

Developing a Two-Arm Robot-Assisted System for Arthroscopic Surgery

by

Teng Li

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Biomedical Engineering

Department of Electrical and Computer Engineering

Faculty of Engineering

University of Alberta

© Teng Li, 2024

Abstract

Robot-assisted arthroscopic surgery has been receiving growing attention in the field of orthopedic surgery. Most of the existing robot-assisted surgical systems in orthopedics take more focus on open surgery than minimally invasive surgery (MIS). In traditional arthroscopic surgery, a specific type of MIS, the surgeon needs to hold an arthroscope with one hand while performing the surgical operations with the other hand, which can restrict the dexterity of the surgical performance and increase the cognitive load. On the other hand, the surgeon heavily relies on the arthroscope view when conducting the surgery whereas the arthroscope view is a largely localized view and lacks depth information. This motivates us to develop a two-arm robotic system, a robot-assisted arthroscope holder, and a robot-assisted surgical tool with haptic feedback, to assist the surgeon in both scenarios.

In a robot-assisted system for arthroscopic surgery, surgical tools attached to the robot end-effector (EE) will affect the robot dynamics inevitably, which could undermine the utility and stability of the robotic system if the dynamic uncertainties (*e.g.*, the mass of the surgical tools) are not identified and compensated for in the robot control system. To solve this problem, an integrated framework of impedance control and nonlinear disturbance observer (NDOB) is proposed, where the former ensures compliant robot behavior and the latter compensates for dynamic uncertainties. By integrating an impedance controller with NDOB, the proposed framework allows an accurate impedance control and stable system when dynamic model inaccuracy and external disturbance exist. However, the NDOB always estimates all of the uncertainties as a lumped term, and it is not able to separate any of the components.

In order to separate some specific components from the dynamic uncertainties, we developed a framework for using a neural network (NN) to learn for some uncertainties thus separating the other uncertainties. The effectiveness and performance of the trained NN model are verified in simulations. However, it is not an efficient approach in practice considering the laborious offline training procedures. Also, the trained NN model is robot-dependent, which means that all of the training process needs to go through again when a different robotic system is employed.

Aiming for a more compact, efficient, and general approach, we developed a gravity iterative learning (Git) scheme with a steady-state scaling strategy specially for gravity compensation. The Git scheme can accurately learn and compensate for gravity when gravity compensation is the main concern. By integrating the Git scheme with an impedance controller, the robot can keep still at a designated pose even with a heavy payload attached to the robot EE. Also, it allows the operator to move it freely via a pedal switch whenever needed.

On the other hand, virtual fixture (VF) has been serving as a vital role in robot-assisted surgeries, such as protecting a beating heart. In orthopedic surgery, preoperative images are often used in the operating room, on which some curves can be drawn to mark out the boundaries for osteophytes to be removed. These curves can be used to generate VF to assist in removing osteophytes during the operation. A challenge is that the hand-drawn curves usually have irregular shapes and cannot be mathematically represented by equations, thus most of the existing VF approaches will not work in this scenario. To this end, a point-based VF-generating algorithm is developed, with which 3D VF can be generated directly from point clouds in any shape including the hand-drawn curves in a preplanned image.

In the end, a prototype of a two-arm robot-assisted system for arthroscopic surgery is built and preliminarily evaluated. The left-arm robot is used as a robot-assisted arthroscope holder, which can hold the arthroscope still at any designated pose, while also allowing the operator to move it around freely via a pedal switch whenever

needed. The left-arm robot is implemented with an impedance controller and the Git scheme, where the former can provide compliant robot behavior thus ensuring a safe human-robot interaction, while the latter can accurately learn and compensate for gravity. The right-arm robot is used as a robot-assisted surgical tool providing VF assistance and haptic feedback during the surgery, and is implemented with the point-based VF algorithm, which can generate VF directly from point clouds in any shape, render haptic force feedback, and deliver it to the operator. Furthermore, the VF, the bone, and the surgical tool with its real-time position are visualized in a 3D virtual environment as additional visual feedback for the operator.

Preface

The ultimate objective of this research project is to develop a two-arm robot-assisted arthroscopic surgical system for assisting surgeons during arthroscopic surgery. This thesis is a product of a multidisciplinary research collaboration, led by Dr. Mahdi Tavakoli (principal investigator) at the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Alberta, Canada, and Dr. Armin Badre, at Western Hand & Upper Limb Facility, Sturgeon Hospital, St. Albert, Canada, and at Division of Orthopaedic Surgery, Department of Surgery, Faculty of Medicine & Dentistry, University of Alberta, Canada. All of the work presented in this thesis was conducted in the Telerobotic and Biorobotic Systems (TBS) Group (Dr. Mahdi Tavakoli’s lab) at the University of Alberta, Canada.

For the work presented in this thesis, I was responsible for most of the work including the experiment design, data acquisition and analysis, manuscript preparation, and making responses to reviewers. My supervisor, Dr. Mahdi Tavakoli, provided overall guidance in the research, including valuable discussions and suggestions, concept formation, experiment improvement, and manuscript revision. Dr. Armin Badre is a proficient surgeon for traditional arthroscopic surgery, providing support on the background knowledge and the necessary demands for building a robotic system for arthroscopic surgery, as well as feedback and suggestions on designing experiments for mimicking arthroscopic surgery scenarios.

Chapter 1 presents literature reviews on this project, which has been published as two review papers: [1] Teng Li, Armin Badre, Farshid Alambeigi, and Mahdi Tavakoli, “*Robotic Systems and Navigation Techniques in Orthopedics: A Historical Review*”,

Applied Sciences, Section: Robotics and Automation, Special Issue: Surgical Robotics Design and Clinical Applications, 2023, 13(17):9768. [2] Teng Li, Hongjun Xing, Ehsan Hashemi, Hamid D. Taghirad, Mahdi Tavakoli, “*A Brief Survey of Observers for Disturbance Estimation and Compensation*”, Robotica, 2023, 41(12), 3818–3845.

Chapter 2 has been published as: Teng Li, Armin Badre, Hamid D. Taghirad, Mahdi Tavakoli, “*Integrating Impedance Control and Nonlinear Disturbance Observer for Robot-Assisted Arthroscope Control in Elbow Arthroscopic Surgery*”, 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022), October 23-27, Kyoto, Japan, 2022, pp. 11172-11179.

Chapter 3 has been published as: Teng Li, Armin Badre, Hamid D. Taghirad, Mahdi Tavakoli, “*Neural Network Learning of Robot Dynamic Uncertainties and Observer-based External Disturbance Estimation for Impedance Control*”, 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2023), June 28-30, Seattle, WA, USA, 2023, pp. 591-597.

Chapter 4 has been published as: Teng Li, Amir Zakerimanesh, Yafei Ou, Armin Badre, and Mahdi Tavakoli, “*Iterative Learning for Gravity Compensation in Impedance Control*”, IEEE/ASME Transactions on Mechatronics, 2024.

Chapter 5 has been published as: Teng Li, Armin Badre, Hamid D. Taghirad, Mahdi Tavakoli, “*Point-Based 3D Virtual Fixture Generating for Image-Guided and Robot-Assisted Surgery in Orthopedics*”, 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2023), June 28-30, Seattle, WA, USA, 2023, pp. 179-186.

Chapter 6 has been submitted as: Teng Li, Armin Badre, and Mahdi Tavakoli, “*Robotic Assistance and Haptic Feedback in Arthroscopic Procedures: Design and Preliminary Evaluation of a Two-Arm System*”, Journal of Medical Robotics Research (JMRR). [Under Review]

Appendix A has been published as: Teng Li, Hongjun Xing, Hamid D. Taghirad, Mahdi Tavakoli, “*EMG-based Hybrid Impedance-Force Control for Human-Robot*

Collaboration on Ultrasound Imaging”, 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022), October 23-27, Kyoto, Japan, 2022, pp. 670-675.

Appendix B has been published as: Teng Li, Xiao Meng, and Mahdi Tavakoli, “*Dual Mode pHRI-teleHRI Control System with a Hybrid Admittance-Force Controller for Ultrasound Imaging*”, Sensors, Section: Sensors and Robotics, Special Issue: Sensors Technology for Medical Robotics, 2022, 22(11):4025.

The work of this thesis is supported in part by the Canada Foundation for Innovation (CFI) under grants LOF 28241 and JELF 35916, in part by the Government of Alberta under grants IAE RCP-12-021 and EDT RCP-17-019-SEG, in part by the Government of Alberta’s grant to Centre for Autonomous Systems in Strengthening Future Communities (RCP-19-001-MIF), in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grants RTI-2018-00681, RGPIN-2019-04662, and RGPAS-2019-00106, and in part by the Edmonton Civic Employee Charitable Assistance Fund.

To my parents.

Table of Contents

List of Tables	xiv
List of Figures	xviii
Abbreviations	xix
1 Introduction	1
1.1 Literature Review	1
1.1.1 Robotic systems (RS) in orthopedics	1
1.1.2 Disturbance estimation approaches	12
1.1.3 Virtual fixture techniques	18
1.2 Motivation and Objectives	20
1.3 Contributions	21
1.4 Structure of This Thesis	22
2 Integrating Impedance Control with Nonlinear Disturbance Observer	25
2.1 Background	26
2.2 Methods	29
2.2.1 Robot dynamics and impedance control	29
2.2.2 Nonlinear disturbance observer	31
2.2.3 Integration of impedance controller and NDOB	33
2.3 Simulations, Experiments, and Results	35
2.3.1 Robotic system	35
2.3.2 Parameterization	35
2.3.3 Condition 1: Constant payload	36
2.3.4 Condition 2: Time-varying payload	39
2.3.5 Condition 3: Set-point regulation	42
2.4 Application on Elbow Arthroscopic Surgery	44
2.5 Conclusion	46

3	Neural Network Learning on Robot Dynamic Uncertainties	48
3.1	Background	49
3.2	Methods	51
3.2.1	Robot dynamics and impedance control	51
3.2.2	Friction model	52
3.2.3	Disturbance observer	53
3.2.4	Neural network	54
3.2.5	Collecting training and testing data	55
3.3	Simulations, Validations, and Results	57
3.3.1	Robotic system	57
3.3.2	Comparing observers	58
3.3.3	Comparing NN models	59
3.3.4	The effect of dynamic model error	61
3.3.5	Application on virtual fixture	62
3.3.6	Limitations	64
3.4	Conclusion	65
4	Iterative Learning for Gravity Compensation	66
4.1	Background	67
4.2	Methods	70
4.2.1	Robot dynamics and disturbances	70
4.2.2	Impedance control	71
4.2.3	Iterative learning for gravity compensation	72
4.2.4	Analysis	74
4.2.5	Steady-state scaling strategy	77
4.3	Simulations, Experiments, and Results	78
4.3.1	Robotic system	78
4.3.2	Parameterization	79
4.3.3	Simulation to evaluate steady-state scaling strategy	80
4.3.4	Simulation on setpoint regulation task	82
4.3.5	Simulation on trajectory tracking task	85
4.3.6	Experiment on trajectory tracking task	88
4.3.7	Experiment on setpoint regulation task	90
4.3.8	Comparing with other methods	93
4.4	Conclusion	94

5	Point-Based 3D Virtual Fixture Generating Method	96
5.1	Background	97
5.2	Methods	99
5.2.1	Main algorithm	101
5.2.2	Proxy algorithms	102
5.2.3	Virtual force rendering	105
5.3	Simulations, Experiments, and Results	106
5.3.1	Apparatus	106
5.3.2	Simulations	107
5.3.3	Experiments	108
5.4	Discussions	112
5.5	Conclusion	114
6	A Prototype of a Two-Arm Robotic System for Arthroscopic Surgery	118
6.1	Background	119
6.2	Methods	124
6.2.1	Left-Arm: Robot dynamics and disturbances	124
6.2.2	Left-Arm: Impedance control	126
6.2.3	Left-Arm: Git scheme	126
6.2.4	Right-Arm: Point-based VF algorithm	128
6.2.5	Two-Arm System: Prototype and control diagram	130
6.3	Experimental Evaluation on the Prototype	133
6.3.1	Robotic system of the prototype	133
6.3.2	Parameterization	135
6.3.3	Left-Arm Evaluation: Holding the arthroscope	136
6.3.4	Right-Arm Evaluation: Burring bone with VF	141
6.3.5	Future work	145
6.4	Conclusion	147
7	Conclusions and Future Directions	149
7.1	Conclusions	149
7.2	Future Research Directions	150
	References	152
	Appendix A: EMG-based Hybrid Impedance-Force Control for Ultrasound Imaging¹	166
A.1	Introduction	167

A.2	Methods	169
A.2.1	Impedance control and force control	169
A.2.2	EMG signal acquisition and processing	171
A.2.3	EMG mapping algorithm	172
A.2.4	Desired force and audio feedback design	173
A.3	Experiments and Results	173
A.3.1	Apparatus	173
A.3.2	Experiment 1: Hybrid impedance-force controller	175
A.3.3	Experiment 2: Lifting task	175
A.3.4	Experiment 3: Application	177
A.4	Conclusion	181
 Appendix B: Dual Mode <i>p</i>HRI-<i>tele</i>HRI Control System for Ultrasound		
	Imaging¹	183
B.1	Introduction	184
B.2	Methods	186
B.2.1	Robot dynamics	187
B.2.2	Admittance controller for <i>p</i> HRI mode	187
B.2.3	Mapping algorithm for <i>tele</i> HRI mode	188
B.2.4	Force controller	190
B.2.5	Apparatus	191
B.2.6	Audio feedback and haptic feedback	191
B.3	Experiments and Results	193
B.3.1	Procedures and metrics	193
B.3.2	Experiment 1: AF vs. (AF + HF)	194
B.3.3	Experiment 2: (AF + FC) vs. (AF + FC + HF)	196
B.3.4	Experiment 3: Dual-mode switching	200
B.3.5	Statistical comparison across experiments	202
B.4	Discussions	203
B.5	Conclusion	206
 Appendix C: Kinematics and Dynamics of the 3DOF PHANToM Pre-		
	mium 1.5A Robot	208
C.1	Robotic System	208
C.2	Robot Kinematics	208
C.3	Robot Dynamics	210
C.4	Dynamic Model Linearization	212

List of Tables

1.1	Robotic systems (RSs) in orthopedics.	6
1.2	A summary of the main features of observers.	16
2.1	Parameterization for simulation, experiment, and application of p HRI on elbow arthroscopic surgery simulator.	35
3.1	Parameterization for simulations.	58
4.1	Parameterization for simulations and experiments.	79
4.2	Comparing with methods in literature.	94
5.1	Major parameters and description.	100
6.1	Parameterization for the experiments.	136
A.1	Parameters for the experiments.	174
A.2	Results of Experiment 1 on normal contact force in rectangular and bowl-shaped tissue scenarios.	176
A.3	Results of Experiment 3 with the proposed EMG-based hybrid impedance- force controller.	180
B.1	Comparing task performance under different feedback in two modes when the hybrid controller is not implemented in Experiment 1. . . .	196
B.2	Time percentage for remaining force into desired range under different feedback in two modes when the proposed hybrid controller is not implemented in Experiment 1.	196
B.3	Comparing the task performance on horizontal tissue surface under different feedback in two modes when the proposed hybrid controller is implemented in Experiment 2a.	198
B.4	Comparing the task performance on slope tissue surface under different feedback in two modes when the proposed hybrid controller is imple- mented in Experiment 2b.	200

B.5	Comparing the general performance of the proposed dual mode <i>p</i> HRI- <i>tele</i> HRI system when mode switching is involved in Experiment 3. . .	202
B.6	Summary of <i>p</i> -values of t-test results.	203
C.1	DH table for the 3DOF Phantom Premium 1.5A robot	209
C.2	Dynamic parameter values for the 3DOF robot	213
C.3	The identified dynamic coefficients in the linear model of the 3DOF robot.	215

List of Figures

1.1	A hierarchical flowchart for the robotic systems and navigation techniques in orthopedics.	4
1.2	A general flowchart for orthopedic surgeries.	4
1.3	Robotic systems (RS) in the timeline.	7
1.4	Structure of this thesis.	23
2.1	Block diagram of the proposed control scheme with integrating impedance controller and nonlinear disturbance observer (NDOB).	29
2.2	Simulation results of the four cases of a circle tracking task in free space in Condition 1.	37
2.3	Experimental results of the four cases of a circle tracking task in free space in Condition 1.	38
2.4	Simulation results of a circle tracking task with sinusoidal time-varying payload in Condition 2.	40
2.5	Simulation results of a circle tracking task with a suddenly added payload (23 g) in Condition 2.	41
2.6	Experimental results of a circle tracking task with a suddenly added payload (23 g) in Condition 2.	42
2.7	Simulation results of a set-point regulation task with a constant payload and 1s-disturbance in Condition 3.	43
2.8	Experimental results of a set-point regulation task with a constant payload and 1s-disturbance in Condition 3.	44
2.9	Setup and free trajectory for implementing the proposed framework.	46
3.1	The structure of feed-forward neural network (fNN) model.	54
3.2	The procedures of the proposed framework of training fNN to learn internal disturbances and thus separate external disturbances.	57
3.3	Comparison on disturbance tracking error of various observers.	59
3.4	Compare the estimation performance of fNN, cNN, and NARX model on external disturbances.	61

3.5	The effect of dynamic model error on fNN estimation performance when tracking a simple figure-eight trajectory.	62
3.6	The effect of dynamic model error on fNN estimation performance when tracking a complex exciting trajectory.	62
3.7	Simulation result of an application scenario on an obstacle of the VF.	64
4.1	Control block diagram of an impedance controller with iterative learning scheme for gravity compensation.	74
4.2	Simulation results of scaling down the iterative update time (T_{it}) while scaling down the iterative learning rate (η) accordingly.	81
4.3	Simulation results in a setpoint regulation task with a pure impedance controller under an ideal dynamic model.	83
4.4	Simulation results in a setpoint regulation task under different iterative learning rates for learning gravity compensation.	84
4.5	Simulation results in a trajectory tracking task with a pure impedance controller under an ideal dynamic model.	86
4.6	Simulation results in a trajectory tracking task under different iterative learning rates for learning gravity compensation.	87
4.7	Simulation results of comparing the gravity learning behavior in a trajectory tracking task and in a setpoint regulation task under different iterative learning rates.	88
4.8	Experimental results of a trajectory tracking task in different scenarios.	91
4.9	Experimental results of disturbance estimation by NDOB and Git in trajectory tracking tasks.	92
4.10	Experimental results in a setpoint regulation task involving p HRI.	92
4.11	Robot-assisted arthroscopy with a FAST simulator for holding with the arthroscope.	93
5.1	Illustration of the proxy in different states.	101
5.2	Simulation results of implementing the VF algorithm in four different scenarios.	109
5.3	Setup and point cloud patterns for Experiment 1 and Experiment 2.	110
5.4	Experiment 1 of implementing VF on a 2D square point cloud.	111
5.5	Preplanned 2D image of a patient with osteoarthritis diagnosed and elbow arthroscopic debridement surgery required.	112
5.6	Experiment 2 of implementing VF on point cloud of a hand-drawn curve.	113

6.1	A schematic diagram and a prototype of the two-arm robot-assisted arthroscopic surgical system, and visualization in Unity.	129
6.2	Control block diagram for the prototype of the two-arm robot-assisted arthroscopic surgical system.	130
6.3	Connectors and surgical instruments.	132
6.4	Left-arm robot experiment setup.	137
6.5	Experiment result of L-Exp.1 in <i>pHRI</i> scenario.	140
6.6	Experiment result of L-Exp.2 in <i>pHRI</i> scenario.	140
6.7	Experiment result of L-Exp.3 in <i>pHRI</i> scenario.	141
6.8	Experiment result of L-Exp.4 in <i>pHRI</i> scenario.	141
6.9	Right-arm robot experiment setup and VF-bone registration.	142
6.10	Probing test result of R-Exp.1 in <i>pHRI</i> scenario.	143
6.11	Experiment results of R-Exp.1 and R-Exp.2.	145
A.1	Schematic setup of an EMG-based hybrid impedance-force control system for human-robot collaboration on Ultrasound imaging task. . . .	170
A.2	Block diagram for the proposed EMG-based hybrid impedance-force control system.	172
A.3	Schematic illustration for the two scenarios in Experiment 1 and typical sample results.	176
A.4	Schematic illustration and results for the lifting task in Experiment 2.	178
A.5	Setup for the application in Experiment 3 with implementing the EMG-based hybrid impedance-force controller.	179
A.6	Sample data for a typical trial in Experiment 3.	179
A.7	Comparison between Experiment 3 and the two scenarios (rectangular and bowl-shaped) in Experiment 1.	181
B.1	Schematic setup and real scene for the proposed dual-mode <i>pHRI-teleHRI</i> control system.	186
B.2	Block diagram of the proposed dual-mode <i>pHRI-teleHRI</i> control system with a custom hybrid admittance-force controller.	189
B.3	Two scenarios of soft tissue surface (horizontal and inclined slope).	193
B.4	Sample data for US scanning on horizontal tissue surface with different feedback are provided in two modes in Experiment 1.	197
B.5	Sample data for US scanning on horizontal tissue surface with different feedback are provided for two modes in Experiment 2a.	199
B.6	Sample data for US scanning on inclined slope tissue surface with different feedback are provided for two modes in Experiment 2b.	201

B.7	Sample data for dual-mode switching test in Experiment 3.	203
B.8	Bar chart for normal contact force and variance.	204
C.1	Schematic of the 3DOF Phantom Premium 1.5A robot and frame at- tachment to each joint.	209

Abbreviations

***p*HRI** physical human-robot interaction.

cNN cascade-forward neural network.

DKF disturbance Kalman filter.

DOF degree-of-freedom.

EE end-effector.

EMG electromyography.

ESO extended state observer.

fNN feed-forward neural network.

Git gravity iterative learning.

GMO generalized momentum observer.

HIP haptic interface point.

MIS minimally invasive surgery.

NARX nonlinear autoregressive network with exogenous inputs.

NDOB nonlinear disturbance observer.

NN neural network.

PID proportional-integral-derivative.

US ultrasound.

VF virtual fixture.

Chapter 1

Introduction

In this chapter, literature reviews are first conducted on three topics respectively, *i.e.*, robotic systems in orthopedics, disturbance estimation approaches, and virtual fixture (VF) techniques. Then, the motivation and objectives are described. Lastly, the contributions and structure of this thesis are presented.

1.1 Literature Review

1.1.1 Robotic systems (RS) in orthopedics¹

In 1954, Devol invented the first digitally operated and programmable robot (later known as the Unimate) in the world, which is viewed as the foundation of the modern robotics industry [31]. Together with Engelberger, they founded the first robotics company in the world named Unimation. The company developed the first Unimate robot based on Devol's patent and sold it to General Motors in 1960 to be used to lift and stack hot pieces of metal [60]. Since then, robots have continually been improved, and have spread their applications to the surgical field. In 1985, the first robotic surgical system, Puma 560, was used for neurosurgical biopsies guided by computed tomography (CT) images [25, 60]. In the early 1990s, Minerva was introduced as the next-generation neurosurgical robot [82]. In 1988, ROBODOC (Integrated Surgical

¹A version of this chapter has been published as: Teng Li, Armin Badre, Farshid Alambeigi, and Mahdi Tavakoli, "Robotic Systems and Navigation Techniques in Orthopedics: A Historical Review", Applied Sciences, Section: Robotics and Automation, Special Issue: Surgical Robotics Design and Clinical Applications, 2023, 13(17):9768. [\[MDPI\]](#)

Systems, DE, USA) was introduced in orthopedics [82]. In the same year, PROBOT performed a clinical trial at Imperial College London with the earliest robotic procedure in urology. In 1993, a robotic arm called AESOP (Automated Endoscopic System for Optimal Positioning) (Computer Motion, Inc., Santa Barbara, CA, USA) was developed to assist in holding and positioning a laparoscopic camera. In 1998, both the ZEUS robotic surgical system (Computer Motion, Inc., Santa Barbara, CA, USA) and the da Vinci surgical system (Intuitive Surgical, Sunnyvale, CA, USA) were introduced into the market for use in teleoperated surgery [82], with the latter receiving the Food and Drug Administration (FDA) approval in 2000 for use in general laparoscopic surgery, which is considered legendary in the field of surgical robotics. Thereafter, medical and surgical robotics started to boom in various fields.

Minimally invasive surgery (MIS) allows the surgeon to conduct surgical procedures through much smaller incisions than traditional open surgery, thus having a faster recovery rate and shorter rehabilitation time as well as less pain for the patient [143]. Robot-assisted MIS involves a robot to improve the quality and precision of surgical procedures. Since the da Vinci surgical system was approved by the Food and Drug Administration (FDA) in 2000, the development and deployment of various robot-assisted MIS systems have been largely expedited [44, 30].

With the rapid advancement of robotic techniques in recent decades, robotic systems have been widely used in various medical fields, such as neurological, laparoscopy, radiosurgery, prosthetics, rehabilitation, orthopedics, ophthalmology, and more [12, 143, 50, 2]: for example, the da Vinci surgical system (Intuitive Surgical, Sunnyvale, CA, USA) and RAVEN II (University of Washington, and University of California, Santa Cruz, CA, USA) for use in teleoperated laparoscopic surgery [105, 12, 56], the CyberKnife System (Accuray Inc., Sunnyvale, CA, USA) for use in radiosurgery [66], and the JHU Steady-Hand Robot for use in retinal microsurgery [139, 90, 3]. These robotic systems and techniques are transforming the conventional ways to perform surgical procedures in a large variety of fields.

Through decades of technique evolution and clinical evaluations in orthopedic surgeries, plenty of studies have proven that robotic systems and navigation techniques can be beneficial in improving and enhancing surgical outcomes, such as increasing the accuracy and precision of bone cutting and component alignment, reducing operative time, and enhancing patients' satisfaction [102, 115, 100].

Numerous review papers on reviewing robots or navigation systems in orthopedics can be found in the literature. Most of them are focused on a meta-analysis or reviewing clinical outcomes and user studies [60, 100, 127, 134, 36], a specific field like MIS [143], or a specific feature like haptic feedback [35]. Instead of those, we conducted a historical review on the robotic systems and navigation techniques that exist and have ever existed in the field of orthopedics, especially on those systems still commercially available at present [74]. The primary focus of [74] is on the historical evolution of the systems as well as the engineering features and techniques from the perspective of engineering. Correspondingly, it covered two main categories, *i.e.*, robotic systems (RS), and computer-aided navigation systems (CANS). The RS is further divided into autonomous RS, hands-on RS, and teleoperated RS, while the CANS is broken down into three key technical elements, including 3D modeling, registration, and navigation.

It is worth noting that in orthopedics, computer-assisted orthopedic surgery (CAOS) divides the surgical systems into three categories, *i.e.*, autonomous (also known as active), semi-autonomous (also known as semi-active), and passive [11, 136, 102]. In our review paper [74], the autonomous systems are equivalent to the autonomous RS, and the semi-autonomous systems are equivalent to the hands-on RS, while the passive systems indicate the computer-aided navigation systems (CANS). A hierarchical flowchart of these categories and their components is illustrated in Figure 1.1.

Common orthopedic surgeries involving RS and/or CANS may include arthroplasty, arthroscopy, and surgical interventions related to tissues in joints. Note that in orthopedics, joint replacement is equivalent to arthroplasty, and similarly, total hip

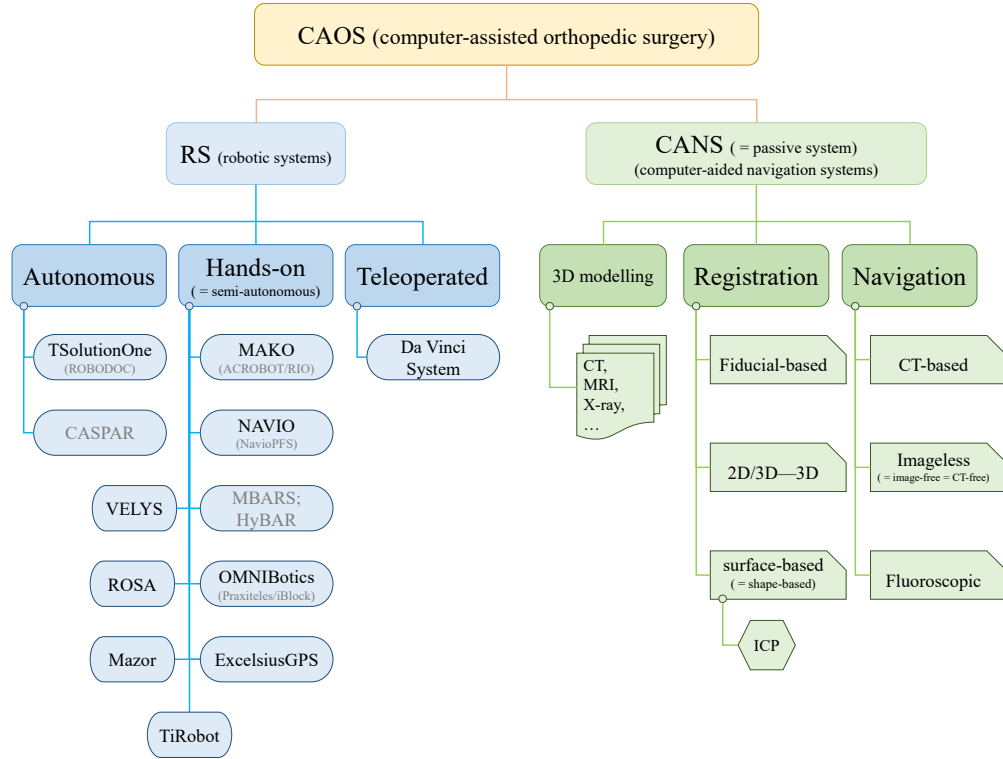


Figure 1.1: A hierarchical flowchart for the robotic systems and navigation techniques in orthopedics. Note: system names in gray color means either not in use anymore or upgraded with new names; the equal symbol “=” means “equivalent to”.

replacement (THR) is equivalent to total hip arthroplasty (THA), and total knee replacement (TKR) is equivalent to total knee arthroplasty (TKA). A general flowchart of surgical procedures for orthopedic surgeries is illustrated in Figure 1.2. Both RS and CANS play an important role in the procedures.

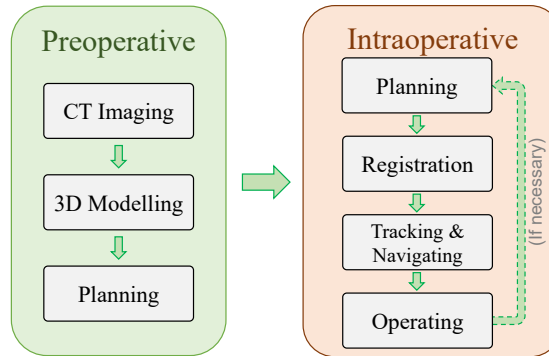


Figure 1.2: A general flowchart for orthopedic surgeries.

Robotic systems (RS) are divided into three subcategories, *i.e.*, autonomous, hands-

on, and teleoperated [35], as illustrated in Figure 1.1. The autonomous RS indicates that the robot can conduct the surgery completely on its own, while the surgeon can only interrupt it by using an emergency stop [70]. The hands-on RS is equivalent to semi-autonomous or semi-active robots in the literature of orthopedic surgeries. It indicates that the surgeon and robot cooperatively move the surgical instrument installed on the robot end effector (EE), which means it requires physical human-robot interaction (*p*HRI) [35]. The teleoperated RS indicates a standard leader-follower teleoperation system in which two robots are required. The leader robot is physically operated by the surgeon, while a follower robot on the remote site (*e.g.*, on the patient side) is controlled by the leader robot via the internet or Ethernet. Table 1.1 presents a summary list of robotic systems (RS) in orthopedics. For more details on each system, please refer to [74].

The RS systems in the timeline are illustrated in Figure 1.3. Currently, the time-honored robotic systems, including TSolution One (formerly ROBODOC; THINK Surgical, Fremont, CA, USA), MAKO (formerly ACROBOT/RIO; Stryker Corporation, Kalamazoo, MI, USA), NAVIO (Smith & Nephew plc, London, UK), and Mazor (Mazor Robotics, Caesarea, Israel), are still available and continue to improve and thrive in the market. Compared to their relatively large robot body, on the other hand, MBAR represents an emerging trend of mini bone-attached robots with potentially more efficiency and lower cost. ROSA and VELYS are newly developed products in the market. The da Vinci surgical system represents a more versatile robot that has a large potential to be used in a large variety of surgeries for the teleoperated systems used in MIS. In general, by reviewing the robotic systems developed for orthopedics in the past decades, we can see that their functions are mainly focused on bone cutting, positioning, and alignment; additionally, precision has been increasing, whereas the variability has been decreasing [60].

For autonomous RS, there is still caution about using it in orthopedic surgery due to the ethical issues and safety concerns surrounding autonomous operation [70].

Table 1.1: Robotic systems (RSs) in orthopedics.

RS	System	Usage	Features	Pros/Cons ($\oplus/\ominus$)
Autonomous	TSolutionOne (ROBODOC)	TKA; THA.	⊙ IBM, 1980s; ⊙ first patient in 1992; ⊙ pre-CT based; ⊙ 3D preoperative planning workstation ORTHODOC; ⊙ 5-axis robot; ⊙ bone milling; bone preparing.	⊖ surgeon cannot intervene but stop the robot; ⊖ recovery process is complex if robot stopped during surgery.
	CASPAR	TKA; THA.	⊙ pre-CT based; ⊙ bone milling & preparation; ⊙ position guiding.	⊖ problematic in many studies; ⊖ thus not in use already.
Hands-on (=semi-autonomous)	ACROBOT	TKA; UKA.	⊙ pre-CT based preop. plan; ⊙ 6DOF robot; ⊙ haptic feedback; ⊙ bone cutting; ⊙ active-constrained control.	⊕ system actively prevents the drill from deviating the predefined cutting zone.
	RIO	TKA; THA; UKA.	⊙ inherited from ACROBOT; ⊙ pre-CT based; ⊙ haptic & audio feedback; ⊙ bone cutting; ⊙ force-controlled tip.	⊕ system actively prevents the tool from deviating the predefined cutting zone; ⊕ short learning curve.
	MAKO	TKA; THA; UKA.	⊙ inherited from RIO; ⊙ pre-CT based; ⊙ bone cutting; ⊙ emergency auto shutdown; ⊙ a computer navigation system.	⊕ emergency auto shutdown; ⊕ haptic & audio feedback; ⊕ VF (virtual fixture).
	NAVIO (NavioPFS)	TKA; UKA; uKA.	⊙ image-free system; ⊙ tool tip extend/retract; ⊙ IR optical-based navigation system; ⊙ safety strategy of modulating burr retraction and speed.	⊕ imageless, thus no pre-CT needed. ⊖ no haptic feedback.
	BRIGIT	TKA.	⊙ teleoperated mode available; ⊙ compliant control strategy.	⊕ collaborative or teleoperated mode.
	MBARS; HyBAR	TKA.	⊙ small & bone-mounted robot; ⊙ can be autonomous.	⊕ more efficient; ⊕ cost-effective.
	iBlock (Praxiteles)	TKA.	⊙ imageless system; ⊙ bone morphing technique; ⊙ computer station OmniBiotics; ⊙ intraop. anatomic data; ⊙ intraop. planning.	⊕ can see the planned bone cut before execute. ⊖ no haptic feedback; ⊖ limited kinematic assessment after implantation.
	OMNIBotics	TKA.	⊙ upgraded from iBlock; ⊙ image-free system; ⊙ bone morphing technique; ⊙ small & bone-mounted robot; ⊙ robotic tensioning tool (active spacer).	⊕ active spacer can help to improve surgical outcomes.
	ROSA	TKA; THA; Spine.	⊙ either imageless or image-based; ⊙ bone resection; bone positioning; ⊙ soft tissue assessment.	⊕ intraop. landmarks data.
	VELYS	TKA.	⊙ imageless; ⊙ patient-specific TKA technique.	⊕ intraop. data on both anatomy and soft tissue; ⊕ allow intraop. planning.
Teleop.	Mazor	Spine.	⊙ SpineAssist is the 1st robot for spine; ⊙ four evolved system versions.	⊕ accept preop. or intraop. CT for planning; ⊕ real-time tool tracking.
	ExcelsiusGPS	Spine.	⊙ surveillance marker; ⊙ shock-absorbing.	⊕ accept preop. or intraop. CT, or radiographs for planning; ⊕ real-time image guidance.
	TiRobot	Spine.	⊙ real-time 3D navigation.	⊕ cross-referencing registration.
	da Vinci	MIS.	⊙ applied to various MIS surgeries.	⊕ versatile for various MIS surgeries. ⊖ suitable for manipulating soft tissue but not rigid bones.

Note: **UKA** (unicompartmental knee arthroplasty); **uKA** (unicondylar knee arthroplasty); **TKA** (total knee arthroplasty); **THA** (total hip arthroplasty). **pre-CT**, preoperative CT image; **preop.**, preoperative; **intraop.**, intraoperative; **Teleop.**, teleoperated; **IR**, infrared; **MIS**, minimally invasive surgery; **DOF**, degree of freedom. System names in gray color mean either not in use anymore or upgraded with new names.

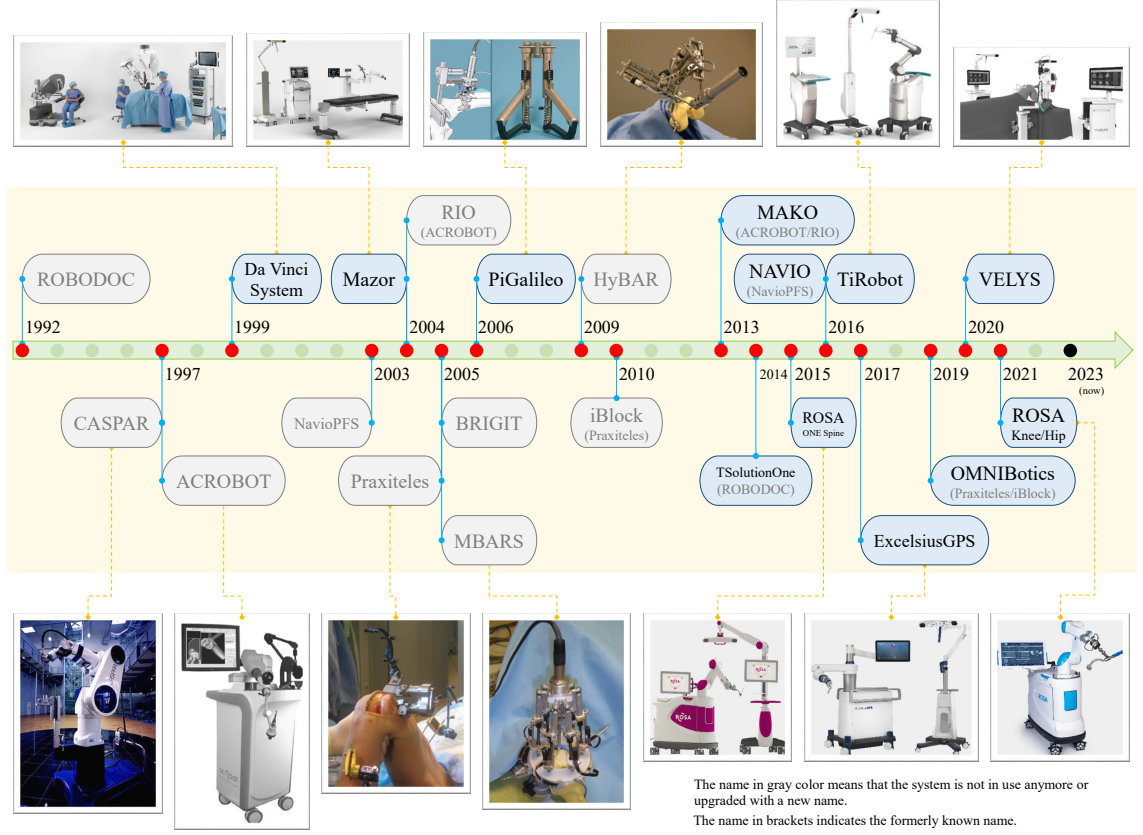


Figure 1.3: Robotic systems (RS) in the timeline.

Typical issues and concerns include potential muscle/nerve damage and technical complications. For example, when a procedure stop occurs during bone cutting, re-registration is required, while, sometimes, frequent registration failures may cause the surgery to abort [136].

Compared to autonomous RS, hands-on RS may be more acceptable by surgeons due to the feature of human-in-the-loop [136]. When operating a hands-on RS, the surgeons have full control of the robot and can stop the surgical operations (*e.g.*, bone cutting) at any time they want. This can ensure the maximal safety of the patient.

For teleoperated RS, their applications in orthopedics are limited. The main reason is that orthopedic surgeries are more related to manipulation with rigid bone cutting and implant alignments, which are not suitable for teleoperated RS. However, for those orthopedic surgeries regarding soft tissues, nerves, and vascular, the

teleoperated RS could still be applicable and useful.

Computer-aided navigation systems (CANS) can be taken as a parallel category to RS. The CANS focuses on navigation with the help of computers. It can be either integrated with an RS or independent from an RS. When integrated together, all the coordinates of the CANS (*e.g.*, CT image frame, and external camera frame) are registered into the coordinates of the RS, then for navigation. When independent from an RS, *i.e.*, no robots appear in surgery, all the coordinates are registered into the digital patient’s model/image or the camera frame, then for navigation.

In the category of computer-aided navigation systems (CANS), three basic elements are included, *i.e.*, 3D modeling, registration, and navigation. Strictly, the CANS are computer-assisted systems rather than robot-assisted systems. However, a robot-assisted system (here equivalent to RS) usually includes a CANS system implicitly or explicitly. Implicitly means that an RS itself can be viewed as a special navigation system since any point in the robot workspace can be tracked based on robot kinematics [102]. Explicitly means that a CANS can be integrated with a robotic system to enhance the system’s ability (*e.g.*, tracking and visualization). Therefore, a CANS can be used either independently or integrated with an RS, which means a CANS can be an essential part of assisting surgeons in surgery, no matter whether a robot is involved or not. This also means that the CANS has wider and more general applications than RS in orthopedics and beyond. For more details on the three basic elements of CANS as well as typical navigation systems, please refer to [74].

Both CANS and RS have been playing an increasingly critical role in modern orthopedic surgeries. According to a review study [24, 102], the ratio of patents to publications related to CANS and RS in knee arthroplasty increased from around 1:10 in 2004 to around 1:3 in 2014. The benefits brought by RS and CANS include but are not limited to augmenting the surgical procedures, fine tuning surgical plans to personalized patient profiles, and proving intraoperative data and real-time visualization to the surgeons for a more accurate and precise surgical outcome [102].

With decades of evolution, RS and CANS themselves have gone through critical improvement and upgrading. Without a doubt, RS and CANS will continue to thrive and play an indispensable role in orthopedics.

A critical unique feature of RS in orthopedics is that they must be capable of dealing with high forces and stiffness due to the rigid nature of their target object of bones, while the da Vinci surgical system is more suitable for soft-tissue-related procedures. The main advantages of RS applied in orthopedics include increased accuracy and precision of implant positioning, enhanced reproducibility, improved implant stability, and less resulting pain. On the other hand, the main disadvantages of RS include potential safety concerns, high economic costs, and potentially longer operative times.

CANS will continue to develop along two parallel paths. One is to integrate with robotic systems, another is to be used alone without involving RS. For the latter, CANS is capable of being used in more versatile surgeries, where robots are not needed or not yet available. In that case, with the help of CANS, surgeons can perform conventional surgeries with potentially better and more accurate surgical outcomes.

From Table 1.1, it can be found that most of the robotic systems are applied for hip/knee surgeries, some are for spine surgeries, and no specific robotic system is exclusively for elbow/shoulder surgeries. For example, ROSA has robot-assisted systems for the hip, knee, and spine, respectively, but only has a computer-aided system for assisting in planning and navigation for the shoulder. RS have been abundantly developed and frequently used in hip/knee surgeries but not in elbow/shoulder/foot/ankle surgeries, while the latter seems more favorable for CANS [17]. The possible reason could be that there are much more cases and higher demand for knee/hip surgery than elbow/shoulder surgery. The knee is the largest hinge joint in the body, while the hip is a large ball-and-socket joint. Both the knee and hip joints take a lot of wear, tear, and stress from daily activities (*e.g.*, walking, running, and jumping) while

supporting the body weight in horizontal and vertical directions, and they are more vulnerable to injury and osteoarthritis than the elbow/shoulder or any other joints.

Technically, all joint replacement/arthroplasty surgeries are open surgeries since a significant incision needs to be made in order to expose the bone for bone cutting or implant positioning. It is worth noting that in joint replacement/arthroplasty, minimally invasive approaches/procedures have different definitions, such as shorter incision length, nondissection of quadriceps tendon, noneversion of the patella, or nondislocation of the tibiofemoral joint [72]. Therefore, strictly speaking, all joint replacement/arthroplasty are open surgeries rather than MIS, the latter of which only requires several trocars to be made in order to insert the surgical instruments for performing the surgery. In this sense, arthroscopic surgeries belong to the category of MIS. For arthroscopic surgeries, several portals are made to insert an arthroscope and surgical tools, and the surgical tools are used to perform the surgery with the help of the arthroscope. Arthroscopic surgeries can be used for a large variety of indications. For example, elbow arthroscopy can be used for the management of stiffness related to degenerative arthritis, loose bodies, lateral epicondylitis, synovitis, osteochondritis dissecans, symptomatic plica, infection, contracture, instability, and fracture management [8].

Arthroscopic surgery (*e.g.*, arthroscopic debridement) is an active field that is being transformed by techniques of RS and CANS, although there is yet no specific robotic system specially designed for them. One possible reason is that there is high demand for the accuracy and precision of surgeries for joint replacement/arthroplasty but not for arthroscopic surgeries. Also, for those arthroscopic surgeries related to soft tissue manipulation rather than rigid bone cutting, laparoscopic-type robotic systems like the da Vinci surgical system can be employed [44, 37]. A robotic system can be helpful in arthroscopic surgeries, such as holding with the arthroscope as a robotic assistance. On the other hand, a navigation system can also bring benefits, such as tracking and visualizing the real-time location of the tool tip. Accompanying the wide

usage of arthroscopic surgeries with the benefits of MIS, RS and CANS are becoming deeply involved in arthroscopic surgeries [53].

In addition to traditional rigid robots, novel types of robots, instruments, and approaches have been developed for use in orthopedic surgeries [122]. A curved drilling approach was developed by integrating curved drilling tools with a continuum dexterous manipulator (CDM) for use in core decompression of the femoral head osteonecrosis [7]. The curved drilling technique and bendable medical screws have been examined on cadaveric specimens for minimally invasive interventions in orthopedic surgery [6]. A redundant robotic system consisting of a rigid-link robot and a CDM was proposed for the treatment of pelvic osteolysis and for the autonomous debridement of osteolytic bone lesions in confined spaces [118, 119]. A miniaturized tendon-driven articulated surgical drill was designed for bone drilling, which can be used in minimally invasive spine fusion [146]. Hand-held robotic systems have also been developed for minimally invasive orthopedic surgeries [85, 147]. Recently, a concentric-tube steerable drilling robot was developed for spinal fixation procedures and implanting flexible pedicle screws [126, 125]. Although these novel designs have not yet been applied in the clinical setting, their benefits in orthopedic surgeries can be expected in the near future.

In parallel to robotic systems, another promising robotic technique is surgical simulators for training novices, *e.g.*, virtual reality (VR) arthroscopy trainer, VirtaMed ArthroS Hip/Knee/Shoulder/Ankle (VirtaMed AG, Zurich, Switzerland), and insightArthroVR (GMV, Madrid, Spain) [145, 106]. By using surgical simulators, the surgical skills of the novice can be improved before they start to conduct surgeries on human patients. This can largely enhance the novice’s confidence and reduce the risk of surgical mistakes in patients caused by lacking practical experience and unfamiliar operations on surgical robotic systems.

As a prominent topic in recent years, artificial intelligence (AI) is becoming extraordinarily popular, especially after the breakthrough made by ChatGPT (OpenAI, San

Francisco, CA, USA), although AI itself is not a novel concept [95]. Some AI-based features have already been applied in robot-assisted surgeries, for example, the AI algorithms presented in [34, 69, 95]. Benefiting from the huge amount of patient data available in literature and hospitals, a series of reliable AI-based techniques can be expected, such as AI-based diagnosis, AI-based pre- and intra-operative planning, AI-based intraoperative navigation, AI-based decision making, and AI-based control of robotic systems [95]. By appropriately incorporating these AI features, the capability of the robot and navigation systems can be further improved and enhanced.

In summary, it is an unstoppable trend for the RS and CANS to be introduced into a greater variety of surgical scenarios besides hip/knee surgeries, and it is rapidly happening. Robotics and navigation techniques have been playing increasingly important roles in elbow/shoulder, foot/ankle, spine surgeries, arthroscopic surgeries, and far beyond [64, 17, 88, 53, 36, 135]. With the newly emerging techniques, such as AI, VR, and soft/flexible robotics, robots and navigation systems in orthopedics will become more and more intelligent, reliable, and economical.

1.1.2 Disturbance estimation approaches¹

In a robot control system, an accurate dynamic model of the robot is fundamentally important for accurate and stable control [47]. This is true for all kinds of robots, such as industrial robots, humanoid robots, medical robots, soft robots, and exoskeletons. However, accurate dynamic models only exist in theory but not in practice, since various uncertainties can be residing in the dynamic model inevitably. Examples of such uncertainties are joint friction, inaccurate center of mass location and link weight, extra payload, and robot-environment interaction [47]. Therefore, it is a fundamental topic for estimating and compensating for uncertainties in the field of robot control.

Many methods have been developed for estimating dynamic uncertainties thus

¹A version of this chapter has been published as: Teng Li, Hongjun Xing, Ehsan Hashemi, Hamid D. Taghirad, Mahdi Tavakoli, “A Brief Survey of Observers for Disturbance Estimation and Compensation”, *Robotica*, 2023, 41(12), 3818–3845. [Robotica]

eliminating their effects on robot dynamics. Disturbance observer is a main solution that can observe the dynamic uncertainties in an online manner, thus making compensation accordingly when necessary.

A variety of observers have been developed [107, 19], but not all of them can be implemented into a robotic system. Some typical observers are identified according to the following procedures. First, a bunch of observers is collected based on three recent review papers [52, 93, 21] and one research paper [57]. Then, the following two rules are applied, (a) the observer is applicable in practice in a physical robotic system; and (b) the observer is independent of the controllers, *i.e.*, not relying on a specific controller. Finally, five typical observers are identified, including generalized momentum observer (GMO) [52], joint velocity observer (JVOB) [52], extended state observer (ESO) [117], nonlinear disturbance observer (NDOB) [94], and disturbance Kalman filter (DKF) method [57, 80].

Generalized momentum observer (GMO), also known as classic first-order momentum observer, is originally proposed for actuator fault detection and isolation, aiming to avoid joint acceleration measurements and inverse of the robot inertia matrix in the control system [27, 28]. Note that in practice, measuring accelerations is usually avoided due to their high price or installation size restrictions. Furthermore, obtaining accelerations via numerical differentiation of velocity or position is not preferred since it will introduce noise into the system and thus may affect the system's stability [52]. The inverse of the robot inertia matrix can increase the computational load on the system. Besides not requiring the accelerations and inverse of inertia matrix, the GMO also has the advantage of being simple, compact, and easy to implement [154], all of which make it one of the most commonly used observers. It is usually used as a benchmark for comparison when designing new observers [45, 57, 80].

Joint velocity observer (JVOB) is a similar observer to the GMO in terms of the procedures of derivation and the final expression, but the inversion of the robot inertia matrix is needed [52]. The JVOB is derived based on the acceleration expressed by

the robot dynamics, then the integration of the acceleration is taken as an estimate of the velocity where the true velocity is assumed to be known. Therefore, the JVOB is a reduced (first-order) observer with a reduced state of dimension n , where n is the number of the generalized coordinates of the robot.

Extended state observer (ESO) was originally proposed by Han in the 1990s, and it is considered as the critical part of active disturbance rejection control (ADRC) which is developed to estimate the lumped uncertainties including both unknown dynamic uncertainties and external disturbances [54, 55]. For a specific survey on ADRC and ESO, please refer to [58]. In another historical survey on observers in 2006 [107], ESO has been viewed as an indicator of an initial shift of design methodology from modern estimators (*e.g.*, Kalman Filter) to disturbance estimators (*e.g.*, ESO). The ESO employs a simple canonical form which is considered as a practical design, thus it receives many applications in different fields, such as power converters, web tension, and bio-mechanics [107]. In the field of robot control, various variations of ESO have been developed in different application scenarios such as collision detection [109] and time-varying interaction force estimation [117].

Nonlinear disturbance observer (NDOB) is originally proposed by Chen, which is considered to overcome the shortcomings of linear disturbance observer (DOB) that is designed or analyzed by linear system techniques [18]. Although this version of NDOB was developed for constant disturbances in theory, it also revealed satisfactory performance on estimating time-varying disturbances like friction. However, it is merely used for planar robots with revolute joints. To solve this problem, Mohammadi *et al.* [94] proposed a general framework for NDOB by unifying linear and nonlinear disturbance observers which released the restrictions on the number of degree-of-freedom (DOF), the types of joints (revolute or prismatic), or the robot configuration.

Kalman filter (KF) is an early approach to be used for disturbance estimation, which is also one of the first estimators that involve the formulation of disturbances [107]. Based on the original KF, disturbance Kalman filter (DKF) is developed to

estimate the dynamic uncertainties in a robot control system [57, 80]. The DKF reveals optimal disturbance tracking performance, but the implementation complexity could be a limitation for its wide use [107].

All the five types of observers introduced above estimate a lumped uncertainty term. While the lumped uncertainties include various components (*e.g.*, model error, joint friction, external payload, human-exerted force, and beyond), it is not possible to separate a specific component out of the lumped uncertainties term. Especially in human-robot interaction scenarios, an operating observer will take the human-exerted force as a part of uncertainties and thus reject it [75].

In our review paper on disturbance observers [78], we reviewed these five typical types of disturbance observers, *i.e.*, GMO, JVOB, ESO, NDOB, DKF. For the ESO, three of its variants are presented, including the original ESO (ESO.orig), a modified ESO (ESO.modi), and an improved ESO (ESO.impr). Also, (1) we presented the basic expressions of the five typical observers with which the observers can be quickly implemented into a robotic system, and (2) we presented the behaviors of different observers by presenting them in the same specific simulated scenarios. Experiments are also conducted to demonstrate the effectiveness of the observers in a real application scenario. For more details on these observers, please refer to [78].

In Table 1.2, some main features of each observer are summarized including whether the observer works for constant/sinusoidal disturbance, a subjective score on the complexity of implementation, and whether the acceleration and the inverse of inertia matrix are needed.

In addition to the disturbance observers, many learning techniques have been also applied for disturbance estimation and compensation, such as using feedforward neural network (NN) [98, 144, 80], and nonlinear autoregressive network with exogenous inputs (NARX) [124].

The interaction force between human-robot or robot-environment is a specific type of disturbances, and estimating the interaction force is a critical problem in the field

Table 1.2: A summary of the main features of observers.

	Observer	Constant	Sinusoidal	Complexity	Acceleration ($\ddot{q}$)	Inertia	Reference
1	NDOB	✓	✓	**	×	M^{-1}	[94]
2	GMO	✓	✓	*	×	M	[32]
3	JVOB	✓	✓	*	×	M^{-1}	[52]
4	DKF	✓	depend.1	*****	×	M	[57, 80]
5.1	ESO.orig	✓	NA	****	×	M^{-1}	[55, 109]
5.2	ESO.modi	✓	depend.2	***	×	M	[109]
5.3	ESO.impr	✓	✓	***	×	M^{-1}	[117]

Note: **Constant**, means constant disturbances; **Sinusoidal**, means sinusoidal disturbances; **Complexity**, means the subjective complexity of implementing the observers, and the more asterisk (*) means the more complexity of the implementation. **Inertia**, means the form of the inertia matrix required.

symbol (✓), means “yes, applicable”; NA, not applicable; symbol (×), means “no need”;

depend.1, means that it depends on what dynamics model of the disturbances is used. If a constant disturbance model is used, then it is only valid for constant disturbance, *e.g.*, a constant payload at the end-effector. If a polynomial disturbance model is used, then it is valid for both constant and time-varying disturbance.

depend.2, means that it depends on parameter tuning. The same set of parameters cannot be valid for both constant and time-varying disturbances at the same time. The parameters need to be tuned separately in different disturbance scenarios in order for optimal effect in each.

of human-robot interaction and the field of collision detection [52]. Some methods have been developed to estimate the robot-environment interaction force by involving learning techniques. Hu and Xiong [57] developed a method to estimate external contact force using a semiparametric model and DKF. In their method, the semiparametric dynamic model containing a multilayer perceptron (MLP) neural network was used to provide a more accurate dynamic model, while a DKF was used to estimate the contact force between the robot and the environment. Similarly, an integrated framework of neural network (NN) and DKF was developed to estimate external contact force [80]. In the framework, an NN model was used to learn the joint friction, while a DKF observer was used to estimate the contact force. In another work, NN was used to approximate the global friction, while a momentum observer was used to estimate the external force. Additionally, a Kalman filter was employed to filter the measurement noise for a more accurate force estimation.

Sharifi *et al.* [124] employed a nonlinear autoregressive network with exogenous inputs (NARX) to learn and estimate the robot dynamics plus the passive dynamics of a user who wore an exoskeleton. Then the human-exoskeleton interaction force was

separated out by subtracting the NARX-learned dynamics from the motor torques in the scenario of active user dynamics. Note that the movement trajectories of the exoskeleton is nearly periodic, which helped with training the NARX model to generate its model predictions.

Learning control has been developed to track repetitive trajectories for both rigid and non-rigid robots. De Luca and Ulivi presented a simple and efficient iterative learning algorithm for robots with joint elasticity [29]. In their work, a learning term was used to learn the necessary modification to the desired trajectory position. They demonstrated the algorithm’s usefulness by good motion performance of simulations on a two-link planar robot. Based on a similar design methodology, an iterative learning scheme for gravity compensation in setpoint regulation problems was initially proposed by De Luca and Panzieri [26, 84]. The learning scheme completes the required gravity compensation at the final steady state in setpoint regulation tasks. It can iteratively learn the constant gravity without the need of introducing an integral error term or using high-gain feedback.

Based on the same contraction mapping theorem, Basovich *et al.* developed an iterative output feedback controller for a 6-degree-of-freedom (DOF) precision positioning system when only position measurement is available [10]. Their proposed controller can learn and compensate for the payload uncertainties with bounded error in setpoint control tasks. Ji *et al.* used the iterative learning method to auto-calibrate gravity compensation when the robot has no contact with the environment thus making the robot EE weightless [61].

An adaptive controller [130, 59] can also deal with dynamic uncertainties including gravity. However, it is a controller rather than an independent approach for disturbance estimation, and it cannot provide compliant robot behavior for a safe human-robot interaction like an impedance controller can do.

Impedance control is widely used for human-robot collaboration due to its intrinsic property of compliance [133]. By an impedance controller, a robot can be

controlled to be soft (compliant) or rigid (non-compliant) as necessary, which can ensure a safe human-robot interaction [129]. Furthermore, the measurement on the interaction force is not necessary for impedance control. Therefore, the sensor-free and compliance properties make it popular for surgical robots. One main issue for implementing an impedance controller is that it requires full knowledge of the robot dynamics whereas the identified dynamic parameters are usually inaccurate in practice. On the other hand, admittance control, which can be viewed as a counterpart of impedance control, does require an external sensor to measure the interaction force though it does not require full knowledge of the robot dynamics [142].

1.1.3 Virtual fixture techniques

Virtual fixture (VF), also known as active constraint and first proposed in [110], is usually categorized into two types according to its purpose, that is guidance virtual fixture (GVF) and forbidden-region virtual fixture (FRVF) [1, 14]. Intuitively, the GVF serves like a ruler to assist in drawing a straight line, while the FRVF serves like an armor to prevent tool tip from entering a protected area. Both types play a vital role during various surgical procedures in robot-assisted surgery, such as suturing [86], knot tying [150], dissection [121], either assisting to move the surgical tool along a prescribed trajectory or preventing it from entering a specific area for protecting the objects (*e.g.*, beating heart or nerve) inside [112, 92].

VF is usually generated based on geometric elements, such as lines, planes, surfaces, and volumetric primitives [14]. Vector field approach is the most common one to be used for VF generating, which works for any shape that can be expressed as mathematical equations [155]. The advantage of the vector field approach is that it is simple, straightforward, and stable, while the disadvantage is that it requires an explicit/implicit mathematical representation for the object to be modelled as VF. For the objects with regular shapes like cube and sphere, their mathematical representations can be easily established then the VF can be constructed relatively

easily. However, for objects with irregular shapes like a humerus bone or skull, they may not be able to expressed by equations, then the vector field approach may not work for these objects any more.

The vector field approach cannot correctly handle situation of in contact with multiple objects simultaneously and situation of thin objects. To solve this problem, Zilles and Salisbury developed a constraint-based god-object algorithm [155]. In their work, the god-object is a proxy of the haptic interface point (HIP) which is attached onto the HIP when the HIP is in free motion. Once the HIP encounters VF, a virtual wall for example, the proxy will always remain on the top of the virtual wall and never penetrate into it. This is ensured in their algorithm by applying Lagrange multiplier technique on a set of active constraints to find the position of the proxy in each servo loop. Meanwhile, the virtual spring and/or damper linkage between the HIP and the proxy will render a haptic force that tries to pull the HIP back out of the virtual wall.

Kapoor *et al.* developed a constrained optimization method for generating VF [63], in which a suitable objective function is required in order to do the optimization. In the method, five basic geometric constraints are established as VF task primitives which can be used for assembling customized VF. With similar technique, Marinho *et al.* employed a method of vector-field inequalities to generate VF for collision avoidance [87] and guidance in a looping task during suturing [86]. Xia *et al.* developed a constrained optimization framework of VF generating for multi-robot collaborative teleoperation tasks, *e.g.*, knot positioning [150].

There are also some other methods for VF generating for different purposes, such as potential field method for collision avoidance or guidance [22, 120] and nonenergy storing method for a more stable robot behavior [65]. Readers is directed to [14] for a comprehensive review on VF.

In the field of robot-assisted surgery, VF has been widely used due to plenty of advantages, such as reduce the surgeon’s cognitive load [150], improve surgeon’s surgical performance [48], make the surgical outcome more accurate and safe. Park *et al.* [99]

conducted a preliminary test before applying VF in a coronary artery bypass surgery. In the test, VF is generated for a blunt dissection task at a position determined from a preoperative CT scan image, and the VF is a regular plane thus the VF generating is in a simple case in their work.

Ryden *et al.* [112] developed a method to generate VF directly from point cloud to protect the beating heart during surgery. They improved their method in [111, 96], which established a solid foundation for point-based VF algorithm.

1.2 Motivation and Objectives

Arthroscopic surgery is a typical type of minimally invasive surgery (MIS) and a very common surgery in orthopedics. For instance, elbow arthroscopy allows the management of elbow stiffness, arthritis, and fractures in a minimally invasive fashion [13]. During traditional arthroscopic surgery, the surgeon needs to hold an arthroscope with one hand while performing the surgical operations with the other hand, which can restrict the dexterity of the surgical performance and increase the cognitive load. According to the literature review on robotic systems in the previous section, most of the existing robot-assisted surgical systems in orthopedics take more focus on open surgery than MIS. There is yet no specific robotic system specially designed for arthroscopic surgeries. This arouses the necessity to develop a robot-assisted system where the robot can assist the surgeon in performing the arthroscopic surgery.

To build a robot-assisted arthroscopic holder where the robot can hold the arthroscope for the surgeon, some requirements need to be satisfied. First, the robot can hold the arthroscope still at a specified pose while rejecting all possible disturbances (*e.g.*, external disturbances delivered to the arthroscope via contact with the patient’s body during surgery). Second, when the surgeon needs to move the arthroscope to a new pose (*e.g.*, for adjusting the scope view perspective), the robot should allow the surgeon to move it around freely (*i.e.*, human-robot interaction). Then, when a new pose is determined by the surgeon, the robot should keep the arthroscope still again

while rejecting any disturbances. The main problem in building such a robot-assisted system is the dynamic model uncertainties and external disturbances, which could largely affect the robot’s task performance accuracy and even stability if they are not appropriately compensated.

On the other hand, the surgeon heavily relies on the arthroscope view when conducting the surgery whereas the arthroscope view is a largely localized view and lacks depth information. Also, in orthopedics, a series of preoperative images of a patient are usually first acquired before the surgery. Some surgical preplans could be made on the images, for example, drawing some curves to mark out areas to remove osteophytes, or to protect nerves inside. During the surgery, these hand-drawn curves can be used to generate VF which can assist in removing osteophytes or protecting the nerves by providing haptic feedback. The main challenge here is that the hand-drawn curves are usually in irregular shapes which may not be able to be presented mathematically by equations that are often required by most of existing VF-generating algorithms. This motivates us to develop a robot-assisted system that can provide additional haptic feedback and 3D visualization to the surgeons.

The ultimate objective of this research project is to develop a two-arm robot-assisted arthroscopic surgical system for assisting surgeons during arthroscopic surgery. More specifically, the left-arm robot will be a robot-assisted arthroscope holder, while the right-arm robot will be a robot-assisted surgical tool. The left-arm robot should be able to keep the arthroscope still while rejecting all disturbances, and also allow the operator to move it freely if needed. The right-arm robot should be able to provide the surgeon with haptic feedback assistance from VF where the VF can be generated for objects in any shape. Also, 3D visualization of the tool position and VF should be provided to the surgeon as additional visual feedback.

