Development of Intelligent Learning Motion Control Systems

ZHAO SHAO

NATIONAL UNIVERSITY OF SINGAPORE
2005

Development of Intelligent Learning Motion Control Systems

ZHAO SHAO
(M.Eng., B.Eng., Xi'an Jiaotong Univ.)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2005

Acknowledgments

I would like to express my sincerest appreciation to all who had helped me during my study in National University of Singapore. First of all, I would like to thank my supervisor Associate Professor Tan Kok Kiong for his helpful discussions, support and encouragement. His vision and passion for research influenced my attitude for research work and spurred my creativity. I also want to thank Associate Professor Xu Jian-Xin, Professor Lee Tong Heng, Dr. Huang Sunan, Mr. Andi Sudjana Putra and Mr. Chua Kok Yong for their collaboration in the research works.

I would like to give my gratitude to all my friends in Mechatronics and Automation Lab. I would especially like to thank Dr. Tang Kok Zuea, Ms. Raihana Ferdous, Mr. Tan Chee Siong, Mr. Goh Han Leong, and Mr. Teo Chek Sing for their inspiring discussions and advice.

Finally, I would like to thank my family for their endless love and support. Specially, I would like to express my deep gratitude to my husband Zheng Jie for his understanding and support.

Contents

List of Figures

List of Tables

List of Abbreviations

DC	*Direct Current*
DSP	*Digital Signal Processing*
et al.	*et alii*
etc.	*et cetera*
FFT	*Fast Fourier Transform*
LPM	*Linear − Piezoelectric Motor*
ILC	*Iterative Learning Control*
I/O	*Input/Output*
LS	*Least Square*
MEMS	*Micro − Electro − Mechanical Systems*
PMLM	*Permanent Magnet Linear Motor*
RAM	*Reference Adjustment Mechanism*
RBF	*Radial Basis Function*
RLS	*Recursive Least Square*
RMS	*Root − Mean − Square*

Summary

Modern mechanical systems such as machine tools, microelectronics manufacturing equipments, mechanical manipulators and automatic inspection machines need precision motion control to achieve good positioning/tracking performance at high speed and high accuracy. This results in increasing demands on higher productivity and product quality in the manufacturing industries. Thus, the requirements on motion control systems become more and more stringent. But conventional control techniques can no longer satisfy the increasingly stringent performance requirements of motion control systems. Recently, intelligent learning control emerges as an effective way to meet the stringent positioning requirements. In this thesis, intelligent learning control algorithms are developed to achieve better positioning/tracking performance in motion control systems.

In this thesis, linear motors as the mechanical servo systems are mainly studied. Linear motors are widely used for applications requiring linear motion at high speed and high accuracy. The most attractive features of linear motors for precision motion control include the high force density achievable, low thermal loss, simple mechanical structure, high dynamic performance and improved reliability. However, the achievable performance of linear motors is unavoidably limited by presence of the nonlinear effects and uncertainties present. The predominant nonlinear effects underlying a linear motor system are the frictional force and force ripples. In some parts of the thesis, the intelligent learning control schemes are proposed to compensate the friction and force ripples. Besides the compensation of the nonlinear effects in linear motors using intelligent control algorithms, this thesis proposes some new ideas that aim at solving the problems faced in the field of the precision motion control. It includes the developments of the Iterative Learning Control (ILC) for time-delay systems and predictive Iterative Learning Control (ILC) for time-varying, linear and repetitive systems.

Firstly, an adaptive control algorithm is presented to suppress the force ripples in Permanent Magnet Linear Motors (PMLMs). The model of force ripples is derived. The idea is to use the Recursive Least Square (RLS) method to model and then reduce the force ripples. Thus, linear regression form of the PMLM model is required. It means that the frequencies of the force ripples should be determined before the implementation of the adaptive control scheme. The displacement periodicity of the force ripple is obtained by using a Fast Fourier Transform (FFT) analysis. Based on the full model, the control algorithm can be commissioned which consists of a PID feedback control component, an adaptive feedforward component for compensation of the force ripple and another adaptive feedforward component based on the inverse dominant linear model.

Then, an Iterative Learning Control (ILC) scheme, a model-free approach, is proposed to compensate the friction and force ripples in the linear motors to achieve good tracking performance for high precision and repetitive motion control applications. It consists of a self-tuning Radial Basis Function (RBF) network and an Iterative Learning Control (ILC) component. The RBF network is applied to model the tracking error over a cycle. The ILC scheme is used to adjust the reference signal repetitively. The ILC component further enhances the tracking performance, particularly over the section of the trajectory where the RBF network is less adequate in its modeling function.

An online automatic tuning method of PID controller based on an Iterative Learning Control (ILC) approach is presented in this thesis. The basic idea is to use ILC to obtain a satisfactory performance for the system to track a periodic reference sequence. A modified ILC scheme iteratively changes the control signal by adjusting the reference signal only. Once the satisfactory performance is achieved, the PID controller is then tuned by fitting the controller to yield a close input and output characteristics of the ILC component.

Next, a new form of repetitive learning control (ILC) approach is proposed which is applied to time-delay systems for the first time. In the thesis, a necessary and sufficient convergence condition is derived for the new proposed repetitive control. Additionally, a robust convergence analysis for the repetitive control under the existence of a time-delay mismatch, initialization errors, disturbances and measurement noise is provided to show the robustness of the new proposed approach.

Finally, a predictive Iterative Learning Control (ILC) algorithm is developed for time-varying, linear and repetitive systems. An error model is introduced, which represents the transition of the tracking error between two successive trials. Based on this model, a predictive iterative learning algorithm is derived, which is only based on the trial number. In the thesis, a rigorous convergence analysis is provided. In addition, the robustness of the algorithm against modeling errors, initial errors, as well as the presence of disturbances are discussed.

Extensive simulation and experimental results are furnished to illustrate the effectiveness of the proposed learning approaches.

Chapter 1

Introduction

Although conventional control had a long history in theory and practice, it has encountered many difficulties in its applications to modern motion control systems. Modern mechanical systems are often required to yield high productivity and quality at high speed and high accuracy. Such an increasingly tight control performance requirements pose a great challenge for researchers and engineers to seek novel algorithms beyond the conventional control theory. Recently, intelligent controls become effective ways to overcome the difficulties. In this thesis, the intelligent learning control approaches are investigated for the precision motion control systems.

1.1 Precision Motion Control

Precision Engineering is the multidisciplinary study and practice of design for precision, metrology, and precision manufacturing. Precision engineering is defined in [1] as the 'set of systematized knowledge and principles for realizing high-precision machinery'. Precision engineering can be generally defined at the micrometre scale which means the accuracy of 1 micron at manufacturing. Currently, many researchers and engineers aim at creating higher precision machines and manufacturings.

Nowadays, manufacturing industries are confronted with increasing demands of higher quality and higher productivity. These demands can be achieved with high speed, highly accurate motion and positioning. Performance of motion depends on electrical and mechanical components, which are used in assembling of drives, as well as the motion controller. Precision motion is an indispensable part of manufacturing, for example, read/write head motion in disk drives, motion of chip placement actuators in surface mount machines, laser drill motion in electronic packaging, scanning motion in confocal microscope, etc. Precision motion is also critical for micro-assembly and Micro-Electro-Mechanical-Systems (MEMS) actuation in applications to RF, micro-optic and micro-fluidic devices. With the continuing demand on high performance and low cost, the requirement on the precision motion control is ever more stringent.

Although a great deal of effort has been devoted to the field of precision motion control, some issues encountered in precision motion control attracted the researchers to explore in this field. One such concern is the control of linear motion. In the real world, many mechanical systems, such as machine tools, semiconductor manufacturing equipment and automatic inspection machines, require linear motions. One common way to realize the linear motion by rotary motors is to use gears, lead screw and other transmission mechanisms to convert rotary motion into linear motion. These mechanisms may influence the speed, accuracy and dynamic response. Also it may introduce the effects of contact-types of nonlinearities and disturbances such as backlash and frictional forces.

The linear motors, as a direct drive, can be used to eliminate the gears and other mechanisms, with accompanying of quietness and reliability. This can significantly reduce the effects of contact-type nonlinearities and some disturbances such as backlash and frictional forces and increase the reliability of the system. In recent years, linear motor has received increased attention for use in applications requiring linear motion at high speed and high accuracy.

In this thesis, two specific types of linear motors are investigated: Permanent Magnet Linear Motors (PMLMs) and Linear-Piezoelectric Motors (LPMs).

1.1.1 Permanent Magnet Linear Motor

Compared to the traditional rotary machines, the main benefits of a PMLM include the high force density achievable, low thermal losses and most importantly, the high precision and accuracy associated with the simplicity in mechanical structure. The PMLM shows superior performance over many conventional rotary motors. However, the nonlinear effects associated with the PMLM are inevitably arising. The more predominant nonlinear effects underlying the PMLM are friction which is inevitably present as long as there is relative motion between two bodies in contact, and force ripples, arising from the magnetic structure of the PMLM and other physical imperfections. The two primary components of the force ripple are the cogging (or detent) force and the reluctance force [2]. The cogging force arises due to the interaction of the permanent magnets in the stator with the iron cores of the translator. This force exits even in the absence of any winding current and it exhibits a periodic relationship with respect to the position of the translator relative to the magnets. The reluctance force is due to the variation of the self-inductance of the winding with respect to the relative position between the translator and the magnets. The reluctance force also has the periodic relationship with the translator-magnet position.

Friction and force ripples pose several difficulties to motion systems. Stiction, for example, induces stick-slip motion. Limit cycle oscillations can also occur due to discontinuous nature of the frictional force with respect to velocity. Force ripples produce "bumps" along the direction of motion, which may cause difficulties in achieving smooth and yet high speed motion with linear control alone. Owing to the typical precision requirements associated with the use of PMLMs, it is thus an important and challenging task to effectively deal with these nonlinear effects.

1.1.2 Linear-Piezoelectric Motors

A piezoelectric motor is a type of actuator that uses mechanical vibrations in the ultrasonic range in a stator structure. Piezoelectric actuators are innovative manipulators which have shown a high potential in applications requiring manipulation within the submicrometer or even nanometer range. There are two main classes of linear-piezoelectric motors (LPMs), classified according to the structures and driving principles. The first class works on a direct-drive principle. Deformations of a piezoelectric element are directly used to drive the load for precision positioning [3][4][5]. This type motor has the superior performance with high resolution and nanometer grade positioning precision, short stroke and a high bandwidth. The second class of LPM is based on the indirect-drive principle. The ultrasonic motor is a kind of this class. In a piezoelectric linear ultrasonic motor, high frequency oscillation are generated by using the piezoelectric effect, and the rotor is driven by the frictional forces generated at the interface between the stator and the rotor. The main characteristics of this ultrasonic motors are: high resolution, wide dynamic range of velocity, hold stability at power off and a small compact structure. In this thesis, this type of indirect-drive LPM is the platform for experiments.

For this type of LPM, friction has been identified as the main problem to be addressed [6]. The highly nonlinear features of friction associated with the servomechanisms pose the challenges for the researchers and engineers in the control areas. The friction needs to be compensated in order to improve the transient performance and to reduce steady-state tracking error.

The nonlinear effects present in the liner motors can be minimized or eliminated either through proper mechanical design or via the control algorithms. But the mechanical design often increase the complexity of the motor structure and the production cost. Therefore more attention focuses on developing the control algorithms for the high precision applications.

1.2 Intelligent Learning Control

Intelligent control is a highly multi-displinary technology where controllers are designed that attempt to model the behaviors of human being. These behaviors include adaptation, learning and making decision. Nowadays, the area of intelligent control tends to include everything that is not covered in conventional control.

The automatic control has been used more than 2000 years since the Romans invented a water-level control device [7]. The notable control invention was the steam engine governor in 18th century. In the early 1920s, the development of control theory began and the feedback controllers were widespreadly adopted in the applications. After that, the Second World War brought tremendous impetus for the advancement of control. From 1960s to 1980s, the developments of the modern control theory and real-time digital computers had a significant impact on the control applications. The application of more powerful computers played a key role in the implementation of more sophisticated control strategies. With the demand for enhanced performance of the highly complex systems, the linear control theory cannot address this demand solely. Intelligent control has arisen as a collection of various control methodologies that have addressed to meet this trend.

An important attribute or dimension of an intelligent control is learning. Learning means that the controller has the ability to improve its future performance based on past experience. In solving some control design problems, the available a priori model information is so limited that it is difficult to design a control system that meets the desired performance specifications. Intelligent learning control provides the solution for this problem with flexibility. With the intelligent learning control, the control system can be designed to on-line adjust itself automatically to suppress the uncertainty and thus to enhance performance. Therefore, the intelligent learning control approaches are developed for precision motion control systems in this thesis. Iterative Learning Control (ILC) is mainly studied with respect to the different problems faced in the precision motion control. Additionally, the adaptive control and Radial Basis Function (RBF) network are also involved as the intelligent learning control approaches in this thesis.

Among the existing intelligent control approaches, the Iterative Learning Control (ILC) has become popular approach, especially when dealing with repetitive tracking control or periodic disturbance rejection problems. The concept of Iterative Learning Control (ILC) began to flourish in 1984 by Arimoto et al. [8]. It is a technique for improving the performance of systems or processes that operate repetitively over a fixed time interval. The monograph by Moore [9] contains more details on the background of the learning algorithm. A recent book [10] surveys the development of this research area from inception till 1998. Nowadays, ILC has attracted some interest in control theory and applications. It has been widely applied to mechanical systems such as robotics, electrical systems such as servo motors, chemical systems such as batch reactors, as well as aerodynamic systems, etc.

The goal of iterative learning control is to improve the tracking performance of a repetitive operation where the system is designed to return to the same initial condition before beginning the next repetition. The concern in ILC is to find an appropriate control input that forces the system output to follow the desired trajectory. The desired trajectory and the trial length are defined for a fixed time interval. In contrast, when system operates to track a periodic signal continuously in time, Repetitive Control

(RC), as one emerging area in ILC research, should be considered. Repetitive control is concerned with canceling an unknown periodic disturbance or tracking an unknown periodic reference signal [11]. Unlike ILC, in repetitive control system the terminal state of previous trial is automatically the initial state of current trial. The early works of repetitive control can be found in [12] and [13]. The summary of repetitive control works can be referred in [14] [15]. There are differences between ILC and repetitive control. However, they are not really different. In fact, there is a bridge between the ILC and repetitive control. The repetitive control can be interpreted as "no-reset" ILC in [11] [16] [17] and [18]. That means that the structure is same as in the ILC but the system is not reset at the beginning of each iteration. Additionally, ILC and RC are bridged with the ideas in Longman's works [19] [20]. In this thesis, the ILC and repetitive control schemes are investigated to enable enhanced performance in motion control systems used in the manufacturing industries.

In the control of linear motion via linear motors, the conventional Proportional-Integral-Derivative (PID) control usually does not suffice in the high precision application domain. It is an interesting and challenging problem to compensate the friction and force ripples adequately. In the literature, a large number of methods has been proposed. For friction reduction, model-based approaches are usually used. In [6], models of varying complexity have been used to approximate the dynamics of friction. In [21], a robust adaptive schemes were developed for friction compensation. In [22], an evolutionary programming approach has been proposed to deal with the same problem. This method can identify the friction by formulating the identification task as an optimization problem. With regard to force ripple suppression, in the early years, it was achieved in the system design phase through good hardware design. The force ripple may be minimized by skewing the magnets [23] or optimizing the disposition and width of the magnets [24] [25]. However, these techniques often increase the complexity of motor structure and the production cost. Recently, the development of the advanced control can compensate the undesirable nonlinear effects by the additional control effort. Some researchers [26] develop the force ripple model and identify the force ripples with a force sensor and a frictionless air bearing support of the motor carriage. In [27], a method of force ripple identification was done in a closed position control loop by measurement of the control signal for movements with respect to the different load forces without the additional force sensors. In [28], a neural-network based feedforward assisted PID controller was proposed. [29] presented a H_∞ optimal feedback control scheme to provide a high dynamic stiffness to external disturbance. The authors in [30] [31] [32] proposed adaptive algorithms for the rejection of sinusoidal disturbances of unknown frequency. In this thesis, it is an objective to compensate the nonlinear effect caused by the friction and force ripples with intelligent approaches.

Moreover, one prominent challenge faced for industrial systems is the time delay, a common characteristic of many industrial systems. It is an applied problem. Delay systems can be classified as function differential equations which are infinite dimensional and include information on the past history. Compared to the systems without time delays, the difficulty of a control system design for time delay systems increases with the value of time delay. It is because there exists time delay term in the characteristic equation of the system. Survey papers provided the overview of the study of the time-delay systems, such as [33] [34] [35] [36]. The book by Gu [37] investigated the stability of linear time-delay systems in detail. In [38], the stability of a linear system with a point-wise, time-varying delay was explored. Besides the stability analysis for linear time-delay systems, many techniques were developed for the nonlinear systems with time delays. In [39], stabilizing controller was designed for a class of nonlinear time-delay systems, based on the Lyapunov-Krasovskii functionals. In [40], robust adaptive control was proposed for a class of parametric-strict-feedback nonlinear systems with unknown time delays. In the manufacturing industries, many tasks, such as batch job of certain chemical processes, spray painting, and arc-welding, are repetitive and require a controller that can track a given desired trajectory. For this issue in the motion control, considerable advanced control algorithms are proposed to compensate the time delay. In [41] and [42], the author investigated the time delay effects in Iterative Learning Control schemes for state-delay systems. In these papers, the focus was the development of ILC for state-delay systems. In [43] and [44], robust ILC design with the Smith Predictor controller was proposed. In this thesis, repetitive learning control is extended to control the systems with the input time delay.

Finally, another challenge confronted in the ILC is how to achieve a rapid and guaranteed reduction in the learning error. The normal ILC scheme, as a model-less approach, cannot guarantee the fast reduction in the error. Additionally, the normal ILC possesses limitations in terms of achievable performance and tuning guidelines, especially in multivariable control problems, in [45]. To overcome these limitations, model-based ILC approaches have been proposed. In [46], the authors introduce some of the ILC work in the Sheffield group, especially in the area of optimal ILC. In [47], parameter optimization through a quadratic performance index was proposed as a method to establish a new iterative learning control law. In [48], the possibility of applying norm-optimal ILC to non-linear plant models was investigated. In [49], a learning control scheme was proposed to find a finite-time optimal control history that minimizes a quadratic cost. In [50], an ILC algorithm was developed based on an optimization principle. In the thesis, predictive ILC algorithm is developed. As a model-based approach, can estimate the future signals through prediction and achieve better performance. Unlike the above mentioned methods, in the proposed method, more than one cycle signals in terms of the trial number are involved in the quadratic performance index, based on the derived prediction model.

1.3 Contributions

This thesis aims at developing the intelligent learning control approaches for the motion control systems to achieve satisfactory performance. The adaptive control algorithm and the RBF network are designed to compensate the nonlinear effects in the linear motors. As the focus of the thesis, not only the normal Iterative Learning Control (ILC) is addressed but also some new forms of ILC and repetitive control are proposed for the different system characteristics.

Adaptive feedforward compensation of force ripples in linear motors

The presence of force ripples is a highly undesirable phenomenon in the realization of precision motion control in PMLMs. Therefore in this thesis, an adaptive control scheme is proposed to suppress force ripple effects impeding motion accuracy in Permanent Magnet Linear Motors (PMLMs). In the literature, many methods have been proposed to deal with the force ripples by identifying the force ripple model. However, in reality, it is much more complex to model. The force ripples are periodic with displacement along the motor. The ripple can be viewed as a sum of sinusoidal functions with unknown frequencies and amplitudes. Therefore, in this thesis, the displacement periodicity of the ripple is obtained by using a Fast Fourier Transform (FFT) analysis, based on the experimental result. It is a soft approach to identify the frequencies in the closed-loop by analyzing the control signals. The control method is based on Recursive Least Squares (RLS) identification of a nonlinear PMLM model which includes a model of the force ripple. Based on this model, the control algorithm can be commissioned which consists of a PID feedback control component, an adaptive feedforward component for compensation of the force ripple and another adaptive feedforward component based on the inverse dominant linear model which can serve to expedite motion tracking response. Simulation and experimental results are presented to show the effectiveness of the proposed method for high precision motion tracking applications.

Iterative reference adjustment for high-precision and repetitive motion control applications

Friction is another prominent nonlinear effect associated with the PMLM. It is highly nonlinear in nature and difficult to model. In order to compensate the friction and suppress the force ripple, an Iterative Learning Control (ILC) scheme is proposed in this thesis, which is suitable for high precision and repetitive motion control applications. The proposed method is a model free approach and no explicit modeling effort is necessary. It comprises of a self-tuning radial basis function (RBF) network operating in parallel with an iterative learning control (ILC) component. The proposed scheme iteratively adjusts the reference signal. The RBF network is employed as a nonlinear function estimator to model the tracking error over a cycle, and this error model is subsequently used implicitly in the iterative adaptation of the reference signal over the next cycle. The ILC component further enhances the tracking performance, particularly over the sections of the trajectory where the RBF network is less adequate in its modeling function. Simulation examples and real-time experimental results are provided to elaborate the various highlights of the proposed method.

Online automatic tuning of PID controller based on an Iterative Learning Control approach

Proportional-Integral-Derivative (PID) controllers are popularly used in various precision motion control systems. Modern industrial controllers are becoming increasingly intelligent due to more stringent requirements. This thesis proposes an approach for closed-loop automatic tuning of PID controller based on an ILC approach. The method does not require the control loop to be detached for tuning. A modified Iterative Learning Control (ILC) scheme iteratively changes the control signal by adjusting the reference signal only. The PID controller is tuned, based on the satisfactory performance achieved. The proposed method is a model-free approach since no more model effort is necessary. Simulation and experimental results are furnished to illustrate the effectiveness of the proposed tuning method.

Repetitive control for time-delay systems

Time-delay systems are difficult to control to achieve satisfactory performance and stability. In this thesis, a new form of repetitive learning control is proposed which is applicable to the systems with time-delay. A convergence condition which is necessary and sufficient is derived for the proposed scheme. In addition, a robust convergence analysis for the learning control under the existence of a time-delay mismatch, initialization errors, disturbances and measurement noise is also derived to show the robustness of the proposed approach. Simulation example illustrates the practical applications of the results for the systems with time delay.

Predictive and Iterative Learning Control algorithm

An Iterative Learning Control algorithm enhanced with predictive features is developed in this thesis. An error model is introduced which can represent the transition of tracking error in two successive trials. Based on this model, a predictive and Iterative Learning Control algorithm is derived which is only based on the trial number (or repetitive index). A rigorous analysis of the convergence is provided. In addition, the robustness analysis of the algorithm against the modeling error, initial error and disturbances is discussed. To show the effectiveness of the proposed method, an simulation example is provided.

1.4 Organization of thesis

The thesis is organized as follows.

Chapter 2 presents an adaptive control scheme to reduce the force ripple effects in Permanent Magnet Linear Motors. A mathematical model of the linear motors is introduced first. Then, a frequency analysis method is developed to derive the dominant displacement periodicity pertaining to the force ripple. Based on the obtained frequency information, the adaptive control scheme is proposed, including the control configuration and the online estimation method. Finally, the simulation and experimental results are provided, respectively.

In Chapter 3, a learning control scheme is proposed which combines the Radial Basis Function (RBF) neural network with Iterative Learning Control (ILC) together to realize the precision motion control. In this chapter, the proposed control scheme is explained in detail. Then, a convergence analysis for learning algorithm in the discrete-time domain is provided. Following that, the simulation and experimental results are presented to elaborate the viability of the proposed control scheme.

Chapter 4 describes an approach for closed-loop automatic tuning of PID controller based on an ILC method. The detailed tuning procedure is elaborated in this chapter. Based on the achieved satisfactory performance with ILC approach, the PID controller is then tuned. The simulation and experimental results are discussed to reinforce that the proposed PID tuning method is applicable.

Chapter 5 extends the learning control approach to the systems with time delay. The repetitive control configuration for the time-delay systems is discussed first. Then, the general convergence analysis are derived respectively. In the consideration of the model error, disturbances and measurement noise, the robust convergence analysis is further derived in this chapter. Finally, the simulation examples are presented to illustrate the effectiveness of the proposed method.

In Chapter 6, a predictive Iterative Learning Control algorithm is developed. An error model is introduced first. Based on this model, the predictive iterative learning algorithm is derived. Then, the convergence analysis is investigated in this chapter. Simulation examples are given to show the effectiveness of the proposed algorithm.

Finally, conclusions and suggestions for future work are discussed in Chapter 7.

Chapter 2

Adaptive Feedforward Compensation of Force Ripples in Linear Motors

2.1 Introduction

Permanent Magnet Linear Motors (PMLMs), as a specific type of linear motors, are widely used in applications requiring linear motion at high speed and high accuracy. As described in Chapter 1, it is known that the nonlinear effects in PMLMs have a significant effect on the system performance. In the realization of precision motion control in PMLMs, the presence of force ripples is a highly undesirable phenomenon which degrades the achievable positioning accuracy. The force ripples are generated due to the magnetic structure of PMLMs. Figure 2.1 shows the velocity-time response of a PMLM manufactured by Linear Drives Ltd (U.K.) for a constant input voltage signal. Figure 2.2 shows the real-time open-loop step response with the input voltage of 1.2v. From the responses, the presence of force ripples is self-evident and they are periodic with displacement along the motor. These ripples yield problems in achieving

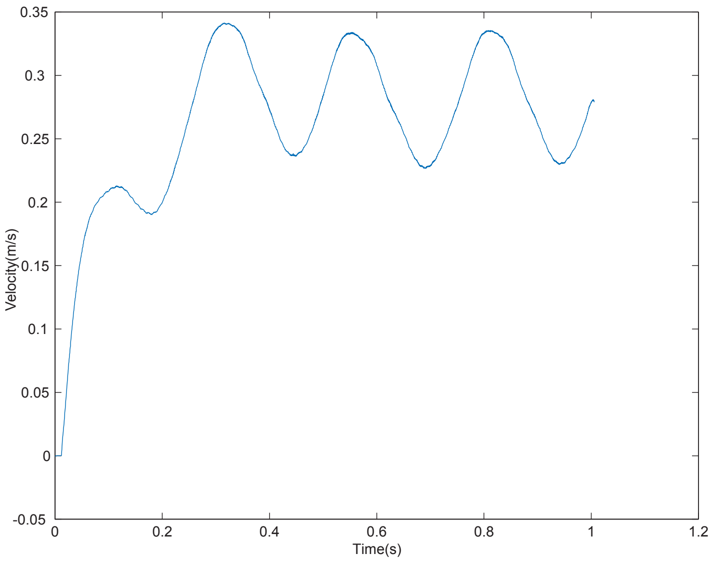


Figure 2.1: Open-loop velocity-time response with input voltage of 0.8V

a smooth and precise motion profile using conventional feedback controllers, since the ripples create “bumps” along the direction of motion.

Some effort has been devoted to suppress the force ripple. A force ripple model was developed and identified with a force sensor, and a feedforward compensation component was used to reduce force ripple [51]. In [28] and [52], a neural-network based learning feedforward controller was applied in the linear motor motion control system. In [53] and [54], an adaptive robust control scheme was proposed for the high speed and high accuracy motion control. In [55] a robust adaptive approach is proposed to compensate the friction and force ripple. In [56], the iterative learning control was applied.

The force ripple phenomenon has been described via a sinusoidal function of the position x [28]. However, in reality, it is much more complex to model. The ripple can constitute the sum total of a number of sinusoidal functions with unknown frequencies

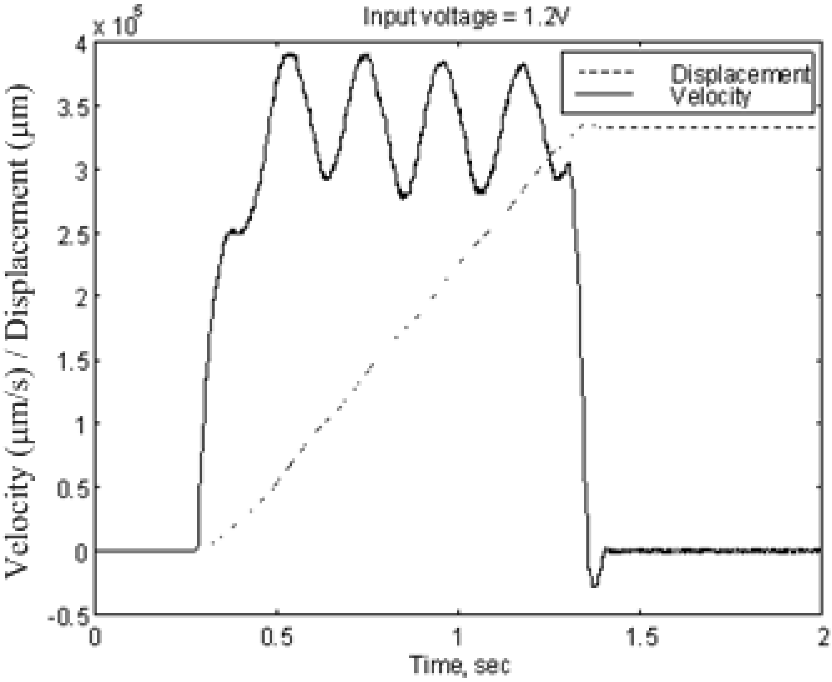


Figure 2.2: Open-loop step response of a PMLM - Displacement (μm) and velocity (μm/s) versus time

and amplitudes. In this thesis, the displacement periodicity of the ripple is determined using a Fast Fourier Transform (FFT) analysis. However, in this case, the periodicity is with respect to displacement and not time. A displacement to time mapping is thus pre-performed in order to directly apply FFT in the usual way. With the spectrum available from the FFT analysis, the dominant frequency components can be extracted. Then, based on an inverse mapping, the displacement periodicity can be derived. Thus, a more accurate model of the force ripples can be built. With the displacement periodicity information available, a model of the PMLM can be posed in the linear regression form to facilitate the application of the Recursive Least Square (RLS) estimation algorithm to identify the remaining model parameters. Based on the model, the control algorithm can also be commissioned. It comprises of a PID feedback control component, an adaptive feedforward component which compensates for the force ripple, and another adaptive feedforward component based on the inverse dominant linear model which serves to speed up the motion tracking response. Simulation and experimental results demonstrate the effectiveness and robustness of the proposed control scheme.

This chapter is devoted to develop an adaptive control method to reduce the force ripple based on the identified frequency information. First, the mathematical model of the PMLM is introduced. Then, a frequency analysis method is developed to derive the dominant displacement periodicity pertaining to the force ripple. Next, the proposed overall control scheme is described, including the control configuration and the online estimation method used to identify the parameters. Finally, the simulation and experimental results are furnished respectively.

2.2 Modeling of the Linear Motor

In this section, a model of the linear motor with parameters specific to an LD series linear motor (LD 3810) is presented. A simplified model which combines the mechanical dynamics and the electrical dynamics is given in [54] and [57]:

$$u(t) = K_e\dot{x} + Ri(t) + Ldi(t)/dt, \tag{2.1}$$

$$f(t) = K_f i(t), \tag{2.2}$$

$$f(t) = M\ddot{x}(t) + f_{ripple}(x) + f_{fric}(\dot{x}) + f_{nl}(t), \tag{2.3}$$

where $u(t)$ and $i(t)$ are the time-varying motor terminal voltage and the armature current, respectively; $x(t)$ is the motor position; $f(t)$ represents the developed force; $f_{fric}(\dot{x})$

Table 2.1: Linear Motor Parameters

Motor	Units	LD 3810
Force Constant(K_f)	N/A	130
Resistance(R)	Ω	16.8
Back EMF(K_e)	V/m/s	123
Length of Travel	mm	2054
Moving Mass(M)	kg	5.4
Armature Inductance(L)	mH	17.4
Electrical Time Constant	msec	1.03
Peak Force(F_p)	N	1300
Peak Velocity	m/sec	2.6
Peak Acceleration	m/sec^2	140
Continuous Current	A	2.5
Continuous Force	N	326
Continuous Working Voltage	V d.c.	320
Continuous Working Power	W	700

and $f_{ripple}(x)$ denotes the friction and ripple force; $f_{nl}(t)$ represents the combined force effects arising from other uncertainty and disturbances present in the linear motor. The physical parameters of a PMLM (LD 3810) are listed in Table 2.1 [58].

Specially, since this chapter focuses on the compensation of the force ripples, $f_{fric}(\dot{x})$ and $f_{nl}(t)$ in (2.3) are ignored. The model of the linear motors for this chapter is simplified as

$$u(t) = K_e\dot{x} + Ri(t) + Ldi(t)/dt, \tag{2.4}$$

$$f(t) = K_f i(t), \tag{2.5}$$

$$f(t) = M\ddot{x}(t) + f_{ripple}(x). \tag{2.6}$$

Since the electrical time constant is low and much smaller than the mechanical one, the delay of electrical response can be ignored. Here Thus, the following equation can be obtained

$$\ddot{x} = (-\frac{K_f K_e}{R}\dot{x} + \frac{K_f}{R}u(t) - f_{ripple}(x))/M. \tag{2.7}$$

Let

$$\begin{aligned} a &= \frac{K_f K_e}{RM}, && (2.8) \\ b &= \frac{K_f}{RM}. && (2.9) \end{aligned}$$

Thus, it can be written as

$$\ddot{x} = -a\dot{x} + bu(t) - \frac{1}{M}f_{ripple}(x). \tag{2.10}$$

Generally, the force ripple can be represented as a sum of a series of harmonics:

$$f_{ripple}(x) = \sum_{k=1}^{J} A_{frk} sin(2\pi k\omega x + \varphi_k), \tag{2.11}$$

where J is the numbers of harmonics and ω is the fundamental frequency.

2.3 Frequency Analysis

One approach to identify the parameters of the ripple model is to conduct experiments with a force sensor as presented in [51]. However, this is a tedious process and is also arguably not effective since it is known that the amplitude of the force ripple varies with velocity. It is also possible to identify these parameters more efficiently via a soft approach by analyzing the control signals in the closed-loop. To this end, the Recursive Least Square (RLS) estimation algorithm (to be highlighted in Section 2.4.2) can be used which requires the model to be posed in a linear regression form. The parameter which causes difficulty to enable such a formulation is the displacement periodicity ω in the ripple model. It is thus necessary to predetermine it.

It may appear that a frequency analysis method may be used for this identification. However, it should be noted that the force ripple is a function of displacement x and not time. If the constant velocity is used in the experiment of frequency analysis, there exists a simple relationship between displacement x and time. A displacement to time mapping should be done prior to direct application of these tools. A direct mapping can be achieved if the motor is run at a constant and low velocity of say $1mm/s$. If the constant velocity can be maintained for sufficiently long travel of the motor, then displacement and time clearly exhibit a direct relationship. An experiment is carried out to demonstrate this point. By running a linear motor at $1mm/s$, the control signal versus time plot is almost identical to the control signal versus displacement plot as shown in Figure 2.3 and Figure 2.4 respectively. There exists only a slight difference between these two figures due to small fluctuations of the velocity from the constant value.

Now, applying the FFT to the control signal, it can be obtained as

$$u_k = f_k = \sum_{n=0}^{N-1} F_n e^{j2\pi(nk)/N} \tag{2.12}$$

$$F_n = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-j2\pi(nk)/N} \tag{2.13}$$

where control signal u is a length N discrete signal.

The frequency spectrum of the control signal can be derived as shown in Figure 2.5

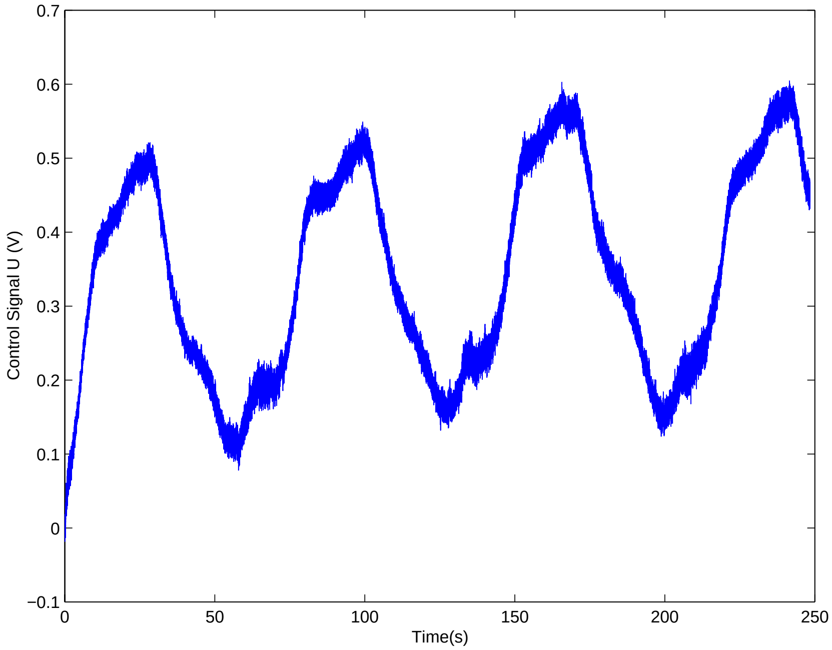


Figure 2.3: Control signal versus time plot

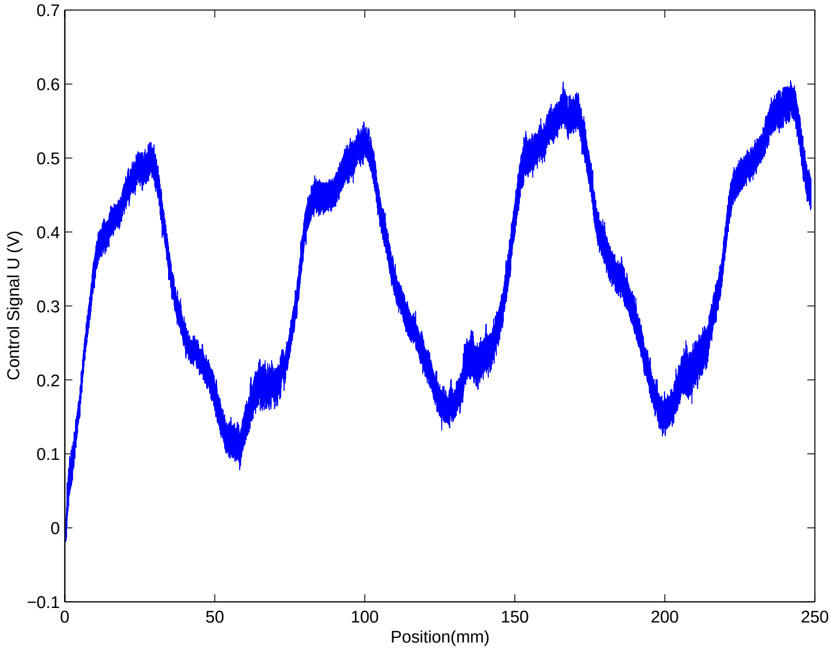


Figure 2.4: Control signal versus displacement plot

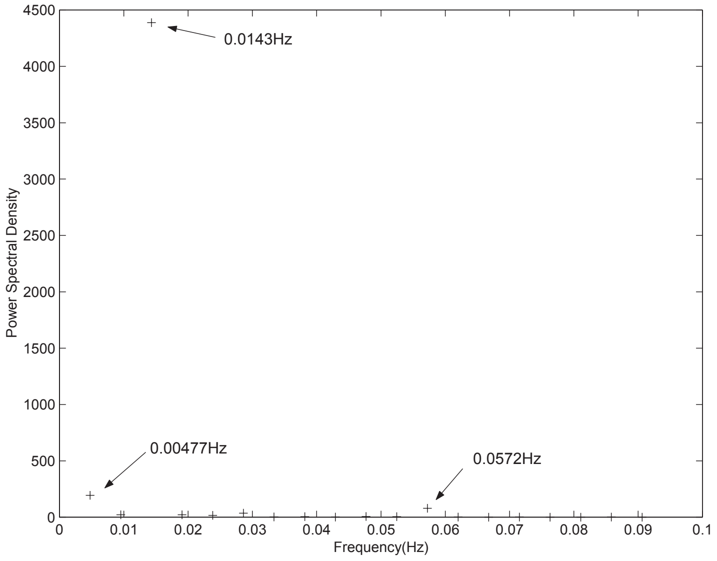


Figure 2.5: Power spectral density of the control signal

in which the DC value is ignored. From the spectrum, the dominant frequency is seen to be $0.0143Hz$, although other harmonics at $0.00477Hz$ and $0.0572Hz$ also appear in the spectrum, but these are less significant in amplitude being less than 10% of the dominant frequency. Therefore, only the dominant frequency is considered in the construction of the ripple model. Since it has a direct mapping from displacement to time, the dominant displacement periodicity is simply given by $0.0143mm^{-1}$. Please note that the displacement x is in the unit mm.

The model of force ripple can thus be described as:

$$\begin{aligned} f_{ripple} &= Asin(2\pi \times 0.0143x + \varphi) \\ &= A_{r1}sin(2\pi \times 0.0143x) + A_{r2}cos(2\pi \times 0.0143x) \end{aligned} \tag{2.14}$$

where A_{r1} and A_{r2} are the amplitudes which have yet to be identified. Thus, the model of the system (2.11) can be rewritten as:

$$\ddot{x} = -a\dot{x} + bu(t) - [A_1 sin(2\pi \times 0.0143x) + A_2 cos(2\pi \times 0.0143x)], \tag{2.15}$$

where $A_1 = A_{r1}/M$ and $A_2 = A_{r2}/M$.

2.4 Proposed Control Scheme

The proposed adaptive control scheme is designed based on the system model. In what follows, the scheme is described systematically in details.

2.4.1 Configuration

Overall, the designed controller is composed of a feedback control component and two adaptive feedforward components. The feedback component is a PID controller which is basically used to ensure nominal stability. The overall feedforward component u_{ff} can be divided into two parts. One part is based on the inverse linear model and this part essentially speeds up the tracking motion response. The other part functions as a compensator for the force ripples. The control signal can be represented as

$$\begin{aligned} u(t) &= u_{ff} + u_{fb} \\ &= u_{ff,inverse} + u_{ff,ripple} + u_{fb}, \end{aligned} \tag{2.16}$$

where $u_{ff,inverse}$ is the feedforward control based on the inverse linear model; $u_{ff,ripple}$ is the feedforward control for the force ripples; u_{fb} represents the feedback control.

