Iterative Learning Reference Trajectory Modification for Contouring Performance Enhancement of Industrial Machine Tool Feed Drive Systems

(工作機械送り駆動系の輪郭制御性能向上のための繰返し学習に基づく参照軌道修正)

July, 2018

Doctor of Philosophy (Engineering)

Yogi Muldani Hendrawan

Toyohashi University of Technology

Declaration of Authorship

I, Yogi Muldani Hendrawan, declare that this thesis titled, “Iterative Learning Reference Trajectory Modification for Contouring Performance Enhancement of Industrial Machine Tool Feed Drive Systems” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“*Knowledge exists potentially in the human soul like the seed in the soil; by learning the potential becomes actual.*”

“*Your knowledge must improve your heart, and purge your ego.*”

Imam Al-Ghazali

TOYOHASHI UNIVERSITY OF TECHNOLOGY

Abstract

System Engineering Laboratory

Department of Mechanical Engineering

Doctor of Philosophy (Engineering)

Iterative Learning Reference Trajectory Modification for Contouring Performance Enhancement of Industrial Machine Tool Feed Drive Systems

by Yogi Muldani Hendrawan

In industrial applications, highly accurate mechanical components are generally required to produce advanced mechanical and mechatronic systems. Most of them are produced by computer numerical control (CNC) machine tools. A fundamental motion in CNC machine tools is a drive axial movement to track a desired trajectory. Not only are there tracking errors in each drive axis, but there are also contour errors, which are directly related to the machined shape of a workpiece, and therefore must be considered in controller design. Although most existing contouring controllers are based on feedback control and estimated contour error, it is generally difficult to replace the feedback controller in commercial CNC machine tools. In order to improve the contouring performance for commercial CNC machine tools, this thesis presents an iterative learning contouring controller (ILCC) design with two contour error estimation approaches.

The proposed method is implemented in three types of CNC machine tool feed drive systems. First, a laboratory biaxial feed drive system is used to prove the proposed methods performance (Chapter 3) with an estimated contour error correction.

Then, an ILCC is applied to a three-axis commercial CNC machine tool (Chapter 4) to improve the contour error estimation. Last, a machine tool with a linear motor mechanism is used as an example of an advanced machine tool (Chapter 5). Descriptions of the physical system configurations and system parameters are presented in Chapter 2.

An ILCC that considers both tracking and contour errors is proposed in Chapter 3. The proposed control iteratively modifies the reference trajectory of each drive axis to reduce the contour error. The proposed controller can be directly applied to commercial machines currently in use without requiring any modification of their original controllers. The proposed method has been experimentally verified through a biaxial feed drive system on a sharp-corner trajectory, which normally leads to a large contour error around the corner due to the discontinuity. A comparison with a conventional ILCC (CILCC) was done in order to evaluate its performance. The experimental results show that the contour error converges within a few iterations, and the maximum contour error can be reduced by 49.2 % in comparison with the CILCC. The limitation is that this method is only effective for a low-curvature trajectory. It requires more iterations to track a high-curvature trajectory.

An ILCC that considers the actual contour error compensation (ACEC) with linear interpolation and the Bézier reposition trajectory (BRT) is proposed in Chapter 4 to improve upon the method presented in Chapter 3. The ACEC enhances tracking performance for a high-curvature trajectory by correcting the reference input with an actual contour error value, and the BRT enables smooth velocity transitions between discrete points in the reference trajectory. For the performance analysis, the proposed controller was implemented in a commercial three-axis CNC machine tool and several experiments were conducted on the basis of typical 3D sharp-corner and half-circular trajectories. The experimental results show that the proposed controller could reduce the maximum and mean contour errors by 45.11 % and 54.48 % on average, compared with the ILCC with estimated contour error. In comparison with the ILCC with ACEC, the maximum and mean contour errors are reduced by 20.54 % and 26.92 %, respectively. However, this method is not effective for circular trajectories. It will be improved by the circular interpolation method in the next chapter.

To improve the effectiveness of the proposed method in an advanced commercial CNC machine tool system, circular interpolation is designed. It is implemented in a CNC machine tool with a linear motor mechanism in Chapter 5. An ILCC that considers ACEC with linear and circular interpolation enhances the contouring performance of linear motor CNC machine tool feed drive systems. The proposed control iteratively modifies the reference trajectory of each drive axis to reduce the contour error. The proposed controller can be directly applied to a commercial CNC machine tool with a linear motor mechanism currently in use without requiring any modification of the original controller. The comparisons between linear and circular interpolation were simulated in both "air-cutting" and machining conditions. The simulation was conducted for non-smooth rhomboidal and circular trajectories. The effectiveness of the proposed method has been experimentally verified with a rhomboidal trajectory. The results show that the proposed controller could reduce the maximum and mean contour errors by 94.58 % and 88.67 % on average, compared with the original controller. In addition, the proposed method improved the control input variance by 37.9 %, and consequently, the consumed energy was reduced by 11.7 % compared with the original NC program. Concluding remarks and future works are described in Chapter 6.

Acknowledgements

First of all, I like to express my thanks to ALLAH, the most beneficent, the Lord of universe, for his many and uncountable blessings upon us. Without his blessing, this work would not be accomplished. Peace and blessings be upon his messenger Muhammad SAW.

I would like to express my deep appreciation to Prof. Naoki Uchiyama for his willingness to take me into his research group and for the wonderful opportunity and passionate guidance, he gave me during the three-years researching. His enthusiasm and interest for science are contagious. I have learned not only scientific techniques from him, but also more importantly, the method of scientific thinking, how to identify a research problem, how to form and carry out a research plan, how to get results and how to communicate an idea to other people so that it could be understandable and acceptable.

I am grateful to my thesis review committe members, Prof. Kaiji Sato and Prof. Naohiro Fukumura for their great support, valuable advice, and constructive comments. In addition, I would like to thank to Associate Prof. Tatsuhiko Sakaguchi and Assistant Prof. Miyori Shirasuna to support me during my research in the System Engineering Laboratory. I also extend my thanks to all members of the System Engineering Laboratory for their assistance and friendship during my graduate study.

Furthermore, I would like to thank to the Ministry of Research, Technology and Higher Education of the Republic of Indonesia for providing me scholarship during my study and Bandung Polytechnic for Manufacturing, Indonesia, which gave me the chance and support to study at Toyohashi University of Technology. I also would like to thanks for all Toyohashi University of Technology staff members for their kindly assistance and friendship.

With my endless love, I would like to thanks to my wife Dini Rismalati. Her encouragement, support, quiet patience and unwavering love are undeniably. Her tolerance of my occasional vulgar moods is a testament in itself of her unyielding devotion and love. I also extend my thanks to all of my children Misykah Shadwah An Nuwairah, Maulana Syafiq Ar Rasyid, Muhammad Salman As Siddiq, and Mutsumi Shafiyah Azzahra. Their smiles become a huge spirit to finish my work.

I extend most heartfelt thanks to my parents for their endless support, sacrifice, and unconditional love. I also thanks to my sister, brother, sister-in-law, and brother-in-law, and for their constant understanding and encouragement. The warm love of my family motivates me to go on.

Last but not least, I would like to thank to Toyohashi Muslim Community, Indonesian Toyohashi Muslim Community (KMIT), Indonesian Student Association Toyohashi regional (PPI Toyohashi), Tsutsuji community, and all my friends at Toyohashi city. They constantly remind me about what our contribution to community.

Yogi Muldani Hendrawan

Contents

List of Figures

List of Tables

List of Abbreviations

AC	**A**lternating **C**urrent
ACEC	**A**ctual **C**ontour **E**rror **C**ompensation
BC	**B**ézier **C**urve
BRT	**B**ézier **R**eposition **T**rajectory
CAD	**C**omputer **A**ided **D**esign
CAE	**C**omputer **A**ided **E**ngineering
CAM	**C**omputer **A**ided **M**anufacturing
CCC	**C**ross-**C**oupled **C**ontroller
CCILC	**C**ross-**C**oupled **I**terative **L**earning **C**ontrol
CCL	**C**urrent **C**ycle **L**earning
CILC	**C**onventional **I**terative **L**earning **C**ontrol
CLM	**C**lose **L**oop **M**achining
CNC	**C**omputer **N**umerical **C**ontrol
DC	**D**irect **C**urrent
DNC	**D**irect **N**umerical **C**ontrol
EDM	**E**lectro **D**ischarge **M**achine
EILCC	**E**mbedded **I**terative **L**earning **C**ontouring **C**ontroller
FBC	**F**eed**B**ack **C**ontroller
IBC	**I**nserted **B**ézier **C**urve
ILC	**I**terative **L**earning **C**ontrol

ILCC	**I**terative **L**earning **C**ontouring **C**ontroller
LQG	**L**inear **Q**uadratic **G**aussian
LQR	**L**inear **Q**uadratic **R**egulator
MRR	**M**aterial **R**emoval **R**ate
MTPO	**M**inimum **T**ime **P**ath **O**ptimization
MTTC	**M**inimum **T**ime **T**racking **C**ontrol
NC	**N**umerical **C**ontrol
NURBS	**N**on-**U**niform **R**ational **B**-**S**pline
OS	**O**perating **S**ystem
PC	**P**ersonal **C**omputer
PCL	**P**revious **C**ycle **L**earning
PCCL	**P**revious and **C**urrent **C**ycle **L**earning
PID	**P**roportional **I**ntegral **D**erivative
RMS	**R**oot **M**ean **S**quare
ZPETC	**Z**ero **P**hase **E**rror **T**racking **C**ontroller

Dedicated to my parents, my wife, my children, and my beloved family, with all of my love . . .

Chapter 1

Introduction

1.1 Introduction

1.1.1 Background

In industrial applications, many processes exist to manufacture precision products. Machining is one of the processes required for metal processing, especially in the automotive, aerospace, energy, and medical industries. Many components such as propellers [1], gears [2], and knee joints [3] require high quality standards in accuracy, surface quality, and geometry, as shown in Fig. 1.1; all of these requirements

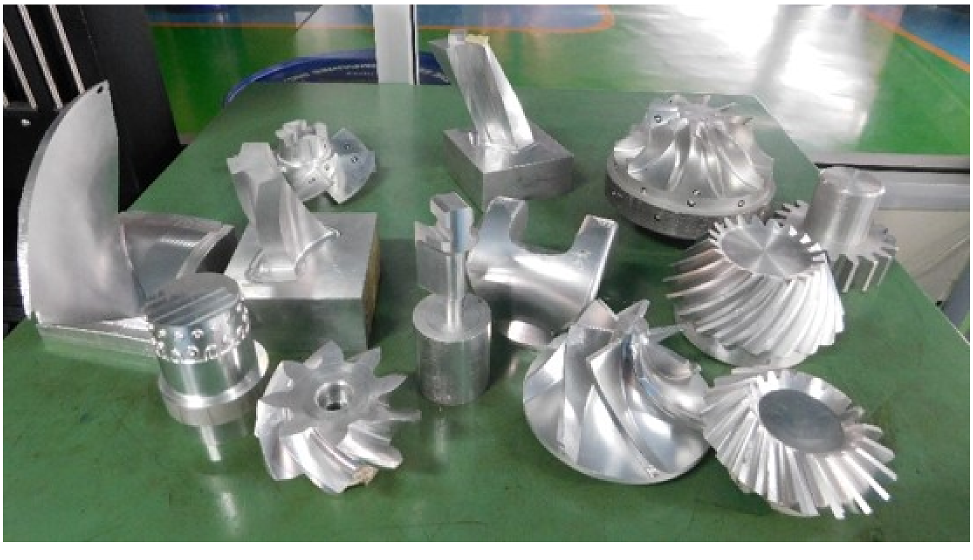


FIGURE 1.1: Machining product sample.

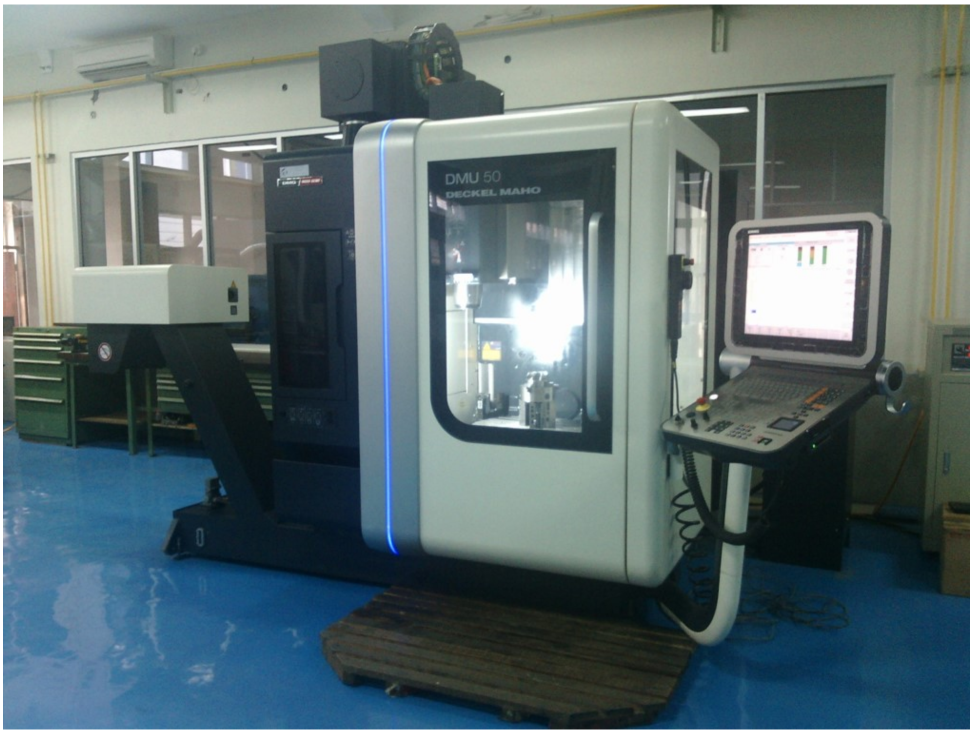


FIGURE 1.2: Five-axis CNC machine tools.

can be satisfied during the machining process in high-precision machine tool feed drive systems.

The basic function of a machine tool (as shown in Fig. 1.2) for removal processes is to move a cutting tool along a more or less complex trajectory with sufficient precision while withstanding the forces from the material removal process and still reaching the required precision or material removal rate (MRR) [4]. This basic function can be applied to produce a wide variety of product specifications, and the machine tool can be designed in a way that satisfies many requirements. The machine tool requirements are related to the following aspects:

- The maximum part size to be machined. Machining would be needed at each point of the part, so the machine workspace must be larger than the workpiece size.
- The main geometry of the workpiece. The global shape of the part is the key consideration when selecting the type of machine tool.

- The number and complexity of axes. If the features are few and simple, they may not require the use of a highly complex machine. On the other hand, if they are numerous or very complex, the required machine tool must be structurally much more complex (e.g., a five-axis milling machine).

- MRR. In some applications, the purpose of the designed driver is precision, but in others, it is mainly to achieve a high productivity via a high MRR.

- Precision. This is a commonly used word that actually involves two different concepts: *accuracy* and *repeatability*. Accuracy is the capability of being on target within a specification, quantified by the bias or difference between the obtained and desired results. Repeatability is the ability to reach the same objective over and over again.

- Kinematic behavior (i.e., speed and acceleration). This requirement represents a fast movement (idle movements) between successive machining operations and during tool changes.

- Batch size has an influence on the automation level of machine tools and the use of auxiliary devices. Generally, a universal CNC machine is able to process a wide range of different workpieces.

- Price. This is an important factor that tends to be linearly dependent on machine size and exponentially dependent on precision [4].

Accuracy and precision are the main objectives of machine tool construction. The guidelines and methodologies for machine design and assembly, the machine element manufacturing, the testing procedures, and the use of auxiliary systems are inspired by this requirement [5], [6]. A description of precision in the machining process is shown in Fig. 1.3. In the last decade, a precision of 1 μm has been achieved for conventional machining, whereas in ultra-precision machining, a hundredth of a micron has been achieved in some cases.

There are many guiding principles to consider in the design and construction of precision machine tools. Some of the considerations include the machine tool structure, the kinematic design principle and smooth motion, Abbe's principle, the position measurement, the principle of error correction, and machine tool position control [8]. Generally, the guiding principles can be classified into either hardware side or control side, the latter being related to error correction and machine tool position control. However, in order to control high-precision motion, it is necessary to utilize high-precision hardware, especially when it comes to machine tool structure, motion equipment, and measurement devices. For example, an error correction technique will work properly with a precise measurement instrument, but if a low-accuracy measurement device is used on a machine tool, it will limit the potential for improvement in the tool's performance or precision. This thesis proposes a method to improve machine tool performance by two of the guiding principles of precision machining, namely, error correction and machine position control, which will be described in the following section.

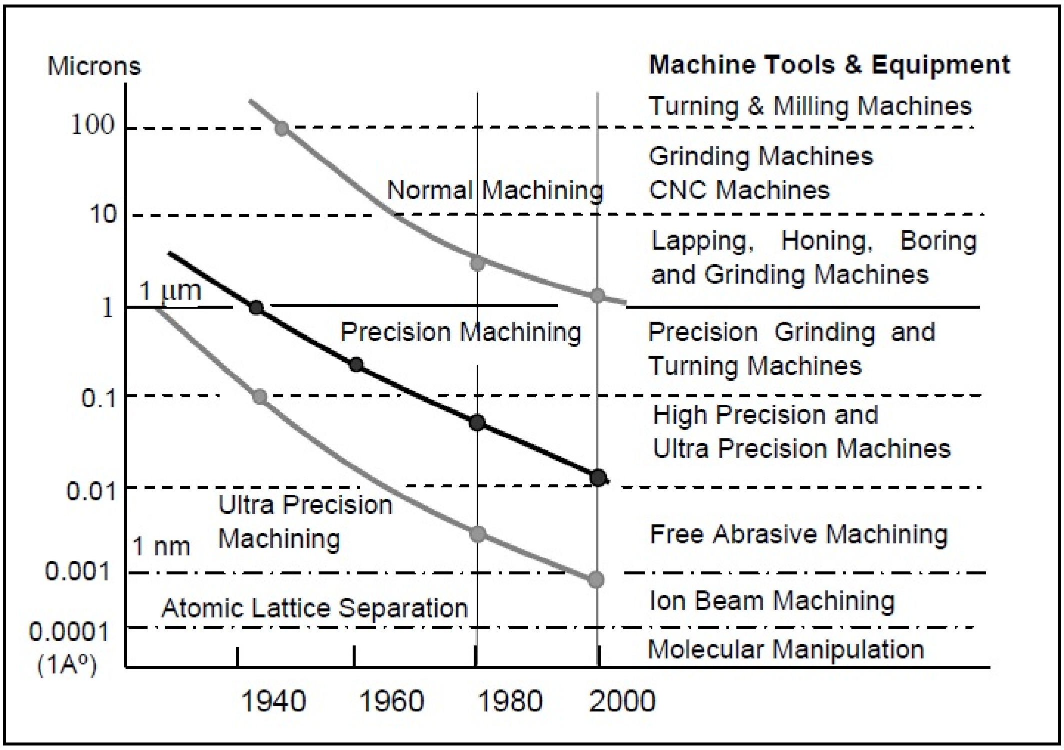


FIGURE 1.3: Evolution of machine tool accuracy [7].

1.1.2 Motion Control of Machine Tools

Machine tool performance can be improved by an error correction method. Many measurement devices can be used to measure the current error, but a precision measuring device can generate an accurate error value that can then be used to correct the machine tool error. There are two types of error that can be corrected in machining applications, tracking error and contour error, as shown in Fig. 1.4. The tracking error is defined as the difference between the desired and actual positions. The contour error is the error components orthogonal to the desired contour curves [9]. Both the tracking and contour errors can be addressed by error correction methods, although each error will have a different effect on the machine tool controller design.

In machining applications, a contour error is directly related to the product shape quality, so many researchers use contour error to correct machining error. However, a contour error can be difficult to estimate accurately. Researchers have estimated contour error by many different methods [10]–[17]. Most of the contour error approximation methods do not represent the actual contour error accurately, especially for a high-curvature surface application. However, in real-time control, the actual contour error cannot be obtained immediately for complex contour profiles. Therefore, it requires a precise contour error estimation to obtain approximate values. This thesis proposes two different types of contour error estimation for low- and high-curvature trajectories. Both types of contour error estimation will be implemented in machine position control, especially iterative learning control (ILC) for machine tool position.

The next important factor on the control side is machine tool position control. It must be designed under the filter effect principle, based on noise correction, in order to obtain precise and smooth motion. One of the noise sources is the resulting linear interpolations of complex curves. Computer-aided design (CAD) files are usually defined with high-order NURB curves. These curves are defined with a relatively small

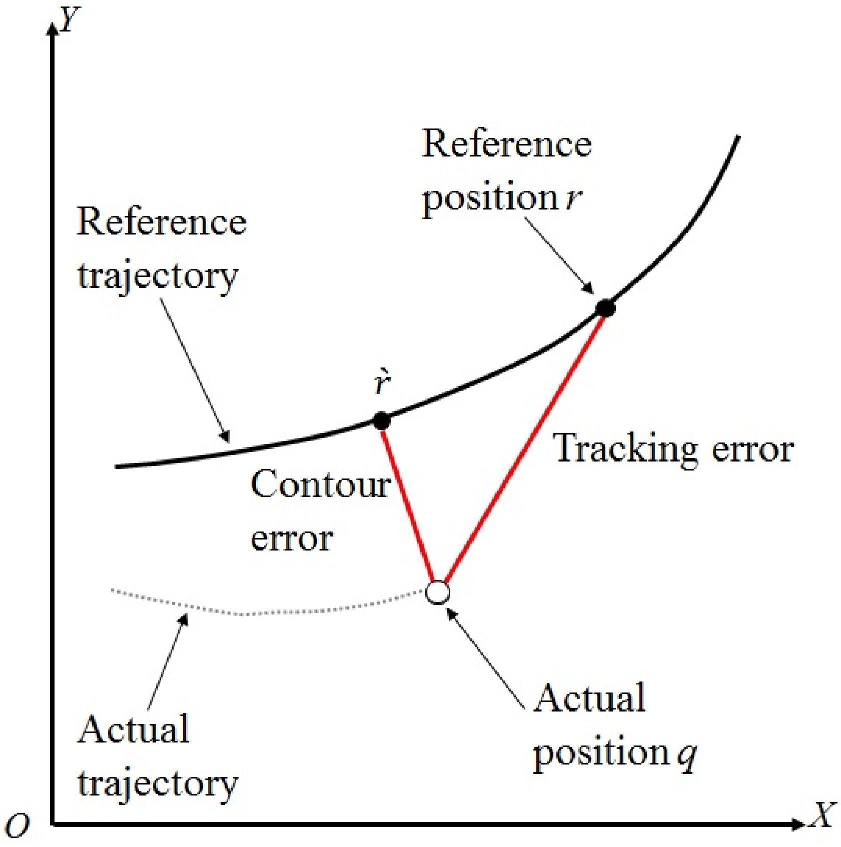


FIGURE 1.4: Tracking and contour error.

number of points; nevertheless, the geometry is well defined because of the mechanical definition of the shapes. However, the NC program generated by computer-aided manufacturing (CAM) systems is usually based on small linear interpolations. As a result, a double negative effect is introduced. First, the toolpath is not exact, although the deviation between the toolpath and the exact curve can be controlled by the CAM software. Second, the sequence of linear interpolations results in a non-smooth toolpath which can result in drive axis instability and finally, in machining vibration [8]. Many control techniques have been developed to solve this specific problem in machine tool position control.

A cross-coupled controller was proposed to improve high-speed contouring accuracy independently of tracking accuracy and contouring performance in biaxial machine tool feed drive servomechanisms. This attribute of cross-coupled controllers is particularly useful in applications such as machining where the reduction of the contour error is far more critical than the reduction of the tracking error [18]. Accordingly, many cross-coupled methods have been developed to control the contouring performance under high feed rates such as a cross-coupled adaptive feed rate, and cross-coupled ILC (CCILC) [19]–[21].

During cross-coupled motion control, increasing the contour feed rate may result in larger contour errors. In order to increase feed rate and productivity without sacrificing the contouring performance, the cross-coupled adaptive feed rate based on a linearly perturbed model was proposed [22]. CCILC takes this further by applying the technique of ILC to a cross-coupled controller. It enables learning of the cross-coupled error which leads to a modified control signal and subsequent improvements in the contour trajectory tracking performance [23]. However, there are three major limitations. First, the CCILC iteratively modifies the control input signal that requires full access to the existing controller for redesigning or reprogramming, which makes it difficult to implement to commercial CNC machines in which access to the controller is generally not allowed. Second, the CCILC requires a greater number of iterations to reduce contour error in higher curvature trajectories because the simple contour error estimation does not provide sufficient estimation performance of the actual contour error. Third, commercial CNC machines are generally already equipped with well-tuned feedback controllers, and hence, they should not need to be replaced for machines currently in use. These three limitations will be addressed by the proposed ILC in this thesis.

By applying an ILC, the convergence of the output error is guaranteed under certain conditions even when the system parameters are not known exactly or are under the existence of bounded unknown external disturbances [24], [25]. ILC research has attracted many researchers in the machine tool community because it can be used to effectively improve machine tool performance. This thesis proposes a new pattern of ILC, which will be described in the next section.

1.1.3 Iterative Learning Control

ILC is an approach to improving the transient response performance of systems that operate repetitively over a fixed time interval. It is useful for problems in which a system must be able to follow different types of inputs, in the face of design or modeling uncertainty. The concept of learning control was motivated by observing the behavior of systems that operate repetitively. Another motivation for the approach of ILC arises from problems in which a system must have the capability to accurately respond to several different types of inputs [26].

The concept of iterative learning for generating the optimal input to a system was first introduced by Uchiyama [27]. The idea was later developed by Arimoto and his co-workers [28]–[40]. Fig. 1.5 illustrates the basic idea. Each time the system operates, its input and output signals ($u_k(t)$ and $y_k(t)$, respectively) are stored in memory (some type of memory device is implicitly assumed in the block of Fig. 1.5 labeled "Learning Controller"). The learning control algorithm then evaluates the performance error $e_k(t) = y_d(t) - y_k(t)$, where $y_d(t)$ is the desired output of the system. On the basis of the error signal, the learning controller then computes a new input signal $u_{k+1}(t)$, which is stored for use during the next trial. The next input command is chosen in such a way as to guarantee that the performance error will be reduced on the next trial [26].

The important task in the design of a learning controller is to find an algorithm for generating the next input in such a way that the performance error is reduced on successive trials. This is usually quantified by saying that the error should converge, with convergences measured in the sense of some norm. The learning control algorithm causes convergence of the error without knowing the model plant under control (or, at least, it should require minimal knowledge of the system parameters). Furthermore, the algorithm should be independent of the functional form of the desired

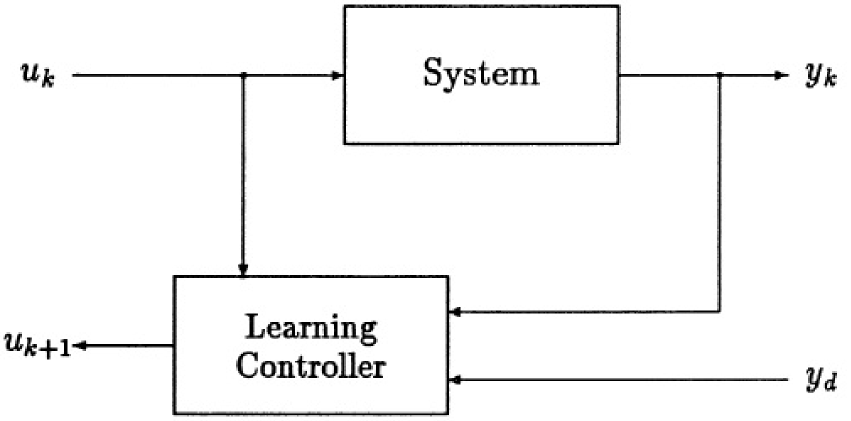


FIGURE 1.5: Learning control configuration [26].

response $y_d(t)$. Thus, the learning controller would “learn” the best possible control signal for a particular desired output trajectory without any need to reconfigure the algorithm. Then, if a new desired trajectory is introduced, the learning controller would simply “learn” the new optimal input, without changing any of its own algorithms, by using input–output data obtained during actual operation [26].

It is important to note that ILC differs from both optimal control and adaptive control. In optimal control, *a priori* design is conducted, based on a model of the system. If the plant changes relative to the model, then the controller will no longer be optimal (although adaptive LQR/LQG control algorithms can be used). On the other hand, if the plant changes in a learning control scheme, the learning controller adapts by adjusting the input for the next trial based on the measured performance error of the current trial. However, ILC is different from conventional adaptive control. Most adaptive control schemes are on-line algorithms that adjust the controller’s parameters until a steady-state equilibrium is reached. In a learning control scheme, it is the commanded reference input that is varied (in an off-line fashion), at the end of each trial or repetition of the system [26].

On the basis of the ILC principle, many researchers have developed ILC for effective implementation in machine tools. One such ILC is based on an iterative estimation of the instantaneous curvature of the reference trajectory and coordinates transformation approach to reduce the estimated contour error in a high-curvature trajectory [41]. In a different approach, the contour error can be significantly reduced by an ILC that considers both the tracking and contour errors [42]. Another ILC improves the accuracies of high speed, computer-controlled machining processes by considering uncertain dynamics such as high-frequency dynamics, actuator saturations, and dry friction [43]. All of these various ILC methods utilize conventional ILC, which assumes to have full access to the concerned system controller [43]–[48]. Normally, commercial systems do not provide full access to the controller, only allowing the adjustment of controller gains or other common parameters, along with operational access. A cascade form is needed for implementing ILC in commercial CNC machine tools. This thesis proposes a special cascade ILC for commercial machine tool applications. The principle cascade ILC will be explained in the next section.

Iterative learning controllers can be constructed in five different ways [49], as shown in the figures below.

- Previous Cycle Learning (PCL)

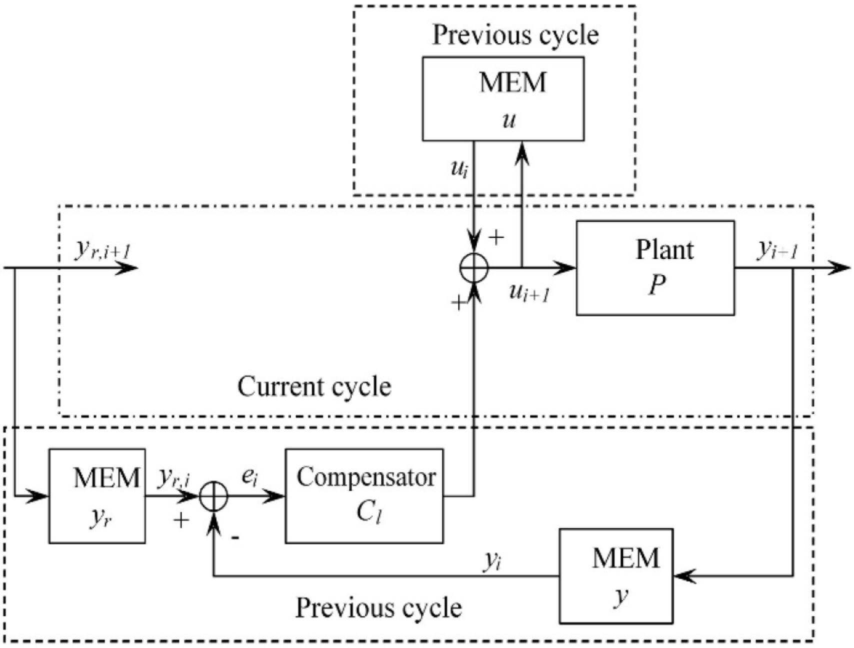


FIGURE 1.6: The schematic of PCL [49].

- Current Cycle Learning (CCL)

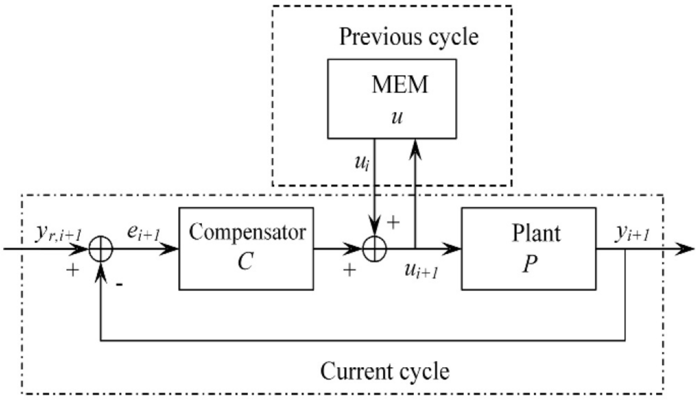


FIGURE 1.7: The schematic of CCL [49].

- Previous and Current Cycle Learning (PCCL)

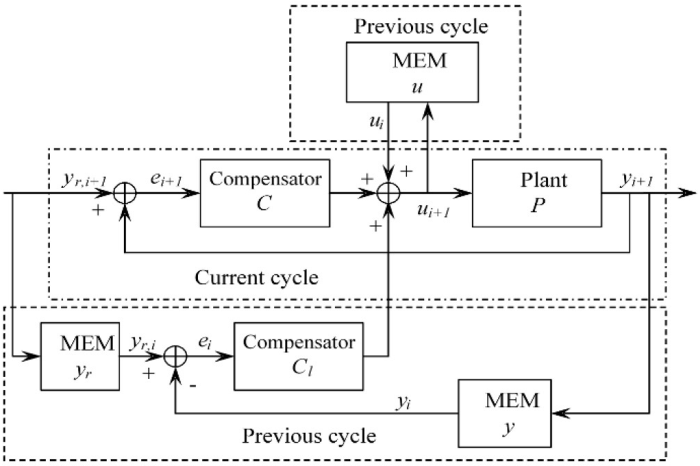


FIGURE 1.8: The schematic of PCCL [49].

- Cascade ILC

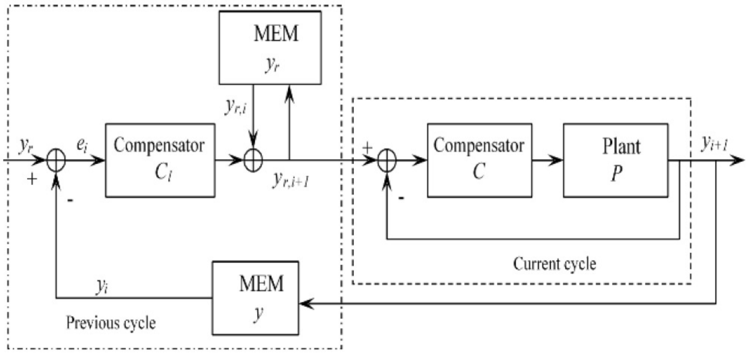


FIGURE 1.9: The schematic of cascade ILC [49].

- Incremental Cascade ILC

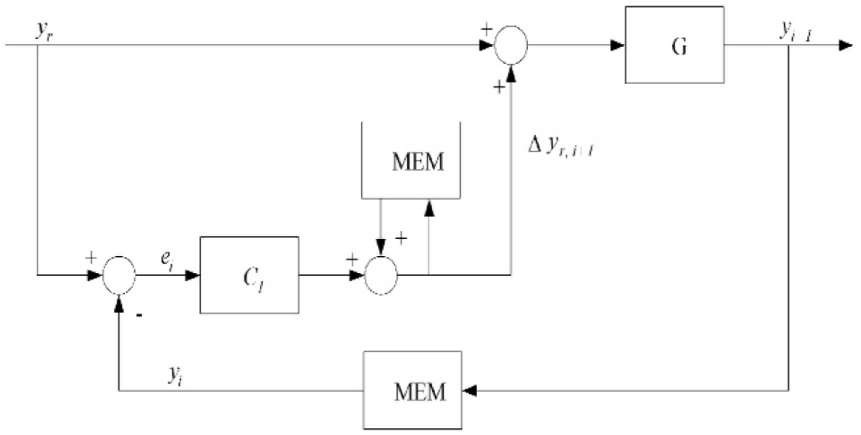


FIGURE 1.10: The schematic of incremental cascade ILC [49].

1.1.4 Cascade Iterative Learning Control

The three ILC schemes (Figs. 1.6, 1.7, and 1.8) need access to the existing control system for redesigning the control system, with a new feed-forward component to the system input channel. A new control block is embedded into the control loop. Such an embedded structure is the common structure for most of the existing real-time ILC schemes. Hence, when an ILC mechanism is to be incorporated into an existing control system, either the core execution program needs to be rewritten or the microcontroller chip needs to be replaced. In many real applications, the reconfiguration of a commercial controller is unacceptable because of the cost, security, and intellectual property constraints. For instance, a rapid thermo-processing device in the wafer industry costs millions of dollars, and the only tunable part is a number of set points. In such a circumstance, the cascade learning method is suitable as it modifies only the reference trajectory iteratively to improve the control performance [49].

The schematic of such an ILC is demonstrated in Fig. 1.9. It can be seen that the ILC block is "cascaded" to the existing control loop or performed as an outer feedback loop in the iteration domain. The ILC with the cascade structure will use the modified reference signals and the actual system output of the previous cycle to generate the new reference signals for the current cycle. Owing to the cascade structure, the ILC need not be embedded into the existing control loop, thus avoiding any reconfiguration of the system hardware or the core execution program. What is needed is essentially some reprogramming of reference signals, which can be easily carried out in real applications [49].

As shown in Fig. 1.9, the learning control law can be expressed as follows:

$$\begin{aligned} Y_i &= GY_{r,i}, && (1.1)\\ E_i &= Y_r - Y_i, && (1.2)\\ Y_{r,i+1} &= Y_{r,i} + C_l E_i, && (1.3)\\ Y_{r,0} &= Y_r, && (1.4) \end{aligned}$$

where $G = PC/1(1+PC)$ denotes the closed-loop transfer function; Yr is the original reference repeated over a fixed operation period; and Yr,i is the reference signal modified via learning for the inner current cycle control loop. According to the learning control law (Eq. (1.3)), the convergence condition for the cascade ILC can be derived as

$$\begin{aligned} E_{i+1} &= Y_r - Y_{i+1}, && (1.5)\\ &= Y_r - GY_{r,i+1}, && (1.6)\\ &= Y_r - G(Y_{r,i} + C_l E_i), && (1.7)\\ &= Y_r - GY_{r,i} - GC_l E_i, && (1.8)\\ &= (1 - GC_l)E_i && (1.9)\\ \frac{E_{i+1}}{E_i} &= 1 - GC_l, && (1.10)\\ \left\| \frac{E_{i+1}}{E_i} \right\| &= \left\| 1 - \frac{PCC_l}{1+PC} \right\| \leq \gamma < 1. && (1.11) \end{aligned}$$

In most cases the cascade ILC is of the PCL type, because set points, once selected, cannot be changed in the midst of a real-time operation process [49].

This thesis proposes ILC by considering contour error. It can be directly applied in commercial machining by a cascade scheme without any modification of the existing controller in CNC machine tools. The principle of the proposed work is the modification of the original trajectory as NC codes form iteratively by contour error correction to improve tracking performance until no further significant reduction of contour error occurs. To improve controller performance, two kinds of contour error estimation, namely, approximated contour error estimation (Chapter 3) and actual contour error estimation (Chapters 4 and 5), are used in the proposed ILC to improve contour error in low- and high-curvature trajectories, respectively. In this study, the proposed ILC is implemented in a laboratory biaxial feed drive system (Chapter 3), a commercial three-axis CNC machine tool with a ball screw mechanical system (Chapter 4), and a CNC machine tool with a linear motor mechanism (Chapter 5).

The proposed thesis has three main advantages. First, the proposed controller is designed by a cascade scheme, which enables it to be implemented in commercial CNC machines currently in use. Because the existing methods require the access to the embedded controller for modifying control input, they cannot be applied in commercial CNC machines. The feedback controller of an industrial machine is generally already well-tuned. Second, the proposed contour error estimation can estimate precise contour error for higher curvature trajectories. Most methods of contour error estimation cannot represent the actual contour error. This proposed method can estimate the actual contour error precisely. Third, a commercial CNC machine already has a well-tuned controller, which has good stability and disturbance rejection properties; unlike the feedback-based approach, the proposed approach is based on feed-forward control by an iterative NC program/G-code modification, which is expected to improve the tracking performance. In other words, this proposed controller can be easily applied to commercial CNC machinery.

The proposed research is suitable for mass production applications because it may be desirable to provide dummy material in the beginning of production to reduce the contour error in the iterative learning contouring controller (ILCC). However, there was also a study on feed drive control that mentioned that feed drive motion accuracy mostly affects the precision of products because typical industrial machines have sufficient robustness to a disturbance such as cutting force. Therefore, once the reference trajectory for feed drive motion is optimized, it may be still effective for finishing the cutting process [50]. In typical heavy cutting cases, after rough cutting, finishing cutting is conducted. In that case, the rough cutting does not need high accuracy. Once the final reference trajectory is optimized by the proposed ILC in the "air-cutting" process, the optimized trajectory can be applied for the finishing cutting process, which enables the production of a high-precision product.

However, the proposed method has two inherent difficulties, those being learning gain optimization and commercial CNC machine plant identification. The proposed method requires well-tuned learning gain to achieve the best controller performance. Convergence analysis is used to optimize learning gain by the "fmincon" function in MATLAB®. Convergence analysis requires precise plant transfer function information for learning gain optimizing. All commercial CNC machine transfer functions are unknown to the user. Thus, it will require accurate machine plant identification, but the precise machine plant identification process is still challenging for many researchers.

1.2 Motivation and Research Objectives

On the basis of the description of the problem and a review of the relevant literature, the research objective is defined as follows. This research aims to develop an ILC for a commercial CNC machine to improve machine performance. Many control techniques have been proposed, most of them requiring either full access to the existing control system or the design of new control system. This is difficult to implement in commercial CNC machine tools because almost all commercial CNC machine tools do not provide full access to the original controller due to cost, security, and intellectual property constraints. An ILC with a cascade schematic form is proposed to improve machine performance by modifying the reference trajectory (NC code) with contour error estimation. This ILC can be implemented in commercial CNC machine tools without modification of the embedded controllers or existing machine controllers. Finally, the proposed control system will be implemented in a biaxial feed drive system, a commercial CNC machine with a ball screw mechanism, and a CNC machine tool with a linear motor mechanism.

In order to increase the accuracy of machine tools, an error correction based on contour error estimation and the PID principle is embedded in the proposed ILC. Most contour error estimations do not well represent the actual contour error. They produce low accuracy contour error estimations that are then used for error correction. An error correction with low accuracy estimation will not result in a significant improvement in machine performance. This research proposes a precise contour error estimation that consists of two parts. First, the contour error is estimated by multiplying a tracking error by a rotational matrix; this can work properly for a low-curvature trajectory but is not effective for high-curvature trajectories. Second, an actual contour error is estimated by calculating the minimum distance between the actual position and the reference trajectory to achieve the desired contour error precisely.

