UNIVERSITY OF CALIFORNIA SAN DIEGO

SAN DIEGO STATE UNIVERSITY

Prescribed-Time, Decentralized and Delay-Adaptive Control Strategies for Robot Manipulators: Design and Experiments

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Engineering Sciences (Mechanical and Aerospace Engineering)

by

Alexander Bertino

Committee in charge:

University of California San Diego

Professor Miroslav Krstić, Co-Chair
Professor Tania Morimoto
Professor Michael Yip

San Diego State University

Professor Peiman Naseradinmousavi, Co-Chair
Professor Ping Lu

2022

The Dissertation of Alexander Bertino is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

San Diego State University

2022

DEDICATION

I dedicate this dissertation to my parents Ces and Rita, my brother Robert, my sister Haley, my extended family, and my friends for providing me with unwavering support and encouragement throughout the pursuit of my degree.

TABLE OF CONTENTS

Dissertation Approval Page iii

Dedication iv

Table of Contents v

List of Figures vii

List of Tables x

Acknowledgements xi

Vita xiii

Abstract of the Dissertation xv

Chapter 1 Introduction 1
1.1 Overview of Proposed Methods 4
1.2 Organization 5

Chapter 2 Mathematical Modeling 6
2.1 Design of Reference Trajectory 8

Chapter 3 Model-Free Decentralized-Adaptive Control 9
3.1 Background 9
3.2 Decentralized Model Formulation 13
3.3 Decentralized Adaptive Controller 14
3.3.1 Derivation of Update Law 14
3.4 Simulation Results 19
3.5 Experimental Results 27
3.6 Conclusion 31
3.7 Acknowledgements 31

Chapter 4 Delay-Adaptive Control 33
4.1 Background 33
4.2 Problem Statement 36
4.3 Delay-Adaptive Control Design 39
4.4 Lyapunov Analysis 46
4.5 Remarks on Delay-Adaptive Control Law 60
4.6 Simulation and Experimental Results 62
4.6.1 Trajectory Tracking without Delay Adaptation when the Delay is Underestimated 62
4.6.2 Trajectory Tracking when the Delay is Underestimated 65
4.6.3 Trajectory Tracking when the Delay is Overestimated 68

4.7 Conclusion 71
4.8 Acknowledgements 73

Chapter 5 Prescribed-Time Control 74
5.1 Background 74
5.2 Prescribed-Time Tracking for Robot Manipulators 78
5.3 Lyapunov Analysis 82
5.4 Remarks on Prescribed-Time Control Law 85
5.5 Simulation and Experimental Results 87
5.6 Conclusion 92
5.7 Acknowledgements 92

Chapter 6 Prescribed-Time Safety Filter 94
6.1 Background 95
6.2 Prescribed-Time Safety Filter for Robot Manipulators 98
6.3 Proof of Theorem 6.1 103
6.4 Simulated and Experimental Results 108
6.5 Conclusion 118
6.6 Acknowledgements 119

Bibliography 121

LIST OF FIGURES

Figure 2.1. The 7-DOF Baxter's arm at DSCL 7

Figure 2.2. The joints' configuration: (a) sagittal view; (b) top view 7

Figure 3.1. The experimental (blue line), simulated (green line), and desired (red dashed line) joint trajectories of Baxter 22

Figure 3.2. The experimental (blue line) and simulated (red dashed line) joint torques of Baxter 23

Figure 3.3. The tuning of adaptive gain k_1 during experimentation (blue line) and simulation (red dashed line) of Baxter 24

Figure 3.4. The tuning of adaptive gain z_1 during experimentation (blue line) and simulation (red dashed line) of Baxter 25

Figure 3.5. The tuning of adaptive gain z_2 during experimentation (blue line) and simulation (red dashed line) of Baxter 26

Figure 3.6. The simulated (a) and experimental (b) joint tracking errors of Baxter 27

Figure 3.7. Baxter tracking a desired trajectory under (a) the decentralized adaptive and (b) model-based centralized adaptive control schemes at Dynamic Systems and Control Laboratory (DSCL); see peimannm.sdsu.edu 29

Figure 4.1. The simulated trajectories (a-g) and joint torque input signals (h-n) of Baxter performing a pick and place task without delay adaptation, with an input delay of 0.9s. Simulations are performed at an estimated delay of 0.9s (red dashed line), 0.85s (blue line), 0.8s (yellow line), and 0.78s (green line). 64

Figure 4.2. The experimental (blue line), simulated (green line), and desired (red dashed line) joint trajectories (a-g) of Baxter, as well as the experimental (blue line) and simulated (red dashed line) joint torque input signals (h-n) of Baxter. The input delay of the system is initially underestimated (0s initial prediction, 0.9s actual delay). 66

Figure 4.3. The simulated (a) and experimental (b) joint tracking errors of Baxter. The input delay of the system is initially underestimated (0s initial prediction, 0.9s actual delay). 67

Figure 4.4. The adaptation of the estimated delay in experiment (blue line) and simulation (green line), compared to the actual input delay (red dashed line). The input delay of the system is initially underestimated (0s initial prediction, 0.9s actual delay). 67

Figure 4.5. The experimental (blue line), simulated (green line), and desired (red dashed line) joint trajectories (a-g) of Baxter, as well as the experimental (blue line) and simulated (red dashed line) joint torque input signals (h-n) of Baxter. The input delay of the system is initially overestimated (0.9s initial prediction, 0.4s actual delay). 69

Figure 4.6. The simulated (a) and experimental (b) joint tracking errors of Baxter. The input delay of the system is initially overestimated (0.9s initial prediction, 0.4s actual delay). 70

Figure 4.7. The adaptation of the estimated delay in experiment (blue line) and simulation (green line), compared to the actual input delay (red dashed line). The input delay of the system is initially overestimated (0.9s initial prediction, 0.4s actual delay). 70

Figure 4.8. Baxter performing a pick and place task while subjected to an input delay of 0.9 seconds (0s initial delay estimate). 72

Figure 5.1. The experimental (blue line), simulated (green line), and desired (red dashed line) joint trajectories of Baxter 88

Figure 5.2. The experimental (blue line) and simulated (red dashed line) joint torque input signals of Baxter 89

Figure 5.3. Baxter tracking a desired trajectory under the prescribed-time control scheme, correcting for a large initial tracking error of 20 degrees in each joint and attenuating a sinusoidal torque disturbance of $D(t) = 0.1\sin(5t)$. The green circles represent the reference trajectory to be tracked, and are spaced at approximately 1 second intervals. 90

Figure 5.4. The simulated (a) and experimental (b) joint tracking errors of Baxter, with $D(t) = 0.1\sin(5t)$ 91

Figure 5.5. The simulated (a) and experimental (b) joint tracking errors of Baxter, when subjected to a disturbance with non-zero mean $D(t) = 0.1\sin(5t) + 0.05$ . . 91

Figure 6.1. Simulations (left column) and experiments (right column) of Baxter following a pick-and-place trajectory while avoiding multiple obstacles, using a prescribed-time safety filter (a, b), an exponential safety filter with a high gain of $\rho = 4$ (c, d), and an exponential safety filter with a low gain of $\rho = 1.5$ (e, f). At $t = 3s$, the end-effector trajectory takes a major turn from moving up to moving below the spherical obstacle. 109

Figure 6.2. The simulated (a) and experimental (b) distance between the robot manipulator and the nearest obstacle. 110

Figure 6.3. The magnitude of the difference between the nominal and filtered joint torques in simulation (a) and experiment (b). 111

Figure 6.4. The simulated (a-g) and experimental (h-n) joint trajectories of Baxter. At $t = 3s$, the end-effector trajectory takes a major turn from moving up to moving below the spherical obstacle. 113

Figure 6.5. The simulated (a-g) and experimental (h-n) joint torque input signals of Baxter. At $t = 3s$, the end-effector trajectory takes a major turn from moving up to moving below the spherical obstacle. 114

Figure 6.6. The simulated joint jerks of Baxter, shown when the safety filter is active. At $t = 3s$, the end-effector trajectory takes a major turn from moving up to moving below the spherical obstacle. 115

Figure 6.7. Simulated (left column) and experimental (right column) tracking errors of Baxter when using a prescribed-time safety filter (a, b), an exponential safety filter with a high gain of $\rho = 4$ (c, d), and an exponential safety filter with a low gain of $\rho = 1.5$ (e, f). 116

Figure 6.8. Simulations of Baxter following a pick-and-place trajectory while avoiding multiple obstacles, using a prescribed-time safety filter. In (a), the center of the spherical obstacle is lowered 120mm, resulting in the end-effector going over the spherical obstacle. In (b), the center of the spherical obstacle is lowered exactly 67 mm, resulting in the end-effector being unable to reach its destination. Note that in this case, the end-effector still travels along the surface of the spherical obstacle without exiting the safe set. . . . 118

LIST OF TABLES

Table 2.1. Baxter's Denavit-Hartenberg Parameters 6

Table 3.1. Controller Parameters for Simulation and Experiment 20

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my advisors, Professor Peiman Naseradinmousavi and Professor Miroslav Krstić, whose experience and motivation have been instrumental in the successful completion of my Ph.D.

I would like to thank my committee members, Professor Peiman Naseradinmousavi, Professor Miroslav Krstić, Professor Ping Lu, Professor Tania Morimoto, and Professor Michael Yip for their time and valuable comments.

Chapter 3 contains adaptations of the following papers: 1) Bertino, Alexander, Peiman Naseradinmousavi, and Atul Kelkar. "Experimental and Analytical Decentralized Adaptive Control of a 7-DOF Robot Manipulator." *Dynamic Systems and Control Conference*. Vol. 84270. American Society of Mechanical Engineers, 2020. 2) Bertino, Alexander, Peiman Naseradinmousavi, and Atul Kelkar. "Analytical and experimental decentralized adaptive control of a high-degrees-of-freedom robot manipulator." *Journal of Dynamic Systems, Measurement, and Control* 143.7 (2021). The dissertation author is the primary investigator and author of these papers.

Chapter 4 contains adaptations of the following papers: 1) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Experimental and Analytical Delay-Adaptive Control of a 7-DOF Robot Manipulator." *2021 American Control Conference (ACC)*. IEEE, 2021. 2) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Delay-Adaptive Control of a 7-DOF Robot Manipulator: Design and Experiments." *IEEE Transactions on Control Systems Technology* (2022). The dissertation author is the primary investigator and author of these papers.

Chapter 5 contains adaptations of the following papers: 1) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić."Experimental and Analytical Prescribed-Time Trajectory Tracking Control of a 7-DOF Robot Manipulator." *2022 American Control Conference (ACC)*. IEEE, 2022. 2) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Design and Experiment of a Prescribed-Time Trajectory Tracking Controller for a 7-DOF Robot Manipulator." *Journal of Dynamic Systems, Measurement, and Control*. Under Revision, 2022.

The dissertation author is the primary investigator and author of these papers.

Chapter 6 contains adaptations of the following papers: 1) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Experiment and Design of Prescribed-Time Safety Filter for a 7-DOF Robot Manipulator Using CBF-QP." *Modeling, Estimation and Control Conference*. Under Review, 2022. 2) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Prescribed-Time Safety Filter for a 7-DOF Robot Manipulator: Experiment and Design." *IEEE Transactions on Control Systems Technology*. Under Review, 2022. The dissertation author is the primary investigator and author of these papers.

This manuscript is based upon work supported by the National Science Foundation under Award #1823951-1823983. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

VITA

2017	Bachelor of Science in Mechanical Engineering, University of California San Diego
2018–2022	Research Assistant, Dynamic Systems and Control Laboratory, San Diego State University
2019	Master of Science in Mechanical Engineering, San Diego State University
2022	Teaching Assistant, San Diego State University
2022	Doctor of Philosophy in Engineering Sciences (Mechanical and Aerospace Engineering), University of California San Diego and San Diego State University

PUBLICATIONS

Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić."Experimental and Analytical Prescribed-Time Trajectory Tracking Control of a 7-DOF Robot Manipulator." *2022 American Control Conference (ACC)*. IEEE, 2022.

Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Delay-Adaptive Control of a 7-DOF Robot Manipulator: Design and Experiments." *IEEE Transactions on Control Systems Technology*, In Press, 2022.

Bertino, Alexander, Peiman Naseradinmousavi, and Atul Kelkar. "Analytical and experimental decentralized adaptive control of a high-degrees-of-freedom robot manipulator." *Journal of Dynamic Systems, Measurement, and Control* 143.7 (2021).

Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Experimental and Analytical Delay-Adaptive Control of a 7-DOF Robot Manipulator." *2021 American Control Conference (ACC)*. IEEE, 2021.

Bertino, Alexander, Hashem Ashrafiuon, and Peiman Naseradinmousavi. "Disturbance Attenuation Through Real-Time Optimization of PD Gains for a Two-Link Robot." *IFAC-PapersOnLine* 54.20 (2021): 84-89.

Bagheri, Mostafa, **Bertino, Alexander**, and Peiman Naseradinmousavi. "Experimental and Analytical Nonzero-Sum Differential Game-Based Control of a 7-DOF Robotic Manipulator." *Dynamic Systems and Control Conference*. Vol. 84270. American Society of Mechanical Engineers, 2020.

Bertino, Alexander, Peiman Naseradinmousavi, and Atul Kelkar. "Experimental and Analytical Decentralized Adaptive Control of a 7-DOF Robot Manipulator." *Dynamic Systems and Control Conference*. Vol. 84270. American Society of Mechanical Engineers, 2020.

Bertino, Alexander, Mostafa Bagheri, Miroslav Krstić, and Peiman Naseradinmousavi. "Experimental autonomous deep learning-based 3d path planning for a 7-dof robot manipulator." *Dynamic Systems and Control Conference*. Vol. 59155. American Society of Mechanical Engineers, 2019.

Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Design and Experiment of a Prescribed-Time Trajectory Tracking Controller for a 7-DOF Robot Manipulator." *Journal of Dynamic Systems, Measurement, and Control*. Under Revision, 2022.

Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Experiment and Design of Prescribed-Time Safety Filter for a 7-DOF Robot Manipulator Using CBF-QP." *Modeling, Estimation and Control Conference*. Under Review, 2022.

Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Prescribed-Time Safety Filter for a 7-DOF Robot Manipulator: Experiment and Design." *IEEE Transactions on Control Systems Technology*. Under Review, 2022.

ABSTRACT OF THE DISSERTATION

Prescribed-Time, Decentralized and Delay-Adaptive Control Strategies for Robot Manipulators: Design and Experiments

by

Alexander Bertino

Doctor of Philosophy in Engineering Sciences (Mechanical and Aerospace Engineering)

University of California San Diego, 2022
San Diego State University, 2022

Professor Peiman Naseradinmousavi, Co-Chair
Professor Miroslav Krstić, Co-Chair

In this manuscript, we formulate and experimentally verify four state-of-the-art control strategies on Baxter, a 7-DOF redundant robot manipulator. The control strategies examined in this manuscript are the subject of active research in the field of non-linear control, and have the potential to significantly improve the performance of robot manipulators when they operate in unstructured environments. The first control strategy we investigate in this manuscript is model-free decentralized-adaptive control. The purpose of this control strategy is to achieve consistent performance across a wide range of joint configurations and end-effector inertias, while having a similar computational efficiency as PID approaches. The second control strategy we investigate in this manuscript is delay-adaptive control. The purpose of this control strategy

is to simultaneously estimate and compensate for an unknown long actuator delay. The third control strategy we investigate in this manuscript is prescribed-time control. A key feature of this control strategy is that the settling time is explicitly assigned by the control designer to a value desired, or "prescribed" by the user, and that the settling time is independent of the initial conditions and of the reference signal. The fourth control strategy we investigate in this manuscript is the prescribed-time safety filter. This formation yields a filter that is capable of avoiding multiple obstacles in a minimally invasive manner with bounded joint torques, while simultaneously allowing a nominal controller to converge to positions located on the boundary of the safe set by the end of a fixed-duration task. Through the formulation and experimental verification of each control strategy we present in this manuscript, we demonstrate that our proposed methods perform well in both theory and in practice.

Chapter 1

Introduction

When it comes to performing manual tasks, robot manipulators have many advantages over humans. They are stronger, faster, and more precise than even the most capable of humans. They are able to perform repetitive tasks with remarkable consistency, and are able to maintain this efficiency and reliability over many hours without a degradation in performance. Furthermore, they are able to safely operate in environments that are hazardous to humans, removing the need to put human operators in dangerous environments. Due to these strengths, robot manipulators have seen widespread use in industrial applications during the past century.

The potential applications of robot manipulators are not limited to just industrial purposes. There is active research towards the use of robot manipulators in numerous applications, some examples being interactive robots, rescue operations, medical robots, and space robots. However, in order for robot manipulators to successfully operate in these environments, there are many structural obstacles that must be overcome when compared to their historical use in industrial applications. Broadly speaking, these challenges derive from the manipulator's environment being less consistent and structured when compared to an ideal industrial environment. A common difficulty present in many of these potential applications is uncertainty in the dynamics of the system. When operating in interactive environments, a robot manipulator would be reasonably expected to manipulate a variety of objects with different inertial properties, effectively changing the dynamics of the robot manipulator plus holding object system in a way that can

not be determined a priori since the held object can be effectively arbitrary. Additionally, the manipulator may be subjected to external disturbances which create additional uncertainty in the system dynamics. Furthermore, in cases such as the control of space robots, there can be a long, unknown delay between when a torque signal is generated versus when the robot applies this requested torque, which can significantly alter the dynamics of the resulting system.

Another common difficulty present in many of these potential applications is that the robot manipulator is expected to perform a wide variety of possible tasks, rather than a single repetitive task. As such, the performance of the robot manipulator is much more difficult to verify, as a large amount of experiments with different initial conditions and reference trajectories need to be performed in order to make a reasonable assumption of the performance of the robot manipulator in a general case. Furthermore, due to the non-linear dynamics of robot manipulators, the dynamics of the robot manipulator, including the manipulator's effective inertia, can vary widely based on the current configuration of the robot manipulator. Thus, while a PID controller is capable of globally stabilizing a robot manipulator, a single set of PID gains will only be optimal for a specific joint configuration. In order to maintain acceptable performance across a larger range of joint configurations, one might consider utilizing a gain scheduling PID controller, in which PID gains are determined at multiple configurations based on a linearization of the manipulator dynamics at each of these configurations. However, while this approach is feasible to perform at several points along a single reference trajectory, for a high-DOF manipulator tracking an arbitrary trajectory, the number of such linearizations required is too large to be practical.

An important difficulty present in many of these potential applications is the assurance of safety during the operation of a robot manipulator. In an unstructured environment, there will likely be numerous obstacles that a robot manipulator needs to avoid collision with while performing its required tasks. Furthermore, for applications such as interactive robots, the robot manipulator must also avoid harmful collisions with humans, which can be reasonably expected to be in the robot manipulator's environment. As collision between a robot manipulator and a

human can be fatal due to the robot manipulator's large inertia, avoiding dangerous collisions is a necessity for robot manipulators used in such applications. Thus, in order to operate robot manipulators in uncertain environments, the development of sophisticated nonlinear control strategies is necessary.

In this manuscript, we formulate and experimentally verify four state-of-the-art control strategies on Baxter, a 7-DOF redundant robot manipulator. The control strategies examined in this manuscript are the subject of active research in the field of non-linear control, and have the potential to significantly improve the performance of robot manipulators when they operate in unstructured environments. It is important to note that in most research manuscripts concerning the control of robot manipulators, only simulations are present with no experimental verification. In manuscripts that do contain experimental verification, it is typically performed on robot manipulators with relatively low degrees of freedom, or which several joints remain fixed for the duration of the experiment to achieve a similar outcome. Through experimental verification, factors that are not present in experiments such as measurement noise, external disturbances, potential modeling inaccuracies due to effects such as friction, and time delays caused by computation of the control law can all affect the performance of a given control strategy, and thus a control method that appears to work on paper may fail when implemented on a real system. Additionally, robot manipulators with a high-DOF have much more complicated dynamics than low-DOF manipulators, increasing the computational burden of computing their dynamics. This additional computational burden can cause control strategies that are effective for low-DOF robot manipulators to be infeasible to implement on high-DOF manipulators at the necessary control frequency. Thus, through the experimental verification of each control strategy we present in this manuscript, we demonstrate that our proposed methods perform well in both theory and in practice.

1.1 Overview of Proposed Methods

The first control strategy we investigate in this manuscript is model-free decentralized-adaptive control. The purpose of this control strategy is to achieve consistent performance across a wide range of joint configurations and end-effector inertias, while having a similar computational efficiency as PID approaches. It is important to note that this method is based only on the tracking errors experienced during operation of the manipulator, and thus does not require a precise dynamic model in order to achieve desirable performance.

The second control strategy we investigate in this manuscript is delay-adaptive control. The purpose of this control strategy is to simultaneously estimate and compensate for an unknown long actuator delay. Through simulations and experiments, we demonstrate that the proposed controller is capable of tracking the desired trajectory with desirable performance despite a large initial delay mismatch, which would cause non-adaptive prediction-based controllers to become unstable.

The third control strategy we investigate in this manuscript is prescribed-time control. A key feature of this control strategy is that the settling time is explicitly assigned by the control designer to a value desired, or "prescribed" by the user, and that the settling time is independent of the initial conditions and of the reference signal. Through both simulation and experiment, we demonstrate that the proposed controller is capable of converging to the desired trajectory within the prescribed time, in spite of large initial tracking errors, and in spite of a sinusoidal disturbance being applied in each joint.

The fourth control strategy we investigate in this manuscript is the prescribed-time safety filter. This formation yields a filter that is capable of avoiding multiple obstacles in a minimally invasive manner with bounded joint torques, while simultaneously allowing a nominal controller to converge to positions located on the boundary of the safe set by the end of a fixed-duration task. The results of our simulations and experiments demonstrated the ability of the prescribed-time safety filter to enforce safety throughout the six second task, while allowing the robot manipulator

to make contact with the boundary and arrive at the desired goal position by the end of the task.

1.2 Organization

The organization of this manuscript is as follows. In Chapter 2, we present a brief overview of the dynamics of Baxter's right manipulator, as well as the design of the reference trajectory utilized for the verification of each presented control strategy. In Chapters 3-6, we formulate and verify through simulation and experiment the performance of the model-free decentralized-adaptive approach, delay-adaptive approach, prescribed-time approach, and prescribed-time safety filter approach respectively.

Chapter 2

Mathematical Modeling

The redundant manipulator, which is being studied here, has 7-DOF as shown in Figure 2.1. The Baxter manipulator's Denavit-Hartenberg parameters are shown in Table 2.1 provided by the manufacturer. The Euler-Lagrange formulation leads to a set of 7 coupled nonlinear second-order ordinary differential equations:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau \tag{2.1}$$

where, $q, \dot{q}, \ddot{q} \in \mathbb{R}^7$ are angles, angular velocities and angular accelerations of joints, respectively, and $\tau \in \mathbb{R}^7$ indicates the vector of joints' driving torques. Also, $M(q) \in \mathbb{R}^{7\times 7}$ is a symmetric mass-inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{7\times 7}$ is a matrix of Coriolis coefficients, $G(q) \in \mathbb{R}^7$ is a vector of gravitational loading, and $F(\dot{q}) \in \mathbb{R}^7$ represents a vector of frictional torques.

Table 2.1. Baxter's Denavit-Hartenberg Parameters

Link	a_i	d_i	α_i	q_i
1	0.069	0.27035	$-\pi/2$	q_1
2	0	0	$\pi/2$	$q_2+\pi/2$
3	0.069	0.36435	$-\pi/2$	q_3
4	0	0	$\pi/2$	q_4
5	0.010	0.37429	$-\pi/2$	q_5
6	0	0	$\pi/2$	q_6
7	0	0.3945	0	q_7

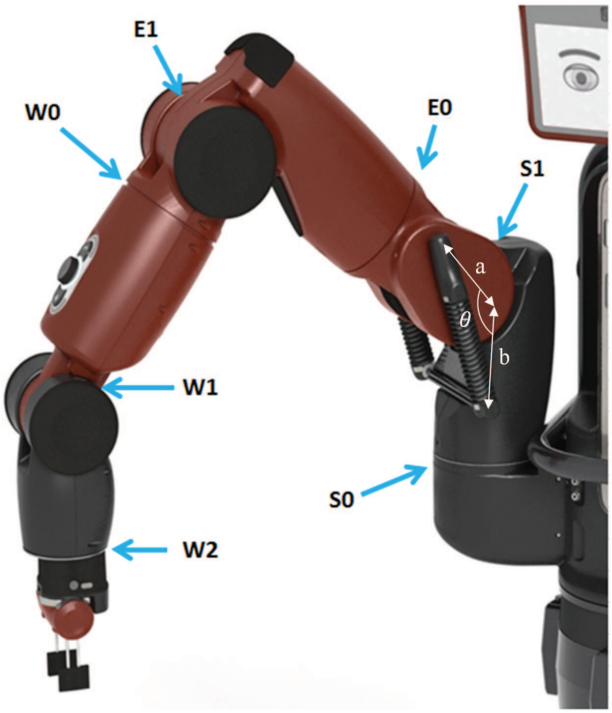
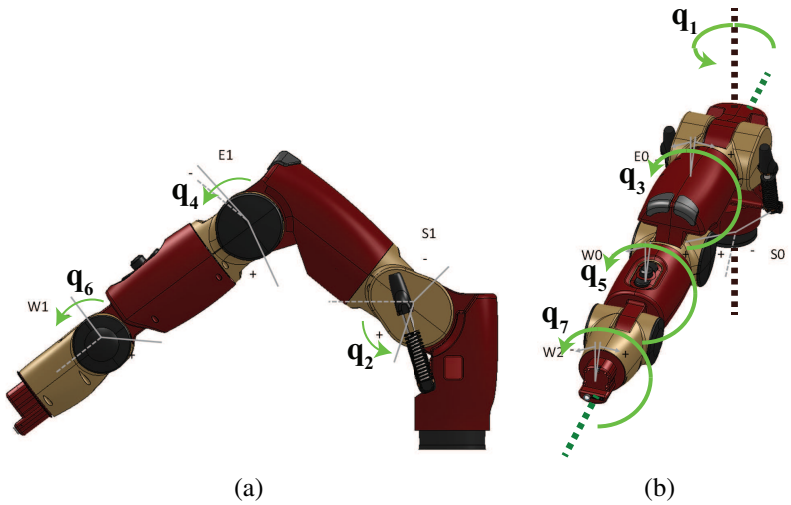


Figure 2.1. The 7-DOF Baxter's arm at DSCL



(a) (b)

Figure 2.2. The joints' configuration: (a) sagittal view; (b) top view

Our verified coupled nonlinear dynamic model of the robot [1–13] is used as the basis of each control strategy investigated in this manuscript.

2.1 Design of Reference Trajectory

For each of the four control methods present in this manuscript, we evaluate their performance on the same reference trajectory. This six second reference trajectory is a parabolic curve that passes through the points $p_0 = (x_0, y_0, z_0)$, $p_1 = (x_0, y_0, z_0 + \Delta z)$, $p_2 = (x_0, y_0 + \Delta y, z_0)$, and is designed to mimic the common pick-and-place task of robot manipulators. To generate the parabolic curve, we employ the following quartic spline method:

$$p_r(t) = \begin{cases} a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4, & \text{if } t \leq 3 \\ a_5 + a_6 t + a_7 t^2 + a_8 t^3 + a_9 t^4, & \text{if } t > 3 \end{cases} \tag{2.2}$$

where $a_0, a_1, ..., a_9 \in \mathbb{R}^3$ are constants to be determined by applying the necessary boundary conditions. These conditions are as follows:

$$\begin{aligned} &p_r(0) = p_0, \quad p_r(3) = p_1, \qquad p_r(6) = p_2, \qquad \dot{p}_r(0) = 0, \qquad \dot{p}_r(6) = 0 \\ &\ddot{p}_r(0) = 0, \quad \ddot{p}_r(6) = 0, \quad \lim_{t \to 3^+} p_r(t) = p_1, \quad \lim_{t \to 3^+} \dot{p}_r(t) = \dot{p}_r(3), \quad \lim_{t \to 3^+} \ddot{p}_r(t) = \ddot{p}_r(3) \end{aligned} \tag{2.3}$$

Once we have determined the desired trajectory in Cartesian coordinates, we proceed to convert this trajectory into individual joint trajectories $q_r(t)$ via integration of the following ODE from $t = 0$ to $t = 6$:

$$\dot{q}_r(t) = J^+ \left(q_r(t) \right) \dot{p}_r(t) \tag{2.4}$$

where $J(q) \in \mathbb{R}^{6 \times 7}$ is the manipulator jacobian, and $J^+(q) \in \mathbb{R}^{7 \times 6}$ is its Moore-Penrose pseudoinverse. This method is the well-known least squares method, and yields the minimum joint velocities necessary for Baxter's end-effector to follow the reference trajectory.

Chapter 3

Model-Free Decentralized-Adaptive Control

In this chapter, we present a model-free decentralized adaptive control strategy for the tracking control of the manipulator. The problem formulation and experimental results demonstrate the computational efficiency and simplicity of the proposed method. The results presented here are one of the first known experiments on a redundant 7-DOF robot. The efficacy of the adaptive decentralized controller is demonstrated experimentally by using the Baxter robot to track a desired trajectory. Simulation and experimental results clearly demonstrate the versatility, tracking performance, and computational efficiency of this method.

3.1 Background

As the global trend is towards increased automation, robot manipulators have seen widespread use in many industrial applications. While the research in adaptive and nonlinear control has seen significant advances, most robot manipulators utilized in industry are driven by simple decentralized PID controllers due to their simplicity in their design and implementation [14, 15]. While these controllers are effective at driving robot manipulators to specific set points, they have difficulty in tracking an arbitrary desired trajectory. Furthermore, due to the strong interconnected nonlinearities inherently present in the dynamic model of such systems, a given set of PID gains will only work well for a specific joint configuration and end-effector mass.

While many pick-and-place type operations in industry not needing navigation through obstacles can be performed effectively using PID type controllers, the tasks requiring sophisticated path planning and tracking need advanced controls. In order to maintain acceptable performance across a larger range of joint configurations, one might consider utilizing a gain scheduling PID controller, such as presented in [16, 17]. While these controllers can theoretically achieve desirable performance under such circumstances, most implementations of these controllers will require determining acceptable PID gains for a multitude of linearized models at different operating conditions. For a 7-DOF manipulator tracking an arbitrary trajectory, the number of such linearizations required will be too large and cumbersome. Additionally, such a method would not account for an unknown end-effector mass. As society looks towards the use of robot manipulators that can interact with humans in social settings, rescue operations, and potential medical applications, the requirement that such manipulators must adhere to an arbitrary desired trajectory during motion becomes an important task. The decentralized adaptive control approach presented here provides one effective control strategy for high performance robot operations for which PID control might not give desirable performance. Such an approach retains much of the simplicity and computational efficiency of the decentralized PID approach, while offering a wide range of applicability with extended joint configuration space and variability of end-effector masses.

Due to the strength of the dynamic interconnection between joints, a model-based approach in which the system is split into a set of decoupled systems is not feasible for robot manipulators. Instead, there are several different methods designed to work around this constraint to achieve desirable performance. First, neural network based methods [18, 19], as well as the disturbance observer method by Yang *et al.* [20], and model-reference method such as by Sundareshan and Koenig [21], attempt to obtain a model of certain system behaviors during the operation of the robot manipulator. Such adaptive-model based methods do not suffer from unmodeled system dynamics, and are well suited for tasks in which the joint dynamics change during the operation of a task. Another popular approach to the decentralized adaptive control

of robot manipulators is the model-free approach [22–25], in which the adaptive control law is governed purely from the performance of the manipulator in the tracking task. Model-free approaches, such as that by Seraji [22], can bear strong similarity to the decentralized PID approach. In such approaches, the static gains associated with the PID approach are replaced with adaptive gains, that change during the execution of the task to better track the desired trajectory. Other research efforts for decentralized control of various systems can be found in [26–42].

The goal of this chapter is to develop a control formulation and conduct an experimental verification of the model-free decentralized adaptive method using Baxter, a 7-DOF redundant robot manipulator. This work is novel in that the experimental verification of a decentralized adaptive controller for a 7-DOF manipulator is not currently addressed in literature. The decentralized adaptive control of such a manipulator is an important and challenging task. The increased degrees of freedom of the robot manipulator leads to an increased dynamic interconnection between joints, which is a challenge for decentralized approaches. Also, the Baxter arm configuration is a more likely choice for the complex tasks to be performed in an industrial setting. Through the analytical formulation and experimental verification of the decentralized adaptive approach, we seek to demonstrate the feasibility and computational effectiveness of said approach, in order to facilitate its adoption into industry practices. For this purpose, the model-free decentralized adaptive approach examined in this paper is an effective choice, as its structure is similar to the decentralized PID controllers currently utilized in industry.

It is important to mention the existence of a similar model-free decentralized approach, known as Model-Free Control (MFC). MFC is a decentralized method developed in order to compensate for uncertainties in nonlinear systems, and has been shown to be effective in the control of many uncertain dynamical systems, including robot manipulators [43–45]. In order to compensate for these uncertainties, such as changes in the inertia matrix during the motion of the robot manipulator, the uncertainties are estimated directly utilizing the torque and state information from the previous timestep. In addition to this compensation, a decentralized PID

controller is typically employed to drive the robot manipulator towards the desired trajectory. Thus, this method is capable of adjusting to a wide range of operating conditions without needing to tune adaptive gains during the procedure. However, despite the simplicity and effectiveness of this method, it has a few important drawbacks, especially when considering its possible implementation on Baxter. First, in order to estimate the nonlinear uncertainties of a robot manipulator with the MFC approach, it is necessary to numerically calculate the angular acceleration of each joint. This numerical approximation of the second derivative is highly susceptible to noise, as it amplifies the noise already encountered when calculating the angular velocity of each joint. Thus, the uncertainty compensation employed by MFC is succeptible to noise when implemented on a robot manipulator. Second, calculating the system uncertainties based on data from the previous timestep introduces bias into the uncertainty estimate. This bias can be large when the system dynamics change quickly, such as changes in frictional terms when the angular velocity of a joint changes sign. Additionally, the controller timestep must be sufficiently small in order to make the bias negligible. As Baxter is typically sampled at 100 Hz, this sampling rate may not be fast enough to ensure a low enough bias. This bias introduces a disturbance in the manipulator error dynamics that can lead to imperfect tracking of the desired trajectory. Due to these potential drawbacks of the MFC method when applied to robot manipulators, the authors believe the model-free decentralized adaptive approach studied here to be the more promising method for the decentralized control of Baxter.

The rest of this chapter is organized as follows. In Section 3.2, we present a decentralized model of Baxter's joint dynamics, as well as the structure of the model-free decentralized adaptive approach. In Section 3.3, we utilize Lyapunov's method to derive the update law for the adaptive gains of the controller, demonstrating asymptomatic stability in the process. In Section 3.4, we demonstrate and analyze the performance of the decentralized adaptive approach on a simulation of Baxter executing the desired trajectory, paying close attention to tracking performance, controller effort, and selection of adaptive gains. In Section 3.5, we repeat the same procedure on the Baxter robot in practice, and thoroughly compare the experimental

performance to that derived from the simulation. Finally, in Section 3.6, we present the case that the decentralized adaptive method is computationally efficient, simple to implement, effective at tracking a desired trajectory, and is a desirable alternative to both decentralized PID controllers and centralized controllers for robot manipulators.

3.2 Decentralized Model Formulation

In order to derive the decentralized adaptive controller, it is necessary to model the dynamics of a single joint, rather than the system as a whole. Rewriting (2.1) as series of 7 differential equations yields:

$$m_{ii}(q)\ddot{q}_i + \left[\sum_{i=1, j\neq i}^{n} m_{ij}(q)\ddot{q}_j\right] + c_i(q,\dot{q})\dot{q} + g_i(q) + F_i(\dot{q}) = T_i(t) \tag{3.1}$$

where m_{ij} is the element in the mass matrix located at (i, j), $c_i(q,\dot{q})$ is the ith row of the Coriolis matrix, $g_i(q)$ is the ith element of the gravity vector, $T_i(t)$ is the input torque at joint i, and $F_i(\dot{q})$ is the frictional torque at joint i. Note that this equation represents the angular acceleration at joint i as a function of the input torque only at joint i, and the dynamics of each link $q, \dot{q}, \ddot{q}$. Thus, (2.1) can be reduced to a series of 7 dynamically interconnected SISO systems. In order to further express this concept, we rewrite (3.1) as:

$$m_{ii}(q)\ddot{q}_i + d_i(q,\dot{q},\ddot{q}) = T_i(t) \tag{3.2}$$

where $d_i(q,\dot{q},\ddot{q}) = \left[\sum_{i=1, j\neq i}^{n} m_{ij}(q)\ddot{q}_j\right] + c_i(q,\dot{q})\dot{q} + g_i(q) + F_i(\dot{q})$ represents the dynamic interconnection between joints.

3.3 Decentralized Adaptive Controller

In order to track an arbitrary desired trajectory, we employ the following decentralized adaptive control structure:

$$T_i(t) = f_i(t) + k_{i1}(t)e_i(t) + k_{i2}(t)\dot{e}_i(t) + z_{i1}(t)\dot{q}_{ri}(t) + z_{i2}(t)\ddot{q}_{ri}(t) \tag{3.3}$$

where $q_{ri}(t)$ is the desired reference trajectory, $e_i(t) = q_{ri}(t) - q_i(t)$ is the tracking error, and $f_i(t), k_{i1}(t), k_{i2}(t), z_{i1}(t), z_{i2}(t)$ are adaptive control signals to be determined through the application of Lyapunov methods. In this formulation, $f_i(t)$ is termed the auxiliary signal, and is the primary driver of the system state $q_i, \dot{q}_i$ towards the desired trajectory. $k_{i1}(t), k_{i2}(t)$ are adaptive PD gains intended to account for current error in the tracking performance, adjusting to the dynamics of the current joint configuration. Similarly, $z_{i1}(t), z_{i2}(t)$ are adaptive feedforward velocity and acceleration gains, intended to ensure that the joint stays on the desired trajectory.