Substituting the contributions from the various components, it can be obtained as

$$\begin{aligned} u(t) &= \frac{1}{b}[\ddot{x}_d + a\dot{x}_d] + \frac{1}{b}[A_1 sin(2\pi \times 0.0143x) + A_2 cos(2\pi \times 0.0143x)] \\ &+ [K_p(x_d - x) + K_d(\dot{x}_d - \dot{x}) + K_i \int_0^t (x_d - x)dt]. \end{aligned} \tag{2.17}$$

Note that the parameters, a, b, A_1 and A_2, are unknown and to compute the control action, their estimates are necessary. Replacing these parameters with their estimates (same variables with hat), it follows

$$\begin{aligned} u(t) &= \frac{1}{\hat{b}}[\ddot{x}_d + \hat{a}\dot{x}_d + \hat{A}_1 sin(2\pi \times 0.0143x) + \hat{A}_2 cos(2\pi \times 0.0143x)] \\ &+ [K_p(x_d - x) + K_d(\dot{x}_d - \dot{x}) + K_i \int_0^t (x_d - x)dt]. \end{aligned} \tag{2.18}$$

It should be noted that $\hat{b}$ must be non-zero. In order to identify the parameters online, the Recursive Least Square (RLS) parameter identification can be used. Figure 2.6 shows the full configuration of the proposed method. The whole procedure in the dotted box can be transferred to a smart device which just needs the u and x signals as inputs and outputs the additional control signals ($u_{ff,ripple}$ and $u_{ff,inverse}$). The identification algorithm can described in details in the next section.

Substituting (2.18) into (2.15), the closed-loop system is obtained as

$$\begin{aligned} \ddot{x} &= -a\dot{x} + b[\frac{1}{\hat{b}}(\ddot{x}_d + \hat{a}\dot{x}_d + \hat{A}_1 sin(2\pi \times 0.0143x) + \hat{A}_2 cos(2\pi \times 0.0143x)) \\ &+ K_p(x_d - x) + K_d(\dot{x}_d - \dot{x}) + K_i \int_0^t (x_d - x)dt] \\ &- [A_1 sin(2\pi \times 0.0143x) + A_2 cos(2\pi \times 0.0143x)]. \end{aligned} \tag{2.19}$$

Ideally, the parameter estimates converge to the real parameters perfectly. Then, introducing the tracking error e by replacing $x = x_d - e$, it follows that the closed-loop

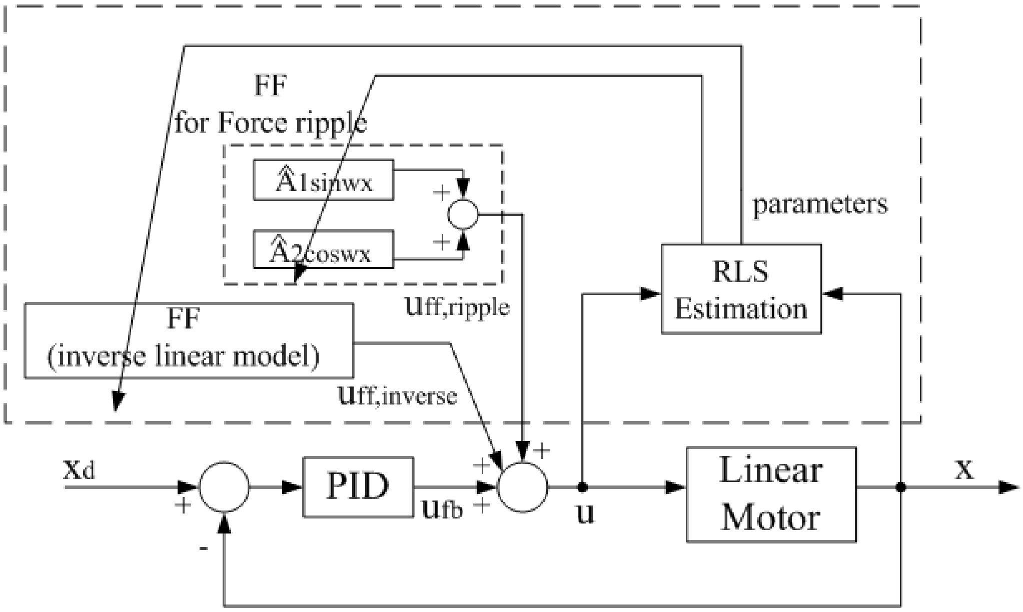


Figure 2.6: Configuration of the proposed method

is reduced to

$$\ddot{e} = -a\dot{e} - b(K_p e + K_d \dot{e} + K_i \int_0^t edt). \tag{2.20}$$

This nominal system is a linear one. It remains to design the PID parameters to ensure nominal stability and performance.

2.4.2 Identification

The parameter identification is based on the model as described in (2.15). With the displacement periodicity identified, the model can be posed in the linear regression form with the parameter vector $\hat{\boldsymbol{\theta}} = [\hat{a} \quad \hat{b} \quad \hat{A}_1 \quad \hat{A}_2]^T$, and the regression vector $\boldsymbol{\phi} = [-\dot{x} \quad u \quad -sin(2\pi \times 0.0143x) \quad -cos(2\pi \times 0.0143x)]^T$. The standard RLS identification algorithm [59] is used which is described by the following equations:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}(t) &= \hat{\boldsymbol{\theta}}(t-1) + \boldsymbol{K}(t)(\ddot{x} - \boldsymbol{\phi}^T(t)\hat{\boldsymbol{\theta}}(t-1)) && (2.21) \\
\boldsymbol{K}(t) &= \boldsymbol{P}(t)\boldsymbol{\phi}(t) = \boldsymbol{P}(t-1)\boldsymbol{\phi}(t)(\boldsymbol{I} + \boldsymbol{\phi}^T(t)\boldsymbol{P}(t-1)\boldsymbol{\phi}(t))^{-1} && (2.22) \\
\boldsymbol{P}(t) &= \boldsymbol{P}(t-1) - \boldsymbol{P}(t-1)\boldsymbol{\phi}(t)(\boldsymbol{I} + \boldsymbol{\phi}^T(t)\boldsymbol{P}(t-1)\boldsymbol{\phi}(t))^{-1}\boldsymbol{\phi}^T(t)\boldsymbol{P}(t-1) \\
&= (\boldsymbol{I} - \boldsymbol{K}(t)\boldsymbol{\phi}^T(t))\boldsymbol{P}(t-1). && (2.23)
\end{aligned}$$

The choice of the initial values of $\hat{\boldsymbol{\theta}}(0)$ and $\boldsymbol{P}(0)$ follow the usual convention. $\hat{\boldsymbol{\theta}}(0)$ can be chosen as $[0\ 1\ 0\ 0]^T$ if no other prior information is available. In order to guarantee the convergence of the RLS algorithm, $\boldsymbol{P}(0)$ should be positive definite and sufficiently large.

Since in real-time applications, usually only the position measurement is available from the optical encoder, the velocity signal needed in the identification cannot be obtained directly. It is not recommended that the velocity is derived using a pure differentiation of the position owing to the presence of measurement noise. The following approach is adopted to make the RLS identification viable without the availability of velocity measurements [60].

Equation (2.15) can be expressed in a general form:

$$A(p)x(t) = bu(t) - [A_1 sin(2\pi \times 0.0143x) + A_2 cos(2\pi \times 0.0143x)], \tag{2.24}$$

where $A(p) = p^2 + ap$, and p is the differential operator $p = d/dt$. The model (2.15) may thus be rewritten as:

$$\begin{aligned} H_f(p)A(p)x(t) &= H_f(p)bu(t) - H_f(p)[A_1 sin(2\pi \times 0.0143x) \\ &\quad +A_2 cos(2\pi \times 0.0143x)], \qquad (2.25) \end{aligned}$$

where $H_f(p)$ is a stable transfer function with a pole excess of two, and it functions as a filter.

Let

$$\begin{aligned} x_f(t) &= H_f(p)x(t) \\ u_f(t) &= H_f(p)u(t) \\ sin_f(t) &= H_f(p)sin(2\pi \times 0.0143x) \\ cos_f(t) &= H_f(p)cos(2\pi \times 0.0143x). \qquad (2.26) \end{aligned}$$

The model can be rewritten as

$$\begin{aligned} A(p)x_f(t) &= bu_f(t) - A_1 sin_f(t) - A_2 cos_f(t) \qquad (2.27) \\ \Longrightarrow \ddot{x}_f &= -a\dot{x}_f + bu_f(t) - A_1 sin_f(t) - A_2 cos_f(t). \qquad (2.28) \end{aligned}$$

Hence, the parameter vector remains as $\hat{\boldsymbol{\theta}} = [\hat{a} \;\; \hat{b} \;\; \hat{A}_1 \;\; \hat{A}_2]^T$. The regression vector becomes $\boldsymbol{\phi_f} = [-\dot{x}_f \;\; u_f \;\; -sin_f \;\; -cos_f]^T$. The controller is still implemented as described in the earlier section. Figure 2.7 shows the proposed scheme of the identification with the filters $H_f(p)$ and controller. In the figure, $\omega = 2\pi \times 0.0143$. In the simulation study of the next section, the simulation results will be provided to show the performance of the proposed method.

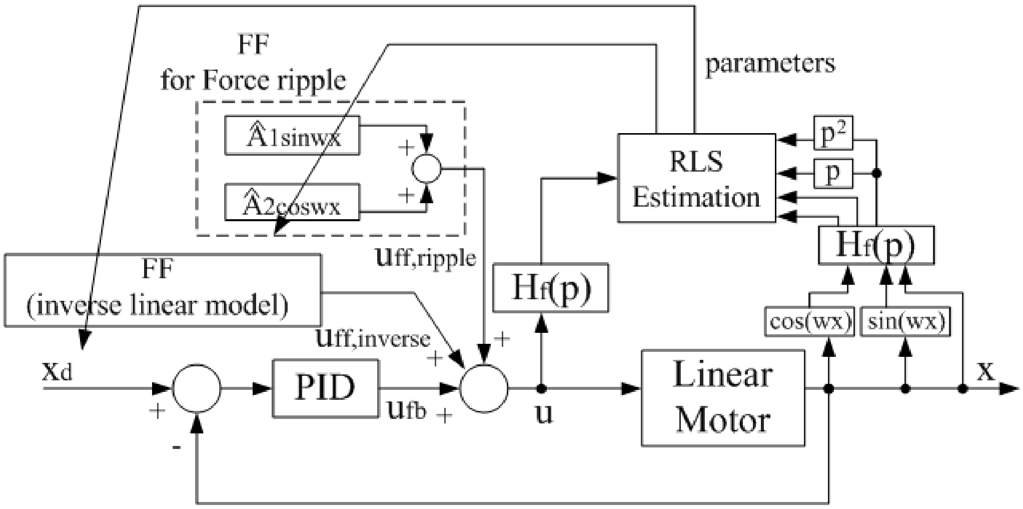


Figure 2.7: Block diagram of overall scheme with filter and control

2.5 Simulation Study

In this section, simulation results are provided pertaining to the performance of the proposed scheme. MATLAB/SIMULINK was used to carry out the simulation work. The simulation study was conducted with respect to the model description (2.15) with $K_f = 130N/A$, $K_e = 123V/m/s$, $R = 16.8\Omega$, $M = 5.4kg$, $L = 17.4mH$ according to the manufacturer specifications for a PMLM (model LD3810) from Linear Drive, thus giving $a = 176.2566$ and $b = 1.433$. The force ripple is also considered with an amplitude of $8N$ and a periodicity of $0.0143mm^{-1}$, i.e.,

$$f_{ripple} = 8\sin(2\pi \times 0.0143x).$$

The initial values of $\hat{\boldsymbol{\theta}}(0)$ and $\boldsymbol{P}(0)$ are chosen as $[0 \;\; 1 \;\; 0 \;\; 0]^T$ and $10^5\boldsymbol{I}$. The filter is designed as

$$H_f(s) = \frac{2400}{s^2 + 110s + 2400}.$$

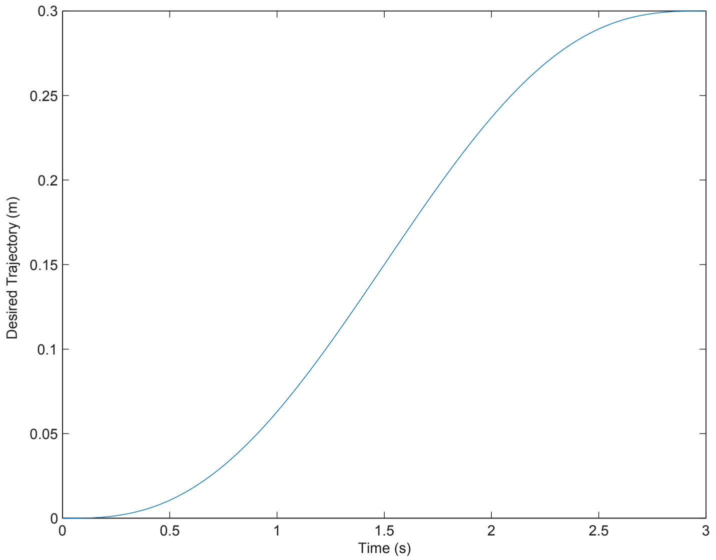


Figure 2.8: Desired trajectory

The desired trajectory is chosen as

$$x_d(\tau) = x_b + (x_b - x_f)(15\tau^4 - 6\tau^5 - 10\tau^3), \tag{2.29}$$

where $\tau = t/(t_f - t_b)$ and the various parameters are set as: initial position $x_b = 0m$, final position $x_f = 0.3m$, $t_b = 0s$ and $t_f = 3s$. The desired trajectory is shown in Figure 2.8.

Figure 2.9 shows the tracking error incurred with only PID control. Figure 2.10 shows the tracking performance with the proposed method. Compared to Figure 2.9, order of 10^{-2} reduction in error can be achieved. Figure 2.10 shows significant improvement obtained in the control performance. Figure 2.11 further confirms the parameter estimates are obtained accurately with $a = 176.259$, $b = 1.433$, $A_1 = 1.481$ and $A_2 = 0$.

In the above study, no unstructured modeling error is considered. To verify the robust-

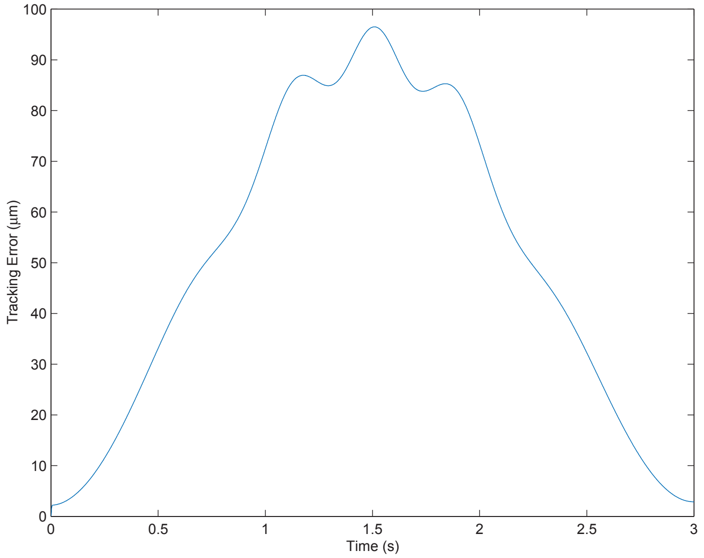


Figure 2.9: Tracking error with only PID control

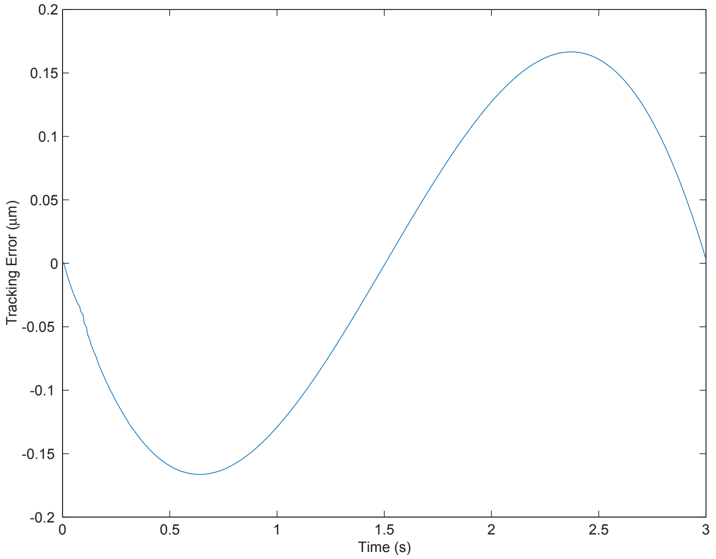


Figure 2.10: Tracking error with the proposed control scheme

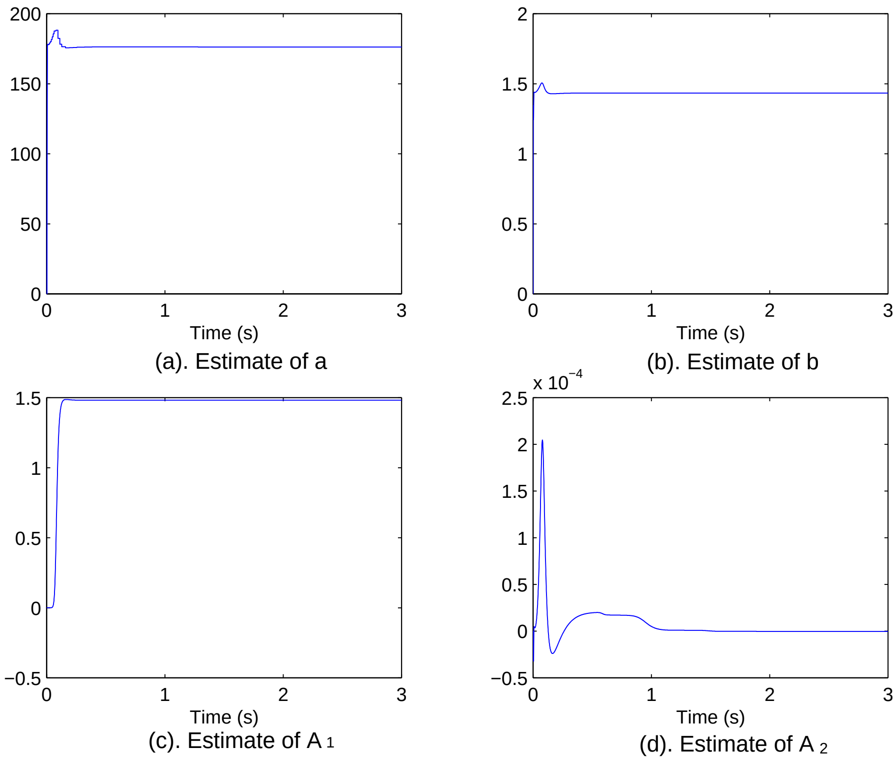


Figure 2.11: Identified parameters: $\hat{a}$, $\hat{b}$, $\hat{A}_1$, $\hat{A}_2$

ness of the proposed control scheme under practical conditions, four kinds of disturbances in the model are introduced and simulated. First, the full model for the actual system (2.4)-(2.6) is simulated, although the reduced model (2.10) continues to be used for the control design. Secondly, measurement noise is deliberately simulated in the system. Thirdly, an actual force ripple phenomenon is simulated, which is given by

$$f_{ripple} = 8sin(2\pi \times 0.0143x) + 0.4sin(2\pi \times 0.0286x), \tag{2.30}$$

although the simpler single sinusoidal function continues to apply to the control system. Finally, possible errors arising from a FFT analysis is allowed and it is assumed that the displacement periodicity of the force ripple is determined to be $0.015mm^{-1}$ instead of $0.0143mm^{-1}$.

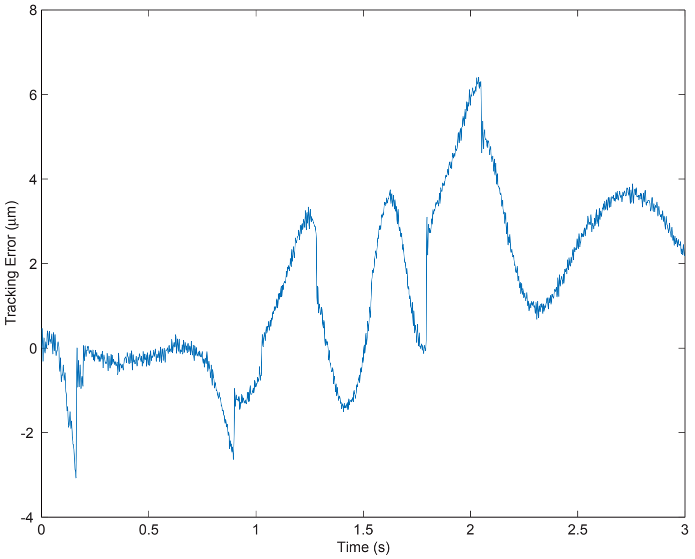
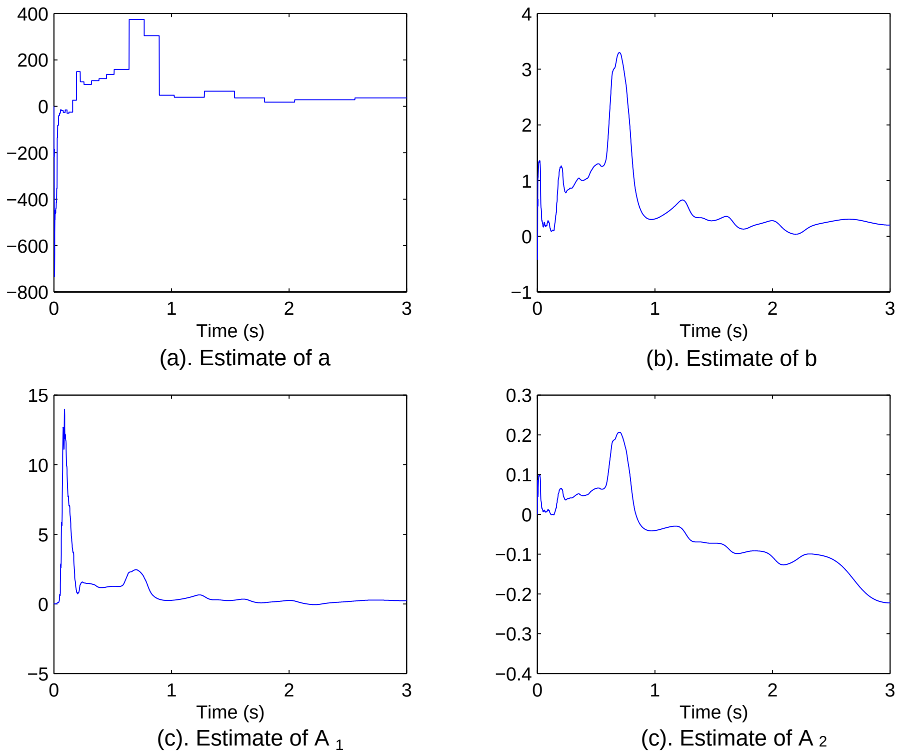


Figure 2.12: Tracking error with the proposed control scheme (with disturbances simulated)

Figure 2.12 shows the tracking performance under the presence of these uncertainties and disturbances. Compared to Figure 2.9, order of 10^{-1} reduction in error is achieved. Figure 2.13 shows the estimates of the parameters. The proposed control continues to achieve satisfactory tracking performance, demonstrating a satisfactory robust performance.

2.6 Experimental Results

To illustrate the applicability of the control scheme to a real system operating under practical conditions, experiments are conducted based on an actual PMLM (LD 3810) manufactured by Linear Drives Ltd (U.K.). The motor components of an LD consist of the thrust rod, thrust block and robotic motor cable. The aluminum thrust block contains a series of cylindrical coils, forming the stator of the motor. The thrust rod



(a). Estimate of a

(b). Estimate of b

(c). Estimate of A 1

(c). Estimate of A 2

Figure 2.13: Identified parameters:$\hat{a}$, $\hat{b}$, $\hat{A}_1$, $\hat{A}_2$ (with disturbances simulated)

contains high-energy permanent magnet pieces within a stainless steel tube. For the system studied, a PWM amplifier with built-in electronic commutation is used to produce a force proportional to the control signal. This is the same motor for which the FFT analysis was done earlier to extract the displacement periodicity. The control scheme is implemented on a dSPACE AlphaCombo system, which is a multiprocessor system. The DS1004 Alpha Board is the main computational platform while a DS1003 DSP Board handles the I/O tasks. A C-coded S-Function is written to perform the control algorithm. Additionally, the proposed algorithms can be downloaded to a dedicated computing device or DSP card. A sampling frequency of 2.5kHz is configured. The resolution of the Renishaw optical encoder used is $1\mu m$. In the real-time experiments,

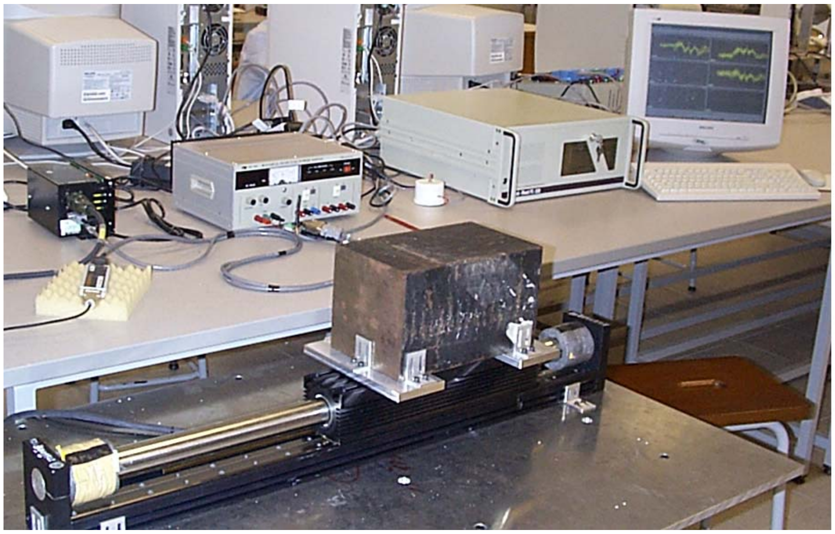


Figure 2.14: Experimental set-up

the PID feedback controller is coarsely tuned with parameters $K_p = 0.045$, $K_i = 0.035$ and $K_d = 0.0001$. Figure 2.14 shows the experimental set-up.

Since the nonlinear effects due to friction is not considered in the proposed scheme, it is an additional source of modeling error especially when the motor is run at high speed. Thus, to demonstrate the adequateness and robustness of the proposed scheme, the experimental results for low-velocity and relatively high-velocity scenarios are illustrated respectively. In addition, the experimental results with only inverse control for the linear dynamic model are provided to help show the effectiveness of the force ripple component by comparison with the complete proposed control method.

The desired trajectory for low speed motion is shown in Figure 2.15. In this case, a maximum velocity is $v_{max} = 0.094m/s$ and maximum acceleration is $a_{max} = 0.096m/s^2$. The tracking error with only PID control is shown in Figure 2.16 with a maximum tracking error of $15.4\mu m$.

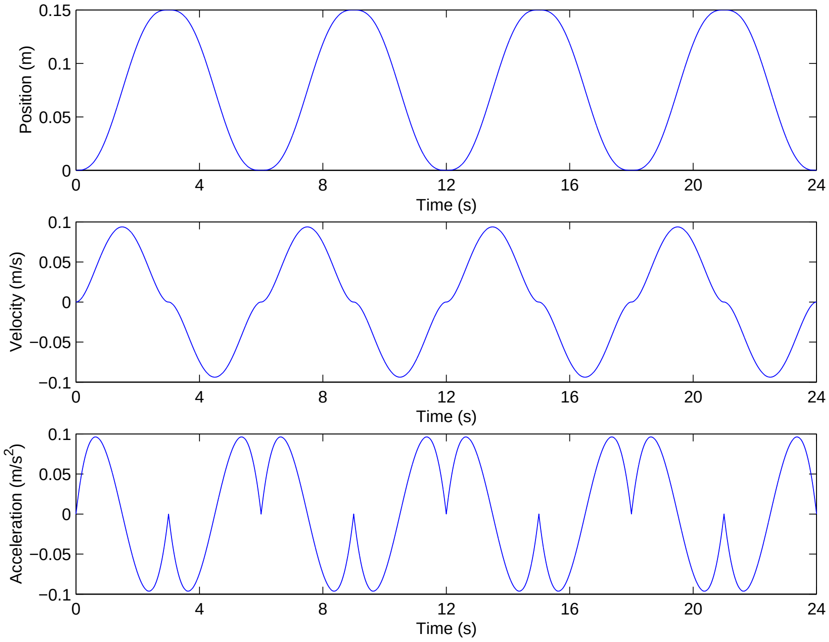


Figure 2.15: Desired motion trajectory at low speed

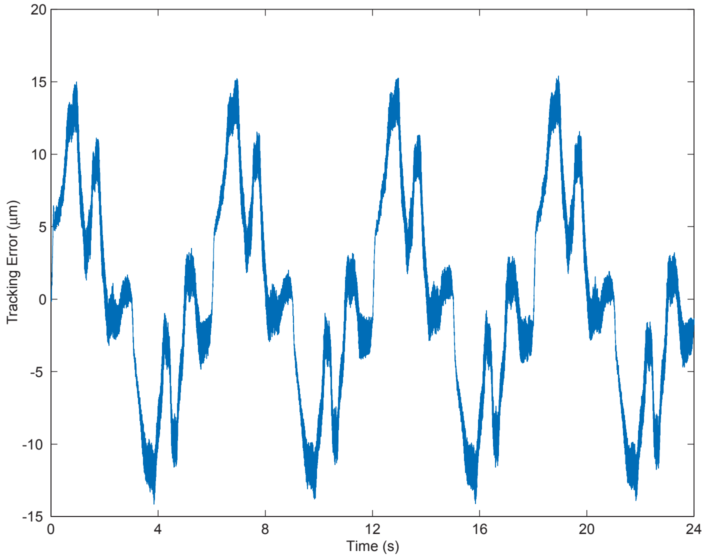


Figure 2.16: Tracking error with PID control (low speed motion trajectory)

The initial parameter estimates are chosen as $\hat{\boldsymbol{\theta}}(0) = [6.5 \;\; 3 \;\; 0.21 \quad -0.12]^T$, and the initial P matrix is fixed at $\boldsymbol{P}(0) = 10^3\boldsymbol{I}$. Since the actual dynamics of the simulated model and the experimental setup are slightly different, the different filter used in the experiment is

$$H_f(s) = \frac{500}{s^2 + 60s + 500}.$$

Figure 2.17 shows the tracking error only with the inverse control based on the dominant linear model. The tracking error with the complete proposed control scheme is shown in Figure 2.18, and Figure 2.19 displays the parameters estimates under this scheme. Compared to Figure 2.16, order of 0.4×10^0 reduction in error is achieved.

For performance comparison, the RMS (root-mean-square) tracking error e_{RMS} and the absolute maximum tracking error e_{MAX} can be used as an indicators to evaluate the tracking performance over the full cycle. A comparison among the three methods (PID only, inverse control for the dominant linear model, and the complete proposed method) is done in terms of e_{MAX} and e_{RMS}. Figure 2.20 and Figure 2.21 compare the maximum tracking error and the RMS tracking error respectively. The figures show that significant improvement, in terms of both maximum and RMS error, can be obtained with the complete proposed control scheme. Additionally, they also verify that the adaptive force ripple compensator is effective towards enhancing the tracking performance.

For the high speed case, the desired motion trajectory is shown as Figure 2.22. It gives rise to a maximum velocity of $v_{max} = 0.75m/s$ and a maximum acceleration of $a_{max} = 4.62m/s^2$. The same $\hat{\boldsymbol{\theta}}(0)$ and $\boldsymbol{P}(0)$ are applied to the adaptive controller.

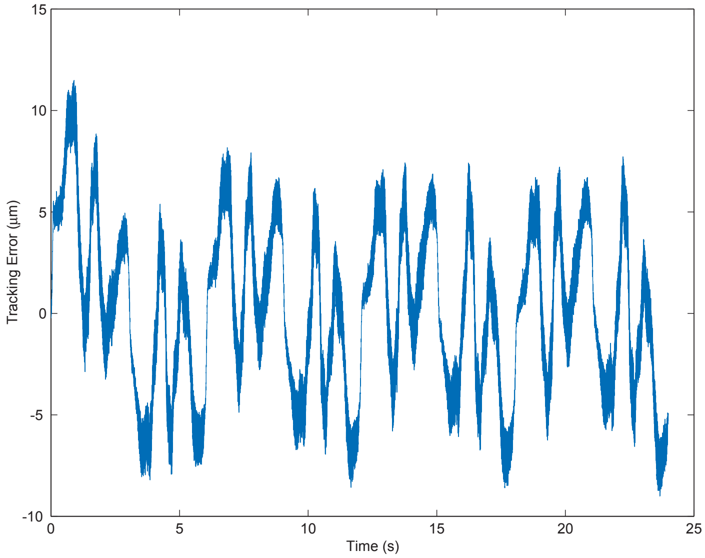


Figure 2.17: Tracking error only with the inverse control for the linear model (low speed motion trajectory)

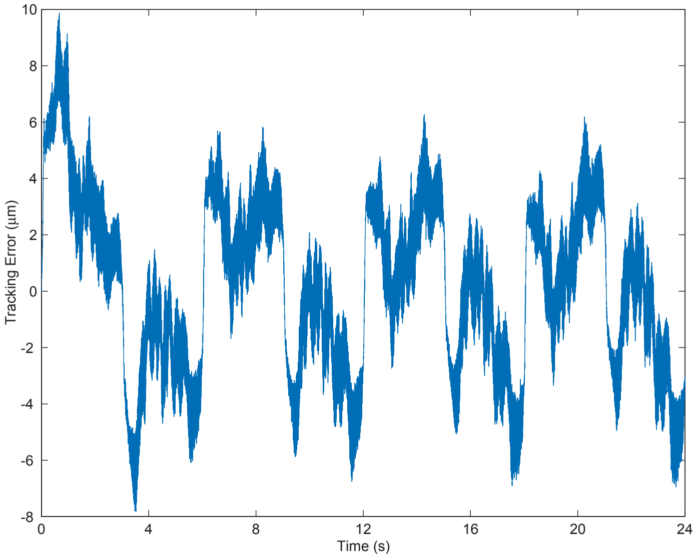
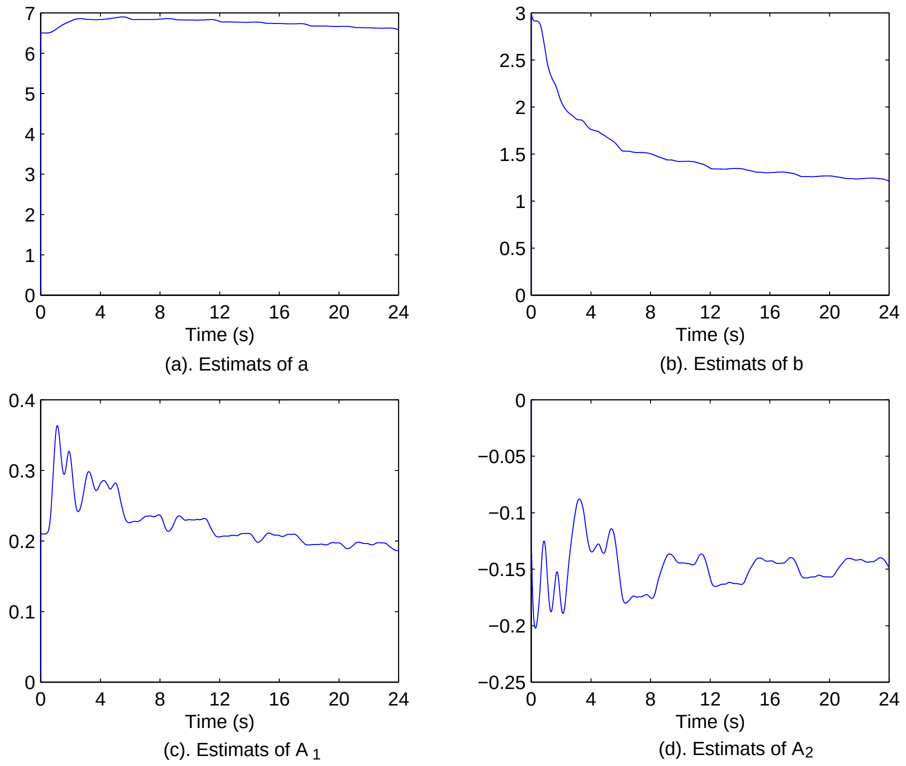


Figure 2.18: Tracking error with the proposed control scheme (low speed motion trajectory)



(a). Estimats of a

(b). Estimats of b

(c). Estimats of A 1

(d). Estimats of A2

Figure 2.19: On-line identified parameters:$\hat{a}$, $\hat{b}$, $\hat{A}_1$, $\hat{A}_2$ (low speed motion trajectory)

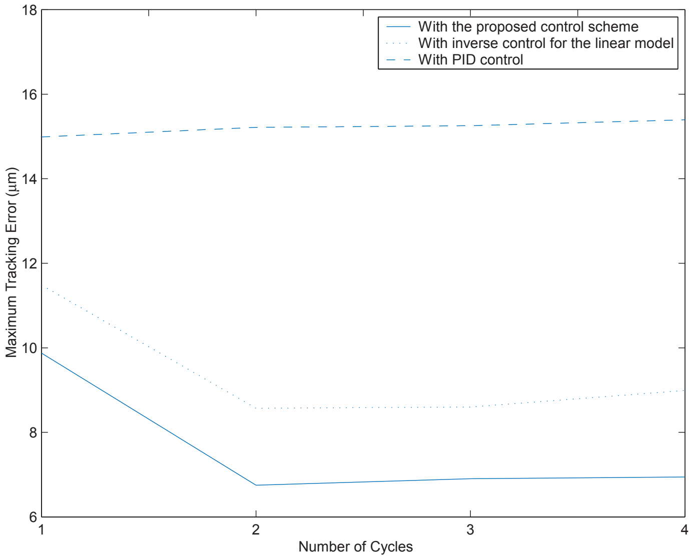


Figure 2.20: Comparison of the maximum tracking error (low speed motion trajectory)

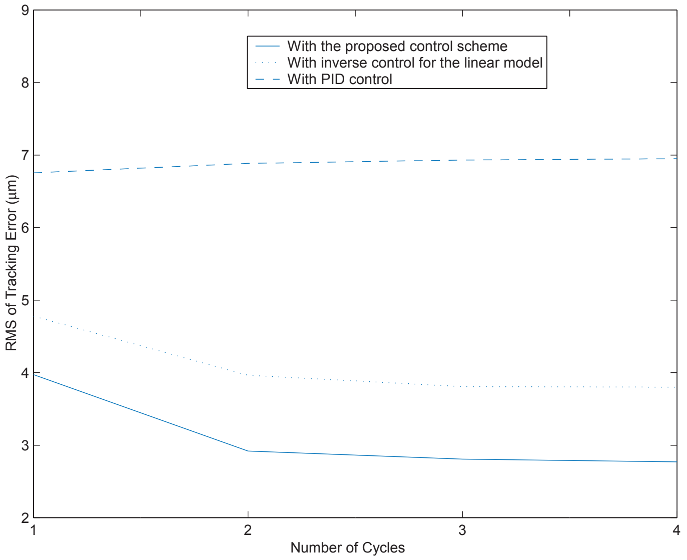


Figure 2.21: Comparison of the RMS tracking error (low speed motion trajectory)

The tracking error incurred by the respective control schemes are shown in Figure 2.23, Figure 2.24 and Figure 2.25. Compared to Figure 2.23, order of 0.4×10^0 reduction in error is achieved.