1.3 Contributions

The main contributions of this thesis are summarized as follows,

- (1) An integrated framework consisting of impedance control and NDOB is constructed for elbow arthroscopic surgery. The proposed framework is able to accurately estimate and compensate for dynamic uncertainties, while guaranteeing compliant robot behavior via the impedance controller.
- (2) A gravity iterative learning (Git) scheme for gravity compensation in Cartesian space is developed, and the converging properties are theoretically analyzed. Furthermore, a steady-state scaling strategy is proposed, which enables the iterative learning update law to be executed in each servo loop, thus extending the validity of the learning scheme to general trajectory-tracking scenarios in addition to setpoint regulation scenarios.
- (3) A point-based VF algorithm is presented which allows to generate VF directly from point cloud data no matter what shape it is in. The presented VF algorithm can be applied in various scenarios in image-guided surgery, as long as a set of point cloud of the target object can be obtained.
- (4) A two-arm robot-assisted arthroscopic surgical system is developed. The left-arm robot is able to keep the arthroscope still while rejecting all disturbances, and also allows the operator to move it freely if needed. The right-arm robot is able to provide haptic assistance from point-based VF.

1.4 Structure of This Thesis

The structure of this thesis is illustrated in Figure 1.4. In the current **Chapter 1**, literature reviews were conducted on the evolvement of robotic systems in orthopedics, disturbance estimation approaches, and virtual fixture techniques, respectively. Then the development of a two-arm robot-assisted arthroscopic surgical system was motivated, and the challenges of building such a robotic system were identified. In **Chapter 2**, an integrated framework of combining impedance control and nonlinear

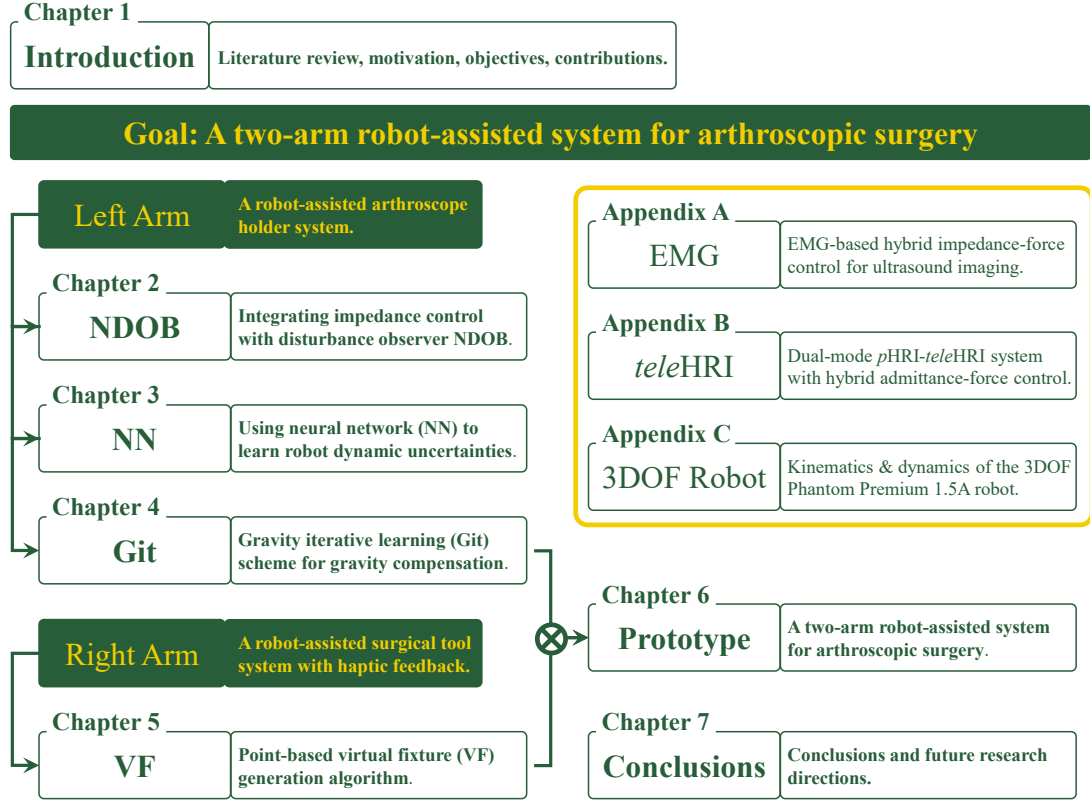


Figure 1.4: Structure of this thesis.

disturbance observer (NDOB) was proposed for arthroscopic surgery. In **Chapter 3**, in aiming to separate some specific components in the uncertainties, a framework for using neural network (NN) to learn dynamic uncertainties is proposed. In **Chapter 4**, an iterative learning scheme with a steady-state scaling strategy for gravity compensation, *i.e.*, Git (gravity iterative learning) scheme, is developed. In **Chapter 5**, an algorithm of point-based virtual fixture (VF) generating method was presented. In **Chapter 6**, a prototype of a two-arm robot-assisted arthroscopic surgical system is presented and evaluated by assembling the work in **Chapter 4** and in **Chapter 5**. In **Chapter 7**, conclusions of this thesis and future research directions are presented.

In **Appendix A**, an EMG-based hybrid impedance-force control system for human-robot collaboration on ultrasound imaging is introduced. This work shows another possibility of how we use impedance control in another medical *pHRI* scenario of ultrasound imaging. In **Appendix B**, a dual-mode *pHRI-teleHRI* control system

with a hybrid admittance-force controller for ultrasound imaging is presented. This work shows an example application of admittance control, another compliant control scheme in parallel to impedance control, in a *p*HRI scenario of ultrasound imaging. In **Appendix C**, the kinematics and dynamics of the 3DOF PHANToM Premium 1.5A robot are presented since this robot was employed frequently in this thesis.

Chapter 2

Integrating Impedance Control with Nonlinear Disturbance Observer¹

Compliance and stability are essential features that a surgical robot must have for safe physical human-robot interaction (*pHRI*). Surgical tools attached at the robot end-effector and human-robot interaction will affect the robot dynamics inevitably. This could undermine the utility and stability of the robotic system if the varying robot dynamics are not identified and updated in the robot control law. In this Chapter, an integrated framework for robot impedance control and nonlinear disturbance observer (NDOB)-based compensation of uncertain dynamics is proposed, where the former ensures compliant robot behavior and the latter compensates for dynamic uncertainties when necessary. The combination of impedance controller and NDOB is analyzed theoretically in three scenarios. A complete simulation and experimental studies involving three common conditions are then conducted to evaluate the theoretical analyses. A preliminary *pHRI* application on arthroscopic surgery is designed to implement the proposed framework on a robotic surgeon-assist system and evaluate its effectiveness experimentally. By integrating impedance controller with NDOB,

¹A version of this chapter has been published as: Teng Li, Armin Badre, Hamid D. Taghirad, Mahdi Tavakoli, “Integrating Impedance Control and Nonlinear Disturbance Observer for Robot-Assisted Arthroscope Control in Elbow Arthroscopic Surgery”, 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022), October 23-27, Kyoto, Japan, 2022. pp. 11172-11179. [\[IEEE Xplore\]](#)

the proposed framework allows an accurate impedance control when dynamic model inaccuracy and external disturbance exist.

2.1 Background

Elbow arthroscopy is a novel and complex procedure that allows management of elbow stiffness, arthritis and fractures in a minimally fashion [13]. Minimally invasive surgery (MIS) has been gaining popularity due to its potential benefits of faster recovery time and decreased pain [43]. Recent advancements in surgical robotics are transforming the traditional orthopaedic surgeries and helping the surgeons generate more successful and precise surgical outcomes [140, 60, 89]. Furthermore, MIS has been adopted due to the widely used da Vinci robot system in more and more types of surgical operations [44, 30].

In robot-assisted elbow arthroscopy, a robot can help to increase surgical accuracy and precision, *e.g.*, for bone alignment. It can also serve as an auxiliary surgical support, *e.g.*, to help surgeons reduce fatigue by holding the arthroscope [89]. In such a robot-assisted surgical support system, the robot dynamics can be dramatically affected due to the extra surgical tools attached onto the robot end-effector (EE) and the physical human-robot interaction (*pHRI*). Here both the mass of the surgical tools and the interaction force can be viewed as external disturbance. Therefore, an accurate estimation and compensation for dynamic uncertainties can be a critical step towards a stable and accurate control system.

Many methods have been proposed for estimating dynamic uncertainties and eliminating their effects on robot dynamics. A main solution is to design a linear or nonlinear disturbance observer (NDOB) which can monitor the dynamic uncertainties in real-time, and make a compensation when necessary. A comprehensive review on NDOB can be found in [93], and a historical review on versatile observers can be found in [107].

A classic first-order momentum observer and its variations of sliding mode momen-

tum observers have been designed to estimate an external disturbance/force [45]. A disadvantage of these observers is that they require an accurate dynamic model and negligible/known friction torque for an accurate estimation. This makes sense since the observer usually estimates a lumped dynamic uncertainty term which includes both the dynamic model inaccuracy and the external disturbance/force. Therefore, only when the dynamic model is accurate, the estimation from the observer is of any accuracy. Additionally, the classic approach can provide exact estimation only when the external force is constant rather than time-varying.

A model-based extended state observer (ESO) [117] is used for estimating human-robot interaction force on an impedance-based three degrees-of-freedom (DOF) rehabilitation robot when a healthy subject and a post-stroke patient operates the robot separately. The effectiveness of ESO is demonstrated by simulation. However, as the authors analyzed, the experiment result on force estimation is not accurate enough due to the inaccurate dynamic model they used.

A NDOB observer is used to estimate constant external payloads on the robot EE of a 6-DOF WallMoBot [152]. Only 1-DOF is involved in the experiments for the sake of simplicity. The effectiveness of the NDOB observer has been demonstrated in that work.

As mentioned earlier, the output from an observer is a lumped uncertainty term that incorporates both the dynamic model inaccuracy and the external disturbances. In order to have an accurate estimation on the external disturbances (*e.g.*, interaction force), the dynamic model inaccuracy (*e.g.*, friction) is better to be estimated independently. Research has focused on this issue by employing an additional observer. A neural network (NN) is utilized to reconstruct friction dynamics while a general momentum disturbance observer is used to estimate the external forces in [81]. The estimation accuracy on the external force is considerably improved since the friction is separately estimated by the NN. Similar approaches of combining a disturbance observer and a deep learning technique are also introduced in [57, 80].

Impedance control is widely used for human-robot collaboration due to its intrinsic property of compliance [133]. By an impedance controller, a robot can be controlled to be soft (compliant) or rigid (non-compliant) as necessary, which can ensure a safe human-robot interaction [129]. Furthermore, the measurement on the interaction force is not necessary for impedance control. Therefore, the sensor-free and compliance properties make it popular for surgical robots. One main issue for implementing an impedance controller is that it requires full knowledge of the robot dynamics whereas the identified dynamic parameters are usually inaccurate in practice. Note that admittance control, which can be viewed as a counterpart of impedance control, does require an external sensor to measure the interaction force though it does not require full knowledge of the robot dynamics [142].

During arthroscopic surgery, knowing the position and orientation of the arthroscope is critical for improving the surgeon’s situational awareness by allowing the surgeon to know the relationship between the current field of view and the surgical pre-plan [40, 42]. A common way to monitor the position and orientation of the arthroscope is to employ a tracking system [128]. The drawback of using a tracking system is that the marker could be easily obstructed by the surgeon’s body or other objects in the operating room. A robot-assisted surgical system is another promising solution where the coordinate system of the arthroscope-holding robot can be used for the tracking. Meanwhile, the robot can help to improve surgical accuracy and reduce the surgeon’s effort by holding the arthroscope’s weight. Developing such a robot-assisted surgical system as the physical user interface is meaningful both for a virtual surgical training simulator and for supporting the surgeon in live surgical procedures.

In our specific application of elbow arthroscopy MIS, an integrated framework consisting of impedance control and NDOB is constructed as illustrated in Figure 2.1. In the proposed framework, the impedance control is selected to guarantee compliant robot behavior without using a force/torque sensor [133], while the NDOB is used to

estimate dynamic uncertainties and compensate for them when necessary. In summary, the contributions of this chapter are: (1) A novel framework for integrating impedance control and NDOB is proposed and the effectiveness is evaluated; (2) The combined output is analyzed in three scenarios, and evaluated by simulation and experimental studies.

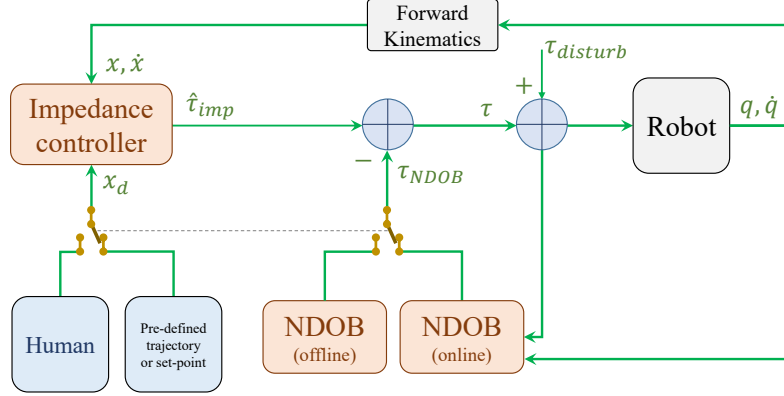


Figure 2.1: Block diagram of the proposed control scheme with integrating impedance controller and nonlinear disturbance observer (NDOB). Note that the dashed line means a linked switch. $\tau_{disturb}$ is the dynamic model inaccuracy and the external disturbance.

2.2 Methods

2.2.1 Robot dynamics and impedance control

A general dynamic model for an n -degree-of-freedom (DOF) rigid robot [41] can be given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J}^T \mathbf{F}_{\text{ext}} \quad (2.1)$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ denotes the inertia matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes a matrix related to the Coriolis and centrifugal forces, $\mathbf{g} \in \mathbb{R}^n$ represents a gravity-related vector, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the commanded joint torque vector, $\mathbf{F}_{\text{ext}} \in \mathbb{R}^6$ is the external force in Cartesian space, and $\mathbf{J} \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix. Note that friction is not included in the dynamic model (2.1) in this chapter.

A full impedance model [129, 133] for robot-environment contact can be given by

$$\mathbf{F}_{\text{imp}} = \mathbf{M}_{\text{m}}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_{\text{d}}) + \mathbf{D}_{\text{m}}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_{\text{d}}) + \mathbf{K}_{\text{m}}(\mathbf{x} - \mathbf{x}_{\text{d}}) \quad (2.2)$$

where $\mathbf{M}_{\text{m}}, \mathbf{D}_{\text{m}}, \mathbf{K}_{\text{m}}$ are user-designed matrices for inertia, damping, and stiffness, respectively. $\mathbf{x}_{\text{d}}, \dot{\mathbf{x}}_{\text{d}}, \ddot{\mathbf{x}}_{\text{d}}$ are the desired position, velocity, and acceleration, respectively in Cartesian space, while $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$ are the actual robot position, velocity, and acceleration, respectively. $\mathbf{F}_{\text{imp}} \in \mathbb{R}^6$ is the interacting wrench (force and torque) between the robot EE and the environment in Cartesian space.

To avoid the requirement for external force measurement, let us set the designed inertia matrix equal to the inherent inertia matrix of the robot, *i.e.*, $\mathbf{M}_{\text{m}} = \mathbf{M}_{\text{x}}$, where $\mathbf{M}_{\text{x}}$ is the inherent inertia of the robot in Cartesian space and $\mathbf{M}_{\text{x}} = \mathbf{J}^{-\text{T}}\mathbf{M}\mathbf{J}^{-1}$ [141]. By substituting (2.2) into (2.1) using $\mathbf{F}_{\text{ext}} = \mathbf{F}_{\text{imp}}$ and setting $\mathbf{M}_{\text{m}} = \mathbf{M}_{\text{x}}$, the first simplified version of impedance control law (V1) can be expressed as

$$\boldsymbol{\tau}_{\text{imp}} = \mathbf{M}\mathbf{J}^{-1}(\ddot{\mathbf{x}}_{\text{d}} - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{S}\dot{\mathbf{q}} + \mathbf{g} + \mathbf{J}^{\text{T}}[\mathbf{D}_{\text{m}}(\dot{\mathbf{x}}_{\text{d}} - \dot{\mathbf{x}}) + \mathbf{K}_{\text{m}}(\mathbf{x}_{\text{d}} - \mathbf{x})] \quad (2.3)$$

where $\mathbf{J}^{-1}$ will be replaced with the pseudo-inverse of the Jacobian $\mathbf{J}^{\#} = \mathbf{J}^{\text{T}}(\mathbf{J}\mathbf{J}^{\text{T}})^{-1}$ when $\mathbf{J}$ is not a square matrix.

In order to represent a real mechanical system, a Coriolis and centrifugal term can be included into the impedance model (2.2). Accordingly, the augmented impedance model is

$$\mathbf{F}_{\text{imp}} = \mathbf{M}_{\text{x}}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_{\text{d}}) + (\mathbf{S}_{\text{x}} + \mathbf{D}_{\text{m}})(\dot{\mathbf{x}} - \dot{\mathbf{x}}_{\text{d}}) + \mathbf{K}_{\text{m}}(\mathbf{x} - \mathbf{x}_{\text{d}}) \quad (2.4)$$

where $\mathbf{S}_{\text{x}}$ is the Coriolis and centrifugal matrix of the robot in Cartesian space and $\mathbf{S}_{\text{x}} = \mathbf{J}^{-\text{T}}\mathbf{S}\mathbf{J}^{-1} - \mathbf{M}_{\text{x}}\dot{\mathbf{J}}\mathbf{J}^{-1}$. By substituting (2.4) into (2.1) using $\mathbf{F}_{\text{ext}} = \mathbf{F}_{\text{imp}}$, the second simplified version of impedance control law (V2) can be expressed as

$$\boldsymbol{\tau}_{\text{imp}} = \mathbf{M}\mathbf{J}^{-1}(\ddot{\mathbf{x}}_{\text{d}} - \dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{x}}_{\text{d}}) + \mathbf{S}\mathbf{J}^{-1}\dot{\mathbf{x}}_{\text{d}} + \mathbf{g} + \mathbf{J}^{\text{T}}[\mathbf{D}_{\text{m}}(\dot{\mathbf{x}}_{\text{d}} - \dot{\mathbf{x}}) + \mathbf{K}_{\text{m}}(\mathbf{x}_{\text{d}} - \mathbf{x})] \quad (2.5)$$

Furthermore, for a set-point regulation problem, we set $\ddot{\mathbf{x}}_{\text{d}} = \dot{\mathbf{x}}_{\text{d}} = \mathbf{0}$. Then, the impedance control law V2 (2.5) can be simplified to V3 as expressed by (2.6), which

is also known as task-space PD controller with gravity compensation.

$$\boldsymbol{\tau}_{\text{imp}} = \mathbf{J}^T[\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \mathbf{g} \quad (2.6)$$

The relationships among the three versions of the impedance control law V1, V2, and V3 are summarized as follows. All of the three versions avoid measuring external force by setting $\mathbf{M}_m = \mathbf{M}_x$. V1 is based on a general impedance model that is widely used in robotics control systems. V2 is based on an augmented impedance model that may represent a real mechanical system by including a Coriolis and centrifugal term into the model as shown in (2.4). V3 focuses on a specific set-point regulation problem based on V2. Therefore, V2 can be viewed as a bridge from V1 to V3, while V2 itself also represents an important simplification. For the subsequent simulations and experiments, only V2 and V3 are employed such that V2 can perfectly reduce to V3 when a set-point regulation problem is encountered. V1 was also implemented for testing but not reported here since no difference was found between V1 and V2 in terms of task performance.

2.2.2 Nonlinear disturbance observer

A nonlinear disturbance observer (NDOB) can be used to estimate all dynamic uncertainties as a lumped term [94], comprised of two categories. One is the dynamic model inaccuracy caused by inaccurate dynamic parameters (*e.g.*, friction, center of mass location, and link weight), while the other is the external disturbance (*e.g.*, extra payload attached onto the robot body, and robot-environment interaction force). From an analytical perspective, in the absence of external disturbance, the NDOB estimates the difference between the real dynamic model and the estimated dynamic model of the robot, and it can be expressed as

$$\begin{aligned} \boldsymbol{\tau}_{\text{NDOB}} &= -[(\mathbf{M} - \hat{\mathbf{M}})\ddot{\mathbf{q}} + (\mathbf{S} - \hat{\mathbf{S}})\dot{\mathbf{q}} + (\mathbf{g} - \hat{\mathbf{g}})] \\ &= -\Delta\mathbf{M}\ddot{\mathbf{q}} - \Delta\mathbf{S}\dot{\mathbf{q}} - \Delta\mathbf{g} \end{aligned} \quad (2.7)$$

where $\hat{\mathbf{M}}$, $\hat{\mathbf{S}}$, $\hat{\mathbf{g}}$ are the estimations on $\mathbf{M}$, $\mathbf{S}$, $\mathbf{g}$, respectively.

An adapted NDOB design based on [18, 94, 138] is used in this chapter which can be expressed as

$$\begin{cases} \mathbf{L} = \mathbf{Y}\hat{\mathbf{M}}^{-1} \\ \mathbf{p} = \mathbf{Y}\dot{\mathbf{q}} \\ \dot{\mathbf{z}} = -\mathbf{L}\mathbf{z} + \mathbf{L}(\hat{\mathbf{S}}\dot{\mathbf{q}} + \hat{\mathbf{g}} - \boldsymbol{\tau} - \mathbf{p}) \\ \boldsymbol{\tau}_{\text{NDOB}} = \mathbf{z} + \mathbf{p} \end{cases} \quad (2.8)$$

where $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the observer gain matrix, $\mathbf{Y} \in \mathbb{R}^{n \times n}$ is a constant invertible matrix needs to be designed, $\hat{\mathbf{M}}$ is designed to be a symmetric and positive definite matrix and thus invertible, $\mathbf{z}$ is an auxiliary variable, $\mathbf{p}$ is an auxiliary vector determined from $\mathbf{Y}$, $\boldsymbol{\tau}_{\text{NDOB}}$ is the estimated lumped uncertainties via the NDOB observer. Note that the disturbance estimation error $\Delta\boldsymbol{\tau}_{\text{NDOB}} = \boldsymbol{\tau}_{\text{NDOB}} - \hat{\boldsymbol{\tau}}_{\text{NDOB}}$, *i.e.*, the difference between the real lumped uncertainties and the estimated lumped uncertainties, is globally uniformly ultimately bounded for this NDOB design according to [94], and will converge asymptotically to zero if the rate of change of the lumped uncertainties is negligible. For simplicity, in this chapter, we assume $\Delta\boldsymbol{\tau}_{\text{NDOB}} = 0$, *i.e.*, the NDOB can accurately estimate the exact difference between the real dynamic model and the estimated dynamic model. Therefore, we denote the NDOB output as $\boldsymbol{\tau}_{\text{NDOB}}$ instead of $\hat{\boldsymbol{\tau}}_{\text{NDOB}}$ in (2.8).

In (2.8), the first two equations are the specific user design on the vector $\mathbf{p}$ and matrix $\mathbf{L}$. The third equation is to update the auxiliary variable $\mathbf{z}$. The advantage of the adapted NDOB design is that it does not require acceleration measurement. Considering that designing an observer is not the focus in this chapter, only summarized information about the adapted NDOB is introduced above. For a specific observer design methodology with detailed procedures, please refer to [94].

It is worth noting that there are some convergence properties in the original NDOB design in [94], and here we briefly rephrase them as the following,

Theorem 1 *The disturbance tracking error $\Delta\boldsymbol{\tau}_{\text{disturb}}$ converges exponentially to zero if the following conditions hold:*

(1) The matrix $\mathbf{Y}$ is invertible.

(2) There exists a positive definite and symmetric matrix $\mathbf{\Gamma}$, such that

$$\mathbf{Y} + \mathbf{Y}^T - \mathbf{Y}^T \dot{\mathbf{M}} \mathbf{Y} \geq \mathbf{\Gamma}.$$

(3) The rate of change of the lumped disturbance acting on the manipulator is negligible (i.e., $\dot{\boldsymbol{\tau}}_{\text{disturb}} \approx \mathbf{0}$), in comparison with the estimation error dynamics.

Theorem 2 The disturbance tracking error $\Delta \boldsymbol{\tau}_{\text{disturb}}$ is globally uniformly ultimately bounded if:

(1) The first two conditions of the [Theorem 1](#) hold.

(2) The rate of change of the lumped disturbance is bounded.

For the full original statements on the convergence properties of the NDOB (i.e., [Theorem 1](#) and [Theorem 2](#)), please refer to the original work [\[94\]](#).

2.2.3 Integration of impedance controller and NDOB

In this section, we mathematically explore the outcome of the combination of impedance controller and NDOB. Three scenarios of the combination are considered according to the variations of the impedance control law introduced earlier.

In practice, for both the impedance control law and NDOB, the estimations of the dynamic coefficient matrices, i.e., $\hat{\mathbf{M}}$, $\hat{\mathbf{S}}$, $\hat{\mathbf{g}}$ are used, due to the fact that their real values ($\mathbf{M}$, $\mathbf{S}$, $\mathbf{g}$) are unavailable.

Scenario 1, for impedance control law V1.

Using $\hat{\mathbf{M}}$, $\hat{\mathbf{S}}$, $\hat{\mathbf{g}}$, the combined outcome of the impedance control law V1 ([2.3](#)) and NDOB ([2.7](#)) can be given by

$$\hat{\boldsymbol{\tau}}_{\text{imp}} - \boldsymbol{\tau}_{\text{NDOB}} = \underbrace{\mathbf{M}\mathbf{J}^{-1}\{\ddot{\mathbf{x}}_{\text{d}} - \dot{\mathbf{J}}\dot{\mathbf{q}}\} + \mathbf{S}\dot{\mathbf{q}} + \mathbf{g} + PD}_{\boldsymbol{\tau}_{\text{imp}}} + \underbrace{\Delta\mathbf{M}\mathbf{J}^{-1}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_{\text{d}})}_{\text{residual}} \quad (2.9)$$

As shown in ([2.9](#)), the combined output is equal to the ideal impedance control law (i.e., the impedance control law assuming full knowledge of robot dynamics) $\boldsymbol{\tau}_{\text{imp}}$ in

(2.3) together with a residual term. This residual term is tending to zero as $\ddot{\mathbf{x}} \rightarrow \ddot{\mathbf{x}}_d$, *e.g.*, it is negligible when Cartesian acceleration tracking performance is accurate enough.

Scenario 2, for impedance control law V2.

Using $\hat{\mathbf{M}}$, $\hat{\mathbf{S}}$, $\hat{\mathbf{g}}$, the combined output of the impedance control law V2 (2.5) and NDOB (2.7) is calculated as

$$\hat{\boldsymbol{\tau}}_{\text{imp}} - \boldsymbol{\tau}_{\text{NDOB}} = \underbrace{\mathbf{M}\mathbf{J}^{-1}\{\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{x}}_d\} + \mathbf{S}\mathbf{J}^{-1}\dot{\mathbf{x}}_d + \mathbf{g} + PD}_{\boldsymbol{\tau}_{\text{imp}}} + \underbrace{\Delta\mathbf{M}(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_d) + \Delta\mathbf{S}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_d)}_{\text{residual}} \quad (2.10)$$

As shown in (2.10), the combined output is equal to the ideal impedance control law $\boldsymbol{\tau}_{\text{imp}}$ in (2.5) together with two residual terms. The two residual terms are tending to zero as $\ddot{\mathbf{q}} \rightarrow \ddot{\mathbf{q}}_d$ and $\dot{\mathbf{q}} \rightarrow \dot{\mathbf{q}}_d$, which means that they are negligible when the joint acceleration and velocity tracking are accurate enough.

Scenario 3, for impedance control law V3.

In this scenario, we assume that a steady state is achieved in set-point regulation, *i.e.*, $\dot{\mathbf{q}} = \ddot{\mathbf{q}} = \mathbf{0}$. Then using $\hat{\mathbf{M}}$, $\hat{\mathbf{S}}$, $\hat{\mathbf{g}}$, the combined output of the impedance control law V3 (2.6) and NDOB (2.7) is calculated as

$$\begin{aligned} \hat{\boldsymbol{\tau}}_{\text{imp}} - \boldsymbol{\tau}_{\text{NDOB}} &= \mathbf{J}^T[\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \hat{\mathbf{g}} - (-\mathbf{g} + \hat{\mathbf{g}}) \\ &= \underbrace{\mathbf{J}^T[\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \mathbf{g}}_{\boldsymbol{\tau}_{\text{imp}}} \end{aligned} \quad (2.11)$$

As shown in (2.11), the combined output is exactly equal to the ideal impedance control law $\boldsymbol{\tau}_{\text{imp}}$ in (2.6). This means that the NDOB can accurately compensate for inaccuracies in the estimation of the gravity term when the steady state is achieved in set-point regulation.

An integrated framework of impedance control and NDOB is proposed based on the analyses presented above. The block diagram of the control system is shown in Figure 2.1 which incorporates the proposed framework.

Table 2.1: Parameterization for simulation, experiment, and application of p HRI on elbow arthroscopic surgery simulator.

Parameters	Simulation	Experiment	p HRI Application
Spring	$\mathbf{K}_m = 7.5\mathbf{I}$		$\mathbf{K}_m = \mathbf{0}$
Damping	$\mathbf{D}_m = 2\sqrt{7.5}\mathbf{I}$		$\mathbf{D}_m = 7.5\mathbf{I}$
Inertia matrix		$\hat{\mathbf{M}} = 1.0 \times 10^{-3} \times \mathbf{I}$	
Observer gain		$\mathbf{Y} = 9.58 \times 10^{-3} \times \mathbf{I}$	

Note: $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ denote identity matrix.

2.3 Simulations, Experiments, and Results

2.3.1 Robotic system

A 3-DOF PHANToM Premium 1.5A robot (3D Systems, Inc., Cary, NC, USA) is used for simulations and experiments in this chapter. For the simulations, we reconstruct the kinematic model and dynamic model of the PHANToM robot based on [16] and conduct the simulations using MATLAB/Simulink (version R2017a, MathWorks Inc., Natick, MA, USA). For the experiments, the physical robot is controlled via joint torque command, which is sent from MATLAB/Simulink using Quarc real-time control software (Quanser Inc., Markham, ON, Canada). The control rate of the robot is 1,000 Hz. The MATLAB/Simulink and Quarc software run on a computer with a 3.33 GHz Intel(R) Core(TM) 2 i5 CPU with a Windows 7 Enterprise 64-bit operating system.

2.3.2 Parameterization

For all simulations and experiments in the remaining part of this chapter, the parameter values used in the impedance model and NDOB are listed in Table 2.1. In order to have a simple but natural movement, a circular and cyclic trajectory is selected for the simulations and experiments, which can be expressed as a function of time as

the following

$$\begin{cases} x_d = R \sin(\frac{2\pi}{t_1}t) \\ y_d = R \cos(\frac{2\pi}{t_1}t) + R \\ z_d = 0 \end{cases} \quad (2.12)$$

where $R = 0.02$ m is the radius of the circle and $t_1 = 5$ s is the period for generating a full cycle.

In the following sub-sections, three conditions related to the combination of impedance controller and NDOB are presented. Both simulation and experimental results are included for each condition. A [video](#) demonstration for the experiments is available online¹.

2.3.3 Condition 1: Constant payload

Simulation

In Condition 1, a constant payload of 22 g is attached onto the robot EE as external disturbance, and there is no dynamic model inaccuracy involved. A circle tracking task in free space is employed and the circle trajectory is given by (2.12). This condition is performed by implementing the impedance control law V2 and NDOB, *i.e.*, the Scenario 2 given by (2.10). Four cases are designed for this condition:

- Case-00: External disturbance *OFF*, NDOB *OFF*;
- Case-01: External disturbance *OFF*, NDOB *ON*;
- Case-10: External disturbance *ON*, NDOB *OFF*;
- Case-11: External disturbance *ON*, NDOB *ON*.

The trajectory tracking results of the simulations are shown in Figure 2.2. Note that no dynamic model inaccuracy is involved in the simulations, which means that the only dynamic uncertainty is the external disturbance of the constant payload. As can be seen in Figure 2.2a, 2.2b, without the disturbance of the constant payload,

¹online demo video link: <https://youtu.be/f54Iah0yuWk>

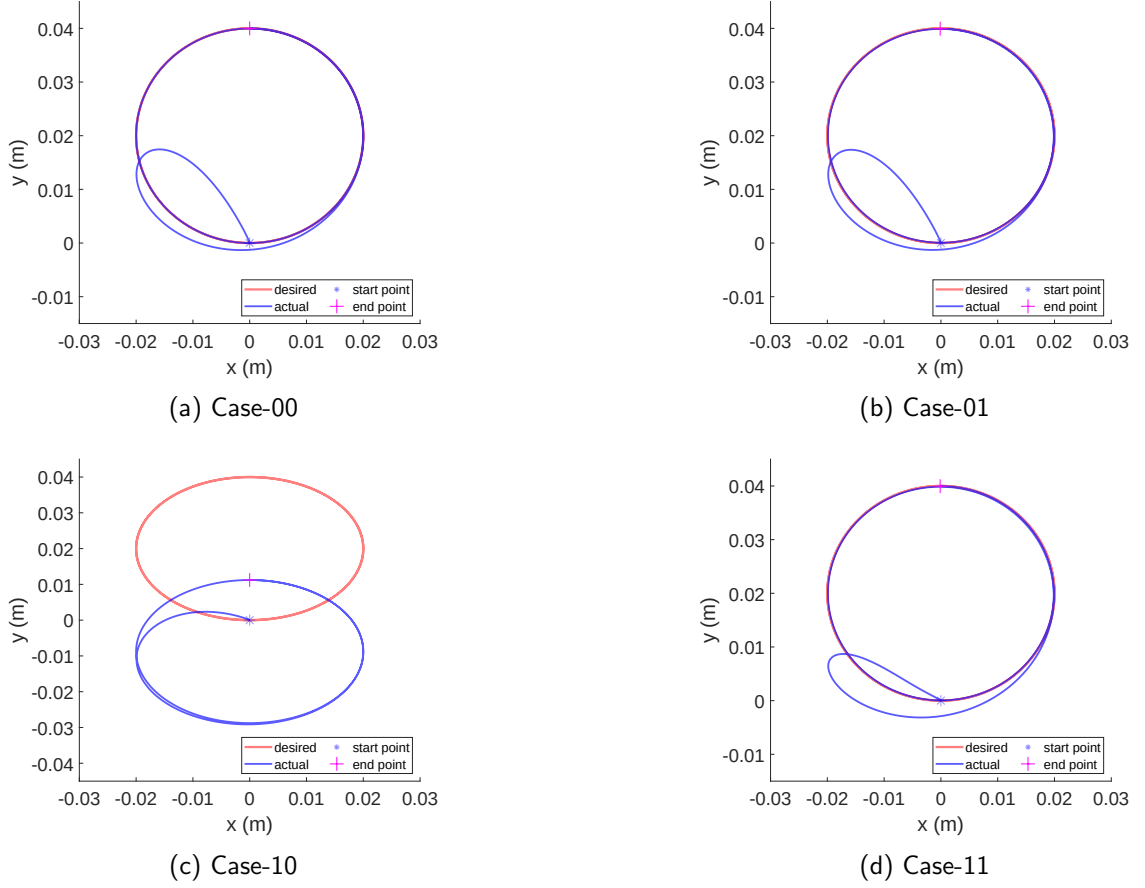


Figure 2.2: Simulation results of the four cases of a circle tracking task in free space in Condition 1.

the tracking results are the same no matter the NDOB is implemented or not. When the payload is attached as shown in Figure 2.2c, the actual tracking trajectory deviates largely from the desired one if the NDOB is not implemented. However, by implementing a NDOB as shown in Figure 2.2d, the disturbance from the constant payload can be fully detected and compensated for, thus good tracking performance is recovered. Note that there is a deviation between the actual trajectory and the desired trajectory at the beginning of the task. The reason is that their initial positions are not exactly the same. To this point, a low-pass infinite impulse response (IIR) filter can be implemented as necessary to ensure the smoothness of the movement at the beginning of the task.

Experiment

Similar to the simulations, experiments on the four cases are conducted by implementing the impedance controller and NDOB on a 3-DOF PHANToM robot. The experimental results of the four cases are shown in Figure 2.3.

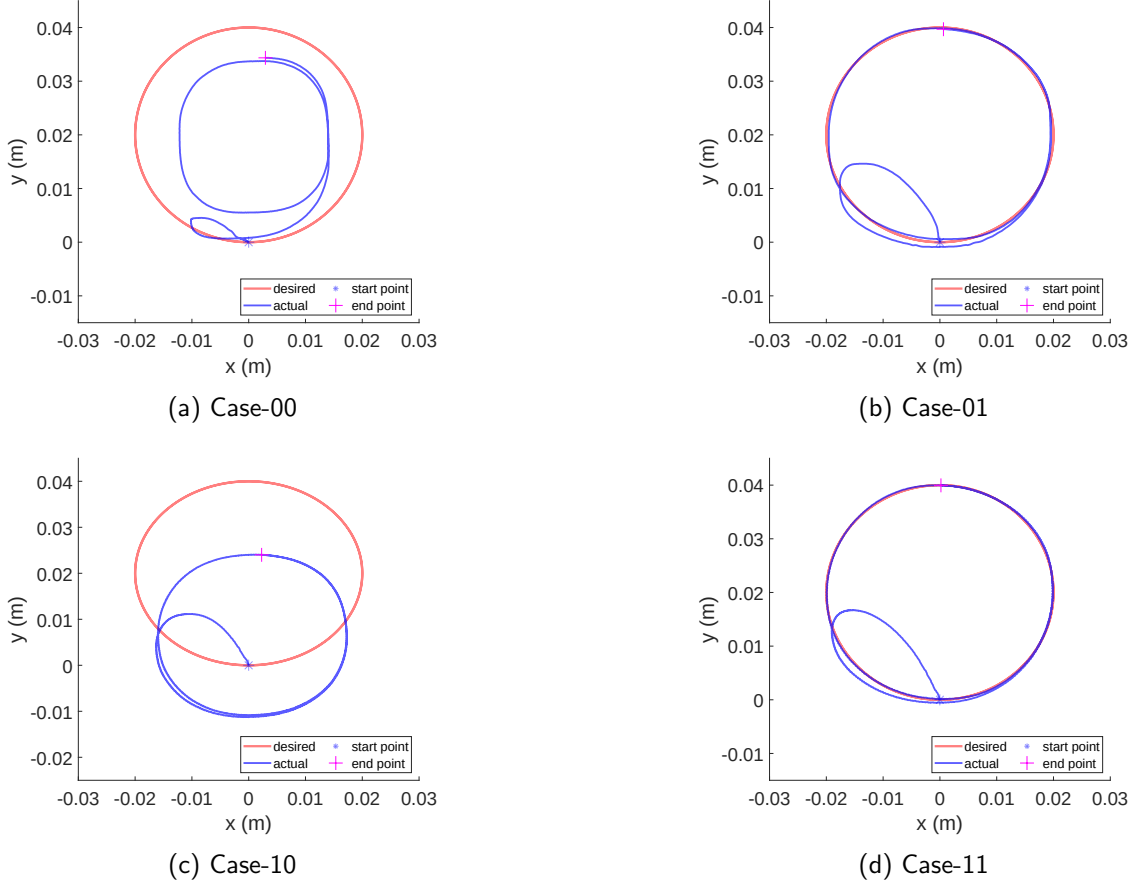


Figure 2.3: Experimental results of the four cases of a circle tracking task in free space in Condition 1.

As can be seen in Case-00 (Figure 2.3a), the circle tracking performance is poor. The reason is that dynamic model inaccuracies do exist (*e.g.*, joint friction) for the physical robot. As expected, when a NDOB is implemented in Case-01, this dynamic model inaccuracy is compensated for, and tracking performance is back to normal as shown in Figure 2.3b. For Case-10 and Case-11, the experimental results similar to the corresponding simulation are obtained, *i.e.*, without NDOB, the tracking task fails due to the external payload as well as the model inaccuracy (Figure 2.3c), while with NDOB, good tracking performance is recovered (Figure 2.3d). Note that, here

the NDOB in Case-11 has detected both the dynamic model inaccuracy (*e.g.*, joint friction) and the external disturbance (*i.e.*, the constant payload).

2.3.4 Condition 2: Time-varying payload

Simulation

In the simulation part of Condition 2, two types of time-varying payload are investigated. One is sinusoidal payload while another is a suddenly added constant payload. The former is to simulate a scenario of time-varying contact force during bone debridement while the latter is to simulate a scenario of sudden contact force when the tool hits a bone.

The same circle tracking task (2.12) is employed, whereas only Case-10 and Case-11 are considered. This condition is also performed by implementing the impedance control law V2 and NDOB, *i.e.*, the Scenario 2 given by (2.10). No dynamic model inaccuracy is involved in the simulations here.

The sinusoidal payload is applied onto the robot EE as external disturbance which can be expressed by

$$\begin{cases} Fx_d = a_1 \sin(\frac{2\pi}{t_1}t) \\ Fy_d = a_2 \sin(\frac{2\pi}{t_2}t) \\ Fz_d = a_3 \cos(\frac{2\pi}{t_3}t) \end{cases} \quad (2.13)$$

where $t_1 = 2$, $t_2 = 5$, $t_3 = 2$ are cycles of the desired time-varying EE payload for each axis in units of second, and $a_1 = 0.01$, $a_2 = 0.2$, $a_3 = 0.01$ are the corresponding amplitudes in units of Newton.

The simulation results of Condition 2 with a sinusoidal payload are shown in Figure 2.4. As can be seen in Figure 2.4a (Case-10), without NDOB, the tracking performance is distorted due to the sinusoidal payload. By implementing NDOB, normal tracking performance is obtained as shown in Figure 2.4b (Case-11). The tracking performance with NDOB in Case-11 is shown in Figure 2.4c, and the disturbance estimation on the external payload from the NDOB is shown in Figure 2.4d.

The simulation results of Condition 2 with a suddenly added payload are shown

in Figure 2.5. In this simulation, a constant payload of 22 g is attached at the robot EE throughout the task, while another payload of 23 g is suddenly added onto the robot EE at 5.5 s and remains there since then. This procedure can be easily identified in Figure 2.5a where the actual tracking trajectory is deviated due to the two payloads when NDOB is not activated. When the NDOB is activated, the tracking performance is well-recovered as shown in Figure 2.5b and Figure 2.5c. And the NDOB can immediately and accurately estimate the suddenly added payload as shown in Figure 2.5d. The simulation results in Condition 2 demonstrate that the NDOB is able to accurately estimate the external time-varying disturbances in real-time as well.

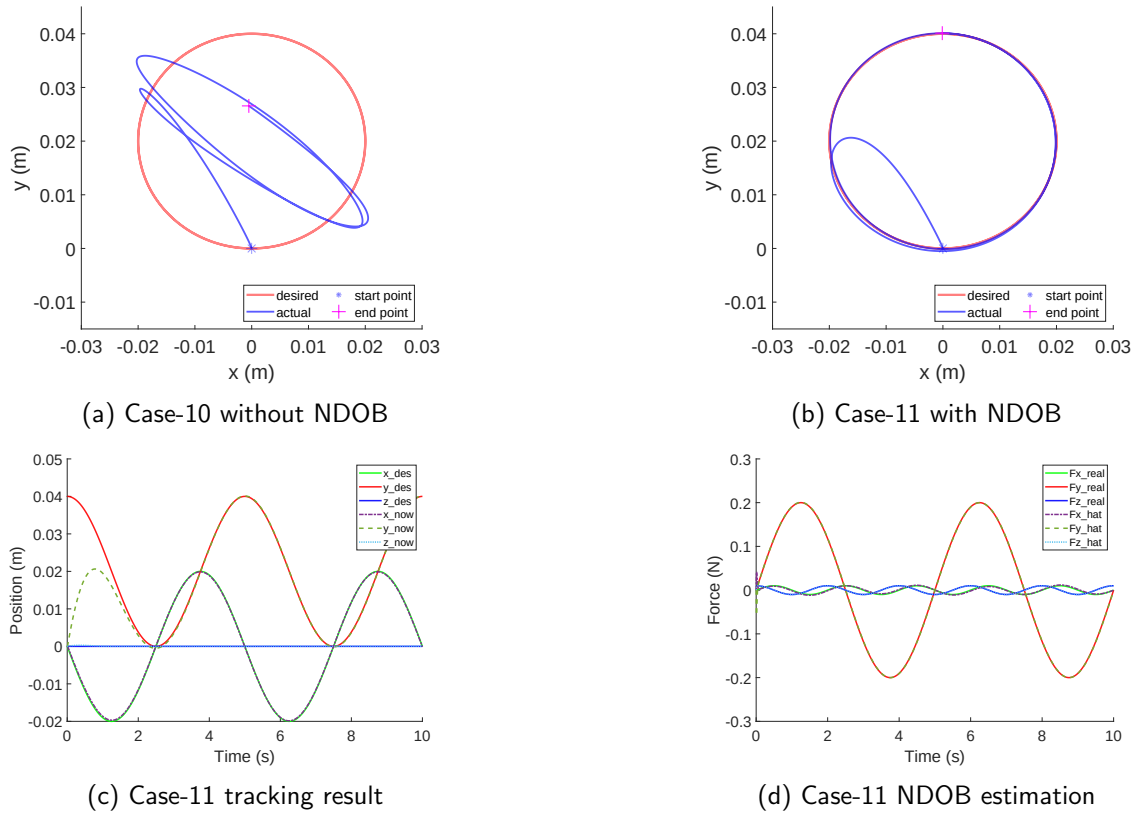


Figure 2.4: Simulation results of a circle tracking task with sinusoidal time-varying payload in Condition 2.

Experiment

For the experiment part of Condition 2, since it is impossible to apply a sinusoidal

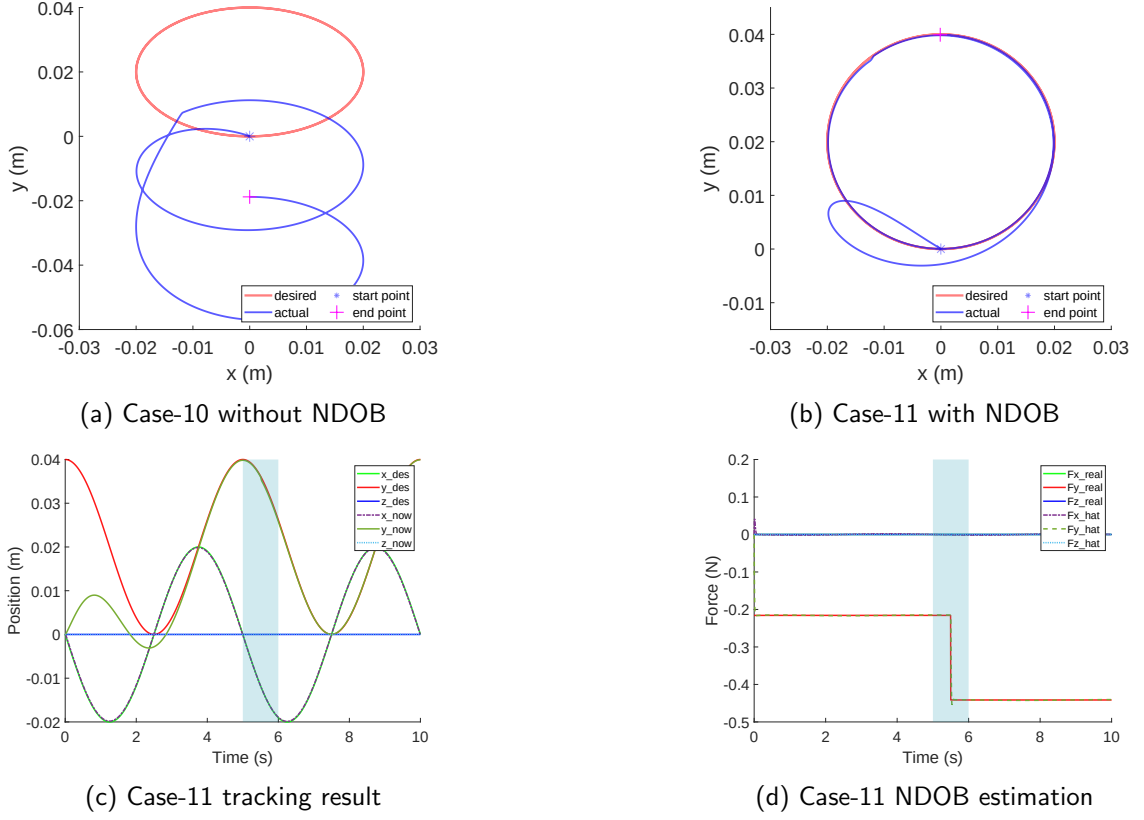


Figure 2.5: Simulation results of a circle tracking task with a suddenly added payload (23 g) in Condition 2. Note that the 23 g payload is added at 5.5 s and remains there since then.

payload of (2.13) in practice, only an experiment with the suddenly added payload is performed. In the experiment, an extra magnetic constant payload of 23 g is added onto the robot EE in the middle of the circle tracking task. Note that a constant payload of 22 g is always attached at the robot EE in this experiment.

The experimental results of Condition 2 are shown in Figure 2.6. As can be seen in Figure 2.6a, only a small perturbation is observed immediately after the extra magnetic payload is added and good tracking performance is recovered very quickly. In Figure 2.6b, the estimated force component F_y from NDOB is not piecewise constant compared with its corresponding simulation in Figure 2.5d. The main reason should be that the estimation from NDOB in the physical experiment (Figure 2.6b) involves both dynamic model inaccuracy and external payloads, while in the simulation (Figure 2.5d) it involves only external payloads.

The experimental results in Condition 2 indicate that the NDOB is able to immediately detect and accurately compensate for a time-varying payload, thus protect the tracking performance from being affected. This verified the corresponding simulation results.

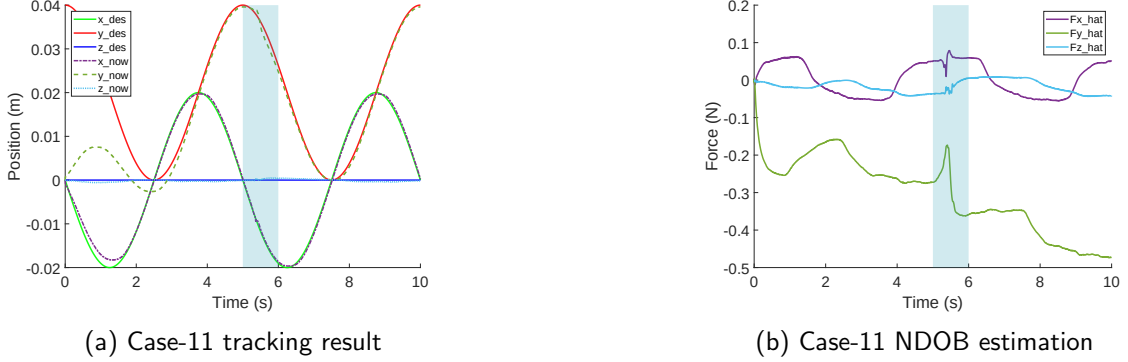


Figure 2.6: Experimental results of a circle tracking task with a suddenly added payload (23 g) in Condition 2. An extra magnetic payload of 23 g is added onto the robot EE at some time point during 5-6 s (the colored area) in practice, and it remains there since then.

2.3.5 Condition 3: Set-point regulation

Simulation

In Condition 3, a set-point regulation problem is explored. The coordinates of the initial point and the desired set-point in Cartesian space are $[0, 0, 0]$ m and $[0.01, 0.04, 0]$ m, respectively. This condition is performed by implementing the impedance control law V3 and NDOB, *i.e.*, the Scenario 3 given by (2.11). Only Case-10 and Case-11 are considered. For both simulation and experiment in Condition 3, a constant payload of 22 g is always attached at the robot EE. Additionally, during the set-point regulation, an external disturbance force in a range of $[0, 2]$ N is applied onto the robot EE along $y+$ axis. In the simulation, the external disturbance is designed by (2.13) with $a_1=a_3=0$, $a_2=t_2=2$, which means that this time-varying disturbance is along y -axis and the maximum amplitude is 2 N. The external disturbance is applied only in the period of 6-7 s in the simulation.

The simulation results of Condition 3 are shown in Figure 2.7. As can be seen in

Case-10 (Figure 2.7a), without NDOB, the regulation is failed with a large constant error between the desired (solid red line) and actual (dashed green line) set-point along y -axis. Here note that, even if with a higher stiffness in the impedance controller in Case-10, a constant error will still be remained if the gravity term cannot be appropriately compensated for. With NDOB in Case-11 (Figure 2.7b), the regulation task achieves good performance.

For the external disturbance during 6-7 s (the colored area in Figure 2.7), without NDOB, a large deviation as high as 14 cm is observed. But with NDOB, the deviation is significantly reduced to be less than 0.5 cm. This confirms the NDOB capability for disturbance rejection. The constant payload and the extra 1s-disturbance can be appropriately compensated for by using NDOB.

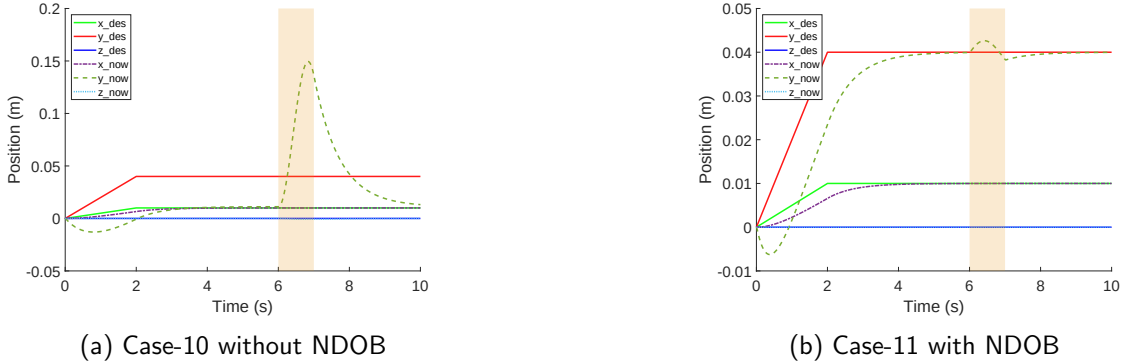


Figure 2.7: Simulation results of a set-point regulation task with a constant payload and 1s-disturbance in Condition 3. Note that, during the first two seconds, the desired point position is linearly increased to the final desired point position to ensure a smooth robot behavior at the moment of startup.

Experiment

For the corresponding experiment in Condition 3, the 1s-disturbance in a range of $[0, 2]$ N is exerted by a human user. Therefore, it is not exactly occurred during 6-7 s and not strictly lasted for 1 s in practice. The experimental results are shown in Figure 2.8. As can be seen from the figure, the NDOB is able to minimize the effect induced by the external 1s-disturbance, which experimentally verified the corresponding simulation results. The results of Condition 3 indicate that the NDOB

is also able to perform external disturbance rejection. From another perspective, the NDOB might undermine the compliant behavior brought by the impedance controller to some extent due to the disturbance rejection effect.

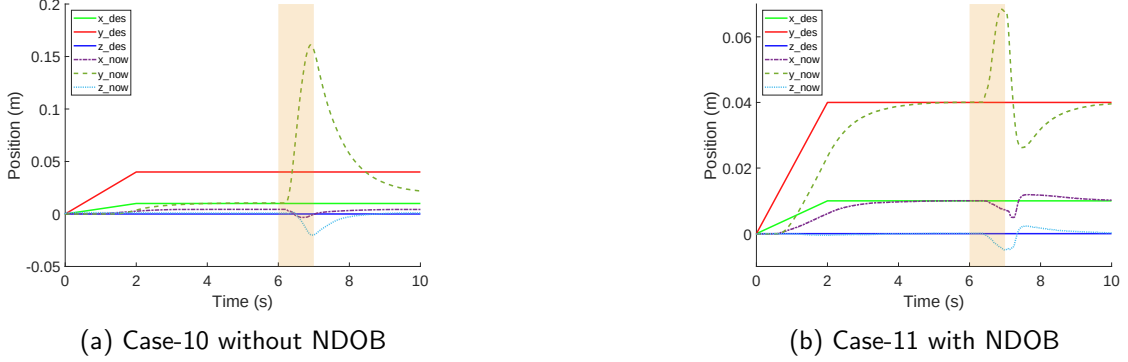


Figure 2.8: Experimental results of a set-point regulation task with a constant payload and 1s-disturbance in Condition 3. Note that the 1s-disturbance here is not exactly occurred during 6-7 s for 1 s since it is controlled by a human user.

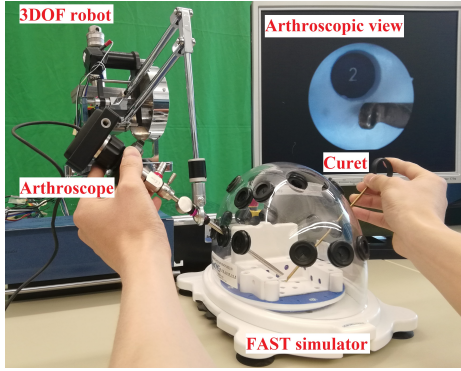
2.4 Application on Elbow Arthroscopic Surgery

The simulation and experimental results of the three conditions in the previous section demonstrate that the NDOB is capable of accurately estimating dynamic uncertainties of both constant (Condition 1) and time-varying (Condition 2) payloads attached at the robot EE, and performs suitable compensation in the control system as necessary. The NDOB has an intrinsic property to reject external disturbance no matter the disturbance is expected or not (*e.g.*, the 1s-disturbance in Condition 3). However, some human-generated disturbance, *e.g.*, human-robot interaction force, may be expected thus shouldn't be rejected. To solve this problem, NDOB-online and NDOB-offline are designed as shown in Figure 2.1. With NDOB-online, a lumped dynamic uncertainties is estimated and compensated in real time, *i.e.*, all uncertainties will be rejected as disturbance. With NDOB-offline, only an appropriate constant gravity is compensated for which enables the human-robot interaction. Please note that this online/offline design is just what we needed since we do not want the robot

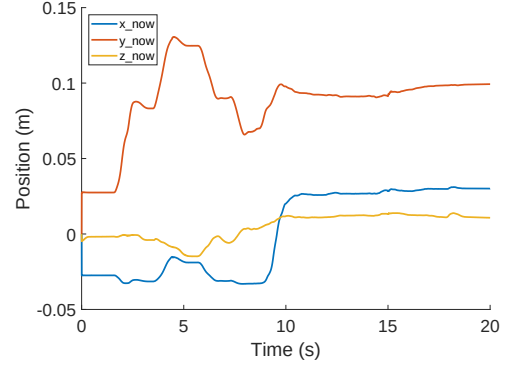
to be compliant to occasional external disturbances when it is holding the tool still. One limitation here is that rigorous stability analysis has to be conducted in future work since controller online-switching is involved.

In a specific application on elbow arthroscopic surgery, a robot will hold the arthroscope as an auxiliary supporter to reduce the surgeon's fatigue. More specifically, when the surgeon moves the arthroscope, the robot should provide compliant behavior while complying with the external human-robot interaction force but compensating all other uncertainties. This can be realized via NDOB-offline. On the other hand, when the arthroscope is left in an unattended state, the robot should keep it stably stay there and be able to reject any external disturbance. This can be realized via NDOB-online. This illustrates the integrated framework in our specific application scenario. To this end, a practical application on the robot-assisted arthroscopic surgery is developed to assess the framework. In the application, it is expected that the proposed control system allows the surgeon to move the arthroscope freely and keeps the arthroscope stay still wherever the surgeon left it. A FAST simulator, as shown in Figure 2.9a, is employed in the application where FAST is short for Fundamentals of Arthroscopic Surgery Training which is a commonly used physical model for training novices with their arthroscopic surgical skills. By implementing the framework, the arthroscope can move freely and stay in the air as shown in Figure 2.9b.

Although this is a preliminary experiment, it is an important step towards constructing a robot-assisted elbow arthroscopic system. The results of this chapter enable the system to accurately estimate and compensate for the robot dynamic uncertainties for a stable control system. Also, the robot holds the arthroscope for the surgeons such that they can focus on other tasks during the surgery. In future work, the tool pose tracked by the marker-free robot system can be visualized and provided to the surgeons in real-time. More practically, the robot-assisted system can be used either for a training simulator by integrating with a virtual/phantom patient body



(a) Setup



(b) Free trajectory

Figure 2.9: Setup and free trajectory for implementing the proposed framework. FAST, fundamentals of arthroscopic surgery training. The robot and arthroscope are connected via a ball joint.

via AR/VR techniques, or in real elbow arthroscopic surgery. The system is also promising to help guide the surgeon to perform tasks precisely and accurately, and to expand the application of elbow arthroscopy into various elbow conditions.

2.5 Conclusion

Robot-assisted arthroscopic surgery is a thriving field in orthopaedics. To build control system for a surgeon-assist arthroscope-holding robot, dynamic uncertainty is a critical point that needs to be estimated and compensated for during p HRI to ensure the system stability and accuracy. This chapter proposed an integrated framework of combining impedance control and nonlinear disturbance observer (NDOB). The combined outcomes were mathematically analyzed. Three common conditions were presented to evaluate the combined outputs by both simulations and experiments on a 3-DOF PHANToM robot. A preliminary application of p HRI on elbow arthroscopic surgery simulator was realized by implementing the proposed framework, and its effectiveness was demonstrated. The core contribution of this chapter is that, by combining impedance control and NDOB, the integrated framework can achieve an accurate impedance control under condition of that external uncertainties exist while only roughly estimated dynamic parameters are known.

The integrated framework is able to provide compliant behavior by the impedance controller during p HRI while compensating for the robot dynamic uncertainties by the NDOB. The NDOB used in this chapter estimates a lumped value for model inaccuracy and external disturbance, and is not able to separate them. In future work, we will try to separate the interaction force from the lumped term by introducing additional observers or learning techniques into our system.

Chapter 3

Neural Network Learning on Robot Dynamic Uncertainties¹

In **Chapter 2**, we have shown that integrating an impedance controller with a disturbance observer can achieve accurate impedance control. However, it works well for robots in free motion but not in robot-environment interaction. Although a disturbance observer is able to accurately estimate the dynamic uncertainties, the estimation is lumped uncertainties that contain all uncertainty sources including both the internal and the external disturbances. Without separating these two parts, the method of combining an impedance controller and an observer will result in the human-applied force being cancelled instead of interacting with the robot. To solve this problem in this Chapter, we propose a framework for learning the internal disturbances and separating the external disturbances by integrating three entities: an impedance controller, a neural network (NN) model, and a disturbance observer. In the framework, the impedance controller provides compliant robot behavior, while the observer captures the lumped uncertainties, and the NN learns to separate the external disturbances. Simulation results of an application scenario with an obstructive virtual fixture demonstrate the effectiveness of the proposed framework.

¹A version of this chapter has been published as: Teng Li, Armin Badre, Hamid D. Taghirad, Mahdi Tavakoli, "Neural Network Learning of Robot Dynamic Uncertainties and Observer-based External Disturbance Estimation for Impedance Control", 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2023), June 28-30, Seattle, WA, USA, 2023, pp. 591-597. [\[IEEE Xplore\]](#)

3.1 Background

In the field of robot control, having an accurate dynamic model of a robot is fundamental for ensuring an accurate and stable control [47]. This is especially the case in medical robotic systems where accuracy and safety are overriding concerns. However, accurate dynamic models only exist in theory but not in practice, which means that dynamic uncertainties in a robotic system are inevitable. Common dynamic uncertainties include joint friction, inaccurate center of mass location and link weight, and extra payloads attached to the robot body [47]. Therefore, estimation and compensation of dynamic uncertainties are critical in robot control.

Various observers have been developed to estimate and thus compensate for dynamic uncertainties [107, 19, 93, 52]. Typical approaches include generalized momentum observer (GMO) [52], extended state observer (ESO) [117], nonlinear disturbance observer (NDOB) [94], disturbance Kalman filter (DKF) method [57, 80], as well as their variations.

GMO, also known as classic first-order momentum observer, is one of the most commonly used observers due to its advantages of being simple and easy to implement [154]. It is often used as a reference when designing new observers [45, 57, 80]. ESO was originally proposed by Han in 1995 [54, 55]. Since then, many of its variations have been developed for different purposes such as collision detection [109] and interaction force estimation [117]. NDOB is designed specifically by considering the nonlinearity of the dynamics of robots which enabled it to have an advantage over the linear ones [18, 94, 152]. DKF is another novel method to estimate the dynamic uncertainties in the control system [57, 80]. The accuracy of its estimation is excellent, but the complexity of implementation could be a limitation for it to be widely used.

All of these observers estimate lumped uncertainties. While the lumped uncertainties do include various components (*e.g.*, joint friction), it is not possible to separate

that specific component out of the lumped estimate. Especially in human-robot interaction scenarios, the observer will take the human-exerted force as a part of uncertainties and thus reject it [75]. On the other hand, estimating contact force between human-robot or robot-environment is a critical problem in the field of human-robot interaction and the field of collision detection [52]. Without separating the human-applied force out of the lumped estimate, a disturbance observer will cancel the human-applied force and thus reject human-robot interaction. Finding a solution to this issue is important for robots with impedance control especially in medical robotic applications involving human-robot interaction.

Some methods have been developed to estimate the robot-environment interaction force by involving learning techniques. Hu and Xiong [57] developed a method to estimate external contact force using a semiparametric model and DKF. In their method, the semiparametric dynamic model containing a multilayer perceptron (MLP) neural network was used to provide a more accurate dynamic model, while a DKF was used to estimate the contact force between the robot and the environment. Similarly, an integrated framework of neural network (NN) and DKF was developed to estimate external contact force [80]. In the framework, an NN model was used to learn the joint friction, while a DKF observer was used to estimate the contact force. In another work, NN was used to approximate the global friction, while a momentum observer was used to estimate the external force. Additionally, a Kalman filter was employed to filter the measurement noise for a more accurate force estimation.

Sharifi *et al.* [124] employed a nonlinear autoregressive network with exogenous inputs (NARX) to learn and estimate the robot dynamics plus the passive dynamics of a user who wore an exoskeleton. Then the human-exoskeleton interaction force was separated out by subtracting the NARX-learned dynamics from the motor torques in the scenario of active user dynamics. Note that the movement trajectories of the exoskeleton is nearly periodic, which helped with training the NARX model to generate its model predictions.