3.3.1 Derivation of Update Law

In order to derive the equations of the adaptive control signals, we first make the following assumption:

Assumption 3.1. *The mass element m_{ii}, and the dynamic interconnection between the joints $d_i(q, \dot{q}, \ddot{q})$, are slowly time varying with respect to the desired trajectory $q_{ri}(t)$. That is, $\dot{m}_{ii} \approx 0$ and $\dot{d}_i \approx 0$.*

Utilizing this assumption, the decentralized model (3.2), and the controller law (3.3), we can express the model plus controller dynamics as:

$$m\ddot{q} + d = f + k_1 e + k_2 \dot{e} + z_1 \dot{q}_r + z_2 \ddot{q}_r \tag{3.4}$$

Note that the *i*th subscript, as well as notations indicating functions of time and joint configuration

$(t, q, \dot{q}, \ddot{q})$, have been removed for the sake of notational simplicity. This equation can be rearranged to obtain:

$$m\ddot{e} + k_2\dot{e} + k_1 e = d - f - z_1\dot{q}_r + (m - z_2)\ddot{q}_r \tag{3.5}$$

Furthermore, defining the error state vector as $X = [e, \dot{e}]^T$, (3.5) can be rewritten in state-space form to obtain:

$$\dot{X} = \begin{bmatrix} 0 & 1 \\ \frac{-k_1}{m} & \frac{-k_2}{m} \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{d-f}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-z_1}{m} \end{bmatrix} \dot{q}_r + \begin{bmatrix} 0 \\ \frac{m-z_2}{m} \end{bmatrix} \ddot{q}_r \tag{3.6}$$

In order to ensure that the robot manipulator follows the desired trajectory, we define the desired performance of the tracking error $e_s(t)$, which we define with the following 2^{nd} order homogeneous differential equation:

$$\ddot{e}_s + 2\xi\omega_n\dot{e}_s + \omega_n^2 e_s = 0 \tag{3.7}$$

where ω_n is the natural frequency of the desired performance and ξ is the damping ratio. Similarly to (3.5), we define the reference state vector $X_s = [e_s, \dot{e}_s]^T$, and rewrite (3.7) in state space form to obtain:

$$\dot{X}_s = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} X_s = AX_s \tag{3.8}$$

Next, we use the following theorem to prove a crucial property of the reference model (3.8).

Theorem 3.1. *Consider the linear state-space model $\dot{x} = Ax$. The equilibrium $x = 0$ is globally asymptotically stable if and only if $\exists P = P^T > 0$, $\exists Q = Q^T > 0$ such that the following Lyapunov equation holds:*

$$PA + A^T P = -Q \tag{3.9}$$

Since we are free to define ξ and ω_n in such a manner as to ensure (3.8) is globally asymptotically stable, then by Theorem 1 there exists a unique symmetric positive definite matrix P that solves (3.9) for the linear system (3.8). We denote the elements in P:

$$P = \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \tag{3.10}$$

Next, we define $E = X_s - X$, and combine (3.6) and (3.8) to obtain the tracking error state-space model:

$$\dot{E} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{bmatrix} E + \begin{bmatrix} 0 & 1 \\ \frac{k_1}{m} - \omega_n^2 & \frac{k_2}{m} - 2\xi\omega_n \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{f-d}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{z_1}{m} \end{bmatrix} \dot{q}_r + \begin{bmatrix} 0 \\ \frac{z_2-m}{m} \end{bmatrix} \ddot{q}_r \tag{3.11}$$

In order to determine the stability properties of (3.11), it is first necessary to define a Lyapunov function for the system. For this system, we define the following Lyapunov function:

$$\begin{aligned} V = E^T PE + Q_0(\frac{f-d}{m} - f^*)^2 + Q_1(\frac{k_1}{m} - \omega_n^2 - k_1^*)^2 + Q_2(\frac{k_2}{m} - 2\omega_n\xi - k_2^*)^2 \\ + Q_3(\frac{z_1}{m} - z_1^*)^2 + Q_4(\frac{z_2-m}{m} - z_2^*)^2 \end{aligned} \tag{3.12}$$

where $Q_0, ..., Q_4$ are positive scalars, and $f^*, k_1^*, k_2^*, z_1^*, z_2^*$ are functions of time to be determined later. Differentiating (3.12) with respect to time and applying Assumption 3.1 yields:

$$\begin{aligned} \dot{V} = -E^T QE + 2\left(\frac{f-d}{m}\right)\left[Q_0(\frac{\dot{f}}{m} - \dot{f}^*) - r\right] - 2Q_0 f^*\left(\frac{\dot{f}}{m} - \dot{f}^*\right) \\ + 2\left(\frac{k_1}{m} - \omega_n^2\right)\left[Q_1(\frac{\dot{k_1}}{m} - \dot{k}_1^*) - re\right] - 2Q_1 k_1^*\left(\frac{\dot{k_1}}{m} - \dot{k_1}^*\right) \\ + 2\left(\frac{k_2}{m} - 2\xi\omega_n\right)\left[Q_2(\frac{\dot{k_2}}{m} - \dot{k}_2^*) - re\dot{}\right] - 2Q_2 k_2^*\left(\frac{\dot{k_2}}{m} - \dot{k_2}^*\right) \\ + 2\left(\frac{z_1}{m}\right)\left[Q_3(\frac{\dot{z_1}}{m} - \dot{z}_1^*) - r\dot{q}_r\right] - 2Q_3 z_1^*\left(\frac{\dot{z_1}}{m} - \dot{z_1}^*\right) \\ + 2\left(\frac{z_2-m}{m}\right)\left[Q_4(\frac{\dot{z_2}}{m} - \dot{z}_2^*) - r\ddot{q}_r\right] - 2Q_4 z_2^*\left(\frac{\dot{z_2}}{m} - \dot{z_2}^*\right) \end{aligned} \tag{3.13}$$

where $r = P_2 e + P_3 \dot{e}$ is the weighted error. Before continuing the derivation, we make note of the following theorem:

Theorem 3.2. *Let $X \in \mathbb{R}^n = 0$ be an equilibrium point of the system $\dot{x} = f(x)$, and let $V : \mathbb{R}^n \to \mathbb{R}$:*

1. *If $V(0) = 0$, $V(X) > 0 \; \forall X \neq 0$, $\dot{V} \leq 0 \; \forall X \neq 0$, then $X = 0$ is globally stable*
2. *If $V(X) \to \infty$ as $|| X || \to \infty$, then $V(X)$ is radially unbounded*
3. *If $X = 0$ is stable, $V(X)$ is radially unbounded, and $\dot{V} < 0 \; \forall X \neq 0$, then $X = 0$ is globally asymptotically stable*

We first note that per our definition of V in (3.12), V is both positive when $E \neq 0$ and radially unbounded. Thus, we seek to derive adaptation parameters f, k_1, k_2, z_1, z_2, and undetermined parameters $f^*, k_1^*, k_2^*, z_1^*, z_2^*$ such that $\dot{V}$ is negative definite, and thus $E = 0$ is globally asymptotically stable. First, we set the following terms in (3.13) to 0:

$$\begin{aligned} &Q_0(\frac{\dot{f}}{m} - \dot{f}^*) - r = 0, \quad Q_1(\frac{\dot{k_1}}{m} - \dot{k}_1^*) - re = 0, \quad Q_2(\frac{\dot{k_2}}{m} - \dot{k}_2^*) - r\dot{e} = 0 \\ &Q_3(\frac{\dot{z_1}}{m} - \dot{z}_1^*) - r\dot{q}_r = 0, \quad Q_4(\frac{\dot{z_2}}{m} - \dot{z}_2^*) - r\ddot{q}_r = 0 \end{aligned} \tag{3.14}$$

Substituting (3.14) into (3.13) yields the following equation:

$$\dot{V} = -E^T Q E - 2f^* r - 2k_1^* re - 2k_2^* r\dot{e} - 2z_1^* r\dot{q}_r - 2z_2^* r\ddot{q}_r \tag{3.15}$$

We then define the following terms:

$$f^* = Q_0^* r, \quad k_1^* = Q_1^* re, \quad k_2^* = Q_2^* r\dot{e}, \quad z_1^* = Q_3^* r\dot{q}_r, \quad z_2^* = Q_4^* r\ddot{q}_r \tag{3.16}$$

where $Q_0^*, ..., Q_4^*$ are positive scalars. Substituting (3.16) into (3.15) yields:

$$\dot{V} = -E^T Q E - 2Q_0^* r^2 - 2Q_1^* r^2 e^2 - 2Q_2^* r^2 \dot{e}^2 - 2Q_3^* r^2 \dot{q}_r^2 - 2Q_4^* r^2 \ddot{q}_r^2 \tag{3.17}$$

which is negative for all $E \neq 0$, thus Theorem 2 is satisfied and $E = 0$ is globally asymptotically stable. However, we must now determine the values of the parameters f, k_1, k_2, z_1, z_2 to satisfy (3.14), which are as follows:

$$\begin{aligned}
\dot{f} &= mQ_0^*\dot{r} + \frac{m}{Q_0}r \\
\dot{k}_1 &= mQ_1^*\frac{d}{dt}(re) + \frac{m}{Q_1}re \\
\dot{k}_2 &= mQ_2^*\frac{d}{dt}(r\dot{e}) + \frac{m}{Q_2}r\dot{e} \\
\dot{z}_1 &= mQ_3^*\frac{d}{dt}(r\dot{q}_r) + \frac{m}{Q_3}r\dot{q}_r \\
\dot{z}_2 &= mQ_4^*\frac{d}{dt}(r\ddot{q}_r) + \frac{m}{Q_4}r\ddot{q}_r
\end{aligned} \tag{3.18}$$

We then define the following terms so that (3.18) is independent of m:

$$\begin{aligned}
&Q_0^* = \frac{\rho}{m}, \quad Q_0 = \frac{m}{\delta}, \quad Q_1^* = \frac{\beta_1}{m}, \quad Q_1 = \frac{m}{\alpha_1} \\
&Q_2^* = \frac{\beta_2}{m}, \quad Q_2 = \frac{m}{\alpha_2}, \quad Q_3^* = \frac{\lambda_1}{m}, \quad Q_3 = \frac{m}{\gamma_1} \\
&Q_4^* = \frac{\lambda_2}{m}, \quad Q_4 = \frac{m}{\gamma_2}
\end{aligned} \tag{3.19}$$

Substituting (3.19) into (3.18) and integrating with respect to time yields the following equations for the decentralized adaptive parameters:

$$\begin{aligned}
f(t) &= f(0) + \delta\int_0^t r(t)dt + \rho r(t) \\
k_1(t) &= k_1(0) + \alpha_1\int_0^t r(t)e(t)dt + \beta_1 r(t)e(t) \\
k_2(t) &= k_2(0) + \alpha_2\int_0^t r(t)\dot{e}(t)dt + \beta_2 r(t)\dot{e}(t) \\
z_1(t) &= z_1(0) + \gamma_1\int_0^t r(t)\dot{q}_r(t)dt + \lambda_1 r(t)\dot{q}_r(t) \\
z_2(t) &= z_2(0) + \gamma_2\int_0^t r(t)\ddot{q}_r(t)dt + \lambda_2 r(t)\ddot{q}_r(t)
\end{aligned} \tag{3.20}$$

Now that we have successfully derived the decentralized adaptive gains, we make the following

notes of its structure. First, the auxiliary signal can be interpreted as a decentralized PID signal, acting to guide the system towards the desired trajectory in a generalized approach. Second, each adaptive gain is updated based on the performance of the signal it multiplies in (3.3), as well as the weighted error. This update law is purely performance based, and does not rely on a model of the system. Finally, the update of each parameter is a simple computation, where a trapezoidal approximation can be used to estimate the value of the integral at each time step.

3.4 Simulation Results

In order to assess the performance of this decentralized adaptive controller, we first apply the control law described in Section 3.3 to Baxter's dynamic model (2.1). We apply our control methodology to a tracking problem where the desired tracking trajectories are specified in Chapter 2. In this simulation, we introduce a sampling rate of 100 Hz in order to effectively model the effect of discrete sampling on the continuous-time controller. Furthermore, the controller parameters we used during this simulation, can be observed in Table 3.1.

In order to determine the controller parameters to implement, the following general procedure can be performed:

1. Initialize all controller parameters to 0.

2. Choose parameters p_2, p_3, δ, ρ of the auxiliary signal $f(t)$ such that the controller adequately tracks the desired trajectory across multiple different joint configurations, ensuring satisfactory general performance. It is important to note that the auxiliary signal is equivalent to a PID controller with the following gains:

$$\begin{aligned} \mathbf{K}_P &= \delta p_3 + \rho p_2 \\ \mathbf{K}_I &= \delta p_2 \\ \mathbf{K}_D &= \rho p_3 \end{aligned} \tag{3.21}$$

Table 3.1. Controller Parameters for Simulation and Experiment

Joint	1	2	3	4	5	6	7
p_{i2}	1	1	1	1	1	1	1
p_{i3}	0.3	0.3	0.3	0.3	0.3	0.3	0.3
δ_i	30	60	40	30	7	40	2
ρ_i	30	60	40	30	7	6	2
α_{i1}	6000	6000	6000	6000	10200	102000	1200
β_{i1}	600	600	600	600	1020	10200	120
α_{i2}	6	6	6	6	6	6	6
β_{i2}	0.6	0.6	0.6	0.6	0.6	0.6	0.6
γ_{i1}	60	60	60	60	60	60	60
λ_{i1}	6	6	6	6	6	6	6
γ_{i2}	60	60	60	60	60	60	60
λ_{i2}	6	6	6	6	6	6	6

Thus, traditional techniques used to tune the decentralized PID controllers can be used in order to determine the parameters of the auxiliary signal.

3. Choose the minimum values of the parameters α_1, β_1 of the adaptive proportional gain $k_1(t)$ that reduce tracking error in the angular position signal during motion of the robot manipulator to a desired amount. At this stage, choose the same parameters for each joint, and set $\beta_1 = \alpha_1/10$.

4. Adjust α_{i1}, β_{i1} of each joint individually if a specific joint updates too slowly or too quickly.

5. Repeat steps 3-4 on the parameters α_2, β_2 of the adaptive derivative gain $k_2(t)$ in order to reduce the tracking error in the angular velocity signal during motion to a desirable amount.

6. Repeat steps 3-4 on the parameters γ_1, λ_1 of the adaptive feedforward velocity gain $z_1(t)$ in order to adequately counteract the effect of friction in the beginning of motion.

7. Repeat steps 3-4 on the parameters γ_2, λ_2 of the adaptive feedforward acceleration gain $z_2(t)$ in order to adequately overcome the robot manipulator's inertia in the regions that the desired acceleration is large.

The simulated joint trajectories, along with the desired joint trajectories can be observed in Figure 3.1. From these graphs, it can be seen that the decentralized adaptive controller achieves close tracking of the desired trajectories. Although the effects of the simulated frictional torque and gravity negligibly impact the tracking performance during the beginning of motion, as can be seen in the performance of joints 3, 5, and 6, these effects are quickly accounted for by the adaptive controller. Furthermore, despite large changes in the joint configuration throughout the course of the operation, the performance based control scheme remains effective at consistently driving each joint towards the desired trajectory. These behaviors can also be observed in Figure 3.6, as the tracking error remains less than 1.5 degrees for all joints after 1.5 seconds of operation.

The torques generated by the decentralized adaptive controller can be observed in Figure 3.2. It is important to note that these torques are significantly lower than the maximum torque output of Baxter's joints, which are 50 Nm for joints 1-4, and 15 Nm for joints 5-7, meaning that saturation of torque is not an issue for this decentralized adaptive scheme. Furthermore, this demonstrates energy efficiency of this control scheme, as the torques generated are consistently small in magnitude. Additionally, it can be observed that the torques generated are smooth throughout the operation, which is potentially beneficial to the motors that are used to generate these torques in practice.

Finally, we observe the tuning of adaptive gains k_1, z_1, and z_2 throughout the simulation, as seen in Figures 3.3, 3.4, and 3.5, respectively. Each of these gains appear to adjust in 2 stages ($0s < t < 3s$ and $3s < t < 6s$). These phases correspond to the picking up and placing down motion of the end manipulator, signifying that a different set of gains is necessary for each task.

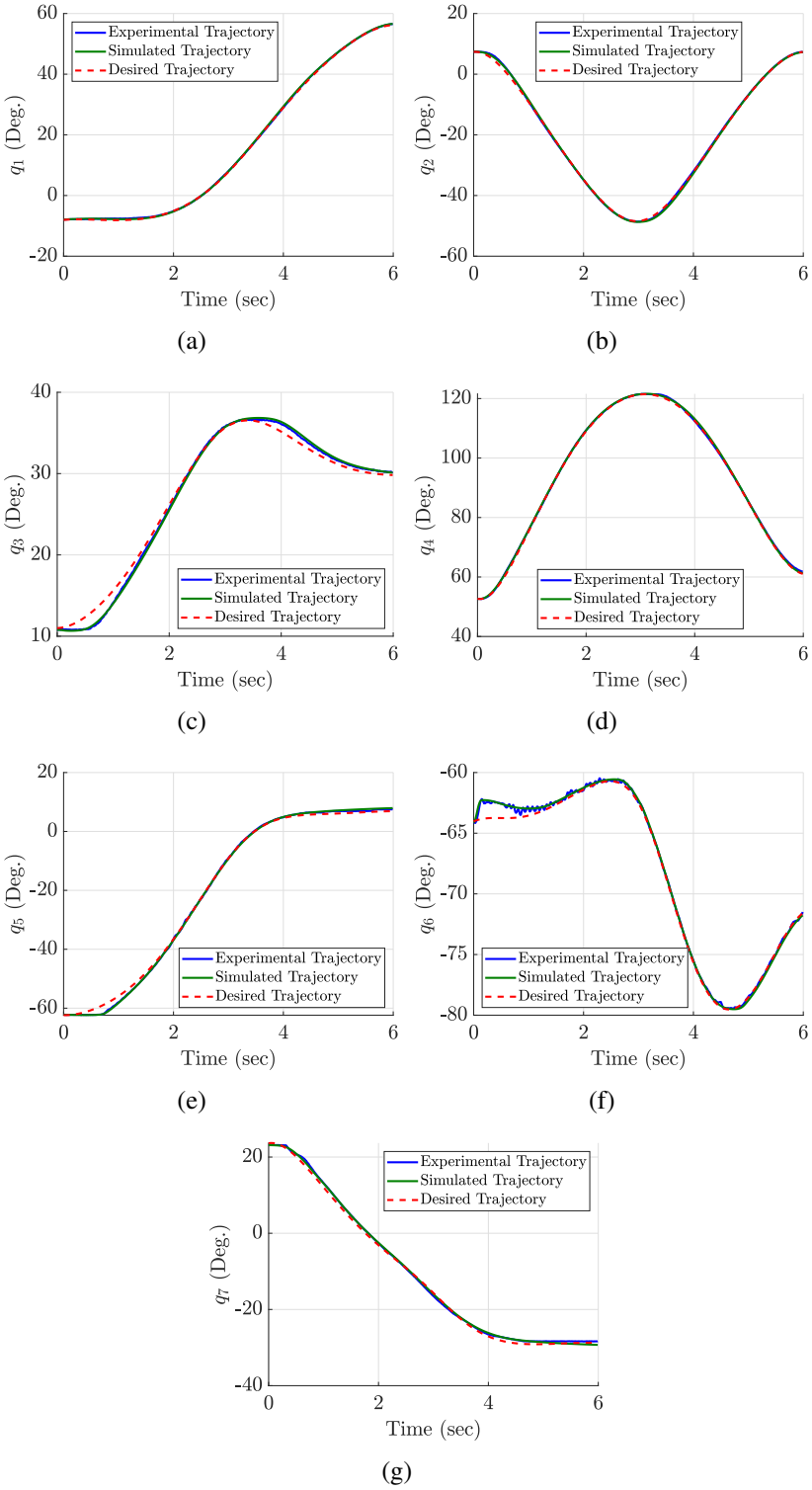


Figure 3.1. The experimental (blue line), simulated (green line), and desired (red dashed line) joint trajectories of Baxter

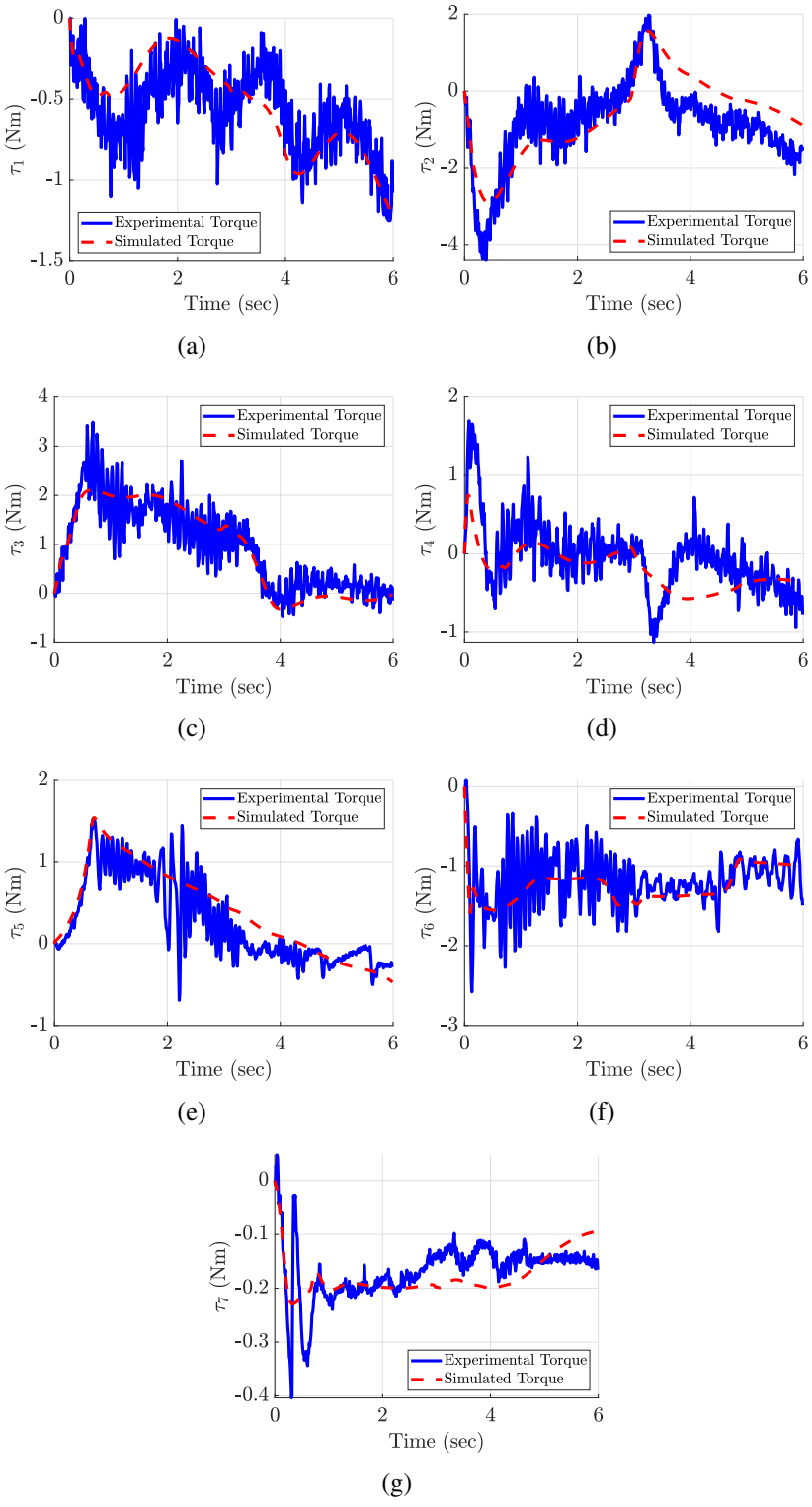


Figure 3.2. The experimental (blue line) and simulated (red dashed line) joint torques of Baxter

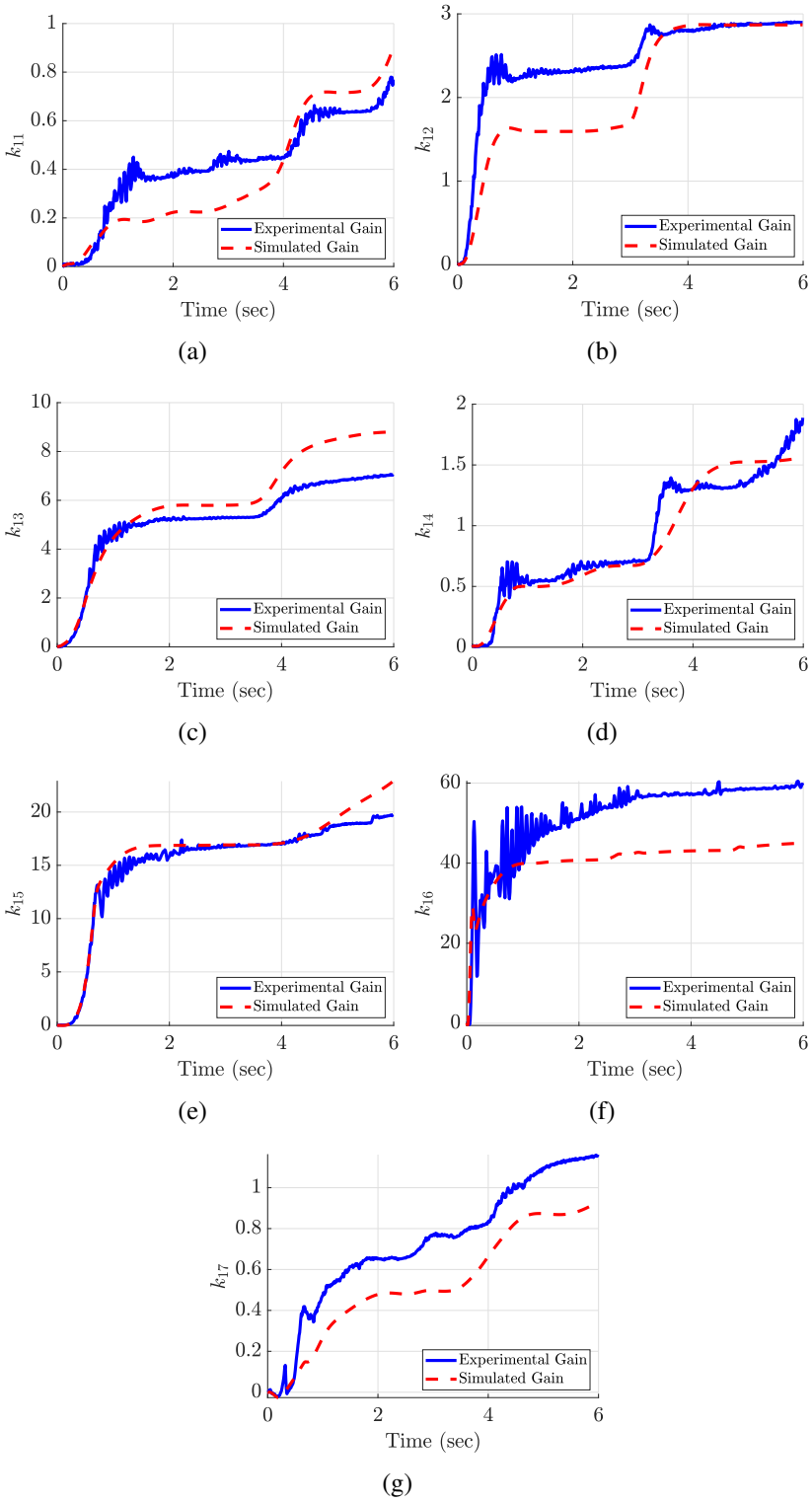


Figure 3.3. The tuning of adaptive gain k_1 during experimentation (blue line) and simulation (red dashed line) of Baxter

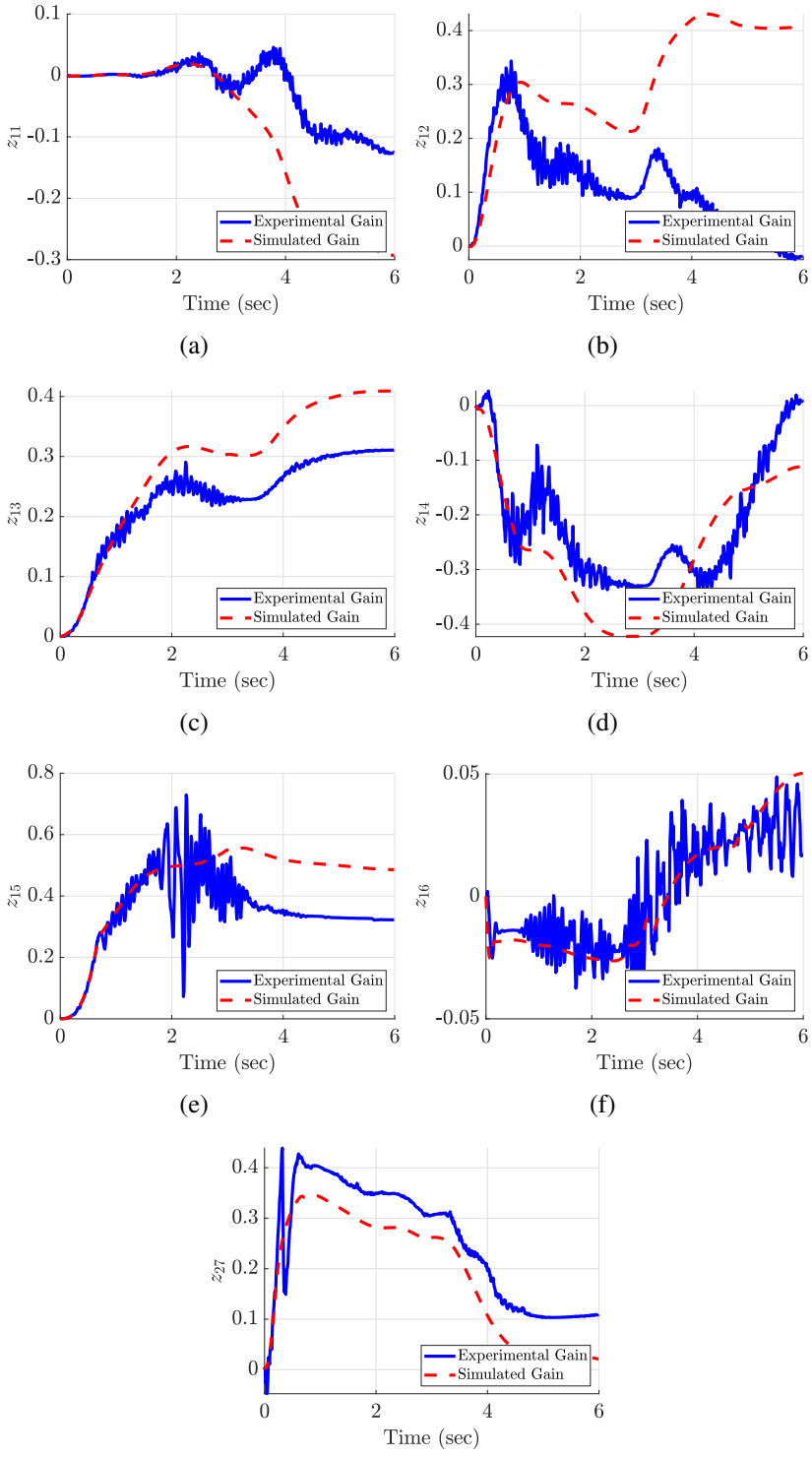


Figure 3.4. The tuning of adaptive gain z_1 during experimentation (blue line) and simulation (red dashed line) of Baxter

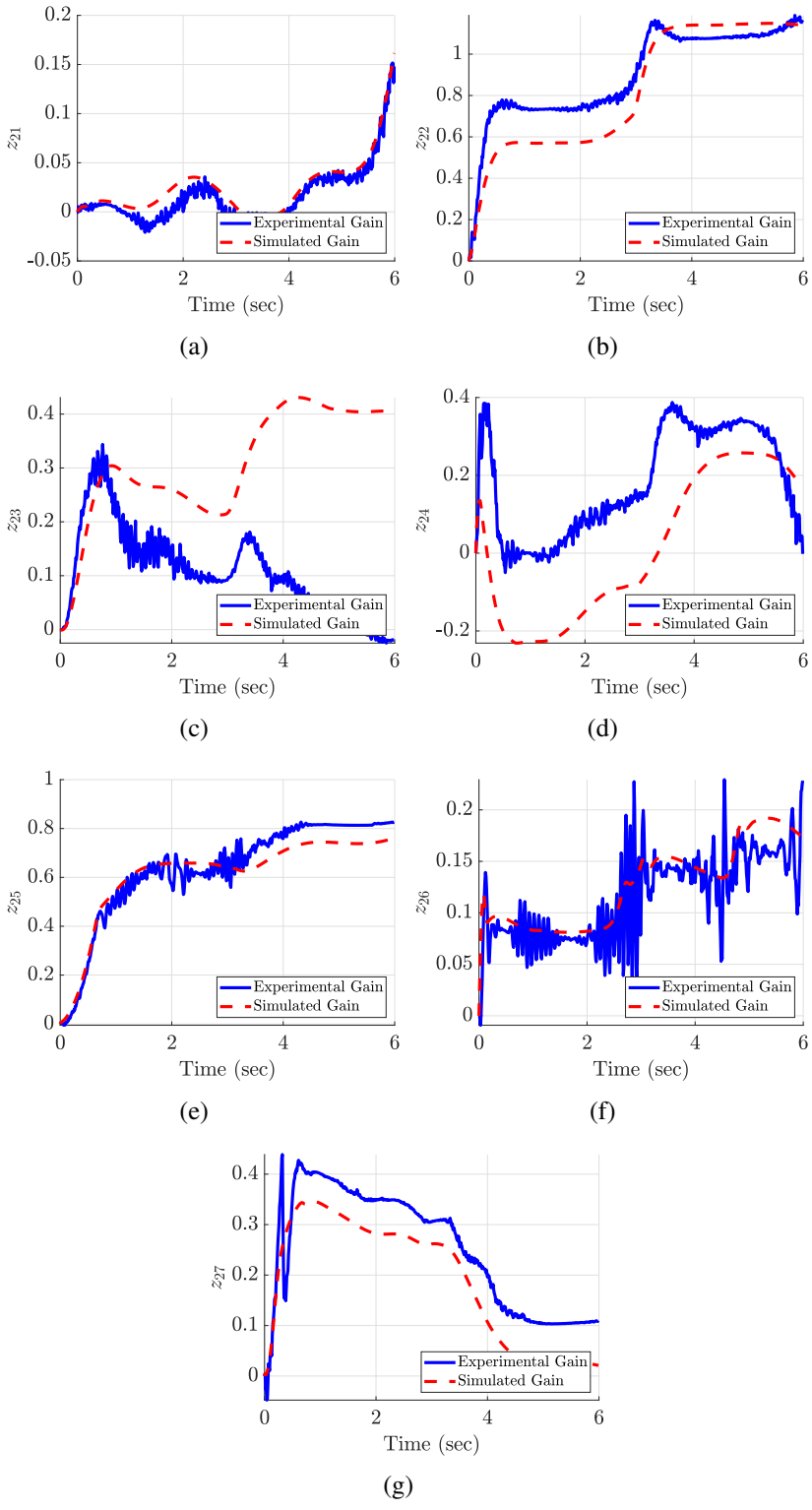


Figure 3.5. The tuning of adaptive gain z_2 during experimentation (blue line) and simulation (red dashed line) of Baxter

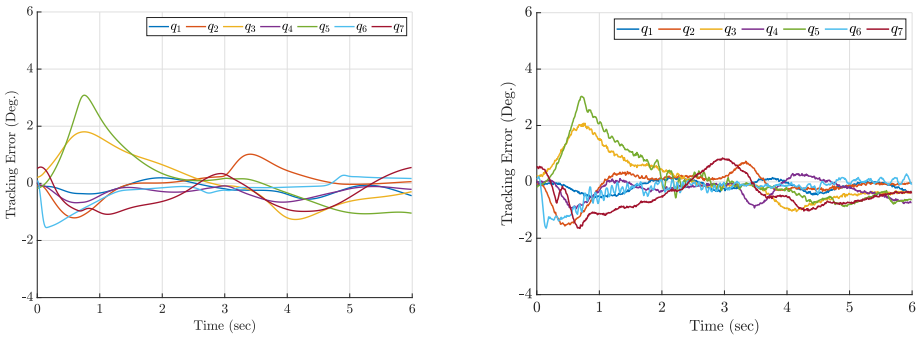


Figure 3.6. The simulated (a) and experimental (b) joint tracking errors of Baxter

Thus, the tuning of these parameters coincide with our expectations of their performance. It is also important to note that these gains are of a significant magnitude when compared to the auxiliary parameters δ_i and ρ_i, meaning that tunings were necessary in order to achieve the desired tracking performance. Furthermore, the joints 3, 5, and 6 with significantly tuned gains experienced the largest frictional torques and gravitational load. These results demonstrate the ability of the decentralized adaptive controller to adjust to different operating conditions. This beneficial quality of this scheme is of key importance when the robot manipulator is expected to reliably perform in a changing environment. From these results, it is evident that the decentralized adaptive controller is effective in simulation.

3.5 Experimental Results

Due to promising results during simulation, we now implement the control law described in Section 3.3 to Baxter in an experimental study. We utilize the same desired trajectories as in Section 3.4 with the same 100 Hz sampling rate. Note that several differences remain between the simulated and experimental study, which include measurement noise in the joint positions and velocities, differences between the idealistic Coulomb Friction model and the actual friction dynamics, small potential inaccuracies in model parameters, and the actuator dynamics of each joint. These factors can lead to results slightly different than those experienced in simulation.

For the experimental pick-and-place task, the controller parameters we used are the same as that of the simulation, and can be observed in Table 3.1.

From Figure 3.7(a), it can be observed that the decentralized adaptive controller is successful at executing the pick-and-place task in practice. The experimental joint trajectories, along with the desired joint trajectories can be observed in Figure 3.1. From these graphs, it can be seen that the decentralized adaptive controller exhibits close tracking of the desired trajectory, that is almost identical to that experienced during simulation. Similar to the Experiment, it can be observed from the graphs that errors experienced in the beginning of the operation are quickly accounted for, and the controller returns to near perfect tracking. This behavior can also be observed in Figure 3.6, as the tracking error remains less than 1.5 degrees after 1.5 seconds of operation.