Figure 2.26 shows the parameter estimates under the proposed scheme. Similar to the low speed case, a comparison in terms of maximum and RMS error is given in Figure 2.27 and Figure 2.28 respectively. In terms of RMS error, the proposed scheme continues to exhibit the best performance among the three schemes. However, in terms of the maximum error, it does not perform better than the inverse control based on the dominant linear model. This phenomenon may be attributed to the variation of the amplitude of the force ripple with velocity. According to [28], force ripple has a high frequency at high speed which can fall beyond the control bandwidth of the system. Under this circumstance, the adaptive method cannot cope effectively, so that a larger

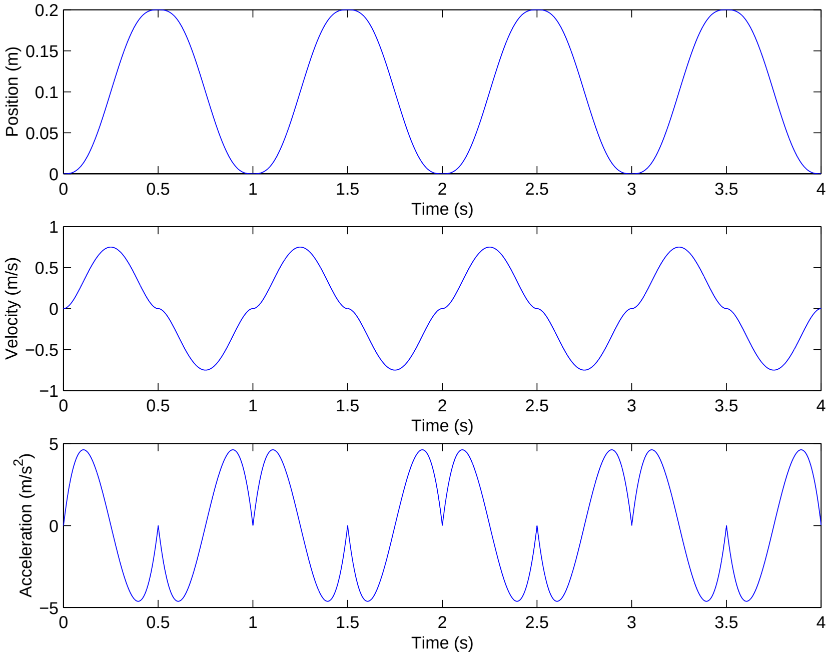


Figure 2.22: Desired motion trajectory at high speed

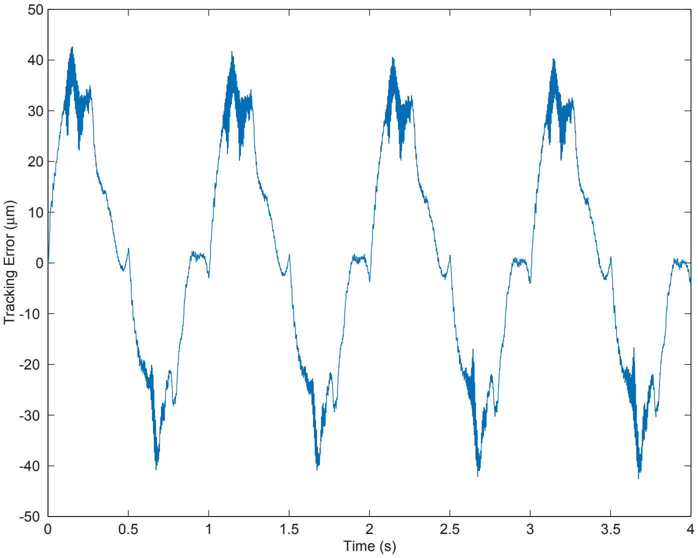


Figure 2.23: Tracking error with PID control (high speed motion trajectory)

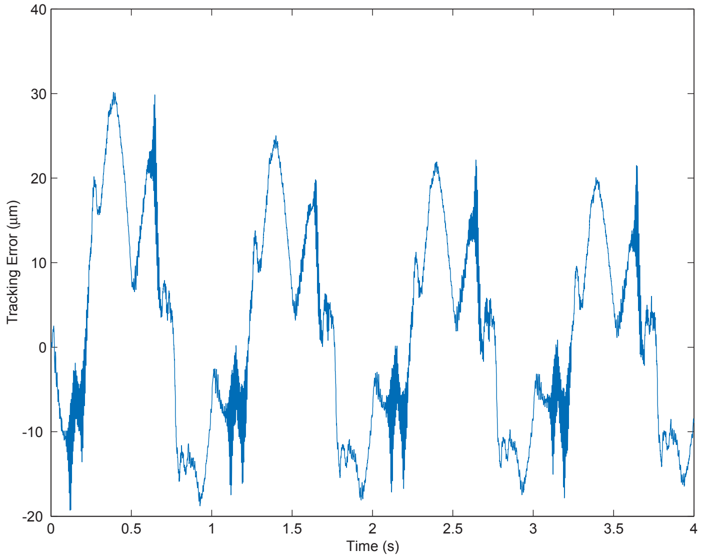


Figure 2.24: Tracking error only with the inverse control for the linear model (high speed motion trajectory)

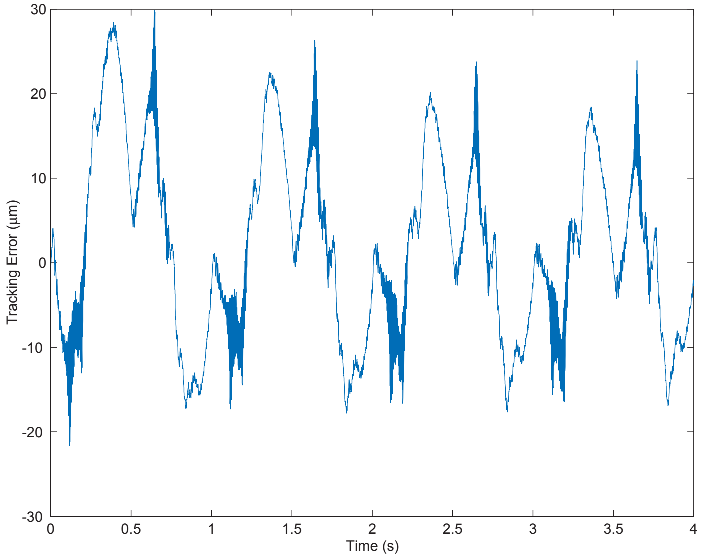
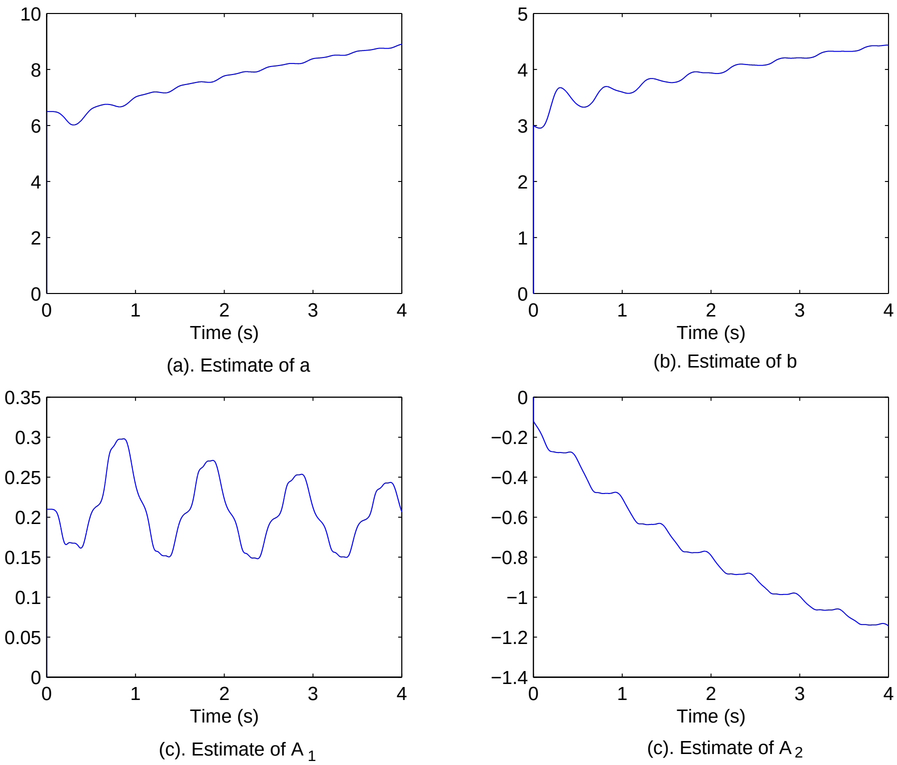


Figure 2.25: Tracking error with the proposed control scheme (high speed motion trajectory)



(a). Estimate of a

(b). Estimate of b

(c). Estimate of A $_1$

(c). Estimate of A $_2$

Figure 2.26: On-line identified parameters:$\hat{a}$, $\hat{b}$, $\hat{A}_1$, $\hat{A}_2$ (high speed motion trajectory)

maximum error arising during the transient phase can occur. However, the average tracking error over the entire cycle still remain to be the lowest for the proposed scheme. Moreover, the result performance can be further analyzed in terms of baseline error [61].

2.7 Conclusions

In this chapter, an adaptive control scheme has been developed to reduce force ripple effects impeding motion accuracy in Permanent Magnet Linear Motors (PMLMs). The control method is based on Recursive Least Square (RLS) identification of a nonlinear PMLM model which includes a model of the force ripple. Based on this model, the control algorithm can be commissioned which consists of a PID feedback component,

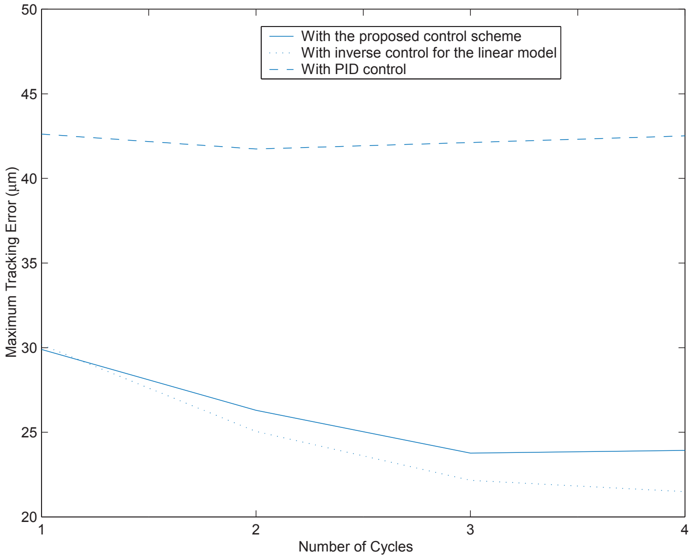


Figure 2.27: Comparison of the maximum tracking error (high speed motion trajectory)

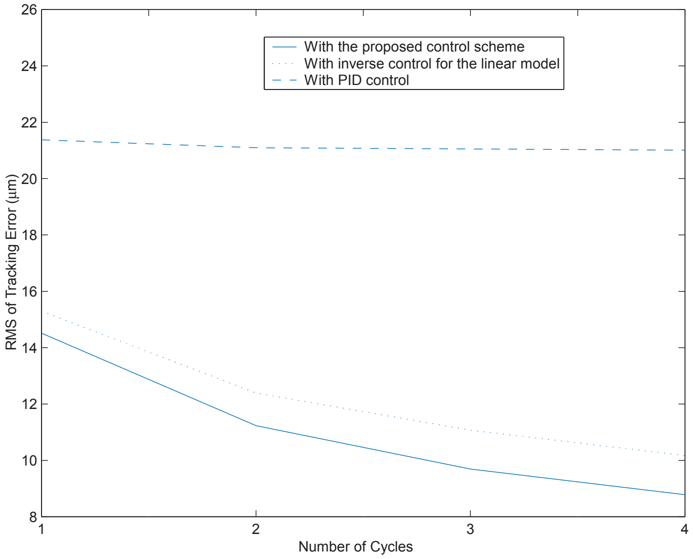


Figure 2.28: Comparison of the RMS tracking error (high speed motion trajectory)

an adaptive feedforward component for compensation of the force ripple and another adaptive feedforward component which may serve as the inverse model of the dominant PMLM model. Simulation and experimental results have been presented which verify the effectiveness of the proposed control scheme for high precision motion tracking applications.

Chapter 3

Iterative Reference Adjustment for High Precision and Repetitive Motion Control Applications

3.1 Introduction

In Chapter 2, the development of the adaptive controller concentrates on the reduction of the force ripples. Actually, friction is also a highly undesirable phenomenon in PMLMs. In this chapter, all the nonlinear effects as shown in (2.1)-(2.3) are considered in the control algorithm design. Friction may be reduced to some extent via the use of more efficient bearings, such as air or magnetic bearings. The force ripples, which may cause oscillation and yield stability problems [2], may be minimized via a reduced-iron magnetic core design, usually at the expense of a lower generated force.

In high precision application domains, conventional Proportional-Integral-Derivative (PID) control usually does not suffice since the need to adequately compensate the nonlinear dynamics of the system become even more important in these applications. Although model-based control strategies to deal with the modeling of these nonlinear effects can be considered, it is generally acknowledged that it is a difficult and challenging to model these nonlinear effects explicitly and accurately. Alternatively, self-learning schemes may be considered [28] [56] [62]. This chapter presents a learning control scheme which is suitable for high precision and repetitive motion control applications, such as those encountered in pick and place precision assembly, or fixed sequence robotic machining processes. For example, in automated manufacturing or assembly, such as flip-chips assembly, high precision repetitive pick and place operations are necessary to yield high density devices. For the proposed method, no explicit modeling effort is necessary. It comprises a Radial Basis Function (RBF) network operating in parallel with an Iterative Learning Control (ILC) component. In the literature, there are some works about iterative learning control based on neural network. In [62], an iterative learning controller using neural network was proposed for the robot trajectory tracking problem. In this work, the neural network was trained off-line iteratively. In [63], a learning feedforward controller was developed based on the B-Spline network. The learning feedforward controller was updated by the filtered feedback control signal. The network was used to derive the filter. For the proposed method, the RBF network is trained on-line. Moreover, the actions of RBF and ILC components iteratively adjust the reference signal, which was discussed by Longman in [15]. This is a useful practical feature deliberately put in place here, since most off-the-shelf industrial controllers do not allow any manipulation of the control signal which is mostly closed to the users. Many of them, however, permit user specification of the reference signal, and its online modification to facilitate correction processes such as those used for the compensation of machine geometrical errors and/or encoder errors. Thus, the proposed approach may be more easily incorporated into existing control systems. The proposed scheme uses a RBF network [64] to model the tracking error over a cycle which is subsequently used implicitly in the iterative adaptation of the reference signal over the next one. In this way, the learning requirements on the ILC component to achieve a certain degree of tracking accuracy can be reduced, thus speeding up the overall convergence speed. The weights of the RBF are recursively adjusted online based on the remnant tracking error from cycle to cycle. The ILC component further enhances the tracking component, especially over the sections of the trajectory where the RBF network does not approximate the error well.

Simulation examples and experimental results are provided to demonstrate the effectiveness and practical viability of the proposed control scheme.

3.2 Proposed Control Scheme

The overall configuration of the proposed control scheme is shown in Figure 3.1.

It comprises a basic feedback-feedforward controller (shown in the shaded portion of Figure 3.1), typically found in standard industrial motion control systems, as well as a Reference Adjustment Mechanism (RAM) comprising of a combined RBF-ILC system which adjusts the reference signal from one cycle to the next. The basic feedback-feedforward controller is termed as the Standard Controller henceforth. In the following

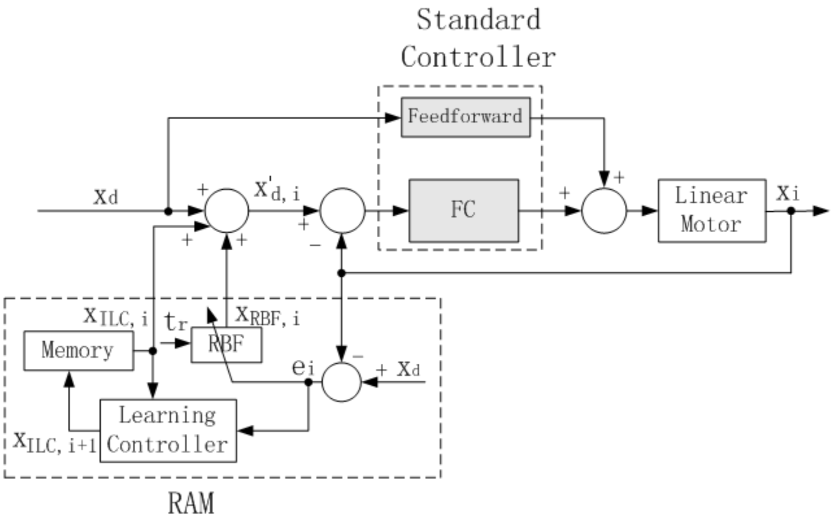


Figure 3.1: Proposed combined RBF-ILC strategy (RBF-ILC scheme)

sections, PD controller is utilised as the feedback controller while the inverse of the linear model is designed as the linear feedforward controller. The RAM represents the learning mechanism for the control system, and it constitutes the main contribution from the thesis. The main function of RAM is to learn and model the remnant tracking error which the standard controller cannot adequately compensate, and use the model to modify the reference signal to achieve a tighter tracking performance.

In what follows, the functions of the RBF network and the ILC component in the RAM is separately explained, followed by an elaboration of the RBF-ILC combination.

3.2.1 Radial Basis Function Network

The proposed control system assumes that the system is essentially time-invariant and the desired trajectory is a periodic one with period T_p. Thus, with the standard controller in place, the remnant tracking error is a periodic signal with the same period T_p.

Define $f_e(t)$ as the remnant tracking error incurred under the standard controller. A key motivation of the proposed control scheme is to approximate and model $f_e(t)$ over a cycle of the operation and use this model to modify the desired trajectory for the standard controller in the subsequent cycles, so as to significantly reduce the tracking error. The remnant tracking error $f_e(t)$ is expected to be a highly nonlinear function of time. An adequate modeling tool for nonlinear functions should thus be considered.

The RBF neural network is a type of feedforward neural network. It has been shown that under mild assumptions, RBF networks are capable of universal approximations [64], i.e., approximating any continuous function over a compact set to any degree of accuracy. Therefore, RBF networks is used to approximate the nonlinear functions associated with the tracking errors. These errors may be viewed as arising from the nonlinear and uncertain dynamics which have not been adequately addressed by the standard controller. In a way, the RBF network can be viewed as functioning as a nonlinear reference corrector to improve on the tracking accuracy. Generally, the RBF network can be represented as

$$F(x) = \sum_{j=0}^{L} \omega_j \phi_j(\| x - c_j \|), \tag{3.1}$$

where $\{\phi_j(\|x - c_j\|)\}$ are the basis functions, L represents the number of weights of the network, $\{\omega_j\}$ are the linear weights and $\{c_j\}$ represent the centers of the radial basis functions. A common form of the RBF is the Gaussian function described by $\phi_j(\|x - c_j\|) = exp(-\frac{\|x-c_j\|^2}{2\sigma_j^2})$ [2], [65], where $\{\sigma_j\}$ represent the spreads of the basis function.

Figure 3.2 shows the use of the RBF network to modify the desired trajectory under the

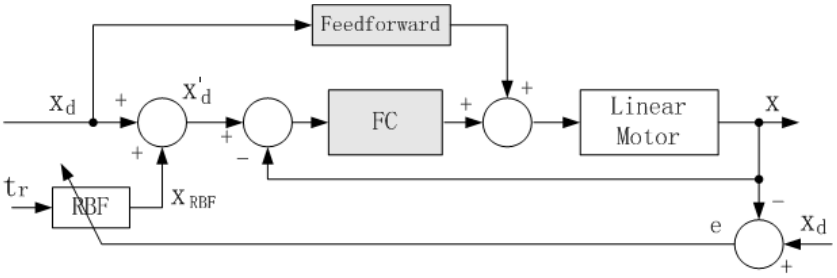


Figure 3.2: Standard control with RBF network (RBF-only scheme)

standard control strategy. Let x_{RBF} denote the output of RBF network. The remnant tracking error $f_e(t)$ can thus be represented as

$$f_e(t) = x_{RBF}(t) + \epsilon, \tag{3.2}$$

with $|\epsilon| \leq \epsilon_M$ where ϵ_M is a bound on the tracking error residual which is not captured by the RBF model.

The reference input to the standard controller is the modified desired trajectory $x'_d = x_d + x_{RBF}(t)$, where x_d is the user specified desired trajectory. This RBF network models a nonlinear time function. Thus, the output of the RBF network $x_{RBF}(t)$ is given by

$$\begin{aligned} x_{RBF}(t_r) &= \sum_{j=0}^{L} \omega_j \phi_j(|t_r - c_j|) \\ &= \sum_{j=0}^{L} \omega_j exp(-\frac{|t_r - c_j|^2}{2\sigma_j^2}), \end{aligned} \tag{3.3}$$

where t_r $(0 \leq t_r < T_p)$ represents the time instant relative to the beginning of each tracking cycle.

Choice of RBF network parameters:

Before a RBF network can be trained and used, a number of structural parameters have to be specified or pre-determined.

The selection of the number of weights L is linked to the complexity of $f_e(t)$ and the required accuracy threshold of the model ϵ_M. Although a larger L normally results in a better fitting model, it also incurs higher computational requirements in order to update the weights within a specific time frame of the application. Thus, a tradeoff in update speed and modeling accuracy is always needed. Typically, an acceptable error bound ϵ_M is specified and L can be chosen just large enough to meet this accuracy threshold specification, while still meeting the time requirement to carry out the online update.

For practical applications of RBF networks, $\{c_j\}$ and $\{\sigma_j\}$ are mostly fixed as constants [66]. A practical rule of thumb is to spread the centers evenly over the time period $[0, T_p]$, and the spread to be selected as $\sigma_j = \frac{d}{\sqrt{2}}$, $\forall j$, where d is the distance between intermediate centers [67].

Once these structural parameters are fixed, the RBF network is ready to be trained. The weights $\{\omega_j\}$ of the RBF can be trained online using the backpropagation learning rule [67]. They are adjusted in the direction of the steepest descent with respect to error, to minimize the squared error of the network. The cost function to quantify the fitness of the RBF network is chosen as

$$E = \frac{1}{2}[f_e(t) - \sum_{j=0}^{L} \omega_j \phi_j(|t_r - c_j|)]^2. \tag{3.4}$$

The energy gradient is given by

$$\begin{aligned}\Delta\omega_j(t) &= -\gamma\frac{\partial E}{\partial \omega_j} \\ &= \gamma[f_e(t) - \sum_{j=0}^{L}\omega_j\phi_j(|t_r - c_j|)]\phi_j(|t_r - c_j|). \qquad (3.5)\end{aligned}$$

Since the neural network is trained on-line, it is direct to use the actual tracking error as the training signal so that it is possible to perform the learning process with the system under closed-loop control. In this case, the actual tracking error is expected to approach zero [68]. Therefore, replacing $[f_e(t) - \sum_{j=0}^{L}\omega_j\phi_j(|t_r - c_j|)]$ with the actual tracking error $e(t)$, the weights adaptation rule in the form of (3.5), is thus obtained as:

$$\omega_j(t+1) = \omega_j(t) + \gamma e(t) exp(-|t_r - c_j|^2/2\sigma_j^2), \qquad (3.6)$$

where γ is a learning gain satisfying $0 < \gamma < 1$. A small learning gain results in a more stable weight convergence at the expense of a slower learning rate. On the other hand, a large learning gain speeds up the learning rate with a relative loss in stability. Therefore, the learning gain γ is selected as a tradeoff between stability and learning rate.

In the precision control application of concern in this chapter, it may be noted that drastic changes of error can occur, commonly encountered during a directional change in motion when the frictional effects can vary in a discontinuous manner. The RBF network does not possess a structure to cope with such discontinuities. For further improvement on the tracking performance, an ILC component can be further augmented to the configuration of Figure 3.2. With an ILC augmented to complement the constraints of the RBF network, the proposed control scheme can also be more tolerant of a less

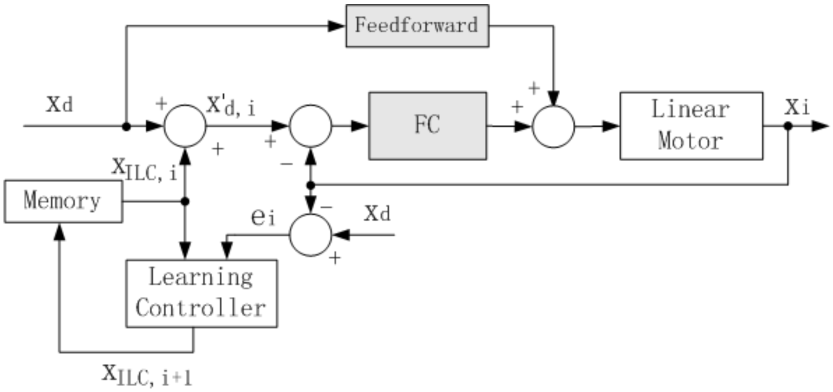


Figure 3.3: Standard control with ILC (ILC-only scheme)

appropriately chosen structure for the RBF network. The basic and RBF-augmented ILC schemes are elaborated in Section 3.2.2 and Section 3.2.3 respectively.

3.2.2 Iterative Learning Control

Iterative learning control (ILC) was proposed by Arimoto et al. to achieve a better system performance of repetitive systems over a finite time interval [8]. The main idea associated with the use of the ILC is to enhance the system performance by using the information from the previous cycle in the next cycle over a period of time until the performance achieved is satisfactory [56] [62] [69] [70]. ILC is a memory-based control scheme that needs to store the tracking errors and control efforts of previous iterations so that the control efforts of the present cycle can be constructed. In this chapter, the P-type update law is adopted for the ILC [69] [70]. An independent ILC scheme, without using the RBF error model, is given in Figure 3.3 to modify the command to the feedback-feedforward controller [69].

Under this configuration, during the *ith* ILC iteration, the modified desired trajectory

$x'_{d,i}$ is given by

$$x'_{d,i}(t) = x_d(t) + x_{ILC,i}(t), \tag{3.7}$$

where t is the discrete time index. The update law for the ILC is

$$x_{ILC,i+1}(t) = x_{ILC,i}(t) + Ke_i(t+1), \tag{3.8}$$

where K is the learning gain. For the P-type ILC, a sufficient condition for learning convergence can be found in [71] which also provides guidelines for the choice of K. A more thorough analysis of convergence and stability for sampled-data ILCs can be found in [72]. Under this basic ILC configuration, the tracking error and the output of the ILC during the previous cycle are used to update the output of ILC during the present cycle.

3.2.3 Combined RBF-ILC System

The proposed control scheme of Figure 3.1 combines the advantages of the RBF network and the ILC control components. The RBF network is used to facilitate fast initial compensation of the tracking errors while the ILC can iteratively induce further improvement when the self-tuning RBF network ceases to yield further tracking error reduction, a situation which can occur due to scenarios such as an under-sized structure, or inherent limitations of the RBF in modeling the discontinuous part of the tracking errors.

Thus, compared to the basic ILC scheme, the proposed control scheme has a higher convergence rate owing to the RBF network in place. Compared to a RBF-only scheme, the proposed scheme is able to yield an additional margin of improvement in tracking performance, especially over portions of the trajectory where the RBF network does not model the error well. These portions include the directional changeover points and load change instances. The combined structure also means that the requirements on either component can be less stringent to achieve the same performance, e.g., a smaller RBF network or a smaller learning gain can be used.

Under this proposed configuration, during the *ith* ILC iteration, the modified trajectory $x'_{d,i}$ is given by

$$x'_{d,i}(t) = x_d(t) + x_{RBF,i}(t) + x_{ILC,i}(t). \tag{3.9}$$

$x_{ILC,i}$ is the output of the ILC during the *ith* iteration. The structure and the function of the RBF network remains the same as the one presented in Section 3.2.1. The update law for the ILC also remains the same as (3.8).

3.3 Convergence Analysis of Proposed Control Scheme

In this section, the discussion of the convergence analysis is based on the model (2.1)-(2.3) shown in Chapter 2. Additionally, the frictional force is also considered in the model. In consideration of the frictional force, the model can be similarly simplified by ignoring the delay of electrical response

$$\ddot{x} = (-\frac{K_f K_e}{R}\dot{x} + \frac{K_f}{R}u(t) - f_{ripple}(x) - f_{fric}(\dot{x}) - f_{nl}(t))/M. \tag{3.10}$$

Let $K_1 = (K_f K_e/R)$ and $K_2 = (K_f/R)$. Thus, it can be written as

$$\ddot{x} = -\frac{K_1}{M}\dot{x} + \frac{K_2}{M}u(t) - \frac{1}{M}(f_{ripple}(x) + f_{fric}(\dot{x}) + f_{nl}(t)). \tag{3.11}$$

As proposed in the above section, the control u is given by

$$\begin{aligned} u(t) &= k_p'[x_d(t) + x_{ILC,i}(t) + x_{RBF,i}(t) - x(t)] \\ &+ k_d'[\dot{x}_d(t) + \dot{x}_{ILC,i}(t) + \dot{x}_{RBF,i}(t) - \dot{x}(t)], \end{aligned} \tag{3.12}$$

where k_p' and k_d' are the PD control parameters, respectively. The closed-loop system, consisting of the feedback controller and the linear motor, is considered as the compensated system.

Substituting u into (3.11) yields

$$\begin{aligned} \ddot{x} &= -\frac{K_2k_p'}{M}x - (\frac{K_1}{M} + \frac{K_2k_d'}{M})\dot{x} + \\ & \frac{K_2k_p'}{M}(x_{ILC,i} + k_d\dot{x}_{ILC,i}) \\ &- \frac{1}{M}[f_{fric}(\dot{x}) + f_{ripple}(x) + f_{nl}] \\ &+ \frac{K_2}{M}[k_p'(x_d + x_{RBF,i}) + k_d'(\dot{x}_d + \dot{x}_{RBF,i})], \end{aligned} \tag{3.13}$$

where $k_d = k_d'/k_p'$.

The state space form of the compensated system may be described as:

$$\begin{aligned} \dot{\chi}_i(t) &= A\chi_i(t) + Br_i(t) + D_i(t) \\ x_i(t) &= C_1\chi_i(t) \\ \dot{x}_i(t) &= C_2\chi_i(t), \end{aligned} \tag{3.14}$$

where

$$
\begin{aligned}
\chi_i &= [x_i\ \dot{x}_i]^T, \quad A = \left[\begin{array}{cc} 0 & 1 \\ -\frac{K_2 k_p'}{M} & -(\frac{K_1}{M} + \frac{K_2 k_d'}{M}) \end{array}\right] \\
B &= \left[\begin{array}{c} 0 \\ \frac{K_2 k_p'}{M} \end{array}\right], C_1 = [1\ 0],\ C_2 = [0\ 1] \\
D_i(t) &= [0\ d]^T \\
d &= -\frac{1}{M}[f_{fric}(\dot{x}) + f_{ripple}(x) + f_{nl}] \\
&+ \frac{K_2}{M}[k_p'(x_d + x_{RBF,i}) + k_d'(\dot{x}_d + \dot{x}_{RBF,i})]. \qquad (3.15)
\end{aligned}
$$

In (3.14), $r_i(t) = x_{ILC,i} + k_d \dot{x}_{ILC,i}$, $D_i(t)$ represents the nonlinear part of the compensated system, and i denotes the ith repetitive operation of the system. The trajectory is considered as a periodic one, i.e., $t \in [t_0, t_0 + T_p]$.

Lemma 3.1. (Theorem 5.D4 of [73]). The time-invariant dynamic linear system $z(k+1) = \Phi z(k)$ is asymptotically stable if and only if all eigenvalues of Φ have magnitudes less than 1, i.e., $|\lambda_j(\Phi)| < 1$.

Theorem 3.1. For the system (3.14) with the assumptions that $\|\chi_i(t_0) - \chi_{i-1}(t_0)\| \leq b_{\chi 0}$, $\|D_i(t) - D_{i-1}(t)\| \leq b_D$, and

$$
\| \left[\begin{array}{cc} I - C_1\theta & -C_1\theta k_d \\ -C_2\theta & I - C_2\theta k_d \end{array}\right] \| < 1,
$$

where $\theta = \int_0^T e^{A(T-\tau)} d\tau BK$, given the desired trajectory $x_d(t)$ over the fixed time interval $[t_0, t_0 + T_p]$, by using the learning control law (3.16), then, the tracking error is bounded

$$
\lim_{i\to\infty} \|\mathbf{e}_i\| \ \leq \ \sqrt{\frac{\sigma(b_{\chi 0}, b_D)}{(1-\rho)\lambda_{min}(P)}},
$$

where ρ and P are defined in (3.50) and (3.43), respectively. Moreover, the tracking error $e(t)$ is converge uniformly to zero at the sampling instants $t = t_1, t_2, ..., t_N$, where $t_i - t_{i-1} = const$, as $i \to \infty$ when $b_{\chi 0}$, and $b_D \to 0$.

Proof. The proof includes two steps. The first step is to derive the convergence in the case of $D_i = 0$ and the control system (3.14) to be exactly re-initialized at $\chi(t_0)$. The second step is to prove the boundedness of the error under the proposed control scheme in the case of $\|D_i - D_{i-1}\| \leq b_D$ and $\|\chi_i(t_0) - \chi_{i-1}(t_0)\| \leq b_0$.

Step 1. The case of $D_i = 0$: It is also assumed that the control system (3.14) is exactly re-initialized at $\chi(t_0)$. Consider a sampling period T where $t_k - t_{k-1} = T$ and give a set of sampled data of the desired output trajectory $x_d(t_k), k \in \{0, 1, 2, ..., N\}$. The control objective is to design a sampled-data iterative learning controller such that when starting from an arbitrary initial state $\chi_i(t_0)$, the output $x_i(t_k)$ at each sampling instant approaches to $x_d(t_k)$, i.e., $x_i(t_k) \to x_d(t)$ as $i \to \infty$.

To derive the convergence, the following sampled-data learning law is used

$$x_{ILC,i}(t_k) = x_{ILC,i-1}(t_k) + Ke_{i-1}(t_{k+1}), \tag{3.16}$$

where $x_{ILC}(t_k)$ is the output of the ILC, $x_{ILC}(t_k)$ denotes the constant input for the compensated system for $t \in [t_k, t_{k+1}]$ and $k = 0, 1, ...N - 1$. Here K is a constant gain. In this sampled-data iterative learning algorithm, the error history is sampled at $t_0, t_1, ..., t_N$, and stored in the memory. The system is analyzed at each sampling instant via a discrete approach. The solution of the state space (3.14) at the sampling instant

t_k can be written as

$$\begin{aligned}\chi_i(t_{k+1}) &= e^{A(t_{k+1}-t_0)}\chi_i(t_0) + [\int_{t_0}^{t_1} e^{A(t_{k+1}-\tau)}d\tau B r_i(t_0) \\ &\quad +... + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}d\tau B r_i(t_k)]. \end{aligned} \tag{3.17}$$

The tracking error $e_i(t_{k+1})$ at the ith repetition is that $e_i(t_{k+1}) = x_d(t_{k+1}) - x_i(t_{k+1})$. From this definition, the following output error equation can be obtained

$$\begin{aligned} e_i(t_{k+1}) &= x_d(t_{k+1}) - x_{i-1}(t_{k+1}) \\ &\quad -(x_i(t_{k+1}) - x_{i-1}(t_{k+1})) \\ &= e_{i-1}(t_{k+1}) - C_1(\chi_i(t_{k+1}) - \chi_{i-1}(t_{k+1})). \end{aligned} \tag{3.18}$$

Substituting (3.17) into (3.18) yields

$$\begin{aligned} e_i(t_{k+1}) &= e_{i-1}(t_{k+1}) - C_1 e^{A(t_{k+1}-t_0)}\Delta\chi_i(t_0) \\ &\quad -C_1[\int_{t_0}^{t_1} e^{A(t_{k+1}-\tau)}d\tau B\Delta r_i(t_0) + ... \\ &\quad + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}d\tau B\Delta r_i(t_k)], \end{aligned} \tag{3.19}$$

where $\Delta\chi_i(t_0) = \chi_i(t_0) - \chi_{i-1}(t_0)$, and $\Delta r_i(t_k) = r_i(t_k) - r_{i-1}(t_k) = x_{ILC,i}(t_k) - x_{ILC,i-1}(t_k) + k_d(\dot{x}_{ILC,i}(t_k) - \dot{x}_{ILC,i-1}(t_k)) = Ke_{i-1}(t_{k+1}) + k_d\dot{e}_{i-1}(t_{k+1})$.

$\dot{e}_i(t_{k+1})$ can be written as

$$\begin{aligned} \dot{e}_i(t_{k+1}) &= \dot{x}_d(t_{k+1}) - \dot{x}_{i-1}(t_{k+1}) \\ &\quad -(\dot{x}_i(t_{k+1}) - \dot{x}_{i-1}(t_{k+1})) \\ &= \dot{e}_{i-1}(t_{k+1}) - C_2(\chi_i(t_{k+1}) - \chi_{i-1}(t_{k+1})) \\ &= \dot{e}_{i-1}(t_{k+1}) - C_2 e^{A(t_{k+1}-t_0)}\Delta\chi_i(t_0) \\ &\quad -C_2[\int_{t_0}^{t_1} e^{A(t_{k+1}-\tau)}d\tau B\Delta r_i(t_0) + ... \\ &\quad + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}d\tau B\Delta r_i(t_k)]. \end{aligned} \tag{3.20}$$

Let

$$
\begin{aligned}
E(k+1) &= I - C_1 \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau BK, \\
E(k+1)' &= -C_1 \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau BKk_d \\
\bar{E}(k+1) &= I - C_2 \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau BKk_d, \\
\bar{E}(k+1)' &= -C_2 \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau BK,
\end{aligned} \tag{3.21}
$$

where I is a unit matrix. At the time point t_{k+1}, since the previous controls $\Delta r_i(t_0)$, $\Delta r_i(t_1)$, ... $\Delta r_i(t_k)$ are available, by the assumption, substituting the control law (3.16) into (3.19) yields

$$
\begin{aligned}
& e_i(t_{k+1}) = [I - C_1 \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau BK] e_{i-1}(t_{k+1}) \\
& -C_1 [\int_{t_0}^{t_1} e^{A(t_{k+1}-\tau)} d\tau B(K e_{i-1}(t_1) + k_d K \dot{e}_{i-1}(t_1)) \\
& +... + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau B k_d K \dot{e}_{i-1}(t_{k+1})].
\end{aligned} \tag{3.22}
$$

$\dot{e}_i(t_{k+1})$ can be written as

$$
\begin{aligned}
& \dot{e}_i(t_{k+1}) = [I - C_2 \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau B k_d K] \dot{e}_{i-1}(t_{k+1}) \\
& -C_2 [\int_{t_0}^{t_1} e^{A(t_{k+1}-\tau)} d\tau B(K e_{i-1}(t_1) + k_d K \dot{e}_{i-1}(t_1)) \\
& + \int_{t_1}^{t_2} e^{A(t_{k+1}-\tau)} d\tau B(K e_{i-1}(t_2) + k_d K \dot{e}_{i-1}(t_2)) \\
& +... + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau BK e_{i-1}(t_{k+1})].
\end{aligned} \tag{3.23}
$$

For $t = t_1$ at the ith iteration, the above equation becomes

$$\begin{aligned}
& e_i(t_1) = [I - C_1 \int_{t_0}^{t_1} e^{A(t_1-\tau)} d\tau BK] e_{i-1}(t_1) \\
& -C_1 \int_{t_0}^{t_1} e^{A(t_2-\tau)} d\tau BK k_d \dot{e}_{i-1}(t_1) \\
& = E(1) e_{i-1}(t_1) + E(1)' \dot{e}_{i-1}(t_1) \\
& \dot{e}_i(t_1) = [I - C_2 \int_{t_0}^{t_1} e^{A(t_1-\tau)} d\tau BK k_d] \dot{e}_{i-1}(t_1) \\
& -C_2 \int_{t_0}^{t_1} e^{A(t_2-\tau)} d\tau BK e_{i-1}(t_1) \\
& = \bar{E}(1) \dot{e}_{i-1}(t_1) + \bar{E}(1)' e_{i-1}(t_1), \qquad (3.24)
\end{aligned}$$

where

$$E(1)' = -C_1 \int_{t_0}^{t_1} e^{A(t_2-\tau)} d\tau BK k_d,$$
$$\bar{E}(1)' = -C_2 \int_{t_0}^{t_1} e^{A(t_2-\tau)} d\tau BK.$$

For $t = t_2$ at the ith iteration, it can be obtained as

$$\begin{aligned}
& e_i(t_2) = E(2) e_{i-1}(t_2) + E(2)' \dot{e}_{i-1}(t_2) \\
& -C_1 \int_{t_0}^{t_1} e^{A(t_2-\tau)} d\tau B [K e_{i-1}(t_1) + K k_d \dot{e}_{i-1}(t_1)]. \qquad (3.25)
\end{aligned}$$

Let

$$\begin{aligned}
E_{2,1} &= -C_1 \int_{t_0}^{t_1} e^{A(t_2-\tau)} d\tau BK, \\
E'_{2,1} &= -C_1 \int_{t_0}^{t_1} e^{A(t_2-\tau)} d\tau BK k_d. \qquad (3.26)
\end{aligned}$$

(3.25) can be rewritten as

$$\begin{aligned}
e_i(t_2) &= E(2) e_{i-1}(t_2) + E(2)' \dot{e}_{i-1}(t_2) \\
& \quad + E_{2,1} e_{i-1}(t_1) + E'_{2,1} \dot{e}_{i-1}(t_1). \qquad (3.27)
\end{aligned}$$

Similarly, it is obtained as

$$\begin{aligned}\dot{e}_i(t_2) &= \bar{E}(2)\dot{e}_{i-1}(t_2) + \bar{E}(2)'e_{i-1}(t_2) \\ &\quad +\bar{E}'_{2,1}e_{i-1}(t_1) + \bar{E}_{2,1}\dot{e}_{i-1}(t_1). \end{aligned} \tag{3.28}$$

Similarly,

$$\begin{aligned} e_i(t_N) = & \quad E(N)e_{i-1}(t_N) + E(N)'\dot{e}_{i-1} + E_{N,1}e_{i-1}(t_1) \\ + & \quad E'_{N,1}\dot{e}_{i-1}(t_1) + E_{N,2}e_{i-1}(t_2) + E'_{N,2}\dot{e}_{i-1}(t_2) + ... \\ + & \quad E_{N,N-1}e_{i-1}(t_{N-1}) + E'_{N,N-1}\dot{e}_{i-1}(t_{N-1}) \end{aligned} \tag{3.29}$$

$$\begin{aligned} \dot{e}_i(t_N) &= \bar{E}(N)\dot{e}_{i-1}(t_N) + \bar{E}(N)'e_{i-1} + \bar{E}'_{N,1}e_{i-1}(t_1) \\ &+ \bar{E}_{N,1}\dot{e}_{i-1}(t_1) + \bar{E}'_{N,2}e_{i-1}(t_2) + \bar{E}_{N,2}\dot{e}_{i-1}(t_2) + ... \\ &+ \bar{E}'_{N,N-1}e_{i-1}(t_{N-1}) + \bar{E}_{N,N-1}\dot{e}_{i-1}(t_{N-1}), \end{aligned} \tag{3.30}$$

where

$$\begin{aligned} E_{N,i} &= -C_1 \int_{t_i-1}^{t_i} e^{A(t_N-\tau)}d\tau BK, \quad E'_{N,i} = E_{N,i}k_d \\ \bar{E}'_{N,i} &= -C_2 \int_{t_i-1}^{t_i} e^{A(t_N-\tau)}d\tau BK, \quad \bar{E}_{N,i} = \bar{E}'_{N,i}k_d, \\ i &= 1, ...N-1. \end{aligned} \tag{3.31}$$

The above equations can be written as a composite form

$$\mathbf{e}_i = \mathbf{E}\mathbf{e}_{i-1}, \tag{3.32}$$

where $\mathbf{e}_i = [e_i(t_1), \dot{e}_i(t_1), e_i(t_2), \dot{e}_i(t_2), ..., e_i(t_N), \dot{e}_i(t_N)]^T$ and

$$\mathbf{E} = \left[\begin{array}{ccccccc} E(1) & E(1)' & 0 & 0 & ... & 0 & 0 \\ \bar{E}(1)' & \bar{E}(2) & 0 & 0 & ... & 0 & 0 \\ E_{2,1} & E'_{2,1} & E(2) & E(2)' & ... & 0 & 0 \\ \bar{E}'_{2,1} & \bar{E}_{2,1} & \bar{E}(2)' & \bar{E}(2) & ... & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ E_{N,1} & E'_{N,1} & E_{N,2} & E'_{N,2} & ... & E(N) & E(N)' \\ \bar{E}'_{N,1} & \bar{E}_{N,1} & \bar{E}'_{N,2} & \bar{E}_{N,2} & ... & \bar{E}(N)' & \bar{E}(N) \end{array} \right].$$

Since K is a constant sequence matrix, k_d is a constant, and $C_1 \int_{t_{j+1}}^{t_j} e^{A(t_{j+1}-\tau)} d\tau B$, $C_2 \int_{t_{j+1}}^{t_j} e^{A(t_{j+1}-\tau)} d\tau B$, $k = 0, 1, ...N-1; j = 0, 1, 2, ...N-1$ and $j < k$ are the constant matrices for given N sampling patterns, each element of $\mathbf{E}$ is a constant matrix with respect to the iteration i. Thus, the system becomes a discrete time-invariant system. According to Lemma 3.1, $\mathbf{e}_i$ is convergent if $\mathbf{E}$ is stable, i.e., all $|\lambda_i[\mathbf{E}]| < 1$.