Inspired by these methods, in this chapter, we propose a framework for training a NN model to learn the robot dynamic uncertainties so that we can later isolate the external disturbances. Specifically, the framework involves three entities, *i.e.*, an impedance controller, an NDOB, and an NN model, where the impedance controller provides compliance, the NDOB measures dynamic uncertainties plus external disturbance (*i.e.*, the lumped uncertainties), and the NN estimates the dynamic uncertainties. Thus, by subtracting the NN output from the NDOB output, the external disturbance (*e.g.*, human-applied force) can be isolated. A simulation of an application scenario is conducted to evaluate the effectiveness of the framework.

3.2 Methods

3.2.1 Robot dynamics and impedance control

A general dynamic model for an n -degree-of-freedom (DOF) rigid robot [41] can be given by

$$\underbrace{\mathbf{M}(\mathbf{q})}_{\hat{\mathbf{M}}+\Delta\mathbf{M}}\ddot{\mathbf{q}} + \underbrace{\mathbf{S}(\mathbf{q},\dot{\mathbf{q}})}_{\hat{\mathbf{S}}+\Delta\mathbf{S}}\dot{\mathbf{q}} + \underbrace{\mathbf{g}(\mathbf{q})}_{\hat{\mathbf{g}}+\Delta\mathbf{g}} + \boldsymbol{\tau}_{\text{fric}}(\dot{\mathbf{q}}) = \boldsymbol{\tau} + \underbrace{\boldsymbol{\tau}_{\text{ext}}}_{\mathbf{J}^T\mathbf{F}_{\text{ext}}} \quad (3.1)$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ denotes the inherent inertia matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes a matrix of the Coriolis and centrifugal forces, $\mathbf{g} \in \mathbb{R}^n$ represents the gravity vector. $\hat{\mathbf{M}}, \hat{\mathbf{S}}, \hat{\mathbf{g}}$ represent the user's model estimates, while $\Delta\mathbf{M}, \Delta\mathbf{S}, \Delta\mathbf{g}$ are the corresponding estimate errors. $\boldsymbol{\tau}_{\text{fric}} \in \mathbb{R}^n$ is joint friction, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the commanded joint torque vector, $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^n$ is the torque caused by external force, $\mathbf{F}_{\text{ext}} \in \mathbb{R}^6$ is the external force in Cartesian space, and $\mathbf{J} \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix.

A desired impedance model [129, 133, 75] for robot-environment contact can be expressed as

$$\mathbf{F}_{\text{imp}} = \mathbf{M}_{\mathbf{m}}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_{\text{d}}) + (\mathbf{S}_{\mathbf{x}} + \mathbf{D}_{\mathbf{m}})(\dot{\mathbf{x}} - \dot{\mathbf{x}}_{\text{d}}) + \mathbf{K}_{\mathbf{m}}(\mathbf{x} - \mathbf{x}_{\text{d}}) \quad (3.2)$$

where $\mathbf{M}_{\mathbf{m}}, \mathbf{D}_{\mathbf{m}}, \mathbf{K}_{\mathbf{m}}$ are user-designed matrices for inertia, damping, and stiffness, respectively. Note that $\mathbf{x}_{\text{d}}, \dot{\mathbf{x}}_{\text{d}}, \ddot{\mathbf{x}}_{\text{d}}$ are the desired position, velocity, and acceleration, respectively in Cartesian space, while $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$ are the actual ones. $\mathbf{S}_{\mathbf{x}}$ is the Coriolis

and centrifugal matrix of the robot in Cartesian space and $\mathbf{S}_x = \mathbf{J}^{-T}\mathbf{S}\mathbf{J}^{-1} - \mathbf{M}_x\dot{\mathbf{J}}\mathbf{J}^{-1}$, where $\mathbf{M}_x = \mathbf{J}^{-T}\mathbf{M}\mathbf{J}^{-1}$ is the inherent inertia of the robot in Cartesian space [141].

To avoid the need for the measurement of external forces, the designed inertia matrix can be set as $\mathbf{M}_m = \mathbf{M}_x$. Then, to reach (3.2) as the closed-loop dynamics governing the robot-environment interaction ($\mathbf{F}_{\text{ext}} = \mathbf{F}_{\text{imp}}$) in an ideal scenario of no model errors and no joint friction, the impedance control law can be given by [75]

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{J}^{-1}(\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{x}}_d) + \mathbf{S}\mathbf{J}^{-1}\dot{\mathbf{x}}_d + \mathbf{g} + \mathbf{J}^T[\mathbf{D}_m(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_m(\mathbf{x}_d - \mathbf{x})] \quad (3.3)$$

Note that when implementing the impedance controller (3.3) in practice, the estimates $\hat{\mathbf{M}}, \hat{\mathbf{S}}, \hat{\mathbf{g}}$ will be used since the accurate model of the robot is usually not accessible.

For robot impedance control around a fixed point in space, *i.e.*, set-point regulation, it has $\ddot{\mathbf{x}}_d = \mathbf{0}$, $\dot{\mathbf{x}}_d = \mathbf{0}$. Then, the impedance control law (3.3) can be simplified to (3.4), which is also known as task-space proportional-derivative (PD) controller with gravity compensation.

$$\boldsymbol{\tau} = \mathbf{J}^T[\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \mathbf{g} \quad (3.4)$$

3.2.2 Friction model

The joint friction in this chapter is modeled by the Stribeck model [94, 80] as expressed by (3.5), which is viewed as the most classical nonlinear expression that can include many characteristics of friction.

$$\boldsymbol{\tau}_{\text{fric}} = \mathbf{F}_c \text{sgn}(\dot{\mathbf{q}}) + (\mathbf{F}_s - \mathbf{F}_c) \text{sgn}(\dot{\mathbf{q}}) e^{-|\dot{\mathbf{q}}/\nu_s|^\sigma} + \mathbf{F}_v \dot{\mathbf{q}} \quad (3.5)$$

where $\mathbf{F}_c$ is the Coulomb friction, $\mathbf{F}_s$ is the static friction, $\mathbf{F}_v$ is the viscous friction, $\dot{\mathbf{q}}$ is the joint velocity, ν_s is the Stribeck parameter, σ is the exponent of the Stribeck nonlinearity and $\sigma = 2$ is employed in this chapter.

3.2.3 Disturbance observer

In order to organize all the disturbances together, the dynamic model (3.1) of a robot can be re-written as

$$\hat{\mathbf{M}}\ddot{\mathbf{q}} + \hat{\mathbf{S}}\dot{\mathbf{q}} + \hat{\mathbf{g}} = \boldsymbol{\tau} + \underbrace{\boldsymbol{\tau}_{\text{ext}} - [\boldsymbol{\tau}_{\text{fric}} + (\Delta\mathbf{M}\ddot{\mathbf{q}} + \Delta\mathbf{S}\dot{\mathbf{q}} + \Delta\mathbf{g})]}_{\boldsymbol{\tau}_{\text{dist}}} \quad (3.6)$$

where $\boldsymbol{\tau}_{\text{dist}}$ denotes the lumped uncertainties that usually include three main aspects, *i.e.*, (source-1) the model error $(\Delta\mathbf{M}\ddot{\mathbf{q}} + \Delta\mathbf{S}\dot{\mathbf{q}} + \Delta\mathbf{g})$, (source-2) the joint friction $\boldsymbol{\tau}_{\text{fric}}$, and the external disturbances $\boldsymbol{\tau}_{\text{ext}}$ which may involve (source-3) constant disturbance and (source-4) time-varying disturbance. The constant disturbance may be a constant payload attached to the robot end-effector (EE) or body, while time-varying disturbance may be robot-environment interaction forces such as human-applied forces during human-robot interaction. In this chapter, internal disturbances refer to the summed uncertainties from source-1,2, while external disturbances refer to the summed uncertainties from source-3,4. Disturbance observer is a commonly used tool to estimate the lumped uncertainties. Note that any other uncertainties beyond the four sources will also be included in the lumped uncertainties that are estimated by the observer.

NDOB is easy to be implemented and has the advantage of estimating the nonlinearities [94]. Therefore, NDOB is employed in this chapter to estimate the lumped uncertainties. An adapted NDOB design based on [94] is used in this chapter which can be expressed as

$$\begin{cases} \mathbf{L} = \mathbf{Y}\hat{\mathbf{M}}^{-1} \\ \mathbf{p} = \mathbf{Y}\dot{\mathbf{q}} \\ \dot{\mathbf{z}} = -\mathbf{L}\mathbf{z} + \mathbf{L}(\hat{\mathbf{S}}\dot{\mathbf{q}} + \hat{\mathbf{g}} - \boldsymbol{\tau} - \mathbf{p}) \\ \boldsymbol{\tau}_{\text{NDOB}} = \mathbf{z} + \mathbf{p} \end{cases} \quad (3.7)$$

where $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the observer gain matrix, $\mathbf{Y} \in \mathbb{R}^{n \times n}$ is a constant invertible matrix that needs to be designed, $\hat{\mathbf{M}}$ is designed to be a symmetric and positive definite matrix and thus invertible, $\mathbf{z}$ is an auxiliary variable, $\mathbf{p}$ is an auxiliary vector determined

from $\mathbf{Y}$, τ_{NDOB} is the estimated lumped uncertainties via the NDOB observer, *i.e.*, $\hat{\tau}_{\text{dist}} = \tau_{\text{NDOB}}$.

3.2.4 Neural network

A promising neural network (NN) structure for modeling nonlinear dynamic systems is the nonlinear autoregressive network with exogenous inputs (NARX), which is a recurrent dynamic network with a special feature of feeding its delayed output back as input. This special feature requires the NARX to be trained by using time-series data, which means that the training data set needs to be collected from a long period of one single continuous task, while concatenating datasets collected from a set of separate short-period tasks is usually not acceptable. This is because the learning effect will be distorted by the outliers occurring at the concatenating points. A common feed-forward neural network (fNN) does not have this requirement.

Considering that machine learning usually requires a large amount of data for training, while a single exciting robot trajectory over a long continuous period of time may not be easy to generate. Therefore, a fNN model as shown in Figure 3.1 is employed. For the fNN model, the training data can be collected from a set of separate short-period trajectory tracking tasks. Also, one of its variations, cascade-forward neural network (cNN), is used to compare the learning performance. The fNN and cNN have the same structure except that the cNN includes an additional connection from its input layer to each of the following layers.

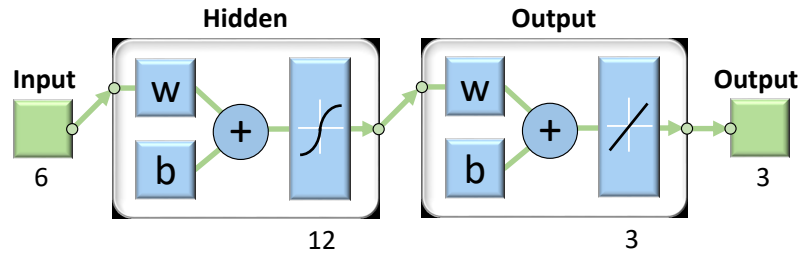


Figure 3.1: The structure of feed-forward neural network (fNN) model.

3.2.5 Collecting training and testing data

For training the fNN model (Figure 3.1), the robot EE position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$ will be taken as the inputs, while the estimate of lumped uncertainties from the NDOB will be taken as the learning targets. Since exciting trajectories are often employed for robot dynamic identification [47], the training data will be collected in a set of separate exciting trajectory tracking tasks. In the tasks, the robot is controlled by an impedance controller (3.3) and NDOB observer (3.7), a simple addition of the two equations when doing the implementation [75].

First of all, a set of exciting trajectories are generated which include joint position $\mathbf{q}$, velocity $\dot{\mathbf{q}}$, and acceleration $\ddot{\mathbf{q}}$ with a sampling rate of 1,000 Hz. Each exciting trajectory is generated from a harmonic function with random parameters and lasts 20 seconds, thus every exciting trajectory is unique. In total, 39 exciting trajectories are generated among which 38 of them will be used for training while another for testing.

As mentioned earlier, the lumped uncertainties in this chapter will come from four sources, which can be categorized as internal disturbances from source-1 (model error) and source-2 (friction), and external disturbances from source-3 (constant payload) and source-4 (time-varying payload). Since this chapter focuses on simulations, all the uncertainties are fully known and can be precisely controlled.

In source-1, dynamic model error is controlled to be at four different levels, *i.e.*, 0%, 10%, 50%, 100%. The level of model error is tuned by value assignment on matrix $\hat{\mathbf{S}}$ and $\hat{\mathbf{g}}$, while the inertia matrix $\hat{\mathbf{M}}$ is fixed at $\hat{\mathbf{M}} = \text{diag}(0.001, 0.001, 0.001)$. For example, model error 10% means $\hat{\mathbf{S}} = 90\%\mathbf{S}$, $\hat{\mathbf{g}} = 90\%\mathbf{g}$, where $\hat{\mathbf{M}}$, $\hat{\mathbf{S}}$, $\hat{\mathbf{g}}$ will be used for the calculation in the impedance controller (3.3) and NDOB observer (3.7). Model error 100% means $\hat{\mathbf{S}} = \mathbf{0}$, $\hat{\mathbf{g}} = \mathbf{0}$.

In source-2, joint friction is controlled by (3.5). In source-3, a constant payload of 22 gram is controlled to be attached to the robot EE or not. In source-4, a time-

varying payload is controlled by a set of Fourier series functions given by

$$\begin{cases} F_x = a_1 \sin(\frac{2\pi n_1}{t_1}t) + a_2 \sin(\frac{2\pi n_2}{t_1}t - \phi_2) + a_3 \sin(\frac{2\pi n_3}{t_1}t - \phi_3) \\ F_y = a_1 \cos(\frac{2\pi n_1}{t_1}t) + a_2 \cos(\frac{2\pi n_2}{t_1}t - \phi_2) + a_3 \cos(\frac{2\pi n_3}{t_1}t - \phi_3) \\ F_z = a_1 \cos(\frac{2\pi n_1}{t_1}t) + a_2 \sin(\frac{2\pi n_2}{t_1}t - \phi_2) + a_3 \cos(\frac{2\pi n_3}{t_1}t - \phi_3) \end{cases} \quad (3.8)$$

where F_x, F_y, F_z are the time-varying payload expressed in Cartesian space, and $t_1 = 10$, $a_1 = 0.1$, $a_2 = 0.15$, $a_3 = 0.05$, $n_1 = 1$, $n_2 = 2$, $n_3 = 3$, $\phi_2 = \frac{1}{2}\pi$, $\phi_3 = \pi$.

In our previous work [75], it has been shown that by integrating an impedance controller and an observer, an accurate impedance control can be achieved in a trajectory tracking task when the actual velocity and acceleration converge to the desired ones. It is noteworthy that the disturbance tracking of the observer works independently from the trajectory tracking of the controller, which means that the observer estimation accuracy is not affected by the trajectory tracking accuracy. In other words, even if the trajectory tracking performance is poor (*e.g.*, due to fully or partially uncompensated disturbances), the observer can still accurately estimate the lumped uncertainties.

The training data are collected by running a trajectory tracking task on each of the 38 exciting trajectories with only internal disturbances (from source-1,2) involved. The testing data are collected by running the trajectory tracking task on a simple figure-eight trajectory as given by (3.9) and a new exciting trajectory with all of the internal disturbances (from source-1,2) and external disturbances (from source-3,4) involved. In the tasks both of collecting training and testing data, the robot is controlled by an impedance controller (3.3) and NDOB observer (3.7), where the former is used to accurately track the trajectory while the latter is used to estimate the lumped uncertainties and compensate them in the controller.

$$\begin{cases} x_d = R \sin(\frac{2\pi}{t_1}t) \cos(\frac{2\pi}{t_1}t) \\ y_d = R \sin(\frac{2\pi}{t_1}t) + R \\ z_d = 0 \end{cases} \quad (3.9)$$

where $R = 0.02$ m is the amplitude of the figure-eight trajectory, $t_1 = 5$ s is the period for generating a full cycle.

The fNN model is first trained on the training dataset and then tested on the two testing datasets individually. The training goal is that the fNN is expected to learn to estimate only internal disturbances (from source-1,2), thus the external disturbances (from source-3,4) can be separated out by subtracting the fNN prediction outcome from the lumped uncertainties estimated by the NDOB. Note that the NDOB observer can always be implemented to online estimate the lumped uncertainties but not necessarily to do the compensation in the controller. The procedures of the proposed learning framework are illustrated in Figure 3.2.

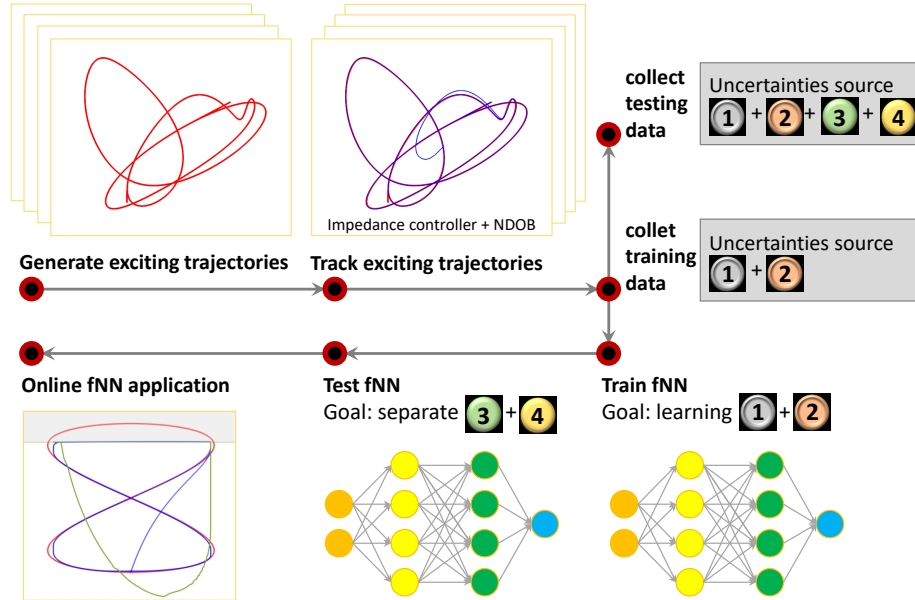


Figure 3.2: The procedures of the proposed framework of training fNN to learn internal disturbances and thus separate external disturbances. Uncertainties source-1, model error; source-2, friction; source-3, constant payload; source-4, time-varying payload.

3.3 Simulations, Validations, and Results

3.3.1 Robotic system

A 3-DOF PHANToM Premium 1.5A robot (3D Systems, Inc., Cary, NC, USA) is used for simulations in this chapter. To build a virtual model of this robot, the kinematic model and dynamic model of the PHANToM robot are reconstructed based

on [16]. All the simulations are conducted by using MATLAB/Simulink (version R2020a, MathWorks Inc., Natick, MA, USA), which is running on a computer with a 3.70 GHz Intel(R) Core(TM) i5-9600K CPU and a Windows 10 Education 64-bit operating system. The control rate of the virtual robot is set as 1,000 Hz, while the sampling rate for acquiring training/testing data is 100 Hz. For all the simulations in this section, the parameter assignments used in the impedance control law, the NDOB, and the fNN model are summarized in Table 3.1.

Table 3.1: Parameterization for simulations.

Description	Parameter	Location
Spring stiffness	$\mathbf{K}_m = 7.5\mathbf{I}$	Eqn. (3.3)
Damping	$\mathbf{D}_m = 2\sqrt{7.5}\mathbf{I}$	Eqn. (3.3)
Inertia matrix	$\hat{\mathbf{M}} = 1.0 \times 10^{-3} \times \mathbf{I}$	Eqn. (3.3),(3.7)
Observer gain	$\mathbf{Y} = 9.58 \times 10^{-2} \times \mathbf{I}$	Eqn. (3.7)
Coulomb friction	$\mathbf{F}_c = [0.0049, 0.0031, 0.001]$	Eqn. (3.5)
Static friction	$\mathbf{F}_s = [0.0035, 0.0028, 0.00165]$	Eqn. (3.5)
Viscous friction	$\mathbf{F}_v = [0.06, 0.048, 0.032]$	Eqn. (3.5)
Stribeck parameter	$\nu_s = [0.00038, 0.0003, 0.00024]$	Eqn. (3.5)
Neurons	12	Fig. 3.1
Input of fNN	$[x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3]'$	Fig. 3.1
Output of fNN	$\hat{\boldsymbol{\tau}}_{\text{dist}}$	Fig. 3.1
Training function	Bayesian regularization	Fig. 3.1
Transfer function	Symmetric sigmoid	Fig. 3.1
Data division	Random	Fig. 3.1
Data division ratio	$\text{trainRatio} = 0.8, \text{valRatio} = 0.2$	Fig. 3.1

Note: $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ denote identity matrix.

3.3.2 Comparing observers

The disturbance tracking performance among several typical observers, *i.e.*, NDOB [94], GMO [52], DKF [57, 80], and ESO [117], are qualitatively compared when a

disturbance of a controlled constant payload (22 gram) along y -axis is tracked. As shown in Figure 3.3, all of the observers can quickly and accurately estimate the controlled disturbance, although the ESO needs slightly longer time to track the disturbance while the other three observers have comparable disturbance tracking performance among each other.

Considering that NDOB has the advantage of capturing nonlinearities and is easy to implement, NDOB is selected for all simulations in the remaining part of this chapter. We also assume that the NDOB can accurately estimate all the lumped uncertainties (*i.e.*, $\tau_{\text{dist}} = \tau_{\text{NDOB}}$), and its outputs will be taken as the target values during the later training process.

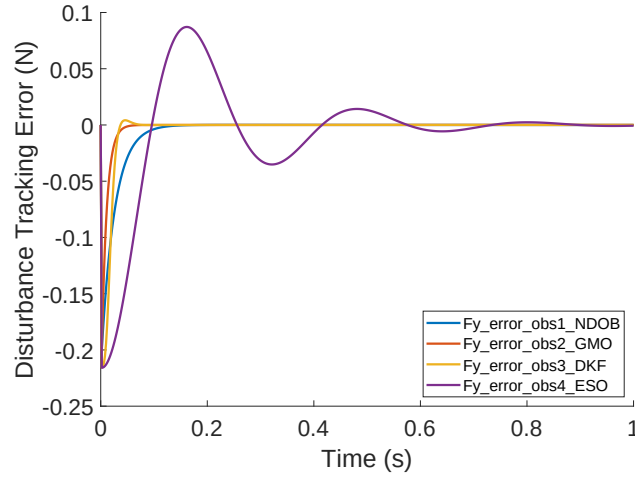


Figure 3.3: Comparison on disturbance tracking error of various disturbance observers. The disturbance is a constant payload of 22 gram along the y -axis.

3.3.3 Comparing NN models

The feed-forward neural network (fNN), cascade-forward neural network (cNN), and NARX model are compared on their estimation accuracy. As introduced earlier, the fNN and cNN have the same structure except that the cNN includes an additional connection from its input layer to each of the following layers. Considering that the learning target in our case is the dynamic model uncertainties rather than the whole

dynamic model, only one hidden layer is designed.

For fNN and cNN, as shown in Figure 3.1, the input is a 6-by-1 vector of Cartesian position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$ of the robot EE, and the hidden layer has 12 neurons with Bayesian regularization training function and symmetric sigmoid transfer function. The output layer has 3 neurons with a linear transfer function, and the output is a 3-by-1 vector.

The fNN, cNN, and NARX models are trained separately on data collected from 38 exciting trajectories with each trajectory lasting 20 seconds. The three NN models are trained to learn the uncertainties from source-1,2 by controlling only these two uncertainties sources to exist in the training data. When it comes to the testing process, the three NN models are expected to predict uncertainties only from source-1,2 although uncertainties from all four sources will exist in the testing data. Then, by subtracting the NN prediction output from the NDOB output, the estimation of external disturbances (sum of source-3,4) can be obtained.

The estimation accuracy of fNN, cNN, and NARX on the external disturbances are compared in Figure 3.4. As shown in the figure, there is no significant difference between the fNN and cNN models, but the NARX has a much worse performance. This is because the NARX requires time-series data for its training, and if training data is concatenated from several separate datasets then the training effect will be distorted. Since the fNN model has a common and more compact structure, it is selected for the subsequent simulations.

It is worth mentioning that the learning outcomes of fNN and cNN models are already relatively good when the training data involve 8 exciting trajectories. More exciting trajectories can further improve the learning effect to some extent but at a slow speed, which means that when higher prediction accuracy is not demanding, a relatively small set of training data may meet the requirements.

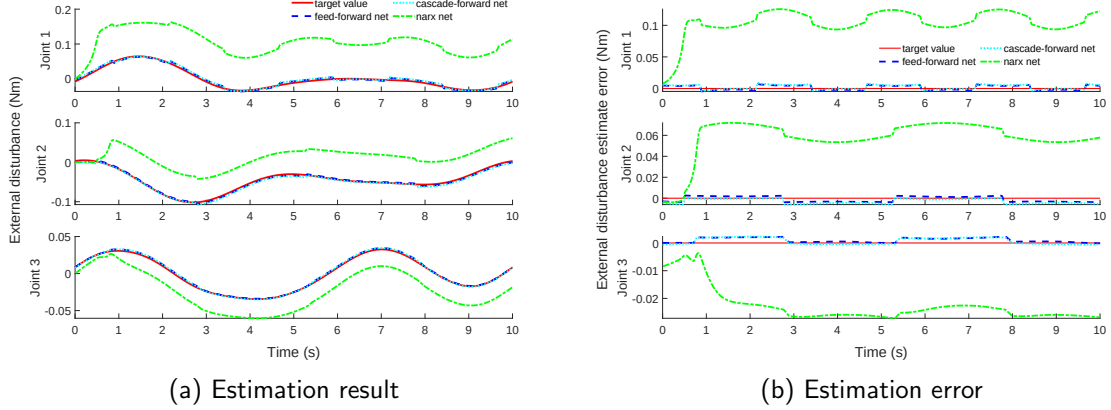


Figure 3.4: Compare the estimation performance of the feed-forward neural network (fNN), cascade-forward neural network (cNN), and NARX model on external disturbances.

3.3.4 The effect of dynamic model error

When building a dynamic model of a robot and identifying its dynamic parameters, dynamic model error is inevitable. The effect of dynamic model error (*i.e.*, uncertainties source-1) on the fNN prediction accuracy is explored. More specifically, different levels of model error (0%, 10%, 50%, 100%) are studied. The level of model error is tuned by value assignment on matrix $\hat{\mathbf{S}}$ and $\hat{\mathbf{g}}$ as introduced earlier.

The estimation performance of the fNN model on external disturbances under various levels of model error when tracking a simple figure-eight curve (3.9) and a complex exciting trajectory curve is shown in Figure 3.5a and Figure 3.6a, respectively. Correspondingly, the estimation errors are shown in Figure 3.5b and Figure 3.6b. From the figures, we can see that under all levels of model error except the level of 100%, there is no significant difference between each other. Even with the level of model error 100%, the estimation performance can be acceptable (see the dashed black line in Figure 3.5a and Figure 3.6a). In general, the effect of model error level on the fNN estimation accuracy is relatively small. The reason could be that for the lightweight robot used in this chapter, the dynamic model errors are on a relatively small scale. For other heavy industrial robots with dynamic model errors on a large scale, the effect of the model error level needs to be re-evaluated.

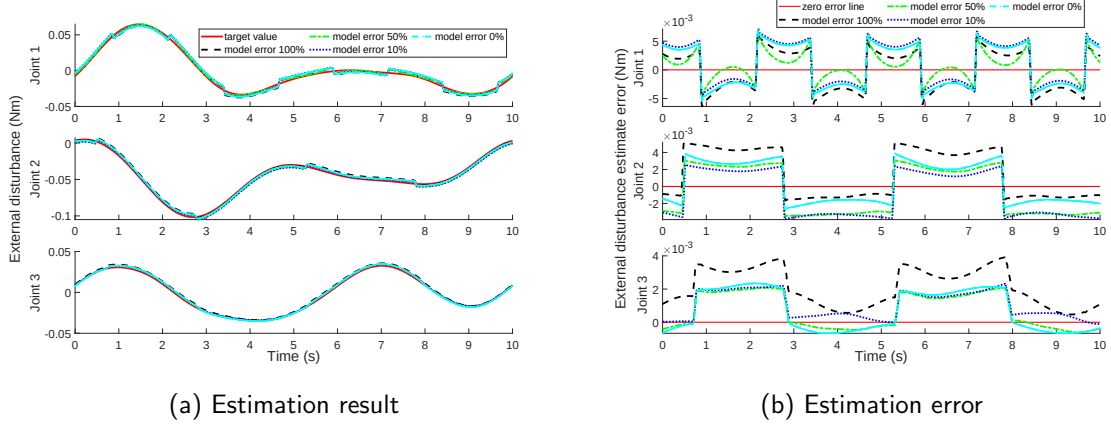


Figure 3.5: The effect of dynamic model error on fNN estimation performance when tracking a simple figure-eight trajectory.

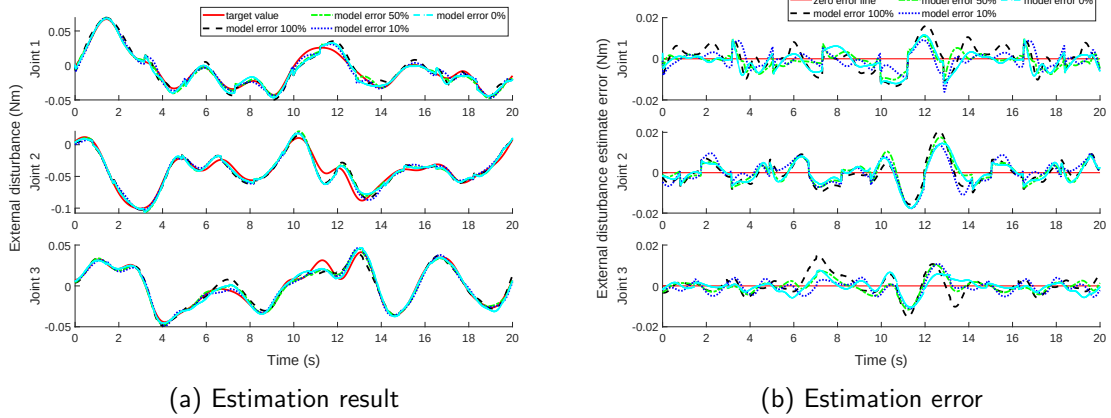


Figure 3.6: The effect of dynamic model error on fNN estimation performance when tracking a complex exciting trajectory.

3.3.5 Application on virtual fixture

During a trajectory tracking task, when the robot encounters external disturbances (*e.g.*, an obstacle in the surrounding environment), the NDOB will estimate and compensate for lumped uncertainties including both internal and external disturbances. This may escalate the adverse effect caused by external disturbances. On the other hand, the fNN model can be trained to estimate only internal disturbances, which can make the robot preserve the compliant behavior provided by the impedance controller when external disturbances occur.

In the above sections, the fNN model has been proved to be capable of predict-

ing the total internal disturbances (sum of source-1,2) thus estimating the external disturbances (sum of source-3,4). Here we further explored a potential application of the proposed framework of fNN learning on dynamic model uncertainties.

There are two conditions designed in a scenario of a robot encounters a stiff obstacle of virtual fixture (VF) on its way (*i.e.*, external disturbances) when executing a trajectory tracking task. Condition 1 is a scenario with implementing NDOB to estimate and compensate for the lumped uncertainties. Condition 2 is a scenario that is the same as Condition 2 except that the NDOB is replaced with an fNN model that was trained to learn the internal uncertainties at a level of model error 10%.

Figure 3.7 shows the trajectory tracking task performance in each of the two conditions. In the figure, the gray area represents the obstacle VF which will exert a contact force on the robot EE when they are in contact, while the green line represents the resulting disturbance torque in joint 2 which is obtained by projecting the Cartesian contact force into the robot joint space. Note that the disturbance torques in joint 1 and joint 3 are approximately zero due to the specific task setting and thus ignored.

By comparing Figure 3.7a and Figure 3.7b, we can see that the contact force in Figure 3.7a is extremely large due to the fact that the NDOB escalated the adverse effect of the obstacle VF. Note that if Condition 1 happens in a physical experiment, the robot and/or the obstacle will be badly destroyed once the robot starts to be in contact with the obstacle, thus Condition 1 should be avoided in physical experiments. In Condition 2 as shown in Figure 3.7b, the robot is much more compliant with a smaller contact force than that in Condition 1. This is because the fNN in Condition 2, as it was trained in the training process, only estimates the internal uncertainties excluding the external ones caused by the VF. This verified the effectiveness of the trained fNN model.

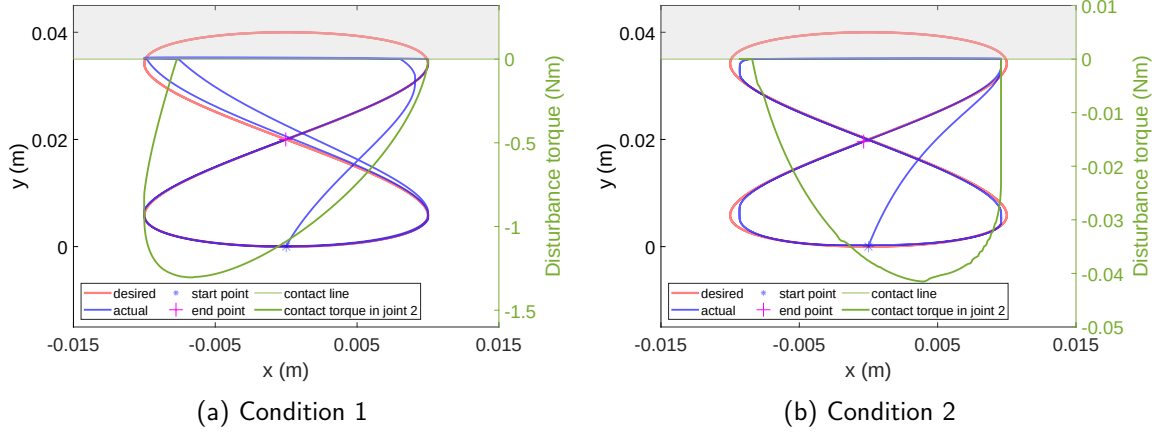


Figure 3.7: Simulation result of an application scenario on an obstacle of the virtual fixture. Condition 1, NDOB only; Condition 2, fNN only.

3.3.6 Limitations

Compared to prior work [57, 80], an advantage of the proposed learning framework is that, the fNN model is not restricted to learning only friction model or only model error which makes it more robust. In fact, it can learn all the uncertainties that exist in the training data, thus separating the later-coming uncertainties in the testing/application scenario. For example, if the fNN model is trained to learn uncertainties from source-1,2,3, then it is able to separate the uncertainties of source-4 when all four sources of uncertainties exist in the application scenario.

A limitation is that the trained fNN in this chapter works in trajectory tracking tasks but not set-point regulation tasks. This is because the fNN is trained by data collected from exciting trajectory tracking tasks. Therefore, the fNN model will be capable of working for both trajectory tracking tasks (*i.e.*, tasks with non-zero velocity) and setpoint regulation tasks (*i.e.*, tasks with zero velocity) if it is trained by data from both. This will be evaluated in future work.

3.4 Conclusion

In this chapter, a framework for learning robot dynamic model uncertainties and separating external disturbances by integrating impedance controller, NDOB observer, and feed-forward neural network (fNN) model was presented. By accurately controlling each of the four uncertainty sources in the simulations, the results show that the fNN can accurately learn the internal disturbances (source-1,2) thus separating the external disturbances (source-3,4). A further simulation on an application scenario, where an external obstacle of virtual fixture (VF) is involved, demonstrated the effectiveness of the presented learning framework. The fNN model in the framework is robust to learn all uncertainties that exist in the training process, then estimate all other newly added uncertainties in the testing process or application scenarios.

In the present work, the fNN model was only validated in scenarios of robots in free motion and obstructive motion. In future work, we will expand it to set-point regulation tasks which will enable human-robot interaction by separating the interaction force. Also, physical experiments will be conducted to evaluate the effectiveness of the proposed framework in real application scenarios.

Chapter 4

Iterative Learning for Gravity Compensation¹

To build a robot-assisted system, dynamic uncertainties can be a critical issue that could bring robot performance inaccuracy or even system instability if cannot be appropriately compensated. In **Chapter 2**, we developed a framework to integrate impedance control with a nonlinear disturbance observer (NDOB), and it can achieve an accurate impedance control when the robot is in free motion. However, the framework will refuse human-robot interaction since the human-applied force will also be regarded as a disturbance by the observer. In **Chapter 3**, we employed a neural network (NN) to learn the dynamic uncertainties thus filtering out certain types of uncertainties, and the NN works well in simulations after training. However, it requires complex procedures before the NN works, such as collecting training/testing data and training the model. Iterative learning for gravity compensation can be another promising way to solve this problem when gravity compensation is the main concern. In this chapter, a gravity iterative learning (Git) scheme in Cartesian space for gravity compensation, integrating with an impedance controller, is presented. A steady-state scaling strategy is then proposed which released the updating requirements of the learning scheme and also extended its validity to trajectory-tracking

¹A version of this chapter has been published as: Teng Li, Amir Zakerimanesh, Yafei Ou, Armin Badre, and Mahdi Tavakoli, “Iterative Learning for Gravity Compensation in Impedance Control”, IEEE/ASME Transactions on Mechatronics, 2024.

scenarios from set-point regulations. The deriving process and convergence properties of the Git scheme are presented and theoretically analyzed, respectively. A series of simulations and physical experiments are conducted to evaluate the validity of the scaling strategy, the learning accuracy of the Git scheme, and the effectiveness of the learning-based impedance controller. Both simulation and experimental results demonstrate good performance and properties of the Git scheme and the learning-based impedance controller.

4.1 Background

Robot-assisted minimally invasive surgery (MIS) has been becoming increasingly popular across various surgical specialties, such as orthopedics.[43]. MIS can bring the benefits of a faster recovery rate and decreased pain to patients thus getting more favor. Robot-assisted surgeries are transforming traditional orthopedic surgeries by helping surgeons achieve more successful and precise surgical outcomes with the assistance of robots [140, 60, 89]. Elbow arthroscopy is a typical type of MIS in orthopedics that allows the management of elbow stiffness, arthritis, and fractures in a minimally invasive fashion [13]. During traditional elbow arthroscopy, the surgeon needs to hold an arthroscope with one hand while performing the surgical operations with the other hand, which can restrict the dexterity of the surgical performance and increase the cognitive load. This arouses the necessity to develop a robot-assisted arthroscope holder where the robot can hold the arthroscope for the surgeon during the surgery.

To build a robot-assisted system for assisting surgeons in holding with arthroscope during orthopedic surgery, some requirements need to be satisfied [75]. First, The robot can hold the arthroscope still at a specified pose (*i.e.*, setpoint regulation) while rejecting all possible disturbances (*e.g.*, external disturbances delivered to the arthroscope via contact with the patient’s body during surgery). Second, when the surgeon needs to move the arthroscope to a new pose (*e.g.*, for adjusting the scope view perspective), the robot should allow the surgeon to move it around freely (*i.e.*,

human-robot interaction). Then, when a new pose is determined by the surgeon, the robot should keep the arthroscope still again while rejecting any disturbances. The main problem to build such a robot-assisted system is the dynamic model uncertainties and external disturbances, which could largely affect the robot’s task performance accuracy and even stability if they are not appropriately compensated. More specifically, incomplete gravity compensation can be the main issue in this case since heavy surgical tools with unknown weights will be attached to the robot end-effector (EE).

Disturbance observer is a promising way to estimate and compensate for dynamic uncertainties. In our previous work [75], we have shown that by integrating impedance control and nonlinear disturbance observer (NDOB), an accurate impedance control can be achieved. In that work, the disturbance observer can accurately estimate and compensate for the lumped uncertainties including incomplete gravity compensation. However, the nonlinear disturbance observer as well as other types of observers [78], such as generalized momentum observer (GMO) [52], joint velocity observer (JVOB) [52], extended state observer (ESO) [117], and disturbance Kalman filter (DKF) method [57, 80], always estimate a lumped uncertainty term and is not able to separate out any one component when several uncertainty sources exist. Moreover, the observer will refuse human-robot interaction since human-applied force will be taken as a part of the lumped disturbances thus being rejected.

Learning control has been developed to track repetitive trajectories for both rigid and non-rigid robots. De Luca and Ulivi presented a simple and efficient iterative learning algorithm for robots with joint elasticity [29]. In their work, a learning term was used to learn the necessary modification to the desired trajectory position. They demonstrated the algorithm’s usefulness by good motion performance of simulations on a two-link planar robot. Based on a similar design methodology, an iterative learning scheme for gravity compensation in setpoint regulation problems was initially proposed by De Luca and Panzieri [26, 84]. The learning scheme completes the required gravity compensation at the final steady state in setpoint regulation tasks.

It can iteratively learn the constant gravity without the need of introducing an integral error term or using high-gain feedback.

Based on the same contraction mapping theorem, Basovich *et al.* developed an iterative output feedback controller for a 6-degree-of-freedom (DOF) precision positioning system when only position measurement is available [10]. Their proposed controller can learn and compensate for the payload uncertainties with bounded error in setpoint control tasks. Ji *et al.* used the iterative learning method to auto-calibrate gravity compensation when the robot has no contact with the environment thus making the robot EE weightless [61].

Incomplete or absent gravity compensation will cause a constant steady-state error [26]. For impedance control, making the robot "stiffer" by tuning up the impedance gains can reduce the error to some extent, but not eliminate it. Especially when unknown heavy external payloads are attached to the robot EE, the method of tuning up impedance gains will be largely limited and be difficult to achieve satisfying results.

In summary, in our target application scenario, *i.e.*, robot-assisted arthroscopic surgery, gravity compensation and physical human-robot interaction (*pHRI*) are the main concerns. There are various disturbance observers available for gravity compensation [78], *e.g.*, NDOB, GMO, etc. However, the output of an observer is a lumped estimate on all uncertainties including gravity, and it will refuse human-robot interaction by taking it as a part of uncertainties [75]. Furthermore, it also requires the estimated dynamic parameters of the robot dynamics [78]. An adaptive controller [130, 59] can also deal with dynamic uncertainties including gravity. However, it is a controller rather than an independent approach for disturbance estimation, and it cannot provide compliant robot behavior for a safe human-robot interaction like an impedance controller can do. Therefore, a simple method that can focus on gravity compensation while enabling *pHRI* and avoiding the necessity of the robot dynamics is needed in our scenario.

Inspired by [26], in this chapter, we proposed a gravity iterative learning (Git)

scheme for gravity compensation in Cartesian space and integrated it with an impedance controller. The convergence properties of the Git scheme are theoretically analyzed. The learning performance and effectiveness are then evaluated by a series of simulations and experiments in both trajectory tracking tasks and setpoint regulation tasks. Finally, an application experiment in *p*HRI scenario is presented to show the effectiveness of the integrated controller. The main contributions in this work can be described as the following,

- An adapted iterative learning scheme for gravity compensation in Cartesian space is presented, and the converging properties are theoretically analyzed.
- A steady-state scaling strategy is proposed, which enables the iterative learning update law to be executed in each servo loop, and more importantly, it extends the validity of the learning scheme to general trajectory-tracking scenarios.

4.2 Methods

4.2.1 Robot dynamics and disturbances

A general dynamic model for an n -degree-of-freedom (DOF) rigid robot with revolute joints [41] can be given by

$$\underbrace{\mathbf{M}(\mathbf{q})}_{\hat{\mathbf{M}}+\Delta\mathbf{M}}\ddot{\mathbf{q}} + \underbrace{\mathbf{S}(\mathbf{q},\dot{\mathbf{q}})}_{\hat{\mathbf{S}}+\Delta\mathbf{S}}\dot{\mathbf{q}} + \underbrace{\mathbf{G}(\mathbf{q})}_{\hat{\mathbf{G}}+\Delta\mathbf{G}} + \boldsymbol{\tau}_{\text{fric}}(\dot{\mathbf{q}}) = \boldsymbol{\tau} + \underbrace{\boldsymbol{\tau}_{\text{ext}}}_{\mathbf{J}^T\mathbf{F}_{\text{ext}}} \quad (4.1)$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ are the joint position, velocity, and acceleration, respectively, $\mathbf{M} \in \mathbb{R}^{n \times n}$ denotes the inherent inertia matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes a matrix of the Coriolis and centrifugal forces, $\mathbf{G} \in \mathbb{R}^n$ represents the gravity vector. $\hat{\mathbf{M}}, \hat{\mathbf{S}}, \hat{\mathbf{G}}$ represent users' model estimates, while $\Delta\mathbf{M}, \Delta\mathbf{S}, \Delta\mathbf{G}$ are the corresponding estimate errors. $\boldsymbol{\tau}_{\text{fric}} \in \mathbb{R}^n$ is joint friction, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the commanded joint torque vector, $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^n$ is the torque caused by external force, $\mathbf{F}_{\text{ext}} \in \mathbb{R}^6$ is the external force in Cartesian space, and $\mathbf{J} \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix.

By collecting all the disturbances together, the dynamic model (4.1) of a robot can be re-written as

$$\hat{\mathbf{M}}\ddot{\mathbf{q}} + \hat{\mathbf{S}}\dot{\mathbf{q}} + \hat{\mathbf{G}} = \boldsymbol{\tau} + \underbrace{\boldsymbol{\tau}_{\text{ext}} - [\boldsymbol{\tau}_{\text{fric}} + (\Delta\mathbf{M}\ddot{\mathbf{q}} + \Delta\mathbf{S}\dot{\mathbf{q}} + \Delta\mathbf{G})]}_{\boldsymbol{\tau}_{\text{dist}}} \quad (4.2)$$

where $\boldsymbol{\tau}_{\text{dist}}$ denotes the lumped uncertainties containing the model error $(\Delta\mathbf{M}\ddot{\mathbf{q}} + \Delta\mathbf{S}\dot{\mathbf{q}} + \Delta\mathbf{G})$, the joint friction $\boldsymbol{\tau}_{\text{fric}}$, and the external disturbances $\boldsymbol{\tau}_{\text{ext}}$.

In this chapter, we will focus on estimating and compensating for the gravity caused by external constant payloads using an iterative learning method. In order to clearly reveal the behavior of the iterative learning algorithm to learn the gravity of the external payloads, in the simulations we assume that, (a) an ideal dynamic model is available, *i.e.*, $\hat{\mathbf{M}} = \mathbf{M}$, $\hat{\mathbf{S}} = \mathbf{S}$, $\hat{\mathbf{G}} = \mathbf{G}$, thus $\Delta\mathbf{M} = \mathbf{0}$, $\Delta\mathbf{S} = \mathbf{0}$, $\Delta\mathbf{G} = \mathbf{0}$; (b) no joint friction, *i.e.*, $\boldsymbol{\tau}_{\text{fric}} = \mathbf{0}$; (c) only constant payloads exists for external disturbances. By applying these assumptions, the dynamic model (4.2) will become (4.3).

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{S}\dot{\mathbf{q}} + \mathbf{G} = \boldsymbol{\tau} + \underbrace{\mathbf{J}^T \mathbf{F}_{\text{ext}}}_{\boldsymbol{\tau}_{\text{dist}}} \quad (4.3)$$

The model (4.3) can be expressed in Cartesian space as

$$\mathbf{M}_x \ddot{\mathbf{x}} + \mathbf{S}_x \dot{\mathbf{x}} + \mathbf{G}_x = \mathbf{J}^{-T} \boldsymbol{\tau} + \mathbf{F}_{\text{ext}} \quad (4.4)$$

where $\mathbf{M}_x, \mathbf{S}_x, \mathbf{G}_x$ have

$$\begin{cases} \mathbf{M}_x = \mathbf{J}^{-T} \mathbf{M} \mathbf{J}^{-1} \\ \mathbf{S}_x = \mathbf{J}^{-T} \mathbf{S} \mathbf{J}^{-1} - \mathbf{M}_x \dot{\mathbf{J}} \mathbf{J}^{-1} \\ \mathbf{G}_x = \mathbf{J}^{-T} \mathbf{G} \end{cases} \quad (4.5)$$

where $\mathbf{M}_x, \mathbf{S}_x, \mathbf{G}_x$ are the $\mathbf{M}, \mathbf{S}, \mathbf{G}$ expressed in Cartesian space, respectively.

4.2.2 Impedance control

A desired impedance model [129, 133, 75] for robot-environment interaction can be expressed as

$$\mathbf{F}_{\text{imp}} = \mathbf{M}_m(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d) + (\mathbf{S}_x + \mathbf{D}_m)(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}_m(\mathbf{x} - \mathbf{x}_d) \quad (4.6)$$

where $\mathbf{M}_m, \mathbf{D}_m, \mathbf{K}_m$ are user-designed matrices for inertia, damping, and stiffness, respectively. Note that $\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d$ are the desired position, velocity, and acceleration, respectively in Cartesian space, while $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$ are the actual ones. $\mathbf{F}_{\text{imp}}$ is the interaction force between the robot and the environment.

To avoid the measurement of external forces, the designed inertia matrix will be set as the inherent inertia matrix of the robot, *i.e.*, $\mathbf{M}_m = \mathbf{M}_x$. Then, by substituting (4.6) into (4.4) with $\mathbf{F}_{\text{ext}} = \mathbf{F}_{\text{imp}}$, the impedance control law can be given by [75]

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{J}^{-1}(\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{x}}_d) + \mathbf{S}\mathbf{J}^{-1}\dot{\mathbf{x}}_d + \mathbf{G} + \mathbf{J}^T[\mathbf{D}_m(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_m(\mathbf{x}_d - \mathbf{x})] \quad (4.7)$$

Note that when implementing the impedance controller (4.7) in practice for physical experiments, the estimates $\hat{\mathbf{M}}, \hat{\mathbf{S}}, \hat{\mathbf{G}}$ will be used for the calculation since an accurate model of a physical robot is usually not available.

For moving robot EE to a fixed point, *i.e.*, set-point regulation, we have $\dot{\mathbf{x}}_d = \mathbf{0}$, $\ddot{\mathbf{x}}_d = \mathbf{0}$. Then, the impedance control law (4.7) can be simplified and reduced to (4.8), which is also known as task-space proportional-derivative (PD) controller with gravity compensation.

$$\boldsymbol{\tau} = \mathbf{J}^T[\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \mathbf{G} \quad (4.8)$$

4.2.3 Iterative learning for gravity compensation

One straightforward way to reduce the effect of dynamic uncertainties (including incomplete or absent gravity compensation/cancellation) is to make the robot stiffer by tuning up the spring gains ($\mathbf{K}_m$) in the impedance model. This could be feasible in simulations where the gains can be set to be very large, but not feasible in practice where the robot may have chattering due to large gains. Especially when heavy external payloads are involved, solely tuning the impedance gains may not be able to obtain a satisfactory result. To solve this problem, we introduce an iterative learning scheme for gravity compensation in Cartesian space.

Inspired by [26] where iterative learning was integrated with a PD controller in joint space, a Cartesian-space impedance control law (at the i -th iteration, $i = 1, 2, \dots$) integrating with a gravity iterative learning (Git) scheme for gravity compensation is proposed which can be expressed by

$$\boldsymbol{\tau}_i = \mathbf{M}\mathbf{J}^{-1}(\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{x}}_d) + \mathbf{S}\mathbf{J}^{-1}\dot{\mathbf{x}}_d + \mathbf{J}^T[\mathbf{D}_m(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x})] + \mathbf{J}^T\mathbf{u}_{i-1} \quad (4.9)$$

where $\mathbf{J}^T\mathbf{u}_{i-1}$ is an iterative learning term for gravity compensation instead of a gravity term $\mathbf{G}$. For setpoint regulation, it will be reduced to be

$$\boldsymbol{\tau}_i = \mathbf{J}^T[\gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \mathbf{J}^T\mathbf{u}_{i-1} \quad (4.10)$$

The update law for the iterative learning $\mathbf{u}_i$ can be given by

$$\mathbf{u}_i = \gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) + \mathbf{u}_{i-1} \quad (4.11)$$

where γ is a positive scalar gain, and setting $\mathbf{u}_0 = \mathbf{0}$ for initialization. Also, different from [26] where one iteration was set as 3 seconds while in this chapter it updates itself in each sampling loop. This ensures the updated values of the iteration term are changing continuously and smoothly from one iteration to the next, and also extends its validity to more general tracking tasks from setpoint regulation. Theoretical analysis will be introduced in detail later.

To avoid a sudden impulse at the moment of the robot starts up due to a potentially large initial error between the initial actual position and the initial desired position, a simple linear interpolating strategy is used which is given by

$$\begin{cases} \mathbf{x}_d = \mathbf{x}_0 + (\mathbf{x}_d - \mathbf{x}_0)\frac{t}{t_1} & \text{if } t \leq t_1 \\ \dot{\mathbf{x}}_d = \dot{\mathbf{x}}_0 + (\dot{\mathbf{x}}_d - \dot{\mathbf{x}}_0)\frac{t}{t_1} & \text{if } t \leq t_1 \\ \ddot{\mathbf{x}}_d = \ddot{\mathbf{x}}_0 + (\ddot{\mathbf{x}}_d - \ddot{\mathbf{x}}_0)\frac{t}{t_1} & \text{if } t \leq t_1 \\ \mathbf{x}_d = \mathbf{x}_d & \text{if } t > t_1 \\ \dot{\mathbf{x}}_d = \dot{\mathbf{x}}_d & \text{if } t > t_1 \\ \ddot{\mathbf{x}}_d = \ddot{\mathbf{x}}_d & \text{if } t > t_1 \end{cases} \quad (4.12)$$

where $\mathbf{x}_0 = \mathbf{constant}$, $\dot{\mathbf{x}}_0 = \mathbf{0}$, $\ddot{\mathbf{x}}_0 = \mathbf{0}$ are the initial actual position, velocity, and acceleration, t_1 is the duration of the transition period defined by the user (in this

chapter, $t_1 = 2$ seconds). Note that the $\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d$ on the right-hand side of the equations represent the theoretical values from the predefined trajectory or setpoint, while those on the left-hand side represent the values used for calculation in the controller. As shown in (4.12), the desired position, velocity, and acceleration are set up increasingly from the initial actual ones ($\mathbf{x}_0, \dot{\mathbf{x}}_0, \ddot{\mathbf{x}}_0$) (at $t = 0$) to the desired ones ($\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d$) (at $t = t_1$) within the very first t_1 seconds. In other words, Eqn. (4.12) ensures errors increase linearly from zeros (at $t = 0$) to the actual errors (at $t = t_1$) when the robot starts up. It should be noted that this smoothing strategy is independent of the control laws and only valid within the first t_1 seconds. The block diagram of the proposed iterative learning for gravity compensation in impedance control is illustrated in Figure 4.1.

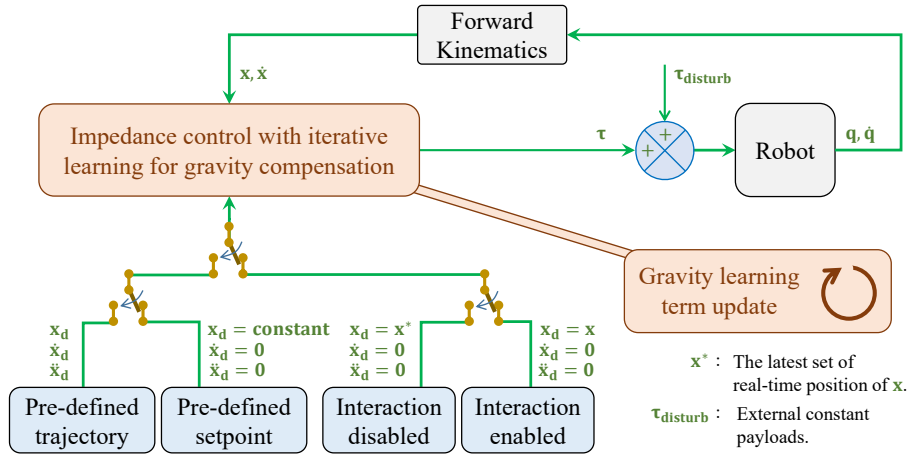


Figure 4.1: Control block diagram of an impedance controller with iterative learning scheme for gravity compensation. When $\mathbf{x}_d = \mathbf{x}$, the position-dependent terms in the impedance controller and the Git update law vanish, meaning that the setpoint regulation is released and interaction is enabled, and now the user can move the robot EE around. When $\mathbf{x}_d = \mathbf{x}^*$, a setpoint regulation task is recovered and interaction is disabled. The latest set of position ($\mathbf{x}^*$) ensures seamless switching between the “interaction enabled” mode and the “interaction disabled” mode, which can be easily realized by a pedal switch.

4.2.4 Analysis

In this section, the process of designing the iterative learning term and the corresponding update law will be presented in detail. Then, the convergence properties of

the iterative learning scheme will be theoretically analyzed in a scenario of setpoint regulation.

At the steady state ($\mathbf{q} = \mathbf{constant}$, $\dot{\mathbf{q}} = \ddot{\mathbf{q}} = \mathbf{0}$) of the i -th iteration in the scenario of setpoint regulation, it has

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{S}\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}_i \\ \boldsymbol{\tau}_i = \mathbf{J}^T[\gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \mathbf{J}^T\mathbf{u}_{i-1} \end{cases} \quad (4.13)$$

Combining the two equations in (4.13) as one equation, yields,

$$\mathbf{G}(\mathbf{q}_i) = \mathbf{J}^T\gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) + \mathbf{J}^T\mathbf{u}_{i-1} \quad (4.14)$$

Based on (4.14), the update law of the iterative learning term can be designed as

$$\mathbf{J}^T\mathbf{u}_i = \mathbf{J}^T\gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) + \mathbf{J}^T\mathbf{u}_{i-1} \quad (4.15)$$

Simplify (4.15), we obtain the update law given by (4.11). By designing the update law in this way and by comparing (4.14) with (4.15), we are actually assuming that at the steady state, the learning term converged to the gravity term, *i.e.*,

$$\mathbf{G}(\mathbf{q}_i) = \mathbf{J}^T\mathbf{u}_i \quad (4.16)$$

The subsequent part will analyze and show proof of the convergence capability of the designed iterative learning scheme. The update law (4.11) can be rewritten as

$$\mathbf{u}_i - \mathbf{u}_{i-1} = \gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}_i) \quad (4.17)$$

Define the position error in Cartesian space as $\mathbf{e}_i = \mathbf{x}_d - \mathbf{x}_i$, Eqn. (4.17) can be rewritten as

$$\mathbf{u}_i - \mathbf{u}_{i-1} = \gamma\mathbf{K}_m\mathbf{e}_i \quad (4.18)$$

Also, the position error in Cartesian space between two adjacent iteration steps can be expressed as

$$\mathbf{x}_i - \mathbf{x}_{i-1} = \mathbf{x}_i - \mathbf{x}_d + \mathbf{x}_d - \mathbf{x}_{i-1} = -\mathbf{e}_i + \mathbf{e}_{i-1} \quad (4.19)$$

Knowing that the derivative of gravity is bounded [26] by

$$\left\| \frac{\partial \mathbf{G}(\mathbf{q})}{\partial \mathbf{q}} \right\| \leq \alpha \quad (4.20)$$

where α is a positive constant. Rewrite (4.20) in the form of finite difference as

$$\left\| \frac{\mathbf{G}(\mathbf{q}_i) - \mathbf{G}(\mathbf{q}_{i-1})}{\mathbf{q}_i - \mathbf{q}_{i-1}} \right\| \leq \alpha \quad (4.21)$$

Also, the relationship between Cartesian velocity and joint velocity is given by

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \quad (4.22)$$

Assuming that the Jacobian matrix is invertible, *i.e.*, $\mathbf{J}^{-1}$ exists. Rewrite (4.22) in the form of finite difference as

$$\mathbf{q}_i - \mathbf{q}_{i-1} = \mathbf{J}^{-1}(\mathbf{x}_i - \mathbf{x}_{i-1}) \quad (4.23)$$

From the relationship between gravity term and iterative learning term at steady state (4.16), it yields

$$\begin{aligned} \|\mathbf{u}_i - \mathbf{u}_{i-1}\| &= \|\mathbf{J}^{-\mathbf{T}}\mathbf{G}(\mathbf{q}_i) - \mathbf{J}^{-\mathbf{T}}\mathbf{G}(\mathbf{q}_{i-1})\| \\ &\leq \|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{G}(\mathbf{q}_i) - \mathbf{G}(\mathbf{q}_{i-1})\| \\ &\leq \alpha \|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{q}_i - \mathbf{q}_{i-1}\| \quad (\text{by (4.21)}) \\ &\leq \alpha \|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{J}^{-1}(\mathbf{x}_i - \mathbf{x}_{i-1})\| \quad (\text{by (4.23)}) \\ &\leq \alpha \|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{J}^{-1}(-\mathbf{e}_i + \mathbf{e}_{i-1})\| \quad (\text{by (4.19)}) \\ &\leq \alpha \|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{J}^{-1}\| (\|\mathbf{e}_i\| + \|\mathbf{e}_{i-1}\|) \end{aligned} \quad (4.24)$$

Assuming that the minimum eigenvalue of the user-defined matrix $\mathbf{K}_m$ meets the condition of $\lambda_{\min}(\mathbf{K}_m) > \alpha$, then it can yield the following inequality property

$$\gamma\alpha\|\mathbf{e}_i\| < \gamma\lambda_{\min}(\mathbf{K}_m)\|\mathbf{e}_i\| \leq \|\gamma\mathbf{K}_m\mathbf{e}_i\| \quad (4.25)$$

By combining (4.18) (4.24) (4.25), yields

$$\gamma\alpha\|\mathbf{e}_i\| < \|\gamma\mathbf{K}_m\mathbf{e}_i\| \leq \alpha\|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{J}^{-1}\| (\|\mathbf{e}_i\| + \|\mathbf{e}_{i-1}\|) \quad (4.26)$$

By simplifying (4.26), yields

$$\gamma \|\mathbf{e}_i\| < \|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{J}^{-1}\| (\|\mathbf{e}_i\| + \|\mathbf{e}_{i-1}\|) \quad (4.27)$$

Reorganizing (4.27), yields

$$\|\mathbf{e}_i\| < \frac{\beta}{\gamma - \beta} \|\mathbf{e}_{i-1}\| \quad (4.28)$$

where $\beta = \|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{J}^{-1}\|$. In order for contraction mapping, requires

$$\frac{\beta}{\gamma - \beta} \leq 1 \quad (4.29)$$

Due to $\beta > 0$ is always true, yields

$$\begin{aligned} \gamma &\geq 2\beta \\ \gamma &\geq 2\|\mathbf{J}^{-\mathbf{T}}\| \|\mathbf{J}^{-1}\| \geq 2\|\mathbf{J}^{-\mathbf{T}}\mathbf{J}^{-1}\| \end{aligned} \quad (4.30)$$

Assuming that the Jacobian matrix $\mathbf{J}$ is bounded, then $\mathbf{J}^{-\mathbf{T}}$ and $\mathbf{J}^{-1}$ are both bounded. Then, set the following boundness

$$b \geq \|\mathbf{J}^{-\mathbf{T}}\mathbf{J}^{-1}\| \quad (4.31)$$

Finally, it can conclude that, on the conditions of (1) Jacobian matrix is invertible and bounded, and (2) $\lambda_{\min}(\mathbf{K}_m) > \alpha$, then, $\gamma \geq 2b$ can ensure the iterative learning term ($\mathbf{J}^{\mathbf{T}}\mathbf{u}_{i-1}$ in (4.10)) being a contraction mapping, in other words, can ensure the iterative learning term converges to the true gravity at the steady state. Note that the convergence condition here is only sufficient, which means that even if it is violated the iterative learning term may still converge. This is consistent with the conclusion made in [26].

4.2.5 Steady-state scaling strategy

For the iterative learning-based update law (4.11), to explicitly display an learning rate η (by default $\eta = 1$), the update law (4.11) can be rewritten as

$$\mathbf{u}_i = \gamma\eta\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) + \mathbf{u}_{i-1} \quad (4.32)$$

It is worth noting that an important assumption has been made for the contraction mapping is "at steady state", and the update law (4.32) should be executed at steady state theoretically. This is consistent with the drawback described in the prior work [26, 29, 84], where the steady state is set as 3 seconds in their simulations. Executing the update law only at steady state (*e.g.*, every 3 seconds in [26]) is ok for simple simulations, but in practice, it would be a significant limitation.

As a further step in this chapter, analog to the concept of finite difference, we scale the common "steady-state" period (taking 1 second here as an example) down to the level of sampling time (0.001 seconds) such that the update law can be executed in each sampling loop. Since the default learning rate ($\eta = 1$) in (4.32) is corresponding to the common steady-state period (1 second), it also needs to be scaled down to be $\eta = 0.001$ in order to match with the scaled steady-state period (0.001 seconds). This enables the easy implementation of the update law (4.32) and allows it to be updated in each sampling loop.

More importantly, the steady-state scaling strategy enables the iterative learning scheme to be valid also for more general trajectory-tracking scenarios. Since learning-based impedance control law (4.10) (for setpoint regulation scenario) is reduced from (4.9) (for the general trajectory-tracking scenario), impedance control law (4.9) and iterative learning update law (4.32) can be used for iterative learning on the gravity compensation in trajectory-tracking scenarios. This strategy will be evaluated with simulations and experiments in the next section.

4.3 Simulations, Experiments, and Results

4.3.1 Robotic system

A 3-DOF PHANToM Premium 1.5A robot (3D Systems, Inc., Cary, NC, USA) is used for simulations and experiments in this chapter. For the simulations, we reconstruct the kinematic model and dynamic model of the PHANToM robot based on [16]

and conduct the simulations using MATLAB/Simulink (version R2020a, MathWorks Inc., Natick, MA, USA). For the experiments, the physical robot is controlled via joint torque command, which is sent from MATLAB/Simulink using Quarc real-time control software (Quanser Inc., Markham, ON, Canada). The control rate of the robot is 1,000 Hz. The MATLAB/Simulink and Quarc software run on a computer with a 3.33 GHz Intel(R) Core(TM) 2 i5 CPU with a Windows 7 Enterprise 64-bit operating system.