The torques generated by the decentralized adaptive controller can be observed in Figure 3.2. While the presence of noise in measurements has caused similar variations in the joint torques, the torques still exhibit moderate continuity, as well as a magnitude much lower than the saturation torque of each joint. It can be seen from these graphs that the overall shape and magnitude of the experimental torque of each joint matches closely to that of the corresponding simulated torques. Thus, the differences in system dynamics between the simulation and experiment do not significantly affect the performance of the decentralized control algorithm.

Finally, we observe the tuning of the adaptive gains k_{1i}, z_{1i}, and z_{2i} throughout the experiment, as seen in Figures 3.3, 3.4, and 3.5, respectively. The behavior of these graphs is similar to that of the simulation in regards to both the stages of tuning, as well as the magnitude of the gains. Slight differences can be observed between the evolution of the gains in the simulation and experiment, which can reasonably be attributed to the small differences in dynamics between the simulated and actual system, such as the difference between the idealistic Coulomb Friction model from the friction experienced in the real system. While these differences lead to the selection of different gains from simulation, the overall performance of the decentralized adaptive controller is not significantly affected by this difference in dynamics, as can be seen in Figures

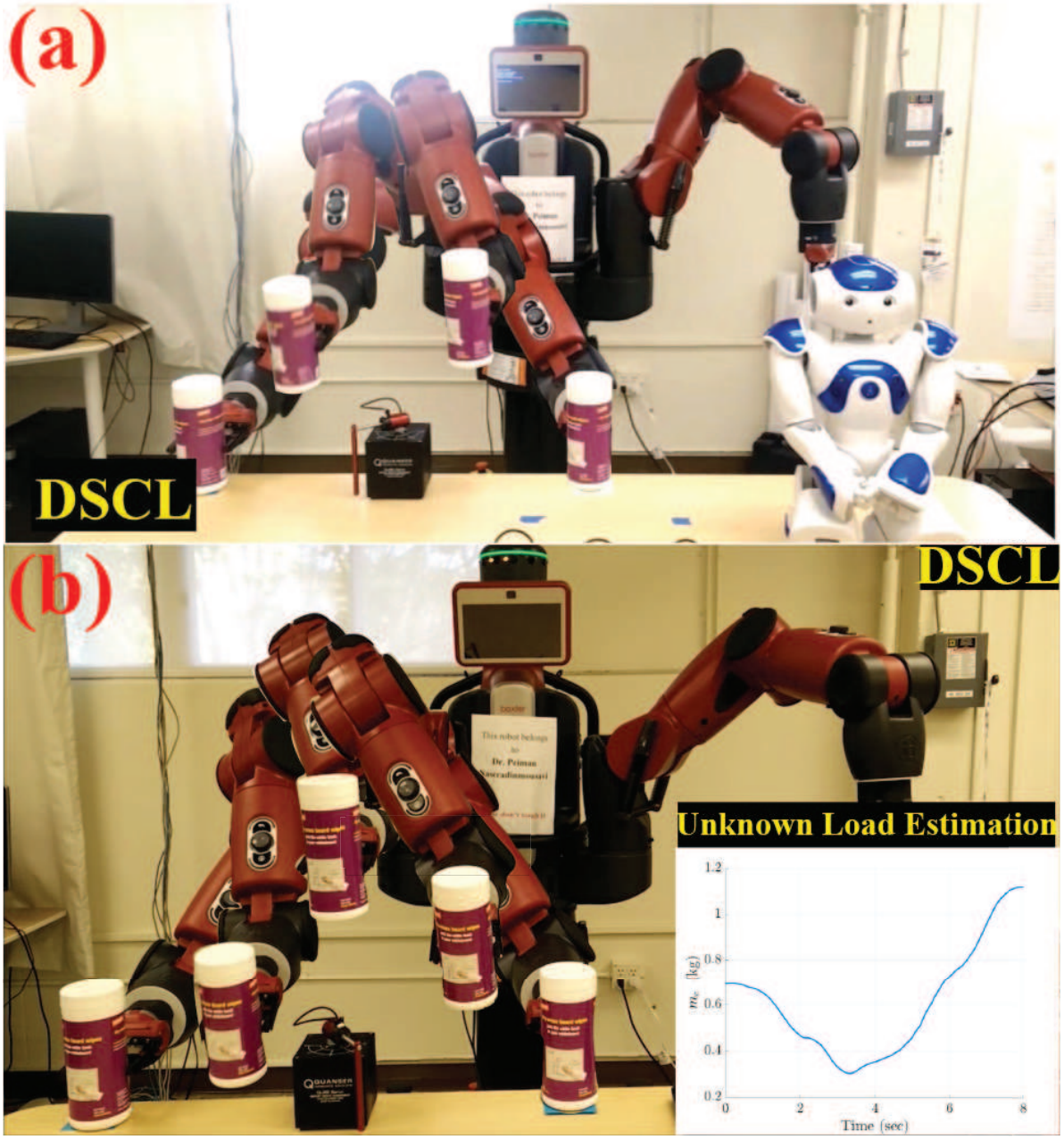


Figure 3.7. Baxter tracking a desired trajectory under (a) the decentralized adaptive and (b) model-based centralized adaptive control schemes at Dynamic Systems and Control Laboratory (DSCL); see peimannm.sdsu.edu

3.1 and 3.2. Thus, these adaptive gains are effective at maintaining desirable performance outside of the conditions in which the decentralized adaptive controller was designed. From these results, it is evident that the decentralized adaptive scheme performs well in experiments as well as in simulation.

Another crucial point to consider is the computational efficiency of the decentralized adaptive scheme compared to centralized ones. We previously carried out experimental work for a simple centralized model-based adaptive scheme to carry an unknown mass avoiding an obstacle, shown in Fig. 3.7(b). The mass of the end-effector was the only unknown parameter to be estimated and we again employed the Damped Least Squares method to calculate desirable joint-space trajectories. The immediate challenge was the computation time of the control scheme in each loop, even when dealing with only one uncertainty, which was incompatible with the minimum time step ($\Delta t_b = 0.001s$ or $f_b = 1kHz$) of Baxter. The computation time of the centralized model-based adaptive scheme was in the range of $\mathbf{0.005s} \leq t_c \leq \mathbf{0.007s}$ leading to the time delay in each control loop. Therefore, we had to address a critical trade-off between the accuracy required and computational cost. To resolve this problem, we increased the Baxter's time step to $\Delta t_b = 0.01s$ or $f_b = 0.1kHz$, along with the sleep command of Python, in order to avoid such a time delay by sacrificing the accuracy needed. Shown in Fig. 3.7(b) is the experimental implementation of the centralized adaptive control of Baxter carried out at the DSC laboratory. We noticed that the estimation of even one uncertainty, without any external disturbance, caused at least three small operational interruptions. Please check the DSCL YouTube Channel, at https://youtu.be/4XWldAXpJ2I, for the AVI file. Note that the decentralized adaptive scheme examined here reveals a significantly lower computation time of $\Delta t_b = \mathbf{1.02\ ms}$ compared to the centralized one. Therefore, we did not observe the operational interruptions discussed for the centralized method whereas the decentralized scheme is at least five times faster than the centralized one. This would be highly beneficial for when we intend to control large-scale (high-DOF) systems.

3.6 Conclusion

In this chapter, we investigated the performance of a model-free decentralized adaptive controller on a 7-DOF redundant manipulator. We first formulated the theory behind the controller, demonstrating the global asymptotic stability of each local controller, as well as revealing the computationally efficient method of adapting each control parameter. Then, through the results of both our simulation and experiment of the decentralized adaptive controller implemented on Baxter, we demonstrated the following beneficial properties of the control scheme:

1. The algorithm is highly computationally efficient and at least five times faster than the centralized adaptive method examined here.
2. Close tracking of the desired trajectory is achieved throughout operation.
3. Large changes in the joint configuration throughout the procedure do not significantly affect the operation.
4. The generated torques are energy efficient, and do not pose the risk of torque saturation.
5. The control scheme can adapt to, and is effective outside of the conditions in which it was designed for.

Thus, we verified the effectiveness of the model-free decentralized adaptive control scheme, and noted its promising potential for a wide variety of applications.

3.7 Acknowledgements

This chapter contains adaptations of the following papers: 1) Bertino, Alexander, Peiman Naseradinmousavi, and Atul Kelkar. ”Experimental and Analytical Decentralized Adaptive Control of a 7-DOF Robot Manipulator.” *Dynamic Systems and Control Conference*. Vol. 84270. American Society of Mechanical Engineers, 2020. 2) Bertino, Alexander, Peiman

Naseradinmousavi, and Atul Kelkar. "Analytical and experimental decentralized adaptive control of a high-degrees-of-freedom robot manipulator." *Journal of Dynamic Systems, Measurement, and Control* 143.7 (2021). The dissertation author is the primary investigator and author of these papers.

This chapter is based upon work supported by the National Science Foundation under Award #1823951. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Chapter 4

Delay-Adaptive Control

We present an analytical design and experimental verification of trajectory-tracking control of a 7-DOF robot manipulator with an unknown long actuator delay. In order to compensate for this unknown delay, we formulate a delay-adaptive prediction-based control strategy in order to simultaneously estimate the unknown delay while driving the robot manipulator towards the desired trajectory. To the best of the authors' knowledge, this chapter is the first to present a delay-adaptive approach for a nonlinear system with multiple inputs. Through Lyapunov analysis, we first establish local input-to-state stability with respect to temporal derivatives of the reference trajectory, along with regulation of the tracking errors when the reference trajectory approaches a stationary configuration. Then, through both simulation and experiment, we demonstrate that the proposed controller is capable of tracking the desired trajectory with desirable performance despite a large initial delay mismatch, which would cause non-adaptive prediction-based controllers to become unstable.

4.1 Background

In this chapter, we pursue an experimental verification of an analytically designed control of a 7-DOF robot manipulator subjected to an unknown constant input delay. Such delays are frequently observed in the control of remote manipulators [46–50], where a long, slowly time-varying (often assumed to be constant) communication delay is likely present. While

input delays can potentially be beneficial in certain special cases, such as in the research of Ulsoy [51], they are typically severely detrimental to the stability of closed-loop systems. In order to account for a known delay, a variety of predictor-based and sliding mode approaches have been developed for linear systems [52–61], nonlinear systems [6, 9, 62–64], as well as systems with a time-varying delay [65–69]. Although such techniques notably improve the transient performance of a controller in the presence of a known delay, they are also well-known to be sensitive to delay mismatch. In the case where a long, slowly time-varying communication delay is difficult to accurately predict or measure, a delay-adaptive control approach has the potential to significantly increase the transient performance of the robot manipulator, through the compensation of the delay mismatch.

In recent papers [70–78], adaptive control strategies were developed in order to estimate an unknown delay while simultaneously compensating for this delay with a predictor-based approach. In order to achieve this, most of these papers represent the constant delay at the input of an Ordinary Differential Equation (ODE) as a transport Partial Differential Equation (PDE) whose convective speed is inversely proportional to the unknown delay. This approach introduces the delay parameter into the model in a linear manner and is, therefore, suitable for adaptive design. In the paper [78] by Bresch-Pietri and Krstić, this strategy was extended to nonlinear dynamics subjected to a constant input delay. The authors examined the case of a measured distributed input, in which a global delay-adaptive stability result is achieved, as well as the more realistic case of an unmeasured distributed input, in which a local delay-adaptive stability result is achieved.

In this chapter, we formulate the local technique developed by Bresch-Pietri and Krstić [78] for an unmeasured distributed input, in order to handle the case of trajectory tracking with multiple actuators. This formulation yields local input-to-state stability with respect to temporal derivatives of the reference trajectory, as well as regulation of tracking errors when the reference trajectory approaches a stationary configuration. Furthermore, through the experimental verification of this delay-adaptive control strategy on Baxter, we demonstrate desirable controller

performance even in the presence of a significant delay mismatch. Two cases are studied here, an underestimation of 0.9 seconds (0s initial prediction, 0.9s actual delay), as well as an overestimation of 0.5 seconds (0.9s initial prediction, 0.4s actual delay). Thus, the delay-adaptive control strategy is both theoretically sound and effective in practice, significantly improving the tracking performance of the predictor-based approach when the delay is unknown.

The organization of this chapter is as follows. In Section 4.2, we formulate the delay adaptation task in mathematical terms, as well as state several assumptions on the system dynamics, feedback law, and desired trajectories that are utilized in the Lyapunov analysis of the delay-adaptive method. In Sections 4.3-4.5, we present the delay-adaptation approach, and demonstrate the local delay-adaptive stability of the method through a Lyapunov analysis utilizing the $\mathscr{L}_1$ norm. In Section 4.6, we present the simulation and experimental results of the proposed method implemented on Baxter's right manipulator, accounting for a large delay mismatch both in the case of delay underestimation and overestimation. Finally, in Section 4.7, we present the case that the proposed delay-adaptive method has the potential to significantly increase the transient performance of a robot manipulator subjected to an unknown delay, through the compensation of an initial delay mismatch.

Notations: In the following, we use the common definitions of class $\mathscr{K}$ and $\mathscr{K}_\infty$ given in [79]. $|\cdot|$ and $|\cdot|_1$ refers to the Euclidean and $\mathscr{L}_1$ norms respectively, the matrix norm is defined accordingly, for $M \in \mathscr{M}_\ell(\mathbb{R})(\ell \in \mathbb{N}^*)$, as $|M| = \sup_{|x|\leq 1} |Mx|$ and the spatial $\mathscr{L}_1$ norm is defined as follows:

$$\left\|u(t)\right\|_1 = \int_0^1 \left|u(x,t)\right|_1 dx$$

For $(a,b) \in \mathbb{R}^2$ such that $a < b$, we define the standard projection operator on the interval $[a,b]$ as a function of two scalar arguments f (denoting the parameter being updated) and g (denoting

the nominal update law) in the following manner:

$$\text{Proj}_{[a,b]}(f,g) = g \begin{cases} 0 & \text{if } f = a \text{ and } g < 0 \\ 0 & \text{if } f = b \text{ and } g > 0 \\ 1 & \text{otherwise} \end{cases}$$

For a distributed function of (x,t) or (y,t), a lowercase subscript indicates differentiation by the corresponding parameter. For example:

$$f_{xt}(x,t) = \frac{\partial^2 f(x,t)}{\partial x \partial t}$$

4.2 Problem Statement

The multi-input nonlinear system (2.1) can be written as 14th-order system of ODEs with the following general state-space form:

$$\dot{X} = f_0(X,U) = \begin{bmatrix} \dot{q} \\ -M(q)^{-1}N(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} 0 \\ M(q)^{-1} \end{bmatrix} U \tag{4.1}$$

$$X = [q_1, \cdots, q_7, \dot{q}_1, \cdots, \dot{q}_7]^T \tag{4.2}$$

$$N(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) \tag{4.3}$$

where $X \in \mathbb{R}^{14}$ is the 14-dimensional vector of states, and $U = \tau \in \mathbb{R}^7$ represents the input torques to the system (4.1).

In order to track a desired trajectory, we reformulate (4.1) in terms of the error dynamics:

$$\dot{Z} = f(Z,U,X_R) = \begin{bmatrix} \dot{z} \\ -M(z+q_R)^{-1}N(z+q_R,\dot{z}+\dot{q}_R) - \ddot{q}_R \end{bmatrix} + \begin{bmatrix} 0 \\ M(z+q_R)^{-1} \end{bmatrix} U \tag{4.4}$$

$$X_R = [q_R^T, \dot{q}_R^T, \ddot{q}_R^T]^T \tag{4.5}$$

$$Z = [z^T, \dot{z}^T]^T \tag{4.6}$$

$$z = q - q_R \tag{4.7}$$

where $X_R \in \mathbb{R}^{21}$ is the state reference trajectory, $Z \in \mathbb{R}^{14}$ is the state error vector, $z \in \mathbb{R}^7$ is the positional error of the robot manipulator, and $q_R \in \mathbb{R}^7$ are the reference joint trajectories to track.

Furthermore, we make the following assumption regarding the reference joint trajectories:

Assumption 4.1. *The desired joint trajectories $q_R(t) \in \mathbb{R}^7$ are class $\mathscr{C}^5$ functions and $X_R(t)$, $\dot{X}_R(t)$, and $\ddot{X}_R(t)$ are uniformly bounded for all $t \geq 0$.*

Consider the following nonlinear plant:

$$\dot{Z}(t) = f(Z(t), U(t-D), X_R(t)) \tag{4.8}$$

in which D is an unknown delay introduced to the error dynamic model of the Baxter manipulator (4.4), belonging to the interval $[\underline{D}, \overline{D}]$, with $\underline{D} > 0$. The objective of the delay-adaptive approach is to stabilize the error dynamics with input delay (4.8), despite the length of the delay being initially unknown. In order to assist the Lyapunov stability analysis in the next section, the following assumptions are made regarding the nonlinear plant (4.8) the corresponding feedback law, and the state reference trajectory $X_R(t)$.

Assumption 4.2. *The plant (4.4) is forward complete.*

Assumption 4.3. *There exists a $\mathscr{C}^2$ feedback law $U = \kappa(Z, X_R)$ such that the closed-loop delay-free plant (4.4) is globally exponentially stable, i.e. there exist $\lambda > 0$ and a class $\mathscr{C}^1$ radially unbounded positive definite function V such that for all $Z \in \mathbb{R}^{14}$*

$$\frac{\partial V}{\partial Z}(Z) f(Z, \kappa(Z, X_R), X_R) \leq -\lambda V(Z) \tag{4.9}$$

$$|Z|^2 \leq V(Z) \leq c_1 |Z|^2 \tag{4.10}$$

$$\left|\frac{\partial V}{\partial Z}(Z)\right| \leq c_2|Z| \tag{4.11}$$

Assumption 4.4. *Values for the state reference trajectory $X_R(t)$ are known at least $\overline{D}$ seconds in advance.*

Assumption 4.2 assures that (4.8) does not escape in finite time. This assumption is necessary to ensure that the system does not escape before the input $U(t-D)$ reaches the system, and has been proven to hold for robot manipulators [6, 9]. Assumption 4.3 is a stronger than necessary condition used in order to prove the local stability of the delay-adaptive approach. Assumption 4.4 ensures that the state error vector $Z(t)$ can be predicted up to $\overline{D}$ seconds in advance. This is a necessary assumption for any predictor-based control strategy involving trajectory tracking since if $Z(t+D)$ can not be predicted, $U(t)$ can not be chosen to compensate for the delay present in the system. While such a control strategy is technically non-causal with respect to the state reference $X_R(t)$, this is not a concern in practice since $X_R(t)$ is a user-defined signal that is independent of the current joint state $X(t)$ and input $U(t)$, and thus can be determined an arbitrary time in advance.

To analyze the closed-loop stability despite delay uncertainties, we use the systematic Lyapunov tools introduced in [62] and first reformulate plant (4.8) in the form:

$$\dot{Z}(t) = f(Z(t), u(0,t), X_R(t)) \tag{4.12}$$

$$Du_t(x,t) = u_x(x,t) \tag{4.13}$$

$$u(1,t) = U(t) \tag{4.14}$$

by introducing the following distributed input:

$$u(x,t) = U(t+D(x-1)), \quad x \in [0,1] \tag{4.15}$$

Thus, the input delay is now represented as a coupling with a transport PDE driven by an input with unknown convection speed $1/D$.

4.3 Delay-Adaptive Control Design

Due to the fact that the distributed input is unmeasured, we introduce an estimate of the distributed input:

$$\hat{u}(x,t) = U(t+\hat{D}(t)(x-1)), \quad x \in [0,1] \tag{4.16}$$

where $\hat{D}(t)$ is the current estimate of the input delay. In order to stabilize (4.12)-(4.14), we must first predict the state of the system (4.12)-(4.14) once the delayed input reaches the system. In order to achieve this, we introduce a distributed predictor estimate:

$$\hat{p}(x,t) = Z(t+\hat{D}(t)x) = Z(t) + \hat{D}(t)\int_0^x f(\hat{p}(y,t),\hat{u}(y,t),\hat{r}(y,t))dy \tag{4.17}$$

in which

$$\hat{r}(x,t) = X_R(t+\hat{D}(t)x) \tag{4.18}$$

is the distributed trajectory estimate. If the input delay was known, the control law $U(t) = \kappa(Z(t+D),X_R(t+D))$ could be used to stabilize the system, exactly compensating for the delay present in the system. Therefore, by the certainty equivalence principle, we choose the control law as:

$$U(t) = \kappa\left(Z(t+\hat{D}(t)),X_R(t+\hat{D}(t))\right) = \kappa\left(\hat{p}(1,t),\hat{r}(1,t)\right) \tag{4.19}$$

In order to derive an adaptation update law for the estimated delay, we define and utilize at time t a prediction of the current system state $X(t)$, starting from a recent previous state

$X(t-\beta)$ with $\beta > 0$, and assuming the correct value of the input delay is $\hat{D}(t)$:

$$X_P(x,t,\hat{D}) = [q_P(x,t,\hat{D})^T, \dot{q}_P(x,t,\hat{D})^T]^T = X(t-\beta) + \beta \int_0^x f_0(X_P, U_P)\, dy \tag{4.20}$$

where

$$U_P(x,t,\hat{D}) = U(t-\hat{D}+\beta(x-1)), \quad x \in [0,1] \tag{4.21}$$

Note that in this section, notations indicating nested functions of $(y,t,\hat{D})$ have been removed for the sake of brevity. An important property of this prediction is the following:

$$X_P(x,t,D) = X(t+\beta(x-1)) \tag{4.22}$$

$$X_P(1,t,D) = X(t) \tag{4.23}$$

and thus if the estimated value of the delay $\hat{D}(t)$ equals the true value of the delay D, then our prediction of the current system state $X_P(1,t,\hat{D})$ is equivalent to the current system state $X(t)$. Leveraging this useful property, we can utilize the mean square error between the predicted system state $X_P(1,t,\hat{D})$ and the current system state $X(t)$ through a gradient descent algorithm, updating the estimate of the delay $\hat{D}(t)$ in order to minimize this prediction error. For this purpose, we utilize the following instantaneous cost function, initially proposed in [75] for a linear plant:

$$J : (t,\hat{D}) \in \left[t_0, \infty\right[\rightarrow \frac{1}{2}\left|X_P(1,t,\hat{D}) - X(t)\right|^2 \tag{4.24}$$

In order to obtain the gradient of this cost function with respect to the estimated delay $\hat{D}$, it is first necessary to determine the partial derivative of X_P with respect to $\hat{D}$:

$$\frac{\partial X_P}{\partial \hat{D}}(x,t,\hat{D}) = \beta \int_0^x \left(\frac{\partial f_0}{\partial X_P}(X_P,U_P)\frac{\partial X_P}{\partial \hat{D}}(y,t,\hat{D}) + \frac{\partial f_0}{\partial U_P}(X_P,U_P)\frac{\partial U_P}{\partial \hat{D}}(y,t,\hat{D}) \right) dy$$

$$= \beta \int_0^x \left(\frac{\partial f_0}{\partial X_P}(X_P, U_P) \frac{\partial X_P}{\partial \hat{D}}(y,t,\hat{D}) - \begin{bmatrix} 0 \\ M(q_P)^{-1} \end{bmatrix} \dot{U}_P(y,t,\hat{D}) \right) dy \tag{4.25}$$

By taking the derivative of this equation with respect to x, it can be seen that $\frac{\partial^2 X_P}{\partial \hat{D} \partial x}$ satisfies the following ODE:

$$\frac{\partial^2 X_P}{\partial \hat{D} \partial x}(x,t,\hat{D}) = \beta \left(\frac{\partial f_0}{\partial X_P}(X_P, U_P) \frac{\partial X_P}{\partial \hat{D}}(x,t,\hat{D}) - \begin{bmatrix} 0 \\ M(q_P)^{-1} \end{bmatrix} \dot{U}_P(x,t,\hat{D}) \right) \tag{4.26}$$

$$\frac{\partial X_P}{\partial \hat{D}}(0,t,\hat{D}) = 0 \tag{4.27}$$

By defining the transition matrix Φ_0 associated to the corresponding homogeneous equation, one solves (4.26-4.27) to obtain:

$$\frac{\partial X_P}{\partial \hat{D}}(x,t,\hat{D}) = -\beta \int_0^x \Phi_0(x,y,t,\hat{D}) \begin{bmatrix} 0 \\ M(q_P)^{-1} \end{bmatrix} \dot{U}_P(y,t,\hat{D}) dy \tag{4.28}$$

where $\Phi_0(x,y,t,\hat{D})$ is the solution to the following homogeneous ODE:

$$\frac{\partial \Phi_0}{\partial x}(x,y,t,\hat{D}) = \beta \frac{\partial f_0}{\partial X_P}(X_P, U_P) \Phi_0(x,y,t,\hat{D}) \tag{4.29}$$

$$\Phi_0(y,y,t,\hat{D}) = I \quad y \in [0,1], x \in [y,1] \tag{4.30}$$

Taking the gradient of the instantaneous cost function (4.24), the following equation is obtained:

$$\frac{\partial J}{\partial \hat{D}}(t,\hat{D}) = \left(X_P(1,t,\hat{D}) - X(t) \right)^T \frac{\partial X_P}{\partial \hat{D}}(1,t,\hat{D}) \tag{4.31}$$

Utilizing this computed gradient, the rate of change of the delay estimate is designed as:

$$\dot{\hat{D}}(t) = \gamma \text{Proj}_{[\underline{D},\overline{D}]}\left\{\hat{D}(t), \rho_D(t)\right\} \tag{4.32}$$

where

$$\rho_D(t) = \frac{-\frac{\partial J}{\partial \hat{D}}(t,\hat{D}(t))}{1+\left|\frac{\partial X_P}{\partial \hat{D}}(1,t,\hat{D}(t))\right|^2} \tag{4.33}$$

and $\gamma > 0$ is the adaptation rate of the delay estimate. The projection operator is utilized in order to ensure that the delay estimate remains in the interval $[\underline{D},\overline{D}]$. Note that normalization by the regressor is employed in the adaptation of the delay estimate (4.33) in order to reduce the speed of adaptation when there are large changes in the input.

In order to analyze the properties of (4.33), it is first necessary to introduce the following technical lemma:

Lemma 4.1. *The transition matrix $\Phi(x,y,\bullet)$ associated with the space-varying homogeneous equation $\frac{\partial \phi}{\partial x}(x,\bullet) = A(x,\bullet)\phi(x,\bullet)$ with $y \in [0,1]$ and $x \in [y,1]$ satisfies the following property:*

$$\left|\Phi(x,y,\bullet)\right| \leq \exp\left\{\max_{x\in[0,1]}\left|A(x,\bullet)\right|\right\} \tag{4.34}$$

Furthermore, there exists a class $\mathscr{K}_\infty$ function α_1 such that for all $y \in [0,1]$, $x \in [y,1]$:

$$\left|\Phi(x,y,\bullet)\right| \leq 1 + \alpha_1(\left\|A(\bullet)\right\|_1) \tag{4.35}$$

Proof. By definition, for a given $y \in [0,1]$ and $t \geq 0$, Φ satisfies the following differential equation in x:

$$\frac{\partial \Phi}{\partial x}(x,y,\bullet) = A(x,\bullet)\Phi(x,y,\bullet) \quad x \in [y,1] \tag{4.36}$$

$$\Phi(y,y,\bullet) = I \tag{4.37}$$

Therefore, its norm satisfies for any $x \in [y,1]$:

$$\begin{aligned}\frac{\partial|\Phi|}{\partial x}(x,y,\bullet) &\leq \left|\frac{\partial \Phi}{\partial x}(x,y,\bullet)\right| \\ &\leq \left|A(x,\bullet)\right|\left|\Phi(x,y,\bullet)\right| \end{aligned} \tag{4.38}$$

with $\left|\Phi(y,y,\bullet)\right| = 1$. Therefore, there exists a $\mathscr{K}_\infty$ function α_1 such as introduced in this lemma. Furthermore, through taking the maximum of $\left|A(x,\bullet)\right|$ with respect to x and applying separation of variables, one obtains the upper bound (4.34) stated in this lemma. □

By utilizing a steepest descent argument [80] along with the application of Lemma 4.1 and an appropriate bounding of terms, one obtains the following properties of (4.33), provided that the initial delay estimate is close enough to the true value of the delay:

Lemma 4.2. *There exist positive parameters $H > 0$ and $\tilde{D}_{\max} > 0$ such that, if $\left|\tilde{D}(0)\right| < \tilde{D}_{\max}$, and $X(t)$, $U(t)$, and $\dot{U}(t)$ are uniformly bounded:*

$$\tilde{D}(t)\rho_D(t) \geq 0 \tag{4.39}$$

$$\left|\rho_D(t)\right| \leq H \tag{4.40}$$

where $\tilde{D}(t) = D - \hat{D}(t)$ is the current estimation error of the delay.

Proof. By utilizing a steepest descent argument [80], one obtains the property $\tilde{D}(t)\rho_D(t) \geq 0$. In order to obtain the property $\left|\rho_D(t)\right| \leq H$, it is necessary to establish the uniform boundedness of X_P and $\frac{\partial X_P}{\partial \hat{D}}$ in (4.33). Due to the forward completeness property of f_0, the uniform boundedness of X and U, and the fixed integration distance in (4.20), one obtains the uniform boundedness of X_P. To establish a bound for $\frac{\partial X_P}{\partial \hat{D}}$, Lemma 4.1 is used to prove the uniformn boundedness of Φ_0. Due to the uniform boundedness of Φ_0, X_P, U, and $\dot{U}$, along with the fact that f_0 is a

class C^2 function, one obtains the uniform boudedness of $\frac{\partial X_P}{\partial \hat{D}}$ from (4.28). Applying the uniform boudedness of X_P and $\frac{\partial X_P}{\partial \hat{D}}$ to (4.33), one obtains the property $\left|\rho_D(t)\right| \leq H$. □

In order to predict the input delay of the robot manipulator, we employ a gradient-based method minimizing the difference between the current system state, and a prediction of what the system state should be if the currently estimated delay is equal to the true value of the delay. It is important to note that contrary to the implementation of the current state predictor X_P in previous works by Bresch-Pietri *et al.* [75, 78] , in which the current state is estimated by simulating from the initial condition $X(t_0)$, the predictor method present in this chapter estimates the current state by simulating from the more recent state $X(t-\beta)$. This modification to the predictor-based update law ensures that the computational cost of performing this method remains consistent due to the fixed bounds of the integral in (4.20), as well as serving to bound the maximum growth of the state transition matrix $\Phi_0(x,y,t,\hat{D})$ and prediction error $X_P(1,t,\hat{D}) - X(t)$. In essence, this cost function is a simulated replay (4.20) of the last β seconds of the robot manipulator motion, and the update of the delay estimate aims to match this simulated replay to the observed value of the system state during this period. If the estimate of the delay is correct, the simulated replay should match perfectly with the motion of the robot manipulator during the most recent β seconds of motion.

Due to the fact that the properties stated in Lemma 4.2 hold for any $\beta > 0$, the selection of the replay length β is motivated primarily by practical considerations. If β is chosen to be too small, measurement noise and external disturbances will make up a large portion of the difference between the predicted and actual current system state, meaning that the update of the estimated delay will be susceptible to high-frequency noise. If β is too large, the accuracy of the simulated replay will suffer due to accumulated inaccuracies, such as small inconsistencies in the robot manipulator model, as well as accumulated error from the numerical methods used in the simulation. Additionally, a larger β increases the computational cost of the method, as a longer simulation will need to be performed. Thus, the selection of β is a balance between

susceptibility to high-frequency noise with a small β, and susceptibility to low-frequency noise and increased computational burden with a large β. Through simulations and experiments, we have found that selecting β to be roughly an order of magnitude smaller than the settling time of the control law $\kappa(Z, X_R)$ is a suitable choice.

Utilizing the delay estimate update law (4.32), as well as the control law (4.19), we are now ready to present the stability theorem for the delay-adaptive controller operating on a robot manipulator with unknown input delay.

Theorem 4.1. *Consider the closed-loop system consisting of the error dynamics of the robot manipulator (4.4), control law (4.19), delay estimate update law (4.32), and desired joint trajectories $q_R(t)$ satisfying Assumptions 4.1-4.4. Define the following functionals:*

$$\Gamma(t) = \left|Z(t)\right| + \left\|\tilde{u}(t)\right\|_1 + \int_0^1 \left|\hat{u}(x,t) - \hat{u}_R(x,t)\right|_1 dx + \int_0^1 \left|\hat{u}_x(x,t) - \hat{u}_{R,x}(x,t)\right|_1 dx \tag{4.41}$$

$$\Gamma_R(t) = \left|\hat{r}_x(1,t)\right|_1 + \left\|\hat{r}_x(t)\right\|_1 + \left\|\hat{r}_{xx}(t)\right\|_1 \tag{4.42}$$

in which $\hat{r}$ is defined in (4.18),

$$\hat{u}_R(x,t) = \kappa(\hat{p}(x,t), \hat{r}(x,t)) \tag{4.43}$$

is the predicted distributed input reference, and

$$\tilde{u}(x,t) = u(x,t) - \hat{u}(x,t) \tag{4.44}$$

is the distributed input estimation error. Then, there exist R^*, Γ^*, δ^*, γ^*, c_3, $c_4 > 0$, and a class $\mathcal{K}_\infty$ function α^* such that, if $X_R(t)$, $\dot{X}_R(t)$, and $\ddot{X}_R(t)$ are uniformly bounded by R^*, $\Gamma(0) \leq \Gamma^*$, $\left|\tilde{D}(0)\right| < \delta^*$, and $\gamma < \gamma^*$ then

$$\Gamma(t) \leq c_3 \Gamma(0) e^{-c_4 t} + \alpha^* \left(\sup_{s \in [0,t]} \{\Gamma_R(s)\} \right), \quad \forall t \geq 0 \tag{4.45}$$

Furthermore, if $\dot{X}_R(t) \to 0$ *as* $t \to \infty$*, then*

$$Z(t) \underset{t\to\infty}{\to} 0 \tag{4.46}$$

4.4 Lyapunov Analysis

In this section, the proof of Theorem 4.1 resembles the proof of Theorem 3 in [78], since the delay-adaptive control approach in this chapter is an extension of their approach. However, significant changes to this proof were necessary in order to adapt it to a trajectory tracking task utilizing a robot manipulator. In the original backstepping transformation utilized by Bresch-Pietri and Krstic in [78], the nonlinear plant f and the corresponding control law κ are assumed to be autonomous. As both f and κ are non-autonomous when tracking a time-varying reference trajectory $X_R(t)$, it was necessary to reformulate the backstepping transformation to handle the non-autonomous case. This reformulation of the backstepping transformation introduces new terms in the Lyapunov analysis, which need to be carefully bounded. In order to bound these terms, it is necessary to do so in terms of the current tracking errors, the distributed input torques, as well as the temporal derivatives of the reference trajectory. As a result of terms being bound by the temporal derivatives of the reference trajectory, local input-to-state stability is established with respect to these derivatives, with regulation of the tracking errors when the reference trajectory approaches a stationary configuration. Thus, the analysis presented here extends upon the stability results of [78] when there is a time-varying reference to track, while preserving the original local asymptotic stability result of [78] when the reference is stationary. Additionally, due to the control affine nature of robot manipulators, the application of the mean value theorem is not necessary in order to bound terms such as $f_{\tilde{u}}$ appearing in this section. This key property of robot manipulators allows for a large reduction in the number of terms that are necessary for the Lyapunov analysis presented in this section when compared to that of [78].

It is important to note that both the original method presented in [78] and the method

we present here utilize the $\mathscr{L}_1$ norm rather than the $\mathscr{L}_2$ norm as is typical when performing Lyapunov analysis. For an additional reference in the use of $\mathscr{L}_1$ norms in Lyapunov analysis, the interested reader can refer to [81].

In order to perform the Lyapunov stability analysis, a backstepping transformation is first used in order to reformulate (4.12)-(4.14).