Since the matrix $\mathbf{E}$ is a lower block triangular, it can be obtained as

$$\lambda_i[\mathbf{E}] = \cup_{j=0}^{N-1} \{ \lambda_k \left[\begin{array}{cc} E(j+1) & E(j+1)' \\ \bar{E}(j+1)' & \bar{E}(j+1) \end{array} \right] \}. \tag{3.33}$$

This implies that $\mathbf{e}_i$ is convergent if $\left[\begin{array}{cc} E(j+1) & E(j+1)' \\ \bar{E}(j+1)' & \bar{E}(j+1) \end{array} \right]$ for each $j = 0, 1, 2, ...N-1$ is a stability matrix.

Note that for a constant sampling

$$\begin{aligned} E(j+1) &= I - C_1 \int_{t_j}^{t_{j+1}} e^{A(t_{j+1}-\tau)} d\tau B K \\ &= I - C_1 \int_0^T e^{A(T-\tau)} d\tau B K. \end{aligned} \tag{3.34}$$

Thus, $\mathbf{e}_i$ is convergent if $\| \left[\begin{array}{cc} I - C_1\theta & -C_1\theta k_d \\ -C_2\theta & I - C_2\theta k_d \end{array} \right] \| < 1$,where $\theta = \int_0^T e^{A(T-\tau)} d\tau B K$. This completes the proof of Step 1.

Step 2: In the above development, convergence is established without considering disturbances. In the following, it shows the uniform convergence of the system against state disturbances, and reinitialization error at each iteration. Consider the repetitive system (3.14) with uncertainty and disturbance.

The solution of (3.14) can be obtained as follows:

$$\begin{aligned} \chi_i(t_{k+1}) &= e^{A(t_{k+1}-t_0)}\chi_i(t_0) \\ &+\int_{t_0}^{t_{k+1}} e^{A(t_{k+1}-\tau)}d\tau B r_i(t_k) \\ &+\int_{t_0}^{t_{k+1}} e^{A(t_{k+1}-\tau)}D_i(\tau)d\tau. \end{aligned} \tag{3.35}$$

For the error model, it can be obtained as

$$\begin{aligned} e_i(t_{k+1}) &= e_{i-1}(t_{k+1}) - C_1 e^{A(t_{k+1}-t_0)}\Delta\chi_i(t_0) \\ &-C_1\int_{t_0}^{t_{k+1}} e^{A(t_{k+1}-\tau)}d\tau B\Delta r_i(t_k) \\ &-C_1\int_{t_0}^{t_{k+1}} e^{A(t_{k+1}-\tau)}\Delta D_i(\tau)d\tau \end{aligned} \tag{3.36}$$

$$\begin{aligned} \dot{e}_i(t_{k+1}) &= \dot{e}_{i-1}(t_{k+1}) - C_2 e^{A(t_{k+1}-t_0)}\Delta\chi_i(t_0) \\ &-C_2\int_{t_0}^{t_{k+1}} e^{A(t_{k+1}-\tau)}d\tau B\Delta r_i(t_k) \\ &-C_2\int_{t_0}^{t_{k+1}} e^{A(t_{k+1}-\tau)}\Delta D_i(\tau)d\tau. \end{aligned} \tag{3.37}$$

Taking a similar procedure as the proof in Step 1 and applying the control law (3.16) at $t_0, t_1, ...t_k$ to (3.36), it can be written as

$$\mathbf{e}_i = \mathbf{E}\mathbf{e}_{i-1} - \bar{H}_1\Delta\chi_i(0) - \bar{H}_2, \tag{3.38}$$

where $\mathbf{e}_i, \mathbf{E}$ are given in (3.32), and

$$\bar{H}_1 = \begin{bmatrix} C_1 e^{A(t_1-t_0)} \\ C_2 e^{A(t_1-t_0)} \\ \vdots \\ C_1 e^{A(t_N-t_0)} \\ C_2 e^{A(t_N-t_0)} \end{bmatrix} \tag{3.39}$$

$$\bar{H}_2 = \begin{bmatrix} C_1 \int_{t_0}^{t_1} e^{A(t_1-\tau)} \Delta D_i(\tau) d\tau \\ C_2 \int_{t_0}^{t_1} e^{A(t_1-\tau)} \Delta D_i(\tau) d\tau \\ \vdots \\ C_1 \int_{t_0}^{t_N} e^{A(t_N-\tau)} \Delta D_i(\tau) d\tau \\ C_2 \int_{t_0}^{t_N} e^{A(t_N-\tau)} \Delta D_i(\tau) d\tau. \end{bmatrix}. \tag{3.40}$$

Notice that

$$\| \quad \bar{H}_1\| = \sqrt{\lambda_{max}(\bar{H}_1^T \bar{H}_1)} \leq \sqrt{\|\bar{H}_1^T \bar{H}_1\|}$$

$$\leq \quad (\|C_1\| + \|C_2\|) e^{\|A\|NT} \sqrt{N}, \tag{3.41}$$

$$\| \quad \bar{H}_2\| = \sqrt{\lambda_{max}(\bar{H}_2^T \bar{H}_2)} \leq \sqrt{\|\bar{H}_2^T \bar{H}_2\|}$$

$$\leq \quad (\|C_1\| + \|C_2\|) T e^{\|A\|NT} \sqrt{N} b_D. \tag{3.42}$$

Since $\|I - C \int_0^T e^{A\tau} d\tau BK\| < 1$, it is shown in Theorem 3.1 that the constant matrix $\mathbf{E}$ is stable. Thus, the following Lyapunov equation holds,

$$\mathbf{E}^T P \mathbf{E} - P = -I, \tag{3.43}$$

where P is positive definite matrix, and I is the unit matrix. Consider the Lyapunov function $L_i = \mathbf{e}_i^T P \mathbf{e}_i$. Then along the solution of (3.38) it can be obtained as

$$\Delta L_{i+1} = L_{i+1} - L_i = \mathbf{e}_{i+1}^T P \mathbf{e}_{i+1} - \mathbf{e}_i^T P \mathbf{e}_i$$

$$= -||\mathbf{e}_i||^2 - 2\mathbf{e}_i^T \mathbf{E}^T P[\bar{H}_1 \Delta\chi_{i+1}(0) + \bar{H}_2]$$

$$+2\Delta\chi_{i+1}^T(0) \bar{H}_1^T P[\bar{H}_2 + \frac{1}{2}\bar{H}_1 \Delta\chi_{i+1}(0)] + \bar{H}_2^T P \bar{H}_2. \tag{3.44}$$

Using $-2\alpha^T\beta \leq \zeta\alpha^T\alpha + \frac{1}{\zeta}\beta^T\beta$ where ζ is an arbitrarily positive constant, with the help of (3.41), (3.42) and (3.44), the following inequalities hold:

$$\begin{aligned}
- &\quad 2\mathbf{e}_i^T\mathbf{E}^T P[\bar{H}_1\Delta\chi_{i+1}(0) + \bar{H}_2] \leq \zeta\mathbf{e}_i^T\mathbf{e}_i \\
+ &\quad \frac{1}{\zeta}[\bar{H}_1\Delta\chi_{i+1}(0) + \bar{H}_2]^T(P\mathbf{E}\mathbf{E}^T P)[\bar{H}_1\Delta\chi_{i+1}(0) + \bar{H}_2] \\
\leq &\quad \zeta\|\mathbf{e}_i\|^2 + \frac{1}{\zeta}\|P\mathbf{E}\|^2[(\|C_1\| + \|C_2\|)e^{\|A\|T}\sqrt{N}b_{\chi 0} \\
+ &\quad (\|C_1\| + \|C_2\|)Te^{\|A\|T}\sqrt{N}b_D]^2. \qquad (3.45)
\end{aligned}$$

Using the definitions of the norm, the following inequalities hold

$$\begin{aligned}
2 &\quad \Delta\chi_{i+1}^T(0)\bar{H}_1^T P[\bar{H}_2 + \frac{1}{2}\bar{H}_1\Delta\chi_{i+1}(0)] \leq 2\|\bar{H}_1\|\|P\|b_{\chi 0}(\|\bar{H}_2\| + \frac{1}{2}\|\bar{H}_1\|b_{\chi 0}) \\
&\leq \quad 2N(\|C_1\| + \|C_2\|)\|P\|e^{\|A\|NT}b_{\chi 0} \\
&\quad .[(\|C_1\| + \|C_2\|)Te^{\|A\|NT}b_D + \frac{1}{2}(\|C_1\| + \|C_2\|)e^{\|A\|T}b_{\chi 0}] \\
\bar{H}_2^T P\bar{H}_2 &\leq \quad \|\bar{H}_2^T P\|(\|\bar{H}_2\|) \\
&\leq \quad N(\|C_1\| + \|C_2\|)Te^{\|A\|NT}b_D\|P\|[(\|C_1\| + \|C_2\|)Te^{\|A\|T}b_D]. \qquad (3.46)
\end{aligned}$$

Incorporating the above inequalities produces

$$\Delta L_{i+1} \leq -(1-\zeta)\|\mathbf{e}_i\|^2 + \sigma(b_{\chi 0}, b_D), \qquad (3.47)$$

where

$$\begin{aligned}\sigma(b_{x0}, b_D) &= \frac{1}{\zeta}\|P\mathbf{E}\|^2[(\|C_1\| + \|C_2\|)e^{\|A\|NT}\sqrt{N}b_{\chi 0} \\ &+ (\|C_1\| + \|C_2\|)Te^{\|A\|NT}\sqrt{N}b_D]^2 \\ &+ 2N(\|C_1\| + \|C_2\|)\|P\|e^{\|A\|NT}b_{\chi 0} \\ &\quad .[(\|C_1\| + \|C_2\|)Te^{\|A\|NT}b_D \\ &\quad +\frac{1}{2}(\|C_1\| + \|C_2\|)e^{\|A\|NT}b_{\chi 0}] \\ &+ N(\|C_1\| + \|C_2\|)Te^{\|A\|NT}b_D\|P\| \\ &\quad .[(\|C_1\| + \|C_2\|)Te^{\|A\|T}b_D], \qquad (3.48)\end{aligned}$$

and $1 > \zeta > 0$ constant.

Since $\lambda_{min}(P)||\mathbf{e}_i||^2 \leq L_i \leq \lambda_{max}(P)||\mathbf{e}_i||^2$, it can be obtained as

$$L_{i+1} - L_i \leq -\frac{1-\zeta}{\lambda_{max}(P)}L_i + \sigma(b_{\chi 0}, b_D). \qquad (3.49)$$

Rearranging terms, it can be written as

$$\begin{aligned}L_{i+1} &\leq [1 - \frac{1-\zeta}{\lambda_{max}(P)}]L_i + \sigma(b_{x0}, b_D) \\ &= \rho L_i + \sigma(b_{\chi 0}, b_D), \qquad (3.50)\end{aligned}$$

where $\rho = 1 - \frac{1-\zeta}{\lambda_{max}(P)}$. The choice for ζ to make $|\rho| < 1$ is obvious, i.e.,

$$1 - \lambda_{max}(P) < \zeta < 1 \;\; if \;\lambda_{max}(P) < 1 \qquad (3.51)$$

$$0 < \zeta < 1 \;\; if \;\lambda_{max}(P) \geq 1. \qquad (3.52)$$

Finally, one can easily find that

$$L_i \leq \rho^i L_0 + \frac{1-\rho^i}{1-\rho}\sigma(b_{\chi 0}, b_D), \lim_{i\to\infty} L_i \leq \frac{\sigma(b_{\chi 0}, b_D)}{1-\rho} \tag{3.53}$$

$$\lim_{i\to\infty} ||\mathbf{e}_i|| \leq \sqrt{\frac{\sigma(b_{\chi 0}, b_D)}{(1-\rho)\lambda_{min}(P)}}. \tag{3.54}$$

This implies that the output error is bounded for $t \in [0, N]$, and even the uncertainties that exist converge to a residual set $\sqrt{\frac{\sigma(b_{\chi 0}, b_D)}{(1-\rho)\lambda_{min}(P)}}$ whose size depends on the bounds of $b_{\chi 0}$, b_D. Furthermore, $\lim_{i\to\infty} ||\mathbf{e}_i|| = 0$ if $b_{\chi 0}$, $b_D \to 0$.

Remark 3.1. As $t \to \infty$, the RBF is trained well. This implies that $\Delta x_{RBF,i}$ approaches to a small number. This also implies that σ decreases. From (3.54), it is observed that this may help to reduce the error. On the other hand, the ILC may help to achieve a bounded error as shown in (3.54), as $i \to \infty$. The bounded error depends on the disturbance bound and RBF training.

3.4 Simulation Study

The proposed control scheme is simulated on a PMLM, decribed in (2.1)-(2.3). The parameters of the PMLM can be found in Table 2.1 in Chapter 2.

In the simulation study, the frictional force is considered as a combination of Coulomb and viscous friction. The friction model may be written as

$$f_{fric}(\dot{x}) = (f_c + f_v|\dot{x}|)sign(\dot{x}), \tag{3.55}$$

where f_c is the minimum level of Coulomb friction and f_v is the viscous friction parameter. Note that the friction force acting is discontinuous when $\dot{x}$ changes sign which

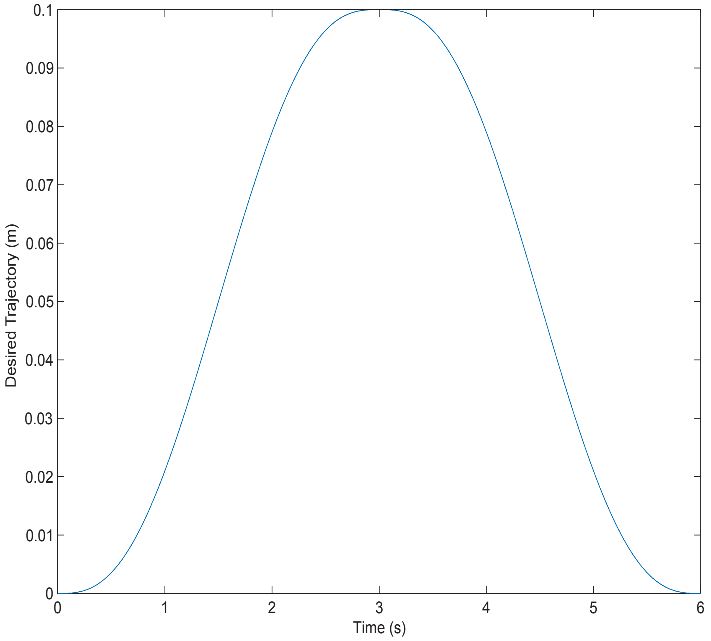


Figure 3.4: Desired trajectory, x_d

corresponds to when the motion changes direction. In the simulation study, these parameters are chosen as:$f_c = 10N$, $f_v = 10N$. They are simply estimated by a spring balance in the experimental study.

The force ripple, dominant in PMLMs, is also considered in the simulation, and it is described as a periodical sinusoidal type signal, with an amplitude of $8.5N$ ([28]) and a spatial period of $71.2mm$, i.e., $f_{ripple} = 8.5 \sin \frac{2\pi}{71200} x$, where x is expressed in μm.

The desired trajectory is chosen as (2.29). The desired trajectory with the period $T_p = 6s$ is shown in Figure 3.4. The PD controller parameters are tuned, using the Ziegler-Nichols frequency response method, to be $k'_p = 0.15$ and $k'_d = 0.00024$. The design of the feedforward controller is based on the second-order linear model. Figure 3.5 shows the tracking error incurred over one cycle when only the standard controller

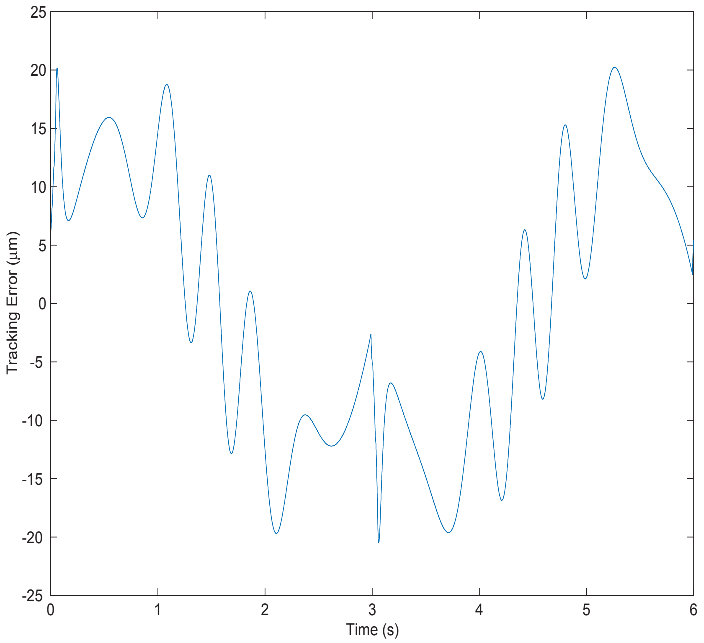


Figure 3.5: Tracking error with the standard controller

is used. This is the remnant tracking error which the proposed control method is aim to model and subsequently reduce by modifying the reference signal.

In terms of the RMS tracking error and the absolute maximum tracking error, the standard controller yields $e_{RMS} = 11.63\mu m$ with a maximum error e_{MAX} of $20.49\mu m$. These errors are thus representative of performance achievable by standard motion controllers, and can be used as benchmarks to compare the performance of various control schemes.

3.4.1 Tracking Performance- RBF-only Scheme

In this part, the RBF network is used as a parametric model for the tracking error shown in Figure 3.5. Following the parameter selection guidelines described in Section 3.2.1, 101 centers (i.e., $L = 101$) are chosen. Thus, the time period $[0, 6]$ is split into 100 equal

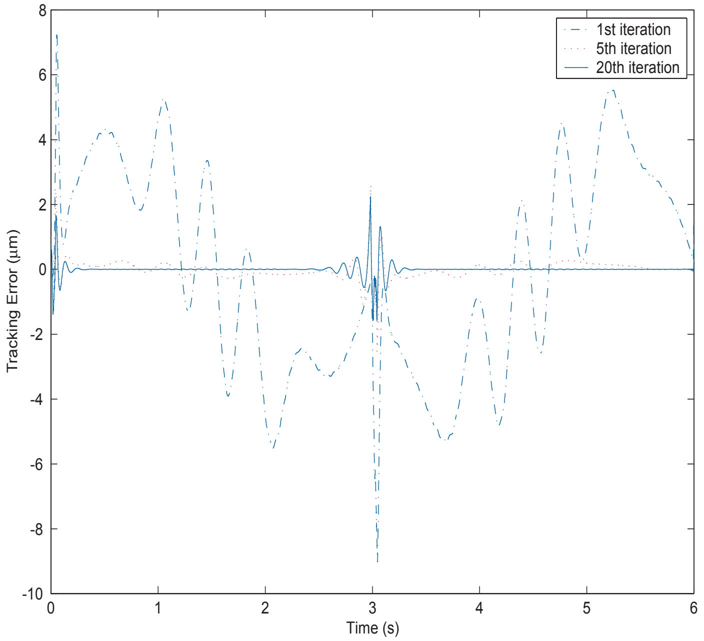


Figure 3.6: Tracking error with the RBF-only scheme

intervals. The center width is chosen as $\sigma = 0.042$, and the learning gain is fixed at $\gamma = 0.015$. Figure 3.6 shows the tracking error incurred during the $1st$, $5th$ and $20th$ cycles when the RBF is additionally used to modify the reference signal.

The RBF network shows a strong capability to approximate the remnant tracking error over the entire period. With e_{RMS} reduced to $3.2\mu m$ at a maximum error of $9.02\mu m$ after only one cycle, it is clear that the tracking error can be reduced significantly with the RBF enhancement.

Figure 3.7 shows the RBF network output during the $20th$ cycle, compared to the actual remnant tracking error with the standard controller. The RBF network is able to approximate the remnant tracking error very well, except at the directional changeover point at $t = 3s$ when the discontinuity in frictional force is experienced. The disconti-

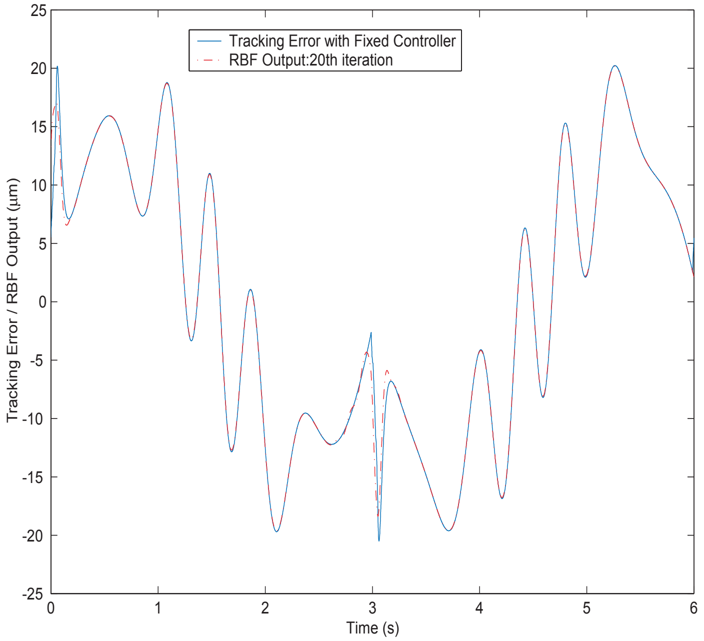


Figure 3.7: Approximation of tracking error by the RBF network

nuity appears at zero velocity at $t = 3s$. It is because the friction force as a function of only velocity is not specified at zero velocity.

Figure 3.8 shows the tracking performance of this scheme over subsequent cycles. It should be noted that with the RBF network alone, while the tracking error has been reduced significantly, it can be observed that incremental improvement tapers and slows considerably after the $5th$ cycle. The limitation of RBF network in approximating the discontinuous part of the time function at $t = 3s$ is also rather evident, as this portion of the error remains even after a large number of cycles have lapsed as seen in Figure 3.7. This limitation can be overcome with an additional ILC component.

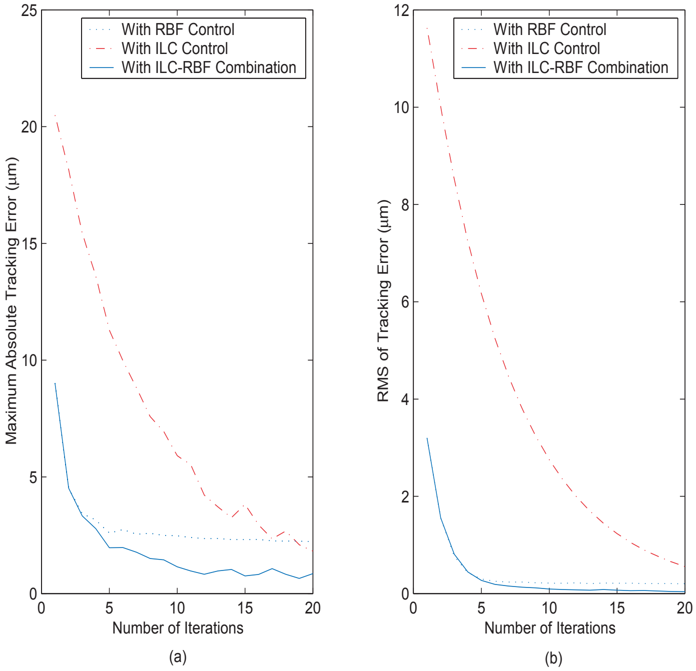


Figure 3.8: Iterative convergence performance with L=101 in terms of e_{MAX} and e_{RMS}

3.4.2 Tracking Performance- ILC-only Scheme

To illustrate the advantages of using a RBF-ILC combination, the tracking performance with only ILC is first simulated to serve as a basis for comparison. Following reported guidelines [71] on the choice of the learning gain, $K = 0.15$ is selected. Figure 3.9 depicts the tracking error incurred with only ILC component during the $1st$, $5th$, and $20th$ cycle. Compared to Figure 3.6, the tracking error with ILC converges relatively more slowly in the absence of the RBF network.

Although it is clear in Figure 3.8 that the ILC-only scheme can yield iterative improvement, the high initial convergence rate offered by the RBF network component cannot be achieved. However, it is able to reduce the maximum tracking error iteratively, and the improvement yield is especially evident at the directional changeover points. This

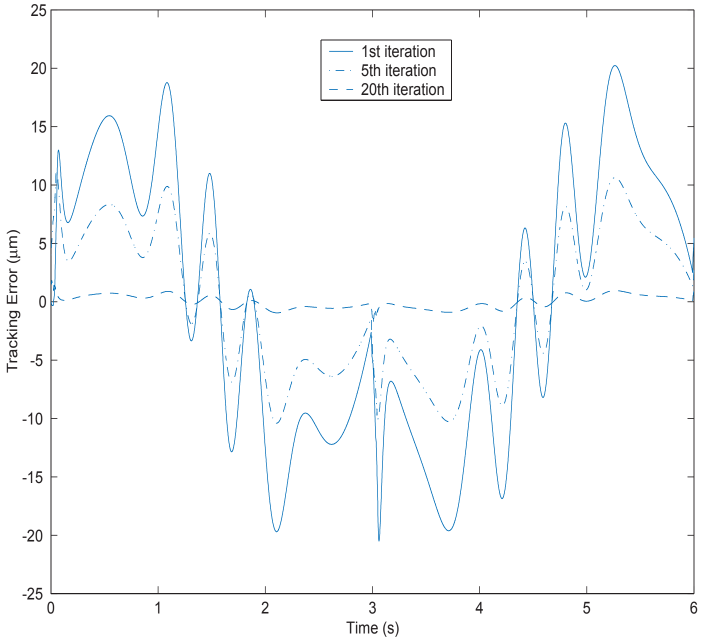


Figure 3.9: Tracking error with only ILC

fares favorably compared to the RBF-only scheme where the incremental improvement after the initial few cycles is limited. The merits of both schemes would be combined in a RBF-ILC hybrid scheme to be presented in the next subsection.

3.4.3 Tracking Performance- RBF-ILC Combined Scheme

The combined scheme is thus expected to exhibit a fast error convergence rate (consequent of the RBF network component) and also to incur good tracking performance over time (consequent of the ILC component). Figure 3.10 shows the tracking error incurred with the combined RBF-ILC scheme. The number of weights used remains fixed at $L = 101$ and the ILC learning gain remains at $K = 0.15$ for a fair comparison. Figure 3.8 compares the learning convergence with RBF-only and the ILC-only scheme. The fast initial convergence rate of the RBF-only scheme is retained, yet the hybrid scheme is

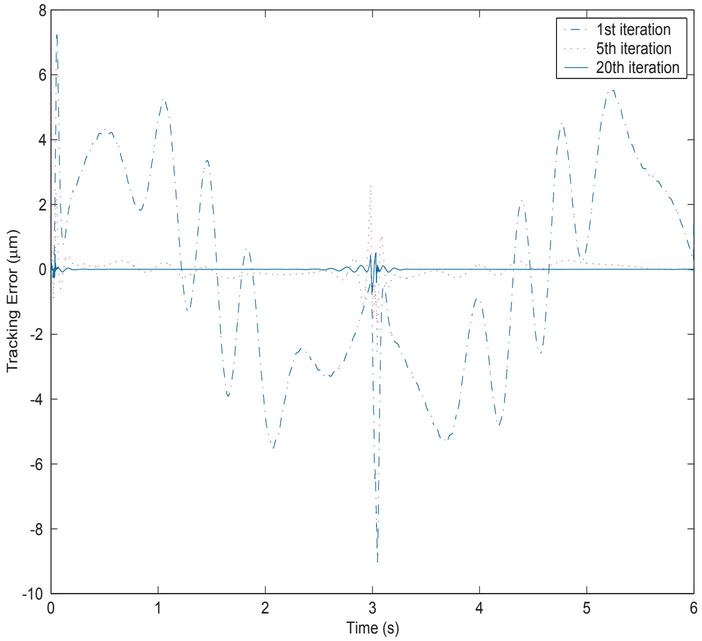


Figure 3.10: Tracking error with the RBF-ILC combined scheme

able to yield further tracking improvement even when the performance of the RBF-only scheme tapers off after the initial few cycles.

It is interesting to also verify that such an RBF-ILC combination is able to relax the requirements on the individual RBF and ILC components. The simulation is re-run with $L = 51$ which is only 50% of the RBF network size earlier considered. After 48 cycles, the proposed scheme is able to meet or better the performance obtained earlier at the $20th$ cycle with $L = 101$, as shown in Figure 3.11. The combined scheme is thus more tolerant of less appropriate parameter selection on the individual components, being able to compensate with more learning cycles.

In order to provide the clear information of every component, Figure 3.12 presents the outputs of X_{RBF}, X_{ILC}, the feedback controller and the feedforward controller in the

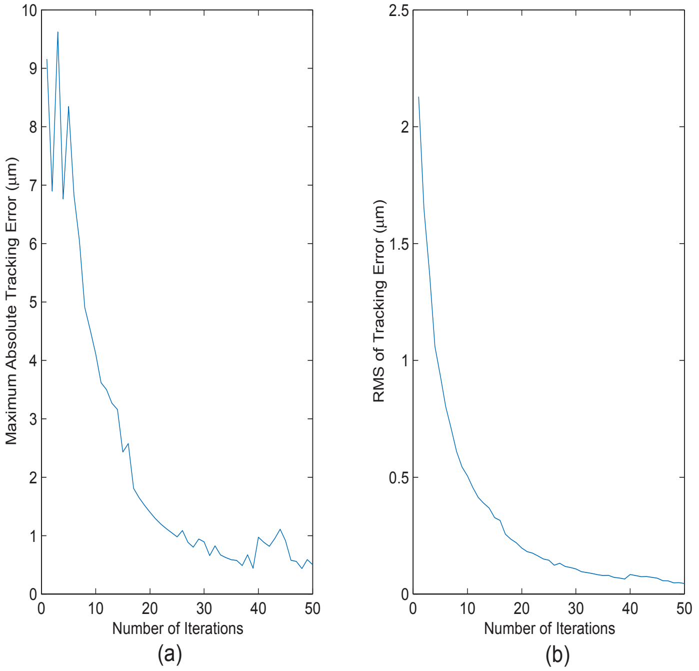


Figure 3.11: Iterative convergence performance with L=51 in terms of e_{MAX} and e_{RMS} last cycle.

3.5 Experimental Results

In this section, actual experimental results are provided to illustrate the effectiveness of the proposed method. Experiments were conducted based on an actual PMLM (LD 3810) described in Section 2.6.

The desired trajectory remains the same as in Figure 3.4. Figure 3.13 shows the tracking error in following the desired trajectory over one cycle with the performance indicator as $e_{RMS} = 12.49\mu m$ at a maximum error of $23.8\mu m$.

In what follows, the performance achieved under the various schemes is presented. It should be pointed out that similar trends to the simulation results are observed.

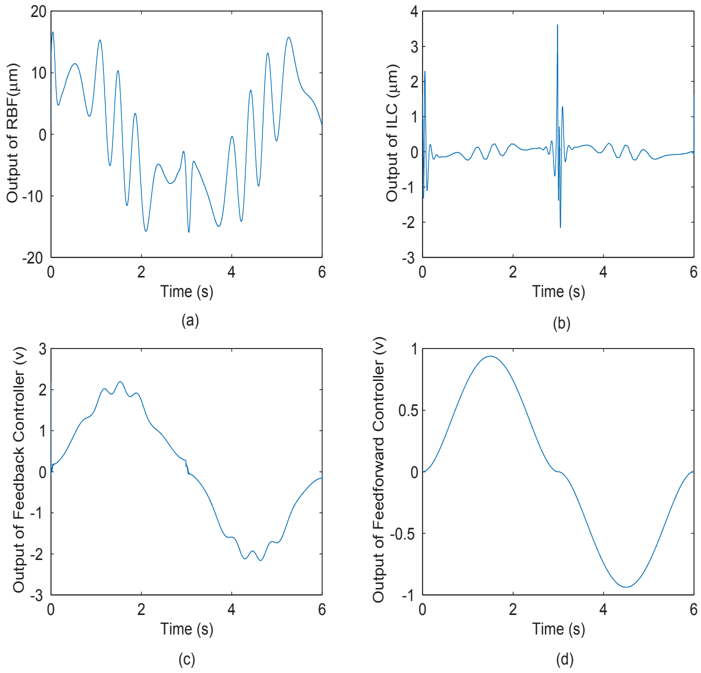


Figure 3.12: Outputs of components in the 20th cycle (a). X_{RBF}; (b). X_{ILC}; (c). feedback controller; (d). feedforward controller

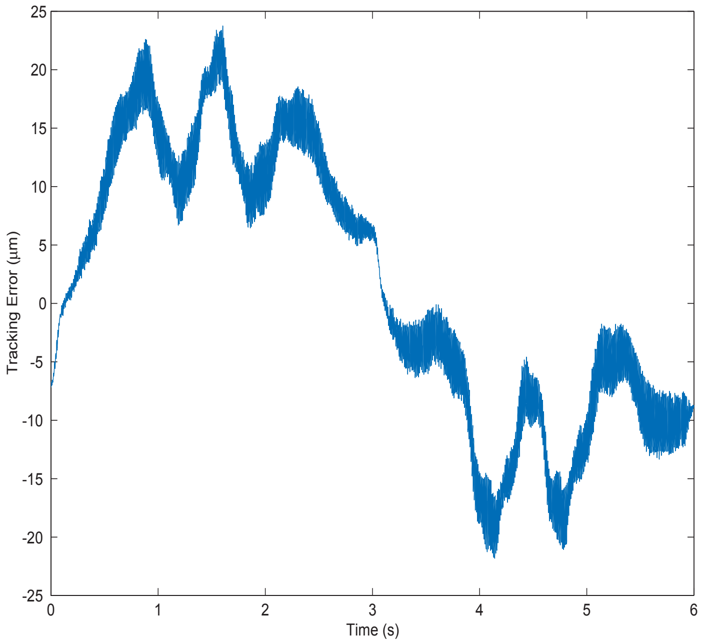


Figure 3.13: Tracking error with standard controller

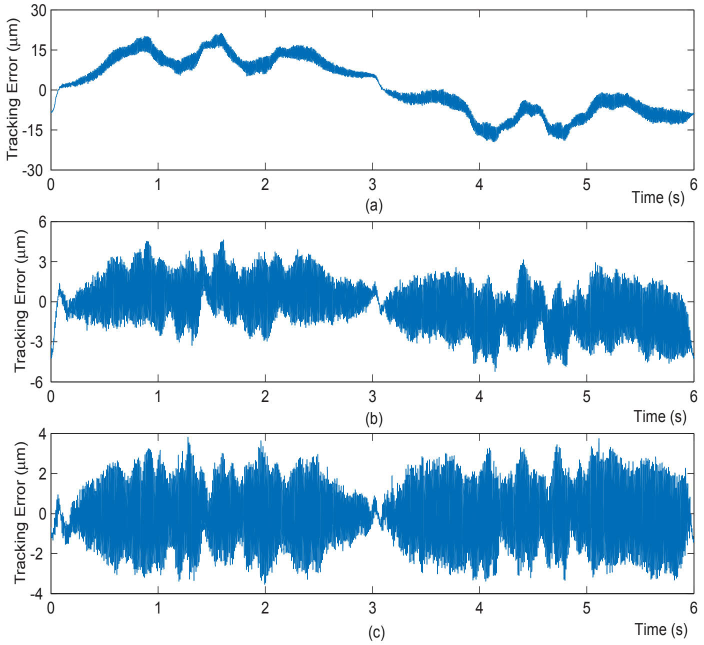


Figure 3.14: Tracking error with RBF enhancement (a). during the 1st iteration (b). during the 10th iteration (c). during the 40th iteration

3.5.1 Experimental Results- RBF-only Scheme

The RBF network is now augmented to the standard controller. 101 weights are selected to enable an efficient online update of the weights within the allowable sampling period. The initial values of the weights are set to zero, the center width is chosen as $\sigma = 0.042$, and the learning rate set to $\gamma = 0.0018$, all in accordance to the guidelines presented in Section 3.2.1. Figure 3.14 shows the tracking error incurred during the $1st$, $10th$ and $40th$ cycles. The performance indicator has improved with $e_{RMS} = 10.3\mu m$ at a maximum error of $21.2\mu m$ after the first cycle. Figure 3.15 shows the good approximation of the tracking error with the RBF network during the 40th cycle.

Figure 3.16 shows the convergence performance of the proposed schemes over 40 cycles.

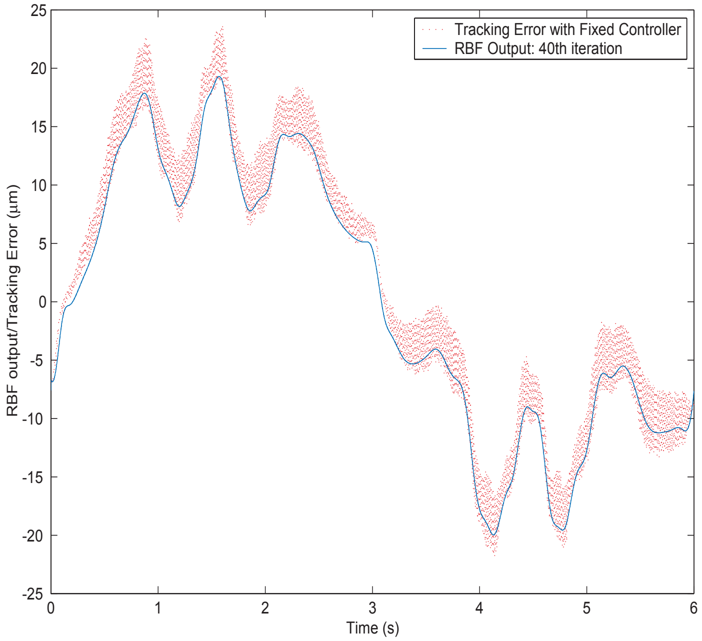


Figure 3.15: Approximation of tracking error by the RBF network

While the improvement with the RBF network addition is self-evident, it can also be noted that the incremental performance begins to taper off after the 10th cycle. After the 10th cycle, $e_{MAX} = 5.19\mu m$ and $e_{RMS} = 1.84\mu m$. During the 40th cycle, $e_{MAX} = 3.9\mu m$ and $e_{RMS} = 1.63\mu m$, indicating slight improvement attained.

3.5.2 Experimental Results- ILC-only Scheme

For the ILC only scheme, the learning rate is chosen as $K = 0.039$, and zero initial conditions are used. Figure 3.17 shows the tracking error during $1st$, $10th$ and $40th$ cycles. The tracking error can be reduced with each incremental cycle. However, the amount of initial error reduction is significantly less than the corresponding reduction when a RBF network is used.