1.3 Thesis Contribution

This thesis presents advanced research about ILC with contour error estimation for commercial CNC machine tools. The precise contour error estimation will also contribute to improve machine tool accuracy or performance. The experimental results show the effectiveness in reducing contour error. By reducing a contour error, machine tool performance will be improved and a higher quality product can be produced. The main contributions are listed as follows:

(1) There are difficulties in implementation of many control techniques to commercial CNC machinery due to cost, security, and intellectual property constraints. Most of the control techniques require full access to the existing controller for redesigning, reprogramming, or modifying the current CNC controller. It takes a substantial investment and a complicated production system to manufacture a new CNC control system (software) in a new CNC machine tool (hardware) for many of the already proposed control techniques. This thesis proposes ILC with cascade form to improve machine performance in CNC machine tools. The proposed ILC modifies the reference trajectory based on contour error estimation with a PID learning pattern without modifying the existing CNC controller. The modified reference trajectory is generated in a G-code form, so it can be executed by the CNC machine tools directly. The PID learning gain is defined by convergence analysis to guarantee that the contour error reduces iteratively. Finally, the proposed ILC is implemented properly to several machine tools: a biaxial feed drive system, a three-axis commercial CNC machine tool with a ball screw mechanism, and a commercial CNC machine tool with a linear motor mechanism.

(2) An iterative learning controller based on contour error estimation is proposed to achieve precision machine tool motion. A contour error estimate is calculated by two algorithms. First, a contour error is calculated by multiplying the tracking error by a rotational matrix to achieve a candidate contour error. However, it can be used directly to modify a reference trajectory by ILC for low-curvature trajectories. A high-curvature trajectory requires a more accurate contour error estimation, which is developed in the next algorithm. Next,

an actual contour error is estimated by calculating the minimum distance between an actual position and a desired trajectory. To increase the accuracy of the contour error estimation, linear and circular interpolations are used between an actual position, the closest point, and the neighborhood around the closest point on the desired trajectory. The precise contour error estimation is demonstrated by simulation and experiment in commercial CNC machine tools.

(3) Overall, the ILCC can be effectively implemented on commercial CNC machine tools. Normally, machine tool performance declines over the life of the equipment, but a used commercial CNC machine tool's performance can be enhanced by ILCC implementation without any modification of the existing CNC controller. The transfer function of the machine tool is needed to calculate the PID learning gain by convergence analysis to achieve the optimum contour error correction. The required information can be obtained through system identification.

1.4 Thesis Outline

The main research in this thesis is about iterative learning reference trajectory modification for contouring performance improvement of industrial CNC machine tool feed drive systems. The rest of this thesis is organized as follows:

- **Chapter 2** introduces the proposed model of ILCC for commercial CNC machine tools and experimental devices that are used in this research. There are three types of feed drive systems: a biaxial feed drive system, a three-axis commercial CNC machine tool with a ball screw mechanism, and a commercial CNC machine tool with a linear motor mechanism. In addition, the physical system configuration and system parameters are explained in this chapter.

- **Chapter 3** describes the ILCC-based trajectory generation with estimated contour error for a biaxial feed drive system. A controller design is shown, which implements the proposed method for a biaxial feed drive system. Convergence analysis is explained also to define the PID learning gain. This method is demonstrated by simulation and experiment with a sharp-corner trajectory that has always generated a high contour error in a feedback controller, normally.
- **Chapter 4** improves upon the previous method by the actual contour error estimation with linear interpolation and the Bézier reposition trajectory (BRT) to further improve the contouring performance. This method is implemented in a three-axis commercial CNC machine tool with a ball screw mechanism. The simulation and results suggest that the proposed method works properly to improve machine performance.
- **Chapter 5** describes the implementation of the updated method in a commercial CNC machine with a linear motor mechanism. Circular interpolation is designed to improve the contouring performance. The system identification of the experimental system and the convergence analysis are explained also to define the PID learning gain. This system is demonstrated by simulation and experiment with rhomboidal and circular trajectories.
- **Chapter 6** summarizes the whole research and highlights the important points. It also identifies important points that can be improved in order to provide guidance for potential future works.

The general outline of all chapters is shown in Fig. 1.11. The research problem definition, research motivation, research objective, thesis contribution, and thesis structure are explained in Chapter 1. It is followed by Chapter 2, which describes the proposed model and all of the experimental devices, including their physical properties and system parameters for implementation. Chapter 3 explains the first research implementation. The proposed ILC and common contour error estimation are applied to a simple biaxial feed drive system. The proposed method is proved by simulation

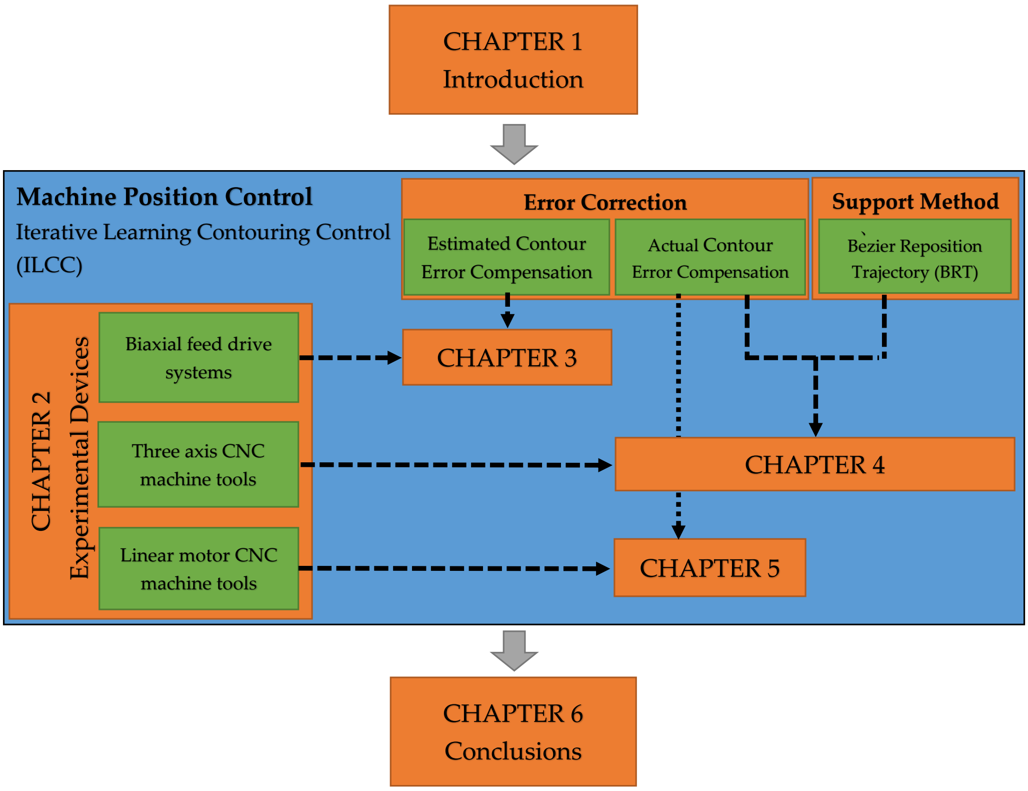

CHAPTER 1
Introduction
Machine Position Control
Iterative Learning Contouring Control (ILCC)
Error Correction
Estimated Contour Error Compensation
Actual Contour Error Compensation
Support Method
Bèzier Reposition Trajectory (BRT)
CHAPTER 2
Experimental Devices
Biaxial feed drive systems
Three axis CNC machine tools
Linear motor CNC machine tools
CHAPTER 3
CHAPTER 4
CHAPTER 5
CHAPTER 6
Conclusions

FIGURE 1.11: Thesis outline.

and experimental processes to validate controller performance. The contour error estimation is extended to precise actual contour error estimation in Chapter 4. For performance analysis, the proposed controller and new contour error estimation are implemented in a commercial three-axis CNC machine with a ball screw mechanism (a common industrial CNC machine tool). The proposed method is challenged with implementation in an advanced CNC machine tool with a linear motor servomechanism in Chapter 5. Finally, the whole research and future works are summarized in Chapter 6.

Chapter 2

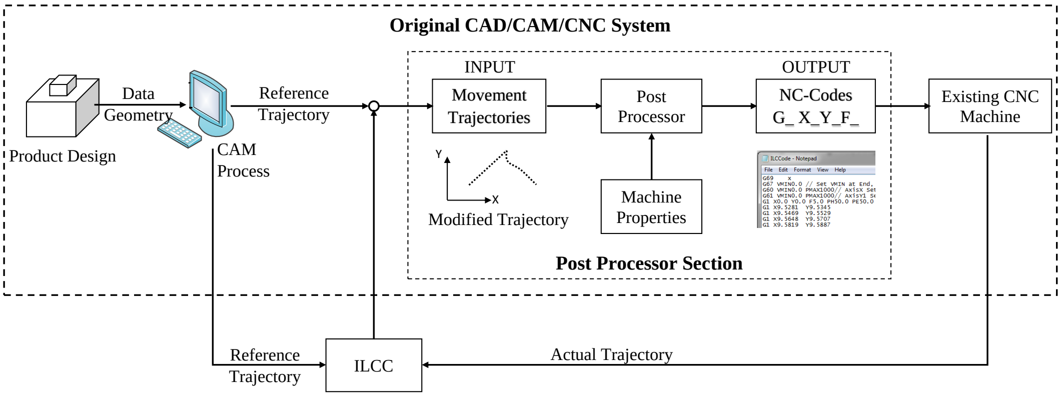
Preliminaries

2.1 Introduction

Due to flexibility in manufacturing, CAD/CAM/CNC system is developed. The two advanced computer-aided technologies that have been extensively used to help companies operate product design to product machining are CAD and CAM [51]. The CAD is the technology concerned with the use of computer systems to assist in the creation, modification, analysis, and optimization of design [52]. CAD software are developed to design product in wide area. The output CAD software is product design such as engineering drawing, product analysis, and simulation which are manufactured by production area. All of CAD output can be used for guide line in manufacturing process. Many idea, concept, problem solution, or product demand can be realized become product design with a detail specification. Some proving is needed to examine a product design will work properly or solve the problem. It can be analyzed by computer-aided engineering (CAE) to predict product performance before production process. The CAM process is the next step after product design which is done by CAD and CAE process as production preparation.

In manufacturing process applications of computer numerical control (CNC) machine tools, a product design is needed as a data source to generate the toolpath in the CAM software. CAM is the technology concerned with the use of computer systems to plan, manage, and control manufacturing operations through either a direct or indirect computer interface with the plant's production resources [52]. The machining conditions and operation are defined on the basis of product shape by the CAM software. The output of CAM is a tool trajectory, including the machining parameters, which is to be executed by the CNC machine tool(s). The tool trajectory is converted to G-code form by a post-processor. A tool trajectory in G-code form is ready to execute in CNC machining.

The last step is the machining process executed by the CNC machine. Many different activities may be required during the machining process, such as clamping of the workpiece and cutting tool, cutting tool position reference setup, and program preparation. The output of the machining process is the product itself. Many factors are required to produce a high quality product in this step. The important factor is the machining conditions, which have two aspects, the machining parameters and a tool trajectory. Each of them can generate error in the product directly. An abnormal cutting process can occur if the machining parameters are wrong, e.g., too rapid feeding or spindle rotation, not the right cutter type, and too deep a cutter depth. A vibration during machining can occur under these conditions. A high vibration during machining causes damage to the cutting tool or workpiece. The other potential effects include an un-smooth product surface or an imperfect shape of the product. Moreover, the tool trajectory can affect the product shape directly; this problem is not controllable by the operator during machining because the small defect will only be known after the machining process has been completed. Generally, the original CAD/CAM/CNC model is an open loop as shown in Fig. 2.1. The original CAD/CAM/CNC system cannot guarantee a certain quality of product due to the potential for some error during machining. This research proposes to apply a closed-loop scheme in the original CAD/CAM/CNC system by the ILC principle. This method is proven to reduce contour error by simulation and experiment. This



```
G69    x
G67 VMIN0.0 // Set VMIN at End,
G60 VMIN0.0 PMAX1000// AxisX Set
G61 VMIN0.0 PMAX1000// AxisY1 Se
G1 X0.0 Y0.0 F5.0 PH50.0 PE50.0
G1 X9.5281  Y9.5345
G1 X9.5469  Y9.5529
G1 X9.5648  Y9.5707
G1 X9.5819  Y9.5887
```

FIGURE 2.1: General structure of ILCC.

chapter explains the general design of ILCC and experimental systems.

2.2 General Design of ILCC

This research proposes an ILCC as shown in Fig. 2.1. This structure provides a closed-loop model on a global scope in the CAD/CAM/CNC system. The ILCC modifies the original trajectory through a learning controller based on a contour error compensation. A CNC machine executes the modified trajectory for better performance than that under the original trajectory. The process is done iteratively until no further significant reduction of the error is observed. The proposed controller can be directly applied to commercial machines currently in use by modifying NC codes without modification of the embedded controller.

Many contour error compensations can be applied to the proposed ILCC due to the effectiveness of error reduction. This thesis proposes two types of contour error compensations, an estimated error compensation and an actual error compensation, which are explained in Chapters 3 and 4, respectively. Both types of error compensation require actual position data to calculate a final contour error, and measurement devices with characteristics such as a linear scale, a rotary encoder, and non-contact measurement are required in order to obtain actual trajectory information during the machining process. The accuracy of the measuring devices will affect the performance of the ILCC system directly.

Because the research objective is to develop an iterative learning contouring control for commercial CNC machine tools to improve machine performance, the proposed method was implemented in three types of machine tools as experimental devices: a biaxial feed drive system, a three-axis CNC machine tool with a ball screw mechanism, and a CNC machine tool with a linear motor mechanism. The physical properties of these devices and the experimental design are explained in the next section.

2.3 Experimental Systems

2.3.1 Biaxial Feed Drive System

Biaxial feed drive systems are commonly applied in commercial industrial machines such as EDM wire cutting machines, water jet machines, laser cutting machines, and CNC punching machines, as shown in Fig. 2.2. These machines have a fundamental movement pattern based on X–Y axis movement, which is driven to track a certain desired trajectory. The output of biaxial movement is a two-dimensional trajectory without any specific vertical motion in the Z direction, so it usually produces a planar product such as a plate with a hole or any form in the inner or outer sides.

All of the machines that are used in this research to represent biaxial feed drive systems are shown in Fig. 2.3. An X–Y table with a feedback controller implemented by C++ language on a personal computer (Windows OS) was used as shown in Fig. 2.4. The control input signal was applied to a biaxial feed drive system with two lead screws driven by two DC servo motors via a DA board (CONTEC DA12-8 (PCL)). The actual position of the X–Y table was measured by encoders with resolution 0.025 μm attached to each axis. The ILCC was programmed in MATLAB® on a separate

(a) EDM wire cutting machine. http://www.alltek.com.au/

(b) Water jet machine. http://www.flowwaterjet.com/

(c) Laser cutting machine. http://www.goldenlaser.cc/

(d) CNC punching machine. http://www.uk.trumpf.com/

FIGURE 2.2: Biaxial feed drive systems application.

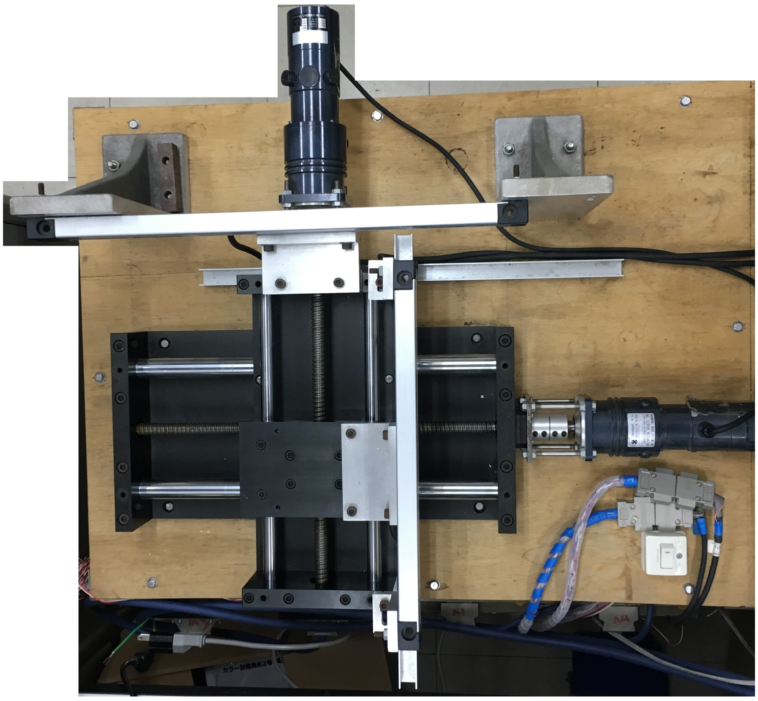
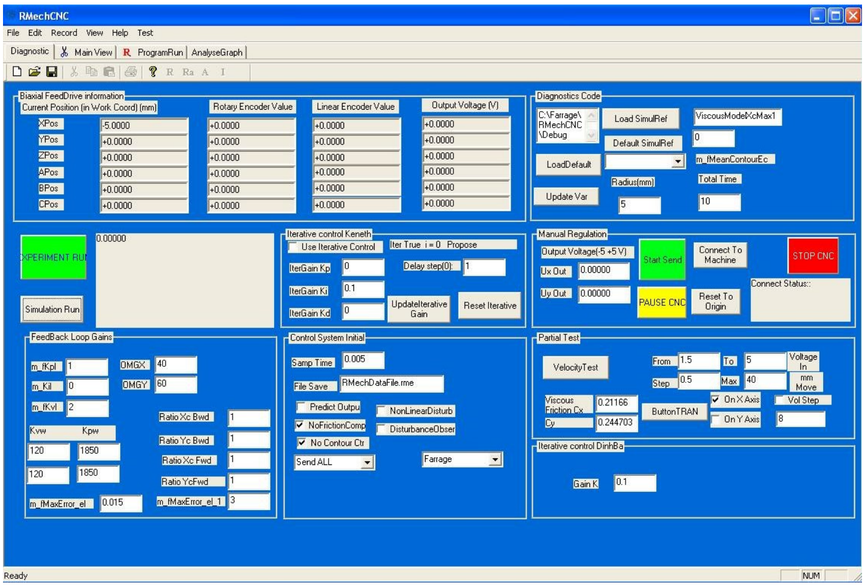


FIGURE 2.3: Laboratory biaxial feed drive system.



Biaxial FeedDrive information

Current Position (in Work Coord) (mm)		Rotary Encoder Value	Linear Encoder Value	Output Voltage (V)
XPos	-5.0000	+0.0000	+0.0000	+0.0000
YPos	+0.0000	+0.0000	+0.0000	+0.0000
ZPos	+0.0000	+0.0000	+0.0000	+0.0000
APos	+0.0000	+0.0000	+0.0000	+0.0000
BPos	+0.0000	+0.0000	+0.0000	+0.0000
CPos	+0.0000	+0.0000	+0.0000	+0.0000

Diagnostics Code

C:\Farrage\RMechCNC\Debug

ViscousModelXcMax1: 0

m_fMeanContourEc

Radius(mm): 5

Total Time: 10

0.00000

Iterative control Keneth

□ Use Iterative Control

Iter True i = 0 Propose

IterGain Kp: 0

Delay step(0): 1

IterGain Ki: 0.1

IterGain Kd: 0

Manual Regulation

Output Voltage(-5 +5 V)

Ux Out: 0.00000

Uy Out: 0.00000

Connect Status::

FeedBack Loop Gains

m_fKpl: 1

m_KiI: 0

m_fKvl: 2

OMGX: 40

OMGY: 60

Kvw	Kpw
120	1850
120	1850

Ratio Xc Bwd: 1

Ratio Yc Bwd: 1

Ratio Xc Fwd: 1

Ratio YcFwd: 1

m_fMaxError_el: 0.015

m_fMaxError_el_1: 3

Control System Initial

Samp Time: 0.005

File Save: RMechDataFile.rme

□ Predict Outpu

☑ NoFrictionComp

☑ No Contour Ctr

□ NonLinearDisturb

□ DisturbanceObser

Send ALL

Farrage

Partial Test

From: 1.5 To: 5 Voltage In

Step: 0.5 Max: 40 mm Move

Viscous Friction Cx: 0.21166

Cy: 0.244703

☑ On X Axis

□ On Y Axis

□ Vol Step

8

Iterative control DinhBa

Gain K: 0.1

FIGURE 2.4: Controller interface of biaxial feed drive systems.

personal computer (Windows OS) and embedded to the biaxial feed drive system as shown in Fig. 2.5. The physical specification and parameter systems are described in Table 2.1.

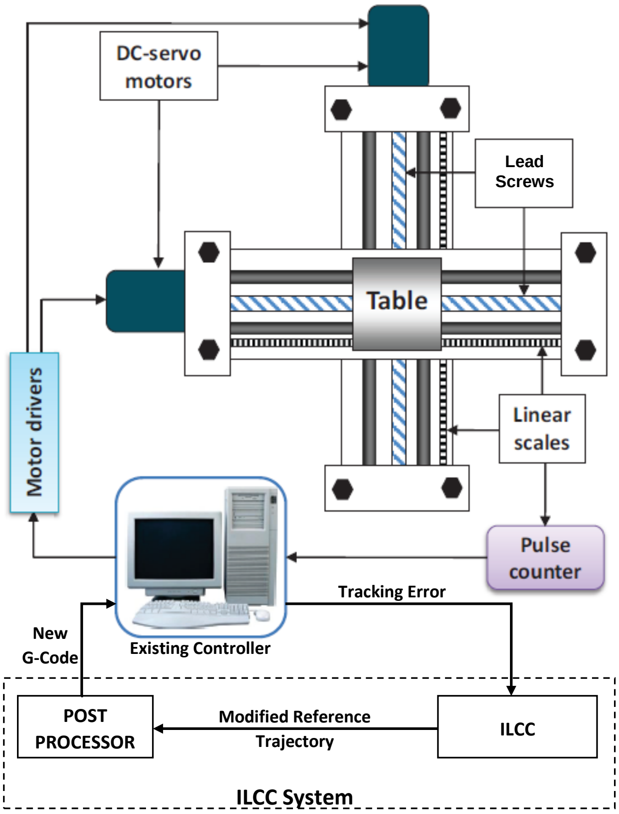
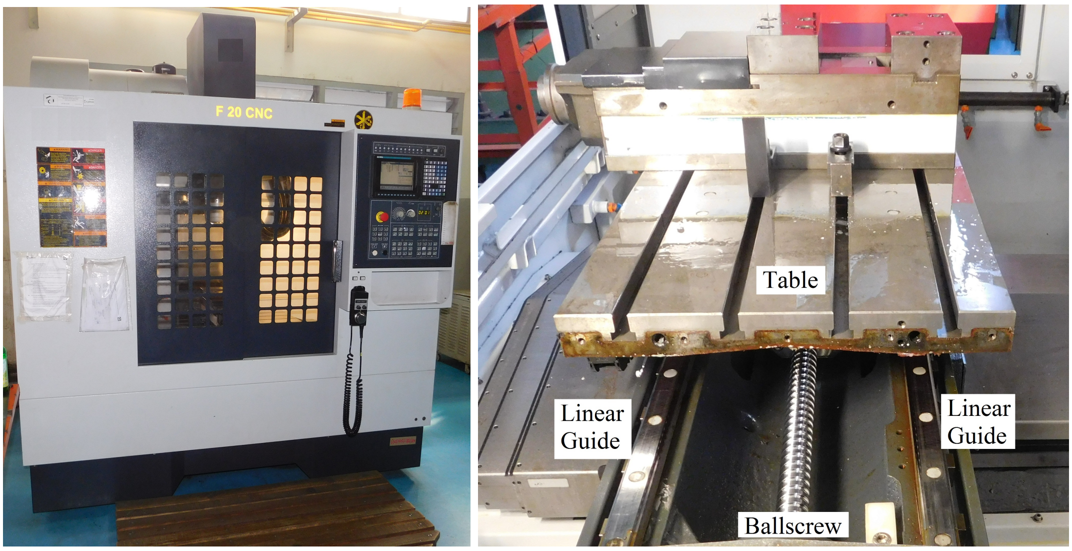


FIGURE 2.5: Biaxial feed drive system with ILCC system.

TABLE 2.1: Physical specification and parameter system of biaxial feed drive systems.

Specification			Value
Travel	X-axis	[mm]	240
	Y-axis	[mm]	210
Table	Table height	[mm]	160
	Table working surface	[mm]	110 x 110
Max Feedrate	X-axis	[m/min]	8.3
	Y-axis	[m/min]	8.3
Motor Power sources	X-axis	[W]	34
	Y-axis	[W]	34
Machine size	length	[mm]	800
	width	[mm]	700
Parameter			
m_i	X-axis	[Vs^2/mm]	0.055
	Y-axis	[Vs^2/mm]	0.055
c_i	X-axis	[Vs/m]	0.21
	Y-axis	[Vs/m]	0.21
K_P	X-axis	[V/mm]	0.96
	Y-axis	[V/mm]	0.96
K_I	X-axis	[V/smm]	0.01
	Y-axis	[V/smm]	0.01
K_D	X-axis	[Vs/mm]	8
	Y-axis	[Vs/mm]	8

The tracking error is collected by the encoders. The ILCC converts tracking error to contour error by contour error estimation (explained in Chapter 3). On the basis of an estimated contour error, ILCC modifies the reference trajectory. After the post-processor has converted the modified reference trajectory to a G-code program, the program is executed by a biaxial feed drive system. This algorithm is conducted iteratively until no further significant contour error reduction can be achieved.



(a) Three axis CNC machine (b) CNC table mechanism.

FIGURE 2.6: Experimental machinery.

2.3.2 Three Axis CNC Machine Tools

A three-dimensional product is typically produced by a three-axis CNC machine tool. Movement can occur on all three axes simultaneously to generate complex curvatures. Generally, a 3D trajectory is provided by CAM software based on the product

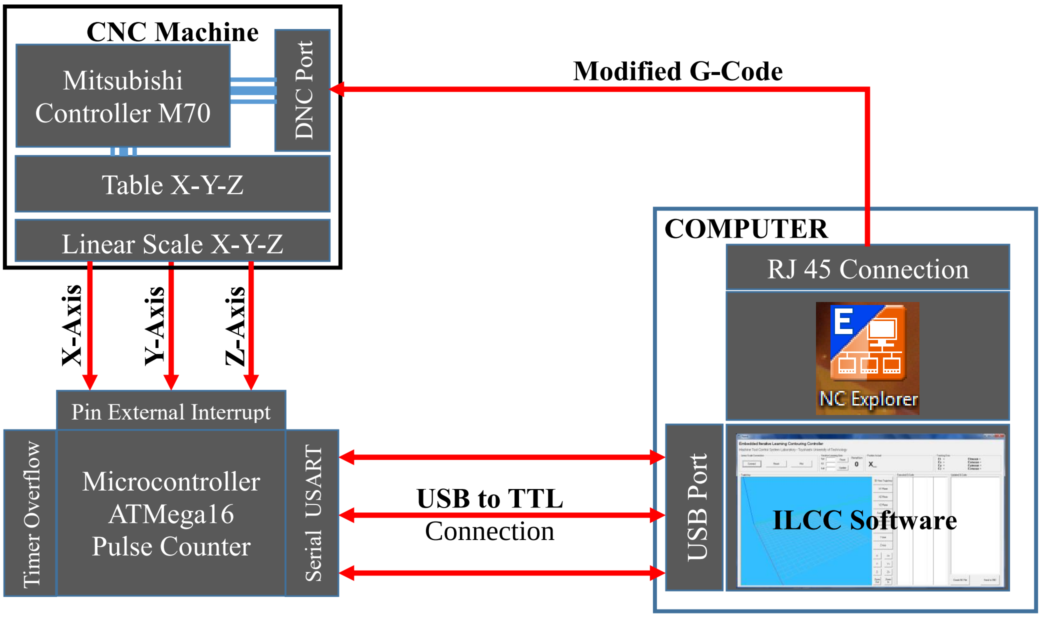


FIGURE 2.7: Experimental interface design.

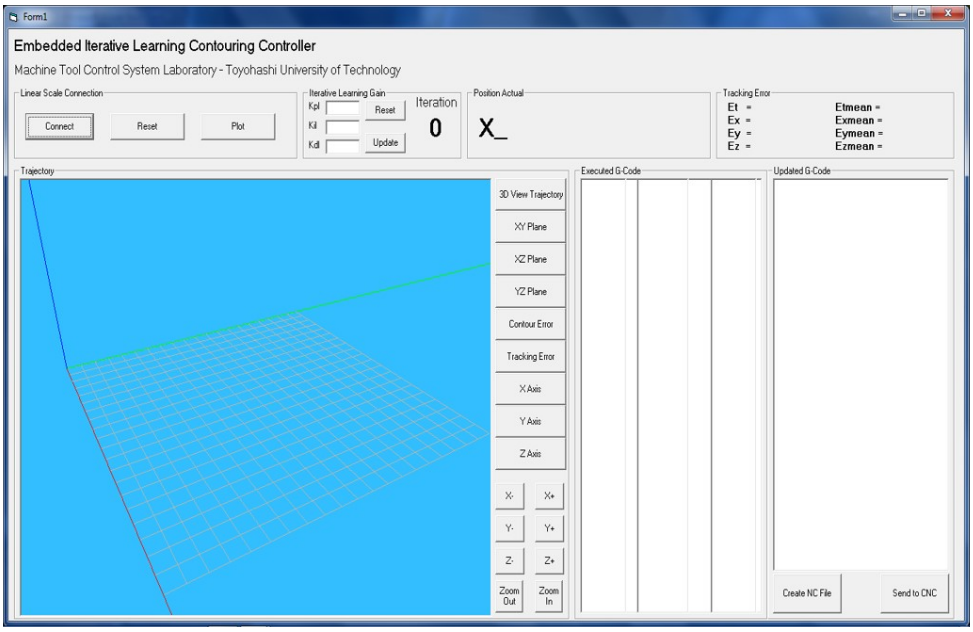
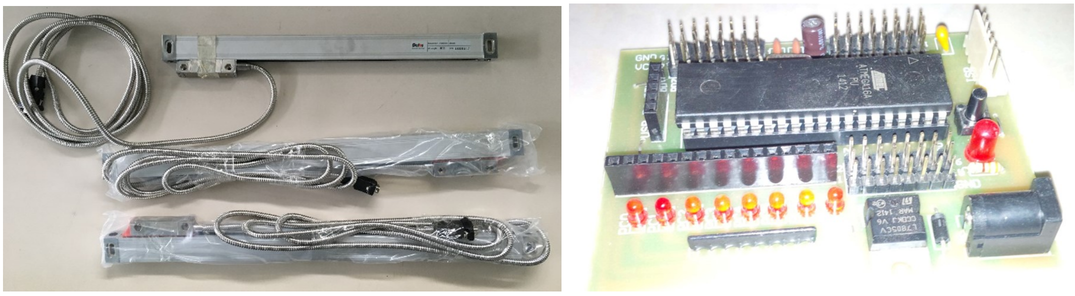


FIGURE 2.8: ILCC software interface.

design and machine conditions. In a three-axis CNC machine, all three axes can contribute to the production of contour error. This condition makes it more difficult to control error in machine tools. The ILCC can also be implemented in a commercial three-axis CNC machine for reducing contour error.

A three-axis commercial CNC machine (Fig. 2.6) with a ball screw mechanism was attached on a table with three servo motors. A Mitsubishi M70 controller was used for the experiment to verify the effectiveness of the proposed controller. The actual X–Y–Z position of the table was measured by linear encoders (Fig. 2.9 (a) and Table 2.3) with a resolution 5 μm. A five-precision bearing system was used in the head sliding of the linear scale. The linear scale grating system was a transmissive and infrared optical measuring system. The linear scale pulse was counted by a microcontroller ATMEGA 16 attached to each axis for the interface, as shown in Fig. 2.9 (b) and Table 2.4. The ILCC was programmed in Microsoft visual basic 6 on a separate personal computer (Windows OS), as shown in Fig. 2.8 and was implemented in the machine by a direct numerical control (DNC) system with an RJ45 connection. The experimental interface design and physical system parameters are shown in Fig. 2.7 and Table. 2.2, respectively.



(a) Linear scale.

(b) Microcontroller ATMega 16.

FIGURE 2.9: Measurement interface for the experiment.

The initial reference trajectory is executed by a CNC controller. The linear scale that is attached on each axis measures the actual position by transmitting the pulse count to the microcontroller as a pulse counter interface. Then, the ILCC software on the personal computer modifies the reference trajectory based on the calculated

TABLE 2.2: Physical specification and parameter system of three axis CNC machine tools.

Specification			Value
Travel	X-axis	[mm]	760
	Y-axis	[mm]	420
	Z-axis	[mm]	510
Table	Dimension	[mm]	890 x 420
	Max. Loading	[kgs]	3000
Max Feedrate	X-axis	[m/min]	36
	Y-axis	[m/min]	36
	Z-axis	[m/min]	36
Spindle	Speed	[rpm]	10000
	Power	[kW]	7.5
Machine size	Length	[mm]	2100
	Width	[mm]	2230
	Height	[mm]	2370
Parameter			
m_i	X-axis	$[\times 10^{-4}\text{kgm}^2]$	9.62
	Y-axis	$[\times 10^{-4}\text{kgm}^2]$	13.99
	Z-axis	$[\times 10^{-4}\text{kgm}^2]$	11.3
c_i	X-axis	[kg/s]	0.24
	Y-axis	[kg/s]	0.24
	Z-axis	[kg/s]	0.24

TABLE 2.3: Specification of linear scale.

Specification		Value
Model		DLS-W
Length	[mm]	400
Zero Ref. point	[mm]	50
Accuracy	[μm]	5
Resolution	[μm]	5
Working temperature	[°C]/[°F]	20 / 68
Max working speed	[m/min]	60
Voltage	[V]	5
Infrared wave length	[nm]	880
Sealing protection		IP55
Output signal		TTL
Head sliding		5 precision bearing system

contour error. The CNC controller then executes the modified trajectory via the DNC system. The NC explorer software is used to support the DNC system by transferring the G-code program from the personal computer to the CNC controller. This entire process is conducted iteratively until no further significant contour error reduction occurs. The research that is explained in Chapter 4 uses this experimental device to demonstrate the effectiveness of the proposed method.

TABLE 2.4: Specification of microcontroller.

Specification		Value
Model		ATMega 16
Speed	[MHz]	16
Operating voltage	[V]	4.5-5.5
Operating range	[°C]	-40 to 85
EEPROM	[Bytes]	512
SRAM	[Kbyte]	1
Cycle	[Flash]	10000
I/O	[I/O]	31
Power		1.1 mA
Features		Peripheral features Two 8-bit timer/counter One 16-bit timer/counter RTC with separate oscillator External & internal interrupt source Programmable serial USART USART serial communication

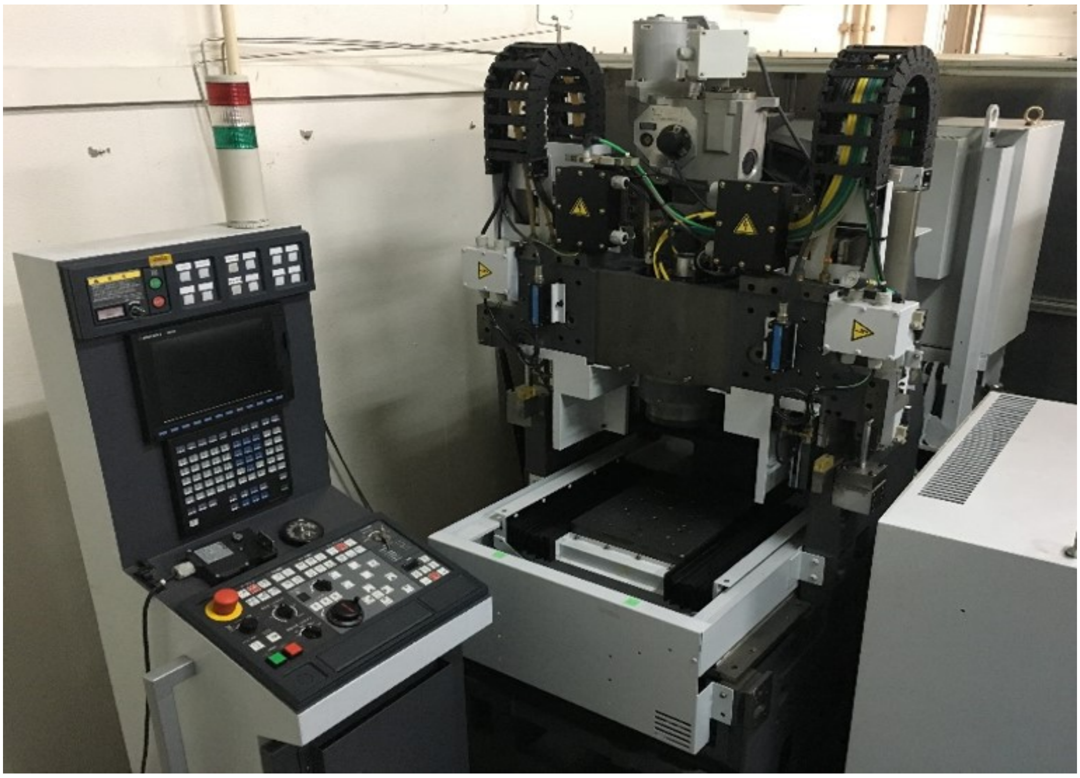


FIGURE 2.10: CNC Machine DMG MORI NV1500.

2.3.3 CNC Machine Tool with Linear Motors

The current machine tool technology can achieve 1 μm accuracy in a linear motor application. Many advantages are shown by linear motor performance in comparison with rotary motors. Many transmission errors can occur in a rotary motor such as backlash and screw, gear, and coupling errors. In contrast, no mechanical transmission is needed in a linear motor application, so higher accuracy can be achieved.

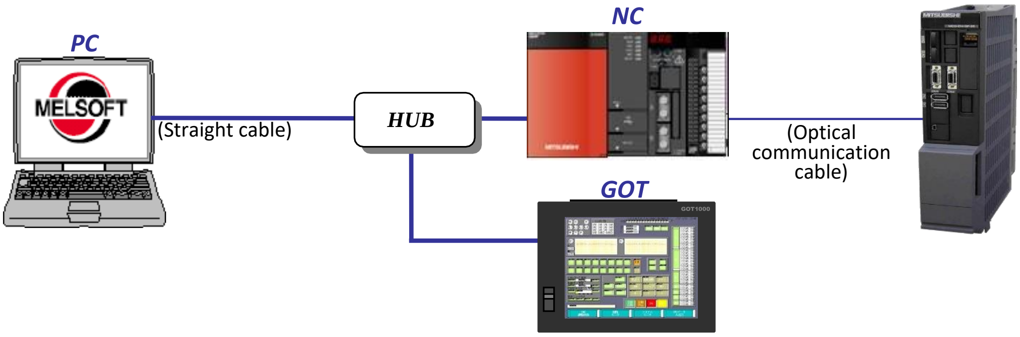
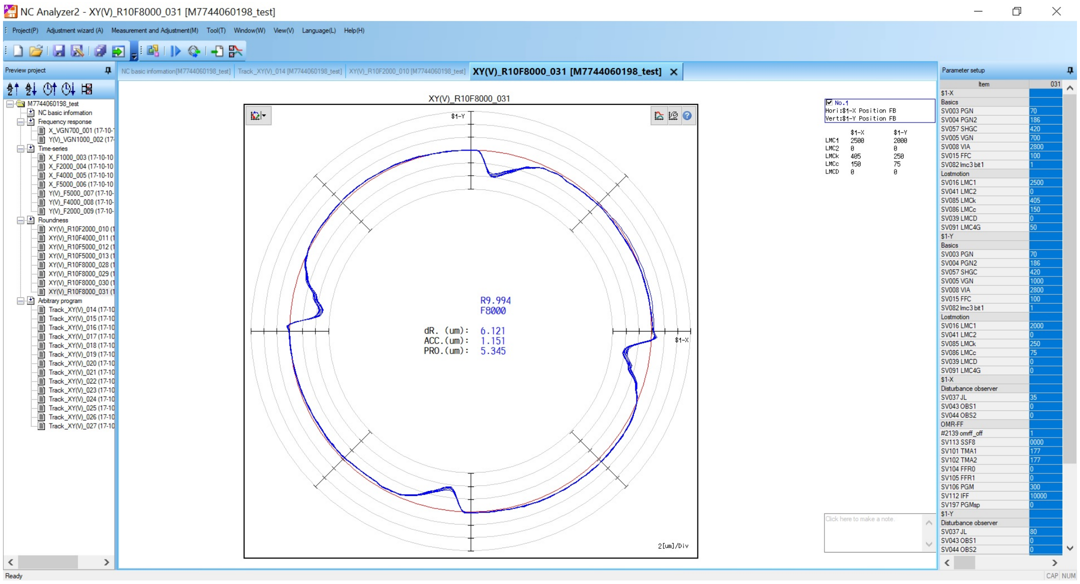


FIGURE 2.11: Experimental interface design for NV1500 [53].



R9.994
F8000

dR. (um): 6.121
ACC.(um): 1.151
PRO.(um): 5.345

☑ No.1
Hori:$1-X Position FB
Vert:$1-Y Position FB

	$1-X	$1-Y
LMC1	2500	2000
LMC2	0	0
LMCk	405	250
LMCc	150	75
LMCD	0	0

Item	031
$1-X	
Basics	
SV003 PGN	70
SV004 PGN2	186
SV057 SHGC	420
SV005 VGN	700
SV008 VIA	2800
SV015 FFC	100
SV082 lmc3 bit1	1
Lostmotion	
SV016 LMC1	2500
SV041 LMC2	0
SV085 LMCk	405
SV086 LMCc	150
SV039 LMCD	0
SV091 LMC4G	50
$1-Y	
Basics	
SV003 PGN	70
SV004 PGN2	186
SV057 SHGC	420
SV005 VGN	1000
SV008 VIA	2800
SV015 FFC	100
SV082 lmc3 bit1	1
Lostmotion	
SV016 LMC1	2000
SV041 LMC2	0
SV085 LMCk	250
SV086 LMCc	75
SV039 LMCD	0
SV091 LMC4G	0
$1-X	
Disturbance observer	
SV037 JL	35
SV043 OBS1	0
SV044 OBS2	0
OMR-FF	
#2139 omrff_off	1
SV113 SSF8	0000
SV101 TMA1	177
SV102 TMA2	177
SV104 FFR0	0
SV105 FFR1	0
SV106 PGM	300
SV112 IFF	10000
SV197 PGMsp	0
$1-Y	
Disturbance observer	
SV037 JL	80
SV043 OBS1	0
SV044 OBS2	0

FIGURE 2.12: Melsoft NC analyzer.

For idle movement, linear motor performs high acceleration and velocity. With high velocity in machining applications, the machine tool must use a low feed rate and high speed cutting ("baby cutting") method because of the machine's structural limitations.

TABLE 2.5: Physical specification and parameter system of the CNC machine tool DMG MORI NV1500.

