Title
Automation and Precision Control for Intraocular Robotic Interventional Surgical System

Author
Chen, Cheng-Wei

Publication Date
2018

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Automation and Precision Control for Intraocular Robotic Interventional Surgical System

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mechanical Engineering

by

Cheng-Wei Chen

2018

ABSTRACT OF THE DISSERTATION

Automation and Precision Control for Intraocular Robotic Interventional Surgical System

by

Cheng-Wei Chen
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2018
Professor Tsu-Chin Tsao, Chair

To improve surgical outcomes and reduce surgical complications in intraocular surgery, an optical coherence tomography (OCT) imaging system is integrated into the intraocular robotic interventional surgical system (IRISS). The OCT images are used for preoperative planning and intraoperative intervention in a series of automated procedures. High-precision motion control of the robot manipulator is enabled by leveraging learning-type control algorithms to the data-based feedforward filter design. Real-time intervention allows a surgeon to evaluate the surgical progress and manually override the autonomous tracking of the predefined trajectory. The developed system was experimentally validated by performing lens extraction, which is a critical surgical step in cataract surgery, on 30 post-mortem pig eyes. Complete lens extraction was achieved on 25 eyes, and "almost complete" extraction was achieved on the rest due to the inability of the OCT to image minute-sized particles of lens behind the iris. No posterior capsule rupture occurred for any of the 30 pig eyes. This work successfully demonstrated automated OCT-guided intraocular surgery using a robotic surgical system.

The dissertation of Cheng-Wei Chen is approved.

Jean Pierre Hubschman

Jacob Rosen

Veronica Santos

Lieven Vandenberghe

Tsu-Chin Tsao, Committee Chair

University of California, Los Angeles

2018

To Mia

whom feeds me never saturated love and patience;

To Manchester

whom tries his best for not being unobservable and uncontrollable.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my deepest thanks to Professor T.-C. Tsao for his guidance and support since the very first time we met at the National Taiwan University. Professor Tsao has infinite passion in research and more than infinite patience in education. Although he has a time-varying delay on reading our writings, he could always give us timely feedback when we need to be stabilized. For that I am very grateful. I'd also like to thank all my committee members, Professors Jean Pierre Hubschman, Jacob Rosen, Veronica Santos, and Lieven Vandenberghe, for their valuable insights and helpful suggestions.

This thesis includes the materials published or to be published. The system integration and surgical automation will be published soon in [CLG18]; the iterative inversion identification was first published in [CT16] and it will be published with more experimental results in [CRT18]; the iterative learning control with progressive updates was published in [CT17]. Thanks to all the co-authors' efforts and contributions.

Thanks to Yen-Chi Chang, Kuo-Tai Teng, Chris Kang, James Simonelli, Niloufar Esfandi, Sandeep Rai, Grant Cavalier, Matt Gerber, Martin Lee, Jack LeCheng, Han Wang, and Edwin Rachmawan in the Mechatronics and Controls Laboratory. Especially to Martin and Matt, together we made the IRISS moving again. This work couldn't be done without your help.

Finally, thanks to my parents, my brother, my wife, and my little boy. All the achievements I've done belong to you.

Vita

2009	B.S. (Electrical and Control Engineering), National Chiao Tung University, Hsinchu, Taiwan.
2011	M.S. (Electrical Engineering), National Taiwan University, Taipei, Taiwan.

Publications

Y. C. Chang, C. W. Chen, and T.-C. Tsao, "Near Time-Optimal Real-Time Path Following Under Error Tolerance and System Constraints," Journal of Dynamic Systems, Measurement and Control, vol. 140, no. 7, pp. 071004, 2018.

J. Wilson, M. J. Gerber, S. Prince, C. W. Chen, S. Schwartz, J. P. Hubschman, and T.-C. Tsao, "Intraocular Robotic Interventional Surgical System (IRISS): Mechanical Design and MasterSlave Manipulation," International Journal of Medical Robotics and Computer Assisted Surgery, vol. 14, no. 1, pp. e1841, 2018.

C. W. Chen, Y. C. Chang, and T.-C. Tsao, "Dynamic Trajectory Tracking by Synergistic Dual-Stage Actuation and Control," IEEE/ASME Transactions on Mechatronics, vol. 22, no. 6, pp. 2600–2610, 2017.

C. W. Chen and T.-C. Tsao, "Data-Driven Progressive and Iterative Learning Control," in proceedings of the IFAC 2017 World Congress, vol. 50, no. 1, pp. 4825–4830, Toulouse, France, Jul. 2017.

J. Simonelli, Y. H. Lee, S. Mikaiel, C. W. Chen, X. Li, K. Sung, D. Lu, H. Wu, and T.-C Tsao, "An MR-Compatible Stage for Respiratory Motion Emulation," in proceedings of the IFAC 2017 World Congress, vol. 50, no. 1, pp. 6073–6078, Toulouse, France, Jul. 2017.

C. W. Chen and T.-C. Tsao, "Data-Based Feedforward Controller Reconstruction from Iterative Learning Control Algorithm," in proceedings of the IEEE/ASME Int'l Conf. on Advanced Intelligent Mechatronics, pp. 683–688, Banff, Canada, Jul. 2016.

Y. C. Chang, C. W. Chen, and T.-C. Tsao, "Real-Time Sub-Count Estimation With State Continuity for Asynchronous and Quantized Sensing," IEEE/ASME Transactions on Mechatronics, vol. 21, no. 3, pp. 1457–1466, 2016.

Y. Wang, M. E. Saad, K. Ni, Y. C. Chang, C. W. Chen, C. Chen, L. Pan, T.-C. Tsao, A. S. Lavine, D. B. Bogy, and X. Zhang, "Scalable Plasmonic Nanolithography: Prototype System Design and Construction," in proceedings of the ASME 2016 11th Int'l Manufacturing Science and Engineering Conf., pp. V001T02A081, Blacksburg, Virginia, Jun. 2016.

M. C. Ke, Y. H. Tseng, C. W. Chen, M. C. Ho, F. L. Lian, J. Y. Yen, W. L. Lin, and Y. Y. Chen, "Preliminary Study of Intracorporeal Localization for Endoscopy," in proceedings of the 2013 CACS Int'l Automatic Control Conf., pp. 130–134, Taiwan, Dec. 2013.

C. W. Chen and Y. Y. Chen, "Recovering Depth from a Single Image Using Spectral Energy of the Defocused Step Edge Gradient," in proceedings of the IEEE Int'l Conf. on Image Processing (ICIP), pp. 1981–1984, Brussels, Belgium, Sep. 2011.

CHAPTER 1

Introduction

Overcoming physiological limitations of a human surgeon and deficiencies in sensing capabilities, robotic surgical platforms have been developed and applied to various kinds of microsurgery for reducing surgical complications and improving surgical outcomes. Despite the success of the Intuitive Surgical's Da Vinci Surgical Platform in general surgery, the robot-assisted intraocular surgery which deals with particular microscale, complexity, and sensitivity still requires continuous effort in the research of robotics and control. For example, cataracts—an opaque clouding of the lens in the eye—are the leading cause of blindness in the world [PM12]. Cataract-induced blindness is remedied by cataract surgery, which is the most frequently performed surgical procedure in the United States [OBC17]. With a success rate over 90%, cataract surgery represents one of the most successful intraocular procedures [Nat]. However, surgical complications still remain, including posterior capsule rupture (1.8–4.4%), incomplete lens removal (1.1%), and corneal incision leakage (1.2%) [DMR99]. To improve surgical outcomes, it is necessary to improve visualization of anatomical features inside the eye and to use that information to autonomously and precisely guide a surgical instrument.

1.1 Automation of Intraocular Surgical Procedures

To visualize intraocular tissues, non-invasive imaging technologies such as magnetic resonance imaging, ultrasound biometry, and optical coherence tomography (OCT) have been developed and used in preoperative and postoperative diagnoses. Magnetic resonance imaging provides high-contrast and high-resolution images [FF12], but is unsuitable for intraop-

erative surgical procedures due to physical space constraints, high field strengths, and low frame rates. High-frequency ultrasound biometry provides real-time, high-resolution images [Sil09], but requires a contact medium between the probe and the eye to reduce signal reflection. In contrast, OCT provides a non-contact, real-time, high-resolution imaging modality that can be integrated into surgical instruments and microscopes [FS16, CVK17]. Most importantly, it has been shown that intraocular tissues can be visualized in OCT scans in both anterior [RRR01] and posterior segments [EPH10].

Cataract surgery (refer to Fig. 1.1) involves a suite of procedural steps including (1) creating a corneal incision, (2) removing the anterior capsule via capsulorhexis, (3) fragmenting the cataract into pieces of lens material, (4) emulsifying and aspirating the lens material using an ultrasonic surgical instrument, (5) aspirating the remaining lens material with an irrigation-aspiration (I/A) tool, and (6) inserting an intraocular lens implant. The first three steps (corneal incision, capsulorhexis, and fragmentation) have been partially or fully automated by OCT-based femtosecond laser systems [DBC13, PSF11, Abb, LEN, Alcb]. Automation of the final step (implant insertion) has also been investigated to improve precision over that of a human surgeon [AHS12]. However, the cataract-extraction procedure remains a manually performed operation, despite being the main source of the aforementioned surgical complications.

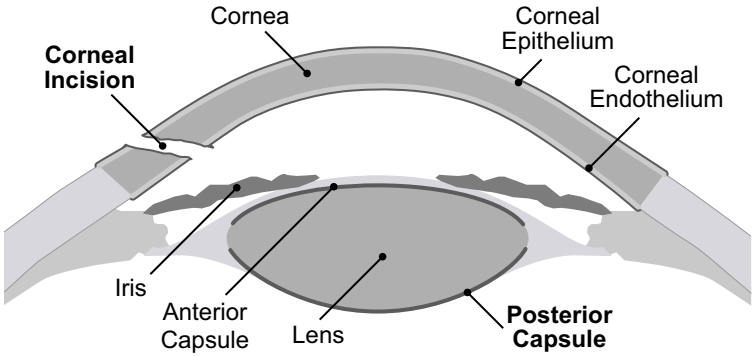


Figure 1.1: A schematic of the anterior segment with relevant features labeled. Both the posterior capsule and the corneal incision are important features and are repeatably referred to throughout this thesis.

Even if provided with the "best" visual feedback, a surgeon cannot perform surgical procedures absolutely accurately unless the physiological limitations of tactile control are overcome. To address the stringent requirements of intricate tool manipulation, robot-assisted surgical platforms have been investigated by several groups. Johns Hopkins University developed a steady-hand manipulator for membrane peeling [UBH10]. The University of Tokyo demonstrated the feasibility of a robotic system in performing vitreoretinal tasks such as posterior vitreous detachment, vessel sheathotomy, and microcannulation [UYS09, THI15]. Vanderbilt University demonstrated robotic capabilities of conducting constrained ocular manipulation, membrane peeling, and stent deployment [YSJ13]. The University of Utah designed a compact, remotely operated manipulator for disposable instruments and performed membrane peeling on an eye phantom [NBA15]. The University of Munich tested the efficacy of a target-based injection platform on pig eyes aimed at assisting the treatment of age-related macular degeneration [NML17]. In our previous work, the IRISS successfully demonstrated the feasibility of remotely operated cataract surgery and vein cannulation via mechanism design and motion filtering [WGP18].

Recognizing the advantages of integrating high-resolution OCT imaging with a high-precision robotic platform, several groups have demonstrated success in robot-assisted intraocular surgical intervention using OCT. A team from the Cleveland Clinic integrated an OCT probe with a surgical microscope to provide high-resolution visual feedback [ESF14]. More recently, the same group proposed using a stereo-tracked i-OCT system to automatically deploy scans in real-time near the surgical instrument tip to ease the effort of probe re-allocation [ET15]. Vanderbilt University integrated an OCT optic fiber cable into the tool and conducted epiretinal membrane peeling using a master-slave configuration guided by OCT B-scans [YSS15]. The same team also demonstrated a semi-automated micro-injection via the assistance of three-dimensional virtual fixtures based on both microscope and B-mode OCT feedback [YSJ16]. Johns Hopkins University incorporated an A-mode OCT as a distal sensor with a piezo-motor in the hand-held tool piece to achieve active depth-locking control [CHC15]. The depth-locking feature was applied to the OCT-embedded micro-forceps for

epiretinal membranectomy [CGT17].

Despite the advances in both robotic platforms and OCT-based technologies, several unresolved issues remain, particularly in the case of cataract extraction. First, the location where a surgical instrument passes through the cornea (referred to as the "corneal incision") must be constrained throughout surgery to decrease undesirable trauma in adjacent corneal tissue. However, the methods in existing work to align the remote center of motion (RCM) of the robot to the corneal incision are contact-based and require well-calibrated kinematics. In addition, no methods exist to automate the alignment procedure. Second, despite the wealth of anatomical information provided by OCT-based systems, no existing work uses OCT scans to reconstruct the surgical environment for trajectory planning of the surgical instrument or for automation of the surgical procedures. Third, the surgical information acquired from real-time A or B mode OCT used in previous work [YSJ16, CGT17] is noisy and insufficient to represent the constantly changing intraocular environment. In other words, an automated surgical platform must be capable of adapting, in real-time, to the dynamic nature of its surgical workspace.

This thesis addresses these concerns by developing and implementing a series of automated intraocular surgical procedures on the OCT-guided robotic platform:

1. An automated procedure that utilizes laser landmarks to reliably align the RCM of the robot to the corneal incision. In contrast to existing methods, this procedure enables the minimization of physical stress in adjacent corneal tissue during surgical operations.

2. A method that uses OCT scans to generate a three-dimensional parameterized model of the anatomical eye structure with safe surgical zones of operation.

3. A method to use the eye model to plan a safe, efficient tool-tip trajectory through the workspace. This method has the advantage of preventing tissue damage and posterior capsule rupture by reducing the risk of inadvertent collision between the instrument and tissues.

4. The development of intraoperative diagnostics and intervention methods to allow the surgeon to override or modify any portion of the automated procedure, including tool-tip trajectory and the predefined workspace. This functionality improves the flexibility and safety of the automation in the event of unexpected disturbances.

1.2 Precise Motion Control for Robot Manipulator

Since intraocular anatomical structure is delicate and fragile, it is important for the robot manipulator to precisely track the predefined trajectory generated during the preoperative planning stage. Feedforward control is commonly applied to applications which require high-precision motion tracking control [YHP17]. Among the efforts, iterative learning control (ILC), which utilizes error signals to update the feedforward inputs, has shown excellent tracking performance on a class of dynamic systems such as chemical processing [CQL04], robot manipulators [Tay04, WNG11], CNC machining [KT04], wafer stage motion systems [BA08], and stroke rehabilitation [FRB15]. A typical limitation of ILC is that the tracking reference must retain identical during the trials. Any slight perturbation of the reference requires certain amount of iterations to converge again for the optimal feedforward input [MTB08].

To relax the reference limitation, ILC is enhanced by imposing basis functions [MTB08, WB10, HS01, FLM17, BO15b, BOS15, BBB16]. Polynomial basis functions were applied to parameterize the feedforward input in terms of the reference [MTB08, WB10, HS01, FLM17]. It requires iterative trials and the Newton method to determine the unknown coefficients on top of the polynomial basis functions. Once the coefficients are determined, a finite-impulse-response (FIR) filter is constructed to filter the reference for the feedforward input. Recently, rational basis function is also investigated for better performance [BO15b, BOS15, BBB16]. Although it is beneficial when impossible to apply a high-order FIR filter due to the limitation of computation resources, this method is challenged by the non-convex optimization for the synthesis of the infinite-impulse-response (IIR) filter.

Basis function approaches are similar to using ILC for system identification, a.k.a. Iterative Learning Identification (ILI) [CSS08, JPS13, LA14], but they directly identify the plant inversion as an FIR (or an IIR when imposing a rational basis function) filter. If plant dynamic model is available, model-based approaches can also be applied to invert the plant dynamics instead of applying iterative learning regarding basis functions. In fact, the performance of the ILC algorithm is determined by its learning filter. It has been shown that the convergence rate can be maximized by applying a well-approximate plant inversion as the learning filter [BTA06, NG02, TT15], and therefore reference variation would not be a problem.

For a minimum phase system with an accurate plant model, exact inversion can be obtained. However, for a typical resonant system, the phase angle around the resonant frequency ramps down by -180 degrees very quickly. It is challenging to acquire an accurate model of this kind of rapid phase changing. Inaccurate modeling around the resonant frequency necessities a manual fine tune of the number of delay/preview steps of the feedforward filter to reduce the phase tracking error. Besides, if there is a non-minimum phase zero in the plant, which is common in applications such as linear motors, an approximation is needed to keep the feedforward filter being stable. Various model-based approaches for handling the non-minimum phase plant inversion problem of a Single-Input-Single-Output (SISO) system have been proposed, such as the zero-magnitude-error tracking controller (ZMETC) [RPL09], zero-phase-error tracking controller (ZPETC) [Tom87, TT87, KT14], time-domain direct inversion (DI) [CT14]. Multi-Input Multi-Output (MIMO) plant inversion, however, is mostly ignored since its complexity and non-commutative property.

Data-based plant inversion approaches, such as filtered B-spline basis [DYO17], Modeling-Free Inversion-Based Iterative Feedforward Control (MIIFC) [KZ13], and above-mentioned basis function approaches that identify the plant inversion using iterative trials, are attractive because the requirement of an accurate plant model is eliminated. Filtered B-spline basis requires determination of basis inputs and the corresponding output signals prior to reference tracking. For a linear time-invariant (LTI) system, superposition of the inputs

that generate the desired output patterns guarantees an optimal feedforward input for any arbitrary reference trajectory. However, solving the least-square problem is required for every new trajectory. On the other hand, the target reference adopted for learning trials in either basis function approaches or MIIFC is never been specified and discussed, though it is an important factor since the plant dynamics must be fully excited by the reference model within the bandwidth of interest.

Intuitively, using a delta impulse as the reference model in basis function approaches results in an FIR feedforward filter which approximates the plant inversion. However, the delta impulse which implies all-band inversion is too aggressive and may induce control saturation during the learning process. Therefore, the proposed method relaxes the delta impulse to a filtered impulse according to the selection of desired inversion bandwidth. Borrowing the idea from ILI, we apply the learning-type control, e.g. repetitive control and ILC, to identify the plant inversion and create the optimal feedforward filter for LTI systems. Instead of using the Newton method to search the coefficients as in ILI, the learned input which generates the smoothed impulse directly forms the impulse response (i.e. the FIR coefficients) of the FIR feedforward filter. This high-order FIR filter can also be approximated by an IIR filter after employing balance realization [BKC92]. Therefore this approach is comparable to the data-driven approaches that apply a rational basis function when the length of the FIR filter is sufficiently long. The proposed method is able to handle both SISO and MIMO non-minimum phase systems under the same framework. No complicated synthesis method or transformation is required for obtaining the plant inversion of a MIMO system.

This study reveals an interesting *chicken-and-egg* problem: It is well-known that the ideal learning filter of the learning-type control in terms of convergence rate is the inverse of the controlled system, but this study shows that the plant inversion can also be constructed from any converged learning-type control scheme (e.g. repetitive control [INK81], PD-type [CM02] and time-reversal based ILC [BO15a]). This study also shows that the identified inversion filter can be applied as the learning filter of the ILC algorithm to further improve its performance in terms of convergence rate. This approach discloses another degree of

freedom for ILC algorithms: for any stable ILC, the iteration of input signals brings smaller tracking error, while the progression of learning filter improves error convergence rate.

1.3 Main Contributions

The main contributions of this thesis are:

1. The integration of the perioperative OCT imaging system and the surgical robot—the IRISS—for OCT-guided robotic intraocular surgery.

2. The automation of critical intraocular surgical procedures, including alignment, modeling of anatomy, preoperative planning, and real-time supervision and intervention.

3. The development of precise motion control using data-based inversion identification for robot manipulator and other physical plants.

These contributions were experimentally validated by performing lens extraction on 30 post-mortem pig eyes.

This thesis is organized as follows. Chapter 2 provides a system overview and introduces the integrated components of the OCT-guided robotic platform. Chapter 3 describes the automated procedures for intraocular surgery using the OCT-guided robotic platform. Chapter 4 presents the proposed method of precise motion control for the robot manipulator. Chapter 5 extends the work in Chapter 4 to further improve the error convergence rate of the ILC algorithm. Chapter 6 presents the experimental results and evaluations on post-mortem pig eyes. Conclusion is given in Chapter 7.

CHAPTER 2

The IRISS for Surgical Automation

The overall system architecture of the OCT-guided robotic system is shown in Fig. 2.1. The IRISS controls both the tool-tip position (here demonstrated by an I/A hand-piece) and the irrigation and aspiration forces generated by the Alcon ACCURUS® system [Alca]. The tool-tip position is programmed based on the anatomical model reconstructed from OCT volume scans, which are obtained from the OCT probe physically positioned over the eye (Fig. 2.2). The OCT and embedded camera provide real-time visual feedback to the surgeon. The surgeon can override the IRISS during automated operation to assess surgical progress and to account for variations in the surgical environment.

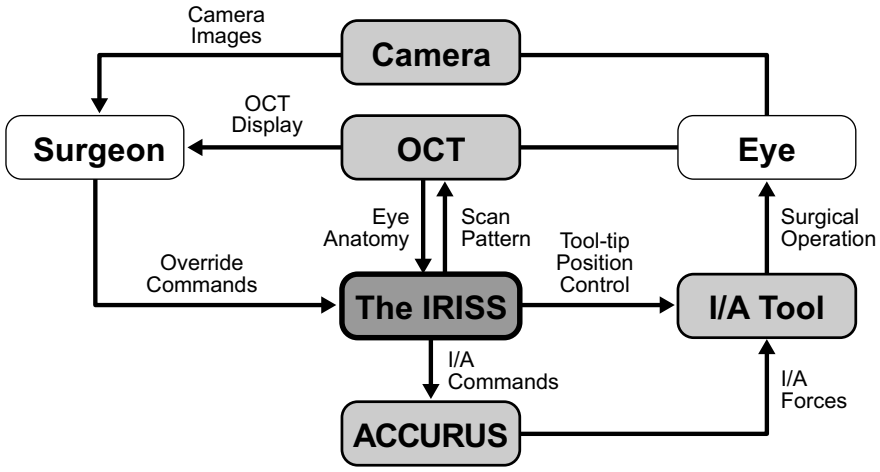


Figure 2.1: The overall system architecture for the OCT-guided robotic system.

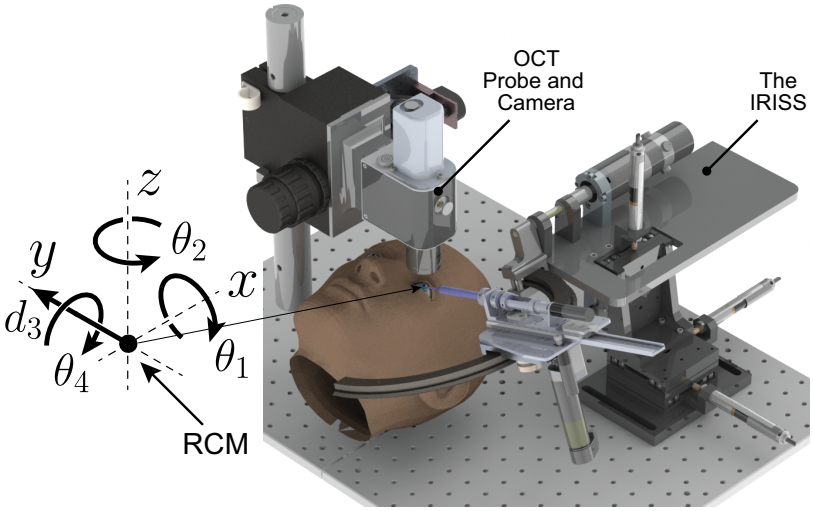


Figure 2.2: The CAD model of the integrated system with the coordinate frame and kinematic variables defined.

2.1 The IRISS

The IRISS consists of two, independently controllable manipulator arms, each holding two interchangeable surgical instruments [WGP18]. Because only a single I/A tool is required for lens extraction on pig eyes, only one manipulator arm equipped with a single I/A tool is used in this study. The I/A tool is mounted to a carriage that rides on a circular track which is rotated about a shaft mounted to the base of the IRISS. This spherical mechanism kinematically guarantees a mechanically fixed RCM. To allow three-dimensional translation of the RCM position relative to the eye, the arm assembly is mounted to a motorized linear stage capable of three-dimensional XYZ translation. This feature enables eye tracking, while not addressed in this work, to compensate for possible eyeball movement as the patient is awake during some specific surgery such as cataract surgery. Further details of the mechanical design of the IRISS and an evaluation of its performance can be found in [WGP18].

For convenient mathematical representation, the Cartesian coordinate frame of the IRISS is chosen to be coincident with the RCM (Fig. 2.2). Desired tool-tip positions are achieved by the coordination of two rotational angles (θ_1 and θ_2) and one translational displacement (d_3). Kinematically, the tool rotation about its centerline (θ_4) has no effect on tool-tip

position. Mechanical calibration is performed prior to surgery to ensure the commands $\theta_1 = \theta_2 = \theta_4 = 0$ and $d_3 = 0$ result in the tool tip located coincident with the RCM and the aspiration port facing anterior. Given a tool-tip position $\boldsymbol{p} = [p_x\ p_y\ p_z]^T$ inside the eye, the inverse kinematics are given by

$$\begin{cases} \theta_1 = \text{atan2}\left(-p_z,\ p_y\right) \\ \theta_2 = \text{atan2}\left(-p_x,\ \sqrt{p_y^2 + p_z^2}\right) \\ d_3 = \|\boldsymbol{p}\|_2 \end{cases} \tag{2.1}$$

where $\text{atan2}\left(\cdot, \cdot\right)$ is the four-quadrant inverse tangent function with two arguments and $\|\cdot\|_2$ is the Euclidean norm.

For the duration of intraocular surgery, the tool tip remains inside the eye and Eq. 2.1 defines a unique map from the tool-tip position to the robotic joint space. However, when the tool tip is outside the eye, motion control is realized in joint space and the kinematic singularity at $[0\ 0\ 0]^T$ (the RCM) is avoided. Doing so has the advantage of directly coupling the desired insertion angles to joint angle commands.

2.2 The OCT Imaging System

A commercially available ThorLabs SD-OCT imaging system (Telesto II 1060LR) is integrated with the IRISS (Fig. 2.2). The OCT system operates with a broadband superluminescent diode with a central wavelength of 1060 nm, axial resolution of 9.18 µm, and imaging depth in air of 9.4 mm. The objective lens (LSM04BB) exhibits a focal length of 54 mm, a lateral resolution of 25 µm, and a 10×10 mm field of view. The OCT probe has been vertically actuated with a custom stage for changing its depth of view.

The integrated OCT system is capable of performing three-dimensional volume scans, real-time A and B scans, and custom scanning patterns programmed through the provided software development kit. Volume scans are used for modeling of anatomical structure (Chapter 3.2) and intermittent assessment (Chapter 3.4). Real-time B scans and custom

scanning patterns are used for real-time intraoperative intervention (Chapter 3.4). Although it is common for anterior-segment OCT to use a laser with central wavelength of 1310 nm to penetrate tissue to a greater depth [RRR01], this system uses an illumination source with central wavelength of 1060 nm as a balance between signal attenuation and imaging depth. To image the anterior segment of pig eyes (which have a thicker lens than that of humans [SMU11]), tissue penetration is sacrificed in favor of signal transmittance by shortening the wavelength of the illumination source.

2.3 System Integration

The hardware-integration architecture is shown in Fig. 2.3. The IRISS is controlled by a National Instruments PXI real-time target at a sampling rate of 1 kHz. Brushed DC motors on the IRISS joints are driven by current-type amplifiers and the control loops are closed by rotary optical encoders integrated into the motors. Aspiration and irrigation commands are sent from the NI PXI target to the ACCURUS® via a pulse-width modulated signal. The joint motions and I/A command either follow a trajectory provided by the trajectory planner

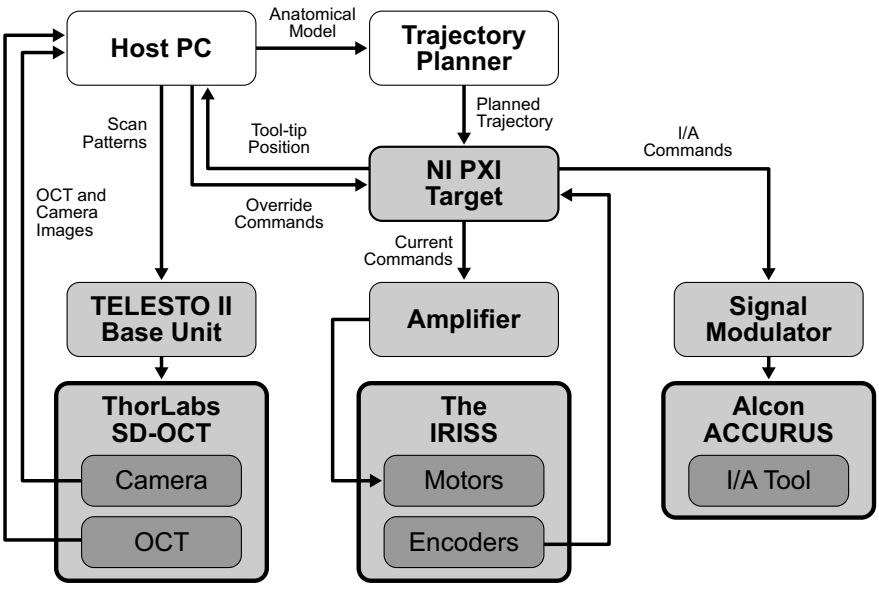


Figure 2.3: The diagram of the hardware architecture.

(Chapter 3.3) or are directly controlled by user command. The host PC acquires OCT scans and camera images from the OCT system, which is driven by a base unit that positions its A-scans. The host PC runs with a graphical user interface programmed in LabVIEW. Through this interface, the user can supervise real-time scanning of the workspace, and—if necessary—override the tool-tip trajectory or the I/A commands (Chapter 3.4).

To enable control of the IRISS using OCT feedback, the registration between the OCT reference frame and the IRISS frame was required. To derive this relationship, the IRISS was autonomously commanded to touch a series of points that formed a unique, non-symmetric pattern representational of the intraocular workspace. At each point, the tool-tip position was determined in the OCT frame from medium-sensitivity (48 kHz) volume scans by applying a custom, image-processing algorithm. At each point, two representations existed: the tool-tip position in the IRISS frame from the ideal forward kinematics and the tool-tip position in the OCT frame from the volume scan. To find the linear transformations (translation, rotation, and uniform scaling) between the frames, Procrustes superimposition was performed [GD04]. With the linear transformations and their inverses known, any point in the OCT reference frame can be converted to a point in the IRISS reference frame—and vice versa. This initial registration is dynamically updated in real-time when the IRISS or the OCT probe are moved, thereby maintaining the validity of the registration despite relative motion between the physical hardware.

To test the accuracy of the OCT–IRISS coordinate frame relationship, the I/A tool was commanded to touch a series of $n = 30$ randomly generated points within its workspace. At each point, a volume scan was acquired, the tool-tip position was determined, and the ideal tool-tip position was calculated from the forward kinematics. The coordinate frame transformation was used to transform the tool-tip positions determined by the OCT scan into the IRISS frame. Ideally, these points would perfectly overlap the points obtained from the forward kinematics with zero error; in reality, some error exists. This error was calculated as the 3D Euclidean distance between each pair of points. The statistical measures of the errors for a typical derivation of the coordinate transformation are shown in Table 2.1.