4.3.2 Parameterization

Table 4.1: Parameterization for simulations and experiments.

Parameters	Simulations	Experiments
Spring	$\mathbf{K}_m = 38.44\mathbf{I}$	$\mathbf{K}_m = 7.29\mathbf{I}$
Damper	$\mathbf{D}_m = 12.40\mathbf{I}$	$\mathbf{D}_m = 5.40\mathbf{I}$
Spring (increased stiffness)	NA	$\mathbf{K}_m = 200\mathbf{I}$
Damper (increased stiffness)	NA	$\mathbf{D}_m = 2\mathbf{I}$
Learning gain γ	1	1
Learning rate η	1/0.001/0.005/0.025	0.001

Note: $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ denote identity matrix. NA, not applied. The parameters are determined via trial and error with a binary search strategy.

For all simulations and experiments in the remaining part of this paper, the parameter values used in the learning-based impedance controller (4.9) and the iterative learning update law (4.32) are listed in Table 4.1. In order to involve acute changes in position and velocity, a concaved-square trajectory is selected for the simulations and experiments, which can be expressed as a function of time given by

$$\begin{cases} x_d = R \cos^3(t) \\ y_d = R \sin^3(t) + R \\ z_d = 0 \end{cases} \quad (4.33)$$

where $R = 0.02$ m is a parameter of the concaved-square. Note that the described trajectory is in a vertical plane in the workspace of the 3DOF robot.

In the following sub-sections, a series of simulations and experiments are conducted to evaluate the effectiveness of the proposed iterative learning scheme both in free-motion mode (*i.e.*, trajectory tracking tasks) and in restricted-motion mode (*i.e.*, setpoint regulation tasks). A [video](#) demonstration for the experiments is available online¹.

It should be noted that the gravity compensation estimated by the iterative learning scheme may include two main components, *i.e.*, gravity term in the dynamic model, and (if applicable) all constant payloads attached to the robot EE or robot body. In order to clearly reveal the converging properties and learning performance of the learning scheme, in the simulations of this work, we will assume the gravity term is fully known in such a way the iterative learning term only learns and compensates for the unknown external payloads.

4.3.3 Simulation to evaluate steady-state scaling strategy

In the original work [26] where the iterative learning scheme was initially proposed with a PD controller in joint space, a significant drawback of the scheme is that the iterative update should be executed at steady-state. This is also true for this work since the same "steady-state" assumption has been used during the theoretical analysis of the convergence properties. This drawback can largely limit the learning scheme to be implemented in practice.

To overcome this drawback, at the end of Section 4.2, we proposed a strategy to scale down the steady-state period (equivalent to iterative update time for updating the update law (4.32)) to the same level as the sampling time in order to improve and generalize the iterative learning scheme. The steady-state scaling strategy requires the learning rate (η) to be scaled to the same level accordingly. In this section, we will evaluate this strategy with simulations.

In Figure 4.2, a comparison of with-scaling and without-scaling the iterative update

¹online demo video link: <https://youtu.be/KH8RxaaRIA4>

time is presented when the robot is in a setpoint regulation task. For the without-scaling ($\eta = 1$) scenario, the gravity learning behavior under various conditions of iterative update time ($T_{it} = 0.5, 1, 2, 3\text{s}$) is investigated as shown by the green dash-dot lines in the figure. Correspondingly, the blue dot lines represent the gravity learning behavior in the with-scaling scenario, where the iterative update time is scaled ($T_{it} = 0.001\text{s}$) to be the same as the sampling time while the learning rate η is scaled accordingly ($\eta = 0.002, 0.001, 0.001/2, 0.001/3\text{s}$). The comparison between the green dash-dot lines and the blue dot lines in Figure 4.2 revealed that the steady-state scaling strategy is effective and reasonably sound. Especially in Figure 4.2a, the similarity between the with-scaling scenario and the without-scaling scenario is clearly revealed.

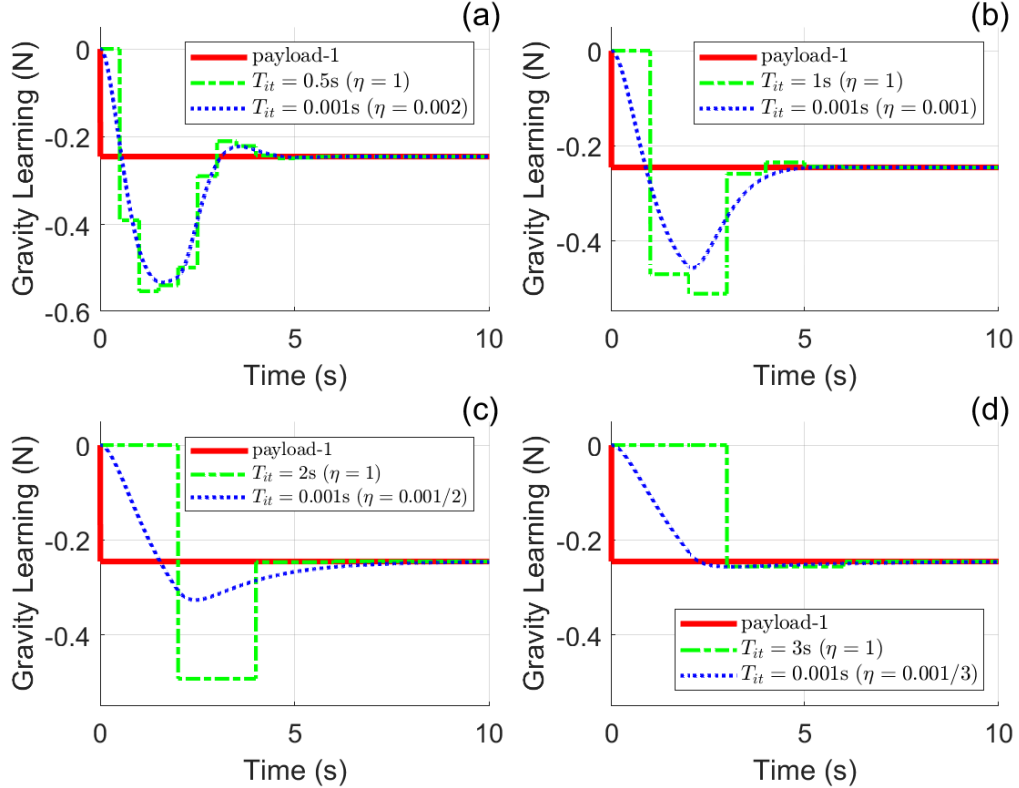


Figure 4.2: Simulation results of scaling down the iterative update time (T_{it} , *i.e.*, the steady-state time) while scaling down the iterative learning rate (η) accordingly. (a) Scaling down T_{it} from 0.5s to 0.001s. (b) Scaling down T_{it} from 1s to 0.001s. (c) Scaling down T_{it} from 2s to 0.001s. (d) Scaling down T_{it} from 3s to 0.001s. The reference is an external payload-1 (25g). The setpoint is set as $[0.01, 0.04, 0]$ m in Cartesian space.

4.3.4 Simulation on setpoint regulation task

Simulations in four cases are conducted in setpoint regulation tasks (restricted-motion mode). Different cases are related to different external payloads attached to the robot EE which can be described below

- Case #0, reference, no external payloads.
- Case #1, payload-1 (25 gram) attached.
- Case #2, payload-2 (125 gram) attached.
- Case #3, payload-3 (250 gram) attached.

A 3DOF robot is implemented with the iterative learning-based control law (4.10) and learning update law (4.32) as well as the steady-state scaling strategy, where the iterative learning scheme is used to iteratively learn and compensate the gravity of the external payloads in different cases. Figure 4.3 shows the simulation results in a setpoint regulation task with a pure impedance controller (4.8) (or equivalently controller (4.10) with setting the learning rate $\eta = 0$) under an ideal dynamic model (*i.e.*, the dynamic model matrices $\mathbf{M}$, $\mathbf{S}$, $\mathbf{G}$ are fully known). As shown in Figure 4.3a (Case #0), with the ideal dynamic model and without any external disturbances, the impedance controller can achieve very good setpoint regulation performance. However, in Figure 4.3b (Case #1), when an external payload-1 (25g) is attached to the robot EE, the regulation result made by the same impedance controller shifted downward due to the incomplete gravity compensation. Furthermore, as the weight of the external payload increases, the shifts get worse as shown in Figure 4.3c (Case #2) and in Figure 4.3d (Case #3).

When the iterative learning-based controller (4.10) and the update law (4.32) are implemented, the external payloads can be accurately compensated via iterative learning thus accurate regulation performance is recovered. Figure 4.4 shows the simulation results in a setpoint regulation task under different iterative learning rates

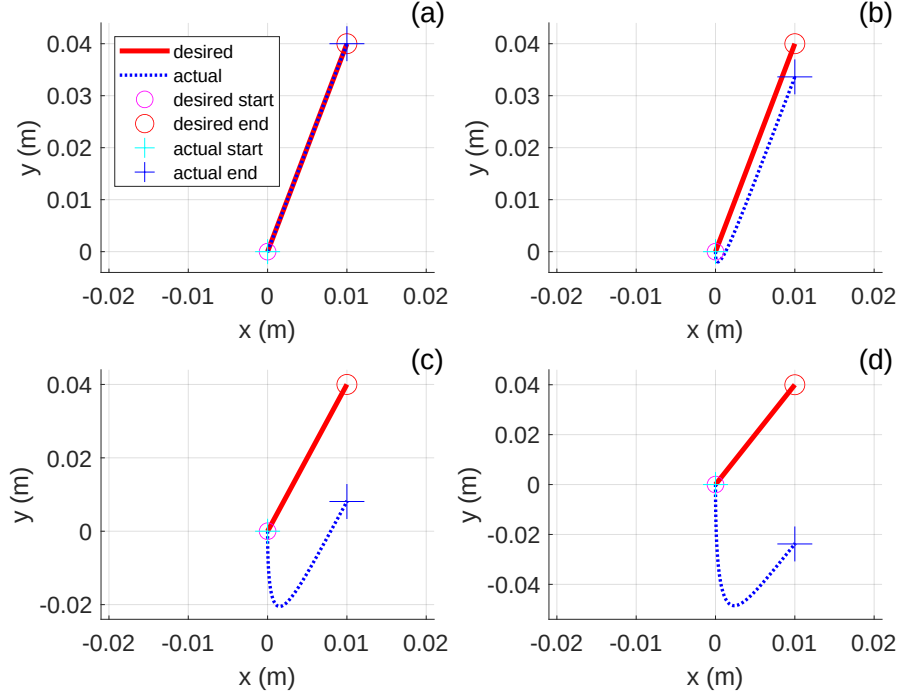


Figure 4.3: Simulation results in a setpoint regulation task with a pure impedance controller under an ideal dynamic model. (a) Case #0, no payloads; (b) Case #1, payload-1 (25g); (c) Case #2, payload-2 (125g); (d) Case #3, payload-3 (250g). The setpoint is set as $[0.01, 0.04, 0]$ m in Cartesian space.

($\eta = 0.001/0.005/0.025$). As shown in Figure 4.4a and Figure 4.4b, when an external payload-1 (25g) is attached to the robot EE (Case #1), the setpoint regulation performance (Figure 4.4a) is recovered to be accurate with the help of iterative learning on gravity compensation, while different learning rates ($\eta = 0.001/0.005/0.025$) may result in different converging behavior (Figure 4.4b). Specifically, a large learning rate (η) may result in an oscillate converging behavior while a smaller learning rate (η) may result in smooth converging behavior. Similar simulation results can be found when the weight of the payload increases (Figure 4.4c,d for payload-2 (125g), and Figure 4.4e,f for payload-3 (250g)). The getting worse oscillation behavior as the learning rate increases also indicates that the learning rate should be matching the iterative update time, which again verified the reasonability of the proposed steady-state scaling strategy.

The simulation results in the regulation task demonstrate that the incomplete

gravity compensation will cause the actual regulated position to shift downward thus the task performance is destroyed. However, with the steady-state scaling strategy and implementing the iterative learning scheme to learn for gravity compensation, the regulation accuracy can be recovered.

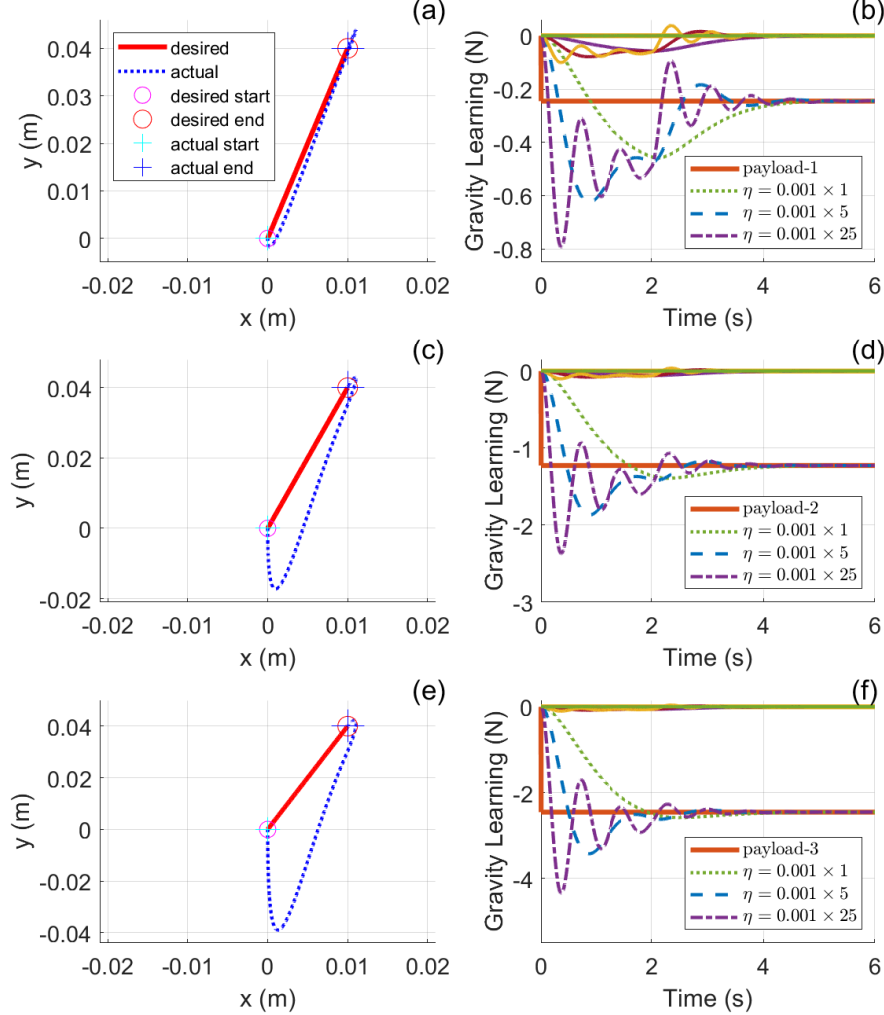


Figure 4.4: Simulation results in a setpoint regulation task under different iterative learning rates for learning gravity compensation. (a) Case #1 with payload-1 (25g); (b) Gravity learning result in Case #1; (c) Case #2 with payload-2 (125g); (d) Gravity learning result in Case #2; (e) Case #3 with payload-3 (250g); (f) Gravity learning result in Case #3. Note that the solid lines converging to zero in subfigure (b,d,f) are the learning results along the non-gravity axes in Cartesian space and their legends are ignored for clarity purposes.

4.3.5 Simulation on trajectory tracking task

By using the steady-state scaling strategy, the steady-state period can be scaled to be on the same level as the robot sampling time. By doing this, the iterative learning scheme can be extended to trajectory-tracking tasks theoretically. This will be evaluated by simulations in this section.

Similar to the procedures used in the setpoint regulation task presented in the previous section, we repeat all the procedures in the trajectory tracking task. The trajectory of concaved-square (4.33) is employed for the trajectory tracking task. The same 3DOF robot model is employed and the iterative learning-based impedance controller (4.9) with the update law (4.32) is implemented.

The simulation results in the trajectory tracking task are similar to that in the setpoint regulation task. Figure 4.5 shows the simulation results in a trajectory tracking task with a pure impedance controller (4.7) (or equivalently controller (4.9) with setting the learning rate $\eta = 0$) under an ideal dynamic model where the $\mathbf{M}$, $\mathbf{S}$, $\mathbf{G}$ are fully known. As shown in Figure 4.5a (Case #0), with the ideal dynamic model and without any external payloads attached, the impedance controller can achieve accurate trajectory tracking performance. However, in Figure 4.5b (Case #1), when an external payload-1 (25g) is attached to the robot EE, the actual trajectory made by the same impedance controller shifted downward. Again, as the weight of the external payload increases, the shifted displacements get larger as shown in Figure 4.5c (Case #2) and Figure 4.5d (Case #3).

When the learning-based impedance controller (4.9) is implemented with the steady-state scaling strategy, the external payloads can be compensated via iterative learning thus accurate tracking performance can be recovered. Figure 4.6 shows the simulated tracking performance under different iterative learning rates ($\eta = 0.001/0.005/0.025$). As shown in Figure 4.6a and Figure 4.6b, when an external payload-1 (25g) is attached to the robot EE (Case #1), the trajectory tracking performance (Figure 4.6a)

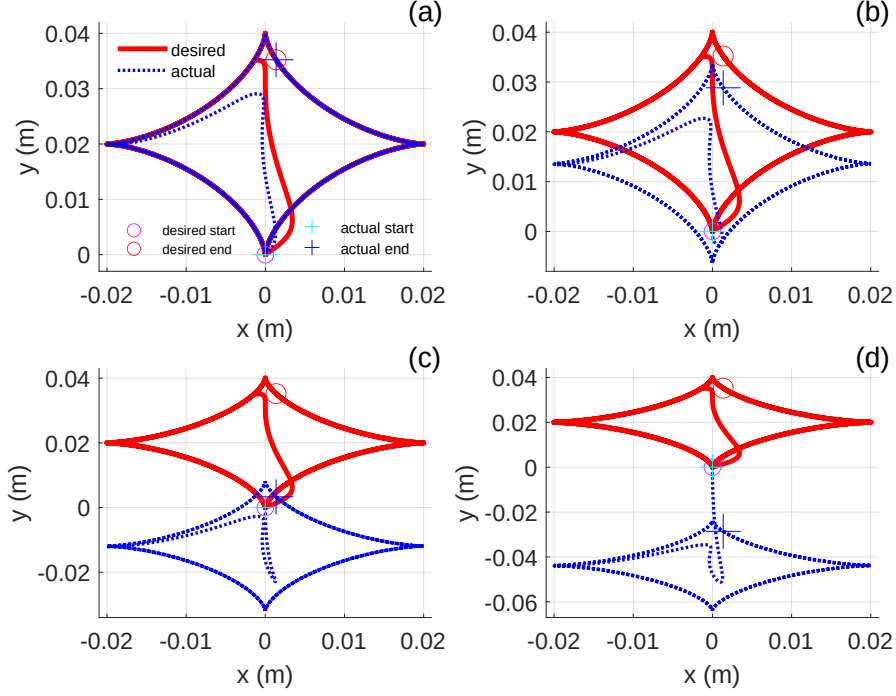


Figure 4.5: Simulation results in a trajectory tracking task with a pure impedance controller under an ideal dynamic model. (a) Case #0, no payloads; (b) Case #1, payload-1 (25g); (c) Case #2, payload-2 (125g); (d) Case #3, payload-3 (250g).

is recovered to be accurate with the help of iterative learning on gravity compensation, while different learning rates may have different converging behaviors (Figure 4.6b) which is affected by the learning rate η . Similar to the observed phenomenons in the setpoint regulation task, a larger learning rate may have an oscillate converging behavior while a smaller learning rate may have a slow but smooth converging behavior. Similar simulation results can be found when the weight of the payload increases (Figure 4.6c,d for payload-2 (125g), and Figure 4.6e,f for payload-3 (250g)).

The simulation results in the trajectory tracking task demonstrate that, by using the steady-state scaling strategy, the iterative learning scheme for gravity compensation is also valid when a robot is in a free-motion mode. With an appropriate setting on the learning rate, the iterative learning term is able to converge to the actual weight of the external payload.

If we take a comparison on the gravity learning behavior in the trajectory tracking

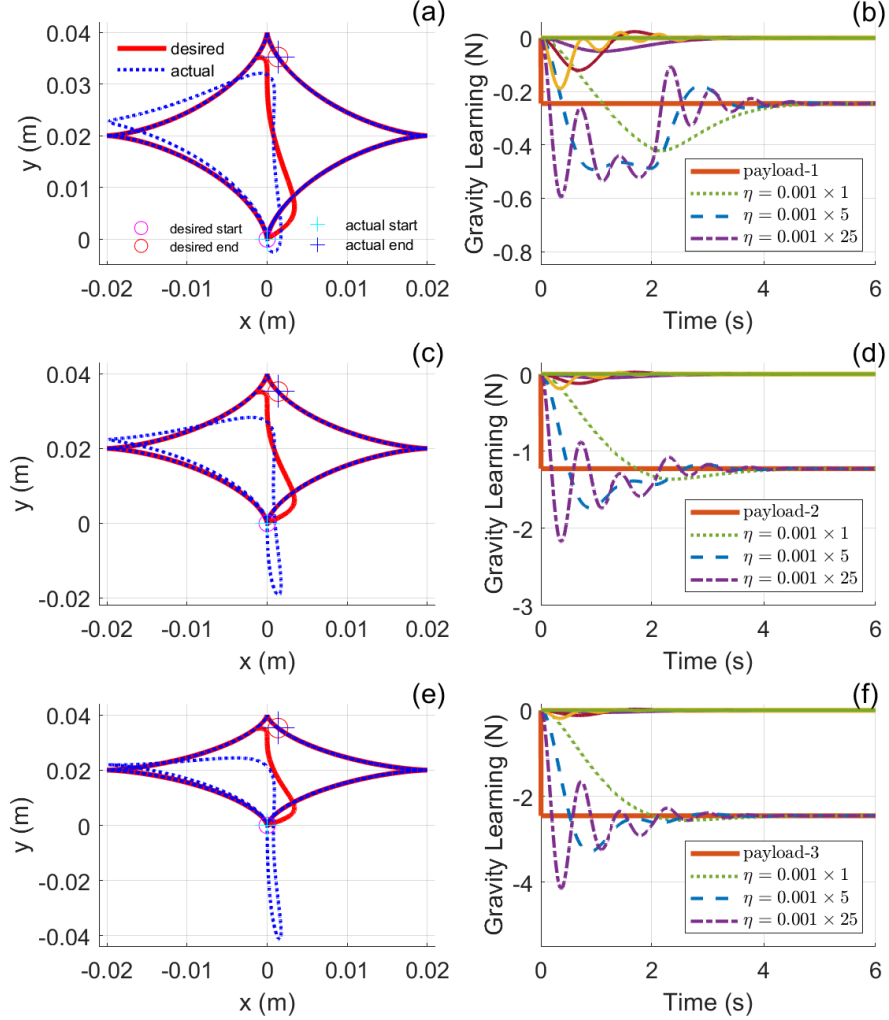


Figure 4.6: Simulation results in a trajectory tracking task under different iterative learning rates for learning gravity compensation. (a) Case #1 with payload-1 (25g); (b) Gravity learning result in Case #1; (c) Case #2 with payload-2 (125g); (d) Gravity learning result in Case #2; (e) Case #3 with payload-3 (250g); (f) Gravity learning result in Case #3. Note that the solid lines converging to zero in subfigure (b,d,f) are the learning results along the non-gravity axes in Cartesian space and their legends are ignored for clarity purposes.

task (Figure 4.6b,d,f) with that in the setpoint regulation task (Figure 4.4b,d,f), and put them in a same figure as shown in Figure 4.7, we can clearly found that the gravity learning behavior is very similar and has almost the same converging process. Especially in Figure 4.7c, the gravity learning behavior is almost the same in the two scenarios. The results in Figure 4.7 indicate that by using the steady-state scaling strategy, the iterative learning scheme for gravity compensation can be used for both setpoint regulation tasks and trajectory-tracking tasks, while their converging process

are almost the same. This verified the feasibility of extending the iterative learning scheme to scenarios of robots in free motions.

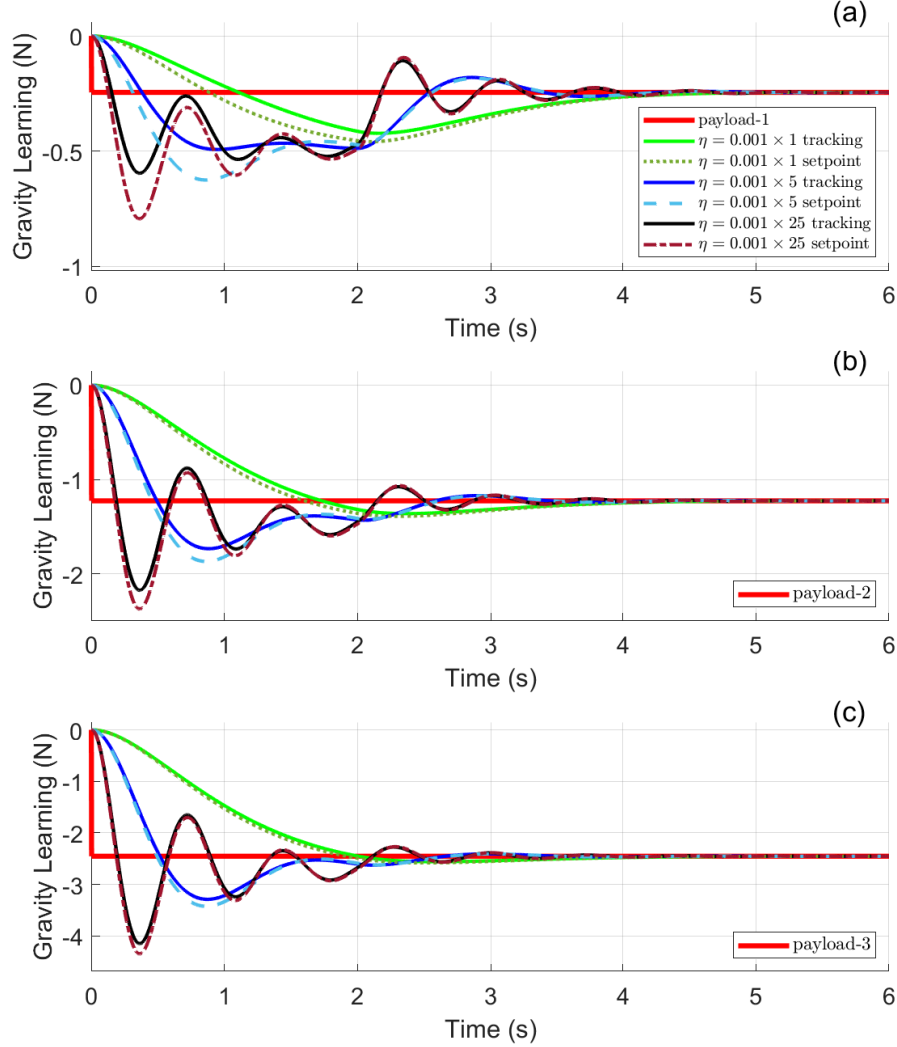


Figure 4.7: Simulation results of comparing the gravity learning behavior in a trajectory tracking task and in a setpoint regulation task under different iterative learning rates. (a) Gravity learning results in Case #1 with payload-1 (25g); (b) Gravity learning results in Case #2 with payload-2 (125g); (c) Gravity learning results in Case #3 with payload-3 (250g).

4.3.6 Experiment on trajectory tracking task

In contrast to simulations, a series of experiments are conducted to evaluate the presented iterative learning-based controller by using a 3DOF Phantom Premium 1.5A robot. The trajectory of concaved-square (4.33) is employed.

The experimental results of trajectory tracking performance in different conditions are shown in Figure 4.8. Figure 4.8a shows the tracking performance when only an impedance controller (4.7) is implemented with relatively small impedance gains. Note that inherent uncertainties of the physical robot system, including but not limited to dynamic model error and unmodeled friction, always exist in all physical experiments. As shown in Figure 4.8a, the tracking performance is significantly affected due to the inherent uncertainties.

A straightforward way to overcome the inherent uncertainties is to make the robot stiffer by tuning the impedance gains. When increasing the robot stiffness (see Table 4.1 for increased stiffness by tuning impedance gains), the inherent uncertainties can be overcome thus accurate tracking can be obtained. However, solely using increased stiffness to overcome the uncertainties is limited when external payloads are attached to the robot EE, especially for heavy external payloads. As shown in Figure 4.8c, when payload-1 (25g) is attached, although most of its mass can be overcome by the increased stiffness, there still have significant shifting-down for the actual trajectory. Especially when two payload-1 (50g in total, all attached since the beginning) are attached, the shift gets larger as shown in Figure 4.8d. The effect of increasing the stiffness could be very good in simulations in order to achieve accurate task performance, but it is limited in practice since too large stiffness can cause robot chattering and thus unstable. Therefore, in practice, smaller impedance gains are used at the cost of task performance accuracy. From Figure 4.8c,d, we can clearly observe that the inaccuracy part is a shift downward away from the desired trajectory which is mainly caused by incomplete gravity compensation.

To reduce the effect brought by incomplete gravity compensation, the iterative learning scheme is employed. By implementing the learning-based impedance controller (4.9), the effect of external payload-1 (25g) can be effectively compensated (Figure 4.8e, Figure 4.8f). Furthermore, a second payload-1 (25g) can also be effectively compensated (Figure 4.8g, Figure 4.8h). One can notice that in Figure 4.8f and

Figure 4.8h, the learning term has bounded errors and cannot converge to the exact weight of the payloads. By comparing with the simulation results on trajectory tracking tasks, we can reasonably conclude that the bounded learning errors are caused by inherent uncertainties in the physical robotic system (*e.g.*, inaccurate dynamic model, joint friction, etc.). In other words, the iterative learning term estimates gravity plus a part of the other uncertainties. This is verified with a further experiment where NDOB is employed to estimate the lumped uncertainties, and the results are shown in Figure 4.9. In the figure, we can see that with an appropriately high learning rate ($\eta = 0.001 \times 10$), the Git algorithm can accurately estimate the lumped uncertainties as the same as the NDOB does. While with a low learning rate ($\eta = 0.001 \times 1$), the iterative learning algorithm can still accurately estimate the gravity part (Figure 4.9b), but only a rough estimation for the other uncertainties (Figure 4.9a). Note that NDOB is a specific type of observer among a variety of disturbance observers, and it is selected here as a reference due to its high accuracy in estimating the lumped uncertainties and its ability to estimate the nonlinearities in the dynamics [78].

4.3.7 Experiment on setpoint regulation task

By implementing the iterative learning-based controller (4.10), experiments on setpoint regulation involving physical human-robot interaction (*pHRI*) are conducted in two scenarios, *i.e.*, *pHRI* disabled, and *pHRI* enabled. Figure 4.10 shows setpoint regulation performance under the iterative learning-based controller when *pHRI* is involved. As shown in Figure 4.10a and Figure 4.10b, when the *pHRI* is disabled the robot will reject human-applied force and keep the robot EE remain at a fixed position. This realizes one expected condition in our application, *i.e.*, the robot holds with an arthroscope and keeps it still while rejecting all potential disturbances. When *pHRI* is enabled (Figure 4.10c, Figure 4.10d), the robot EE can be freely moved by the human user to wherever the user wanted. This realizes another expected condition in our application, *i.e.*, the robot allows the surgeon to freely move it to a new

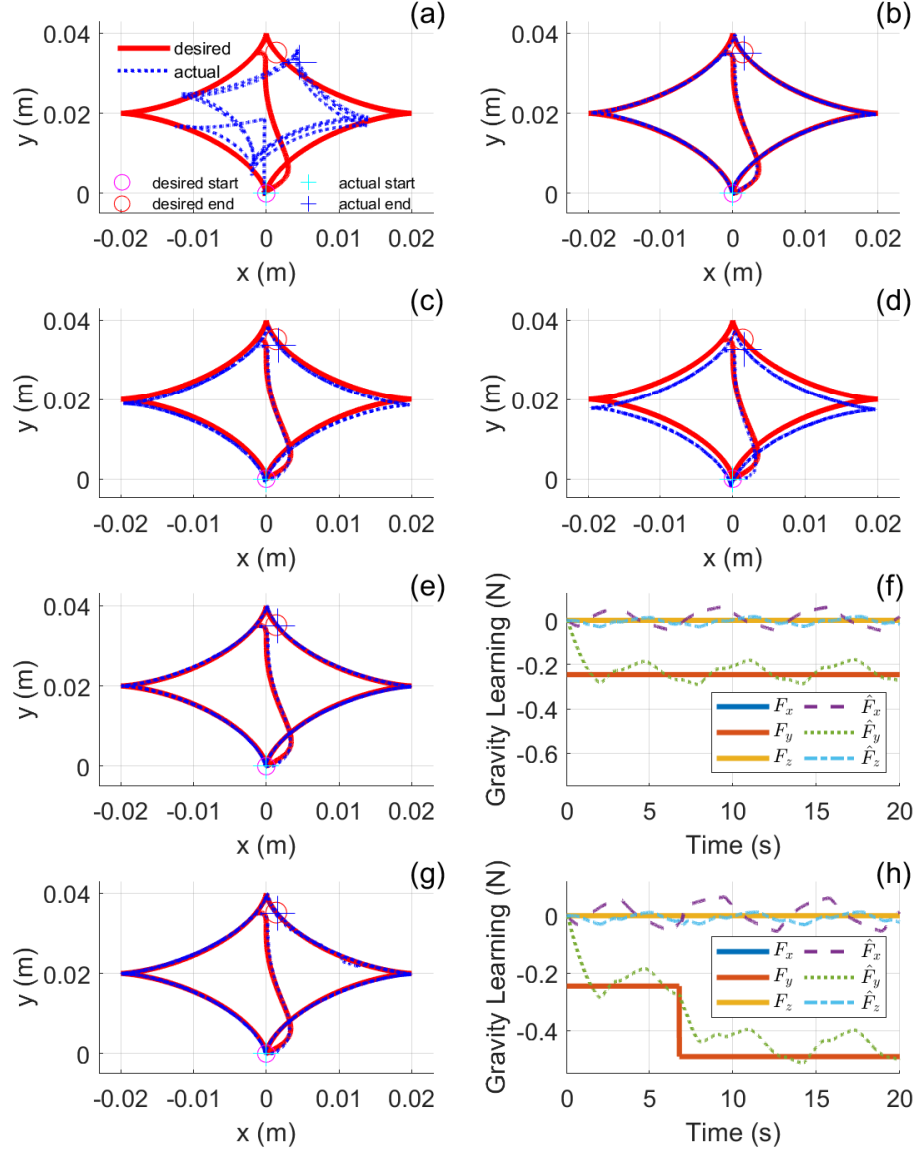
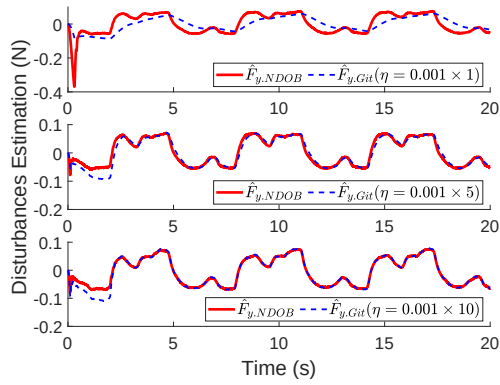


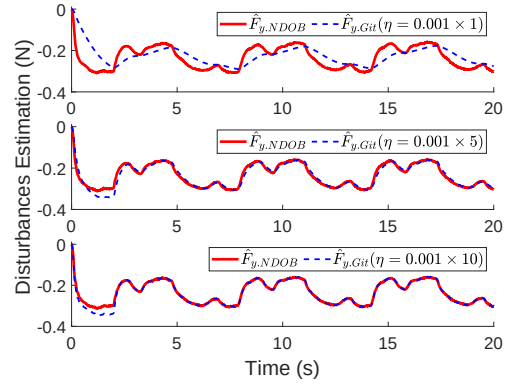
Figure 4.8: Experimental results of a trajectory tracking task in different scenarios. (a) Impedance controller only; (b) Impedance controller only, but with increased robot stiffness; (c) Increased robot stiffness with payload-1 (25g); (d) Increased robot stiffness with two payload-1 (50g in total, both attached since the beginning); (e) Increased robot stiffness with payload-1 and iterative learning; (f) Iterative learning result with payload-1; (g) Increased robot stiffness with two separate payload-1 (25g+25g) and iterative learning; (h) Iterative learning result with two separate payload-1. Note, for (g),(h) where two payload-1 appear, the first payload-1 is attached since the beginning while the second payload-1 is attached at around the 6th second.

position for adjusting the arthroscope view when necessary.

A further evaluation is to implement the controller in an application scenario mimicking robot-assisted arthroscopic surgery with a FAST (fundamentals of arthroscopic



(a) Without external payloads.



(b) With external payload-1 (25g).

Figure 4.9: Experimental results of disturbance estimation by NDOB and Git in trajectory tracking tasks. Note that only estimation along the y -axis is displayed for clarity since gravity is along the y -axis in this work.

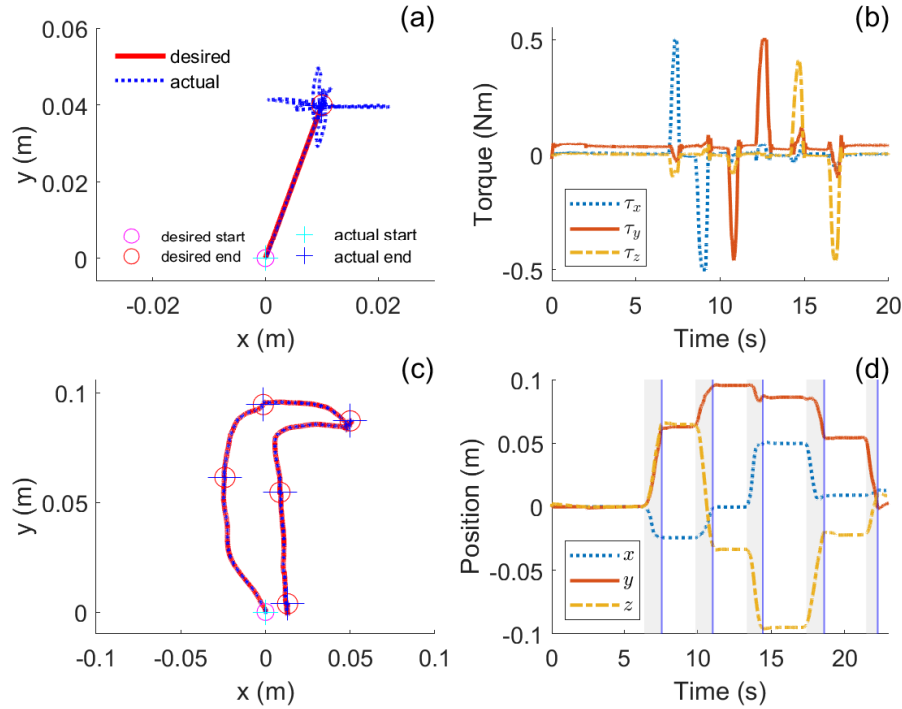


Figure 4.10: Experimental results in a setpoint regulation task involving pHRI. (a) Trajectory when pHRI disabled; (b) Computed torque when pHRI disabled; (c) Trajectory when pHRI enabled; (d) Actual position when pHRI enabled. Note, the five shaded areas in (d) indicate five times of interaction during which the user moves the robot EE from one point to another as shown in (c). The five vertical blue lines in (d) are the time points corresponding to the five actual endpoints in (c).

surgery training) simulator as shown in Figure 4.11a. In the application scenario, the robot EE is expected to hold with an arthroscope still while rejecting all potential disturbances. And when necessary, the arthroscope can be freely moved to a new position for adjusting the scope view. The experimental results of this application scenario are shown in Figure 4.11b. The two shaded gray areas in Figure 4.11b represent two periods of holding the arthroscope still by the robot with different scope views. And during these two periods, we can see that the robot EE position is accurately kept constant which verified the effectiveness of the implemented iterative learning-based controller.

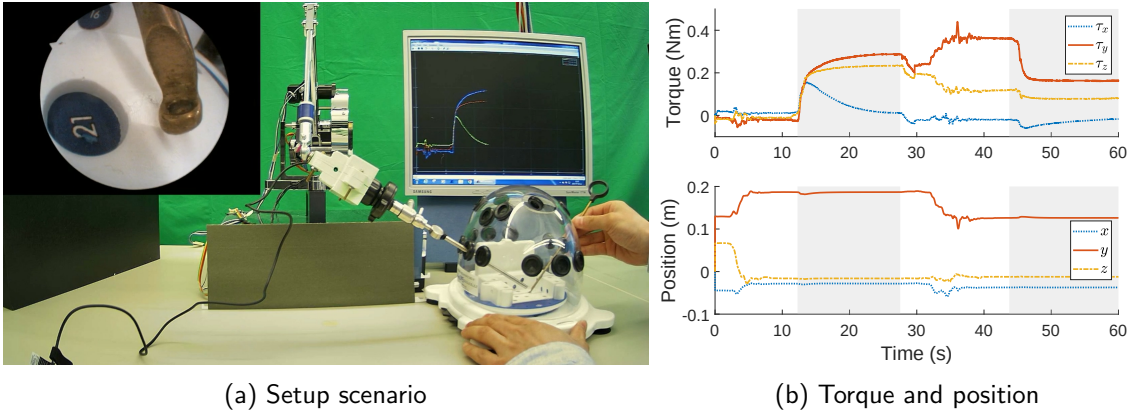


Figure 4.11: Robot-assisted arthroscopy with a FAST simulator. Gray areas represent two periods of holding the arthroscope still by the robot with different scope views.

4.3.8 Comparing with other methods

The presented Git scheme in this work extended the validity of the prior work [26] into both trajectory tracking tasks and setpoint regulation tasks. The simulation and experimental results demonstrate the good performance of the Git scheme in learning and compensating for gravity. The disturbance observers like NDOB can accurately estimate the lumped uncertainties including gravity, but it is not suitable for *pHRI* scenarios since it will prevent human-robot interaction [78, 75]. An adaptive controller can also compensate for dynamic uncertainties including gravity

[130, 59], but it is a controller rather than an independent strategy of disturbance estimation. Moreover, as a controller, it cannot provide compliant robot behavior like an impedance controller can do. The main differences among these methods are summarized in Table 4.2.

In future work, we will further investigate the similarities and differences in the performance of simulations and experiments by comparing with the methods in the literature, *e.g.*, NDOB, adaptive controller, and conventional PID controller. Also, we will use the full impedance model without simplifications which will enable the inertia term to be tunable.

Table 4.2: Comparing with methods in literature.

Methods	Uncertainties	Applicability	Requirements
Git	Gravity	①, ②	$\mathbf{x}, \mathbf{x}_d, \mathbf{J}^T$
Git [26]	Gravity	①	$\mathbf{q}, \mathbf{q}_d$
NDOB [78]	All	①, ②	$\dot{\mathbf{q}}, \mathbf{M}^{-1}, \mathbf{S}, \mathbf{G}$
Adapt [130, 59]	Dynamics	①, ②	$\mathbf{q}, \dot{\mathbf{q}}, \mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, \mathbf{Y}$

Note: ① Setpoint regulation tasks; ② Trajectory tracking tasks; subscript $_d$ means “desired”; $\mathbf{Y}$ is the regressor matrix in a linearized dynamic model which is derived via sophisticated process based on the general dynamic model, which means that $\mathbf{M}, \mathbf{S}, \mathbf{G}$ are implicitly required in order to obtain $\mathbf{Y}$.

4.4 Conclusion

In this chapter, we presented a simple and compact gravity iterative learning (Git) scheme for gravity compensation in Cartesian space. The whole process of developing the Git scheme is presented in detail, including motivation, theoretical analysis, simulations, experiments, and application. First, the convergence properties are theoretically analyzed. Then, a steady-state scaling strategy is proposed to improve the Git scheme which also extends its validity to more general trajectory tracking scenarios. By integrating the Git scheme with an impedance controller, an iterative

learning-based impedance controller is constructed, where the Git algorithm can accurately learn for gravity compensation while the impedance controller can provide a robot with compliant behavior thus ensuring a safe human-robot interaction in *p*HRI scenario. The learning accuracy of the Git scheme together with the scaling strategy are verified by simulations on both setpoint regulation tasks and trajectory tracking tasks. The effectiveness of the learning-based controller is further validated by physical experiments on both trajectory tracking tasks and setpoint regulation tasks. An application experiment in a simplified scenario of robot-assisted arthroscopic surgery also evaluated the effectiveness of the implemented learning-based controller. The results demonstrated that the integrated controller can achieve good tracking performance and regulation accuracy when heavy external payloads are attached to the robot EE. Moreover, it allows seamless switching between setpoint regulation and human-robot interaction.

The major benefits of the presented Git scheme for gravity compensation can include (1) simple and compact formulation and no need for the robot dynamics, (2) no need for any information about external payloads, (3) no need for higher impedance gains for reducing the effects of incomplete gravity compensation, and (4) it is valid for both setpoint regulation tasks and trajectory tracking tasks.

Chapter 5

Point-Based 3D Virtual Fixture Generating Method¹

Virtual fixture (VF) has been playing a vital role in robot-assisted surgeries, such as guiding surgical tools' movement and protecting a beating heart. In orthopedic surgery, preplanned images are often used in the operating room, on which planning curves might be drawn, for instance, to mark out the boundaries for osteophytes to be removed. These curves can be used to generate VF to assist in removing osteophytes during the operation. A challenge is that the hand-drawn curves usually have a random shape and cannot be mathematically represented by equations, thus most of the existing algorithms will not work in this scenario. In this Chapter, an algorithm of VF generating based on point clouds is presented, with which VF can be generated directly from cloud points, for example, point clouds of hand-drawn curves extracted from an image. The effectiveness of the VF algorithm is evaluated by a series of simulations and experiments. The VF algorithm is also tested in an image-based scenario and its effectiveness is demonstrated.

¹A version of this chapter has been published as: Teng Li, Armin Badre, Hamid D. Taghirad, Mahdi Tavakoli, "Point-Based 3D Virtual Fixture Generating for Image-Guided and Robot-Assisted Surgery in Orthopedics", 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2023), June 28-30, Seattle, WA, USA, 2023, pp. 179-186. [\[IEEE Xplore\]](#)

5.1 Background

Virtual fixture (VF), also known as active constraint and first proposed in [110], is usually categorized into two types according to its purpose, that is guidance virtual fixture (GVF) and forbidden-region virtual fixture (FRVF) [1, 14]. Intuitively, the GVF serves like a ruler to assist in drawing a straight line, while the FRVF serves like an armor to prevent tool tip from entering a protected area. Both types play a vital role during various surgical procedures in robot-assisted surgery, such as suturing [86], knot tying [150], dissection [121], either assisting in moving the surgical tool along a trajectory or preventing it from entering a specific area for protecting the objects (*e.g.*, beating heart or nerve) inside [112, 92].

VF is usually generated based on geometric elements, such as lines, planes, surfaces, and volumetric primitives [14]. The vector field approach is the most common one to be used for VF generating, which works for any shape that can be expressed as mathematical equations [155]. The advantage of the vector field approach is that it is simple, straightforward, and stable, while the disadvantage is that it requires an explicit/implicit mathematical representation for the object to be modeled as VF. For objects with regular shapes like cubes and spheres, their mathematical representations can be easily established then the VF can be constructed relatively easily. However, for objects with irregular shapes like a humerus bone or skull, they may not be able to be expressed by equations, then the vector field approach may not work for these objects anymore.

The vector field approach cannot correctly handle situations of being in contact with multiple objects simultaneously and situations of thin objects. To solve this problem, Zilles and Salisbury developed a constraint-based god-object algorithm [155]. In their work, the god-object is a proxy of the haptic interface point (HIP) which is attached to the HIP when it is in free motion. Once the HIP encounters VF (*e.g.*, a virtual wall), the proxy will always remain on the top of the virtual wall and

never penetrate into it. This is ensured in their algorithm by applying the Lagrange multiplier technique on a set of active constraints to find the position of the proxy in each servo loop. Meanwhile, the virtual spring and/or damper linkage between the HIP and the proxy will render a haptic force that tries to pull the HIP back out of the virtual wall.

Kapoor *et al.* developed a constrained optimization method for generating VF [63], in which a suitable objective function is required in order to do the optimization. In the method, five basic geometric constraints are established as VF task primitives which can be used for assembling customized VF. With similar techniques, Marinho *et al.* employed a method of vector-field inequalities to generate VF for collision avoidance [87] and guidance in a looping task during suturing [86]. Xia *et al.* developed a constrained optimization framework of VF generating for multi-robot collaborative teleoperation tasks, *e.g.*, knot positioning [150].

There are also some other methods for VF generating for different purposes, such as potential field method for collision avoidance or guidance [22, 120] and nonenergy storing method for a more stable robot behavior [65]. Readers are directed to [14] for a comprehensive review of VF.

In the field of robot-assisted surgery, VF has been widely used due to plenty of advantages, such as reducing surgeons' cognitive load [150], improving surgical performance [48], making the surgical outcome more accurate and safe. Park *et al.* [99] conducted a preliminary test before applying VF in coronary artery bypass surgery. In the test, VF is generated for a blunt dissection task at a position determined from a preoperative CT scan image, and the VF is a regular plane thus the VF generating is relatively easy in their work.

Ryden *et al.* [112] developed a method to generate VF directly from point cloud to protect the beating heart during surgery. They improved their method in [111, 96], which established a solid foundation for point-based VF algorithm.

In orthopedics, a series of preoperative images of a patient are first acquired before

the surgery. Then, some surgical preplans will be made on the images, for example, drawing some curves to mark out areas to remove osteophytes, or to protect nerves inside. During the surgery, these hand-drawn curves can be used to generate VF which can assist in removing osteophytes or protecting the nerves by providing haptic feedback. The main challenge here is that the hand-drawn curves are usually in irregular shapes which may not be able to be presented mathematically by equations that are often required by most of existing VF-generating algorithms.

Inspired by Ryden’s work and motivated to solve the challenge mentioned above, in this chapter, an algorithm for VF generating from cloud points is developed and presented. The effectiveness of the VF algorithm is evaluated by a series of simulations and experiments on geometric entities with regular or irregular shapes. Lastly, the algorithm is tested in a specific preplanned image-based scenario which can be further generalized to image-based surgery.

5.2 Methods

In this section, an algorithm for point cloud based virtual fixture (VF) generation method is presented, which includes one main algorithm and three embedded proxy algorithms for the proxy in different states. The major parameters used in the VF algorithm are summarized in [Table 5.1](#). The general idea of the VF algorithm is explained as follows. The robot end-effector (EE) position denotes as $\mathbf{P}_{\text{HIP}}$ while its virtual proxy denotes as $\mathbf{P}_{\text{proxy}}$. As illustrated in [Figure 5.1](#), taking the proxy point $\mathbf{P}_{\text{proxy}}$ as the center, three spheres with radii ($r_1 < r_2 < r_3$) are defined as proxy regions while a contacting region r_c is determined by $r_c = (r_1 + r_2)/2$. When there is no contact between the robot EE and the point cloud, the $\mathbf{P}_{\text{proxy}}$ always coincides with $\mathbf{P}_{\text{HIP}}$, whereas they may be detached from each other in order to generate force feedback when a contact or penetration occurred. The state of the proxy ($state_{\text{proxy}}$) will be determined as $-1/0/1/2$ based on the relative position relationship between the proxy regions and the point cloud, *i.e.*, no neighbor (-1), in free motion (0),

in contact (1), entrenched (2). The last three states may be combined together as a state of in-neighbor (0/1/2) for some explanations. For each proxy state, a step vector $\hat{\mathbf{s}}$ will be determined by the proxy algorithms which will be introduced later in this section. Finally, the determined $\hat{\mathbf{s}}$ will be used to move the proxy point at the end of each servo loop while ensuring that it is always on the surface of the point cloud and never penetrates into it.

The general idea described above will be explained in detail in the remaining part of this section based on one main algorithm ([algorithm 1](#)) and three proxy algorithms ([algorithm 2](#), [algorithm 3](#), [algorithm 4](#)). An illustration of the proxy in different states is shown in Figure 5.1. Note that in this chapter, point cloud refers to a set of points representing a target object, and virtual fixture refers to all areas defined by the cloud points and their radius r_v .

Table 5.1: Major parameters and description.

Parameters	Description
$\mathbf{P}_{\text{HIP}}$	haptic interface point (= robot EE point)
$\mathbf{P}_{\text{proxy}}$	proxy point, the avatar of HIP
$\mathbf{L}_{\text{pcloud}}$	point cloud list, the collection of all cloud points
$\mathbf{p}_i$	a single point in the point cloud
N	the number of points in the point cloud
$state_{\text{proxy}}$	proxy state, from $\{-1,0,1,2\}$
$r_{\{1,2,3,c\}}$	proxy regions radii, $r_1 < r_c < r_2 < r_3$, $r_c = \frac{(r_1+r_2)}{2}$
r_v	radius of each cloud point (default as 0)
$\hat{\mathbf{s}}$	step vector for proxy movement in each servo loop
$\hat{\mathbf{n}}$	normal vector of the point cloud
$\vec{u}$	a vector pointing from $\mathbf{P}_{\text{proxy}}$ to $\mathbf{P}_{\text{HIP}}$

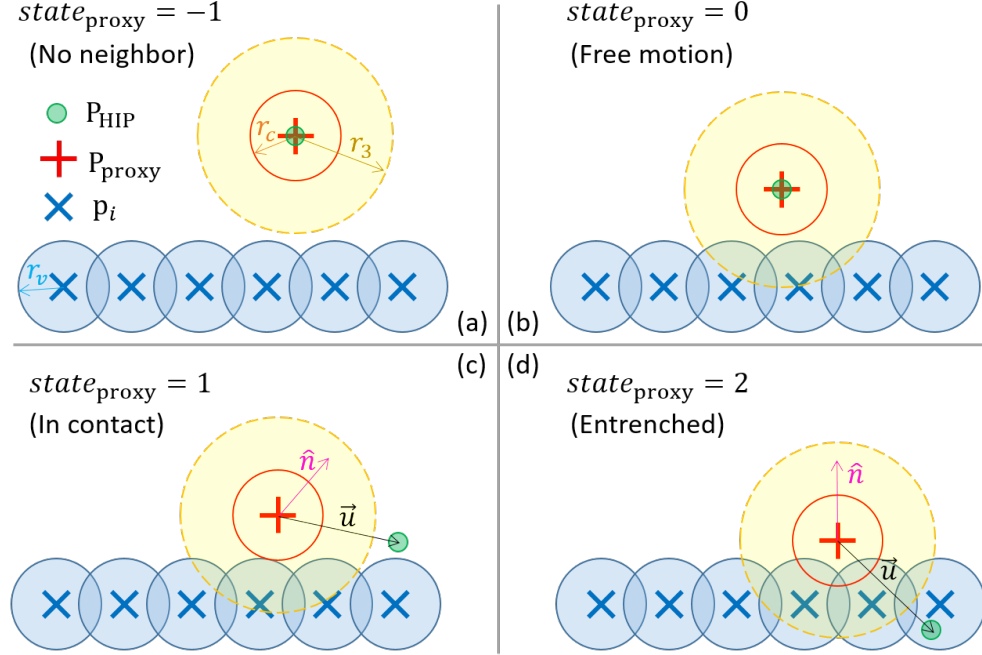


Figure 5.1: Illustration of the proxy in different states.

5.2.1 Main algorithm

The main algorithm for generating VF from point cloud is presented in [algorithm 1](#) including parameterization realized in this chapter. A set of point cloud is acquired in advance and their 3D coordinates are expressed in the robot base frame.

First of all, all parameters are initialized including the initial position of the HIP and the proxy. Note that $\mathbf{P}_{\text{HIP}}$ is initialized as $[-1, 0, 0]$ for the simulations while $[0, 0, 0]$ for the experiments. The proxy point position is initialized as the same as the HIP point position ($\mathbf{P}_{\text{proxy}} = \mathbf{P}_{\text{HIP}}$).

In each servo loop of the VF algorithm, the HIP position ($\mathbf{P}_{\text{HIP}}$) will first be updated as the real-time position of the robot EE. Then, the distance de between the proxy position ($\mathbf{P}_{\text{proxy}}$) and each point in the cloud will be calculated. Based on the distance de , each point in the cloud will be categorized into one of four lists, *i.e.*, list of entrenched ($\mathbf{L}_{\text{Entrenched}}$), in contact ($\mathbf{L}_{\text{InContact}}$), in free motion ($\mathbf{L}_{\text{FreeMotion}}$), and out neighbor ($\mathbf{L}_{\text{OutNeighbor}}$), respectively. Note that the first three lists together composed a new list of in-neighbor ($\mathbf{L}_{\text{InNeighbor}}$), while all the four lists together composed the

whole point cloud. Then, based on the number of points in each of the four lists, the proxy state ($state_{\text{proxy}}$) can be determined as one of the four states, *i.e.*, state of entrenched (2), state of in contact (1), state of in free motion (0), state of no neighbor (-1). Note that for high computing efficiency, the point cloud needs to be treated as a whole matrix when calculating the distance de and doing the categorization.

A normal vector ($\hat{\mathbf{n}}$) needs to be determined when the proxy is in state of in contact or entrenched ($state_{\text{proxy}} = 1/2$). The normal vector ($\hat{\mathbf{n}}$) is originated from the proxy point, normal to the local surface formed by the point cloud, and pointing outwards. The normal vector ($\hat{\mathbf{n}}$) is determined by all the cloud points that fell in the proxy neighbor region ($\mathbf{L}_{\text{InNeighbor}}$). Let $\mathbf{p}_i$, $i = 1, 2, \dots, M$ be the points fell in $\mathbf{L}_{\text{InNeighbor}}$, then the normal vector $\hat{\mathbf{n}}$ can be determined by

$$\begin{cases} \vec{nk} = \sum_{i=1}^M \frac{\mathbf{P}_{\text{proxy}} - \mathbf{p}_i}{\|\mathbf{P}_{\text{proxy}} - \mathbf{p}_i\|_2} \phi(r) \\ \hat{\mathbf{n}} = \frac{\vec{nk}}{\|\vec{nk}\|_2} \quad (\text{normalization}) \end{cases} \quad (5.1)$$

where $\phi(r)$ is a modified version of the Wendland function [149, 111] given by (5.2), which can provide a smoothly and monotonically decreasing between r_1 and r_3 .

$$\phi(r) = \begin{cases} 1 & \text{for } r \in [0, r_1] \\ [1 + \frac{4(r-r_1)}{r_3-r_1}](1 - \frac{r-r_1}{r_3-r_1})^4 & \text{for } r \in (r_1, r_3) \\ 0 & \text{for } r \in [r_3, +\infty) \end{cases} \quad (5.2)$$

where $r = \|\mathbf{P}_{\text{proxy}} - \mathbf{p}_i\|_2 - r_v$ is the distance between a cloud point and the proxy point while the cloud point is considered as a sphere with a radius of $r_v \geq 0$.

Finally, in each of the four states ($state_{\text{proxy}} = -1/0/1/2$), a step vector ($\hat{\mathbf{s}}$) of the proxy movements will be determined based on proxy algorithms (algorithm 2, algorithm 3, algorithm 4) that will be introduced subsequently. Once the step vector ($\hat{\mathbf{s}}$) is determined, the proxy point position ($\mathbf{P}_{\text{proxy}}$) can be updated correspondingly.

5.2.2 Proxy algorithms

In this subsection, three proxy algorithms are presented and explained in detail in order to determine the step vector ($\hat{\mathbf{s}}$) in different proxy states, *i.e.*, algorithm 2 for

state of in free motion ($state_{\text{proxy}} = 0$), [algorithm 3](#) for state of in contact ($state_{\text{proxy}} = 1$), [algorithm 4](#) for state of entrenched and no neighbor ($state_{\text{proxy}} = -1/2$). Then, the determined step vector ($\hat{\mathbf{s}}$) will be used to move the proxy a step in each servo loop while ensuring the proxy point does not penetrate into the point cloud.

(a) State of in free motion ($state_{\text{proxy}} = 0$)

The proxy movement algorithm for this state is presented in [algorithm 2](#). When the proxy is in free motion state ($state_{\text{proxy}} = 0$) and assuming the HIP is going to penetrate into the point cloud, the proxy needs to move towards the HIP in aiming to be in contact with the point cloud. In this scenario, the step vector ($\hat{\mathbf{s}}$) for the proxy movement can be determined based on the cloud points fell in $\mathbf{L}_{\text{InNeighbor}}$ (note that now $\mathbf{L}_{\text{InNeighbor}} = \mathbf{L}_{\text{FreeMotion}}$) by solving for $d0$ in (5.3).

$$r_c + r_v = \|\mathbf{p}_i - \mathbf{P}_{\text{proxy}} - d0_i \frac{\vec{u}}{\|\vec{u}\|_2}\|_2 \quad (5.3)$$

where $\vec{u} = \mathbf{P}_{\text{HIP}} - \mathbf{P}_{\text{proxy}}$ is a vector pointing from $\mathbf{P}_{\text{proxy}}$ to $\mathbf{P}_{\text{HIP}}$, and $\mathbf{p}_i, i = 1, 2, \dots, M$ is the cloud point fell in $\mathbf{L}_{\text{InNeighbor}}$, and $d0_i$ is a scalar step size needs to be solved for the i th point in $\mathbf{L}_{\text{InNeighbor}}$. Therefore, this procedure has to be done M times. After that, step size d will be determined by the minimum value of $d0_i$, *i.e.*, $d = \min(d0_i)$. Then, the step vector ($\hat{\mathbf{s}}$) will be determined by $\hat{\mathbf{s}} = d * \frac{\vec{u}}{\|\vec{u}\|}$ which means to bring the proxy point a step towards the HIP point.

Before updating the proxy point position, three special scenarios may need to be considered:

- If $d = 0$ and the HIP is inside of the point cloud, then a projection vector $\vec{u}_p$, which is obtained by projecting $\vec{u}$ onto the normal plane determined by the normal vector $\hat{\mathbf{n}}$, will be used to determine d (5.5). In this scenario, the step vector will be determined by $\hat{\mathbf{s}} = d * \frac{\vec{u}_p}{\|\vec{u}_p\|}$.
- If the HIP is outside of the point cloud and $\frac{1}{M} \sum_{i=1}^M \|\mathbf{p}_i - \mathbf{P}_{\text{HIP}}\|_2 > \frac{1}{M} \sum_{i=1}^M \|\mathbf{p}_i - \mathbf{P}_{\text{proxy}}\|_2$ ($\mathbf{p}_i \in \mathbf{L}_{\text{InNeighbor}}$), *i.e.*, the HIP is moving away from the point cloud,

then set $\hat{\mathbf{s}} = \mathbf{P}_{\text{HIP}} - \mathbf{P}_{\text{proxy}}$, which means to detach cloud-proxy and attach proxy-HIP by setting $\mathbf{P}_{\text{proxy}} = \mathbf{P}_{\text{HIP}}$.

The normal vector $\hat{\mathbf{n}}$ can solely determine a normal plane that is going through the proxy point and normal to $\hat{\mathbf{n}}$. The vector $\vec{u} = \mathbf{P}_{\text{HIP}} - \mathbf{P}_{\text{proxy}}$ pointing from $\mathbf{P}_{\text{proxy}}$ to $\mathbf{P}_{\text{HIP}}$ can be projected onto the normal plane, and the projection $\vec{u}_p$ can be obtained by

$$\vec{u}_p = \vec{u} - (\hat{\mathbf{n}} \cdot \vec{u})\hat{\mathbf{n}} \quad (5.4)$$

Based on the projection $\vec{u}_p$, the step size d can be determined by

$$d = \begin{cases} \xi \|\vec{u}_p\|_2 & \text{for } \|\vec{u}_p\|_2 \leq r_1 \\ \xi r_1 & \text{for } \|\vec{u}_p\|_2 > r_1 \end{cases} \quad (5.5)$$

where $0 < \xi \leq 1$ is a constant gain used to ensure that one step size is not greater than the smallest proxy region r_1 .

(b) State of in contact ($state_{\text{proxy}} = 1$)

The proxy movement algorithm for this state is presented in [algorithm 3](#). When the proxy and the point cloud are in contact ($state_{\text{proxy}} = 1$), the step size d will be determined by the projection $\vec{u}_p$ via (5.5). Then, the proxy step vector $\hat{\mathbf{s}}$ will be determined by

$$\hat{\mathbf{s}} = \begin{cases} d * \frac{\vec{u}_p}{\|\vec{u}_p\|_2} & \text{for HIP is inside of VF} \\ d * \frac{\vec{u}}{\|\vec{u}\|_2} & \text{for HIP is outside of VF} \end{cases} \quad (5.6)$$

where d is the step size, $\vec{u}_p$ indicates a direction tangential to the VF, $\vec{u}$ indicates a direction pointing from the proxy to the HIP, while whether the HIP is inside or outside of VF is determined by the angle between the vector $\hat{\mathbf{n}}$ and $\vec{u}$.

(c) State of entrenched ($state_{\text{proxy}} = 2$)

The proxy algorithm for this state is presented in [algorithm 4](#). When the proxy is entrenched into the point cloud ($state_{\text{proxy}} = 2$) occasionally, the proxy needs to be moved onto the top of the point cloud surface with a single step. The step size d can be determined by solving for $d2_i$ in (5.7).

$$r_c + r_v = \|\mathbf{p}_i - \mathbf{P}_{\text{proxy}} - d2_i * \hat{\mathbf{n}}\|_2 \quad (5.7)$$

where $\mathbf{p}_i \in \mathbf{L}_{\text{Entrenched}}$, $i = 1, 2, \dots, M$, and $d2_i$ is a scalar step size corresponding to the i th cloud point. Therefore, this procedure has to be done M times. Then, the step size d will be determined by the maximum value of $d2_i$, *i.e.*, $d = \max(d2_i)$, which means that a max step size will be used to bring the proxy point out of the point cloud along the direction of the normal vector $\hat{\mathbf{n}}$.