Specification			Value
Travel	X-axis	[mm]	150
	Y-axis	[mm]	150
	Z-axis	[mm]	200
Table	Dimension	[mm]	230 x 220
	Max. Loading	[kgs]	3000
Max Feedrate	X-axis	[mm/min]	15000
	Y-axis	[mm/min]	15000
	Z-axis	[mm/min]	15000
Spindle	Speed	[rpm]	24000
	Power	[kW]	5.5
Machine size	Length	[mm]	1935
	Width	[mm]	850
	Height	[mm]	2092
	Mass	[kg]	2500
Input resolution		[μm]	0.1

The CNC machine tool DMG MORI NV1500 (Fig. 2.10) with a linear motor controlled by the Mitsubishi M700V was used for the experiment to verify the effectiveness of the proposed method, as described in Chapter 5. The actual position was measured with Melsoft NC Analyzer 2 software (Fig. 2.12) with an RJ45 connection and a resolution of 1 *n*m. The ILCC was programmed in MATLAB® on a separate personal computer (Windows OS) and implemented in the machine by DNC with the NC Analyzer 2 software. The experimental interface structure is shown in Fig. 2.11. The sampling rate was 1.7 ms. The system's physical specifications are listed in Table 2.5. The method described in Chapter 4 was extended to the CNC machine tool with a linear mechanism application to achieve the best performance in advanced machine tool technology.

Chapter 3

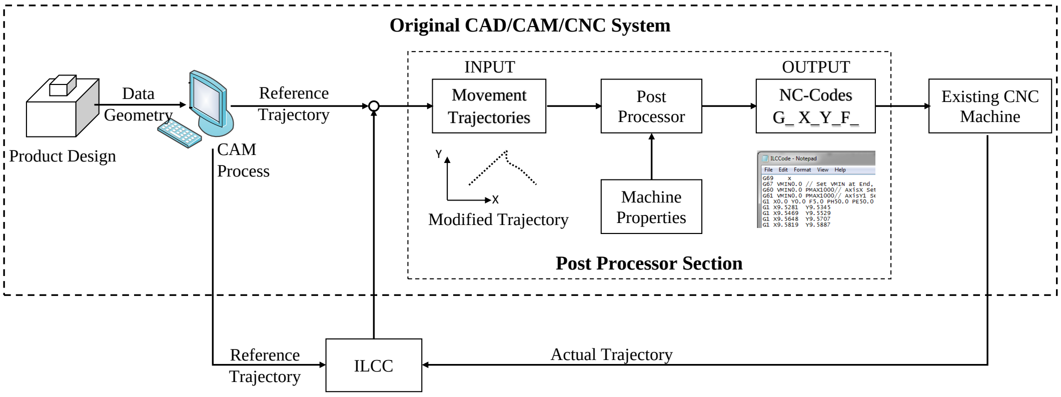
Iterative Learning Based Motion Trajectory Generation

In machine tool performance, a fundamental factor is an axial movement which is driven to track a desired trajectory. Not only tracking errors in each drive axis but also contour errors, which are directly related to the machined shape of a workpiece, should be considered. Although most existing contouring controllers are based on feedback control, this chapter proposes an iterative learning contouring controller (ILCC) by considering both tracking and contour errors. The proposed control iteratively modifies the reference trajectory of each drive axis to reduce the contour error. The proposed controller can be directly applied to commercial machines currently in use without any modification of their original controllers. The proposed method has been experimentally verified through a biaxial feed drive system on a sharp-corner trajectory which normally leads to a large contour error around the corner due to the discontinuity. Comparison with a conventional iterative learning contouring controller (CILCC) was done so as to evaluate its performance. Experimental results have shown that the contour error converges within a few iterations, and the maximum contour error can be reduced by about 49.2 % as compared to the CILCC.

3.1 Introduction

Feed drive systems such as biaxial feed drive systems are commonly applied in commercial industrial machines such as water jet machines, laser cutting machines, CNC punching machines, EDM wire cutting machines, etc. All have a fundamental movement based on X-Y axis movement which is driven to track a particular desired trajectory. The tracking ability is generally not so perfect that it may result in position errors [54]. However, the error has to comply with an allowable tolerance for producing required products. Achieving high precision in machining highly depends on control performance of each axis as well [55]. The common approach is to design an independent controller for each drive axis based on feedback control of the tracking error. However, the motion contour to achieve the desired shape of the workpiece is normally complex, whereby drive axes have to move in a synchronous manner with one another to obtain the desired contour. Under independent axial controllers, load disturbance or performance variance of either axis causes contour error [56]. Many control techniques have been developed in the past few decades to reduce both the tracking and contour errors.

Many cross-coupled methods are improved to control the contouring performance by coupling the individual axis error under high-feed-rate such as cross-coupled adaptive feed rate, and a cross-coupled iterative learning control (ILC) [18]–[23]. Moreover, the optimal contouring control deals with the evaluation of a cross-coupled compensator aimed specifically at improving the contouring accuracy in multi-axial feed drives by minimizing the weights of the contour error explicitly [57] and a novel cross-coupling control design by adjusting reference trajectory [58]. By applying an ILC, the convergence of the output error is guaranteed under certain conditions even when the system parameters are not known exactly or under the existence of bounded unknown external disturbances [24], [25]. Furthermore, an iterative contouring controller minimizes the contour error through an iterative estimation of the instantaneous curvature of the reference trajectory and coordinates transformation [41]. In



```
G69    x
G67 VMIN0.0 // Set VMIN at End,
G60 VMIN0.0 PMAX1000// AxisX Set
G61 VMIN0.0 PMAX1000// AxisY1 Se
G1 X0.0 Y0.0 F5.0 PH50.0 PE50.0
G1 X9.5281  Y9.5345
G1 X9.5469  Y9.5529
G1 X9.5648  Y9.5707
G1 X9.5819  Y9.5887
```

FIGURE 3.1: Proposed ILCC concept.

addition, the contour error can be significantly reduced by an ILC which considers both the tracking and contour errors [42].

All of the above mentioned and other ILC methods consider conventional iterative learning control (CILC) which assumes to have full access to the concerned system controller [43]–[48]. Normally, commercial systems do not provide full access to the controller except the adjustment of control gains. In addition, a few existing methods based CILC consider either tracking or contour error [59], [60].

ILCC method is proposed by considering both the tracking and contour errors for machine tool feed drive system. The contour error is estimated from the tracking error and a rotational matrix. Then, the original reference trajectory is iteratively modified by the ILCC through a learning compensator. The modified trajectory is executed by CNC machine for better performance than that under the original trajectory. The process is done iteratively until when no significant reduction of the contour error is observed. Referring to controller proposed in [42], the conventional iterative learning contouring controller (CILCC) is used as baseline for comparison, and the ILCC was proven to provide better performance than the CILCC by simulation [61]. This chapter explains convergence analysis using the contour error, simulation, and experimental verification on a biaxial feed drive system. The proposed controller can be directly applied to commercial machines currently in use by modifying NC codes without modification of the embedded controllers as shown in Fig. 3.1. Furthermore, by using this new controller, the contour error, directly controlled and reduced.

The rest of this chapter is arranged as follows: Section 3.2 gives a brief description of the dynamics of machine tool feed drive systems, tracking error and contour error definition, the design of the proposed controller, and convergence analysis. Simulation results which compare the proposed and conventional methods are described in Section 3.3. Experimental results and discussion are given in Section 3.4 followed by concluding remarks in Section 3.5.

3.2 ILCC for Biaxial Feed Drive Systems

3.2.1 Dynamic Model of Biaxial Feed Drive Systems

The dynamics of a biaxial feed drive system is represented as follows:

$$\begin{aligned} &M\ddot{q} + C\dot{q} = u, \\ &M = \mathrm{diag}\{m_i\},\ C = \mathrm{diag}\{c_i\}, \quad i = x,\ y, \\ &q = [q_x,\ q_y]^{\mathrm{T}},\ u = [u_x,\ u_y]^{\mathrm{T}}, \end{aligned} \tag{3.1}$$

where m_i, c_i, q_i, and u_i are the inertia, viscous friction coefficient, actual position, control voltage for the axis i, respectively.

3.2.2 Contour Error Estimation

The tracking error in each axis is defined as the difference between the desired and actual positions, while the contour error is the error components orthogonal to the desired contour curves [9]. The description of both the tracking and contour errors are shown in Fig. 3.2. The desired position of the feed drive system at time t in the coordinate frame Σ_w are denoted as r, while the actual position is denoted as q. The closest position of the desired contour to q is denoted by $\grave{r}$. The tracking error in each feed drive axis is defined as

$$e_w = [e_x,\, e_y]^{\mathrm{T}} = q - r. \tag{3.2}$$

$$e = \|e_w\| \tag{3.3}$$

The tracking error vector $e_l(t)$ with respect to $\sum_l$ can be expressed as

$$e_l = [e_t,\, e_n]^{\mathrm{T}} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} e_w, \tag{3.4}$$

where θ is the inclination of the coordinate frame Σ_w to Σ_l. Based on Eq. (3.4) and the assumption that desired trajectories are usually finely discretized with respect to sampling time, such that the curvature between two discrete points are relatively very

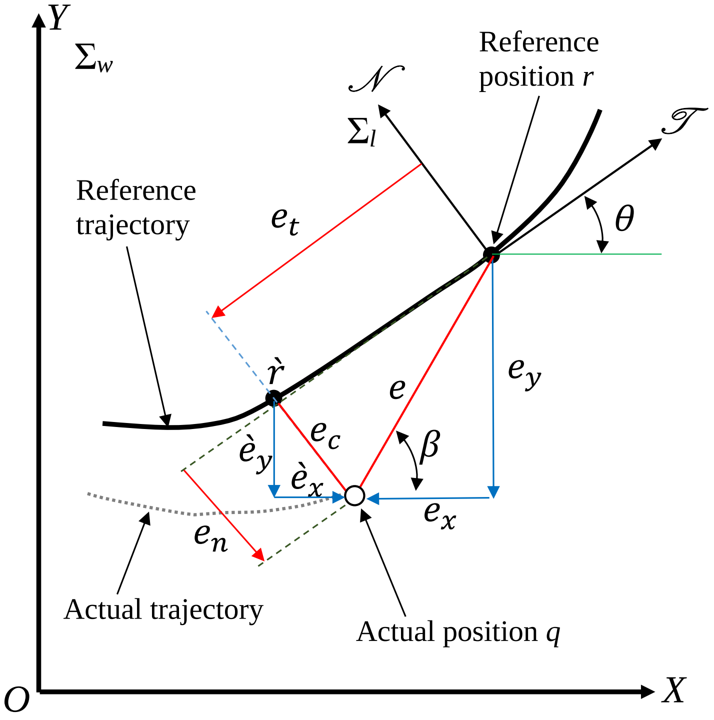


FIGURE 3.2: Tracking and contour errors.

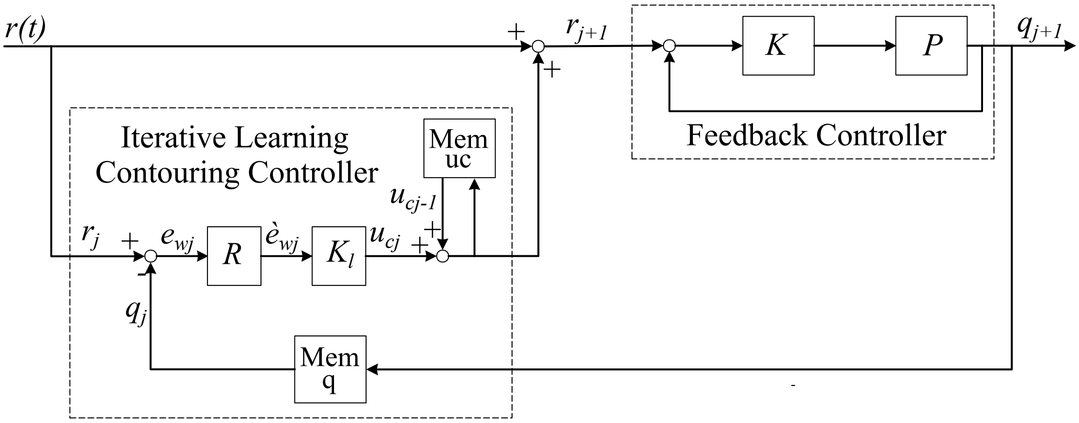


FIGURE 3.3: Iterative learning contouring controller design: Mem u_c and Mem q are the memories of learning compensation values and actual position, respectively.

small. Hence, the proposed method is applicable to high curvature trajectories. The contour error e_c can be assumed to be equal to e_n and represented as

$$e_c \approx e_n = -e_x \sin\theta + e_y \cos\theta. \tag{3.5}$$

The axial elements of the contour error $\grave{e}_x$ and $\grave{e}_y$ are given by

$$\grave{e}_w = \begin{bmatrix} \grave{e}_x \\ \grave{e}_y \end{bmatrix} = Re_w,$$
$$R = \begin{bmatrix} -\sin^2\theta & \sin\theta\cos\theta \\ -\sin\theta\cos\theta & \cos^2\theta \end{bmatrix}. \tag{3.6}$$

$$e_c = ||\grave{e}_w|| \tag{3.7}$$

3.2.3 Controller Design

The proposed controller consists of a feedback controller (FBC) and an ILC compensator. The input of the FBC is the modified reference trajectory which is compensated by the ILC to form an ILCC as in Fig. 3.3. The transfer function of the plant including the FBC is represented as follows:

$$G_i\left(s\right) = \frac{P_i\left(s\right)K_i\left(s\right)}{1+P_i\left(s\right)K_i\left(s\right)}, \tag{3.8}$$

where G_i, P_i, K_i, and s are the system transfer function, plant (feed drive system), feedback compensator, and variable of the Laplace transform for i^{th} axis, respectively. The FBC is the traditional PID, therefore $K_i(s)$ is as follows:

$$K_i\left(s\right) = K_{Pi} + \frac{1}{s}K_{Ii} + sK_{Di}, \tag{3.9}$$

where K_{Pi}, K_{Ii}, and K_{Di} are the proportional, integral, and derivative gains for i^{th} axis, respectively.

The learning compensator K_{li} is designed in a similar manner as K_i,

$$K_{li}\left(s\right) = K_{Pli} + \frac{1}{s}K_{Ili} + sK_{Dli}, \tag{3.10}$$

and for every iteration j, the reference trajectory is given as

$$r_{ij+1}\left(t\right) = r_{ij}\left(t\right) + u_{cij}\left(t\right), \tag{3.11}$$

$$u_{cij}\left(t\right) = K_{li}\left(s\right)\grave{e}_{wij}\left(t\right), \tag{3.12}$$

$$u_{cij}\left(t\right) = K_{Pli}\grave{e}_{wij}\left(t\right) + K_{Ili}\int_0^t \grave{e}_{wij}\left(\tau\right)d\tau + K_{Dli}\dot{\grave{e}}_{wij}\left(t\right), \tag{3.13}$$

therefore,

$$r_{ij+1}(t) = r_{i1}(t) + \sum_{n=0}^{j} u_{cij-n}(t), \tag{3.14}$$

where K_{PIi}, K_{IIi}, and K_{DIi} are the proportional, integral, and derivative gains for the learning compensator for i^{th} axis, respectively. r_{ij+1} is the reference signal for the $j+1^{\text{th}}$ iteration for i^{th} axis. r_{ij} is the reference signal for the j^{th} iteration for i^{th} axis. $\grave{e}_{wij}$ is the contouring error in the j^{th} iteration for i^{th} axis. r_{i1} is the initial reference trajectory. u_{cij-n} is contouring compensated input from initial iteration n to the j^{th} iteration for i^{th} axis. The FBC is assumed to be stable because it comes with the commercial machine. Therefore, the focus is kept on the contour error convergence of the ILCC system as explained in the following section.

3.2.4 Convergence Analysis

The convergence condition of the ILC is represented as

$$\varepsilon_i(t) = \frac{e_{cij+1}(t)}{e_{cij}(t)} < 1, \tag{3.15}$$

where $\varepsilon_i(t)$, and e_{cij} and e_{cij+1} are the convergence factor, and the contour errors (e_c) in the j^{th} and $j+1^{\text{th}}$ iteration for i^{th} axis, respectively. Based on the cascade ILC [62], the convergence can be guaranteed if the error magnitude at time t in iteration $j+1$ is smaller than the error magnitude in iteration j. The convergence condition can be derived for each axis by considering the components of the contour error in each axis. Substituting $\tan\beta = e_y/e_x$ to Eq. (3.6), leads to

$$\grave{e}_w = \grave{R}e_w, \tag{3.16}$$

where

$$\grave{R} = \begin{bmatrix} \tan\beta\sin\theta\cos\theta - \sin^2\theta & 0 \\ 0 & \cos^2\theta - \frac{\sin\theta\cos\theta}{\tan\beta} \end{bmatrix} = \begin{bmatrix} \grave{R}_x & 0 \\ 0 & \grave{R}_y \end{bmatrix},$$

$$\begin{aligned} \tan\beta = \frac{e_y}{e_x} &= \frac{r_y - q_y}{r_x - q_x}, \\ &= \frac{r_y - G_y(s)\, r_y}{r_x - G_x(s)\, r_x}, \\ &= \frac{r_y(1 - G_y(s))}{r_x(1 - G_x(s))}, \\ &= \frac{\|r\|\cos\theta(1 - G_y(s))}{\|r\|\sin\theta(1 - G_x(s))}, \\ &= \frac{(1 - G_y(s))\cos\theta}{(1 - G_x(s))\sin\theta}. \end{aligned} \tag{3.17}$$

where $\grave{R}$ and β are rotation matrix for convergence analysis and the inclination of the trajectory error vector e to the X-axis. By using $\grave{R}_i, (i = x, y)$, we can deal with each axis dynamics separately hereafter.

$$e_{wj+1} = r - q_{j+1}. \tag{3.18}$$

where e_{wj+1} and q_{j+1} are tracking error and actual position in the $j+1^{\text{th}}$ iteration, respectively. r is desired position unchanged for iteration. We have the following relation for each axis :

$$\begin{aligned}
\grave{R}^{-1}\grave{e}_{wj+1} &= r - G(s)\, r_{j+1}, \\
\grave{e}_{wj+1} &= \grave{R}\left(r - G(s)\left\{r_j + \grave{e}_{wj}K_{li}(s)\right\}\right), \\
&= \grave{R}\left\{r - r_j G(s) - K_l(s)\,\grave{e}_{wj}G_i(s)\right\}, \\
&= \grave{R}\left\{e_{wj} - K_l(s)\,\grave{e}_{wj}G_i(s)\right\}, \\
&= \grave{R}\left\{\grave{R}^{-1}\grave{e}_{wj} - K_l(s)\,\grave{e}_{wj}G_i(s)\right\}, \\
&= \grave{e}_{wj} - K_l(s)\,\grave{R}\grave{e}_{wj}G(s), \\
&= \left\{1 - \grave{R}K_l(s)\,G(s)\right\}\grave{e}_{wj}, \\
&= \left\{1 - \grave{R}\frac{P(s)\,K(s)\,K_l(s)}{1 + P(s)\,K(s)}\right\}\grave{e}_{wj}, \\
\frac{\grave{e}_{wj+1}}{\grave{e}_{wj}} &= \left\{1 - \grave{R}\frac{P(s)\,K(s)\,K_l(s)}{1 + P(s)\,K(s)}\right\}. \qquad (3.19)
\end{aligned}$$

where K_l, K, and P are the learning compensator, feedback compensator, and plant, respectively. $\grave{e}_{wj}$ and $\grave{e}_{wj+1}$ are contour error in the j^{th} and $j+1^{\text{th}}$ iteration, respectively. Then, by substituting Eq. (3.7) to (3.15), we have

$$\varepsilon(t) = \left\| \frac{\grave{e}_{wj+1}(t)}{\grave{e}_{wj}(t)} \right\| < 1 \qquad (3.20)$$

Substituting Eq. (3.19) to (3.20) leads to

$$\varepsilon(s) = \left\| 1 - \grave{R}\frac{P(s)\,K(s)\,K_l(s)}{1 + P(s)\,K(s)} \right\| < 1. \qquad (3.21)$$

Considering a discrete-time form for implementation, $P(s)$ is represented as follows:

$$P(z) = \frac{1}{m\left(\frac{z-1}{T_s z}\right)^2 + c\frac{z-1}{T_s z}}, \;\; z = e^{j\omega T_s}, \qquad (3.22)$$

where T_s and ω are the sampling time and the angular frequency, respectively. The feedback and learning compensator in the z-domain are represented as,

$$K(z) = K_P + K_I\left(\frac{T_s z}{z-1}\right) + K_D\left(\frac{z-1}{T_s z}\right), \tag{3.23}$$

$$K_l(z) = K_{Pl} + K_{Il}\left(\frac{T_s z}{z-1}\right) + K_{Dl}\left(\frac{z-1}{T_s z}\right). \tag{3.24}$$

The convergence speed depends on the parameter ε which should be kept as minimum as possible for fast convergence with a delay factor γ due to the system delay. Considering Eq. (3.21) and the delay, the following objective function for minimization is considered:

$$J = \min_{K_{Pl},\, K_{Il},\, K_{Dl}} \left\| 1 - \grave{R}\frac{z^{-\gamma}P(z^{-1})\,K(z^{-1})\,K_l(z^{-1})}{1+P(z^{-1})\,K(z^{-1})} \right\|_\infty \quad \forall\omega \in \Omega \text{ and } \forall\theta \in \Theta \tag{3.25}$$

where z^{-1} is a delay operator. Ω and Θ are the considered domains of the operational frequency and θ for a specific trajectory. K_{Pl}, K_{Il}, and K_{Dl} are the proportional, integral, and derivative learning compensator gains, respectively. Eq. (3.25) is solved by “fmincon” function in MATLAB® to find learning compensator gains.

3.3 Simulation

3.3.1 Simulation Condition

A typical biaxial feed drive system (Fig. 2.5) with $m_i = 0.055\,\mathrm{Vs^2/mm}$ and $c_i = 0.21\,\mathrm{Vs/mm}$ was considered for the simulation (Table 3.1). The simulation was done by the MATLAB® software under Windows OS, whereby the considered sampling rate was 5 ms. The initial reference trajectory was defined in a G-code form to create

TABLE 3.1: Plant and control parameters.

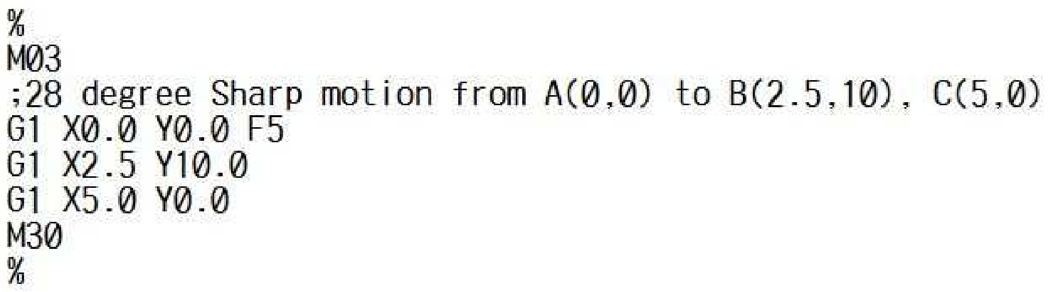
Parameter	X-Axis	Y-Axis
m_i [Vs^2/mm]	0.055	0.055
c_i [Vs/m]	0.21	0.21
K_P [V/mm]	0.96	0.96
K_I [V/smm]	0.01	0.01
K_D [Vs/mm]	8	8
CILCC	X-Axis	Y-Axis
K_{Pl} [V/mm]	0.4	0.4
K_{Il} [V/smm]	0.01	0.01
K_{Dl} [Vs/mm]	0.1	0.1
ILCC	X-Axis	Y-Axis
K_{Pl}	0.3	0.3
K_{Il} [1/s]	0.02	0.02
K_{Dl} [s]	0.5	0.5

a typical sharp-corner trajectory with an angle of 28° as shown in Fig. 3.4. The considered domain was $-76^\circ \leq \Theta \leq 76^\circ$ with a discretization of 1° and $0 \leq \Omega \leq 100\,\text{Hz}$ with a discretization of 1 Hz. The feedback compensator gains were chosen as shown in Table 3.1. Optimal gains for the learning compensator $(K_{Pl}, K_{Il}, \text{ and } K_{Dl})$ were obtained as shown in Table 3.1 by solving Eq. (3.25), which led to a convergence factor $\varepsilon_i = 0.603$.

The estimated contour error is calculated by Eqs. (3.2) - (3.7). According to the optimized learning gain and contour error estimation, contour error compensation is conducted by Eq. (3.13) to modify the reference trajectory as in Eq. (3.11). The modified reference trajectory is applied to the original feedback controller which is modeled by Runge-Kutta method with Couloumb friction as disturbance. The above process is done iteratively until no significant contour error reduction is observed.

3.3.2 Simulation Results

In order to evaluate the performance of the proposed controller, comparisons with the FBC and the CILCC were considered. The evaluation was done based on the



```
%
M03
;28 degree Sharp motion from A(0,0) to B(2.5,10), C(5,0)
G1 X0.0 Y0.0 F5
G1 X2.5 Y10.0
G1 X5.0 Y0.0
M30
%
```

FIGURE 3.4: G-code form for initial trajectory.

actual contour error as per its definition (not the approximated one in Eq. (3.4)). The simulation was conducted in 12 iterations for both the CILCC and the ILCC. Results are shown in Figs. 3.5-3.10, where Fig. 3.5 shows the convergence profiles of the maximum contour errors. Convergence of the contour error in ILCC is faster than that of CILCC, and there is no significant error reduction after the 9th iteration. Fig. 3.6 shows that the ILCC reduction ratio for each iteration increases continuously from the theoretical convergence factor, and the ILCC mean reduction ratio is 0.742, which is better than that of CILCC ($\varepsilon = 0.853$). Figs. 3.7 (a) and (b) show the actual trajectories for CILCC and ILCC, respectively. Figs. 3.10 (a) and (b) show the contour errors for CILCC and ILCC, respectively. In each figure, the first iteration refers to the FBC which has the largest contour error of 0.1212 mm. By applying

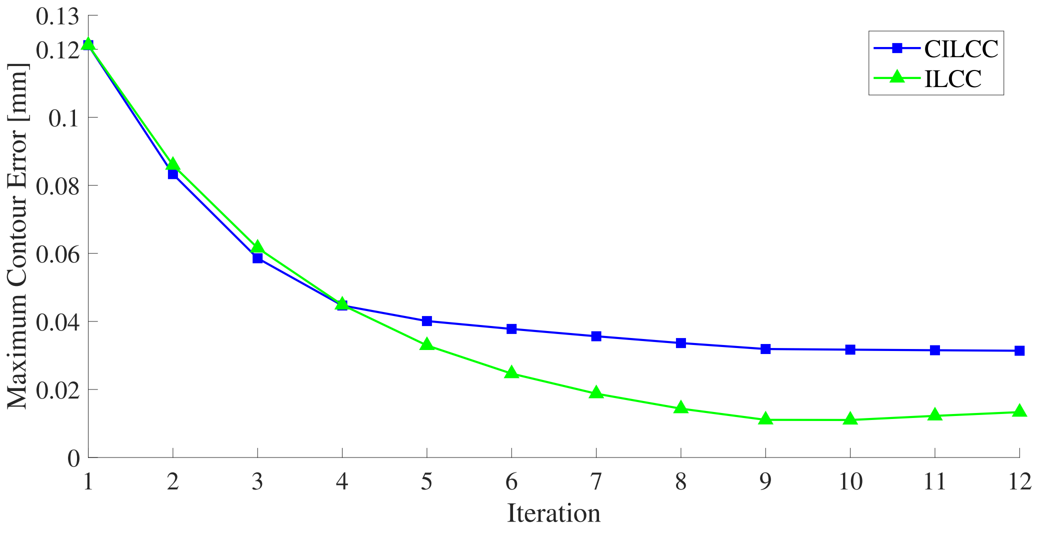


FIGURE 3.5: Simulation comparison for 12 iterations.

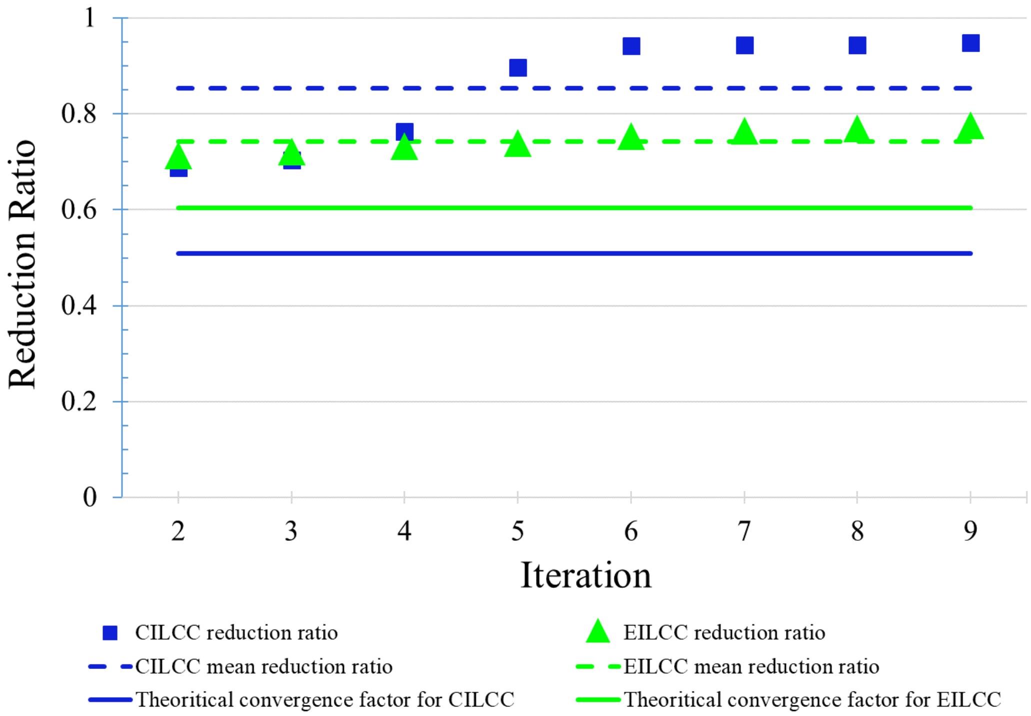
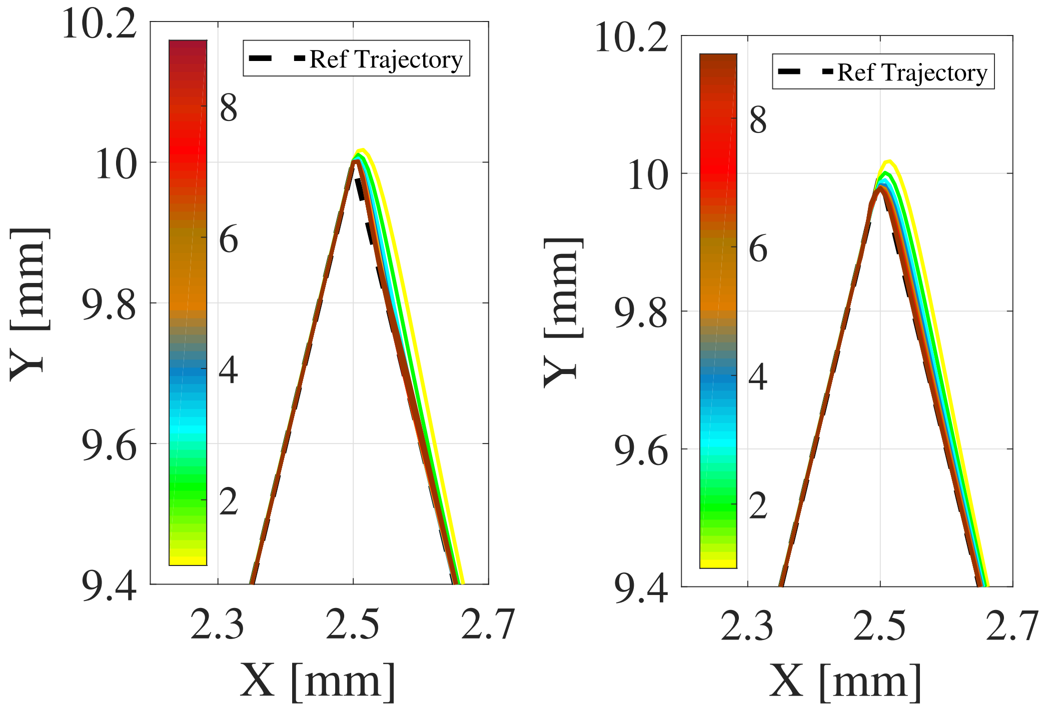


FIGURE 3.6: Convergence ratios in simulation and comparison to theoretical value.



(a) CILCC

(b) ILCC

FIGURE 3.7: Simulation result of actual trajectory .

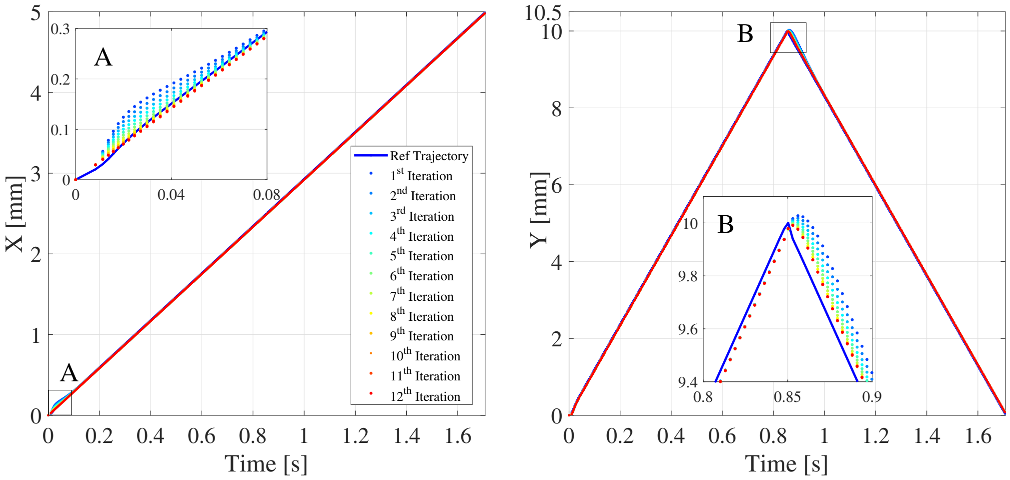


FIGURE 3.8: CILCC simulation result of time-series trajectory.

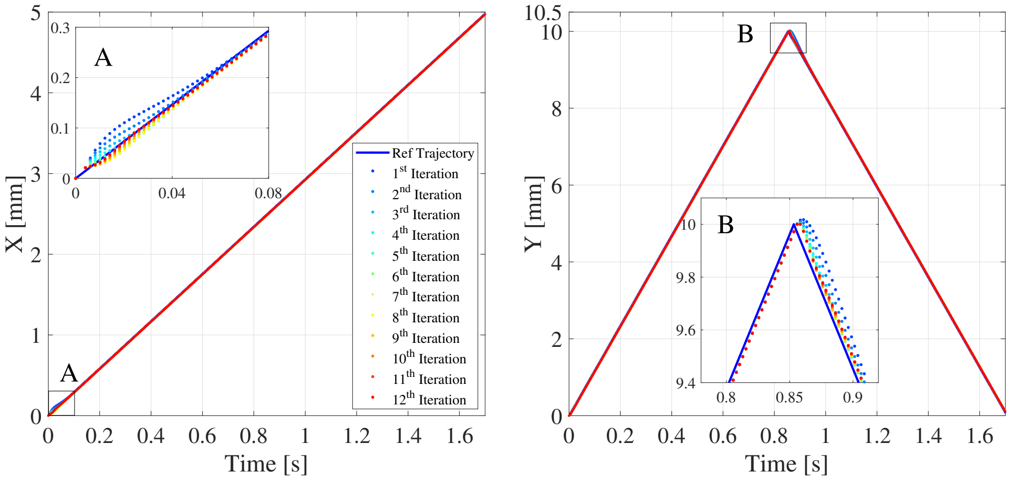
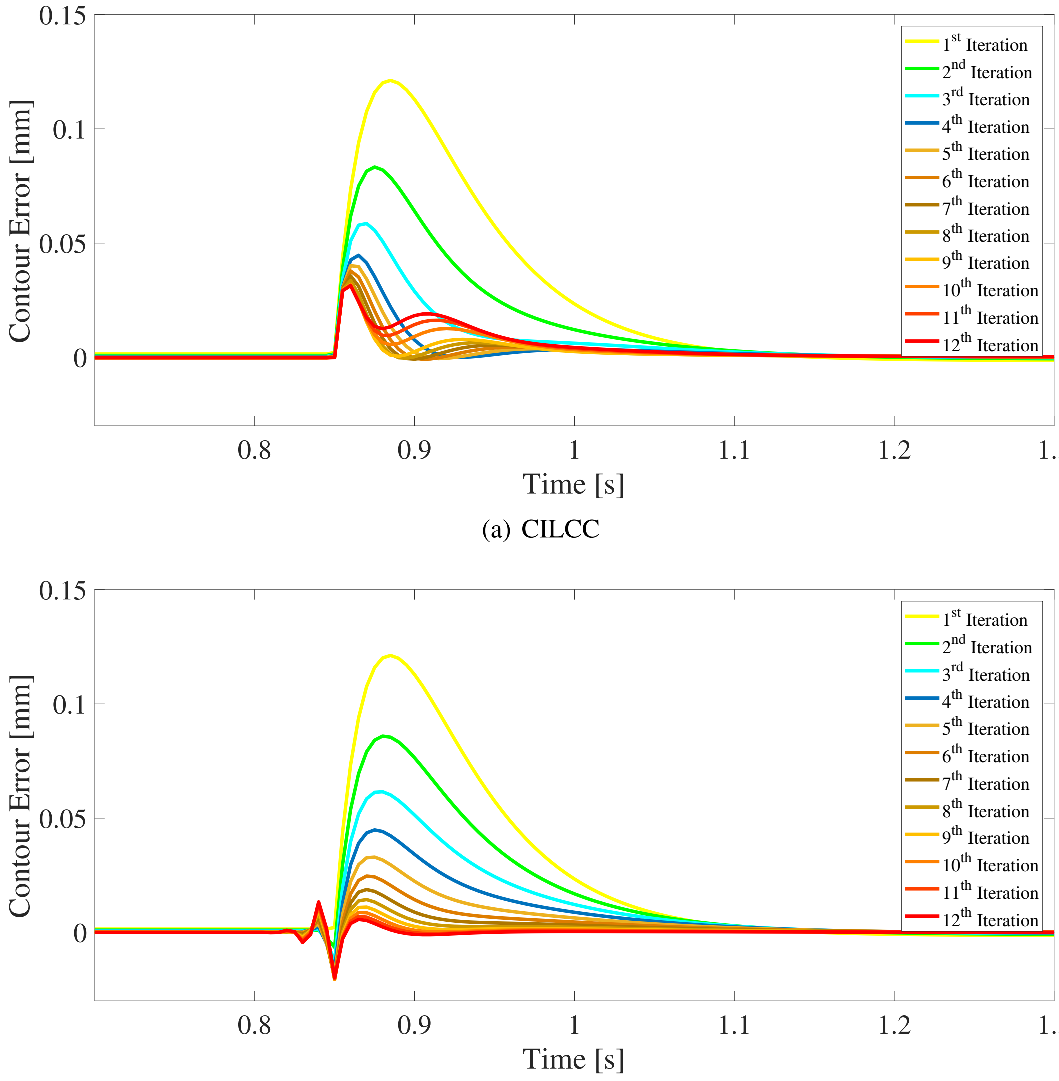


FIGURE 3.9: ILCC simulation result of time-series trajectory.

the CILCC, the maximum contour error was reduced to 0.0318 mm, which is equivalent to a reduction of 73.76 %. The proposed controller provides better performance as it reduced the maximum contour error to 0.0111 mm. This is equivalent to a reduction of 65.09 % and 90.84 % as compared to the CILCC and FBC, respectively. For clarity, simulation results are summarized in Table 3.2.

TABLE 3.2: Summary of simulation results [mm].

Controller	e_{tr}max		$e_c$max
	X-Axis	Y-Axis	
FBC	0.0252	0.1185	0.1212
CILCC	0.0066	0.0311	0.0318
ILCC	0.0023	0.0108	0.0111



(a) CILCC

(b) ILCC

FIGURE 3.10: Simulation result of contour error.

3.4 Experiment

3.4.1 Experimental Condition

In order to verify the effectiveness of the proposed controller, an X-Y table with a feedback controller implemented by C++ language in a personal computer (Windows OS) was used. The control input signal was applied to a biaxial feed drive system with two lead screws driven by two DC servo motors via DA board (CONTEC DA12-8(PCI)) as shown in Fig. 2.3. The actual position of X-Y table was measured by encoders with resolution 0.025 μm attached to each axis. The ILCC was programmed in MATLAB® in a separate personal computer (Windows OS) and embedded to the biaxial feed drive system as shown in Fig. 2.5. The same reference trajectory and parameters in Table 3.1 with simulation were employed.

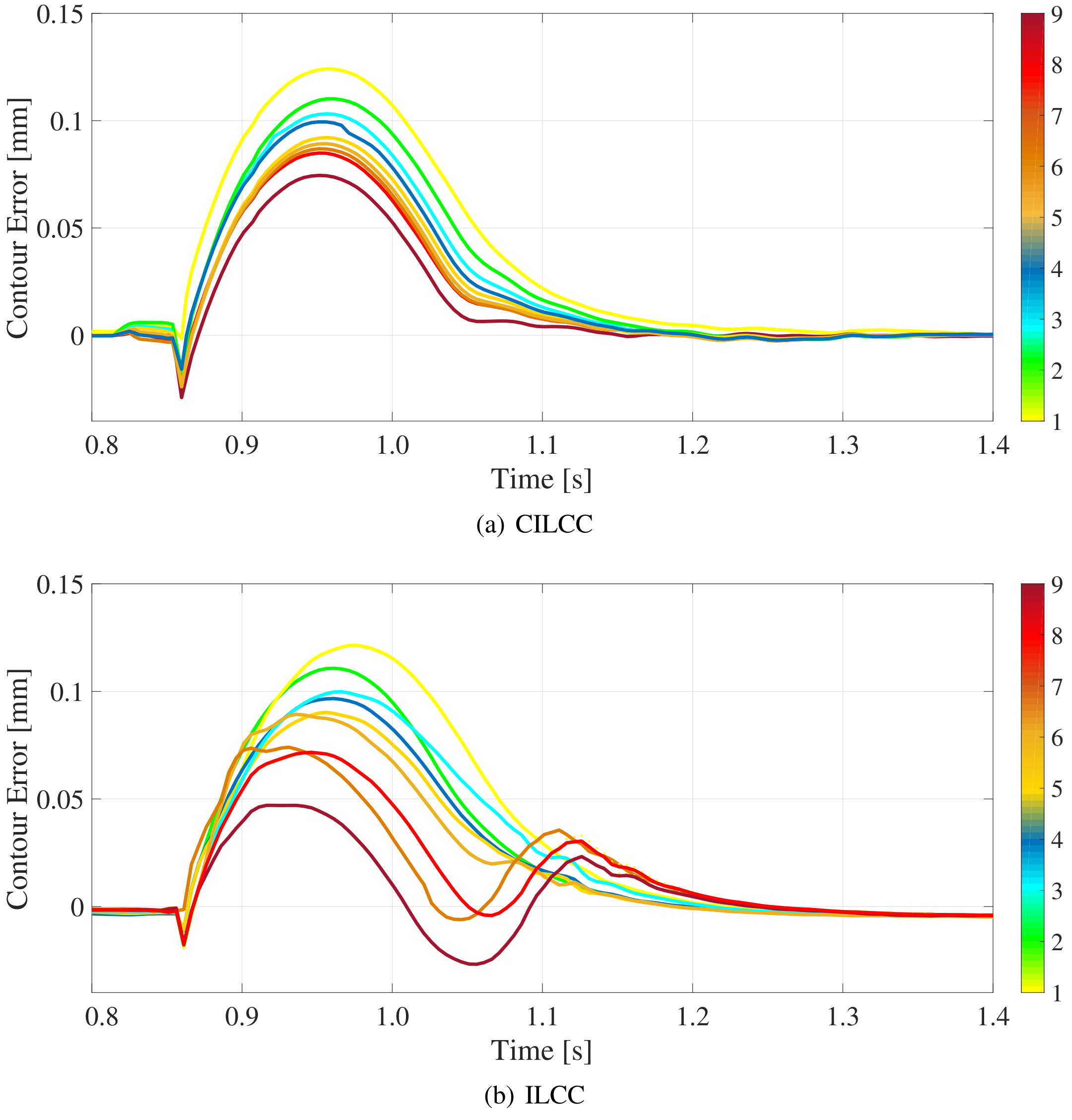
3.4.2 Experiment Results

Five times experiments with 12 iterations for each were conducted and the minimum contour error was reached after 9 iterations as shown in Figs. 3.11-3.16. Fig. 3.11 (a) and (b) show the contour errors based on the CILCC and ILCC, respectively, around the sharp-corner area (0.8 - 1.4 second). In both controllers, the first iteration

TABLE 3.3: Summary of experimental results: emr (%) is the error magnitude ratio to FBC.