Table 2.1: Error of Typical Coordinate Transformation

Min.	Mean	RMS	Max.	Std.
0.047	0.19	0.21	0.34	0.077

* All values in units of mm

2.4 Software Design

The software state chart is shown in Fig. 2.4. In the mechanism homing state, each joint is incrementally actuated until its homed position is found via a photo-interrupter sensor. In the initialization state, the I/A tool is commanded to a safe, predefined initial position and the RCM of the IRISS is automatically aligned to the corneal incision in the eye (Chapter 3.1). Then, the I/A tool is inserted into the eye, a surgical trajectory is loaded into the software, and the tool begins tracking the predefined trajectory (Chapter 3.3 and 4). During the tracking, the user may pause the motion or reroute the tool tip to a specific point. The tracking process is considered complete when either the end of the trajectory has been reached or a finishing command has been sent from the user. At this point, the tool returns to the engaged state to await tool retraction or additional track commands.

For safety reasons, emergency retraction and shutdown are included in the state chart. For example, in the event of homing sensor malfunction, the program will terminate to prevent hardware damage. On the other hand, emergency tool retraction is employed in the states where the tool tip is within the eye. To prevent surgical damage that would occur during emergency shutdown if the tool remained inside the eye, the system is powered off only when the tool is fully retracted and any existing operations have been successfully terminated.

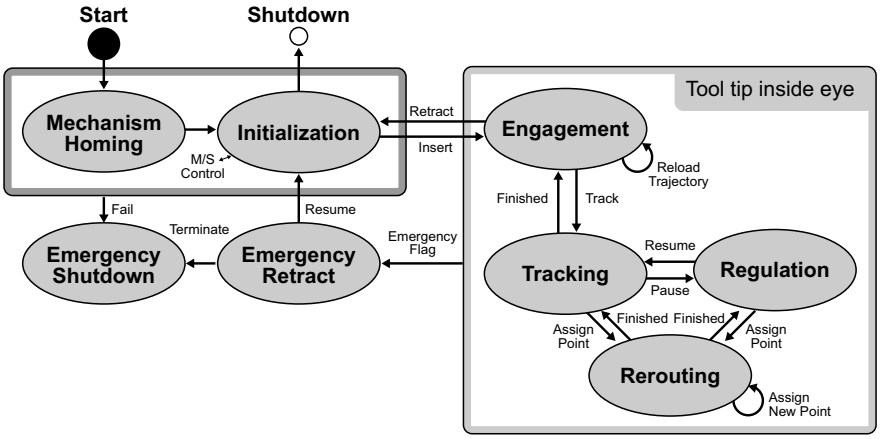


Figure 2.4: The software state chart for the OCT-guided surgical robotic system.

CHAPTER 3

Automation of Robotic Intraocular Surgery

The core contribution of this thesis is the proposed methodology for automated intraocular surgery, as illustrated in the flow chart Fig. 3.1. This methodology can be divided into four distinct steps: laser-assisted alignment (Chapter 3.1), segmentation and modeling of anatomical structures (Chapter 3.2), preoperative trajectory planning (Chapter 3.3), and real-time intraoperative supervision and intervention (Chapter 3.4).

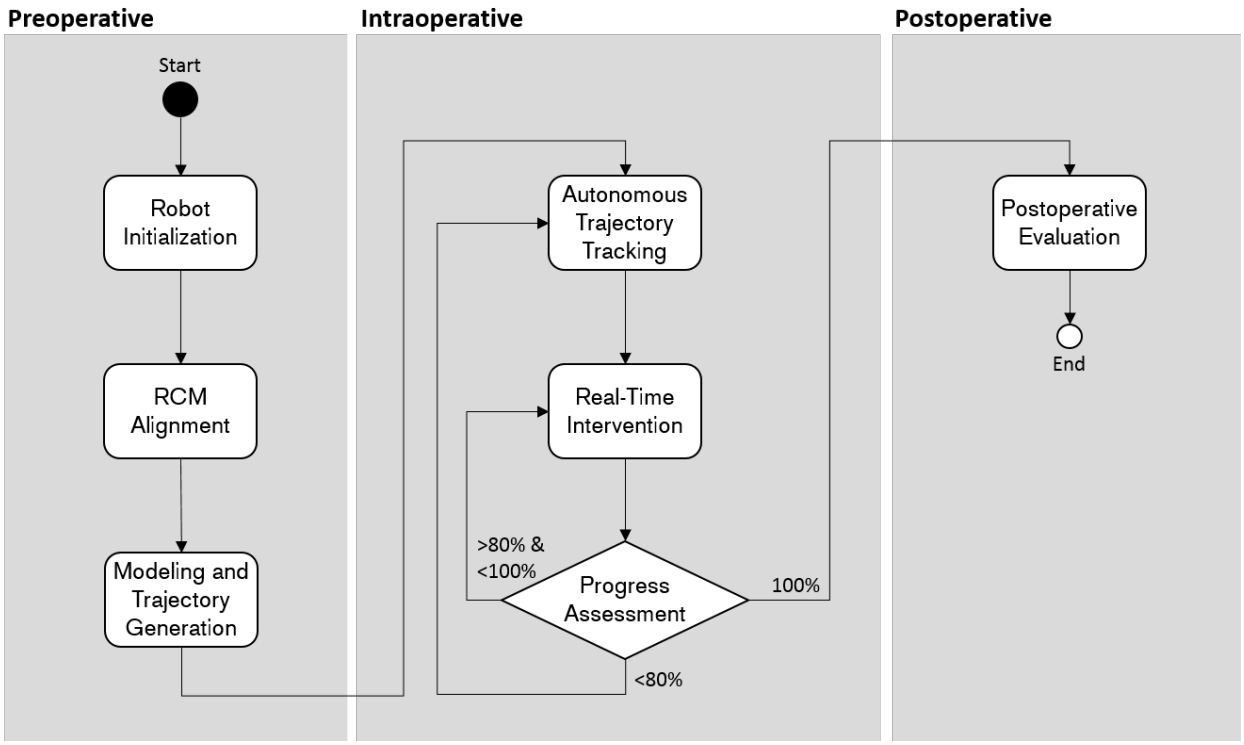


Figure 3.1: The flow chart of automated intraocular surgery.

3.1 Laser-Assisted Alignment

To reduce stress at the entry site during intraocular surgery, the first step of automated intraocular surgery involves aligning the IRISS to the eye such that the mechanical RCM of the robot is coincident with an approximation of the corneal incision, a point $\boldsymbol{p}^*_{CI}$. If this condition is not satisfied, then the self-sealing properties of the surrounding corneal tissue may be diminished, resulting in post-operative corneal incision leakage. This clinical requirement necessitates the use of a mechanical RCM and forms the basis for the mechanical design of the IRISS robot.

Though the RCM is an invisible point in space, two lasers were mounted on the tool carriage to intersect at the RCM (Fig. 3.2). When the visible points produced by the lasers converge at a single point on a surface, that point is coincident with the mechanical RCM. In this way, aligning the mechanical RCM of the IRISS to the corneal incision point can be automatically performed by integrating control of the automated XYZ stage with computer-vision feedback. After the surgeon creates the corneal incision, visual feedback guides the stage to first align one of the laser spots to the incision, and then registers the optical axis of the aligned laser. By moving the stage along the registered optical axis and searching for the minimal detected area of the laser points on the microscope image, alignment of the RCM to the incision point can be achieved. This process requires less than two minutes to complete and is only required once per surgery.

Due to mechanical construction error of the lasers themselves, as well as the assembly and manufacturing tolerance stack-up error of the laser mount sub-system, it cannot be guaranteed that the laser beams will accurately intersect at $\boldsymbol{p}_{RCM}$ after mounting. For example, typical commercial LED lasers have a deflection angle of $\pm 10^\circ$ and a beam offset of ± 1 mm. This fact necessitates a post-mounting precision adjustment mechanism on the laser mounts, which are incorporated into the design as two orthogonally mounted thumb screws. The laser is secured in a spherical bearing, and the thumb screws—when tightened—push into the laser housing thereby changing the pitch angle ϕ_i and the yaw angle ψ_i of the laser

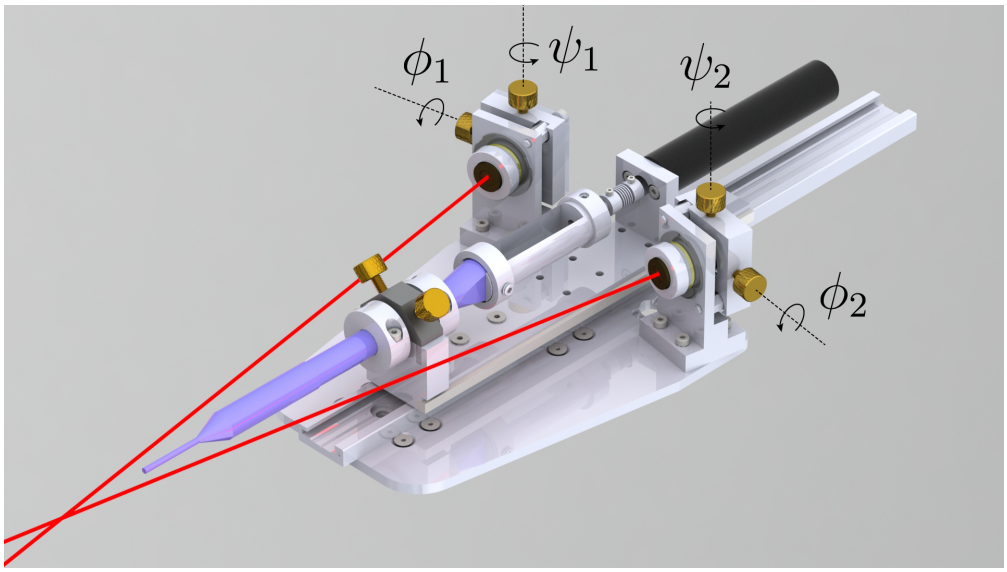


Figure 3.2: The CAD model of the RCM laser tool holder (isometric view).

beam. Compression springs provide a return motion.

The laser beams were independently aligned to pass through $\boldsymbol{p}_{RCM}$ by a manual, two-step procedure. First, a flat precision-ground plate was mounted perpendicular to the fixed robot base and placed such that its surface was aligned with the $\hat{Z}$ axis. The carriage was then repeatedly cycled through its θ_2 rotation and the laser yaw angle ψ_i manually adjusted until the rotational motion repeatedly produced a stationary laser point on the flat plate. Once satisfied, the angular deflection in ψ_i was considered to be sufficiently reduced. Second, the plate was moved along its normal direction (along $\hat{X}$) and the laser pitch angle ϕ_i was manually adjusted such that the laser point remained stationary as the plate was moved, thereby sufficiently reducing the angular deflection in ϕ_i. Once complete, the process was repeated on the other laser. After this alignment process, the lasers were considered aligned to intersect at the RCM and could be confirmed by actuating the tool tip through the RCM and noting the laser deflection. While the manual nature of this process was slow and inherently inaccurate, it must only be performance once, and was considered adequate for the purpose of visualizing the RCM location in space and aligning the RCM with a surgical incision in the eye.

3.2 Segmentation and Modeling of Anatomical Structure

To account for anatomical variation between eyes, a pair of preoperative OCT volume scans are acquired of the anterior segment. Because the depth of the posterior capsule (approximately 10–15 mm from the top of the cornea) is greater than the sensing depth of the OCT (9.4 mm), the OCT probe is physically translated via its translational stage to cover the entire range of interest. Two scans are sufficient: a representative B-scan of the upper portion is shown in Fig. 3.3 and includes the iris and cornea; a representative B-scan of the lower portion is shown in Fig. 3.4 and includes the posterior capsule and the inverted iris. The iris appears inverted in the lower scan due to the limited sensing depth of the OCT system. To avoid overlapping the posterior capsule with the inverted iris, the lower scan is taken when the posterior capsule appears (by visual inspection) within the top 3 mm of the OCT frame and the inverted iris is located in the bottom 5 mm (Fig. 3.4).

After acquisition, the two scans are stitched together with the known displacement of the OCT probe to create a single, composite scan of the entire anterior segment. The volume scan data are converted to a point cloud by automated binary thresholding. The point-cloud represents tissue whose reflection is stronger than that of water or balanced salt solution and is therefore a down-sampled representation of anatomical features of interest. For tool-tip navigation within the eye, it is critical to determine the positions of the cornea, iris, and posterior capsule from this point-cloud data.

Although various approaches exist for OCT image segmentation of anterior structures, all require significant computational resources to produce accurate results [WZB13]. If implemented, the surgical consequence would be a protracted delay prior to the start of surgical procedures, during which the cornea will begin to collapse. Therefore, a custom algorithm was developed that considers *a priori* knowledge of the eye anatomy to quickly generate (within one minute) a parameterized anatomical model with acceptable accuracy (< 100 µm). The pupil was modeled as a 2D ellipse in 3D space. For segmentation of the cornea, a second-order parabolic surface was chosen to represent the corneal epithelium and was fit to

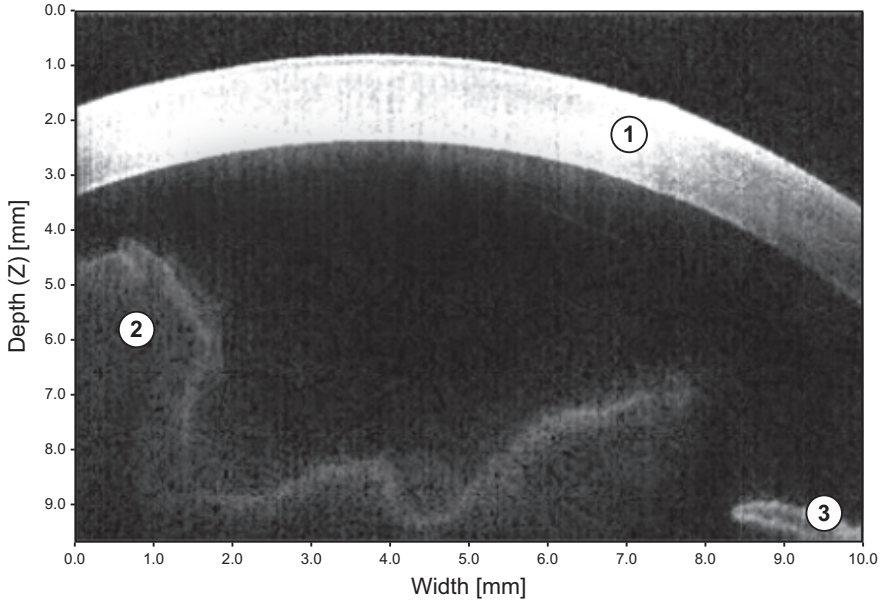


Figure 3.3: An OCT scan of the upper anterior segment including (1) the cornea, (2) lens material, and (3) the iris.

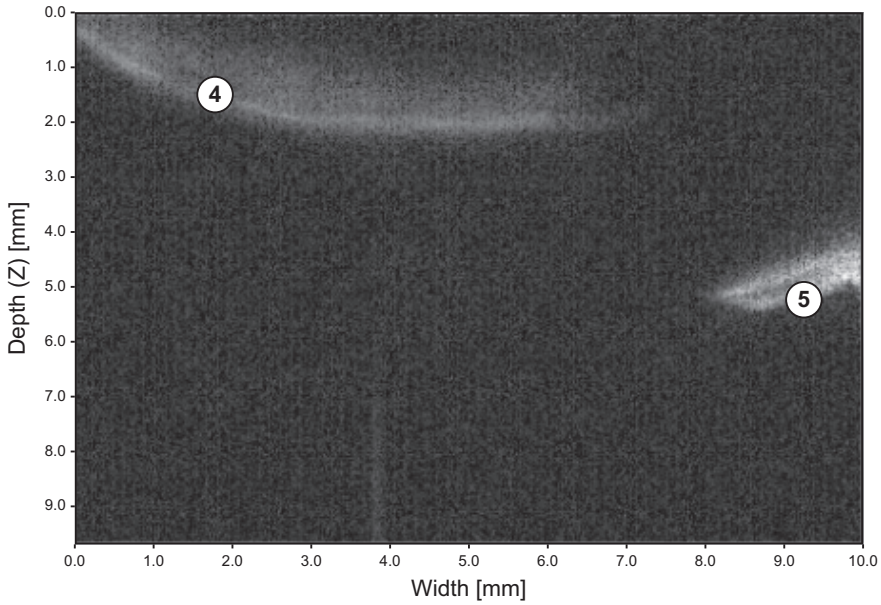


Figure 3.4: An OCT scan of the lower anterior segment including (4) the lens and posterior capsule and (5) the inverted iris.

the point-cloud data using a least-squares method. This fit takes advantage of the knowledge that the cornea must be located near the top of the scan volume. Similarly, for segmentation of the posterior capsule, a second-order parabolic surface was chosen.

The three generated surfaces define the workspace within the anterior segment (Fig. 3.5). This model is parameterized and is used to preoperatively plan the cataract-extraction trajectory. Due to the effective range of the I/A tool aspiration, a safety margin around all three anatomical features was enforced in the model. This guaranteed that any composite error from the segmentation and modeling was alleviated, providing the error magnitude was less than this margin.

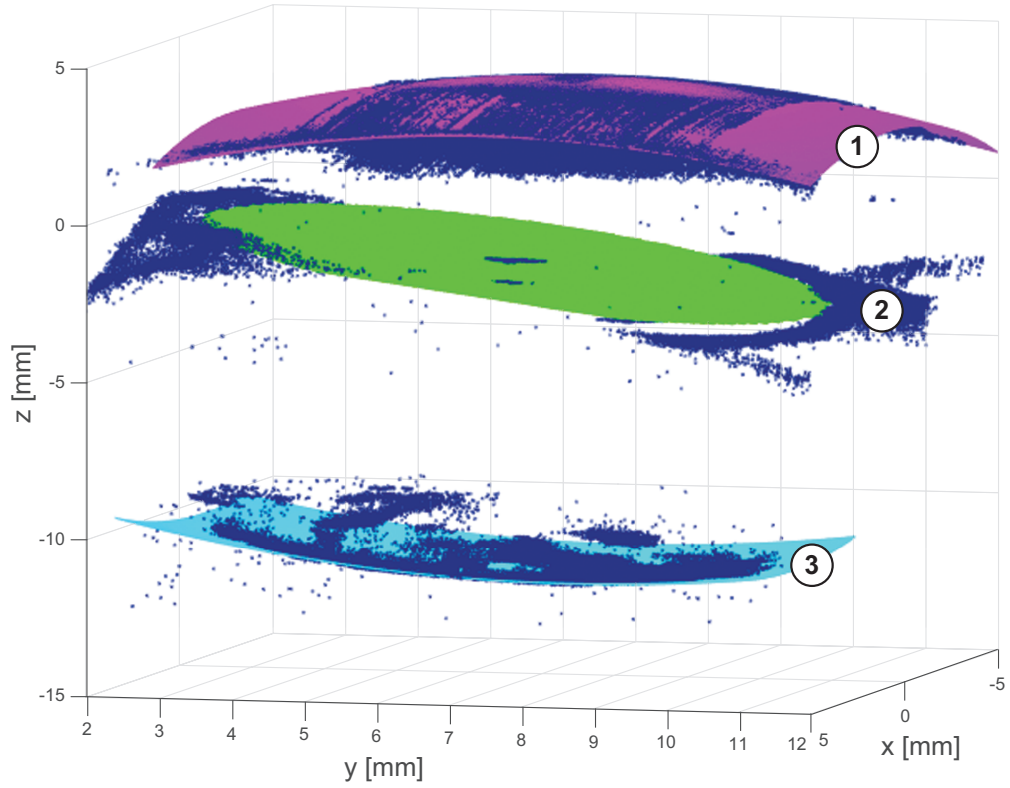


Figure 3.5: The reconstructed eye model. (1) Magenta: reconstructed cornea, (2) green: reconstructed pupil, (3) cyan: reconstructed posterior capsule, blue points: raw OCT data.

It is important to note that the medium through which the OCT laser signal propagates is variable. Specifically, when the laser signal passes from air into the corneal epithelium, the refractive index changes from 1 to approximately 1.35 (that of water). For this reason,

the depth values acquired from the OCT system, z_o, are adjusted to update the anatomical model to account for this change:

$$z_o' = \begin{cases} z_o, & z_o < z_e \\ z_e + n\left(z_o - z_e\right), & z_o \geq z_e \end{cases} \tag{3.1}$$

where z_o' is the adjusted depth value, z_e is the depth of the corneal epithelium along an A-scan line, and n is the ratio of the refractive index of air over that of water ($n \approx 0.74$).

To evaluate the accuracy of the anatomical model, it was compared to human-labeled ground truths on $n = 10$ randomly sampled OCT images from nine different eyes. The accuracy of posterior capsule detection, with the worst image quality among all tissue, was 79.6 ± 23.3 µm (the 95% confidence interval around the average). This modeling error was considered negligible in comparison to the safety margins established near intraocular tissue, which are on the order of 1.5 mm.

3.3 Preoperative Trajectory Planning for Cataract Extraction

In this step, a tool-tip trajectory is generated based on the eye model obtained in the previous step, allowing the IRISS to guide the surgical instrument with respect to anatomical structures within the eye. To demonstrate the capability of performing automated cataract extraction, the generated tool-tip trajectory exhibits the following characteristics:

(i) The geometry of the trajectory is chosen to mobilize the lens material and to increase surgical efficacy and efficiency.

(ii) The tool-tip motion deepens and decelerates after each repetition of a baseline trajectory to avoid posterior capsule rupture.

(iii) The aspiration and θ_4 are scheduled as functions of the tool-tip position to increase surgical efficacy and provide additional protection against posterior capsule rupture.

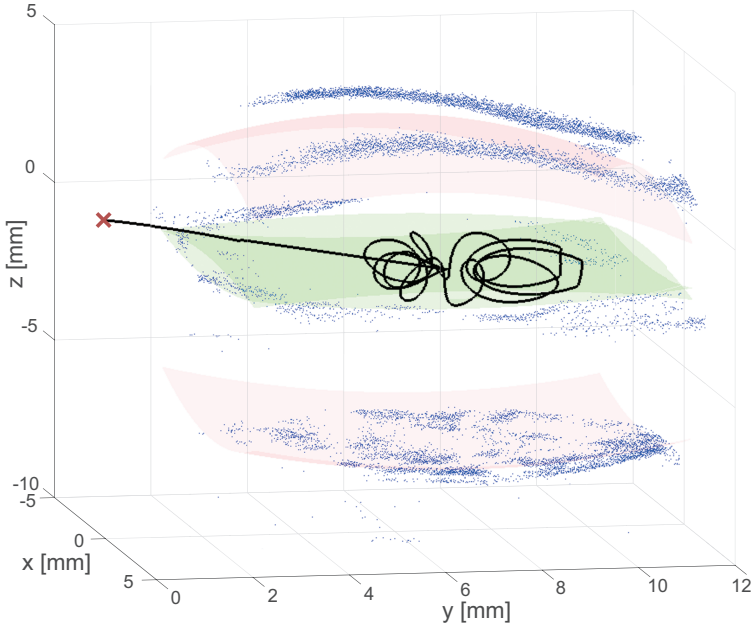


Figure 3.6: The preoperatively planned cataract-extraction trajectory in the IRISS frame. Point cloud: raw OCT data, solid black line: tool-tip trajectory, red surfaces: safety margins, green surfaces: tool-tip workspace, red X: the RCM.

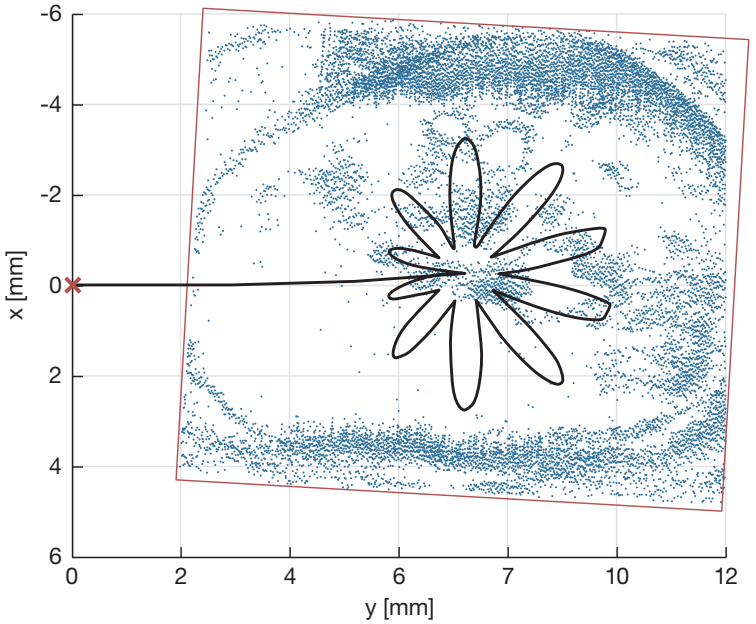


Figure 3.7: The top view of Fig. 3.6. Point cloud: raw OCT data, solid black line: tool-tip trajectory, red X: the RCM, red box: OCT scanning volume.

3.3.1 Trajectory Pattern Design for Lens Removal

Based on surgeon input and clinical feedback, a flower-shaped pattern with a scooping motion was developed (Fig. 3.6 and 3.7). The geometric shape in the plane of the pupil is intended to disaggregate conglomerated lens material, while the scooping motion is intended to detach lens material from the posterior capsule. Given a desired tool-tip position, $\boldsymbol{p}(t) = [p_x(t)\ p_y(t)\ p_z(t)]^T$, then

$$\begin{cases} p_x(t) = c_x + q\cos(\theta(t)) \\ p_y(t) = c_y + q\sin(\theta(t)) \\ p_z(t) = c_z - wa\sin(n\theta(t)) \end{cases} \tag{3.2}$$

where $\boldsymbol{c} = [c_x\ c_y\ c_z]^T$ is the coordinate of the iris center, $w(\theta)$ is a window function between $[0, 1]$ such that the trajectory will start and end at $\boldsymbol{c}$, $a(\theta)$ is a position-dependent amplitude bounded by the distances to the anterior and posterior capsular surfaces, n is the number of "flower petals" in the geometric pattern, and $q(\theta)$ is the two-dimensional motion of the tool-tip in polar coordinates defined as (Fig. 3.8):

$$q(\theta) = (sR - wr)\sin^2\left(\frac{n\theta(t)}{2}\right) + wr \tag{3.3}$$

where $s(\theta)$ is the radial scaling of the flower pattern which preserves the safety gap between the tool-tip and iris, R is the pupil radius, r is the inner radius of the flower pattern, and $\theta(t)$ is the angle between the x-axis and a line from $\boldsymbol{c}$ to the tool tip in the range $[0, 2\pi]$. The value of $\theta(t)$ was chosen with equidistant sampling in polar coordinates with an approximately averaged tool-tip velocity of 0.6 mm/s and cycle time of 90 s.

Based on clinical experience, it is necessary to move the tool slower and deeper after each cycle to detach and aspirate lens material. Therefore, the tool moves fastest for the first cycle (approximately averaged tool-tip velocity: 0.6 mm/s) and at a conservative, shallow depth (within the top 30% of the capsular bag thickness). Each subsequent cycle progresses

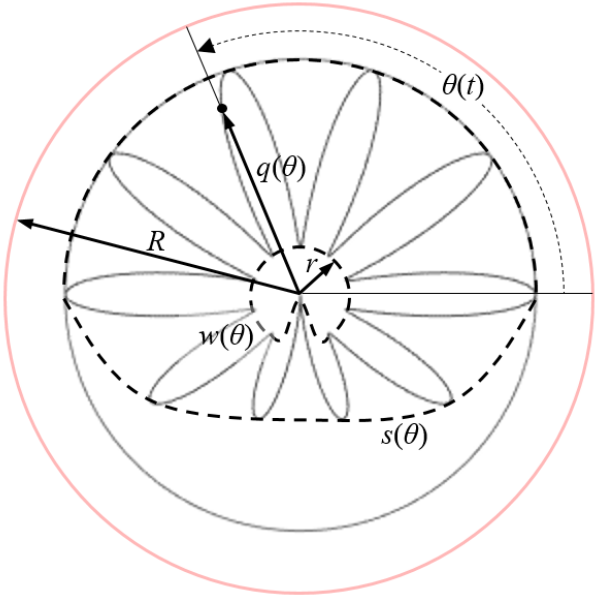


Figure 3.8: The definition of parameters for the generation of cataract-extraction trajectory.

the tool tip deeper (+10% lens thickness per cycle) and slower (−25% tool-tip velocity).

3.3.2 Instrument Rotation

The aspiration port of the I/A tool is located on the side of the instrument near its tip. Posterior capsule rupture may occur when the port aspirates the capsular surface in close proximity. For this reason, θ_4 is scheduled such that the aspiration port is always facing away from the posterior capsule. Also, when the tool tip is near the equator of the capsular bag, the normal vector of the port is directed towards a virtual axis above the iris center $\boldsymbol{c}$,

$$\left\{(x, y, z) \in \mathbb{R}^3 : x = c_x \text{ and } z = c_z + \Delta z_a\right\} \tag{3.4}$$

where $\Delta z_a > 0$ indicates the offset between the iris center and the virtual axis. The position-dependent instrument rotation can be described as:

$$\theta_4(t) = \text{atan2}\left(p_x - c_x,\ p_z - c_z - \Delta z_a\right) \tag{3.5}$$

3.3.3 Aspiration Force Scheduling

Even if the aspiration direction is accounted for, posterior capsule rupture may still occur if the aspiration *magnitude* is ignored. Therefore, the aspiration is also scheduled based on $\Delta z_p(t)$, the distance between the tool tip and the posterior capsule (Fig. 3.9). This is achieved by the following position-dependent aspiration scheduling:

$$f_a(t) = f_{a,lb} + (f_{a,ub} - f_{a,lb})\,\mathrm{sat}\left[\frac{\Delta z_p(t) - z_{lb}}{z_{ub} - z_{lb}}\right] \tag{3.6}$$

where $f_a(t)$ is the aspiration force as a function of tool-tip position; $(\cdot)_{lb}$ and $(\cdot)_{ub}$ are shorthand notations for the lower and upper bounds of a parameter; $\mathrm{sat}(\cdot)$ is the saturation function with both domain and co-domain $[0, 1]$; and z_{lb} and z_{ub} are user-defined bounds on the magnitude of $\Delta z_p(t)$. This saturation is necessary to prevent the aspiration force from becoming too aggressive or from deforming the intraocular tissue, while providing sufficient force to continue aspirating the lens material.

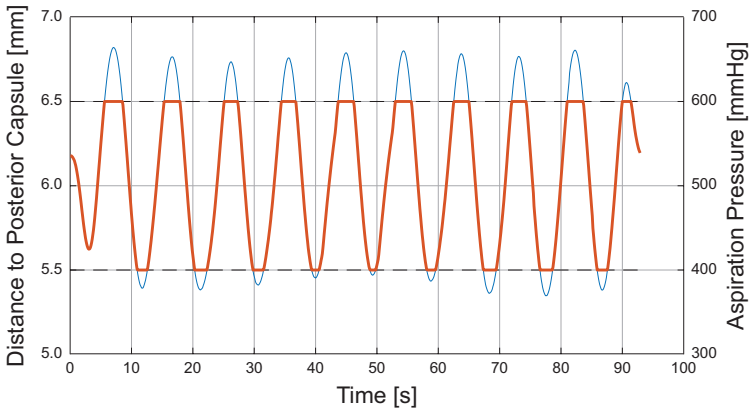


Figure 3.9: An example of aspiration force profile scheduled along the tool-tip trajectory with $z_{lb} = 5.5$ mm, $z_{ub} = 6.5$ mm, $f_{a,lb} = 400$ mmHg, and $f_{a,ub} = 600$ mmHg.

Once the cataract-extraction trajectory is generated, a top view of the calculated tool-tip trajectory is projected on an anterior view of the eye in the graphical user interface. Likewise, the side view of the calculated tool-tip trajectory is projected on a superior view of the eye. If any adjustment is requested by the surgeon, offsets and range resizing can be

manually performed.

3.4 User Interface for Intraoperative Intervention

Automated intraocular surgery is conducted by tracking the predefined trajectory. However, while the preoperative scan is useful in constructing an initial anatomical model and tool trajectory, the intraoperative anterior segment is a dynamic environment subject to change during surgical operations. To account for the changing surgical environment, intraoperative supervision and manual intervention strategies are provided to the surgeon during the autonomous surgical procedure.