(d) State of no neighbor ($state_{\text{proxy}} = -1$)

When there is no cloud point in the neighbor region of the proxy ($state_{\text{proxy}} = -1$), simply set $\hat{\mathbf{s}} = \mathbf{P}_{\text{HIP}} - \mathbf{P}_{\text{proxy}}$. This means that the proxy point ($\mathbf{P}_{\text{proxy}}$) always coincides with the HIP point ($\mathbf{P}_{\text{HIP}}$).

5.2.3 Virtual force rendering

Once the proxy point $\mathbf{P}_{\text{proxy}}$ is determined in each servo loop, the virtual force can be rendered based on the coordinates of the HIP point ($\mathbf{P}_{\text{HIP}}$) and the proxy point ($\mathbf{P}_{\text{proxy}}$). The force rendering algorithm can be expressed as

$$\mathbf{F}_{\mathbf{v}} = \mathbf{K}(\mathbf{P}_{\text{proxy}} - \mathbf{P}_{\text{HIP}}) \quad (5.8)$$

where $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix indicating the stiffness of the VF along each axis and can be tuned as necessary, and $\mathbf{F}_{\mathbf{v}}$ is the rendered VF force which will be delivered to the human user via the robot. Note that friction is not rendered for the VF in this chapter.

For implementing the VF algorithm onto a robot, in this chapter, the rendered VF force is directly added into the impedance control law as an independent term since our robot employs an impedance controller. Note that the VF algorithm is independent of the controller design and controller implementation, thus they work independently and do not affect each other in terms of functioning. For robot running with other controllers, *e.g.*, admittance controller, velocity controller, the rendered VF force may need first to be converted to a displacement or velocity by differentiation/integration or an appropriate gain.

5.3 Simulations, Experiments, and Results

In this section, a series of simulations and experiments are conducted to evaluate the effectiveness of the point-based VF algorithm. The employed point clouds are with various shapes and dimensionality (ranging from 0D to 3D). The corresponding results are presented and analyzed. The last experiment is designed to assess the VF algorithm in a preplanned image scenario which can be generalized to image-guided surgery in orthopedics.

5.3.1 Apparatus

The simulations are conducted in MATLAB/Simulink (version R2020a, MathWorks Inc., Natick, MA, USA). The MATLAB/Simulink runs on a computer with a 3.70 GHz Intel(R) Core(TM) i5-9600K CPU and a Windows 10 Education 64-bit operating system. The HIP point (*i.e.*, the robot EE) in the simulations is represented by the mouse pointer controlled by a human user, and the 2D position of the pointer is retrieved at a frequency of about 50 Hz when it moves in a MATLAB figure window.

The experiments are performed on a 7-DOF Franka Emika Panda robot (Franka Emika GmbH, Munich, Germany). The proposed VF algorithm is implemented on the Panda robot via an impedance control interface and C++ code. The Panda robot is controlled on a workstation computer of Intel(R) Core(TM) i5-8400 CPU @ 2.80 GHz $\times$ 6 with the Ubuntu 16.04 LTS (Xenial Xerus) 64-bit operating system. The control rate of the Panda robot is 1000 Hz. MATLAB/Simulink (version R2019a) with Quarc real-time control software (Quanser Inc., Markham, ON, Canada) is used for visualizing the real-time position of the robot EE, the point cloud, and the rendered VF force. The communications between the Robot Operating System (ROS) nodes of the workstation computer and the MATLAB/Simulink (version R2019a) are realized via User Datagram Protocol (UDP) at a frequency of 100 Hz. A [video](#) demonstration

for the experiments is available online¹.

5.3.2 Simulations

In this section, simulations in four scenarios are conducted separately in aiming to evaluate the point-based VF algorithm. The four scenarios are regarding to point clouds with various shapes and dimensions as follows,

1. a 0D single point, (see Figure 5.2a);
2. a 1D line segment, (see Figure 5.2b);
3. a 2D teapot, (see Figure 5.2c);
4. a 2D hand-drawn Ω symbol from image, (see Figure 5.2d).

Note that there is no scenario of a 3D point cloud is considered in the simulations due to the fact that the mouse pointer representing the HIP point cannot be controlled to accurately move in a virtual 3D space. The ability of the VF algorithm in 3D space will be evaluated by experiments.

In each of the four simulated scenarios, as shown in Figure 5.2, the HIP point represented by the mouse pointer, moves from left to right in a 2D MATLAB figure window, and during the movement, it will encounter the point cloud. The VF algorithm will calculate the position of the proxy point in real time while the proxy point is expected to remain on the surface of the point cloud and never penetrate into it. The HIP point can penetrate into the point cloud and the relative position between the HIP point and the proxy point will determine the rendered VF force. Note that the VF force is not considered in the simulations but will be rendered in the physical experiments.

The simulation results of the four scenarios are presented in Figure 5.2. As can be seen in the figure, as the HIP point (represented as green point) moves along the

¹online demo video link: <https://youtu.be/ROSREHC9zU0>

contour of the point cloud from left to right, the proxy point (represented as the center of the red circle) always stay outside of the point cloud (represented as the blue cross markers) which is expected. The proxy point with its contacting region r_c are represented by a circle in 2D space whose radius is predefined and can be tuned as necessary (in this work all $r_c = 5$ mm). The normal vector $\hat{\mathbf{n}}$ (represented as arrows in magenta color) should be always orthogonal to the local contour of the point cloud and be pointing outwards.

The simulation results indicated that the proposed VF algorithm works well for point cloud not only with regular shapes (*e.g.*, single point, line) but also with free-style irregular shapes (*e.g.*, teapot, hand-drawn Ω symbol). The effectiveness of the VF algorithm on geometric entities from 0D to 2D is well demonstrated by the simulations.

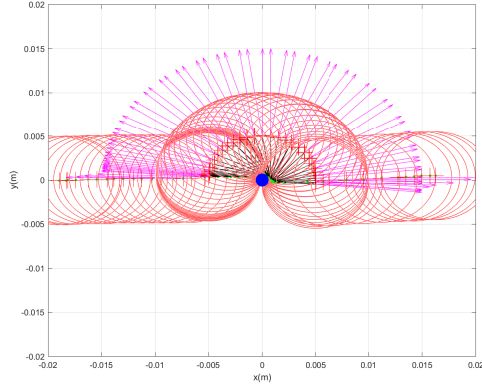
5.3.3 Experiments

Two physical experiments on a 7-DOF robot are designed to evaluate the point-based VF algorithm. In Experiment 1, a set of point cloud with a regular 2D square shape in 3D space (Figure 5.3b) is employed, while in Experiment 2 of an image-based scenario, a set of point cloud with a shape of hand-drawn curve (Figure 5.3d) from a preplanned 2D image (Figure 5.5) is employed. The point clouds are registered in the robot base frame as illustrated in Figure 5.3a, 5.3c.

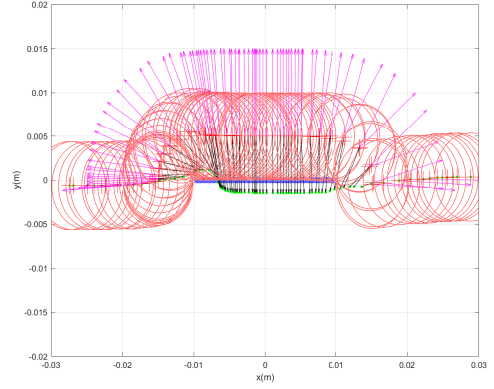
Experiment 1: 2D square

In Experiment 1, a set of point cloud with a 2D square shape is generated in the area of $x = [0.4, 0.5], y = 0.1, z = [0.4, 0.5]$ m in robot base frame, and the step size is 0.5 mm (meaning a density of 2.01 points/mm) for x -axis and z -axis. Therefore, a total of 40401(= $201 \times 1 \times 201$) points are generated for the 2D square point cloud.

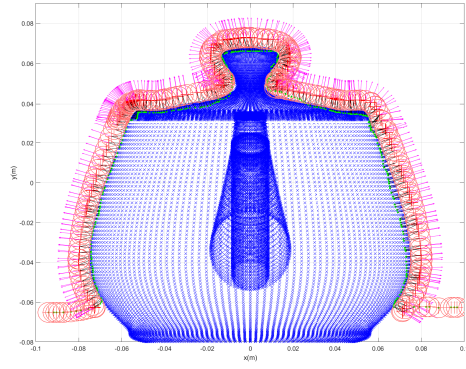
During the experiment, the user moves the robot EE to probe both sides of the VF (*i.e.*, the 2D square point cloud), and the rendered VF force, as shown in Figure 5.4a, is delivered to the user via the robot. In the figure, the green-colored area represents



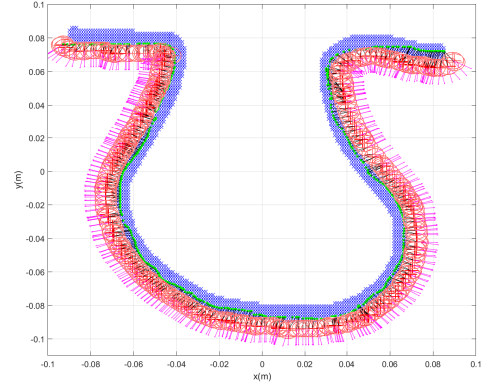
(a) A single point ($N = 1$)



(b) A line segment ($N = 100$)



(c) A 2D teapot ($N = 41472$)

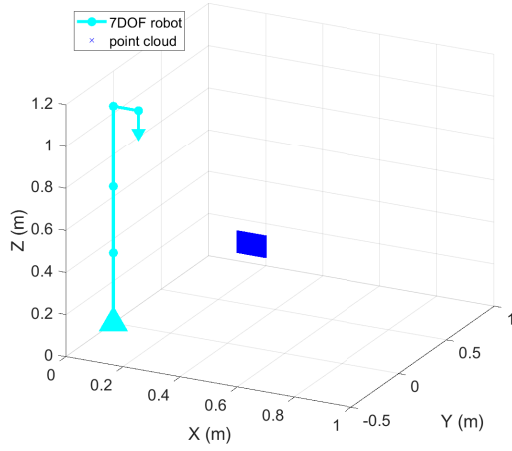


(d) An inverted Ω ($N = 2062$)

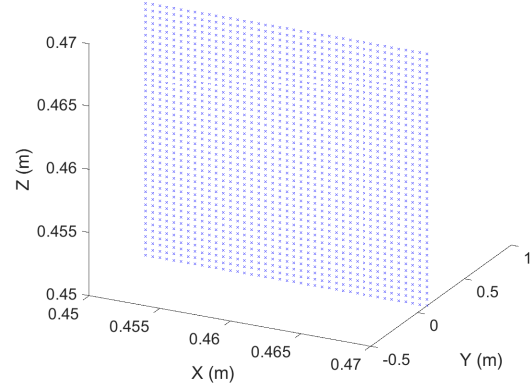
Figure 5.2: Simulation results of implementing the VF algorithm in four different scenarios. The HIP point is represented by green point, the proxy contacting region is represented by red circle while the center represents the proxy point, the normal vector $\hat{\mathbf{n}}$ is represented by magenta arrow. Black arrow represents the vector $\vec{u}$ pointing from the proxy to the HIP. The point cloud of a 2D teapot is obtained from MATLAB via command `pcread('teapot.pl')`, then scaled in this work. Note that the movement of the green point is controlled by a human user via mouse, thus the trajectory is irregular and the speed is not constant.

a specific trial. The trajectory of the specific trial is shown in Figure 5.4b, with the proxy and its contacting region (red circle and its center), and the normal vector $\hat{\mathbf{n}}$ (magenta arrow) are visualized in a frequency of 10 Hz.

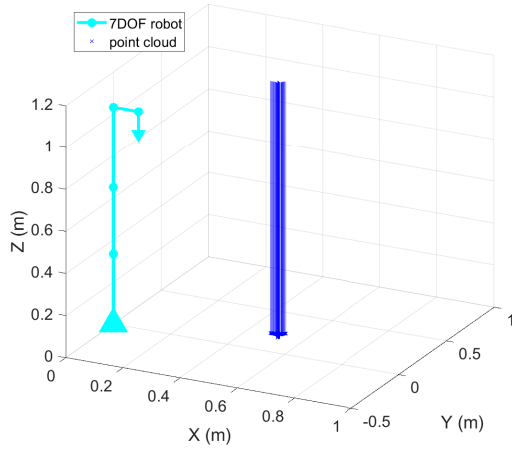
Experiment 1 demonstrates that the VF force can be appropriately rendered, and the robot EE may penetrate into the point cloud but the proxy point should never. The VF force can be rendered on both sides of the 2D square point cloud (*i.e.*, $y < 0.1$, and $y > 0.1$), which indicates that the VF algorithm is valid in 3D space. The results



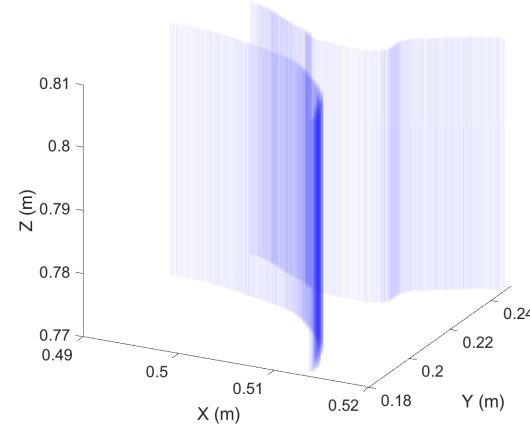
(a) Experiment 1 setup



(b) Experiment 1 point cloud



(c) Experiment 2 setup



(d) Experiment 2 point cloud

Figure 5.3: Setup and point cloud patterns for Experiment 1 and Experiment 2. Experiment 1 employs a set of point cloud with a 2D square shape ($N = 40401 = 201 \times 1 \times 201$), while Experiment 2 employs a set of point cloud with a hand-drawn curve shape ($N = 2204$). Note that the VF in Experiment 2 along z -axis is set as continuous and infinite by ignoring the z -axis coordinate during VF force rendering.

in Experiment 1 show that the VF algorithm in the physical experiment behaves the same as that in the simulations.

Experiment 2: Image-based scenario

In Experiment 2, the point-based VF algorithm is implemented and evaluated in a preplanned image scenario. Before the experiment, some preparation work is needed. First, a series of 2D CT images are acquired from a patient who has been diagnosed

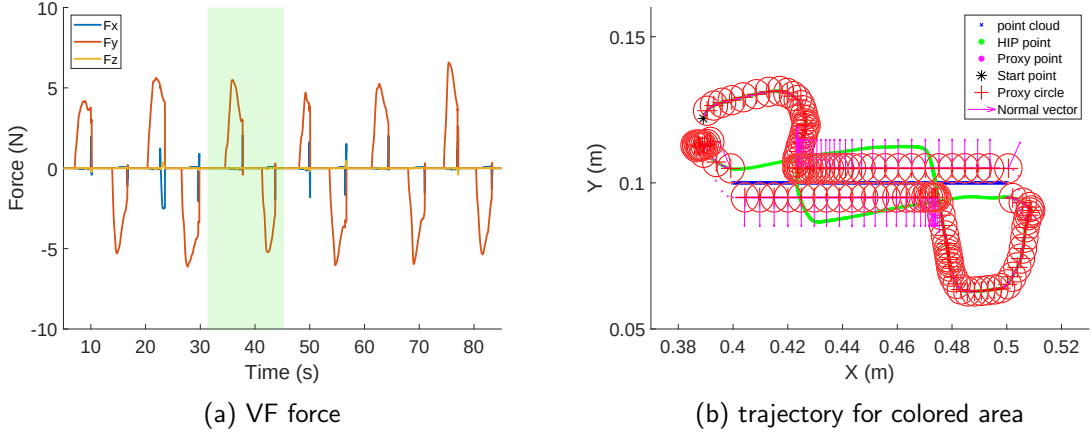


Figure 5.4: Experiment 1 of implementing VF on a 2D square point cloud with a size of $N = 40401 = 201 \times 1 \times 201$. The blue cross, green point, and magenta point represent the point cloud, the robot EE (the HIP), and the proxy, respectively. The proxy contacting region and the normal vector $\hat{n}$ are represented by red circle and magenta arrow, respectively.

with osteoarthritis and an elbow arthroscopic debridement surgery is required. Then, preplans are conducted on one image as shown in Figure 5.5a. In the preplanned image, a hand-drawn curve in red color is shown for planning to remove osteophytes. Nine markers (x1-x9) are marked in the image which will be used later for image-robot registration. Once the preplans are done, the hand-drawn curve is extracted and the corresponding binary image is shown in Figure 5.5b. Lastly, a set of point cloud representing the preplans is extracted from the binary image.

Now we start to do image-robot registration. Considering that the main purpose of this experiment is to evaluate the effectiveness of the VF algorithm, and for simplicity, a paper-printed 2D bone instead of a 3D physical bone is used in the registration. The paper-printed 2D bone is fixed on a horizontal desktop in the workspace of the Panda robot (see Figure 5.3c for illustration). The registration is done by using the ordinary least-squares (OLS) method [47] based on the nine markers (x1-x9) on both the paper and the digital image.

The point cloud of the hand-drawn curve extracted from the image consists of 2204 points with a density of 5.3 points/mm. During the experiment, the user moves the robot EE to probe the point cloud several rounds, and the corresponding VF force

is rendered and delivered to the human user as shown in Figure 5.6a, in which the green-colored area indicates one specific trial. The robot EE trajectory in the specific trial is represented by green dots as shown in Figure 5.6b. In the figure, the proxy and its contacting region (red circle and its center), and the normal vector $\hat{\mathbf{n}}$ (magenta arrow) are visualized in a frequency of 10 Hz. Note that the z -axis is ignored when rendering the VF force, which means that the generated VF along z -axis is continuous and infinitely long.

Experiment 2 demonstrates that the proposed VF algorithm is valid in the image-base scenario. The VF can be generated based on a hand-drawn curve from an image while the VF force can be appropriately rendered in 3D space.

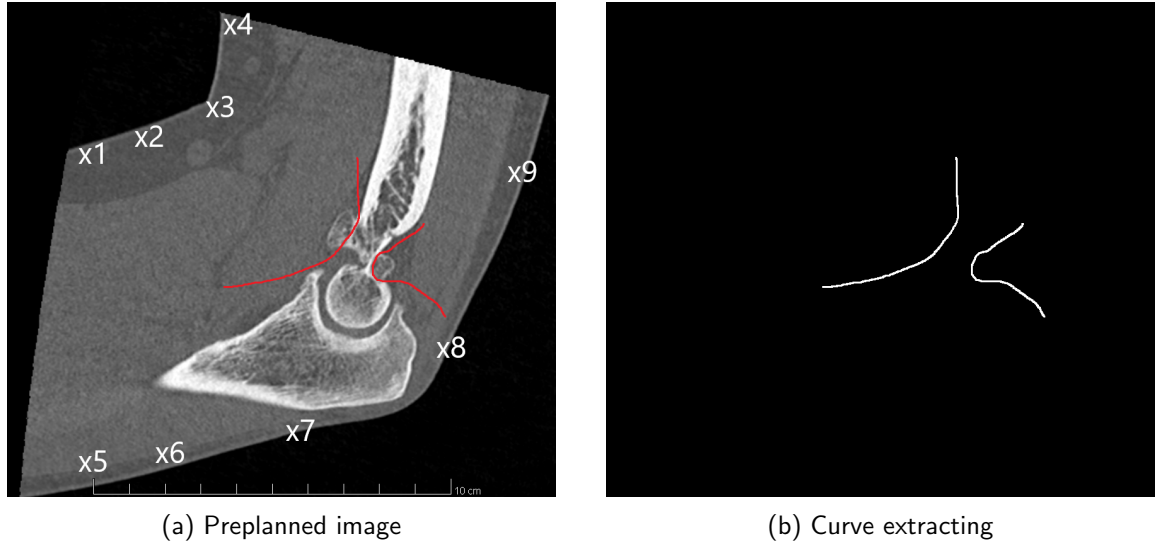


Figure 5.5: Preplanned 2D image of a patient with osteoarthritis diagnosed and elbow arthroscopic debridement surgery required. Preplanned image size 871×786 (width $\times$ height) pixels. The red curve is hand-drawn for planning to remove osteophytes. The markers x1 to x9 will be used to do an image-robot registration.

5.4 Discussions

In this chapter, an algorithm for generating virtual fixture (VF) directly from a set of point cloud data is presented. The effectiveness of the VF algorithm is evaluated by a series of simulations and experiments.

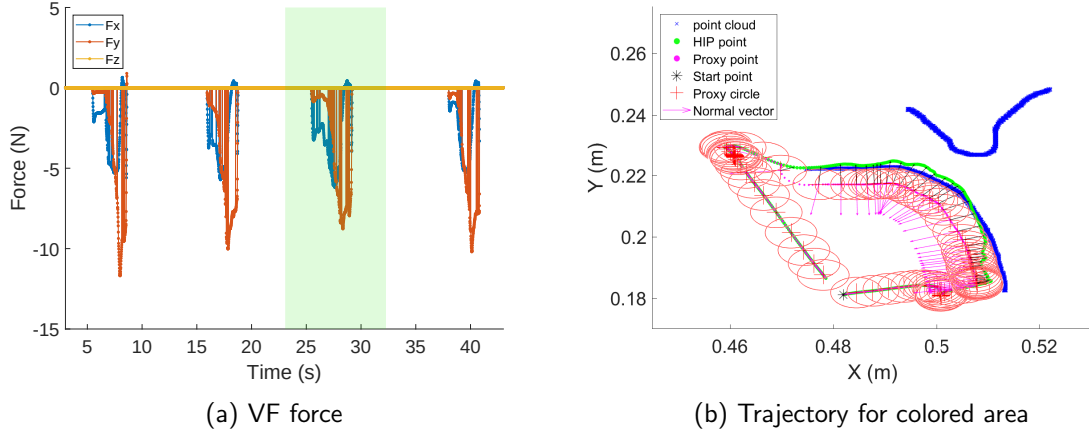


Figure 5.6: Experiment 2 of implementing VF on point cloud of a hand-drawn curve with a size of $N = 2204$. The blue cross, green point, and magenta point represent the point cloud, the robot EE (the HIP), and the proxy, respectively. The proxy contacting region and the normal vector $\hat{n}$ are represented by red circle and magenta arrow, respectively.

The simulations demonstrated that the VF algorithm works well on point clouds of various geometric entities, *i.e.*, a single point, a line segment, a 2D teapot, and a 2D hand-drawn curve. In Experiment 1, the results demonstrated that the VF algorithm works well in 3D space by employing a set of point cloud in the shape of a 2D square, while the user can feel the resistant force generated by the VF when touching either side of the square via robot EE. In Experiment 2, 3D VF is generated based on a hand-drawn curve extracted from a preplanned image, and the VF force is appropriately rendered. Both the simulations and the experiments verified the effectiveness of the VF algorithm. Particularly, the results of Experiment 2 showed the possibility to implement the VF algorithm in image-guided surgery.

The VF algorithm is used for static VF in this work. It should be noted that the algorithm is capable of serving for dynamic VF, *i.e.*, generating and updating VF in an online manner. This can be realized by online updating the point cloud dataset. This feature could be very useful for some surgical scenarios, such as bone burring during arthroscopic surgery, in which case the VF can online update itself based on the real-time shape of the target bone.

One advantage of the point-based VF algorithm is that the VF is generated directly

from points, the simplest geometric entity. Therefore, there is no need for 2D/3D surface/volume reconstruction before generating VF which can save a substantial amount of procedures and computations. More importantly, the point cloud can be in any shape.

A limitation of the work in this chapter is visualization. In our experiments, only the point cloud and the robot EE (*i.e.*, the HIP) is visualized on a monitor during the task. In future work, all key VF features (*e.g.*, proxy point, contacting region, normal vector) in 3D space will be visualized by using the augmented reality (AR) technique, which can better help surgeons to utilize the VF.

5.5 Conclusion

Virtual fixture (VF) plays an important role in robot-assisted surgeries. A variety of algorithms for generating VF have been developed for various surgical applications. However, generating VF for a free-style curve/surface, *e.g.*, a hand-drawn spline, is still a challenging problem due to the fact that an accurate mathematical function cannot always be found for such types of curves and surfaces. In this chapter, a point-based VF algorithm is presented which allows to generate VF directly from the point cloud data. The effectiveness of the algorithm is demonstrated by both simulations and experiments. An experiment in an image-based scenario verified the capability of the algorithm to generate VF based on a hand-drawn curve in an image.

The point-based VF algorithm is promising to be applied in various surgical scenarios in robot-assisted surgery and image-guided surgery, as long as a set of point cloud of the target object can be obtained. In future work, we will implement the VF algorithm into a realistic arthroscopic surgery scenario by using 3D physical bones and develop accurate registration methods for robot-image-bone registration.

Algorithm 1: Main algorithm

Data: A set of point cloud ($\mathbf{L}_{\text{pcld}}$) is predefined.

Result: A step vector ($\hat{\mathbf{s}}$) is calculated in each servo loop according to different states of the proxy ($state_{\text{proxy}}$), which can ensure that the proxy point ($\mathbf{P}_{\text{proxy}}$) always remains outside of the point cloud and never penetrates into it.

```
%(Initialization) ;
 $\mathbf{P}_{\text{HIP}} \leftarrow [-1, 0, 0]$  (for simulations) ;
 $\mathbf{P}_{\text{proxy}} \leftarrow \mathbf{P}_{\text{HIP}}$  ;
 $r_c \leftarrow 5 * 10^{-3}$  ;
 $r_1 \leftarrow r_c - 0.01 * 10^{-3}$  ;
 $r_2 \leftarrow r_c + 0.01 * 10^{-3}$  ;
 $r_3 \leftarrow 2 * r_c$  ;
 $r_v \leftarrow 0$  ;
while in a servo loop do
    %(Update HIP position) ;
     $\mathbf{P}_{\text{HIP}} \leftarrow$  real-time robot EE position ;
    %(To categorize point cloud) ;
    for each point  $\mathbf{p}_i$  in cloud  $\mathbf{L}_{\text{pcld}}$  do
         $de = \|\mathbf{p}_i - \mathbf{P}_{\text{proxy}}\|_2$  ;
         $\mathbf{L}_{\text{Entrenched}} \leftarrow$  who has  $de < r_1 + r_v$  ;
         $\mathbf{L}_{\text{InContact}} \leftarrow$  who has  $r_1 + r_v \leq de \leq r_2 + r_v$  ;
         $\mathbf{L}_{\text{FreeMotion}} \leftarrow$  who has  $r_2 + r_v < de < r_3 + r_v$  ;
         $\mathbf{L}_{\text{OutNeighbor}} \leftarrow$  who has  $de \geq r_3 + r_v$  ;
    end
    Note: Here the for-loop is only for illustration. ;
    For efficiency, the point cloud needs to be treated as a whole matrix when doing the categorization. ;
    %(To determine proxy state  $state_{\text{proxy}}$ ) ;
    if  $\mathbf{L}_{\text{Entrenched}} \neq \text{null}$  then
         $state_{\text{proxy}} = 2$  (entrenched) ;
    else
        if  $\mathbf{L}_{\text{InContact}} \neq \text{null}$  then
             $state_{\text{proxy}} = 1$  (in contact) ;
        else
            if  $\mathbf{L}_{\text{FreeMotion}} \neq \text{null}$  then
                 $state_{\text{proxy}} = 0$  (free motion) ;
            else
                 $state_{\text{proxy}} = -1$  (no neighbor) ;
            end
        end
    end
    %(To determine normal vector  $\hat{\mathbf{n}}$ ) ;
    if  $state_{\text{proxy}} = 0/1/2$  then
        find the normal vector  $\hat{\mathbf{n}}$  via (5.1) and (5.2) ;
    end
    %(To determine proxy movement step  $\hat{\mathbf{s}}$ ) ;
    if  $state_{\text{proxy}} = -1/0/1/2$  then
        To determine  $\hat{\mathbf{s}}$  from proxy algorithms (algorithm 2, algorithm 3, algorithm 4) ;
        ( $\hat{\mathbf{s}}$  determined) ;
    end
    %(Update proxy position);
     $\mathbf{P}_{\text{proxy}} = \mathbf{P}_{\text{proxy}} + \hat{\mathbf{s}}$  ;
end
```

Algorithm 2: Proxy movement step ($state_{\text{proxy}} = 0$)

Result: To determine step vector $\hat{\mathbf{s}}$ for proxy state of in free motion ($state_{\text{proxy}} = 0$).

```
% (When proxy state is in free motion) ;
if  $state_{\text{proxy}} = 0$  then
    if  $\mathbf{P}_{\text{proxy}} = \mathbf{P}_{\text{HIP}}$  then
         $\hat{\mathbf{s}} = [0, 0, 0]$  ;
    else
        if  $\mathbf{P}_{\text{proxy}} \neq \mathbf{P}_{\text{HIP}}$  then
             $d = \min(d0_i)$ , if  $\min(d0_i) < \|\vec{u}\|$  ;
             $d = \|\vec{u}\|$ , if  $\min(d0_i) \geq \|\vec{u}\|$  ;
             $\hat{\mathbf{s}} = d * \frac{\vec{u}}{\|\vec{u}\|}$  ;
        end
        % (To do a special case-1 check on  $\hat{\mathbf{s}}$ ) ;
        if  $d = 0$  & HIP inside VF then
            To determine  $d$  via projection  $\vec{u}_p$  (5.5) ;
             $\hat{\mathbf{s}} = d * \frac{\vec{u}_p}{\|\vec{u}_p\|}$  ;
        end
        % (To do a special case-2 check on  $\hat{\mathbf{s}}$ ) ;
        if HIP is moving away from point cloud then
             $\hat{\mathbf{s}} = \mathbf{P}_{\text{HIP}} - \mathbf{P}_{\text{proxy}}$  ;
        end
        ( $\hat{\mathbf{s}}$  determined) ;
    end
end
end
```

Algorithm 3: Proxy movement step ($state_{\text{proxy}} = 1$)

Result: To determine step vector $\hat{\mathbf{s}}$ for proxy state of in contact
($state_{\text{proxy}} = 1$).

```
%(When proxy state is in contact) ;
if  $state_{\text{proxy}} = 1$  then
    HIP inside/outside VF  $\leftarrow$  angle between  $\hat{\mathbf{n}}$  &  $\vec{u}$  ;
    projection vector  $\vec{u}_p \leftarrow$  from  $\hat{\mathbf{n}}$  &  $\vec{u}$  ;
    if  $\|\vec{u}_p\| \leq r_1$  then
        |  $d = \xi \|\vec{u}_p\|$  ;
    else
        | if  $\|\vec{u}_p\| > r_1$  then
        | |  $d = \xi r_1$  ;
        | end
    end
    end
    ( $d$  determined) ;
    if HIP inside VF then
        |  $\hat{\mathbf{s}} = d * \frac{\vec{u}_p}{\|\vec{u}_p\|}$  ;
    else
        | if HIP outside VF then
        | |  $\hat{\mathbf{s}} = d * \frac{\vec{u}}{\|\vec{u}\|}$  ;
        | end
    end
    end
    ( $\hat{\mathbf{s}}$  determined) ;
end
```

Algorithm 4: Proxy movement step ($state_{\text{proxy}} = -1/2$)

Result: To determine step vector $\hat{\mathbf{s}}$ for proxy state of entrenched
($state_{\text{proxy}} = 2$) and no neighbour ($state_{\text{proxy}} = -1$).

```
%(When proxy state is entrenched) ;
if  $state_{\text{proxy}} = 2$  then
     $d = \max(d_{2_i})$  from (5.7) ;
     $\hat{\mathbf{s}} = d * \hat{\mathbf{n}}$  ;
    ( $\hat{\mathbf{s}}$  determined) ;
end
%(When proxy state is no neighbour) ;
if  $state_{\text{proxy}} = -1$  then
     $\hat{\mathbf{s}} = \mathbf{P}_{\text{HIP}} - \mathbf{P}_{\text{proxy}}$  ;
    ( $\hat{\mathbf{s}}$  determined) ;
end
```

Chapter 6

A Prototype of a Two-Arm Robotic System for Arthroscopic Surgery

In this chapter, we present a prototype of a two-arm robot-assisted arthroscopic surgical system by assembling our previous work together. The left-arm robot is used as a robot-assisted arthroscope holder, and it can hold the arthroscope still at a fixed pose, while also allowing the operator to move it via a pedal switch whenever needed. The left-arm robot is implemented with an impedance controller and a gravity iterative learning (Git) scheme based on **Chapter 4**, where the former can provide compliant robot behavior thus ensuring a safe human-robot interaction, while the latter can accurately learn and compensate for gravity. The right-arm robot is used as a robot-assisted surgical tool providing VF assistance and haptic feedback during the surgery. The right-arm robot is implemented with a point-based virtual fixture (VF) algorithm from **Chapter 5**, which can generate VF directly from a point cloud with any shape, render haptic force feedback, and deliver it to the operator. Furthermore, the VF, the bone, and the surgical bur with its real-time position are visualized in a 3D digital environment as additional visual feedback for the operator. A series of experiments are conducted to evaluate the effectiveness of the prototype.

¹A version of this chapter has been submitted as: Teng Li, Armin Badre, and Mahdi Tavakoli, “*Robotic Assistance and Haptic Feedback in Arthroscopic Procedures: Design and Preliminary Evaluation of a Two-Arm System*”, Journal of Medical Robotics Research (JMRR). [Under Review]

6.1 Background

Robotic systems and techniques for orthopedic surgery have been developed and evolved for several decades [74]. Nowadays, robot-assisted surgical systems have predominated over many orthopedic surgeries, such as total hip arthroplasty (THA), total knee arthroplasty (TKA), unicompartmental knee arthroplasty (UKA), and spine surgery [74]. On the other hand, robot-assisted minimally invasive surgery (MIS) has received more and more attention in orthopedics and beyond due to its advantages of a faster recovery rate and decreased pain [43]. However, most of the existing robotic systems in orthopedics have more focus on open surgery than MIS like arthroscopic surgery [74, 43].

Elbow arthroscopy is a common arthroscopic surgery in orthopedics that is commonly used for the management of elbow arthritis, stiffness, tendinosis, fractures, and instability in a minimally invasive fashion [13]. During traditional elbow arthroscopy, the surgeon needs to hold an arthroscope with one hand while conducting the surgery with the other hand under the arthroscope view. The arthroscope view may need to be adjusted many times during the surgery in order to observe the surgical site from different perspectives or change to another surgical site. Holding the arthroscope still is important for the surgeon to conduct the surgery smoothly since the arthroscope view is the main visual feedback the surgeon relies on to visually observe and locate the surgical site, but this could make the surgeon easy-to-fatigue and high cognitive load thus have adverse effect on the surgical performance. This arouses the necessity to develop a robot-assisted system where the robot can hold the arthroscope still for the surgeon, which can free the surgeon's hand for other more important tasks, *e.g.*, replacing the surgical bur with another one in a different shape.

To build a robotic assistant as an arthroscope holder, some requirements need to be met [75]. First, The robot can hold the arthroscope still at a fixed position while rejecting all possible disturbances (*e.g.*, external disturbances delivered to the

arthroscope via contact with the patient’s body during surgery). This will ensure the surgeon always receives stable visual feedback even when some occasional disturbances are delivered to the arthroscope. Second, when the surgeon needs to move the arthroscope to adjust the scope view perspective or to a new surgical site, the robot should enable physical human-robot interaction (*pHRI*) and allow the arthroscope to be moved around freely. The main concerns in building such a robot-assisted arthroscope holder are the dynamic model uncertainties and external disturbances, which could largely affect the robot’s performance accuracy and even stability if they are not appropriately compensated for in the robot dynamics. More specifically, incomplete gravity compensation can be the main issue since heavy external surgical tools (*e.g.*, the arthroscope) will be attached to the robot end-effector (EE).

In the two-arm system proposed in this work, the left-arm robot will be designed as a robot-assisted arthroscope holder to satisfy the requirements described above. It will mainly tackle the problem of disturbance estimation and gravity compensation while ensuring a robust and safe human-robot interaction. To this end, we have explored different approaches including disturbance observer, neural network (NN), and gravity iterative learning (Git) scheme in our previous work[75, 76, 79].

Disturbance observer is a promising way to estimate and compensate for dynamic uncertainties including gravity. In our previous work [75], we have shown that by integrating impedance control and nonlinear disturbance observer, an accurate impedance control can be achieved. In that work, the disturbance observer can accurately estimate and compensate for the lumped uncertainties including incomplete gravity compensation. However, the problem is that the nonlinear disturbance observer (NDOB) as well as any other types of observers, such as generalized momentum observer (GMO), joint velocity observer (JVOB), extended state observer (ESO), and disturbance Kalman filter (DKF) method, always estimate all the uncertainties as a lumped term and is not able to separate any one component out [75, 78]. Moreover, the observer will refuse human-robot interaction since human-applied force will

also be taken as a part of the lumped disturbances thus being rejected [75]. Then, we tried to use an NN model to learn and separate a specific component from the uncertainties [76]. Although it works well, it requires tremendous data and time to train the NN model before use. To solve the problem more efficiently, we developed a gravity iterative learning (Git) scheme in [79], especially for gravity compensation since the gravity of the external surgical tools attached to the robot EE is the main issue in our application scenario. With the Git scheme, the uncompensated gravity can be accurately learned and compensated for in an online manner.

In the left-arm robot-assisted arthroscope holder, an integrated framework of integrating an impedance controller and the Git scheme will be implemented. The impedance controller will ensure compliant robot behavior thus a robust and safe human-robot interaction. The Git scheme will iteratively learn and compensate for the gravity in the robot dynamics thus ensuring an accurate and stable robot control system, and also enable human-robot interaction via a pedal switch when necessary.

On the other hand, it could be helpful for surgeons by providing them with additional haptic feedback via virtual fixture (VF). In the field of robot-assisted surgery, VF has been widely used due to many potential benefits, such as reducing the surgeon’s cognitive load [150], improving surgeon’s surgical performance [48], and making the surgical outcome more accurate and safe. Park *et al.* conducted a preliminary test on VF in a blunt dissection task [99]. Their results indicated faster and more precise task performance with the VF-assisted method than the conventional free-hand method. Ryden *et al.* developed a method to generate VF directly from point cloud to protect the beating heart during surgery [112]. They improved their method further in [111, 96].

The haptic VF has been playing a vital role during various surgical procedures in robot-assisted surgery, such as suturing [86], knot tying [150], dissection [121], either assisting in moving the surgical tool along a trajectory or preventing it from entering a specific area for protecting the objects inside (*e.g.*, beating heart or nerve) [112,

92]. Many research works have proved that VF with haptic feedback can provide effective help to improve performance in surgical tasks. Johansson *et al.* evaluated the feasibility and repeatability of using haptic VF to guide fibula osteotomies in mandible reconstruction surgery [62]. As a further step, Cheng *et al.* proposed a robotic assistant incorporating augmented reality (AR) visualization and haptic VF for fibula osteotomies in mandible reconstruction surgery [20]. By comparing several methods on the same fibula osteotomy task, their results showed that with the help of AR and VF, the task precision can be improved.

In orthopedics, surgical plans are usually made based on preoperative images of a patient. For example, determining the location and amount of osteophytes to be debrided or the location of critical neurovascular structures to be avoided. In traditional arthroscopic surgery, as mentioned earlier, the surgeon heavily relies on the visual feedback from the arthroscope view to perform the surgical procedures at hand. Also, the surgeon may need to mental image the surgical site and conduct the surgical procedure (*e.g.*, removing the osteophytes) by intuition and experience since the arthroscope view is largely localized and lacks of depth information. A robot-assisted surgical tool (*e.g.*, a surgical bur) with haptic assistance can be designed to relieve this problem, *e.g.*, to help the surgeon reduce the mental load by providing additional visual feedback and haptic feedback. To this end, some curves can be drawn in the preoperative images to mark out boundaries of removing osteophytes, or of protecting nerves inside. Then, with a robotic system, the hand-drawn curves or the patient's bone in the preoperative images can be used to generate VF which can assist in removing osteophytes or protecting the nerves by providing haptic feedback. However, the hand-drawn curves or the bone are usually in irregular shapes which may not be able to be presented mathematically by equations that are often required by most existing VF generating algorithms. To solve this problem, we developed a point-based VF generation algorithm in [77], which allows us to generate VF directly from point clouds in any shape.

Therefore, the right-arm robot in the proposed two-arm system will be designed to be a robot-assisted surgical tool with haptic feedback from VF, where the VF can be generated directly from point clouds in any shape, *e.g.*, a hand-drawn curve, or a patient-specific bone model. Furthermore, augmented 3D visual feedback will be provided to the surgeon to indicate the generated VF and the real-time location of the surgical tool in a more global view in addition to the localized arthroscope view.

In summary, by integrating our previous work together [75, 77, 79], a prototype of a two-arm robot-assisted arthroscopic surgical system is designed and experimentally evaluated in this chapter. The left-arm robot, a robot-assisted arthroscope holder, is implemented with an integrated framework of impedance control and Git scheme developed in [79], which can ensure a safe human-robot interaction while accurately learning for gravity compensation. The left-arm robot can help to hold the arthroscope still at a designated pose and also allow the operator to move it freely via a pedal switch whenever necessary. The right-arm robot, a robot-assisted surgical tool, is implemented with a point-based VF generation algorithm developed in [77], which can provide VF assistance with haptic feedback to assist the operator in performing surgical operations. Moreover, the VF, the surgical tool, and the force feedback values are visualized in a 3D digital environment to provide the operator with additional visual feedback during the surgery. A series of experiments are conducted to evaluate the effectiveness of the prototype. The main contributions of this chapter are described as the following,

- A two-arm robot-assisted system (6DOF + 6DOF) is designed and assembled for arthroscopic surgery, while the control systems consist of techniques developed in our previous work.
- The effectiveness of each arm is experimentally evaluated and verified, respectively.

This chapter builds upon our prior work by not only refining the individual tech-

nologies but also demonstrating their synergistic operation within a dual-arm robotic system. The integration of a Git scheme for the arthroscope-holding arm and a point-based VF algorithm for the tool-operating arm presents an advancement that addresses a gap in arthroscopic surgery: the need for a comprehensive system that enhances the surgeon’s dexterity and cognitive focus. By unifying these technologies within a single and cohesive framework, we provide a solution that mitigates the cognitive load on surgeons, offering both enhanced stability for the arthroscope and intuitive haptic guidance for the surgical tool.

6.2 Methods

Robot dynamics governs the motion of a robot in response to external forces or disturbances. Without properly handling the external disturbances (*e.g.*, the mass of the surgical tool attached to robot EE) in robot dynamics, the robot may perform inaccurately or even have unstable or dangerous behavior. In this section, we will first introduce the robot dynamics and disturbances, then we introduce impedance control which can provide compliant robot behavior, and then we introduce a Git scheme that can accurately compensate for external disturbances, especially gravity. At the end of this section, a prototype of the proposed two-arm robot-assisted arthroscopic surgical system will be presented, as well as the control block diagram.

6.2.1 Left-Arm: Robot dynamics and disturbances

A general dynamic model for an n -degree-of-freedom (DOF) rigid robot with revolute joints [41] can be given by

$$\underbrace{\mathbf{M}(\mathbf{q})}_{\hat{\mathbf{M}}+\Delta\mathbf{M}} \ddot{\mathbf{q}} + \underbrace{\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})}_{\hat{\mathbf{S}}+\Delta\mathbf{S}} \dot{\mathbf{q}} + \underbrace{\mathbf{G}(\mathbf{q})}_{\hat{\mathbf{G}}+\Delta\mathbf{G}} + \boldsymbol{\tau}_{\text{fric}}(\dot{\mathbf{q}}) = \boldsymbol{\tau} + \underbrace{\boldsymbol{\tau}_{\text{ext}}}_{\mathbf{J}^T \mathbf{F}_{\text{ext}}} \quad (6.1)$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ are the joint position, velocity, and acceleration, respectively, $\mathbf{M} \in \mathbb{R}^{n \times n}$ denotes the inherent inertia matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes a matrix of the Coriolis and centrifugal forces, $\mathbf{G} \in \mathbb{R}^n$ represents the gravity vector. $\hat{\mathbf{M}}, \hat{\mathbf{S}}, \hat{\mathbf{G}}$ represent

users' model estimates, while $\Delta\mathbf{M}, \Delta\mathbf{S}, \Delta\mathbf{G}$ are the corresponding estimate errors. $\boldsymbol{\tau}_{\text{fric}} \in \mathbb{R}^n$ is joint friction, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the commanded joint torque vector, $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^n$ is the torque caused by external force, $\mathbf{F}_{\text{ext}} \in \mathbb{R}^6$ is the external force in Cartesian space, and $\mathbf{J} \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix.

By collecting all the disturbances together, the dynamic model (6.1) of a robot can be re-written as

$$\hat{\mathbf{M}}\ddot{\mathbf{q}} + \hat{\mathbf{S}}\dot{\mathbf{q}} + \hat{\mathbf{G}} = \boldsymbol{\tau} + \underbrace{\boldsymbol{\tau}_{\text{ext}} - [\boldsymbol{\tau}_{\text{fric}} + (\Delta\mathbf{M}\ddot{\mathbf{q}} + \Delta\mathbf{S}\dot{\mathbf{q}} + \Delta\mathbf{G})]}_{\boldsymbol{\tau}_{\text{dist}}} \quad (6.2)$$

where $\boldsymbol{\tau}_{\text{dist}}$ denotes the lumped uncertainties containing the model error $(\Delta\mathbf{M}\ddot{\mathbf{q}} + \Delta\mathbf{S}\dot{\mathbf{q}} + \Delta\mathbf{G})$, the joint friction $\boldsymbol{\tau}_{\text{fric}}$, and the external disturbances $\boldsymbol{\tau}_{\text{ext}}$.

In our target application, *i.e.*, a robot-assisted arthroscopic surgical system, setpoint regulation and physical human-robot interaction (*pHRI*) are the two main scenarios we are considering. In the steady state of setpoint regulation (*i.e.*, $\mathbf{q} = \mathbf{constant}$, $\dot{\mathbf{q}} = \ddot{\mathbf{q}} = \mathbf{0}$), it will have, (a) joint friction disappeared, *i.e.*, $\boldsymbol{\tau}_{\text{fric}} = \mathbf{0}$; (b) model error $(\Delta\mathbf{M}\ddot{\mathbf{q}} + \Delta\mathbf{S}\dot{\mathbf{q}} + \Delta\mathbf{G})$ will be reduced to only gravity compensation error $(\Delta\mathbf{G})$; (c) only the gravity of the external constant payloads exists for external disturbances (*e.g.*, the surgical tools attached to the robot EE). By applying these conditions, the dynamic model (6.1) will become (6.3).

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{S}\dot{\mathbf{q}} + \hat{\mathbf{G}} = \boldsymbol{\tau} + \underbrace{\mathbf{J}^T \mathbf{F}_{\text{ext}} - \Delta\mathbf{G}}_{\boldsymbol{\tau}_{\text{dist}}} \quad (6.3)$$

The model (6.3) can be expressed in Cartesian space as

$$\mathbf{M}_x \ddot{\mathbf{x}} + \mathbf{S}_x \dot{\mathbf{x}} + \mathbf{G}_x = \mathbf{J}^{-T} \boldsymbol{\tau} + \mathbf{F}_{\text{ext}} \quad (6.4)$$

where $\mathbf{M}_x, \mathbf{S}_x, \mathbf{G}_x$ have

$$\begin{cases} \mathbf{M}_x = \mathbf{J}^{-T} \mathbf{M} \mathbf{J}^{-1} \\ \mathbf{S}_x = \mathbf{J}^{-T} \mathbf{S} \mathbf{J}^{-1} - \mathbf{M}_x \dot{\mathbf{J}} \mathbf{J}^{-1} \\ \mathbf{G}_x = \mathbf{J}^{-T} \mathbf{G} \end{cases} \quad (6.5)$$

where $\mathbf{M}_x, \mathbf{S}_x, \mathbf{G}_x$ are the $\mathbf{M}, \mathbf{S}, \mathbf{G}$ expressed in Cartesian space, respectively.

6.2.2 Left-Arm: Impedance control

A desired impedance model [129, 133, 75] for robot-environment interaction can be expressed as

$$\mathbf{F}_{\text{imp}} = \mathbf{M}_{\text{m}}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_{\text{d}}) + (\mathbf{S}_{\text{x}} + \mathbf{D}_{\text{m}})(\dot{\mathbf{x}} - \dot{\mathbf{x}}_{\text{d}}) + \mathbf{K}_{\text{m}}(\mathbf{x} - \mathbf{x}_{\text{d}}) \quad (6.6)$$

where $\mathbf{M}_{\text{m}}, \mathbf{D}_{\text{m}}, \mathbf{K}_{\text{m}}$ are user-designed matrices for inertia, damping, and stiffness, respectively. Note that $\mathbf{x}_{\text{d}}, \dot{\mathbf{x}}_{\text{d}}, \ddot{\mathbf{x}}_{\text{d}}$ are the desired position, velocity, and acceleration, respectively in Cartesian space, while $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$ are the actual ones. $\mathbf{F}_{\text{imp}}$ is the interaction force between the robot and the environment.

To avoid the measurement of external forces, the designed inertia matrix will be set as the inherent inertia matrix of the robot, *i.e.*, $\mathbf{M}_{\text{m}} = \mathbf{M}_{\text{x}}$. Then, by substituting (6.6) into (6.4) with $\mathbf{F}_{\text{ext}} = \mathbf{F}_{\text{imp}}$, the impedance control law can be given by [75]

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{J}^{-1}(\ddot{\mathbf{x}}_{\text{d}} - \dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{x}}_{\text{d}}) + \mathbf{S}\mathbf{J}^{-1}\dot{\mathbf{x}}_{\text{d}} + \mathbf{G} + \mathbf{J}^{\text{T}}[\mathbf{D}_{\text{m}}(\dot{\mathbf{x}}_{\text{d}} - \dot{\mathbf{x}}) + \mathbf{K}_{\text{m}}(\mathbf{x}_{\text{d}} - \mathbf{x})] \quad (6.7)$$

For set-point regulation, *i.e.*, let robot EE stay at a fixed point, it will have $\dot{\mathbf{x}}_{\text{d}} = \mathbf{0}$, $\ddot{\mathbf{x}}_{\text{d}} = \mathbf{0}$. Then, the impedance control law (6.7) will be simplified and reduced to (6.8), which is also known as task-space proportional-derivative (PD) controller with gravity compensation.

$$\boldsymbol{\tau} = \mathbf{J}^{\text{T}}[\mathbf{K}_{\text{m}}(\mathbf{x}_{\text{d}} - \mathbf{x}) - \mathbf{D}_{\text{m}}\dot{\mathbf{x}}] + \mathbf{G} \quad (6.8)$$

With an impedance controller, the robot can behave with compliance and robustness. By tuning the parameters in the impedance model ($\mathbf{K}_{\text{m}}$ and $\mathbf{D}_{\text{m}}$), the robot can be configured to be “soft” or “stiff”. And with a “soft” robot behavior, a safe human-robot interaction can be ensured.

6.2.3 Left-Arm: Git scheme

Since gravity compensation is the main problem in our target application, a gravity iterative learning (Git) scheme is used to solve this problem [26, 79]. In our previous work [79], we developed a Git scheme that can accurately learn and compensate for

gravity. A brief introduction to the Git scheme will be presented in this subsection, while for more details on the Git scheme please refer to [79]. A Cartesian-space impedance control law (at the i -th iteration, $i = 1, 2, \dots$) integrating with the Git scheme for gravity compensation can be expressed by

$$\boldsymbol{\tau}_i = \mathbf{M}\mathbf{J}^{-1}(\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}\mathbf{J}^{-1}\dot{\mathbf{x}}_d) + \mathbf{S}\mathbf{J}^{-1}\dot{\mathbf{x}}_d + \mathbf{J}^T[\mathbf{D}_m(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x})] + \mathbf{J}^T\mathbf{u}_{i-1} \quad (6.9)$$

where $\mathbf{J}^T\mathbf{u}_{i-1}$ is an iterative learning term for gravity compensation instead of a gravity term $\mathbf{G}$. For setpoint regulation, it will be reduced to be

$$\boldsymbol{\tau}_i = \mathbf{J}^T[\gamma\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \mathbf{J}^T\mathbf{u}_{i-1} \quad (6.10)$$

The update law for $\mathbf{u}_i$ can be given by

$$\mathbf{u}_i = \gamma\eta\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) + \mathbf{u}_{i-1} \quad (6.11)$$

where $\mathbf{u}_i$ is the iterative learning result at the i -th iteration ($i = 1, 2, \dots$), setting $\mathbf{u}_0 = \mathbf{0}$ for initialization, γ is a scalar gain, and η is the learning rate with steady-state scaling strategy to enable the iterative learning term updates itself in each sampling loop [79]. The scalar gain γ was mainly used for convergence analysis when the Git scheme was developed in [79], and it is usually set as $\gamma = 1$. According to [79], the learning rate η will first be scaled down from $\eta = 1$ to $\eta = 0.001$ by integrating the steady-state scaling strategy based on the sampling loop (here the sampling loop in our work is 0.001 s), then fine-tuned accordingly.

It is noteworthy that the dynamic model of the specific robot we employed in this work is not available, while in our target surgical scenario, only setpoint regulation and human-robot interaction are needed for the left-arm robot as an arthroscope holder. Therefore, the impedance controller Eqn. (6.8) is sufficient to meet our requirement in this work, which can also avoid involving the dynamic parameters ($\mathbf{M}$ and $\mathbf{S}$). Hence, the integrated control law Eqn. (6.10) is implemented in the left-arm robot.

The major advantages of employing the Git scheme in this work include (1) its simple and compact formulation, (2) its lack of any need for robot dynamics, and (3) its lack of any need for a priori knowledge about the external payloads [79]. The Git scheme will be implemented in the left-arm robot serving as a robot-assisted arthroscope holder. It will address the challenge of maintaining steady tool positioning, or in other words, external disturbances, *i.e.*, uncompensated gravity caused by the mass of the surgical tool attached to the robot EE. Maintaining steady tool positioning, *i.e.*, holding the arthroscope still, is a crucial factor in avoiding surgical errors during delicate arthroscopic procedures, and the Git scheme will help to achieve that.

Moreover, the integrated framework of impedance control and the Git scheme allows human-robot interaction to be enabled via a pedal switch whenever necessary, thus the arthroscope can be moved around to a new arthroscopic view or a new surgical site. It is worth noting that, as illustrated in Figure 6.2, the setpoint regulation mode (Interaction Disabled) and the *p*HRI mode (Interaction Enabled) are two interchange behaviors rather than simultaneous. The two modes can be seamlessly switched from one to another in an online manner via a pedal, and the pedal actions are independent of the impedance gains ($\mathbf{K}_m$, $\mathbf{D}_m$), *i.e.* the impedance gains remain unchanged in both modes.

6.2.4 Right-Arm: Point-based VF algorithm

To render haptic feedback, a virtual fixture (VF) generation algorithm is required. Most of the existing VF algorithms can deal with the target object with a regular shape by finding out the mathematical representation, but it could be a challenge for the objects with irregular shapes, *e.g.*, a patient-specific bone model, or a hand-drawn curve/surface in a preplan image. Although it still can be processed with some approaches like the god-object algorithm by reconstructing a triangle-meshed model [155], it could be complicated and time-consuming.

In our previous work [77], we developed a point-based VF generation algorithm

that allows us to generate VF directly from point clouds in any regular or irregular shape as long as a set of point clouds of the object can be obtained. The point-based VF algorithm consists of one main algorithm and three embedded algorithms for moving the proxy in different conditions. The effectiveness of the VF algorithm has been evaluated by a series of simulations and experiments. The VF algorithm has also been evaluated successfully in an image-based scenario where a hand-drawn curve in an irregular shape was extracted from an image as a set of point clouds. The VF force is rendered by a simple spring model in the algorithm, and delivered to the operator by the robot as haptic feedback. The details of the point-based VF algorithm are available in [77].

Therefore, the point-based VF generation algorithm will be employed in the right-arm robot. By implementing the point-based VF generation algorithm, VF assistance can be generated and VF force can be rendered. The VF force can assist the surgeon in conducting surgical operations. For example, a VF surface can be set at the bottom of the osteophytes as a boundary, and the surgeon will receive haptic feedback once the surgical tooltip is in contact with the VF surface. The VF assistance and haptic feedback allow for natural hand movements while facilitating precise maneuvers in tight joint spaces.

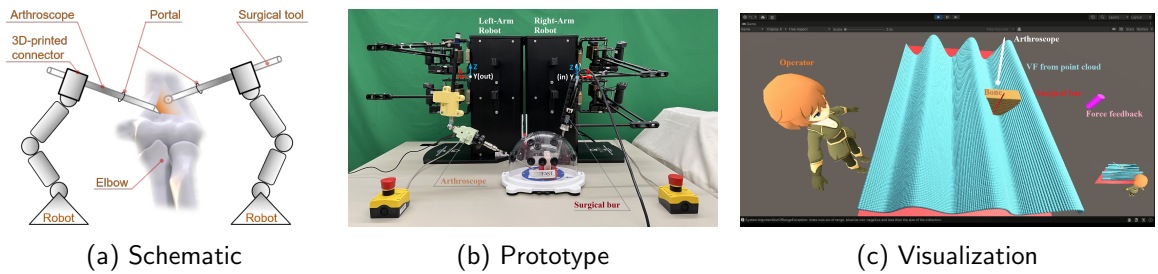


Figure 6.1: A schematic diagram of the two-arm system in arthroscopic surgery, a prototype of the proposed two-arm robot-assisted arthroscopic surgical system and visualization in Unity. A modified FAST (fundamentals of arthroscopic surgery training) simulator is used as the experimental platform for an arthroscopic surgery mockup.

6.2.5 Two-Arm System: Prototype and control diagram

A schematic diagram of robot-assisted arthroscopic surgery is illustrated in Figure 6.1a. The schematic diagram serves as a visual summary of the two-arm robotic system's interaction within the surgical environment. Specifically, it illustrates the spatial relationship between the arthroscope, the surgical tool, and the target anatomy (*e.g.*, the elbow). A physical prototype of a two-arm robot-assisted system for arthroscopic surgery is developed as shown in Figure 6.1b and Figure 6.1c, where Figure 6.1b shows the hardware of the prototype while Figure 6.1c shows a screenshot of the visualization in Unity. The control block diagram of the prototype system is illustrated in Figure 6.2.

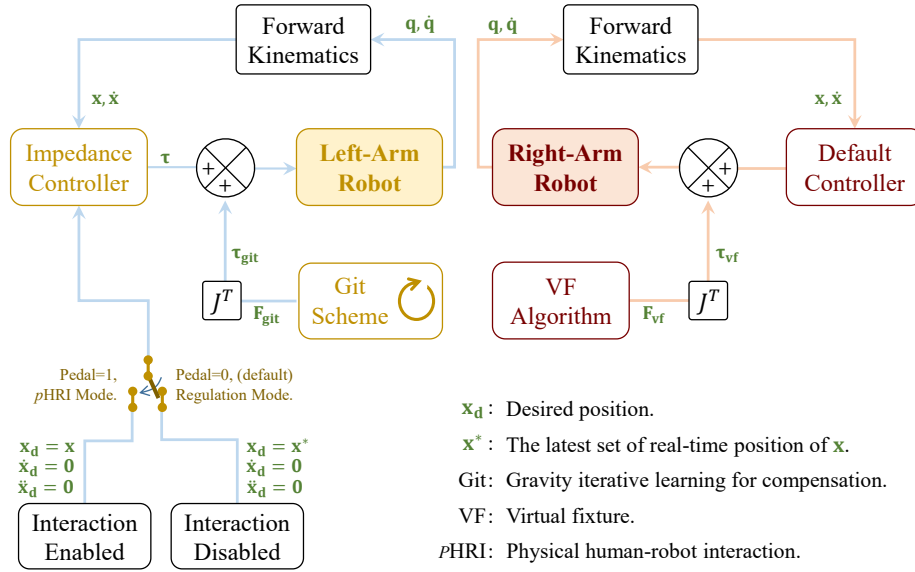


Figure 6.2: Control block diagram for the prototype of the two-arm robot-assisted arthroscopic surgical system. When the pedal is pressed ($x_d = x$), the position-dependent terms in the impedance controller and the Git update law vanish, whereas the learned result in the Git update law will remain valid, meaning that the pHRI mode is activated and interaction is enabled, and now the user can move the robot EE around. When the pedal is not pressed ($x_d = x^*$, default), the setpoint regulation mode is recovered and interaction is disabled. Now the Git update law is resumed, and it will continue to learn based on the previous learned result. The latest set of position (x^*) ensures seamless switching between the regulation mode and the pHRI mode via the pedal switch.

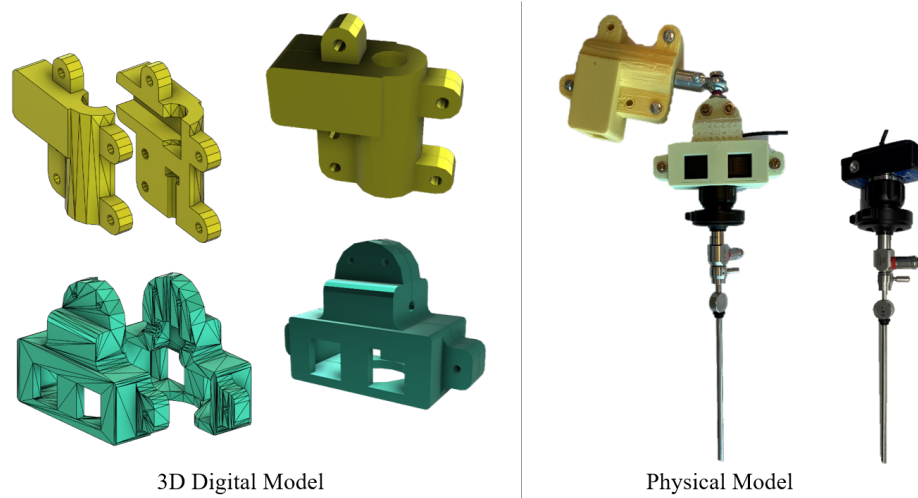
In the prototype, the left-arm robot has an arthroscope attached to its end-effector

(EE), while the right-arm robot has a handheld surgical bur attached to its EE. The left-arm robot is implemented with an impedance controller and a Git scheme. The former can provide compliant robot behavior thus ensuring a safe human-robot interaction, and the latter can accurately learn and compensate for gravity that is mainly caused by the attached external arthroscope. The right-arm robot is implemented with the point-based VF generation algorithm which can generate VF directly from point clouds with any shape, and rendering and providing VF force feedback to the operator as haptic clues. Since both the left-arm robot and right-arm robot in our prototype are haptic devices, the right-arm robot does not require a user-defined controller when implementing the point-based VF algorithm, and it can also provide haptic feedback with high fidelity. As shown in Figure 6.1c, the VF, force feedback, and surgical tools are visualized in Unity to provide additional visual feedback.

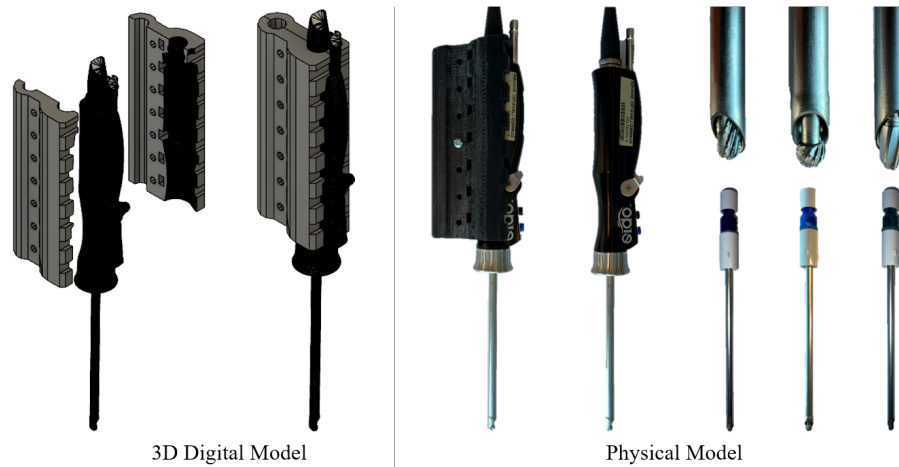
As shown in the prototype of Figure 6.1b, some connectors are designed and 3D-printed for the left-arm robot and the right-arm robot in order to attach the surgical instruments (*e.g.*, the arthroscope, and the handheld surgical bur) to the robot EEs. Especially, to attach the handheld bur to the right-arm robot, 3D scanning on the handheld bur is conducted first, and then the 3D model of the handheld bur having an ergonomic shape design is used to design the connector. The surgical instruments (arthroscope and surgical burs) and designed connectors are illustrated in Figure 6.3.

It is worth noting that for the right-arm robot, the surgical tool is rigidly mounted to the robot EE (a cylindrical handle bar) via the customized 3D-printed connector and bolts-nuts as shown in Figure 6.3b. The connector is deliberately designed to connect to the robot EE through an interference fit joint, thus no slippages can occur. The tool tip is then calibrated to be the new robot EE via the robot kinematics.

The dual-arm configuration of our prototype is designed to mimic the coordination between a surgeon's two hands while exercising dexterity and control, which is vital in navigating the surgical tools in the confined spaces during arthroscopic surgery, while providing haptic feedback as additional haptic assistance. With the left-arm



(a) Arthroscope



(b) Surgical burs

Figure 6.3: Connectors and surgical instruments.

robot-assisted arthroscope holder, the surgeon's hand will be freed to focus on some other more important tasks, and when necessary the surgeon can easily operate the arthroscope via a pedal switch. With the right-arm robot-assisted surgical tool with haptic feedback, surgeons can have a lower cognitive load and higher confidence when navigating the tool in confined spaces and conducting dexterous operation procedures. With additional 3D visual feedback, the surgeon can easily figure out where the tooltip is located in the big picture during the surgery.

