y}(k,i) + F(z_1) \cdot e(k,i-1)
\end{aligned}
$$

By applying the proposed control algorithm in Equation (4.18), the 2D system in Figure 4.2 can be illustrated by Figure 4.3.

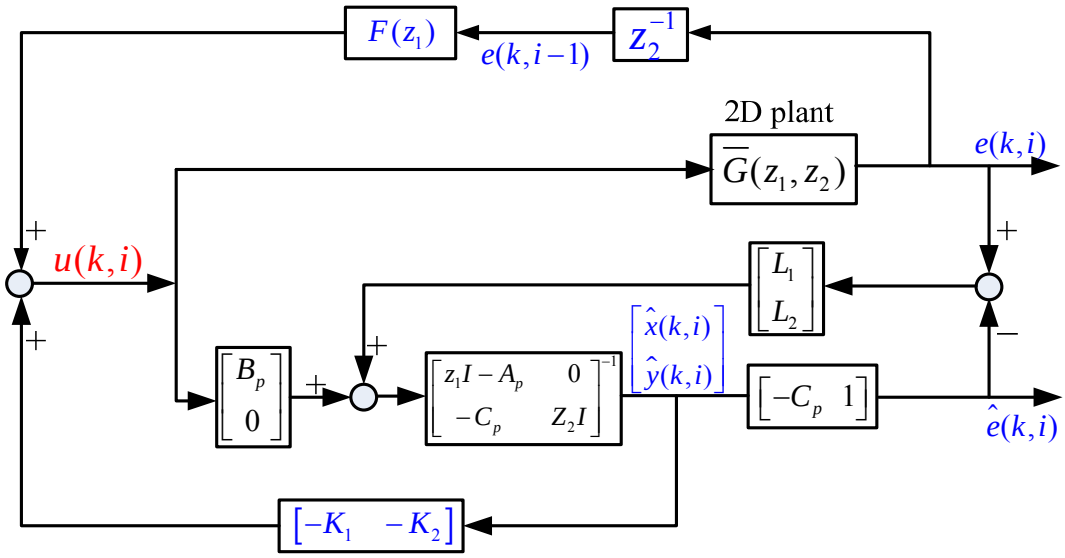

Figure 4.3: Block diagram of concentric SSW position control loop in 2D Roesser model with control algorithm (4.18)

Analysis

- In Equation (4.18), $F(z_1)$ is a non-causal error learning filter to be designed, which serves the same purpose as the error learning filter $F(z)$ in Equation (3.3). Notice that $F(z_1)$ in Equation (4.18) is in terms of only z_1; that means it is single-track error learning.
- Comparing Figure 4.3 with Figure 3.6, the gain K_1 for the horizontal state $x(k,i)$ functions as the feedback compensation in Figure 3.6 and K_2 for the vertical state $y(k,i)$ can be regarded as the feedforward compensation in Figure 3.6.

- Since the gain K_1 is designed together with K_2 (see the details in Section 4.5.2), the performance of the feedback control and feedforward control in the system is guaranteed simultaneously.

4.4.2 Design Example

In this subsection, a design example is provided. The control algorithm is designed with two-step error learning, i.e.

$$u(k,i) = \begin{bmatrix} -K_1 & -K_2 \end{bmatrix} \begin{bmatrix} \hat{x}(k,i) \\ \hat{y}(k,i) \end{bmatrix} + F_0 \cdot e(k,i-1) + F_1 \cdot e(k+1,i-1) \tag{4.19}$$

where F_0 and F_1 are constant filter coefficients to be designed.

By combining and reforming equations (4.13), (4.14), (4.16), (4.17), and (4.19), we obtain the closed-loop system equation as

$$\begin{bmatrix} x(k+1,i) \\ \hat{x}(k+1,i) \\ y(k+1,i) \\ \hat{y}(k+1,i) \end{bmatrix} = \begin{bmatrix} A_p & -B_pK_1 & 0 & -B_pK_2 \\ -L_1C_p & A_p - B_pK_1 + L_1C_p & L_1 & -L_1 - B_pK_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k,i) \\ \hat{x}(k,i) \\ y(k,i) \\ \hat{y}(k,i) \end{bmatrix} +$$
$$\begin{bmatrix} -B_pF_0C_p & 0 & B_pF_0 & 0 \\ -B_pF_0C_p & 0 & B_pF_0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k,i-1) \\ \hat{x}(k,i-1) \\ y(k,i-1) \\ \hat{y}(k,i-1) \end{bmatrix} +$$
$$\begin{bmatrix} -B_pF_1C_p & 0 & B_pF_1 & 0 \\ -B_pF_1C_p & 0 & B_pF_1 & 0 \\ C_p & 0 & 0 & 0 \\ -L_2C_p & C_p + L_2C_p & L_2 & -L_2 \end{bmatrix} \begin{bmatrix} x(k+1,i-1) \\ \hat{x}(k+1,i-1) \\ y(k+1,i-1) \\ \hat{y}(k+1,i-1) \end{bmatrix}$$

$$e(k,i) = \begin{bmatrix} -C_p & 1 \end{bmatrix} \begin{bmatrix} x(k,i) \\ y(k,i) \end{bmatrix} = \begin{bmatrix} C_p & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(k,i) \\ \hat{x}(k,i) \\ y(k,i) \\ \hat{y}(k,i) \end{bmatrix} \tag{4.20}$$

Notice that Equation (4.20) can be simplified to

$$\begin{aligned} \mathrm{X}(k+1,i) &= A_1\mathrm{X}(k,i) + A_2\mathrm{X}(k,i-1) + A_3\mathrm{X}(k+1,i-1) \\ e(k,i) &= C\mathrm{X}(k,i) \end{aligned} \tag{4.21}$$

where

$$\mathrm{X}(k,i) = \begin{bmatrix} x(k,i) & \hat{x}(k,i) & y(k,i) & \hat{y}(k,i) \end{bmatrix}^T,$$

$$A_1 = \begin{bmatrix} A_p & -B_pK_1 & 0 & -B_pK_2 \\ -L_1C_p & A_p - B_pK_1 + L_1C_p & L_1 & -L_1 - B_pK_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -B_pF_0C_p & 0 & B_pF_0 & 0 \\ -B_pF_0C_p & 0 & B_pF_0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; A_3 = \begin{bmatrix} -B_pF_1C_p & 0 & B_pF_1 & 0 \\ -B_pF_1C_p & 0 & B_pF_1 & 0 \\ C_p & 0 & 0 & 0 \\ -L_2C_p & C_p + L_2C_p & L_2 & -L_2 \end{bmatrix}$$

$$C = \begin{bmatrix} -C_p & 0 & 1 & 0 \end{bmatrix}$$

Equation (4.21) can be further written as

$$\begin{bmatrix} \mathrm{X}(k,i) \\ \mathrm{X}(k+1,i) \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & A_1 \end{bmatrix} \begin{bmatrix} \mathrm{X}(k-1,i) \\ \mathrm{X}(k,i) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ A_2 & A_3 \end{bmatrix} \begin{bmatrix} \mathrm{X}(k,i-1) \\ \mathrm{X}(k+1,i-1) \end{bmatrix}$$

$$e(k,i) = \begin{bmatrix} 0 & C \end{bmatrix} \begin{bmatrix} \mathrm{X}(k-1,i) \\ \mathrm{X}(k,i) \end{bmatrix}$$

Defining the augmented state vector $w(k,i) = \begin{bmatrix} \mathrm{X}(k-1,i) \\ \mathrm{X}(k,i) \end{bmatrix}$, we have

$$\begin{aligned} w(k+1,i) &= W_1 w(k,i) + W_2 w(k+1,i-1) \\ e(k,i) &= W_3 w(k,i) \end{aligned} \tag{4.22}$$

where,

$$W_1 = \begin{bmatrix} 0 & I \\ 0 & A_1 \end{bmatrix}, \; W_2 = \begin{bmatrix} 0 & 0 \\ A_2 & A_3 \end{bmatrix}, \; W_3 = \begin{bmatrix} 0 & C \end{bmatrix}$$

By defining the new state vectors

$$\begin{aligned} \phi(k,i) &= w(k,i) - W_2 \, w(k,i-1) \\ \gamma(k,i) &= W_2 w(k,i-1) \end{aligned}$$

we can recast Equation (4.22) into

$$\begin{bmatrix} \phi(k+1,i) \\ \gamma(k,i+1) \end{bmatrix} = \begin{bmatrix} W_1 & W_1 \\ W_2 & W_2 \end{bmatrix} \begin{bmatrix} \phi(k,i) \\ \gamma(k,i) \end{bmatrix}$$
$$e(k,i) = \begin{bmatrix} W_3 & W_3 \end{bmatrix} \begin{bmatrix} \phi(k,i) \\ \gamma(k,i) \end{bmatrix} \tag{4.23}$$

Notice that the system in Equation (4.23) is a 2D Roesser state-space model, where $\phi(k,i)$ is the horizontal state vector and $\gamma(k,i)$ is the vertical state vector. In other words, the closed-loop system in Figure 4.3 can be described by the 2D Roesser model (4.23) with the 2D state vector $\begin{bmatrix} \phi(k,i) & \gamma(k,i) \end{bmatrix}^T$ and output $e(k,i)$.

4.5 Controller Parameters Design

4.5.1 Observer Gains Design

The observer gains L_1 and L_2 are designed first.

By defining the 2D error state vectors as

$$\tilde{x}(k,i) = x(k,i) - \hat{x}(k,i)$$
$$\tilde{y}(k,i) = y(k,i) - \hat{y}(k,i)$$

we obtain the error dynamics from equations (4.13) – (4.14) and (4.16) – (4.19)

$$\begin{bmatrix} \tilde{x}(k+1,i) \\ \tilde{y}(k,i+1) \end{bmatrix} = \begin{bmatrix} A_p + L_1 C_p & -L_1 \\ C_p + L_2 C_p & -L_2 \end{bmatrix} \begin{bmatrix} \tilde{x}(k,i) \\ \tilde{y}(k,i) \end{bmatrix}$$

Notice that the above equation is a 2D Roesser model. In order to make sure that the error dynamics is asymptotically/BIBO stable, we apply Theorem 4.2.4 and the following conditions are obtained:

(a) $\left\| A_p + L_1 C_p \right\| < 1$ (4.24a)

(b) $\left\| L_2 \right\| + \left\| C_p + L_2 C_p \right\| \left(1 - \left\| A_p + L_1 C_p \right\| \right)^{-1} \left\| L_1 \right\| < 1$ (4.24b)

Analysis

- Equation (4.24a) implies that

$$
\begin{aligned}
& \left\| A_p + L_1 C_p \right\| < 1 \\
\Leftrightarrow\ & \overline{\sigma}\left(A_p + L_1 C_p \right) < 1 \\
\Leftrightarrow\ & \sqrt{\lambda_{\max}\left(A_p + L_1 C_p \right)\left(A_p + L_1 C_p \right)^T} < 1 \\
\Leftrightarrow\ & I_{np\times np} - \left(A_p + L_1 C_p \right)\left(A_p + L_1 C_p \right)^T \succ 0 \\
\Leftrightarrow\ & \begin{bmatrix} I_{np\times np} & \left(A_p + L_1 C_p \right) \\ \left(A_p + L_1 C_p \right)^T & I_{np\times np} \end{bmatrix} \succ 0
\end{aligned}
\tag{4.24c}
$$

where n_p is the dimension of A_p.

As a result, the design of L_1 satisfying Equation (4.24a) is equivalent to solving the linear matrix inequality (LMI) feasibility problem in Equation (4.24c), which can be solved by a standard LMI solver [73] or semidefinite programming (SDP) solver in Yalmip package.

- L_2 is designed through solving Equation (4.24b), which is also a feasibility problem and can be solved easily by the optimization programming.

4.5.2 State Feedback Gains Design

When designing the 2D state feedback gains K_1 and K_2, we assume that the control algorithm only contains the state feedback term, i.e.

$$
u(k,i) = \begin{bmatrix} -K_1 & -K_2 \end{bmatrix} \begin{bmatrix} \hat{x}(k,i) \\ \hat{y}(k,i) \end{bmatrix} \tag{4.25}
$$

Applying the 2D state feedback control algorithm in Equation (4.25) to the 2D plant $\overline{G}(z_1, z_2)$ in equations (4.13) – (4.14) yields the closed-loop system equation

$$
\begin{bmatrix} x(k+1,i) \\ \hat{x}(k+1,i) \\ y(k,i+1) \\ \hat{y}(k,i+1) \end{bmatrix} = \begin{bmatrix} A_p & -B_p K_1 & 0 & -B_p K_2 \\ -L_1 C_p & A_p - B_p K_1 + L_1 C_p & L_1 & -B_p K_2 - L_1 \\ C_p & 0 & 0 & 0 \\ -L_2 C_p & C_p + L_2 C_p & L_2 & -L_2 \end{bmatrix} \begin{bmatrix} x(k,i) \\ \hat{x}(k,i) \\ y(k,i) \\ \hat{y}(k,i) \end{bmatrix} \tag{4.26}
$$

By defining $\bar{x}(k,i) = \begin{bmatrix} x(k,i) \\ \hat{x}(k,i) \end{bmatrix}$ as the horizontal state vector and $\bar{y}(k,i) = \begin{bmatrix} y(k,i) \\ \hat{y}(k,i) \end{bmatrix}$ as the vertical state vector, we rewrite Equation (4.26) as

$$
\begin{bmatrix} \bar{x}(k+1,i) \\ \bar{y}(k,i+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \bar{x}(k,i) \\ \bar{y}(k,i) \end{bmatrix} \tag{4.27}
$$

where

$$A_{11}=\begin{bmatrix} A_p & -B_pK_1 \\ -L_1C_p & A_p-B_pK_1+L_1C_p \end{bmatrix},\ A_{12}=\begin{bmatrix} 0 & -B_pK_2 \\ L_1 & -B_pK_2-L_1 \end{bmatrix},$$

$$A_{21}=\begin{bmatrix} C_p & 0 \\ -L_2C_p & C_p+L_2C_p \end{bmatrix},\ A_{22}=\begin{bmatrix} 0 & 0 \\ L_2 & -L_2 \end{bmatrix}$$

Notice that the system in Equation (4.27) is a 2D Roesser model. Hence, in order to fulfill Theorem 4.2.4, the gains K_1 and K_2 must meet the following conditions:

$$\text{(a) } \|A_{11}\| = \left\| \begin{bmatrix} A_p & -B_pK_1 \\ -L_1C_p & A_p-B_pK_1+L_1C_p \end{bmatrix} \right\| < 1 \tag{4.28a}$$

$$\text{(b) } \|A_{22}\|+\|A_{21}\|\left(1-\|A_{11}\|\right)^{-1}\|A_{12}\| < 1 \tag{4.28b}$$

Analysis

- Since $A_{11}=\begin{bmatrix} A_p & -B_pK_1 \\ -L_1C_p & A_p-B_pK_1+L_1C_p \end{bmatrix}$ satisfies the separation principle, i.e. the $2\times np$ eigenvalues of A_{11} consist of *np* eigenvalues of $(A_p-B_pK_1)$ and *np* eigenvalues of $(A_p+L_1C_p)$. Therefore, Equation (4.28a) implies that

$$\begin{aligned} \|A_p-B_pK_1\| &< 1 \\ \|A_p+L_1C_p\| &< 1 \end{aligned} \tag{4.28c}$$

  Thus, similar to designing L_1, the state feedback gain K_1 is designed by solving the LMI feasibility problem

$$\begin{bmatrix} I_{np\times np} & \left(A_p-B_pK_1\right) \\ \left(A_p-B_pK_1\right)^T & I_{np\times np} \end{bmatrix} \succ 0$$

- A feasible K_2 can be derived by solving the feasibility problem in Equation (4.28b) using the optimization programming.

4.5.3 Error Learning Filter Design

After the observer gains L_1, L_2 and the state feedback gains K_1, K_2 are obtained, the error learning filter coefficients F_0 and F_1 are found to satisfy the stability condition and reduce the track-to-track error propagation.

By using Theorem 4.2.4 again, the following conditions are derived to guarantee asymptotic/BIBO stability of the 2D Roesser system in Equation (4.23):

(a) $\|W_1\| < 1$ (4.29a)

(b) $\|W_2\| + \|W_2\|(1-\|W_1\|)^{-1}\|W_1\| < 1$ (4.29b)

Analysis

- Using the gains L_1, L_2, K_1, and K_2 obtained in sections 4.5.1 and 4.5.2, it must be verified that $\|W_1\| < 1$ is satisfied. If it is not satisfied, a new K_2 should be obtained by solving both (4.28b) and (4.29b), while using the same L_1, L_2 and K_1.
- The feasible values of F_0 and F_1 are derived by solving the feasibility problem in Equation (4.29b).

4.6 Simulation Study

In the simulation study, the disk disturbance and sensor noise are considered as shown in Figure 4.4, which are the same as that used in Section 3.4. In the figure, $n(k,i)$ represents the sensor noise and $d(k,i)$ represents the disk/spindle disturbance.

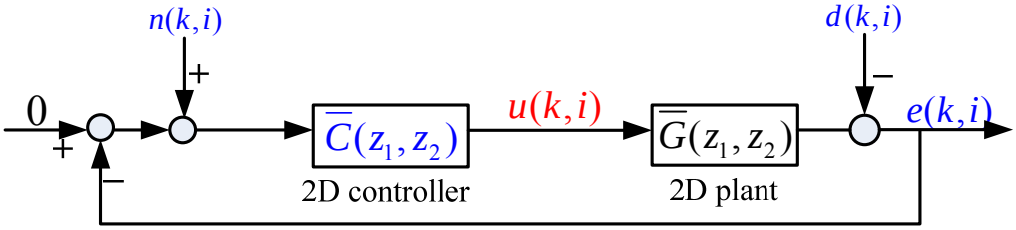

Figure 4.4: Block diagram of 2D concentric SSW position control loop with disk disturbance and sensor noise

The HDD model used for the simulation study is the same as that used in Chapter 3, which is shown in Figure 3.8. The 2D controller parameters $(K_1, K_2, L_1, L_2, F_0, F_1)$ are listed in Table 4.1. They satisfy the conditions (4.24a-c), (4.28a-c), and (4.29a-b) as stated in the previous section.

Table 4.1: Controller parameters $(K_1, K_2, L_1, L_2, F_0, F_1)$

K_1	$[-0.806 \ -0.916 \ 0.027 \ -0.142 \ -0.023 \ 0.298 \ -0.117 \ -0.140 \ -0.147 \ -0.051 \ -0.047 \ 0.158]$
K_2	2.208×10^{-3}
L_1	$[0.228 \ -0.231 \ 0.005 \ 0.024 \ -0.004 \ -0.051 \ -0.019 \ 0.023 \ -0.023 \ -0.008 \ 0.006 \ 0.022]$
L_2	-0.468
F_0	0.662
F_1	-4.048×10^{-3}

The seed track $y(k,0)$ is also the same as that used in the ILC design (as shown in Figure 3.13). In the simulation study, 1000 servo tracks are written.

Figure 4.5 shows the written track profiles in 2D domain (i.e. on the entire disk surface). It is obvious that the position deviation from the write head $y(k,i)$ greatly reduces and converges after the seed track.

In order to show the feasibility of the proposed 2D control scheme, the following simulation study compares its results with the ILC scheme, which were presented in Section 3.4.

Figure 4.6 shows the comparison of the written track profile for 1000 tracks. The solid line represents the write head profile in the system using the proposed 2D scheme, and the dotted line represents the profile in the system using the ILC scheme. It can be seen that the write head position deviation in 2D scheme has an average $\sigma[y_i]$ of 1.73% track, which is smaller than the result in the system using the ILC scheme (with average $\sigma[y_i] = 2.41\%$ track).

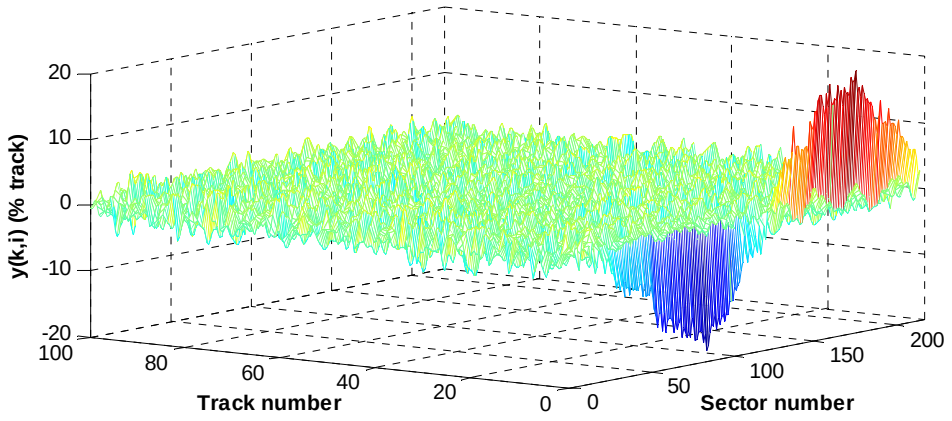

Figure 4.5: Written track profile in 2D domain

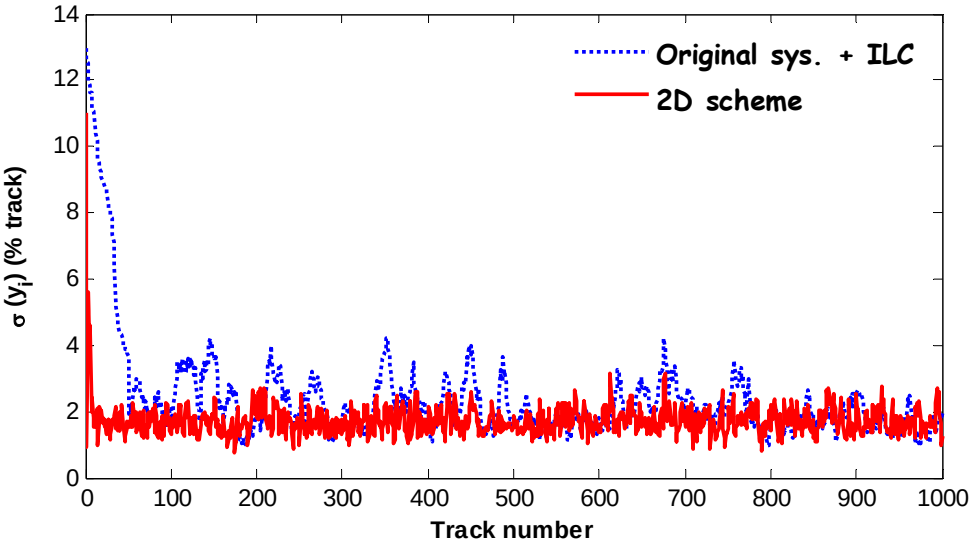

Figure 4.6: Comparison of written track profile

The comparison of the AC track squeeze (T_{sq}) is shown in Figure 4.7. In the system using the proposed 2D control scheme, the average AC track squeeze is 93.30% track width as shown in the solid line, which is slightly higher than that in the system using the ILC scheme with the average T_{sq} of 92.10% track width, as shown by the dotted line. The smaller write head position deviation and higher AC track squeeze proves good performance of the proposed 2D control scheme.

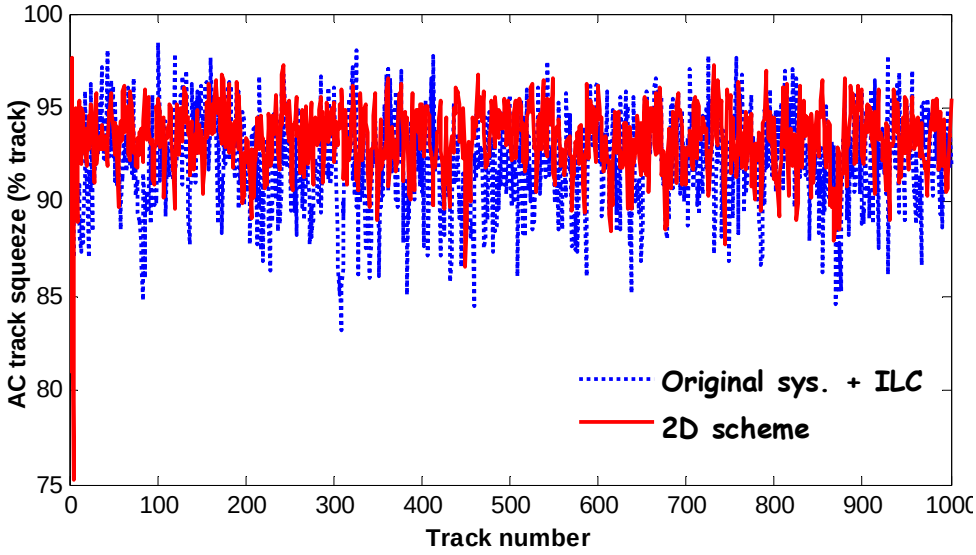

Figure 4.7: Comparison of AC track squeeze

Figure 4.8 shows the PES comparison in the systems using the proposed 2D scheme and the ILC scheme. From the figure, we can clearly see that the PES in the proposed 2D control scheme (as shown in the solid line) is smoother and has a smaller average $\sigma[e_i]$ of 2.33% track width than that in the ILC scheme (as shown in the dotted line) with average $\sigma[e_i] = 3.27\%$ track.

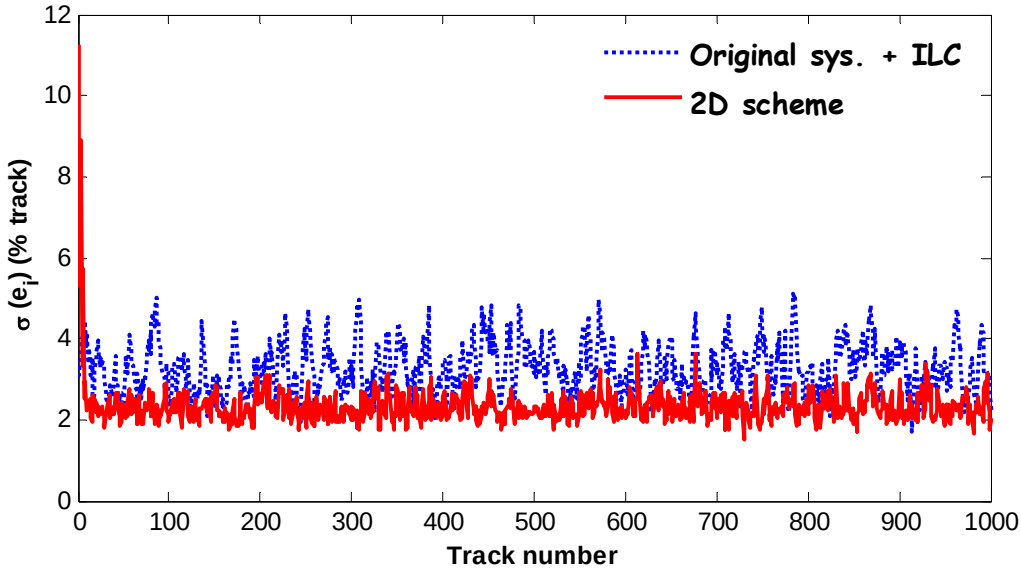

Figure 4.8: Comparison of PES profile

4.7 Chapter Summary

In this chapter, an alternative control approach was presented to deal with the iteration-to-iteration error propagation in discrete linear repetitive processes. The analysis and design were conducted on the position control loop in HDD concentric SSW process, which is the same system used in Chapter 3.