3.4.1 Real-Time OCT Assessment

During automated tracking, either real-time OCT B-scans (Fig. 3.10) or localized B-scans that intersect and track the tool tip (Fig. 3.11) are displayed to the user for monitoring surgical progress. Both custom, real-time OCT scans can be acquired to provide timely information regarding the surgical instrument and anatomical features.

Acknowledging the difficulty inherent to real-time image processing of OCT images, the user interface was designed with a focus on improving visualization and enabling directed targeting of lens material (Fig. 3.11). The OCT probe can also be translated to track the posterior capsule, lens material, or other features of interest. The frame rate of a B-scan (the time interval between each update without saving the data) is approximately 8 Hz and is limited by the motion bandwidth of the mechanical galvo mirror. The frame rate of a localized scan volume is approximately 1.5 Hz.

3.4.2 Intervention Strategy

Based on the real-time OCT feedback, several options are provided to the user.

(i) The user can offset and resize the workspace of the predefined trajectory. The trajectory

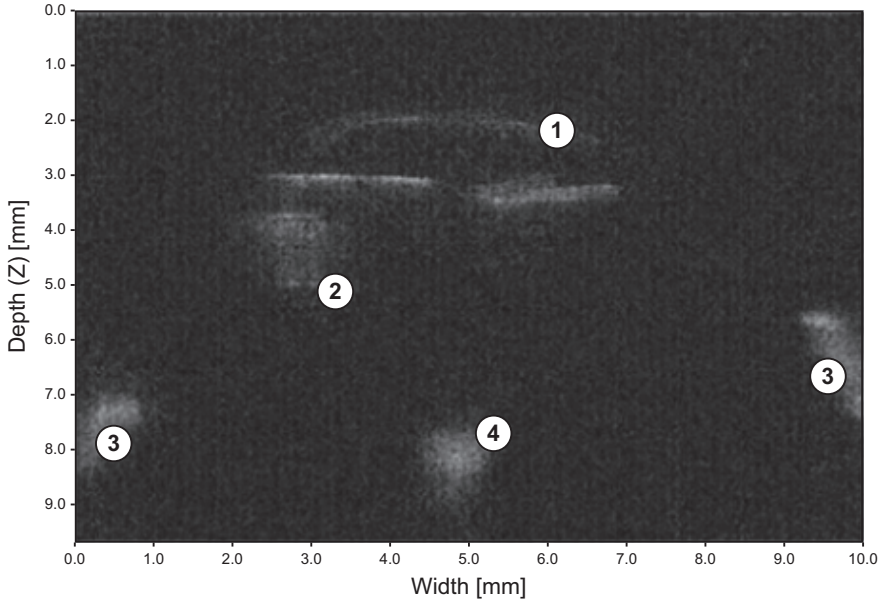


Figure 3.10: An example of the real-time OCT image feedback. Visible features are (1) the posterior capsule, (2) lens material, (3) the iris, and (4) the tool tip. Note: all images are inverted.

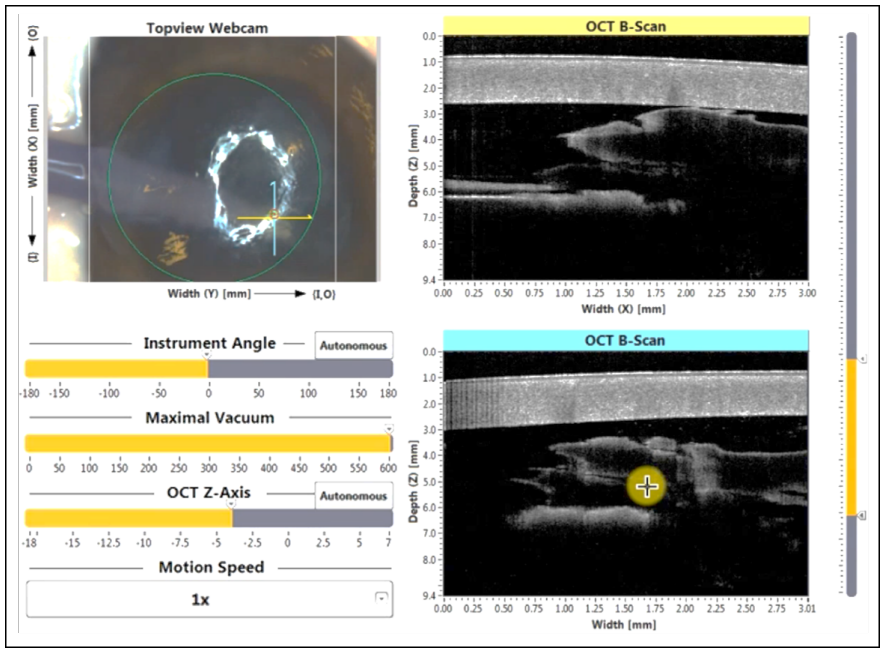


Figure 3.11: The user interface for intraoperative intervention.

$\boldsymbol{p}(t)$ is modified according to the adjusted workspace:

$$\boldsymbol{p}'(t) = \boldsymbol{R}\boldsymbol{S}\left[\boldsymbol{p}(t) - \boldsymbol{c}\right] + \boldsymbol{T} + \boldsymbol{c} \tag{3.7}$$

where $\boldsymbol{R}$ and $\boldsymbol{S}$ are the rotation and scaling matrices representing the workspace deformation, $\boldsymbol{T}$ is the translation vector representing the workspace translation, and the iris center $\boldsymbol{c}$ is the resizing center. This function is critical in the event of eye deformation or corneal collapse because, in such an event, the anatomical structure is different than that at the beginning of the surgery.

(ii) The user can modify the tool-tip velocity or pause the motion while maintaining the aspiration.

(iii) The user can override the rotation angle of the surgical instrument. This function is used to create intraocular turbulence which is useful for detaching larger pieces of lens material from the posterior capsule.

(iv) The user can command the tool tip to any point inside the workspace by directly clicking on the displayed images. The camera view determines the (x, y) coordinates of the target while the z coordinate is assigned from localized OCT B-scans. This function can be used to directly target and remove floating lens material and air bubbles. In the event a user requests a point outside the tool workspace or beyond the safety margins, a bisection algorithm efficiently determines the boundary point closest to the desired target (Algorithm 3.1) without formulating and solving a convex optimization problem. The tool tip will move towards the commanded point and then stop when its distance to the boundary is less than the threshold ϵ.

(v) The user can request emergency termination of the surgery. Once asked, the instrument will cease aspiration and begin to retract from the eye within 200 ms.

Algorithm 3.1 Algorithm for Rerouting Assigned Point

Inputs: Current position $\boldsymbol{p}_0$, assigned position $\boldsymbol{p}_{trg}$

Output: Feasible assigned point $\boldsymbol{p}^*$

Step 1: if $\boldsymbol{p}_{trg}$ is inside workspace, **return** $\boldsymbol{p}^* = \boldsymbol{p}_{trg}$

Step 2: $\Delta\boldsymbol{p} = \boldsymbol{p}_{trg} - \boldsymbol{p}_0$

$(\alpha_{lb}, \alpha_{ub}) = (0, 1)$ and $\boldsymbol{p}^* = \boldsymbol{p}_0$

do

$\boldsymbol{p}_{prev} = \boldsymbol{p}^*$

$\alpha = \frac{1}{2}(\alpha_{ub} - \alpha_{lb})$

$\boldsymbol{p}^* = \boldsymbol{p}_0 + \alpha\Delta\boldsymbol{p}$

if $\boldsymbol{p}^*$ is out of bounds, $\alpha_{ub} = \alpha$

else $\alpha_{lb} = \alpha$

end if

while $(\left\|\boldsymbol{p}^* - \boldsymbol{p}_{prev}\right\|_2 > \epsilon)$ or ($\boldsymbol{p}^*$ is out of bounds)

return $\boldsymbol{p}^*$

3.4.3 Intermittent Volume Scan Assessment

To assess surgical progress, evaluation is performed every two minutes by returning the tool to the standby position, pausing its motion, and acquiring an OCT volume scan (as demonstrated in Fig. 6.4). At this point, the surgeon can study the volume scan and visually inspect the eye. Based on this evaluation, one of three options can be chosen: (1) continuation of the cataract-extraction trajectory, (2) declaration of specific points to target, or (3) termination of the surgery. Option (1) is useful if significant amounts of lens material remain in the anterior segment. Option (2) is useful for extracting small pieces of lens material. Option (3) is applied when no lens material appears in the OCT volume scans and the lens extraction is considered complete.

CHAPTER 4

Learning-Based Precise Motion Control

To ensure surgical safety and improve the efficiency of robotic intraocular surgery, the robot manipulator must precisely track the predefined trajectory as generated in Chapter 3.3. To achieve precise motion control, a novel data-based approach to construct plant inversion filter from learning-type control algorithms, such as Iterative Learning Control (ILC) and repetitive control, is proposed. Once the plant inversion is identified, the inversion filter can be applied to feedforward filtering and high-precision tracking control. Compared to existing inversion methods, the proposed approach eliminates the requirements of accurate dynamic modeling and complicated synthesis methods. It is also shown the proposed method can be applied to MIMO systems, such as robot manipulators, without additional diagonalization or transformation. The design analysis, simulation validation, and experimental results are presented to demonstrate the effectiveness and performance of the proposed method.

4.1 Learning-Type Control Algorithms

A method using learning-type control schemes, such as repetitive control and ILC, to identify the plant inversion within the selected bandwidth is proposed. To start with, both repetitive control and ILC are reviewed in this section followed by a discussion of the role of plant inversion in the learning-type control.

4.1.1 Iterative Learning Control and Repetitive Control

Applying internal model principle, repetitive control is developed for tracking or rejecting periodic signals without steady-state error [INK81]. Considering a SISO plant $G(z)$,

Assumption 4.1 *The controlled discrete-time plant $G(z)$ is asymptotically stable, i.e., all its poles are inside the unity circle.*

A generic repetitive control on $G(z)$ can be expressed as

$$u(k) = L(z)\frac{z^{-N}}{1 - z^{-N}}e(k) \tag{4.1}$$

where N is the period of the target signal, k is the discrete time index, $u(k)$ is the control input to plant $G(z)$, $e(k) = r(k) - y(k)$ is the error signal, $r(k)$ is the reference, and $y(k)$ is the output of $G(z)$. $L(z)$, here referred as the learning filter, stabilizes the closed-loop system. Eq. 4.1 can be further organized as

$$u(k) = u(k - N) + L(z)e(k - N) \tag{4.2}$$

The block diagram of Eq. 4.2 is illustrated in Fig. 4.1.

When N equals the length of the reference $r(k)$, Eq. 4.2 can also be seen as the control updating law in the ILC algorithm, where $0 \leq k \leq N - 1$. Although repetitive control is designed for on-line tracking of a periodic signal and ILC does batch process on a fixed reference trajectory, the similarities of these two control schemes have been investigated [WGD09]. Introducing the iteration index j, Eq. 4.2 can be rewritten into the batch form to represent a generic ILC algorithm,

$$u_j(k) = u_{j-1}(k) + L(z)e_{j-i}(k) \tag{4.3}$$

and the control input $u_j(k)$ at iteration j can be derived as

$$u_1(k) = L(z)r(k) \tag{4.4}$$

$$u_j(k) = [I - (I - L(z)G(z))^j]G^{-1}(z)r(k) \tag{4.5}$$

$$e_j(k) = (I - G(z)L(z))^j r(k) \tag{4.6}$$

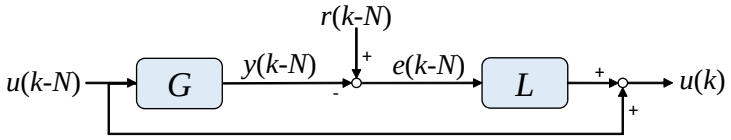


Figure 4.1: The block diagram of learning-type control.

Note that the $G^{-1}(z)$ term in Eq. 4.5 is algebraically canceled after multiplying with the $[I-(I-L(z)G(z))^j]$ term, so Eq. 4.5 holds for both minimum and non-minimum phase systems, while a stable $G^{-1}(z)$ does not exist for the systems with non-minimum phase zeros.

4.1.2 Stability of Learning-Type Control

The stability criteria of the generic ILC algorithm expressed in Eq. 4.3 can be observed from the learning error $e_j(k)$ in Eq. 4.6. [NG02] has proven that Eq. 4.5 and 4.6 are asymptotically stable iff

$$\|I - G(z)L(z)\|_\infty < 1 \tag{4.7}$$

If fact, the repetitive control as described in Eq. 4.1 also shares the same stability criteria. The importance of the plant inversion can be seen here. If the learning filter $L(z)$ is assigned as the perfect plant inversion of $G(z)$, the stability condition of Eq. 4.7 holds and the tracking error in Eq. 4.6 converges within a single iteration. Therefore, a well-approximate plant inversion, especially when $G(z)$ has non-minimum phase zero, is useful on many aspects, including feedforward filtering, repetitive control, and ILC.

Assumption 4.2 *The learning-type control schemes referred in this Chapter, either repetitive control or ILC, are stabilized by a pre-selected learning filter $L(z)$ such that the stability criteria of Eq. 4.7 holds.*

4.1.3 Plant Inversion with Selected Bandwidth

For a plant inversion filter $F(z)$ that perfectly inverts the controlled plant $G(z)$,

$$G(z)F(z) = I \tag{4.8}$$

perfect tracking can then be done by feedforward filtering of the reference $r(k)$,

$$y(k) = G(z)u(k) = G(z)F(z)r(k) = r(k) \tag{4.9}$$

However, perfect inversion implies high-gain feedforward inputs at frequencies beyond the bandwidth of the controlled plant $G(z)$. Beside, model uncertainties are usually proportional to the frequency in reality. As a result, the feedforward inputs must be filtered by a low-pass filter to ensure the control saturation is not triggered.

Instead of cascading a low-pass filter after feedforward filtering, one can relax the perfect inversion (Eq. 4.8) to only invert the plant dynamics within certain bandwidth. In other words, it is to find the plant inversion filter $F(z)$ such that

$$G(z)F(z) = M(z)z^{-d} \tag{4.10}$$

where $M(z)$ is a zero-phase low-pass filter limiting the bandwidth for plant dynamic inversion, and z^{-d} maintains the causality and allows for d steps delay at the output. When $M(z)$ is constructed as an FIR filter, d must be larger than half of its order. Allowing more delay also helps approximate the dynamic of non-minimum phase zeros. When all the frequency components of the reference trajectory $r(k)$ fall in the bandwidth of $M(z)$, i.e., $M(z)r(k) = r(k)$, perfect tracking can be still obtained with d steps delay,

$$\begin{aligned} y(k) &= G(z)u(k) = G(z)F(z)r(k) \\ &= M(z)z^{-d}r(k) = r(k-d) \end{aligned} \tag{4.11}$$

Remark 4.1 (Zero-phase low-pass filter) *The zero-phase low-pass filter $M(z)$ may be created by*

$$M(z) = N(z)N(z^{-1}) \tag{4.12}$$

where $N(z)$ is an FIR low-pass filter (not necessary to be zero-phase) designed with the selected bandwidth.

4.2 Iterative Identification of Plant Inversion

For construction of the plant inversion filter $F(z)$ subject to the selected bandwidth characterized by $M(z)$, as stated in Eq. 4.10, we propose using learning-type control algorithms to do so. The core idea is intuitive: for a LTI system, assigning the impulse response $f(k)$ of the plant inversion filter $F(z)$ as the control input $u(k)$ generates a delta impulse (or a filtered impulse $m(k)$) at the output. On the other words, if we can find the $u(k)$ that generates a *symmetric* impulse at the output, it can then be used to form an FIR feedforward filter. In the following subsections, the algorithm for learning the plant inversion of a SISO system is explained in detail followed by its extension to MIMO systems.

4.2.1 Identification of Plant Inversion on SISO Systems

The identification of plant inversion can be achieved by determining the control input $u(k)$ that generates a delta impulse function at the output. For the cases with the relaxation of inversion bandwidth, i.e., Eq. 4.10, the desired output is changed from the delta impulse to the impulse response of the target filter, $M(z)z^{-d}$. Since learning-type control algorithms feature finding the optimal control input for tracking a specific reference with zero steady-state error (once again, for LTI systems), it perfectly fits the requirement for plant inversion identification.

Starting from an all-band plant inversion that inverts the plant over the entire frequency band from D.C. to the Nyquist frequency, i.e., Eq. 4.8,

Method 4.1 (Learning Plant Inversion Using ILC). For a SISO controlled plant $G(z)$ with an arbitrary stable ILC learning filter $L(z)$ that satisfies Eq. 4.7, the plant inversion $F(z)$ is constructed by the converged input to track a delta impulse with $N-1$ points zero

padding as the reference:

$$\begin{aligned} f(k) &\triangleq \lim_{j\to\infty} u_j(k) \\ &= \lim_{j\to\infty} [I - (I - L(z)G(z))^j] G^{-1}(z)\delta(k) \end{aligned} \tag{4.13}$$

where $f(k)$ is the impulse response of $F(z)$,

$$F(z) = \sum_{k=0}^{N-1} f(k) z^{-k} \tag{4.14}$$

□

Ideally, since the ILC is asymptotically stable and $(I - L(z)G(z))^j$ converges to zero as j increases, a perfect plant inversion can be obtained from Method 4.1. However, this is not the case happening during implementation because (1) $G(z)$ may contain non-minimum phase zeros that make exact plant inversion unstable; (2) learning the delta impulse is too aggressive and beyond the capability of the controlled plant. The control signal may saturate during the learning process; and (3) the updating of high frequency components may be slow or even be shutdown by the ILC algorithm due to stability concerns. Since the performance of the iterative inversion identification on LTI systems depends on how well and how fast the ILC algorithm can learn for tracking the delta impulse, slow ILC convergence for high-frequency components implies significant amount of iterations is required to identify the inversion at the high-frequency region.

To solve the above-mentioned problems, the reference model in Method 4.1 is modified as a smoothed impulse with certain delay steps to compensate for non-minimum phase zeros and to maintain the causality of the reference model. Most importantly, it does not require high gain inputs to generate the filtered impulse at the output. Therefore control saturation is avoided. The generalized method is given as

Method 4.2 (Plant Inversion Construction with Assigned Reference Model). For a SISO controlled plant $G(z)$ with an arbitrary stable ILC learning filter $L(z)$ that satisfies Eq. 4.7, the plant inversion filter $F(z)$ is constructed as

$$F(z) \triangleq H^{-1}(z) U_n(z) \tag{4.15}$$

after the learning error $e_n(k) = r_M(k) - y_n(k)$ converges to a specified tolerance ϵ_l at the n-th iteration, i.e., $\|e_n(k)\|_\infty \leq \epsilon_l$, and $U_n(z)$ is the FIR filter constructed as:

$$U_n(z) = \sum_{k=0}^{N-1} u_n(k) z^{-k} \tag{4.16}$$

where $u_n(k)$ is the learned input at the n-th iteration to generate the output that tracks the impulse response $r_M(k)$ of the reference model $R_M(z)$,

$$R_M(z) = H(z)M(z)z^{-d} \tag{4.17}$$

□

Method 4.2 only needs to be performed one time for every plant, unless the physical plant is changed after learning.

Remark 4.2 (Required number of iterations) *The number of iterations n required for learning the target impulse depends on the learning filter $L(z)$ of the learning-type control. n could be minimized if $L(z)$ approximates the plant inversion. In this Chapter, however, we only assume $L(z)$ satisfies the stability condition as stated in Eq. 4.7. The performance of the embryonic learning-type control, in terms of minimizing n, is not addressed. A way to progressively update the learning filter $L(z)$ will be proposed in Chapter 5.*

Remark 4.3 (Frequency Weight) *$H(z)$ indicates the frequency weight over the reference model, which can be used to prevent control saturation but not affect the identification result. It must be invertible (without any non-minimum phase zero). A typical selection used in this study is an integrator (therefore the ILC is learning for a filtered step function),*

$$H(z) = \frac{1}{1 - z^{-1}} \tag{4.18}$$

$$H^{-1}(z) = 1 - z^{-1} \tag{4.19}$$

Corresponding analysis is discussed in Chapter 4.3.

Theorem 4.1 *The plant inversion filter $F(z)$ constructed from Method 4.2 inverts the controlled plant $G(z)$ over the selected bandwidth specified by $M(z)$ with a bounded error, i.e.,*

$$\left\|M(z)z^{-d} - G(z)F(z)\right\|_\infty \leq \epsilon_l \left\|H^{-1}(z)\right\|_\infty \tag{4.20}$$

Proof 4.1 *Assuming at the n-th iteration of the ILC learning process, the learning error for the smoothed impulse converges below to the specified error boundary ϵ_l,*

$$\|e_n(k)\|_\infty \leq \epsilon_l \tag{4.21}$$

Let $E_n(z)$ be the FIR filter constructed by $e_n(k)$,

$$E_n(z) = \sum_{k=0}^{N-1} e_n(k) z^{-k} \tag{4.22}$$

and ϵ_l is conservatively bounding $\|E_n(z)\|_\infty$

$$\begin{aligned} &\|E_n(z)\|_\infty \\ =&\|[I - G(z)L(z)]^n R_M(z)\|_\infty \\ \leq&\epsilon_l \end{aligned} \tag{4.23}$$

Therefore

$$\begin{aligned} &\left\|M(z)z^{-d} - G(z)F(z)\right\|_\infty \\ =&\left\|M(z) - G(z)F(z)z^d\right\|_\infty \\ =&\left\|M(z) - G(z)H^{-1}(z)U_n(z)z^d\right\|_\infty \\ =&\|M(z) - M(z) + (I - L(z)G(z))^n M(z)\|_\infty \\ =&\left\|E_n(z)H^{-1}(z)z^d\right\|_\infty \\ \leq&\epsilon_l \left\|H^{-1}(z)\right\|_\infty \end{aligned} \tag{4.24}$$

where $U_n(z)$ is the FIR form of the learned input $u_n(k)$ at the n-th iteration. By applying Eq. 4.5,

$$U_n(z) = [I - (I - L(z)G(z))^n]G^{-1}(z)R_M(z) \tag{4.25}$$

□

Theorem 4.1 shows that the plant inversion can be constructed within an arbitrary small error bound ϵ_l. This method can even handle the system with non-minimum phase zeros as long as the number of delay steps d is sufficiently large to cover the dynamics of the unstable zeros. The ILC algorithm autonomously figures out the approximation of the inversion of non-minimum phase zeros. Last but not least, although ILC is used in Method 4.1 and 4.2, it can be replaced by any type of learning control algorithm such as repetitive control which has be shown similar to ILC and also offers zero steady-state tracking error. The only difference is that for ILC, batch process is performed iteratively on the target impulse, while a long, repeating reference connected by multiple impulses is applied when using repetitive control to determine the coefficients of the inversion filter.

4.2.2 Extension to MIMO Systems

Similar to system identification of multi-variable systems, the proposed method can also be applied to a MIMO plant by constructing the inversion filter *channel by channel*. Assuming a learning-type control has been implemented on a MIMO system $G(z)$ (of course, this may not be easy), a multi-variable inversion filter can be constructed by

Method 4.3 (Plant Inversion Construction of MIMO systems). For a $v \times w$ MIMO stable plant $G(z)$, using a learning-type control to track $r_p(k) \in \mathbb{R}^{v\times 1}$, where the p-th channel of $r_p(k)$ is assigned as the impulse response of the SISO reference model $R_M(z) = H(z)M(z)z^{-d}$ and 0 for other channels. For the p-th output channel, let

$$F_p(z) \triangleq diag(H^{-1}(z))U_p(z) \in \mathbb{R}^{w\times 1} \tag{4.26}$$

Note that

$$U_p(z) = \sum_{k=0}^{N-1} u_p(k)z^{-k} \tag{4.27}$$

and $u_p(k) \in \mathbb{R}^{w\times 1}$ is the learned input for generating $r_p(k)$ at the outputs. By repeating the learning process channel by channel (increasing p from 1 to v), the MIMO plant inversion

filter can be constructed as

$$F(z) = \begin{bmatrix} F_1(z) & F_2(z) & \cdots & F_v(z) \end{bmatrix} \tag{4.28}$$

□

An example is given as the following. Considering a 2×2 MIMO plant $G(z)$ ($v = w = 2$), a reference model $R_M(z)$ is selected with a unity frequency weight, i.e., $H(z) = 1$. The impulse response of the SISO reference model is represented as $r_M(k) = m(k-d)$. For the first output channel ($p = 1$), $r_1(k)$ is assigned as $[r_M(k); 0;]$. After the tracking error for $r_1(k)$ converges, the learned input $u_1(k)$ is used to construct $F_1(z)$ as shown in Eq. 4.26. Repeat the same process for the second output channel ($p = 2$) with $r_2(k) = [0; r_M(k);]$ and obtain $F_2(z)$. The MIMO plant inversion filter of $G(z)$ is then constructed as $F(z) = \begin{bmatrix} F_1(z) & F_2(z) \end{bmatrix}$.

Theorem 4.2 *The feedforward filter $F(z)$ constructed from Method 4.3 inverts the MIMO controlled plant $G(z)$ over the selected bandwidth specified by $M(z)$, i.e.,*

$$G(z)F(z) \to diag(M(z)z^{-d}) \tag{4.29}$$

Proof 4.2 *For the p-th output channel,*

$$\begin{aligned} & G(z)F_p(z) \\ =& G(z)diag(H^{-1}(z))U_p(z) \\ \to& diag(H^{-1}(z)) \sum_{k=0}^{N-1} r_p(k)z^{-k} \\ =& \begin{cases} M(z)z^{-d}, & p^{th}\ channel \\ 0, & others \end{cases} \end{aligned} \tag{4.30}$$

because the tracking error for $r_p(k)$ converges to zero and

$$G(z)U_p(z) \to \sum_{k=0}^{N-1} r_p(k)z^{-k} \tag{4.31}$$

Therefore,

$$G(z)F(z) = \begin{bmatrix} G(z)F_1(z) & G(z)F_2(z) & \cdots & G(z)F_v(z) \end{bmatrix} \rightarrow diag(M(z)z^{-d}) \tag{4.32}$$

□

Note that it is also possible to assign different bandwidth to each channel. In this case, the reference model $R_M(z)$ is a diagonal $v \times v$ MIMO transfer function. The impulse response of the (p, p) element of $R_M(z)$ is then assigned as the target reference $r_p(k)$.

4.3 Design Analysis and Simulation Validation

The identification of plant inversion filter can be summarized as follows: (1) decide the reference model, (2) assign the impulse response of the reference model as the desired trajectory to a learning-type control algorithm, (3) update iteratively the control input until the tracking error of the impulse converges below the specified error bound, and (4) create an FIR plant inversion filter using the learned input as the FIR coefficients (with necessary filtering if a frequency weighting filter is added in the reference model). Couple design factors have raised, such as the influence of the order of the FIR filter, the determination of bandwidth, and the selection of frequency weight. These critical design factors are discussed in the following subsections accompanying with simulation validation on a SISO linear motor model.

4.3.1 Selection of Bandwidth and Frequency Weights

One of the most important issues in the proposed method is how to decide the reference model $R_M(z)$, namely the bandwidth represented by $M(z)$, the frequency weight $H(z)$, and the number of delay steps d. For the purpose of selecting a proper reference model, two things need to be considered. First, the energy of the impulse response should be as large as possible for a better signal-noise-ratio (SNR), but there is an upper bound for the energy to

prevent the control inputs from saturation. Second, $M(z)$ that specifies the bandwidth of the plant inversion filter should provide enough bandwidth for the desired feedforward tracking performance. Intuitively, higher bandwidth is also more prone to control saturation.

H_2 system norm is used to predict the peak value of the input signal and check if the assigned reference model is acceptable without triggering control saturation. Assuming the learning-type control *monotonically* converges to zero steady-state error on the controlled SISO plant $G(z)$, both the peak value of the control input at the first iteration ($u_1(k) = L(z)r_M(k)$) and after converged ($G(z)u_\infty(k) = r_M(k)$) must be less than the control saturation limit $\bar{u}$. Specifically, the following inequality must hold:

$$\|m(k)\|_2 \leqslant min(\left\|H^{-1}(z)G(z)\right\|_2, \frac{1}{\|H(z)L(z)\|_2}) \cdot \bar{u} \tag{4.33}$$

where the H_2 system norm for an arbitrary SISO plant model $G(z)$ with its state-space expression (A_G, B_G, C_G, D_G) is defined as

$$\begin{aligned} \|G(z)\|_2 &\equiv \sup_{u \in L_2} \{\|y\|_\infty \mid \|u\|_2 \leqslant 1\} \\ &= C_G X C_G^T,\ A_G X A_G^T - X + B_G B_G^T = 0 \end{aligned} \tag{4.34}$$

where X is the controllability Gramian.

Aside from control saturation, another factor that affects the selection of the bandwidth of $M(z)$ is the performance of the learning-type control algorithm. If the updating of the control law is filtered by a low-pass filter, $M(z)$ should also limit its bandwidth corresponding to what the *embryonic* learning-type control can do (or it would learn nothing beyond the cut-off frequency). Once $M(z)$ is determined after considering control saturation and the bandwidth of the learning-type control algorithm, the number of delay steps d is decided to maintain the causality. In the implementation, we select an arbitrary large d such that the reference model is causal and the dynamic of non-minimum phase zeros can be well-approximated.

The frequency weight $H(z)$, on the other hand, does not affect the performance if there is no control saturation. As long as the same number of iterations is applied to learn the

impulse response of the reference model, with or without $H(z)$ creates an identical filter. As an optional design factor, $H(z)$ can be added in order to perform more iterations before the learning-type control algorithm hits the control saturation boundary. This study applies an integrator (Eq. 4.19) as the frequency weight, but it is not required for every controlled plant.

4.3.2 Analysis of Truncation Error

The other important design factor is the length N of the target impulse response, or equivalently the order of the constructed FIR plant inversion filter. The proposed method can be seen as a special case of polynomial basis function approaches [MTB08, WB10, HS01, FLM17] as we are using a delta/filtered impulse to form the basis functions. However, the performance shown in [MTB08, WB10, HS01, FLM17] is limited by the chosen basis functions where a relatively small N is applied. Ideally, the length of the target impulse response should be sufficiently large to cover the plant dynamics. With sufficiently large N, the performance of the proposed FIR plant inversion filter is comparable with the rational basis function approaches where an IIR plant inversion filter is constructed.

A typical way to determine the order N of the FIR inversion filter $F(z)$ is by observing the impulse response of the controlled plant $G(z)$. In fact, we can also guess the bandwidth of $M(z)$ based on the experimental impulse response of $G(z)$. If, for example, the impulse response of $G(z)$ takes 10 ms to settle down, then it is required to select an N larger than 10 ms and design a similar "smoothness" of the impulse response for $M(z)$. When N is not selected large enough, significant truncation error (when using ILC to learn the target impulse after zero padding) or time-domain aliasing (when using repetitive control) would deteriorate the performance of the plant inversion filter.