Lemma 4.3. *The backstepping transformation of the distributed input estimates (4.16)*

$$\hat{w}(x,t) = \hat{u}(x,t) - \kappa(\hat{p}(x,t), \hat{r}(x,t)) \tag{4.47}$$

in which the distributed predictor estimate is defined in (4.17), together with the control law (4.19), transforms plant (4.12)-(4.14) into:

$$\dot{Z}(t) = f(Z(t), \kappa(Z(t), \hat{r}(0,t)) + \hat{w}(0,t) + \tilde{u}(0,t), \hat{r}(0,t)) \tag{4.48}$$

$$D\tilde{u}_t(x,t) = \tilde{u}_x(x,t) - \tilde{D}(t)g_1(x,t) - \dot{\hat{D}}(t)g_2(x,t) \tag{4.49}$$

$$\tilde{u}(1,t) = 0 \tag{4.50}$$

$$\hat{D}(t)\hat{w}_t(x,t) = \hat{w}_x(x,t) + \dot{\hat{D}}(t)h_1(x,t) - h_2(x,t)f_{\tilde{u}}(t) \tag{4.51}$$

$$\hat{w}(1,t) = 0 \tag{4.52}$$

Additionally, the spatial derivative of the backstepping transformation (4.47) satisfies the following PDE:

$$\hat{D}(t)\hat{w}_{xt}(x,t) = \hat{w}_{xx}(x,t) + \dot{\hat{D}}(t)h_3(x,t) - h_4(x,t)f_{\tilde{u}}(t) \tag{4.53}$$

$$\hat{w}_x(1,t) = -\dot{\hat{D}}(t)h_1(1,t) + h_2(1,t)f_{\tilde{u}}(t) \tag{4.54}$$

The expressions for the terms g_1, g_2, h_1, h_2, h_3, h_4, and $f_{\tilde{u}}$ are as follows, noting that notations

indicating distributed functions of (x, t) and (y, t) have been removed for the sake of brevity:

$$g_1 = \frac{1}{\hat{D}(t)}\hat{u}_x \tag{4.55}$$

$$g_2 = \frac{D}{\hat{D}(t)}(x-1)\hat{u}_x \tag{4.56}$$

$$h_1 = (x-1)\hat{u}_x - \hat{D}(t)\frac{\partial \kappa}{\partial \hat{p}}(\hat{p},\hat{r})\int_0^x \Phi(x,y,t)\psi dy - x\frac{\partial \kappa}{\partial \hat{r}}(\hat{p},\hat{r})\hat{r}_x \tag{4.57}$$

$$h_2 = \hat{D}(t)\frac{\partial \kappa}{\partial \hat{p}}(\hat{p},\hat{r})\Phi(x,0,t) \tag{4.58}$$

$$h_3 = \hat{u}_x + (x-1)\hat{u}_{xx} - \frac{d}{dx}\left[\hat{D}(t)\frac{\partial \kappa}{\partial \hat{p}}(\hat{p},\hat{r})\int_0^x \Phi(x,y,t)\psi dy - x\frac{\partial \kappa}{\partial \hat{r}}(\hat{p},\hat{r})\hat{r}_x\right] \tag{4.59}$$

$$h_4 = \hat{D}(t)\frac{d}{dx}\left[\frac{\partial \kappa}{\partial \hat{p}}(\hat{p},\hat{r})\right]\Phi(x,0,t) + \hat{D}(t)^2\frac{\partial \kappa}{\partial \hat{p}}(\hat{p},\hat{r})\frac{\partial f}{\partial \hat{p}}(\hat{p},\hat{u},\hat{r})\Phi(x,0,t) \tag{4.60}$$

$$\begin{aligned} f_{\tilde{u}}(t) &= f(\hat{p}(0,t),u(0,t),\hat{r}(0,t)) - f(\hat{p}(0,t),\hat{u}(0,t),\hat{r}(0,t)) \\ &= \begin{bmatrix} 0 \\ M(q(t))^{-1} \end{bmatrix}\tilde{u}(0,t) \end{aligned} \tag{4.61}$$

where

$$\psi = f(\hat{p},\hat{u},\hat{r}) + (x-1)\frac{\partial f}{\partial \hat{u}}(\hat{p},\hat{u},\hat{r})\hat{u}_x + x\frac{\partial f}{\partial \hat{r}}(\hat{p},\hat{u},\hat{r})\hat{r}_x \tag{4.62}$$

$$\hat{u} = \hat{w} + \kappa(\hat{p},\hat{r}) \tag{4.63}$$

$$\hat{u}_x = \hat{w}_x + \frac{d}{dx}\left[\kappa(\hat{p},\hat{r})\right] \tag{4.64}$$

$$\hat{u}_{xx} = \hat{w}_{xx} + \frac{d^2}{dx^2}\left[\kappa(\hat{p},\hat{r})\right] \tag{4.65}$$

and $\Phi(x,y,t)$ is the solution to the following homogeneous ODE:

$$\frac{\partial \Phi}{\partial x}(x,y,t) = \hat{D}(t)\frac{\partial f}{\partial \hat{p}}(\hat{p},\hat{u},\hat{r})\Phi(x,y,t) \tag{4.66}$$

$$\Phi(y,y,t) = I \tag{4.67}$$

Proof. Note that in this proof, notations indicating distributed functions of (x, t) and (y, t) have been removed for the sake of brevity.

First, (4.48) is obtained from applying equations (4.47) and (4.44) to that of the nonlinear plant (4.12)-(4.14). Next, by examining the spatial and temporal derivatives of the distributed input estimate (4.16) and distributed trajectory estimate (4.18), the following relationships are obtained:

$$\hat{D}(t)\hat{u}_t = \hat{u}_x + \dot{\hat{D}}(t)(x-1)\hat{u}_x \tag{4.68}$$

$$\hat{u}(1,t) = U(t) \tag{4.69}$$

$$\hat{D}(t)\hat{r}_t = \hat{r}_x + \dot{\hat{D}}(t)x\hat{r}_x \tag{4.70}$$

$$\hat{r}(0,t) = X_R(t) \tag{4.71}$$

By combining (4.68) with that of the relationship between the spatial and temporal derivatives of the input (4.13)-(4.14), and applying the backstepping transformation (4.47), (4.49)-(4.50) are obtained.

In order to obtain the governing equation for $\hat{w}$, we first substitute (4.47) and (4.70) into (4.68) to obtain the following expression:

$$\hat{D}(t)\hat{w}_t = \hat{w}_x - \frac{\partial \kappa}{\partial \hat{p}}(\hat{p},\hat{r})(\hat{D}(t)\hat{p}_t - \hat{p}_x) + \dot{\hat{D}}(t)(x-1)\hat{u}_x - \dot{\hat{D}}(t)x\frac{\partial \kappa}{\partial \hat{r}}(\hat{p},\hat{r})\hat{r}_x \tag{4.72}$$

In order to obtain (4.51)-(4.52) from (4.72), it is necessary to study the behavior of the temporal and spatial derivatives of the distributed predictor. By taking the temporal and spatial derivatives of $\hat{p}$ and substituting in (4.68) and (4.70), the following relationships are obtained:

$$\hat{p}_t = f(\hat{p}(0,t),u(0,t),\hat{r}(0,t)) + \int_0^x \left(\frac{\partial f}{\partial \hat{p}}(\hat{p},\hat{u},\hat{r})\hat{D}(t)\hat{p}_t + \frac{\partial f}{\partial \hat{u}}(\hat{p},\hat{u},\hat{r}) \left[\hat{u}_y + \dot{\hat{D}}(t)(y-1)\hat{u}_y \right] \right.$$

$$+\frac{\partial f}{\partial \hat{r}}\left[\hat{r}_y+\dot{\hat{D}}(t)y\hat{r}_y\right]\Bigg)dy+\dot{\hat{D}}(t)\int_0^x f(\hat{p},\hat{u},\hat{r})dy \tag{4.73}$$

$$\hat{p}_x=\hat{D}(t)f(\hat{p},\hat{u},\hat{r})=\hat{D}(t)f(\hat{p}(0,t),\hat{u}(0,t),\hat{r}(0,t))$$
$$+\hat{D}(t)\int_0^x\left(\frac{\partial f}{\partial \hat{p}}(\hat{p},\hat{u},\hat{r})\hat{p}_y+\frac{\partial f}{\partial \hat{u}}(\hat{p},\hat{u},\hat{r})\hat{u}_y+\frac{\partial f}{\partial \hat{r}}(\hat{p},\hat{u},\hat{r})\hat{r}_y\right)dy \tag{4.74}$$

Then, we define $\phi=\hat{D}(t)\hat{p}_t-\hat{p}_x$. It is observed from substitution of (4.73) and (4.74) into this definition that ϕ satisfies the following equation in x, parametrized in t:

$$\frac{\partial \phi}{\partial x}=\hat{D}(t)\frac{\partial f}{\partial \hat{p}}(\hat{p},\hat{u},\hat{r})\phi+\dot{\hat{D}}(t)\hat{D}(t)\psi \tag{4.75}$$

$$\phi(0,t)=\hat{D}(t)f_{\tilde{u}}(t) \tag{4.76}$$

where $f_{\tilde{u}}(t)$ and ψ are defined in (4.61) and (4.62) respectively.

By defining the transition matrix Φ associated to the corresponding homogeneous equation, one solves this equation to obtain:

$$\hat{D}(t)\hat{p}_t-\hat{p}_x=\dot{\hat{D}}(t)\hat{D}(t)\int_0^x\Phi(x,y,t)\psi dy+\Phi(x,0,t)\hat{D}(t)f_{\tilde{u}}(t) \tag{4.77}$$

Substituting this equation along with the backstepping transformation (4.47) into (4.72) yields (4.51)-(4.52).

By taking the spatial derivative of (4.51), one obtains the governing equation (4.53). Additionally, the boundary condition (4.54) is directly obtained by rearranging the governing equation (4.51), along with the knowledge that $\hat{w}_t(1,t)=0$ due to taking the time derivative of the boundary condition (4.52). □

For the purpose of the Lyapunov analysis, we consider the following Lyapunov-

Krasovskii functional candidate:

$$W(t) = V_0(Z(t)) + b_0 D \int_0^1 (1+x)\left|\tilde{u}(x,t)\right|_1 dx + b_1 \hat{D}(t) \int_0^1 (1+x)\left|\hat{w}(x,t)\right|_1 dx$$
$$+ b_2 \hat{D}(t) \int_0^1 (1+x)\left|\hat{w}_x(x,t)\right|_1 dx \tag{4.78}$$

in which $V_0 = \sqrt{V}$, which was previously defined in Assumption 4.3. This functional measures the current tracking errors of the robot manipulator, as well as the difference between the distributed input and the desired distributed input. Utilizing the properties of Assumption 4.3, along with the state space model of the robot manipulator (4.4), the following inequality is obtained:

$$\dot{V}_0(t) \leq -\frac{\lambda}{2}\left|Z(t)\right| + \frac{c_2}{2}\left|M(q)^{-1}\right|\left|\tilde{u}(0,t) + \hat{w}(0,t)\right|_1 \leq -\frac{\lambda}{2}\left|Z(t)\right| + M_1\left|\tilde{u}(0,t) + \hat{w}(0,t)\right|_1 \tag{4.79}$$

where

$$M_1 = \max_{q\in[0,2\pi)} \frac{c_2}{2}\left|M(q)^{-1}\right| \tag{4.80}$$

By utilizing (4.53), the temporal derivative of the fourth term of $W(t)$ can be bounded as follows:

$$\begin{aligned}
&\frac{d}{dt}\left[b_2 \hat{D}(t) \int_0^1 (1+x)\left|\hat{w}_x(x,t)\right|_1 dx\right] \\
&= b_2 \hat{D}(t) \int_0^1 (1+x)\hat{w}_{xt}(x,t)\cdot \operatorname{sign}(\hat{w}_x(x,t))dx + b_2 \dot{\hat{D}}(t) \int_0^1 (1+x)\left|\hat{w}_x(x,t)\right|_1 dx \\
&\leq b_2 \int_0^1 (1+x)\hat{w}_{xx}(x,t)\cdot \operatorname{sign}(\hat{w}_x(x,t))dx + b_2\left|\dot{\hat{D}}(t)\right| \int_0^1 (1+x)\left|h_3(x,t)\right|_1 dx \\
&\quad + b_2 \int_0^1 (1+x)\left|h_4(x,t) f_{\tilde{u}}(t)\right|_1 dx + b_2\left|\dot{\hat{D}}(t)\right| \int_0^1 (x+1)\left|\hat{w}_x(x,t)\right|_1 dx
\end{aligned} \tag{4.81}$$

Using integration by parts, the first term in this inequality can be simplified as follows:

$$b_2\int_0^1(1+x)\hat{w}_{xx}(x,t)\cdot\operatorname{sign}(\hat{w}_x(x,t))dx = (1+x)\left|\hat{w}_x(x,t)\right|_1\Big|_0^1 - \int_0^1\left|\hat{w}_x(x,t)\right|_1 dx$$
$$= 2b_2\left|\hat{w}_x(1,t)\right|_1 - b_2\left|\hat{w}_x(0,t)\right|_1 - b_2\left\|\hat{w}_x(t)\right\|_1 \quad (4.82)$$

By utilizing the same method for the terms containing $\hat{w}$ and $\tilde{u}$ within (4.78), the inequality (4.79), and applying the boundary conditions (4.50) and (4.52), the following inequality is obtained:

$$\begin{aligned}\dot{W}(t) \leq & -\frac{\lambda}{2}\left|Z(t)\right| + M_1\left|\tilde{u}(0,t)+\hat{w}(0,t)\right|_1 - b_0\left\|\tilde{u}(t)\right\|_1 - b_0\left|\tilde{u}(0,t)\right|_1 - b_1\left\|\hat{w}(t)\right\|_1 \\ & -b_1\left|\hat{w}(0,t)\right|_1 - b_2\left\|\hat{w}_x(t)\right\|_1 + 2b_2\left|\hat{w}_x(1,t)\right|_1 - b_2\left|\hat{w}_x(0,t)\right|_1 \\ & +b_0\left|\tilde{D}(t)\right|\int_0^1(x+1)\left|g_1(x,t)\right|_1 dx + b_0\left|\dot{\hat{D}}(t)\right|\int_0^1(x+1)\left|g_2(x,t)\right|_1 dx \\ & +b_1\left|\dot{\hat{D}}(t)\right|\int_0^1(x+1)\left|h_1(x,t)\right|_1 dx + b_1\int_0^1(x+1)\left|h_2(x,t)f_{\tilde{u}}(t)\right|_1 dx \\ & +b_2\left|\dot{\hat{D}}(t)\right|\int_0^1(1+x)\left|h_3(x,t)\right|_1 dx + b_2\int_0^1(x+1)\left|h_4(x,t)f_{\tilde{u}}(t)\right|_1 dx \\ & +b_1\left|\dot{\hat{D}}(t)\right|\int_0^1(x+1)\left|\hat{w}(x,t)\right|_1 dx + b_2\left|\dot{\hat{D}}(t)\right|\int_0^1(x+1)\left|\hat{w}_x(x,t)\right|_1 dx \end{aligned} \quad (4.83)$$

In order to bound the positive terms in the previous expression, we define the following combined functional:

$$S(t) = \frac{1}{2}\left(W(t)+\Gamma_R(t)\right) \quad (4.84)$$

where $\Gamma_R(t)$ is defined in (4.42). This functional measures the current tracking errors of the robot manipulator, the difference between the distributed input and the desired distributed input, and the magnitude of temporal derivatives of the desired trajectory.

Through the application of Assumption 4.3, it is found that $\left|Z(t)\right|$ satisfies the following

inequality:

$$\frac{\lambda}{2}\left|Z(t)\right| \geq \frac{\lambda}{2\sqrt{c_1}} V_0(Z(t)) \tag{4.85}$$

Furthermore, the term $b_2\left\|\hat{w}_x(t)\right\|_1$ satisfies the following inequality:

$$b_2\left\|w_x(t)\right\|_1 \geq \frac{1}{2} b_2 \int_0^1 (1+x)\left|\hat{w}_x(x,t)\right|_1 dx \geq \frac{1}{2\overline{D}} b_2 \hat{D}(t) \int_0^1 (1+x)\left|\hat{w}_x(x,t)\right|_1 dx \tag{4.86}$$

Similar inequalities can be formulated for the terms $b_0\left\|\tilde{u}(t)\right\|_1$ and $b_1\left\|\hat{w}(t)\right\|_1$ appearing in (4.83). In order to bound further terms present in (4.83), it is first necessary to introduce several technical lemmas:

Lemma 4.4. *There exists a class $\mathscr{K}_\infty$ function α_2 such that for all $x \in [0,1]$:*

$$\left|\hat{p}(x,t)\right| \leq \alpha_2\left(|Z| + \left\|\hat{w}(t)\right\|_1\right) \tag{4.87}$$

Proof. The distributed predictor satisfies the following spatial ODE:

$$\hat{p}_x(x,t) = \hat{D}(t) f(\hat{p}(x,t), \hat{w}(x,t) + \kappa(\hat{p}(x,t), \hat{r}(x,t)), \hat{r}(x,t)) \tag{4.88}$$

$$\hat{p}(0,t) = Z(t) \tag{4.89}$$

Therefore, as $\hat{D}(t) \in [\underline{D}, \overline{D}]$, f is continuous, the plant (4.12)-(4.14) is forward complete, $f(0, \kappa(0,\hat{r}), \hat{r}) = 0$, and $\hat{r}$ is uniformly bounded, there exists a $\mathscr{K}_\infty$ function α_2 such as introduced in this lemma. □

Lemma 4.5. *There exist class $\mathscr{K}_\infty$ functions α_3 and α_4 such that for all $x \in [0,1]$:*

$$\left|\hat{p}_x(x,t)\right| \leq \alpha_3\left(\left|\hat{p}(x,t)\right|_1 + \left|\hat{w}(x,t)\right|_1\right) \tag{4.90}$$

$$\left|\hat{p}_{xx}(x,t)\right| \leq \alpha_4\left(\left|\hat{p}(x,t)\right|_1 + \left|\hat{w}(x,t)\right|_1 + \left|\hat{w}_x(x,t)\right|_1 + \left|\hat{r}_x(x,t)\right|_1\right) \tag{4.91}$$

Proof. Note that in this proof, notations indicating distributed functions of (x, t) have been removed for the sake of brevity.

The quantity $\hat{p}_x$ is equivalent to the following expression:

$$\hat{p}_x = \hat{D}(t) f(\hat{p},\hat{u},\hat{r}) = \hat{D}(t) f(\hat{p},\hat{w}+\kappa(\hat{p},\hat{r}),\hat{r}) \tag{4.92}$$

Therefore, as $\hat{D}(t) \in [\underline{D},\overline{D}]$, $f(0,\kappa(0,\hat{r}),\hat{r}) = 0$, and $\hat{r}$ is uniformly bounded, there exists a $\mathscr{K}_\infty$ function α_3 such as introduced in this lemma.

By taking the spatial derivative of $\hat{p}_x$, the following expression for $\hat{p}_{xx}$ is obtained:

$$\hat{p}_{xx} = \hat{D}(t)\left[\frac{\partial f}{\partial \hat{p}}(\hat{p},\hat{u},\hat{r})\hat{p}_x + \frac{\partial f}{\partial \hat{u}}(\hat{p},\hat{u},\hat{r})\hat{u}_x + \frac{\partial f}{\partial \hat{r}}(\hat{p},\hat{u},\hat{r})\hat{r}_x\right] \tag{4.93}$$

For a given $\hat{D}$ and $\hat{r}$, this expression can be bounded by the terms $\hat{p}$, $\hat{u}$, $\hat{p}_x$, $\hat{u}_x$, and $\hat{r}_x$. $\hat{u}$ can be bounded by $\hat{p}$ and $\hat{w}$, and $\hat{p}_x$ can be bounded by α_2. Through the investigation of (4.64), it can be seen that $\hat{u}_x$ can be bounded by $\hat{w}_x$, $\hat{p}$, $\hat{p}_x$, and $\hat{r}_x$, with $\hat{u}_x = 0$ if $\hat{w}_x = \hat{p}_x = \hat{r}_x = 0$. Furthermore, $p_{xx} = 0$ if $\hat{u}_x = \hat{p}_x = \hat{r}_x = 0$. Therefore, as $\hat{D}(t) \in [\underline{D},\overline{D}]$ and $\hat{r}$ is uniformly bounded, there exists a $\mathscr{K}_\infty$ function α_4 such as introduced in this lemma. □

Lemma 4.6. *There exist class $\mathscr{K}_\infty$ functions α_i, with $i = 5,\ldots,13$ and constants $M_2, M_3, M_4 > 0$ such that:*

$$\int_0^1 (x+1)\left|g_1(x,t)\right|_1 dx \leq \alpha_5(|Z| + \left\|\hat{w}(t)\right\|_1 + \left\|\hat{w}_x(t)\right\|_1 + \left\|\hat{r}_x(t)\right\|_1) \leq \alpha_6(S(t)) \tag{4.94}$$

$$\int_0^1 (x+1)\left|g_2(x,t)\right|_1 dx \leq \alpha_7(S(t)) \tag{4.95}$$

$$\int_0^1 (x+1)\left|h_1(x,t)\right|_1 dx \leq \alpha_8(S(t)) \tag{4.96}$$

$$\int_0^1 (1+x)\left|h_3(x,t)\right|_1 dx \leq \left|\hat{w}_x(0,t)\right|_1 + \alpha_9(S(t)) \tag{4.97}$$

$$\int_0^1 (x+1)\left|h_2(x,t)f_{\tilde{u}}(t)\right|_1 dx \leq \left(M_2 + \alpha_{10}(S(t))\right)\left|\tilde{u}(0,t)\right|_1 \tag{4.98}$$

$$\int_0^1 (x+1)\left|h_4(x,t)f_{\tilde{u}}(t)\right|_1 \leq \left(M_3 + \alpha_{11}(S(t))\right)\left|\tilde{u}(0,t)\right|_1 \tag{4.99}$$

$$\left|\hat{w}_x(1,t)\right|_1 \leq \left|\dot{\hat{D}}(t)\right|\alpha_{12}(S(t)) + \left(M_4 + \alpha_{13}(S(t))\right)\left|\tilde{u}(0,t)\right|_1 \tag{4.100}$$

Proof. By the application of Lemma 4.4, Lemma 4.5, and Assumption 4.1, along with the fact that f and κ are class C^2 functions, one obtains the existence of α_5 as stated in this lemma. Then, through the application of the following inequality:

$$\begin{aligned}&\left(\left|Z(t)\right| + \left\|\tilde{u}(t)\right\|_1 + \left\|\hat{w}(t)\right\|_1 + \left\|\hat{w}_x(t)\right\|_1 + \left|\hat{r}_x(1,t)\right|_1 + \left\|\hat{r}_x(t)\right\|_1 + \left\|\hat{r}_{xx}(t)\right\|_1\right) \\ &\leq 2\max\left\{1, \frac{1}{b_0\underline{D}}, \frac{1}{b_1\underline{D}}, \frac{1}{b_2\underline{D}}\right\} S(t)\end{aligned} \tag{4.101}$$

the existence of α_6 as stated in this lemma is obtained. Applying these same considerations, the existence of α_7 as stated in this lemma is also obtained.

In order to bound the higher order spatial derivative term in h_3, integration by parts may be used to bound this term as a function of w_x:

$$\int_0^1 (x^2-1)\left|\hat{w}_{xx}(x,t)\right|_1 dx \leq \left|\hat{w}_x(0,t)\right|_1 + \int_0^1 2x\left|\hat{w}_x(x,t)\right|_1 dx \tag{4.102}$$

Applying this bound, along with the previously stated considerations and Lemma 4.1, the existence of α_8 and α_9 as stated in this lemma are obtained.

To bound further terms present in this lemma, additional considerations must be made. First, due to the control affine property of robot manipulators, the term $f_{\tilde{u}}$ can be bounded as

follows:

$$f_{\tilde{u}}(t) \leq \max_{q\in[0,2\pi)} \left| M(q)^{-1} \right| \left| \tilde{u}(0,t) \right|_1 \tag{4.103}$$

Second, utilizing previous considerations, the terms h_2 and h_4 can be bounded in terms of S as follows:

$$\left| h_2(x,t) \right|_1 \leq C_{h_2} + \alpha_{h_2}(S(t)) \tag{4.104}$$

$$\left| h_4(x,t) \right|_1 \leq C_{h_4} + \alpha_{h_4}(S(t)) \tag{4.105}$$

where $C_{h_2}, C_{h_4} \geq 0$ and $\alpha_{h_2}, \alpha_{h_4}$ are class $\mathscr{K}_\infty$ functions. Note that constants C_{h_2} and C_{h_4} are necessary since we do not have $h_2 = h_4 = 0$ when $S = 0$. Applying all previously stated considerations, the existence of class $\mathscr{K}_\infty$ functions α_{10}, α_{11}, α_{12}, α_{13} and constants M_2, M_3, and M_4 as stated in this lemma are obtained. □

Thus, by applying Lemma 4.6 located in the Appendix, and introducing $\eta = (1/2)\min\{\lambda/\sqrt{c_1}, 1/\overline{D}\}$, one bounds (4.83) in terms of class $\mathscr{K}_\infty$ functions $\alpha_i(S(t))$ with $i = 5, ..., 13$ of (4.84) and constants $M_2, M_3, M_4 > 0$:

$$\begin{aligned}
\dot{W}(t) \leq -\Bigg(& \eta W(t) - \left| \dot{\hat{D}}(t) \right| \Bigg(b_0\alpha_7(S(t)) + b_1\alpha_8(S(t)) + b_2\alpha_9(S(t)) + 2b_2\alpha_{12}(S(t)) \\
& + \frac{2}{\underline{D}}(b_1 + b_2)S(t) \Bigg) - \left| \tilde{D}(t) \right| b_0\alpha_6(S(t)) \Bigg) - \left| \tilde{u}(0,t) \right|_1 \Bigg(b_0 - b_1\alpha_{10}(S(t)) \\
& - b_2\alpha_{11}(S(t)) - 2b_2\alpha_{13}(S(t)) - M_1 - b_1M_2 - b_2(M_3 + 2M_4) \Bigg) \\
& - \left| \hat{w}(0,t) \right|_1 \Bigg(b_1 - M_1 \Bigg) - \left| \hat{w}_x(0,t) \right|_1 b_2 \Bigg(1 - \left| \dot{\hat{D}}(t) \right| \Bigg)
\end{aligned} \tag{4.106}$$

In order to ensure that the terms corresponding to $\left| \tilde{u}(0,t) \right|_1$ and $\left| \hat{w}(0,t) \right|_1$ in (4.106) are

negative $\forall t \geq 0$, we define the following constant parameters:

$$W^* > 0, \quad \Gamma_R^* > 0, \quad S^* = \frac{W^* + \Gamma_R^*}{2} \tag{4.107}$$

Furthermore, we choose the constant parameters of the functional $W(t)$ such that:

$$b_0 > b_1\alpha_{10}(S^*) + b_2\alpha_{11}(S^*) + 2b_2\alpha_{13}(S^*) + M_1 + b_1M_2 + b_2(M_3 + 2M_4) \tag{4.108}$$

$$b_1 > M_1, \tag{4.109}$$

To further reduce (4.106), we apply Lemma 4.2 and introduce the following functions:

$$\alpha_1^*(S(t)) = b_0\alpha_7(S(t)) + b_1\alpha_8(S(t)) + b_2\alpha_9(S(t)) + 2b_2\alpha_{12}(S(t)) + \frac{2}{\underline{D}}(b_1 + b_2)S(t) \tag{4.110}$$

$$\alpha_2^*(S(t)) = b_0\alpha_6(S(t)) \tag{4.111}$$

For $W(t) \leq W^*$ and $\Gamma_R(t) \leq \Gamma_R^*$, (4.106) reduces to:

$$\begin{aligned}\dot{W}(t) \leq &-\Big(\eta W(t) - \gamma H\alpha_1^*(W(t)) - \gamma H\alpha_1^*(\Gamma_R(t)) - \big|\tilde{D}(t)\big|\alpha_2^*(W(t)) - \big|\tilde{D}(t)\big|\alpha_2^*(\Gamma_R(t))\Big) \\ &-b_2(1-\gamma H)\big|\hat{w}_x(0,t)\big|_1\end{aligned} \tag{4.112}$$

Noting that $\Gamma_R(t)$ is uniformly bounded due to Assumption 4.1, by choosing for a given $\nu \in (0,1)$:

$$\gamma < \frac{1}{H}\min\left\{1, \frac{(1-\nu)\eta}{2\max\limits_{x\in[0,W^*]}\alpha_1^{*\prime}(x)}, \frac{(1-\nu)\nu\eta W^*}{2\alpha_1^*(\Gamma_R^*)}\right\} \tag{4.113}$$

$$\big|\tilde{D}(0)\big| < \min\left\{\tilde{D}_{\max}, \frac{(1-\nu)\eta}{2\max\limits_{x\in[0,W^*]}\alpha_2^{*\prime}(x)}, \frac{(1-\nu)\nu\eta W^*}{2\alpha_2^*(\Gamma_R^*)}\right\} \tag{4.114}$$

$$W(0) \leq W^* \tag{4.115}$$

$$\Gamma_R(t) \leq \Gamma_R^*, \quad \forall t \geq 0 \tag{4.116}$$

one ensures that:

$$\dot{W}(t) \leq -\nu^2 \eta W(t), \qquad \text{for} \quad W(t) \geq \alpha_3^*(\Gamma_R(t)) \tag{4.117}$$

$$W(t) \leq W^*, \qquad \forall t \geq 0 \tag{4.118}$$

where

$$\alpha_3^*(\Gamma_R(t)) = \frac{\gamma H \alpha_1^*(\Gamma_R(t)) + \left|\tilde{D}(t)\right| \alpha_2^*(\Gamma_R(t))}{(1-\nu)\nu\eta} \tag{4.119}$$

Through careful examination of (4.117), the following inequality is obtained:

$$W(t) \leq W(0) e^{-\nu^2 \eta t} + \alpha_3^* \left(\sup_{s \in [0,t]} \{\Gamma_R(s)\} \right) \tag{4.120}$$

In order to provide a stability result in terms of Γ and Γ_R, Assumption 4.3 is used to prove the existence of two constants c_1^*, $c_2^* > 0$ such that $c_1^* \Gamma(t) \leq W(t) \leq c_2^* \Gamma(t)$. By combining this inequality with (4.120), the property (4.45) stated in Theorem 4.1 is obtained, with $c_3 = c_2^*/c_1^*$, $c_4 = \nu^2 \eta$, and $\alpha^*(\Gamma_R(s)) = \alpha_3^*(\Gamma_R(s))/c_1^*$.

In order to prove convergence when $\dot{X}_R(t) \to 0$ as $t \to \infty$, we first analyze the convergence properties of $\alpha_3^*(\Gamma_R(t))$. As $q_R \in \mathscr{C}^5$ due to Assumption 4.1, it can be observed that $\Gamma_R(t)$ and consequently $\alpha_3^*(\Gamma_R(t))$ converge to 0 as $t \to \infty$. Thus, $\forall \varepsilon > 0$, $\exists T_1 > 0$ such that:

$$\alpha_3^*(\Gamma_R(t)) \leq \varepsilon, \quad \forall t \geq T_1 \tag{4.121}$$

and thus consequently:

$$\dot{W}(t) \leq -\nu^2 \eta W(t), \quad \text{for} \quad W(t) \geq \varepsilon, \forall t \geq T_1 \tag{4.122}$$

Integrating this inequality from T_1 to t, it can be observed that $\forall \varepsilon > 0$, $\exists T_2 > 0$ such that:

$$W(t) \leq \varepsilon, \quad \forall t \geq T_2 \tag{4.123}$$

and thus we conclude:

$$\lim_{t\to\infty} W(t) = \lim_{t\to\infty} Z(t) = 0 \tag{4.124}$$

Last, we verify the assumed uniform boundedness of $X(t)$, $U(t)$ and their derivatives in Lemma 4.2. From the relationship (4.118), it can be observed that $W(t)$, and thus $S(t)$ and $Z(t)$ are uniformly bounded. As both $Z(t)$ and $X_R(t)$ are uniformly bounded, $X(t)$ is consequently uniformly bounded. Additionally, through the application of Lemma 4.4 and Assumption 4.1 to (4.19), $U(t)$ can be bounded in terms of $W(t)$ and is consequently uniformly bounded.

To obtain a bound for $\dot{U}(t)$, (4.16) is substituted into (4.47). Taking the partial derivative of this result with respect to x yields:

$$\hat{D}(t)\dot{U}(t+\hat{D}(t)(x-1)) = \hat{w}_x(x,t) + \frac{\partial \kappa}{\partial \hat{p}}(\hat{p}(x,t),\hat{r}(x,t))\hat{p}_x(x,t) + \frac{\partial \kappa}{\partial \hat{r}}(\hat{p}(x,t),\hat{r}(x,t))\hat{r}_x(x,t) \tag{4.125}$$

Evaluating this equation at $x = 1$ yields:

$$\hat{D}(t)\dot{U}(t) = \hat{w}_x(1,t) + \frac{\partial \kappa}{\partial \hat{p}}(\hat{p}(1,t),\hat{r}(1,t))\hat{p}_x(1,t) + \frac{\partial \kappa}{\partial \hat{r}}(\hat{p}(1,t),\hat{r}(1,t))\hat{r}_x(1,t) \tag{4.126}$$

Through the application of Lemma 4.4 to bound $\hat{p}(1,t)$, Lemma 4.5 to bound $\hat{p}_x(1,t)$, Lemma 4.6 and Lemma 4.2 to bound $\hat{w}_x(1,t)$, noting that $\left|\tilde{u}(0,t)\right|_1$ is uniformly bounded due to the uniform boundedness of $U(t)$, and the fact that $\hat{D}(t) \in [\underline{D}, \overline{D}]$, $\dot{U}(t)$ can be bounded in terms of $S(t)$ and $U(t)$ and is consequently uniformly bounded.

4.5 Remarks on Delay-Adaptive Control Law

In this research effort, our proposed delay-adaptive control law for a high-DOF robot manipulator consists of a predictor (4.17) of the system state after the delayed input reaches the system, a globally exponentially stable controller (4.19) for the delay-free system, and a gradient-based estimator of the input delay (4.32). It should be noted that this structure of the delay-adaptive control law allows for a wide selection of possible controllers for the delay-free system. One such permissible controller with (global) exponential stability is the feedback linearization based controller, a popular control strategy for nonlinear systems that we have previously utilized in a prediction-based control law for a known input delay [6, 9]. Compared to our previous work, this research effort possesses the following key differences:

- In our previous work, the input delay was a known constant parameter. In this research effort, it is an unknown constant parameter.
- In our previous work, a predictor was employed without delay adaptation to compensate for a known delay. In this research effort, we perform both prediction (4.17) and delay adaptation (4.32) in order to compensate for an unknown delay.
- In our previous work, we compared the performance of a control law with prediction to a control law without prediction in simulations and experiments. In this research effort, we compare the performance of a control law with prediction and delay adaptation to a control law with prediction, but without delay adaptation, in simulations and experiments.

It is important to note that the properties of the delay estimator (4.32) stated in Lemma 4.2 are contingent on the initial delay estimation error $\left|\tilde{D}(0)\right|$ being less than a critical value $\tilde{D}_{\max}$. Due to the strongly nonlinear relation between the estimated delay $\hat{D}$ and the gradient of X_P given by (4.28), obtaining even a conservative mathematical expression for $\tilde{D}_{\max}$ is a difficult task [82]. As a mathematical expression for $\tilde{D}_{\max}$ is not currently known, and $\left|\tilde{D}(0)\right|$ is initially unknown, the necessary conditions for Lemma 4.2 may seem restrictive. However, it is

still possible to determine an estimate for $\tilde{D}_{\max}$ through repeated simulations and experiments. As we demonstrate in the next section through both simulations and experiments, the delay estimator (4.32) is capable of correcting a significant initial delay mismatch, both in cases of overestimation and underestimation. Thus, when a reasonable initial estimate of the delay is available, and thus we can upper bound $\left|\tilde{D}(0)\right|$, the properties stated in Lemma 4.2 can safely be assumed to hold.

In order to implement the proposed delay-adaptive control strategy we have proposed, several numerical approximations are necessary. The governing equation (4.17) for $\hat{p}$, and the governing equation (4.20) for X_P are both ODEs, and thus can be numerically solved by a variety of different methods, such as Euler's method, Heun's method, and RK4. In simulations as well as experiments, we have observed that utilizing Euler's method provides sufficient accuracy in the estimation of $\hat{p}$ and X_P at the least computational burden out of the tested methods. Once X_P is determined, $\frac{\partial X_P}{\partial \hat{D}}$ can be determined through a trapezoidal Reimann sum, as (4.28) is a definite integral. However, solving for $\frac{\partial X_P}{\partial \hat{D}}$ is complicated by the presence of Φ_0, which is governed by the ODE given in (4.29) with initial condition (4.30). While this ODE can technically be evaluated in the same manner as $\hat{p}$ and X_P, it is infeasible to compute in practice due to the function $\frac{\partial f_0}{\partial X_P}$ present in the ODE containing computationally expensive terms to calculate such as $\frac{\partial C}{\partial X_P}$, which is a 7 x 7 x 14 tensor. However, for a sufficiently small value of β, $\frac{\partial \Phi_0}{\partial X_P} \approx 0$ and thus Φ_0 can be approximated as the identity matrix. Alternatively, it is possible to avoid calculation of the integral equation (4.28) through the use of finite difference methods:

$$\frac{\partial X_P}{\partial \hat{D}}(1,t,\hat{D}) \approx \frac{X_P(1,t,\hat{D}+h) - X_P(1,t,\hat{D})}{h} \tag{4.127}$$

where h is the timestep of the controller. In this method, $X_P(1,t,\hat{D}+h)$ is determined in the same manner as $X_P(1,t,\hat{D})$, and thus involves the solution of an additional ODE. In practice, approximating Φ_0 as the identity matrix and utilizing the presented finite difference method have yielded near identical results. In simulations and experiments, we have opted to approximate Φ_0

as the identity matrix.

4.6 Simulation and Experimental Results

In order to assess the performance of the delay-adaptive approach, we perform both a series of simulations using ODE methods on Baxter's dynamic equation (2.1), as well as several corresponding experiments. In each simulation and experiment, Baxter must track the six second trajectory specified in Chapter 2, while initially suffering from a large delay mismatch. Two cases are studied here, an underestimation of 0.9 seconds (0s initial prediction, 0.9s actual delay), as well as an overestimation of 0.5 seconds (0.9s initial prediction, 0.4s actual delay). These large delay mismatches are intentionally chosen in order to demonstrate the ability of the delay-adaptive approach to achieve stability under conditions that would cause a purely predictor-based approach to fail. In each simulation and experiment, the robot manipulator is commanded to remain stationary for a length of time equal to the initial estimated delay, then follow the 6 second pick and place trajectory described previously. This initial stationary period was chosen so that if the initial estimated delay were in fact equal to the actual input delay, the robot manipulator would be able to perfectly track the desired trajectory.

In both the simulation and experiment, we utilized an adaptation rate of $\gamma = 40$ and a replay length of $\beta = 0.1s$. For the control law for the delay free system described in Assumption 4.3, we utilized the following feedback linearization based controller proposed in [6, 9]:

$$\kappa(Z, X_R) = M(z+q_R)[\ddot{q}_R - k_1 k_2 z - (k_1+k_2)\dot{z}] + N(z+q_R, \dot{z}+\dot{q}_R) \tag{4.128}$$

with $k_1 = k_2 = 5$.

4.6.1 Trajectory Tracking without Delay Adaptation when the Delay is Underestimated

In order to compare the performance of the delay-adaptive approach studied in this research effort to a predictive approach without delay adaptation, we first performed several

simulations in which the input delay is underestimated. Simulations are performed at an estimated delay of 0.9 seconds, 0.85 seconds, 0.8 seconds, and 0.78 seconds in order to examine the destabilizing effect of several magnitudes of delay mismatch.