Firstly, this repetitive process was equivalently described by a 2D Roesser state-space model. By applying the 2D systems theory, the convergence problems in this repetitive process were translated to the stability problems in a 2D system.

Then, a 2D state feedback controller was designed to guarantee asymptotic stability in both the time and the iteration domains. An error learning filter was also added later for the purpose of containing the radial error propagation from track to track, which learns the previous-track error information and guarantees the convergence along the iteration direction.

The effectiveness of the proposed 2D control scheme was verified by comparing with the iterative leaning control scheme which was presented in Chapter 3. It was shown that the performance in the system using the proposed 2D control scheme is comparable to that in the system using the iterative learning control scheme. Both control schemes

provide much improvement with small write head position deviation and good quality of the written servo tracks.

Chapter 5

A Novel Adaptive Feedforward Control Scheme Design using Filtered-X Least Mean Square Algorithm

5.1 Introduction

In this chapter, a novel control scheme based on adaptive feedforward control (AFC) is proposed for the timing control loop in concentric SSW process, which is a good example of discrete linear repetitive process with error propagation along both the time and the iteration directions. In Section 2.3.2, we learned that in concentric SSW servo system, the timing control loop has the same block diagram (as shown in Figure 5.11) as the position control loop (as shown in Figure 2.4). However, each of these two repetitive systems has its own control issues and objectives. In the timing control loop, there are two main issues: one is the timing error propagation from track to track (which is analogous to the radial error propagation in the position control loop); the other is the closure error within each individual track (the details are presented in Section 5.3). The iterative learning control scheme presented in Chapter 3 and the 2D control scheme presented in Chapter 4 are not suitable for the timing control loop because these two methods can deal with the track-to-track timing error propagation well but have poor ability of containing the closure error within each individual track. Therefore, we need to design a control scheme which is able to effectively deal with both of the two issues in timing control loop.

The timing control problem in concentric SSW process is analogous to a typical problem in manufacturing. Most manufacturing processes, including machining, are subject to one or more sources of disturbance and process noise that lead to inaccuracy in manufactured parts. Disturbance can be somewhat rejected by making the process less sensitive to it using the passive means. Such treatments include, but not limited to, the use of precise components and massive machine tool structures. Another way of rejecting disturbance is the use of active techniques. This is based on the premise of generating some control action(s) leading to elimination or reduction on the effects of disturbances. If the disturbance, or some attributes of it, is measurable, the feedforward controllers could be used very effectively, knowing the mathematical model of the process. When

the process model is unknown or it has some time-varying parameters, the adaptive feedforward techniques are well suited. The adaptive control problem can be stated as follows:

Take as given a plant whose parameters are unknown or time-varying. Then construct an finite impulse response (FIR) filter such that the output of a reference model $S(z)$, $z(k)$, will adaptively follow (in the mean square error (MSE) sense) the output of the plant, $r(k)$. The timing index is k. The basic scheme of constructing an adaptive filter is shown in Figure 5.1 [28, 13, 42], which uses the standard least mean square (LMS) algorithm.

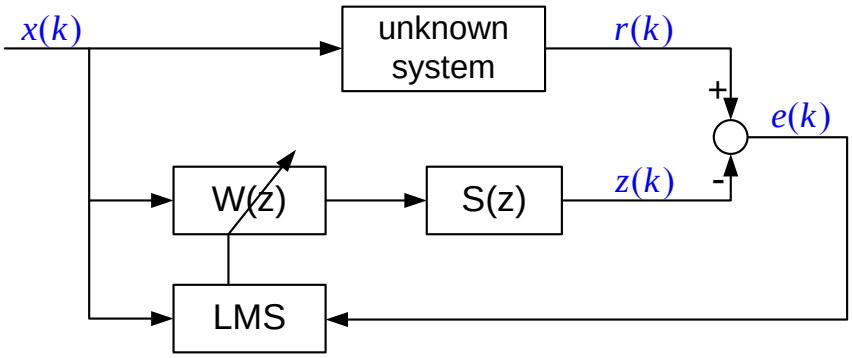

Figure 5.1: Basic block diagram of adaptive filter system using LMS algorithm

Notice that in Figure 5.1, the introduction of the secondary-path transfer function $S(z)$ generally causes instability [24]. This is because the error signal $e(k)$ is not correctly aligned in time with the reference signal $x(k)$, due to the presence of $S(z)$. There are a number of possible schemes that can be used to compensate for the effect of $S(z)$. Morgan [59] suggested two approaches to solve this problem. The first solution is to place an inverse filter, $1/S(z)$, in series with $S(z)$ to remove its effect. The second solution is to place an identical filter in the reference signal path to the weight update of the LMS algorithm, which realizes the so-called filtered-x least mean square (FXLMS) algorithm [80], as shown in Figure 5.2. Since an inverse does not necessarily exist for $S(z)$, the FXLMS algorithm is more effective. The FXLMS algorithm was independently developed by Widrow [81] in the context of adaptive control and Burgess [11] for active noise control (ANC) applications.

In this chapter, a novel control scheme based on AFC is proposed for concentric SSW timing control loop to minimize the closure error within each individual track and contain the track-to-track timing error propagation. The adaptive filter is designed by applying the filtered-x least mean square (FXLMS) algorithm.

The remainder of this chapter is organized as follows. Section 5.2 presents the basic operations of phase locked loop (PLL) circuit in the timing control loop. The control problems are described in Section 5.3. The proposed adaptive feedforward control scheme using FXLMS algorithm is presented in Section 5.4. In Section 5.5, the

simulation results show the effectiveness of the proposed control scheme. Finally, the chapter summary is given in Section 5.6.

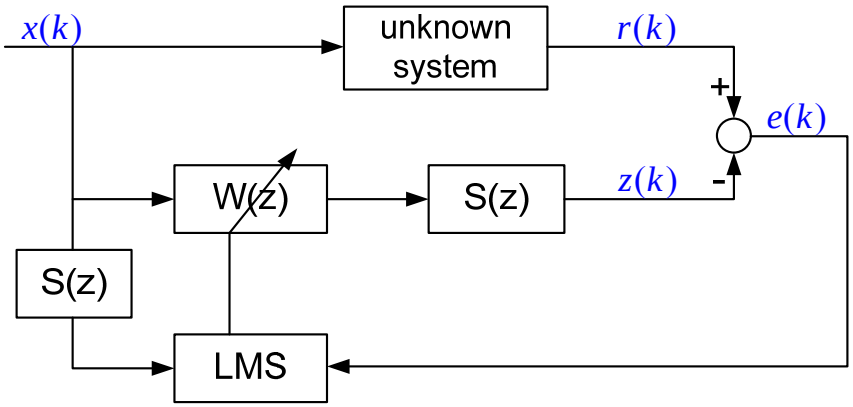

Figure 5.2: Black diagram of adaptive filter using FXLMS algorithm

5.2 Timing Control Loop in Concentric Self-Servowriting Process

In concentric SSW process, in order to ensure proper alignment of the servo patterns, the propagation of servo sectors should be synchronized to the timing marks in the seed clock track or the previously written track. In general, the seed clock track is prewritten at the outer diameter (OD) of the disk by a separated clock head before the SSW process. In SSW timing control loop, a phase locked loop (PLL) circuit (as shown in Figure 5.3) is generally employed to generate a stable and accurate clock signal in the presence of media defects and disk/spindle disturbances.

Basic operations of phase locked loop [1]

The PLL is a circuit which is used for tracking a particular signal. More precisely to say, PLL is a feedback system which is used for locking the reference signal in phase as well as in frequency. It is like the cruise control system for the motor vehicles. PLL has been widely used in our daily life, such as radio, telecommunications, and data storage systems. For example, when we turn on a TV set, one PLL will keep the human heads at the top of the screen while the feet at the bottom of screen. Another PLL makes sure that the green color remains green and red color remains red. The most basic block diagram of a PLL circuit in HDD is shown in Figure 5.3, which shows the components that every PLL must have, namely:

- A phase detector (PD) (or phase comparator). This is a nonlinear device whose output contains the phase difference between the oscillator output signal and the reference signal.
- A voltage controlled oscillator (VCO). This is another nonlinear device which generates the write clock signal used for propagating the servo sectors across the disk during SSW process.

- A loop filter (LF). This is generally designed as a low pass filter and filters the phase error from the PD to generate the voltage control signal for the VCO.

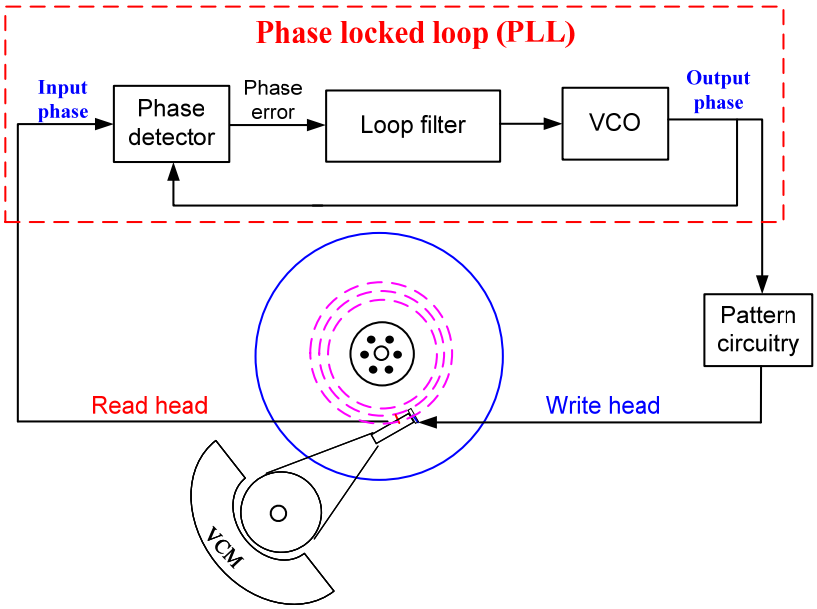

Figure 5.3: PLL circuit in SSW timing control loop

As shown in Figure 5.4, the basic idea of a PLL is that if one injects a sinusoidal signal into the reference input, the internal oscillator in the loop will lock to the reference sinusoid in such a way that the frequency and phase differences between the reference sinusoid and the internal sinusoid will be driven to zero or some constant value (depending on the system type). The internal sinusoid then represents a smoothed version of the reference sinusoid, which is filtered by the low pass filter $C(s)$.

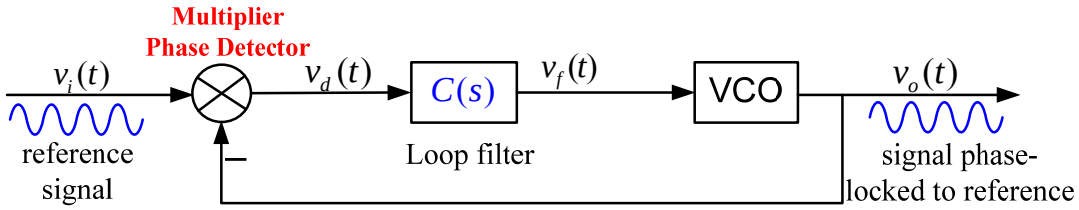

Figure 5.4: A classic mixing PLL

A general sinusoidal signal at the reference input of a PLL as shown in Figure 5.4 can be written as

$$v_i(t) = A\sin(\omega_i t + \theta_i) \tag{5.1}$$

where ω_i and θ_i are frequency and phase of the input signal; A is its amplitude.

Without loss of generality, we can assume that the output signal from the VCO into the mixer is given by

$$v_o(t) = \cos(\omega_o t + \theta_o) \tag{5.2}$$

where ω_o and θ_o are the frequency and phase of the output signal.

The output of the mixer (multiplier PD) in Figure 5.4 is then given by

$$v_d(t) = AK_m \sin(\omega_i t + \theta_i)\cos(\omega_o t + \theta_o) \tag{5.3}$$

where K_m is the gain of the mixer.

Typically, analysis of such a PLL is done by taking several simplifying steps. Using the familiar trigonometric identity in terms of the PLL

$$\begin{aligned} & 2\sin(\omega_i t + \theta_i)\cos(\omega_o t + \theta_o) \\ = & \left[\sin\left((\omega_i + \omega_o)t + (\theta_i + \theta_o)\right) + \sin\left((\omega_i - \omega_o)t + (\theta_i - \theta_o)\right)\right] \end{aligned} \tag{5.4}$$

and then making two fundamental assumptions leads to the commonly used model of the analog PLL. Let $\theta_d = \theta_i - \theta_o$, then these assumptions are

(1) The first term in Equation (5.4) is attenuated by the loop filter (which is a low pass filter) in Figure 5.4.

(2) $\omega_i \approx \omega_o$, so that the difference can be incorporated into θ_d. This means that the VCO can be mathematically modeled as an integrator.

Making these assumptions leads to the PLL model shown in Figure 5.5, where $K_d = AK_m / 2$.

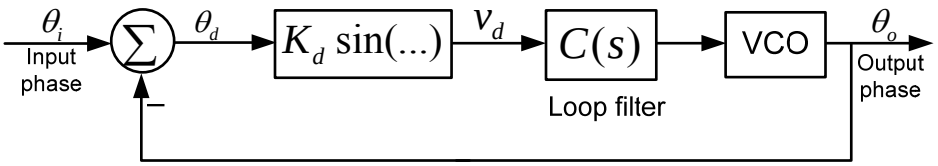

Figure 5.5: Conceptual block diagram of PLL with sine detector

Now the problem is that this is still a nonlinear system, and as such is in general difficult to analyze. The typical method of analysis is linearization. For θ_d small and slowing varying, the following holds

$$\sin(\theta_d) \approx \theta_d$$

while this is useful for studying loops that are near locked, it does not help for analyzing the loop when θ_d is large.

The linearized model is shown in Figure 5.6. This is what is used for most analysis and measurements of PLL circuits.

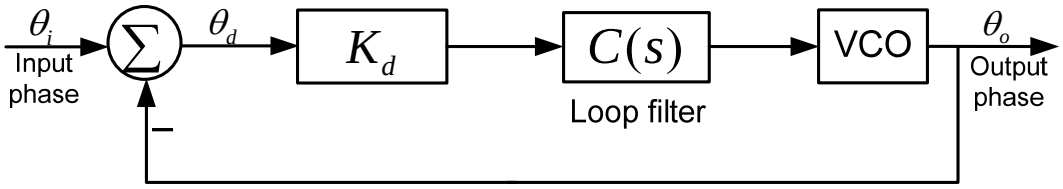

Figure 5.6: Block diagram of linearized PLL circuit

In SSW timing control loop as shown in Figure 5.3, the input of PLL circuit is the phase of read head signal and the output is the phase of write head signal. In industry, a good PLL circuit is required to synchronize the write head signal with the read head signal.

5.3 Control Problem Formulation

5.3.1 Phase and Timing Jitter Definitions

Without loss of generality, we assume that there are N servo sectors in one data track, and their timing marks on the disk are denoted by $t(0)$, $t(1)$,…, $t(N-1)$ as shown in the solid lines in Figure 5.7. The timing marks are desired to be uniformly and accurately written along the circular track. But in practical manufacturing process, the spacing between the adjacent timing marks is not uniform due to the timing jitters. Let $t'(0)$, $t'(1)$,…, $t'(N-1)$ denote the actual timing marks as shown in the dotted lines in Figure 5.7.

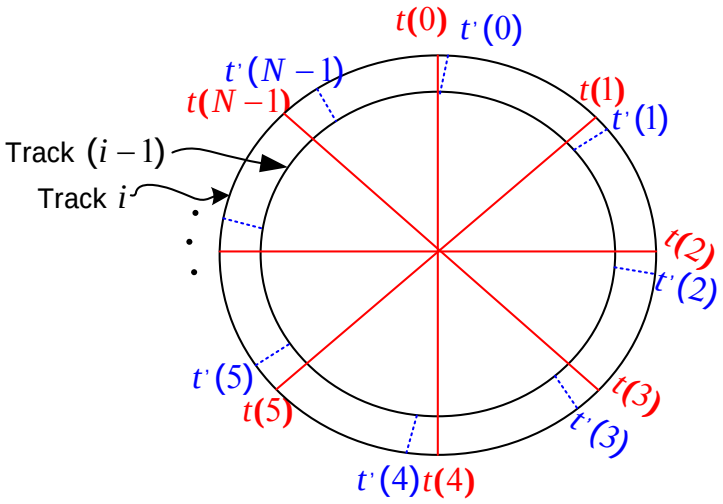

Figure 5.7: Illustration of ideal and actual timing marks of servo sectors

Definition 5.3.1 The difference between the ideal timing mark $t(k)$ and the actual timing mark $t'(k)$ is called the **timing jitter** at the servo sector k, i.e.

$$\Delta t(k) \triangleq t'(k) - t(k) \tag{5.5}$$

Definition 5.3.2 The **phase** at the servo sector k is defined as

$$\phi(k) \triangleq \frac{\left(t'(k) - t(k)\right) \cdot 2\pi}{T_c} \tag{5.6}$$

where T_c denotes the nominal sampling period.

Notice that Equation (5.6) implies that the phase is the normalized timing jitter.

5.3.2 Control Issues and Control Objectives

In order to write the servo sectors uniformly and accurately along the circular track, it is desired to eliminate the timing jitters. For this purpose, an effective control scheme is necessary to resolve the issues in concentric SSW timing control loop. There are two main issues in this repetitive control system:

(1) The first and most important issue is the closure error within each individual track, which refers to the timing error propagation along the circumferential direction. As shown in Figure 5.8, d_k denotes the angular distance between two adjacent timing marks $t'(k-1)$ and $t'(k)$. Wherein d_N, the angular distance between the first timing mark $t'(0)$ and the last timing mark $t'(N-1)$, is named the timing closure. From Figure 5.8, we know that the ideal angular distance between two adjacent timing marks is $2\pi/N$. The closure error denoted by Δd is defined as

$$\Delta d \triangleq d_N - 2\pi/N$$

In SSW process, the closure error Δd is generated if the servo sectors are not uniformly and accurately written along the circular tracks. It is unwanted because it adversely disturbs the PLL circuit operation and affects the propagation of the servo patterns.

Remark 5.3.1

(a) In a circular track, if $\Delta d > 0$, i.e. $d_N > 2\pi/N$, such a track is said to be “unclosed”.

(b) In a circular track, if $\Delta d < 0$, i.e. $d_N < 2\pi/N$, the pattern of the last servo sector may be incomplete or squeezed.

(c) Both of the above cases are caused by the timing error propagation within one track along the circumferential (i.e. time) direction.

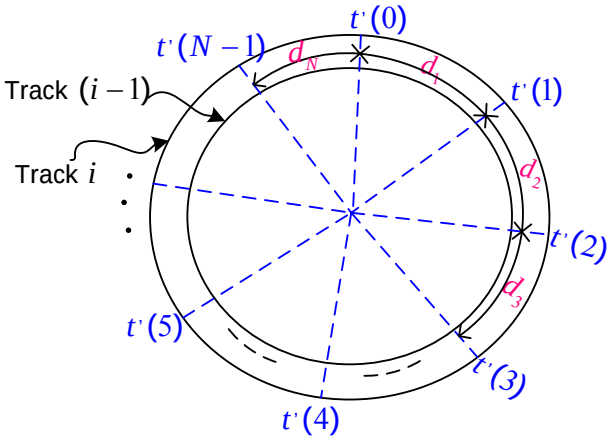

Figure 5.8: Illustration of angular distance between adjacent sectors

(2) The other significant issue in the timing control loop is track-to-track timing error propagation which is also known as the phase propagation along the radial direction. In the PLL circuit, when the write head is synchronized with the read head to propagate the servo sectors, the disk flutter and spindle speed varaition are amplified by the loop dynamics. These components of timing jitters will propagate from track to track and grow unboundedly, resulting in 'wandering' or 'warping' servo sectors across the disk surface, as shown in Figure 5.9. This deteriorates the quality of the written servo patterns. Figure 5.10 shows an example of timing error propagation. It shows that the written timing deviation grows rapidly if this issue is not well handled.

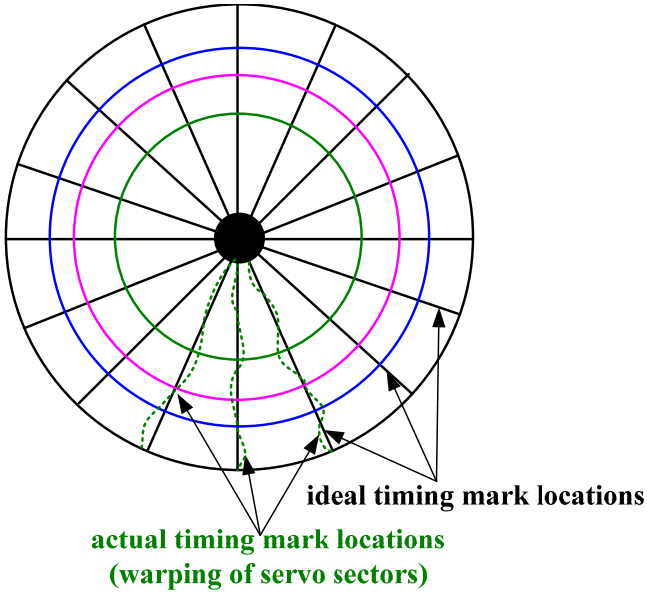

Figure 5.9: Warping of servo sectors on the disk

Therefore, in order to deal with the above two issues and improve the quality of the written servo patterns, an effective control scheme is required to attenuate the closure error within each individual track (along the time direction) as well as the timing error propagation from track to track (along the track direction).

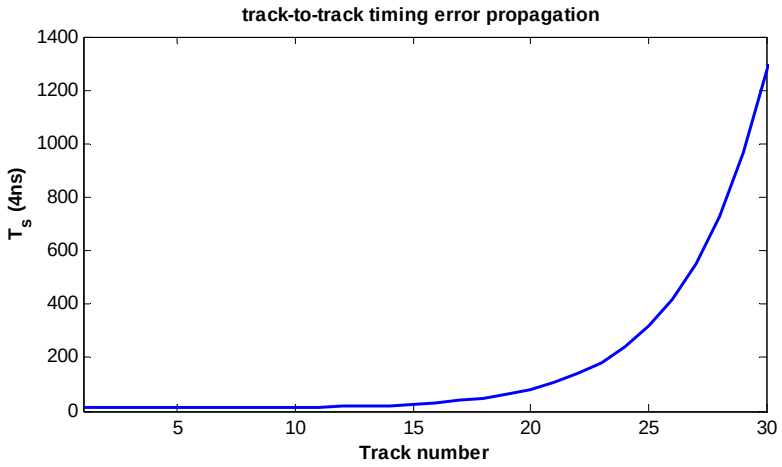

Figure 5.10: Write head timing error propagation without proper control action

5.4 Adaptive Feedforward Control (AFC) Algorithm Design

5.4.1 Modeling of Timing Control Loop

The digital PLL circuit used in concentric SSW timing control loop is modeled as shown in Figure 5.11. The signals and components of interest in the PLL circuit are defined as follows.

- $\phi_i(k)$: the read head phase signal at the sector k in the track i.
- $\phi_{i+1}(k)$: the write head phase signal at the sector k in the track $i+1$.
- $e_i(k)$: the phase error generated by the phase detector (PD).
- k_p : the gain of PD.
- $C(z)$: the loop filter (LF), which is generally designed as a low pass filter.
- $\frac{1}{z-1}$: the transfer function of numerically controlled oscillator (NCO), which is the digital counterpart of the voltage controlled oscillator (VCO).
- $P(z)=\frac{k_p \cdot C(z)}{z-1}$: the PLL open-loop transfer function.

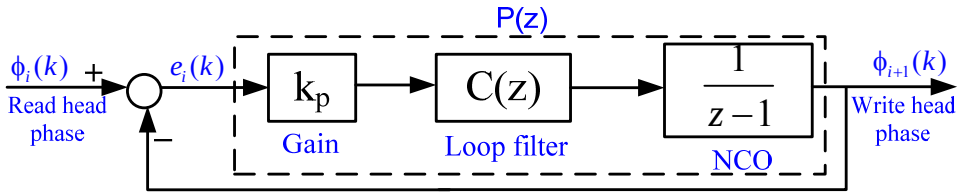

Figure 5.11: Block diagram of linearized digital PLL circuit in concentric SSW timing control loop

Notice that the closed-loop block diagram in Figure 5.11 can be equivalently represented by an open loop block diagram as shown in Figure 5.12, where $S(z)$ is the sensitivity function, i.e.

$$S(z) = \frac{1}{1+P(z)} \tag{5.7}$$

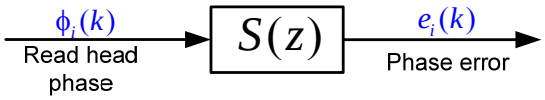

Figure 5.12: Open loop representation of Figure 5.11

Before introducing the proposed control algorithm, the following two definitions are given.

Definition 5.4.1 The timing error between two adjacent tracks with the same sector index is defined as the **radial timing error**. Notice that $e_i(k)$ in Figure 5.11 and Figure 5.12 is the radial timing error at the sector k in the track i, i.e.

$$e_i(k) = \phi_i(k) - \phi_{i+1}(k) \tag{5.8}$$

Definition 5.4.2 The timing error between two adjacent servo sectors in the same track is defined as the **circumferential timing error**. Let $\varepsilon_i(k)$ denote the circumferential timing error at the sector k in the track i, i.e.

$$\varepsilon_i(k) = \phi_i(k-1) - \phi_i(k) \tag{5.9}$$

5.4.2 Proposed AFC Design for Timing Control Loop

In this subsection, a novel adaptive feedforward control (AFC) scheme using the filtered-x least mean square (FXLMS) algorithm is proposed for the timing control loop in concentric SSW process as shown in Figure 5.12.