Theorem 4.3 *The error of the truncated FIR plant inversion filter $F_c(z)$ is bounded by the truncation threshold ϵ_t times twice of the H_∞ gain of the controlled plant, i.e.,*

$$\left\| M(z)z^{-d} - G(z)F_c(z) \right\|_\infty \leq 2\epsilon_t \left\| G(z) \right\|_\infty \tag{4.35}$$

Proof 4.3 *With the truncation threshold ϵ_t, let $f_c(k)$ be the truncated impulse response of the inversion filter. $f_p(k)$ and $f_f(k)$ are the discarded signals at left and right side, respectively.*

$$f_p(k) = \begin{cases} f(k), & \forall k < k_1,\ |f(k)| \leq |f(k_1)| \leq \epsilon_t \\ 0, & else \end{cases} \tag{4.36}$$

$$f_f(k) = \begin{cases} f(k), & \forall k \geq k_2,\ |f(k)| \leq |f(k_2)| \leq \epsilon_t \\ 0, & else \end{cases} \tag{4.37}$$

$$f_c(k) = \begin{cases} f(k), & \forall k,\ k_1 \leq k < k_2 \\ 0, & else \end{cases} \tag{4.38}$$

$$f(k) = f_p(k) + f_c(k) + f_f(k) \tag{4.39}$$

Therefore,

$$\begin{aligned} &\left\| M(z)z^{-d} - G(z)F_c(z) \right\|_\infty \\ =&\left\| M(z)z^{-d} - G(z)[F(z) - F_p(z) - F_f(z)] \right\|_\infty \\ =&\left\| M(z) - M(z) + G(z)[F_p(z) + F_f(z)]z^d \right\|_\infty \\ \leq& 2\epsilon_t \left\| G(z) \right\|_\infty \end{aligned} \tag{4.40}$$

where $F(z)$ is assumed to yield $G(z)F(z) = M(z)z^{-d}$. □

4.3.3 Simulation Validation

The proposed method is validated on a SISO closed-loop plant model $G(z)$ of a linear motor [TT15, CCT17],

$$G(z) = \frac{-0.02(z - 1.664)(z - 0.648)(z + 0.036)}{(z^2 - 1.804z + 0.835)(z^2 - 1.599z + 0.764)} \tag{4.41}$$

with sampling rate 10 kHz (sampling period $T_s = 0.1$ ms). Note that there is an unstable zero locating at 1.664. This model is only used to emulate the system dynamics in the following simulations. No plant model is required (although it is optional to apply a model-based ILC to learn the target impulse) for constructing the plant inversion filter.

The time-reversal based ILC [BO15a] is applied here as the learning-type control algorithm to demonstrate the iterative identification of the plant inversion. By filtering the

reversed error signal $r_M(N-k) - y(N-k)$ using the same controlled plant $G(z)$, the ILC algorithm can be stabilized by selecting a sufficiently small learning gain, which is set as 0.8 in the following simulations. Subject to the bandwidth of $G(z)$, the learning gain of the time-reversal ILC at high-frequency regions is almost negligible. Therefore, it takes significant amount of iterations to learn the high-frequency components of the desired reference trajectory. Alternatively, applying model-based ILC that uses an approximated inversion as the learning filter could improve the convergence rate. However, to avoid falling into the *chicken-and-egg* situation, the convergence rate is compromised by applying the model-less time-reversal ILC algorithm.

The simulation result shown in Fig. 4.2 and 4.3 uses the delta impulse as the reference model, i.e., $M(z) = 1$ and therefore $m(k) = \delta(k) = 1/T_s = 10^4$ (1 mm physical displacement) when $k = 0$; $m(k) = 0$ elsewhere. To account for the dynamic of the non-minimum phase zero, the number of delay step d is set as 25 ms and the order N of the FIR plant inversion filter is conservatively selected as 50 ms (500 points under 10 kHz control sampling rate). The performance with and without adding the frequency weight, here demonstrated by an integrator as shown in Eq. 4.19, is also compared. For the ideal simulation that does not take control saturation into account, both cases achieve nearly perfect plant inversion under 1 kHz after 10^4 ILC iterations. There is no significant difference between the case with (grey line) and without (black line) adding the frequency weight. If the ILC iteration continues after 10^4 iterations, the error of the constructed inversion will be further reduced until it reaches the level of numerical error (-250 dB). Note that totally 10^4 ILC iterations is performed because the time-reversal ILC updates high-frequency components very slowly. Much less iterations are required for constructing the plant inversion under 500 Hz, where the -3 dB cut-off frequency of the plant $G(z)$ locates.

The simulation of the ideal cases is not suitable for implementation due to control saturation. The simulation results with the consideration of control command saturation ($\bar{u} = 10^5$, i.e. ± 10 mm physical traveling range) are shown as dot lines in Fig. 4.2 and 4.3. For the case without the frequency weight, the ILC convergence saturates after 1500 iterations while

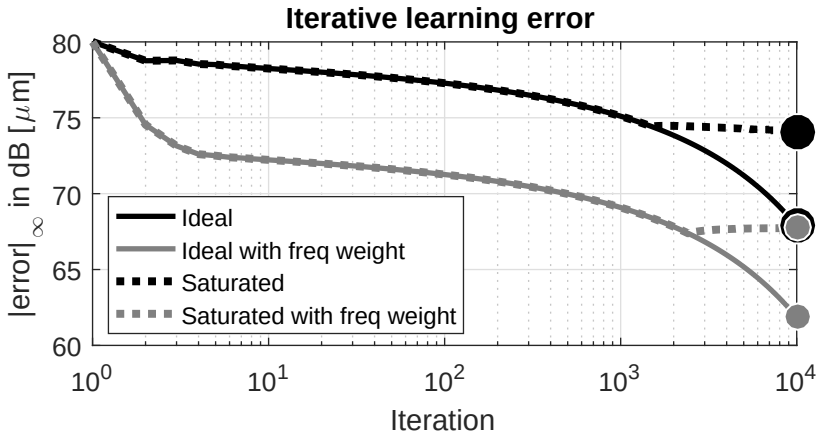


Figure 4.2: Simulation results of learning the delta impulse ($M(z) = 1$, $d = 0$) for the plant inversion filter – the convergence of the learning error.

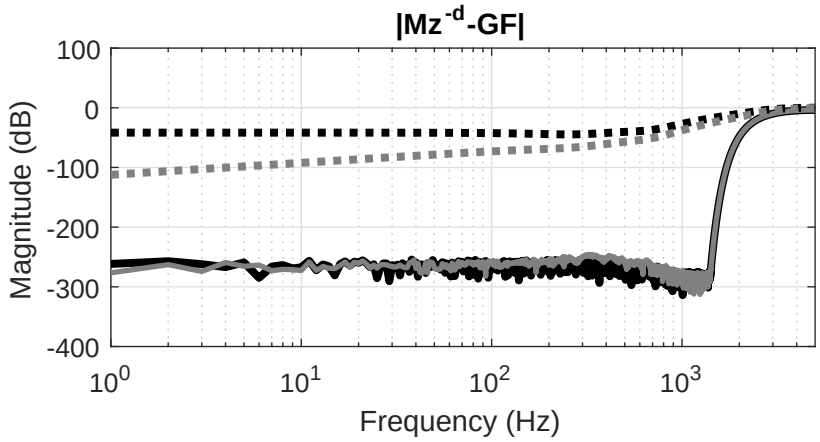


Figure 4.3: Simulation results of learning the delta impulse ($M(z) = 1$, $d = 0$) for the plant inversion filter – the error spectrum of the identified inversion.

2400 iterations for the case using an integrator as the frequency weight. From Fig. 4.3 one can see the benefit of applying the frequency weight. Since 900 more ILC iterations are performed before saturated, the constructed plant inversion filter after adding the frequency weight outperforms the one directly learns the delta impulse.

To avoid control saturation, $M(z)$ is designed to have 300 Hz bandwidth by applying FIR filter design techniques (see Fig. 4.4 for its bode diagram). The performance of constructing

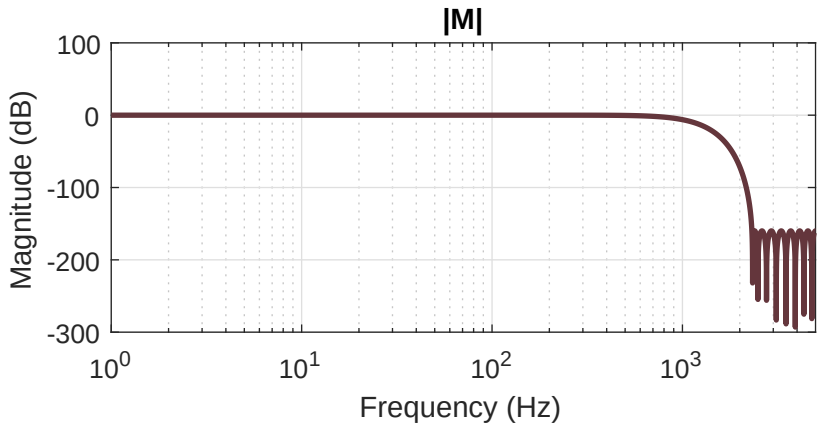


Figure 4.4: Simulation results of learning a filtered step function for the plant inversion – the Bode plot of $M(z)$.

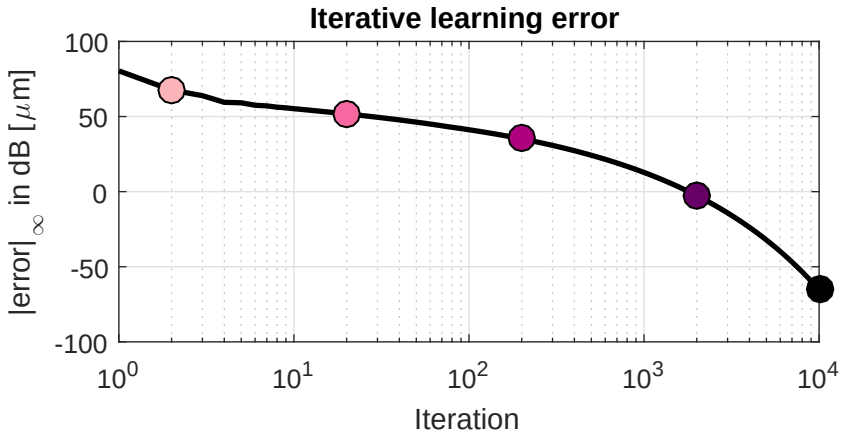


Figure 4.5: Simulation results of learning a filtered step function for the plant inversion – the convergence of the learning error.

the plant inversion filter $F(z)$ by learning a 500 Hz bandwidth smoothed step function is shown in Fig. 4.5. The error upper bounds estimated from Theorem 4.1 are also plotted as dash lines (Fig. 4.6). Similar to the case learning $M(z) = 1$, the error of the constructed plant inversion is reduced and bounded by the learning error as the iteration number increases. After only 20 iterations of the iterative identification, the plant inversion is identified, with the error less than -90 dB, up to 400 Hz. Since the plant inversion at high frequency is

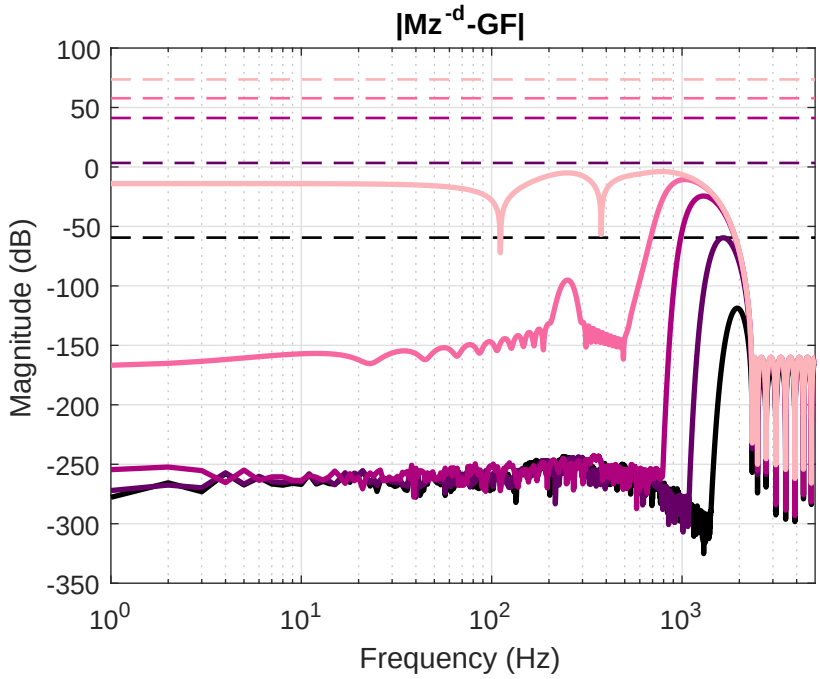


Figure 4.6: Simulation results of learning a filtered step function for the plant inversion: – the error spectrum of the identified inversion. Dash lines indicate the estimated error bounds based on the learning error shown in Fig. 4.5.

inheritedly shutdown by the 300 Hz bandwidth $M(z)$, the control input is never saturated during the learning process.

Last but not least, the effect of truncating the order N of the FIR plant inversion filter is simulated. The truncated impulse response and the error spectrum of the identified inversion is shown in Fig. 4.7 and 4.8, where the ILC is performed 20 iterations before constructing the inversion filter. This simulation result suggests that N should be sufficiently large for a better performance; it also shows the limitation of the methods proposed in [MTB08, WB10, HS01, FLM17] where a small N is usually applied. From the error spectrum, one can see the accuracy of the inversion is significantly deteriorated after truncating N from 50 ms to 4 ms (500 to 40 points). One way to improve the performance is to multiply a windowing function at the beginning and the end, such that there is no abrupt changes induces by the rectangle truncation window. This improvement can be observed at the low-frequency bands

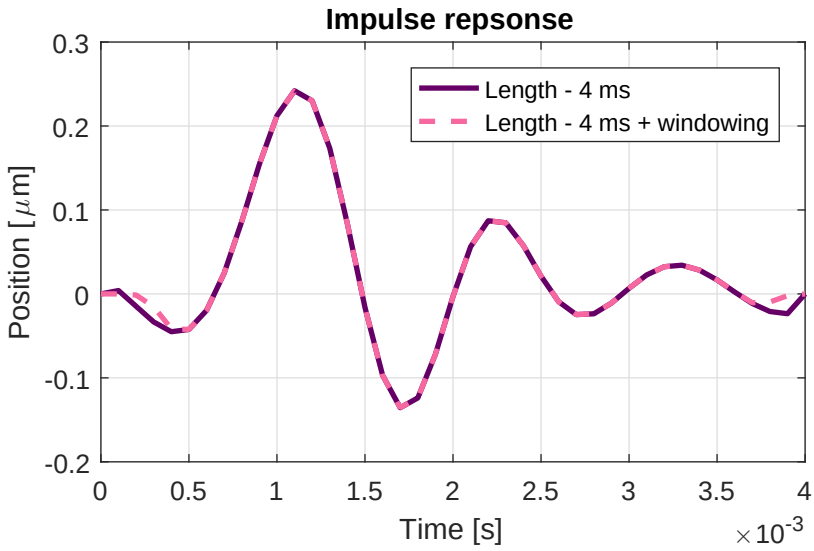


Figure 4.7: Simulation results when the FIR length is not sufficiently large – the impulsed response of the identified inversion.

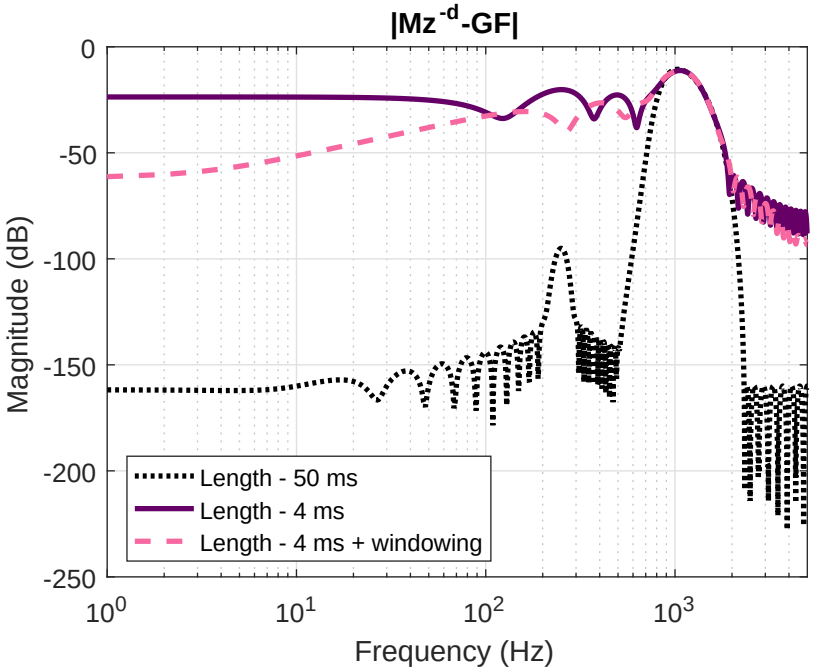


Figure 4.8: Simulation results when the FIR length is not sufficiently large – the error spectrum of the identified inversion.

in Fig. 4.8. Of course, whether multiplying a windowing function or not would not affect the performance when N is sufficiently large.

4.4 Experimental Results

In this section, the proposed method for iterative plant inversion identification and feedforward filtering was implemented on a SISO linear motor, a MIMO active magnetic bearing spindle system, and the IRISS robot manipulator (a decoupled MIMO system). Feedforward tracking was performed to evaluate the performance of the constructed plant inversion filters.

4.4.1 SISO System: Linear Motor

The proposed method was first implemented on the SISO linear motor used in the simulation (see Chapter 4.3). This linear motor was designed for a fast tool servoing system with measurement quantization of 0.1 μm, and actuated by a current amplifier. The control algorithm was implemented on a dual-core Labview real-time target with 10 kHz control sampling rate. The control diagram for the iterative identification process is illustrated in Fig. 4.9, where the learning filter $L(z)$ was implemented by the model-based zero-phase-error tracking controller (ZPETC) [Tom87, TT87, KT14] for increasing the convergence rate. As

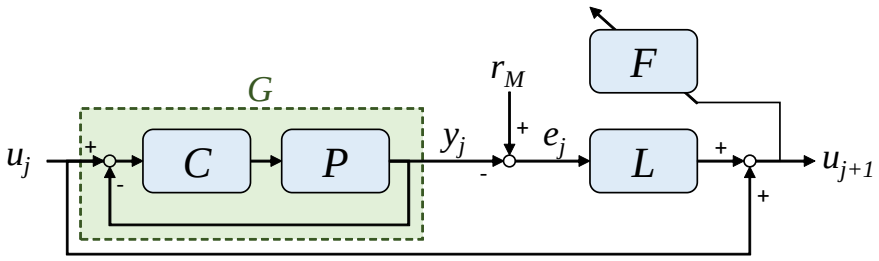


Figure 4.9: The control diagram of iterative identification of the plant inversion for the linear motor system.

for the reference model, a zero-phase low-pass filter with bandwidth around 300 Hz was used,

$$M(z) = (\frac{z^{-1} + 2.1 + z}{4.1})^q \tag{4.42}$$

with q was set as 40, d and N was 8 and 50 ms for preserving the causality, respectively; $H(z)$ was assigned as an integrator as shown in Eq. 4.19 with the step size of 100 μm.

Assigning $r_M(k)$, the impulse response of the reference model $R_M(z) = H(z)M(z)z^{-d}$, as the reference to the ILC algorithm, the plant inversion filter $F(z)$ was constructed after the ILC converged. The error convergence is shown in Fig. 4.10. The ILC algorithm converged after 9 iterations with RMS error of 0.41 μm. The converged output and the corresponding input for tracking $r_M(k)$ is shown in Fig. 4.11. After performing normalization and discrete-time differentiation on the converged control input as shown in Fig. 4.11, the plant inversion filter was constructed as a FIR filter in length of $N = 500$, as shown in Fig. 4.12.

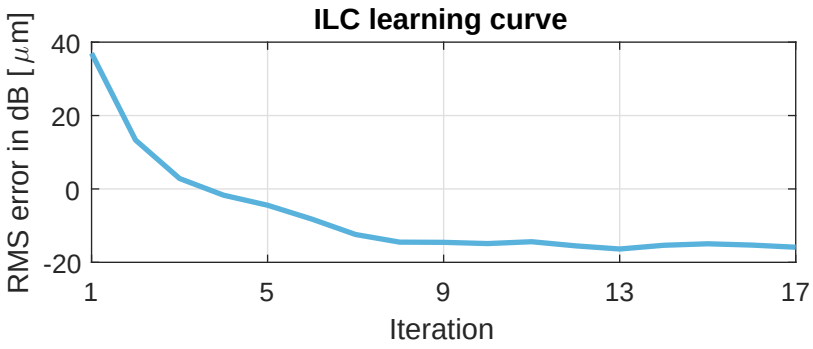


Figure 4.10: Error convergence of iterative learning the target step on the linear motor.

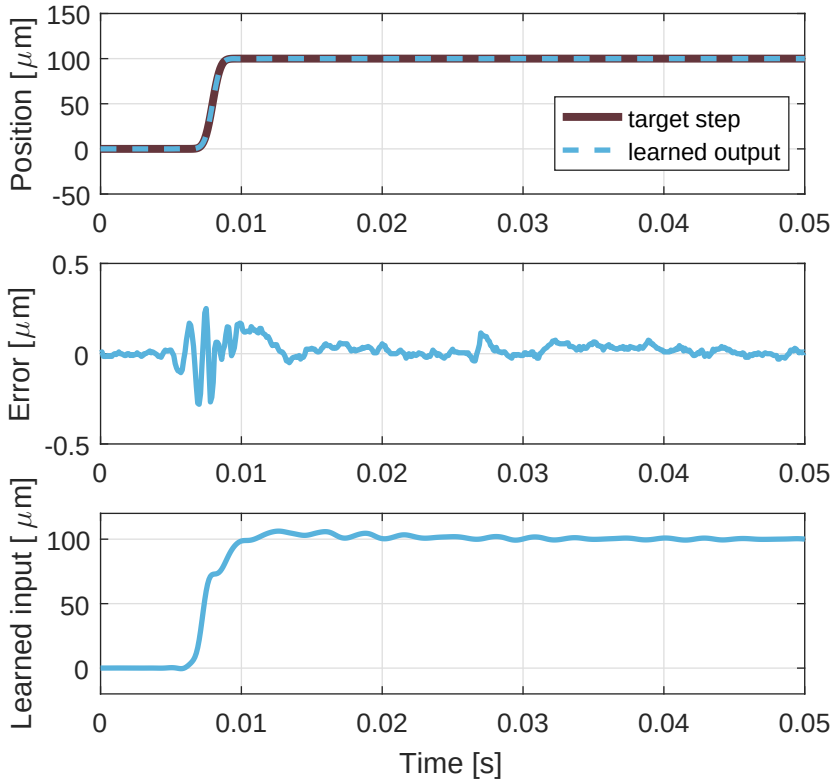


Figure 4.11: Converged results of tracking the target step ($d = 8$ ms) on the linear motor.

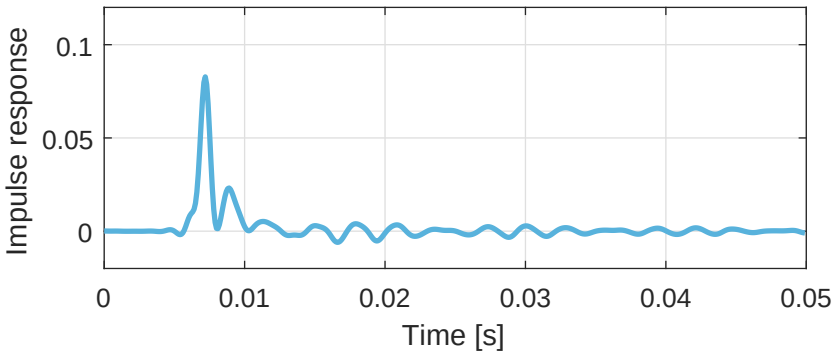


Figure 4.12: Impulse response of the identified plant inversion filter for the linear motor.

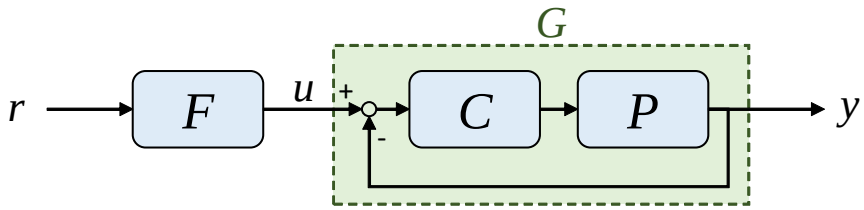


Figure 4.13: The control diagram of feedforward tracking on the linear motor system.

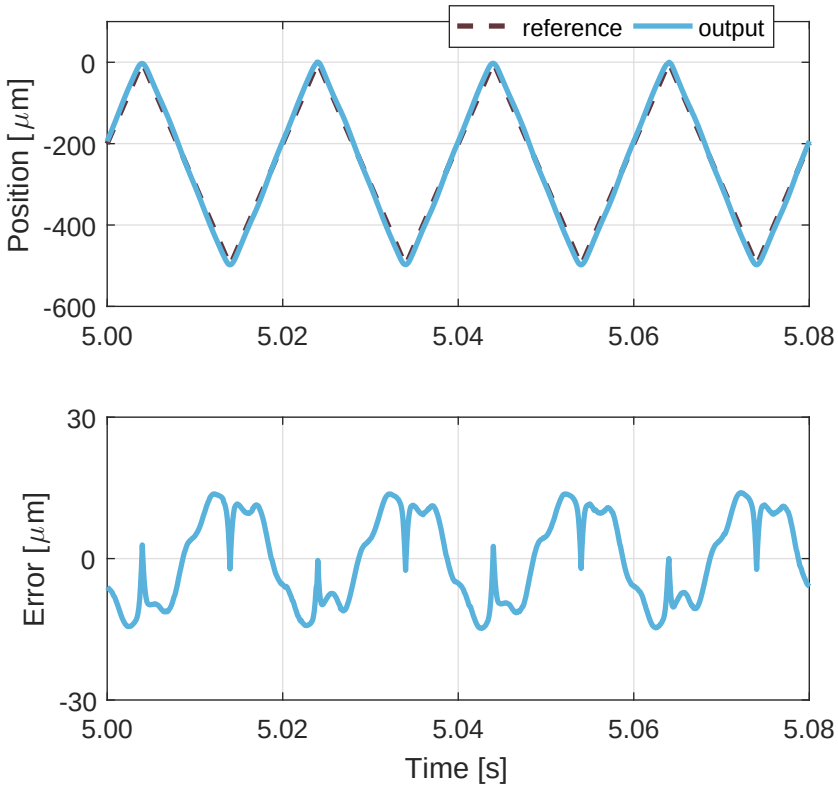


Figure 4.14: Feedforward tracking of a 50 Hz triangular wave on the linear motor using the identified inversion filter.

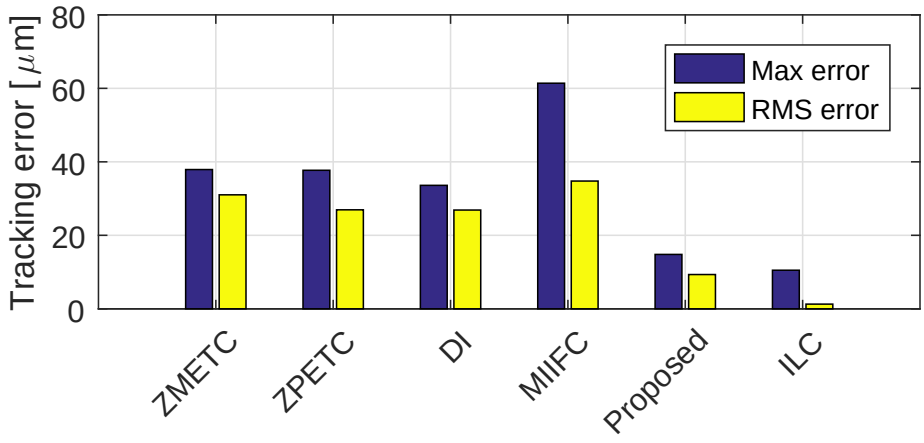


Figure 4.15: Comparison of feedforward tracking of the 50 Hz triangular wave on the linear motor using different methods. Acronyms are as defined in Chapter 1.2.

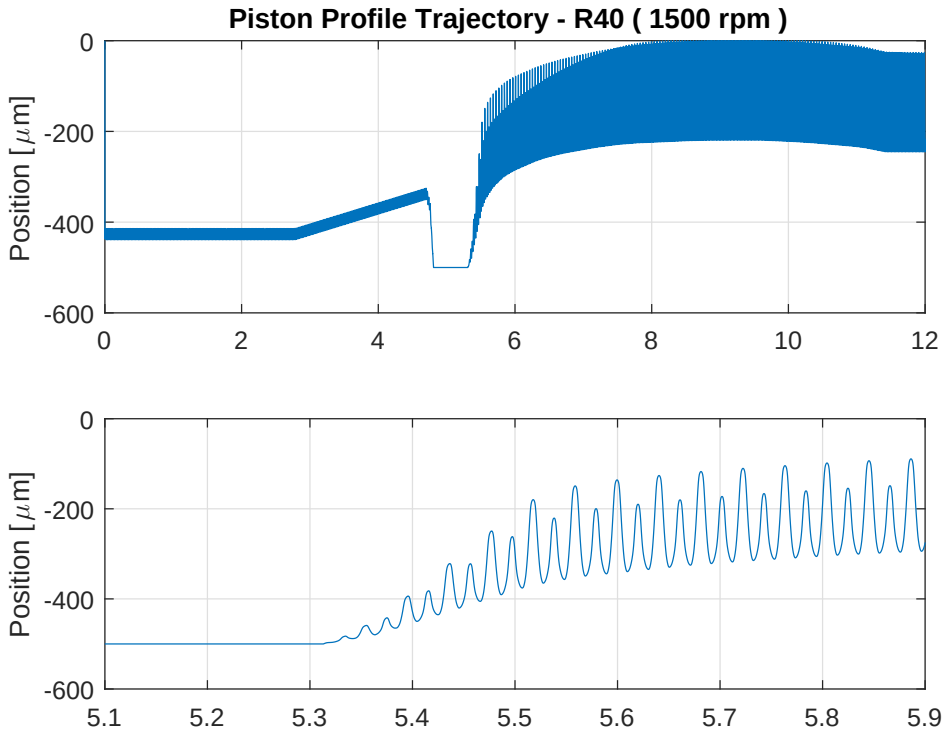


Figure 4.16: Reference trajectory for engine piston manufacturing

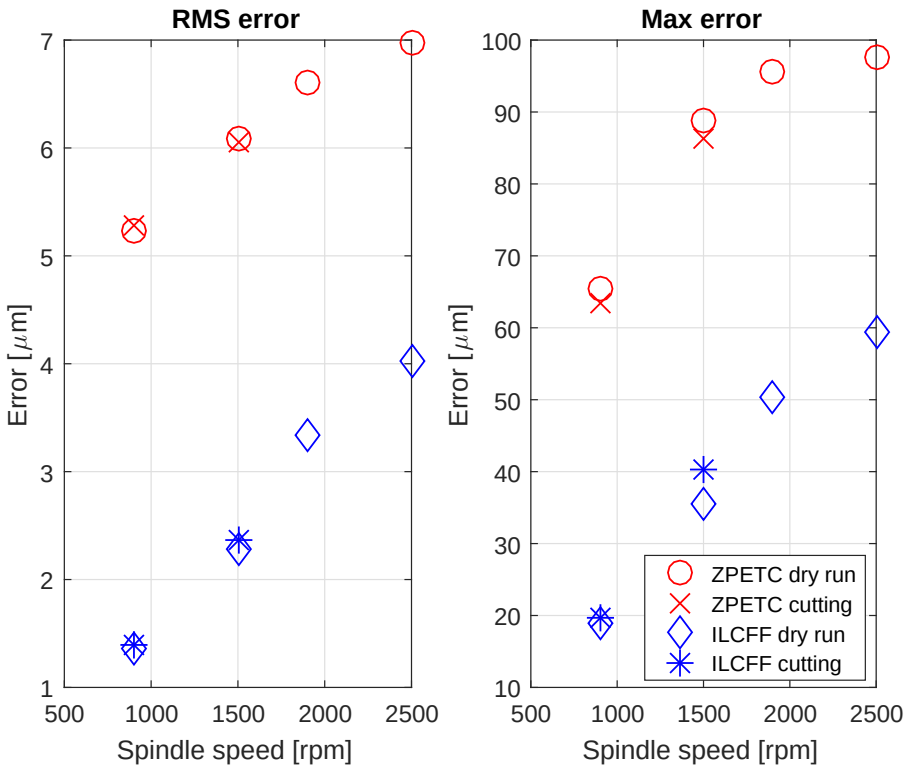


Figure 4.17: Comparison between the proposed approach and the model-based approach (ZPETC) on tracking the reference trajectory as shown in Fig 4.16.

