Enhanced Iterative Learning Control with Applications to A Wafer Scanner System

by

Shu-Wen Yu

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 J. Karl Hedrick
Professor Roberto Horowitz
Professor Laurent El Ghaoui

Spring 2011
Enhanced Iterative Learning Control with Applications to A Wafer Scanner System

Copyright 2011
by
Shu-Wen Yu
Abstract

Enhanced Iterative Learning Control with Applications to A Wafer Scanner System

by

Shu-Wen Yu

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

This thesis addresses the improvement of wafer scanner technology from the controls aspect. In particular, the thesis proposes to enhance basic Iterative Learning Control (ILC) in several ways. ILC is a feedforward control strategy used to improve the performance of a system that performs a repeated task by considering the error from previous iterations of the process. The existence of non-repeating events, however, degrades the performance of ILC. Furthermore, the selection of the Q-filter and learning function in the iterative control law also limits the performance of ILC. To relax these limitations, we first introduce a disturbance observer into the learning scheme. As a result, we are able to reduce the effect of non-repetitive disturbances on the ILC scheme. The combination of ILC and DOB information when performing one task can also be used to provide valuable information for selecting initial ILC effort for a different task. By doing so, we can improve the convergence rate of the ILC algorithm for the new task. Studies performed in this thesis will show that by properly selecting the initial ILC effort, one can reduce the number of iterations before ILC achieves high tracking accuracy.

In addition to the ILC effort, we also apply a pre-designed optimal feedforward control input to minimize the initial tracking error. An optimization strategy is presented to obtain this optimal feedforward input as well as the optimal feedback controller for integration with the ILC scheme. By using this optimal feedback-feedforward controller combination, which replaces the standard PID feedback controller, it becomes possible to utilize a simple P-type ILC algorithm without compromising performance.

This thesis also investigates the synchronization problem to ensure small alignment mismatch between the wafer stage and reticle stage. The challenge is to meet tracking requirements of each individual stage as well as maintaining the relative positioning between the two stages in the presence of cross-coupling dynamics. Therefore, to achieve the ultrahigh precision motion control requirements demanded for such scanners, a synchronization ILC algorithm is proposed and designed to reduce the synchronization error while minimizes the individual tracking error.
To my family and friends, for supporting me all these years.
# Contents

List of Figures iv

List of Tables vii

## 1 Introduction

1.1 Control Objectives for Photolithography .......................... 1
1.2 Experimental Setup of the Prototype Wafer Scanner System ........ 3
1.3 System Modeling and Description ....................................... 5
1.4 Overview of Iterative Learning Control ............................... 6
1.5 Thesis Contributions .................................................. 7
1.6 Outline of Thesis .................................................... 8

## 2 Iterative Learning Control

2.1 Iterative Learning Control ............................................. 10
2.2 Stability and Performance Analysis of ILC Algorithms .............. 11
2.2.1 Stability Analysis ................................................ 11
2.2.2 Performance Analysis ........................................... 12
2.3 Different ILC Design Methods ........................................ 13
2.3.1 P-type ILC ..................................................... 14
2.3.2 Higher-Order ILC ............................................... 14
2.3.3 Plant-Inversion Based ILC ..................................... 15
2.3.4 Optimization Based ILC ....................................... 16
2.4 Chapter Summary .................................................... 17

## 3 Non-Repetitive Disturbance and Initial Control Effort Selection in ILC

3.1 Disturbance Observer Design ......................................... 20
3.1.1 Continuous-time Disturbance Observer ......................... 20
3.1.2 Unmodeled Dynamics Effects and Bandwidth Limits .......... 22
3.2 ILC in Combination with DOB ........................................ 25
3.2.1 Results ..................................................... 28
3.3 Selection of Initial ILC Effort ....................................... 30
3.4 Chapter Summary ...................................................... 35

4 Iterative Learning Control with Optimal Feedback and Feedforward Control ..................................................... 38
  4.1 Model Predictive Control .............................................. 39
  4.2 ILC Combined with The Optimal Feedback-Feedforward Control System .................................................... 41
  4.3 A Systematic Design Procedure ....................................... 43
  4.4 Comparison of Different ILC Algorithms ............................ 48
  4.5 Chapter Summary ...................................................... 51

5 ILC Design for Synchronization ........................................ 53
  5.1 Introduction .......................................................... 53
  5.2 Synchronization Control Scheme ...................................... 54
    5.2.1 Cross-Coupled Control ........................................... 54
    5.2.2 Master-Slave Control .......................................... 56
    5.2.3 Results .......................................................... 57
  5.3 ILC Design for Synchronization ........................................ 61
    5.3.1 The Synchronization ILC Scheme ................................ 64
    5.3.2 Experiment Results ............................................ 66
  5.4 Chapter Summary ...................................................... 70

6 Conclusions .................................................................. 71
  6.1 Summary and Conclusions ............................................. 71
  6.2 Further Research ...................................................... 72

Bibliography ................................................................ 74
List of Figures

1.1 Schematic of a wafer scanner .................................................. 2
1.2 The stages and the laser interferometry measurement system ................. 3
1.3 The block diagram of overall hardware and software interface ................. 4
1.4 Block diagram of closed loop system with standard ILC .......................... 6

3.1 Continuous-time disturbance observer ........................................... 21
3.2 Block diagrams of (a) the discrete-time disturbance observer and (b) the equivalent
discrete-time system .............................................................. 23
3.3 $Q_D$ filter bandwidth limitations ................................................ 24
3.4 Overall control block diagram ..................................................... 24
3.5 Comparison on the performance in terms of disturbance rejection with and without
disturbance observer .............................................................. 25
3.6 Comparison on the performance in terms of command following with and without
disturbance observer .............................................................. 26
3.7 A Comparison on stability robustness (The ILC system is stable if the magnitude
is less than one for all frequencies) .............................................. 28
3.8 Performance comparison of ILC alone and ILC in combination with DOB with
respect to (a) two norm of tracking errors over iterations and (b) tracking error in
the 45th iteration when only repetitive disturbances enter into the system .......... 29
3.9 Performance comparison of ILC alone and ILC in combination with DOB with
respect to (a) two norm of tracking errors over iterations (b) tracking error in the
45th iteration when both repetitive and non-repetitive disturbances enter into the
system .......................................................... 30
3.10 Experimental results for comparison of tracking error in the closed-loop system
with and without DOB .................................................. 31
3.11 Experimental results for performance comparison of ILC alone and ILC in combi-
nation with DOB with respect to (a) two norm of tracking errors over iterations
and (b) tracking error in the 7th iteration ........................................ 31
3.12 Plot of the 1st reference (a) position, (b) velocity, and (c) acceleration against time
index .......................................................... 33
3.13 Plot of the 2\textsuperscript{nd} reference (a) position, (b) velocity, and (c) acceleration against time index .......................................................... 34
3.14 Two-norm of the errors over iterations for tracking the 2\textsuperscript{nd} trajectory .......................................................... 35
3.15 Errors in the 1\textsuperscript{st} iteration for tracking the 2\textsuperscript{nd} trajectory. (a) The tracking error against time index and (b) zoomed-in detail of the error plot ................................. 36
3.16 Errors in the 45\textsuperscript{th} iteration for tracking the 2\textsuperscript{nd} trajectory. (a) The tracking error against time index and (b) zoomed-in detail of the error plot ................................. 36
3.17 Plot of (a) the learned control effort and (b) zoomed-in detail in the 45\textsuperscript{th} iteration. .......................................................... 37

4.1 Block diagram of ILC in combination with the optimal feedback-feedforward control 43
4.2 Comparison on stability robustness for (a) Case A and (b) Case B with different learning gains (The ILC system is stable if the magnitude is under 0 db for all frequencies) .............................................................. 44
4.3 Performance comparison in terms of disturbance rejection ($\|T_d(z)\|/\|1-Q(z)[1-L(z)]T_d(z)\|$) with $Q(z)=1$ and $L(z) = \alpha z^n$) for (a) Case A and (b) Case B .............................................................. 44
4.4 Maximum-norm error over iterations with different cutoff frequencies of Q-filters .............................................................. 46
4.5 Performance comparison of P-type ILC with and without optimal feedforward control in terms of tracking errors in the 25\textsuperscript{th} iteration .............................................................. 47
4.6 Maximum-norm errors over iterations in three different feedforward strategies .............................................................. 48
4.7 Performance comparison of different types of ILC algorithms in combination with (a) PID feedback control and (b) with optimal feedback-feedforward control in terms of the maximum-norm error. Assume the applied system model to have 10\% parameter uncertainty. .............................................................. 50
4.8 Performance comparison of different types of ILC algorithms in combination with (a) PID feedback control and (b) with optimal feedback-feedforward control in terms of two norm of the tracking error. Assume the applied system model to have 10\% parameter uncertainty. .............................................................. 51
4.9 Performance comparison of different types of ILC algorithms in combination with (a) PID feedback control and (b) with optimal feedback-feedforward control in terms of two norm of the tracking error. Assume the applied system to have multiplicative modeling uncertainty. .............................................................. 52

5.1 The structure of the cross-coupled control scheme .......................................................... 55
5.2 Illustration of tracking error and the synchronization error .......................................................... 56
5.3 The structure of the master-slave control scheme .......................................................... 57
5.4 System model of the wafer and reticle stages .......................................................... 57
5.5 Plots of (a) desired trajectories and (b) velocity profiles .......................................................... 58
5.6 Comparison of individual axes tracking errors of (a) wafer stage and (b) reticle stage, when two stages have identical dynamics .......................................................... 59
5.7 Comparison of synchronization error when two stages have identical dynamics .................. 59
5.8 Comparison of individual axes tracking errors of (a) wafer stage and (b) reticle stage, when two stages have different dynamics ........................................ 60
5.9 Comparison of synchronization error when two stages have different dynamics ........................................ 60
5.10 Experiment results for the comparison of individual axes tracking errors of (a) wafer stage and (b) reticle stage ........................................ 62
5.11 Experiment results for the comparison of synchronization error ........................................ 62
5.12 Zoomed-in detail of the synchronization error plot ........................................ 63
5.13 Synchronization error under CCC with different weighting values ........................................ 63
5.14 Illustration of the virtual target and synchronization error ........................................ 65
5.15 Performance comparison with different $\alpha$ values. Experiment results of the wafer stage for (a) two norm of tracking error over iterations, and (b) trajectory tracking error in the 9th iteration ........................................ 67
5.16 Performance comparison with different $\alpha$ values. Experiment results of the reticle stage for (a) two norm of tracking error over iterations, and (b) trajectory tracking error in the 9th iteration ........................................ 67
5.17 Performance comparison with different $\alpha$ values. Experiment results for (a) two norm of synchronization error over iterations, and (b) synchronization error in the 9th iteration ........................................ 68
5.18 Performance comparison with iteration-varying and time-varying $\alpha$. Experiment results for (a) two norm of synchronization error over iterations, and (b) synchronization error in the 9th iteration ........................................ 68
5.19 Performance comparison with iteration-varying and time-varying $\alpha$. Experiment results for two norm of tracking errors for (a) the wafer stage and (b) the reticle stage over iterations ........................................ 69
List of Tables

4.1 Systematic design procedure for a P-type ILC combined with the optimal feedback-feedforward controllers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
Acknowledgments

Firstly I would like to express my sincere gratitude to my advisor, Professor Tomizuka, for his guidance and expertise, for his insistence on the highest standards, and especially for his invaluable input into this thesis. I am also grateful to the rest of my thesis committee: Professor Horowitz, Professor Hedrick, and Professor El Ghaoui for reading and giving me feedback on the manuscript, these comments have improved the final version considerably.

All the members in the Mechanical Systems Control Laboratory are gratefully acknowledged: Yuki, Sugi, Chun-Chih, Shashi, Ahmed, Soo, Kiyo, Takashi, KC, Lucy, Steve, Haifei, Joobumn, Nancy, Wenjie, Evan, Mike, Max, Oak, Sanggyum, Chi-Shen, and all others. The people at MSC have provided me with a new ”family” at Berkeley. A special thank you to the Precision Motion Control Group: Sandipan, Ben, Hoday, and Will. It was really exciting working with you all and I look forward to continued collaboration with you. I am grateful to all of you.

This research could not have been possible without the financial support of Nikon Research Corporation of America (NRCA) and UC Discovery, which are gratefully acknowledged. Hardware and software support from National Instruments was also greatly appreciated.

My time at Berkeley was made enjoyable in large part due to many friends and groups that became a part of my life. I am grateful to my roommates and friends for their constant help and support through the thesis.

Lastly, I would like to thank my family, my Mother and my Father, my sister Amy and her husband Andrew. You have always taken cared of me. I especially want to thank Shu-Yun for all the love, support, and encouragement he has given me all these years.
Chapter 1

Introduction

The aim of this thesis is to achieve high precision and high throughput in the photolithography process by integrating Iterative Learning Control (ILC) into the wafer and reticle stages system. Several enhancements of the performance of basic ILC systems are proposed. High-precision synchronization and control of the wafer and reticle scanners are crucial for the manufacturing semiconductors. Higher throughput will increase the chip yield and as consequently decrease the per-unit cost. These two goals may be achieved by investigating advanced control strategies that can allow a system to meet stringent positioning accuracy, speed, and residual vibrations requirements. To accomplish these objectives, this thesis will focus on three main ideas: ILC algorithms, disturbance rejection methods, and synchronization control methods between the reticle and wafer stages.

1.1 Control Objectives for Photolithography

Integrated circuits (ICs) have become vital and ubiquitous in the modern world. IC manufacturing is a complex multiple step process. One step in the process is the printing of patterns onto a wafer by exposing it to a light source and is commonly known as photolithography. These patterns, however, usually have nanometer resolution; hence the apparatus needs to have high precision and accuracy. A photolithographic apparatus contains three functional parts: the reticle stage, lens system, and wafer stage (shown in Figure 1.1). The reticle stage holds a mask which contains the image that will be transferred to the wafer. The lens structure is housed between the reticle and wafer stage to reduce the image size. A light source transfers the geometric pattern from the mask, through the lens reducer onto the silicon wafer which is mounted on the wafer stage. The exposed material can then be etched off, leaving a circuit pattern on the wafer. The workhorse of the photolithography process is a machine called a wafer scanner, whose task is to accurately position and align components while the wafer is being exposed. Accurate pattern transfer involves precise positioning and alignment of the wafer and reticle stages.

With the rapid advances in IC technology, next generation IC designs may be considerably
Figure 1.1: Schematic of a wafer scanner

more complex and densely packed. Consequently, the next generation lithography mechanisms are also expected to operate under higher performance requirements. These performance requirements directly impact critical dimension variability (CD) and overlay accuracy. State-of-the-art scanning technology in the semiconductor manufacturing industry requires CD 3-sigma to be between 2-5nm and overlay accuracies to be within 3nm [1]. CD variability is influenced by a number of processes including exposure, the post-exposure bake, developing, etching, and chemical mechanical planarization (CMP). The exposure-related component of CD variability and overlay accuracy depends on accurately synchronizing and controlling the wafer and reticle stages during the scan phase. The synchronization of both stages is vital for reducing the alignment errors. Not only controllers for individual stages but also synchronization controllers for coordinating both stages need to be designed such that drive each stage follow the desired trajectory and at the same time reduce the synchronization error.

Motion control of scanners and steppers play an essential role in enabling high-throughput pattern transfer. The possibilities for control architectures become more complex when synchronization of multiple stages is considered. This study will involve both the theoretical development of control system design as well as the experimental verification of control methodologies that assure high speed and high accuracy motion control for the wafer and reticle stage. These methodologies will be developed based on generic wafer scanner models and then tailored to the specific application to the Nikon wafer scanner series. Realistic performance and robustness specifications
will be derived from Nikon’s long-standing expertise in wafer scanner design and control.

Our control strategies will be validated by using a reticle and wafer stage prototype located in the Mechanical Systems Control Laboratory at the University of California, Berkeley.

### 1.2 Experimental Setup of the Prototype Wafer Scanner System

The prototype wafer scanner system has two stages that are vertically stacked. This two level structure is commonly used in photolithography for transferring the image pattern from the mask (reticle stage) to the surface of a silicon wafer (wafer stage). This set up has been developed jointly with the project sponsor, Nikon Research Center of America (NRCA).

Figure 1.2 shows the setup of the wafer scanner system and the laser interferometry measurement system. Each level of the structure consists of a stage and a countermass both driven by current-controlled three-phase linear motors. The stage is supported by air bearings, and countermass is guided by roller bearing. The stage and the countermass are both controlled by their own separate amplifiers, hence allowing for the control of the countermass to balance the forces exerted by the stage. This prototype wafer scanner system will be used to test and verify our proposed algorithms and methodologies presented in this dissertation.

The laser interferometer measurement system is used within the feedback control system to control the movement of the stages. Shown in Figure 1.2(b) are the laser head, three adjustable mirrors, one beam splitter, two remote receivers, two remote sensors, and two interferometers. The mirrors are placed along the side of the stages to take highly accurate position measurements.
Having this highly precise measurement system is crucial for allowing us to continue our research on controlling photolithography wafer scanners.

The electronic systems for control of the stages are motor amplifiers for driving the stages and countermasses, power supplies, and the computing system which is interfaced with the all of the hardware system through a PXI platform. The PXI platform contains both an FPGA and a Real-Time module. Both are programmable in LabVIEW. The LabVIEW Real-Time and FPGA hardware supplies the necessary software and hardware interfacing for rapid prototyping and validation. To interface with the N1231B PCI three-axis board, a custom PCB was made to connect the ribbon cable connectors from the PCI three-axis board and FPGA connectors. With the PCB, we can easily receive signals and send commands from and to the PCI three-axis board.