Figure 3.16 shows the performance of this scheme over 40 cycles. While the absolute

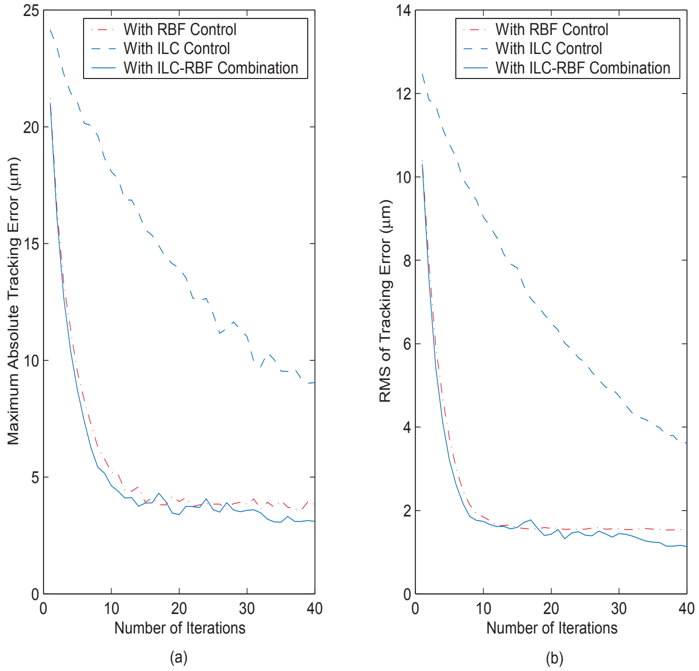


Figure 3.16: Iterative convergence performance with L=101 in terms of e_{MAX} and e_{RMS}

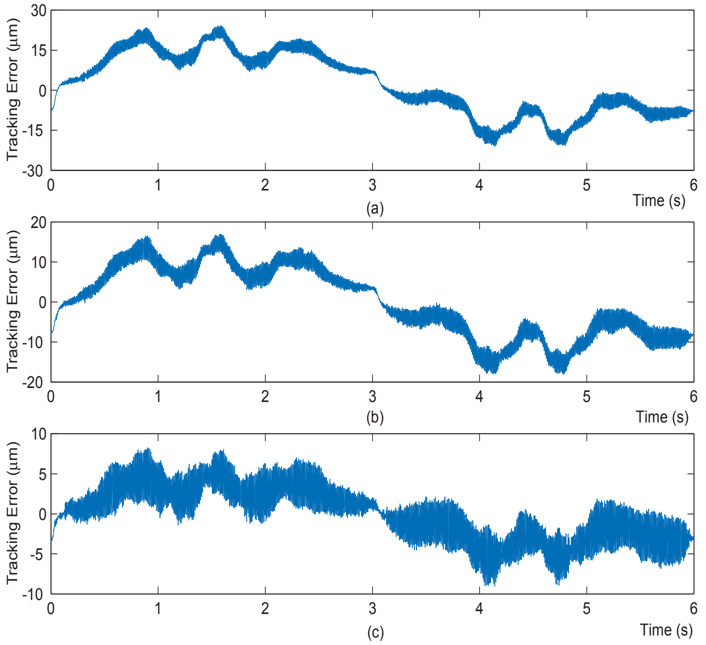


Figure 3.17: Tracking error with only ILC (a). during the 1st iteration (b). during the 10th iteration (c). during the 40th iteration

error reduced is less than the RBF-only scheme, the ILC-only scheme is able to persist to attain a more consistent incremental error reduction even after the 10th cycle.

3.5.3 Experimental Results- RBF-ILC Combined Scheme

From the above experiment results, it may be noted that the RBF and the ILC components offer rather distinct advantages. Similar parameters are retained, with $L = 101$, $K = 0.039$, and zero initial conditions. Figure 3.18 shows the tracking error incurred during the $1st$, $10th$ and $40th$ cycles. The convergence performance of the scheme is also shown in Figure 3.16. Under this RBF-ILC combined scheme, the performance indicators improve from $e_{RMS} = 1.74\mu m$ after the 10th cycle to $e_{RMS} = 1.1322\mu m$ after the 40th cycle. The maximum error also reduces from $4.64\mu m$ to $3.05\mu m$. Compared to the performance of RBF only, a slight improvement is obtained. It is partially due to the restrictions in the measuremetn resolution. The tracking error cannot reduce further. In addition, the experimental setup has smaller friction than expected. So In the RBF+ILC scheme, ILC takes relative slight effect on the performance. However, in terms of both absolute error reduction and error reduction rate, this scheme exhibits the best performance among the schemes experimented.

3.6 Conclusions

A learning control scheme is developed which is suitable for high precision and repetitive motion control applications. It uses a Radial Basis Function (RBF) network as a model for the tracking error, and an Iterative Learning Control (ILC) component to further

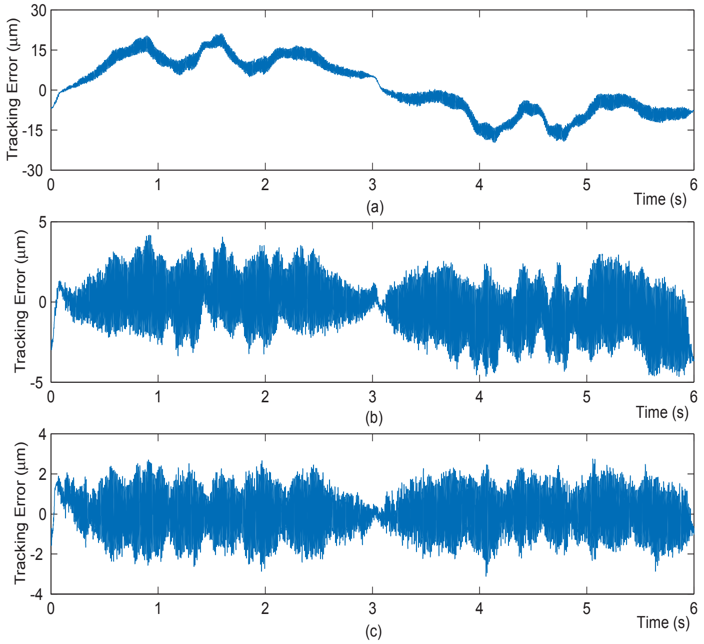


Figure 3.18: Tracking error with The RBF-ILC combination (a). during the 1st iteration (b). during the 10th iteration (c). during the 40th iteration

improve on the error model. Unlike the usual ILC scheme which adapts a feedforward control signal to achieve improved tracking performance over time, the proposed scheme iteratively adjusts the reference signal. The RBF network speeds up the initial error convergence while the ILC component can yield further improvement with time. Simulation examples and experiment results presented have verified the desirable features of the proposed method.

Chapter 4

Online Automatic Tuning of PID Controller Based on an Iterative Learning Control Approach

4.1 Introduction

Proportional-Integral-Derivative (PID) controllers are now widely used in various industrial applications where the tracking and regulation of time-continuous variables is necessary. The strong affinity with industrial applications is due largely to its simplicity and the satisfactory level of control robustness which it offers. Apart from possible minor structural differences, the distinct factor governing how well the controller performs is the tuning method adopted. To-date, many different approaches are available for tuning the PID controller (e.g., [74] [75] [76]). In more recent times, automatic tuning methods have evolved (e.g., [77] [78] [79]), where the user of the industrial controller only needs to provide simple performance specifications, initiate the tuning process with a push button, and the PID controller can be tuned satisfactorily. These tuning approaches can be generally classified under offline and online approaches. In the latter case, the controller is tuned while it is still performing the control function, with no loss in production time. From economy, practical usage and application domain viewpoints, the closed-loop online approach is an attractive approach.

To-date, however, a specific PID tuning approach is typically limited to certain classes of systems only. It also typically requires a linear model of the system, in an implicit or explicit form, based on which the controller is tuned. It is unrealistic to assume that the system of concern fits the assumed model well, since all systems encountered in practice are nonlinear in nature. As a result, the final control performance can be rather limited and unacceptable when the user requirements become stringent. Under this situation, one response may be to develop a more complex version of the PID controller. In [80], an adaptive PID controller based on the model reference technique is proposed. In [81], a direct adaptive PID control scheme has been proposed for both off-line and on-line tuning of PID parameters. In [82], a learning-enhanced nonlinear PID controller has been developed specifically for nonlinear systems. Central to all these work is a model that becomes more complicated and unwieldy, in order to yield the incremental performance needed. Correspondingly, the entire control design procedure also becomes complicated.

This chapter presents a new scheme to tune the PID control parameters based on an Iterative Learning Control (ILC) approach. The basic idea is to use ILC to derive the ideal control signal for the system to track a periodic reference sequence. This reference sequence can be the natural reference signal for the control system when it is executing repetitive operations, e.g., a servo-mechanical system executing repeated pick and place operations. It can also be a deliberate periodic sequence purely for tuning the PID controller, after which the natural reference sequence can be applied. This deliberate excitation signal can be derived, for example, by subjecting the closed-loop system to relay feedback and inducing steady state oscillation signal at an interesting frequency of the closed system. In [83], the author proved that for each causal linear time-invariant ILC, there is an equivalent feedback that achieves the ultimate ILC error with no iterations. In this chapter, the PID controller is directly constructed from the ILC algorithm, which is preferred in pratice. In this chapter, deliberate effort is put in to ensure that the tuning is done online, i.e., the tuning procedure is carried out while closed-loop operations is in progress. To this end, the ILC deviates from the usual configuration ([84]), by iteratively changing the reference signal rather than the control signal. Most industrial control systems do not allow the control signals to be changed, although the reference signal can be subject to user specifications. Thus, the method represents an approach which can be more readily incorporated into existing closed-architecture systems.

Once the ILC yields a satisfactory overall control signal, as far as a selected function of the tracking error is concerned, the PID controller is ready to be tuned. A system identification approach is adopted where the PID parameters are adjusted such that the best fit to the overall input and output signal of the ILC-augmented control system is obtained. For this purpose, it is possible that a higher order controller is adopted instead

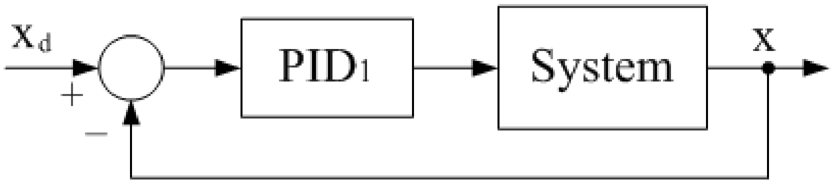


Figure 4.1: Basic PID feedback control system

of the PID controller. The proposed method is a model-free approach since no model, implicit or explicit, is assumed, and at no time, is the controller detached from the system. In this chapter, the proposed method is subject to simulation-based evaluation as well as real-time experimental tests on a piezoelectric linear motor.

4.2 Proposed Approach

In this section, the proposed tuning approach of the PID controller tuning using an ILC approach is elaborated. The entire procedure is essentially carried out over two phases. In the first phase, a modified ILC procedure is carried out to yield the ideal input and output signals of the overall ILC-augmented control system. The second phase is to use these signals to identify the best fitting PID parameters with a standard Least Square (LS) algorithm. In the following subsections, these two phases are elaborated.

4.2.1 Phase 1: Iterative Refinement of Control

Figure 4.1 shows the system under PID feedback control (PID$_1$). The controller PID$_1$ is described by:

$$u(t) = K_{p1}e(t) + K_{i1}\int_0^t e(t)dt + K_{d1}\frac{de(t)}{dt}. \tag{4.1}$$

An ILC component is now added to the basic control system to iteratively obtain enhanced control signals for tracking of the periodic reference signal. Figure 4.2 shows the configuration with the ILC augmentation. Instead of the usual approach of refining the control signal which may not be permitted in the typical closed-architecture control system, the ILC component modifies the desired reference signal through successive trials to improve the tracking performance. In this chapter, the P-type update law is adopted for ILC. Under the configuration shown in Figure 4.2, during the i-th iteration, the modified desired trajectory $x'_{d,i}$ is given by

$$x'_{d,i}(t) = x_d(t) + \Delta x_{d,i}(t), \tag{4.2}$$

where t is the discrete time index. The update law for the ILC is

$$\Delta x_{d,i+1}(t) = \Delta x_{d,i}(t) + \lambda e_i(t+1), \tag{4.3}$$

where λ is the learning gain. For the P-type ILC, a sufficient condition for learning convergence can be found in [71] which also provides guidelines for the choice of λ. A more thorough analysis of convergence and stability for sampled-data ILC can be found in [72]. Under this ILC configuration, the tracking error and the output of the ILC during the previous cycle are used to update the output of ILC during the present cycle.

Figure 4.2 can be configured in the equivalent form as shown in Figure 4.3(a), where the ILC structure for enhancement of the reference signal can be viewed instead as a parallel learning controller to PID$_1$, comprising of a ILC component and PID$_1$ in series.

When a satisfactory level of control performance has been achieved, the ideal input e

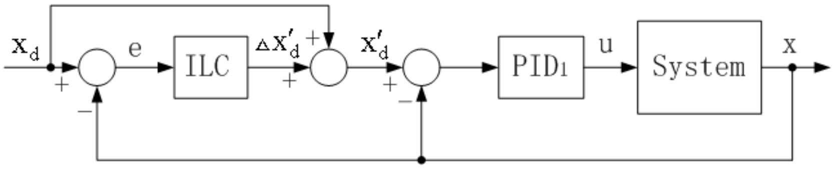


Figure 4.2: Iterative Learning Control block diagram

and output Δu for a cycle of the reference signal would have been available for the next phase.

4.2.2 Phase 2: Identifying New PID Parameters

In this phase, an equivalent PID controller PID$_2$ is derived in place of the parallel ILC+PID$_1$ component, so that Figure 4.3(b) is as close to Figure 4.3(a) as possible, as far as the response of the signal Δu to e is concerned.

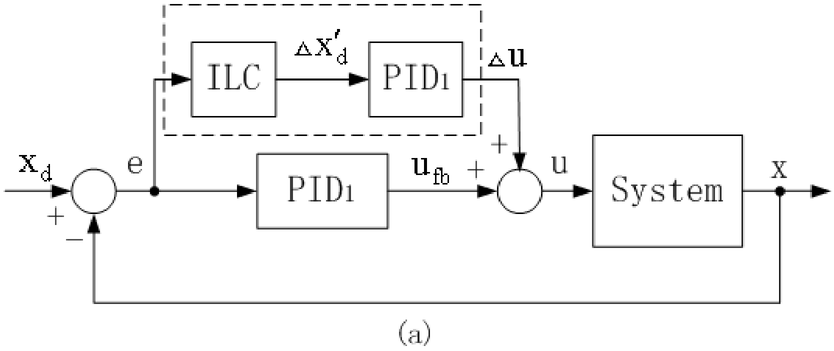
The PID$_2$ controller can be expressed as:

$$\Delta u(t) = K_{p2}e(t) + K_{i2}\int_0^t e(t)dt + K_{d2}\frac{de(t)}{dt}. \tag{4.4}$$

The standard least Square (LS) algorithm is used to obtain the parameters of PID$_2$. Equation (4.18) can be written in the linear regression form:

$$\Delta u(t) = \varphi^T(t)\theta, \tag{4.5}$$

where $\theta = [K_{p2}\ K_{i2}\ K_{d2}]^T$ and $\varphi^T(t) = [e(t)\ \int_0^t e(t)dt\ de(t)/dt]$. In practical applications, the derivative signal is seldom obtained via direct measurement, and measurement noise is amplified if it is derived via direct differentiation. In this chapter, the differential filter is used to derive the derivatives [60]. Figure 4.4 shows the block diagram of the estimator with filters $H_f(p)$, where $p = d/dt$ represents the differential operator.



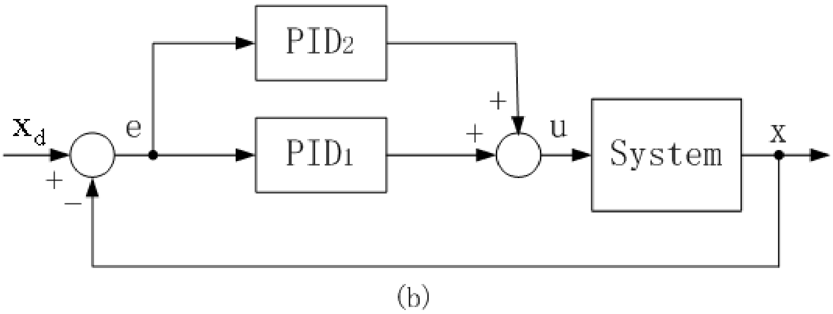


Figure 4.3: (a). Equivalent representation of the ILC-augmented control system (b). Approximately equivalent PID controller

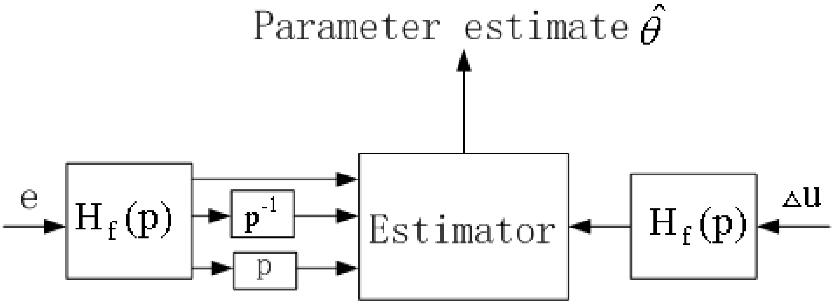


Figure 4.4: Block diagram of the estimator with filters, H_f

Equation (4.18) can be expressed in a general form:

$$\Delta u(t) = A(p)e(t), \tag{4.6}$$

where $A(p) = p + \frac{1}{p} + 1$. With the additional filters $H_f(p)$, (4.6) can be rewritten as:

$$H_f(p)\Delta u(t) = H_f(p)A(p)e(t), \tag{4.7}$$

where the filter $H_f(p)$ is a stable transfer function.

Let

$$\begin{aligned} \Delta u_f(t) = H_f(p)\Delta u(t) \\ e_f(t) = H_f(p)e(t). \end{aligned} \tag{4.8}$$

Thus, (4.7) can be written as:

$$\Delta u_f(t) = K_{p2}e_f(t) + K_{i2}\int_0^t e_f(t)dt + K_{d2}\frac{de_f(t)}{dt}. \tag{4.9}$$

Hence, the parameter vector remains as $\theta = [K_{p2}\ K_{i2}\ K_{d2}]^T$. The regression vector becomes

$$\begin{aligned} \varphi_f^T(t) &= [e_f(t)\ \ \int_0^t e_f(t)dt\ \ \frac{de_f(t)}{dt}] \\ &= [H_f(p)e(t)\ \ \frac{1}{p}H_f(p)e(t)\ \ pH_f(p)e(t)]. \end{aligned} \tag{4.10}$$

Define:

$$\begin{aligned} U &= [\Delta u_f(1)\Delta u_f(2)...\Delta u_f(N)]^T, \\ \Phi &= \begin{bmatrix} \varphi_f^T(1) \\ \varphi_f^T(2) \\ \vdots \\ \varphi_f^T(N) \end{bmatrix}, \end{aligned} \tag{4.11}$$

where N is the number of data used in the estimation. Thus, the least squares estimates of the parameters can be determined efficiently as:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T U. \tag{4.12}$$

Once the best fit PID$_2$ controller is identified. the final PID controller is the combination of PID$_1$ and PID$_2$ which can be written as:

$$\begin{aligned} u(t) &= (K_{p1} + K_{p2})e(t) + (K_{i1} + K_{i1}) \int_0^t e(t)dt + (K_{d1} + K_{d2}) \frac{de(t)}{dt} \\ &= K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}, \end{aligned} \tag{4.13}$$

where K_p, K_i and K_d are the three overall parameters of the final PID controller. In this way, the PID controller is tuned in the closed-loop.

It should be noted that the approach is applicable to control systems where the natural reference signal may not be a repetitive signal. In these cases, a deliberate periodic sequence, at an interesting frequency of the control system, can be injected purely for tuning the PID controller, and the above mentioned steps remain applicable. Thereafter, the deliberate signal can be replaced by the natural reference signal of the system.

4.3 Simulation Results

In this section, simulation study is conducted to verify the effectiveness of the proposed tuning method. The simulation example adopts the model described in Section 2.2 for precise repetitive positioning applications. The following specific system model is used in the simulation:

$$5.4\ddot{x} = -35.1\dot{x} + 8.1u - f_{fric} - f_{ripple} - f_{nl}. \tag{4.14}$$

The force ripple in (4.14) is described as a sinusoidal function with a period of $71.2mm$ and an amplitude of $3N$, i.e.,

$$f_{ripple} = 3sin(2\pi x/71200). \tag{4.15}$$

The frictional force model can refer to Section 3.4. According to the friction model in Equation (3.55), the model parameters of the friction force used in the simulation study are given as:

$$f_c = 3N, \ and \ f_v = 10N.$$

In the simulation study, a sinusoidal desired trajectory is chosen with a frequency of $0.25Hz$ as shown in Figure 4.5. The initial feedback controller PID_1 has parameters $K_{p1} = 0.045$, $K_{i1} = 0.03$ and $K_{d1} = 0.0002$. In order to test the robustness of the proposed tuning method under practical conditions, measurement noise is introduced to the system in the simulation study. Figure 4.6 shows the tracking error incurred under the feedback controller PID_1.

As shown in Figure 4.6, the initial PID controller (PID_1) alone yields $e_{RMS} = 12.9\mu m$ and $e_{MAX} = 28.3\mu m$ in terms of the absolute maximum tracking error and the RMS tracking error.

Next, the ILC scheme, as discussed in Section 4.2, is applied to the system to further reduce the tracking error, and in the process, yields the input and output signals neces-

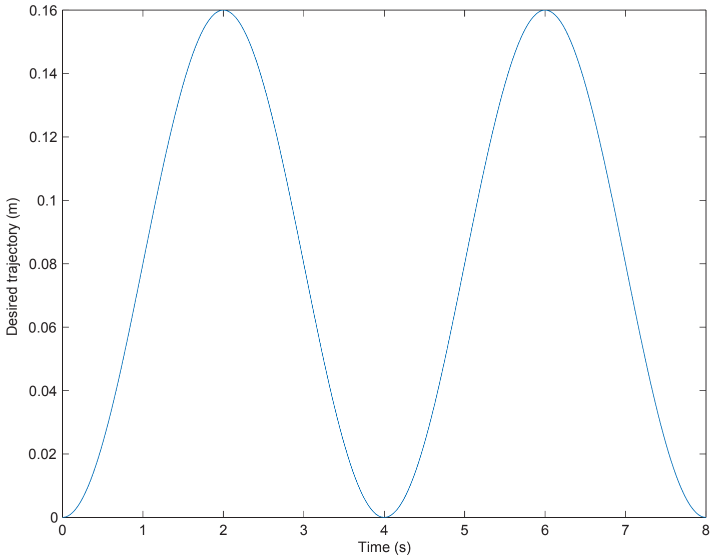


Figure 4.5: Desired trajectory

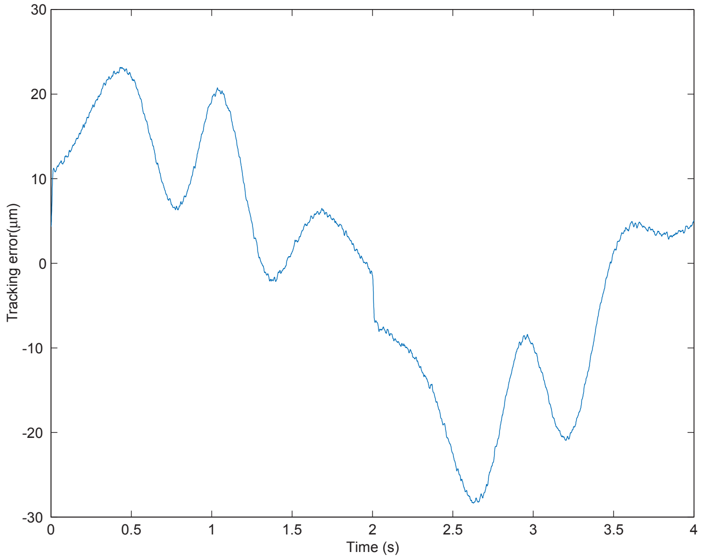


Figure 4.6: Tracking error with the feedback controller PID_1

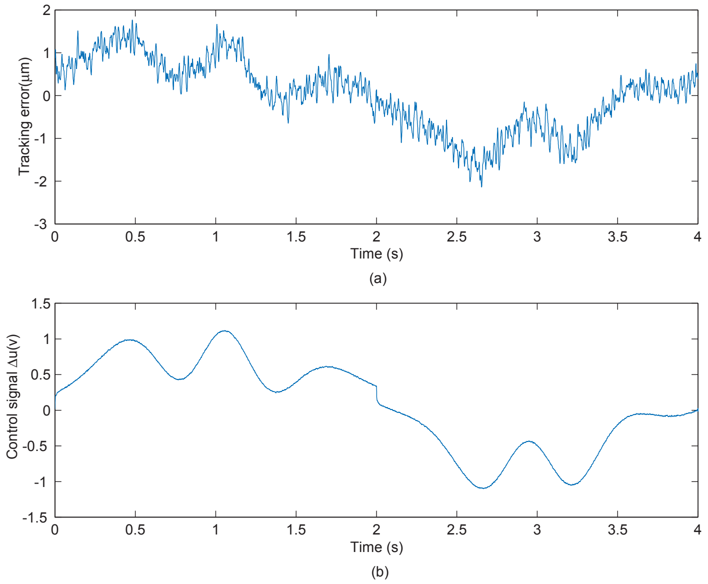


Figure 4.7: Tracking error during the 20th cycle (a). tracking error (μm) (b). control signal Δu(v)

sary for the tuning of the new PID controller. The learning gain is chosen as $\lambda = 0.14$. The result is shown in Figure 4.7 after 20 iterations. The tracking performance is enhanced significantly with $e_{RMS} = 0.82\mu m$ and $e_{MAX} = 2.1\mu m$. Figure 4.8 shows the convergence performance of the ILC scheme over 20 cycles.

At this stage, it is ready to compute the parameters of PID$_2$. The tracking error $e(t)$ and the control signal $\Delta u(t)$, during the 20th cycle, are used as the input and the output signals to determine the parameters. These signals are first filtered using the low pass filter $H_f(s)$. The filter $H_f(s)$ is designed as

$$H_f(s) = \frac{1600}{s^2 + 80s + 1600}. \tag{4.16}$$

The least square algorithm is used to determine the estimates of the PID$_2$ parameters.

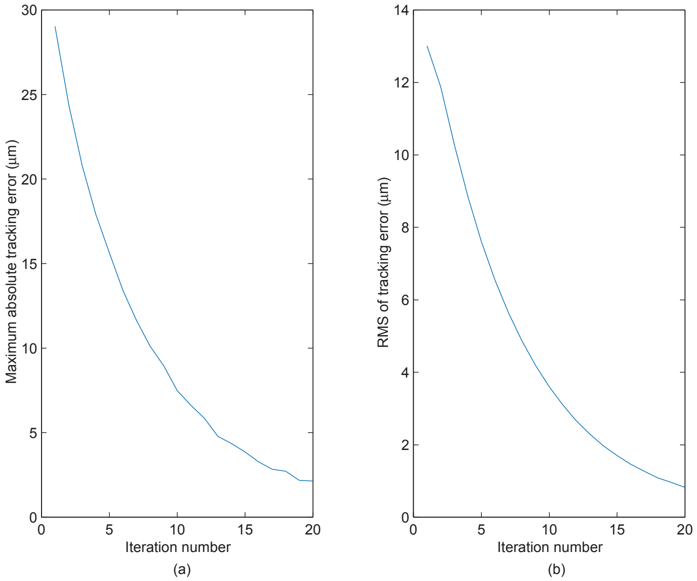


Figure 4.8: Iterative convergence performance (a). maximum tracking error (b). RMS tracking error

The best fitting parameters are calculated to be

$$K_{p2} = 0.7436,\ K_{i2} = 0.2612\ and\ K_{d2} = 0.0051. \tag{4.17}$$

Thus, the final PID controller is tuned with the parameters: $K_p = 0.7886$, $K_i = 0.2912$ and $K_d = 0.0053$. Figure 4.9 shows the tracking performance when the tuned PID controller is applied to the system without any ILC component. A good performance with $e_{RMS} = 0.81\mu m$ and $e_{MAX} = 1.63\mu m$ is achieved. The maximum tracking error is reduced significantly, compared to the ILC scheme shown in Figure 4.7. This is possible as the PID controller can suppress the noise in the system effectively, whereas the ILC is well-known to be sensitive to noise and sharp changes in reference commands.

Next, a non-repetitive reference signal is used to simulate the case where the natural reference signal is a non-periodic one, such as step-types of reference signal for non-

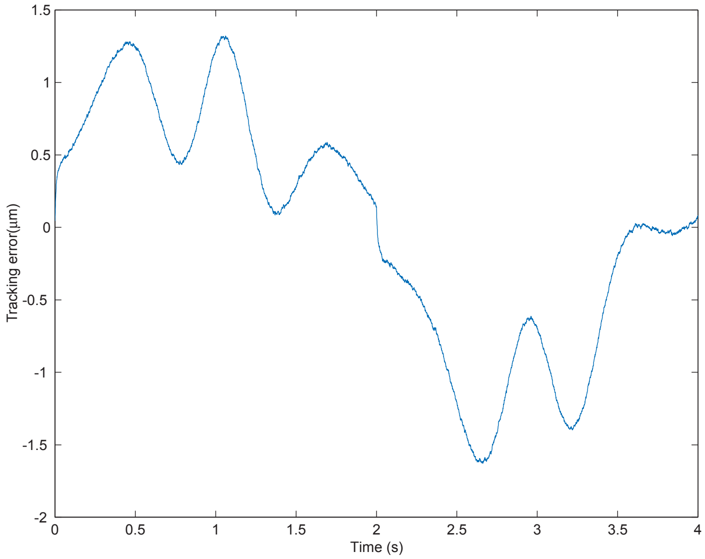


Figure 4.9: Tracking error with the tuned PID controller

repetitive point-to-point positioning. Figure 4.10 shows the performance comparison for the setpoint following. The circled parts in the figure are enlarged as shown in Figure 4.11. From the figures, it can be observed that good performance can be achieved with the proposed PID tuning method, achieving shorter rise time and settling time. Note that the vertical scale is in terms of $10^4 \mu m$, so the improvement is significant.

4.4 Experimental Study

In order to demonstrate the effectiveness of the proposed PID tuning method, experiments were conducted on a single axis stage manufactured by Steinmeyer. A SP-8 piezoelectric motor is used to drive the stage. Table 4.1 shows the specifications of the stage and the motor. The experimental studies were conducted on a dSPACE DS1102 control board. MATLAB and SIMULINK were the control platforms used for the ex-

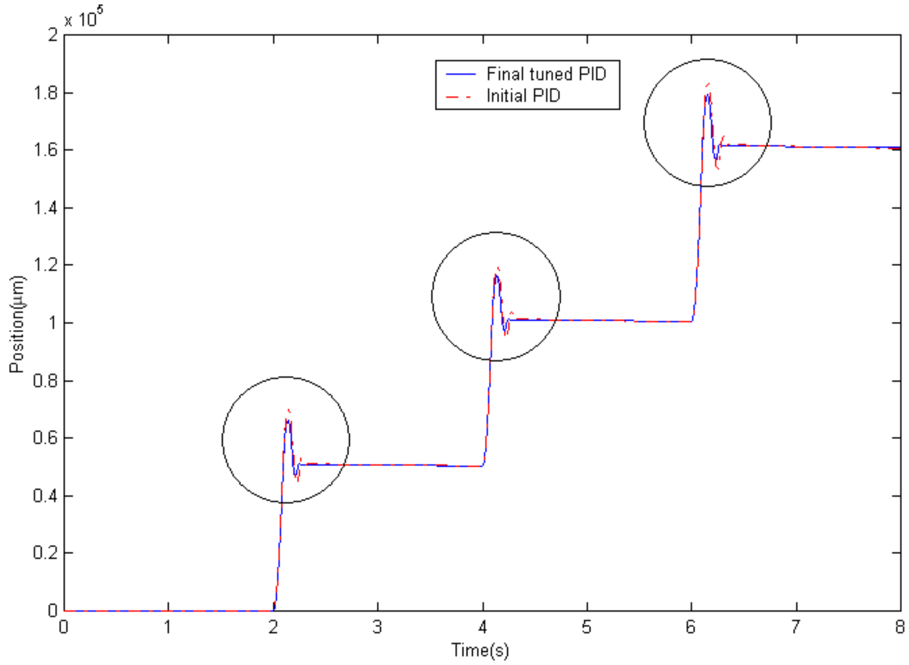


Figure 4.10: Comparison of performances for step changes in setpoint

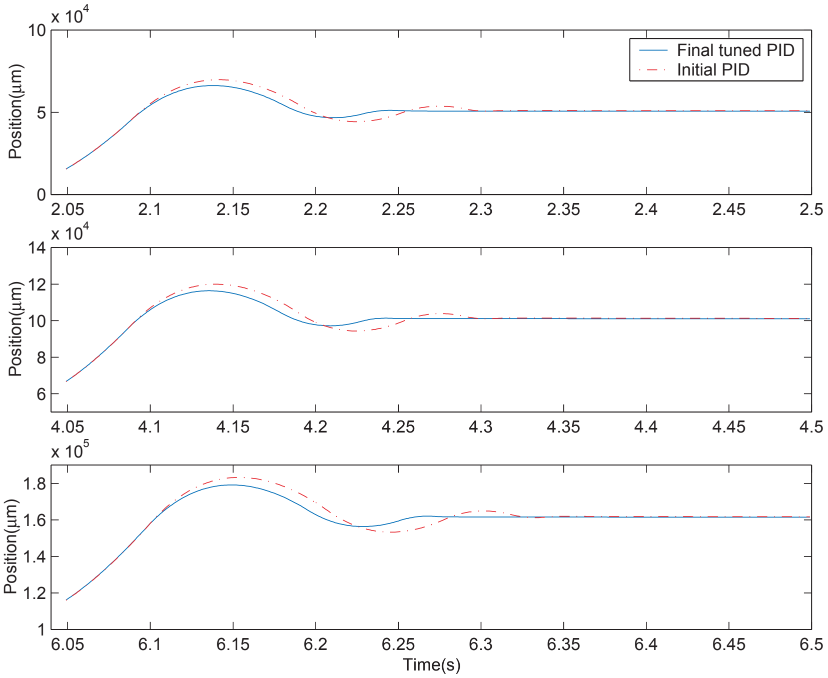


Figure 4.11: Magnified parts

Table 4.1: Specifications of Piezoelectric Linear Motor

Travel	Velocity(Max)	Resolution	Output force(Max)
200mm	250 mm/s	0.1μ m	40N

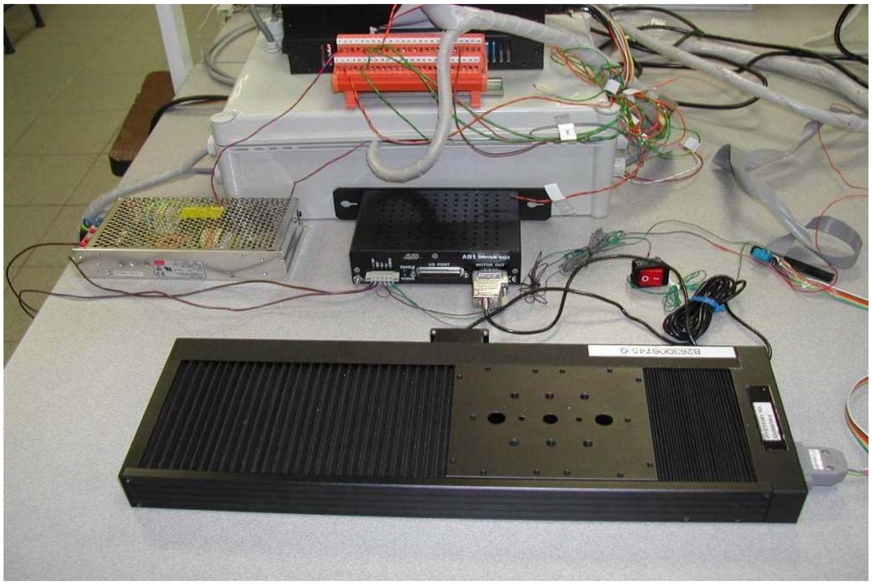


Figure 4.12: Setup of the linear-piezoelectric motor

periment. The control scheme is implemented in the form of a C-coded S-function. A sampling frequency of $1kHz$ is configured. Figure 4.12 shows the experimental test platform.

A sinusoidal reference signal is used in the experimental study with a period of $3s$, shown in Figure 4.13. The parameters of the initial feedback controller PID_1 are set as: $K_{p1} = 0.009$ $K_{i1} = 0.021$ and $K_{d1} = 0.000001$. Figure 4.14 shows the tracking error incurred under the initial feedback controller PID_1. The above experimental result yields $e_{RMS} = 102.99\mu m$ and $e_{MAX} = 245.71\mu m$.

Next, the ILC scheme is applied to the system to reduce the tracking error further and achieve the input and output signal for the tuning of the new PID controller. The learning gain is chosen as $\lambda = 0.04$. Under the ILC scheme, the result is shown in Figure

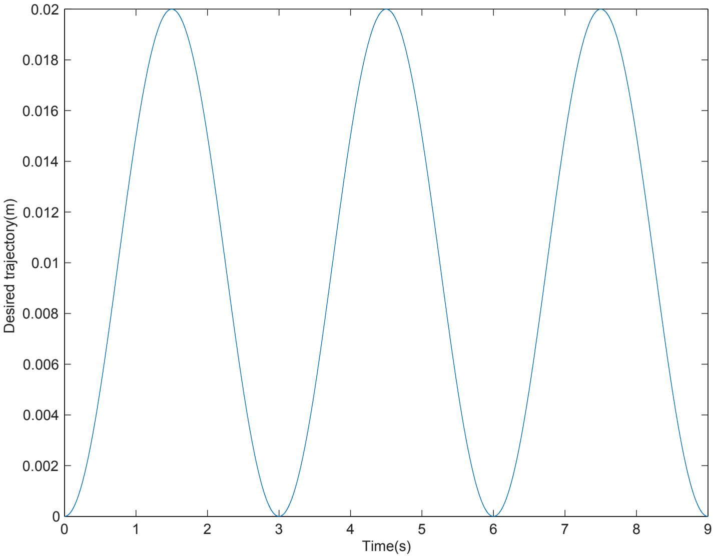


Figure 4.13: Desired trajectory used in the experimental study

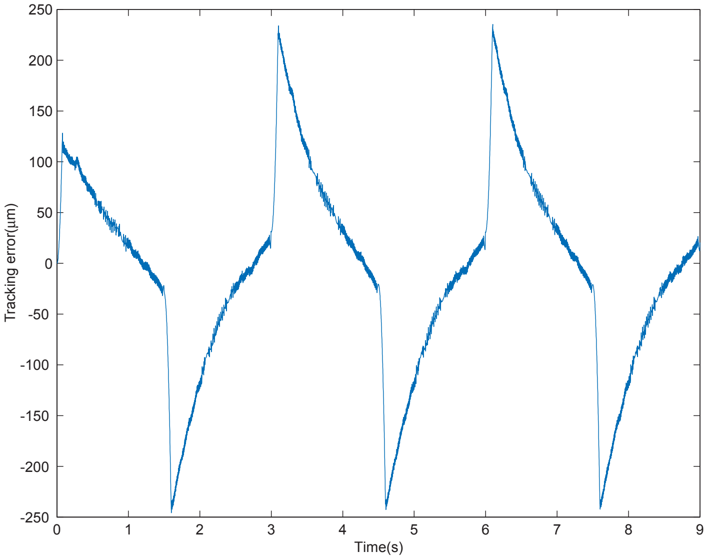


Figure 4.14: Tracking error with the initial feedback controller PID_1

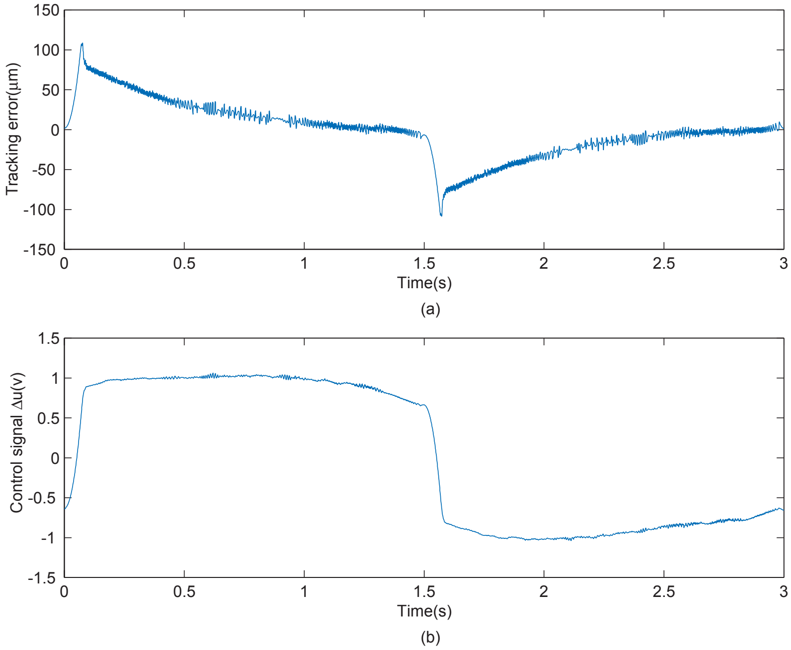


Figure 4.15: Tracking error during the 30th cycle (a). tracking error (μm) (b). control signal $\Delta u(v)$

4.15 after 30 iterations. From the figure, it can be observed that the tracking performance is clearly enhanced with the learning scheme, compared to the initial feedback controller alone. In the 30th cycle, the performance indices of $e_{RMS} = 34.34\mu m$ and $e_{MAX} = 107.72\mu m$ are achieved. The tracking convergence is shown in Figure 4.16. Note that the blip in Figure 4.16(b) is due to the initialization transience associated with the ILC law.