Design Motivations

From Equation (5.6), we know that the ideal value of read head phase $\phi_i(k)$ should be zero. Therefore, it is desired to estimate $\phi_i(k)$ and subtract the estimate from it, as shown in Figure 5.13.

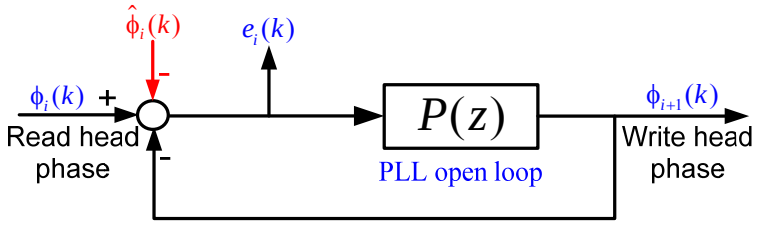

Figure 5.13: Timing control loop with feedforward correction signal $\hat{\phi}_i(k)$

General AFC structure for the system in Figure 5.12

The timing control loop in Figure 5.13 can be drawn as an AFC system if we generate $\hat{\phi}_i(k)$ as shown in Figure 5.14, which is equivalent to Figure 5.2. The AFC aims to generate the input of the dynamic system $S(z)$, i.e. $\hat{\phi}_i(k)$, so that its output $z_i(k)$ follows the desired output $r_i(k)$ best. The scheme consists of two main elements: an adaptive filter $W(z)$ and a LMS algorithm block. The LMS algorithm adaptively adjusts the coefficients of $W(z)$ in order to minimize the performance index. In Figure 5.14, the reference signal $x_i(k)$ is filtered by the dynamic system $S(z)$ before it is utilized by the LMS algorithm for adjusting the filter coefficients. Because of this filtering notion, the adaptive LMS algorithm is called filtered-x LMS algorithm.

Proposed control scheme based on AFC

Notice that the radial timing error $e_i(k)$ in Figure 5.11 and Figure 5.12 is the only measurable signal in the timing control system. In other words, the head phase signal $\phi_i(k)$ is not directly measurable and the circumferential timing error $\varepsilon_i(k)$ is not available for designing the control scheme. However, if $\varepsilon_i(k)$ is utilized as well as the radial timing error $e_i(k)$ to design the adaptive filter, not only the closure error within each individual track, but also the track-to-track timing error propagation will be attenuated. Hence, it is intuitive and necessary to estimate $\varepsilon_i(k)$ in order to design the filter $W(z)$.

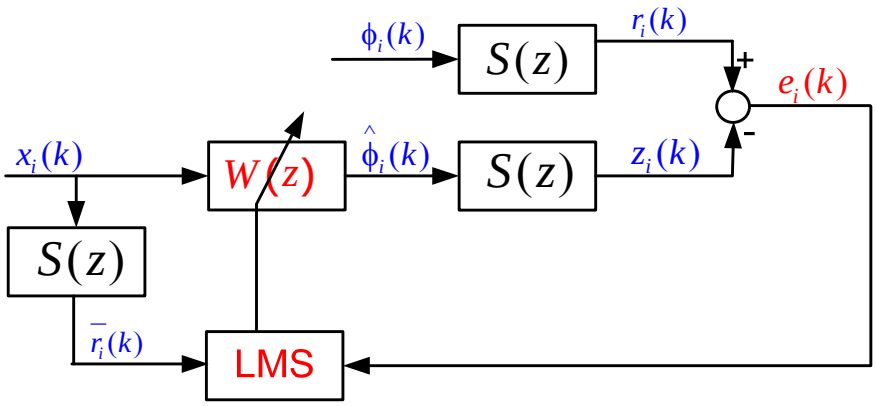

Figure 5.14: Block diagram of general FXLMS based AFC for the system in Figure 5.12

Analysis

(1) In Figure 5.14, $\hat{\phi}_i(k)$ is the estimated head phase signal which can be regarded as the feedforward correction signal in the system, as shown in Figure 5.13.

(2) In Figure 5.13, the closer $\hat{\phi}_i(k)$ to $\phi_i(k)$, the smaller the radial timing error $e_i(k)$ is. In the meanwhile, the closure error within each individual track will also be contained if both the radial timing error $e_i(k)$ and the circumferential timing error $\varepsilon_i(k)$ contribute to designing the adaptive filter $W(z)$. The details are presented in Section 5.4.3.

The proposed adaptive feedforward control scheme using FXLMS algorithm is shown in Figure 5.15, where $W(z)$ is the adaptive filter to be designed for generating the estimated head phase signal $\hat{\phi}_i(k)$. A popular choice of $W(z)$ is the digital finite impulse response (FIR) filter.

Remark 5.4.1

(1) The FIR filter is always stable since all of its poles are located at the origin of the z-plane.

(2) The class of FIR adaptive filter using FXLMS algorithm is well known for its convergence conditions and steady state performance.

Assume that the FIR filter $W(z)$ has the same length as the number of servo sectors in one data track, i.e.

$$W(z) = w_i(0) + w_i(1)z^{-1} + w_i(2)z^{-2} + \cdots + w_i(N-1)z^{-N+1} \tag{5.10}$$

where $w_i(k)$ denotes the filter coefficient at the servo sector k in the track i.

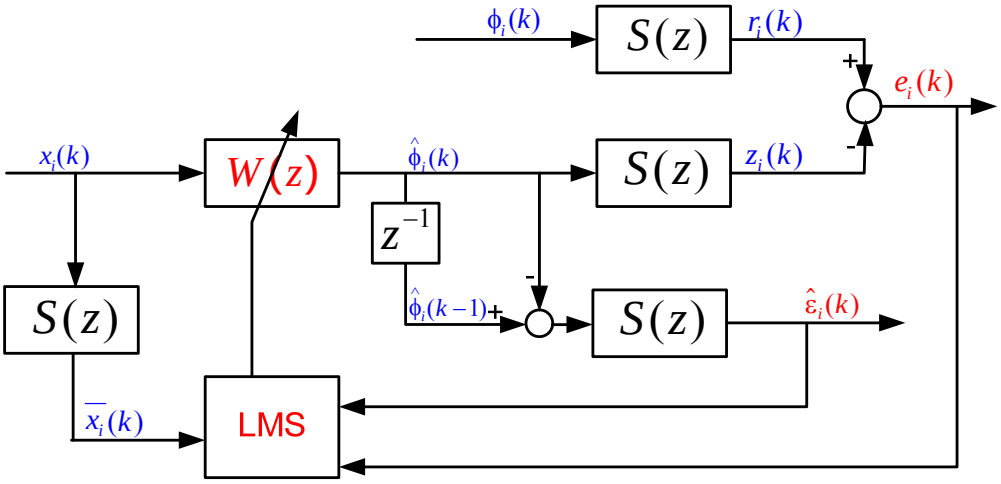

Figure 5.15: The proposed control scheme for concentric SSW timing control loop

In Figure 5.15, the adaptive filter $W(z)$ receives the reference input $x_i(k)$ and generates the estimated head phase signal $\hat{\phi}_i(k)$ and the estimated circumferential timing error signal $\hat{\varepsilon}_i(k)$. To simplify the design, the reference signal $x_i(k)$ is chosen as an impulse train, i.e.

$$x_i(k) = \begin{cases} 1 & k = 0;\ i = 1,\ 2,\ \cdots \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

Notice that the sensitivity function $S(z)$ in Equation (5.7) can be written as

$$S(z) = s(0) + s(1)z^{-1} + s(2)z^{-2} + \cdots + s(N-1)z^{-N+1} + \cdots \tag{5.12}$$

where $s(k)$ $(k = 0,\ 1,\ \cdots)$ is Markov coefficients of $S(z)$.

Introducing the estimated head phase signal $\hat{\phi}_i(k)$ in Figure 5.15, the radial timing error $e_i(k)$ in Equation (5.8) is rewritten as

$$\begin{aligned} e_i(k) &= r_i(k) - z_i(k) \\ &= r_i(k) - \sum_{j=0}^{\infty} s(j) \cdot \hat{\phi}_i(k-j) \\ &= r_i(k) - \sum_{j=0}^{\infty} s(j) \left[\sum_{n=0}^{N-1} w_i(n) \cdot x_i(k-j-n) \right] \end{aligned} \tag{5.13}$$

The estimated circumferential timing error $\hat{\varepsilon}_i(k)$ is represented as

$$\begin{aligned}\hat{\varepsilon}_i(k) &= S(z)\left(\hat{\phi}_i(k-1) - \hat{\phi}_i(k)\right) \\ &= \sum_{j=0}^{\infty} s(j)\left(\hat{\phi}_i(k-1-j) - \hat{\phi}_i(k-j)\right) \\ &= \sum_{j=0}^{\infty} s(j)\left[\sum_{n=0}^{N-1} w_i(n)\cdot x_i(k-1-n-j) - \sum_{n=0}^{N-1} w_i(n)\cdot x_i(k-n-j)\right]\end{aligned} \tag{5.14}$$

5.4.3 Adaptive Filter Design

This subsection presents the procedure of designing the adaptive filter $W(z)$ by using the estimated head phase signal $\hat{\phi}_i(k)$ and estimated circumferential timing error $\hat{\varepsilon}_i(k)$. First, the following two definitions are introduced.

Definition 5.4.3 The **circumferential timing error energy** of the servo sector k is defined as

$$\begin{aligned}\beta_i(k) &= \hat{\varepsilon}_i^2(k) \\ &= \left(\sum_{j=0}^{\infty} s(j)\left[\sum_{n=0}^{N-1} w_i(n)\cdot x_i(k-1-n-j) - \sum_{n=0}^{N-1} w_i(n)\cdot x_i(k-n-j)\right]\right)^2\end{aligned} \tag{5.15}$$

Definition 5.4.4 The **radial timing error energy** along the track i is defined as

$$\begin{aligned}\alpha_i &= \sum_{k=0}^{N-1} e_i^2(k) \\ &= \sum_{k=0}^{N-1}\left(r_i(k) - \sum_{j=0}^{\infty} s(j)\left[\sum_{n=0}^{N-1} w_i(n)\cdot x_i(k-j-n)\right]\right)^2\end{aligned} \tag{5.16}$$

The proposed FXLMS algorithm aims to minimize both $\beta_i(k) = \hat{\varepsilon}_i^2(k)$ and $\alpha_i = \sum_{k=0}^{N-1} e_i^2(k)$ across the entire disk surface by adjusting the filter coefficient $w_i(k)$ adaptively. Hence, the design problem is equivalent to the following two minimization problems.

(1) Minimization of the circumferential timing error energy at the servo sector k, i.e.

$$\min_{w_i(k)}\left(\beta_i(k) = \hat{\varepsilon}_i^2(k)\right)$$

It is considered that the filter coefficient $w_i(k)$ is updated from sector to sector in order to minimize the squared error $\beta_i(k) = \hat{\varepsilon}_i^2(k)$. By doing the partial differentiation on both sides of Equation (5.15) with respect to the filter coefficient $w_i(k)$ at the servo sector k, we have

$$\begin{aligned}\frac{\partial \beta_i(k)}{\partial w_i(k)} &= 2\hat{\varepsilon}_i(k)\cdot\left(\sum_{j=0}^{\infty} s(j)\left[\sum_{n=0}^{N-1}\delta_n^k(n)\cdot x_i(k-1-n-j)-\sum_{n=0}^{N-1}\delta_n^k\cdot x_i(k-n-j)\right]\right)\\ &= 2\hat{\varepsilon}_i(k)\left(\sum_{j=0}^{\infty} s(j)\left[x_i(0-1-j)-x_i(0-j)\right]\right)\\ &= 2\hat{\varepsilon}_i(k)\left(-\sum_{j=0}^{\infty} s(j)x_i(0-j)\right)\\ &= -2\hat{\varepsilon}_i(k)\cdot\bar{x}_i(0)\end{aligned} \tag{5.17}$$

Applying the LMS algorithm yields the update equation for the filter coefficient at the servo sector k in the track i:

$$w_i^{l+1}(k) = w_i^l(k) + \mu_1\hat{\varepsilon}_i^l(k)\bar{x}_i(0) \tag{5.18}$$

where μ_1 is the step size parameter; $l = 1, 2, \cdots, L$ denotes the update iteration for the filter coefficient $w_i(k)$ and L is the constant update iteration number to be decided by the designer. $\bar{x}_i(0)$ is the initial value of $\bar{x}_i(k)$, which is the filtered $x_i(k)$. From Equation (5.11) and Equation (5.12), we know that

$$\bar{x}_i(k) = s(k) \quad (k = 0, 1, \cdots, N-1)$$

The update equation in (5.18) implies that the filter coefficient $w_i(k)$ is updated from $k = 0$ till $k = N-1$ in the track i for L times before moving to the next track $i+1$.

(2) Minimization of the radial timing error energy along the track i, i.e.

$$\min_{w_i}\left(\alpha_i = \sum_{k=0}^{N-1} e_i^2(k)\right)$$

In this minimization problem, the filter coefficient vector in the track i, i.e. $w_i = \left[w_i(0)\ \ w_i(1)\ \ \cdots\ \ w_i(N-1)\right]^T$ is updated (from track i to track $i+1$) to minimize the sum of squared error $\alpha_i = \sum_{k=0}^{N-1} e_i^2(k)$. By doing the partial differentiation on both sides of Equation (5.16) with respect to $w_i(n)$ $(n = 0,1,\cdots,N-1)$, we have

$$\begin{aligned}\frac{\partial \alpha_i}{\partial w_i(n)} &= \sum_{k=0}^{N-1} 2e_i(k)\left\{-\sum_{j=0}^{\infty} s(j)x_i(k-j-n)\right\}\\ &= -\sum_{k=0}^{N-1} 2e_i(k)\left(\sum_{j=0}^{\infty} s(j)x_i(k-j-n)\right)\\ &= -\sum_{k=0}^{N-1} 2e_i(k)\bar{x}_i(k-n)\end{aligned} \tag{5.19}$$

Applying the LMS algorithm yields the update equation for the filter coefficient vector $w_i = [w_i(0)\ \ w_i(1)\ \ \cdots\ \ w_i(N-1)]^T\ \ (i = 1, 2, \cdots)$

$$w_{i+1}(k) \ = \ w_i(k) + \mu_2 \left(\sum_{n=0}^{N-1} e_i(n) \overline{x_i}(n - k) \right) \tag{5.20}$$

where μ_2 is the step size parameter.

Remark 5.4.2

(1) The filter coefficient vector w_{i+1} in the track $i+1$ is obtained by using the update algorithm in Equation (5.20). Then the filter coefficient at the servo sector k, i.e. $w_{i+1}(k)$, is updated from $k = 0$ till $k = N-1$ for L times by using the update algorithm in Equation (5.18) before moving to the track $i+2$. In the *Lth* update iteration, the write head starts to write the timing marks on the track $i+1$.

(2) Minimizing the circumferential timing error energy at the servo sector k (i.e. $\beta_i(k) = \hat{\varepsilon}_i^2(k)$) ensures that the closure error within the track i is contained; and minimizing the radial timing error energy along track i (i.e. $\alpha_i = \sum_{k=0}^{N-1} e_i^2(k)$) ensures that the timing error propagation from track i to track $i+1$ is attenuated.

(3) From Equation (5.18) and Equation (5.20), we have the filter coefficient at every servo sector, which is optimally and adaptively updated to ensure minimum closure error within each individual track and minimum timing error propagation from track to track.

Combining Equation (5.18) and Equation (5.20), the proposed FXLMS-based AFC algorithm for the filter coefficients on the entire disk surface can be described as

$$\begin{cases} \overline{w}_i^{l+1}(k) \ = \ \overline{w}_i^{l}(k) + \mu_1 \hat{\varepsilon}_i^{l}(k) \overline{x}_i(0) \\ \text{Initial condition:} \quad \overline{w}_i^{1}(k) \ = \ w_i(k) \end{cases} \tag{5.21a}$$

$$w_{i+1}(k) \ = \ \overline{w}_i^{L}(k) + \mu_2 \left(\sum_{n=0}^{N-1} e_i^{L}(n) \overline{x}_i(n - k) \right) \tag{5.21b}$$

where $l = 1,\ 2,\ \cdots,\ L;\ \ k = 0,\ 1,\ \cdots, N-1;$ and $i = 1,\ 2,\ \cdots$.

5.5 Simulation Study

In the simulation study, we consider the mechanical jitter $d_i(k)$ and noise jitter $n_i(k)$, as shown in Figure 5.16. In HDD manufacturing process, the mechanical jitter is caused by the disk flutter and the spindle speed fluctuation due to mechanical vibrations such as ball-bearing vibration. The noise jitter is mainly caused by the head noise and electronics noise in the read-back process.

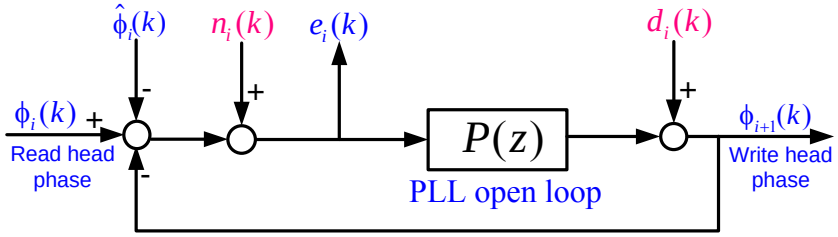

Figure 5.16: Concentric SSW timing control loop with jitters

Figure 5.17 and Figure 5.18 show the mechanical jitter and noise jitter injected into the system for the simulation study, which are collected from the real disk drive.

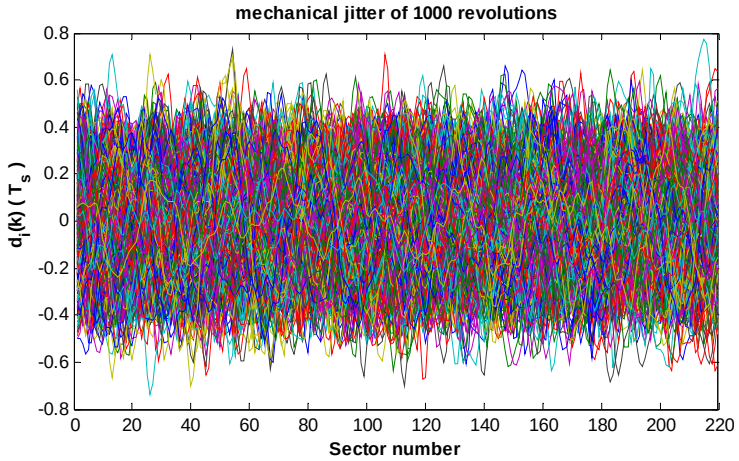

Figure 5.17: Mechanical jitter in concentric SSW process

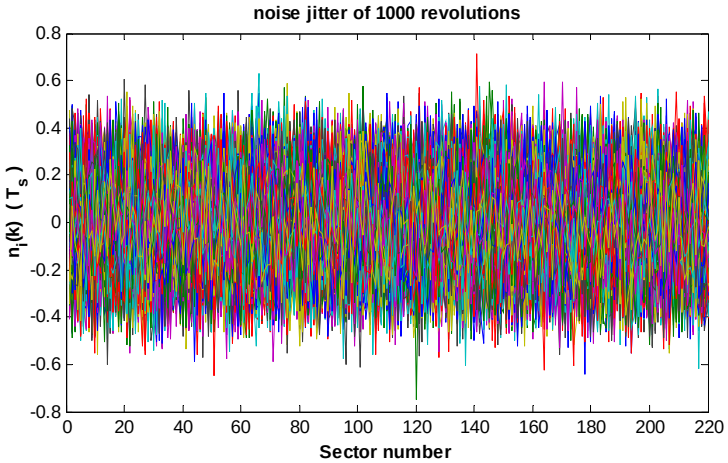

Figure 5.18: Noise jitter in concentric SSW process

Remark 5.5.1

In simulations, the timing signals and timing jitters are normalized to $T_s = 4ns$.

Figure 5.19 shows the frequency response of the PLL open loop system used in the simulation study.

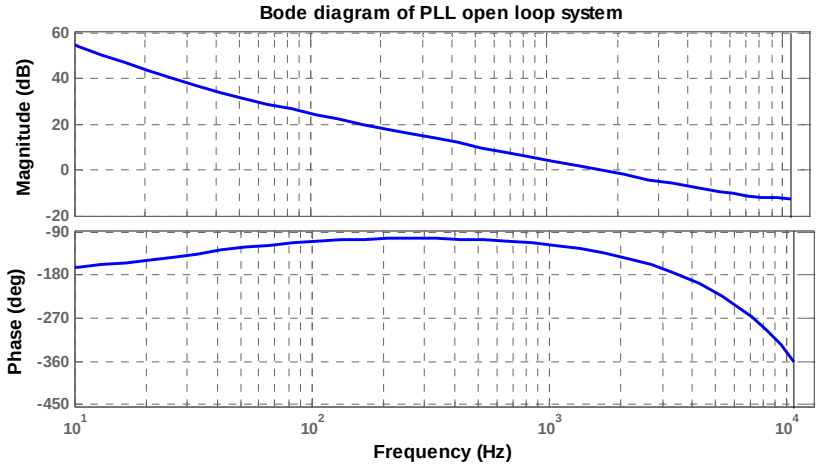

Figure 5.19: Frequency response of PLL open loop transfer function

An example of the adaptive filter $W(z)$ is shown in Figure 5.20, which is the frequency response of $W(z)$ in the 800th track.

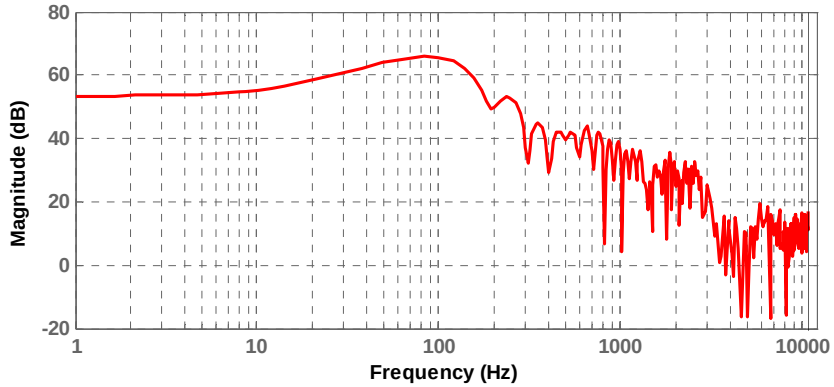

Figure 5.20: Frequency response of adaptive filter $W(z)$ in the 800th track

Figure 5.21 shows the seed clock track profile, i.e. $\phi_0(k)$, which has a sigma value of $12.6\,T_s$.

In order to show the feasibility of the proposed control algorithm, the iterative learning control (ILC) scheme presented in Chapter 3 was also designed for this control system and its results are used to compare with that of the proposed AFC method. Figure

5.22 shows the comparison of the write head phase profile for 1000 tracks. The dotted line represents the profile in the system using the ILC scheme, and the solid line represents the profile in the system using the proposed AFC scheme. From this figure, we can see that in both control schemes, the write head timing distortion converges. However, the system with the proposed AFC scheme has a better written timing pattern with the average $\sigma[\phi_i] = 0.79\,T_s$, which is much smaller than the system with the ILC scheme ($\sigma[\phi_i] = 3.42\,T_s$).

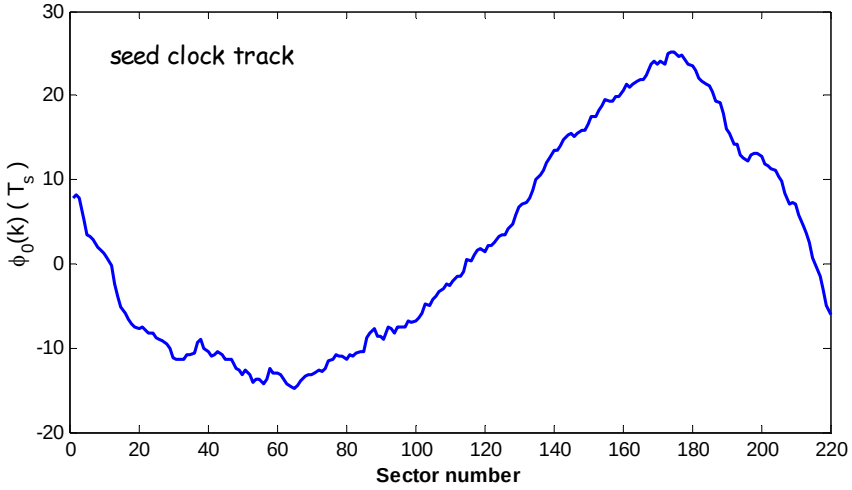

Figure 5.21: Seed clock track profile $\phi_0(k)$

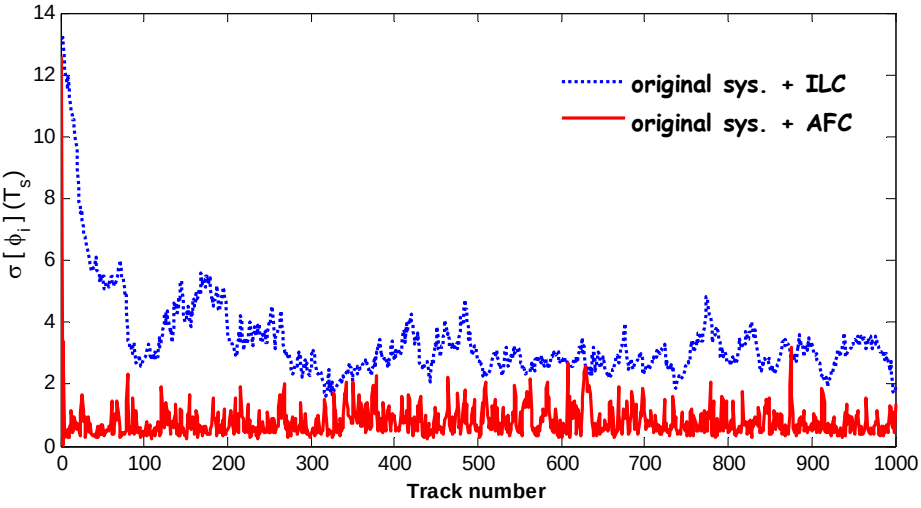

Figure 5.22: Comparison of write head phase profile

The comparison of the radial timing error profile e_i for 1000 tracks is shown in Figure 5.23. It can be seen that the system using the proposed AFC scheme deals with the radial timing error better (with the resulting average $\sigma[e_i] = 0.57\,T_s$ as shown in the solid line) than the system using the ILC scheme, which has the average $\sigma[e_i] = 2.24\,T_s$ as shown in the dotted line.