Experiment Number	e_c max				
	FBC	CILCC		ILCC	
		[mm]	emr (%)	[mm]	emr (%)
1st	0.128	0.0768	60.0	0.0544	42.5
2nd	0.1247	0.0957	76.7	0.0362	29.0
3rd	0.1304	0.1003	76.9	0.0584	44.8
4th	0.1281	0.0948	74.0	0.0363	28.3
5th	0.1238	0.0932	75.3	0.0485	39.2
Mean	0.127	0.0921	72.6	0.0468	36.8
Experiment Number	e_c mean				
	FBC	CILCC		ILCC	
		[mm]	emr (%)	[mm]	emr (%)
1st	0.0222	0.0098	44.1	0.009	40.5
2nd	0.0203	0.0131	64.5	0.0082	40.4
3rd	0.021	0.0147	70.0	0.01	47.6
4th	0.0202	0.0133	65.8	0.0082	40.6
5th	0.0196	0.0126	64.3	0.0097	49.5
Mean	0.0207	0.0127	61.5	0.009	43.7

was controlled only by the FBC. Under the CILCC, the initial largest contour error was 0.128 mm which was reduced to 0.0768 mm in the 9th iteration and no further significant error reduction was observed thereafter. On the other hand, under the ILCC, the final maximum contour error was 0.0485 mm in the 9th iteration. This is equivalent to the error reduction of 60.8 % from the initial maximum contour error. Fig. 3.12 shows modified reference trajectories iteratively and real trajectories based on the ILCC. Fig. 3.14 shows the contour error convergence properties for both controllers whereby, although the ILCC converges relatively slower than the CILCC, it has minimum final contour error. The experimental reduction ratio is greater than



(a) CILCC

(b) ILCC

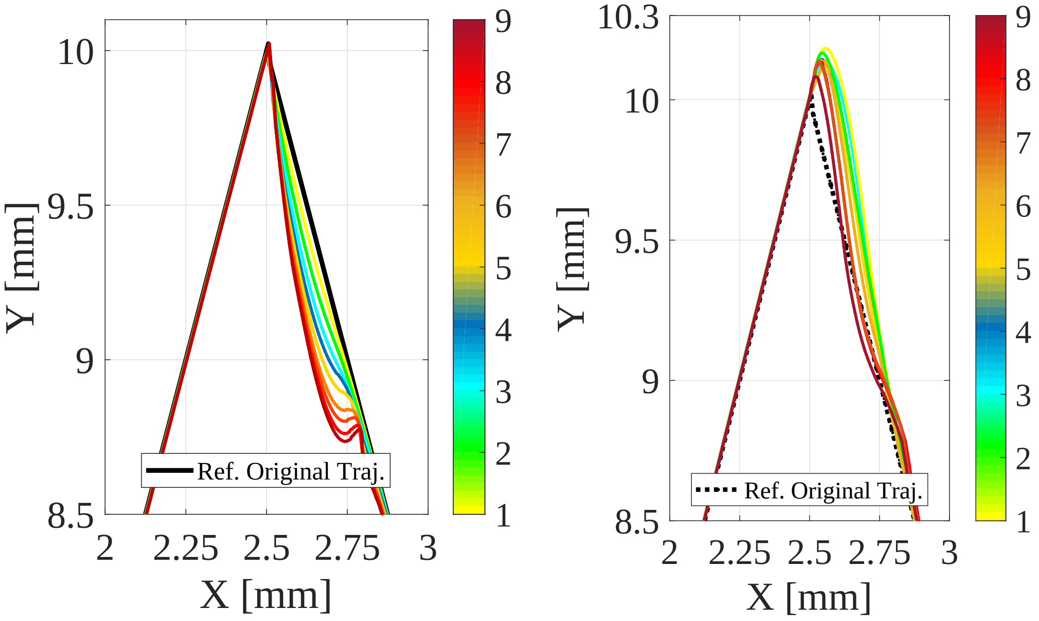
FIGURE 3.11: Experiment results for the contour error.

TABLE 3.4: Average convergence ratios until the 9th iteration.

	CILCC	ILCC
Calculated	0.509	0.603
Simulation	0.854	0.742
Experiment	0.96	0.922

the theoretical convergence factor ($\varepsilon_i = 0.603$) because of uncertain disturbance in the experimental machine.

For clarity, all the experimental results are summarized in Fig. 3.15 and Table 3.3, and comparisons with simulation results are shown in Fig. 3.16 and Table 3.4. It can be seen that similar error reduction tendency is observed between simulation and experiment. This concludes that the proposed controller is effective for performance improvement in feed drive systems.



(a) Modified reference trajectory

(b) Real trajectory

FIGURE 3.12: ILCC experimental results of trajectory profiles.

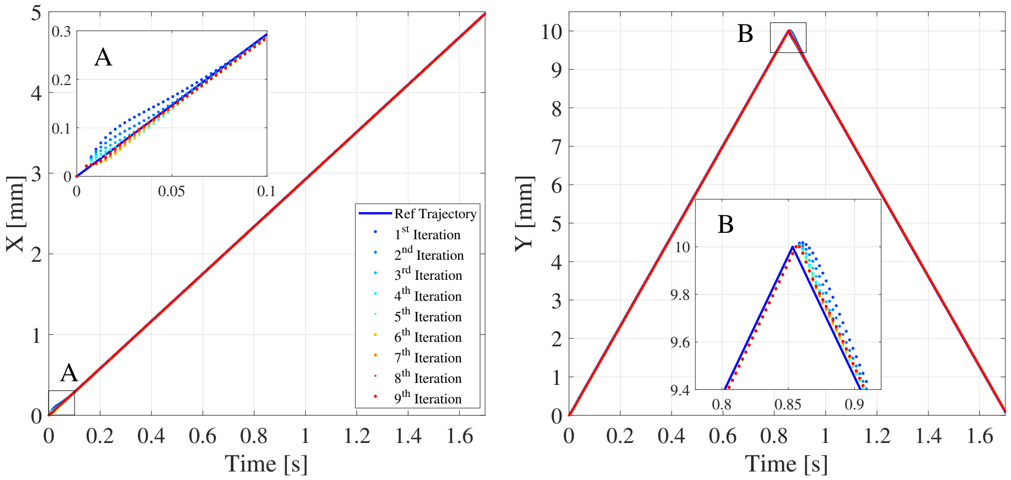


FIGURE 3.13: ILCC experiment result of time-series trajectory.

3.4.3 Discussion

Sharp corners are generally difficult to be tracked by feed drive systems under continuous motion because of infinite acceleration due to a trajectory discontinuity at the corner. Considering the reference trajectory in Fig. 3.12 as an example, both the X and Y drive axes move in the same direction before the sharp corner. After the corner, drive axis X continues with the same motion direction whereas drive axis Y reverses its direction. This requires an instant deceleration and acceleration in drive axis Y in order to track the desired motion, it is practically impossible. If normal-PID control is used, an overshoot will occur at the corner point [63]. On the other hand, since

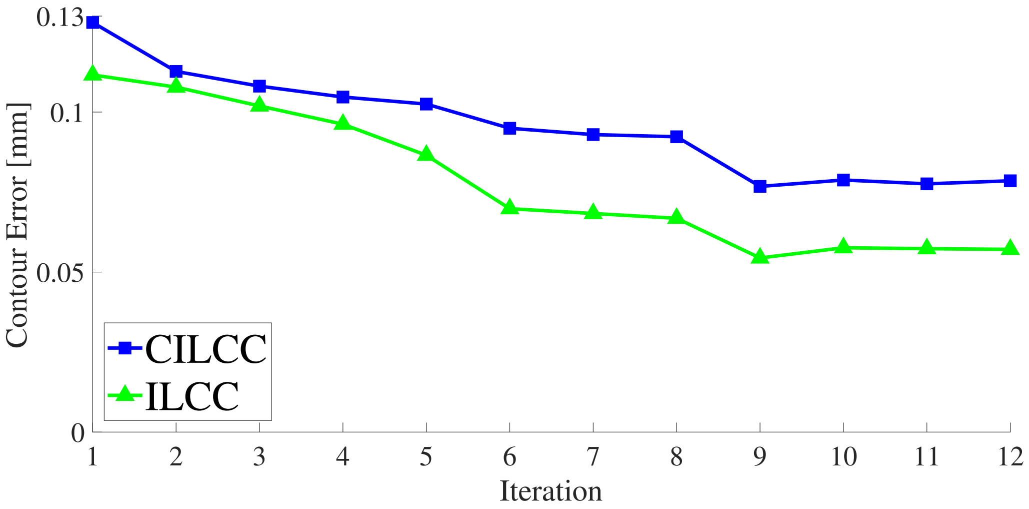


FIGURE 3.14: Experimental convergence profiles.

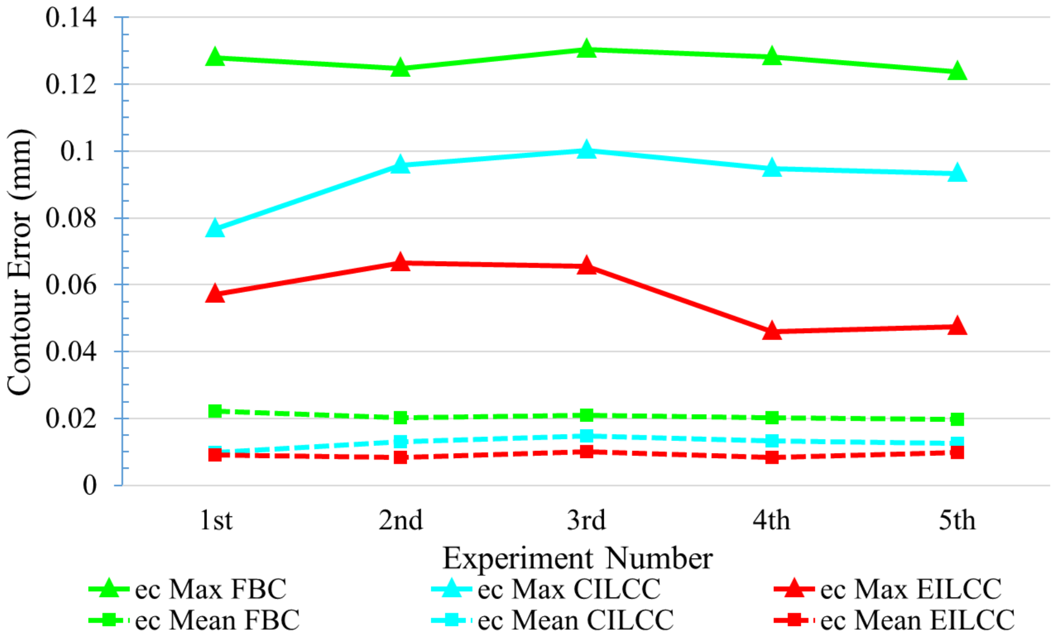


FIGURE 3.15: Summary of experimental result.

the CILCC iteratively modifies the control input, it can perform effectively. However, there are two major limitations. It may result in saturation of the control input, and cannot be applied to commercial machines due to limited access of embedded controllers.

The proposed ILCC overcomes the mentioned limitations since it deals with only

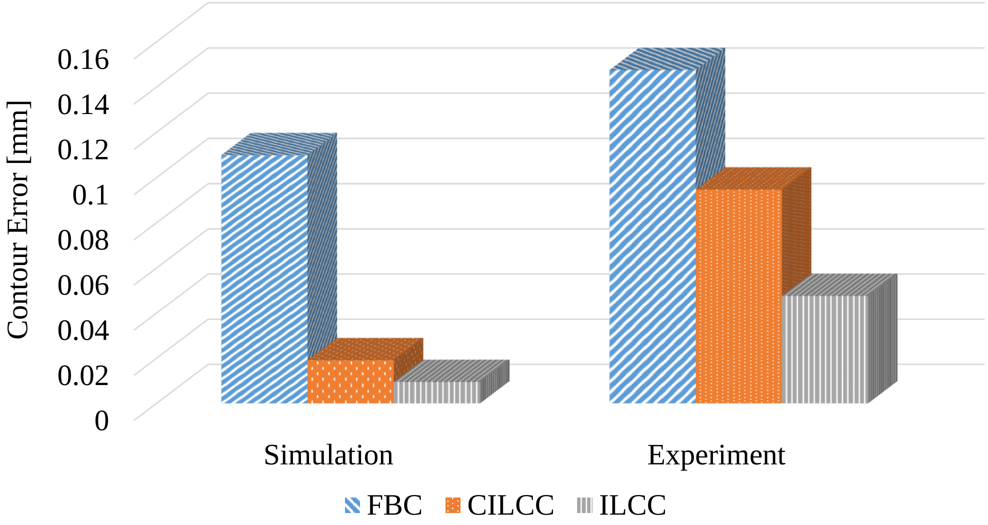


FIGURE 3.16: Comparison of simulation and experimental results.

the reference trajectory. In this case, the reference trajectory is iteratively modified as shown in Fig. 3.12 (a). The modified reference trajectory is executed by a CNC machine as an input trajectory to achieve better tracking performance as shown in Fig. 3.12 (b). Therefore, the proposed ILCC can be easily implemented in both existing and developing machines without any access or information of the embedded controller.

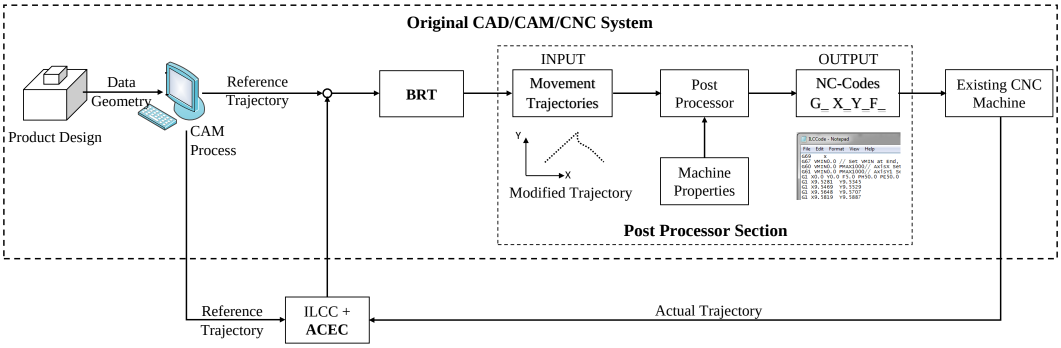
3.5 Conclusion

An ILCC is proposed for biaxial feed drive systems, and the performance has been demonstrated by simulation and experimental results. The proposed method can modify the reference trajectory iteratively to enhance the tracking performance. Experimental results have shown that the maximum contour error can be reduced by 63.1 % or 49.2 % when the ILCC is added to the conventional feedback controllers or compared to the CILCC, respectively. However, it needs more iteration to reach a minimum contour error for high curvature trajectory because of the assumption that linear connection between two discrete points on trajectory as shown in Fig. 3.2. And, this method generates non-uniform trajectory points along the reference path as explained later in Chapter 4 and Fig. 4.4, which may produce un-smooth velocity profiles. In the next chapter, the contour error estimation will be improved for high curvature trajectory and smoothing algorithm is needed to obtain smooth velocity profiles. Then, ILCC will be implemented in a commercial three-axis CNC machine tools.

Chapter 4

ILCC Based on Motion Trajectory Generation with Actual Contour Error Estimation and Bézier Reposition Trajectory

In industrial applications, highly accurate mechanical components are generally required to produce advanced mechanical and mechatronic systems. In machining mechanical components, contour error represents the product shape quality directly, and therefore it must be considered in controller design. Although most existing contouring controllers are based on feedback control and estimated contour error, it is generally difficult to replace the feedback controller in commercial computerized numerical control (CNC) machines. This paper proposes an iterative learning contouring controller (ILCC) by considering the actual contour error compensation (ACEC) and Bézier reposition trajectory (BRT), which can be applied in CNC machines currently in use without any modification of their original feedback controllers. While the ACEC enhances tracking performance by compensating the reference input with an actual value, the BRT enables smooth velocity transitions between discrete points



```
G69    X
G67 VMIN0.0 // Set VMIN at End,
G60 VMIN0.0 PMAX1000// AxisX Set
G61 VMIN0.0 PMAX1000// AxisY1 Se
G1 X0.0 Y0.0 F5.0 PH50.0 PE50.0
G1 X9.5281  Y9.5345
G1 X9.5469  Y9.5529
G1 X9.5648  Y9.5707
G1 X9.5819  Y9.5887
```

FIGURE 4.1: ILCC block diagram [64].

in the reference trajectory. For performance analysis, the proposed controller was implemented in a commercial three-axis CNC machine and several experiments were conducted based on typical 3D sharp-corner and half-circular trajectories. Experimental results showed that the proposed controller could reduce the maximum and mean contour errors by 45.11 % and 54.48 % on average, compared to ILCC with estimated contour error. By comparing to ILCC with ACEC, the maximum and mean contour errors are reduced to 20.54 % and 26.92 %, respectively.

4.1 Introduction

Highly accurate mechanical components with high-curvature surface are widely required in the fields of aerospace, energy, power, medical, and automobile, such as propellers [1], gears [2], knee joints [3], impellers with small splitter blades [65]–[67], ship structural components [68], and so on. To produce such components, many engineers choose CNC machining process, which has advantages of high production efficiency, high machining precision, and stable product quality. Five-axis CNC machining further improves the flexibility, precision, and efficiency [69], [70]. Due to the above objective, two or more axes on these machines have to simultaneously move and their motions must be controlled accurately in order to produce precise machined parts [71]. Achieving high precision in machining, highly depends on control performance of each individual feed drive axis [55]. Under independent drive axial control, load disturbance or performance variance of either axis causes contour error [56]. Efforts to reduce or eliminate contour errors have been made through either the design of advanced feedback controllers for each feed drive or modifying the reference axial positions [71]. This issue has attracted many researchers in the machine tool community because it is directly related to the product quality.

A cross-coupled control system was proposed for biaxial manufacturing systems to improve the contour accuracy by coupling the individual axial errors [19]. A new structure of cross-coupling controller with a simpler design process for precise tracking in motion control was proposed [58], which can be implemented easily on most motion systems via reprogramming the reference position command subroutine. Many cross-coupled methods improved to enhance the contouring performance by coupling axial errors under high feed-rates such as adaptive feed rate and cross-coupled iterative learning control (CCILC) [18], [20]–[23]. In ILC, a machining error is reduced iteratively through modification of the control input or the reference trajectory. In linear-type ILCs, the convergence of the output error is guaranteed under a certain condition even when system parameters are not known exactly and periodic disturbances exist [24], [25]. Besides, ILCs with contouring controller or an iterative contouring controller minimizes the contour error through an iterative estimation of the instantaneous curvature of the reference trajectory and coordinate transformation [41]. Furthermore, the contour error can be significantly reduced by an ILC which considers both tracking and contour errors [42]. An ILCC with contour error approximation by coordinate transformation was proposed in [61], [64], which could be applied to a commercial machine without any modification in the existing controller.

All of the methods mentioned above use approximated contour errors instead of actual ones to design contouring controllers or modify reference trajectories [41], [44]. Most of the error approximation methods do not represent the contour error accurately, especially for a high-curvature surface application. However, contouring control with actual contour errors is difficult to be implemented in real-time control systems because the actual contour error cannot be obtained immediately for complex contour profiles. Therefore, it requires contour error estimation to obtain approximated values. On the other hand, with an embedded iterative control system, it is possible to obtain actual contour errors by offline calculation. However, this method generates non-uniform trajectory points along the reference path as explained later in Section 4.4, which may produce un-smooth velocity profiles. Therefore, smoothing algorithm is needed to obtain smooth velocity profiles. Parametric splines, particularly Bézier curves have been widely used for trajectory smoothing by interpolating linear trajectory points for both mobile robots and feed drive systems [72], [73]. Thus, they can be applied for the above mentioned matter as well.

This study proposes an ILCC with ACEC and BRT to generate uniform trajectory points as shown in Fig. 4.1. By calculating the minimum distance between the actual position and the reference trajectory, the ACEC is used to achieve the desired contour precisely. The ILCC modifies the original trajectory (NC program) through an ACEC as a learning controller. The modified reference trajectory is smoothed by quintic Bézier curves through interpolation of the reference trajectory with uniformly distanced reference points to obtain a BRT. The BRT is executed by a CNC machine to reduce the contour error, iteratively. The proposed method is proven to provide better performance than the ILCC either with or without ACEC.

The rest of this chapter is organized as follows: Sections 4.2 - 4.6 gives a description of the dynamic model of three-axis machine tools, definition of the contour error, ACEC, BRT, controller design, and convergence analysis, respectively. Simulation results, which compare ILCC and ILCC with ACEC in 2D sharp corner trajectories, are presented in Section 4.7. Experimental results, which compare ILCC, ILCC with ACEC, and ILCC with ACEC and BRT in 3D sharp corner and 3D half-circular trajectories, are presented in Section 4.8, followed by concluding remarks in Section 4.9.

4.2 Dynamic Model of Three Axis CNC Machine Tools

The dynamics of a three-axis machine tools is represented as follows:

$$\begin{aligned} & M\ddot{q}+C\dot{q}=u, \\ & M=\mathrm{diag}\{m_i\},\ C=\mathrm{diag}\{c_i\}, \quad i=x,\ y,\ z, \\ & q=[q_x,\ q_y,\ q_z]^{\mathrm{T}},\ u=[u_x,\ u_y,\ u_z]^{\mathrm{T}}, \end{aligned} \tag{4.1}$$

where m_i, c_i, q_i, and u_i are the inertia, viscous friction coefficient, actual position, and control voltage for the axis i, respectively.

4.3 Actual Contour Error Compensation

4.3.1 Definition of Contour Error

The contour error is defined as the shortest distance between the actual and desired contours. The difference between tracking error and contour error in machine tools is shown in Fig. 4.2. The tracking error in each axis is the difference between the reference and actual positions. The reference position from the starting position of the machine tool system at time t in the coordinate frame Σ_w is denoted by $r = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T$. The tracking error in each axis is defined as

$$e_w = \begin{bmatrix} e_x & e_y & e_z \end{bmatrix}^T = r - q, \tag{4.2}$$

$$e = \|e_w\|. \tag{4.3}$$

The coordinate frame Σ_l is attached at r and its axial directional vectors are $\mathfrak{T}$, $\mathfrak{N}$, and $\mathfrak{B}$, respectively. The axis $\mathfrak{T}$ is in the tangential direction of the reference trajectory at

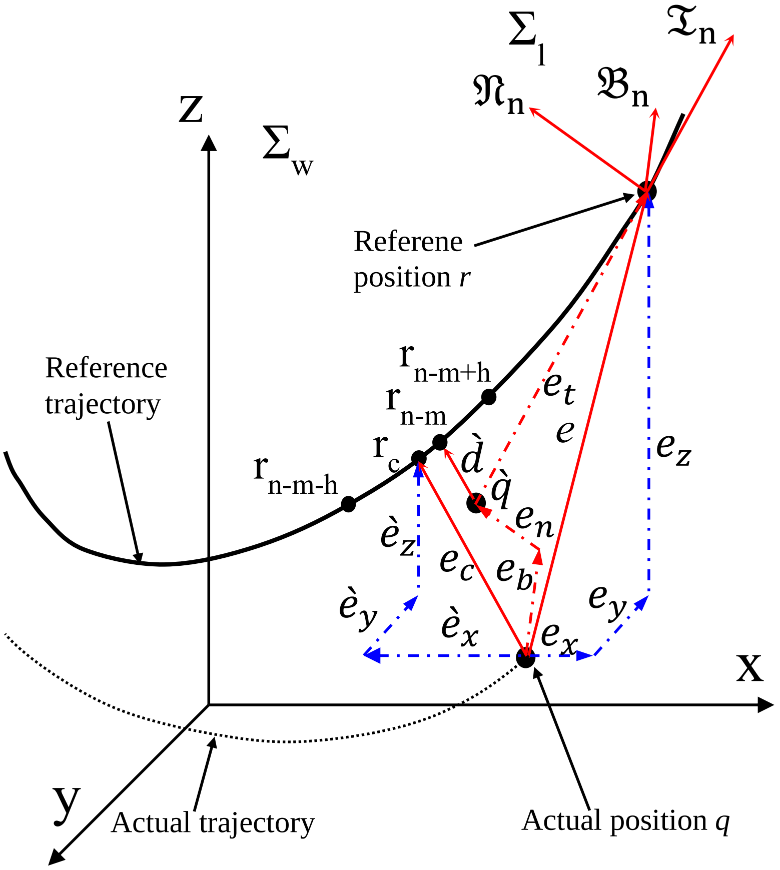


FIGURE 4.2: 3D tracking and contour errors.

r, the direction of $\mathfrak{N}$ is perpendicular to $\mathfrak{T}$, and the $\mathfrak{B}$-axis is the bi-normal component to $\mathfrak{T}$ and $\mathfrak{N}$. The axial directions of frame Σ_l are calculated as follows:

$$\mathfrak{T} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T = \frac{\dot{r}}{\|\dot{r}\|}, \tag{4.4}$$

$$\mathfrak{N} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T = \frac{\dot{\mathfrak{T}}}{\|\dot{\mathfrak{T}}\|}, \tag{4.5}$$

$$\mathfrak{B} = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T = \mathfrak{T} \times \mathfrak{N}. \tag{4.6}$$

The tracking error vector e_l with respect to Σ_l can be expressed as

$$e_l = \begin{bmatrix} e_t & e_n & e_b \end{bmatrix} = L^T e_w, \tag{4.7}$$

$$L = \begin{bmatrix} t_x & n_x & b_x \\ t_y & n_y & b_y \\ t_z & n_z & b_z \end{bmatrix}^T . \tag{4.8}$$

The closest position of the desired contour to q is denoted by $r_c = \begin{bmatrix} r_{cx} & r_{cy} & r_{cz} \end{bmatrix}^T$. The actual contour error, which is the shortest distance between the actual position q and the reference trajectory is calculated as follows:

$$e_c = \begin{bmatrix} e_{cx} & e_{cy} & e_{cz} \end{bmatrix}^T = r_c - q, \tag{4.9}$$

$$\grave{e} = ||e_c|| . \tag{4.10}$$

4.3.2 Actual Contour Error Compensation (ACEC)

ACEC is a direct method to compensate for the actual contour error by finding the closest position between the actual position and the reference trajectory as shown in Fig. 4.3. Here we assume that the reference trajectory is represented by discrete points as $\grave{r}_c$, $\grave{r}_{c+1}$ ... in the figure and r_n denotes the current reference position coresponding to the position q. $\grave{q}$ is a candidate position, which is defined in the same manner with contour error estimation by a rotation matrix to estimate projected contour error in tangential direction on coordinate frame Σ_l as in [64] and is represented as follows:

$$\grave{q} = q + L^{-1} \begin{bmatrix} 0 \\ e_n \\ e_b \end{bmatrix} . \tag{4.11}$$

The shortest distance $\grave{d}$ between $\grave{q}$ and the reference trajectory is calculated as follows:

$$\grave{d} = \min_{0<m<n} ||r_{n-m} - \grave{q}|| , \tag{4.12}$$

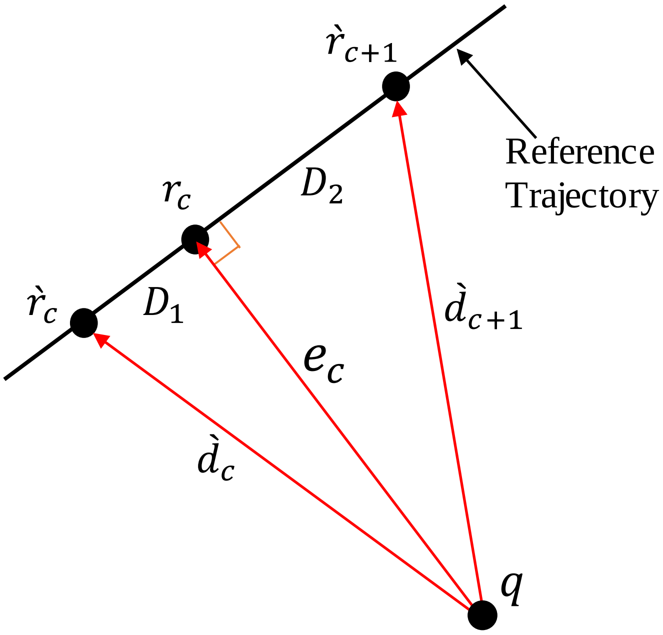


FIGURE 4.3: Contour error interpolation.

where r_{n-m} is the closest candidate reference position in $(n-m)^{\text{th}}$ point, and m is determined to find shortest distance $\grave{d}$. The approximated closest position of the desired contour to q is denoted by $\grave{r}_c$, which is obtained by finding the minimum distance $\grave{d}_c$ between the actual position q and a part of reference trajectory from $(n-m-k)^{\text{th}}$ to $(n-m+k)^{\text{th}}$ discrete positions, where $(n-m)^{\text{th}}$ is calculated by Eq. (4.12) and k is determined to avoid high computation process. $\grave{r}_c$ is calculated as follows

$$\grave{r}_c = q + \grave{d}c, \tag{4.13}$$

$$\grave{d}_c = \min_{-k<h<k} \|e_c\|, \tag{4.14}$$

where

$$e_c = r_{n-m+h} - q$$

The actual position r_c is defined by linear interpolation of the neighborhood points as shown in Fig. 4.3 with equations below:

$$r_c = \grave{r}_c + \frac{\left(\grave{r}_{c+1} - \grave{r}_c\right) D_1}{D_2}, \tag{4.15}$$

$$D_1 = \frac{\grave{d}_c^2 - \grave{d}_{c+1}^2 + \left\|\grave{r}_{c+1} - \grave{r}_c\right\|^2}{2\left\|\grave{r}_{c+1} - \grave{r}_c\right\|}, \tag{4.16}$$

$$D_2 = \left\|\grave{r}_{c+1} - \grave{r}_c\right\| - D_1, \tag{4.17}$$

where $\grave{r}_{c+1}$, $\grave{d}_{c+1}$, D_1, and D_2 are the discrete position next to the approximated discrete position $\grave{r}_c$, the shortest distance from $\grave{r}_{c+1}$ to the actual position q, the distance between $\grave{r}_c$ and r_c, and the distance from r_c to $\grave{r}_{c+1}$, respectively.

4.4 Bézier reposition trajectory

Herein, BRT refers to the modified reference trajectory that has been smoothed by quintic Bézier curves through interpolation of the ILCC-modified reference trajectory with uniformly distanced discrete points as shown in Fig 4.4. Bézier curve is a parametric curve defined by several control points depending on its order, and it always passes through the starting and ending control points [72]. A Bézier curve of order w can be represented mathematically as follows [74]:

$$B\left(\psi\right) = \sum_{v=0}^{w} b_v^w\left(\psi\right) P_v, \qquad 0 \leq \psi \leq 1, \tag{4.18}$$

where b_v^w, P_v, v, and ψ are the w^{th} order Bernstein polynomial, the control points, control points number, and the Bézier parameterization variable. The Bernstein polynomial is represented as follows:

$$b_v^w\left(\psi\right) = \frac{w!}{v!\left(w-v\right)!}\psi^v\left(1-\psi\right)^{w-v}, \\ v = 0,\ 1,\ 2,\ 3,\ 4,\ 5. \tag{4.19}$$

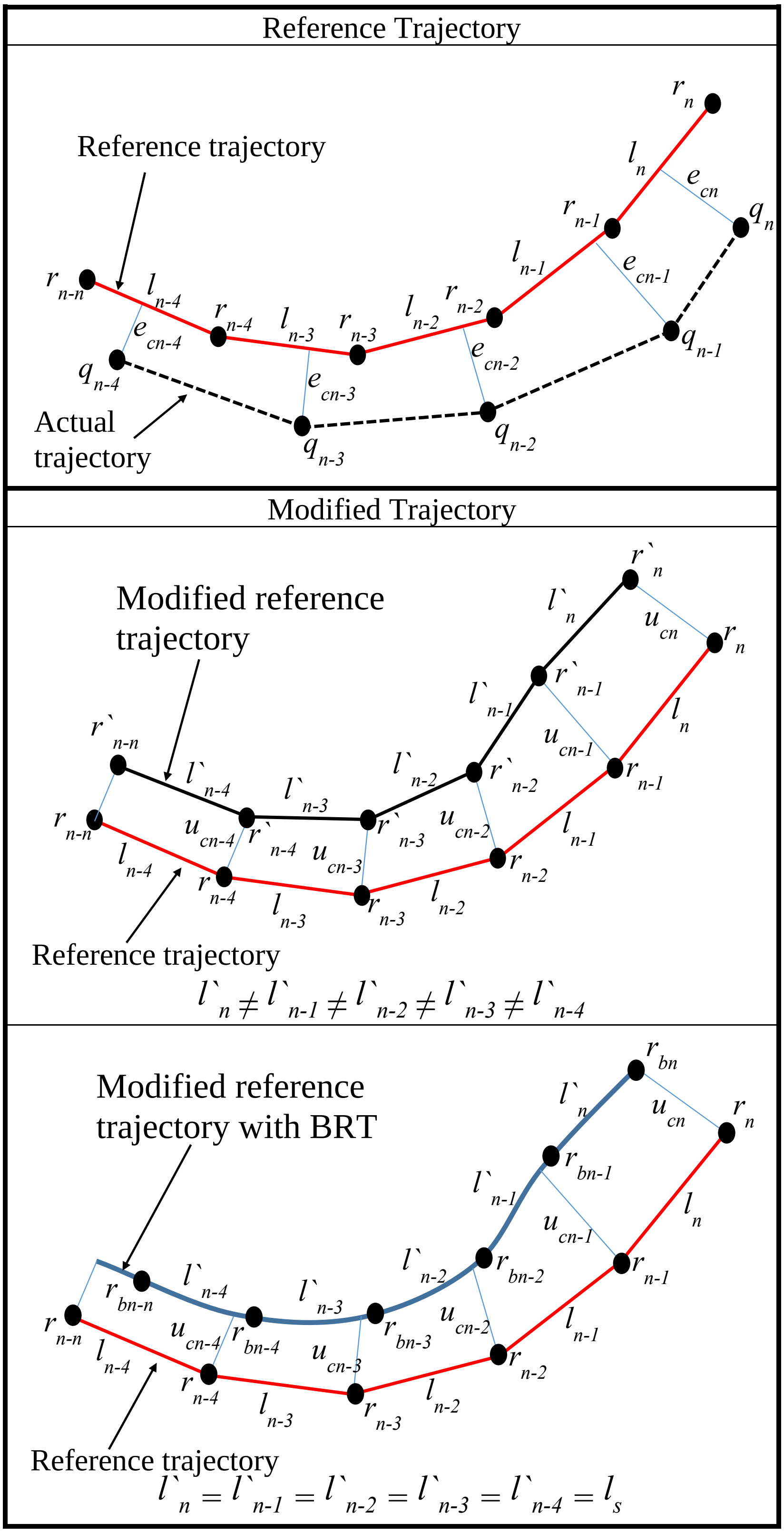


FIGURE 4.4: Bézier reposition trajectory concept.

The BRT has two main parts; Bézier curve generation and reposition of the trajectory points. Bézier curve generation considers parameterization of a linear segment by a Bézier curve (BC) and corner smoothing by an inserted Bézier curve (IBC) as detailed in [73].

From Eqs. (4.18) and (4.19), the BC and the IBC with 5th order Bernstein polynomials for n^{th} segments are represented as follows:

$$B_n(\psi) = (1-\psi)^5 P_{0n} + 5\psi(1-\psi)^4 P_{1n} + 10\psi^2(1-\psi)^3 P_{2n} \tag{4.20}$$

and

$$\grave{B}_n(\psi) = (1-\psi)^5 \grave{P}_{0n} + 5\psi(1-\psi)^4 \grave{P}_{1n} + 10\psi^2(1-\psi)^3 \grave{P}_{2n} \tag{4.21}$$

where $\grave{B}_n$ and $\grave{P}_{in}$, $i = 0, ..., 5$ are the inserted Bézier curve, and its control points. The control points for the BC and the IBC are defined by the following equations:

$$\begin{aligned}
P_{0n} &= r_{n-1} + l_{n-1}\mu_n, \\
P_{1n} &= r_{n-1} + l_{n-1}\mu_n(2-\alpha_n), \\
P_{2n} &= r_{n-1} + l_{n-1}\mu_n(\beta_n - 4\alpha_n + 4), \\
P_{3n} &= r_n - l_n\mu_n(\beta_n - 4\alpha_n + 4), \\
P_{4n} &= r_n - l_n\mu_n(2-\alpha_n), \\
P_{5n} &= r_n - l_n\mu_n,
\end{aligned} \tag{4.22}$$

and

$$\begin{aligned}
\grave{P}_{0n} &= r_n - l_n\mu_n, \\
\grave{P}_{1n} &= r_n - l_n\alpha_n\mu_n, \\
\grave{P}_{2n} &= r_n - l_n\beta_n\mu_n, \\
\grave{P}_{3n} &= r_n + l_n\beta_n\mu_{n+1}, \\
\grave{P}_{4n} &= r_n + l_n\alpha_n\mu_{n+1}, \\
\grave{P}_{5n} &= r_n + l_n\mu_{n+1},
\end{aligned} \tag{4.23}$$

where l_n, μ_n, α_n, and β_n are the length, tangential vector, and the fractions of the length l_n for n^{th} trajectory segment. Because a geometrical error γ is induced at the corners by the smoothing process, the smoothing algorithm must guarantee that γ is within the predefined tolerance of the contour error [75]. The maximum geometry error occurs at the middle point of the IBC, i.e., at $\psi = 0.5$ as follows:

$$\gamma = \left\| r_n - \grave{B}_n(0.5) \right\|. \tag{4.24}$$

From Eqs. (4.21) and (4.23), the point $\grave{B}(0.5)$ is found by

$$\grave{B}_n(0.5) = r_n + \frac{l_n}{32}(1 + 5\alpha_n + 10\beta_n)(\mu_{n+1} - \mu_n). \tag{4.25}$$

From Eqs. (4.24) and (4.25), the length l_n is calculated as

$$l_n = \frac{32\gamma}{(1 + 5\alpha_n + 10\beta_n)\|\mu_n - \mu_{n+1}\|}. \tag{4.26}$$

Because μ_n and μ_{n+1} are unit vectors, Eq. (4.26) can be written as

$$l_n = \frac{32\gamma}{(1 + 5\alpha_n + 10\beta_n)\sqrt{2 - 2\cos\theta_n}}, \tag{4.27}$$

where θ_n is the inclination between the corresponding linear segments.

The generated BRT has higher number of discrete points than the original reference trajectory, which directly affect the velocity profile. Therefore, the number of points is re-calculated based on the sampling time and the desired velocity as follows:

$$n_s = \frac{\sum_{k=1}^{n} \|B_k(\psi) - B_{k-1}(\psi)\|}{l_s}, \tag{4.28}$$

$$l_s = t_s f, \tag{4.29}$$

where n_s, l_s, t_s, and f are the re-calculated number of discrete points, length between the points, sampling time, and velocity, respectively. The BRT is re-defined by considering the following condition:

$$r_{B_n} = B_n(\psi), \quad \text{if} \sum_{k=1}^{n} \|B_k(\psi) - B_{k-1}(\psi)\| = l_s n, \tag{4.30}$$
$$1 \leq n \leq n_s.$$

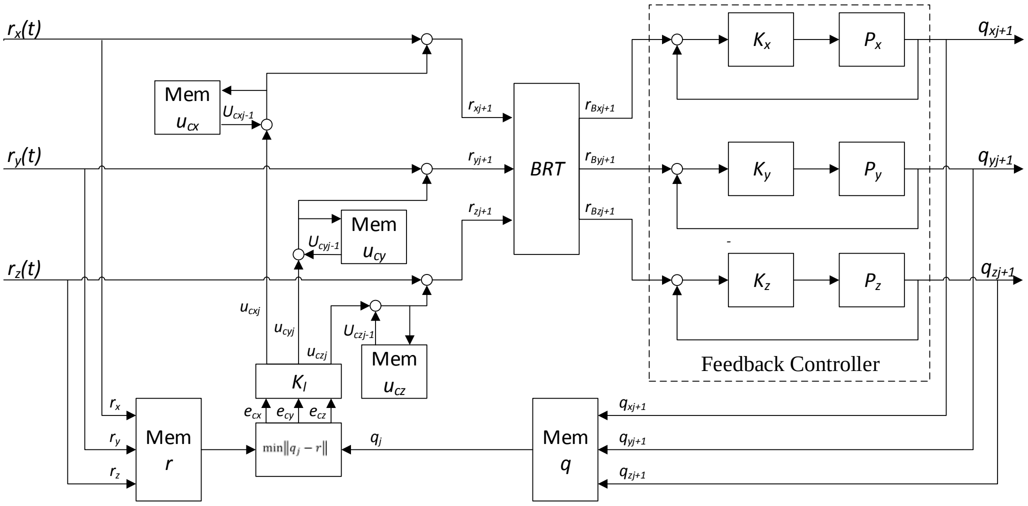


FIGURE 4.5: Proposed controller design.