With the information obtained from the 30th cycle, the parameters of PID$_2$ can be estimated. In the experimental study, the low pass filter is still designed as in (4.17). The best fitting parameters are determined as

$$K_{p2} = 0.0186,\ K_{i2} = 0.0282\ and\ K_{d2} = 0.0000105. \tag{4.18}$$

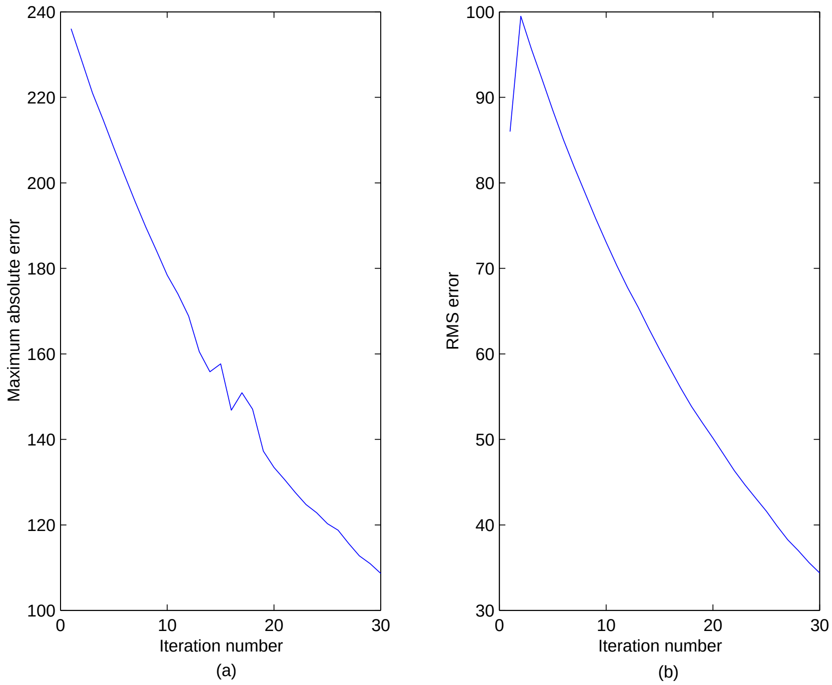


Figure 4.16: Iterative convergence performance (a). maximum tracking error (b). RMS tracking error

The final PID controller is thus obtained with $K_p = 0.0276$, $K_i = 0.0492$ and $K_d = 0.0000115$. Then, the final PID controller is applied to the system without the ILC component. The tracking performance is shown in Figure 4.17 with $e_{RMS} = 39.3\mu m$ and $e_{MAX} = 93.4\mu m$. Improved tracking performance is obtained with the tuned PID controller. Similar to the phenomenon observed in the simulation study, the maximum tracking error is reduced significantly.

In the experimental study, tracking results for non-repetitive setpoint following are also observed. Figure 4.18 shows the performance for the setpoint following. In order to present the performance clearly, the circled parts in the figured are magnified and shown in Figure 4.19. Improved performance is observed when compared to the initial

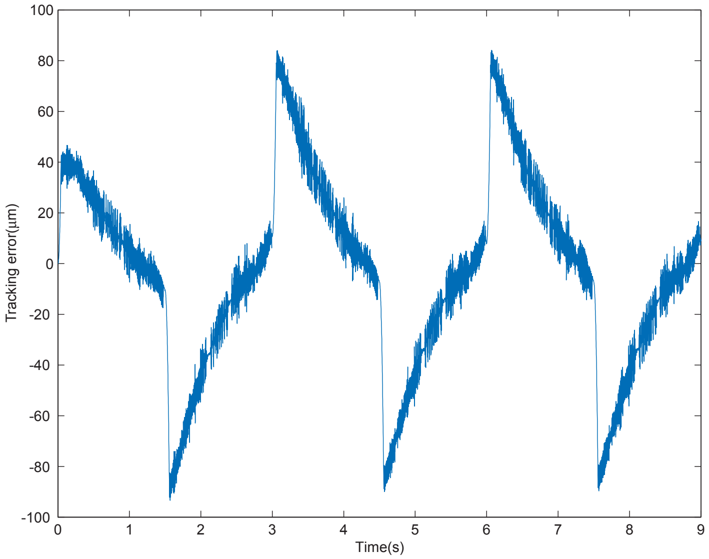


Figure 4.17: Tracking error with the tuned PID controller

PID controller. Note again that the vertical scale is in terms of $10^4 \mu m$.

4.5 Conclusions

In this chapter, a new approach for closed-loop automatic tuning of PID controller based on an Iterative Learning Control (ILC) approach is proposed and developed. The method does not require the control loop to be detached for tuning purposes, but it requires the input of a periodic reference signal. Such a reference signal can be the natural reference signal of the control system when it is used to execute repetitive operational sequence, or it can be an excitation signal purely for tuning the PID controller. A modified iterative learning control scheme iteratively changes the control signal by adjusting the reference signal only. Once a satisfactory performance is achieved, the PID controller is tuned by fitting the controller to yield a close input and output characteristics of the ILC

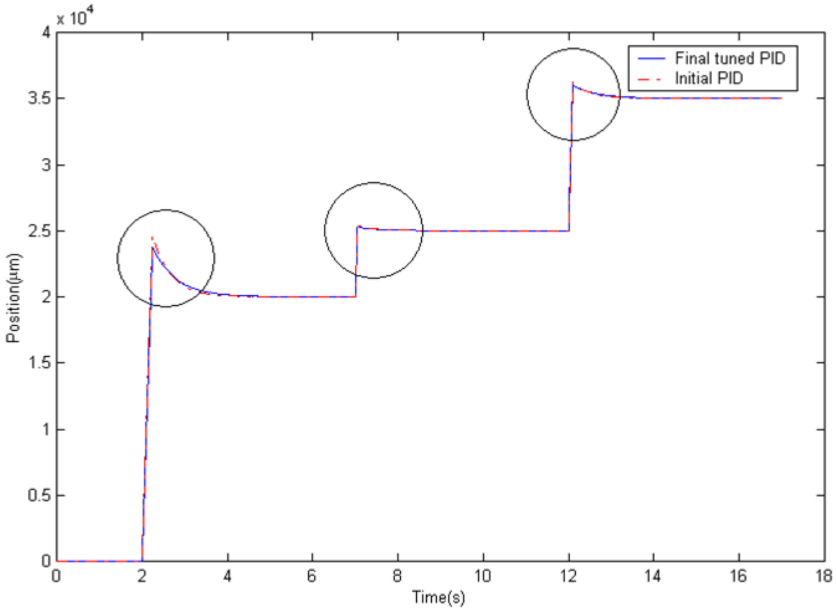


Figure 4.18: Performance comparison for step changes in setpoint

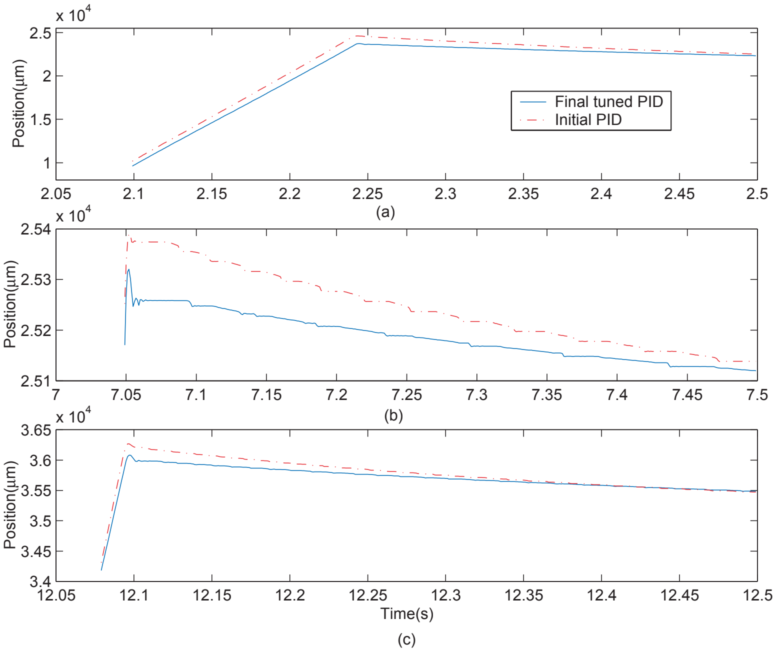


Figure 4.19: Magnified parts

component via a standard least squares algorithm. Simulation and experimental results verify the effectiveness of the proposed tuning method positively.

Chapter 5

Repetitive Control for Time-Delay Systems

5.1 Introduction

Time delays are inevitable in many manufacturing processes. For example, in the steel rolling process, the thickness measurement point is often located at a distance from the hard press point, giving rise to a measurement delay. Compared to systems without delay, time-delay systems are well-known to be difficult to control to achieve satisfactory performance and stability. Many authors deal with the control of the time-delay systems via current state feedback controllers [85][86][87]. These approaches are sensitive to parameter uncertainties, and they assume that bounds on these uncertainties are known and available prior to the control design.

The diffences and connections between iterative learning control and repetitive control are pointed out in Chapter 1. Repetitive Control (RC) is a learning controller applicable to execute repetitive operations, although it has yet to achieve the same level of effectiveness for time-delay systems as for delay-free systems. The RC in its current state cannot yield satisfactory error convergence for time-delay systems even if a small learning gain is used, when the systems are not pre-compensated with some dead-time compensation schemes which require a full process model. In many cases, divergence can occur despite the use of a small gain. However, for delay-free systems, RC has become popular, and its potential has been demonstrated in real industrial applications. RC is able to utilize the system's repetition to compensate or reject uncertainties and disturbances, and hence able to track a prescribed periodic trajectory. RC can be considered as "no-reset" ILC. Several important findings of ILC for linear systems have appeared in the open literature [8][88][89][90][91]. For nonlinear systems, robust ILC schemes have been proposed by Horowitz et al. [92], Xu and Qu [93], Xu et al. [94] and Chen et al. [95]. However, studies which consider the presence of time delay in the input signal have been scarce. This is a reason why the application of ILC or RC to process control problems has been rarely reported. This phenomenon motivates some effort to develop RC schemes for time-delay systems. Chen et al. [96] investigated a robust control problem of state-delay systems using the ILC algorithm. In [43] and [44], robust ILC designs under the framework of a Smith predictor controller is proposed, where the time-delay is compensated via the Smith structure so that the compensated system appears as delay-free to the ILC.

In this chapter, a modified RC configuration is proposed, which is more general than currently reported ones, and which is applicable to extended classes of processes, including time-delay processes. A key and prominent feature of this new RC is the inclusion of a time shift block to allow the scheme to be adapted to systems with a large time-delay and phase lag phenomenon. Via the time shift block, this new and modified RC configuration is able to provide time-delay compensation during the learning phase. A new necessary and sufficient condition is derived to ensure convergence of the tracking error under this configuration. In addition, a robust convergence analysis is provided to study the convergence of the RC under time-delay modelling error, initialization errors, disturbances and measurement noise. In the chapter, simulation examples illustrating the practical application of this repetitive control to process control problems are provided.

5.2 RC Configuration for Time-Delay Systems

In this chapter, the system under consideration possesses an input time delay. Figure 5.1 shows the standard RC configuration. Unfortunately, while this configuration works well for robotic and servo control applications with relatively small time delays, it fails in the realm of industrial control applications and requirements due to the typical presence of time-delay and large phase lag. When the usual RC is applied to the time-delay system, the error at k time instant is used to calculate the next RC output. However, due to the time lag phenomenon, the actual system output can be affected only after a time duration. This typically results in large or even divergent tracking errors, even if a small learning gain is used. To-date, RC systems which are applicable to industrial control applications are very rarely reported.

A new RC configuration is proposed as shown in Figure 5.2 which is suitable for industrial control applications. The system under control is represented by $G_0(s)e^{-Ls}$,

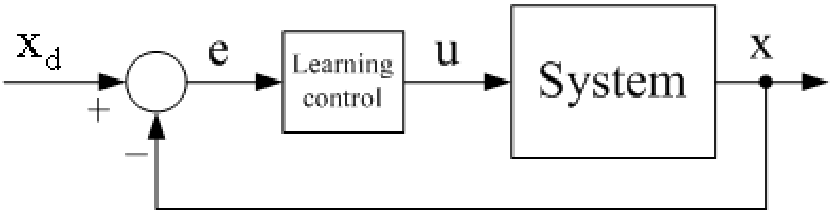


Figure 5.1: Learning Control block diagram

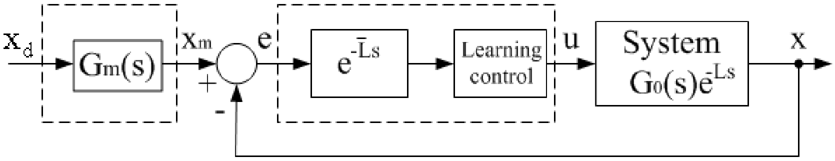


Figure 5.2: Learning control structure for the time-delay system

where L is the time delay present in the input. Unlike the standard RC scheme, an additional time delay component is included to delay the tracking error $e(t)$ by a time duration of $\bar{L}$ before it is fed to the RC. An appropriate design of this delay can achieve the effect of time-delay compensation for convergent RC tuning. An analysis is provided to show this key aspect of the modified RC.

In the figure, the dotted block denoted by $G_m(s)$ represents an optional reference model for the closed-loop which can then be used to generate the tracking error more effectively. G_m can be fixed as $G_m = 1$ (i.e., no G_m block at all) or a simple rational function to obtain a continuous and more realizable reference signal. In the subsequent developments in the chapter, unless otherwise specified, the development is illustrated by using $G_m = 1$ with no loss in generality.

In the state space form, the time-delay system with input delay can be described by:

$$\begin{aligned} \dot{\chi}(t) &= A\chi(t) + Bu(t-L), \\ x(t) &= C\chi(t). \end{aligned} \tag{5.1}$$

Correspondingly, in the discrete-time domain, it can be obtained

$$\begin{aligned} \chi(k+1) &= F\chi(k) + Gu(k-N), \quad \chi(0) = \chi_0 \\ &= F\chi(k) + Gq^{-N}u(k), \\ x(k) &= C\chi(k), \end{aligned} \tag{5.2}$$

where $F = e^{Ah}$, $G = \int_0^h e^{Ah}Bdt$, $N = L/h$, and h denotes the sampling interval.

For the new proposed configuration, the RC updating law, including the time delay $e^{-\bar{L}s}$, is given by

$$u_{i+1}(k) = u_{i-1}(k) + \gamma e_i(k - \bar{N} + 1), \tag{5.3}$$

where the subscript i represents the *ith* iteration cycle, $e_i = x_d - x_i$ and $\bar{N} = \frac{\bar{L}}{h}$.

Theorem 5.1. Consider the system described by (5.2). Using the RC law (5.3) with $\bar{N} = T - N$ where $T = \frac{T_r}{h}$ and T_r is the period of the reference signal, a necessary and sufficient condition for error convergence is given by

$$|z_{1,2}| = |1 - \gamma CG| < 1. \tag{5.4}$$

Proof:

In the idea of repetitive control, consider a discrete-time system with the plant initial conditions carried over from previous iteration. Consider the ith repetitive cycle with $k = 0, 1....T$, the system (5.2) can be described [97] as

$$\begin{aligned} x_i(0) &= x_{i-1}(T) = C\chi_{i-1}(T) \\ x_i &= M_0\chi_0 + Mq^{-N}u_i \\ &= M_0\chi_0 + \tilde{M}u_i, \end{aligned} \tag{5.5}$$

with

$$x_i = [x_i(1)\ x_i(2)\ ...\ x_i(T)]^{\mathtt{T}}, \tag{5.6}$$

$$u_i = [u_i(0)\ u_i(1)\ ...\ u_i(T-1)]^{\mathtt{T}}, \tag{5.7}$$

$$M_0 = \begin{bmatrix} CF \\ CF^2 \\ \vdots \\ CF^T \end{bmatrix}, \tag{5.8}$$

$$M = \begin{bmatrix} CG & 0 & ... & 0 \\ CFG & CG & ... & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CF^{T-1}G & ... & CFG & CG \end{bmatrix}, \tag{5.9}$$

$$\tilde{M} = Mq^{-N}. \tag{5.10}$$

The error, during the i-th cycle, is given by $e_i = [e_i(1)\ e_i(2)\ ...\ e_i(T)]^{\mathtt{T}}$. With this definition, the output error equation can be obtained as

$$\begin{aligned} e_{i+1} &= x_d - x_{i+1} \\ &= x_d - x_{i-1} + x_{i-1} - x_{i+1} \\ &= e_{i-1} + M_0\chi_{i-1}(0) + \tilde{M}u_{i-1} - M_0\chi_{i+1}(0) - \tilde{M}u_{i+1} \\ &= e_{i-1} + \tilde{M}(u_{i-1} - u_{i+1}) + M_0(\chi_{i-1}(0) - \chi_{i+1}(0)). \end{aligned} \tag{5.11}$$

Using the RC updating law (5.3), (5.11) can be rewritten in terms of the error variable only as

$$\begin{aligned} e_{i+1} &= e_{i-1} - \gamma \tilde{M} q^{-\bar{N}} e_i + M_0(\chi_{i-1}(0) - \chi_{i+1}(0)), \\ &= e_{i-1} - \gamma M q^{-(\bar{N}+N)} e_i + M_0(\chi_{i-1}(0) - \chi_{i+1}(0)). \end{aligned} \tag{5.12}$$

Using $\bar{N} = T - N$, (5.12) becomes

$$\begin{aligned} e_{i+1} &= e_{i-1} - \gamma M q^{-T} e_i + M_0(\chi_{i-1}(0) - \chi_{i+1}(0)), \\ &= e_{i-1} - \gamma M e_{i-1} + M_0(\chi_{i-1}(0) - \chi_{i+1}(0)) \\ &= (I - \gamma M) e_{i-1} + M_0(\chi_{i-1}(0) - \chi_{i+1}(0)). \end{aligned} \tag{5.13}$$

Here, the residual part $M_0(\chi_{i-1}(0) - \chi_{i+1}(0))$ can be considered as disturbance, which has no influence on the convergence. Since M is a lower triangular matrix, the difference equation (5.13) in term of iteration axis for $k = 1, 2...T$ has the same characteristic equation

$$z^2 - (1 - \gamma CG) = 0. \tag{5.14}$$

If all the roots of the characteristic equation lie within the unit circle, the error e_i is convergent as $i \to \infty$. Thus, the necessary and sufficient condition for convergence is given by

$$|z_{1,2}| = |1 - \gamma CG| < 1.$$

The proof is completed.

Remark 5.1. Note that this condition is similar to the convergence condition for delay-free systems, and it is obtained without a need to have a model-based error prediction mechanism in place.

Remark 5.2. Note also that with the modified RC configuration, error convergence can be achieved, via only a time shift, for time-delay systems without any error prediction requirement ahead of time (i.e., $e_i(t+N)$). Only prior knowledge of the time-delay L is necessary, but not the full process model which is the case when ILC is used in conjunction with a Smith predictor controller [43].

5.3 Robust Convergence Analysis

Since L is a parameter needed in this repetitive control scheme, it is interesting to explore the convergence in the face of error in the modelling of L, as well as the existence of initialization errors, disturbances and measurement noise. Note that the conventional RC (i.e., $\bar{L} = 0$) applied to delay systems is thus a special case of this analysis, with the time-delay modeling error of

$$\Delta L = T_r - L. \tag{5.15}$$

In this section, the robust convergence analysis of the proposed method is presented. Uniform error convergence can be shown to be attainable when disturbances and uncertainty tend to zero.

The following time-delay system with disturbances and measurement noise is considered

$$\begin{aligned} \chi_i(k+1) &= F\chi_i(k) + Gu_i(k-N) + w_i(k) \\ x_i(k) &= C\chi_i(k) + v_i(k), \end{aligned} \tag{5.16}$$

where $w_i(k)$ and $v_i(k)$ represent the disturbances and the measurement noise respectively. Considering a modeling error in time-delay, let $\bar{N} = T - (N + \Delta N)$, where $\Delta N = \frac{\Delta L}{h}$, and ΔL represents the modeling error in the estimate of the time-delay L. The following assumptions are made

Assumptions:

(1). The initialization error is bounded, i.e.,

$$\parallel \chi_d(0) - \chi_i(0) \parallel \leq b_{\chi_0}, \ \forall i.$$

(2). The desired output $x_d(k)$ is realizable. This implies that for a given bounded

$x_d(k)$, there exists a unique bounded desired input $u_d(k)$, $k \in [0,T]$ which satisfies

$$\begin{aligned} \chi_d(k+1) &= F\chi_d(k) + Gu_d(k-N), \\ x_d(k) &= C\chi_d(k). \end{aligned} \tag{5.17}$$

(3). The disturbance $w_i(t)$ and the measurement noise $v_i(t)$ are bounded, i.e.,

$$b_w \overset{\triangle}{=} \sup_{k\in[0,T]} \parallel w_i(k) \parallel, b_v \overset{\triangle}{=} \sup_{k\in[0,T]} \parallel v_i(k) \parallel .$$

Definitions:

The following norms are used in this chapter. They are defined as

$$\begin{aligned} \parallel g \parallel &= \max_{1\leq i\leq n} \mid g_i \mid, \\ \parallel H \parallel &= \max_{1\leq i\leq m} (\sum_{j=1}^{n} \mid h_{i,j} \mid), \end{aligned} \tag{5.18}$$

where $g = [g_1,\ g_2,\ ...g_n]^T$ is a vector and $H = [h_{i,j}]_{m\times n}$ is a matrix.

Define $a = \parallel F \parallel$. The λ norm is defined as

$$\parallel f(k) \parallel_\lambda = \sup_{k\in[0,T]} a^{-\lambda k} \parallel f(k) \parallel, \tag{5.19}$$

where $f(k)$ $(k = 0, 1, 2...T)$ is a discrete-time vector. λ is chosen as $\mid \lambda \mid > 1$ and the choice of the sign depends on a, i.e.,

$$\begin{cases} \lambda > 0 & if \ \ a > 1 \\ \lambda < 0 & if \ \ a < 1. \end{cases}$$

Theorem 5.2. Consider the system described by (5.16) which satisfies the above assumptions. Given a realizable trajectory $x_d(.)$, there exists a γ which results in the output error $e_i(k)$, and input error $\Delta u_i(k)$ being bounded under the proposed scheme.

Proof: Accounting for the modeling error in the time-delay, the control update law can be written as

$$\begin{aligned} u_{i+1}(k) &= u_{i-1}(k) + \gamma e_i(k - \bar{N} + 1) \\ &= u_{i-1}(k) + \gamma e_{i-1}(k + N + 1 + \Delta N). \end{aligned} \tag{5.20}$$

Assume that e is continuous and differentiable. According to Lagrange's Theorem, there exists a τ value such that $e_{i-1}(k + N + 1 + \Delta N)$ can be written as

$$e_{i-1}(k + N + 1 + \Delta N) = e_{i-1}(k + N + 1) + \Delta N e'_{i-1}(\tau). \tag{5.21}$$

Define

$$\varepsilon = \gamma \Delta N e'_{i-1}(\tau).$$

Since e is continuous and differentiable, ε is bounded as

$$\sup_{k \in [0,T]} \parallel \varepsilon \parallel \leq b_{\varepsilon}.$$

(5.20) can this be written as

$$u_{i+1}(k) = u_{i-1}(k) + \gamma e_{i-1}(k + N + 1) + \varepsilon. \tag{5.22}$$

Bound on Input Error:

Define $\Delta u_i(k) = u_d(k) - u_i(k)$ such that

$$\begin{aligned}
\Delta u_{i+1}(k) &= u_d(k) - u_{i+1}(k) \\
&= \Delta u_{i-1}(k) - \gamma e_{i-1}(k+N+1) - \varepsilon \\
&= \Delta u_{i-1}(k) - \gamma[C\chi_d(k+N+1) - C\chi_{i-1}(k+N+1) - v_{i-1}(k+N+1)] \\
&\quad -\varepsilon. \qquad (5.23)
\end{aligned}$$

By referring to (5.16) and (5.17), (5.23) can be written as

$$\begin{aligned}
\Delta u_{i+1}(k) &= \Delta u_{i-1}(k) - \gamma[CF\chi_d(k+N) + CGu_d(k) - CF\chi_{i-1}(k+N) - CGu_{i-1}(k)] \\
&\quad +\gamma C w_{i-1}(k+N) + \gamma v_{i-1}(k+N+1) - \varepsilon \\
&= \Delta u_{i-1}(k) - \gamma[CF(\chi_d(k+N) - \chi_{i-1}(k+N)) + CG\Delta u_{i-1}] \\
&\quad +\gamma C w_{i-1}(k+N) + \gamma v_{i-1}(k+N+1) - \varepsilon \\
&= [I - \gamma CG]\Delta u_{i-1}(k) - \gamma CF(\chi_d(k+N) - \chi_{i-1}(k+N)) \\
&\quad +\gamma C w_{i-1}(k+N) + \gamma v_{i-1}(k+N+1) - \varepsilon. \qquad (5.24)
\end{aligned}$$

Taking norms for (5.24) gives

$$\begin{aligned}
\parallel \Delta u_{i+1}(k) \parallel &\leq \parallel I - \gamma CG \parallel \parallel \Delta u_{i-1}(k) \parallel \\
&\quad + \parallel \gamma CF \parallel \parallel \chi_d(k+N) - \chi_{i-1}(k+N) \parallel \\
&\quad +b_w \parallel \gamma C \parallel + b_v\gamma + b_\varepsilon. \qquad (5.25)
\end{aligned}$$

$\chi_d(k) - \chi_i(k)$ can be written in terms of the control signal by referring to (5.5)

$$\begin{aligned}\chi_d(k) - \chi_i(k) &= F^k(\chi_d(0) - \chi_i(0)) + \sum_{j=0}^{k-1} F^{k-1-j} G(u_d(j-N) - u_i(j-N)) \\ &\quad - \sum_{j=0}^{k-1} F^{k-1-j} w_i(k). \end{aligned} \tag{5.26}$$

Thus, it can be obtained as

$$\begin{aligned}\chi_d(k+N) - \chi_{i-1}(k+N) &= F^{k+N}(\chi_d(0) - \chi_{i-1}(0)) \\ &\quad + \sum_{j=0}^{k+N-1} F^{k+N-1-j} G(u_d(j-N) - u_{i-1}(j-N)) \\ &\quad - \sum_{j=0}^{k+N-1} F^{k+N-1-j} w_{i-1}(j). \end{aligned} \tag{5.27}$$

Taking the norm for the above equation yields

$$\begin{aligned}\parallel \chi_d(k+N) - \chi_{i-1}(k+N) \parallel &\leq a^{k+N} \parallel \chi_d(0) - \chi_{i-1}(0) \parallel \\ &\quad + \sum_{j=0}^{k+N-1} a^{k+N-1-j} \parallel G \parallel \parallel u_d(j-N) - u_{i-1}(j-N) \parallel \\ &\quad + b_w \sum_{j=0}^{k+N-1} a^{k+N-1-j}. \end{aligned} \tag{5.28}$$

Define $\parallel I - \gamma CG \parallel \leq \rho < 1$, $b_C = \parallel C \parallel$ and $b_G = \parallel G \parallel$. Therefore, (5.25) can be written as

$$\begin{aligned}\parallel \Delta u_{i+1}(k) \parallel &\leq \rho \parallel \Delta u_{i-1}(k) \parallel + \gamma b_C a^{k+N+1} b_{\chi_0} \\ &\quad + \gamma b_C b_G a \sum_{j=0}^{k+N-1} a^{k+N-1-j} \parallel \Delta u_{i-1}(j-N) \parallel \\ &\quad + \gamma b_w b_c a \sum_{j=0}^{k+N-1} a^{k+N-1-j} + \gamma b_w b_C + b_v \gamma + b_\varepsilon. \end{aligned} \tag{5.29}$$

Multiplying $a^{-\lambda k}$ to both sides of (5.29), it can be obtained as

$$\begin{aligned}\parallel \Delta u_{i+1}(k) \parallel_\lambda \;\; &\leq \;\; \rho \parallel \Delta u_{i-1}(k) \parallel_\lambda + \gamma b_C b_{\chi 0} a^{N+1} \sup_{k\in[0,T]} (a^{-\lambda k} a^k) \\ &\quad + \gamma b_C b_G a \sup_{k\in[0,T]} a^{-\lambda k} \sum_{j=0}^{k+N-1} a^{k+N-1-j} \parallel \Delta u_{i-1}(j-N) \parallel \\ &\quad + \gamma b_w b_c a \sup_{k\in[0,T]} a^{-\lambda k} \sum_{j=0}^{k+N-1} a^{k+N-1-j} + \gamma b_w b_C + b_v \gamma + b_\varepsilon. \end{aligned} \tag{5.30}$$

At this stage, it is noted that

$$\sup_{k\in[0,T]} (a^{-\lambda k} a^k) = a. \tag{5.31}$$

Let us define

$$b_1 = \gamma b_C b_{\chi 0} a^{N+2} + \gamma b_w b_c a \sup_{k\in[0,T]} a^{-\lambda k} \sum_{j=0}^{k+N-1} a^{k+N-1-j} + \gamma b_w b_C + b_v \gamma + b_\varepsilon.$$

Additionally, it follows that

$$\begin{aligned} & \gamma b_C b_G a \sup_{k\in[0,T]} a^{-\lambda k} \sum_{j=0}^{k+N-1} a^{k+N-1-j} \parallel \Delta u_{i-1}(j-N) \parallel \\ = \;\; & \gamma b_c b_G \sup_{k\in[0,T]} a^{-(\lambda-1)k} \sum_{j=0}^{k+N-1} a^{-\lambda(j-N)} \parallel \Delta u_{i-1}(j-N) \parallel a^{(\lambda-1)(j-N)} \\ \leq \;\; & \gamma b_C b_G \parallel \Delta u_{i-1}(k) \parallel_\lambda \sup_{k\in[0,T]} a^{-(\lambda-1)k} \left(\frac{a^{(\lambda-1)k}-1}{a^{\lambda-1}-1} \right) \\ = \;\; & \gamma b_C b_G \parallel \Delta u_{i-1}(k) \parallel_\lambda \left(\frac{1-a^{-(\lambda-1)T}}{a^{\lambda-1}-1} \right). \end{aligned} \tag{5.32}$$

Define $b_2 = \gamma b_C b_G \left(\frac{1-a^{-(\lambda-1)(T+N)}}{a^{\lambda-1}-1} \right)$. Substituting (5.32) into (5.30), it can be obtained as

$$\parallel \Delta u_{i+1}(k) \parallel_\lambda \;\; \leq \;\; (\rho + b_2) \parallel \Delta u_{i-1}(k) \parallel_\lambda + b_1. \tag{5.33}$$

Define $\tilde{\rho} = \rho + b_2$. Thus, (5.33) can be written as

$$\parallel \Delta u_{i+1}(k) \parallel_\lambda \;\; \leq \;\; \tilde{\rho} \parallel \Delta u_{i-1}(k) \parallel_\lambda + b_1. \tag{5.34}$$

Since $\rho < 1$, there exists a sufficiently large $\mid \lambda \mid$ which makes $\tilde{\rho} < 1$. Applying Lemma A.0.2 in [98], it can be obtained

$$\lim_{i \to \infty} \parallel \Delta u_i(k) \parallel_\lambda = \frac{b_1}{1 - \tilde{\rho}}. \tag{5.35}$$

Since b_1 is bounded, (5.35) shows that the input error is bounded $\forall i$ in $[0, \quad T]$. Also when all the disturbances tend to zero, the input error tend to zero.

Bound on Output Error:

Next, it is proven that the output error is also bounded with appropriate choice of the learning gain γ. Following (5.28), it can be written as

$$\begin{aligned} \parallel e_i(k) \parallel &\leq \parallel C \parallel \parallel \chi_d(k) - \chi_i(k) \parallel \\ &\leq b_c a^k \parallel \chi_d(0) - \chi_i(0) \parallel \\ &\quad + b_c \sum_{j=0}^{k-1} a^{k-1-j} \parallel G \parallel \parallel u_d(j-N) - u_i(j-N) \parallel \\ &\quad + b_c b_w \sum_{j=0}^{k+N-1} a^{k-1-j}. \end{aligned} \tag{5.36}$$

Multiplying $a^{-\lambda k}$ to both sides of (5.36),

$$\begin{aligned} \parallel e_i(k) \parallel_\lambda &\leq b_c b_{\chi_0} \sup_{k \in [0,T]} (a^{-\lambda k} a^k) \\ &\quad + b_c b_G \sup_{k \in [0,T]} a^{-\lambda k} \sum_{j=0}^{k-1} a^{k-1-j} \parallel u_d(j-N) - u_i(j-N) \parallel \\ &\quad + b_c b_w \sup_{k \in [0,T]} a^{-\lambda k} \sum_{j=0}^{k-1} a^{k-1-j}. \end{aligned} \tag{5.37}$$

Define

$$b_3 = b_c b_w \sup_{k \in [0,T]} a^{-\lambda k} \sum_{j=0}^{k-1} a^{k-1-j}.$$

According to (5.32), it follows that

$$\begin{aligned}\| e_i(k) \|_\lambda \;\le\;& b_c b_{\chi_0} a \\ & + b_c b_G \| \Delta u_{i-1}(k) \|_\lambda \sup_{k\in[0,T]} a^{-(N+1)} a^{-(\lambda-1)k} \left(\frac{a^{(\lambda-1)(k-N)-1}}{a^{\lambda-1}-1} \right) \\ & + b_3 \\ \le\;& b_c b_{\chi_0} a \\ & + b_c b_G a^{-1} \| \Delta u_{i-1}(k) \|_\lambda \left(\frac{a^{-\lambda N} - a^{-(\lambda-1)T-N}}{a^{\lambda-1}-1} \right) \\ & + b_3. \end{aligned} \tag{5.38}$$

Define $b_4 = b_c b_{\chi_0} a + b_3$. Taking the limit as $i \to \infty$ and using (5.35), it can be obtained as

$$\lim_{i\to\infty} \| e_i(k+N) \|_\lambda = b_4 + b_c b_G a^{-1} \frac{b_1}{1-\bar{\rho}} \left(\frac{a^{-\lambda N} - a^{-(\lambda-1)T-N}}{a^{\lambda-1}-1} \right). \tag{5.39}$$

The output error is bounded as $i \to \infty$. The proof is completed.

5.4 Simulation Examples

In this section, simulation examples are given to demonstrate the effectiveness and performance of the proposed algorithm in comparison with the usual RC. Consider the following second-order system with time-delay

$$G(s) = \frac{25}{(s+5)^2} e^{-2s}. \tag{5.40}$$

In the simulation study, the reference signal is chosen to be a sinusoidal signal with a period of $4s$, as shown in Figure 5.3.

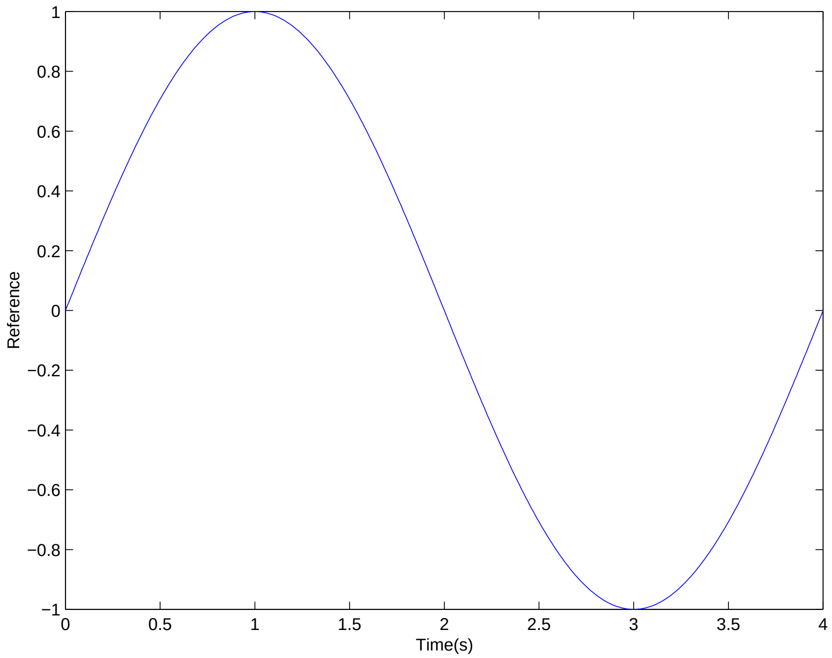


Figure 5.3: Reference signal

5.4.1 Usual RC

When the usual RC algorithm is applied, a divergent trend may be observed, even when a small learning gain $\gamma = 0.05$ is chosen. The divergent performance is shown in Figure 5.4. Additionally, Figure 5.5 also reveals the divergent phenomenon under the usual RC which can be attributed to the input time-delay present in the system.

5.4.2 New RC

The new proposed RC scheme is applied to the same system by first assuming the delay is known exactly. Tracking convergent performance can be achieved over 30 cycles with a learning gain of $\gamma = 0.4$, as shown in Figure 5.6. e_{MAX} (maximum absolute error) is reduced from 1.83 in first cycle to 0.021 in the 30th cycle, while e_{RMS} (root-mean-square error) is reduced from 1.05 to 0.014 correspondingly. This gain of $\gamma = 0.4$ is

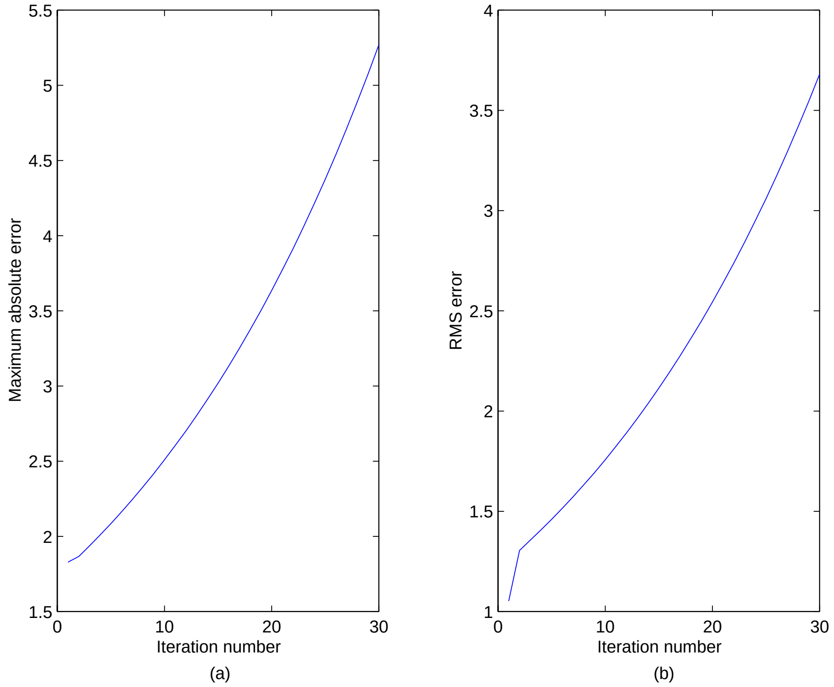


Figure 5.4: Divergent tracking performance under the usual RC

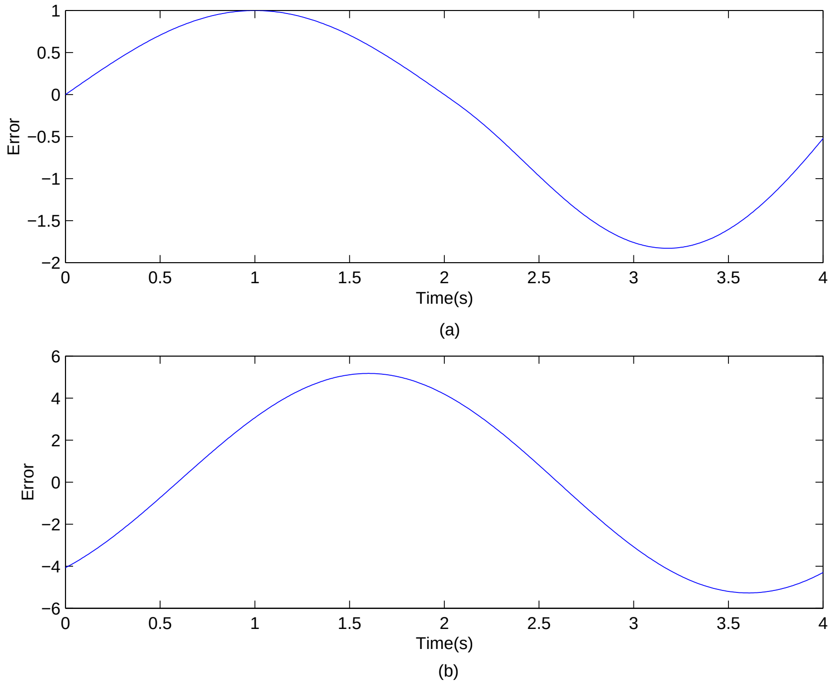


Figure 5.5: Tracking performance comparison under the usual RC (a). error in the first cycle (b). error in the 30th cycle

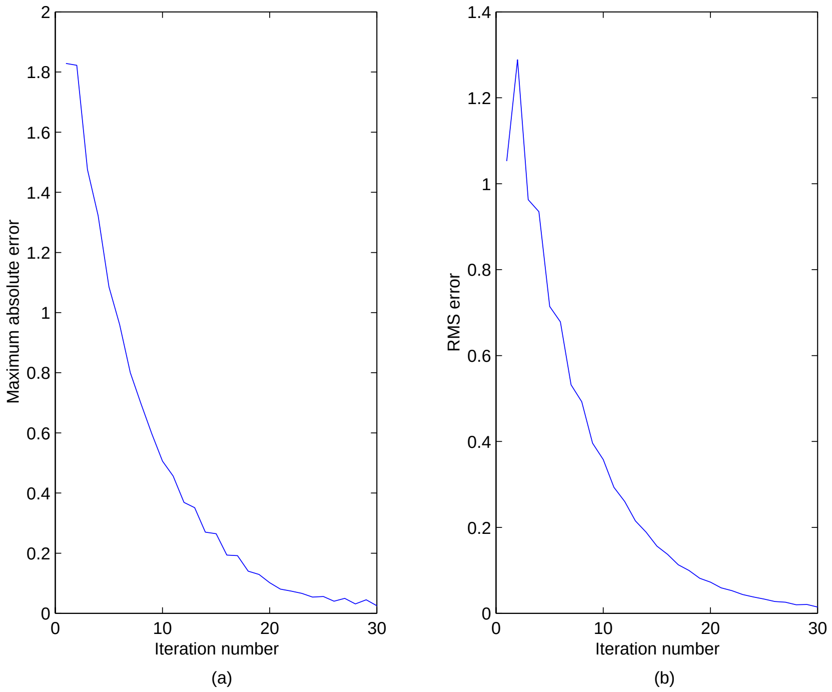
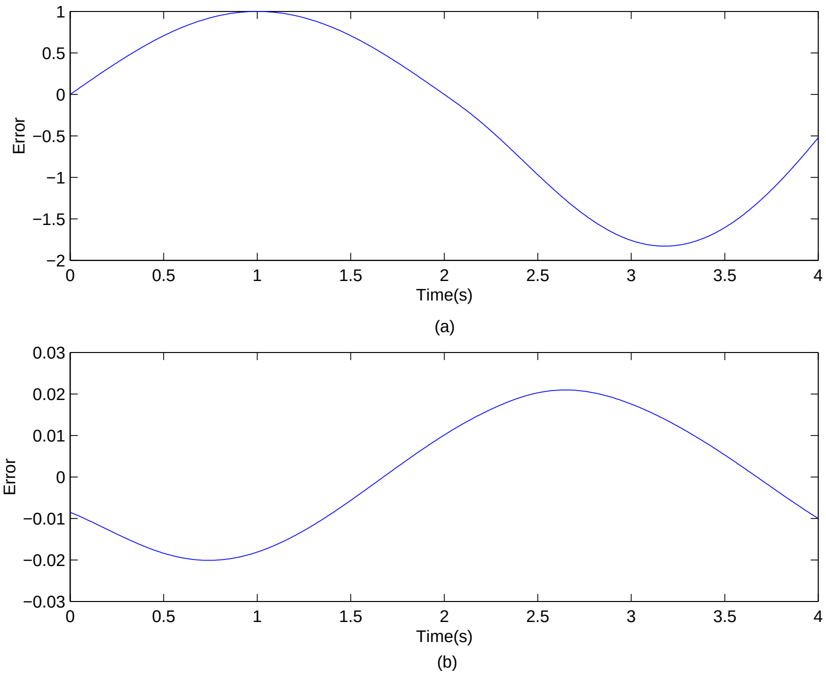


Figure 5.6: Convergent tracking performance under the new proposed RC

eight times the gain of $\gamma = 0.05$ used for the earlier simulation with the usual RC, yet error convergence is attainable. The initial blip appearing in Figure 5.6(b) reflects the transient period for initialization of the iterative learning to complete. Figure 5.7 shows the comparison of the tracking errors in the first cycle and 30th cycle. From the figures, it is clear that the new proposed RC algorithm can yield a satisfactory tracking performance for the system with time-delay.