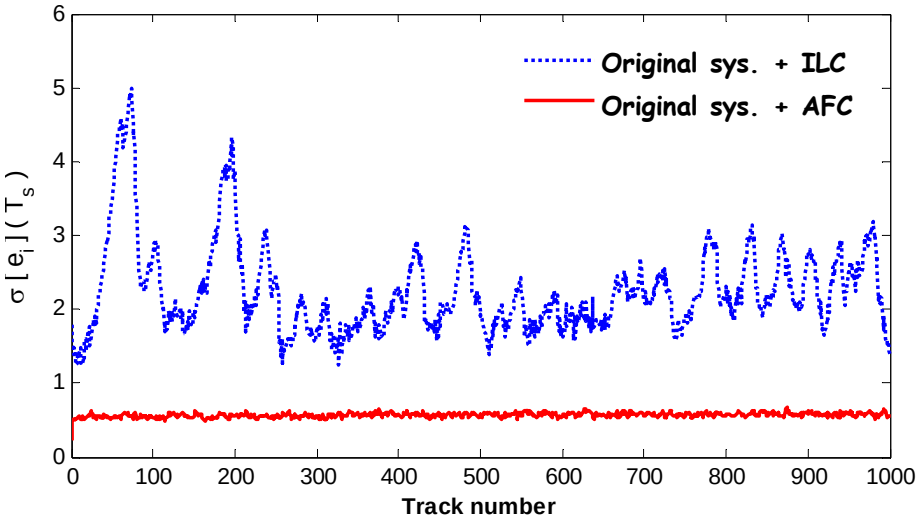

Figure 5.23: Comparison of radial timing error profile

Figure 5.24 shows the circumferential timing error profile in the systems using the ILC scheme and the proposed AFC scheme. We learn that the resulting average $\sigma[\varepsilon_i]$ is $0.28\,T_s$ in the system with the proposed AFC scheme (as shown in the solid line), which is within the accepted limit. While in the system with the ILC scheme as shown in the dotted line, the resulting average $\sigma[\varepsilon_i]$ is $0.93\,T_s$ and the maximum $\sigma[\varepsilon_i]$ is $2.13\,T_s$, which may cause the servowriting process to fail.

Figure 5.23 and Figure 5.24 show that both the circumferential timing error and the radial timing error are smaller and smoother in the system using the proposed control scheme than that in the system using the ILC scheme. This ensures better alignment of the written servo patterns on the disk surface and improves the quality of self-servowriting process.

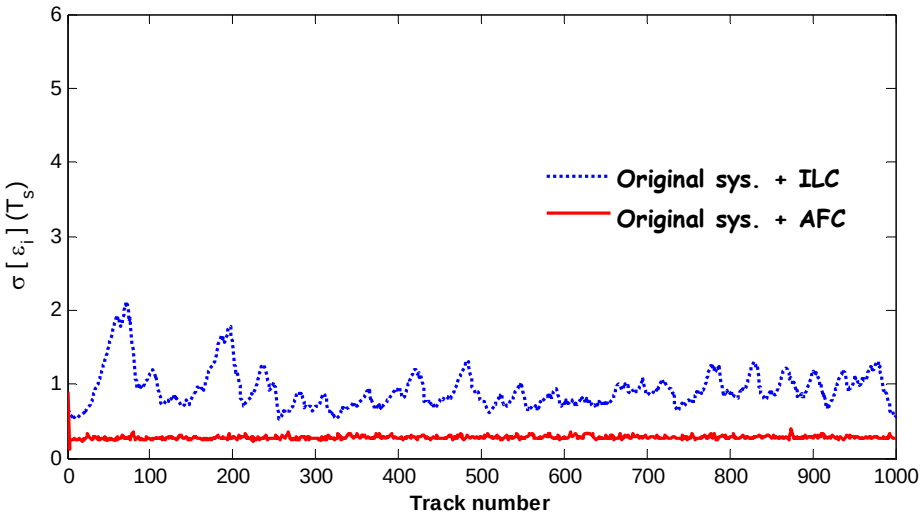

Figure 5.24: Comparison of circumferential timing error profile

5.6 Chapter Summary

In this chapter, a novel adaptive feedforward control algorithm for discrete linear repetitive process was presented, which aimed at containing the error propagation along the time direction as well as the iteration direction.

The timing control loop in HDD concentric SSW process is a good example of such repetitive process, where the error propagation along the time direction is as significant as the error propagation along the iteration direction. An adaptive feedforward control based algorithm was proposed for this repetitive process. The adaptive filter was designed to estimate the circumferential timing error which was utilized together with the radial timing error to contain the closure error within each individual track and attenuate the timing error propagation from track to track. The filter coefficients were derived by applying the filtered-x least mean square algorithm, which minimizes both the circumferential timing error energy and the radial timing error energy. Therefore, the filter coefficient at every servo sector is optimal for minimum closure error and minimum timing error propagation. To verify the effectiveness of the proposed control algorithm, simulation was conducted and the results are compared to the iterative learning control design from Chapter 3. The comparison showed that the written servo patterns in the system using the proposed adaptive feedforward control scheme had higher quality with smaller and smoother closure error and timing error propagation.

Chapter 6

Recursive Least Square based Parameter Adaptation Algorithm Design

6.1 Introduction

It is often the case that repeatable error is present in discrete linear repetitive processes, such as the timing control loop in HDD spiral based SSW process. In this repetitive process, the system dynamics is dominated by the repeatable timing error (RTE) which is read from the prewritten spiral tracks and used as the reference. The RTE consequently results in non-uniform distribution of the written product servo sectors (the details are presented in Section 6.2.2). Hence, in order to write the product servo patterns uniformly and accurately along the circular tracks, it is desired to design an effective control scheme for compensating the RTE.

A lot of research work has been done on the repeatable runout (RRO) compensation for the position control loop in SSW process. In [83, 64, 84, 70], various adaptive filters were designed to compensate the RRO and improve the track-following performance in HDD servo system. Their control objective is to minimize the off-track (i.e. the deviation of the read/write heads from the center of circular tracks) and the filter coefficients are updated from track to track. These methods are not suitable for the timing control loop in spiral based SSW process because they can not guarantee that the servo sectors are uniformly and accurately written along the circular tracks. The uniform and accurate distribution of servo sectors requires that the timing error is adaptively corrected from sector to sector.

Little research effort has been devoted to the study of RTE compensation for spiral based SSW process in HDD manufacturing. In [58], a feedforward compensation method was proposed. This method measures and records the RTE at all the servo sectors in one track; then the mean value is calculated and used for generating the feedforward compensation signal. Thus, it spends much time to compute the feedforwad compensation value before writing the product servo patterns on the track. In Chapter 5 or [21], a novel adaptive feedforward control (AFC) scheme was designed to contain the

closure error within each individual track and attenuate the track-to-track timing error propagation in HDD concentric SSW process. However, this approach is not suitable for spiral based SSW process since it doesn't deal with the repeatable timing error (in the reference signal) appropriately.

In spiral based SSW process, except for the RTE, there exists some non-repeatable timing error (NRTE) induced by the spindle speed variation and sensor noise. NRTE causes the phase incoherency at the servo sectors between adjacent tracks (the details are presented in Section 6.2.2). Hence, in order to generate an accurate servowriting clock signal in spiral based SSW timing control loop, both RTE and NRTE need to be properly handled. In this chapter, a recursive least square (RLS) based parameter adaptation algorithm (PAA) is proposed to estimate and cancel the RTE from the reference signal in the read head. A comb filter is further designed as the error shaping filter to improve the PAA estimation performance and reduce the contamination of non-repeatable timing distortion in SSW process.

The remainder of this chapter is organized as follows. In Section 6.2, the timing errors in spiral based SSW process are introduced and the control problem is formulated. Section 6.3 presents two existing control methods used for rejecting the periodic disturbances. In Section 6.4, a RLS based PAA is proposed to estimate and cancel the RTE. For further enhancement of the RTE cancellation and NRTE reduction, an error shaping filter is designed. Section 6.5 provides the simulation study using the industry supplied data and finally the chapter summary is given in Section 6.6.

6.2 Control Problem Formulation

6.2.1 Challenges in Spiral based SSW Process

In spiral based SSW process, the concentric product servo patterns are written by referring to the prewritten spiral tracks (see Section 2.3.3 for details). Hence, the quality of the product servo patterns is closely affected by that of the spiral tracks. In other words, any imperfection of spiral tracks damages the quality of SSW. There are two main obstacles impacting the quality of SSW in the current HDD manufacturing.

Drifts of spiral locations

The first and most significant obstacle in spiral based SSW process is known as the drifts of spiral locations; that means the spiral tracks are shifted away from their ideal locations due to disk eccentricity and/or thermal expansion (as shown in Figure 6.1). The disk eccentricity is common in spiral based SSW process since the spiral tracks are written by an external spiral writer, while the SSW is performed using the internal heads. It occurs if the center of seeding spirals is different from the center of servowriting, as shown in Figure 6.2.

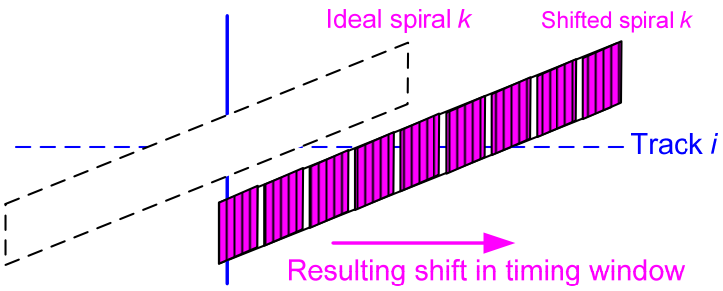

Figure 6.1: Drifts of spiral locations

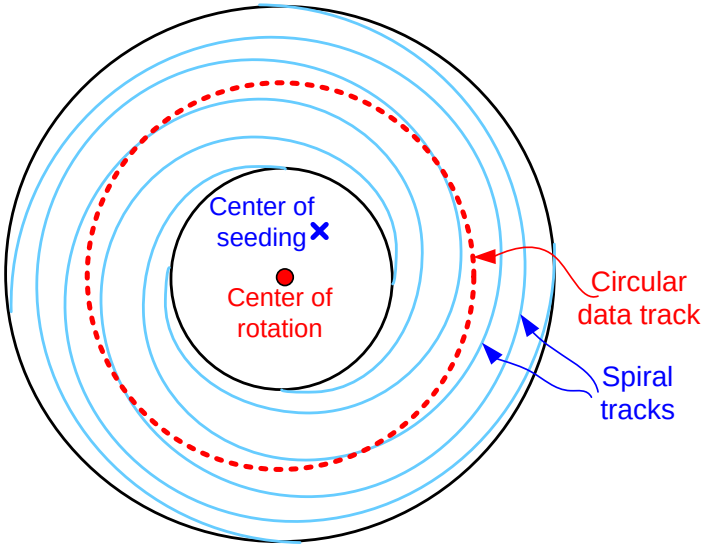

Figure 6.2: Illustration of disk eccentricity

Position error in sync marks

The second obstacle in spiral based SSW process is the position error of sync marks, which means the circumferential position of sync marks within a spiral track, relative to the ideal circumferential position, is shifted as shown in Figure 6.3. The sync mark position error is mainly caused by the mechanical jitter and/or thermal expansion.

During the manufacturing process, when the spiral tracks are written on the disk by the external spiral writer, the spiral shifts due to the thermal expansion and sync marks' position error are also written together with the spiral tracks and introduce the written-in timing error, which are periodic and synchronized to the disk rotation.

Remark 6.2.1

Each spiral track, as shown in Figure 6.3, comprises a high frequency signal which is interrupted at a predetermined constant interval by the sync marks.

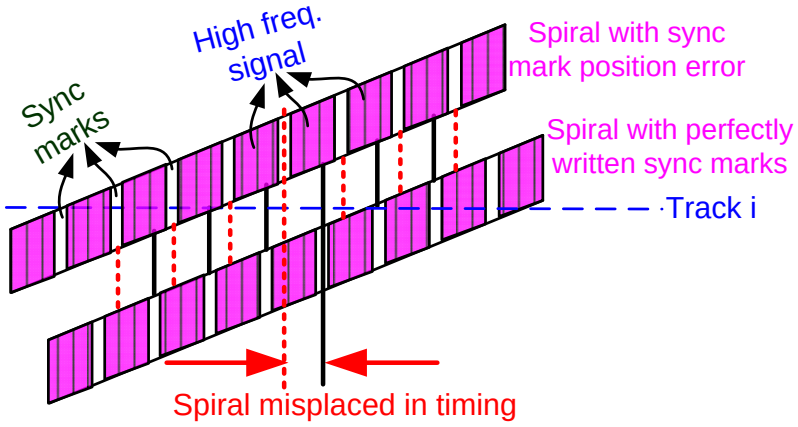

Figure 6.3: Position error in sync marks

6.2.2 Timing Errors in Spiral based SSW Process

Repeatable timing error

With the continuously increasing data density, HDD manufacturers write data to the disks such that the data density is substantially uniform throughout the disk. Uniform density (as shown in Figure 6.4(a)) requires an accurate clock during writing the spiral tracks. After the spiral tracks are written, the disk drive is operated to self-write the product servo patterns, wherein the drifts of spiral tracks and sync marks' position error appear as repeatable timing error (RTE), which is read by the read head and used as the reference signal. The RTE shifts the demodulation windows in the demodulation circuit, as shown in Figure 6.5. The shifts of demodulation windows directly and adversely affect the performance of both position control loop and timing control loop in SSW servo system. Consequently, the product servo patterns are also shifted from sector to sector, resulting in non-uniform alignment, as shown in Figure 6.4(b). However, we desire to write the product servo patterns uniformly along the circular tracks.

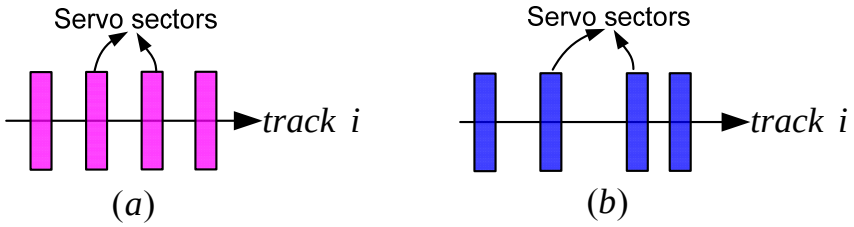

Figure 6.4(a): Servo sectors with ideally even spacing (b): Servo sectors with uneven spacing due to RTE

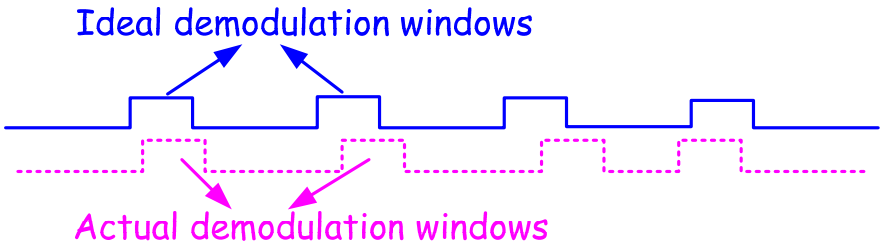

Figure 6.5: Shifts of timing windows in demodulation circuit

Non-repeatable timing error

The non-repeatable timing error (NRTE) in spiral based SSW process is mainly induced by the spindle speed variation and sensor noise, and it causes the phase incoherency at the servo sectors between adjacent tracks, as shown in Figure 6.6(b).

Therefore, in order to improve the quality of product servo patterns, it is necessary to compensate the RTE and reduce NRTE when the product servo patterns are written by referring to the prewritten spiral tracks.

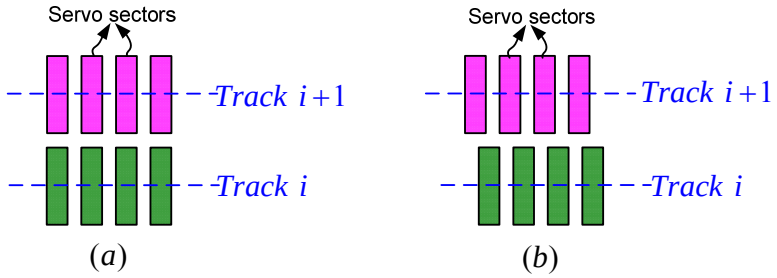

Figure 6.6(a): Coherent phase at the sectors between adjacent tracks (b): Incoherent phase at the sectors between adjacent tracks due to NRTE

6.2.3 Control Objectives

In spiral based SSW process, the quality of the written servo patterns must be degraded due to the timing distortion in the spiral tracks since the product servo patterns are written based on the prewritten spirals. Therefore, by referring to the block diagram of spiral based SSW timing control loop in Figure 2.7, the first and foremost significant control objective is to reduce the repeatable timing error $\phi_{R\_i}(k)$ which is the major portion of timing distortion in spiral based SSW process. In order to effectively compensate the RTE, it is necessary to estimate it first because it is not directly measurable during SSW process. The details are presented in Section 6.4.1.

The other equally important goal is to ensure a good quality of the product servo patterns even in the presence of non-repeatable timing error (NRTE) $\phi_{NR\_i}(k)$ (as shown in Figure 6.10). Namely, the timing signal from the write head is desired to be evenly

distributed along the circular tracks, i.e. the written timing deviation is as minimum as possible. By using the write head phase signal $\hat{\phi}_i(k)$ in Figure 2.7, the performance index can be written as

$$\sum_{i=1}^{m}\sigma\left[\hat{\phi}_i\right] = \sum_{i=1}^{m}\sqrt{\frac{1}{N}\sum_{k=0}^{N-1}\hat{\phi}_i^{\,2}(k)} \qquad (m \geq 1)$$

Ideally, $\hat{\phi}_i(k)=0$ for all k. In Section 6.4.2, it is shown that in order to achieve this target, the NRTE should be appropriately dealt with as well as the RTE.

6.3 Rejection of Periodic Disturbance

This section reviews two existing control methods used for rejecting the periodic disturbances.

Consider the setup in Figure 6.7, which shows a linear time invariant (LTI) plant $G(s)$ perturbed by a periodic disturbance $d(t)$ represented as follows

$$\begin{aligned} d(t) &= D\sin(\omega_0 t+\phi) \\ &= a\sin(\omega_0 t)+b\cos(\omega_0 t) \end{aligned} \qquad (6.1)$$

where $D=\sqrt{a^2+b^2}$, $\phi=\tan^{-1}(b/a)$, a and b are the Fourier coefficients of $d(t)$ at the frequency ω_0. The feedback controller $C(s)$ is assumed to be designed to stabilize the system.

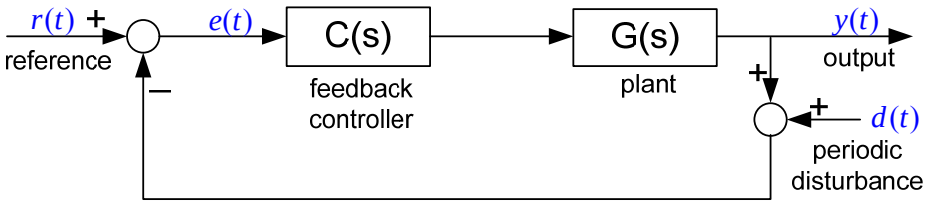

Figure 6.7: Block diagram of feedback control system with periodic disturbance

Several methods have been used to design the control scheme for eliminating the adverse effects of periodic disturbances. The most common approach is based on the use of a notch filter [12, 49] as shown in Figure 6.8. A second order notch filter $N(s)$ is given as an example:

$$N(s) = \frac{s^2+\omega_0^{\,2}}{s^2+(\omega_0/Q)s+\omega_0^{\,2}} \qquad (6.2)$$

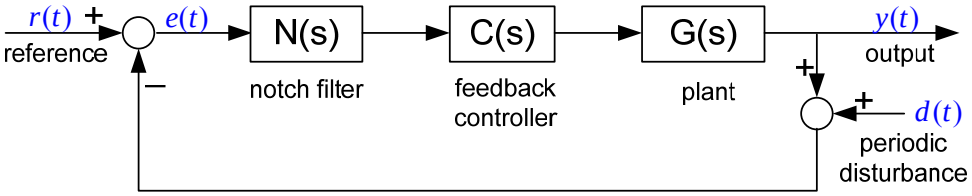

Figure 6.8: Block diagram of feedback control system using notch filter to reject periodic disturbance

Notice that the notch filter in Equation (6.2) has an infinite notch depth corresponding to the disturbance frequency ω_0 and the steepness of the notch is determined by the parameter Q (i.e. the larger Q makes the notch steeper). However, removing all signals at the synchronous frequency (i.e. the rotational speed) may result in closed-loop instability, limiting the usefulness of this approach.

Another method is based on the adaptive feedforward control (AFC) scheme [72, 71, 41] as shown in Figure 6.9. The periodic disturbance is cancelled by adding the negative of its estimated value, i.e. $v(k)$, which is written as

$$v(k) = \hat{a}(k)\sin(\omega_0 k) + \hat{b}(k)\cos(\omega_0 k)$$

where $\hat{a}(k)$ and $\hat{b}(k)$ are the parameters to be estimated by using the adaptive algorithm.

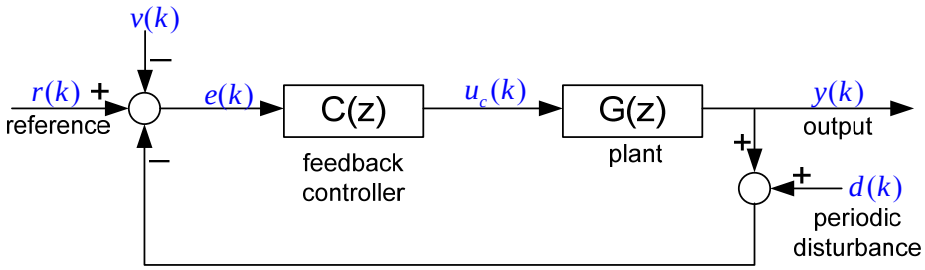

Figure 6.9: Block diagram of AFC scheme for rejecting periodic disturbance

The conventional adaptive law governing $\hat{a}(k)$ and $\hat{b}(k)$ is given by the least mean square (LMS) algorithm and written as follows [81]:

$$\hat{a}(k+1) = \hat{a}(k) + 2\eta u_c(k)\sin(\omega_0 k)$$
$$\hat{b}(k+1) = \hat{b}(k) + 2\eta u_c(k)\cos(\omega_0 k)$$

where η is the step size parameter that regulates the speed and stability of the adaptation and $u_c(k)$ is the control input for the plant.

A linear transfer function of the AFC scheme in Figure 6.9 can be obtained by analyzing the signal propagation from the system error $e(k)$ to the feedback error $r(k)-y(k)$ to the AFC output $v(k)$, by ignoring the time-varying component in the z-transform of the AFC output $v(k)$. It is written as follows [81, 29]:

$$\frac{E(z)}{R(z)-Y(z)} = \frac{z^2-2z\cos(\omega_0)+1}{z^2-2(1-\eta)z\cos(\omega_0)+1-2\eta} \tag{6.3}$$

where $E(z)=R(z)-Y(z)-V(z)$. Equation (6.3) is the transfer function of a second order digital notch filter with a notch at the disturbance frequency ω_0. The sharpness of the notch is determined by the parameter η. Therefore, a strong similarity between the notch filter and the conventional AFC exists and it may lead to instability for both of the above cases.

The following section proposes a recursive least square based parameter adaptation algorithm to compensate the repeatable timing error in spiral based SSW timing control loop. This approach is able to easily guarantee the stability of the system.

6.4 Timing Error Compensator Design

The proposed control scheme is shown in Figure 6.10, where $\phi_{NR\_i}(k)$ represents the NRTE. The design includes two steps. First, a recursive least square (RLS) based parameter adaptation algorithm (PAA) is proposed to estimate the RTE and the output $v_i(k)$ is then used to cancel it. Secondly, an error shaping filter $W(z)$ is added to enhance the performance of PAA and reduce the effect of NRTE in the system.

6.4.1 Recursive Least Square based Parameter Adaption Algorithm

PAA refers to a problem of identifying or estimating unknown parameters of a system, the structure of which is assumed known and identical to that of the system. And we can use a mathematical model and adjust the model parameters so that the input-output behavior of the model and that of the system become close to each other.

The RTE $\phi_{R\_i}(k)$ in Figure 2.6 and Figure 6.10 can be written as a sum of n sinusoids of known frequencies since it is periodic and synchronized to the disk rotation:

$$\begin{aligned} \phi_{R\_i}(k) &= \sum_{j=1}^{n}\left[a_{ij}\sin(2\pi kj/N)+b_{ij}\cos(2\pi kj/N)\right] \\ &= \psi^T(k)\,\theta_i \qquad (k=0,\,1,\,\cdots,\,N-1 \text{ and } n<N/2) \end{aligned} \tag{6.4}$$

where

$$\psi^T(k)=\left[\sin(2\pi k/N),\ \sin(4\pi k/N),\cdots,\ \sin(2\pi kn/N),\ \cos(2\pi k/N),\cos(4\pi k/N),\cdots,\ \cos(2\pi kn/N)\right],$$

$\theta_i^T = [a_{i1}, a_{i2}, \cdots, a_{in}, b_{i1}, b_{i2}, \cdots, b_{in}]$, and the Fourier coefficients a_{ij} and b_{ij} are not dependent on k, i.e. they are constants.

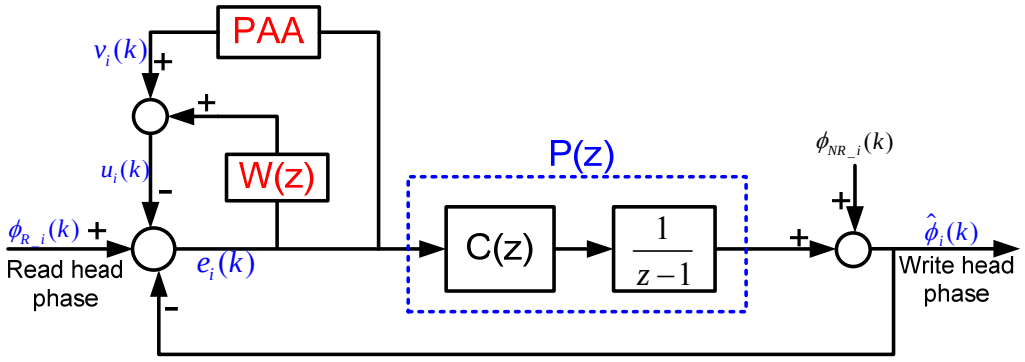

Figure 6.10: Proposed control scheme with PAA and error shaping filter

In Figure 6.10, the PAA output $v_i(k)$ is the estimated RTE and written as

$$\begin{aligned} v_i(k) &= \sum_{j=1}^{n}\left[\hat{a}_{ij}(k-1)\sin(2\pi kj/N)+\hat{b}_{ij}(k-1)\cos(2\pi kj/N)\right] \\ &= \psi^T(k)\,\hat{\theta}_i(k-1) \end{aligned} \tag{6.5}$$

where $\hat{\theta}_i^T(k) = \left[\hat{a}_{i1}(k),\ \hat{a}_{i2}(k), \cdots,\ \hat{a}_{in}(k),\ \hat{b}_{i1}(k),\ \hat{b}_{i2}(k), \cdots,\ \hat{b}_{in}(k)\right]$.