The simulated and desired joint trajectories of Baxter can be seen in Fig. 4.1 (a-g). When the estimated delay is 0.85 seconds, the predictive approach without delay adaptation still manages to closely track the desired trajectory. This indicates that even without delay adaptation, the predictive approach has a small degree of robustness to a mismatch in delay. However, when the estimated delay is 0.8 seconds, the trajectory tracking appears to become worse throughout the procedure, with several large oscillations observed in the last second of the simulation. At a slightly larger delay mismatch when the estimated delay is 0.78 seconds, oscillations are observed throughout the procedure, with a large divergence from the desired trajectory at the end of the simulation. Thus, without delay adaptation, the tracking performance of the predictive approach is significantly reduced in the presence of a delay mismatch.

The simulated joint torque input signals can be seen in Fig. 4.1 (h-n). Significant issues can be observed in the behavior of these torque signals. When the estimated delay is 0.85 seconds, several large oscillations can be observed in the torque input signal, which are not present when the estimated delay is equivalent to the true input delay. This indicates that even a relatively small delay mismatch can noticeably impact the predictive approach without delay adaptation. Observing the behavior of the 0.8 seconds estimated delay, as well as that of the 0.78 seconds estimated delay, it is clear that both torque signals are unstable. In particular, at an estimated delay of 0.78 seconds, the input torque signal rapidly oscillates between the minimum and maximum torque output of Baxter during the last second of the procedure. Such a torque signal would be likely to damage the actuators of the robot manipulator, and demonstrates dangerous behavior by a control scheme.

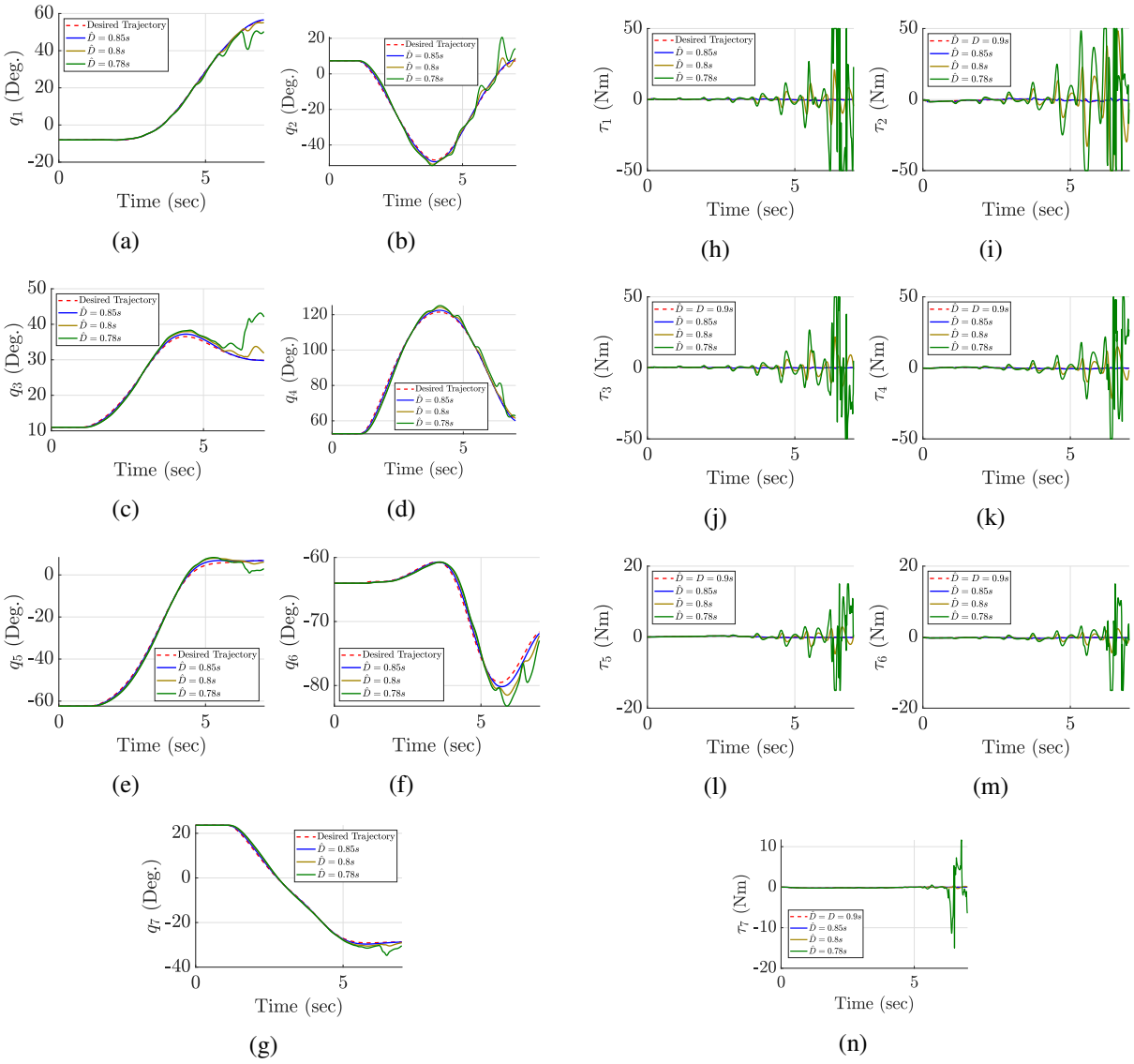


Figure 4.1. The simulated trajectories (a-g) and joint torque input signals (h-n) of Baxter performing a pick and place task without delay adaptation, with an input delay of 0.9s. Simulations are performed at an estimated delay of 0.9s (red dashed line), 0.85s (blue line), 0.8s (yellow line), and 0.78s (green line).

4.6.2 Trajectory Tracking when the Delay is Underestimated

The experimental, simulated, and desired joint trajectories of Baxter can be seen in Fig. 4.2 (a-g). Despite the large initial delay mismatch, the delay-adaptive approach is effective at driving the robot manipulator towards the desired trajectory. After the initial 0.9 seconds of operation, in which the robot manipulator was expectedly stationary due to the input delay, the robot manipulator quickly corrects itself towards the desired trajectory. This behavior can also be observed in Fig. 4.3, as both the simulated and experimental joint tracking errors decrease rapidly after around 1 second of operation. Furthermore, both the experimental and simulated trajectories appear to be smooth, indicating that changes to the estimated delay during adaptation did not cause disturbances in the tracking performance of the manipulator. Thus, as the delay adaptive approach studied here is capable of handling an initial delay mismatch of 0.9 seconds in the case of underestimation, it is capable of handling a much larger delay mismatch than without delay adaptation, which could only safely handle a mismatch of 0.05 seconds.

The experimental and simulated joint torque input signals can be seen in Fig. 4.2 (h-n). It is important to note that these torques are significantly lower than the maximum torque output of Baxter's joints, which are 50 Nm for joints 1-4, and 15 Nm for joints 5-7. Thus, the delay-adaptive approach is able to compensate for a large delay underestimation without producing excessive joint torques. In the simulation and experiment, slight chattering can be observed in the input joint torque signal. This chattering, which is most prominent during the 1st 2 seconds of the simulation, is caused by large changes in the estimated delay in the beginning of the simulation. This behavior is not of concern however, as the chattering is of a small amplitude, and is mostly eliminated after 2 seconds.

The adaptation of the estimated delay in the experiment and simulation can be seen in Fig. 4.4. In both the experiment and simulation, the estimated delay quickly converges to the actual delay. Furthermore, the following important observations can be made regarding the behavior of the adaptation. First of all, the rate of change of the adaptation appears to be constant during the

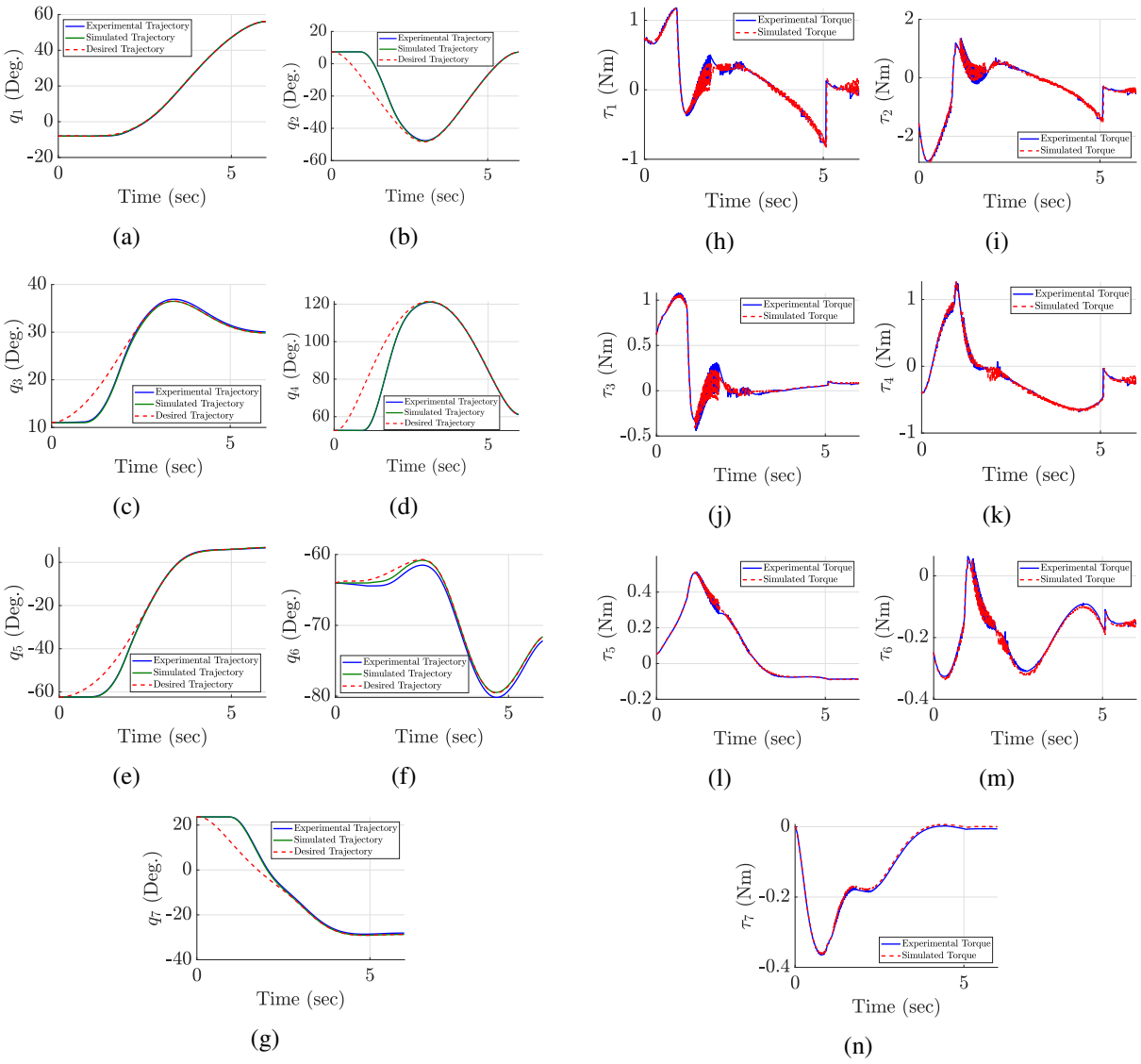


Figure 4.2. The experimental (blue line), simulated (green line), and desired (red dashed line) joint trajectories (a-g) of Baxter, as well as the experimental (blue line) and simulated (red dashed line) joint torque input signals (h-n) of Baxter. The input delay of the system is initially underestimated (0s initial prediction, 0.9s actual delay).

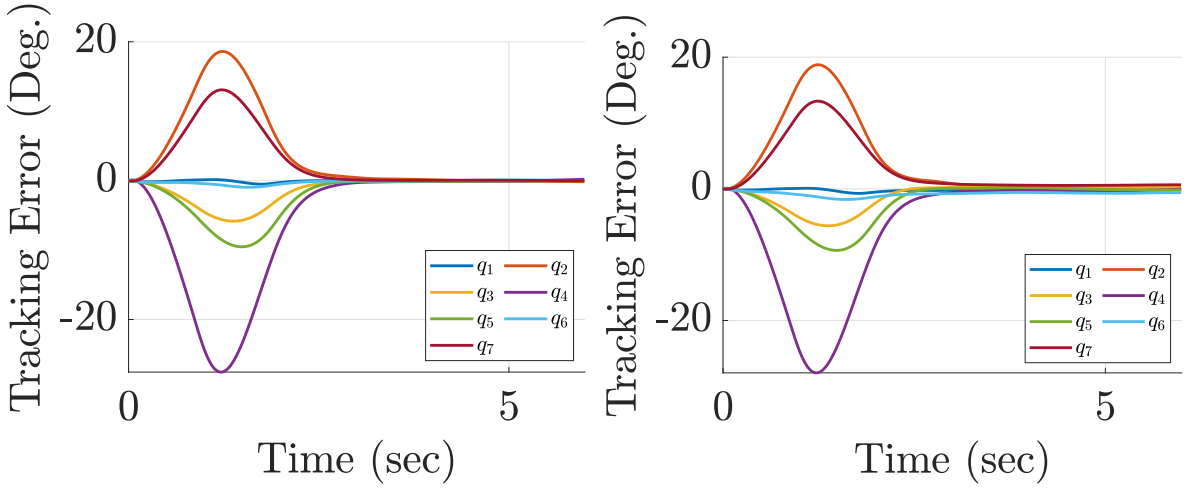


Figure 4.3. The simulated (a) and experimental (b) joint tracking errors of Baxter. The input delay of the system is initially underestimated (0s initial prediction, 0.9s actual delay).

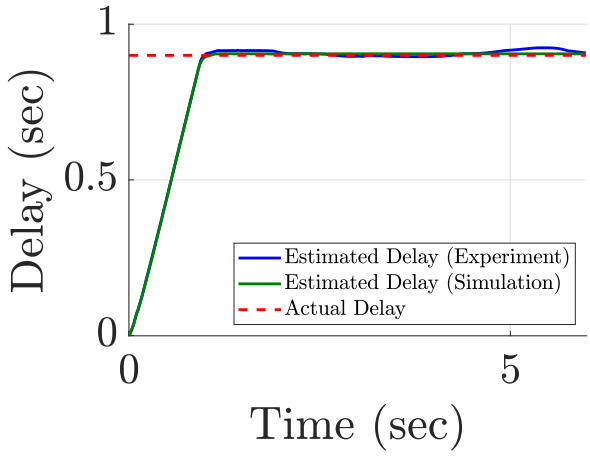


Figure 4.4. The adaptation of the estimated delay in experiment (blue line) and simulation (green line), compared to the actual input delay (red dashed line). The input delay of the system is initially underestimated (0s initial prediction, 0.9s actual delay).

1st 0.9 seconds of the simulation and experiment, until the estimated delay coincides with the actual delay. Due to the initial state of the robot manipulator being at rest, delays longer than the elapsed time t are indistinguishable from a delay of t seconds. Thus, as a consequence of the properties stated in Lemma 4.2, the estimated delay is upper bounded by the elapsed time in the simulation and experiment. Second of all, we observe that the estimation of the delay in simulation and experiment follow a nearly identical curve. This indicates that the delay-adaptive procedure does not suffer significantly from factors such as measurement noise of joint states which are present in the experiment but not in the simulation. Finally, it can be seen that there is slight overshoot in the estimated delay during the experiment. Although this behavior technically violates Lemma 4.2, it can reasonably be attributed to discretization of the control law.

4.6.3 Trajectory Tracking when the Delay is Overestimated

The experimental, simulated, and desired joint trajectories of Baxter can be seen in Fig. 4.5 (a-g). As was the case for an underestimated delay, the delay-adaptive approach is effective at driving the robot manipulator towards the desired trajectory. After the initial 0.4 seconds of operation, the manipulator starts following the curve of the desired trajectory. However, the manipulator was intended to remain stationary for 0.9 seconds, and thus has begun to accumulate error. After 1.5 seconds of operation, a shift in the manipulator behavior is observed, as the robot manipulator quickly moves to align to the desired trajectory. After 2 seconds have elapsed, the manipulator achieves near-perfect tracking of the desired trajectory for the remainder of the task. This behavior can also be observed in Fig. 4.6, as the errors increase after 0.4 seconds, reach a maximum at 1.5 seconds, and taper off after 2 seconds. As was the case for the underestimated delay, both the simulated and experimental trajectories appear to be smooth, indicating that changes to the estimated delay during adaptation did not cause disturbances in the tracking performance of the manipulator.

The experimental and simulated joint torque input signals can be seen in Fig. 4.5 (h-n). As was the case with underestimation of the delay, the generated input torques are significantly

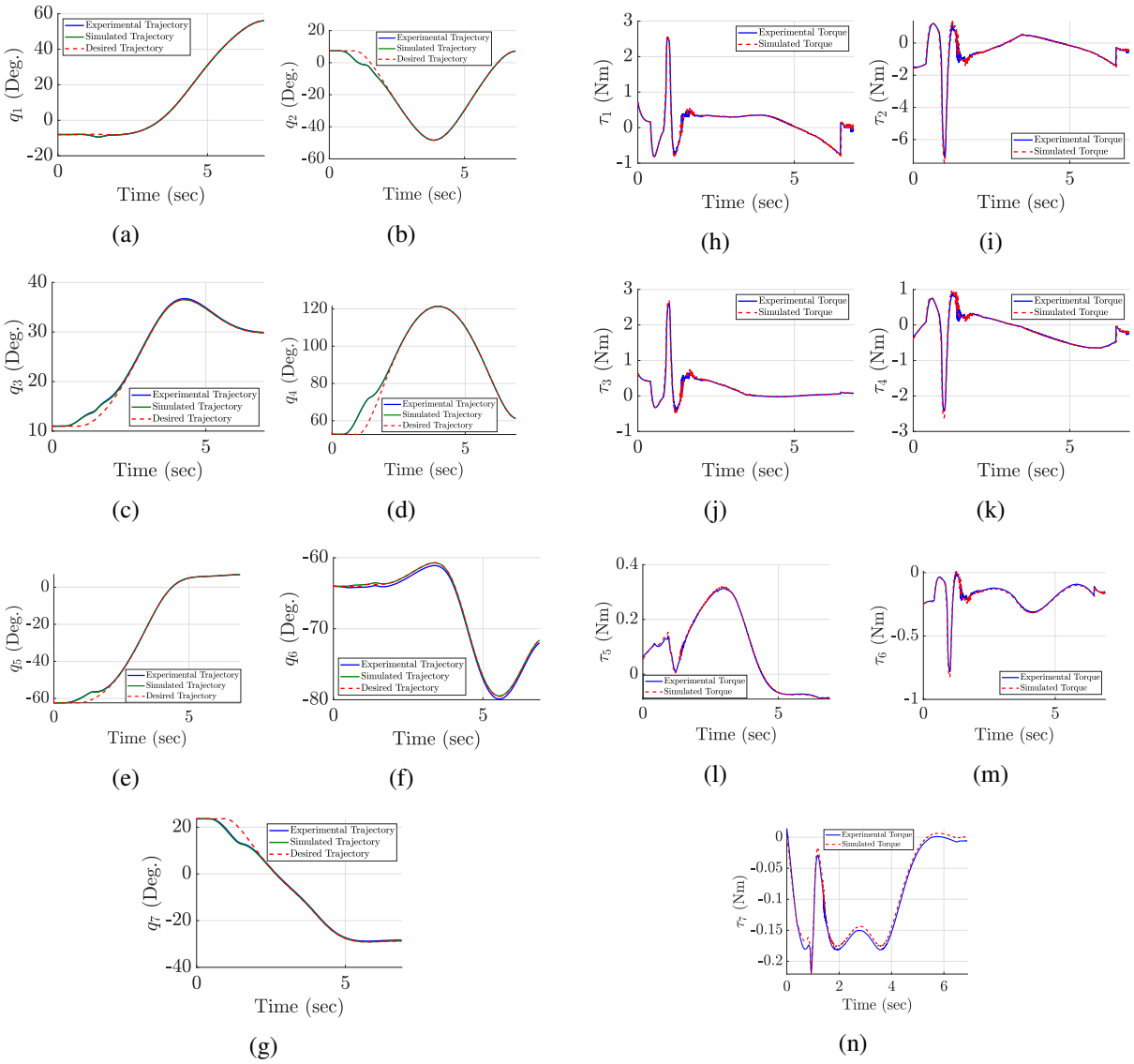


Figure 4.5. The experimental (blue line), simulated (green line), and desired (red dashed line) joint trajectories (a-g) of Baxter, as well as the experimental (blue line) and simulated (red dashed line) joint torque input signals (h-n) of Baxter. The input delay of the system is initially overestimated (0.9s initial prediction, 0.4s actual delay).

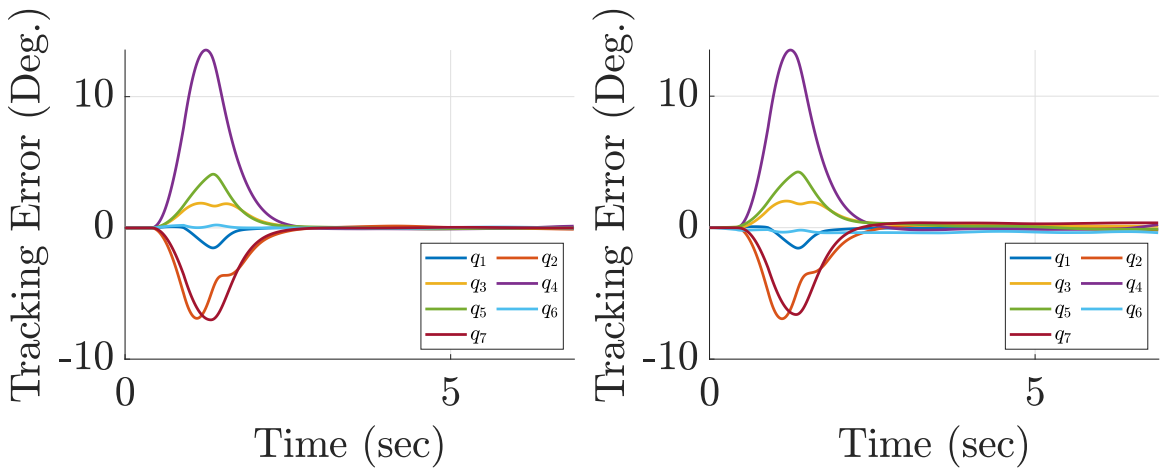


Figure 4.6. The simulated (a) and experimental (b) joint tracking errors of Baxter. The input delay of the system is initially overestimated (0.9s initial prediction, 0.4s actual delay).

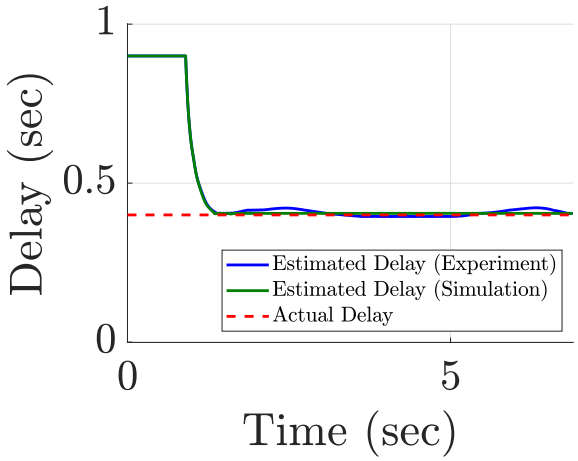


Figure 4.7. The adaptation of the estimated delay in experiment (blue line) and simulation (green line), compared to the actual input delay (red dashed line). The input delay of the system is initially overestimated (0.9s initial prediction, 0.4s actual delay).

lower than the maximum torque output of Baxter's joints. Thus, the delay-adaptive approach is able to compensate for a large delay overestimation without producing excessive joint torques. Compared to the case of underestimated delay, there does not appear to be chattering in the torque input signal. However, several joints exhibit a large spike in the torque input signal at around 1 second of operation, with smaller spikes at 0.4 seconds and 1.5 seconds. These torque spikes are likely due to the initial unexpected robot manipulator motion, the start of delay adaptation, and the convergence of delay adaptation respectively. Thus, qualitatively distinct behavior is observed between an underestimation and an overestimation of the delay, although the robot manipulator displays near perfect tracking throughout the majority of the procedure in both cases.

The adaptation of the estimated delay in the experiment and simulation can be seen in Fig. 4.7. Several important observations can be made regarding the behavior of the adaptation. During the 1st 0.9 seconds of operation, no changes are visible in the estimated delay. This behavior is due to the local nature of the gradient-based delay estimate approach. As previously described while examining the behavior of the delay adaptation when the delay was underestimated, delays longer than the elapsed time t are indistinguishable from a delay of t seconds. Thus, the gradient is 0 around the estimated delay of 0.9 seconds, until at least 0.9 seconds have elapsed in the procedure. After the operation time reaches 0.9 seconds, the estimated delay quickly converges to the actual delay. It is important to note that the behavior of the adaptation matches the observed behavior in the joint positions and torques. The estimated delay starts adapting at 0.9 seconds and finishes adapting around 1.5 seconds, which aligns with the spikes observed in the torque profiles, as well as changes in the joint error signals.

4.7 Conclusion

In this chapter, we investigated the analytical and experimental trajectory-tracking control of a 7-DOF robot manipulator with an unknown long actuator delay. In order to compensate for

this unknown delay, we formulated a delay-adaptive prediction-based control strategy in order to simultaneously estimate the unknown delay while driving the robot manipulator towards the desired trajectory. To the best of the authors' knowledge, this chapter is the first to present a delay-adaptive approach for a nonlinear system with multiple inputs. Through Lyapunov analysis utilizing the $\mathscr{L}_1$ norm, we obtained a local asymptotic stability result of the proposed controller. Then, we demonstrated through both simulation and experiment that the proposed controller is capable of achieving desirable trajectory tracking performance, even in the case of a large initial delay mismatch. As shown in Fig. 4.8, the delay-adaptive approach significantly improves the tracking performance of the Robot Manipulator when there is a delay mismatch, without sacrificing performance when the delay is properly identified. This research represents a large improvement upon the predictor-based approach in the case of an unknown delay, and thus has promising potential for use cases in which the delay is difficult to accurately predict or measure directly.

(a)

Figure 4.8. Baxter performing a pick and place task while subjected to an input delay of 0.9 seconds (0s initial delay estimate).

4.8 Acknowledgements

The authors in this research effort would like to extend their gratitude to professor Delphine Bresch-Pietri, both for her work in developing the original delay-adaptive method this chapter extends upon, as well as providing valuable insight behind several formulations present in her research.

This chapter contains adaptations of the following papers: 1) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Experimental and Analytical Delay-Adaptive Control of a 7-DOF Robot Manipulator." *2021 American Control Conference (ACC)*. IEEE, 2021. 2) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Delay-Adaptive Control of a 7-DOF Robot Manipulator: Design and Experiments." *IEEE Transactions on Control Systems Technology*, In Press, 2022. The dissertation author is the primary investigator and author of these papers.

This chapter is based upon work supported by the National Science Foundation under Award #1823951-1823983. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Chapter 5

Prescribed-Time Control

We present an analytical design and experimental verification of trajectory tracking control of a 7-DOF robot manipulator, which achieves convergence of all tracking errors to the origin within a finite terminal time, also referred as the "settling time." A key feature of this control strategy is that the settling time is explicitly assigned by the control designer to a value desired, or "prescribed" by the user, and that the settling time is independent of the initial conditions and of the reference signal. In order to achieve this beneficial property with the controller, a scaling of the state by a function of time that grows unbounded towards the terminal time is employed. Through Lyapunov analysis, we first demonstrate that the proposed controller achieves regulation of all tracking errors within the prescribed time as well as the uniform boundedness of the joint torques, even in the presence of a matched, non-vanishing disturbance. Then, through both simulation and experiment, we demonstrate that the proposed controller is capable of converging to the desired trajectory within the prescribed time, despite large distance between the initial conditions and the reference trajectory, i.e., in spite of large initial tracking errors, and in spite of a sinusoidal disturbance being applied in each joint.

5.1 Background

In many applications where robot manipulators are utilized, the convergence time of the underlying controller plays a crucial role. In many tasks there are strict requirements on the

maximum duration of convergence, and thus a failure to achieve convergence by the required time could lead to the inability of the robot manipulator to perform its task. Convergence time also plays a role in the planning and reliability of robot manipulators. For example, if an accurate estimate of the convergence time is known for a certain task, an accurate and reliable estimate on the productivity of the robot manipulator can be made. Expanding on this point, when multiple robot manipulators are used cooperatively, such as in an assembly line in industrial applications, having reliable estimates of the completion time of each individual task is crucial in order to effectively plan the operation of each manipulator. A considerable amount of research has been devoted towards the development of control methods for robot manipulators which are capable of guaranteeing an upper bound on the convergence time (potentially dependent on initial conditions), achieving convergence to zero within a finite period of time.

The literature on finite-time convergence methods concerning robot manipulators can be broadly organized into three distinct categories: the finite-time methods [83–87], the fixed-time methods [88–94], and the prescribed-time methods [95–107].

Finite-time methods are characterized by a finite convergence time that is bounded by the norm of the initial condition, as well as a function of the controller parameters. While these methods are useful in order to obtain more consistent convergence results, obtaining a specific desired convergence time requires determining the maximum initial conditions of the task to be completed, then tuning the controller parameters based on this value. Thus, in order to complete a larger set of tasks, with different initial conditions and maximum allowable operation times, separate controller parameters must be determined for each task.

Fixed-time methods are characterized by a finite convergence time that is bounded by a function of the controller parameters which is independent of the initial conditions. Thus, the process for tuning the controller parameters for a specific task is considerably simplified, as one no longer needs to consider the maximum expected initial conditions of the task, only the maximum required completion time. However, it is important to note that this upper bound of the convergence time is typically conservative, and thus the robot manipulator will usually complete

the task well before the required completion time is exceeded. Additionally, depending on the implementation of the fixed-time controller, the upper bound of the finite convergence time may not be able to be arbitrarily set, meaning that certain maximum completion times may be too stringent for the controller to effectively handle.

Prescribed-time methods are characterized by a finite convergence time that is explicitly prescribed as a controller parameter. This desirable property of prescribed-time methods enables the same set of controller parameters to be utilized for a wide variety of tasks with different required completion times. Due to this desirable property, the development of prescribed-time methods has become an active research topic in recent years.

The first design of a prescribed-time stabilizing (and disturbance rejecting) controller was introduced by Song *et al.* [95], who employed a scaling of the state of a normal-form nonlinear system by a function of time that grows unbounded towards the terminal time. By stabilizing the system in the scaled representation, regulation in prescribed finite time is achieved for the original state, along with a smooth, uniformly bounded control input and the rejection of a matched non-vanishing disturbance. Another important class of prescribed-time controllers for robot manipulators, introduced by Becerra *et al.* [97] and improved upon by Obregón-Flores *et al.* [98], utilizes time base generators, which are state trajectories designed such that the system state smoothly converges to zero at the prescribed terminal time. A key feature, and arguably a disadvantage, of this control method is the explicit use of the initial conditions in the controller structure as a feedforward term, along with a sliding-mode control scheme to correct for a uniformly bounded matched uncertainty. Notably, this scheme exhibits prescribed-time convergence in the ideal case of no matched uncertainty, and finite-time convergence when uncertainties are present. In a separate approach, Cao *et al.* [102] utilizes a scaling system transformation technique to transform the Euler-Lagrange system considered into a new set of variables, in which the boundedness of the variables ensures that both partial and full state constraints will not be violated. In addition, this transformation also ensures that for any time greater than the prescribed convergence time, the remaining tracking errors will be less than a

prescribed value. This approach is notable in that the scaling transformation utilized does not approach infinity as the terminal time is approached, and thus numerical difficulties caused by an unbounded gain are avoided in this method. However, a potential drawback to using this method is that the controller does not allow for separate control gains for each joint, meaning that aggressive torques are likely applied to certain joints of the robot manipulator when there is a large difference in inertia between joints, which is typically the case for high-DOF robot manipulators. In another approach, Garg and Panagou [94] utilize the concept of robust fixed-time control Lyapunov functions as well as control barrier functions to provide a framework for ensuring robustness to disturbances with fixed-time convergence to a user-defined goal set, along with ensuring the system state remains in a user-specified safe set throughout the operation.

In this effort, we reformulate the prescribed-time controller initially developed by Song *et al.* [95] in order to handle the case of trajectory tracking with a robot manipulator. This formulation yields convergence of the tracking errors to the origin within the prescribed terminal time, even in the presence of model uncertainties and a non-vanishing matched disturbance. Furthermore, in order to address the practical issues that can arise when employing an unbounded scaling of the state, due to factors such as measurement noise, numerical issues when applying a large scaling to small errors, and a finite controller frequency, we employ a gain-clipping strategy in order to limit the scaling of the state to a sufficiently high value. Through the experimental verification of this prescribed-time control strategy with gain clipping on Baxter, a 7-DOF redundant robot manipulator, we demonstrate convergence of the tracking errors to a small neighborhood of zero by the prescribed terminal time, despite a significant initial angular position tracking error of 20 degrees on each joint, as well as a sinusoidal torque disturbance of $0.1 \sin 5t$ applied to each joint. Thus the prescribed-time control strategy studied here is both theoretically sound and effective in practice.

The organization of the rest of this chapter is as follows. In Section 5.2, we present the design of the prescribed-time trajectory-tracking controller. In Section 5.3, we demonstrate the prescribed-time regulation of the robot manipulator tracking errors through Lyapunov Analysis.

In Section 5.4, we briefly discuss the practical implementation of the proposed control law. In Section 5.5, we present the simulation and experimental results of the proposed method implemented on Baxter's right manipulator, achieving convergence of the tracking errors to a small neighborhood of zero by the prescribed terminal time despite large initial tracking errors and a non-vanishing matched disturbance. Finally, in Section 5.6, we present the case that the proposed method is theoretically sound, straightforward to implement in a real system, and ultimately effective in practice.

Notations: In the following, we use the common definitions of class $\mathscr{K}$ and $\mathscr{KL}$ given in [79]. $|\cdot|$ refers to the Euclidean norm, the matrix norm is defined accordingly, for $M \in \mathscr{M}_\ell(\mathbb{R})(\ell \in \mathbb{N}^*)$, as $|M| = \sup_{|x|\leq 1} |Mx|$ and the spatial norm is defined as follows:

$$\|f\|_{[a,b]} = \sup_{t\in[a,b]} \left|f(t)\right|$$

5.2 Prescribed-Time Tracking for Robot Manipulators

In order to formulate a controller that is robust to modeling uncertainty, the values of the mass matrix, gravity vector, and frictional torques derived from our dynamic model of Baxter are treated as estimates, and are denoted as $\hat{M}(q), \hat{G}(q), \hat{F}(\dot{q})$, respectively. We make the following assumptions concerning the difference between our dynamic model and the true dynamics of Baxter:

Assumption 5.1. *The true and estimated values of the mass matrix, Coriolis matrix, gravity vector, frictional torques, and the disturbance torques satisfy the following inequalities:*

$$\left|M^{-1}(q)\hat{M}(q) - I\right| \leq c_1 \tag{5.1}$$

$$\left|M^{-1}(q)C(q,\dot{q})\dot{q}\right| \leq c_2|\dot{q}|^2 \tag{5.2}$$

$$\left|M^{-1}(q)(\hat{G}(q) - G(q))\right| \leq c_3 \tag{5.3}$$

$$\left|M^{-1}(q)(\hat{F}(\dot{q})-F(\dot{q}))\right| \leq c_4|\dot{q}| \tag{5.4}$$

$$\left|M^{-1}(q)D(t)\right| \leq c_5\left|D(t)\right| \tag{5.5}$$

Assumption 5.2. *The true mass matrix $M(q)$, and the estimated mass matrix $\hat{M}(q)$ are symmetric and positive definite.*

Furthermore, we make the following assumption regarding the reference joint trajectories:

Assumption 5.3. *The desired joint trajectories are designed such that $q_r(t)$, $\dot{q}_r(t)$, and $\ddot{q}_r(t) \in \mathbb{R}^7$ exist and are uniformly bounded for all $t \in [0,T]$, where $T > 0$ is the prescribed terminal time.*

We consider the following trajectory tracking system:

$$\dot{E} = \begin{bmatrix} \dot{E}_1 \\ \dot{E}_2 \end{bmatrix} = \begin{bmatrix} E_2 \\ M^{-1}(\tau - C\dot{q} - G - F - D) - \ddot{q}_r \end{bmatrix} \tag{5.6}$$

$$E = \begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} = \begin{bmatrix} q - q_r \\ \dot{q} - \dot{q}_r \end{bmatrix} \tag{5.7}$$

where $E \in \mathbb{R}^{14}$ is the state error vector, and $\varepsilon \in \mathbb{R}^7$ is the vector of joint angular position tracking errors.

In order to regulate this system in prescribed-time, we first introduce the following monotonically increasing scaling function, as well as it's inverse:

$$\mu_1(t) = \frac{T}{T-t}, \quad t \in [0,T) \tag{5.8}$$

$$\nu_1(t) = \frac{1}{\mu(t)_1} = \frac{T-t}{T}, \quad t \in [0,T) \tag{5.9}$$

where $T > 0$ is the prescribed terminal time, with the properties $\mu(0) = 1$, $\mu(T) = +\infty$, $\nu(0) = 1$ and $\nu(T) = 0$. To achieve prescribed-time regulation of the tracking errors, we introduce the

following change of coordinates:

$$w(t) = \mu(t)\varepsilon(t) \tag{5.10}$$

$$z(t) = \dot{w}(t) + \alpha w(t) \tag{5.11}$$

where

$$\mu(t) = \mu_1(t)^2 = \frac{1}{\nu_1^2} \tag{5.12}$$

and $\alpha > 0$. This change of coordinates results in the following forward and inverse scaling transforms:

$$Z = \begin{bmatrix} w \\ z \end{bmatrix} = \mu \begin{bmatrix} I & 0 \\ (\alpha + \mu_1 \frac{2}{T})I & I \end{bmatrix} E = P(\mu_1)E \tag{5.13}$$

$$E = \nu_1 \begin{bmatrix} \nu_1 I & 0 \\ (-\nu_1 \alpha - \frac{2}{T})I & \nu_1 I \end{bmatrix} Z = Q(\nu_1)Z \tag{5.14}$$

where $I \in \mathbb{R}_{7\times 7}$ is the identity matrix, $P(\mu_1) \in \mathbb{R}^{7\times 7}$ is the forward scaling transform, $Q(\nu_1) \in \mathbb{R}^{7\times 7}$ is the inverse scaling transform, and $Z \in \mathbb{R}^{14}$ is the scaled state error vector. By taking the time derivative of (5.13) and substituting the inverse transformation (5.14), the dynamics of the scaled state error vector are obtained:

$$\dot{w} = z - \alpha w \tag{5.15}$$

$$\dot{z} = \mu \left[\ddot{\varepsilon} - \left(\alpha^2 \nu_1^2 + \alpha \nu_1 \frac{4}{T} + \frac{2}{T^2} \right) w + \left(\nu_1 \frac{4}{T} + \alpha \nu_1^2 \right) z \right] \tag{5.16}$$

where

$$\ddot{\varepsilon} = M^{-1}(\tau - C\dot{q} - G - F - D) - \ddot{q}_r \tag{5.17}$$

Before presenting the design of the prescribed-time control law, it is first necessary to

present several definitions concerning notions of stability within a finite prescribed interval of time.