5.4.3 Robust Performance

Finally, in order to verify the robustness of the proposed method, the simulation was conducted by deliberately including measurement noise and a modelling error in the time-delay. The time-delay is chosen as $2.8s$ in the simulation, representing a 40% modelling error. Figure 5.8 shows the convergent tracking performance over 30 cycles



(a)

(b)

Figure 5.7: Tracking performance improvement with the new proposed RC (a). error in the first cycle (b). error in the 30th cycle

with the same learning gain of $\gamma = 0.4$. e_{MAX} (maximum absolute error) is reduced from 1.83 in first cycle to 0.155 in the 30th cycle. Correspondingly, e_{RMS} (root-mean-square error) is also reduced from 1.05 to 0.054. The new RC has shown a satisfactory resilience to the effects of disturbance and modelling error.

Figure 5.9 shows the comparison of the errors in the first cycle and 30th cycle. From the figures, it is clear that the new proposed RC algorithm is robust to these uncertainties.

5.5 Conclusions

In this chapter, a new form of repetitive learning control has been proposed which is applicable to an extended class of systems, including time-delay systems. A new convergence condition which is necessary and sufficient has been derived for this new

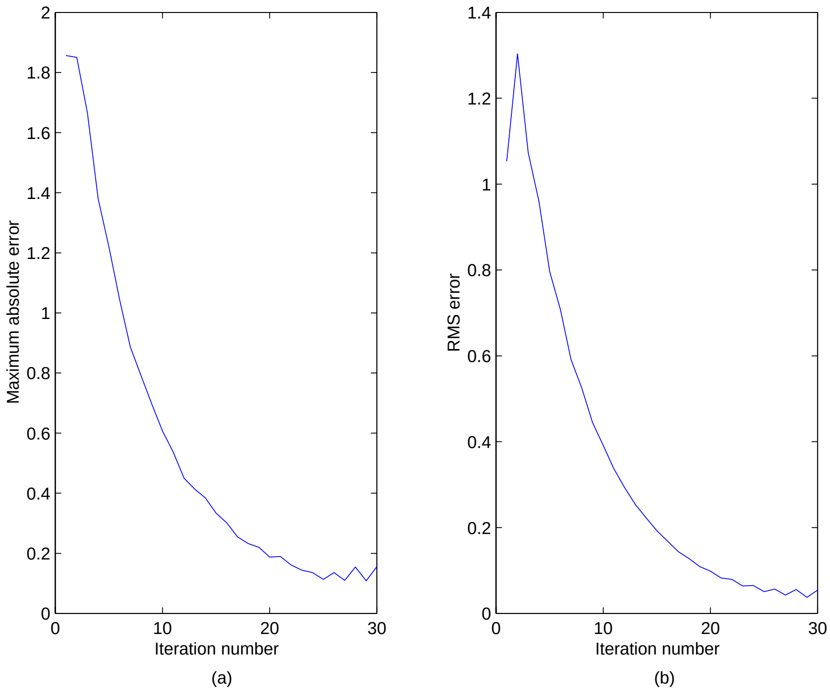


Figure 5.8: Convergent tracking performance with the system experiencing disturbances and modelling error

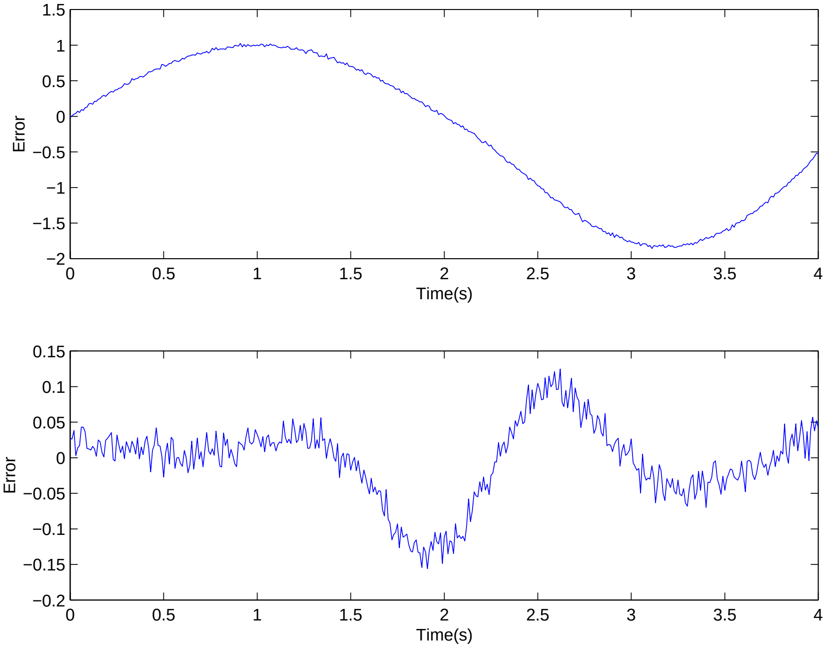


Figure 5.9: Tracking performance comparison with the system experiencing disturbances and modelling error (a). error in the first cycle (b). error in the 30th cycle

RC. In addition, a robust convergence analysis for the RC under the existence of a time-delay mismatch, initialzation errors, disturbances and measurement noise has shown the robustness of the new proposed approach. Simulation examples have also verified viable practical applications of the new RC to motion control problems.

Chapter 6

Predictive and Iterative Learning Control Algorithm

6.1 Introduction

Since the ILC method was proposed in 1984, many of the ILC algorithms are based on generic structures which do not explicitly contain the system models [99] [100][56]. The "model-free" approach, however, possesses limitations in terms of achievable performance and tuning guidelines, especially in multivariable control problems [45]. To overcome this limitation, model-based ILC algorithms have been proposed in [101][102][89][103]. With a model-based approach, estimates of future signals become available through prediction, and predictive ILC control has been suggested to achieve better performance and exhibit better convergence properties as compared to the basic ILC algorithms with inevitably less 'foresight' [104] [105]. In [104], an iterative learning algorithm with predictive control is developed, in which the predictor is designed in the time domain. For a repetitive process, however, it is often more interesting of the predictor design which is based on the trial number instead of continuous time. In [105], an optimal predictive ILC controller for linear systems is presented, in which the predictor is constructed based on the trial number (or repetitive number). The full state feedback is introduced in their proposed algorithm. However, in most applications, many state variables are not directly measurable and must be estimated [91]. In [106], an optimal ILC algorithm is proposed for non-minimum phase systems with the interpretation of frequency domain. In addition, as indicated in the conclusion of [105], the robustness issue has not yet been adequately addressed in their results.

Based on the above considerations, this chapter is dedicated to develop a novel predictive iterative learning algorithm for time-varying, linear and repetitive systems. This algorithm is simpler compared to that of [105], while keeping the basic predictive control features. An error model that represents the transition of the tracking error between two successive trials is first introduced. Based on this model, a predictive and iterative learning algorithm is derived which is only based on the trial number (or repetition index). The convergence properties of this algorithm is investigated rigorously in this chapter. In addition, the robustness of the learning system against the modeling errors, initial errors, and the presence of disturbances are derived by using a sup-norm approach rather than the traditional $\lambda-$norm [107] towards convergence analysis. An example is given to demonstrate the effectiveness of the proposed algorithm.

6.2 Problem Formulation

Consider a repetitive linear discrete-time system with uncertainty and disturbance as follows

$$
\begin{aligned}
x_i(t+1) &= A(t)x_i(t) + B(t)u_i(t) + w_i(t), &\quad (6.1)\\
y_i(t) &= C(t)x_i(t), &\quad (6.2)
\end{aligned}
$$

where i denotes the ith repetitive operation of the system; $x_i(t) \in R^n, u_i(t) \in R^r$, and $y_i(t) \in R^m$ are the state, control input, and output of the system, respectively; $w_i(t)$ is uncertainty or disturbance; $t \in [0, N]$ represents time; and $A(t)$, $B(t)$, and $C(t)$ are matrices with appropriate dimensions.

The problem is stated as follows: Find an update mechanism for the input trajectory of a new repetition based on the information from the previous repetitive operation so that the controlled output converges to the desired reference over the time horizon $[0, N]$.

Due to the cyclic nature of the repetitive processes, it is convenient to pack the information in each cycle (or trial) together. The equation (6.1) becomes

$$
\begin{aligned}
x_i(1) &= A(0)x_i(0) + B(0)u_i(0) + w_i(0),\\
x_i(2) &= A(1)A(0)x_i(0) + A(1)B(0)u_i(0) + B(1)u_i(1) + A(1)w_i(0) + w_i(1),\\
&\vdots\\
x_i(N) &= \Pi_{k=0}^{N-1}A(k)x_i(0) + \Pi_{k=1}^{N-1}A(k)B(0)u_i(0) + ... + B(N-1)u_i(N-1)\\
&\quad +\Pi_{k=1}^{N-1}A(k)w_i(0) + ... + w_i(N).
\end{aligned}
$$

Packing the results in a matrix form for the time steps within a repetition, it can be obtained as

$$\mathbf{y}_i = G\mathbf{u}_i + G_0 x_i(0) + \mathbf{w}_i, \tag{6.3}$$

where $\mathbf{y}_i = [y_i^T(1), y_i^T(2), ..., y_i^T(N)]^T$,

$$\begin{aligned}
G &= \begin{bmatrix} C(1)B(0) & 0 & ... & 0 \\ C(2)A(1)B(0) & C(2)B(1) & ... & 0 \\ & \vdots & ... & \\ C(N)\Pi_{k=1}^{N-1}A(k)B(0) & C(N)\Pi_{k=2}^{N-1}A(k)B(1) & ... & C(N)B(N-1) \end{bmatrix}, \\
G_0 &= \begin{bmatrix} C(1)A(0) \\ C(2)A(1)A(0) \\ \vdots \\ C(N)\Pi_{k=0}^{N-1}A(k) \end{bmatrix}, \\
\mathbf{w}_i &= \begin{bmatrix} C(1)w_i(0) \\ C(2)A(1)w_i(0) + w_i(1) \\ \vdots \\ C(N)\Pi_{k=1}^{N-1}A(k)w_i(0) + ... + C(N)w_i(0) \end{bmatrix}, \\
\mathbf{u}_i &= [u_i^T(0), u_i^T(1), ..., u_i^T(N-1)]^T.
\end{aligned}$$

In the above, i denotes the ith repetitive operation of the system and $\mathbf{w}_i$ represents the batch-wise independent error (including measurement noises). G is an impulse response matrix which can be derived through identification or linearization of a nonlinear model.

In the framework to be developed, the following norms are used.

$$\parallel f \parallel \;=\; \max_{1\le i\le n} |f_i|, \;\parallel S \parallel = \max_{1\le i\le m} (\sum_{j=1}^{n} |s_{ij}|),$$

where $f = [f_1, ..., f_n]^T$ is a vector, and $S = [s_{ij}] \in R^{m\times n}$ is a matrix.

6.3 Predictive and Iterative Learning Control Algorithm

In this section, the formulation of the predictive and learning algorithm is described.

6.3.1 Predictor Construction

Once the state space model is available, the subsequent steps for predictor construction is straightforward. The output error at the $(i+1)$th iteration can be written as:

$$\begin{aligned}\mathbf{e}_{i+1} &= \mathbf{y}_d - \mathbf{y}_{i+1} \\ &= \mathbf{e}_i - (\mathbf{y}_{i+1} - \mathbf{y}_i) \\ &= \mathbf{e}_i - G\Delta\mathbf{u}_{i+1} - G_0\Delta x_{i+1}(0) - \Delta\mathbf{w}_{i+1}. \qquad (6.4)\end{aligned}$$

The structure of error model (6.4) can be useful in formulating predictive controllers. In order to do this, a prediction model may be defined as follows:

$$\hat{\mathbf{e}}_{i+1} = \hat{\mathbf{e}}_i - G\Delta\mathbf{u}_{i+1}, \qquad (6.5)$$

where $\hat{\mathbf{e}}_{i+1}$ denotes the error predicted at instant i for instant $i+1$. This model is redefined at each sampling instant i from the actual error vector previously applied, that is $\hat{\mathbf{e}}_i = \mathbf{e}_i$. Comparing (6.5) with (6.4), one may observe that the disturbance noise vector is not included since it is assumed to be unknown. In addition, $\Delta x_{i+1}(0)$ is omitted as it may complicate the control formulation. Applying the equation (6.5) recursively, it follows that

$$\begin{aligned}
\hat{\mathbf{e}}_{i+2} &= \mathbf{e}_i - G\Delta\mathbf{u}_{i+2} - G\Delta\mathbf{u}_{i+1}, && (6.6)\\
\hat{\mathbf{e}}_{i+3} &= \mathbf{e}_i - G(\Delta\mathbf{u}_{i+3} + \Delta\mathbf{u}_{i+2} + \Delta\mathbf{u}_{i+1}), && (6.7)\\
&\vdots \\
\hat{\mathbf{e}}_{i+h} &= \mathbf{e}_i - G(\Delta\mathbf{u}_{i+h} + \Delta\mathbf{u}_{i+h-1} + ... + \Delta\mathbf{u}_{i+1}), && (6.8)
\end{aligned}$$

where h is the prediction horizon.

6.3.2 Derivation of Algorithm

A linear quadratic performance index is considered which may be written in the following form:

$$J = \sum_{j=1}^{h} \left[\hat{\mathbf{e}}_{i+j}^T \hat{\mathbf{e}}_{i+j} + \bar{\gamma} \Delta\mathbf{u}_{i+j}^T \Delta\mathbf{u}_{i+j} \right], \qquad (6.9)$$

where $\bar{\gamma} > 0$ is a control weight and h is a prediction horizon. Note that this criterion includes not only the error associated with the next trial, but also those associated with the next h trials, as well as the corresponding changes in input. The weight parameter $\bar{\gamma} > 0$ determines the relative importance of more distant (future) errors and incremental inputs compared with the present ones. By including more distant signals, the learning algorithm becomes less ‘short sighted’ [105].

The complexity introduced by the equations (6.5)-(6.8) is basically the result of the number of unknown control sequence Δu_{i+j}. One way of reducing the number of unknowns is to predetermine the form of the control sequence. It has been proven useful to impose a step control sequence together with a cost function such as the one given in the equation (6.9), thus reducing the number of unknowns to a single one [108][109]. The control sequence can be imposed to be constant over the prediction interval, i.e., $\Delta\mathbf{u}_{i+1} = \Delta\mathbf{u}_{i+2} = ... = \Delta\mathbf{u}_{i+h}$. Now, the prediction equations over the $i+h$ horizon can be obtained.

$$\begin{aligned}
\hat{\mathbf{e}}_{i+1} &= \mathbf{e}_i - G\Delta\mathbf{u}_{i+1}, && (6.10)\\
\hat{\mathbf{e}}_{i+2} &= \mathbf{e}_i - 2G\Delta\mathbf{u}_{i+1}, && (6.11)\\
\hat{\mathbf{e}}_{i+3} &= \mathbf{e}_i - 3G\Delta\mathbf{u}_{i+1}, && (6.12)\\
&\vdots \\
\hat{\mathbf{e}}_{i+h} &= \mathbf{e}_i - hG\Delta\mathbf{u}_{i+1}. && (6.13)
\end{aligned}$$

The optimization problem becomes

$$J = \sum_{j=1}^{h} \hat{\mathbf{e}}_{i+j}^T \hat{\mathbf{e}}_{i+j} + \gamma\Delta\mathbf{u}_{i+1}^T \Delta\mathbf{u}_{i+1}, \tag{6.14}$$

where $\gamma = h\bar{\gamma}$. Substituting (6.10)-(6.13) into (6.14) yields

$$\begin{aligned}
J &= (\hat{\mathbf{e}}_{i+1}, \hat{\mathbf{e}}_{i+2}, ..., \hat{\mathbf{e}}_{i+h}) \begin{pmatrix} \hat{\mathbf{e}}_{i+1} \\ \hat{\mathbf{e}}_{i+2} \\ \vdots \\ \hat{\mathbf{e}}_{i+h} \end{pmatrix} + \gamma\Delta\mathbf{u}_{i+1}^T \Delta\mathbf{u}_{i+1} \\
&= [\begin{pmatrix} I_{Nm} \\ I_{Nm} \\ \vdots \\ I_{Nm} \end{pmatrix} \mathbf{e}_i - \begin{pmatrix} I_{Nm} \\ 2I_{Nm} \\ \vdots \\ hI_{Nm} \end{pmatrix} G\Delta\mathbf{u}_{i+1}]^T [\begin{pmatrix} I_{Nm} \\ I_{Nm} \\ \vdots \\ I_{Nm} \end{pmatrix} \mathbf{e}_i - \begin{pmatrix} I_{Nm} \\ 2I_{Nm} \\ \vdots \\ hI_{Nm} \end{pmatrix} G\Delta\mathbf{u}_{i+1}] \\
&+ \gamma\Delta\mathbf{u}_{i+1}^T I_{Nr} \Delta\mathbf{u}_{i+1},
\end{aligned}$$

where I_{Nm} and I_{Nr} are the $Nm \times Nm$ and $Nr \times Nr$ unit matrices, respectively. Imposing the condition on the gradient, $\frac{\partial J}{\partial \Delta \mathbf{u}_{i+1}} = 0$, the control action can be derived as:

$$\Delta \mathbf{u}_{i+1} = (G^T F_1^T F_1 G + \gamma I_{Nr})^{-1} G^T F_1^T F_2 \mathbf{e}_i, \tag{6.15}$$

where $F_1 = [I_{Nm}, 2I_{Nm}, ..., hI_{Nm}]^T$ and $F_2 = [I_{Nm}, I_{Nm}, ..., I_{Nm}]^T$. In this equation, it is noted that

$$\begin{aligned} F_1^T F_1 &= [I_{Nm}, 2I_{Nm}, ..., hI_{Nm}] \begin{bmatrix} I_{Nm} \\ 2I_{Nm} \\ \vdots \\ hI_{Nm} \end{bmatrix} && (6.16) \\ &= (1 + 2^2 + ... + h^2) I_{Nm} = aI_{Nm}, && (6.17) \end{aligned}$$

where $a = \frac{1}{6}h(h+1)(2h+1)$. It is also noted that

$$\begin{aligned} F_1^T F_2 &= [I_{Nm}, 2I_{Nm}, ..., hI_{Nm}] \begin{bmatrix} I_{Nm} \\ I_{Nm} \\ \vdots \\ I_{Nm} \end{bmatrix} && (6.18) \\ &= (1 + 2 + ... + h) I_{Nm} = bI_{Nm}, && (6.19) \end{aligned}$$

where $b = \frac{1}{2}h(1+h)$. Thus, (6.15) becomes

$$\mathbf{u}_{i+1} = \mathbf{u}_i + b(aG^T G + \gamma I_{Nr})^{-1} G^T \mathbf{e}_i, \tag{6.20}$$

where $\gamma > 0$.

Remark 6.1. The proposed ILC scheme has a feedforward structure, and the ith current input is generated by the earlier data at the $i-1$th trial. One advantage of this algorithm is that it takes the same dimensionality of the matrix as the non-predictive ILC with a quadratic criterion [91], thus achieving a prediction capability while keeping the basic predictive control features.

Remark 6.2. When the system disturbances and noises present become significant, a similar observer algorithm as that of Lee et al. [91], may be considered:

$$\mathbf{u}_{i+1} = \mathbf{u}_i + b(aG^TG + \gamma I_{Nr})^{-1}G^T\bar{\mathbf{e}}_{i/i}, \tag{6.21}$$

where $\bar{\mathbf{e}}_{i/i}$ is the estimate of $\mathbf{e}_i$. The following observer is used for obtaining $\bar{\mathbf{e}}_{i/i}$:

$$\bar{\mathbf{e}}_{i/i-1} = \bar{\mathbf{e}}_{i-1/i-1} - G\Delta\mathbf{u}_i, \tag{6.22}$$

$$\bar{\mathbf{e}}_{i/i} = \bar{\mathbf{e}}_{i/i-1} + K(\mathbf{e}_i - \bar{\mathbf{e}}_{i/i-1}), \tag{6.23}$$

where K is the filter gain matrix which can be obtained through various means, such as the pole placement and Kalman filtering techniques.

6.3.3 Convergence and Robustness of Algorithm

For the proposed control laws, the following convergence properties can be established.

Theorem 6.1. Consider the system (6.4) under the assumptions that $\mathbf{w}_{i+1} - \mathbf{w}_i = 0$, $x_{i+1}(0) - x_i(0) = 0$ and G is full row rank. Given the desired trajectory $\mathbf{y}_d$ over the fixed time interval $[0, N]$, by using the learning control law (6.20), the tracking error converges to zero for $h \geq 1$ as $i \to \infty$.

Proof: To analyze the convergence, the closed-loop system could be derived. Substituting (6.20) into (6.4) produces a closed-loop

$$\begin{aligned}\mathbf{e}_{i+1} &= [I_{Nm} - bG(aG^TG + \gamma I_{Nr})^{-1}G^T]\mathbf{e}_i \\ &= [I_{Nm} - \frac{b}{a}G(G^TG + \frac{\gamma}{a}I_{Nr})^{-1}G^T]\mathbf{e}_i.\end{aligned}$$

Let

$$E = I_{Nm} - \frac{b}{a}G(G^TG + \frac{\gamma}{a}I_{Nr})^{-1}G^T. \tag{6.24}$$

Applying the matrix inversion lemma ([110]) leads to

$$\begin{aligned}(G^TG + \frac{\gamma}{a}I_{Nr})^{-1} &= \frac{a}{\gamma}I_{Nr} - \frac{a^2}{\gamma^2}G^T(\frac{a}{\gamma}GG^T + I_{Nm})^{-1}G \\ &= \frac{a}{\gamma}[I_{Nr} - G^T(GG^T + \frac{\gamma}{a}I_{Nm})^{-1}G].\end{aligned}$$

Applying this formula to (6.24) yields:

$$\begin{aligned}E &= I_{Nm} - \frac{b}{a}G(G^TG + \frac{\gamma}{a}I_{Nr})^{-1}G^T \\ &= I_{Nm} - \frac{b}{\gamma}[GG^T - GG^T(GG^T + \frac{\gamma}{a}I_{Nm})^{-1}GG^T] \\ &= I_{Nm} - \frac{b}{\gamma}[GG^T(GG^T + \frac{\gamma}{a}I_{Nm})^{-1}(GG^T + \frac{\gamma}{a}I_{Nm}) - GG^T(GG^T + \frac{\gamma}{a})^{-1}GG^T] \\ &= I_{Nm} - \frac{b}{a}GG^T(GG^T + \frac{\gamma}{a}I_{Nm})^{-1}.\end{aligned}$$

Since G is full row rank, GG^T is positive definite and non-singular (can be obtained directly from the Singular Value Decomposition (SVD)). Thus, it follows that

$$\begin{aligned}E &= I_{Nm} - \frac{b}{a}[(GG^T + \frac{\gamma}{a}I_{Nm})(GG^T)^{-1}]^{-1} \\ &= I_{Nm} - \frac{b}{a}[I_{Nm} + \frac{\gamma}{a}(GG^T)^{-1}]^{-1}.\end{aligned}$$

For the convergence analysis, it is necessary to know the eigenvalues of E.

$$\begin{aligned}\lambda_i(E) &= \lambda_i\{I_{Nm} - \frac{b}{a}[I_{Nm} + \frac{\gamma}{a}(GG^T)^{-1}]^{-1}\} = 1 - \frac{b}{a}\lambda_i\{[I_{Nm} + \frac{\gamma}{a}(GG^T)^{-1}]^{-1}\} \\ &= 1 - \frac{b}{a}\frac{1}{\{1 + \frac{\gamma}{a}\lambda_i[(GG^T)^{-1}]\}}.\end{aligned}$$

Since GG^T is positive definite and non-singular, this implies that $\lambda_i[(GG^T)^{-1}] > 0$. This implies that $1 + \frac{\gamma}{a}\lambda_i[(GG^T)^{-1}] > 1$ since $\gamma > 0, a > 0$. Further, this implies that $0 < \frac{1}{1+\frac{\gamma}{a}\lambda_i[(GG^T)^{-1}]} < 1$. Also, since $0 < \frac{b}{a} = \frac{3}{2h+1} \leq 1$ for the prediction horizon $h \geq 1$, this together with $0 < \frac{1}{1+\frac{\gamma}{a}\lambda_i[(GG^T)^{-1}]} < 1$ yields

$$0 < \frac{b}{a}\frac{1}{\{1 + \frac{\gamma}{a}\lambda_i[(GG^T)^{-1}]\}} < 1. \tag{6.25}$$

Notice that E is a constant matrix. This implies that $|\lambda_i(E)| < 1$. The conclusion follows.

Another advantage of the proposed ILC is the availability of tuning parameters like the input weighting matrix γ and prediction horizon h which can be used to enhance the robustness against model uncertainty. This feature can be shown by considering a case where G contains uncertainties.

The error evolution equation for the true system is written as:

$$\mathbf{e}_{i+1} = \mathbf{e}_i - G^{true}\Delta\mathbf{u}_{i+1} = \mathbf{e}_{i+1} - (G + \Delta G)\Delta u_{i+1}, \tag{6.26}$$

where G is the nominal matrix, and ΔG is the perturbation matrix. Assume that

$$\parallel \Delta G \parallel \leq \varphi, \tag{6.27}$$

where φ is a constant. The upper and lower bounds on φ are to be found such that if φ is within these bounds, the error convergence still holds.

Since E is an asymptotically stable matrix as shown in the proof of Theorem 6.1, the following Lyapunov equation holds:

$$E^T P E - P = -I_{Nm}, \tag{6.28}$$

where P is a positive definite matrix. This equation is used in the proof of the following theorem.

Theorem 6.2. Consider the system (6.26) under the assumptions that $\mathbf{w}_{i+1} - \mathbf{w}_i = 0$, $x_{i+1}(0) - x_i(0) = 0$ and the nominal matrix G is full row rank. Given the desired trajectory $\mathbf{y}_d$ over the fixed time interval $[0, N]$, by using the learning control law (6.20), the tracking error converges to zero for $h \geq 1$ as $i \to \infty$, if

$$\frac{-c - \sqrt{c^2 + d}}{d} < \varphi < \frac{-c + \sqrt{c^2 + d}}{d}, \tag{6.29}$$

where

$$\begin{aligned} c &= \parallel E^T P \parallel \parallel (aG^T G + \gamma I_{Nr})^{-1} G^T \parallel, & (6.30) \\ d &= \parallel (aG^T G + \gamma I_{Nr})^{-1} G^T \parallel^2 \parallel P \parallel . & (6.31) \end{aligned}$$

Proof: Substituting (6.20) into (6.4) gives

$$\begin{aligned}
\mathbf{e}_{i+1} &= \mathbf{e}_i - (G + \Delta G)\Delta\mathbf{u}_{i+1} \\
&= [I_{Nm} - bG(aG^TG + \gamma I_{Nr})^{-1}G^T]\mathbf{e}_i - b\Delta G(aG^TG + \gamma I_{Nr})^{-1}G^T\mathbf{e}_i \\
&= E\mathbf{e}_i - b\Delta G(aG^TG + \gamma I_{Nr})^{-1}G^T\mathbf{e}_i.
\end{aligned}$$

For simplicity, denote $H = (aG^TG + \gamma I_{Nr})^{-1}G^T$. Define the Lyapunov function $V_{i+1} = \mathbf{e}_{i+1}^T P\mathbf{e}_{i+1}$ and it follows that

$$\begin{aligned}
\Delta V_{i+1} &= V_{i+1} - V_i = \mathbf{e}_{i+1}^T P\mathbf{e}_{i+1} - \mathbf{e}_i^T P\mathbf{e}_i \\
&= (E\mathbf{e}_i - \Delta GHe_i)^T P(E\mathbf{e}_i - \Delta GH\mathbf{e}_i) - \mathbf{e}_i^T P\mathbf{e}_i \\
&= \mathbf{e}_i^T(E^TPE - P)\mathbf{e}_i - 2\mathbf{e}_i^T E^T P\Delta GH\mathbf{e}_i + \mathbf{e}_i^T H^T\Delta G^T P\Delta GH\mathbf{e}_i \\
&\leq -||\mathbf{e}_i||^2 + 2\varphi \parallel E^TP \parallel\parallel H \parallel\parallel \mathbf{e}_i \parallel^2 + \varphi^2 \parallel H \parallel^2\parallel P \parallel\parallel \mathbf{e}_i \parallel^2 . \qquad (6.32)
\end{aligned}$$

From the definitions of (6.30) and (6.31), it can be obtained that

$$\begin{aligned}
\Delta V_{i+1} &\leq (-1 + 2c\varphi + d\varphi^2)||\mathbf{e}_i||^2 \\
&= d(\varphi - \frac{-c + \sqrt{c^2 + d}}{d})(\varphi - \frac{-c - \sqrt{c^2 + d}}{d})||\mathbf{e}_i||^2. \qquad (6.33)
\end{aligned}$$

It is obvious that if $\frac{-c-\sqrt{c^2+d}}{d} < \varphi < \frac{-c+\sqrt{c^2+d}}{d}$, then $d\varphi^2 + 2c\varphi - 1 < 0$. This together with $d > 0$, implies that the convergence is achieved. The proof is completed.

In Theorem 6.1 and Theorem 6.2, the convergence properties are established without considering the measurement noises, disturbances, and initialization errors. In practical applications, the robustness of ILC algorithms against these uncertainties is an important issue which should be addressed.

For ease of presentation, the following notations for abbreviation purposes are defined:

$$\begin{aligned}\lambda_{G_0} &= \lambda_{max}(G_0^T P G_0), \;\; \lambda_P = \lambda_{max}(P), \;\; P_{G_0} = ||G_0^T P||, \\ \lambda_{G_0E} &= \lambda_{max}[(G_0^T PE)(E^T PG_0)], \; \lambda_E = \lambda_{max}[(PE)(E^T P)].\end{aligned}$$

Theorem 6.3. Consider the system (6.4) under the assumptions that $||\mathbf{w}_i - \mathbf{w}_{i-1}|| \leq b_w$, $||x_i(0) - x_{i-1}(0)|| \leq b_{x0}$ and G is full row rank. Given the desired trajectory $\mathbf{y}_d$ over the fixed time interval $[0, N]$, by using the learning control law (6.20), the tracking error converges to the following bound for $h \geq 1$ as $i \to \infty$,

$$\lim_{i\to\infty} ||\mathbf{e}_i|| \;\; \leq \;\; \sqrt{\frac{g(b_{x0}, b_w)}{(1-\rho)\lambda_{min}(P)}}, \tag{6.34}$$

where $g(b_{x0}, b_w)$ is constant proportional to constants b_{x0}, and b_w. Moreover, $\lim_{i\to\infty} ||\mathbf{e}_i|| = 0$ if $b_{x0} = b_w = 0$.

Proof: The error model is derived according to the current conditions.

$$\begin{aligned}\mathbf{e}_{i+1} &= [I_{Nm} - bG(aG^TG + \gamma I_{Nr})^{-1}G^T]\mathbf{e}_i - G_0\Delta x_{i+1}(0) - \Delta\mathbf{w}_{i+1} \\ &= E\mathbf{e}_i - G_0\Delta x_{i+1}(0) - \Delta\mathbf{w}_{i+1}.\end{aligned}$$

Similar to the proof in Theorem 6.2, the Lyapunov function can be selected as $V_i =$

$\mathbf{e}_i^T P \mathbf{e}_i$. Then, it can be shown that

$$\begin{aligned}
\Delta V_{i+1} &= V_{i+1} - V_i = \mathbf{e}_{i+1}^T P \mathbf{e}_{i+1} - \mathbf{e}_i^T P \mathbf{e}_i \\
&= [E\mathbf{e}_i - G_0 \Delta x_{i+1}(0) - \Delta \mathbf{w}_{i+1}]^T P [E\mathbf{e}_i - G_0 \Delta x_{i+1}(0) - \Delta \mathbf{w}_{i+1}] - \mathbf{e}_i^T P \mathbf{e}_i \\
&= \mathbf{e}_i^T (E^T P E - P)\mathbf{e}_i - 2\mathbf{e}_i^T E^T P G_0 \Delta x_{i+1}(0) - 2\mathbf{e}_i^T E^T P \Delta \mathbf{w}_{i+1} \\
&+ 2\Delta x_{i+1}^T(0) G_0^T P \Delta \mathbf{w}_{i+1} + \Delta x_{i+1}^T(0) G_0^T P G_0 \Delta x_{i+1}(0) + \Delta \mathbf{w}_{i+1}^T P \Delta \mathbf{w}_{i+1} \\
&= -\mathbf{e}_i^T \mathbf{e}_i - 2\mathbf{e}_i^T E^T P G_0 \Delta x_{i+1}(0) - 2\mathbf{e}_i^T E^T P \Delta \mathbf{w}_{i+1} + 2\Delta x_{i+1}^T(0) G_0^T P \Delta \mathbf{w}_{i+1} \\
&+ \Delta x_{i+1}^T(0) G_0^T P G_0 \Delta x_{i+1}(0) + \Delta \mathbf{w}_{i+1}^T P \Delta \mathbf{w}_{i+1}. \qquad (6.35)
\end{aligned}$$

Since $-2\alpha^T \beta \leq \eta \alpha^T \alpha + \frac{1}{\eta}\beta^T \beta$ where η is an arbitrarily positive constant, the following inequalities hold:

$$\begin{aligned}
-2\mathbf{e}_i^T E^T P G_0 \Delta x_{i+1}(0) &\leq \eta \mathbf{e}_i^T \mathbf{e}_i + \frac{1}{\eta} \Delta x_{i+1}^T(0)(G_0^T P E)(E^T P G_0)\Delta x_{i+1}(0) \\
&\leq \eta ||\mathbf{e}_i||^2 + \frac{1}{\eta} \lambda_{max}[(G_0^T P E)(E^T P G_0)] b_{x0}^2 \\
-2\mathbf{e}_i^T E^T P \Delta \mathbf{w}_{i+1} &\leq \eta ||\mathbf{e}_i||^2 + \frac{1}{\eta} \lambda_{max}[(PE)(E^T P)] b_w^2.
\end{aligned}$$

It is also noted that

$$\begin{aligned}
\Delta x_{i+1}^T(0) G_0^T P G_0 \Delta x_{i+1}(0) &\leq \lambda_{max}(G_0^T P G_0) b_{x0}^2 \\
\Delta \mathbf{w}_{i+1}^T P \Delta \mathbf{w}_{i+1} &\leq \lambda_{max}(P) b_w^2 \\
2\Delta x_{i+1}(0)^T G_0^T P \Delta \mathbf{w}_{i+1} &\leq 2||G_0^T P|| b_{x0} b_w.
\end{aligned}$$

According to the definition of the sup-norm, $\mathbf{e}_i^T \mathbf{e}_i \geq ||\mathbf{e}_i||^2$. This implies that $-\mathbf{e}_i^T \mathbf{e}_i \leq -||\mathbf{e}_i||^2$. Then, it follows

$$\Delta V_{i+1} \leq -(1 - 2\eta)||\mathbf{e}_i||^2 + g(b_{x0}, b_w), \qquad (6.36)$$

where $g(b_{x0}, b_w) = (\lambda_{G_0} + \frac{1}{\eta}\lambda_{G_0E})b_{x0}^2 + 2P_{G_0}b_{x0}b_w + (\lambda_P + \frac{1}{\eta}\lambda_E)b_w^2$.

From $\lambda_{min}(P)||\mathbf{e}_i||^2 \leq V_i \leq \lambda_{max}(P)||\mathbf{e}_i||^2$, it follows

$$\begin{aligned} V_{i+1} - V_i &\leq -\frac{1-2\eta}{\lambda_{max}(P)}V_i + g(b_{x0}, b_w) & (6.37) \\ V_{i+1} &\leq [1 - \frac{1-2\eta}{\lambda_{max}(P)}]V_i + g(b_{x0}, b_w) = \rho V_i + g(b_{x0}, b_w), & (6.38) \end{aligned}$$

where the value of η is chosen as

$$\frac{1-\lambda_{max}(P)}{2} < \eta < 1/2 \;\; if \; \lambda_{max}(P) < 1, \tag{6.39}$$

$$0 < \eta < 1/2 \;\; if \; \lambda_{max}(P) \geq 1, \tag{6.40}$$

such that $0 < \rho < 1$. Finally, the following inequailities can be established:

$$\begin{aligned} V_i &\leq \rho^i V_0 + \frac{1-\rho^i}{1-\rho}g(b_{x0}, b_w), & (6.41) \\ \lim_{i\to\infty} V_i &\leq \frac{g(b_{x0}, b_w)}{1-\rho}, & (6.42) \\ \lim_{i\to\infty} ||\mathbf{e}_i|| &\leq \sqrt{\frac{g(b_{x0}, b_w)}{(1-\rho).\lambda_{min}(P)}} & (6.43) \end{aligned}$$

Hence, $\lim_{i\to\infty} ||\mathbf{e}_i|| = 0$ if $b_{x0}, b_w \to 0$. The proof is completed.

Remark 6.3. This chapter provides a novel predictive ILC scheme. The algorithm is simple while maintaining the basic predictive control structure. The proposed learning control law (6.20) differs from the results of Lee and Lee [104] and Amann et al. [105]. The following differences are noted:

In [104], a model-based predictive iterative learning law is proposed based on a quadratic criterion, and the predictor is designed in the time domain.

$$\delta u_k(t) = (G^{mT} Q G^m + R)^{-1} G^{mT} Q e_k(t), \tag{6.44}$$

One difference between the algorithm of Lee and Lee [104] and the proposed algorithm is that the predictive learning control in the former works along the time domain, while the proposed predictor design is based on the trial index. It is clear that if the prediction is based on the trial number, the ILC control performance can be improved since the ILC is inherently a learning process based the trial number.

In [105], an optimal iterative learning algorithm which is also based on the trial number for linear time-invariant systems is given. The control scheme requires a full state feedback, and the solution of a Riccati equation and a recursive equation. However, in most applications, state variables are not directly measurable and must be estimated . In addition, solving a Riccati equation and a recursive equation results in additional computational burden for ILC control, especially for time-varying systems. Also, as indicated in the conclusion of Amann et al. [105], robustness has not been adequately addressed in their results. The proposed algorithm is able to circumvent the above-mentioned difficulties, i.e., no state estimate is necessary for the algorithm. This is the main difference from the result of Amman et al. [105]. Specifically, the robustness of the proposed algorithm against modeling errors, initial state errors, and the presence of disturbances, is also analyzed in details, showing the effects of various types of disturbances on the final error bound.

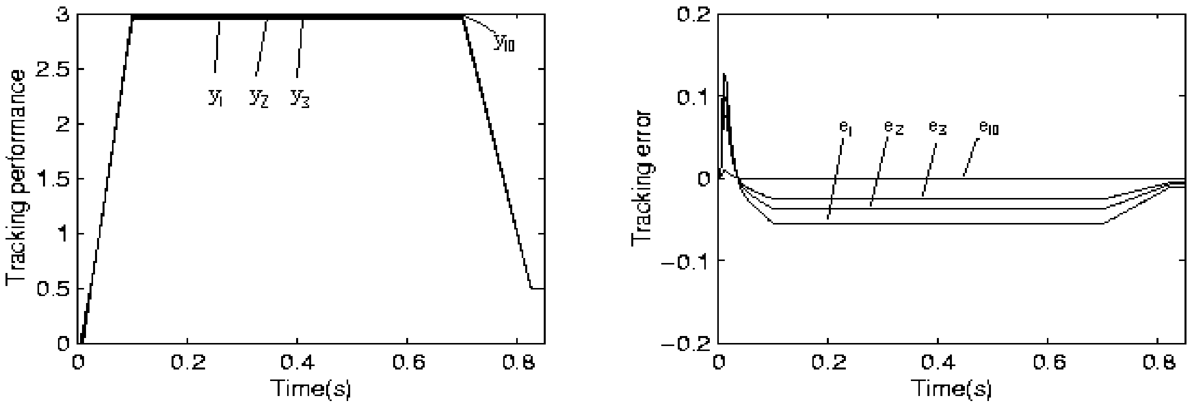


Figure 6.1: Tracking performance by the proposed controller: No uncertainty is considered.