Remark 6.4.1

- $\psi(k)$ in equations (6.4) and (6.5) is known for all of k.
- θ_i in Equation (6.4) is the parameter vector to be estimated; and $\hat{\theta}_i(k)$ in Equation (6.5) is the estimated parameter vector.
- If $\theta_i = \hat{\theta}_i(k-1)$, then $v_i(k)$ will completely cancel $\phi_{R\_i}(k)$ from the reference.

Now we assume that the following $(k+1)$ sets of data are known

$$\begin{cases} \phi_{R\_i}(0),\ \phi_{R\_i}(1),\ \cdots,\ \phi_{R\_i}(k) \\ \psi(0),\ \psi(1),\ \cdots,\ \psi(k) \end{cases}$$

The design problem is to find the parameter vector $\hat{\theta}_i(k)$ to minimize the following cost function

$$G\left(\hat{\theta}_i(k)\right) = \frac{1}{2}\sum_{j=0}^{k}\left[\phi_{R\_i}(j) - \psi^T(j)\,\hat{\theta}_i(k)\right]^2 \tag{6.6}$$

Then $\hat{\theta}_i(k)$ is obtained by solving $\frac{\mathrm{dG}\left(\hat{\theta}_i(k)\right)}{d\hat{\theta}_i(k)}=0$ and the RLS based PAA can be described by the following equations

$$\begin{aligned}\hat{\theta}_i(k) &= \hat{\theta}_i(k-1)+F(k)\psi(k)e_i^0(k)\\ e_i^0(k) &= \phi_{R\_i}(k)-\psi^T(k)\,\hat{\theta}_i(k-1)\\ F_i(k) &= F_i(k-1)-\frac{F_i(k-1)\psi(k)\psi^T(k)\,F_i(k-1)}{1+\psi^T(k)\,F_i(k-1)\psi(k)}\end{aligned} \tag{6.7}$$

where $F_i(k)$ is the RLS gain for the sector k in the track i.

We note that in spiral based SSW timing control loop, only the system error $e_i(k)$ is measurable, i.e. $\phi_{R\_i}(k)$ is not directly measurable and the a-priori prediction error $e_i^0(k)$ cannot be used in the PAA. However, in Figure 6.10, before adding the error shaping filter $W(z)$ and assuming the system is not contaminated by NRTE $\phi_{NR\_i}(k)$, the system error $e_i(k)$ is represented as

$$\begin{aligned}e_i(k) &= \phi_{R\_i}(k)-v_i(k)-\hat{\phi}_i(k)\\ &= \phi_{R\_i}(k)-v_i(k)-P(z)e_i(k)\\ &= S(z)\left[\phi_{R\_i}(k)-v_i(k)\right]\\ &= S(z)\left[\phi_{R\_i}(k)-\psi^T(k)\hat{\theta}_i(k-1)\right]\\ &= S(z)\,e_i^0(k)\end{aligned} \tag{6.8}$$

where $S(z)=\frac{1}{1+P(z)}$ is the sensitivity function; $P(z)=\frac{C(z)}{z-1}$ is the open loop transfer function of the digital PLL circuit.

Hence, $e_i^0(k)$ can be obtained from the measurable quantity $e_i(k)$ if $[S(z)]^{-1}$ exists and is implementable, i.e.

$$\begin{aligned}e_i^0(k) &= [S(z)]^{-1}e_i(k)\\ &= \bar{s}(k)*e_i(k)\end{aligned} \tag{6.9}$$

where $\bar{s}(k)$ is the impulse response of $[S(z)]^{-1}$.

The PAA in Equation (6.7) can be modified in various ways. For example, it is possible to introduce a forgetting factor $0<\lambda<1$. In such case, the updating equation for $F_i(k)$ becomes

$$F_i(k) = \frac{1}{\lambda}\left[F_i(k-1)-\frac{F_i(k-1)\psi(k)\psi^T(k)\,F_i(k-1)}{\lambda+\psi^T(k)\,F_i(k-1)\psi(k)}\right] \tag{6.10}$$

Notice that Equation (6.7) is a standard least square algorithm. Hence, $\lim_{k\to\infty} e_i^0(k) = 0$ and $\lim_{k\to\infty} \hat{\theta}_i(k) = \theta_i$ can be easily proved, which guarantee asymptotic stability of the feedback control system.

6.4.2 Effect of Non-repeatable Timing Error (NRTE) on the Performance of PAA

If NRTE $\phi_{NR\_i}(k)$ is present in the system, Equation (6.8) can be rewritten as

$$\begin{aligned} e_i(k) &= \phi_{R\_i}(k) - v_i(k) - \hat{\phi}_i(k) \\ &= \phi_{R\_i}(k) - v_i(k) - P(z)e_i(k) - \phi_{NR\_i}(k) \\ &= S(z)\left[\phi_{R\_i}(k) - v_i(k) - \phi_{NR\_i}(k)\right] \\ &= S(z)\left[\psi^T(k)\left(\theta_i - \hat{\theta}_i(k-1)\right) - \phi_{NR\_i}(k)\right] \end{aligned} \tag{6.11}$$

Equation (6.11) shows that the performance of PAA is affected by the presence of NRTE $\phi_{NR\_i}(k)$ since $e_i(k)$ is used in PAA updating equation. Hence the estimation accuracy of $v_i(k)$ is degraded because of $\phi_{NR\_i}(k)$.

Moreover, from Figure 6.10, the write head phase signal $\hat{\phi}_i(k)$ is written as

$$\hat{\phi}_i(k) = \phi_{NR\_i}(k) + P(z)e_i(k)$$

which shows the quality of the written servo patterns is adversely affected due to the contamination of NRTE $\phi_{NR\_i}(k)$.

6.4.3 Error Shaping Filter Design

In this subsection, an error shaping filter $W(z)$ (as shown in Figure 6.10) is designed to enhance the estimation performance of PAA and reduce the effect of NRTE in the system. The error shaping filter is designed by introducing the new correction signal

$$u_i(k) = v_i(k) + W(z)e_i(k) \tag{6.12}$$

Introducing Equation (6.12) and adding the filter $W(z)$, the system error $e_i(k)$ in Equation (6.11) is rewritten as

$$\begin{aligned} e_i(k) &= S(z)\left[\phi_{R\_i}(k) - u_i(k) - \phi_{NR\_i}(k)\right] \\ &= S(z)\left[\phi_{R\_i}(k) - v_i(k) - \phi_{NR\_i}(k) - W(z)e_i(k)\right] \end{aligned} \tag{6.13}$$

Analysis

- In Equation (6.13), the term $\left(\phi_{R\_i}(k) - v_i(k)\right)$ is repeatable residual error which is determined by the PAA in Equation (6.7).

- The term $\left(-\phi_{NR\_i}(k)-W(z)e_i(k)\right)$ is non-repeatable residual error. It can be seen that if the magnitude of $\left(-\phi_{NR\_i}(k)\right)$ is reduced by $W(z)e_i(k)$, then the magnitude of the system error $e_i(k)$ will also be reduced.

Therefore, it is desired to design $W(z)$ to reduce the NRTE $\phi_{NR\_i}(k)$. For this purpose, $W(z)$ can be designed to extract the non-repeatable components from $e_i(k)$, i.e. to remove its repeatable components out.

The filter $W(z)$ is considered as a comb filter

$$W(z) = \frac{\sqrt{1+\alpha^N}}{2}\left(\frac{1-z^{-N}}{1-\alpha^N z^{-N}}\right) \tag{6.14}$$

where α is the scaling factor.

Analysis

- The filter $W(z)$ is stable if and only if $|\alpha|<1$. Its zeros are $Z_k = e^{2k\pi j/N}$ and poles are $P_k = \alpha \cdot e^{2k\pi j/N}$ for $k = 0, 1, \cdots, N-1$.
- The frequency response of $W(z)$ consists of a series of evenly-spaced notches with appearance of a comb. Figure 6.11 shows an example of the comb filter with delay length $N = 50$ and factor coefficient $\alpha = 0.69$.

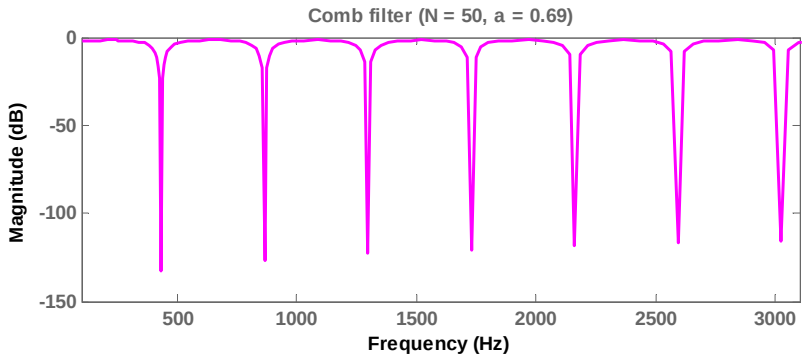

Figure 6.11: Frequency response of a comb filter in Equation (6.14) ($N = 50$ and $\alpha = 0.69$)

- In Equation (6.14), the factor $\sqrt{1+\alpha^N}/2$ is designed for setting the filter gain to unity in the passband (i.e. between the notches). Figure 6.12 shows the closed-loop responses (i.e. from $\phi_{R\_i}(k)$ to $\hat{\phi}_i(k)$) before and after adding the filter $W(z)$, which have the same bandwidth except that the system with $W(z)$ has the notches that are useful for reducing the NRTE in the system.
- The repeatable components of the system error $e_i(k)$ are eliminated by $W(z)$. In other words, its non-repeatable components are isolated to reduce the NRTE $\phi_{NR\_i}(k)$.

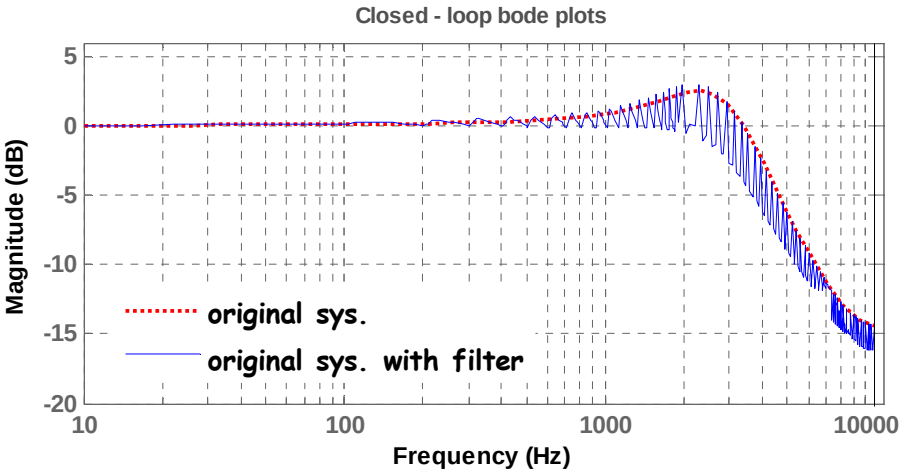

Figure 6.12: Closed-loop response of the systems with and without the comb filter

6.5 Simulation Study

In the simulation study, 19 tracks of RTE ϕ_{R_i} (as shown in Figure 6.13) and NRTE ϕ_{NR_i} (as shown in Figure 6.14) are collected from a real disk drive. The spindle speed is 7200 *rpm* and the number of servo sector in one track is $N = 208$. The low pass filter in the PLL circuit is chosen as

$$C(z) = \frac{0.477z - 0.4725}{z^2 - z} \tag{6.15}$$

The sampling period is $T = 4.63 \times 10^{-5}$ sec.

Remark 6.5.1

- In practical SSW process, the spindle speed is generally slower than the nominal value. Hence the sampling period in the simulation is chosen as $T = 4.63 \times 10^{-5} sec$ which is larger than $\frac{60}{208 \times 7200} = 4 \times 10^{-5} sec$.
- The RTE profile looks like sinusoidal because it is mainly contributed by the disk eccentricity and written-in timing error, which are periodic and synchronized to the disk rotation.
- The NRTE profile is continuous since it is mainly caused by the spindle speed variation.
- In the simulations, all the signals are normalized to $T_s = 4ns$.

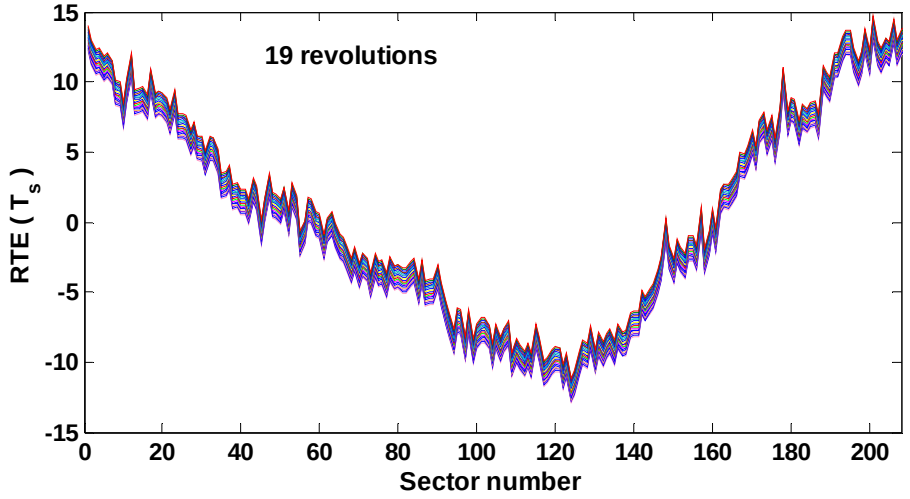

Figure 6.13: RTE profile (19 tracks)

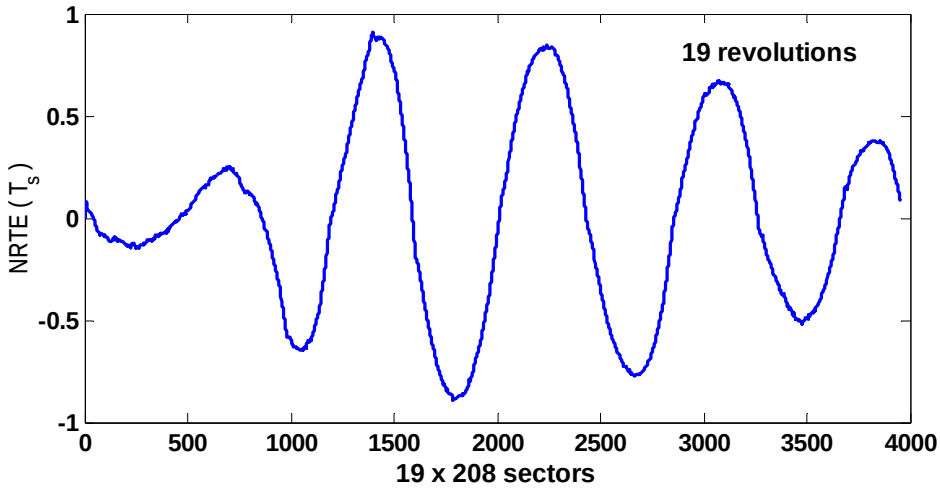

Figure 6.14: NRTE profile (19 tracks)

6.5.1 PAA with and without NRTE

In order to show the effectiveness of the proposed PAA, RTE ϕ_{R_i} and NRTE ϕ_{NR_i} are injected to SSW system for 19 revolutions. Two cases are firstly studied and compared: without PAA (i.e. the original system in Figure 2.7) and with PAA (i.e. the system in Figure 6.10 but without the filter $W(z)$). In the simulation study, the forgetting factor was chosen as 0.86 and n in Equation (6.4) was 1.

Figure 6.15 shows the write head performance ($\sigma\left[\hat{\phi}_i\right]$) for 19 tracks. The solid line represents the case with PAA, which achieves a much better timing pattern with 77.7% smaller average timing variation than the original system as shown in the dotted line.

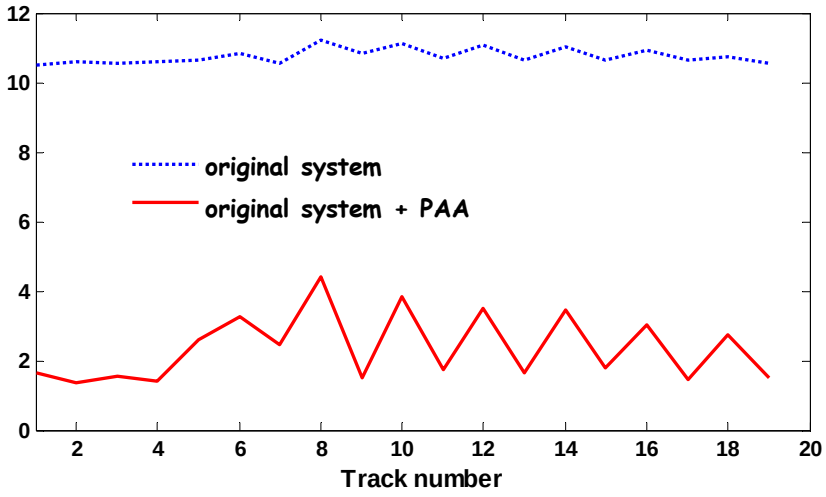

Figure 6.15: Written timing deviation for 19 tracks with PAA ($avg\left[\sigma\left(\hat{\phi}_i\right)\right] = 2.36\,T_s$) and without PAA ($avg\left[\sigma\left(\hat{\phi}_i\right)\right] = 10.56\,T_s$)

In the second study, the estimation performance of PAA is compared between the systems with and without NRTE ϕ_{NR_i}. The estimation performance in the track i is defined as

$$P_i = \frac{\sigma\left[v_i - \phi_{R\_i}\right]}{\sigma\left[\phi_{R\_i}\right]} \times 100\% \qquad (6.16)$$

In Figure 6.16, the solid line represents the injected RTE source. The dashed line represents the estimated RTE in ideal case, i.e. NRTE ϕ_{NR_i} does not exist in the system. However, when ϕ_{NR_i} is present, the estimation performance is clearly degraded as shown in the dotted line. This implies that the presence of ϕ_{NR_i} adversely affects PAA estimation performance. The average P_i is compared in Figure 6.17. It can be seen that the estimation performance is degraded from 18.0% to 28.8% if the system is contaminated by NRTE ϕ_{NR_i}.

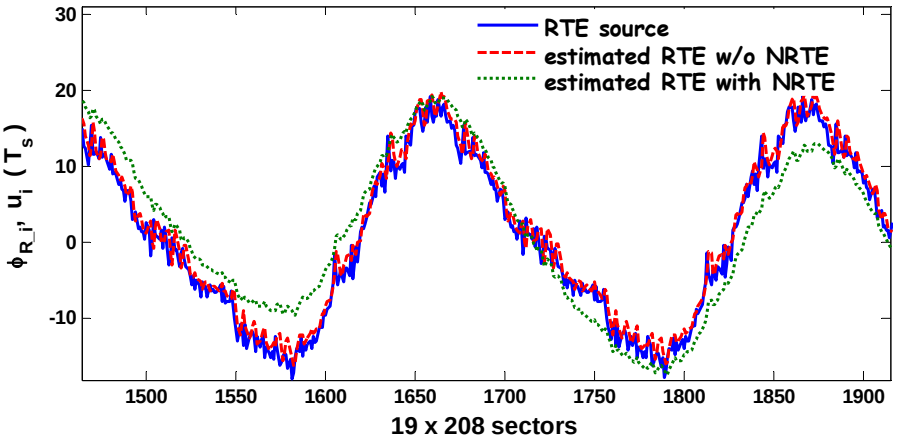

Figure 6.16: RTE and estimated RTE profiles for 19 tracks

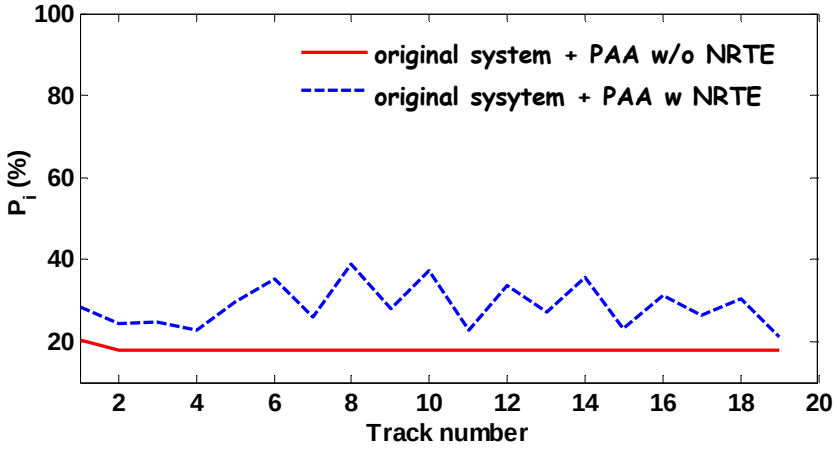

Figure 6.17: Comparison of PAA performance with ϕ_{NR_i} ($avg[P_i] = 28.8\%$) and without ϕ_{NR_i} ($avg[P_i] = 18.0\%$)

Figure 6.18 shows the average $\sigma[\hat{\phi}_i]$ in the systems with and without NRTE ϕ_{NR_i}. From the figure, we learn that the quality of the written timing pattern is 62.7% better if the system is not contaminated by NRTE ϕ_{NR_i}.

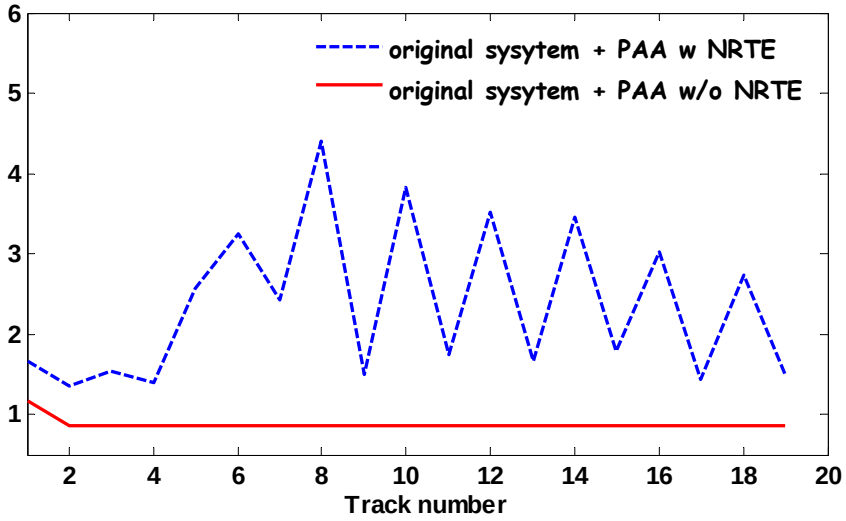

Figure 6.18: Written timing deviation using PAA with ϕ_{NR_i} ($avg\left[\sigma(\phi_i)\right] = 2.36\,T_s$) and without ϕ_{NR_i} ($avg\left[\sigma(\phi_i)\right] = 0.88\,T_s$)

6.5.2 PAA with Error Shaping Filter

An error shaping filter $W(z)$ in Equation (6.14) is designed to reduce the effect of NRTE ϕ_{NR_i} in the system and improve the servowriting quality. The scaling factor in Equation (6.14) is $\alpha = 0.69$. Figure 6.19 – Figure 6.21 show the performance of $W(z)$. Figure 6.19 and Figure 6.20 compare the performance of the feedforward signal u_i for 19 tracks using PAA with and without the filter $W(z)$. The performance of u_i is defined in Equation (6.17), which is similar to Equation (6.16),

$$P_i = \frac{\sigma\left[u_i - \phi_{R\_i}\right]}{\sigma\left[\phi_{R\_i}\right]} \times 100\% \qquad (6.17)$$

It is shown that the system with PAA and $W(z)$ achieves 26.8% more accurate RTE estimation than the system with the PAA only.

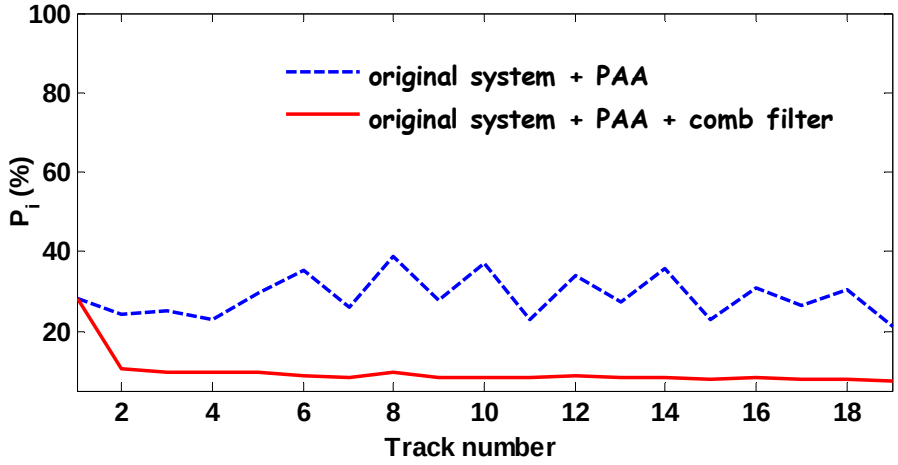

Figure 6.19: Performance of u_i using PAA with $W(z)$ ($avg[P_i]=9.67\%$) and without $W(z)$ ($avg[P_i]=28.8\%$)

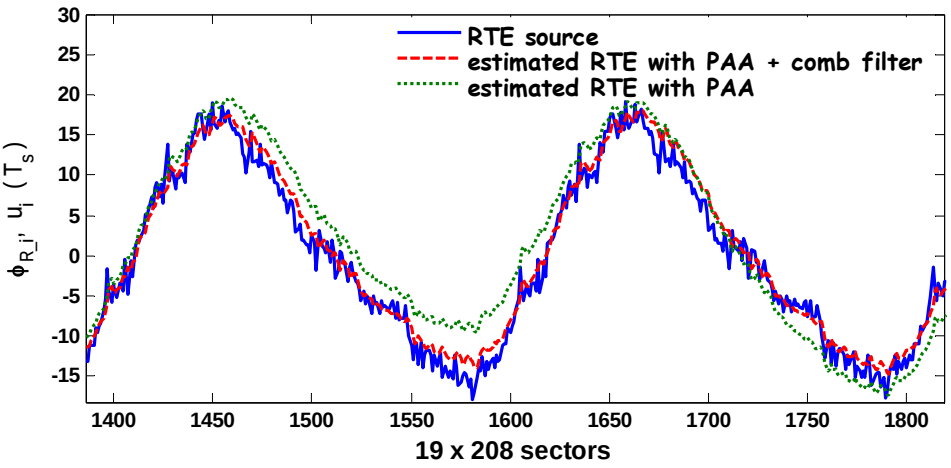

Figure 6.20: RTE and feedforward signal u_i profile for 19 tracks

In order to further verify the performance of the error shaping filter $W(z)$, Figure 6.21 shows $\sigma\left[\hat{\phi}_i\right]$ in the systems with and without $W(z)$. The solid line represents the system using PAA and the comb filter $W(z)$ and it is much smoother since the effect of NRTE ϕ_{NR_i} has been reduced by $W(z)$. The resulting average $\sigma\left[\hat{\phi}_i\right]$ is reduced from $2.36\,T_s$ (using the PAA only as shown in the dashed line) to $0.93\,T_s$.