4.5 Controller Design

The proposed controller for the three-axis machine tool consists of a feedback controller (FBC) and an ILCC with ACEC and BRT. The input of the FBC is the modified reference trajectory BRT generated from the ILC with ACEC as shown in Fig. 4.5, where mem u_c, q, and r are the memories of learning compensation value, actual position, and reference trajectory for the x, y, and z axes. The transfer function of the plant including the FBC is represented as follows:

$$G_i(s) = \frac{M_i(s) K_i(s)}{1 + M_i(s) K_i(s)}, \tag{4.31}$$

where G_i, M_i, K_i, and s are the transfer function from the reference command to the actual position, the machine plant, the feedback compensator, and the variable of the Laplace transform for the i^{th} axis, respectively. The FBC compensator $K_i(s)$ is defined as a PID compensator as follows:

$$K_i(s) = K_{Pi} + \frac{1}{s} K_{Ii} + s K_{Di}, \tag{4.32}$$

where K_{Pi}, K_{Ii}, and K_{Di} are the proportional, integral, and derivative gains for the i^{th} axis, respectively. The BRT for the $j+1$ iteration is represented as follows:

$$r_{B_{ij+1}}(t) = B(\psi) r_{i_{j+1}}, \tag{4.33}$$

where B, t, ψ, $r_{B_{ij+1}}$, and r_{ij+1} are three dimensional Bézier curve, sampling instant, control points, Bézier modified trajectory, and reference signal for $j+1^{\text{th}}$ and i^{th} axis, respectively. Reference signal $r_{ij+1}(t)$ is compensated by an iterative learning compensator K_l with ACEC approximation which is represented as follows:

$$r_{ij+1}(t) = r_{ij}(t) + u_{cij}(t), \tag{4.34}$$

with

$$\begin{aligned} u_{cij}(t) &= K_{li}(s)\, e_{cij}(t), \\ K_{li}(s) &= K_{Pli} + \frac{1}{s} K_{Ili} + sK_{Dli}, \end{aligned}$$

where r_{ij}, u_{cij}, e_{cij}, K_{Pli}, K_{Ili}, and K_{Dli} are reference signal, contouring compensated input, contour error for the j^{th} iteration, and the proportional, integral, and derivative learning compensator gain K_l for i^{th} axis, respectively. Therefore,

$$u_{cij}(t) = K_{Pli} e_{cij}(t) + K_{Ili} \int_0^t e_{cij}(\tau)\, d\tau + K_{Dli} \dot{e}_{cij}(t). \tag{4.35}$$

substituting Eq. (4.34) to Eq. (4.33), leads to

$$r_{B_{ij+1}}(t) = B(\psi) \left(r_{i1}(t) + \sum_{n=0}^{j} u_{cij-n}(t) \right), \tag{4.36}$$

4.6 Convergence Analysis

Performance of the ILCC with ACEC and BRT can be guaranteed through convergence analysis to ensure that the contour error is iteratively reduced. The convergence condition of ILC is represented as

$$\varepsilon_i(t) = \left\| \frac{e_{cij+1}(t)}{e_{cij}(t)} \right\| < 1, \tag{4.37}$$

where $\varepsilon_i(t)$ is the convergence factor for the i^{th} axis. Based on the cascade ILC [62], the convergence can be guaranteed if the error magnitude at time t in iteration $j+1$ is smaller than the error magnitude in iteration j. From Eqs. (4.9) and (4.14) and the assumption that $r_{B_{ij+1}}$ is equal to r_{ij+1}, convergence factor is derived as follows:

$$\begin{aligned} e_{cij+1} &= \min\left(r_i - G_i(s)\, r_{B_{ij+1}}\right), \qquad r_{B_{ij+1}} \approx r_{ij+1}, \\ &= \min\left(r_i - G_i(s)\, r_{ij+1}\right), \\ &= \min\left(r_i - G_i(s)\left(r_{ij} + u_{cij}\right)\right), \\ &= \min\left(r_i - G_i(s)\, r_{ij} - G_i(s)\, K_{li}(s)\, e_{cij}\right), \\ &= \min\left(r_i - G_i(s)\, r_{ij}\right) - G_i(s)\, K_{li}(s)\, e_{cij}, \\ &= e_{cij} - G_i(s)\, K_{li}(s)\, e_{cij}, \\ &= \left(1 - G_i(s)\, K_{li}(s)\right) e_{cij}, \\ \frac{e_{cij+1}}{e_{cij}} &= 1 - \frac{M_i(s)\, K_i(s)\, K_{li}(s)}{1 + M_i(s)\, K_i(s)}, \end{aligned} \tag{4.38}$$

substituting Eq. (4.38) to Eq. (4.37) leads to

$$\varepsilon_i(s) = \left\| 1 - \frac{M_i(s)\, K_i(s)\, K_{li}(s)}{1 + M_i(s)\, K_i(s)} \right\| < 1. \tag{4.39}$$

Considering the discrete-time form for the implementation, $M_i(s)$ is represented as follows:

$$M_i\left(z^{-1}\right) = \frac{1}{m_i\left(\frac{z^{-1}-1}{t_s z^{-1}}\right)^2 + c_i \frac{z^{-1}-1}{t_s z^{-1}}}, \quad z = e^{j\omega t_s}, \tag{4.40}$$

where z^{-1}, ω, and z are the delay operator, the angular frequency and the z-domain variable, respectively. The feedback and learning compensator in the z-domain are represented as

$$K_i\left(z^{-1}\right) = K_{Pi} + K_{Ii}\left(\frac{t_s z^{-1}}{z^{-1}-1}\right) + K_{Di}\left(\frac{z^{-1}-1}{t_s z^{-1}}\right), \tag{4.41}$$

$$K_{li}\left(z^{-1}\right) = K_{Pli} + K_{Ili}\left(\frac{t_s z^{-1}}{z^{-1}-1}\right) + K_{Dli}\left(\frac{z^{-1}-1}{t_s z^{-1}}\right). \tag{4.42}$$

The convergence speed depends on the parameter ε, which should be kept as minimum as possible for fast convergence with a delay factor η due to the system delay.

Considering Eq. (4.39) and the delay, the following objective function for minimization is considered:

$$J_i = \min_{K_{Pli},\, K_{Ili},\, K_{Dli}} \left\| 1 - \frac{z^{\eta} M_i\left(z^{-1}\right) K_i\left(z^{-1}\right) K_{li}\left(z^{-1}\right)}{1 + M_i\left(z^{-1}\right) K_i\left(z^{-1}\right)} \right\|_{\infty}, \qquad (4.43)$$

$$\forall \omega \in \Omega,$$

where Ω is the considered domain of the operational frequency. Eq. (4.43) is solved by “fmincon” function in MATLAB® to find learning compensator gains.

4.7 Simulation

4.7.1 Simulation Condition

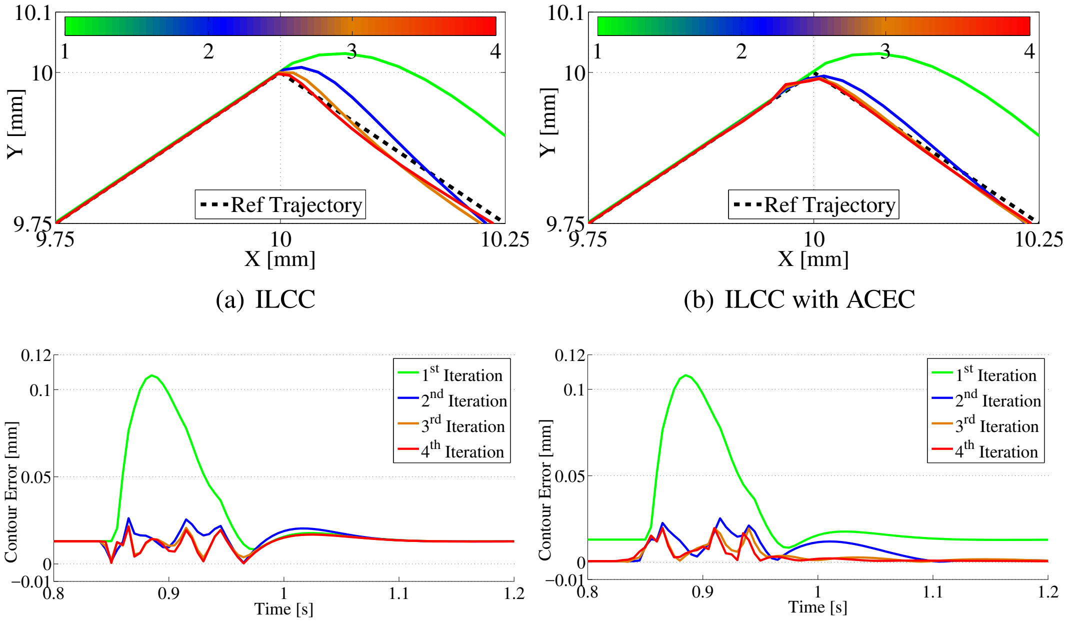
A three axis machine tools system (Fig. 2.6) with $m_i = [0.45 \quad 0.65 \quad 0.65]$ $\mathrm{Vs^2/mm}$ and $c_i = [0.144 \quad 0.24 \quad 0.24]$ $\mathrm{Vs/mm}$ were chosen for the simulation. Its table and ball screws in each axis are driven by AC servo motors. The simulation for a biaxial table (X-Y axis) was done by the MATLAB® software. The initial reference trajectory was defined in a G-code form to generate a right-angled sharp-corner trajectory.

The feedback controller gains were chosen from default gains of the existing machine as $[K_P \quad K_I \quad K_D] = [0.96\ \mathrm{V/mm} \quad 0.01\ \mathrm{V/smm} \quad 8\ \mathrm{Vs/mm}]$. The optimal learning compensator gains were calculated from Eq. (4.43) as $[K_{Pl} \quad K_{Il} \quad K_{Dl}] = [0.468 \quad 0.044\ 1/\mathrm{s} \quad 0.443\ \mathrm{s}]$. Based on the defined feedback and learning controller gains, the convergence factor $\varepsilon_i = 0.6025$ for $0 < \Omega < 100\ Hz$ is obtained. The actual contour error Eq. (4.9) is calculated by Eqs. (4.11) - (4.17). According to the optimized learning gain and the ACEC, contour error compensation is conducted by Eq. (4.35) to modify the reference trajectory as in Eq. (4.34). The modified reference trajectory is applied to the original feedback controller which is modeled by

Runge-Kutta method with Couloumb friction as disturbance. Above process is done iteratively until no significant contour error reduction is observed.

4.7.2 Simulation Results

The simulation was conducted in 4 times iterations for ILCC to compare the performance with/without ACEC (with only estimated contour error in Eq. (3.5)). Simulation results are shown in Fig. 4.6 and Table 4.1 where in Figs. 4.6 (a) and (b) show the real trajectory for the ILCC without/with ACEC, respectively. Figs. 4.6 (c) and (d) show the contour error for the ILCC without/with ACEC, respectively. In all figures, the first iteration refers to the only feedback controller result. The feedback controller without any embedded system produces the largest contour error of 0.1082 mm. By applying the ILCC, contour error was reduced to 0.0214 mm which is equivalent to the error reduction of 80.22 %. The proposed controller exhibits the best performance as it has reduced the contour error to 0.0198 mm which is equivalent to an error reduction of 81.7 % and 7.5 % as compared to the feedback controller



(a) ILCC

(b) ILCC with ACEC

(c) ILCC

(d) ILCC with ACEC

FIGURE 4.6: Simulation results: (a) and (b) are real trajectory profiles; (c) and (d) are contour errors.

TABLE 4.1: Simulation results.

Controller	Contour Error [mm]	
	Max	Mean
Feedback controller	0.1082	0.0149
ILCC without ACEC	0.0214	0.013
ILCC with ACEC	0.0198	0.012

TABLE 4.2: Plant and control parameters.

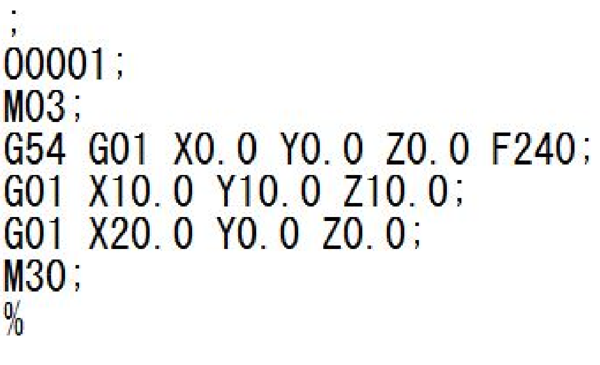
Parameter	X-Axis	Y-Axis	Z-Axis
m_i [$\times 10^{-4}$kgm^2]	9.62	13.99	11.3
c_i [kg/s]	0.24	0.24	0.24

and the ILCC, respectively.

4.8 Experiment

4.8.1 Experimental Condition

A three-axis commercial CNC machine (Fig. 2.6) with a ball-screw mechanism, which is attached on table and three servo motors. It is controlled by Mitsubishi M70 controller was used for the experiment to verify the effectiveness of the proposed controller. The actual position of X-Y-Z table was measured by linear encoders with resolution of 5 μm and a microcontroller ATMega16 attached to each axis for



```
;
O0001;
M03;
G54 G01 X0.0 Y0.0 Z0.0 F240;
G01 X10.0 Y10.0 Z10.0;
G01 X20.0 Y0.0 Z0.0;
M30;
%
```

FIGURE 4.7: 3D Sharp-corner trajectory G-code.

interface. The ILCC was programmed in Microsoft visual basic 6 in a separate personal computer (Windows OS) and embedded to the machine by direct numerical control (DNC) system with an RJ45 connection. The experimental interface design and system parameters are shown in Fig. 2.7 and Table. 4.2, respectively. The sampling rate was 5 ms and the considered operational frequency $0 < \omega < 100$ Hz with a discretization of 1 Hz. The optimal control gains $[K_P \quad K_I \quad K_D \quad K_{Pl} \quad K_{Il} \quad K_{Dl}] = [0.96 \text{ V/mm} \quad 0.01 \text{ V/smm} \quad 8 \text{ Vs/mm} \quad 0.468 \quad 0.044 \text{ 1/s} \quad 0.443 \text{ s}]$.

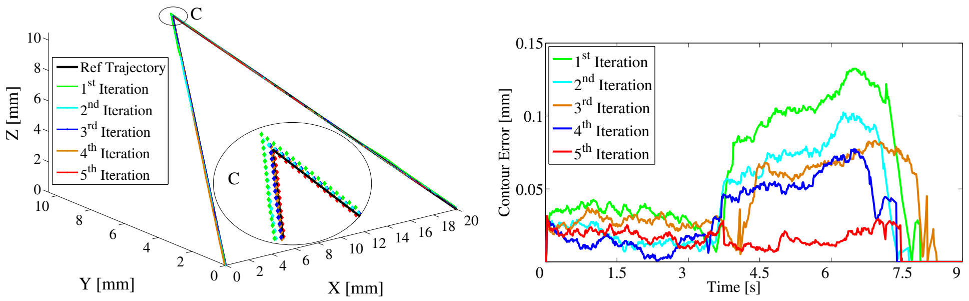
The following three types of controllers were considered for the experiment:

1. ILCC (with estimated contour error),
2. ILCC with ACEC, and
3. ILCC with ACEC and BRT.

For each controller, two types of reference trajectories were executed, 3D sharp-corner trajectory and 3D half-circular trajectory, which were implemented in G-code form.

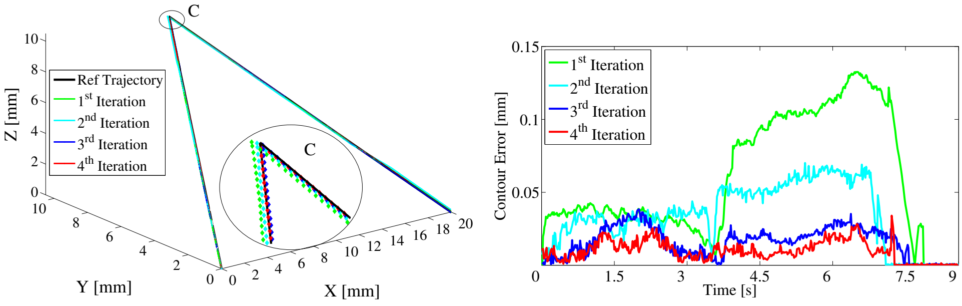
4.8.2 Experimental Results for 3D Sharp-Corner Trajectory

The 3D sharp-corner trajectory is represented in G-code form as shown in Fig. 4.7. For each controller, five-times experiment was conducted until the minimum contour



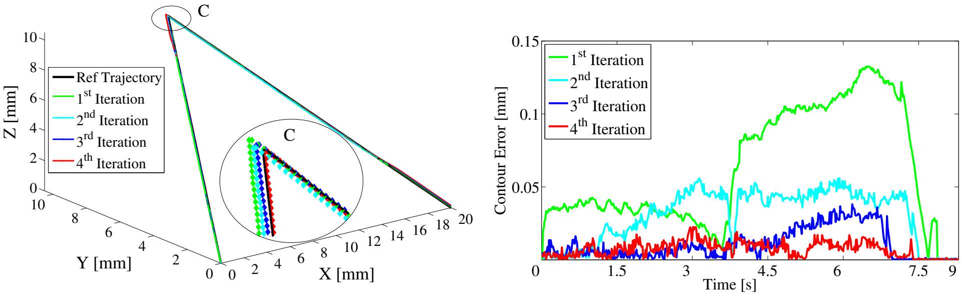
(a) Trajectory tracking profiles (b) Contour error

FIGURE 4.8: Experimental results for 3D sharp-corner trajectory by ILCC.



(a) Trajectory tracking profiles (b) Contour error

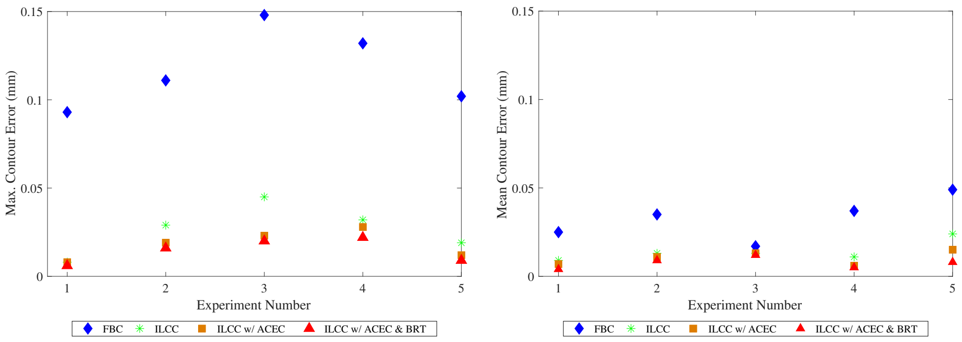
FIGURE 4.9: Experimental results for 3D sharp-corner trajectory by ILCC with ACEC.



(a) Trajectory tracking profiles (b) Contour error

FIGURE 4.10: Experimental results for 3D sharp-corner trajectory by ILCC with ACEC and BRT.

error was reached as shown in Figs. 4.8 - 4.11. Figs. 4.8 (a), 4.9 (a), and 4.10 (a) and Figs. 4.8 (b), 4.9 (b), and 4.10 (b) show the trajectory tracking profiles for the 3D sharp-corner trajectory and the contour errors based on the ILCC, ILCC with ACEC, and ILCC with ACEC and BRT, respectively. In each case, results of the first iteration



(a) Maximum contour error. (b) Mean contour error.

FIGURE 4.11: Contour error experimental result for 3D sharp-corner trajectory.

TABLE 4.3: Summary of experimental results: sharp-corner trajectory.

3D sharp-corner trajectory				
Experiment Number	e_c max [mm]			
	FBC	ILCC	ILCC + ACEC	ILCC + ACEC + BRT
1	0.093	0.008	0.008	0.006
2	0.111	0.029	0.019	0.016
3	0.148	0.045	0.023	0.02
4	0.132	0.032	0.028	0.022
5	0.102	0.019	0.012	0.009
Mean	0.1172	0.0266	0.018	0.0146
Reduction Ratio	0%	77.3%	84.64%	87.54%
Experiment Number	e_c mean [mm]			
	FBC	ILCC	ILCC + ACEC	ILCC + ACEC + BRT
1	0.025	0.009	0.007	0.004
2	0.035	0.013	0.011	0.009
3	0.017	0.013	0.013	0.012
4	0.037	0.011	0.006	0.005
5	0.049	0.024	0.015	0.008
Mean	0.0326	0.014	0.0104	0.0076
Reduction Ratio	0%	57.06%	68.10%	76.69%

refers to the FBC only. Under ILCC, the initial largest contour error was 0.1172 mm on average, which was reduced to 0.0266 mm in the 5th iteration as shown in Fig. 4.8 (b). On the other hand, under ILCC with ACEC, the final average maximum contour error was 0.018 mm in the 4th iteration, which is equivalent to the reduction of 32.33 % of the maximum contour error from that of ILCC. In the last controller, ILCC with ACEC and BRT, the smallest maximum contour error of 0.0146 mm on average was achieved as show in Fig. 4.10 (b). This is equivalent to the reduction of 45.11 % and 18.89 % of the maximum contour error compared to the ILCC and the ILCC with ACEC. The experimental maximum contour error is greater than the simulation result because of uncertain disturbance in the experimental machine.

Figs. 4.11 (a) and (b) respectively show the comparisons of the maximum and mean contour errors for the five-times experiments of the three controllers. It can be noted that the proposed controller, ILCC with ACEC and BRT yielded smallest values of the maximum and mean contour errors. Details of the experimental results are summarized in Table 4.3.

4.8.3 3D Half-Circular Trajectory Experiment Result

The 3D half-circular trajectory in G-code form was generated by the following equation:

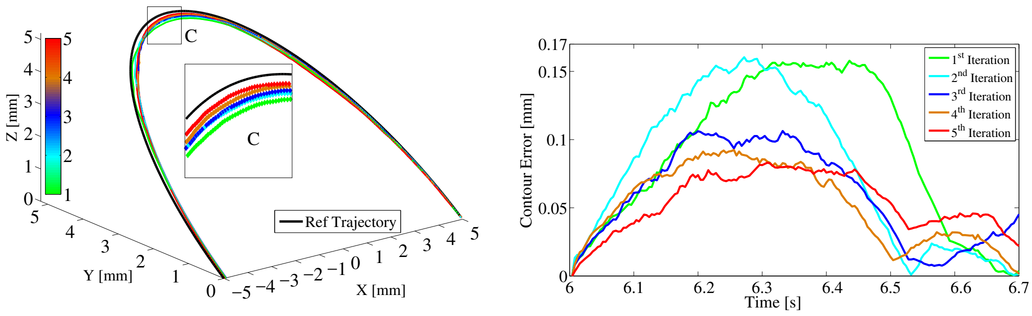
$$q_x = 5\cos\left(\frac{\pi}{12}t\right)\ \text{mm}, \qquad q_y = q_z = 5\sin\left(\frac{\pi}{12}t\right)\ \text{mm}. \tag{4.44}$$

Similar to the 3D sharp-corner trajectory, five-times experiment was conducted for each controller until the minimum contour error was reached as shown in Figs. 4.12 - 4.15. Figs. 4.12 (a), 4.13 (a), and 4.14 (a) show the trajectory tracking profiles while Figs. 4.12 (b), 4.13 (b), and 4.14 (b) show the contour error based on the ILCC, ILCC with ACEC, and ILCC with ACEC and BRT, respectively. Here also, results of the first iteration refers to the FBC only. Under ILCC, the initial maximum contour error was 0.1814 mm on average and was reduced to 0.0618 mm in the 5$^{\text{th}}$ iteration as shown in Fig. 4.12 (b). By using ILCC with ACEC, the final average maximum contour error was 0.0516 mm in the 4$^{\text{th}}$ iteration, which is equivalent to the reduction 16.5 % of the maximum contour error compared to that of ILCC. On the other hand, by using ILCC with ACEC and BRT, the smallest maximum contour error of 0.041 mm obtained on average as show in Fig. 4.14 (b). This is equivalent to the reduction of the maximum contour error by 33.66 % and 20.54 % compare to ILCC and ILCC with ACEC, respectively. All of experiments show that all of ILCC generates oscillated contour error profile and reaches zero contour error slowly after contour error peak appeared as shown in Fig. 4.14, because ILCC does not modify control input voltage directly but enforce the reference trajectory only. Hence, oscillated profile may increase the mean of contour error, although the maximum contour error is more reduced.

Comparison of the maximum and mean contour errors of the three controllers in five-times experiments are shown by Figs. 4.15 (a) and (b), respectively. The average mean and maximum contour errors based on ILCC with ACEC and BRT were respectively reduced by 65.54 % and 77.4 % compared to FBC. The rest of the experimental results are summarized in Table 4.4, where the tendency of error reduction is similar in both the 3D sharp-corner and half-circular trajectories. However, maximum contour error in half-circular trajectory is greater than maximum contour error in 3D sharp-corner. Because an ACEC with linear interpolation estimates insufficient contour error on circular trajectory so this method is not effective for circular or free-form trajectories. It will be improved by circular interpolation which is proposed in the next chapter.

4.9 Conclusion

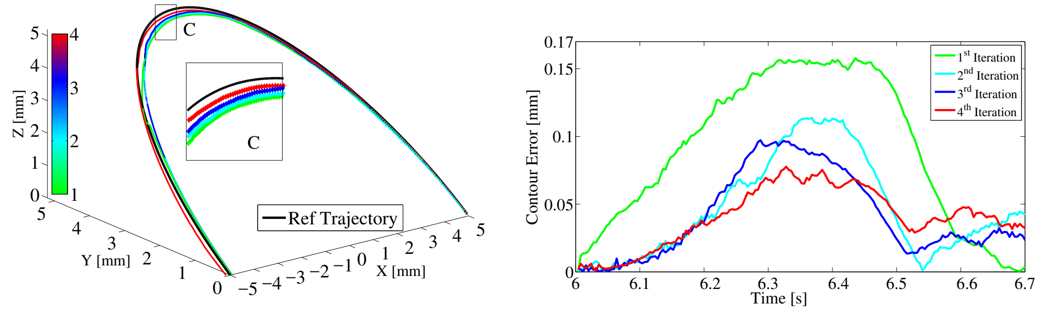
This study proposed a method to enhance the tracking performance of machine tool feed drive systems using ILCC with ACEC and BRT. Its performance was verified experimentally through 3D sharp-corner and half-circular trajectories in a commercial three-axis CNC machine. By comparing to the traditional FBC, proposed method



(a) Trajectory tracking profiles

(b) Contour error

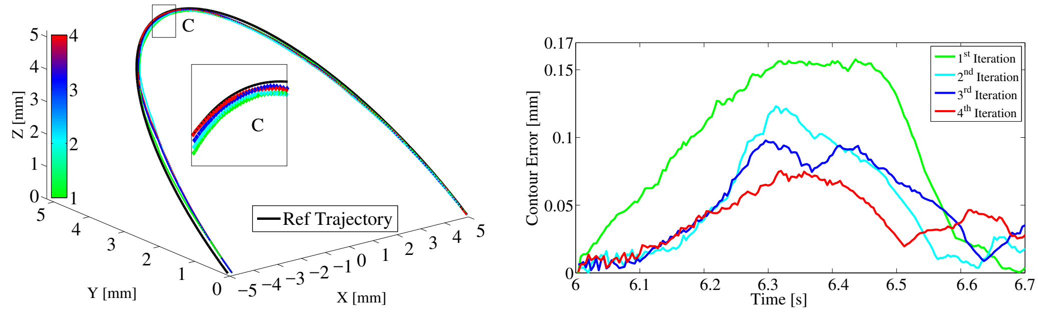
FIGURE 4.12: Experimental results for 3D half-circular trajectory by ILCC.



(a) Trajectory tracking profiles

(b) Contour error

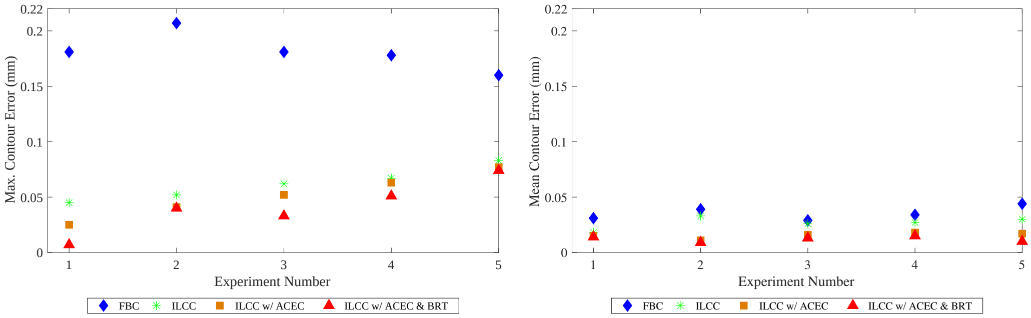
FIGURE 4.13: Experimental results for 3D half-circular trajectory by ILCC with ACEC.



(a) Trajectory tracking profiles

(b) Contour error

FIGURE 4.14: Experimental results for 3D half-circular trajectory by ILCC with ACEC and BRT.



(a) Maximum contour error.

(b) Mean contour error.

FIGURE 4.15: Contour error experimental result for 3D half-circular trajectory.

performance can be reduced maximum and mean contour error by up to 87.54 % and 76.69 % on average, respectively. The experimental results of the proposed controller was compared to the ILCC with estimated contour error, and ILCC with ACEC, where the proposed controller revealed that the maximum and mean contour errors

TABLE 4.4: Summary of experimental results: half-circular trajectory.

3D half-circular trajectory				
Experiment Number	e_c max [mm]			
	FBC	ILCC	ILCC + ACEC	ILCC + ACEC + BRT
1	0.181	0.045	0.025	0.007
2	0.207	0.052	0.041	0.04
3	0.181	0.062	0.052	0.033
4	0.178	0.067	0.063	0.051
5	0.16	0.083	0.077	0.074
Mean	0.1814	0.0618	0.0516	0.041
Reduction Ratio*	0%	65.93%	71.55%	77.40%
Experiment Number	e_c mean [mm]			
	FBC	ILCC	ILCC + ACEC	ILCC + ACEC + BRT
1	0.031	0.018	0.015	0.014
2	0.039	0.033	0.011	0.009
3	0.029	0.026	0.016	0.013
4	0.034	0.027	0.018	0.015
5	0.044	0.03	0.017	0.01
Mean	0.0354	0.0268	0.0154	0.0122
Reduction Ratio*	0%	24.29%	56.5%	65.54%

can be reduced by up to 45.11 % and 54.48 % and 20.54 % and 26.92 % on average compared to ILCC with approximated contour error and ILCC with ACEC respectively. However, the ACEC with linear interpolation is not effective for circular trajectory and the performance BRT smoothing algorithm improves ILCC with ACEC only is not significantly. The ILCC with ACEC with circular interpolation and without BRT will be designed and implemented to an advanced machine tool technology such as a CNC machine tool with a linear motor mechanism in the next chapter.

Chapter 5

Application of ILCC to a CNC Machine Tool with Linear Motors

This chapter proposes a design of an iterative learning contouring controller (ILCC) by considering actual contour error compensation (ACEC) to enhance the contouring performance of CNC machine tool feed drive systems with linear motors. The ACEC with linear and circular interpolation is designed to estimates contour errors precisely. The proposed control iteratively modifies the numerical control (NC) programs for each drive axis to reduce contour errors. Hence, the proposed approach can be directly applied for a commercial CNC machine tool with linear motors currently in use without any modification of their original controllers. Both linear and circular interpolation are verified by simulation in both "air-cutting" and machining conditions. The simulation is conducted for a non-smooth rhombus and circular trajectory. The effectiveness of the proposed methods have been experimentally verified through a CNC machine tool with linear motors for a non-smooth rhombus trajectory. Experimental results show that the proposed controller could reduce the maximum and mean contour errors by 94.58 % and 88.67 % on average, respectively. The proposed method improved the control input variance by 37.9 %, and consequently energy consumption was reduced by 11.7 % compared to the original NC program.

5.1 Introduction

Highly accurate mechanical components are widely required in many applicable fields. The CNC machining process has advantages of high production efficiency, high machining precision, and stable product quality. Achieving a high precision motion in machining largely depends on the control performance of each individual feed drive axis [55]. Under independent drive axial control, load disturbance or performance variance of either axis causes an increase of contour error [56]. Much efforts to reduce contour errors have been undertaken through either the design of advanced feedback controllers for each feed drive or by modifying the reference axial positions [71]. This modification has attracted many researchers in the machine tool community because it is directly related to product quality.

Two typical servo drives are used in CNC machine tools; linear servo motors and rotary motors with ball-screws. Accuracy and deviation with linear servo motors are much better than those with conventional rotary drives, especially at high feed rates and around points where feed direction is changed [76]. It is recognized that machine tools with direct linear drives claim noteworthy advantages over conventional ball-screw machines in speed, consumption, and accuracy [77], [78]. Hence, further research of control techniques for linear motors are highly required.

To improve the machine performance of a CNC machine with linear servo motors, many control researches were proposed. The implementation of H_∞ optimal feedback controller for large stiffness and closed-loop tracking performance was introduced to reduce the effect of machining forces on the tool position in a linear servo drive [79]. The minimum-time path optimization (MTPO) and minimum-time tracking control (MTTC) are applied for linear motors to accommodate the dominating saturation effect in high-speed machine tools [80]. The motion behaviour of the direct linear drive is improved by compensation of disturbances like the friction and cogging forces [81]. A virtual friction compensator and a design method for control systems is proposed to improve the response characteristics of linear motor feed drive systems. It ables to cancel the real nonlinear friction of feed drive systems by using the nonlinear friction model [82]. The periodic tracking error is eliminated by the proposed repetitive control. It increases the stabilizing range and enhances performance by adopting the prototype repetitive controller design principle [83]. The cross-coupled iterative learning control (CCILC) improved the motion control performance of a linear motor experimental system by combining individual axial iterative learning control (ILC) and a cross-coupled controller which modifies the control signals and improves contour trajectory tracking performance [23]. By applying the ILC, the convergence of the output error is guaranteed under certain conditions even when the system parameters are not known exactly or under the existence of bounded unknown external disturbances [25]. Furthermore, the contour error can be significantly reduced by the ILC which considers both tracking and contour errors [42].

The ILC with contour error approximation base coordinate transformation was proposed in [64], which could be applied to a commercial machine without any modification of the existing controller. The above method uses approximated contour errors instead of actual ones to design contouring controllers or modify reference trajectories, so the error approximation methods do not represent the contour error accurately, especially for a high-curvature trajectory application. However, contouring control with actual control error is difficult to be implemented in real-time control systems because the actual contour error cannot be immediately obtained for complex contour profiles. Actual contour error compensation (ACEC) is proposed for the ILCC application by calculating the minimum distance between the actual position and the reference trajectory, where it is used to achieve the desired contour precisely by linear interpolation [84]. However, the contour error estimation is not effective for circular or free-form trajectories and this method requires an additional measurement device to implement in commercial CNC machine tools and precision plant dynamics information to achieve higher performance. The plant dynamics identification part is not included in the previous study. The novelty of this study is that the proposed approach studied in [84] is extended for a CNC machine tool with linear motors, and ACEC with quadratic estimation, system identification method and consumed energy evaluation are presented. The proposed method modifies the original trajectory (NC program) through the ACEC as a learning controller. The proposed method is proven by simulation and experiment to enhance machine tool control performance.

5.2 Literature Review

Because the contour error is caused by single axial tracking errors [85], many researchers improve contouring performance indirectly by reducing tracking error. For example, a zero-phase error-tracking controller (ZPETC) is proposed by canceling all the closed-loop poles and assuring zero phase shift for all dominant frequencies, and thus the tracking-error can be effectively reduced to a large degree [86]. However, the gain error is increased due to the higher frequency, which also degrades tracking accuracy, especially for a high feed rate. The feed-forward terms are added to compensate for gain errors. The ZPETC is extremely sensitive to plant parameter variance because the feedforward algorithm is typically an inverse of the control system [87]. There are also other methods to improve tracking performance such as model-based control [88], adaptive controller-parameter adjustment [89], iterative learning control [90], disturbance-observer based control [91], and metaheuristic algorithms for gain tuning [92]. Although the axial tracking error causes the contour error, an improvement of the tracking error does not lead to efficient reduction of the contour error [85].

To reduce the contour error directly, many cross-coupling control (CCC) methods have been proposed. The first one is proposed to improve the contouring performance by coupling the individual axis error based on contouring error estimation [19]. Then, the CCC is designed as a proportional compensator in a biaxial feed drive application

[18]. Many studies have been made to reduce the contour error based on combination of the CCC with several control algorithms [23], [58], [93]–[96]. All the above CCC methods are designed to reduce the contouring error in a feedback manner. Pre-compensation methods with contouring error prediction are proposed to control the contouring error before it occurs [97].

Another strategy suggested to reduce contour error is an interpolator design. The functions of CNC interpolators are based on feed rate scheduling and reference command generation. The feed rate scheduling for continuous short-line segment toolpaths with contour error constraints is proposed [98]. The contour error is normally varied and the maximum error exceeds error constraint values because the feed drive system model is simplified as a first-order model. To reduce modeling error effects, high order models with a feedforward controller, a dynamic-based NURBS interpolator, and a NURBS interpolator with a constant feed rate are proposed [99]. Because the system model and NURBS formula are not simplified, it may be difficult to implement in a real time manner.

All the above-mentioned methods require access to a CNC controller for implementation. The control approach that reduces resultant contour errors in real time, cannot be implemented for existing commercial CNC machines generally. For this reason, offline contour error reduction methods are proposed. Zhang et al. reduced the contour error based on a spline trajectory generation technique under jerk constraints [100]. However, the feed rate typically needs to be decreased. In order to address this, iterative tracking error pre-compensation was presented to reduce the contour error indirectly without reducing the feed rate [101]. Many feed rate scheduling strategies have been proposed to reduce the contour error under contour error constraints such as the curvature evaluation based feed rate program, feed drive inverse dynamics, and feed rate offline prediction [102]–[105]. Therefore, once the axial dynamic models are not matched, the feed rate scheduling becomes an ineffective method to reduce the contour error.

The embedded iterative learning contouring controller (EILCC) with contour error approximation can iteratively modify the reference trajectory to improve contouring performance [61], [64]. Contour error estimation is a key aspect to achieve higher contour error reduction. The precise contour error estimation with linear interpolation has been developed and implemented by previous ILC principle on three axis commercial CNC machine tools [84]. However, the above ILC performance can not reach optimum performance because of an imprecise plant dynamic model and imprecise error estimation for a typical circular trajectory. Precise plant dynamic models and contour error estimation are required to improve control performance, so that the ILC can be enhanced by adding a precise plant dynamics model and an ACEC with circular interpolation. This study proposes ILC by modifying the NC program iteratively with precise contour error estimation (ACEC) and a precise plant dynamic model for a CNC machine tool feed drive system with linear motors. To prove the proposed method's performance, simulation was carried out with two desired trajectories; rhombus and circle. Then, experiment was conducted for a rhombus trajectory with sharp corners which produces a large contour error in each corner.

5.3 Methodology

In order to improve the contouring performance of commercial CNC machines, in which their controllers can not be accessed, the ILC by NC program modification is considered. The details of the proposed ILC are explained in Section 5.4. The NC program is modified by using the actual contour error with a PID structure to achieve smooth and fast contour error reduction. The actual contour error is estimated by minimizing the distance between actual and reference feed drive positions. However, the closest actual position with minimum distance to a reference contour is not in the orthogonal direction exactly. Hence, accurate motion cannot be achieved, especially for higher curvature contours. For this reason, the linear and circular interpolations are designed to calculate the contour error precisely. The reference position is described by the NC program, and the actual position is measured by a linear scale encoder and NC Analyzer 2 software [53].

In order to perform fast contour error reduction, the PID learning gain must be optimized by a suitable gain tuning method. Many gain tuning methods have been proposed currently. For example, artificial neural networks are used by simulating the structure or functional aspects of biological neural networks to construct a self-tuning PID controller [106]. However, the number of layers and the number of neurons per layer are often hard to be determined. The ant colony optimization is a solution of optimization problems based on the working behavior of an ant colony which searches for the shortest path to the food or solution. It was used to minimize a multi-objective function for PID tuning [107]. A simple genetic algorithm is applied for tuning PID controllers with the cascade control systems. The sum of integral absolute error values of the regulatory response is used as the objective function [108]. The convergence analysis provides PID learning compensator gains by guaranteeing contour error reduction in a cascade ILC system [62]. To reach fast convergence of the contour error in this study, the PID learning compensator gain is optimized by a contour error convergence analysis. It is solved by using "fmincon" function in MATLAB®. The convergence analysis requires a precise plant dynamic model which is obtained by control system identification. The reference command (NC progam) input and output signals are collected to identify the plant transfer function using system identification toolbox in MATLAB®. The control system identification procedure is shown in Section 5.5.

The reference trajectory is compensated by actual contour error compensation iteratively until convergence. The proposed method is proven by simulation and experiment in a commercial CNC machine with linear motors, which is explained in Sections 5.6 and 5.7, respectively. The performance of the ILC combined with linear and circular interpolations are compared under rhombus sharp corner and circular

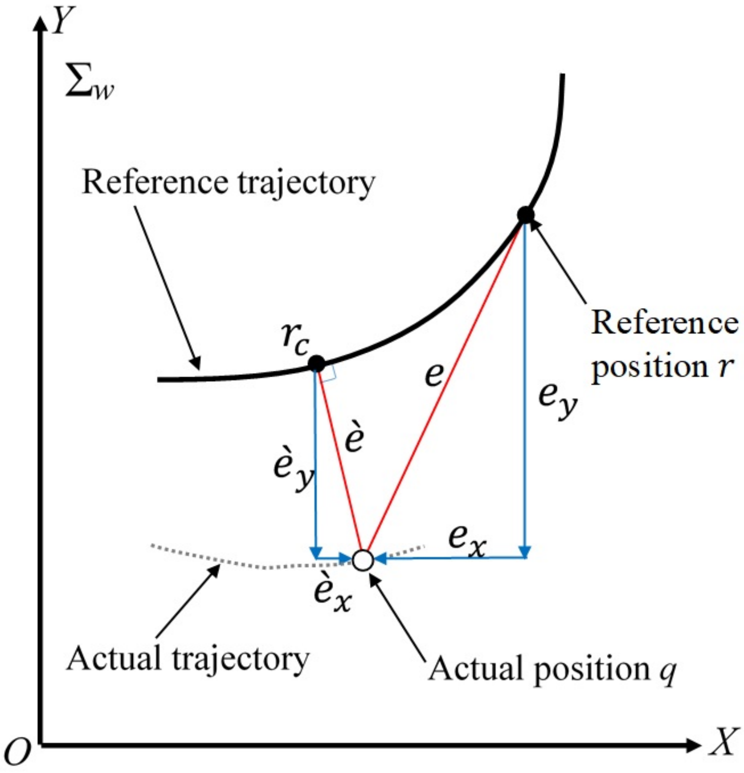


FIGURE 5.1: Tracking and contour errors.

trajectories. The experimental results are analyzed with the mean, maximum, and the root mean square (RMS) error values to confirm the effectiveness of the proposed approach. A similar analysis is done for energy consumption.

5.4 Control Design of ILCC with ACEC

This section modifies original ACEC [84] for implementation to a biaxial machine tools feed drive system with linear motors.