The constructed plant inversion filter was then used as a feedforward controller to track a 50 Hz triangular wave with the amplitude of 250 μm. The control diagram for feedforwad tracking is illustrated in Fig. 4.13. The tracking performance is shown in Figure 4.14, where the RMS error is 9.33 μm. The identified inversion was very accurate and eliminated the phase error. The tracking error at DC was almost eliminated. As expected, the inversion filter was less precise for the high-frequency components due to non-linearity of the system such as friction force and sensor quantization. The comparison of the tracking error by using different inversion methods is illustrated in Fig. 4.15. The "best" tracking result we could obtain by applying the ILC algorithm to track the triangular wave is also shown. The proposed approach outperformed both model-based and data-based approaches. The tracking performance done by the identified inversion filter was comparable with the one

achieved by the ILC, without any training process taken on this specific reference trajectory. The proposed method could be seen as an alternative approach for the applications where iterative learning process is not suitable.

In addition to the tracking of a 50 Hz triangular wave, the comparison between the proposed method and the model-based approach (ZEPTC) is also performed on the linear motor for the application of engine piston manufacturing. Both dry run and actual cutting cases are performed with various spindle speeds which determine the fundamental frequency of the reference trajectory. An example trajectory (R40) sampled at spindle speed of 1500 rpm is illustrated in Fig. 4.16. When the spindle speed increased, the tracking error became larger because the linear motor must move faster to track higher frequency components. However, the proposed method still outperformed the model-based approach on both dry run and actual cutting in all the cases of spindle speed (Fig. 4.17).

4.4.2 Coupled MIMO System: Active Magnetic Bearing System

The proposed method is capable of constructing the plant inversion filter for MIMO systems. This was demonstrated on a 4×4 MIMO active magnetic bearing spindle system that was applied to high speed machining [RCS16], as illustrated in Fig. 4.18. An LQGi controller $C(z)$ implemented at 10 kHz control smapling was used to stabilize the open-loop plant. In this experiment, we applied MIMO repetitive control to identify the plant inversion filter. Because the analog gap sensors of each axes were not co-located with those of the actuators, the strong off-diagonal coupling appeared and made it difficult to identify and control the system. The details of modeling and repetitive control design of the magnetic bearing spindle system can be found in [RCS16].

For the reference model, a zero-phase low-pass filter with bandwidth around 300 Hz was again used, but the transition band of this $M(z)$ filter was minimized through designing an 80 points equiripple FIR filter. The number of delay steps d and the order N of the FIR inversion filter (which was also the delayed time period set in the repetitive control) was set

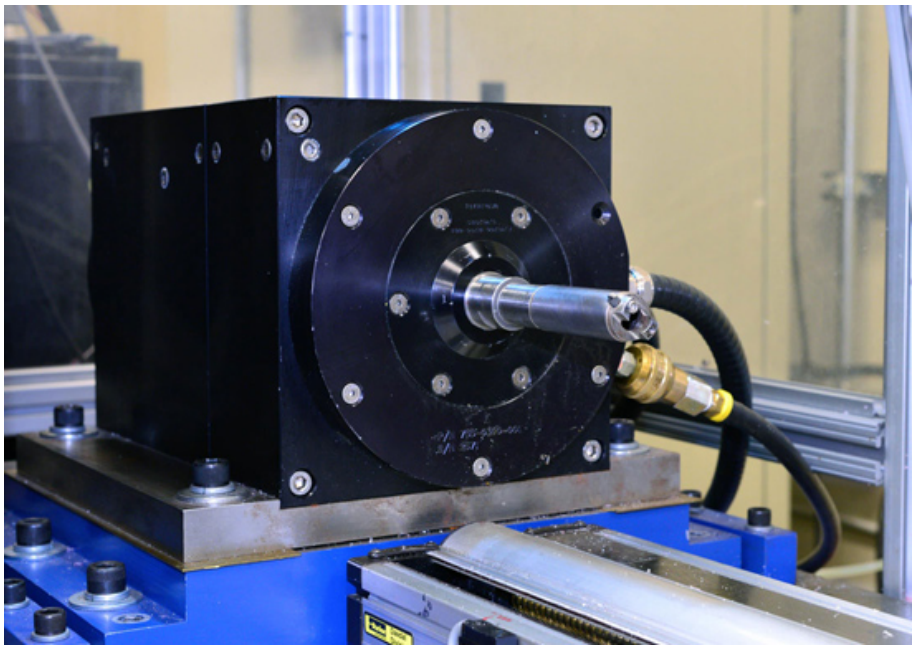


Figure 4.18: The active magnetic bearing spindle system.

as 4 and 20 ms, respectively; $H(z)$ was assigned as an integrator as shown in Eq. 4.19 with the step size of 25 μm. The impulse response of the reference model $R_M(z) = H(z)M(z)z^{-d}$ along with the converged learning results are shown in Fig. 4.19. Note that Fig. 4.19 only shows the result for the first output channel ($p = 1$). As proposed in Chapter 4.2, the same process must be repeated for the rest of the output channels. The impulse responses of the constructed MIMO plant inversion are illustrated in Fig. 4.20. As expected, there are coupling terms between the first and third channel, and so as the second and forth channel.

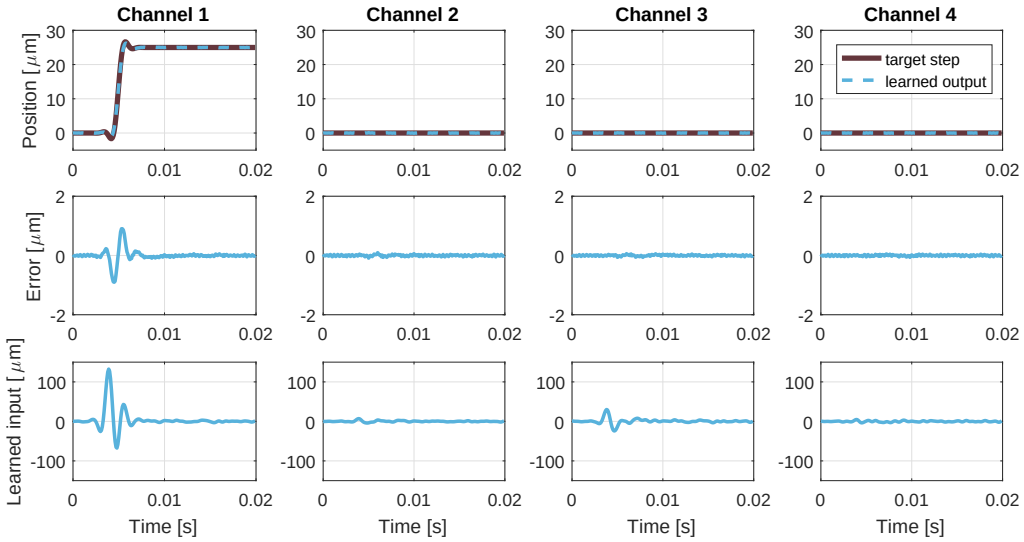


Figure 4.19: Identification of the plant inversion of the MIMO magnetic bearing spindle system – the repetitive control tracks the target step at channel 1.

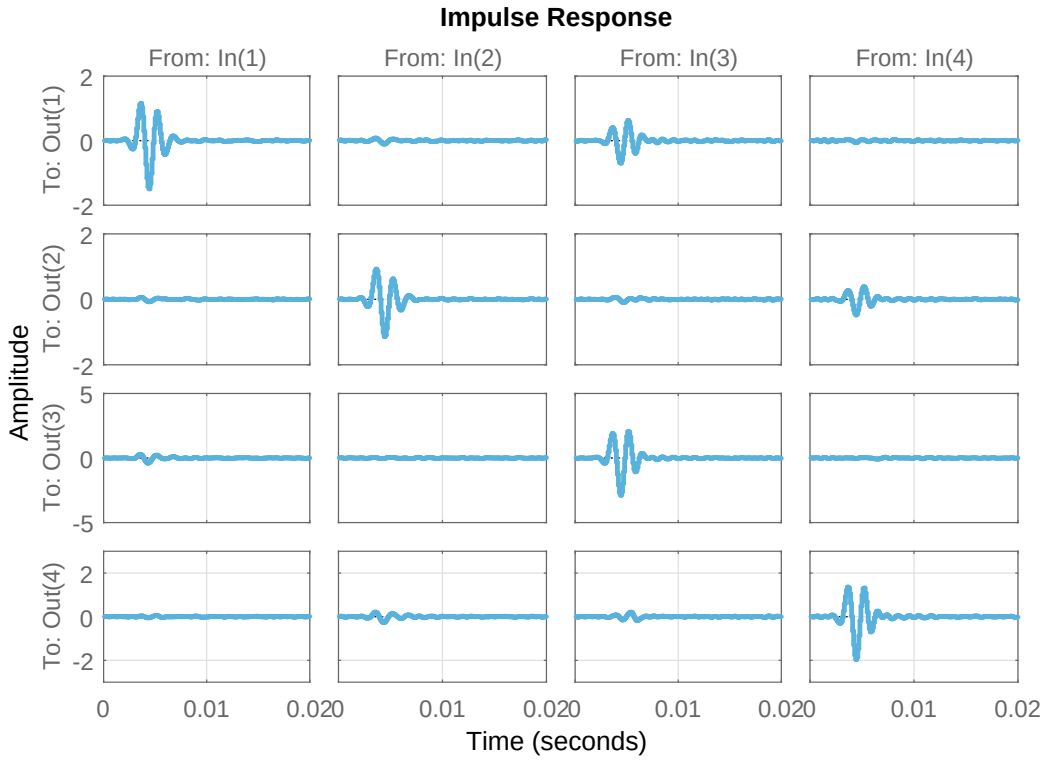


Figure 4.20: Impulse responses of the identified plant inversion filter for the magnetic bearing spindle system.

Similar to the SISO linear motor experiment, the constructed plant inversion filter was evaluated by feedforward tracking. The purpose of identifying the plant inversion of the magnetic bearing spindle system was using feedforwad action to account for non-periodic references, which were demonstrated here by tracking a combination of sinusoidal signals:

$$r_c(t) = \sum_{i=1}^{3} [1 - 0.1(c-1)] A_i sin(2\pi f_i t) \tag{4.43}$$

where c represents c-th channel, $1 \leq c \leq 4$, f_1 was chosen as 40 Hz; f_2 and f_3 was set as 70 and 170 Hz, respectively. The amplitude A_i was selected as 12, 8, and 4 μm for each frequency, respectively.

The feedforward tracking result is shown in Fig. 4.21, where the references were filtered by the constructed inversion filter starting from the first second of tracking. Once the feedforward filtering was turned on, the phase error caused by the LQGi controller was compensated for, and therefore, the tracking error was significantly reduced. As shown in the zoomed-in portion of channel 3 (Fig. 4.22) and the error spectrum (Fig. 4.23), the references were non-periodic because their frequency components were not integral times to each other. Traditional feedback controllers such as repetitive controllers are not able to handle this kind of reference, therefore a feedforward controller is required to improve the tracking performance by eliminating phase error and increasing the tracking bandwidth. The RMS error before and after applying the identified inversion filter is listed in Table 4.1, where the performance was improved by at least 77.4% after feedforward filtering.

Table 4.1: The RMS tracking error on the magnetic bearing spindle system before and after feedforward filtering. Unit in μm.

Channel	**1**	**2**	**3**	**4**
without feedforward	8.62	7.66	9.38	5.82
with feedforward	1.95	1.71	1.27	0.72

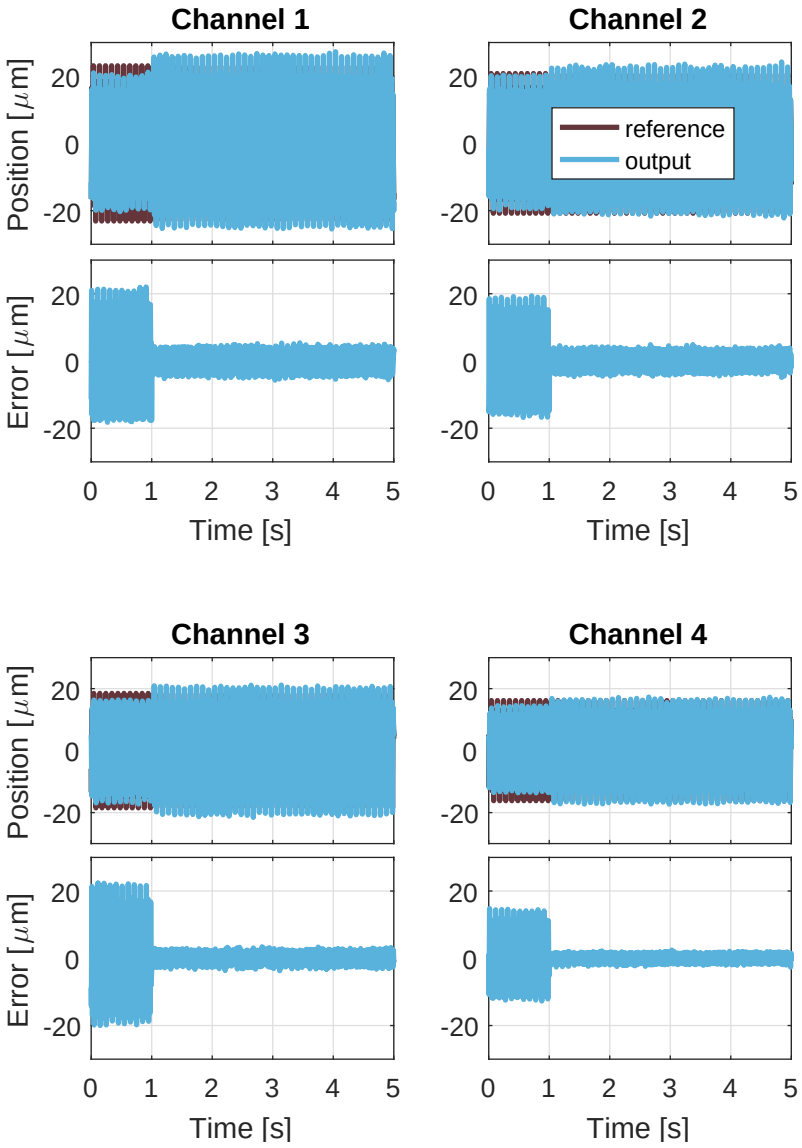


Figure 4.21: Feedforward filtering by using the identified plant inversion. The feedforward action was activated after the first second. See Fig. 4.22 for the zoomed-in portion at the third channel.

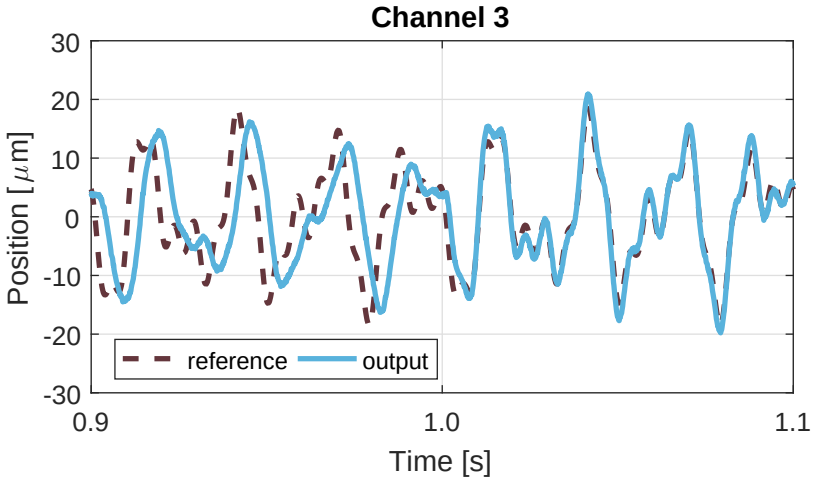


Figure 4.22: MIMO feedforward filtering on the magnetic bearing spindle system by using the identified plant inversion (zoomed in at the third output channel). The feedforward action was activated after the first second.

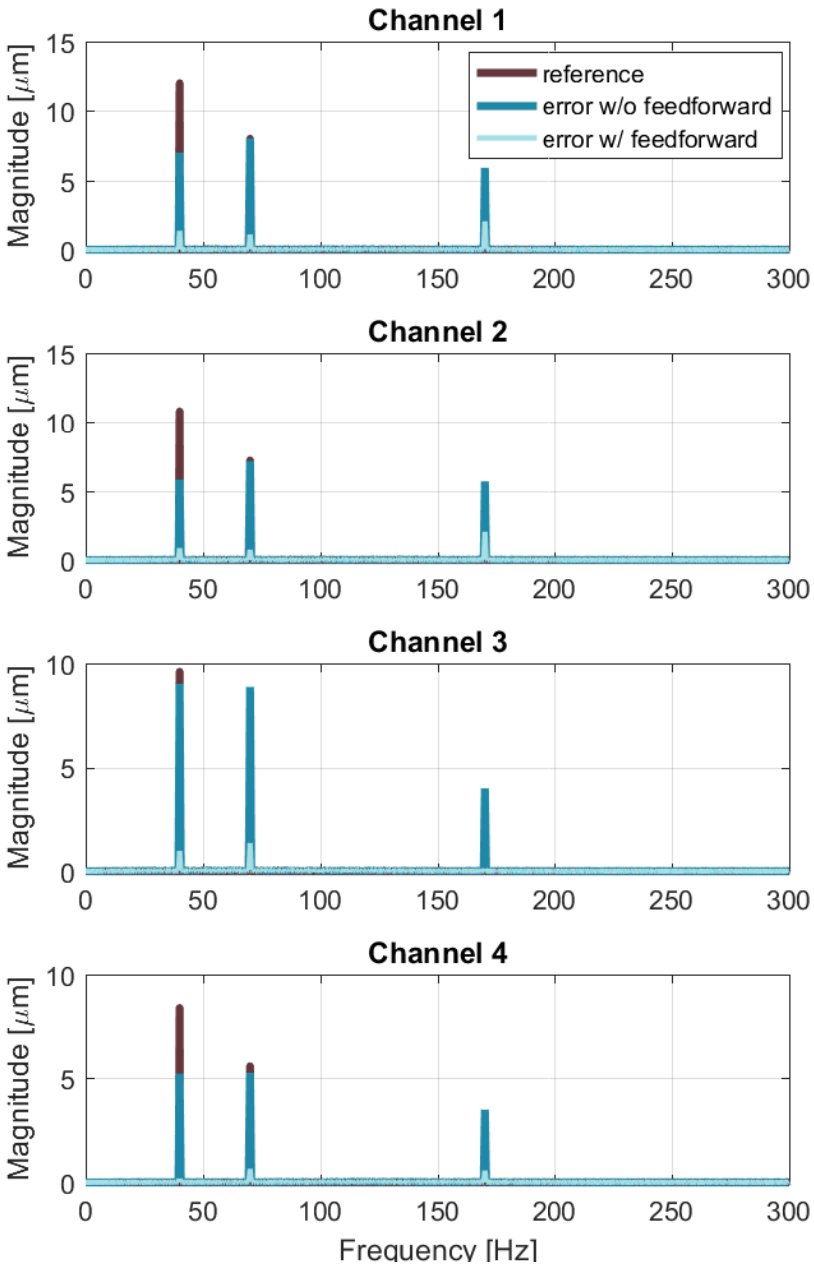


Figure 4.23: Error spectrum of the feedforward tracking on the magnetic bearing spindle system.

4.4.3 Decoupled MIMO System: IRISS

Lastly, the proposed method was implemented on the IRISS robot manipulator. Three joints, namely θ_1, θ_2, and θ_3 as defined in Fig. 2.2, were controlled for tracking the predefined cataract-extraction trajectory as generated in Chapter 3.3. Although the IRISS manipulator is inheritedly an MIMO plant, each of its joints is an independent controlled plant as it is not significantly affected by the motion of other joints. In other words, the IRISS is a decoupled MIMO system. A typical PID control scheme was implemented for tracking the desired reference trajectory. Although the overall performance was acceptable, feedback control created overshoots at the peaks of the trajectory (see Fig. 4.28). The overshoots may cause inadvertent collision between the instrument and the tissues during intraocular surgery, hence, feedforward compensation is required. On the other hand, ILC is not applicable in this application since the predefined trajectory is not the repeatable as the target eye changes. To remove those overshoots without iterative learning process, feedforward filtering is the best option to be implemented for the IRISS manipulator control.

Since the IRISS is a decoupled MIMO system, the identification of the plant inversion and high-precision feedforward control can be achieved by applying Method 4.2 on every each of the three joints. Similar to the experiments on the linear motor and the active magnetic bearing spindle, a zero-phase low-pass filter with bandwidth around 14 Hz was assigned as the reference model $M(z)$ that represents the desired bandwidth of inversion for each joint. The number of delay steps d and the order N of the constructed FIR inversion filters were set as 0.2 and 2 s under 1 kHz sampling rate, respectively. $H(z)$ was assigned as an integrator with proper step size (e.g. 0.9 deg for θ_1) to avoid current command saturation. Three PD-type ILC algorithms were implemented for three individual joints to facilitate the proposed method.

Among the three controlled axes, θ_1 is the most critical joint since it rotates the arc which carries the instrument holder. If θ_1 rotates too aggressively, the inertia created by the mass of the arc and instrument holder will cause significant oscillation. Applying Method

4.2, the output of θ_1 joint took 20 iterations (see Fig. 4.24) to converge to the target step function (Fig. 4.25). It is obvious that the learned inversion filter, as shown in Fig. 4.26, was canceling the oscillation mode of the controlled plant.

The identified inversion filters of each joint were then applied to feedforward filtering of a cataract-extraction trajectory. Note that to demonstrate the capability for high-precise tracking, the IRISS was commanded to move through the predefined trajectory around 5 times than its regular speed. The tracking result on θ_1, comparing to the ones achieved by applying only PID servo control and ILC, is shown in Fig. 4.27. The tracking performance was improved by 40% compared to using only the PID control. The overshoots created by the PID controller were removed after applying the proposed feedforward filtering. As expected, the performance of the proposed method was in the middle between conventional PID servo loop and ILC. The comparison of tool-tip positioning error using the PID control and the proposed feedforward tracking is as listed in Table 4.2.

Table 4.2: The RMS tracking error on the IRISS before and after feedforward filtering. Unit in mm.

Control method	**Max error**	**RMS error**
PID	0.519	0.129
with feedforward	0.314	0.078

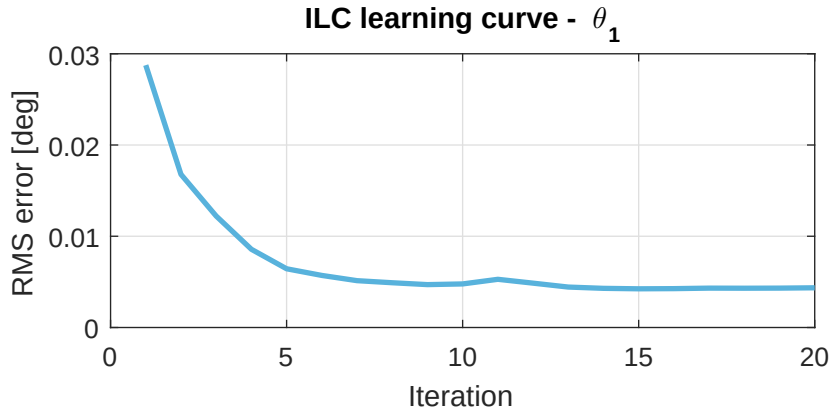


Figure 4.24: Error convergence of iterative learning the target step on the IRISS.

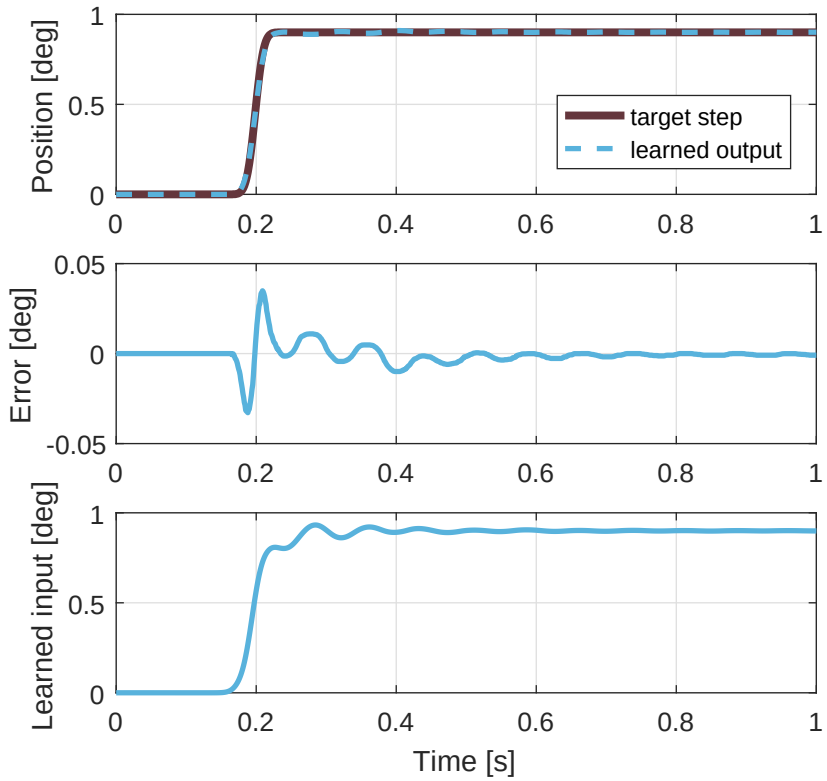


Figure 4.25: Converged results of tracking the target step ($d = 0.2$ s) on the IRISS.

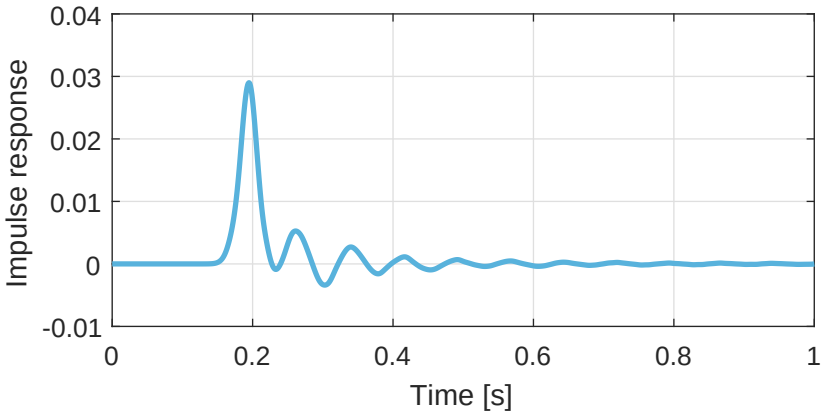


Figure 4.26: Impulse response of the identified plant inversion filter for the IRISS.

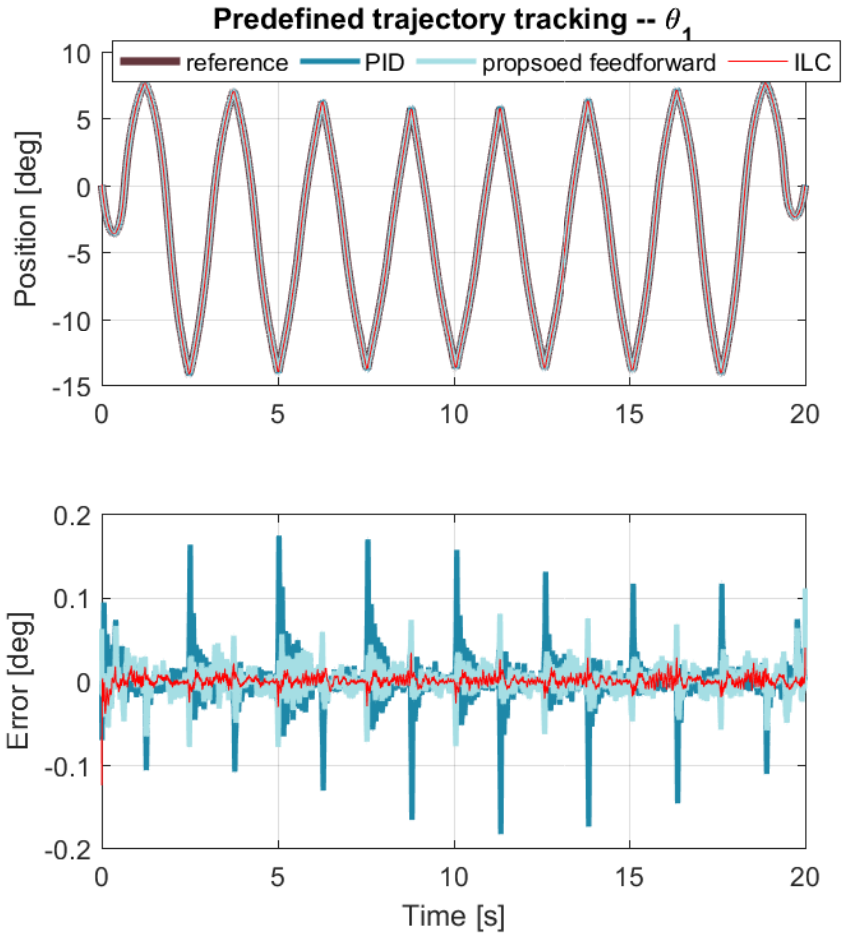


Figure 4.27: Comparison between PID control, ILC, and the proposed feedforward filtering on the tracking of the predefined cataract-extraction trajectory.

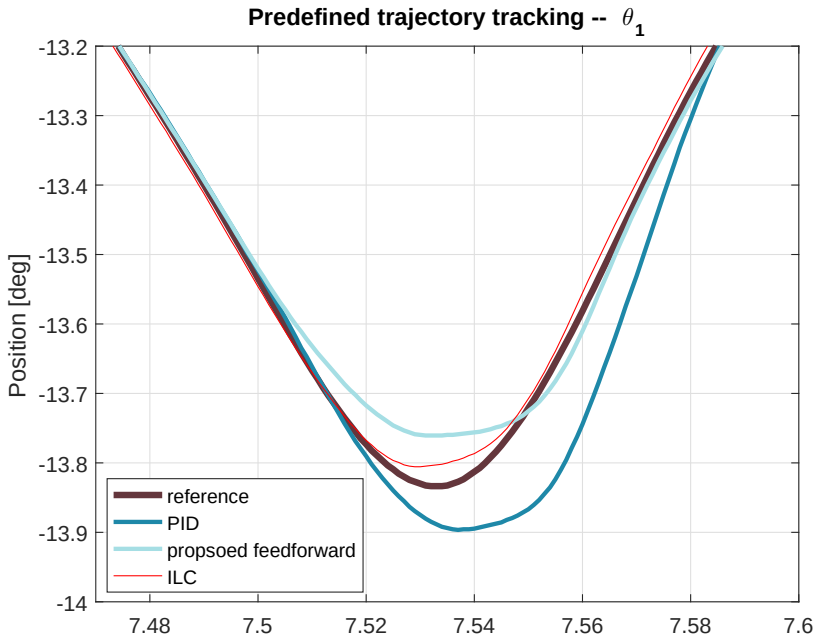


Figure 4.28: Comparison between PID control, ILC, and the proposed feedforward filtering on the tracking of the predefined cataract-extraction trajectory. (zoomed in at the peak).

CHAPTER 5

Iterative Learning Control with Progressive Updates

A novel method which utilizes learning-type control algorithms to identify the plant inversion is proposed and applied to high-precise feedforward tracking control in Chapter 4. This method does not require the model of plant dynamics, however, a pre-stabilized learning-type control algorithm is a prerequisite. Although data-based ILC exists, such learning-type control algorithms usually require the knowledge of plant dynamics for a faster error convergence. To address this problem, this Chapter proposes a novel way to improve the error convergence rate of data-driven ILC. A data-based ILC, here demonstrated by a PD-type ILC or a time-reversal filtering ILC, is first applied to Method 4.2 which is devoted to identifying the plant inversion. After tracking of the target impulse function slowly converges, the learned control input which is formerly used for feedforward tracking is now serving as the learning filter of the ILC. It is also shown that this upgrade of the learning filter can be done "progressively" – meaning it is not necessary to wait until it converges before the next upgrade. The progression of the ILC learning filter brings an additional degree of freedom for the learning filter design with proven stability and convergence properties. Simulation and experimental results on the linear motor positioning system have demonstrated the effectiveness and performance of the approach.