Figure 1.2 depicts the interface between the hardware system and the LabVIEW Real-Time and FPGA module. The N1231B axis board is placed in a PCI slot of a standard Windows PC. A custom driver written in LabVIEW is used to change the axis board to the direct mode of data output. Data is read from the axis board via the hardware interconnects at the top of the axis board. The hardware connectors are connected to an FPGA (NI 7831R) through a custom interconnect PCB. The FPGA is used to interface with all the sensors and actuators. The sensor data is processed by a LabVIEW-based controller, which calculates the appropriate control output to send to the motor drivers. The measurement system has a sampling rate of 20 MHz and has the ability to take measurements on objects moving at speeds up to 1 m/s.

Figure 1.3: The block diagram of overall hardware and software interface
1.3 System Modeling and Description

This section will emphasize on modeling a single degree-of-freedom (DOF) wafer stage system. The stage system will be discussed later in Chapter 5. As mentioned earlier, the stage is driven by linear permanent magnet motors (LPMMs) which are widely used in high precision positioning instruments due to their high speed and accuracy. LPMMs, however, lose positioning accuracy because of position dependent nonlinearities like cogging force and force ripple.

Cogging is a magnetic disturbance force caused by the attracting and repelling forces between the permanent magnets of the stator and the ferromagnetic core of the translator. This force is a periodic position dependent function and exists even in the absence of motor current. Cogging is negligible when the translator is made from ironless core materials such as epoxy. Force ripples, on the other hand, are caused by irregular electro-magnetic effects due to the variance of winding self-inductance between the permanent magnets and the translator. Therefore, force ripples depends on the motor current as well as the relative position of the coils wound translator with respect to the permanent magnets. Force ripples can cause significant tracking errors in LPMM systems [41].

The linear motor considered here is an LPMM (Trilogy 310) with an ironless core. Therefore, cogging forces can be neglected. In general, a linear motor mounted on an air bearing can be closely approximated as a second-order dynamic system. Ignoring the amplifier time constant, the mathematical model of the system can be expressed as:

\[ M\ddot{x}(t) + b\dot{x}(t) = \kappa u(t) - d(t,x,\dot{x},u) \]  

(1.1)

where \( M \) is the effective mass, \( x(t) \) is the stage position, \( b \) is the damping coefficient, \( \kappa \) is the amplifier gain, and \( u(t) \) is control input command. The uncertain non-linearity, \( d \), is a combination of force ripples and un-modeled disturbances. An accurate mathematical expression of force ripples [15] is too complicated to implement. With the assumption that the permanent magnets are equally aligned at pitch \( P \), the force ripples can be approximated with a simpler model which includes the first several space harmonics [59]:

\[ d_{\text{rip}}(t,x) = \sum_{i=1}^{n} A_i \cos\left(\frac{2\pi}{P} ix\right) + B_i \sin\left(\frac{2\pi}{P} ix\right) \]  

(1.2)

\( A_i \) and \( B_i \) are the unknown parameters that can be estimated by several methods [46][58]. \( n \) is the number of frequency components used to approximate the force ripples. Force ripples, however, are not only position dependent but also current dependent [42]. Lee et. al. [41] suggested that the amplitudes of force ripples are highly dependent on both position and velocity. That is, reasonable force ripples can be modeled as follows:

\[ d_{\text{rip}}(t,x) = \dot{x} \sum_{i=1}^{n} A_i \cos\left(\frac{2\pi}{P} ix\right) + B_i \sin\left(\frac{2\pi}{P} ix\right) \]  

(1.3)

An adaptive feedforward compensator was proposed in [59] for on-line estimation of the unknown parameters and compensation for the force ripple.
1.4 Overview of Iterative Learning Control

The use of feedforward control is beneficial for high performance trajectory tracking in many motion control systems [23][52][50]. Iterative Learning Control is a feedforward control strategy used to improve the performance of a system that executes the same task repeatedly. The method uses the error information gathered from past cycles to improve the performance for the current cycle. Compared to other feedforward methods such as Iterative Controller Tuning and Adaptive Feedforward Control, ILC is the least computationally complex of the three methods and is the most effective at reducing peak errors [45]. ILC is usually integrated into an existing closed-loop system since ILC itself is incapable of stabilizing the system or compensating for non-repetitive disturbances. There are two main ways to inject the learned ILC effort into an existing closed-loop system [10]. One injection point is between the controller and plant as shown in Figure 1.4(a). This architecture utilizes direct access to plant control. The control effort is a sum of the feedback and ILC (feedforward) signals. Because both control efforts are combined at the same point, the feedback controller will have to do less work. The other approach is to inject the ILC command at the reference input as shown in Figure 1.4(b). This architecture directly modifies the reference command and treats the feedback control structure as a closed loop system. Analysis and comparison of the two ILC injection architectures can be found in [19][30]

ILC scheme was originally developed for robot learning and training by Uchiyama [53] and Arimoto et al. [6][49]. Since then many sophisticated ILC schemes have been developed and have been widely applied to practical problems [10][3]. In particular, it has been successfully applied at least in laboratory settings to industrial robots [32][20], engine control [33][55], batch processes [57][29], and semiconductor manufacturing [16][30][60][13].

In practical applications, P-type ILC schemes [6] are most common because of their ease of design and implementation. Still many advanced ILC schemes have been proposed for performance improvement, fast convergence rate, and robustness [38]. To design a stable and efficient ILC algorithm, it is necessary to have certain level of details of the controlled plant model. For example, model-based ILC [27] converges quickly but relies heavily on modeling and may be sensitive to model uncertainty. Optimization-based ILC uses a quadratic performance criterion to obtain an op-
timal learning update law [37]. The $H_{\infty}$ design technique [14] is used to design a robust monotonic convergence but at the expense of performance.

Many problems, however, still exist in applying ILC algorithms. First, disturbances entering into an ILC system are expected to be repetitive. In most situations, disturbances include repetitive disturbances and non-repetitive disturbances. ILC can only compensate for the repetitive disturbances by remembering information from past iterations. The presence of non-repetitive disturbances can severely degrade ILC performance. This is because ILC remembers non-repetitive disturbances even if the disturbance has changed or disappeared. To improve ILC performance, previous research has focused on the concept of filtering out non-repetitive disturbances to reduce their effect on the ILC learning process. A number of ILC with filter design methods have been proposed such as the use of forgetting factor [22], iteration varying filters[35], and time-varying filters [48].

Another problem in applying ILC is that the ILC tuning results are only applicable for a specific trajectory. For each new trajectory, ILC has to be reset and relearned. It can sometimes take several iterations before the tracking error converges to an acceptable level. To reduce tracking error in the first several iterations, many researchers have tried to establish a relationship between the learned ILC input and its respective trajectory. Based on the relationship, the initial ILC output for a new trajectory can be computed in advance. But the difficulty arises when external disturbances contribute to a large portion of the tracking error. In this thesis, we will examine the effect of non-repetitive disturbances and imperfect resetting of ILC systems. Additional investigations will focus on how a good design procedure, combining feedback and ILC, can mitigate these effects.

1.5 Thesis Contributions

The contributions of this thesis can be categorized into two interconnected thrusts: ultra-high positioning performance control in wafer scanning systems and synchronization control of the wafer and reticle stages:

- **Ultra-high positioning performance to achieve further size reduction in scanning process**

  ILC has been used in wafer stage positioning systems for improving tracking error and rejecting repetitive disturbances. ILC algorithms, however, can only compensate for repetitive disturbance. In the presence of non-repetitive disturbance, the performance of ILC scheme degrades. To address this problem, we first analyze the effects of non-repetitive disturbance on the overall system performance and integrate advanced control methods into iterative learning control algorithms. In particular, we place emphasis on how to directly attenuate repetitive disturbances as well as non-repetitive disturbance in time domain. As a result, non-repetitive disturbances have less effect on ILC. To further improve the performance of the ILC scheme, an optimal feedback-feedforward control approach is developed for the integration with the ILC algorithm. Incorporation of optimal feedback-feedforward control offers
significant performance improvement of the ILC scheme when compared to PID feedback control.

Another drawback of ILC is the requirement of invariant trajectory in all iterations. If there is a change in the trajectory, the learning process have to be redone and relearn. Research on the selection of initial ILC effort is also conducted to achieve low tracking error without the need to repeat learning iterations on the new trajectory.

- **Synchronization control of the wafer and reticle stages** A successful scanning process requires the wafer and reticle stages to be synchronized within stringent accuracy. Thus synchronization of the wafer and reticle stages in order to reduce the alignment errors is critical. To this end, this thesis contributes to the development of the controllers for individual stages as well as the controllers for synchronizing the two stages that drive each stage to follow the desired trajectory and at the same time reduce the synchronization error.

### 1.6 Outline of Thesis

The outline of the thesis is as follows. Chapter 2 provides an overview of Iterative Learning Control. This includes a review of commonly deployed design methods, as well as a lifted time domain and frequency domain ILC analysis. The analysis procedures described in this chapter will be used as analysis tools for the ILC designs presented in the rest of the thesis.

Chapter 3 deals with a critical issue in the control of repetitive processes: the effect of non-repeating disturbances. We investigate how a control scheme based on the combination of ILC and disturbance observer (DOB) can be used to mitigate the effects of non-repetitive disturbances. As a result, the combined control scheme can decouple the tracking error into two parts: tracking error due to disturbances and tracking error due to trajectory. The decoupled tracking error then is utilized to properly initialize the ILC for new trajectories so as to speed up the convergence rate.

Chapter 4 is devoted to investigating the performance improvement for ILC in the combination with different feedback controllers. We develop the optimal feedback-feedforward controller and compare results with a standard PID feedback controller. Because we are embedding the ILC into the design of the optimal controller, it is critical to specify the injection points for the ILC effort. In this chapter, performance analysis and stability analysis of different ILC injection architectures combined with the optimal feedback-feedforward control are also discussed.

Chapter 5 focuses on synchronizing the wafer and reticle stages while ensuring that the tracking error of each individual loop is also minimized. Two common synchronization approaches are surveyed and presented. Inspired from these synchronization approaches, a design method for synchronization ILC is proposed. In this chapter, experiments are conducted on the prototype wafer scanner system. Since the two stages of the wafer scanner are stacked vertically, their vibrations are coupled and affect each other. This allows us to study the coupling effects found in actual photolithography systems. By utilizing the two stages, we are able to fully study various synchronization controllers through experimentation as well as simulation.
Chapter 6 presents conclusions and summarizes the major contributions of this thesis. In addition, future research recommendations and possible extensions of the work in this thesis are discussed.
Chapter 2

Iterative Learning Control

This chapter is devoted to survey existing literature on iterative learning control (ILC). The content provides a perspective of the important ideas, assumptions, and limitations of ILC.

This chapter is organized as follows. Section 2.1 introduces the basic ILC set up for a discrete-time SISO system. Stability analysis and performance analysis are presented in Section 2.2. Section 2.3 discusses the most popular ILC design methods along with their stability and performance consideration.

2.1 Iterative Learning Control

The idea behind Iterative Learning Control, as the name implies, is based on the paradigm of human learning. In a repetitive process, information from earlier iterations of the process can be used to improve performance in the current iteration.

ILC is attractive because of its simplicity of design, implementation, and analysis. There are four basic assumptions while applying ILC algorithms. The first assumption requires the current system to be stable from the learning control input to the measured output. The next three require the trajectory, disturbances and initial conditions to be the same for every iteration. Due to the repetitive nature of the wafer scanning process, the ideal feedforward signal can be generated through iterative refinement. The ILC scheme was originally formulated for continuous-time systems, but parallel results can be obtained for discrete-time systems. It is better to develop an ILC algorithm for discrete-time operation since storing error information and calculating it requires digital signal processing.

As stated, ILC can be injected into a stable closed-loop system in two ways. The parallel structure and the serial structure are fundamentally similar except for the transfer functions from the ILC injection point to system output. Once the parallel structure is analyzed, the serial structure is easy to derive in the similar way. Assume that the system executes a repetitive process with period of \( N \) samples and starts at rest condition at the beginning of each iteration. From Figure 1.4,
the system output \( y_j(k) \) can be obtained by
\[
y_j(k) = T_r(z) r(k) + T_u(z) u_{j}^{ILC}(k) + T_d(z) d_j(k)
\]
(2.1)
where in the case of the parallel architecture
\[
T_u(z) = T_d(z) = \frac{P(z)}{1 + P(z)C(z)}, \quad T_r(z) = \frac{P(z)C(z)}{1 + P(z)C(z)}
\]
(2.2)
and in the case of the serial architecture
\[
T_u(z) = T_r(z) = \frac{P(z)C(z)}{1 + P(z)C(z)}, \quad T_d(z) = \frac{P(z)}{1 + P(z)C(z)}
\]
(2.3)

The subscript \( j \) reflects the iteration number of the process. Time index, \( k \), ranges from 0 to \( N - 1 \). Note that \( r(k) \) has no subscript \( j \) because it is repetitive for every iteration. The feedback controller \( C(z) \) needs to be designed in advance to ensure the stability of the closed-loop system. The disturbance is expressed as \( d_j(k) \) which changes from iteration to iteration. For simplify, a single-input and single-output system, \( P(z) \), is considered. The goal of ILC is to obtain the ideal \( u_{j}^{ILC}(k) \) to make the error as small as possible. A standard SISO ILC learning law can be formulated as:
\[
u_{j+1}^{ILC}(k) = Q(z)[u_{j}^{ILC}(k) + L(z)e_j(k)]
\]
(2.4)
where \( Q(z) \) is defined as Q-filter to add robustness and \( L(z) \) is defined as learning function. \( L(z) \) determines how much and what parts of the tracking error are used to update \( u_{j+1}^{ILC} \). The tracking error \( e_j(k) \) is defined by \( e_j(k) = r(k) - y_j(k) \). Since we have collected all accessible data in advance, the Q-filter and the learning function can be designed as noncausal functions. This makes it possible for ILC to anticipate and preemptively respond to repetitive disturbances.

2.2 Stability and Performance Analysis of ILC Algorithms

This section provides the stability and performance analysis of an ILC algorithm combined with a close-loop wafer stage system with respect to the iteration domain. As will be seen, the choice of Q-filters and learning functions affects the performance, the convergence rate, and the robustness of the ILC system.

2.2.1 Stability Analysis

The ILC system is stable if \( u_{j+1}^{ILC} \) is bounded as iteration number goes to infinity. To develop stability analysis for an ILC scheme, frequency representation or lifted formulation is commonly used [10]. It is easier to analyze the overall stability and performance in the frequency domain. By applying z-transformation to (2.1) in the case of the parallel architecture and (2.4), we obtain respectively
\[
Y_j(z) = T_u(z)[u_{j}^{ILC}(z) + D_j(z)] + T_r(z)R(z)
\]
(2.5)
and

\[ U_{j+1}^{ILC}(z) = Q(z)[U_j^{ILC}(z) + L(z)E_j(z)] \]  

(2.6)

where \( E_j(z) = R(z) - Y_j(z) \). Then, (2.6) can be rewritten as

\[ U_{j+1}^{ILC}(z) = Q(z)[U_j^{ILC}(z) + L(z)(R(z) - Y_j(z))] \]

\[ = Q(z)[1 - L(z)T_u(z)]U_j^{ILC}(z) + Q(z)L(z)[(1 - T_r(z))R(z) - T_dD_j(z)] \]  

(2.7)

Given that disturbances are bounded,

\[ \|D_j(z)\|_\infty < \beta \in \mathbb{R}^1 \ \forall \ j \in 1,2,... \]  

(2.8)

when \( N = \infty \), a sufficient condition for the asymptotic stability of the ILC system is

\[ \|Q(z)[1 - L(z)T_u(z)]\|_\infty < 1. \]  

(2.9)

In the other words, the converged control effort (\( \lim_{j \to \infty} U_j^{ILC} \)) exists when the above condition is satisfied. Note that the stability condition obtained in terms of z-domain representations is usually more conservative since it is assumed that the time horizon is infinite. A lifted formulation of the ILC problem introduced later in this chapter, on the other hand, is considered over a finite time horizon. Therefore, the stability condition derived from the lifted formulation is necessary and sufficient [37]. From (2.9), it indicates that \( \|L(z)T_u(z)\| \) should stay within a unit circle with center at 1 to ensure the stability of the ILC system if \( Q(z) = 1 \). It is clear that setting \( Q(z) \) to be a filter with gain less than one can enlarge the stability region. The price paid for this action is that the performance of the ILC system degrades. This trade-off will be further discussed in the next section.

It should be noticed that the necessary condition for stability in (2.9) ensures only asymptotic convergence of the error. That means even though the stability condition is satisfied the large transient growth before the error converges may occur. Transient growth is extremely undesirable in practice because of high overshoots of the system trajectory. To avoid large learning transients, monotonic convergence analysis for an ILC algorithm is also important. The system described in (2.5) and (2.6) is monotonically convergent in the sense of the Euclidean 2-norm if \( \|e_\infty - e_{j+1}\|_2 \leq \|e_\infty - e_j\|_2 \) where \( e_\infty \) is a vector of tracking error when \( j \) goes to infinity. It was shown in [10] that the stability condition (2.9) is identical to the monotonic convergence condition when the system is SISO. We also refer interested reader to [2] and [30] for a detailed development and analysis of monotonic convergence analysis.