6.4 Simulations

An injection molding problem for ram velocity control is considered [111]. The following equation describes the dynamics of the system:

$$G(s) = \frac{2.144 \times 10^{11}}{(s+125)(s+1138)[(s+383)^2+1135^2]}. \tag{6.45}$$

A discrete-system state equation in the time domain may be obtained directly from (6.45) with zero-order hold and the sampling time $T = 0.005s$. Since the injection molding is a cyclic process, it is attractive to use a learning controller. In this section, the proposed learning controller is used for the cycle-to-cycle control of the injection molding process. First, to illustrate the performance of the proposed controller, simulations are presented with the prediction horizon $h = 4$. In this case, the selection of $\gamma = 0.05$ is made. Figure 6.1 shows the control performances at the 1st, 2nd, 3rd, and 10th cycles. It is observed that the proposed predictive ILC scheme can yield a good set-point tracking performance as the cycle number increases.

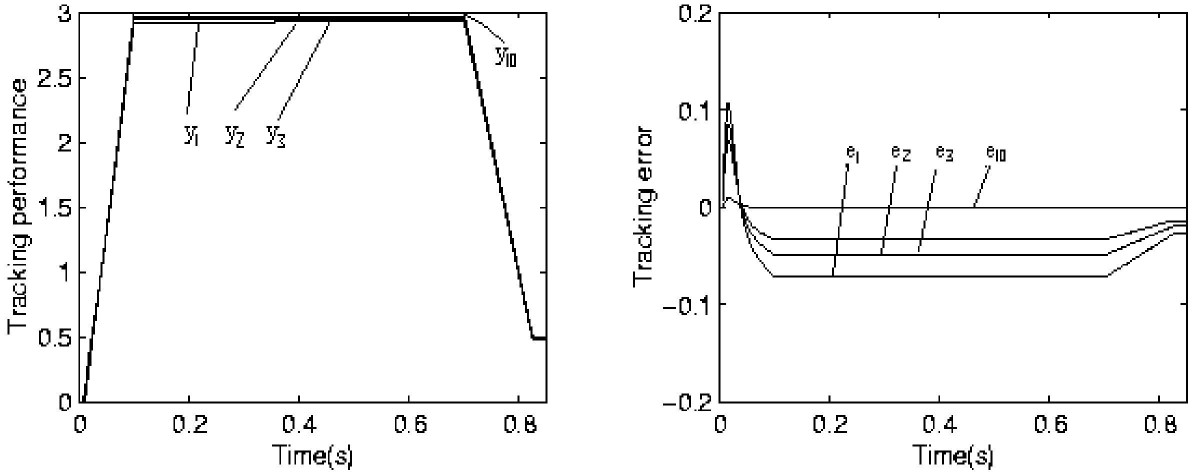


Figure 6.2: Tracking performance by the proposed controller: Modelling error is considered.

To test the robustness of the proposed algorithm, it is assumed that there is a modeling error in the identified and actual models. The learning controller is designed based on the nominal model derived from

$$G_m(s) = \frac{2 \times 10^{11}}{(s+100)(s+1100)[(s+383)^2+1135^2]} \tag{6.46}$$

which has a large model uncertainty compared to the actual model (6.45). Figure 6.2 shows the control result of the proposed predictive learning algorithm. Clearly, the control can achieve satisfactory tracking performance even under perturbation of the model parameters.

To further test the robustness of the proposed algorithm, measurement noise is introduced into the system. Now the uncertainties include both the modeling error and measurement noise. The control performance is shown in Figure 6.3. Additional disturbance is added to the system, a repetitive one given by $sin(0.0314t)$. Figure 6.4 shows the control performance. The error convergence and robustness of the control system

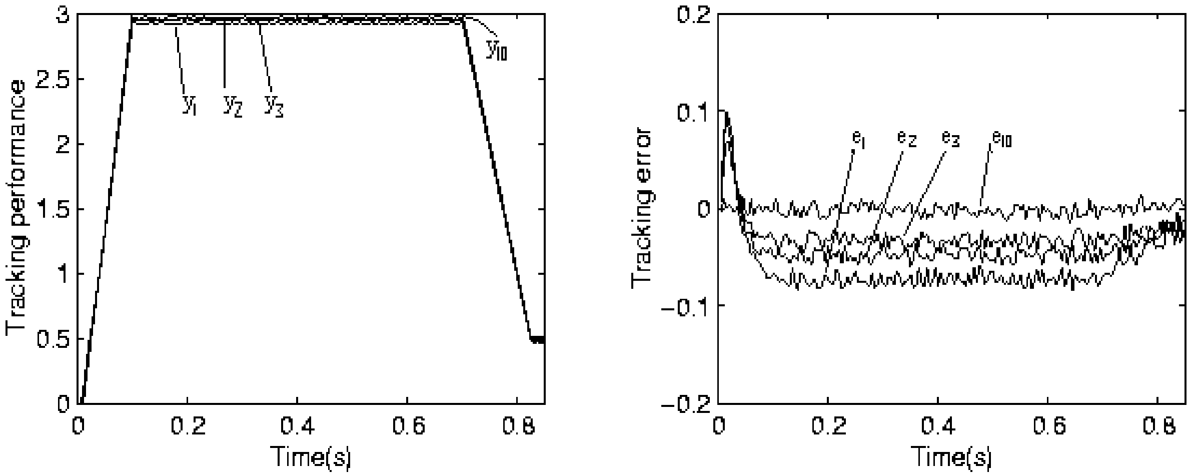


Figure 6.3: Tracking performance by the proposed controller: Modelling error and measurement noise are considered.

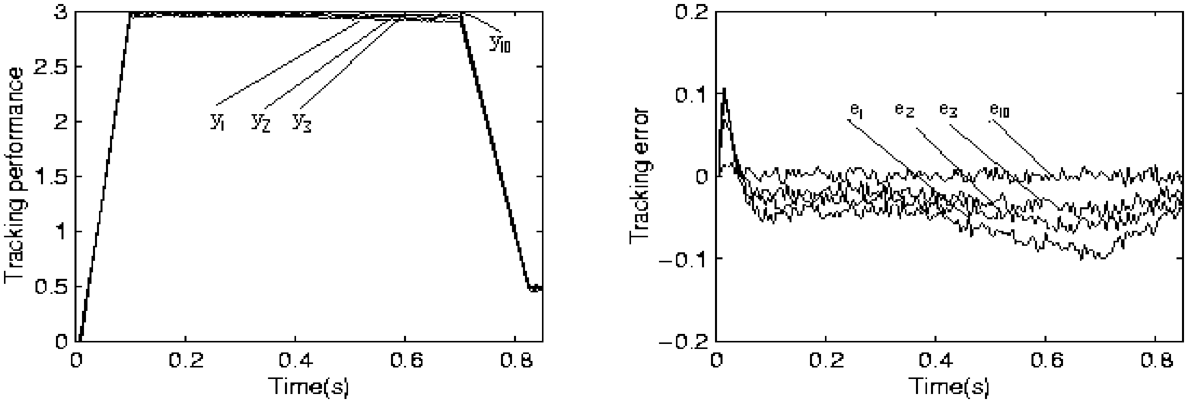


Figure 6.4: Tracking performance by the proposed controller: Modelling error, measurement noise and repetitive disturbance are considered.

are favorably verified.

To show the better performance of this scheme, a comparison with the pure ILC scheme without any predictive feature is done. Figure 6.5-6.7 shows the performance of the pure ILC scheme with a learning gain of 0.5 simulated under the same scenarios as the predictive-ILC scheme. A better performance from the proposed scheme is observed in all the comparisons.

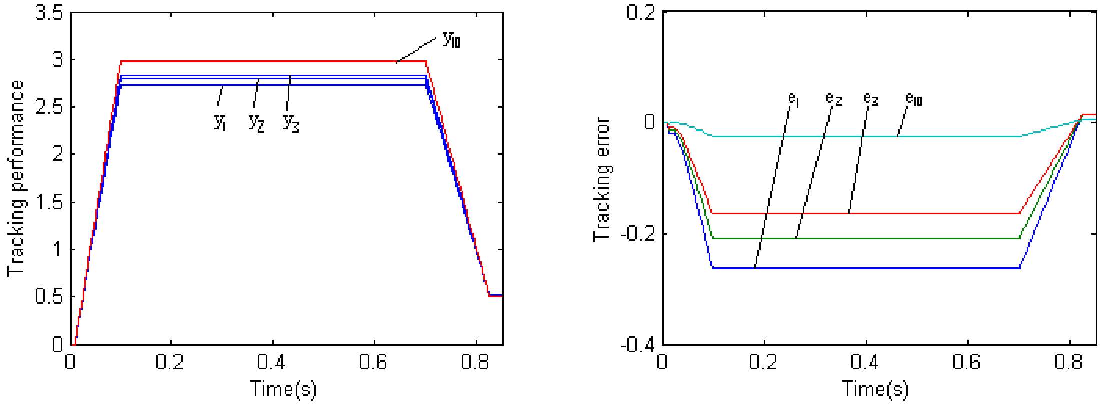


Figure 6.5: Tracking performance by a pure ILC

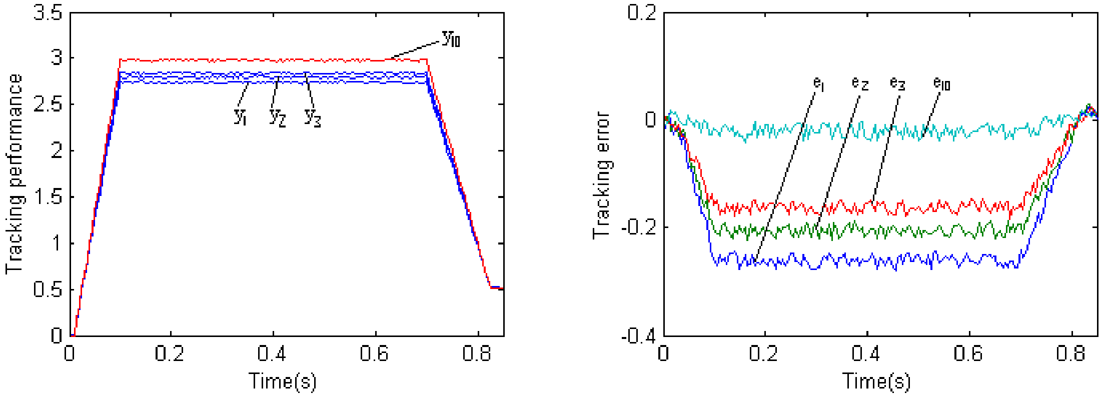


Figure 6.6: Tracking performance by a pure ILC: Measurement noise is considered.

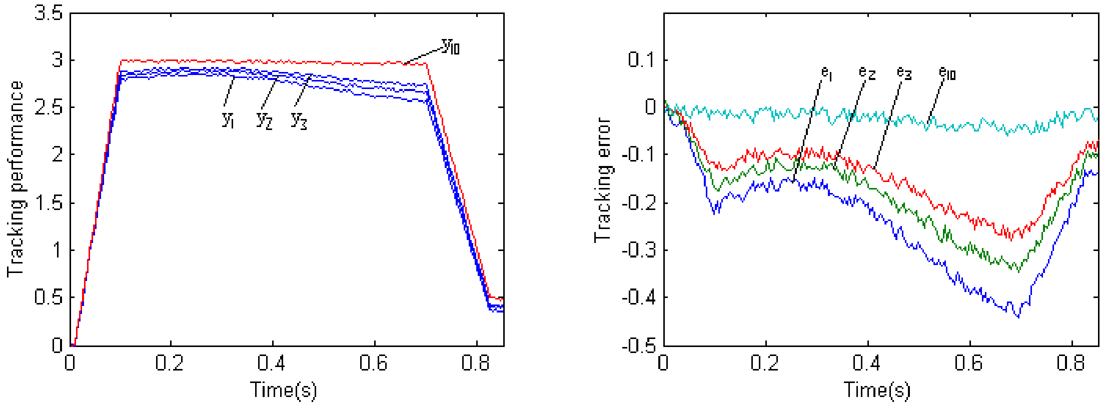


Figure 6.7: Tracking performance by a pure ILC: Measurement noise and repetitive disturbance are considered

6.5 Conclusions

An iterative learning control algorithm enhanced with predictive features has been developed in this chapter for time-varying linear systems. An error model is introduced which can represent the transition of tracking error in two successive trials. Based on this model, a predictive and iterative learning control algorithm is derived which is only based on the trial number (or repetition index). A rigorous analysis of the convergence of this hybrid algorithm is provided. In addition, the robustness of the algorithm against modeling errors, initial errors, as well as the presence of disturbances are discussed. An example on injection molding is provided which has verified the applicability of the proposed approach under various non-ideal scenarios.

Chapter 7

Conclusions

7.1 Summary of Contributions

As the micro and nanotechnology is gradually penetrating the development of the modern manufacturing industries, high speed, high accuracy positioning systems become essential to yield higher quality products with a higher productivity. The increasingly stringent requirements pose an great challenge for the controller design. The researchers attempt to seek the new and novel control algorithms to meet these demands beyond the traditional control theory. In this thesis, intelligent learning control algorithms are developed to achieve better performance for the precision motion control systems.

First, an adaptive control is presented to reduce the force ripple effects. The displacement periodicity of the force ripples is obtained by using FFT analysis. With a full model structure, the Recursive Least Square (RLS) estimation algorithm is designed to identify the parameters. The adaptive feedforward controller is formed to fulfill the objective.

Then, an Iterative Learning Control (ILC) scheme is developed which is suitable for high precision and repetitive motion control applications. Unlike the usual ILC scheme which applies a feedforward control signal to improve the tracking performance, the proposed control scheme is used to adjust the command to the feedback controller. The weights of RBF network are tuned online based on the remnant tracking error from cycle to cycle. The ILC component further enhances the tracking performance.

Subsequently, this thesis presents an approach for closed-loop automatic tuning of PID controller based on an Iterative Learning Control (ILC) method. A modified iterative learning control scheme iteratively changes the control signal by adjusting the periodic reference signal. With the obtained satisfactory performance, the PID controller is tuned by fitting the controller to yield a close input and output characteristics of the ILC component.

Following that, the repetitive control scheme is extended to a class of systems with time delay. The convergence condition is derived for the new repetitive control. Additionally, a robust convergence analysis is discussed in consideration of the time-delay modeling error, initialization error, disturbances and measurement noise.

Finally, a predictive Iterative Learning Control (ILC) algorithm is developed for time-varying linear systems. An error model is introduced, which represents the transition of the tracking error between two successive trials. Based on the error model, the predictive and iterative learning algorithm is derived. The convergence and robust convergence analyses are discussed respectively.

In this thesis, the proposed intelligent learning control algorithms are supported by the simulation and experimental results.

7.2 Suggestions for Future Work

The thesis has presented the research works on the intelligent learning control for the precision motion control systems. Further research topics in this field are suggested as follows.

In Chapter 4, the Iterative Learning Control (ILC) is applied to tune the PID controller. It is realized by fitting the input and output signal obtained by ILC. The equivalent PID controller is chosen as a linear controller. The final PID controller is tuned to be the closest to the obtained input and output signals by best fitting. ILC, as a model-free method, can reduce the tracking error as small as possible even though in the presence of the nonlinear effects in the system. However, in this case, although the ILC scheme can achieve almost perfect performance, the finally tuned linear PID controller cannot obtain the same performance as ILC, attributed to the presence of the nonlinear effects. Thus, there is a compromise between the achieved ILC performance and the PID tuning. In order to solve this issue, in the ongoing and future work, a nonlinear PID controller is suggested as the equivalent PID controller to achieve the further improvement.

In Chapter 5, the repetitive learning control method is extended to the time-delay systems. Good convergence performance can be obtained with the modified RC method. It is naturally expected that the idea in Chapter 4 can be extended to the systems with time delay. Thus, a new PID tuning method can be obtained for the time-delay systems. This represents one direction of the future research work. Additionally, in Chapter 5, the proposed method is verified by the simulation study. Time delays can be found in many manufacturing industries. The proposed method provides a new view of controlling the systems with time delay. The experimental study is helpful in practical implementation. It is suggested that an experimental study can be conducted in the ongoing work for the time-delay systems.

In some chapters of this thesis, the designs of the control algorithms are conducted on the single axis linear motors. However, the proposed intelligent learning controls are not confined to the single axis linear motors. They can be extended to the gantry systems. Gantry systems are widely used in machining industries, such as lathes and milling machines, and semiconductor manufacturing industries. Among the various gantry systems, the XY table and H-type are most popular positioning systems. For the gantry systems, the consideration of the design focuses on the control of the individual motors to track the desired motion trajectories and the synchronization. The intelligent learning controls can help the gantry systems operate efficiently with high speed and high accuracy.

Bibliography

[1] Hiromu N. Principles of precision engineering. Oxford New York Tokyo, Oxford University Press. 1994.

[2] Tan, K. K., Lee, T. H., Dou, H. F. and Huang, S. N. Precision Motion Control Design and Implementation. Chapter 2, Precision Tracking Motion Control, Springer, Advances in Industrial Control. 2001.

[3] Moriwaki, T. and Shamoto, E. Ultraprecision Feed System Based on Walking Drive. Annals CIRp. Vol.46, no.1, pp. 505-508, Jan. 1997.

[4] Kim, J. and Kim, D. Development and application of an ultraprecision lathe, Int. J. Adv. Manufact. Technol. Vol.13, pp. 164-171. 1997.

[5] Wang, Z., Jouaneh, M.K. and Dornfeld, D.A. Design and characterization of a linear motion piezoelectric micropositioner, IEEE Control System Magzine, pp. 10-15, Feb. 1990.

[6] Takahashi, M., Kurosawa, M. and Higuchi, T. Direct friction driven surface acoustic wave motor, Proc. 8th Int. Conf. Solid-State Sensors and Actuators, pp. 401-404. 1995.

[7] Thomas, W. W. An introduction to process dynamics and control. A Wiley-Insterscience Publication, pp. 3-5. 1973.

[8] Arimoto, S., Kawamura, S. and Miyazaki, F. Bettering operation of robots by learning, J. Robotic Systems, vol.1, pp. 123-140. 1984.

[9] Moore, K. L. Iterative Learning Control for Deterministic Systems, Advances in Industrial Contr. Ser.London, U.K. Springer-Verlag. 1992.

[10] Chen, Y. Q. and Wen, C. Iterative Learning Control: Convergence, Robustness, and Applications, Springer-Verlag. 1999.

[11] Moore, K. L. Iterative learning control: A expository overview. Applied and Computational Controls, Signal Proceeding, and circuits, 1, pp. 151-214, 1998.

[12] Hara, S., Yamamoto, Y., Omata, T. and Nakano, M. Repetitive control system: A new type servo system for periodic exogenous signals. IEEE Transactions on Automatic Control, Vol. 33, No. 7, pp. 659-668, July 1988.

[13] Tomizuka, M. Tsao T. C. and Chew, K. K. Analysis and synthesis of discrete-time repetitive controllers. Journal of Dynamic Systems, Measurement and Control, Vol. 111, pp. 353-358, Sept. 1989.

[14] Ledwich, G. F. and Bolton, A. Repetitive and periodic controller design. In IEE proceedings Part D, Control Theory and Applications, 140, pp. 19-24, 1993.

[15] Longman, R. W. Design iterative learning and repetitive controllers, In Iterative Learning Control: Analysis, Design, Integrarion and Applications, Edited by Z. Bien and J.X. Xu, pp. 107-146, Kluwer Academic, 1998.

[16] Sison, L. G. and Chong, E. K. P. No-reset iterative learning control. In proceedings of the 35th IEEE Conference on Devision and Control, Kobe, Japan, Vol. 3, pp. 3062-3063, 1996.

[17] Siston, L. G. and Chong, E. K. P. Repetitive lerning control for linear periodic plants. Proceedings of the 37th IEEE conferences on Decision & Control, pp. 1242-1247, Tampa, Florida USA, December 1998.

[18] Moore, K. L. A non-standard iterative learning control approach to tracking periodic signals in discrete-time non-linear systems. Int. J. Control, Vol. 73, No. 10, pp. 955-967, 2000.

[19] Hsin, Y. P., Longman, R. W., Solcz, E. J. and Jong, J. de. Experiments bridging learning and repetitive control. Advances in the Astronautical Sciences, Vol. 95, pp. 671-690, 1997.

[20] Wen, H. P., Phan, M. Q. and Longman, R. W. Bridging learning control and repetitive control using basis functions. Advances in the Astronautical Sciences, Vol. 99, pp. 335-354, 1998.

[21] Lee, S. W. and Kim, J. H. Robust adaptive stick-slip friction compensation, IEEE Indust. Electron. Mag., Vol.42, pp. 474C479, Oct. 1995.

[22] Kim, J. H., Chae, H. K., Jeon, J. Y. and Lee, S. W. Identification and control of systems with friction using accelerated evolutionary programming, IEEE Contr. Syst. Mag., pp. 38-47, Aug. 1996.

[23] Akmese, R. and Eastham, J. F. Design of permanent magnet linear motors for standstill applications, IEEE Trans. Mag., Vol.28, no.5, pp. 3042-3044. 1992.

[24] Li, T. and Slemon, G. Reduction of cogging torque in permanent magnet motors, IEEE trans. Mag., Vol.24, no.6, pp. 2901-2903. 1988.

[25] Yoshimura, T., Watada, M., Torii, S. and Ebihara, D. Study on the reduction of detent force of permanent magnet linear synchronous motor, Proc. of 1st International Symposium on Linear Drives for Industry Applications, June, 1995, Nagasaki, Japan, pp. 207-210.

[26] P. Van Den Braembussche, Swevers, J., H. Van Brussel and Vanherck, P. Accurate tracking control of linear synchronous motor machine tool axes. Mechatronics, Vol.6, no.5, pp. 507-521. 1996.

[27] Christof, R. and Andreas J. Identification and compensation of force ripple in linear permanent magnet motors. Proc. of the American Control Conference, June 2001, Arlington, VA, pp. 2161-2166.

[28] Otten, G., J.A.de Vries, J.van Amerongen, A.M.Rankers and Gaal, E. W. Linear motor motion control using a learning forward controller. IEEE trans. on Mechatronics, vol.2, no.3, pp.179-187,1997.

[29] Alter, D.M. and Tsao, T.C. Control of linear motors for machine tool feed drives: design and implementation of H_∞ optimal feedback control. ASME J. of Dynamic Systems, Measurement and Control, Vol.118, pp. 649-658. 1996.

[30] Bodson, M. and Douglas, S. C. Adaptive algorithms for the rejection of sinusoidal disturbances with unknown frequency. Automatica, Vol. 33, No. 12, pp. 2213-2221, 1997.

[31] Bodson, M. Performance of an adaptive algorithm for sinusoidal disturbance rejection in high noise. Automatica, 37, pp. 1133-11401, 2001.

[32] Guo, X. Y. and Bodson, M. Adaptive rejection of disturbances having two sinusoidal components with close and unknown frequencies. American Control Conference, pp. 2619-2624, June 8-10, Portland, OR, USA, 2005.

[33] Kharitonov, V. Robust stability analysis of time delay systems: A survey. In Fourth IFAC conference on system structure and control, Nantes, Frnace, 8-10 July, Penary lecture pp. 1-12, 1998.

[34] Kolmanovskii, V.B., Niculescu, S. I. and Gu, K. Delay effects on stability: A survey. In 38*th* IEE CDC'99 (Conference on decision and control), Phoenix, AZ, Decmber, pp. 1993-1998,1999.

[35] Mirkin, L. and Tadmor, G. H_∞ control of systems with I/O delay: a review of some problem-oriented methods. IMA Journal of Mathematical Control and Information, 19(1-2), pp. 185-200, 2002.

[36] Richard, J. P. Time-delay systems: an overview of some recent advances and open problems, Automatica, vol. 39, pp. 1667-1694, 2003.

[37] Gu, K., Kharitonov, V. L. and Chen, J. Stability of Time-Delay Systems. Birkhauser, 2003.

[38] Michiels, W. Assche, V. and Niculescu, S. I. Stability of time-delay systems with a controlled time-varying delay and applications. IEEE Transaction on Automatic Control, Vol. 50, No. 4, pp. 493-504, 2005.

[39] Nguang, S. K. Robust stabilization of a class of time-delay nonlinear systems. IEEE Transaction on Automatic Control, Vol. 45, No. 4, pp. 756-762, 2000.

[40] Ge, S. S., Hong, F. and Lee, T. H. Robust adaptive control of nonlinear systems with unknown time delays. Automatica, 41, pp. 1181-1190, 2005.

[41] Hideg, L. M. Time delays in iterative learning control schemes. In Proceedings of the 1995 IEEE International symposium on Intelligent Control, pp. 5-20, Monterey, CA, August, 1995.

[42] Hideg, L. M. Stablility and convergence issues in iterative learning control - II. In Proceedings of the 1995 IEEE International symposium on Intelligent Control, pp. 480-485, Dearborn, MI, September, 1996.

[43] Xu, J.X., Hu, Q.P., Lee, T.H. and S. Yamamoto. Iterative learning control with Smith time delay compensator for batch processes, Journal of Process Control, Vol.11, no.3, pp. 321-328. 2001.

[44] Hu, Q.P., Xu, J.X. and Lee, T.H. Iterative leaning control design for smith predictor, Systems and Control Letters, vol.44, 201-210, 2001.

[45] Lee, K. S. and Lee, J. H. Design of quadratic criterion based iterative learning control, In Iterative Learning Control: Analysis, Design, Integrarion and Applications, Edited by Z. Bien and J.X. Xu, pp. 165-191, Kluwer Academic, 1998.

[46] Owens, D. H. and Hatonen, J. Iterative learning control - An optimization paradigm, Annual Reviews in Control, 29, pp. 57-70, 2005.

[47] Owens, D. H. and Feng, K. Parameter optimization in iterative learning control. Int. J. Control, Vol. 76, No. 11, pp. 1059-1069, 2003.

[48] Hatzikos, V. Hatonen, J. and Owens, D. H. Genetic algorithms in norm-optimal linear and non-linera iterative learning control. Int. J. Control, Vol. 77, No. 2, pp. 188-197, 2004.

[49] Frueh, J. A. and Phan, M. Q. Linear quadratic optimal leraning control. Int. J. Control, Vol. 73, No. 10, pp. 832-839, 2000.

[50] Amann, N., Owens, D. H. and Rogers, E. Iterative learning control for discrete time systems using optimal feedback and feedforward actions. In Proceedings of the 34th

Conference on Decision & Control, pp. 1696-1701, New Orleans, LA, December, 1995.

[51] Braembussche, P. V., Swevers, J., Brussel, H. V. and Vanherck, P. Accurate tracking control of linear synchronous motor machine tool axes, Mechatronics 6(5), pp. 507-521. 1996.

[52] Hu, A. P., Register, A. and Sadegh, N. Using a learning controller to achieve accurate linear motor motion control, Proc. 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 611-616, 19-23 September, 1999, Atlanta, USA.

[53] Yao, B. and Xu, L. Adaptive robust precision motion control of linear motors with ripple force compensations: theory and experiments, Proc. 2000 IEEE International Conference on Control Applications, pp. 373-378, 25-27 September, 2000, Anchorage, Alaska, USA.

[54] Yao, B. and Xu, L. Adaptive robust motion control of linear motors for precision manufacturing, Mechatronics 12, pp. 595-616. 2002.

[55] Huang, S. N., Lee, T. H. and Tan, K. K. Robust adaptive numerical compensation for friction and force ripple in permanent magnet linear motors, IEEE Transactions on Magnetics, 38(1), pp. 221-228. 2002.

[56] Lee, T. H., Tan, K. K., Lim, S. Y. and Dou, H. F. Iterative learning of permanent magnet linear motor with relay automatic tuning, Mechatronics 10, pp. 169-190. 2000.

[57] Fujimoto, Y. and Kawamura, A. Robust servo-system based on two-degree-of - freedom control with sliding mode, IEEE trans. on Industrial Electronics 42(3), pp. 272-280. 1995.

[58] Direct Thrust Linear Servo Motors and Systems, Linear Drives Limited. 1997.

[59] Ljung, L. System identification theory for the user. 2nd-Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey. 1997.

[60] Astrom, K. J. and Wittenmark, B. Adaptive control. Second Edition, Addison-Wesley. 1995.

[61] Chen, Y. Q. and Moore, K. L. Harnessing the nonrepetitiveness in Iterative Learning Control, Proceedings of the 41st IEEE Conference on Decision and Control, pp. 3305-3355, Las Vegas, Nevada USA, December 2002.

[62] Xiao, J., Song, Q. and Wang, D. A learning control scheme based on neural networks for repeatable robot trajectory tracking, Proc. of the 1999 IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics, pp. 102-107, 1999.

[63] Chen, Y. Q., Moore, K. L. and Bahl, V. Learning feedforward control using a dilated B-Spline network: frequency domain analysis and design. IEEE Transactions on Neural Networks, Vol. 15, No. 2, pp. 355-366, 2004.

[64] Hornik, K., Stinchcombe, M. and White, H. Multilayer feedforward networks are universal approximators, Neural Network, Vol. 2, pp. 359-366. 1989.

[65] Holcomb, T. and Morari, M. Local training of radial basis function networks: Towards solving the hidden unit problem, Proc. of the American Control Conference, pp. 2331-2336. 1991.

[66] Haykin, S. Neural Networks A Comprehensive Foundation, 2nd ed., Prentice Hall, pp. 299. 1999.

[67] Gorinevsky, D. On the persistency of excitation in radial basis function network identification of nonlinear systems, IEEE Trans. Neural Networks, Vol. 6, no. 5 , pp. 1237-1244. 1995.

[68] Mitsuo, K., Yoji, U., Michiaki, I. and Ryoji, S. Hierarchical neural network model for voluntary movement with application to robotics, IEEE Control System Magazine, pp. 8-15. 1988.

[69] Longman, R. W. Iterative learning and repetitive control for engineering practice, International Journal of Control, Vol.73, no.10, pp. 930-954. 2000.

[70] Moore, K. Iterative learning for trajectory control, Proc. of the 28th Conference on Decision and Control, pp. 860-865. 1989.

[71] Dou, H. F., Zhou, Z. Y., Sun, M. and Chen, Y. Robust High-Order P-type Iterative Learning Control for a Class of Uncertain Nonlinear Systems, Proc. of the IEEE Int. Conf. On Systems, Man, and Cybernetics, pp. 923-928. 1996.

[72] Huang, S. N., Tan, K. K. and Lee, T. H. Necessary and sufficient condition for convergence of iterative learning algorithm, Automatica, Vol.38, Issue 7, pp. 1257-1260. 2002.

[73] Chen, C. T. Linear System Theory and Design. 3rd ed., Oxford University Press, Inc., New York. 1999.

[74] Rivera, D. E., Morari, M. and Skogestad, S. Internal model control for PID controller design, Industrial Engineering Chemical Process Design Development, Vol.25(1), pp. 252-265. 1986.

[75] Gawthrop, P. J. Self-tuning PID controllers: Algorithms and implementations, IEEE Transaction on Automatic Control, Vol.31, no.3, pp. 201-209. 1986.

[76] Jiawen, D. and Brosilow, C. B. . Nonlinear PI and gain-scheduling, Proceedings of the 1998 American control conference, Vol.1, pp. 323-327.

[77] Astrom, K. J., Hagglund, T., Hang, C. C. and Ho, W. K. Automatic tuning and adaptation for PID controllers- a survey, Control Engineering Practice, Vol.1, pp. 699-714. 1993.

[78] Tan, K. K., Wang, Q-G. and Hang, C. C. Advances in PID control. Springer Verlag London. 1999.

[79] Huang, H. P., Chen, C. L., Lai, C. W. and Wang, G. B. Autotuning for model based PID controllers, The American Institute of Chemical Engineers Journal, Vol.42(9), pp. 2687-2691.

[80] Reza, G. and Blankenship, G. L. An adaptive PID controller for nonlinear systems, Proceedings of the 30th Conference on Decision and Control, December, 1991, Brighton, England, pp. 2488-2492.

[81] Badreddine, B. M. and Lin, F. Adaptive PID controller for stable/unstabel linear and non-linear systems, Proceedings of the 2001 IEEE International conference on Control Applications, September 5-7, 2001, Mexico, pp. 1031-1036.

[82] Tan, K. K., Lee T. H. and Zhou, H. X. Micro-positioning of linear-piezoelectric motors based on a learning nonlinear PID controller, IEEE/ASME Trans. on Mechatronics, Vol.6, no.4, pp. 428-436. 2001.

[83] Goldsmith, P. B. On the equivalence of causal LTI iterative learning control and feedback control, Automatica, 38(4), pp. 703-708, 2002.

[84] Tan, K. K., Zhao S. and Huang, S. N. Iterative Reference Adjustment for High Precision and Repetitive Motion control Applications, IEEE Trans. on Control Systems Technology, Vol.13, no.1, pp. 85-97. 2005.

[85] Choi, H. H. and Chung, M. J. Memoryless stabilization of uncertain dynamic systems with time varying delayed states and controls, Automatica, Vol.31, pp. 1349-1351. 1995.

[86] Huang, S. N. and Ren, W. Longitudinal Vehicle Following Control with Time Delays in Platooning, IEE Proceedings-D, Vol.145, pp. 211-217. 1998.

[87] Roh, Y. H. and Oh, J. H. Robust stabilization of uncertain input-delay systems by sliding mode control with delay compensation, Automatica, Vol.35, pp. 1861-1865. 1999.

[88] Geng, Z., Carroll, R. L. and Xie, J. Two-dimensional model algorithm analysis for a class of iterative learning control systems. Int. J. Control, Vol.52, pp. 833-862. 1990.

[89] Kurek, J. E. and Zaremba, M. B. Iterative learning control synthesis based on 2-D system theory, IEEE Trans.on Automatic Control, AC-38, no.1, pp. 121-125. 1993.

[90] Saab, S. S. A discrete learning control algorithm for a class of linear time-invariant systems, IEEE Trans.on Automatic Control, AC-40, no.6, pp. 1138-1141. 1995.

[91] Lee, J. H., Lee, K. S. and Kim, W. C. Model-based iterative learning control with a quadratic criterion for time-varying linear systems, Automatica, Vol.36, pp. 641-657. 2000.

[92] Horowitz, R., Messner, W. and Boals, M. Exponential convergence of a learning control for robot manipulator, IEEE Trans. Automatic Control, AC-36, pp. 890-894. 1991.

[93] Xu, J. X. and Qu, Z. Robust iterative learning control for a class of nonlinear systems, Automatica, Vol.34, pp. 983-988. 1998.

[94] Xu, J. X., Chen, Y. Q., Lee, T. H. and Yamamoto, S. Terminal iterative learning control with an application to RTPCVD thickness control, Automatica, Vol.35, pp. 1535-1542. 1999.

[95] Chen, Y., Wen, C., Gong, Z. and M. Sun. An iterative learning controller with initial state learning", IEEE Trans. Automat. Control, Vol.44, pp. 371-376. 1999.

[96] Chen, Y., Gong, Z. and Wen, C. Analysis of a high-order iterative learning control algorithm for uncertain nonlinear systems with state delays, Automatica, Vol.34, pp. 345-353. 1998.

[97] Hillenbrand, S. and Pandit, M. Discrete-time iterative learning control law with exponential rate of convergence, Proceedings of the 38th IEEE Conference on Decision & Control, Phoenix, Arizona USA, December 1999, Vol.2, pp. 1575-1580.

[98] Chen, Y. Q., Xu, J. X. and Lee, T.H. High-order iterative learning control of discrete-time nonlinear systems using current iteration tracking error, In Iterative Learning Control: Analysis, Design, Interation and Applications, Edited by Z. Bien and J.X. Xu, pp. 83-107, Kluwer Academic. 1998.

[99] Bondi, P., Casalino, G. and Gambardella, L. On the iterative learning control theory for robotic manipulators, IEEE Trans.on Robotics and Automation, Vol.4(2), pp. 14-22. 1988.

[100] Bien, Z. and Huh, K. M. High-order iterative control algorithm, IEE Proceedings, Part-D, Control Theory and Applications, Vol.13(3), pp. 105-112. 1989.

[101] Oh, S. R., Bien, Z. and Suh, I. H. An iterative learning control method with application for the robot manipulator, IEEE Trans.on Robotics and Automation, Vol.4(5), pp. 508-514. 1988.

[102] Lucibello, P. Learning control of linear systems, Proc.of American Control Conference, pp. 1888-1892. 1992.

[103] Sogo, T. and Adachi, N. A gradient-type learning control algorithm for linear systems, Proc.of ASCC, Vol.3, pp. 227-230, Tokyo,Japan. 1994.

[104] Lee, K.S. and Lee, J.H. Model-based predictive control combined with iterative learning for batch processes, pp. 314-334, Edited by Z.Bien and J.X. Xu,Kluwer Academic. 1998.

[105] Amann, N., Owens, D.H. and Rogers, E. Predictive optimal iterative learning control, Int. J. Control, Vol.69, pp. 203-226. 1998.

[106] Gunnarsson, S. and Norrlof, M. On the design of ILC algorithms using optimization, Automatica, vol.37, pp. 2011-2016, 2001.

[107] Arimoto, S. Learning control theory for robotic motion, International Journal of Adaptive Control and Signal Processing, Vol.4, pp. 277-293. 1990.

[108] Clarke, D. et al. Generalized predictive control-part 1, basic algorithm", Automatica, Vol.23, pp. 137-148. 1987.

[109] Huang, S. N., Tan, K. K. and Lee, T. H. Adaptive GPC control of melt temperature in injection moulding, ISA Transactions, Vol.38, pp. 361-373. 1999.

[110] Burl, J. B. Linear Optimal Control: H_2 and H_∞ Methods, Addison-Wesley Longman, Inc. 1999.

[111] Huang, S. N., Tan, K. K. and Lee, T. H. Generalized predictive observer-controller for injection molding with input delay and non-measurable noise, Intern. Polymer Processing, Vol.14(4), pp. 399-408. 1999.

Author's Publications

Journal Papers:

1. K.K. Tan, S.N. Huang and S. Zhao. Novel Predictive and Iterative Learning Control Algorithm, Control and Intelligent Systems, Vol.31, No.1, pp. 1-9, 2003.

2. K.K. Tan, T.H. Lee, H.F. Dou, S.J. Chin and S. Zhao. Precision Motion Control With Disturbance Observer for PWM-Driven Permanent Magnet Linear Motors, IEEE Transactions on Magnetics, Vol.39, No.3 May, pp. 1813-1818, 2003.

3. K.K. Tan and S. Zhao. Precision Motion Control with a High Gain Disturbance Compensator for Linear Motors, ISA Transactions, Vol.43, No.3, 2004.

4. K.K. Tan, T.H. Lee, H. Dou and S. Zhao. Force ripple suppression in iron-core permanent magnet linear motors using an adaptive dither, Journal of the Franklin Institute, 341, pp. 375-390, 2004.

5. K.K. Tan, S. Zhao and S.N. Huang. Iterative Reference Adjustment for High Precision and Repetitive Motion control Applications, IEEE Trans on Control Systems Technology, Vol.13, Issue.1, pp. 85-97, 2005.

6. S. Zhao and K.K. Tan. Adaptive Feedforward Compensation of Force Ripples in Linear Motors, Control Engineering Practice - to appear, 2005.

7. K.K. Tan and S. Zhao. Intelligent Compensation of Friction and/or Ripple Compensation Via a Regulated Chatter, International Journal of Advanced Robotics - to appear, 2005.

8. K.K. Tan, S. Zhao, T.H. Lee and S.N. Huang. New Iterative Learning Control for Time-Delay Systems, submitted to Automatica, Jan, 2005.

9. K.K. Tan, S. Zhao and J.X. Xu. Online Automatic Tuning of PID Controller Based on an Iterative Learning Control Approach, submitted to IEE Proc. Control Theory & Applications, Jan, 2005.

Conference Papers:

1. K.K. Tan and S. Zhao. Adaptive Force Ripple Suppression in Iron-core Permanent Magnet Linear Motors, Intelligent Control, 2002. Proceedings of the 2002 IEEE Internatinal Symposium on, pp. 266-269, 2002.

2. K.K. Tan and S. Zhao. Iterative Reference Adjustment for High Precision and Repetitive Motion control Applications, Intelligent Control, 2002. Proceedings of the 2002 IEEE Internatinal Symposium on, pp. 131-136, 2002.

3. K.K. Tan and S. Zhao. Regulated Chatter for Friction and/or Ripple Compensation in Linear Motors, Fourth International Conference on Intelligent Technologies (Intech'03), pp. 247-253.

4. K.K. Tan, H. Dou and S. Zhao. Adaptive feedforward compensation of force ripples in linear motors, Mechatronics and Machine Vision In Practice 10th Annual International Conference Perth, P04-1 to P04-8, Western Australia, 2003.

5. S. Zhao and K.K. Tan. Challenges in the development of precise positioning systems for MEMS/Nanotechnology, SPIE International Symposium Microelectronics, MEMS, and Nanotechnology, 10-12 December 2003, pp. 119-130, Perth, Australia.

Chapters in Books:

1. K.K. Tan, H. Dou and S. Zhao. Adaptive Feedforward Compensation of Force Ripples in Linear Motors, in Mechatronics and Machine Vision 2003: Future Trends, John. Billingley, Research Study Press, Baldock, Hertfordshire, England, ISBN 0-86380-290-7 (Hardcover), pp. 367-376.

2. K.K. Tan and S. Zhao. Iteratively Learning Motion Control of Linear Motors, Editor: Branko Katalinic, DAAAM International Scientific Book 2004, DAAAM International Vienna, Vienna 2004, ISSN 1726-9687, ISBN 0-901509-38-0 (Hardcover) - Chapter 54, pp. 587-604.