5.1 Data-Based Iterative Learning Control

In the ILC algorithm, the key to success is properly choosing the update law for control input. More specifically, the learning filter of the ILC algorithm determines its performance

in terms of stability, error convergence rate, and converged error. Without the knowledge of plant dynamics, there are several data-based approaches to stabilize the ILC algorithm.

5.1.1 P-Type and PD-Type ILC

P-Type and PD-Type ILC are very popular in the literature since ILC was first proposed in 1970's [CM02]. Without using the knowledge of system dynamics, the learning filter is determined by the proportional and derivative gains, K_p an K_d, on top of the tracking error:

$$u_j(k) = u_{j-1}(k) + K_p e_{j-i}(k) + K_d \dot{e}_{j-i}(k) \tag{5.1}$$

As long as the stability criteria Eq. 4.7 is met, the tracking error to the desired reference will gradually converge.

The performance (in terms of convergence rate and converged error) of P-Type and PD-Type ILC, however, relies on how the gains are tuned. As discussed in the literature, it depends on how well the learning filter approximates the plant inversion [BTA06, NG02, TT15]. Although P-type and PD-type ILC are intuitive to implement, it is not possible to accurately approximate the plant inversion by a constant gain or a first order filter. To ensure the performance of ILC, applying a high order learning filter is essential.

5.1.2 Reverse Time Filtering Based ILC

To enable the construction of inverse filter and to use it as the learning filter for improving the performance of ILC, [CT16] used the model-based ZPETC [Tom87] as the learning filter in the generic ILC. Applying the adjoint system, ZPETC was used to stabilize repetitive controller [TTC89] and was shown could be used as the learning filter for ILC algorithm [Lon00]. The convergence and robustness properties of adjoint operator were studied in the norm optimal ILC [KSA02, RHL08, OHD09], where a precise plant model is applied.

For model-free ILC design, time-reversal technique was proposed to realize the adjoint operator [WYZ14, Gus96, YW05]. This approach was later extended to handle point-to-point

tracking [PW04], non-minimum phase plants [FLR07], stochastic disturbance [BKL08], and multivariable systems [BO15a].

Definition 5.1 (Time-reversal operator) *A time reversal operator $\mathcal{R}$ which flips a sequence of signal is defined as an N-by-N involutory permutation matrix such that*

$$y^*(k) = y(N-k) = \mathcal{R}y(k) \tag{5.2}$$

Definition 5.2 (Adjoint system) *G^*, with an input y and an output w, is the adjoint system of the SISO plant G where*

$$w(k) = G^*(z)y(k) = \mathcal{R}G(z)\mathcal{R}y(k) \tag{5.3}$$

Because G^*G forms a system with positive magnitudes and zero phase angles, G^* can serve as the learning filter and stabilize the model-free iterative process after scaled by a sufficiently small constant learning gain α:

$$L(z) = \alpha G^*(z) \tag{5.4}$$

When α is sufficiently small, the stability criteria of Eq. 4.7 is met and therefore the stability of the reverse time filtering based ILC is guaranteed. However, small α results in conservative updating and thus slows down the error convergence rate.

5.2 Progression of ILC

To improve the performance of ILC without tedious modeling or tuning process, an algorithm that updates the learning filter progressively is proposed in this study. The main idea is to use a non-optimized data-based ILC to identify the plant inversion (Method 4.2), and then plug the approximated inversion back as the learning filter to track other desired reference trajectories. This updating method is very similar to the approach proposed in [SFT09] where the inversion filter is obtained by solving a least-square optimization and is used to calculate the feedforward input in the next ILC iteration. With the quality of identified

inversion filter as demonstrated in Chapter 4, the performance of ILC after evolution will undoubtedly be improved. The only problem of this method is that, it will require significant amount of iterations for the *embryonic* data-based ILC to identify the plant inversion. This is once again a chicken-and-egg problem: a well-approximate inversion improves the learning-type control in terms of error convergence rate, while a fast-converging learning-type control is somehow required for identifying the plant inversion. In this section, progressive update of the learning filter is proposed to resolve this problem, even when the *embryonic* ILC is conservative and slow.

5.2.1 Evolution of Learning Filter

With any non-optimized (in terms of error convergence rate) data-based ILC, Method 4.2 is first applied to identify the plant inversion. Once the tracking of target impulse/step function converges, the dynamic inversion is identified and the learned input is used to construct an FIR inversion filter. The *embryonic* data-based learning filter is then replaced by the identified plant inversion filter. This process is called evolution of learning filter:

Method 5.1 (Data-Based Evolution of ILC). For a SISO controlled plant $G(z)$ with an arbitrary stable ILC learning filter $L(z)$ that satisfies Eq. 4.7, the plant inversion $F(z)$ is approximated by the converged input to track a smoothed delta impulse $m(k-d)$:

$$\begin{aligned} f(k) &= \lim_{j_{evol}\to\infty} u_{j_{evol}}(k) \\ &= \lim_{j_{evol}\to\infty} [1-(1-L(z)G(z))^{j_{evol}}]G^{-1}(z)m(k-d) \end{aligned} \tag{5.5}$$

where j_{evol} is the iteration index in the identification stage, $f(k)$ is the impulse response of $F(z)$, and $L(z)$ is the learning filter of the non-optimized ILC.

After obtaining the approximated plant inversion $F(z)$, a d-step non-causal preview z^d along with the inversion, i.e. $F(z)z^d$, will replace $L(z)$ and will be used as the learning filter of ILC to track the pre-defined references. □

Remark 5.1 (Bandwidth of the inversion) *The bandwidth of the approximation of in-*

version is determined by the target impulse $m(k)$, which is the impulse response of a zero-phase low pass filter $M(z)$. The inversion filter $F(z)$ approximates $G(z)F(z)$ to $M(z)$. In the other words, the inversion at the stop-band of $M(z)$ converges to zero. This can protect the controlled plant from control saturation, which may cause instability onto ILC convergence.

As stated before, the performance of ILC in terms of stability and error convergence rate is the main concern when designing the learning filter. It can be shown that the evolved learning filter $F(z)z^d$ can stabilize the ILC algorithm if the *embryonic* learning filter $L(z)$ satisfies the ILC stability criteria Eq. 4.7:

Theorem 5.1 *(Stability of the evolved learning filter). The evolved learning filter $F(z)z^d$ stabilizes the ILC algorithm on the plant $G(z)$, i.e.,*

$$\|1 - F(z)z^d G(z)\|_\infty < 1 \quad if \quad \|1 - L(z)G(z)\|_\infty < 1 \tag{5.6}$$

Proof 5.1

$$\begin{aligned}
&\|1 - F(z)M^{-1}(z)z^d G(z)\|_\infty \\
=&\|1 - \lim_{j_{evol}\to\infty}[1 - (1 - L(z)G(z))^{j_{evol}}]G^{-1}(z)z^{-d}M(z)M^{-1}(z)z^d G(z)\|_\infty \\
=&\|\lim_{j_{evol}\to\infty}(1 - L(z)G(z))^{j_{evol}}\|_\infty \\
=&\lim_{j_{evol}\to\infty}(\|1 - L(z)G(z)\|_\infty)^{j_{evol}} \\
<&\|1 - L(z)G(z)\|_\infty < 1
\end{aligned} \tag{5.7}$$

This implies that in the polar coordinate $F(z)M^{-1}(z)z^d G(z)$ is inside the unity circle around 1. Therefore,

$$\begin{aligned}
\|1 - F(z)z^d G(z)\|_\infty &= \|1 - M(z)(F(z)M^{-1}(z)z^d G(z))\|_\infty \\
&< 1
\end{aligned} \tag{5.8}$$

because $M(z)$ is zero phase and

$$\|M(z)\|_\infty \leq 1 \tag{5.9}$$

□

It has also been shown that the evolved learning filter by applying Method 5.1 not only stabilizes the ILC algorithm, but also exponentially improves the error convergence rate:

Theorem 5.2 *(Error convergence rate of the evolved learning filter). Assume the bandwidth of the reference trajectory $r(k)$ is limited by the low-pass filter $M(z)$, i.e., $(1 - M(z))r = 0$, the evolved learning filter $F(z)z^d$ exponentially improves the error convergence rate.*

Proof 5.2

$$\begin{aligned}
\|e_j^{<F>}(k)\|_\infty &= \|(1 - G(z)F(z)z^d)^j r(k)\|_\infty \\
&= \|[(1 - M(z)) + M(z)(1 - F(z)M^{-1}(z)z^d G(z))^j]r(k)\|_\infty \\
&= \|(1 - M(z))r(k) + M(z) \lim_{j_{evol}\to\infty}(1 - L(z)G(z))^{j_{evol}j} r(k)\|_\infty \qquad (5.10) \\
&= \| \lim_{j_{evol}\to\infty}(1 - L(z)G(z))^{j_{evol}j} r(k)\|_\infty \\
&< \|(1 - G(z)L(z))^j r(k)\|_\infty = \|e_j^{<L>}(k)\|_\infty
\end{aligned}$$

where $e_j^{<L>}(k)$ and $e_j^{<F>}(k)$ denote the tracking error at j-th iteration when using $L(z)$ and $F(z)z^d$ as the learning filter, respectively. □

Although the stability and better error convergence rate of the evolved learning filter is proven, Method 5.1 faces a major problem when using the non-optimized data-based ILC to identify the plant inversion. If the upper bound of $\|1-L(z)G(z)\|_\infty$ is too close to 1, meaning the *embryonic* learning filter is very conservative and therefore slows down the convergence rate, it will take thousands of iteration before the tracking of the target impulse/step function converges. This issue makes Method 5.1 hard to be implemented.

5.2.2 Progressive Update of Learning Filter

To further reduce the number of iteration required for plant inversion identification and learning filter evolution, a method that progressively updates the learning filter during the identification stage is proposed. As the block diagram shown in Fig. 5.1, the update of the learning filter is taken place every n iterations:

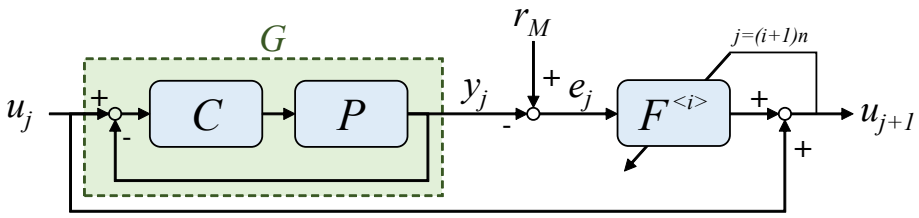


Figure 5.1: The control diagram of progressive ILC; i is the index of ILC progression and j is the index of ILC iteration.

Method 5.2 (Data-Based Progression of ILC). For a SISO controlled plant $G(z)$ with an arbitrary stable ILC learning filter $F^{<i>}(z)$ that satisfies Eq. 4.7, the learned input $u_n(k)$ (for tracking a smoothed delta impulse $m(k-d)$ after n iterations) is used to upgrade the learning filter as $F^{<i+1>}(z)$:

$$\begin{aligned} f^{<i+1>}(k) &= u_n(k+d) \\ &= [1-(1-F^{<i>}(z)G(z))^n]G^{-1}(z)m(k) \end{aligned} \tag{5.11}$$

where $f^{<i+1>}(k)$ is the impulse response of $F^{<i+1>}(z)$. For the initial condition $i = 0$, the *embryonic* learning filter $F^{<0>}(z)$ could be any non-optimized data-based ILC such as P-type, PD-type, or reverse time based filtering.

By progressively updating the learning filter, the required number of iteration for identifying the plant inversion is significantly reduced. Once the plant inversion is identified (i.e. the tracking error of the target impulse/step function is below to the error threshold as stated in Method 4.2), the latest upgraded learning filter $F^{<i>}(z)$ will be used as the learning filter of ILC to track any pre-defined reference. □

Method 5.2 illustrates the ILC progression for the dynamic inversion and learning filter $F^{<i+1>}(z)$. At the initial progression, i.e. i=0 (denoted as Prog. 0), the *embryonic* learning filter of the ILC is conservatively selected to meet the stability criteria Eq. 4.7. After tracking the target smoothed impulse/step function for n iterations, the error slightly decreased (contributed by the conservative learning filter $F^{<0>}(z)$). Then, the learned input $u(k)$ at the n-th iteration are used to construct the FIR learning filter $F^{<1>}(z)$ by applying Eq.

5.11 in Method 5.2. The same process is repeated until $y(k)$ converges to $m(k-d)$. Similar to Method 5.1, the stability and error convergence rate of the progressively updated learning filters can be proven:

Theorem 5.3 *(Stability of the progressively updated learning filter). The progressively updated learning filter $F^{<i+1>}(z)$ stabilizes the ILC algorithm on the plant $G(z)$, i.e.,*

$$\|1 - F^{<i+1>}(z)G(z)\|_\infty < 1 \quad if \quad \|1 - F^{<i>}(z)G(z)\|_\infty < 1 \tag{5.12}$$

Proof 5.3

$$\begin{aligned} &\|1 - F^{<i+1>}(z)M^{-1}(z)G(z)\|_\infty \\ =&\|1 - [1 - (1 - F^{<i>}(z)G(z))^n]G^{-1}(z)M(z)M^{-1}(z)G(z)\|_\infty \\ =&\|(1 - F^{<i>}(z)G(z))^n\|_\infty \\ =&(\|1 - F^{<i>}(z)G(z)\|_\infty)^n \\ <&\|1 - F^{<i>}(z)G(z)\|_\infty < 1 \end{aligned} \tag{5.13}$$

This implies that in the polar coordinate $F^{<i+1>}(z)M^{-1}(z)G(z)$ is inside the unity circle around 1. Therefore,

$$\begin{aligned} \|1 - F^{<i+1>}(z)G(z)\|_\infty &= \|1 - M(z)(F^{<i+1>}(z)M^{-1}(z)G(z))\|_\infty \\ &< 1 \end{aligned} \tag{5.14}$$

because $M(z)$ is zero phase and

$$\|M(z)\|_\infty \leq 1 \tag{5.15}$$

□

The error convergence rate is also improved exponentially after every each progression:

Theorem 5.4 *(Error convergence rate of the progressively updated learning filter). Assume the bandwidth of the reference trajectory $r(k)$ is limited by the low-pass filter $M(z)$, i.e., $(1-M(z))r(k) = 0$, the progressively updated learning filter $F^{<i+1>}(z)$ exponentially improves the error convergence rate over $F^{<i>}(z)$.*

Proof 5.4 *According to Eq. 4.6, the error convergence rates when using* $F^{<i+1>}(z)$ *and* $F^{<i>}(z)$ *as the learning filter are* $(1 - G(z)F^{<i+1>}(z))$ *and* $(1 - G(z)F^{<i>}(z))$*, respectively.*

$$\begin{aligned} 1 - G(z)F^{<i+1>}(z) &= 1 - M(z) + M(z)[1 - F^{<i+1>}(z)M^{-1}(z)G(z)] \\ &= 1 - M(z) + M(z)[(1 - F^{<i>}(z)G(z))^n] \end{aligned} \tag{5.16}$$

For the desired pre-defined trajectory $r(k)$*,* $(1 - M(z))r(k) = 0$ *and* $M(z)r(k) = r(k)$*. Therefore, the error convergence rate can be further reduced to* $(1 - F^{<i>}(z)G(z))^n$*, which is the error convergence rate when using* $F^{<i>}(z)$ *as the learning filter to the* n*-th.* □

The proposed progressive and iterative learning control reveals an additional degree of freedom for designing ILC algorithm. While iterative process decreases tracking error, progression of ILC exponentially accelerates error convergence. Therefore, the slow convergence of the non-optimized data-based ILC, due to conservative learning gain assignment, can be significantly improved. This approach solves the chick-and-egg problem. Starting with a conservative learning filter, the plant inversion can be quickly identified. The identified inversion filter is beneficial to either data-based feedforward filtering or ILC.

5.3 Simulation Results

The proposed progressive ILC was simulated on a SISO model of the single axis linear motor positioning system. A PD-controller was first used to stabilize the motor because the system is open-loop unstable. A 2000 points FIR model for the following simulation was obtained from experimental step response data of the closed-loop system, with sampling rate of 10 kHz. As the impulse response shown in Fig. 5.2, the main dynamics occur at first 5 ms, while oscillation lasts for 30 ms. To validate the proposed data-driven method, this plant model was only used for emulating the motor dynamics, but not for designing any filter or controller.

The *embryonic* learning filter for ILC progression (Prog. 0) was implemented by reverse time based filtering as stated in Chapter 5.1.2. A small gain α was set as 0.01 to ensure the

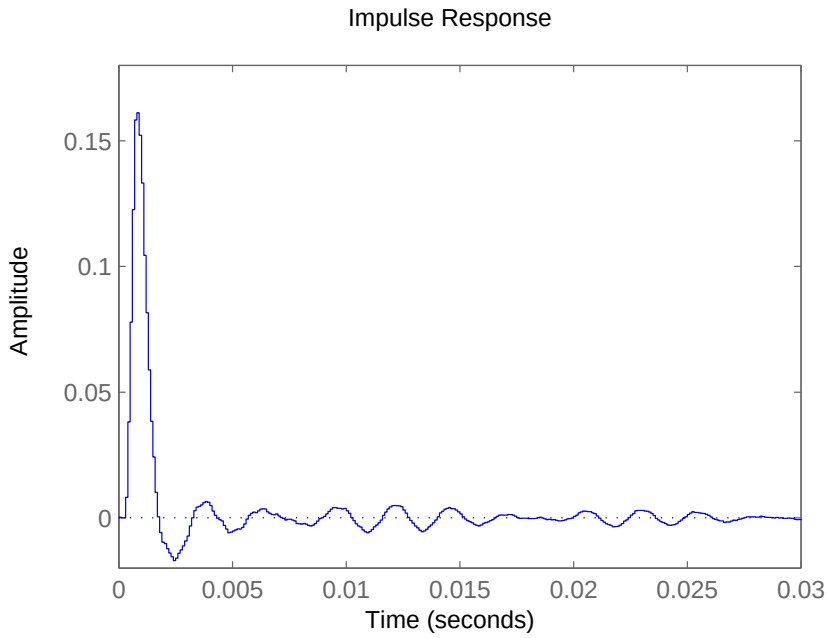


Figure 5.2: Impulse response of the FIR linear motor model $G(z)$.

stability criteria Eq. 4.7 was met. The reference model $M(z)$ was assigned as a zero-phase low pass filter,

$$M(z) = (\frac{z^{-1} + 2.1 + z}{4.1})^r \tag{5.17}$$

where r equals 40 for 300 Hz bandwidth. The impulse response of $M(z)$, also denoted as the target filtered delta function $m(k)$, can be found in Fig. 5.5.

In the simulation of ILC progression (Method 5.2), 9 progressions were performed, and 10 iterations of error updating were run in each progression (n=10 and max(i)=9). In order to demonstrate the difference of the error convergence rate of each progression, the control input $u(k)$, which was trained to track the targeted impulse $m(k)$, was reset whenever new progression was generated. The results of ILC progression is shown in Fig. 5.3.

In Fig. 5.3, Prog. 0 converged extremely slow because the learning gain α was set to be 0.01 such that the model-free iterative process was stable and converging. As the progression number i increased, the error convergence rate was accelerated. At the 9th progression, the

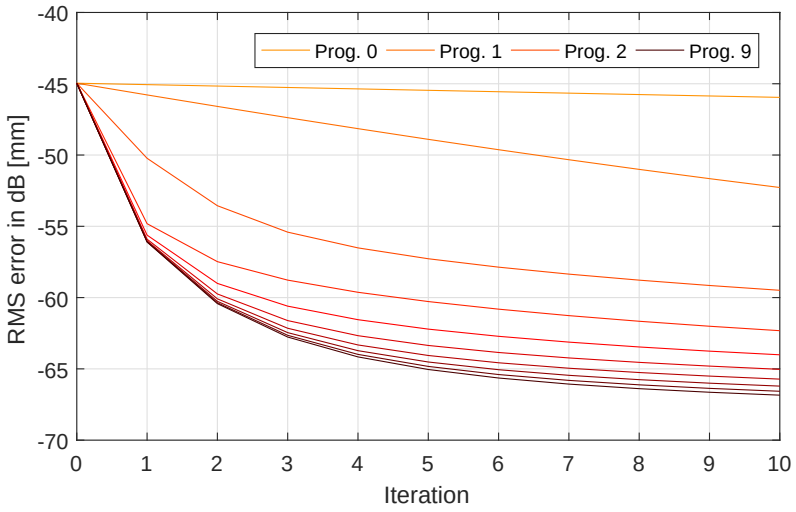


Figure 5.3: Progression of ILC with input reset in the simulation.

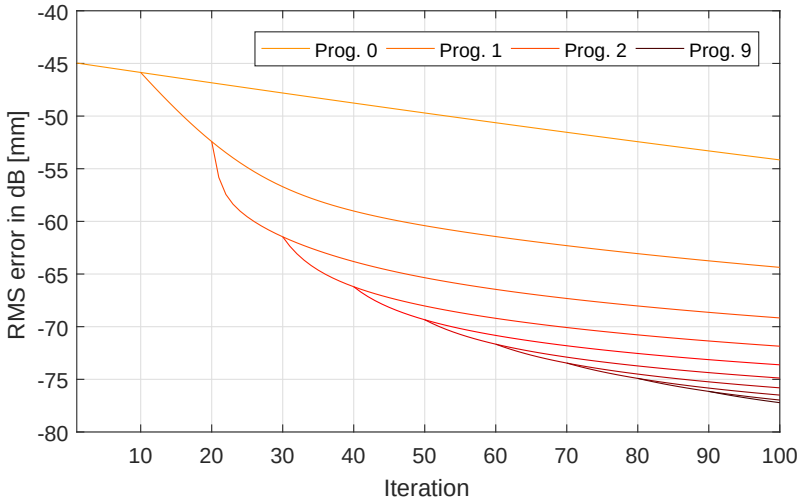


Figure 5.4: Progression of ILC without input reset in the simulation.

tracking error of the target impulse $m(k)$ converged to 0.45 μm, meaning that the dynamic inversion has been successfully approximated. If there was no progressive algorithm to update the learning filter $F^{<0>}(z)$, Prog. 0 might take hundreds or even thousands of trials to reach the same level of error amount as in Prog. 9.

In actual applications, there is no need to reset the control input, therefore much less

iteration and progression are required for identifying the plant inversion. For the case without resetting after each progression, the converged error to track the filtered impulse $m(k)$ (see Fig. 5.4) is smaller compared to the one with resetting initial input (Fig. 5.3).

The learned output $y(k)$ and the corresponding impulse response of the identified dynamic inversion $F^{<i>}(z)$ at Prog. 0, 1, 2, and 9 are shown in Fig. 5.5 and Fig. 5.6, respectively. At the 9th progression, the learned output $y(k)$ had the smallest error to the desired filtered delta function $m(k)$ (bottom right of Fig. 5.6) than the others, meaning that the corresponding dynamic inversion $F^{<9>}(z)$ more precisely approximates the plant inversion than $F^{<0>}(z)$, $F^{<1>}(z)$, and $F^{<2>}(z)$. Similarly, $F^{<2>}(z)$ is better than $F^{<0>}(z)$ and $F^{<1>}(z)$ because it has smaller amount of error, $m(k) - y(k)$, as shown in Fig. 5.5.

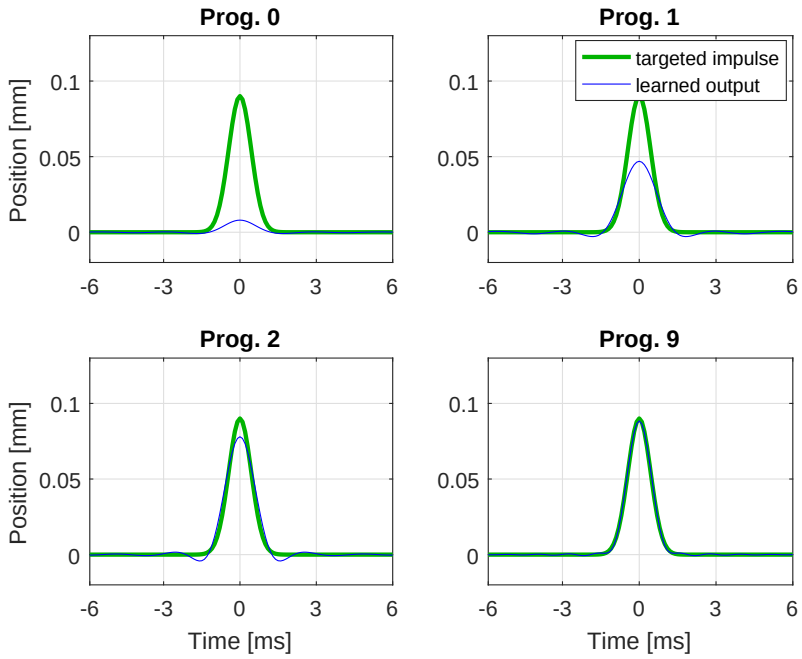


Figure 5.5: Progressively learned filtered impulse in the simulation.

After the ILC progression was performed, the progressively identified dynamic inversions $F^{<1>}(z)$, $F^{<2>}(z)$, and $F^{<9>}(z)$ were then applied as the learning filter of ILC for reference tracking. Though the progressive ILC can be applied to track any arbitrary trajectory, a 2 s chirp signal with the maximal frequency of 100 Hz was assigned as the tracking reference

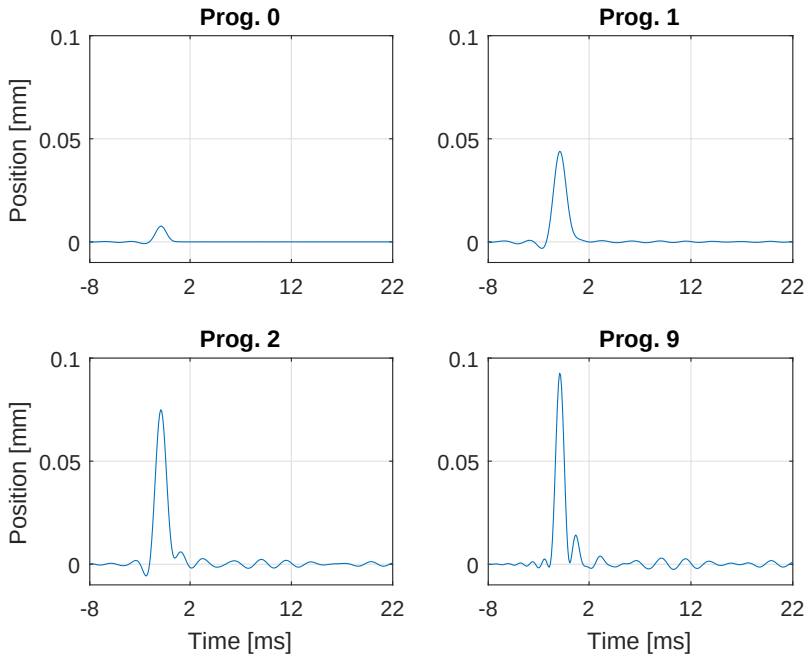


Figure 5.6: Impulse responses of progressively learned dynamic inversion $F^{<i>}(z)$ in the simulation.

to verify the performance of the learning filters over broadband.

The comparison of the error convergence rate by using different progressively identified dynamic inversions $F^{<i>}(z)$ as the learning filter is shown in Fig. 5.7. As discussed in Chapter 5.2, the convergence rate was accelerated as the progression number i increased. At the 2nd iteration, for example, Prog. 0 decreased the error to 19.96 dB while 3.64 dB by Prog. 1 and -37.31 dB by Prog. 9. After Prog. 2, the error converged below -30dB (31.6 μm) within only 3 iterations, compared to 67 iterations took by Prog. 0. Clearly, the later progressed ILC generated less tracking error at every iteration. The benefits of using the proposed progressive ILC in terms of error convergence rate and converged error are both demonstrated by this simulation.

The tracking performance of the identified dynamic inversion $F^{<i>}(z)$ as a feedforward controller was also examined at the first iteration (Fig. 5.8). There was a large amount of error on magnitude when using $F^{<0>}(z)$. This suggests time-reversal approaches could only

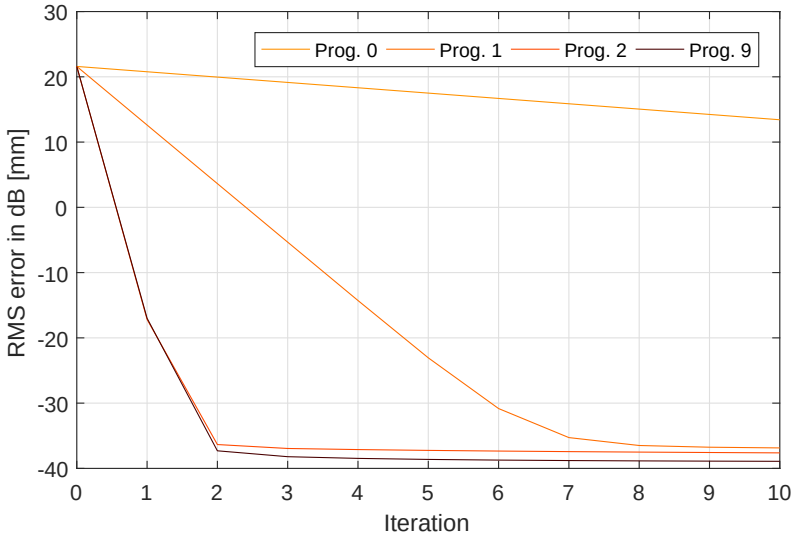


Figure 5.7: Simulated comparison of the ILC error convergence rate by applying different progressively identified inversion $F^{<i>}(z)$ as the learning filter for tracking a chirp reference.

ensure zero phase cancellation and take a long time to learn the desired magnitude. On the other hand, $F^{<9>}(z)$ derived after 9 progressions was able to track the chirp signal with the RMS error of 0.14 mm.

At the 10-th iteration, the performance of progressive ILC tracking is illustrated in Fig. 5.9. Clearly, $F^{<9>}(z)$ achieved the minimum tracking error at the 10th iteration. Please be noted that Prog. 2 and 9 are mostly overlapped in Fig. 5.9, except the transient from 0 to 0.1 s where the 9th Progression converged faster than the 2nd. This also indicates that Prog. 9 provides a better learning filter than the previous progressions.

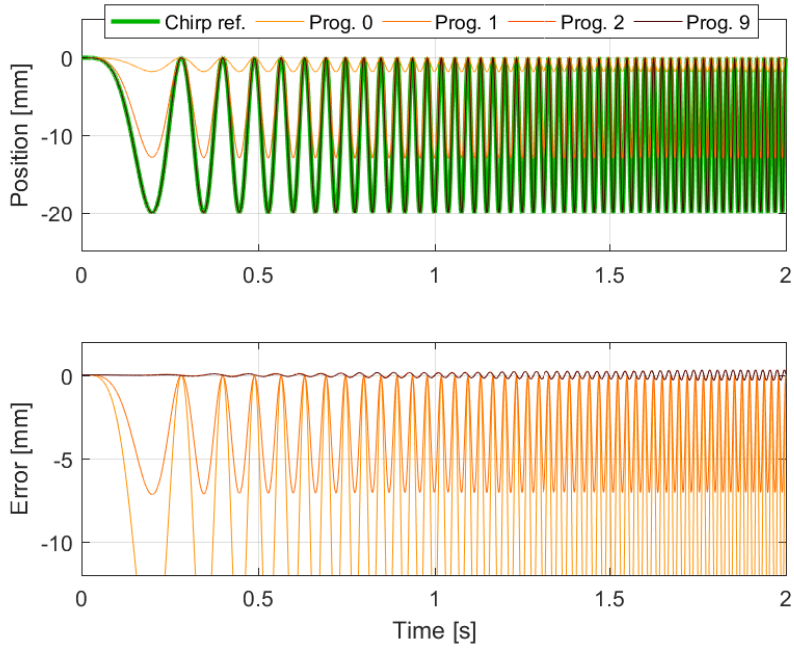


Figure 5.8: Simulated comparison of the feedforward tracking performance (the first iteration).