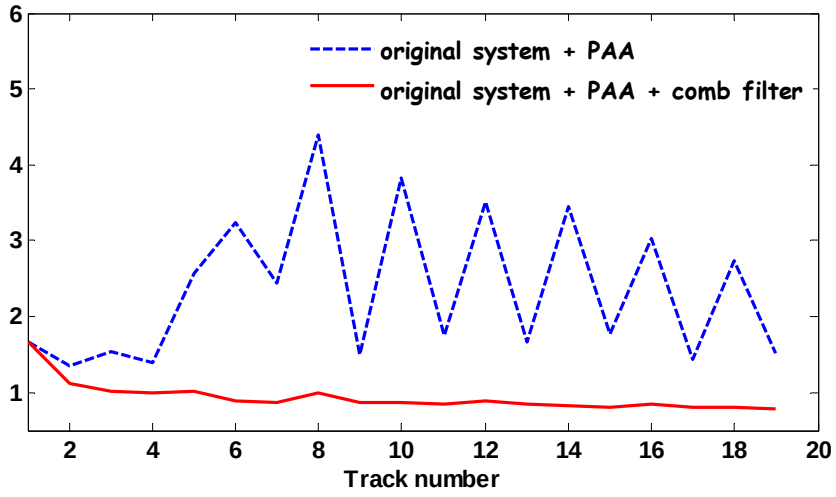

Figure 6.21: Written timing variation using PAA with $W(z)$ ($avg\left[\sigma\left(\hat{\phi}_i\right)\right] = 0.93\,T_s$) and without $W(z)$ ($avg\left[\sigma\left(\hat{\phi}_i\right)\right] = 2.36\,T_s$)

6.6 Chapter Summary

This chapter presented a control algorithm to tackle with the repeatable error in discrete linear repetitive processes, which dominates the system dynamics. In the timing control loop of HDD spiral based SSW process, the repeatable timing error is present in the reference and therefore it is amplified through the closed-loop dynamics to generate the written timing patterns. Using this system as the test bed, a recursive least square based parameter adaptive algorithm was proposed to estimate and cancel the repeatable timing error. In the simulation result, the repeatable timing error was cancelled up to 90%. An error shaping filter was further designed to improve the estimation accuracy of repeatable timing error and to reduce the effect of non-repeatable timing error in the process. The resulting quality of the written timing patterns was improved by as much as 91% over the original system.

Chapter 7

A Real-time Control Scheme Design

7.1 Introduction

In Chapter 6, we learned that in spiral based SSW process, the drifts of spiral locations (due to the disk eccentricity and thermal expansion) and sync marks' position error (due to the mechanical jitter and thermal expansion) cause the repeatable timing error. The repeatable timing error consequently leads to the shifts of timing windows in the demodulation circuit. As a result, the written product servo sectors are shifted from sector to sector, resulting in non-uniform alignment. In other words, the servo write clock (i.e. the sampling period of the servo system) is non-uniform or irregular. Analyzed from the computer system perspective, such system is also referred to as the real-time or event-driven control system (the write head clock is updated only when a sync mark is detected in the demodulation window by the read head, i.e. the system sampling is triggered by the read head signal). In this real-time control system, the sampling period is time-varying due to the repeatable timing error.

The feedback controllers in SSW servo system (including the position control loop and timing control loop), however, are classically designed. In other words, they are designed by assuming the sampling period is strictly constant. Generally, there are two ways to design the feedback controllers. One is to use the discrete-time control theory; the other is to discretize a continuous-time design. Both of these methods assume a constant sampling period to calculate the controller parameters. When such feedback controller is implemented in a non-uniform sampling system, the control input to the plant is inaccurate. As a result, the controlled system response degrades or even causes instability and sample data loss.

In this chapter, a novel scheme to determine the control parameters in the phase-locked loop based on Kalman filter theory is proposed to deal with the non-uniform sampling issue in this repetitive process and improve the quality of self-servowriting. The remainder of this chapter is organized as follows. Section 7.2 presents the effect of sampling jitter in the real-time control system through two examples. The effect of repeatable timing error in spiral based SSW servo system is analyzed in Section 7.3. In Section 7.4, the proposed control scheme based on Kalman filter theory is presented.

Section 7.5 verifies the effectiveness of the proposed control algorithm through the computer simulations. Finally, Section 7.6 gives the chapter summary.

7.2 Real-time Control System of Spiral based SSW Process

7.2.1 Timing in Real-time Control Loop

In classical digital control system design, all the sampled data signals are assumed to be regularly, synchronously, and equally time spaced, being T_c the constant sampling period [3]. As shown in Figure 7.1, a digital control loop consists of three main operations: sampling (i.e. data collection), control algorithm computation, and actuation (i.e. output transmission). In ideal case, the control algorithm is executed periodically with a constant sampling period determined by the process dynamics and the requirements on the closed loop performance.

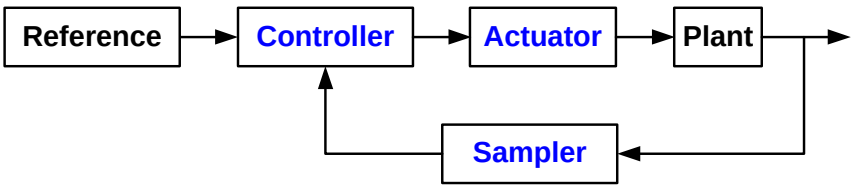

Figure 7.1: Digital control loop

The two basic timing constraints of a digital control loop are shown in Figure 7.2. The first is the period which should be constant, i.e. the sampling should be performed at the sampling instants in every period. The second constraint involves the input-output latency, also known as the control delay. This should be as small as possible, i.e. the control computation should start and finish quickly after the sample is available and actuation should occur immediately after the control computation, or at a fixed instant after the sampling. However, this is not the case in most industrial applications given that the computations take time and may have to contend with other computations for processing and other resources.

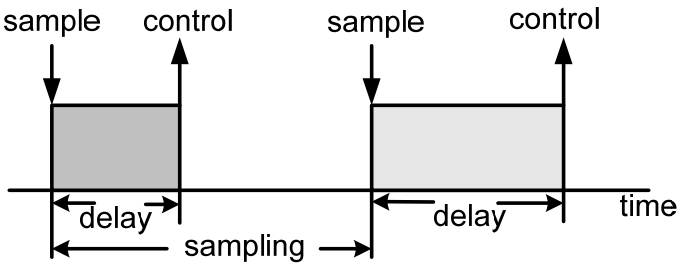

Figure 7.2: Basic timing constraints in a digital control loop

When a control algorithm is executed by a task (i.e. performing the three actions sequentially) or by a set of subtasks (where each task performs one or more parts of a control loop) in a multitasking real-time system, the assumptions of constant sampling period or zero control latency are often not met and various forms of jitters are introduced. These jitters can be classified into two categories:

(1) Sampling jitter: the time intervals between consecutive sampling points are not constant, which is also referred to as non-uniform or irregular sampling.

(2) Sampling-to-actuation delays: even if the sampling occurs at regular intervals, there is a delay between when the sampling arrives and when the actuation response occurs after the control computation.

The above two jitters cause control performance degradation, instability, or even sample data loss [3]. From the perspective of a control system, the sampling jitter and delay jitter can be interpreted as disturbances acting on the control system. In Chapter 6, the sampling jitter in spiral based SSW process was interpreted as the repeatable timing error in the system. In this chapter, we are analyzing from the perspective of a computer system to deal with the non-uniform sampling period issue caused by the sampling jitter. We are only interested in the sampling jitter since it is the main issue in spiral based SSW process.

7.2.2 Effect of Sampling Jitter and Compensation Methods

Generally speaking, computer-based control systems can be designed by two methods: discretization of a continuous-time design or discrete-time design. In both cases, the final controller obtained by using a suitable control design strategy must meet the specified closed-loop system performance requirements taking into account the dynamics of the controlled process. Such controller is characterized by the design parameters that are highly dependent on the nominal sampling period T_c used in the design procedure. The control computation, however, is executed at the actual sampling periods which may not be of the constant value T_c. Therefore, the control performance degrades.

In many cases, however, better control performance can be achieved if the controller is allowed to actively compensate for the variations from sample to sample by recalculating or redesigning the controller parameters. This requires that the necessary sampling period measurements are available.

In the following subsection, two examples are provided to show how these two compensation methods reduce the effect of the sampling jitter, respectively.

Example 1 – DC servo motor

Consider proportional-derivative (PD) control of a DC motor, as shown in Figure 7.3. The goal of the control algorithm is to make the servo position $y(t)$ follow the reference position $r(t)$ as closely as possible. The servo motor is mathematically described by a

continuous-time transfer function $G(s)=\frac{1000}{s(s+1)}$. A good implementation of the PD controller, which includes filtering the derivative part, is given as

$$\begin{aligned} P(t) &= K\big(r(t)-y(t)\big) \\ D(t) &= a_d D(t-T_c)+b_d\big(y(t-T_c)-y(t)\big) \\ u(t) &= P(t)+D(t) \end{aligned} \tag{7.1}$$

where K is the proportional gain, T_c is the nominal sampling period, T_d is the derivative time constant, M is typically in the range of $3\sim 30$, $a_d=\frac{T_d}{MT_c+T_d}$, and $b_d=\frac{MKT_c}{MT_c+T_d}$.

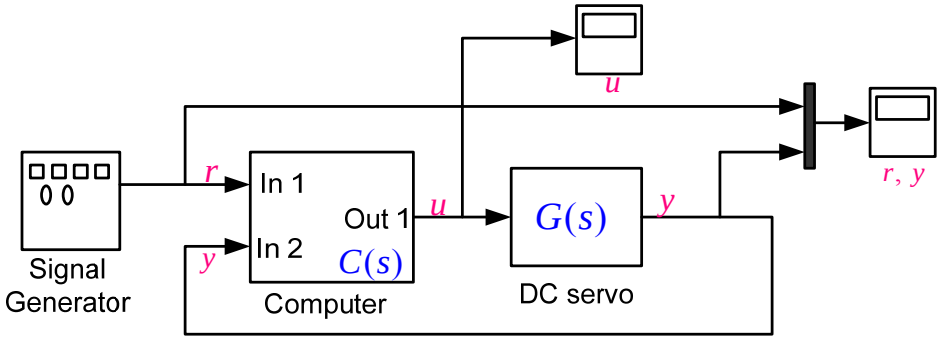

Figure 7.3: Block diagram of PD controlled DC servo motor

A nominal sampling period of $T_c=10ms$ is chosen, and the PD controller is tuned to give a fast and well-damped response to set-point changes. The resulting parameters are $K=1$, $T_d=0.04$, and $M=30$. The parameters a_d and b_d are precalculated, assuming that the sampling interval takes the nominal value, i.e. $10ms$. Three simulation cases are studied.

(1) The first simulation of the closed-loop system, where there is no jitter in the sampling intervals, is shown in Figure 7.4. It shows that the controller behaves as expected and the system performance is satisfactory.

(2) The second simulation study, where the actual sampling interval denoted by T_k varies randomly between $T_{\min}=2ms$ and $T_{\max}=18ms$, is shown in Figure 7.5. In this case, the control signal $u(t)$ is calculated by using the output signal measured at the time instants with the actual sampling interval T_k instead of the nominal sampling interval T_c, i.e.

$$\begin{aligned} u(t) &= P(t)+D(t) \\ &= K\big(r(t)-y(t)\big)+a_d D(t-T_k)+b_d\big(y(t-T_k)-y(t)\big) \end{aligned}$$

While the parameters a_d and b_d have been fixed by T_c. It can be seen that the sampling jitter causes the controller to repeatedly take either too small or too large

actions. The resulting performance is quite poor. This is especially visible in the control signal.

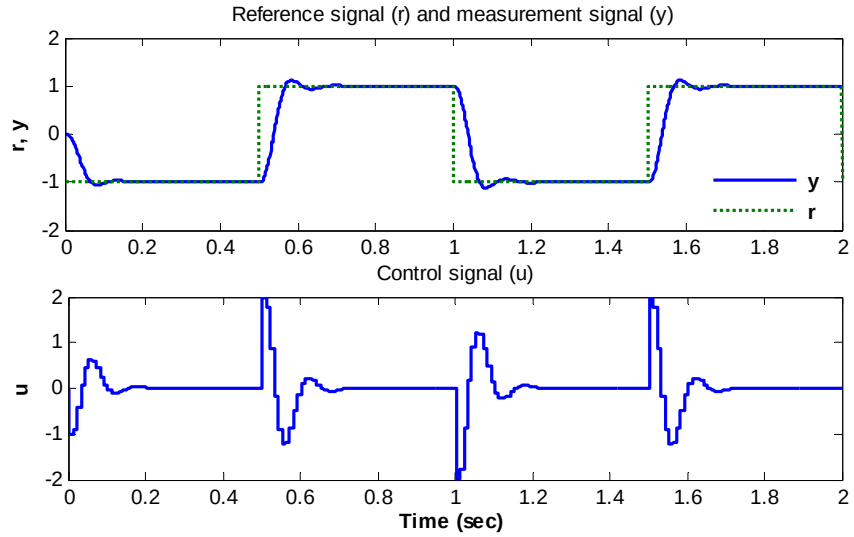

Figure 7.4: Good control performance without sampling jitter

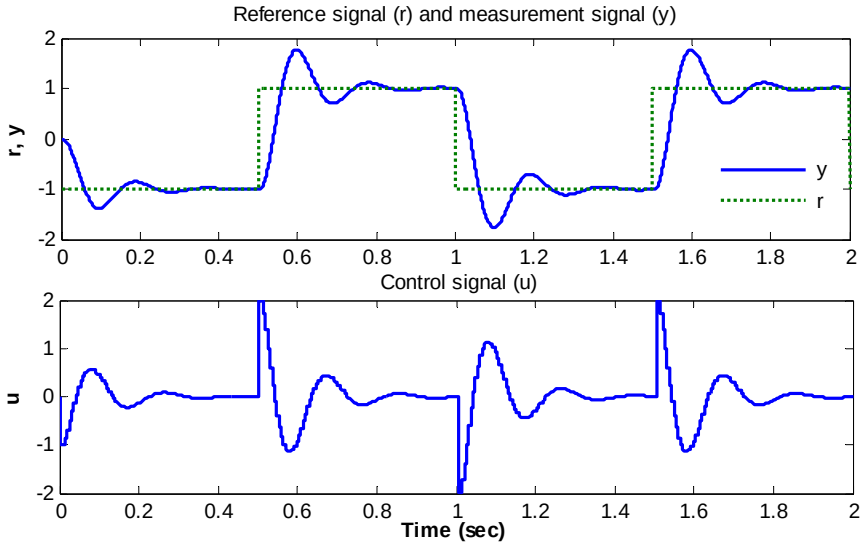

Figure 7.5: Degraded control performance due to sampling jitter

(3) Finally, in the third study case, the compensation for sampling jitter is considered. This is done by measuring the actual sampling intervals T_k and using them to recalculate the controller parameters a_d and b_d at each sample instant, i.e.

$$a_d = \frac{T_d}{MT_k + T_d}$$

$$b_d = \frac{MKT_k}{MT_k + T_d}$$

Figure 7.6 shows that this version of controller handles the sampling jitter well.

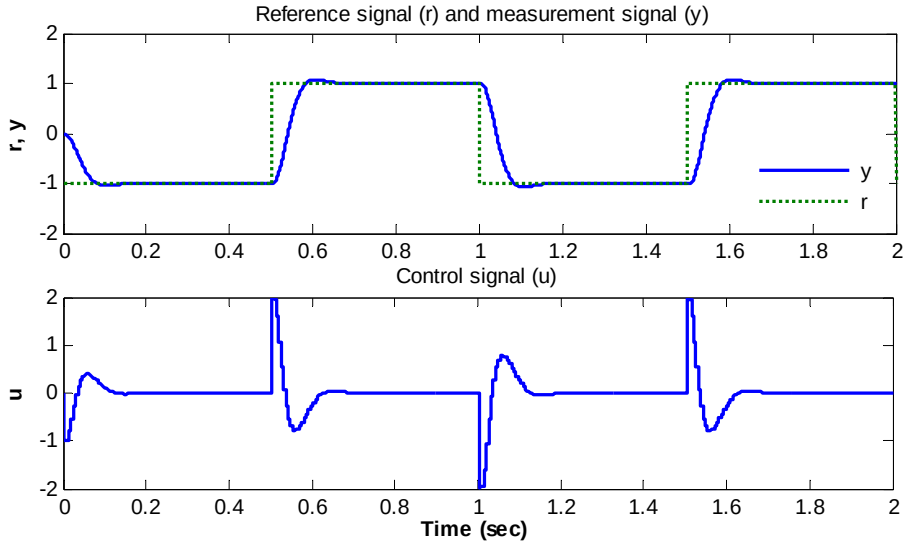

Figure 7.6: Good control performance after compensating the sampling jitter

Example 2 – PLL circuit in spiral based SSW system

In this example, the controlled process $G(s)$ in Figure 7.3 is the voltage controlled oscillator (VCO) in the PLL circuit used for spiral based SSW timing control system. It is mathematically modeled as an integrator ($G(s) = \frac{2\times10^4}{s}$) and the feedback controller $C(s)$ is chosen as a low pass filter (LPF), which is designed by using a nominal sampling period of $T_c = 4.63\times10^{-5}$ sec. Compared to Example 1, four simulation cases are studied.

(1) In the first simulation, the nominal sampling period $T_c = 4.63\times10^{-5}$ sec is assumed, i.e. there is no sampling jitter in the system. The result is represented by the line with the circles (○) in Figure 7.7. It is obvious that the controller performs well.

(2) In the second simulation, the sampling period is randomly varied between $T_{\min} = 3\times10^{-5}$ sec and $T_{\max} = 8\times10^{-5}$ sec. The simulation result is illustrated by the line with the squares (□) in Figure 7.7. It shows that the controller performance deteriorates substantially due to the sampling jitter.

(3) In the third study, the compensation method used in Example 1 is considered, i.e. the parameters of the LPF are recalculated by using the actual sampling intervals. This

compensation method, however, does not work well for the spiral based SSW timing control loop (as shown in the line with the triangles (Δ) in Figure 7.7).

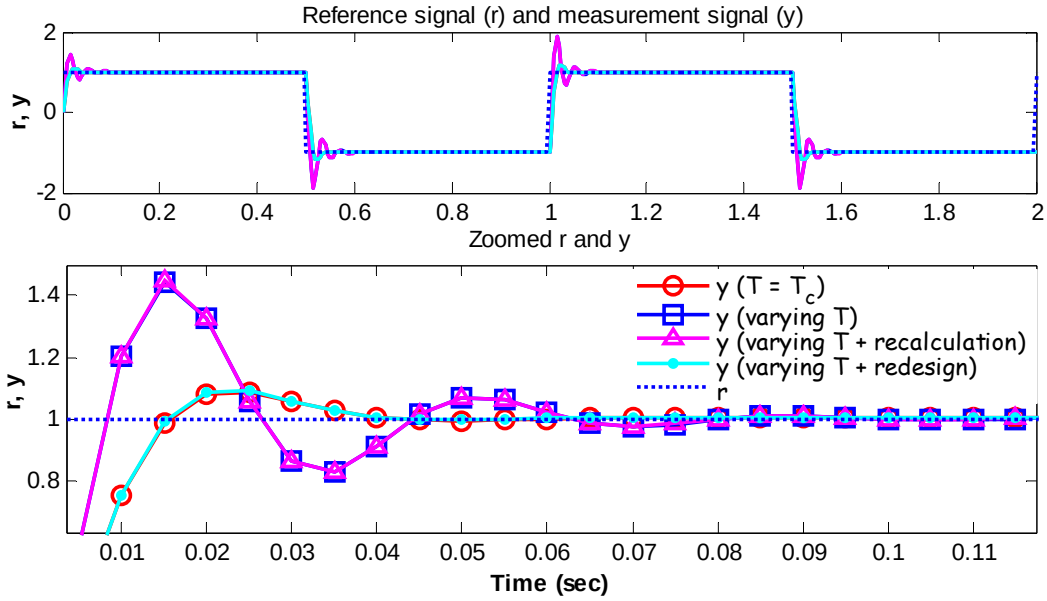

Figure 7.7: Control performance comparison in real-time spiral based SSW process

(4) Therefore, the forth simulation case is conducted, where the LPF is redesigned by tuning the gain. In the same figure, the resulting response is shown in the line with the dots (•). It is observed that the redesign of LPF parameter compensates the sampling jitter very well and the control performance is as good as that of the system without the sampling jitter, i.e. the system in the first simulation study.

Notice that in the above two examples, both of the sampling jitter compensation methods (recalculation and redesign) are applied assuming the actual sampling intervals are measurable. However, in practical spiral based SSW process, the actual sampling intervals are not directly measurable, i.e. the above two compensation methods are not applicable. Hence, it is necessary to design a novel approach to compensate for the control degradation due to the sampling jitter and guarantee the system stability in real-time spiral based SSW process.

Before presenting the proposed control algorithm, the effect of repeatable timing error is analyzed from the real-time control system perspective in the following section, assuming there is no sampling-actuation delay or sample data loss.

7.3 Effect of Repeatable Timing Error in Spiral based SSW Servo System

7.3.1 Non-uniform sampling due to Repeatable Timing Error

During spiral based SSW process, the read head reads the prewritten spiral tracks and generates a spiral wedge crossing signal, which is then processed by the demodulation circuit for recovering the spiral position and timing information in the spiral tracks. Figure 7.8 shows the control block diagram of spiral based SSW servo system, which includes two control loops: position control loop and timing control loop. These two control loops use the demodulated spiral position information and timing information to generate the position error signal (PES) and servo write clock for the write head. In the position control loop, the voice coil motor (VCM) is controlled to maintain the head over the target track centerline; and in the timing control loop, a phase locked loop (PLL) is used and controls the numerically controlled oscillator (NCO) to update the servo write clock $T_w(k)$, where k denotes the index of servo sectors, i.e. the index of sampling instants in the servo system.

Notice that these two control loops are sampled with the same sampling period $T_w(k)$, which is time-varying and non-uniform due to the repeatable timing error caused by the drifts of spiral locations and sync marks' position error. However, the feedback controllers in these two control loops, $C_p(z)$ and $C(z)$, are designed by using the nominal constant sampling period T_c. Therefore, the control inputs, $u_p(k)$ for the VCM and $u(k)$ for the NCO, are inaccurate. Moreover, it is desired to write the product servo patterns uniformly along the circular tracks. In other words, the actual servo write clock $T_w(k)$ should ideally and quickly converges to T_c. So $T_w(k)$ is the key for improving the quality of product servo patterns and the timing control loop in spiral based SSW servo system is critical.

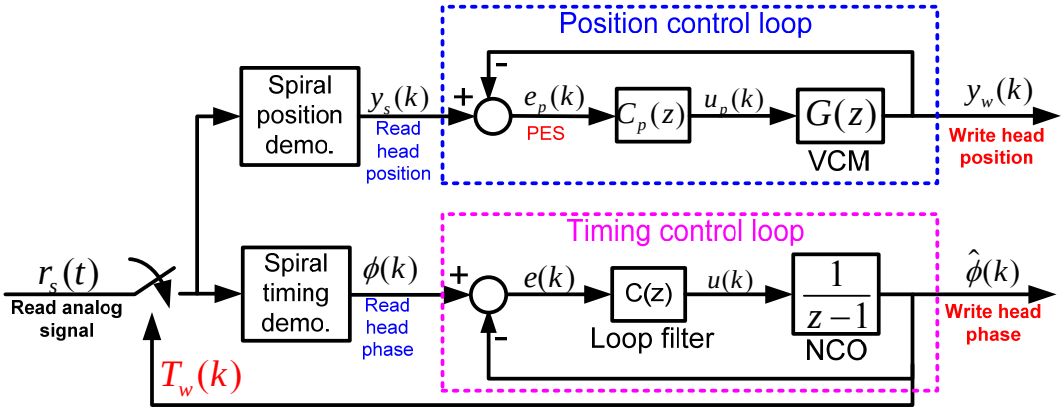

Figure 7.8: Control block diagram of spiral based SSW servo system

7.3.2 Digital Phase Locked Loop in Timing Control loop

In HDD read/write channel, the second order digital phase locked loop (PLL) as shown in Figure 7.9 is widely used for synchronizing the read head clock with the write head clock. Its loop filter is generally a proportional integral (PI) controller and written as

$$C(z) = C_1 + C_2 \frac{1}{1 - z^{-1}} \tag{7.2}$$

where C_1 is the parameter of the proportional term and C_2 is the parameter of the integral term. Notice that the PLL circuit in Figure 7.9 is equivalent to the Figure 2.6, except that the track index i is ignored in the signals for the convenience of analysis.

The PLL is known as a clock tracking control system, wherein the phase and frequency of its output signal follow the phase and frequency of its input signal. So in spiral based SSW timing control loop, the write head phase and frequency follow the read head phase and frequency. The read head frequency means the frequency of detected sync marks in the read head signal.