5.4.1 Definition of Contour Error

The tracking error in each axis is recognized as the difference between the reference and actual positions, while the contour error is defined as the shortest distance between the actual and desired contours. The tracking error and contour error in a feed drive system are illustrated in Fig. 5.1. The reference position at time t is denoted by $r = \begin{bmatrix} r_x & r_y \end{bmatrix}^T$ with respect to the fixed coordinate frame $\sum_w$. The tracking error e and its magnitude are defined as

$$e_w = \begin{bmatrix} e_x & e_y \end{bmatrix}^T = r - q, \tag{5.1}$$

$$e = \|e_w\|. \tag{5.2}$$

The closest position of the desired contour to q is denoted by $r_c = \begin{bmatrix} r_{cx} & r_{cy} \end{bmatrix}^T$, and the actual contour error $\grave{e}$, which is the shortest distance between the actual position q and the reference trajectory, is calculated as follows:

$$e_c = \begin{bmatrix} e_{cx} & e_{cy} \end{bmatrix}^T = r_c - q, \tag{5.3}$$

$$\grave{e} = \|e_c\|. \tag{5.4}$$

5.4.2 Actual Contour Error Compensation

This section modifies the original ACEC [84] for implementation to a biaxial feed drive system with linear motors. ACEC is a direct method to compensate for the actual contour error searching about the closest position r_c in the reference trajectory to q as shown in Fig. 5.2. The reference trajectory is represented by discrete points at each sampling period as $\grave{r}_c$, $\grave{r}_{c+1}$, ... First, the closest reference position among these points to q denoted by $\grave{r}_c$, is obtained by finding the minimum distance $\grave{d}_c$ between the actual position q and reference positions from n^{th} to $(n-m)^{\text{th}}$ ones, where m is predefined $(0 \le m \le n)$ and n is the current reference position number.

$$\grave{d}_c = \min_{0<m<n} \|e_{cm}\|, \tag{5.5}$$

where

$$e_{cm} = \grave{r}_{n-m} - q. \tag{5.6}$$

In this study, the closest position r_c is determined using linear or circular interpolations to generalize the proposed approach.

Linear Interpolation

The closest position r_c is determined by linear interpolation of the neighborhood points as shown in Fig. 5.2 with equation below:

$$r_c = \grave{r}_c + \frac{(\grave{r}_{c+1} - \grave{r}_c)\,D_1}{D_2}, \tag{5.7}$$

$$D_1 = \frac{\grave{d}_c^2 - \grave{d}_{c+1}^2 + \|\grave{r}_{c+1} - \grave{r}_c\|^2}{2\,\|\grave{r}_{c+1} - \grave{r}_c\|}, \tag{5.8}$$

$$D_2 = \|\grave{r}_{c+1} - \grave{r}_c\| - D_1, \tag{5.9}$$

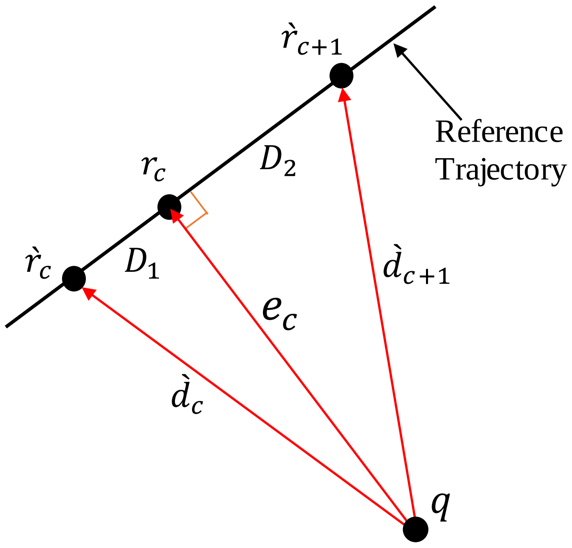


FIGURE 5.2: Contour error interpolation.

where $\grave{r}_{c+1}$, $\grave{d}_{c+1}$, D_1, and D_2 are the discrete positions next to the approximated discrete position $\grave{r}_c$, the shortest distance from $\grave{r}_{c+1}$ to the actual position q, the distance between $\grave{r}_c$ and r_c, and the distance from r_c to $\grave{r}_{c+1}$, respectively. In this estimation, $\grave{d}_{c+1}$ is assumed shorter than $\grave{d}_{c-1}$. If $\grave{d}_{c-1}$ is shorter than $\grave{d}_{c+1}$, $\grave{d}_{c+1}$ and $\grave{r}_{c+1}$ is replaced by $\grave{d}_{c-1}$ and $\grave{r}_{c-1}$ in Eqs. (5.7)-(5.9).

Circular Interpolation

The circular interpolation of neighborhood points is generated from the discrete reference position $\grave{r}_c$, the point before, and one after discrete reference position $\grave{r}_c$, $\grave{r}_{c-1}$, and $\grave{r}_{c+1}$, respectively. The circular interpolation is shown in Fig. 5.3. The circular center is defined by $C = \begin{bmatrix} C_x & C_y \end{bmatrix}$ and calculated as follows:

$$C_y = \frac{A_1 A_3 - A_2 A_4}{2\left(A_2 A_5 - A_1 A_6\right)}, \tag{5.10}$$

$$C_x = \frac{A_4 + 2 C_y A_5}{2 A_1}, \tag{5.11}$$

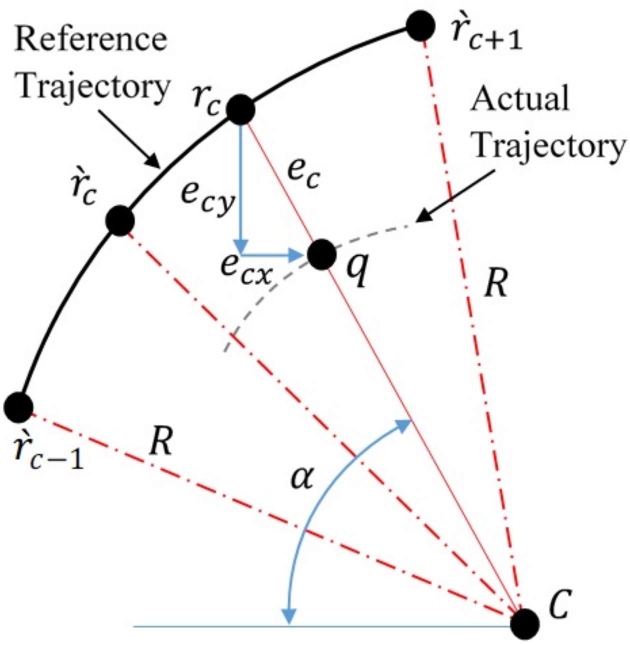


FIGURE 5.3: Contour error estimation by circular interpolation.

where

$$\begin{aligned} A_1 &= \grave{r}_{cx} - \grave{r}_{c-1x}, & A_5 &= \grave{r}_{c-1y} - \grave{r}_{cy}, \\ A_2 &= \grave{r}_{c+1x} - \grave{r}_{c-1x}, & A_6 &= \grave{r}_{c-1y} + \grave{r}_{c+1y}, \\ A_3 &= \grave{r}^2_{c+1x} - \grave{r}^2_{c-1x} + \grave{r}^2_{c+1y} - \grave{r}^2_{c-1y}, \\ A_4 &= \grave{r}^2_{cx} - \grave{r}^2_{c-1x} + \grave{r}^2_{cy} - \grave{r}^2_{c-1y}. \end{aligned}$$

The radius of curvature R is calculated as follows:

$$R = \|C - \grave{r}_c\|. \tag{5.12}$$

So the contour error magnitude is

$$\grave{e} = R - \|q - C\|, \tag{5.13}$$

The axial elements of contour error e_{cx} and e_{cy} are given by

$$\begin{bmatrix} e_{cx} \\ e_{cy} \end{bmatrix} = \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} e_c, \tag{5.14}$$

where

$$\alpha = \arctan\left(\frac{q_y - C_y}{q_x - C_x}\right). \tag{5.15}$$

5.4.3 Determination of Number of Discrete Points

The modified trajectory has a higher number of discrete points than the original reference trajectory considering the sampling time and actual velocity. The high number of discrete points increases the operation time because of acceleration and deceleration between each point. Therefore, the number of points is re-calculated based on the desired distance l_s for discretization considering input resolution in NC codes as

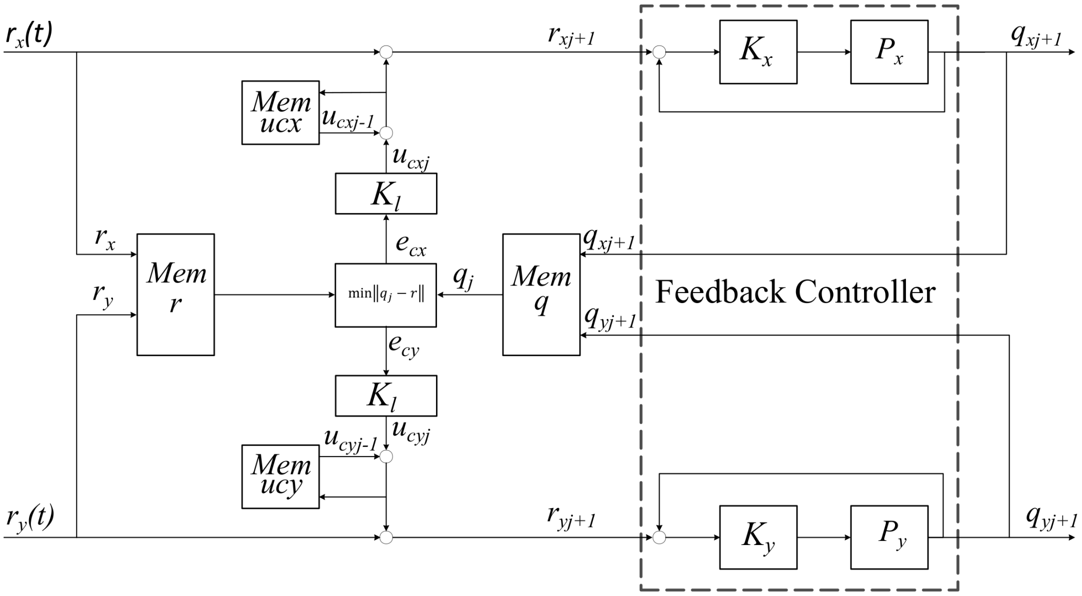


FIGURE 5.4: Proposed control system design.

follows:

$$n_s = \frac{\sum_{M=1}^{N} \|r_M - r_{M-1}\|}{l_s}, \tag{5.16}$$

$$\widehat{r}_n = r_N, \text{ if } \sum_{M=1}^{N} \|r_M - r_{M-1}\| = l_s n, \tag{5.17}$$

$$1 \leq n \leq n_s.$$

where n_s, N, and $\widehat{r}_n$ are re-defined number of discrete points, original number of discrete points, re-defined trajectory position.

5.4.4 Proposed Controller Design

The proposed controller consists of a feedback controller (FBC) and an iterative learning contouring controller with ACEC. The reference input of the FBC is the reference trajectory modified by the ILCC with ACEC as shown in Fig. 5.4, where *mem* u_c, q, and, r are the memories of learning compensation values, actual position, and reference trajectory, respectively. The closed loop transfer function of the system plant in s-domain is represented as follows:

$$G_i(s) = \frac{P_i(s) K_i(s)}{1 + P_i(s) K_i(s)}, \tag{5.18}$$

where G_i and K_i are the machine plant and the feedback compensator for the i^{th} axis, respectively. In this study, $K_i(s)$ is assumed to be a PID compensator as follows:

$$K_i(s) = K_{Pi} + \frac{1}{s} K_{Ii} + s K_{Di}, \tag{5.19}$$

where K_{Pi}, K_{Ii}, and K_{Di} are the proportional, integral, and derivative gains for the i^{th} axis, respectively. $r_{ij+1}(t)$ is the modified reference trajectory for the $(j+1)^{\text{th}}$ iteration in the i^{th} axis using an iterative learning compensator K_l with ACEC as follows:

$$r_{ij+1}(t) = r_{ij}(t) + u_{cij}(t), \tag{5.20}$$

$$\begin{aligned} u_{cij}(t) &= K_{li}(s) e_{cij}(t), \\ K_{li}(s) &= K_{Pli} + \frac{1}{s} K_{Ili} + s K_{Dli}, \end{aligned}$$

where r_{ij}, u_{cij}, e_{cij}, K_{Pli}, K_{Ili}, and K_{Dli} are reference signal, control input, contour error for the j^{th} iteration, the proportional, integral, and derivative learning compensator gain K_l for the i^{th} axis, respectively. Therefore,

$$u_{cij}(t) = K_{Pli} e_{cij}(t) + K_{Ili} \int_0^t e_{cij}(\tau)\, d\tau + K_{Dli} \dot{e}_{cij}(t). \tag{5.21}$$

leads to

$$r_{ij+1}(t) = r_{i1}(t) + \sum_{n=0}^{j} u_{cin}(t). \tag{5.22}$$

5.4.5 Convergence Analysis

The ILCC performance with ACEC can be guaranteed through convergence analysis to ensure reduction of the contour error. The convergence condition of ILC is represented as

$$\varepsilon_i(t) = \left\| \frac{e_{cij+1}(t)}{e_{cij}(t)} \right\| < 1, \tag{5.23}$$

where $\varepsilon_i(t)$ is the convergence factor for the i^{th} axis. Based on the cascade ILC [62], the convergence can be confirmed when the error magnitude of the $(j+1)^{\text{th}}$ iteration is smaller than one of the j^{th} iteration. From Eqs. (5.3) and (5.5), the convergence factor is derived as follows:

$$\begin{aligned}
e_{cij+1} &= \min_{0\le m\le n}\left[r_{mi} - q_{ij+1}\right], \\
&= \min_{0\le m\le n}\left[r_{mi} - G_i(s)\left(r_{mij} + u_{cij}\right)\right], \\
&= \min_{0\le m\le n}\left[r_{mi} - G_i(s)\, r_{mij+1}\right], \\
&= \min_{0\le m\le n}\left[r_{mi} - G_i(s)\left(r_{mij} + u_{cij}\right)\right], \\
&= \min_{0\le m\le n}\left[r_{mi} - G_i(s)\, r_{mij} - G_i(s) K_{li}(s)\, e_{cij}\right], \\
&= \min_{0\le m\le n}\left[r_{mi} - G_i(s)\, r_{mij}\right] - G_i(s) K_{li}(s)\, e_{cij}, \\
&= e_{cij} - G_i(s) K_{li}(s)\, e_{cij}, \\
&= \left[1 - G_i(s) K_{li}(s)\right] e_{cij}, \\
\frac{e_{cij+1}}{e_{cij}} &= 1 - \frac{P_i(s) K_i(s) K_{li}(s)}{1 + P_i(s) K_i(s)},
\end{aligned} \tag{5.24}$$

substituting Eq. (5.24) into Eq. (5.23) leads to

$$\varepsilon_i(s) = \left\| 1 - \frac{P_i(s) K_i(s) K_{li}(s)}{1 + P_i(s) K_i(s)} \right\| < 1. \tag{5.25}$$

Considering the discrete-time form for the implementation, $P_i(s)$ is represented as follows:

$$P_i\left(z^{-1}\right) = \frac{1}{m_i\left(\frac{z^{-1}-1}{t_s z^{-1}}\right)^2 + c_i \frac{z^{-1}-1}{t_s z^{-1}}}, \quad z = e^{j\omega t_s}, \tag{5.26}$$

where ω, z, and z^{-1} are the angular frequency, the z-domain variable, and the delay operator, respectively. The feedback and learning compensation in the z-domain is represented as, follows:

$$K_i\left(z^{-1}\right) = K_{Pi} + K_{Ii}\left(\frac{t_s z^{-1}}{z^{-1}-1}\right) + K_{Di}\left(\frac{z^{-1}-1}{t_s z^{-1}}\right), \tag{5.27}$$

$$K_{li}\left(z^{-1}\right) = K_{Pli} + K_{Ili}\left(\frac{t_s z^{-1}}{z^{-1}-1}\right) + K_{Dli}\left(\frac{z^{-1}-1}{t_s z^{-1}}\right). \tag{5.28}$$

The convergence speed depends on the parameter ε, which has to be kept as minimum as possible to obtain the fast convergence with a delay factor η due to the system delay. Considering Eq. (5.25) and the delay period, the following objective function is considered:

$$J_i = \min_{K_{Pli},\, K_{Ili},\, K_{Dli}} \left\| 1 - \frac{z^{\eta} P_i\left(z^{-1}\right) K_i\left(z^{-1}\right) K_{li}\left(z^{-1}\right)}{1 + P_i\left(z^{-1}\right) K_i\left(z^{-1}\right)} \right\|_{\infty}, \tag{5.29}$$
$$\forall \omega \in \Omega,$$

In order to find the learning compensator gains, the previous equation (3.25) has been solved using “fmincon” function in the MATLAB® toolbox.

5.5 Control System Identification for a CNC Machine

In this study, control system identification is conducted to obtain the transfer function Eq. (5.18) of a CNC machine tool from reference trajectory to actual trajectory. The transfer function is used to calculate the learning compensator gain in Eq. (5.29). Fig. 5.5 shows the linear motor CNC machine tool DMG MORI NV1500 used for the experimental setup. Based on the dynamics model of a CNC machine tool with the linear motors as shown in Fig. 5.6, the dynamics model is represented as follows:

$$m_i \frac{d^2 q_i}{dt} + c_i \frac{dq_i}{dt} + k_i q_i = f_i, \quad i = x,\, y. \tag{5.30}$$

where m_i, q_i, c_i, k_i, and f_i are the total mass, position, viscous friction coefficient, stiffness, and the driving force in the i^{th} axis, respectively. The transfer function from driving force to position is derived as follows

$$\frac{Q_i(s)}{F_i(s)} = \frac{1}{m_i s^2 + c_i s + k_i}, \tag{5.31}$$

$$m_x = M_t + Mx, \qquad m_y = m_x + M_y.$$

where Q_i and F_i are the actual position and the driving force, respectively. M_t, M_x, and M_y are table mass, motor mass in the x and y axis, respectively. Based on Fig. 5.7, transfer function $P_i(s)$ from control input $U_i(s)$ to actual position $Q_i(s)$ is obtained by multiplying motor and amplifier gains to the driving force transfer function as follows:

$$P_i(s) = \frac{Q_i(s)}{U_i(s)} = \frac{K_a K_t}{m_i s^2 + c_i s + k_i}. \tag{5.32}$$

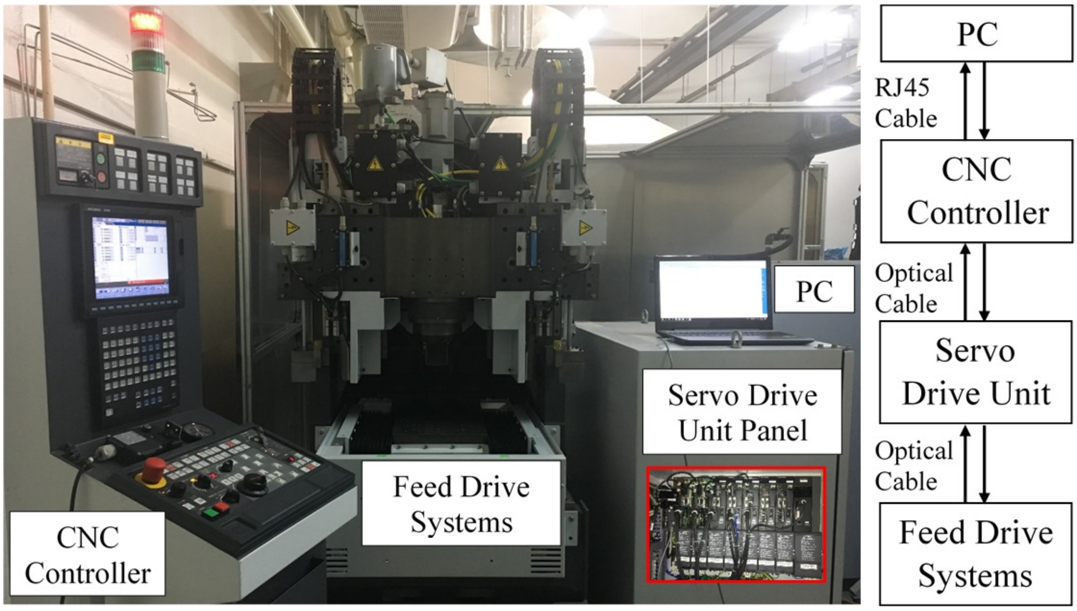


FIGURE 5.5: Experimental CNC machine tool (DMG MORI NV1500).

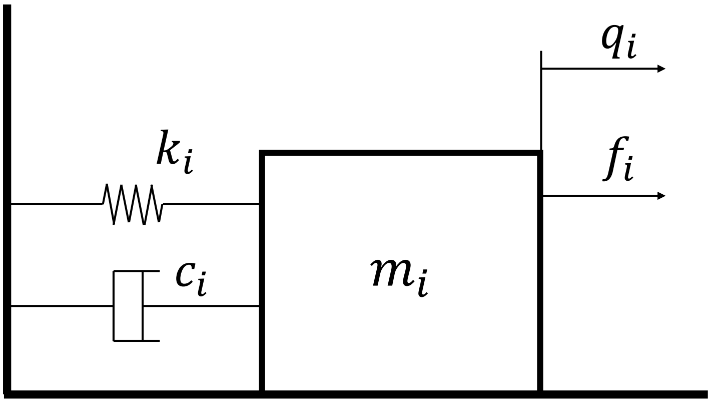


FIGURE 5.6: Dynamic model of CNC machine tool with linear motor.

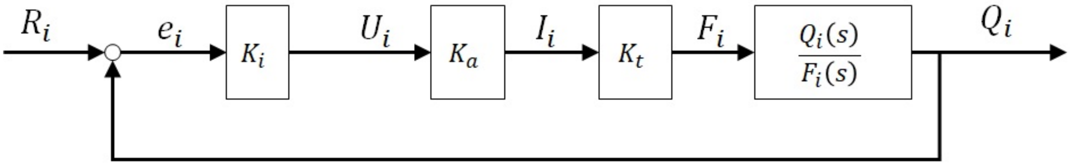


FIGURE 5.7: Assumed existing feedback controller scheme.

where K_a and K_t are amplifier and motor gains, respectively. Because the details of the existing controller is unknown, we assume that the general PID feedback controller shown in Fig. 5.7 is implemented in the CNC machine tool. By substituting Eqs. (5.19) and (5.32) to Eq. (5.18), the transfer function G_i is obtained as follows:

$$G_i\left(s\right) = \frac{B_1 s^2 + B_2 s + B_3}{B_4 s^3 + B_5 s^2 + B_6 s + B_7}. \tag{5.33}$$

where

$$\begin{aligned} B_1 &= K_a K_t K_{Di}, & B_5 = K_a K_t K_{Di} c_i, \\ B_2 &= K_a K_t K_{Pi}, & B_6 = K_a K_t K_{Pi} k_i, \\ B_3 &= K_a K_t K_{Ii}, & B_7 = A_3, \\ B_4 &= m_i, & \end{aligned}$$

The CNC controller Mitsubishi M700V provides 1 nm input resolution for the reference trajectory. The measurement results provided through Melsoft NC Analyzer 2 software [53] are used to identify the system model. After collecting reference command inputs and actual position output signals, the system identification toolbox in MATLAB® is used to estimate the plant transfer function in Eq. (5.33). The transfer functions from the reference command input to the actual position for x and y axes are obtained as follows:

$$G_x(s) = \frac{122.2s^2 + 4.745s + 2.296 \times 10^4}{s^3 + 122.5s^2 + 194.1s + 2.294 \times 10^4}. \tag{5.34}$$

$$G_y(s) = \frac{195.7s^2 + 186.6s + 3.868 \times 10^4}{s^3 + 198s^2 + 384.1s + 3.883 \times 10^4}. \tag{5.35}$$

Based on Eqs. (5.34) and (5.35), the bode diagrams of x and y axis are plotted in Figs. 5.8 and 5.9, respectively. Applying the zero-order hold and sampling rate of 1.7 ms, the pulse transfer functions are obtained as follows:

$$G_x\left(z^{-1}\right) = \frac{0.1875z^{-1} - 0.3749z^{-2} + 0.1875z^{-3}}{1 - 2.811z^{-1} + 2.623z^{-2} - 0.812z^{-3}}. \tag{5.36}$$

$$G_y\left(z^{-1}\right) = \frac{0.2827z^{-1} - 0.5648z^{-2} + 0.2822z^{-3}}{1 - 2.713z^{-1} + 2.427z^{-2} - 0.7141z^{-3}}. \tag{5.37}$$

5.6 Simulation

5.6.1 Simulation Condition

In order to confirm the proposed method, the rhombus and circular trajectories have been conducted in addition to estimate the cutting force in simulation for each trajectory as shown in Figs. 5.10 and 5.12. The simulation of a bi-axial table was done by Matlab® software under a Windows OS. The considered operational frequency $0 \leq \omega \leq 100$ rad/s with a discretization of 1 rad/s due to each axial bode

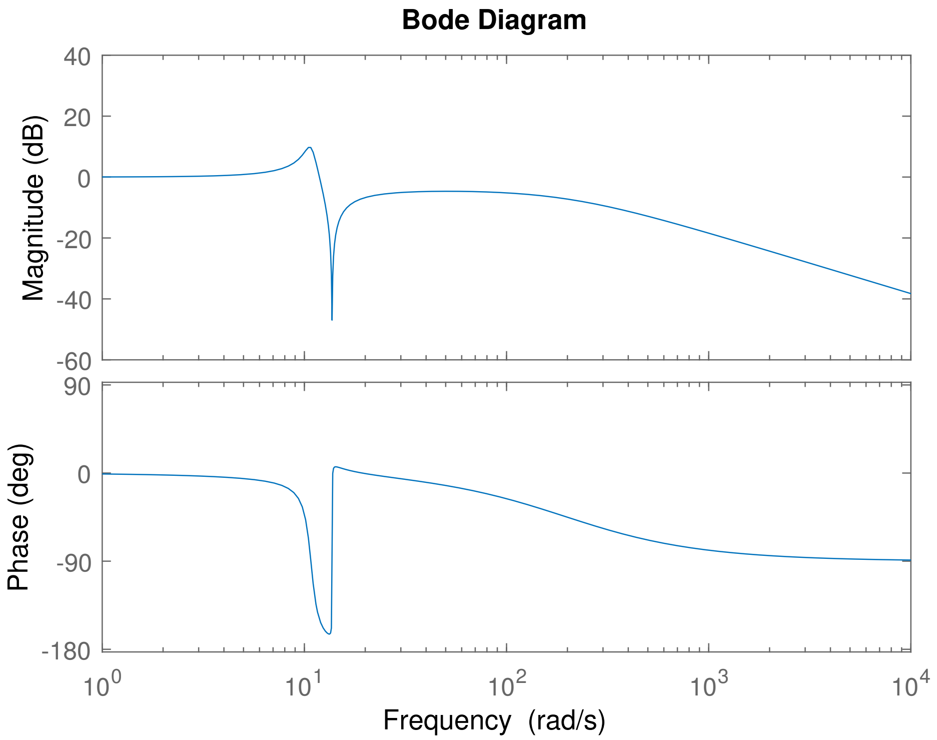


FIGURE 5.8: Bode plot for closed-loop system of x axial drive.

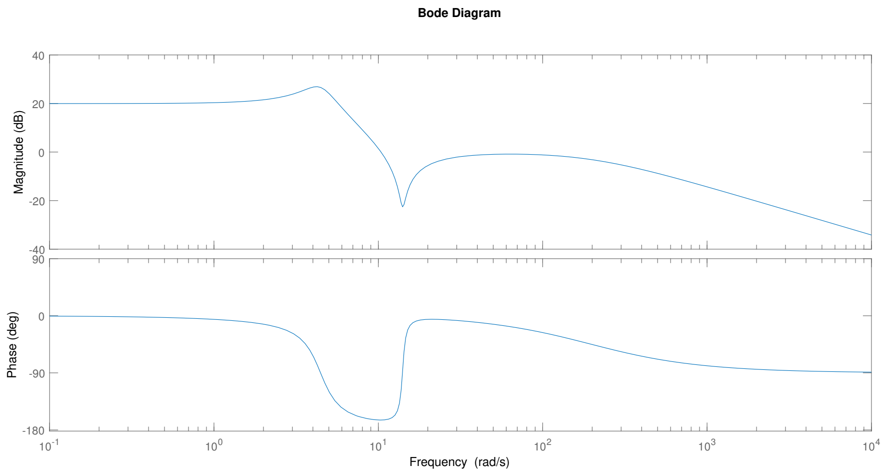


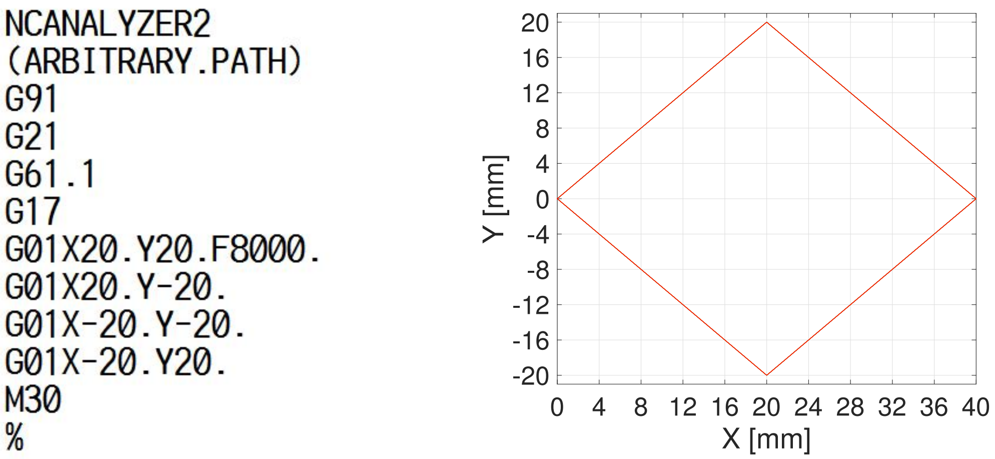
FIGURE 5.9: Bode plot for closed-loop system of y axial drive.

diagram (Figs. 5.8 and 5.9). The optimal learning control gains $[K_{Pl} \quad K_{Il} \quad K_{Dl}] = [0.079 \quad 0.05\ 1/\mathrm{s} \quad 0.452\ \mathrm{s}]$ were obtained, which led to a convergence factor ε_i to be 0.752. The performance under linear and circular interpolations are compared for both rhombus and circular trajectories. Simulation is also performed under a cutting force condition to confirm the effectiveness of the proposed approach under machining conditions.

5.6.2 Simulation Results

Rhombus Trajectory

In order to evaluate the performance of the ACEC with linear interpolation in ILCC implementation. The rhombus with a typical sharp corner trajectory in Fig. 5.10 is designed as a reference trajectory because it is difficult for feed drive systems generally to track a sharp corner trajectory. The circular interpolation can not be applied for the rhombus motion trajectory because of its infinity radius properties. The simulation is conducted in 20 iterations. Simulation results are shown in Figs. 5.11 and 5.14, where Fig. 5.11 shows the contour error profile for all iterations. It has three peaks of contour error values on each corner area. Fig. 5.14 shows the convergence profiles of the maximum and mean contour errors. In each figure, the first iteration refers to the original control which has the largest contour error of 4.535 μm and the mean contour error of 0.398 μm, respectively. By applying ACEC with linear interpolation, the maximum contour error was reduced to 0.398 μm and 0.003 μm for the mean contour error, which is equivalent to a reduction by 91.22 % and 95.95 %, respectively. There is no significant error reduction after the 9^{th} iteration. All of the simulation results are summarized in Table 5.1.



```
NCANALYZER2
(ARBITRARY.PATH)
G91
G21
G61.1
G17
G01X20.Y20.F8000.
G01X20.Y-20.
G01X-20.Y-20.
G01X-20.Y20.
M30
%
```

(a) G-code for initial trajectory.

(b) Rhombus reference trajectory.

FIGURE 5.10: Reference trajectory in experiment. (a) G-code for initial trajectory. (b) Rhombus reference trajectory.

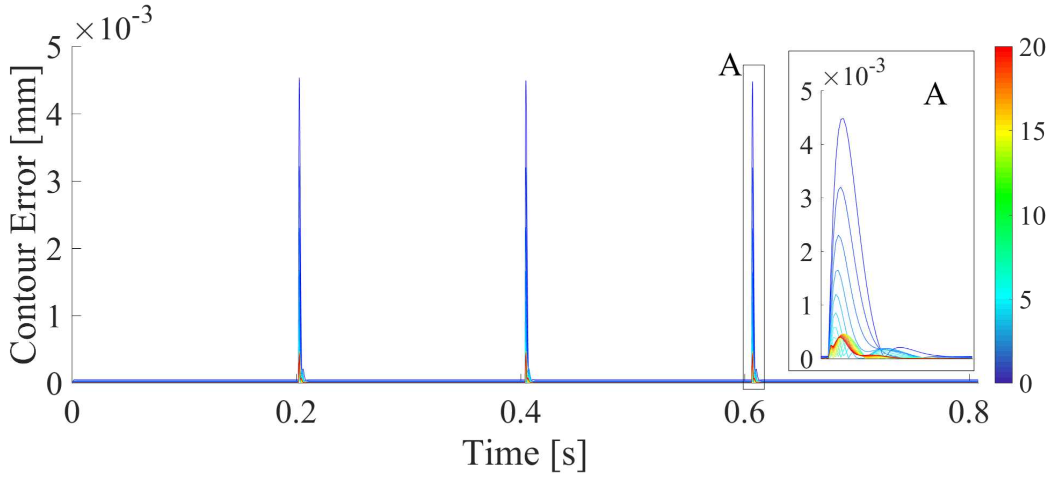


FIGURE 5.11: Simulation result of contour error with linear interpolation.

Circular Trajectory

The circular motion shown in Fig. 5.12, is performed in simulation. The simulation was conducted 20 iterations for ILCC with ACEC, and their performance was compared under both linear and circular interpolations. Simulation results are shows in Figs. 5.13, 5.14, and Table 5.1. Figs. 5.13 (a) and (b) show the contour error profiles for linear and circular interpolations, respectively. Fig. 5.14 shows the convergence profiles of the maximum and mean contour error. In all figures, the first iteration refers to the original controller which produces the largest contour error of 0.903 μm

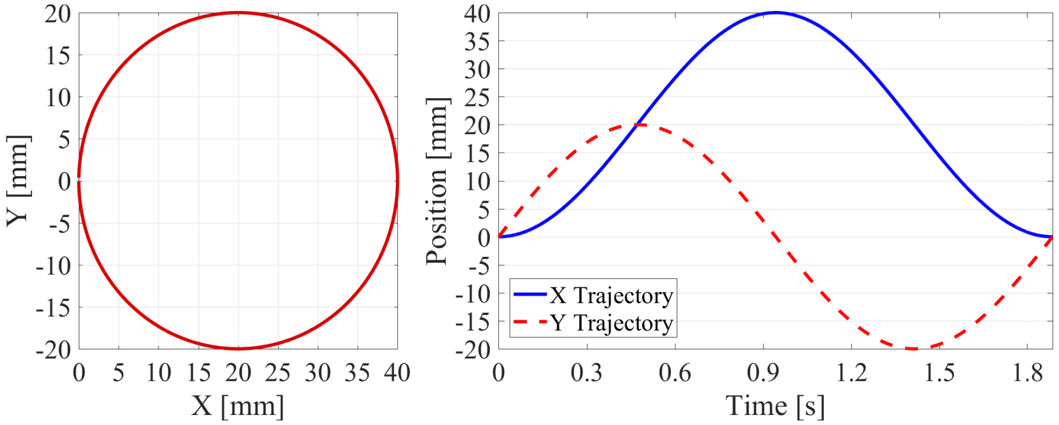
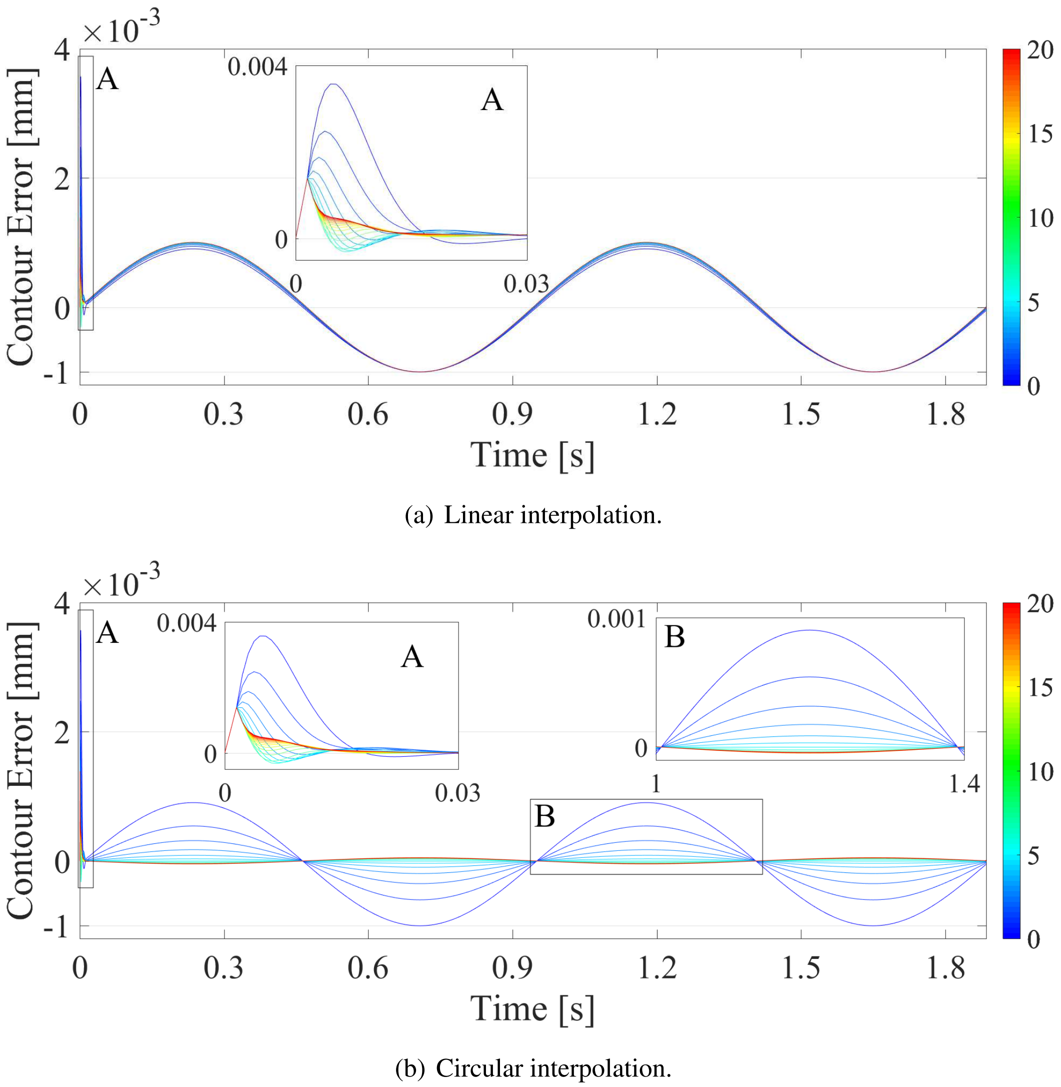


FIGURE 5.12: Circle reference trajectory.

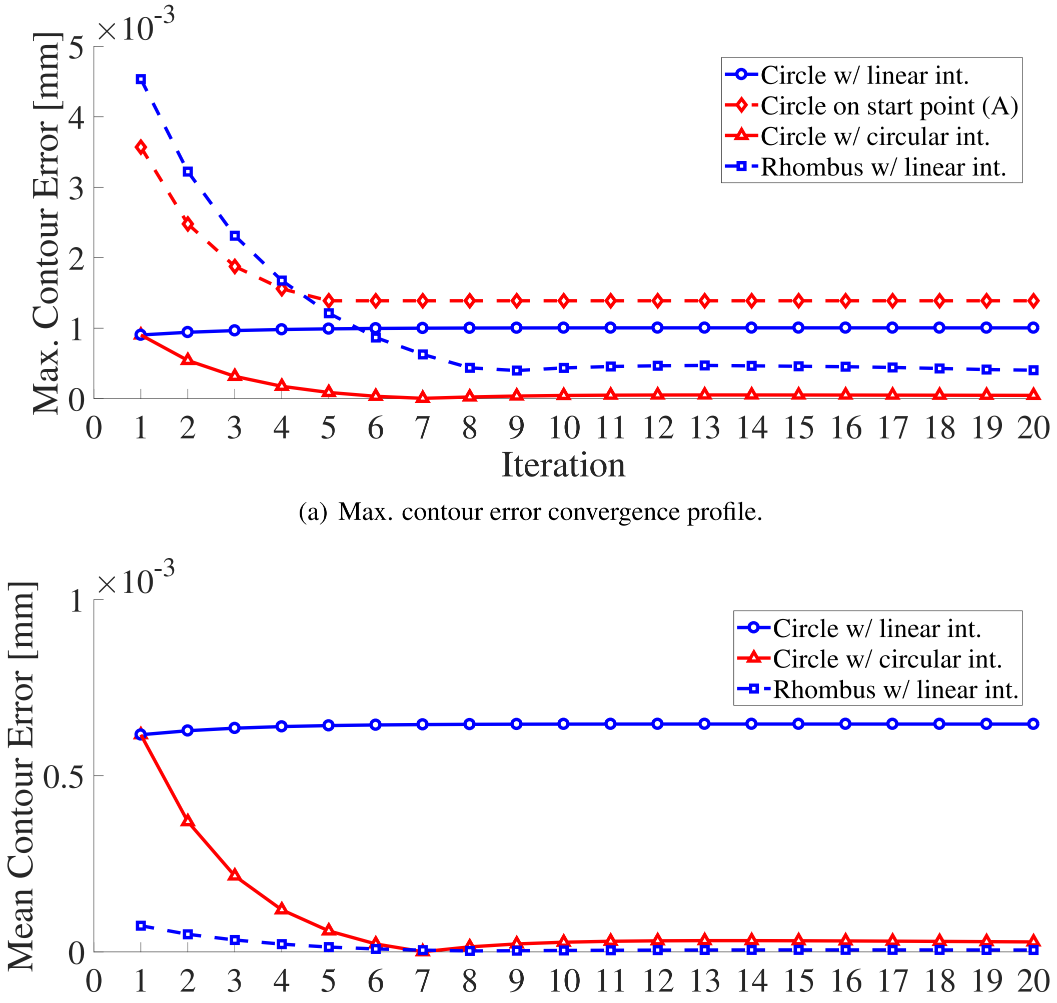
with a mean contour error of 0.616 μm. By applying linear interpolation, the maximum contour error was increased to 1 μm. It shows that the linear interpolation is not effective for the circular motion trajectory. In contrast, the proposed circular interpolation improves the maximum and mean contour errors to 0.004 μm and 0.002 μm, respectively. This is equivalent to a contour error reduction of 99.56 % and 99.08 %, respectively, and there is no significant error reduction after the 8^{th} iteration. All of the simulation results are summarized in Table 5.1.