### 2.2.2 Performance Analysis

Performance analysis of an ILC system is based on the asymptotic value of the error. In order to understand the effect of disturbances on performance, especially non-repetitive disturbances, we
express the tracking error as a function of reference trajectory and disturbances in z-domain.

\[
E_{j+1}(z) = R - T_u(z)[U_{j+1}^{ILC}(z) + D_{j+1}(z)] - T_r(z)R(z) \\
= [1 - T_r(z)]R(z) - T_u(z)Q(z)U_{j+1}^{ILC}(z) - Q(z)L(z)T_u(z)E_j(z) - T_d(z)D_{j+1}(z)
\]

(2.10)

If the converged control input exists, then the asymptotic error is obtained by using (2.5) and (2.10).

\[
\lim_{j \to \infty} E_j(z) = \frac{(1 - Q(z))(1 - T_r(z))}{1 - Q(z)[1 - L(z)T_u(z)]} R(z) + \frac{T_d(z)(Q(z)D_j(z) - D_{j+1}(z))}{1 - Q(z)[1 - L(z)T_u(z)]}
\]

(2.11)

With the design of \( Q(z) = 1 \), the tracking error depends only on the difference of the disturbances from iteration to iteration.

\[
\lim_{j \to \infty} E_j(z) = \frac{T_d(z)(D_j(z) - D_{j+1}(z))}{L(z)T_u(z)}
\]

(2.12)

Above equation indicates that non-repetitive disturbances contribute to the error by the differences between iterations. In the other words, the asymptotic error converges to zero only if no non-repetitive disturbances exists, i.e. \( D_j(z) = D_{j+1}(z) \) for all \( j \). However, disturbances in practical applications include repetitive as well as non-repetitive disturbances. This makes the asymptotic error impossible to converge to zero. Furthermore, it is contradicting the existence of converged control input, an assumption for (2.11). To relive the effect of non-repetitive disturbances, a proper design of the learning function \( L(z) \) and/or the close-loop feedback controller with higher gain should be considered.

In general cases, the stability condition (2.9) restricts on setting Q-filter to be one. It also restricts the selection of \( L(z) \) resulting in the limitation on the attenuation of non-repetitive disturbances. This makes the asymptotic error depend on the designs of Q-filters, learning functions, and feedback controllers. The choice of Q-filter and the learning function in fact is a three-way trade-off between performance, robustness, and convergence. Therefore, this thesis aims to integrate advanced control techniques into an ILC algorithm with proper design of Q-filter and learning function so as to achieve better performance without compromising the robustness and convergence.

### 2.3 Different ILC Design Methods

Many different types of ILC have been proposed to date. This section introduces four commonly known ILC design methods: P-type ILC, higher-order ILC, plant-inversion based ILC, and optimization based ILC. The first two methods can easily achieve high performance with very limit plant knowledge, but require more iterations to converge. The last two methods can converge faster and have promising performance, but require more knowledge of the plant. One may notice later that the different methods presented here can all be formulated in the form of (2.4). That is, these methods essentially differ in the designs of Q-filters and leaning functions. In the following section, the ILC methods are consistently implemented in the parallel architecture and the system of interest is SISO.
2.3.1 P-type ILC

P-type ILC was first proposed by Arimoto et al\[6\]. Since then the P-, D-, and PD-type learning functions were proposed for practical implementation. As its name implies, P-type ILC updates the control effort by multiplying the previous error information with a scalar. The D-type ILC, similar idea to P-type ILC, uses the derivative of the error information for updating the control effort. A PD-type ILC which consists of proportional and derivative terms in the learning update law is expressed as:

$$u_{ILC}^{j+1}(k) = u_{ILC}^{j}(k) + k_p e_j(k + m) + k_d [e_j(k + 1) - e_j(k)]$$ (2.13)

where $m$ is the number of step advance which typically equals to the delay of the system, $k_p$ is the proportional gain, and $k_d$ is the derivative gain. The Q-filter in the PD-type ILC is set to one and the learning function in the z-transformation form is

$$L(z) = k_p z^m + k_d (z - 1)$$ (2.14)

Just like PD feedback controllers, the P-,D-, or PD-type ILC formulations are obtained by tuning $k_d$ and $k_p$ \[6\][9][18]. For stability of P-type ILC, $k_p$ needs to be chosen to fulfill

$$\| 1 - k_p h(m) \| < 1$$ (2.15)

where $h(i)$ is the time-indexed plant impulse response\[10\]. The stability condition for the D-type or PD-type laws can be derived from the frequency domain stability criterion (2.9) with setting $Q(z) = 1$. This class of ILC design method is attractive because of its simple implementation and minimal requirement of plant knowledge. It should be noticed that designing the Q-filter as a low-pass filter rather than one allows $k_p$ and $k_d$ gains to be tuned higher without violating the stability condition.

2.3.2 Higher-Order ILC

Higher-Order ILC, as its name suggested, uses not only the most recent previous control input and error information, but all of the previous available past information for the performance improvement. system.

In other words, a higher-order ILC (HOILC) utilizes information from more than one previous iteration for updating the control effort in the ILC algorithm\[8] \[11\]. The updating law of a second-order ILC, for a simple example, is

$$u_{ILC}^{j+1}(k) = Q_1(z) [u_{ILC}^{j}(k) + L_1(z) e_j(k)] + Q_2(z) [u_{ILC}^{j-1}(k) + L_2(z) e_{j-1}(k)]$$ (2.16)

It was shown that the second-order ILC has faster convergence speed than the first-order ILC. However, since HOILC involves information processing more than one iteration, the difficulty in designing the Q-filters and the learning functions arises. The stability analysis also becomes
more complicated. Saab [43] showed that HOILC does not add to the optimality of standard first-order ILC in the sense of minimizing the trace of the control error covariance matrix. A thorough analysis and experimental results on a second-order ILC were presented in [34]. The study of monotonic convergence, however, is still an open question for HOILC. It leads to the conclusion that the second-order ILC works as well as the first-order ILC in the perspective of performance or robustness.

2.3.3 Plant-Inversion Based ILC

In feedforward control, the plant-inversion approach can yield perfect tracking in the absence of external disturbances and modeling errors. ILC, as already stated, can be seen as a feedforward strategy. Thus it is straightforward to design the learning function based on the inversion of the plant. Let us consider a system which is commanded to follow a specific trajectory without being subjected to disturbances. The tracking error in the first iteration is denoted as

\[ e_1(k) = r(k) - y_1(k) \]

where

\[ y_1(k) = T_u(z)u_1(k) + T_r(z)r(k). \]

Assume that the initial ILC control effort, \( u_1(k) \), equals to zero. Then, \( y_1(k) = T_r(z)r(k) \). The ILC control effort in the second iteration is obtained by

\[
 u_{ILC}^2(k) = u_1(k) + L(z)e_1(k) = L(z)e_1(k)
 = L(z)(r(k) - T_r(z)r(k))
\]

Then the tracking error in the second iteration becomes

\[
 e_2(k) = r(k) - y_2(k)
 = r(k) - T_r(z)r(k) - T_u(z)u_2(k)
 = [r(k) - T_r(z)r(k)][1 - T_u(z)L(z)]
\]

The plant-inversion based ILC sets \( L(z) = T_u^{-1}(z) \). Then it needs only one iteration to achieve zero tracking error at the second iteration if external disturbances and modeling errors do not exist. As can be seen, the plant-inversion method highly depends on the accuracy of the model. The model uncertainty, especially at high frequencies, causes the approach to be sensitive to noise and disturbances, and prevents the system from achieving perfect tracking performance. In order to circumvent this problem, the plant-inversion based ILC usually comes with a low-pass Q-filter. The updating law then is modified as

\[
 u_{j+1}^{ILC}(k) = Q(z)[u_j^{ILC}(k) + T_u^{-1}(z)e_j(k)]
\]

The low-pass filter plays an important role in filtering out the high frequency components and leads the plant-inversion method to overcome the model uncertainty problem [28][17].
2.3.4 Optimization Based ILC

In many applications such as wafer scanning systems and robotic systems, models are available and/or they can be obtained using modern identification techniques. This makes it possible for researchers to design ILC algorithms by using optimization techniques \[4\][40][21]. Based on the user-defined optimality criterion, the corresponding optimal Q-filters and optimal learning functions are derived. A general optimality criteria of the optimization based ILC is

\[
J_{j+1}(u_{j+1}^{ILC}) = e_{j+1}^T W_e e_{j+1} + (u_{j+1}^{ILC})^T W_u u_{j+1}^{ILC} + \lambda (u_{j+1}^{ILC} - u_{j}^{ILC})^T (u_{j+1}^{ILC} - u_{j}^{ILC}) \tag{2.18}
\]

where \(e_{j+1}\) and \(u_{j+1}\) are the tracking error and the ILC effort in the j-th iteration in the lifted domain as explained below. \(W_e\) and \(W_u\) are the positive definite tracking error and positive semi-definite control weighting matrices. \(\lambda\) is a non-negative weighting scalar which penalizes the difference between two consequential control effort such that a smooth solution of the optimal learning control effort can be obtained. Note that the optimization based ILC is typically developed in the lifted formulation. Instead of discrete signals, vectors stacking all signals from time index 0 to N-1 are used. For example, \(u_{j+1}^{ILC}\) and \(e_{j+1}\) in the bold font in (2.18) mean

\[
u_{j+1}^{ILC} = \begin{bmatrix}
u_{j+1}^{ILC}(0) \\
u_{j+1}^{ILC}(1) \\
\vdots \\
u_{j+1}^{ILC}(N-1)
\end{bmatrix}, \quad e_{j+1} = \begin{bmatrix}e_{j+1}(m) \\
\vdots \\
e_{j+1}(m+N-1)
\end{bmatrix} \tag{2.19}
\]

The lifted tracking error is given by

\[
e_{j+1} = r - y_{j+1} \\
y_{j+1} = T_r r + T_u u_{j+1}^{ILC} \tag{2.20}
\]

where \(T_r\) and \(T_u\) in bold font represent matrices formed by the impulse response coefficients of the transfer function \(T_r(z)\) and \(T_u(z)\), respectively:

\[
T_r = \begin{bmatrix}
h_r(0) & 0 & \cdots & 0 \\
h_r(1) & h_r(0) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
h_r(N-1) & h_r(N-2) & \cdots & h_r(0)
\end{bmatrix}, \quad T_u = \begin{bmatrix}
h_u(0) & 0 & \cdots & 0 \\
h_u(1) & h_u(0) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
h_u(N-1) & h_u(N-2) & \cdots & h_u(0)
\end{bmatrix} \tag{2.21}
\]

The optimal solution is obtained by setting the derivative of \(J_{j+1}\) with respect to \(u_{j+1}^{ILC}\) zero, i.e. 

\[
\frac{\partial J_{j+1}(u_{j+1}^{ILC})}{\partial u_{j+1}^{ILC}}. \text{ It becomes}
\]

\[
u_{j+1}^{ILC} = (T_u^T W_e T_u + W_u + \lambda I)^{-1} \left[ (T_u^T W_e T_u + \lambda I) u_{j}^{ILC} + T_u^T W_e e_j \right] \tag{2.22}
\]
The optimal Q-filter and the optimal learning function are expressed as the standard form like (2.4).

$$Q^{opt} = (T_u^TW_eT_u + W_u + \lambda I)^{-1}(T_u^TW_eT_u + \lambda I)$$

$$L^{opt} = (T_u^TW_eT_u + \lambda I)^{-1}T_u^TW_e$$ (2.23)

Note that the ILC update law shown in (2.22) is stable and monotonically convergent in control effort in the sense of Euclidean 2-norm. The monotonic convergence can be proven by deriving the control effort evolution from iteration to iteration.

$$\|u^{ILC}_\infty - u^{ILC}_{j+1}\|_2 = (T_u^TW_eT_u + W_u + \lambda I)^{-1}\lambda \|u^{ILC}_\infty - u^{ILC}_j\|_2$$ (2.24)

Since $W_e$ and $W_u$ are positive definite,

$$\rho((T_u^TW_eT_u + W_u + \lambda I)^{-1}\lambda) < 1$$ (2.25)

where $\rho(\cdot)$ denotes the largest eigenvalue of the matrix. Then it can be claimed from (2.24) that $\|u^{ILC}_\infty - u^{ILC}_{j+1}\|_2 \leq \|u^{ILC}_\infty - u^{ILC}_j\|_2$. Furthermore, the monotonic convergence in error in the sense of Euclidean 2-norm can be achieved if $W_e = \rho_e I$ where $\rho$ is a positive value [30].

The converged ILC effort can be found as

$$u^{ILC}_\infty = (I - Q^{opt} + Q^{opt}L^{opt}T_u)^{-1}Q^{opt}L^{opt}(I - T_r)r$$ (2.26)

Then the converged error for the optimization-based ILC is

$$e_\infty = r - (T_r r - T_u u^{ILC}_\infty)$$

$$= [I - T_u(I - Q^{opt} + Q^{opt}L^{opt}T_u)^{-1}](I - T_r)r$$

$$= [I - T_u(T_u^TW_eT_u + W_u)^{-1}T_u^TW_e](I - T_r)$$ (2.27)

As can be seen, the weighting scalar, $\lambda$, has no effect on the converged ILC effort and error. However, it does affect the rate of the convergence in the sense that $\lambda$ determines the smoothness of the ILC effort. In brief, the smaller the weighting scalar $\lambda$ and the larger the weighting matrices $W_u$ and $W_e$, the faster the convergence of ILC effort and error. To achieve better converged error, however, $W_e$ should be relative larger than $W_u$.

### 2.4 Chapter Summary

A standard Iterative Learning Control (ILC) algorithm has been presented in this chapter. Associated stability and performance analysis were also studied. We provided relevant background material and reviewed some of the literature on ILC. In addition, we pointed out there exists a trade-off between the performance, the convergence rate, and the robustness with the choice of Q-filter and learning functions. Recently, more and more research efforts turn to design time-varying filters utilizing time-frequency information to examine the frequency content of error signals. Alleyne...
[48] proposed a time-frequency adaptive Q-filter to enhance robustness, while taking advantage of the superior performance properties of a high bandwidth Q-filter where needed. In [61], discrete wavelet packet algorithm, as a time-frequency analysis tool, is employed to decompose the tracking error into different frequency regions so that a more efficient cutoff frequency tuning method is designed. However, the time-varying Q-filter requires complex computation and large memory resources.

This chapter also surveyed several types of ILC methods. Among them, P-type ILC is the simplest and requires the least information of the plant model. This makes P-type ILC robust to system variations and system uncertainty. However, the price to pay is slow convergence rate and poor performance. In the next two chapters, advanced control methods are integrated with ILC schemes to overcome the drawbacks of the use of P-type ILC. P-type ILC introduced here will be used as a basis for the ILC designs presented in the rest of the thesis. The notations presented in this chapter will also be used throughout the thesis.
Chapter 3

Non-Repetitive Disturbance and Initial Control Effort Selection in ILC

ILC is a very effective technique to improve the tracking performance of systems carrying out repetitive tasks. However, a main assumption of ILC is that disturbances of systems have to be iteration independent. The tracking performance of an ILC system degrades while non-repetitive disturbances and measurement noises are present [36]. To mitigate the effect of non-repetitive disturbances, Heinzinger [22] proposed a D-type ILC law with the use of a forgetting factor. The idea behind using a forgetting factor is that as the iterations progress, information from the oldest iterations will be forgotten and thus the non-repetitive disturbances will have less effect on the current control effort. The use of a forgetting factor, however, actually results worse in tracking performance than a standard PD-type ILC with carefully chosen learning gains. Similar to the idea of using a forgetting factor, Norrlof [35] designed an iteration-varying learning function in the learning update law which takes measurement noise into account. Instead of decreasing the effect of oldest information, the learning gain is decreased iteration by iteration. This algorithm only takes measurement noise into account. Moreover, the error contraction rate reduces with iteration number and eventually the learning practically stops before achieving promising performance.

Another assumption that limits the use of ILC is that the trajectory is expected to be the same in all iterations. If the control objectives or tasks change, the ILC system will need to reset and relearn the learning process. All efforts that went into the previous learning processes go into vain. In order to improve the efficiency of utilizing ILC algorithms, it is essential that ILC is capable of learning consecutively from different tracking tasks. Xu [56] proposed a direct learning for a class of trajectories with the same shape but different magnitude scales. Arif et al. [5] incorporated the information learned from previous ILC into the selection of the initial ILC effort by means of locally weighted learning to model the inverse dynamics of the system. But the difficulty is that the learned ILC effort includes information about the tracking error due to trajectory and the tracking error due to disturbances. Without enough data of the system states and the control inputs gathered from different trajectory tracking tasks, it is hard to build a model to compute a proper initial ILC effort for each new trajectory.
To circumvent these problems, disturbance observer (DOB) is introduced and incorporated into the ILC system in this chapter. The proposed control scheme combined the unique advantages of ILC and DOB such that the performance of ILC is improved. DOB is used to estimate repetitive disturbances as well as non-repetitive disturbances, and then utilize the estimates to cancel disturbances in time-domain rather than in iteration-domain. By directly removing disturbances in time-domain, attenuation of the error due to both repetitive and non-repetitive disturbance is achieved. One more benefit of removing disturbances by DOB is that the tracking error can be separated into the error due to disturbances and the error due to a trajectory following. Since the main disturbances are eliminated by DOB, the information used to update ILC becomes highly related to tracking error due to the trajectory. This makes it easier to approximate the plant model without subject to disturbance and update the plant parameters from previous learning process. Then a proper selection of an initial ILC effort can be simply obtained from the updated parameters. At the same time, the updated parameters are also used to update the design of DOB to achieve better disturbance rejection.

This chapter is organized as follows. Section 3.1 describes the concept and the formula of DOB. Section 3.2 presents the combined control scheme of ILC and DOB along with stability and performance analysis. Simulation and experimental results are also presented to demonstrate the performance improvement by the proposed scheme. In section 3.3, an initial control effort in ILC is selected based on the combined ILC scheme to speed up the convergence for a new trajectory. Simulation studies are conducted to show that the convergence performance can be further improved when a proper initial control effort is selected.