Definition 5.1 (FT-ISS [95])**.** The system $\dot{x} = f(x,t,d)$ (of arbitrary dimensions of x and d) is said to be *fixed-time input-to-state stable in time T* (FT-ISS) if there exists a class $\mathscr{KL}$ function β and a class $\mathscr{K}$ function γ, such that, for all $t \in [0,T)$:

$$\left|x(t)\right| \leq \beta\left(|x_0|, \mu_1(t) - 1\right) + \gamma\left(\|d\|_{[0,t]}\right) \tag{5.18}$$

Definition 5.2 (FT-ISS+C [95])**.** The system $\dot{x} = f(x,t,d)$ (of arbitrary dimensions of x and d) is said to be *fixed-time input-to-state stable in time T and convergent to zero* (FT-ISS+C) if there exist class $\mathscr{KL}$ functions β and β_f, and a class $\mathscr{K}$ function γ, such that, for all $t \in [0,T)$:

$$\left|x(t)\right| \leq \beta_f\left(\beta\left(|x_0|, \mu_1(t) - 1\right) + \gamma\left(\|d\|_{[0,t]}\right), \mu_1(t) - 1\right) \tag{5.19}$$

As the function $\mu_1(t) - 1$ starts at zero and grows monotonically to infinity as $t \to T$, a system that is FT-ISS is also ISS, with the additional property that in the absence of a disturbance d, it is fixed-time globally asymptotically stable in time T. Additionally, a system that is FT-ISS+C is also FT-ISS, with the additional property that the state converges to zero even in the presence of a disturbance.

Now, we present the design of the prescribed-time control law.

Theorem 5.1. *Under Assumptions 5.1-5.3, consider the system (5.6) with the controller:*

$$\tau(t) = -\hat{M}(q)\left[(k + \theta + \eta\,\psi(\dot{q})^2)z(t) + \ddot{q}_r(t)\right] + \hat{G}(q) + \hat{F}(\dot{q}) \tag{5.20}$$

where

$$\psi(\dot{q}) = |\dot{q}|^2 + |\dot{q}| + 1 \tag{5.21}$$

If the controller gains are chosen such that $\rho, k, \eta > 0$*,*

$$\rho k \alpha^2 > \frac{1}{4\lambda_M^2}, \tag{5.22}$$

and

$$\theta \geq \frac{1}{\lambda_M}\left(\alpha + \frac{4}{T}\right) + \rho\left(\alpha^2 + \alpha\frac{4}{T} + \frac{2}{T^2}\right)^2 \tag{5.23}$$

where

$$\lambda_M = \min_{q\in[0,2\pi)} \lambda_{\min}\left(\frac{M^{-1}(q)\hat{M}(q) + \hat{M}(q)M^{-1}(q)}{2}\right) \tag{5.24}$$

then the closed loop system (5.6) with (5.20) is FT-ISS+C and the joint torques τ *remain bounded over* $[0, T)$*.*

5.3 Lyapunov Analysis

For the purpose of the Lyapunov analysis, we propose the following Lyapunov function:

$$V = \frac{1}{2}|z|^2 \tag{5.25}$$

Taking the derivative of this function yields:

$$\begin{aligned}\dot{V} = \mu z^T \Bigg[&- M^{-1}\hat{M}(k + \theta + \eta\psi^2)z + (M^{-1}\hat{M} - I)\ddot{q}_r + M^{-1}\left(\hat{G} - G + \hat{F} - F - C\dot{q} - D\right) \\ &- \left(\alpha^2\nu_1^2 + \alpha\nu_1\frac{4}{T} + \frac{2}{T^2}\right)w + \left(\frac{4}{T}\nu_1 + \alpha\nu_1^2\right)z\Bigg]\end{aligned} \tag{5.26}$$

First, we seek to obtain an upper bound for the 1st term of $\dot{V}$. Utilizing the positive definite symmetric property of the mass matrices as stated in Assumption 5.2, we obtain the following inequality:

$$z^T M^{-1}\hat{M}z = z^T\left(\frac{M^{-1}\hat{M} + \hat{M}M^{-1}}{2}\right)z \geq \lambda_M|z|^2 \tag{5.27}$$

where λ_M is first defined in (5.24).

Next, we examine the second and third terms of $\dot{V}$. Through the application of Assumptions 5.1 and 5.3, the following inequality can be obtained:

$$(M^{-1}\hat{M}-I)\ddot{q}_r + M^{-1}\left(\hat{G}-G+\hat{F}-F-C\dot{q}-D\right) \leq \psi d \tag{5.28}$$

where

$$d(t) = \max\left\{c_1\|\ddot{q}_r\|_{[0,t]} + c_3 + c_5\|D\|_{[0,t]}, c_2, c_4\right\} \tag{5.29}$$

Applying (5.27) and (5.28) to (5.26), along with Young's inequality yields the following inequality:

$$\begin{aligned}\dot{V} \leq -\mu\lambda_M|z|^2(k+\theta+\eta\psi^2) + \mu\eta\lambda_M\psi^2|z|^2 + \frac{\mu}{4\eta\lambda_M}d^2 + \mu\rho\lambda_M|z|^2\left(\alpha^2\nu_1^2 + \alpha\nu_1\frac{4}{T} + \frac{2}{T^2}\right) \\ + \frac{\mu}{4\rho\lambda_M}|w|^2 + \mu|z|^2\left(\frac{4}{T}\nu_1 + \alpha\nu_1^2\right)\end{aligned} \tag{5.30}$$

Through the application of (5.23), this inequality can be further reduced:

$$\dot{V} \leq -2\mu\lambda_M kV + \frac{\mu}{4\eta\lambda_M}d^2 + \frac{\mu}{4\rho\lambda_M}|w|^2 \tag{5.31}$$

In order to proceed with the Lyapunov analysis, it is necessary to introduce a technical lemma from the work of Song *et al.* [95].

Lemma 5.1. *If a continuously differentiable function $V : [0,T) \to [0,+\infty)$ satisfies:*

$$\dot{V}(t) \leq -2k\mu(t)V(t) + \frac{\mu(t)}{4\lambda}d(t)^2 \tag{5.32}$$

for positive constants k, λ, where $\mu(t)$ is defined in (5.8), then:

$$V(t) \leq \xi(t)^{2k}V(0) + \frac{\|d\|^2_{[0,t]}}{8k\lambda}, \quad \forall t \in [0,T) \tag{5.33}$$

where ξ is the monotonically decreasing function:

$$\xi(t) = e^{T(1-\mu_1(t))} \tag{5.34}$$

with the properties that $\xi(0) = 1$ and $\xi(T) = 0$.

Through the application of this lemma to (5.31), it can be seen that:

$$\left|z(t)\right| \leq \xi(t)^{\lambda_M k}|z_0| + \frac{1}{2\lambda_M\sqrt{k}}\left(\frac{\|w\|_{[0,t]}}{\sqrt{\rho}} + \frac{\|d\|_{[0,t]}}{\sqrt{\eta}}\right) \tag{5.35}$$

and thus the z-system is FT-ISS w.r.t. the w-input with a gain of $\frac{1}{2\lambda_M\sqrt{k\rho}}$ and is also FT-ISS w.r.t the d-input. In order to obtain the behavior of the w-system, one can rearrange (5.11) to obtain $\dot{w}(t) = -\alpha w(t) + z(t)$. From this point, it is straightforward to obtain a bound on w:

$$\left|w(t)\right| \leq |w_0|e^{-\alpha t} + \frac{1}{\alpha}\|z\|_{[0,t]} \tag{5.36}$$

and thus the w-system is ISS w.r.t the z-input with a gain of $\frac{1}{\alpha}$. Thus by the small-gain theorem, if condition (5.22) is satisfied, then the combined system Z is ISS w.r.t. d and thus there exist constants $\Gamma, \delta, \gamma > 0$ such that:

$$\left|Z(t)\right| \leq \Gamma|Z_0|e^{-\delta t} + \gamma\|d\|_{[0,t]} \tag{5.37}$$

Through the substitution of the scaling transformation (5.13) into the right side of (5.37), followed by the substitution of the resulting inequality into the right side of the inverse scaling transformation (5.14), the following inequality is obtained:

$$\left|E(t)\right| \leq \nu_1(t)\left[\breve{\Gamma}|E_0|e^{-\delta t} + \breve{\gamma}\|d\|_{[0,t]}\right] \tag{5.38}$$

where

$$\breve{\Gamma} = \Gamma \left| P(1) \right| \max_{\nu_1 \in [0,1]} \left| Q(\nu_1) \right| \tag{5.39}$$

$$\breve{\gamma} = \gamma \max_{\nu_1 \in [0,1]} \left| Q(\nu_1) \right| \tag{5.40}$$

Due to the fact that $\nu_1(T) = 0$, this inequality establishes that the closed loop system (5.6) with (5.20) is FT-ISS+C. Due to the boundedness of $Z(t)$ established in (5.37), the boundedness of $E(t)$ established in (5.38), and the boundedness of q_r, $\dot{q}_r$, and $\ddot{q}_r$ established in Assumption 5.3, the uniform boundedness of the input τ is established from (5.20).

5.4 Remarks on Prescribed-Time Control Law

Through the substitution of the scaling transform (5.13) to the control law (5.20), it is possible to obtain an expression for the control law in terms of the joint angular position and velocity errors $\varepsilon, \dot{\varepsilon}$ rather than the scaled state z:

$$\tau = -\mu_1^2 \hat{M} \left[(k + \theta + \eta \psi^2) \left(\left(\alpha + \mu_1 \frac{2}{T} \right) \varepsilon + \dot{\varepsilon} \right) + \ddot{q}_r \right] + \hat{G} + \hat{F} \tag{5.41}$$

From this representation, the role of the controller parameters k, θ, η, and α can be observed. The sum $k + \theta$ is a scaled PD gain, and thus is the primary driver of the error signal to zero, η is the gain of the nonlinear damping term ψ, which aims to attenuate the effects of uncertainties on the control law, and α is a weighting factor which determines the ratio between the proportional and derivative gains of the control law. Thus, implementing the proposed prescribed-time control law control law requires the tuning of just three parameters (treating $k + \theta$ as one parameter), whose effect on the control law is readily observed. Furthermore, due to the direct dependence of the control law on the prescribed final time T, these three controller parameters need only be determined once for a given robot manipulator, regardless of the specific

tasks the manipulator needs to perform. Thus, the proposed control law can be readily applied to a wide variety of tasks with different convergence time constraints.

A potential barrier to the practical application of this proposed method is the consequences of employing an unbounded gain $\mu_1(t)$. While the proposed control law guarantees boundedness of the control torques $\tau(t)$ even in the presence of non-vanishing uncertainties, problems may still arise due to measurement noise, numerical issues when multiplying large gains with small errors, and a finite controller frequency. In order to combat these practical issues, one effective strategy that can be employed is gain clipping. Using this strategy, we define the constants

$$\zeta = 1 - \underline{\nu}_1 = \frac{1 - \bar{\mu}_1}{\bar{\mu}_1} \in (0,1) \tag{5.42}$$

$$\underline{\nu}_1 = \frac{1}{\bar{\mu}_1} = 1 - \zeta \in (0,1) \tag{5.43}$$

$$\bar{\mu}_1 = \frac{1}{\underline{\nu}_1} = \frac{1}{1-\zeta} \in (1, +\infty) \tag{5.44}$$

and redefine $\mu_1(t)$ in (5.41) as:

$$\begin{aligned} \mu_1(t) &= \min\left\{\mu_1(t), \bar{\mu}_1\right\} \\ &= \frac{1}{\max\{\nu_1(t), \underline{\nu}_1(t)\}} \\ &= \frac{T}{T - \min\{t, \zeta T\}} \end{aligned} \tag{5.45}$$

This redefinition of (5.8) upper bounds the scaling gain μ_1 by the value $\bar{\mu}_1 = \frac{1}{1-\zeta}$, ensuring that the controller gains do not grow past the point where the previously mentioned issues begin to noticeably affect the closed-loop system. A consequence of this modification is that the regulation of the tracking errors is to a small neighborhood of zero, rather than exactly zero as when utilizing an unbounded gain. Employing a ζ that is sufficiently close to 1 can ensure that this neighborhood is negligible, achieving performance that is qualitatively similar to that of utilizing an unbounded gain.

5.5 Simulation and Experimental Results

In order to assess the performance of the proposed prescribed-time approach, we perform both a simulation using ODE methods on Baxter's dynamic equation (2.1), as well as an experiment. In both the simulation and experiment, Baxter must track the six second trajectory specified in Chapter 2, while under the influence of a torque disturbance of $D(t) = 0.1\sin(5t)$ applied to each joint. In addition, this task is purposely started from a large initial angular position error of 20 degrees for each joint. Thus, this simulation and experiment demonstrates the ability of the proposed method to converge from a large initial condition to the desired trajectory within the prescribed finite time, while rejecting a large torque disturbance. The controller parameters used in both the simulation and experiment are $T = 6$, $k + \theta = 5$, $\eta = 0.005$, $\alpha = 2$, and $\zeta = 0.4$. Note that $\bar{\mu}_1^2$, which is employed in lieu of μ_1^2 in (5.41), which would without clipping go to infinity, is as low as 2.78.

From Figure 5.3 it can be observed that the prescribed-time controller is successful at executing the pick-and-place task in practice. Comparing the reference trajectory, highlighted by the green circles in the figure, to the initial position of Baxter's right end effector located at the bottom of the figure, it can be seen that an angular position error of 20 degrees in each joint corresponds to a large error in Cartesian coordinates. Despite this large initial tracking error, the prescribed-time controller is shown to be effective at quickly attenuating this tracking error, and achieving close tracking of the desired trajectory for the remainder of the operation. After only 1 second of operation, the distance between the desired trajectory and Baxter's right end effector is significantly reduced, and after 3 seconds of operation, Baxter's right end effector appears to coincide exactly with the desired trajectory.

The experimental, simulated, and desired joint trajectories can be seen in Figure 5.1. Despite the large initial joint tracking errors, as well as the large sinusoidal disturbance applied to the system, negligible tracking errors are achieved after around 2.5 seconds of operation. Throughout the procedure, oscillations in the joint angular positions can not be observed from

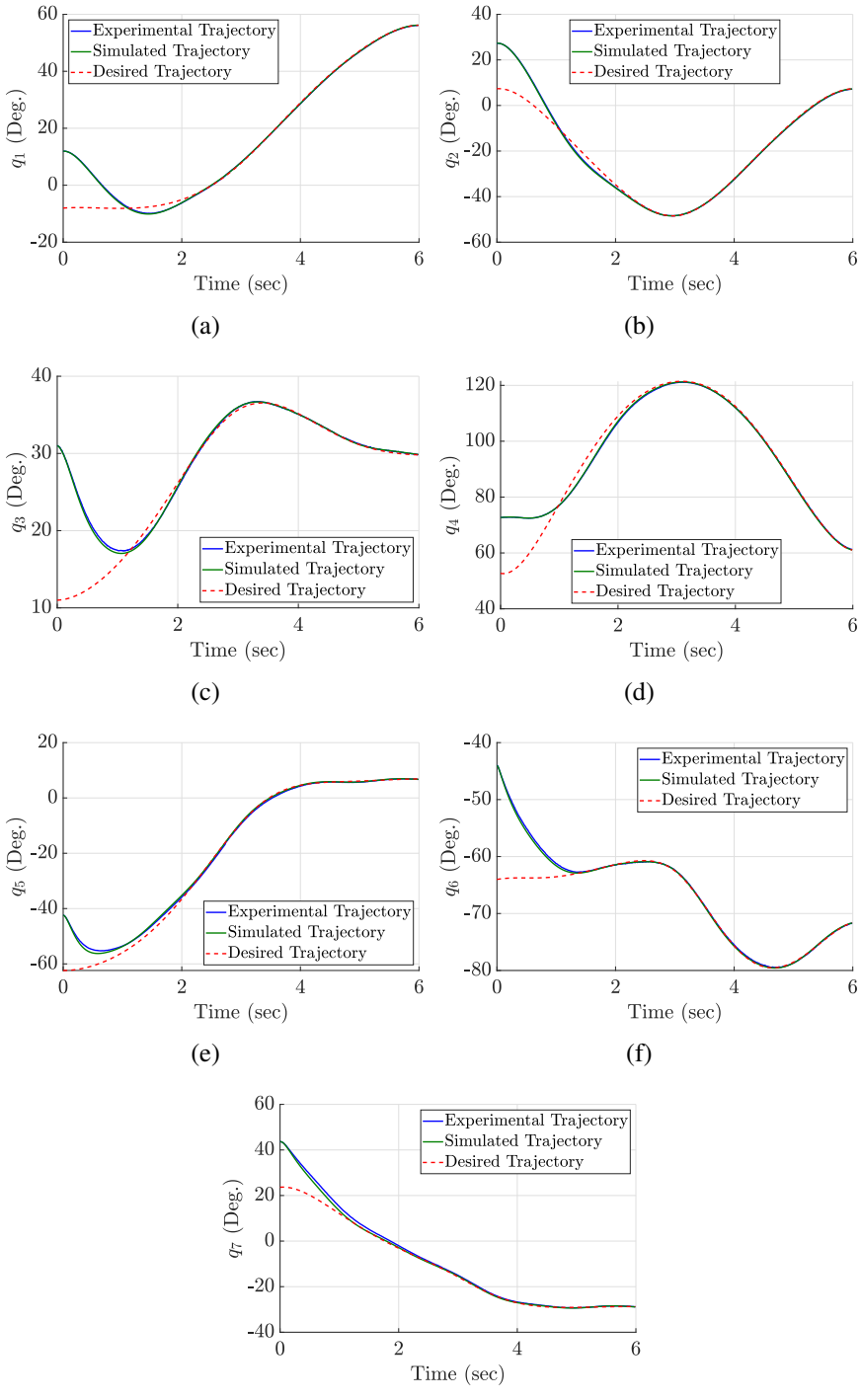


Figure 5.1. The experimental (blue line), simulated (green line), and desired (red dashed line) joint trajectories of Baxter

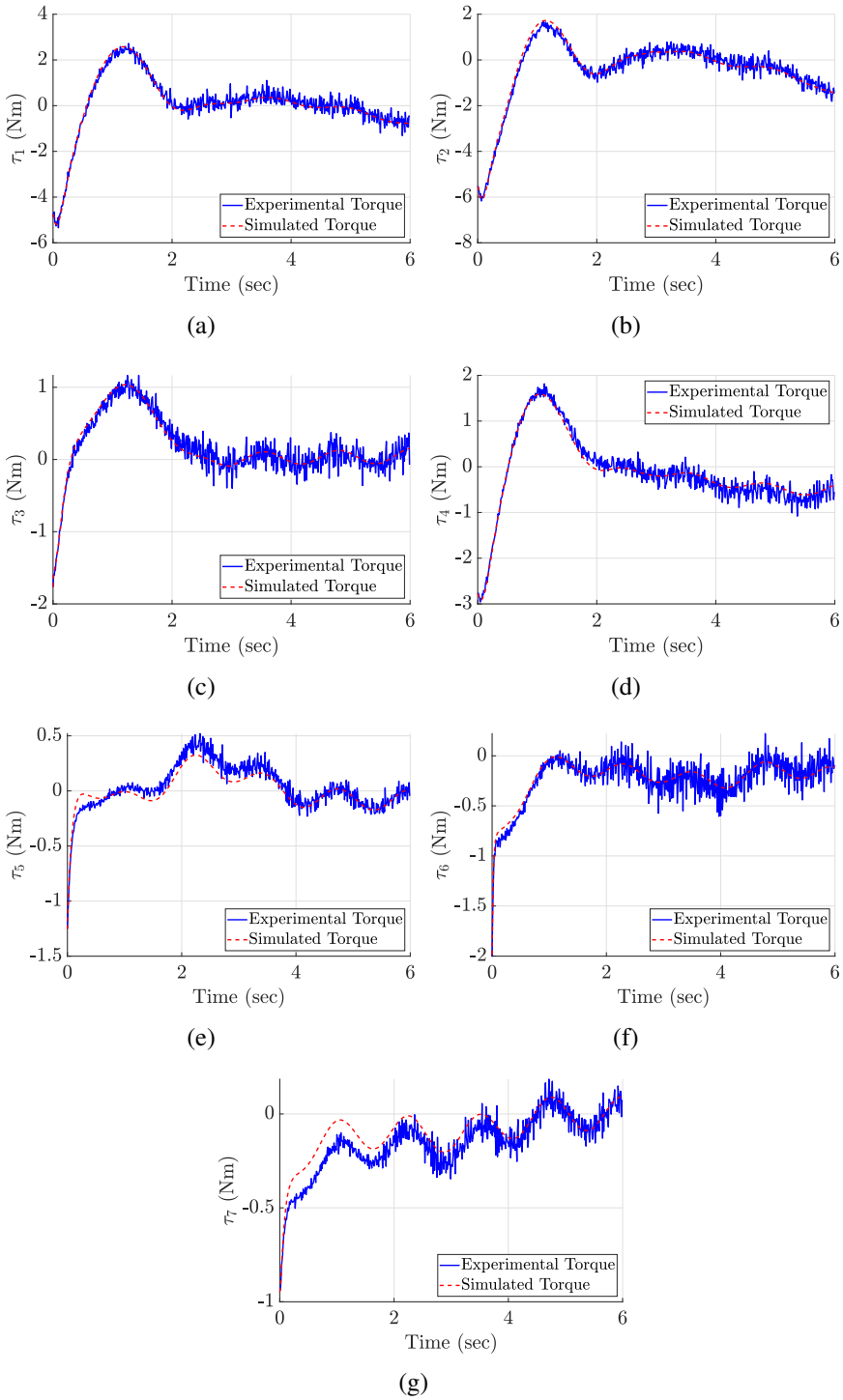
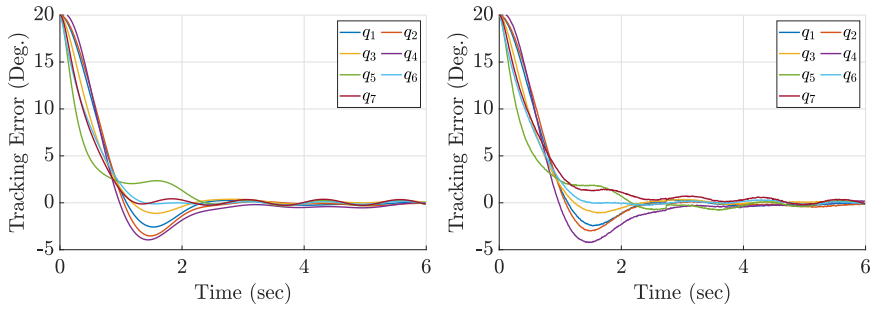


Figure 5.2. The experimental (blue line) and simulated (red dashed line) joint torque input signals of Baxter

Figure 5.3. Baxter tracking a desired trajectory under the prescribed-time control scheme, correcting for a large initial tracking error of 20 degrees in each joint and attenuating a sinusoidal torque disturbance of $D(t) = 0.1\sin(5t)$. The green circles represent the reference trajectory to be tracked, and are spaced at approximately 1 second intervals.

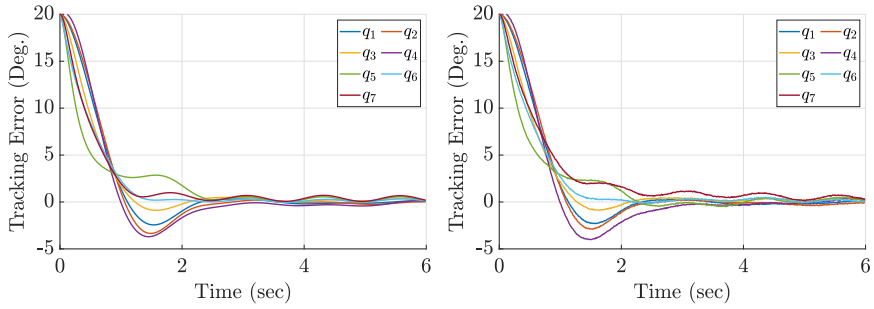
this figure, indicating that the nonlinear-damping method employed was effective at absorbing the effect of the sinusoidal disturbance. Furthermore, minimal overshoot is observed during the 1st 2.5 seconds of operation, indicating that the proposed control law is acting neither too aggressively or too leniently in the beginning of the task. Observing Figure 5.4, it is possible to see the convergence behavior of the proposed method in more detail. After 2.5 seconds of operation, roughly coinciding to the time of $\zeta T = 2.4$ seconds where the gain multiplier μ_1 stops increasing, the majority of the tracking errors have already been significantly attenuated. From 2.5 seconds onward, the residual tracking errors, mostly resulting from the sinusoidal torque disturbance, are attenuated to an acceptably small value of less than 0.2 degrees.

The experimental and simulated joint toque input signals can be seen in Figure 5.2. It is important to note that these torques are significantly lower than the maximum torque output of Baxter's joints, which are 50 Nm for joints 1-4, and 15 Nm for joints 5-7. Thus, the prescribed-



(a) (b)

Figure 5.4. The simulated (a) and experimental (b) joint tracking errors of Baxter, with $D(t) = 0.1\sin(5t)$



(a) (b)

Figure 5.5. The simulated (a) and experimental (b) joint tracking errors of Baxter, when subjected to a disturbance with non-zero mean $D(t) = 0.1\sin(5t) + 0.05$

time approach is able to correct for a large initial error without producing excessive joint torques. Additionally, the simulated torques remain smooth throughout the procedure and do not display chattering, which can negatively affect the lifespan of the actuators used to control the robot manipulator. Furthermore, while the presence of noise in angular velocity measurements has caused similar variations in the experimental joint torques, these torques still exhibit moderate continuity, and do not appear to be affected by chattering. An important observation regarding both the simulated and experimental joint torques is that oscillation can be observed throughout the procedure, which is most noticeable in joint 7. Both the peak-to-peak difference in this observed oscillation, as well as its frequency closely match that of the applied torque disturbance $D(t) = 0.1\sin(5t)$, indicating that the proposed controller is able to "absorb" the disturbance as the prescribed final time of 6 seconds is reached.

In order to verify the ability of the proposed method to reject a torque disturbance with a non-zero mean, an additional simulation and experiment was performed with a torque disturbance of $D(t) = 0.1\sin(5t) + 0.05$. In the interest of brevity, we present only the convergence of the joint tracking errors, which can be seen in Figure 5.5. It can be observed from this figure that both the simulated and experimental tracking errors display nearly identical behavior to Figure 5.4, demonstrating that the convergence of the proposed method is not negatively effected by a disturbance torque with a non-zero mean.

5.6 Conclusion

In this research effort, we formulated and experimentally verified the prescribed-time trajectory tracking control of a 7-DOF robot manipulator. In order to ensure regulation of the tracking errors by the prescribed final time, we employed a scaling of the state by a function of time that grows unbounded towards the terminal time. Through Lyapunov analysis, we demonstrated that the proposed controller achieves regulation of all tracking errors within the prescribed time with a torque that is uniformly bounded, even in the presence of a matched non-vanishing disturbance. Through inspection of the control law, we demonstrated that the choice of parameters for the proposed control law is intuitive and straightforward, and that the controller could be implemented in a practical system with minimal modifications. Then, through both simulation and experiment, we demonstrated that the proposed controller is capable of converging to the desired trajectory within the prescribed time, despite large initial conditions of the tracking errors and a sinusoidal disturbance being applied in each joint.

5.7 Acknowledgements

This Chapter contains adaptations of the following papers: 1) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić."Experimental and Analytical Prescribed-Time Trajectory Tracking Control of a 7-DOF Robot Manipulator." *2022 American Control Conference*

(ACC). IEEE, 2022. 2) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. "Design and Experiment of a Prescribed-Time Trajectory Tracking Controller for a 7-DOF Robot Manipulator." *Journal of Dynamic Systems, Measurement, and Control*. Under Revision, 2022. The dissertation author is the primary investigator and author of these papers.

This chapter is based upon work supported by the National Science Foundation under Award #1823951-1823983. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Chapter 6

Prescribed-Time Safety Filter

In this chapter, we formulate a prescribed-time safety filter for the case of a redundant manipulator performing a fixed-duration task. This formation, which is based on a quadratic programming approach, yields a filter that is capable of avoiding multiple obstacles in a minimally invasive manner with bounded joint torques, while simultaneously allowing the nominal controller to converge to positions located on the boundary of the safe set by the end of the fixed-duration task. In order to demonstrate the efficacy of the proposed method, we performed a series of simulations and experiments on Baxter, a 7-DOF collaborative robot manipulator. In these simulations and experiments, Baxter must follow a six second parabolic trajectory as closely as possible while navigating around a large spherical obstacle blocking its path, and place an object precisely on the surface of a table without overshoot by the end of the six seconds. The results of our simulations and experiments demonstrated the ability of the PTSf to enforce safety throughout the six second task, while allowing the robot manipulator to make contact with the table and thus achieve the desired goal position by the end of the task. Furthermore, when compared to the exponential safety filter, which is the state-of-the-art in current literature, our proposed method yielded consistently lower joint jerks. Thus, for tasks with a fixed duration, the proposed PTSf offers performance benefits over the exponential filters currently present in literature.

6.1 Background

As the usage of robot manipulators in collaborative environments has dramatically risen in recent years, ensuring that a robot manipulator is able to operate safely has become an important goal for modern control systems [108–112]. In this context, safety refers to the ability of a robot manipulator to avoid dangerous collisions, both with humans as well as other potential obstacles. In order to ensure safety during the operation of a collaborative robot manipulator, safety must be considered at every level of the design and operation of the manipulator. From a mechanical perspective, collaborative robot manipulators should be designed to be compliant, so that potential collisions are less damaging. From a planning perspective, the reference trajectories generated for collaborative robot manipulators should be designed in order to avoid collisions with obstacles. From a controls perspective, preventative torques should be applied to the manipulator whenever necessary in order to avoid collisions with obstacles. We focus our efforts towards this controls perspective, and thus on the design of control torques that ensure the robot manipulator remains within a user-defined safe set.

In the past several years, a large amount of research has been devoted towards the design of control barrier functions (CBFs) for robot manipulators [113–129]. CBFs function as a safety filter for a potentially unsafe nominal controller, overriding the nominal control torques when the boundary of the safe set is approached faster than a designed convergence rate. Typically, this override torque is determined via a quadratic program minimizing the difference between the nominal and override torque, and thus CBFs can be characterized as minimally invasive. The majority of CBFs formulated for robot manipulators are based on the concept of exponential safety filters (ESfs), which were first introduced by Nguyen and Sreenath [130]. Using this method, the maximum rate of convergence to the boundary of the safe set is limited to be exponential, and consequently the robot manipulator can approach but will never reach the boundary of the safe set.

While ESfs are designed to be minimally invasive, their presence can interfere with the

operation of set-duration tasks when the goal position of the end-effector is located near the boundary of the safe set. For an example of such a scenario, one could consider a robot that brings a glass of water to a patient's mouth without risking the injury of the patient's teeth. Under ideal conditions such as zero initial tracking error, a trajectory-tracking nominal controller operating in such a scenario would converge to the desired goal position within a fixed time that is governed by the design of the trajectory. However, when an ESf is applied to this nominal controller, the rate of approach to the goal position, which lies on the boundary of the safe set, is limited to be exponential. Thus, the manipulator will not reach the desired goal position by the fixed time, and will instead be located near the desired goal position. After an additional period of time governed by the conservativeness of the ESf, the tracking error will become negligible. Thus, in this circumstance, the ESf has destroyed the prescribed-time convergence property of the nominal controller, and introduced uncertainty into the system as to when the tracking error will become negligible. In circumstances such as the provided example, the additional duration of the task may not be a significant detriment, as the delay in the patient receiving water is unlikely to be life-threatening. However, there are numerous cases in which both safety and timing are critical factors, and thus the limiting behavior of the ESf is undesirable. Consider instead a robot manipulator that is assisting with the surgery of a patient. In this case, safety is desired in order to not further injure the patient or doctors, but timeliness is also critical in order to preserve the life of the patient. In order to address the time-critical aspect of scenarios such as these, a considerable amount of research has been devoted towards the development of control methods for robot manipulators which are capable of guaranteeing an upper bound on the convergence time, achieving convergence of tracking errors to zero within a finite period of time [83–107]. When the nominal controller is capable of ensuring convergence within a finite-time, enforcing a condition of exponential convergence to the boundary of the safe set is counterproductive.

Utilizing concepts from prescribed-time stabilization [95], in which convergence to the desired setpoint is achieved in a time explicitly prescribed as a controller parameter, Abel *et*

al. [131] have recently developed a prescribed time safety filter (PTSf) for a chain of integrators. Rather than enforcing safety for an indefinite period of time, the PTSf enforces safety only for a finite period of time that is explicitly set as a filter parameter T. Notably, this procedure allows the nominal controller to reach the boundary of the safe set by the end of the prescribed duration T. If the boundary of the safe set is approached in this manner, all temporal derivatives of the system state will approach 0 as $t \to T$, meaning the convergence to the barrier will be infinitely soft. In order to achieve this beneficial property, a scaling of the filter gains by a function of time that grows unbounded towards the terminal time is employed. This approach can be interpreted as a safety filter that becomes less strict as the terminal time is approached, allowing the nominal controller to converge to states that are nearby or even located on the boundary of the safe set.

In this chapter, we reformulate the PTSf initially proposed by Abel *et al.* [131] for the case of a redundant manipulator performing a fixed-duration task. This formation yields a filter that is capable of avoiding multiple obstacles in a minimally invasive manner with bounded joint torques, while simultaneously allowing the nominal controller to converge to positions located on the boundary of the safe set by the end of the fixed-duration task. In order to demonstrate the efficacy of the proposed method, we perform a series of simulations and experiments on Baxter, a 7-DOF collaborative robot manipulator. In these simulations and experiments, Baxter must follow a six second parabolic trajectory as closely as possible while navigating around a large spherical obstacle blocking its path, and place an object precisely on the surface of a table without overshoot by the end of the six seconds. To highlight the ability of this method to allow convergence to the barrier within a finite period of time, the nominal controller utilized in both simulation and experiment is the prescribed-time controller which we previously formulated in Chapter 5. The results of our simulations and experiments demonstrate the ability of the PTSf to enforce safety throughout the six second task, while allowing the robot manipulator to make contact with the table and thus achieve the desired goal position by the end of the task. Furthermore, we compare the performance of the PTSf method presented here to an ESf with a high gain, as well as an ESf with a low gain. When performed on the same task, the ESf with a

high gain is able to make contact with the table within the six second task, but exhibits a much higher jerk in the joint torques than the PTSf in the beginning of the task. Conversely, the ESf with a low gain exhibits similar jerk values to the PTSf in the beginning of the task, but is not able to make contact with the table within the six second task. Thus, the PTSf proposed here outperforms the ESf when applied to fixed-duration tasks.

The organization of this chapter is as follows. In Section 6.2, we present the design of the PTSf. In Section 6.3, we mathematically verify the ability of the PTSf to enforce the invariance of the safe set, as well as the feasibility of the proposed filter and boundedness of the filter torques when safety is enforced. In Section 6.4, we present the results of the simulations and experiments performed on Baxter. Finally, in Section 6.5, we present the case that for tasks with a fixed duration, the proposed PTSf offers performance benefits over the exponential filters currently present in literature.

6.2 Prescribed-Time Safety Filter for Robot Manipulators

We consider the following state space representation of (2.1):

$$\dot{Q} = \begin{bmatrix} \dot{Q}_1 \\ \dot{Q}_2 \end{bmatrix} = \begin{bmatrix} Q_2 \\ \nu(t) \end{bmatrix} \tag{6.1}$$

in which we define:

$$Q(t) = \begin{bmatrix} Q_1(t) \\ Q_2(t) \end{bmatrix} = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} \tag{6.2}$$

$$\nu(t) = \ddot{q}(t) = M^{-1}(q)(\tau(t) - C(q,\dot{q})\dot{q}(t) - G(q) - F(\dot{q})) \tag{6.3}$$

The purpose of the proposed PTSF is to ensure that Baxter's end-effector remains within the following user-defined safe set for the duration of the task:

$$\mathscr{H}_p = \{p \in \mathbb{R}^3 | h_i(p) \geq 0, \quad i = 1, ..., m\} \tag{6.4}$$

where

$$\frac{\partial^2 h_i}{\partial p^2} \geq 0, \quad \forall p \in \mathbb{R}^3, \quad i = 1, ..., m \tag{6.5}$$

and $p(q) \in R^3$ is the position of the end-effector in Cartesian coordinates, which is a function of the joint angles q. Thus, we can redefine this safe set in terms of Baxter's joint angles:

$$\mathscr{H} = \{q \in \mathbb{R}^7 | h_i(p(q)) \geq 0, \quad i = 1, ..., m\} \tag{6.6}$$

In this formulation, the robot manipulator must prevent collision between m obstacles, which each has a corresponding CBF h_i. This barrier is positive when there is no collision, 0 when the robot manipulator and the obstacle make contact, and negative when the robot manipulator is within the barrier. Thus, ensuring the joint positions of Baxter are kept within the defined safe set (6.6) is equivalent to preventing a collision between the end-effector and an obstacle. The goal of the PTSf is formally defined as follows:

$$q(t) \in \mathscr{H} \quad \forall t \in [0, T) \tag{6.7}$$

where $T > 0$ is the user-defined duration of the prescribed-time task, as well as the duration of enforcement of the PTSf.