6.3 Experimental Evaluation on the Prototype

In this section, we conducted a series of experimental tests to evaluate the prototype of the proposed two-arm system in conditions simulating arthroscopic surgical environments. Note that we conducted the evaluations on the left-arm robot and the right-arm robot respectively since the functions and control systems of the two robots are independent although they work collaboratively in the surgical scenario. This also ensures the evaluation goals are clear and focused for each robot.

We configure the two-arm system to replicate common surgical tasks, with the left-arm robot holding an arthroscope and providing an arthroscope view through the camera-like device, and the right-arm robot manipulating a surgical bur. These tests are meant to evaluate the system’s precision, responsiveness, and ability to handle complex maneuvers typical of joint arthroscopic surgeries. The following sections detail each experiment, outlining the setup, execution, and specific objectives aligned with our research goals in advancing robotic-assisted arthroscopic surgery.

6.3.1 Robotic system of the prototype

A prototype of a two-arm robot-assisted arthroscopic surgical system is constructed and illustrated in Figure 6.1. A pair of 6DOF Quanser’s High Definition Haptic Device (HD²) robots (Quanser Inc., Markham, ON, Canada), PY (positive y -axis) robot and NY (negative y -axis) robot are used as the left-arm robot and the right-arm robot, respectively, as illustrated in Figure 6.1. Note that the two robots have different definitions on the base frame, as shown in the figure. A relevant kinematics analysis of the HD² PY robot is available in [71]. The Cartesian workspace of each of the HD² robots is $[x, y, z, roll, pitch, yaw] : [800mm, 250mm, 350mm, 180^\circ, 180^\circ, continuous]$. The HD² robot highlights its features on large workspace and very low intervening dynamics, as well as highly back-drivable joints with negligible friction due to the parallel mechanism design. For more details on their kinematic features, please refer

to the system specifications¹. It is noteworthy that the large workspace and the kinematic design of the robotic arms help to accommodate the range of motion required in arthroscopic surgery, and it allows the arms to replicate the complex movements of a surgeon's hands within the constrained space of an arthroscopic procedure. In this work, the two robots are controlled via joint torque commands, which are sent from MATLAB/Simulink (version R2016a, MathWorks Inc., Natick, MA, USA) using Quarc real-time control software (Quanser Inc., Markham, ON, Canada). The control rate of the robot is 1,000 Hz. The MATLAB/Simulink and Quarc software run on a computer with a 3.20 GHz Intel(R) Core(TM) i5-3470 CPU with a Windows 7 Enterprise 64-bit operating system.

In the prototype, as shown in Figure 6.1, the left-arm robot EE is attached to an arthroscope (Sawbones[®], A Pacific Research Company, Vashon Island, Washington, USA) via one customized 3D-printed connector, while the right-arm robot EE is attached to a handheld surgical bur (Ergo[™] Shaver Handpiece, CONMED LINVATEC SHAVER, Linvatec Corporation, Largo, Florida, USA) via another customized 3D-printed connector. A modified FAST (fundamentals of arthroscopic surgery training) simulator (Sawbones[®], A Pacific Research Company, Vashon Island, Washington, USA) is used as a platform for an arthroscopic surgery mockup, while a soap block with a size of $22 \times 88 \times 48$ mm is used to represent the bone.

The two robots work independently but collaboratively in the proposed prototype. The left-arm robot is used as a robotic arthroscope holder which can hold the arthroscope still for the surgeon and prevent external disturbances. Its position can be adjusted when the surgeon needs to change the arthroscope view perspective. This is realized by a pedal, *i.e.*, when the pedal is unpressed (default), the arthroscope will be held still by the robot (interaction disabled in the robot control system), and when the pedal is pressed, the arthroscope can be moved to a new position (interaction enabled in the robot control system). The right-arm robot is attached to a

¹Quanser: <https://www.quanser.com/products/hd2-high-definition-haptic-device/>

surgical instrument at its EE, and the surgeon can conduct the surgery via the hand-held surgical instrument. During the surgery, the pose (position and rotation) of the instrument will be tracked in real-time by the robot, and visualized in Unity (version 2022.3.11f1, Unity Technologies, San Francisco, CA, USA). Moreover, a VF generated from a customized point cloud [77] is also visualized in Unity, and it is designed to help the surgeon remove the extra bone based on a preoperative plan and provide haptic feedback to the surgeon.

In the control block diagram of the prototype, as illustrated in Figure 6.2, a Git scheme is implemented with an impedance controller in the left-arm robot, where the former can accurately learn and compensate for gravity while the latter can ensure compliant robot behavior during physical human-robot interaction (*pHRI*). On the other hand, an algorithm of a point-based 3D VF-generating method [77] is implemented in the right-arm robot. With the VF algorithm, the operator can receive force feedback as additional haptic assistance when the surgical instrument is in contact with the VF. The communications between the right-arm robot and Unity are realized by user datagram protocol (UDP) at a rate of 100 Hz, where the real-time pose of the right-arm robot EE is sent to Unity for visualization.

6.3.2 Parameterization

For all the experiments in this work, the parameter values used in the controller and algorithms are listed in Table 6.1. In the following sub-sections, a series of experiments are conducted to evaluate the effectiveness of the proposed prototype of a two-arm robot-assisted arthroscopic surgical system. Note that the two robots are evaluated individually since they work independently despite being collaboratively in the system. A [video](#) demonstration for the experiments is available online².

²online demo video link: <https://youtu.be/ux10fDeb8dY>

Table 6.1: Parameterization for the experiments.

Robot	Parameter	Assigned Value
L	spring gain	$\mathbf{K}_m = 400\mathbf{I}$
L	damper gain	$\mathbf{D}_m = 40\mathbf{I}$
L	Git gain γ	1
L	learning rate η	0.001
R	VF force gain k_{vf}	500
R	VF sphere r_{vf}	0.002 m
R	VF r_c	0.005 m
R	VF r_1	0.00049 m
R	VF r_2	0.00501 m
R	VF r_3	0.010 m
R	VF sine wave points	$N = 30351$
R	VF sine wave px	$[-0.3, 0.1]$ m, step=0.002
R	VF sine wave py	$[-0.5, -0.2]$ m, step=0.002
R	VF sine wave pz	$pz = 0.03 \sin(\frac{2\pi}{0.1}(py + \frac{0.1}{2})) - 0.2$
-	bone block points	$N = 13500$
-	bone block size	$x \times y \times z = 0.022 \times 0.088 \times 0.048$ m

Note: $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ denote identity matrix. L, left-arm robot; R, right-arm robot; Git, gravity iterative learning scheme; VF, virtual fixture. The tunable gains are determined via trial and error with a binary search strategy. Note that for the left-arm robot, all four experiments share the same set of impedance gains for a fair comparison across the experiments.

6.3.3 Left-Arm Evaluation: Holding the arthroscope

The left-arm robot is evaluated by four experiments. The evaluation aim is to show that, by implementing the impedance controller and Git scheme, the robot can hold the arthroscope still (interaction disabled), while if needed, the robot allows the operator to adjust the scope view perspective (interaction enabled) via a pedal switch. In other words, when the pedal is unpressed ($Pedal = 0$, in regulation mode, default), the robot EE keeps the position still while rejecting any disturbances; when the pedal

is pressed ($Pedal = 1$, default, in $pHRI$ mode), the robot allows the operator to move the robot EE freely. The experiment setup for the left-arm robot is shown in Figure 6.4. Note that when the robot EE keeps the position still in the setpoint regulation mode, the robot will behave like a stiff spring if any accidental short disturbance (*e.g.*, user-applied force) is applied onto the robot EE, and the stiffness level can be tuned via the impedance gains for robustness. This ensures a safe human-robot interaction (expected or unexpected) through compliant robot behavior.

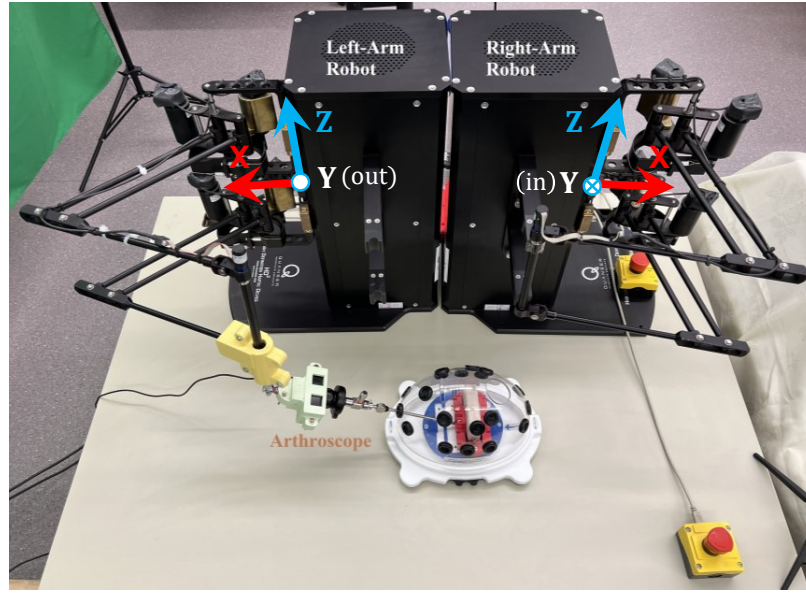


Figure 6.4: Left-arm robot experiment setup.

More specifically, Experiment 1 aims to evaluate the control accuracy of only an impedance controller. Experiment 2 aims to evaluate the control accuracy of only an impedance controller when a heavy external payload is attached to robot EE. Experiment 3 aims to evaluate the capabilities of the Git scheme to learn and compensate for the heavy external payload. The objective of Experiment 4 is to evaluate the precision, stability, and control capabilities of the prototype in mimicking a robot-assisted arthroscopic holder.

In Experiment 1 of the left-arm robot (L-Exp.1), nothing is attached to the robot EE. The robot is implemented with only an impedance controller (reduced to a PD

controller in *pHRI* scenario). The result of L-Exp.1 is shown in Figure 6.5. As can be seen in the figure, when the pedal is unpressed ($Pedal = 0$), the robot EE position can be accurately regulated, while when the pedal is pressed ($Pedal = 1$), the robot EE position is allowed to move freely. Noticed that in the yellow-colored area, when the pedal is pressed ($Pedal = 1$) but human-robot interaction is not involved, the robot EE will drift downward slowly as indicated by the red line due to inaccurate gravity compensation.

In Experiment 2 of the left-arm robot (L-Exp.2), nothing is attached to the robot EE at the beginning, and an external payload (515g) is attached to the robot EE during the task (18~60s). Same as the L-Exp.1, the robot is implemented with only an impedance controller. The result of L-Exp.2 is shown in Figure 6.6. Similar to the result of L-Exp.1, when the pedal is unpressed ($Pedal = 0$), the robot EE position can still be effectively regulated even with a heavy payload (515g) attached (18~60s), while when the pedal is pressed ($Pedal = 1$), the robot EE position is allowed to move freely. However, as can be observed in Figure 6.6a, the actual positions shift downward a bit (along z -axis) compared to the desired ones in regulation mode ($Pedal = 0$), revealing a relatively lower regulation accuracy than that in L-Exp.1 due to the uncompensated payload. Moreover, as indicated in the yellow-colored area, when the pedal is pressed ($Pedal = 1$) but human-robot interaction is not involved, the robot EE will drift downward quickly to the ground as indicated by the red line due to the heavy yet uncompensated payload (515g). The quick-dropping process is more clearly reflected in Figure 6.6b during which the commanded force is remarkably affected with a short oscillation occurred.

In Experiment 3 of the left-arm robot (L-Exp.3), the task is similar to that in L-Exp.2, *i.e.*, nothing is attached to the robot EE at the beginning, and an external payload (515g) is attached to the robot EE during the task (20~60s). Different from L-Exp.1 and L-Exp.2, the robot in L-Exp.3 is implemented with an additional Git scheme for gravity learning and compensation. The result of L-Exp.3 is shown in

Figure 6.7. Similarly, when the pedal is unpressed ($Pedal = 0$), the robot EE position can be accurately regulated even with a heavy payload (515g) attached (20~60s), while when the pedal is pressed ($Pedal = 1$), the robot EE position is allowed to move freely. Interestingly, as indicated in the yellow-colored area, when the pedal is pressed ($Pedal = 1$) but human-robot interaction is not involved, the robot EE does not drift downward anymore due to the Git scheme having accurately learned and compensated for the gravity (see Figure 6.7b). Note that here the yellow-colored area involves physical human-robot interaction as well. Additionally, it is worth noting that the Git scheme revealed a quick and smooth convergence process during 20~25s immediately after the heavy payload attached to the robot EE.

In Experiment 4 of the left-arm robot (L-Exp.4), the robot EE is attached with an arthroscope (713g), and the robot is implemented with a Git scheme for gravity compensation in addition to an impedance controller, which is the same as that in L-Exp.3. The result of L-Exp.4 is shown in Figure 6.8. As shown in the figure, when the pedal is unpressed ($Pedal = 0$), the robot EE position (*i.e.*, the arthroscope) can be accurately regulated, while when the pedal is pressed ($Pedal = 1$), the robot EE position is allowed to move freely to adjust the arthroscope view perspective. This evaluated the effectiveness of the left-arm robot for holding with the arthroscope. It is worth noting that Figure 6.8b revealed another quick and smooth convergence process during 0~10s which demonstrated the stability of the Git scheme in the transient process of converging.

In summary, the results from L-Exp.1 indicate that an impedance controller can provide robot compliance, but the control accuracy can be potentially affected by uncompensated gravity. The findings of L-Exp.2 emphasize the results of L-Exp.1 more clearly that an impedance controller alone is not capable of dealing with heavy external payloads when mimicking a heavy arthroscope attached to the robot EE. In regulation mode, the uncompensated heavy payload will lower the regulation accuracy, while in *pHRI* mode, the uncompensated heavy payload will drive the robot

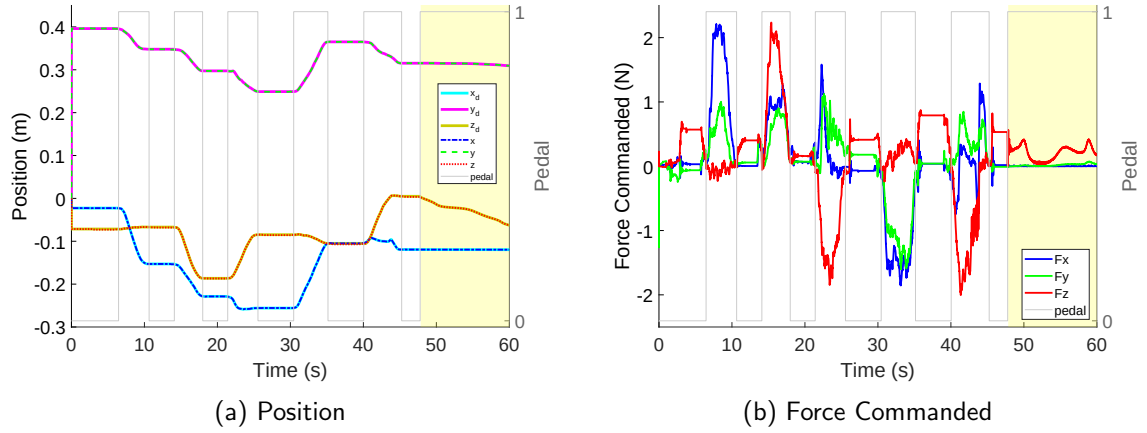


Figure 6.5: Experiment result of L-Exp.1 in p HRI scenario with only an impedance controller implemented. Note that the yellow-colored area corresponds to the p HRI mode ($Pedal = 1$) but no p HRI occurring.

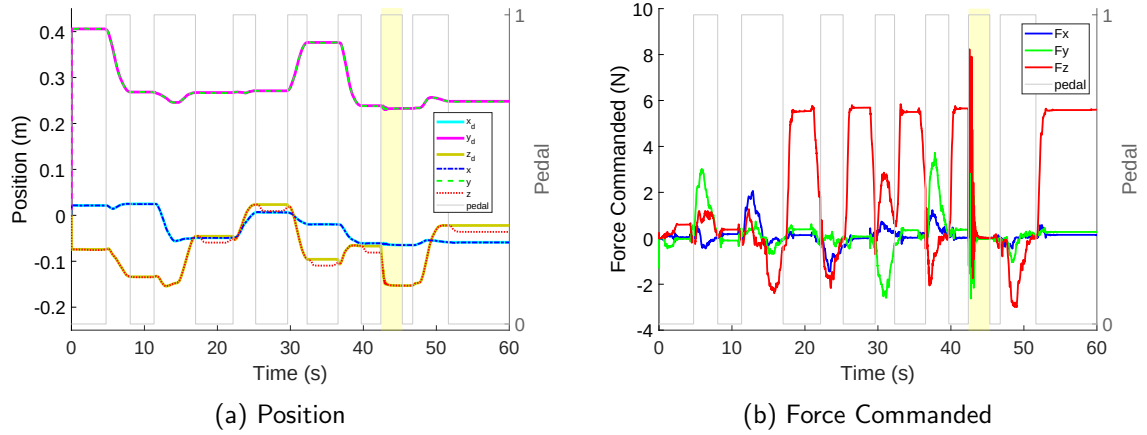


Figure 6.6: Experiment result of L-Exp.2 in p HRI scenario with only an impedance controller implemented, while an extra payload (515g) attached to the robot EE. Note that the yellow-colored area corresponds to the p HRI mode ($Pedal = 1$) but no p HRI occurring.

EE to drop toward the ground if there is no human-robot interaction involved. The outcomes of L-Exp.3 help validate the capability of the Git scheme to compensate for the heavy payload gravity. In regulation mode, the regulation accuracy can be recovered to a high level, while in p HRI mode, the robot EE attached with a heavy payload can stay in the air even if there is no human-robot interaction involved, and all of those are due to the payload is accurately compensated by the Git scheme. The results of L-Exp.4 validate and support our primary goal on system accuracy and

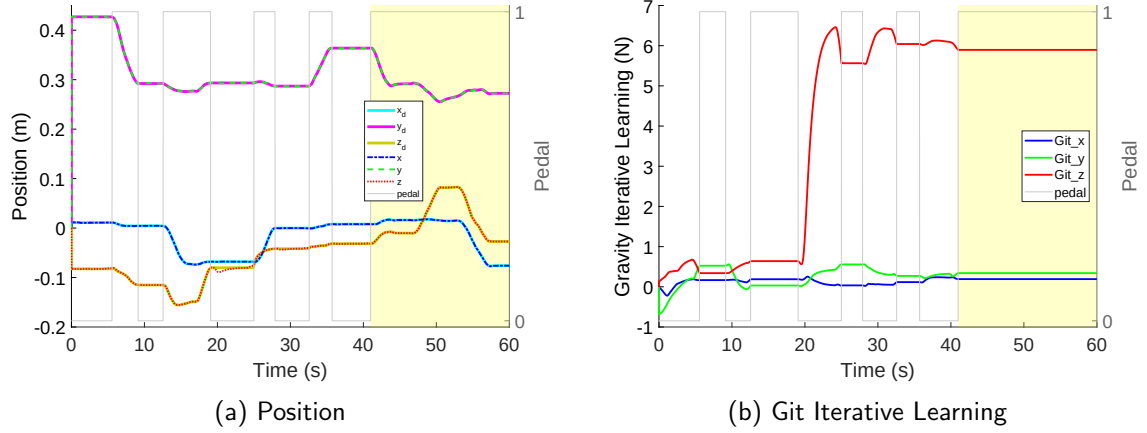


Figure 6.7: Experiment result of L-Exp.3 in p HRI scenario with an impedance controller and Git scheme implemented, while an extra payload (515g) is attached to the robot EE. Note that the yellow-colored area corresponds to the p HRI mode ($Pedal = 1$), and both p HRI and non- p HRI are involved.

stability of the robot-assisted arthroscope holder.

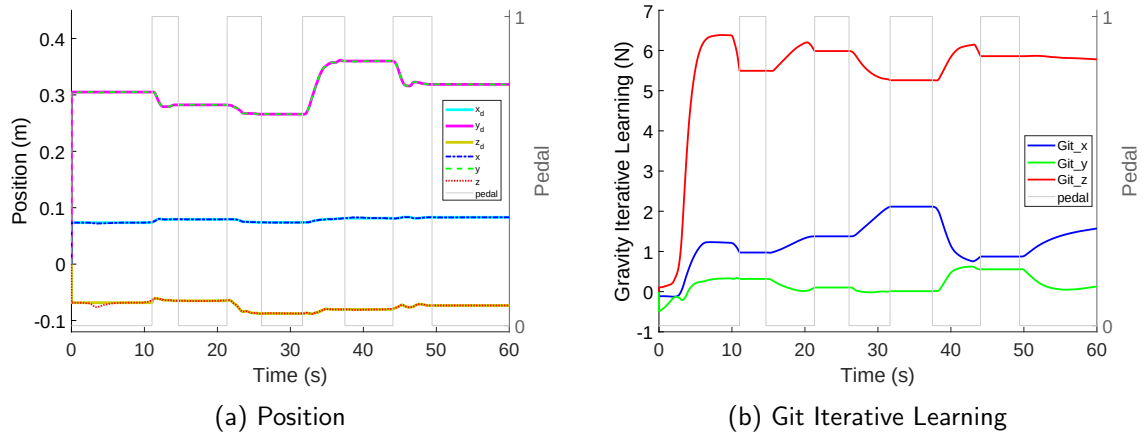


Figure 6.8: Experiment result of L-Exp.4 in p HRI scenario with an impedance controller and Git scheme implemented, while an arthroscope (713g) is attached to the robot EE.

6.3.4 Right-Arm Evaluation: Burring bone with VF

The right-arm robot is implemented with a point-based VF algorithm where the VF can be generated directly from a point cloud with any shape [77]. Here in this work, a point cloud of a sinusoidal wave is employed as shown in Figure 6.9a. In the figure, the brown rectangular area represents a bone block, while the red circle represents

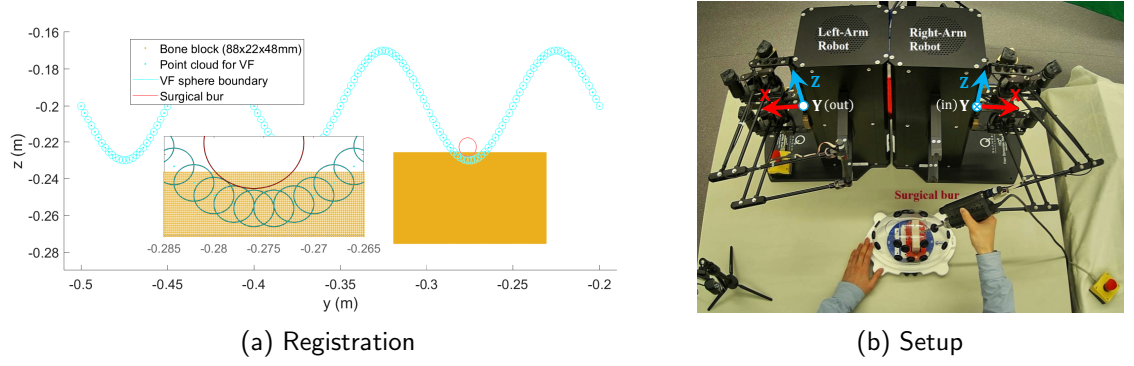


Figure 6.9: Right-arm robot experiment setup and VF-bone registration.

the surgical bur. The bone above the sine wave needs to be removed during the task. As shown in Figure 6.9b, a surgical bur is attached to the right-arm robot EE which is used to remove the extra bone. The detailed parameterization related to the point cloud is summarized in Table 6.1.

The VF, surgical bur, and bone block are also visualized in Unity in real-time during the task as illustrated in Figure 6.1c. The evaluation aim for the right-arm robot is to show that, by implementing the point-based VF algorithm, the VF can provide force feedback to the operator during surgery operations (*e.g.*, extra bone removal), thus assist the operator in bone-removing with both haptic cues and visualization together. In order to ensure the bone block is burred based on the designed VF as illustrated in Figure 6.9a, a VF-bone registration needs to be done. The procedures for VF-bone registration we used are the following, (1) the bone is stably fixed on an experimental platform (*i.e.*, the FAST simulator shown in Figure 6.9b); (2) the platform is then stably fixed at an appropriate location within the robot workspace; (3) the bone position coordinates in the robot workspace can then be retrieved via the robot EE (*i.e.*, the tool tip); (4) the VF is then registered such that the VF valley penetrates into the top side of the bone for 3 mm along z -axis as illustrated in Figure 6.9a. It is worth mentioning that the VF-bone registration method we used has a main focus on the z -axis registration. However, the actual burring depth by the bur along the z -axis still will be affected by the user-configured stiffness of the VF,

e.g., a “soft” VF will allow bigger burring depth while a “stiff” VF will allow smaller burring depth.

There are two experiments designed for evaluating the right-arm robot. The first experiment (R-Exp.1) is to conduct a bone-removing task in an open space (mimicking open surgery) where the cover of the FAST simulator is removed (see Figure 6.10), while the second experiment (R-Exp.2) is to conduct the same bone-moving task via a small portal (mimicking MIS) where the cover of the FAST simulator is involved (see Figure 6.9b).

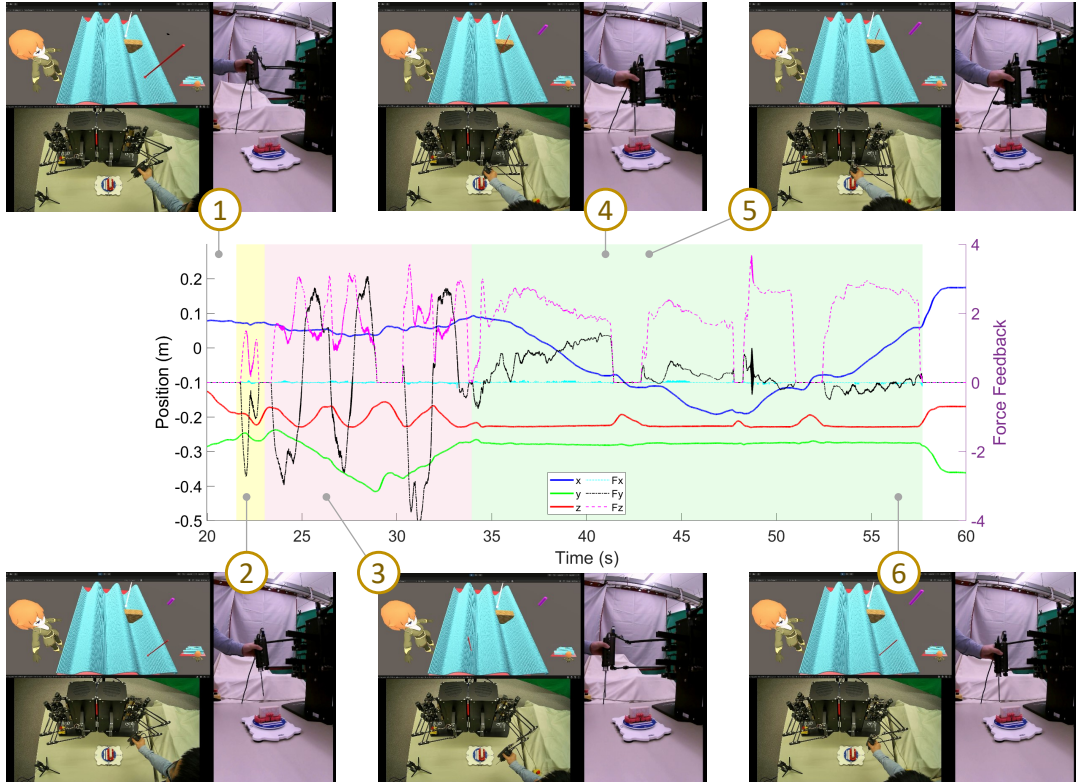


Figure 6.10: Probing test result of R-Exp.1 in *pHRI* scenario with an algorithm of point-based VF implemented, while a handheld surgical bur (435g) is attached to the robot EE. Scenario ①, preparation stage; Scenario ②, VF probing and force feedback test; Scenario ③, VF sinusoidal outline probing test; Scenario ④ ⑤, VF valley probing test; Scenario ⑥, ending stage.

A probing test on the right-arm robot is performed first based on R-Exp.1 as shown in Figure 6.10. Some scenario diagrams for typical procedures are also presented in the figure. Scenario ① is in preparation stage. Scenario ② is in a VF probing and

force feedback test, where the goal is to test the VF force rendering as well as its visualization in Unity. Scenario ③ is in a VF outline probing test, where the goal is to detect the sinusoidal outline of the VF. Scenario ④ and ⑤ are in a VF valley probing test, where the goal is to move along the valley of the VF. Scenario ④ is on one side of the bone block, and scenario ⑤ is on the other side of the bone block. Scenario ⑥ is in an ending stage. The probing test results show that the VF force can be appropriately rendered and delivered to the operator, and the VF, surgical bur, and force feedback values are also correctly visualized in Unity.

In traditional MIS elbow arthroscopy, the surgeon removes the osteophytes under the arthroscope view, and the actual amount of the bone to be removed mainly relies on the surgeon's experience and the memorized preplan in the surgeon's brain. The right-arm robot can assist the surgeon in removing bone that is bound by surgeon-defined VF, while providing haptic feedback to the surgeon to indicate where the VF boundary is. Therefore, the goal of the bone-burring task in this work is to remove the bone bounded by the VF as illustrated in Figure 6.9a.

The bone-burring task results of R-Exp.1 and R-Exp.2 are shown in Figure 6.11a and Figure 6.11b, respectively. As can be seen in the figure, both experiments of R-Exp.1 and R-Exp.2 can generate relatively good bone-burring task performance with the VF assistance. In other words, the bone-burring task can be effectively conducted with the right-arm robot either in simulated open surgery (Figure 6.11a) or in simulated MIS surgery (Figure 6.11b), although the former results in a relatively smooth and a bit wider bone-burring surface than the latter. This is reasonable since the latter of MIS surgery is performed via a restricted portal. These results evaluated the effectiveness of the right-arm robot including the point-based VF generation algorithm, VF force rendering, haptic feedback, and visualization in Unity. Note that for the bone-burring task results, we provide a qualitative result rather than a quantitative result, this is because the quantitative performance heavily depends on the parameter configuration when setting the force rendering of the VF, *i.e.*, when

a “soft” VF is configured, more bone material will be removed than planned since higher penetration to the VF boundary is allowed, while on the contrary, when a “stiff” VF is configured, few more bone material will be removed than planned since low penetration to the VF boundary is allowed.

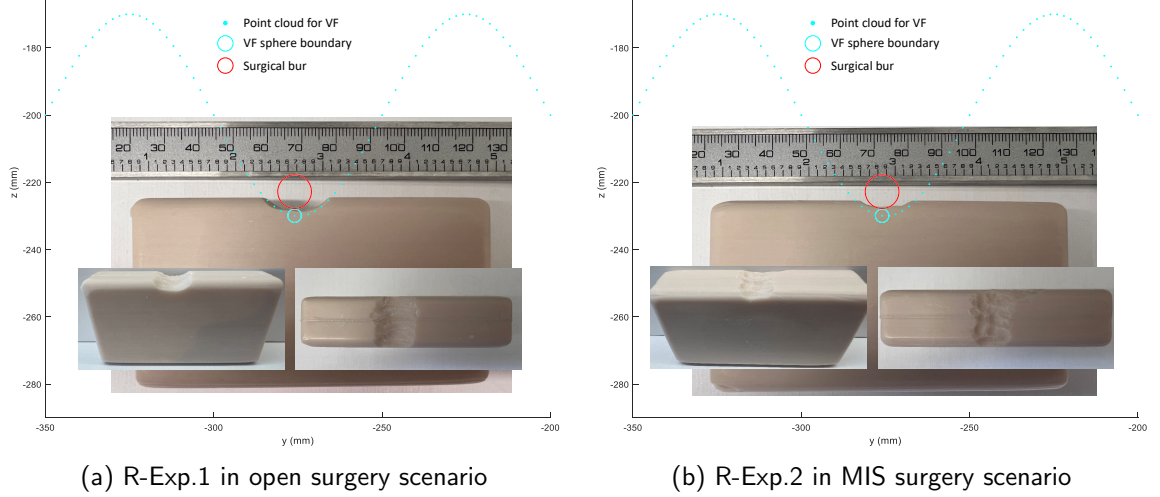


Figure 6.11: Experiment results of R-Exp.1 and R-Exp.2. The bone is represented by a soap block with a size of $22 \times 88 \times 48$ mm.

6.3.5 Future work

In surgical scenarios, safety is of paramount importance. For the left-arm robot in the proposed robot-assisted system, a safe and compliant human-robot interaction can be ensured by the impedance controller, which can guarantee compliant robot behavior. For the right-arm robot, since VF assistance and haptic feedback are the main features, the safety of human-robot interaction solely relies on the stability and reliability of the VF algorithm, which has been systematically verified in our previous work [77]. On the other hand, system accuracy is another main concern. The Git scheme implemented in the left-arm robot can guarantee accurate regulation through accurate gravity estimation and compensation, while the performance accuracy of the right-arm robot relies on the user-configured stiffness of the VF.

For the bone-burring tasks conducted via the right-arm robot, we provided only qualitative performance results as introduced earlier. In future work, we will make

a more deliberate experimental design such that the experimental results can be evaluated quantitatively. For example, we can reconstruct the 3D digital models of the bone before and after the surgical operations, and then make a comparison quantitatively. We can also design an experiment to compare the task performance with and without VF assistance, to evaluate to what extent the VF assistance and haptic feedback can help to improve the task performance.

There are some other aspects that can be improved in future work. First, the two robots work independently despite collaboratively in this work. To visualize the real-time position of the left-arm robot in Unity, it needs to be registered into the same coordinate system as the right-arm robot. Second, the bone is represented by a soap block which could have different stiffness of material properties from the bone. A real bone would be used in future work. Third, the registration of the bone into the right-arm robot workspace is only conducted along the vertical z -axis as a simplified case. In future work, full registration of a real bone should be conducted along all three Cartesian axes. Additionally, a preplanned image-based surgical scenario can be also involved. The inclusion of preoperative images would allow for more precise mapping of the surgical field and enable the robot to navigate and interact with the patient's anatomy with a higher degree of fidelity. Last but not least, the experiments in this work are dry experiments without fluid filling around the bone. In future work, wet experiments should be designed for more realistic arthroscopic surgeries.

Some other improvements can be made in future work. For example, the miniaturization of the components of the connectors between robot and surgical tool, enhancing the additional 3D visual feedback by introducing augmented reality (AR) techniques, and designing more realistic and more complex surgical scenarios where robotic assistance can make a difference.

6.4 Conclusion

In this chapter, a prototype of a two-arm robot-assisted arthroscopic surgical system is presented. The system is composed of a pair of haptic devices. The left-arm robot is attached to an arthroscope while the right-arm robot is attached to a handheld surgical bur. The left-arm robot is implemented with an impedance controller and a Git scheme, where the former ensures a safe human-robot interaction while the latter accurately learns and compensates for gravity. The right-arm robot is implemented with a point-based VF generation algorithm, which can generate VF directly from point clouds with any shape. A series of experiments are conducted to evaluate the effectiveness of the prototype system. The results show that the left-arm robot can effectively hold the arthroscope still and allow the operator to move the arthroscope via a pedal switch whenever needed, and the right-arm robot can render appropriate VF force feedback from the VF algorithm and deliver it to the operator as haptic assistance. Also, the VF, the bone, and the surgical bur with its real-time position are visualized in Unity to provide additional visual feedback to the operator.

Debridement of osteophytes is a specific example surgery we used to evaluate the prototype in this work, and it is a small arthroscopic surgery that is common in the elbow and hip. Beyond this, the proposed prototype is promised to be used in a wide variety of arthroscopic surgeries in orthopedics including (1) soft tissue repair/reconstruction, which is most commonly happened in the shoulder (*e.g.*, rotator cuff repair, labral repair) and the knee (*e.g.*, anterior cruciate ligament (ACL) repair/reconstruction, meniscal repair), and these are by far the most common applications of arthroscopy in orthopedics based on the volume of work, but there is also some interest in elbow recently for ligament augmentation, repair/reconstruction; (2) bony debridement, which is commonly happened in elbow (osteophytes), hip, and shoulder (distal clavicle excision); (3) trauma, *e.g.*, fracture reduction/fixation, *etc.* With the robotic assistance of the proposed two-arm system, it is expected to help increase the

surgeon's accuracy and reliability while reducing invasiveness.

The implications of this work can extend beyond orthopedics, suggesting wider applicability of such advanced robotic systems in various medical fields. The development and successful evaluation of our prototype underline the potential for more precise, controlled, and ergonomic surgical procedures using similar concepts derived from impedance control for robot compliance, a Git scheme for gravity compensation, and a point-based VF generation algorithm for objects in any shape.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

Robot-assisted surgery has received growing attention in the field of orthopedics and beyond. Most of the existing robot-assisted surgical systems in orthopedics take more focus on open surgery than minimally invasive surgery (MIS). In this thesis, a prototype of a two-arm robot-assisted system is developed for arthroscopic surgery.

In **Chapter 1**, the evolvement of robotic systems and techniques in orthopedics was reviewed, then the development of a two-arm robot-assisted arthroscopic surgical system was motivated, and the challenges of building such a system were identified.

In **Chapter 2**, an integrated framework of combining impedance control and non-linear disturbance observer (NDOB) was proposed for arthroscopic surgery. The impedance controller can provide compliant robot behavior for a safe human-robot interaction while the NDOB can compensate for dynamic uncertainties. By combining impedance control and NDOB, the integrated framework can achieve accurate impedance control when dynamic uncertainties exist in the robotic system. However, the NDOB observer always estimates for lumped uncertainties and cannot separate any of the components.

In **Chapter 3**, in aiming to separate some specific components in the uncertainties, a framework for using neural network (NN) to learn dynamic uncertainties is proposed. The effectiveness and performance of the trained NN model were verified

by simulations. However, it is not efficient to implement the NN model in practice.

In **Chapter 4**, an iterative learning scheme with a steady-state scaling strategy for gravity compensation, *i.e.*, Git (gravity iterative learning), is developed. The Git scheme can accurately learn for gravity compensation in an iterative and online manner, and it is in a simple and compact form. Moreover, by integrating an impedance controller and the Git scheme, the robot EE can keep still at any designated pose, while also allowing the operator to move it freely via a pedal switch whenever needed.

In **Chapter 5**, an algorithm of point-based virtual fixture (VF) generating method was presented. The VF algorithm can generate VF directly from point clouds with any shape. In applications, the VF algorithm can be used to generate VF from hand-drawn curves with free-style shapes in a preplanned image, by extracting the hand-drawn curve as a set of point clouds.

In **Chapter 6**, a prototype of a two-arm robot-assisted system for arthroscopic surgery is presented and evaluated by assembling the work from **Chapter 4** and from **Chapter 5**. The left-arm robot is used as a robot-assisted arthroscope holder, and it can hold the arthroscope still at a designated pose, while also allowing the operator to move it via a pedal switch whenever needed. The left-arm robot is implemented with an impedance controller and a gravity iterative learning (Git) scheme, while the right-arm robot is implemented with a point-based virtual fixture (VF) algorithm. The right-arm robot is used as a robot-assisted surgical bur providing VF assistance and haptic feedback during the surgery. Furthermore, the VF, the bone, and the surgical bur with its real-time position are visualized in a 3D digital environment as additional visual feedback for the operator.

7.2 Future Research Directions

In future work, in aiming to further improve the prototype and bring it into a real clinical environment, the following research topics are proposed:

- (1) Registration of robot-image-patient coordinate systems is a critical step before the operation in surgery. More importantly, it directly determines the accuracy of surgical outcomes. An intelligent, accurate, and dynamic registration strategy will be developed in future work.
- (2) A robot-assisted arthroscopic surgical simulator based on the current prototype can be developed. The simulator can be used to train surgeons and help them acquire necessary surgical skills before conducting real surgery on patients.
- (3) A more realistic surgical environment can be set up for further evaluating and improving the prototype. For example, using a real physical bone instead of a mock-up bone, designing a wet experimental environment where the bone is surrounded by water, making the bone enclosed and only visible via the arthroscope, and performing a surgical task by using the two robots collaboratively.
- (4) User studies should be conducted by employing arthroscopic surgeons to test the prototype. Arthroscopic surgeons know best the requirements of the prototype for arthroscopic surgery, and their user experience and feedback should be collected for further improvements of the prototype.
- (5) A 3D digital bone can be used for generating VF (virtual fixture) to provide haptic assistance during surgery. A patient-specific 3D digital bone can be obtained before the surgery, then the surgeons can conduct a virtual surgery on the 3D bone using software, *e.g.*, removing osteophytes, and then an expected 3D bone can be obtained. During surgery, the expected 3D bone can be used to generate VF assistance to assist the surgeon in accurately removing osteophytes.

References

- [1] Jake J Abbott, Panadda Marayong, and Allison M Okamura. “Haptic Virtual Fixtures for Robot-Assisted Manipulation”. In: *Robotics Research*. Springer, 2007, pp. 49–64.
- [2] Alaa Eldin Abdelaal, Prateek Mathur, and Septimiu E Salcudean. “Robotics in Vivo: A Perspective on Human-Robot Interaction in Surgical Robotics”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), pp. 221–242.
- [3] Elan Z Ahronovich, Nabil Simaan, and Karen M Joos. “A Review of Robotic and OCT-Aided Systems for Vitreoretinal Surgery”. In: *Advances in Therapy* 38.5 (2021), pp. 2114–2129.
- [4] Arash Ajoudani, Nikos Tsagarakis, and Antonio Bicchi. “Tele-Impedance: Teleoperation With Impedance Regulation Using a Body-Machine Interface”. In: *The International Journal of Robotics Research* 31.13 (2012), pp. 1642–1656.
- [5] Mojtaba Akbari, Jay Carriere, Tyler Meyer, Ron Sloboda, Siraj Husain, Nawaid Usmani, and Mahdi Tavakoli. “Robotic Ultrasound Scanning With Real-Time Image-Based Force Adjustment: Quick Response for Enabling Physical Distancing During the COVID-19 Pandemic”. In: *Frontiers in Robotics and AI* 8 (2021), p. 645424.
- [6] Farshid Alambeigi, Mahsan Bakhtiarinejad, Shahriar Sefati, Rachel Hegeman, Iulian Iordachita, Harpal Khanuja, and Mehran Armand. “On the Use of a Continuum Manipulator and a Bendable Medical Screw for Minimally Invasive Interventions in Orthopedic Surgery”. In: *IEEE Transactions on Medical Robotics and Bionics* 1.1 (2019), pp. 14–21.
- [7] Farshid Alambeigi, Yu Wang, Shahriar Sefati, Cong Gao, Ryan J Murphy, Iulian Iordachita, Russell H Taylor, Harpal Khanuja, and Mehran Armand. “A Curved-Drilling Approach in Core Decompression of the Femoral Head Osteonecrosis Using a Continuum Manipulator”. In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1480–1487.
- [8] George S Athwal. *PART IV, Elbow and Forearm. Chapter 26, Elbow Arthroscopy*. Elsevier Inc., 2017.
- [9] Aleks Attanasio, Bruno Scaglioni, Elena De Momi, Paolo Fiorini, and Pietro Valdastri. “Autonomy in Surgical Robotics”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2021), pp. 651–679.

- [10] Sergei Basovich, Shai A Arogeti, Yonattan Menaker, and Ziv Brand. “Magnetically Levitated 6-DOF Precision Positioning Stage With Uncertain Payload”. In: *IEEE/ASME Transactions on Mechatronics* 21.2 (2015), pp. 660–673.
- [11] Andrew K Battenberg, Nathan A Netravali, and Jess H Lonner. “A Novel Handheld Robotic-Assisted System for Unicompartmental Knee Arthroplasty: Surgical Technique and Early Survivorship”. In: *Journal of Robotic Surgery* 14 (2020), pp. 55–60.
- [12] Ryan A Beasley. “Medical Robots: Current Systems and Research Directions”. In: *Journal of Robotics* 2012 (2012).
- [13] Karis Bennett and Srinath Kamineni. “History of Elbow Arthroscopy”. In: *Journal of Arthroscopic Surgery and Sports Medicine* 1.1 (2020), pp. 23–31.
- [14] Stuart A Bowyer, Brian L Davies, and Ferdinando Rodriguez y Baena. “Active Constraints/Virtual Fixtures: A Survey”. In: *IEEE Transactions on Robotics* 30.1 (2013), pp. 138–157.
- [15] Jay Carriere, Jason Fong, Tyler Meyer, Ron Sloboda, Siraj Husain, Nawaid Usmani, and Mahdi Tavakoli. “An Admittance-Controlled Robotic Assistant for Semi-Autonomous Breast Ultrasound Scanning”. In: *2019 International Symposium on Medical Robotics (ISMR)*. IEEE. 2019, pp. 1–7.
- [16] Murat Cenk Çavuşoğlu, David Feygin, and Frank Tendick. “A Critical Study of the Mechanical and Electrical Properties of the Phantom Haptic Interface and Improvements for High Performance Control”. In: *Presence: Teleoperators & Virtual Environments* 11.6 (2002), pp. 555–568.
- [17] Antonia F Chen, Gregory S Kazarian, Galen W Jessop, and Asim Makhdum. “Robotic Technology in Orthopaedic Surgery”. In: *The Journal of Bone & Joint Surgery* 100.22 (2018), pp. 1984–1992.
- [18] Wen-Hua Chen, Donald J Ballance, Peter J Gawthrop, and John O’Reilly. “A Nonlinear Disturbance Observer for Robotic Manipulators”. In: *IEEE Transactions on Industrial Electronics* 47.4 (2000), pp. 932–938.
- [19] Wen-Hua Chen, Jun Yang, Lei Guo, and Shihua Li. “Disturbance-Observer-Based Control and Related Methods—An Overview”. In: *IEEE Transactions on Industrial Electronics* 63.2 (2016), pp. 1083–1095.
- [20] Lingbo Cheng, Jay Carriere, Jakub Piwowarczyk, Daniel Aalto, Nabil Zemiti, Marie de Boutray, and Mahdi Tavakoli. “Admittance-Controlled Robotic Assistant for Fibula Osteotomies in Mandible Reconstruction Surgery”. In: *Advanced Intelligent Systems* 3.1 (2021), p. 2000158.
- [21] Shih-Hsuan Chien, Jyun-Hsiang Wang, and Ming-Yang Cheng. “Performance Comparisons of Different Observer-Based Force-Sensorless Approaches for Impedance Control of Collaborative Robot Manipulators”. In: *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*. IEEE. 2020, pp. 326–331.

- [22] Nak Young Chong, Tetsuo Kotoku, Kohtaro Ohba, and Kazuo Tanie. “Virtual Repulsive Force Field Guided Coordination for Multi-Telerobot Collaboration”. In: *2001 IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 1. IEEE. 2001, pp. 1013–1018.
- [23] François Conti, Jaeheung Park, and Oussama Khatib. “Interface Design and Control Strategies for a Robot Assisted Ultrasonic Examination System”. In: *Experimental Robotics*. Springer. 2014, pp. 97–113.
- [24] David M Dalton, Thomas P Burke, Enda G Kelly, and Paul D Curtin. “Quantitative Analysis of Technological Innovation in Knee Arthroplasty: Using Patent and Publication Metrics to Identify Developments and Trends”. In: *The Journal of Arthroplasty* 31.6 (2016), pp. 1366–1372.
- [25] Brian Davies. “A Review of Robotics in Surgery”. In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 214.1 (2000), pp. 129–140.
- [26] A De Luca and S Panzieri. “A Simple Iterative Scheme for Learning Gravity Compensation in Robot Arms”. In: *Proceedings of the 36th ANIPLA Annual Conference (Automation 1992)*. 1992, pp. 459–471.
- [27] Alessandro De Luca and Raffaella Mattone. “Actuator Failure Detection and Isolation using Generalized Momenta”. In: *2003 IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 1. IEEE. 2003, pp. 634–639.
- [28] Alessandro De Luca and Raffaella Mattone. “Sensorless Robot Collision Detection and Hybrid Force/Motion Control”. In: *2005 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2005, pp. 999–1004.
- [29] Alessandro De Luca and Giovanni Ulivi. “Iterative Learning Control of Robots With Elastic Joints”. In: *1992 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Computer Society. 1992, pp. 1920–1926.
- [30] Claudia D’Ettorre, Andrea Mariani, et al. “Accelerating Surgical Robotics Research: Reviewing 10 Years of Research With the dVRK”. In: *arXiv preprint arXiv:2104.09869* (2021).
- [31] George Charles Devol. “Programmed Article Transfer”. In: *USA Patent 2.988* (1961), p. 237.
- [32] Liang Ding, Hongjun Xing, Haibo Gao, Ali Torabi, Weihua Li, and Mahdi Tavakoli. “VDC-Based Admittance Control of Multi-DOF Manipulators Considering Joint Flexibility via Hierarchical Control Framework”. In: *Control Engineering Practice* 124 (2022), p. 105186.
- [33] Bin Duan, Linfei Xiong, Xiaolong Guan, Yongqing Fu, and Yongwei Zhang. “Tele-Operated Robotic Ultrasound System for Medical Diagnosis”. In: *Biomedical Signal Processing and Control* 70 (2021), p. 102900.

- [34] Anders El-Galaly, Clare Grazal, Andreas Kappel, Poul Torben Nielsen, Steen Lund Jensen, and Jonathan A Forsberg. “Can Machine-Learning Algorithms Predict Early Revision TKA in the Danish Knee Arthroplasty Registry?” In: *Clinical Orthopaedics and Related Research* 478.9 (2020), p. 2088.
- [35] Nima Enayati, Elena De Momi, and Giancarlo Ferrigno. “Haptics in Robot-Assisted Surgery: Challenges and Benefits”. In: *IEEE Reviews in Biomedical Engineering* 9 (2016), pp. 49–65.
- [36] Khemerin Eng, Alistair Eyre-Brook, and David W Shields. “A Systematic Review of the Utility of Intraoperative Navigation During Total Shoulder Arthroplasty”. In: *Cureus* 14.12 (2022).
- [37] Sybille Facca, Sarah Hendriks, Gustavo Mantovani, Jesse C Selber, and Philippe Liverneaux. “Robot-Assisted Surgery of the Shoulder Girdle and Brachial Plexus”. In: *Seminars in Plastic Surgery*. Vol. 28. 01. Thieme Medical Publishers. 2014, pp. 039–044.
- [38] Ting-Yun Fang, Haichong K Zhang, Rodolfo Finocchi, Russell H Taylor, and Emad M Bactor. “Force-Assisted Ultrasound Imaging System Through Dual Force Sensing and Admittance Robot Control”. In: *International Journal of Computer Assisted Radiology and Surgery* 12.6 (2017), pp. 983–991.
- [39] Navid Feizi, Mahdi Tavakoli, Rajni V Patel, and S Farokh Atashzar. “Robotics and AI for Teleoperation, Tele-Assessment, and Tele-Training for Surgery in the Era of COVID-19: Existing Challenges, and Future Vision”. In: *Frontiers in Robotics and AI* 8 (2021).
- [40] Gregory S Fischer, Anton Deguet, Csaba Csoma, Russell H Taylor, Laura Fayad, John A Carrino, S James Zinreich, and Gabor Fichtinger. “MRI Image Overlay: Application to Arthrography Needle Insertion”. In: *Computer Aided Surgery* 12.1 (2007), pp. 2–14.
- [41] Jason Fong, Hossein Rouhani, and Mahdi Tavakoli. “A Therapist-Taught Robotic System for Assistance During Gait Therapy Targeting Foot Drop”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 407–413.
- [42] Javad Fotouhi, Clayton P Alexander, Mathias Unberath, Giacomo Taylor, Sing Chun Lee, Bernhard Fuerst, Alex Johnson, Greg M Osgood, Russell H Taylor, Harpal Khanuja, et al. “Plan in 2-D, Execute in 3-D: An Augmented Reality Solution for Cup Placement in Total Hip Arthroplasty”. In: *Journal of Medical Imaging* 5.2 (2018), p. 021205.
- [43] Masakatsu G Fujie and Bo Zhang. “State-Of-The-Art of Intelligent Minimally Invasive Surgical Robots”. In: *Frontiers of Medicine* 14.4 (2020), pp. 404–416.
- [44] Jose Carlos Garcia Jr and Edna Frasson de Souza Montero. “Endoscopic Robotic Decompression of the Ulnar Nerve at the Elbow”. In: *Arthroscopy Techniques* 3.3 (2014), e383–e387.

- [45] Gianluca Garofalo, Nico Mansfeld, Julius Jankowski, and Christian Ott. “Sliding Mode Momentum Observers for Estimation of External Torques and Joint Acceleration”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6117–6123.
- [46] Francesco Garzotto, Rosanna Irene Comoretto, Marlies Ostermann, Federico Nalesso, Dario Gregori, Maria Giuseppina Bonavina, Giorgio Zanardo, and Gaudenzio Meneghesso. “Preventing Infectious Diseases in Intensive Care Unit by Medical Devices Remote Control: Lessons from COVID-19”. In: *Journal of Critical Care* 61 (2021), pp. 119–124.
- [47] Claudio Gaz, Marco Cognetti, Alexander Oliva, Paolo Robuffo Giordano, and Alessandro De Luca. “Dynamic Identification of the Franka Emika Panda Robot With Retrieval of Feasible Parameters Using Penalty-Based Optimization”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4147–4154.
- [48] Tricia L Gibo, Lawton N Verner, David D Yuh, and Allison M Okamura. “Design Considerations and Human-Machine Performance of Moving Virtual Fixtures”. In: *2009 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2009, pp. 671–676.
- [49] Matthew W Gilbertson and Brian W Anthony. “An Ergonomic, Instrumented Ultrasound Probe for 6-Axis Force/Torque Measurement”. In: *2013 the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2013, pp. 140–143.
- [50] Tirth Ginoya, Yaser Maddahi, and Kourosh Zareinia. “A Historical Review of Medical Robotic Platforms”. In: *Journal of Robotics* 2021 (2021), pp. 1–13.
- [51] Alain Gourdon, Philippe Poignet, Gerard Poisson, Pierre Vieyres, and Pierre Marche. “A New Robotic Mechanism for Medical Application”. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 1999, pp. 33–38.
- [52] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. “Robot Collisions: A Survey on Detection, Isolation, and Identification”. In: *IEEE Transactions on Robotics* 33.6 (2017), pp. 1292–1312.
- [53] Bang-Sang Hahn and Jeong-Yoon Park. “Incorporating New Technologies to Overcome the Limitations of Endoscopic Spine Surgery: Navigation, Robotics, and Visualization”. In: *World Neurosurgery* 145 (2021), pp. 712–721.
- [54] Jingqing Han. “A Class of Extended State Observers for Uncertain Systems”. In: *Control and Decision* 10.1 (1995). [In Chinese], pp. 85–88.
- [55] Jingqing Han. “From PID to Active Disturbance Rejection Control”. In: *IEEE Transactions on Industrial Electronics* 56.3 (2009), pp. 900–906.
- [56] Blake Hannaford, Jacob Rosen, Diana W Friedman, Hawkeye King, Phillip Roan, Lei Cheng, Daniel Glozman, Ji Ma, Sina Nia Kosari, and Lee White. “Raven-II: An Open Platform for Surgical Robotics Research”. In: *IEEE Transactions on Biomedical Engineering* 60.4 (2012), pp. 954–959.

- [57] Jin Hu and Rong Xiong. “Contact Force Estimation for Robot Manipulator Using Semiparametric Model and Disturbance Kalman Filter”. In: *IEEE Transactions on Industrial Electronics* 65.4 (2018), pp. 3365–3375.
- [58] Yi Huang and Wenchao Xue. “Active Disturbance Rejection Control: Methodology and Theoretical Analysis”. In: *ISA Transactions* 53.4 (2014), pp. 963–976.
- [59] Satoshi Ito, Mohammad Darainy, Minoru Sasaki, and David J Ostry. “Computational Model of Motor Learning and Perceptual Change”. In: *Biological Cybernetics* 107 (2013), pp. 653–667.
- [60] David J Jacofsky and Mark Allen. “Robotics in Arthroplasty: A Comprehensive Review”. In: *The Journal of Arthroplasty* 31.10 (2016), pp. 2353–2363.
- [61] Hechao Ji, Shiqi Li, Jia Wang, and Zhao Ruan. “Improving Teleoperation Through Human-Aware Haptic Feedback: A Distinguishable and Interpretable Physical Interaction Based on the Contact State”. In: *IEEE Transactions on Human-Machine Systems* 53.1 (2022), pp. 24–34.
- [62] Rinde Johansson, Iara Santelices, Daniel O’Connell, Mahdi Tavakoli, and Daniel Aalto. “Evaluation of the Use of Haptic Virtual Fixtures to Guide Fibula Osteotomies in Mandible Reconstruction Surgery”. In: *2019 IEEE 15th International Conference on Automation Science and Engineering*. 2019.
- [63] Ankur Kapoor, Ming Li, and Russell H Taylor. “Constrained Control for Surgical Assistant Robots”. In: *ICRA*. 2006, pp. 231–236.
- [64] K Karthik, T Colegate-Stone, Prokar Dasgupta, Adel Tavakkolizadeh, and Joydeep Sinha. “Robotic Surgery in Trauma and Orthopaedics: A Systematic Review”. In: *The Bone & Joint Journal* 97.3 (2015), pp. 292–299.
- [65] Ryo Kikuuwe, Naoyuki Takesue, and Hideo Fujimoto. “A Control Framework to Generate Nonenergy-Storing Virtual Fixtures: Use of Simulated Plasticity”. In: *IEEE Transactions on Robotics* 24.4 (2008), pp. 781–793.
- [66] Warren Kilby, Michael Naylor, John R Dooley, Calvin R Maurer Jr, and Sohail Sayeh. “A Technical Overview of the CyberKnife System”. In: *Handbook of Robotic and Image-Guided Surgery* (2020), pp. 15–38.
- [67] Norihiro Koizumi, Shin’ichi Warisawa, Mitsuru Nagoshi, Hiroyuki Hashizume, and Mamoru Mitsuishi. “Construction Methodology for a Remote Ultrasound Diagnostic System”. In: *IEEE Transactions on Robotics* 25.3 (2009), pp. 522–538.
- [68] Peter Konrad. “The ABC of EMG”. In: *A Practical Introduction to Kinesiological Electromyography* 1.2005 (2005), pp. 30–5.
- [69] Andrew P Kurmis and Jamie R Ianunzio. “Artificial Intelligence in Orthopedic Surgery: Evolution, Current State and Future Directions”. In: *Arthroplasty* 4.1 (2022), pp. 1–10.
- [70] JE Lang, S Mannava, AJ Floyd, MS Goddard, BP Smith, A Mofidi, T M Seyler, and RH Jinnah. “Robotic Systems in Orthopaedic Surgery”. In: *The Journal of Bone and Joint Surgery. British volume* 93.10 (2011), pp. 1296–1299.

- [71] Leng-Feng Lee, Madusudanan S Narayanan, Frank Mendel, Venkat N Krovi, and Paul Karam. “Kinematics Analysis of In-Parallel 5 DOF Haptic Device”. In: *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2010, pp. 237–241.
- [72] Seth S Leopold. “Minimally Invasive Total Knee Arthroplasty for Osteoarthritis”. In: *New England Journal of Medicine* 360.17 (2009), pp. 1749–1758.
- [73] Ming Li and Allison M Okamura. “Recognition of Operator Motions for Real-Time Assistance Using Virtual Fixtures”. In: *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003*. IEEE. 2003, pp. 125–131.
- [74] Teng Li, Armin Badre, Farshid Alambeigi, and Mahdi Tavakoli. “Robotic Systems and Navigation Techniques in Orthopedics: A Historical Review”. In: *Applied Sciences* 13.17 (2023), p. 9768.
- [75] Teng Li, Armin Badre, Hamid D. Taghirad, and Mahdi Tavakoli. “Integrating Impedance Control and Nonlinear Disturbance Observer for Robot-Assisted Arthroscope Control in Elbow Arthroscopic Surgery”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 11172–11179.
- [76] Teng Li, Armin Badre, Hamid D Taghirad, and Mahdi Tavakoli. “Neural Network Learning of Robot Dynamic Uncertainties and Observer-based External Disturbance Estimation for Impedance Control”. In: *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2023, pp. 591–597.
- [77] Teng Li, Armin Badre, Hamid D Taghirad, and Mahdi Tavakoli. “Point-Based 3D Virtual Fixture Generating for Image-Guided and Robot-Assisted Surgery in Orthopedics”. In: *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2023, pp. 179–186.
- [78] Teng Li, Hongjun Xing, Ehsan Hashemi, Hamid D Taghirad, and Mahdi Tavakoli. “A Brief Survey of Observers for Disturbance Estimation and Compensation”. In: *Robotica* 41.12 (2023), pp. 3818–3845.
- [79] Teng Li, Amir Zakerimanesh, Yafei Ou, and Mahdi Tavakoli. “Iterative Learning for Gravity Compensation in Impedance Control”. In: *IEEE/ASME Transactions on Mechatronics* (2024). [Accepted], pp. 1–12.
- [80] Sichao Liu, Lihui Wang, and Xi Vincent Wang. “Sensorless Force Estimation for Industrial Robots Using Disturbance Observer and Neural Learning of Friction Approximation”. In: *Robotics and Computer-Integrated Manufacturing* 71 (2021), p. 102168.
- [81] Xing Liu, Fei Zhao, Shuzhi Sam Ge, Yuqiang Wu, and Xuesong Mei. “End-Effector Force Estimation for Flexible-Joint Robots With Global Friction Approximation Using Neural Networks”. In: *IEEE Transactions on Industrial Informatics* 15.3 (2018), pp. 1730–1741.

- [82] Jess H Lonner. *Robotics in Knee and Hip Arthroplasty: Current Concepts, Techniques and Emerging Uses*. Springer, 2019.
- [83] Dylan P Losey, Craig G McDonald, Edoardo Battaglia, and Marcia K O'Malley. "A Review of Intent Detection, Arbitration, and Communication Aspects of Shared Control for Physical Human-Robot Interaction". In: *Applied Mechanics Reviews* 70.1 (2018).
- [84] A de Luea and S Panzieri. "An Iterative Scheme for Learning Gravity Compensation in Flexible Robot Arms". In: *IFAC Proceedings Volumes* 26.2 (1993), pp. 575–582.
- [85] Justin H Ma, Shahriar Sefati, Russell H Taylor, and Mehran Armand. "An Active Steering Hand-Held Robotic System for Minimally Invasive Orthopaedic Surgery Using a Continuum Manipulator". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1622–1629.
- [86] Murilo M Marinho, Hisashi Ishida, Kanako Harada, Kyoichi Deie, and Mamoru Mitsuishi. "Virtual Fixture Assistance for Suturing in Robot-Aided Pediatric Endoscopic Surgery". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 524–531.
- [87] Murilo Marques Marinho, Bruno Vilhena Adorno, Kanako Harada, and Mamoru Mitsuishi. "Dynamic Active Constraints for Surgical Robots Using Vector-Field Inequalities". In: *IEEE Transactions on Robotics* 35.5 (2019), pp. 1166–1185.
- [88] Colin P McDonald, James A Johnson, Terry M Peters, and Graham JW King. "Image-Based Navigation Improves the Positioning of the Humeral Component in Total Elbow Arthroplasty". In: *Journal of Shoulder and Elbow Surgery* 19.4 (2010), pp. 533–543.
- [89] Jake Michael McDonnell, Daniel P Ahern, Tiarnan Ó Doinn, Denys Gibbons, Katharina Nagassima Rodrigues, Nick Birch, and Joseph S Butler. "Surgeon Proficiency in Robot-Assisted Spine Surgery: A Narrative Review". In: *The Bone & Joint Journal* 102.5 (2020), pp. 568–572.
- [90] Ben Mitchell, John Koo, Iulian Iordachita, Peter Kazanzides, Ankur Kapoor, James Handa, Gregory Hager, and Russell Taylor. "Development and Application of a New Steady-Hand Manipulator for Retinal Surgery". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2007, pp. 623–629.
- [91] Mamoru Mitsuishi, Shin'ichi Warisawa, Taishi Tsuda, Takuya Higuchi, Norihiro Koizumi, Hiroyuki Hashizume, and Kazuo Fujiwara. "Remote Ultrasound Diagnostic System". In: *2001 IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 2. IEEE. 2001, pp. 1567–1574.
- [92] Rocco Moccia, Cristina Iacono, Bruno Siciliano, and Fanny Ficuciello. "Vision-Based Dynamic Virtual Fixtures for Tools Collision Avoidance in Robotic Surgery". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1650–1655.