### 3.1 Disturbance Observer Design

In order to attain higher precision motion control, several approaches have been proposed to suppress the effect of external disturbances to the system such as high-gain controllers, disturbance observer, adaptive control, and robust control. Among these approaches, disturbance observer can be designed as an add-on controller so as to reject external disturbances without compromising the performance. This allows the existing feedback controller and the disturbance observer to be designed independently.

#### 3.1.1 Continuous-time Disturbance Observer

The disturbance observer was originally proposed by Ohnishi [39] and has become a popular method for eliminating disturbances in motion control since then. The disturbance observer estimates the disturbance from the control input and measurement output, and then utilizes the estimate to cancel the disturbance. Since the actual plant model is not known precisely in most cases, the disturbance is estimated by the inversion of nominal plant model rather than the inversion of actual plant model. The block diagram for disturbance observer is shown in Fig. 3.1. From the figure, we can estimate disturbance by
Figure 3.1: Continuous-time disturbance observer

\[
\dot{D}(s) = P_n^{-1}(s)[Y(s) + N(s)] - U(s)
\]  

(3.1)

where \(P_n(s)\) is the nominal plant model. \(Y(s), N(s),\) and \(U(s)\) are measurement output, noise, and control input, respectively. In practice, the inverse is not realizable. Therefore, \(Q_D(s)\) in Fig.3.1 is chosen such that the product \(Q_D(s)P_n^{-1}(s)\) is realizable.

The main issues in the design of disturbance observer are the selection of \(Q_D(s)\) and its implementation. As is well known, \(Q_D(s)\) should provide a good balance between disturbance rejection versus stability robustness and noise sensitivity. From the block diagram, the transfer functions from command input, disturbance, and noise to measurement output are respectively

\[
G_{UY} = \frac{Y(s)}{U(s)} = \frac{P(s)P_n(s)}{P_n(s) + Q_D(s)(P(s) - P_n(s))}
\]  

(3.2)

\[
G_{DY} = \frac{Y(s)}{D(s)} = \frac{P(s)P_n(s)(1 - Q_D(s))}{P_n(s) + Q_D(s)(P(s) - P_n(s))}
\]  

(3.3)

\[
G_{NY} = \frac{Y(s)}{N(s)} = -\frac{P(s)Q_D(s)}{P_n(s) + Q_D(s)(P(s) - P_n(s))}
\]  

(3.4)

Note that as \(Q_D(s)\) gets closer to one, \(G_{UY}\) becomes closer to \(P_n(s)\) and \(G_{DY}\) becomes closer to zero. This implies that the disturbance observer forces the actual plant to behave like a nominal plant and provides perfect disturbance rejection. On the other hand, as \(Q_D(s)\) gets closer to zero \(G_{UY}\) become closer to \(P(s)\) and \(G_{NY}\) becomes closer to zero. Normally, \(Q_D(s)\) is set to 1 at low frequency for model uncertainties and disturbance rejection, and is set to 0 at high frequency for avoiding noise feedback and robustness. A simple design method of \(Q_D(s)\), which balances performance and robustness, has been suggested by Umeno and Hori [54].

The plant dynamics of interest here is described by a second order transfer function. Thus, a third order filter is consistent with the above desired properties and has been chosen for this research.

\[
Q_D(s) = \frac{3\tau s + 1}{\tau^3 s^3 + 3\tau^2 s^2 + 3\tau s + 1}
\]  

(3.5)
where \( \tau \) is the time constant which determines the cutoff frequency of \( Q_D(s) \). The time constant should be larger than the sampling time, \( T_s \), in digitized implementation. Hence, \( \tau \) can be expressed as \( \tau = \alpha T_s \) where \( \alpha \) is a design parameter and larger than 1. The following section will discuss how to select a proper value for \( \alpha \).

### 3.1.2 Unmodeled Dynamics Effects and Bandwidth Limits

The mathematical model of our plant has been discussed in Chapter 1. Based on data from system identification, the transfer function of the nominal plant model is

\[
P_n(s) = \frac{11.79}{5.3s^2 + 7.2s}
\]

The inaccuracy of the nominal plant model is due to unmodeled dynamics such as actuator dynamics, structural models, force ripple, vibrations, and so on. To illustrate the effects of unmodeled dynamics, assume that actuator dynamics is the only source of unmodeled dynamics. Then the actual model can be expressed as

\[
P(s) = [1 + \Delta(s)]P_n(s)
\]

where \( \Delta(s) = T_a s / (T_a s + 1) \) is the multiplicative uncertain term caused by the actuator dynamics with time constant \( T_a \). Since the controller is implemented in digital form, it is convenient to design the controller in the discrete-time domain. We refer the reader to [24][47][26] for further details of discrete-time disturbance observer. Transforming nominal plant model (3.6) to discrete-time representation yields

\[
P_n(z) = \frac{z^{-1}(1.779e^{-7} + 1.779e^{-7}z^{-1})}{1 - 1.999z^{-1} + 0.9995z^{-2}}
\]

where the sampling period is 400 \( \mu \)sec. The above equation implies that the system includes only one step time delay. The one step delay is accommodated by the design of a discrete-time DOB which is shown in Fig. 3.2(a). For analysis purpose, an equivalent discrete-time DOB is shown in Fig. 3.2(b). From the equivalent block diagram, the transfer functions in the \( j^{th} \) iteration from the control input, \( u_j(k) \), and disturbances, \( d_j(k) \), to the system output are easily derived.

\[
G_{UY}(z) = \frac{P(z)P_n(z)}{P_n(z) + z^{-1}Q_D(z)(P(z) - P_n(z))}
\]

\[
G_{DY}(z) = \frac{P(z)P_n(z)(1 - z^{-1}Q_D(z))}{P_n(z) + z^{-1}Q_D(z)(P(z) - P_n(z))}
\]

The discrete-time \( Q_D \) is obtained by converting (3.5) into \( z \) domain representation. As already discussed, increasing the bandwidth of the discrete-time \( Q_D \) helps performance in terms of disturbance rejection, but may cause the system to become unstable due to the unmodeled dynamics. To
Consider the stability robustness, the characteristic polynomial of the disturbance loop is

\[
A_c(z) = P_n(z) + z^{-1}Q_D(z)(P(z) - P_n(z))
\]

(3.11)

Clearly, the asymptotic stability of the disturbance observer is assured if

\[
\|\Delta(e^{j\omega})Q_D(e^{j\omega})\| < 1 \text{ for all } \omega
\]

(3.12)

A bode plot of the magnitudes of \(1/\Delta\) and \(Q_D(z)\) with different bandwidths is shown in Fig. 3.3 to visualize the bandwidth limitation of \(Q_D(z)\) imposed by unmodeled dynamics.

In the bode plot, three different time constants of \(Q_D(z) (\tau = 10T_s, 20T_s, \text{and } 40T_s)\) are selected for the comparison. The corresponding cutoff frequencies are 410, 205, and 102 Hz which are respectively denoted as \(Q_{D1}, Q_{D2}, \text{and } Q_{D3}\). The unmodeled actuator dynamics is set to have time constant \(T_a = 10T_s\). Among three filter designs, \(Q_{D3}\) is the most robust and conservative design but has the worse performance in terms of disturbance rejection. \(Q_{D2}\) can achieve better disturbance rejection but pays for stability robustness. \(Q_{D3}\) violates the stability condition and it is not a serious contender. In most cases, \(Q_{D2}\) is selected because it has better performance and acceptable robustness margin.

To investigate the closed-loop properties, a discrete-time PID feedback controller sampled at 2.5 kHz \((T_s = 400 \mu \text{sec})\) is designed

\[
C(z) = K_pE(z) + \frac{K_iT_s}{1 - z^{-1}}E(z) + \frac{K_d}{T_s}(1 - z^{-1})E(z)
\]

(3.13)

where \(K_p = 30000, K_d = .012K_p, \text{ and } K_i = 2K_p\). Figure 3.4 shows the overall control diagram for a system that includes ILC and DOB. From the figure, the closed loop transfer functions from
Figure 3.3: $Q_D$ filter b

Figure 3.4: Overall control block diagram
Figure 3.5: Comparison on the performance in terms of disturbance rejection with and without disturbance observer

control input, disturbance, and reference input to system output become

\[
T_u'(z) = \frac{G_{UY}(z)}{1 + C(z)G_{UY}(z)} = \frac{P(z)}{1 + P(z)C(z) + z^{-1}Q_D(z)\Delta(z)} \tag{3.14}
\]

\[
T_d'(z) = \frac{G_{DY}(z)}{1 + C(z)G_{DY}(z)} = \frac{P(z)(1 - z^{-1}Q_D(z))}{1 + P(z)C(z) + z^{-1}Q_D(z)\Delta(z)} \tag{3.15}
\]

\[
T_r'(z) = \frac{C(z)G_{UY}(z)}{1 + C(z)G_{UY}(z)} = \frac{P(z)C(z)}{1 + P(z)C(z) + z^{-1}Q_D(z)\Delta(z)} \tag{3.16}
\]

The frequency responses of the resulting closed-loop system with and without disturbance observer to disturbance input are shown in Figure 3.5. Figure 3.6 compares the closed-loop system frequency responses from the reference input to system output with and without disturbance observer. The plot indicates that the performance in terms of command following is maintained as long as the frequency content of the entire reference trajectory is concentrated at low frequencies.

### 3.2 ILC in Combination with DOB

In section 2.2, we have learned that the presence of non-repetitive disturbances degrades the performance of ILC systems. It was concluded that the design of the learning function \(L(z)\) is a trade
off between attenuation of non-repetitive disturbances and stability. [10] directly pointed out that the best approach to reject non-repetitive disturbances is a feedback controller used in combination with the ILC. The efficiency of DOB for disturbance rejection has been discussed in the previous section. A combined control scheme is developed in this section to enhance the performance of ILC systems by using DOB. Resulting stability and performance properties of the combined control scheme are also analyzed. Simulation and experimental results illustrate the performance improvement of the proposed control scheme including the ability to reduce error in the presence of non-repetitive disturbances.

Recall the overall control diagram in Figure 3.4. The purpose of DOB is to estimate repetitive disturbances as well as non-repetitive disturbances and utilize the estimate to cancel disturbances in the time domain. The purpose of ILC is to learn a feedforward control input for a specific trajectory in the iteration domain. Since the closed-loop transfer functions have been derived in (3.14), (3.15), and (3.16), rewriting $Y_j(z)$ as a function of ILC effort, disturbance input, and reference trajectory is straightforward.

$$Y_j(z) = T'_u(z)U_j^{ILC}(z) + T'_r(z)D_j(z) + T'_d(z)R(z)$$

(3.17)

where

$$U_{j+1}^{ILC}(z) = Q(z)[U_j^{ILC}(z) + L(z)E_j(z)]$$

(3.18)

Intuitively $T'_u(z) \approx T_u(z)$, $T'_r(z) \approx T_r(z)$, and $T'_d(z) \approx T_u(z)(1 - z^{-1}Q_D(z))$ when $\Delta(z)$ is small.
or \(z^{-1}Q_D(z)\Delta(z)\) is relatively smaller than \((1 + P(z)C(z))\) where \(T_u(z)\) and \(T_r(z)\) are as given by Eq. (2.2). This means that the non-repetitive disturbances \(D_j(z)\) has less significant effect on the output due to the multiplicative term, \((1 - z^{-1}Q_D(z))\). On the other hand, the ILC effort \(U_{j+1}^{ILC}(z)\) and reference trajectory \(R(z)\) have the same effects on the output.

Stability analysis of the proposed control scheme follows the similar procedure as that of the standard ILC scheme as explained in (2.5)-(2.7). Therefore, the ILC effort of the proposed control scheme can be expressed as

\[
U_{j+1}^{ILC}(z) = Q(z)[1 - L(z)T_u'(z)]U_{j}^{ILC}(z) + Q(z)L(z)[(1 - T_r'(z))R(z) - T_dD_j(z)]
\]  

(3.19)

Given that the magnitude of disturbances is bounded, when either \(Q(z)\) and \(L(z)\) are causal or \(N = \infty\), a sufficient condition for asymptotic stability of the ILC system is \(\|Q(z)[1 - L(z)T_u'(z)]\|_{\infty} < 1\).

The stability condition is the same as that of a standard ILC scheme except for slight difference between \(T_u(z)\) and \(T_u'(z)\). To further visualize the stability analysis, a simple P-type ILC, \(L(z) = k_pz^{n_d}\), is applied to the plant (3.7). \(n_d\) is in general set to the index of the first nonzero value of the plant impulse response. For an ILC design with \(k_p = 45000\), \(n_d = 1\), and \(Q(z) = 1\), the stability conditions for ILC alone and ILC in the combination with DOB are plotted in Figure 3.7. The bode plot indicates that the closed-loop system still yields comparable results for stability robustness even when there exists unmodeled dynamics. Note that \(Q(z)\) is set to one only for simplifying the analysis. \(Q(z)\) is typically a zero-phase low pass filter to ensure that the magnitude plot of \(\|1 - L(z)T_u'(z)\|_{\infty}\) stay under 0 db line.

To see the effect of non-repetitive disturbances, the asymptotic error in the proposed control scheme is expressed as

\[
\lim_{j \to \infty} E_j(z) = \frac{[1 - Q(z)][1 - T'_u(z)]}{1 - Q(z)[1 - L(z)T_u'(z)]}R(z) + \frac{T_u'(z)[Q(z)D_j(z) - D_{j+1}(z)]}{1 - Q(z)[1 - L(z)T_u'(z)]}
\]  

(3.20)

Let \(G_{DE}(z)\) and \(G'_{DE}(z)\) represent the transfer functions from disturbance term to tracking error in the standard ILC scheme and the proposed control scheme, respectively. Rewriting \(G_{DE}(z)\) and \(G'_{DE}(z)\) in more explicit way, we have

\[
G_{DE}(z) = \frac{T_d(z)}{1 - Q(z)[1 - zL(z)T_u(z)]} = \frac{P(z)}{[1 + P(z)C(z)][1 - Q(z)] + Q(z)L(z)P(z)}
\]  

(3.21)

\[
G'_{DE}(z) = \frac{T'_d(z)}{1 - Q(z)[1 - zL(z)T_u'(z)]} = \frac{P(z)(1 - z^{-1}Q_D(z))}{[1 + P(z)C(z) + z^{-1}Q_D(z)\Delta(z)][1 - Q(z)] + Q(z)L(z)P(z)}
\]

By setting \(Q(z) = 1\), \(G_{DE}(z)\) and \(G'_{DE}(z)\) become

\[
G_{DE}(z) = \frac{1}{L(z)}
\]  

(3.22)

\[
G'_{DE}(z) = \frac{(1 - z^{-1}Q_D(z))}{L(z)}
\]  

(3.23)
From (3.22), disturbance attenuation can be achieved by designing the learning function, $L(z)$. As stated, the design of the learning function is restricted by the stability consideration. On the other hand, the proposed control scheme gives us additional control degree of freedom to attenuate the disturbances by designing the learning function as well as $Q_D(z)$. As can be seen, $Q_D(z)$ is designed to be a low-pass filter which makes the $G_D(z)$ close to zero at low frequencies. With the same design of learning functions, disturbance rejection can be improved at least at low frequencies compared to the standard ILC scheme.

### 3.2.1 Results

#### Simulation Results

To understand the enhancement of ILC in the combination with DOB, we simulated the wafer stage system in MATLAB before running experiments. In a linear permanent magnet motor, the main sources of disturbance include nonlinear forces such as friction force, force ripple, and unmodeled uncertainties. We assume nonlinear friction force is negligible as the stage is supported by air bearings. This means that force ripple is the dominant disturbance that causes the tracking error in the constant velocity phase [59]. Disturbances in the system are expected to include force ripple which is regarded as a repetitive disturbance, cable forces, and vibrations which are regarded as non-repetitive disturbances.
Figure 3.8: Performance comparison of ILC alone and ILC in combination with DOB with respect to (a) two norm of tracking errors over iterations and (b) tracking error in the 45th iteration when only repetitive disturbances enter into the system.

Figure 3.8(a) shows the two norm of tracking error over iterations when only repetitive disturbance enters the system. In the simulation, the repetitive disturbance is within the magnitude range $\pm 0.1$. Figure 3.8(b) shows the tracking error in the 45th iteration. DOB in this case shows no benefit on attenuating the repetitive disturbance in the time domain, since ILC alone is capable of compensating repetitive disturbances. This explains why the convergence performance with and without introducing DOB into ILC scheme is the same. Now we intentionally add non-repetitive disturbances within the magnitude range $\pm 0.02$ into the simulation system. Figure 3.2.1 indicates that performance improvement in the combination of ILC and DOB scheme becomes evident after 15th iteration. The performance is improved because ILC learned less information about non-repetitive disturbance which is effectively eliminated by DOB. This leads to a conclusion that introducing DOB into an ILC scheme helps attenuate non-repetitive disturbance resulting in a better tracking performance.

**Experimental Results**

The proposed combined ILC/DOB scheme was experimentally verified on the laboratory wafer stage. Figure 3.10 shows that in absence of ILC, DOB attenuates disturbance in the constant velocity phases and provides better baseline performance. P-type ILC algorithm was then applied to the wafer stage. The tracking error in the 7th iteration and the two norm of tracking error over iterations are plotted and compared in Figures 3.11(a) and 3.11(b). ILC scheme can achieve better
Figure 3.9: Performance comparison of ILC alone and ILC in combination with DOB with respect to (a) two norm of tracking errors over iterations (b) tracking error in the 45th iteration when both repetitive and non-repetitive disturbances enter into the system

performance when combined with DOB which is consistent with what we observed in the simulations. The standard ILC scheme has a worse performance because of non-repetitive disturbances. Without attenuating disturbances in time-domain, the effect of non-repetitive disturbances shows up in the subsequent learning cycles. The proposed control scheme, on the other hand, attenuates the non-repetitive disturbances by DOB in each cycle independently. Consequently, ILC effort mainly contains the feedforward control input for the reference trajectory. As will be presented later, the learned ILC effort which is not subject to disturbances allows us to obtain a proper initial ILC effort for a new trajectory.