Notice that the read head frequency is non-uniform because of the repeatable timing error, so the write head clock $T_w(k)$ is also non-uniform. However, the PI loop filter is designed based on the nominal constant sampling period. In other words, the PI constant parameters C_1 and C_2 are designed for uniform sampling system. Hence, the non-uniform sampling period $T_w(k)$ damages the quality of the product servo patterns and causes two problems in the timing control loop: (1) the control input $u(k)$ for the NCO is inaccurate; (2) the written product servo patterns are not uniformly aligned in the circular tracks. In the following section, the proposed compensation scheme is proposed to deal with these two issues.

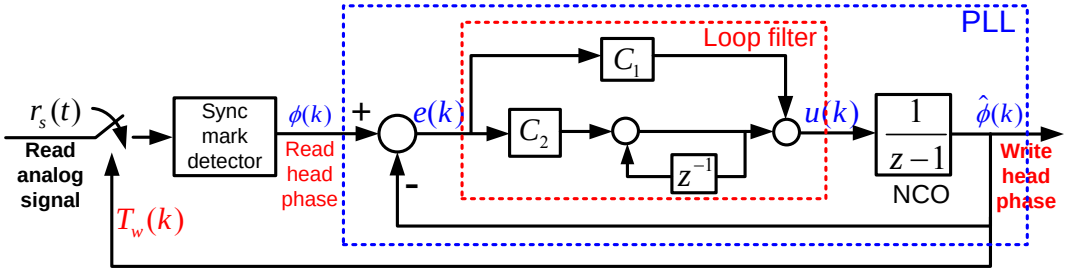

Figure 7.9: Phase locked loop in spiral based SSW timing control loop

7.4 Proposed Compensation Scheme based on Kalman Filter

7.4.1 Noisy Timing System Modeling

Before designing the compensation scheme, it is important to analyze the timing system in the prewritten spiral tracks. Refer to Figure 7.10, where $t(k)$ $(k = 0,1,\cdots N-1)$ denotes

the ideal uniform timing mark locations with constant interval T_c. In conventional PLL circuit, $t(k)$ is used as the reference timing location. In the figure, $t_s(k)$ denotes the timing location of spiral track k as shown in the dashed lines. And $T_w(k)$ represents the timing location of the write head, as shown in the dotted lines.

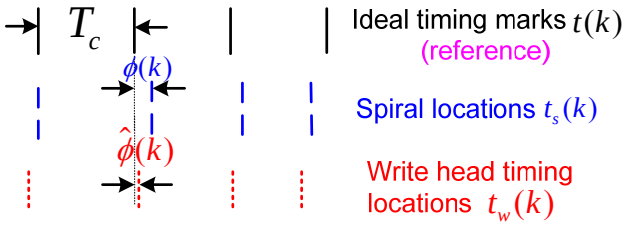

Figure 7.10: Illustration of read/write head timing locations in conventional PLL circuit

Using the above notations, the read head phase (expressed in radians), $\phi(k)$ in Figure 7.9, can be written as

$$\phi(k) = \left(t_s(k) - t(k)\right) \cdot \frac{2\pi}{NT_c} \tag{7.3}$$

where N denotes the number of servo sectors in one circular data track, and T_c is the nominal sampling period of the servo system.

The corresponding angular frequency error (expressed in radians per sample) is denoted by $f(k)$. It can be written as

$$f(k) = \left(\frac{1}{T_c} - \frac{1}{t_s(k+1) - t_s(k)} \right) \left(t_s(k+1) - t_s(k)\right) \cdot \frac{2\pi}{N} \tag{7.4}$$

Then we have the mathematical model of the noisy timing system in the spiral tracks, which is written as

$$\begin{aligned} \phi(k+1) &= \phi(k) + f(k) \\ f(k+1) &= f(k) + w(k) \end{aligned} \tag{7.4}$$

where $w(k)$ is the mechanical jitter (i.e. process noise) related to the angular frequency error. The mechanical jitter includes the disk disturbance and noise in the spiral writer. Here we assume the read head phase $\phi(k)$ is indirectly affected by $w(k)$.

The read head noisy signal is expressed as

$$r(k) = \phi(k) + v(k) \tag{7.5}$$

where $v(k)$ is the measurement jitter (i.e. observation noise) including the preamplifier noise and the A/D converter noise.

Defining the augmented state vector $x(k)=\begin{bmatrix}\phi(k)\\ f(k)\end{bmatrix}$, Equation (7.4) and Equation (7.5) can be rewritten as

$$\begin{aligned} x(k+1) &= \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}x(k)+\begin{bmatrix}0\\ 1\end{bmatrix}w(k) \\ &= A_c x(k)+B_c w(k) \\ r(k) &= \begin{bmatrix}1 & 0\end{bmatrix}x(k)+v(k) \\ &= C_c x(k)+v(k) \end{aligned} \tag{7.6}$$

The noisy timing system in Equation (7.6) can be illustrated by Figure 7.11.

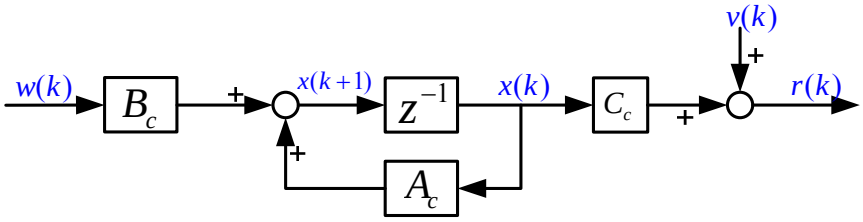

Figure 7.11: Noisy timing system in the spiral tracks

Since the physical sources of mechanical jitter $w(k)$ and measurement jitter $v(k)$ are independent, hence it is reasonable to assume that they are independent. It is also assumed that both of them are white and Gaussian with zero mean and covariance W and V, respectively, i.e.

$$\begin{aligned} w(k) &\sim N\left(0,\sigma_w^2\right) = N(0,\ W) \\ v(k) &\sim N\left(0,\sigma_v^2\right) = N(0,\ V) \end{aligned}$$

The initial state vector $x(0)$ is also assumed to be Gaussian and independent with $w(k)$ and $v(k)$, i.e.

$$\begin{aligned} x(0) &= \begin{bmatrix}\phi(0)\\ f(0)\end{bmatrix} \sim N\left(\begin{bmatrix}\phi_0\\ f_0\end{bmatrix},\begin{bmatrix}\sigma_\phi^2 & 0\\ 0 & \sigma_f^2\end{bmatrix}\right) = N\left(x_0, X_0\right) \\ E\left[(x(0)-x_0)w^T(k)\right] &= E\left[(x(0)-x_0)v^T(k)\right] = 0 \end{aligned}$$

It should be noted that the sampling jitter has be represented by two additive noise terms $w(k)$ and $v(k)$ in the discrete time Kalman filter formulation.

7.4.2 Proposed Control Scheme Design

Notice that the digital oscillator NCO in Figure 7.9 is an integrator and it has the exact same mathematical model as the timing system in Equation (7.6). Hence, it can be said that the NCO is the estimate of the noisy timing system in the prewritten spiral tracks. In other words, the digital PLL circuit in Figure 7.9 is a state estimator, wherein the loop filter controls the NCO to estimate and follow the phase and frequency of the read head signal.

In this section, we propose to redesign the digital PLL circuit by using the Kalman filter theory since Kalman filter has been well known as optimal state estimator. By using the noisy timing system in Equation (7.6), the discrete-time Kalman filter equations can be easily obtained as

$$\begin{aligned}
&\hat{x}(k|k) = \hat{x}(k|k-1) + F(k)\left[r(k) - C_c \hat{x}(k|k-1) \right] \\
&\hat{x}(k|k-1) = A_c \hat{x}(k-1|k-1) \\
&F(k) = M(k|k-1)C_c^T \left[C_c M(k|k-1) C_c^T + V \right]^{-1} \\
&M(k|k) = \left[I - F(k)C_c \right] M(k|k-1) \\
&M(k|k-1) = A_c M(k-1|k-1) A_c^T + B_c W B_c^T
\end{aligned} \tag{7.7}$$

where $\hat{x}(k|k) = \begin{bmatrix} \hat{\phi}(k|k) \\ \hat{f}(k|k) \end{bmatrix}$ is the a-posteriori state estimate including the phase estimate $\hat{\phi}(k|k)$ and angular frequency error estimate $\hat{f}(k|k)$; $\hat{x}(k|k-1)$ is the a-priori state estimate; $F(k) = \begin{bmatrix} F_1(k) \\ F_2(k) \end{bmatrix}$ is the Kalman filter gain; matrix $M(k|k)$ is the a-posteriori state estimation error covariance; and $M(k|k-1)$ is the a-priori state estimation error covariance.

The Kalman filter in Equation (7.7) can be equivalently represented as

$$\begin{bmatrix} \hat{\phi}(k+1|k) \\ \hat{f}(k+1|k) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi}(k|k-1) \\ \hat{f}(k|k-1) \end{bmatrix} + \begin{bmatrix} F_1(k) \\ F_2(k) \end{bmatrix} \left(r(k) - [1 \quad 0] \begin{bmatrix} \hat{\phi}(k|k-1) \\ \hat{f}(k|k-1) \end{bmatrix} \right) \tag{7.8}$$

Equation (7.8) can be illustrated by the following figure.

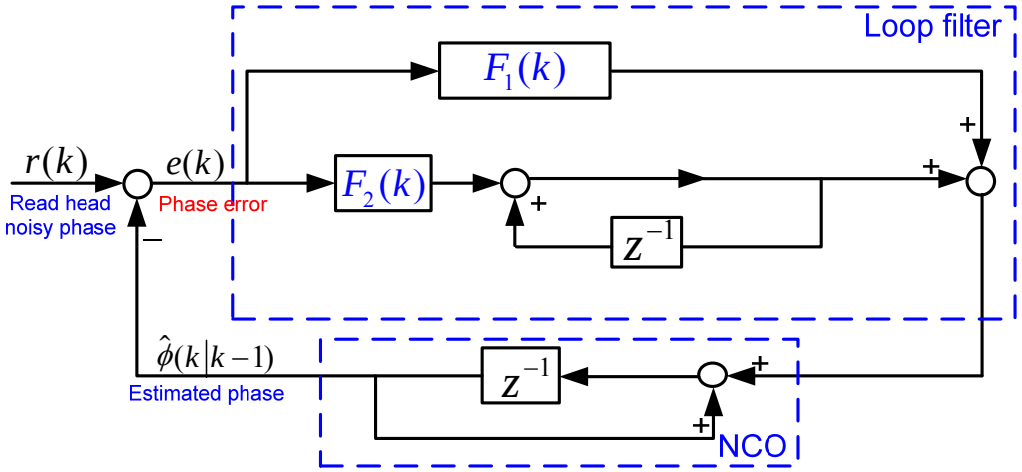

Figure 7.12: Illustration of Kalman filter in Equation (7.7)

From Figure 7.12, we can see that the proposed Kalman filter in Equation (7.7) has the same structure as the second order digital PLL circuit in Figure 7.9, where the upper part is a loop filter with the filter gain $F_1(k)$ and $F_2(k)$;and the lower part is the estimated timing system and it is equivalent to the digital oscillator NCO in the PLL circuit. The only difference between them is that: in PLL, the estimated timing system is the NCO, which is a digital circuit in the actual implementation; while in Kalman filter, the estimated timing system is involved and calculated in the Kalman filter equations. Hence, the proposed Kalman filter can be implemented as simply as the conventional PLL circuit in industry. Namely, the proposed method is a tuning method for the PLL parameters C_1 and C_2 based on the Kalman filtering theory. Comparing the PLL circuit in Figure 7.9 with the Kalman filter block diagram in Figure 7.11, it can be done by simply replacing C_1 with $F_1(k)$ and replacing C_2 with $F_2(k)$.

7.4.3 Analysis of Kalman Filter Design

Figure 7.13 shows the overall picture of the proposed timing control loop for spiral based SSW process. By using the noisy timing system model (Equation (7.6)) of the prewritten spiral tracks, it is easy to design a Kalman filter to optimally estimate the read head phase and frequency by minimizing the estimation error covariance, i.e.

$$\min E\left[\tilde{\phi}^2(k)+\tilde{f}^2(k)\right] \tag{7.9}$$

where $\tilde{\phi}(k)=\phi(k)-\hat{\phi}(k|k-1)$ and $\tilde{f}(k)=f(k)-\hat{f}(k|k-1)$ represent the state estimation error; $E[\ ]$ represents the expectation value.

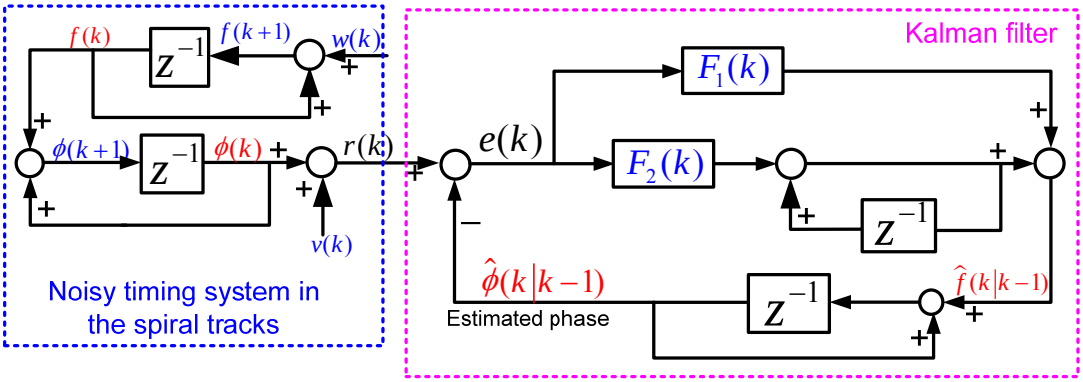

Figure 7.13: Proposed timing control loop based on Kalman filter

Equation (7.9) implies minimizing the phase error between the read head and write head subject to the non-uniform spiral location $t_s(k)$ which is caused by the repeatable timing error. Therefore, using the Kalman filter design, the write head clock is updated accurately. This is the advantage of the Kalman filter over the conventional PLL circuit. In conventional PLL circuit as shown in Figure 7.9, the PI loop filter is designed for uniform sampling system, so the control input $u(k)$ for NCO is inaccurate and the write head clock $T_w(k)$ is not the correct value.

Changing the reference clock

After the Kalman filter is designed, it is simple to deal with the other issue in the timing control loop: non-uniform sampling caused by the repeatable timing error. We propose to fix it by changing the reference clock in the system. In conventional PLL circuit as shown in Figure 7.9, the reference clock is $t(k)$, which is the ideal uniform timing mark locations with constant interval T_c. The read head phase $\phi(k)$ is defined in Equation (7.3) and the write head phase $\hat{\phi}(k)$ is defined as

$$\hat{\phi}(k) = \left(t_w(k) - t(k)\right) \cdot \frac{2\pi}{NT_c} \tag{7.9}$$

Since the PLL is a tracking control system, so the write head phase $\hat{\phi}(k)$ will follow the read head phase $\phi(k)$, i.e. the write head timing location $t_w(k)$ will approach to the spiral location $t_s(k)$, which is not desirable because the spiral location $t_s(k)$ is non-uniform due to the repeatable timing error caused by the drifts of spirals and sync marks' position error.

In our proposed compensation algorithm, we choose the spiral location $t_s(k)$ as the reference clock, as shown in Figure 7.14.

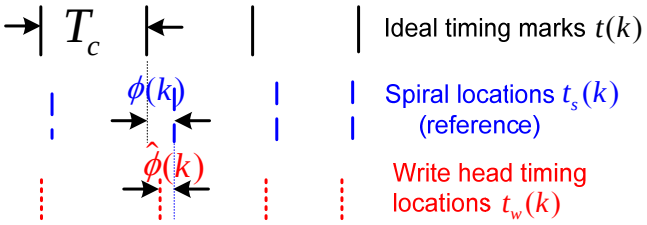

Figure 7.14: Timing locations of read/write heads with proposed reference clock

So now the phase $\phi(k)$ in the reference signal is rewritten as

$$\phi(k) = \left(t(k) - t_s(k)\right) \cdot \frac{2\pi}{NT_c} \tag{7.10}$$

And the write head phase $\hat{\phi}(k)$ is rewritten as

$$\hat{\phi}(k) = \left(t_w(k) - t_s(k)\right) \cdot \frac{2\pi}{NT_c} \tag{7.11}$$

Since the Kalman filter is an optimal estimator, hence it will achieve that: the write head phase $\hat{\phi}(k)$ optimally follows the reference phase $\phi(k)$, i.e. the write head timing location $t_w(k)$ approaches to the spiral location $t_s(k)$. As a result, the servo write clock in the servo system $t_w(k)$ converges to the nominal constant sampling period T_c. This achieves our objective: the product servo sectors are uniformly and accurately written in the circular data tracks.

Implementation of the proposed algorithm

From Figure 7.12, we know that the write head phase $\hat{\phi}(k)$ can be calculated when $e(k-1)$ is measured. But the actual spiral location $t_s(k)$ in equations (7.10)-(7.11) is not measurable. In other words, the servo write clock $t_w(k)$ is not available even $\hat{\phi}(k)$ is known. However, $\hat{\phi}(k)$ is required to be known at least one step advance for sampling the servo system at the sampling instant k. Notice that $\hat{\phi}(k)$ is generated by the NCO and it can be rewritten as the following equation based on the principle of its operation.

$$\hat{\phi}(k) = \left(t_w(k) - t_{cen}(k)\right) \cdot \frac{2\pi}{NT_c} \tag{7.12}$$

where $t_{cen}(k)$ is the center frequency of the NCO.

In order to generate the servo write clock $t_w(k)$, $t_{cen}(k)$ must be known. For implementation, we propose to calculate $t_{cen}(k)$ as

$$t_{cen}(k) = t(k) - \hat{\phi}(k) \frac{2\pi}{NT_c} \tag{7.13}$$

where $t(k)$ is the ideal timing mark location and is known for all k.

From Equation (7.12) and Equation (7.13), we have $t_w(k) = t(k)$ for all k, i.e. the NCO generates the uniform sampling period $T_c = t(k) - t(k-1)$ for the servo system.

In a summary, the proposed Kalman filter with changing the reference clock is able to deal with the non-uniform sampling issue in timing control loop and make sure the written product servo patterns are uniformly and accurately aligned in the circular tracks. In the following section, the proposed compensation algorithm is verified by the simulation studies.

7.5 Simulation Study

In this simulation study, the plant NCO model, PI loop filter, and nominal sampling period T_c are chosen the same ones as that used in Section 6.5. The same 19 tracks of repeatable timing error $\phi_{R\_i}(k)$ (as shown in Figure 6.13) and non-repeatable timing error $\phi_{NR\_i}(k)$ (as shown in Figure 6.14) are injected in the system. They are collected from a real disk drive and used for calculating the jitter covariance W and V, i.e. $\phi_{R\_i}(\bullet)$ is used to calculate the covariance W in the track i and $\phi_{NR\_i}(\bullet)$ is used to calculate the covariance V in the track i. Hence, in different tracks, the Kalman filter gains $F_1(k)$ and $F_2(k)$ have different steady-state values, as shown in Figure 7.15 and Figure 7.16. In the figures, each line represents the Kalman filter gains in one servo track, where the circles (‘○’) and asterisks (‘∗’) represent the gain at the servo sector k, i.e. $F_1(k)$ and $F_2(k)$, respectively.

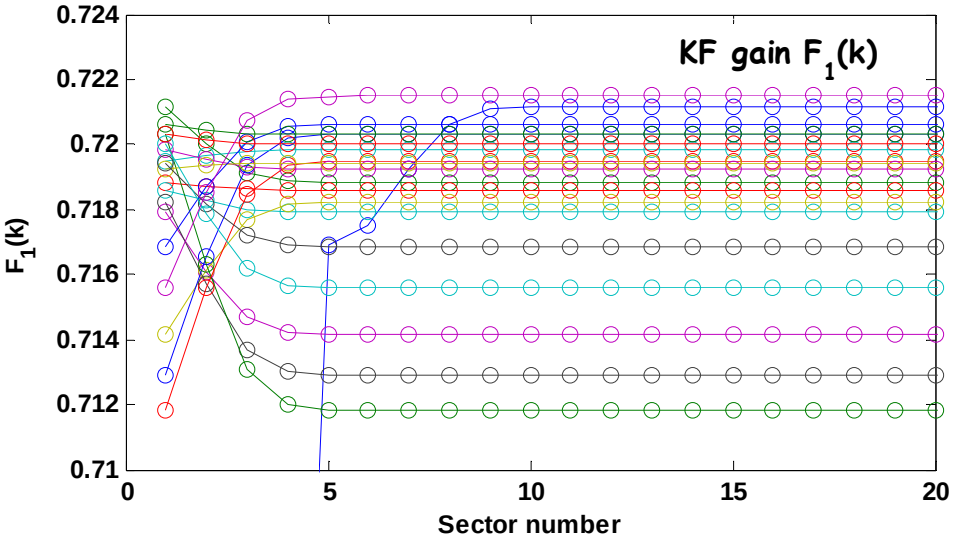

Figure 7.15: Kalman filter gain $F_1(k)$

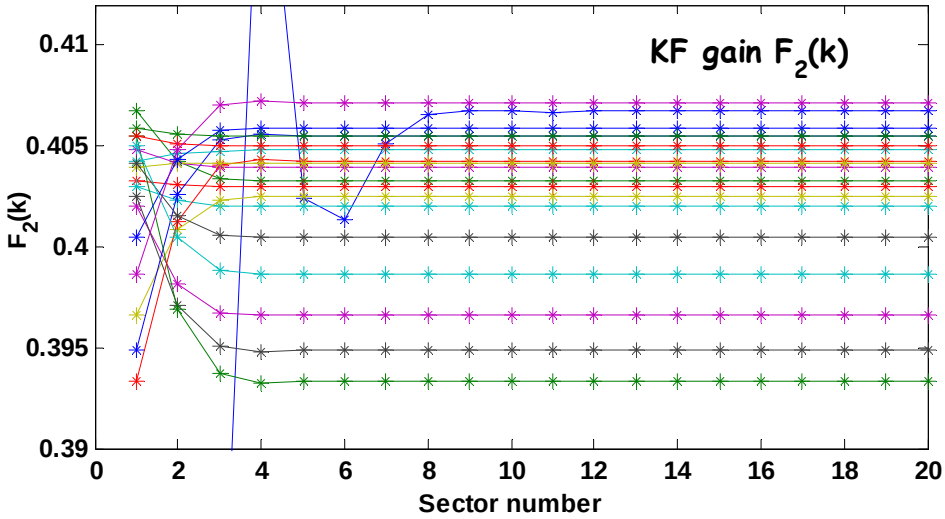

Figure 7.16: Kalman filter gain $F_2(k)$

From Figure 7.15 and Figure 7.16, we know that $F_1(k)$ and $F_2(k)$ are time-varying in each track but they quickly converge to constants within several sectors when the system is in steady state. We also note that in this design example, $F_1(k)$ and $F_2(k)$ have small variations between different tracks, so it is feasible to use the steady-state Kalman filter gains instead of the time-varying Kalman filter gains.

Figure 7.17 shows the phase error comparison between the original PLL circuit and the system using the proposed Kalman filter design. From the figure, we can see that using the Kalman filter design, the phase error in the system (as shown in the solid line with the average $\sigma[e_i] = 0.30T_s$) is reduced about 90% from the original PLL circuit (as shown in the dashed line with the average $\sigma[e_i] = 2.90T_s$). The performance of steady-state Kalman filter is also remarkable, as shown in the dotted line. This can be proved by the results in every single track.

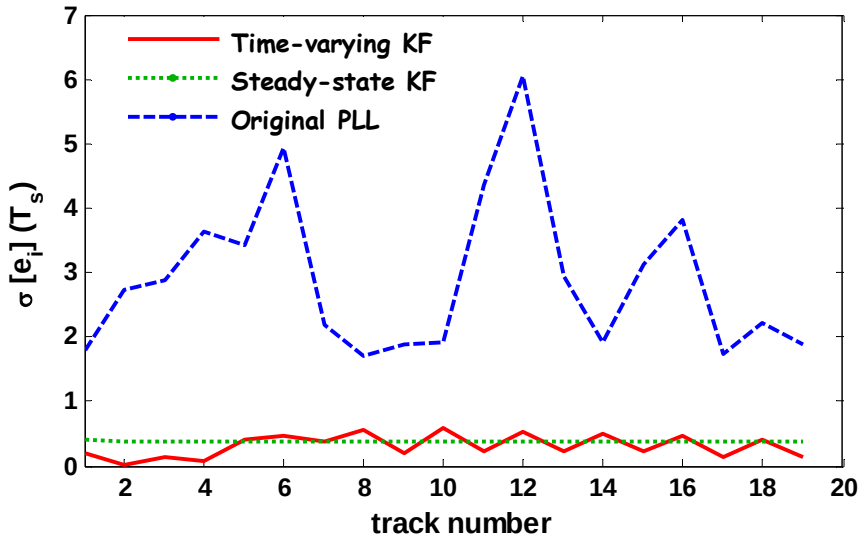

Figure 7.17: Phase error comparison between the original PLL circuit ($avg(\sigma[e_i]) = 0.30T_s$), time-varying Kalman filter ($avg(\sigma[e_i]) = 2.90T_s$), and steady-state Kalman filter

For example, Figure 7.18 shows the phase error comparison in the first track. Clearly, the phase error in both time-varying Kalman filter (as shown in the solid line) and steady-state Kalman filter (as shown in the dotted line) is much smaller than that in the original PLL circuit (as shown in the dashed line). However, we notice that the performance of the steady-state Kalman filter is poorer than the time-varying Kalman filter, as shown in Figure 7.19. The reason is that the steady-state Kalman filter gains are not close enough to the time-varying Kalman filter gains in this track. Let's see the results in another track. Figure 7.20 shows the results comparison in the 18^{th} track. Similarly, the phase error in both time-varying (as shown in the solid line) and steady-state (as shown in the dotted line) Kalman filters is greatly reduced from the original PLL circuit (as shown in the dashed line). Moreover, the performance of these two Kalman filters is almost the same, as shown in Figure 7.21. It's because their filter gains are close to each other in this track.