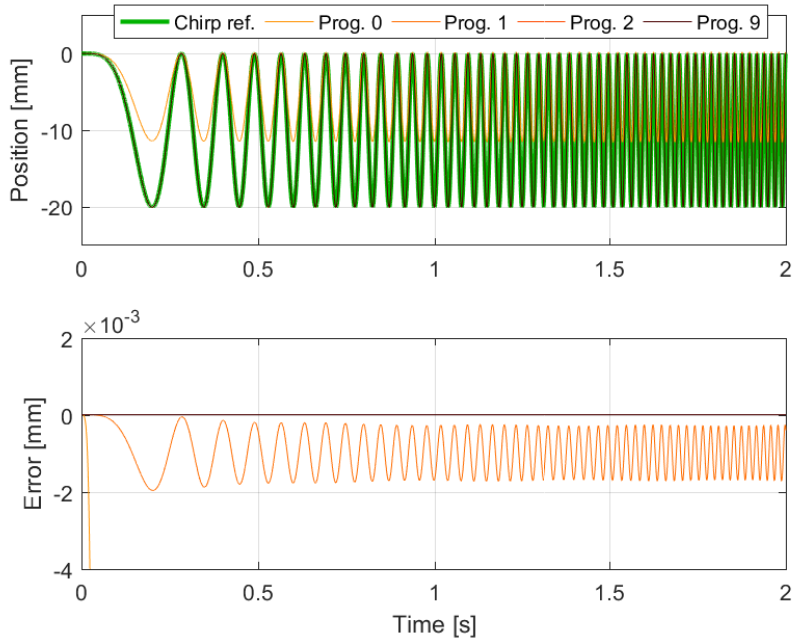


Figure 5.9: Simulated comparison of the tracking performance at the 10th iteration.

5.4 Experimental Results

The proposed progressive ILC was implemented on the actual linear motor. Similar to the simulation, the *embryonic* learning filter for ILC progrssion (Prog. 0) was implemented by reverse time based filtering with a small gain $\alpha = 0.005$ to ensure the stability. A zero-phase low-pass filter with bandwidth of 500 Hz was designed and assigned as the target reference model $M(z)$. The bode diagram of $M(z)$ is illustrated in Fig. 5.11.

In the experiment, 9 progressions were performed without resetting the input, and 10 iterations of error updating were run in each progression (n=10 and max(i)=9). The experimental result is shown in Fig. 5.10. Similar to the simulation result, Prog. 2 converged and identified the plant inversion within 30 iterations, while Prog. 0 had to take more than 100 iterations to do so. Actually, the converged error at the 100th iteration by using reverse time based filtering $F^{<0>}(z)$ is even larger than the 20th iteration done by the first progression $F^{<1>}(z)$. This experimental result also validates Theorem 5.4 that the error convergence

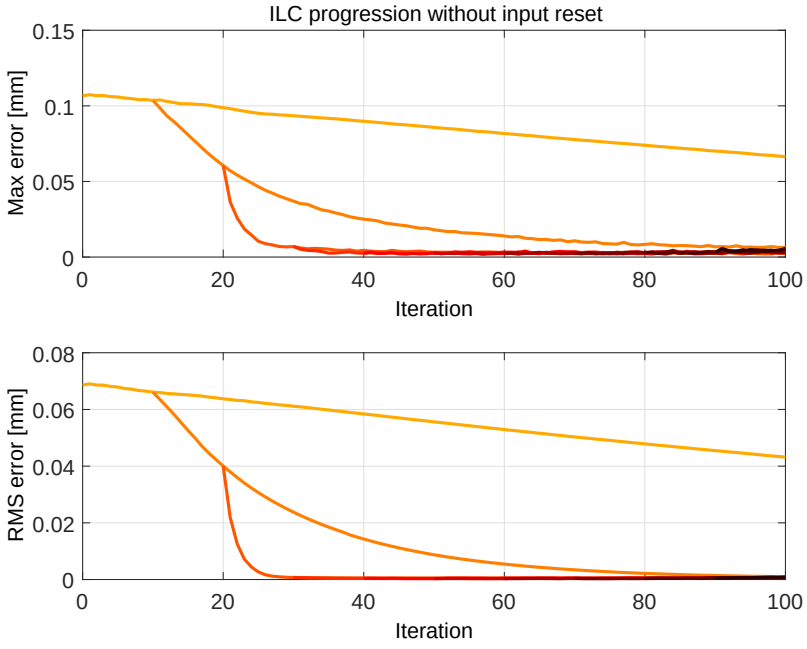


Figure 5.10: Progression of ILC without input reset in the experimental result.

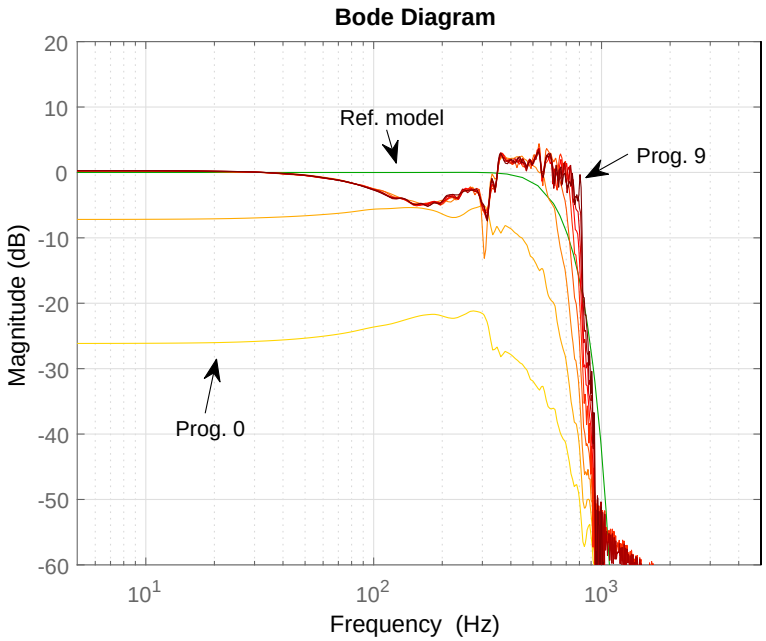


Figure 5.11: Spectrum of progressively learned dynamic inversion in the experimental result.

rate accelerates exponentially after each progression.

The spectra of progressively learned inversion filters are shown in Fig. 5.11. After Prog. 2, the spectra under 500 Hz converged, meaning the dynamic inversion was well-approximated. Since the bandwidth of reference model $M(z)$ was selected as 500 Hz, the cut-off frequencies of the identified inversion filters $F^{<i>}(z)$ are also around 500 Hz. This prevents the feedforward input from being saturated, and therefore deteriorating the performance of ILC.

Similar to the simulation, the progressively identified dynamic inversions $F^{<1>}(z)$, $F^{<2>}(z)$, and $F^{<9>}(z)$ were then applied as the learning filter of ILC for reference tracking after the ILC progression was performed (Fig. 5.12). The performance of feedforward control and ILC are illustrated in Fig. 5.13 and Fig. 5.14, respectively. It is obvious that with ILC progression, the tracking error could be minimized with much less iterations spending during identification stage.

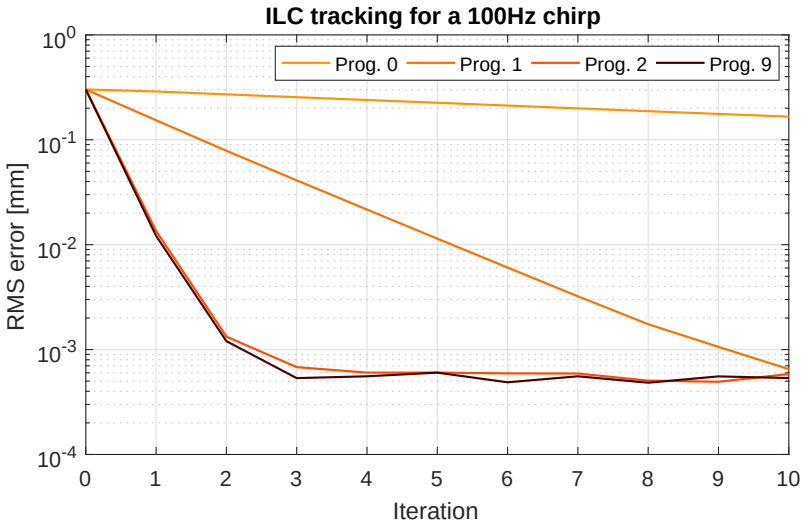


Figure 5.12: Experimental comparison of the ILC convergence rate by applying different progressively reconstructed inversion $F^{<i>}$ as the learning filter for tracking a chirp reference.

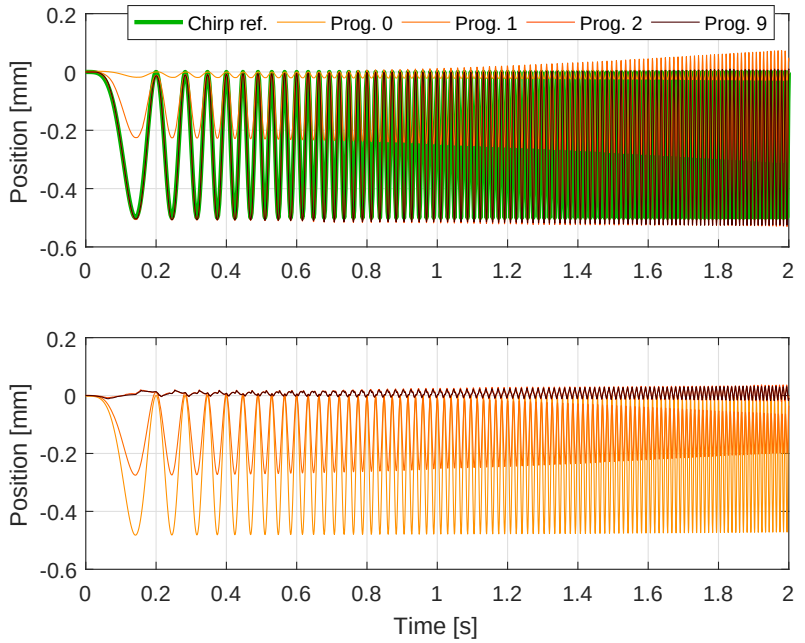


Figure 5.13: Experimental comparison of feedforward tracking performance (the first iteration).

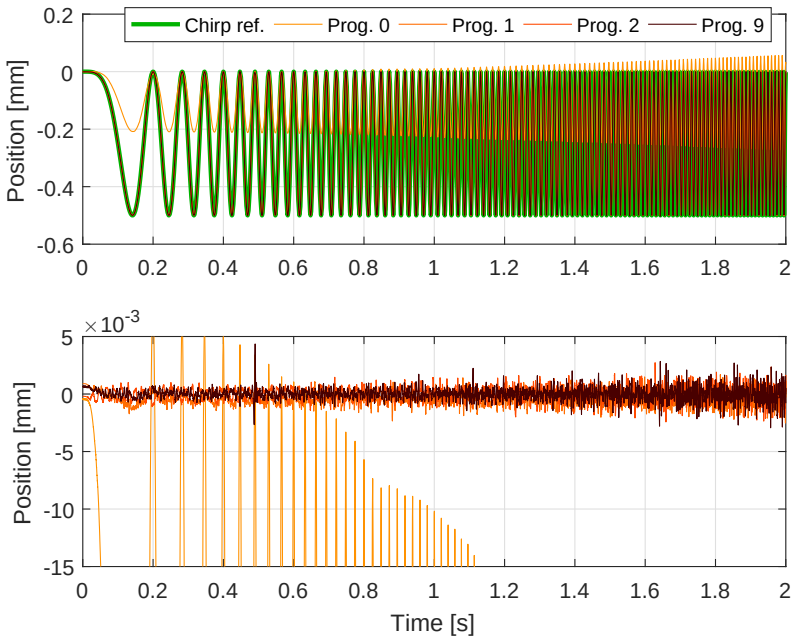


Figure 5.14: Experimental comparison of tracking performance at the 10th iteration.

CHAPTER 6

Evaluations on an Animal Model

The integrated system was tested in a clinical environment (Fig. 6.1). An animal model evaluation was done by performing automated lens removal on post-mortem pig eyes.

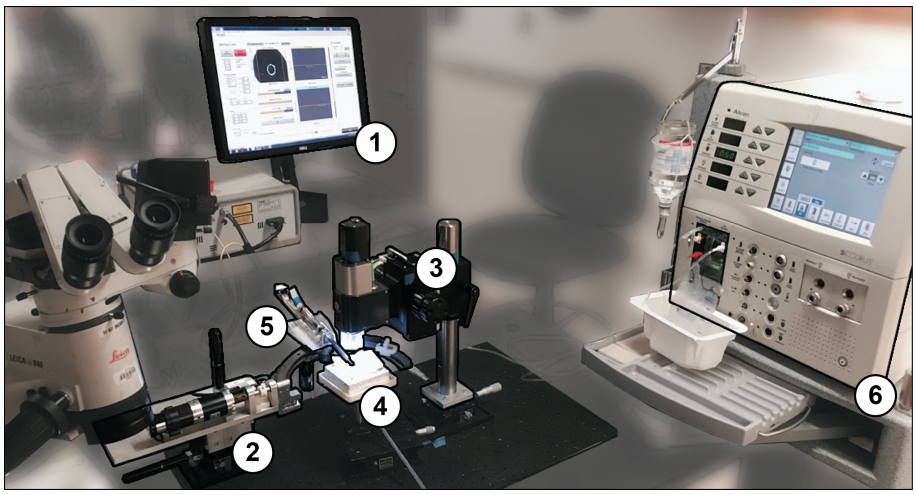


Figure 6.1: Shown is the experimental setup for performing the animal model evaluation of the system on post-mortem pig eyes. (1) User interface, (2) the IRISS, (3) OCT probe with integrated camera, (4) pig eye in fixture, (5) I/A tool in holder, and (6) the ACCURUS® system.

6.1 Experimental Setup and Evaluation Metrics

The evaluation of the integrated system was performed on post-mortem pig eyes pinned in a Styrofoam holder. The main objective was to autonomously remove the entire lens without rupturing the posterior capsule. Because femtosecond laser equipment was unavailable

for the experiments, manual preparation of each eye was performed by a human surgeon under a surgical microscope. The surgeon created a uniplanar corneal incision with a 2.8 mm keratome knife, performed circular and continuous capsulorhexis of 5 mm diameter, and performed hydrodissection and hydrodelamination of the lens with balanced salt solution. Because the automated alignment, tool insertion, volume scanning, and anatomical model generation required approximately ten minutes following eye preparation, as a final preparation step, the anterior chamber was filled with viscoelastic liquid to prevent cornea collapse.

The I/A hand-piece tool (92-IA21 Handle, Millennium Surgical) was mounted on the IRISS and fitted with a straight tip with side aspiration port (92-IA225). The tubing of the I/A tool was connected to the ACCURUS® to provide irrigation and aspiration, the magnitude of which was controlled by the IRISS.

The automated OCT-guided lens extraction was performed and tested on $n = 30$ post-mortem pig eyes. For every eye, preoperative, intraoperative, and postoperative OCT volume scans were acquired for analysis. Microscope-based examination was performed by a trained surgeon to assess the integrity of the tissues and to determined if lens material remained. The evaluation metrics were:

1. Completeness of lens removal (Y/N)
2. Posterior capsule rupture (Y/N)
3. Iris damage (damage score 0–3)
4. Cornea damage (damage score 0–3)
5. Incision stress (stress score 0–3)
6. Completion time (in s)

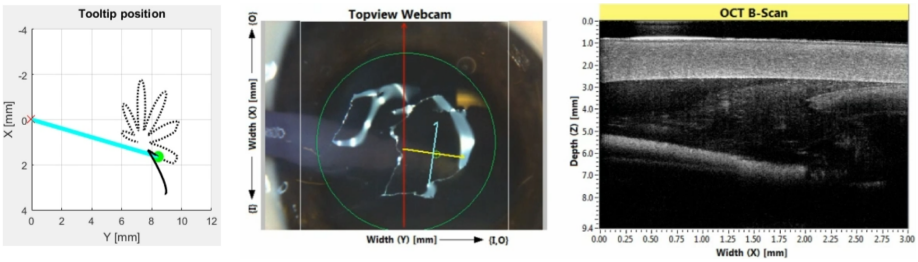
where the scores are qualitatively defined in Table 6.1.

Table 6.1: Description of Post-surgical Evaluation Scores

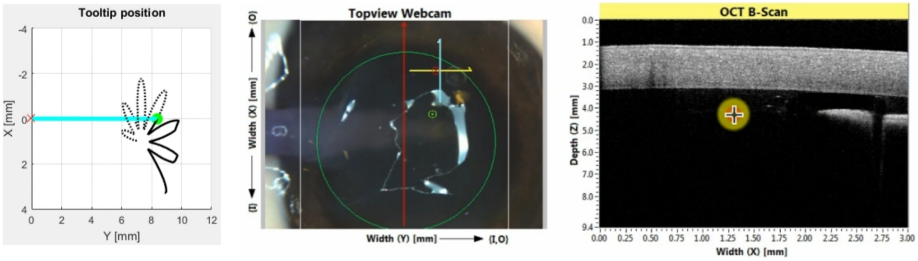
Score	Cornea Damage	Iris Damage	Incision Stress
0	No evidence of endothelial or stromal defect	No iris touch	Preserved incision
1	Mild descemet folds, no stromal defect	Iris touch without damage	Mild opening of the incision, does not compromise sealing
2	Descemet fold and mild corneal edema	Iris touch and damage in one site	Opening of the incision, compromised sealing
3	Opaque cornea	Iris touch and damage in multiple sites	Opening and widening of the incision with compromised sealing

6.2 Bubble Removal by Intraoperative Intervention

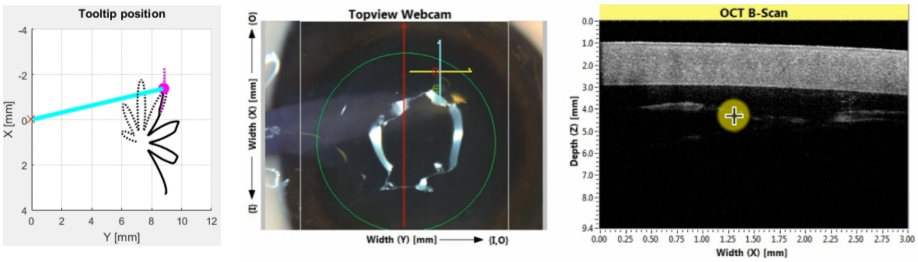
To validate the ability of real-time intraoperative targeted intervention, the I/A tool was commanded to remove an artificially injected air bubble from the anterior chamber (Fig. 6.2). Removing a bubble is technically more challenging than removing a piece of lens material because the bubble has lower visual contrast in OCT images and is smaller than a piece of lens material. Despite these challenges, the depth information provided by the real-time OCT images enabled the system to target the bubble and successfully aspirate it.



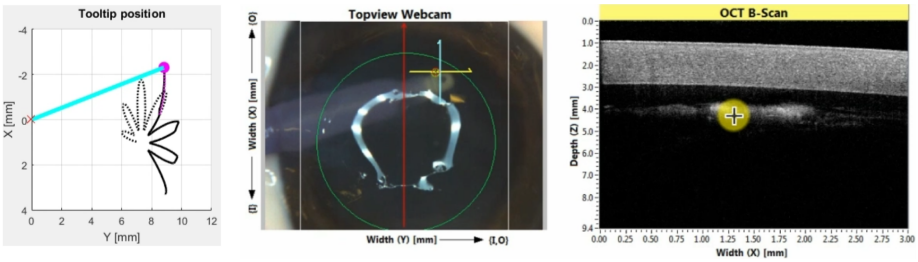
(a) Here, the tool tip follows the predefined cataract-extraction trajectory.



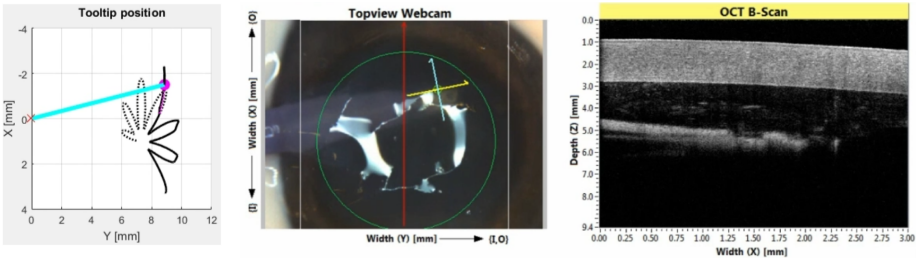
(b) At this point, the bubble is assigned as a target by clicking the camera and OCT image.



(c) The tool tip moves towards the bubble.



(d) The tool has reached the bubble and has successfully aspirated it.



(e) Next, the tool resumes tracking the cataract-extraction trajectory.

Figure 6.2: Shown are snapshots of a real-time intraoperative intervention with accompanying OCT B-scans. The surgical instrument is commanded to remove a bubble near the upper-right ($x = -2,\ y = 9$) of the pupil.

6.3 Semi-Automated Cataract Removal on an Animal Model

OCT volume scan focusing at the lens bag is a deterministic evidence of the surgical progress and the integrity of posterior capsule. As shown in Fig. 6.3, when the lens materials are completely removed from the lens bag without rupturing the posterior capsule, the OCT volume scan is clear and the image of the posterior capsule presents a convex shape in the volume scan and circular shape in its xy-slice. Otherwise, the OCT image of posterior capsule appears an irregular shape since the pressure of anterior segment cannot be regulated once the posterior capsule is broken. By examining the OCT volume scan intraoperatively and postoperatively, the surgical progress can be monitored and analyzed.

The surgical progress of an example surgery (Eye #1) is shown in Fig. 6.4. After two minutes of tracking, a majority of lens remains in the anterior segment (Fig. 6.4a). The OCT volume scan indicates a second, deeper extraction cycle is required. After four minutes of tracking (Fig. 6.4b), most of the lens material has been removed and only a small piece of lens material remains attached to the posterior capsule. At this state, a third cycle would be less efficient than directly targeting the piece; therefore, the I/A tool is commanded to this position and to aspirate the remaining material. A postoperative OCT volume scan (Fig. 6.4c) reveals that no lens material remains in the anterior segment. Based on the convex shape of the posterior capsule, it is clear that the capsule remains intact.

More experimental OCT data is given from Fig. 6.5 to Fig. 6.9. Similar to the progress observed in Fig. 6.4, the postoperative volume scan of every each eye shows a convex shape for the posterior capsule since it remained intact through the entire operation. Since the surgical duration varies, some of eye was completed within two minutes and therefore only one volume scan is available (Eye #3 and #4). Note that although there is no lens material appears in the postoperative OCT scan of Eye #4 , this case was recognized as incomplete since a small piece of lens material was found behind the iris after the histologic examination.

The statistical data of the evaluated metrics is presented in Fig. 6.10. The histogram of completion time is demonstrated in Fig. 6.10. Among the 30 eyes, iris was touched during

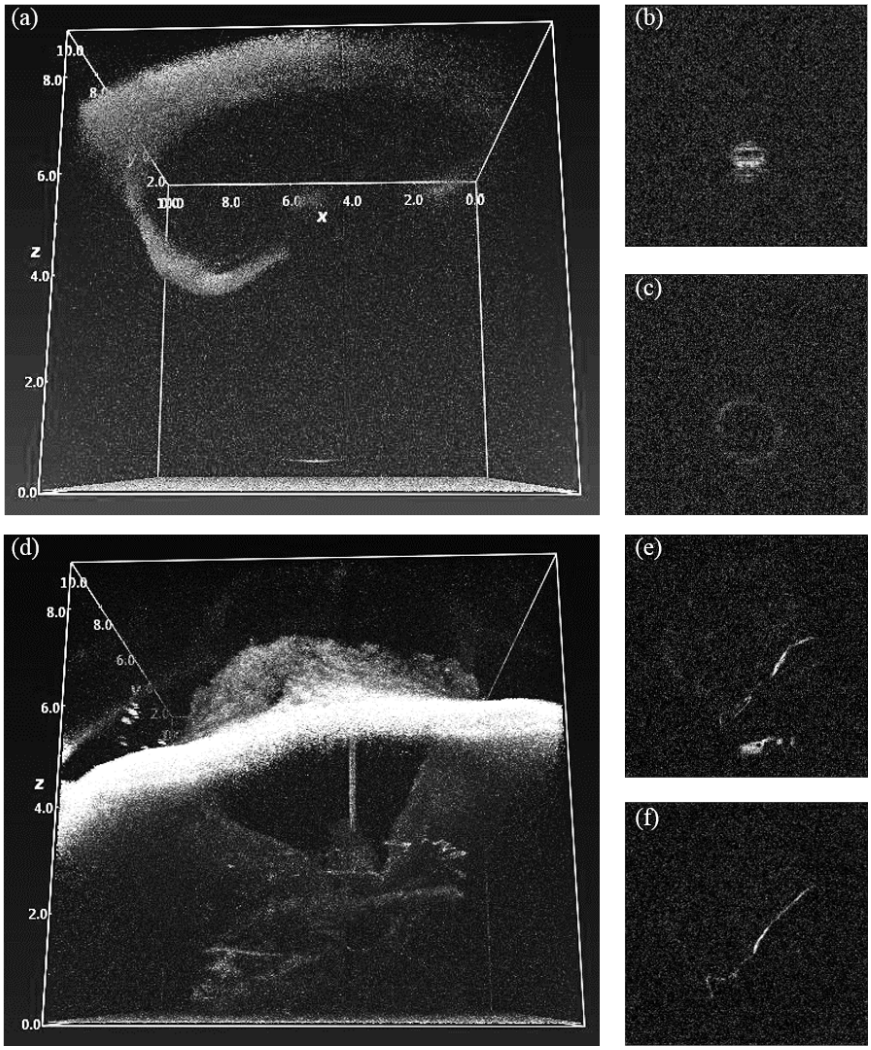
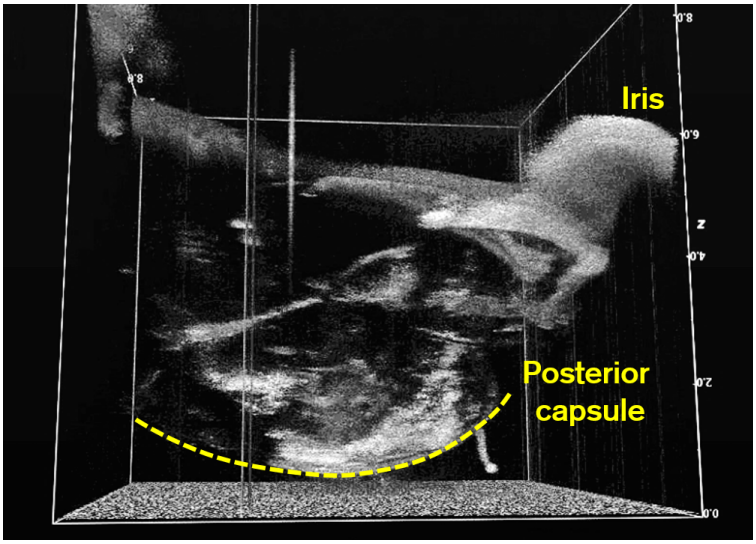
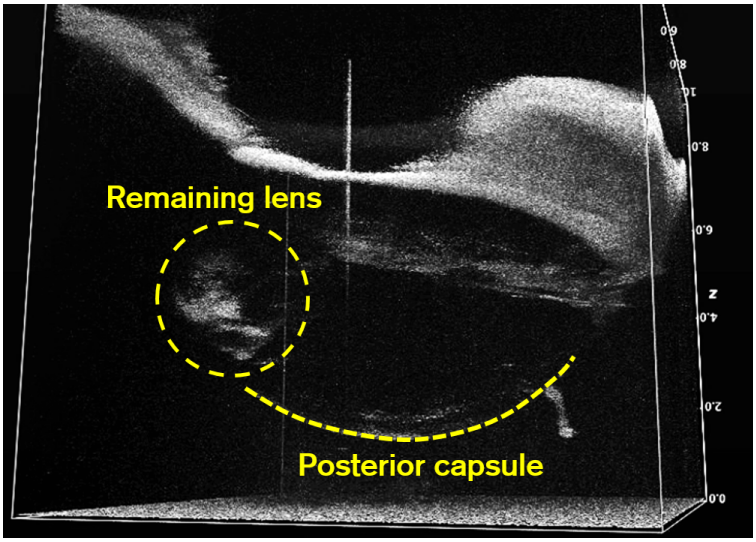


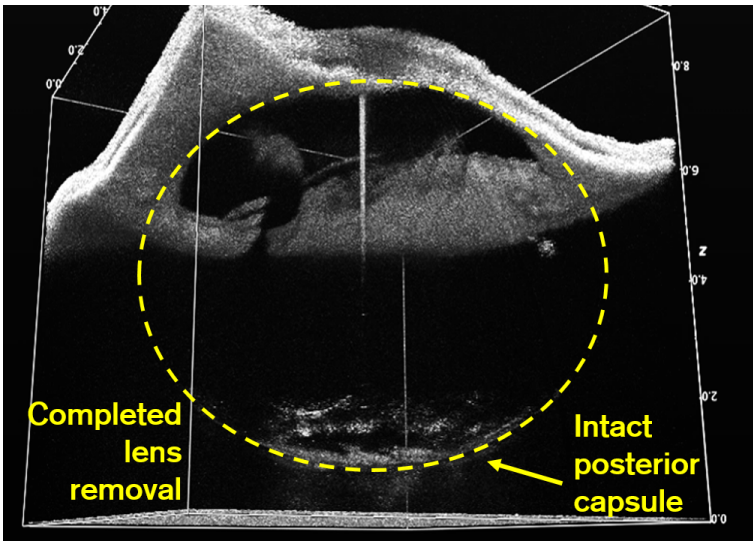
Figure 6.3: Two examples of OCT volume scans focusing at the lens bag and posterior capsule. (a) The lens materials were completely removed from the lens bag and the posterior capsule (the convex shape at the bottom of the image) was intact. (b) The xy-slice of (a) at $z = 0.55$ mm. (c) The xy-slice of (a) at $z = 0.65$ mm. The circular shape in (b) and (c) confirms the integrity of the capsule. (d) The lens materials were completely removed but the posterior capsule (the irregular shape at the bottom of the image) was ruptured. (e) The xy-slice of (d) at $z = 0.92$ mm. (f) The xy-slice of (d) at $z = 1.32$ mm. Since the capsule was ruptured and folded, there is no longer a circular pattern as shown in (b) and (c).



(a) After two minutes of operation, an intermittent volume scan reveals an intact posterior capsule and a majority of lens material remaining.



(b) After four minutes of operation, only a small piece of lens material remains.



(c) This postoperatively volume scan shows an intact posterior capsule and complete removal of lens removal.

Figure 6.4: Shown are three examples of OCT volume scans for used for intraoperative and postoperative evaulation – Eye #1 .