3.3 Selection of Initial ILC Effort

A standard ILC algorithm stores the feedforward signal that compensates the plant dynamics and the repetitive disturbances. Consequently, the learned ILC effort is only applicable to a specific trajectory. Each time the reference motion changes, ILC must be reset and relearn a new feedforward signal. The reset ILC algorithm sometimes takes several iterations to let the tracking error converge to an acceptable level. In order to overcome this problem and have fast and robust convergence of learning iterations, a proper selection of initial ILC effort is necessary.

An easy and intuitive way is to determine an initial ILC effort as a plant-inversion based feedforward signal. However, due to the possible change of the load of wafer stage in operations, the
Figure 3.10: Experimental results for comparison of tracking error in the closed-loop system with and without DOB

Figure 3.11: Experimental results for performance comparison of ILC alone and ILC in combination with DOB with respect to (a) two norm of tracking errors over iterations and (b) tracking error in the 7th iteration
effective mass would also change. That is, the parameters of plant model may deviate from its original values. Besides, there may still be parameter uncertainties even if there is no payload variations. The plant-inversion based feedforward control input may deliver biased estimations for an initial ILC effort due to model uncertainties.

Having better estimates of plant parameters allows us to obtain more appropriate initial ILC effort when the trajectory changes. A data-based approach is proposed here to precisely obtain the parameters of the plant model by inspecting the tracking error through experiments. To this end, first we parameterize ILC effort as a linear function of position, velocity, and/or higher derivative terms depending on the complexity of the system. For example, the wafer stage is modeled as a mass-damping system which means that the ILC effort can be duly parameterized by velocity and acceleration.

\[ u_{0}^{ILC} = \hat{m}a_{d} + \hat{b}v_{d} \]  

(3.24)

where \( a_{d} \) and \( v_{d} \) are respectively a lifting vector of the desired acceleration and that of the desired velocity. \( \hat{m} \) and \( \hat{v} \) are the estimates of the effective mass and the damping coefficient. Then parameters can be obtained by means of minimizing the equation below.

\[ \min_{\theta} \| u^{ILC} - \Psi(v_{d},a_{d}) \begin{bmatrix} \hat{m} \\ \hat{v} \end{bmatrix} \| \]  

(3.25)

where \( u^{ILC} \) is a vector of the converged ILC effort learned from the previous experiments. \( \Psi \) is a matrix with velocity and acceleration vectors as columns. As discussed, the learned ILC effort in the combination scheme is less affected by disturbances. (3.25), however, still yields biased estimates because DOB forces the actual plant to behave close to the nominal plant. That is, the ILC effort learns the nominal plant rather than the actual plant when DOB is introduced into the learning scheme.

To be able to approximate the actual plant parameters, a correction term is necessary. This correction term is specifically applicable for the scanning trajectory. The scanning trajectory consists of two major phases (a) acceleration to a fixed velocity, (b) scanning at constant velocity. During the acceleration phase, we expect high frequency components in the reference trajectory, which may result in tracking error due to phase mismatch, especially around the gain crossover frequency. In the constant velocity scan phase force ripple and cable force effects can be expected to cause most of the error. In other words, feedforward control effort in the acceleration phase is highly affected by the effective mass and is mainly dependent on the damping coefficient and disturbances in the constant velocity phase. To remove the estimation bias due to DOB, parameters of the actual plant model are estimated by

\[ \begin{bmatrix} \hat{m} \\ \hat{v} \end{bmatrix} = (\Psi^{T}\Psi)^{-1}\Psi^{T}(u^{ILC} - \delta) \]  

(3.26)

where the correction term is defined as

\[ \delta(k) = \begin{cases} \hat{d}(k) & \text{if } a_{d}(k) \neq 0; \\ 0 & \text{otherwise}. \end{cases} \]  

(3.27)
Figure 3.12: Plot of the 1st reference (a) position, (b) velocity, and (c) acceleration against time index

The purpose of the correction term is to restore the control signal by adding the disturbance estimate particular in the acceleration phase. Then the estimated parameters are utilized to obtain the initial ILC effort by (3.24) for a new trajectory. Furthermore, DOB can also be redesigned based on the estimated parameters.

In the following, simulation results are presented to compare different methods for the selection of initial ILC effort. The parameters of nominal plant and feedback controller are the same as previous settings except for 30% parameter uncertainty for the actual plant model. After ILC is done with learning for a specific trajectory shown in Figure 3.12 and the convergence performance is satisfied, given a new trajectory shown in Figure 3.13, we can calculate a proper initial ILC effort in advance for faster convergence rate. The estimated parameters are also utilized to update DOB for better disturbance rejection.

Figures 3.14 to 3.17 show a series of comparison for five different methods listed below.

- M1: Zeros are selected for an initial ILC effort, and the design of DOB is based on the nominal plant parameters
- M2: Nominal plant parameters are used to obtain an initial ILC effort, and ILC scheme is combined with DOB design based on the nominal plant parameters
- M3: Plant parameters are estimated and updated by (3.25). Then updated plant parameters are used to obtain an initial ILC effort, and the design of DOB is based on the nominal plant parameters.
- M4: Plant parameters are estimated and updated by (3.26). Then updated plant parameters are used to obtain an initial ILC effort, and the design of DOB is based on the nominal plant parameters
• M5: Plant parameters are estimated and updated by (3.26). Then updated plant parameters are used to obtain an initial ILC effort, and the design of DOB is based on the updated plant parameters.

Figure 3.15 shows that the ILC scheme converges only in few iterations with the proper selection of an initial ILC effort. As expected, the first method has slowest convergence rate since the learning algorithm takes several iterations to relearn the new trajectory. It should be noted also that the first method ends up with larger tracking error. The second method intuitively selects the plant-inversion feedforward signal for an initial ILC effort. The third method is data-based approach that estimates the plant parameters from the learned control effort and then utilizes them to obtain the plant-inversion feedforward signal for initializing ILC effort. The second and third methods both significantly reduce transient tracking error in the first iteration resulting in faster convergence rate. Due to the effect of DOB which makes the actual plant act like nominal plant model, the parameters estimated by the third method are similar to the nominal parameters. Therefore, results yielded from the second and third methods are indistinguishable.

Then a correction term is proposed to eliminate the estimation bias as described in the fourth method. Compared to the second and the third methods, the fourth method converges to better tracking performance as shown in Figure 3.16. The zoomed-in detail of the tracking error in the 45th iteration shows that the peak tracking error in the acceleration phase is reduced by the use of the fourth method. However, the fourth method yields larger tracking error in the first iteration as shown in Figure 3.15 resulting in a slower convergence rate shown in Figure 3.14. The reason for this is that the corrected parameters are only used for the selection of initial ILC effort, but DOB is still designed based on the original nominal plant parameters. Without updating the design of DOB, the ILC needs several iterations to converge. Therefore, the fifth method is proposed to use the corrected parameters for initial ILC effort selection as well as DOB design. As can be seen,
the fifth method achieves the best tracking performance in the first iteration and requires only few iterations to converge. Figure 3.16 indicates that the fifth method not only reduces the peak error in the acceleration phase but also makes the effect of disturbances on tracking error small in the constant velocity phase. In other words, performance in terms of disturbance rejection is improved by the use of the fifth method because DOB has more accurate plant model. Figure 3.17 is a plot of the learned control efforts for the five methods. As is clear from this figure, the ILC effort is less affected by disturbances when DOB can efficiently eliminate disturbances. From the above simulation studies, a conclusion can be drawn that a proper initial ILC selection provides faster convergence to lower tracking error.

3.4 Chapter Summary

In this chapter, the concept of attenuating disturbances in time domain by means of DOB was addressed. It is an effective strategy for improving performance of an ILC system that is subject to non-repetitive disturbances. In previous research, ILC algorithms had to balance the trade-off between convergence performance and attenuation of non-repetitive disturbances. The combined control scheme of ILC and DOB unnecessitated such trade-off. In particular, this work examines the effect of non-repetitive disturbances and imperfect resetting on ILC systems. Additional investigation focuses on how a good design procedure, ILC in the combination with DOB, can mitigate these effects. Stability and performance analysis indicated that we can attenuate non-repetitive disturbances without loss of ILC performance. Another advantage of introducing DOB into ILC scheme is that the learned ILC effort becomes mainly dependent on trajectory following. Therefore, an initial ILC effort for a new trajectory can be obtained without adverse effect. Both
Figure 3.15: Errors in the 1st iteration for tracking the 2nd trajectory. (a) The tracking error against time index and (b) zoomed-in detail of the error plot.

Figure 3.16: Errors in the 45th iteration for tracking the 2nd trajectory. (a) The tracking error against time index and (b) zoomed-in detail of the error plot.
Figure 3.17: Plot of (a) the learned control effort and (b) zoomed-in detail in the 45th iteration.

Simulation results and experimental results demonstrated the effectiveness of the proposed control scheme.
Chapter 4

Iterative Learning Control with Optimal Feedback and Feedforward Control

In the previous Chapter, it was stated that iterative learning control (ILC) is a feedforward control strategy used to improve the performance of a system that executes the same task repeatedly, but that it is incapable of compensating for non-repetitive disturbances and stabilizing the system. Thus a well-designed feedback controller must be used in combination with ILC. A PID feedback controller was applied to stabilize the system, achieve basic performance requirement, and make the system robust to disturbance. Disturbance observer has also been introduced into an ILC system for further attenuation of disturbance effect. The performance of ILC, however, is still affected by the design of Q-filters and learning functions. The Q-filter is essential for the stability of ILC systems. The price to pay for stability enhancement by Q-filter is deterioration of performance for tracking the repetitive reference and cancellation of repetitive disturbances. To circumvent this, a pre-designed feedforward control input is applied in addition to ILC. Shortly stated, a well-designed feedback controller is important, and a well-designed feedforward controller further improves performance.

This chapter focuses on developing optimal feedback-feedforward control to be used in combination with a P-type ILC. The whole design procedure will be kept simple but the controller achieves promising performance and robustness. The optimization is inspired by predictive control because of its properties of pre-evaluating the system behavior to improve the performance and robustness against model-mismatching. It will be shown that the choice of the injection point of the learned ILC effort in this combination scheme is crucial for a tradeoff between stability and performance. Therefore, the stability and performance analysis based on different injection points are studied. A design procedure for the combination scheme is also proposed. The effectiveness of the proposed method is verified by simulation.

This chapter is organized as follows. Section 4.1 derives an optimal feedback-feedforward control law by procedure similar to classic predictive control. In Section 4.2, a combination scheme of ILC and optimal closed-loop control is proposed along with discussion of different injection points for the ILC effort. Section 4.3 presents a design procedure for the combination scheme.
Finally, simulation results for verifying performance improvement are presented in Section 4.4.

### 4.1 Model Predictive Control

Model predictive controllers rely on dynamic models of plants to predict the future output behavior of dynamical systems. This prediction capability allows us to solve the optimal control problem to minimize the difference between the predicted output and the desired reference over a future horizon.

Let us first consider a linear time-invariant discrete system described in state-space form

\[
\begin{align*}
\mathbf{x}(k+1) &= A\mathbf{x}(k) + B[\mathbf{u}(k) + \mathbf{d}(k)] \\
\mathbf{y}(k) &= C\mathbf{x}(k)
\end{align*}
\]  

(4.1)

where \(\mathbf{x}(k) \in \mathbb{R}^n\) is a vector of system states, \(\mathbf{u}(k) \in \mathbb{R}^m\) is a vector of control inputs, \(\mathbf{d}(k) \in \mathbb{R}^m\) a vector of disturbances, and \(\mathbf{y}(k) \in \mathbb{R}^p\) is a vector of system outputs. In order to incorporate integral control, the model (4.1) is transformed into an incremental state-space model. The integral action is desirable because it can provide a stronger smoothing action to avoid sudden jumps of the manipulated variables. The incremental state-space model is expressed as

\[
\begin{bmatrix}
\Delta\mathbf{x}(k+1) \\
\Delta\mathbf{y}(k+1)
\end{bmatrix} =
\begin{bmatrix}
A & 0 \\
CA & I_m
\end{bmatrix}
\begin{bmatrix}
\Delta\mathbf{x}(k) \\
\Delta\mathbf{y}(k)
\end{bmatrix} +
\begin{bmatrix}
B \\
CB
\end{bmatrix}
\begin{bmatrix}
\Delta\mathbf{u}(k) + \Delta\mathbf{d}(k)
\end{bmatrix}.
\]

(4.2)

where \(\Delta\) denotes the increment of a variable from one time step to the next. For example, \(\Delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1)\) and \(\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)\). For brevity, the augmented states and matrices are expressed as

\[
\mathbf{x}_e(k) = \begin{bmatrix} \Delta\mathbf{x}(k) \\ \mathbf{y}(k) \end{bmatrix},
\mathbf{A}_e = \begin{bmatrix} A & 0 \\ CA & I_m \end{bmatrix},
\mathbf{B}_e = \begin{bmatrix} B \\ CB \end{bmatrix},
\mathbf{C}_e = \begin{bmatrix} 0_n & I \end{bmatrix}
\]

(4.3)

With given \(\mathbf{x}_e(k)\), the \(i\)-step ahead prediction of the output signal, denoted as \(\hat{\mathbf{y}}(k+i|k)\), can be calculated based on the incremental state-space model \((\mathbf{A}_e, \mathbf{B}_e, \mathbf{C}_e)\).

\[
\hat{\mathbf{y}}(k+i|k) = \mathbf{C}_e\mathbf{A}_e^i\mathbf{x}_e(k) + \sum_{l=1}^{i} \mathbf{C}_e\mathbf{A}_e^{i-l}\mathbf{B}_e\Delta\mathbf{u}(k+l-1)
\]

(4.4)

Sequentially calculating (4.4) from \(i = 1\) to \(i = N_p\), a vector of output predictions \(\hat{\mathbf{Y}}_{k,N_p}\) at time index \(k\) can be written in a compact form

\[
\hat{\mathbf{Y}}_{k,N_p} = \mathbf{F}\mathbf{x}_e(k) + \Phi[\Delta\mathbf{U}_{k,N_p} + \Delta\mathbf{D}_{k,N_p}]
\]

(4.5)
where
\[
\begin{align*}
\dot{Y}_{k,N_p} &= \begin{bmatrix} \dot{y}(k+1|k) & \dot{y}(k+2|k) & \cdots & \dot{y}(k+N_p|k) \end{bmatrix}^T, \\
\Delta U_{k,N_c} &= \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \cdots & \Delta u(k+N_c-1) \end{bmatrix}^T, \\
\Delta D_{k,N_c} &= \begin{bmatrix} \Delta d(k) & \Delta d(k+1) & \cdots & \Delta d(k+N_c-1) \end{bmatrix}^T, \\
F &= \begin{bmatrix} C_eA_e & C_eA_e^2 & \cdots & C_eA_e^{N_p} \end{bmatrix}, \\
\Phi &= \begin{bmatrix} C_eB_e & 0 & \cdots & 0 \\
C_eA_eB_e & C_eB_e & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
C_eA_e^{N_p-1}B_e & C_eA_e^{N_p-2}B_e & \cdots & C_eA_e^{N_p-N_c}B_e \end{bmatrix},
\end{align*}
\]
and \(N_p\) is prediction horizon determining the length of the optimization window. \(N_c\) is control horizon dictating the number of parameters used to capture the future control trajectory. Control inputs, \(u(i)\), will be held constant for \(i \geq k + N_c\). In general, \(N_p \geq N_c\) is required. It should be noted that (4.6c) is realizable when \(N_c\) time step prediction of disturbances is possible. In the sequel, the subscripts of \((k,N_p)\) and \((k,N_c)\) will be omitted when this does not lead to any confusion.

With a given reference trajectory \(r(k) \in \mathbb{R}^p\), the optimal control increments can be determined to minimize the performance index,
\[
J = (R - Y)^T Q_e (R - Y) + \Delta U^T Q_u \Delta U
\]
where \(Q_e = \text{diag}\{\lambda_e\} \succ 0\) and \(Q_u = \text{diag}\{\lambda_u\} \succeq 0\) are the error weighting matrix and control weighting matrix, respectively. \(R\) is a vector of reference trajectory
\[
R = \begin{bmatrix} r(k+1) & r(k+2) & \cdots & r(k+N_p) \end{bmatrix}^T
\]
Then using (4.5) and finding the stationary point, the vector of the optimal control increments can be found as
\[
\Delta U = (\Phi^T Q_e \Phi + Q_u)^{-1} \Phi^T Q_e (R - Fx_e(k) - \Phi \Delta D)
\]
Note that the vector of the optimal control increments, \(\Delta U\), contains the control input from \(\Delta u(k)\) to \(\Delta u(k+N_c-1)\). The optimization is performed according to the receding horizon approach. That is, only the first sample of the optimal input sequence is actually applied to the plant and the remaining optimal inputs are discarded. Then the control increments at time \(k\) can be rewritten as:
\[
\Delta u(k) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \Delta U \equiv K_r R - K_x x_e(k) - K_d \Delta D
\]
where \(K_r \in \mathbb{R}^{m \times N_p}\), \(K_x \in \mathbb{R}^{m \times n}\), and \(K_d \in \mathbb{R}^{m \times m}\) are constant matrices. The optimal control input obtained above consists of optimal state feedback control and feedforward control on the reference and disturbance. It should be noted that when disturbance is not measurable and/or predictable in advance, the disturbance term can be ignored in the design resulting in \(K_d = 0\). Then, the structure reduces to the optimal state feedback control and reference feedforward (preview) control. In our application, disturbance is not measurable so the disturbance compensation part is omitted.
4.2 ILC Combined with The Optimal Feedback-Feedforward Control System

As mentioned, the P-type ILC algorithm uses the error signal advanced by a few steps to update the learning control effort. This is illustrated by designing $L(z) = \alpha z^{nd}$ in which error signal is advanced by $n_d$ steps. For stability of the P-type learning algorithm, $n_d$ is chosen corresponding to the delay of the system. P-type ILC is attractive because of its simplicity and robustness. The tracking error, however, becomes difficult to handle by this simple design of the learning function when peak accelerations become larger. For example, a typical scanning operation requires the wafer stage to accelerate to constant velocity as soon as possible and maintain this velocity for the length of the scan. During the acceleration and deceleration phases, we expect high frequency components in the reference trajectory, which may result in large tracking errors due to phase mismatch.