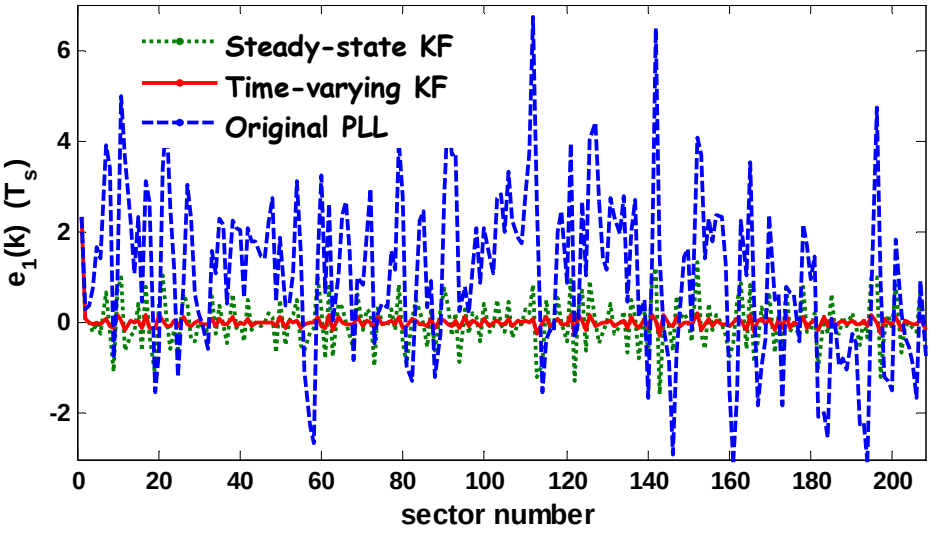

Figure 7.18: Phase error comparison in the 1st track

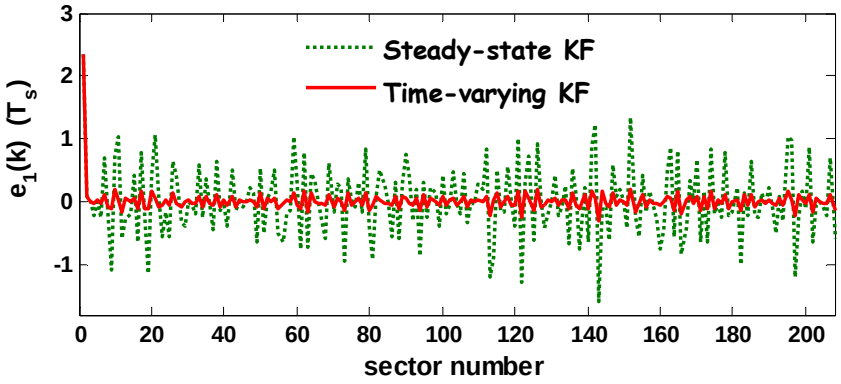

Figure 7.19: Phase error comparison in the 1st track

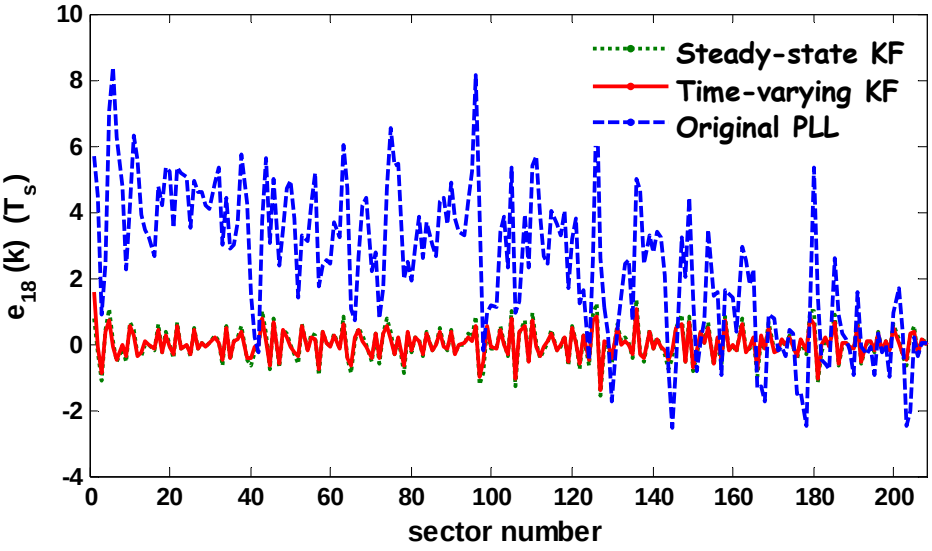

Figure 7.20: Phase error comparison in the 18th track

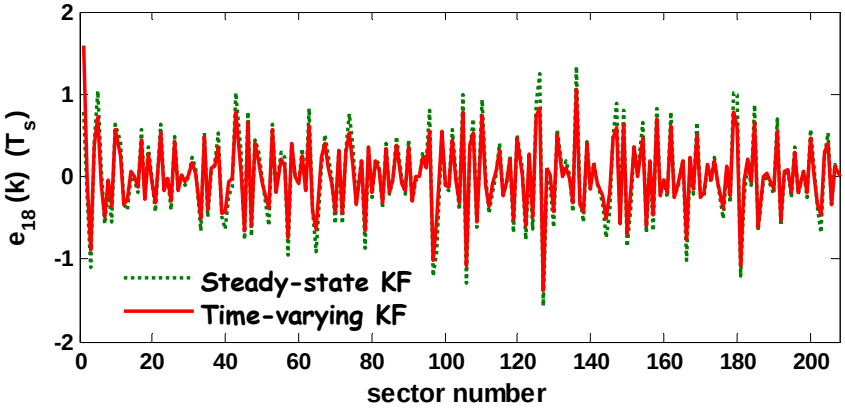

Figure 7.21: Phase error comparison in the 18th track

In order to further verify the effectiveness of the proposed compensation algorithm, the simulation studies are conducted by changing the system initial uncertainty and mechanical jitter. Table 7.1 shows the comparison between the results in Figure 7.17 and the results in the system with larger system initial uncertainty. It shows that the proposed Kalman filter is less sensitive to the system initial uncertainty.

Table 7.2 shows the comparison of the results between the systems with different mechanical jitter. From the results, it can be seen that the original PLL circuit is more sensitive to the mechanical jitter than the Kalman filter. Since the mechanical jitter is the main contributor of the repeatable timing error and it is more significant than the measurement jitter, hence it can be concluded that the proposed Kalman filter is more effective in dealing with the repeatable timing error than the conventional PLL circuit.

Table 7.1: Comparison of $\sigma[e_i]$ with different system initial uncertainty

	Original PLL	Time-varying KF
System with normal initial uncertainty	$2.90T_s$	$0.31T_s$
System with larger initial uncertainty	$6.10T_s$	$0.41T_s$

Table 7.2: Comparison of $\sigma[e_i]$ with different mechanical jitter

	Original PLL	Time-varying KF
System with normal mechanical jitter	$2.90T_s$	$0.31T_s$
System with smaller mechanical jitter	$2.90T_s$	$0.20T_s$

Notice that in Table 7.1 and Table 7.2, we choose the time-varying Kalman filter to conduct the comparisons in order to show the potential improvement of the proposed Kalman filter design. From the figures 7.17 – 7.21, we know that the performance of the steady-state Kalman filter is close to that of the time-varying Kalman filter. Moreover, using the constant Kalman filter gains to tune the PLL parameters is more practical and more easily implemented.

7.6 Chapter Summary

This chapter presented an alternative compensation algorithm for HDD spiral based SSW timing control loop, wherein the repeatable timing error dominates the system dynamics.

Analyzing the repeatable timing error from the computer system perspective, this repetitive process was viewed as a real-time or event-driven control system since the write head clock is updated only at the time instants when the sync mark is detected within the demodulation window by the read head. From this perspective, the repeatable timing error was interpreted as non-uniform sampling period of the servo system. To tackle with the non-uniform sampling issue, a novel control scheme based on Kalman filter theory was proposed to redesign the PLL circuit in the timing control loop. The proposed compensation algorithm is not only able to resolve the non-uniform sampling issue, but also can be implemented as simply as the conventional PLL circuit in the industry. By using the proposed scheme, the quality of the product servo patterns is improved up to 90%.

Chapter 8

Conclusion

This dissertation presented advanced control algorithms to deal with three main problems observed in discrete linear repetitive processes, and particularly in hard disk drive self-servowriting process.

Error propagation along the iteration direction

In a repetitive process, the iteration profile acts as a forcing function and contributes to the dynamics of the next iteration. As a result, the system error is propagated from iteration to iteration, which may cause the industrial process to fail without appropriate control actions.

Chapter 3 presented an iterative learning control scheme to deal with this problem and implemented it in the position control loop of HDD concentric SSW process. In this repetitive process, the system error is the only measurable signal. Hence, the recorded previous-track error information was intuitively utilized to construct the learning control law. The conditions of asymptotic convergence and monotonic convergence were derived and used to design the error learning filter. An L_1 optimal control problem was formulated in the sense of minimizing the maximum magnitude of the position deviation profile in each track. The proposed control scheme contained the track-to-track radial error propagation well and assured a good quality of the servowriting process.

An alternative control approach was proposed for this repetitive process in Chapter 4. Firstly, the system was equivalently represented by a 2D Roesser state-space model. The convergence problems in the iterative learning control system were then translated to the stability problems in a 2D control system. By using the 2D systems theory, a 2D state feedback controller was designed to guarantee asymptotic stability in both the time domain and the iteration domain. An error learning filter was further added later to mitigate the radial error propagation from track to track. The 2D control scheme was proved to have comparable performance to the iterative learning control scheme since it was able to guarantee the feedback control performance and the feedforward control performance simultaneously.

Error propagation along both the iteration and the time directions

In some discrete linear repetitive processes such as the timing control loop in HDD concentric SSW process, the error propagation occurs not only along the iteration direction but also over the time. This problem requires more advanced control method.

In Chapter 5, a novel adaptive feedforward control scheme was proposed to deal with the error propagation along both the iteration and the time directions. In this design, an adaptive filter was designed to estimate the circumferential timing error which, together with the radial timing error, was utilized to contain the closure error within each individual track and attenuate the timing error propagation from track to track. The filter coefficients were derived by using the filtered-x least mean square algorithms, which minimizes both the circumferential timing error energy and the radial timing error energy. Therefore, the filter coefficient at every servo sector is optimal with minimum closure error and minimum timing error propagation.

Repeatable error domination on the process dynamics

Another major contribution this dissertation made is the development of a framework for analyzing the repeatable error in discrete linear repetitive process, and the design of effective control algorithms to reduce its dominance on the system dynamics. A typical example of dominant repeatable error can be found in the timing control loop of HDD spiral based SSW process, where the repeatable timing error is amplified through the closed-loop response to generate the written timing profile.

Analyzing the problem from two different perspectives, two control schemes were derived for this repetitive process in Chapter 6 and Chapter 7, respectively. Using the classical control theory, Chapter 6 proposed a recursive least square based parameter adaptation algorithm to estimate and cancel the repeatable timing error. Examined from the computer system perspective in Chapter 7, however, this repetitive process is a real-time or event-driven control system and the repeatable timing error was interpreted as the sampling jitter in the system, which causes the sampling period of the servo system to be non-uniform. Furthermore, such sampling jitter is represented by two additive noise terms: mechanical jitter and noise jitter. The covariance of the jitter terms were provided to apply the Kalman filter theory and determine the PLL parameters C_1 and C_2 as the Kalman filter gains F_1 and F_2. The statistical properties of jitters can be estimated if the spiral tracks are written by the same spiral writer and the SSW is conducted in the same tester.

Bibliography

[1] D. Abramovitch. Phase-locked loop: a control centric tutorial. *Proc. of ACC*, 2002.

[2] H.S. Ahn, C.H. Choi, and K.B. Kim. Iterative learning control for a class of nonlinear systems. *Automatica*, 19(6):1575-1578, 1993.

[3] P. Albertos and A. Crespo. Real-time control of non-uniformly sampled systems. *Control Engineering Practice*, 7(4), 445-458. 1999.

[4] P. Apkarian and R. J. Adams. Advanced gain-scheduling techniques for uncertain systems. *IEEE Trans. on Control Systems Technology*, 6(1), 21-32.

[5] K.J. Åström and B. Wittenmark, *Computer controlled systems*. Second edition. Prentice Hall. 1990.

[6] N. Amann, D.H. Owens, and E. Rogers. Robustness of norm-optimal iterative learning control. *In Proceedings of International Conference on Control*, volume 2, pages 1119-1124, 1996.

[7] N. Amann, D.H. Owens, and E. Rogers. 2D systems theory applied to learning control systems. *In Proc. of the 33rd Conf. on Decision and Control*, pages 985-986, 1994.

[8] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *J. of Robotic Systems*, 1(2):123-140, 1984.

[9] D. A. Bristow, M. Tharayil, and A.G. Alleyne. A survey of iterative learning control. *Control Systems Magazine, IEEE*, 26(3):96-114, June 2006.

[10] D. A. Bristow and A.G. Alleyne. Monotonic convergence of iterative learning control for uncertain systems using a time-varying q-filter. *Proceedings of the American Controls Conference*, 2005.

[11] J. C. Burgess. Active adaptive sound control in a duct: a computer simulation. *J. Acoust. Soc. Amer.*, vol. 70, pp. 715-726, Sep., 1981.

[12] R. Batty. *Notch filter control of magnetic bearings to improve rotor synchronous response.* S. M. Thesis. M.I.T., Cambridge, MA, 1988.

[13] S. Boyd, L. Gaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in systems and control theory*. Philadelphia, PA: SIAM.

[14] C. F. N. Cowan and P. M. Grant. *Adaptive filters.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[15] Y. Chen, J.-X. Xu, T. H. Lee, and S. Yamamoto. An iterative learning control in rapid thermal processing. *In Proc. the IASTED Int. Conf. on Modeling, Simulation and Optimization (MSO'97)*, pages 189-192, Singapore, August 1997.

[16] B. G. Dijkstra and O.H.Bosgra. Exploiting iterative learning control for input shaping, application to a waferstage. *Proceedings of the American Control Conference*, 2003.

[17] C. A. Desoer and M. Vidyasagar. *Feedback systems: input-output properties.* Academic Press, Inc., New York, NY, 1975.

[18] C. Du, L. Xie, and C. Zhang. H_∞ control and robust stabilization of two-dimensional systems in Roesser models. *Automatica,* 37, 205-211, 2001.

.

[19] C. Du and L. Xie. LMI approach to output feedback stabilization of 2-D discrete systems. *Int. J. Control*. 72 (2), 97-106, 1997.

[20] F. Dong and M. Tomizuka. Timing error compensator design for spiral based self-servowriting in disk drives. *Proceedings of IFAC congress*, 2010.

[21] F. Dong and M. Tomizuka. Timing error compensator design for self-sevowriting system in hard disk drives. *Proceedings of DSC Conference*, Oct. 2009.

[22] F. Dong and M. Tomizuka. An iterative learning control design for self-sevowriting system in hard disk drives using L_1 optimal control. *Proceedings of America Control Conference*, Jun. 2009.

[23] J. B. Edwards and D. H. Owens. Analysis and control of multipass processes. *Research Studies Press*, 1982.

[24] S. J. Elliott and P. A. Nelson. The applications of adaptive filtering to the active control of sound and vibration. *ISVR, Univ. Southampton, U.K., Tech. Rep.* 136, Sep. 1985.

[25] B. A. Francis and W. M. Wonham. The internal model principlw for linear multivariable regulators. *Applied Mathematics and Optimization*, 2(2);170-194, July 2005.

[26] J. A. Frueh and M. Phan. Linear quadratic optimal control (lql). *Int. Journal of Control,* 73(10):832-839, 1999.

[27] K. Furuta and M. Yamakita. The design of a learning control system for multivariable systems. *In Proc. of IEEE Int. Symp. On Intelligent Control,* pages 371-376, 1987.

[28] G. C. Goodwin and K. S. Sin. *Adaptive filtering prediction and control.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[29] J. R. Glover. Adaptive noise cancelling applied to sinusoidal interferences. *IEEE Trans. on Acoustics, Speech, Signal Processing,* 25:464-491. 1977.

[30] K. Galkowski, E. Rogers, D. H. Owens. New 2D models and a transition matrix for discrete linear repetitive processes. *Int. J. Control,* 72(15):1365-1380, 1999.

[31] K. Galkowski, E. Rogers, and D. H. Owens. New 2D models and a transition matrix for discrete linear repetitive processes. *Int. J. Control.* 72 (15), 1365-1380, 1999.

[32] K. Galkowski, E. Rogers, S. Xu, J. Lam, and D. H. Owens. LMIs - a fundamental tool in anaysis and controller design for discrete linear repetitive processes. *Trans. Circuits Syst.-I: Fundam. Theory Appl.*, 49 (6), 768-778. 2002.

[33] K. Galkowski, W. Paszke, E. Rogers, S. Xu, J. Lam, and D. H. Owens. Stability and control of differential linear repetitive processes using an LMI Setting. *IEEE Trans. Circuits Syst.-II: Analog Digital Signal Process,* 50 (9), 662-666, 2003.

[34] K. Galkowski, W. Paszke, B. Sulikowski, E. Rogers, S. Xu, J. Lam, and D. H. Owens. Stability and control of a physical class of 2D continuous-discrete linear systems using an LMI setting. *Proceedings of the American Control Conference, Denver, Colorado*, pp 5058-5063, 2003.

[35] X.Guan, C. Long, and G. Duan. Robust optimal guaranteed cost control for 2D discrete systems. *IEE Proc. Control Theory Appl*. 148 (5), 355-361, 2001.

[36] Z. Geng, R. Carroll, and J. Xie. Two-dimensional model and algorithm analysis for a class of iterative learning control systems. *Int. J. Control*, 52(4), 833-862, 1990.

[37] H. Hjalmarsson, S. Gunnarsson, and M. Gevers. A convergent iterative restricted complexity control design scheme, *Proceedings of the 33rd IEEE Conference on Decision and Control*, volume 2, pages: 1735-1740, Dec. 1994.

[38] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin. Iterative feedback tuning: theory and applications. *IEEE Control Systems Magazine*, 18(4):26-41, Aug. 1998.

[39] H. Hjalmarsson. Iterative feedback tuning – an overview. *Int. J. Adapt. Control Signal Process*, 16:373-395, 2002.

[40] H. Havlicsek and A. Alleyne. Nonlinear control of an electro-hydraulic injection molding machine via iterative adaptive learning. *IEEE/ASME Trans. On Mechatronics,* 4(3):312323, 1999.

[41] J. Hu and M. Tomizuka. A new plug-in adaptive controller for rejection of periodic disturbances. *J. of Dynamic Systems*, 115:543-546, 1993.

[42] M. L. Honig and D. G. Messerschmitt. *Adaptive filters: structures, algorithms, and applications*. Boston, MA: Kluwer, 1986.

[43] T. Hinamoto. 2-D Lyapunov equation and filter design based on the Fornasini-Marchesini second model. *IEEE Trans. Circuits Syst.-I: Fundam. Theory Appl.,* 40 (2), 102-110, 1993.

[44] T. Hinamoto. Stability of 2-D discrete systems described by the Fornasini-Marchesini second model. *IEEE Trans. Circuits Syst.-I: Fundam. Theory Appl.*, 44 (3), 254-257, 1997.

[45] T. S. Huang. *Two-dimensional digital signal processing.* Springer-Verlag: Berlin, New York, 1981.

[46] P. A. Ioannou and J. Sun. *Robust adaptive control.* Prentice Hall, Upper Saddle River, NJ, 1996.

[47] A. Kanellakis. New stability results for 2-D discrete systems based on the Fornasini-Marchesini state space model. *IEE Proc. Circuits, Devices Syst.*, 141 (5), 427-432. 1994.

[48] C. Kempf, W. C. Messner, M. Tomizuka, and R. Horowitz. Comparison of four discrete-time repetitive control algorithms. *Contorl Systems magazine, IEEE*, 13(6):48-54, Dec. 1993.

[49] C. R. Knopse. *Reducing unbalance response with magnetic bearings.* Internal Report of the Center of Magnetic Bearings. University of Virginia, 1992.

[50] Dong-Il Kim and Sungkwun Kim. An iterative learning control method with application for CNC machine tools. *IEEE Transactions on Industry Applications*, 32(1):66-72, January-February 1996.

[51] J. E. Kurek and M. B. Zaremba, Iterative learning control synthesis based on 2-D system theory. *IEEE Trans. Autom. Control*, 38 (1), 121-125, 1993.

[52] T. Kaczorek. *Two-dimensional linear systems.* Springer: Berlin, 1985.

[53] J. Lam, S. Xu, Y. Zou, Z. Lin, and K. Galkowski. Robust Output feedback stabilization for two-dimensional continuous systems in Roesser form. *Appl. Math. Lett.*, 17, 1331-1341, 2004.

[54] K. L.Moore, Y. Q. Chen, and V. Bahl. Intermittent iterative learning control. *Automatica,* 41(9):1529-1537, 2004.

[55] W.-S. Lu. Some new results on stability and stability robustness of Fornasini-Marchesini state-space 2-D digital filters. *Proceedings of IEEE Southeastcon'94. Creative Technology Transfer-A Global Affair*, 21-25, 1994.

[56] W.-S. Lu and A. Antoniou. *Two-dimensional digital filters.* Marcel Dekker: New York, Basel, Hong Kong, 1992.

[57] W.-S. Lu. Stability analysis for two-dimensionalsystems. *IEEE Trans. On Circuits and Systems*, vol. cas-30, no. 7, July, 1983.

[58] Y. M. Lifchifts, W. Ying, Y. Cai, and S. Weerasooriya, Servo writing a disk drive by synchronizing a servo write clock to a reference pattern on the disk and compensating for repeatable phase error", *US Patent, 7,333,280.*

[59] D. R. Morgan. An analysis of multiple correlation cancellation loops with a filter in the auxiliary path. *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-28, pp. 454-467, Aug. 1980.

[60] K. L. Moore, M. Dahleh, and S. P. Bhattacharyya. Learning control for robotics. *In Proceedings of International Conference on Communications and Control*, pages 976-987, Baton Rouge, Louisiana, October 1988.

[61] H. Melkote and R. McNab. A repetitive process approach to modeling and control for 'bootstrap' self-servowriting in hard disk drives. *Int. J. Adapt. Control Signal Process.,* 22:402-412, 2008.

[62] H. Melkote, Z. Wang, and R. McNab. An iterative learning controller for reduction of repetitive runout in disk drives. *IEEE Trans. On Control Systems Technology*, 14:467-473, 2006.

[63] D. H. Owens. Stability of multipass processes. *Proceedings of The institution of Electrical Engineers*, 124(11), pp. 1079-1082, 1977.

[64] D. H. Oh, J. C. Koo, and S. M. Suh. A robust adaptive feedforward method for the compensation of harmonic disturbances. *Microsyst. Technol.,* 13:1261-1269, 2007.

[65] J. B. Pearson and M. A. Dahleh. L_1-optimal feedback controller for MIMO discrete-time systems. IEEE Trans. Automat. Contr., vol. 32, no. 4, pp. 314-322, 1987.

[66] E. Rogers and D. H. Owens. 2D systems theory and applications – a maturing area. *Int. Conf. Control,* 1: 63-69, 1994.

[67] E. Rogers and D. H. Owens. *Stability analysis for linear repetitive processes*. Springer-Verlag: Berlin, Heidelberg, 1992.

[68] E. Rogers and D. H. Owens. *Stability analysis for linear repetitive processes.* Lecture Notes in Control and Information Sciences Series, vol. 175, Springer-Verlag, Berlin, 1992.

[69] E. Rogers and D. H. Owens. Output feedback control of discrete linear repetitive processes. *IMA J. of Mathematical Control & Information,* 10:177-193, 1993.

[70] A. Sacks, M. Bodson, and W. Messner, Adaptive methods for repeatable runout compensation, *IEEE Trans. On Magn.*, vol. 31, no.2, 1031-1036, 1995.

[71] A. Sacks, M. Bodson, and P. Khosla. Experimental results of adaptive periodic disturbance cancellation in a high performance magnetic disk drive. *Proceedings of the American Control Conference,* 1993.

[72] B. Shafai, S. Beale, P. Larocca, and E. Cuccon. Magnetic bearing control systems and adaptive forced balancing. *IEEE Control Systems*, 14:4-13, 1994.

[73] J. Shi, F. Gao, and T.-J. Wu. Robust design of integrated feedback and iterative learning control of a batch process based on a 2D Roesser system. *J. Process Control,* 15, 907-924, 2005.

[74] M. D. Schultz, E. J. Yarmchuk, B. C. Webb, and T. J. Chainer. A self-servowrite clocking process. *IEEE Trans. Magn.*, vol.37, no.4, pp. 1878-1880, Jul. 2001.

[75] M. Šebek and F. J. Kraus. Stochastic LQ-optimal control for 2-D systems. *Multidimensional Syst. Signal Process*. 6, 275-285, 1995.

[76] S. K. Tso and X. Ma. Discrete learning control for robots: strategy, convergence and robustness. *International Journal of Control*, 57(2):273-291, 1993.

[77] M. Uchiyama. Formulation of high-speed motion pattern of a mechanical arm by trial. *Trans. SICE (Soc. Instrum. Contr.Eng.)*, 14(6):706-712 (in Japanese), 1978.

[78] M. Vidyasagar, *Nonlinear system analysis*. Second edition. Prentice Hall. Englewood Cliffs. NJ. 1993.

[79] M. Vidyasagar, Optimal rejection of persistent bounded disturbances. *IEEE Trans. Automat. Contr.,* vol. AC-31, pp.527-534, June, 1986.

[80] B. Widrow and S. D. Stearns. *Adaptive signal processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[81] B. Widrow, D. Shur, and S. Shaffer. On adaptive inverse control. *Proc. 15th Asilomar Conf.,* pp. 185-189, 1981.

[82] D. Wang, B. Zhang, and Y. Ye. Wavelet transform-based frequency tuning iterative learning control. *IEEE Trans. on Sys. Man and Cybernetics,* 35(1):107-114, 2005.

[83] S. C. Wu and M. Tomizuka. New schemes for repeatable runout compensation using adaptive feedforward cancellation. *Proceedings of the American Control Conference,* Jul. 2007.

[84] S. C. Wu and M. Tomizuka. Repeatable runout compensation for hard disk drives using adaptive feedforward cancellation. *Proceedings of the American Control Conference*, Jun. 2006.

[85] S. Wu and M. Tomizuka. An iterative learning control design for self-servowriting in hard disk drives. *Mechatronics*. 20(1):53-58, 2009.

[86] L. Xie, C. Du, Y. C. Soh, and C. Zhang. H_∞ and robust control of 2-D systems in FM second model. *Multidimensional Syst. Signal Process*. 13, 265-287, 2002.

[87] http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php

[88] http://mizugaki.iis.u-tokyo.ac.jp/nss/