In order to allow the robot manipulator to make contact with the barrier at time T, we employ a scaling of the PTSf gains by a function of time that grows unbounded towards the time

T:

$$\mu_k(t) = \left(\frac{T}{T-t}\right)^k, \quad t \in [0,T), \quad k \in \mathbb{N} \tag{6.8}$$

Note that the temporal derivative of this function can be computed as:

$$\dot{\mu}_k(t) = \frac{k}{T}\mu_{k+1}(t), \quad t \in [0,T), \quad k \in \mathbb{N} \tag{6.9}$$

Due to the relative degree of the CBFs h_i being greater than 1, it is necessary to pursue a backstepping design in order to enforce the invariance of (6.7). To this end, we formulate the following output functions:

$$y_{i1}(t) = h_i(p(q(t))) \tag{6.10}$$

$$y_{i2}(t) = \frac{\partial y_{i1}(q(t))}{\partial q}\dot{q}(t) + c\mu_2(t)y_{i1}(t) \tag{6.11}$$

where $y_{i1}(t), y_{i2}(t) \in \mathbb{R}$ and $c \in \mathbb{R}$ is a design parameter to be determined. In this formulation, if we ensure that y_{i1} and y_{i2} are initially positive and remain positive for the duration of the prescribed-time task, the condition (6.7) will also be satisfied. In order for $y_{i1}(0) > 0$, the system must initially be safe. In order for $y_{i2}(0) > 0$, we must choose:

$$c > \max\left\{0, -\frac{\frac{\partial y_{i1}(0)}{\partial q}\dot{q}(0)}{y_{i1}(0)}\right\} \tag{6.12}$$

In order to ensure that y_{i1} and y_{i2} remain positive in the interval $t \in [0,T)$, it is necessary to examine their underlying dynamics:

$$\dot{y}_{i1} = -c\mu_2 y_{i1} + y_{i2} \tag{6.13}$$

$$\dot{y}_{i2} = \frac{2}{T}c\mu_3 y_{i1} + c\mu_2\frac{\partial y_{i1}}{\partial q}\dot{q} + \dot{q}^\top\frac{\partial^2 y_{i1}}{\partial q^2}\dot{q} + \frac{\partial y_{i1}}{\partial q}\nu \tag{6.14}$$

where (6.13) is obtained from rearranging (6.11), and (6.14) is obtained from taking the temporal derivative of (6.11) and applying (6.9). In order to enforce the positivity of h_1 and h_2 for $t \in [0, T)$, we permit only $\nu(t)$ such that the following condition is satisfied:

$$\dot{y}_{i2} + c\mu_2 y_{i2} \geq 0, \quad i = 1, ..., m \tag{6.15}$$

In the next section, we will show that this is a sufficient condition for the positivity of h_i over the duration $t \in [0, T)$.

Before presenting the design of the PTSf, we first reformulate (6.15) in terms of the CBFs h_i, as well as the joint angles q and joint velocities $\dot{q}$. To this end, we first obtain several derivatives of the CBFs with respect to the joint angles:

$$\frac{\partial h_i}{\partial q} = \frac{\partial h_i}{\partial p} J(q) \tag{6.16}$$

$$\dot{q}^\top \frac{\partial^2 h_i}{\partial q^2} \dot{q} = \dot{q}^\top J^\top(q) \frac{\partial^2 h_i}{\partial p^2} J(q)\dot{q} + \frac{\partial h_i}{\partial p} \dot{J}(q)\dot{q} \tag{6.17}$$

in which

$$J(q) = \frac{\partial p(q)}{\partial q} \tag{6.18}$$

$$\dot{J}(q) = \sum_{i=1}^{7} \frac{\partial J(q)}{\partial q_i} \dot{q}_i \tag{6.19}$$

where $J(q) \in \mathbb{R}^{3\times 7}$ is the jacobian of the end-effector. We make the following assumption of its structure:

Assumption 6.1. *There exists a positive constant $\sigma_{\text{lb}} > 0$ such that the minimum singular value of the end-effector jacobian $J(q(t))$ satisfies the following inequality:*

$$\sigma_{\min}\big(J(q(t))\big) \geq \sigma_{\text{lb}}, \quad \forall t \in [0, T) \tag{6.20}$$

Then, we substitute (6.10), (6.11), (6.14), and (6.16)-(6.18) into (6.15) to obtain:

$$b_i^\top \nu \geq a_i, \quad i = 1, ..., m \tag{6.21}$$

where

$$\begin{aligned} a_i = &-\dot{q}^\top J^\top(q) \frac{\partial^2 h_i}{\partial p^2} J(q)\dot{q} - \frac{\partial h_i}{\partial p}\dot{J}(q)\dot{q} \\ &-c\mu_3 \left(\frac{2}{T} + c\mu_1\right) h_i - 2c\mu_2 \frac{\partial h_i}{\partial p} J(q)\dot{q} \end{aligned} \tag{6.22}$$

$$b_i^\top = \frac{\partial h_i}{\partial p} J(q) \tag{6.23}$$

In order to enforce safety for the prescribed duration T in a minimally invasive manner, we apply quadratic programming minimizing the difference in joint acceleration caused by the filtered and nominal control torque:

$$\nu_{\text{safe}} = \underset{w \in \mathbb{R}^7}{\arg\min} \|w - \nu_{\text{nom}}\|^2 \tag{6.24}$$

$$\text{s.t.} \quad b_i^\top w \geq a_i, \quad i = 1, ..., m \tag{6.25}$$

where

$$\nu_{\text{nom}} = M^{-1}(q)(\tau_{\text{nom}} - C(q, \dot{q})\dot{q}(t) - G(q) - F(\dot{q})) \tag{6.26}$$

and τ_{nom} is the nominal control torque. The filtered control torque can then be determined as:

$$\tau_{\text{safe}} = M(q)\nu_{\text{safe}} + C(q, \dot{q})\dot{q}(t) + G(q) + F(\dot{q}) \tag{6.27}$$

We can now state our main result.

Theorem 6.1. *If $q(0) \in \mathscr{H}$, and the controller parameter c is chosen such that (6.12) is satisfied, then the filtered controller (6.22)-(6.27) applied to the system (2.1) ensures that*

$q(t) \in \mathscr{H}, \forall t \in [0,T)$. *Furthermore, the filtered torque* τ_{safe} *is uniformly bounded provided that the nominal torque* τ_{nom} *is continuous in* t *and Lipschitz in* Q.

6.3 Proof of Theorem 6.1

In order to prove the invariance of the set $\mathscr{H}$ during the interval $t \in [0,T)$, it is first necessary to show that the linear inequalities (6.25) always have a jointly feasible solution. To this end, we construct the following feasible solution:

$$v_{\text{feasible}} = -2c\mu_2\dot{q} - J^+(q)\dot{J}(q)\dot{q} \tag{6.28}$$

where $J^+(q)$ is the Moore-Penrose pseudoinverse of $J(q)$. Note that due to Assumption 6.1, $J(q)$ is non-singular and thus $J(q)J^+(q) = I$. Substituting (6.28) into the condition (6.25) yields:

$$0 \geq -\dot{q}^\top J^\top(q)\frac{\partial^2 h_i}{\partial p^2}J(q)\dot{q} - c\mu_3\left(\frac{2}{T} + c\mu_1\right)h_i \tag{6.29}$$

Utilizing the property (6.5), we can further simplify this inequality:

$$0 \geq -c\mu_3\left(\frac{2}{T} + c\mu_1\right)h_i \tag{6.30}$$

and thus we determine that for $q(t) \in \mathscr{H}$, (6.25) has a jointly feasible solution.

Next, we show that $q(0) \in \mathscr{H}$ ensures that $q(t) \in \mathscr{H}, \forall t \in [0,T)$. Through the application of the Comparison Lemma to (6.14), we obtain the following inequality:

$$y_{i2}(t) \geq y_{i2}(0)e^{cT(1-\mu_1(t))} > 0, \quad \forall t \in [0,T) \tag{6.31}$$

Integrating (6.13) from 0 to t, and substituting this inequality yields:

$$y_{i1}(t) = y_{i1}(0)e^{cT(1-\mu_1(t))} + \int_0^t e^{cT(\mu_1(s)-\mu_1(t))} y_{i2}(s)ds > y_{i1}(0)e^{cT(1-\mu_1(t))} > 0, \quad \forall t \in [0,T) \tag{6.32}$$

Through applying the relationship (6.10), we obtain:

$$h_i(t) > 0, \quad \forall t \in [0,T) \tag{6.33}$$

and thus $q(t) \in \mathscr{H}, \forall t \in [0,T)$.

We now pursue the uniform boundedness of the filtered control law (6.22)-(6.27). We partition the time horizon $[0,T)$ into separate intervals based on which CBFs h_i are active at time t. To this end, we define at time t the set of active constraints $\mathscr{A}(t)$ as:

$$\mathscr{A}(t) = \{i \in 1,...,m | b_i^\top(t)\nu_{\text{safe}}(t) = a_i(t)\} \tag{6.34}$$

where $a_i(t)$ and $b_i^\top(t)$ are defined in (6.22) and (6.23) respectively. Then, we define the partition times t_k:

$$t_k := \begin{cases} \min\{t_{k-1} < t \leq T : \mathscr{A}(t_k) \neq \mathscr{A}(t_{k-1})\}, & \text{if it exists} \\ T, & \text{otherwise} \end{cases} \tag{6.35}$$

for $k \in \mathbb{N}$ with $t_0 = 0$ where

$$[0,T) = \bigcup_{\substack{k \in \mathbb{N} \cup \{0\} \\ t_{k+1} \leq T}} [t_k, t_{k+1}) \tag{6.36}$$

We have constructed this partition such that the filtered control law (6.22)-(6.27) remains continuous at t_k, precluding Zeno behavior of the closed-loop system.

We now examine the behavior of ν_{safe} in the partition $[t_k, t_{k+1})$. If we have the condition $\mathscr{A}(t_k) = \emptyset$, then no constraints are active during this interval, and thus we have $\nu_{\text{safe}}(t) = \nu_{\text{nom}}(t)$

which is uniformly bounded.

If we have the condition $\mathscr{A}(t_k) \neq \emptyset$, then at least one constraint is active during the interval $[t_k, t_{k+1})$ and we do not necessarily have $v_{\text{safe}} = v_{\text{nom}}$. In order to determine a bound for v_{safe} in this case, we first investigate the boundedness of the active CBFs h_i where $i \in \mathscr{A}(t_k)$. For each active barrier function, the output functions $y_{i1}(t)$ and $y_{i2}(t)$ satisfy the following differential equations:

$$\dot{y}_{i1}(t) = -c\mu_2(t)y_{i1}(t) + y_{i2}(t) \tag{6.37}$$

$$\dot{y}_{i2}(t) = -c\mu_2(t)y_{i2}(t) \tag{6.38}$$

Integrating these equations from t_k to t_{k+1} yields:

$$y_{i1}(t) = \left(y_{i2}(t_k) + ty_{i2}(t_k)\right) e^{cT(\mu_1(t_k)-\mu_1(t))} \tag{6.39}$$

$$y_{i2}(t) = y_{i2}(t_k)e^{cT(\mu_1(t_k)-\mu_1(t))} \tag{6.40}$$

Then, (6.39) can be substituted into (6.10) to obtain an expression for $h_i(t)$, and can be differentiated with respect to t to obtain an expression for $\dot{h}_i(t)$:

$$h_i(t) = \left(y_{i2}(t_k) + ty_{i2}(t_k)\right) e^{cT(\mu_1(t_k)-\mu_1(t))} \tag{6.41}$$

$$\dot{h}_i(t) = -c\mu_2(t)\left(y_{i2}(t_k) + ty_{i2}(t_k)\right) e^{cT(\mu_1(t_k)-\mu_1(t))} + y_{i2}(t_k)e^{cT(\mu_1(t_k)-\mu_1(t))} \tag{6.42}$$

Due to the negative dominating behavior of the exponentials in (6.41) and (6.42), it can be seen that $h_i(t)$ and $\dot{h}_i(t)$ remain bounded within the interval $[t_k, t_{k+1})$. Furthermore, if $t_{k+1} = T$, we can utilize l'Hôpital's rule to determine the behavior of the active CBFs as the terminal time is approached:

$$\lim_{t\to T^-} h_i(t) = \lim_{t\to T^-} \dot{h}_i(t) = 0, \quad \forall i \in \mathscr{A}(t_k) \tag{6.43}$$

Thus, the active CBFs remain bounded in the interval $[t_k, t_{k+1})$, and approach 0 if $t_{k+1} = T$.

Utilizing the boundedness of $\dot{h}_i(t)$, we now investigate the boundedness of q and $\dot{q}$ within the interval $[t_k, t_{k+1})$. Differentiating the expression $h_i(p(q(t)))$ yields:

$$\dot{h}_i = \frac{\partial q}{\partial p} J(q)\dot{q} = b_i^\top \dot{q} \tag{6.44}$$

where $b_i^\top$ is initially defined in (6.23). Thus, the component of $\dot{q}$ that is parallel to $b_i^\top$ is bounded, and approaches 0 if $t_{k+1} = T$. As this property holds for each $b_i^\top$ where $i \in \mathscr{A}(t_k)$, we deduce the following properties of $\dot{q}$:

$$|B\dot{q}| < \infty, \quad \forall t \in [t_k, t_{k+1}) \tag{6.45}$$

$$\lim_{t \to T^-} B\dot{q} = 0, \quad \text{if} \quad t_{k+1} = T \tag{6.46}$$

where B is a projection matrix to the minimum subspace spanned by the vectors $b_i^\top$ where $i \in \mathscr{A}(t_k)$. Using this projection matrix, we separate $\dot{q}$ into the components that lie within and outside of this minimum subspace:

$$\dot{q} = \dot{q}_\parallel + \dot{q}_\perp \tag{6.47}$$

where

$$\dot{q}_\parallel = B\dot{q} \tag{6.48}$$

$$\dot{q}_\perp = (I - B)\dot{q} \tag{6.49}$$

From integrating $\dot{q}_\parallel$, we can additionally obtain the boundedness of $q_\parallel = Bq$.

Next, we obtain the boundedness of $q_\perp$ and $\dot{q}_\perp$. We make use of the fact that since ν_{safe} is obtained via quadratic programming, the difference between the safe and nominal control $\nu_{\text{safe}} - \nu_{\text{nom}}$ lies in the subspace spanned by B, and thus we have $\nu_{\text{safe},\perp} = \nu_{\text{nom},\perp}$. This property can be verified by contradiction, as for every feasible solution where this property does not hold, a feasible solution with a lower value of the objective function $\|w - \nu_{\text{nom}}\|^2$ can be obtained via

eliminating the perpendicular component of $w - v_{\text{nom}}$. Thus, $q_\perp$ and $\dot{q}_\perp$ satisfy the following differential equations:

$$\dot{q}_\perp = q_\perp \tag{6.50}$$

$$\ddot{q}_\perp = v_{\text{nom},\perp} \tag{6.51}$$

As v_{nom} is continuous in t and Lipschitz in Q, and $q_\parallel$ and $\dot{q}_\parallel$ are bounded, these differential equations can be integrated from t_k to t_{k+1} to simultaneously obtain the boundedness of v_{nom}, $q_\perp$ and $\dot{q}_\perp$, and consequently q and $\dot{q}$.

In order to determine a bound for v_{safe}, we first investigate the boundedness of the following feasible solution:

$$v_{\text{alt}} = -2c\mu_2\dot{q}_\parallel - J^+(q)\dot{J}(q)\dot{q} \tag{6.52}$$

in which we have replaced $\dot{q}$ in the 1st term of (6.28) with $\dot{q}_\parallel$, noting that this does not alter the feasibility of the solution. Through the application of Assumption 6.1, it can be observed that the second term in (6.52) is bounded in the interval $[t_k, t_{k+1})$. To bound the 1st term in (6.52), we utilize l'Hôpital's rule to obtain the following limit:

$$\lim_{t \to T^-} \mu_2(t)\dot{h}_i(t) = 0, \quad \forall i \in \mathscr{A}(t_k) \tag{6.53}$$

Applying this limit and (6.44) to the 1st term in (6.52), it can be observed that this term and consequently v_{alt} are bounded within the interval $[t_k, t_{k+1})$.Then, it can be seen from (6.24) that we must also have:

$$v_{\text{safe}} \leq \|v_{\text{alt}} - v_{\text{nom}}\|^2 \tag{6.54}$$

and thus v_{safe} is bounded in the interval $[t_k, t_{k+1})$. As we have determined that v_{safe} is bounded in each interval $[t_k, t_{k+1})$, we conclude that v_{safe} and consequently τ_{safe} are uniformly bound in

the interval $[0, T)$.

6.4 Simulated and Experimental Results

In order to assess the performance of the proposed PTSf approach, we perform both simulations using ODE methods on Baxter's dynamic equation (2.1), as well as experiments. In both the simulation and experiment, Baxter must track the six second trajectory specified in Chapter 2, while simultaneously avoiding collision with a large spherical obstacle blocking the trajectory, and placing its held object precisely on the surface of a table. To highlight the ability of this method to allow convergence to the barrier within a finite period of time, the nominal controller utilized in both simulations and experiments is a prescribed-time controller which we previously formulated in Chapter 5. As we demonstrated in the previous chapter, this prescribed-time nominal controller is capable of achieving zero tracking errors by the end of the six second desired trajectory. Thus, our simulations and experiments will serve to demonstrate the ability of the proposed PTSf method to allow convergence to the barrier within a finite period of time. Furthermore, we compare the performance of the PTSf method presented here to an ESf with a high gain, as well as an ESf with a low gain to highlight the strengths of the proposed method.

In the simulations and experiments, the CBF preventing collision with the spherical obstacle is formulated as:

$$h_1(p(q)) = \left\| p(q) - p_{\text{sphere}} \right\|_2^2 - R_{\text{sphere}}^2 \tag{6.55}$$

where $p_{\text{sphere}} \in \mathbb{R}^3$ is the position of the sphere, and $R_{\text{sphere}} \in \mathbb{R}$ is the minimum safe distance between the robot manipulator and the center of the obstacle. We formulate the CBF preventing collision with the table as:

$$h_2(p(q)) = p(q) \cdot (0,0,1) - z_{\text{table}} \tag{6.56}$$

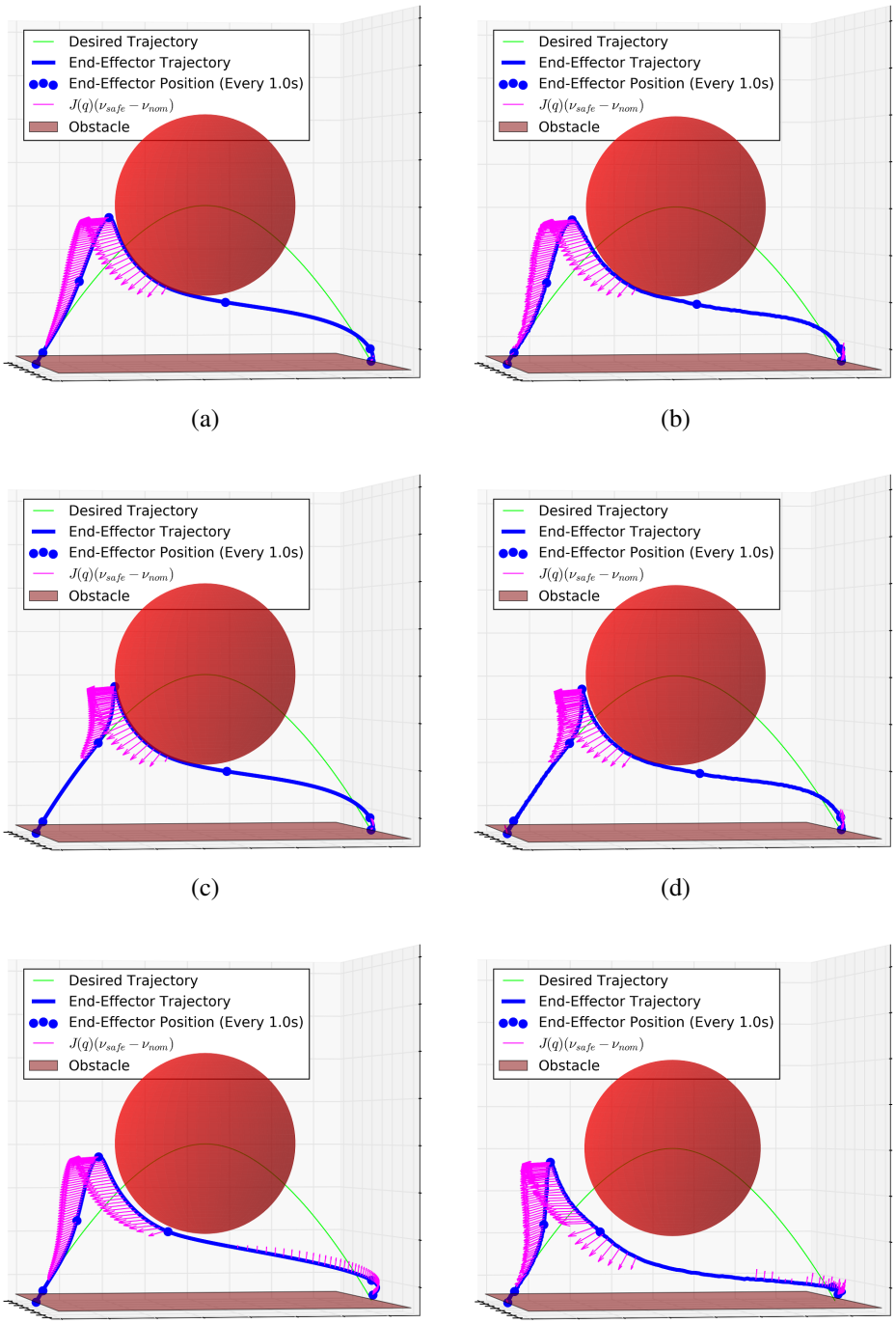


Figure 6.1. Simulations (left column) and experiments (right column) of Baxter following a pick-and-place trajectory while avoiding multiple obstacles, using a prescribed-time safety filter (a, b), an exponential safety filter with a high gain of $\rho = 4$ (c, d), and an exponential safety filter with a low gain of $\rho = 1.5$ (e, f). At $t = 3s$, the end-effector trajectory takes a major turn from moving up to moving below the spherical obstacle.

where $z_{\text{table}} \in \mathbb{R}$ is the height of the table. For the PTSf, we set the controller parameter as $c = 1.2$. In order to prevent numerical issues arising from employing an unbounded scaling of the gain μ_2, we clip this scaling at a maximum value $\mu_{2,\max} = 6.25$. For our six second task, this maximum is reached after 3.6 seconds of operation. For the nominal controller, we utilize the same controller parameters as in [132], so that the interested reader can compare the performance of the PTSf + nominal controller to that of the nominal controller alone.

In order to convert our formulation of a PTSf to that of an ESf, the following expression can be utilized in substitute of (6.22):

$$a_{i,\text{esf}} = -\dot{q}^\top J^\top(q)\frac{\partial^2 h_i}{\partial p^2}J(q)\dot{q} - \frac{\partial h_i}{\partial p}\dot{J}(q)\dot{q} - 2\rho^2 h_i - 3\rho\frac{\partial h_i}{\partial p}J(q)\dot{q} \tag{6.57}$$

with $\rho > 0$. For our high gain ESf, we set $\rho = 4$ so that the high gain ESf + nominal controller can achieve negligible tracking error at the end of the six second task. For our low gain ESf, we instead set $\rho = 1.5$ so that the low gain ESf begins to take evasive action at the same instance of time as the proposed PTSf.

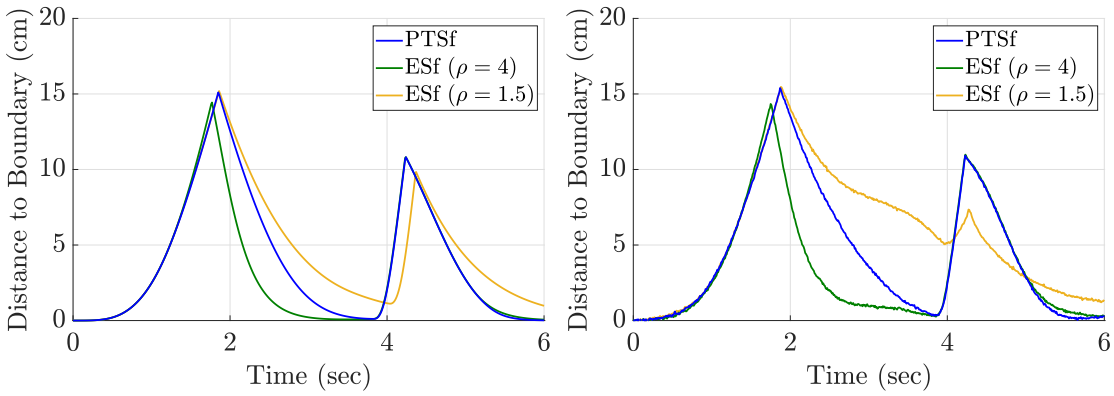


Figure 6.2. The simulated (a) and experimental (b) distance between the robot manipulator and the nearest obstacle.

From Figure 6.1 it can be observed that the proposed PTSf successfully avoids the spherical obstacle in both simulation and experiment, while simultaneously placing the held

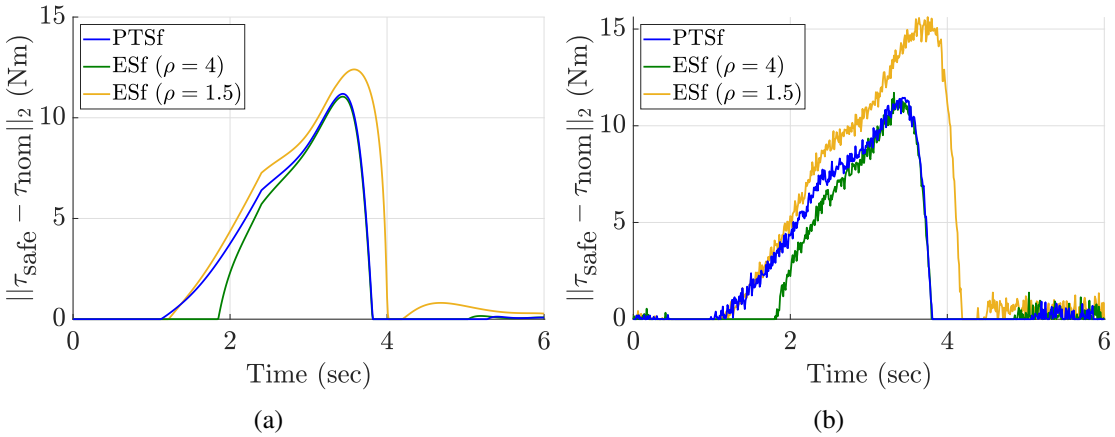


Figure 6.3. The magnitude of the difference between the nominal and filtered joint torques in simulation (a) and experiment (b).

object precisely on the table at the end of the six second task. Please check the DSCL YouTube Channel, at https://youtu.be/yRr6D2oFSeQ, for the video of Baxter performing the experiment. Furthermore, the magnitude of the torque applied by the PTSf gradually increases as the spherical obstacle is approached, indicating that the control action is smooth and that the magnitude of the joint jerks of Baxter are not large. In comparison, the ESf with a high gain is also successful at placing the held object precisely on the table. However, the magnitude of the torque applied by the PTSf increases much more rapidly as the obstacle is approached, indicating a sharper discontinuity in the control action as well as higher joint jerks. Conversely, the ESf with a low gain appears to have joint jerks with a similar magnitude as that of the proposed PTSf, but is unable to achieve zero tracking error by the end of the six second task. Unlike both the PTSf as well as the ESf with a high gain, the ESf with a low gain becomes active towards the end of the task, limiting the rate of approach of the table to a slow exponential approach, rather than the prescribed-time approach governed by the nominal controller. The distance between the end-effector and the nearest obstacle in both simulation and experiment can be more closely observed in Figure 6.2. While the PTSf and high gain ESf both are able to place the held object precisely on the table, the low gain ESf instead holds the object roughly 1 cm above the table by the end of the six second task. Additionally, it is notable to mention that the ESf with a low

gain appears to behave much more conservatively in experiment than in simulation, maintaining a large distance between the end-effector and the spherical obstacle. This is likely due to a discrepancy between the modeled and actual friction in each joint of Baxter, and is more apparent when the filter gains are not large enough to counteract their contribution.

In Figure 6.3, the magnitude of the difference between the nominal and filtered joint torques, $\|\tau_{\text{safe}} - \tau_{\text{nom}}\|_2$, can be seen. It can be observed from this figure that the PTSf and the low gain ESf both become active after around one second of operation, whereas the high gain ESf becomes active after around two seconds of operation. In order to avoid collision with the spherical obstacle while reacting at a later time, the high gain ESf ramps up in magnitude much faster than the PTSf and low gain ESf, indicating larger joint jerks during the operation of the robot manipulator at this time. After around four seconds of operation, each safety filter rapidly drops in magnitude. This period in time corresponds with the end-effector quickly passing under the spherical obstacle, meaning that the obstacle is no longer blocking the end-effector from approaching the reference trajectory. Thus, this large change in magnitude of each safety filter is primarily due to the shape of the obstacle CBF, as well as the nominal controller rather than the convergence properties of the utilized safety filter. It is important to mention that at this instant, the safety filter with the highest experienced joint jerks is the low gain ESf. This is due to the conservative low gains of the ESf keeping the end-effector further from the reference trajectory in the beginning of the task, resulting in a larger nominal control torque to drive the system back towards the reference trajectory. During the last 2 seconds of the task, the low gain ESf maintains operation with a small magnitude, while both the high gain ESf and the PTSf do not noticably interfere with the motion of the end-effector at this time.

The experimental, simulated, and desired joint trajectories of Baxter can be seen in Figure 6.4. In order to avoid the large spherical obstacle, joints 1 and 3 experience large deviations from the desired trajectory, with the low gain ESf experiencing the largest tracking errors. After this large deviation, the joint trajectories smoothly converge back to the desired trajectory. Observing Figure 6.7, it is possible to see the convergence behavior of each method in more detail. While

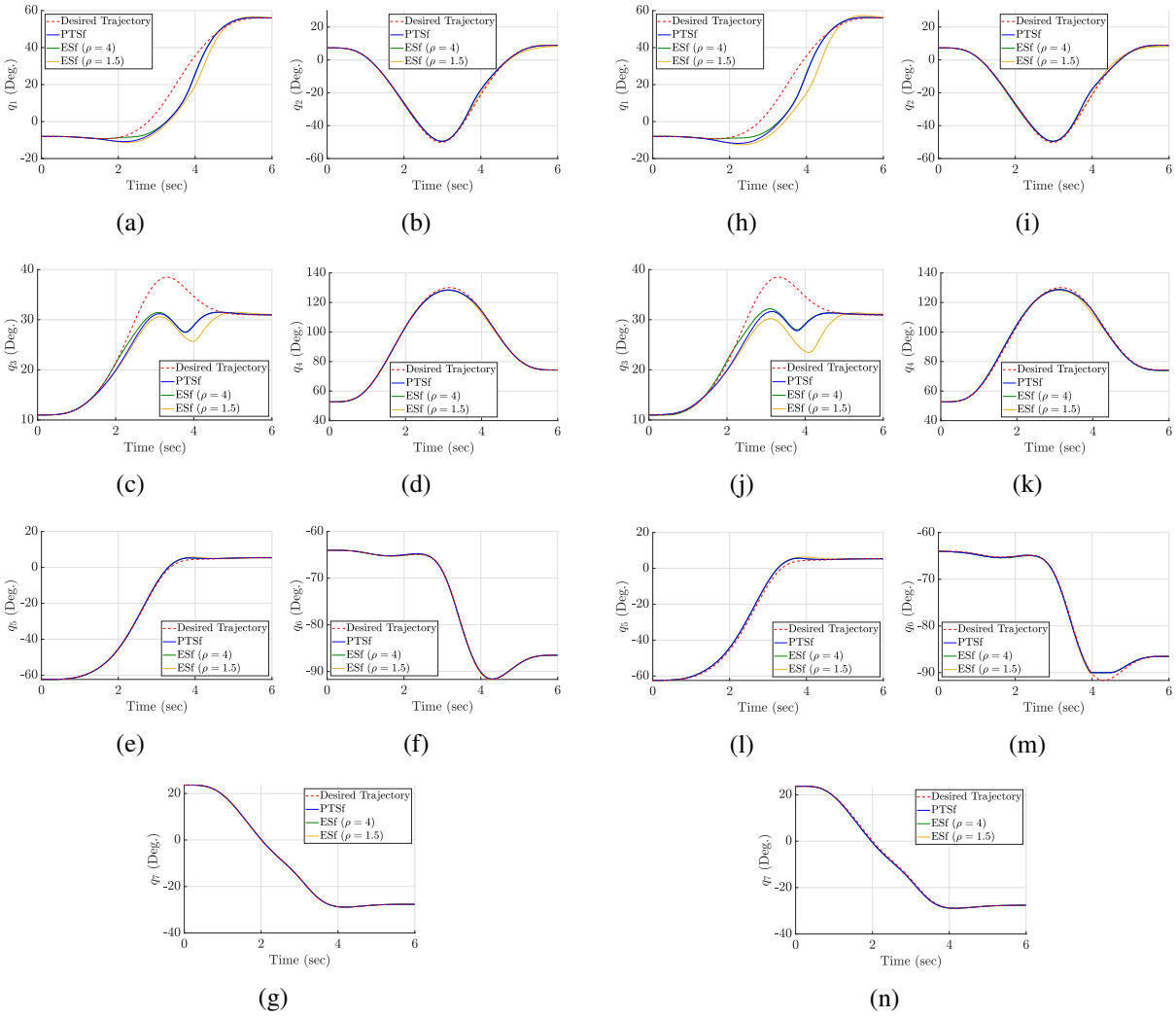


Figure 6.4. The simulated (a-g) and experimental (h-n) joint trajectories of Baxter. At $t = 3s$, the end-effector trajectory takes a major turn from moving up to moving below the spherical obstacle.

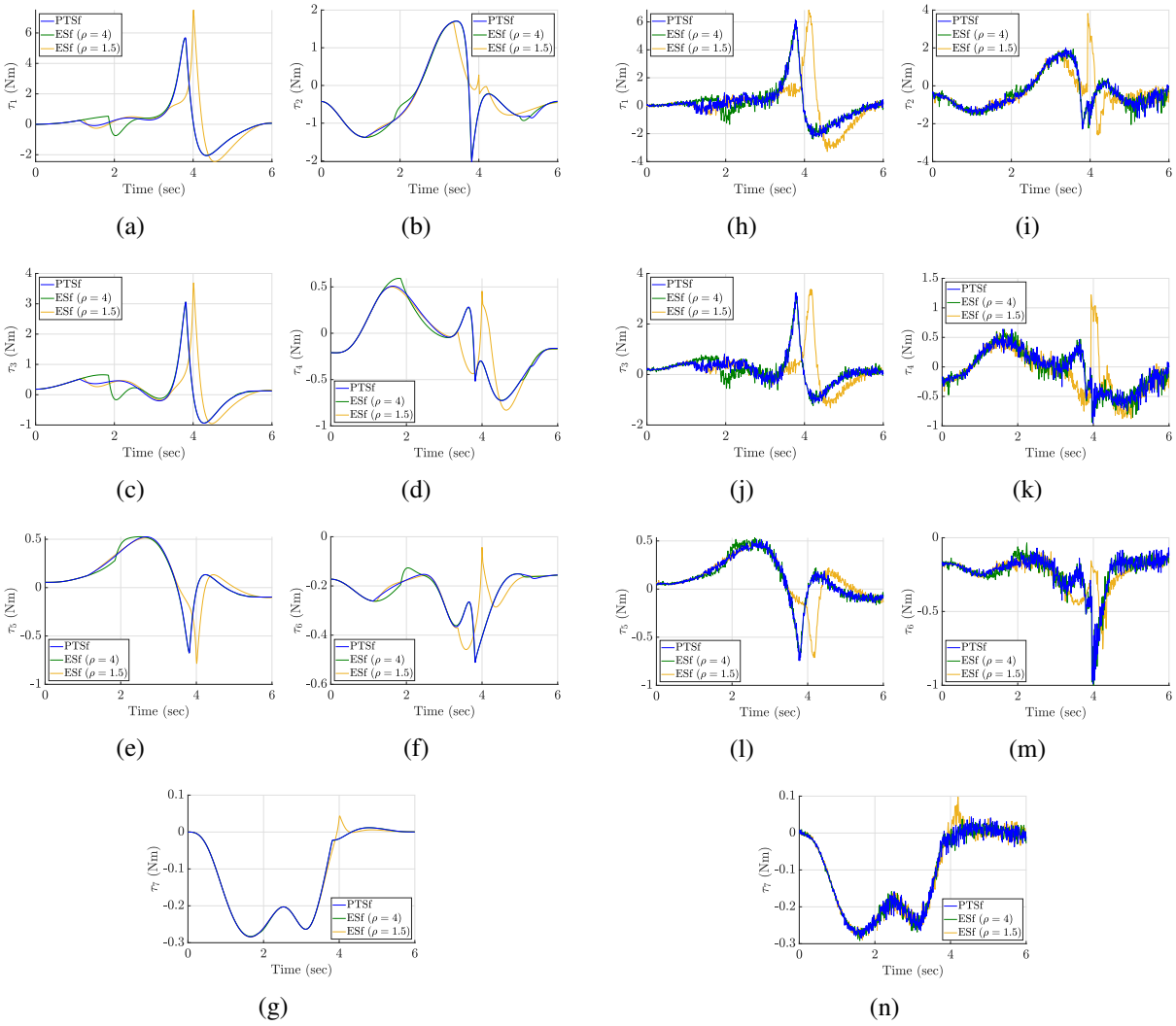


Figure 6.5. The simulated (a-g) and experimental (h-n) joint torque input signals of Baxter. At $t = 3s$, the end-effector trajectory takes a major turn from moving up to moving below the spherical obstacle.