(a) Linear interpolation.

(b) Circular interpolation.

FIGURE 5.13: Simulation results of contour error for circular trajectory. (a) Linear interpolation. (b) Circular interpolation.



(a) Max. contour error convergence profile.

(b) Mean. contour error convergence profile.

FIGURE 5.14: Contour error convergence profiles in simulation. (a) Max. contour error convergence profile. (b) Mean. contour error convergence profile.

Simulation with Cutting Force

The cutting force is estimated in simulation to verify the effectiveness of the proposed method under a machining condition. An aluminum alloy material and an end-mill cutter diameter with $D = 12$ mm, two flutes, and a helixal angle of 45^o are chosen to simulate the cutting process. The cutting scenario is simulated using the following conditions: depth of cut $d_{oc} = 0.1$ mm, width of cut / side step $w_{oc} = 0.2$ mm, feed rate $v_c = 60$ mm/minute, spindle rotation speed 1600 rpm, and feed $s_t = 0.07$

mm/teeth. The estimated cutting force was calculated as follows [109]:

$$dF_{t,j}\left(\phi_j,z\right) = K_{te}dS\left(z\right) + K_{tc}s_t\sin\phi_j dz, \tag{5.38}$$

$$dF_{r,j}\left(\phi_j,z\right) = K_{re}dS\left(z\right) + K_{rc}s_t\sin\phi_j dz, \tag{5.39}$$

$$dF_{a,j}\left(\phi_j,z\right) = K_{ae}dS\left(z\right) + K_{ac}s_t\sin\phi_j dz, \tag{5.40}$$

where dF_t, dF_r, dF_a, j, ϕ, and, dS are the differential tangential, radial, and axial cutting force, the flute number, the immersion angle, and the edge length, respectively.

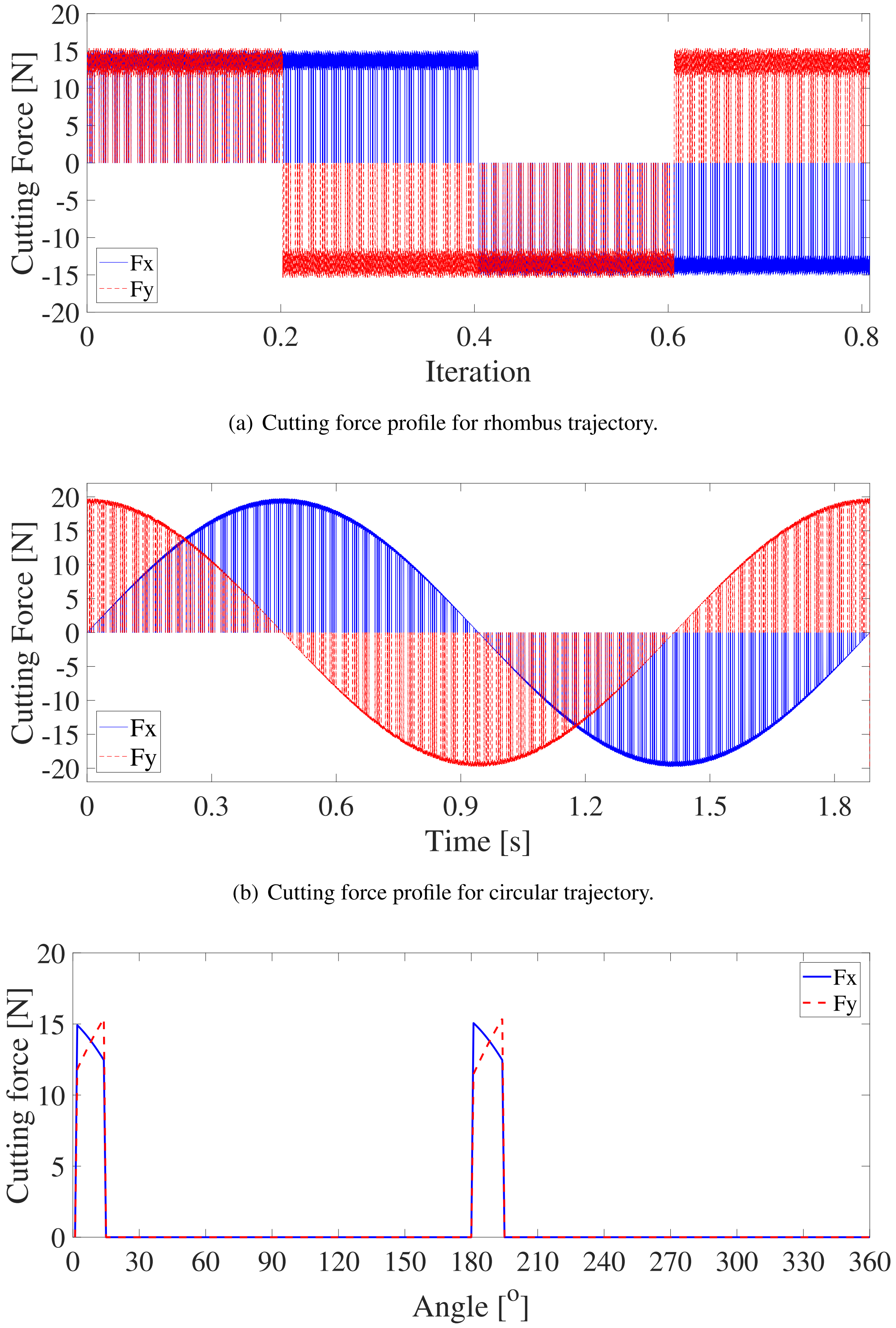
The specific cutting force coefficients for tangential, radial, and axial direction are $[K_{tc} \quad K_{rc} \quad K_{ac}] = [1843.9 \quad 513 \quad 1118.6]$ N/mm^2.

The specific edge force coefficients for tangential, radial, and axial direction are $[K_{te} \quad K_{re} \quad K_{ae}] = [24.0 \quad 43.0 \quad -3.0]$ N/mm^2. Both of X-Y axial cutting forces are calculated by the transformation below:

$$\begin{bmatrix} dF_{x,j}\left(\phi_j\right) \\ dF_{y,j}\left(\phi_j\right) \end{bmatrix} = \begin{bmatrix} -\cos\phi_j & -\sin\kappa\sin\phi_j \\ \sin\phi_j & -\sin\kappa\cos\phi_j \end{bmatrix} \times \begin{bmatrix} dF_{t,j}\left(\phi_j\right) \\ dF_{r,j}\left(\phi_j\right) \end{bmatrix} \tag{5.41}$$

where κ is the radial lag angle. Both of the X-Y estimated cutting force for both rhombus and circular trajectories are shown in Fig. 5.15. Detail of the cutting force estimation procedure is described in [109].

Simulation of ILCC was conducted in 20 iterations with and without cutting force under rhombus and circular motion trajectories. Simulation results are shown in Figs. 5.16, 5.17 and Table 5.2. Fig. 5.16 (a) shows the contour error profile under cutting force in the rhombus trajectory. The peak of contour error appears in each sharp corner where the initial maximum and average contour error are 22 μm and



(a) Cutting force profile for rhombus trajectory.

(b) Cutting force profile for circular trajectory.

(c) Cutting force profile in one rotation.

FIGURE 5.15: Simulation results of cutting force.

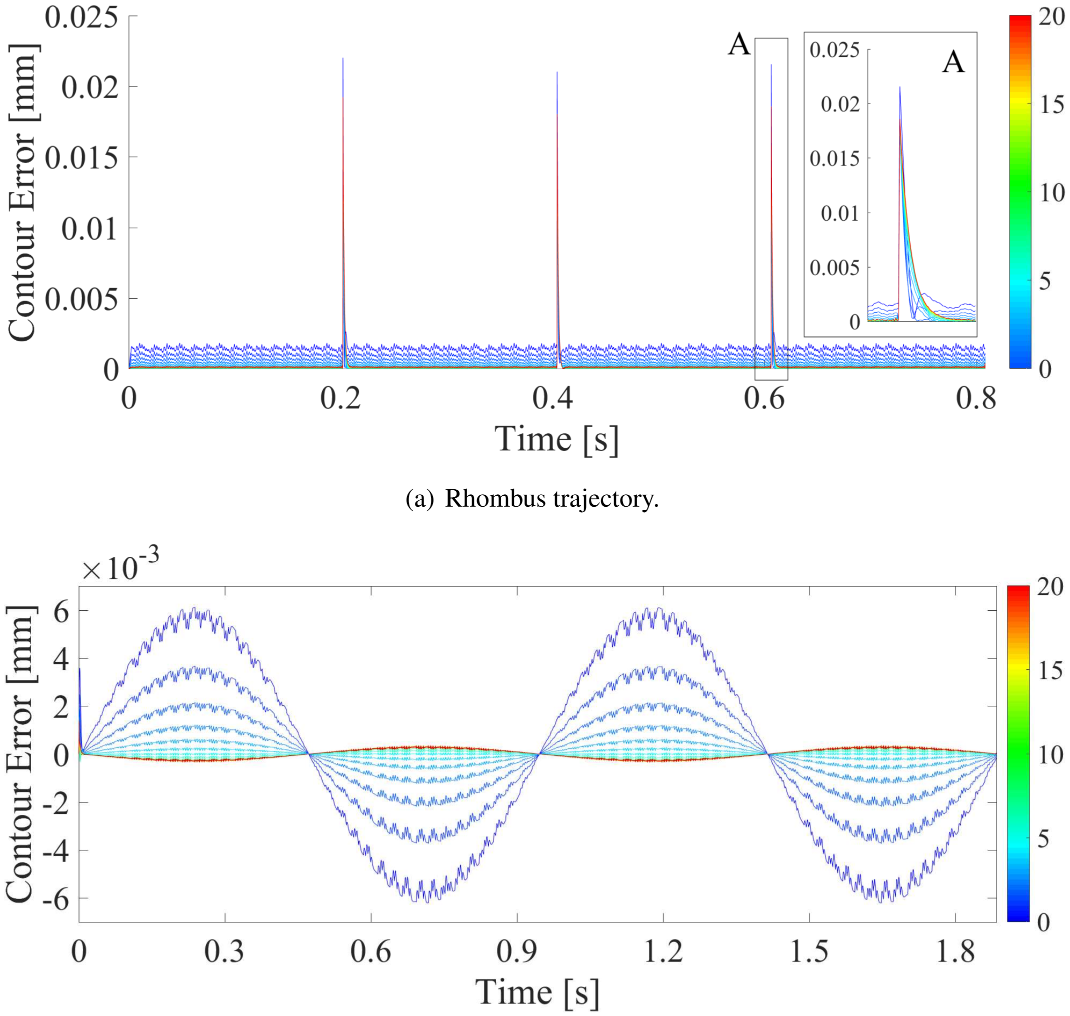
TABLE 5.1: Simulation results of contour error.

Trajectory	Interpolation	Max. Contour Error [μm]			Mean Contour Error [μm]		
		Initial	Final	Reduction	Initial	Final	Reduction
Rhombus	Linear	4.535	0.398	91.22 %	0.074	0.003	95.95 %
Circle	Linear	0.903	1	-10.74 %	0.616	0.647	-5.03 %
	Circular	0.903	0.004	99.56 %	0.616	0.002	99.68 %

TABLE 5.2: Simulation results of contour error under cutting force.

Trajectory	Cutting Force	Max. Contour Error [μm]			Mean Contour Error [μm]		
		Initial	Final	Reduction	Initial	Final	Reduction
Rhombus	without	4.535	0.398	91.22 %	0.074	0.003	95.95 %
	with	22	18	18.18 %	1.469	0.099	93.26 %
Circle	without	0.903	0.004	99.56 %	0.616	0.002	99.68 %
	with	6.11	0.35	94.27 %	3.737	0.015	99.60 %

1.469 μm, respectively. Fig. 5.16 (b) shows the contour error profile under cutting force for a circular trajectory with the initial maximum and average contour error are 0.903 μm and 0.616 μm, respectively. By applying the ILCC, the contour error is reduced in both cases as shown in Fig. 5.17. However, the proposed method is still effective for reducing contour error under the machining condition. By applying the proposed method in the rhombus trajectory case, the ILCC with ACEC reduces the maximum and mean contour errors to 18 μm and 0.1 μm which are equivalent to 18.2 % and 93.3 %, respectively. In the circular trajectory, the proposed method improves the maximum and mean contour errors effectively to 0.35 μm and 0.015 μm, respectively. For clarity, all of the simulation results are summarized in Table 5.1.



(a) Rhombus trajectory.

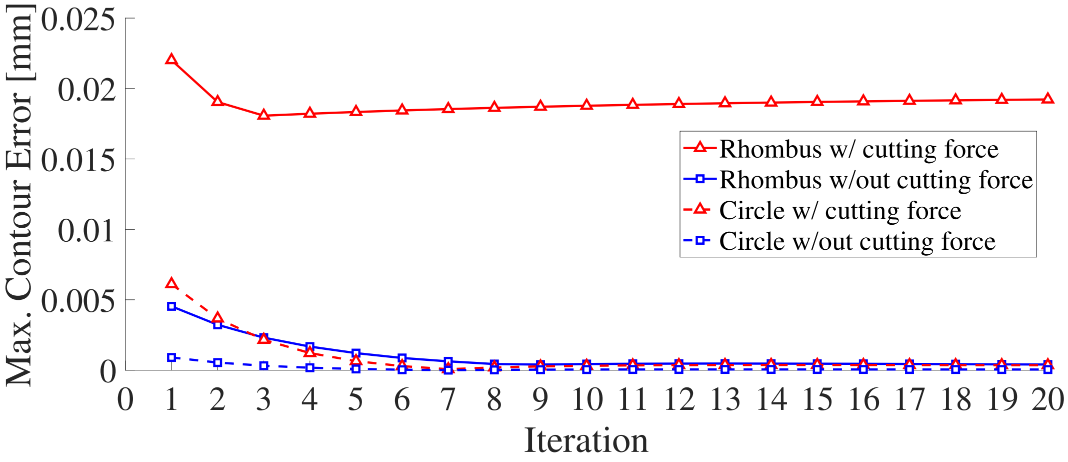
(b) Circle trajectory.

FIGURE 5.16: Simulation results of contour error under cutting force. (a) Rhombus trajectory. (b) Circle trajectory.

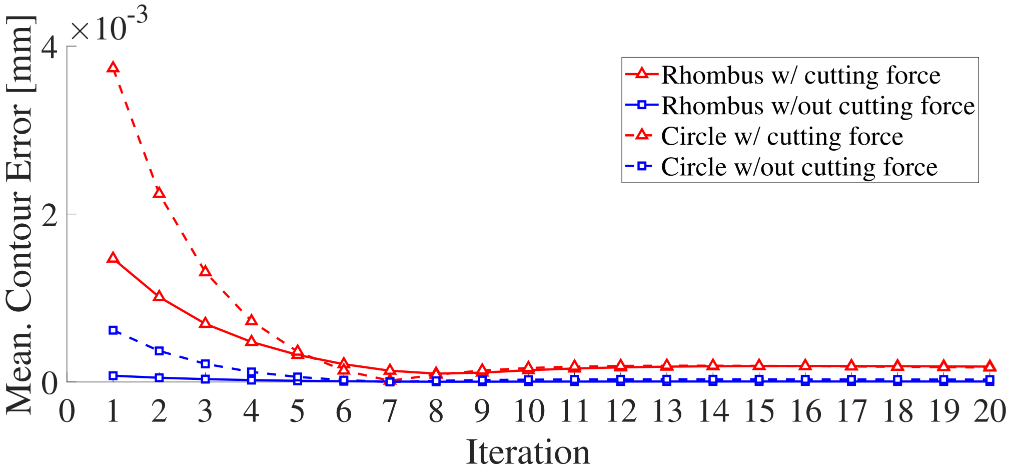
5.7 Experiment

5.7.1 Experimental Condition

As described in the previous sections, a CNC machine tool DMG MORI NV1500 (5.5) with linear motors controlled by a Mitsubishi M700V was used for experiment to verify the effectiveness of the proposed approach. The actual position of the feed drive system was obtained through Melsoft NC Analyzer 2 software with an RJ45 connection. The ILCC was programmed in MATLAB® in a separate personal computer (Windows OS) and embedded to the machine by direct numerical



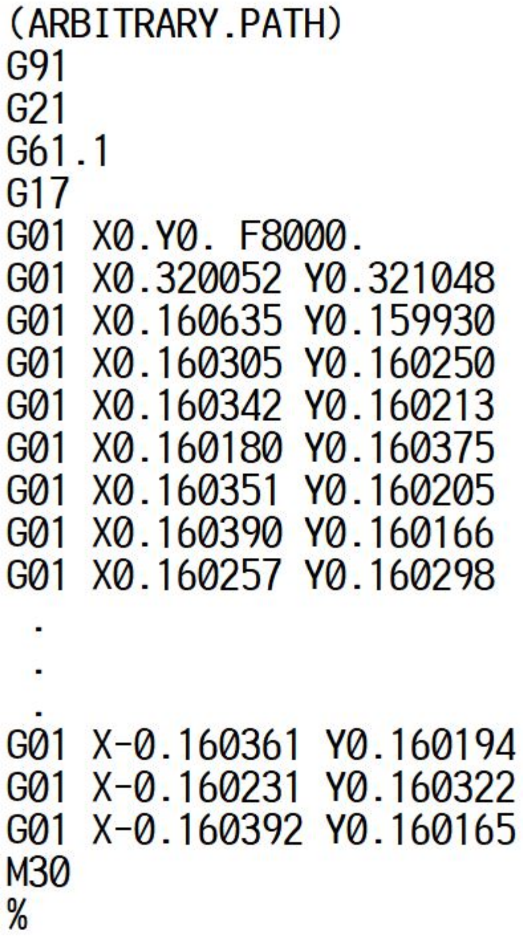
(a) Max. contour error convergence profile.



(b) Mean. contour error convergence profile.

FIGURE 5.17: Simulation results of error convergence profiles under cutting force. (a) Max. contour error convergence profile. (b) Mean. contour error convergence profile.

control (DNC) with the NC Analyzer 2 software. The sampling rate was 1.7 ms and the considered operational frequency $0 \leq \omega \leq 100$ rad/s with a discretization of 1 rad/s due to each axial bode diagram (Figs. 5.8 and 5.9). The experimental interface structure is shown in Figs. 5.5 and 2.11. The optimal learning control gains $[K_{Pl} \quad K_{Il} \quad K_{Dl}] = [0.079 \quad 0.05\ 1/\mathrm{s} \quad 0.452\ \mathrm{s}]$ were obtained, which led to a convergence factor ε_i to be 0.752. According to simulation results, the circular motion trajectory provides a better contouring performance than the rhombus one because of the difficulty of tracking a sharp corner trajectory. If the proposed method is proven in rhombus trajectory by experiment, it will also work for a circular trajectory. The



```
(ARBITRARY.PATH)
G91
G21
G61.1
G17
G01 X0.Y0. F8000.
G01 X0.320052 Y0.321048
G01 X0.160635 Y0.159930
G01 X0.160305 Y0.160250
G01 X0.160342 Y0.160213
G01 X0.160180 Y0.160375
G01 X0.160351 Y0.160205
G01 X0.160390 Y0.160166
G01 X0.160257 Y0.160298
  .
  .
  .
G01 X-0.160361 Y0.160194
G01 X-0.160231 Y0.160322
G01 X-0.160392 Y0.160165
M30
%
```

FIGURE 5.18: Finally generated modified G-code program.

initial reference trajectory was defined in a G-code program to create a typical rhombus sharp corner trajectory with velocity of 8000 mm/min as shown in Figs. 5.10 (a) and (b), respectively.

First, the initial reference trajectory is executed by the CNC machine which is connected to a PC. Then, the actual position data measured by linear scales with 1 *n*m accuracy are obtained by NC Analyzer 2 software [53]. Third, the ILCC is implemented to modify the NC program based on the resultant contour error after executing the NC program by CNC machine. All of the above procedures are conducted iteratively until no significant contour error reductions are seen. Based on simulation, 20 iterations are decided to show the convergence of contour error reduction. To guarantee the repetitiveness of the proposed approach, experiments are repeated 5 times under the same condition.

5.7.2 Performance Evaluation

To evaluate the performance, the experimental results are analyzed by calculating the mean, maximum, and root mean square (RMS) of contour errors which are represented as follows:

$$\overline{e_c} = \frac{\sum_{M=1}^{N} e_{cM}}{N} \tag{5.42}$$

$$e_{c\max} = \max_{0<M\leq N} |e_{cM}| \tag{5.43}$$

$$e_{c\mathrm{rms}} = \sqrt{\frac{\sum_{M=1}^{N} e_{cM}^2}{N}} \tag{5.44}$$

where $\overline{e_c}$, $e_{c\max}$, $e_{c\mathrm{rms}}$, N, and M are the mean, maximum, and RMS of contour error, total discrete points, and number point, respectively. The maximum contour error $e_{c\max}$ usually appears in the critical section such as a sharp corner section.

5.7.3 Experiment Results

Five times experiments with 20 iterations with a rhombus reference trajectory shown in Fig. 5.10 have been conducted, and the minimum contour error was obtained after

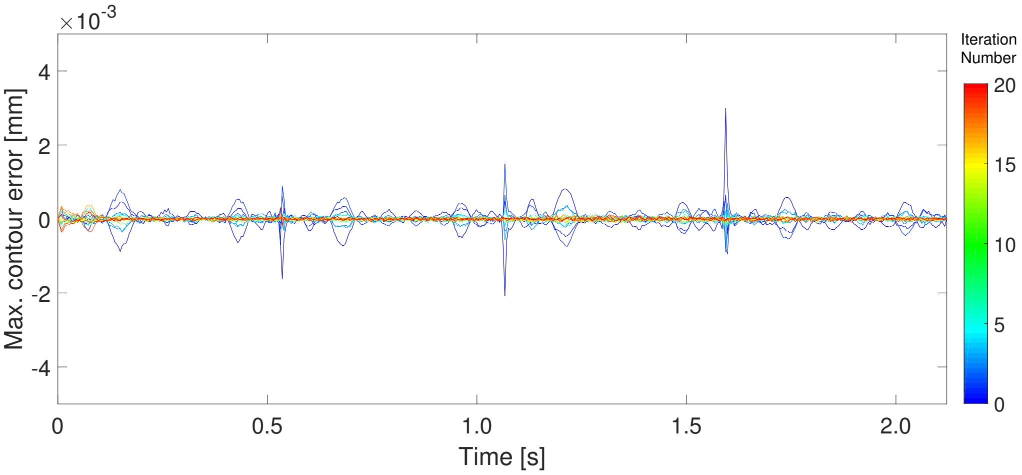


FIGURE 5.19: Contour error: *x* axial component.

‘

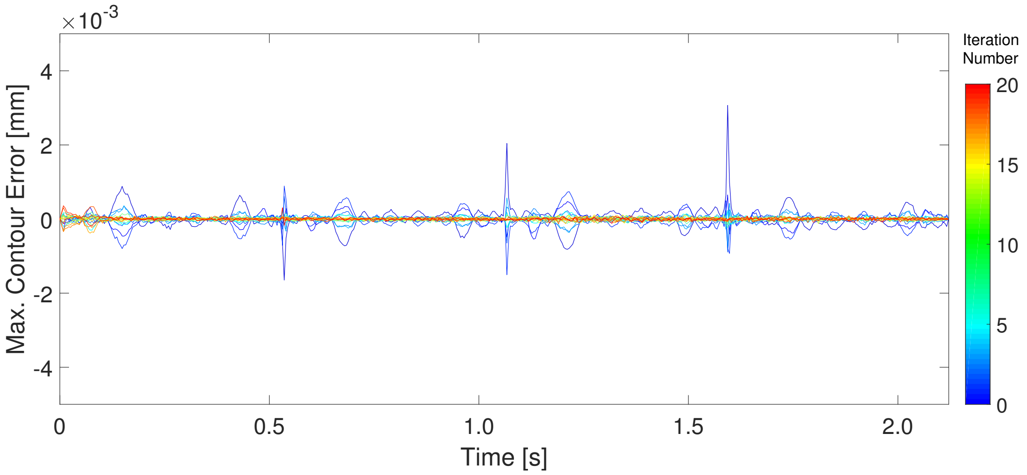


FIGURE 5.20: Contour error: y axial component.

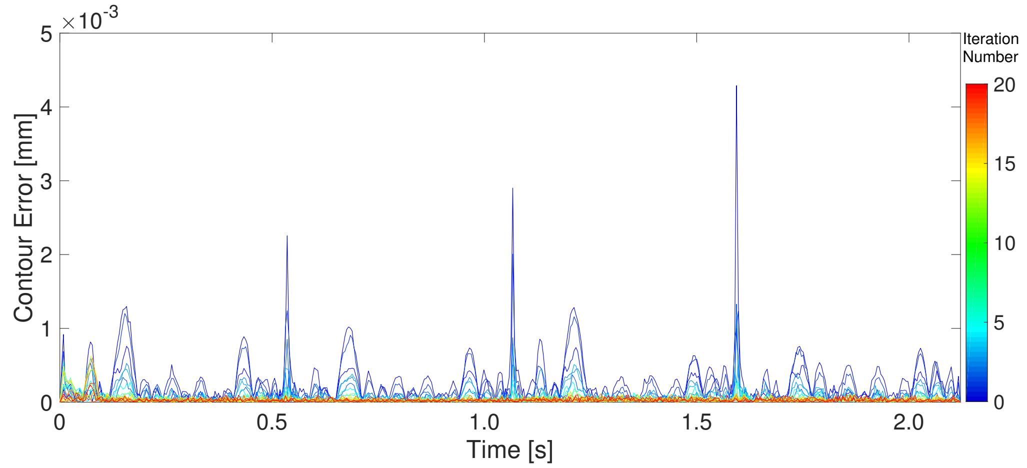


FIGURE 5.21: Contour error magnitude.

12 iterations or after 25.8 second as shown in Figs. 5.19 - 5.24, and Table 5.3. Fig. 5.18 shows a part of the finally generated G-code program. Figs. 5.19 and 5.20 show the resulted contour errors in x and y components. Fig. 5.21 shows the contour error profiles based on ILCC with ACEC. The first iteration was conducted only by the originally equipped controller with initial contour error 4.25 μm on average. Under the proposed method, the contour error was reduced to 0.23 μm on average of each iteration in the 12th iteration and no further significant error reduction was observed thereafter. This is equivalent to the error reduction of 94.58 % from the initial maximum contour error. Fig. 5.22 shows real trajectories based on ILCC with ACEC. Fig. 5.24 shows the contour error convergences properties for the proposed method.

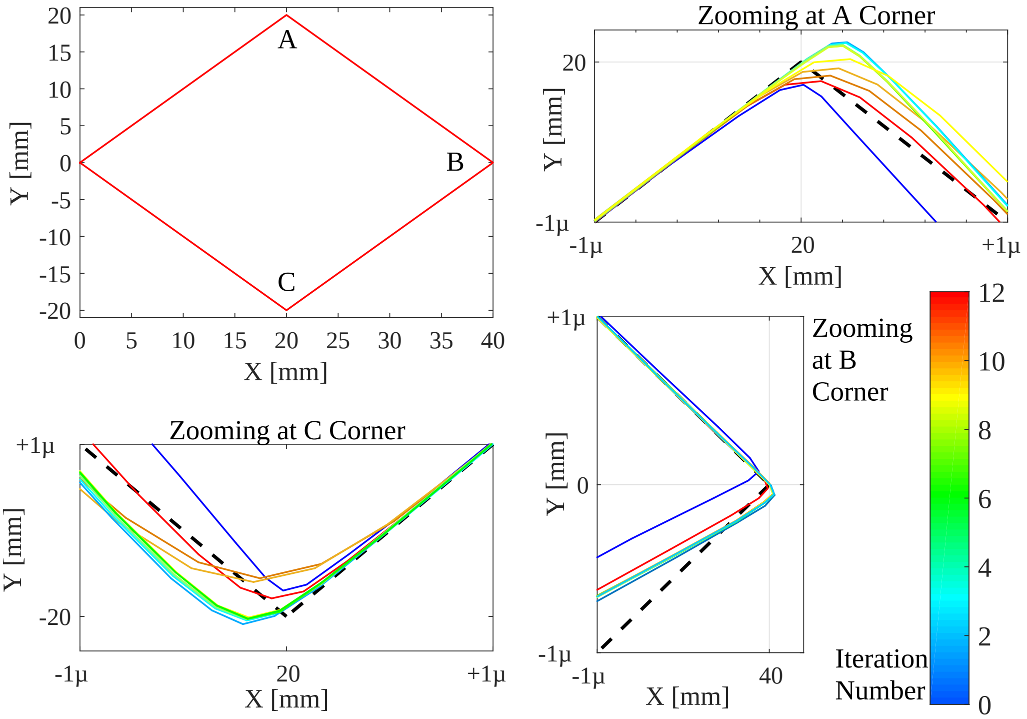


FIGURE 5.22: Desired and obtained trajectory at corners.

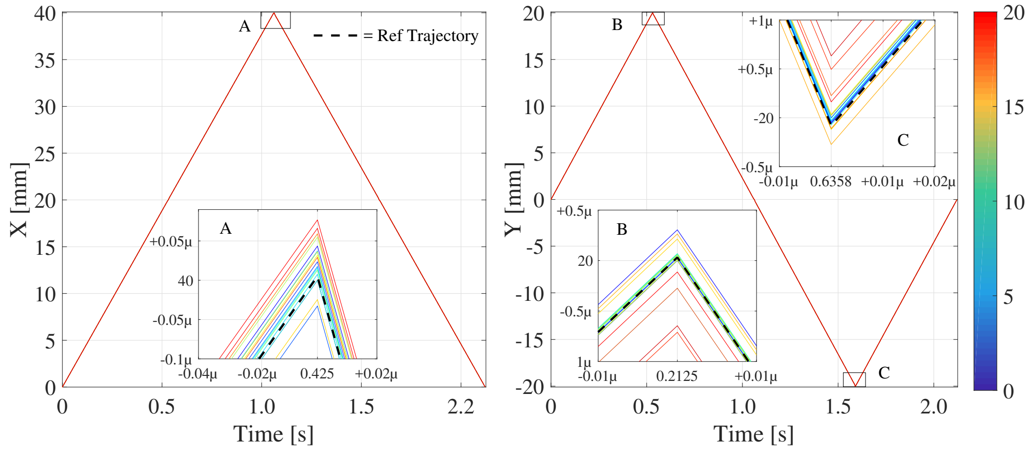


FIGURE 5.23: Experiment result of time-series trajectory.

The experimental convergence ratio ($\varepsilon_i = 0.829$) is greater than the theoretical factor ($\varepsilon_i = 0.752$) because of uncertain disturbance in the experimental machine. For clarity, all the experimental results are summarized in Fig. 5.24 and Tables 5.3 - 5.4. It can be seen that an almost monotonically contour error reduction tendency is observed in all experimental results. This concludes that the proposed controller is practically effective for performance improvement in CNC machine tool feed drive systems.

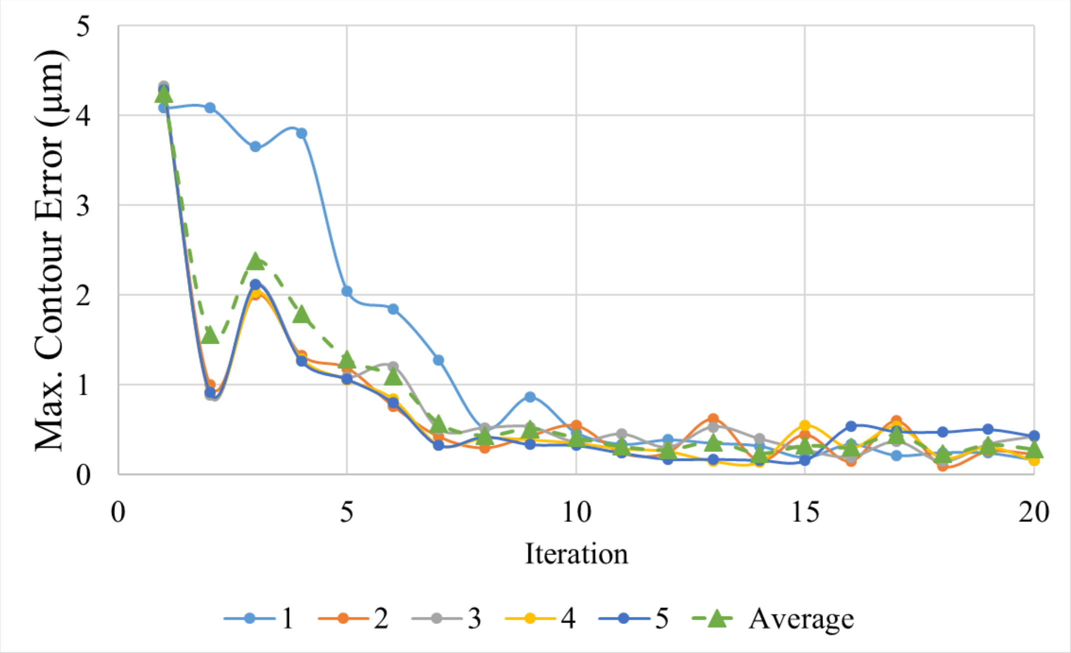


FIGURE 5.24: Experimental convergence profiles.

5.7.4 Energy Evaluation

The same experiments with Fig. 5.22 were repeated 5 times to evaluate energy consumption. The energy evaluation results are shown as Figs. 5.25-5.26. The consumed energy was evaluated by measuring electric current through NC Analyzer 2 software

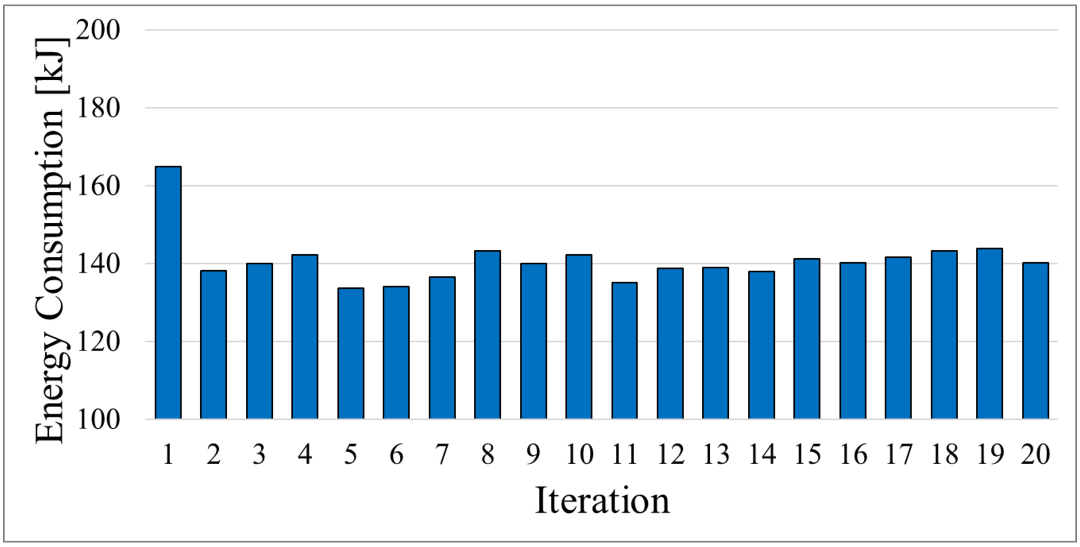


FIGURE 5.25: Experimental energy results.

TABLE 5.3: Contour error results in experiment [μm].

Iteration (j)	Experiment Number					Average	$\sum_{i=0}^{j}\bar{t}_i$ [s]
	1	2	3	4	5		
1	4.087	4.290	4.334	4.254	4.284	4.250	1.6
2	4.087	1.002	0.883	0.922	0.918	1.562	3.8
3	3.653	2.007	2.121	2.035	2.115	2.386	6.0
4	3.800	1.328	1.282	1.289	1.264	1.793	8.2
5	2.046	1.186	1.083	1.058	1.064	1.288	10.4
6	1.846	0.765	1.203	0.845	0.803	1.093	12.6
7	1.274	0.424	0.507	0.333	0.328	0.573	14.8
8	0.516	0.299	0.521	0.409	0.424	0.434	17.0
9	0.861	0.434	0.531	0.389	0.339	0.511	19.2
10	0.467	0.546	0.362	0.352	0.328	0.411	21.4
11	0.338	0.241	0.451	0.289	0.244	0.313	23.6
12	0.386	0.247	0.303	0.261	0.175	0.275	25.8
13	0.347	0.619	0.532	0.150	0.175	0.365	28.0
14	0.319	0.160	0.399	0.138	0.164	0.236	30.2
15	0.191	0.444	0.274	0.547	0.158	0.323	32.4
16	0.336	0.152	0.195	0.300	0.538	0.304	34.6
17	0.213	0.599	0.367	0.539	0.483	0.440	36.8
18	0.245	0.096	0.146	0.187	0.477	0.230	39.0
19	0.239	0.262	0.337	0.310	0.508	0.331	41.2
20	0.164	0.229	0.423	0.157	0.432	0.281	43.4
Max	4.087	4.290	4.334	4.254	4.284	4.250	
Min	0.164	0.096	0.146	0.138	0.158	0.230	
Reduction	95.98%	97.76%	96.63%	96.75%	96.31%	94.58%	

[53]. Energy consumption is calculated by the following equations.

$$\begin{aligned} E &= \int_0^N P(t)\,dt = \int_0^N \sqrt{3}\lambda V I dt = \sqrt{3}\lambda R \int_0^N I^2 dt \\ &\simeq \sqrt{3}\lambda R t_s \sum_{n=1}^{N} I_n^2 \end{aligned} \qquad (5.45)$$

where λ, R, V, t_s, and I_n are the power factor, resistance, electric voltage, sampling time, and current at the n^{th} sampling instant. N is the total number of sampling instants $(n = 1, 2, ..., N)$. The power factor λ is set to 0.7 which is obtained by observing the phase difference between $V(t)$ and $I(t)$. This observation was done in the previous study [110] which used the same experimental system. The resistance R and sampling time t_s are equal 4.05 Ω and 1.7 ms, respectively. The control input

TABLE 5.4: Summary of experimental results.

Contour Error	Experiment Number					Average
	1	2	3	4	5	
Max. Contour Error [μm]						
Initial	4.087	4.29	4.334	4.254	4.284	4.25
Final	0.164	0.096	0.146	0.138	0.158	0.23
Reduction	95.99%	97.76%	96.63%	96.76%	96.31%	94.59%
Mean Contour Error [μm]						
Initial	0.267	0.290	0.274	0.259	0.266	0.271
Final	0.034	0.025	0.030	0.031	0.033	0.031
Reduction	87.31%	91.20%	89.01%	88.14%	87.46%	88.67%
RMS Contour Error [μm]						
Initial	0.428	0.454	0.442	0.421	0.433	0.436
Final	0.044	0.032	0.038	0.04	0.044	0.040
Reduction	89.72%	92.95%	91.40%	90.50%	89.84%	90.91%

variance was determined using the following equation:

$$\sigma_i = \sqrt{\frac{\sum_{n=1}^{N} (f_{in} - \Theta_i)^2}{N}}, \qquad i = x, y. \tag{5.46}$$

where Θ_i and f_{in} are the mean of all control signals and the control signal value at the n^{th} sampling instant of the i^{th} axis, respectively.

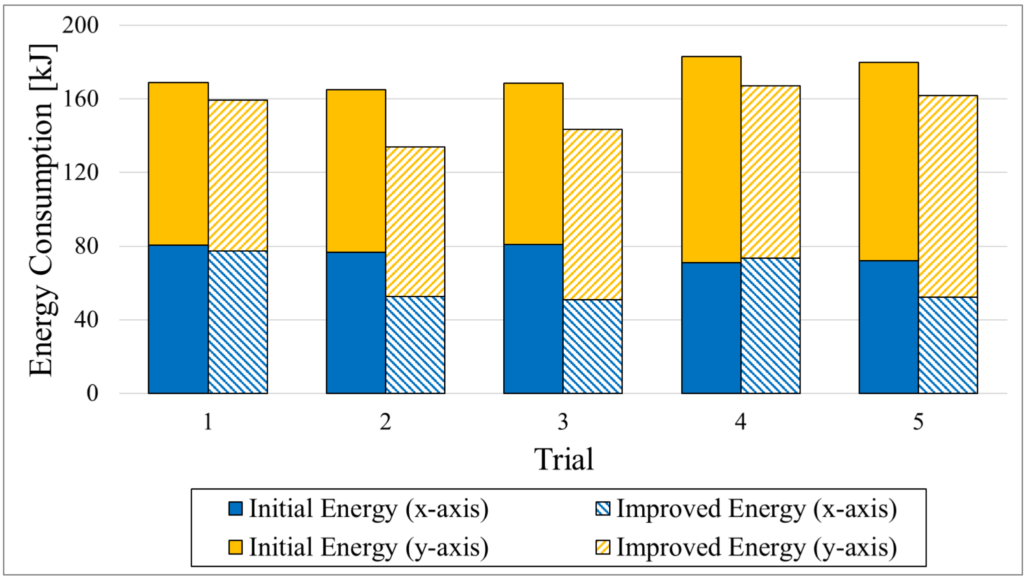
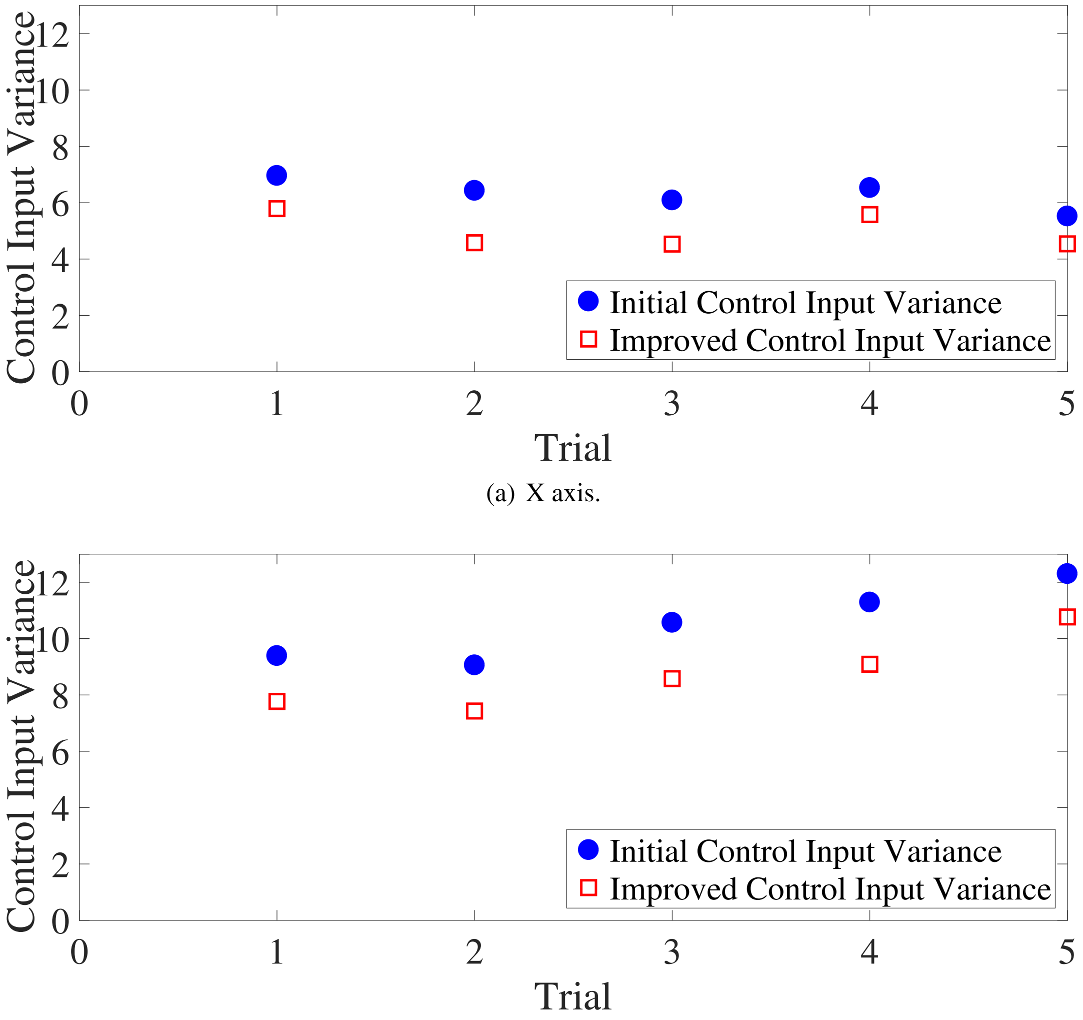


FIGURE 5.26: Consumed energy evaluation results.