To reduce peak errors especially in the acceleration phase, we propose a combined control scheme in which optimal feedback-feedforward control is used in combination with P-type ILC. This section will revisit the optimal solutions for control inputs to apply on a wafer scanner system, and then discuss the efficient injection points for combining with ILC.

From the model of the wafer stage described in (3.6), the associated augmented states as developed in (4.3) can be expressed in a more explicit way

$$x_e(k) = \begin{bmatrix} x_1(k) - x_1(k-1) \\ x_2(k) - x_2(k-1) \\ y(k) \end{bmatrix} = \begin{bmatrix} (1-z^{-1})x_1(k) \\ (1-z^{-1})x_2(k) \\ y(k) \end{bmatrix}$$

where the output $y(k) = x_1(k)$ is the position measured by the laser interferometer. Then the optimal state feedback gain in (4.10) becomes a 1 by 3 vector, and is denoted as $K_x = [K_{x,1} K_{x,2} K_{x,3}]$. The velocity term, $x_2(k)$, is approximated by backward difference, which takes the difference between current and previous position measurements, divided by sampling time. The optimal state feedback control law in (4.10) can be rewritten as

$$K_x x_e(k) = [K_{x,1}(1-z^{-1}) + \frac{K_{x,2}}{T_s} (1-z^{-1})^2 + K_{x,3}] y(k)$$

Substituting (4.12) into (4.10), the optimal control input in the $j$ iteration can be rewritten as

$$u_j(k) = \frac{\sum_{i=1}^{N_p} K_r(i) z^{i-1}}{1-z^{-1}} r(k) - [K_{x,1} + \frac{K_{x,2}}{T_s} (1-z^{-1}) + \frac{K_{x,3}}{1-z^{-1}}] y_j(k)$$

Note that the first term on the right-hand side of (4.13) represents feedforward control which includes preview of the reference signal over $N_p$ steps into the future. As it can be imagined, the feedforward part has objective similar to the ILC algorithm: taking future information to improve the performance and to reduce the transient response. The optimal feedforward control obtained
here predicts the future output of the system and compares it to the future reference. Note that previewed future is deterministic as opposed to probabilistic. In contrast to the optimal feedforward control, ILC adjusts feedforward signals based on the error information gathered from previous iterations. ILC foresees the system behavior by assuming that the system behavior is invariant from previous iterations. Therefore, it makes sense to combine the optimal feedback-feedforward control scheme and ILC algorithms. It will be shown later that the combined control scheme is outstanding because of its simplicity, robustness, and performance.

The block diagram of ILC in combination with the optimal feedback-feedforward control is shown in Figure 4.1. It also suggests two injection points for combining the learned ILC effort into the optimal control scheme. The two different injection points are labeled A and B, respectively in Figure 4.1. From the figure, transfer functions from the ILC injection point to the output are obtained as

\[
\text{Case A : } \ T_u^A(z) = \frac{P(z) \sum_{i=1}^{N_p} K_r(i) z^{-i}}{1 + P(z) C_{PID}^{opt}(z) 1 - z^{-1}} \tag{4.14}
\]

\[
\text{Case B : } \ T_u^B(z) = \frac{P(z) 1}{1 + P(z) C_{PID}^{opt}(z) 1 - z^{-1}} \tag{4.15}
\]

where

\[
G_{ff}(z) = \frac{\sum_{i=1}^{N_p} K_r(i) z^{-i}}{1 - z^{-1}} - C_{PID}^{opt}(z) \tag{4.17}
\]

\[
C_{PID}^{opt}(z) = K_{x,1} + \frac{K_{x,2}}{T_s} (1 - z^{-1}) + \frac{K_{x,3}}{1 - z^{-1}} \tag{4.18}
\]

Then, the optimal control input can be expressed as

\[
u(k) = G_{ff}(z) r(k) + C_{PID}^{opt}(z) e(k) \tag{4.19}\]

As can be seen, Case A is inherently the same as the serial architecture shown in Figure 1.4(b) and, Case B is similar to the parallel architecture shown in Figure 1.4(a). In Case B, the integral action on the learned ILC effort is indispensable for avoiding interference with the existing close-loop system. This makes sense since the closed-loop controller is designed based on the incremental model rather than the original model.

Recall stability analysis of ILC systems presented in section 2.2.1. The stability conditions for the two cases are

\[
\text{Case A : } \|Q^A(z)[1 - L^A(z) T_u^A]\|_\infty < 1 \tag{4.20}
\]

\[
\text{Case B : } \|Q^B(z)[1 - L^B(z) T_u^B]\|_\infty < 1 \tag{4.21}
\]
where $Q^A(z)$ and $Q^B(z)$ are Q-filters and $L^A(z)$ and $L^B(z)$ are learning functions for Case A and Case B, respectively. Figure 4.2 plots the stability conditions for the two cases with different learning gains. In the plots, $Q^A(z) = Q^B(z) = 1$ is selected, and $N_p = N_c = 10$ and $\lambda_u = 10^{-9}$ are chosen. Then, the resulting optimal feedback control is

$$C_{opt_{PID}}(z) = 95977 + \frac{20482}{1 - z^{-1}} + \frac{290}{T_s} (1 - z^{-1}).$$

(4.23)

In Case A, $L^A(z) = \alpha^A z$ where the learning gain, $\alpha^A$, varies from 0.5 to 1.5 with increment 0.25. As for Case B, $L^B(z) = \alpha^B z$ where the learning gain, $\alpha^B$, varies from $0.5K_R$ to $1.5K_R$ with increment $0.25K_R \ (K_R = \sum_{i=1}^{N_p} K_r(i))$. Note that the learning gain in Case B is set higher than that in Case A for a fair comparison, since the ILC injection point B bypasses the feedforward control.

The frequency response shows that the combined control scheme offers promising stability properties since the inclusion of integral action in the controller enlarges the stability margin at low frequencies. Hence, both cases allow Q-filters to be unity at low frequencies without violating the stability conditions. More precisely, the stability condition of Case B is satisfied at almost all frequencies. As opposed to Case B, the stability condition of Case A is violated above 40 rad/s frequencies. Thus the magnitude of Q-filter for Case A needs to be less than one beyond that frequency. This leads to a conclusion that Case B is capable of achieving better performance since Case B allows higher cutoff frequency of Q filter. The details of designing Q-filters for both cases will be presented later in this chapter. Moreover, the disturbance rejection in Case B is more efficient than that in Case A as shown in Figure 4.2.

### 4.3 A Systematic Design Procedure

With given the system model, the systematic procedures to design a P-type ILC in combination with the optimal feedback-feedforward controllers are listed in Table 4.1. Initially, we need to the
Figure 4.2: Comparison on stability robustness for (a) Case A and (b) Case B with different learning gains (The ILC system is stable if the magnitude is under 0 db for all frequencies)

Figure 4.3: Performance comparison in terms of disturbance rejection ($\| \frac{T_d(z)}{1 - Q(z)\left[1 - L(z)T_d(z)\right]} \|$ with $Q(z)=1$ and $L(z) = \alpha \bar{z}^{\rho_d}$) for (a) Case A and (b) Case B
Table 4.1: Systematic design procedure for a P-type ILC combined with the optimal feedback-feedforward controllers

<table>
<thead>
<tr>
<th>No.</th>
<th>Step</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Determine $N_p$ and $N_c$</td>
<td>$N_p$ larger than the rise time of the plant is suggested. ($N_c \leq N_p$)</td>
</tr>
<tr>
<td>2.</td>
<td>Set $\lambda_e = 1$ and tune $\lambda_u$</td>
<td>$0 &lt; \lambda_u &lt; 1$</td>
</tr>
<tr>
<td>3.</td>
<td>Calculate $K_r$ and $K_x$</td>
<td>Back to Step 2 until the closed-loop system has good behaviour</td>
</tr>
<tr>
<td>4.</td>
<td>Determine $L(z) = \alpha z^{n_d}$</td>
<td>$n_d$ is system delay and $\alpha = f_{safe} \max(|T_u(z)|<em>1)$ where $f</em>{safe} \approx 0.8$</td>
</tr>
<tr>
<td>5.</td>
<td>Design Q-filter</td>
<td>A zero-phase filter with cutoff frequency equal to $(0.2 \sim 0.5) \times \frac{1}{\tau}$</td>
</tr>
</tbody>
</table>

determine values of the prediction horizon and control horizon. In our system, the selection of $N_p$ and $N_c$ has minor effect on the computed optimal control gains as long as $N_p$ is close to the rise time of the plant. Setting $N_p = N_c = 10$ and $\lambda_e = 1$, the optimal control gains ($K_r$ and $K_x$) were computed for selected value of $\lambda_u$. Note that setting $\lambda_e = 1$ makes sense because the optimal solution for the performance index in (4.7) depends only on the ratio of $\lambda_e$ and $\lambda_u$. As $\lambda_u$ decreases which means smaller weighting on control input, the tracking performance is expected to be better but the computed control input may exceed the saturation control value resulting in unstable closed-loop system. Therefore, we started with a conservative value $\lambda_u = 0.01$ and decreased the value until the closed-loop system has good behavior. The learning gain ($\alpha$) for P-type ILC can be easily determined from the stability point of view. Based on the stability condition, $\alpha$ has to be smaller than the reciprocal of the maximum magnitude of $T_u(z)$. As shown in Table 4.1, $\alpha$ equals to the reciprocal of the maximum magnitude of $T_u(z)$ times a safety factor ($f_{safe}$) for providing robustness to model uncertainty.

The last step is to determine the cutoff frequency of the Q-filter. As already shown, an ideal Q-filter is unity for perfect tracking performance. Normally, a sensible choice is to let $Q(z)$ close to 1 at low frequencies for performance consideration, and let $Q(z)$ close to 0 at high frequencies for robustness. Since all error information is accessible and gathered from previous iterations, Q-filter is typically designed as a zero-phase low-pass filter.

Figure 4.4 shows a comparison of the maximum-norm error over iterations in Case A (dash line) and Case B (solid line with square markers) with different cutoff frequencies of Q-filters. With fixed learning gains, Q-filter has minor effect on the convergence rate of the error, but it determines the magnitude of the converged error and stability of the system. In other words, increased bandwidth improves the error but may result in unstable learning. From the figure, we observe that the learning algorithm in Case A with cutoff frequency at 400Hz diverges. On the other hand, the learning algorithm in Case B is capable of pushing the cutoff frequency of Q-filter to higher frequencies without violating the stability condition. As a result, Case B can achieve better convergence performance as compared to Case A. Based on the Figure 4.4, a 300-Hz cutoff frequency is chosen for the Q-filter in Case A and a 400-Hz cutoff frequency is chosen for the
Figure 4.4: Maximum-norm error over iterations with different cutoff frequencies of Q-filters

Q-filter in Case B.

So far, we have developed a design procedure of P-type ILC with optimal feedback-feedforward control for a wafer scanner system. Since Q-filter is not unity over all frequencies, it impossible to let the tracking error approach to zero even if non-repetitive disturbance is absent. Thus the optimal feedforward control plays a central role for performance improvement. With given $N_p$-step preview of the reference input, the optimal feedforward control improves the transient tracking error by weighing on the reference signals. To demonstrate the effectiveness of adding optimal feedforward control in addition to ILC, a simulation study is conducted. Note that the following discussions and figures are all evaluated and performed in the simulation environment which includes repetitive disturbance and non-repetitive disturbances. For the values of the parameters in the simulation model, we refer to Section 3.2.1.

Figure 4.5 compares tracking errors in the 25th iteration applying the same optimal feedback control and ILC algorithm but with and without applying the optimal feedforward control ($G_{ff}(z)$). It points out that adding the pre-designed optimal feedforward controller into system has superior peak error attenuation, especially in the acceleration phase. Figure 4.6 shows another comparison of the maximum-norm error under the three different feedforward strategies: no feedforward control, optimal feedforward control ($G_{ff}(z)$), and plant-inversion feedforward control ($P(z)^{-1}$). Of the three feedforward strategies, adding the plant-inversion feedforward control (PI-FF) results in the smallest tracking error in the first iteration. This result can be explained by considering their
Figure 4.5: Performance comparison of P-type ILC with and without optimal feedforward control in terms of tracking errors in the 25th iteration

transfer functions from reference trajectory to output.

\[
\text{Opt. FB only: } \quad \frac{P(z)C_{\text{opt}}^\text{PID}(z)}{1 + P(z)C_{\text{opt}}^\text{PID}(z)}
\]  

\[
\text{Opt. FB-FF: } \quad \frac{P(z)\sum_{i=1}^{N_p} K_r(i)z^{-i}}{1 + P(z)C_{\text{opt}}^\text{PID}(z)}
\]  

\[
\text{Opt. FB + PI-FF only: } \quad \frac{P(z)C_{\text{opt}}^\text{PID}(z) + P_n^{-1}(z)P(z)}{1 + P(z)C_{\text{opt}}^\text{PID}(z)}
\]

The transfer functions shown above need to be as close to one as possible for achieving better trajectory tracking performance. As can be seen, the transfer function for plant-inversion feedforward control is closest to one if the model uncertainties are minor. On the other hand, the transfer function for optimal feedforward control is not unity, a reason for which is the finite time steps preview of the reference signal (\(\sum_{i=1}^{N_p} K_r(i)z^{-i}\)). Adding the optimal feedforward controller, however, provides with the best converged performance of peak error elimination. The explanation for this is that P-type ILC only takes one step information ahead for computing the control input, but the optimal feedforward control previews 10 steps ahead of reference signal for improving the transient error. This leads to the conclusion that adding feedforward control to P-type ILC reduces peak error. Thus a well-designed feedforward control in combination with P-type ILC is important for performance improvement.
4.4 Comparison of Different ILC Algorithms

This section will present performance comparisons of different ILC algorithms such as P-type ILC, plant-inversion ILC (PI-ILC), and optimization-based ILC by gradient descent method (GD-ILC) in combination with PID controller versus the proposed optimal feedback-feedforward controller. We are interested in comparing these algorithms in terms of the number of ILC iterations required to converge to steady state error, the maximum-norm error at steady state, and the two-norm error. Design details for P-type ILC combined with the optimal feedback-feedforward controller have been discussed in the preceding section. An integral action on the learned ILC effort is essential to the P-type ILC. PI-ILC and GD-ILC, however, directly take the incremental model into account. That is, there is no need to have the integral action on the learned ILC effort. The learning functions for PI-ILC algorithm combined with PID feedback controller and the optimal feedback-feedforward controller are given as

\[
L_{PID}^{PI}(z) = \frac{1 + P(z)C_{PID}(z)}{P(z)}
\]

\[
= \frac{1 - 2.546z^{-1} + 2.597z^{-2} - 1.432z^{-3} + 0.4001z^{-4}}{z^{-1}(1.112e^{-6} - 5.033e^{-10}z^{-1} - 1.111e^{-6}z^{-2})}
\]

\[
L_{Opt}^{PID}(z) = \frac{1 + P(z)C_{Opt}^{PID}(z)}{P(z)}
\]

\[
= \frac{1 - 2.546z^{-1} + 2.697z^{-2} - 1.428z^{-3} + 0.3229z^{-4}}{z^{-1}(1.112e^{-6} - 5.033e^{-10}z^{-1} - 1.111e^{-6}z^{-2})}
\]

respectively. One may notice that the PI-ILC algorithms shown in (4.27) and (4.28) are noncausal. Differing from the usual notion of non-causality, the noncausal learning algorithm is realizable.
in practice because the entire error information has been gathered from the previous iteration in advance. A zero-phase low-pass Q-filter is essential to the PI-ILC algorithm due to the model uncertainty.