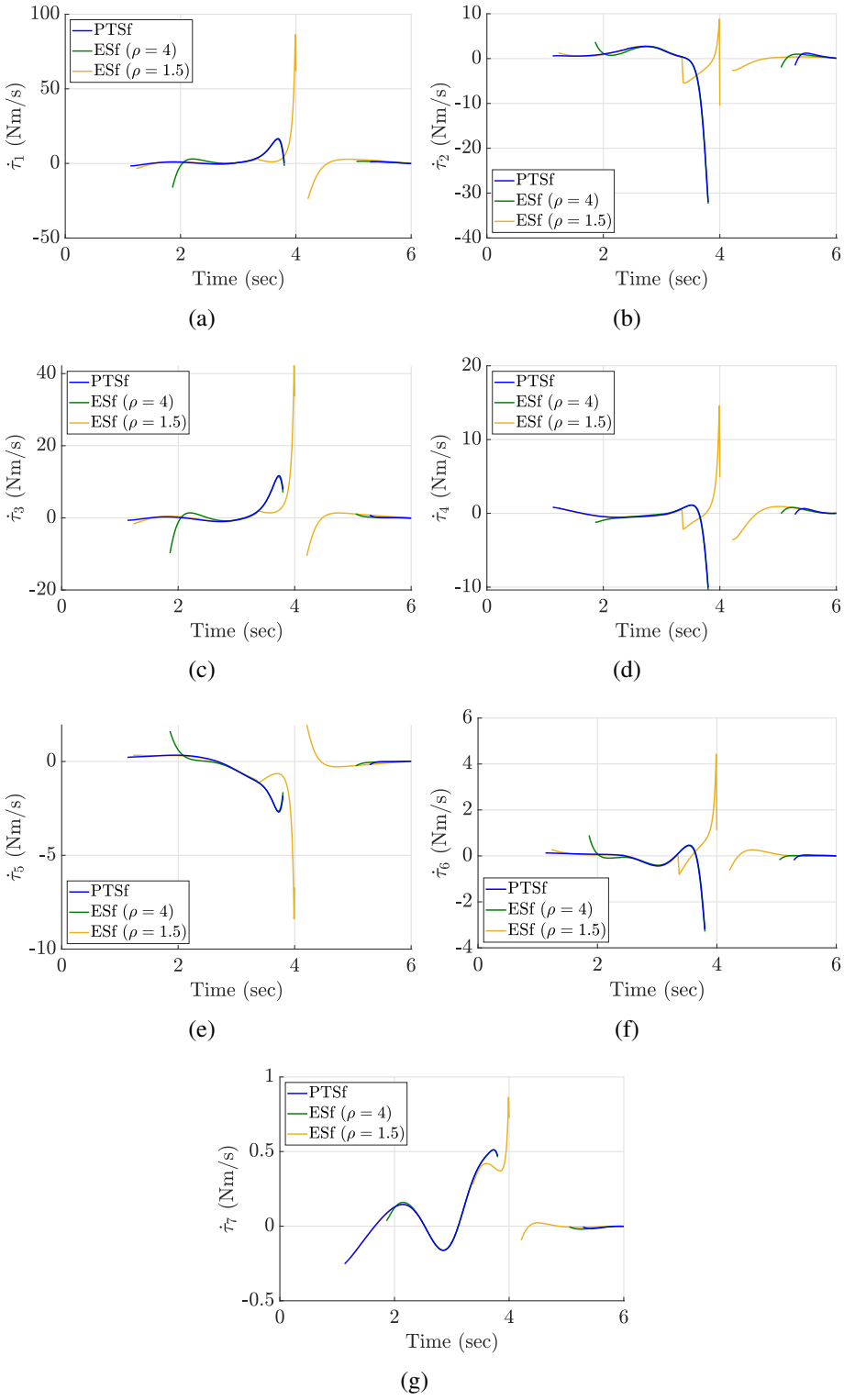


Figure 6.6. The simulated joint jerks of Baxter, shown when the safety filter is active. At $t = 3s$, the end-effector trajectory takes a major turn from moving up to moving below the spherical obstacle.

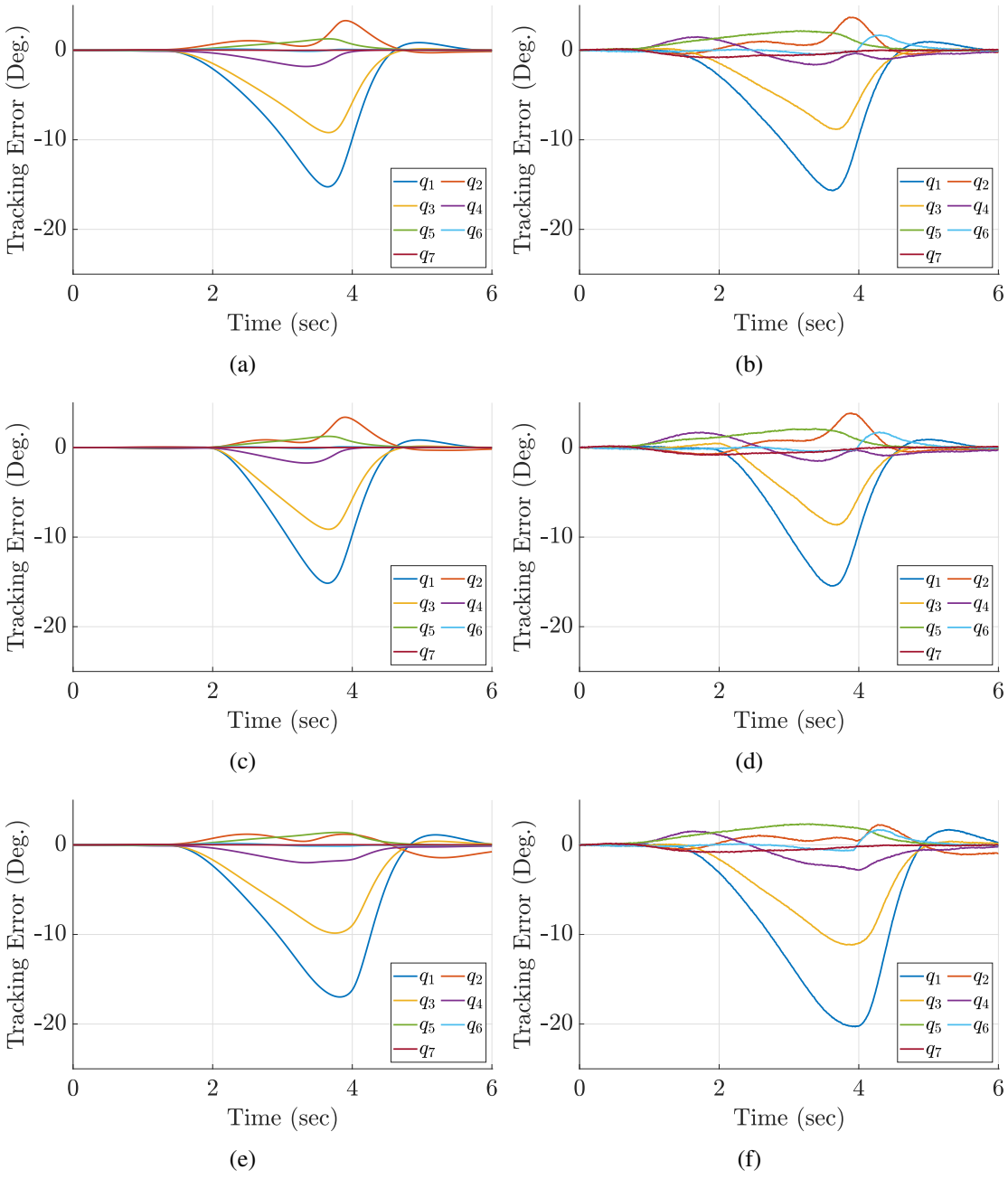


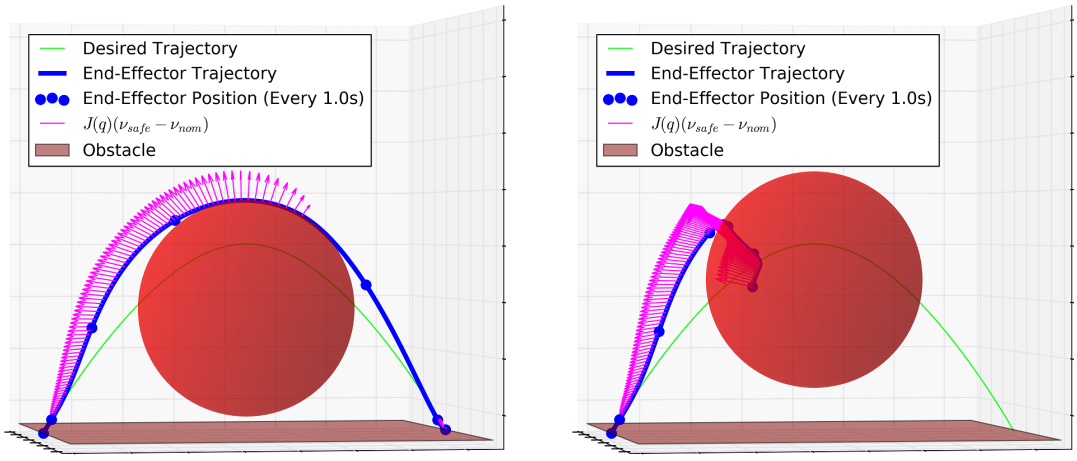
Figure 6.7. Simulated (left column) and experimental (right column) tracking errors of Baxter when using a prescribed-time safety filter (a, b), an exponential safety filter with a high gain of $\rho = 4$ (c, d), and an exponential safety filter with a low gain of $\rho = 1.5$ (e, f).

the tracking errors for the PTSf and high gain ESf reach negligible values, there is a residual tracking error of roughly one degree on joint 2 for the low gain ESf. This joint is primarily responsible for the height of the end-effector, and thus this tracking error is present due to the low gain ESf limiting the rate of approach to the table.

The experimental and simulated joint toque input signals of Baxter can be seen in Figure 6.5. It is important to note that these torques are significantly lower than the maximum torque output of Baxter's joints, which are 50 Nm for joints 1-4, and 15 Nm for joints 5-7. Thus, none of the tested methods pose the risk of torque saturation. Furthermore, while the presence of noise in angular velocity measurements has caused similar variations in the experimental joint torques, these torques still exhibit moderate continuity, and do not appear to be affected by chattering. There does not appear to be a large difference between the tested methods in regards to their susceptibility to noise, as each curve appears to have a similar degree of "fuzziness".

The simulated joint jerks of Baxter can be seen in Figure 6.6. In the beginning of the task, the joint jerk from the high gain ESf is an order of magnitude larger than either the PTSf or the low gain ESf. As the task progresses, the jerk from the high gain ESf becomes nearly identical to that of the PTSf, due to the end-effector following along the surface of the spherical obstacle. At the end of the task, both the PTSf and the high gain ESf have a negligible joint jerk compared to the low gain ESf, which actively limits the rate of approach of the end-effector to the surface of the table. Across the duration of the task, the PTSf consistently achieves the smallest joint jerks out of the tested methods, only increasing in magnitude due to the influence of the shape of the obstacle and the trajectory tracking task.

It is important to note that the path of the end-effector when utilizing a PTSf, as with CBF approaches in general, depends on the structure of the barrier functions h_i. For example in Figure 6.8(a), lowering the center of the spherical obstacle by 120mm causes the end-effector trajectory to go over the spherical obstacle rather than below it. Furthermore, as the quadratic programming filter strategy (6.24), (6.25) is a local optimization scheme, it is possible for the end-effector to get stuck on an obstacle even when there is a valid path back to the reference



(a) (b)

Figure 6.8. Simulations of Baxter following a pick-and-place trajectory while avoiding multiple obstacles, using a prescribed-time safety filter. In (a), the center of the spherical obstacle is lowered 120mm, resulting in the end-effector going over the spherical obstacle. In (b), the center of the spherical obstacle is lowered exactly 67 mm, resulting in the end-effector being unable to reach its destination. Note that in this case, the end-effector still travels along the surface of the spherical obstacle without exiting the safe set.

trajectory. In Figure 6.8(b), by lowering the spherical obstacle precisely 67 mm, the end-effector is no longer able to return to the reference trajectory. Even in this case however, the end-effector does not violate the safe set, and instead gently comes into contact with the spherical obstacle at the end of the six second task. The primary purpose of our proposed PTSf, as well as ESfs and other CBF based approaches is to ensure the system does not leave the safe set in a minimally invasive manner. In the context of safety, these approaches should not be seen as a substitute for path-planning, but instead as an additional layer of safety, ensuring the system remains safe even when the system does not perfectly follow the reference trajectory, or if the reference trajectory is not suitably designed to prevent collision with obstacles.

6.5 Conclusion

In this chapter, we reformulated the PTSf initially proposed by Abel *et al.* [131] for the case of a redundant manipulator performing a fixed-duration task. This formation yields a filter

that is capable of avoiding multiple obstacles in a minimally invasive manner with bounded joint torques, while simultaneously allowing the nominal controller to converge to positions located on the boundary of the safe set by the end of the fixed-duration task. In order to demonstrate the efficacy of the proposed method, we performed a series of simulations and experiments on Baxter, a 7-DOF collaborative robot manipulator. In these simulations and experiments, Baxter must follow a six second parabolic trajectory as closely as possible while navigating around a large spherical obstacle blocking its path, and place an object precisely on the surface of a table without overshoot by the end of the six seconds. To highlight the ability of this method to allow convergence to the barrier within a finite period of time, the nominal controller utilized in both simulation and experiment is a prescribed-time controller which we previously formulated in Chapter 5. The results of our simulations and experiments demonstrated the ability of the PTSf to enforce safety throughout the six second task, while allowing the robot manipulator to make contact with the table and thus achieve the desired goal position by the end of the task. Furthermore, when compared to the ESf, which is the state-of-the-art in current literature, our proposed method yielded consistently lower joint jerks. Thus, for tasks with a fixed duration, the proposed PTSf offers performance benefits over the exponential filters currently present in literature.

6.6 Acknowledgements

This chapter contains adaptations of the following papers: 1) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. ”Experiment and Design of Prescribed-Time Safety Filter for a 7-DOF Robot Manipulator Using CBF-QP.” *Modeling, Estimation and Control Conference*. Under Review, 2022. 2) Bertino, Alexander, Peiman Naseradinmousavi, and Miroslav Krstić. ”Prescribed-Time Safety Filter for a 7-DOF Robot Manipulator: Experiment and Design.” *IEEE Transactions on Control Systems Technology*. Under Review, 2022. The dissertation author is the primary investigator and author of these papers.

This chapter is based upon work supported by the National Science Foundation under Award #1823951-1823983. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Bibliography

[1] M. Bagheri and P. Naseradinmousavi, "Novel analytical and experimental trajectory optimization of a 7-dof baxter robot: global design sensitivity and step size analyses," *The International Journal of Advanced Manufacturing Technology*, vol. 93, no. 9-12, pp. 4153–4167, December 2017.

[2] M. Bagheri, P. Naseradinmousavi, and R. Morsi, "Experimental and novel analytical trajectory optimization of a 7-dof baxter robot: Global design sensitivity and step size analyses," in *ASME 2017 Dynamic Systems and Control Conference*, vol. 1, October 11-13 2017, p. V001T30A001.

[3] M. Bagheri, M. Krstić, and P. Naseradinmousavi, "Joint-space trajectory optimization of a 7-dof baxter using multivariable extremum seeking," in *2018 Annual American Control Conference (ACC)*, Milwaukee, WI, June 27-29 2018, pp. 2176–2181.

[4] ——, "Multivariable extremum seeking for joint-space trajectory optimization of a high-degrees-of-freedom robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 11, p. 111017, 2018.

[5] ——, "Analytical and experimental predictor-based time delay control of baxter robot," in *ASME 2018 Dynamic Systems and Control Conference*, vol. 1, Oct. 2018, p. V001T04A011.

[6] M. Bagheri, P. Naseradinmousavi, and M. Krstić, "Feedback linearization based predictor for time delay control of a high-dof robot manipulator," *Automatica*, vol. 108, p. 108485, 2019.

[7] A. Bertino, M. Bagheri, M. Krstić, and P. Naseradinmousavi, "Experimental autonomous deep learning-based 3d path planning for a 7-dof robot manipulator," in *ASME 2019 Dynamic Systems and Control Conference*, vol. 2, Oct. 2019, p. V002T14A002.

[8] M. Bagheri, P. Naseradinmousavi, and M. Krstić, "Time delay control of a high-DOF robot manipulator through feedback linearization based predictor," in *ASME 2019 Dynamic Systems and Control Conference*, vol. 3. American Society of Mechanical Engineers, oct 2019, p. V003T16A001.

[9] M. Bagheri, P. Naseradinmousavi, and M. Krstić, "Time delay control of a high-dof robot manipulator through feedback linearization based predictor," in *Dynamic Systems and*

Control Conference, vol. 59162. American Society of Mechanical Engineers, 2019, p. V003T16A001.

[10] A. Bertino, P. Naseradinmousavi, and A. Kelkar, "Analytical and experimental decentralized adaptive control of a high-degrees-of-freedom robot manipulator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 143, no. 7, p. 071007, 2021.

[11] A. Bertino, P. Naseradinmousavi, and M. Krstic, "Experimental and analytical delay-adaptive control of a 7-dof robot manipulator," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 72–77.

[12] A. Bertino, P. Naseradinmousavi, and A. Kelkar, "Experimental and analytical decentralized adaptive control of a 7-dof robot manipulator," in *Dynamic Systems and Control Conference*, vol. 84270. American Society of Mechanical Engineers, 2020, p. V001T05A004.

[13] A. Bertino, P. Naseradinmousavi, and M. Krstić, "Delay-adaptive control of a 7-dof robot manipulator: Design and experiments," *IEEE Transactions on Control Systems Technology*, 2022.

[14] J. Alvarez-Ramirez, V. Santibanez, and R. Campa, "Stability of robot manipulators under saturated pid compensation," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1333–1341, Nov 2008.

[15] Youngjin Choi, Wan Kyun Chung, and Il Hong Suh, "Performance and h/sub /spl infin// optimality of pid trajectory tracking controller for lagrangian systems," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 857–869, Dec 2001.

[16] M. A. Jarrah and O. M. Al-Jarrah, "Position control of a robot manipulator using continuous gain scheduling," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, 1999, pp. 170–175 vol.1.

[17] Ya Lei Sun and Meng Joo Er, "Hybrid fuzzy control of robotics systems," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 6, pp. 755–765, 2004.

[18] A. Karakasoglu, S. I. Sudharsanan, and M. K. Sundareshan, "Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 919–930, Nov 1993.

[19] K. K. Tan, S. Huang, and T. H. Lee, "Decentralized adaptive controller design of large-scale uncertain robotic systems," *Automatica*, vol. 45, no. 1, pp. 161–166, 2009.

[20] Z. Yang, Y. Fukushima, and P. Qin, "Decentralized adaptive robust control of robot manipulators using disturbance observers," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1357–1365, Sep. 2012.

[21] M. K. Sundareshan and M. A. Koenig, "Decentralized model reference adaptive control of robotic manipulators," in *1985 American Control Conference*. IEEE, 1985, pp. 44–49.

[22] H. Seraji, "Decentralized adaptive control of manipulators: theory, simulation, and experimentation," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 183–201, April 1989.

[23] R. Colbaugh, H. Seraji, and K. Glass, "Decentralized adaptive control of manipulators," *Journal of Robotic Systems*, vol. 11, no. 5, pp. 425–440, 1994. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.4620110508

[24] M. Tarokh, "Decentralized adaptive tracking control of robot manipulators," *Journal of robotic systems*, vol. 13, no. 12, pp. 803–816, 1996.

[25] M. Liu, "Decentralized control of robot manipulators: nonlinear and adaptive approaches," *IEEE Transactions on Automatic control*, vol. 44, no. 2, pp. 357–363, 1999.

[26] Y. H. Chen, "Decentralized adaptive robust control design: The uncertainty is time varying," *Journal of Dynamic Systems, Measurement, and Control*, vol. 113, no. 3, pp. 515–518, sep 1991.

[27] C. Hua, X. Guan, and P. Shi, "Robust decentralized adaptive control for interconnected systems with time delays," *Journal of Dynamic Systems, Measurement, and Control*, vol. 127, no. 4, pp. 656–662, mar 2005.

[28] J. Lyou and Z. Bien, "Decentralized adaptive stabilization of a class of large-scale interconnected discrete systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 106–109, mar 1985.

[29] Y. Li, Z. Lu, F. Zhou, B. Dong, K. Liu, and Y. Li, "Decentralized trajectory tracking control for modular and reconfigurable robots with torque sensor: Adaptive terminal sliding control-based approach," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 6, feb 2019.

[30] R. Hernández-Alemán, O. Salas-Peña, and J. D. León-Morales, "Decentralized formation control based on adaptive super twisting," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 4, feb 2017.

[31] K. Watanabe, "A decentralized multiple model adaptive filtering for discrete-time stochastic systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 111, no. 3, pp. 371–377, sep 1989.

[32] J.-J. Yan, "Memoryless adaptive decentralized sliding mode control for uncertain large-scale systems with time-varying delays," *Journal of Dynamic Systems, Measurement, and Control*, vol. 125, no. 2, pp. 172–176, jun 2003.

[33] S. H. Hashemipour, N. Vasegh, and A. K. Sedigh, "Decentralized MRAC for large-scale interconnected systems with state and input delays by integrators inclusion," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 9, jun 2017.

[34] A. Elmahdi, A. F. Taha, D. Sun, and J. H. Panchal, "Decentralized control framework and stability analysis for networked control systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, may 2015.

[35] A. H. Ghasemi, J. B. Hoagg, and T. M. Seigler, "Decentralized vibration and shape control of structures with colocated sensors and actuators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 3, jan 2016.

[36] M. Yeatman, G. Lv, and R. D. Gregg, "Decentralized passivity-based control with a generalized energy storage function for robust biped locomotion," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 10, jun 2019.

[37] C. Wang and D. Li, "Decentralized PID controllers based on probabilistic robustness," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, no. 6, nov 2011.

[38] S. T. Kalat, S. G. Faal, and C. D. Onal, "A decentralized, communication-free force distribution method with application to collective object manipulation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 9, apr 2018.

[39] Z. Kan, J. R. Klotz, J. M. Shea, E. A. Doucette, and W. E. Dixon, "Decentralized rendezvous of nonholonomic robots with sensing and connectivity constraints," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 2, nov 2016.

[40] P. R. Pagilla and Y. Zhu, "A decentralized output feedback controller for a class of large-scale interconnected nonlinear systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 127, no. 1, pp. 167–172, apr 2004.

[41] M. Pavone and E. Frazzoli, "Decentralized policies for geometric pattern formation and path coverage," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 633–643, oct 2006.

[42] B. Brahmi, A. Brahmi, M. Saad, G. Gauthier, and M. H. Rahman, "Robust adaptive tracking control of uncertain rehabilitation exoskeleton robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 12, sep 2019.

[43] Y. Al Younes, A. Drak, H. Noura, A. Rabhi, and A. El Hajjaji, "Robust model-free control applied to a quadrotor uav," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 37–52, 2016.

[44] J. Villagra, C. Join, R. Haber, and M. Fliess, "Model-free control for machine tools," *arXiv preprint arXiv:2005.08546*, 2020.

[45] D. K. Han and P.-h. Chang, "Robust tracking of robot manipulator with nonlinear friction using time delay control with gradient estimator," *Journal of mechanical science and technology*, vol. 24, no. 8, pp. 1743–1752, 2010.

[46] W. S. Kim, B. Hannaford, and A. K. Bejczy, "Force-reflection and shared compliant control in operating telemanipulators with time delay," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 2, pp. 176–185, 1992.

[47] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," in *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1. IEEE, 1988, pp. 131–138.

[48] T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Transactions on robotics and Automation*, vol. 9, no. 5, pp. 592–606, 1993.

[49] A. K. Bejczy, W. S. Kim, and S. C. Venema, "The phantom robot: predictive displays for teleoperation with time delay," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 546–551.

[50] E. Nuño, L. Basañez, R. Ortega, and M. W. Spong, "Position tracking for non-linear teleoperators with variable time delay," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 895–910, 2009.

[51] A. Galip Ulsoy, "Time-delayed control of siso systems for improved stability margins," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 4, 2015.

[52] Z. Artstein, "Linear systems with delayed controls: a reduction," *IEEE Transactions on Automatic control*, vol. 27, no. 4, pp. 869–879, 1982.

[53] W. Kwon and A. Pearson, "Feedback stabilization of linear systems with delayed control," *IEEE Transactions on Automatic control*, vol. 25, no. 2, pp. 266–269, 1980.

[54] A. Manitius and A. Olbrot, "Finite spectrum assignment problem for systems with delays," *IEEE transactions on Automatic Control*, vol. 24, no. 4, pp. 541–552, 1979.

[55] I. Karafyllis, "Stabilization by means of approximate predictors for systems with delayed input," *SIAM Journal on Control and Optimization*, vol. 49, no. 3, pp. 1100–1123, 2011.

[56] F. Mazenc and S.-I. Niculescu, "Generating positive and stable solutions through delayed state feedback," *Automatica*, vol. 47, no. 3, pp. 525–533, 2011.

[57] S. Mondié and W. Michiels, "A safe implementation for finite spectrum assignment: robustness analysis," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 5. IEEE, 2003, pp. 4569–4574.

[58] T. G. Molnár, W. B. Qin, T. Insperger, and G. Orosz, "Application of predictor feedback to compensate time delays in connected cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 545–559, 2017.

[59] M. Ariola and A. Pironti, "H∞ optimal terminal control for linear systems with delayed states and controls," *Automatica*, vol. 44, no. 10, pp. 2676–2679, 2008.

[60] S. Gumussoy and H. Özbay, "Stable h∞ controller design for time-delay systems," *International Journal of Control*, vol. 81, no. 4, pp. 546–556, 2008.

[61] Y. Yildiz, A. Annaswamy, I. V. Kolmanovsky, and D. Yanakiev, "Adaptive posicast controller for time-delay systems with relative degree n$*\leq$ 2," *Automatica*, vol. 46, no. 2, pp. 279–289, 2010.

[62] M. Krstic, *Delay compensation for nonlinear, adaptive, and PDE systems*. Springer, 2009.

[63] F. Mazenc, S.-I. Niculescu, and M. Krstic, "Some remarks on lyapunov-krasovskii functionals and reduction approach for input delay compensation," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 6229–6234.

[64] Z.-L. Tang, S. S. Ge, and W. He, "Predictive control for a class of nonholonomic constrained mechanical systems with input delay," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*. IEEE, 2014, pp. 879–884.

[65] N. Bekiaris-Liberis and M. Krstic, "Compensation of time-varying input and state delays for nonlinear systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 1, 2012.

[66] ——, *Nonlinear control under nonconstant delays*. SIAM, 2013.

[67] M. Nihtila, "Finite pole assignment for systems with time-varying input delays," in *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*. IEEE, 1991, pp. 927–928.

[68] X.-G. Yan, S. K. Spurgeon, and Y. Orlov, "Sliding mode observer based-controller design for nonlinear systems with time varying delay," in *Recent Results on Nonlinear Delay Control Systems*. Springer, 2016, pp. 347–365.

[69] X.-G. Yan, S. K. Spurgeon, and C. Edwards, "Sliding mode control for time-varying delayed systems based on a reduced-order observer," *Automatica*, vol. 46, no. 8, pp. 1354–1362, 2010.

[70] Y. Zhu and M. Krstic, *Delay-adaptive linear control*. Princeton University Press, 2020, vol. 70.

[71] Y. Zhu and E. Fridman, "Sub-predictors for network-based control under uncertain large delays," *Automatica*, vol. 123, p. 109350, 2021.

[72] Y. Zhu, M. Krstic, and H. Su, "Adaptive output feedback control for uncertain linear time-delay systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 545–560, 2016.

[73] ——, "PDE boundary control of multi-input LTI systems with distinct and uncertain input delays," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4270–4277, 2018.

[74] M. Krstic and A. Banaszuk, "Multivariable adaptive control of instabilities arising in jet engines," *Control Engineering Practice*, vol. 14, no. 7, pp. 833–842, 2006.

[75] D. Bresch-Pietri, J. Chauvin, and N. Petit, "Adaptive control scheme for uncertain time-delay systems," *Automatica*, vol. 48, no. 8, pp. 1536–1552, 2012.

[76] D. Bresch-Pietri and M. Krstic, "Adaptive trajectory tracking despite unknown input delay and plant parameters," *Automatica*, vol. 45, no. 9, pp. 2074–2081, 2009.

[77] ——, "Delay-adaptive predictor feedback for systems with unknown long actuator delay," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2106–2112, 2010.

[78] ——, "Delay-adaptive control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1203–1218, 2014.

[79] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[80] E. Polak, *Optimization: algorithms and consistent approximations*. Springer Science & Business Media, 2012, vol. 124.

[81] G. Tao and P. V. Kokotovic, *Adaptive control of systems with actuator and sensor nonlinearities*. John Wiley & Sons, Inc., 1996.

[82] A. O'Dwyer, "A survey of techniques for the estimation and compensation," 2000.

[83] S. I. Han and J. Lee, "Finite-time sliding surface constrained control for a robot manipulator with an unknown deadzone and disturbance," *ISA transactions*, vol. 65, pp. 307–318, 2016.

[84] Y. Hong, Y. Xu, and J. Huang, "Finite-time control for robot manipulators," *Systems & control letters*, vol. 46, no. 4, pp. 243–253, 2002.

[85] S. Yu, X. Yu, B. Shirinzadeh, and Z. Man, "Continuous finite-time control for robotic manipulators with terminal sliding mode," *Automatica*, vol. 41, no. 11, pp. 1957–1964, 2005.

[86] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8112–8123, 2018.

[87] D. Zhao, S. Li, Q. Zhu, and F. Gao, "Robust finite-time control approach for robotic manipulators," *IET control theory & applications*, vol. 4, no. 1, pp. 1–15, 2010.

[88] Y. Wang, M. Chen, and Y. Song, "Robust fixed-time inverse dynamic control for uncertain robot manipulator system," *Complexity*, vol. 2021, 2021.

[89] D. Zhang, L. Kong, S. Zhang, Q. Li, and Q. Fu, "Neural networks-based fixed-time control for a robot with uncertainties and input deadzone," *Neurocomputing*, vol. 390, pp. 139–147, 2020.

[90] Y. Su, C. Zheng, and P. Mercorelli, "Robust approximate fixed-time tracking control for uncertain robot manipulators," *Mechanical Systems and Signal Processing*, vol. 135, p. 106379, 2020.

[91] X. Jin, "Adaptive fixed-time control for mimo nonlinear systems with asymmetric output constraints using universal barrier functions," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 3046–3053, 2018.

[92] L. Zhang, Y. Wang, Y. Hou, and H. Li, "Fixed-time sliding mode control for uncertain robot manipulators," *IEEE Access*, vol. 7, pp. 149 750–149 763, 2019.

[93] N. Espitia, A. Polyakov, D. Efimov, and W. Perruquetti, "Boundary time-varying feedbacks for fixed-time stabilization of constant-parameter reaction–diffusion systems," *Automatica*, vol. 103, pp. 398–407, 2019.

[94] K. Garg and D. Panagou, "Robust control barrier and control lyapunov functions with fixed-time convergence guarantees," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 2292–2297.

[95] Y. Song, Y. Wang, J. Holloway, and M. Krstic, "Time-varying feedback for regulation of normal-form nonlinear systems in prescribed finite time," *Automatica*, vol. 83, pp. 243–251, 2017.

[96] Y. Song, Y. Wang, and M. Krstic, "Time-varying feedback for stabilization in prescribed finite time," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 3, pp. 618–633, 2019.

[97] H. M. Becerra, C. R. Vázquez, G. Arechavaleta, and J. Delfin, "Predefined-time convergence control for high-order integrator systems using time base generators," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1866–1873, 2017.

[98] J. Obregón-Flores, G. Arechavaleta, H. M. Becerra, and A. Morales-Díaz, "Predefined-time robust hierarchical inverse dynamics on torque-controlled redundant manipulators," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 962–978, 2021.

[99] A. J. Muñoz-Vázquez, J. D. Sánchez-Torres, E. Jimenez-Rodriguez, and A. G. Loukianov, "Predefined-time robust stabilization of robotic manipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1033–1040, 2019.

[100] J. D. Sánchez-Torres, A. J. Muñoz-Vázquez, M. Defoort, E. Jiménez-Rodríguez, and A. G. Loukianov, "A class of predefined-time controllers for uncertain second-order systems," *European Journal of Control*, vol. 53, pp. 52–58, 2020.

[101] Y. Cao and Y.-D. Song, "Adaptive pid-like fault-tolerant control for robot manipulators with given performance specifications," *International Journal of Control*, vol. 93, no. 3, pp. 377–386, 2020.

[102] Y. Cao, J. Cao, and Y. Song, "Practical prescribed time control of euler-lagrange systems with partial/full state constraints: A settling time regulator-based approach," *IEEE Transactions on Cybernetics*, 2021.

[103] P. Krishnamurthy, F. Khorrami, and M. Krstic, "Adaptive output-feedback stabilization in prescribed time for nonlinear systems with unknown parameters coupled with unmeasured states," *International Journal of Adaptive Control and Signal Processing*, vol. 35, no. 2, pp. 184–202, 2021.

[104] W. Li and M. Krstic, "Stochastic nonlinear prescribed-time stabilization and inverse optimality," *IEEE Transactions on Automatic Control*, 2021.

[105] D. Steeves, M. Krstic, and R. Vazquez, "Prescribed–time estimation and output regulation of the linearized schrödinger equation by backstepping," *European Journal of Control*, vol. 55, pp. 3–13, 2020.

[106] J. Holloway and M. Krstic, "Prescribed-time observers for linear systems in observer canonical form," *IEEE Transactions on Automatic Control*, vol. 64, no. 9, pp. 3905–3912, 2019.

[107] P. Krishnamurthy, F. Khorrami, and M. Krstic, "A dynamic high-gain design for prescribed-time regulation of nonlinear systems," *Automatica*, vol. 115, p. 108860, 2020.

[108] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[109] C.-S. Chung, H. Wang, and R. A. Cooper, "Functional assessment and performance evaluation for assistive robotic manipulators: Literature review," *The journal of spinal cord medicine*, vol. 36, no. 4, pp. 273–289, 2013.

[110] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26 754–26 773, 2017.

[111] A. Bauer, D. Wollherr, and M. Buss, "Human–robot collaboration: a survey," *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 47–66, 2008.

[112] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[113] A. Singletary, S. Kolathaya, and A. D. Ames, "Safety-critical kinematic control of robotic systems," *IEEE Control Systems Letters*, vol. 6, pp. 139–144, 2021.

[114] H. Wang, J. Peng, F. Zhang, H. Zhang, and Y. Wang, "High-order control barrier functions-based impedance control of a robotic manipulator with time-varying output constraints," *ISA transactions*, 2022.

[115] C. T. Landi, F. Ferraguti, S. Costi, M. Bonfè, and C. Secchi, "Safety barrier functions for human-robot interaction with industrial manipulators," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2565–2570.

[116] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: a control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[117] Y. Wang and X. Xu, "Disturbance observer-based robust control barrier functions," *arXiv preprint arXiv:2203.12855*, 2022.

[118] M. Rauscher, M. Kimmel, and S. Hirche, "Constrained robot control using control barrier functions," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 279–285.

[119] A. W. Farras and T. Hatanaka, "Safe control with control barrier function for euler-lagrange systems facing position constraint," in *2021 SICE International Symposium on Control Systems (SICE ISCS)*. IEEE, 2021, pp. 28–32.

[120] E. A. Basso and K. Y. Pettersen, "Task-priority control of redundant robotic systems using control lyapunov and control barrier function based quadratic programs," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9037–9044, 2020.

[121] F. Ferraguti, M. Bertuletti, C. T. Landi, M. Bonfè, C. Fantuzzi, and C. Secchi, "A control barrier function approach for maximizing performance while fulfilling to iso/ts 15066 regulations," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5921–5928, 2020.

[122] W. S. Cortez and D. V. Dimarogonas, "Correct-by-design control barrier functions for euler-lagrange systems with input constraints," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 950–955.

[123] M. Lippi and A. Marino, "A control barrier function approach to human-multi-robot safe interaction," in *2021 29th Mediterranean Conference on Control and Automation (MED)*. IEEE, 2021, pp. 604–609.

[124] V. Kurtz, P. M. Wensing, and H. Lin, "Control barrier functions for singularity avoidance in passivity-based manipulator control," *arXiv preprint arXiv:2109.13349*, 2021.

[125] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe multi-agent interaction through robust control barrier functions with learned uncertainties," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 777–783.

[126] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 4542–4548.

[127] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[128] V. R. Kamidi, J. Kim, R. T. Fawcett, A. D. Ames, and K. A. Hamed, "Distributed quadratic programming-based nonlinear controllers for periodic gaits on legged robots," *IEEE Control Systems Letters*, 2022.

[129] R. T. Fawcett, A. Pandala, A. D. Ames, and K. A. Hamed, "Robust stabilization of periodic gaits for quadrupedal locomotion via qp-based virtual constraint controllers," *IEEE Control Systems Letters*, vol. 6, pp. 1736–1741, 2021.

[130] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 322–328.

[131] I. Abel, D. Steeves, and M. Krstic, "Prescribed-time safety design for a chain of integrators," *arXiv preprint arXiv:2201.09447*, 2022.

[132] A. Bertino, P. Naseradinmousavi, and M. Krstic, "Experimental and analytical prescribed-time trajectory tracking control of a 7-dof robot manipulator," in *2022 American Control Conference (ACC)*. IEEE, 2022.