- [93] Alireza Mohammadi, Horacio J Marquez, and Mahdi Tavakoli. “Nonlinear Disturbance Observers: Design and Applications to Euler-Lagrange Systems”. In: *IEEE Control Systems Magazine* 37.4 (2017), pp. 50–72.
- [94] Alireza Mohammadi, Mahdi Tavakoli, Horacio J Marquez, and F Hashemzadeh. “Nonlinear Disturbance Observer Design for Robotic Manipulators”. In: *Control Engineering Practice* 21.3 (2013), pp. 253–267.
- [95] Phani Teja Nallamothu and Jasmin Praful Bharadiya. “Artificial Intelligence in Orthopedics: A Concise Review”. In: *Asian Journal of Orthopaedic Research* 9.1 (2023), pp. 17–27.
- [96] Sina Nia Kosari, Fredrik Rydén, Thomas S Lendvay, Blake Hannaford, and Howard Jay Chizeck. “Forbidden Region Virtual Fixtures from Streaming Point Clouds”. In: *Advanced Robotics* 28.22 (2014), pp. 1507–1518.
- [97] Christian Ott, Ranjan Mukherjee, and Yoshihiko Nakamura. “Unified Impedance and Admittance Control”. In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2010, pp. 554–561.
- [98] Vikas Panwar and Nagarajan Sukavanam. “Design of Optimal Hybrid Position/Force Controller for a Robot Manipulator using Neural Networks”. In: *Mathematical Problems in Engineering* 2007 (2007).
- [99] Shinsuk Park, Robert D Howe, and David F Torchiana. “Virtual Fixtures for Robotic Cardiac Surgery”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2001, pp. 1419–1420.
- [100] Itay Perets, Brian H Mu, Michael A Mont, Gurion Rivkin, Leonid Kandel, and Benjamin G Domb. “Current Topics in Robotic-Assisted Total Hip Arthroplasty: A Review”. In: *Hip International* 30.2 (2020), pp. 118–124.
- [101] Luka Peternel, Nikos Tsagarakis, Darwin Caldwell, and Arash Ajoudani. “Robot Adaptation to Human Physical Fatigue in Human-Robot Co-Manipulation”. In: *Autonomous Robots* 42.5 (2018), pp. 1011–1021.
- [102] Frederic Picard, Angela Helen Deakin, Philip E Riches, Kamal Deep, and Joseph Baines. “Computer Assisted Orthopaedic Surgery: Past, Present and Future”. In: *Medical Engineering & Physics* 72 (2019), pp. 55–65.
- [103] François Pierrot, Etienne Dombre, Eric Dégoulange, Loïc Urbain, Pierre Caron, Sylvie Boudet, Jérôme Gariépy, and Jean-Louis Ménégnien. “Hippocrate: A Safe Robot Arm for Medical Applications With Force Feedback”. In: *Medical Image Analysis* 3.3 (1999), pp. 285–300.
- [104] Alan M Priester, Shyam Natarajan, and Martin O Culjat. “Robotic Ultrasound Systems in Medicine”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 60.3 (2013), pp. 507–523.
- [105] F Pugin, P Bucher, and Philippe Morel. “History of Robotic Surgery: From AESOP® and ZEUS® to da Vinci®”. In: *Journal of Visceral Surgery* 148.5 (2011), e3–e8.

- [106] David Putzer, Dietmar Dammerer, Martina Baldauf, Florian Lenze, Michael C Liebensteiner, and Michael Nogler. “A Prospective Assessment of Knee Arthroscopy Skills Between Medical Students and Residents—Simulator Exercises for Partial Meniscectomy and Analysis of Learning Curves”. In: *Surgical Innovation* 29.3 (2022), pp. 398–405.
- [107] Aaron Radke and Zhiqiang Gao. “A Survey of State and Disturbance Observers for Practitioners”. In: *2006 American Control Conference*. IEEE. 2006, pp. 5183–5188.
- [108] Rodolfo Reda, Alessio Zanza, Andrea Cicconetti, Shilpa Bhandi, Gabriele Miccoli, Gianluca Gambarini, and Dario Di Nardo. “Ultrasound Imaging in Dentistry: A Literature Overview”. In: *Journal of Imaging* 7.11 (2021), p. 238.
- [109] Tianyu Ren, Yunfei Dong, Dan Wu, and Ken Chen. “Collision Detection and Identification for Robot Manipulators Based on Extended State Observer”. In: *Control Engineering Practice* 79 (2018), pp. 144–153.
- [110] Louis B Rosenberg. “Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation”. In: *Proceedings of IEEE Virtual Reality Annual International Symposium*. IEEE. 1993, pp. 76–82.
- [111] Fredrik Ryden and Howard Jay Chizeck. “A Proxy Method for Real-Time 3-DOF Haptic Rendering of Streaming Point Cloud Data”. In: *IEEE Transactions on Haptics* 6.3 (2013), pp. 257–267.
- [112] Fredrik Ryden and Howard Jay Chizeck. “Forbidden-Region Virtual Fixtures From Streaming Point Clouds: Remotely Touching and Protecting a Beating Heart”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2012, pp. 3308–3313.
- [113] Septimiu E Salcudean, Gordon Bell, Simon Bachmann, WH Zhu, P Abolmaesumi, and PD Lawrence. “Robot-Assisted Diagnostic Ultrasound—Design and Feasibility Experiments”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 1999, pp. 1062–1071.
- [114] Septimiu E Salcudean, Wen Hong Zhu, P Abolmaesumi, Simon Bachmann, and Peter D Lawrence. “A Robot System for Medical Ultrasound”. In: *Robotics Research*. Springer, 2000, pp. 195–202.
- [115] Dominique Saragaglia. “More Than 20 Years Navigation of Knee Surgery With the Orthopilot Device”. In: *Handbook of Robotic and Image-Guided Surgery*. Elsevier, 2020, pp. 425–441.
- [116] Christopher Schindlbeck and Sami Haddadin. “Unified Passivity-Based Cartesian Force/Impedance Control for Rigid and Flexible Joint Robots via Task-Energy Tanks”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 440–447.

- [117] Gijo Sebastian, Zeyu Li, Vincent Crocher, Demy Kremers, Ying Tan, and Denny Oetomo. "Interaction Force Estimation Using Extended State Observers: An Application to Impedance-Based Assistive and Rehabilitation Robotics". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1156–1161.
- [118] Shahriar Sefati, Rachel Hegeman, Farshid Alambeigi, Iulian Iordachita, Peter Kazanzides, Harpal Khanuja, Russell H Taylor, and Mehran Armand. "A Surgical Robotic System for Treatment of Pelvic Osteolysis Using an FBG-Equipped Continuum Manipulator and Flexible Instruments". In: *IEEE/ASME Transactions on Mechatronics* 26.1 (2020), pp. 369–380.
- [119] Shahriar Sefati, Rachel Hegeman, Iulian Iordachita, Russell H Taylor, and Mehran Armand. "A Dexterous Robotic System for Autonomous Debridement of Osteolytic Bone Lesions in Confined Spaces: Human Cadaver Studies". In: *IEEE Transactions on Robotics* 38.2 (2021), pp. 1213–1229.
- [120] Mario Selvaggio, Fei Chen, Boyang Gao, Gennaro Notomista, Francesco Trapani, and Darwin Caldwell. "Vision Based Virtual Fixture Generation for Teleoperated Robotic Manipulation". In: *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE. 2016, pp. 190–195.
- [121] Mario Selvaggio, Giuseppe Andrea Fontanelli, Fanny Ficuciello, Luigi Villani, and Bruno Siciliano. "Passive Virtual Fixtures Adaptation in Minimally Invasive Robotic Surgery". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3129–3136.
- [122] Alexander Sendrowicz, Marta Scali, Costanza Culmone, and Paul Breedveld. "Surgical Drilling of Curved Holes in Bone—A Patent Review". In: *Expert Review of Medical Devices* 16.4 (2019), pp. 287–298.
- [123] Naseem Shah, Nikhil Bansal, and Ajay Logani. "Recent Advances in Imaging Technologies in Dentistry". In: *World Journal of Radiology* 6.10 (2014), p. 794.
- [124] Mojtaba Sharifi, Javad K Mehr, Vivian K Mushahwar, and Mahdi Tavakoli. "Adaptive CPG-Based Gait Planning With Learning-Based Torque Estimation and Control for Exoskeletons". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 8261–8268.
- [125] Susheela Sharma, Tarunraj G Mohanraj, Jordan P Amadio, Mohsen Khadem, and Farshid Alambeigi. "A Concentric Tube Steerable Drilling Robot for Minimally Invasive Spinal Fixation of Osteoporotic Vertebrae". In: *IEEE Transactions on Biomedical Engineering* (2023), pp. 1–11.
- [126] Susheela Sharma, Yuewan Sun, Sarah Go, Jordan P Amadio, Mohsen Khadem, Amir Hossein Eskandari, and Farshid Alambeigi. "Towards Biomechanics-Aware Design of a Steerable Drilling Robot for Spinal Fixation Procedures with Flexible Pedicle Screws". In: *2023 International Symposium on Medical Robotics (ISMR)*. IEEE. 2023, pp. 1–6.
- [127] Jobe Shatrov and David Parker. "Computer and Robotic-Assisted Total Knee Arthroplasty: A Review of Outcomes". In: *Journal of Experimental Orthopaedics* 7 (2020), pp. 1–15.

- [128] Atsuo Shigi, Kunihiro Oka, Hiroyuki Tanaka, Shingo Abe, Satoshi Miyamura, Masaki Takao, Tatsuo Mae, Hideki Yoshikawa, and Tsuyoshi Murase. “Validation of the Registration Accuracy of Navigation-Assisted Arthroscopic Debridement for Elbow Osteoarthritis”. In: *Journal of Shoulder and Elbow Surgery* 28.12 (2019), pp. 2400–2408.
- [129] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. “Robotics: Modelling, Planning and Control”. In: Springer, 2010.
- [130] Jean-Jacques E Slotine and Weiping Li. “On the Adaptive Control of Robot Manipulators”. In: *The International Journal of Robotics Research* 6.3 (1987), pp. 49–59.
- [131] N Smith-Guerin, L Al Bassit, G Poisson, C Delgorge, Ph Arbeille, and P Vieyres. “Clinical Validation of a Mobile Patient-Expert Tele-Echography System Using ISDN Lines”. In: *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, 2003*. IEEE. 2003, pp. 23–26.
- [132] Abed Soleymani, Ali Torabi, and Mahdi Tavakoli. “A Low-Cost Intrinsically Safe Mechanism for Physical Distancing Between Clinicians and Patients”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1–7.
- [133] Peng Song, Yueqing Yu, and Xuping Zhang. “A Tutorial Survey and Comparison of Impedance Control on Robotic Manipulation”. In: *Robotica* 37.5 (2019), pp. 801–836.
- [134] Jean-Pierre St Mart, En Lin Goh, and Zameer Shah. “Robotics in Total Hip Arthroplasty: A Review of the Evolution, Application and Evidence Base”. In: *EFORT Open Reviews* 5.12 (2020), p. 866.
- [135] Taylor P Stauffer, Billy I Kim, Caitlin Grant, Samuel B Adams, and Albert T Anastasio. “Robotic Technology in Foot and Ankle Surgery: A Comprehensive Review”. In: *Sensors* 23.2 (2023), p. 686.
- [136] Nobuhiko Sugano. “Computer-Assisted Orthopaedic Surgery and Robotic Surgery in Total Hip Arthroplasty”. In: *Clinics in Orthopedic Surgery* 5.1 (2013), pp. 1–9.
- [137] Amir M Tahmasebi, Babak Taati, Farid Mobasser, and Keyvan Hashtrudi-Zaad. “Dynamic Parameter Identification and Analysis of a PHANToM Haptic Device”. In: *Proceedings of 2005 IEEE Conference on Control Applications (CCA)*. IEEE. 2005, pp. 1251–1256.
- [138] Ran Tao, Renz Ocampo, Jason Fong, Abed Soleymani, and Mahdi Tavakoli. “Modeling and Emulating a Physiotherapist’s Role in Robot-Assisted Rehabilitation”. In: *Advanced Intelligent Systems* 2.7 (2020), p. 1900181.
- [139] Russell Taylor, Pat Jensen, Louis Whitcomb, Aaron Barnes, Rajesh Kumar, Dan Stoianovici, Puneet Gupta, ZhengXian Wang, Eugene Dejuan, and Louis Kavoussi. “A Steady-Hand Robotic System for Microsurgical Augmentation”. In: *The International Journal of Robotics Research* 18.12 (1999), pp. 1201–1210.

- [140] Russell H. Taylor, Arianna Menciassi, Gabor Fichtinger, Paolo Fiorini, and Paolo Dario. “Medical Robotics and Computer-Integrated Surgery”. In: *Springer Handbook of Robotics*. Ed. by Bruno Siciliano and Oussama Khatib. Cham: Springer International Publishing, 2016, pp. 1657–1684.
- [141] Ali Torabi, Mohsen Khadem, Kourosh Zareinia, Garnette Roy Sutherland, and Mahdi Tavakoli. “Application of a Redundant Haptic Interface in Enhancing Soft-Tissue Stiffness Discrimination”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1037–1044.
- [142] Tomer Valency and Miriam Zacksenhouse. “Accuracy/Robustness Dilemma in Impedance Control”. In: *Journal of Dynamic Systems, Measurement, and Control* 125.3 (2003), pp. 310–319.
- [143] Valentina Vitiello, Su-Lin Lee, Thomas P Cundy, and Guang-Zhong Yang. “Emerging Robotic Platforms for Minimally Invasive Surgery”. In: *IEEE Reviews in Biomedical Engineering* 6 (2012), pp. 111–126.
- [144] Hong-Rui Wang, Li Yang, and Li-Xin Wei. “Fuzzy-Neuro Position/Force Control for Robotic Manipulators with Uncertainties”. In: *Soft Computing* 11 (2007), pp. 311–315.
- [145] Monan Wang, Donghui Li, Xiping Shang, and Jian Wang. “A Review of Computer-Assisted Orthopaedic Surgery Systems”. In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 16.5 (2020), pp. 1–28.
- [146] Yan Wang, Hoi-Wut Yip, Hao Zheng, Hongbin Lin, Russell H Taylor, and Kwok Wai Samuel Au. “Design and Experimental Validation of a Miniaturized Robotic Tendon-Driven Articulated Surgical Drill for Enhancing Distal Dexterity in Minimally Invasive Spine Fusion”. In: *IEEE/ASME Transactions on Mechatronics* 26.4 (2021), pp. 1858–1866.
- [147] Yan Wang, Hao Zheng, Russell H Taylor, and Kwok Wai Samuel Au. “A Handheld Steerable Surgical Drill With a Novel Miniaturized Articulated Joint Module for Dexterous Confined-Space Bone Work”. In: *IEEE Transactions on Biomedical Engineering* 69.9 (2022), pp. 2926–2934.
- [148] Marcel K Welleweerd, Antonius Gerardus de Groot, SOH de Looijer, Françoise J Siepel, and Stefano Stramigioli. “Automated Robotic Breast Ultrasound Acquisition Using Ultrasound Feedback”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 9946–9952.
- [149] Holger Wendland. “Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree”. In: *Advances in Computational Mathematics* 4.1 (1995), pp. 389–396.
- [150] Tian Xia, Ankur Kapoor, Peter Kazanzides, and Russell Taylor. “A Constrained Optimization Approach to Virtual Fixtures for Multi-Robot Collaborative Teleoperation”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2011, pp. 639–644.

- [151] Hongjun Xing, Ali Torabi, Liang Ding, Haibo Gao, Zongquan Deng, Vivian K. Mushahwar, and Mahdi Tavakoli. “An Admittance-Controlled Wheeled Mobile Manipulator for Mobility Assistance: Human–Robot Interaction Estimation and Redundancy Resolution for Enhanced Force Exertion Ability”. In: *Mechatronics* 74 (2021), p. 102497.
- [152] Shirin Yousefizadeh and Thomas Bak. “Nonlinear Disturbance Observer for External Force Estimation in a Cooperative Robot”. In: *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE. 2019, pp. 220–226.
- [153] Ajmal Zemmar, Andres M Lozano, and Bradley J Nelson. “The Rise of Robots in Surgical Environments During COVID-19”. In: *Nature Machine Intelligence* 2.10 (2020), pp. 566–572.
- [154] Tie Zhang and Xiaohong Liang. “Disturbance Observer-Based Robot End Constant Contact Force-Tracking Control”. In: *Complexity* 2019 (2019).
- [155] Craig B Zilles and J Kenneth Salisbury. “A Constraint-Based God-Object Method for Haptic Display”. In: *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Human Robot Interaction and Cooperative Robots*. Vol. 3. IEEE. 1995, pp. 146–151.

Appendix A: EMG-based Hybrid Impedance-Force Control for Ultrasound Imaging¹

Ultrasound (US) imaging is a common but physically demanding task in the medical field, and sonographers may need to put in considerable physical effort for producing high-quality US images. During physical human-robot interaction on US imaging, robot compliance is a critical feature that can ensure human user safety while automatic force regulation ability can help to improve task performance. However, higher robot compliance may mean lower force regulation accuracy, and vice versa. Especially, the contact/non-contact status transition can largely affect the control system stability. In this work, a novel electromyography (EMG)-based hybrid impedance-force control system is developed for US imaging task. The proposed control system incorporates the robot compliance and force regulation ability via a hybrid controller while the EMG channel enables the user to online modulate the trade-off between the two features as necessary. Two experiments are conducted to examine the hybrid controller and show the necessity of involving an EMG-based modulator. A proof-of-concept study on US imaging is performed with implementing the proposed EMG-based control system, and the effectiveness is demonstrated. The proposed control system is promising to ensure robot's stability and patient's safety, thus obtain high-quality US images, while monitoring and reducing sonographer's fatigue. Fur-

¹A version of this chapter has been published as: Teng Li, Hongjun Xing, Hamid D. Taghirad, Mahdi Tavakoli, "EMG-based Hybrid Impedance-Force Control for Human-Robot Collaboration on Ultrasound Imaging", 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022), October 23-27, Kyoto, Japan, 2022. pp. 670-675. [\[IEEE Xplore\]](#)

thermore, it can be easily adapted to other physically demanding tasks in the field of medicine.

A.1 Introduction

During human-human collaboration, *e.g.*, lifting and moving a heavy box together, one naturally assumes to be the leader, while the other be the follower [83]. Aiming for good performance on a collaborated task, the follower is required to be able to shift between “compliant” and “rigid” behaviors whenever necessary according to the leader’s intention. This is also true for human-robot collaboration when the human is the leader, in which case the robot is expected to be able to understand the user’s intention and adapt its compliance level in real-time according to the requirement of the task.

Ultrasound (US) imaging task is conducted by sonographer in a way that manually holding and moving the US probe on patient’s body or target tissue. This procedure demands considerable physical effort from the sonographer due to multi-tasking requirement, *e.g.*, regulating the probe-tissue contact force while moving the probe along a trajectory. Various robot-assisted US imaging methods [104] have been developed aiming both to reduce the sonographer’s physical effort and to improve the task performance, *i.e.*, acquiring high-quality scanning images.

Robot teleoperation has been used for US imaging for a long time. Two decades ago, Mitsuishi *et al.* [91] developed a remote US diagnostic system. The distance between the user interface and teleoperated manipulator holding the US probe was about 700 km. Conti *et al.* [23] presented a new teleoperation robotic system assisting sonographers to conduct US imaging task aiming to reduce physical fatigue and better interpretation of US imaging data.

Apart from teleoperation systems, physical human-robot interaction (*pHRI*) for a collaborative US imaging task is very beneficial to sonographers. Carriere *et al.* [15] designed an admittance-controlled semi-autonomous system for US imaging. Their

system enabled the robot to automatically control the US probe's orientation and the probe-tissue contact force, while the user controls the lateral position of the probe on the patient's body. In their system, 3D reconstruction technique is used to model the tissue surface.

To obtain compliant behavior from a robot, an impedance or admittance controller may be employed [83]. In *pHRI* applications, the robot can be controlled to be soft (compliant) or rigid (non-compliant) based on identified human intention and task requirements [73]. One main advantage of impedance control is its potentially better compliant robot behavior compared to admittance control. Another advantage is that the measurement or estimation on human-robot interaction force is not necessary for impedance control whereas it is indispensable for admittance control. On the other hand, implementing an impedance controller is usually more complex than an admittance controller because it usually requires full knowledge of the robot dynamics and accurately identified dynamic parameters [15].

The stable and appropriate normal contact force between the US probe and the tissue during scanning is one of the most important factors that can guarantee the US image quality [23, 15]. Different exam types may need different desired contact force range [131]. The requirement on accurately regulating the contact force into a desired range is a major reason that induces the sonographer's fatigue which could further affect image quality and even patient's safety. A force tracking controller can allow a robot to track or regulate the robot-environment interaction force in an autonomous manner, which could effectively help the sonographer to do the force regulation during US image scanning process [15].

Electromyography (EMG) is increasingly incorporated into robot control systems for better interpreting human intention and enhancing *pHRI*, because it can be more easily measured than some other physiological signals like electroencephalography (EEG) and electrocardiography (ECG). The concept of teleimpedance was first introduced in a work done by Ajoudani *et al.* [4], where the EMG measured from the

human arm was used to regulate the robot impedance in real-time. By using EMG signal, the robot impedance was modified as needed in different phases of a *pHRI* task like peg-in-hole insertion. EMG was also used for online monitoring the user's fatigue during a *pHRI* task such that the robot could adapt itself to take over more physical work and allow the human partner to have some rest [101].

Contact/non-contact status switching is commonly encountered during US imaging especially at the start/end phase of the task. A critical issue during the status transition is that it can adversely affect the system stability thus patient's safety. Aiming to incorporate robot compliance and force regulation ability together while ensuring robot's stability especially during contact/non-contact status transition, in this work, an EMG-based hybrid impedance-force control system is developed as shown in Figure A.1. The proposed system incorporates advantages of compliant robot behavior coming from an impedance controller and accurate force regulation ability coming from a force controller, while the EMG signal is used as a modulator which enables the human user to tune the trade-off between robot compliance and force regulation ability in an online manner. The effectiveness of the proposed control system is evaluated by a preliminary application on human-robot collaborated US imaging task.

A.2 Methods

A.2.1 Impedance control and force control

The general dynamic model for an n -degree-of-freedom (DOF) rigid robot [41] may be expressed as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J}^T \mathbf{F}_{\text{ext}} \quad (\text{A.1})$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ denotes the inertia matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes a matrix related to the Coriolis and centrifugal forces, $\mathbf{g} \in \mathbb{R}^n$ represents a vector related to gravity, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the commanded joint torque vector, $\mathbf{F}_{\text{ext}} \in \mathbb{R}^6$ is external force in Cartesian space,

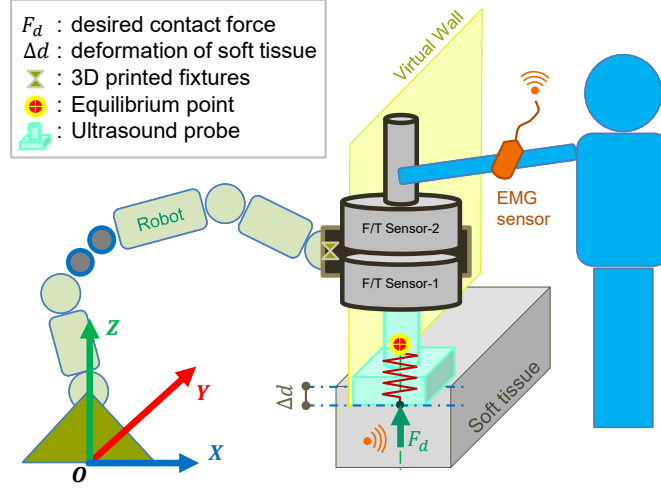


Figure A.1: Schematic setup of an EMG-based hybrid impedance-force control system for human-robot collaboration on Ultrasound imaging task.

and $\mathbf{J} \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix. A full impedance model [151] can be given by

$$\mathbf{F}_{\text{imp}} = \mathbf{M}_m(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d) + \mathbf{D}_m(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}_m(\mathbf{x} - \mathbf{x}_d) \quad (\text{A.2})$$

where $\mathbf{M}_m, \mathbf{D}_m, \mathbf{K}_m$ are the user-defined matrices for inertia, damping, and stiffness, respectively. $\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d$ are the desired position, velocity, and acceleration, respectively in Cartesian space, while $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$ are the actual position, velocity, and acceleration, respectively. $\mathbf{F}_{\text{imp}} \in \mathbb{R}^6$ is the contact wrench (force and torque) between the robot end-effector (EE) and the environment in Cartesian space.

To avoid external force measurement, we set the desired inertia matrix equal to the natural inertia matrix of the robot, *i.e.*, $\mathbf{M}_m = \mathbf{M}_x$, where $\mathbf{M}_x$ is the natural inertia of the robot in Cartesian space, and $\mathbf{M}_x = \mathbf{J}^{-T} \mathbf{M} \mathbf{J}^{-1}$ [141]. In order to represent a real mechanical system, a Coriolis and centrifugal term should also be included into the impedance model (A.2). Then, the full impedance model is augmented as

$$\mathbf{F}_{\text{imp}} = \mathbf{M}_x(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d) + (\mathbf{S}_x + \mathbf{D}_m)(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}_m(\mathbf{x} - \mathbf{x}_d) \quad (\text{A.3})$$

where $\mathbf{S}_x$ is the Coriolis and centrifugal matrix of the robot in Cartesian space and $\mathbf{S}_x = \mathbf{J}^{-T} \mathbf{S} \mathbf{J}^{-1} - \dot{\mathbf{M}}_x \mathbf{J} \mathbf{J}^{-1}$. For set-point regulation problem, it has $\ddot{\mathbf{x}}_d = \dot{\mathbf{x}}_d = \mathbf{0}$.

Then by substituting (A.3) into (A.1) via $\mathbf{F}_{\text{ext}} = \mathbf{F}_{\text{imp}}$, a simplified impedance control law can be obtained as given by (A.4), which is also known as task-space PD controller with gravity compensation.

$$\boldsymbol{\tau}_{\text{imp}} = \mathbf{J}^T[\mathbf{K}_m(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}_m\dot{\mathbf{x}}] + \mathbf{g}, \quad (\text{A.4})$$

A general form of Cartesian-space force tracking controller [116] can be given by

$$\boldsymbol{\tau}_f = \mathbf{K}_p\mathbf{J}^T(\mathbf{F} - \mathbf{F}_d) + \mathbf{K}_i\mathbf{J}^T \int_0^t (\mathbf{F} - \mathbf{F}_d)dt + \mathbf{K}_d\mathbf{J}^T(\dot{\mathbf{F}} - \dot{\mathbf{F}}_d) \quad (\text{A.5})$$

where $\mathbf{K}_p, \mathbf{K}_i, \mathbf{K}_d \in \mathbb{R}^{n \times n}$ are the gain matrices of P-regulator, I-regulator, and D-regulator, respectively in the joint space, which need to be designed. $\mathbf{F}_d, \mathbf{F} \in \mathbb{R}^6$ are the desired and actual interaction force between the robot EE and the environment, respectively. For simplicity, a simplified PI force tracking controller is employed which is given by

$$\boldsymbol{\tau}_f = \mathbf{K}_p\mathbf{J}^T(\mathbf{F} - \mathbf{F}_d) + \mathbf{K}_i\mathbf{J}^T \int_0^t (\mathbf{F} - \mathbf{F}_d)dt \quad (\text{A.6})$$

where $\mathbf{F}$ is measured by an external Force/Torque (F/T) sensor, $\mathbf{K}_p = k_p\mathbf{I}$, $\mathbf{K}_i = k_i\mathbf{I}$, and $\mathbf{I}$ is an appropriate identity matrix. Theoretically, the P-regulator ($\mathbf{K}_p$) term can be viewed as a spring which reduces the force error between $\mathbf{F}$ and $\mathbf{F}_d$. The I-regulator ($\mathbf{K}_i$) term acts as a compensator which can compensate the possible steady state force error.

The block diagram of the proposed EMG-based hybrid impedance-force control system for human-robot collaboration task is shown in Figure A.2. EMG-related processing and mapping methods will be introduced subsequently.

A.2.2 EMG signal acquisition and processing

In this work, raw electromyography (EMG) signal from human user's arm (biceps brachii) [68] is collected and processed in real-time. A simple moving average (SMA) algorithm given by (A.7) is employed as the filter.

$$e_{\text{sma}} = \frac{1}{N} \sum_{n=1}^N e_{\text{raw}} \quad (\text{A.7})$$

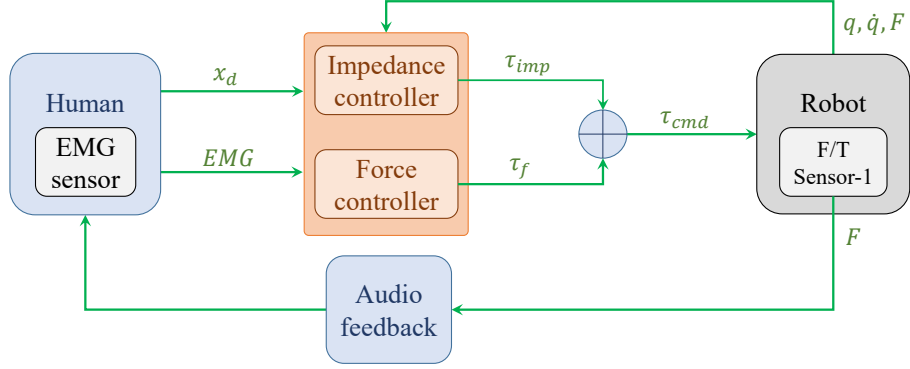


Figure A.2: Block diagram for the proposed EMG-based hybrid impedance-force control system.

where e_{sma} is the filtered EMG signal, N is the moving window size in units of sample points, e_{raw} is the raw EMG.

After passing through a filter, the filtered EMG signal e_{sma} is normalized into a range of $[0, 1]$ by $e_{\text{norm}} = e_{\text{sma}}/e_{\text{mvc}}$, via a user-specific parameter called maximum voluntary contraction (MVC). Therefore, the MVC needs to be calibrated for each user. The calibration procedure [68] is that, the user maximize his/her arm muscle effort for three times, then the average of the three maximum, denoted by e_{mvc} , will be taken as the MVC of this user.

A.2.3 EMG mapping algorithm

The normalized EMG signal e_{norm} is mapped to the P-regulator ($\mathbf{K}_p = k_p \mathbf{I}$ in (A.6)) in the hybrid impedance-force controller using the following mapping algorithm,

$$k_p = \begin{cases} (k_{p,\max} - k_{p,\min}) \times (1 - e_{\text{norm}}) & \text{if } e_{\text{norm}} \leq \sigma \\ k_{p,\min} & \text{if } e_{\text{norm}} > \sigma \end{cases} \quad (\text{A.8})$$

where $0 < \sigma < 1$, $k_{p,\min}$ and $k_{p,\max}$ represent the minimum and maximum force regulation ability of the robot, respectively. The transition between the two conditions in (A.8) is realized via a low-pass filter to ensure the smoothness.

The general idea of the mapping algorithm (A.8) is that, when the human user exerts a larger interaction force on the robot EE (detected by Sensor 2 in Figure A.1), the robot should become more compliant. When the user exert a large-enough force

($e_{\text{norm}} > \sigma$) on the robot EE, it indicates that the user intends to totally control the robot, thus the robot should provide the maximum compliance ($k_p = k_{p,\min}$). To the contrary, when the user relaxes his/her arm ($e_{\text{norm}} \leq \sigma$), the robot should assume control of the contact force between the probe and tissue for an accurate force regulation, such that the user can focus on other tasks.

A.2.4 Desired force and audio feedback design

As introduced in the Introduction, different purposes may require different desired force range for the contact force between the US probe and the tissue. Empirically, a desired range of 4.5 ± 1 N for the contact force is used in this work for pilot tests. Note that in the force controller, it is possible to set other constant or time-varying desired force.

Accordingly, audio feedback is provided to the user to indicate which range the current normal contact force is located in, *i.e.*, lower range $[-inf, 3.5]$ N, ideal range $[3.5, 5.5]$ N, or upper range $[5.5, inf]$ N. Continuous beep is provided via Arduino board to indicate the ideal range, while discontinuous fast beep is provided to indicate the upper range. Otherwise, no audio feedback is provided.

A.3 Experiments and Results

A.3.1 Apparatus

A 7-DOF Franka Emika Panda robot (Franka Emika GmbH, Munich, Germany) is employed for developing and evaluating the proposed EMG-based hybrid impedance-force control system for US imaging task as illustrated in Figure A.1. The hybrid controller is implemented via `libfranka`, the C++ implementation of the client side of the Franka Control Interface (FCI). The `libfranka` run with ROS control on a workstation computer of Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz $\times$ 6 with the Ubuntu 16.04 LTS (Xenial Xerus) 64-bit operating system. The control rate of the robot is 1 kHz.

A classical Arduino MEGA2560 (R3) board is used to collect the raw EMG signal from EMG sensor and then transmit it to the control system. Meanwhile, it also provides audio feedback to the user to indicate the real-time contact status between the US probe and the soft tissue. The EMG signal sent from Arduino to the robot controller is at 1 kHz.

In this work, the contact force between the US probe and the soft tissue is measured by a 6-DOF F/T sensor (Sensor 1 in Figure A.1, Axia80-M20-ZC22, ATI Industrial Automation, Inc., USA). Meanwhile, a second F/T sensor (Sensor 2 in Figure A.1) with exactly the same type is used to measure the external interaction force exerted by the human as an independent measurement to indicate the user effort. Please note that the data from Sensor 2 is only used for post-analysis and not used in the control system, and Sensor 2 can be removed in the future in order for a more compact system.

The main parameters used in the experiments are listed in Table A.1. A video demonstration for the experiments is available online¹.

Table A.1: Parameters for the experiments.

Parameter	Location	Experiment
$\mathbf{K}_m = \text{diag}\{10, 10, 10, 0, 0, 0\}$	Eqn. (A.4)	Exp.1,2,3
$\mathbf{D}_m = \text{diag}\{0, 0, 0, 0, 0, 0\}$	Eqn. (A.4)	Exp.1,2,3
$\mathbf{F}_d = [0, 0, 4.5, 0, 0, 0]^T$	Eqn. (A.6)	Exp.1,2,3
$k_p = 3; k_i = 0.5$	Eqn. (A.6)	Exp.1,3
$k_p = 0/1/3; k_i = 0.5$	Eqn. (A.6)	Exp.2
$k_{p,min} = 0; k_{p,max} = 3; \sigma = 0.5$	Eqn. (A.8)	Exp.2,3

Note: $\mathbf{K}_m, \mathbf{D}_m \in \mathbb{R}^{6 \times 6}$ are diagonal matrices. The desired contact force $\mathbf{F}_d$ is defined in the frame of Sensor 1, then transformed into the robot base frame. A preliminary test on k_i at three levels ($k_i = 0/0.2/0.5$) was conducted, then $k_i = 0.5$ was determined based on the optimal results. $\sigma = 0.5$ is determined for a balanced level of human user's arm muscle contraction.

¹online demo video link: <https://youtu.be/kgMYiFkA3qk>

A.3.2 Experiment 1: Hybrid impedance-force controller

Experiment 1 is designed to evaluate the hybrid impedance-force controller in two scenarios. The first scenario is US imaging on a rectangular soft tissue as shown in Figure A.3a, and the second scenario is on bowl-shaped soft tissue (representing human breast) as shown in Figure A.3c.

A user performs an US imaging task in the two scenarios separately. In each trial, the user moves the US probe on the surface of the soft tissue from one side to the other and then back. Three continuous trials compose as one session, and six sessions for each scenario. Note that, this experiment does not involve contact/non-contact status transition.

A typical sample data for each of the two scenarios are shown in Figure A.3b and Figure A.3d. The detailed results of Experiment 1 are summarized in Table A.2. These results show that the performance in rectangular scenario has no significant difference ($p = 0.2542$) with that in bowl-shaped scenario in terms of normal force regulation accuracy. However, the rectangular scenario has significantly more stable force regulation behavior ($p = 0.0097$) than the bowl-shaped scenario in terms of standard deviation. This is reasonable considering that the latter scenario involves more complex rotational movements while the former does not.

The results of Experiment 1 indicate that, the hybrid impedance-force controller is able to help the user regulate the probe-tissue contact force in both simple (rectangular) scenario and complex (bowl-shaped) scenario without significant difference in terms of average contact force. With the help of the hybrid controller, the robot will regulate the contact force while the user can focus on other tasks, *e.g.*, moving the probe on the tissue along a desired trajectory.

A.3.3 Experiment 2: Lifting task

During US imaging, contact/non-contact status transition, *e.g.*, moving the probe away from or onto the tissue surface, is a major factor that could affect the control

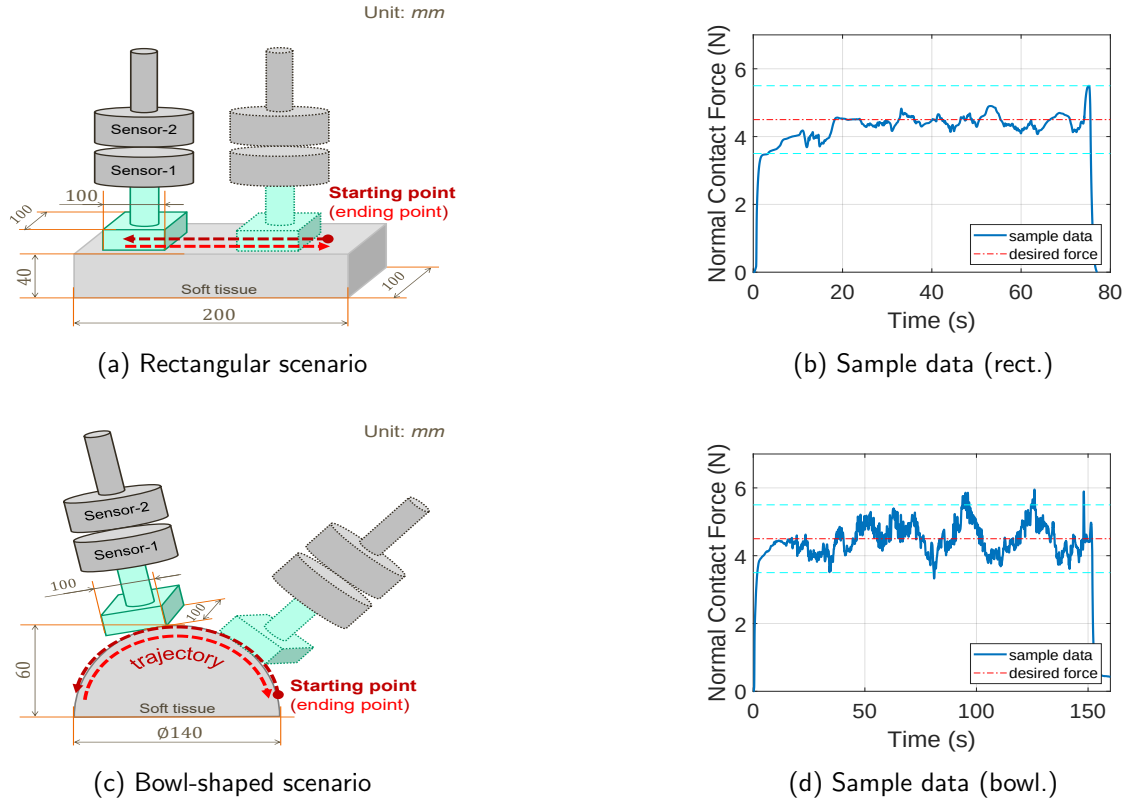


Figure A.3: Schematic illustration for the two scenarios in Experiment 1 and typical sample results. Two cyan dash lines indicate the predefined ideal range $[3.5, 5.5]$ N, while the red dash-dot line denotes the desired force 4.5 N.

Table A.2: Results of Experiment 1 on normal contact force in rectangular and bowl-shaped tissue scenarios.

	Mean (N)		std. (N)	
	Rect.	Bowl.	Rect.	Bowl.
s1	4.426	4.517	0.291	0.461
s2	4.446	4.397	0.245	0.581
s3	4.390	4.473	0.249	0.359
s4	4.304	4.431	0.369	0.458
s5	4.438	4.456	0.167	0.543
s6	4.472	4.432	0.293	0.430
	$p = 0.2542$		$p = 0.0097 (*)$	

Note: s1, means session-1; Rect., means rectangular tissue scenario; Bowl., means bowl-shaped tissue scenario; std., means standard deviation.

system stability, thus potentially endanger the patient’s safety. Therefore, it is necessary to examine the system stability and robot compliance when contact/non-contact status transition is involved. For this purpose, a lifting task is designed in Experiment 2 as illustrated in Figure A.4a. In each trial, the user needs to lift the US probe from one surface to a predefined height and then move onto another surface. During the task, the maximum force exerted by the user is measured by the F/T Sensor 2 and recorded as the user’s effort in each trial. There are three levels for the P-regulator are tested due to a mapping relationship between the EMG signal and the P-regulator in the final controller. Six trials for each level are conducted.

The results on the lifting task are shown in Figure A.4b. As can be seen in the figure, the user’s maximum efforts significantly increase as the increasing of the P-regulator. This means that with a lower level of k_p , the robot can provide better compliance, thus the user can easily lift and move the US probe. However, with a higher level of k_p , the robot can provide better force regulation accuracy but the user needs to make more effort to lift the US probe. More importantly, the latter case could easily trigger an unstable system due to the potentially large external force, *e.g.*, trigger an automatic emergency stop for the Panda robot.

The results of Experiment 2 indicate that, the P-regulator in the force controller is able to do a trade-off between the robot compliance and the force regulation ability. An EMG-based modulator will be introduced into the control system such that the trade-off can be tuned online by the user.

A.3.4 Experiment 3: Application

In Experiment 3, a proof-of-concept application study on the US imaging is conducted by implementing the proposed EMG-based hybrid impedance-force controller. The user’s arm EMG signal is mapped with the P-regulator of the force controller via the algorithm (A.8). This allows not only the robot to regulate the contact force autonomously, but also the user to modulate the robot compliance in an online manner.

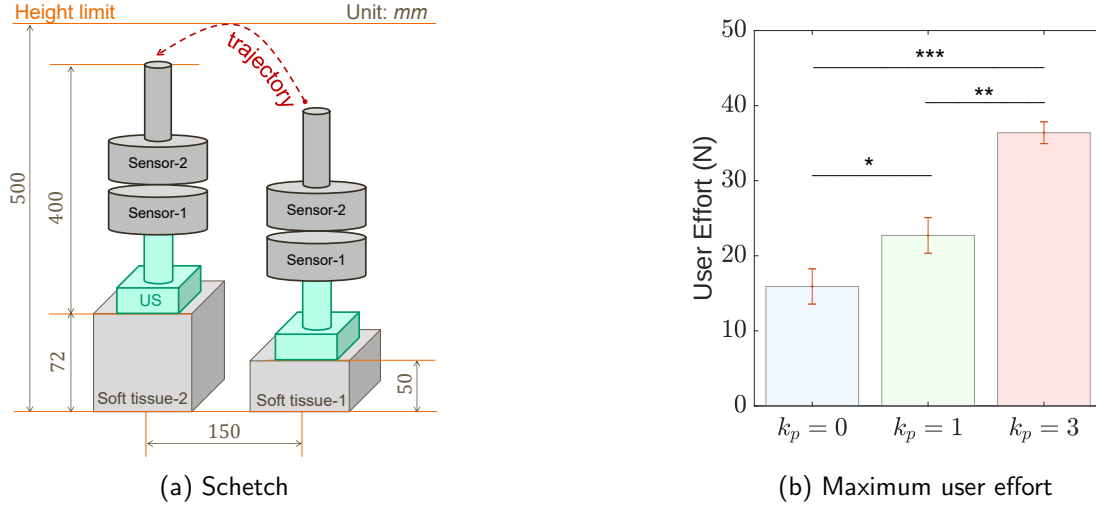


Figure A.4: Schematic illustration and results for the lifting task in Experiment 2. The maximum user effort are from three levels of the P-regulator in the PI force controller.

As shown in Figure A.5, a general US imaging task is designed in Experiment 3 to evaluate the effectiveness of the proposed EMG-based hybrid controller. The task includes a lifting sub-task that involves contact/non-contact status transition, and an US imaging sub-task on a complex tissue surface geometry which consists of a horizontal plane and an inclined slope plane. For each trial in the task, the user first needs to lift the US probe to reach a predefined height (the same height as that in Experiment 2), then puts it onto the complex tissue surface, and then moves it on the surface to the end and then back (see the red dash trajectory in the figure). To assist the user in moving the probe along the trajectory, a vertical virtual wall is set along the trajectory. Six separate sessions are conducted and each session includes only one trial. The user effort, *i.e.*, maximum lifting force, and normal contact force are recorded in each session.

A sample data for a typical session in Experiment 3 is presented in Figure A.6. As shown in the figure, the first colored area is for the lifting sub-task that involves contact/non-contact status transition while the second colored area is for the US imaging sub-task. The area between the two colored areas is a recovery phase in which the contact force will recovered to the desired level driven by the hybrid controller.

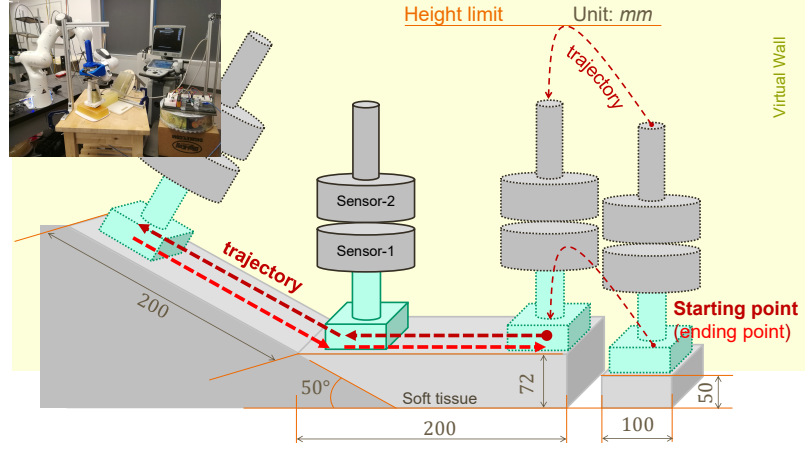


Figure A.5: Setup for the application in Experiment 3 with implementing the EMG-based hybrid impedance-force controller.

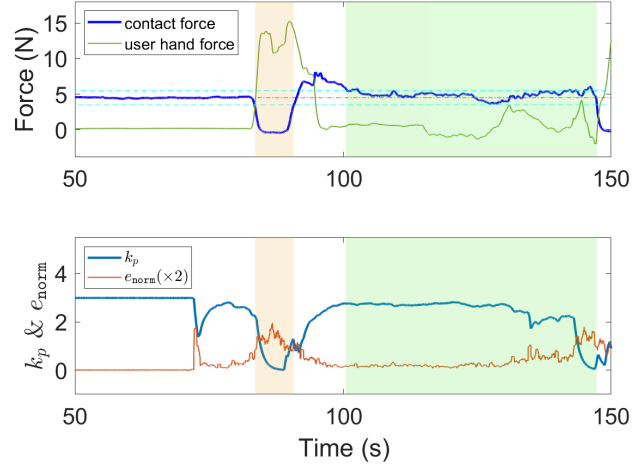


Figure A.6: Sample data for a typical trial in Experiment 3. The yellow colored area is for the lifting sub-task while the green colored area is for the US imaging sub-task.

The maximum user effort in the lifting sub-task and the normal contact force in the US imaging sub-task in each session are summarized in [Table A.3](#). As shown in the table and [Figure A.6](#), with the help of EMG-based modulator, the robot compliance can be tuned online as needed, and the user effort for lifting the probe is kept in a reasonable range, which ensure system stability and patient's safety.

A comparison is conducted between the results of Experiment 3 and the two scenarios in Experiment 1 as show in [Figure A.7](#). As can be seen in [Figure A.7a](#), there

is no significant difference in terms of the force regulation accuracy between the scenario in Experiment 3 and either of the rectangular and bowl-shaped tissue scenarios in Experiment 1. This indicates that the performance on force regulation accuracy in Experiment 3 is as good as that in the rectangular and bowl-shaped tissue scenario in Experiment 1. For the standard deviation as shown in Figure A.7b, however, there is significant difference ($p = 0.002$) between Experiment 3 and the rectangular scenario in Experiment 1 which indicates that the latter had a significantly more stable force regulation behavior. This is reasonable since the latter scenario has not involved complex rotations of the robot EE. There is no significant difference on the standard deviation ($p = 0.053$) between Experiment 3 and the bowl-shaped scenario in Experiment 1.

Table A.3: Results of Experiment 3 with the proposed EMG-based hybrid impedance-force controller.

	Normal contact force (N)		User effort (N)
	mean	std.	max
s1	4.848	0.663	16.319
s2	4.881	0.479	15.241
s3	4.615	0.764	14.807
s4	4.554	0.575	16.021
s5	4.379	0.725	16.092
s6	4.424	0.898	16.502

Note: s1, means session-1; std., means standard deviation.

One limitation of the proposed EMG-based method is that the EMG acquisition system needs to be calibrated for each individual in order to obtain the MVC, although the calibration procedure is simple. In the future work, machine learning algorithms will be employed to automatically identify the MVC online and on a user-specific basis. Another limitation is that normal contact force rather than acquired image is used as the metric to evaluate the proposed system in the present work. Although the contact force is a main indicator for obtaining high-quality US scanning images, directly evaluating the quality of the acquired scanning images will be a necessary

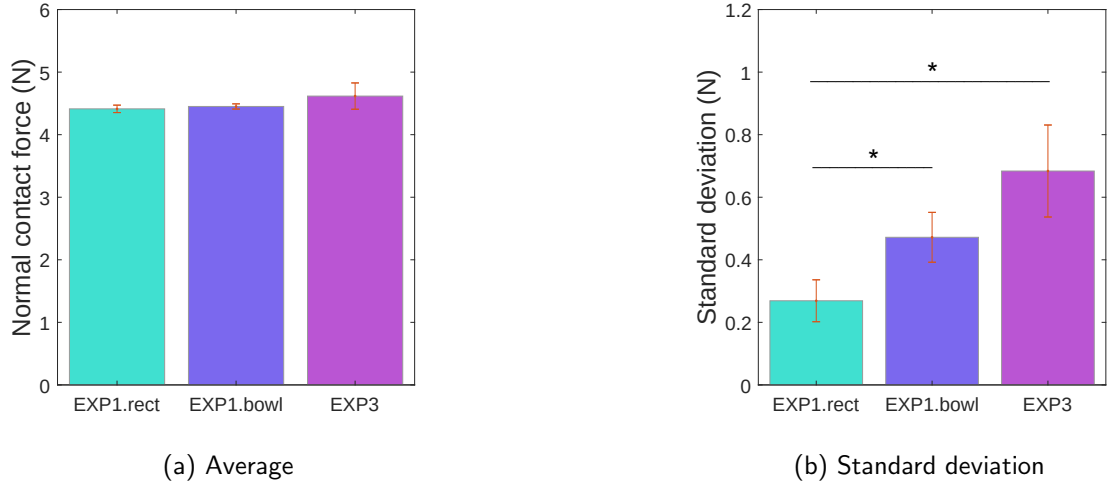


Figure A.7: Comparison between Experiment 3 and the two scenarios (rectangular and bowl-shaped) in Experiment 1. Note, EXP1.rect and EXP1.bowl mean rectangular scenario and bowl-shaped scenario in Experiment 1, respectively.

part for evaluating the effectiveness of the proposed system in future work. Also, user performance study on the proposed system needs to be systematically conducted and evaluated in the next step by medical experts like sonographers. The EMG acquisition device with wired connection is cumbersome to some extent for the operator in our current experiment. In future work, wireless communication will be employed for a more compact system.

A.4 Conclusion

Contact status transition between contact and non-contact is a main factor that may cause system instability. Compliant robot behavior can be expected from an impedance controller during physical human-robot interaction while accurate force regulation can be expected from a force controller. However, higher compliance may mean lower force regulation accuracy, and vice versa. In this work, a novel EMG-based hybrid impedance-force control system for human-robot collaborative Ultrasound (US) imaging task is developed and evaluated. The proposed control system incorporates the robot compliance and force regulation ability via a hybrid

impedance-force controller. EMG signal of the user is mapped with the hybrid controller as a modulator which allows the user to tune the trade-off between robot compliance and force regulation ability in an online manner. The effectiveness of the proposed control system is demonstrated by a proof-of-concept application study on US imaging.

The proposed control system is promising to be used in the US imaging task for monitoring the sonographer's fatigue, ensuring the patient's safety, and improving US imaging quality. This proposed system can be easily adapted to many other medical tasks that require strenuous physical human effort like procedures in orthopedic surgery.

Appendix B: Dual Mode *p*HRI-*tele*HRI Control System for Ultrasound Imaging¹

The COVID-19 pandemic has brought unprecedented extreme pressure on the medical system due to the physical distance policy, especially for procedures such as ultrasound (US) imaging, which are usually carried out in person. Tele-operation systems are a promising way to avoid physical human–robot interaction (*p*HRI). However, the system usually requires another robot on the remote doctor side to provide haptic feedback, which makes it expensive and complex. To reduce the cost and system complexity, in this work, we present a low-cost, easy-to-use, dual-mode *p*HRI-*tele*HRI control system with a custom-designed hybrid admittance-force controller for US imaging. The proposed system requires only a tracking camera rather than a sophisticated robot on the remote side. An audio feedback is designed for replacing haptic feedback on the remote side, and its sufficiency is experimentally verified. The experimental results indicate that the designed hybrid controller can significantly improve the task performance in both modes. Furthermore, the proposed system enables the user to conduct US imaging while complying with the physical distance policy, and allows them to seamlessly switch modes from one to another in an online manner. The novel system can be easily adapted to other medical applications beyond the pandemic, such as tele-healthcare, palpation, and auscultation.

¹A version of this chapter has been published as: Teng Li, Xiao Meng, and Mahdi Tavakoli, “Dual Mode *p*HRI-*tele*HRI Control System with a Hybrid Admittance-Force Controller for Ultrasound Imaging”, *Sensors*, Section: Sensors and Robotics, Special Issue: Sensors Technology for Medical Robotics, 2022, 22(11):4025. [\[MDPI\]](#)

B.1 Introduction

The COVID-19 pandemic has posed a huge challenge to medical systems due to the physical distance policy [46], especially for medical procedures that usually require physical contact, such as injections, palpation, and ultrasound (US) imaging. This has led to a growing interest in robotics in the fields of medicine and healthcare, such as robot-assisted systems, automated control systems, and tele-operation systems, which are regarded as promising substitute methods when physical human–robot interaction (p HRI) is limited or not available [46, 153, 9, 39, 148].

US imaging is widely used in the medical field, even including dentistry, due to its characteristics of being non-invasive, inexpensive, and radiation-free [123, 108]. Traditionally, US imaging relies on sonographers to physically move the US probe on the patient’s body. A robotized method is to attach the probe to a robot end-effector (EE) for assistance [15, 38]. For example, Carriere *et al.* [15] designed an admittance-controlled robot-assisted system for US scanning. Their system enabled the robot to automatically control the probe orientation and the probe–tissue contact force while the user controlled the lateral position of the US probe. A three-dimensional (3D) scanner was required to reconstruct the 3D surface of the soft tissue in real time in their system. Fang *et al.* [38] developed a cooperatively controlled robotic system to enable adaptive contact force assistance by involving a dual force sensor setup. Their system was demonstrated to have the ability to reduce user effort and improve image stability. Akbari *et al.* [5] developed an image quality online assessment algorithm for US scanning systems with which the system can automatically adjust the contact force. Soleymani *et al.* [132] designed a 3D-printed US scanning mechanism that enabled the operator to perform the US imaging task two meters away from the patient. A survey on robotic US systems in medicine can be found in [104].

In order to respect physical distancing, *i.e.*, to avoid physical interaction, tele-operation systems could be an alternative, promising solution for US imaging. There

is a long history in the use of tele-operation systems for US imaging even before the pandemic [104, 103, 114, 51, 91, 67, 33]. Two decades ago, Mitsuishi *et al.* [91] developed a remote US diagnostic system. Their tele-operation system consists of a 6-degree-of-freedom (DOF) leader manipulator attached with a three-axis force sensor on the doctor side, and a 7-DOF follower manipulator attached with another three-axis force sensor on the patient side. To ensure safety, their US probe can be retracted at any time to avoid injuring the patient. Their leader and follower manipulator has a distance as far as approximately 700 km.

Conti *et al.* [23] presented a tele-operation robotic system to assist sonographers in conducting US imaging in aiming to reduce physical fatigue. Their system utilized a 7-DOF Kuka LWR robot as the follower on the patient side and a 6-DOF haptic device as the leader on the doctor side. The system allows users to remotely operate the follower with force feedback, while the contact force remains at a pre-defined level.

Most recently, Duan *et al.* [33] developed a tele-operated robotic system for remote US diagnosis. Their system has a set of sophisticated control consoles on the doctor side, which can be used to remotely control the robot on the patient side to conduct US imaging.

The traditional tele-operation system is able to provide relatively accurate position/force control and realistic haptic feedback [91, 23]. However, as introduced above, it usually requires a second robot and/or a sophisticated control panel to be deployed on the remote side to establish a leader–follower system, which could make the system expensive and complicated to install and operate. This is also viewed as a major factor that hinders the popularity of tele-operation systems in the healthcare industry [23]. Moreover, many other challenges, such as control algorithm complexity and controller stability, must be dealt with during the development of such a leader–follower system.

In aiming for a low-cost, easy-to-use system, in this work, a novel dual-mode *pHRI-teleHRI* control system with a custom hybrid admittance-force controller is developed,

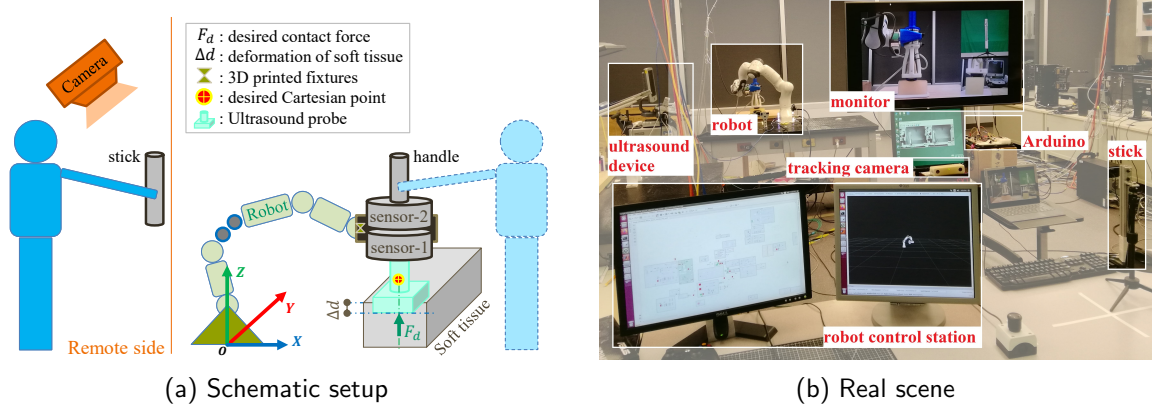


Figure B.1: Schematic setup and real scene for the proposed dual-mode *pHRI-teleHRI* control system. Note that sensor-1 and sensor-2 are external force/torque sensors of the same type, and they are stacked and installed by using a specially designed fixture to ensure that the two sensors work independently without affecting each other. In *pHRI* mode, the user can directly apply force on the handle attached at the robot EE in order to move the robot EE. In *teleHRI* mode, the user will hold and move the stick in order to move the robot EE while the stick pose is tracked by a tracking camera in real time.

as illustrated in Figure B.1. The proposed system needs only a tracking camera on the remote doctor side, which can largely reduce the cost and system complexity. During the development of such a system, a major challenge is the feedback design, since haptic feedback is not available on the remote doctor side. To address this issue, a concise audio feedback is designed to indicate the real-time contact force status, and its sufficiency for replacing haptic feedback is experimentally verified. Then, the effectiveness and robustness of the proposed dual-mode system are experimentally evaluated.

B.2 Methods

This paper describes an experimental study, where we first develop a dual-mode control system and then evaluate its effectiveness experimentally. In this section, we introduce the design methodology of the proposed dual-mode system in detail. First, the robot dynamics are described. Then, the admittance control used in *pHRI* mode is elaborated, followed by the mapping algorithm between the robot and the tracking camera, which will be implemented in *teleHRI* mode. The force controller

is then introduced, which will be employed in both modes. Lastly, we introduce the apparatus and audio feedback design that will be used for subsequent physical experiments.

B.2.1 Robot dynamics

The general dynamic model for an n -degree-of-freedom (DOF) rigid robot [41] can be given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J}^T \mathbf{F}_{\text{ext}} \quad (\text{B.1})$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ denotes the inertia matrix, $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes a matrix related to the Coriolis and centrifugal forces, $\mathbf{g} \in \mathbb{R}^n$ represents a gravity related vector, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the commanded joint torque vector, $\mathbf{F}_{\text{ext}} \in \mathbb{R}^6$ is the external force in Cartesian space (*e.g.*, robot–human interaction force, robot–environment contact force), and $\mathbf{J} \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix.

As shown in (B.1), the robot can directly receive and execute torque command $\boldsymbol{\tau}$ if a torque control interface is available. For some commercial robotic manipulators, however, interfaces for direct torque control may not be provided by the manufacturers due to safety issue considerations or other reasons. Instead, a velocity and/or position control interface is commonly provided. In this scenario, torque-related control methods such as impedance control [133] will not be usable. Alternatively, an admittance controller can be implemented on top of the velocity/position control interface in order to enable physical human–robot interaction.

B.2.2 Admittance controller for *p*HRI mode

The general mass–spring–damper model for admittance control in Cartesian space [97, 133] can be expressed as

$$\mathbf{F}_h = \mathbf{M}_1 \ddot{\mathbf{x}}_{\text{ad}} + \mathbf{D}_1 \dot{\mathbf{x}}_{\text{ad}} + \mathbf{K}_1 \mathbf{x}_{\text{ad}} \quad (\text{B.2})$$

where $\mathbf{F}_h \in \mathbb{R}^6$ is the external force applied on the robot EE (*e.g.*, by the human user), and $\mathbf{M}_1, \mathbf{D}_1, \mathbf{K}_1 \in \mathbb{R}^{6 \times 6}$ are the coefficient matrices of mass, damper, and

spring, respectively, and need to be designed. $\mathbf{x}_{ad}$, $\dot{\mathbf{x}}_{ad}$, $\ddot{\mathbf{x}}_{ad}$ are the relative position, velocity, and acceleration, respectively, in Cartesian space, which are caused by $\mathbf{F}_h$.

For admittance control, the input signal in the mass–spring–damper model (B.2) is the external force $\mathbf{F}_h$, while the output signal is the relative position displacement $\mathbf{x}_{ad}$ (or the relative velocity $\dot{\mathbf{x}}_{ad}$ when the spring term is removed). Taking the relative displacement $\mathbf{x}_{ad}$ as the output and rewriting (B.2), the transfer function from $\mathbf{F}_h$ to $\mathbf{x}_{ad}$ for an admittance controller can be expressed in the time domain as

$$\mathbf{x}_{ad} = \mathbf{K}_1^{-1}[\mathbf{F}_h - \mathbf{M}_1\ddot{\mathbf{x}}_{ad} - \mathbf{D}_1\dot{\mathbf{x}}_{ad}] \quad (\text{B.3})$$

It can be simplified as (B.4) when the spring term $\mathbf{K}_1\mathbf{x}_{ad}$ is disabled.

$$\dot{\mathbf{x}}_{ad} = \mathbf{D}_1^{-1}[\mathbf{F}_h - \mathbf{M}_1\ddot{\mathbf{x}}_{ad}] \quad (\text{B.4})$$

In this work, Eqn. (B.4) will be used as the transfer function of the admittance controller in the *pHRI* mode. Without involving the spring term in (B.4), the robot will not recover the initial position after the external human force is removed.

In summary, the procedures of admittance control used in this work are as follows. As illustrated in Figure B.2, first, an external human force $\mathbf{F}_h$ is applied on the robot EE and measured by a force/torque (F/T) sensor. Then, $\mathbf{F}_h$ is converted into a relative velocity $\dot{\mathbf{x}}_{ad}$ via the admittance controller (B.4). Then, the output velocity $\dot{\mathbf{x}}_{ad}$ from the admittance controller is converted into Cartesian displacement $\mathbf{x}_{ad}$ by an integrator. Finally, the displacement $\mathbf{x}_{ad}$ is added onto the initial desired robot EE pose $\mathbf{x}_d$ in a P-controller. Therefore, the *pHRI* mode is established based on the admittance controller (B.4).

B.2.3 Mapping algorithm for *teleHRI* mode

Besides the *pHRI* mode from the admittance controller, another *teleHRI* mode is designed for remote operation. As mentioned earlier, only a tracking camera and a stick (with a marker attached) are needed on the remote doctor side.

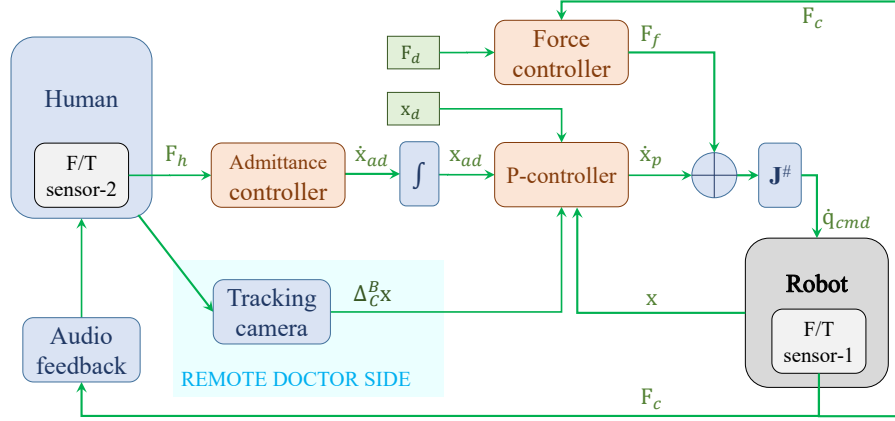


Figure B.2: Block diagram of the proposed dual-mode *pHRI-teleHRI* control system with a custom hybrid admittance-force controller. $\mathbf{F}_c$ represents the probe–tissue contact force measured by sensor-1, $\mathbf{F}_h$ represents the external human force measured by sensor-2, $\mathbf{F}_f$ represents the output of the force controller, $\dot{\mathbf{x}}_p$ represents the generated Cartesian velocity from the P-controller, $\mathbf{x}$ represents the actual pose of the robot EE in Cartesian space, $\mathbf{J}^\#$ represents the pseudoinverse of the Jacobian matrix, and $\dot{\mathbf{q}}_{cmd}$ represents the joint velocity command sent to the robot.