As for the optimization-based ILC, the gradient descent method is adopted to iteratively approach to the optimal ILC effort. In the gradient method, an iterative learning update law is of the form

$$u_{ILC}^{j+1} = u_{ILC}^j + \gamma_j g_j$$

where $g_j$ is the search direction and $\gamma_j$ is called the step size in the $j^{th}$ iteration. Recall the optimization-based ILC introduced in Chapter 2. The cost function can be

$$J(u_{ILC}^{j+1}) = \frac{1}{2} e_j^T e_j$$

where

$$e_j = r - y_j$$

$$= (I - T_r)r - Tu_{ILC}^j - T_d d_j$$

The search direction determined by the gradient method is chosen to be along the gradient of the cost function, i.e.,

$$g_j = -\nabla J(u_{ILC}^j)$$

$$= Tu^T [(I - T_r)r - T_d d_j] + Tu^T u_{ILC}^j$$

$$= Tu^T e_j$$

Then the step size is obtained by solving $\frac{d J(\gamma_j)}{d \gamma_j} = 0$. That is,

$$\frac{d}{d \gamma_j} [(I - T_r)r - T_d d_j - Tu(u_{ILC}^j + \gamma_j g_j)]^T [(I - T_r)r - T_d d_j - Tu(u_{ILC}^j + \gamma_j g_j)] = 0$$

After some manipulations, we obtain

$$\gamma_j = \frac{g_j^T g_j}{g_j^T Tu^T u_{ILC}^j}$$

From Eq. (4.29), the update law for the GD-ILC algorithm is given as

$$u_{ILC}^{j+1} = u_{ILC}^j + \frac{g_j^T g_j}{g_j^T Tu^T u_{ILC}^j} Tu^T e_j$$

We refer the reader to [30] for the detailed proof of the convergence of the update law given above.
Figure 4.7: Performance comparison of different types of ILC algorithms in combination with (a) PID feedback control and (b) with optimal feedback-feedforward control in terms of the maximum-norm error. Assume the applied system model to have 10% parameter uncertainty.

Simulations are conducted for comparisons of three types of ILC algorithms combined with two different feedback controls applied to system with 10% parameter uncertainty. In combination with PID feedback control, three types of ILC algorithm are all capable of achieving steady maximum norm error around $1e^{-6}$ and steady two norm error around $2e^{-5}$ as shown in Figure 4.7(a) and Figure 4.8(a). Among them PI-ILC has fastest convergence rate. Figure 4.7(b) and Figure 4.8(b) show the performance comparisons of different ILC algorithms combined with optimal feedback-feedforward control. In this combination scheme, P-type ILC and PI-ILC algorithms achieve lower converged errors (steady maximum norm error around $2e^{-7}$ and steady two norm error around $4e^{-6}$) as opposed to that in combination with PID control scheme. GD-ILC, however, converges to the same error level no matter which feedback controller is combined with.

To investigate the robustness of each type of ILC algorithms, ILC algorithms combined with PID feedback control and optimal feedback-feedforward control are separately applied to the system with multiplicative modeling uncertainty. Similar to Chapter 3, assume the multiplicative modeling uncertainty to be actuator dynamics, $\frac{5Ts}{5Ts+1}$. Figure 4.9 shows that the two norm of tracking errors in the PI-ILC scheme increases over iterations both in combination with PID feedback control and with optimal feedback-feedforward control. The divergence of learning process is caused by the plant model uncertainties. GD-ILC algorithm is stable in combination with PID feedback control but is unstable in combination with optimal feedback-feedforward control. P-type ILC, on the other hand, is stable for combining with both feedback controls. As can be seen in Figure 4.9, P-type ILC in combination with our proposed optimal feedback-feedforward control is...
most robust to modeling uncertainties and capable of achieving a promising tracking performance.

4.5 Chapter Summary

An optimal feedback-feedforward control combined with P-type ILC for two different injection points was presented and studied. The optimization is inspired by classic predictive control and determines the optimal feedback gains as well as optimal feedforward gains. Two contributions are made: i) the injection point of learned ILC effort was studied based on the optimal feedback-feedforward control configuration which inherently includes an integrator for achieving promising performance and ii) a design procedure for the proposed control scheme was established. We also found that applying a pre-designed optimal feedforward (preview) control in addition to ILC is effective to reduce the tracking error.

The proposed control scheme is attractive because of its simplicity, easy tuning, and performance improvement comparable to plant-inversion ILC and optimization-based ILC in combination with PID feedback control. Adding the optimal feedforward control is suitable for utilizing a simple P-type ILC algorithm without compromising performance due to its preview action on the reference signal. Furthermore, P-type ILC in combination with proposed optimal feedback-feedforward control is robust to modeling uncertainties. The effectiveness of the proposed control scheme was demonstrated by simulation.
Figure 4.9: Performance comparison of different types of ILC algorithms in combination with (a) PID feedback control and (b) with optimal feedback-feedforward control in terms of two norm of the tracking error. Assume the applied system to have multiplicative modeling uncertainty.
Chapter 5

ILC Design for Synchronization

Whereas the previous chapters has focused on control of a single wafer stage, subsequent research is expanded to the control of a wafer stage and a reticle stage system. In addition to higher precision positioning requirement, another stringent requirement of photolithography process is the ability to align precisely a pattern being exposed on the wafer. In the process, the step-and-scan method does not expose the entire pattern by moving single stage, but instead use a synchronized scanning of the wafer and the reticle through a fixed slit within the associated optics. The wafer stage and reticle stage work together to position the wafer and reticle with respect to each other so that a circuit can be formed on the wafer. The reticle and wafer stages are both driven to increase exposure length. In the scanning phase, the stages must move at a constant velocity. The wafer and reticle stages scan in opposite directions with speed ratio, 4, predetermined by a projection magnification of the optical projection system [31]. This mode of operation allows large patterns to be shrunk to small sizes on the wafer through a lens system. To this end, this chapter is devoted to develop an ILC algorithm applied in the synchronization framework for further improvement of synchronization error.

This chapter is organized as follows. Section 5.1 introduces the concept and the importance of synchronization controls. Section 5.2 presents two commonly applied synchronization control approaches. Properties of each synchronization control approach are also studied by simulation and experimentation. In Section 5.3, an ILC scheme for synchronization is proposed to reduce synchronization error while minimizing the individual axes tracking error. Experiential results are obtained to validate the effectiveness of the proposed synchronization ILC scheme.

5.1 Introduction

To reduce synchronization error, an indirect approach by increasing the position tracking accuracy of individual axes is commonly used. However, the scanning process requires that the two stages be synchronized to within stringent accuracy. Therefore, not only do we wish to minimize the tracking errors of the individual stages, but we also wish to minimize the synchronization error between the
two. Since synchronization of the two stages is important, having individual axes control around each of the stages is insufficient. The development of control architectures to accomplish stage alignment is essential. For meeting CD control and overlay requirements, the positioning errors of both stages must be within 1 nm, and the synchronization error between the two stages must be within 10 nm.

Two common synchronization approaches are the "cross-coupled" approach [25] and the "master-slave" approach [44]. In this chapter, these two synchronization control approaches are first surveyed and presented. The first approach, cross-coupling control (CCC), improves the synchronization error by coupling the individual axes tracking errors together. The second approach, master-slave control, assigns one stage as the master and the other as the slave. The master simply follows its desired trajectory, and the slave tracks the motion of the master.

As will be seen later, these two synchronization methods are typically developed in feedback frameworks. Feedback control alone, however, has limitation on improving performance. Thus a feedforward control is required in addition to synchronization feedback control. Considering the repetitive nature of scanning, a possible solution to improve synchronization of the two stages is to incorporate the idea of synchronization methods into the design of ILC algorithms.

5.2 Synchronization Control Scheme

Two advanced synchronization control approaches suitable for the control of wafer scanner systems will be described in the following sections: CCC and master-slave control. To investigate the properties of each approach, simulations and experiments are also conducted.

5.2.1 Cross-Coupled Control

In classic feedback control of multi-axes systems, controllers are usually independently designed for each axis without consideration of motions of other axes. This results in a decoupled system which makes the disturbance in one axis have no direct effect on the control input of other axes. However, for synchronization applications, decoupling sometimes degrades the overall performance. Cross-coupled control (CCC) has been shown to be effective in improving the synchronization accuracy.

Figure 5.1 illustrates the overall CCC scheme specific for the wafer and reticle stages. The main idea behind CCC is to allow each axis to share the errors of two axes. In addition to feedback of its own tracking error, each axis also takes the synchronization error into account. Here the synchronization error denoted by \( \epsilon^\perp \) in Figure 5.2 is defined as the shortest distance between the actual position and the desired path. The overall system can be expressed as

\[
y_r(k) = P_r(z)C_r(z)e'_r(k) + P_r(z)d_r(k) \tag{5.1}
y_w(k) = P_w(z)C_w(z)e'_w(k) + P_w(z)d_w(k) \tag{5.2}
\]
where

\[ e'_r(k) = e_r(k) + w\epsilon_r \]
\[ e'_w(k) = e_w(k) + w\epsilon_w \]

and

\[ e_r(k) = r_r(k) - y_r(k) \]  \hspace{1cm} (5.3)
\[ e_w(k) = r_w(k) - y_w(k) \]  \hspace{1cm} (5.4)
\[ \epsilon_r \perp (k) = \beta_r(\beta_re_r(k) - \beta_we_w(k)) \]  \hspace{1cm} (5.5)
\[ \epsilon_w \perp (k) = \beta_w(\beta_we_w(k) - \beta_re_r(k)) \]  \hspace{1cm} (5.6)

\( e_w(k) \) and \( e_r(k) \) are the wafer stage tracking error and the reticle stage tracking error, respectively. The feedback controllers for the wafer and reticle stages, \((C_w(z), C_r(z))\), are designed for basic performance requirement. The plant models and disturbances are denoted by \((P_w(z), P_r(z))\) and \((d_w(k), d_r(k))\). The parameters, \(\beta_r\) and \(\beta_w\), are the coupling gains which convert the individual axes tracking errors into a synchronization error. Note that the parameters are determined by the desired trajectories of the wafer and reticle stages. Figure 5.2, for example, indicates that \(\beta_r = \sin \theta\) and \(\beta_w = \cos \theta\). \(\epsilon_r \perp (k)\) and \(\epsilon_w \perp (k)\) are denoted as the synchronization error mapping onto reticle axis and onto wafer axis, respectively. An additional weighting gain \(w\) is added to determine how large the influence of synchronization error should be in each individual close-loop system.

An improvement over the classic feedback system can be expected by adding the synchronization error into its individual loops. However, the cross-coupled controller equally treats wafer and reticle stage. When the dynamics are significantly different among the two stages which is usually the case, the synchronization performance is limited by the slower axis. Advanced techniques
have approached this problem by weighting the individual axes differently based on their physical nature and/or limitations [25][12][51].

### 5.2.2 Master-Slave Control

Master-slave control is another approach for improving the synchronization performance. As opposed to CCC, it is typically applied to biaxial systems with different dynamics. In photolithography, the reticle stage is lighter and capable of implementing higher bandwidth control than the wafer stage. Thus it makes sense to choose the wafer stage as the master and the reticle stage as the slave. This master stage directly executes track following without consideration of the motion of the slave stage. The actual master stage position is then used to re-formulate the reference trajectory for the slave stage. Figure 5.3 depicts a block diagram of the master-slave control scheme. The reticle stage is commanded to follow a scaled version of the wafer stage trajectory. The block \( S \) in the figure represents the scaling factor. The overall system is described by

\[
\begin{align*}
    y_r(k) &= P_r(z)C_r(z)e_r'(k) + P_r(z)d_r(k) \\
    y_w(k) &= P_w(z)C_w(z)e_w(k) + P_w(z)d_w(k)
\end{align*}
\]

where

\[
\begin{align*}
    e_r'(k) &= Sy_w(k) - y_r(k) \\
    e_w(k) &= r_w(k) - y_w(k)
\end{align*}
\]

Compared to CCC, the master-slave control is relatively simple and more suitable for dynamical systems with different dynamics between the two axes. Recall that the slave stage follows the motion of the master stage and the master stage acts independently. Thus the tracking deviation of the master stage will degrade the tracking performance even more in the slave stage. In addition, when the slave stage encounters a disturbance, the master stage will not be able to know and appropriately address it.
5.2.3 Results

Simulation Results

Simulation studies are conducted on a wafer scanner system consisting of stages, counter-masses, bases, and support structure. A lumped-parameter model is shown in Figure 5.4. In simulation, parameters such as spring and damper coefficients between two stages are chosen based on experimental data. Simulation environment includes repetitive disturbance and non-repetitive disturbances for each stage. The trajectories for the two stages are in opposite directions with the reticle stage moving four times as fast as the wafer stage. Figure 5.5(a) shows the desired position trajectories and Figure 5.5(b) shows the velocity profiles for the wafer stage and the reticle stage. Based on the trajectories, $\beta_r = \sin \theta$, $\beta_w = \cos \theta$ with $\tan^{-1} \theta = \frac{1}{4}$, and $w = 0.5$ are selected for cross-coupled control scheme. $S$ is set to -4 for master-slave control scheme.

First we compare the individual axes tracking performances as well as the synchronization performance when the wafer and the reticle stages have identical dynamic properties. Two nominal
Figure 5.5: Plots of (a) desired trajectories and (b) velocity profiles

plant models are both modeled as a simple mass-damper system as formulated in (3.6). Two systems also have the same PID gains for feedback control ($K_p = 30000$, $K_i = 60000$, and $K_d = 360$).

Figure 5.6 shows the comparison of individual axes tracking errors under three control architectures: individual axes control only, CCC scheme, and master-slave control scheme. In the figure, the blue line represents the axis tracking errors without synchronization control. The green dashed line and the red line are the axis tracking errors with CCC and with master-slave control, respectively. Figure 5.6(a) indicates that the tracking performance of both the wafer stage and the reticle stage is slightly better under CCC. Under the master-slave control, the tracking performance of the master (wafer) stage stays the same as under individual axes control. However, the tracking error of the slave (reticle) stage under master-slave control evidently depends on the motion of the master stage as shown in Figure 5.6(b). Figure 5.7 shows the synchronization performance. The synchronization error in the master-slave control scheme seems to be much reduced during the constant velocity phase as compared to the synchronization error in the CCC scheme. The reason is that the master-slave control can effectively compensate for external disturbance to the master stage and keep the synchronization error between the two stages small. However, as will be shown later by experimental results, the synchronization error during the constant velocity phase is not improved under the master-slave control because of the existence of oscillation which is not modeled in simulation.

Another simulation study is conducted to investigate the properties of synchronization controllers when the two stages have different dynamics. The desired trajectories in this simulation are the same as the previous simulation shown in Figure 5.5(a). The reticle stage is assumed to be lighter and be able to be applied higher bandwidth control ($K_p = 49000$, $K_i = 98000$, and $K_d = 104$). Assume that the reticle stage has the nominal dynamics

$$P_n(s) = \frac{11.79}{1.3s^2 + 1.2s}$$

From Figure 5.8 and 5.9, we observe the trade-off between the tracking performance of individ-
Figure 5.6: Comparison of individual axes tracking errors of (a) wafer stage and (b) reticle stage, when two stages have identical dynamics

Figure 5.7: Comparison of synchronization error when two stages have identical dynamics
Figure 5.8: Comparison of individual axes tracking errors of (a) wafer stage and (b) reticle stage, when two stages have different dynamics

Figure 5.9: Comparison of synchronization error when two stages have different dynamics
ual axes and the synchronization performance. CCC is capable of improving the synchronization performance without compromising the individual axes tracking performance. As already discussed, improvement of synchronization performance is limited by the slower stage. On the other hand, the master-slave control clearly exhibits a better synchronization performance but causes larger tracking errors of the slave stage.

**Experiment Results**

Experiments were conducted to investigate CCC and master-slave control approaches. Trajectory design and parameters settings are basically the same as the simulation settings except $w = 0.15$ is conservatively chosen in the CCC scheme. Figures 5.10 to 5.12 show the comparisons of experiment results. The blue line shows the baseline result when only individual axes control is applied to the wafer and reticle stages. As expected, the tracking performance of the slave (reticle) stage degrades under master-slave control, especially during acceleration and deceleration transients. However, master-slave control approach ended up with larger synchronization error since our wafer and reticle stages have the same dynamical properties and neither of the two stages is appropriate as a slave stage. The reason for increased synchronization error can be explained by considering that two stages have similar resonant frequencies. Oscillations in the master stage inevitably are fed into the slave stage and consequently are amplified. On the other hand, the CCC apart from the individual axes control utilized the axis tracking error to construct an additional control effort for synchronization purposes. Under CCC scheme, the improvement of tracking error for each axis is relatively minor when the weighting parameter, $w$, is less than one. As shown in Figure 5.12, CCC with $w = 0.15$ achieved a smaller synchronization error in the transient phase (constant velocity phase starts from time index 626) and improved only a little the synchronization error afterward. The larger the weighting parameter in CCC scheme, the more aggressive the feedback controller, the better the performance with respect to both individual axes tracking errors and synchronization error. Figure 5.13 compares the synchronization error under CCC with different weighting values. Note that synchronization performance is decent, but is obtained at the price of high control chattering when $w$ is set to 0.5. That is, larger weighting parameters help improve tracking errors as well as synchronization error but at the same time adversely affect the stability of the overall system.

### 5.3 ILC Design for Synchronization

Successful scanning requires extremely precise synchronization between the wafer and reticle stages during exposure. In the preceding sections, synchronization control methods were developed in the feedback framework resulting in limited performance improvement. Using the CCC and master-slave methods, enhanced performance for the tracking error of individual axes may need to be scarified if we place more emphasis on the synchronization error. In this Chapter, a synchronization ILC algorithm is proposed to improve synchronization performance without com-
Figure 5.10: Experiment results for the comparison of individual axes tracking errors of (a) wafer stage and (b) reticle stage

Figure 5.11: Experiment results for the comparison of synchronization error
Figure 5.12: Zoomed-in detail of the synchronization error plot

Figure 5.13: Synchronization error under CCC with different weighting values
promising the synchronization error and the individual axes tracking errors. The effectiveness of
the synchronization ILC scheme is verified by experiment.