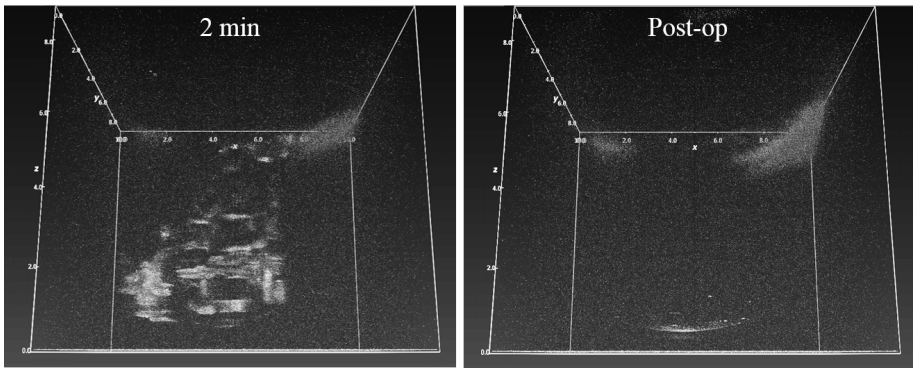


Figure 6.5: Intraoperative and postoperative OCT volume scans – Eye #2 . Left: the intraoperative scan after two minutes. Massive amount of lens material still remained in the lens bag. Right: the postoperative scan. The lens was completely removed and the posterior capsule was intact.

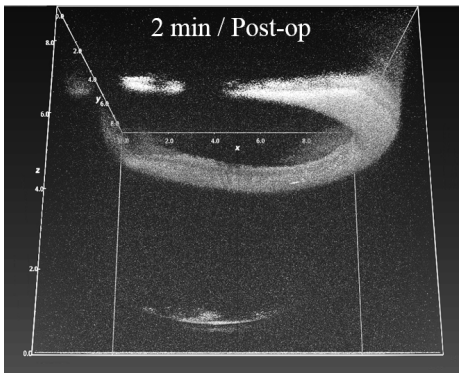


Figure 6.6: Intraoperative and postoperative OCT volume scans – Eye #3 . The lens removal was finished within two minutes without posterior capsule rupture.

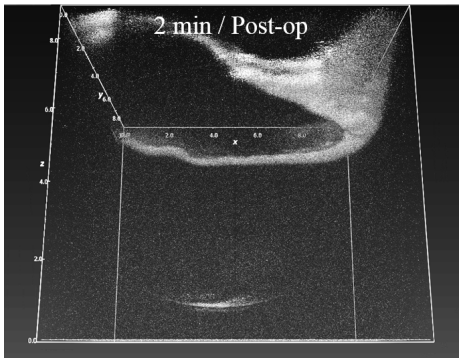


Figure 6.7: Intraoperative and postoperative OCT volume scans – Eye #4 . Note that a small piece of lens material was found behind the iris after the histologic examination.

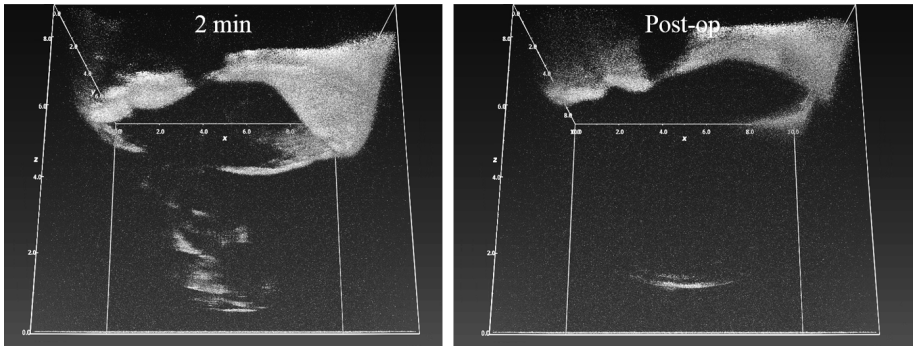


Figure 6.8: Intraoperative and postoperative OCT volume scans – Eye #5 . The surgical progress was similar to Eye #2 .

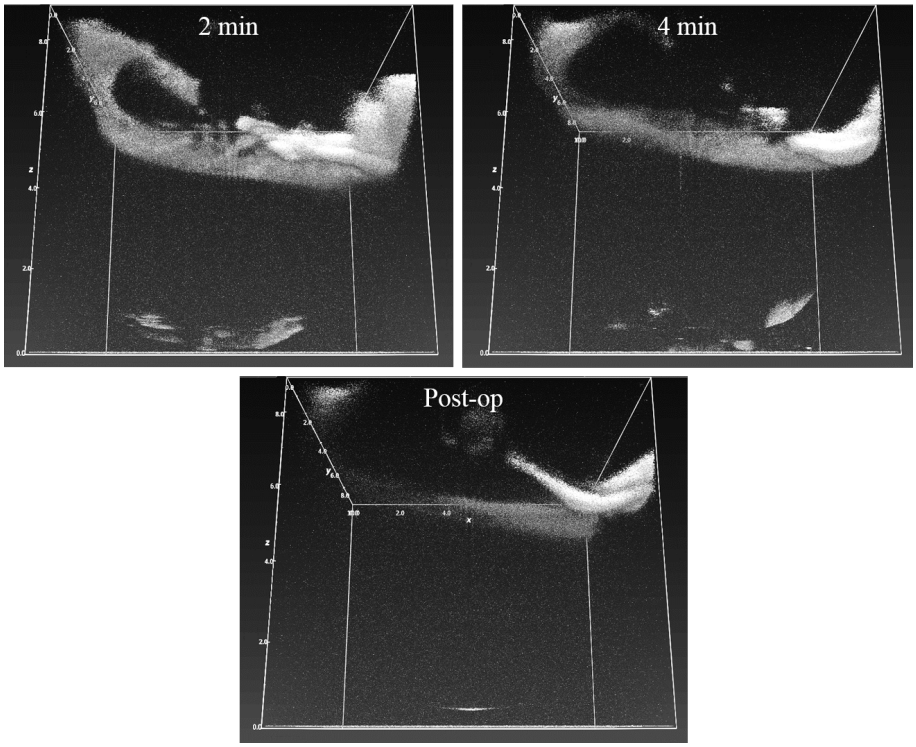


Figure 6.9: Intraoperative and postoperative OCT volume scans – Eye #6 . The surgical progress was similar to Eye #1 .

tracking the cataract-extraction trajectory in 9 cases. This is because the eye was slightly moving when the surgical instrument was operating. This complication could be mitigated by implementing eye tracking or deploying a larger safety gap to the iris. The damage to the cornea was expected due to dehydration and a degradation in freshness in the pig eyes which were shipped overnight from the slaughterhouse. The cornea damage was proportional to the surgical duration (the averaged surgical duration of the eyes with cornea damage score of 1 is 220.6 s; 333.5 s for the eyes with cornea damage score of 2), since the eye was exposed to air and started dehydrating once it was placed on the eye holder. The instrument never touched the cornea during the tests and therefore was not a source of damage. The cornea damage due to dehydration and degradation will not be a problem when performing *in vivo* experiments. Last, the incision stress was well regulated contributed by the automated alignment procedure in almost every eye.

In summary, posterior capsule rupture was prevented in all 30 trials. Complete lens extraction (100%) was achieved on 25 of the samples. In all five cases where 100% extraction was not achieved, minute particles of lens material were discovered post-surgery hidden behind or attached to the iris or posterior capsule where the OCT was unable to image. Therefore, the imperfect success rate was due to limitations of the sensing technology and not the system itself. The average completion time was less than 5 minutes (277±42 s). The iris damage score was 0.33±0.20, cornea damage score was 1.47±0.20, and incision stress score of 0.97±0.11.

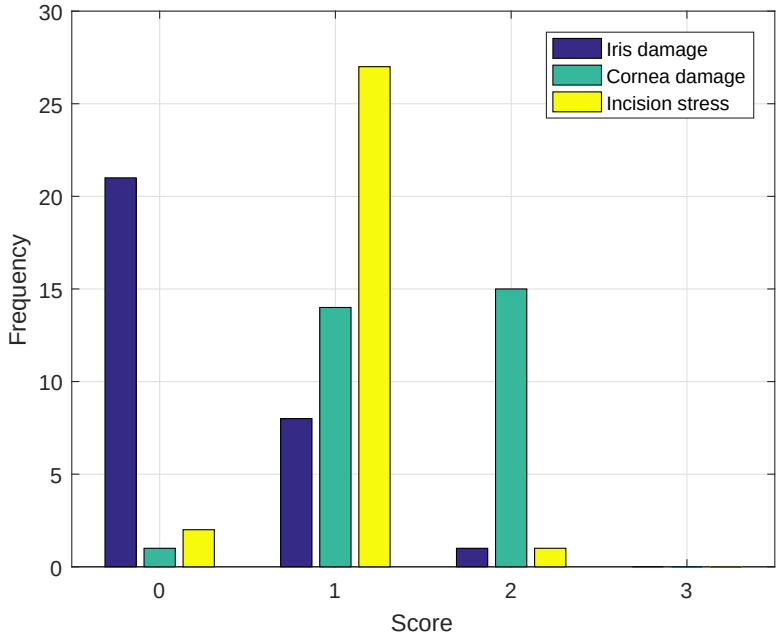


Figure 6.10: The histologic examination result of the 30 pig eye experiment.

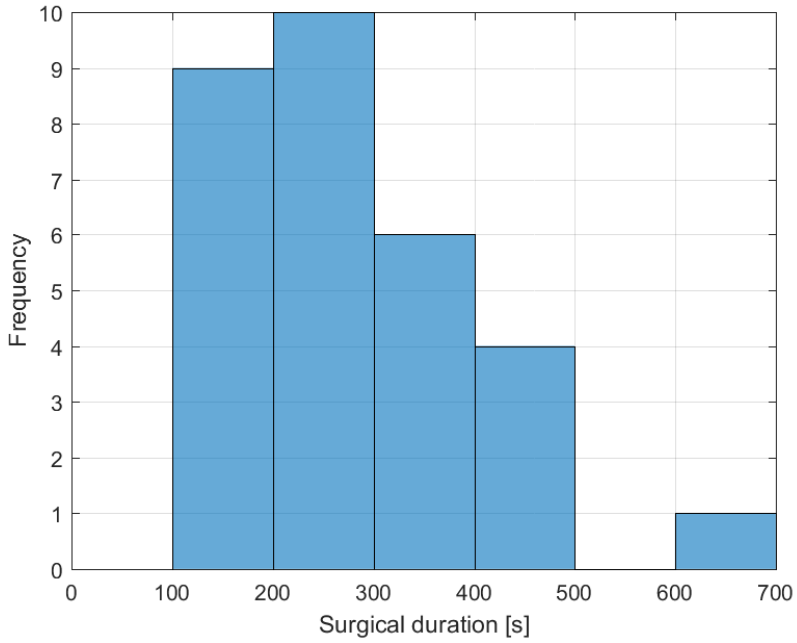


Figure 6.11: The surgical duration of the 30 pig eye experiment.

CHAPTER 7

Conclusions

To avoid common complications and to improve surgical outcomes in intraocular surgery, a robotic surgical platform integrated with OCT imaging, automation methodology, high-precision control, and intervention interface is developed in this thesis. Using OCT scan data, both preoperative planning and intraoperative intervention for the robotic system—the IRISS—was demonstrated. Laser-assisted alignment of the mechanical RCM to the corneal incision reduced the stress on the corneal incision and reduced corneal leakage. The eye anatomy was segmented through parametric modeling, within which a tool-tip trajectory was generated while preserving a safety margin with anatomical bounds.

To account for the changing surgical environment, various strategies for monitoring and intervening the intraocular surgery were adopted. For example, the surgeon can override the robot motion and assign a specific target position for the tool tip. The proposed strategy for intraoperative targeting was validated by aspirating an air bubble from the anterior chamber. In addition, the surgical progress can be monitored by intermittent OCT volume scans.

For high-precision tracking of the preoperative planned tool-tip trajectory, a data-based approach to identify the dynamic inversion utilizing learning-type control algorithms is proposed. The convergence rate of the iterative identification process is also improved by the proposed progressive and iterative learning control scheme. Without using the knowledge of plant dynamics, the identified inversion filter is applied to feedforward control which minimizes the tracking error of the desired reference trajectory on both SISO and MIMO system, including the IRISS robot manipulator.

The integrated system was evaluated by performing the automated cataract-extraction

procedure on 30 post-mortem pig eyes. Only in five cases did the system leave small particles of lens material due to the inability of the OCT to sense them. In addition, there was no posterior capsule rupture among all 30 eyes tested.

Recommendations for future research directions include

1. Implementing an OCT-based tissue-detection algorithm is desirable to facilitate the automation of intraoperative intervention. The information should be used to update the workspace and adjust the navigation strategy to prevent tissue damage and improve surgical efficacy.

2. Including another imaging modality which can visualize the lens equator and can detect small piece of lens behind the iris will improve the completion rate of lens removal.

3. Regulating the intraocular pressure (IOP) via active irrigation control will stabilize the intraocular tissues and therefore reduce the risk of surgical complication.

4. Integrating an eye tracking algorithm is helpful for maintaining the alignment between the RCM and the surgical incision. The XYZ transnational stage will be adjusted according to the eye motion, which is inevitable when the patient is not unconscious during the surgery.

5. Introducing another surgical instrument which collaboratively assists the I/A tool for lens extraction will significantly improve the surgical efficiency and therefore reduce the surgical duration.

6. Extending the proposed inversion identification method to nonlinear system is appealing especially when coupling effect and gravitational force are not negligible to the motion actuation of the robot manipulator.

References

[Abb] Abbott Medical Optics. "CATALYS Precision Laser System." Last accessed: 10/17/2016.

[AHS12] David Allen, Maged Habib, and David Steel. "Final incision size after implantation of a hydrophobic acrylic aspheric intraocular lens: new motorized injector versus standard manual injector." *J. of Cataract & Refractive Surgery*, **38**(2):249–255, 2012.

[Alca] Alcon. "ACCURUS® Surgical System." Last accessed: 11/16/2017.

[Alcb] Alcon. "LenSx Laser." Last accessed: 10/17/2016.

[BA08] K. L. Barton and A. G. Alleyne. "A cross-coupled iterative learning control design for precision motion control." *IEEE Trans. on Control Systems Technology*, **16**(6):1218–1231, 2008.

[BBB16] L. Blanken, F. Boeren, D. Bruijnen, and T. Oomen. "Rational iterative feedforward tuning: Approaches, stable inversion, and experimental comparison." In *American Control Conference (ACC)*, pp. 2629–2634, 2016.

[BKC92] B. Beliczynski, I. Kale, and G. D. Cain. "Approximation of FIR by IIR digital filters: An algorithm based on balanced model reduction." *IEEE/ASME Trans. on Mechatronics*, **40**(3):532–542, 1992.

[BKL08] M. Butcher, A. Karimi, and R. Longchamp. "Iterative Learning Control based on Stochastic Approximation." **41**(6):1478–1483, 2008.

[BO15a] J. Bolder and T. Oomen. "Data-driven optimal ILC for multivariable systems: Removing the need for L and Q filter design." In *American Control Conference (ACC)*, pp. 3546–3551, 2015.

[BO15b] J. Bolder and T. Oomen. "Rational basis functions in iterative learning control - with experimental verification on a motion system." *IEEE Trans. on Control Systems Technology*, **23**(2):722–729, 2015.

[BOS15] F. Boeren, T. Oomen, and M. Steinbuch. "Iterative motion feedforward tuning: A data-driven approach based on instrumental variable identification." *Control Engineering Practice*, **37**:11–19, 2015.

[BTA06] D. A. Bristow, M. Tharayil, and A. G. Alleyne. "A survey of iterative learning control." *Control Systems, IEEE*, **26**(3):96–114, 2006.

[CCT17] C. W. Chen, Y. C. Chang, and T. C. Tsao. "Dynamic Trajectory Tracking by Synergistic Dual-Stage Actuation and Control." *IEEE/ASME Trans. on Mechatronics*, **22**(6):2600–2610, 2017.

[CGT17] Gyeong Woo Cheon, Berk Gonec, Russell H Taylor, Peter L Gehlbach, and Jin U Kang. "Motorized Micro-Forceps with Active Motion Guidance based on Common-Path SSOCT for Epiretinal Membranectomy." *IEEE/ASME Trans. on Mechatronics*, 2017.

[CHC15] Gyeong Woo Cheon, Yong Huang, Jaepyeng Cha, Peter L Gehlbach, and Jin U Kang. "Accurate real-time depth control for CP-SSOCT distal sensor based handheld microsurgery tools." *Biomedical optics express*, **6**(5):1942–1953, 2015.

[CLG18] Cheng-Wei Chen, Yu-Hsiu Lee, Matthew J Gerber, Harrison Cheng, Yan-Chao Yang, Govetto Andrea, Francone Anibal Andres, Stefano Soatto, Warren S. Grundfest, Jean-Pierre Hubschman, and Tsu-Chin Tsao. "Intraocular Robotic Interventional Surgical System (IRISS): Semi-Automated OCT-Guided Cataract Removal (submitted)." *The International J. of Medical Robotics and Computer Assisted Surgery*, 2018.

[CM02] Y. Q. Chen and K. L. Moore. "An optimal design of pd-type iterative learning control with monotonic convergence." In *the 2002 IEEE International Symposium on Intelligent Control*, pp. 55–60, 2002.

[CQL04] I. Chin, S. J. Qin, K. S. Lee, and M. Cho. "A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection." *Automatica*, **40**(11):1913–1922, 2004.

[CRT18] Cheng-Wei Chen, Sandeep Rai, and Tsu-Chin Tsao. "Feedforward Control by Iterative Inversion Identification Using Filtered Impulse Model (in preparation)." 2018.

[CSS08] M. C. Campi, T. Sugie, and F. Saka. "An iterative identification method for linear continuous-time systems." *IEEE Trans. on automatic control*, **53**(7):1661–1669, 2008.

[CT14] H. L. Chang and T. Tsao. "High-sampling rate dynamic inversion filter realization and applications in digital control." *IEEE/ASME Trans. on Mechatronics*, **19**(1):238–248, 2014.

[CT16] Cheng-Wei Chen and Tsu-Chin Tsao. "Data-based feedforward controller reconstruction from iterative learning control algorithm." In *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*, pp. 683–688. IEEE, 2016.

[CT17] Cheng-Wei Chen and Tsu-Chin Tsao. "Data-Driven Progressive and Iterative Learning Control." *IFAC-PapersOnLine*, **50**(1):4825–4830, 2017.

[CVK17] Oscar M Carrasco-Zevallos, Christian Viehland, Brenton Keller, Mark Draelos, Anthony N Kuo, Cynthia A Toth, and Joseph A Izatt. "Review of intraoperative optical coherence tomography: technology and applications." *Biomedical Optics Express*, **8**(3):1607–1637, 2017.

[DBC13] Kendall E Donaldson, Rosa Braga-Mele, Florence Cabot, Richard Davidson, Deepinder K Dhaliwal, Rex Hamilton, Mitchell Jackson, Larry Patterson, Karl Stonecipher, Sonia H Yoo, et al. "Femtosecond laser–assisted cataract surgery." *J. of Cataract & Refractive Surgery*, **39**(11):1753–1763, 2013.

[DMR99] Parul Desai, DC Minassian, and Angela Reidy. "National cataract surgery survey 1997–8: a report of the results of the clinical outcomes." *British J. of Ophthalmology*, **83**(12):1336–1340, 1999.

[DYO17] M. Duan, D. Yoon, and C. E. Okwudire. "'A limited-preview filtered B-spline approach to tracking control With application to vibration-induced error compensation of a 3D printer." *Mechatronics*, 2017.

[EPH10] Marieh Esmaeelpour, Boris Považay, Boris Hermann, Bernd Hofer, Vedran Kajic, Ketan Kapoor, Nik JL Sheen, Rachel V North, and Wolfgang Drexler. "Three-dimensional 1060-nm OCT: choroidal thickness maps in normal subjects and improved posterior segment visualization in cataract patients." *Investigative ophthalmology & visual science*, **51**(10):5260–5266, 2010.

[ESF14] Justis P Ehlers, Sunil K Srivastava, Daniel Feiler, Amanda I Noonan, Andrew M Rollins, and Yuankai K Tao. "Integrative advances for OCT-guided ophthalmic surgery and intraoperative OCT: microscope integration, surgical instrumentation, and heads-up display surgeon feedback." *PLoS One*, **9**(8):e105224, 2014.

[ET15] Mohamed T El-Haddad and Yuankai K Tao. "Automated stereo vision instrument tracking for intraoperative OCT guided anterior segment ophthalmic surgical maneuvers." *Biomedical optics express*, **6**(8):3014–3031, 2015.

[FF12] Laura Fanea and Andrew J Fagan. "Magnetic resonance imaging techniques in ophthalmology." *Molecular vision*, **18**:2538, 2012.

[FLM17] Z. Feng, J. Ling, M. Ming, and X. Xiao. "A model-data integrated iterative learning controller for flexible tracking with application to a piezo nanopositioner." *Trans. of the Institute of Measurement and Control*, p. 0142331217719958, 2017.

[FLR07] C. Freeman, P. Lewin, and E. Rogers. "Robust monotone gradient-based discrete-time iterative learning control." *Int'l Journal of Control*, **80**:569–582, 2007.

[FRB15] C. T. Freeman, E. Rogers, J. H. Burridge, A.-M. Hughes, and K. L. Meadmore. *Iterative Learning Control for Electrical Stimulation and Stroke Rehabilitation.* Springer-Verlag London, 2015.

[FS16] James Fujimoto and Eric Swanson. "The Development, Commercialization, and Impact of Optical Coherence TomographyHistory of Optical Coherence Tomography." *Investigative ophthalmology & visual science*, **57**(9):OCT1–OCT13, 2016.

[GD04] John C Gower and Garmt B Dijksterhuis. *Procrustes problems*, volume 30. Oxford University Press on Demand, 2004.

[Gus96] Fredrik Gustafsson. "Determining the initial states in forward-backward filtering." *IEEE Transactions on Signal Processing*, **44**(4):988–992, 1996.

[HS01] K. Hamamoto and T. Sugie. "An iterative learning control algorithm within prescribed input-output subspace." *Automatica*, **37**(11):1803–1809, 2001.

[INK81] T. K. S. M. T. Inoue, M. Nakano, T. Kubo, S. Matsumoto, and H. Baba. "High accuracy control of a proton synchrotron magnet power supply." *IFAC Proceedings Volumes*, **14**(2):3137–3142, 1981.

[JPS13] P. Janssens, G. Pipeleers, and J. Swevers. "A data-driven constrained norm-optimal iterative learning control framework for LTI systems." *IEEE Trans. on Control Systems Technology*, **21**(2):546–551, 2013.

[KSA02] Koji Kinosita, Takuya Sogo, and Norihiko Adachi. "Iterative learning control using adjoint systems and stable inversion." *Asian Journal of Control*, **4**(1):60–67, 2002.

[KT04] K. Krishnamoorthy and T. Tsao. "Repetitive learning control for precision machining of complex profiles." In *ASME 2004 International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers*, pp. 63–69, 2004.

[KT14] K. Krishnamoorthy and T. Tsao. "Design and implementation of repetitive control based noncausal zero-phase iterative learning control." *arXiv preprint arXiv:1408.2490*, 2014.

[KZ13] K.-S. Kim and Q. Zou. "A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems." *IEEE/ASME Trans. on Mechatronics*, **18**(6):1767–1777, 2013.

[LA14] N. Liu and A. G. Alleyne. "Iterative learning identification applied to automated off-highway vehicle." *IEEE Trans. on Control Systems Technology*, **22**(1):331–337, 2014.

[LEN] LENSAR, LLC. "The LENSAR Laser System." Last accessed: 10/17/2016.

[Lon00] Richard W Longman. "Iterative learning control and repetitive control for engineering practice." *International journal of control*, **73**(10):930–954, 2000.

[MTB08] S. van der Meulen, R. Tousain, and O. Bosgra. "Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer." *J. Dyn. Sys. Meas. Control*, **130**(5):1–16, 2008.

[Nat] National Eye Institute. "Facts About Cataract." Last accessed: 11/16/2017.

[NBA15] Manikantan Nambi, Paul S Bernstein, and Jake J Abbott. "A compact retinal-surgery telemanipulator that uses disposable instruments." In *International conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 258–265. Springer, 2015.

[NG02] M. Norrlöf and S. Gunnarsson. "Time and frequency domain convergence properties in iterative learning control." *International Journal of Control*, **75**(14):1114–1126, 2002.

[NML17] MA Nasseri, M Maier, and CP Lohmann. "A targeted drug delivery platform for assisting retinal surgeons for treating Age-related Macular Degeneration (AMD)." In *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International conf. of the IEEE*, pp. 4333–4338. IEEE, 2017.

[OBC17] Randall J Olson, Rosa Braga-Mele, Sherleen Huang Chen, Kevin M Miller, Roberto Pineda, James P Tweeten, and David C Musch. "Cataract in the Adult Eye Preferred Practice Pattern®." *Ophthalmology*, **124**(2):1–119, 2017.

[OHD09] D. H. Owens, J. J. Hatonen, and S. Daley. "Robust monotone gradientbased discretetime iterative learning control." *International Journal of Robust and Nonlinear Control*, **19**(6):634–661, 2009.

[PM12] Donatella Pascolini and Silvio Paolo Mariotti. "Global estimates of visual impairment: 2010." *British J. of Ophthalmology*, **96**(5):614–618, 2012.

[PSF11] Daniel Palanker, Georg Schuele, Neil Friedman, Dan Andersen, and William Culbertson. "Cataract Surgery with OCT-guided Femtosecond Laser." In *Bio-Optics: Design and Application*, p. BTuC4. Optical Society of America, 2011.

[PW04] Benjamin Potsaid and John T. Wen. "High performance motion tracking control." In *the 2004 IEEE International Conference on Control Applications*, pp. 719–723, 2004.

[RCS16] S. Rai, G. Cavalier, J. Simonelli, and T. C. Tsao. "MIMO Repetitive Control of an Active Magnetic Bearing Spindle." *IFAC-PapersOnLine*, **49**(21):192–199, 2016.

[RHL08] James D. Ratcliffe, Jari J. Hatonen, Paul L. Lewin, Eric Rogers, and D. H. Owens. "Robustness analysis of an adjoint optimal iterative learning controller with experimental verification." *International Journal of Robust and Nonlinear Control*, **18**(10):1089–1113, 2008.

[RPL09] B. P. Rigney, L. Y. Pao, and D. A. Lawrence. "Nonminimum phase dynamic inversion for settle time applications." *IEEE Trans. on Control Systems Technology*, **17**(5):989–1005, 2009.

[RRR01] Sunita Radhakrishnan, Andrew M Rollins, Jonathan E Roth, Siavash Yazdanfar, Volker Westphal, David S Bardenstein, and Joseph A Izatt. "Real-time optical coherence tomography of the anterior segment at 1310 nm." *Archives of ophthalmology*, **119**(8):1179–1185, 2001.

[SFT09] Hoday Stearns, Benjamin Fine, and Masayoshi Tomizuka. "Iterative Identification of Feedforward Controllers for Iterative Learning Control." *IFAC Proceedings Volumes*, **42**(16):203–208, 2009.

[Sil09] Ronald H Silverman. "High-resolution ultrasound imaging of the eye–a review." *Clinical & experimental ophthalmology*, **37**(1):54–67, 2009.

[SMU11] Irene Sanchez, Raul Martin, Fernando Ussa, and Ivan Fernandez-Bueno. "The parameters of the porcine eyeball." *Graefe's Archive for Clinical and Experimental Ophthalmology*, **249**(4):475–482, 2011.

[Tay04] A. Tayebi. "Adaptive iterative learning control for robot manipulators." *Automatica*, **40**(7):1195–1203, 2004.

[THI15] Shinichi Tanaka, Kanako Harada, Yoshiki Ida, Kyohei Tomita, Ippei Kato, Fumihito Arai, Takashi Ueta, Yasuo Noda, Naohiko Sugita, and Mamoru Mitsuishi. "Quantitative assessment of manual and robotic microcannulation for eye surgery using new eye model." *The International J. of Medical Robotics and Computer Assisted Surgery*, **11**(2):210–217, 2015.

[Tom87] M. Tomizuka. "Zero phase error tracking algorithm for digital control." *Journal of Dynamic Systems, Measurement, and Control*, **109**(1):65–68, 1987.

[TT87] T. Tsao and M. Tomizuka. "Adaptive zero phase error tracking algorithm for digital control." *Journal of Dynamic Systems, Measurement, and Control*, **109**(4):349–354, 1987.

[TT15] K. T. Teng and T. Tsao. "A comparison of inversion based Iterative Learning Control Algorithms." In *American Control Conference (ACC)*, pp. 3564–3569, 2015.

[TTC89] Masayoshi Tomizuka, Tsu-Chin Tsao, and Kok-Kia Chew. "Analysis and synthesis of discrete-time repetitive controllers." *Journal of Dynamic Systems, Measurement, and Control*, **111**(3):353–358, 1989.

[UBH10] Ali Üneri, Marcin A Balicki, James Handa, Peter Gehlbach, Russell H Taylor, and Iulian Iordachita. "New steady-hand eye robot with micro-force sensing for vitreoretinal surgery." In *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International conf. on*, pp. 814–819. IEEE, 2010.

[UYS09] Takashi Ueta, Yoshiharu Yamaguchi, Yoshihiro Shirakawa, Taiga Nakano, Ryuichi Ideta, Yasuo Noda, Akio Morita, Ryo Mochizuki, Naohiko Sugita, Mamoru Mitsuishi, et al. "Robot-assisted vitreoretinal surgery: Development of a prototype

and feasibility studies in an animal model." *Ophthalmology*, **116**(8):1538–1543, 2009.

[WB10] J. van de Wijdeven and O. Bosgra. "Using basis functions in iterative learning control: analysis and design theory." *International Journal of Control*, **83**(4):661–675, 2010.

[WGD09] Y. Wang, F. Gao, and F. J. Doyle. "Survey on iterative learning control, repetitive control, and run-to-run control." *Journal of Process Control*, **19**(10):1589–1600, 2009.

[WGP18] Jason T Wilson, Matthew J Gerber, Stephen W Prince, Cheng-Wei Chen, Steven D Schwartz, Jean-Pierre Hubschman, and Tsu-Chin Tsao. "Intraocular robotic interventional surgical system (IRISS): Mechanical design, evaluation, and master–slave manipulation." *The International J. of Medical Robotics and Computer Assisted Surgery*, **14**(1):e1841, 2018.

[WNG11] J. Wallen, M. Norrlöf, and S. Gunnarsson. "A framework for analysis of observer-based ILC." *Asian J. of Control*, **13**(1):3–14, 2011.

[WYZ14] Danwei Wang, Yongqiang Ye, and Bin Zhang. "Reverse Time Filtering Based ILC." In *Practical Iterative Learning Control with Frequency Domain Design and Sampled Data Implementation*, pp. 75–102. Springer, 2014.

[WZB13] Dominic Williams, Yalin Zheng, Fangjun Bao, and Ahmed Elsheikh. "Automatic segmentation of anterior segment optical coherence tomography images." *J. of biomedical optics*, **18**(5):056003–056003, 2013.

[YHP17] T. Yamaguchi, M. Hirata, and J. C. K. Pang. *High-speed precision motion control.* CRC press, 2017.

[YSJ13] Haoran Yu, Jin-Hui Shen, Karen M Joos, and Nabil Simaan. "Design, calibration and preliminary testing of a robotic telemanipulator for OCT guided retinal surgery." In *Robotics and Automation (ICRA), 2013 IEEE International conf. on*, pp. 225–231. IEEE, 2013.

[YSJ16] Haoran Yu, Jin-Hui Shen, Karen M Joos, and Nabil Simaan. "Calibration and Integration of B-Mode Optical Coherence Tomography for Assistive Control in Robotic Microsurgery." *IEEE/ASME Trans. on Mechatronics*, **21**(6):2613–2623, 2016.

[YSS15] Haoran Yu, Jin-Hui Shen, Rohan J Shah, Nabil Simaan, and Karen M Joos. "Evaluation of microsurgical tasks with OCT-guided and/or robot-assisted ophthalmic forceps." *Biomedical optics express*, **6**(2):457–472, 2015.

[YW05] Yongqiang Ye and Danwei Wang. "Zero phase learning control using reversed time input runs." *Journal of Dynamic Systems, Measurement, and Control(Transactions of the ASME)*, **127**(1):133–139, 2005.