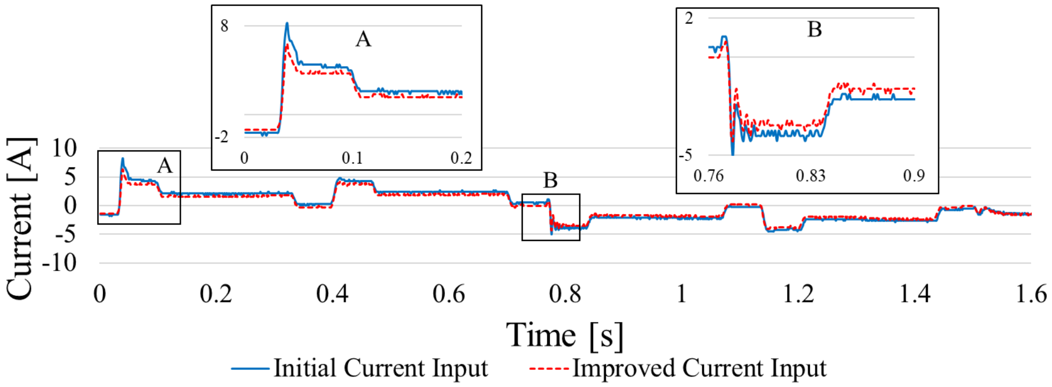


(a) X axis.

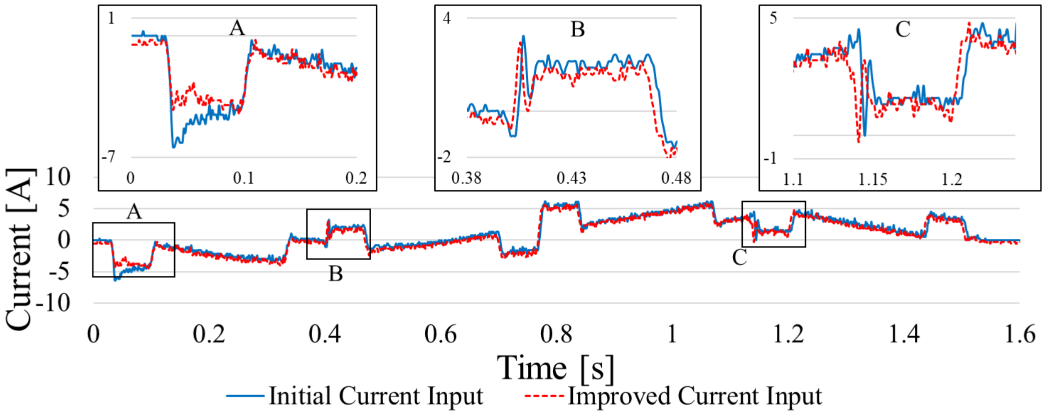
(b) Y axis.

FIGURE 5.27: Control input variance. (a) X axis. (b) Y axis.

Fig. 5.25, Tables 5.5 and 5.6 shows an energy consumption for 20 iteration. The energy reduction appeared in the 2$^{\text{nd}}$ iteration as similar with contour error reduction in Fig. 5.24. Because the desired trajectory is smoothed around the corner in the first iteration, there is no significant reduction after the second iteration in Fig. 5.25. The consumed energy and control input variance of the original controller and the ILCC are compared for each feed drive axis as shown in Figs. 5.26-5.27. Fig. 5.26 show consumed energy evaluation results for x, y, and total magnitude of axis. The initial trajectory was conducted only by the originally equipped controller with initial energy consumption 77.4 kJ and 106.2 kJ for x and y axis on average, respectively.



(a) X axis.



(b) Y axis.

FIGURE 5.28: Control input profiles. (a) X axis. (b) Y axis.

Under the proposed method, they were reduced to 58.2 kJ and 90.3 kJ, respectively. This is equivalent to energy reduction of 24.9 % and 14.7 %, respectively. Fig. 5.26 shows the total energy reduction from 173.2 kJ to 153.2 kJ or 11.7 % energy reduction on average. The proposed method achieved a reduction of control input variance by 20.7 % and 17.2 % for x and y axis, respectively as well as consumed energy reduction because the trajectory is already smoothed in the modified trajectory. Fig. 5.28 shows improved control input current profiles which are slightly reduced around the start and corner positions.

TABLE 5.5: Energy consumption results.

Iteration Number	Trial [kJoule]					Average
	1	2	3	4	5	
1	169.7	164.9	168.7	183.1	179.8	173.2
2	163.5	138.3	145.2	167.2	161.9	155.2
3	165.2	140.1	143.7	169.8	163.5	156.4
4	159.3	142.2	146.3	169.3	166.2	156.7
5	163.7	133.8	145.1	170.1	165.7	155.7
6	163.2	134.1	150.2	171.5	162.3	156.3
7	164.2	136.6	145.1	170.4	165.4	156.3
8	159.9	143.3	146.2	173.0	165.0	157.5
9	160.3	140.1	146.6	173.6	164.9	157.1
10	165.2	142.4	148.5	175.6	164.2	159.2
11	166.2	135.1	146.4	175.1	172.7	159.1
12	165.1	138.8	150.3	177.8	165.0	159.4
13	163.5	139.1	147.1	180.7	166.5	159.4
14	163.7	138.1	148.0	174.6	169.9	158.8
15	161.4	141.3	151.0	176.1	177.7	161.5
16	161.5	140.3	161.2	180.1	172.0	163.0
17	163.6	141.8	161.0	180.2	171.7	163.7
18	163.9	143.4	162.8	182.9	171.0	164.8
19	160.2	144.0	164.3	178.4	177.1	164.8
20	164.7	140.3	156.3	179.3	172.4	162.6
Initial	169.7	164.9	168.7	183.1	179.8	173.2
Min	159.3	133.8	143.7	167.2	161.9	155.2
Reduction	6.2%	18.8%	14.9%	8.7%	9.9%	10.4%

5.8 Discussion

The contour error estimation of ILCC is a key factor to achieve higher contouring performance. ACEC with a linear interpolation is an effective way to track a rhombus sharp corner trajectory, which is typically difficult to be tracked because of the infinite acceleration occurred by trajectory discontinuity. In the corner area, drive axis X or Y reverses suddenly to track the desired motion. It requires an instant deceleration and acceleration which needs higher energy consumption, and such tracking is practically impossible for a typical-PID control which normally produces overshoot [63]. The ACEC with linear interpolation modifies a reference trajectory or NC program, and it modifies sharp corner to be smooth. By this modification, contour

TABLE 5.6: Summary of energy evaluation results.

Energy Consumption	Experiment Number					Average
	1	2	3	4	5	
X-Axis [kJoule]						
Initial	80.5	76.8	80.8	76.7	72.1	77.4
Final	71.9	51.1	50.7	67.1	50.1	58.2
Reduction	10.7%	33.4%	37.2%	12.5%	30.5%	24.9%
Y-Axis [kJoule]						
Initial	92.0	91.7	109.6	111.9	126.0	106.2
Final	81.8	81.1	87.9	93.1	107.7	90.3
Reduction	11.0%	11.5%	19.7%	16.8%	14.5%	14.7%
Total [kJoule]						
Initial	169.7	164.9	168.7	183.1	179.8	173.2
Final	159.3	133.8	143.7	167.2	161.9	153.2
Reduction	6.2%	18.8%	14.9%	8.7%	9.9%	11.7%

error and energy consumption reduce [111]. The ACEC with linear interpolation is not suitable to track typical circular or free-form trajectories. It exhibits lower contouring performance because linear interpolation can not estimate contour error precisely in higher curvature trajectories.

ACEC with circular interpolation is designed to estimate contour error precisely for a circular motion trajectory. It requires three discrete points to interpolate circular curvature and estimates contour error from actual position and circular radius. This method provides higher contouring ability for a circular trajectory. However, it can not be applied to a linear path because of infinite radius properties. Both linear and circular trajectories are required to be combined for considering a general trajectories that include both.

Machining process is simulated to evaluate the proposed method under cutting force. Cutting force is estimated and applied for both rhombus and circular trajectories. Generally, the proposed method reduces the contour errors. However, it is not effective for rhombus trajectory especially in sharp corner areas, because the cutting force direction changes to be opposite suddenly. Simulation shows that rhombus trajectory with a sharp corner exhibits the largest contour error. To prove the proposed method experimentally, the rhombus trajectory is chosen. Once the proposed method reduces contour error for a difficult trajectory such as a rhombus trajectory, it can work also for other trajectories properly. Both simulation and experiment show a similar error reduction trend. This concludes that the proposed method is effective for performance improvement in CNC machine tools.

5.9 Conclusion

The CNC machine tool with linear motors is an advanced technology. The ILCC with ACEC under linear and circular interpolations are implemented with rhombus and circular motion trajectories, and the performance has been demonstrated by simulation and experimental results. The proposed method modifies the NC program considering the previous actual contour error iteratively to enhance tracking performance. Simulation results show that the linear interpolation can reduce the maximum and mean contour errors by 91.22 % and 95.95 %, respectively for a rhombus motion trajectory. However, it was not effective for a circular trajectory as circular interpolation was designed so that the maximum and mean contour errors were reduced by 99.55 % and 99.68 %, respectively. The ILCC with ACEC under a machining condition was also simulated to evaluate the proposed method under such conditions. The linear interpolation with the rhombus trajectory reduces the maximum and mean contour errors by 18.18 % and 93.26 %, respectively. The circular interpolation for a circular motion trajectory provides a better result reducing the maximum and mean errors by to 94.27 % and 99.6 %, respectively.

The rhombus motion trajectory provides the largest initial contour error. For this reason, this trajectory is chosen to conduct the experiment. Experimental results have shown that the maximum and mean contour errors can be reduced by 94.58 %

and 88.67 % on average, respectively. The proposed method improves the contouring performance from normal machining (> 1 μm accuracy) to precision machining (< 1 μm accuracy). In addition, consumed energy is improved under the proposed method. Energy consumption of x, y, and combined axis in machine tools can be reduced by 24.9 %, 14.7 % ,and 11.7 % on average from original energy consumption, respectively. One advantage of the proposed approach is that it can be directly implemented in CNC machine tools currently in use throughout the world, because only an NC program modification is required.

In future work, a hybrid interpolation which combines circular and linear interpolations will be developed to machine a free-form surface. The ILCC with ACEC under linear, circular, and hybrid interpolations will be compared by experiment. The machining process will be conducted under cutting force. Modifying the NC program based on “G01” program is done in this study. To improve the proposed method, modifying the NC program based on “G02” code for a typical circular trajectory will be interesting to be implemented.

Chapter 6

Conclusions

6.1 Summary

In industrial applications, many processes are required to manufacture precision products. High-precision machining is one of the methods utilized for metal processing, especially for applications that require a high-precision product. To construct a high-precision machine, hardware and control improvements are required, especially when it comes to the machine tool structure, motion equipment, and measurement devices. However, the hardware machine condition can change over the life of the equipment due to wear, long working hours, and high production loads. In order to adapt to these changes, a CNC machine needs a control technique that has the ability to learn about disturbances or changes to the actual hardware condition during production. Many control techniques have been developed, but most of them require open access to a commercial CNC machine control. For this reason, ILCC with a cascade design has been proposed in this study to improve the performance of existing machine tools. The error compensation is also expected to increase the ILCC performance. In addition, the proposed method has been implemented in a biaxial feed drive machine tool, a three-axis CNC machine tool with a ball screw mechanism, and a CNC machine tool with a linear motor mechanism, the latter as an advanced technology in machine tool applications. This dissertation consists of three parts.

In the first part, an ILCC considering estimated contour error was proposed for biaxial feed drive systems, and the performance has been demonstrated by simulation and experimental results. The proposed method can modify the reference trajectory of each drive axis iteratively to enhance the tracking performance. The proposed controller can be directly applied to commercial machines currently in use without any modification of their original controllers. The estimated contour error is calculated by multiplying the tracking error by a rotational matrix. The experimental results show that the maximum contour error can be reduced by 63.1 % when the ILCC is added to the conventional feedback controllers or by 49.2 % compared with the CILCC. However, it requires additional iterations to reach the minimum contour error for a higher curvature trajectory because of the assumption of a linear connection between two discrete points on the trajectory, as shown in Fig. 3.2. Furthermore, this method generates non-uniform trajectory points along the reference path, as explained in Chapter 4 and Fig. 4.4, which may produce erratic velocity profiles. The contour error estimation can be improved for high-curvature trajectories by using a smoothing algorithm to obtain smooth velocity profiles.

The second part considered the actual contour error compensation (ACEC) in order to estimate the contour error more accurately. This study proposed a method to enhance the tracking performance of machine tool feed drive systems using ILCC with ACEC and BRT. Its performance was verified experimentally through 3D sharp-corner and half-circular trajectories in a commercial three-axis CNC machine. In comparison with the traditional FBC, the proposed method's performance exhibits average reductions in the maximum and mean contour errors by up to 87.54 % and 76.69 %, respectively. The experimental results of the proposed controller were also compared with the ILCC with estimated contour error and with ILCC with ACEC, demonstrating reductions in the maximum and mean contour errors by up to 45.11 % and 54.48 %, respectively. Furthermore, average reductions of 20.54 % and 26.92 % were demonstrated compared with ILCC with approximated contour error and ILCC with ACEC, respectively. However, the ACEC with linear interpolation is not effective for circular trajectories and the BRT smoothing algorithm does not significantly improve the performance of the ILCC with ACEC system. Therefore, an ILCC with ACEC that includes circular interpolation and not BRT will be designed and implemented in advanced machine tool technologies such as a CNC machine tool with a linear motor mechanism.

The third part of this research showed the implementation of the proposed method to CNC machine tool feed drive systems with a linear motor, and the performance has been demonstrated by simulation and experimental results. The proposed method modifies the reference trajectory by considering the previous actual contour error iteratively to enhance the tracking performance. The experimental results show that the maximum and mean contour error can be reduced by 94.58 % and 88.67 % on average, respectively, when the ILCC with ACEC is added to the existing machine. The implementation of the proposed method to the CNC machine tool with a linear motor mechanism improves the CNC machine tool performance from normal machining ($> 1\ \mu$m) to precision machining ($< 1\ \mu$m). In addition, energy consumption is improved under the proposed method. The average energy consumption of the x, y, and combined axes in machine tools can be reduced by 24.9 %, 14.7 %, and 11.7 %, respectively.

6.2 Future Work

- A hybrid interpolation that combines circular and linear interpolations will be developed to machine a free-form surface. The ILCC with ACEC systems under linear, circular, and hybrid interpolations will be compared experimentally.
- The machining process is a nonlinear phenomenon. The current ILC research assumes that the machine plant is a linear model. A nonlinear model can be

used to improve the ILCC performance. The machining process will be conducted under a cutting force.

- In this study, the NC program was based on a “G01” program. To improve the proposed method, an interesting approach would be to modify the NC program based on “G02” code for a typical circular trajectory.
- The automation of industrial applications is still a challenging topic, especially in regard to the machining process. The ILCC makes it possible to develop closed-loop machining (CLM). The iterative process is conducted for machining raw material to the target product shape. The CLM algorithm has several steps. First, the reference trajectory is defined on the basis of the product shape. Then, an initial trajectory is defined on the basis of scaling up the reference trajectory to the outer side of the raw material; the rest of the material is calculated as actual contour error estimation. Next, the contour error compensation is defined by considering the standard MRR of machine tools; the CLM then modifies an initial trajectory and executes a modified trajectory in the CNC machine tool. Finally, an actual contour error is calculated as the difference between the current actual position and the final reference trajectory. This entire process is repeated iteratively until the actual contour error is within the tolerance range. In this future work, we will combine the CAD/CAM and CNC processes into one smart CLM system. Normally, the CAD/CAM process is external to the CNC machine.

Bibliography

[1] T. Breteau, T. Damay, E. Duc, and J.-Y. Hascoët, “Design for manufacturing with tool paths adapted to marine propeller”, *International Journal on Interactive Design and Manufacturing*, vol. 5, no. 4, pp. 271–275, 2011.

[2] G. Hyatt, M. Piber, N. Chaphalkar, O. Kleinhenz, and M. Mori, “A review of new strategies for gear production”, *Procedia CIRP*, vol. 14, pp. 72–76, 2014.

[3] O. Castiblanco and I. Shareef, “Optimization of steplock® orthotic knee joint design”, *Procedia Manufacturing*, vol. 10, pp. 622–633, 2017.

[4] N. L. de Lacalle and A. L. Mentxaka, “Machine tools for removal processes: A general view”, in *Machine tools for high performance machining*, Springer Science & Business Media, 2008, pp. 1–45.

[5] A. H. Slocum, “Precision machine design: Macromachine design philosophy and its applicability to the design of micromachines”, in *Proceedings of IEEE Micro Electro Mechanical Systems*, IEEE, 1992, pp. 37–42.

[6] M. Weck, *Handbook of Machine tools: Metrological analysis and performance tests*. Wiley, 1984, vol. 4.

[7] G. Byrne, D. Dornfeld, and B. Denkena, “Advancing cutting technology”, *CIRP Annals-Manufacturing Technology*, vol. 52, no. 2, pp. 483–507, 2003.

[8] A. Lamikiz, L. N. L. de Lacalle, and A. Celaya, “Machine tool performance and precision”, in *Machine tools for high performance machining*, Springer Science & Business Media, 2009, pp. 219–260.

[9] N. Uchiyama, T. Nakamura, and H. Yanagiuchi, “The effectiveness of contouring control and a design for three-dimensional machining”, *International*

Journal of Machine Tools and Manufacture, vol. 49, no. 11, pp. 876–884, 2009.

[10] J. Yang and Z. Li, “A novel contour error estimation for position loop-based cross-coupled control”, *IEEE/ASME transactions on mechatronics*, vol. 16, no. 4, pp. 643–655, 2011.

[11] A. Ghaffari and A. G. Ulsoy, “Dynamic contour error estimation and feedback modification for high-precision contouring”, *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1732–1741, 2016.

[12] S. Pi, Q. Liu, and Q. Liu, “A novel dynamic contour error estimation and control in high-speed cnc”, *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 1-4, pp. 547–560, 2018.

[13] X. Li, H. Zhao, J. Yang, and H. Ding, “A high accuracy on-line contour error estimation method of five-axis machine tools”, in *International Conference on Intelligent Robotics and Applications*, Springer, 2015, pp. 565–576.

[14] H.-R. Chen, M.-Y. Cheng, C.-H. Wu, and K.-H. Su, “Real time parameter based contour error estimation algorithms for free form contour following”, *International Journal of Machine Tools and Manufacture*, vol. 102, pp. 1–8, 2016.

[15] C. Hu, Z. Wang, Y. Zhu, and M. Zhang, “Accurate three-dimensional contouring error estimation and compensation scheme with zero-phase filter”, *International Journal of Machine Tools and Manufacture*, vol. 128, pp. 33–40, 2018.

[16] J. Yang, Y. Chen, Y. Chen, and D. Zhang, “A tool path generation and contour error estimation method for four-axis serial machines”, *Mechatronics*, vol. 31, pp. 78–88, 2015.

[17] N. Uchiyama *et al.*, “Estimation of tool orientation contour errors for five-axis machining”, *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 5, pp. 271–277, 2013.

[18] K. Srinivasan and P. Kulkarni, “Cross-coupled control of biaxial feed drive servomechanisms”, *Journal of dynamic systems, measurement, and control*, vol. 112, no. 2, pp. 225–232, 1990.

[19] Y. Koren, “Cross-coupled biaxial computer control for manufacturing systems”, *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 4, pp. 265–272, 1980.

[20] S.-S. Yeh and P.-L. Hsu, “A new approach to bi-axial cross-coupled control”, in *Proceedings of the IEEE International Conference on Control Applications*, IEEE, 2000, pp. 168–173.

[21] C. Lee, C. K. Stepanick, S.-K. Lee, and J. A. Tarbutton, “Cross-coupling effect of large range {xy} nanopositioning stage fabricated by stereolithography process”, *Precision Engineering*, vol. 46, pp. 81–87, 2016.

[22] H.-Y. Chuang and C.-H. Liu, “Cross-coupled adaptive feedrate control for multiaxis machine tools”, *Journal of dynamic systems, measurement, and control*, vol. 113, no. 3, pp. 451–457, 1991.

[23] K. L. Barton and A. G. Alleyne, “A cross-coupled iterative learning control design for precision motion control”, *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1218–1231, 2008.

[24] D.-I. Kim and S. Kim, “On iterative learning control algorithm for industrial robots and cnc machine tools”, in *Proceedings of International Conference on Industrial Electronics, Control, and Instrumentation*, IEEE, 1993, pp. 601–606.

[25] D.-I. Kim and S. Kim, “An iterative learning control method with application for cnc machine tools”, *IEEE Transactions on Industry Applications*, vol. 32, no. 1, pp. 66–72, 1996.

[26] K. L. Moore, “Iterative learning control for deterministic systems”, *Automatica*, vol. 6, no. 32, pp. 948–949, 1996.

[27] M. Uchiyama, “Formation of high-speed motion pattern of a mechanical arm by trial”, *Transactions of the Society of Instrument and Control Engineers*, vol. 14, no. 6, pp. 706–712, 1978.

[28] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of robots by learning”, *Journal of Field Robotics*, vol. 1, no. 2, pp. 123–140, 1984.

[29] ——, “Can mechanical robots learn by themselves?”, in *Proceedings of The 2nd International Symposium on Robotics Research*, 1984, pp. 127–134.

[30] S. Arimoto, “Iterative learning control for robot systems”, in *Proceedings of International Conference on Industrial Electronics, Control, and Instrumentation*, 1984, pp. 393–398.

[31] S. Arimoto, S. Kawamura, and F. Miyazaki, “Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems”, in *The 23rd IEEE Conference on Decision and Control*, IEEE, vol. 23, 1984, pp. 1064–1069.

[32] S. Kawamura, “Hybrid position/force control of robot manipulators based on learning method”, in *Proceedings of the 2nd International Control on Advanced Robotics*, 1985, pp. 235–242.

[33] S. Arimoto, “Mathematical theory of learning with applications to robot control”, in *Adaptive and Learning Systems*, Springer, 1986, pp. 379–388.

[34] ——, “System theoretic study of learning control”, *Transactions of Society of Instrumentation and Control Engineers*, vol. 21, no. 5, pp. 445–450, 1985.

[35] S. Arimoto, S. Kawamura, F. Miyazaki, and S. Tamaki, “Learning control theory for dynamical systems”, in *24th IEEE Conference on Decision and Control*, IEEE, vol. 24, 1985, pp. 1375–1380.

[36] S. Kawamura, F. Miyazaki, and S. Arimoto, “Applications of learning method for dynamic control of robot manipulators”, in *24th IEEE Conference on Decision and Control*, IEEE, vol. 24, 1985, pp. 1381–1386.

[37] S. Arimoto, “Convergence, stability, and robustness of learning control schemes for robot manipulators”, in *Proceedings of the International Symposium*

on Robot Manipulators on Recent Trends in Robotics: Modeling, Control, and Education, 1986, pp. 307–316.

[38] F. Miyazaki, S. Kawamura, M. Matsumori, and S. Arimoto, "Learning control scheme for a class of robot systems with elasticity", in *25th IEEE Conference on Decision and Control*, IEEE, vol. 25, 1986, pp. 74–79.

[39] S. Kawamura, "Intelligent control of robot motion based on learning method", in *Proceedings of IEEE International Symposium on Intelligent Control*, 1987, pp. 365–370.

[40] S. Kawamura, F. Miyazaki, and S. Arimoto, "Realization of robot motion based on a learning method", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 126–134, 1988.

[41] A. Khalik and N. Uchiyama, "Contouring controller design based on iterative contour error estimation for three-dimensional machining", *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 4, pp. 802–807, 2011.

[42] K. R. Simba, N. Uchiyama, and S. Sano, "Iterative contouring controller design for biaxial feed drive systems", in *IEEE 20th Conference on Emerging Technologies & Factory Automation*, IEEE, 2015, pp. 1–5.

[43] K.-S. Hsu, M.-G. Her, and M.-Y. Cheng, "Analysis and experiments of iterative learning-control system with uncertain dynamics", *The International Journal of Advanced Manufacturing Technology*, vol. 25, no. 11-12, pp. 1119–1129, 2005.

[44] B. D. Bui, N. Uchiyama, and K. R. Simba, "Contouring control for three-axis machine tools based on nonlinear friction compensation for lead screws", *International Journal of Machine Tools and Manufacture*, vol. 108, pp. 95–105, 2016.

[45] C.-H. Li and P.-L. Tso, "Experimental study on a hybrid-driven servo press using iterative learning control", *International Journal of Machine Tools and Manufacture*, vol. 48, no. 2, pp. 209–219, 2008.

[46] M.-S. Tsai, C. Yen, and H. Yau, “Development of a hybrid iterative learning control for contouring nurbs curves”, *Asian Journal of Control*, vol. 13, no. 1, pp. 107–125, 2011.

[47] T. Haas, N. Lanz, R. Keller, S. Weikert, and K. Wegener, “Iterative learning for machine tools using a convex optimisation approach”, *Procedia CIRP*, vol. 46, pp. 391–395, 2016.

[48] M.-S. Tsai, C.-L. Yen, and H.-T. Yau, “Integration of an empirical mode decomposition algorithm with iterative learning control for high-precision machining”, *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 878–886, 2013.

[49] J.-X. Xu, S. K. Panda, and T. H. Lee, “Introduction to ilc: Concepts, schematics, and implementation”, in *Real-time iterative learning control: design and applications*, Springer Science & Business Media, 2008, pp. 7–27.

[50] R. Ramesh, M. Mannan, and A. Poo, “Error compensation in machine tools—a review: Part i: Geometric, cutting-force induced and fixture-dependent errors”, *International Journal of Machine Tools and Manufacture*, vol. 40, no. 9, pp. 1235–1256, 2000.

[51] M. M. Isnaini, Y. Shinoki, R. Sato, and K. Shirase, “Development of a cad-cam interaction system to generate a flexible machining process plan”, *International Journal of Automation Technology*, vol. 9, no. 2, pp. 104–114, 2015.

[52] M. Groover and E. Zimmers, *CAD/CAM: computer-aided design and manufacturing*. Pearson Education, 1983.

[53] *NC Analyzer 2: Instruction Manual*. Mitsubishi Electric Corporation, 2015.

[54] C.-C. Lo and C.-Y. Chung, “Tangential-contouring controller for biaxial motion control”, *Journal of dynamic systems, measurement, and control*, vol. 121, no. 1, pp. 126–129, 1999.

[55] R. Ramesh, M. Mannan, and A. Poo, “Tracking and contour error control in cnc servo systems”, *International Journal of Machine Tools and Manufacture*, vol. 45, no. 3, pp. 301–326, 2005.

[56] H.-C. Ho, J.-Y. Yen, and S.-S. Lu, “A decoupled path-following control algorithm based upon the decomposed trajectory error”, *International Journal of Machine Tools and Manufacture*, vol. 39, no. 10, pp. 1619–1630, 1999.

[57] P. Kulkarni and K. Srinivasan, “Optimal contouring control of multi-axial feed drive servomechanisms”, *ASME Journal of Engineering for Industry*, vol. 111, no. 2, pp. 140–148, 1989.

[58] Y.-T. Shih, C.-S. Chen, and A.-C. Lee, “A novel cross-coupling control design for bi-axis motion”, *International Journal of Machine Tools and Manufacture*, vol. 42, no. 14, pp. 1539–1548, 2002.

[59] Z. Jamaludin, H. Brussel, and J. Swevers, “Classical cascade and sliding mode control tracking performances for a xy feed table of a high-speed machine tool”, *International Journal of Precision Technology*, vol. 1, no. 1, pp. 65–74, 2007.

[60] Y. Zhou, F. Yu, J. Xu, S. Hong, *et al.*, “The cascade type iterative learning cross-coupled contour error control method”, *Bulletin of Science and Technology*, vol. 27, no. 5, pp. 737–739, 2011.

[61] Y. M. Hendrawan, K. R. Simba, and N. Uchiyama, “Embedded iterative learning contouring controller design for biaxial feed drive systems”, in *International Electronics Symposium*, IEEE, 2016, pp. 37–41.

[62] J.-X. Xu, S. K. Panda, and T. H. Lee, *Real-time iterative learning control: design and applications*. Springer Science & Business Media, 2008.

[63] Y. Koren and C.-C. Lo, “Advanced controllers for feed drives”, *CIRP Annals-Manufacturing Technology*, vol. 41, no. 2, pp. 689–698, 1992.

[64] Y. M. Hendrawan, K. R. Simba, and N. Uchiyama, “Iterative learning based trajectory generation for machine tool feed drive systems”, *Robotics and Computer-Integrated Manufacturing*, vol. 51C, pp. 230–237, 2018.

[65] Z. Qiao, T. Wang, Y. Wang, M. Hu, and Q. Liu, “Bézier polygons for the linearization of dual nurbs curve in five-axis sculptured surface machining”, *International Journal of Machine Tools and Manufacture*, vol. 53, no. 1, pp. 107–117, 2012.

[66] M.-H. Cho, D.-W. Kim, C.-G. Lee, E.-Y. Heo, J.-W. Ha, and F. F. Chen, “Cbims: Case-based impeller machining strategy support system”, *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 6, pp. 980–988, 2009.

[67] T. Umehara, K. Teramoto, T. Ishida, and Y. Takeuchi, “Tool posture determination for 5-axis control machining by area division method”, *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, vol. 49, no. 1, pp. 35–42, 2006.

[68] C.-C. Lo, “Efficient cutter-path planning for five-axis surface machining with a flat-end cutter”, *Computer-Aided Design*, vol. 31, no. 9, pp. 557–566, 1999.

[69] X. Beudaert, S. Lavernhe, and C. Tournier, “5-axis local corner rounding of linear tool path discontinuities”, *International Journal of Machine Tools and Manufacture*, vol. 73, pp. 9–16, 2013.

[70] L. Guran, “Third-order local contact and application in 5-axis machining of sculptured surfaces”, *Chinese Journal of Mechanical Engineering*, vol. 19, no. 2, pp. 265–267, 2006.

[71] F. Huo and A.-N. Poo, “Improving contouring accuracy by using generalized cross-coupled control”, *International Journal of Machine Tools and Manufacture*, vol. 63, pp. 49–57, 2012.

[72] K. R. Simba, N. Uchiyama, and S. Sano, “Real-time smooth trajectory generation for nonholonomic mobile robots using bézier curves”, *Robotics and Computer-Integrated Manufacturing*, vol. 41, pp. 31–42, 2016.

[73] K. R. Simba, G. Heppeler, B. D. Bui, Y. M. Hendrawan, O. Sawodny, and N. Uchiyama, “Bézier curve based trajectory generation and nonlinear friction compensation for feed drive contouring control”, *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1944–1951, 2017.

[74] L. Biagiotti and C. Melchiorri, “Multidimensional trajectories and geometric path planning”, in *Trajectory planning for automatic machines and robots*, Springer Science & Business Media, 2008, pp. 341–414.

[75] B. Sencer, K. Ishizaki, and E. Shamoto, “A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of nc systems along linear tool paths”, *The International Journal of Advanced Manufacturing Technology*, vol. 76, no. 9-12, pp. 1977–1992, 2015.

[76] A. W.-J. Hsue, M.-T. Yan, and S.-H. Ke, “Comparison on linear synchronous motors and conventional rotary motors driven wire-edm processes”, *Journal of Materials Processing Technology*, vol. 192, pp. 478–485, 2007.

[77] C. Choi and T.-C. Tsao, “Control of linear motor machine tool feed drives for end milling: Robust mimo approach”, *Mechatronics*, vol. 15, no. 10, pp. 1207–1224, 2005.

[78] A. Kochan, “Machine tools turn to linear motors.”, *Design news*, vol. 54, no. 18, pp. 80–82, 1999.

[79] D. M. Alter and T.-C. Tsao, “Control of linear motors for machine tool feed drives: Experimental investigation of optimal feedforward tracking control”, *Journal of dynamic systems, measurement, and control*, vol. 120, no. 1, pp. 137–142, 1998.

[80] D. Renton and M. Elbestawi, “Motion control for linear motor feed drives in advanced machine tools”, *International Journal of Machine Tools and Manufacture*, vol. 41, no. 4, pp. 479–507, 2001.

[81] B. Denkena, H. Tönshoff, X. Li*, J. Imiela, and C. Lapp, “Analysis and control/monitoring of the direct linear drive in end milling”, *International journal of production research*, vol. 42, no. 24, pp. 5149–5166, 2004.

[82] H. Itagaki and M. Tsutsumi, “Control system design of a linear motor feed drive system using virtual friction”, *Precision Engineering*, vol. 38, no. 2, pp. 237–248, 2014.

[83] S.-L. Chen and T.-H. Hsieh, “Repetitive control design and implementation for linear motor machine tool”, *International Journal of Machine Tools and Manufacture*, vol. 47, no. 12-13, pp. 1807–1816, 2007.

[84] Y. M. Hendrawan and N. Uchiyama, “Embedded iterative learning contouring controller based on precise estimation of contour error for cnc machine tools”, in *26th Mediterranean Conference on Control and Automation*, IEEE, 2018, pp. 1–6.

[85] Z.-y. Jia, J.-w. Ma, D.-n. Song, F.-j. Wang, and W. Liu, “A review of contouring-error reduction method in multi-axis cnc machining”, *International Journal of Machine Tools and Manufacture*, vol. 125, pp. 34–54, 2018.

[86] M. Tomizuka, “Zero phase error tracking algorithm for digital control”, *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 1, pp. 65–68, 1987.

[87] D. Torfs, J. De Schutter, and J. Swevers, “Extended bandwidth zero phase error tracking control of nonminimal phase systems”, *Journal of dynamic systems, measurement, and control*, vol. 114, no. 3, pp. 347–351, 1992.

[88] L. Tian, Z. Xiong, J. Wu, and H. Ding, “A comprehensive inversion approach for feedforward compensation of piezoactuator system at high frequency”, *Smart Materials and Structures*, vol. 25, no. 9, p. 095 046, 2016.

[89] W. Sun, Y. Zhang, Y. Huang, H. Gao, and O. Kaynak, “Transient-performance-guaranteed robust adaptive control and its application to precision motion control systems”, *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6510–6518, 2016.

[90] J. Bolder and T. Oomen, “Rational basis functions in iterative learning control—with experimental verification on a motion system”, *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 722–729, 2015.

[91] J. Wu, Y. Han, Z. Xiong, and H. Ding, "Servo performance improvement through iterative tuning feedforward controller with disturbance compensator", *International Journal of Machine Tools and Manufacture*, vol. 117, pp. 1–10, 2017.

[92] V. Stojanovic, N. Nedic, D. Prsic, L. Dubonjic, and V. Djordjevic, "Application of cuckoo search algorithm to constrained control problem of a parallel robot platform", *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 9-12, pp. 2497–2507, 2016.

[93] P. R. Ouyang, T. Dam, and V. Pano, "Cross-coupled pid control in position domain for contour tracking", *Robotica*, vol. 33, no. 6, pp. 1351–1374, 2015.

[94] S.-S. Yeh and P.-L. Hsu, "Theory and applications of the robust cross-coupled control design", *Journal of dynamic systems, measurement, and control*, vol. 121, no. 3, pp. 524–530, 1999.

[95] J.-H. Chin, Y.-M. Cheng, and J.-H. Lin, "Improving contour accuracy by fuzzy-logic enhanced cross-coupled precompensation method", *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 1, pp. 65–76, 2004.

[96] W. Chen, D. Wang, Q. Geng, and C. Xia, "Robust adaptive cross-coupling position control of biaxial motion system", *Science China Technological Sciences*, vol. 59, no. 4, pp. 680–688, 2016.

[97] F. Huo, X.-C. Xi, and A.-N. Poo, "Generalized taylor series expansion for free-form two-dimensional contour error compensation", *International Journal of Machine Tools and Manufacture*, vol. 53, no. 1, pp. 91–99, 2012.

[98] J. Dong, T. Wang, B. Li, and Y. Ding, "Smooth feedrate planning for continuous short line tool path with contour error constraint", *International Journal of Machine Tools and Manufacture*, vol. 76, pp. 1–12, 2014.

[99] Z.-y. Jia, D.-n. Song, J.-w. Ma, G.-q. Hu, and W.-w. Su, "A nurbs interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints", *International Journal of Machine Tools and Manufacture*, vol. 116, pp. 1–17, 2017.

[100] K. Zhang, J.-X. Guo, and X.-S. Gao, “Cubic spline trajectory generation with axis jerk and tracking error constraints”, *International Journal of Precision Engineering and Manufacturing*, vol. 14, no. 7, pp. 1141–1146, 2013.

[101] D. Zhang, Y. Chen, and Y. Chen, “Iterative pre-compensation scheme of tracking error for contouring error reduction”, *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 9-12, pp. 3279–3288, 2016.

[102] Z. Jia, L. Wang, J. Ma, K. Zhao, and W. Liu, “Feed speed scheduling method for parts with rapidly varied geometric feature based on drive constraint of nc machine tool”, *International Journal of Machine Tools and Manufacture*, vol. 87, pp. 73–88, 2014.

[103] M. Chen, J. Xu, and Y. Sun, “Adaptive feedrate planning for continuous parametric tool path with confined contour error and axis jerks”, *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 1-4, pp. 1113–1125, 2017.

[104] C. A. Ernesto and R. T. Farouki, “Solution of inverse dynamics problems for contour error minimization in cnc machines”, *The International Journal of Advanced Manufacturing Technology*, vol. 49, no. 5-8, pp. 589–604, 2010.

[105] D.-n. Song, J.-w. Ma, Z.-y. Jia, and Y.-y. Gao, “Estimation and compensation for continuous-path running trajectory error in high-feed-speed machining”, *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 5-8, pp. 1495–1508, 2017.

[106] P. Dolezel and J. Mares, “Self tuning pid control using genetic algorithm and artificial neural networks”, *ASR 2009 Instruments and Control*, pp. 33–39, 2009.

[107] I. Chiha, N. Liouane, and P. Borne, “Tuning pid controller using multiobjective ant colony optimization”, *Applied Computational Intelligence and Soft Computing*, vol. 2012, pp. 1–7, 2012.

[108] M. Sadasivarao and M Chidambaram, “Pid controller tuning of cascade control systems using genetic algorithm”, *Journal of Indian Institute of Science*, vol. 86, no. 7, pp. 343–354, 2006.

[109] J. Gradišek, M. Kalveram, and K. Weinert, “Mechanistic identification of specific force coefficients for a general end mill”, *International Journal of Machine Tools and Manufacture*, vol. 44, no. 4, pp. 401–414, 2004.

[110] N. Uchiyama, K. Goto, and S. Sano, “Analysis of energy consumption in fundamental motion of industrial machines and experimental verification”, in *American Control Conference*, IEEE, 2015, pp. 2179–2184.

[111] E. W. Nshama and N. Uchiyama, “Time and energy optimal trajectory generation in feed drive systems using kinematic corner smoothing with interrupted acceleration”, in *26th Mediterranean Conference on Control and Automation*, IEEE, 2018, pp. 102–107.

Publication Lists

The contents of this thesis are the results of original research and have not been submitted for a higher degree to any other university or institution.

Much of work presented in this thesis has been published as journal or conference papers. Following is a list of these papers.

International Journal Papers

- Y. M. Hendrawan, K. R. Simba, and N. Uchiyama, "Iterative Learning Based Trajectory Generation for Machine Tool Feed Drive Systems", *Robotics and Computer-Integrated Manufacturing*, vol. 51C, pp. 230 - 237, 2018. (Q1, *Impact Factor: 3.464*)

- Y. M. Hendrawan, A. Farrage, and N. Uchiyama, "Iterative NC Program Modification and Energy Saving for a CNC Machine Tool Feed Drive System with Linear Motors", *The International of Advanced Manufacturing Technology*, pp. 1 - 20, 2019. (Q1, *Impact Factor: 2.601*)

International Conferences Papers

- Y. M. Hendrawan, K. R. Simba, and N. Uchiyama, "Embedded Iterative Learning Contouring Controller Design for Biaxial Feed Drive Systems", International Electronics Symposium (IES), IEEE, pp. 37-41, September 29-30, 2016, Bali, Indonesia.

- K. R. Simba, G. Heppeler, B. D. Bui, Y. M. Hendrawan, O. Sawodny, and N. Uchiyama, "Bézier Curve Based Trajectory Generation and Nonlinear Friction Compensation for Feed Drive Contouring Control", The 20th International Federation of Automatic Control (IFAC) World Congress, pp. 1944-1951, July 9-14, 2017, Toulouse, France. (Finalist Paper for Application Paper Prize)

- Y. M. Hendrawan and N. Uchiyama, "Embedded Iterative Learning Contouring Controller Based on Precise Estimation of Contour Error for CNC Machine Tools", IEEE The 26th Mediterranean Conference on Control and Automation (MED), pp. 412-417, June 19-22, 2018, Zadar, Republic of Croatia.
- N. Uchiyama and Y. M. Hendrawan, "Application of Iterative Learning and Adaptive Control to Feed Drive Systems", The MTTRF and iAM-CNC Annual Meeting 2018, pp. 177-182, July 1-3, 2018, San Fransisco, USA. (Invited speech)
- Y. M. Hendrawan and N. Uchiyama, "Application of Iterative Learning Contouring Control to a Machine Tool Feed Drive System with Linear Motors", International Symposium on System Integration 2019 (SII), pp. 159-164, January 14-16, 2019, Paris, France.

Yogi Muldani Hendrawan

Toyohashi University of Technology

July, 2018