5.3.1 The Synchronization ILC Scheme

In this section, a synchronization ILC scheme is developed for the wafer and reticle stages. A
straightforward method of implementing ILC to the wafer scanners is to have an individual ILC
loop for each stage. ILC efforts to the wafer stage and the reticle stage are independently updated
and can be expressed as

\[
\begin{align*}
    u_{w,j+1}^{\text{ILC}}(k) &= Q_w(z)[u_{w,j}^{\text{ILC}}(k) + L_w(z)e_{w,j}(k)] \\
    u_{r,j+1}^{\text{ILC}}(k) &= Q_r(z)[u_{r,j}^{\text{ILC}}(k) + L_r(z)e_{r,j}(k)]
\end{align*}
\] (5.10)

where \( Q_w(z) \) and \( Q_r(z) \) are low-pass filters \( L_w(z) \) and \( L_r(z) \) are the learning functions for the wafer
stage and the reticle stage, respectively. Those filters and learning functions are designed based on
robustness and performance as we have discussed in the previous chapters. For brevity, the super-
script in \( u_{w,j+1}(k) \) and \( u_{r,j+1}(k) \) will be dropped when no confusion arises. Applying individual
ILC loops, the increase of tracking accuracy for individual stages can be achieved resulting in bet-
ter synchronization accuracy. To further improve the synchronization performance, incorporation
of the synchronization error into the learning update law is essential. Borrowing from the concept
of cross-coupled control, the synchronization ILC updating law can be determined by combining
the the synchronization error with individual axes tracking errors.

\[
\begin{align*}
    w_{j+1}(k) &= Q_w(z)[u_{w,j}(k) + L_w(z)(\alpha e_{w,j}(k) + (1 - \alpha) e_{w,j}^\perp)] \\
    r_{j+1}(k) &= Q_r(z)[u_{r,j}(k) + L_r(z)(\alpha e_{r,j}(k) + (1 - \alpha) e_{r,j}^\perp)]
\end{align*}
\] (5.11)

\( e_{w,j}^\perp \) and \( e_{r,j}^\perp \), as defined in (5.6) and (5.5), represent the synchronization errors in the \( j^{th} \) iteration
mapped onto the wafer stage and the reticle stage, respectively. \( \alpha \) is a scalar between from 0 and 1
for determining the emphasis on the synchronization error. For example, only the tracking errors
are present in the ILC algorithms when \( \alpha = 1 \). In other words, the synchronization ILC algorithm
becomes exactly the same as individual ILC. On the contrary, only synchronization errors are
present in the ILC algorithms when \( \alpha = 0 \). By selecting \( 0 < \alpha < 1 \), we substitute a virtual target
position for the original desired position. Then the error utilized to update the ILC algorithm
includes information involving tracking error and synchronization error. As illustrated in Figure
5.14, the virtual target position is represented as a pink dot and the updated error is indicated by
the vector between the actual position and the virtual target position.

Using the properties of the coupling gains \( (\beta_w^2 + \beta_r^2 = 1) \), the learning update law can be rewrit-
ten in a more compact form. To study the stability properties of the synchronization ILC scheme,
we set \( Q_w(z) \) and \( Q_r(z) \) equal to one and present the learning update law in the z-transform.

\[
\begin{align*}
    U_{w,j+1}(z) = U_{w,j}(z) + L_w(z)(\beta_w^2 + \alpha \beta_r^2)E_{w,j}(z) - (1 - \alpha) \beta_w \beta_r E_{r,j}(z) \\
    U_{r,j+1}(z) = U_{r,j}(z) + L_r(z)(\beta_r^2 + \alpha \beta_w^2)E_{r,j}(z) - (1 - \alpha) \beta_w \beta_r E_{w,j}(z)
\end{align*}
\] (5.14)
Then the error evolution equations can be expressed as

\[
E_{w,j+1}(z) = [1 - (\beta_w^2 + \alpha \beta_r^2)T_{w,u}(z)L_w(z)]E_{w,j}(z) - (1 - \alpha)\beta_w \beta_r T_{w,u}(z)L_w(z)E_{r,j}(z) \tag{5.16}
\]

\[
E_{r,j+1}(z) = [1 - (\beta_r^2 + \alpha \beta_w^2)T_{r,u}(z)L_r(z)]E_{r,j}(z) - (1 - \alpha)\beta_w \beta_r T_{r,u}(z)L_r(z)E_{w,j}(z) \tag{5.17}
\]

where \(T_{w,u}(z)\) and \(T_{r,u}(z)\) are the transfer functions from the ILC injection point to the system output respectively for the wafer and the reticle stages. In matrix form, we can write the above equations as

\[
\begin{bmatrix}
E_{w,j+1}(z) \\
E_{r,j+1}(z)
\end{bmatrix}
= \begin{bmatrix}
M_{11}(z) & M_{12}(z) \\
M_{21}(z) & M_{22}(z)
\end{bmatrix}
\begin{bmatrix}
E_{w,j}(z) \\
E_{r,j}(z)
\end{bmatrix}
\]

where the four elements of \(M(z)\) are given by

\[
M_{11}(z) = 1 - (\beta_w^2 + \alpha \beta_r^2)T_{w,u}(z)L_w(z) \\
M_{12}(z) = -(1 - \alpha)\beta_w \beta_r T_{w,u}(z)L_w(z) \\
M_{21}(z) = 1 - (\beta_r^2 + \alpha \beta_w^2)T_{r,u}(z)L_r(z) \\
M_{22}(z) = -(1 - \alpha)\beta_w \beta_r T_{r,u}(z)L_r(z)
\]

The synchronization ILC scheme is stable and the monotonic convergence of the error is guaranteed if

\[
\rho(M(z)) \leq \bar{\sigma}(M(z)) < 1 \tag{5.18}
\]

To simplify calculation, the above condition for monotonic convergence can be rewritten as

\[
\bar{\sigma}(M(z)) \leq \bar{\sigma}(M_a(z)) + \bar{\sigma}(M_b(z)) < 1 \tag{5.19}
\]

where \(\bar{\sigma}(M_a(z))\) and \(\bar{\sigma}(M_b(z))\) are respectively

\[
\bar{\sigma}(M_a(z)) = \bar{\sigma}\left(\begin{bmatrix}
M_{11}(z) & 0 \\
0 & M_{22}(z)
\end{bmatrix}\right) \tag{5.20}
\]
\[
\bar{\sigma}(M_b(z)) = \bar{\sigma}\left(\begin{bmatrix} 0(z) & M_{12}(z) \\ M_{21}(z) & 0 \end{bmatrix}\right)
\] (5.21)

Using the norm properties,
\[
\bar{\sigma}\left(\begin{bmatrix} 0(z) & M_{12}(z) \\ M_{21}(z) & 0 \end{bmatrix}\right) = \max\{\bar{\sigma}(M_{12}(z)), \bar{\sigma}(M_{21}(z))\}
\] (5.22)

Then the condition for monotonic convergence proposed in [7] can be deduced to
\[
\bar{\sigma}(M(z)) \leq \max\{\bar{\sigma}(M_{11}(z)), \bar{\sigma}(M_{22}(z))\} + \max\{\bar{\sigma}(M_{12}(z)), \bar{\sigma}(M_{21}(z))\} < 1
\] (5.23)

The learning functions are designed based on this condition.

The remaining question is how to determine the value of \(\alpha\). If \(\alpha\) is not selected properly, the algorithm may exhibit either poor tracking performance or disappointing synchronization performance. One possible solution is to select \(\alpha\) by the iteration-varying method. This iteration varying method compares the individual axes tracking performance to the synchronization performance for each iteration.

\[
\alpha_{j+1} = \frac{e_j^T e_j}{e_j^T e_j + e_j^T e_j}
\] (5.24)

Clearly, the iteration-varying \(\alpha\) suggests that the algorithm emphasizes the individual axes tracking errors whenever \(\|e_j\|^2 > \|e_j\|^2\) and emphasizes on the synchronization error otherwise.

One may notice that the iteration-varying \(\alpha\) is obtained by evaluating performance over the entire time indexes. As long as the overall tracking error is more evident than the overall synchronization error, the algorithm concerns more about the tracking error even if the synchronization error is more significant at certain time indexes. To be capable of handling this situation, we make \(\alpha\) time-varying by slightly modifying (5.24).

\[
\alpha_{j+1}(k) = \frac{e_j^T(k) e_j(k)}{e_j^T(k) e_j(k) + e_j^T(k) e_j(k)}
\] (5.25)

The time-varying strategy ensures that the synchronization ILC algorithm properly balances the tracking performance and the synchronization performance at any time index.

### 5.3.2 Experiment Results

The synchronization ILC scheme presented in the previous section is illustrated using the experimental wafer and reticle stages in the Mechanical System Control Laboratory described in Chapter 1. Several experiments were conducted to investigate different synchronization approaches and demonstrate the effectiveness of the proposed method. Figure 5.15-5.17 show performance comparison when \(\alpha\) is 0, 0.5, and 1. Without compensation of the synchronization errors (\(\alpha = 1\)), both the wafer and reticle stages can achieve promising tracking performance, but the synchronization
Figure 5.15: Performance comparison with different $\alpha$ values. Experiment results of the wafer stage for (a) two norm of tracking error over iterations, and (b) trajectory tracking error in the 9th iteration.

Figure 5.16: Performance comparison with different $\alpha$ values. Experiment results of the reticle stage for (a) two norm of tracking error over iterations, and (b) trajectory tracking error in the 9th iteration.
Figure 5.17: Performance comparison with different $\alpha$ values. Experiment results for (a) two norm of synchronization error over iterations, and (b) synchronization error in the $9^{th}$ iteration.

Figure 5.18: Performance comparison with iteration-varying and time-varying $\alpha$. Experiment results for (a) two norm of synchronization error over iterations, and (b) synchronization error in the $9^{th}$ iteration.
error is evident. During the constant velocity phase, the peak to peak synchronization error is roughly 160nm when $\alpha = 1$. The peak to peak synchronization error can be reduced to 40nm if $\alpha = 0$ is selected, but the tracking errors in the two stages are unacceptable. With $\alpha = 0.5$, the peak to peak synchronization error is around 70nm, but the tracking errors in the two stages are not comparable to the tracking errors when $\alpha = 1$. This leads to the conclusion that selecting a fixed $\alpha$ has a limitation in dealing with the trade-off between tracking performance and synchronization performance.

It is expected that an adjustable $\alpha$ is more capable of handling the trade-off. Figure 5.18 indicates that the iteration-varying method proposed in (5.24) and the time-varying method proposed in (5.25) can achieve comparable synchronization performance compared to the case of a fixed $\alpha$ of 0. Meanwhile, the two methods maintain the tracking performance of the two stages compared to the case of a fixed $\alpha$ of 1 as shown in Figure 5.19. The advantage of the time-varying method over iteration-varying method can be seen in Figure 5.18 (b). The peak synchronization error in acceleration phase as marked in the figure is around 90nm for the iteration-varying method. The peak synchronization error is reduced to 35nm for the time-varying method. The reason for the peak synchronization error reduction is that the time-varying method allows $\alpha$ to be flexibly adjusted in each time index. This makes the time-varying method capable of balancing the tracking performance and synchronization performance at each time index.
5.4 Chapter Summary

In this chapter, two common control approaches for synchronization are presented: the master-slave control and cross-coupling control. In the master-slave control scheme, the wafer stage serves as the master and the reticle stage as the slave. This is an intuitive solution since the reticle stage has a smaller structure which means a wider control bandwidth. The wafer stage position is used to formulate the reference trajectory for the reticle stage. In addition, the disturbance entering into the slave system is suppressed only by the disturbance control capability of the slave control system because there is no synchronization correction loop which passes a signal from the slave control system to the master control system. A cross-coupled control method is another way to improve the synchronization performance by coupling the individual axes errors and applying a controller to the combined signal. The control input is calculated by taking into consideration of the mutual influences among axes, which makes the control input on a single axis dependent on the tracking performance of the other axes. The cross-coupled method, however, takes no advantage of the physical nature of the axis which has a wider control bandwidth. The performance of the two approaches and their limitations were investigated and compared by simulation and experiment. The results show that the synchronization improvement is limited because the two approaches are developed in the feedback control framework.

To achieve a better synchronization performance, we proposed a synchronization ILC design methodology that synchronizes the two stages while ensuring that the tracking error of each individual loop is also minimized. In the synchronization ILC method, a variable $\alpha$ has impact on balancing the tracking performance and the synchronization performance. From the experiment results, we concluded that $\alpha$ needs to be adjustable so as to fulfill the tracking performance requirement as well as the synchronization performance requirement.

Strategies for the time-varying $\alpha$ and the iteration-varying $\alpha$ in the synchronization ILC scheme were developed. Experimental results showed that both methods yield outstanding performance improvement in terms of synchronization accuracy without compromising individual axes tracking performance. Experiment results also showed that the time-varying $\alpha$ is more capable of reducing peak errors by comparing the tracking and synchronizing errors in every time index.

It should be noted that introducing DOB into each axis control-loop developed in chapter 3 can further improve axis tracking performance. This would also result in better synchronization performance. Since we are more interested in improving synchronization performance by ILC strategy, adding additional DOB was not considered. As for optimal feedback-feedforward control developed in Chapter 4, deriving the overall optimal control requires accurate MIMO system models and increases complexity for implementation.
Chapter 6

Conclusions

This chapter summarizes the conclusions of this thesis and provides some thought on the future direction of this research.

6.1 Summary and Conclusions

The objective of this thesis has been to explore the use of advanced control methodologies to significantly enhance the throughput in modern photolithographic systems.

Iterative Learning Control (ILC) has been applied in wafer stage positioning systems for improving tracking error and rejecting repeatable disturbances. The presence of non-repetitive disturbances limits the performance of ILC. To address this problem, we first studied ILC from the perspective of enhancing its performance in the presence of non-repetitive disturbances. Instead of preventing non-repetitive disturbances from entering into the iterative learning algorithm, we proposed to directly attenuate non-repetitive disturbances in the time domain by the use of a disturbance observer (DOB). The performance improvement of incorporating DOB into the ILC scheme was demonstrated by both simulations and experiments. With less subject to disturbances, we were able to separate control efforts to disturbance compensation from trajectory following. The learned ILC effort then was parameterized as a linear function of the effective mass and the damping coefficient to obtain better approximation of the plant model. Based on this parametrization, the plant parameters can be adjusted for system variations such as a change in the payload. An initial selection of control effort in ILC then can be pre-calculated for achieving a faster convergence rate and lower tracking error. The proposed method of pre-determining the initial ILC effort allowed the learning data from a single motion to be generalized to a class of motion profiles. Moreover, the updated plant parameters are utilized to re-design the DOB for achieving better disturbance rejection.

We learned that DOB is an effective way to directly reject disturbances so that they do not enter into the learning process. We also found that a well-designed feedforward control effort in addition to ILC effort improves the convergence rate as well as the performance of the converged
system. This inspired us to develop an optimal feedback-feedforward control procedure which is integrated with the ILC scheme. The procedure includes the selection of the learning gain and cutoff frequency of the Q-filter for P-type ILC in combination with the optimization of the controller parameters. The procedure improves the overall system performance by synergizing the roles of both feedback and ILC controllers.

Finally, the synchronization issue of the wafer and reticle stages was discussed. In order to reduce the alignment errors, we developed a synchronized ILC algorithm by borrowing ideas of cross-coupling control. In particular, a time-varying $\alpha$ was proposed to adjust the weighting on the synchronization error versus tracking error. The adjustable weighting variable makes it possible to balance the trade-off between individual tracking performance and synchronization performance. Analytical conditions for stability and error convergence have also been derived. The effectiveness of the proposed synchronized ILC algorithm was validated through experimental implementation.

### 6.2 Further Research

PID is widely adopted in industry because it is simple and a precise plant model is rarely available for advanced model based controller design. Analogously, P-type ILC is also simple and little prior knowledge from the plant model is needed. For practical applications, P-type ILC still is most robust and easiest to implement among other sophisticated ILC algorithms. This thesis mainly focused on the P-type ILC integrated with different advanced control approaches such as DOB and model predictive control. It is also possible to design more complicated Q-filters as well as learning functions to increase convergence rate with lower convergence error. Since learning control algorithms are iterative schemes, any accumulation of undesired signal can lead to divergence. There are still numerous open problems left to researchers for theoretical analysis and practical implementation.

The stability analysis presented in this thesis was developed in the frequency domain representation for linear time invariant systems. In the conversion to the frequency representation, it is assumed that the time horizon is infinite. In actual applications, this is not the case. Instead, the tracking performance is considered over a finite time horizon. When using the frequency domain representation to analyze ILC systems, it has to be taken into account that the model is an approximation. The time varying systems impose the further challenges for stability and performance analysis.

In this thesis DOB has been integrated with ILC to deal with non-repetitive disturbances. It should be noted that DOB and ILC worked individually when the model uncertainty or time delay of a system is small. In the case of significant uncertainties in plant parameters, the DOB can be updated based on the data from the previous learning process as proposed in Chapter 3. DOB, however, has difficulty in handling systems with a large time-delay. In this case, the proposed control scheme may have limitation on improving performance. A possible way to deal with large time-delays is to incorporate the errors at future steps in the previous iteration cycle. This modification, however, will not work if the disturbance is totally non-repetitive.
This thesis also pointed out that adding a pre-design optimal feedforward control in addition to ILC is attractive for further performance improvement. Inspired by this work, a possible future work is to derive an optimal learning function by combining the features of ILC and preview control since the two share similar perspective about taking future information to improve the performance and to reduce the transient response.

For synchronization, a time-varying method was presented to adaptively weigh on synchronization error so to ensure achieving minimal synchronization error without compromising individual tracking performance. Instead of time-varying gains, an optimal choice of the weighting gain based on some cost functions is another possible direction for future research. The proposed synchronization ILC scheme was designed for the stages with similar dynamical properties. Synchronization ILC for the stages with dissimilar dynamical properties is also a potential direction to work on.
Bibliography