Assume that the tracking camera frame on the remote doctor side is denoted as $\{C\}$ and the robot base frame on the patient side is denoted as $\{B\}$. A direct frame mapping method between the two frames is established by

$$\begin{cases} \mathbf{B}\mathbf{x} = \mathbf{B}_C^B \mathbf{T} \mathbf{C}\mathbf{x} \\ \mathbf{B}_C^B \mathbf{T} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \end{cases} \quad (\text{B.5})$$

where $\mathbf{B}\mathbf{x}$ and $\mathbf{C}\mathbf{x}$ are the poses of the stick in the robot base frame $\{B\}$ and in the camera frame $\{C\}$, respectively. $\mathbf{B}_C^B \mathbf{T}$ is the direct transformation matrix from $\{C\}$ to $\{B\}$. Please note that in (B.5), $\mathbf{B}_C^B \mathbf{T}$ can be customized as necessary, and the main advantage of using the direct frame mapping method here is the robustness compared with using a 4-by-4 homogeneous transformation matrix, which requires accurate rotating angles and translations.

Based on the mapping method (B.5), a relative-displacement-based mapping algorithm between the pose of the stick (on the remote doctor side) and the pose of the robot EE (on the local patient side) is designed, and it can be expressed as

$$\begin{cases} {}^B\mathbf{x}_d = {}^B\mathbf{x}_d + \Delta_C^B\mathbf{x} \\ \Delta_C^B\mathbf{x} = {}^B_C\mathbf{T}\Delta^C\mathbf{x} \\ \Delta^C\mathbf{x} = {}^C\mathbf{x} - {}^C\mathbf{x}_0 \end{cases} \quad (\text{B.6})$$

where ${}^B\mathbf{x}_d \in \mathbb{R}^6$ is the desired pose of the robot EE (attached with US probe) in the robot base frame $\{B\}$, $\Delta_C^B\mathbf{x} \in \mathbb{R}^6$ is the relative displacement of the stick in tracking the camera frame $\{C\}$ as expressed in the robot base frame $\{B\}$, ${}^C\mathbf{x} \in \mathbb{R}^6$ and ${}^C\mathbf{x}_0 \in \mathbb{R}^6$ are the real-time pose and the initial pose of the stick in the camera frame $\{C\}$, respectively. Note that the first equation in (B.6) describes the updating rule for the desired robot EE pose (${}^B\mathbf{x}_d$) based on the real-time relative displacement of the stick ($\Delta_C^B\mathbf{x}$).

B.2.4 Force controller

The general form of the Cartesian space force tracking controller [116] can be given by

$$\mathbf{F}_f = \mathbf{K}_p(\mathbf{F}_c - \mathbf{F}_d) + \mathbf{K}_i \int_0^t (\mathbf{F}_c - \mathbf{F}_d)dt + \mathbf{K}_d(\dot{\mathbf{F}}_c - \dot{\mathbf{F}}_d) \quad (\text{B.7})$$

where $\mathbf{K}_p, \mathbf{K}_i, \mathbf{K}_d \in \mathbb{R}^{6 \times 6}$ are the designed proportional, integral, and derivative coefficient matrices, respectively, in Cartesian space, and are typically diagonal. $\mathbf{F}_d, \mathbf{F}_c \in \mathbb{R}^6$ are the desired and actual contact force between the robot EE and environment, respectively. For simplicity, in this work, a PI force controller is employed, as given by

$$\mathbf{F}_f = \mathbf{K}_p(\mathbf{F}_c - \mathbf{F}_d) + \mathbf{K}_i \int_0^t (\mathbf{F}_c - \mathbf{F}_d)dt \quad (\text{B.8})$$

where the actual contact force $\mathbf{F}_c$ is measured by an external F/T sensor.

Combining the admittance controller (B.4) for *pHRI* mode, the mapping algorithm (B.6) for *teleHRI* mode, and the force controller (B.8) together, a hybrid admittance-force controller for a dual-mode *pHRI-teleHRI* control system is constructed. The

block diagram for the proposed dual-mode control system is shown in Figure B.2, while the corresponding setup is illustrated in Figure B.1a.

B.2.5 Apparatus

A 7-DOF Franka Emika Panda robot (Franka Emika GmbH, Munich, Germany) is used for developing the proposed dual-mode *pHRI-teleHRI* control system, as shown in Figure B.1. The proposed control system is implemented on the Panda robot via a joint velocity control interface and MATLAB/Simulink (version R2019a, MathWorks Inc., Natick, MA, USA) code. The Simulink runs on a workstation computer of Intel(R) Core(TM) i5-8400 CPU @ 2.80 GHz $\times$ 6 with the Ubuntu 16.04 LTS (Xenial Xerus) 64-bit operating system. The control rate of the Panda robot is 1000 Hz.

A commercial MicronTracker with interface library MTC 3.8 (ClaroNav Inc., Toronto, ON, Canada) is used as the tracking camera on the remote doctor side to track the pose of the stick in real time with a frequency of 20 Hz. Please note that the commercial MicronTracker can be replaced by a commonly used regular camera for a lower cost.

In this work, the contact force between the US probe and tissue is measured by a 6-DOF F/T sensor (sensor-1 in Figure B.1, Axia80-M20-ZC22, ATI Industrial Automation, Inc., Apex, NC, USA). In the meantime, a second F/T sensor (sensor-2 in Figure B.1) of the same type is used to measure the external interaction force exerted on the robot EE handle by the human user, thus indicating user effort. The two sensors are stacked together as illustrated in Figure B.1a, and an exclusively designed fixture for mounting the two sensors on the robot EE is used to ensure that the two sensors work independently.

B.2.6 Audio feedback and haptic feedback

Audio, visual, and haptic feedback are the most commonly used feedback types in research. As a potential replacement for haptic feedback, audio feedback (AF) is

selected as a low-cost and simple alternative. Audio feedback is designed to indicate the real-time contact force status between the US probe and the soft tissue.

The normal contact force between the US probe and the tissue during scanning is one of the most important indicators since a stably controlled contact force can guarantee the US image quality [15, 23]. Different clinical examination types usually involve different desired ranges of contact force—for example, a general range of 5–20 N for cardiac, renal, and abdominal examinations [131, 49], and 6.4 N for carotid examinations [113]. Empirically, in this work, acceptable image quality can be obtained when the contact force is around 4.5 N. Therefore, an a priori decision is made to use a desired range of 4.5 ± 1 N for the normal contact force under the assumption that it can ensure high-quality scanning images. Note that the force controller in the proposed system is capable of setting other constant or varying desired forces.

Based on the desired force range, audio feedback is designed to indicate which range the current normal contact force is located in, *i.e.*, lower range $[-inf, 3.5]$ N, desired range $[3.5, 5.5]$ N, and upper range $[5.5, +inf]$ N. Audio feedback is provided in all experiments via an Arduino board and a buzzer. A continuous beep sound is used to indicate the desired range, while a discontinuous fast beep is used to indicate the upper range. Otherwise, no audio feedback is provided. In detail, the audio feedback signals are generated by supplying 5V DC signals from a classical Arduino MEGA2560 (R3) board to a passive buzzer (OSOYOO TMB12A05). The designed audio signals are given by

$$h(t) = \begin{cases} 0 & F_z \in (-inf, 3.5) \\ h_1 & F_z \in [3.5, 5.5] \\ h_1 \times a \times f(Freq, t) & F_z \in (5.5, +inf) \end{cases} \quad (B.9)$$

where $h(t)$ is the generated time-related audio signal, $h_1 = 20$ is the selected factor for pulse width modulation (PWM), F_z is the real-time normal contact force between the US probe and tissue, $a \times f(Freq, t)$ is a function of square wave form with respect to time t with setting $Freq = 8, a = 3$.

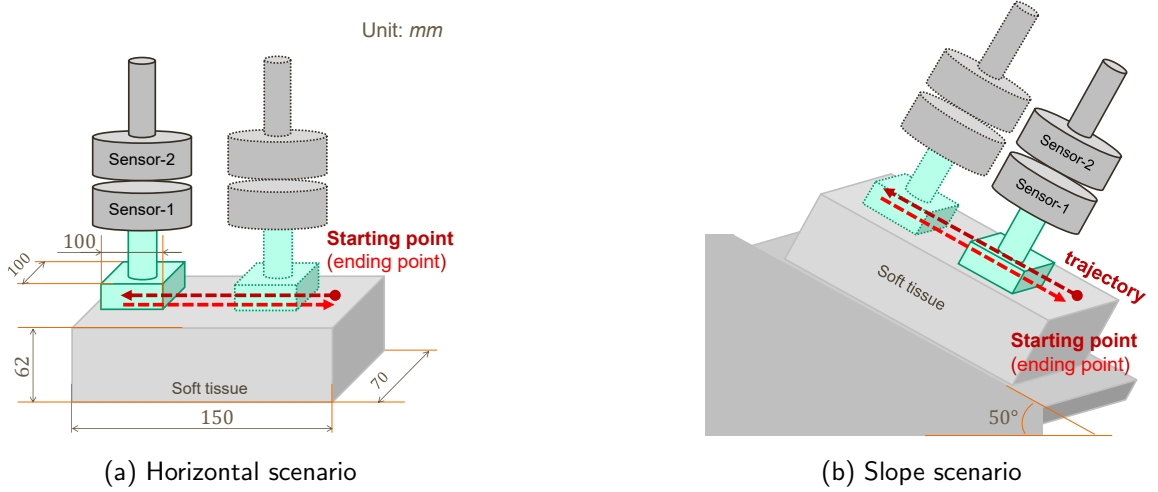


Figure B.3: Two scenarios of soft tissue surface, *i.e.*, horizontal and inclined slope.

On the other hand, haptic feedback (HF) is presented in *pHRI* mode. Please note that haptic feedback in this work refers to the natural haptic feedback when physical human–robot interaction occurs, rather than specially designed feedback.

B.3 Experiments and Results

B.3.1 Procedures and metrics

There are three experiments designed in this section for developing and evaluating the proposed dual-mode *pHRI-teleHRI* control system. In particular, Experiment 2 involves two tissue surface scenarios of horizontal and slope, as shown in Figure B.3. All experiments employ the same procedure for performing the ultrasound (US) imaging task, *i.e.*, 6 sessions are required in each mode (*pHRI* or *teleHRI*) while each session includes 3 trials. One trial is defined as the US probe moving over the soft tissue surface, starting at one end, moving to the other end, and then returning to the starting point. The *pHRI* mode represents the physical human-robot interaction method established by an admittance controller, while the *teleHRI* mode represents the tele-operation method established by a tracking camera system.

The performance metrics involved in the experiments are listed as follows:

- Normal contact force, mean and variance (squared standard deviation) between the US probe and the soft tissue. The former indicates task performance accuracy while the latter indicates task performance stability.
- User effort, in units of Newton. It is indicated by the force exerted on the robot EE by the human user in *p*HRI mode, and also serves as input signals for the admittance controller. It is measured by sensor-2, as shown in Figure B.1.
- Time percentage. A percentage for retaining the normal contact force within the desired range in one trial.

In the admittance controller (B.4), the coefficient matrices $\mathbf{M}_1$ and $\mathbf{D}_1$ are parameterized as $\mathbf{M}_{11} = 0.01\mathbf{I}_{3 \times 3}$ and $\mathbf{D}_{11} = 14\mathbf{I}_{3 \times 3}$ for the translational part and $\mathbf{M}_{22} = 0.0001\mathbf{I}_{3 \times 3}$ and $\mathbf{D}_{22} = 1.5\mathbf{I}_{3 \times 3}$ for the orientational part. For simplicity, the US probe is assumed to be exactly perpendicular to the tissue surface during the task (a more sophisticated 3D soft tissue reconstruction method may be required for cases beyond this assumption [15]); then, the normal contact force can be measured by the z -axis of the F/T sensor directly in the sensor frame. In the force controller (B.8), the desired force is set as $\mathbf{F}_d = [0, 0, 4.5, 0, 0, 0]$ in the F/T sensor frame and then transformed into the robot base frame. A t-test is employed for statistical analysis and a p -value of 0.05 is adopted as the significance level. A video demonstration for the experiments is available online¹.

B.3.2 Experiment 1: AF vs. (AF + HF)

In Experiment 1, audio feedback and haptic feedback (AF + HF) are presented in *p*HRI mode while audio-only feedback (AF) is presented in *tele*HRI mode during the US imaging task. In this experiment, we investigate how different feedback affects task performance. No force controller is implemented in this experiment, which means

¹online demo video link: <https://youtu.be/NkqlawDmJrM>

that both the lateral movement of the US probe and the normal contact force are controlled by the user.

This experiment requires the user to perform the US imaging task on a horizontal tissue surface in *pHRI* mode and *teleHRI* mode, respectively, with different feedback. As described earlier, a total of six sessions are required in each mode, while each session includes 3 trials. During the task, the user needs to manually control the lateral movement of the US probe and also needs to maintain the normal contact force between the the probe and the tissue in the desired range. As mentioned earlier, the task performance accuracy is indicated by the mean normal contact force throughout this paper, while the task performance stability is indicated by the corresponding variance.

Statistical analysis on the results (see Table B.1 for details) shows that there is no significant difference ($p = 0.5457$) in the mean normal contact force between the two modes, but there is a significant difference ($p = 0.0420$) between their variances, which means that the human user has significantly more stable task performance (*i.e.*, smaller variance) with AF in *teleHRI* mode than with AF + HF in *pHRI* mode. A sample of data is presented in Figure B.4. As can be seen in the figure, the normal contact force cannot stably remain in the desired range in either mode. This is also reflected by the time percentages for retaining the force in the desired range (see Table B.2 for details), which are lower than 75% in both modes (58.75% in *pHRI* mode and 74.37% in *teleHRI* mode).

The results from Experiment 1 show that the task performance accuracy in *teleHRI* mode with AF is comparable to that in *pHRI* mode with AF + HF in terms of averaged normal contact force. The task performance stability in *teleHRI* mode with AF is significantly better than that in *pHRI* mode with AF + HF in terms of their variances. These results indicate that the audio-only feedback (AF) is as good as audio-haptic feedback (AF + HF); thus, the audio feedback is able to serve as a replacement for the haptic feedback in our case.

Table B.1: Comparing task performance under different feedback in two modes when the proposed hybrid controller is not implemented in Experiment 1.

		mean		variance (<i>std.</i> ²)	
		AF+HF (<i>p</i> HRI)	AF (<i>tele</i> HRI)	AF+HF (<i>p</i> HRI)	AF (<i>tele</i> HRI)
F_z (N)	s1	4.45	4.59	2.1316	0.9409
	s2	4.42	4.46	2.4964	0.6561
	s3	4.55	4.31	0.9025	0.8100
	s4	4.29	4.35	1.5129	0.8464
	s5	4.76	4.69	1.1664	1.0404
	s6	4.03	4.45	1.3225	0.7744
		$p = 0.5457$		$p = 0.0420$ (*)	

Note: s1 ~ s6, represent session 1 ~ 6, respectively; AF means audio feedback; HF means haptic feedback; *std.* means standard deviation; F_z means normal contact force.

Table B.2: Time percentage for remaining force into desired range under different feedback in two modes when the proposed hybrid controller is not implemented in Experiment 1.

		trial 1	trial 2	trial 3	mean $\pm$ std.
<i>p</i> HRI mode (%)	s1	55.36	51.26	47.48	51.37 $\pm$ 3.94
	s2	63.52	64.64	35.83	54.66 $\pm$ 16.32
	s3	62.54	72.51	75.54	70.20 $\pm$ 6.80
	s4	62.87	63.05	46.68	57.53 $\pm$ 9.40
	s5	66.73	50.50	75.56	64.26 $\pm$ 12.71
	s6	53.78	59.49	50.15	54.47 $\pm$ 4.71
		58.75 $\pm$ 10.73 (mean $\pm$ std.)			
<i>tele</i> HRI mode (%)	s1	73.91	70.40	79.09	74.47 $\pm$ 4.37
	s2	75.46	70.79	91.51	79.25 $\pm$ 10.87
	s3	63.92	73.29	74.15	70.45 $\pm$ 5.67
	s4	64.49	72.29	80.27	72.35 $\pm$ 7.89
	s5	78.11	69.44	68.63	72.06 $\pm$ 5.25
	s6	76.31	75.85	80.72	77.63 $\pm$ 2.69
		74.37 $\pm$ 6.47 (mean $\pm$ std.)			

Note: s1 ~ s6, represent session 1 ~ session 6, respectively; *std.* means standard deviation.

B.3.3 Experiment 2: (AF + FC) vs. (AF + FC + HF)

In Experiment 2, the hybrid admittance-force controller is implemented. More specifically, an additional force controller (FC) is implemented into the control system in

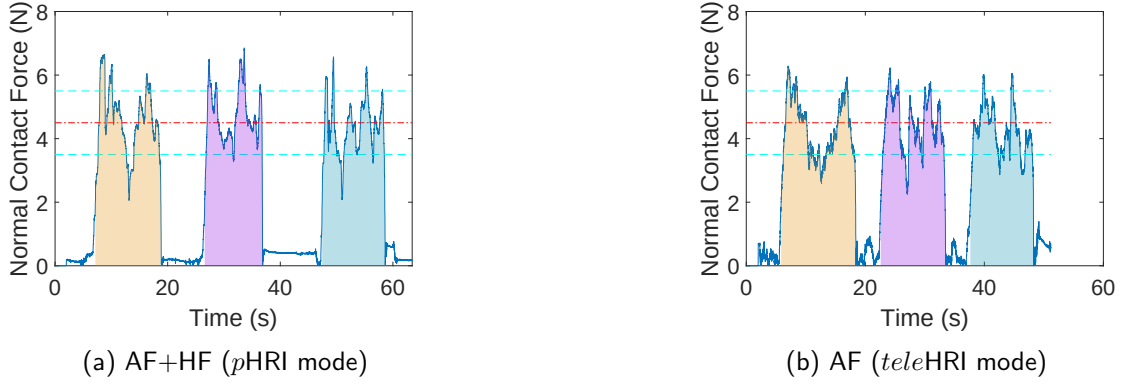


Figure B.4: Sample data for US scanning on horizontal tissue surface with different feedback are provided in two modes in Experiment 1. AF means audio feedback; HF means haptic feedback. Two horizontal dashed cyan lines define the tolerance area for the desired force.

both *pHRI* and *teleHRI* modes based on Experiment 1. This means that the normal contact force is regulated by the robot FC while the lateral movement is controlled by the human user in Experiment 2. As a further step based on Experiment 1, this experiment investigates how the different feedback will affect task performance when the proposed hybrid controller is implemented. The task procedures are the same as those described in Experiment 1. In particular, two tissue surface scenarios, namely the horizontal scenario and slope scenario (Figure B.3), are considered in Experiment 2 in order to test the flexibility of the proposed system.

(1) Experiment 2a: Horizontal scenario

In Experiment 2a, the US imaging task is conducted on a horizontal soft tissue surface. Statistical analysis (see Table B.3 for details) shows that the task performance accuracy in *teleHRI* mode with AF + FC is significantly better than in *pHRI* mode with AF + FC + HF ($p = 2.6999 \times 10^{-4}$) in terms of mean normal contact force. Despite this significance, it is worth noting that the max-min magnitude difference on the normal contact force across all sessions is only 0.17 N, which is close to the F/T sensor resolution 0.1 N. There is no significant difference between their variances in the two modes ($p = 0.1755$), which indicates that the task performance stability in the two modes is comparable.

Table B.3: Comparing the task performance on horizontal tissue surface under different feedback in two modes when the proposed hybrid controller is implemented in Experiment 2a.

		mean		variance ($std.^2$)	
		AF+FC+HF (<i>p</i> HRI)	AF+FC (<i>tele</i> HRI)	AF+FC+HF (<i>p</i> HRI)	AF+FC (<i>tele</i> HRI)
F_z (N)	s1	4.43	4.53	0.0064	0.0121
	s2	4.41	4.51	0.0036	0.0100
	s3	4.41	4.50	0.0144	0.0225
	s4	4.42	4.49	0.0100	0.0169
	s5	4.36	4.51	0.0100	0.0324
	s6	4.39	4.52	0.0324	0.0225
		$p = 2.6999 \times 10^{-4} (*)$		$p = 0.1755$	

Note: s1 ~ s6, represent session 1 ~ 6, respectively; AF means audio feedback; HF means haptic feedback; FC means force controller; std. means standard deviation; F_z means normal contact force.

A sample of data for Experiment 2a is shown in Figure B.5. As can be seen from the figure, user effort in *p*HRI mode is in the range of $[-5, 5]$ N, which indicates that the user can easily control the lateral movements of the US probe when an additional force controller is implemented.

(2) Experiment 2b: Slope scenario

In Experiment 2b, the US imaging task is conducted on an inclined slope soft tissue surface in *p*HRI mode and *tele*HRI mode separately. This slope tissue scenario could be further generalized to slopes with other angles of inclination or even an inverted tissue surface, which may be encountered in the clinical setting.

It is worth noting that in *tele*HRI mode, a regular camera on one side of the slope for side view is mounted with the same angle of inclination as the slope such that the inclined tissue surface in the camera view appears as a horizontal tissue surface. This setting is reasonable since the user is able to use any angle of view for a good viewpoint in *p*HRI mode. Moreover, since the pose mapping algorithm between the robot and the stick is based on relative displacements to their own initial poses, the motion of the stick on a horizontal surface can be automatically mapped to control

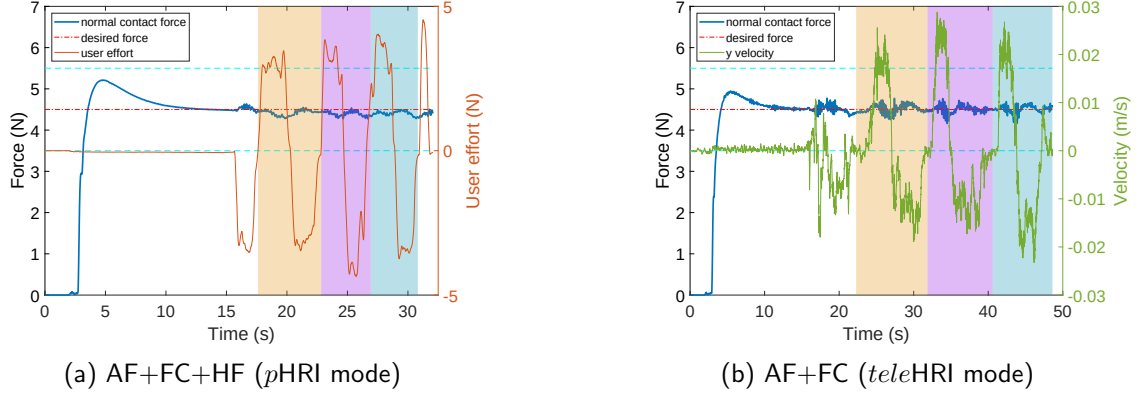


Figure B.5: Sample data for US scanning on horizontal tissue surface with different feedback are provided for two modes in Experiment 2a. AF means audio feedback; HF means haptic feedback; FC means force controller. Two horizontal dashed cyan lines define the tolerance area for the desired force.

the motion of the US probe on the inclined slope. This operational flexibility can help the user to obtain a better view and perform comfortable movements on the remote doctor side in *teleHRI* mode if needed.

A sample of data from Experiment 2b is shown in Figure B.6. In Figure B.6a, user effort is represented by the user-exerted force along the movement direction of the US probe (*i.e.*, along the slope in this experimental scenario). As can be seen from the figure, user effort in *pHRI* mode is in the range of $[-4, 4]$ N, which is relatively small. This means that the human user can easily control the lateral movements of the probe on the slope when an additional force controller is implemented.

Statistical analysis (see Table B.4 for details) shows that there is no significant difference ($p = 0.1412$) between the two modes in the mean normal contact force, and also no significant difference ($p = 0.1504$) in their variances.

The results in Experiments 2a and 2b show comparable task performance accuracy and task performance stability in *teleHRI* mode (with AF + FC) and in *pHRI* mode (with AF + FC + HF), which indicates the potential capability of *teleHRI* mode to be taken as an alternative for *pHRI* mode even without HF. In addition, compared to Experiment 1, task performance stability in Experiments 2a and 2b is significantly

Table B.4: Comparing the task performance on slope tissue surface under different feedback in two modes when the proposed hybrid controller is implemented in Experiment 2b.

		mean		variance (<i>std.</i> ²)	
		AF+FC+HF (<i>p</i> HRI)	AF+FC (<i>tele</i> HRI)	AF+FC+HF (<i>p</i> HRI)	AF+FC (<i>tele</i> HRI)
F_z (N)	s1	4.49	4.51	0.0289	0.0036
	s2	4.43	4.49	0.0144	0.0025
	s3	4.49	4.50	0.0144	0.0121
	s4	4.49	4.50	0.0121	0.0064
	s5	4.49	4.50	0.0361	0.0081
	s6	4.50	4.49	0.0081	0.0196
		$p = 0.1412$		$p = 0.1504$	

Note: s1 ~ s6, represent session 1 ~ 6, respectively; AF means audio feedback; HF means haptic feedback; FC means force controller; *std.* means standard deviation; F_z means normal contact force.

improved (all $ps < 0.002$).

The results in Experiment 2 indicate the same conclusion as that obtained in Experiment 1, *i.e.*, audio feedback can be a good replacement for haptic feedback. More importantly, the hybrid admittance-force controller implemented in Experiment 2 further relieves the need for haptic feedback in *tele*HRI mode.

B.3.4 Experiment 3: Dual-mode switching

Experiment 3 is designed to evaluate the overall performance of the proposed dual-mode *p*HRI-*tele*HRI control system when mode switching is involved. This experiment requires the human user to perform the task using a “1-2-1-2” sequence, *i.e.*, first to perform the task using the stick (in *tele*HRI mode) for one session, then perform the task using the robot EE handle (in *p*HRI mode) for another session, then perform the task in *tele*HRI mode again for one session, then perform the task in *p*HRI mode again. This procedure is repeated another two times in order to generate six sessions for each mode.

The task procedure in this experiment can be better understood via the sample data shown in Figure B.7. In the figure, two short bar areas represent the *tele*HRI

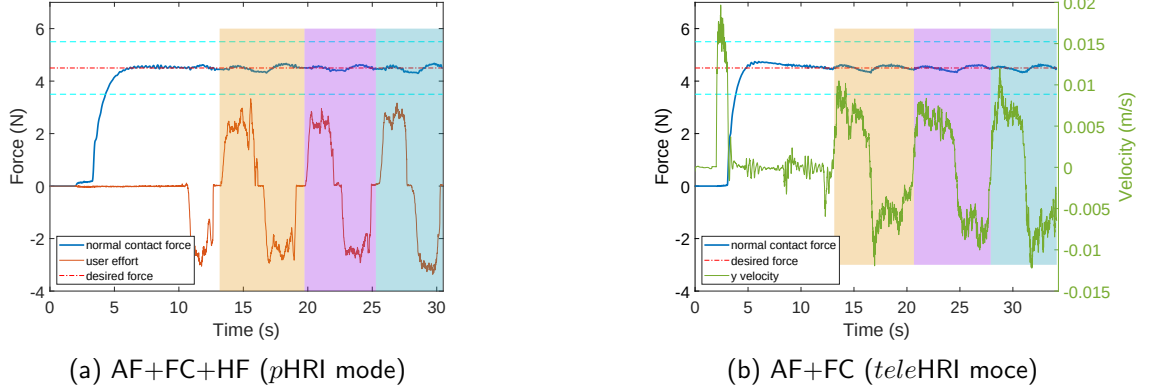


Figure B.6: Sample data for US scanning on inclined slope tissue surface with different feedback are provided for two modes in Experiment 2b. AF means audio feedback; HF means haptic feedback; FC means force controller. Two horizontal dashed cyan lines define the tolerance area for the desired force.

mode, while two long bar areas represent the *pHRI* mode. User effort indicates the user-exerted force on the robot EE handle (in *pHRI* mode) along the lateral movement direction of the US probe. As can be seen in the figure, the switching between the *pHRI* and *teleHRI* mode is seamless, smooth, and stable, and it can be performed whenever necessary, without involving stability issues. This is reasonable and expected due to the relative-displacement-based mapping method, which will be discussed in more detail in the next section.

Statistical analysis (see Table B.5 for details) shows that there is no significant difference in the task performance accuracy between the two modes ($p = 0.1747$) in terms of the normal contact force. Although there is a significant difference statistically in their variances ($p = 0.0033$), it is noticed that all the standard deviation values are less than 0.1 N (*i.e.*, less than the F/T sensor resolution). Considering this, it can be safely concluded here that there is no significant difference found between the two modes in terms of either normal contact force or their variances when switching is involved, which indicates the robustness of the proposed dual-mode system during mode switching.

Table B.5: Comparing the general performance of the proposed dual mode *pHRI-teleHRI* system when mode switching is involved in Experiment 3.

		mean		variance (<i>std.</i> ²)	
		<i>pHRI</i>	<i>teleHRI</i>	<i>pHRI</i>	<i>teleHRI</i>
F_z (N)	s1	4.50	4.50	0.0025	0.0016
	s2	4.49	4.50	0.0049	0.0016
	s3	4.50	4.50	0.0036	0.0025
	s4	4.49	4.50	0.0064	0.0036
	s5	4.50	4.50	0.0049	0.0016
	s6	4.50	4.50	0.0049	0.0016
		$p = 0.1747$		$p = 0.0033(*)$	

Note: s1 ~ s6, represent session 1 ~ 6, respectively; *std.* means standard deviation; F_z means normal contact force.

B.3.5 Statistical comparison across experiments

The longitudinal comparison of the task performance accuracy and the task performance stability across Experiments 1, 2a, 2b, and 3 is conducted in *pHRI* mode and in *teleHRI* mode separately by using a t-test. Hereafter, for compactness, EX.1, 2a, 2b, and 3 will be used to represent Experiments 1, 2a, 2b, and 3, respectively.

In *pHRI* mode, there is no significant difference in the normal contact force between EX.1 and either of the other experiments (see Figure B.8a). However, it should be noted that the mean values cannot truly reflect the task performance stability, which mainly depends on their variances. For their variances, there is a significant difference (all $ps < 0.002$) between EX.1 and either of the other experiments in *pHRI* mode (see Figure B.8b). Similar statistical results are obtained for *teleHRI* mode (see Figure B.8c,d). These results indicate that both in *pHRI* and *teleHRI* modes, the task performance is significantly improved in terms of task performance stability and reliability by implementing the designed hybrid admittance-force controller (EX.2a, 2b, 3). The statistical analysis results for the longitudinal comparison across experiments are summarized in Table B.6.

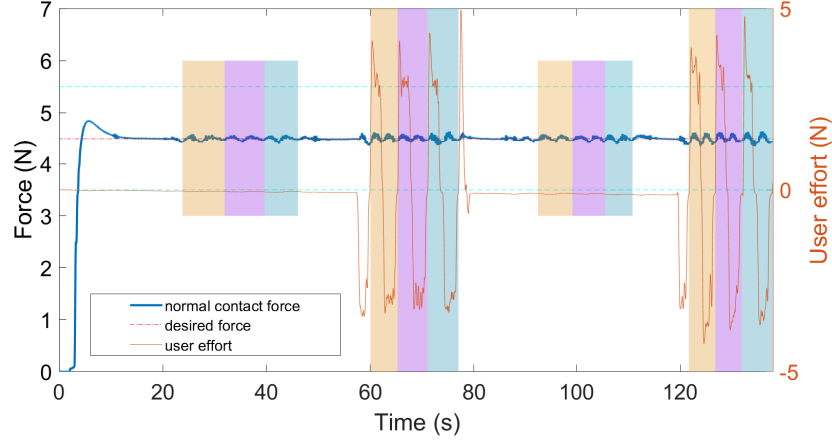


Figure B.7: Sample data for dual-mode switching test using a “1-2-1-2” switching sequence in Experiment 3. Each color bar period represents a trial. Two short color bar areas represent the *teleHRI* mode, while two long color bar areas represent the *pHRI* mode. Two horizontal cyan lines define the tolerance area for the desired force.

Table B.6: Summary of p -values of t-test results.

		EX.2a	EX.2b	EX.3	EX.2a	EX.2b	EX.3
		mean			variance ($std.^2$)		
EX.1	F_z (<i>pHRI</i>)	0.9031	0.5530	0.4600	0.0015*	0.0014*	0.0014*
EX.2a		-	0.0042*	0.0004*	-	0.4436	0.1035
EX.2b		-	-	0.1647	-	-	0.0276*
EX.1	F_z (<i>teleHRI</i>)	0.5546	0.7033	0.6876	0.00002*	0.00002*	0.00002*
EX.2a		-	0.1099	0.1438	-	0.0152*	0.0037*
EX.2b		-	-	0.6109	-	-	0.0529

Note: EX.1,2a,2b,3 mean Experiment 1,2a,2b,3, respectively; F_z means normal contact force; * for significance level under 5%.

B.4 Discussions

In this work, we propose a dual-mode *pHRI-teleHRI* control system with a custom-designed hybrid admittance-force controller for US imaging. The effectiveness of the proposed system is experimentally evaluated. Experiment 1 is conducted to investigate the possible effects of different feedback types on task performance in two modes, *i.e.*, audio and haptic feedback (AF + HF) in *pHRI* mode and audio-only feedback (AF) in *teleHRI* mode. Despite the absence of haptic feedback (HF) in *teleHRI* mode,

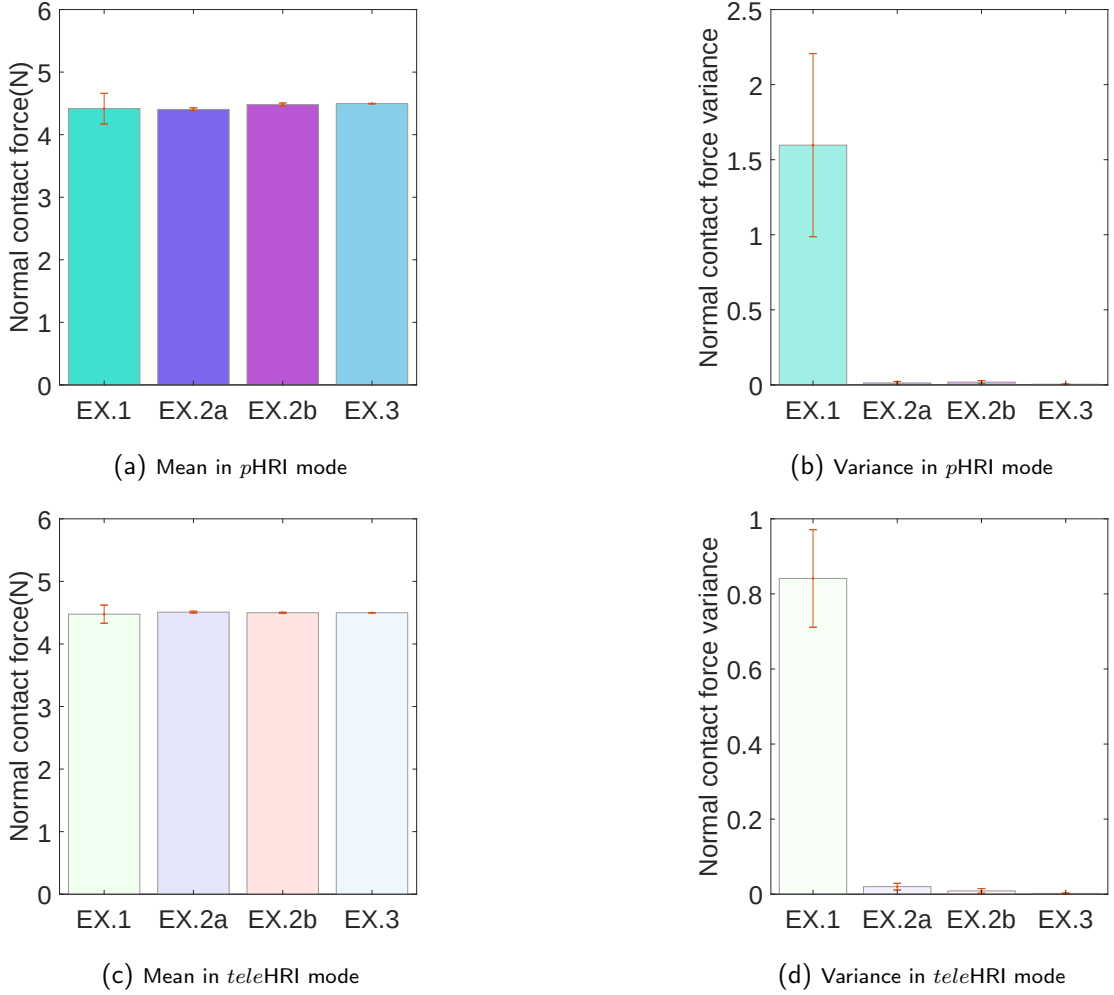


Figure B.8: Bar chart for normal contact force and variance. ‘EX.1’, ‘EX.2a’, ‘EX.2b’, ‘EX.3’ represent Experiments 1, 2a, 2b, and 3, respectively.

the task performance is comparable to that in *pHRI* mode. In other words, the audio feedback is capable of being a sufficient replacement for the haptic feedback in our case.

Experiment 2 is an improved version of Experiment 1 while implementing the hybrid admittance-force controller. The results show that, again, the task performance in the mode without haptic feedback (*teleHRI*) is as good as that in the mode with haptic feedback (*pHRI*). In other words, the audio feedback can be a sufficient alternative to the haptic feedback in this work. Additionally, the implementation of the hybrid admittance-force controller in Experiment 2 significantly improves the task

performance when compared with Experiment 1.

In Experiment 2b with the *teleHRI* mode, benefiting from the relative-displacement-based mapping method, the movement of the stick on a horizontal surface could be mapped to control the movement of the US probe on an inclined slope surface. In addition, the rotation angle of the regular camera for side view can be tuned as necessary such that a slope surface in the physical world can be shown as a horizontal surface in the camera view. This operational flexibility allows the user to perform comfortable movements and obtain a good view in *teleHRI* mode if needed. The optional setting for aligning the camera with the slope makes the *teleHRI* mode comparable to the *pHRI* mode considering that the human user is able to adjust their view point for a better perspective in either mode.

Experiment 3 assessed the overall performance of the proposed dual-mode system when switching mode (*i.e.*, switch *pHRI* mode to *teleHRI* mode, or vice versa) is involved. The proposed system does not require an actual switch “button” since the two modes are co-existent and coupled via a summation operator to the desired Cartesian pose (in the P-controller in Figure B.2). Therefore, in order to perform a switch between the two modes, the user only needs to switch the tool used for the operation, *i.e.*, the robot EE handle for *pHRI* mode or the stick for *teleHRI* mode. Then, the system will automatically run into the corresponding mode according to the tool used by the user. Due to the relative-displacement-based mapping method, the dual-mode switching is seamless and smooth, and does not involve stability issues. The results also indicated that the dual modes can be switched from one to another at any time point, which can ensure the robustness and safety of the proposed dual-mode system in case of emergency cases. A potential advantage of this dual-mode design is that even during the tele-operation in *teleHRI* mode, other users (*e.g.*, an assistant) on the patient side can interfere in the ongoing tele-operation whenever necessary and manually move the US probe away from the tissue/patient by using the robot EE handle.

The overall experimental results indicated that the newly designed tele-operation method (*teleHRI* mode) is capable of being used as an alternative to the *pHRI* method for US scanning when physical distancing is required or when *pHRI* is not available. The capability of allowing seamless switching between the dual modes at any time enables the robustness of the proposed system.

One potential benefit from the proposed system is the low-cost, easy-to-deploy device on the remote doctor side compared with the traditional leader–follower tele-operation systems. High-cost devices has been taken as one main factor inhibiting the implementation of the traditional tele-operation system in the healthcare field [23]. The remote operation method proposed in this work (*teleHRI* mode) only needs a tracking camera rather than a sophisticated, expensive, multi-DOF haptic device on the doctor side. Additionally, the cost can be further lowered by choosing a cheaper, regular camera as the tracking device.

Another potential benefit is that it can potentially relieve the strenuous physical efforts and constraints experienced during physical interaction [23] since the stick used on the remote side could be made as light as possible. Moreover, it allows the user to use any available support or any comfortable body posture, thus reducing fatigue. Especially for US scanning tasks that require a long time to complete, this flexibility could be beneficial to sonographers.

One limitation of the proposed system is the low accuracy of the registration between the remote tracking camera frame and the local robot frame. Therefore, the proposed system is not suitable for high-accuracy-demanding tele-operation scenarios.

B.5 Conclusion

In this work, a dual-mode *pHRI-teleHRI* control system with a hybrid admittance-force controller is developed for US imaging. Instead of employing an expensive and sophisticated robot as a leader on the remote doctor side, a low-cost tracking camera and a stick attached with a tracking marker are utilized to remotely control the robot

that is on the patient side. The tele-operation method with only audio feedback (*i.e.*, in *teleHRI* mode) showed comparable task performance to the physical interaction method with audio and haptic feedback (*i.e.*, in *pHRI* mode). This verified that the designed audio feedback can be a sufficient replacement for haptic feedback in our case, and the *teleHRI* mode is capable of being used as an alternative method when physical distancing needs to be respected. Furthermore, experimental results showed that the *pHRI* and the *teleHRI* modes can be switched from one to another seamlessly at any time point without affecting system stability, which demonstrates the robustness and stability of the proposed system.

The dual-mode control system and hybrid admittance-force controller can be easily adapted to other applications where tele-operation is needed beyond the pandemic, such as needle insertion, auscultation, and palpation. In future work, automatic path planning and trajectory tracking, as well as virtual fixture guiding, will be introduced into the system for better task repeatability, which can result in a more intelligent and autonomous system.

Appendix C: Kinematics and Dynamics of the 3DOF PHANToM Premium 1.5A Robot

C.1 Robotic System

In the thesis, a 3DOF PHANToM Premium 1.5A robot (3D Systems, Inc., Cary, NC, USA) is frequently used for simulations and experiments. Therefore, its kinematics, dynamic model, and linearized dynamic model are summarized in this appendix.

C.2 Robot Kinematics¹

The Denavit–Hartenberg (DH) table for the 3-DOF PHANToM Premium 1.5A robot is provided in Table C.1. The corresponding frames’ definitions are illustrated in Figure C.1. Please note that the base frame of this robot is defined to be coincident with the initial robot end-effector (EE) position (see the gray pose in Figure C.1).

According to the DH parameters in Table C.1 and the frames determined in Figure C.1, the homogeneous transformation matrix T from EE frame $\{5\}$ to base frame $\{0\}$ can be obtained as

$$T = \begin{bmatrix} c_1, & -s_1s_3, & c_3s_1, & s_1(L_1c_2 + L_2s_3) \\ 0, & c_3, & s_3, & L_2 - L_2c_3 + L_1s_2 \\ -s_1, & -c_1s_3, & c_1c_3, & L_1c_1c_2 - L_1 + L_2c_1s_3 \\ 0, & 0, & 0, & 1 \end{bmatrix} \quad (\text{C.1})$$

¹This part of the content has been published in: Teng Li, Hongjun Xing, Ehsan Hashemi, Hamid D. Taghirad, Mahdi Tavakoli, “A Brief Survey of Observers for Disturbance Estimation and Compensation”, *Robotica*, 2023, 41(12), 3818–3845. [\[Robotica\]](#)

Table C.1: Denavit–Hartenberg (DH) parameters for the 3DOF Phantom Premium 1.5A robot's kinematic chain (for the homogeneous transform in the modified convention).

	no.	Joint	$a(m)$	$\alpha(rad)$	$d(m)$	$\theta(rad)$
	1	Joint 01	0	$-\frac{\pi}{2}$	L_2	$-\frac{\pi}{2}$
$\oplus$	2	Joint 02	$-L_1$	0	0	0
$\oplus$	3	Joint 1	0	0	0	q_1
	4	Joint 2	0	$\frac{\pi}{2}$	0	q_2
	5	Joint 3	L_1	0	0	$q_3 - q_2 - \frac{\pi}{2}$
$\oplus$	6	Joint 4 (EE)	L_2	0	0	0
$\oplus$	7	Joint 5 (EE)	0	$-\frac{\pi}{2}$	0	$\frac{\pi}{2}$

Note: EE, end-effector. L_1 and L_2 are link length. Symbol $\oplus$ means that the DH parameters of these two adjacent joints can be directly summed together, respectively, to be as one joint. Joint 01, 02, and 5 are virtual joints that are only used for transforming one frame to another desired one via translation and/or rotation.

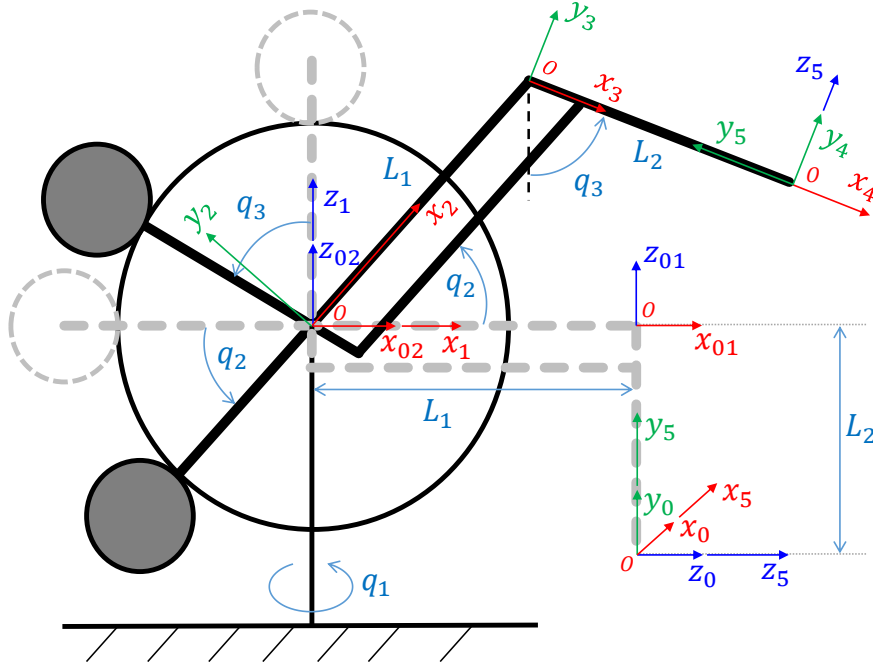


Figure C.1: Schematic of the 3DOF Phantom Premium 1.5A robot and frame attachment to each joint. Frame $\{0\}$ is the base frame while frame $\{5\}$ is the end-effector (EE) frame. L_1, L_2 are link lengths. q_1, q_2, q_3 are joint angle variables.

where s_i, c_i represent $\sin(q_i), \cos(q_i), i = 1, 2, 3$, respectively. L_1, L_2 are link lengths.

The Jacobian matrix J can be expressed by

$$J = \begin{bmatrix} c_1(L_1c_2 + L_2s_3), & -L_1s_1s_2, & L_2c_3s_1 \\ 0, & L_1c_2, & L_2s_3 \\ -s_1(L_1c_2 + L_2s_3), & -L_1c_1s_2, & L_2c_1c_3 \\ 0, & 0, & -c_1 \\ 1, & 0, & 0 \\ 0, & 0, & s_1 \end{bmatrix} \quad (\text{C.2})$$

Note that the Jacobian matrix here is the space Jacobian which can be calculated from the body Jacobian used in [16]. In this thesis, we only use the upper 3-by-3 linear part of the Jacobian in Eqn. (C.2), which means that the rotational angles in Cartesian space are ignored.

C.3 Robot Dynamics

The dynamic model of the 3DOF Phantom Premium 1.5A robot can be expressed in the form of

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_{\text{fric}}(\dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (\text{C.3})$$

where $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ is the inertia matrix, $\mathbf{S} \in \mathbb{R}^{3 \times 3}$ is a matrix related to Coriolis and Centrifugal forces, $\mathbf{g} \in \mathbb{R}^3$ is a vector related to gravity, $\boldsymbol{\tau}_{\text{fric}} \in \mathbb{R}^3$ is a vector related to joint friction, $\boldsymbol{\tau} \in \mathbb{R}^3$ is a vector related to the torques in the joints. Note that $\mathbf{M}$ is a symmetric, positive definite matrix, and always invertible for any joint configuration $\mathbf{q}$; $\mathbf{S}$ is not a symmetric matrix in general; $\dot{\mathbf{M}} - 2\mathbf{S}$ is skew-symmetric when using Christoffel symbols to define matrix $\mathbf{S}$.

According to [16], for the inertia matrix $\mathbf{M}$ of the 3DOF Phantom Premium 1.5A robot, it can be expressed as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q}) \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & M_{23} \\ 0 & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} \quad (\text{C.4})$$

For the $\mathbf{S}$ matrix related to Coriolis and Centrifugal forces, it can be expressed as

$$\mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}}) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & 0 & S_{23} \\ S_{31} & S_{32} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \quad (\text{C.5})$$

Note that the Coriolis and Centrifugal related vector is also often represented by the letter $\mathbf{c}$. It is worth noting that the expression of the $\mathbf{S}$ matrix may not be unique since the term related to Coriolis and Centrifugal forces is in quadratic form with respect to $\dot{\mathbf{q}}$. In other words, the quadratic term $[c_1 \ c_2 \ c_3]^T$ is fixed for a specific robot but the $\mathbf{S}$ matrix can be different, depending on the factorization method.

For the gravity vector $\mathbf{g}$, it can be expressed as

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \quad (\text{C.6})$$

For the joint friction vector $\boldsymbol{\tau}_{\text{fric}}$, it can be expressed as

$$\boldsymbol{\tau}_{\text{fric}}(\dot{\mathbf{q}}) = \begin{bmatrix} \tau_{f1} \\ \tau_{f2} \\ \tau_{f3} \end{bmatrix} \quad (\text{C.7})$$

There are two commonly used methods for building the dynamic model and calculating each of the components, *i.e.*, Newton-Euler (NE) method, and Euler-Lagrange (EL) method. The NE method is to build the dynamic model by balancing the forces/-torques, which is an iterative algorithm in numeric/recursive form. This method is best for implementing control schemes and can be used in real-time. It can also be used to get a symbolic/closed form of a robot dynamic model theoretically. The EL method is an energy-based approach, and usually expresses the dynamic equations in symbolic/closed form. This method is best for the study of dynamic properties and analysis of control schemes.

Using the EL method, the analytic expressions of elements in $\mathbf{M}$, $\mathbf{S}$, $\mathbf{g}$, and $\boldsymbol{\tau}_{\text{fric}}$ for the 3DOF Phantom Premium 1.5A robot can be found as [16]

$$\left\{ \begin{array}{l} M_{11} = \frac{1}{8}(4I_{ayy} + 4I_{azz} + 8I_{baseyy} + 4I_{beyy} + 4I_{bezz} + 4I_{cyy} + 4I_{czz} + 4I_{dfyy} \\ \quad + 4I_{dfzz} + 4L_1^2m_a + L_2^2m_a + L_1^2m_c + 4L_3^2m_c) \\ \quad + \frac{1}{8}(4I_{beyy} - 4I_{bezz} + 4I_{cyy} - 4I_{czz} + L_1^2(4m_a + m_c)) \cos(2q_2) \\ \quad + \frac{1}{8}(4I_{ayy} - 4I_{azz} + 4I_{dfyy} - 4I_{dfzz} - L_2^2m_a - 4L_3^2m_c) \cos(2q_3) \\ \quad + L_1(L_2m_a + L_3m_c) \cos(q_2) \sin(q_3) \\ M_{22} = \frac{1}{4}(4(I_{bexx} + I_{cxx} + L_1^2m_a) + L_1^2m_c) \\ M_{23} = -\frac{1}{2}L_1(L_2m_a + L_3m_c) \sin(q_2 - q_3) \\ M_{32} = M_{23} \\ M_{33} = \frac{1}{4}(4I_{axx} + 4I_{dfxx} + L_2^2m_a + 4L_3^2m_c) \end{array} \right. \quad (\text{C.8})$$

$$\left\{ \begin{array}{l} S_{11} = \frac{1}{8}(-2 \sin(q_2)((4I_{beyy} - 4I_{bezz} + 4I_{cyy} - 4I_{czz} \\ \quad + 4L_1^2m_a + L_1^2m_c) \cos(q_2) + 2L_1(L_2m_a + L_3m_c) \sin(q_3))\dot{q}_2 \\ \quad + 2 \cos(q_3)(2L_1(L_2m_a + L_3m_c) \cos(q_2) \\ \quad + (-4I_{ayy} + 4I_{azz} - 4I_{dfyy} + 4I_{dfzz} + L_2^2m_a + 4L_3^2m_c) \sin(q_3))\dot{q}_3) \\ S_{12} = -\frac{1}{8}((4I_{beyy} - 4I_{bezz} + 4I_{cyy} - 4I_{czz} + L_1^2(4m_a + m_c)) \sin(2q_2) \\ \quad + 4L_1(L_2m_a + L_3m_c) \sin(q_2) \sin(q_3))\dot{q}_1 \\ S_{13} = -\frac{1}{8}(-4L_1(L_2m_a + L_3m_c) \cos(q_2) \cos(q_3) \\ \quad - (-4I_{ayy} + 4I_{azz} - 4I_{dfyy} + 4I_{dfzz} + L_2^2m_a + 4L_3^2m_c) \sin(2q_3))\dot{q}_1 \\ S_{21} = -S_{12} \\ S_{23} = \frac{1}{2}L_1(L_2m_a + L_3m_c) \cos(q_2 - q_3)\dot{q}_3 \\ S_{31} = -S_{13} \\ S_{32} = \frac{1}{2}L_1(L_2m_a + L_3m_c) \cos(q_2 - q_3)\dot{q}_2 \end{array} \right. \quad (\text{C.9})$$

$$\left\{ \begin{array}{l} g_1 = 0 \\ g_2 = \frac{1}{2}g(2L_1m_a + 2L_5m_{bc} + L_1m_c) \cos(q_2) \\ g_3 = \frac{1}{2}g(L_2m_a + 2L_3m_c - 2L_6m_{df}) \sin(q_3) \end{array} \right. \quad (\text{C.10})$$

The values of the dynamic parameters from [16] are summarized in Table C.2.

C.4 Dynamic Model Linearization

The joint friction can be modeled by the Stribeck model [94, 80] as given by Eqn. (C.11), which is viewed as the most classical nonlinear expression that can include many characteristics of friction.

$$\boldsymbol{\tau}_{\text{fric}} = \mathbf{F}_c \text{sgn}(\dot{\mathbf{q}}) + (\mathbf{F}_s - \mathbf{F}_c) \text{sgn}(\dot{\mathbf{q}}) e^{-|\dot{\mathbf{q}}/\nu_s|^\sigma} + \mathbf{F}_v \dot{\mathbf{q}} \quad (\text{C.11})$$

Table C.2: The values of the dynamic parameters for the 3DOF Phantom Premium 1.5A robot.

Parameters	Values
Link-1 length	$L_1 = 0.215$
Link-2 length	$L_2 = 0.170$
Distance between parallel links	$L_3 = 0.0325$
Mass of segment A	$m_a = 0.0202$
Inertia of segment A	$I_a = \text{diag}\{I_{axx}, I_{ayy}, I_{azz}\}$
Inertia of segment A	$I_{axx} = 0.4864 \times 10^{-4}$
Inertia of segment A	$I_{ayy} = 0.001843 \times 10^{-4}$
Inertia of segment A	$I_{azz} = 0.4864 \times 10^{-4}$
Mass of segment C	$m_c = 0.0249$
Inertia of segment C	$I_c = \text{diag}\{I_{cxx}, I_{cyy}, I_{czz}\}$
Inertia of segment C	$I_{cxx} = 0.959 \times 10^{-4}$
Inertia of segment C	$I_{cyy} = 0.959 \times 10^{-4}$
Inertia of segment C	$I_{czz} = 0.0051 \times 10^{-4}$
Mass of segment BE	$m_{be} = 0.2359$
Inertia of segment BE	$I_{be} = \text{diag}\{I_{bexx}, I_{beyy}, I_{bez z}\}$
Inertia of segment BE	$I_{bexx} = 11.09 \times 10^{-4}$
Inertia of segment BE	$I_{beyy} = 10.06 \times 10^{-4}$
Inertia of segment BE	$I_{bez z} = 0.591 \times 10^{-4}$
COM location of BE	$L_5 = -0.0368$
Mass of segment DF	$m_{df} = 0.1906$
Inertia of segment DF	$I_{df} = \text{diag}\{I_{dfxx}, I_{dfyy}, I_{dfzz}\}$
Inertia of segment DF	$I_{dfxx} = 7.11 \times 10^{-4}$
Inertia of segment DF	$I_{dfyy} = 0.629 \times 10^{-4}$
Inertia of segment DF	$I_{dfzz} = 6.246 \times 10^{-4}$
COM location of DF	$L_6 = 0.0527$
Inertia of the base	$I_{baseyy} = 11.87 \times 10^{-4}$

Note: COM, center of mass.

where $\mathbf{F}_c \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix for the Coulomb friction coefficient, $\mathbf{F}_s \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix for the static friction coefficient, $\mathbf{F}_v \in \mathbb{R}^{3 \times 3}$ is a diagonal matrix for the viscous friction coefficient, $\dot{\mathbf{q}}$ is the joint velocity, ν_s is the Stribeck parameter, σ is the exponent of the Stribeck nonlinearity and $\sigma = 2$ is usually employed.

Instead of the sophisticated Stribeck model Eqn. (C.11), a simple friction model only involving viscous friction and Coulomb friction is employed for the 3DOF Phan-

tom Premium 1.5A robot in [137] which is given by

$$\boldsymbol{\tau}_{\text{fric}} = \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{F}_c \text{sgn}(\dot{\mathbf{q}}) \quad (\text{C.12})$$

The dynamic model Eqn. (C.3) can be linearized by extracting a minimum set of dynamic coefficients $\boldsymbol{\pi}$ [137], and the linearized dynamic model can be given by

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \boldsymbol{\pi} = \boldsymbol{\tau} = \mathbf{Y}_{[3 \times 14]} \boldsymbol{\pi}_{[14 \times 1]} = \begin{bmatrix} \mathbf{Y}_{d[3 \times 6]} & \mathbf{Y}_{g[3 \times 2]} & \mathbf{Y}_{f[3 \times 6]} \end{bmatrix} \begin{bmatrix} \boldsymbol{\pi}_{d[6 \times 1]} \\ \boldsymbol{\pi}_{g[2 \times 1]} \\ \boldsymbol{\pi}_{f[6 \times 1]} \end{bmatrix} \quad (\text{C.13})$$

where $\mathbf{Y} \in \mathbb{R}^{3 \times 14}$ is the regressor matrix (also named observation matrix), $\mathbf{Y}_d \in \mathbb{R}^{3 \times 6}$ is the dynamics-related part, $\mathbf{Y}_g \in \mathbb{R}^{3 \times 2}$ is the gravity-related part, $\mathbf{Y}_f \in \mathbb{R}^{3 \times 6}$ is the friction-related part, $\boldsymbol{\pi} \in \mathbb{R}^{14 \times 1}$ is the extracted dynamic coefficients vector. Note that the minimal linear parameterization of robot dynamics, *i.e.*, the minimal set of dynamic coefficients $\boldsymbol{\pi}$, is not unique.

The identified values of the dynamic coefficients in Eqn. (C.18) and Eqn. (C.19) in the linearized model of the 3DOF Phantom Premium 1.5A robot [137] are summarized in Table C.3.

Based on the linearization results from [137], the regressor matrix $\mathbf{Y}$ (including $\mathbf{Y}_d$, $\mathbf{Y}_g$, $\mathbf{Y}_f$) and the extracted dynamic coefficients $\boldsymbol{\pi}$ are expressed as follows

$$\mathbf{Y}_d = \begin{bmatrix} Y_{d.11} & Y_{d.12} & Y_{d.13} & Y_{d.14} & 0 & 0 \\ 0 & Y_{d.22} & 0 & Y_{d.24} & Y_{d.25} & 0 \\ 0 & 0 & Y_{d.33} & Y_{d.34} & 0 & Y_{d.36} \end{bmatrix} \quad (\text{C.14})$$

Table C.3: The identified values of the dynamic coefficients in the linear model of the 3DOF Phantom Premium 1.5A robot.

Coefficients	Calculated values from [16]	Identified values from [137]
γ_1	$\gamma_1 = 0.0076$	$\gamma_1 = 0.0039$
γ_2	$\gamma_2 = 0.0030$	$\gamma_2 = 0.0037$
γ_3	$\gamma_3 = -0.0011$	$\gamma_3 = -0.0011$
γ_4	$\gamma_4 = 0.0025$	$\gamma_4 = 0.0019$
γ_5	$\gamma_5 = 0.0066$	$\gamma_5 = 0.0057$
γ_6	$\gamma_6 = 0.0025$	$\gamma_6 = 0.0026$
γ_7 (gravity related)	$\gamma_7 = -0.0445$	$\gamma_7 = -0.0525$
γ_8 (gravity related)	$\gamma_8 = -0.2015$	$\gamma_8 = -0.3002$
$\gamma_9 (= \pi_{fv1})$ (friction related)	NC	$\gamma_9 = -0.0057$
$\gamma_{10} (= \pi_{fv2})$ (friction related)	NC	$\gamma_{10} = -0.0035$
$\gamma_{11} (= \pi_{fv3})$ (friction related)	NC	$\gamma_{11} = -0.0005$
$\gamma_{12} (= \pi_{fc1})$ (friction related)	NC	$\gamma_{12} = 0.0707$
$\gamma_{13} (= \pi_{fc2})$ (friction related)	NC	$\gamma_{13} = 0.0251$
$\gamma_{14} (= \pi_{fc3})$ (friction related)	NC	$\gamma_{14} = 0.0248$

Note: NC, not considered.

$$\left\{ \begin{array}{l}
 Y_{d.11} = \ddot{q}_1 \\
 Y_{d.12} = \ddot{q}_1 \cos(2q_2) - 2\dot{q}_1\dot{q}_2 \sin(q_2) \cos(q_2) - \dot{q}_1\dot{q}_2 \sin(2q_2) \\
 Y_{d.13} = \ddot{q}_1 \cos(2q_3) - 2\dot{q}_1\dot{q}_3 \sin(q_3) \cos(q_3) - \dot{q}_1\dot{q}_3 \sin(2q_3) \\
 Y_{d.14} = \ddot{q}_1 \cos(q_2) \sin(q_3) - \dot{q}_1\dot{q}_2 \sin(q_2) \sin(q_3) + \dot{q}_1\dot{q}_3 \cos(q_2) \cos(q_3) \\
 Y_{d.22} = \dot{q}_1^2 \sin(2q_2) \\
 Y_{d.24} = -\frac{1}{2}\ddot{q}_3 \sin(q_2 - q_3) + \frac{1}{2}\dot{q}_1^2 \sin(q_2) \sin(q_3) + \frac{1}{2}\dot{q}_3^2 \cos(q_2 - q_3) \\
 Y_{d.25} = \ddot{q}_2 \\
 Y_{d.33} = \dot{q}_1^2 \sin(2q_3) \\
 Y_{d.34} = -\frac{1}{2}\ddot{q}_2 \sin(q_2 - q_3) - \frac{1}{2}\dot{q}_1^2 \cos(q_2) \cos(q_3) + \frac{1}{2}\dot{q}_2^2 \cos(q_2 - q_3) \\
 Y_{d.36} = \ddot{q}_3
 \end{array} \right. \quad (C.15)$$

$$\mathbf{Y}_g = \begin{bmatrix} 0 & 0 \\ \cos(q_2) & 0 \\ 0 & \sin(q_3) \end{bmatrix} \quad (C.16)$$

$$\mathbf{Y}_f = \begin{bmatrix} \dot{q}_1 & 0 & 0 & \text{sgn}(\dot{q}_1) & 0 & 0 \\ 0 & \dot{q}_2 & 0 & 0 & \text{sgn}(\dot{q}_2) & 0 \\ 0 & 0 & \dot{q}_3 & 0 & 0 & \text{sgn}(\dot{q}_3) \end{bmatrix} \quad (\text{C.17})$$

$$\boldsymbol{\pi} = \begin{bmatrix} \boldsymbol{\pi}_{d[6 \times 1]} \\ \boldsymbol{\pi}_{g[2 \times 1]} \\ \boldsymbol{\pi}_{f[6 \times 1]} \end{bmatrix} = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 & \gamma_7 & \gamma_8 & \gamma_9 & \gamma_{10} & \gamma_{11} & \gamma_{12} & \gamma_{13} & \gamma_{14} \end{bmatrix}^T \quad (\text{C.18})$$

$$\left\{ \begin{array}{l} \gamma_1 = \frac{1}{8}(4I_{ayy} + 4I_{azz} + 8I_{baseyy} + 4I_{beyy} + 4I_{bezz} + 4I_{cyy} + 4I_{czz} \\ \quad + 4I_{dfyy} + 4I_{dfzz} + L_1^2 m_c + L_2^2 m_a + 4L_3^2 m_c + 4L_1^2 m_a) \\ \gamma_2 = \frac{1}{8}(4I_{beyy} - 4I_{bezz} + 4I_{cyy} - 4I_{czz} + 4L_1^2 m_a + L_1^2 m_c) \\ \gamma_3 = \frac{1}{8}(4I_{ayy} - 4I_{azz} + 4I_{dfyy} - 4I_{dfzz} - L_2^2 m_a - 4L_3^2 m_c) \\ \gamma_4 = L_1(L_2 m_a + L_3 m_c) \\ \gamma_5 = \frac{1}{4}(4I_{bexx} + 4I_{cxx} + 4L_1^2 m_a + L_1^2 m_c) \\ \gamma_6 = \frac{1}{4}(4I_{axx} + 4I_{dfxx} + L_2^2 m_a + 4L_3^2 m_c) \\ \gamma_7 = \frac{1}{2}g(2L_1 m_a + 2L_5 m_{be} + L_1 m_c) \\ \gamma_8 = \frac{1}{2}g(L_2 m_a + 2L_3 m_c - 2L_6 m_{df}) \\ \gamma_9 = \pi_{fv1} \\ \gamma_{10} = \pi_{fv2} \\ \gamma_{11} = \pi_{fv3} \\ \gamma_{12} = \pi_{fc1} \\ \gamma_{13} = \pi_{fc2} \\ \gamma_{14} = \pi_{fc3} \end{array} \right. \quad (\text{C.19})$$

where π_{fvi} and π_{fci} are the viscous and Coulomb friction coefficients for the i -th joint ($i = 1, 2, 3$), respectively.